

Lowering the implementation barrier of neutral-atom quantum computing with agentic workflows

Constantin Dalyac ¹, Alexandre Dauphin ¹, Loïc Henriët ¹ and Christophe Jurczak ²

¹*PASQAL SAS, 24 rue Emile Baudot, 91120 Palaiseau, France*

²*Quantonation Ventures, Dallas, TX 75206, USA*

(Dated: July 28, 2026)

Quantum computers are moving from research laboratories to industrial machines accessible via the cloud and integrated into high-performance computing facilities. However, translating theoretical quantum protocols into hardware experiments remains a major bottleneck, requiring expertise across protocol design, compilation, simulation, and cloud execution. Here, we introduce an agentic workflow that automates this pipeline on neutral-atom quantum processors (here two Pasqal QPUs available on the cloud) while keeping the researcher in the loop for critical validation. In three case studies from many-body physics and optimization, the agent went from published paper or patent to a QPU campaign run overnight. In particular, human intervention was crucial to ensure scientific validity: the agent selected an inadequate observable in one experiment and constructed a plausible but incorrect hardware diagnosis in another, with both failures detected only through domain-expert review. Finally, we use a second agent to classify a corpus of 633 Rydberg-array arXiv papers and show that nearly half are implementable on present-day QPUs while identifying specific hardware upgrades needed for the rest. Together, these results demonstrate that agentic workflows provide a practical bridge between theoretical ideas and physical hardware, opening quantum experimentation to a much broader scientific community.

I. INTRODUCTION

Validating a quantum-computing application ultimately means confronting a theoretical proposal with physical hardware. For much of the field’s history, the binding constraint on that step was hardware availability: quantum processors were scarce, laboratory-bound systems accessible mainly to the groups that built them. Today, however, quantum processing units (QPUs) are increasingly stable, remotely accessible resources embedded in cloud and high-performance-computing ecosystems. Neutral-atom platforms are a representative case: arrays of individually controlled atoms interacting through Rydberg states realize programmable many-body Hamiltonians [1], and scale naturally into the 100–1000-qubit range with flexible geometries and both analog and digital modes of operation [2, 3]. This versatility supports applications spanning quantum simulation, graph machine learning and combinatorial optimization [4, 5].

Accessible hardware, however, does not imply easy experimentation. Between a published protocol and a QPU lies a long chain of research and engineering tasks: extracting the procedure from the literature, translating it into device-level controls, validating it in a simulation, navigating cloud-execution infrastructure, and interpreting noisy data. Each step is individually well understood and the ensemble is well supported by a software stack, but the full chain demands fluency across quantum physics, software engineering, numerical simulation, cloud computing, and hardware characterization. The resulting *implementation barrier* often exceeds the effort of formulating the original scientific idea, leaving many physically viable proposals unexplored despite suitable hardware being available.

This barrier is, at its core, a coordination problem across specialized domains, precisely the kind of problem that recent agents powered by large language models (LLM) have begun to address. Agentic systems can orchestrate heterogeneous software tools, scientific workflows, and laboratory infrastructure [6–8], carrying a task across interfaces that have traditionally required separate human specialists. Rather than replacing scientific judgment, such systems offer a way to fold expertise distributed across technical layers into a single operational workflow.

Building on this observation, our contributions are threefold. First, we introduce an agentic workflow that carries a quantum-computing task all the way from a prompt to hardware execution. The workflow is tailored for Pasqal’s QPUs, and executes specifically on FC1 and SA1, which are located respectively in Canada and in Saudi Arabia. Second, we validate the framework across three increasingly difficult applications: two many-body quantum simulations and a graph-coloring algorithm. Finally, we apply a separate, lighter agentic workflow to the literature itself: it screens arXiv for Rydberg-array theory papers compatible with the platform, using back-of-the-envelope rather than detailed calculations. Of the 526 papers classified, roughly half are viable starting points for experimentation on present-day hardware. For the remaining papers, the workflow identifies the hardware specifications that would need to be reached, providing direct feedback for the design of future hardware.

II. AGENTIC WORKFLOW

We start by describing known agentic workflows that have been developed for scientific research, before pre-

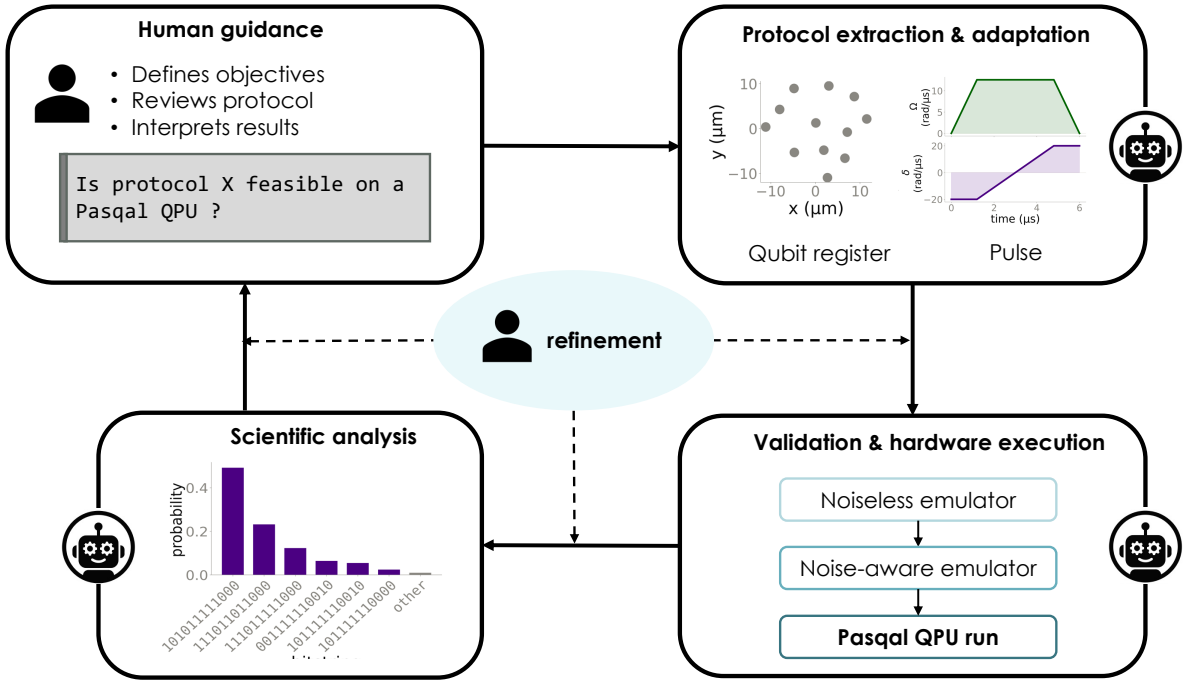


FIG. 1. **AI-assisted workflow for translating quantum protocols into hardware experiments.** A researcher defines the scientific objective and interacts with the AI workflow through natural-language instructions. The AI extracts or generates an experimental protocol, including the neutral-atom register and pulse sequence, from a scientific publication, patent, or user request. The pulse sequence is determined by the qubit register, the time evolution of the Rabi frequency Ω ($\text{rad}\cdot\mu\text{s}^{-1}$) and the detuning δ ($\text{rad}\cdot\mu\text{s}^{-1}$). Before execution on quantum hardware, the protocol is validated using noiseless and noise-aware emulation to identify implementation issues and support iterative refinement. Once validated, the experiment is executed on a cloud-accessible neutral-atom quantum processor. The resulting measurement data are automatically processed into scientific observables, which the researcher interprets to assess the outcome and, when necessary, refine the protocol for subsequent iterations. The workflow keeps the researcher in the decision loop while automating the engineering tasks required to bridge the gap between a scientific idea and a hardware experiment.

senting our specific workflow for this study, presented at a high-level in Fig. 1.

A. Prior work

LLM agents are now used to run increasingly autonomous scientific workflows, in which they coordinate tools, execute multi-step procedures, and interface with external computational or experimental infrastructure [9].

Coscientist [6], ChemCrow [7], and the AI Scientist [8] established the feasibility of agentic automation: the first two translate natural-language objectives into executable chemistry experiments through tool-augmented reasoning, while the AI Scientist automates machine-learning from a hypothesis to a manuscript. Role-specialised multi-agent systems (AtomAgents [10], ProtAgents [11]) outperform monolithic ones on design tasks, and self-driving laboratories [12] run modular instruments in closed loops (e.g. ChemOS 2.0 [13], with multi-agent benchmarks [14]).

Within quantum technologies, reinforcement-learning

controllers handle real-time qubit feedback [15] and error-correction stabilisation [16], and LLM-based program synthesis (Agent-Q [17], with retrieval [18] or simulator feedback [19]) generates circuits from specifications. Multi-agent architectures have similarly been applied to translate natural-language requirements into traceable quantum-application workflows [20]. Most closely, Shiraishi *et al.* [21] expose QPU and HPC execution to an agent through a Model Context Protocol server. These approaches lower the expertise to operate hardware but assume a well-defined target. AI-Mandel [22] goes one step further and lets the agent choose its own research question in quantum optics, using a domain-specific design tool to turn each idea into an implementable experiment.

As agentic workflows become autonomous, they also raise questions about the evolving role of the human scientist. Consequently, many recent frameworks emphasize the importance of maintaining meaningful human oversight throughout the research process [23, 24]. Systematic evaluations have shown that LLMs can generate persuasive yet fundamentally incorrect analyses due to hallucinations and flawed reasoning [25, 26]. Recent bench-

marks on open physics problems further demonstrate that even advanced autonomous agents struggle with the tacit domain knowledge, contextual judgment, and physical intuition required for scientific discovery [27]. We observe similar limitations in this study, particularly during result interpretation, and discuss these challenges throughout the case studies presented in this work.

B. Description of our agentic workflow

Our framework builds on Pasqal’s existing software stack: Pulser [28], an open-source Python library for designing pulse sequences on neutral-atom QPUs and their emulators; the matrix-product-state emulator emumps [29]; and the Pasqal Cloud SDK [30, 31], which provides programmatic access to the QPUs, their calibration data, and noise models.

On top of this stack, the workflow is implemented as a collection of skills, each responsible for one stage such as: translating a scientific source into a specification, compiling that specification into a pulse sequence, validating it in emulation, executing it on hardware, or analyzing the returned data (Fig. 1). The individual skills are detailed in Appendix A. We here use Anthropic’s Claude Code with the models Opus 4.8 and Fable 5.

The skills operate on a single structured artifact, a machine-readable experimental specification `experiment_spec.json` capturing the physical parameters, observables, and constraints of the experiment. This specification is the primary interface between researcher and agent: it can be inspected, modified or approved before anything is executed. By construction, any modification propagates to all downstream stages.

Importantly, no protocol reaches the QPU without passing through emulation. A candidate sequence is first simulated noiselessly to establish a reference prediction, then re-simulated with device-specific noise parameters retrieved from the cloud SDK. Comparing the two indicates whether the target observable is expected to survive realistic hardware conditions and helps the researcher in deciding to submit.

Finally, the skills can be composed into iterative refinement loops that follow the researcher–AI dialogue: validation or hardware results may motivate changes to the register geometry, pulse schedule, or target observable, upon which the agent regenerates the sequence and re-runs the relevant stages.

III. END-TO-END EXAMPLES

We present three examples of increasing complexity. First, we reproduce a well-known quantum simulation result expected to be achievable on FC1 and SA1 (Example A). Next, we examine a theoretical proposal to assess whether the predicted phase of matter can be realized on FC1 (Example B). Finally, we address an op-

timization application derived from a patent, specifically a graph-coloring algorithm (Example C).

Before turning to the examples, we briefly recall the control knobs these experiments are expressed in. A neutral-atom QPU realizes an Ising-like Hamiltonian whose terms map onto three handles: the *qubit register*, i.e. the positions of the atoms, which fix the pairwise interactions C_6/r_{ij}^6 through the chosen geometry; the *Rabi frequency* $\Omega(t)$, which coherently couples the ground and Rydberg states and sets the transverse field; and the *detuning* $\delta(t)$ of the driving laser from resonance, which acts as a longitudinal field controlling the Rydberg population. A protocol is thus a register plus time profiles $(\Omega(t), \delta(t))$, typically applied globally to all atoms. Each handle comes with limits: a maximum number of atoms and a minimum interatomic spacing for the register, caps on Ω and on the reachable detuning range, finite ramp rates, and a maximum sequence duration. These bounds are device-specific and evolve with hardware upgrades.

A. Reproduction of a literature hardware result

1. Context and task

Our objective is to reproduce known results that have already been run on hardware. We select a landmark paper by Keesling *et al.* [32] from 2019 where the authors study the Kibble-Zurek mechanism when driving a 1D array of atoms through a quantum phase transition from the paramagnetic to several ordered states, namely $\mathbb{Z}_2$, $\mathbb{Z}_3$ and $\mathbb{Z}_4$. From a domain expert point of view, we know that the quantum phase transition to $\mathbb{Z}_2$ is realizable but not the higher order ones.

We intentionally prompt the agent towards the $\mathbb{Z}_3$ phase to judge if the workflow can recognize hardware-imposed limitations. We use the following prompt:

Reproduce the $\mathbb{Z}_3$ density-wave results from Keesling et al. on FC1. Read the paper, identify the protocol for the $\mathbb{Z}_3$ phase, adapt it to FC1 constraints, and run the pipeline through EMU validation. If the EMU verdict is GO, propose a QPU submission plan. If any aspect of the paper’s protocol cannot be faithfully reproduced on FC1, document the deviation and justify whether it preserves the essential physics.

2. Workflow and dialogue

The agent extracts the protocol directly from the paper: a 51-atom 1D chain, the blockade ratio R_b/a defining the ordered lobe ($R_b/a \approx 2.16$ – 2.43 for $\mathbb{Z}_3$, with R_b the blockade radius and a the atomic spacing), and the pulse schedule together with the sweep rates tabulated

for each interaction range. It then fetches FC1’s live device specification through the cloud SDK and combines the two to reach the following verdict:

- (i) The $\mathbb{Z}_3$ phase is not native: reaching the targeted regime at the spacing floor forces $\Omega/2\pi$ down to 0.087 MHz, stretching the dynamical timescale so that neither one Rabi period nor the nearest-neighbour interaction time fit inside the sequence cap. By the same scaling argument, the still-higher $\mathbb{Z}_4$ phase places it further out of reach.
- (ii) The $\mathbb{Z}_2$ phase on the contrary is native: at the $5\mu\text{m}$ spacing floor and full Ω , $R_b/a = 1.28$ falls well inside the ordering lobe, and the paper’s slowest schedule fits the $6\mu\text{s}$ FC1 specification.

These observations fit very well with our intuition, confirming that the agent identifies correctly the hardware constraints.

Redirected to the $\mathbb{Z}_2$ phase, the agent uses the `noise_emulate` skill and obtains 80 % noise retention with the saturated order parameter, reaching 71 % of its ideal value. The order parameter is the density-density correlator introduced in the original work [32]

$$G(r) = \frac{1}{N_r} \sum_i \left(\langle n_i n_{i+r} \rangle - \langle n_i \rangle \langle n_{i+r} \rangle \right). \quad (1)$$

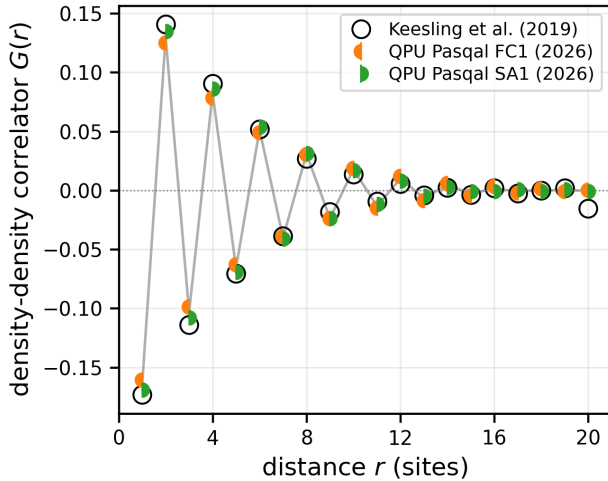


FIG. 2. Density-density correlator $G(r)$ in the saturated $\mathbb{Z}_2$ -ordered phase as a function of the distance r in number of sites. Open black circles: values taken from Fig. 1d of Keesling *et al.* [32] (51-atom chain, fitted $\xi = 3.9$ sites). Half-filled markers: this work, on a 56-atom ring on Pasqal QPU FC1 (orange, 2000 shots, $\xi = 4.2$) and a 96-atom ring on Pasqal QPU SA1 (green, 1000 shots, $\xi = 4.2$). The grey line is a guide to the eye. All three datasets show that the saturated ordered-phase correlator is a robust observable, reproduced by our agentic workflow on two Pasqal QPUs, 7 years after the landmark experiment.

Prompted to make better use of the field of view, we suggest replacing the open chain with a ring register to add atoms and remove edge effects. The agent then sizes a 56-atom ring at the spacing floor and reruns the pipeline. We furthermore note that the second Pasqal QPU SA1 offers a larger field of view. The agent sizes a 96-atom ring for it.

In Fig. 2, we show that the reconstructed correlator from the 2019 apparatus and from both 2026 devices coincide to within statistical error at all 20 lattice sites probed, with fitted correlation lengths of $\xi = 3.9$ (paper), 4.2 (FC1) and 4.2 (SA1) sites.

3. Comments

This first example already demonstrates excellent agentic capability. In addition to independently constructing the noiseless-vs-ideal comparison, it also recovered from infrastructure failures (a paused QPU, errors in jobs) without derailing the broader campaign. Agents have the healthy habit of consistently adding checkpoints, error logs and try/except errors that help recover from failures. Most noticeably, the pipeline was extremely fast: in less than an hour and three researcher prompts the agentic workflow had QPU jobs up and running.

That said, the higher-level scientific moves that improved the outcome, such as switching to a ring register or extending the study to a second QPU originated from the researcher. Note that every such piece of input is retained by the agentic workflow, so that future experiments start from an accumulated body of domain know-how rather than repeating the same reasoning from scratch. Once directed, the agent executed each multi-hour, multi-device campaign autonomously.

B. Preparation of frustrated quantum magnetic phases on a triangular lattice

We now turn to a harder use-case for which a theory paper exists but no QPU implementation is proposed.

1. Context and task

Guo *et al.* [33] explore the quantum magnetic phases of a frustrated triangular Rydberg-atom array. Using quantum Monte Carlo, the authors map out a phase diagram containing three ordered lobes as a function of the reduced detuning δ/Ω : a $1/3$ -filling plateau, a $1/2$ -filling “order-by-disorder” phase known to be experimentally fragile, and a $2/3$ -filling plateau.

We give the agent a deliberately open-ended system prompt. This tests whether it can autonomously isolate the relevant physical variables and map out an experimental realisation:

Read Guo et al. paper (link provided). For each distinct phase of matter described in the paper, assess whether it can realistically be prepared and observed on Pasqal QPU FC1 using our pipeline. Produce a verdict table (feasible / marginal / infeasible) with your reasoning. For every phase you judge feasible or marginal, output a complete experiment_spec.json ready for spec-to-sequence.

From an expert perspective, preparing a frustrated ground state on hardware requires several trade-offs: choosing commensurate system sizes, designing an adiabatic annealing trajectory, and tracking a combination of observables sufficient to certify the phase unambiguously. Our goal is to assess how effectively the agent navigates these nuances.

2. Workflow and dialogue

Invoking the **paper-to-spec** skill, the agent correctly parsed the paper’s core results, identifying the 1/3 and 2/3 plateaus alongside the delicate 1/2-filling order-by-disorder phase. Its verdict table again demonstrated a firm grasp of the hardware constraints. The decisive limitation is FC1’s maximum detuning, $|\delta|/2\pi \leq 10$ MHz: since each lobe is specified by a fixed reduced detuning δ/Ω , reaching a deeper lobe forces a proportionally smaller Rabi frequency, and hence a longer adiabatic ramp measured against a fixed coherence time. Working from the tunable knobs the agent reasoned that

- (i) the 1/3-filling phase ($\delta_{c1}/\Omega = 1.22$) is *feasible*, since its lobe is reached at the full $\Omega/2\pi = 2$ MHz;
- (ii) the 1/2-filling phase ($\delta/\Omega \simeq 9.5$) is only *marginally* feasible, as it requires halving the Rabi frequency to $\Omega/2\pi \simeq 1$ MHz, which lengthens the ramp and risks non-adiabatic transitions out of the ground state;
- (iii) the 2/3-filling phase ($\delta_{c2}/\Omega = 17.8$) is *infeasible*, as it would demand $\Omega/2\pi \lesssim 0.6$ MHz, a ramp far too slow compared with the atomic coherence time.

We instructed the agent to generate and test **Pulser** sequences preparing the viable phases. Left to its own devices, the agent exhibited a form of cognitive narrowing: of the four observables detailed in the original paper it retained only the structure factor and the mean magnetisation. While the structure factor is an appropriate two-body diagnostic, the mean magnetisation is not the one-body order parameter of the paper, and cannot distinguish the particular pattern from any other state at the same density. The agent also overlooked the finite-size scaling arguments that are central to interpreting

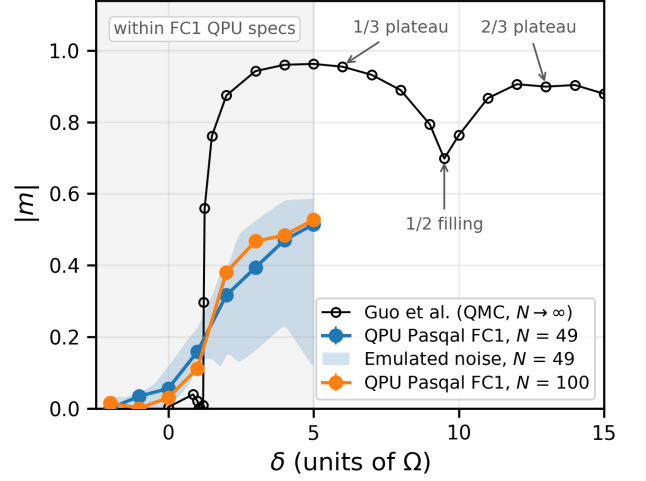


FIG. 3. Normalised order parameter $|m|$ versus δ/Ω on the bulk sublattice. Open black circles: QMC in the thermodynamic limit (Guo et al. [33]). Filled circles: QPU Pasqal FC1 at $N = 49$ (blue) and $N = 100$ (orange), on rhombic $L \times L$ triangular registers with $L \equiv 1 \pmod{3}$, so the open boundary pins a single $\sqrt{3}$ sublattice. Shaded band: MPS emulation of the $N = 49$ register including the calibrated FC1 noise model ($\pm 2\sigma$ over 40 quantum-jump trajectories). All data share $a = 5.3 \mu\text{m}$ ($R_b/a = 1.21$), $\Omega/2\pi = 2$ MHz, and a $4 \mu\text{s}$ linear detuning ramp from $\delta_i = -4$ MHz, with 300 shots per point.

small registers. Guided by the researcher on both points, it identified the staggered magnetisation

$$m = \frac{3}{N} \sum_{j=1}^N n_j e^{i\mathbf{Q} \cdot \mathbf{r}_j}, \quad \mathbf{Q} = \left(\frac{2\pi}{3a}, \frac{2\pi}{\sqrt{3}a} \right), \quad (2)$$

where $n_j \in \{0, 1\}$ is the Rydberg occupation of atom j and the prefactor normalises perfect 1/3 order to $|m| = 1$. Because the phase is three-fold degenerate, the sublattice selected varies from shot to shot and $\langle m \rangle$ averages to zero, hence the importance of selecting the shot-resolved magnitude $\langle |m| \rangle$. We then directed the agent to design a scaling study for commensurate rhombic registers ($L \times L$ with $L \equiv 1 \pmod{3}$), yielding valid sizes $N = 49$ ($L = 7$) and $N = 100$ ($L = 10$). The resulting QPU data (Fig. 3) clearly display the 1/3 plateau. A deficit of roughly half the QMC amplitude arises from finite-size rounding, diabatic ramp excitations, and hardware noise. The emulated noise faithfully envelopes the QPU datapoints.

3. Comments

The agent demonstrated a significant ability to extract, translate, and execute a hardware experiment directly from a theory paper, eliminating tedious boilerplate script generation and hardware parameter mapping to sharply reduce the time-to-QPU.

However, we observed a key failure mode: the agent consistently optimized for the nearest tractable sub-task at the expense of the core physics, selecting the observable that was easiest to compute rather than the one that genuinely certifies the phase. Crucially, because the pipeline ran smoothly and generated plausible-looking data, only domain expertise could identify that the wrong physical quantity was being measured. Consequently, while Example A required just 3 human messages, this example required 43 exchanges to reach a physically accurate implementation. This highlights that agentic workflows must be evaluated not merely on execution success, but on physical accuracy.

Finally, these experiments clarify current QPU boundaries. The 1/2- and 2/3-filling lobes are restricted here solely by the maximum detuning $|\delta|_{\max}$, and could be unlocked either by expanding the detuning range or rescaling the blockade ratio to shift the lobes to smaller δ/Ω , a regime already accessed on academic setups [34].

C. Patent implementation: graph coloring via iterative Rydberg IS peeling

We now move to our last example, taken from combinatorial optimization. The source here is distinctive: not a journal paper but a Pasqal *patent*. This tests whether the same workflow mines patents as readily as the published literature, a second and largely untapped corpus of hardware-ready protocols. The patent specifies the algorithm but not its realization on present hardware, so the agent must bridge from a protocol description to a working multi-round pipeline.

1. Context and task

Graph coloring (assigning colors to the vertices of $G = (V, E)$ such that adjacent vertices receive different colors) is equivalent to a sequence of independent-set (IS) problems: peel one IS per round until all vertices are covered, with the round count upper-bounding the chromatic number $\chi(G)$. A *project-and-expel* algorithm maps this decomposition onto the native Rydberg IS primitive [35, 36]: starting from all N atoms in $|0\rangle$, a resonant global pulse drives the current sub-register into a superposition of independent sets; one measurement shot collapses onto a valid IS; those atoms are assigned a color and retired; the residual subgraph defines the next round.

Read the attached Pasqal patent describing the project-and-expel algorithm for estimating the chromatic number of a graph on a Rydberg array. Implement it on Pasqal QPU FC1: map the iterative IS-peeling decomposition onto the native

Rydberg IS primitive, handle the multi-round orchestration on present hardware, and benchmark the estimated chromatic number against a classical baseline on random unit-disk graphs. Document any hardware constraint that forces a deviation from the patent’s idealized procedure.

2. Workflow and dialogue

On present hardware, physical atom expulsion is replaced by a classical measurement-feedback loop: the agent compiles the full N -atom layout once into a **MappableRegister** batch and submits each round as a separate job within that open batch, activating only the subset of trap sites corresponding to the current subgraph. This eliminates re-queuing latency between rounds while holding a single QPU queue slot. To cope with finite shot counts, the agent runs a beam search (width $w=3$, $n_s=200$ shots per round per branch), maintaining three candidate colorings in parallel and retaining the best final result. A classical repair step corrects any blockade-imperfect shots that violate the IS constraint.

The multi-round orchestration rests on Pulser [28] for pulse-level compilation and the Pasqal Cloud SDK for batch lifecycle management. The use of **MappableRegister** and open-batch primitives was not specified a priori; the agent designed and refined this pipeline through iterative back-and-forth with the user, consulting the SDK and hardware documentation to identify constraints that do not surface programmatically—notably, the minimum trap-site filling fraction the cloud API enforces when the active subgraph shrinks in late rounds, which would otherwise cause job rejections.

We benchmark on random unit-disk graphs with $N \in \{15, 20, 30, 40, 50, 60, 70, 80\}$ atoms, 75 instances in total. The classical chromatic number χ (DSATUR) is 4–5 throughout, confirming that χ stays bounded as N grows from 15 to 80 for these sparse graphs. At $N=20$ the QPU achieves the exact chromatic number on all 10 instances (mean gap $\langle \hat{\chi} - \chi \rangle = 0.00$); the gap grows to at most 1.44 ± 0.50 at $N=80$, and never exceeds 2 across all 75 instances (overall mean 0.84). Figure 4 shows the scaling.

3. Comments

This example extends the workflow beyond a single-shot protocol to a *multi-round, state-dependent* pipeline: the agent compiles the register once, routes $\hat{\chi}$ sequential jobs within one open batch, manages beam-search branches across rounds, and aggregates the final coloring, an orchestration pattern that would require substantial bespoke engineering outside an agentic scaffold. It also

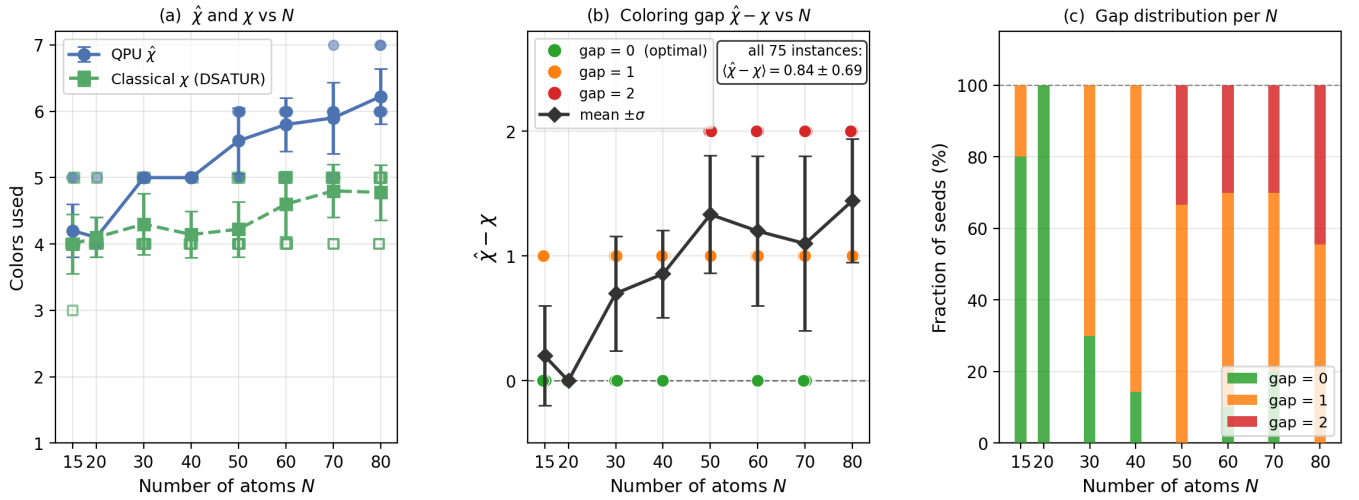


FIG. 4. Example C, QPU beam-search graph coloring on FC1 (unit-disk graphs at density 0.35, blockade radius $R_b=10\mu\text{m}$; beam-width $w=3$, $n_s=200$ shots/round, 75 instances). (a) QPU estimate $\hat{\chi}$ and classical optimum χ (DSATUR) vs. N ; lines show mean $\pm \sigma$, dots show per-instance values. (b) Coloring gap $\hat{\chi} - \chi$ per instance (coloured by gap value) and mean $\pm \sigma$; the gap is zero at $N=20$ and never exceeds 2. (c) Fraction of seeds at each gap value per N .

demonstrates that the corpus an agent can draw on is not limited to published papers: a patent, read directly, served as a complete and actionable protocol source.

As in the previous example, though, the agent required direction from the researcher and occasionally went astray. Confronted with an unexpectedly high rate of constraint-violating shots, the agent confidently attributed it to physics, diagnosing substantial Rydberg-blockade leakage on the QPU. The true cause was a mismatch between the ordering of the atomic coordinates and the ordering of the measured bitstrings, which scrambled the vertex labels and made valid independent sets appear to violate the adjacency constraints. The pathology lay in the workflow, neither the hardware nor the physics, and only expert scrutiny of the raw shot-to-graph mapping surfaced the labelling bug behind the agent’s plausible but incorrect physical story.

IV. CORPUS-SCALE ANALYSIS OF THE RYDBERG-ARRAY LITERATURE

The examples in the previous section demonstrate the workflow on individual protocols. The same agentic approach can also be applied at corpus scale: a second agentic workflow surveys the Rydberg-array theory literature and performs a rapid feasibility assessment based on the physical resources required by each proposal. The complete corpus-construction procedure, search queries, filtering pipeline, and classification workflow are described in Appendix B.

The search identified 633 Rydberg-array theory papers. Of these, 107 could not be reliably classified because the corresponding PDF was unreadable or insufficiently detailed for automated analysis. The remaining 526 papers

were classified according to the hardware capabilities required to implement their proposed protocols. Table I summarizes the resulting distribution.

Among the classified papers, 258 (49.0%) were found to be compatible with publicly available Pasqal QPUs. Of these, 225 describe sufficiently complete experimental protocols that can be translated directly into executable implementations, while the remaining 33 require modest adaptation, such as specifying an observable, pulse schedule, or parameter range. The remaining 268 papers require hardware capabilities that are not yet publicly available, most commonly XY-type interactions (141 papers) or local addressing (87 papers).

The 258 implementable papers represent viable starting points for experimentation on present-day hardware. In practice, obtaining a meaningful experimental signal still requires adapting each protocol to the characteristics of the device and selecting observables that remain robust in the presence of hardware noise. The agent substantially reduces this engineering effort while scientific judgment remains essential when designing the experiment and interpreting its outcome.

Beyond estimating the current experimental reach of neutral-atom platforms, the classification also provides a quantitative view of future hardware priorities. The vast majority of currently inaccessible protocols require either XY interactions or local addressing, which together account for approximately 85% of the non-implementable papers. The same analysis therefore serves two complementary purposes: it estimates how much of today’s Rydberg-array literature can already be brought to hardware using an agentic workflow, and it identifies the hardware capabilities that would unlock the largest fraction of the remaining scientific literature.

TABLE I. Classification of the 526 Rydberg-array theory papers the agent could read and classify (out of 633 on-topic; 107 unclassified). The top two rows (258 papers, 49.0%) are implementable on a present-day neutral-atom QPU; the remaining rows group the rest by which rule they fail.

Class	Papers	%
Implementable today	225	42.8
Implementable with modest adaptation	33	6.3
Needs different atomic encoding (XY)	141	26.8
Needs local addressing	87	16.5
Too big ($N > 100$)	17	3.2
Needs measurements during the pulse	9	1.7
Needs basis rotations beyond on/off readout	5	1.0
Genuinely under-specified	9	1.7
Total	526	100

V. SUMMARY AND OUTLOOK

While quantum hardware has become increasingly accessible through cloud platforms and high-performance computing infrastructures, translating a scientific idea into a hardware experiment remains a substantial undertaking. In this work, we introduced an agentic framework designed to bridge this gap, structured as a collection of modular AI-agent skills. By operating on a common experimental specification and state representation, these skills can be iteratively orchestrated by the researcher into refinement loops that regenerate sequences, run simulations, or adjust parameters to accelerate the time to run a real experiment on a Pasqal QPU.

Beyond these individual demonstrations, our corpus-scale analysis provides a broader perspective: approximately half of the analyzed theory papers describe protocols that are compatible with Pasqal QPUs and could immediately be explored experimentally using our workflow. Concurrently, the analysis of the remaining papers quantifies exactly which hardware capabilities—notably XY interactions and local addressing—would unlock the

largest fraction of the literature, providing a demand-driven perspective for future hardware development.

A central observation emerging from our study is that the operational bottleneck has shifted. The agent proves highly effective at coordinating software tools, navigating device interfaces and generating experiments on the QPU. Its limitations appear rather in the selection of physically meaningful observables, recognizing subtle scientific assumptions, and deciding whether an experimental signal genuinely supports a physical claim. If anything, the researcher’s role becomes more decisive than ever—hypothesis formulation, critical evaluation, and scientific interpretation of the data remain irreducibly human—now that the surrounding engineering no longer competes for their attention. The same tooling works in the theorist’s favour upstream as well: by pairing noiseless predictions with noise-aware emulation, it gives a rapid, quantitative read on which observables are likely to rise above the hardware noise, concentrating effort on the signals a QPU can actually resolve.

Looking forward, we believe this work also paves the way for scientific reproducibility. Reproducing a published quantum experiment often requires reconstructing a large amount of implicit engineering knowledge that is only partially captured in the manuscript itself. By translating scientific protocols into structured, executable specifications, agentic workflows offer a scalable framework to preserve this engineering layer alongside the publication. As hardware and software platforms evolve, these specifications can be re-instantiated, adapted to new devices, and revalidated with minimal manual effort. In this sense, publications can evolve from static descriptions of experiments into executable scientific artifacts that can be systematically revisited, verified, and extended over time.

Ultimately, we expect that the most transformative quantum applications will emerge when quantum hardware becomes accessible not only to the teams that build it, but also to a broader scientific community capable of bringing new questions, unique datasets, and fresh perspectives to the field [37].

-
- [1] A. Browaeys and T. Lahaye, *Nature Physics* **16**, 132 (2020).
 - [2] L. Henriot, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, *Quantum* **4**, 327 (2020).
 - [3] A. J. Menssen, M. Gullans, T. Manovitz, J. M. Taylor, J. Cong, J. Sinclair, Z. Aqua, D. J. Blumenthal, J. Ataides, J. Borregaard, *et al.*, arXiv preprint arXiv:2607.21554 (2026).
 - [4] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, *et al.*, *Science* **376**, 1209 (2022).
 - [5] C. Dalyac, L. Leclerc, L. Vignoli, M. Djellabi, W. d. S. Coelho, B. Ximenez, A. Dareau, D. Dreon, V. E. Elfving, A. Signoles, *et al.*, *The European Physical Journal A* **60**, 177 (2024).
 - [6] D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes, *Nature* **624**, 570 (2023).
 - [7] A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, and P. Schwaller, *Nature Machine Intelligence* **6**, 525 (2024).
 - [8] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha, arXiv preprint (2024), [arXiv:2408.06292 \[cs.AI\]](#).
 - [9] T. Zheng *et al.*, arXiv preprint (2025), [arXiv:2505.13259 \[cs.AI\]](#).
 - [10] A. Ghafarollahi and M. J. Buehler, *Proceedings of the National Academy of Sciences* **122**, e2414074122 (2025).
 - [11] A. Ghafarollahi and M. J. Buehler, *Digital Discovery* **3**, 1389 (2024).
 - [12] G. Tom *et al.*, *Chemical Reviews* **124**, 9633 (2024).

- [13] M. Sim *et al.*, *Matter* **7**, 2959 (2024).
- [14] I. Mandal *et al.*, *Nature Communications* **16**, 9104 (2025).
- [15] K. Reuer *et al.*, *Nature Communications* **14**, 7138 (2023).
- [16] V. Sivak *et al.*, arXiv preprint (2025), [arXiv:2511.08493](https://arxiv.org/abs/2511.08493) [quant-ph].
- [17] L. Jern *et al.*, in *IEEE International Conference on Quantum Computing and Engineering (QCE)* (2025).
- [18] H. Asif *et al.*, arXiv preprint (2025), [arXiv:2503.02497](https://arxiv.org/abs/2503.02497) [quant-ph].
- [19] T. Mikuriya *et al.*, arXiv preprint (2025), [arXiv:2510.26101](https://arxiv.org/abs/2510.26101) [cs.CL].
- [20] M. Tao, Y. Li, T. Yue, M. Zhang, and A. Arrieta Marcos, arXiv preprint (2026), [arXiv:2607.00939](https://arxiv.org/abs/2607.00939) [cs.SE].
- [21] M. Shiraishi *et al.*, in *International Conference on Quantum Communications, Networking, and Computing (QCN)* (2026).
- [22] S. Arlt, X. Gu, and M. Krenn, Towards autonomous quantum physics research using LLM agents with access to intelligent tools (2025), [arXiv:2511.11752](https://arxiv.org/abs/2511.11752) [cs.AI].
- [23] S. Kumar, S. Datta, V. Singh, D. Datta, S. K. Singh, and R. Sharma, *IEEE Access* **12**, 75735 (2024).
- [24] X. Tang, Q. Jin, K. Zhu, T. Yuan, Y. Zhang, W. Zhou, M. Qu, Y. Zhao, J. Tang, Z. Zhang, A. Cohan, Z. Lu, and M. Gerstein, *Nature Communications* **16**, 8317 (2025).
- [25] L. Messeri and M. J. Crockett, *Nature* **627**, 49 (2024).
- [26] J. Huang, X. Chen, S. Mishra, H. S. Zheng, A. W. Yu, X. Song, and D. Zhou, in *International Conference on Learning Representations (ICLR)* (2024).
- [27] M. Malinowski, Can AI answer open questions in physics?, <https://m-malinowski.github.io/2026/06/05/ai-open-physics-problems.html> (2026), accessed: June 11, 2026.
- [28] H. Silvério, S. Grijalva, C. Dalyac, L. Leclerc, P. J. Karalekas, N. Shammah, M. Beji, L.-P. Henry, and L. Henriet, *Quantum* **6**, 629 (2022).
- [29] K. Bidzhiev, S. Grava, P. le Henaff, M. Mendizabal Pico, E. Merhej, and A. Quelle, arXiv preprint (2025), [arXiv:2510.09813](https://arxiv.org/abs/2510.09813) [quant-ph].
- [30] Pasqal, pasqal-cloud: Python SDKs to interact with Pasqal Cloud Services, <https://github.com/pasqal-io/pasqal-cloud> (2026), accessed: 2026-06-10.
- [31] A. Wennersteen, K. Bidzhiev, M. D’Arcangelo, M. Moreau, A. Quelle, A. Dauphin, and M. Beji, in *Quantum Engineering Sciences and Technologies for Industry and Services (QUEST-IS 2025)*, Communications in Computer and Information Science, Vol. 2744, edited by F. Barbaresco and F. Gerin (Springer, Cham, 2026) pp. 117–125.
- [32] A. Keesling, A. Omran, H. Levine, H. Bernien, H. Pichler, S. Choi, R. Samajdar, S. Schwartz, P. Silvi, S. Sachdev, P. Zoller, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, *Nature* **568**, 207 (2019).
- [33] S. Guo *et al.*, Order-by-disorder and emergent Kosterlitz–Thouless phase in a triangular Rydberg array (2023), [arXiv:2302.08963](https://arxiv.org/abs/2302.08963) [cond-mat.quant-gas].
- [34] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys, *Nature* **595**, 233 (2021).
- [35] L.-P. Henry and L. Henriet, A method for estimating the chromatic number of a graph, PCT Patent Application No. PCT/EP2024/081690 (2024), assignee: PASQAL; priority EP 23306936.8, filed 8 November 2024.
- [36] C. Vercellino, G. Vitali, P. Viviani, E. Giusto, A. Scionti, A. Scarabosio, O. Terzo, and B. Montrucchio, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 2 (2023) pp. 141–147, [arXiv:2605.03524](https://arxiv.org/abs/2605.03524).
- [37] C. Jurczak, QQuLiPo: What a quantum computer sees when it reads a book (2026), [arXiv:2605.14188](https://arxiv.org/abs/2605.14188) [cs.CL].

ACKNOWLEDGMENTS

The hardware results reported here were obtained on the Pasqal QPUs FC1 and SA1 through standard Pasqal Cloud accounts available to external users. FC1 is located in Sherbrooke, in the DistriQ quantum-innovation zone (Québec, Canada); SA1 is located in the Kingdom of Saudi Arabia. The AI coding agent used throughout this work is Anthropic’s Claude Fable 5 and Opus 4.8.

Appendix A: Agentic workflow description

The framework exposes six modular skills, which share a common per-experiment state and exchange information through structured artifacts.

- **paper_to_spec** translates a scientific objective into a structured experimental specification. The input may be a research publication, patent, technical report, or a high-level user description. The resulting `experiment_spec.json` captures the relevant physical parameters, observables, and experimental constraints.
- **spec_to_sequence** converts the specification into an executable Pulser implementation. In addition to generating the pulse sequence itself, the skill exposes a standard `build_sequence` interface that is reused by the simulation, validation, calibration, and hardware-execution stages, so that modifications to the specification propagate consistently downstream.
- **validate_emu** evaluates a candidate protocol with `emu-mps`. A typical validation pairs a noiseless simulation, which provides the reference prediction, with a noise-aware simulation that incorporates device-specific noise parameters obtained through the Pasqal Cloud SDK and generates a statistical envelope from multiple quantum-jump trajectories. This skill is aimed in particular at Pulser-literate users who may not be aware of the more delicate subtleties of quantum hardware.
- **noise_emulate** performs a high-fidelity noisy emulation of a candidate sequence on local GPU

resources, complementing the faster cloud-based screening of `validate_emu`. The sequence is simulated with matrix-product-state trajectories: a single noiseless reference run plus an ensemble of forty noisy trajectories, whose statistics define a quantile envelope for each observable. The underlying noise model is fetched live from the device through the Pasqal Cloud SDK at execution time, so that the emulation reflects the current operating point of the hardware rather than static datasheet values. Finally, the envelope is widened by a calibration-offset band that propagates residual uncertainty in the Rabi frequency and detuning calibration, producing the acceptance region against which measured QPU data are later evaluated by `harvest_and_analyze`.

- `qpu_submit` manages execution on quantum hardware. Before submission, it can run a short calibration procedure to characterize the current operating point of the device; the resulting calibration values are propagated to both the experimental parameters and the corresponding noise models. The calibrated experiment is then submitted through the Pasqal Cloud SDK.
- `harvest_and_analyze` retrieves completed experimental data and produces higher-level scientific observables. Raw bitstrings are processed using device-reported corrections and compared against both the noiseless prediction and the noise-aware simulation envelope, giving a direct assessment of the agreement between theory, hardware-aware simulation, and measurement.

Appendix B: Details of the search query

We issued nine topical queries against the arXiv abstract index, retrieving 1,966 raw hits, which deduplicated to 1,108 unique papers. A subsequent filtering stage removed off-topic matches (e.g. Fresnel optics, Pulser signal-processing literature, and Rydberg vapour-cell sensor work), resulting in a final corpus of 633 Rydberg-array theory papers.

Each paper was then processed by an automated agent that parses the full text PDF and assigns a protocol class using a five-rule decision tree:

- (i) system size $N \leq 100$; (ii) global drive only (i.e. no site-selective driving or local detuning); (iii) Ising or blockade interactions (excluding XY models, reso-

nant dipole–dipole exchange, dual-species architectures, and microwave-dressed couplings); (iv) absence of mid-circuit measurement; and (v) computational-basis bit-string readout without basis rotations for tomography.

Papers satisfying all five criteria are marked as *implementable*. Otherwise, the first violated rule determines

TABLE II. Verbatim arXiv `search_query` strings used in the corpus scan (executed 2026-05-23) and corresponding hit counts per query. All queries target the abstract field via the `abs:` prefix. The aggregate count includes overlaps across queries; deduplication yields 1,108 unique papers.

Label	search_query	Hits
<code>rydberg_array_dyn</code>	<code>abs:"rydberg atom array" AND (abs:dynamics OR abs:quench OR abs:sweep)</code>	106
<code>rydberg_scars</code>	<code>abs:"rydberg" AND (abs:scar OR abs:"many-body scar" OR abs:"Z2 state" OR abs:"Z3 state")</code>	100
<code>rydberg_mis_opt</code>	<code>abs:"rydberg" AND (abs:"maximum independent set" OR abs:"MIS" OR abs:"combinatorial optimization")</code>	49
<code>rydberg_floquet</code>	<code>abs:"rydberg" AND (abs:"Floquet" OR abs:"periodic driv")</code>	91
<code>rydberg_phase_trans</code>	<code>abs:"rydberg atom array" AND (abs:"phase transition" OR abs:"ordered phase" OR abs:"adiabatic")</code>	68
<code>rydberg_2d_lattices</code>	<code>abs:"rydberg" AND (abs:kagome OR abs:triangular OR abs:"square lattice")</code>	115
<code>pulser_pasqal_sp</code>	<code>abs:"Pulser" OR abs:"Pasqal" OR abs:"FRESNEL"</code>	1,188
<code>rydberg_xy_dipolar</code>	<code>abs:"rydberg" AND (abs:"XY model" OR abs:"dipolar" OR abs:"resonant dipole" OR abs:"flip-flop")</code>	204
<code>rydberg_topology</code>	<code>abs:"rydberg atom array" AND (abs:topological OR abs:"spin liquid" OR abs:"gauge theory")</code>	45
Sum (with overlap)		1,966
Unique		1,108

a hardware feature label indicating the missing capability (see Table I). For each classification, the agent stores rule-level justification with citations to the relevant section or figure in the paper, enabling auditability without requiring full manual review.