

Article

Dynamical Localization Simulated on Actual Quantum Hardware

Andrea Pizzamiglio ¹, Su Yeon Chang ^{2,3} , Maria Bondani ⁴ , Simone Montangero ^{5,6,7} , Dario Gerace ⁸
and Giuliano Benenti ^{1,9,10,*} 

- ¹ Dipartimento di Scienza e Alta Tecnologia, Università degli Studi dell'Insubria, Via Valleggio 11, 22100 Como, Italy; apizzamiglio@studenti.uninsubria.it
- ² Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland; su.chang@epfl.ch
- ³ CERN, 1211 Meyrin, Switzerland
- ⁴ Istituto di Fotonica e Nanotecnologie, Consiglio Nazionale delle Ricerche, Via Valleggio 11, 22100 Como, Italy; maria.bondani@uninsubria.it
- ⁵ Dipartimento di Fisica e Astronomia "G. Galilei", Università di Padova, 35131 Padova, Italy; simone.montangero@unipd.it
- ⁶ Istituto Nazionale di Fisica Nucleare, Sezione di Padova, 35131 Padova, Italy
- ⁷ Padua Quantum Technology Research Center, Università di Padova, 35131 Padova, Italy
- ⁸ Dipartimento di Fisica, Università di Pavia, Via Bassi 6, 27100 Pavia, Italy; dario.gerace@unipv.it
- ⁹ Istituto Nazionale di Fisica Nucleare, Sezione di Milano, Via Celoria 16, 20133 Milano, Italy
- ¹⁰ NEST, Istituto Nanoscienze-CNR, 56126 Pisa, Italy
- * Correspondence: giuliano.benenti@uninsubria.it



Citation: Pizzamiglio, A.; Chang, S.Y.; Bondani, M.; Montangero, S.; Gerace, D.; Benenti, G. Dynamical Localization Simulated on Actual Quantum Hardware. *Entropy* **2021**, *23*, 654. <https://doi.org/10.3390/e23060654>

Academic Editors: Antonino Messina and Agostino Migliore

Received: 31 March 2021
Accepted: 18 May 2021
Published: 23 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Quantum computers are invaluable tools to explore the properties of complex quantum systems. We show that dynamical localization of the quantum sawtooth map, a highly sensitive quantum coherent phenomenon, can be simulated on actual, small-scale quantum processors. Our results demonstrate that quantum computing of dynamical localization may become a convenient tool for evaluating advances in quantum hardware performances.

Keywords: digital quantum simulation; quantum algorithms; complex quantum systems

1. Introduction

The simulation of complex systems is expected to be one of the main applications of future quantum computers [1]. The special role of quantum mechanics in simulation was pointed out long time ago by Feynman [2]. He observed that, while in a classical computer the memory requirements for the simulation of a many-body quantum system can grow exponentially with the number of particles, the growth is only polynomial in a quantum computer, which is itself a many-body quantum system. Lloyd [3] then validated the conjecture of an exponential speedup in the simulation of quantum systems by means of a quantum computer, with respect to a classical computer. Applications have been discussed for many physical models, so far (see, e.g., Refs. [4–13], and references therein). An ideal quantum computer operating with more than fifty qubits could outperform a classical computer, and the quantum advantage for specific problems has been recently claimed [14,15]. However, quantum advantage can only be reached with a high enough quantum gate precision and through processes generating a large enough amount of entanglement, otherwise they can be efficiently simulated via tensor network methods [16]. On the other hand, present-day quantum computers suffer from significant decoherence and the effects of various noise sources, such that the amount of entanglement that can be reached is still limited. Therefore, achieving the quantum advantage in complex, practically relevant problems such as chemical reactions, new materials design, or biological processes, is still an imposing challenge. It is then important to benchmark the progress of

currently available quantum computers by simulating less demanding but still physically significant tasks.

Dynamical localization is one of the most interesting phenomena that characterize the quantum behavior of classically chaotic systems: quantum interference effects suppress the diffusion taking place in the underlying classical model, leading to exponentially localized wave functions. This phenomenon, first discovered in the quantum kicked-rotor model [17,18], has been observed experimentally in the microwave ionization of Rydberg atoms [19], in atom-optic systems [20–22], and in nuclear magnetic resonance quantum ensemble computation [23]. Dynamical localization has deep analogies with Anderson localization of electronic transport in disordered materials [24]. Localization is indeed ubiquitous in wave physics, since it originates from the interference between multiple scattering paths. It has been shown that in a quantum computer simulating dynamical localization the amount of entanglement is deeply connected with the localization length of the system being simulated [25].

In this paper, we simulate dynamical localization with three qubits by means of several IBM quantum processors, with the aim of analyzing the performances of current noisy quantum hardware, freely available for cloud quantum computing. In particular, we use the known quantum algorithm for the quantum sawtooth map [26,27], which is the ideal model to investigate dynamical localization, and more generally quantum chaotic dynamics on a quantum computer. This is because all of the qubits are actually used to simulate dynamics, without the need for auxiliary qubits. Moreover, the quantum algorithm performs forward-backward Fourier transform, thus exploring the entire Hilbert space of the quantum register in a complex multiple-path interferometer that leads to wave-function localization. As such, dynamical localization is a very fragile quantum phenomenon, extremely sensitive to noise. Its effective simulation on an actual quantum hardware is therefore a potentially ideal test bench to illustrate the power of quantum computation.

Here we show that it is possible to detect quantum localization with $n = 3$ qubits and a single step of the sawtooth map, requiring $4n = 12$ single-qubit and $2(n^2 - n) = 12$ two-qubit quantum gates. We find that the height of the localization peak is smaller than the value expected for a noiseless quantum computing machine. Moreover, the peak reduction in real quantum processors is significantly larger than expected from the Qiskit simulator, which only takes into account a limited number of noise channels. Our results show that for the best performing and freely available IBM quantum processors the localization peak does emerge from noise for about 3 map steps.

The paper is organized as follows. In Section 2 we recall the main steps of the quantum algorithm for simulating dynamical localization (see Refs. [26,27] for a detailed description). In Section 3 results obtained from IBM quantum processors are shown and compared with those from the IBM simulator and with the exact, noiseless evolution. Our conclusions are finally drawn in Section 4.

2. Quantum Algorithm for the Dynamical Localization

2.1. Quantum Algorithm for the Sawtooth Map

The most convenient model to simulate dynamical localization on a quantum computer is the quantum sawtooth map. This map describes the dynamics of a periodically driven system, as derived from the Hamiltonian

$$H(\theta, I; \tau) = \frac{I^2}{2} + V(\theta) \sum_{j=-\infty}^{+\infty} \delta(\tau - jT), \quad (1)$$

where (I, θ) are conjugate action-angle variables ($0 \leq \theta < 2\pi$), with the usual quantization rules, $\theta \rightarrow \theta$ and $I \rightarrow I = -i\partial/\partial\theta$ (we set $\hbar = 1$) and $V(\theta) = -k(\theta - \pi)^2/2$. This Hamiltonian is the sum of two terms, $H(\theta, I; \tau) = H_0(I) + H_k(\theta; \tau)$, where $H_0(I) = I^2/2$ is the Hamiltonian of a particle freely moving on a circle parameterized by the coordinate θ , while $H_k(\theta; \tau) = V(\theta) \sum_j \delta(\tau - jT)$ represents a force acting on the particle and switched

on and off instantaneously (kicking potential) at time intervals T . The evolution from time tT^- (prior to the t -th kick) to time $(t + 1)T^-$ (prior to the $(t + 1)$ -th kick) is described by a unitary operator U acting on the wave function ψ :

$$\begin{aligned} \psi_{t+1} &= U \psi_t = U_T U_k \psi_t ; \\ U_T &= e^{-iT^2/2}, \quad U_k = e^{ik(\theta-\pi)^2/2}. \end{aligned} \tag{2}$$

This map is known as quantum sawtooth map, since the force $F(\theta) = -dV(\theta)/d\theta = k(\theta - \pi)$ has a sawtooth shape, with a discontinuity at $\theta = 0$.

The map (2) can be efficiently simulated on a quantum computer. The quantum algorithm simulating the sawtooth map dynamics is based on the forward/backward quantum Fourier transform between action (momentum) and angle bases. Such an approach is convenient since the operator U , introduced in Equation (2), is the product of two operators, U_k and U_T , diagonal in the θ and I representations, respectively. This quantum algorithm requires the following steps for one map iteration:

- We apply U_k to the wave function $\psi(\theta)$. In order to decompose the operator U_k into one- and two-qubit gates, first of all we write θ in binary notation:

$$\theta = 2\pi \sum_{j=1}^n \alpha_j 2^{-j}, \tag{3}$$

with $\alpha_j \in \{0, 1\}$. As n is the number of qubits, the total number of levels in the quantum sawtooth map is $N = 2^n$. Using expansion (3), we can decompose U_k into a product of n^2 two-qubit operations, each acting non-trivially only on the qubits j_1 and j_2 . In the computational basis $\{|\alpha_{j_1} \alpha_{j_2}\rangle\}$ each two-qubit gate can be written as $\exp(i2\pi^2 k D_{j_1, j_2})$, where D_{j_1, j_2} is a diagonal matrix:

$$D_{j_1, j_2} = \begin{bmatrix} \frac{1}{4n^2} & 0 & 0 & 0 \\ 0 & -\frac{1}{2n} \left(\frac{1}{2^{j_1}} - \frac{1}{2n} \right) & 0 & 0 \\ 0 & 0 & -\frac{1}{2n} \left(\frac{1}{2^{j_2}} - \frac{1}{2n} \right) & 0 \\ 0 & 0 & 0 & \left(\frac{1}{2^{j_1}} - \frac{1}{2n} \right) \left(\frac{1}{2^{j_2}} - \frac{1}{2n} \right) \end{bmatrix}. \tag{4}$$

Neglecting a global phase factor of no physical significance, we can decompose $\exp(i2\pi^2 k D_{j_1, j_2})$ in terms of quantum gates (P and CP) available in the IBM Qiskit interface:

$$e^{i2\pi^2 k D_{j_1, j_2}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{-\frac{i\pi^2 k}{2^{j_1 n}}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-\frac{i\pi^2 k}{2^{j_1 n}}} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-\frac{i\pi^2 k}{2^{j_2 n}}} & 0 \\ 0 & 0 & 0 & e^{-\frac{i\pi^2 k}{2^{j_2 n}}} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\frac{i\pi^2 k}{2^{j_1 + j_2 - 1}}} \end{pmatrix}. \tag{5}$$

The first two factors in the product correspond to phase-shift gates (P gates) applied to qubit j_1 and j_2 , respectively, while the last one corresponds to a controlled phase-shift gate (CP gates) applied on both qubits, if $j_1 \neq j_2$, and to a P gate if $j_1 = j_2$. Overall we need then $n^2 - n$ CP e $2n^2 + n$ P gates to implement U_k . As U_k is the product of diagonal matrices in the θ representation, the order of P and CP gates is irrelevant. Therefore, changing their order allows combining gates applied to the same qubits, thus improving the efficiency of the quantum algorithm. We then reduce the number of quantum gates needed to implement U_k to $n(n - 1)/2$ CP and n P gates. We note that the above decomposition for U_k is possible thanks to the particular form of $V(\theta)$ for the sawtooth map. This is the reason for which the quantum sawtooth map is the most convenient model to simulate dynamical localization. For other potentials, like

$V(\theta) = \cos \theta$ corresponding to the kicked-rotor model [17,18], auxiliary qubits are introduced to efficiently compute U_k [9], thus increasing the overall simulation cost.

- The change from the θ to the I representation is obtained by means of the quantum Fourier transform (QFT), which requires n Hadamard (H) gates and $\frac{1}{2}n(n - 1)$ CP gates. The correct order of qubits after the Fourier transform is obtained by simply relabeling qubits, thus avoiding $\lfloor n/2 \rfloor$ physical SWAP gates.
- In the I representation, the operator U_T has essentially the same form as the operator U_k in the θ representation and can therefore be similarly decomposed.
- We return to the initial θ representation by application of the inverse QFT.

Overall, this quantum algorithm requires $O(n^2)$ gates per map iteration. This number is to be compared with the $O(n2^n)$ operations required by a classical computer to simulate one map iteration using the fast Fourier transform. The quantum simulation of the quantum sawtooth map dynamics is then exponentially faster than any known classical algorithm. Note that the resources required to the quantum computer to simulate the sawtooth map are only logarithmic in the system size N . With the above discussed improvements of the quantum circuit, we need $2(n^2 - n)$ CP, $2n$ P, and $2n$ Hadamard (H) gates to simulate one step of the quantum sawtooth map. Which corresponds to 24 quantum gates (12 single-qubit a 12 two-qubit gates) for the case $n = 3$ which we have simulated on the IBM quantum hardware. The angles used in the P gates are obtained by changing order of the gates and combining them. The explicit formulas for the angles are given as the following:

$$\theta_k^j = -\frac{2\pi^2 k}{2^j} + \frac{\pi^2 k}{2^{2j-1}}, \quad \theta_T^j = \frac{2N^2 T}{2^{j+2}} - \frac{N^2 T}{2^{2j+1}}, \quad \phi_k^{j_1, j_2} = \frac{2\pi^2 k}{2^{j_1+j_2-1}}, \quad \phi_T^{j_1, j_2} = -\frac{2N^2 T}{2^{j_1+j_2+1}}. \quad (6)$$

The circuit (see Figure 1) avoids SWAP gates via relabeling of qubits after the QFT and the inverse QFT.

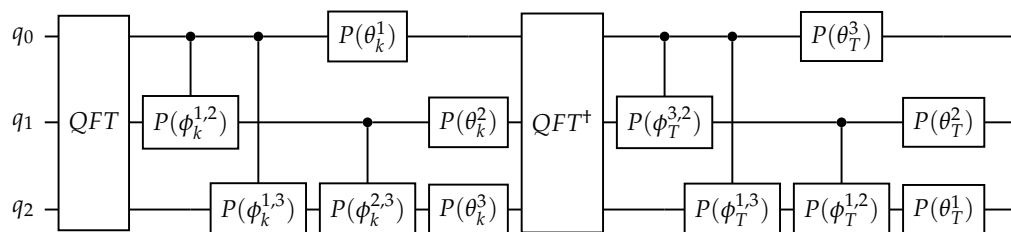


Figure 1. Quantum circuit of sawtooth map with $n = 3$.

2.2. Dynamical Localization for the Sawtooth Map

The classical limit of the quantum sawtooth map is obtained for $k \rightarrow \infty$ and $T \rightarrow 0$, at constant $K = kT$. The classical motion is chaotic for $K < -4$ and $K > 0$. Although the sawtooth map is a deterministic system, in this regime the motion along the momentum direction is in practice indistinguishable from a random walk, with diffusion in the momentum variable. If we consider a classical ensemble of trajectories with fixed initial momentum m_0 and random initial angle θ , the second moment of the distribution grows linearly with the number t of map iterations,

$$\langle (\Delta m)^2 \rangle \approx Dt, \quad (7)$$

with the diffusion coefficient $D(k) \approx (\pi^2/3)k^2$ for $K > 1$.

The quantum sawtooth map, in agreement with the correspondence principle, initially exhibits diffusive behavior, with the classical diffusion coefficient D . However, after a break time t^* , quantum interference leads to suppression of diffusion. For $t > t^*$ the quantum distribution reaches a steady state which decays exponentially over the momentum eigenbasis:

$$W_m \equiv |\langle m | \psi \rangle|^2 \approx \frac{1}{\ell} \exp\left[-\frac{2|m - m_0|}{\ell}\right], \quad (8)$$

where the index m singles out the momentum eigenstates ($I|m\rangle = m|m\rangle$) and the system is initially prepared in the eigenstate $|m_0\rangle$. Therefore, for $t > t^*$ only

$$\sqrt{\langle(\Delta m)^2\rangle} \approx \sqrt{Dt^*} \approx \ell \quad (9)$$

levels are populated.

An estimate of t^* and ℓ can be obtained by means of a heuristic argument [28]. Since the number of levels involved grows diffusively, $\propto \sqrt{t}$, and, due to the Heisenberg principle, the discreteness of levels is resolved down to an energy spacing $\propto 1/t$, then the discreteness of spectrum eventually dominates. The localized wavefunction has significant projection over about ℓ eigenstates of the Floquet operator U , which determines the evolution of the system over one map step. This operator is unitary and therefore its eigenvalues can be written as $\exp(i\lambda_i)$, with λ_i (known as quasienergies) distributed in the interval $[0, 2\pi]$. The mean level spacing between quasienergy eigenstates which significantly determine the dynamics is $\Delta E \approx 2\pi/\ell$. The Heisenberg principle tells us that the minimum time required to the dynamics to resolve this energy spacing is given by

$$t^* \approx 1/\Delta E \approx \ell. \quad (10)$$

Relations (9) and (10) imply

$$t^* \approx \ell \approx D. \quad (11)$$

The quantum localization can be detected if ℓ is smaller than the system size N .

3. Quantum Computing of Dynamical Localization

We simulate dynamical localization with $n = 3$ qubits on a real quantum hardware. The initial condition is peaked in momentum, $\psi_0(m) = \delta_{m,m_0}$, with $m_0 = 0$. The quantum algorithm for the sawtooth map allows us to compute the wave vector $\psi_t(m)$ as a function of the number of map steps, and then the momentum probability distribution $W_t(m) = |\psi_t(m)|^2$. We consider $k \approx 0.273 < 1$, so that the distribution is already localized after a single map step. On the other hand, we use $K = 1.5$, corresponding to diffusive behavior for the underlying classical dynamics. In Figure 2 we compare after $t = 1$ map step the ideal, noiseless distribution (green curve) with the Qiskit simulator (blue curve), used to test the quantum algorithm before submitting it to a real quantum hardware (red curve). Data for the quantum hardware are for the recently released (8th January 2021) *lima* quantum processor. This machine has a nominal quantum volume $V_Q = 8$ (i.e., a single number meant to encapsulate the quantum computer performance [29]), which is equal or smaller than the quantum volume of other freely available quantum processors. Nevertheless, in our simulation we obtained the best results for the localization problem with this machine (see discussion below).

As we can see in Figure 2, the quantum hardware exhibits a localization peak after a single map step. On the other hand, the height of the peak, $W_1(0) \approx 0.56$, is significantly smaller than the noiseless value, $W_1(0) \approx 0.83$, and the prediction of the Qiskit simulator, $W_1(0) \approx 0.74$. These results show that the Qiskit simulator may at least underestimate some of the relevant noise channels. It is also interesting to remark that, while the ideal distribution is symmetric around its peak, the noisy ones are markedly asymmetric. We interpret this asymmetry as a consequence of the binary coding of momentum eigenstates. For instance, a noise channel leading to a flipping of the most significant qubit would induce a probability transfer from the peak ($m = 0$, corresponding to the state $|100\rangle$ of the three qubits) to the state $|000\rangle$ (corresponding to $m = -4$).

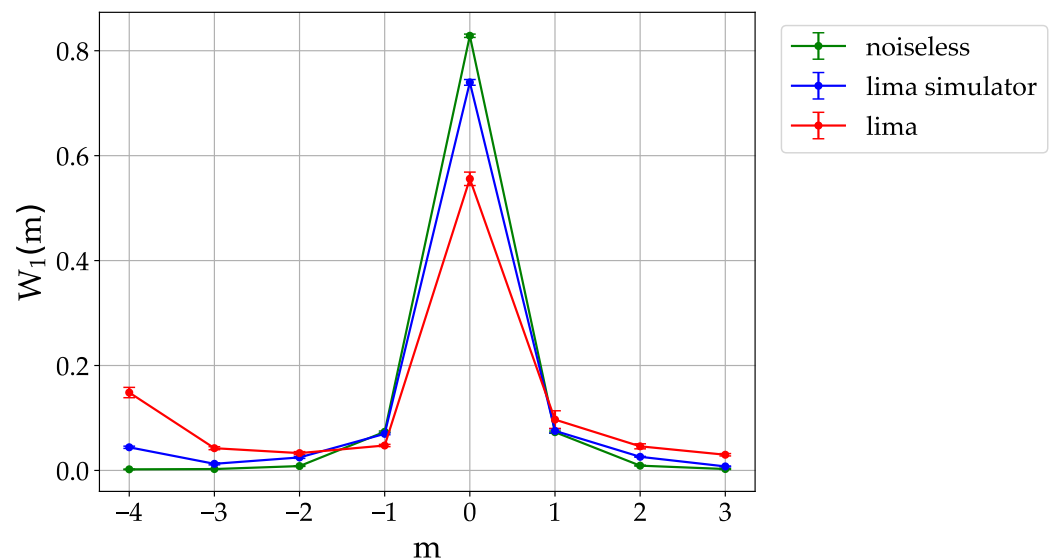


Figure 2. Dynamical localization in the quantum sawtooth map with $n = 3$ qubits, $K = 1.5$, $k \approx 0.273$ ($T = 2\pi L/N$, with $L = 7$). Hereafter data from the quantum processors (here *lima*) are obtained after averaging over 10 repetitions of 8192 experimental runs. Data taken on 26th February 2021.

To compare the performance of different freely available quantum machines, we ran the sawtooth quantum algorithm for one map step for several days (from 24th February to 5th March 2021). In Figure 3, we show data for *lima*, *belem*, and *quito* (release date 8th January 2021, nominal quantum volume $V_Q = 16$), *santiago* (3rd June 2020, $V_Q = 32$), *athens* (3rd March 2020, $V_Q = 32$), and *ibmqx2-yorktown* (24th January 2017, $V_Q = 8$). This latter is an old machine having a low quantum volume (nominally), but its bow-tie layout is in principle more favorable for the present application, as it allows direct connectivity between all the three qubits involved in the algorithm. Actually the unfavorable layout of the other machines is not very relevant for our 3-qubit simulations. Indeed, the number of extra quantum gates needed to relate non-physically connected qubits, in order to make the quantum algorithm work on these devices, is such that the execution time of the algorithm never exceeds the minimum decoherence time of the qubits used for all the machines. For such qubits, the algorithm run time for yorktown is $\approx 11.1 \mu\text{s}$, while for the other devices requiring extra gates it increases only up to $\approx 16.7 \mu\text{s}$ on average. By comparing the height of the localization peak for several days of runs on the available quantum processors, we conclude that *lima* is to be considered the best performing machine at time of writing, both in terms of the best achieved absolute performance ($W_1(0) \approx 0.56$) and stability of the results obtained.

In order to quantitatively relate the effectiveness of the dynamical localization quantum algorithm to the error rate, we compare the height of the localization peak to the noise level in each of the analyzed devices, as measured from the day-to-day calibrations (nominally given on the IBM Q website). As a measure of the quality of the qubits, we define an average decoherence time $\langle T_{\text{dec}} \rangle$, obtained after averaging over the $n = 3$ qubits used in a given simulation, $T_{\text{dec}} = \min\{T_1, T_2\}$, where T_1 and T_2 are the standard relaxation and dephasing times, respectively. Then, we also estimate the overall relative error of a quantum computer run, defined as \mathcal{E}_{tot} , as the sum of the relative errors of each quantum gate and of the final readout error. Results are then reported in Figure 4, where we use a color code from red to violet to quantify $\langle T_{\text{dec}} \rangle$, while \mathcal{E}_{tot} is proportional to the size of the symbols. It can be seen that the height of the localization peak is larger for larger $\langle T_{\text{dec}} \rangle$ and smaller \mathcal{E}_{tot} , i.e., corresponding to small violet circles (in particular, see data for the *lima* processor). On the other hand, the worst performances are obtained for smaller $\langle T_{\text{dec}} \rangle$ and larger \mathcal{E}_{tot} , i.e., corresponding to large red circles (in particular, see data for the *yorktown* processor). However, large fluctuations around this average behavior can be noticed. This is expected, since the data from the Qiskit simulator that are obtained by as-

suming the same nominal parameters used to define $\langle T_{\text{dec}} \rangle$ and \mathcal{E}_{tot} , largely underestimate noise, as it is shown in Figure 2. In particular, fluctuations of the qubit quality parameters between different calibration procedures, memory effects, and cross-talks between qubits (even not specifically employed in a given simulation) could also affect the performance of a quantum processor, and may not be fully captured by the currently implemented noise models.

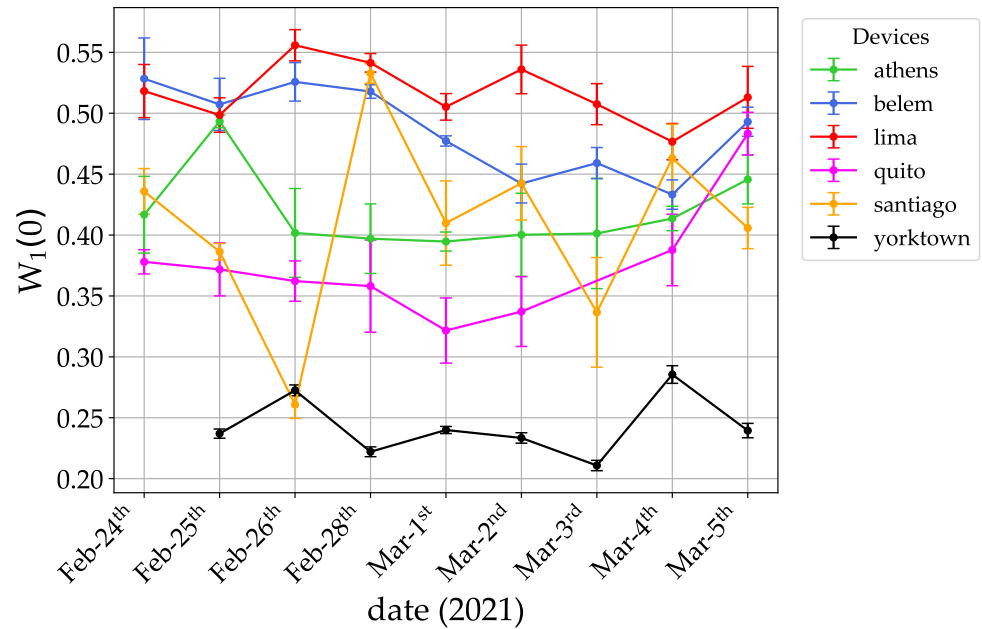


Figure 3. Height of the localization peak after one map step for several quantum processors. Data taken from 24th February to 5th March 2021.

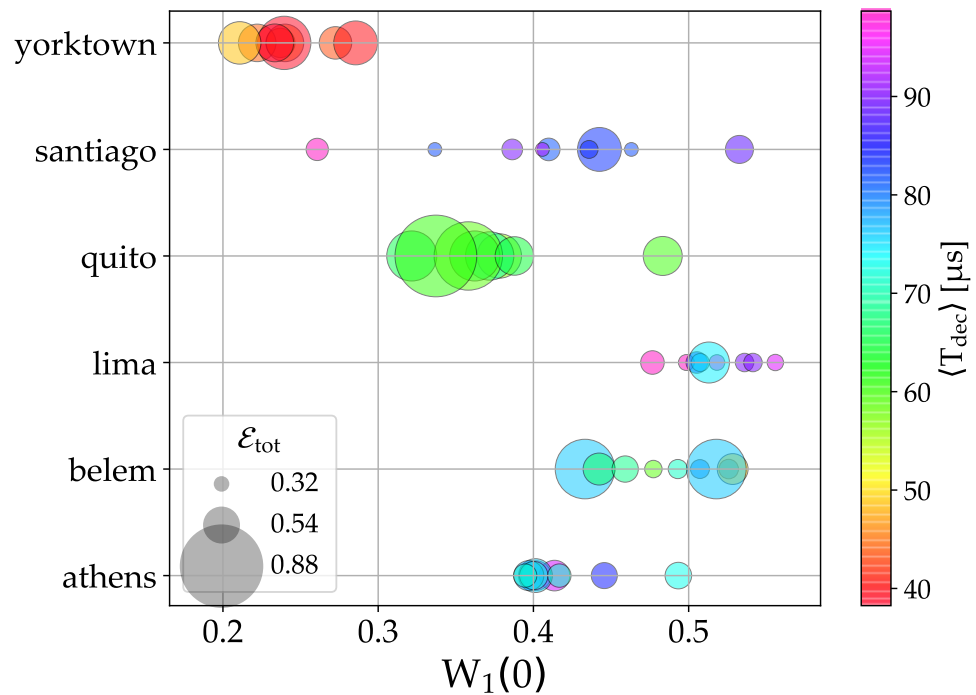


Figure 4. Height of the localization peak for the six quantum processors used in this work. For all the data from Figure 3, we show the average decoherence time, $\langle T_{\text{dec}} \rangle$ (color coded from red to violet), and the overall relative error, \mathcal{E}_{tot} (size of the circles), as defined in the text.

Finally, with the aim of further testing the capabilities of quantum processors to simulate dynamical localization, we show in Figure 5 the localization peak value as a function of the number of kicks. For the noiseless case the distribution remains localized with fluctuations of $W_t(0)$ around 0.9. For the *ibmqx2 - yorktown* quantum processor the peak at $m = 0$ does not emerge from noise already at $t = 2$. On the other hand, for the *lima* quantum processor the peak is visible up to $t = 3$ kicks. Evidently, the Qiskit simulator underestimates the level of noise in the corresponding devices, and predicts a peak decay in 3–5 map steps (for *ibmqx2-yorktown*) and in 10–15 kicks for *lima*. Overall, the data of Figures 3 and 5 testify the significant improvement of quantum hardware performance in the last few years. As a closing remark, it is worth reminding that error mitigation techniques can be applied to reduce the impact of various sources of noise on near term quantum processors [30–32]. We have applied an error mitigation procedure based on Ref. [31] in combination with the Qiskit noise model, and only found a limited effect in improving the specific quantum localization algorithm (results not shown). A more detailed characterization of noise in such devices is by itself an interesting problem, which goes, however, beyond the scope of the present work.

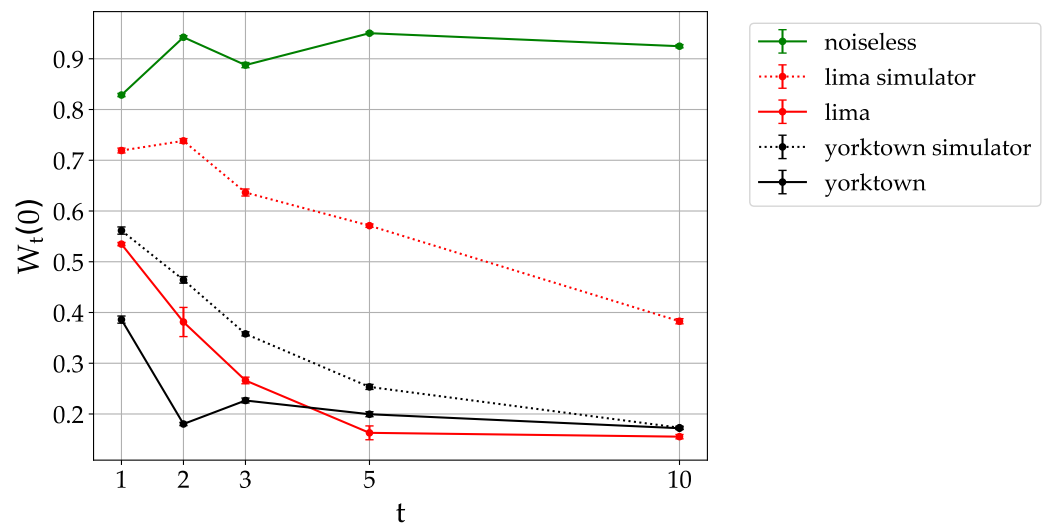


Figure 5. Height of the localization peak as a function of the number of map steps, for parameter values as in Figure 2. Data are for *lima* (red) and *ibmqx2-yorktown* (black). Data taken on 14th February (*ibmqx2-yorktown*) and 5th March 2021 (*lima*).

4. Conclusions

We have simulated the dynamical localization phenomenon of the quantum sawtooth map by means of several freely available IBM quantum processors, remotely accessed through cloud quantum programming. The localization peak emerges from noise, even though significantly reduced with respect to the noiseless dynamics, with up to three steps of the sawtooth map model for three qubits. Our results demonstrate a significant improvement of the performances of the available quantum processors over the past few years. This study shows that quantum computing of dynamical localization may be employed as a fast and convenient benchmark to check the performance of future generations of quantum processors. In particular, we could envision studying the maximum number n of qubits for which the peak height after one kick is reproduced with a given accuracy. Since the number of elementary quantum gates required to simulate a single step of the map scales as n^2 , in this “localization test” both the number of qubits and the circuit depth grow with n . The results of this work show that, for $n = 3$, the localization test does not necessarily agree with the quantum volume measure.

Author Contributions: A.P. and S.Y.C. performed quantum simulations by coding actual IBM quantum processors. G.B. supervised the work, with inputs from M.B., S.M. and D.G., and drafted the manuscript. All authors discussed the results and contributed to revision of the manuscript. All authors have read and agreed to the published version of the manuscript

Funding: S.M. acknowledges support by the Italian PRIN 2017, the Fondazione CARIPARO, and the Horizon 2020 research and innovation programme under grant agreement No 817482 (Quantum Flagship - PASQuanS). D.G. acknowledges the Italian Ministry of Education, University and Research (MIUR): “Dipartimenti di Eccellenza Program (2018–2022)”, Department of Physics, University of Pavia. G.B. acknowledges the financial support of the INFN through the project QUANTUM.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used and analyzed in the current study are available from the corresponding author on reasonable request.

Acknowledgments: We acknowledge use of the IBM Quantum Experience for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM company or the IBM-Q team. F. Tacchino is gratefully acknowledged for co-supervising the work of S.-Y. Chang during her visit at the University of Pavia in February 2019.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Benenti, G.; Casati, G.; Rossini, D.; Strini, G. *Principles of Quantum Computation and Information (A Comprehensive Textbook)*; World Scientific: Singapore, 2019.
2. Feynman, R.P. Simulating Physics with Computers. *Int. J. Theor. Phys.* **1982**, *21*, 467. [[CrossRef](#)]
3. Lloyd, S. Universal quantum simulators. *Science* **1996**, *273*, 1073. [[CrossRef](#)] [[PubMed](#)]
4. Abrams, D.; Lloyd, S. Simulation of many-body Fermi systems on a universal quantum computer. *Phys. Rev. Lett.* **1997**, *79*, 2586. [[CrossRef](#)]
5. Zalka, C. Efficient simulation of quantum systems by quantum computers. *Fortschritte Phys.* **1998**, *46*, 877. [[CrossRef](#)]
6. Schack, R. Using a quantum computer to investigate quantum chaos. *Phys. Rev. A* **1998**, *57*, 1634 [[CrossRef](#)]
7. Terhal, B.M.; DiVincenzo, D.P. Problem of equilibration and the computation of correlation functions on a quantum computer. *Phys. Rev. A* **2000**, *61*, 022301. [[CrossRef](#)]
8. Ortiz, G.; Gubernatis, J.E.; Knill, E.; Laflamme, R. Quantum algorithms for fermionic simulations. *Phys. Rev. A* **2001**, *64*, 022319. [[CrossRef](#)]
9. Georgeot, B.; Shepelyansky, D.L. Exponential gain in quantum computing of quantum chaos and localization. *Phys. Rev. Lett.* **2001**, *86*, 2890. [[CrossRef](#)]
10. Benenti, G.; Strini, G. Quantum simulation of the single-particle Schrodinger equation. *Am. J. Phys.* **2008**, *76*, 657. [[CrossRef](#)]
11. Georgescu, I.M.; Ashhab, S.; Nori, F. Quantum simulation. *Rev. Mod. Phys.* **2014**, *86*, 153. [[CrossRef](#)]
12. Chiesa, A.; Tacchino, F.; Grossi, M.; Santini, P.; Tavernelli, I.; Gerace, D.; Carretta, S. Quantum hardware simulating four-dimensional inelastic neutron scattering. *Nat. Phys.* **2019**, *15*, 455. [[CrossRef](#)]
13. Tacchino, F.; Chiesa, A.; Carretta, S.; Gerace, D. Quantum computers as universal quantum simulators: State-of-the-art and perspectives. *Adv. Quantum Technol.* **2020**, *3*, 190052. [[CrossRef](#)]
14. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505. [[CrossRef](#)] [[PubMed](#)]
15. Zhong, H.S.; Wang, H.; Deng, Y.H.; Chen, M.C.; Peng, L.C.; Luo, Y.H.; Qin, J.; Wu, D.; Ding, X.; Hu, Y.; et al. Quantum computational advantage using photons. *Science* **2020**, *370*, 1460. [[PubMed](#)]
16. Zhou, Y.; Stoudenmire, E.M.; Waintal, X. What Limits the Simulation of Quantum Computers? *Phys. Rev. X* **2020**, *10*, 041038.
17. Casati, G.; Chirikov, B.V.; Ford, J.; Izrailev, F.M. Stochastic behavior of a quantum pendulum under a periodic perturbation. *Lect. Notes Phys.* **1979**, *93*, 334.
18. Izrailev, F.M. Simple models of quantum chaos: spectrum and eigenfunctions. *Phys. Rep.* **1990**, *196*, 299. [[CrossRef](#)]
19. Koch, P.M.; van Leeuwen, K.A.H. The importance of resonances in microwave “ionization” of excited hydrogen atoms. *Phys. Rep.* **1995**, *255*, 289. [[CrossRef](#)]
20. Moore, F.L.; Robinson, J.C.; Barucha, C.F.; Sundaram, B.; Raizen, M.G. Atom optics realization of the quantum delta-kicked rotor. *Phys. Rev. Lett.* **1995**, *75*, 4598. [[CrossRef](#)] [[PubMed](#)]
21. Chabé, J.; Lemarié, G.; Grémaud, B.; Delande, D.; Szriftgiser, P.; Garreau, J.C. Experimental Observation of the Anderson Metal-Insulator Transition with Atomic Matter Waves. *Phys. Rev. Lett.* **2008**, *101*, 255702. [[CrossRef](#)]

22. Manai, I.; Clément, J.-F.; Chicireanu, R.; Hainaut, C.; Garreau, J.C.; Szriftgiser, P.; Delande, D. Experimental Observation of Two-Dimensional Anderson Localization with the Atomic Kicked Rotor. *Phys. Rev. Lett.* **2015**, *115*, 240603. [[CrossRef](#)] [[PubMed](#)]
23. Henry, M.K.; Emerson, J.; Martinez, R.; Cory, D.G. Study of localization in the quantum sawtooth map emulated on a quantum information processor. *Phys. Rev. A* **2006**, *74*, 062317. [[CrossRef](#)]
24. Fishman, S.; Grepel, D.R.; Prange, R.E. Chaos, Quantum Recurrences, and Anderson Localization. *Phys. Rev. Lett.* **1982**, *49*, 509. [[CrossRef](#)]
25. Montangero, S. Dynamically localized systems: Exponential sensitivity of entanglement and efficient quantum simulations. *Phys. Rev. A* **2004**, *70*, 032311. [[CrossRef](#)]
26. Benenti, G.; Casati, G.; Montangero, S.; Shepelyansky, D.L. Efficient quantum computing of complex dynamics. *Phys. Rev. Lett.* **2001**, *87*, 227901. [[CrossRef](#)]
27. Benenti, G.; Casati, G.; Montangero, S.; Shepelyansky, D.L. Dynamical localization simulated on a few-qubit quantum computer. *Phys. Rev. A* **2003**, *67*, 052312. [[CrossRef](#)]
28. Chirikov, B.V.; Izrailev, F.M.; Shepelyansky, D.L. Dynamical stochasticity in classical and quantum mechanics. *Sov. Sci. Rev. C* **1981**, *2*, 209.
29. Cross, A.W.; Bishop, L.S.; Sheldon, S.; Nation, P.D.; Gambetta, J.M. Validating quantum computers using randomized model circuits. *Phys. Rev. A* **2019**, *100*, 032328. [[CrossRef](#)]
30. Temme, K.; Bravyi, S.; Gambetta, J.M. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.* **2017**, *119*, 180509. [[CrossRef](#)]
31. Kandala, A.; Temme, K.; Córcoles, A.D.; Mezzacapo, A.; Chow, J.M.; Gambetta, J.M. Error mitigation extends the computational reach of a noisy quantum processor. *Nature* **2019**, *567*, 491. [[CrossRef](#)] [[PubMed](#)]
32. Suchs, P.; Tacchino, F.; Fischer, M.H.; Neupert, T.; Barkoutsos, P.K.; Tavernelli, I. Algorithmic Error Mitigation Scheme for Current Quantum Processors. *arXiv* **2021**, arXiv:2008.10914v2.