# Machine learning in lattice quantum gravity

J. Ambjorn, <sup>1, 2</sup> Z. Drogosz, <sup>3</sup> J. Gizbert-Studnicki, <sup>3, 4, \*</sup> A. Görlich, <sup>3, 4</sup> D. Németh, <sup>5</sup> and M. Reitz<sup>6</sup>

<sup>1</sup>The Niels Bohr Institute, Copenhagen University, Blegdamsvej 17, DK-2100 Copenhagen Ø, Denmark.

<sup>2</sup>IMAPP, Radboud University, Houtlaan 4, 6525 XZ Nijmegen, The Netherlands

<sup>3</sup>Institute of Theoretical Physics, Jagiellonian University, Lojasiewicza 11, Kraków, PL 30-348, Poland.

<sup>4</sup>Mark Kac Center for Complex Systems Research,

Jagiellonian University, Lojasiewicza 11, Kraków, PL 30-348, Poland.

<sup>5</sup> IMAPP, Radboud University, Houtlaan 4 6525 XZ Nijmegen, The Netherlands

<sup>6</sup> TNO – Netherlands Organization for Applied Scientific Research,

High Tech Campus 21, 5656 AE Eindhoven, The Netherlands.
(Dated: October 3, 2025)

Using numerical data coming from Monte Carlo simulations of four-dimensional Causal Dynamical Triangulations, we study how automated machine learning algorithms can be used to recognize transitions between different phases of quantum geometries observed in lattice quantum gravity. We tested seven supervised and seven unsupervised machine learning models and found that most of them were very successful in that task, even outperforming standard methods based on order parameters.

#### I. INTRODUCTION

# Phase structure recognition and phase transition analysis constitute an important problem in many lattice quantum field theories. In particular in the context of quantum gravity, the prospective continuum limit, ideally one consistent with the putative UV fixed point of quantum gravity postulated by asymptotic safety conjecture [1] and sought in the functional renormalization group approaches [2–4], should be related to a phase transition point of lattice theory [5–7]. At the same time, one aims to reproduce the correct infrared limit, consistent with a small quantum perturbation of general relativity (GR). Therefore, recognizing different phases of quantum geometry and analyzing phase transitions using Monte Carlo (MC) data remains a vital task. For example, in the Causal Dynamical Triangulations (CDT) approach, one observes a rich phase structure, with four different phases of quantum geometry, of which only three were initially recognized [8, 9], and it took more than a decade before the fourth one (the so-called phase $C_b$ or the "bifurcation phase") was discovered [10]. This was due to the limited set of order parameters used at that time, which were insensitive to the bifurcation phase transition. It is thus tempting to ask whether machine learning (ML) techniques can be used to give some insight into the nature of the observed phase transitions and (prospectively) to automatically explore the CDT parameter space in search of potential new phases. In this work, we made the first step in this direction by analyzing phase transitions already observed in CDT, using a number of ML methods that included both supervised and unsupervised techniques.

### \* jakub.gizbert-studnicki@uj.edu.pl

## II. CAUSAL DYNAMICAL TRIANGULATIONS

Causal Dynamical Triangulations (CDT) is an approach to quantizing gravity using a nonperturbative lattice quantum field theory framework; see [11] for its detailed formulation and [12–14] for reviews. CDT regularizes the (formal) quantum-gravitational path integral over geometries by a sum over a countable set of triangulations constructed in four dimensions from two types of elementary simplicial building blocks with fixed edge lengths. An important assumption of CDT is the restriction of the path integral to globally hyperbolic geometries (triangulations), which can be foliated into spatial hypersurfaces of identical fixed topology. The triangulations are combinatorial, i.e., each simplex is uniquely specified by its set of vertices, and simplices are glued along entire faces only. This ensures that the resulting complex is a genuine simplicial manifold of fixed global topology, without self-identifications or conical singularities beyond the standard Regge-type curvature defects.

In each triangulation, a spatial slice with integer lattice time coordinate t is constructed by gluing together equilateral tetrahedra so that the chosen spatial topology is preserved. The neighboring spatial slices at t and t+1 are connected by timelike edges that together with the tetrahedra form four-dimensional simplices with either four vertices on one time-slice and one vertex on the other one – the (4,1)-simplex – or three vertices on one time-slice and two on the other one – the (3,2)-simplex. The four-simplices are internally flat. Curvature is defined by deficit angles around two-dimensional "bones" (triangles), so by gluing simplices together along their three-dimensional faces nontrivial geometries emerge.

The triangulations are summed over with a weight dependent on the Einstein-Hilbert action, which for a piecewise flat triangulation becomes the Regge action [15]. In CDT, the Regge action takes the simple form of a linear combination of certain global numbers characteriz-

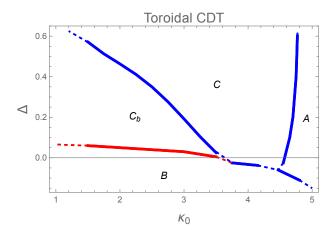


FIG. 1. The phase diagram of the toroidal CDT. Solid lines denote measured phase transition lines, where first-order transitions are shown in blue while higher-order transitions in red. Dashed lines are extrapolations.

ing a triangulation, i.e., the total number of vertices, denoted  $N_0$ , and the total numbers of the two types of foursimplices, denoted  $N_{41}$  and  $N_{32}$ , are weighted by three dimensionless coupling constants,  $\kappa_0$ ,  $\kappa_4$  and  $\Delta$ , related to the gravitational constant, the cosmological constant and the asymmetry of lengths of spacelike and timelike links in the lattice, respectively. The CDT time foliation enables a well-defined Wick rotation from Lorentzian to Euclidean metric signature, after which CDT becomes a statistical theory of triangulated random geometries, which can be investigated using MC techniques. In a MC simulation, we let the total number of (4,1)-simplices fluctuate around a fixed target volume  $\bar{N}_{41}$  (the average number of (4,1)-simplices) by fine-tuning the lattice cosmological constant  $\kappa_4 \to \kappa_4^c(\bar{N}_{41})$ . This results in a two-dimensional parameter space:  $(\kappa_0, \Delta)$ .

For two choices of the fixed spatial topology, that of a three-sphere and a three-torus, the parameter space has been thoroughly scanned in search of distinct phases of dynamically emerging quantum geometry. Four phases (named A, B, C and  $C_b$ ) were found, and they are separated by both first- [16–19] and higher-order [16, 20–22] phase transition lines; see Fig. 1. In the following, we will focus on the toroidal CDT case, and we will analyze three of the phase transitions, namely the A-B, A-C and  $B-C_b$ , which, using standard order parameter approach, were classified as a first-order [19], a "weak" first-order [18], and a higher-order transition [22], respectively.

#### III. NUMERICAL SETUP

As the first step, Monte Carlo simulations were performed at various points of the CDT phase diagram to generate data characterizing generic quantum spacetime geometries appearing in each of the four phases (compare Fig. 1). All the simulations were performed for the fixed

spatial toroidal topology and the fixed number of spatial slices  $N_t = 4$  (using time-periodic boundary conditions). In order to test possible volume dependence, all measurements were repeated for a set of different lattice volumes ranging from  $\bar{N}_{41} = 20\,000$  to  $\bar{N}_{41} = 600\,000$ . The measurement data obtained were subsequently used as input features for all tested machine learning algorithms. Since numerical CDT simulations provide a very large number of observables characterizing quantum spacetime geometries (a typical MC configuration contains hundreds of thousands of degrees of freedom), for practical reasons we chose a set of 30 features characterizing purely geometric properties of the MC-generated triangulations. The features did not include any information about the values of the CDT coupling constants or other MC simulation parameters; see the Appendix for details. In order to apply supervised machine learning algorithms and validate the results (both in supervised and unsupervised learning approaches), a "manual" classification of the measurement datasets into individual CDT phases was performed. We tested three CDT phase transitions (the A-B, A-C, and  $B-C_b$  transitions), whose orders had previously been determined using standard methods of statistical physics based on order parameters. For each phase transition studied, we fixed one of the CDT coupling constants in the MC simulations ( $\Delta = 0.6$  for the A - C transition,  $\kappa_0 = 4.8$  for the A - B transition and  $\kappa_0 = 2.2$  for the  $B-C_b$  transition), and the phase transition was triggered by changing the other coupling constant ( $\kappa_0$  or  $\Delta$ , respectively).

For the purposes of data analysis and machine learning, we used built-in functions of Wolfram Mathematica 12: Classify and ClusterClassify for the supervised and unsupervised ML methods, respectively [23, 24]. Seven supervised ML methods (Decision Tree, Gradient Boosted Trees, Logistic Regression, Nearest Neighbors, Neural Network, Random Forest, Support Vector Machines) and seven unsupervised ML methods (Agglomerate, DBSCAN, Gaussian Mixture, K-Means, MeanShift, Neighborhood Contraction, Spectral) were tested [25]. We started by applying "Automatic" options, but certain machine learning algorithms required manual optimization of their hyperparameters. To test the effectiveness of those methods in identifying the phase transitions of the CDT model, the following procedure was adopted for each phase transition studied and each MC simulation volume  $\bar{N}_{41}$  separately:

- As input data for each ML algorithm, measurement results from a single MC simulation of the CDT model were selected for parameter values ( $\kappa_0$  or  $\Delta$ ) located deepest within each of the relevant phases (for instance, for the A-B transition, the input data corresponded to two points, one deepest inside phase A and the other one deepest inside phase B, respectively).
- The ML model was then trained on a subset of the above-defined data (*training dataset*).

- The next step was to verify whether the trained ML model was capable of correctly classifying the data as belonging to the appropriate phases (for supervised learning models) or as belonging to two distinct phases (for unsupervised learning models). Testing was carried out both on the training dataset and on a larger validation dataset (not used during the learning process).
- If the algorithm successfully passed these tests, i.e., it was able to classify or cluster the data measured deepest within the chosen phases with high accuracy (> 99.9%), the trained model was subsequently applied to classify or cluster data obtained from single MC simulations located closer to the respective phase transition. For each such dataset, the mean probability (as determined by the machine learning model) of belonging to a given phase/cluster was computed, along with the variance (susceptibility) of this probability.
- The phase transition point was then identified as the location where the probability changed from approximately 0 to approximately 1, which typically coincided with a sharp increase in the susceptibility of the classification/clustering probability.
- The location of the phase transition predicted by the ML model was then compared with the position of the transition determined by standard methods based on CDT order parameters.

As an example, take the A-B phase transition with fixed  $\kappa_0=4.8$  and lattice volume  $\bar{N}_{41}=100\,000$ . Using MC simulations we prepared ML datasets for 11 values of  $\Delta$ , ranging from  $\Delta_{\min}=-0.128$  (inside phase B) to  $\Delta_{\max}=-0.108$  (inside phase A). Then, for each of the ML models separately, the following procedure was applied:

- 1. Take all data measured for the highest value of  $\Delta$  ( $\Delta_{\text{max}} = -0.108$ ), i.e., deepest inside phase A.
- 2. Take all data measured for the lowest value of  $\Delta$  ( $\Delta_{\min} = -0.128$ ), i.e., deepest inside phase B.
- 3. Split data from points 1 and 2 into *training* and *validation* datasets.
- 4. Train the selected ML model using the training sets from points 1 and 2, this step also includes optimizing hyperparameters of the ML model, if necessary.
- 5. Check accuracy of the trained model using validation sets from points 1 and 2.
- 6. If the accuracy test was passed then use the trained ML model to classify / cluster other datasets measured for  $\Delta$  between  $\Delta_{\min}$  and  $\Delta_{\max}$ , i.e., closer to the phase transition point than data from points 1 and 2.

- 7. For each data point  $d_j(\Delta)$ , j=1,..., dataset length, that comes from a dataset measured for a given value of  $\Delta$  compute the probability  $\Pr(d_j(\Delta) \in A)$  that the data belong to phase A.
- 8. Compute the mean value of the probability  $\langle \Pr(d_j(\Delta) \in A) \rangle$  measured for each dataset  $(\Delta)$  and its susceptibility (variance).
- 9. Find the transition point predicted by a given ML model  $\Delta_{ML}^{crit}$ , where the probability  $\langle \Pr(d_j(\Delta) \in A) \rangle$  jumps from approximately 0 to approximately 1; see Fig. 2. That point is also usually associated with the jump in the measured susceptibility.
- 10. Compare  $\Delta_{ML}^{crit}$  with  $\Delta^{crit}$  measured using standard CDT order parameters.

### IV. RESULTS

The results obtained for all tested ML models using datasets comprising 30 selected features measured in the CDT Monte Carlo simulations are summarized in Fig. 3. For each studied phase transition, all seven supervised learning models were able to learn the classification of the individual phases of quantum gravity in CDT with high accuracy and, notably, without the need for manual hyperparameter optimization. Among them, five models (Gradient Boosted Trees, Logistic Regression, Nearest Neighbors, Neural Network, Support Vector Machines) produced consistent phase transition signals, in agreement with standard methods based on CDT order parameters; see Fig. 2 and Fig. 4. By contrast, two models (Decision Tree, Random Forest) indicated phase transition points at different locations; see Fig. 4. This is most likely due to the fact that in the learning process of these two models perfect classification to the respective phases could be done using just one of the measured features, but such an approach proved too simplistic in recognizing phase transition point(s) correctly. As expected, the performance of unsupervised learning models was worse. Most such models required manual hyperparameter optimization, with the choice depending on the type of phase transition under study. Moreover, their effectiveness depended strongly on the type of transition. Models that allowed the maximum number of clusters to be explicitly set to two (Agglomerate, K-Means, Spectral) performed relatively well. In contrast, models lacking such an option either required manual hyperparameter tuning (MeanShift) or failed to operate properly for certain

<sup>&</sup>lt;sup>1</sup> It is not entirely clear how much data preprocessing and hyper-parameter optimization is automatically done by Wolfram Mathematica 12 build-in ML functions that we used. We could not find such information in Wolfram's documentation.

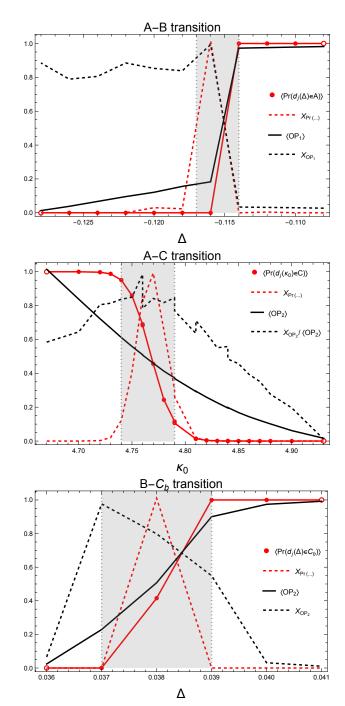


FIG. 2. Machine learning analysis of the A-B transition for fixed  $\kappa_0=4.8$ , the A-C transition for fixed  $\Delta=0.6$ , and the  $B-C_b$  transition for fixed  $\kappa_0=2.2$ . All data were measured for  $\bar{N}_{41}=100$ k. Red points indicate the mean probabilities that Monte Carlo–generated data at a given parameter value belong to one of the phases. The probabilities were computed with the Logistic Regression model, trained on subsets of data from the lowest and highest parameter values (empty dots). The solid black line denotes the standard CDT order parameter  $OP_1=N_0/N_{41}$  (A-B transition) or  $OP_2=N_{32}/N_{41}$  (A-C and  $B-C_b$  transitions), whereas the dashed lines denote the susceptibilities  $\chi$  of the probabilities and the order parameters;  $\langle OP_1 \rangle$ ,  $\langle OP_2 \rangle$  and  $\chi$  were rescaled and shifted to fit in range (0,1). The phase transition region is shaded.

	ML model	A-B (1 <sup>st</sup> order)	A-C ("weak" 1st order)	B-C <sub>b</sub> (higher-order)
Supervised learning	Decision Tree			
	Gradient Boosted Trees			
	Logistic Regression			
	Nearest Neighbors			
	Neural Network			
	Random Forest			
	Support Vector Machines			
Unsupervised learning	Agglomerate (2)			
	DBSCAN			
	Gaussian Mixture			
	K-Means (2)			
Der.	MeanShift			
Insu	Neighborhood Contraction			
_	Spectral (2)			

FIG. 3. Summary of the results obtained by different ML models in the study of individual phase transitions. Legend: dark green – the model correctly identifies phase transitions without the need for "manual" hyperparameter optimization; light green – the model correctly identifies phase transitions but requires "manual" hyperparameter optimization; yellow – the model identifies phase transitions, but produces results different from those of other models and standard methods; red – the model fails to work correctly.

phase transitions (*DBSCAN*, Gaussian Mixture, Neighborhood Contraction). In many cases (e.g., for some MC simulation volumes), these models identified too many clusters, or the resulting split of the data into clusters did not correspond to the actual division into CDT phases.

## V. DISCUSSION AND PROSPECTS

We have tested seven supervised and seven unsupervised machine learning methods in the analysis of three phase transitions observed in CDT. Most of the supervised models were demonstrated to be very efficient in correctly identifying phase transition points. Some unsupervised models, especially those allowing the number of clusters to be set to two, were also very successful. Remarkably, the probabilities generated by automated ML algorithms were not only consistent with standard statistical physics methods based on order parameters, but also produced much stronger transition signals. This allowed very precise identification of the phase transition points, outperforming traditional methods; see Fig. 2. Contrary to our expectations, more models (including the unsupervised methods) performed well in the case of the "weak" first-order A-C phase transition than in the case of the "typical" first-order A-B transition. The latter was correctly identified by as many models as the higher-order  $B-C_b$  phase transition.

The results presented here provide a promising foundation for further investigations into the applicability of ML techniques for detecting phase transitions in lattice quantum gravity approaches, such as CDT, or, more generally, in other lattice quantum field theories. Several natural directions for future research emerge:

• Unsupervised learning approaches. We plan to ex-

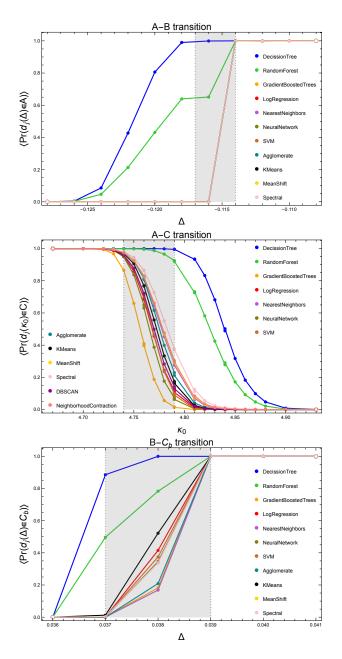


FIG. 4. Machine learning analysis of the A-B transition for fixed  $\kappa_0=4.8$ , the A-C transition for fixed  $\Delta=0.6$ , and the  $B-C_b$  transition for fixed  $\kappa_0=2.2$ . All data were measured for  $\bar{N}_{41}=100$ k. Tested ML models are denoted by different colors. Phase transition region is shaded. In the case of the A-B transition, results of models other than Decision Tree and Random Forest are optically indistinguishable.

tend the analysis to larger datasets containing a greater number of measured features and revisit unsupervised learning models that failed to perform satisfactorily in this study. The collection of such data from CDT Monte Carlo simulations is currently in progress.

• Multi-phase classification. An important extension is a test of the capability of machine learn-

ing algorithms to recognize more than just two phases simultaneously. This work is also currently in progress.

• Different spatial topologies. Thus far, our analysis has been restricted to CDT with toroidal spatial topology. Future work will include extending the study to CDT with spherical spatial topology, which will allow us to assess the robustness of the ML methods applied.

#### ACKNOWLEDGMENTS

The research was supported by a grant from the Priority Research Area DigiWorld under the Strategic Programme Excellence Initiative at Jagiellonian University. DN is supported by the VIDI programme with project number VI.Vidi.193.048, which is financed by the Dutch Research Council (NWO).

#### **APPENDIX**

Herein we list all the 30 features which were measured in the CDT Monte Carlo simulations and then input to the tested ML algorithms. All of the features are purely geometric observables characterizing CDT triangulations and do not include any CDT coupling constants or other parameters of the MC simulations.

- a. Global parameters:
- $N_0$  total number of vertices,
- $N_1$  total number of links,
- $N_2$  total number of triangles,
- $N_4$  total number of four-simplices,
- $N_{41}$  total number of (4,1)-simplices,
- MO maximal coordination number of vertices (maximal number of simplices sharing a vertex).
- b. Local parameters related to time-foliation:
- $N_{41}(t)$  the number of (4,1)-simplices with 4 vertices in t and 1 vertex in t+1,
- $N_{14}(t)$  the number of (4, 1)-simplices with 1 vertex in t and 4 vertices in t + 1,
- $N_{32}(t)$  the number of (3,2)-simplices with 3 vertices in t and 2 vertices in t+1,
- $N_{23}(t)$  the number of (3,2)-simplices with 2 vertices in t and 3 vertices in t+1,
- $N_0(t)$  the number of vertices with time coordinate t.

• MO(t) – the maximal coordination number of all vertices with time coordinate t.

In all cases t = 1, 2, 3, 4 (with periodic boundary conditions).

In order to encode the time shift symmetry of CDT, we quadrupled the dataset size by performing a time shift of

all local parameters by (periodically) changing their time coordinates  $t = (1, 2, 3, 4) \rightarrow (4, 1, 2, 3) \rightarrow (3, 4, 1, 2) \rightarrow (2, 3, 4, 1)$ . The values of the global parameters were kept unchanged.

The abovementioned observables also enable one to compute standard CDT order parameters used in phase transition studies:  $OP_1 = N_0/N_{41}$  and  $OP_2 = N_{32}/N_{41} = N_4/N_{41} - 1$ .

- S. Weinberg, "Ultraviolet Divergences in Quantum Theories of Gravitation," in *General Relativity: An Einstein Centenary Survey* (Cambridge University Press, 1980) pp. 790–831.
- [2] A. Codello, R. Percacci, and C. Rahmede, Annals of Physics 324, 414 (2009).
- [3] M. Reuter and F. Saueressig, New Journal of Physics 14, 055022 (2012).
- [4] M. Reuter and F. Saueressig, Quantum Gravity and the Functional Renormalization Group: The Road towards Asymptotic Safety, Cambridge Monographs on Mathematical Physics (Cambridge University Press, 2019).
- [5] J. Ambjørn, A. Görlich, J. Jurkiewicz, A. Kreienbuehl, and R. Loll, Class. Quant. Grav. 31, 165003 (2014), arXiv:1405.4585 [hep-th].
- J. Ambjorn, J. Gizbert-Studnicki, A. Görlich, J. Jurkiewicz, and R. Loll, Frontiers in Physics Volume 8
   2020 (2020), 10.3389/fphy.2020.00247.
- [7] J. Ambjørn, J. Gizbert-Studnicki, A. Görlich, and D. Németh, Phys. Rev. D 110, 126006 (2024).
- [8] J. Ambjorn, J. Jurkiewicz, and R. Loll, Phys. Rev. D 72, 064014 (2005), arXiv:hep-th/0505154.
- [9] J. Ambjørn, A. Görlich, S. Jordan, J. Jurkiewicz, and R. Loll, Phys. Lett. B 690, 413 (2010), arXiv:1002.3298 [hep-th].
- [10] J. Ambjørn, D. Coumbe, J. Gizbert-Studnicki, and J. Jurkiewicz, JHEP 08, 033 (2015), arXiv:1503.08580 [hep-th].
- [11] J. Ambjørn, J. Jurkiewicz, and R. Loll, Nucl. Phys. B 610, 347 (2001), arXiv:hep-th/0105267.
- [12] J. Ambjørn, A. Görlich, J. Jurkiewicz, and R. Loll, Phys.

- Rept. **519**, 127 (2012), arXiv:1203.3591 [hep-th].
- [13] R. Loll, Class. Quant. Grav. 37, 013002 (2020), arXiv:1905.08669 [hep-th].
- [14] J. Ambjørn, Z. Drogosz, J. Gizbert-Studnicki, A. Görlich, J. Jurkiewicz, and D. Németh, Universe 7, 79 (2021), arXiv:2103.15610 [gr-qc].
- [15] T. Regge, Nuovo Cim. **19**, 558 (1961).
- [16] J. Ambjørn, S. Jordan, J. Jurkiewicz, and R. Loll, Phys. Rev. D 85, 124044 (2012).
- [17] J. Ambjørn, J. Gizbert-Studnicki, A. Görlich, J. Jurkiewicz, and D. Németh, JHEP 06, 111 (2018), arXiv:1802.10434 [hep-th].
- [18] J. Ambjørn, D. Coumbe, J. Gizbert-Studnicki, A. Görlich, and J. Jurkiewicz, Classical and Quantum Gravity 36, 224001 (2019).
- [19] J. Ambjørn, J. Gizbert-Studnicki, A. Görlich, and D. Németh, JHEP 04, 103 (2022).
- [20] J. Ambjørn, S. Jordan, J. Jurkiewicz, and R. Loll, Phys. Rev. Lett. 107, 211303 (2011).
- [21] D. Coumbe, J. Gizbert-Studnicki, and J. Jurkiewicz, JHEP 02, 144 (2016), arXiv:1510.08672 [hep-th].
- [22] J. Ambjørn, G. Czelusta, J. Gizbert-Studnicki, A. Görlich, J. Jurkiewicz, and D. Németh, JHEP 05, 030 (2020), arXiv:2002.01051 [hep-th].
- [23] W. Research, "Classify," https://reference.wolfram. com/language/ref/Classify.html (2025).
- [24] W. Research, "ClusterClassify," https://reference. wolfram.com/language/ref/ClusterClassify.html (2025).
- [25] W. Research, "Machine Learning Methods," https://reference.wolfram.com/language/guide/ MachineLearningMethods.html (2025).