

A Separability-Entanglement Classifier via Machine Learning

Sirui Lu,^{1,*} Shilin Huang,^{2,3,*} Keren Li,^{1,2} Jun Li,^{2,4,5,†} Jianxin Chen,⁶
 Dawei Lu,^{2,7,5,‡} Zhengfeng Ji,^{8,9} Yi Shen,¹⁰ Duanlu Zhou,¹¹ and Bei Zeng^{5,2,7}

¹*Department of Physics, Tsinghua University, Beijing, 100084, China*

²*Institute for Quantum Computing, University of Waterloo, Waterloo N2L 3G1, Ontario, Canada*

³*Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, China*

⁴*Beijing Computational Science Research Center, Beijing, 100193, China*

⁵*Department of Mathematics & Statistics, University of Guelph, Guelph N1G 2W1, Ontario, Canada*

⁶*Joint Center for Quantum Information and Computer Science,
University of Maryland, College Park, Maryland, USA*

⁷*Department of Physics, Southern University of Science and Technology, Shenzhen 518055, China*

⁸*Centre for Quantum Computation & Intelligent Systems, School of Software,*

Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia

⁹*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China*

¹⁰*Department of Statistics and Actuarial Science, University of Waterloo, Waterloo N2L 3G1, Ontario, Canada*

¹¹*Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China*

(Dated: June 15, 2017)

The problem of determining whether a given quantum state is entangled lies at the heart of quantum information processing, which is known to be an NP-hard problem in general. Despite the proposed many methods such as the positive partial transpose (PPT) criterion and the k -symmetric extendibility criterion to tackle this problem in practice, none of them enables a general, effective solution to the problem even for small dimensions. Explicitly, separable states form a high-dimensional convex set, which exhibits a vastly complicated structure. In this work, we build a new separability-entanglement classifier underpinned by machine learning techniques. Our method outperforms the existing methods in generic cases in terms of both speed and accuracy, opening up the avenues to explore quantum entanglement via the machine learning approach.

Born from pattern recognition, machine learning possesses the capability to make decisions without being explicitly programmed after learning from large amount of data. Beyond its extensive applications in industry, machine learning has also been employed to investigate physics-related problems in recent years. A number of promising applications have been proposed to date, such as the Hamiltonian learning [1], automated quantum experiments generation [2], identification of phases and phase transition [3–5], efficient representation of quantum many-body states [6, 7], just to name a few. Nevertheless, there are yet a myriad of significant but hard problems in physics to be assessed, in which should machine learning provide more novel insights. For example, to determine whether a generic quantum state is entangled or not is a fundamental and NP-hard problem in quantum information processing [8], and machine learning is demonstrated to be exceptionally effective in tackling it as shown in this work.

As one of the key features in quantum mechanics, entanglement allows two or more parties to be correlated in a way that is much stronger than they can be in any classical way [9]. It also plays a key role in many quantum information processing tasks such as teleportation and quantum key distribution [10]. As a result, one question naturally arises: is there a universal criterion to tell if an arbitrary quantum state is separable or entangled? This is a typical classification problem, which remains of great challenge even for bipartite states. In fact, such an entanglement detection problem is proved to be NP-hard [8], implying that it is almost impossible to devise an efficient algorithm in complete generality.

Here, we focus on the task of detecting bipartite entangle-

ment. Consider a bipartite system AB with the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$, where \mathcal{H}_A has dimension d_A and \mathcal{H}_B has dimension d_B , respectively. A state ρ_{AB} is separable if it can be written as a convex combination $\rho_{AB} = \sum_i \lambda_i \rho_{A,i} \otimes \rho_{B,i}$ with a probability distribution $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. Here $\rho_{A,i}$ and $\rho_{B,i}$ are density operators acted on \mathcal{H}_A , \mathcal{H}_B respectively. Otherwise, ρ_{AB} is entangled [11, 12]. To date, many criteria have been proposed to detect bipartite entanglement, each with its own pros and cons. For instance, the most famous criterion is the positive partial transpose (PPT) criterion, saying that a separable state must have PPT; however, it is only necessary and sufficient when $d_A d_B \leq 6$ [13, 14]. Another widely used one is the k -symmetric extension hierarchy [15, 16], which is presently one of the most powerful criteria, but hard to compute in practice due to its exponentially growing complexity with k [17].

In this work, we employ the machine learning techniques to tackle the bipartite entanglement detection problem by recasting it as a learning task, namely we attempt to construct a separability-entanglement classifier. Due to its renowned effectiveness in pattern recognition for high-dimensional objects, machine learning is a powerful tool to solve the above problem. In particular, a reliable separability-entanglement classifier in terms of speed and accuracy is constructed via the supervised learning approach. The idea is to feed our classifier by a large amount of sampled trial states as well as their corresponding class labels (separable or entangled), and then train the classifier to predict the class labels of new states that it has not encountered before. It is worthy stressing that, there is also a remarkable improvement with respect to universality

*Corresponding author. E-mail: shilinhuang@ust.hk

†Corresponding author. E-mail: jli@ust.hk

‡Corresponding author. E-mail: dawei@ust.hk

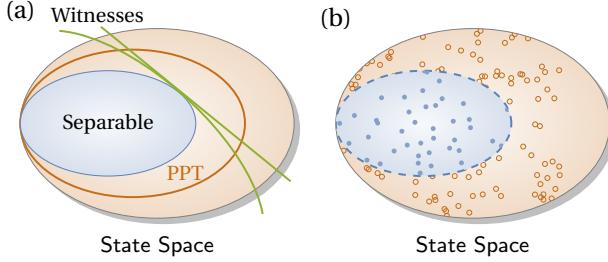


FIG. 1. (a) In the high-dimensional space, the set of all states is convex, while separable states form a convex subset. Many criteria, such as the linear (green straight line) or nonlinear (green curve) entanglement witnesses [9, 18] and PPT tests, are based on this geometric structure and detect a limited set of entangled states. (b) Classifier built from supervised learning has a decision boundary of highly complex shape.

in our classifier compared to the conventional methods. Previous methods only detect a limited part of the state space, e.g. different entangled states often require different entanglement witnesses. In contrast, our classifier can handle a variety of input states once properly trained, as shown in Fig. 1.

Supervised learning – The bipartite entanglement detection problem can be formulated as a supervised binary classification task. Following the standard procedure of supervised learning [19, 20], the feature vector representation of the input objects (states) in a bipartite system AB is first created. Indeed, any quantum state ρ , as a density operator acting on $\mathcal{H}_A \otimes \mathcal{H}_B$ can be represented as a real vector in $\mathcal{X} = \mathbb{R}^{d_A^2 d_B^2 - 1}$, which is due to the fact that ρ is Hermitian and of trace 1 (see Supplementary Material [17]). In the machine learning language, we refer x as the feature vector of ρ and \mathcal{X} the feature space.

Next, a dataset of training examples is produced, with the form $\mathcal{D}_{\text{train}} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, where n is the size of the set, $x_i \in \mathcal{X}$ is the i -th sample, and y_i is its corresponding label signifying which class it belongs to, i.e., y_i equals to 1 if x_i is entangled or -1 otherwise. When $d_A d_B \leq 6$, the labeling process can be directly computed via the PPT criterion. For higher-dimensional cases, we attempt to estimate the labels by convex hull approximation, which we will describe later. The task is to analyze these training data and produce an inferred classifier that predicts the unknown class labels for generic new input states.

Explicitly, the aim of supervised learning is to infer a function (classifier) $h : \mathcal{X} \rightarrow \{-1, 1\}$ among a fixed class of functions \mathcal{H} such that h is expected to be close to the true decision function. One basic approach to choose h is the so-called empirical risk minimization, which seeks the function that best fits the training data among the class \mathcal{H} . In particular, to evaluate how well h fits the training data $\mathcal{D}_{\text{train}}$, a loss function is defined as

$$\mathcal{L}(h, \mathcal{D}_{\text{train}}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{train}}} \mathbb{1}(y_i \neq h(x_i)), \quad (1)$$

where $\mathbb{1}(\cdot)$ is the truth function of its arguments. For a generic new input test dataset $\mathcal{D}_{\text{test}}$ that contains previously unseen data, function $\mathcal{L}(h, \mathcal{D}_{\text{test}})$ gives a quantification of the generalization error from $\mathcal{D}_{\text{train}}$ to $\mathcal{D}_{\text{test}}$.

Numerous supervised learning algorithms have been developed, each with its strength and weakness. These algorithms, which have distinct choices of class \mathcal{H} , include support vector machine (SVM) [21], decision trees [22], bootstrap aggregating [23], and boosting [24], etc. We have applied these algorithms to the separability problem directly, but neither of them provided an acceptable accuracy, which is mainly due to the lack of prior knowledges for training, e.g., the geometric shape of the set of separable states \mathcal{S} . Taking the kernel SVM approach [21] as an example, it uses a kernel function to map data from the original feature space to another Hilbert space, and then finds a hyperplane in the new space to split the data into two subclasses. It turns out that using common kernels such as radial basis function and polynomials, the error rate on the test dataset is always around 10% (see Supplementary Material [17] for details). This suggests that the boundary of \mathcal{S} is too complicated to be portrayed by manifolds with ordinary shapes.

Convex hull approximation – The above discussions suggest that it is desirable to examine the detailed geometric shape of \mathcal{S} in advance. One well-known approach is to approximate \mathcal{S} from outside via k -symmetric extendible set Θ_k , where $\Theta_k \supset \Theta_{k+1}$ and Θ_k converges exactly to \mathcal{S} as k goes to infinity [31]. Unfortunately, it is impractical to compute the boundary of Θ_k for large k , while it is still far from approximating \mathcal{S} for small k [17].

However, it is much easier to approximate \mathcal{S} from inside, since \mathcal{S} is a closed convex set, and its extreme points are exactly all the separable pure states, which can be straightforwardly parameterized and generated numerically. We randomly sample m separable pure states $c_1, \dots, c_m \in \mathcal{X}$ to form a convex hull $\mathcal{C} := \text{conv}(\{c_1, \dots, c_m\})$. \mathcal{C} is said to be a convex hull approximation (CHA) of \mathcal{S} , with which we can approximately tell whether a state ρ is separable or not by testing if its feature vector p is in \mathcal{C} . This is equivalent to determining whether p can be written as a convex combination of c_i by solving the following linear programming:

$$\begin{aligned} & \max \quad \alpha \quad \text{s.t.} \quad \alpha p \in \mathcal{C}, \\ & \text{i.e.} \quad \alpha p = \sum_{i=1}^m \lambda_i c_i, \quad \lambda_i \geq 0, \quad \sum_i \lambda_i = 1. \end{aligned} \quad (2)$$

Here $\alpha = \alpha(\mathcal{C}, p)$ is a function of \mathcal{C} and p . If $\alpha(\mathcal{C}, p) \geq 1$, p is in \mathcal{C} and thus ρ is separable; otherwise, ρ is highly possible to be an entangled state. In principle, \mathcal{C} will be a more accurate CHA of \mathcal{S} if we construct \mathcal{C} with more extreme points. We test the error rate of CHA on a set of 2×10^4 random two-qubit states, which is sampled under a specified distribution [17] and labeled by PPT criterion. The results are shown by the blue curve in Fig. 3(c), where the error rate decreases quickly to 3% when the number of extreme points m increases to 10^4 .

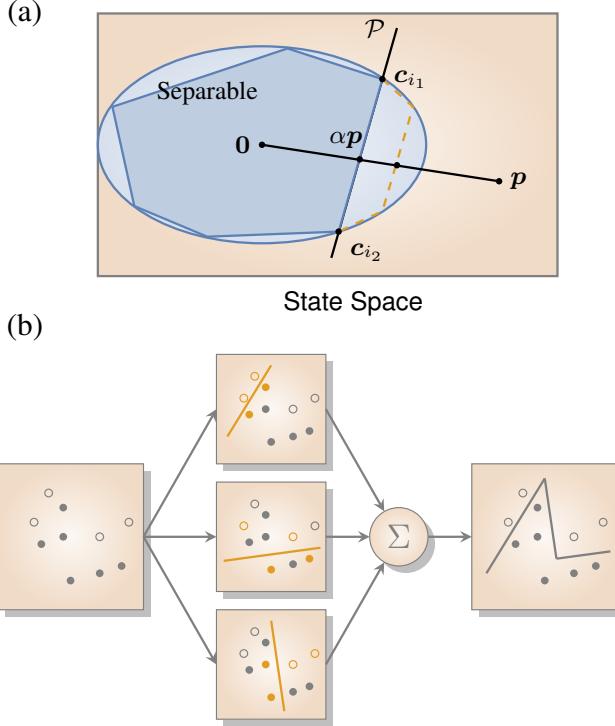


FIG. 2. (a) Illustration of the iterative algorithm for detecting the separability. Initially we build a CHA \mathcal{C} . For a state ρ with feature vector p , we find the maximum α such that αp is still in \mathcal{C} . If $\alpha \geq 1$, then ρ is surely separable. Otherwise, suppose αp lies on a hyperplane \mathcal{P} , such that $\mathcal{P} \cap \mathcal{C}$ is the boundary of \mathcal{C} . Let c_{i_1}, \dots, c_{i_D} be extreme points of \mathcal{C} that are in $\mathcal{P} \cap \mathcal{C}$. We enlarge \mathcal{C} by sampling separable pure states that are near c_{i_1}, \dots, c_{i_D} , then repeat the above procedure for many times, until $\alpha \geq 1$ or α converges. (b) Illustration of the learning algorithm: ensemble methods are models composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction. For example, each time we draw a subset of training data (marked as yellow dots in the figure), and we train a model based on this subset, which is a weak model on the whole training set. We repeat the process for many times and obtain a batch of weak models, and combine them as a committee.

However, we can not directly test the accuracy of CHA on generic two-qutrit states, since PPT criterion is no longer sufficient for detecting separability. To illustrate the power of CHA beyond the PPT criterion, we use a specific example that is previously well-studied. Consider a set of two-qutrits pure states $\{|v_1\rangle, \dots, |v_5\rangle\}$ that form the well-known unextendible product basis [26], where $|v_1\rangle = (|00\rangle - |01\rangle)/\sqrt{2}$, $|v_2\rangle = (|21\rangle - |22\rangle)/\sqrt{2}$, $|v_3\rangle = (|02\rangle - |12\rangle)/\sqrt{2}$, $|v_4\rangle = (|10\rangle - |20\rangle)/\sqrt{2}$, and $|v_5\rangle = (|0\rangle + |1\rangle + |2\rangle)^{\otimes 2}/3$. It is known that $\rho_{\text{tiles}} = (\mathbb{I} - \sum_{i=1}^5 |v_i\rangle\langle v_i|)/4$ is an entangled state with PPT [27]. Due to the fact that \mathcal{S} is convex and closed, there must exist a unique critical point $\alpha_{\text{tiles}} \in [0, 1]$ such that $\alpha_{\text{tiles}}\rho + (1 - \alpha_{\text{tiles}})\mathbb{I}/(d_A d_B)$, the probabilistic mixture of ρ and the maximally-mixed state $\mathbb{I}/(d_A d_B)$, is on the boundary of \mathcal{S} . Ref. [28] compared the effectiveness of various sepa-

m	2000	5000	10000	20000	50000	100000
$\alpha(\mathcal{C}, p_{\text{tiles}})$	0.5264	0.5868	0.6387	0.6759	0.7150	0.7459

TABLE I. Numerical results for approximating α_{tiles} by $\alpha(\mathcal{C}, p_{\text{tiles}})$. Here, m is the number of random extreme points for building \mathcal{C} .

rability criteria, and concluded that $\alpha_{\text{tiles}} \in (0.5643, 0.8649]$. Note that for a CHA \mathcal{C} , $\alpha(\mathcal{C}, p_{\text{tiles}})$ actually provides a lower bound approximation of α_{tiles} , where p_{tiles} is the feature vector of ρ_{tiles} . Now we apply CHA and attempt to improve the lower bound of α_{tiles} , and the result is shown in Table I.

We find that the lower bound of α_{tiles} has been raised to 0.7459. However, in Table I, the value of $\alpha(\mathcal{C}, p_{\text{tiles}})$ has not converged yet. To reach the convergence of $\alpha(\mathcal{C}, p_{\text{tiles}})$, we have to enlarge \mathcal{C} by adding more extreme points. However, note that the point $\alpha(\mathcal{C}, p_{\text{tiles}})$ lies on a part of the boundary of \mathcal{C} , which is the intersection of a hyperplane and \mathcal{C} . Let c_{i_1}, \dots, c_{i_D} be the extreme points of \mathcal{C} that lie on the hyperplane as well. Clearly, if we enlarge \mathcal{C} by sampling the separable pure states that are near c_{i_1}, \dots, c_{i_D} rather than sampling uniformly over the whole set of separable pure states, it will boost the value of $\alpha(\mathcal{C}, p_{\text{tiles}})$ more effectively.

Subsequently, we refine CHA as an iterative algorithm [17], with the idea shown in Fig. 2(a). The iterative algorithm gives the result $\alpha_{\text{tiles}} > 0.8648$. As the upper bound of α_{tiles} is 0.8649 [28], we can explicitly conclude that $\alpha_{\text{tiles}} \approx 0.8649$. It is worthy emphasizing that the algorithm also gives the critical point for a generic entangled state with small error, and detects the separability for generic separable states [17].

Combining CHA and supervised learning – There is yet a noticeable drawback of the above CHA approach from the perspective of the tradeoff between the accuracy and time consumption. Boosting the accuracy means adding additional extreme points to enlarge the convex hull, which leads to more time costs to determine if a point is inside the enlarged convex hull or not. To overcome this, we combined CHA with supervised learning, as machine learning has the power to speed up such computations.

To design a learning process that is suitable for our problem, for each state ρ with feature vector p , we extend the feature vector as $(p, \alpha(\mathcal{C}, p))$ in order to encode the geometric information of the CHA \mathcal{C} into the dataset. In this manner, the training dataset is written as $\mathcal{D}_{\text{train}} = \{(x_1, \alpha_1, y_1), \dots, (x_n, \alpha_n, y_n)\}$, where $\alpha_i = \alpha(\mathcal{C}, x_i)$. A classifier h is now a binary function defined on $\mathcal{X} \times \mathbb{R}$, and the loss function of a classifier h is then redefined as

$$\mathcal{L}(h, \mathcal{D}_{\text{train}}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x_i, \alpha_i, y_i) \in \mathcal{D}_{\text{train}}} \mathbb{1}(y_i \neq h(x_i, \alpha_i)). \quad (3)$$

Subsequently, we employ a standard ensemble learning approach [23] to train a classifier with training data $\mathcal{D}_{\text{train}}$, described as follows.

The essential idea of ensemble learning is to improve the predictive performance by combining multiple classifiers into

a committee. Even though the prediction from each constituent might be poor, the combined classifier could often still perform excellent. For the binary classification problem, we can train different classifiers to give their respective binary votes for each prediction, and use the majority rule to choose the value which receives more than half votes as the final answer, see Fig. 2(c) for a schematic diagram.

Here, we choose bootstrap aggregating (bagging) [29] as our training ensemble algorithm. In each run, a training subset is randomly drawn from the whole set $\mathcal{D}_{\text{train}}$, and a model is trained from the training subset using another learning algorithm, e.g., decision trees learning. We repeat the process for $L = 100$ times and obtain L different models, which are finally combined together as the committee. Since

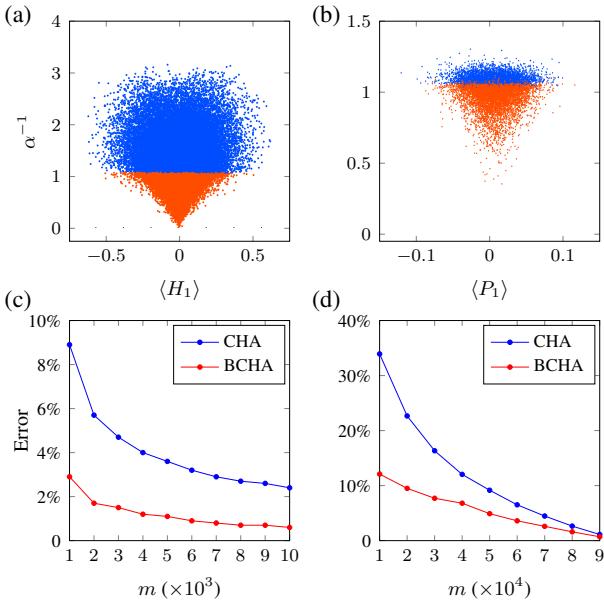


FIG. 3. Results of the CHA approach and BCHA classifier. More details of the BCHA classifier can be found in [17]. (a) Test results of BCHA when $m = 10^3$ for the two-qubit case. The random density matrices in the test dataset are projected on a plane by projection $\pi : (x, \alpha) \rightarrow (x_1, \alpha^{-1})$. Here $H_1 = |0\rangle\langle 0| \otimes \sigma_z/\sqrt{2}$. The red points are states predicted as separable, while the blue points are states predicted as entangled. Some states with $\alpha < 1$ are predicted as separable, which is different from CHA. (b) Test results of BCHA when $m = 2 \times 10^4$ for the two-qutrit case. Here $P_1 = (|00\rangle\langle 00| - |01\rangle\langle 01|)\sqrt{3}/2$. All the states in the test dataset are PPT states. The red points are states predicted as separable, while the blue points are states predicted as bound entangled. (c) Comparison between CHA and BCHA for two-qubit states. For the same m , BCHA clearly suppresses the error rate significantly. And to achieve the same error rate, BCHA requires much less running time (which mainly depends on the value of m). For instance, to decrease the error rate to less than 3%, CHA requires a convex hull with $m \approx 7 \times 10^3$, while BCHA only requires a convex hull with $m \approx 10^3$, which considerably reduces the computational cost. (d) Comparison between CHA and BCHA for two-qutrit PPT states. From the data, similar to the two-qubit case, BCHA also outperforms CHA in terms of speed and accuracy.

$\alpha_i = \alpha(\mathcal{C}, x_i)$ contains the geometric information of CHA \mathcal{C} , our method is indeed a combination of bagging and CHA. We call this combined method BCHA.

The computational cost contains two parts: the cost of computing α_i via linear programming, and the time of computing each constituent in the committee. The latter cost is much smaller than the former. Therefore, by using a convex hull of much smaller size and implementing a bagging algorithm, a significant boost in terms of accuracy is anticipated if the total computational cost is fixed. For the two-qubit case, we have demonstrated such a remarkable boost of accuracy in our BCHA classifier, as shown in Fig. 3(c), where the advantages of the BCHA classifier in terms of both accuracy and speed are shown.

We further extend the classifier to the two-qutrit scenario. Unlike the two-qubit case, the critical question now is how to set an appropriate criterion to evaluate whether the classifier is working correctly, since PPT criterion is not sufficient for detecting separability in two-qutrit systems. As the convex hull is capable of approximating the set of separable states \mathcal{S} to an arbitrary precision, we use 10^5 random separable pure states as extreme points to form the hull, and assumed it to be the true \mathcal{S} . The learning procedure is analogous to the one used for two qubits. Figure 3(d) shows the accuracy of the BCHA classifier compared to that of the sole CHA approach. Similar as the two-qubit case, the BCHA classifier shows clear advantage in terms of both accuracy and speed in comparison with the sole CHA method.

Conclusion – In summary, we study the entanglement detection problem via the machine learning approach, and build a reliable separability-entanglement classifier by combining supervised learning and the CHA method. Compared to the conventional criteria for entanglement detection, our method can classify an unknown state into the separable or entangled category more precisely and rapidly. The classifier can be extended to higher dimensions in principle, and the developed techniques in this work would also be incorporated in future entanglement-engineering experiments. We anticipate that our work would provide new insights to employ the machine learning techniques to deal with more quantum information processing tasks in the near future.

Acknowledgments – This work is supported by Chinese Ministry of Education under grants No.20173080024. We thank Daniel Gottesman and Nathaniel Johnston for helpful discussions. JL is supported by the National Basic Research Program of China (Grants No. 2014CB921403, No. 2016YFA0301201, No. 2014CB848700 and No. 2013CB921800), National Natural Science Foundation of China (Grants No. 11421063, No. 11534002, No. 11375167 and No. 11605005), the National Science Fund for Distinguished Young Scholars (Grant No. 11425523), and NSAF (Grant No. U1530401). DL and BZ are supported by NSERC and CIFAR. JC was supported by the Department of Defense (DoD). DZ is supported by NSF of China (Grant No. 11475254), BNKBRSF of China (Grant No. 2014CB921202), and The National Key Research and Devel-

opment Program of China(Grant No. 2016YFA0300603).

* These authors contributed equally to this work.

[†] lijunwu@mail.ustc.edu.cn

[‡] d29lu@uwaterloo.ca

- [1] N. Wiebe, C. Granade, C. Ferrie, and D. G. Cory, *Phys. Rev. Lett.* **112**, 190501 (2014).
- [2] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, *Phys. Rev. Lett.* **116**, 090405 (2016).
- [3] S. S. Schoenholz, E. D. Cubuk, D. M. Sussman, E. Kaxiras, and A. J. Liu, *Nat. Phys.* **12**, 469 (2016).
- [4] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, *Nat. Phys.* (2017).
- [5] J. Carrasquilla and R. G. Melko, *Nat. Phys.* (2017).
- [6] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, arXiv:1701.04831 (2017).
- [7] X. Gao and L.-M. Duan, arXiv:1701.05039 (2017).
- [8] L. Gurvits, in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing* (ACM, 2003) pp. 10–19.
- [9] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, *Rev. Mod. Phys.* **81**, 865 (2009).
- [10] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge university press, 2010).
- [11] R. F. Werner, *Phys. Rev. A* **40**, 4277 (1989).
- [12] O. Guhne and G. Tth, *Phys. Rep.* **474**, 1 (2009).
- [13] A. Peres, *Phys. Rev. Lett.* **77**, 1413 (1996).
- [14] M. Horodecki, P. Horodecki, and R. Horodecki, *Phys. Lett. A* **223**, 1 (1996).
- [15] M. Navascu s, M. Owari, and M. B. Plenio, *Phys. Rev. A* **80**, 052306 (2009).
- [16] F. G. Brand o, M. Christandl, and J. Yard, in *Proceedings of the forty-third annual ACM symposium on Theory of computing* (ACM, 2011) pp. 343–352.
- [17] See Supplementary Material for more details.
- [18] O. Guhne and N. Ltkenhaus, *Phys. Rev. Lett.* **96**, 170502 (2006).
- [19] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning* (The MIT Press, 2012).
- [20] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms* (Cambridge University Press, 2014).
- [21] C. Cortes and V. Vapnik, *Mach. Learn.* **20**, 273 (1995).
- [22] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees* (CRC press, 1984).
- [23] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms* (Taylor & Francis Group, 2012).
- [24] R. E. Schapire, in *Nonlinear estimation and classification* (Springer, 2003) pp. 149–171.
- [25] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri, *Phys. Rev. Lett.* **88**, 187904 (2002).
- [26] I. Bengtsson and K. Zyczkowski, *Geometry of quantum states: an introduction to quantum entanglement* (Cambridge University Press, 2007).
- [27] C. H. Bennett, D. P. DiVincenzo, T. Mor, P. W. Shor, J. A. Smolin, and B. M. Terhal, *Phys. Rev. Lett.* **82**, 5385 (1999).
- [28] N. Johnston, “Entanglement detection,” (2014).
- [29] L. Breiman, *Mach. Learn.* **24**, 123 (1996).
- [30] R. A. Bertlmann and P. Krammer, *Journal of Physics A: Mathematical and Theoretical* **41**, 235303 (2008).
- [31] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri, *Phys. Rev. Lett.* **88**, 187904 (2002).
- [32] B. Zeng, X. Chen, D.-L. Zhou, and X.-G. Wen, arXiv:1508.02595 (2015).
- [33] A. A. Klyachko, *J. Phys.: Conf. Ser.* **36**, 72 (2006).
- [34] A. J. Coleman, *Rev. Mod. Phys.* **35**, 668 (1963).
- [35] R. M. Erdahl, *J. Math. Phys.* **13**, 1608 (1972).
- [36] Y.-K. Liu, in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Lecture Notes in Computer Science, Vol. 4110, edited by J. Diaz, K. Jansen, J. D. Rolim, and U. Zwick (Springer Berlin Heidelberg, 2006) pp. 438–449.
- [37] Y.-K. Liu, M. Christandl, and F. Verstraete, *Phys. Rev. Lett.* **98**, 110503 (2007).
- [38] T.-C. Wei, M. Mosca, and A. Nayak, *Phys. Rev. Lett.* **104**, 040501 (2010).
- [39] K. Zyczkowski, *Phys. Rev. A* **60**, 3496 (1999).
- [40] N. Johnston, “Qetlab: A matlab toolbox for quantum entanglement, version 0.9,” (2016).
- [41] “Qmlab: Global collaboration on quantum machine learning,” (2017).

A. Generalized Gell-Mann Matrices

To represent a n -by- n density matrix ρ as a real vector x in \mathbb{R}^{n^2-1} , we can find a Hermitian orthogonal basis that contains identity such that ρ can be expanded in such a basis with real coefficients. For example, the Pauli basis is a commonly used one. In our numerical tests, we take the generalized Gell-Mann matrices and the identity as the Hermitian orthogonal basis. In this section, we recall the definition of the generalized Gell-Mann matrices, which is shown in [30].

Let $\{|1\rangle, \dots, |n\rangle\}$ be the computational basis of the n -dimensional Hilbert space, and $E_{j,k} = |j\rangle\langle k|$. We now define three collections of matrices. The first collection is symmetric:

$$s_{j,k} = E_{j,k} + E_{k,j}$$

for $1 \leq j < k \leq n$. The second collection is antisymmetric:

$$a_{j,k} = -i(E_{j,k} - E_{k,j})$$

for $1 \leq j < k \leq n$. The last collection is diagonal:

$$d_l = \sqrt{\frac{2}{l(l+1)}} \left(\sum_{j=1}^l E_{j,j} - lE_{l+1,l+1} \right)$$

for $1 \leq l \leq n-1$.

The generalized Gell-Mann matrices are elements in the set $\{\lambda_i\} = \{s_{j,k}\} \cup \{a_{j,k}\} \cup \{d_l\}$, which gives a total of $n^2 - 1$ matrices. We can easily check that

$$\text{tr}(\lambda_i) = \text{tr}(\lambda_i \mathbb{I}) = 0$$

and

$$\text{tr}(\lambda_i \lambda_j) = 2\delta_{ij},$$

which implies that $\{\lambda_i\} \cup \{\mathbb{I}\}$ forms an orthogonal basis of observables in n -dimensional Hilbert space.

For every n -by- n density matrix ρ , ρ can be expressed as a linear combination of λ_i and \mathbb{I} as follows:

$$\rho = \frac{1}{n} \left(\mathbb{I} + \sqrt{\frac{n(n-1)}{2}} x \cdot \vec{\lambda} \right),$$

where $x = (x_1, x_2, \dots, x_{n^2-1}) \in \mathbb{R}^{n^2-1}$ satisfies

$$x_i = \sqrt{\frac{n}{2(n-1)}} \text{tr}(\rho \lambda_i).$$

B. The set of k -extendible states

In this section, we recall facts regarding k -extendible states and its relationship to separability.

A bipartite state ρ_{AB} is said to be k -symmetric extendible if there exists a global state $\rho_{AB_1\dots B_k}$ whose reduced density matrices ρ_{AB_i} are equal to ρ_{AB} for $i = 1, \dots, k$. The set of all k -extendible states, denoted by Θ_k , is convex with a hierarchy structure $\Theta_k \supset \Theta_{k+1}$. Moreover, when $k \rightarrow \infty$, Θ_k converges exactly to the set of separable states [31].

The Θ_k is known to be closely related to the ground state of some $(k+1)$ -body Hamiltonians [32]. To be more precise, consider a 2-local Hamiltonian H of a $(k+1)$ -body system with Hilbert space $\mathbb{C}^{d_A} \otimes_{i=1}^k \mathbb{C}^{d_{B_i}}$ of dimension $d_A d_B^k$, as given in the following form $H = \sum_{i=1}^k H_{AB_i}$. Here H_{AB_i} is any Hermitian operator acting nontrivially on particles A and B_i , and trivially on other $k-1$ parties. In other words, we will have $H_{AB_1} = H_{AB} \otimes I_{2,\dots,k}$ ($I_{2,\dots,k}$ is the identity operator of B_2, \dots, B_k), and given the symmetry of B_i s, we can always write the nontrivial action of H_{AB_i} on $\mathbb{C}^{d_A} \otimes_{i=1}^k \mathbb{C}^{d_{B_i}}$ in terms of some H_{AB} acted on $d_A d_B$ -dimensional Hilbert space.

For any given H , denote its normalized ground state by $|\psi_g\rangle \in \mathbb{C}^{d_A} \otimes_{i=1}^k \mathbb{C}^{d_{B_i}}$, and $\rho_g = |\psi_g\rangle\langle\psi_g|$. Then the extreme points of Θ_k are given by the marginals of ρ_g on particles AB_i , which are the same for any i . Denote this marginal by ρ_H , since it is completely determined by H .

To generate random extreme points of Θ_k , we will need to first parametrize them. Denote $\{O_{AB}^{lm}\}$ as set of orthonormal Hermitian basis for operators on $H_A \otimes H_{B_j}$ (see section A), then we can always write $H_{AB} = \sum_{lm} a_{lm} O_{AB}^{lm}$, with parameters a_{lm} . Without loss of generality, we assume $O_{AB}^{00} = I$, and we will assume $a_{00} = 0$, so there are only $d_A^2 d_B^2 - 1$ terms in the sum. Since H_{AB} is a Hermitian matrix, a_{lm} can be chosen as real, and we can further require that $\sum_{l,m} a_{lm}^2 = 1$. Consequently, ρ_H will be a point in $\mathbb{R}^{d_A^2 d_B^2 - 1}$, which is parametrized by $\{a_{lm}\}$. And the coordinate of ρ_H are explicitly given by $b_{lm} = \text{tr}(\rho_H O_{AB}^{lm})$.

Also, each H_{AB} gives an entanglement witness. The ground state energy of H_{AB} is given by $E_0 = \langle\psi_g|H|\psi_g\rangle = \sum_{lm} a_{lm} b_{lm}/k$. For any density matrix ρ_{AB} , if $\text{tr}(\rho_{AB} H_{AB}) < E_0$, then ρ_{AB} has no k -symmetric extension, hence is surely entangled.

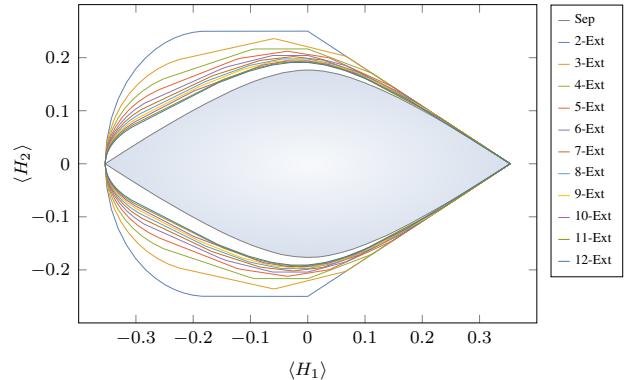


FIG. S4. Projections of the boundaries of the separable states and the k -symmetric extendible states ($k = 2, \dots, 12$) on the plane spanned by the operators $H_1 = |0\rangle\langle 0| \otimes \sigma_z/\sqrt{2}$ and $H_2 = (\sigma_y \otimes \sigma_x - \sigma_x \otimes \sigma_y)/2$. Here $\sigma_x, \sigma_y, \sigma_z$ are the three Pauli operators. As we can see, there is still a large gap between Θ_{12} and separable set.

Since the dimension of H grows exponentially with k , to generate these extreme points for Θ_k becomes hard when k increases. In practice, we can generate the extreme points of Θ_k for $k = 12$ and $d_A = d_B = 2$. However, as depicted in Fig. S4, there is still a large gap between the separable boundary and the k -extension boundary k .

More general properties on k -extendability and its relationship to the quantum marginal problem can be found in Refs. [33–38].

C. Generating Random Density Matrices

Since our aim is to determine the separability of generic bipartite states, we require a bunch of random density matrices with full rank to test the performance of our approaches. In our numerical tests, we sample random density matrices under the probability measure $\mu = \nu \times \Delta_\lambda$, where ν is the uniform distribution on $U(n)$ according to Haar measure, Δ_λ is the Dirichlet distribution on the simplex $\sum_{i=1}^n d_i = 1$. The probability density function of Dirichlet distribution is

$$\Delta_\lambda(d_1, \dots, d_n) = C_\lambda \prod_{i=1}^n d_i^{-\lambda},$$

where $\lambda > 0$ is a parameter and C_λ is the normalization constant. Since every density matrix is unitarily similar to a real diagonal density matrix, μ is a probability measure on the set of all density matrices. Such a probability measure is discussed in Ref. [39], section II.A. We implemented the sampling on ν via directly calling the function `RandomUnitary` in [40]. The entire implementation is in the code `RandomState.m` on the website of QMLab [41].

For the 2-qubit case, we set $\lambda = 1/2$ and generate 5×10^4 random quantum states, which are put in the file `2x2rdm.mat`. We have found that 35% of the states are PPT

states, i.e., separable states, which is consistent with the result shown in [39].

For the 2-qutrit case, we also set $\lambda = 1/2$. As shown in [39], only 2.2% of the random states are PPT states when $\lambda = 1/2$. However, our main interest is determining whether a PPT state is entangled. Thus, we reject all the states with negative partial transpose while sampling, and obtain 2×10^4 PPT states eventually. These states are put in the file **3x3rdm.mat** on [41]. We can verify that at least 66.24% of the PPT states are separable under the probability measure we have chosen, using the convex hull approximation, which will be discussed in section E.

We determine whether a state is a PPT state via **IsPPT** in [40].

D. Testing CHA and BCHA

To approximate the set of separable states \mathcal{S} with a convex hull \mathcal{C} , we generate a bunch of extreme points of \mathcal{S} , i.e., random separable pure states in $\mathcal{H}_A \otimes \mathcal{H}_B$ in a straightforward way. The procedure for each time of sampling is demonstrated as follows:

1. Sample a state vector $|\psi_A\rangle \in \mathcal{H}_A \cong \mathbb{C}^{d_A}$ from uniform distribution on the unit hypersphere in \mathbb{C}^{d_A} , according to Haar measure [40].
2. Sample another state vector $|\psi_B\rangle \in \mathcal{H}_B \cong \mathbb{C}^{d_B}$ from uniform distribution on the unit hypersphere in \mathbb{C}^{d_B} , according to Haar measure.
3. Return $|\psi_A\rangle|\psi_B\rangle$.

We execute the above procedure for M times to gain M extreme points $\mathbf{c}_1, \dots, \mathbf{c}_M$. Let

$$\mathcal{C}_m := \text{conv}(\{\mathbf{0}, \dots, \mathbf{c}_m\})$$

for $m = 1, \dots, M$. It is easy to see that $\mathcal{C}_m \subseteq \mathcal{C}_{m+1}$ for $m = 1, \dots, M-1$. Recall that we can decide whether a point \mathbf{p} is in \mathcal{C}_m by solving the following linear programming

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \alpha \mathbf{p} = \sum_{i=0}^m \lambda_i \mathbf{c}_i, \\ & \lambda_i \geq 0, \quad \sum_i \lambda_i = 1. \end{aligned} \quad (4)$$

If $\alpha \geq 1$, \mathbf{p} is in \mathcal{C}_m and thus separable; otherwise, it is possibly an entangled state. The solver for the linear programming 4 is implemented in **CompAlpha.m** on [41].

For the two-qubit case, we sample $M = 10^4$ extreme points, which is saved in the file **2x2extreme.mat** on [41]. We split the data in **2x2rdm.mat** into two, one for training BCHA and the other for testing both CHA and BCHA. To compare the performance of CHA and BCHA, we test the error rate of the CHA \mathcal{C}_m and BCHA based on \mathcal{C}_m on the test

m	1000	2000	3000	4000	5000
error of CHA (%)	8.55	6.01	4.85	4.05	3.60
error of BCHA (%)	3.03	1.97	1.47	1.17	1.15
m	6000	7000	8000	9000	10000
error of CHA (%)	3.25	2.95	2.76	2.64	2.55
error of BCHA (%)	1.01	0.75	0.79	0.71	0.65

TABLE S2. The error rate of CHA \mathcal{C}_m for two-qubit separable set and BCHA based on \mathcal{C}_m for some critical m .

dataset. The result is shown in table S2. We also apply different supervised learning algorithms with the same training and test dataset, without combining CHA. The result is shown in Table S3.

Method	Bagging	Boosting	SVM(rbf)	Decision Tree
Error (%)	12.03	14.8	8.4	23.3

TABLE S3. Error rate of classifiers trained by different algorithms. The error rate is difficult to be reduced due to the lack of prior knowledge.

For the two-qutrit case, we sample $M = 10^5$ extreme points, which is saved in the file **3x3extreme.mat** on [41]. It can be verified that 66.24% of the PPT random states in **3x3rdm.mat** are in the convex hull \mathcal{C}_{10^5} , which implies that at least 66.24% of the PPT random states are separable. We used \mathcal{C}_{10^5} as the criterion for separability, i.e., regard \mathcal{C}_{10^5} as the true separable set. Similar to the two-qubit case, we also tested the accuracy of CHA \mathcal{C}_m as well as BCHA based on \mathcal{C}_m . The result is shown in table S4.

m	10000	20000	30000	40000	50000
error of CHA (%)	33.40	22.69	16.64	12.60	9.63
error of BCHA (%)	12.23	9.54	7.52	6.07	5.03
m	60000	70000	80000	90000	100000
error of CHA (%)	6.86	4.64	2.95	1.39	0
error of BCHA (%)	3.75	2.73	1.81	1.02	0

TABLE S4. The error rate of CHA \mathcal{C}_m for two-qutrit separable set and BCHA based on \mathcal{C}_m for some critical m . Since we used \mathcal{C}_{10^5} as the criterion, the error rate of \mathcal{C}_{10^5} is 0.

E. Iterative Algorithm for Computing the Critical Point

Recall that for an entangled state ρ , there exists a critical point α_ρ such that $\alpha\rho + (1-\alpha)\mathbb{I}/(d_Ad_B)$ ($0 \leq \alpha \leq 1$) is separable when $\alpha \leq \alpha_\rho$ and entangled when $\alpha > \alpha_\rho$. Based on CHA, we developed an iterative algorithm for approximating α_ρ in a more efficient way, which is shown as follows:

1. Randomly sample 1000 extreme points and form a convex hull \mathcal{C} . Let \mathbf{p} be the feature vector of ρ . Set $\epsilon = 1$,

- $\gamma = 0.95$.
2. Update $\alpha_\rho \leftarrow \alpha(\mathcal{C}, \mathbf{p})$.

3. Suppose now $\mathcal{C} := \text{conv}(\{\mathbf{c}_1, \dots, \mathbf{c}_m\})$, and $\alpha_\rho \mathbf{p} = \sum_i \lambda_i \mathbf{c}_i$. Let $\mathbf{c}_{i_1}, \dots, \mathbf{c}_{i_D}$ be the extreme points such that $\lambda_{i_k} > 0$. Set $\mathcal{C} \leftarrow \text{conv}(\{\mathbf{c}_{i_1}, \dots, \mathbf{c}_{i_D}\})$.
4. For each $k = 1, \dots, D$, suppose \mathbf{c}_{i_k} is the feature vector of $|a_k\rangle|b_k\rangle$. We randomly generate two Hermitian operators $H_1 \in \text{End}(\mathcal{H}_A)$, $H_2 \in \text{End}(\mathcal{H}_B)$ such that $\|H_1\|_2 = 1$ and $\|H_2\|_2 = 1$. Let ξ be a random number

in $[0, \epsilon]$. Set $|a'_k\rangle|b'_k\rangle = (e^{i\xi H_1} \otimes e^{i\xi H_2}) |a_k\rangle|b_k\rangle$. Set $\mathcal{C} \leftarrow \text{conv}(\{\mathcal{C}, \mathbf{c}'_k\})$, where \mathbf{c}'_k as the feature vector of $|a'_k\rangle|b'_k\rangle$.

5. Set $\epsilon \leftarrow \gamma\epsilon$ and back to step 2.

What step 4 does is sampling in the neighborhood of \mathbf{c}_{i_k} . In practice, we repeat step 4 for 10 times to get a bunch of neighbors. The detailed implementation is in the code **CriticalPoint.m** [41].