Learning robust and high-precision quantum controls

Re-Bing Wu* and Haijin Ding
Department of Automation, Tsinghua University, Beijing 100084, People's Republic of China
and Center for Quantum Information Science and Technology, BNRist, Beijing 100084, People's Republic of China

Daoyi Dong†
School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia

Xiaoting Wang‡
Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, People's Republic of China

(Received 12 November 2018; published 18 April 2019)

Robust and high-precision quantum control is extremely important but challenging for the functionalization of scalable quantum computation. In this paper, we show that this hard problem can be translated to a supervised machine learning task by thinking of the time-ordered quantum evolution as a layer-ordered neural network (NN). The seeking of robust quantum controls is then equivalent to training a highly *generalizable* NN, to which numerous tuning skills matured in machine learning can be transferred. This opens up a door through which a family of robust control algorithms can be developed. We exemplify such potential by introducing the commonly used trick of batch-based optimization, and the resulting batch-based gradient algorithm is numerically shown to be able to remarkably enhance the control robustness while maintaining high fidelity.

DOI: 10.1103/PhysRevA.99.042327

I. INTRODUCTION

Highly accurate and stable control of quantum hardware is crucial for achieving expected quantum supremacy in the near future [1]. Usually, the control design is easy with respect to a deterministic model. However, finding a single-shot solution that also tolerates the system's uncertainties, e.g., imprecisely identified parameters [2] or time-varying noises in the Hamiltonian [3], is much harder. In the literature, this problem has been tackled from various aspects. Most of these evaluate the control robustness by the geometric curvature of some high-dimensional manifold [4], which can be minimized to enhance the robustness. This point of view leads to various expansion-based methods that have been experimentally very successful, including the adiabatic approach (stimulated Raman adiabatic passage) [5] against control pulse imprecisions, dynamical decoupling [6–10] and difference evolution subspace-selective self-adaptive differential evolution [11,12] algorithms against environmental noises, and other Taylor-expansion-based approaches [13,14].

From a different but more unified point of view, the control against uncertainties can be thought of as manipulating a collection of quantum systems under a uniform control. Along this route, ensemble- and sampling-based approaches [15–17] were proposed to minimize the average error of the entire collection and a subset of samples of it, respectively. These algorithms have been successful in overcoming inhomogeneity of control fields in NMR experiments and are, in principle, applicable to arbitrary types of uncertainties, which can vary with time or not. Compared to the geometric approaches, sampling-based methods are not restricted to the perturbation regime and can thus explore larger uncertainties. However, in practice they are limited to low-dimensional systems or systems with few uncertainty parameters due to the exponentially increasing computational cost. Even if the computation is affordable, the search for robust controls is often hindered by poor solutions due to the loss of controllability over a large collection of sampled control quantum systems.

Our studies follow the latter route for its applicability to nonperturbation regimes and capability of dealing with different types of uncertainties. We find that the search for robust quantum controls can be formulated as a supervised learning task, and the controlled quantum evolution is thought of as a deep neural network (DNN) to be trained for accomplishing the task. More importantly, the pursuit of control robustness can be naturally translated to the training goal of a highly *generalizable* DNN model [18]. This connection provides a new angle for understanding and solving robust quantum control problems enlightened by vast studies in deep learning (DL). For example, the algorithm we present in this paper is illuminated by ways of improving the generalizability of a DNN [19] through the following two aspects:

(1) *Data augmentation.* One always learns better with more samples. Many DL problems have to learn from limited labeled training samples that are hard to obtain (e.g., diagnoses of professional doctors from medical images). However, as

*rbwu@tsinghua.edu.cn
†daoyidong@gmail.com
‡xiaoting@uestc.edu.cn

will be shown later, an unlimited number of training samples is available in our problem setting.

(2) *Minibatch optimization.* In classical big-data applications, a smart way of alleviating the computational burden is, instead of evaluating the loss with all samples, to calculate the loss and gradient functions with randomly selected minibatches of samples that vary from iteration to iteration. In this way, an unlimited number of samples can be explored after sufficiently many iterations. More importantly, the noisy and thus less stable training dynamics can effectively improve the *generalizability* by pulling the search away from the weakly attractive (i.e., poorly generalizable) solutions. This merit has been extensively approved in the practice of DL.

In the following, we will show how the robust quantum control problem is translated to a supervised learning task and how the minibatch training skill is employed to improve the robustness of quantum controls. The rest of this paper is organized as follows. Section II presents a batch-based gradient-descent (b-GRAPE) algorithm, following which Sec. III demonstrates the effectiveness of the b-GRAPE algorithm via two typical examples that involve time-invariant parametric uncertainties and time-varying noises. Finally, conclusions are drawn in Sec. IV.

II. MINIBATCH TRAINING OF ROBUST AND HIGH-PRECISION QUANTUM CONTROLS

In this section, we first show how the robust control design problem can be translated to the seeking of a generalizable learning model, following which the b-GRAPE algorithm is presented by incorporating the minibatch training into the GRAPE optimization process.

A. Robust control design as a supervised learning task

Let us start from the general model of uncertain quantum control systems:

$$\dot{U}(t,\epsilon) = -iH[u(t),\epsilon]U(t,\epsilon), \tag{1}$$

in which the $N \times N$ unitary propagator $U(t,\epsilon)$ is steered from the identity matrix $\mathbb{I}_N$ by the control function $u(t)$. The variable $\epsilon \in \mathbb{R}^k$ denotes the uncertainties in the Hamiltonian that can be some constant but unknown parameters or time-dependent noises (discretized into a vector of uncertainty parameters). We expect to find a robust control $u(t)$ that steers the gate operation $U(T,\epsilon) \in \mathbb{C}^{N\times N}$ to the target gate U_f for all possible ϵ. Since such a goal is usually not achievable, we can approach it by minimizing the average infidelity

$$L[u(t)] = \int_{\mathbb{R}^k} \|U(T,\epsilon) - U_f\|^2 P(\epsilon) d\epsilon, \tag{2}$$

where $P(\epsilon)$ is the *a priori* probability distribution of the uncertainty parameter. To alleviate the computation burden of the integral (2), we approximate the average infidelity over a finite number of uncertainty parameters sampled from $P(\epsilon)$, say, $\mathcal{S} = \{\epsilon_1, \epsilon_2, \dots\}$, as follows:

$$L[u(t),\mathcal{S}] = |S|^{-1} \sum_{\epsilon\in S} \|U(T,\epsilon) - U_f\|^2. \tag{3}$$

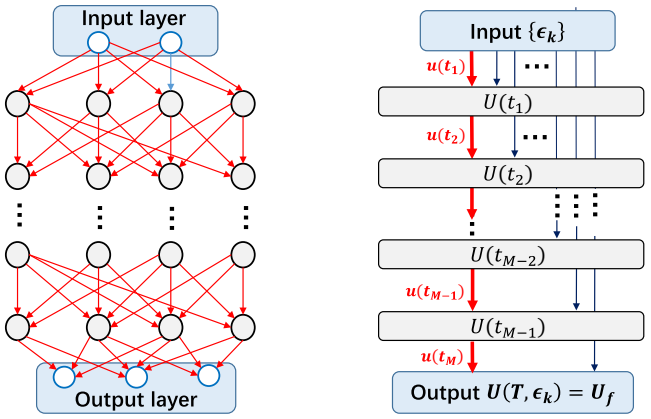

FIG. 1. The similarity between a deep neural network (left) and a quantum system under piecewise-constant controls (right). The quantum system outputs a unitary propagator from a given input ϵ, where the time-ordered control amplitudes $u(t_1), u(t_2), \dots, u(t_M)$ play the role of hyperparameters in a quantum neural network.

The so-called ensemble-based and sample-based algorithms are subject to cost functions (2) and (3), respectively.

The way we improve the robustness through minimizing the average loss of fidelity is actually the same as a supervised machine learning process through minimizing its empirical risk on a set of training samples. As illustrated in Fig. 1, the quantum control system can be envisioned as a linear neural network (NN) that outputs a unitary propagator from an input uncertainty parameter ϵ. Under piecewise-constant controls (e.g., generated from an arbitrary waveform generator), the unitary propagator at each sampling time corresponds to a layer in an NN (with width being N^2), while the time-ordered control amplitudes play the role of weight parameters between adjacent layers. Note that the equivalent quantum neural network does not have a standard feed-forward structure because every layer is affected by the network's input (i.e., the uncertainty parameter ϵ). In this regard, the network is more like a residue neural network [20] in which shortcuts can be made between nonadjacent layers. In this way, we may translate the robust quantum control design to a supervised DL task that aims at finding an NN model that outputs the same desired quantum gate U_f for all input uncertainty samples. It should be noted that this picture is different from recent robust quantum control studies that are also inspired by machine learning [21–24]. These existing works introduce external *artificial* DNNs to the training of robust quantum controls, but we take the controlled quantum system itself as a *natural* quantum DNN.

B. b-GRAPE algorithm

Now let us see how the minibatch technique is applied to the search for robust quantum controls. We can randomly sample the uncertainty parameters ϵ_k, and each pair (ϵ_k, U_f) forms a labeled sample for the supervised learning. In this way, we have access to *uncountably* many labeled samples [i.e., pairs of $\epsilon \to U_f$ for all admissible $\epsilon \in \mathbb{R}^k$] that can be used for free, although they cannot be efficiently exploited

by the existing sampling-based algorithms due to the required computing resources with a large number of samples.

Concretely, we first choose a proper batch size, say B, and draw samples according to the probability distribution $P(\epsilon)$, forming the following batches

$$\mathcal{S}^{(j)} = \{\epsilon_1^{(j)}, \ldots, \epsilon_B^{(j)}\}, \tag{4}$$

where $j = 1, 2, \ldots$ are the indices of iterations. These batches are used for calculating the gradient direction in each iteration:

$$g[u(t), \mathcal{S}^{(j)}] = \frac{\delta L[u(t), \mathcal{S}^{(j)}]}{\delta u(t)} = \frac{1}{B}\sum_{k=1}^{B} \frac{\delta \left\| U\left(T, \epsilon_k^{(j)}\right) - U_f \right\|}{\delta u(t)}. \tag{5}$$

The simplest control-updating strategy is to take the "steepest descending" direction along the stochastic gradient (5) with some prescribed learning rate α_j:

$$u^{(j+1)}(t) = u^{(j)}(t) - \alpha_j g[u^{(j)}(t), \mathcal{S}^{(j)}]. \tag{6}$$

However, the noisy gradient (5) caused by randomly chosen batches may destabilize the steepest-descent iteration, especially when the batch size is very small. To stabilize the training dynamics, one can introduce a momentum term, i.e., the gradient direction in the previous iteration, to reduce the variance of the loss function:

$$u^{(j+1)}(t) = u^{(j)}(t) - \alpha_j\{\lambda g[u^{(j)}(t), \mathcal{S}^{(j)}] + (1-\lambda) g[u^{(j-1)}(t), \mathcal{S}^{(j-1)}]\}. \tag{7}$$

In practice, the weight parameter λ is usually chosen to be a small, positive real number (e.g., 0.1 or 0.01), so that the iteration is dominated by the momentum.

For simplicity, we term the proposed algorithm b-GRAPE (where the b stands for "batch"), which can easily be remolded from the renowned GRAPE algorithm [25] that has been extensively applied for quantum control. Correspondingly, we denote the sampling-based algorithm as s-GRAPE for (where s is for "sampling") [17]. The s-GRAPE algorithm is actually a special case of the b-GRAPE algorithm when using a fixed batch in all iterations, while GRAPE (for deterministic quantum systems) is a special case of s-GRAPE when only one sample is used.

Note that our algorithm is distinct from the simultaneous perturbation stochastic algorithm (SPSA), another type of stochastic gradient algorithm which calculates the gradient by randomizing the projected direction instead of the samples. The latter was proposed for online model-free learning of robust quantum control and tomography [26], and it can be combined with b-GRAPE for broader applications.

III. NUMERICAL SIMULATIONS

In this section, we show by two simulation examples how the DL-illuminated b-GRAPE algorithm can effectively harden quantum controls by learning from the uncertainties.

A. Example 1: Parametric uncertainty

The first example considers time-invariant parametric uncertainties in a three-qubit control system:

$$H(t) = (1+\epsilon_1)\sigma_{1z}\sigma_{2z} + (1+\epsilon_2)\sigma_{2z}\sigma_{3z} + \sum_{k=1}^{3}[u_{kx}(t)\sigma_{kx} + u_{ky}(t)\sigma_{ky}], \tag{8}$$

where $\sigma_{k\alpha}$, with $k = 1, 2, 3$ and $\alpha = x, y, z$, are the Pauli operators on the kth qubit. The uncertainty parameters ϵ_1 and ϵ_2 represent the identification errors in the coupling constants (dimensionless after normalization). Each qubit is manipulated by two independent control fields, $u_{kx}(t)$ (along the x axis) and $u_{ky}(t)$ (along the y axis). The target three-qubit gate U_f is chosen as the Toffoli gate (or controlled-controlled-NOT gate).

To start with, we set the time duration as $T = 10$ and divide each control field evenly into $M = 100$ piecewise constant segments. Assuming that the uncertain coupling constants vary by at most $\pm 20\%$, we uniformly sample ϵ_1 and ϵ_2 from the set $\mathcal{S} = \{(\epsilon_1, \epsilon_2) : |\epsilon_1| \leqslant 0.2, |\epsilon_2| \leqslant 0.2\}$. The b-GRAPE algorithm is tested under three typical batch sizes, $B = 1$, 10, and 100, and is compared with s-GRAPE under identical batch sizes and initial guesses on the control. Because the training process is stabler under large batches, the learning rates are correspondingly chosen as $\alpha = 0.002$, 0.02, and 0.2, respectively, to be proportional to the batch size.

In the simulations, we optimize the control fields along the momentum-based stochastic gradient (7). The resulting training curves, namely, the average infidelity evaluated on each batch versus the number of evaluated samples (equal to the batch size times the number of iterations), are shown in Fig. 2 for both b-GRAPE and s-GRAPE algorithms. The batch-induced noises can be seen in all b-GRAPE training curves, whose variance is large when using small batches. Nevertheless, an evident trend of decrease is still observable.

Because the average infidelity calculated with small batches may not reflect the actual performance, we reevaluate the performance of the control functions obtained in each iteration by 1000 independent testing samples drawn from the same probability distribution as a better approximation of the true average infidelity (2). For s-GRAPE with batch size $B = 1$, the training error can approach the computer machine precision, which is far below the range displayed in the plot, but the testing performance is very poor ($\approx$0.1). Such an *overfitting* characteristic is more clearly indicated by the gap between the testing curve (above) and the training curve (below) when $B = 10$. In contrast, the testing curves of b-GRAPE always fit (on average) very well to the training curves, exhibiting much better generalizability owing to the ability of exploring many more samples.

The most significant difference, as can be seen with all tested batch sizes, is that b-GRAPE finds much more robust controls than s-GRAPE. For the example of $B = 100$ [see Fig. 2(c)], the generalization gap of s-GRAPE is almost invisible, implying that the batch size has been sufficiently large to avoid overfitting. However, b-GRAPE still performs much better than s-GRAPE, owing to the batch-induced noises that steer the search away from poorer solutions. The best result

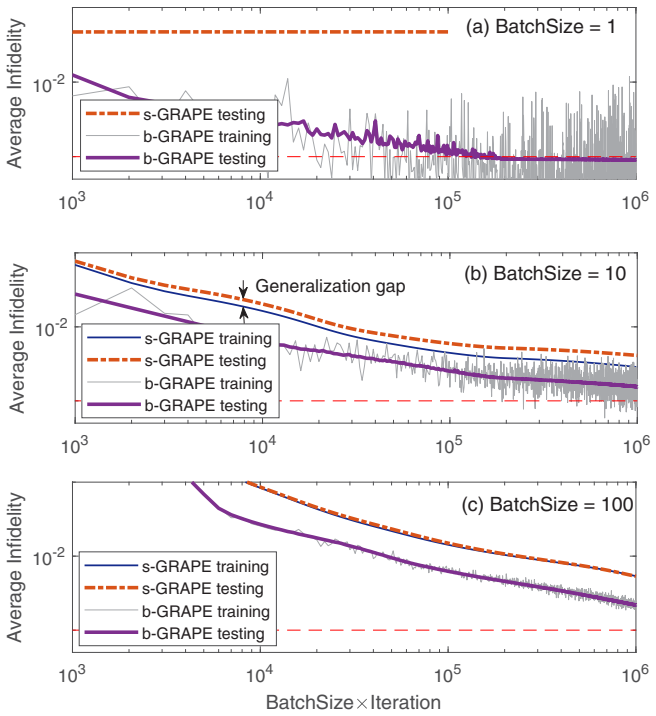

FIG. 2. The training and testing curves with b-GRAPE and s-GRAPE algorithms, where the control time is $T = 10$, the number of control segments is $M = 100$, and the batch sizes are (a) $B = 1$, (b) $B = 10$, and (c) $B = 100$. In all cases, the batch-based b-GRAPE algorithm outperforms the sample-based s-GRAPE algorithm.

in all simulations is achieved when using the smallest batch size $B = 1$ [see Fig. 2(a)], under which the average infidelity can be reduced to be below 0.001 (lower than the error correction threshold). The contrast again testifies to the active role of batch-induced noises, which is strongest when $B = 1$, in guiding the search toward more robust solutions.

To manifest the degree of robustness enhancement, we compare control fields obtained by the GRAPE (fixed sample $\epsilon_1 = \epsilon_2 = 0$), s-GRAPE ($B = 100$, fixed and large batch), and b-GRAPE ($B = 1$, random minibatch) algorithms via their three-dimensional robustness landscapes (i.e., the infidelity versus the two uncertainty parameters; see Fig. 3). The landscape also facilitates the quantification of control robustness, which can be evaluated by the area enclosed by the level set at some threshold value (say, 0.001 in Fig. 3, which is below the quantum error correction threshold [27]). The control obtained by GRAPE achieves extremely high precision at the chosen sample $\epsilon_1 = \epsilon_2 = 0$, but it is very sensitive to the uncertainty, as indicated by the sharp minimum. By contrast, the landscapes corresponding to the s-GRAPE and b-GRAPE algorithms are much flatter. The obtained controls maintain high precision in a much broader region, at the price of sacrificing the precision at the center. Quantitatively, s-GRAPE enhances the robustness by about 4 times, and the control found by b-GRAPE is more than 10 times stronger than that of s-GRAPE. The level set at 0.001 achieved by b-GRAPE almost fills the full 0.2×0.2 square from which the training samples are drawn.

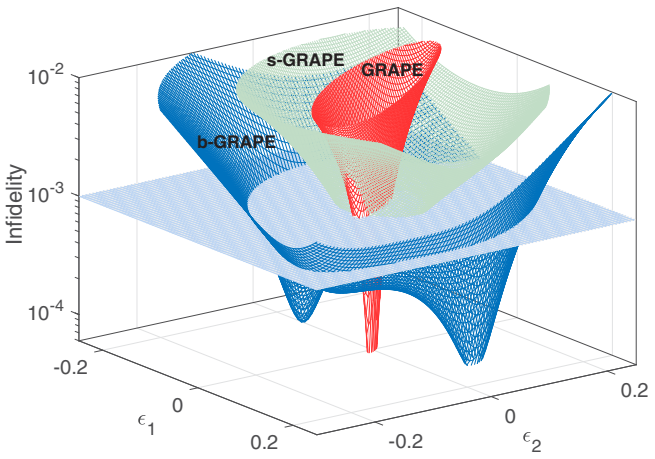

FIG. 3. The robustness control landscape defined as the gate infidelity versus two uncertainty parameters under controls optimized with GRAPE (red), s-GRAPE (green) using 100 samples, and b-GRAPE with batch size $B = 1$ (blue). The b-GRAPE obtains a much more robust control than the other two schemes, where the robustness is quantified by the area enclosed by the level set at 0.001.

We also test the performance of b-GRAPE with less available control resources, i.e., using a shorter time duration ($T = 5$) and fewer ($M = 50$) control segments. The simulations are all stopped after evaluating 1×10^6 samples. The simulations consistently prove the superiority of b-GRAPE (bold black lines) over s-GRAPE (dotted red lines) in all cases shown in Fig. 4. However, not surprisingly, the robustness is less enhanced when the control is more limited. In fact, b-GRAPE should be more advantageous under such circumstances because otherwise the search will be more easily trapped by local optima.

The simulations also include the exceptional example shown in Fig. 4(b1), whose robustness achieved by b-GRAPE is supposed to be stronger than those in Fig. 4(a1), where the control time is shorter, and Figs. 4(b2) and 4(b3), where the batch size is larger. This poor solution results from an instable training process, during which the average infidelity over training batches rises up after exploring about 800 000 samples. The level set at 0.001 (in blue), corresponding to the best solution before losing stability, is also depicted in Fig. 4(b1), which is disconnected due to the coexistence of two minima like in Fig. 3. Note that the stochastic training process is not always instable, and very robust solutions can be obtained after restarting the stochastic b-GRAPE optimization and reselecting the initial guess or decreasing the learning rates.

B. Example 2: Time-varying noise

To demonstrate the applicability of the b-GRAPE algorithm to more general uncertainties, we consider the following single-qubit system that contains time-varying noises:

$$H(t) = [1 + n(t)][u_x(t)\sigma_x + u_y(t)\sigma_y], \tag{9}$$

where $\sigma_{x,y}$ are the Pauli matrices, with $u_{x,y}(t)$ being the Rabi driving fields. The noise $n(t)$ represents the multiplicative time-varying noises in the control amplitudes.

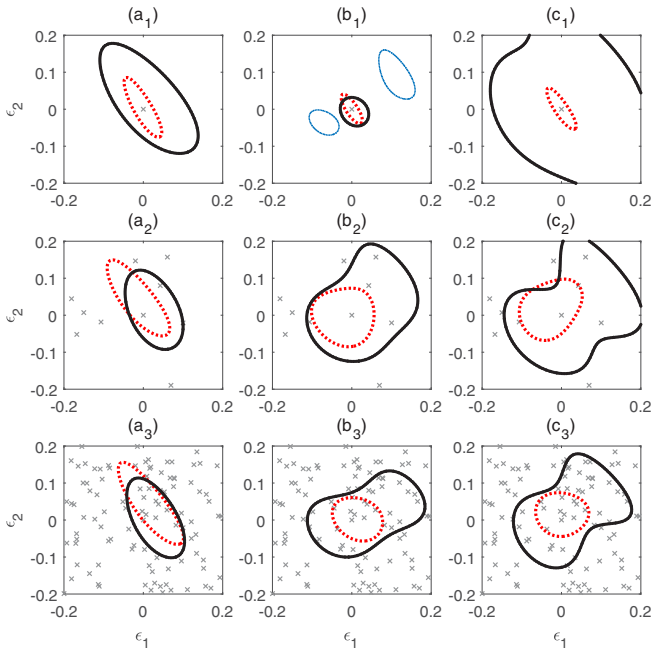

FIG. 4. The robustness (quantified by the area enclosed by the level set at 0.001) of control fields obtained by b-GRAPE (black solid lines) and s-GRAPE (red dotted lines) algorithms. The time duration T and number of control segments M are, respectively, (a1)–(a3) $T = 5$ and $M = 50$, (b1)–(b3) $T = 10$ and $M = 50$, and (c1)–(c3) $T = 10$ and $M = 100$. The corresponding batch sizes are $B = 1$ (first row), $B = 10$ (second row), and $B = 100$ (third row).

Since the qubit is insensitive to high-frequency noises, we sample only the low-frequency noises as follows:

$$n(t) = \sum_{k=1}^{10} (a_k \cos \omega_k t + b_k \sin \omega_k t), \tag{10}$$

where the frequency components ω_k are uniformly sampled from 0 to 2π rad/s and the amplitudes a_k and b_k are sampled from a Gaussian distribution, $\mu(0, 0.05)$. These parameters form a 30-dimensional sample space, which the sampling-based algorithms can hardly handle because a formidably large number samples will be required.

In the simulation, the target unitary transformation is chosen as the qubit flip, i.e., a π rotation $U_f = R_x(\pi)$ around the x axis. In the absence of noise, the rotation can easily be achieved by applying an arbitrary $u_x(t)$ whose pulse area is π [with $u_y(t)$ being turned off], e.g., rectangular or Gaussian. However, robustness is not guaranteed for these pulses.

We set the simulation time as $T = 2$ and bound the control fields by $|u_{x,y}(t)| \leqslant \pi$. The batch size is chosen to be $B = 10$, and in total 10 000 iterations are performed, after which the average fidelity is reduced to below 10^{-2}. The wave forms of the initial guess and the optimized field are shown in Fig. 5.

To see how the robustness is enhanced, we test the obtained control field by analyzing the statistical distribution of the gate errors using 10 000 random noise samples picked from the same distribution. As shown in Fig. 6, the gate error under the optimized control exhibits a typical Gaussian distribution whose center is below 10^{-3}. We also depict the cumulative

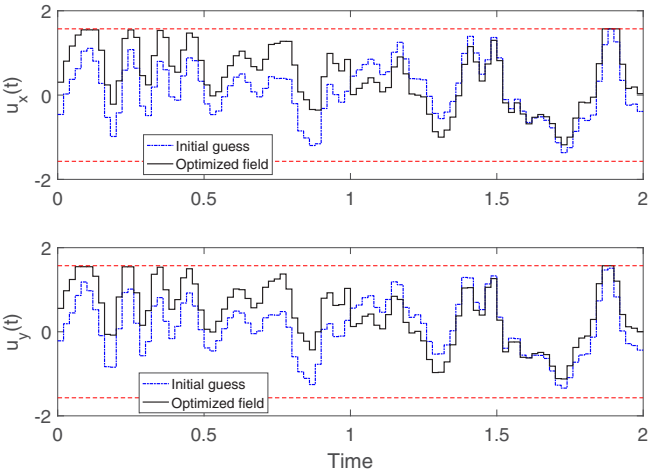

FIG. 5. The initial guess and optimized wave forms of the x-axis and y-axis control fields.

probability distribution function, from which it can clearly be seen that the probability for the error to be below 10^{-2} is almost 100%, and the probability for the error to be below 10^{-3} is about 76%. We also evaluate the robustness of the standard rectangular and Gaussian π pulses, whose probabilities for the error to be below 10^{-2} are only 62% and 43%, respectively. Apparently, the optimized control field is much more robust to the time-varying noises.

IV. CONCLUSION

To conclude, we proposed b-GRAPE, a deep-learning-illuminated algorithm, for efficiently discovering highly robust quantum controls in the high-precision regime. The algorithm can easily be implemented by randomizing the renowned GRAPE algorithm with batches of samples, and numerical simulations demonstrate its effectiveness owing to the endowed ability of exploring uncountably many uncertainty samples and the ability to escape poor optima driven by the batch-induced randomness. Our algorithm can also be conveniently paralleled. Although the theoretically best performance is achieved when $B = 1$, in practice we can

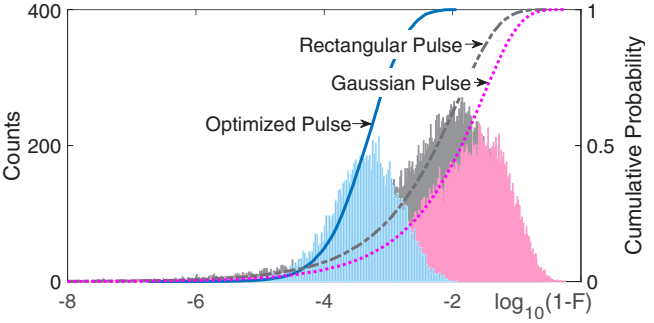

FIG. 6. Error distribution of single-qubit quantum controls counted from 10 000 random samples of multiplicative noises in the amplitudes, where the target is the qubit-flip transformation. The field optimized by b-GRAPE is very strong in that the probability for the error to be below 10^{-2} is almost 100%, which is much higher than those of rectangular and Gaussian π pulses.

adequately increase the batch size to improve the computational efficiency as well as the algorithmic stability.

In our numerical tests, no evident traps (i.e., local optima that are far from global optima) were encountered. This implies a nice control landscape that has been observed in both quantum control [28–30] and deep learning [31,32] studies. These results are closely related to the control landscape of uncertain quantum systems, and a further study on this topic will be very important to a full understanding of robust quantum control problems.

This work is only the start of a potentially large family of robust control design algorithms. As can be seen in our simulations, there is still much room for the control robustness to be enhanced, e.g., by fine-tuning the learning rates, increasing the number of iterations, and introducing more advanced tuning strategies. More deep neural network tuning skills developed in the deep learning practice [19] can be easily transferred here, [33] as well as gradient-free algorithms (e.g., genetic or differential evolution algorithms). We expect that such DL-inspired algorithms will produce signification impacts on the design of high-quality controls over quantum information processing hardware.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China (Grants No. 2018YFA0306703 and No. 2017YFA0304304), NSFC (Grants No. 61833010, No. 61828303, and No. 61773232) and the Australian Research Councils Discovery Projects funding scheme under Project No. DP190101566.

[1] A. W. Harrow and A. Montanaro, Nature (London) **549**, 203 (2017).
[2] M. Dahleh, A. P. Peirce, and H. Rabitz, Phys. Rev. A **42**, 1065 (1990).
[3] M. Demiralp and H. Rabitz, Phys. Rev. A **57**, 2420 (1998).
[4] D. Hocker, C. Brif, M. D. Grace, A. Donovan, T.-S. Ho, K. M. Tibbetts, R. Wu, and H. Rabitz, Phys. Rev. A **90**, 062309 (2014).
[5] N. V. Vitanov, A. A. Rangelov, B. W. Shore, and K. Bergmann, Rev. Mod. Phys. **89**, 015006 (2017).
[6] L. Viola, E. Knill, and S. Lloyd, Phys. Rev. Lett. **82**, 2417 (1999).
[7] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, Nature (London) **458**, 996 (2009).
[8] A. M. Souza, G. A. Alvarez, and D. Suter, Phys. Rev. Lett. **106**, 240501 (2011).
[9] T. J. Green, J. Sastrawan, H. Uys, and M. J. Biercuk, New J. Phys. **15**, 095004 (2013).
[10] L. F. Santos and L. Viola, New J. Phys. **10**, 083009 (2008).
[11] E. Zahedinejad, J. Ghosh, and B. C. Sanders, Phys. Rev. Lett. **114**, 200502 (2015).
[12] E. Zahedinejad, J. Ghosh, and B. C. Sanders, Phys. Rev. Appl. **6**, 54005 (2016).
[13] D. Daems, A. Ruschhaupt, D. Sugny, and S. Guérin, Phys. Rev. Lett. **111**, 050404 (2013).
[14] C.-H. Huang and H.-S. Goan, Phys. Rev. A **95**, 062325 (2017).
[15] J. S. Li and N. Khaneja, Phys. Rev. A **73**, 030302(R) (2006).
[16] J. Ruths and J.-S. Li, IEEE Trans. Autom. Control **57**, 2021 (2012).
[17] C. Chen, D. Dong, R. Long, I. R. Petersen, and H. A. Rabitz, Phys. Rev. A **89**, 023402 (2014).
[18] H. Xu and S. Mannor, Mach. Learn. **86**, 391 (2012).
[19] L. Bottou, F. E. Curtis, and J. Nocedal, SIAM Rev. **60**, 223 (2018).
[20] K. He, X. Zhang, S. Ren, and J. Sun, arXiv:1512.03385.
[21] X.-C. Yang, M.-H. Yung, and X. Wang, Phys. Rev. A **97**, 042324 (2018).
[22] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Phys. Rev. X **8**, 031086 (2018).
[23] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, Quantum Sci. Technol. **4**, 024004 (2019).
[24] M. Yuezhen Niu, S. Boixo, V. Smelyanskiy, and H. Neven, arXiv:1803.01857.
[25] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbruggen, and S. J. Glaser, J. Magn. Reson. **172**, 296 (2005).
[26] C. Ferrie and O. Moussa, Phys. Rev. A **91**, 052306 (2015).
[27] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Phys. Rev. A **54**, 3824 (1996).
[28] H. A. Rabitz, M. M. Hsieh, and C. M. Rosenthal, Science **303**, 1998 (2004).
[29] H. Rabitz, M. Hsieh, and C. Rosenthal, Phys. Rev. A **72**, 052337 (2005).
[30] R. B. Wu, H. Rabitz, and M. Hsieh, J. Phys. A **41**, 015006 (2008).
[31] K. Kawaguchi, *30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016), pp. 586–594.
[32] P. Zhou and J. Feng, arXiv:1705.07038.
[33] M. Pilanci and M. Wainwright, SIAM J. Optim. **27**, 205 (2017).