

Learning the black hole metric from holographic conductivity

Kai Li,^{1,2,3,*} Yi Ling,^{2,3,†} Peng Liu,^{4,‡} and Meng-He Wu^{5,6,§}

¹*Kuang Yaming Honors School, Nanjing University, Nanjing 210023, China*

²*Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China*

³*School of Physics, University of Chinese Academy of Sciences, Beijing 100049, China*

⁴*Department of Physics and Siyuan Laboratory, Jinan University, Guangzhou 510632, China*

⁵*School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, Shanghai 201620, China*

⁶*Center of Application and Research of Computational Physics, Shanghai University of Engineering Science, Shanghai 201620, China*



(Received 1 December 2022; accepted 6 March 2023; published 22 March 2023)

We construct a neural network to learn the Reissner-Nordström-anti-de Sitter black hole metric based on the data of optical conductivity by holography. The linear perturbative equation for the Maxwell field is rewritten in terms of the optical conductivity such that the neural network is constructed based on the discretization of this differential equation. In contrast to all previous models in anti-de Sitter/deep learning duality, the derivative of the metric function appears in the equation of motion and we propose distinct finite difference methods to discretize this function. The notion of the reduced conductivity is also proposed to avoid the divergence of the optical conductivity near the horizon. The dependence of the training outcomes on the location of the cutoff, the temperature as well as the frequency range is investigated in detail. This work provides a concrete example for the reconstruction of the bulk geometry with the given data on the boundary by deep learning.

DOI: [10.1103/PhysRevD.107.066021](https://doi.org/10.1103/PhysRevD.107.066021)

I. INTRODUCTION

AdS/CFT correspondence as a typical implementation of gauge/gravity duality reveals the deep connections between a $(d+1)$ -dimensional gravity theory and a d -dimensional quantum field theory [1–4]. In particular, due to the feature of strong/weak duality in a large N limit, AdS/CFT correspondence has been proven to be a powerful tool to study the strongly correlated physics via classical gravitational theories. When applying many-body systems in condensed matter physics, it has formed an important subject that now is dubbed as AdS/CMT (condensed matter theory) duality [5–11]. In this field, the traditional way is properly setting the structure of bulk geometry, and then deriving the properties of the many-body system living on the boundary by solving the equations of motion in the bulk with the use of the holographic dictionary. Surprisingly,

one finds some of the properties simulate the transport behavior of a strongly coupled system which has been observed in laboratory but is very hard to understand based on the standard perturbative method in quantum field theory [6,12–22]. In this direction, remarkable progress has been made in understanding the longstanding problems in condensed matter physics by holography, such as the mechanism of high-temperature superconductivity and the non-Fermi liquid behavior of strange metals [23,24]. Nevertheless, this route contains a vital limitation that prevents us from thoroughly solving the problems faced by experimental physicists in laboratory. That is, the derived properties of the dual system greatly depend on the specific structure of spacetime in the bulk. Once the setup of bulk geometry is given, then the transport property of the dual system on the boundary is determined. Usually, based on the consideration of symmetry, one may construct the bulk geometry with essential ingredients to observe the expected phenomenon for a dual system, but this is not the general case. In general, we do not know what kind of bulk geometry would give rise to the specific property of the boundary theory as one expects. For instance, there are two fundamental problems which have not been solved by AdS/CMT duality. One is to reproduce the expected phase diagram for the high-temperature superconductivity, the other is to reproduce all the observed features of strange metal in a single holographic model. In this situation, we

*lik@ihep.ac.cn

†lingy@ihep.ac.cn

‡phylp@email.jnu.edu.cn

§mhwu@sues.edu.cn

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

are facing the inverse problem of the traditional AdS/CMT method: Given the observed features of practical materials in lab, can we construct a holographic model to reproduce such features properly and finally provide a theoretical understanding on these features? Or more generally, in the context of holography, given the data on the boundary, how can we reconstruct the geometry of the bulk?

Without doubt, this inverse problem is much harder. A lot of efforts have been made to learn the bulk geometry by machine learning/deep learning (DL), which is now viewed as the most advanced technique in artificial intelligence [25–47]. As far as we know, one may classify the current application of deep learning to holography into two categories by the types of boundary data used. One is taking the entanglement on the boundary as the input data [34,35] and the other is taking the vacuum expectation value as the data [36,37,39,42]. For the first category, one usually constructs a tensor network as the discretized version of AdS/CFT correspondence, and then transfers this network to a Boltzmann machine, and a method called entanglement feature learning is suggested to learn the spatial geometry from the feature of entanglement on the boundary [34]. A generic neural network is also proposed to recover the geometry fluctuation from the multiregion entanglement entropy on the boundary [35]. The second category is less ambitious but more relevant to AdS/CMT duality. One just specifies the metric of the bulk to be some simple form with one or more unknown functions, and then constructs a neural network based on the equations of motion for matter fields. The goal of the neural network is to learn unknown functions in the metric by boundary data which are the vacuum expectation value of dual operators on the boundary. Now such an approach is called the anti-de Sitter (AdS)/DL method. As the first step, in [36], it is assumed that only one single function in the spacetime metric is unknown. With this ansatz, one attempted to construct the neural network based on the equations of motion of a scalar field and then train the neural network to learn the corresponding spacetime metric by inputting the experimental data of magnetization and external magnetic field as initial data. Moreover, this AdS/DL method has also been applied to AdS/QCD duality [37,40,41]. Later, the neural network is replaced by another machine learning algorithm called neural ordinary differential equation, which can serve the same purpose meanwhile yielding more accurate results [43].

In all above work, the neural network is constructed by considering the perturbations of a scalar field. In [42], the perturbation of the metric tensor has been considered and the neural network reflects the renormalization group (RG) flow equation of the shear viscosity. In this paper, aiming to apply AdS/DL method to AdS/CMT duality, we intend to extend the setup to investigate the perturbations of a vector field in the bulk. Specifically, we consider the electromagnetic field A_μ in a charged black hole background and

construct the neural network based on the RG flow equation of the optical conductivity of the dual current operator, then train the black hole metric from the data of the optical conductivity on the boundary.

We organize this paper as follows. In Sec. II, we derive the equation of motion for the linear perturbation of the Maxwell field and then rewrite this equation with the optical conductivity as the fundamental variable. The neural network is constructed based on the discretized version of this equation. In Sec. III, we explain the preparation of the input and output training data. In Sec. IV, we illustrate the results of deep learning and discuss the effects of the chemical potential μ and the region of frequency ω on the final training results. We suggest a novel regularization term to improve training accuracy and save time in hyperparameter tuning. We also propose several conditions on the regularization term, such as smooth metric and asymptotic AdS, to obtain physically reasonable results. Conclusions and discussions are given in Sec. V. Appendix A gives details of four kinds of finite difference methods for $f'(z)$. Appendix B gives the detailed training methods, hyperparameters, and all the training results. Appendix C states the running environment of the code.

II. BUILDING DEEP NEURAL NETWORK BY HOLOGRAPHIC CONDUCTIVITY

In this section, we derive the equation of motion for the optical conductivity in AdS/CMT duality and then construct the corresponding neural network for training the black hole metric. We start with the action of Einstein-Maxwell theory with a negative cosmological constant,

$$S = \frac{1}{2\kappa^2} \int d^4x \sqrt{-g} \left(R + \frac{6}{L^2} - \frac{F^{ab}F_{ab}}{4} \right), \quad (1)$$

where $\kappa^2 = 8\pi G$ with G the Newton constant, and $\frac{6}{L^2}$ is the cosmological constant term with L being the AdS radius. The field strength is $F = dA$, where A is the Maxwell field. From this action, the equations of motion can be derived as

$$R_{ab} - \frac{1}{2}g_{ab}R - \frac{3g_{ab}}{L^2} - \left(F_{ac}F_b{}^c - \frac{1}{4}g_{ab}F^2 \right) = 0, \\ \nabla^a F_{ab} = 0. \quad (2)$$

We consider the following charged black brane solution to (2) with spatially planar symmetry which is also called the AdS-RN metric,¹

$$ds^2 = \frac{1}{z^2} \left[-f(z)dt^2 + \frac{dz^2}{f(z)} + dx^2 + dy^2 \right], \\ A = \mu(1-z)dt, \quad (3)$$

¹For concreteness, we fix $\kappa = 1$ and $L = 1$ throughout this paper.

where $z \in [0, 1]$ is the radial direction of RG flow and $f(z) \equiv 1 - z^3 - \mu^2 z^3/4 + \mu^2 z^4/4$ is the theoretical result by solving Einstein equations. In this paper, we will treat $f(z)$ as the target function that should be learned by the neural network via data training. Parameter μ is the chemical potential of the dual system on the boundary. In this metric form, the horizon of black brane locates at $z = 1$ while the boundary of spacetime locates at $z = 0$. Moreover, the Hawking temperature, which is given by $T = \frac{12-\mu^2}{16\pi}$, is identified as the temperature of the dual system in equilibrium. Here, we set $\mu \in (0, \sqrt{12}]$ to ensure that the Hawking temperature is positive. Thus, one may change the temperature of the system by adjusting the value of μ .

Now we consider the optical conductivity of the system following the standard procedure of the linear response theory in holographic gravity [7,48]. To obtain the optical conductivity, we turn on an electric field along x direction by considering the following linear perturbation in the bulk,

$$\delta A_x = A_x(z)e^{-i\omega t}. \quad (4)$$

Plugging it into the Maxwell equation, one obtains the linearized equation of motion for A_x as

$$z^4 A_x''(z) + 2z^3 A_x'(z) - z^2 A_x'(z) B(z) + C(z) A_x = 0, \quad (5)$$

where

$$\begin{aligned} B(z) &= -\frac{z^2 f'(z)}{f(z)} + 2z, \\ C(z) &= \frac{z^4 \omega^2}{f^2(z)} - \frac{\mu^2 z^6}{f(z)}. \end{aligned} \quad (6)$$

The Green function $\langle J_x J_x \rangle$ can be collected as $-\partial_z A_x / A_x$ from holographic dictionary, while the applied electric field associated with A_x reads as $E_x = -\partial_t A_x = i\omega A_x$. From the

Kubo formula in holographic gravity [6,49], the optical conductivity can be expressed as

$$\sigma(z, \omega) = \frac{\partial_z A_x(z)}{i\omega A_x(z)}. \quad (7)$$

Next, we intend to rewrite Eq. (5) as the differential equation in terms of the optical conductivity. For this purpose, we notice that

$$\partial_z \sigma = \frac{iA_x'(z)^2}{\omega A_x(z)^2} - \frac{iA_x''(z)}{\omega A_x(z)}. \quad (8)$$

Furthermore, dividing (5) by $A_x(z)$ one obtains

$$z^4 A_x''(z)/A_x + 2z^3 A_x'(z)/A_x - z^2 B(z) A_x'(z)/A_x + C(z) = 0. \quad (9)$$

With the use of Eq. (8), we obtain the differential equation for the optical conductivity as

$$z^4 (i\omega \sigma'(z) - \omega^2 \sigma^2) + i\omega \sigma (2z^3 - z^2 B(z)) + C(z) = 0. \quad (10)$$

Now the original second-order differential equation with variable $A_x(z)$ becomes a first-order differential equation with variable $\sigma(z)$. We are ready to construct the neural network and train the metric function $f(z)$ based on this equation. First, we discretize Eq. (10) by evenly sampling along the z axis,

$$\Delta z = \frac{z_h - z_b}{N-1}, \quad z(n) = z_b + n\Delta z, \quad (11)$$

where N is the number of layers of the network, while z_b and z_h are the locations of the cutoff on the boundary and the horizon, respectively. In addition, $n \in [0, N-1]$, $n \in \mathbf{Z}$, $N \in \mathbf{N}^*$, such as $z(0) = z_b$ and $z(N-1) = z_h$.

We rewrite Eq. (10) into the real part and the imaginary part separately as

$$\begin{aligned} \text{Re}\sigma(z + \Delta z) &= \text{Re}\sigma(z) + \Delta z \left[-\frac{f'(z)}{f(z)} \text{Re}\sigma(z) + 2\omega \text{Im}\sigma(z) \text{Re}\sigma(z) \right], \\ \text{Im}\sigma(z + \Delta z) &= \text{Im}\sigma(z) + \Delta z \left[-\frac{f'(z)}{f(z)} \text{Im}\sigma(z) + \frac{\omega}{f^2(z)} - \frac{\mu^2 z^2}{\omega f(z)} + \omega (\text{Im}\sigma(z))^2 - \omega (\text{Re}\sigma(z))^2 \right]. \end{aligned} \quad (12)$$

Next, we are concerned with the boundary conditions on both ends of the z axis. At the horizon, we impose the ingoing boundary condition for $A_x(z)$, which takes the form as

$$A_x(z) = (1-z)^{-\frac{i\omega}{4\pi T}} a_x(z). \quad (13)$$

As a result, the optical conductivity becomes

$$\sigma(z, \omega) = \frac{1}{4\pi T(1-z)} + \frac{\partial_z a_x(z)}{i\omega a_x(z)} \equiv \frac{1}{4\pi T(1-z)} + \sigma_r(z, \omega), \quad (14)$$

where we have denoted the second term as $\sigma_r(z, \omega)$, and we name it the reduced optical conductivity. It is noticed that with the ingoing boundary condition, the first term of the

conductivity becomes divergent near the horizon, which easily sabotages the numerics. Therefore, we intend to treat the reduced optical conductivity as the basic variable for the construction of a deep neural network. The discretized version of the equations of motion for the reduced conductivity is given by

$$\begin{aligned}
 \text{Re}\sigma_r(z + \Delta z) &= \text{Re}\sigma_r(z) + \Delta z \left[-\frac{f'(z)}{f(z)} \left(\text{Re}\sigma_r(z) + \frac{1}{4\pi T(1-z)} \right) + 2\omega \text{Im}\sigma_r(z) \text{Re}\sigma_r(z) \right. \\
 &\quad \left. + \frac{2\omega}{4\pi T(1-z)} \text{Im}\sigma_r(z) - \frac{1}{4\pi T(1-z)^2} \right], \\
 \text{Im}\sigma_r(z + \Delta z) &= \text{Im}\sigma_r(z) + \Delta z \left[-\frac{f'(z)}{f(z)} \text{Im}\sigma_r(z) - \frac{\omega}{(4\pi T)^2(1-z)^2} - \frac{2\omega}{4\pi T(1-z)} \text{Re}\sigma_r(z) \right. \\
 &\quad \left. + \frac{\omega}{f^2(z)} - \frac{\mu^2 z^2}{\omega f(z)} + \omega (\text{Im}\sigma_r(z))^2 - \omega (\text{Re}\sigma_r(z))^2 \right].
 \end{aligned} \tag{15}$$

In order to reveal the relation between the above discretized equation and a neural network, we convert the equation into a matrix form as below,

$$\begin{pmatrix} \text{Re}\sigma_r(z + \Delta z) \\ \text{Im}\sigma_r(z + \Delta z) \end{pmatrix} = \mathbf{W}_{2 \times 2} \begin{pmatrix} \text{Re}\sigma_r(z) \\ \text{Im}\sigma_r(z) \end{pmatrix} + \vec{\mathbf{b}}, \tag{16}$$

where

$$\mathbf{W}_{2 \times 2} = \begin{pmatrix} 1 - \Delta z \frac{f'(z)}{f(z)} & \Delta z \frac{2\omega}{4\pi T(1-z)} \\ -\Delta z \frac{2\omega}{4\pi T(1-z)} & 1 - \Delta z \frac{f'(z)}{f(z)} \end{pmatrix} \tag{17}$$

and

$$\vec{\mathbf{b}} = \Delta z \begin{pmatrix} -\frac{f'(z)}{f(z)} \frac{1}{4\pi T(1-z)} - \frac{1}{4\pi T(1-z)^2} + 2\omega \text{Im}\sigma_r(z) \text{Re}\sigma_r(z) \\ -\frac{\omega}{(4\pi T)^2(1-z)^2} + \frac{\omega}{f^2(z)} - \frac{\mu^2 z^2}{\omega f(z)} + \omega (\text{Im}\sigma_r(z))^2 - \omega (\text{Re}\sigma_r(z))^2 \end{pmatrix}. \tag{18}$$

Here, $\mathbf{W}_{2 \times 2}$, which contains the information of the spacetime metric, can be regarded as the weight matrix of a neural network. The weight matrix in deep learning represents the connecting parameters of the neurons of adjacent layers. And naturally, $\vec{\mathbf{b}}$ can be regarded as the bias

term of a network. In addition, according to the matrix form, the activation function is the identical mapping.

As a result, we construct the following neural network to represent the discretized equation of motion for the reduced conductivity (Fig. 1).

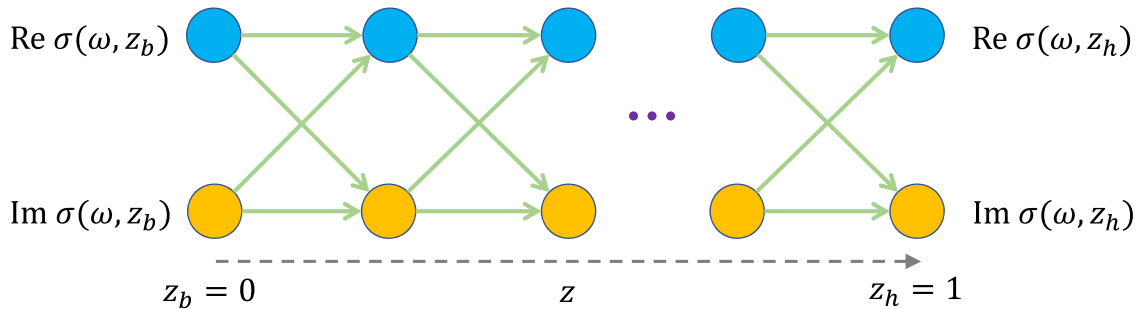


FIG. 1. The structure of the deep neural network. The holographic optical conductivity propagates from the boundary to the horizon.

Physically speaking, we can consider the network structure as the spacetime structure because the connecting weights of the network contain the metric information and the propagation direction of the network is the holographic direction [36]. Also, we can imagine a scene where the conductivity travels along the network by perceiving the spacetime information locally.

However, we notice that the derivative of the metric function, namely $f'(z)$, appears in the equations of motion as well, which is in contrast to the discretized version of the equations of motion appearing in previous literature on deep learning in holography [36,42]. In principle, we may treat $f(z)$ and $f'(z)$ as two independent variables and train them independently. In practice, however, we find that this makes the optimization process of deep learning much more difficult. To obtain a feasible deep learning process, we discretize $f'(z)$ in terms of its relation to $f(z)$. There are many different ways to discretize $f'(z)$. Here we select four distinct varieties as listed below and investigate their effects on the final results (see detailed information in Appendix A).

- (1) $\ln f(z)$ forward: $\frac{f'(z)}{f(z)} = (\ln f(z))' \approx \frac{\ln f(z+\Delta z) - \ln f(z)}{\Delta z}$.
- (2) $\ln f(z)$ middle: $\frac{f'(z)}{f(z)} \approx \frac{\ln f(z+\Delta z) - \ln f(z-\Delta z)}{2\Delta z}$.
- (3) $f(z)$ forward: $f'(z) \approx \frac{f(z+\Delta z) - f(z)}{\Delta z}$.
- (4) $f(z)$ middle: $f'(z) \approx \frac{f(z+\Delta z) - f(z-\Delta z)}{2\Delta z}$.

We introduce the loss function to evaluate the difference between the true values and the results predicted by the neural network. One criterion of designing a neural network is to make the loss function as small as possible. According to the previous work [36,42], we introduce two loss functions as below,

$$L_{\text{total},1} = \frac{1}{N} \sum_{\text{data}} |\sigma(\omega, z_h) - \hat{\sigma}(\omega, z_h)| + L_{\text{REG}}, \quad (19)$$

$$L_{\text{total},2} = \frac{1}{N} \sum_{\text{data}} |\sigma(\omega, z_h) - \hat{\sigma}(\omega, z_h)|^2, \quad (20)$$

where the regularization term is

$$L_{\text{REG}} = c_1 [f(z(0)) - 1]^2 + \frac{c_2}{(n_{\text{epoch}}/10)^{1.5}} \sum_{n=0}^{N-1} \frac{1}{z(n)^2} [f(z(n+1)) - f(z(n))]^2. \quad (21)$$

In the above equations, σ is the input data while $\hat{\sigma}$ is what the network predicts. c_1 and c_2 are hyperparameters, which we can tune manually. n_{epoch} is the number of epochs we run, where an epoch is defined as the period that the full dataset propagates through the neural network once. Here the regularization term L_{REG} contains two terms and they are used to find a reasonable metric, which differs greatly from the common effect of overcoming the overfitting. The first term is to guarantee the asymptotically AdS property of spacetime at $z_b = 0$, while the second term is to suppress the possibility of large gradients to promote the efficiency of the neural network to figure out a smooth metric function numerically. We find that both terms are important for the deep learning process, just as in previous works [36,39,42].

III. THE SETUP FOR TRAINING DATA AND DISCRETIZATION

In this section, we present the setup for the training data and figure out the best way to discretize $f'(z)$. Given the theoretical result of the metric function $f(z)$, then from (10) one can numerically obtain the data of optical conductivity from the boundary ($\sigma(z_b)$) to the horizon ($\sigma(z_h)$) for any specified frequency $\omega > 0$, as performed in ordinary holographic approach. Now we reverse the problem by setting $f(z)$ as an unknown function and try to learn it by inputting the data of optical conductivity. For this purpose, near the boundary, $z = 0$ we fix the location of the cutoff at $z_b = 0.01$ and input 2000 numerical data of optical conductivity

with ω uniformly sampled along $(0.1, 1]$ as initial data at the cutoff. Next, since the conductivity becomes divergent at the horizon $z = 1$, we also need to introduce a cutoff z_h near the horizon. Now for each input data, one can generate the data of conductivity at z_h as output data through the neural network. Finally, one can train the neural network to learn the metric function by comparing the output data with the theoretical results. All the training data $\{(\sigma(z_b), \sigma(z_h))\}$ we use can be found in [50]. Here, we study two cases, $z_h = 0.9$ and $z_h = 0.99$, to test the learning ability of the neural network.

Next, we need to fix the number of layers in the neural network. The discretization in the process of deep learning introduces truncation error, which can be decreased by increasing the number of layers in the neural network. In theory, constructing deeper neural networks with more layers would improve the accuracy, however, with the price of consuming time and intensifying resources. In practice, we find that an 11-layer neural network in this work suffices to provide results that are strikingly close to those of deeper networks.

Finally, we intend to pick out the best way of discretizing $f'(z)$ for the neural network. For this purpose, we show the output data of the standard conductivity and the reduced conductivity at $z_h = 0.99$ with $\mu = 1$ in Figs. 2 and 3, respectively, which are generated by the neural network with various discretizations of $f'(z)$. We also present the numerical result by directly solving the differential equation with the finite difference method, which might be

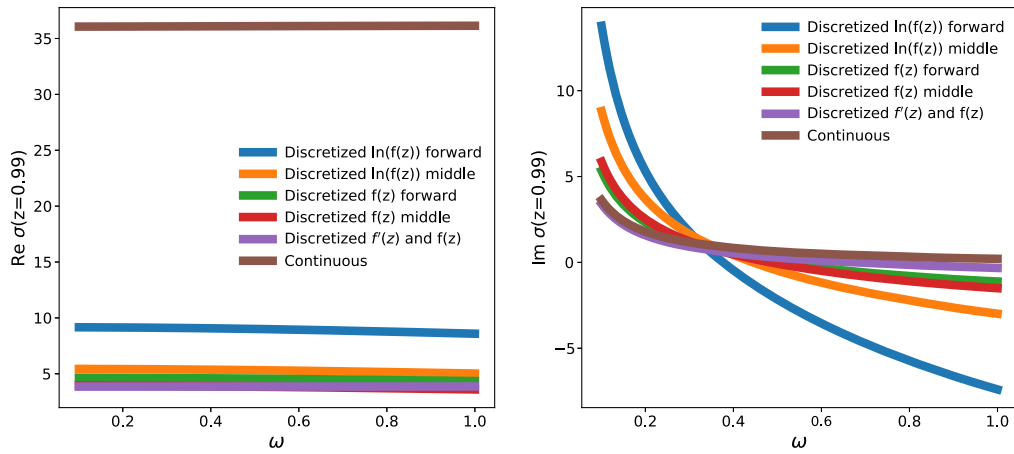


FIG. 2. The standard conductivity at $z = 0.99$ generated by the neural network with different discretizations of $f'(z)$. The left (right) plot is the real (imaginary) part of the optical conductivity. The brown curve is the numerical result by directly solving the differential equation with the finite difference method.

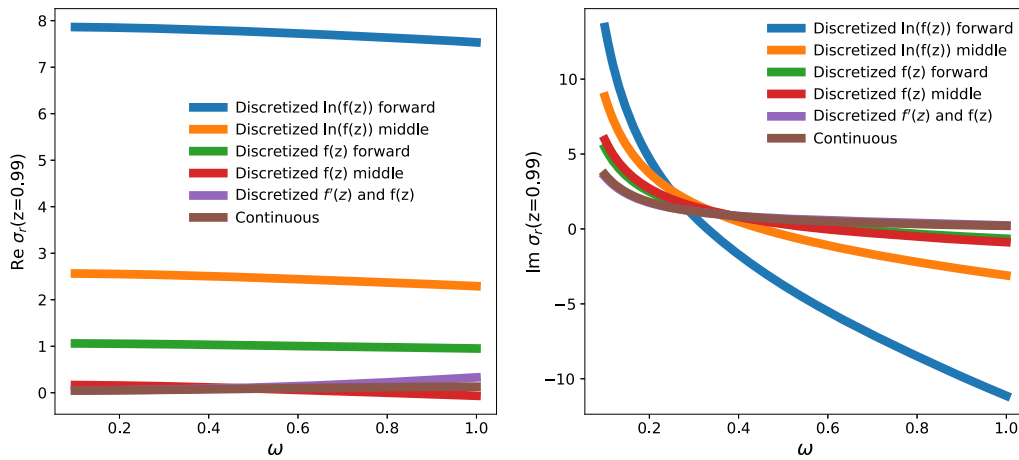


FIG. 3. The reduced conductivity at $z = 0.99$ generated by the neural network with different discretizations of $f'(z)$. The left (right) plot is the real (imaginary) part of the optical conductivity.

viewed as the “true” values of the conductivity, namely the data obtained by the deep neural network in the continuous limit.

First, let us focus on the output of the standard conductivity in Fig. 2. It is noticed that the data of the conductivity obtained by the “ $f(z)$ forward” discretization looks closer to the data by the continuous limit. However, in practice, we find that the “ $\ln f(z)$ forward” discretization performs more robustly, and with it one can get more accurate metric information than “ $f(z)$ forward” discretization (see the comparison in Table I). We can make the difference of results between “ $f(z)$ forward” and “ $\ln f(z)$ forward” smaller by increasing the number of network layers, but the “ $\ln f(z)$ forward” method is intrinsically more robust. Therefore, with comprehensive consideration, we decide to adopt “ $\ln f(z)$ forward” method to train the

network. Similarly, we compare the output of the reduced conductivity in Fig. 3, and find that “ $f(z)$ forward” discretization is the best way for deep learning process of the reduced conductivity.

We obtain the similar results for the case with $(z_h = 0.99, \mu = 2)$, $(z_h = 0.9, \mu = 2)$, and other combinations (in Appendix B). As a result, we choose “ $\ln f(z)$ forward” discretization for the standard conductivity and “ $f(z)$ forward” discretization for the reduced conductivity in the construction of the neural network.

More importantly, we find that the output data of the reduced conductivity at z_h is much closer to the data of the continuous limit than that of the standard conductivity. In particular, as z_h approaches the location of the horizon, the reduced conductivity exhibits its advantages more evidently since the divergent part has been peeled off.

So in the next section, we focus on the results of the neural network constructed with the reduced conductivity. For full results, please see [50].

IV. RESULTS OF THE LEARNED METRIC

A. The result of learned metric with $\mu = 1$ and $z_h = 0.99$

First, we show a typical example of the training results for the metric function $f(z)$ with $\mu = 1$ and $z_h = 0.99$, which is illustrated in Fig. 4. The left figure is the result of

the learned metric. It shows that after the deep learning process, the initial randomly selected metric becomes the true metric. Two plots on the right-hand side are the output data of the reduced conductivity at z_h . It shows that the reduced conductivity generated by the initial metric is far away from the true one, while after the deep learning process, it is quite close to the true conductivity, indicating that the neural network has successfully learned the metric from the reduced conductivity. Also, we find that, after the first training process, the results of both the metric and

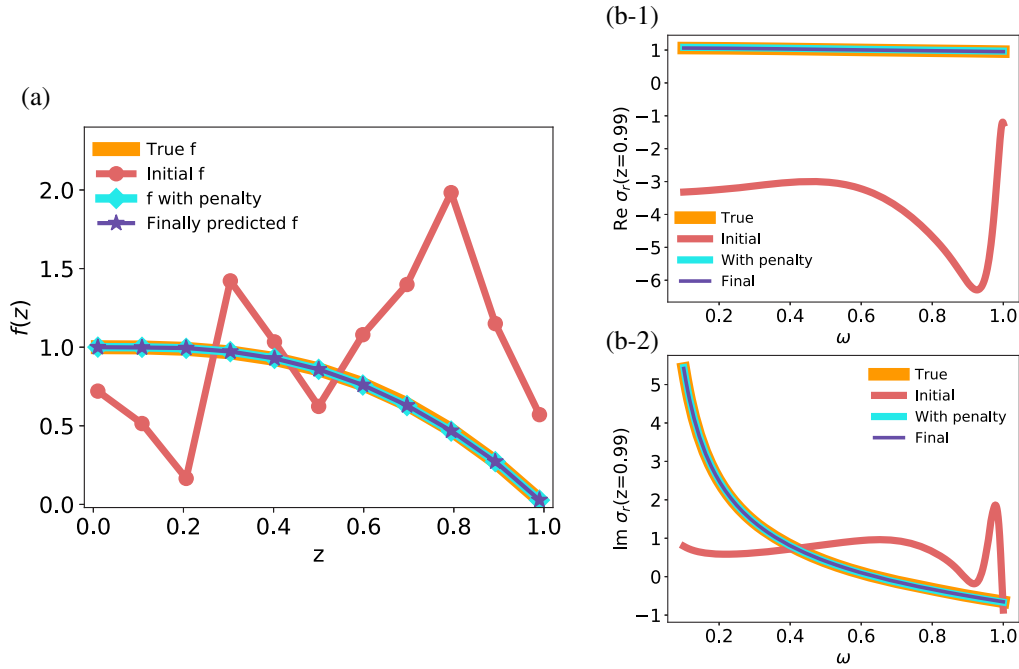


FIG. 4. The results of deep learning. (a) The learned metric results. The orange curve represents the true metric $f(z)$. The red curve represents the random initial weights of the metric in the network. The blue curve represents the trained metric after the first training process, and the penalty means the regularization term in the first loss function. The purple one is the final metric after double training procedures. (b-1) The real part of the reduced conductivity generated by different metrics. The orange, red, blue, and purple curves represent the real part of reduced conductivity generated by the true metric, the random initial metric, the metric after the first training process, and the final metric after double training procedures, respectively. (b-2) The imaginary part of the reduced conductivity generated by different metrics.

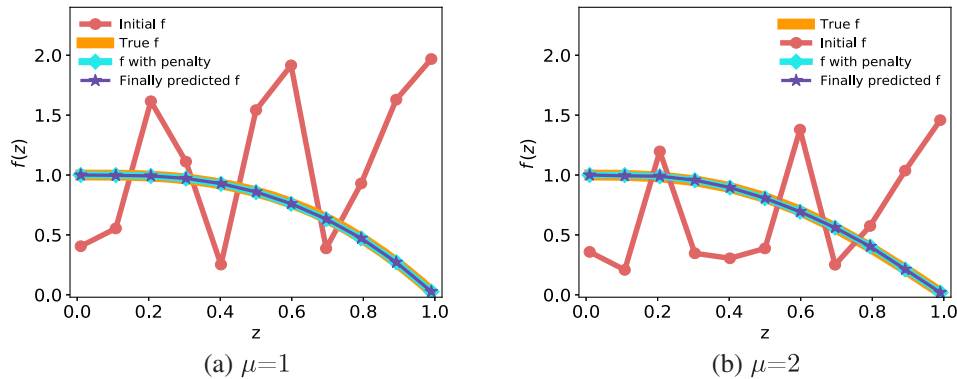


FIG. 5. The DNN (deep neural network) training results of the reduced conductivity with different values of chemical potential ($z_h = 0.99$, $\omega \in (0.1, 1]$).

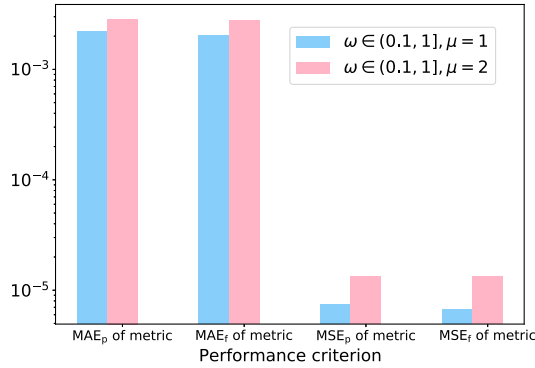
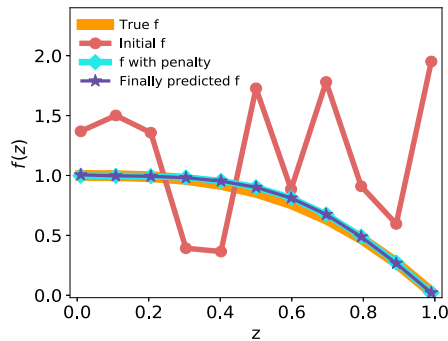


FIG. 6. The comparison of deep learning results for the reduced conductivity ($z_h = 0.99$, $\mu = 1, 2$). MAE (mean absolute error) and MSE (mean square error) are two types of criteria functions used to evaluate the performance of training the network, and their definitions may be found in Appendix B. The subscript p represents the penalty and it means the results for the first training procedure. Subscript f represents final and it means the results for the final training procedure.

reduced conductivity are almost as good as the final ones. For more details on the training methods and the training results, please see Appendix B. Because the target of the deep learning is the metric, next we show the results for the metric only.

B. The comparison of results between $\mu = 1$ and $\mu = 2$ at $z_h = 0.99$

In this subsection, we discuss the effects of the chemical potential μ on the training results, which can also be understood as the influence of temperature. Figure 5 is the deep learning results of the reduced conductivity at $z_h = 0.99$ with $\mu = 1$ (left plot) and $\mu = 2$ (right plot), respectively. We have tried various initial guesses and found they all converge to the true values of $f(z)$ with great



(a) $0.99 < \omega \leq 1$

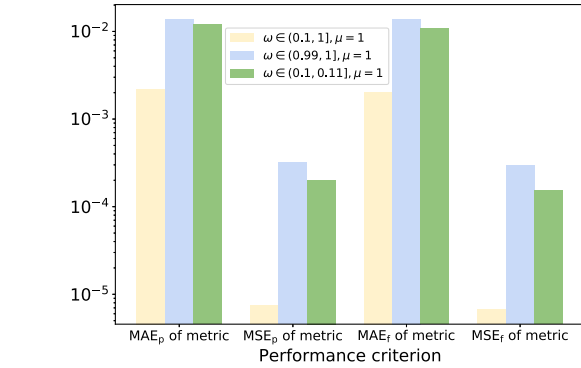


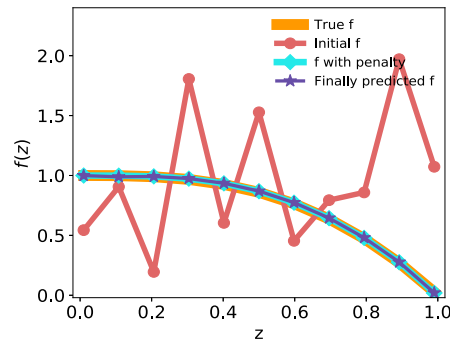
FIG. 8. The comparison of deep learning results for the reduced conductivity among three different ranges of ω .

accuracy. This shows that the neural network is powerful and robust in learning the metric from optical conductivity.

We show more concrete performance criteria in Fig. 6. It is noticed that the effect of the deep learning for $\mu = 1$ is better than that of the case $\mu = 2$. This result holds also for many other training data [50]. Nevertheless, one can see that both training results are greatly improved after the second training process.

C. The comparison of training results for $\mu = 1$ and $z_h = 0.99$ with different ω ranges

In this subsection, we discuss how the different ranges of the frequency ω affect the final training results. Previously, the research on the shear viscosity provided a positive answer to whether the neural network can learn the metric with the data in a narrow range of ω [42]. Here we intend to justify if this is also true for optical conductivity. For this purpose, we study two different ranges of ω : (0.1, 0.11] and (0.99, 1], each of which takes 2000 data points into account.



(b) $0.1 < \omega \leq 0.11$

FIG. 7. The DNN training results of the reduced conductivity at $z_h = 0.99$ with $\mu = 1$ and $\omega \in (0.99, 1]$ (left plot) and (0.1, 0.11] (right plot).

Figure 7 shows the deep learning results of the reduced conductivity at $z_h = 0.99$ with $\mu = 1$ and $\omega \in (0.99, 1]$ (left plot) and $(0.1, 0.11]$ (right plot), respectively. Figure 8 gives the concrete performance criteria of three different ω ranges at $z_h = 0.99$ of $\mu = 1$. We find that the training performance is better when the range of ω is wider. In addition, a larger μ will worsen the training outcomes, as illustrated in Appendix B.

V. CONCLUSION AND DISCUSSION

We have constructed a neural network to learn the RN-AdS black hole metric in the bulk based on the data of optical conductivity on the boundary by holography. The equation of motion that we recast into a neural network is generated by perturbing the vector field, thus enriching the prior research that only studied the scalar field or metric tensor field. In contrast to previous models, in this circumstance, the derivative of the metric function $f'(z)$ appears in the equation of motion, and we have proposed four distinct finite difference methods to discretize $f'(z)$. We have investigated their performance during the deep learning process in detail. Furthermore, to recast the equations of motion into a numerically feasible neural network, we have defined the reduced conductivity to avoid the divergence of the optical conductivity near the horizon. In addition, we have proposed a novel regularization term that automatically tunes the hyperparameters, which ensures the robustness and efficiency of the training methods. We have also discussed the dependence of the training outcomes on the location of the cutoff z_h , the temperature as well as the frequency range. It turns out that the network is harder to train as z_h approaches the horizon and as the temperature decreases. Given the number of data points, the training results with a wider range of ω are better than those with a narrower range. This can be understood from the fact that data from wider ranges of frequency contains more information than that from the narrower ranges.

This work has explicitly demonstrated the remarkable power of deep learning in the reconstruction of the spacetime with the given data on the boundary. For further study, we expect the AdS/DL method may be applied to AdS/CMT duality and shed light on the open problems in strongly coupled many-body system. For instance, given the RG flow data of the optical conductivity of the strange metal, the neural network would learn the metric of the bulk geometry that is capable of reproducing all the transport features of the strange metal. Currently, such kind of metric in the framework of AdS/CMT is unknown. Without doubt, the neural network presented in this paper is too simple to accomplish this task. We expect it could be developed into a network with more abundant structure and functions, such that its ability of learning the background information could be greatly improved. As the next step, one could consider a neural network with more neurons such that it could learn

more unknown functions rather than a single unknown function in the metric. In addition, we expect that the AdS/DL method may be applied to more holographic models and learn the bulk geometry by inputting the data of other transport quantities such as the thermal conductivity, etc.

An even more ambitious goal of AdS/DL is to learn the action of the dual gravity system from boundary data, which is of crucial significance not only for finding holographic models to understand important phenomena in dual systems but also for comprehending the implications of machine learning in holographic reconstruction of spacetime geometry. However, many challenges persist in realizing this goal, such as the degeneracy between the boundary data and the action of the dual theory, the construction of a machine learning model that can establish a relationship between the boundary data and the action represented by a symbolic system, etc. Recent advancements in machine learning offer promising prospects for directly learning the action from boundary data. For instance, the representation of symbolic space is comparable to natural language, and there exist highly effective methods, such as seq2seq [51], that can effectively address the problem. The SymbolicMathematics [52], empowered by seq2seq, is even more powerful in solving integral problems than well-known commercial software such as *Mathematica* and Matlab. These methods provide valuable strategies for representing and exploring the symbolic space. To solve the problem of degeneracy, on one hand, we can reduce the necessary variables in the model based on physical considerations, such as symmetry requirements. On the other hand, compared to the electrical conductivity that we currently consider, one may further reduce the degeneracy by increasing the type of boundary data, such as thermal conductivity, entropy, etc. Furthermore, a crucial capability of machine learning is generalization, meaning that it has the potential to address problems outside its training data range. Currently, advances such as diffusion models and ChatGPT have robustly demonstrated this [53,54]. We have reason to believe that given sufficiently high-quality datasets, machine learning has the potential to learn more intrinsic properties of holographic gravity and greatly contribute to the development of the AdS/DL.

ACKNOWLEDGMENTS

We are very grateful to Chen Bai, Yi Gu, Jiahao He, Yu Tian, Xiaoning Wu, and Hongbao Zhang for helpful discussions. This work is supported in part by the Natural Science Foundation of China (Grants No. 11875053, No. 11905083, and No. 12035016), the Science and Technology Planning Project of Guangzhou (202201010655), the Beijing Natural Science Foundation (Grant No. 1222031), and the creative practice training project of UCAS.

APPENDIX A: FOUR TYPES OF FINITE DIFFERENCES OF $f'(z)$ (1) $\ln f(z)$ forward

$$\begin{aligned}\operatorname{Re}\sigma(z + \Delta z) &= \operatorname{Re}\sigma(z) + (\ln f(z) - \ln f(z + \Delta z))\operatorname{Re}\sigma(z) + \Delta z[2\omega\operatorname{Im}\sigma(z)\operatorname{Re}\sigma(z)], \\ \operatorname{Im}\sigma(z + \Delta z) &= \operatorname{Im}\sigma(z) + (\ln f(z) - \ln f(z + \Delta z))\operatorname{Im}\sigma(z) + \Delta z\left[\frac{\omega}{f^2(z)} - \frac{\mu^2 z^2}{\omega f(z)} + \omega(\operatorname{Im}\sigma(z))^2 - \omega(\operatorname{Re}\sigma(z))^2\right].\end{aligned}\quad (\text{A1})$$

(2) $\ln f(z)$ middle

$$\begin{aligned}\operatorname{Re}\sigma(z + \Delta z) &= \operatorname{Re}\sigma(z) + \frac{(\ln f(z - \Delta z) - \ln f(z + \Delta z))}{2}\operatorname{Re}\sigma(z) + \Delta z[2\omega\operatorname{Im}\sigma(z)\operatorname{Re}\sigma(z)], \\ \operatorname{Im}\sigma(z + \Delta z) &= \operatorname{Im}\sigma(z) + \frac{(\ln f(z - \Delta z) - \ln f(z + \Delta z))}{2}\operatorname{Im}\sigma(z) + \Delta z\left[\frac{\omega}{f^2(z)} - \frac{\mu^2 z^2}{\omega f(z)} + \omega(\operatorname{Im}\sigma(z))^2 - \omega(\operatorname{Re}\sigma(z))^2\right].\end{aligned}\quad (\text{A2})$$

(3) $f(z)$ forward

$$\begin{aligned}\operatorname{Re}\sigma(z + \Delta z) &= \operatorname{Re}\sigma(z) + \left(1 - \frac{f(z + \Delta z)}{f(z)}\right)\operatorname{Re}\sigma(z) + \Delta z[2\omega\operatorname{Im}\sigma(z)\operatorname{Re}\sigma(z)], \\ \operatorname{Im}\sigma(z + \Delta z) &= \operatorname{Im}\sigma(z) + \left(1 - \frac{f(z + \Delta z)}{f(z)}\right)\operatorname{Im}\sigma(z) + \Delta z\left[\frac{\omega}{f^2(z)} - \frac{\mu^2 z^2}{\omega f(z)} + \omega(\operatorname{Im}\sigma(z))^2 - \omega(\operatorname{Re}\sigma(z))^2\right].\end{aligned}\quad (\text{A3})$$

(4) $f(z)$ middle

$$\begin{aligned}\operatorname{Re}\sigma(z + \Delta z) &= \operatorname{Re}\sigma(z) + \frac{f(z - \Delta z) - f(z + \Delta z)}{2f(z)}\operatorname{Re}\sigma(z) + \Delta z[2\omega\operatorname{Im}\sigma(z)\operatorname{Re}\sigma(z)], \\ \operatorname{Im}\sigma(z + \Delta z) &= \operatorname{Im}\sigma(z) + \frac{f(z - \Delta z) - f(z + \Delta z)}{2f(z)}\operatorname{Im}\sigma(z) + \Delta z\left[\frac{\omega}{f^2(z)} - \frac{\mu^2 z^2}{\omega f(z)} + \omega(\operatorname{Im}\sigma(z))^2 - \omega(\operatorname{Re}\sigma(z))^2\right].\end{aligned}\quad (\text{A4})$$

We remark that for both of the middle methods the first layer is not defined since the data of the current layer depends on the data in the previous and next layers. Thus, in practice we adopt the forward difference method to define the first layer. For more details, please see Appendix C.

APPENDIX B: DNN TRAINING PARAMETERS AND RESULTS

In this appendix we present the details of the training process, including the setup for epochs, loss functions, learning rate, the optimization algorithm, as well as the training criteria.

As a whole, the training process is divided into two steps. The first step contains 3001 epochs, while the second step contains 2001 epochs. The loss function for each step has been shown in the main body of the paper. The learning speed with L_1 loss is faster while L_2 loss can make the final metric more smooth.

For the optimization algorithm, we use the RMSprop optimizer in the first step and the Adam optimizer in the second step. The batch size is fixed as 200.

For the learning rate, we reduce it gradually along with the increase of the epoch by applying the module `torch.optim.lr_scheduler.MultiStepLR` in PyTorch. At the first step, when the number of epochs is within (1–1000), the learning rate is set as 10^{-3} . In the range of (1001–1500) and (1501–3001), the learning rate is set as 9×10^{-4} and 8.1×10^{-5} respectively. Similarly, at the second step when the number of epoch is within (1–500), (501–1000), (1001–1250), (1251–1500), (1501–1750), and (1751–2001), the learning rate is 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} , 10^{-7} , and 10^{-8} , respectively. In principle, this kind of selection for learning rate is empirical and the epoch is large enough such that the loss will not reduce and fluctuate at some quantity.

The values of hyperparameters c_1 and c_2 are specified quite casually because of our special design of the

TABLE I. All data of the black hole metrics training results.

Number	Type	μ	z_h	Difference type	ω	c_1	c_2	MAE_p	MSE_p	MAE_f	MSE_f
1	real	1	0.9	ln(f(z)) forward	(0.1,1]	1	50	2.856×10^{-3}	1.306×10^{-5}	1.519×10^{-3}	3.596×10^{-6}
2	real	1	0.9	ln(f(z)) forward	(0.99,1]	1	50	2.724×10^{-3}	1.213×10^{-5}	2.457×10^{-3}	1.014×10^{-5}
3	real	1	0.9	ln(f(z)) forward	(0.1,0.11]	10	50	3.313×10^{-3}	1.570×10^{-5}	3.296×10^{-3}	1.627×10^{-5}
4	real	1	0.99	ln(f(z)) forward	(0.1,1]	1	50	3.205×10^{-3}	2.422×10^{-5}	2.278×10^{-3}	9.641×10^{-6}
5	real	1	0.99	f(z) forward	(0.1,1]	1	50	4.599×10^{-2}	3.235×10^{-3}	5.121×10^{-2}	4.516×10^{-3}
6	real	1	0.99	ln(f(z)) forward	(0.99,1]	10	100	1.956×10^{-2}	6.080×10^{-4}	7.725×10^{-3}	1.128×10^{-4}
7	real	1	0.99	ln(f(z)) forward	(0.1,0.11]	20	100	1.564×10^{-2}	3.622×10^{-4}	1.664×10^{-2}	3.632×10^{-4}
8	reduced	1	0.9	f(z) forward	(0.1,1]	1	50	3.485×10^{-3}	1.894×10^{-5}	2.564×10^{-4}	1.411×10^{-7}
9	reduced	1	0.9	f(z) forward	(0.99,1]	1	50	5.942×10^{-3}	5.149×10^{-5}	4.782×10^{-3}	3.399×10^{-5}
10	reduced	1	0.9	f(z) forward	(0.1,0.11]	1	50	9.860×10^{-3}	1.377×10^{-4}	9.216×10^{-3}	1.273×10^{-4}
11	reduced	1	0.99	f(z) forward	(0.1,1]	1	50	2.208×10^{-3}	7.432×10^{-6}	2.042×10^{-3}	6.681×10^{-6}
12	reduced	1	0.99	f(z) middle	(0.1,1]	1	50	1.668×10^{-2}	9.275×10^{-4}	2.212×10^{-2}	1.175×10^{-3}
13	reduced	1	0.99	f(z) forward	(0.99,1]	10	200	1.377×10^{-2}	3.220×10^{-4}	1.383×10^{-2}	2.970×10^{-4}
14	reduced	1	0.99	f(z) forward	(0.1,0.11]	10	200	1.203×10^{-2}	2.009×10^{-4}	1.076×10^{-2}	1.526×10^{-4}
15	real	2	0.9	ln(f(z)) forward	(0.1,1]	1	50	4.261×10^{-3}	2.773×10^{-5}	1.661×10^{-3}	5.128×10^{-6}
16	real	2	0.9	ln(f(z)) forward	(0.99,1]	10	200	3.718×10^{-2}	2.372×10^{-3}	2.521×10^{-2}	1.070×10^{-3}
17	real	2	0.9	ln(f(z)) forward	(0.1,0.11]	10	100	6.360×10^{-2}	5.514×10^{-3}	6.351×10^{-2}	5.479×10^{-3}
18	real	2	0.99	ln(f(z)) forward	(0.1,1]	1	200	1.036×10^{-2}	2.017×10^{-4}	4.278×10^{-3}	3.327×10^{-5}
19	real	2	0.99	ln(f(z)) forward	(0.99,1]	10	200	4.773×10^{-2}	4.332×10^{-3}	5.715×10^{-2}	6.621×10^{-3}
20	real	2	0.99	ln(f(z)) forward	(0.1,0.11]	10	200	4.973×10^{-2}	3.772×10^{-3}	4.459×10^{-2}	3.045×10^{-3}
21	reduced	2	0.9	f(z) forward	(0.1,1]	1	50	3.292×10^{-3}	1.736×10^{-5}	2.139×10^{-3}	1.041×10^{-5}
22	reduced	2	0.9	f(z) forward	(0.99,1]	10	100	4.151×10^{-2}	3.483×10^{-3}	3.457×10^{-2}	2.473×10^{-3}
23	reduced	2	0.9	f(z) forward	(0.1,0.11]	10	100	5.500×10^{-2}	5.048×10^{-3}	5.534×10^{-2}	5.152×10^{-3}
24	reduced	2	0.99	f(z) forward	(0.1,1]	1	50	2.860×10^{-3}	1.336×10^{-5}	2.779×10^{-3}	1.323×10^{-5}
25	reduced	2	0.99	f(z) forward	(0.99,1]	10	100	6.015×10^{-2}	6.861×10^{-3}	4.822×10^{-2}	4.326×10^{-3}
26	reduced	2	0.99	f(z) forward	(0.1,0.11]	10	200	8.112×10^{-2}	1.012×10^{-2}	8.092×10^{-2}	1.013×10^{-2}

regularization term. In practice, we choose 50, 100, and 200 for c_2 and 1, 10, and 20 for c_1 correspondingly. During the training process, the harder is the training task, the larger the value of c_1 needs to be. Nevertheless, in general the setup of c_1 and c_2 does not affect the training results much. In particular, the involvement of the second training step makes the specification of c_1 and c_2 less important.

The training criteria (MAE and MSE) are shown below,

$$\begin{aligned}
 MAE &= \sum_{i=1}^N \frac{|f_p^i - f_t^i|}{N}, \\
 MSE &= \sum_{i=1}^N \frac{(f_p^i - f_t^i)^2}{N}, \quad (\text{B1})
 \end{aligned}$$

where i represents the i th layer and f_p^i is the quantity that the network trains, which is just the metric function $f(z)$ in

this work. f_p^i represents the metric of prediction, while f_t^i refers to the true metric.

To prevent the influence of contingency factors and statistical fluctuations on the learning process, we train five times for each training process and set the average of these results as our final results. All the training results are listed in Table I.

APPENDIX C: THE Python CODE

We employ PyTorch (CPU version 1.6.0) and Python (version 3.7.9) to implement the deep learning process in this work. It is suggested to run this in Anaconda3. The modules such as PyTorch, TORCHVISION, and MoviePy should be installed in advance. The main code and a set of training data can be found in [50].

[1] G. 't Hooft, Dimensional reduction in quantum gravity, Conf. Proc. C **930308**, 284 (1993), <https://arxiv.org/abs/gr-qc/9310026>.

[2] J. M. Maldacena, The large N limit of superconformal field theories and supergravity, Adv. Theor. Math. Phys. **2**, 231 (1998).

- [3] E. Witten, Anti-de Sitter space and holography, *Adv. Theor. Math. Phys.* **2**, 253 (1998).
- [4] O. Aharony, S. S. Gubser, J. M. Maldacena, H. Ooguri, and Y. Oz, Large N field theories, string theory and gravity, *Phys. Rep.* **323**, 183 (2000).
- [5] S. S. Gubser, Breaking an Abelian gauge symmetry near a black hole horizon, *Phys. Rev. D* **78**, 065034 (2008).
- [6] S. A. Hartnoll, C. P. Herzog, and G. T. Horowitz, Building a Holographic Superconductor, *Phys. Rev. Lett.* **101**, 031601 (2008).
- [7] S. A. Hartnoll, Lectures on holographic methods for condensed matter physics, *Classical Quantum Gravity* **26**, 224002 (2009).
- [8] J. McGreevy, Holographic duality with a view toward many-body physics, *Adv. High Energy Phys.* **2010**, 723105 (2010).
- [9] S. A. Hartnoll, A. Lucas, and S. Sachdev, Holographic quantum matter, [arXiv:1612.07324](https://arxiv.org/abs/1612.07324).
- [10] J. Zaanen, Y. Liu, Y. Sun, and K. Schalm, *Holographic Duality in Condensed Matter Physics* (Cambridge University Press, Cambridge, England, 2015).
- [11] M. Ammon and J. Erdmenger, *Gauge/Gravity Duality* (Cambridge University Press, Cambridge, England, 2015).
- [12] S. A. Hartnoll, J. Polchinski, E. Silverstein, and D. Tong, Towards strange metallic holography, *J. High Energy Phys.* **04** (2010) 120.
- [13] G. T. Horowitz, Introduction to holographic superconductors, *Lect. Notes Phys.* **828**, 313 (2011).
- [14] M. Cubrovic, J. Zaanen, and K. Schalm, String theory, quantum phase transitions and the emergent Fermi-liquid, *Science* **325**, 439 (2009).
- [15] N. Iqbal, H. Liu, and M. Mezei, Lectures on holographic non-Fermi liquids and quantum phase transitions, in *String Theory and Its Applications* (World Scientific, Singapore, 2011), 10.1142/97898143505250013.
- [16] G. T. Horowitz and J. E. Santos, General relativity and the cuprates, *J. High Energy Phys.* **06** (2013) 087.
- [17] Y. Ling, C. Niu, J. Wu, Z. Xian, and H. b. Zhang, Metal-Insulator Transition by Holographic Charge Density Waves, *Phys. Rev. Lett.* **113**, 091602 (2014).
- [18] Y. Ling, P. Liu, and J. P. Wu, Note on the butterfly effect in holographic superconductor models, *Phys. Lett. B* **768**, 288 (2017).
- [19] Y. Ling and Z. Y. Xian, Holographic butterfly effect and diffusion in quantum critical region, *J. High Energy Phys.* **09** (2017) 003.
- [20] Y. Ling, P. Liu, J. P. Wu, and M. H. Wu, Holographic superconductor on a novel insulator, *Chin. Phys. C* **42**, 013106 (2018).
- [21] Y. Ling, P. Liu, and M. H. Wu, Holographic superconductor induced by charge density waves, *Phys. Rev. D* **102**, 126013 (2020).
- [22] Y. Ling and M. H. Wu, Holographic striped superconductor, *J. High Energy Phys.* **03** (2021) 260.
- [23] S. Sachdev, What can gauge-gravity duality teach us about condensed matter physics?, *Annu. Rev. Condens. Matter Phys.* **3**, 9 (2012).
- [24] J. Zaanen, Planckian dissipation, minimal viscosity and the transport in cuprate strange metals, *SciPost Phys.* **6**, 061 (2019).
- [25] P. Hayden, S. Nezami, X. L. Qi, N. Thomas, M. Walter, and Z. Yang, Holographic duality from random tensor networks, *J. High Energy Phys.* **11** (2016) 009.
- [26] B. Swingle, Entanglement renormalization and holography, *Phys. Rev. D* **86**, 065007 (2012).
- [27] B. Swingle, Constructing holographic spacetimes using entanglement renormalization, [arXiv:1209.3304](https://arxiv.org/abs/1209.3304).
- [28] B. Cédric, Deep learning and the renormalization group, [arXiv:1301.3124](https://arxiv.org/abs/1301.3124).
- [29] J. Chen, S. Cheng, H. D. Xie, L. Wang, and T. Xiang, Equivalence of restricted Boltzmann machines and tensor network states, *Phys. Rev. B* **97**, 085104 (2018).
- [30] P. Mehta and D. J. Schwab, An exact mapping between the variational renormalization group and deep learning, [arXiv:1410.3831](https://arxiv.org/abs/1410.3831).
- [31] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [32] F. Ruehle, Data science applications to string theory, *Phys. Rep.* **839**, 1 (2020).
- [33] W. C. Gan and F. W. Shu, Holography as deep learning, *Int. J. Mod. Phys. D* **26**, 1743020 (2017).
- [34] Y. Z. You, Z. Yang, and X. L. Qi, Machine learning spatial geometry from entanglement features, *Phys. Rev. B* **97**, 045153 (2018).
- [35] J. Lam and Y. Z. You, Machine learning statistical gravity from multi-region entanglement entropy, *Phys. Rev. Res.* **3**, 043199 (2021).
- [36] K. Hashimoto, S. Sugishita, A. Tanaka, and A. Tomiya, Deep learning and the AdS/CFT correspondence, *Phys. Rev. D* **98**, 046019 (2018).
- [37] K. Hashimoto, S. Sugishita, A. Tanaka, and A. Tomiya, Deep learning and holographic QCD, *Phys. Rev. D* **98**, 106014 (2018).
- [38] K. Hashimoto, AdS/CFT correspondence as a deep Boltzmann machine, *Phys. Rev. D* **99**, 106017 (2019).
- [39] J. Tan and C. B. Chen, Deep learning the holographic black hole with charge, *Int. J. Mod. Phys. D* **28**, 1950153 (2019).
- [40] T. Akutagawa, K. Hashimoto, and T. Sumimoto, Deep learning and AdS/QCD, *Phys. Rev. D* **102**, 026020 (2020).
- [41] K. Hashimoto, K. Ohashi, and T. Sumimoto, Deriving the dilaton potential in improved holographic QCD from the meson spectrum, *Phys. Rev. D* **105**, 106008 (2022).
- [42] Y. K. Yan, S. F. Wu, X. H. Ge, and Y. Tian, Deep learning black hole metrics from shear viscosity, *Phys. Rev. D* **102**, 101902 (2020).
- [43] K. Hashimoto, H. Y. Hu, and Y. Z. You, Neural ordinary differential equation and holographic quantum chromodynamics, *Mach. Learn. Sci. Tech.* **2**, 035011 (2021).
- [44] H. Y. Hu, S. H. Li, L. Wang, and Y. Z. You, Machine learning holographic mapping by neural network renormalization group, *Phys. Rev. Res.* **2**, 023369 (2020).
- [45] I. M. Comsa, M. Firsching, and T. Fischbacher, SO(8) supergravity and the magic of machine learning, *J. High Energy Phys.* **08** (2019) 057.
- [46] C. Krishnan, V. Mohan, and S. Ray, Machine learning $\mathcal{N} = 8$, $D = 5$ gauged supergravity, *Fortschr. Phys.* **68**, 2000027 (2020).

- [47] J. Bao, Y. H. He, E. Heyes, and E. Hirst, Machine learning algebraic geometry for physics, [arXiv:2204.10334](https://arxiv.org/abs/2204.10334).
- [48] N. Iqbal and H. Liu, Universality of the hydrodynamic limit in AdS/CFT and the membrane paradigm, *Phys. Rev. D* **79**, 025023 (2009).
- [49] R. Kubo, Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems, *J. Phys. Soc. Jpn.* **12**, 570 (1957).
- [50] The PYTHON code for this work is available at <https://github.com/DiracMisaka/DL-holographic-conductivity.git>.
- [51] I. Sutskever, O. Vinyals, and Q. V. Le, Sequence to Sequence learning with neural networks, [arXiv:1409.3215](https://arxiv.org/abs/1409.3215).
- [52] G. Lample and F. Charton, Deep learning for symbolic mathematics, [arXiv:1912.01412](https://arxiv.org/abs/1912.01412).
- [53] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong, DiffuSeq: Sequence to sequence text generation with diffusion models, [arXiv:2210.08933](https://arxiv.org/abs/2210.08933).
- [54] <https://chat.openai.com>.