# Machine learning for many-body physics: efficient solution of dynamical mean-field theory

Louis-François Arsenault<sup>1</sup>, O. Anatole von Lilienfeld<sup>2,3</sup>, and Andrew J. Millis<sup>1</sup>

<sup>1</sup> Department of Physics, Columbia University, New York, New York 10027, USA

<sup>2</sup> Institute of Physical Chemistry, Department of Chemistry,

University of Basel, Klingelbergstrasse 80, CH-4056 Basel, Switzerland

<sup>3</sup> Argonne Leadership Computing Facility, Argonne National Laboratory, 9700 S. Cass Avenue, Lemont, IL 60439, USA

(Dated: July 1, 2015)

Machine learning methods for solving the equations of dynamical mean-field theory are developed. The method is demonstrated on the three dimensional Hubbard model. The key technical issues are defining a mapping of an input function to an output function, and distinguishing metallic from insulating solutions. Both metallic and Mott insulator solutions can be predicted. The validity of the machine learning scheme is assessed by comparing predictions of full correlation functions, of quasi-particle weight and particle density to values directly computed. The results indicate that with modest further development, machine learning approach may be an attractive computational efficient option for real materials predictions for strongly correlated systems.

PACS numbers: 71.10.-w,71.27.+a,89.20.Ff

The quantum many-body problem of predicting properties of systems containing electrons or other fermionic entities has challenged physicists and chemists for decades. This is so because the minus sign associated with fermionic exchange creates a host of difficulties including long-ranged entanglement and a Monte-Carlo sign problem. The net effect is to place the generic fermion many-body problem in the class of problems whose full solution is exponentially hard. Although new developments such as matrix product and tensor network methods may provide solutions to ground-state properties with only power-law cost, search for efficient approximate methods to handle a wide range of phenomena at a wide range of temperatures remains a key goal of condensed matter physics and quantum chemistry.

In this work, we investigate the use of Machine Learning (ML) [1] to leverage existing results and provide an efficient approximate solution to a generic class of problems in quantum many-body physics. ML is in essence a way to use a database of known solutions to infer information about a new problem. In the condensed matter physics context it has been used as an intermediate step in molecular dynamics calculations [2–7], to predict density functionals (so far only in the 1D context)[8], to obtain transmission coefficients for electron transport [9] and, very recently, to predict the fermi level density of states of weakly correlated solids [10] and find formation energies of materials [11]. These applications relate to classical physics and to single-particle quantum mechanics. In the quantum chemistry context, ML has been successfully applied to predict energies and other scalar properties of molecules [12–20]. In addition, non-ML ideas from data science have been recently proposed as ways to help the solution of the non-equilibrium manybody problem [21].

We propose to use machine learning methods to solve

true quantum many-body problems. A technical issue arises: in many applications of machine learning, including most of the ones referred to above, the goal is to infer a scalar property (e.g. an energy) of a model specified by a modest number of scalar parameters. However, the generic quantum many-body problem is the solution of a functional equation relating an input function (for example a bare electron Green's function) to an output function (e.g. an electron self-energy). Here we build on previous work [22] which involved learning a function specified by a modest number of input parameters, to develop a formalism capable of solving the more general problem of mapping a function to a function. In independent contemporaneous work, questions related to learning functions have been studied in the context of density functional theory where one seeks to learn the relation between a position-dependent charge density and an exchange-correlation potential [23, 24].

The context for our work is Dynamical Mean-Field Theory (DMFT)[25], a widely used approximate method for determining the properties of materials with strong electronic correlations. DMFT approximates the solution to an interacting fermion system in terms of the solution of an auxiliary quantum impurity problem. The impurity model, although simpler than the full problem, is still a quantum many-body problem. The impurity model is specified by a hybridization function  $\Delta(\omega)$ ; the many-body physics by a local Green function  $G(\omega)$  or self-energy  $\Sigma(\omega)$ . The hybridization function itself is obtained from a self-consistency condition which involves  $\Sigma(\omega)$  and an initial band structure which encodes the chemistry and crystal structure of the material in question and may be parametrized as a bare or initial hybridization function  $\Delta^0(\omega)$ . In standard applications, the self-consistency condition is solved by iteration. One may imagine a ML process to solve the impurity

model (relating  $\Delta$  and  $G/\Sigma$ ) or a ML process to solve the entire DMFT self-consistency loop (relating  $\Delta^0$  and  $\Delta^f, G^f/\Sigma^f$ ). In this paper, we only present results for the full solution of DMFT. Use of ML as impurity solver may also be valuable as an intermediate step, enabling the rapid construction of a database of solved problems in the real-materials context. Our formalism is general enough to apply to this possibility.

We test our methods using the Hubbard model defined on a three dimensional cubic lattice with first and second-neighbor hoppings, with Hamiltonian  $H = \sum_{k\sigma} (\varepsilon_k - \mu) \, c_{k\sigma}^\dagger c_{k\sigma} + U \sum_i n_{i\uparrow} n_{i\downarrow}$ . Here  $\mu$  is the chemical potential and  $\varepsilon_k = -2t \sum_{\alpha=1}^3 \cos\left(k_\alpha\right) - 4t' \left[\cos\left(k_1\right)\cos\left(k_2\right) + \cos\left(k_1\right)\cos\left(k_3\right) + \cos\left(k_2\right)\cos\left(k_3\right)\right]$ . The bare hybridization function is  $\Delta^0(\omega) = \omega + \mu - \left(\sum_k \frac{1}{\omega - \varepsilon_k + \mu}\right)^{-1}$ . We define energy units such that the full non-interacting bandwidth W = 2 where W = 12t if  $|t'| \leq t/4$  and W = 8t + 16|t'| if |t'| > t/4. Varying the ratio t'/t changes the structure in the density of states, in particular shifting the location of the density of states peaks relative to the band center (see Section I of the supplemental material for examples).

We seek a machine that enables us to map a  $\Delta^0$  to an output local Green function or self-energy. DMFT admits two classes of solutions: metallic ones with a nonvanishing density of states at the fermi level and a smooth self-energy, and Mott insulating solutions with a gap at the fermi level due to Coulomb repulsion and (in many cases) a self-energy with a pole near the chemical potential. We have found it advantageous to introduce a binary classification step that identifies a given solution as metallic or insulating and to use two different machines to determine the properties of the two kinds of solutions. For classification, we use the entire database minus one as training and the one remaining as the testing problem. We then repeat for all members of the database. We tested three different ML for classification: simple support vectors machine sym[28] with  $\sim 96\%$  accuracy, neural networks [29] with  $\sim 97\%$  accuracy and decision forests[30] with  $\sim 99.6\%$  accuracy. The only misplaced problems are critical metals extremely close to the transition. We only kept the decision forest as it outperformed the two others. Once the state of a new problem has been decided, the Kernel Ridge Regression (KRR) method [1, 22] (more details follow) is employed to determine the solution using the sub-databases containing only metal or Mott insulating solutions. The full ML process for DMFT is shown in Fig. 1 while some details about the parameters are explained later in the text.

The first step in implementing machine learning is to generate a database of initial conditions, in other words a set of bare hybridization functions that span a range of physically reasonable possibilities. We consider the set of hybridization functions defined by t' =

[0,-0.1t,-0.2t,-0.3t] (the case of positive t'/t could be accounted for by considering electron doping) and  $\mu=0$ . Sections I and II.A of the supplemental material give more details. We then obtain the database of solved problems by using the exact diagonalization (ED) method [25–27] to solve the single-site dynamical mean field approximation for interaction strengths in the range  $0.16 \leq U \leq 4$  and densities in the range of  $0.6 < n_d < 1.05$ . Particularities of the ED database are discussed in Section II.B of the supplemental material.

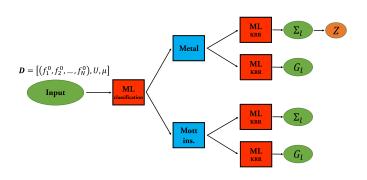


Figure 1: (Color online) Schematic view of DMFT as seen in a machine learning perspective. From an input description of a problem we are seeking a solution, the ML chooses first if the solution is metallic or insulating. Then the ML predicts the solution for the correlation function of choice by predicting the coefficients of the Legendre polynomials expansion of either the Green's function or self-energy. In the case of the self-energy for the metal, the ML predicted quasi-particle weight Z can also be extracted.

The second step in implementing machine learning is the construction of a representation of the information to be learned and of the descriptor  $\mathbf{D}$ , a unique identifier of a problem. Our input and output data are functions. Functions may be specified as a vector of coefficients in a space of basis functions  $\phi_m$  (e.g.  $\Sigma(z) = \sum_m s_m \phi_m(z)$ ). Our previous work[22], following work by Boehnke et al.[31] found that Legendre polynomials were a very efficient choice of basis, so we adopt this representation here. The Legendre representation is most naturally formulated in imaginary time  $0 < \tau < \beta$  with  $\beta$  the inverse temperature and hence a correlation function is  $f(\tau) = \sum_{l=0}^{\infty} \frac{\sqrt{2l+1}}{\beta} f_l P_l(x(\tau))[31]$ , where  $P_l$  are the Legendre polynomials. The Fourier transform to  $f(i\omega_n)$  can be done analytically [31]. The representation is general, we fit either the local Green's function or the self-energy as shown in Fig. 1 (or even the hybridization function). See Section IV of the Supplementary material for details.

The descriptor consists of the input function (hybridization function) plus a few scalar parameters; we denote the expansion coefficients of the function as f and the scalar parameters U (interaction

strength) and  $\mu$  (chemical potential) such that  $\mathbf{D} =$  $[(f_1, f_2, \ldots, f_N)_{input}, U, \mu]$  (see Fig. 1). Note that both the full DMFT problem and the impurity solving part are the same problem as far as ML is concerned, the only difference being what database one chooses. The exact diagonalization method used here provides a representation of the input hybridization function in terms of bath level energies  $\{\varepsilon_{ml}\}$  and hybridization parameters  $\{V_{ml}\}$ (m labels entries in the database and l labels the different bath energies and hybridization parameters for a given entry in the database) so in practice we use these for the  $f_m$ . We have also implemented machine learning using the representation of the input function in terms of Legendre polynomials, with essentially identical results (see Section V of the supplementary material). Section II.A of the supplemental material shows how the bare ED parameters are obtained from a known band structure.

Machine learning then estimates the solution  $f(z) \to f = (f_1, f_2, \dots, f_N)_{output}$  of a new problem in terms of an interpolation between known solutions. We use KRR, an expansion in the abstract multidimensional space of descriptors (each point **D** of this space represents a unique problem and the distance between two points is the distance metric), obtaining

$$\{f_m\} \approx \sum_{lm} \alpha_{lm} K_m \left( \mathbf{D}_l, \mathbf{D} \right),$$
 (1)

where l labels points in the dataset, m labels entries in the output vector and the kernel K is a function whose main characteristic is to weight most heavily the contributions of l for which  $\mathbf{D}_l$  is close to  $\mathbf{D}$ . As in [22], we use the weighted exponential kernel, and use the Manhattan distance between  $\mathbf{D}_l$  and  $\mathbf{D}$  (both are defined in Section III.A of the supplemental material). The expansion coefficients  $\alpha_{lm}$  are  $\overline{\overline{\alpha}} = \left(\overline{\overline{K}} + \lambda \overline{\overline{I}}\right)^{-1} \overline{\overline{f}}[22]$ , where  $\overline{\overline{\alpha}}$  is a matrix containing all the  $\alpha_{lm}$ ,  $\overline{\overline{K}}$  is the kernel matrix and  $\lambda$  is a regularization parameter.  $\lambda$  and the free parameter of the kernel are chosen using standard cross-validation, see Section III.A of the supplemental material. In particular, as also found in [18], we found that the actual value chosen for  $\lambda$  is not really important. This formalism is very general and could be applied to the learning of other

As first two tests of our predictive power we present scalar properties, the quasi-particle weight  $Z=(1-d\Sigma'/d\omega|_{\omega\to 0})^{-1}$  and the lattice density of electrons  $n_{Lattice}=-2/\pi\int_{-\infty}^{0}\sum_{k}\mathrm{Im}G_{Lattice}(k,\omega)=-2G_{Lattice}(\tau=\beta^{-})$  as predicted from reconstructed correlation functions with ML obtained Legendre polynomial coefficients. We estimate Z from a quadratic fit to the values of the reconstructed self-energies at the three lowest Matsubara frequencies (see for example [32] for why Z can be estimated on imaginary axis). As easily seen from Fig. 11 of [31], values of  $G(i\omega_n)$  (or  $\Sigma(i\omega_n)$ )

types of functions.

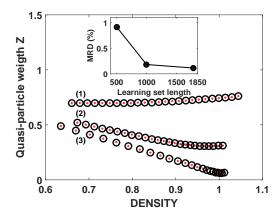


Figure 2: (Color online) Machine Learning predicted quasiparticle weight Z (black circles) as compared to the exact results (red dots) as a function of filling of the impurity for different U and t' (1) U=0.64 t'=-0.3t, (2) U=1.44t'=-0.1t, (3) U=2.08 t'=0. Inset: Median relative difference as a function of the size of the learning set

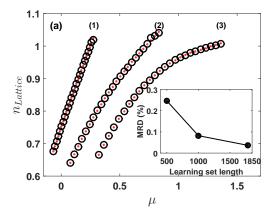


Figure 3: (Color online) Machine Learning predicted lattice density (black circles) as compared to the exact results (red dots) as a function of the chemical potential ( $\mu$ ) for different U and t' (1) U=0.64 t'=-0.2t, (2) U=1.44 t'=0, (3) U=2.08 t'=-0.1t (axis shifted  $\mu+0.2$ ). Inset: Median relative difference as a function of the size of the learning set

for the first few  $\omega_n$  are given solely by the first few coefficients of the expansion in Legendre polynomials. Hence, the prediction of Z shows how well the first few coefficients are learned. The results are shown in Fig. 2 for typical values of interaction from weak to correlated metals and for different t'. The predictions for these specific  $\mathbf{D}$  from the database are obtained by using all other examples as the training set. The predictions are in general very good with a slightly worst predicting power for larger correlation close to half-filling where Z is close to zero. To study the error in a more rigorous way, we present in inset of Fig. 2 what we call the median relative

2.5

difference (MRD) for Z as a function of the learning set size (see supplemental material Section VI for details). This shows the median value of predictions for fifty different random examples each re-calculated with twenty different random learning set. As shown in the inset of Fig. 2, the MRD of Z is slightly below 1% for a small size of 500 and gets to around 0.1% for the largest learning set. A predictive power of smaller than 1% error even for a small database is very interesting especially since choosing completely random datasets is the worst case scenario.

In the case of the lattice density of electrons as a function of chemical potential, the ML path from Fig. 1 is the one where we learn the  $G_l$ 's of the expansion of the local lattice Green's function then reconstruct it in imaginary time. Since  $P_l(1) = 1$  for all l, the density is  $n_{Lattice} = -\frac{2}{\beta} \sum_{l=0}^{\infty} \sqrt{2l+1} G_l$ . Therefore, contrary to the case of Z', the prediction of the density uses all predicted coefficients of the expansion. We show results in Fig. 3 for typical parameters, yet different than those presented for Z. To improve readability, we shifted curve (3) by 0.2. Once again the results are in good agreement with slightly worst predictions for  $n_{Lattice} > 1$ . This region tends also to be more problematic for Z. This is not fundamental but rather because our DMFT database is not as well constructed there. In the inset of Fig. 3 we show the MRD calculated the same way as for Z. ML does even better in this case where the MRD is at worst  $\sim 0.25\%$ .

We now show in Fig. 4-(a) and -(b) the prediction of the imaginary part of the impurity Green's function in Matsubara frequency for two typical set of parameters. As can be seen, ML does a very good job at predicting both the metal and the insulator, although the number of insulating solutions in the database is not very large. In the inset of Fig. 4-(a), we present the average relative difference (ARD) for the metallic case as a function of the size of the learning set. The ARD was defined in [22] as a way to measure on average the accuracy of the prediction of a full function using only one number. The values are obtained by averaging predictions for many random test sets. The global average prediction of a full function in the metallic case has an error in the worst case of  $\sim 0.8\%$  which shows the predictive power of our ML scheme.

We finally analyse the question of prediction of a totally new problem and the importance of training. Because our database is very homogeneous, for out of database predictions, we chose to use a width (arbitrarily set to be 5W=10) larger than the actual lowest possible error in the cross-validation training used for previous results to avoid overfitting. In the supplemental material (Section III.B.), we show how overfitting influences the predictive power of our ML approach. We show in Fig. 5 that indeed we can very well predict DMFT solution for new problems sharing no equal values of U, t' and  $\mu$  in the database by choosing as an example t' = -0.16t, U = 2

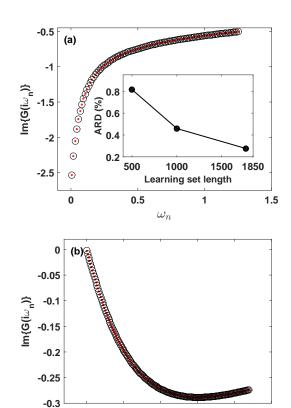


Figure 4: (Color online) Machine Learning predicted impurity Green's function (black circles) as compared to the exact results (red dots) (a) U=2.24, t'=0,  $n_d\approx 0.92$ , (b) U=3.68 t'=-0.2t,  $n_d=1$ . (c) ARD as a function of the size of the learning set for the metallic phase.

 $\omega_n$ 

1.5

0

0.5

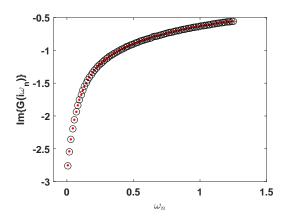


Figure 5: (Color online) Machine Learning predicted imaginary part of the impurity Green's function (black circles) as compared to the exact results (red dots) for U=2,  $t'=-0.16t,\,n_d\approx0.85.$ 

and  $\mu$  such that  $n_d \approx 0.85$ . We trained a machine with the full database of 1783 metallic solutions, the metal being well predicted by decision tree classification. By this process (larger width), we loose some of the predictive precision we had for intra-database testing sets, but this is of no consequences as once we prove that we can accurately train a machine, what really matters is the predictive power for out of database unsolved problems.

In this paper, we have investigated how machine learning can be used in many-body physics as a method to predict correlation functions. We applied the scheme to DMFT and showed that we can accurately predict its solutions. Our approach maps input functions to output functions and can be applied without any changes as an impurity solver for DMFT rather than to learn the fully converged solution or for any other cluster embedding theory with a self-consistency relation. Impurity solving might be the best way to use ML for real materials predictions since accuracy depends largely on having a large database. It is also general enough to be applied to other problems where learning a function is important. The learning of a function using Kernel Ridge Regression might be improved by adding a simple form of constraints in the minimization problem at small computational cost[33]. In real materials applications a more complicated system has to be taken into account; Hunds coupling, multi-band etc. However, our presented approach for ML is general enough to be adapted.

L.-F.A. was supported by the Office of Science of the U.S. Department of Energy under Subcontract No. 3F-3138 and A.J.M. by DOE FG-ER04169. L.-F.A. thanks Ara Go for numerous discussions on implementing an exact diagonalization code and Alejandro Lopez-Bezanilla for helpful discussions.

- B. Schölkopf and A. J. Smola, Learning with Kernels (MIT Press, 2002).
- [2] Bobby G. Sumpter and Donald W. Noid, Chem. Phys. Lett. 192, 455 (1992).
- [3] S. Lorenz, A. Gross and M. Scheffler, Chem. Phys. Lett. 395, 210 (2004).
- [4] S. Manzhos and T. Carrington, Jr., J. Chem. Phys. 125, 084109 (2006).
- [5] J. Behler and M. Parrinello, Phys. Rev. Lett. 98, 146401 (2007).
- [6] Albert P. Bartók, Mike C. Payne, Risi Kondor and Gábor Csányi, Phys. Rev. Lett. 104, 136403 (2010).
- [7] Zhenwei Li, James R. Kermode and Alessandro De Vita, Phys. Rev. Lett. 114, 096405 (2015).
- [8] J. C. Snyder, M. Rupp, K. Hansen, K-R. Müller and K. Burke, Phys. Rev. Lett. 108, 253002 (2012).
- [9] A. Lopez-Bezanilla and O. A. von Lilienfeld, Phys. Rev. B 89, 235411 (2014).
- [10] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K.-R. Müller and E. K. U. Gross, Phys. Rev. B 89, 205118 (2014).

- [11] F. Faber, A. Lindmaa, O. A. von Lilienfeld and R. Armiento, Int. J. Quantum Chem. in print 2015.
- [12] J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. S. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. M. Brockway, and A. Aspuru-Guzik, The Jour. of Phys. Chem. Lett. 2, 2241 (2011)
- [13] M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld, Phys. Rev. Lett. 108, 058301 (2012).
- [14] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld, New Journal of Physics 15, 095003 (2013).
- [15] R. Ramakrishnan, P. O. Dral, M. Rupp and O. A. von Lilienfeld, J. Chem. Theory Comput. 11, 2087 (2015).
- [16] P. O. Dral, O. A. von Lilienfeld and W. Thiel, J. Chem. Theory Comput. 11, 2120 (2015).
- [17] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp and A. Knoll, Int. J. Quantum Chem. in print (2015).
- [18] R. Ramakrishnan and O. A. von Lilienfeld, CHIMIA Int. J. for Chem. 69, 182 (2015).
- [19] T. Bereau, D. Andrienko and O. A. von Lilienfeld, J. Chem. Theory Comput. in print (2015).
- [20] R. Ramakrishnan, M. Hartmann, E. Tapavicza and O. A. von Lilienfeld, arXiv:1504.01966v1
- [21] J. K. Freericks, B. K. Nikolić, and O. Frieder, Int. J. Mod. Phys. B 28, 1430021 (2014).
- [22] L.-F. Arsenault, A. Lopez-Bezanilla, O. Anatole von Lilienfeld and A. J. Millis, Phys. Rev. B 90, 155136 (2014).
- [23] L. Li, J. C. Snyder, I. M. Pelaschier, J. Huang, U.-N. Ni-ranjan, P. Duncan, M. Rupp, K.-R. Müller and K. Burke, arXiv:1404.1333v2.
- [24] K. Vu, J. Snyder, L. Li, M. Rupp, B. F. Chen, T. Khelif, K.-R. Müller and K. Burke, Int. J. Quantum Chem. in print (2015).
- [25] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg Rev. Mod. Phys. 68, 13 (1996).
- [26] E. Dagotto, Rev. Mod. Phys. 66, 763 (1994).
- [27] M. Caffarel and W. Krauth, Phys. Rev. Lett. 72, 1545 (1994).
- [28] Corinna Cortes and Vladimir Vapnik, Machine Leaming 20, 273 (1995).
- [29] J. Hertz, A. Krogh and R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Reading, 1996.
- [30] Antonio Criminisi, Jamie Shotton and Ender Konukogly, Foundations and Trends in Computer Graphics and Vision 7, 81 (2011).
- [31] L. Boehnke, H. Hafermann, M. Ferrero, F. Lechermann, and O. Parcollet, Phys. Rev. B 84, 075145 (2011).
- [32] L.-F. Arsenault, P. Sémon and A.-M.S. Tremblay, Phys. Rev. B 86, 085133 (2012).
- [33] L.-F. Arsenault and A. J. Millis, to be published.

#### Supplemental Material for Machine learning for many-Body physics: efficient solution of dynamical mean-field theory

## I. DENSITY OF STATES FOR THE SIMPLE CUBIC LATTICE

The dispersion relation for the single band tightbinding simple cubic lattice with nearest and nextnearest hopping is given by

$$\varepsilon_k = -2t \sum_{\alpha=1}^{3} \cos(k_\alpha) - 4t' \Big[ \cos(k_1) \cos(k_2) + \cos(k_1) \cos(k_3) + \cos(k_2) \cos(k_3) \Big],$$

$$(1)$$

where  $-\pi \leq k_{\alpha} \leq \pi$   $\alpha = 1, 2, 3$  labels the three Cartesian directions of the nearest neighbor bonds of the cubic lattice. The bandwith  $W = \varepsilon_{[\pi,\pi,\pi]}$  or  $[0,\pi,\pi] - \varepsilon_{[0,0,0]}$  is given by

$$W = \begin{cases} 12t & \text{for } |t'| \le t/4, \\ 8t + 16|t'| & \text{for } |t'| > t/4. \end{cases}$$
 (2)

As mentioned in the main text, we define the energy unit by fixing W=2. This fixes the value of t for the different t'. Our database contains data for t'=[0,-0.1t,-0.2t,-0.3t]. We also tested our predictive power by using a lattice with t'=-0.16t. We show in Fig. 1. the density of states

$$N_0(\omega) = \sum_k \delta\left(\omega - \varepsilon_k\right) \tag{3}$$

of these five lattices to show the effect of next neighbor hopping.

### II. DETAILS OF THE EXACT DIAGONALIZATION SOLVER

### A. Fitting of the non-interacting hybridization function

In exact diagonalization, the bath is replaced by a finite number of sites  $(N_b)$ , each characterized by an onsite energy  $\varepsilon_l$  and hybridization with the impurity  $V_l$ . Therefore, the hybridization function is given by

$$\Delta_{ED}(i\omega_n) \equiv \sum_{l=1}^{N_b} \frac{V_l^2}{i\omega_n - \varepsilon_l} \Rightarrow \{\varepsilon_l, V_l\}, \tag{4}$$

which represents in real frequency the approximation of replacing a continuous function by a sum of poles and strengths. These poles and strengths  $\{\varepsilon_l, V_l\}$  have to be

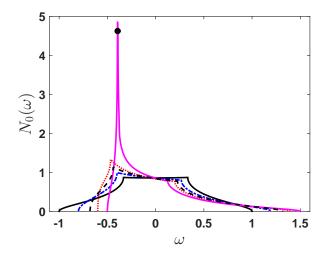


Figure 1: (Color online) Non-interacting density of states for the three dimensional simple cubic lattice with different values of the next nearest neighbor hopping t'=0 (black solid curve), t'=-0.1t (blue dash-dot curve), t'=-0.16t (black dashed curve), t'=-0.2t (red dot curve) and t'=-0.3t (magenta solid curve and filled black circle).

found by fitting Eq. (4) to the target function  $G_{imp,0}$  obtained via the lattice Green's function  $G_{Lattice}^{-1}(i\omega_n) = G_{imp,0}^{-1}(i\omega_n) - \Sigma(i\omega_n)$ . This is achieved by defining a distance function and minimizing it.

$$d = \frac{1}{N+1} \sum_{n=0}^{N} W(\omega_n) \left| G_{imp,0}^{-1}(i\omega_n) - G_{imp,0}^{-1,Ns}(i\omega_n) \right|^2,$$
(5)

where the function  $W(\omega_n)$  is chosen to give more weight to some frequencies if wanted. We use  $W(\omega_n) = \frac{1}{\omega_n}$  to have a better fit of the low frequencies. N is the maximum number of frequency used to define d. What is important in its choice is that  $\omega_{N_{max}} \gg max(\varepsilon_l)$ . Finally,  $G_{imp,0}^{-1,Ns}(i\omega_n)$  is the inverse non-interacting Green's function of the Hamiltonian with a finite number of bath sites and is written as

$$G_{imp,0}^{-1,Ns}(i\omega_n) = z + \mu - \sum_{l=1}^{N_b} \frac{V_l^2}{z - \varepsilon_l}$$
 (6)

Therefore, we need to find the set of parameters  $\{\varepsilon_l, V_l\}$  that minimizes d. This is a problem of unconstrained optimization in several variables. In DMFT, this is done as many time as necessary to converge the solution. We can define a bare hybridization function, for U=0, a function containing the information about the crystal structure and chemistry of the problem. For simplicity, we choose to fix  $\mu=0$ . For the non-interacting case,  $\Sigma(i\omega_n)=0$  and the lattice Green's function is given by the band dis-

persion only

$$G_{Lattice}^{0}(i\omega_n) = \sum_{k} \frac{1}{i\omega_n - \varepsilon_k}.$$
 (7)

We can therefore define the bare hybridization function

$$\Delta^{0}(i\omega_{n}) = i\omega_{n} - \left(\sum_{k} \frac{1}{i\omega_{n} - \varepsilon_{k}}\right)^{-1}.$$
 (8)

This can be fitted to  $\{\varepsilon_l^0, V_l^0\}$  using Eq. (5) and gives a representation of the crystal of size equal to  $2N_b$  where  $N_b$  is the number of bath sites. The function in this case is thus represented in this basis as  $\Delta^0(i\omega_n) \to \mathbf{f} = (\varepsilon_1^0, \varepsilon_2^0, \dots, \varepsilon_{N_b}^0, V_1^0, V_2^0, \dots, V_{N_b}^0)$ . It could perhaps be seen as a general and compact way to describe the lattice for ML, irrespective of how the database is constructed (using ED or not). After all, what is needed for creating a  $\mathbf{D}$  is a unique way to describe a problem and  $[\{\varepsilon_l^0, V_l^0\}, U, \mu]$  provides one. This is true for both for model Hamiltonians and real materials as well as for solutions obtained from ED, quantum Monte Carlo etc.

#### B. Particularities of the insulating state in ED

The database contains 1783 converged DMFT standard metal solutions, 218 converged critical metal solutions and 494 Mott insulating solutions, obtained from exact diagonalization solutions of the DMFT equations. In the Mott insulating state, physically, any choice of chemical potential in the gap should lead to the same solution with a shifted zero point of the frequency axis. However, the bath discretization of the ED method means that for otherwise identical parameters different values of  $\mu$  lead to different insulating solutions. For this reason we need to include many (here 494) different Mott insulators in the database. We train insulators using all these solutions.

#### III. CROSS VALIDATION

#### A. Principle and process

In kernel ridge regression (KRR), there are two free parameters or hyperparameters. In our case, for K, we use the weighted exponential kernel

$$K(\mathbf{D}_i, \mathbf{D}) = e^{-\frac{|\mathbf{d}_i|}{\sigma}},\tag{9}$$

where  $|\mathbf{d}_l| = |\mathrm{D}_{l1} - \mathrm{D}_1| + |\mathrm{D}_{l2} - \mathrm{D}_2| + \dots$  is the Manhattan distance between the two parameter sets in descriptors space and  $\sigma$  gives the radius of effect that a particular point of the data set  $\mathbf{D}_l$  will have in the prediction process. Therefore, the two free parameters are  $\sigma$  (entering

the kernel function) and  $\lambda$ , the regularization parameter used in the cost function that is minimized in KRR. To fix them, we use cross-validation. Cross validation proceeds by first creating a large number of pairs  $[\sigma, \lambda]$ . Then, for each pair we randomly split the database so that the testing set contains about ten examples. For each of these ( $\sim 10$ ) tests, we predict only the first five Legendre polynomials coefficients  $G_{l=0...4}$  and calculate the total mean absolute error. Note that the actual metric to calculate the error is not really important, we are just looking for the pair that minimizes an error. We look for a pair  $[\sigma, \lambda]$  that gives as small as possible error, then this  $[\sigma, \lambda]$  is used to learn all the necessary Legendre polynomials coefficients and not only the first five. As an example, we show the result for the training of the metallic impurity Green's function as a contour plot in Fig. 2 when the descriptor uses ED representation for the non-interacting hybridization function and Fig. 3 when the Legendre representation for the non-interacting hybridization function is chosen. The white color indicates regions where no data are available. We see that if the width  $\sigma$  of the kernel is too small (ML will use not enough solutions in descriptors spaces) or too large (ML will use too many far away solutions in descriptors spaces), the error is the largest. For the training/testing using solved problems from our database, the best possible  $\sigma$  would be around but smaller than  $\sigma = 1$  for Fig. 2 and around  $\sigma = 1.5$  for Fig. 3. The results also show that the value of  $\lambda$  is not extremely important. Practically we chose the pair  $[\sigma, \lambda]$  that produced the smallest error among the ones tested while we increase the  $\sigma$  to prevent overfitting in the case of out of database prediction (see Section III.B below).

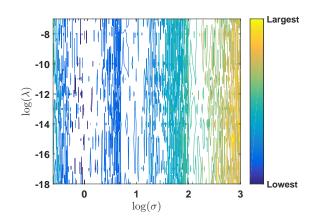


Figure 2: (Color online) Contour plot showing the mean absolute error for the first five Legendre polynomials coefficients as a function of different hyperparameters pair  $[\sigma, \lambda]$  in cross-validation for the impurity Green's function when the descriptor is chosen to be  $\mathbf{D} = [\{\varepsilon_l^0, V_l^{0}\}, U, \mu]$ .

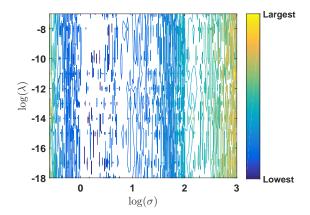


Figure 3: (Color online) Contour plot showing the mean absolute error for the first five Legendre polynomials coefficients as a function of different hyperparameters pair  $[\sigma, \lambda]$  in cross-validation for the impurity Green's function when the descriptor is chosen to be  $\mathbf{D} = [\{\Delta_l^0\}, U, \mu]$ .

#### B. Overfitting and prediction for a new problem

Here we discuss in detail the prediction of the DMFT solution for the case t' = -0.16t, U = 2 and  $n_d \sim 0.85$ which is a case completely outside our database. Let us first use the width of the kernel that gave the lowest error during cross-validation using  $\mathbf{D} = [\{\varepsilon_l^0, V_l^0\}, U, \mu]$  as the descriptor which is around  $\sigma = 1$  as shown is Section III.A. The result for the imaginary part of the reconstructed impurity Green's function is shown in Fig. 4-(a) while Fig. 4-(b) shows the ML predicted coefficients  $G_l$ . We see a systematic discrepancy in the prediction of the reconstructed  $\text{Im}\{G(i\omega_n)\}\$ , but it is interesting to realize that this is due to a systematic discrepancy of  $\sim 2\%$  only on the even  $G_l$  coefficients at low order. This is true for every possible example at t' = -0.16t(we calculated a total of 295 different examples with this t'). The discrepancy is not due to a failing of ML but solely on a too tight choice of kernel width (over-fitting) which for example might not use enough of the solutions for t' = -0.1t and/or t' = -0.2t. We trained our machine again, but with a larger width of  $\sigma = 10$  which, according to Fig. 2 is not the optimal value but still give a pretty acceptable error. The result is Fig. 5 of the main text and is reproduced here as Fig. 4-(c) showing that we must be careful with over-fitting and if this is properly taken into account, we can predict new DMFT solutions. The specific choice of  $\sigma$  is not well defined since out of database predictions means no comparison with exact results in principle. A sensible approach is to start from the  $\sigma$  given by cross-validation of Section III.A and search for a larger value. To do so, contour plots of Figs. 2 and 3 are essential. We need a value to the right of the cross-validated one since we are looking for a larger

radius of effect for database points. How large will be given by the condition of still keeping a low enough error for intra-database prediction. Hence by looking at Figs. 2 and 3 we see that values of  $\sigma$  somewhere between  $\log(\sigma)=0.5$  and  $\log(\sigma)=1$  respect these conditions. We chose  $\log(\sigma)=1$ , but we could have tested many other values.

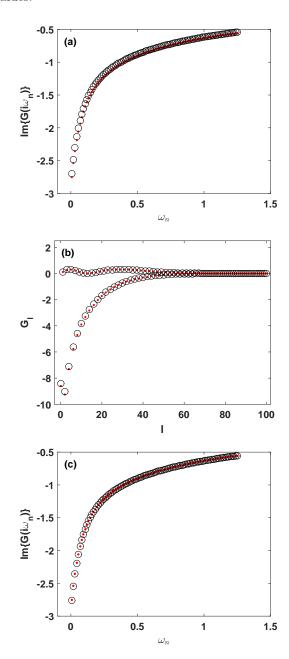


Figure 4: (Color online) Machine Learning predicted impurity Green's function (black circles) with descriptor  $\mathbf{D} = [\{\varepsilon_l^0, V_l^0\}, U, \mu]$  as compared to the exact results (red dots) for  $U=2, \ t'=-0.16t, \ n_d\approx 0.85$  (a) Prediction of  $\mathrm{Im}\{G(i\omega_n)\}$  with a too small (over-fitted) width of kernel (b) Prediction of  $G_l$  with a too small (over-fitted) width of kernel (c) Prediction of  $\mathrm{Im}\{G(i\omega_n)\}$  with a right width of kernel.

### IV. OBTAINING THE LEGENDRE COEFFICIENTS

We expand the output solution in terms of Legendre polynomials. Considering the result as a function of imaginary time  $0 < \tau < \beta$  ( $\beta = 1/T$  is the inverse temperature) we have for correlation functions [1]

$$f(\tau) = \sum_{l=0}^{\infty} \frac{\sqrt{2l+1}}{\beta} f_l P_l(x(\tau)). \tag{10}$$

The ED method we used to do DMFT calculations provides correlation functions in energy space (real or Matsubara frequency); we must therefore Fourier transform the result. To perform the Fourier transform we note that the functions  $\Delta(i\omega_n)$ ,  $G_{imp}(i\omega_n)$ ,  $G_{Lattice}(i\omega_n)$  and  $\Sigma(i\omega_n)$  of interest here all decay as  $C/i\omega_n$  at large  $|\omega_n|$ . Therefore, let us define the general Matsubara frequency function  $f(i\omega_n)$  with asymptotic behaviour  $f(i\omega_n)_{n\to\infty}\to \frac{C}{i\omega_n}$ . Depending on the specific correlation function, the constant C is given by

$$C = \begin{cases} 1 & \text{for } G, \\ U^{2} \frac{n_{d}}{2} \left( 1 - \frac{n_{d}}{2} \right) & \text{for } \Sigma, \\ \sum_{k} \varepsilon_{k}^{2} = \sum_{l} V_{l}^{2} & \text{for } \Delta. \end{cases}$$
(11)

The imaginary time function is given by Fourier transform  $f(\tau) = T \sum_n e^{-i\omega_n \tau} f(i\omega_n)$ . Direct numerical evaluation of the sum is complicated by the slow  $1/\omega_n$  decay. We therefore treat the  $1/\omega_n$  term analytically:

$$f(\tau) = T \sum_{n} e^{-i\omega_{n}\tau} \left[ f(i\omega_{n}) - \frac{C}{i\omega_{n}} + \frac{C}{i\omega_{n}} \right]$$

$$= 2T \sum_{n>0} \left[ \operatorname{Re} \{ f(i\omega_{n}) \} \cos(\omega_{n}\tau) + \left( \operatorname{Im} \{ f(i\omega_{n}) \} + \frac{C}{i\omega_{n}} \right) \sin(\omega_{n}\tau) \right]$$

$$- \frac{C}{2}.$$
(12)

This way, we can obtain the value of  $f(\tau)$  for any desired  $\tau$ . Therefore, we can compute the coefficients of the expansion in Legendre polynomials as was done in [2] by using the algorithm based on Chebyshev-Legendre transform [3] exploiting the idea that smooth functions can also be represented by expansions in Chebyshev polynomials using fast Fourier transform. This algorithm is implemented in a free MATLAB toolbox called CHEB-FUN [4].

Once the coefficients have been obtained, not only the function in imaginary time can be reconstructed (10), but the Fourier transform to  $f(i\omega_n)$  can also be done analytically [1] to give

$$f(i\omega_n) = \sum_{l=0}^{\infty} T_{nl} f_l, \tag{13}$$

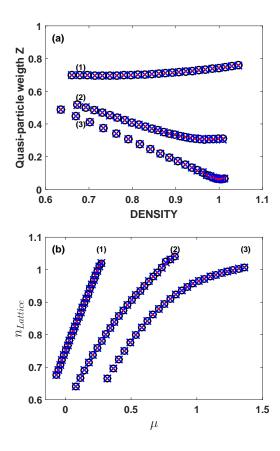


Figure 5: (Color online) Machine Learning predictions (black circles) for descriptor  $\mathbf{D} = [\{\varepsilon_l^0, V_l^0\}, U, \mu]$  and (blue x) for descriptor  $\mathbf{D} = [\{\Delta_l^0\}, U, \mu]$  as compared to the exact results (red dots). (a) Quasi-particle weight Z as a function of filling of the impurity for different U and t' (1) U = 0.64 t' = -0.3t, (2) U = 1.44 t' = -0.1t, (3) U = 2.08 t' = 0. (b) Lattice density as a function of the chemical potential ( $\mu$ ) for different U and t' (1) U = 0.64 t' = -0.2t, (2) U = 1.44 t' = 0, (3) U = 2.08 t' = -0.1t (axis shifted  $\mu + 0.2$ ).

where  $T_{nl} = (-1)^n i^{l+1} \sqrt{2l+1} j_l \left(\frac{(2n+1)\pi}{2}\right)$  and  $j_l(z)$  are the spherical Bessel functions.

# V. CHOICE OF REPRESENTATION FOR THE BARE HYBRIDIZATION FUNCTION $\Delta^0(i\omega_n)$

In the main text we claimed that the results we presented hold irrespective if the chosen representation of the bare hybridization function is from an ED-like fitting  $\{\varepsilon_l^0, V_l^0\}$  or as in term of Legendre polynomials  $\{\Delta_l^0\}$  expansion. We show here two examples to support our statement. In Fig. 5-(a) and (b), we reproduce Fig. 2-(a) and Fig. 3-(a) of the main text. In addition, we add the predictions (blue x) as obtained from the descriptor of the form  $\mathbf{D} = [\{\Delta_l^0\}, U, \mu]$ . The predictions for the quasi-particle weight are very close to the case when the machine is trained with  $\mathbf{D} = [\{\varepsilon_l^0, V_l^0\}, U, \mu]$ . Other op-

timizations could be done to get even closer to the exact solutions and thus for intended purpose it can be considered as being equal. The case of the lattice density is similar, the two representations give again practically the same answer. In conclusion, the two choices of representation for the bare hybridization function in the descriptor give essentially the same results and therefore its choice is a matter of which one can be obtained and how easy to calculate it is for a particular situation.

#### VI. MEDIAN RELATIVE DIFFERENCE

We present exactly how we calculated the median relative difference for the quasi-particle weight. The same approach is taken for the density. Its meaning is the following: 1) A size  $N_{learning}$  of learning set is chosen. 2) From the database, we select randomly one example that will be the testing system. 3) From the remaining examples in the database, we randomly take  $N_{learning}$  solutions and train a machine (calculate the  $\alpha$  matrix of KRR). 4) We predict the self-energy for the testing example of 2) using our trained machine. From it, we obtain Z and can calculate the relative difference with the exact answer  $100\frac{|Z_{ML}-Z_{exact}|}{|Z_{exact}|}$ . We then repeat steps 3) and 4) twenty times to obtain the prediction of the same example from many different trained machines and thus assure we have large distributions of learning sets. Finally, we

go back to step 2) and start again with a new example and do it fifty time in total to have a large distribution of testing examples. We therefore have one thousand relative difference. If a randomly chosen learning set only contains examples very far from the testing example we are predicting, the relative difference will be large, but in that case, not because ML is bad, but rather because the learning set was badly picked. Therefore to have a good idea of how well a learning set of size  $N_{learning}$  does, we argue that a good choice is to take the median value of the one thousand predictions. For the largest possible size of learning set 1782, we instead calculated the relative difference for every example in the database choosing the reminder 1782 examples as learning set. Then we randomly choose fifty of these relative difference and calculated the median.

- L. Boehnke, H. Hafermann, M. Ferrero, F. Lechermann, and O. Parcollet, Phys. Rev. B 84, 075145 (2011).
- [2] L.-F. Arsenault, A. Lopez-Bezanilla, O. Anatole von Lilienfeld and A. J. Millis, Phys. Rev. B 90, 155136 (2014).
- [3] N. Hale and A. Townsend, SIAM J. Sci. Comput. 36, A148 (2014).
- [4] T. A. Driscoll, N. Hale, and L. N. Trefethen (eds.), Chebfun Guide (Pafnuty Publications, Oxford, 2014).