Fast evaluation of interaction integrals for confined systems with machine learning

A. Mreńca-Kolasińska, K. Kolasiński, and B. Szafran
AGH University of Science and Technology,
Faculty of Physics and Applied Computer Science,
al. Mickiewicza 30, 30-059 Kraków, Poland

The calculation of interaction integrals is a bottleneck for the treatment of many-body quantum systems due to its high numerical cost. We conduct configuration interaction calculations of the few-electron states confined in III-V semiconductor two-dimensional structures using a shallow neural network to calculate the two-electron integrals, which can be used for general isotropic interaction potentials. This approach allows for a speed-up of the evaluation of the energy levels and controllable accuracy.

I. INTRODUCTION

Computation of quantum problems for interacting particles is a challenging task that has been dealt with using various approaches, including machine learning (ML)\cite{11-13}. The difficulty of accurate many-body calculations arises from the high dimensionality of the space required when taking into account the electron-electron and electron-ion interactions. Methods developed to solve the many-body quantum problems include the density functional theory (DFT)\cite{12} Hartree-Fock method\cite{15,16}, and configuration interaction (CI) method\cite{17,19}. These methods contain the computation of the non-local potentials due to the particle-particle interaction. This is a computationally demanding task because of its $O(n_x^2)$ nature, with $n_x$ being the mesh size. A number of approaches trying to improve the scalability of the non-local potential computation exist, with probably the most common being methods based on the Fourier transform which allow us to reduce the computational complexity to $O(n_x \log(n_x))$.

The problem of the nonlocal potential calculation has been tackled using plane-wave functions and Gaussian-sum (GS) approximations\cite{21-23}. The authors of Ref.\cite{22} developed a method of calculating the kernel via expansion of the density in terms of scaling functions and approximation of the $1/r$ potential in terms of Gaussian functions, which served to avoid the costly three-dimensional integral of the original kernel. Reference\cite{23} proposed an approach to improve the accuracy based on the GS approximation of the kernel with a near-field correction added to account for the discrepancy between the GS-approximated and original kernels.

On the other hand, calculations for complex quantum systems were done using machine-learning methods. The problem was solved through the variational quantum Monte Carlo method\cite{26} or using ML for effective models\cite{11,12} which are also used in self-learning Monte Carlo methodologies\cite{11,13}.

In this work we propose an approach to solve the few-electron problem via the CI method. The need to calculate a huge number of Coulomb integrals\cite{22,23} is a main bottleneck of this method. Although the number of integrals can be reduced by taking into consideration symmetries of the problem and building a basis out of functions that satisfy some constraints (e.g., have the necessary spatial symmetry or spin), the calculation time of Coulomb integrals prevails in this problem, especially in two or more dimensions. The methods of the calculation of the nonlocal potential developed in Refs.\cite{21,23} mostly aim to obtain the best precision of the calculations. On the other hand, our objective is to develop an approximate and fast method which can be used to evaluate the energies of few-electron states with a precision sufficient to describe the quantum phenomena in mesoscopic systems.

In this paper we develop a method to calculate the two-electron integrals based on a shallow neural network\cite{27} which can be used for any interaction potential that is isotropic. We present the application of our approach for Coulomb as well as non-Coulomb potentials in one or two dimensions. This method can be extended to three dimensions as well. The source code for the implementation of the proposed method is available online\cite{28}.

II. METHODS

A. Hamiltonian

We consider $N$ electrons confined in a one- or two-dimensional III-V semiconductor nanostructure. The Hamiltonian, within the effective-mass approximation, is

$$H = \sum_{i=1}^{N} h_i + \sum_{i=1}^{N} \sum_{j>i}^{N} u(r_i - r_j),$$  \hspace{1cm} (1)

where $u(r_i - r_j)$ is the interaction potential between the $i$th and $j$th electron and $h_i$ is the single-electron Hamiltonian

$$h_i = -\frac{\hbar^2}{2m^*} \nabla_i^2 + V(r_i),$$  \hspace{1cm} (2)

with $V(r_i)$ being the external potential at position $r_i$. The interaction potential may have various forms, depending on the dimensionality of the system. In thin insulating layers it gets an effective form different from the Coulomb interaction potential due to screening\cite{29}, and for
quasi-one-dimensional systems it has the form of the exponentially scaled complementary error function. We will focus on isotropic interaction potentials that satisfy $u(r_i - r_j) = u(|r_i - r_j|)$.

The calculation is performed using the CI method within the basis set formed by Slater determinants constructed from the one-electron wave functions

$$
\Psi_n(r_1, \ldots, r_N) = \sum_k c_k(n) \Phi_k(r_1, \ldots, r_N)
$$

As the single-electron wave functions $\phi_k$ we use the eigenfunctions of the $\hat{h}$ operator in the first-order finite-difference approximation on a mesh of points. The coefficients $c_k^{(n)}$ are found by solving the eigenproblem of the matrix with the elements given by

$$
H_{pq} = \langle \Phi_p | H | \Phi_q \rangle.
$$

Using the Slater-Condon rules, one can show that the matrix elements $H_{pq}$ can be represented in terms of the one-electron and two-electron integrals:

$$
\langle i | h | j \rangle = \int dr_1 \phi_i^*(r_1) h(r_1) \phi_j(r_1),
$$

$$
\langle i | u | j \rangle = \int dr_1 dr_2 \phi_i^*(r_1) \phi_j(r_2) u(r_1 - r_2) \phi_i(r_1) \phi_j(r_2).
$$

We use a basis formed with Slater determinants built with a finite number of one-electron states $\phi_k$. Their number is determined by verifying, by the convergence of the energies of an $N$-electron system and given a basis containing $n$ states, we have $\langle \Psi | H | \Psi \rangle$ Slater determinants.

### B. Evaluation of the two-electron integrals

The two-electron integral can be written as

$$
\langle ij | u(r_1 - r_2) | kl \rangle = \int dr_1 dr_2 \phi_i^*(r_1) u_{ij}(r_1) \phi_k(r_2),
$$

where the effective potential $u_{ij}(r_1)$ is

$$
u_{ij}(r_1) = \int dr_2 \phi_j^*(r_2) u(r_1 - r_2) \phi_i(r_2).
$$

Denoting the complex function $\rho_{ij}(r_2) = \phi_j^*(r_2) \phi_i(r_2)$, one can reformulate the integral as a convolution operation

$$
u_{ij}(r_1) = \int dr_2 u(r_1 - r_2) \rho_{ij}(r_2) = (u * \rho_{ij})(r_1),
$$

where the asterisk ($*$) is the convolution operator and $u(r)$ is a nonlocal kernel, e.g., Coulomb interaction $u(r) \approx 1/r$.

#### 1. Evaluating integrals with fast Fourier transform

The integral in Eq. (8) can be efficiently evaluated using the fast Fourier transform (FFT) method available in many numerical libraries. For example, in the case of a one-dimensional grid of size $n_x$, the computational cost of evaluating a single integral with FFT is

$$
O((n_x + P) \log(n_x + P)),
$$

where $P$ is a padding which is added to both sides of the grid symmetrically. The size of the padding depends on the size of the convolutional kernel. For closed systems (i.e., systems for which $\rho_{ij} = 0$ outside the computational box) with long-range interactions, the size of the kernel is $2n_x + 1$, and the size of the padding is equal to the size of the computational box $P = n_x$. We use this approach as our baseline method. However, in the special case of short-range interactions, the convolution kernel may be truncated, and $P \ll n_x$. In such a case padding has a negligible effect on the computation time. From the above we can see that the evaluation time of the integral in Eq. (8) can be improved in two ways, (a) truncating the kernel size, which will reduce padding, or (b) using faster implementation of FFT.

In this paper we show that the long-range interaction kernel can be approximated by a series of finite-size kernels for which $P \ll n_x$, so that the total computational time is smaller than the baseline FFT implementation. Additionally, we show that using the existing neural network frameworks, we can improve further the computational cost using efficient computing unit (GPU) implementations of the convolution operator. The CPU implementation of the convolution is performed using the fast Fourier transform from the Intel® MKL library in the Fortran language and GPU implementation is provided in the Python language and TensorFlow framework.

#### 2. Approximating the integral in one dimension

Let $\rho$ be a discretized one-dimensional (1D) density array of size $n_x$. We define the downsampling operator

![FIG. 1. Schematic of the neural net trained to evaluate the two-electron integrals described by Eq. (9).](image)
\[ D_i \text{ as} \]

\[ \hat{\rho} = D_i \rho, \]

which reduces the spatial size of \( \rho \) by a factor of 2 in the \( i \)th step (for example, starting from \( s = 0 \) with \( n_x = 512 \), operator \( D_1 \) downsamples \( \rho \) to \( n_x = 256, D_2 \) downsamples \( \rho \) from 256 to 128, and so on). In our implementation we use a standard average pool operator, defined as

\[ \hat{\rho}_i = \frac{1}{2}(\rho_{2i} + \rho_{2i+1}), \quad i \in (1, n_x/2). \]

Similarly, we define the upsampling operator \( U_s \) as

\[ \hat{\rho} = U_s \rho, \]

which in the \( i \)th step resizes the input array back to size \( n_x \) (for example, for \( s = 0 \) it is an identity operation). In practice, the composition of the downsampling and upsampling operations results in an approximated identity operation

\[ U_s D_s D_{s-1} \cdots D_1 \rho \approx \rho. \]

For the upsampling operation we use the standard bilinear interpolation (e.g., resize_bilinear operator from the TensorFlow library\cite{33}). The role of the downsampling operation is to increase the receptive field.

We approximate the integral in Eq. (8) with the following definition of the linear neural network:

\[ u^{approx} = F_0 \rho + U_1 F_1 D_1 \rho + U_2 F_2 D_2 D_1 \rho + \cdots + U_{S-1} F_{S-1} D_{S-1} \cdots D_2 D_1 \rho \quad (9) \]

\[ = N_S(\rho), \]

where \( S \) denotes the number of scales used to approximate the long-range interaction. In each of the \( S \) steps, the network downsamples the density, performs a convolution with filter \( F_s \), upsamples the result back to the original mesh size, and sums all the contributions to get the approximate effective potential. Here we assume that \( n_x \) is a power of 2 or can be divided by 2 at least \( S - 1 \) times, and \( F_1 \) is the convolution operator at the \( S \)th scale with a learnable kernel of size \( K = 2n_x / 2^{S-1} + 1 \) (we keep the filter size odd). The kernel parameters are not shared between the scales. The schematic of the architecture is shown in Fig. 1. In the next section we describe how to find the optimal parameters \( N_S \).

In the special case with the number of scales \( S = 1 \) we have a single convolution with a kernel of size \( K = 2n_x + 1 \), and we recover the exact baseline method \( u^{approx} = F_0 \rho = u \) described in the previous section. If \( S > 1 \), the computations are no longer exact; however, as we show in Sec. 3, in such a case we can gain a significant improvement in the performance. Note that we do not use any nonlinear activation function in our neural network \( N_S \), Eq. (9); hence, it preserves the physically required charge superposition condition \( N_S(a \rho + b \rho') = a N_S(\rho) + b N_S(\rho') \).

In order to find the optimal kernel parameters of the neural network \( N_S \) we apply the widely used gradient descent (GD) method\cite{33}. We use the TensorFlow library\cite{33}, which uses the back-propagation method (i.e., chain rule) to compute the analytical value of the gradient of the loss function with respect to the network parameters. In the following we present the loss function and the methodology used to train the network\cite{33}.

The effective potential can be treated as a superposition of contributions from point like charges at each mesh point. We use this property to define the loss function for our problem. Given a point charge \( \rho_p(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_p) \) at position \( \mathbf{r}_p \), we get the exact solution for the integral (8), and the effective potential is given by the kernel function \( u_p(\mathbf{r}) = u(\mathbf{r} - \mathbf{r}_p) \). Substituting Dirac’s \( \delta \) for the density in Eq. (8), one obtains

\[ u_p(\mathbf{r}_1) = \int d\mathbf{r}_2 u(\mathbf{r}_1 - \mathbf{r}_2) \delta(\mathbf{r}_2 - \mathbf{r}_p) = u(\mathbf{r}_1 - \mathbf{r}_p). \quad (10) \]

To obtain the same result on a discrete grid we use Kronecker’s \( \delta \) instead of Dirac’s \( \delta \). Using this property, we train the network \( N_S \) to minimize the difference between the exact \( u_p \) discretized potential and the approximated one [Eq. (9)],

\[ \mathcal{L} = \frac{1}{n_x} \sum_{p=1}^{n_x} ||N_S(\rho_p) - u_p||^2, \quad (11) \]

where \( ||\mathbf{x}||^2 = \sum_i x_i^2 \) is the \( L_2 \) norm of vector \( \mathbf{x} \) and the sum in the above equation runs over all grid sites.

In order to minimize Eq. (11) we use the standard gradient descent using the basic momentum GD optimizer with a decaying learning rate. We decay the learning rate \( \lambda \) by a factor \( \epsilon \) every \( N_{it} \) gradient updates. The hyperparameters \( \lambda, \epsilon \), and \( N_{it} \) are obtained semi-automatically via a grid search. \( N_{it} \) is chosen to be of the order of 2000, and we train the by kernels varying \( \lambda \) and \( \epsilon \) in discrete
steps and find their combination that yields the lowest loss function. Note that once the model is trained, it can be reused in many problems assuming that the grid size or the estimated kernel does not change.

Figure 2 shows the loss function throughout the training for the Coulomb interaction potential on a mesh of \( n_x = 256, S = 4, K = 65 \), and initial \( \alpha = 0.003 \). The abrupt drop at each multiple of \( N_{it} = 1000 \) steps occurs when the learning rate is decreased by \( \epsilon = 0.5 \). The potential evaluated by the integral (7) with the baseline method is shown in the inset of Fig. 2. The difference between the baseline result and the potential obtained with the trained kernels is shown by the black line. The error is scaled by a factor of 100 to be visible, but the approximated potential is close to the baseline. A more quantitative assessment of the accuracy of our method will follow in Sec. III.

4. Isotropic potentials in two dimensions

In the case of isotropic potentials a one-dimensional kernel \( k^{1D}_{ij} \) array obtained from the method described in the previous section can be projected to two-dimensional (2D) Cartesian coordinates using the projection tensor \( R \),

\[
k_{ij}^{2D} = \sum_{l=1}^{K} R_{il} k^{1D}_{lj}, \quad i, j \in (1, K);
\]

for more information about the details of implementation of \( R \) see Ref. 28. Having computed the two-dimensional kernels \( k^{2D} \) from Eq. (12), we can use them to solve the two-dimensional integrals by replacing all the 1D operators in Eq. (9) by their 2D analog. A similar approach can be used to project the 1D kernel to three dimensions.

Figure 3 shows the best kernels at subsequent scales obtained from the training in Fig. 2. The kernels projected to two dimensions are shown in the bottom row of Fig. 3. We calculate 100 integrals with \( n_x \) scales, compared to the baseline. The average time of the computation for various mesh sizes and kernels obtained by our method. Figure 4 shows the average time for more information about the details of implementation of \( R \) see Ref. 28. Having computed the two-dimensional kernels \( k^{2D} \) from Eq. (12), we can use them to solve the two-dimensional integrals by replacing all the 1D operators in Eq. (9) by their 2D analog. A similar approach can be used to project the 1D kernel to three dimensions.

Figure 3 shows the best kernels at subsequent scales obtained from the training in Fig. 2. The kernels projected to two dimensions are shown in the bottom row of Fig. 3.

C. Benchmark

First, we determine the computation time using the kernels obtained by our method. Figure 4 shows the average time of the computation for various mesh sizes and numbers of scales, compared to the baseline. The time includes only the computation of the integrals, and not the training. We calculate 100 integrals with \( \rho \) and kernels filled with random values. The dashed lines show the average time \( T = 2 \) and 5 ms for reference. The computational time is shown for the calculation for a single thread and 40 threads, with the neural net implemented in Fortran, and for the calculation on a CPU and GPU for the TensorFlow implementation. We used the GPU GeForce GTX 1080 Ti.

The implementation on a GPU works faster for more scales. On the other hand, the MKL implementation is faster with a smaller number of scales, i.e., for the case that is potentially more precise.

III. APPLICATIONS

A. Two electrons in a harmonic potential

As a first application for the method we present the solution of a problem of two interacting electrons confined in a 2D harmonic potential

\[
V(r_i) = \frac{1}{2} m^* \omega^2 r_i^2,
\]

with the Coulomb interaction potential

\[
u(|r_i - r_j|) = \frac{\epsilon^2}{4\pi\varepsilon_0} \frac{1}{|r_i - r_j|}.
\]

We use the GaAs parameters \( m^* = 0.067m_e \) and \( \varepsilon = 12.4 \), where \( m_e \) is the electron mass. We solve the problem using the configuration interaction method with the Coulomb integrals calculated by (i) a convolution with an exact \( (2n_z+1) \times (2n_z+1) \) filter by Fourier transform and (ii) using our method.

This problem can also be solved for two electrons using the semi-analytical method described in Ref. 36. In the center-of-mass coordinates the Hamiltonian can be written in the form

\[
H = H_{cm} + H_{rel},
\]

where \( H_{cm} \) is the center-of-mass Hamiltonian and \( H_{rel} \) describes the relative motion of the electrons. \( H_{cm} \) is independent of the interaction, and the center-of-mass energy is \( E_R = \hbar \omega (n^x_R + n^y_R + 1) \), where \( n^x_R, n^y_R \) are the quantum numbers of the center-of-mass energy. Further noting that \( H_{rel} \) commutes with the \( z \) component of the angular momentum operator, one can write it in the cylindrical coordinates

\[
H_{rel,\rho} = -\frac{\hbar^2}{2\rho^2} \left( \frac{d^2}{d\rho^2} + \frac{1}{\rho} \frac{d}{d\rho} - \frac{M^2}{\rho^2} - \gamma^2 \rho^2 - \frac{1}{\rho} \right),
\]

with \( \gamma = \omega/2 \), which yields states with a well defined angular momentum quantum number \( M \). Equation (16) is written in donor units with energy in \( R_D = m_e^* \epsilon e^2/\ell^2 \), length in units of \( a_D = m_e^* \epsilon e^2/2\hbar^2 \), and \( \kappa = 1/4\pi\varepsilon_0 \). Further, \( E = E_{rel} + E_R \). We calculate \( E_{rel} \) using the shooting method (see Appendix A). The four lowest levels and their degeneracies \( d \) are given in Fig. 5.

For the CI method we take as a basis set \( n_{basis} = 20 \) spin orbitals. We solve the problem on an \( n_x \times n_x \) mesh. The results of the calculation with \( n_x = 64 \) are shown in Fig. 5 together with the results of the shooting method. The results of both methods agree very well. For completeness, we show the results of methods (i) and (ii) for three electrons in Fig. 6.
FIG. 3. The kernels obtained from the training in Fig. 2 using Coulomb interaction potential. Top row: the one-dimensional kernels; bottom row: the kernels after the projection to two dimensions [Eq. (12)].

(a) \( n_x=128 \)  
(b) \( n_x=256 \)  
(c) \( n_x=512 \)

FIG. 4. The calculation time averaged over 100 integrals as a function of the number of scales for mesh size (a) \( n_x = 128 \), (b) \( n_x = 256 \) and (c) \( n_x = 512 \).

FIG. 5. The four lowest energy levels of two electrons trapped in a harmonic potential as a function of \( \omega \). The lines show the solution with the shooting method, and the circles show the solution with the CI method with the Coulomb integrals calculated by our method. \( M \) is the angular momentum quantum number, and \( d \) denotes the degeneracy of the levels.

We present the performance of our method as \( n_x \) is varied from 64 to 512, doubling \( n_x \) at each step. The results are obtained for \( \hbar \omega = 200 \) meV. The hyperparameters used for each mesh size are summarized in Table I.

Figures 7(a) and (b) show the difference between the energies obtained with both methods relative to the result of method (i). Method (i), although not exact, will...
TABLE I. Hyperparameters used for training the kernels for Fig. 7 for each mesh of size $n_x \times n_x$ and the obtained loss function $\mathcal{L}$ for the interaction potential (14) in two dimensions.

| $n_x$ | $S$ | $K$ | $l_0$ | $\varepsilon$ | $N_{it}$ | $\mathcal{L}$ |
|------|----|----|-----|--------|--------|--------|
| 64   | 2  | 65 | 0.004 | 0.2     | 2000   | 0.0122 |
| 128  | 3  | 65 | 0.007 | 0.4     | 2000   | 0.0212 |
| 256  | 4  | 65 | 0.003 | 0.4     | 2000   | 0.0271 |
| 512  | 5  | 65 | 0.002 | 0.4     | 2000   | 0.0293 |

TABLE II. Hyperparameters used for training the kernels for Fig. 8 for a mesh of $256 \times 256$ for each $K$ and the obtained loss function $\mathcal{L}$ for the interaction potential (14) in two dimensions.

| $n_x$ | $S$ | $K$ | $l_0$ | $\varepsilon$ | $N_{it}$ | $\mathcal{L}$ |
|------|----|----|-----|--------|--------|--------|
| 256  | 2  | 257 | 0.008 | 0.2     | 2000   | 0.0050 |
| 256  | 3  | 129 | 0.010 | 0.2     | 2000   | 0.0142 |
| 256  | 4  | 65  | 0.005 | 0.3     | 2000   | 0.0267 |
| 256  | 5  | 33  | 0.008 | 0.3     | 2000   | 0.0549 |
| 256  | 6  | 17  | 0.004 | 0.3     | 1800   | 0.1251 |

**FIG. 7.** The difference between energies obtained with both methods relative to the result of method (i) for the respective energy levels shown in the same line color as in Figs. 5 and 6 for (a) two electrons and (b) three electrons.

As can be expected, for smaller kernels (and a higher number of scales $S$) the error increases. However, even for the smallest kernel size, the errors do not exceed $4 \times 10^{-3}$ of the reference energies. The benchmark in Fig. 4 shows that the method tends to be faster for a larger $S$. Thus choosing the hyperparameters $K$ and $S$ is a trade-off between speedup and accuracy.

**B. Effective 1D interaction potential**

As the next example of the application of our approach we present the results for the integration of a non-Coulomb interaction potential in 1D systems. We consider a quasi-one-dimensional quantum dot, formed in a semiconductor by strong confinement in two directions. We assume a harmonic oscillator confining potential in the $(x, y)$ direction. Assuming that for a strong lateral confinement the electrons are frozen to the ground harmonic potential state and integrating over the lateral coordinates, one obtains the interaction potential:

$$u(z_{ij}) = \left(\frac{\pi}{2}\right)^{\frac{3}{2}} \kappa (2l^2) \operatorname{erfc}(z_{ij}/2l) \exp(z_{ij}^2/2l^2).$$

Here $z_{ij} = |z_i - z_j|$, and $l = \sqrt{\hbar/m^*\omega}$. The single-electron Hamiltonian (2) is

$$h_i = -\frac{\hbar^2}{2m^*} \frac{d^2}{dz_i^2} + V(z_i),$$

and we assume a 1D infinite well confinement potential in $V(z_i)$.

For a few electrons confined in such quasi-one-dimensional systems, formation of Wigner molecules was observed for sufficiently long dots. Figures 9(a) and 9(c) show the energies as a function of the length $d$ of the potential well in $z$ for two and three electrons confined in the dot, respectively. The calculations are done
for mesh $n_x = 256$. For the evaluation of the kernels for our method we used the parameters: $S = 4$, $K = 65$, an initial learning rate of 0.02 decayed by $\epsilon = 0.65$ in ten steps, and $N_{it} = 300$ iterations.

In Figs. 9(b) and 9(d) the relative difference between the results of method (ii) and method (i) is shown. The line colors correspond to the energy levels in Figs. 9(a) and 9(c). The relative error is of the order of $10^{-4}$, which allows for a sufficiently good evaluation of the energy levels.

### IV. SUMMARY

The calculation of the energy levels of many-body quantum systems is a long-established challenge. Even with the approximate methods including DFT and CI, the computation is time-consuming due to its high complexity, resulting from the need to evaluate a large number of two-electron integrals, among other causes. The aim of this work was to develop a fast and efficient approach to calculate the two-electron integrals for the few-electron calculations. For many problems, it is not crucial to obtain extremely high precision of the integration, and the acceleration of the computation is beneficial provided that the error is much smaller than the order of magnitude of the energies in the system. Our method allows us to significantly reduce the computation time, while maintaining reasonable accuracy. Picking the number of scales in our method one can choose between higher precision and faster computation. The optimized evaluation of the two-electron integrals can also be used in other methods used on a discrete mesh, e.g., the Hartree-Fock method.

### V. ACKNOWLEDGMENT

This research was supported in part by PLGrid Infrastructure.

#### Appendix A: Shooting method

The problem of two electrons confined in a 2D harmonic potential can be solved semi-analytically. The relative motion of the electrons is described in cylindrical coordinates by the Hamiltonian \[ \{H(\rho, \phi)\} \] and the mesh is discretized into nodes $\rho_i$. The relative motion of electrons written in cylindrical coordinates

\[
\begin{align*}
-\frac{1}{2} \left( \frac{d^2}{d\rho^2} + \frac{1}{\rho} \frac{d}{d\rho} - \frac{M^2}{\rho^2} - \frac{\gamma^2}{4} \rho^2 - \frac{1}{\rho} \right) R(\rho) &= \frac{E}{4} R(\rho),
\end{align*}
\]

(A1)

can be written in the finite-difference approximation with $E' = E/4$,

\[
\begin{align*}
&\left( \frac{1}{(\Delta \rho)^2} + \frac{1}{2 \rho \Delta \rho} \right) R_{i+1} \\
&\quad + \left( \frac{2}{(\Delta \rho)^2} + \frac{M}{\rho^2} + \frac{\gamma^2}{4} \rho^2 + \frac{1}{\rho} - 2E' \right) R_i \\
&\quad + \left( - \frac{1}{(\Delta \rho)^2} + \frac{1}{2 \rho \Delta \rho} \right) R_{i-1},
\end{align*}
\]

(A2)

where $R_i = R(\rho_i)$ is the wave function at node $\rho_i$ of the finite-difference mesh. In the shooting method we assume the boundary condition $R_0 = 0$ at the left edge of the mesh, and for a given energy we calculate the values of $R_i$ at the nodes of the mesh. We proceed to the right edge of the mesh, and $R_{n_x}$ needs to vanish. This condition is satisfied at discrete values of energy. The problem essentially is to find energies $E'$ at which $R_{n_x} = 0$ in Eq. (A2).

#### Appendix B: Effectiveness of the method

We consider the effectiveness of the method with respect to the number of electrons and dimensionality. We use the CI method, with $n$ basis states $\phi_n(\mathbf{r})$ from which we form the Slater determinants \[ \{\phi_1, \phi_2, \ldots, \phi_n\} \], and the number of the two-electron integrals depends only on the size of the basis $n$, irrespective of the number of electrons. In our method the calculation of two-electron integrals
is optimized via ML, thus the speedup depends only on the number of the basis states. In Fig. 10 we present the speedup (the ratio of the time of calculation by our method to the baseline time) as a function of $n$, obtained for electrons in one dimension [Fig. 10(a)] and two dimensions [Fig. 10(b)] on a mesh with $n_x = 256$, and a kernel with $K = 65$, $S = 4$. In two dimensions the calculation with our method is several times faster than using FFT. In one dimension our method is slightly slower than the baseline because it performs several additional operations (upsampling, downsampling) that in one dimension get less boost in parallelization. Importantly, our method gains more boost for two or more dimensions, and the results in one dimension are shown for presentation purposes.

Table III shows the total energies and the interaction energies of the two lowest levels calculated for two to four electrons and the error of our method relative to the baseline. The error tends to increase with the number of electrons; however, it does not change linearly, as one would expect. We found that the increase is of the same order as the increase of the interaction energy. The reason is that our method optimizes the evaluation of the interaction integrals; thus, the error will scale in a manner similar to the interaction energy.

| $N$ | Level | $E_{\text{FFT}}$ (eV) | $E_{\text{int}}$ (eV) | $E_{\text{FFT}} - E_{\text{kernels}}$ (eV) |
|-----|-------|------------------------|------------------------|--------------------------------------|
| 2   | 1     | 0.32727                | 0.04727               | $6 \times 10^{-6}$                  |
| 2   | 2     | 0.44457                | 0.16457               | $3.2 \times 10^{-5}$                |
| 3   | 1     | 0.66788                | 0.24788               | $6.1 \times 10^{-5}$                |
| 3   | 2     | 0.773875               | 0.353875              | $9.6 \times 10^{-5}$                |
| 4   | 1     | 1.03282                | 0.47282               | $1.6 \times 10^{-4}$                |
| 4   | 2     | 1.04097                | 0.48097               | $1.8 \times 10^{-4}$                |

Figure 10. Speedup of the method as a function of the number of base states $n$ for calculations of two electrons in (a) one dimension and (b) two dimensions and (c) the time of the calculation for the 2D case.
ergies," J. Chem. Phys. 87, 5968 (1987).

19 Massimo Rontani, Carlo Cavazzoni, Devis Bellucci, and Guido Goldoni, “Full configuration interaction approach to the few-electron problem in artificial atoms,” J. Chem. Phys. 124, 124102 (2006).

20 Irene Puerto Gimenez, Marek Korkusinski, and Pawel Hawrylak, “Linear combination of harmonic orbitals and configuration interaction method for the voltage control of exchange interaction in gated lateral quantum dot networks,” Phys. Rev. B 76, 075336 (2007).

21 Gregory Beylkin, Christopher Kurcz, and Lucas Monzón, “Fast convolution with the free space Helmholtz Green’s function,” J. Comput. Phys. 228, 2770 (2009).

22 Luigi Genovese, Thierry Deutsch, Alexey Neelov, Stefan Goedecker, and Gregory Beylkin, “Efficient solution of Poisson’s equation with free boundary conditions,” J. Chem. Phys. 125, 074105 (2006).

23 Lukas Exl, Norbert J. Mauser, and Yong Zhang, “Accurate and efficient computation of nonlocal potentials based on gaussian-sum approximation,” J. Comput. Phys. 327, 629 (2016).

24 S. C. Mukherjee, K. Roy, and N. C. Sil, “Evaluation of the Coulomb integral for scattering problems,” Phys. Rev. A 12, 1719 (1975).

25 Michal Lesnuk and Robert Moszynski, “Reexamination of the calculation of two-center, two-electron integrals over Slater-type orbitals. i. Coulomb and hybrid integrals,” Phys. Rev. E 90, 063318 (2014).

26 Mieke Peels and Gerald Knizia, “Fast evaluation of two-center integrals over gaussian charge distributions and gaussian orbitals with general interaction kernels,” Journal of Chemical Theory and Computation 16, 2570 (2020).

27 Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning (MIT Press, Cambridge, MA, 2016).

28 The source code for the implementation of the proposed method can be found at: https://github.com/kmkolasinski/approxkernel

29 Pierluigi Cudazzo, Ilya V. Tokatly, and Angel Rubio, “Dielectric screening in two-dimensional insulators: Implications for excitonic and impurity states in graphene,” Phys. Rev. B 84, 085406 (2011).

30 Stanislaw Bednarek, Bartłomiej Szafran, Tomasz Chwiej, and Janusz Adamowski, “Effective interaction for charge carriers confined in quasi-one-dimensional nanostructures,” Phys. Rev. B 68, 045328 (2003).

31 Bartłomiej Szafran, Francois M. Peeters, Stanislaw Bednarek, Tomasz Chwiej, and Janusz Adamowski, “Spatial ordering of charge and spin in quasi-one-dimensional Wigner molecules,” Phys. Rev. B 70, 035401 (2004).

32 For the Intel® MKL documentation we refer to https://software.intel.com/en-us/mkl-developer-reference-c-convolution-and-correlation.

33 Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernando Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” http://tensorflow.org.

34 Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in Advances in Neural Information Processing Systems 32 edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Curran Associates, 2019) p. 8024.

35 The optimization problem of the network is linear, but it is not trivial to reduce it to a system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$; thus, we use the standard approach of training the network that is quick and efficient.

36 Bartłomiej Szafran, Janusz Adamowski, and Stanislaw Bednarek, “Electron-electron correlation in quantum dots,” Phys. E 5, 185 (1999).