Signed particles and neural networks, Towards efficient simulations of quantum systems

Jean Michel Sellier*[a], Gaétan Marceau Caron[a], Jacob Leygonie[a]

[a] *Montréal Institute for Learning Algorithms, Montréal, Québec, Canada*
* *jeanmichel.sellier@gmail.com*

Abstract

Recently, two approaches were suggested which combine signed particles and neural networks to speed up the time-dependent simulation of quantum systems. Both specialize on the efficient computation of a multi-dimensional function defined over the phase space known as the Wigner kernel. In particular, as a first step, a completely analytically defined network was proposed, based on prior physics knowledge and not necessitating any training phase. Then, a simplified architecture was presented provided with generalization capabilities and, therefore, requiring a training process. Although relatively simple, these architectures have shown to provide important advantages in practical applications. They drastically reduce the amount of memory needed and provide valuable computational speedup. One quickly notes, though, that both approaches keep very similar structures and, consequently, are both affected by the very same issues. For example, they both utilize computationally expensive activation functions such as the sine functions. In this work, we go beyond neural networks depicted on physics based motivations and focus on more general architectures which do not require any prior knowledge. In more details, we suggest a network consisting of a set of different hidden layers which are based on more common, and less computationally expensive, activation functions such as rectified linear units, and which focus on predicting only one column of the discretized Wigner kernel at a time. This approach requires a training process to determine its weights and biases, which is performed over a dataset consisting of a position, a potential, and the corresponding column of the kernel at that position. This newly suggested architecture proves to accurately learn the transform from the space of potential functions to the space of Wigner kernels and performs very well during the simulation of quantum systems. Moreover, it allows a further reduction of the amount of memory required along with a much lower computational burden (20 times faster). To conclude, a validation test is presented consisting of a one-dimensional Gaussian wave packet impinging on a potential barrier. Comparisons with another validated approach are provided, clearly showing the validity of this newly suggested approach.

Keywords: Quantum mechanics, Machine learning, Signed particle formulation, Neural networks, Simulation of quantum systems

1. Introduction

Several years ago, one of the authors of this work introduced a new formulation of quantum mechanics known as the signed particle formulation [1]. This relatively new formalism is based on the concept of an ensemble of field-less Newtonian particles, provided with a sign, and which create couples of new particles according to some probability depending on the so-called Wigner kernel. Because of its simplicity, and in spite of its recent appearance, this new framework has already been applied with success to the simulation of a relatively big number of different situations, both in the context of single- and many-body systems, showing unprecedent advantages in terms of computational resources [2] (practical examples involving time-dependent simulations of quantum many-body systems are discussed in [3]-[7]). Even if this innovative approach has important and unique features, its main bottleneck remains the computation of the Wigner kernel, which is mandatory to evolve signed particles. In practice, this function is defined over a phase space which dimensions are equal to $2 \times N \times d$, where N is the number of physical bodies involved and d is the dimension of the real space ($d = 1, 2, 3$). Therefore, it can quickly become a critical aspect for the simulation of quantum systems since both the amount of memory and time required for its computation are cursed by the total dimensionality of the system or, equivalently, by the dimensionality of the configuration space.

Consequently, the use of Artificial Neural Networks (ANN) to address the problem of computing the Wigner kernel rapidly and reliably was suggested by the same author in [8]. In this preliminary investigation, a new technique was introduced which is based on an appropriately tailored ANN in the context of the signed particle formalism. The suggested network architecture has the peculiar feature of not requiring any training phase, since mathematical and physical knowledge of the problem is enough to retrieve the weights and biases analytically. Although this first approach has shown some important advantages, its computational complexity remains an issue (see [9] for a complete discussion on this topic). Subsequently, a more general approach was introduced which uses a different network architecture and, unlike the previous approach, requires a training process [9]. This method has the main advantage of reducing the complexity of the previous ANN (since it utilizes less neurons in the hidden layer) and, therefore, allows a faster computation of the Wigner kernel.

In this work, we present a completely different approach based on more common techniques coming from the field of machine learning. In particular, we suggest a new architecture consisting of three hidden layers with neurons which implement the rectified linear unit (ReLU) as their activations, i.e. a function which is drastically less demanding in terms of computational resources when compared to the previously utilized sine activation functions. In order to show the validity of this new approach, we apply it to the standard problem consisting

of a one-dimensional Gaussian wave packet going towards a potential barrier. For the sake of clarity and completeness, comparisons with one of the previously implemented techniques are presented.

This paper is organized as follows. In the next section, we briefly discuss the previous techniques which combine the use of signed particles with ANNs. Afterwards, we proceed with the description of the new ANN architecture which radically improves the previous techniques. Finally, a validation test is performed to assess the reliability and speed of the new suggested approach and conclusive comments are provided. The authors believe that this investigation represents a further step in the direction of depicting robust, fast and reliable methods to simulate time-dependent quantum systems, with a potential important impacts in fields such as quantum chemistry and quantum computing electronic device design.

2. Neural Network Architectures

In this section, we start by providing the context of the problem we face in this work. In particular, for the sake of self-consistency, we report the second postulate on which the signed particle formulation of quantum mechanics is based. Then, we proceed with a short description of the solutions proposed in [8] and [9]. Finally, we present a novel ANN architecture which is reliable and fast but does not carry the computational burden of the previously suggested techniques.

2.1. The second postulate of the signed particle formulation

The signed particle of quantum mechanics is based on three postulates which completely determines the time-dependent evolution of an ensemble of signed particles and, in turn, of a quantum system. In this section, we briefly discuss about postulate II, which eventually represents the bottleneck of this novel approach. Postulates I and III have been thoroughly discussed elsewhere and can be summarized as *1)* a quantum system is described by an ensemble of signed field-less classical particles which completely defines the system (essentially, in the same way the wave function does), and *3)* particles with opposite signs but equal position and momentum always annihilate. Postulate II, in full details, is reported below for a one-dimensional, single-body system (for the generalization to many-dimensional, many-body systems the reader is invited to refer to [2]).

Postulate. *A signed particle, evolving in a given potential $V = V(x)$, behaves as a field-less classical point-particle which, during the time interval dt, creates a new pair of signed particles with a probability $\gamma(x(t))\,dt$ where*

$$\gamma(x) = \int_{-\infty}^{+\infty} \mathcal{D}p' V_W^+(x;p') \equiv \lim_{\Delta p' \to 0^+} \sum_{M=-\infty}^{+\infty} V_W^+(x; M\Delta p'), \tag{1}$$

and $V_W^+(x;p)$ *is the positive part of the quantity*

$$V_W(x;p) = \frac{i}{\pi\hbar^2}\int_{-\infty}^{+\infty} dx' e^{-\frac{2i}{\hbar}x'\cdot p}\left[V(x+x') - V(x-x')\right], \tag{2}$$

known as the Wigner kernel [10]. If, at the moment of creation, the parent particle has sign s*, position* x *and momentum* p*, the new particles are both located in* x*, have signs* $+s$ *and* $-s$*, and momenta* $p+p'$ *and* $p-p'$ *respectively, with* p' *chosen randomly according to the (normalized) probability* $\frac{V_W^+(x;p)}{\gamma(x)}$*.*

Therefore, one can view the signed particle formulation as made of two parts: *1)* the evolution of field-less particles, which is always performed analytically, and *2)* the computation of the kernel (2), which is usually performed numerically. In particular, the computation of the Wigner kernel can, sometimes, represent a problem in terms of computational implementation as it is equivalent to a multi-dimensional integral which complexity increases rapidly with the dimensions of the configuration space. It is clear that a naive approach to this task is not appropriate (for more technical details the reader is encouraged to visit [16] for a free implementation of the approach).

We now briefly describe the previously suggested methods of [8] and [9], and then present our new approach.

2.2. Previous approaches

At a first glance, it might seem relatively simple to train an ANN to predict the kernel (2) once a potential is provided. In other words, one simply looks for a map between the space of vectors representing physical potentials and the space of the corresponding kernels, a rather common problem in machine learning (usually referred to as supervised learning).

It initially appeared to the authors that a simple naive approach based on a completely general ANN aiming to learn the mapping by itself would be difficult to depict and train. Therefore, we first decided to exploit some prior knowledge to make the problem approachable (interestingly enough, similar conclusions have been obtained in [11] and [12]). Surprisingly, it was eventually shown in [8], that by performing some relatively simple algebraic manipulation, it is possible to obtain such ANN *without* any training since we are in front of one rare example of neural network which weights can be found analytically. The network consists of an input layer, a hidden layer and an output layer. The input layer receives a discretized position in the phase space, indicated by the couple of integers (i,j), along with a discretized potential $V = V(x)$, represented by the vector $[V_1 = V(x_1), \ldots, V_n = V(x_n)]$. To speed up the network, an initial pre-computation of the angles θ_l and the corresponding sine functions is performed. Eventually, these values are utilized to define the activation functions of the hidden layer and, consequently, an weighted average is computed in the last layer which represents the output of the network (see [8] for all details). This quite

uncommon approach brings two important advantages: *1)*, it completely avoids the need to compute the Wigner kernel everywhere on the phase-space, *2)*, the curse of dimensionality affecting the amount of memory required is completely removed from the picture. One important drawback remains though since this network still retains the initial complexity of the problem.

Eventually, in order to give generalization capabilities and, therefore, reduce the numerical complexity of the problem, an improvement to the previous approach was suggested based on introducing an arbitrary number of parameters (in other words, weights and biases) to be learned during a training process [9]. To achieve such goal, one starts from the previous approach and carefully simplifies it to do not loose accuracy. In particular, one introduces the physically reasonable hypothesis that a given potential can be approximated by a baricentric interpolation. Although arbitrary and dependent on the discretization lenght in the configuration space, this assumption offered a first simple way to improve our previous approach in terms of generalization and, therefore, numerical performance. Eventually, this interpolation is generalized to a weighted average of the potential values and the weights are learned by the network during the training process. In order to find those values, we search for the weights which provide the best network approximation of the function $V_W = V_W(x; p)$ (representing the dataset) by means of a standard machine learning method known as stochastic gradient descent.

2.3. Learning how to compute the Wigner kernel

We, now, introduce a new (and very different) approach to the problem of computing the Wigner kernel by means of a completely general neural network (i.e. not based on any prior physical knowledge). In more details, the purpose of this ANN is to learn a mapping $f : \mathbb{R}^m \times I \to \mathbb{R}^n$, which generates a ($n$-dimensional) column of the discretized Wigner kernel once a (m-dimensional) discretized potential and an index in I are provided, and where the index associated to the column represents a position in the discretized configuration space. To achieve it, we use a totally agnostic approach to quantum physics by casting the problem into a standard supervised learning problem where the mapping is learned from a finite set of training examples. By using a simple feedforward neural network, we show that it is possible to learn such a mapping which, in turn, allows the simulation of quantum systems efficiently and accurately in the context of the signed particle formulation discussed above.

However, it is well known that, compared to approximating the kernel (2) by some standard numerical method (e.g. finite differences), the price to pay consists in not having any theoretical guarantees that the ANN will generalize well enough for *any* discretized potential in $\mathbb{R}^m$, or even worse, for all indices for a given potential. Therefore, we are forced to trade-off this property for a significant speedup in terms of computational time without losing too much accuracy. In any case, in the next section, we will empirically demonstrate that the approximation error is indeed negligible for a large family of physically

meaningful potentials and, in particular, for the barrier potential (a quintessential validation test for quantum simulations). In the light of these results, one can claim that *the model used does not only generalize in the statistical sense, it also learns a very good approximation of the real transformation f.*

In more details, to achieve the goal described above, the main effort is represented by the optimization of two antagonist objectives, i.e. computational accuracy and speed, by selecting the appropriate ANN architecture. Thus, to facilitate this task, we restrict the search space of ANN architectures to feedforward neural networks. This is mainly motivated by their simplicity along with their ability to easily capture global interactions between the inputs[1]. Moreover, for a fixed potential, the diversity of outputs becomes important when we vary the index, since this diversity is controlled only by changing the index variable. To increase the influence of this input variable, we encode it as a one-hot vector and we embed it in a high-dimensional space with five times more entries than the number of positions, as depicted in Fig. 1. Eventually, this embedding is transformed again into a hidden representation that is concatenated to the hidden representation of the potential. Finally, this vector is then processed by two pairs of rectified linear units (ReLU) followed by a linear transformation before generating the kernel column. At the end, this model has $19,550,233$ free parameters (for the case discussed in the next section), and yet it is the smallest architecture which achieves accurate computations that we were able to find. Even though the number of parameters might sound quite large at a first glance, we actually achieve a speedup about 18 times faster than the simplest finite difference quadrature (such as a C implementation of the midpoint and the rectangular quadrature methods). In fact, the architecture presented in Fig. 1 can be efficiently implemented on Graphical Processing Units (GPUs), without any effort, with the use of modern deep learning libraries such as Pytorch [15]. Consequently, most of the computations consists of matrix multiplications which allows to increase the speedup by using large batches of data at once. On table 1, we show the average time in milliseconds to compute a whole kernel. We take into account the initial cost associated to the initialization of the GPU. This cost is then amortized by computing several columns of a kernel inside a single batch of data.

3. Numerical validation

In this section, for the sake of completeness, we perform the same validation test as the one presented in [8] and [9]. In particular, we simulate a representative one-dimensional quantum system made of a Gaussian wave packet moving against a potential barrier positioned at the center of a finite domain

[1] Interestingly, we also experimented with convolutional neural networks, but their performances were not satisfactory to justify their use in a quantum simulation.

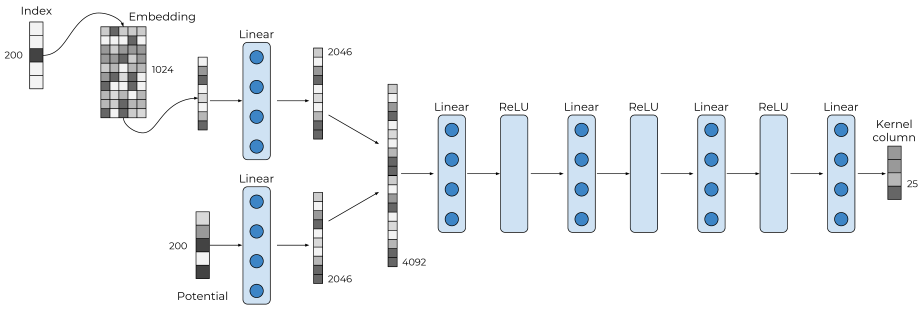

Figure 1: Visual representation of the ANN suggested in this work. The index is represented by a one-hot vector which selects one column of the embedding. This column is linearly transformed into a larger vector space and concatenated with the potential and, successively, this new vector is transformed by a composition of linear transformations and ReLU activations. The hidden representations after the concatenation module all have 2046 dimensions (not shown on the figure). The circles mean that the module contains parameters.

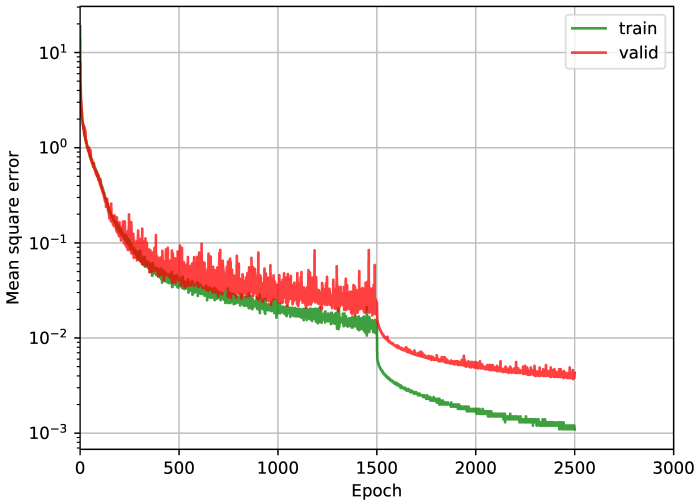

Figure 2: The learning curve of the model on the training and validation sets. The MSE is given on a log-scale. After 1500 epochs, we divide the learning rate by ten.

# Potential	Avg. time (ms.)	Speedup
1	1.69 ± 0.4	9x
10	0.98 ± 0.03	15x
25	0.89 ± 0.03	17x
100	0.84 ± 0.03	18x

Table 1: Computation time in milliseconds (ms) required to compute a complete Wigner kernel from a given potential. By gathering several potentials together, the computation time per potential is amortized because of the efficiency of the GPU for matrix-matrix computations with high-dimensions. The average and standard deviation are computed over 10 runs. The reference time for the computation time on the CPU is 15ms.

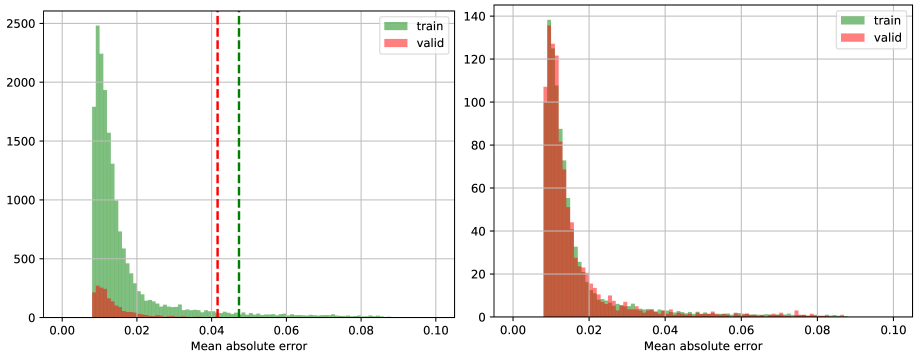

Figure 3: Comparison of the mean absolute errors between the training set and the validation set in absolute (left-hand side) and relative (right-hand side) frequency.

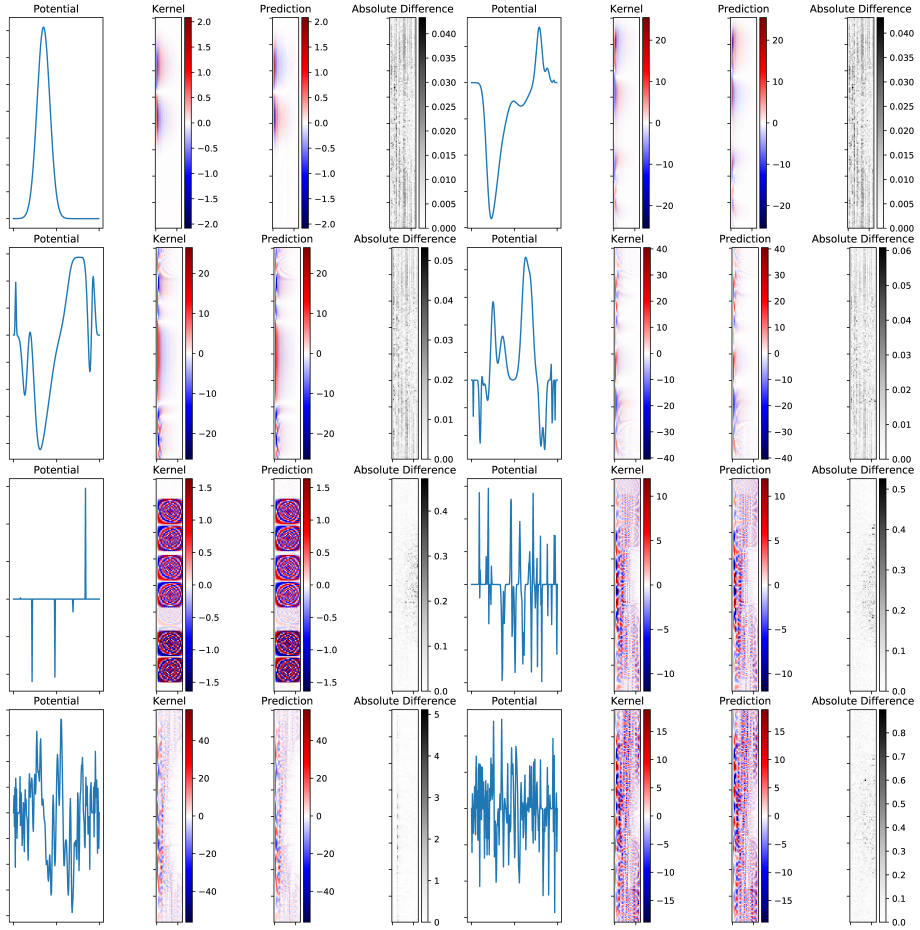

Figure 4: The model is able to generalize to a large range of different potentials obtained by linear combinations of Gaussian functions.

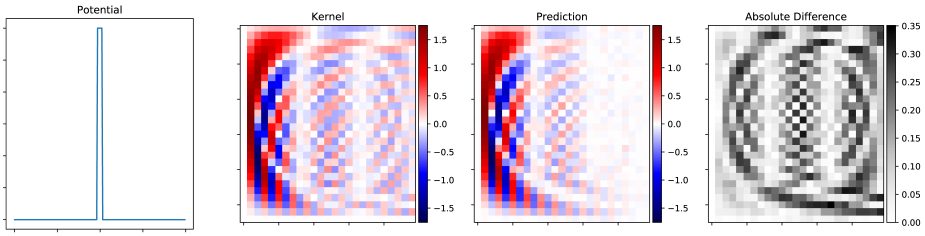

Figure 5: The model is able to generalize to an abrupt potential barrier which is not part of the training set but which can be accurately approximated by narrow Gaussian functions (he kernel and its prediction are cropped to the region with non-zero values).

(200nm), with width and height equal to 6nm and -0.3eV, respectively. The initial conditions for this system consists of:

$$f_W^0(x;M) = Ne^{-\frac{(x-x_0)^2}{\sigma^2}} e^{-\frac{1}{\hbar^2}(M\Delta p - p_0)^2\sigma^2} \tag{3}$$

with N, p_0, x_0 and σ the constant of normalization. As usual, the initial position, dispersion and wave number of the packet are equal to 68.5nm, 10nm and $6.28 \cdot 10^{-2}\text{nm}^{-1}$, respectively. It can be easily shown that this corresponds to an initial wave packet energy smaller than the energy of the barrier. Consequently, one expects both reflection and tunneling effects appearing during the time-dependent simulation. Finally, absorbing boundary conditions are applied at the edge of the simulation domain. In spite of its simplicity, this numerical experiment represents a solid validation test. Obviously, more complex situations could be simulated but would be out of the scope of this paper.

Concerning the training process of the neural network, as a starting point, a dataset must be created. One is readily obtained by, first, discretizing the position into 200 entries and 20 entries for the momentum. Then, we generate $20,000$ couples of potentials and their corresponding Wigner kernel, and a final split into a $90:10$ between the training set and the validation set is applied randomly. In more details, the potentials are randomly selected from a family of Gaussian bell shapes with different number of peaks, positioned randomly and with different dispersions. To validate the training process on such set, we adopt the standard cross-validation technique where one trains several ANNs on the training set and select the best one according to the error on the validation set. Moreover, we use as the final test set a single potential, which belongs to a completely different family than the one used during the training phase. More precisely, we test our newly suggested ANN on an abrupt potential barrier. Consequently, we make sure that the model captures the underlying structure of the transformation f. By observations on various numerical experiments performed (see below), we conclude that this procedure is sufficient to stabilize the loss across different random splits. Moreover, one should note that from a couple consisting of a potential and its corresponding kernel, one can extract several training examples, one per column of the kernel. Therefore, in order to help the optimizer of the training process, we build mini-batches by gathering

the columns of a kernel sharing the same potential.

The parameters of the model are optimized by means of the well-known ADAM method [14], which minimizes the Mean Square Error (MSE). In particular, we use a learning rate equal to 10^{-4} which we manually decrease after 1500 epochs by one order of magnitude. The size of the minibatch is equal to 10 potentials, which is equivalent to 2000 training examples. The optimizer minimizes the MSE to 0.0011 on the training set and 0.0041 on the validation set, as depicted in Fig. 2. The use of ADAM is necessary to obtain low errors in a reasonable time. [2] Although in the presence of a model with 20M parameters and no regularization term at all, the overfitting proves to be unsignificant. This can be explained by the large number of training examples used to train the model. In fact, this is confirmed by comparing the distributions of errors between the training and the validation sets. As we observe in Fig. 3 (right), the normalized distributions are hardly distinguishable. Moreover, the distributions are highly skewed towards zero, and 95% of the examples have a mean absolute error smaller than 0.05. In Fig. 4, the potential input of the model is fixed and the index varies in order to generate the whole kernel. The kernels are transposed, so that the model predicts each line of the kernel individually. We observe that, once the model is parametrized by a fixed potential, it can generate accurately any position without explicitly using the spatial information of the kernel. In Fig. 5, we show the prediction of the model for the barrier potential described above (the kernel and the prediction are cropped to the region with non-zero values). One observes that the model is still able to provide accurate predictions even for low momenta, but loses the structure for high momenta with a maximum error around 0.35. While this error is high relatively to the range of values for the kernel, as previously observed in [8] and [9], the simulation of the physical system is robust to this errors associated to high momenta. This is mainly due to two factors: on the one hand, the signed particle formulation is an intrinsically stochastic approach, therefore robust to noise/perturbations in the kernel, on the other hand, signed particles rarely explore the area corresponding to high momenta due to trivial energetic reasons. As a matter of fact, these approximation errors do not have a significant impact on the simulations as clearly shown in Fig. 6.

4. Conclusions

In this work, we introduced a novel technique combining neural networks and signed particles to achieve faster, and still reliable, time-dependent simulations of quantum systems. This newly suggested approach represents an important generalization over the previous techniques presented in [8] and [9]. In practice, we depicted a feedforward neural network, i.e. a simple architecture able to easily capture global interactions between the inputs, and consisting of a layer embedding the input, encoded as a one-hot vector, in a high-dimensional space

[2] On a NVIDIA GTX 1080Ti, the training time is less than 1 day.

with five times more entries than the number of positions, a layer transforming this embedding into a hidden representation concatenated to the hidden representation of the potential and two pairs of ReLU layers processing this vector followed by a linear transformation (see Fig. 1). Obviously, in this new context, a trade-off between computational time and accuracy is clearly introduced. Interestingly enough, in spite of the simplicity of this model, we are actually able to achieve a speedup about 18 times faster than the simplest finite difference quadrature (see table 1). A representative validation test consisting of a wave packet impinging on a potential barrier has been performed which clearly shows the validity and accuracy of our newly suggested method in practical situations.

Nowadays, important experiments are being performed in many branches of the broad field of quantum technologies, such as quantum computing, quantum chemistry, nanoelectronics, etc. In this promising context, it is clear that our quantum simulation and design capabilities are going to play a fundamental role which is going to grow in importance in the next future. Moreover, it is becoming clear that, to solve modern (and highly challenging) technological problems related to quantum mechanics, the adoption of dramatically different approaches is going to be necessary. The authors of this paper believe that the technique suggested in this work is a promising candidate.

Acknowledgments. One of the authors, JMS, would like to thank M. Anti for her support, ethusiasm and encouragement.

References

[1] J.M. Sellier, A Signed Particle Formulation of Non-Relativistic Quantum Mechanics, Journal of Computational Physics 297, pp. 254-265, (2015).

[2] J.M. Sellier, M. Nedjalkov, I. Dimov, An Introduction to the Applied Quantum Mechanics in the Wigner Monte Carlo Formalism, Physics Reports 577, pp. 1-34, (2015).

[3] J.M. Sellier, I. Dimov, A Wigner Monte Carlo Approach to Density Functional Theory, Journal of Computational Physics 270, pp. 265-277, (2014).

[4] J.M. Sellier, I. Dimov, The Many-body Wigner Monte Carlo Method for Time-dependent Ab-initio Quantum Simulations, Journal of Computational Physics 273, pp. 589-597, (2014).

[5] J.M. Sellier, I. Dimov, On the Simulation of Indistinguishable Fermions in the Many-Body Wigner Formalism, Journal of Computational Physics 280, pp. 287-294, (2015).

[6] J.M. Sellier, I. Dimov, The Wigner-Boltzmann Monte Carlo method applied to electron transport in the presence of a single dopant, Computer Physics Communications 185, pp. 2427-2435, (2014).

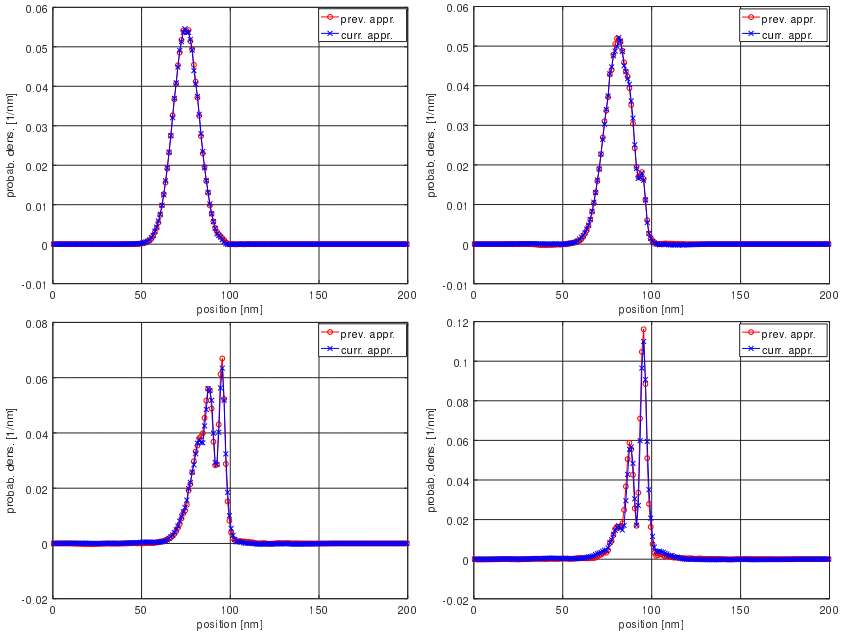

Figure 6: Time-dependent evolution of a wave packet interacting with a potential barrier positioned at the center of the spatial domain, at times 1fs (top left), 2fs (top right), 3fs (bottom left) and 4fs (bottom right) respectively, and with two different approaches for the kernels corresponding to the case in [8] (red circles) and the one suggested in this work (blue crosses).

[7] J.M. Sellier, K.G. Kapanova, A Study of Entangled Systems in the Many-Body Signed Particle Formulation of Quantum Mechanics, International journal of quantum chemistry, 10.1002/qua.25447, (2017).

[8] J.M. Sellier, Combining Neural Networks and Signed Particles to Simulate Quantum Systems More Efficiently, Physica A 496, pp. 62-71, (2017).

[9] J.M. Sellier, J. Leygonie, G. Marceau Caron, Combining Neural Networks and Signed Particles to Simulate Quantum Systems More Efficiently, Part II, under review on Physica A and currently available at https://arxiv.org/abs/1806.00082, (2018).

[10] E. Wigner, On the Quantum Correction for Thermodynamic Equilibrium, Physical Review 40, no. 5, 749, (1932).

[11] G. Carleo, Ma. Troyer, Solving the Quantum Many-body Problem with Artificial Neural Networks, Science 355, pp. 602606, (2017).

[12] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, K.-R. Muller, Bypassing the Kohn-Sham Equations with Machine Learning, Nature Communications, vol. 8, 872, (2017).

[13] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, (2016).

[14] D.P. Kingma, J.L. Ba, ADAM: A Method for Stochastic Optimization, ICLR proceeding, (2015).

[15] A. Paszke, et al., Automatic Differentiation in PyTorch, NIPS Workshop, (2017).

[16] J.M. Sellier, nano-archimedes, accessed 24 May 2018,
URL: www.nano-archimedes.com