

# Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks

A. Pukrittayakamee,<sup>1</sup> M. Malshe,<sup>2</sup> M. Hagan,<sup>1</sup> L. M. Raff,<sup>3,a)</sup> R. Narulkar,<sup>2</sup>  
S. Bukkapatnum,<sup>4</sup> and R. Komanduri<sup>2</sup>

<sup>1</sup>Electrical and Computer Engineering, Oklahoma State University, Stillwater, Oklahoma 74078, USA

<sup>2</sup>Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma 74078, USA

<sup>3</sup>Chemistry Department, Oklahoma State University, Stillwater, Oklahoma 74078, USA

<sup>4</sup>Industrial Engineering, Oklahoma State University, Stillwater, Oklahoma 74078, USA

(Received 25 July 2008; accepted 13 February 2009; published online 1 April 2009)

An improved neural network (NN) approach is presented for the simultaneous development of accurate potential-energy hypersurfaces and corresponding force fields that can be utilized to conduct *ab initio* molecular dynamics and Monte Carlo studies on gas-phase chemical reactions. The method is termed as combined function derivative approximation (CFDA). The novelty of the CFDA method lies in the fact that although the NN has only a single output neuron that represents potential energy, the network is trained in such a way that the derivatives of the NN output match the gradient of the potential-energy hypersurface. Accurate force fields can therefore be computed simply by differentiating the network. Both the computed energies and the gradients are then accurately interpolated using the NN. This approach is superior to having the gradients appear in the output layer of the NN because it greatly simplifies the required architecture of the network. The CFDA permits weighting of function fitting relative to gradient fitting. In every test that we have run on six different systems, CFDA training (without a validation set) has produced smaller out-of-sample testing error than early stopping (with a validation set) or Bayesian regularization (without a validation set). This indicates that CFDA training does a better job of preventing overfitting than the standard methods currently in use. The training data can be obtained using an empirical potential surface or any *ab initio* method. The accuracy and interpolation power of the method have been tested for the reaction dynamics of H+HBr using an analytical potential. The results show that the present NN training technique produces more accurate fits to both the potential-energy surface as well as the corresponding force fields than the previous methods. The fitting and interpolation accuracy is so high (rms error=1.2 cm<sup>-1</sup>) that trajectories computed on the NN potential exhibit point-by-point agreement with corresponding trajectories on the analytic surface. © 2009 American Institute of Physics. [DOI: 10.1063/1.3095491]

## I. INTRODUCTION

The interpolation of *ab initio* potential energy surface (PES) data using neural networks (NNs) (Refs. 1–14) and other methods<sup>15–33</sup> for the purpose of executing molecular dynamics (MD) simulations has been a subject of considerable interest. Recently, Raff *et al.*<sup>1</sup> investigated in detail a method involving the use of NNs and novelty sampling (NS) to affect the interpolation of *ab initio* PES data. The results of such studies<sup>1</sup> performed for Si<sub>n</sub> ( $n=3,4,\dots,7$ ),<sup>34</sup> SiO<sub>2</sub>,<sup>35</sup> HONO,<sup>36</sup> and vinyl bromide<sup>2</sup> systems show that the NN/NS method is very promising in terms of accuracy, convenience, and the requirement of Central Processing Unit (CPU) time. Our recent study<sup>2</sup> on the vinyl bromide dissociation dynamics demonstrates the success of the NN method for MD simulations of a six-atom system having six open reaction channels.

All of the MD studies performed up to the present time have concentrated only on the fitting of the potential energy, although the force field is the key component for the MD

simulation. This situation is due to two factors. First, the *ab initio* computation of surface gradients requires far more computational effort than just the potential itself. Second, Shepard<sup>15,18,20–22,24–26,28</sup> and Interpolative Moving Least Squares (IMLS)<sup>31–33</sup> methods are not easily adaptable to fitting surface gradients and adapting NN methods to the simultaneous fitting of both the potential and its gradients requires a significant modification<sup>37</sup> of the usual NN fitting methods.<sup>38</sup> Since the forces can be computed by differentiating the potential surface, it has been assumed in the past that an accurate potential surface will yield accurate forces without specifically fitting the forces. Witkoskie and Doren<sup>39</sup> recently reported a NN method in which the gradient of a function is utilized to improve the NN fit to the function. As this research is related to the present method it is discussed in more details in the next section.

In section II of this paper we present a NN training method that explicitly and simultaneously fits both the potential surface and its corresponding force field. Although the output layer of the NN contains only a single neuron that gives the potential energy, the derivatives of the NN with respect to its inputs are automatically adjusted to match the

<sup>a)</sup>Electronic mail: lionel.raff@okstate.edu.

target gradients. This procedure is more efficient than having the gradients produced by separate neurons in the output layer as such a procedure would greatly increase the complexity of the NN.

## II. NN TRAINING METHOD

Previous efforts to fit NNs to PESs involved an optimization procedure in which the mean square error between the network output and the desired potential surface was minimized. The typical mean square error is computed as follows:

$$J_f(\mathbf{w}) = \frac{1}{Q} \sum_{q=1}^Q (V(\mathbf{p}_q) - a(\mathbf{p}_q))^2, \quad (1)$$

where  $V$  is the potential energy,  $a$  is the output of the network, and  $\mathbf{p}_q$  is the column vector of the internal coordinates of the  $q$ th cluster in the data set currently being used to determine the NN force field.

Witkoskie and Doren<sup>39</sup> modified Eq. (1) to incorporate the use of gradient information to improve the NN fit to the function with perhaps fewer data and number of neurons. However, no data are given concerning the accuracy with which their improved function fit also fits the known gradients. Also, their method does not allow for adjustment of the relative weights to be assigned to fitting the surface and fitting the gradients. Since there is only one potential but many gradients which will span a very large range of values on a typical potential-energy hypersurface, relative weighting can be important in obtaining optimum interpolation accuracy. Witkoskie and Doren<sup>39</sup> applied the method to a simple, separable, two-dimensional function that was a Morse potential in one dimension modified by a multiplicative sinusoidal function in the second dimension. Since their work concentrated on using the gradient to improve the fit of the NN to the function rather than accurately fit the gradient itself, Witkoskie and Doren<sup>39</sup> did not report the accuracy of the gradients predicted by their final NN fit. No MD simulations on a realistic, highly coupled model potential-energy hypersurface for a chemical system were reported.

The present paper along with those in Ref. 37 presents a general method for simultaneously fitting both a function and its gradient. The underlying equations summarized here and derived in detail in the paper in Ref. 37 are applicable to any system for which a sufficiently large database is available. They apply to NNs with any number of elements in the input vector, any number of layers and neurons in each layer, and any transfer functions. The basis of the method is the use of a performance function that explicitly requires the simultaneous fitting of both function and gradient and the incorporation of relative weighting of surface and gradient fitting. We apply the resulting method to a realistic, three-body potential-energy hypersurface and demonstrate that MD trajectories computed on the NN resulting from the combined function derivative approximation (CFDA) method exhibit a point-by-point agreement with the corresponding trajectories computed on the model analytic potential hypersurface.

The mean square error performance function given by Eq. (1) does not explicitly penalize errors in the potential

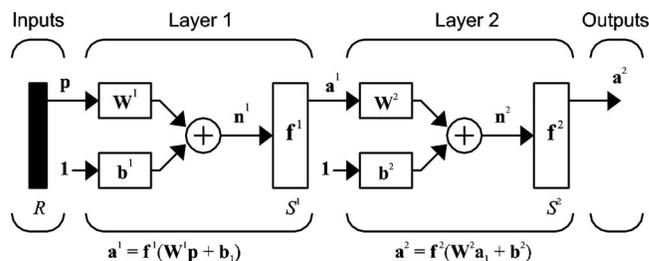


FIG. 1. Schematic of a two-layer NN.

gradients, which determine the force field. If the potential is accurately captured by the NN, then we would expect that the gradients would also be reasonably approximated, but this is not assured. We have recently proposed a procedure that explicitly penalizes both the errors in the potential and in the gradient. The performance index is defined as

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{Q} \sum_{q=1}^Q (V(\mathbf{p}_q) - a(\mathbf{p}_q))^2 \\ &+ \frac{\rho}{QR} \sum_{q=1}^Q \sum_{i=1}^R \left( \left. \frac{\partial V(\mathbf{p})}{\partial p_i} \right|_{\mathbf{p}=\mathbf{p}_q} - \left. \frac{\partial a(\mathbf{p})}{\partial p_i} \right|_{\mathbf{p}=\mathbf{p}_q} \right)^2, \\ &= J_f(\mathbf{w}) + \rho J_d(\mathbf{w}), \end{aligned} \quad (2)$$

where  $\mathbf{w}$  is the vector containing all of the weights and biases in the NN and  $\rho$  is a scale factor that determines the relative importance of potential error and gradient error.<sup>37</sup>

By using Eq. (2), we are able to approximate both a potential surface and its gradient with a NN that has a single neuron in the last layer, representing only the potential. In the remainder of the paper, we will demonstrate the operation of this CFDA training algorithm on a classic analytical potential hypersurface representing the H<sub>2</sub>Br system and conduct MD studies on the exchange and abstraction dynamics occurring in H+HBr collisions. It will be shown that trajectories computed using NN fits obtained from Eq. (2) produce a point-by-point agreement with those computed using the analytic hypersurface.

The CFDA algorithm minimizes the performance function defined in Eq. (2) using standard gradient-based (e.g., the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method<sup>40</sup>) or Jacobian-based (e.g., the Levenberg-Marquardt method<sup>41</sup>) optimization procedures. The difficulty in using Eq. (2), instead of Eq. (1), for NN training is that it requires the computation of second derivatives of the NN response. This is in contrast to the standard backpropagation<sup>42</sup> training algorithm for NNs, which is designed for Eq. (1) and requires only the computation of first derivatives of the NN response.

The details of the gradient calculation are provided by Pukrittayakamee *et al.*<sup>37</sup> We will provide a summary of the calculations here. We begin by introducing the notation that we will use for the NN. Figure 1 is an example of a multilayer network (two layers are shown in this case, but an arbitrary number of layers can be used). The input vector to the network is represented by  $\mathbf{p}$ , which has  $R$  elements. The input is connected to layer 1 through the input weight  $\mathbf{W}^1$ ,

where the superscript represents the layer number. The bias for the first layer is represented by  $\mathbf{b}^1$ . The net input to layer 1 is denoted by  $\mathbf{n}^1$  and is computed as

$$\mathbf{n}^1 = \mathbf{W}^1 \mathbf{p} + \mathbf{b}^1. \quad (3)$$

In scalar notation this becomes

$$n_i^1 = \sum_{j=1}^R w_{i,j}^1 p_j + b_i^1. \quad (4)$$

The output of layer 1, which is  $\mathbf{a}^1$ , is computed by passing the net input through a transfer function according to

$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{n}^1). \quad (5)$$

This layer output has  $S^1$  elements. The output of the first layer is input to the second layer through the layer weight  $\mathbf{W}^2$ . A network can have any number of layers although two- and three-layer networks are generally used. The equations for successive layers can be expressed as follows:

$$\mathbf{n}^m = \mathbf{W}^m \mathbf{a}^{m-1} + \mathbf{b}^m, \quad \mathbf{a}^m = \mathbf{f}^m(\mathbf{n}^m). \quad (6)$$

If there are  $M$  layers in the network, the output of the final layer is identified as the network output  $\mathbf{a} = \mathbf{a}^M$ .

We are now ready to calculate the gradient of  $J(\mathbf{w})$  with respect to the weights and biases of the network. The gradient of  $J_f(\mathbf{w})$  is computed using the standard backpropagation algorithm.<sup>40</sup> In the following, we will summarize the gradient calculation for  $J_d(\mathbf{w})$ .

An element of this gradient that corresponds to a network weight will take the form

$$\frac{\partial J_d(\mathbf{w})}{\partial w_{i,j}^m} = \frac{-2}{QR} \sum_{q=1}^Q \sum_{r=1}^R \left( \left. \frac{\partial V(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} - \left. \frac{\partial a(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} \right) \times \left( \frac{\partial}{\partial w_{i,j}^m} \left. \frac{\partial a(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} \right). \quad (7)$$

If we define the two terms

$$v_{i,q}^m = \sum_{r=1}^R \left( \left( \left. \frac{\partial V(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} - \left. \frac{\partial a(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} \right) \times \left( \frac{\partial}{\partial p_r} \left. \frac{\partial a(\mathbf{p})}{\partial n_i^m} \right|_{\mathbf{p}=\mathbf{p}_q} \right) \right) \quad (8)$$

and

$$u_{q,i,r}^m = \left( \left. \frac{\partial V(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} - \left. \frac{\partial a(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} \right) \times \left( \frac{\partial a(\mathbf{p})}{\partial n_i^m} \right)_{\mathbf{p}=\mathbf{p}_q}, \quad (9)$$

we can rewrite the element of the gradient as

$$\frac{\partial J_d(\mathbf{w})}{\partial w_{i,j}^m} = \frac{-2}{QR} \left( \left( \sum_{q=1}^Q a_j^{m-1}(\mathbf{p}_q) v_{i,q}^m \right) + \sum_{q=1}^Q \sum_{r=1}^R \left( \left. \frac{\partial a_j^{m-1}(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} u_{q,i,r}^m \right) \right). \quad (10)$$

To use Eq. (10), we need to compute the following three terms:  $v_{i,q}^m$ ,  $u_{q,i,r}^m$ , and

$$\left. \frac{\partial a_j^{m-1}(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q}.$$

The term

$$\left. \frac{\partial a_j^{m-1}(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q}$$

is updated forward from layer to layer using

$$\left. \frac{\partial a_j^m(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} = f^m(n_{j,q}^m) \sum_{l=1}^{S^{m-1}} w_{j,l}^m \times \left. \frac{\partial a_l^{m-1}(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q}, \quad (11)$$

which is initialized at the first layer with

$$\left. \frac{\partial a_j^m(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} = \begin{cases} 1, & j=r \\ 0, & j \neq r. \end{cases} \quad (12)$$

The term  $u_{q,i,r}^m$  is backpropagated from layer to layer using

$$u_{q,i,r}^m = f^m(n_{i,q}^m) \sum_{l=1}^{S^{m+1}} (w_{l,i}^{m+1} \times u_{q,l,r}^{m+1}), \quad (13)$$

which is initialized at the last layer (layer  $M$ ) using

$$u_{q,i,r}^M = f^M(n_{i,q}^M) \left( \left. \frac{\partial V(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} - \left. \frac{\partial a(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} \right). \quad (14)$$

The term  $v_{i,q}^m$  is backpropagated from layer to layer using

$$v_{i,q}^m = f^m(n_{i,q}^m) \sum_{l=1}^{S^{m+1}} (w_{l,i}^{m+1} \times v_{i,q}^{m+1}) + \sum_{r=1}^R \frac{\partial f^m(n_{i,q}^m)}{\partial p_r} \sum_{l=1}^{S^{m+1}} (w_{l,i}^{m+1} \times u_{q,l,r}^{m+1}), \quad (15)$$

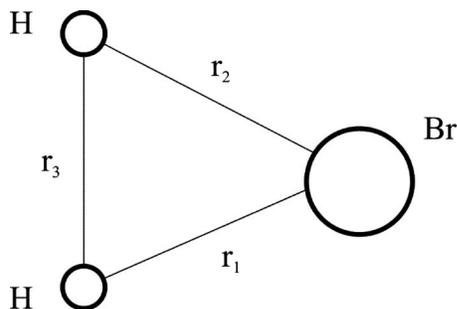
which is initialized at the last layer (layer  $M$ ) using

$$v_{i,q}^M = \sum_{r=1}^R \left( \left( \left. \frac{\partial V(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} - \left. \frac{\partial a(\mathbf{p})}{\partial p_r} \right|_{\mathbf{p}=\mathbf{p}_q} \right) \times \frac{\partial f^M(n_{i,q}^M)}{\partial p_r} \right). \quad (16)$$

Therefore, using Eqs. (11), (13), and (15), we can compute the terms needed in Eq. (10) to compute the elements of the gradient which correspond to weights in the NN. For the terms in the gradient which correspond to bias elements, Eq. (10) is replaced with Eq. (17),

$$\frac{\partial J_d(\mathbf{w})}{\partial b_i^m} = \frac{-2}{QR} \sum_{q=1}^Q v_{i,q}^m. \quad (17)$$

This completes the gradient calculation. Once the gradient has been computed, any gradient-based optimization algorithm can be used to optimize the network weights and biases (e.g., conjugate gradient, BFGS quasi-Newton, etc.). For Jacobian-based algorithms (e.g., Gauss-Newton,

FIG. 2. H<sub>2</sub>Br coordinate labels.

Levenberg–Marquardt, extended Kalman filter, etc.), a similar procedure to that described above can be used to compute the Jacobian. In our experience, both BFGS quasi-Newton and Levenberg–Marquardt algorithms have produced excellent results.

There is one other step that is required to use the CFDA algorithm, and that is the selection of  $\rho$ . The details of this selection are given by Pukrittayakamee *et al.*<sup>37</sup> We will provide a summary of the calculations here. The first step was to write  $\rho$  as

$$\rho = \frac{\lambda}{\eta^2},$$

where  $\eta$  is the calculated ratio of the maximum absolute value of the derivative (force) to the maximum absolute value of the potential energy in the training set. In this way,  $\eta$  will account for the difference in scales between  $J_f(\mathbf{w})$  and  $J_d(\mathbf{w})$ . The same value of the parameter  $\lambda$  can then be used over a variety of problems. This was tested by Pukrittayakamee *et al.*<sup>37</sup> and it was found that  $\lambda = 10^4$  produced accurate results over a wide variety of fitting problems. That is the value that we will use in the remainder of this paper.

### III. APPLICATION TO THE H<sub>2</sub>Br SYSTEM

#### A. Description of analytic potential

The system used to validate the CFDA training procedure is the H+HBr system defined in Fig. 2.

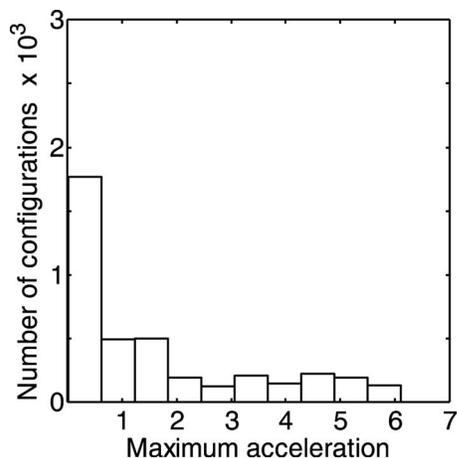


FIG. 3. Histogram of number of configurations stored and corresponding maximum acceleration when the configurations are stored at equally spaced time intervals. Acceleration is given in units of  $\text{\AA} \text{at}^{-2}$ , where one atomic time unit (atu) is  $1.019 \times 10^{-14}$  s.

The system was evaluated on the London-Eyring-Polanyi-Sato (LEPS)-type PES developed by Kuntz *et al.*,<sup>43</sup>

$$V(r_1, r_2, r_3) = \frac{Q_1}{(1+a)} + \frac{Q_2}{(1+b)} + \frac{Q_3}{(1+c)} - \left[ \frac{J_1^2}{(1+a)^2} + \frac{J_2^2}{(1+b)^2} + \frac{J_3^2}{(1+c)^2} - \frac{J_1 J_2}{(1+a)(1+b)} - \frac{J_2 J_3}{(1+b)(1+c)} - \frac{J_1 J_3}{(1+a)(1+c)} \right]^{1/2}, \quad (18)$$

where

$$\frac{Q_1}{(1+a)} = \frac{1}{2} \left\{ {}^1E(r_1) + \left[ \frac{(1-a)}{(1+a)} \right]^3 {}^3E(r_1) \right\} \quad (19)$$

and

$$\frac{J_1}{(1+a)} = \frac{1}{2} \left\{ {}^1E(r_1) - \left[ \frac{(1-a)}{(1+a)} \right]^3 {}^3E(r_1) \right\} \quad (20)$$

with similar expressions for the other terms. Morse- and “anti-Morse-type” functions are used to represent the singlet and triplet energy states, respectively,

$${}^1E(r) = D \{ \exp[-2\alpha(r-r_c)] - 2 \exp[-\alpha(r-r_c)] \} \quad (21)$$

and

$${}^3E(r) = \frac{1}{2} D \{ \exp[-2\alpha(r-r_c)] + 2 \exp[-\alpha(r-r_c)] \}, \quad (22)$$

where the Morse parameters  $D$ ,  $\alpha$ , and  $r_c$  are obtained from bond dissociation energies, fundamental vibration frequencies, and equilibrium bond distances in the usual manner. The values for all parameters for the H<sub>2</sub>Br system are given for surface I by Sudhakaran and Raff.<sup>44</sup>

This LEPS surface has a discontinuity in the gradient. The location of the discontinuity is defined by the following conditions:

$$r_2 = r_3 \quad \text{and} \quad r_1 = -\frac{1}{\alpha_1} \ln \left( \frac{k_2}{2k_1} \pm \frac{1}{2} \sqrt{\left( \frac{k_2}{k_1} \right)^2 + \frac{8P_2}{k_1}} \right), \quad (23)$$

where

$$k_1 = \left\{ D_1 - \frac{1}{2} D_1 \frac{(1-a)}{(1+a)} \right\} \exp(2\alpha_1 r_c),$$

$$k_2 = - \left\{ 2D_1 + D_1 \frac{(1-a)}{(1+a)} \right\} \exp(\alpha_1 r_c),$$

and

$$P_2 = \frac{J_2}{(1+b)}.$$

The existence of this discontinuity has caused no problems in the many MD investigations that have employed Eqs. (18)–(22) for the potential surface because the loci of points on the discontinuity seam fall at such high energies that the discontinuity is never reached for trajectories computed at energies typical of chemical experiments and calculations.

We ensured that the same is true in the investigations conducted in this research.

## B. Sampling procedure and fitting results

The database used for training the NN was obtained by computing 1000 trajectories for H+HBr using the analytic potential given by Eqs. (18)–(22). This method was originally proposed and used by Collins.<sup>28</sup> Each trajectory in a given batch was computed at a constant translational energy. As the trajectories were propagated, H<sub>2</sub>Br configurations along with their corresponding potential energies and gradients were stored at equally spaced time intervals. The ensemble of these stored configurations, energies, and gradients comprises the database for NN fitting.

To avoid sampling the discontinuity, NN training data were collected during MD trajectories that were computed at relative translational energies of 0.8, 1.0, and 1.2 eV and a maximum impact parameter  $b_{\max}$  of 0.2 Å. The impact parameter in individual trajectories was randomly selected over the range from zero to  $b_{\max}$ . Since the minimum energy at the discontinuity lies at 1.65 eV above the reactant H+HBr energy, the sampling trajectories will never reach the discontinuities at the energies used in the sampling.

For every trajectory, the bond distance between H–Br was stretched by  $(0.1 \text{ \AA}) \cdot r_n$ , where  $r_n$  is a random number selected from a uniform distribution over the interval [0,1]. Since the HBr vibrational amplitude in the ground vibrational state is  $\sim 0.1 \text{ \AA}$ , this procedure effectively averages over vibrational phase. In all cases, HBr was taken to have zero rotational energy. Initially, HBr was placed in its equilibrium configuration and given a vibrational kinetic energy corresponding to its zero-point energy plus a small additional potential energy due to the displacement from its equilibrium configuration. The maximum value of this potential energy can be obtained from Eq. (21) with  $r=r_e+0.1 \text{ \AA}$ . The velocity Verlet algorithm was employed to integrate the equations of motion during the MD simulations.

To effectively sample the configuration space with something close to a uniform density of points over all regions,

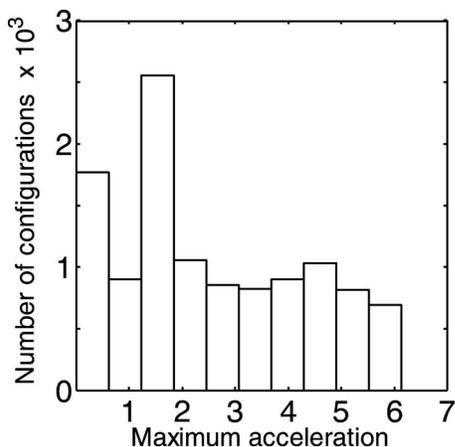


FIG. 4. Histogram of a number of configurations stored and corresponding maximum acceleration when the time interval for sampling is obtained from Eqs. (24) and (25). Acceleration is given in units of  $\text{\AA} \text{ au}^{-2}$ , where one atomic time unit (atu) is  $1.019 \times 10^{-14} \text{ s}$ .

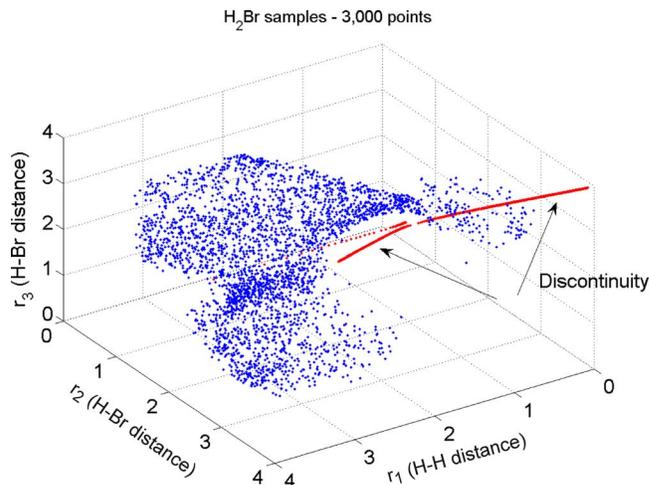


FIG. 5. (Color online) Data sampling and location of the discontinuity in the Sudhakaran–Raff analytic potential denoted as surface I in Ref. 33. The curve in the figure shows the location of the discontinuity.

the time interval between sampling  $\tau$  during the trajectories must be a function of the atomic accelerations. If configurations are stored at every integration step, the database will include more points from regions where the forces are small than from regions characterized by large forces simply because the atoms will spend less time in regions where the forces are large. To minimize this effect, we have adopted the following algorithms for the sampling time interval:

$$\tau = \text{trunc}[5(a_{\max})^{-1}]\Delta t \quad \text{if } \text{trunc}[5(a_{\max})^{-1}] > 0$$

and

$$\tau = \text{trunc}[5(a_{\max})^{-1}]\Delta t + \Delta t \quad \text{if } \text{trunc}[5(a_{\max})^{-1}] = 0, \quad (24)$$

where

$$a_{\max} = \max[a_1, a_2, a_3]$$

and

$$a_i = (m_i)^{-1}|P_i|. \quad (25)$$

In Eqs. (24) and (25),  $a_i$  is the absolute value of the acceleration of atom  $i$ ,  $m_i$  and  $P_i$  are the corresponding atomic

TABLE I. rms test set errors for Bayesian regularization and CFDA for ten different NNs.

Bayesian regularization		CFDA	
Potential (eV)	Forces (eV/Å)	Potential (eV)	Forces (eV/Å)
$3.3035 \times 10^{-4}$	$7.6882 \times 10^{-3}$	$1.4525 \times 10^{-4}$	$1.8398 \times 10^{-3}$
$6.5614 \times 10^{-4}$	$9.4262 \times 10^{-3}$	$1.5777 \times 10^{-4}$	$1.6662 \times 10^{-3}$
$3.5913 \times 10^{-4}$	$5.8481 \times 10^{-3}$	$1.5975 \times 10^{-4}$	$1.7043 \times 10^{-3}$
$2.3396 \times 10^{-4}$	$4.3689 \times 10^{-3}$	$1.9860 \times 10^{-4}$	$2.4132 \times 10^{-3}$
$4.2139 \times 10^{-4}$	$7.7385 \times 10^{-3}$	$1.4370 \times 10^{-4}$	$1.2003 \times 10^{-3}$
$3.0202 \times 10^{-4}$	$7.6737 \times 10^{-3}$	$2.3570 \times 10^{-4}$	$3.4086 \times 10^{-3}$
$5.3780 \times 10^{-4}$	$2.3082 \times 10^{-2}$	$1.1207 \times 10^{-4}$	$1.3807 \times 10^{-3}$
$5.0875 \times 10^{-4}$	$7.2271 \times 10^{-3}$	$1.3950 \times 10^{-4}$	$1.4527 \times 10^{-3}$
$1.4652 \times 10^{-3}$	$1.2323 \times 10^{-2}$	$1.4333 \times 10^{-4}$	$2.5245 \times 10^{-3}$
$7.8513 \times 10^{-4}$	$9.3164 \times 10^{-3}$	$2.0005 \times 10^{-4}$	$1.5358 \times 10^{-3}$

TABLE II. Median rms test set errors for Bayesian regularization and CFDA. Training set contains 3000 data points. Test set contains 467 000 data points. The potential errors are expressed in eV and the force errors are expressed in eV/Å.

Bayesian regularization			CFDA		
Potential	Force	Second derivative	Potential	Force	Second derivative
$4.65 \times 10^{-4}$	$7.71 \times 10^{-3}$	$1.10 \times 10^{-1}$	$1.51 \times 10^{-4}$	$1.68 \times 10^{-3}$	$2.76 \times 10^{-2}$

mass and momentum vector, respectively, and  $\Delta t$  is the integration step size.  $a_{\max}$  is the maximum of the acceleration of the three atoms and the operation  $\text{trunc}(x)$  yields the integer part of  $x$ . The value of  $\tau$  is periodically updated during the trajectory. Thus, as  $a_{\max}$  becomes large when the forces are large,  $\tau$  will approach  $\Delta t$ , and we sample at every integration point in the trajectory. In regions of small force where  $a_{\max}$  is small,  $\tau$  will assume larger values, and we will sample less frequently. The numerator of Eq. (24) is system dependent and must be determined empirically as was done in the present work.

Figures 3 and 4 show the operation of the time interval algorithm given by Eqs. (24) and (25). Each figure shows a histogram of the number of configurations stored as a function of the maximum acceleration for 100 trajectories on the H+HBr analytic surface. The results reported in Fig. 3 were obtained by sampling at equally spaced time intervals throughout all trajectories. The sampling for Fig. 4 was executed using Eqs. (24) and (25). When sampling is done at constant time intervals, a preponderance of the sampled points lie in regions where forces are small. When the weighted sampling described by Eqs. (24) and (25) is employed, all regions are sampled much more uniformly.

Figure 5 shows the regions where the trajectories were sampled and the location of the discontinuity, which is indicated by the curve in the figure.

A total of 470 000 data points were sampled. From these available data, 3000 points were selected at random to train the NN. When the reaction of interest is a single two-center bond dissociation gas-phase reaction in a three- or four-atom system, the database employed generally contains 300–6500 points. For a representative sampling of 23 such systems, the reader may consult Table II of Ref. 45. Consequently, the number of training points employed here is typical for reactions of this degree of complexity. The 467 000 points not included in the training set were used as the test set. This testing set is far more extensive than that usually employed in most applications of NNs to potential-energy surfaces. Since the purpose of the present investigation is validation of the CFDA method, we have elected to employ such a test set

to ensure that NNs produced by the method are thoroughly and completely tested in all important regions of the H<sub>2</sub>Br configuration space.

Because the results can be sensitive to the data selected and to the initial NN weights, ten different NNs were trained. Each network was trained with a different selection of data points and different initial weights. The median results on the complete 470 000 data points were then used to assess the accuracy of the NN surface. The results for each of the ten NNs are given in Table I. The median values are given in Table II. As can be seen, all NNs give very close to identical results.

Several choices for the number of neurons in the hidden layer were examined. The extremely small interpolation errors on the testing set demonstrate that our final choice of 150 neurons is close to the optimum number. The implementation of the NNs is as follows:

$$\mathbf{p} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \quad (26)$$

$$\mathbf{a}^1 = \tanh(\mathbf{W}^1 \mathbf{p} + \mathbf{b}^1), \quad (27)$$

$$a^2 = \mathbf{W}^2 \mathbf{a}^1 + b^2, \quad (28)$$

where  $\mathbf{W}^1$  is a  $150 \times 3$  weight matrix,  $\mathbf{b}^1$  is a  $150 \times 1$  bias vector,  $\mathbf{W}^2$  is a  $1 \times 150$  weight matrix, and  $b^2$  is a scalar bias. The network output  $a^2$  is the system potential, which is trained to match the potential energy and the associated gradients using the CFDA method.

Table II shows the median rms test set errors for the Bayesian regularization method, where only the potential is fitted, and the CFDA algorithm. In previous work, the Bayesian regularization method has produced the best performance; therefore, we will use it as a benchmark. Here, we can see that the CFDA algorithm provides significantly smaller errors on both the potential surface and the force field.

TABLE III. Median rms test set errors for Bayesian regularization and CFDA. Training set contains 1500 data points. Test set contains 468 500 data points. The potential errors are expressed in eV and the force errors are expressed in eV/Å.

Bayesian regularization			CFDA		
Potential	Force	Second derivative	Potential	Force	Second derivative
$1.27 \times 10^{-3}$	$1.39 \times 10^{-2}$	$1.09 \times 10^{-1}$	$9.82 \times 10^{-4}$	$7.49 \times 10^{-3}$	$5.65 \times 10^{-2}$

TABLE IV. Median rms test set errors for Bayesian regularization and CFDA. Training set contains 750 data points. Test set contains 469 250 data points. The potential errors are expressed in eV and the force errors are expressed in eV/Å.

Bayesian regularization			CFDA		
Potential	Force	Second derivative	Potential	Force	Second derivative
$3.52 \times 10^{-3}$	$2.28 \times 10^{-2}$	$2.00 \times 10^{-1}$	$2.22 \times 10^{-3}$	$1.19 \times 10^{-2}$	$9.81 \times 10^{-2}$

The errors shown in Table II were obtained with a network trained with only 3000 data points. The errors were measured over a test set of 467 000 data points. The fact that the test set errors for the CFDA algorithm are lower than those for the Bayesian regularization algorithm (which was designed to prevent overfitting) on such a large test set demonstrates that the CFDA algorithm (without a validation set) does a better job of preventing overfitting than the standard methods. One would not generally have such a large test set available. We have used a large one in this case in order to clearly demonstrate the reduction in overfitting when using CFDA methods (without a validation set). The large test set also permits us to demonstrate that the CFDA algorithm provides an excellent fit over the entire configuration space important in the  $\text{H}_2\text{Br}$  reaction dynamics.

To provide further demonstrations of the ability of the CFDA algorithm to reduce overfitting without a validation set, we trained networks with even fewer data points. Table III shows the results of training with 1500 randomly selected data points and Table IV shows the results of training with only 750 data points. In both cases, the remainder of the 470 000 points is used for testing. The test errors increase in size as the number of training points is decreased. This is to be expected since we are not covering the input space adequately. However, in each case, the CFDA error on the test set is lower than the Bayesian regularization error. Since Bayesian regularization is the best standard method we have found for reducing overfitting, this presents a strong case that the CFDA method does a better job of reducing overfitting without a validation set, even when the training set is very small. We also note that the very small errors for the second derivatives indicate that the first derivative cannot be

changed rapidly between data points. In this regard, we note that no one would attempt to study a three-dimensional system with three energetically open reaction channels, inelastic scattering, abstraction, and exchange with fewer than 750 data points.

In a previously reported study,<sup>37</sup> we employed the CFDA to fit five other systems that include four analytical potentials and a database for  $\text{Si}_5$  obtained from electronic structure calculations. In each case, every test that we have run shows that CFDA training (without a validation set) has produced smaller, out-of-sample testing error than early stopping (with a validation set) or Bayesian regularization (without a validation set). This indicates that CFDA training does a better job of preventing overfitting than the standard methods currently in use.

Figure 6 shows a comparison of the LEPS potential energies and the corresponding NN results for the test set configurations. If the NN fit were perfect, all points would fall on a 45° line. Figure 7 shows the distribution of errors in the potential for the median network. The median rms deviation of the NN and Sudhakaran/Raff energies is  $1.51 \times 10^{-4}$  eV ( $1.2 \text{ cm}^{-1}$ ). Clearly, the fit is nearly exact. Since the energies in the database span a range of about 2 eV, this interpolation accuracy represents an average percent error of about 0.0075%.

Figure 8 shows a comparison of the LPS force along  $r_1$  and the corresponding NN results for the test set configurations. Figure 9 shows the distribution of errors in the  $r_1$  force for the median network. The median rms deviation of the NN and Sudhakaran–Raff  $r_1$  force is  $1.68 \times 10^{-3}$  eV/Å. Since the forces span a range of about 10.5 eV/Å, this interpolation error represents a percent error of about 0.016%.

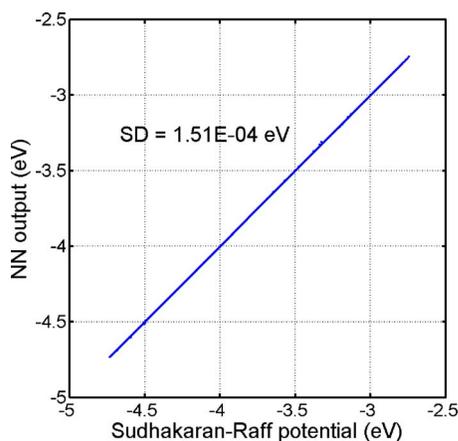


FIG. 6. (Color online) Comparison of Sudhakaran–Raff energies with the predictions of the median NN for a test set of 470 000 configurations.

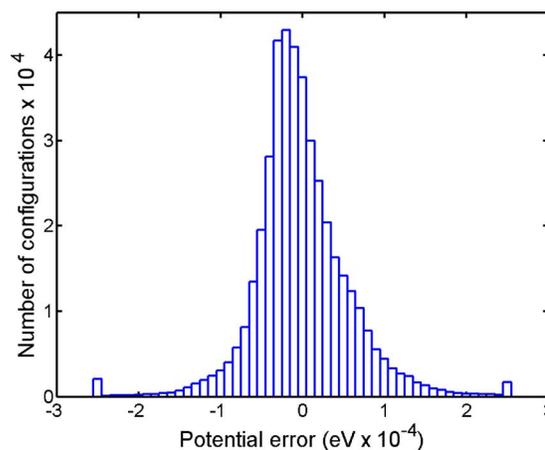


FIG. 7. (Color online) Distribution of interpolation errors in the potential for the median NN.

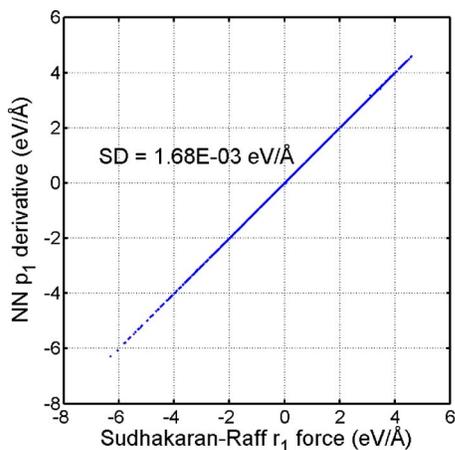


FIG. 8. (Color online) Comparison of Sudhakaran–Raff forces ( $r_1$ ) with the predictions of the median NN for a test set of 470 000 configurations.

These extremely small errors over such an extensive testing set demonstrate that the NN obtained from the CFDA method produces an excellent fit in all regions of configuration space which include the repulsive walls, potential minima, and the flat regions of the potential. The excellent fit also demonstrates that we do not experience overfitting in spite of the fact that the CFDA method does not employ a validation set. If overfitting were present, such a small testing set error could not be obtained.

### C. Reaction dynamics

The reaction dynamics of H+HBr were investigated using the median NN trained as described in section III B. The three open reaction or scattering channels are

- (1) Exchange:  $\text{H}^2 + \text{H}^1\text{Br} \rightarrow \text{H}^2\text{Br} + \text{H}^1$ ,
- (2) Abstraction:  $\text{H}^2 + \text{H}^1\text{Br} \rightarrow \text{H}^1\text{H}^2 + \text{Br}$ , and
- (3) No reaction.

The reaction channel resulting in the dissociation of all three atoms was energetically closed for our tests.

MD trajectories were run using both the median NN surface and the LEPS-type Sudhakaran–Raff potential-energy surface. The velocity Verlet algorithm was employed to inte-

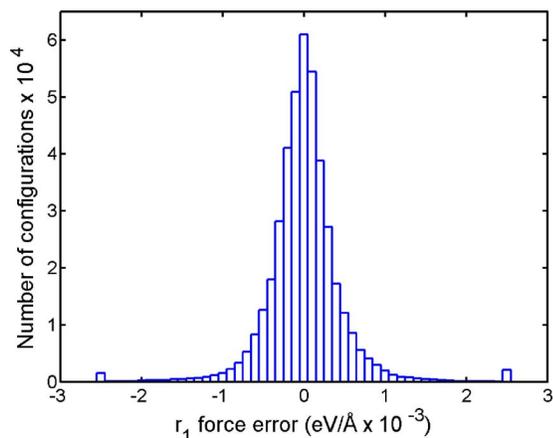


FIG. 9. (Color online) Distribution of interpolation errors in the force ( $r_1$ ) for the median NN.

TABLE V. Reaction yields out of 1000 trajectories at 1.2 eV.

	Analytical potential	NN
Exchange	728	728
Abstraction	3	3
No reaction	269	269

grate the equations of motion during the MD simulations. A total of 1000 MD trajectories were run at relative translational energies of 0.8, 1.0, and 1.2 eV. None of these trajectories were contained in the set used for sampling the configuration space to obtain the database. The maximum impact parameter and the initial internal HBr energies were selected in the same manner as previously described for sampling trajectories. The impact parameters for the 1000 MD simulations were selected randomly over the range from zero to  $b_{\text{max}}$ .

Table V shows the number of trajectories reacting into each energetically open channel computed using the analytical potential and the NN at a relative translational energy of 1.2 eV. The results for each of the 1000 trajectories were identical.

Figures 10 and 11 show the temporal dependence of the difference between the bond distances and corresponding energies, respectively, computed using the analytical potential and the NN during the MD simulation for one randomly chosen trajectory. The figures are both highly magnified to allow us to appropriately illustrate the very small errors. The maximum absolute deviation for the three bond distances in this trajectory is 0.0003 Å. During most of the trajectory, the deviations are less than 0.0001 Å. The absolute deviation of the total energy lies between 0.0001 and 0.0002 eV over most of the trajectory. This particular trajectory shows a momentary increase to 0.0006 eV. These results are in accordance with the  $1.51 \times 10^{-4}$  eV standard energy deviation shown in Figs. 6 and 7. Although Figs. 10 and 11 show the results for a single trajectory, these figures are typical of all 1000 trajectories investigated.

These extremely small differences seen in Figs. 10 and 11 demonstrate that NNs trained using the CFDA method produce essentially a point-by-point match of trajectories computed on the analytic and NN potential surfaces. That is,

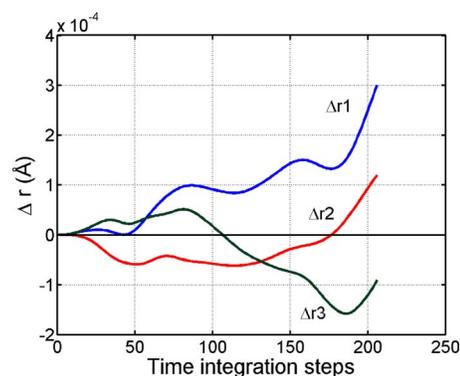


FIG. 10. (Color online) Difference between bond distances computed using the analytical potential and the NN during the MD simulation vs time integration steps of 0.1 fs.

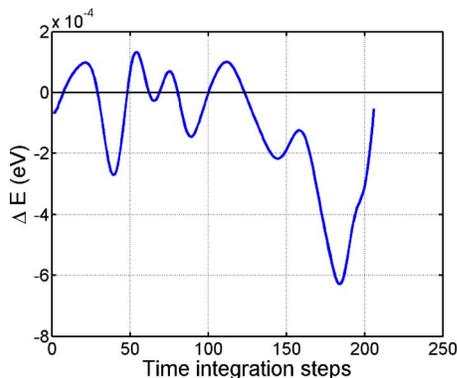


FIG. 11. (Color online) Difference between energies computed using the analytical potential and the NN during the MD simulation vs time integration steps of 0.1 fs.

plots of the temporal variation in  $r_i$  ( $i=1,2,3$ ) for a given trajectory on the analytic and NN surfaces are superimposable within the accuracy of the plot. Previous attempts to obtain such point-by-point matching of trajectories on analytic surfaces and numerical fits to such surfaces have failed.<sup>46</sup>

#### IV. CONCLUSIONS

A CFDA method for using NNs to simultaneously fit the potential-energy hypersurfaces and their corresponding force fields has been described and tested. The method generalizes and extends the work of Witkoskie and Doren.<sup>39</sup> With this method, it is possible to accurately fit both the potential and the force field with a single network, whose only output neuron represents the potential. This feature greatly simplifies the architecture of the NN. The formulation provides for differential weighting of function versus gradient fitting. The training process adjusts the NN weights so that the NN output matches the potential, while at the same time the derivatives of the NN output with respect to its inputs match the corresponding force fields.

The CFDA procedure has been tested on an analytical potential for the  $\text{H}_2\text{Br}$  system. It is shown that because the NN must fit both the function and the gradients, CFDA training without a validation set reduces overfitting better than the methods that use a validation set. Similar results have been reported for five other systems.<sup>37</sup> The extremely small interpolation errors obtained on an extensive 467 000 point testing set show that the CFDA produces a NN that fits well in all regions of the potential, repulsive walls, potential minima, and in the relatively flat portions of the potential. The resulting NN fit is almost indistinguishable from the analytic surface. Reaction dynamics show that the fitting method produces an almost point-by-point match with the analytic surface for every trajectory.

The CFDA can be directly applied to fitting potential surfaces whose databases are obtained by *ab initio* methods. We have previously reported such application for the  $\text{Si}_5$  system.<sup>37</sup> There will be, of course, more computational difficulties in executing the electronic structure calculations to obtain the potential and its gradients, but once completed, the application of the CFDA to fitting and interpolating the re-

sulting database is the same. The method is also applicable to systems of larger dimensionality without modification of the underlying equations. We have previously developed methods for sampling systems containing six atoms that are undergoing simultaneous reaction into six different two- and three-center reaction channels.<sup>1,2</sup> In such cases, the input vector to the NN will be larger and more neurons will be required. There will be substantially more gradients in Eq. (2), but the equations underlying the CFDA method are the same. The additional difficulty of treating larger systems using *ab initio* methods lies primarily in the increased computational time required, but this is to be expected as the complexity and size of the system increase.

#### ACKNOWLEDGMENTS

This project is funded by the National Science Foundation, Division of Civil, Mechanical, and Manufacturing Innovation (CMMI) (Grant No. DMI-0457663). The authors thank Dr. Jocelyn Harrison, Program Director for Materials Processing and Manufacturing, for the interest and support of this work. One of the authors (R.K.) also thanks A. H. Nelson, Jr. Endowed Chair for additional financial support.

- <sup>1</sup>L. M. Raff, M. Malshe, M. Hagan, D. I. Doughan, M. G. Rockley, and R. Komanduri, *J. Chem. Phys.* **122**, 084104 (2005).
- <sup>2</sup>M. Malshe, L. M. Raff, M. G. Rockley, M. Hagan, P. M. Agrawal, and R. Komanduri, *J. Chem. Phys.* **127**, 134105 (2007).
- <sup>3</sup>S. Lorenz, A. Groß, and M. Scheffler, *Chem. Phys. Lett.* **395**, 210 (2004).
- <sup>4</sup>D. F. R. Brown, M. N. Gibbs, and D. C. Clary, *J. Chem. Phys.* **105**, 7597 (1996).
- <sup>5</sup>T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, *J. Chem. Phys.* **103**, 4129 (1995).
- <sup>6</sup>T. B. Blank and S. D. Brown, *Anal. Chim. Acta* **277**, 273 (1993).
- <sup>7</sup>S. Hobday, R. Smith, and J. BelBruno, *J. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* **153**, 247 (1999).
- <sup>8</sup>E. Tafeit, W. Estelberger, R. Horejsi, R. Moeller, K. Oettl, K. Vrecko, and G. J. Reibnegger, *J. Mol. Graphics* **14**, 12 (1996).
- <sup>9</sup>H. Gassner, M. Probst, A. Lauenstein, and K. Hermansson, *J. Phys. Chem. A* **102**, 4596 (1998).
- <sup>10</sup>S. Manzhos, Z.-G. Wang, R. Dawes, and T. Carrington, Jr., *J. Phys. Chem. A* **110**, 5295 (2006).
- <sup>11</sup>S. Manzhos and T. Carrington, Jr., *J. Chem. Phys.* **125**, 194105 (2006).
- <sup>12</sup>S. Manzhos and T. Carrington, Jr., *J. Chem. Phys.* **125**, 084109 (2006).
- <sup>13</sup>S. Manzhos and T. J. Carrington, Jr., *J. Chem. Phys.* **127**, 014103 (2007).
- <sup>14</sup>J. Ludwig and D. G. Vlachos, *J. Chem. Phys.* **127**, 154716 (2007).
- <sup>15</sup>J. Ischtwan and M. A. Collins, *J. Chem. Phys.* **100**, 8080 (1994).
- <sup>16</sup>K. A. Nguyen, I. Rossi, and D. G. Truhlar, *J. Chem. Phys.* **103**, 5522 (1995).
- <sup>17</sup>T. Ishida and G. C. Schatz, *J. Chem. Phys.* **107**, 3558 (1997).
- <sup>18</sup>K. C. Thompson, M. J. T. Jordan, and M. A. Collins, *J. Chem. Phys.* **108**, 8302 (1998).
- <sup>19</sup>L. A. Pederson, G. C. Schatz, T.-S. Ho, T. Hollebeek, H. Rabitz, L. B. Harding, and G. Lendvay, *J. Chem. Phys.* **110**, 9091 (1999).
- <sup>20</sup>R. P. A. Bettens and M. A. Collins, *J. Chem. Phys.* **111**, 816 (1999).
- <sup>21</sup>R. P. A. Bettens, M. J. T. Jordan, D. H. Zhang, and M. A. Collins, *J. Chem. Phys.* **112**, 10162 (2000).
- <sup>22</sup>D. H. Zhang, M. A. Collins, and S.-Y. Lee, *Science* **290**, 961 (2000).
- <sup>23</sup>L. A. Pederson, G. C. Schatz, T. Hollebeek, T.-S. Ho, H. Rabitz, and L. B. Harding, *J. Phys. Chem. A* **104**, 2301 (2000).
- <sup>24</sup>K. Song and M. A. Collins, *Chem. Phys. Lett.* **335**, 481 (2001).
- <sup>25</sup>G. E. Moyano and M. A. Collins, *J. Chem. Phys.* **119**, 5510 (2003).
- <sup>26</sup>M. A. Collins and L. Radom, *J. Chem. Phys.* **118**, 6222 (2003).
- <sup>27</sup>R. Z. Pascual, G. C. Schatz, G. Lendvay, and D. Troya, *J. Phys. Chem. A* **106**, 4125 (2002).
- <sup>28</sup>M. A. Collins, *Theor. Chem. Acc.* **108**, 313 (2002).
- <sup>29</sup>C. Crespos, M. A. Collins, E. Pijper, and G. J. Kroes, *Chem. Phys. Lett.*

- 376, 566 (2003).
- <sup>30</sup>C. Crespos, M. A. Collins, E. Pijper, and G. J. Kroes, *J. Chem. Phys.* **120**, 2392 (2004).
- <sup>31</sup>G. G. Maisuradze, D. L. Thompson, A. F. Wagner, and M. Minkoff, *J. Chem. Phys.* **119**, 10002 (2003).
- <sup>32</sup>A. Kawano, Y. Guo, D. L. Thompson, A. F. Wagner, and M. Minkoff, *J. Chem. Phys.* **120**, 6414 (2004).
- <sup>33</sup>Y. Guo, A. Kawano, D. L. Thompson, A. F. Wagner, and M. Minkoff, *J. Chem. Phys.* **121**, 5091 (2004).
- <sup>34</sup>M. Malshe, R. Narulkar, L. M. Raff, M. Hagan, S. Bukkapatnam, P. M. Agrawal, and R. Komanduri, "Development of generalized potential-energy surfaces (GPES) using many-body expansion, neural networks (NN), and moiety energy (ME) approximation," *J. Chem. Phys.* (submitted).
- <sup>35</sup>P. M. Agrawal, L. M. Raff, M. Hagan, and R. Komanduri, *J. Chem. Phys.* **124**, 134306 (2006).
- <sup>36</sup>H. M. Le and L. M. Raff, *J. Chem. Phys.* **128**, 194310 (2008).
- <sup>37</sup>A. Pukrittayakamee, M. Hagan, L. M. Raff, S. Bukkapatnam, and R. Komanduri, *Intelligent Engineering Systems Through Artificial Neural Networks*, ANNIE Vol. 17 (ASME Press, Fairfield, NJ, 2007).
- <sup>38</sup>M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design* (PWS Publishing Company, Boston, MA, 1996).
- <sup>39</sup>J. B. Witkoskie and D. J. Doren, *J. Chem. Theory Comput.* **1**, 14 (2005).
- <sup>40</sup>J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983).
- <sup>41</sup>D. Marquardt, *J. Soc. Ind. Appl. Math.* **11**, 431 (1963).
- <sup>42</sup>D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Nature (London)* **323**, 533 (1986).
- <sup>43</sup>P. J. Kuntz, E. M. Nemeth, J. C. Polanyi, S. D. Rosner, and C. E. Young, *J. Chem. Phys.* **44**, 1168 (1966).
- <sup>44</sup>M. P. Sudhakaran and L. M. Raff, *Chem. Phys.* **95**, 165 (1985).
- <sup>45</sup>M. Malshe, R. Narulkar, L. M. Raff, M. Hagan, S. Bukkapatnam, and R. Komanduri, *J. Chem. Phys.* **129**, 044111 (2008).
- <sup>46</sup>N. Sathyamurthy and L. M. Raff, *J. Chem. Phys.* **63**, 464 (1975).