Kohn-Sham regularizer for spin density functional theory and weakly correlated systems

Bhupalee Kalita  
Department of Chemistry, University of California, Irvine

Ryan Pederson and Jiedun Chen  (陈捷伦)  
Department of Physics and Astronomy, University of California, Irvine

Li Li  (李力)  
Google Research

Kieron Burke  
Department of Chemistry, University of California, Irvine and  
Department of Physics and Astronomy, University of California, Irvine

Kohn-Sham regularizer (KSR) is a machine learning approach that optimizes a physics-informed exchange-correlation functional within a differentiable Kohn-Sham density functional theory (DFT) framework. We generalize KSR to spin DFT and create local, semilocal, and nonlocal approximations for the exchange-correlation functional. We explore KSR for weakly correlated systems, by training on atoms and testing on molecules at equilibrium. The generalization error from our semilocal approximation is comparable to other differentiable approaches. Our nonlocal functional outperforms any existing machine learning functionals by predicting the ground-state energies of the test systems with a mean absolute error of 2.7 milli-Hartrees.

1. INTRODUCTION

Determining the ground-state properties of many-electron systems is fundamental to molecular modeling problems in chemical and material sciences. However, solving the Schrödinger equation explicitly for more than a few hundred electrons is computationally intractable. Among several methods of approximation, Kohn-Sham density functional theory (KS-DFT or simply DFT) [1, 2], a method based on the electron density distribution rather than the many-electron wave function, provides chemically useful results with $O(N^3)$ scaling for an $N$-electron system [3]. DFT is formally exact, but the exchange-correlation (XC) energy, resulting from the quantum-mechanical interaction between electrons, must be approximated in practice. Hundreds of XC energy functional approximations have been formulated in the past few decades [4]. Functionals can be designed non-empirically, for example using physics and chemical-based intuition and satisfying known exact constraints [5], or can involve some fitting to reference data [6]. However, in any approach, these functional approximations do not yield chemical accuracy in general, that is, with errors less than 1.6 milli-Hartrees (mH) in atomic units (or 1 kcal/mol). Improving the accuracy of XC functional approximations often incurs additional computational cost in the practical DFT calculation [7]. However, there is no systematic way in general to develop and improve XC functional approximations.

In recent years, machine learning (ML) has been used to find better DFT approximations. Attempts have been made to enhance either the speed or accuracy of DFT. Some used ML techniques to enhance computational efficiency by approximating the non-interacting kinetic energy without solving the KS equations [8–11]. In an effort to improve the accuracy of ML-DFT, a significant leap was achieved by Nagai et al. [12], who used a neural network (NN) model to approximate the XC functional and trained it with high accuracy coupled cluster (CCSD(T)) energies and densities of just three small molecules, while self-consistently solving the KS equations. This functional impressively generalized to 148 small molecules [13] to predict their energies and densities with accuracies comparable to human-designed functionals. However, the test set atomization energies were not chemically accurate. Also, they didn’t have access to gradient information and were therefore limited to a gradient-free optimization scheme, which is inherently slow, often suffers poor convergence issues, and is difficult to scale to more complex NN models.

In DFT, many useful properties are extracted from the density, although an XC functional approximation need not produce accurate densities along with accurate energies [14]. In KS-DFT, we calculate the density self-consistently, and there is a nonlinear dependence of the XC functional on the density. Learning this relationship requires not only the ground truth mapping of the functional inputs to outputs but also how the functional performs in the underlying process. Hence the use of differentiable programming [15] becomes more intuitive [16]. With differentiable programming, conditioning the networks with physical insights becomes much simpler, and it can further help to ease the process of training.

Recently, Li et al. [17] made a valuable step in
this direction by considering the entire DFT self-consistent calculation as a differentiable program. They implemented an end-to-end differentiable DFT code for 1-dimensional (1D) systems using JAX [18], a library that provides differentiation, vectorization, just-in-time compilation, and other composable transformations of Python and NumPy programs [19]. They parameterized the XC functional with an NN which incorporated non-local information about the density, along with known physical constraints. The self-consistent KS calculations were embedded into the training process by backpropagating the gradients through the KS iterations. It was dubbed the Kohn-Sham regularizer (KSR). It could yield chemically accurate energies for uniformly separated 1D hydrogen chains at any separation by training on highly accurate energies and densities from only a few separations.

Following a similar approach, Kasim and Vinko [20] implemented an end-to-end differentiable DFT code in 3D for Gaussian-type orbitals and trained local and semi-local NN-based XC functional approximations, evaluating performance on small molecules. In another work, Dick et al. [21] constructed a semilocal XC functional that was carefully curated to account for several known exact conditions and pretrained to match SCAN, a popular meta-GGA functional [22]. While both of these works explore the generalizability of ML approximations for weakly correlated molecules with differentiable DFT codes, they do not incorporate global information, and their accuracy is limited to that of human-designed semilocal functionals. A slightly different approach involves introducing an ML correction term to a nonempirical or semi-empirical XC functional within a KS-DFT self-consistent framework [23, 24]. In such an approach, only a portion of the XC energy is approximated using ML and the functionals retain the characteristics of the baseline XC functional used. The recently proposed ML local hybrid functional, DM21 [25], addresses spin-symmetry breaking and delocalization error in DFT functionals. Consequently, it performs well on several main-group benchmark datasets and also correctly dissociates molecules. Unlike KSR, this functional is trained on large datasets of highly accurate reaction energies (not densities) in the loss function without explicitly supervising the self-consistent iterations.

Li et al. [17] explored the generalizability of KSR for a few strongly correlated systems with stretched bonds (near the dissociation limit). Examples were limited to spin-unpolarized hydrogen chains. In the present work, we test the KSR for a few heteronuclear molecules at equilibrium (i.e. weakly correlated molecules). We propose a spin-polarized version of the KSR and create local, semilocal, and nonlocal approximations to the XC functionals to deal with spin-polarized systems. We show that by training spin-polarized KSR on just a few atoms and ions, we find extremely accurate energies and densities for several molecules (near chemical accuracy). Our study exemplifies the usefulness of differentiable programming in formulating physics-informed models and demonstrates the scope of such trained models in terms of generalizability and applicability to a broader class of unseen systems.

2. Kohn-Sham Density Functional Theory (KS-DFT)

The practical implementation of DFT involves solving the Kohn-Sham (KS) equations to calculate the ground-state electron density,

\[
\left\{ -\frac{1}{2} \nabla^2 + v_{\text{e}}[n](\mathbf{r}) \right\} \phi_i(\mathbf{r}) = \epsilon_i \phi_i(\mathbf{r}).
\]

The electron density, \(n(\mathbf{r})\), is the sum of the probability density over all occupied one-electron KS orbitals, \(n(\mathbf{r}) = \sum_i |\phi_i(\mathbf{r})|^2\). The KS potential, \(v_{\text{e}}[n](\mathbf{r})\), contains the external one-body potential, the Hartree potential, and the XC potentials,

\[
v_{\text{e}}[n](\mathbf{r}) = v(\mathbf{r}) + \sum_i \epsilon_i |\phi_i(\mathbf{r})|^2 = \frac{1}{2} \nabla^2 + v_{\text{e}}[n](\mathbf{r}) + v_{\text{xc}}[n](\mathbf{r}).
\]

The XC potential is the functional derivative of the XC energy, \(E_{\text{xc}}[n]\), with respect to the electron density [2],

\[
v_{\text{xc}}[n](\mathbf{r}) = \frac{\delta E_{\text{xc}}[n](\mathbf{r})}{\delta n(\mathbf{r})}.
\]

We can express \(E_{\text{xc}}[n]\) in terms of an XC energy density per electron, \(\epsilon_{\text{xc}}[n](\mathbf{r})\):

\[
E_{\text{xc}}[n] = \int d^3r \epsilon_{\text{xc}}[n](\mathbf{r}) n(\mathbf{r}).
\]

The exact form of \(E_{\text{xc}}\) is not known and must be approximated [5, 26]. Since the potential depends on the density which is in turn obtained from the solution of the KS equations, a self-consistent solution is sought by a fixed-point iteration. Once self-consistency is reached, the ground-state energy is calculated from the density by summing the non-interacting kinetic energy, \(T_{\mathbf{0}}\), the external potential energy, \(V\), the Hartree energy, \(U\), and the XC energy,

\[
E_0 = T_{\mathbf{0}}[n] + V[n] + U[n] + E_{\text{xc}}[n].
\]

The computational efficiency is also affected by the level of approximation used for the XC functional [27], and the cost generally increases with the complexity of these approximations.

3. 1D Model Systems

In 1D we model the electron-electron interaction with an exponential,

\[
v_{\text{exp}}(x) = A \exp(-\kappa |x|),
\]

where the parameters, \(A = 1.071295\) and \(\kappa^{-1} = 2.385345\) are adjusted to mimic the soft-Coulomb interaction [28].
Similarly, the external potentials for a 1D molecular system are expressed as
\[ v(x) = -\sum_j Z_j v_{\text{exp}}(x - x_j), \tag{6} \]
where \( Z_j \) is the nuclear charge and \( x_j \) is the position of the \( j^{th} \) nucleus. This allows us to create 1D analogs of atomic systems and linear molecules, such as BeH\(_2\). Density matrix renormalization group (DMRG) [29] can be used to efficiently generate highly accurate benchmark energies and densities for these 1D analog systems. Similarly, we can address such systems using 1D KS-DFT calculations with suitable XC energy functional approximations, such as the 1D local spin-density approximation (LSDA) which was constructed in Ref. 28 from the 1D exponentially repelling uniform electron gas.

4. KOHN-SHAM REGULARIZER (KSR) AND PHYSICS-INFORMED NN XC FUNCTIONALS

In essence, KSR is a ML-DFT regularization technique that utilizes a differentiable analog of the standard self-consistent DFT computational flow during training to train a suitable parameterized model for \( E_{\text{xc}}[n] = E_{\text{xc},\theta}[n] \), where \( \theta \) are trainable parameters [17]. In this work, we consider NN-based (neural) XC models, but KSR as a regularization technique can apply more broadly to any differentiable model choice. The loss function was based on total energy and density results. Energy and density errors are minimized with respect to the reference DMRG calculations in one dimension [28]. In Ref. 17 it was shown that the KS equations work as an implicit regularizer as the gradients are backpropagated through the KS iterations to optimize the neural XC functionals during the training process. Learning the dependence of energy and density across KS iterations in this way makes KSR data-efficient and improves its generalization.

Knowledge of physical properties and constraints in the exact XC functional can help guide the construction of a neural XC approximation. The NN that parameterizes the XC functional in KSR is carefully curated to account for a few of the expected behaviors of the exact XC-functional. The exact functional is known to be nonlocal. Nonlocality is facilitated by adding a global convolution layer to the 1D exponential repelling uniform electron gas.

An end-to-end differentiable DFT code requires a continuously differentiable activation function. Since \( v_{\text{xc}} \) is determined by the functional derivative of \( E_{\text{xc}} \), the smoothness of \( v_{\text{xc}} \) is only guaranteed by smooth activation functions. The second and higher-order derivatives are required during the optimization of the NN parameters with quasi-Newton methods [31]. In KSR, the sigmoid linear unit (SiLU or Swish) [32, 33] is used throughout and is infinitely differentiable. With no bias term, the linearity of SiLU about the origin makes \( \epsilon_{\text{xc}} \) vanish wherever the density does in real space. A final negative transformation layer added to \( \epsilon_{\text{xc},\theta}[n] \) applies a negative SiLU to change positive output values to negative, since it is expected that \( \epsilon_{\text{xc}}[n](r) \) is negative in most regions of space given the exact condition \( E_{\text{xc}}[n] \leq 0 \).

In Ref. 17 several neural XC functional models were proposed: a local functional which only depends on the density at each point (KSR-LDA), a semi-local functional that uses local and gradient information about each point (KSR-GGA), and a global functional which utilized the global convolution layer and the SIG described above (KSR-global).

KSR-LDA and KSR-GGA approximations to XC are devoid of global information and SIG. For KSR-LDA, a convolution layer with filter size one ensures that the density, \( n(x) \), is mapped to \( \epsilon_{\text{xc}} \) at the same spatial point \( x \). For KSR-GGA, a convolution layer with filter size three is used to yield \( \epsilon_{\text{xc}} \) as a function of \( n(x) \) at three neighboring points in 1D, effectively including gradient information. Other than the global convolution layer and SIG, the KSR-global, KSR-LDA, and KSR-GGA architectures are equivalent to each other (See Fig. S3 in Ref. 17).

The models proposed in Ref. 17 do not explicitly account for spins and can perform poorly for spin-polarized systems. In the following, we will generalize and extend these models and techniques to incorporate spin.

5. SPIN-ADAPTED KOHN-SHAM REGULARIZER

Spin is often incorporated in the neural XC functional using relative polarization, \( \zeta \), as a feature [12]. For up and down spin densities, \( \{ n_\uparrow, n_\downarrow \} \), \( \zeta = (n_\uparrow - n_\downarrow)/n \). While \( \zeta \) can be introduced as an additional input channel to KSR \( \epsilon_{\text{xc}} \), its scale can be very different relative to \( n \) in general. Instead, we use up and down spin densities as input features, which have similar scales. The usual models and concepts for KSR can be extended to obtain a spin-adapted KSR (sKSR).

In the case of sKSR-global, the global convolution layer is generalized using the spin densities, and the kernel takes the form:
\[ G(n_\sigma(x), \xi_p) = \frac{1}{2} \xi_p \int dx' n_\sigma(x') e^{-|x-x'|/\xi_p}, \tag{7} \]
where \( \sigma \in \{\uparrow, \downarrow\} \) and \( \xi_p \) is a trainable parameter that represents an interaction scale. We input each spin density to this global convolution layer consisting of 8 channels. We then concatenate the output on the channel dimension and input it to the latter convolution layers. We do not include the SIG since self-interaction error
is less severe in weakly-correlated systems. The rest of
the network architecture is kept unchanged. Analogous
changes are made to obtain sKSR-LDA and sKSR-GGA.
In the latter, we specify the total density gradient as an
additional input channel along with the spin densities.
Fig. 1(a) shows the comparative network structures for
all three types of approximations. In all cases, the
resulting $\epsilon_{XC}$ is symmetrized with respect to the input
of the up and down densities:
\[
\epsilon_{XC}^{\text{symm}}[n_\uparrow, n_\downarrow] = \frac{1}{2} \left[ \epsilon_{XC}[n_\uparrow, n_\downarrow] + \epsilon_{XC}[n_\downarrow, n_\uparrow] \right].
\] (8)

Our approximation replaces the $\epsilon_{XC}$ in a spin-polarized
self-consistent KS-DFT framework. Fig. 1(b) shows the
conventional computational flow and the flow of the
gradients during the self-consistent optimization. In
the initial KS iteration, we use initial spin-densities
(corresponding to those in KS potential $v_{KS,\sigma}(r) = v(r)$.
The XC energy densities are calculated for spin-densities
and the spin-polarized XC potentials ($v_{XC,\uparrow}$, $v_{XC,\downarrow}$) are
extracted from the integrated XC energy. JAX [18]
can be used in practice to obtain functional derivatives using
automatic differentiation,
\[
v_{XC,\sigma}(x) = \frac{\delta E_{XC}[n_\sigma]}{\delta n_\sigma(x)}(x) = \frac{\delta}{\delta x'} n(x') \epsilon_{XC}[n_\uparrow, n_\downarrow](x'),
\] (9)
where $\epsilon_{XC}$ is calculated from the density using one of
the three functional approximations shown in Fig. 1(a).
The resulting KS potential is then used to solve the KS
equation. We use the solutions to calculate the output spin densities and the total density, $n = n_\uparrow + n_\downarrow$. By summing KS kinetic energy, Hartree energy and XC
energy, we get the total electronic energy. Before the
next KS cycle, the spin-densities are updated through
linear spin-density mixing with an exponentially decaying
mixing factor $\alpha$ [17],
\[
n_{\sigma,k+1}^{in} = n_{\sigma,k}^{out} + \alpha (n_{\sigma,k}^{out} - n_{\sigma,k}^{in}).
\] (10)
We repeat this process until the integrated absolute
difference in the input and output densities becomes
negligibly small (of the order of $10^{-6}$). To train the
neural XC functional, we use the following loss function:
\[
L(\theta) = E_{\text{train}} \left[ (E_{\text{sKSR}} - E_{\text{DMRG}})^2/N_e \right] + E_{\text{train}} \left[ \int dx (n_{\text{sKSR}} - n_{\text{DMRG}})^2/N_e \right],
\] (11)
where $E_{\text{sKSR}}$ and $n_{\text{sKSR}}$ are the converged total
energy and total density obtained from the neural XC
approximations, and $E_{\text{DMRG}}$ and $n_{\text{DMRG}}$ are the exact
ground-state electronic energy and total density for each
of the test systems. The total loss is evaluated as an
expectation over training examples, where $N_e$ is the
number of electrons for a given training example. All
quantities are in atomic units. We only consider the
converged energy in the energy loss term rather than the
energy trajectory throughout KS iterations, which
was explored in Ref. 17. In this work we find that
the self-consistent calculations converge quickly for the
small atoms and ions used in training, and incorporating
energy loss from each KS iteration minimally affects the
efficiency of the optimization process. The gradients are
calculated based on the total loss with respect to the

---

**FIG. 1.** (a) sKSR-global, sKSR-LDA and sKSR-GGA architectures to calculate $\epsilon_{XC}$ from spin-densities. (b) sKSR – differentiable KS-DFT with spin-polarization. Black arrows refer to the conventional computational flow. The gradients flow along red-dashed arrows to minimize the loss during training.
parameters, $\theta$, through automatic differentiation. They are back-propagated across the self-consistent cycles and the parameters of the neural XC approximation are updated until the total loss is minimized.

6. RESULTS

6.1. sKSR-LDA reproduces LSDA results

![Graph showing the comparison between sKSR-LDA, LSDA, and XC potentials](image)

**TABLE I.** Training, validation and test sets for generalizability experiment. The molecules in the test set refer to the relaxed structures.

| Training  | Validation | Testing       |
|-----------|------------|---------------|
| H, He, Li | Be $^+$    | H$_2$, H$_3$, H$_4$, H$_4^+$, H$_5$ |
| Be, Be$^{++}$ | LiH, BeH$_2$, HeH$^+$ | H-He-He-H$^{++}$ |
|           |            | He-H-He-H$^{++}$ |

As a simple consistency test, we pose the question: can KSR learn human-designed functionals from their observable results? Here we specifically investigate whether sKSR-LDA can learn the relatively simple but general human-designed 1D LSDA XC functional. Since our sKSR-LDA model utilizes hundreds of parameters, it is unclear whether training on just a few LSDA generated DFT results will yield a neural XC model that matches LSDA. We find that by training sKSR-LDA on LSDA-generated He and Li$^{++}$, we recover the LSDA XC functional almost exactly for unpolarized and fully polarized systems, see Figure 2. The sKSR-LDA model deviates at the high-density limit (low $r_s$ limit) due to the limitation that our training densities only consist of $r_s > 0.5$.

6.2. Generalizability for weakly correlated molecules

In this section, we assess generalizability by training sKSR models using a few 1D atomic systems and testing on unseen 1D molecular systems. We trained all three models on DMRG energies and densities of H, He, Li, Be, and Be$^{++}$ and validated on Be$^{++}$. For training and validation details see Appendix. The trained model was later used to calculate the properties of several molecules in their equilibrium ground-state or relaxed form (see Table I). The errors in total energies, ionization, and atomization energies, as well as the average density losses for all three neural XC approximations, are reported in Table II. Compared to LSDA, the mean absolute error (MAE) in sKSR-LDA calculated energies is reduced by a factor of three. On the other hand, sKSR-global is an order of magnitude higher in accuracy and yields total energies with an MAE of 2.7 mH, not so far from the chemical accuracy limit of 1.6 mH. The cumulative MAEs for the training, validation, and test datasets are reported in the appendix.

The size of our dataset is practically limited by the chemical space provided by 1D and the associated exponential interaction. Even though we are dealing with a much smaller dataset, we trained the sKSR models on the ground-state energies and densities of 5 atomic systems only and did not include any molecules, contrary to results in Ref. 12 and Ref. 20 which train on derived quantities, such as atomization and ionization energies, and include molecules in training.

Using KSR-global, the predicted densities of each molecule have little noticeable error, see Fig. 3(a). The corresponding XC potentials are shown in Fig. 3(b). For all unpolarized systems, we run restricted KS calculations, and the up and down XC potentials match, while for polarized systems (Li, Be$^+$, H$_3^+$, and H$_3$ only) we run unrestricted KS calculations. The sKSR-LDA and sKSR-GGA total densities and XC potentials for the test set are reported in the appendix.

The comparison to exact XC potentials is not expected to be as precise as potentials are extremely sensitive to densities. However, for each of these examples, we see that the sKSR-global XC potential closely mimics the exact XC potential, even though we did not include XC potentials in the training. Furthermore, seemingly large deviations in the XC potentials can result in similar resulting densities. For example, this can be seen in the case of BeH$_2$ where the XC potentials are noticeably different but the resulting densities are very similar. The KS potentials are reasonably accurate for the test set (see appendix). Note that similar to the exact XC potentials, the sKSR-global XC potentials are smooth, due to the use of a smooth activation function.

We can use these potentials to validate the known theoretical properties of the exact XC potentials for different test systems, compare with other XC approximations, and utilize them to
6.2 Generalizability for weakly correlated molecules

**FIG. 3.** (a) The exact ground-state density (gray) and the densities obtained using sKSR-global (orange dashes), (b) average XC potentials calculated from sKSR-global approximation (red dashes) to $\epsilon_{xc}$ and their exact counterparts calculated with DMRG (light blue) for the test molecules in Table. I at equilibrium separations. The sKSR potentials are shifted by a constant for a better comparison with the exact XC potentials. sKSR-global was trained on H, He, Li, Be, and Be$^{++}$ and validated on Be$^+$. Note that, in general, these 1D densities and XC potentials can differ even qualitatively from their 3D analogs.

introduce corrections to existing local and semilocal approximations. Similarly, sKSR-global can also produce quite accurate spin-densities even though we did not incorporate spin-densities in the loss function while training the XC functionals (see Fig. A1 in Appendix).

Since our sKSR models are trained on atomic systems alone, we do not expect them to perform well for stretched molecules. However, in many cases, the predictability of sKSR can extend well beyond the equilibrium bond distance. Fig. 4 shows the complete dissociation energy curve of LiH. Near equilibrium, sKSR-LDA and sKSR-GGA underestimate the binding energy but perform better than LSDA. As the bond is stretched, sKSR-GGA and sKSR-LDA quickly deviate from the expected trajectory. However, sKSR-global performs well throughout, extending its predictive accuracy well beyond the equilibrium bond distance. We show the total density and the XC potential of stretched LiH at 5.92 Bohr in Fig. 5. LSDA largely overestimates the total energy of the stretched molecule, but its density remains reasonably accurate. The XC potentials are also comparable, with sKSR-global closely approximating the exact behavior. A comparison of the sKSR-global and the exact total density and XC potential of stretched LiH with respect to the atomic contributions from Li and Hydrogen can be found in the appendix.
TABLE II. Total energy errors (in mH), density losses (in $10^{-4}$ Bohr$^{-1}$), and errors in ionization potentials for atoms and atomization energies in molecules (in mH) calculated using uniform gas LSDA [28], sKSR-LDA, sKSR-GGA, and sKSR-global respectively, for the training, validation, and test sets in Table I.

| Dataset | Symbol | LSDA | sKSR-LDA | sKSR-GGA | sKSR-global |
|---------|--------|------|----------|----------|-------------|
|         |        | $\Delta E$ | $L_n$ | $\Delta IP$ | $\Delta E$ | $L_n$ | $\Delta IP$ | $\Delta E$ | $L_n$ | $\Delta IP$ | $\Delta E$ | $L_n$ | $\Delta IP$ |
| Training | H      | 26.6 | 5.35 | -26.6 | 4.51 | 0.55 | -4.50 | 4.49 | 0.31 | -4.49 | 0.85 | 0.33 | -0.85 |
|         | He     | 41.4 | 2.89 | -8.46 | 20.2 | 0.63 | -21.3 | 7.49 | 0.24 | -10.2 | -0.69 | 0.03 | 0.62 |
|         | Li     | 33.7 | 5.02 | 16.6 | -11.5 | 0.40 | 37.4 | -12.0 | 1.37 | 20.2 | -2.37 | 0.12 | 2.79 |
|         | Be     | 24.5 | 1.18 | 21.4 | -23.5 | 1.03 | 12.1 | -2.70 | 0.65 | -5.29 | 1.16 | 0.07 | -1.23 |
|         | Be$^{++}$ | 55.3 | 0.75 | -18.1 | 29.2 | 0.16 | -46.1 | 6.55 | 0.49 | -34.1 | 0.41 | 0.02 | -1.43 |
|         |        | MAE  | 36.3 | 3.04 | 18.3 | 17.8 | 0.56 | 24.3 | 6.65 | 0.16 | 14.8 | 1.10 | 0.12 | 1.38 |
| Validation | Be$^+$ | 46.0 | 1.95 | 9.37 | -11.3 | 0.12 | 40.5 | -7.99 | 0.61 | -10.2 | -0.69 | 0.03 | 0.62 |
|         |        | $\Delta AE$ | $\Delta AE$ | $\Delta AE$ | $\Delta AE$ | $\Delta AE$ |
| Test | H$_2$ | 34.04 | 1.82 | 19.2 | 19.5 | 0.35 | -10.5 | 6.83 | 1.99 | 2.14 | -0.73 | 0.07 | 2.43 |
|       | H$_3$ | 35.6 | 1.93 | 44.3 | 0.45 | 0.21 | 13.1 | -3.07 | 5.57 | 16.5 | -3.56 | 3.22 | 6.11 |
|       | H$_4$ | 32.3 | 3.82 | 74.3 | 7.66 | 1.59 | 10.4 | -9.34 | 4.18 | 27.3 | 2.87 | 1.46 | 0.53 |
|       | H$_3^+$ | 19.6 | 6.68 | 7.09 | 2.78 | 0.71 | 1.73 | 1.68 | 1.71 | 2.81 | -1.94 | 1.04 | 2.79 |
|       | H$_2^+$ | 31.2 | 0.78 | 22.1 | 20.6 | 1.87 | -11.6 | 15.4 | 11.5 | -6.44 | -0.40 | 0.47 | 2.09 |
|       | LiH   | 30.9 | 3.72 | 29.5 | -8.55 | 2.47 | 1.53 | -16.6 | 3.86 | 9.14 | -4.38 | 0.66 | 2.86 |
|       | BeH$_2$ | 32.8 | 7.49 | 45.0 | -27.8 | 5.5 | 13.4 | -34.6 | 3.09 | 40.9 | -5.07 | 1.29 | 7.93 |
|       | HeH$_2$ | 37.3 | 1.71 | 4.18 | 18.8 | 0.17 | 1.40 | 5.18 | 0.59 | 2.31 | -1.60 | 0.13 | 0.91 |
|       | H-He-He-H$_2^+$ | 36.7 | 14.7 | 46.1 | 5.00 | 6.00 | 35.5 | -9.04 | 2.50 | 24.0 | 5.39 | 4.52 | -6.77 |
|       | He-H-He-H$_2^+$ | 46.1 | 7.40 | 36.7 | 19.9 | 6.48 | 20.6 | 4.35 | 4.75 | 10.6 | 0.79 | 5.47 | -2.18 |
|       |        | MAE  | 33.6 | 5.00 | 32.9 | 13.1 | 2.53 | 12.0 | 10.6 | 3.98 | 14.2 | 2.67 | 1.83 | 3.46 |

FIG. 4. The complete dissociation energy curve of LiH molecule generated with sKSR-LDA (orange), sKSR-GGA (green) and sKSR-global(red). The DMRG (black dashes) and the uniform gas LSDA (blue dashes) results are also shown. The neural XC approximations were trained and validated on atoms and ions given in Table. I.

7. CONCLUSION

KSR achieves remarkable accuracy and generalization in a very data-efficient manner by including the self-consistent KS equations into the training. By extending to a spin-adapted KSR, we can now accurately treat both spin-unpolarized and spin-polarized systems.

We show that sKSR with global information (sKSR-global) can predict the ground-state energy of ten unseen molecules in equilibrium with a mean absolute error of 2.7 mH ($\sim$1.7 kcal/mol) when trained with just five atoms and ions. Hence, a nonlocal XC approximation trained with atoms within an end-to-end differentiable DFT code can generate predictions for molecules with near chemical total density calculated from the self-consistent solution of the KS equations works as an excellent approximation to the exact density for most systems. Hence, the density-driven error is often negligible. However, some approximations can have significant density-driven errors [35]. For our test molecules, the errors in the self-consistent densities were trivial and consequently had minimal impacts on the atomization energy errors. The functional and density-driven errors in our neural XC approximations are reported for the hydrogen molecule in the supplementary information section.
accuracy. Combining differentiable programming with inherent physical intuition thus takes us one step closer to a chemically accurate XC functional.

The application of the current KSR algorithm is limited to 1D systems and our test set does not include real 3D molecules. However, the methods presented are transferable to 3D and we anticipate that the characteristic performance is not unique to 1D systems, as these systems tend to mimic their 3D analogs [36]. The low-dimensional examples are useful for quick and rigorous assessment of the quality of an approximation. Besides, the predictions from the local and semilocal approximation explored in our study are consistent with the 3D differentiable formulations in Ref 20 and Ref. 21.

8. DATA AVAILABILITY STATEMENT

The training and testing data and the one-dimensional density functional theory solver used for the uniform electron-gas LDA calculations are available at https://github.com/pedersor/DFT_1d. The ML models and the JAX version of the DFT code are available from the corresponding author upon reasonable request.
Appendix A: Appendices

1. Calculation details

a. Data generation

All training data are generated from 1D DMRG calculations \cite{29} with exponential approximation. The extended Hubbard-like Hamiltonian \cite{28} for 1D systems is solved in real space on a grid of 513 points within the range \( x \in \{-20.48, ..., 20.48\} \) with a separation distance of 0.08 Bohr and center at \( x = 0 \). Calculations are done using the ITensor library \cite{37} with an energy convergence threshold of \( 10^{-7} \) Hartree.

Exact KS potentials and XC potentials were generated for DMRG-calculated spin densities using a modified version of the KS-inversion algorithm outlined in Ref. \cite{38}. The code used to perform KS-inversion is publicly available at \cite{39}.

b. 1D KS calculations

The 1D KS-DFT code is also implemented with the external potential given in Eq. 6. Same real space grids are considered for solving the KS equations (Eq. 1). No symmetry conditions are enforced in our calculations as our training and test set contain asymmetric examples.

The LSDA approximation is implemented in 1D with the uniform gas exchange energy for the exponential interaction given in Ref. \cite{28} and an accurate parameterized model for the correlation energy.

c. Training, validation and test

For the training set and validation set containing atoms and ions, we used a fixed number of iterations during the training process for all three XC functionals. Based on the convergence of standard 1D KS-DFT calculations with local density approximation for these systems, the number of KS iterations was fixed at 10.

We repeated the training process for sKSR-global, sKSR-LDA, and sKSR-GGA with 30 random seeds. The model was trained with L-BFGS algorithm \cite{31}. Parameters checkpoints were saved at the interval of 10 steps until L-BFGS was converged. The optimal checkpoint for each seed was determined as the checkpoint that predicted the total energy and density loss of the validation set with the lowest mean absolute error (MAE). Then the analysis was repeated for all seeds to determine the best set of parameters.

For the test system, the number of KS iterations required for convergence varies based on the complexity of the system. While running the 1D KS code with the parameterized neural XC for these systems, we fixed the number of KS iterations at 30, sufficient for the largest molecule in the test set.

d. Computational resources

We generalized the codes available in the open-sourced JAX-DFT library \cite{40} to build the spin-adapted Kohn-Sham regularizer. Training and testing can be accomplished on NVidia GPUs or conventional CPU nodes.

2. Optimizing NN architectures

The number of convolution layers for each one of the three KSR networks at different levels of approximation is set according to the proposed architectures in Ref. \cite{17}. All the results reported in this paper used 8 channels in the global convolution layer of sKSR-global as well as the first convolution layer in the sKSR-LDA and sKSR-GGA architectures. Increasing the number of channels from 8 to 16 for both up and down densities does not affect the final energy and density predictions, but increases the cost of the calculation. Adding the self-interaction gate to sKSR-global also does not improve generalization for weak correlation.

3. Experimental details

The MAE in energies for \( N \) examples in the test set was calculated as,

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |E_{\text{KSR}} - E_{\text{DMRG}}|,
\]

and for the density, the average density loss \( L_n \) was calculated from the test set density losses (see Eq. 11).

The cumulative MAEs in total energies and density losses for the training, validation, and test datasets with all the XC approximations examined in the main text are reported in Table A1. The MAEs in the ionization potentials for the 6 atomic systems in the training and validation sets and the MAEs in atomization energies for the molecules in the test set are also included.

Fig. A1 has the up and down spin-densities calculated with the sKSR-global approximation to the XC energy density for the ten test molecules. Fig. A2 shows the corresponding KS potentials for these molecules. Total densities and spin-up and spin-down XC potentials calculated with sKSR-LDA and sKSR-GGA are shown in Fig. A3 and Fig. A4, respectively.

We compare the atomic contributions of Li and H to the total density and the XC potentials of stretched LiH at 5.92 Bohr bond distance in Fig. A5. The exact densities of Li and H visibly add up to the stretched total density of LiH. The exact hydrogen XC potential is

9
shifted vertically by 0.27 Hartree to match the hydrogen peak in LiH. At the dissociation limit, this value should approach the ionization potential difference of H and Li (0.35 Hartree).

TABLE A1. MAE for total energies, ionization potentials (IP), and atomization energies (AE), and average density losses (∗10−4 Bohr−1) with each KSR XC approximations for all the atoms, ions and molecules in all datasets in Table I. All energies are in mH. For all KSR models, we used the same training and validation sets from Table. LSDA corresponds to the reference 1D uniform gas XC functional [28].

| Method       | ∆E  | L_n | IP   | AE      |
|--------------|-----|-----|------|---------|
| LSDA         | 35.3| 4.29| 16.8 | 32.9    |
| sKSR-LDA     | 14.5| 1.76| 27.0 | 11.9    |
| sKSR-GGA     | 9.21| 2.58| 14.8 | 14.2    |
| sKSR-global  | 2.02| 1.18| 1.24 | 3.46    |

For any approximate energy functional $\tilde{E}$ and approximate density $\tilde{n}$, the total error $\Delta E$ is,

$$\Delta E = \Delta E_D + \Delta E_F,$$

where $\Delta E_D$ is the density-driven error, and $\Delta E_F$ is the functional error. The total error is the sum of the two, $\Delta E = \tilde{E}[\tilde{n}] - E[exact][n[exact]]$. We calculated $\Delta E_D$ and $\Delta E_F$ for the atomization energies of equilibrium H₂ molecule with our three neural XC approximations and compared the values with uniform gas LSDA. These errors are reported in Table A2.

4. Density-driven errors

For any approximate energy functional $\tilde{E}$ and approximate density $\tilde{n}$, the total error $\Delta E$ is,

$$\Delta E = \tilde{E}[\tilde{n}] - E[exact][n[exact]] + \tilde{E}[n[exact]] - E[exact][n[exact]].$$

(A2)
Density-driven errors

FIG. A4. The exact ground-state density (gray) and the densities obtained using sKSR-GGA (orange dashes), (b) average XC potentials calculated from sKSR-GGA approximation (red dashes) and their exact counterparts (light blue) for the test molecules in Table I at equilibrium separations.

TABLE A2. The density driven errors and the functional driven errors in the atomization energy (in mH) for the H$_2$ molecule at equilibrium. For all KSR models, we use the same training set of 5 atomic systems: H, He, Li, Be, and Be$^{++}$, and validated on Be$^+$.  

| Method       | $\Delta AE_D$ | $\Delta AE_F$ | $\Delta AE$ | $|\Delta AE_D|$   |
|--------------|---------------|---------------|-------------|-------------------|
| LSDA         | -0.10         | 19.4          | 19.3        | 0.51              |
| sKSR-LDA     | -0.05         | -10.5         | -10.5       | 0.45              |
| sKSR-GGA     | -0.07         | 2.21          | 2.14        | 3.18              |
| sKSR-global  | -0.02         | 2.45          | 2.43        | 1.06              |

FIG. A5. (a) DMRG and the sKSR-global densities of stretched LiH (5.92 Bohr) and the atomic densities of Li (blue dashes) and hydrogen (green dashes). (b) The exact (black dashes) and the sKSR-global (red) average xc potentials of LiH at the same bond distance. The exact average xc potentials of Li (orange dashed) and H (green dashes) and the corresponding sKSR-global average XC potentials of Li (blue) and H (green) are included here for comparison.
1. P. Hohenberg and W. Kohn, Phys. Rev. 136, B864 (1964).
2. W. Kohn and L. J. Sham, Phys. Rev. 140, A1133 (1965).
3. R. M. Dreizler and E. K. U. Gross, Density Functional Theory: An Approach to the Quantum Many-Body Problem (Springer Berlin Heidelberg, Berlin, Heidelberg, 1990).
4. N. Mardirossian and M. Head-Gordon, Molecular Physics 115, 2315 (2017).
5. J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. 77, 3865 (1996).
6. Y. Zhao and D. G. Truhlar, The Journal of Chemical Physics 125, 194101 (2006), https://doi.org/10.1063/1.2370993.
7. K. Burke, J. Chem. Phys. 136, 150001 (2012).
8. J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, Phys. Rev. Lett. 108, 253002 (2012).
9. L. Li, J. C. Snyder, I. M. Pelaschier, J. Huang, U.-N. Niranjan, P. Ducan, M. Rupp, K.-R. Müller, and K. Burke, International Journal of Quantum Chemistry 116, 819 (2016), https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.25040.
10. F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, Nature Communications 8, 872 (2017).
11. B. Kalita, L. Li, R. J. McCarty, and K. Burke, Accounts of Chemical Research 54, 818 (2021).
12. R. Nagai, R. Akashi, and O. Sugino, npj Computational Materials 6, 43 (2020).
13. L. A. Curtiss, K. RagHAVachari, G. W. Trucks, and J. A. Pople, The Journal of Chemical Physics 94, 7221 (1991), https://doi.org/10.1063/1.460205.
14. M.-C. Kim, E. Sim, and K. Burke, Phys. Rev. Lett. 111, 073003 (2013).
15. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, J. Mach. Learn. Res. 18, 5595-5637 (2017).
16. M. Innes, A. Edelman, K. Fischer, C. Rackauckas, E. Saba, V. B. Shah, and W. Tebbutt, A differentiable programming system to bridge machine learning and scientific computing (2019), arXiv:1907.07587.
17. L. Li, S. Hoyer, R. Pederson, R. Sun, E. D. Cubuk, P. Riley, and K. Burke, Phys. Rev. Lett. 126, 036401 (2021).
18. J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: composable transformations of Python+NumPy programs (2018).
19. C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, P. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, Nature 585, 357 (2020).
20. M. F. Kasim and S. M. Vinko, Phys. Rev. Lett. 127, 126403 (2021).
21. S. Dick and M. Fernandez-Serra, Using differentiable programming to obtain an energy and density-optimized exchange-correlation functional (2021), arXiv:2106.04481.
22. J. Sun, A. Ruzsinszky, and J. P. Perdew, Phys. Rev. Lett. 115, 036402 (2015).
23. S. Dick and M. Fernandez-Serra, Nature Communications 11, 3509 (2020).
24. Y. Chen, L. Zhang, H. Wang, and W. E, Journal of Chemical Theory and Computation 17, 170 (2021).
25. J. Kirkpatrick, B. McMorrow, D. H. P. Turban, A. L. Gaunt, J. S. Spencer, A. G. D. G. Matthews, A. Obika, L. Thiry, M. Fortunato, D. Pfau, L. R. Castellanos, S. Petersen, A. W. R. Nelson, P. Kohli, P. Mori-Sánchez, D. Hassabis, and A. J. Cohen, Science 374, 1385 (2021), https://www.science.org/doi/pdf/10.1126/science.abj6511.
26. A. D. Becke, The Journal of Chemical Physics 98, 5648 (1993), https://doi.org/10.1063/1.464913.
27. J. P. Perdew, A. Ruzsinszky, J. Tao, V. N. Staroverov, G. E. Scuseria, and G. I. Csonka, The Journal of Chemical Physics 123, 062201 (2005), https://doi.org/10.1063/1.1904565.
28. T. E. Baker, E. M. Stoudenmire, L. O. Wagner, K. Burke, and S. R. White, Phys. Rev. B 91, 235141 (2015).
29. S. R. White, Phys. Rev. Lett. 69, 2863 (1992).
30. A. Pribram-Jones, D. A. Gross, and K. Burke, Annual Review of Physical Chemistry 66, 283 (2015), pMID: 25830374, https://doi.org/10.1146/annurev-physchem-040214-121420.
31. D. C. Liu and J. Nocedal, Mathematical Programming 45, 503 (1989).
32. S. Elfwing, E. Uchibe, and K. Doya, Neural Networks 107, 3 (2018), special issue on deep reinforcement learning.
33. P. Ramachandran, B. Zoph, and Q. V. Le, Searching for activation functions (2018).
34. S. Vuckovic, S. Song, J. Koziolowski, E. Sim, and K. Burke, Journal of Chemical Theory and Computation 15, 6636 (2019), pMID: 31682433.
35. Y. Kim, S. Song, E. Sim, and K. Burke, The Journal of Physical Chemistry Letters 10, 295 (2019).
36. L. O. Wagner, E. Stoudenmire, K. Burke, and S. R. White, Physical Chemistry Chemical Physics 14, 8581 (2012).
37. M. Fishman, S. R. White, and E. M. Stoudenmire, The ITensor Software Library for Tensor Network Calculations (2020), arXiv:2007.14822.
38. K. Finzel, P. W. Ayers, and P. Bultinck, THEORETICAL CHEMISTRY ACCOUNTS 137, 6 (2018).
39. https://github.com/pedersor/DFT_1d (2021).
40. L. Li, S. Hoyer, R. Pederson, R. Sun, E. D. Cubuk, P. Riley, and K. Burke, Kohn-sham equations as regularizer: Building prior knowledge into machine-learned physics (2020).