Title: Finding density functionals with machine-learning

Date: Aug 11, 2016 09:30 AM

URL: http://pirsa.org/16080014

Abstract: Density functional theory (DFT) is an extremely popular approach to electronic structure problems in both materials science and chemistry and many other fields. Over the past several years, often in collaboration with Klaus Mueller at TU Berlin, we have explored using machine-learning to find the density functionals that must be approximated in DFT calculations. I will summarize our results so far, and report on two new works.

Finding density functionals with ML

Kieron Burke and friends
UC Irvine Physics & Chemistry

http://dft.uci.edu

Outline

- Review of density functional theory (DFT)
- ML for finding functionals for box problems
- ML for bond breaking
- Latest results in 3D and for E_{XC}
- Summary

The electronic structure problem

- Use atomic units
- Born-Oppenheimer approximation
- All non-relativistic (but can be added back in)
- Wavefunctions antisymmetric and normalized
- Only discuss ground-state electronic problem here, but many variations.

Hamiltonian for N electrons in the presence of external potential $v(\mathbf{r})$:

$$\hat{H} = \hat{T} + \hat{V}_{\rm ee} + \hat{V},$$

where the kinetic and elec-elec repulsion energies are

$$\hat{T} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2, \qquad \hat{V}_{\rm ee} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|},$$

and difference between systems is N and the one-body potential

$$\hat{V} = \sum_{i=1}^{N} v(\mathbf{r}_i)$$

Often $v(\mathbf{r})$ is electron-nucleus attraction

$$v(\mathbf{r}) = -\sum_{\alpha}\frac{Z_\alpha}{|\mathbf{r} - \mathbf{R}_\alpha|}$$

where α runs over all nuclei, plus weak applied $\mathbf{E}$ and $\mathbf{B}$ fields.

$$\{\hat{T} + \hat{V}_{\rm ee} + \hat{V}\}\,\Psi = E\,\Psi, \qquad E = \min_{\Psi}\langle\Psi|\hat{T} + \hat{V}_{\rm ee} + \hat{V}|\Psi\rangle$$

HK theorem (1964)

- Makes TF an approximation to an exact theory
- Can find both ground-state density and energy via Euler equation

1. Rewrite variational principle (Levy 79):

$$E = \min_{\Psi} \langle \Psi | \hat{T} + \hat{V}_{ee} + \hat{V} | \Psi \rangle$$
$$= \min_{n} \left\{ F[n] + \int d^3r \; v(\mathbf{r}) n(\mathbf{r}) \right\}$$

where

$$F[n] = \min_{\Psi \to n} \langle \Psi | \hat{T} + \hat{V}_{ee} | \Psi \rangle$$

   - The minimum is taken over all positive $n(\mathbf{r})$ such that $\int d^3r \; n(\mathbf{r}) = N$

2. The external potential $v(\mathbf{r})$ and the hamiltonian $\hat{H}$ are determined to within an additive constant by $n(\mathbf{r})$

- P. Hohenberg and W. Kohn, Phys. Rev. **136**, B 864 (1964).
- M. Levy, Proc. Natl. Acad. Sci. (U.S.A.) **76**, 6062 (1979).

KS DFT method (1965)

1964: HK theorem: There exists F[n]

Define *fictitious* non-interacting electrons satisfying:

$$\left\{-\frac{1}{2}\nabla^2 + v_{\rm S}(\mathbf{r})\right\}\phi_j(\mathbf{r}) = \epsilon_j\phi_j(\mathbf{r}), \qquad \sum_{j=1}^{N}|\phi_j(\mathbf{r})|^2 = n(\mathbf{r}).$$

where $v_{\rm S}(\mathbf{r})$ is *defined* to yield $n(\mathbf{r})$.

Define $T_{\rm S}$ as the kinetic energy of the KS electrons, U as their Hartree energy and

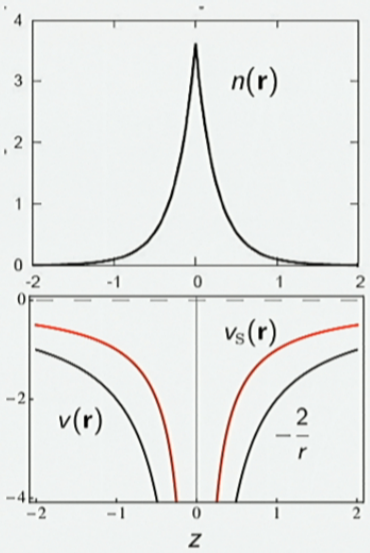
$$\text{F= } T + V_{\rm ee} = T_{\rm S} + U + E_{\rm XC}$$

the remainder is the exchange-correlation energy.

Most important result of exact DFT:

$$v_{\rm S}(\mathbf{r}) = v(\mathbf{r}) + \int d^3r \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} + v_{\rm XC}[n](\mathbf{r}), \qquad v_{\rm XC}(\mathbf{r}) = \frac{\delta E_{\rm XC}}{\delta n(\mathbf{r})}$$

Knowing $E_{\rm XC}[n]$ gives closed set of self-consistent equations.



Orbital-free DFT: Approximate $T_s[n]$ directly, and go much, much faster.

Today's commonly-used functionals

- Local density approximation (LDA)
  - Uses only n(**r**) at a point.

$$E_{\rm X}^{\rm LDA}[n] = A_{\rm X} \int d^3r \; n^{4/3}(\mathbf{r})$$

$$A_{\rm X} = -(3/4)(3/\pi)^{1/3} = -0.738.$$

- Generalized gradient approx (GGA)
  - Uses both n(**r**) and |∇n(**r**)|
  - Should be more accurate, corrects overbinding of LDA
  - Examples are PBE and BLYP

$$E_{\rm XC}^{\rm GGA} = \int d^3r \, e_{\rm XC}^{\rm GGA}(n(\mathbf{r}), |\nabla n(\mathbf{r})|)$$

- Hybrid:
  - Mixes some fraction of HF with GGA
  - Examples are B3LYP and PBE0

$$E_{\rm XC}^{\rm hyb} = a\,(E_{\rm X} - E_{\rm X}^{\rm GGA}) + E_{\rm XC}^{\rm GGA}$$

Importance of DFT

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..
- Gigo principle: Results only as good as $E_{XC}[n]$ approximation used.

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..
- Gigo principle: Results only as good as $E_{XC}[n]$ approximation used.
- Much faster than wavefunction/green's function methods for large systems.

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..
- Gigo principle: Results only as good as $E_{XC}[n]$ approximation used.
- Much faster than wavefunction/green's function methods for large systems.
- Never fast enough for e.g., biomolecular MD simulations.

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..
- Gigo principle: Results only as good as $E_{XC}[n]$ approximation used.
- Much faster than wavefunction/green's function methods for large systems.
- Never fast enough for e.g., biomolecular MD simulations.
- Often inaccurate for strongly correlated systems, including many energy-related materials.

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..
- Gigo principle: Results only as good as $E_{XC}[n]$ approximation used.
- Much faster than wavefunction/green's function methods for large systems.
- Never fast enough for e.g., biomolecular MD simulations.
- Often inaccurate for strongly correlated systems, including many energy-related materials.
- Catalysts found by DFT being patented and used.

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..
- Gigo principle: Results only as good as $E_{XC}[n]$ approximation used.
- Much faster than wavefunction/green's function methods for large systems.
- Never fast enough for e.g., biomolecular MD simulations.
- Often inaccurate for strongly correlated systems, including many energy-related materials.
- Catalysts found by DFT being patented and used.
- High pressure 200K superconductors predicted from DFT calculations.

Importance of DFT

- Spans many fields: chemistry, materials science, condensed-matter physics,..
- Gigo principle: Results only as good as $E_{XC}[n]$ approximation used.
- Much faster than wavefunction/green's function methods for large systems.
- Never fast enough for e.g., biomolecular MD simulations.
- Often inaccurate for strongly correlated systems, including many energy-related materials.
- Catalysts found by DFT being patented and used.
- High pressure 200K superconductors predicted from DFT calculations.
- Last year, 30,000 scientific papers published using DFT.

DFT: A Theory Full of Holes, Aurora Pribram-Jones, David A. Gross, Kieron Burke, Annual Review of Physical Chemistry (2014).

Axel Becke: 2016 Killam Prize in Natural Sciences

Systematic approach to DFT

- Lieb and Simon proved TF is leading order in an unusual semiclassical limit.
- For atoms, this is same as keeping neutral and taking N to infinity.
- With collaborators, I have shown LDA is leading term in same limit.
- In model cases, can find leading corrections, which are uniform asymptotic expansions in hbar and far more accurate than present-day DFT approximations.

Orbital-free DFT

- Long-time dream of electronic structure
- If you know $T_s[n]$ sufficiently accurately, you avoid computational cost of solving KS equations.
- Like TF, only accurate enough for prediction.
- Go from hundreds to millions of atoms
- See work by Emily Carter and Sam Trickey
- Also, extremely important in plasma physics simulations at million K scale

Strong correlation: Where DFT fails

- Ongoing project with Steve White at UCI
- Apply DMRG to continuum problems
- Understand limitations and failures of standard DFT approximations

B. Machine learning- demo

ML applications in electronic structure

- Most with Klaus Mueller of TU Berlin, computer science.


Klaus-Robert Müller

ML applications in electronic structure

- Most with Klaus Mueller of TU Berlin, computer science.


Klaus-Robert Müller

- ML now being applied directly to, e.g., molecular energies from geometries for drug design, many by Matthias Rupp (FHI Berlin).


Matthias Rupp

ML applications in electronic structure

- Most with Klaus Mueller of TU Berlin, computer science.


Klaus-Robert Müller

- ML now being applied directly to, e.g., molecular energies from geometries for drug design, many by Matthias Rupp (FHI Berlin).


Matthias Rupp

- Our efforts are focused on finding $T_s[n]$ from examples, work by John Snyder (Humboldt fellow at TU Berlin/MPI Halle)


John Snyder

Demo problem in DFT

- *N* non-interacting same-spin fermions confined to 1d box
- Define class of potential:

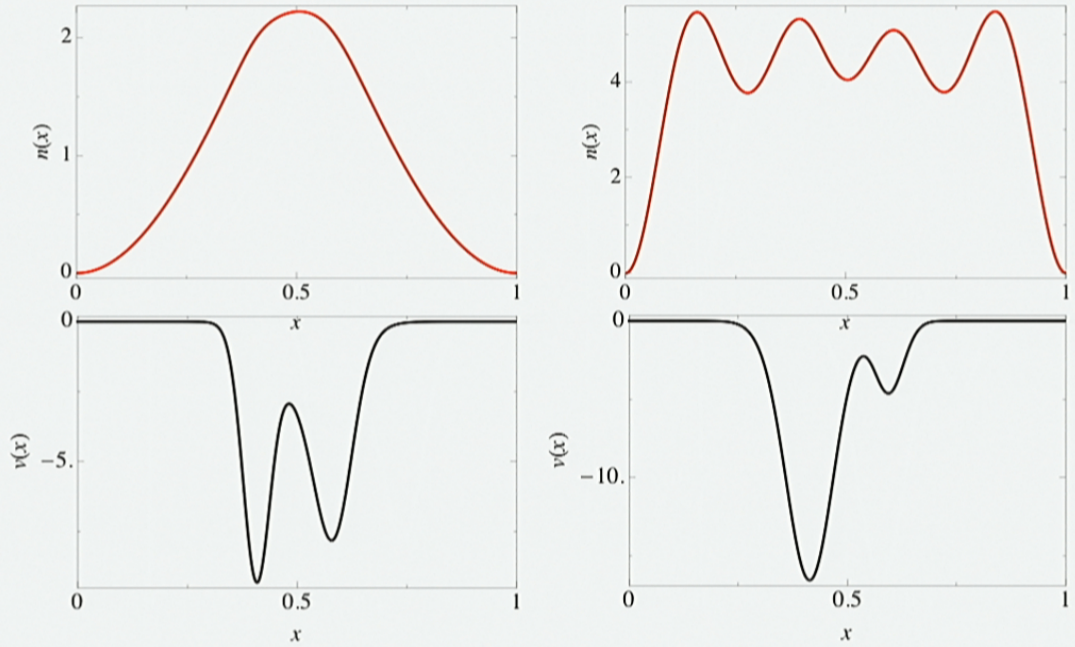
$$v(x) = -\sum_{i=1}^{3} a_i \exp(-(x-b_i)^2/(2c_i^2))$$

- Represent the density on a grid with spacing $\Delta x = 1/(G-1)$
- ML-DFA for KE:

$$\hat{T}(\mathbf{n}) = \bar{T} \sum_{j=1}^{M} \alpha_j k(\mathbf{n}_j, \mathbf{n})$$

Dataset

Generate 2000 potentials. Solve for up to 4 electrons.



Performance for T_s

kcal/mol

N	M	λ	σ	$\overline{\lvert\Delta T\rvert}$	$\lvert\Delta T\rvert^{\mathrm{std}}$	$\lvert\Delta T\rvert^{\mathrm{max}}$
1	40	2.4×10^{-5}	238	3.3	3.0	23.
	60	1.0×10^{-5}	95	1.2	1.2	10.
	80	6.7×10^{-6}	48	0.43	0.54	7.1
	100	3.4×10^{-7}	43	0.15	0.24	3.2
	150	2.5×10^{-7}	33	0.060	0.10	1.3
	200	1.7×10^{-7}	28	0.031	0.053	0.65
2	100	1.3×10^{-7}	52	0.13	0.20	1.8
3	100	2.0×10^{-7}	74	0.12	0.18	1.8
4	100	1.4×10^{-7}	73	0.078	0.14	2.3
1-4†	400	1.8×10^{-7}	47	0.12	0.20	3.6

LDA ~ 223 kcal/mol, Gradient correction ~ 159 kcal/mol

functional derivative?

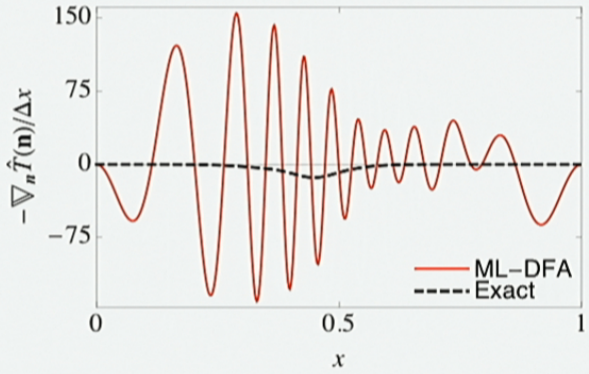
Exact

$$\frac{\delta T[n]}{\delta n(x)} = \mu - v(x)$$

ML-DFA

$$\frac{1}{\Delta x}\nabla_{\mathbf{n}}\hat{T}(\mathbf{n}) = \sum_{j=1}^{M}\alpha'_j(\mathbf{n}_j - \mathbf{n})k(\mathbf{n}_j, \mathbf{n})$$

$$\alpha'_j = \alpha_j/(\sigma^2\Delta x)$$



- Functionals are defined on infinite-dimensional spaces
- With finite interpolation, can always find bad directions
- Can we make a cruder definition that will work for our purposes?

Principal component analysis

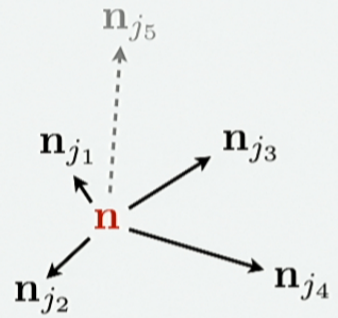
$$X = (\mathbf{n}_{j_1} - \mathbf{n}, \ldots, \mathbf{n}_{j_m} - \mathbf{n})^\top$$

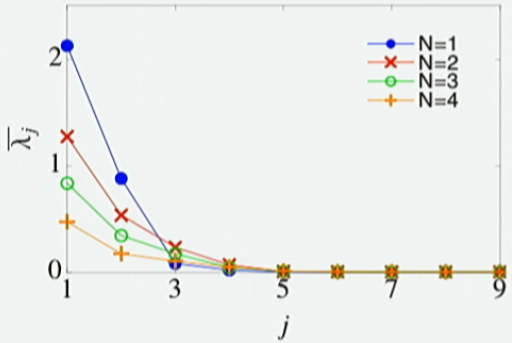
$$C = \frac{1}{m} X^\top X$$

$$\lambda_j, \mathbf{x}_j$$

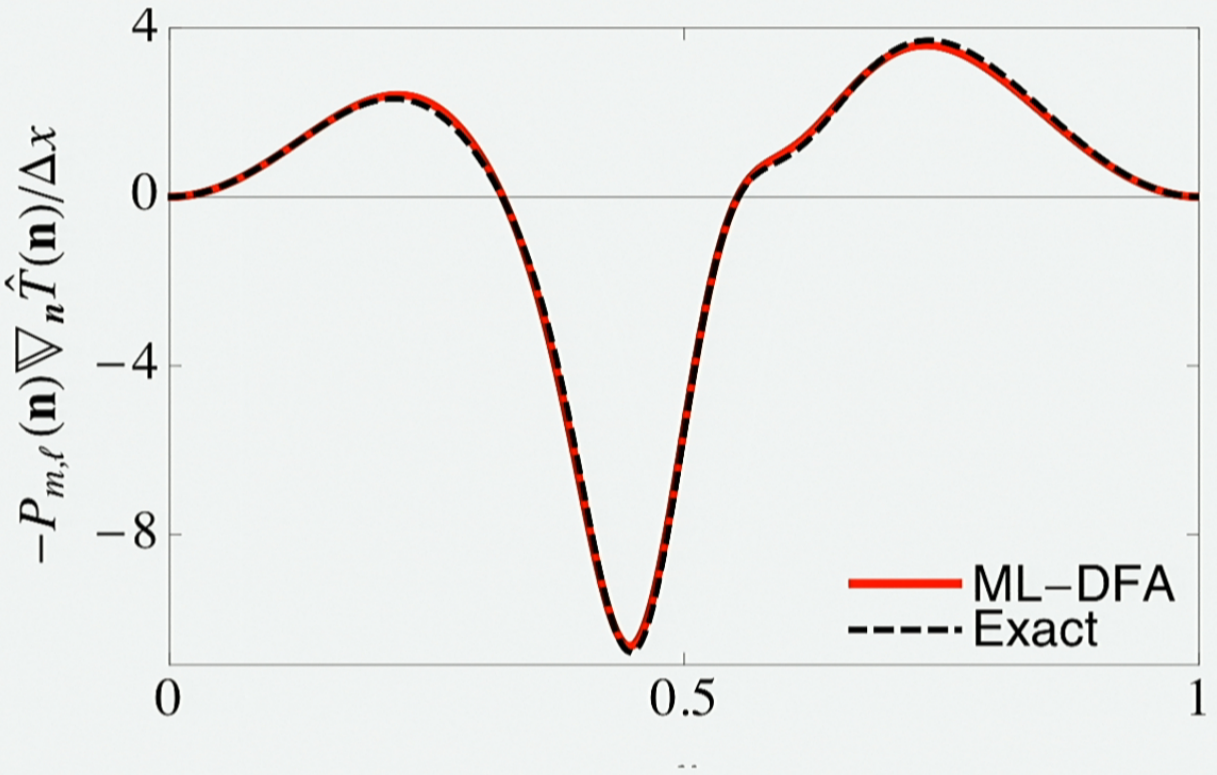
$$P_{m,\ell}(\mathbf{n}) = V^\top V$$

$$V = (\mathbf{x}_1, \ldots, \mathbf{x}_\ell)^\top$$





Projected functional derivative

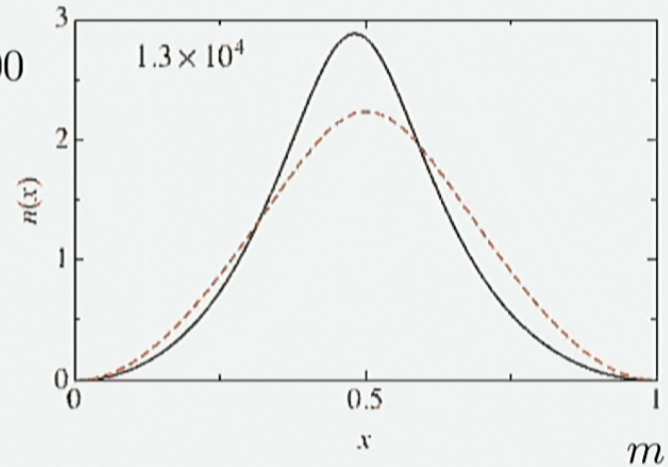


Constrained optimized density

- Gradient descent search:

$$\mathbf{n}^{(j+1)} = \mathbf{n}^{(j)} - \epsilon P_{m,\ell}(\mathbf{n}^{(j)})(\mathbf{v} + \nabla_{\mathbf{n}}\hat{T}(\mathbf{n}^{(j)})/\Delta x)$$

$N = 1$
$M = 100$



$\Delta T_j = 0.154$ kcal/mol
$\Delta T_j^{\mathrm{sc}} = 6.53$ kcal/mol

Ratio $= 43$

$m = 15, \ell = 5$

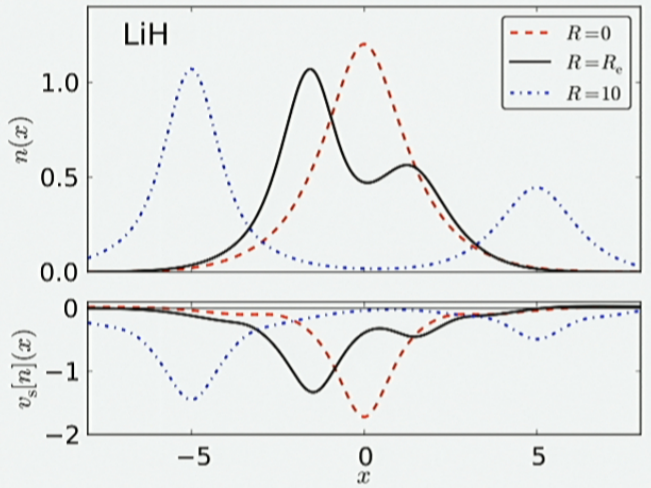
First ‘ML makes a functional’ paper

26	−22.389731490641	3.751807395744921	0.08526241073357	0.4453450848668749	7.559337237177321	0.05840909867970421	0.4990223467101435	4.729884445483556	0.09292187123067	0.5590503374710945
27	5.595799173773367	4.522456026868127	0.03620201946841309	0.5050210060676815	1.885091277540955	0.06521780385848891	0.4961601035955842	9.474219454963	0.05567401333961	0.4834045749740091
28	1.9164558566549	2.751346713386582	0.09481955285168	0.4901204212506473	7.863592227266858	0.09517979839518	0.5196620049802691	9.51608680035275	0.07785377315463	0.5806015592262945
29	−1.105559822533069	5.725277193011525	0.03097303116261406	0.5511177785567803	7.988513342309234	0.09283231789688	0.5146415116367569	1.101098863639624	0.08829390285197	0.5265108634130272
30	−3.886381532577754	4.95871172966177	0.06832354438777603	0.5645951313008806	9.081352253176	0.04472856905840844	0.4299117124064201	8.249736378493	0.0885443136185	0.5416193648861009
31	−1.341930696378109	6.451869235304859	0.05482804889270922	0.5694170176403242	4.416924333519995	0.04212208612720639	0.5897125000677342	8.86521100790565	0.0866581794842557	0.5311744230227635
32	−10.16467121571373	6.705508349541454	0.0934749698077619	0.4420693171728495	2.858371131410651	0.03722557506068529	0.4216580500297413	6.670274811535954	0.0977332470946321	0.5387560851291 85
33	−1.107633246744663	7.390220519784453	0.07970661674181083	0.5081940602224835	3.815123656821275	0.04965529863397879	0.497023655525965	9.699452453117 18	0.07039341131023045	0.4990744291774679
34	10.39451871866851	1.478170300165134	0.03602109509210016	0.5799113381048154	2.105782960173057	0.0861928382367878	0.4812528298377607	7.010226383015251	0.03535327789989237	0.4969989664577149
35	−0.139925361351959	4.045190240200832	0.03579955552582782	0.5678362348598895	9.32386837735074	0.0858648976559044	0.5698752996855582	9.818959037995 08	0.04704020956533079	0.5708692337579542
36	3.656124569605117	4.944179742822627	0.09319165253330 49	0.4834640277977335	9.38479794194 91	0.0840810063497973	0.4090254720704045	3.673880603434117	0.07158934023489803	0.5598196803643543
37	8.06734679112684	2.988178201634472	0.07482186947515447	0.4950144569660614	1.910770315838324	0.0387715290304512 2	0.5786659568499292	4.285706479098184	0.05940487550770145	0.4844513368813702
38	0.30115666218357 22	8.789962073643 55	0.04833452393773747	0.4592039373603 75	9.934677577777873	0.0973964303071365	0.4280349684145 508	4.931948094146255	0.09694102877556 26	0.4560660353529726
39	−0.2169178958324283	6.242957525364009	0.0989568314234112	0.4205662570527381	4.739832300049 83	0.0450110225161678	0.559400017887981	9.119789479431 94	0.06189826031162779	0.4931008194 19519
40	−5.704905642148452	4.447521595114223	0.06770523205457797	0.5047965897298675	3.987888630073714	0.0359513661525433 1	0.5929011044925536	5.221218237153 44	0.04029431905167698	0.5141294116814 18
41	0.1425795779687519	6.919606258393273	0.08217563689753 12	0.5976799418614426	8.040990370181 68	0.0312926211088143 8	0.4686530881925831	3.421695873681 74	0.05166215779587051	0.4208560803818513
42	0.6843702894502789	8.272005873231 67	0.0547013388431661	0.4453254786423237	7.433883810628654	0.0461220234382699 8	0.5269730370279639	8.364161952182 4	0.07996879390804805	0.5036755718396144
43	−14.065717573691 53	6.220712308892191	0.04765956545437692	0.4898959293071353	8.722494312412 57	0.0505996770523558 8	0.4412174052875277	2.416276423317733	0.07373660930828208	0.5403850652535799
44	8.323461094437 68	9.297160676629 43	0.06568106034148 27	0.4960066859011466	6.796263459296078	0.0403711610709775 5	0.4389885737869158	2.206183147865627	0.08049109890810 56	0.5806221323132373
45	−5.252317079780442	3.421683176707511	0.08675299855756 43	0.4322548560407479	6.859255296971943	0.0955962752949951	0.5176203999311 94	5.826144942819697	0.0910408442631135	0.4167014408128501
46	6.186413888987 92	3.789578783769164	0.07617326304382499	0.5418490045761073	1.363177133649232	0.0319724924524031 9	0.5284999992906743	4.98790019555 38	0.04126125322920861	0.4895167206754735
47	0.4821906326173532	1.195769022526768	0.04928749082970199	0.5286859472378347	8.767140630336 32	0.0870852412295278	0.5989638152903285	2.489565114307663	0.05568685410478115	0.5787056957942209
48	5.707663438868804	4.058614230578957	0.04743875933294599	0.4981149890354436	1.015171237842736	0.085826684616538	0.547034932077 4566	6.343764732282688	0.08787694361858 23	0.5449452367497044
49	−0.185795867902261 2	5.159763617905121	0.08505704225876 59	0.5311491284165588	7.238990148015965	0.0586704871139365 2	0.5662410781222187	8.706006093342 84	0.04088387548046445	0.5305143392409284
50	2.321594443893878	3.319181325761367	0.05391080867981007	0.5914435434600773	9.11776045377491	0.0358335 4001918901	0.4669413688494417	6.117669336479262	0.04553014156422459	0.4133132426352717 5
51	2.451069994722774	4.005247740100538	0.07620861067494149	0.5759225433332955	6.743529978394792	0.0967347280360008	0.4980631985926895	5.505849306951131	0.09714403635868 83	0.451734677211 02
52	−0.961350897030818	7.710615861039985	0.07131617125307138	0.4007362522722102	3.372061685311945	0.0457933543696741 7	0.4845503268496217	8.602199592923 5	0.06346181510833 68	0.4207242810824543
53	−10.67971490671366	4.728468083127391	0.06509200806253813	0.4959400801351127	9.812664816641 75	0.0951346706530719	0.5047212015277842	3.180268423645401	0.06891913385037023	0.5758580789792047
54	1.375998039323297	2.740626958875014	0.06438323836566841	0.5077158168564866	1.813952930718781	0.0845523320869031	0.407247659218711	1.054506345314588	0.09982747891443 16	0.5956344420441539
55	−4.330863793299111	3.209443235555092	0.05150806767094947	0.5932432195310449	4.682408324655706	0.0364137622661213 4	0.4491940043173043	9.578004704227 46	0.0897042234654 67	0.4702887735348434
56	6.961655278126058	3.013659237176324	0.08026842118833 49	0.5733460791404235	7.441047648005995	0.0924462539370359	0.5216671466381739	8.154031319345 62	0.05611934350568269	0.4910483719005932
57	−5.305400832815 13	2.088882156026381	0.07464091227469428	0.4740072756248339	4.228694369663863	0.0399317834897460 1	0.4402946238148492	4.912298433960689	0.06528359492264517	0.4923580906843156
58	3.395104400983763	7.9733680986292	0.09118435835952 36	0.5314007871406556	6.375772536276909	0.0933026364108379	0.4472442156555568	7.387561983870292	0.09776433844549 12	0.4798245902933 56
59	1.330224789015745	5.105339790874075	0.06331718284342962	0.5503804706506275	9.57408216418 42	0.0746350064413284 6	0.5814481952546582	5.810342341443878	0.03327635127803191	0.5550216356360829
60	−0.29206049552630 79	6.773414983805557	0.07595819905259191	0.5210834626114553	4.229974804881476	0.0353217754533547	0.5725615302150 63	8.884528232549 66	0.09329857106854 46	0.5046388124857028
61	−26.418409905255 62	6.417011618542 83	0.06246944646172664	0.4258923441426071	1.379491105318868	0.0630636792781810 8	0.4531443030507202	2.189456974183496	0.09807805933584	0.5174384564079299
62	6.630393088825135	6.476325542442256	0.07599435647188182	0.4345528705798566	9.969363897694016	0.0666930203717610 8	0.5174278943985 95	3.961825639220393	0.09291153222449 88	0.4621043570336795
63	3.065748817227897	1.300093759156537	0.06741885075599173	0.4866345116967513	6.942052331590217	0.0710796392075602 3	0.4930672781311588	7.834870626014938	0.06461156038706 24	0.5367219218216266
64	−1.743105697039855	3.145607697855832	0.04152929						0.07636214080338054	0.4330291254331895
65	−1.631255302904965	3.121291975208031	0.03133						0.09361924811385 18	0.4523240987335903
66	−9.509826926425 15	7.367027947839741	0.07398108						0.04986976063479617	0.5394709698394433
67	−4.496736421068983	8.564986505866069	0.08709503						0.05651923264880363	0.4832351547748815
68	−1.002977917543076	9.431218143626 08	0.0368989						0.05983735955021298	0.4837560035001434
69	−9.231065451706 94	8.479296961594 59	0.0547214						0.06892772167363432	0.5908529311844791
70	14.99926304831282	4.708917711658332	0.0907364						0.05825558914206748	0.5664341996427 86
71	4.189181104505013	7.583635162152007	0.09368295						0.04763840334231468	0.5010633160850224
72	8.328371881388 42	3.414315967033806	0.05054048464274665	0.5043307410524706	3.070996567281799	0.0696827632897450 9	0.5702669702356818	5.360783854112533	0.03888888383066967	0.5784155467566237
73	−21.84200888491582	1.625552430388586	0.04546236831654188	0.5426078576324179	4.091176923293506	0.0383688503965284 2	0.5048795129490 07	4.527748284314512	0.05975894102927981	0.4670598506496871
74	6.604738977945651	3.421878979312 62	0.08944995069718 7	0.4941485628974172	4.96749741331263	0.0603115112378358 2	0.4119028637456998	4.581927607849956	0.08719496369227 9	0.4176805754532423
75	15.27988423246628	6.229538426957454	0.08215210912158 63	0.4118748449252315	2.377028497509498	0.0963110933106458	0.4746736556808109	5.398545541594 87	0.08225746567806 43	0.5258947233190487

Finding Density Functionals with Machine Learning John C. Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, Kieron Burke, Phys. Rev. Lett. **108**, 253002 (2012)

Bond-breaking with ML

- Standard $T_s[n]$ approximations like TF go bad when bonds break.
- Performed many 1d KS calculations of diatomics as function of bond length, using LDA with soft-Coulomb repulsion, including several with more than 2 electrons



Orbital-free Bond Breaking via Machine Learning John C. Snyder, Matthias Rupp, Katja Hansen, Leo Blooston, Klaus-Robert Müller, Kieron Burke, J. Chem. Phys. **139**, 224104 (2013)

Constrained optimal density

- Convergence of constrained optimal density with # of training points.

Kernels, Pre-Images and Optimization John Snyder, Sebastian Mika, Kieron Burke, Klaus-Robert Müller, Chapter in Empirical Inference - Festschrift in Honor of Vladimir N. Vapnik (2013)

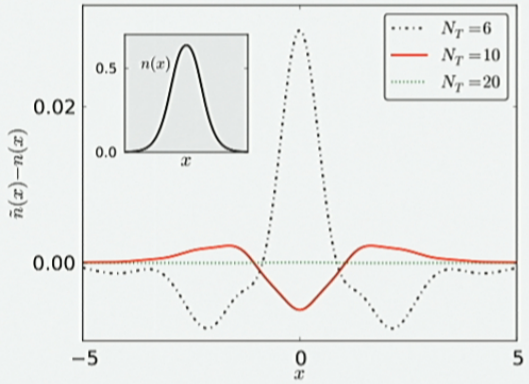
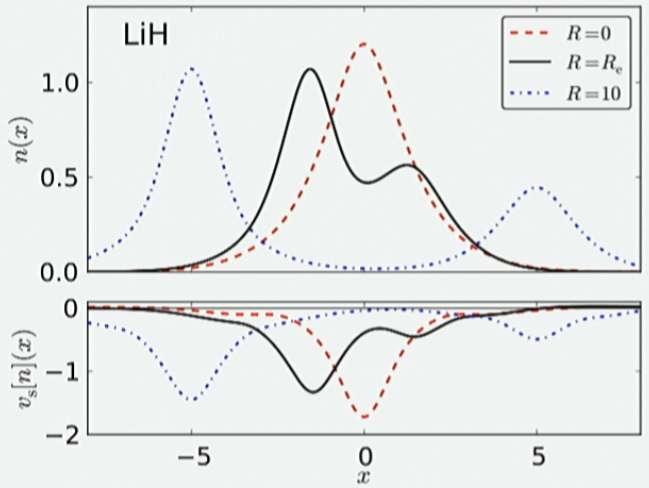


FIG. 7. Difference between the constrained optimal density $\tilde{n}(x)$ and the KS density $n(x)$ for various numbers of training densities N_T. The error decreases uniformly for all x. The system is H_2 at equilibrium bond length. The inset shows the KS density.

Bond-breaking with ML

- Standard $T_s[n]$ approximations like TF go bad when bonds break.
- Performed many 1d KS calculations of diatomics as function of bond length, using LDA with soft-Coulomb repulsion, including several with more than 2 electrons



Orbital-free Bond Breaking via Machine Learning John C. Snyder, Matthias Rupp, Katja Hansen, Leo Blooston, Klaus-Robert Müller, Kieron Burke, J. Chem. Phys. **139**, 224104 (2013)

Constrained optimal density

- Convergence of constrained optimal density with # of training points.

Kernels, Pre-Images and Optimization John Snyder, Sebastian Mika, Kieron Burke, Klaus-Robert Müller, Chapter in Empirical Inference - Festschrift in Honor of Vladimir N. Vapnik (2013)

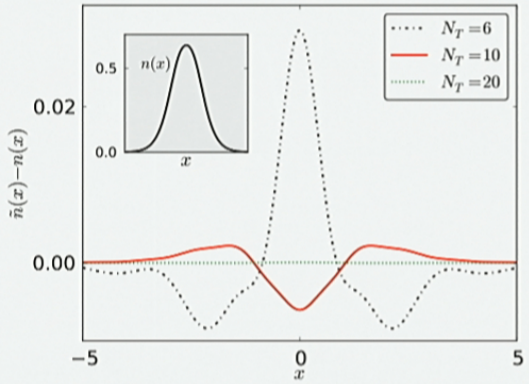


FIG. 7. Difference between the constrained optimal density $\tilde{n}(x)$ and the KS density $n(x)$ for various numbers of training densities N_T. The error decreases uniformly for all x. The system is H_2 at equilibrium bond length. The inset shows the KS density.

Types of errors in DFT

- $\Delta E_F = \bar{E}_{xc}[n] - E_{xc}[n]$
- $\Delta E_D = \bar{E}_{xc}[\tilde{n}] - \bar{E}_{xc}[n]$
- $\Delta E = \Delta E_F + \Delta E_D$
- Error analysis of energies in kcal/mol as a function of R with different numbers of training data, on constrained optimal densities

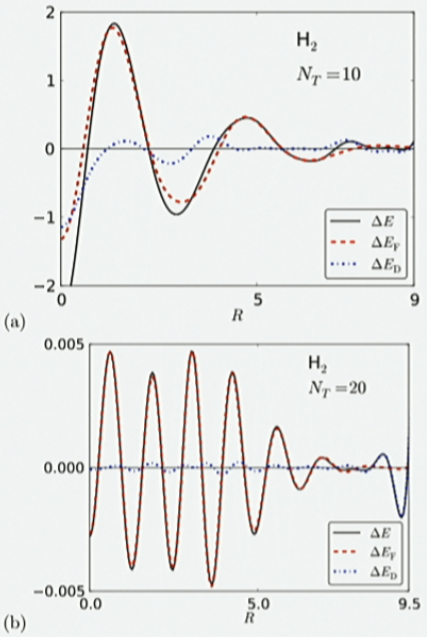


FIG. 8. The total error of the model and the functional- and density-driven errors ΔE_F and ΔE_D for H_2 with (a) 10 and (b) 20 training densities.

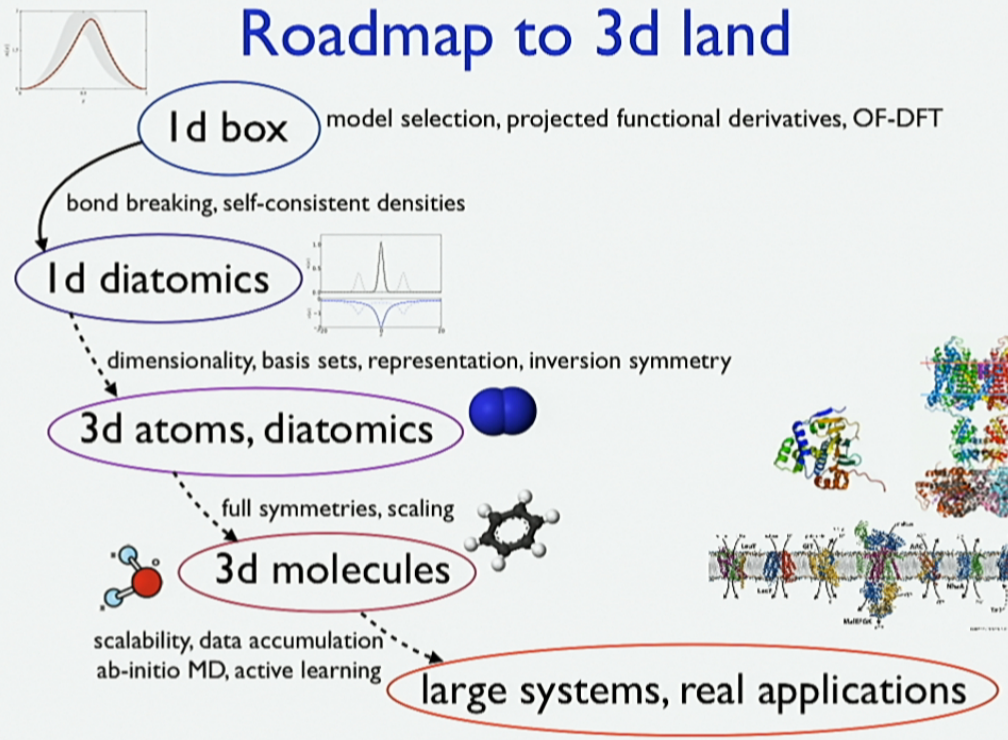
Functional derivatives and densities

- How can we get accurate densities from lousy derivatives?
- Once solution density is within interpolation manifold, simply constrain derivative to stay on that manifold
- Analogy:
  - Problem: find global minimum of 2D surface, given exact data along a 1D curve in that surface that passes through the minimum.
  - Solution: Make sure you stay on the path.
- PS: Inspired density-corrected DFT, which corrects many self-interaction errors!

Understanding and reducing errors in density functional calculations Min-Cheol Kim, Eunji Sim, Kieron Burke, Phys. Rev. Lett. **111**, 073003 (2013).

Ions in solution: Density corrected density functional theory (DC-DFT) Min-Cheol Kim, Eunji Sim, Kieron Burke, The Journal of Chemical Physics **140**, 18A528 (2014)

Road map back to reality



D. Recent results

- Either submitted or about to be.


Felix Brockherde

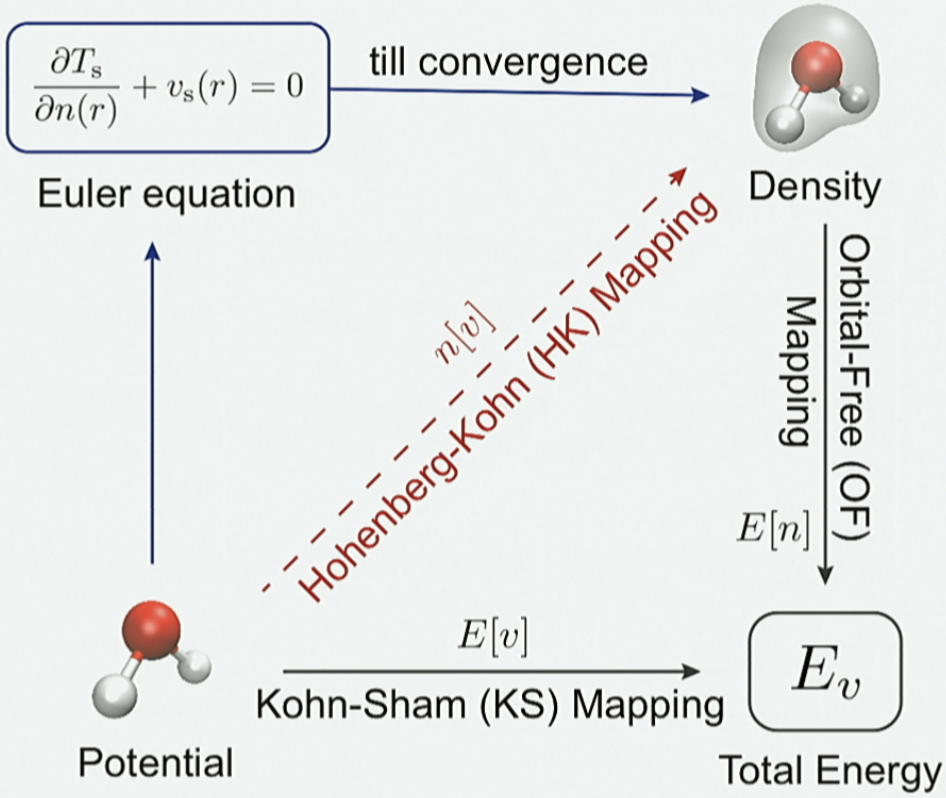

Li Li


Thomas E. Baker

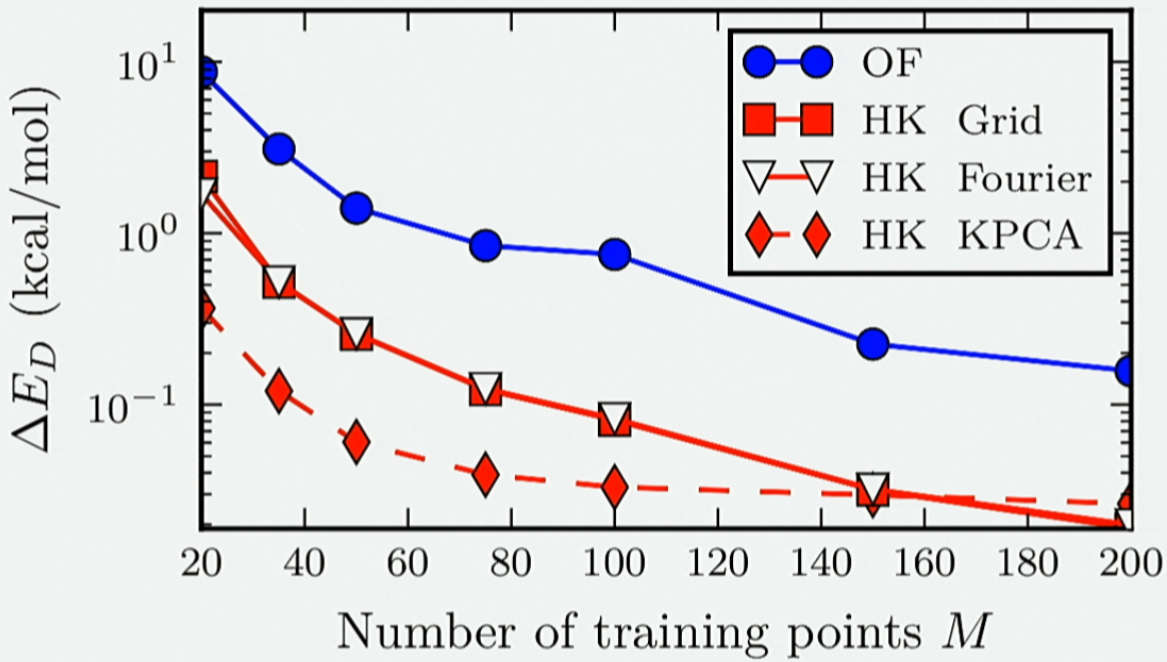
2 new papers

- *By-passing the KS equations with ML*
  - Felix Brockherde, Li Li, Klaus Muller, KB
  - Avoids functional derivative
  - Applied in 3D
  - Still doing KS problem, $T_s[n]$
- *Pure Density Functional for Strong Correlations and the Thermodynamic Limit Using Machine Learning*
  - Li Li, Thomas E. Baker, Steven R. White and KB
  - Do interacting functional (ie. Exact Exc)
  - Do strong correlation
  - Do thermodynamic limit
  - Still in 1d

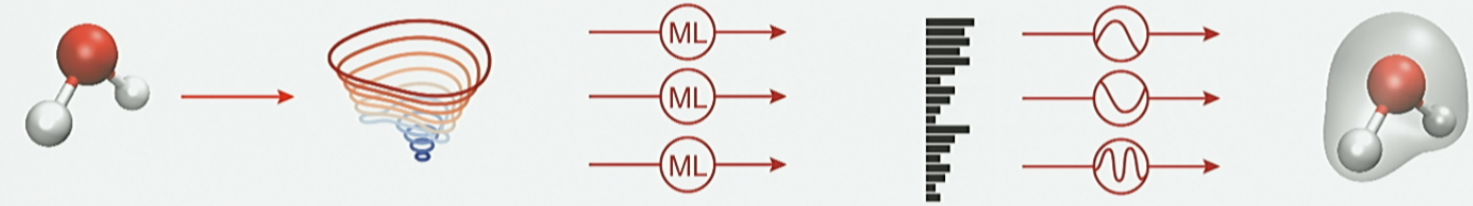
By-passing KS



Convergence of different HK maps



Non-interacting HK map



Potential

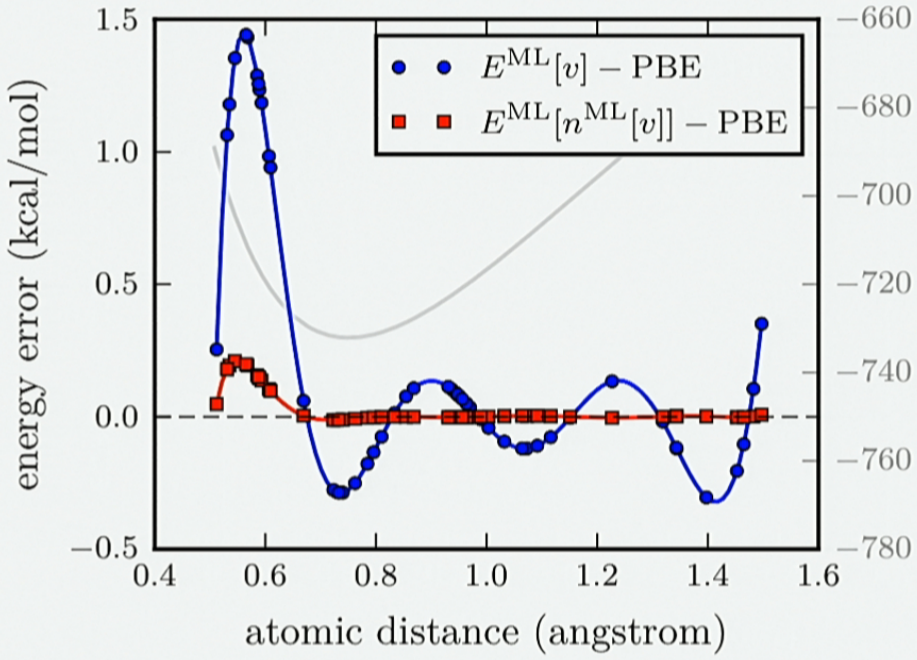
Potential as Gaussian blobs

Independent ML models

Data-driven and physicaly motivated basis represen-tations

Density

Error for H_2


$E^{\mathrm{ML}}[v] - \mathrm{PBE}$
$E^{\mathrm{ML}}[n^{\mathrm{ML}}[v]] - \mathrm{PBE}$
energy error (kcal/mol)
atomic distance (angstrom)
1.5
1.0
0.5
0.0
−0.5
0.4
0.6
0.8
1.0
1.2
1.4
1.6
−660
−680
−700
−720
−740
−760
−780

ML of exact functionals

- Use DMRG to solve continuum problems in 1d.
- Much success in past, showing failures of DFT approximations for strong correlation.
- Here use DMRG to generate much data of exact densities and energies
- All restricted to 1d.
- We train and test a machine learning F[n], the universal part of the electronic density functional, to within quantum chemical accuracy. We (a) bypass the standard Kohn-Sham approach, (b) include the strong correlation of highly-stretched bonds and (c) create a model for the infinite chain limit.

Guaranteed Convergence of the Kohn-Sham Equations Lucas O. Wagner, E. M. Stoudenmire, Kieron Burke, Steven R. White, Phys. Rev. Lett. **111**, 093003 (2013).

One-Dimensional Continuum Electronic Structure with the Density-Matrix Renormalization Group and Its Implication for Density-Functional Theory E.M. Stoudenmire, Lucas O. Wagner, Steven R. White, Kieron Burke, Phys. Rev. Lett. **109**, 056402 (2012).

Convergence for H_2

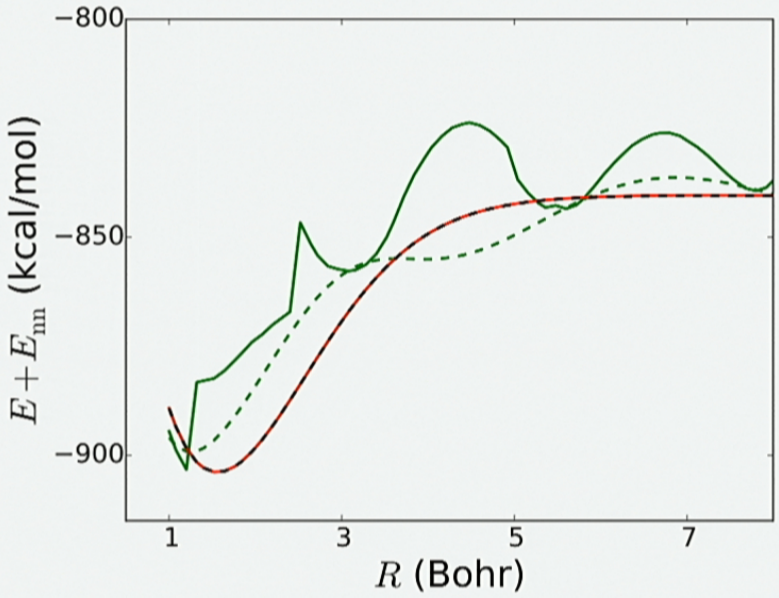


FIG. 3. (Color online) Same as Fig. 2. The green curves are ML with $N_{\rm T} = 5$ on both the exact (dashed) and ML-optimized (solid) densities. Red curves are the same with $N_{\rm T} = 20$.

Densities

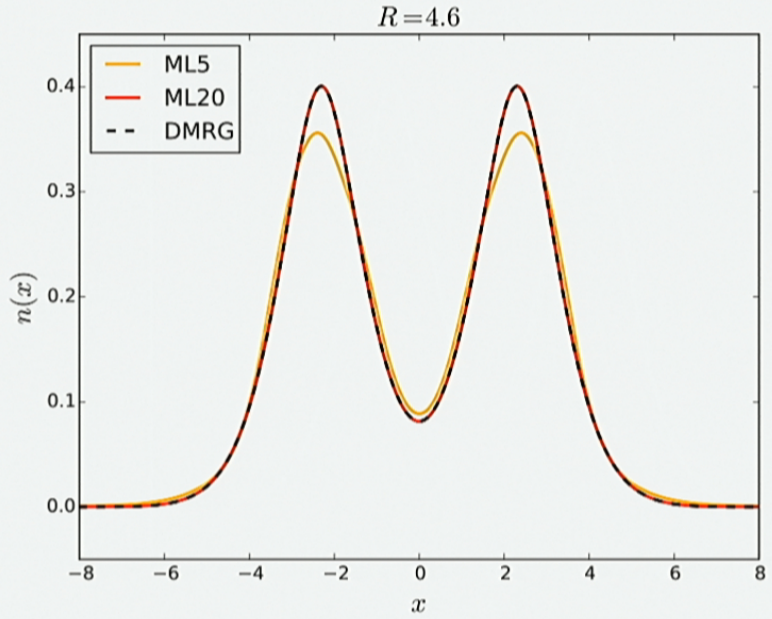


FIG. 4. (Color online) Optimal densities for 1d H_2 molecule in the test set: DMRG (black), ML with $N_T = 5$ (orange), ML with $N_T = 20$ (red).

Chains

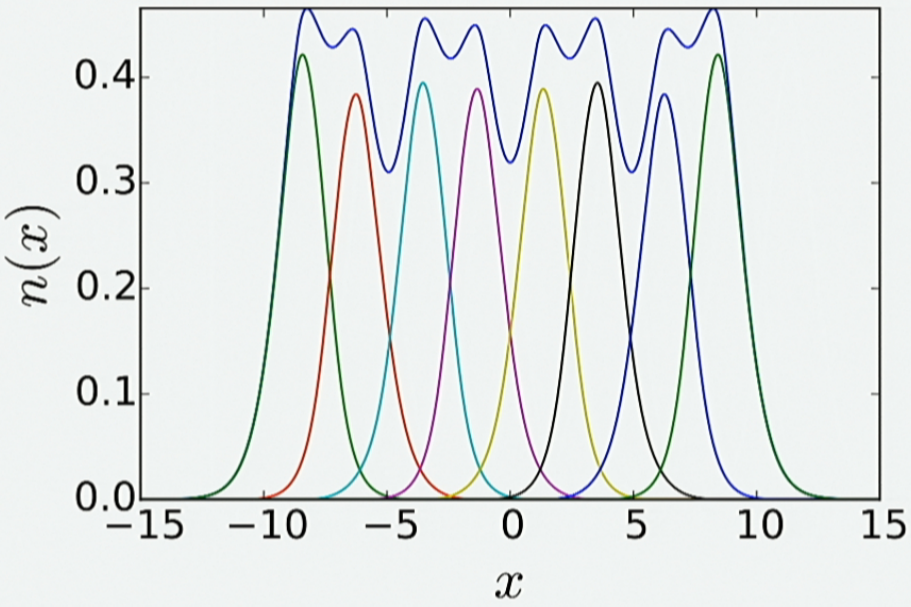
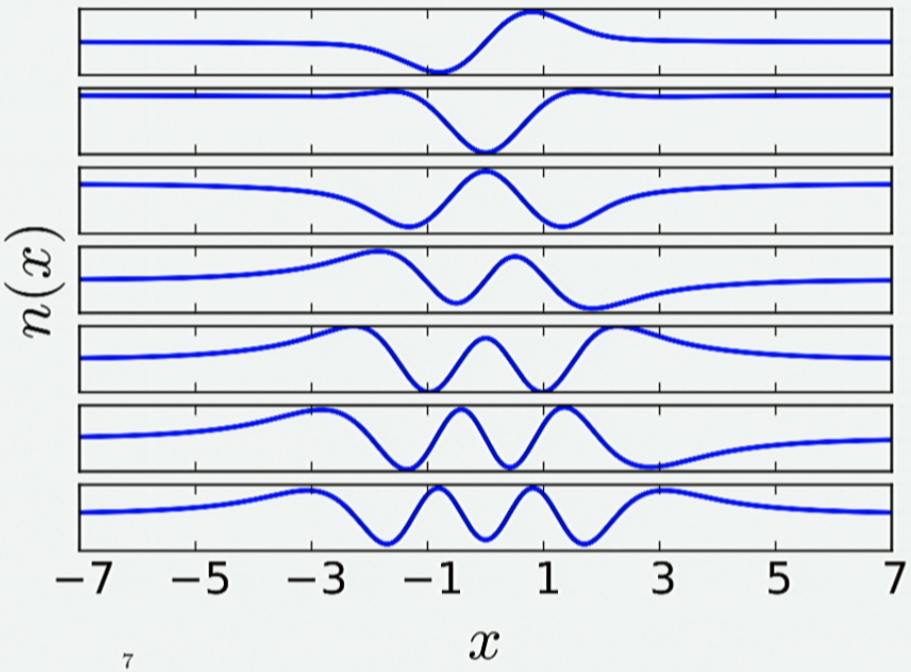


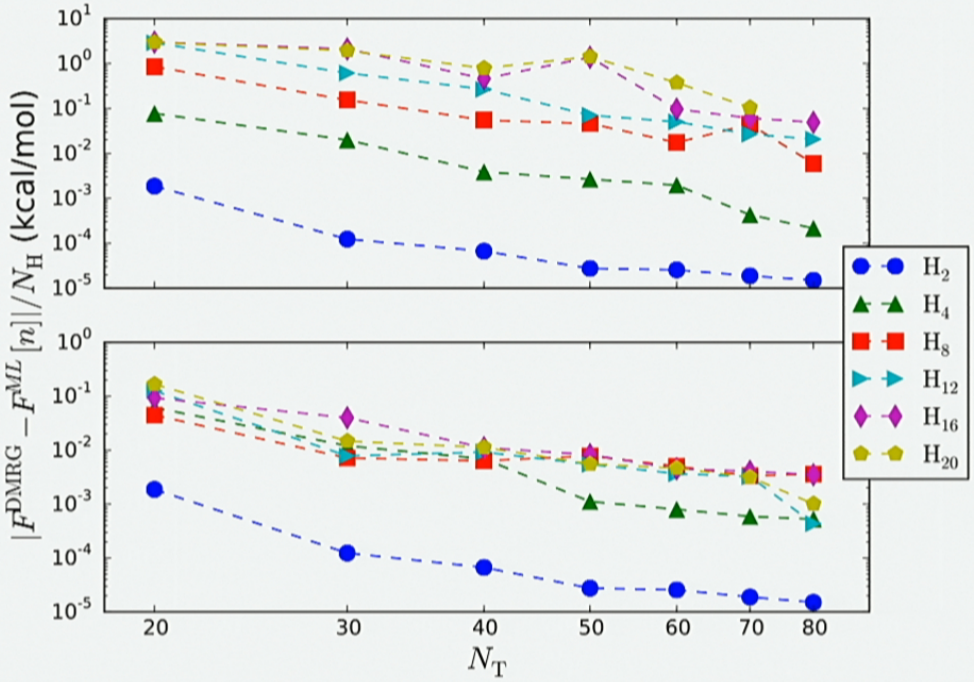
FIG. 5. Partition density of each H atom in H_8.

PCA basis for atomic densities



$$n_i(R,x) = f_0(x) + \sum_{p=1}^{7} c_{i,p}(R) f_p(x).$$

Improved convergence from basis



Infinite periodic solid

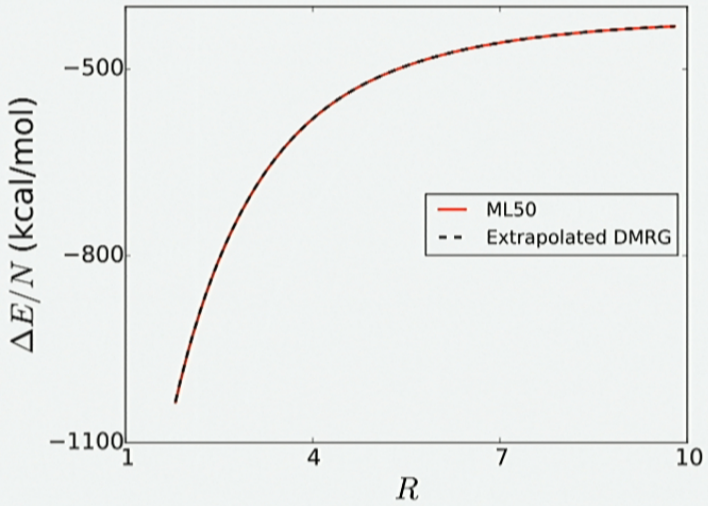


FIG. 9. (Color online) Electronic energy per atom in the thermodynamic limit, both via DMRG chains (extrapolated to infinity) and using machine learning with 50 data points per chain.

Convergence for infinite chain

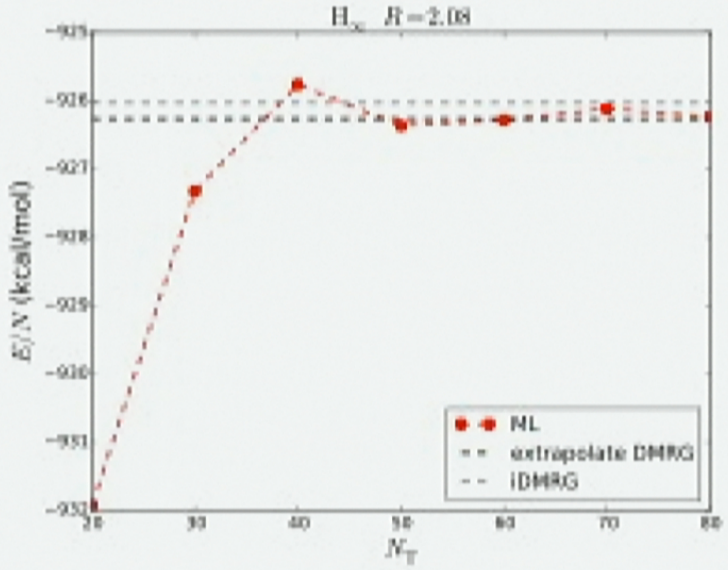


FIG. 1. (Color online) Electronic energy of infinite chain from model learned from extrapolated chain densities and energies.

Papers (all on dft.uci.edu)

Nonlinear gradient denoising: Finding accurate extrema from inaccurate functional derivatives John C. Snyder, Matthias Rupp, Klaus-Robert Müller, Kieron Burke, *International Journal of Quantum Chemistry* **115**, 1102--1114 (2015).

Understanding kernel ridge regression: Common behaviors from simple functions to density functionals Kevin Vu, John C. Snyder, Li Li, Matthias Rupp, Brandon F. Chen, Tarek Khelif, Klaus-Robert Müller, Kieron Burke, *International Journal of Quantum Chemistry* **115**, 1115--1128 (2015).

Understanding machine-learned density functionals Li Li, John C. Snyder, Isabelle M. Pelaschier, Jessica Huang, Uma-Naresh Niranjan, Paul Duncan, Matthias Rupp, Klaus-Robert Müller, Kieron Burke, *International Journal of Quantum Chemistry* n/a--n/a (2015).

Kernels, Pre-Images and Optimization John C. Snyder, Sebastian Mika, Kieron Burke, Klaus-Robert Müller, *Chapter in Empirical Inference - Festschrift in Honor of Vladimir N. Vapnik* (2013).

Orbital-free Bond Breaking via Machine Learning John C. Snyder, Matthias Rupp, Katja Hansen, Leo Blooston, Klaus-Robert Müller, Kieron Burke, *J. Chem. Phys.* **139**, 224104 (2013).

Finding Density Functionals with Machine Learning John C. Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, Kieron Burke, *Phys. Rev. Lett.***108**, 253002 (2012).

Summary

- ML of functionals works in model cases to produce highly accurate approximate functionals

Summary

- ML of functionals works in model cases to produce highly accurate approximate functionals
- Totally different approach from anything in DFT before

Summary

- ML of functionals works in model cases to produce highly accurate approximate functionals
- Totally different approach from anything in DFT before
- ML can even
  - find accurate densities
  - say when it will work within tolerance (makes Klaus nervous)
  - break bonds
  - Do the full functional

Summary

- ML of functionals works in model cases to produce highly accurate approximate functionals
- Totally different approach from anything in DFT before
- ML can even
  - find accurate densities
  - say when it will work within tolerance (makes Klaus nervous)
  - break bonds
  - Do the full functional
- But
  - only demonstrated in 1d
  - Need to do arbitrary-sized system (representation question)

Summary

- ML of functionals works in model cases to produce highly accurate approximate functionals
- Totally different approach from anything in DFT before
- ML can even
  - find accurate densities
  - say when it will work within tolerance (makes Klaus nervous)
  - break bonds
  - Do the full functional
- But
  - only demonstrated in 1d
  - Need to do arbitrary-sized system (representation question)
- Ongoing projects:
  - ML exact interacting functional in thermodynamic limit
  - Full 3D calculation of water molecule, using ML for n[v](r)

Summary

- ML of functionals works in model cases to produce highly accurate approximate functionals
- Totally different approach from anything in DFT before
- ML can even
  - find accurate densities
  - say when it will work within tolerance (makes Klaus nervous)
  - break bonds
  - Do the full functional
- But
  - only demonstrated in 1d
  - Need to do arbitrary-sized system (representation question)
- Ongoing projects:
  - ML exact interacting functional in thermodynamic limit
  - Full 3D calculation of water molecule, using ML for n[v](r)
- Thanks to
  - Students: Li Li, John Snyder, Kevin Vu, Isabelle Pelaschier
  - Collaborators: Klaus Mueller, Matthias Rupp, Katia Hansen
  - Funders: NSF from chem, DMR, math