Variational principle to regularize machine-learned density functionals

Pablo del Mazo Sevillano
pablo.delmazo@uam.es

Universidad Autónoma de Madrid
Departamento de Química Física Aplicada

Universidad Autónoma de Madrid

Motivation

- KS-DFT does no fully exploit the usage of the density
  $\rightarrow$ needs orbitals.

Motivation

- KS-DFT does no fully exploit the usage of the density
  $\rightarrow$ needs orbitals.

- In OF-DFT the kinetic energy cannot be computed exactly
  $\rightarrow$ we need accurate KEF.

Motivation

- KS-DFT does no fully exploit the usage of the density
  $\rightarrow$ needs orbitals.
- In OF-DFT the kinetic energy cannot be computed exactly
  $\rightarrow$ we need accurate KEF.
- Finding an accurate KEF has proven to be a hard task.
  Can we machine learn it?

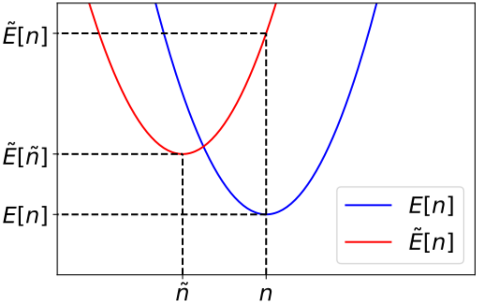
Motivation

- KS-DFT does no fully exploit the usage of the density
  $\rightarrow$ needs orbitals.
- In OF-DFT the kinetic energy cannot be computed exactly
  $\rightarrow$ we need accurate KEF.
- Finding an accurate KEF has proven to be a hard task.
  Can we machine learn it?
- No proper regularization technique for KEF
  $\rightarrow$ either computationally expensive or not applicable for OF-DFT.

Regularizing functional training

2 degrees of freedom in density functional fitting:

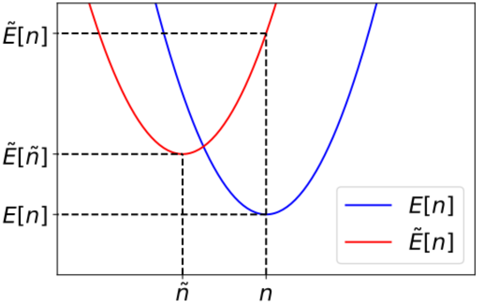
- Energy
- Density



Regularizing functional training

2 degrees of freedom in density functional fitting:

- Energy
- Density

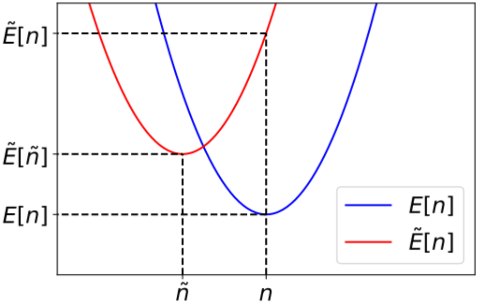


Density functionals cannot be trained just matching energies:

Regularizing functional training

2 degrees of freedom in density functional fitting:

- Energy
- Density



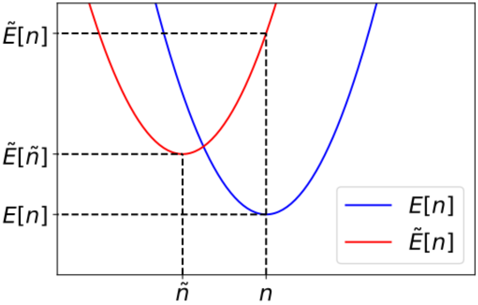
Density functionals cannot be trained just matching energies:

$$\mathcal{L} = \mathbb{E}[(E - E_{\text{ref}})^2]$$

Regularizing functional training

2 degrees of freedom in density functional fitting:

- Energy
- Density



Density functionals cannot be trained just matching energies:

$$\mathcal{L} = \mathbb{E}[(E - E_{\text{ref}})^2] + \mathcal{L}_{\text{reg}}$$

Regularizing functional training

- Ground-state electronic energy of an N-electron system:

$$E[n] = T_s[n] + \int v_{ext}(r)n(r)dr + J[n] + E_{xc}[n]$$

Regularizing functional training

- Ground-state electronic energy of an N-electron system:

$$E[n] = T_s[n] + \int v_{ext}(r)n(r)dr + J[n] + E_{xc}[n]$$

- In KS-DFT: $T_s[n] = -\frac{1}{2}\sum_i n_i \langle \psi_i | \nabla^2 | \psi_i \rangle$

Regularizing functional training

- Ground-state electronic energy of an N-electron system:

$$E[n] = T_s[n] + \int v_{ext}(r)n(r)dr + J[n] + E_{xc}[n]$$

- In KS-DFT: $T_s[n] = -\frac{1}{2}\sum_i n_i \langle \psi_i | \nabla^2 | \psi_i \rangle$
- In OF-DFT: $T_s[n] = -N\frac{1}{2}\langle \psi | \nabla^2 | \psi \rangle + T_p[n]$

Regularizing functional training

- Ground-state electronic energy of an N-electron system:

$$E[n] = T_s[n] + \int v_{ext}(r)n(r)dr + J[n] + E_{xc}[n]$$

- In KS-DFT: $T_s[n] = -\frac{1}{2}\sum_i n_i \langle\psi_i|\nabla^2|\psi_i\rangle$
- In OF-DFT: $T_s[n] = -N\frac{1}{2}\langle\psi|\nabla^2|\psi\rangle + T_p[n]$
- Variational principle to find ground-state electronic energy and density.

$$L[n] = E[n] - \sum_i \varepsilon_i(\langle\psi_i|\psi_i\rangle - 1) \rightarrow |g_j\rangle = \frac{\delta L}{\delta\psi_j}$$

$$|g_j\rangle = 0 \Rightarrow \left[-\frac{1}{2}\nabla^2 + v_{eff}\right]\psi_j = \varepsilon_j\psi_j$$

Regularizing functional training

- Ground-state electronic energy of an N-electron system:

$$E[n] = T_s[n] + \int v_{ext}(r)n(r)dr + J[n] + E_{xc}[n]$$

- In KS-DFT: $T_s[n] = -\frac{1}{2}\sum_i n_i \langle \psi_i | \nabla^2 | \psi_i \rangle$
- In OF-DFT: $T_s[n] = -N\frac{1}{2}\langle \psi | \nabla^2 | \psi \rangle + T_p[n]$
- Variational principle to find ground-state electronic energy and density.

$$L[n] = E[n] - \sum_i \varepsilon_i(\langle \psi_i | \psi_i \rangle - 1) \rightarrow |g_j\rangle = \frac{\delta L}{\delta \psi_j}$$

$$|g_j\rangle = 0 \Rightarrow \left[-\frac{1}{2}\nabla^2 + v_{eff} \right] \psi_j = \varepsilon_j \psi_j$$

* In an SCF calculation we look for the ψ_j that makes $|g_j\rangle = 0$.

Regularizing functional training

- Ground-state electronic energy of an N-electron system:

$$E[n] = T_s[n] + \int v_{ext}(r)n(r)dr + J[n] + E_{xc}[n]$$

- In KS-DFT: $T_s[n] = -\frac{1}{2}\sum_i n_i \langle\psi_i|\nabla^2|\psi_i\rangle$
- In OF-DFT: $T_s[n] = -N\frac{1}{2}\langle\psi|\nabla^2|\psi\rangle + T_p[n]$
- Variational principle to find ground-state electronic energy and density.

$$L[n] = E[n] - \sum_i \varepsilon_i(\langle\psi_i|\psi_i\rangle - 1) \rightarrow |g_j\rangle = \frac{\delta L}{\delta\psi_j}$$

$$|g_j\rangle = 0 \Rightarrow \left[-\frac{1}{2}\nabla^2 + v_{eff}\right]\psi_j = \varepsilon_j\psi_j$$

* In an SCF calculation we look for the ψ_j that makes $|g_j\rangle = 0$.
* In functional training we look for the hamiltonian (v_{eff}) that makes $|g_j\rangle = 0$ with fixed ψ_j.

Training density functionals

- Loss function:

$$\begin{aligned}\mathcal{L} =& \mathbb{E}[(\tilde{E}[n] - E_{\mathrm{ref}})^2] \\ &+ \mathbb{E}[\langle g|g\rangle]\end{aligned}$$

P. del Mazo-Sevillano, J. Hermann, arXiv:2306.17587, 2023.

Training density functionals

- Loss function:

$$\mathcal{L} = \mathbb{E}[(\tilde{E}[n] - E_{\text{ref}})^2] + \mathbb{E}[\langle g|g\rangle]$$

P. del Mazo-Sevillano, J. Hermann, arXiv:2306.17587, 2023.

- Functional:

$$F[n^\alpha, n^\beta] = \int \pm f_{\boldsymbol{\theta}}\big(\mathrm{z}[n^\alpha, n^\beta](\mathrm{r})\big)\, n(\mathrm{r})\, d\mathrm{r} \begin{cases} + \Rightarrow \text{Pauli functional} \\ - \Rightarrow \text{XC functional} \end{cases}$$

Training density functionals

- Loss function:

$$\mathcal{L} = \mathbb{E}[(\tilde{E}[n] - E_{\text{ref}})^2] + \mathbb{E}[\langle g|g\rangle]$$

P. del Mazo-Sevillano, J. Hermann, arXiv:2306.17587, 2023.

- Functional:

$$F[n^\alpha, n^\beta] = \int \pm f_{\boldsymbol{\theta}}\big(\mathrm{z}[n^\alpha, n^\beta](\mathrm{r})\big)\, n(\mathrm{r}) d\mathrm{r} \begin{cases} + \Rightarrow \text{Pauli functional} \\ - \Rightarrow \text{XC functional} \end{cases}$$

  * $f_{\boldsymbol{\theta}}$: Multilayer perceptron.

Training density functionals

- Loss function:

$$
\begin{aligned}
\mathcal{L} =& \mathbb{E}[(\tilde{E}[n] - E_{\text{ref}})^2] \\
&+ \mathbb{E}[\langle g | g \rangle]
\end{aligned}
$$

P. del Mazo-Sevillano, J. Hermann, arXiv:2306.17587, 2023.

- Functional:

$$
F[n^\alpha, n^\beta] = \int \pm f_{\boldsymbol{\theta}}\big(\mathrm{z}[n^\alpha, n^\beta](\mathrm{r})\big)\, n(\mathrm{r})\, d\mathrm{r} \begin{cases} + \Rightarrow \text{Pauli functional} \\ - \Rightarrow \text{XC functional} \end{cases}
$$

  * $f_{\boldsymbol{\theta}}$: Multilayer perceptron.
  * z: Features computed from the electron density.

Training density functionals

- Loss function:

$$\mathcal{L} = \mathbb{E}[(\tilde{E}[n] - E_{\text{ref}})^2] + \mathbb{E}[\langle g|g\rangle]$$

P. del Mazo-Sevillano, J. Hermann, arXiv:2306.17587, 2023.

- Functional:

$$F[n^\alpha, n^\beta] = \int \pm f_{\boldsymbol{\theta}}\big(\mathrm{z}[n^\alpha, n^\beta](\mathrm{r})\big)\, n(\mathrm{r})\, d\mathrm{r} \begin{cases} + \Rightarrow \text{Pauli functional} \\ - \Rightarrow \text{XC functional} \end{cases}$$

  * $f_{\boldsymbol{\theta}}$: Multilayer perceptron.
  * z: Features computed from the electron density.

- Data:

Training density functionals

- Loss function:

$$\mathcal{L} = \mathbb{E}[(\tilde{E}[n] - E_{\text{ref}})^2] + \mathbb{E}[\langle g|g\rangle]$$

  P. del Mazo-Sevillano, J. Hermann, arXiv:2306.17587, 2023.

- Functional:

$$F[n^\alpha, n^\beta] = \int \pm f_{\boldsymbol{\theta}}\big(\mathrm{z}[n^\alpha, n^\beta](\mathrm{r})\big)\, n(\mathrm{r})\, d\mathrm{r} \begin{cases} + \Rightarrow \text{Pauli functional} \\ - \Rightarrow \text{XC functional} \end{cases}$$

  * $f_{\boldsymbol{\theta}}$: Multilayer perceptron.
  * z: Features computed from the electron density.

- Data:
  * Accurate energies.

Training density functionals

- Loss function:

$$\mathcal{L} = \mathbb{E}[(\tilde{E}[n] - E_{\text{ref}})^2] + \mathbb{E}[\langle g|g\rangle]$$

P. del Mazo-Sevillano, J. Hermann, arXiv:2306.17587, 2023.

- Functional:

$$F[n^\alpha, n^\beta] = \int \pm f_{\boldsymbol{\theta}}\big(\mathrm{z}[n^\alpha, n^\beta](\mathrm{r})\big)\, n(\mathrm{r}) d\mathrm{r} \begin{cases} + \Rightarrow \text{Pauli functional} \\ - \Rightarrow \text{XC functional} \end{cases}$$

  * $f_{\boldsymbol{\theta}}$: Multilayer perceptron.
  * z: Features computed from the electron density.

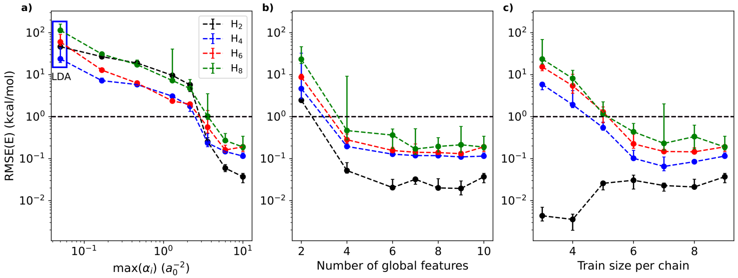
- Data:
  * Accurate energies.
  * Orbital coefficients obtained with other density functional.

Training the Pauli functional — 1D hydrogen chain

Reference data: KS-DFT/LDA (n=2,4,6,8)

$$z_i = \log(G[n](x;\alpha_i) + 10^{-4})$$

$$G[n](x;\alpha) = \frac{1}{\sqrt{2\pi\alpha}} \int_{-\infty}^{\infty} n(x') \exp\left(-\frac{(x-x')^2}{2\alpha}\right) dx'$$



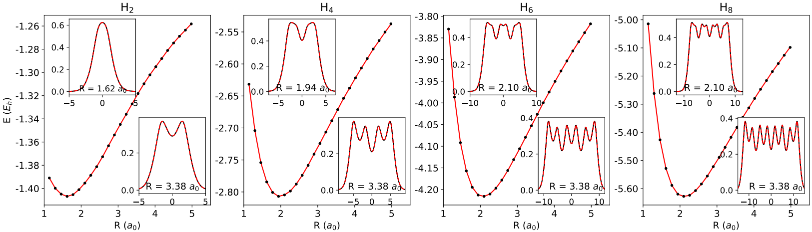
- The most relevant aspect is the inclusion of non-local features.

Training the Pauli functional — 1D hydrogen chain

Reference data: KS-DFT/LDA (n=2,4,6,8)

$$z_i = \log(G[n](x;\alpha_i) + 10^{-4})$$

$$G[n](x;\alpha) = \frac{1}{\sqrt{2\pi\alpha}} \int_{-\infty}^{\infty} n(x')\exp\left(-\frac{(x-x')^2}{2\alpha}\right) dx'$$



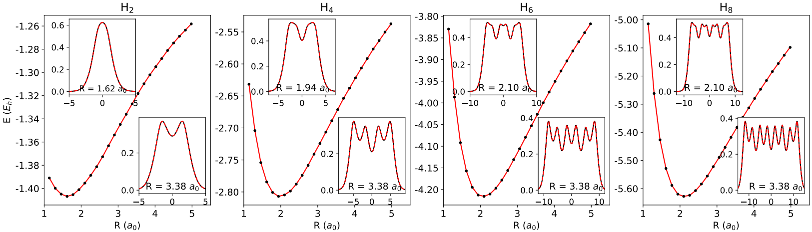
- The most relevant aspect is the inclusion of non-local features.

Training the Pauli functional — 1D hydrogen chain

Reference data: KS-DFT/LDA (n=2,4,6,8)

$$z_i = \log(G[n](x;\alpha_i) + 10^{-4})$$

$$G[n](x;\alpha) = \frac{1}{\sqrt{2\pi\alpha}} \int_{-\infty}^{\infty} n(x') \exp\left(-\frac{(x-x')^2}{2\alpha}\right) dx'$$



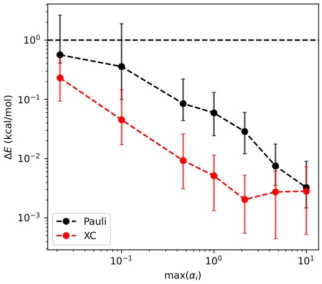
- The most relevant aspect is the inclusion of non-local features.
- The trained functional is able to generalize over new chain *geometries*.

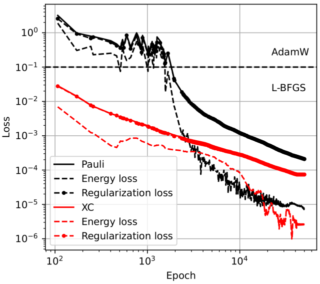
Training the Pauli and XC functional — Atoms (H-Ne)

Reference data (Pauli): KS-DFT/LDA
Reference data (XC): CCSD(T)/cc-pVTZ energies and KS-DFT/LDA-PW92 densities.

$$z_i^\sigma = \log(G[n^\sigma](r;\alpha_i) + 10^{-4})$$

$$G[n](r;\alpha) = \frac{1}{\sqrt{2\pi\alpha}} \int_0^\infty n(r') \exp\left(-\frac{(r-r')^2}{2\alpha}\right) dr'$$





Conclusions

- A new method to regularize the training of density functionals is proposed. It is based on the variational condition that the global energy functional presents a minima for the ground electronic state density.
- This regularization is directly applicable to the training of KE and XC functionals.
- It is computationally cheap and stable.
- Future attempts to train the KEF will heavily depend on the inclusion of non-local features.
- KEF is highly susceptible to overfitting on small datasets in contrast to XC functional. We expect larger datasets will be needed for the former functional to enable acquisition of sufficient physical knowledge.

THANKS FOR YOUR ATTENTION