HEIDELBERG UNIVERSITY
BACHELOR THESIS IN PHYSICS
INTERDISCIPLINARY CENTER FOR SCIENTIFIC COMPUTING (IWR)

Prime & Refine: A Two-Stage Neural Framework for Ground State Prediction in OF-DFT

Dominic Plein
info@splines.me

First supervisor	Prof. Dr. Fred Hamprecht
Second supervisor	Prof. Dr. Tristan Bereau
Advisor	Peter Lippmann

April 1, 2026

Prime & Refine:
A Two-Stage Neural Framework for Ground State Prediction in OF-DFT

Written by: Dominic Plein
ORCID: 0009-0008-5812-7326 | Email: info@splines.me | https://splines.me
First supervisor: Prof. Dr. Fred Hamprecht
Second supervisor: Prof. Dr. Tristan Bereau
Advisor: Peter Lippmann

Bachelor Thesis in Physics | April 1, 2026

Heidelberg University | Interdisciplinary Center for Scientific Computing (IWR)

This thesis was written using **Typst**, a modern typesetting system with an open-source compiler.

Declaration

GitHub Copilot [AI chatbot powered by Large Language Models (LLMs), integrated into VSCode] was used to assist with writing (formulation and wording) and coding. Any output by LLMs was critically reviewed and edited by the author, and all final decisions regarding content and phrasing were made by the author. The author is solely responsible for the content of this thesis.

I hereby declare that I have written this thesis independently and have not used any sources or aids other than those specified.

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, April 1, 2026

Dominic Plein

Abstract

Orbital-free density functional theory (OF-DFT) promises to reduce the computational cost of electronic structure calculations by replacing the Kohn–Sham orbital equations with a direct optimization over the electron density, but requires an accurate kinetic energy density functional (KEDF). Machine-learned approximations have recently shown promising results in this direction. This thesis builds on the OF-DFT project Structures25, and contributes two complementary strategies for improving ground state prediction.

First, we develop *Direct Ground State Prediction* (DGSP), where a graph neural network predicts the ground state density coefficients from the molecular geometry in a single forward pass, bypassing the iterative density optimization loop. With a natural L^2 loss that accounts for the basis function overlap, the Equiformer DGSP model achieves a mean L^2 density error of 0.0075 on the QM9 test set, roughly halving the baseline error at an order of magnitude faster inference.

Second, we propose a new data generation method that decouples perturbation-based training data from the Kohn–Sham SCF procedure. By perturbing from the converged ground state according to a user-specified probability distribution, we gain direct control over the distribution of training densities and eliminate the coverage gap near the ground state present in prior approaches. This opens a broadened design space: while initial experiments have not yet surpassed the strong Structures25 baseline in density optimization, the framework makes it straightforward to generate new perturbed datasets and explore further perturbed data generation strategies.

Both methods are combined in a *Prime & Refine* framework, where DGSP initializes a subsequent variational density optimization. We identify a "drifting" phenomenon in the low-error regime as a key open challenge.

Zusammenfassung

Orbitalfreie Dichtefunktionaltheorie (OF-DFT) zielt darauf ab, den Rechenaufwand quantenchemischer Berechnungen zu reduzieren, indem sie anstelle von Kohn-Sham-Orbitalen direkt mit der Elektronendichte arbeitet. Dafür ist jedoch ein genaues kinetisches Energiefunktional (KEDF) erforderlich. Maschinell gelernte Näherungen haben kürzlich vielversprechende Ergebnisse in dieser Richtung gezeigt. Diese Arbeit baut auf dem OF-DFT Projekt Structures25 auf und erweitert es um zwei komplementäre Strategien zur Verbesserung der Grundzustandsberechnung.

Der erste Beitrag ist die *direkte Grundzustandsvorhersage* (DGSP): Ein Graph-neuronales Netz sagt die Grundzustandsdichte-Koeffizienten in einem einzigen Forward-Pass direkt aus der Molekülgeometrie vorher und umgeht damit die iterative Dichteoptimierung vollständig. Durch Training mit einer physikalisch motivierten natürlichen L^2-Kostenfunktion, die die Überlappmatrix der Basisfunktionen einbezieht, erreicht das Equiformer-DGSP-Modell einen mittleren L^2-Dichtefehler von $0,0075$ auf dem QM9-Testset, was den Structures25 Referenzwert um etwa die Hälfte reduziert und dabei eine Größenordnung schneller in der Inferenz ist.

Zweitens schlagen wir eine neue Methode zur Datengenerierung vor, die unabhängig von Kohn–Sham SCF-Iterationen perturbierte Daten ausgehend vom Grundzustand gemäß einer benutzerdefinierten Wahrscheinlichkeitsverteilung erzeugt. Dadurch kann die Verteilung der Trainingsdichten direkt gesteuert werden und die in vorherigen Ansätzen vorhandene Lücke bei kleinen L^2-Abständen zum Grundzustand geschlossen werden. Obwohl erste Experimente die starke Structures25-Referenz in der Dichteoptimierung nicht übertreffen, ermöglicht das Framework die einfache Erzeugung neuer, modifizierter Datensätze und die Erforschung weiterer Strategien zur perturbierten Datengenerierung.

Beide Strategien werden in einem *Prime- & Refine*-Framework kombiniert, in dem DGSP einen initialen Startpunkt für eine anschließende variationale Dichteoptimierung liefert. Dabei bleibt eine offene Herausforderung ein “Drift”-Phänomen im Bereich kleiner Fehler.

Contents

1 Introduction

Understanding the electronic structure of molecules is one of the central problems in quantum chemistry. The properties of any chemical system, from the stability of a drug molecule to the reactivity of a catalyst, are ultimately governed by the arrangement of its electrons. In principle, all of this information is encoded in the solution of the Schrödinger equation. For a system of N electrons, the wave function $\psi(\boldsymbol{r_1}, \ldots, \boldsymbol{r_N})$ is a complex-valued function of $3N$ spatial variables, and solving for it numerically is computationally prohibitive for all but the smallest systems.

The Hohenberg–Kohn theorems [1] allow us to bypass the wave function entirely and work with the electron density $\rho(\boldsymbol{r})$ instead, which depends only on three spatial variables, regardless of the number of electrons. Figure 1.1 shows such a density for a small organic molecule. This result, together with the variational principle [1], reduces the problem from optimizing a $3N$-dimensional wave function to finding the density that minimizes a universal energy functional and lays the foundation of Density Functional Theory (DFT), one of the most widely used methods in electronic structure theory.

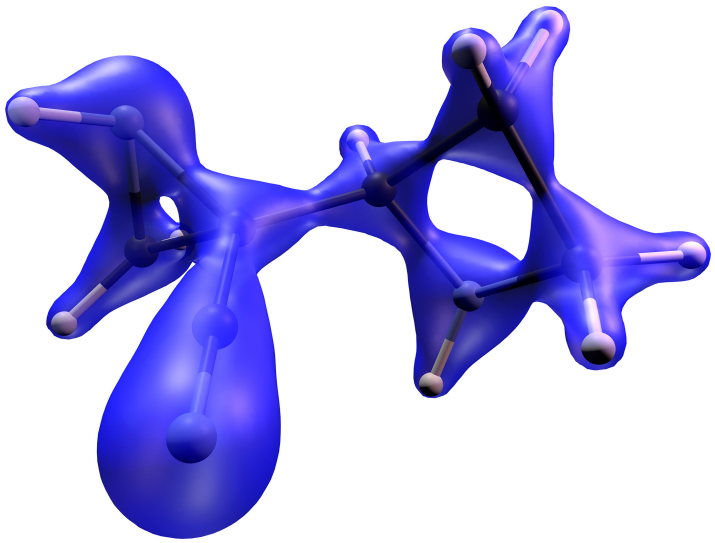


Figure 1.1: Isosurface of the ground-state electron density $\rho(\boldsymbol{r})$ for the molecule $H_9C_6N_3$, obtained via Kohn–Sham SCF iterations.

In practice, however, the exact energy functional is unknown, and the Kohn–Sham construction [2] introduces auxiliary orbitals to make the kinetic energy tractable, at the cost of reintroducing an orbital optimization step that scales with at least $\mathcal{O}(N^3)$, therefore becoming prohibitive for large systems such as proteins or extended materials.

Orbital-free density functional theory (OF-DFT) returns to the original density-only formulation and seeks to recover the ground state by minimizing directly over the density $\rho(\boldsymbol{r})$. If a sufficiently accurate kinetic energy density functional were available, this would promise a substantially cheaper route to electronic-structure calculations, reducing the scaling to potentially $\mathcal{O}(N)$.

Recent progress in machine learning has made orbital-free functionals for molecules increasingly practical. Equivariant graph neural networks such as KineticNet [3], M-OFDFT [4], and Structures25 [5] show that learned OF-DFT models can reach high accuracy on molecular benchmarks such as QM9 [6]. In [4] and [5], the density is represented in an atom-centered basis and a graph neural network is used to model an energy functional over the density coefficients, which is then minimized by variational density optimization.

Structures25 builds on M-OFDFT architecturally by introducing message passing on a graph with finite radius rather than global attention, and by generalizing the Graphormer backbone from scalar to tensorial messages between nodes, improving geometric expressivity. It also changes the learning target to the combined E_{TXC} functional and its gradient, which removes the need for an exchange-correlation quadrature grid during inference. Together with the more diverse densities generated by perturbing the effective potential during Kohn–Sham self-consistent field (SCF) iterations, these changes yield a well-formed energy landscape with a genuine minimum near the correct density, enabling fully convergent variational density optimization across all molecules in the QM9 test set with a mean absolute energy error of 0.64 mHa, below the "chemical accuracy" threshold of 1.6 mHa [5]. The present thesis builds on that line of work, both on the Structures25 pipeline and on the perturbation-based data generation first developed in [7] and [8].

Contributions of This Thesis

This thesis contributes two complementary strategies for improving ground state prediction in OF-DFT, which together form a *Prime & Refine* framework.

- **Prime** refers to *Direct Ground State Prediction* (DGSP, Chapter 4), where a graph neural network predicts the ground state density coefficients from the molecular geometry in a single forward pass, bypassing iterative optimization entirely. This yields a density that is already close to the ground state.
- **Refine** refers to subsequent variational Density Optimization (Denop, Chapter 5): starting from the DGSP prediction as an initial guess, the density is iteratively updated along the energy gradient of the learned functional until convergence. To improve the Graphormer model near the ground state, this thesis proposes a new data-generation scheme that decouples perturbation-based training data from the Kohn–Sham SCF procedure. Instead of perturbing SCF trajectories, it perturbs converged ground states with a user-defined probability distribution, giving better control over the training-density distribution. Training on samples concentrated near the ground state is expected to produce a more reliable optimization trajectory in that regime.

This work studies both components individually; results on their interaction when used together are still preliminary, as further development of the refine model is necessary to achieve a stable combined workflow.

Outline

The thesis is organized as follows:

- Chapter 2 introduces the DFT background needed for the remainder of the thesis, including the Hohenberg–Kohn theorems, the Kohn–Sham construction, and the basis representations used in the codebase.
- Chapter 3 summarizes Structures25, the machine-learned OF-DFT pipeline on which all experiments in this thesis build.
- Chapter 4 presents DGSP as the *prime* model, describes its architecture and loss, and evaluates direct ground-state prediction on QM9.
- Chapter 5 develops the *refine* model: it reviews perturbation-based data generation, introduces the new ground-state-centered perturbation framework, evaluates the resulting models in Density Optimization, and discusses the combined Prime–Refine workflow.
- Chapter 6 concludes with a summary of the results and the main open questions.

2 Density Functional Theory (DFT)

Density Functional Theory (DFT) replaces the many-electron wave function ψ with the electron density ρ as the central quantity to minimize over. This makes the high-dimensional problem of solving the Schrödinger equation for N electrons in $3N$ dimensions tractable by reducing it to a problem of optimizing a function of three spatial variables. In this chapter, we introduce the density, formulate the energy as a functional of that density within the Hohenberg–Kohn theorems, and derive the Kohn–Sham framework used in modern electronic-structure calculations.

2.1 Notation

Throughout this chapter:

- We consider a system of N electrons indexed by $i \in \{1, ..., N\}$.
- By $\boldsymbol{r}_i \in \mathbb{R}^3$, we denote the electron positions.
- Let $\psi(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) \in \mathcal{H}$ be the N-electron wave function in Hilbert space (ignoring spin) $\mathcal{H} = \bigwedge_{i=1}^{N} L^2(\mathbb{R}^3)$ subject to normalization

$$\langle \psi | \psi \rangle \stackrel{!}{=} 1, \quad \text{i.e.} \int_{\mathbb{R}^{3N}} |\psi(\boldsymbol{r}_1, ..., \boldsymbol{r}_N)|^2 \, \mathrm{d}\boldsymbol{r}_1 \cdots \mathrm{d}\boldsymbol{r}_N \stackrel{!}{=} 1 \tag{1}$$

- We assume a *non-degenerate* ground state ψ_0.
- We use atomic units where $\hbar = m_e = e = 1$ and $\frac{1}{4\pi\varepsilon_0} = 1$.

2.2 Electron Density

Definition 2.1 (Density): The **density operator** $\hat{\rho}$ for point $\boldsymbol{r} \in \mathbb{R}^3$ is given by

$$\hat{\rho}(\boldsymbol{r}) := \sum_{i=1}^{N} \delta(\boldsymbol{r} - \boldsymbol{r}_i) \tag{2}$$

The expectation value of the density operator is called the **electron density**

$$\rho(\boldsymbol{r}) : \mathbb{R}^3 \to \mathbb{R}, \quad \rho(\boldsymbol{r}) := \langle \psi | \hat{\rho}(\boldsymbol{r}) | \psi \rangle \tag{3}$$

We can explicitly calculate the density by plugging (2) into (3):

$$\rho(\boldsymbol{r}) := \langle \psi | \hat{\rho}(\boldsymbol{r}) | \psi \rangle = \int_{\mathbb{R}^{3N}} \psi^*(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) \left(\sum_{i=1}^{N} \delta(\boldsymbol{r} - \boldsymbol{r}_i) \right) \psi(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) \, \mathrm{d}\boldsymbol{r}_1 \cdots \mathrm{d}\boldsymbol{r}_N \tag{4}$$

$$= \sum_{i=1}^{N} \int_{\mathbb{R}^{3N}} \left| \psi(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) \right|^2 \delta(\boldsymbol{r} - \boldsymbol{r}_i) \, \mathrm{d}\boldsymbol{r}_1 \cdots \mathrm{d}\boldsymbol{r}_N \tag{5}$$

$$= N \int_{\mathbb{R}^{3(N-1)}} \left| \psi(\boldsymbol{r}, \boldsymbol{r}_2, ..., \boldsymbol{r}_N) \right|^2 \mathrm{d}\boldsymbol{r}_2 \cdots \mathrm{d}\boldsymbol{r}_N \tag{6}$$

In the last step, while evaluating the delta distribution, we used the anti-symmetry of the wave function ψ, rendering $|\psi(\boldsymbol{r}_1, ..., \boldsymbol{r}_N)|^2$ symmetric under permutations.

Corollary 2.2 (Normalization of density):

$$\int_{\mathbb{R}^3} \rho(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = N \tag{7}$$

Proof:

$$\int_{\mathbb{R}^3} \rho(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = N \underbrace{\int_{\mathbb{R}^{3N}} \left|\psi(\boldsymbol{r}, \boldsymbol{r}_2, ..., \boldsymbol{r}_N)\right|^2 \mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}_2 \cdots \mathrm{d}\boldsymbol{r}_N}_{=1 \text{ due to normalization } (1)} = N \tag{8}$$

■

2.3 Hohenberg–Kohn & Constrained Search

The presentation of this section (until (HK1)) follows Section I.1 of [1]. We consider a system of N electrons subject to an external potential $v(\boldsymbol{r})$ and the Coulomb repulsion V_{ee} (mutual electron-electron interaction). The Hamiltonian is given by:

$$\hat{H} := \hat{T} + \hat{V} + \hat{V}_{\text{ee}}, \quad \hat{T} := -\frac{1}{2}\sum_{i=1}^{N} \nabla_i^2, \quad \hat{V}_{\text{ext}} := \sum_{i=1}^{N} v(\boldsymbol{r}_i), \quad \hat{V}_{\text{ee}} := \sum_{1 \le i < j \le N} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} \tag{9}$$

Corollary 2.3: The potential energy V_{ext} (also called E_{ext}) can be expressed as

$$V_{\text{ext}} := \left\langle \psi \middle| \hat{V}_{\text{ext}} \middle| \psi \right\rangle = \int_{\mathbb{R}^3} v(\boldsymbol{r})\rho(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{10}$$

Proof: With (2), we can also write operator $\hat{V}_{\text{ext}}$ as

$$\hat{V}_{\text{ext}} = \sum_{i=1}^{N} v(\boldsymbol{r}_i) = \int_{\mathbb{R}^3} v(\boldsymbol{r}) \left(\sum_{i=1}^{N} \delta(\boldsymbol{r} - \boldsymbol{r}_i) \right) \mathrm{d}\boldsymbol{r} \overset{(2)}{=} \int_{\mathbb{R}^3} v(\boldsymbol{r})\hat{\rho}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{11}$$

From this, we directly obtain the desired result:

$$V_{\text{ext}} := \left\langle \psi \middle| \hat{V}_{\text{ext}} \middle| \psi \right\rangle \overset{(3)}{=} \int_{\mathbb{R}^3} v(\boldsymbol{r}) \underbrace{\langle \psi | \hat{\rho}(\boldsymbol{r}) | \psi \rangle}_{\rho(\boldsymbol{r})}\,\mathrm{d}\boldsymbol{r} \tag{12}$$

■

Theorem 2.4 (Hohenberg–Kohn 1): The ground state density $\rho^*(\boldsymbol{r})$ determines the external potential $v(\boldsymbol{r})$ up to an additive constant.

There exists only *one* potential $v(\boldsymbol{r})$ (up to an additive constant) that leads to a density $\rho^*(\boldsymbol{r})$. The correspondence between the ground state density $\rho^*(\boldsymbol{r})$ and the external potential $v(\boldsymbol{r})$ is one-to-one. We have the following bijections:

$$v \leftrightarrow \psi^* \leftrightarrow \rho^* \tag{13}$$

Corollary 2.5: $\rho^*(\boldsymbol{r})$ uniquely determines the external potential operator $\hat{V}_{\text{ext}}$, because $v(\boldsymbol{r})$ is fixed by $\rho^*(\boldsymbol{r})$. Since $\hat{T}$ and $\hat{V}_{\text{ee}}$ are known, the full Hamiltonian $\hat{H}$ is fixed by $\rho^*(\boldsymbol{r})$, and thus also ψ^* (finding ψ^* would entail solving the Schrödinger equation). The ground state density ρ^* uniquely determines the ground state energy E^* by this chain:

$$\rho^* \leftrightarrow V_{\text{ext}} \leftrightarrow H \leftrightarrow \psi^* \rightarrow E^* \tag{14}$$

For a proof of Theorem 2.4, we refer the reader to Chapter 4 in [9] and to [1].

Now for the second Hohenberg–Kohn theorem, we consider the variational optimization problem over anti-symmetric wave functions $\psi \in \mathcal{H}$ (respecting the normalization condition (1)) to obtain the ground state energy E^*:

$$E^* = \min_{\psi \in \mathcal{H}} \left\langle \psi \middle| \hat{H} \middle| \psi \right\rangle = \min_{\psi \in \mathcal{H}} \left\langle \psi \middle| \hat{T} + \hat{V}_{\text{ee}} + \hat{V}_{\text{ext}} \middle| \psi \right\rangle \tag{15}$$

Following Levy–Lieb's constrained-search formulation [10, 11], we define:

Definition 2.6: The universal functional is defined as

$$Q[\rho] := \min_{\substack{\psi \in \mathcal{H} \\ \langle \psi | \hat{\rho} | \psi \rangle = \rho}} \left\langle \psi \middle| \hat{T} + \hat{V}_{\text{ee}} \middle| \psi \right\rangle \tag{16}$$

By $\langle \psi | \hat{\rho} | \psi \rangle = \rho$, we denote the constraint that the wave function ψ must give rise to the density ρ.

The original Hohenberg–Kohn universal functional $F[\rho] := \left\langle \psi \middle| \hat{T} + \hat{V}_{\text{ee}} \middle| \psi \right\rangle$ is only defined for v-representable densities. A density ρ is called v-representable if there exists an external potential $v(\boldsymbol{r})$, such that ρ is the ground-state density of the corresponding Hamiltonian $\hat{H} = \hat{T} + \hat{V}_{\text{ee}} + \hat{V}_{\text{ext}}$. In other words, a density is v-representable if it can be obtained from an antisymmetric *ground-state* wave function of some physically realizable electronic system.

The constrained-search functional $Q[\rho]$ extends this to all N-representable densities, i.e. all densities that can be obtained from *some* anti-symmetric wave function. This set is strictly larger than the set of v-representable densities, because not every anti-symmetric wave function is a *ground state* for some external potential. One can show that $Q[\rho] = F[\rho]$ whenever ρ is v-representable. In the following, we will only consider $Q[\rho]$. $Q[\rho]$ is also named $U[\rho]$ (for universal functional) in some literature [4].

Definition 2.7: The energy functional is defined as

$$E[\rho] \coloneqq Q[\rho] + E_{\text{ext}}[\rho], \quad E_{\text{ext}}[\rho] \overset{(10)}{=} \int_{\mathbb{R}^3} v(\boldsymbol{r})\rho(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \tag{17}$$

Definition 2.8: We define the set $\mathcal{D}$ of admissible densities

$$\mathcal{D} \coloneqq \left\{\rho : \mathbb{R}^3 \to \mathbb{R} \mid \rho \text{ is } N\text{-representable}\right\} \tag{18}$$

Theorem 2.9 (Hohenberg–Kohn 2 under the Levy–Lieb constrained search formulation): The non-degenerate ground state electron density $\rho^*(\boldsymbol{r})$ minimizes the energy functional $E[\rho]$ over all N-representable densities $\rho(\boldsymbol{r})$:

$$E^* = \min_{\rho\in\mathcal{D}} E[\rho] \tag{19}$$

That is, we can rewrite the optimization problem over wave functions (15) as two-step optimization: first, minimize over wave functions that yield a given density ρ to obtain the universal functional $Q[\rho]$, and then minimize over densities to obtain the ground state energy E^*:

$$E^* = \min_{\rho\in\mathcal{D}} E[\rho] = \min_{\rho\in\mathcal{D}}(Q[\rho] + E_{\text{ext}}[\rho]) \tag{20}$$

$$= \min_{\rho\in\mathcal{D}} \left(\min_{\substack{\psi\in\mathcal{H} \\ \langle\psi|\hat{\rho}|\psi\rangle=\rho}} \left(\left\langle \psi \middle| \hat{T} + \hat{V}_{\text{ee}} \middle| \psi \right\rangle \right) + E_{\text{ext}}[\rho] \right) \tag{21}$$

The theorem guarantees that this minimization yields the same ground state energy E^* as the variational principle over wave functions in (15).

We denote the minimizing ground state density as $\rho^*(\boldsymbol{r}) \coloneqq \operatorname{argmin}_{\rho\in\mathcal{D}} E[\rho]$.

For a proof of Theorem 2.9, we refer the reader to the original paper for the constrained-search formulation [10] and to Chapter 4 in [9].

In (20), the kinetic and interaction energy are collected in the universal functional $Q[\rho]$ that is independent of the external potential $v(\boldsymbol{r})$ and the molecular system $\mathcal{M}$. Assuming the explicit form of $Q[\rho]$ was known, we could solve the optimization problem (20) over densities $\rho(\boldsymbol{r})$ on $\mathbb{R}^3$, instead of the original optimization (15) over wave functions $\psi(\boldsymbol{r}_1, ..., \boldsymbol{r}_N)$ on $\mathbb{R}^{3N}$, thus avoiding the exponential cost (when discretizing the wave function on a grid).

For any two systems with the same number of electrons N, the operators $\hat{T}$ and $\hat{V}_{\text{ee}}$ don't differ, thus $Q[\rho]$ is a universal functional that does not depend on the system. The only system-specific part of the energy functional is $E_{\text{ext}}[\rho]$, which depends on the external potential $v(\boldsymbol{r})$ that differs between systems. Thus, also the ground state density $\rho^*(\boldsymbol{r})$ that minimizes the energy functional differs between systems.

2.4 Approximations to the Energy Functional

Unfortunately, the universal functional $Q[\rho]$ is generally not known in closed form, and thus the variational principle (19) cannot be used directly. The main challenge of DFT is to find good approximations to $Q[\rho]$ that are accurate and computationally efficient. To this end, we extract from $Q[\rho]$ the parts that can be handled explicitly and collect the rest into an exchange-correlation functional $E_{\text{XC}}[\rho]$ that must be approximated.

$$E[\rho] \overset{(17)}{:=} Q[\rho] + E_{\text{ext}}[\rho] \overset{(16)}{:=} \underbrace{\min_{\substack{\psi \in \mathcal{H} \\ \langle\psi|\hat{\rho}|\psi\rangle = \rho}} \left\langle \psi \middle| \hat{T} + \hat{V}_{\text{ee}} \middle| \psi \right\rangle}_{\approx T_{\text{S}}[\rho] + E_{\text{H}}[\rho]} + E_{\text{ext}}[\rho] \tag{22}$$

$$= T_{\text{S}}[\rho] + E_{\text{H}}[\rho] + E_{\text{XC}}[\rho] + E_{\text{ext}}[\rho] \tag{23}$$

The **kinetic energy density functional (KEDF)** $T_{\text{S}}[\rho]$ is defined as its own constrained search, minimizing kinetic energy over all wave functions ψ consistent with density ρ:

$$T_{\text{S}}[\rho] := \min_{\substack{\psi \in \mathcal{H} \\ \langle\psi|\hat{\rho}|\psi\rangle = \rho}} \left\langle \psi \middle| \hat{T} \middle| \psi \right\rangle \tag{24}$$

This makes the KEDF a true functional of the density ρ, independent of the wave function ψ. We will see in Section 2.5 how $T_{\text{S}}[\rho]$ can be computed exactly from Kohn–Sham orbitals.

Note how $T_{\text{S}}[\rho]$ is an approximation to the kinetic contribution in $Q[\rho]$. Since the minimizer of $\hat{T} + \hat{V}_{\text{ee}}$ is generally different from the minimizer of $\hat{T}$ alone, and

$$\tilde{\psi}^* := \operatorname*{argmin}_{\substack{\psi \in \mathcal{H} \\ \langle\psi|\hat{\rho}|\psi\rangle = \rho}} \left\langle \psi \middle| \hat{T} + \hat{V}_{\text{ee}} \middle| \psi \right\rangle \tag{25}$$

is a feasible but not necessarily optimal choice for the T_{S} minimization, we have:

$$T_{\text{S}}[\rho] = \min_{\substack{\psi \in \mathcal{H} \\ \langle\psi|\hat{\rho}|\psi\rangle = \rho}} \left\langle \psi \middle| \hat{T} \middle| \psi \right\rangle \leq \left\langle \tilde{\psi}^* \middle| \hat{T} \middle| \tilde{\psi}^* \right\rangle \tag{26}$$

The deficit is called kinetic correlation and is absorbed into E_{XC}.

The **classical Hartree energy** $E_{\mathrm{H}}[\rho]$ approximates the electron-electron interaction by treating the density as a classical continuous charge distribution:

$$E_{\mathrm{H}}[\rho] \coloneqq \frac{1}{2} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\rho(\boldsymbol{r})\rho(\boldsymbol{r}')}{\|\boldsymbol{r} - \boldsymbol{r}'\|} \,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}' \tag{27}$$

This is an approximation because the true interaction $\left\langle \psi \middle| \hat{V}_{\mathrm{ee}} \middle| \psi \right\rangle$ depends on the two-body reduced density matrix (not just ρ), i.e. on quantum correlations between pairs of electrons. The discrepancy is also absorbed into E_{XC}. Note that unlike $\left\langle \psi \middle| \hat{V}_{\mathrm{ee}} \middle| \psi \right\rangle$, $E_{\mathrm{H}}[\rho]$ depends only on ρ (and not on ψ) and is therefore an explicit density functional.

The remaining part of $Q[\rho]$ is collected in the **exchange-correlation functional** $E_{\mathrm{XC}}[\rho]$ that accounts for all quantum effects beyond the classical Hartree energy and the KEDF.

$$E_{\mathrm{XC}}[\rho] \coloneqq Q[\rho] - T_{\mathrm{S}}[\rho] - E_{\mathrm{H}}[\rho] \tag{28}$$

The **external energy** $E_{\mathrm{ext}}[\rho] \coloneqq \left\langle \psi \middle| \hat{V}_{\mathrm{ext}} \middle| \psi \right\rangle$ can be evaluated exactly for nuclei positions $\boldsymbol{R}_a$ and nuclear charges Z_a (where $a \in \{1, \ldots, A\}$ indexes the nuclei):

$$E_{\mathrm{ext}}[\rho] \overset{(10)}{=} \int_{\mathbb{R}^3} \rho(\boldsymbol{r}) v(\boldsymbol{r}) \,\mathrm{d}\boldsymbol{r} = -\int_{\mathbb{R}^3} \mathrm{d}\boldsymbol{r}\, \rho(\boldsymbol{r}) \sum_{a=1}^{A} \frac{Z_a}{\|\boldsymbol{r} - \boldsymbol{R}_a\|} \tag{29}$$

2.5 Kohn–Sham

This part follows end of Section A.1 to Section A.3 of [4].

Unfortunately, there exists no known exact closed-form expression, nor a satisfactory approximation for the kinetic energy density functional (KEDF) $T_{\mathrm{S}}[\rho]$. The key insight of Kohn and Sham [2] was to instead introduce an auxiliary system of N non-interacting electrons, which allows us to compute the kinetic energy exactly from the orbitals of the non-interacting system.

This non-interacting system is described by a set of N orthonormal one-electron wave functions (orbitals) $\boldsymbol{\Phi} \coloneqq \{\phi_i(\boldsymbol{r})\}_{i=1}^{N}$ that are solutions to a single-particle Schrödinger equation with an effective potential. The wave function of the non-interacting system is then described by a single Slater determinant wave function ψ, which is the simplest anti-symmetric N-electron wave function respecting the Pauli exclusion principle.

$$\psi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) = \frac{1}{\sqrt{N!}} \det\left(\phi_i(\boldsymbol{r}_j)\right)_{ij} = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\boldsymbol{r}_1) & \phi_1(\boldsymbol{r}_2) & \cdots & \phi_1(\boldsymbol{r}_N) \\ \phi_2(\boldsymbol{r}_1) & \phi_2(\boldsymbol{r}_2) & \cdots & \phi_2(\boldsymbol{r}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\boldsymbol{r}_1) & \phi_N(\boldsymbol{r}_2) & \cdots & \phi_N(\boldsymbol{r}_N) \end{vmatrix} \tag{30}$$

Theorem 2.10: The electron density for orthonormal orbitals $\mathbf{\Phi} \coloneqq \{\phi_i(\boldsymbol{r})\}_{i=1}^N$ is

$$\rho_{\mathbf{\Phi}}(\boldsymbol{r}) \overset{(3)}{=} \langle\psi|\hat{\rho}(\boldsymbol{r})|\psi\rangle = \sum_{i=1}^{N}|\phi_i(\boldsymbol{r})|^2 \tag{31}$$

We prove this theorem on page 60 in the appendix.

The optimization problem for the kinetic energy in (24) can now be rewritten as an optimization problem over the orbitals $\mathbf{\Phi}$ that generate the density ρ. The equivalence to the original definition of $T_\mathrm{S}[\rho]$ in (24) is shown in Theorem 4.6 (and 4.9) in [11]. The index "S" in $T_\mathrm{S}[\rho]$ stands for S-system, referring to "single Slater determinant".

$$T_\mathrm{S}[\rho] \overset{(24)}{\coloneqq} \min_{\substack{\psi\in\mathcal{H}\\ \langle\psi|\hat{\rho}|\psi\rangle=\rho}} \left\langle\psi\middle|\hat{T}\middle|\psi\right\rangle = \min_{\substack{\mathbf{\Phi}\\ \rho_{\mathbf{\Phi}}=\rho}} \sum_{i=1}^{N}\left\langle\phi_i\middle|\hat{T}\middle|\phi_i\right\rangle \tag{32}$$

This expression also justifies the word *non-interacting* kinetic energy, as we sum over the kinetic energy of each orbital separately, without any interaction terms between the orbitals.

Finally, we can rewrite the full energy functional (23) in terms of the orbitals $\mathbf{\Phi}$ by substituting the Slater-determinant kinetic energy from (32).

$$E[\rho] = T_\mathrm{S}[\rho] + E_\mathrm{H}[\rho] + E_\mathrm{XC}[\rho] + E_\mathrm{ext}[\rho] \tag{33}$$

$$= \min_{\substack{\mathbf{\Phi}\\ \rho_{\mathbf{\Phi}}=\rho}} \sum_{i=1}^{N}\left\langle\phi_i\middle|\hat{T}\middle|\phi_i\right\rangle + E_\mathrm{H}[\rho] + E_\mathrm{XC}[\rho] + E_\mathrm{ext}[\rho] \tag{34}$$

The minimization yielding the ground state energy E^* can then be written as

$$E^* \overset{(19)}{=} \min_{\rho\in\mathcal{D}} E[\rho] = \min_{\rho\in\mathcal{D}}\left(\min_{\substack{\mathbf{\Phi}\\ \rho_{\mathbf{\Phi}}=\rho}} \sum_{i=1}^{N}\left\langle\phi_i\middle|\hat{T}\middle|\phi_i\right\rangle + E_\mathrm{H}[\rho] + E_\mathrm{XC}[\rho] + E_\mathrm{ext}[\rho]\right) \tag{35}$$

$$= \min_{\mathbf{\Phi}}\left(\sum_{i=1}^{N}\left\langle\phi_i\middle|\hat{T}\middle|\phi_i\right\rangle + E_\mathrm{H}[\rho_{\mathbf{\Phi}}] + E_\mathrm{XC}[\rho_{\mathbf{\Phi}}] + E_\mathrm{ext}[\rho_{\mathbf{\Phi}}]\right) \tag{36}$$

$$= \min_{\mathbf{\Phi}}\left(\sum_{i=1}^{N}\left\langle\phi_i\middle|\hat{T}\middle|\phi_i\right\rangle + E_\mathrm{eff}[\rho_{\mathbf{\Phi}}]\right) =: \min_{\mathbf{\Phi}} E[\mathbf{\Phi}] \tag{37}$$

The step from two-level optimization over densities and orbitals to a single-level optimization over just the orbitals $\mathbf{\Phi}$ is justified by the fact that the density ρ is generated by the orbitals $\mathbf{\Phi}$, so we can directly optimize over the orbitals that generate admissible densities. In the last line, we have introduced the effective potential $E_\mathrm{eff}[\rho]$.

$$E_\mathrm{eff}[\rho] \coloneqq E_\mathrm{H}[\rho] + E_\mathrm{XC}[\rho] + E_\mathrm{ext}[\rho] \tag{38}$$

This appears to be a step backward compared to the density-only formulation of (HK2), since we now optimize N orbitals instead of a single density. The advantage is that T_S is no longer an unknown functional: it is computed directly from the orbitals. The only remaining approximation in the energy is the exchange-correlation functional $E_\mathrm{XC}[\rho]$.

Variational problem over orbitals

In order to solve the minimization problem $E^* = \min_{\mathbf{\Phi}} E[\mathbf{\Phi}]$ in (37), we consider the Lagrangian formulation of the problem:

$$\mathcal{L}[\mathbf{\Phi}, \boldsymbol{\varepsilon}] := E[\mathbf{\Phi}] - \sum_{i=1}^{N} \varepsilon_i \Big(\langle \phi_i | \phi_i \rangle - 1 \Big) \tag{39}$$

The Lagrange multipliers ε_i enforce the normalization of each orbital: $\forall i : \langle \phi_i | \phi_i \rangle = 1$. The stationary condition $\forall i : \frac{\delta \mathcal{L}}{\delta \phi_i} \overset{!}{=} 0$ now yields the Euler–Lagrange equation corresponding to (37). For the energy functional $E[\mathbf{\Phi}]$, the variation with respect to orbitals $\mathbf{\Phi}$ is given by

$$\frac{\delta E[\mathbf{\Phi}]}{\delta \phi_i}(\boldsymbol{r}) = \frac{\delta \left\langle \phi_i \middle| \hat{T} \middle| \phi_i \right\rangle}{\delta \phi_i}(\boldsymbol{r}) + \int \underbrace{\frac{\delta E_\mathrm{eff}[\rho_{\mathbf{\Phi}}]}{\delta \rho_{\mathbf{\Phi}}(\boldsymbol{r}')}}_{V_\mathrm{eff}[\rho_{\mathbf{\Phi}}](\boldsymbol{r}')} \frac{\delta \rho_{\mathbf{\Phi}}(\boldsymbol{r}')}{\delta \phi_i(\boldsymbol{r})} \, \mathrm{d}\boldsymbol{r}' \tag{40}$$

$$\frac{\delta \left\langle \phi_i \middle| \hat{T} \middle| \phi_i \right\rangle}{\delta \phi_i}(\boldsymbol{r}) \overset{(9)}{=} \frac{\delta \left(\sum_{j=1}^{N} -\frac{1}{2} \langle \phi_j | \nabla^2 | \phi_j \rangle \right)}{\delta \phi_i(\boldsymbol{r})} = -\nabla^2 \phi_i(\boldsymbol{r}) \overset{(9)}{=} 2\hat{T}\phi_i(\boldsymbol{r}) \tag{41}$$

$$V_\mathrm{eff}[\rho](\boldsymbol{r}) := \frac{\delta E_\mathrm{eff}[\rho]}{\delta \rho(\boldsymbol{r})} \overset{(38)}{=} \frac{\delta E_\mathrm{H}[\rho] + E_\mathrm{XC}[\rho] + E_\mathrm{ext}[\rho]}{\delta \rho(\boldsymbol{r})} \overset{(27)}{=} \underbrace{\int \frac{\rho(\boldsymbol{r}')}{\|\boldsymbol{r} - \boldsymbol{r}'\|} \, \mathrm{d}\boldsymbol{r}'}_{=: V_\mathrm{H}[\rho](\boldsymbol{r})} + \underbrace{\frac{\delta E_\mathrm{XC}[\rho]}{\delta \rho(\boldsymbol{r})}}_{=: V_\mathrm{XC}[\rho](\boldsymbol{r})} + V_\mathrm{ext}(\boldsymbol{r}) \tag{42}$$

$$\frac{\delta \rho_{\mathbf{\Phi}}(\boldsymbol{r}')}{\delta \phi_i(\boldsymbol{r})} \overset{(31)}{=} \frac{\delta \left(\sum_{j=1}^{N} |\phi_j(\boldsymbol{r}')|^2 \right)}{\delta \phi_i(\boldsymbol{r})} = 2\phi_i(\boldsymbol{r}) \delta(\boldsymbol{r} - \boldsymbol{r}') \tag{43}$$

With this, we obtain

$$\frac{\delta E[\mathbf{\Phi}]}{\delta \phi_i}(\boldsymbol{r}) = 2\hat{T}\phi_i(\boldsymbol{r}) + 2V_\mathrm{eff}[\rho_{\mathbf{\Phi}}](\boldsymbol{r})\phi_i(\boldsymbol{r}) = 2\Big(\hat{T} + V_\mathrm{eff}[\rho_{\mathbf{\Phi}}]\Big)\phi_i(\boldsymbol{r}) \tag{44}$$

Theorem 2.11: The stationary condition $\forall i : \frac{\delta \mathcal{L}}{\delta \phi_i} \overset{!}{=} 0$ yields the **Kohn–Sham equations** with Fock operator $\hat{F}[\rho_{\mathbf{\Phi}}] := \hat{T} + V_\mathrm{eff}[\rho_{\mathbf{\Phi}}]$.

$$\hat{F}[\rho_{\mathbf{\Phi}}]\phi_i(\boldsymbol{r}) = \varepsilon_i \phi_i(\boldsymbol{r}), \qquad i = 1, ..., N \tag{45}$$

Proof: The stationary condition $\forall i : \frac{\delta \mathcal{L}}{\delta \phi_i} \overset{!}{=} 0$ yields

$$0 \overset{!}{=} \frac{\delta \mathcal{L}}{\delta \phi_i}(\boldsymbol{r}) \overset{(39)}{=} \frac{\delta E[\boldsymbol{\Phi}]}{\delta \phi_i}(\boldsymbol{r}) - \sum_{j=1}^{N} \varepsilon_j \frac{\delta\big(\langle \phi_j | \phi_j \rangle - 1\big)}{\delta \phi_i(\boldsymbol{r})} \tag{46}$$

$$\overset{(44)}{=} 2\Big(\hat{T} + V_{\text{eff}}[\rho_{\boldsymbol{\Phi}}]\Big)\phi_i(\boldsymbol{r}) - 2\varepsilon_i \phi_i(\boldsymbol{r}) = 2\Big(\hat{F}[\rho_{\boldsymbol{\Phi}}] - \varepsilon_i\Big)\phi_i(\boldsymbol{r}) \overset{!}{=} 0 \tag{47}$$

Rearranging yields the Kohn–Sham equations (45). ∎

Note that we only enforced the normalization constraint for each orbital, but not the orthogonality constraint between different orbitals in (39). This is because the Kohn–Sham equations (45) are eigenvalue equations for the Fock operator $\hat{F}$. Since $\hat{F}$ is Hermitian, its eigenstates ϕ_i are guaranteed to be orthogonal without explicit enforcement. The eigenvalues ε_i are interpreted as orbital energies.

Self-consistent field (SCF) iterations

The Kohn–Sham equations (45) are nonlinear: the Fock operator $\hat{F}$ depends on the density $\rho_{\boldsymbol{\Phi}}$, which is generated by the orbitals it acts on. Direct diagonalization is therefore not possible. Instead, the equations are solved by a self-consistent field (SCF) method. We start from an initial guess for the orbitals $\boldsymbol{\Phi}^{(0)}$, obtained via the *minimal atomic orbital* (MINAO) method [12, 13] if not otherwise specified. The SCF iteration then proceeds by repeatedly updating the orbitals and the Fock operator until convergence:

1. Construct the Fock operator from the previous orbitals: $\hat{F}^{(k)} := \hat{T} + V_{\text{eff}}[\rho_{\boldsymbol{\Phi}^{(k-1)}}]$.
2. Solve the eigenvalue problem for eigenstates $\boldsymbol{\Phi}^{(k)}$:

$$\hat{F}^{(k)} \phi_i^{(k)} = \varepsilon_i^{(k)} \phi_i^{(k)}, \quad i = 1, ..., N \tag{48}$$

The procedure is continued until the orbitals $\boldsymbol{\Phi}^{(k)}$ converge, meaning they no longer change significantly between successive steps (defined by a threshold) and are thus said to be *self-consistent* with the Fock operator they induce. The final orbitals $\boldsymbol{\Phi}^{(k)}$ at convergence are taken as the solution to the Kohn–Sham equations (45).

Note how in every SCF step, the potential $V_{\text{eff}}^{(k)} := V_{\text{eff}}[\rho_{\boldsymbol{\Phi}^{(k-1)}}]$ does not depend on the current orbitals $\boldsymbol{\Phi}^{(k)}$ that we are solving for, only on the previous orbitals $\boldsymbol{\Phi}^{(k-1)}$. The orbitals $\boldsymbol{\Phi}^{(k)}$ define the ground state of a non-interacting system in an effective one-body potential $V_{\text{eff}}^{(k)}$. This can be seen as follows. Starting with Hamiltonian $\hat{H} := \hat{T} + \hat{V}_{\text{eff}}^{(k)}$ for a non-interacting system ($V_{\text{ee}} = 0$) in the effective potential $V_{\text{eff}}^{(k)}$, we obtain, similar to (23):

$$E[\rho] = \underbrace{\min_{\substack{\psi \in \mathcal{H} \\ \langle \psi | \hat{\rho} | \psi \rangle = \rho}} \Big\langle \psi \Big| \hat{T} + 0 \Big| \psi \Big\rangle}_{\overset{(24)}{=:} T_{\text{S}}[\rho]} + V_{\text{eff}}^{(k)} \quad \Rightarrow \quad E^* = \min_{\rho \in \mathcal{D}} \Big(T_{\text{S}}[\rho] + V_{\text{eff}}^{(k)} \Big) \tag{49}$$

By introducing the orbitals $\mathbf{\Phi}$ that generate the density ρ, we can rewrite this like (34):

$$E[\rho] = \min_{\substack{\mathbf{\Phi} \\ \rho_{\mathbf{\Phi}}=\rho}} \sum_{i=1}^{N} \left\langle \phi_i \middle| \hat{T} \middle| \phi_i \right\rangle + V_{\text{eff}}^{(k)} \tag{50}$$

This leads to the same one-level optimization (37), just with a redefined effective potential:

$$E^* = \min_{\mathbf{\Phi}} \left(\sum_{i=1}^{N} \left\langle \phi_i \middle| \hat{T} \middle| \phi_i \right\rangle + V_{\text{eff}}^{(k)} \right) \tag{51}$$

Finally, varying with respect to the orbitals $\mathbf{\Phi}$ yields the same Kohn–Sham equations (45), solved by the SCF procedure at step k by eigenstates $\mathbf{\Phi}^{(k)}$. Equation (51) then implies that these orbitals minimize the non-interacting kinetic energy $T_{\text{S}}[\rho]$ over all orbital sets $\mathbf{\Phi}$ producing the same density $\rho_{\mathbf{\Phi}^{(k)}}$; otherwise, one could lower (51) further by selecting different orbitals with that same density but smaller kinetic energy. Hence, for every SCF step k, we obtain a **training label for the KEDF** $T_{\text{S}}[\rho]$ by evaluating the kinetic energy of the current orbitals $\mathbf{\Phi}^{(k)}$:

$$T_{\text{S}}[\rho_{\mathbf{\Phi}^{(k)}}] \overset{(32)}{=} \min_{\substack{\mathbf{\Phi} \\ \rho_{\mathbf{\Phi}}=\rho_{\mathbf{\Phi}^{(k)}}}} \sum_{i=1}^{N} \left\langle \phi_i \middle| \hat{T} \middle| \phi_i \right\rangle \overset{(32),\,(45),\,(51)}{=} \sum_{i=1}^{N} \left\langle \phi_i^{(k)} \middle| \hat{T} \middle| \phi_i^{(k)} \right\rangle \tag{52}$$

By the equivalence of (51) and (49) via (32), the density $\rho_{\mathbf{\Phi}^{(k)}}$ also minimizes (49) over all ground-state densities $\rho \in \mathcal{D}$. Therefore it satisfies the stationarity condition of that constrained minimization, with normalization $\int_{\mathbb{R}^3} \rho(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = N$, which is enforced by a Lagrange multiplier $\mu^{(k)}$ (the *chemical potential*). The corresponding Euler–Lagrange equation for this non-interacting system in the effective potential $V_{\text{eff}}^{(k)}$ (51) is given by

$$\frac{\delta T_{\text{S}}[\rho_{\mathbf{\Phi}^{(k)}}]}{\delta \rho(\boldsymbol{r})} + V_{\text{eff}}^{(k)}(\boldsymbol{r}) = \mu^{(k)} \tag{53}$$

This allows us to also retrieve in every SCF step a **training label for the gradient of the KEDF T_{S} with respect to the density**. The chemical potential μ, however, is an undetermined constant, and thus the gradient of the KEDF is only recoverable up to this additive constant. That is, only its projection onto the subspace of density variations that preserve the electron number N is available as a training label. Fortunately, this is sufficient for our purposes.

Atomic-basis formulation

In order to numerically solve the Kohn–Sham equations (45), the orbitals ϕ_i are represented in an atom-centered orbital basis $\{\eta_\alpha(\boldsymbol{r})\}_{\alpha=1}^{B}$:

$$\phi_i(\boldsymbol{r}) = \sum_{\alpha=1}^{B} C_{\alpha i}\eta_\alpha(\boldsymbol{r}) \tag{54}$$

With this, the density ρ can be rewritten as

$$\rho(\boldsymbol{r}) \overset{(31)}{=} \sum_{i=1}^{N}|\phi_i(\boldsymbol{r})|^2 \overset{(54)}{=} \sum_{\alpha,\beta=1}^{B} \underbrace{\sum_{i=1}^{N} C_{\alpha i}C_{\beta i}}_{=:\Gamma_{\alpha\beta}}\ \eta_\alpha(\boldsymbol{r})\,\eta_\beta(\boldsymbol{r}) \tag{55}$$

with density matrix $\boldsymbol{\Gamma} \coloneqq \boldsymbol{C}\boldsymbol{C}^T$. Inserting this into the Kohn–Sham equations (45) yields

$$\sum_{\beta=1}^{B} \hat{F}^{(k)} C_{\beta i}^{(k)}\eta_\beta(\boldsymbol{r}) = \varepsilon_i^{(k)} \sum_{\beta=1}^{B} C_{\beta i}^{(k)}\eta_\beta(\boldsymbol{r}) \tag{56}$$

We project onto the basis functions by multiplying with $\eta_\alpha(\boldsymbol{r})$ and integrating over $\boldsymbol{r}$ in $\mathbb{R}^3$:

$$\sum_{\beta=1}^{B} C_{\beta i}^{(k)} \underbrace{\left\langle \eta_\alpha \middle| \hat{F}^{(k)} \middle| \eta_\beta \right\rangle}_{=:F_{\alpha\beta}^{(k)}} = \varepsilon_i^{(k)} \sum_{\beta=1}^{B} C_{\beta i}^{(k)} \underbrace{\langle \eta_\alpha | \eta_\beta \rangle}_{=:S_{\alpha\beta}} \tag{57}$$

This lets us write the Kohn–Sham equations in matrix form.

Theorem 2.12: The Kohn–Sham equations (45) in the atom-centered orbital basis can be written as a generalized eigenvalue problem with Fock matrix $\boldsymbol{F}^{(k)}$, overlap matrix $\boldsymbol{S}$ and diagonal matrix of orbital energies $\boldsymbol{\varepsilon}^{(k)} = \operatorname{diag}\left(\varepsilon_1^{(k)}, \ldots, \varepsilon_N^{(k)}\right)$ as:

$$\boldsymbol{F}^{(k)}\boldsymbol{C}^{(k)} = \boldsymbol{S}\boldsymbol{C}^{(k)}\boldsymbol{\varepsilon}^{(k)} \tag{58}$$

The orthonormality of the orbitals $\langle \phi_i | \phi_j \rangle = \delta_{ij}$ translates into the condition $\boldsymbol{C}^T\boldsymbol{S}\boldsymbol{C} = \boldsymbol{I}$, thus the eigenvectors (columns of $\boldsymbol{C}$) have to be normalized with respect to the overlap matrix $\boldsymbol{S}$. As before, orthogonality is already guaranteed by the Hermiticity of the Fock operator, so we only need to enforce normalization.

In practice, we use the 6-31G(2df,p) basis set [14] with Gaussian-type orbitals (GTOs) as basis functions η_α. Calculations are performed with the PySCF package [15–17]. It should be mentioned that the mere fact of expanding the orbitals in a finite basis introduces an additional approximation (beyond the approximation of the exchange-correlation functional

$E_{\mathrm{XC}}[\rho]$), since the eigenvalue problem is formulated in an infinite-dimensional Hilbert space, while the basis expansion restricts us to a finite-dimensional subspace. However, with a sufficiently large basis set, this approximation error can be reduced.

Density basis and density fitting

In the density representation of the atom-centered orbital basis (55), the coefficient matrix $\boldsymbol{C}$ is of size $B \times N$, thus we have a $\mathcal{O}(NB) = \mathcal{O}(N^2)$ scaling (B scales linearly with N). To mitigate this quadratic scaling in system size N, we employ a **linear combination of atomic basis function (LCAB)** with basis functions $\left\{\omega_\mu(\boldsymbol{r})\right\}_{\mu=1}^{M}$ and coefficients $\boldsymbol{p}$ of size M that scale linearly with system size. The density is expanded as

$$\rho(\boldsymbol{r}) = \sum_{\mu=1}^{M} p_\mu \omega_\mu(\boldsymbol{r}) \tag{59}$$

The Kohn–Sham density from (55) lives in the paired orbital basis $\left\{\eta_\alpha \eta_\beta\right\}$, while the LCAB formulation is in density basis $\left\{\omega_\mu\right\}$. **Density fitting** bridges the two representations by finding coefficients $\boldsymbol{p}$ that map well to the density matrix $\boldsymbol{\Gamma} \coloneqq \boldsymbol{C}\boldsymbol{C}^T$ (that is, to the corresponding orbital coefficients $\boldsymbol{C}$):

$$\rho_{\boldsymbol{C}}(\boldsymbol{r}) \coloneqq \sum_{\alpha,\beta} \Gamma_{\alpha\beta} \eta_\alpha(\boldsymbol{r}) \eta_\beta(\boldsymbol{r}) \quad \overset{!}{\approx} \quad \sum_{\mu} p_\mu \omega_\mu(\boldsymbol{r}) =: \rho_{\boldsymbol{p}}(\boldsymbol{r}) \tag{60}$$

We follow [4] and fit $\boldsymbol{p}$ in a combined least squares problem that minimizes the Hartree energy of the residual density $E_{\mathrm{H}}\left[\rho_{\boldsymbol{p}} - \rho_{\boldsymbol{C}}\right]$ and the squared error in the external energy $\left(E_{\mathrm{ext}}\left[\rho_{\boldsymbol{p}}\right] - E_{\mathrm{ext}}[\rho_{\boldsymbol{C}}]\right)^2$. An even-tempered basis set [18] with $\beta = 2.5$ is used to calculate $\left\{\omega_\mu\right\}$ from the orbital basis $\{\eta_\alpha\}$.

3 Structures25

Structures25 [5] is a machine-learned orbital-free DFT pipeline that achieves convergent variational density optimization on organic molecules while maintaining chemical accuracy relative to the Kohn–Sham reference. It builds on M-OFDFT [4] with key improvements in training on more diverse data generated from perturbed effective potentials, and architectural changes that replace the fully connected attention of M-OFDFT with local message passing within a radial cutoff, improving scalability while maintaining accuracy.

Our code and all experiments in this thesis build directly on Structures25. We summarize its architecture, training targets, density optimization procedure, and the used dataset, referring to [5] for full technical details.

3.1 Density Representation

Both M-OFDFT and Structures25 represent the electron density via the LCAB ansatz (59):

$$\rho(\boldsymbol{r}) = \sum_{\mu} p_\mu \omega_\mu(\boldsymbol{r}) \tag{61}$$

where $\{\omega_\mu\}$ are atom-centered Gaussian basis functions (even-tempered with $\beta = 2.5$) and $\boldsymbol{p}$ is the coefficient vector. This representation is much more compact than a real-space grid: typically $\sim 10N$ coefficients suffice for a molecule with N atoms, compared to a grid with 1000 times more points (needed to capture non-local features of the density). The coefficients are obtained by density fitting of the KS density at each SCF iteration see (60).

3.2 Training Data

When sampling from an unperturbed KS-DFT SCF procedure to generate training data, most densities lie very close to the ground state, as the SCF iterations converge rapidly. A model trained on such data has no incentive to shape the energy landscape correctly away from the ground state, as it only needs to be accurate in a tiny neighborhood around it. This leads to non-convex energy surfaces with saddle points, causing gradient-based density optimization to diverge into unphysical densities, as observed for M-OFDFT [5].

Structures25 therefore augments each SCF trajectory by adding random perturbations to the effective potential V_{eff} at each iteration step. The perturbed potential drives the SCF solution away from equilibrium and produces training densities spread more broadly around the ground state. Training on these enriched data directly yields a well-formed energy functional with a genuine minimum, enabling convergent variational density optimization. The perturbation strategy is described in detail in Section 5.1.

3.3 Training Target

The total electronic energy decomposes as (23):

$$E[\rho] = T_{\text{S}}[\rho] + E_{\text{H}}[\rho] + E_{\text{XC}}[\rho] + E_{\text{ext}}[\rho] \tag{62}$$

M-OFDFT predicts the difference $T_{\text{S}} - T_{\text{APBEK}}$ relative to the APBEK functional [19] via delta learning. This requires evaluating E_{XC} on a real-space grid at inference time. Structures25 instead learns the combined functional $E_{\text{TXC}} \coloneqq T_{\text{S}} + E_{\text{XC}}$, which aggregates all energy contributions not analytically accessible from $\boldsymbol{p}$ alone. Since E_{H} and E_{ext} can be evaluated directly from $\boldsymbol{p}$ and $\mathcal{M}$, the entire density optimization loop requires no quadrature grid.

Beyond energy values, the model is also trained on gradient labels $\nabla_{\boldsymbol{p}} E_{\mathrm{TXC}}$, that is, the functional derivative of E_{TXC} projected onto the density basis (related to (53)). Accurately modelling also the gradient of the energy landscape is necessary to enable Density Optimization as described in Section 3.5.

3.4 Architecture

The model maps density coefficients $\boldsymbol{p}$ and molecular geometry $\mathcal{M} := \{(\boldsymbol{R}_a, Z_a)\}_{a=1}^{A}$, with $\boldsymbol{R}_a$ and Z_a the position and atomic number of atom a, to the scalar prediction E_{TXC}. For brevity, we keep the dependence on the geometry $\mathcal{M}$ implicit in what follows. The gradient label $\nabla_{\boldsymbol{p}} E_{\mathrm{TXC}}$ is obtained by automatic differentiation of the predicted energy with respect to the input coefficients. Before the GNN backbone, the coefficients undergo the preprocessing pipeline from M-OFDFT [4]: natural reparametrization into an orthonormal density basis, followed by dimension-wise rescaling and an atomic reference correction. The natural reparametrization is a global operation over all N atoms and has $\mathcal{O}(N^3)$ complexity.

Graphormer
The primary backbone is based on the Graphormer [20, 21], which extends the standard transformer to graph-structured molecular data. Its core mechanism is self-attention: each atom's representation is updated as a weighted average over neighboring atoms, with attention weights derived from pairwise features and distances. This corresponds to a fully-connected graph with A atoms as nodes.

The original Graphormer processes graphs using scalar node features and graph-structural encodings and does not enforce $\mathrm{SE}(3)$-equivariance. M-OFDFT introduces rotational invariance by constructing an equivariant local coordinate frame for each atom from its neighborhood geometry and rotating the density coefficients into a canonical orientation prior to message passing. The local frames themselves are not propagated through the network; after canonicalization, all features are scalar and the attention layers operate in a frame-independent representation.

Structures25 extends the local-frame construction of M-OFDFT by *tensorial message passing* [22]: when sending a message from atom a to atom b, features are rotated from a's local frame through the global frame into b's local frame, enabling directional, higher-order geometric information to flow between atoms, while retaining a Graphormer-style attention backbone.

A second key modification is replacing the fully connected attention graph with *local message passing* within a radial cutoff of $6\, r_{\mathrm{Bohr}}$. The fully connected graph of M-OFDFT incurs $\mathcal{O}(N^2)$ complexity per layer and eventually becomes prohibitive for large systems; local message passing reduces this to $\mathcal{O}(N)$, consistent with the nearsightedness principle of electronic matter [23]: for systems with a bandgap, the density at any point is determined predominantly by the nearby potential.

Equiformer

The Structures25 codebase also supports the Equiformer architecture [24] as alternative backbone. EquiformerV2 [25] achieves SE(3)-equivariance by a fundamentally different strategy: stacking equivariant layers, each using the *tensor product* as its core bilinear operation to mix features of different irreducible representations, while each linear layer only operates on features of the same representation.

Structures25 [5] finds comparable accuracy between the Graphormer and Equiformer families, with the Graphormer variant currently offering the better cost–performance trade-off.

3.5 Density Optimization (Denop)

At inference time, the ground state density for a given geometry $\mathcal{M}$ is obtained by minimizing the total energy functional $E_{\text{tot}}[\rho]$ with respect to density $\rho(\boldsymbol{r})$. This process is referred to as *density optimization* (Denop). The minimum is found via gradient descent with momentum[1]. Figure Figure 3.2 visualizes this iterative procedure.

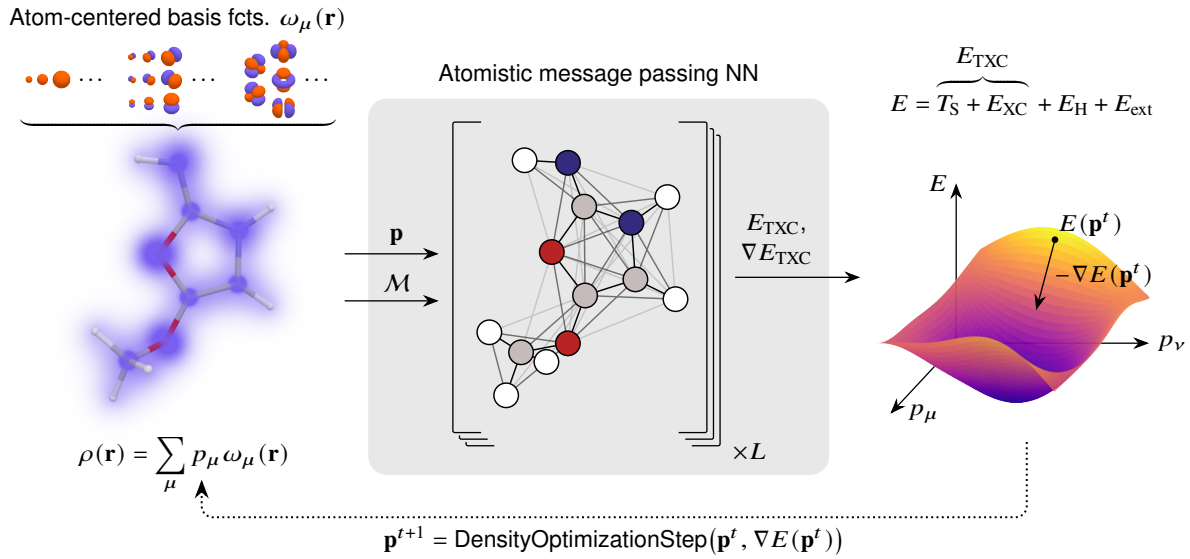


Figure 3.2: Schematic overview of Density Optimization in Structures25. Starting from an initial guess of density coefficient vector $\boldsymbol{p}$, and given the molecular geometry $\mathcal{M}$, the network predicts the energy functional $E_{\text{TXC}}(\boldsymbol{p}, \mathcal{M})$. Its gradient with respect to $\boldsymbol{p}$, where the latter is obtained by automatic differentiation. The coefficients $\boldsymbol{p}$ are iteratively refined by following the energy gradient of the learned functional until convergence is reached. Plot taken without modification from Figure 1 (B) in [5].

[1]The PyTorch [26] implementation is used with a learning rate of 0.003 and momentum of 0.9 for QM9, see also Section S.6 in [5].

During optimization, the update step is projected onto the hyperplane defined by the electron number constraint $\int \rho(\boldsymbol{r})d\boldsymbol{r} = N$. This ensures that the total number of electrons remains constant throughout optimization. The energy gradient is obtained by automatic differentiation of the predicted energy with respect to the input coefficients, and the optimization proceeds until convergence, defined as when the norm of the energy gradient falls below 10^{-4} Ha. Convergence requires the learned functional to have a genuine energy minimum at or near the ground state density, which is achieved by training on perturbed data as described in Section 3.2 and Section 5.1.

3.6 Dataset: QM9

Structures25 is (among others) trained and evaluated on QM9 [6], a dataset of 133 885 small organic molecules $\mathrm{C}_c\mathrm{H}_h\mathrm{N}_n\mathrm{O}_o\mathrm{F}_f$ with $c + n + o + f \leq 9$ (at most 9 non-hydrogen atoms). Reference labels are computed at the PBE/6-31G(2df,p) level with PySCF [15, 16]. From each SCF iteration, a training tuple $\left(\boldsymbol{p}, E_{\mathrm{TXC}}, \nabla_{\boldsymbol{p}} E_{\mathrm{TXC}}\right)$ is extracted. The dataset is randomly split in an 80:10:10 ratio for training, validation and testing sets.

On the test set containing 13 389 molecules, Structures25 achieves 100% convergence in density optimization (gradient norms around 10^{-13}) and a mean absolute energy error of 0.64 mHa, below the "chemical accuracy" threshold of 1.6 mHa (1 kcal/mol). The mean L^2 density error is 0.0140; notably, it is smaller than the approximation error from density fitting where the KS density is projected onto the orbital-free basis, see also Fig. S.2 in [5].

3.7 Extrapolation to Larger Molecules

A primary motivation for OF-DFT is its favorable cost scaling. To assess transferability beyond the training regime, Structures25 is evaluated on QMugs [27], a dataset of drug-like molecules substantially larger than QM9. Despite the local radial cutoff, the per-atom energy error does not grow with molecule size across the QMugs test set, and density optimization converges on all 850 test molecules. The three largest energy outliers all contain a trifluoromethoxy group found in only a single training molecule, illustrating that sufficient coverage of chemical substructures remains necessary. Here, Structures25 achieves roughly an order of magnitude speedup over Kohn–Sham DFT on identical hardware, reflecting the cheaper per-iteration cost (no matrix diagonalization required to solve the SCF equations during inference) [5].

4 Direct Ground State Prediction (DGSP)

The previous chapters described how a learned KEDF model is used together with Density Optimization (Denop) to find the ground state. This variational approach is physically principled but computationally expensive at inference time, since every test molecule requires an iterative optimization loop. An alternative is to predict the ground state density directly from the molecular geometry in a single forward pass, thereby bypassing optimization entirely. This approach is called *Direct Ground State Prediction* (DGSP). In this chapter, we discuss related work on DGSP, describe our implementation, and present results.

4.1 Related Work

Several recent methods have explored the idea of predicting the ground state electron density ρ^* directly from the molecular geometry $\mathcal{M}$, without running a full SCF or Denop process. They differ mainly in how the density is represented and in whether the model predicts the density itself or only a correction to an existing initialization.

ELECTRA [28] predicts the ground state density directly as a sum of floating Gaussian orbitals. Rather than restricting the representation to atom-centered basis functions, it predicts Gaussian weights, positions, and covariance matrices, which gives the model freedom to place its basis functions anywhere in space and thus capture the density also in inter-atomic regions without manually designing virtual nodes or floating orbitals.

BOA [29] also predicts the ground state density directly, but employs an atom-centered basis representation. It makes use of a quadratic expansion, inspired by the density-matrix formalism of KS-DFT, instead of predicting a simple linear coefficient vector. Products of basis functions centered on different atoms also cover the inter-atomic regions, even without explicit floating orbitals. BOA employs a novel message passing scheme where messages are transformed from the basis of the sending node to that of the receiving node via the overlap matrix. This provides a strong geometry-aware bias.

M-OFDFT [4] operates on density coefficients $\boldsymbol{p}$ under an atomic basis, and minimizes the electronic energy through gradient descent on $\boldsymbol{p}$ (Denop). Within that framework, a practical difficulty is that the standard MINAO initialization [15] is not generated by the same mechanism as the SCF densities seen during training and can therefore lie off the training-data manifold. To address this, the M-OFDFT paper proposes *ProjMINAO*, where an auxiliary network $\Delta\boldsymbol{p}(\boldsymbol{p}_{\text{init}}, \mathcal{M})$ learns to predict a correction to the initial MINAO coefficients in coefficient space, in order to project the initial guess toward the ground state and on the training manifold. The corrected coefficients are then used as the starting point for Denop.

4.2 Method

Overview

In our standard pipeline, the GNN (Graphormer or Equiformer) takes as input the molecular geometry $\mathcal{M}$ and the current density coefficients $\boldsymbol{p}$, and predicts the KEDF energy T_{S} and its gradient $\nabla_{\boldsymbol{p}} T_{\text{S}}$. The gradient is then used for Denop.

For DGSP, we repurpose the same GNN backbone and add a dedicated MLP readout head[2] that predicts the coefficient correction $\Delta\hat{\boldsymbol{p}}$ directly. Concretely, the GNN processes the molecular graph and produces node features; these are passed through the *initial guess delta module*, an MLP that maps atom-level features to per-basis-function coefficient corrections.

[2] This readout head was already present in the codebase, but not used by default.

We define the difference between initial coefficients $\boldsymbol{p}_{\text{init}}$ and the label ground state coefficients $\boldsymbol{p}^*$ as $\Delta\boldsymbol{p}^*$, while the difference between the predicted coefficients $\hat{\boldsymbol{p}}$ to the initial coefficients is $\Delta\hat{\boldsymbol{p}}$. Figure 4.3 illustrates the geometric relationship between these quantities.

$$\Delta\boldsymbol{p}^* := \boldsymbol{p}^* - \boldsymbol{p}_{\text{init}} \tag{63}$$

$$\Delta\hat{\boldsymbol{p}} := \hat{\boldsymbol{p}} - \boldsymbol{p}_{\text{init}} \tag{64}$$

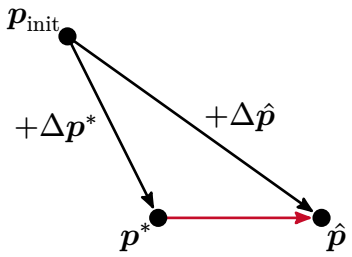


Figure 4.3: Schematic relation between initial coefficients $\boldsymbol{p}_{\text{init}}$, label ground state coefficients $\boldsymbol{p}^*$, and predicted ground state coefficients $\hat{\boldsymbol{p}}$. Red: difference vector between predicted and label ground state coefficients.

For the sample molecule $H_9C_6N_3$, Figure 4.4 visualizes the density difference $\Delta\rho^*(\boldsymbol{r})$ between the initial MINAO guess and the label ground state. The blue and red regions indicate where the initial guess overestimates or underestimates the label density, respectively. The DGSP model's task is to learn to predict a correction that moves from the initial guess toward the ground state in a single step, effectively "filling in" the red regions and "trimming down" the blue regions to match the label density.

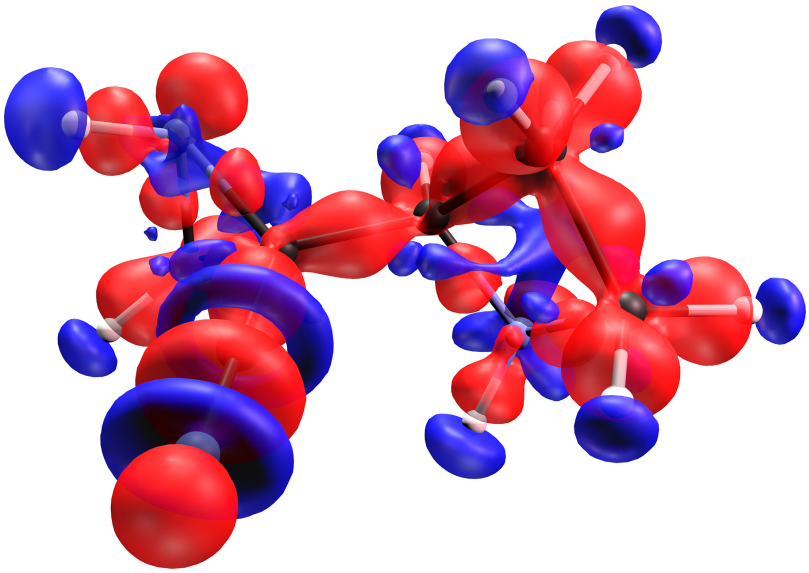


Figure 4.4: Isosurface of the density difference $\Delta\rho^*(\boldsymbol{r}) := \rho_{\text{init}}(\boldsymbol{r}) - \rho^*(\boldsymbol{r})$ between the initial MINAO guess ρ^{init} and the label ground state ρ^* (last SCF iteration) for molecule $H_9C_6N_3$. Blue: positive difference. Red: negative difference.

Model: MLP Head

The readout head is a simple MLP mapping from the GNN's node features (dimension 768) to the per-basis-function coefficient corrections $\Delta\hat{\boldsymbol{p}}$. It consists of two hidden linear layers with GeLU activation in-between, and no dropout. In total, the MLP has 957 K trainable parameters, with 590 K for the first linear layer and 366 K for the second linear layer.

Since DGSP does not require energy gradients, we use a simplified network variant that does not compute $\nabla_{\boldsymbol{p}} T_{\mathrm{S}}$. This avoids the computational overhead of backpropagation through the energy prediction during the forward pass. The energy and gradient loss weights are set to zero, and only the coefficient loss is active.

Loss Function: Natural L^2 Loss

Figure 4.3 might suggest defining the loss as L^1 or L^2 distance between the predicted and label coefficient corrections:

$$\mathcal{L}' \coloneqq \left\|\Delta\hat{\boldsymbol{p}} - \Delta\boldsymbol{p}^*\right\|_1 \quad \text{or} \quad \mathcal{L}'' \coloneqq \left\|\Delta\hat{\boldsymbol{p}} - \Delta\boldsymbol{p}^*\right\|_2 \tag{65}$$

The coefficient-space L^1 loss $\mathcal{L}'$ is problematic because its sparsity bias in the weights encourages the model to focus on a few basis functions while ignoring others, which can lead to suboptimal density predictions. The L^2 loss, on the other hand, distributes the error more evenly across all basis functions. Furthermore, it penalizes larger errors more strongly, which can help the model to learn more accurate corrections.

But a naive coefficient-space L^2 loss $\mathcal{L}''$ is still suboptimal. It treats all coefficient directions as equally important, although basis functions have different spatial extent and nontrivial overlap. Hence, a small Euclidean coefficient error can still produce a large real-space density error and vice versa. Instead, we optimize the physically relevant quantity: the L^2 error of the density difference $\Delta\rho(\boldsymbol{r})$. This yields the natural L^2 loss, which incorporates the basis overlap matrix and is invariant to the chosen coefficient representation.

$$\mathcal{L} \coloneqq \left\|\Delta\hat{\rho} - \Delta\rho^*\right\|_2 \tag{66}$$

This loss directly measures the error in the predicted density difference, rather than just the coefficient difference, and thus provides a more meaningful training signal for DGSP. With Theorem 4.1, we obtain two expressions that differ by the presence of the overlap matrix $\boldsymbol{W}$.

$$\mathcal{L}'' = \sqrt{(\Delta\hat{\boldsymbol{p}} - \Delta\boldsymbol{p}^*)^T (\Delta\hat{\boldsymbol{p}} - \Delta\boldsymbol{p}^*)} \tag{67}$$

$$\mathcal{L} = \sqrt{(\Delta\hat{\boldsymbol{p}} - \Delta\boldsymbol{p}^*)^T \boldsymbol{W} (\Delta\hat{\boldsymbol{p}} - \Delta\boldsymbol{p}^*)} \tag{68}$$

Theorem 4.1 (L^2-Norm of density)**:** The density in the LCAB ansatz (59) has L^2 norm:

$$\rho(\boldsymbol{r}) = \sum_{\mu} p_\mu \omega_\mu(\boldsymbol{r}) \quad \Rightarrow \quad \|\rho(\boldsymbol{r})\|_2 = \sqrt{\boldsymbol{p}^T \boldsymbol{W} \boldsymbol{p}} \tag{69}$$

where $W_{\mu\nu} \coloneqq \int_{\mathbb{R}^3} \omega_\mu(\boldsymbol{r})\, \omega_\nu(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}$ is the overlap matrix of the density basis.

Proof: We plug the density expansion into the definition of the L^2 norm.

$$\begin{aligned}
\Big(\|\rho(\boldsymbol{r})\|_2\Big)^2 &= \int_{\mathbb{R}^3} |\rho(\boldsymbol{r})|^2 \, \mathrm{d}\boldsymbol{r} && (70)\\
&= \int_{\mathbb{R}^3} \left(\sum_{\mu} p_\mu \omega_\mu(\boldsymbol{r})\right)\left(\sum_{\nu} p_\nu \omega_\nu(\boldsymbol{r})\right) \mathrm{d}\boldsymbol{r} && (71)\\
&= \sum_{\mu,\nu} p_\mu p_\nu \underbrace{\int_{\mathbb{R}^3} \omega_\mu(\boldsymbol{r})\ \omega_\nu(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}_{W_{\mu\nu}} && (72)\\
&= \sum_{\mu,\nu} p_\mu W_{\mu\nu} p_\nu = \boldsymbol{p}^T \boldsymbol{W} \boldsymbol{p} && (73)
\end{aligned}$$

■

Overlap Matrix $\boldsymbol{W}$

Notice how the natural L^2 loss $\mathcal{L}$ (68) incorporates the overlap matrix $\boldsymbol{W}$, which captures the geometry of the basis functions. This means that the loss is sensitive to how coefficient differences translate into real-space density errors, rather than treating all coefficient errors as equally bad regardless of their spatial impact.

The overlap matrix $\boldsymbol{W}$ depends on the molecular geometry and the density basis set. We precompute it for each sample using PySCF [15]. During collation, these per-sample overlap matrices are zero-padded to a common size and stacked into a batched tensor, such that (68) can be evaluated efficiently in a batched manner.

Natural Reparameterization

Natural reparameterization (NatRep), as introduced in M-OFDFT [4], constructs a new matrix $\boldsymbol{M}$ from the overlap matrix such that $\boldsymbol{W} = \boldsymbol{M}\boldsymbol{M}^T$, and then transforms the coefficients via $\tilde{\boldsymbol{p}} = \boldsymbol{M}^T \boldsymbol{p}$. In this transformed representation, the Euclidean norm of $\Delta\tilde{\boldsymbol{p}}$ matches the physical L^2 norm of the density difference:

$$\|\Delta\tilde{\boldsymbol{p}}\|_2 = \sqrt{\Delta\tilde{\boldsymbol{p}}^T \Delta\tilde{\boldsymbol{p}}} = \sqrt{\Delta\boldsymbol{p}^T \boldsymbol{M}\boldsymbol{M}^T \Delta\boldsymbol{p}} = \sqrt{\Delta\boldsymbol{p}^T \boldsymbol{W} \Delta\boldsymbol{p}} = \|\Delta\rho\|_2 \tag{74}$$

In the main DGSP experiments described here, NatRep is not applied in preprocessing. For the natural L^2 setup, we instead add the overlap matrix in order to compute the loss in the original coefficient space.

Local and Global Frames

In the *Graphormer* variant, the cached coefficients are represented in local frames so that the input becomes $\mathrm{SE}(3)$-invariant. The overlap matrix $\boldsymbol{W}$, however, is stored in the global frame. It would be computationally costly to transform the collated overlap matrix to the local frame. Instead, our implementation of the natural L^2 loss first maps both the predicted and target coefficient differences back to the global frame and only then evaluates the L^2 loss with $\boldsymbol{W}$. Since the local-frames transformation is orthogonal, its inverse is simply the transpose. For the *Equiformer* variant, this step is not necessary because no local frames are used.

Data

In the standard Structures25 pipeline, each molecule contributes multiple SCF iterations (one coefficient–gradient pair for each iteration). For DGSP, we restrict the dataset to only the *initial guess* (0^{th} SCF iteration), since the network should learn to map the initial MINAO density to the ground state in a single step.

4.3 Results

We train two DGSP models on the QM9 dataset with perturbed Fock data generated in [7] and used in [5]:

- *Graphormer*: with local frames; natural L^2 loss
- *Equiformer*: (EquiformerV2) without local frames; natural L^2 loss

The predicted density correction $\Delta\hat{\boldsymbol{p}}$ gives an approximation of the ground state density in a single forward pass (see Figure 4.3). In Figure 4.5, we evaluate the L^2 density error $\|\hat{\rho} - \rho^*\|_2$ for the DGSP models and compare it against a Density Optimization run using the Graphormer model trained with the default Structures25 settings. The evaluation is performed on our QM9 test set containing 13 389 molecules.

Both the Graphormer and Equiformer DGSP models outperform the baseline Denop approach and have less variance. The best results are achieved by the Equiformer, which attains a mean L^2 density error of 0.0075, compared to 0.0101 for the Graphormer and 0.0167 for the Denop baseline. This shows that DGSP can get very close to the ground state density in a single step, without any optimization iterations.

DGSP also has a clear inference-time advantage: while the baseline requires 194 iterations on average, DGSP needs only a single forward pass. We parallelize DGSP evaluation, such that with 32 workers the full test set can be processed in approximately 25 minutes per DGSP model on an NVIDIA A100 GPU. By contrast, the Denop baseline takes approximately 36.57 s per molecule, which amounts to approximately 4 h for the full test set with 32 workers.

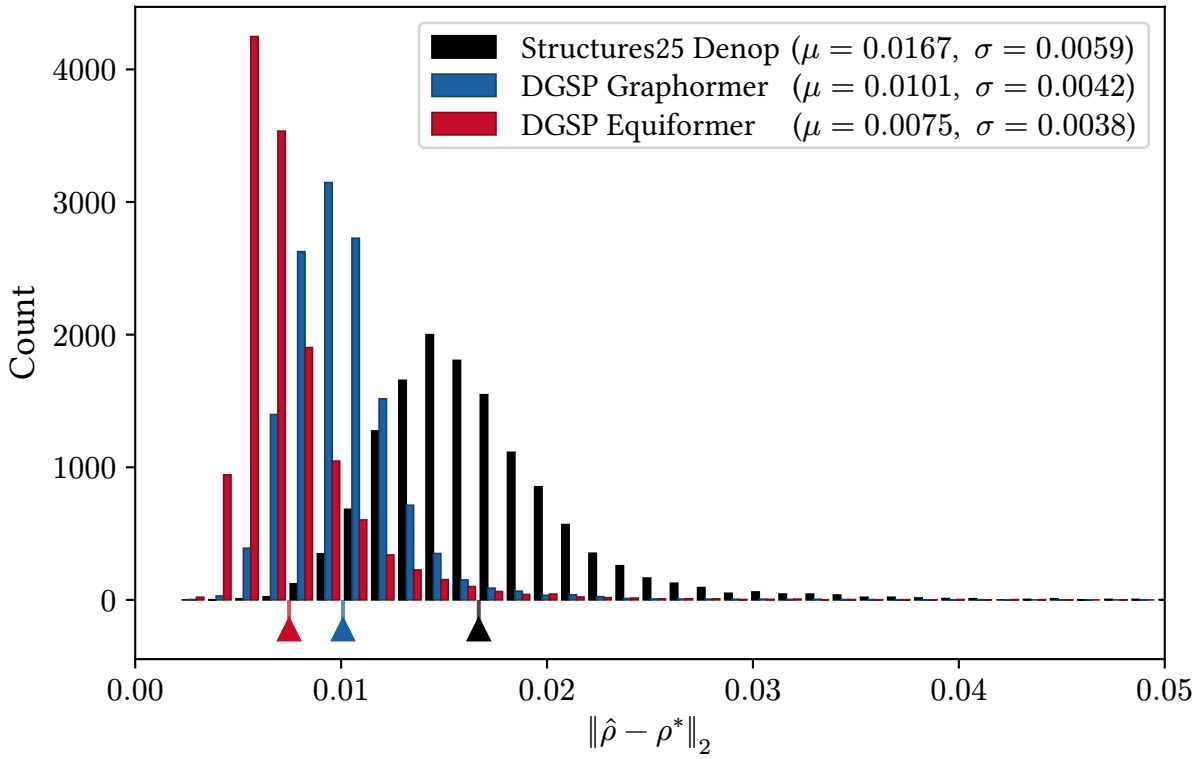


Figure 4.5: Comparison of L^2 density differences to the ground state for the baseline (Structures25 Denop), the DGSP Graphormer, and the DGSP Equiformer on the test set containing 13 389 molecules. The equiformer performs best with a mean L^2 density error of 0.0075. Plot cut off at 0.05.

However, this improved performance and inference efficiency comes at the cost of significantly longer training times. While the Baseline Graphormer model trained with the default Structures25 settings took ≈ 3 days to train on two NVIDIA A100 GPUs, both DGSP models took ≈ 23 days. In principle, the models should train faster since no backpropagation is needed to obtain gradients. The bottleneck is likely due to the computational overhead of the natural L^2 loss, which requires matrix multiplications with the overlap matrix $\boldsymbol{W}$ during training. Future work on improving training efficiency would be important to enable faster iteration on DGSP experiments and to explore larger model variants.

Furthermore, the current DGSP models only report the coefficient loss. Combining the DGSP coefficient loss with the variational energy and gradient losses in a multi-task setup could yield a model that both predicts good initial densities and provides accurate KEDF gradients for Denop. This is attempted in Chapter 5. Furthermore, future work should explore model generalization to larger datasets and more complex molecules beyond QM9, for example using the QMugs dataset [27]. If DGSP can generalize well to larger and more complex molecules, it could become a powerful tool for fast density prediction in a wide range of chemical applications.

5 Refine-Model

The quality and variety of training densities strongly influence the learned energy landscape and, therefore, whether density optimization converges to physically meaningful ground states. We begin by outlining the Structures25 strategy of perturbing the effective potential during the Kohn–Sham SCF iterations to produce training data, and then introduce an alternative that fully separates perturbation-based data generation from the SCF trajectory. Finally, we evaluate models trained on these datasets and report preliminary results for combining DGSP (Chapter 4) with a Refine-Model in a Prime–Refine workflow.

5.1 Data Generation with Perturbations

As shown in Section 2.5, each SCF iteration k produces orbitals $\mathbf{\Phi}^{(k)}$ that are the ground state of a non-interacting system in the potential $V_{\text{eff}}^{(k)}$. This yields training labels for both the KEDF value $T_{\text{S}}[\rho_{\mathbf{\Phi}^{(k)}}]$ via (52) and the KEDF gradient via the Euler–Lagrange equation (53). In principle, every SCF iteration thus provides one label pair $\left(T_{\text{S}}, \nabla_{\boldsymbol{p}} T_{\text{S}}\right)$.

However, the densities encountered along an unperturbed SCF trajectory lack diversity: they rapidly converge to the ground state, leaving only a small fraction of the density space covered by the training data. Therefore, we follow the approach pioneered in [5, 7, 8] and apply perturbations during the SCF procedure to generate a richer variety of labels. This strategy also aims at reducing convergence difficulties noted in the M-OFDFT paper [4] (Section B.5.3 Stopping Criterion). Models trained on perturbed data have been shown to outperform those trained on unperturbed SCF trajectories [7].

We follow the presentation in [7] to describe the perturbation method and derive the resulting gradient labels. We then propose a different perturbation strategy that addresses some practical limitations of the SCF-based approach.

External potential perturbation
In SCF iteration k, the effective potential is perturbed as

$$V_{\text{eff}}^{(k)}(\boldsymbol{r}) \;\rightarrow\; V_{\text{eff}}^{(k)}(\boldsymbol{r}) + \delta V^{(k)}(\boldsymbol{r}) \tag{75}$$

where $\delta V^{(k)}$ is a perturbation potential, constructed as a linear combination of density basis function $\left\{\omega_\mu\right\}_{\mu=1}^{M}$ (59):

$$\delta V^{(k)}(\boldsymbol{r}) = \sum_{\mu=1}^{M} \delta V_\mu^{(k)} \omega_\mu(\boldsymbol{r}) \tag{76}$$

The coefficients $\delta V_\mu^{(k)}$ are drawn from a suitable distribution. Expressing this perturbation in the orbital basis $\{\eta_\alpha\}_{\alpha=1}^{B}$ (54) amounts to a perturbed Fock matrix $\boldsymbol{F}^{(k)}$ (57) with entries

$$F_{\alpha\beta}^{(k)} \;\rightarrow\; F_{\alpha\beta}^{(k)} + \left\langle \eta_\alpha \middle| \delta V^{(k)} \middle| \eta_\beta \right\rangle \tag{77}$$

$$= F_{\alpha\beta}^{(k)} + \int \delta V^{(k)}(\boldsymbol{r}) \eta_\alpha(\boldsymbol{r}) \eta_\beta(\boldsymbol{r}) \,\mathrm{d}\boldsymbol{r} \tag{78}$$

$$\overset{(76)}{=} F_{\alpha\beta}^{(k)} + \int \left(\sum_{\mu=1}^{M} \delta V_\mu^{(k)} \omega_\mu(\boldsymbol{r}) \right) \eta_\alpha(\boldsymbol{r}) \eta_\beta(\boldsymbol{r}) \,\mathrm{d}\boldsymbol{r} \tag{79}$$

$$= F_{\alpha\beta}^{(k)} + \sum_{\mu=1}^{M} \delta V_\mu^{(k)} \underbrace{\int \omega_\mu(\boldsymbol{r}) \eta_\alpha(\boldsymbol{r}) \eta_\beta(\boldsymbol{r}) \,\mathrm{d}\boldsymbol{r}}_{:= W_{\mu\alpha\beta}} = F_{\alpha\beta}^{(k)} + \sum_{\mu=1}^{M} \delta V_\mu^{(k)} W_{\mu\alpha\beta} \tag{80}$$

where $W_{\mu\alpha\beta}$ is the three-center overlap tensor between the density basis and the orbital basis: $W_{\mu\alpha\beta} := \int \omega_\mu(\boldsymbol{r})\, \eta_\alpha(\boldsymbol{r})\, \eta_\beta(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}$.

Sampling the perturbation in the density basis rather than perturbing the Fock matrix directly is essential for physical consistency. Any perturbation of the form (76) corresponds to a potential that uniquely determines the resulting ground-state density (see Theorem 2.4), which in turn makes the gradient label (derived next) consistent with that density.

Perturbing the Fock matrix entries $F_{\alpha\beta}$ directly would not guarantee this. For instance, if two products of orbital basis functions coincide, $\eta_\alpha\eta_\beta = \eta_\gamma\eta_\delta$, then for any one-body potential $\hat{V}$ we must have $\left\langle \eta_\alpha \middle| \hat{V} \middle| \eta_\beta \right\rangle = \left\langle \eta_\gamma \middle| \hat{V} \middle| \eta_\delta \right\rangle$. A random, unconstrained matrix perturbation can violate this constraint, yielding a Fock matrix that does not correspond to any physical potential. The resulting density–gradient pair would then be inconsistent.

Gradient label under perturbation

The orbitals $\boldsymbol{\Phi}^{(k)}$ obtained from the perturbed Fock matrix are the ground state of a non-interacting system in the potential $V_{\mathrm{eff}}^{(k)} + \delta V^{(k)}$. By the same Euler–Lagrange argument that led to (53), we find:

$$\frac{\delta T_{\mathrm{S}}[\rho_{\boldsymbol{\Phi}^{(k)}}]}{\delta\rho(\boldsymbol{r})} = -\Big(V_{\mathrm{eff}}^{(k)}(\boldsymbol{r}) + \delta V^{(k)}(\boldsymbol{r})\Big) + \mu^{(k)} \tag{81}$$

where $\mu^{(k)}$ is the chemical potential enforcing $\int \rho\, \mathrm{d}\boldsymbol{r} = N$. Projecting onto the density basis (multiplying by ω_μ and integrating) gives the gradient with respect to density coefficients $\boldsymbol{p}$:

$$\frac{\partial T_{\mathrm{S}}}{\partial p_\mu} = \int \frac{\delta T_{\mathrm{S}}}{\delta\rho(\boldsymbol{r})} \frac{\mathrm{d}\rho(\boldsymbol{r})}{\mathrm{d}p_\mu}\, \mathrm{d}\boldsymbol{r} = \int \frac{\delta T_{\mathrm{S}}}{\delta\rho(\boldsymbol{r})} \omega_\mu(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \tag{82}$$

$$\overset{(81)}{=} \underbrace{\int \Big(\mu^{(k)} - V_{\mathrm{eff}}^{(k)}(\boldsymbol{r})\Big)\omega_\mu(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}_{\text{gradient without perturbation}} - \underbrace{\int \delta V^{(k)}(\boldsymbol{r})\omega_\mu(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}}_{\text{perturbation contribution } \delta v_\mu^{(k)}} \tag{83}$$

The first term is the gradient label that would be obtained without perturbation, while the second term captures the contribution of the perturbation:

$$\delta v_\mu^{(k)} := \int \delta V^{(k)}(\boldsymbol{r})\omega_\mu(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} \overset{(76)}{=} \sum_{\nu=1}^{M} \delta V_\nu^{(k)} \int \omega_\mu(\boldsymbol{r})\omega_\nu(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} = \sum_{\nu=1}^{M} \delta V_\nu^{(k)} W_{\mu\nu} \tag{84}$$

where $W_{\mu\nu} = \int \omega_\mu\omega_\nu\, \mathrm{d}\boldsymbol{r}$ is the density-basis overlap matrix. Again, the chemical potential $\mu^{(k)}$ enforcing the density normalization is eliminated by a projection on the tangent space of normalized densities as originally described in Section A.4.3 of [4] and illustrated in Section 5.3 of [8].

Sampling Perturbations

Following [5, 7, 8], the perturbation coefficients $\delta V_\mu^{(k)}$ are applied during the Kohn–Sham calculations and are sampled independently from a zero-mean normal distribution $\mathcal{N}(0, \sigma_k^2)$. The standard deviation σ_k follows a linearly decreasing schedule over SCF iterations, from $\sigma = 0.102$ at iteration 6 down to $\sigma = 0.002$ at iteration 26. Outside this range, no perturbation is applied: the SCF calculation runs unperturbed for the first five iterations and again from iteration 27 onward until convergence.

The schedule was calibrated in Section 5.2 of [7], such that the perturbed densities are approximately as far from the ground state as the MINAO initialization [12, 13]. Starting perturbations only at iteration 6 prevents excessive density deviations during the early SCF steps, where the density is still far from convergence.

Reproducing the perturbation method from [5], we depict in Figure 5.6 the distribution of L^2 density differences to the ground state for the Structures25 QM9 train split per SCF iteration. The L^2 distance is computed as $\|\rho - \rho^*\|_2$. In the first iterations, no perturbations are applied and the density rapidly approaches the ground state. It is unclear why most samples at iteration 2 have a larger L^2 distance than at iteration 1. Overall, however, the trend from iterations 0 to 5 is a rapid decrease in L^2 distance due to the SCF procedure. At iteration 6, perturbations are turned on, causing a jump in the median L^2 distance. From there, the median L^2 distance decreases roughly linearly through iteration 26, following the perturbation schedule with linearly decreasing σ_k. After iteration 26, perturbations stop and the density continues to converge towards the ground state.

A notable exception is iteration 14 (9^{th} perturbed iteration), where the median stalls for one iteration, before resuming its downward trend. This is likely due to the Direct Inversion in the Iterative Subspace (DIIS) procedure [30], used to speed up SCF convergence. DIIS extrapolates the new Fock matrix as linear combination of Fock matrices from past iterations (we only cache the *unperturbed* Fock matrices for DIIS) and solves a least-squares problem to find the optimal coefficients for this linear combination.

In our case, we use 8 cached Fock matrices for DIIS. Before perturbations begin, the subspace is populated with Fock matrices computed from fully unperturbed densities. Once perturbations start, these pre-perturbation matrices serve as an anchor that biases the DIIS extrapolation towards the ground state. After exactly 8 perturbed iterations, the DIIS subspace size is exhausted and the last pre-perturbation Fock matrix is evicted. At that point (9^{th} perturbed iteration), DIIS loses this anchor entirely and operates solely on Fock matrices built from perturbed densities. This likely explains the temporary stalling of the median L^2 distance at iteration 14.

We also notice this effect in Figure 5.7, which depicts a histogram of L^2 density differences to the ground state for only *perturbed* samples (corresponding to iterations 6–26 marked in red in Figure 5.6). The Structures25 model [5] only trains on these perturbed samples (and additionally on the true ground states, excluded from this plot). In the interval $[0.06, 0.22]$, the number of samples is approximately uniform across L^2 distances. The bump at around L^2 distance 0.31 corresponds to the aforementioned stalling of the median at iteration 14, which causes a local accumulation of samples at that distance. Interestingly, the valley at around L^2 distance 0.22 is not as accentuated as one might expect from Figure 5.6, where the stalling of the median seems to not affect the linear downward trend for the subsequent 15^{th} iteration, that is, removing iteration 14 entirely, we would get a very good linear fit of the median L^2 distance across iterations 6–26.

Furthermore, Figure 5.7 reveals a more fundamental issue of the training data: there is a gap in the L^2 distance distribution for very small distances to the ground state. By choosing the lower bound of the perturbation schedule as $\sigma = 0.002$, [7] tried to avoid this issue, ensuring that the smallest perturbations generate samples that are close to the ground state. However, we see in Figure 5.7 that there is still a gap at small L^2 distances. We think this is detrimental for training, since the model will not see any samples close to the ground state, which is a relevant region for density optimization.

This motivates our alternative perturbation strategy described next, which decouples perturbations from the SCF procedure and allows to directly prescribe the distribution of L^2 distances to the ground state.

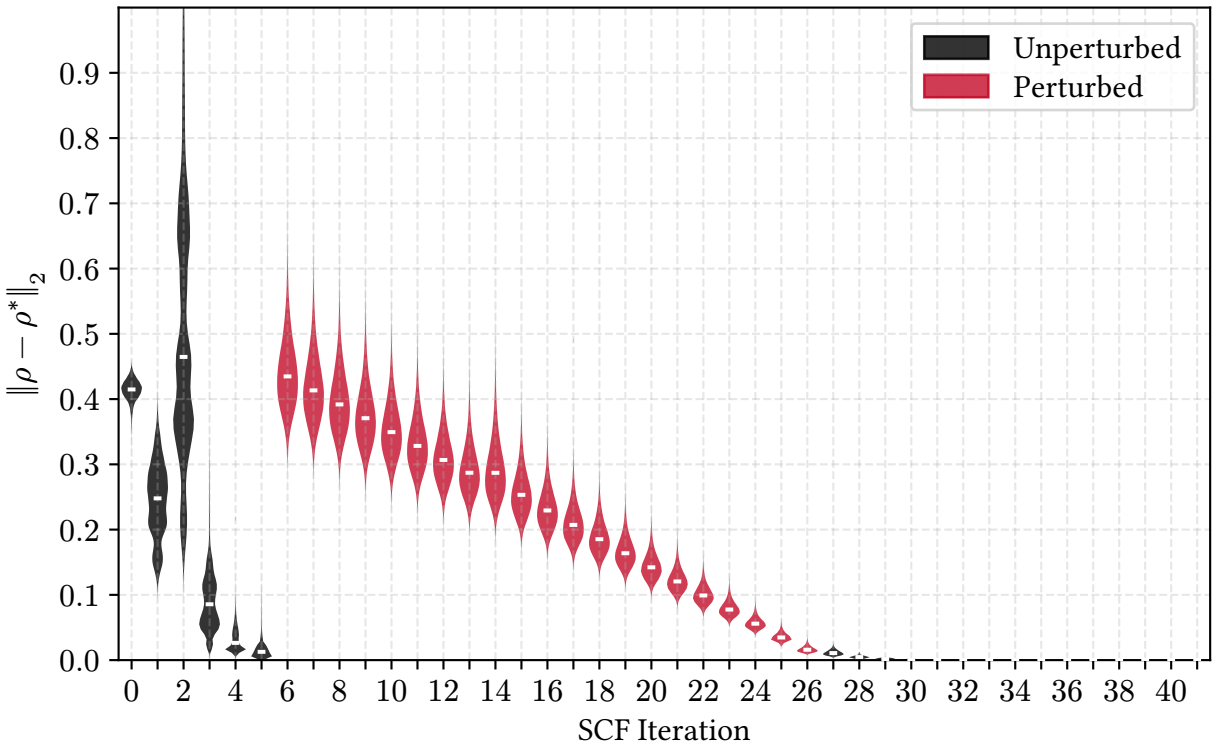


Figure 5.6: Violin plot of L^2 density differences to the ground state by SCF iteration. The ground states themselves are excluded. White markers indicate the median. Perturbations start at iteration 6 and end at iteration 26 (red). Plot cut off at L^2 distance 1.0 (excluding 0.05% of non-ground-state samples).

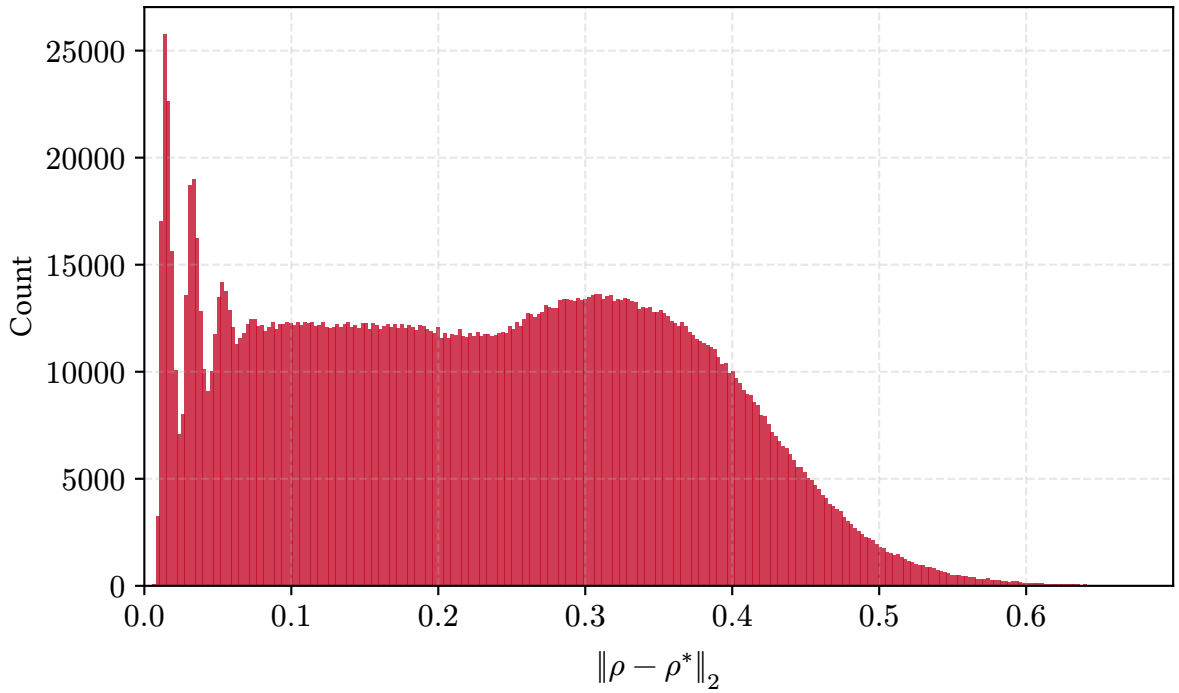


Figure 5.7: Histogram of L^2 density differences to the ground state. Only perturbed SCF samples are shown, corresponding to the red region in Figure 5.6. Plot cut off at L^2 distance 0.7 (excluding 0.02% of perturbed samples). Max L^2 distance: 1.6598.

5.2 Perturbations from the Ground State

The method described in Section 5.1 applies perturbations *during* the SCF procedure: in each iteration k, a perturbation $\delta V^{(k)}$ is added before solving the eigenvalue problem (48), and the resulting orbitals (and thus density) provides one training sample. This approach, used in [5, 7, 8], has several practical limitations.

Problems with SCF-based perturbation

Limited number of samples per molecule. The number of perturbed samples is bounded by the number of SCF iterations (typically 20–30). Generating more samples would require running additional, unnecessary SCF iterations past convergence, or restarting the calculation entirely.

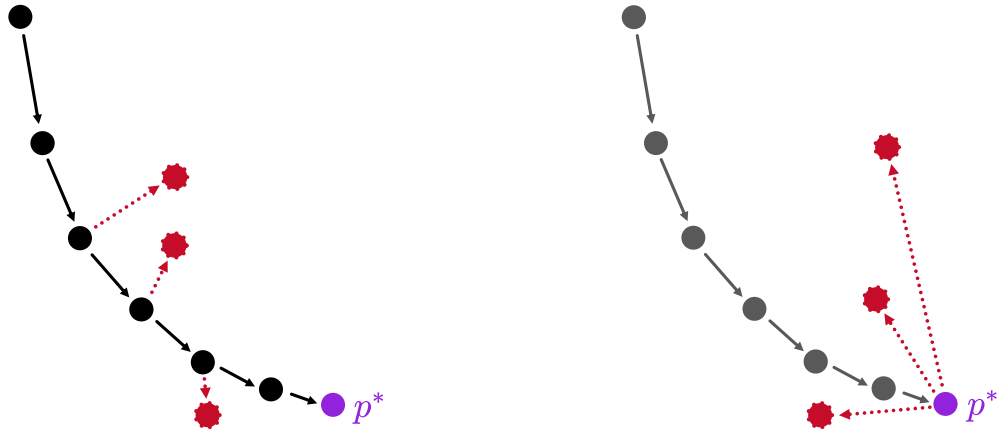
SCF trajectory necessary. New datasets might already contain ground-state information, for instance in the form of converged densities or Fock matrices, but no SCF trajectories. With the SCF-based perturbation method, such data cannot be leveraged since the method relies on the presence of an SCF trajectory to apply perturbations at each iteration. This also limits the use of external chemistry software packages like ORCA, where no access to SCF trajectories might be available.

Brittle implementation. We use the Python package PySCF [15–17] for data generation and monkey-patch its Kohn–Sham calculations to apply perturbations during the SCF procedure. This creates a brittle dependency on the internal workings of PySCF's SCF implementation, which might change across versions and is not designed for this use case.

Dependence on the exact SCF trajectory. The perturbation in iteration k depends on the density obtained from the previous iteration, which in turn depends on all past perturbations. This makes it difficult to control the distribution of perturbed densities and to avoid gaps in it. In [7], this problem is partially addressed by delaying the start of perturbations until iteration 6, since in iteration 5, the density is already close to the ground state. However, this requires manual tuning of the perturbation start point and still does not guarantee a gap-free distribution of L^2 distances to the ground state, as seen in Figure 5.7.

Method

We propose a new data generation method: run the KSDFT calculation to convergence *without* perturbations, then generate all perturbed samples from the converged ground state. This decouples perturbation-based data generation from the SCF procedure entirely. Figure 8 illustrates the two approaches.



(a) Perturbing during SCF iterations (b) Perturbing from the ground state (ours)

Figure 8: Schematic illustration of perturbation strategies. Black/Gray: SCF iterations. Red: Perturbations. Purple: True ground state p^* according to the last SCF iteration.

Let $\boldsymbol{p}^*$ denote the ground-state density coefficients obtained from the converged KSDFT calculation (Figure 8b), with ground-state density matrix $\boldsymbol{\Gamma}^*$ and Fock matrix $\boldsymbol{F}^*$. Let K denote the number of perturbations to generate per molecule, which is now a free parameter that can be chosen independently of the SCF iteration count. To generate a perturbation, we follow the same general steps as in the SCF-based approach, but now starting from the fixed ground state instead of the iteratively changing SCF trajectory.

Let $\delta\boldsymbol{V}$ denote the perturbation vector in the density basis, holding coefficients δV_μ of the perturbation potential $\delta V(\boldsymbol{r})$ in (76). To allow for fine-grained control over the distribution of perturbed densities, we parameterize $\delta\boldsymbol{V}$ in terms of radius $r > 0$ and unit direction $\boldsymbol{u}$:

$$\delta\boldsymbol{V} := r \cdot \boldsymbol{u}, \quad \boldsymbol{u}^T \boldsymbol{W} \boldsymbol{u} \stackrel{!}{=} 1 \tag{85}$$

The direction $\boldsymbol{u}$ is normalized to ensure that the perturbation vector has length r:

$$\|\delta\boldsymbol{V}\|_2 = \sqrt{(\delta\boldsymbol{V})^T \boldsymbol{W} \delta\boldsymbol{V}} = \sqrt{r^2 \boldsymbol{u}^T \boldsymbol{W} \boldsymbol{u}} = r \tag{86}$$

Here, the overlap matrix $\boldsymbol{W}$ of the density basis occurs since the perturbation is expressed in that basis (76). In practice, $\boldsymbol{u}$ is obtained by drawing from a normal distribution $\boldsymbol{z} \sim \mathcal{N}(0, 1)$ and normalizing

$$\boldsymbol{u} := \frac{\boldsymbol{z}}{\sqrt{\boldsymbol{z}^\top \boldsymbol{W} \boldsymbol{z}}} \tag{87}$$

while rejecting near-zero samples where the normalization would be unstable. This procedure yields perturbation directions uniformly on the unit sphere defined by the $\boldsymbol{W}$-norm, ensuring that no direction is favored over another.

We can summarize the perturbation sampling procedure as follows:

1. Sample the radius r from a prescribed radial probability distribution function (PDF) and a normalized direction $\boldsymbol{u}$, to obtain the perturbation vector $\delta\boldsymbol{V} := r \cdot \boldsymbol{u}$ in the density basis.

2. Construct the perturbed Fock matrix based on the ground-state Fock matrix $\boldsymbol{F}^*$ and the perturbation $\delta\boldsymbol{V}$. By (80), the entries of the perturbed Fock matrix are

$$\boldsymbol{F}_{\mathrm{pert},\alpha\beta} := \boldsymbol{F}^*_{\alpha\beta} + \sum_{\mu=1}^{M} \delta V_\mu W_{\mu\alpha\beta} \tag{88}$$

3. Solve the generalized eigenvalue problem $\boldsymbol{F}_{\mathrm{pert}}\boldsymbol{C} = \boldsymbol{S}\boldsymbol{C}\boldsymbol{\varepsilon}$ for the perturbed orbital coefficients $\boldsymbol{C}$, from which the perturbed density matrix $\boldsymbol{\Gamma}$ and density coefficients $\boldsymbol{p}$ follow by density fitting (60).

4. Compute the energy labels (T_S, E_H, E_XC, E_ext) for the perturbed density $\boldsymbol{p}$ and the kinetic energy gradient (83), using the same procedures as for unperturbed SCF iterations. The effective potential at the ground state is computed once and reused for all perturbations.

The unperturbed ground state itself is included as an additional sample, such that the data for each molecule comprises the converged KSDFT result together with any number K of perturbations around it. Results are stored in `.zarr.zip` files.

Sampling the perturbation radius

With the new radius parameter r, we can prescribe the distribution of L^2 density distances to the ground state by choosing a suitable PDF for r. Note however that this radius determines only the length of the perturbation vector $\delta\boldsymbol{V}$ in the density basis, not the actual L^2 distance of the resulting perturbed density to the ground state.

As described in Section 5.1, we perturb the effective potential V_eff, which in turn gives an additive contribution to the Fock matrix $\boldsymbol{F}$ in (80). This perturbed Fock matrix is then used to solve the eigenvalue problem (48), yielding the perturbed density coefficients $\boldsymbol{p}$ via density fitting (60). Due to this chain of non-linear transformations, finding a closed-form relationship between the perturbation radius r and the resulting L^2 distance of the perturbed density to the ground state is difficult and would essentially require inverse KSDFT calculations.

Despite this limitation, we find that for small perturbation radii, the resulting L^2 density distance to the ground state is approximately proportional to r. One might think of this as a first-order approximation of the response of the system to perturbations.

We leverage this by proposing a new perturbation sampling framework that allows to generate perturbations according to an arbitrary, continuous PDF $f : \mathbb{R} \to [0, \infty)$, where $\int f(x)\,\mathrm{d}x = 1$. In order to sample perturbations matching the distribution f, we make use of the inversion principle presented as Theorem 2.1 in chapter II.2 in [31].

Theorem 5.1 (Inversion Principle): Let F be the cumulative distribution function (CDF) on $\mathbb{R}$ corresponding to PDF f, defined by $F(x) = \int_{-\infty}^{x} f(t)\,\mathrm{d}t$. Let F^{-1} be the generalized inverse distribution of F, defined by

$$F^{-1}(u) := \inf\left\{x \in \mathbb{R} \;\middle|\; F(x) \geq u\right\}, \quad u \in [0, 1] \tag{89}$$

If u is a uniform random variable on $[0, 1]$, then $F^{-1}(u)$ has F as its CDF, and therefore follows the distribution defined by PDF f.

Proof: Let $u \sim \mathrm{Uniform}[0, 1]$. Let $a \in \mathbb{R}$. Take $x := F^{-1}(u) \in \mathbb{R}$.

$$\mathbb{P}(x \leq a) = \mathbb{P}(F^{-1}(u) \leq a) = \mathbb{P}(\inf\{y \mid F(y) \geq u\} \leq a) \tag{90}$$

$$\overset{(*)}{=} \mathbb{P}(u \leq F(a)) = F(a) \tag{91}$$

The last step uses that u is uniform on $[0, 1]$. In step $(*)$, we employ the equivalence $\inf\{y \mid F(y) \geq u\} \leq a \Leftrightarrow u \leq F(a)$, justified as follows:

- ($\Leftarrow$): If $u \leq F(a)$, then $a \in \{y \mid F(y) \geq u\}$, so the infimum of that set is at most a.
- ($\Rightarrow$): Let $x^* = \inf\{y \mid F(y) \geq u\}$ and suppose $x^* \leq a$. By definition of the infimum, there exists a sequence $y_n \searrow x^*$ with $F(y_n) \geq u$ for all n. Right-continuity of F then gives $F(x^*) = \lim_{n\to\infty} F(y_n) \geq u$. Since F is weakly monotonically increasing and $x^* \leq a$, we conclude $F(a) \geq F(x^*) \geq u$. Right-continuity and monotonicity of F are guaranteed by the definition of a CDF.

Overall, we remark that $\mathbb{P}(x \leq a) = F(a)$. And since x was chosen as $x := F^{-1}(u)$, we see that $F^{-1}(u)$ has F as its CDF. ∎

This suggests a simple **sampling procedure**: generate a uniform random variable u on $[0, 1]$. Then take $x := F^{-1}(u)$ as sample. As seen, x follows the distribution with CDF F as desired.

In practice, we specify a PDF f rather than a closed-form CDF F, so F is computed numerically before applying the inversion principle. Here, f describes a non-negative perturbation radius with support $[0, r_{\max}]$ for some finite $r_{\max} > 0$. Therefore, $f(r) = 0$ for $r < 0$, which implies $F(0) = \int_{-\infty}^{0} f(r)\,\mathrm{d}r = 0$. We construct a uniform grid with $N \geq 2$ points

$$r_i = \frac{i}{N-1} r_{\max}, \quad i = 0, 1, ..., N-1 \tag{92}$$

The CDF is approximated by the trapezoidal rule, initialized with $\tilde{F}(r_0) = 0$:

$$\tilde{F}(r_i) = \sum_{j=1}^{i} \frac{f(r_{j-1}) + f(r_j)}{2} (r_j - r_{j-1}), \quad i = 1, ..., N-1 \tag{93}$$

In our code, we do not want the user to worry about normalization of the PDF f, so we allow f to be specified up to a normalization constant. Note how the above theorem does not require f to be normalized, so the inversion principle still holds for the unnormalized CDF $\tilde{F}$. However, since we want to sample from the distribution defined by f, we need to ensure that the CDF is properly normalized:

$$F(r_i) := \frac{\tilde{F}(r_i)}{\tilde{F}(r_{N-1})} \tag{94}$$

enforcing $F(r_0) = 0$ and $F(r_{N-1}) = 1$. The generalized inverse F^{-1} is evaluated by linear interpolation on the pairs $(F(r_i), r_i)$.

Generated Data

With our proposed perturbation method, we generate new training data based on the QM9 dataset [6], while reusing already existing converged KSDFT results. This shows how our method can leverage existing ground-state data without requiring SCF trajectories. For each molecule, we generate $K = 20$ perturbations from the ground state according to different PDFs for the perturbation radius r. We only specify the functional form of the PDFs up to a normalization constant.

Name	**PDF**	
Uniform	$f(r) = \begin{cases} r_{\max} & r \in [0, r_{\max}] \\ 0 & \text{otherwise} \end{cases}$	(95)
Linear	$f(r) = \begin{cases} 1 - \frac{r}{r_{\max}} & r \in [0, r_{\max}] \\ 0 & \text{otherwise} \end{cases}$	(96)
Cosine	$f(r) = \begin{cases} \left(1 + \cos\left(\pi \frac{r}{r_{\max}}\right)\right)^p, \ p \in \mathbb{R}^+ & r \in [0, r_{\max}] \\ 0 & \text{otherwise} \end{cases}$	(97)

Heuristically, we fix $r_{\max} := 0.34$, which roughly corresponds to a mean L^2 density distance of 0.042 to the ground state. This choice is motivated by the DGSP Equiformer results in Figure 4.5, where we find that 99.89% of the predicted densities have an L^2 distance to the ground state of at most 0.042. For the cosine PDF, we set the annealing power exponent to $p = 1.5$ (the exact numerical value is arbitrary, but we choose $p > 1$ to put more emphasis on small perturbation radii).

All data was generated on the "bwForCluster Helix" using at most 1000 CPU cores in parallel. Data generation took at most 24 hours ($\approx$ 130000 CPU hours) for each PDF. The resulting datasets are each around 200 GB in size for all 133,885 molecules. The original labels used in [5] are 337 GB in size, but also use one more SCF iteration per molecule (21 instead of 20) and include the unperturbed SCF trajectory. We check the consistency of the stored data by making sure that the energy gradient label is 0.0 (within numerical precision) for the ground state sample of each molecule.

Figures 5.10, 5.11 and 5.12 show the distribution of L^2 density differences to the ground state for the perturbed samples generated with the uniform, linear and cosine PDFs, respectively. In Figure 5.9, we overlay the three distributions for comparison. The ground state itself is excluded from these plots.

We find that in all cases, the distribution of L^2 distances to the ground state nicely follows the shape of the underlying PDF (with some small deviations in counts likely due to numerical inaccuracies). This validates the effectiveness of our perturbation sampling method in controlling the distribution of perturbed densities. Furthermore, the problem of a gap at small L^2 distances to the ground state, observed in Figure 5.7 for the SCF-based perturbation method, is successfully resolved by our new method, where by design we choose the grid points (92) to start at $r = 0$.

However, at the right tail of every distribution, we observe an exponential decay. This is particularly pronounced in Figure 5.10, where a harder cutoff would have been expected. As discussed above, the relationship between perturbation radius r and the resulting L^2 distance to the ground state is non-linear. For larger perturbations, the linear response approximation no longer holds, and the system's response to perturbations becomes more complex. A systematic analysis of this behavior is an interesting subject for future research.

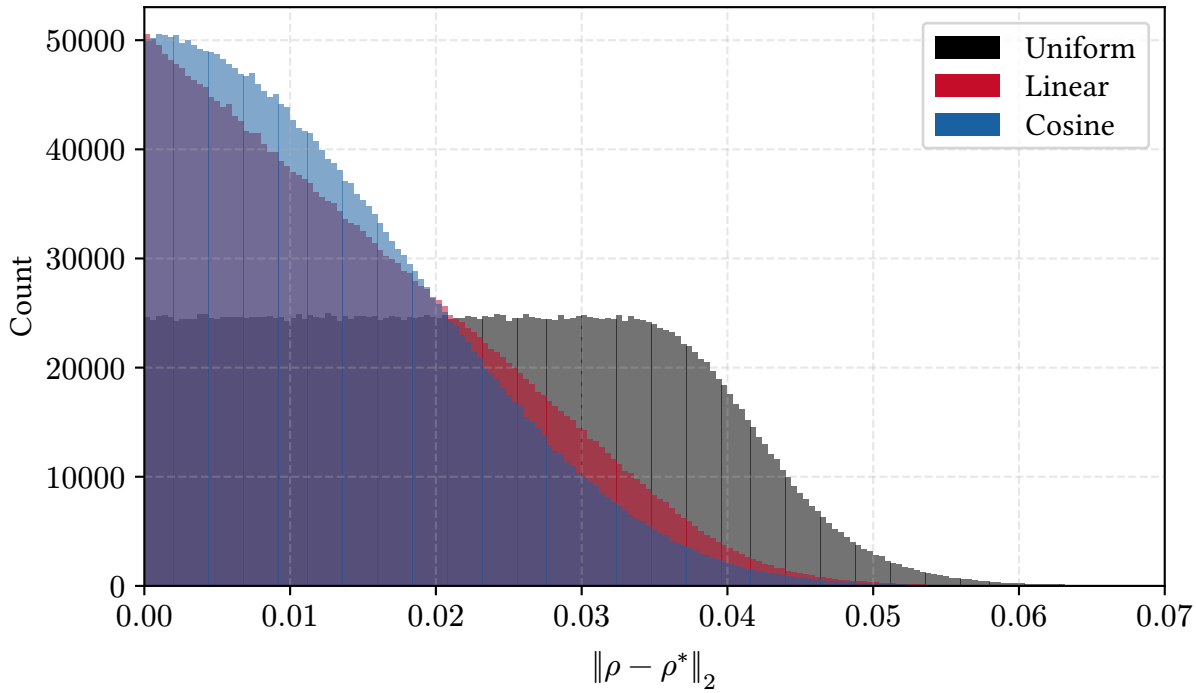


Figure 5.9: Generated perturbed data according to the proposed perturbation method for more controllable data generation (133 885 molecules with 20 perturbations each). Overlay of Figures 5.10, 5.11 and 5.12.

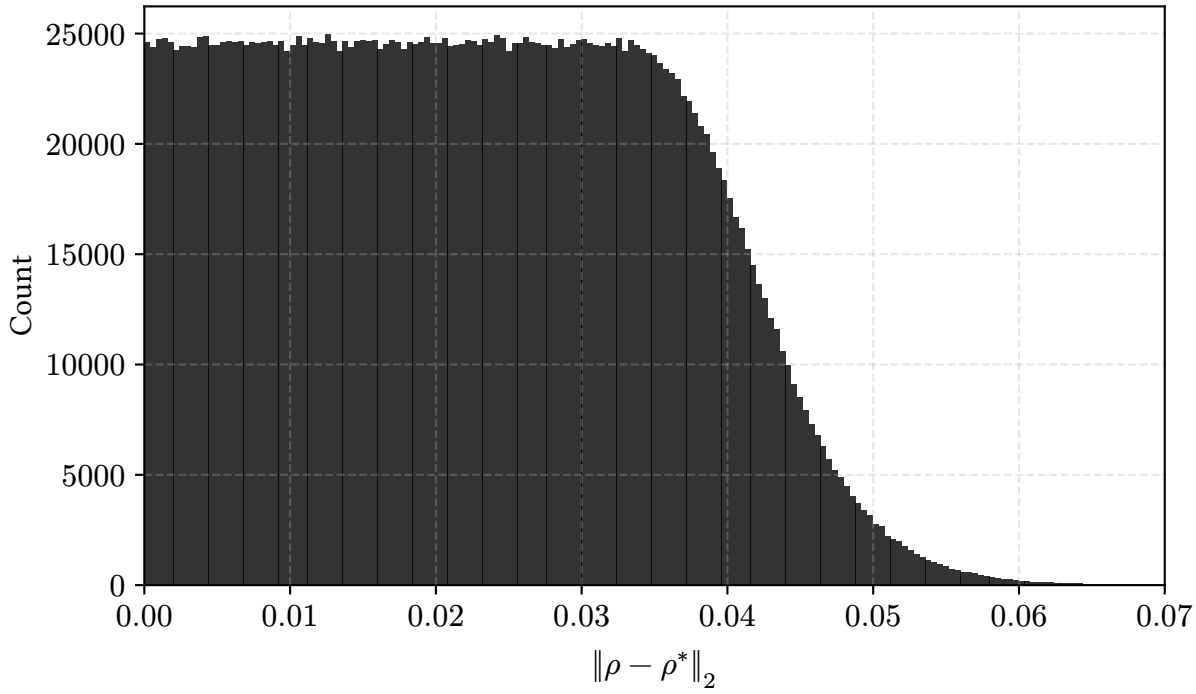


Figure 5.10: Perturbed data generated with the Uniform PDF (95).
Plot cut off at L^2 distance 0.07 (excluding 96 samples, i.e. 0.004%). 1200 bins.

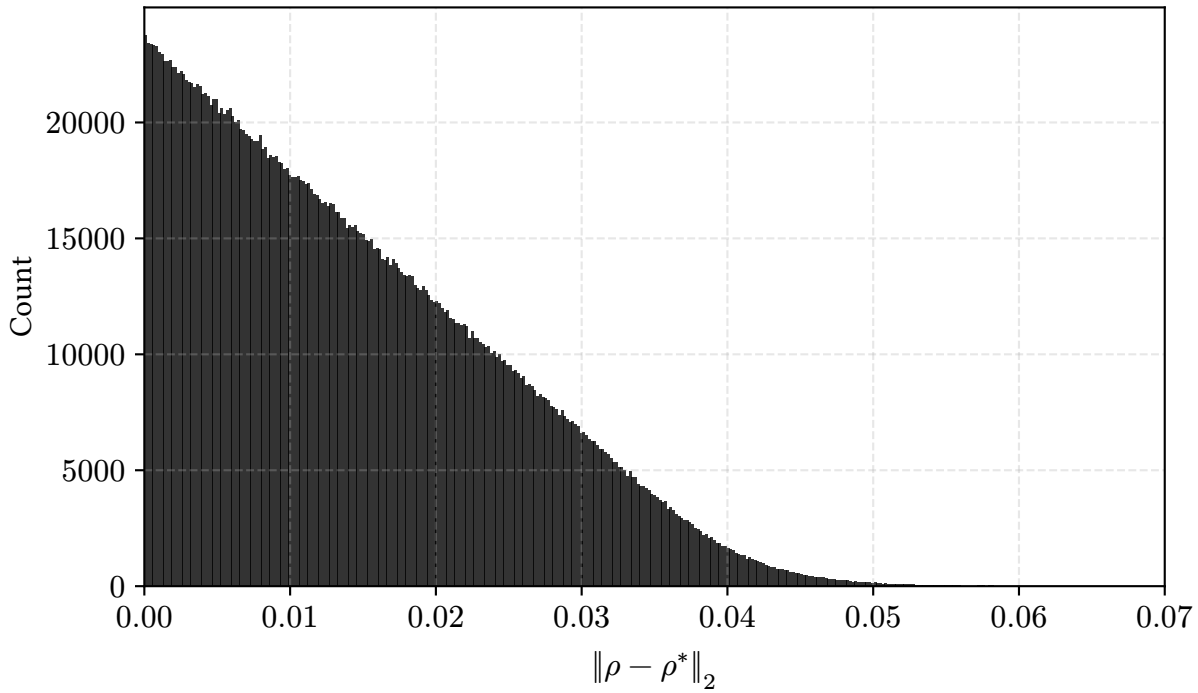


Figure 5.11: Perturbed data generated with the Linear PDF (96).
Plot cut off at L^2 distance 0.07 (excluding 19 samples, i.e. 0.0007%). 1200 bins.

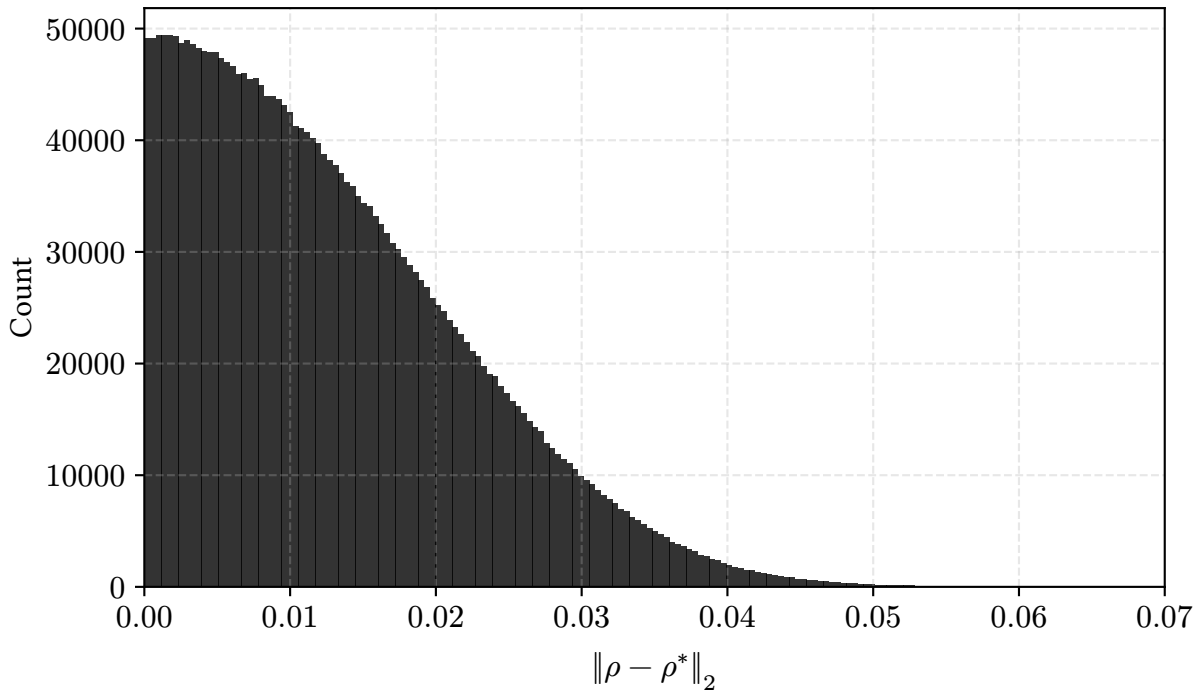


Figure 5.12: Perturbed data generated with the Cosine PDF (97).
Plot cut off at L^2 distance 0.07 (excluding 17 samples, i.e. 0.0006%). 1200 bins.

5.3 Evaluation on the new data

For every newly generated dataset, we train the Graphormer model with the same hyperparameters as in Structures25 [5], and evaluate the resulting model by running Density Optimization on the test split of QM9. This allows us to evaluate how far the learned ground state is from the label ground state, given different training data. We compare the results to a baseline model trained on the original SCF-based perturbation data from [5]. Results are shown in Table 5.1.

Trained on PDF	**Mean Total Energy Error** ΔE [mHa]	**Mean Density Error** $\|\Delta\rho\|_2$
Baseline	0.7182	0.0167
Uniform	2.2528	0.0292
Linear	2.4823	0.0261
Cosine	2.2786	0.0297

Table 5.1: Density Optimization on the test split (13,389 molecules), initialized at the true ground state (label).

Contrary to our expectations, the new training data does not lead to improved performance. In fact, the baseline model trained on the original SCF-based perturbation data from [5] outperforms all other models. Among the newly trained models, the one trained on the uniform PDF performs best in terms of energy error, while the one trained on the linear PDF performs best in terms of density error (highlighted in bold).

One reason for the poor performance of the new models could be the fact that the original data contains samples up to an L^2 distance of around 0.6 to the ground state (see Figure 5.7), while the new data only contains samples up to an L^2 distance of around 0.06 (see Figure 5.9). It might be that the presence of samples farther away from the ground state in the original data helps the model to learn a better optimization landscape for Density Optimization, even in the region close to the ground state. This suggests that having a wide range of perturbations, including larger ones, might be beneficial for training.

Figures 5.13 to 5.16 show the Density Optimization trajectories on the test split, relating the L^2 density error to the energy gradient norm along each trajectory. Comparing Figure 5.13 with 5.15, we observe that initialization at the label leads to smaller initial gradient norms for the uniform model, as expected because its training data contains more samples close to the ground state. For both models, each trajectory contains a region with an inverse "S" shape: as the gradient norm decreases, both the gradient norm and the density error briefly increase before decreasing again. This may indicate that the optimizer passes over a local hill in the optimization landscape, but it is not evident where this behavior comes from.

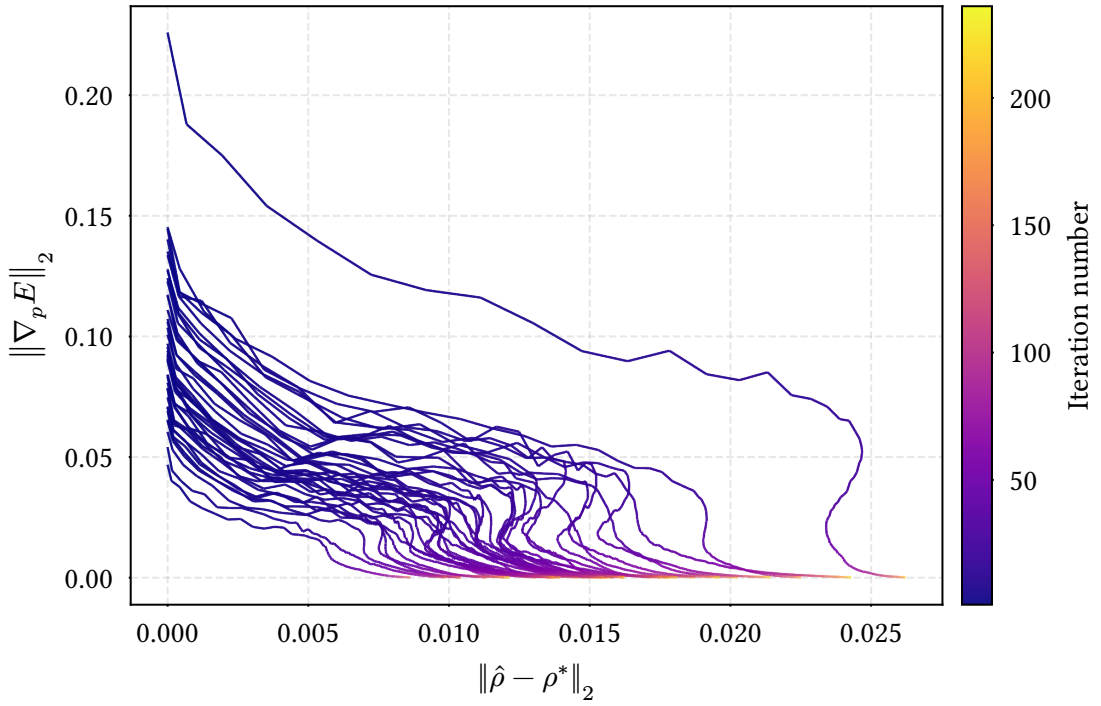


Figure 5.13: Baseline model. Each line shows a density optimization trajectory for one of 40 randomly sampled test molecules, plotted in the space of L^2 density error vs. energy gradient norm. Points along the same trajectory are connected, and the color encodes the iteration index. The trajectories are initialized at the true ground state (label) and end at the stopping index of Density Optimization, where the energy gradient norm falls below 10^{-4}. Number of maximum optimization steps needed: 237.

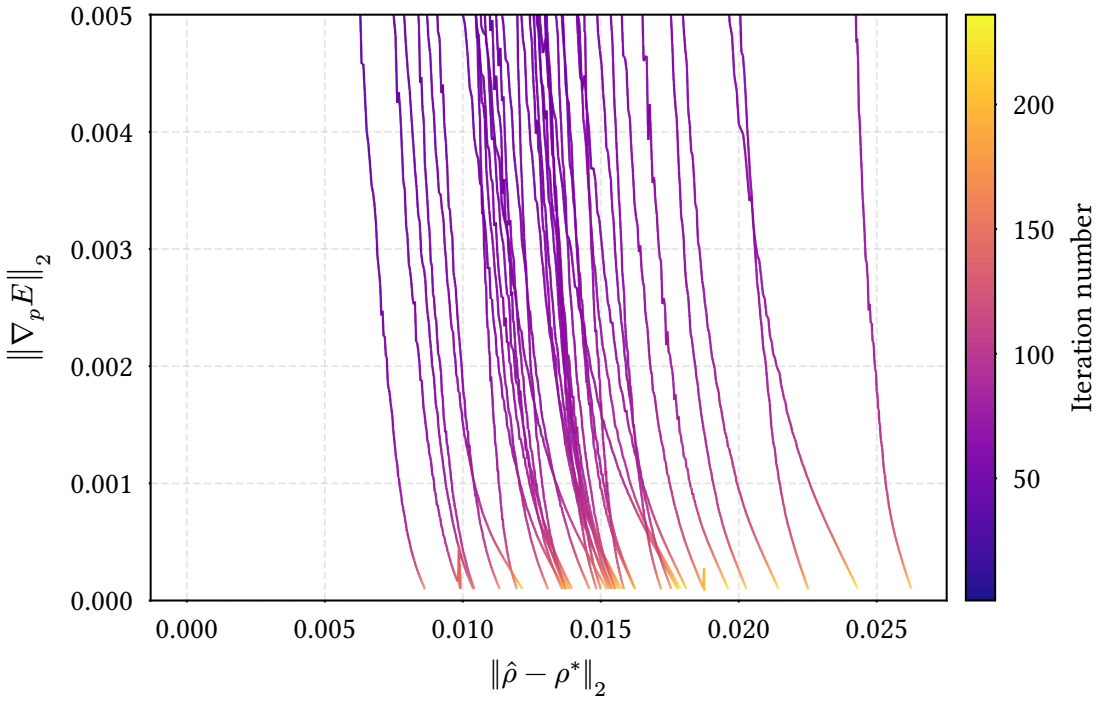


Figure 5.14: Baseline model (zoom). Same trajectories as in Figure 5.13, restricted to the low-error region.

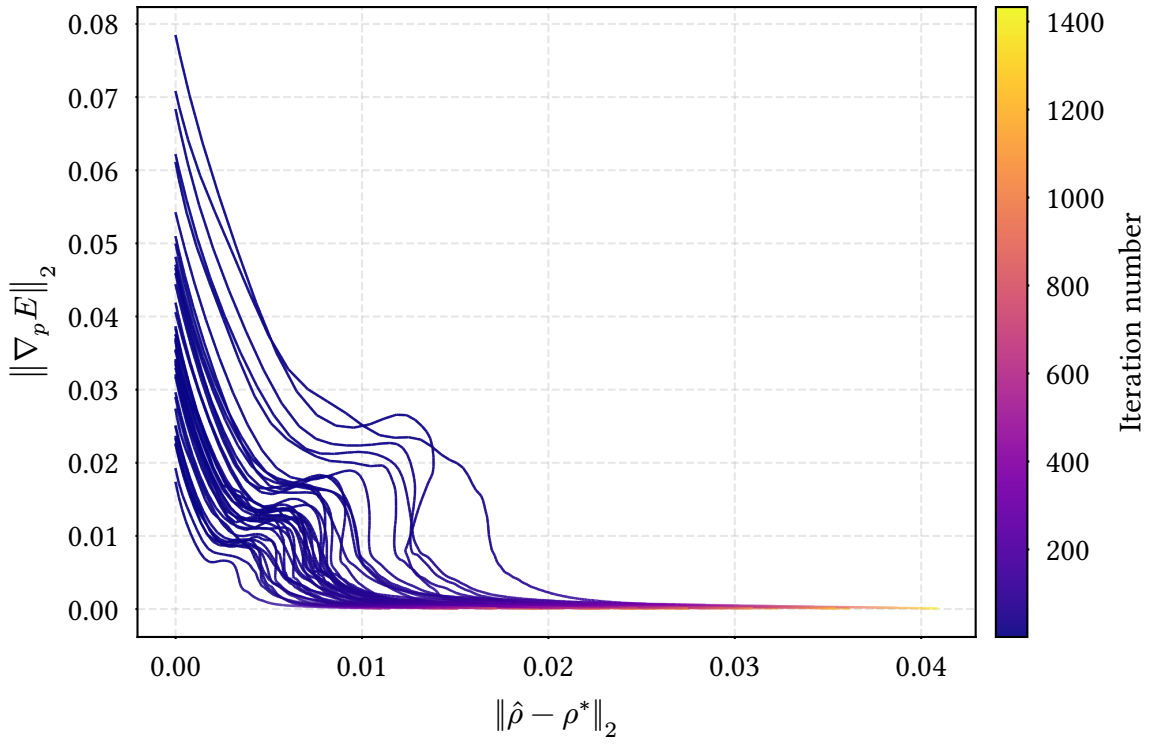


Figure 5.15: Model trained on uniform data. Density Optimization trajectories for the same 40 randomly sampled test molecules as in Figure 5.13. See Figure 5.13 for more details on the plot setup. Number of maximum optimization steps needed: 1 434.

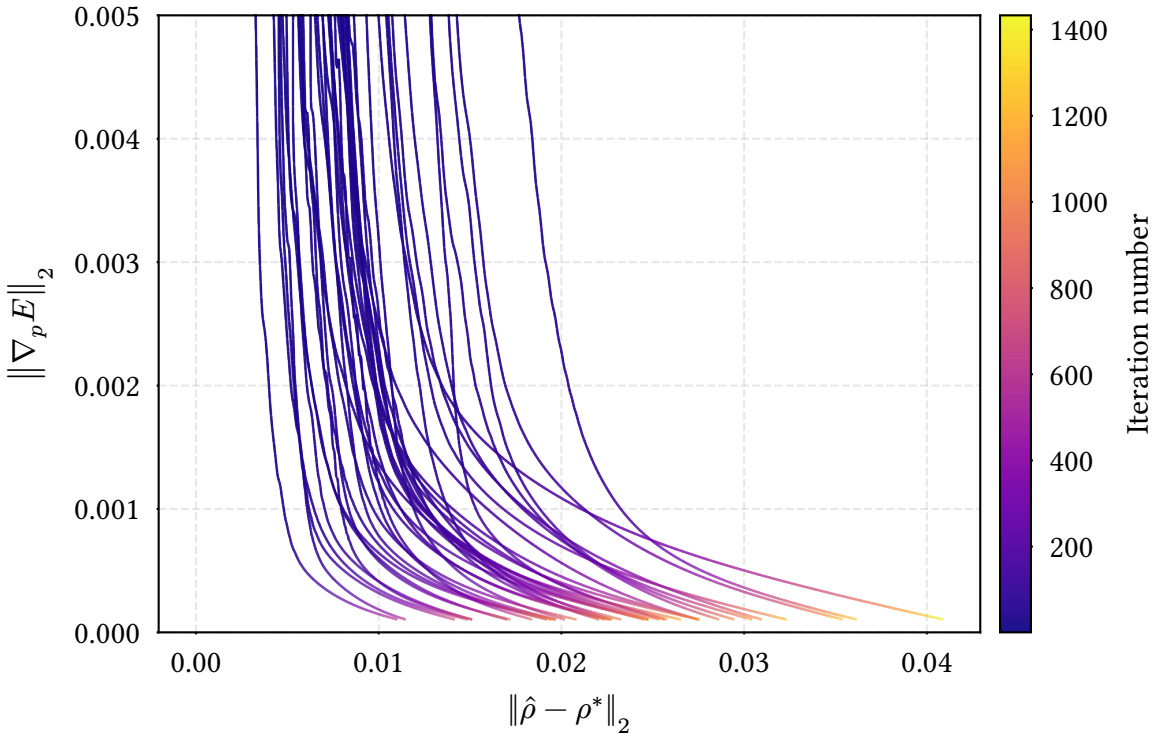


Figure 5.16: Uniform model (zoom). Same trajectories as in Figure 5.15, restricted to the low-error region.

After the first bulge, comparing Figure 5.14 with 5.16, we observe that the baseline model rapidly decreases the energy gradient norm in the low-error region while leaving the L^2 density error nearly unchanged. In contrast, the uniform model requires many more steps to reach gradient norms below 0.0015 (which is why the lines appear smoother in Figure 5.16).

The tail of this "S" shape can be best described as "drifting" behavior, especially visible in the uniform model in Figure 5.16, where the gradient norm continues to decrease while the density error increases. This results in negatively curved trajectory lines. Therefore, the final reported density errors are higher than what one would obtain when extrapolating a linear fit through the region above an energy gradient norm of 0.0015 in Figure 5.16.

A possible explanation is optimizer momentum: if the minimum of the learned functional is slightly shifted relative to the true ground state, the trajectory may first move away from the true ground state while building momentum, then slightly overshoot the learned minimum, and finally return towards it in the last iterations. Note that the "drifting" behavior can already be observed for the Baseline model in Figure 5.14, but in a much less pronounced way, which may explain why the final density errors are still lower than for the model trained on the uniform data. Further analysis is required to understand the origin of this "S" shape and the drifting behavior, and to investigate how it can be mitigated.

Combining the Prime- with the Refine-Model

So far, the trained models have only been evaluated on Density Optimization trajectories initialized from the true ground-state density (label). The long-term objective, however, is to use a Refine-Model to improve the predictions of the DGSP (Prime-Model). In this setup, a single forward pass of the Prime-Model provides an initial density guess for the Refine-Model, which can subsequently be optimized to high accuracy through Density Optimization while also producing accurate energy labels.

In the Structures25 [5] codebase, Density Optimization is modified. An additional path to a DGSP checkpoint and DGSP model config can be provided, which triggers the Prime–Refine workflow. The DGSP model is loaded once per process, then shared across all worker threads. For each sample, the stored MINAO initial guess (SCF iteration 0) is loaded from the dataset and fed through the DGSP model, which predicts a correction. This correction is added to the MINAO coefficients to obtain a DGSP-corrected density, which is then used directly as the initial density for Density Optimization.

Preliminary results for the Prime–Refine workflow with the Equiformer DGSP model suggest that DGSP already produces densities very close to the ground state (see Section 4.3, where it reaches a L^2 density error of 0.0075 on the QM9 test split). However, during the subsequent optimization, the Refine-Model appears to move away from the ground state, so that the final results are worse than those obtained from the DGSP prediction alone. This matches

the “drifting” behavior seen in Figure 5.16: even when the optimization is initialized at the label ground state, it departs from the ground state in the low-error regime.

Future work should focus on two directions. First, the drifting behavior and the “S”-shaped optimization trajectories need to be understood and resolved before the Refine-Model can be used reliably. One approach, deviating from what was done here, is to generate training data with a wider range of L^2 distances to the ground state, including perturbations much larger than those used here, so that the KEDF model has better coverage of the optimization landscape encountered during Density Optimization. Alternatively, one could explore training objectives that explicitly penalize drifting, for instance by adding a regularization term that discourages the modeled gradient from pointing away from the ground state in the low-error regime. Second, once the drifting issue is resolved, the complete Prime–Refine pipeline should be evaluated end-to-end: using the DGSP prediction as the starting density for Density Optimization, and assessing the overall performance in terms of final energy and density errors.

6 Conclusion

This thesis investigates two complementary strategies for improving ground state prediction in orbital-free density functional theory (OF-DFT): *Direct Ground State Prediction* (DGSP), which approximates the ground state density in a single forward pass, and improves *data generation* for the variational Density Optimization approach.

Building on the Structures25 codebase, we first introduce DGSP models that predict the ground state density coefficients directly from the molecular geometry, bypassing the iterative density optimization loop entirely. By adding an MLP readout head to the existing GNN backbone and training with a physically motivated natural L^2 loss that incorporates the basis overlap matrix, both the Graphormer and Equiformer variants achieve L^2 density errors that surpass the Denop baseline on the QM9 test set. The Equiformer model reaches a mean L^2 density error of 0.0075, roughly halving the baseline error of 0.0167, and does so in a single forward pass rather than ≈ 200 optimization iterations. Evaluating the full test set with DGSP is approximately an order of magnitude faster than with Denop.

Second, we propose a new perturbation-based data generation method that decouples the creation of training samples from the Kohn–Sham SCF procedure. Instead of perturbing the effective potential during the SCF iterations, our method generates all perturbed samples from the converged ground state. By decomposing the perturbation vector into a radius and a normalized direction, we gain direct control over the distribution of L^2 density distances to the ground state through a user-specified probability distribution function (PDF). We implement a general sampling framework based on the inversion principle that supports arbitrary continuous PDFs, and validate it by generating training data for three different distributions (uniform, linear, cosine). The resulting L^2 distance distributions closely follow the prescribed PDFs and, importantly, the gap at small L^2 distances present in the original SCF-based data is eliminated.

Despite these promising properties, models trained on the newly generated data do not yet surpass the Structures25 baseline in Density Optimization. We attribute this shortcoming to the narrower range of perturbation radii employed relative to the original data, indicating that broad coverage of the energy landscape, including regions far from the ground state, remains essential. This conclusion is reinforced by a "drifting" phenomenon observed during Density Optimization: when a model trained on the new data is initialized at the label ground state, the optimization trajectory moves away from the ground state, suggesting that the learned energy landscape is not sufficiently accurate in all directions around the ground state.

Several directions for future work emerge from these findings. First, the drifting behavior during Density Optimization in the low-error regime should be investigated more closely, for instance through regularization of the learned gradient or by training on data that covers a wider range of L^2 distances, including much larger perturbations. Second, the generalization of DGSP beyond QM9 to larger and more chemically diverse molecules, such as those in

QMugs, should be explored to assess whether the single-pass prediction remains reliable as molecular complexity grows. Third, the new perturbation framework could be combined with active learning strategies that adaptively sample perturbation radii in regions where the model's energy landscape is least accurate.

A list of software contributions made during this thesis, including infrastructure improvements, visualization tools and codebase enhancements that go beyond the scope of the main text, is provided in Appendix A.1.

7 Acknowledgements

I thank my family and friends for their support and encouragement throughout this project. I also like to thank the whole Scientific AI Lab for the inspiring and supportive research environment, and for the many thrilling table football games.

Supervision. I am very grateful to Peter Lippmann who introduced me to the topic of this thesis and supervised me throughout the project with very valuable input that made me learn a lot and helped shaping the direction of the thesis. I also thank Prof. Dr. Fred Hamprecht for giving me the opportunity to work on this project, as well as for creating a very warm and supportive research environment in the Scientific AI Lab. Moreover, I would like to thank Prof. Dr. Tristan Bereau for agreeing to be the second supervisor of this thesis.

Proofreading. I'd like to thank (in alphabetical order) Jannis Demel, Manuel Klockow, Marie Müller, Tobias Rieger, and Peter Lippmann for proofreading parts or all of this thesis. Their detailed feedback helped fix mistakes and improve the presentation of the material.

Discussions. I am grateful for the open and constructive discussions, especially with Peter Lippmann, Manuel Klockow and Tobias Kaczun. I also thank Tobias Kaczun for his very responsive code reviews and for his chemical expertise.

Typst. This thesis was written using Typst, a modern typesetting system with an open-source compiler. I'd like to thank the founders of Typst, Martin Haug and Laurenz Mädje, for creating and maintaining this great software. Thanks to the whole Typst community for their contributions in the Typst Universe (from where many amazing packages were imported for this thesis) and for their help in the forums. I'd like to also thank Myriad Dreamin, the creator of tinymist, a language server for Typst that I made extensive use of in VSCode.

Computational Resources. I thank Stefan Sander for maintaining the IWR compute cluster, without which training the models for this thesis would not have been possible. Moreover, as described in Section 5.2, new training data was generated using the "bwForCluster Helix". I am grateful for these provided computational resources and acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG.

8 Bibliography

[1] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," *Physical Review*, vol. 136, no. 3, pp. B864–B871, Nov. 1964, doi: 10.1103/PhysRev.136.B864.

[2] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects," *Physical Review*, vol. 140, no. 4, pp. A1133–A1138, Nov. 1965, doi: 10.1103/PhysRev.140.A1133.

[3] R. Remme, T. Kaczun, M. Scheurer, A. Dreuw, and F. A. Hamprecht, "KineticNet: Deep learning a transferable kinetic energy functional for orbital-free density functional theory," *The Journal of Chemical Physics*, vol. 159, no. 14, Oct. 2023, doi: 10.1063/5.0158275.

[4] H. Zhang *et al.*, "Overcoming the Barrier of Orbital-Free Density Functional Theory for Molecular Systems Using Deep Learning," *Nature Computational Science*, vol. 4, no. 3, pp. 210–223, Mar. 2024, doi: 10.1038/s43588-024-00605-8.

[5] R. Remme *et al.*, "Stable and Accurate Orbital-Free DFT Powered by Machine Learning," July 22, 2025. doi: 10.48550/arXiv.2503.00443.

[6] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific Data*, vol. 1, no. 1, p. 140022, Aug. 2014, doi: 10.1038/sdata.2014.22.

[7] M. V. Klockow, "Stable and Accurate Orbital-Free Density Functional Theory Enabled by Machine Learning," Master Thesis, Department of Physics and Astronomy Heidelberg University, 2024.

[8] R. Remme, "Machine Learning Chemically Accurate Orbital-Free Density Functional Theory," Doctoral Thesis, heiDOK (Heidelberger Dokumentenserver), Heidelberg, 2025. doi: 10.11588/heidok.00036155.

[9] V. Sahni, "Hohenberg–Kohn, Kohn–Sham, and Runge-Gross Density Functional Theories," *Quantal Density Functional Theory*. Springer, Berlin, Heidelberg, pp. 135–183, 2016. doi: 10.1007/978-3-662-49842-2_4.

[10] M. Levy, "Universal variational functionals of electron densities, first-order density matrices, and natural spin-orbitals and solution of the v-representability problem", *Proceedings of the National Academy of Sciences*, vol. 76, no. 12, pp. 6062–6065, Dec. 1979, doi: 10.1073/pnas.76.12.6062.

[11] E. H. Lieb, "Density functionals for coulomb systems," *International Journal of Quantum Chemistry*, vol. 24, no. 3, pp. 243–277, 1983, doi: 10.1002/qua.560240302.

[12] J. Almlöf, K. Faegri Jr., and K. Korsell, "Principles for a direct SCF approach to LICAO–MOab-initio calculations," *Journal of Computational Chemistry*, vol. 3, no. 3, pp. 385–399, 1982, doi: https://doi.org/10.1002/jcc.540030314.

[13] J. H. Van Lenthe, R. Zwaans, H. J. J. Van Dam, and M. F. Guest, "Starting SCF calculations by superposition of atomic densities," *Journal of Computational Chemistry*, vol. 27, no. 8, pp. 926–932, 2006, doi: https://doi.org/10.1002/jcc.20393.

[14] R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople, "Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions," *The Journal of Chemical Physics*, vol. 72, no. 1, pp. 650–654, 1980, doi: 10.1063/1.438955.

[15] Q. Sun *et al.*, "PySCF: the Python-based simulations of chemistry framework," *WIREs Computational Molecular Science*, vol. 8, no. 1, p. e1340, 2018, doi: 10.1002/wcms.1340.

[16] Q. Sun *et al.*, "Recent developments in the PySCF program package," *The Journal of Chemical Physics*, vol. 153, no. 2, p. 24109, July 2020, doi: 10.1063/5.0006074.

[17] Q. Sun, "Libcint: An efficient general integral library for Gaussian basis functions," *Journal of Computational Chemistry*, vol. 36, no. 22, pp. 1664–1671, 2015, doi: 10.1002/jcc.23981.

[18] R. D. Bardo and K. Ruedenberg, "Even-tempered atomic orbitals. VI. Optimal orbital exponents and optimal contractions of Gaussian primitives for hydrogen, carbon, and oxygen in molecules," *The Journal of Chemical Physics*, vol. 60, no. 3, pp. 918–931, 1974, doi: 10.1063/1.1681168.

[19] L. A. Constantin, E. Fabiano, S. Laricchia, and F. Della Sala, "Semiclassical Neutral Atom as a Reference System in Density Functional Theory," *Phys. Rev. Lett.*, vol. 106, no. 18, p. 186406, May 2011, doi: 10.1103/PhysRevLett.106.186406.

[20] C. Ying *et al.*, "Do Transformers Really Perform Bad for Graph Representation?." [Online]. Available: https://arxiv.org/abs/2106.05234

[21] Y. Shi *et al.*, "Benchmarking Graphormer on Large-Scale Molecular Modeling Datasets." [Online]. Available: https://arxiv.org/abs/2203.04810

[22] P. Lippmann, G. Gerhartz, R. Remme, and F. A. Hamprecht, "Beyond Canonicalization: How Tensorial Messages Improve Equivariant Message Passing." [Online]. Available: https://arxiv.org/abs/2405.15389

[23] W. Kohn, "Density Functional and Density Matrix Method Scaling Linearly with the Number of Atoms," *Phys. Rev. Lett.*, vol. 76, no. 17, pp. 3168–3171, Apr. 1996, doi: 10.1103/PhysRevLett.76.3168.

[24] Y.-L. Liao and T. Smidt, "Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs." [Online]. Available: https://arxiv.org/abs/2206.11990

[25] Y.-L. Liao, B. Wood, A. Das, and T. Smidt, "EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations." [Online]. Available: https://arxiv.org/abs/2306.12059

[26] J. Ansel *et al.*, "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation," in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*, ACM, Apr. 2024. doi: 10.1145/3620665.3640366.

[27] C. Isert, K. Atz, J. Jiménez-Luna, and G. Schneider, "QMugs, quantum mechanical properties of drug-like molecules," *Scientific Data*, vol. 9, no. 1, p. 273, June 2022, doi: 10.1038/s41597-022-01390-7.

[28] J. Elsborg, L. Thiede, A. Aspuru-Guzik, T. Vegge, and A. Bhowmik, "ELECTRA: A Cartesian Network for 3D Charge Density Prediction with Floating Orbitals," 2026. [Online]. Available: https://arxiv.org/abs/2503.08305

[29] M. V. Klockow, M. K. Ickler, P. Lippmann, and F. A. Hamprecht, "A Function-Centric Graph Neural Network Approach for Predicting Electron Densities," in *The Fourteenth International Conference on Learning Representations*, 2026. [Online]. Available: https://openreview.net/forum?id=HDdkFjFEZd

[30] P. Pulay, "Convergence acceleration of iterative sequences. the case of scf iteration," *Chemical Physics Letters*, 1980, doi: 10.1016/0009-2614(80)80396-4.

[31] L. Devroye, *Non-uniform random variate generation*. Springer, 1986. doi: https://doi.org/10.1007/978-1-4613-8643-8.

A Appendix

A.1 Software Contributions

During the course of this thesis, the author made several contributions to the existing codebase of the OF-DFT implementation by SciAI Lab. The public branch is available on GitHub, but does not reflect all latest changes.

- Implemented L^2 natural loss for usage in DGSP with the Graphormer and Equiformer.
- Added a framework for perturbations around the ground state without having to perturb during Kohn–Sham SCF iterations.
- Added a framework to perturb according to a given probability distribution function (PDF).
- Implemented DGSP-initialized Density Optimization.
- Added a Jupyter Notebook for an in-depth view into how Density Optimization works. This is useful for new users of the codebase.
- Improved the visualization pipeline with Blender's Python API and the Molecular Nodes Add-in to create high-quality renderings of molecular structures and differences between predicted and true ground state densities.
- Uploaded the Structures25 models to HuggingFace for easier access and reproducibility.
- Simplified configuration for running Density Optimization on all molecules.
- Improved terminal feedback during Density Optimization by means of a global progress bar across all processes and threads.
- Updated Python version & software packages, fixed small bugs and pinned dependencies via the Python package manager uv to ensure reproducibility of results.
- Added a Copilot Instructions file to guide the AI assistant in code generation and review.

A.2 Further Proofs

Proof of Theorem 2.10: We make use of the definition of the determinant of an $N \times N$ matrix A with entries $A_{i,j}$ as

$$\det(A) = \sum_{\sigma \in S_N} \left(\operatorname{sgn}(\sigma) \prod_{i=1}^{N} A_{\sigma(i),i} \right) \tag{98}$$

With this, we can write the density $\rho_{\mathbf{\Phi}}(\boldsymbol{r})$ as

$$\rho_{\mathbf{\Phi}}(\boldsymbol{r}) \overset{(6)}{=} N \int_{\mathbb{R}^{3(N-1)}} \left| \psi(\boldsymbol{r}, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N) \right|^2 \, \mathrm{d}\boldsymbol{r}_2 \cdots \mathrm{d}\boldsymbol{r}_N \tag{99}$$

$$= N \int_{\mathbb{R}^{3(N-1)}} \psi(\boldsymbol{r}, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N) \;\; \psi^*(\boldsymbol{r}, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N) \;\, \mathrm{d}\boldsymbol{r}_2 \cdots \mathrm{d}\boldsymbol{r}_N \tag{100}$$

$$\overset{(30)}{=} \frac{N}{N!} \int_{\mathbb{R}^{3(N-1)}} \sum_{\sigma, \tau \in S_N} \operatorname{sgn}(\sigma) \operatorname{sgn}(\tau) \cdot \swarrow \tag{101}$$

$$\phi_{\sigma(1)}(\boldsymbol{r}) \cdots \phi_{\sigma(N)}(\boldsymbol{r}_N) \;\cdot\; \phi^*_{\tau(1)}(\boldsymbol{r}) \cdots \phi^*_{\tau(N)}(\boldsymbol{r}_N) \;\, \mathrm{d}\boldsymbol{r}_2 \cdots \mathrm{d}\boldsymbol{r}_N \tag{102}$$

$$= \frac{1}{(N-1)!} \sum_{\sigma, \tau \in S_N} \operatorname{sgn}(\sigma) \operatorname{sgn}(\tau) \; \phi_{\sigma(1)}(\boldsymbol{r}) \; \phi^*_{\tau(1)}(\boldsymbol{r}) \cdot \swarrow \tag{103}$$

$$\prod_{i=2}^{N} \left(\int_{\mathbb{R}^3} \phi_{\sigma(i)}(\boldsymbol{r}_i) \;\; \phi^*_{\tau(i)}(\boldsymbol{r}_i) \;\, \mathrm{d}\boldsymbol{r}_i \right) \tag{104}$$

$$= \frac{1}{(N-1)!} \sum_{\sigma, \tau \in S_N} \operatorname{sgn}(\sigma) \operatorname{sgn}(\tau) \; \phi_{\sigma(1)}(\boldsymbol{r}) \; \phi^*_{\tau(1)}(\boldsymbol{r}) \prod_{i=2}^{N} \delta_{\sigma(i),\tau(i)} \tag{105}$$

The last step used the orthonormality of the orbitals ϕ_i. Now, we have a double sum over permutations (σ, τ). In each summand, we have the product $\prod_{i=2}^{N} \delta_{\sigma(i),\tau(i)}$ that is only non-zero if $\sigma(i) = \tau(i)$ for all $i \in \{2, ..., N\}$. This means that σ and τ must be identical on the set $\{2, ..., N\}$. Therefore, we can replace the sum over σ and τ with a simple sum only over σ:

$$\rho_{\mathbf{\Phi}}(\boldsymbol{r}) = \frac{1}{(N-1)!} \sum_{\sigma \in S_N} \overbrace{\operatorname{sgn}(\sigma) \operatorname{sgn}(\sigma)}^{1} \; \phi_{\sigma(1)}(\boldsymbol{r}) \; \phi^*_{\sigma(1)}(\boldsymbol{r}) \tag{106}$$

$$= \frac{1}{(N-1)!} \sum_{\sigma \in S_N} \left| \phi_{\sigma(1)}(\boldsymbol{r}) \right|^2 \tag{107}$$

The symmetric group S_N has $N!$ elements. Exactly $(N-1)!$ of them have $\sigma(1) = i$ since, once $\sigma(1)$ is fixed, the remaining $N-1$ elements can be permuted arbitrarily. With this, we obtain the desired result

$$\rho_{\mathbf{\Phi}}(\boldsymbol{r}) = \frac{(N-1)!}{(N-1)!} \sum_{i=1}^{N} |\phi_i(\boldsymbol{r})|^2 = \sum_{i=1}^{N} |\phi_i(\boldsymbol{r})|^2 \tag{108}$$

■