(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 February 2025 (20.02.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/035285 A1

(51) International Patent Classification:
G16C 10/00 (2019.01)

(21) International Application Number:
PCT/CN2023/112628

(22) International Filing Date:
11 August 2023 (11.08.2023)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, Washington 98052 (US).

(72) Inventor; and

(71) Applicant *(for SC only)*: **LIU, Chang** [CN/CN]; One Microsoft Way, Redmond, Washington 98052 (US).

(72) Inventors: **ZHENG, Shuxin**; One Microsoft Way, Redmond, Washington 98052 (US). **SHAO, Bin**; One Microsoft Way, Redmond, Washington 98052 (US). **LIU, Tieyan**; One Microsoft Way, Redmond, Washington 98052 (US).

(74) Agent: **SHIHUI PARTNERS**; 42/F, Tower C, Beijing Yintai Centre, No. 2 Jianguomenwai Avenue, Chaoyang District, Beijing 100022 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) **Title:** KINETIC ENERGY DENSITY FUNCTIONAL MACHINE LEARNING MODEL

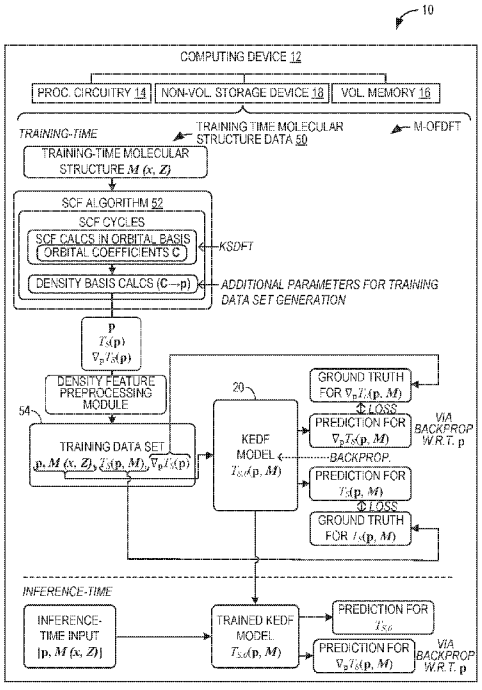


FIG. 8

(57) **Abstract:** A computing system is provided including processing circuitry configured to receive inference-time molecular structure data for an inference-time molecular structure (I), including inference-time atomic basis density expansion coefficients p, and data indicating atomic attributes for each atom in the inference-time molecular structure (I), input the inference-time molecular structure data to a trained kinetic energy density functional machine learning model $T_{S,\theta}$ (p, (I)) to thereby generate a predicted value for electronic non-interacting kinetic energy T_S (p, (I)) of a density function ρ (r), wherein the trained machine learning model has been trained on a training data set that includes, as input, atomic attributes for each atom in a training-time molecular structure (I), and atomic basis density expansion coefficients ρ for the training-time molecular structure (I), and corresponding ground truth values for non-interacting kinetic energy T_S (p, (I)), and output the predicted value for non-interacting kinetic energy T_S (p, (I)) for the inference-time molecular structure (I).

[Continued on next page]

WO 2025/035285 A1

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

KINETIC ENERGY DENSITY FUNCTIONAL MACHINE LEARNING MODEL

BACKGROUND

[0001] 1 Introduction

[0002] Density functional theory (DFT) is a quantum chemistry method for solving for molecular energy and properties. It is widely adopted due to its widely desirable accuracy-efficiency trade-off, and has fostered many scientific discoveries. In the common Kohn-Sham formulation to solve a molecular system with N electrons, by maintaining N one-electron wavefunctions $\{\phi_i(\mathbf{r})\}_{i=1}^{N}$ known as orbitals, the majority of kinetic energy can be covered explicitly, and after also covering the majority of internal potential energy explicitly with the classical counterpart (i.e. Hartree energy), the remaining unknown portion, known as the exchange-correlation (XC) energy, only makes up a small portion and is thus easier to approximate. Nevertheless, optimizing N orbitals deviates from the original idea of DFT to optimize the one-electron density , and this immediately introduces an order-N more computation complexity (Fig. 1A). This is increasingly undesired in the current age when simulating large-scale systems for applications in practice is eagerly desired. For this reason, there is an increasing interest in methods that follow the original spirit of DFT, now called orbital-free DFT (OFDFT).

[0003] Given well-developed XC functionals, the central task in OFDFT is to approximate the non-interacting kinetic energy (the kinetic energy computed from orbitals) as a density functional (KEDF) $T_S[\rho]$. Classical approximations take local density value or semi-local density features (derivatives of the density) to compute the energy. Considering the limited accuracy and fewer semi-local features (e.g., kinetic and exchange energy densities) in OFDFT, explicit nonlocal calculation

(convolution/double integral) is found indispensable and modern developments mostly follow this form. Parameters in these formulations are determined by the uniform electron gas (UEG) theory. Many successes have been achieved for material systems, but accuracy is still limited for molecules, mainly due to the electron density in molecules being highly concentrated and far from the case in UEG.

[0004] There are also attempts to use machine learning to approximate KEDF by leveraging data to compensate for the theoretical mismatch. Kernel ridge regression is on known approach, and deep neural networks have also been explored recently. Yet, these methods use a regular grid (or plane-wave basis) to represent density, which is still not suitable enough for molecule systems as the steep density change near atoms requires a high resolution but at the cost of unnecessarily many grid points in low-density regions. Even using irregular grids commonly needs a number of grids in the tens of thousands of times the number of electrons. This is especially unaffordable in nonlocal calculation. Moreover, few of these machine learning methods showed extrapolation of their performance to molecules far larger than those in training data (out-of-distribution generalization). Extrapolation to larger molecules for such machine learning methods is a significant technical challenge for OFDFT.

SUMMARY

[0005] To address these issues and other issues discussed herein, a computing system is provided including processing circuitry configured to receive inference-time molecular structure data for an inference-time molecular structure, including inference-time atomic basis density expansion coefficients, and data indicating atomic attributes for each atom in the inference-time molecular structure, input the inference-time molecular structure data to a trained kinetic energy density functional machine learning

model to thereby generate a predicted value for electronic non-interacting kinetic energy of a density function, wherein the trained machine learning model has been trained on a training data set that includes, as input, atomic attributes for each atom in a training-time molecular structure, and atomic basis density expansion coefficients for the training-time molecular structure and corresponding ground truth values for non-interacting kinetic energy, and output the predicted value for non-interacting kinetic energy for the inference-time molecular structure.

[0006] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to implementations that solve any or all disadvantages noted in any part of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Fig. 1A shows a schematic overview of two conventional DFT approaches, and their orders of complexity.

[0008] Fig. 1B shows a schematic view of a computing system configured with a trained kinetic energy density function (KEDF) machine learning model that, at inference time, receives inference-time molecular structure data and outputs a prediction of non-interaction kinetic energy, according to one example of the present disclosure.

[0009] Fig. 1C shows a view illustrating how the KEDF machine learning model is trained using density coefficients and the kinetic energy and gradient computed from each of a plurality of self-constrained field (SCF) approximation cycles,

to predict non-interacting kinetic energy that can be used to compute ground-state energy of the molecular system.

[0010] Fig. 2 at (a) – (c) shows three charts that illustrate comparisons of the predictions of the KEDF machine learning model to KSDFT calculations for several test datasets.

[0011] Fig. 3 at (a) – (c) shows three charts that illustrate the calculation efficiency and extrapolation ability of the trained KEDF machine learning model as used to make OFDFT predictions, in a system referred to as M-OFDFT, as compared to conventional KSDFT.

[0012] Fig. 4 at (a) – (d) shows four charts that illustrate the performance of the trained KEDF model in M-OFDFT vs. other conventional approaches, at different predictive tasks for the to the 10-residue Chignolin protein.

[0013] Fig. 5 shows density optimization curves for the trained KEDF model in M-OFDFT as applied to a QM9 molecule, with different initializations functions for comparison.

[0014] Fig. 6 shows a schematic view of the transformer neural network architecture of the KEDF machine learning model, referred to as the Graphormer architecture.

[0015] Fig. 7 illustrates concatenation of auxiliary basis for different atom types, in the KEDF machine learning model, according to one example.

[0016] Fig. 8 illustrates a schematic view of the computing system of FIG. 1B, performing generation of the training data set and training of the KEDF machine learning model, at training time, and outputting of the trained KEDF machine learning model for use at inference time.

[0017] Figs. 9A – 9C illustrate a method for using a kinetic energy density function (KEDF) machine learning model at inference time and training time, according to one example of the present disclosure.

[0018] Fig. 10 includes graphs showing coefficient scale changes after the local frame transformation. For each coefficient dimension μ, the height of the depicted bar represents the relative change in scale, with negative values indicating a reduction in the scale of the coefficient due to the local frame transformation. These graphs demonstrate that the local frame substantially decreases the coefficient scale across diverse basis dimensions and atom types.

[0019] Fig. 11 includes graphs showing gradient scale changes after the local frame transformation. For each coefficient dimension μ, the height of the bar represents the relative change in gradient scale. In the graphs, the local frame transformation notably reduces the gradient scale across a wide range of basis dimensions and atom types.

[0020] Fig. 12 includes graphs that show the gradient and density coefficient scale of the QM9 dataset. The $L\infty$ metric is used to measure the maximum scale of data. (a)-(b) present the gradient scale before and after the dimension-wise rescaling transformation. This technique can significantly reduce the gradient scale across various coefficient dimensions. (c) shows the coefficient scale after the dimension-wise rescaling transformation.

[0021] Fig. 13 is a graph showing gradient scale changes without being centralized by the atomic reference model, demonstrating that the atomic reference model can considerably reduce the gradient scale.

[0022] Fig. 14 includes a pair of graphs that show curves of total energy and gradient norm during density optimization for a randomly selected molecule from the

QM9 dataset. In Fig. 14, 1stLocGradNorm denotes the step of the first local minima of gradient norm, 1stLocEngUpd denotes the step of the first local minima of single-step energy update and GlobEngUpd denotes the step of the global minima of single-step energy update. E, E⋆ and ‖∇pE‖ denote optimized total energy, ground-truth total energy and the norm of energy gradient, respectively. 1stLocGradNorm denotes the step of the first local minima of gradient norm, 1stLocEngUpd denotes the step of the first local minima of single-step energy update and GlobEngUpd denotes the step of the global minima of single-step energy update. E, E⋆ and ‖∇pE‖ denote optimized total energy, ground-truth total energy and the norm of energy gradient, respectively.

[0023] Fig. 15 is a graph showing composition ratios of different QMugs bins.

[0024] Fig. 16 includes two graphs showing additional visualizations of the optimized density around different atoms in an ethanol molecule.

[0025] Fig. 17 includes a pair of graphs showing HF force prediction accuracy on two extrapolation settings.

[0026] Fig. 18 includes a pair of graphs showing ablation study results for non-locality of the model architecture, wherein graph (a) shows energy prediction accuracy on test ethanol molecules, and graph (b) shows HF force prediction accuracy on test ethanol molecules.

[0027] Fig. 19 shows a schematic view of an example computing environment in which the computing systems and methods shown in Figs. 1B-18 may be enacted.

DETAILED DESCRIPTION

[0028] Herein, a computing system and method for OFDFT is described that is referred to as M-OFDFT (M stands for molecules), and which is particularly suited for such molecules using a KEDF machine learning model. For an efficient density

representation for molecules, expansion coefficients on an atomic basis are used in the KEDF machine learning model. An atomic basis naturally fits the pattern of density in molecules because both concentrate around atoms, as shown in Fig. 1B. This alignment allows a much smaller representation dimension, where commonly only tens of the number of electrons suffices for the number of bases. The KEDF model receives input of the species (e.g., data indicating the atomic number and number of electrons) and positions of atoms, i.e., the structure of the molecule, together with the density coefficients attributed to each atom. As shown in Fig. 1B The KEDF model is implemented using a transformer-based neural network architecture (referred to herein as Graphormer) which makes an explicit nonlocal calculation, and has the potential for better generalization and extrapolation with more data and parameters than conventional models. Graphormer is described with reference to Fig. 6 and in Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? Advances in Neural Information Processing Systems, 34:28877–28888, 2021, the disclosure of which is herein incorporated by reference.

[0029] Learning a density functional is a more challenging task than conventional machine learning. The model is used to construct an objective function to optimize density coefficients, so it needs to know the fine energy landscape and keep the optimization on track, as merely fitting point-to-point data does not achieve sufficiently accurate results. As shown in Fig. 1C, to address this challenge, the inventors of the subject application reviewed the data generation process and managed to produce multiple coefficient data points for each molecular structure, and also provide a gradient label for each data point. As explained in further detail below, they also chose proper density initializations to ensure the optimization stayed in regions

where the model is reliable. Optimizing the training objective is also challenging due to the geometric variability of input data and the vast range of gradient data. The inventors correspondingly used local frames to orient the atomic basis to reduce the geometry variability and meanwhile enhance chemical environment invariance, and introduced a series of techniques to enable effective optimization.

[0030] The practical utility and attraction of the computing systems and methods described herein including the KEDF machine learning model and M-OFDFT is demonstrated herein in the following ways. First, M-OFDFT achieves an accuracy never seen by OFDFT with classical KEDFs on a range of molecular systems, including those as large as a protein. Chemical accuracy is achieved on in-scale test molecules, and on larger-scale molecules, the error is still much smaller than that from classical KEDFs. The optimized density also shows a clear shell structure, which is deemed hard for OFDFT. This suggests the functional enables a working OFDFT for molecular systems. Second, M-OFDFT indeed shows a lower empirical time scaling of $O(N^{1.46})$ than that $O(N^{2.49})$ of KSDFT. The absolute time is always smaller, particularly, achieving a 5.7-fold speedup on Chignolin (684 electrons), and a 27.4-fold speedup on the protein B system (2,750 electrons) where KSDFT is around the margin of affordability. Third, an extrapolation study has been shown to larger-scale systems that other machine-learned functionals have seldom achieved. Here, the functional achieves a non-increasing per-atom energy error with the increase of number of heavy atoms in molecule (exponent -0.11), which is seldom seen in direct energy prediction (exponent 0.235 using the same architecture). This demonstrates the merit of leveraging a more fundamental formulation to use machine learning for molecular sciences. The better extrapolation of learning an objective than learning an end-to-end map also aligns with the observation in machine learning. In view of these benefits, the computing systems

and methods described herein push forward the accuracy-scalability trade-off frontier in quantum chemistry, and could benefit various molecular science and engineering problems.

[0031] Turning now to a detailed description of Fig. 1B, a computing system 10 is illustrated which comprises a computing device 12. Computing device 12 comprises processing circuitry 14, associated volatile memory 16, and non-volatile storage device 18 linked by a communications bus. The non-volatile storage device stores instructions, that when executed using parts of the memory 16 by the processing circuitry 14, causes the processing circuitry 14 to perform the functions described herein. In this way, processing circuitry 14 is configured to perform these functions.

[0032] As shown, at inference the processing circuitry 14 is configured to receive inference-time molecular data 20 including an inference-time molecular structure $\mathcal{M}$ and inference-time atomic basis density expansion coefficients $\mathbf{p}$. The inference time molecular structure $\mathcal{M}$ is a data structure that includes data that structurally describes a physical molecular system 22. More specifically, inference-time molecular data 20 includes data indicating atomic attributes of each atom $\mathbf{Z}$ in the inference-time molecular structure $\mathcal{M}$. For example, the atomic attributes may be one or more of atomic positions $\mathbf{X}$ and atomic number $\mathbf{Z}$. Any suitable encoding for the data may be used to indicate the atomic positions $\mathbf{X}$ and atomic number $\mathbf{Z}$. The atomic basis density expansion coefficients $\mathbf{p}$ represent the electrical density functional ρ® as the expansion coefficients under atomic basis, as shown. These density expansion coefficients $\mathbf{p}$ may simply be referred to as density coefficients, and may be supplied after preprocessing by a density feature preprocessing module described below.

[0033] The processing circuitry 14 is further configured to input the inference-time molecular structure data 20 to a trained kinetic energy density functional (KEDF)

machine learning (ML) model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ 24 to thereby generate a predicted value for electronic non-interacting kinetic energy $T_{S,\theta}$ 26 of a density function $\boldsymbol{\rho}(\mathrm{r})$, which can be based on or determined by atomic basis density expansion coefficients **p**. As described below in more detail with reference to FIG. 8, for example, the trained KEDF ML model (or KEDF model for simplicity) has been trained on a training data set that includes, as input, atomic attributes, such as one or more of atomic positions **X** and/or atomic numbers **Z**, of each atom in a training-time molecular structure $\mathcal{M}$, and atomic basis density expansion coefficients **p** for the training-time molecular structure $\mathcal{M}$, the atomic basis density expansion coefficients **p** indicating values by which electron charge density is represented on an atomic basis, and that includes corresponding ground truth values for non-interacting kinetic energy $T_S(\mathbf{p}, \mathcal{M})$. The training data set can also include ground truth values for the kinetic energy density expansion gradient $\nabla_{\mathbf{p}} T_S(\mathbf{p}, \mathcal{M})$. In one example configuration the trained KEDF ML model 20 includes a neural network 28, such as the transformer-based neural network shown in FIG. 8. An output layer 30 is provided to sum the energy contributions of each atom in the molecular structure $\mathcal{M}$.

[0034] The processing circuitry is further configured to output the predicted value for non-interacting kinetic energy $T_{S,\theta}$ 26 for the inference-time molecular structure $\mathcal{M}$ 20. The processing circuitry may further be configured to output a predicted value for the kinetic energy density expansion gradient $\nabla_{\mathbf{p}} T_S(\mathbf{p}, \mathcal{M})$ through backpropagation for the inference-time molecular structure $\mathcal{M}$. The processing circuitry may further be configured to perform an orbital-free Density Functional Theory (DFT) calculation 32 using the predicted value of the kinetic energy density expansion gradient. An example form of an OFDFT calculation for determining ground state energy is shown in FIG. 2, and further described below.

[0035] 2 Results

[0036] 2.1 The Orbital-Free Density Functional Theory for Molecules

[0037] In OFDFT, the energy functional is decomposed in the same way as in KSDFT: $E[\rho] = T_S[\rho] + E_H[\rho] + E_{XC}[\rho] + E_{ext}[\rho]$, where $E_{ext}[\rho] := \int \rho(r)V_{ext}(r)\,dr$ is the external potential energy from the nuclei in the molecule, $E_H[\rho] := \frac{1}{2}\int \frac{\rho(r)\rho(r')}{\|r-r'\|}\,drdr'$ is the classical internal potential energy (Hartree energy) that covers the major part of the electric interaction energy among electrons, $T_S[\rho]$ is the non-interacting part of the kinetic energy, and the exchange-correlation (XC) energy $E_{XC}[\rho]$ accounts for the rest of the kinetic and internal potential energy, and accurate explicit approximations are available. In KSDFT, the non-interacting kinetic energy can be directly calculated from orbitals. But as a density functional, its expression is unavailable. Hence, the focus in OFDFT is to develop an accurate approximation to the KEDF $T_S[\rho]$.

[0038] As mentioned, M-OFDFT adopts atomic bases (atom-centered contracted Gaussians with spherical harmonics) for an efficient density representation in molecules, which concentrates on atoms around which the density is distributed, as shown in FIG. 1B. They are even designed to mimic the nuclear cusp condition for accurately describing the sharp change near the nuclei, which is hard for grid or plane-wave bases. As a result, the commonly required number of atomic bases is much fewer (~103) than the number of grids, which is highly favorable especially for nonlocal calculation indispensable for KEDF. Moreover, atomic bases often come in a shell structure, i.e., basis functions concentrate on different radii from the nucleus. This helps the OFDFT method to easier find the shell structure of density in molecules, which is deemed hard without orbitals. Under an atomic basis $\{\omega_\mu(r)\}_{\mu=1}^{M}$, the density can be expanded as: $\rho(r) = \sum_\mu p_\mu \omega_\mu(r)$. As the atomic basis depends on the structure $\mathcal{M}$:

$= \{X, Z\}$ of the molecule, i.e., the positions $X := (x^{(1)}, \cdots, x^{(A)})$ (conformation) and atomic numbers $Z := (Z^{(1)}, \cdots, Z^{(A)})$ of all atoms (constitution) in the molecule, the density is fully described by the tuple $(p, \mathcal{M})$. The KEDF model hence follows the form $T_S(p, \mathcal{M})$, which directly outputs the integrated value instead of kinetic-energy density on grids.

[0039] Fig. 1B and C show an overview of the computing system of the present disclosure (also referred to as M-OFDFT), while Fig. 1A shows conventional approaches. As shown in Fig. 1A, conventional KSDFT solves the Schrödinger equation by maintaining N one-electron wavefunctions, while OFDFT relies on only a single one-electron density function, reducing the computational complexity by an order of N. At Fig. 1B, the computing system 10 is shown using the expansion coefficients **p** on an atomic basis to represent electron density. The KEDF model based on the non-local Graphormer architecture takes as inputs density coefficient features (or simply density coefficients) **p** and molecular structure $\boldsymbol{M}$ to predict kinetic energy $T_{S,\theta}$. In Fig. 1C, the trained KEDF model is shown being used as the objective function for optimizing the density coefficients **p**.

[0040] A machine-learning model for $T_S(p, \mathcal{M})$ is employed and trained on data, due to lack of instructive theory for KEDF far from uniform. For relevance in deployment, data are generated on a certain set of molecular structures. However, learning a density functional is more challenging than conventional machine training, since the trained KEDF model is used as the objective function for optimizing the density coefficient **p** for a given molecular structure, as shown in Fig. 1C. To capture the function landscape on the coefficient space of **p** for each molecular structure $\mathcal{M}$, the inventors managed the KSDFT calculation process to generate multiple **p** samples for each molecular structure $\mathcal{M}$ and also calculate the gradient label $\nabla_p T_S$, leading to

data following the pattern $\left\{\mathcal{M}^{(d)}, \left\{p^{(d,k)}, T_S^{(d,k)}, \nabla_p T_S^{(d,k)}\right\}_k\right\}_d$. Also introduced herein are some techniques to handle geometric variability and the vast range of gradient data for effective training.

[0041] For the architecture of the $T_S, \theta(p, \mathcal{M})$ model, noting that the coefficients in **p** can be attributed to each atom according to the center of the corresponding basis function, the input $(p, \mathcal{M})$ can be seen in a graph structure, where the atom type and coefficients define the node feature, and the relative location of an atom pair defines the edge feature, as shown in Fig. 1B. The inventors hence use a graph neural network based on the Graphormer model as $T_S(p, \mathcal{M})$, as shown in Figs. 1B, 6 and 8. The nonlocality is covered by the self-attention layer from the transformer architecture of the KEDF model, which explicitly computes the interaction between every pair of node features. To better accommodate the graph structure, edge features are also involved in the corresponding node pair interaction, which is shown to improve graph-theoretical performance. The interaction results are used to update node features, and after several layers of such updates, the T_S output is given by aggregating the final node features. According to an ablation study that was conducted, the nonlocal formulation is crucial for the desired extrapolation performance of decreasing the per-atom error as system scale increases.

[0042] After the $T_{S,\theta}(p, \mathcal{M})$ model is trained, the ground-state energy and density of a given molecular structure $\mathcal{M}$ can be solved from the density optimization procedure (See Fig. 1C):

$$\min_{p:\, normalized} E_\theta(p, \mathcal{M}) := T_{S,\theta}(p, \mathcal{M}) + E_H(p, \mathcal{M}) + E_{XC}(p, \mathcal{M}) + E_{ext}(p, \mathcal{M}).$$

[0043] Here, E_H and E_{ext} can be explicitly computed from $(p, \mathcal{M})$, and E_{XC} can be computed on grids in a conventional manner. The optimization is solved by

gradient descent as self-consistent iteration is unnatural for the density. In the optimization, the normalization constraint, $\int \rho(r)\, dr = p^{\top} w = N$ where $w_{\mu} := \int \omega_{\mu}(r)\, dr$, is linear, so the gradient is projected onto the admissible plane before updating the coefficients:

$$p^{(k+1)} := p^{(k)} + \varepsilon \left(I - \frac{w w^{\top}}{w^{\top} w} \right) \nabla_p E_{\theta}(p^{(k)}, \mathcal{M}), \qquad (1)$$

[0044] where ε is a step size. Notably, due to directly operating on density, the complexities to calculate E_{ext}, E_{XC}, and E_H terms are square-rooted over those in KSDFT, resulting in $O(N)$, $O(N)$ (for semi-local pure XC functionals) and $O(N^2)$ complexities. The $T_{S,\theta}(p, \mathcal{M})$ model has $O(N^2)$ complexity, so the complexity for the OFDFT of the present disclosure in each optimization iteration is $O(N^2)$. This achieves order-N less cost than KSDFT, which is typically $O(N^3)$.

[0045] 2.2 M-OFDFT Achieves Chemical Accuracy on Molecular Systems

[0046] The inventors studied the performance of M-OFDFT on molecules with similar sizes as those in the training data. Two cases were considered for this study. In the first, data was generated on a set of ethanol conformations from the MD17 dataset, and in the second, on molecules in equilibrium conformation from the QM9 dataset. The former studies the generalization in conformation space while the latter in the chemical space. Each dataset was split into three parts for the training and validation of the KEDF model and the test of M-OFDFT using the trained model. For ease of training, the APBE KEDF was used as a reference and the Graphormer model (See Fig. 6 for details) was allowed to learn the residue.

[0047] M-OFDFT was evaluated in terms of the mean absolute error (MAE) in energy, as well as in the Hellmann-Feynman (HF) force. The results are 0.18 kcal/mol and 1.18 kcal/mol/Å on ethanols, and 0.93 kcal/mol and 2.91 kcal/mol/Å on QM9. M-OFDFT achieves chemical accuracy (1 kcal/mol energy MAE) in both cases.

[0048] To show the significance of this result for OFDFT on molecules, the performance is compared with classical KEDFs, including Thomas-Fermi (TF) KEDF which is exact in the uniform electron gas limit, its correction with density gradient $TF + \frac{1}{9}vW$ where vW represents the von Weizsäcker KEDF, an empirical variant TF+vW, and the reference method APBE. Noting that different KEDFs may have different absolute energy biases, it is required to compare the MAE in relative energy. On the ethanol dataset, a relative energy is taken between a conformation and the equilibrium conformation. On QM9, as each molecule only has one conformation, a relative energy is taken between a pair from the 6,095 isomers of $C_7H_{10}O_2$ in the QM9 dataset, which can be seen as different conformations of the same "molecule." Fig. 2 shows the model predictions on the test datasets compared to KSDFT calculations. Fig. 2 at (a)-(b) shows a comparison in energy and HF force prediction of M-OFDFT with classical KEDFs (with 95% confidence interval bars). Fig. 2 at (c) shows a visualization of densities optimized by various methods. The curves plot the integrated density on spheres with varying radii, centered at the oxygen atom in an ethanol molecule. As shown in Fig. 2 at (a) and (b), M-OFDFT achieves chemical accuracy on relative energy, and is two orders more accurate than OFDFT using classical KEDFs.

[0049] As a qualitative investigation of M-OFDFT, the optimized density on ethanol by these methods is visualized in Fig. 2 at (c). The major peaks on the left and right show the density of core electrons of the oxygen and the bonded carbon, while the minor peak in the middle reflects the density of electrons in the covalent bonds to the hydrogen and the carbon. M-OFDFT accurately recovers this shell structure, which is deemed difficult for OFDFT, and the optimized density almost coincides with the density by KSDFT. In comparison, the classical KEDF of APBE does not align well

with the true density around the covalent bonds. These results suggest that the inventors have developed a working OFDFT for molecular systems.

[0050] 2.3 M-ODFT Has Lower Empirical Time Complexity than KSDFT

[0051] As mentioned, when using atomic basis to represent density, OFDFT achieves $O(N^2)$ time complexity, which is lower than that $O(N^3)$ of KSDFT. This advantage of M-OFDFT is empirically verified in Fig. 3 at (a). Both M-OFDFT and KSDFT are tested on a subset of molecular structures from the QMugs dataset with 106 to 696 electrons. The running time values are grouped and averaged within each bin of the number of electrons of width 20. The absolute running time of M-OFDFT is always shorter than that of KSDFT, and achieves up to 6.7-fold speedup over KSDFT. The complexity of M-OFDFT is lower than $O(N^2)$, and is indeed order-N less than that of KSDFT.

[0052] With sophisticated atomic orbitals and numerical optimization, KSDFT can typically handle molecular systems of up to approximately 500 atoms using a modern PC cluster. To demonstrate the scaling advantage of M-OFDFT, the model was tested on two large-scale systems that are seldom encountered in the context of KSDFT calculations: (1) the 47-residue (709 atoms) peripheral subunit-binding domain BBL-H142W (PDB ID: 2WXC), and (2) the 47-residue (738 atoms) K5I/K39V double mutant of the Albumin binding domain of protein B (PDB ID: 1PRB). The all-atom geometry conformations for each system are extracted for evaluation from Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw, How fast-folding proteins fold, Science, 334(6055):517–520, 2011. The results indicate that M-OFDFT requires 0.41 hours and 0.45 hours to calculate the two test systems, respectively, achieving 25.6-fold and 27.4-fold speedup compared to KSDFT calculations. The PySCF software was used for KSDFT calculations and the default setting was adopted.

[0053] 2.4 M-OFDFT Extrapolates Well to Larger Scale Molecules

[0054] OFDFT has the advantage of handling large systems with affordable cost, for which other quantum chemistry methods such as KSDFT become sharply cost-demanding. But due to the same reason, data are scarce for such systems. Accordingly, a machine-learning-based OFDFT method that can extrapolate to molecules in larger scales than seen in training (out-of-distribution generalization) is generally desirable, though technically challenge to achieve. In conducting such a study on M-OFDFT, the KEDF ML model (Graphormer model – see Fig. 8 for details) was allowed to learn the sum of KEDF and the XC energy altogether. This is to completely get rid of calculation on grids (for the APBE KEDF reference and the PBE XC functional), whose additional $O(MN_{Grid})$ memory cost in gradient descent becomes unaffordable for large molecules. It is noted that such a modification does not lead to obvious accuracy loss in the in-scale study.

[0055] To evaluate the significance of the extrapolation behavior, the error of M-OFDFT was compared with a natural variant of machine-learning-based quantum chemistry method that directly predicts the ground-state energy from the molecular structure $\mathcal{M}$ in an end-to-end manner. This is referred to herein as M-PES, representing the potential energy surface for molecules, and uses the same Graphormer model architecture for a fair comparison. To study the effect of density feature on extrapolation, the inventors also considered a variant that has the MINAO density feature in its input in addition to $\mathcal{M}$, which is called M-PES-Den herein.

[0056] Fig. 3 shows calculation efficiency and extrapolation ability of M-OFDFT. In Fig. 3 at (a), an empirical time cost of M-OFDFT and KSDFT for molecules in various scales is shown. At (b) and (c), the extrapolation performance on QMugs is shown. The shading in Fig. 3 represents a 95% confidence interval. At (b), all models

have been trained on molecules with less than 15 heavy atoms, and extrapolated to larger QMugs systems with increasing molecule scale. At (c), all models have been trained on a suite of QMugs datasets with increasing molecule scale (up to 30 heavy atoms), and extrapolated to 50 QMugs conformations with 50-60 heavy atoms. The black dotted line denotes the performance of M-OFDFT trained on the first QMugs dataset.

[0057] The inventors studied the extrapolation on the QMugs dataset, which contains molecules much larger than those in QM9 that only have no more than 9 heavy atoms. To construct an extrapolation task, the KEDF model was trained on QMugs molecules with no more than 15 heavy atoms. As QMugs molecules with no more than 9 heavy atoms are few, all the QM9 molecules were also included in training. To study the trend of extrapolation, the rest of the QMugs molecules were grouped according to their number of heavy atoms into bins of width 5, and the per-atom energy MAE of M-OFDFT was evaluated on 50 molecular structures from each bin with increasing scale.

[0058] The result is shown in Fig. 3 at (b). The per-atom MAE by M-OFDFT is not only orders smaller in absolute value than the end-to-end counterparts M-PES and M-PES-Den, but also keeps non-increasing and even decreases (the negative exponent) with increasing molecule scale. This is beyond the expectation on the common local-structure-based energy prediction models, since the error in each local environment would not be reduced by a larger global environment, and almost always, the prediction on each local environment cannot cover the effect from global interaction, which becomes stronger as the molecule increases, and hence increases the per-atom error when amortized. The non-increasing per-atom MAE is not achieved by directly using a nonlocal model in end-to-end energy prediction as shown by the M-PES curve,

and the similar increasing trend of the M-PES-Den curve indicates that even augmenting with a density feature does not trigger the non-increasing trend.

[0059] With this evidence, the inventors postulate that the qualitatively better extrapolation performance comes from modeling a density functional as the objective function of the target density and energy, instead of directly modeling the map to the targets. In this way, part of the complexity in the map is transferred to the optimization process, while the objective function therein may be simpler and easier to extrapolate. Particularly for the KEDF model, its dependency on molecular structure $\mathcal{M}$ is only for locating and characterizing the basis of density, but not for the result of the entire interaction process among electrons and atoms, which appears very complicated and subtle if hiding the electronic descriptor. Notably, such a conclusion has also been observed in machine learning, indicating a potentially general conclusion of better extrapolation from modeling the objective function of the target.

[0060] To further substantiate the significance of the extrapolation capability of M-OFDFT, the amount of additional large-scale molecular data required by end-to-end counterparts to achieve extrapolation performance comparable to that of M-OFDFT was investigated. For this purpose, 50 QMugs molecules with 50-60 heavy atoms were selected as the extrapolation benchmark and the models were trained on a suite of molecular datasets with increasing molecular size. Specifically, the QMugs molecules with no more than 30 heavy atoms were grouped into bins with a width of 5 and larger-scale molecules were progressively incorporated into the training set. A similar molecule count was also maintained across all training datasets to avoid the influence of data size. As illustrated in Fig. 3 at (c), both the M-PES and M-PES-Den models required additional molecules with at least twice the larger scale (30 versus 15) to achieve energy prediction accuracy comparable to M-OFDFT (0.068 kcal/mol). These

results supportively demonstrate the superior extrapolation performance of M-OFDFT. Moreover, the high efficiency of M-OFDFT in learning density functionals from accessible molecules is very promising for pushing the quantum chemistry methods to unaffordable large-scale systems.

[0061] In recent years, accelerating first-principle calculations for large-scale biomolecular systems has become an appealing task in quantum chemistry, particularly in the context of protein structures. To assess the generalization capability of M-OFDFT for protein systems, the 10-residue Chignolin protein (comprising 168 atoms) was selected as a model system. This protein has been extensively studied experimentally and has abundant conformational structures derived from molecular dynamics (MD) simulations. For the extrapolation task, the KEDF model was trained on short-peptide fragments obtained from approximately 1,000 Chignolin conformations, each fragment containing no more than five residues. The model was then extrapolated to the same protein conformations. The fragments were constructed by "cutting out" subsequences of the Chignolin sequence, including discontinuous subsequences with residue gaps. They were chosen due to being long enough to sample important long-range effects, but still short enough such that reference KSDFT calculations were accessible in a reasonable time. Both the Chignolin conformations and fragments were neutralized, since the focus was on isolated molecular systems where all atoms or functional groups stay neutral and all electrons are paired. In this task, the external potential energy was also introduced into the learning objective since it can increase the training stability of the KEDF model.

[0062] The results are presented in Fig. 4 at (a). Fig. 4 shows extrapolation of performance on Chignolin. Fig. 4 at (a) Energy prediction accuracy of 1,000 Chignolin conformations of three methods with increasing training scale. In Fig. 4 at (b), relative

energy prediction accuracy of 50 Chignolin conformations of three methods compared to APBE. All deep learning models are trained with all fragments here. Fig. 4 at (c) and (d) generally show finetuning performance on larger-scale molecules. Fig. 4 at (c) shows energy and Fig. 4 at (d) shows Hellmann-Feynman (HF) force prediction accuracy of 50 Chignolin conformations of three methods with/without finetuning models on 500 Chignolin conformations. The error reduction ratios of finetuned models over 'FromScratch' baselines are listed.

[0063] Impressively, M-OFDFT consistently outperforms both M-PES and M-PES-Den in terms of energy prediction accuracy as the scale of training fragments increases. Furthermore, the performance of M-OFDFT, trained with all fragments, was compared to the classical KEDF APBE on 50 random Chignolin conformations. Fig. 4 at (b) demonstrates that the method disclosed herein achieves a significantly lower per-atom relative energy MAE compared to APBE (0.098 kcal/mol vs. 0.684 kcal/mol), highlighting the strong extrapolation capacity of this approach for biomolecular systems.

[0064] Extrapolation to out-of-scale molecules remains a grand challenge for machine learning-driven quantum chemistry methods, especially for large-scale systems with extremely scarce training labels. One potential strategy to improve extrapolation performance is pretraining the models on numerous small molecules that are computationally accessible, and then finetuning the models on a smaller sample set of systems of interest. A technical challenge exists to design an effective pretraining framework that can efficiently exploit small molecules to extract transferrable knowledge. To evaluate such a transfer learning strategy, the extrapolation performance of M-OFDFT and the end-to-end counterparts on 50 random Chignolin conformations were compared after finetuning all models on 500 Chignolin conformations. The

models trained from scratch on the same 500 conformations serve as "FromScratch" baselines. As depicted in Fig. 4 at (c) and (d), the disclosed method achieves greater performance gains over "FromScratch" models in both energy and force accuracy. Notably, M-OFDFT attains a significant 35.4% and 40.8% error reduction for force and energy prediction, respectively. This result implies that the disclosed formulation can exploit the transferrable and physical knowledge from accessible protein fragments more efficiently than the end-to-end learning paradigm.

[0065] 3 Conclusion and Discussion

[0066] Described herein is a computing system and methods referred to as M-OFDFT, an orbital-free density functional theory that works successfully on molecules. The kinetic energy density functional (KEDF) for OFDFT is regarded as challenging to approximate (harder than the exchange-correlation functional), especially for the electronic density space for molecular systems. By leveraging machine learning models such as the Graphormer architecture shown in Fig. 6, such approximation can be achieved much more accurately, as demonstrated on a wide range of molecular systems. With the achievement of working accuracy on molecules, M-OFDFT provides desirable scalability of OFDFT into solving molecular systems. The empirical cost of M-OFDFT is orders lower than that of the popular KSDFT, and the advantage becomes more dominant as the empirical scaling $O(N^{1.46})$ is reduced by more than order-N over KSDFT. Therefore, M-OFDFT is believed to have pushed forward the accuracy-cost trade-off frontier of quantum chemistry.

[0067] The present disclosure also demonstrates the benefit of using an appropriate formulation of quantum chemistry when leveraging machine learning. M-OFDFT has shown qualitatively better larger-scale extrapolation performance than the direct ground-state energy prediction formulation M-PES and M-PES-Den, even

though they also consume $O(N^2)$ complexity and achieve a lower in-scale validation error. M-OFDFT can also provide the ground-state density which enables calculation of various properties. The KEDF model could also be used for constructing quantum embedding for more accurate multi-scale calculation.

[0068] In contrast to existing work on learning KEDF, the density is represented here on atomic basis, which greatly reduces the feature dimension, enabling an efficient nonlocal KEDF model that captures long-range effects to enhance accuracy. Some other approaches to learning the XC functional also adopt atomic basis, but the absence of the molecular structure input disallows the interaction of density components hosted on different atoms. To inform the model of the optimization landscape, the training data is enriched with gradient labels and labels for multiple densities on the same molecular structure. Comparisons of the present approach to other approaches to supervising the optimization behavior of a functional model, show that other approaches are not as effective. Moreover, some other approaches require the projection onto the in-distribution manifold be applied in each step of density optimization for stability, while M-OFDFT only needs the initialization to be in distribution (see Fig. 5), indicating better optimization behavior.

[0069] While statistical guarantees are established for the in-distribution generalization of machine learning models, reliable extrapolation remains an open challenge, which has long hindered the application of machine learning in the science domain. The present disclosure achieves better extrapolation performance over ground-state energy prediction by learning a density functional, whose underlying rule is simpler and more universal, and therefore easier to extrapolate. The density optimization process also takes over a part of the complication in the dependency of ground-state properties on the molecular structure.

[0070] Incorporating exact properties of KEDF into the machine learning model could further improve extrapolation. Geometric invariance is built into the model, which significantly improves the accuracy. Nevertheless, it is not always straightforward to gain benefits from these properties, since some would introduce more training challenges, or require specific architectures that come with unintended capacity restriction. For example, an attempt was made to use the von Weizsäcker KEDF as the reference KEDF which introduces the positivity property to the residual KEDF model $T_{S,\theta,S,res}$, but the range of the resulting gradient labels for $T_{S,\theta,S,res}$ are too large to train an ML model effectively. An attempt was also made to directly learn the KEDF (without a reference) and the universal functional, which have convex surrogates, but the results are far from desirable again due to more challenging training problems. The KEDF also has a scaling property, but it cannot be translated into an exact equation under atomic basis.

[0071] From the machine learning perspective, extrapolation can be further improved with more data and a larger model. Notably, the transformer architecture has enabled foundation models for language, and the Graphormer model itself (see Fig. 6) has shown remarkable extrapolation performance on molecular property prediction. Accordingly, better extrapolation could be expected with a larger dataset and model size, all the way to serve as a foundation model of KEDF.

[0072] 4 Methods

[0073] Outlined below are methods to train and use the KEDF model. As mentioned, learning a density functional faces more challenges beyond conventional machine learning. This comes from two sides: (1) The model is used not as an end-to-end predictor, but as an objective to optimize the density. This requires the model to be aware of the energy landscape on the density-coefficient space, and proper density

optimization methods. These challenges are addressed herein. (2) The characteristics of the target functional itself poses difficulties in model training (e.g., geometric variability and vast gradient range). Such considerations are taken into account in designing the model and developing techniques to address the training difficulties.

[0074] 4.1 Training the KEDF Model

[0075] As a density functional, the KEDF model is used to optimize density coefficient p for a given molecular structure $\mathcal{M}$ (Fig. 1C and FIG. 8), and therefore it needs to know how to vary with p for a fixed $\mathcal{M}$ through training. To achieve this, the training data first needs to provide T_S labels for multiple p values given each $\mathcal{M}$ value, i.e., following the form $\{\mathcal{M}^{(d)}, \{p^{(d,k)}, T_S^{(d,k)}\}_k\}_d$. This can be achieved by leveraging solutions in intermediate steps of the self-consistent field (SCF) iteration in a KSDFT calculation for each molecular structure $\mathcal{M}^{(d)}$. The rationale is that the task in each SCF step k is to solve a non-interacting fermion system in an effective one-body potential, whose ground state can be solved exactly using the orbitals. The non-interacting kinetic energy $T_S^{(d,k)}$ can thus be exactly calculated from these orbitals, and the corresponding density coefficient $p^{(d,k)}$ can be calculated from these orbitals by density fitting. In particular, adopted herein is an even-tempered auxiliary basis with its β parameter taken as 2.5 as the density basis. A well-acknowledged pure XC functional (i.e., only depends on density but not orbitals), e.g., PBE, is used to make sure the data pairs demonstrate a density functional. Restricted-spin calculation is adopted on the 6-31g(2df,p) basis set, which is sufficient for the considered molecules which are uncharged, in near-stable conformations, and only involve light atoms (up to fluorine). The training loss function for the $T_{S,\theta}(p, \mathcal{M})$ model is:

$$\sum_d \sum_k \left| T_{S,\theta}(p^{(d,k)}, \mathcal{M}^{(d)}) - T_S^{(d,k)} \right|. \qquad (2)$$

[0076] However, although the predicted energy by a trained KEDF model is accurate at the ground-state density coefficient, it was found that subsequent density optimization using the model still decreases the energy. This indicates that T_S labels from multiple SCF steps alone do not yet sufficiently demonstrate the energy landscape on the coefficient space for density optimization. With the inventors' realization that the optimization curve should be stationary at the ground-state density, providing further supervision on the gradient $\nabla_p T_S(p, \mathcal{M})$, which provides more information on the energy landscape and directly affects the optimization process, was considered. Fortunately, the gradient label is also available from each SCF step, again due to the solution to each SCF step k on molecular structure $\mathcal{M}^{(d)}$ being the ground state of non-interacting fermions in an effective one-body potential. The fitted density coefficient $p^{(d,k)}$ from the solution then represents the ground-state density of the system, which minimizes the total energy $T_S\left(p, \mathcal{M}^{(d)}\right) + p^\top v_{eff}^{(d,k)}$ subject to the normalization constraint $p^\top w^{(d)} = N$, where $v_{eff}^{(d,k)}$ is the effective one-body potential in vector form under the atomic basis of density.

[0077] This means $\left(I - \frac{w^{(d)} w^{(d)\top}}{w^{(d)\top} w^{(d)}}\right)\left(\nabla_p T_S\left(p^{(d,k)}, \mathcal{M}^{(d)}\right) + v_{eff}^{(d,k)}\right) = 0$, which gives gradient supervision on the subspace of normalized density coefficients. An efficient implementation was made to calculate the unprojected gradient label $v_{eff}^{(d,k)}$ from each SCF step solution. The corresponding gradient loss function is:

$$\Sigma_d \Sigma_k \left\| \left(I - \frac{w^{(d)} w^{(d)\top}}{w^{(d)\top} w^{(d)}}\right)\left(\nabla_p T_{S,\theta}\left(p^{(d,k)}, \mathcal{M}^{(d)}\right) - \nabla_p T_S^{(d,k)}\right) \right\|. \qquad (3)$$

[0078] An ablation study was conducted on the effect of multiple-step data and gradient supervision.

[0079] 4.2 Geometric Invariance

[0080] The second extraordinary challenge beyond conventional machine-learning tasks is that the data shows large variability. In the input feature to the model $T_{S,\theta}(p, \mathcal{M} = \{X, Z\})$, atomic coordinates X require a specific coordinate system, the choice of which introduces an arbitrary variability of the input feature, also known as SE(3) equivariance. Moreover, in addition to the radial component (contracted Gaussians), atomic basis also has an angular component, which depends on the coordinate system. The coefficient feature p then also has this variability. It is then desired that the features or model be invariant with respect to the change of coordinate system, so as to learn the essential dependency on density irrespective of geometric representation.

[0081] For the atomic coordinates X, the Graphormer model is naturally SE(3)-invariant to its shift and rotation, since the model only leverages relative distances of atom pairs for later processing, which is naturally invariant. For the density coefficient p, possible approaches include choosing a standard global coordinate system (e.g., by PCA on the atom positions) or building a model naturally invariant to the shift and rotation of the global coordinate system. But these approaches cannot make an invariant feature for the same pattern of density distribution. For example, consider the coefficients corresponding to the atomic basis on the three hydrogen atoms in a methyl group. As the three H-C bonds are symmetric, the densities on the bonds are the same (ignoring influence from nearby non-bonded atoms) and thus have the same contribution to the kinetic energy from themselves. But as the three bonds have different orientations relative to a common coordinate system, the coefficients are different. To further eliminate this unnecessary variability, adopted herein is a local frame to endow each atom with its own coordinate system for its atomic basis and coefficients, depending on the local chemical environment, basically the bonding

pattern, of the atom. Similar ideas are also considered in many previous attempts, where the x-axis unit vector $\hat{x}$ is pointed from the center atom to the nearest neighbor atom, $\hat{z} := \hat{x} \times (x^{(2)} - x^{(0)})$ where $x^{(2)}$ is the coordinates of the second-nearest atom to this atom $x^{(0)}$ which is not on $\hat{x}$, and $\hat{y} := \hat{z} \times \hat{x}$. Following this, hydrogen atoms in the neighborhood of the center atom are excluded to obtain more stable local structures. Since the local frame is chosen all based on relative positions of the atoms, it is equivariant to the global shift and rotation, making the density coefficient SE(3)-invariant. In addition, the coefficient on the $\hat{x}$ direction reflects the density on the most significant (shortest length) bond at the atom. This leads to the same density coefficients on the three hydrogen atoms in a (perfectly symmetric) methyl group even though the H-C bonds have different orientations, and the change in the coefficient reflects the density change on the H-C bond. Both the scale (standard deviation) of atomic density coefficients and gradient labels are significantly reduced with the use of local frames, especially for H atoms where most of the coefficient dimensions exhibit a > 60% scale reduction ratio. Moreover, this approach immediately reduces the training loss and stabilizes the optimization process, resulting in a significant improvement in the density optimization stage.

[0082] 4.3 Learning with a Vast Gradient Range

[0083] Even with the geometric variability removed, the range of data is still vast. What makes it worse is that conventional data normalization technique is not applicable in this task, since there is also a gradient label, minimizing whose scale is conflicting with minimizing the coefficient scale (dividing a scale to the coefficient means multiplying the scale to the gradient). Having both large coefficients and gradient implies a function with a large Lipschitz coefficient. To handle this challenge, a series of techniques are introduced to enable effective training, including a

reparameterization of the density coefficient, atomic reference model, and dimension-wise rescaling.

[0084] Dimension-wise Rescaling

[0085] First, a numerical treatment of dimension-wise rescaling is considered to allow more flexibility in data normalization. Specifically, for each coefficient dimension, an attempt is made to normalize its gradient data points to a target scale. If the resulting coefficient standard deviation is larger than a chosen maximum scale, then that dimension is rescaled to normalize the coefficients to the maximum scale. The procedure of obtaining the rescaling factor λ_μ^r of dimension μ can be summarized as

$$\lambda_\mu^r = \begin{cases} \min\left\{\frac{\left\|\nabla_{p_\mu} T_S\right\|_\infty}{\bar{s}_g}, \frac{\bar{s}_c}{\sigma(p_\mu)}\right\}, & \left\|\nabla_{p_\mu} T_S\right\|_\infty \geq \bar{s}_g \\ 1, & \left\|\nabla_{p_\mu} T_S\right\|_\infty \leq \bar{s}_g \end{cases} \quad (4)$$

[0086] where $\bar{s}_g$ is the target scale of gradients, $\bar{s}_c$ is the maximum scale of coefficients, $\left\|\nabla_{p_\mu} T_S\right\|_\infty$ is the infinity norm (i.e., L^∞ metric) of gradient and $\sigma(p_\mu)$ is the standard deviation of coefficients. Each coefficient dimension will be rescaled by $p'_\mu = \lambda^r_\mu p_\mu$ independently. Then the corresponding gradient scale that the KEDF model need to fit will be reduced by $\nabla_{p_\mu} T'_S = \nabla_{p_\mu} T_S / \lambda^r_\mu$ ($\lambda^r_\mu \geq 1$ for most cases).

[0087] Natural Reparameterization

[0088] Nevertheless, even with this approach to trade-off coefficient scale and gradient scale, simultaneously reducing both scales is still difficult. Two techniques are introduced before applying dimension-wise rescaling. Firstly, since different bases have different importance in representing density, different coefficient dimensions have different sensitivities to the density function, causing some dimensions to have particularly large Lipschitz coefficient for the energy. Therefore, a natural reparameterization of the density coefficients is conducted. The principle is to make the

Euclidean metric in the reparameterized coefficient space reflect the L2-metric in the density-function space, so that the sensitivities across coefficient dimensions are balanced, allowing a better dimension-wise rescaling result to make learning easier. Under atomic basis, this principle induces a metric on density coefficient $d(p,p')^2 = (p-p')^{\mathsf{T}}W(p-p')$, where W is the overlap matrix of the density basis. Thus the density can be reparameterized by $\tilde{p} := M^{\mathsf{T}}p$, where M satisfies $MM^{\mathsf{T}} = W$. This reparameterization also leads to natural gradient descent in density optimization, which is shown to converge faster than standard gradient descent.

[0089] Atomic Reference Model

[0090] To further improve the coefficient-gradient scale trade-off, the coefficient absolute scale can be reduced by subtracting a statistical mean (data centering), but this cannot reduce the gradient absolute scale. Therefore, an atomic reference model is introduced for the kinetic energy that is linear in density coefficients. For a simple dependency on molecular structure $\mathcal{M}(n)$, density coefficients corresponding to the same atom type/species/chemical element share the same linear weight. The linear model is constructed by taking the per-type linear weights $\bar{g}_{\mathcal{M}(n)}$ as the statistical mean of gradient data for each atom type, and fit a per-type bias $\bar{T}_{\mathcal{M}(n)}$ on the training dataset, as illustrated by

$$\left(T_S^{(k)}\right)^{ref}_{\mathcal{M}(n)} = \bar{g}^{\mathsf{T}}_{\mathcal{M}(n)} p^{(k)}_{\mathcal{M}(n)} + \bar{T}_{\mathcal{M}(n)}.$$

[0091] This effectively centers the gradient labels for the main Graphormer Model to learn.

[0092] Density Optimization

[0093] In the deployment stage, the common density initialization is out-of-distribution with respect to training density, as the latter is the solution to an effective non-interacting system while the former is the superposition of isolated atomic density.

Two approaches bypass the problem: use an initialization that is a solution to a non-interacting system, or train a model that projects the initialized density in-distribution, specifically toward the ground-state density. Proper stopping criteria have been designed for in-scale and larger-scale systems.

[0094] After training, the KEDF model $T_{S,\theta}(p, \mathcal{M})$ is used to solve for the ground-state density and energy of a given molecule in structure $\mathcal{M}$ by minimizing the total electronic energy with respect to density coefficient p. Gradient descent is used to optimize p, as it is unnatural to formulate the process into SCF iteration, and gradient descent also works in KSDFT more stably than SCF iteration. Note that the normalization constraint of density is handled by projecting the standard gradient g onto the admissible plane before updating the coefficients.

[0095] Fig. 5 shows typical density optimization curves for a QM9 molecule with different initializations. Initializing the optimization with ground density ("Ground Density" in figure) leads to a stationary line. All other curves, excluding MINAO, consistently remain above the KSDFT energy. Notably, at the final step, ProjMINAO yields an energy that is only marginally higher (0.60 kcal/mol) than KSDFT. The energy convergence obtained from the Hückel initialization is 5.34 kcal/mol, while the energy convergence achieved using the MINAO+1SCF initialization is 1.90 kcal/mol.

[0096] For initialization, note the common method MINimal Atomic Orbital (MINAO) gives densities out of distribution from the training data, since they do not exactly correspond to the solution of an eigenvalue problem. This mismatch leads to unreliable results even in previous simple tasks and with kernel methods. Therefore, an initialization that is indeed an eigenvalue solution, e.g., the Hückel initialization, is considered. Alternatively, employing a machine learning model to project the MINAO density onto the space of eigenvalue solutions (ProjMINAO) may also be considered.

[0097] Initialization of the density requires special attention. As shown in Fig. 5, with all the training techniques disclosed herein, the optimization curve from the ground-state density keeps stationary. But the result from the MINAO density, which is the common KSDFT initialization, is undesired: the curve stabilizes for a while but with a gap below the ground-state energy, and even diverges later. A hint on this problem can be drawn from the curve from the density after the first SCF iteration from MINAO (MINAO+1SCF), which although starts with the same extent of deviation from the ground-state density as MINAO in terms of energy, converges stably and well approaches the true value. This observation implies that the problem with MINAO is probably due to its out-of-distribution nature: the model is trained on densities that come from the eigensolutions to effective one-electron Hamiltonian matrices, while the MINAO density is from the superposition of orbitals on each atom in isolation. This inspires us to employ an initial density that is in distribution.

[0098] The first choice is the Hückel initialization, which is from the eigenvalue-problem solution of a simple Hamiltonian matrix, hence is an in-distribution density. From Fig. 5, although the initialization seems much worse than MINAO in terms of total energy, it surprisingly converges well to the ground-state density.

[0099] The second choice, dubbed ProjMINAO, is to employ a model to correct the MINAO density towards an in-distribution density. In practice, an additional output branch is introduced to the model for predicting the required correction on the input coefficient towards the ground-state coefficient. Accordingly, a density loss is introduced to also train this branch. As shown in Fig. 5, after correction, the density optimization curve indeed converges well and gives an accurate energy result. The result appears even better than using the Hückel initialization as the density corrector provides a starting point near the ground-state where the model approximates the

functional more confidently (since near-convergent SCF solutions distribute around the ground state). Note that even after the correction, density optimization is still effective, which suggests the potential to give better results than the end-to-end "structure→ground-state density→ground-state energy" pipeline.

[00100] Another issue that needs to be addressed for density optimization is a proper stopping criterion. In principle, the optimization process converges at the stationary point (zero projected gradient) of the total energy functional. But in practice, due to discretization error, the exact zero-gradient point may be missed. There, the result at the step of minimum single-step energy update after a chosen number of gradient-descent steps is chosen instead. For larger-scale systems, however, this criterion is not as suitable since the model more or less shows some extrapolation error, and even a minor error around the ground-state may lead the optimization towards unreliable regions of the model. Therefore, in this case, results after a long optimization are not reliable. Accordingly, the methods disclosed herein stop when the projected gradient norm first stops decreasing. If there is no such a step within the chosen number of steps, then the step where the energy update first stops decreasing may be chosen. If such a step is still absent, the step with the minimal energy update within the chosen number of steps may be chosen. The priority of projected gradient norm is higher than that of energy update is due to the former is more sensitive to extrapolation hence provides a safer stop.

[00101] Fig. 8 shows computing device 12 of computing system 12 at training time. Processing circuitry 14 of computing device 12 is configured to train the kinetic energy density function machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ on a training data set 56 that is generated at least in part by the following operations. At training time, based on orbital values (e.g., coefficients **C** for wave functions) computed in intermediate cycles

of a self-consistent field (SCF) algorithm 52 evaluation of training-time molecular structure data 50 for the training-time molecular structure, computing the atomic basis density expansion coefficients $\mathbf{p}$, non-interacting kinetic energy $T_S(\mathbf{p})$, and gradients of the non-interacting kinetic energy $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ for the intermediate cycles. The processing circuitry 14 is further configured to store the density expansion coefficients and training-time molecular structure as training-time inputs, and the non-interacting kinetic energy and the kinetic energy density expansion gradients as ground truth output in the training data set 54. These values may be processed by the density feature preprocessing module in the manner described herein, prior to being stored in the training data set, or prior to be being inputted into the KEDF model 20.

[00102] As shown in Fig. 6, the kinetic energy density function machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ can have a graph neural network transformer architecture configured to implement a non-local attention mechanism. It will be appreciated that the kinetic energy density function machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ includes a transformer neural network with a multi-headed attention mechanism that is trained by backpropagation using the training-time inputs and the ground truth output. As shown in Fig. 8, one example of the multi-headed attention mechanism is configured to compute attention and loss for a predictive task of predicting the non-interacting kinetic energy $T_S(\mathbf{p})$ given the density expansion coefficients $\mathbf{p}$ as input, and is configured to compute loss by backpropagation with respect to the input density expansion coefficients for a predictive task of predicting the kinetic energy density expansion gradient. Each loss is typically measured by comparing the predicted value to the ground truth value for the inputs. A backpropagation algorithm may be used to train the KEDF model 20 based on the loss functions.

[00103] As shown in Fig. 8, the training-time molecular structure data 50 includes data indicating atomic attributes, such as atomic coordinates **X** and atomic number **Z**. In addition, the molecular structure data 50 may indicate atomic attributes such as a number of electrons in the training-time molecular structure, and an orbital basis set. The SCF algorithm 52 is executed at least in part by iterating through a plurality of SCF cycles. Herein, SCF cycles are also referred to as steps and denoted by the index **k**. In each SCF cycle the processing circuitry is configured to: compute a plurality of orbital coefficients by which the orbital basis set is expanded to determine orbital values for each electron, based on the orbital coefficients construct a matrix approximating the single-electron energy operator of a given quantum system, such as a Fock matrix **F**, using the constructed matrix compute updated orbital coefficients **C**, based on the updated orbital coefficients compute non-interacting kinetic energy, compute a gradient value $\nabla_{\mathbf{p}} T_S(\mathbf{p}, \mathcal{M})$ for the non-interacting kinetic energy, compute density expansion coefficients **p** under atomic basis based on the orbital coefficients **C**, store values for the non-interacting kinetic energy density expansion coefficients, non-interacting kinetic energy, and kinetic energy density expansion gradients, and determine whether a total energy and/or the density matrix has converged to within a convergence threshold, and if converged, update a density matrix for the training-time molecular structure based on the orbital coefficients and cease iterating.

[00104] The training-time molecular structure data can be stored in a graph data structure, in which the atomic position and atomic number are represented as node features in the graph data structure, and the atomic positions are used to compute edge values in the graph data structure. As shown in Fig. 8 at the density feature preprocessing module, the density coefficients can be expressed in a local atomic frame of reference, which is equivariant to global shift and rotation, and makes the density

expansion coefficients SE(3)-invariant. Further, the density coefficients can be subject to natural reparameterization to make a Euclidean metric in a reparameterized coefficient space trend towards an L2-metric in density-function space. Yet further, the density coefficients can be subject to dimension-wise rescaling. An atomic reference model can be adopted for the kinetic energy that is linear in density coefficients, such that density coefficients corresponding to a same atom type, species and/or chemical element share a same linear weight.

[00105] Turning now to FIG. 9A-9C, a computerized method 100 according to one example implementation is illustrated. The method may be implemented using the hardware and software elements described herein, or other suitable hardware and software. Method 100 is implemented at two different times, inference time and training time. Training time typically occurs before inference time. Steps 102-112 are implemented at inference time, and steps 114-156 are implemented at training time. Continuing with Fig. 9A, method 100 includes, at 102 receiving inference-time molecular structure data for an inference-time molecular structure $\mathcal{M}$. The inference time molecular structure data that is received includes inference-time atomic basis density expansion coefficients $\mathbf{p}$ and data indicating atomic attributes, such as atomic positions $\mathbf{X}$ and atomic number, of each atom $\mathbf{Z}$ in the inference-time molecular structure $\mathcal{M}$. At 104, the method includes inputting the molecular structure data to a trained kinetic energy density functional machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ to thereby generate a predicted value for electronic non-interacting kinetic energy $T_S(\mathbf{p}, \mathcal{M})$ of a density function $\rho(\mathrm{r})$, which can be based on or determined by determined by atomic basis density expansion coefficients $\mathbf{p}$. As shown at 106, the trained kinetic energy density machine learning model has been trained prior to inference time on a training data set that includes, as input, atomic attributes such as atomic positions $\mathbf{X}$ and/or

atomic numbers **Z** of each atom in the training-time molecular structure $\mathcal{M}$, and atomic basis density expansion coefficients **p** for the training-time molecular structure $\mathcal{M}$. As shown at 108, the atomic basis density expansion coefficients **p** indicate values by which electron charge density is represented on an atomic basis, and that includes corresponding ground truth values for non-interacting kinetic energy $T_S(\mathbf{p}, \mathcal{M})$ and as shown at 109, the training data set can further include ground truth values for the kinetic energy density expansion gradient $\nabla_{\mathbf{p}} T_S(\mathbf{p}, \mathcal{M})$. At 110, the method includes outputting the predicted value for non-interacting kinetic energy for the inference-time molecular structure. At 111, the method includes outputting from the trained kinetic energy density function machine learning model a predicted value of the kinetic energy density expansion gradient through backpropagation at inference time. At 112, the method includes performing an orbital-free Density Functional Theory (DFT) calculation using the predicted value of the kinetic energy density expansion gradient. An example equation for such calculation is shown in Fig. 1C.

[00106] Fig. 9B illustrates steps performed at training time. As shown, method 100 further includes, at 114 generating a training data set. The training data set may be generated at least in part by the steps shown at 116-128. At 116, the method includes based on orbital values computed in intermediate cycles of a self-consistent field (SCF) algorithm evaluation of training-time molecular structure data for the training-time molecular structure, computing the atomic basis density expansion coefficients, non-interacting kinetic energy, and gradients of the non-interacting kinetic energy for the intermediate cycles. At 118, the method may include performing density feature preprocessing, such as shown at steps 120-126. At 120, the method may include expressing the density coefficients in a local atomic frame of reference, which is equivariant to global shift and rotation, and makes the density expansion coefficients

SE(3)-invariant, as described herein. At 122, the method may include subjecting the density coefficients to natural reparameterization to make a Euclidean metric in a reparameterized coefficient space trend towards an L2-metric in density-function space, as described herein. At 124, the method may include subjecting the density coefficients to dimension-wise rescaling, as described herein. At 126, the method may include adopting an atomic reference model for the kinetic energy that is linear in density coefficients, such that density coefficients corresponding to a same atom type, species and/or chemical element share a same linear weight, as described herein. It will be appreciated that the inference-time molecular structure data can be stored in a graph data structure, the atomic position and atomic number are represented as node features in the graph data structure, and the atomic positions are used to compute edge values in the graph data structure.

[00107] At 128, the method includes storing the computed density expansion coefficients and training-time molecular structure as training-time inputs, and the non-interacting kinetic energy and the kinetic energy density expansion gradients as ground truth output in the training data set . At 130, the method further includes training the trained kinetic energy density function machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ on the training data set.

[00108] At 132, the method can include configuring the kinetic energy density function machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ to have a graph neural network transformer architecture configured to implement a non-local attention mechanism. At 134, the method can include configuring the non-local attention mechanism to be a multi-headed attention mechanism that is trained by backpropagation using the training-time inputs and the ground truth output. As shown at 136, the multi-headed attention mechanism is configured to compute attention and loss for a predictive task

of predicting the non-interacting kinetic energy given the density expansion coefficients as input, and, as shown at 138, is configured to compute loss by backpropagation with respect to the input density expansion coefficients for a predictive task of predicting the kinetic energy density expansion gradient loss. At 140, the method may include outputting the trained kinetic energy function machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$.

[00109] Turning now to Fig. 9C, step 116 of method 100 may be implemented by performing step 116A. It will be appreciated that the training-time molecular structure data can include data indicating atomic coordinates, and an atomic number. The molecular structure data can further include a number of electrons in the training-time molecular structure, and an orbital basis set. At 116A, the method includes executing the SCF algorithm at least in part by iterating through a plurality of SCF cycles, and in each SCF cycle performing the following operations: at 142 computing a plurality of orbital coefficients by which the orbital basis set is expanded to determine orbital values for each electron, at 144 based on the orbital coefficients constructing a matrix approximating the single-electron energy operator of a given quantum system (e.g., a Fock matrix **F**), at 146 using the Fock matrix compute updated orbital coefficients **C**, at 148 based on the updated orbital coefficients compute non-interacting kinetic energy $T_S(\mathbf{p}, \mathcal{M})$, at 150 compute a gradient value $\nabla_{\mathbf{p}} T_S(\mathbf{p}, \mathcal{M})$ for the non-interacting kinetic energy $T_S(\mathbf{p}, \mathcal{M})$, at 152 compute density expansion coefficients **p** under atomic basis based on the updated orbital coefficients **C**, at 154 store values for the non-interacting kinetic energy density expansion coefficients, non-interacting kinetic energy, and kinetic energy density expansion gradients, and at 156 determine whether a total energy and/or the density matrix has converged to within a convergence threshold, and if converged, update a density matrix for the training-time molecular structure based on the orbital coefficients and cease iterating. In this manner, training

data can be produced during the execution of an SCF algorithm, when performing KSDFT.

[00110] Further, the computerized method can include iteratively optimizing the density expansion coefficients **p** for an inference-time molecular structure $\mathcal{M}$ using the trained kinetic energy density function machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$. This can be accomplished by, in in each of a plurality of iteration cycles: inputting the molecular structure data and density expansion coefficients **p** to the trained kinetic energy density functional machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ to thereby generate a predicted value for electronic non-interacting kinetic energy $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ of the density function $\boldsymbol{\rho}(\mathrm{r})$, and then generating the predicted kinetic energy density expansion gradient $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ by backpropagation through the trained kinetic energy density functional machine learning model $T_{S,\theta}(\mathbf{p}, \mathcal{M})$ with respect to the input atomic basis density expansion coefficients **p** computing the gradient of the sum of Hartree energy, external potential energy, and exchange-correlation energy, with respect to the input atomic basis density expansion coefficients **p**, and adding it to the computed gradient $\nabla_{\mathbf{p}} T_S(\mathbf{p})$; projecting the gradient onto a linear subspace of density expansion coefficients that determine normalized densities; updating the density expansion coefficients **p** using the projected gradient; and determining whether a total energy change value from a previous cycle or a projected gradient normalized value is larger than a value in the previous cycle, and if larger, cease iterating.

[00111] The computerized method can further include initializing the density expansion coefficients by adding upon MINAO initialized density with an increment that is predicted using a trained initialization value machine learning model that takes as input the inference-time molecular structure, and a standard MINAO initialized density expansion coefficient, wherein the trained initialization value machine learning

model has been trained on a training data set that includes, as input, atomic positions **X** and data indicating atomic numbers **Z** of each atom in the training-time molecular structure $\mathcal{M}$, and atomic basis density expansion coefficients **p** for the training-time molecular structure $\mathcal{M}$, the atomic basis density expansion coefficients **p** indicating values by which electron charge density is represented on an atomic basis, and that includes corresponding ground truth values for an increment with respect to the density expansion coefficients at a SCF converged step.

[00112] APPENDIX

[00113] A Mechanism of Density Functional Theory

[00114] The following are details regarding DFT, including the formulation of OFDFT under atomic basis, and the mechanism and details to use KSDFT to generate value and gradient data for learning KEDF. Atomic units are used throughout.

Table 1: Notations

Basic concepts	
$r \in \mathbb{R}^3$	Electron coordinate
N	Number of electrons in a molecular system
$\psi(r^{(1)}, \dots, r^{(N)})$	N-electron wavefunction
$\langle f \mid g \rangle := \int f(\boldsymbol{r}) g(\boldsymbol{r})\, d\boldsymbol{r}$	The standard inner product in function space
$\langle f \mid \hat{O} \mid g \rangle := \int f(\boldsymbol{r})(\hat{O}g)(\boldsymbol{r})\, d\boldsymbol{r}$	Function-space inner product with operator
$(f \mid g) := \int \frac{f(r)g(r')}{\lVert r - r' \rVert} dr dr'$	Coulomb integral
Molecular system	
$x \in \mathbb{R}^3$	Nucleus coordinate
$a, b \in \{1,2, \dots, A\}$	Indices for atoms in a molecule
$X := (X^{(1)}, \dots, X^{(A)})$	Molecular conformation/geometry
$Z := (Z^{(1)}, \dots, Z^{(A)})$	Molecular composition
$M \in \{X, Z\}$	Molecular structure
$\{M^{(d)}\}_{d=1}^{D}$	Molecular structures in a dataset
Density functional theory	
$\Phi = \{\phi_i(r)\}_{i=1}^{N}$	Orbitals
$i, j \in \{1,2, \dots, N\}$	Indices for orbitals or electrons
$\psi_{[\Phi]}(r^{(1)}, \dots, r^{(N)}) := \det[\phi_i(r^j)]_{ij}$	Slater determinant from orbitals $\Phi = \{\phi_i(r)\}_{i=1}^{N}$

$\{\eta_a(r)\}_{a=1}^{B}$	Orbital basis
Subscripts $\alpha, \beta, \gamma, \delta \in \{1,2,\dots,B\}$	Indices for orbital basis
$C_{\alpha i}$	Orbital coefficients
$\Gamma_{\alpha b} = \sum_{i=1}^{N} C_{\alpha i} C_{bi}$	Density matrix
$S_{\alpha b} = \langle \eta_\alpha \vert \eta_\beta \rangle$	Overlap matrix of orbital basis
$D_{\alpha b,\gamma\delta} = \langle \eta_\alpha \eta_\beta \vert \eta_\gamma \eta_\delta \rangle$	Overlap matrix of paired orbital basis
$\widetilde{D}_{\alpha b,\gamma\delta} = \langle \eta_\alpha \eta_\beta \vert \eta_\gamma \eta_\delta \rangle$	4-center-2-electron Coulomb integral of oribtal basis
$\rho(r)$	(One-electron reduced) density (function)
$\{\omega_\mu(r)\}_{\mu=1}^{M}$	Density basis
Subscripts $\mu, \upsilon \in \{1,2,\dots,M\}$	Indices for density basis
Subscript $\mu = (a, \tau)$	Atom assignment decomposition of basis index
p_μ	Density coefficient
$w_\mu := \int w_\mu(\boldsymbol{r})\, d\boldsymbol{r}$	Density basis integral
$W_{\mu\upsilon} := \langle \omega_\mu \vert \omega_\upsilon \rangle$	Overlap matrix of density basis
$\widetilde{W}_{\mu\upsilon} := \langle \omega_\mu \vert \omega_\upsilon \rangle$	2-center-2-electron Coulomb integral of density basis
$L_{\mu,\alpha\beta} := \langle \omega_\mu \vert \eta_\alpha \eta_\beta \rangle$	Overlap matrix between density basis and orbital basis
F	Fock matrix
k	Step index for SCF iteration or density optimization process ($\star$ for the converged step)
$(V_{\text{eff}})_{\alpha\beta} := \langle \eta_\alpha \vert V_{\text{eff}} \vert \eta_\beta \rangle$	Effective potential matrix under orbital basis
$(v_{\text{eff}})_\mu := \langle \omega_\mu \vert V_{\text{eff}} \rangle$	Effective potential vector under density basis
$\mu \in \mathbb{R}$	Chemical potential
$\varepsilon := \text{Diag}[\varepsilon_1, \dots, \varepsilon_N]$	The diagonal N × N matrix of orbital energies
$U[\rho]$	Universal functional
$E_{XC}[\rho]$	Exchange-correlation (XC) functional
$T_S[\rho]$	Kinetic energy density functional (KEDF)
$T_S(p, M)$	KEDF under atomic basis of density
$T_{S,\theta}(p, M)$	KEDF model/approximation
T_{ABPE}	The APBE kinetic functional as the reference functional
$T_{S,\text{res}}$	Residual KEDF from the reference functional
E_{TXC}	Kinetic and XC functional
f	Hellmann-Feynman force

[00115] A.1 Basic Formulation of Density Functional Theory

[00116] For brevity, the following formulation is for spinless fermions therefore only consider spacial states. For the restricted Kohn-Sham calculation for data generation, a pair of electrons of opposite spins share a common spacial orbital is adopted, which amounts to duplicate the orbitals in the following formulation. The mechanism of DFT may be more intuitively introduced under Levy's constrained search formulation. The N-electron Schrodinger equation for ground state is equivalent to the following optimization problem on N-electron wavefunctions $\psi\left(r^{(1)},\ldots,r^{(N)}\right)$ under the variational principle:

$$E^{\star}=\min_{\psi:\,\text{antisym.},\langle\psi|\psi\rangle=1}\langle\psi|\hat{T}+\hat{V}_{ee}+\hat{V}_{\text{ext}}|\psi\rangle. \tag{5}$$

where $\hat{T}:=-\frac{1}{2}\sum_{i=1}^{N}\nabla^{(i)^2}$ is the kinetic operator, $\hat{V}_{ee}:=\sum_{1\le i<j\le N}\frac{1}{\|r^{(i)}-r^{(j)}\|}$ is the electron-electron Coulomb interaction (internal potential), and $\hat{V}_{ext}:=\sum_{i=1}^{N}V_{ext}\left(\mathrm{r}^{(i)}\right)$ comes from a one-body external potential $V_{ext}(r)$ that commonly arises from the electrostatic field of the nuclei specified by the given molecular structure $M\in\{X,Z\}$ where $X:=\left(X^{(1)},\ldots,X^{(A)}\right)$ and $Z:=\left(Z^{(1)},\ldots,Z^{(A)}\right)$:

$$V_{\text{ext}}(\mathbf{r})=-\sum_{a=1}^{A}\frac{Z^{(a)}}{\|\mathbf{r}-\mathbf{x}^{(a)}\|}. \tag{6}$$

[00117] Although the optimization problem is exactly defined, directly optimizing the N-electron wavefunction is very challenging computationally. Specifying the wavefunction ψ and evaluating the energy already require an exponential cost in principle, as ψ is a function on whose dimension $\mathbb{R}^{3N}$ increases with N. To make an easier optimization problem, it is then desired to optimize a functional of the one-electron reduced density,

$$\rho_{[\psi]}(\mathbf{r}):=N\int|\psi(\mathbf{r},\mathbf{r}^{(2)},\cdots,\mathbf{r}^{(N)})|^2\,d\mathbf{r}^{(2)}\cdots d\mathbf{r}^{(N)}. \tag{7}$$

which has an intuitive physical interpretation of charge density even under a classical view, and more importantly, the cost to specify a density is constant (w.r.t N) in principle, as the density is a function on $\mathbb{R}^3$whose dimension is constant. This is the starting point of Density Functional Theory (DFT) and was first formally verified by Hohenberg and Kohn.

[00118] In terms of the density, the external potential energy, i.e., the last term in Eq. (5), is already an explicit density functional, since the external potential is one-body:

$$\langle \psi | \hat{V}_{\text{ext}} | \psi \rangle = \int \rho_{[\psi]}(\mathbf{r}) V_{\text{ext}}(\mathbf{r})\, \mathrm{d}\mathbf{r} = E_{\text{ext}}[\rho_{[\psi]}], \quad \text{where } E_{\text{ext}}[\rho] := \int \rho(\mathbf{r}) V_{\text{ext}}(\mathbf{r})\, \mathrm{d}\mathbf{r}. \tag{8}$$

For the other energy terms, using the density as an intermediate, the optimization problem in Eq. (5) can be equivalently carried out in two levels:

$$E^\star = \min_{\rho:\geqslant 0, \int \rho(\mathbf{r})\,\mathrm{d}\mathbf{r}=N} \left(\min_{\psi:\,\text{antisym}, \rho_{[\psi]}=\rho} \langle \psi | \hat{T} + \hat{V}_{\text{ee}} | \psi \rangle \right) + E_{\text{ext}}[\rho] \tag{9}$$

$$= \min_{\rho:\geqslant 0, \int \rho(\mathbf{r})\,\mathrm{d}\mathbf{r}=N} \left\{ E[\rho] := U[\rho] - E_{\text{ext}}[\rho] \right\}. \tag{10}$$

Here, the result of the first-level optimization problem carrying out a constrained search in Eq. (9) defines a density functional,

$$U[\rho] := \min_{\psi:\,\text{antisym}, \rho_{[\psi]}=\rho} \langle \psi | \hat{T} + \hat{V}_{\text{ee}} | \psi \rangle. \tag{11}$$

called the universal functional, as it is independent of system specification (*i.e., M* or V_{ext}). It is composed of the kinetic and internal potential energy of the electrons. The optimization objective is then formally converted to a density functional $E[\rho]$ as shown in Eq. (10).

[00119] To carry out practical computation, variants of the kinetic and internal potential energy that allow explicit calculation or have known properties are introduced, to cover the major part of the corresponding energies in $U[\rho]$. The internal potential

energy is covered by its classical version, *i.e.* assuming no correlation, called the Hartree energy:

$$E_{\mathrm{H}}[\rho] := \frac{1}{2}\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|}\,\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'. \tag{12}$$

The kinetic energy is covered by the kinetic energy density functional (KEDF), which is defined in a similar way as the universal functional:

$$T_{\mathrm{S}}[\rho] := \min_{\psi:\,\text{antisym},\rho_{[\psi]}=\rho} \langle\psi|\hat{T}|\psi\rangle. \tag{13}$$

The remainder in the universal functional is called the *exchange-correlation (XC) functional*:

$$E_{\mathrm{XC}}[\rho] := U[\rho] - T_{\mathrm{S}}[\rho] - E_{\mathrm{H}}[\rho],$$

which is by definition also a density functional. Under this decomposition, the density optimization problem in Eq. (10) becomes:

$$E^{\star} = \min_{\rho:\geqslant 0,\int \rho(\mathbf{r})\,\mathrm{d}\mathbf{r}=N} \left\{ E[\rho] = T_{\mathrm{S}}[\rho] + \underbrace{E_{\mathrm{H}}[\rho] + E_{\mathrm{XC}}[\rho] + E_{\mathrm{ext}}[\rho]}_{=:E_{\mathrm{eff}}[\rho]} \right\}. \tag{14}$$

[00120] Here $E_{\mathrm{eff}}[\rho]$ is defined for future convenience and named after the effective-potential interpretation of its variation detailed later. Using carefully designed explicit expressions or machine-learning models to approximate the $T_{\mathrm{S}}[\rho]$ and $E_{\mathrm{XC}}[\rho]$ functionals, practical computation can be conducted. This is the formulation of orbital-free density functional theory (OFDFT). Indeed, the object to be optimized is the electron density, which is one function on the constant-dimensional space of $\mathbb{R}^3$, hence greatly reduces computation complexity over the original variational problem Eq. (5). Under properly designed $T_{\mathrm{S}}[\rho]$ and $E_{\mathrm{XC}}[\rho]$ approximations, the complexity is favorably $O(N^2)$ under atomic basis.

[00121] Considering the KEDF $T_{\mathrm{S}}[\rho]$ is more challenging to approximate than the XC functional $E_{\mathrm{XC}}[\rho]$, Kohn and Sham leverage an equivalent formulation of KEDF

to allow its accurate calculation, at the cost of increasing the complexity. The alternative formulation optimizes *determinantal* wavefunctions. A determinantal wavefunction for N electrons is specified by N one-electron wavefunctions $\Phi = \{\phi_i(r)\}_{i=1}^N$, a.k.a orbitals, following the form:

$$\psi_{[\Phi]}(\mathbf{r}^{(1)}, \cdots, \mathbf{r}^{(N)}) := \frac{1}{\sqrt{N!}} \det[\phi_i(\mathbf{r}^{(j)})]_{ij}$$

given orbitals $\Phi = \{\phi_i(r)\}_{i=1}^N$.

[00122] The equivalent optimization problem in parallel with Eq. (13) is:

$$T_S[\rho] = \min_{\{\phi_i\}_{i=1}^N \to \rho_{[\Phi]}=\rho} \langle \psi_{[\Phi]} | \hat{T} | \psi_{[\Phi]} \rangle = \min_{\{\phi_i\}_{i=1}^N : \text{orthonormal}, \rho_{[\Phi]}=\rho} \sum_{i=1}^N \langle \phi_i | \hat{T} | \phi_i \rangle. \tag{15}$$

where $\rho_{[\Phi]}(r) := \sum_{i=1}^N |\phi_i(r)|^2$, given orthonormal orbitals $\Phi = \{\phi_i(r)\}_{i=1}^N$. (16)

[00123] In Eq. (15), the second equality holds for spinless fermions or restricted-spin electrons calculation, since the density normalizes to N, and a set of (non-collinear) functions can always be orthogonalized using *e.g.* the Gram-Schmidt process. This equivalence can be understood from the interpretation of $T_S[\rho]$ as the *non-interacting* portion of kinetic energy. Indeed, for a non-interacting system, there are only kinetic energy and external potential energy (*i.e.*, taking $\hat{V}_{ee} = 0$), so the two-level optimization in parallel with Eq. (9) becomes:

$$\begin{aligned} E^\star &= \min_{\psi:\, \text{antisym.}, \langle \psi | \psi \rangle = 1} \langle \psi | \hat{T} + \hat{V}_{\text{ext}} | \psi \rangle \\ &= \min_{\rho: \geq 0, \int \rho(\mathbf{r})\, d\mathbf{r} = N} \left(\min_{\psi:\, \text{antisym.}, \rho_{[\psi]}=\rho} \langle \psi | \hat{T} | \psi \rangle \right) + E_{\text{ext}}[\rho] \\ &= \min_{\rho: \geq 0, \int \rho(\mathbf{r})\, d\mathbf{r} = N} T_S[\rho] + E_{\text{ext}}[\rho]. \end{aligned} \tag{17}$$

from which it can be seen that $T_S[\rho]$ is the ground-state kinetic energy of the non-interacting system whose ground-state density is ρ. On the other hand, it is known that the ground-state wavefunction of a noninteracting system is commonly determinantal (at least when the ground state is non-degenerate [Thm. 4.6]). Hence the optimization can be rewritten as:

$$
\begin{aligned}
E^\star &= \min_{\{\phi_i\}_{i=1}^N} \langle \Psi_{[\Phi]} | \hat{T} + \hat{V}_{\text{ext}} | \Psi_{[\Phi]} \rangle \\
&= \min_{\rho:\geqslant 0, \int \rho(\mathbf{r})\,\mathrm{d}\mathbf{r}=N} \left(\min_{\{\phi_i\}_{i=1}^N : \rho_{[\Phi]}=\rho} \langle \Psi_{[\Phi]} | \hat{T} | \Psi_{[\Phi]} \rangle \right) + E_{\text{ext}}[\rho] \\
&= \min_{\rho:\geqslant 0, \int \rho(\mathbf{r})\,\mathrm{d}\mathbf{r}=N} \left(\min_{\substack{\{\phi_i\}_{i=1}^N : \text{orthonormal.} \\ \rho_{[\Phi]}=\rho}} \sum_{i=1}^N \langle \phi_i | \hat{T} | \phi_i \rangle \right) + E_{\text{ext}}[\rho].
\end{aligned}
\tag{18}
$$

which indicates Eq. (15).

[00124] Back to the main problem, leveraging this knowledge about $T_S[\rho]$ in the variational problem Eq. (14) gives:

$$
E^\star = \min_{\rho:\geqslant 0, \int \rho(\mathbf{r})\,\mathrm{d}\mathbf{r}=N} \left(\min_{\substack{\{\phi_i\}_{i=1}^N : \text{orthonormal.} \\ \rho_{[\Phi]}=\rho}} \sum_{i=1}^N \langle \phi_i | \hat{T} | \phi_i \rangle \right) + E_{\text{H}}[\rho] + E_{\text{XC}}[\rho] + E_{\text{ext}}[\rho].
$$

which can be converted into directly optimizing the orbitals in a single-level optimization:

$$
E^\star = \min_{\{\phi_i\}_{i=1}^N : \text{orthonormal}} \left\{ E[\Phi] := \sum_{i=1}^N \langle \phi_i | \hat{T} | \phi_i \rangle + \underbrace{E_{\text{H}}[\rho_{[\Phi]}] + E_{\text{XC}}[\rho_{[\Phi]}] + E_{\text{ext}}[\rho_{[\Phi]}]}_{E_{\text{en}}[\rho_{[\Phi]}]} \right\}.
\tag{19}
$$

[00125] This is the formulation of Kohn-Sham density functional theory (KSDFT). With decades of development of XC functional approximations, KSDFT has achieved remarkable success and become among the most popular quantum chemistry method. In its formulation, the object to be optimized is a set of orbitals $\{\phi_i(r)\}_{i=1}^N$, which are N functions on $\mathbb{R}^3$. This is still substantially cheaper than optimizing an N-electron wavefunction on $\mathbb{R}^{3N}$, but has a complexity at least $O(N)$ times more than OFDFT, due to N times more $\mathbb{R}^3$functions to optimize. Under atomic basis, KSDFT has a complexity at least $O(N^3)$ (using density fitting) without further approximations. The machine learning models described herein provide the opportunity to approximate KEDF accurately enough to match successful XC functional approximations. This enables accurate and practical OFDFT calculation, with lower complexity to advance the accuracy-scalability trade-off in quantum chemistry.

[00126] A.2 KSDFT Calculation Produces Labels of KEDF

[00127] The reasons why a KSDFT calculation procedure could provide value and gradient labels of KEDF will now be explained. Computation details under atomic basis are postponed in Supplementary Sec. A.4. First, a typical algorithm to solve the optimization problem in KSDFT is described. To determine the optimal solution of orbitals $\Phi = \{\phi_i(r)\}_{i=1}^N$, the variation of the energy functional $E[\Phi]$ in Eq. (19) w.r.t each orbital ϕ_i is required:

$$\begin{aligned}\frac{\delta E[\Phi]}{\delta \phi_i}(\mathbf{r}) &= \frac{\delta \sum_{j=1}^N (-1/2)\langle \phi_j|\nabla^2|\phi_j\rangle}{\delta \phi_i}(\mathbf{r}) + \int \frac{\delta E_{\text{eff}}[\rho_{[\Phi]}]}{\delta \rho}(\mathbf{r}')\frac{\delta \rho_{[\Phi]}(\mathbf{r}')}{\delta \phi_i(\mathbf{r})}\,\mathrm{d}\mathbf{r}' \\ &= 2\hat{T}\phi_i(\mathbf{r}) + 2V_{\text{eff}[\rho_{[\Phi]}]}(\mathbf{r})\phi_i(\mathbf{r}). \qquad (20)\end{aligned}$$

$$\text{where} \quad V_{\text{eff}[\rho]}(\mathbf{r}) := \frac{\delta E_{\text{eff}}[\rho]}{\delta \rho}(\mathbf{r}) = \underbrace{\int \frac{\rho(\mathbf{r}')}{\|\mathbf{r}'-\mathbf{r}\|}\,\mathrm{d}\mathbf{r}'}_{=:V_{\text{H}[\rho]}(\mathbf{r})} + \underbrace{\frac{\delta E_{\text{XC}}[\rho]}{\delta \rho}(\mathbf{r})}_{=:V_{\text{XC}[\rho]}(\mathbf{r})} + V_{\text{ext}}(\mathbf{r}). \qquad (21)$$

[00128] The term $V_{\text{eff}[\rho[\Phi]]}$ arises as the variation w.r.t density ρ since the orbitals determine the other-than-T_S energy E_{eff} (defined in Eq. (14)) only through the density $\rho_{[\Phi]}$ they define. By Eq. (16), $\frac{\delta \rho[\Phi](\mathbf{r}')}{\delta \phi_i(r)} = 2\phi_i(r)\delta(r - r')$, which then gives Eq. (20). Combining Eq. (20) with the variation of the orthonormal constraint yields the optimality equation for the problem Eq. (19):

$$\hat{F}_{[\rho_{[\Phi]}]}\phi_i := \hat{T}\phi_i + V_{\text{eff}[\rho_{[\Phi]}]}\phi_i = \varepsilon_i\phi_i. \quad \forall i = 1, \cdots, N. \qquad (22)$$

[00129] In the derivation, only the Lagrange multipliers ε_i for the normalization constraints $\langle\phi_i|\phi_i\rangle = 1$ are imposed, since from the resulting equations (22), $\{\phi_i\}_{i=1}^N$ are eigenstates of an Hermitian operator $\hat{F}_{[\rho[\Phi]]}$ called the Fock operator, hence are naturally orthogonal in the general case of non-degeneracy. These equations resemble the Schrodinger equation for N non-interacting fermions, where $V_{\text{eff}[\rho[\Phi]]}(\mathbf{r})$, as a function on $\mathbb{R}^3$, acts as an *effective* one-body external potential, hence the name.

[00130] Note that $V_{\text{eff}[\rho[\Phi]]}$ is unknown beforehand, as itself depends on the solution of orbitals. Hence a fixed point iteration is employed: starting from a set of initial orbitals $\Phi^{(0)} = \{\phi_i{}^{(0)}\}_{i=1}^{N}$, construct the Fock operator using results in previous iterations,

$$\hat{F}^{(k)} := \hat{T} + \hat{V}_{\text{eff}}^{(k)}. \tag{23}$$

where $V_{eff}^{(k)}$ is taken as $V_{eff\left[\rho_{[\Phi^{(k-1)}]}\right]}$ following this derivation, *i.e.*, $\hat{F}^{(k)} = \hat{F}_{\left[\rho_{[\Phi^{(k-1)}]}\right]}$ corresponding eigenvalue problem for the orbitals in the current iteration:

$$\hat{F}^{(k)}\phi_i^{(k)} = \varepsilon_i^{(k)}\phi_i^{(k)}, \quad \forall i = 1, \cdots, N. \tag{24}$$

[00131] The iteration stops until "self-consistency" is achieved, *i.e.*, the eigenstate solution $\Phi^{(k)} = \{\phi_i{}^{(k)}\}_i$ in the current step coincides (up to an acceptable error) with the orbitals $\Phi^{(k-1)} = \{\phi_i{}^{(k-1)}\}_{i=1}^{N}$ in the previous step that define $\hat{F}^{(k)}$. This is the self-consistent field (SCF) method.

[00132] An important fact of SCF is that, in each iteration k, the solution $\{\phi_i{}^{(k)}\}_{i=1}^{N}$ exactly defines the ground-state of a *non-interacting* system of N fermions moving in the effective one-body potential $V_{eff}^{(k)}$ as the external potential V_{ext}. Indeed, the variational problem Eq. (18) that describes the non-interacting system can be reformulated into a single-level optimization as:

$$E^* = \min_{\{\phi_i\}_{i=1}^{N}\text{:orthonormal}} \sum_{i=1}^{N} \langle \phi_i | \hat{T} | \phi_i \rangle + \int \rho_{[\Phi]}(\mathbf{r}) V_{\text{eff}}^{(k)}(\mathbf{r})\, \mathrm{d}\mathbf{r}. \tag{25}$$

whose variation coincides with Eq. (24) thus solved by $\{\phi_i{}^{(k)}\}_{i=1}^{N}$. This reveals the relation of SCF solution to the KEDF: this solution of orbitals $\{\phi_i{}^{(k)}\}_{i=1}^{N}$ achieves the

minimum non-interacting kinetic energy $\sum_{i=1}^{N}\langle\psi_{[\Phi]}|\hat{T}|\psi_{[\Phi]}\rangle$ among all orthonormal orbitals that lead to the same density $\rho_{[\Phi^{(k)}]}$ (otherwise Eq. (25) can be further minimized; can also be seen from the equivalence to Eq. (18)); by the alternative form of KEDF Eq. (15) as non-interacting ground-state kinetic energy, thus:

$$T_S\left[\rho_{[\Phi^{(k)}]}\right] = \langle\psi_{[\Phi^{(k)}]}|\hat{T}|\psi_{[\Phi^{(k)}]}\rangle = \sum_{i=1}^{N}\langle\phi_i^{(k)}|\hat{T}|\phi_i^{(k)}\rangle. \tag{26}$$

[00133] This indicates that *every SCF iteration produces a label for* T_S. Moreover, as the non-interacting variation problem Eq. (25) is equivalent to its two-level optimization form Eq. (18), which is in turn equivalent to the density optimization form using KEDF Eq. (17) (which explains the alternative KEDF form Eq. (15)), the density $\rho_{[\Phi^{(k)}]}$ from the solution $\Phi^{(k)} = \{\phi_i{}^{(k)}\}_i$ of each SCF iteration minimizes Eq. (17). Therefore, it satisfies the variation equation (Euler equation) of Eq. (17) (taking V_{ext} as $V_{eff}^{(k)}$) subject to the normalization constraint with Lagrange multiplier (chemical potential) $\mu^{(k)}$:

$$\frac{\delta T_S[\rho_{[\Phi^{(k)}]}]}{\delta\rho} + V_{\text{eff}}^{(k)} = \mu^{(k)}. \tag{27}$$

[00134] The variation of KEDF $\frac{\delta T_S}{\delta\rho}$ is related to the gradient w.r.t density coefficients when the density ρ is expanded on a basis (see Supplementary Sec. A.4.3). Hence, *every SCF iteration also produces a label for the gradient of* T_S*, up to a projection.*

[00135] It is worth noting that these arguments still hold when the effective potential $V_{eff}^{(k)}$ in SCF iteration k is *not* $V_{eff}\left[\rho_{[\Phi^{(k-1)}]}\right]$, since the deductions from Eq. (25) to Eq. (27) only require $V_{eff}^{(k)}$ to be a one-body potential. This allows more flexible data generation process since in common DFT calculation settings, $V_{eff}^{(k)}$ indeed deviates

from $V_{eff}\left[\rho_{[\Phi^{(k-1)}]}\right]$ for more stable and faster convergence, *e.g.* when using the "direct inversion in the iterative subspace" (DIIS) method. This also indicates that even when the XC functional used in data generation is not accurate, the generated value and gradient labels for $T_S[\rho]$ are still exact, since the XC functional still gives an effective one-body potential to define the non-interacting system Eq. (24) or Eq. (25), as long as it is pure (*i.e.*, only depends on density features). In this sense, data generation for KEDF is easier than that for the XC functional.

[00136] A.3 Formulation under Atomic Basis

[00137] For practical calculation, KSDFT typically uses an atomic basis $\{\eta_a(r)\}_{a=1}^{B}$ to expand the orbitals for conducting the SCF iteration in Eq. (24) for molecular systems. The expansion gives:

$$\phi_i(\mathbf{r}) = \sum_{\alpha} \mathbf{C}_{\alpha i}\eta_\alpha(\mathbf{r}), \tag{28}$$

which converts solving for eigenfunctions into the common problem of solving for eigenvectors of a matrix. On the other hand, as emphasized in Introduction 1 and Results 2.1, it is a goal to represent the density on an atomic basis $\{\omega_\mu(r)\}_{\mu=1}^{M}$ for efficient OFDFT implementation,

$$\rho(\mathbf{r}) = \sum_{\mu} \mathbf{p}_\mu\omega_\mu(\mathbf{r}). \tag{29}$$

The left-hand-sides of Eq. (28) and Eq. (29) may also be denoted as $\phi_{i,\mathbf{C}}$ or $\Phi_\mathbf{C}$ and $\rho_\mathbf{p}$ to highlight the dependency on the coefficients. Note that both the orbital basis $\{\eta_a(r)\}_{a=1}^{B}$and density basis $\{\omega_\mu(r)\}_{\mu=1}^{M}$ depend on the molecular structure M, as the location and type of each basis function is determined by the coordinates $\mathbf{x}^{(a)}$ and atomic number $Z^{(a)}$ of the corresponding atom. Nevertheless, the development in this subsection is for one given molecular system M, so its appearance is omitted for density

or orbital representation. Typically, the numbers of basis B and M increase linearly with the number of electrons N, *i.e.*, $O(B) = O(M) = O(N)$.

[00138] A.3.1 KSDFT under Atomic Basis

[00139] For the SCF iteration in Eq. (24), using the expansion of orbitals in Eq. (28), it becomes: $\sum_\beta C_{\beta i}^{(k)} \hat{F}^{(k)} \eta_\beta(r) = \varepsilon_i^{(k)} \sum_\beta C_{\beta i}^{(k)} \eta_\beta(r)$, $\forall i = 1, \ldots, N$. Integrating each function equation with basis function $\eta_\alpha(\mathbf{r})$ gives: $\sum_\beta C_{\beta i}^{(k)} \langle \eta_\alpha | \hat{F}^{(k)} | \eta_\beta \rangle = \varepsilon_i^{(k)} \sum_\beta C_{\beta i}^{(k)} \langle \eta_\alpha | \eta_\beta \rangle$, which can then be formulated as a generalized eigenvalue problem in matrix form:

$$\mathbf{F}^{(k)}\mathbf{C}^{(k)} = \mathbf{S}\mathbf{C}^{(k)}\boldsymbol{\varepsilon}^{(k)}. \tag{30}$$

$$\text{where} \quad \mathbf{F}_{\alpha\beta}^{(k)} := \langle \eta_\alpha | \hat{F}^{(k)} | \eta_\beta \rangle \overset{\text{Eq. (23)}}{=} \underbrace{\langle \eta_\alpha | \hat{T} | \eta_\beta \rangle}_{= -\frac{1}{2} \int \eta_\alpha(\mathbf{r}) \nabla^2 \eta_\beta(\mathbf{r}) \, d\mathbf{r} =: \mathbf{T}_{\alpha\beta}} + \underbrace{\langle \eta_\alpha | V_{\text{eff}}^{(k)} | \eta_\beta \rangle}_{=: (\mathbf{V}_{\text{eff}}^{(k)})_{\alpha\beta}}. \tag{31}$$

$$\mathbf{S}_{\alpha\beta} := \langle \eta_\alpha | \eta_\beta \rangle. \quad \boldsymbol{\varepsilon}^{(k)} := \begin{pmatrix} \varepsilon_1^{(k)} & & \\ & \ddots & \\ & & \varepsilon_N^{(k)} \end{pmatrix}.$$

Here, $\mathbf{F}^{(k)}$ is called the Fock matrix, and $\mathbf{S}$ is the overlap matrix of the orbital basis.

[00140] To show the expression of the Fock matrix, first the expression of the density defined by the orbital coefficients from Eq. (16) and Eq. (28) can be represented:

$$\begin{aligned} \rho_{\mathbf{C}}(\mathbf{r}) := \rho_{[\Phi_{\mathbf{C}}]}(\mathbf{r}) &= \sum_{i=1}^{N} \left| \sum_\alpha \mathbf{C}_{\alpha i} \eta_\alpha(\mathbf{r}) \right|^2 = \sum_{i=1}^{N} \sum_{\alpha\beta} \mathbf{C}_{\alpha i} \mathbf{C}_{\beta i} \eta_\alpha(\mathbf{r}) \eta_\beta(\mathbf{r}) \\ &= \sum_{\alpha\beta} \boldsymbol{\Gamma}_{\alpha\beta} \eta_\alpha(\mathbf{r}) \eta_\beta(\mathbf{r}), \end{aligned} \tag{32}$$

where the density matrix is defined corresponding to the orbital coefficients:

$$\boldsymbol{\Gamma} := \mathbf{C}\mathbf{C}^\top. \quad \boldsymbol{\Gamma}_{\alpha\beta} := \sum_{i=1}^{N} \mathbf{C}_{\alpha i} \mathbf{C}_{\beta i}. \tag{33}$$

Note that Eq. (16) requires orthonormal orbitals, hence Eq. (32) requires $\mathbf{C}$ to satisfy the corresponding orthonormality constraint shown in Eq. (43) below. Orbital coefficient solutions $\mathbf{C}^{(k)}$ in SCF iterations satisfy this constraint as explained later.

In the derivation of SCF iteration where $V_{eff}^{(k)}$ is taken as $V_{eff\left[\rho_{[\Phi^{(k-1)}]}\right]}$, *i.e.*, $\hat{F}^{(k)} = \hat{F}_{\left[\rho_{[\Phi^{(k-1)}]}\right]}$. This allows explicit calculation of $V_{eff}^{(k)}$ as $V_{eff_{C^{(k-1)}}}$ based on Eq. (21) and $F^{(k)}$ as $F_{C^{(k-1)}}$, for which the following series of definitions is introduced:

$$\mathbf{F}_{\mathbf{C}} := [\langle \eta_\alpha | \hat{F}_{[\rho_{\mathbf{C}}]} | \eta_\beta \rangle]_{\alpha\beta} = \mathbf{T} + \mathbf{V}_{\mathrm{eff}\mathbf{C}} \quad (\mathbf{T} \text{ defined in Eq. (31)}).$$

$$\mathbf{V}_{\mathrm{eff}\mathbf{C}} := [\langle \eta_\alpha | V_{\mathrm{eff}[\rho_{\mathbf{C}}]} | \eta_\beta \rangle]_{\alpha\beta} = \mathbf{V}_{\mathrm{H}\mathbf{C}} + \mathbf{V}_{\mathrm{XC}\mathbf{C}} + \mathbf{V}_{\mathrm{ext}}. \tag{34}$$

where $(\mathbf{V}_{\mathrm{H}\mathbf{C}})_{\alpha\beta} := \langle \eta_\alpha | V_{\mathrm{H}[\rho_{\mathbf{C}}]} | \eta_\beta \rangle \overset{\text{Eqs. (21, 32)}}{=} \iint \eta_\alpha(\mathbf{r}) \eta_\beta(\mathbf{r}) \frac{\sum_{\gamma\delta} \mathbf{\Gamma}_{\gamma\delta} \eta_\gamma(\mathbf{r}') \eta_\delta(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} \mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{r}$

$$= \sum_{\gamma\delta} \hat{\mathbf{D}}_{\alpha\beta,\gamma\delta} \mathbf{\Gamma}_{\gamma\delta} = (\hat{\mathbf{D}}\bar{\mathbf{\Gamma}})_{\alpha\beta}. \tag{35}$$

$$\text{where } \hat{\mathbf{D}}_{\alpha\beta,\gamma\delta} := (\eta_\alpha \eta_\beta | \eta_\gamma \eta_\delta). \ \bar{\mathbf{\Gamma}} \text{ is the vector of flattened } \mathbf{\Gamma}. \tag{36}$$

$$(\mathbf{V}_{\mathrm{XC}\mathbf{C}})_{\alpha\beta} := \langle \eta_\alpha | V_{\mathrm{XC}[\rho_{\mathbf{C}}]} | \eta_\beta \rangle = \int V_{\mathrm{XC}[\rho_{\mathbf{C}}]}(\mathbf{r}) \eta_\alpha(\mathbf{r}) \eta_\beta(\mathbf{r}) \, \mathrm{d}\mathbf{r}.$$

$$(\mathbf{V}_{\mathrm{ext}})_{\alpha\beta} := \langle \eta_\alpha | V_{\mathrm{ext}} | \eta_\beta \rangle = -\sum_{a=1}^{A} Z^{(a)} \int \frac{\eta_\alpha(\mathbf{r}) \eta_\beta(\mathbf{r})}{\|\mathbf{r} - \mathbf{x}^{(a)}\|} \mathrm{d}\mathbf{r}.$$

In defining $\tilde{\mathbf{D}}$, the notation of Coulomb integral is used for brevity:

$$(f|g) := \int \frac{f(\mathbf{r}) g(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} \, \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'. \tag{37}$$

[00141] In practice, integrals in $\mathbf{S}$, $\mathbf{T}$, $\mathbf{V}_{\mathrm{ext}}$, and the Coulomb integral $\tilde{\mathbf{D}}$ can be calculated analytically under Gaussian-Type Orbitals (GTO) as the basis $\{\eta_a\}_{a=1}^{B}$. The integral in $\mathbf{V}_{\mathrm{XC}\mathbf{C}}$ is conducted numerically on a quadrature grid, as typically used in a DFT calculation. After convergence, the total energy can be calculated from the orbital coefficients $\mathbf{C}$ by:

$$E(\mathbf{C}) := E[\Phi_{\mathbf{C}}] \overset{\text{Eq. (10)}}{=} T_{\mathrm{S}}(\mathbf{C}) + \underbrace{E_{\mathrm{H}}(\mathbf{C}) + E_{\mathrm{XC}}(\mathbf{C}) + E_{\mathrm{ext}}(\mathbf{C})}_{=:E_{\mathrm{eff}}(\mathbf{C})}. \tag{38}$$

where $$T_{\mathrm{S}}(\mathbf{C}) := \sum_{i=1}^{N} \langle \phi_{i,\mathbf{C}} | \hat{T} | \phi_{i,\mathbf{C}} \rangle = \sum_{\alpha\beta} \mathbf{\Gamma}_{\alpha\beta} \mathbf{T}_{\alpha\beta} = \bar{\mathbf{\Gamma}}^{\top} \bar{\mathbf{T}}. \tag{39}$$

$$E_{\mathrm{H}}(\mathbf{C}) := E_{\mathrm{H}}[\rho_{\mathbf{C}}] = \frac{1}{2} \sum_{\alpha\beta\gamma\delta} \mathbf{\Gamma}_{\alpha\beta} \hat{\mathbf{D}}_{\alpha\beta,\gamma\delta} \mathbf{\Gamma}_{\gamma\delta} = \frac{1}{2} \bar{\mathbf{\Gamma}}^{\top} \hat{\mathbf{D}} \bar{\mathbf{\Gamma}}. \tag{40}$$

$$E_{\mathrm{XC}}(\mathbf{C}) := E_{\mathrm{XC}}[\rho_{\mathbf{C}}] = E_{\mathrm{XC}}\Big[\sum_{\alpha\beta} \mathbf{\Gamma}_{\alpha\beta} \eta_{\alpha} \eta_{\beta} \Big]. \tag{41}$$

$$E_{\mathrm{ext}}(\mathbf{C}) := E_{\mathrm{ext}}[\rho_{\mathbf{C}}] = \sum_{\alpha\beta} \mathbf{\Gamma}_{\alpha\beta} (\mathbf{V}_{\mathrm{ext}})_{\alpha\beta} = \bar{\mathbf{\Gamma}}^{\top} \bar{\mathbf{V}}_{\mathrm{ext}}. \tag{42}$$

where $\bar{\mathbf{\Gamma}}$, $\bar{\mathbf{T}}$, $\bar{\mathbf{V}}_{ext}$, are the vectors of flattened $\mathbf{\Gamma}$, $\mathbf{T}$, $\mathbf{V}_{\mathrm{ext}}$, respectively. The term $E_{\mathrm{XC}}[\rho_{\mathbf{C}}]$ is again calculated by numerically integrating the defined density by $\mathbf{C}$ on a quadrature grid.

[00142] Computational Complexity

[00143] Note that the construction of $\mathbf{V}_{\mathrm{HC}}$ from Eq. (35) and the evaluation of $E_{\mathrm{H}}(\mathbf{C})$ from Eq. (40) require $O(B^4) = O(N^4)$ complexity. Even when using density fitting which decreases the complexity to $O(N^2)$, the complexity in each SCF iteration of KSDFT is $O(N^3)$ since the complexity of density fitting itself is $O(N^3)$ (see Supplementary Sec. A.4.1).

[00144] Orbital Orthonormality

[00145] Under the atomic basis, orbital orthonormality $\langle \phi_i | \phi_j \rangle = \delta_{ij}$ becomes $\sum_{\alpha\beta} C_{\alpha i} C_{\beta j} \langle \eta_{\alpha} | \eta_{\beta} \rangle = \delta_{ij}$, or in matrix form,

$$\mathbf{C}^{\mathsf{T}}\mathbf{S}\mathbf{C} = \mathbf{I}. \tag{43}$$

As mentioned after Eq. (22), only the normalization constraints need to be taken care of, as the orbitals are eigenstates of an Hermitian operator hence are already orthogonal if non-degenerate. This property transmits to the matrix form of the problem:

$$\mathbf{C}_{:i}^{\top} \mathbf{S} \mathbf{C}_{:i} = 1. \quad \forall i = 1. \cdots . N. \tag{44}$$

Fulfilling this constraint only needs to normalize each eigenvector $\tilde{C}_{:i}^{(k)}$ of the problem in Eq. (30) to form $C^{(k)}$; explicitly $C_{:i}^{(k)} = \tilde{C}_{:i}^{(k)} / \sqrt{\tilde{C}_{:i}^{(k)\mathrm{T}} S \tilde{C}_{:i}^{(k)}}$.

[00146] Relation to Direct Gradient Derivation

[00147] The matrix form of the optimality equation under basis hence the SCF iteration problem Eq. (30) can also be derived directly from Eq. (19) by taking the gradient of the energy function of coefficients: $E(C) := \mathrm{E}[\Phi_C] = E\left[\left\{\sum_\alpha C_{\alpha i}\eta_\alpha\right\}_{i=1}^{N}\right]$. Its gradient is related to the variation of the functional of orbitals by integral with the basis:

$$\left(\nabla_{\mathbf{C}} E(\mathbf{C})\right)_{\alpha i} = \int \frac{\delta E[\Phi_{\mathbf{C}}]}{\delta \phi_i}(\mathbf{r})\left(\nabla_{\mathbf{C}}\phi_{i,\mathbf{C}}(\mathbf{r})\right)_{\alpha i}\,\mathrm{d}\mathbf{r} = \int \frac{\delta E[\Phi_{\mathbf{C}}]}{\delta \phi_i}(\mathbf{r})\eta_\alpha(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{45}$$

The variation is given by Eq. (20), which is $2\hat{F}_{[\rho_{\Phi_C}]}\phi_{i,C}(r) = 2\sum_\beta C_{\alpha i}\hat{F}_{[\rho_{\Phi_C}]}\eta_\beta(r)$, which turns the gradient into matrix form:

$$\nabla_{\mathbf{C}} E(\mathbf{C}) = 2\mathbf{F}_{\mathbf{C}}\mathbf{C}. \tag{46}$$

For the orbital orthonormality constraint, as mentioned, only the normalization constraints require explicit treatment. By introducing Lagrange multiplier ε_i for each constraint in Eq. (44) and taking the gradient for the corresponding Lagrange term gives $\nabla_C \sum_{i=1}^{N} \varepsilon_i \left(C_{:i}^{\mathrm{T}} S C_{:i} - 1\right) = 2\mathrm{SC}_\varepsilon$. This leads to the optimality equation in matrix form:

$$\mathbf{F}_{\mathbf{C}}\mathbf{C} = \mathbf{S}\mathbf{C}\varepsilon. \tag{47}$$

By constructing the corresponding fixed-point iteration, Eq. (30) is derived.

[00148] Accelerating and Stabilizing SCF Iteration

[00149] As mentioned, $\mathbf{F}^{(k)}$ and $\mathbf{V}_{\mathrm{eff}}^{(k)}$ may be taken differently from $\mathbf{F}_{\mathbf{C}^{(k-1)}}$ and $\mathrm{V}_{effC^{(k-1)}}$ for more stable and faster convergence. The direct inversion in the iterative subspace (DIIS) method is one choice for this. In DIIS, the Fock matrix $\mathbf{F}^{(k)}$ in the eigenvalue problem Eq. (24) for each SCF iteration k is taken as a weighted mixing of the standard Fock matrices $\mathbf{F}_{\mathbf{C}^{(k')}}$, $k' < k$ in previous steps:

$$\mathbf{F}^{(k)} := \sum_{k'=0}^{k-1} \pi_{k'}^{(k)} \mathbf{F}_{\mathbf{C}^{(k')}}.$$

where $\left\{\pi_{k'}^{(k)}\right\}_{k'=0}^{k-1}$ are the weights that are positive and normalized $\sum_{k'=0}^{k-1} \pi_{k'}^{(k)} = 1$.

Due to the normalization, the kinetic part $\mathbf{T}$ of the matrix remains the same, so it agrees with the form in Eq. (31) (or Eq. (23) in operator form), where:

$$\mathbf{V}_{\text{eff}}^{(k)} := \sum_{k'=0}^{k-1} \pi_{k'}^{(k)} \mathbf{V}_{\text{eff}\,\mathbf{C}^{(k')}}. \tag{48}$$

[00150] A.3.2 OFDFT under Atomic Basis

[00151] To solve the optimization problem of OFDFT in Eq. (14), it is unnatural to construct a fixed-point SCF iteration process from its variation in Eq. (27). Hence, direct gradient-based density optimization is conducted. For this, the energy functional of density function in Eq. (14) needs to be converted into a function of density coefficients using the basis expansion of density function in Eq. (29):

$$E(\mathbf{p}) := E[\rho_{\mathbf{p}}] = E\Big[\sum_{\mu} \mathbf{p}_\mu \omega_\mu\Big] = T_{\text{S}}(\mathbf{p}) + \underbrace{E_{\text{H}}(\mathbf{p}) + E_{\text{XC}}(\mathbf{p}) + E_{\text{ext}}(\mathbf{p})}_{=E_{\text{ee}}(\mathbf{p}):=E_{\text{ee}}[\rho_{\mathbf{p}}]}. \tag{49}$$

where $T_{\text{S}}(\mathbf{p}) := T_{\text{S}}[\rho_{\mathbf{p}}] = T_{\text{S}}\Big[\sum_{\mu} \mathbf{p}_\mu \omega_\mu\Big]$. (50)

$$E_{\text{H}}(\mathbf{p}) := E_{\text{H}}[\rho_{\mathbf{p}}] = \frac{1}{2}\iint \frac{\sum_\mu \mathbf{p}_\mu \omega_\mu(\mathbf{r}) \sum_\nu \mathbf{p}_\nu \omega_\nu(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|} \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' = \frac{1}{2}\mathbf{p}^\top \tilde{\mathbf{W}} \mathbf{p}. \tag{51}$$

$$E_{\text{XC}}(\mathbf{p}) := E_{\text{XC}}[\rho_{\mathbf{p}}] = E_{\text{XC}}\Big[\sum_{\mu} \mathbf{p}_\mu \omega_\mu\Big]. \tag{52}$$

$$E_{\text{ext}}(\mathbf{p}) := E_{\text{ext}}[\rho_{\mathbf{p}}] = \int \sum_{\mu} \mathbf{p}_\mu \omega_\mu(\mathbf{r}) V_{\text{ext}}(\mathbf{r})\, \mathrm{d}\mathbf{r} = \mathbf{p}^\top \mathbf{v}_{\text{ext}}. \tag{53}$$

where $\tilde{\mathbf{W}}_{\mu\nu} := (\omega_\mu|\omega_\nu)$. $(\mathbf{v}_{\text{ext}})_\mu := \langle \omega_\mu | V_{\text{ext}} \rangle$.

[00152] The Coulomb integral notation $(\omega_\mu|\omega_\nu)$ is defined in Eq. (37). Recall that the dependency of density basis $\{\omega_\mu\}_\mu$ has been omitted below from the functions *e.g.* $T_{\text{S}}(\mathbf{p})$ on the molecular structure M in this subsection. Integrals for $\tilde{\mathbf{W}}$ and $\mathbf{v}_{\text{ext}}$ can be calculated efficiently under Gaussian-Type Orbitals (GTO) as the basis $\{\omega_\mu\}_\mu$, using software libraries *e.g.* libcint in PySCF. The term $E_{\text{XC}}[\rho_{\mathbf{p}}]$ is calculated by numerically

integrating the defined density $\rho_{\mathbf{p}}$ on a quadrature grid as typically used in a DFT calculation. In M-OFDFT, $T_S(\mathbf{p})$ is calculated directly from the coefficient $\mathbf{p}$ using the machine-learning model $T_{S,\theta}(\mathbf{p},M)$.

[00153] To carry out direct optimization using a learned KEDF model $T_{S,\theta}(\mathbf{p})$, the gradient of the energy function in Eq. (49) is required, which is given by:

$$\nabla_{\mathbf{p}} E_\theta(\mathbf{p}) = \nabla_{\mathbf{p}} T_{S,\theta}(\mathbf{p}) + \underbrace{\tilde{\mathbf{W}}\mathbf{p} + \nabla_{\mathbf{p}} E_{XC}(\mathbf{p}) + \mathbf{v}_{ext}}_{\nabla_{\mathbf{p}} E_{eff}(\mathbf{p})}. \tag{54}$$

This gradient is then used to update the density coefficient after projected onto the linear subspace of normalized densities, following Eq. (1).

[00154] Relation to Derivation as Integral of the Variation with Basis

[00155] The gradient $\nabla_{\mathbf{p}} E(\mathbf{p})$ can also be derived by the relation between gradient and variation that has already been seen in Eq. (45):

$$\left(\nabla_{\mathbf{p}} E(\mathbf{p})\right)_\mu = \left(\nabla_{\mathbf{p}} E[\rho_{\mathbf{p}}]\right)_\mu = \int \frac{\delta E[\rho_{\mathbf{p}}]}{\delta \rho}(\mathbf{r}) \left(\nabla_{\mathbf{p}} \rho_{\mathbf{p}}(\mathbf{r})\right)_\mu \mathrm{d}\mathbf{r} = \int \frac{\delta E[\rho_{\mathbf{p}}]}{\delta \rho}(\mathbf{r}) \omega_\mu(\mathbf{r}) \mathrm{d}\mathbf{r}. \tag{55}$$

Integrating the variations given in Eq. (21) with the basis functions $\{\omega_\mu\}_\mu$ recovers the Hartree energy gradient $\tilde{\mathbf{W}}\mathbf{p}$ and external energy gradient $\mathbf{v}_{ext}$ in Eq. (54). The formula also applies to the gradient of the kinetic energy $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ and the gradient of the XC energy $\nabla_{\mathbf{p}} E_{XC}(\mathbf{p})$.

[00156] Automatic Differentiation Implementation for the Gradient

[00157] In the implementation of M-OFDFT, the gradient of the KEDF model $\nabla_{\mathbf{p}} T_{S,\theta}(\mathbf{p},M)$ is evaluated directly using automatic differentiation, which can be conveniently done if implementing the model using common machine-learning programming frameworks, *e.g.*, PyTorch. To calculate $\nabla_{\mathbf{p}} E_{XC}(\mathbf{p})$ conveniently, the PBE XC functional in PySCF using PyTorch was re-implemented and its gradient also evaluated by automatic differentiation. When using the residual version $T_{S,res,\theta}$ of KEDF

model, which is detailed in Eq. (67) in Supplementary Sec. B.4 later, the base KEDF (taken as the APBE KEDF) is also implemented in this way.

[00158] Computational Complexity

[00159] As will be detailed in Supplementary Sec. B, the transformer-based KEDF model for M-OFDFT has a quadratic complexity $O(A^2) = O(N^2)$. The PBE functional for E_{XC} and the APBE functional for the base KEDF are at the GGA level (generalized gradient approximation), so evaluating the energies amounts to calculating the density features with $O(M)$ cost on each grid point, in total with $O(MN_{\mathrm{grid}})$ cost where N_{grid} is the number of grid points, and then conducting the quadrature with $O(N_{\mathrm{grid}})$ cost. The complexity for these energies is thus $O(MN_{\mathrm{grid}})$ which is also quadratic $O(N^2)$ since $N_{\mathrm{grid}} = O(N)$ (though with a large prefactor). Evaluating the gradient using automatic differentiation is in the same order of evaluating the function, hence also has $O(N^2)$ complexity. Evaluating $E_{\mathrm{H}}(\mathbf{p})$ and $E_{\mathrm{ext}}(\mathbf{p})$ using Eq. (51) and Eq. (53) and their gradients using Eq. (54) require $O(M^2) = O(N^2)$ complexity. Therefore, the complexity in M-OFDFT has a quadratic complexity $O(N^2)$, which is indeed lower than that of KSDFT (detailed in Supplementary Sec. A.3.1 above).

[00160] Besides the advantage in asymptotic complexity, the fact that M-OFDFT is implemented in PyTorch enables it to leverage GPUs efficiently. These factors jointly facilitate the much higher throughput of M-OFDFT than KSDFT.

[00161] A.4 Label Calculation under Atomic Basis

[00162] This subsection details the calculation of the data tuple $\left(\mathrm{p}^{(k)}, T_{\mathrm{S}}^{(k)}, \nabla_p T_{\mathrm{S}}^{(k)}\right)$ for learning a KEDF model from the orbital coefficients $\mathbf{C}^{(k)}$ of the orbital solution $\Phi^{(k)} := \left\{\phi_i{}^{(k)}\right\}_{i=1}^{N}$ in each SCF iteration k. Following the previous subsection, omit the appearance of M for density and orbital representation has been omitted (*e.g.*, in $T_{\mathrm{S}}(\mathbf{p},\mathrm{M})$) and omit the index d for different molecular systems. The k

index is kept to reflect that the deduction is based on the solution in an SCF iteration but does not apply to arbitrary orbital coefficients **C**.

[00163] A.4.1 Density Fitting

[00164] The process starts with calculating the density coefficient $\mathbf{p}^{(k)}$ under the density basis $\{\omega_\mu(r)\}_{\mu=1}^{M}$ for representing the density defined by the orbital coefficient solution $\mathbf{C}^{(k)}$. This process is called *density fitting*, which is also used in KSDFT for acceleration, in which context the atomic basis for the density is also called *auxiliary basis*. The density coefficient $\mathbf{p}^{(k)}$ needs to fit represented density $\rho_{\mathbf{p}^{(k)}}$ by Eq. (29) to the density $\rho_{\mathbf{C}^{(k)}}$ defined by Eq. (32). Noting that Eq. (32) essentially expands the density onto the paired basis $\{\eta_\alpha(\mathbf{r})\eta_\beta(\mathbf{r})\}_{\alpha\beta}$ with coefficient as the vector $\bar{\mathbf{\Gamma}}^{(k)}$ of flattened $\mathbf{\Gamma}^{(k)}$, this is the classical coordinate transformation problem from the paired basis to the density basis. The classical approach is by minimizing the standard L2 norm of the residue density:

$$\int \left|\rho_{\mathbf{p}^{(k)}}(\mathbf{r}) - \rho_{\mathbf{C}^{(k)}}(\mathbf{r})\right|^2 \mathrm{d}\mathbf{r} = \mathbf{p}^{(k)\top}\mathbf{W}\mathbf{p}^{(k)} - 2\mathbf{p}^{(k)\top}\mathbf{L}\bar{\mathbf{\Gamma}}^{(k)} + \bar{\mathbf{\Gamma}}^{(k)\top}\mathbf{D}\bar{\mathbf{\Gamma}}^{(k)}.$$

where $\mathbf{W}_{\mu\nu} := \langle\omega_\mu|\omega_\nu\rangle$, $\mathbf{L}_{\mu,\alpha\beta} := \langle\omega_\mu|\eta_\alpha\eta_\beta\rangle$, and $\mathbf{D}_{\alpha\beta,\gamma\delta} := \langle\eta_\alpha\eta_\beta|\eta_\gamma\eta_\delta\rangle$ are the overlap matrices of the density basis, between the density basis and the paired basis, and of the paired basis, respectively.

[00165] Noting that this is a quadratic form of $\mathbf{p}^{(k)}$, the solution is

$$\mathbf{p}^{(k)} = \mathbf{W}^{-1}\mathbf{L}\bar{\mathbf{\Gamma}}^{(k)}$$

[00166] However, the standard L2 metric on the density function may not be the most favorable metric for density fitting. Instead, energy is the directly concerned quantity. The kinetic energy is the most desired metric, since this would minimize the mismatch of the fitted density $\mathbf{p}^{(k)}$ to the kinetic energy label $T_S^{(k)}$. But there is no explicit expression to calculate the kinetic energy from density coefficient. Hence using the Hartree energy and the external energy as the metric is considered. (Using the XC

energy requires an arbitrary choice of a functional approximation, and the calculation is more costly.)

[00167] For the Hartree energy as defined in Eq. (12), noting that it is quadratic in density, $\mathbf{p}$ is fit by minimizing the (2×) Hartree energy arising from the residue density:

$$E_{\mathrm{H}}[\rho_{\mathbf{p}^{(k)}} - \rho_{\mathbf{C}^{(k)}}] = \iint \frac{\left(\rho_{\mathbf{p}^{(k)}}(\mathbf{r}) - \rho_{\mathbf{C}^{(k)}}(\mathbf{r})\right)\left(\rho_{\mathbf{p}^{(k)}}(\mathbf{r}') - \rho_{\mathbf{C}^{(k)}}(\mathbf{r}')\right)}{\|\mathbf{r} - \mathbf{r}'\|} \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'$$
$$= \mathbf{p}^{(k)\top}\tilde{\mathbf{W}}\mathbf{p}^{(k)} - 2\mathbf{p}^{(k)\top}\tilde{\mathbf{L}}\bar{\mathbf{\Gamma}}^{(k)} + \bar{\mathbf{\Gamma}}^{(k)\top}\tilde{\mathbf{D}}\bar{\mathbf{\Gamma}}^{(k)}.$$

where symbols with tilde are the corresponding overlap matrices under integral kernel $\frac{1}{\|r-r'\|}$, which, using the symbol of Coulomb integral defined in Eq. (37), are $\mathbf{W}_{\mu\nu} := (\omega_\mu|\omega_\nu)$, $\mathbf{L}_{\mu,\alpha\beta} := (\omega_\mu|\eta_\alpha\eta_\beta)$, and $\mathbf{D}_{\alpha\beta,\gamma\delta} := (\eta_\alpha\eta_\beta|\eta_\gamma\eta_\delta)$ as already defined in Eq. (36). As a quadratic form, the solution is $\mathbf{p}^{(k)} = \tilde{\mathbf{W}}^{-1}\tilde{\mathbf{L}}\bar{\mathbf{\Gamma}}^{(k)}$. This result can be understood as if the Hartree energy (Coulomb integral) defines a metric on the space of density functions.

[00168] For the external energy as defined in Eq. (8), as it is linear in density, $\mathbf{p}$ is fit by directly minimizing the difference between the defined external energies:

$$\left(E_{\mathrm{ext}}[\rho_{\mathbf{p}^{(k)}}] - E_{\mathrm{ext}}[\rho_{\mathbf{C}^{(k)}}]\right)^2 = \left(E_{\mathrm{ext}}(\mathbf{p}^{(k)}) - E_{\mathrm{ext}}(\mathbf{C}^{(k)})\right)^2 \overset{\text{Eqs. (53, 42)}}{=} (\mathbf{p}^{(k)\top}\mathbf{v}_{\mathrm{ext}} - \bar{\mathbf{\Gamma}}^{(k)\top}\bar{\mathbf{V}}_{\mathrm{ext}})^2.$$

To combine the two metrics, the final optimization problem is a combined least-square problem:

$$\mathbf{p}^{(k)} = \underset{\mathbf{p}}{\operatorname{argmin}}\, \mathbf{p}^\top\tilde{\mathbf{W}}\mathbf{p} - 2\mathbf{p}^\top\tilde{\mathbf{L}}\bar{\mathbf{\Gamma}}^{(k)} + \bar{\mathbf{\Gamma}}^{(k)\top}\tilde{\mathbf{D}}\bar{\mathbf{\Gamma}}^{(k)} + (\mathbf{p}^\top\mathbf{v}_{\mathrm{ext}} - \bar{\mathbf{\Gamma}}^{(k)\top}\bar{\mathbf{V}}_{\mathrm{ext}})^2.$$

which corresponds to the over-determined linear equations in matrix form:

$$\mathbf{W}\mathbf{p}^{(k)} = \mathbf{b}^{(k)}. \quad \text{where } \mathbf{W} := \begin{pmatrix} \tilde{\mathbf{W}} \\ \mathbf{v}_{\mathrm{ext}}^\top \end{pmatrix}. \mathbf{b}^{(k)} := \begin{pmatrix} \tilde{\mathbf{L}}\bar{\mathbf{\Gamma}}^{(k)} \\ \bar{\mathbf{\Gamma}}^{(k)\top}\bar{\mathbf{V}}_{\mathrm{ext}} \end{pmatrix}.$$

This is directly solved using least-square solvers. In this conversion, no explicit consideration was made of the normalization constraint, $\mathbf{p}^{(k)\top}\mathrm{w} = N$, since it is already satisfied with a high accuracy, due to the close fit to the original density.

[00169] A.4.2 Value Label Calculation

[00170] The label for the value of KEDF can be calculated from Eq. (26) by leveraging the expression Eq. (39) under atomic basis:

$$T_{\mathrm{S}}(\mathbf{C}^{(k)}) = \sum_{\alpha\beta} \mathbf{\Gamma}^{(k)}_{\alpha\beta}\mathbf{\Gamma}_{\alpha\beta} = \bar{\mathbf{\Gamma}}^{(k)\top}\bar{\mathbf{\Gamma}}.$$

where $\mathbf{\Gamma}^{(k)} = \mathbf{C}^{(k)}\mathbf{C}^{(k)\top}$ from Eq. (33), and $\bar{\mathbf{\Gamma}}^{(k)}$ is the vector by flattening. The corresponding density coefficient $\mathbf{p}^{(k)}$ is calculated from $\mathbf{C}^{(k)}$ using density fitting as detailed above. A subtlety arises since the fitted density $\mathbf{p}^{(k)}$ may differ a little from the original density defined by $\mathbf{C}^{(k)}$ due to finite-basis incompleteness, so $T_{\mathrm{S}}(\mathbf{C}^{(k)})$ may not be the best kinetic energy label for $\mathbf{p}^{(k)}$. Indeed, as mentioned, in density fitting, there is no known way to directly find the density coefficient $\mathbf{p}^{(k)}$ that achieves the kinetic energy closest to $T_{\mathrm{S}}(\mathbf{C}^{(k)})$. Instead, $\mathbf{p}^{(k)}$ is fitted to match the Hartree and external energy. It is hence assumed that the total energy is less affected by density-fitting error than the kinetic energy, and instead of taking $T_{\mathrm{S}}(\mathbf{p}^{(k)}) \approx T_{\mathrm{S}}(\mathbf{C}^{(k)})$, $E(\mathbf{p}^{(k)}) \approx E(\mathbf{C}^{(k)})$ is taken, which means $T_{\mathrm{S}}(\mathbf{p}^{(k)}) + E_{\mathrm{eff}}(\mathbf{p}^{(k)}) \approx T_{\mathrm{S}}(\mathbf{C}^{(k)}) + E_{\mathrm{eff}}(\mathbf{C}^{(k)})$. Hence the KEDF value label for $\mathbf{p}^{(k)}$ is taken as:

$$T_{\mathrm{S}}^{(k)} := T_{\mathrm{S}}(\mathbf{C}^{(k)}) + E_{\mathrm{eff}}(\mathbf{C}^{(k)}) - E_{\mathrm{eff}}(\mathbf{p}^{(k)}).$$

$$\text{where} \quad E_{\mathrm{eff}}(\mathbf{C}^{(k)}) = \underbrace{\frac{1}{2}\bar{\mathbf{\Gamma}}^{(k)\top}\tilde{\mathbf{D}}\bar{\mathbf{\Gamma}}^{(k)}}_{E_{\mathrm{H}}(\mathbf{C}^{(k)})} + E_{\mathrm{XC}}(\mathbf{C}^{(k)}) + \underbrace{\bar{\mathbf{\Gamma}}^{(k)\top}\bar{\mathbf{V}}_{\mathrm{ext}}}_{E_{\mathrm{ext}}(\mathbf{C}^{(k)})},$$

$$E_{\mathrm{eff}}(\mathbf{p}^{(k)}) = \underbrace{\frac{1}{2}\mathbf{p}^{(k)\top}\tilde{\mathbf{W}}\mathbf{p}^{(k)}}_{E_{\mathrm{H}}(\mathbf{p}^{(k)})} + E_{\mathrm{XC}}(\mathbf{p}^{(k)}) + \underbrace{\mathbf{p}^{(k)\top}\mathbf{v}_{\mathrm{ext}}}_{E_{\mathrm{ext}}(\mathbf{p}^{(k)})},$$

following definitions and expressions of Eqs. (38-42) and Eqs. (49-53). Labels for the two variants of functional models detailed in Supplementary Sec. B.4 can be calculated accordingly. For the residual version of KEDF $T_{\mathrm{S,res}}$, the label is correspondingly modified using values at $\mathbf{p}^{(k)}$:

$$T_{\mathrm{S,res}}^{(k)} := T_{\mathrm{S}}^{(k)} - T_{\mathrm{APBE}}(\mathbf{p}^{(k)}).$$

For the version E_{TXC} that learns the sum of KEDF and the XC energy, the corresponding label is: $T_{\mathrm{S}}^{(k)} + E_{\mathrm{XC}}(p^{(k)}) = T_{\mathrm{S}}(\mathrm{C}^{(k)}) + E_{\mathrm{H}}(\mathrm{C}^{(k)}) + E_{\mathrm{XC}}(\mathrm{C}^{(k)}) + E_{\mathrm{ext}}(\mathrm{C}^{(k)}) - \left(E_{\mathrm{H}}(\mathrm{p}^{(k)}) + E_{\mathrm{XC}}(\mathrm{p}^{(k)}) + E_{\mathrm{ext}}(\mathrm{p}^{(k)})\right) + E_{\mathrm{XC}}(\mathrm{p}^{(k)}) = T_{\mathrm{S}}(\mathrm{C}^{(k)}) + E_{\mathrm{XC}}(\mathrm{C}^{(k)}) + \left(E_{\mathrm{H}}(\mathrm{C}^{(k)}) - E_{\mathrm{H}}(\mathrm{p}^{(k)}) + E_{\mathrm{ext}}(\mathrm{C}^{(k)}) - E_{\mathrm{ext}}(\mathrm{p}^{(k)})\right)$, where the last term can be omitted since the Hartree energy difference and the external energy difference are minimized by $\mathbf{p}^{(k)}$ in density fitting. The label is thus taken as:

$$E_{\mathrm{TXC}}^{(k)} := T_{\mathrm{S}}(\mathbf{C}^{(k)}) + E_{\mathrm{XC}}(\mathbf{C}^{(k)})$$

[00171] A.4.3 Gradient Label Calculation

[00172] Under an atomic basis, the kinetic energy functional of density is converted into a function of density coefficient $T_{\mathrm{S}}(\mathbf{p}) := T_{\mathrm{S}}[\rho_{\mathrm{p}}]$ following Eq. (50). For its gradient $\nabla_{\mathbf{p}}T_{\mathrm{S}}(\mathbf{p})$, following the fact in Eq. (55), it is related to the variation of the functional $T_{\mathrm{S}}[\rho]$ by integral with the basis:

$$\left(\nabla_{\mathrm{p}} T_{\mathrm{S}}(\mathbf{p})\right)_{\mu} = \int \frac{\delta T_{\mathrm{S}}[\rho_{\mathrm{p}}]}{\delta \rho}(\mathbf{r})\left(\nabla_{\mathbf{p}} \rho_{\mathrm{p}}(\mathbf{r})\right)_{\mu} \mathrm{d}\mathbf{r} = \int \frac{\delta T_{\mathrm{S}}[\rho_{\mathrm{p}}]}{\delta \rho}(\mathbf{r}) \omega_{\mu}(\mathbf{r}) \mathrm{d}\mathbf{r}$$

The variation corresponding to a known density is given by Eq. (27) above, which comes from the solution of orbitals in a KSDFT SCF iteration. If omitting the error in density fitting and approximating $\frac{\delta}{\delta\rho} T_{\mathrm{S}}\left[\rho_{\mathrm{p}^{(k)}}\right]$ with $\frac{\delta}{\delta\rho} T_{\mathrm{S}}\left[\rho_{\mathrm{C}^{(k)}}\right] = \frac{\delta}{\delta\rho} T_{\mathrm{S}}\left[\rho_{[\Phi^{(k)}]}\right]$, then the gradient can be accessed by integrating both sides of Eq. (27) with the density basis:

$$\nabla_{\mathrm{p}} T_{\mathrm{S}}(\mathrm{p}^{(k)}) + \mathbf{v}_{\mathrm{eff}}^{(k)} = \mu^{(k)} \mathbf{w}. \tag{56}$$

$$\text{where} \quad (\mathbf{v}_{\mathrm{eff}}^{(k)})_{\mu} := \langle \omega_{\mu} | V_{\mathrm{eff}}^{(k)} \rangle = \int V_{\mathrm{eff}}^{(k)}(\mathbf{r}) \omega_{\mu}(\mathbf{r}) \mathrm{d}\mathbf{r}. \quad \mathrm{w}_{\mu} := \int \omega_{\mu}(\mathbf{r}) \mathrm{d}\mathbf{r}. \tag{57}$$

In practice, the chemical potential $\mu^{(k)}$ is not needed, since the gradient matters in its projection on the tangent space of normalized densities in order to keep the density normalized in density optimization (see Eq. (1)). The space of normalized densities is a linear space $\{\mathbf{p}|\mathbf{p}^{\top}\mathbf{w} = N\}$ since $\int \rho_{\mathrm{p}}(\boldsymbol{r})\, d\boldsymbol{r} = \sum_{\mu} \mathrm{p}_{\mu} \int \mathrm{w}_{\mu}(\boldsymbol{r})\, d\boldsymbol{r} = N$, so it coincides with its tangent space. The projection onto the tangent space is achieved by applying $\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^{\top}}{\mathbf{w}^{\top}\mathbf{w}}$, which gives:

$$\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^{\top}}{\mathbf{w}^{\top}\mathbf{w}}\right) \nabla_{\mathrm{p}} T_{\mathrm{S}}(\mathrm{p}^{(k)}) = -\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^{\top}}{\mathbf{w}^{\top}\mathbf{w}}\right) \mathbf{v}_{\mathrm{eff}}^{(k)}. \tag{58}$$

Due to the same reason, the gradient loss function Eq. (3) also only matches the projected gradient of the model to the projected gradient label.

[00173] The remaining task is to evaluate $\mathbf{v}_{\mathrm{eff}}^{(k)}$ in Eq. (57). Considering the complication that $\mathbf{V}_{\mathrm{eff}}^{(k)}(\mathbf{r})$ may not be taken as the explicit-form effective potential $\mathrm{V}_{eff\left[\rho_{[\Phi^{(k-1)}]}\right]}(r) = \mathrm{V}_{eff\left[\rho_{[\mathrm{C}^{(k-1)}]}\right]}$(Eq. (21) and Eq. (32)), such as when DIIS (Eq. (48)) is used in SCF iteration, evaluating $\mathbf{v}_{\mathrm{eff}}$ can be done by leveraging the orbital-basis representation $\mathrm{V}_{eff}^{(k)}$ already available in the SCF problem (Eq. (30)), which is defined in Eq. (31) as the integral with paired orbital basis: $\left(\mathrm{V}_{eff}^{(k)}\right)_{\alpha\beta} := \int \mathrm{V}_{eff}^{(k)}(r)\eta_{\alpha}(r)\eta_{\beta}(r)\, \mathbf{dr}$. Using the expansion coefficients $\mathbf{K}$ of the density basis

onto the paired orbital basis, that is $\omega_\mu(\mathbf{r}) = \sum_{\alpha\beta} \mathbf{K}_{\mu,\alpha\beta}\eta_\alpha(\mathbf{r})\eta_\beta(\mathbf{r})$, the conversion can be conducted by $\left(\mathrm{V}_{eff}^{(k)}\right)_\mu = \sum_{\alpha\beta} \mathrm{K}_{\mu,\alpha\beta}\left(\mathrm{V}_{eff}^{(k)}\right)_{\alpha\beta}$.

[00174] However, solving for **K** is unaffordable: using least-square, this amounts to solving $\mathbf{DK} = \mathbf{L}$ (or solving $\tilde{\mathbf{D}}\mathbf{K} = \tilde{\mathbf{L}}$). Since **D** (or $\tilde{\mathbf{D}}$) has shape $B^2 \times B^2$, the complexity is $O(MB^4) = O(N^5)$. Even it is only called once for one molecular structure, the cost is still intractable even for medium-sized molecules. Moreover, the approximation $\frac{\delta}{\delta\rho}T_\mathrm{S}[\rho_{\mathrm{p}^{(k)}}] \approx \frac{\delta}{\delta\rho}T_\mathrm{S}[\rho_{\mathrm{C}^{(k)}}]$ does not guarantee the optimality of density optimization using the learned KEDF model on the same molecular structure as explained later.

[00175] Hence another approximation is considered, to use a more direct way to calculate the gradient. The approximation is that Eq. (27) also holds for the fitted density $\rho_{\mathbf{p}^{(k)}}$:

$$\frac{\delta T_\mathrm{S}[\rho_{\mathbf{p}^{(k)}}]}{\delta\rho} + V_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}} = \mu'^{(k)}, \tag{59}$$

where $V_{\mathrm{eff}\{\mathbf{p}^{(k)}\}^{k'<k}}$ is the $\mathrm{V}_{eff}^{(k)}$ constructed from fitted density coefficients in previous SCF iterations, instead of orbital coefficient solutions in previous SCF iterations that the V_{eff} in Eq. (27) uses. The chemical potential μ′(k) may be different, but due to the above argument for Eq. (58), it is not used. The approximation holds if density fitting error can be omitted, *e.g.* when using a large basis set. Following the procedure above, the corresponding kinetic-energy gradient after projection is given by:

$$\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\dagger}{\mathbf{w}^\top\mathbf{w}}\right)\nabla_\mathbf{p}T_\mathrm{S}(\mathbf{p}^{(k)}) = -\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\dagger}{\mathbf{w}^\top\mathbf{w}}\right)\mathbf{v}_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}}. \tag{60}$$

$$\text{where} \quad \left(\mathbf{v}_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}}\right)_\mu := \left\langle\omega_\mu\middle|V_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}}\right\rangle = \int V_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}}(\mathbf{r})\omega_\mu(\mathbf{r})\,\mathrm{d}\mathbf{r}. \tag{61}$$

correspondingly. Calculating $V_{\mathrm{eff}\{\mathbf{p}^{(k)}\}^{k'<k}}$ requires a direct approach. This can be done following the relation between the known functions $\mathrm{V}_{eff}\left[\rho_{\left[\Phi^{(k')}\right]}\right]$ and the constructed

$V_{eff}^{(k)}$ in the SCF iteration. In DIIS, this relation can be drawn from the construction in Eq. (48) by noting the definitions Eq. (31) and Eq. (34) of the matrices, which is a weighted average: $V_{eff}^{(k)} = \sum_{k'=0}^{k-1} \pi_{k'}^{(k)} V_{eff\left[\rho_{\left[c^{(k')}\right]}\right]}$. Following this pattern, the required effective potential in Eq. (59) is constructed as: $V_{eff\left\{\rho^{(k')}\right\}_{k'<k}} = \sum_{k'=0}^{k-1} \pi_{k'}^{(k)} V_{eff\left[\rho_{\left[p^{(k')}\right]}\right]}$. The weights $\left\{\pi_{k'}^{(k)}\right\}_{k'=0}^{k-1}$ are taken as the same as those computed in the SCF iteration. Its vector form under the density basis is given by:

$$\mathbf{v}_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}} = \sum_{k'=0}^{k-1} \pi_{k'}^{(k)} \mathbf{v}_{\mathrm{eff}\mathbf{p}^{(k')}}, \quad \text{where } \left(\mathbf{v}_{\mathrm{eff}\mathbf{p}}\right)_{\mu} := \langle \omega_{\mu} | V_{\mathrm{eff}[\rho_{\mathbf{p}}]} \rangle = \int V_{\mathrm{eff}[\rho_{\mathbf{p}}]}(\mathbf{r}) \omega_{\mu}(\mathbf{r}) \, \mathrm{d}\mathbf{r}. \quad (62)$$

[00176] Calculation of each $\mathbf{v}_{\mathrm{eff}\mathbf{p}^{(k)}}$ can be carried out directly following Eq. (21) that gives $V_{\mathrm{eff}[\rho_{\mathbf{p}}]}$ explicitly. In the subject implementation, each $\mathbf{v}_{\mathrm{eff}\mathbf{p}^{(k)}}$ is conveniently calculated using automatic differentiation implementation mentioned in Supplementary Sec. A.3.2, since the inventors noticed the fact that:

$$\mathbf{v}_{\mathrm{eff}\mathbf{p}} = \nabla_{\mathbf{p}} E_{\mathrm{eff}}(\mathbf{p}), \quad (63)$$

where $E_{\mathrm{eff}}(\mathbf{p}) := E_{\mathrm{eff}}[\rho_{\mathbf{p}}]$ is defined in Eq. (49) and its gradient $\nabla_{\mathbf{p}} E_{\mathrm{eff}}(\mathbf{p})$ is given by Eq. (54). This fact is again due to the relation between gradient and variation revealed in Eq. (55) and noting that $V_{\mathrm{eff}[\rho]}$ is defined as the variation $\frac{\delta E_{eff}[\rho]}{\delta \rho}$ of the effective energy functional in Eq. (21). As analyzed at the end of Supplementary Sec. A.3.2, evaluating the gradient has the same complexity as evaluating the energy $E_{\mathrm{eff}}(\mathbf{p})$, which is $O(M^2) + O(MN_{\mathrm{grid}}) = O(N^2)$, which is much lower than the $O(N^5)$ complexity above.

[00177] To sum up, $\{\mathbf{v}_{\mathrm{eff}\mathbf{p}^{(k)}}\}_k$ are first calculated using automatic differentiation following Eq. (63), which are used to construct $\mathbf{v}_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}}$ following Eq. (62), then the gradient $\nabla_{\mathbf{p}} T_S(\mathbf{p}^{(k)})$ is given by Eq. (60) up to a projection. Since the loss function Eq.

(3) for gradient supervision explicitly projects the gradient error, the gradient label itself does not have to be projected before evaluating the loss (*i.e.*, the loss is the same whether the gradient label is projected; since projection is idempotent). The gradient label $\nabla_{\mathbf{p}} T_{\mathrm{S}}^{(k)}$ is taken directly as the density-constructed DIIS effective potential vector:

$$\nabla_{\mathbf{p}} T_{\mathrm{S}}^{(k)} := -\mathbf{v}_{\mathrm{eff}\{\mathbf{p}^{(k')}\}_{k'<k}}.$$

[00178] For the residual version of KEDF $T_{\mathrm{S,res}}$ and the version E_{TXC} that also includes XC energy as detailed in Supplementary Sec. B.4, the labels are produced accordingly:

$$\nabla_{\mathbf{p}} T_{\mathrm{S,res}}^{(k)} := \nabla_{\mathbf{p}} T_{\mathrm{S}}^{(k)} - \nabla_{\mathbf{p}} T_{\mathrm{APBE}}(\mathbf{p}^{(k)}), \quad \nabla_{\mathbf{p}} E_{\mathrm{TXC}}^{(k)} := \nabla_{\mathbf{p}} T_{\mathrm{S}}^{(k)} + \nabla_{\mathbf{p}} E_{\mathrm{XC}}(\mathbf{p}^{(k)}),$$

where $\nabla_{\mathbf{p}} T_{\mathrm{APBE}}(\mathbf{p}^{(k)})$ and $\nabla_{\mathbf{p}} E_{\mathrm{XC}}(\mathbf{p}^{(k)})$ are also calculated using automatic differentiation.

[00179] Apart from the convenient and efficient calculation using automatic differentiation, this choice of gradient label could also train a KEDF model that leads to the correct optimal density, since the labeling approach is compatible with the density optimization procedure of M-OFDFT shown in Eq. (1) and Eq. (54). More specifically, upon the convergence of SCF iteration (see quantities marked with "$\star$"), the DIIS effective potential $V_{\mathrm{eff}}^{\star}$ is converged to the single-step, explicit-form effective potential $V_{\mathrm{eff}[\rho^{\star}]}$ (Eq. (21) and Eq. (32)) by design. Correspondingly, the density-constructed DIIS effective potential vector $\mathbf{v}_{\mathrm{eff}\{\mathbf{p}^{(k)}\}^{k<\star}}$ (Eq. (61)) at convergence coincides with $\mathbf{v}_{\mathrm{eff}\mathbf{p}^{\star}}$ (Eq. (62)), which is $\nabla_{\mathbf{p}} E_{\mathrm{eff}}(\mathbf{p}^{\star})$ by Eq. (63). This gives a gradient label to the KEDF model through Eq. (60), which enforces the model to satisfy:

$$\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^{\top}}{\mathbf{w}^{\top}\mathbf{w}}\right)\left(\nabla_{\mathbf{p}} T_{\mathrm{S},\theta}(\mathbf{p}^{\star}) + \nabla_{\mathbf{p}} E_{\mathrm{eff}}(\mathbf{p}^{\star})\right) \overset{\text{Eq. (57)}}{=} \left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^{\top}}{\mathbf{w}^{\top}\mathbf{w}}\right)\nabla_{\mathbf{p}} E_{\theta}(\mathbf{p}^{\star}) = 0.$$

which in turn enforces the density optimization process Eq. (1) to converge to $\mathbf{p}^{\star}$, the true ground-state density coefficient. The optimality of density optimization using the learned KEDF model can then be expected.

[00180] A.5 Scaling Property under Atomic Basis

[00181] The KEDF has an exact scaling property which describes its change after uniformly scaling (stretching or squeezing) a density. Under a scaling (squeezing) rate λ, the uniformly scaled density is given by the rule of change of variables: $\hat{\lambda}\rho(\mathbf{r}) := \lambda^3\rho(\lambda\mathbf{r})$. The scaling property is stated as the following:

$$\mathrm{T_S}\left[\hat{\lambda}\rho\right] := \lambda^2\mathrm{T_S}[\rho] \tag{64}$$

[00182] Now it is considered to leverage this property for the KEDF model $T_{\mathrm{S}}(\mathbf{p},\mathrm{M})$ under atomic basis. As input density is represented under an atomic basis $\{\omega_\mu\}_{\mu=1}^{M}$ using coefficient $\mathbf{p}$ as $\rho_{\mathbf{p}}$ given by Eq. (29), first the scaled density $\hat{\lambda}\rho_{\mathbf{p}}$ is represented under the same basis with coefficient $\hat{\lambda}(\mathbf{p})$:

$$\hat{\lambda}\rho_{\mathbf{p}} := \sum_{\mu} \mathbf{p}_\mu \hat{\lambda}\omega_\mu(\mathbf{r}) = \sum_{\mu} \hat{\lambda}(\mathbf{p})_\mu \omega_\mu(\mathbf{r}). \tag{65}$$

Solving for $\hat{\lambda}(\mathbf{p})$ using least square gives $\hat{\lambda}(\mathbf{p}) = \mathbf{W}^{-1}\mathbf{M}^{\lambda}\mathbf{p}$, where $\mathbf{M}_{\mu\upsilon}^{(\lambda)} := \langle\omega_\mu|\hat{\lambda}\omega_\upsilon\rangle$ here, and $\mathbf{W}_{\mu\nu} := \langle\omega_\mu|\omega_\nu\rangle$ as the same as above. The scaling property Eq. (64) is then transformed as:

$$T_{\mathrm{S}}(\mathbf{W}^{-1}\mathbf{M}^{(\lambda)}\mathrm{p}.\mathcal{M}) = \lambda^2 T_{\mathrm{S}}(\mathrm{p}.\mathcal{M}). \tag{66}$$

[00183] However, to preserve Eq. (64) exactly, the expansion Eq. (65) must hold exactly. But this is not the case: the finite basis set $\{\omega_\mu\}_{\mu=1}^{M}$ is incomplete, and the scaled basis functions $\{\hat{\lambda}\omega_\mu\}_{\mu=1}^{M}$ $\{\hat{\lambda}\omega_\mu\}_{\mu=1}^{M}$ are not linearly dependent on the original ones.

Hence, the transformed scaling property Eq. (66) under the atomic basis is *not exact*, thus may not provide much benefit to the KEDF model $T_{S,\theta}(\mathbf{p},\mathrm{M})$.

[00184] There seems to be a possibility when using the even-tempered atomic basis, which comes in the form $\omega_{\mu=(a,\tau,\xi)}(\mathbf{r}) = w_{a,\tau,\xi}(\mathbf{r}-\mathbf{x}_{(a)})$, where $w_{a,\tau,\xi}(\mathbf{r} = (x,y,z)) := x^{\xi_1}y^{\xi_2}z^{\xi_3}\exp(-\alpha_{a,|\xi|}\beta^{\tau}\|\mathbf{r}\|_2)$, with tempering ratio $\beta > 1$, monomial-exponent parameter $\xi = (\xi_1,\xi_2,\xi_3)$, and exponent parameter $\alpha_{a,|\xi|}$ shared across ξ values with the same $|\xi| := \xi_1 + \xi_2 + \xi_3$. The index τ runs from 0 to $\mathrm{T}_{a,|\xi|}$. Therefore, under the scaling with $\beta^{-\frac{1}{2}}$:

$$\widehat{\beta^{-\frac{1}{2}}}\omega_{\mu=(a,\tau,\xi)}(\mathbf{r}) = \widehat{\beta^{-\frac{1}{2}}}w_{a,\tau,\xi}(\mathbf{r}-\mathbf{x}^{(a)}) = \beta^{-\frac{3}{2}}w_{a,\tau,\xi}(\beta^{-\frac{1}{2}}\mathbf{r} - \mathbf{x}^{(a)}) = (\beta^{-\frac{1}{2}})^{3+|\xi|}w_{a,\tau-1,\xi}(\mathbf{r}-\mathbf{x}^{(a)}/\beta^{-\frac{1}{2}}),$$

which is in the same even-tempered basis set but on a scaled molecular structure

$$\mathcal{M}/\beta^{-\frac{1}{2}} := \{\mathbf{Z}, \mathbf{X}/\beta^{-\frac{1}{2}}\},$$

as long as $\tau > 0$. For $\tau = 0$, it corresponds to the most flat basis function, and its coefficient $\mathbf{p}_{\mu=(a,\tau=0,\xi)}$ is close to zero hence can be omitted for density representation. Therefore, under the scaling with$\beta^{-\frac{1}{2}}$, the scaled density can be expressed as:

$$\begin{aligned}
\widehat{\beta^{-\frac{1}{2}}}\rho_{\mathbf{p},\mathcal{M}}(\mathbf{r}) &= \sum_{a,\xi}\sum_{\tau=0}^{\mathcal{T}_{a,|\xi|}}\mathbf{p}_{a,\tau,\xi}\widehat{\beta^{-\frac{1}{2}}}\omega_{a,\tau,\xi}(\mathbf{r}) \\
&= \sum_{a,\xi}\sum_{\tau=1}^{\mathcal{T}_{a,|\xi|}}\mathbf{p}_{a,\tau,\xi}(\beta^{-\frac{1}{2}})^{3+|\xi|}w_{a,\tau-1,\xi}(\mathbf{r}-\mathbf{x}^{(a)}/\beta^{-\frac{1}{2}}) + \sum_{a,\xi}\mathbf{p}_{a,0,\xi}\widehat{\beta^{-\frac{1}{2}}}\omega_{a,0,\xi}(\mathbf{r}) \\
&\approx \sum_{a,\xi}\sum_{\tau=1}^{\mathcal{T}_{a,|\xi|}}\mathbf{p}_{a,\tau,\xi}(\beta^{-\frac{1}{2}})^{3+|\xi|}w_{a,\tau-1,\xi}(\mathbf{r}-\mathbf{x}^{(a)}/\beta^{-\frac{1}{2}}) \\
&=: \sum_{a,\xi}\sum_{\tau=0}^{\mathcal{T}_{a,|\xi|}}\mathbf{p}^{(\beta^{-\frac{1}{2}})}_{a,\tau,\xi}w_{a,\tau,\xi}(\mathbf{r}-\mathbf{x}^{(a)}/\beta^{-\frac{1}{2}}) = \rho_{\mathbf{p}^{(\beta^{-\frac{1}{2}})},\mathcal{M}/\beta^{\frac{1}{2}}}(\mathbf{r}),
\end{aligned}$$

where the new coefficients are defined as:

$$\mathbf{p}^{(\beta^{-\frac{1}{2}})}_{a,\tau,\xi} := \begin{cases} (\beta^{-\frac{1}{2}})^{3-|\xi|}\mathbf{p}_{a,\tau-1,\xi}, & \tau < \mathcal{T}_{a,|\xi|}, \\ 0, & \tau = \mathcal{T}_{a,|\xi|}. \end{cases}$$

The scaling property Eq. (64) can then be written as:

$$T_S(\mathbf{p}^{(\beta^{-\frac{1}{2}})}, \mathcal{M}/\beta^{-\frac{1}{2}}) = (\beta^{-\frac{1}{2}})^2 T_S(\mathbf{p}, \mathcal{M}).$$

However, this holds only for one value of scaling depending on the basis set (also for $(\beta^{-1/2})^n$ for integer n as long as the contributions from the first n coefficients can be omitted). Moreover, this constraint connects the original input to a *scaled conformation* $M/\beta^{-1/2}$, which can be far from an equilibrium structure. The behavior of the $T_{S,\theta}$ model for these scaled conformations may be less relevant for real applications. Hence still, it does not seem definite to gain benefits from the scaling property.

[00185] B Details in the KEDF Model

[00186] The KEDF model takes as inputs atomic numbers **Z**, positions **X** and density coefficients **p** of all atoms, which are used to construct node features encoding information about the electron density around each atom. These features are further transformed and used to predict the kinetic functional energy, while the gradient of kinetic functional w.r.t density coefficients is obtained by auto-differentiation. Considering the fact that non-local calculation is indispensable for KEDF, a neural network transformer architecture with non-local attention, known as the Graphormer architecture is adopted, as shown in Fig. 6. The architecture has shown attractive performance using molecular structure to predict various properties, *e.g.*, ground-state energy and optical gap. Furthermore, distance features (as edge features) are also incorporated in the interaction calculation to accommodate for what the atomic features of the pair cannot cover. Based on the original architecture, a node embedding layer is proposed to generate initial atomic representations from various inputs, where the intra-atom density interaction is modeled by an MLP module (Fig. 6 at (a) and (b)). Furthermore, the self-attention module in each G3D layer accounts for the inter-atom interaction between density coefficients hosted on any pair of atoms. Note that the nonlocal architecture has an $O(N^2)$ complexity, which maintains better scalability than

KSDFT. In addition, the local frame is designed to eliminate geometric variability, and several efficient training techniques are proposed to deal with the vast gradient range. Explorations of different functional variants of the Graphormer model are also discussed.

[00187] B.1 Model Specification

B.1.1 Auxiliary Basis Concatenation

As mentioned in Methods 4, M-OFDFT adopts atomic basis as an efficient density representation in molecules, and the basis coefficients that can be attributed to each atom could naturally serve as nodewise density features for the molecule graph. Specifically, an even-tempered auxiliary basis with its β parameter taken as 2.5 is adopted as the density basis set. The number of basis in a molecule, M, is in the same order as the number of atoms A or electrons N. Each chemical element has its own set of basis functions and the size of each set T_{elem} varies from different element types. The detailed orbital composition of each element type is summarized in Supplementary Table 2. To make the coefficient vector homogeneous over all the atoms, these basis functions are concatenated and broadcast to all atoms, making a T -dimensional density coefficient vector $\mathbf{p}_{a:}$ on any atom, $T = P_{elem}\, T_{elem}$. Each basis ω_μ is then specified by the position of the center atom and the basis function index. Hence (a,τ) is used to index the τ-th basis of atom a. In more detail, in the QM9 dataset, the 477-dimensional concatenated coefficient vector consists of 20, 109, 116, 116 and 116 basis functions which correspond to H, C, N, O and F elements, respectively. For example, given the coefficient of a hydrogen atom, they are placed at the first 20 dimensions and use zero-valued vectors to mask other positions.

Fig. 6 illustrates an architecture of the atomic basis-based KEDF model. (a) The KEDF model predicts the kinetic energy $T_{S,\theta}$ from the inputs of atomic numbers $\mathbf{Z}$, positions $\mathbf{X}$ and density coefficient features $\tilde{\mathbf{p}}$. A *Node Embedding* module generates atomic features $\mathbf{h}$ by combining different input features, which are updated through a series of *G3D* layers. A *GBF* layer and a Multilayer Perceptron (MLP) module produce distance features $\tilde{\boldsymbol{\varepsilon}}$ from positions $\mathbf{X}$, which are incorporated into each *G3D* layer to model atomic interactions. An MLP module converts the updated features into atomic energy, which is summed to obtain the kinetic energy. The *Atom Ref* module provides an analytical reference for kinetic energy. (b) The *Node Embedding* module integrates information from three sources: atomic features obtained by an *Atom Embedding* layer that assigns a learnable weight vector to each element, density features processed by a *Shrink Gate* layer along with an MLP and contracted GBF embeddings that encode chemical environment. (c) The *G3D* layer is based on the standard Transformer encoder layer, and uses the distance feature as an auxiliary attention bias. (d) The density feature preprocessing module transforms the original density coefficients $\mathbf{p}$ into SE(3)-invariant and easy-controllable density features $\tilde{\mathbf{p}}$, using three techniques: *local frame*, *natural parameterization* and *dimension-wise rescaling*. A graphic illustration is shown in Fig. 7. The zero-valued mask is also employed to mask the predicted gradient during density optimization, avoiding introducing irrelevant gradient information from masked positions.

[00188] Fig. 7 illustrates concatenation of auxiliary basis for different atom types. In Fig. 7, τ is the index of density coefficient dimension and $p_{a,\tau}$ denotes the coefficient vector for element a. Note that this is a schematic illustration, so the coefficient dimensions may not correspond to the actual dimensions precisely.

Table 2: Orbital composition of basis set associated with each element.

Element	Orbitals
H	6s3p1d
C	11s8p7d3f2g
N	11s8p7d4f2g
O	11s8p7d4f2g
F	11s8p7d4f2g

[00189] B.1.2 Node Embedding Layer

[00190] As shown in Fig. 6 at (b), a node embedding layer was designed to integrate various input features into initial hidden node representations **h**, which are then passed to a Graphormer model to estimate the kinetic energy functional. Specifically, for atom a, an atom embedding layer generates atomic feature representation by assigning a learnable weight vector to each atom type $Z^{(a)}$. Note that the density coefficient vector $\mathbf{p}_{a:} \in \mathrm{R}^{\mathrm{T}}$ is a SO(3)-equivariant geometric tensor of different orders, which cannot be directly handled by Graphormer. To address this issue, a density feature preprocessing module was devised consisting of several submodules (Fig. 6 at (d)), which can not only transform the original coefficient vector $\mathbf{p}_{a:}$ into a SO(3)-invariant feature representation $\tilde{\mathbf{p}}$, but also effectively reduce the geometric variability of the atom-centered electron density. The technical details of this module are mentioned in Methods 4.2 and further explained in Supplementary Sec. B.2-B.3. It is also observed that the processed $\tilde{\mathbf{p}}$ exhibits a large numerical range, which is unfriendly for stable optimization of common neural network architectures. Thus, a shrink function $\lambda_{\mathrm{co}} \cdot \tanh(\lambda_{\mathrm{mul}} \cdot \tilde{\mathbf{p}})$ is applied to map each density coefficient into a bounded space, where λ_{co} and λ_{mul} are learnable parameters. Due to the invariant property of $\tilde{\mathbf{p}}$ w.r.t arbitrary rotations, applying any nonlinear operations (*e.g.*, shrink function) on this feature will not violate the SO(3) symmetry. After that, a Multilayer

Perceptron (MLP) is used to further project the coefficient vector into hidden density features. Following Graphormer, the relative distance between two atoms with a set of edge-type-aware Gaussian basis functions (GBF) is embedded, where the mean and variance values of each basis are learnable parameters. To capture the chemical environment around atom a, the contracted GBF embedding $\sum_{b=1}^{A} \boldsymbol{\varepsilon_{ab}}$ is incorporated as an auxiliary node feature. These three types of features are summed to form the initial node representations $\mathbf{h}_a \in \mathbb{R}^D$, where D is the hidden dimension of the model.

[00191] Figure 7 shows concatenation of auxiliary basis for different atom types. τ is the index of density coefficient dimension and $\mathbf{p}_{a,\tau}$ denotes the coefficient vector for element a. Note that this is a schematic illustration, so the coefficient dimensions may not correspond to the actual dimensions precisely.

[00192] B.1.3 Backbone Architecture

[00193] The KEDF model is based on the Graphormer model, which is a Transformer-based graph neural network, and can efficiently perform global computation over the molecular graph while preserving the powerful expressiveness of the Transformer architecture. The model mainly consists of an embedding layer and several stacked Graphormer-3D (G3D) layers, where each G3D layer is composed of an attention layer and a feed-forward layer (Fig. 6 at (c)). The attention layer takes the hidden node representation $\mathbf{h}$ as the input tokens and uses the relative distance feature $\tilde{\boldsymbol{\varepsilon}}$ as a learnable attention bias for the attention mechanism. Specifically, given the node feature $\mathbf{h} = \{\mathbf{h}_1, \cdots, \mathbf{h}_A\}$ and pairwise attention bias $\tilde{\boldsymbol{\varepsilon}} = \{\tilde{\boldsymbol{\varepsilon}}_{\boldsymbol{ab}}\}_{a,b=1}^{A}$, where a and b denote the atom index, the G3D layer computes the attention score A_{ab} for one attention head as:

$$A_{ab} = \frac{(\mathbf{h}_a \mathbf{U}^{(Q)})(\mathbf{h}_b \mathbf{U}^{(K)})^{\top}}{\sqrt{d}} - \tilde{\boldsymbol{\varepsilon}}_{ab},$$

where $\mathbf{U}^{(Q)}$ and $\mathbf{U}^{(K)}$ are the "query" and "key" linear projections for the node features and d is the dimension of the query and key vectors. To account for interactions between atom-centered electron density features, the GBF embedding $\boldsymbol{\varepsilon}$ is projected into pairwise attention bias $\tilde{\boldsymbol{\varepsilon}}$.

[00194] B.2 Local Frame Details

[00195] As mentioned in Methods 4.2, by expanding the electron density on a set of atomic basis (*i.e.*, contracted Gaussians with spherical harmonics), the expansion coefficient $\mathbf{p}$ mathematically acts as geometric tensors of various angular components covariant to rotations and translations. A local frame simultaneously rotated with the structure is adopted to decouple the density feature from the change of the coordinate system and unnecessary geometric variability. To construct the local frame at an atom at $\mathbf{x}_a$, $\hat{\mathbf{x}}$ pointing to its nearest atom is chosen. The $\hat{\mathbf{z}}$ axis lies in the line of the cross-product of $\hat{\mathbf{x}}$ with the direction to the second-nearest not-on-$\hat{\mathbf{x}}$ atom and the $\hat{\mathbf{y}}$ axis is then given by $\hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}}$. Note that the hydrogen atoms in the neighborhood of the center atom are excluded, making the obtained local frame depend more on heavy atoms, whose positions are more stable than those of hydrogen atoms and reflect more reliable local structures. Denote the local frame associated with atom a as

$$\mathcal{R}_a = (\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}),$$

the density coefficient vector and the gradient vector under the local frame are calculated by,

$$\mathbf{p}_a'^{l} = D^l(\mathcal{R}_a)\mathbf{p}_a^l.$$

and

$$\nabla_{\mathbf{p}_a'^{l}} T_{\mathrm{S}} = D^l(\mathcal{R}_a)\nabla_{\mathbf{p}_a^l} T_{\mathrm{S}}.$$

[00196] As shown in Fig. 10, by projecting the atom-centered density onto the most significant chemical bond, the local frame always attains a noticeable scale (standard deviation) reduction across various atom types, demonstrating its capability to eliminate unnecessary geometric variability caused by flexible orientations of the chemical bond. More importantly, this approach can also bring a significant scale reduction for gradient labels across various atom types (as shown in Fig. 11), especially for H atoms, where all coefficient dimensions exhibit a > 50% gradient scale reduction. This is crucial to alleviate the large gradient range and make the subsequent dimension-wise rescaling module easier (See more discussions in Supplementary Sec. B.3.3). These two advantages of the local frame are beneficial for the efficient optimization of the KEDF model. The empirical results in Supplementary Sec. F.4.3 suggest that the KEDF model can achieve a 2-fold lower training and test error by using this technique.

[00197] B.3 Techniques for Efficient Training

[00198] As mentioned in Methods 4.3, the range of KEDF data is vast, and conventional data normalization technique is not applicable in task at hand since minimizing the gradient scale is conflicting with minimizing the coefficient scale. For the same reason the logarithm of density input feature cannot be adopted, since the gradient would be correspondingly scaled up. Having both large coefficients and gradient implies a function with a large Lipschitz coefficient, which leads to a great challenge for the optimization of common neural networks. It neither works to use a separate model to learn the gradient directly, as the energy model would easily overfit, and in density optimization the gradient model unnecessarily decreases the energy from the energy model and even appears non-conservative. Another effective strategy to decrease the gradient scale is to downscale the energy scale. However, the KEDF model equipped with this approach suffers from a loss of accuracy in energy prediction since

increasing the scale or unit of energy lead to a loss of resolution. According to these observations, a series of elaborated techniques is introduced to enable effective training.

[00199] B.3.1 Natural Reparameterization

[00200] If directly measured in Euclidean metric, this may induce unbalanced sensitivity to the output. It would be better to use a parameterization that reflects the metric of physical meaning. The coefficient represents the density, and a natural metric for density is the L2 metric: $d(\rho,\rho')^2 = \int|\rho(\mathbf{r}) - \rho'(\mathbf{r})|^2 \mathrm{d}\mathbf{r}$. Plugging in the basis expansion expression, this then induces a metric on density coefficient: $d(\mathbf{p},\mathbf{p}')^2 = d\left(\sum_\mu \mathbf{p}_\mu\omega_\mu, \sum_\mu \mathbf{p}'_\mu,\omega_\mu\right)^2 = \sum_{\mu,\mu'}(\mathbf{p}_{\mu'} - \mathbf{p}'_{\mu'})(\mathbf{p}_\mu - \mathbf{p}'_\mu)\int \omega_{\mu'}(\mathbf{r}) - \omega_\mu(\mathbf{r})\mathrm{d}\mathbf{r} = (\mathbf{p}-\mathbf{p}')^{\mathrm{T}}\mathbf{W}(\mathbf{p}-\mathbf{p}')$, where $\mathbf{W}_{\mu\nu} := \int \omega_\mu(\mathbf{r})\omega_\nu(\mathbf{r})\mathrm{d}\mathbf{r}$ is the overlap matrix of the density basis. Using this natural metric makes the scale of change in $\mathbf{p}$ reflect the scale of change in the represented density ρ, and the resulting density loss and gradient loss make better physical meaning and have balanced sensitivity over dimensions of $\mathbf{p}$.

[00201] Noting the metric $\mathbf{W}$ is independent of $\mathbf{p}$, an alternative way to incorporate this metric is by reparameterizing the density: let $\tilde{\mathbf{p}} := \mathbf{M}^{\mathrm{T}}\mathbf{p}$, where $\mathbf{M}$ satisfies $\mathbf{M}\mathbf{M}^{\mathrm{T}} = \mathbf{W}$. In this way, the Euclidean metric in the new parameter, $(\tilde{\mathbf{p}} - \tilde{\mathbf{p}}')^{\mathrm{T}}(\tilde{\mathbf{p}} - \tilde{\mathbf{p}}'') = (\mathbf{p} - \mathbf{p}')\mathbf{M}\mathbf{M}^{\mathrm{T}}(\mathbf{p} - \mathbf{p}') = (\mathbf{p} - \mathbf{p}')\mathbf{W}(\mathbf{p} - \mathbf{p}')$ recovers the above natural metric. This reparameterization makes a cheaper implementation of density/gradient loss and natural gradient descent.

[00202] Choosing the matrix

[00203] $\sqrt{\mathbf{M}}$ still faces a degree of freedom of an orthogonal transformation. $\mathbf{M} = \sqrt{(\mathbf{Q}\boldsymbol{\Lambda})}$ is chosen, where the diagonal $\boldsymbol{\Lambda}$ and orthogonal $\mathbf{Q}$ gives the eigenvalue decomposition of $\mathbf{W} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{T}}$(note $\mathbf{W}$ is symmetric positive definite so such

decomposition exists), and $\sqrt{\mathbf{\Lambda}}$ amounts to elementwise square-root. It was found to perform better than using the Cholesky decomposition. The use of natural reparameterization can obviously stabilize the optimization of the KEDF model and bring lower training and validation loss, underscoring the necessity of balancing the density coefficient with a physical metric, see more details in Supplementary Sec. F.4.3.

[00204] B.3.2 Atomic Reference Model

[00205] The standard gradient label has a large scale, which is challenging for the model's gradient to capture. It is desirable to subtract a gradient reference to center the gradient label. The problem is the dimension and structure of the input **p** thus the gradient **g** depend on the chemical constitution (number of atoms of each element) of the molecule and vary across molecules, since different chemical elements have non-overlapping auxiliary basis. The gradient mean is hence taken for each element: average gradient dimensions corresponding to auxiliary basis of the element. The resulting elementwise gradient reference represents the average response in T_S to the change of density coefficient for the auxiliary basis of an element. In this way, a gradient reference $\bar{\mathbf{g}}_M$ for a molecule in structure M can be constructed by tiling the elementwise gradient reference according to the constitution.

[00206] To match the new gradient, each energy label $\mathrm{T}_S^{(k)}M^{(n)}$ for molecule in structure $M^{(n)}$ should also be subtracted with $\bar{\mathbf{g}}_{M^{(n)}}^{\mathsf{T}}\mathbf{p}_{M^{(n)}}^{(k)} + \bar{\mathrm{T}}_{M^{(n)}}$, where $\bar{\mathrm{T}}_{M^{(n)}}$ is a bias to center the transformed energy labels of molecule in structure M for easier learning. This bias $\bar{\mathrm{T}}_{M^{(n)}}$ faces the same problem of constitution dependency, so similarly an energy bias is fit for each element, and construct $\bar{\mathrm{T}}_{M^{(n)}}$ by summing the biases of all atoms in $M^{(n)}$. A global bias was added once to each molecule.

[00207] As for normalizing the standard deviation of the transformed energy label, further improvement was not found. This may be due to the trade-off between easy learning and model resolution: an error in the normalized (small) scale will be enlarged in the original scale. In the KEDF model, after being processed by the local frame and natural reparametrization techniques, the density coefficients are fed to the atomic reference model to provide an analytic reference for the final energy prediction (Fig. 6 at (a)).

[00208] B.3.3 Dimension-wise Rescaling

[00209] As mentioned in Methods 4.3, there is a trade-off in scaling the input coefficient label: a moderate (small) coefficient scale would enlarge the gradient scale. Under this consideration, in Eq. (4), the target gradient scale $\bar{s}_g$ is set to 0.05 and the maximum coefficient scale $\bar{s}_c$ is set to 50.

[00210] To illustrate the importance of dimension-wise rescaling, the label scale of kinetic functional gradients, rescaled kinetic functional gradients and density coefficients of the QM9 dataset is illustrated in Fig. 12. The energy functional is chosen as the residual energy between non-interacting kinetic energy and empirical kinetic functional APBE. In the context of deep learning, the Lipschitz coefficient (*i.e.*, the maximum absolute gradient of the neural network model) is usually used to indicate the capability of a model fitting gradient label. Following this convention, the L^{∞} metric is taken to measure the label scale. It should also be noted that the gradient labels here have been centralized by subtracting the gradient mean of each dimension with the atomic reference model. Thus the dimension-wise rescaling operation only needs to reduce the variation range around the zero gradient mean. The combination of these two techniques can significantly improve the robustness of optimization. As plotted in

Fig. 12 at (a), many gradient dimensions have an extremely large scale. This leads to great difficulty in the practical optimization of deep learning models. After dimension-wise rescaling, most dimensions can be rescaled to have the desired gradient scale, and the gradient scale of other dimensions is still in an acceptable range (Fig. 12 at (b)). As shown in Fig. 12 at (c), rescaled density coefficients also exhibit a vast range (up to 250) that is difficult to process for neural networks even with data normalization techniques, thus a shrink function is introduced to compress the coefficients into a dense space, as mentioned in Supplementary Sec. B.1.2. In practice, the KEDF model without the dimension-wise rescaling technique is extremely hard to converge and the loss curve is particularly volatile. The reason is that the smoothness of common neural network architectures restricts the range of the output gradient of the KEDF model. Applying the dimension-wise rescaling technique can effectively mitigate this dilemma and enable efficient optimization.

[00211] In addition, the atomic reference model technique contributes considerably to the dimension-wise rescaling procedure. Fig. 13 shows that using the atomic reference model can significantly reduce the variance/scale of gradient labels, especially for dimensions corresponding to 's' atomic orbitals. Since 's' orbitals usually have large gradient mean values but small variance values, subtracting the gradient mean of these dimensions greatly simplifies the dimension-wise rescaling.

[00212] Density Feature Preprocessing Module

[00213] Finally, the local frame technique, natural reparameterization, and dimension-wise rescaling are combined into the density feature preprocessing module, as shown in (Fig. 6 at (d)). It transforms the density coefficients $\mathbf{p}$ to density features $\tilde{\mathbf{p}}$ as the direct input to the Graphormer model, which makes the model much easier to

learn. This module eliminates unnecessary geometric variability of density features and facilitates the efficient learning of vast gradient labels through each component. This preprocessing module is only required *once* for each molecular structure M in training or running density optimization. Therefore, although natural reparameterization requires a cubic computational cost, it does not dominate the scaling of the total cost compared to the iterated gradient calculation of the energy terms in Eq. (49) including the Transformer-based KEDF model, which sets up the $O(N^2)$ complexity of M-OFDFT.

[00214] B.4 Functional Variants

[00215] In principle, besides the unknown KEDF, any energy density functionals in the decomposition formulation of ground-state energy (Eq. (14)) can be taken as the training objective of the proposed M-OFDFT framework. Here, the two versions mainly employed in the implementation of M-OFDFT are first described, and an exploration in other functional variants is described.

[00216] B.4.1 Residual KEDF

[00217] In the implementation of M-OFDFT, the first functional version introduced aims to reduce the difficulty of modeling KEDF directly by employing an existing KEDF as the base functional and learning the residue:

$$T_{\mathrm{S},\theta}(\mathbf{p},\mathrm{M}) := T_{\mathrm{S,res},\theta}(\mathbf{p}) + T_{\mathrm{base}}(\mathbf{p},\mathrm{M}). \tag{67}$$

Specifically, APBE KEDF was chosen as the reference/baseline KEDF since it best fits the training data on the QM9 dataset among four functionals mentioned in Supplementary Sec. D.2. Although the von Weizsacker KEDF provides a lower bound and could restrict the residue model to be positive, using it as a baseline introduces vast gradient labels for the residue that makes training difficult.

[00218] B.4.2 TXC Functional

[00219] Although the KEDF residual functional provides a simple learning objective with an easy-controlled gradient range, it has an inevitable computational bottleneck that the gradient of the APBE KEDF reference and PBE XC functionals need to be back-propagated w.r.t density coefficients through grids. As discussed in Supplementary Sec. A.3.2, the memory requirement of evaluating the gradient on grid points is $O(MN_{\text{grid}})$. Considering the large prefactor of N_{grid} (commonly $\sim 10^3 N$), the computational cost of KEDF residue in gradient descent becomes unaffordable for large-scale systems. Such cost is witnessed when running KEDF residual functional on the QMugs dataset. To completely get rid of calculation on grids, the second version $E_{\text{TXC},\theta}(\mathbf{p},\text{M})$ is introduced to learn the sum of KEDF and the XC energy (named TXC functional E_{TXC}) altogether.

[00220] B.4.3 Learning Other Functionals

[00221] Besides the two versions above, other direct targets for the machine learning model to learn were explored, including directly learning the KEDF T_S (*i.e.*, without a reference KEDF), the universal functional $U[\rho]$ (Eq. (11)), and the total energy functional $E[\rho]$ (Eq. (14)). Both directly training of the KEDF model and training of the universal functional are hard to optimize due to their large gradient range, which is even larger than that for learning the residual KEDF model and the TXC model and cannot be effectively handled even using the proposed techniques. Interestingly, although the learned functional would be made intrinsically system-dependent, learning the total energy functional $E_\theta[\rho]$, *i.e.*, adding the external energy E_{ext} to the universal functional as the target, significantly reduces the gradient range and enables stable optimization, especially on protein systems considered. Specifically, on 1,000

Chignolin conformations, learning the $E_\theta[\rho]$ model gives a better energy result through density optimization (MAE 0.071kcal/mol) than learning the $E_{\mathrm{TXC},\theta}$ model. The $E_{\mathrm{TXC},\theta}$ model still achieves a reasonable result (MAE 0.102kcal/mol), which is still much better than classical KEDFs.

[00222] C Density Optimization

[00223] C.1 Gradient-based Density Optimization

[00224] As mentioned in Methods 4.4, during the density optimization, the trained KEDF model $T_{S,\theta}(\mathbf{p},\mathrm{M})$ is employed to minimize the total electronic energy w.r.t density coefficient $\mathbf{p}$. This procedure is implemented with a gradient descent process instead of a fixed-point SCF iteration process, as the optimality Eq. (27) or Eq. (56) for the density coefficient is hard to be converted to a fixed-point iteration. Details in calculating the gradient of energy are discussed in Supplementary Sec. A.3.2. Fig. 14 shows a typical energy optimization curve of a randomly selected molecule in the QM9 dataset, along with the curve of gradient scale (measured by the L2-metric). Both curves for two initialization densities converge stably and closely approach the ground-truth total energy.

[00225] C.2 Stopping Criterion

[00226] In principle, the solution of M-OFDFT can be obtained at the stationary point (zero projected gradient). But the exact zero-gradient point may be missed due to discretization error. Therefore, a set of stop criteria is proposed to determine the practical stationary point, illustrated in Fig. 14. For in-scale molecules such as the QM9 and Ethanol datasets, *GlobEngUpd* always indicates a satisfactory stationary point and exhibits the best performance on a set of evaluation molecules. Hence, it is used as the

standard criterion for in-scale molecules. For larger-scale systems, however, this criterion is not suitable since the inevitable extrapolation error may make the model fall into unreliable density regions. Thus, more conservative criteria *1stLocGradNorm* or *1stLocEngUpd* were considered, and then the *GlobEngUpd* criterion would be adopted if the first two criteria do not exist.

[00227] C.3 Density Initialization

[00228] To implement the ProjMINAO initialization for the density coefficient, an additional output branch $\Delta\mathbf{p}_\theta(\mathbf{p},\mathrm{M})$ was constructed to predict the difference of the given density coefficient $\mathbf{p}$ from the ground-state density coefficient $\mathbf{p}^\star$ for the molecular structure M. The branch is built on top of the last *G3D* layer shown in Fig. 6 at (a) of the KEDF model. It processes the hidden representations of each atom through an MLP module to predict the corresponding coefficient difference. The branch is trained to project the density coefficient data along the SCF iteration of KSDFT onto the corresponding ground-state coefficient:

$$\sum_d \sum_k \left\| \Delta\mathbf{p}_\theta(\mathbf{p}^{(d,k)}, \mathcal{M}^{(d)}) - (\mathbf{p}^{(d,k)} - \mathbf{p}^{(d,\star)}) \right\|. \tag{68}$$

[00229] The acquisition of fitted MINAO or Huckel density coefficients can be achieved by applying density fit-¨ ting (Supplementary Sec. A.4.1) to the original MINAO or Huckel initialized orbitals. This procedure is¨ performed only *once* for density initialization, so the cubic computational cost of density fitting (and generating Huckel initialized orbitals) does not dominate over the quadratic scaling of density optimization¨ on regular workloads. To retain the quadratic scaling on very large molecules, an alternative initialization is also proposed to directly generate the density coefficients by superposition of isolated-atom densities, in the same spirit of MINAO which uses the superposition of isolated-atom orbitals. This amounts to concatenating

the fitted MINAO density coefficients of each atom as if in isolation, which only has a linear computational cost. In practice, this technique drastically reduces the time needed to construct the initial coefficients, further reducing the overall time consumption of M-OFDFT, and renders only a minor increase of error (approximately 0.02kcal/mol in the per-atom energy error on the interested protein systems).

[00230] D General Experimental Setup

[00231] D.1 Data Preparation

[00232] KSDFT Calculations

[00233] All KSDFT calculations were carried out using the PySCF software package, which is open-sourced and has a convenient Python interface that allows flexible customization, *e.g.*, for implementing the data generation process detailed in Supplementary Sec. A.4, and implementing M-OFDFT calculation described in Results 2.1 using automatic differentiation packages in Python. All KSDFT calculations are at the PBE/6-31G(2df,p) level. To accelerate the calculations, density fitting with the def2-universal-jfit basis set was enabled in the calculations for molecules with more than 30 atoms. Grid level was set to 2. Convergence tolerence was set to 1 meV. Slight modifications to the PySCF code were made to log the required information (*e.g.*, the molecular orbitals) for SCF intermediate steps. Direct Inversion in Iterative Subspace (DIIS) was enabled, following the defaults. Per-molecule statistics such as running time and number of atoms were collected alongside the calculation for benchmarking purposes.

[00234] Initialization

[00235] The MINAO initialization is employed for conducting KSDFT calculations, adhering to the default settings. Using the Huckel initialization to run KSDFT for data generation was considered, as it is based on the eigenvalue-problem solution of a simple Hamiltonian matrix, making it more akin to an in-distribution density for the ground-state density. However, the range of SCF results obtained from the Huckel initialization is more diverse and challenging to optimize compared to the MINAO initialization, resulting in a notable performance decline in density optimization.

[00236] Hardware Details

[00237] All KSDFT calculations and preprocessing (e.g., density fitting) procedures were carried out on a cluster of 700 capable CPU servers on Azure. Each server in the cluster has 256 GiB of memory and 32 Intel Xeon Platinum 8272CL cores with hyperthreading disabled.

[00238] Dataset Preprocessing

[00239] After the SCF runs, extra procedures were done to transform the SCF results into datasets for training and evaluation. The density coefficients were obtained using the density fitting procedure detailed in Supplementary Sec. A.4.1, with the auxiliary basis set being an even-tempered basis set generated on-the-fly with $\beta = 2.5$. After that, gradient and force labels were calculated using the obtained density coefficients following Supplementary Sec. A.4.3 and D.4. Energy labels were calculated following Supplementary Sec. A.4.2. For large orbital integrals in the procedure, proper symmetries in the atomic orbital indices were leveraged to save computation and reduce memory footprint. Local frame transformation and natural

reparameterization (detailed in Supplementary Sec. B.2 and Supplementary Sec. B.3.1) were applied to the coefficients to obtain the model inputs.

[00240] D.2 Classical Baselines

[00241] To benchmark the advantages of M-OFDFT, several classical kinetic energy density functionals were selected as baselines. The Thomas-Fermi (TF) KEDF is exact in the uniform electron gas limit. It takes the form $T_{\mathrm{TF}}[\rho] := \frac{3}{10}(3\pi^2)^{2/3}\int \rho(\mathbf{r})^{5/3}\mathrm{d}\mathbf{r}$. The von Weizsacker KEDF takes the form

$$T_{\mathrm{vW}}[\rho] := \frac{1}{8}\int \frac{\|\nabla\rho(\mathbf{r})\|}{\rho(\mathbf{r})}\,\mathrm{d}\mathbf{r},$$

which is exact for Bosons and single Fermion. When expanding KEDF up to the first-order derivatives of the density, the result is $T_{\mathrm{TF}}[\rho] + \frac{1}{9}T_{\mathrm{vW}}[\rho]$. A variant of this correction $T_{\mathrm{TF}}[\rho] + T_{\mathrm{vW}}[\rho]$ was also considered. The APBE reference KEDF is also considered. For a fair comparison, these functionals are implemented in the same OFDFT framework as M-OFDFT. Gradient descent with the same stopping criterion is used for density optimization. The kinetic energy value is evaluated on grids. For initialization, the MINAO method was used, which was found better than the Huckel method.

[00242] We also attempted to use other state-of-the-art OFDFT software as baselines, such as PROFESS, GPAW, ATLAS and DFTpy. However, most of these are tailored for periodic material systems and depend heavily on sophisticated pseudopotentials. Adapting these codes to molecular systems of interest is challenging since they only provide local pseudopotentials for a few main group elements, which do not cover the elements (HCNOF), and difficulties were encountered in re-generating reliable pseudopotentials for these elements. Although GPAW can handle molecule

systems, it was found that its OFDFT calculations on molecular systems are hard to converge even for small molecules. Therefore, only the classical KEDFs were implemented in the present framework as baselines.

[00243] D.3 M-PES Baselines

[00244] Results 2.4 demonstrate that M-OFDFT has the ability to extrapolate to large molecular systems, a valuable advantage for quantum chemistry methods. The significance of this advantage was measured by contrasting M-OFDFT with two machine-learning-based alternatives, M-PES and M-PES-Den, that estimate the ground-state energy from the molecular structure M. Both alternatives employ the same Graphormer backbone architecture as M-OFDFT, but they do not have the gradient prediction and density correction prediction branches, and M-PES also do not use the node embedding layer that incorporates density features (Fig. 6 at (b)). The model parameters are also adjusted to be comparable to those of M-OFDFT.

[00245] D.4 Hellmann-Feynman Force Calculation

[00246] To more comprehensively evaluate M-OFDFT, the Hellmann-Feynman (HF) force was also employed as a metric to evaluate the results of M-OFDFT. The force experienced by atoms in a molecule is itself an important quantity as it is directly required for geometry optimization and molecular dynamics simulation. The HF force is a simple way to estimate the force, and is exact in the limit of complete basis. Due to the simplicity, the HF force can also be seen as an evaluation of the optimized density.

[00247] Hellmann-Feynman Force

[00248] Formally, force is the negative gradient of the total energy of a molecule as a function $E_{\text{tot}}(\mathbf{X})$ of atom coordinates $\mathbf{X} = (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(A)})$, namely the molecular

conformation. The dependency on the molecular composition **Z** is omitted for brevity. The total energy $E_{\text{tot}}(\mathrm{X}) := E^*_{\mathrm{X}} + E_{\text{nuc}}(\mathrm{X})$ comprises both the energy E^*_{X} of the electrons in their ground state (including interaction with the nuclei), and also the energy due to inter-nucleus interaction, $E_{\text{nuc}}(\mathrm{X}) = \frac{1}{2}\sum_{\substack{a,b=1,\dots,A \\ a\neq b,}} \frac{Z^{(a)}Z^{(b)}}{\|\mathrm{x}^{(a)}-\mathrm{x}^{(b)}\|}$, which gives the inter-nucleus part of the force,

$$-\nabla_{\mathrm{x}^{(a)}} E_{\text{nuc}}(\mathbf{X}) = -Z^{(a)} \sum_{b:\neq a} Z^{(b)} \frac{\mathbf{x}^{(b)} - \mathbf{x}^{(a)}}{\|\mathbf{x}^{(b)} - \mathbf{x}^{(a)}\|^3}. \tag{69}$$

[00249] The electronic energy E^*_{X} is the minimum after a variational optimization process for solving the electronic ground state of the molecule in conformation **X**. In the most fundamental form, E^*_{X} is determined by the variational problem on N-electron wavefunctions as shown in Eq. (5). The Hamiltonian operator therein $\hat{H}_{\mathrm{X}} = \hat{T} + \hat{V}_{\text{ee}} + \hat{V}_{\text{ext,X}}$ depends on the conformation **X** through $V_{\text{ext,X}}$, which is given in Eq. (6). The ground-state wavefunction $\psi^*_{[\Phi]}$ and energy $E^*_{\mathrm{X}} = \langle\psi^*_{[\Phi]}|\,\hat{H}_{\mathrm{X}}|\psi^*_{[\Phi]}\rangle$ hence also depend on **X**. The gradient of E^*_{X} can then be reformed as: $\nabla_{\mathrm{X}} E^*_{\mathrm{X}} =$

$$\begin{aligned}
\nabla_{\mathbf{X}}\langle\psi_{\mathbf{X}}|H_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle &= \langle\nabla_{\mathbf{X}}\psi_{\mathbf{X}}|H_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle + \langle\psi_{\mathbf{X}}|\nabla_{\mathbf{X}}H_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle - \langle\psi_{\mathbf{X}}|H_{\mathbf{X}}\nabla_{\mathbf{X}}\psi_{\mathbf{X}}\rangle \\
&\overset{(*)}{=} E^*_{\mathbf{X}}\langle\nabla_{\mathbf{X}}\psi_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle + \langle\psi_{\mathbf{X}}|\nabla_{\mathbf{X}}H_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle + E^*_{\mathbf{X}}\langle\psi_{\mathbf{X}}|\nabla_{\mathbf{X}}\psi_{\mathbf{X}}\rangle \\
&= \langle\psi_{\mathbf{X}}|\nabla_{\mathbf{X}}H_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle + E^*_{\mathbf{X}}\nabla_{\mathbf{X}}\langle\psi_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle \\
&\overset{(\#)}{=} \langle\psi_{\mathbf{X}}|\nabla_{\mathbf{X}}H_{\mathbf{X}}|\psi_{\mathbf{X}}\rangle.
\end{aligned} \tag{70}$$

[00250] where the equality (*) is due to that $\psi^*_{[\Phi]}$ is an eigenstate of the Hermitian operator $\hat{H}_{\mathrm{X}}$ with real eigenvalue E^*_{X}, and the equality (#) is due to that the wavefunction is normalized $\langle\psi^*_{[\Phi]}|\psi^*_{[\Phi]}\rangle = 1$ for all **X**. Eq. (70) is the Hellmann-Feynman (HF) theorem. To continue the calculation, the gradient of the Hamiltonian operator in the expression can be derived as $\nabla_{\mathrm{X}^{(a)}}\hat{H}_{\mathrm{X}} = \nabla_{\mathrm{X}^{(a)}}\hat{V}_{\text{ext,X}}$, and by noting that $\hat{V}_{\text{ext,X}}$ is

multiplicative and one-body as shown in Eq. (6), $\nabla_{\mathrm{X}^{(a)}}\mathrm{V}_{\mathrm{ext,X}}(\mathrm{r}) = -Z^{(a)}\frac{\mathrm{r}-\mathrm{x}^{(a)}}{\|\mathrm{r}-\mathrm{x}^{(a)}\|^3}$, and subsequently, the electronic force can be calculated as:

$$-\nabla_{\mathbf{x}^{(a)}} E_{\mathbf{X}}^{*} = -\langle \psi_{\mathbf{X}}^{*} | \nabla_{\mathbf{x}^{(a)}} H_{\mathbf{X}} | \psi_{\mathbf{X}}^{*} \rangle = Z^{(a)} \int \rho_{\mathbf{X}}^{*}(\mathbf{r}) \frac{\mathbf{r} - \mathbf{x}^{(a)}}{|\mathbf{r} - \mathbf{x}^{(a)}|^3} d\mathbf{r} =: \mathbf{f}_{\mathbf{X}}^{(a)}. \tag{71}$$

where $\rho_{\mathrm{X}}^{*}(\mathrm{r}) := \rho_{\psi_{[\Phi]}^{*}}(\mathrm{r})$ defined in Eq. (7). This is the *Hellmann-Feynman (HF) force* $\mathbf{f}_{\mathbf{X}}$. This expression coincides with the electrostatic force under a classical view, indicating "there are no 'mysterious quantum-mechanical forces' acting in molecules". From the expression, evaluating the HF force only requires a good approximation to the ground-state electronic density $\rho_{\mathrm{X}}^{*}(\mathrm{r})$, which is available in various quantum chemistry methods, including$\rho_{\mathrm{C}_X^{*}}$ in KSDFT given by Eq. (32) and$\rho_{\mathrm{P}_X^{*}}$ in M-OFDFT given by Eq. (29). The total force on the nuclei is the sum with the inter-nucleus force shown in Eq. (69).

[00251] Analytical Force

[00252] However, when using the atomic basis, there emerges approximation error in the function representation, since the basis is incomplete. Consequently, the conditions of the HF theorem do not hold exactly. This makes the HF force in Eq. (71) only an approximation to the true electronic force $-\nabla_{\mathrm{X}^{(a)}}E_{\mathrm{X}}^{*}$, and other approximations are possible. For example, $-\nabla_{\mathrm{X}^{(a)}}E_{\mathrm{X}}^{*}$ can also be estimated by directly taking the analytical gradient of the electronic energy $E\mathrm{x}^{\star}$ expressed under the atomic basis with the optimal coefficients. This way of estimating the electronic force is hence called *analytical force* $\mathbf{f}_{\mathrm{ana}}$. For KSDFT, $E_{\mathrm{X}}^{*} = E_{\mathrm{X}}(\mathrm{C}_{\mathrm{X}}^{*})$, where $E\mathrm{x}(\mathbf{C})$ is given by Eqs. (38-42) (note that the basis functions η_α, η_β and matrices $\mathbf{T}$, $\tilde{\mathbf{D}}$ and $\mathbf{V}_{\mathrm{ext}}$ all depend on $\mathbf{X}$), and

$\mathbf{C}^*_\mathbf{X}$ is the optimal orbital coefficients. The corresponding analytical force on an atom a can be expanded as:

$$\mathbf{f}^{(a)}_{\text{ana}} := -\nabla_{\mathbf{X}^{(a)}} E_\mathbf{X}(\mathbf{C}^*_\mathbf{X}) = -(\nabla_{\mathbf{X}^{(a)}} E_\mathbf{X})(\mathbf{C}^*_\mathbf{X}) - \text{tr}\left((\nabla_{\mathbf{X}^{(a)}} \mathbf{C}^*_\mathbf{X})^\top \nabla_\mathbf{C} E_\mathbf{X}(\mathbf{C})\big|_{\mathbf{C}=\mathbf{C}^*_\mathbf{X}}\right). \quad (72)$$

where $(\nabla_{\mathbf{X}^{(a)}} E_\mathbf{X})$ takes the gradient with a fixed $\mathbf{C}$, and $\nabla_{\mathbf{X}^{(a)}}\mathbf{C}^*_\mathbf{X}$ is the Jacobian, $\left(\nabla_{\mathbf{X}^{(a)}}\mathbf{C}^*_\mathbf{X}\right)_{\alpha i,\varepsilon} := \partial(\mathbf{C}^*_\mathbf{X})_{\alpha i}/\partial \mathbf{X}^{(a)}_\varepsilon$ in which $\xi \in \{1,2,3\}$ indices the three spacial components, and matrix operations, including transpose, matrix multiplication and trace, act on indices α and i. When $\mathbf{C}^*_\mathbf{X}$ is indeed accurately optimized, self-consistency in Eq. (47) and orthonormality in Eq. (43) are satisfied. By also noting Eq. (46), the second term in Eq. (72) vanishes:

$$\text{tr}\left((\nabla_{\mathbf{X}^{(a)}} \mathbf{C}^*_\mathbf{X})^\top \nabla_\mathbf{C} E_\mathbf{X}(\mathbf{C})\big|_{\mathbf{C}=\mathbf{C}^*_\mathbf{X}}\right) \overset{\text{Eq. (46)}}{=} 2\,\text{tr}\left((\nabla_{\mathbf{X}^{(a)}} \mathbf{C}^*_\mathbf{X})^\top \mathbf{F}_{\mathbf{C}^*_\mathbf{X}} \mathbf{C}^*_\mathbf{X}\right) \overset{\text{Eq. (47)}}{=}$$

$$2\,\text{tr}\left((\nabla_{\mathbf{X}^{(a)}} \mathbf{C}^*_\mathbf{X})^\top \mathbf{S}_\mathbf{X} \mathbf{C}^*_\mathbf{X} \boldsymbol{\varepsilon}^*_\mathbf{X}\right) = \text{tr}\left((\nabla_{\mathbf{X}^{(a)}} \mathbf{C}^*_\mathbf{X})^\top \nabla_{\mathbf{C}^*_\mathbf{X}} \text{tr}\left((\mathbf{C}^{*\top}_\mathbf{X} \mathbf{S}_\mathbf{X} \mathbf{C}^*_\mathbf{X} - \mathbf{I})\boldsymbol{\varepsilon}^*_\mathbf{X}\right)\right) =$$

$$\nabla_{\mathbf{X}^{(a)}} \text{tr}\left((\mathbf{C}^{*\top}_\mathbf{X} \mathbf{S}_\mathbf{X} \mathbf{C}^*_\mathbf{X} - \mathbf{I})\boldsymbol{\varepsilon}^*_\mathbf{X}\right) \overset{\text{Eq. (43)}}{=} 0.$$

If in practice $\mathbf{C}^*_\mathbf{X}$ is not exactly optimized, the contribution from $\nabla_{\mathbf{X}^{(a)}}\mathbf{C}^*_\mathbf{X}$ should also be considered.

[00253] Note that only the $-\left(\nabla_{\mathbf{X}^{(a)}} \widetilde{\mathbf{V}}_{\text{ext},\mathbf{X}}{}^\top \bar{\boldsymbol{\Gamma}}^*_\mathbf{X}\right)$ term ($\bar{\boldsymbol{\Gamma}}^*_\mathbf{X}$ is the vector of flattened optimal density matrix $\bar{\boldsymbol{\Gamma}}^*_\mathbf{X} := \mathbf{C}^*_\mathbf{X}\mathbf{C}^{*\top}_\mathbf{X}$), as a part of $-\left(\nabla_{\mathbf{X}^{(a)}} E_\mathbf{X}\right)(\mathbf{C}^*_\mathbf{X})$, corresponds to the HF force (see Eq. (42)). Other terms in the analytical force $\mathbf{f}^{(a)}_{\text{ana}}$ in Eq. (72) are collectively called the Pulay force. They come in the form of the gradient of the basis functions w.r.t atom coordinates, hence are non-zero when using atomic basis and differentiate the analytical force from the HF force.

[00254] Implementation

[00255] Although the analytical force is regarded as a more accurate estimation when using atomic basis, the HF force is still taken to evaluate the results, since the way to calculate the analytical force is different for different methods: KSDFT and M-OFDFT require different types of Coulomb integrals under different basis sets, and M-PES and M-PES-Den only require back-propagating the gradient through the machine learning model. Moreover, the HF force in Eq. (71) only depends on the density from the ground-state solution, so it can also be seen as a relevant metric to evaluate the solved density.

[00256] For each molecular system, the true value of HF force is taken as that given by the KSDFT solution, where the density is taken after density fitting (see Supplementary Sec. A.4.1). The HF force by M-OFDFT is calculated directly from the optimized density. For M-PES and M-PES-Den, as they cannot provide the density, only the corresponding analytical forces, $-\nabla_{\mathbf{X}^{(a)}} E_\theta(\mathbf{X})$ and $-\nabla_{\mathbf{X}^{(a)}} E_\theta(\mathbf{X}, \mathrm{p}^{\mathrm{init}})$, are available. To convert them into HF forces, extract from them with the Pulay force from the KSDFT calculation: $\mathrm{f}^{(a)}_{\mathrm{M-PES,X}} = -\nabla_{\mathbf{X}^{(a)}} E_\theta(\mathbf{X}) - \left(\mathrm{f}^{(a)}_{\mathrm{ana,X}} - \mathrm{f}^{(a)}_{\mathrm{X}} \right)$, and similarly for $\mathrm{f}^{(a)}_{\mathrm{M-PES-Den,X}}$, where $\mathrm{f}^{(a)}_{\mathrm{ana,X}}$ and $\mathrm{f}^{(a)}_{\mathrm{X}}$ are calculated from KSDFT. The error in predicted force is measured by the mean absolute error (MAE) over each of the three spacial components of the force on each atom in each molecule in the test set.

[00257] D.5 Curve Fitting

[00258] For fitting the curves in Fig. 3, the formulas $(aN_{\mathrm{heavy}} + b)^c + d$ of all curves in each figure are restricted to have the same scale, i.e., the same a. Otherwise, the flexibility of a would diminish the reflection of the exponent c on the curvature of the curve on the provided range of N_{heavy} (*i.e.*, a smaller a allows a larger exponent c). Sharing the same **a** in different curves hardly hinders the closeness to fitted data points.

A two-phase fitting strategy is put forward based on this idea: (1) jointly fit all curves with a shared a, obtaining the pre-optimized a'; (2) taking a' as an initial guess, re-fit all curves independently to obtain the post-optimized a^* using the Trust Region Reflective optimization algorithm, where the search space of variable a is restricted into $[a'(1-\xi), a'(1+\xi)]$, with $\xi = 0.5$. The two-phase fitting strategy approximately keeps all post-optimized a^* in the same scale as well as bring better fitting accuracy. The fitting process is conducted using the SciPy package in Python.

[00259] E Implementation Details

[00260] E.1 Datasets

[00261] To substantiate the efficacy of the proposed method, evaluations were conducted on two distinct molecular datasets: Ethanol and QM9. These datasets are specifically chosen to evaluate the generalization performance of M-OFDFT within both conformation space and chemical space. Furthermore, the inventors sought to assess the extrapolation capability of M-OFDFT with larger molecular systems, for which two additional datasets are prepared: QMugs and Chignolin. Comprehensive supplementary information and generation details for each dataset are provided below.

[00262] E.1.1 Ethanol

[00263] To investigate the generalization ability of M-OFDFT in conformation space, a set of 100,000 non-equilibrium ethanol geometries was generated, randomly drawn from the MD17 dataset. These geometries were then randomly partitioned into training, validation and test sets using an 8:1:1 ratio.

[00264] E.1.2 QM9

[00265] The QM9 dataset serves as a popular benchmark for predicting quantum-chemical properties using deep learning methods. It comprises equilibrium geometries of approximately 134k small organic molecules, composed of H, C, O, N, and F atoms. These molecules represent a subset of all species with up to nine heavy atoms from the GDB-17 chemical universe dataset. As the dataset contains only equilibrium geometries, it was employed as a benchmark for evaluating the generalization ability of M-OFDFT in chemical space. Furthermore, to compare M-OFDFT with classical KEDFs, the 6905 isomers of $C_7H_{10}O_2$ present in the QM9 dataset were included as a benchmark for assessing M-OFDFT in conformation space. It was continued to randomly split the molecules into training, validation, and test sets using an 8:1:1 ratio.

[00266] E.1.3 QMugs

[00267] The QMugs dataset, proposed by Isert et al., comprises over 665k biologically and pharmacologically relevant molecules extracted from the ChEMBL database, totaling approximately 2M conformers. Notably, QMugs provides molecular samples that are considerably larger than those in the QM9 and MD17 datasets, with an average of 30.6 and a maximum of 100 heavy atoms per compound. This allows us to study the extrapolation capabilities of M-OFDFT.

[00268] We categorized QMugs molecules based on the number of heavy atoms into bins with a width of 5, except for the first bin, which contains molecules with 10-15 heavy atoms. To construct an extrapolation task, the union of QM9 was divided and the first bin into training and validation sets using a 9:1 ratio, and test M-OFDFT on 50 molecular structures from each of the other bins with increasing scales. To reduce the

costs associated with expensive DFT calculations, only 50 molecular geometries were sampled for each test bin.

[00269] Moreover, to investigate the amount of additional large-scale molecular data required by end-to-end counterparts to achieve extrapolation performance comparable to that of M-OFDFT, a series of training sets was constructed with gradually increasing molecular scales. Specifically, initially a subset were sampled from each of the first four bins in QMugs molecules and assemble these subsets, along with the entire QM9 dataset, into four training sets. These training sets used elaborate composition ratios designed to maintain the total size of all training sets while reflecting the ratio of each bin in the original QMugs dataset. The composition ratios of different bins are shown in Fig. 15.

[00270] E.1.4 Chignolin

[00271] Data Selection

[00272] To further evaluate the extrapolation ability of M-OFDFT on large-scale biomolecular molecules, such as proteins, sampled 1,000 Chignolin conformations were sampled from an extensive molecular dynamic (MD) simulation. Due to its fast-folding property and short length (TYR-TYR-ASP-PROGLU-THR-GLY-THR-TRP-TYR), Chignolin has been widely used in molecular dynamics studies. First the backbone conformations were featurized into all pairwise α-carbon-atom distances, excluding pairs of nearest neighbor residues. Then, a time-lagged independent component analysis (TICA) was conducted with a lag time of $\tau_{lag} = 20$ ns, projecting the conformational space into a low-dimensional subspace. The first eight TICA components were clustered into 1,000 groups using the k-Means algorithm. The

centroid structure of each cluster was extracted using MDTraj and taken as the representative structure.

[00273] Conformation Neutralization

[00274] In this study the focus is on isolated molecular systems where all atoms or functional groups stay neutral and all electrons are paired. However, protein molecules extracted from MD trajectories often contain many ionizable groups, such as the amine group in the N-terminal amino acid and the carboxyl group in the N-terminal amino acid. Several strategies have been proposed to neutralize ionizable groups in protein conformations, including adding neutral terminal caps (e.g., n-methyl and methyl-acetate) and incorporating counterions. But introducing extra terminal groups or ions into extracted protein conformations could cause significant geometry changes to the original conformations and potentially lead to instability. To avoid this, protein conformations were neutralized by editing (adding/deleting) missing/extra hydrogen atoms in ionizable groups using OpenMM. The hydrogen definition template was modified to handle hydrogen atoms not defined in standard amino acids, such as the missing hydrogen atom in the carboxyl group in the C-terminal amino acid. Then local energy optimization was performed on the neutralized conformations using Amber, with the Amber-ff14SB force field and a maximum of 100 optimization cycles. To prevent significant geometry changes, only hydrogen atoms associated with heavy atoms involved in the hydrogen editing process were allowed to be optimized. For neutralized amino acids without standard force field templates, force field parameters were built using the antechamber and parmchk2 tools in Amber.

[00275] Protein Fragmentation

[00276] To investigate the extrapolation utility of M-OFDFT from "local" fragments to "global" proteins, datasets were created by cutting the 1,000 Chignolin conformations into protein fragments (i.e., polypeptides) of varying lengths. To construct these fragments, all possible polypeptides with sequence lengths up to five in the Chignolin sequence were enumerated. Specifically, residue gaps were allowed to exist in tetrapeptides and pentapeptides, i.e., dipeptide-[GAP]-dipeptide and tripeptide-[GAP]dipeptide. The extracted fragment molecules were neutralized and then used as training and validation sets with a 9:1 ratio. It should be noted that when benchmarking the performance of M-OFDFT models trained with different molecule scales, the training set with a maximum peptide length of L_{pep} is comprised of all polypeptides with sequence lengths less than L_{pep}.

[00277] Data Filtration

[00278] After preliminary analysis, the range of energy and gradient labels in fragments datasets was found to be too vast to fit even if equipped with several efficient training techniques (see Supplementary Sec. B.3). Moreover, most of the vast data points come from the first several steps of the SCF iterations that are far from convergent status. To mitigate this issue, an SCF-augmented data point was filtered out if the residual between its energy and the ground-state energy is larger than 500kcal/mol. This strategy can empirically improve the optimization robustness and prediction performance.

[00279] E.2 Model Configuration

[00280] In all experimental settings, the same backbone architecture, Graphormer, was employed, specifically utilizing the Graphormer-3D (G3D) encoder layer. To ensure a fair comparison, most hyperparameters in all models are maintained

consistently (e.g., model depth and hidden dimension). A summary of all hyperparameter choices can be found in Supplementary Table 3. It is worth mentioning that in the node embedding layer's shrink gate, the initial value of learnable parameters v_{co} is set to 10, while the initial value of v_{mul} is set to 0.02 for the ethanol dataset and 0.05 for all other datasets. No extensive hyperparameter search was conducted, and most hyperparameters were chosen following Graphormer.

Table 3: Hyperparameters of the Graphormer model used in all methods.

Hyperparameter	M-OFDFT
G3D Layers	12
Hidden Dimension	768
Feed-Foward Hidden Dimension	768
Number of Heads	32
GBF Dimension	128
Dropout	0.1
Attention Dropout	0.1
Optimizer	Adam
Learning Rate Schedule	Linear decay
Adam (β_1, β_2)	(0.95, 0.99)
Adam ϵ	1×10^{-8}

[00281] E.3 Training Hyperparameters

[00282] Following Graphormer, all models are trained using the Adam optimizer and a linear decay learning schedule. The peak learning rate is set to 1×10^{-4} for functional models and 3×10^{-4} for baseline models (*i.e.*, M-PES and M-PES-Den). Other optimizer hyperparameters can be found in Supplementary Table 3. A warmup stage featuring a linearly increasing learning rate is introduced to stabilize training during the initial stage. The number of warmup steps is set to 30k for M-OFDFT and 60k for baseline models. The batch size is set to 256 for the ethanol dataset and 128 for all other datasets. All models are trained on Nvidia Tesla V100 GPUs.

[00283] For different molecule datasets, the number of training epochs is determined by examining the loss curve. Training is halted once the validation loss fails to decrease for 20 epochs. Specifically, functional models were trained for approximately 600 and 700 epochs on the Ethanol and QM9 datasets, respectively. For the QMugs dataset, the model is trained for approximately 700 epochs, while the M-PES and M-PESDen models are trained for 2,300 epochs. Notably, a series of QMugs datasets is proposed with increasing molecular size in Results 2.4, where the number of SCF data points will slightly increase as the average molecular size increases (larger molecules generally require more SCF iterations to converge). To ensure a fair comparison, approximately the same number of training epochs were maintained for different datasets. In the Chignolin experiment, there are four Chignolin datasets with increasing peptide length and data size. The functional models are trained for 1,400, 1,100, 800, and 750 epochs, respectively, while the M-PES and M-PES-Den models are trained for 3,000, 3,000, 1,000, and 900 epochs, respectively.

[00284] In practice, a weighted loss function was employed to optimize the KEDF model:

$$L = \xi_{\text{eng}} L_{\text{eng}} + \xi_{\text{grad}} L_{\text{grad}} + \xi_{\text{den}} L_{\text{den}},$$

where L_{eng}, L_{grad}, and L_{den} represent the KEDF energy loss (Eq. (2)), gradient loss (Eq. (3)), and projected density loss (Eq. (68)), respectively. ξ_{eng}, ξ_{grad}, and ξ_{den} are the corresponding loss weights. The selection of loss weights is determined through grid search on the validation set for various datasets independently, and the utilized loss weights are provided in Supplementary Table 4.

Table 4: Loss weights for various datasets and learning targets.

Dataset	ξ_{eng}	ξ_{grad}	ξ_{den}
Ethanol-$T_{\text{S,res}}$	1	0.1	0.08
Ethanol-E_{TXC}	1	0.03	1
QM9-$T_{\text{S,res}}$	1	0.12	0.005
QM9-E_{TXC}	1	0.12	0.05
QMugs	1	0.1	1
Chignolin	1	0.1	1

[00285] E.4 Density Optimization Hyperparameters

[00286] During the deployment stage, the adopted gradient descent steps and step size ε are chosen according to the performance of evaluation datasets. The gradient descent steps and step size are set to 1000 and 1×10^{-3} for all molecular datasets and functional variants (including traditional KEDF baselines), with the exception of implementing the learned KEDF models $T_{\text{S,res},\theta}$ and $E_{\text{TXC},\theta}$ on the ethanol dataset, where the step size ε is set to 5×10^{-4}. A Stochastic Gradient Descent (SGD) optimizer is adopted in the density optimization process.

[00287] F Additional Results

[00288] F.1 Additional Visualizations of Optimized Densities

[00289] To further investigate the validity of the optimized density produced by M-OFDFT, the integrated density centered on two other heavy atoms in the ethanol molecule is visualized in Fig. 16. The results demonstrate that the density of M-OFDFT closely aligns with the KSDFT density for both atoms. In contrast, the classical KEDF of APBE struggles to accurately recover the shell structure, especially in the minor peaks associated with covalent bonds. In these regions, the APBE density exhibits a substantial deviation from the ground-truth KSDFT density.

[00290] Table 5 shows the performance of M-OFDFT with different learning-target and initialization configurations. The mean absolute errors (MAEs) in energy and Hellmann-Feynman (HF) force are listed in kcal/mol and kcal/mol/A, respectively.

Table 5

Method	Ethanol		OM9	
	Energy	HF force	Energy	HF force
$T_{S,res}$-Huckel"	5.27	27.73	9.88	12.73
$T_{S,res}$-ProjMINAO	0.18	1.18	0.93	2.91
E_{TXC}-Huckel"	3.95	126.65	9.69	123.51
E_{TXC}-ProjMINAO	0.18	2.07	0.73	4.73

[00291] F.2 More In-Scale Results

[00292] As explained in Methods 4.4, two initialization strategies are proposed to address the out-of-distribution issue of initialized densities. The results in Fig. 2(a)-(b) highlight the exceptional performance of the ProjMINAO initialization. Notably, as shown in Supplementary Table 5, even without the machine learned shortcut, M-OFDFT still achieves a similar order of chemical accuracy when using the standard Huckel initialization. Since the density corrector provides a starting point close to the ground state, where the model approximates the functional more confidently, the ProjMINAO initialization can yield better ground-state densities and significantly lower HF force loss. This demonstrates the importance of incorporating the density corrector prediction branch.

[00293] F.3 More Extrapolation Results

[00294] As previously mentioned, the extrapolation capability of M-OFDFT was evaluated on two datasets, QMugs and Chignolin, using two end-to-end counterparts with the same architecture, M-PES and M-PES-Den, as baselines. It is important to note that while M-PES and M-PES-Den may achieve better validation errors, which does not necessarily imply superior extrapolation performance for larger-scale molecules. In contrast to M-OFDFT, M-PES methods rely solely on molecular structure to model the total energy, neglecting the fine-grained interaction process among electrons. As a result, the end-to-end mapping from M to E appears highly complex and subtle, making generalization to molecular systems with unseen structural patterns challenging. This observation is supported by the sharp increase in generalization error of the M-PES and M-PES-Den models in energy prediction, as demonstrated in Supplementary Table 6.

[00295] Furthermore, M-OFDFT also exhibits superior extrapolation performance in HF force prediction. As illustrated in Fig. 17 at (a), the M-OFDFT model, trained on molecules with fewer than 15 heavy atoms, consistently maintains a lower HF force MAE than M-PES and M-PES-Den on larger-scale molecules. Although M-PES-Den has a relatively lower error-increasing exponent, M-OFDFT achieves a significantly lower absolute MAE. Additionally, the disclosed method outperforms the two end-to-end counterparts across various fragment datasets in the Chignolin experiment.

[00296] Table 6 shows generalization results of M-PES and M-PES-Den baselines on QMugs and Chignolin datasets. For QMugs, both the M-PES and M-PES-Den models are trained on molecules with fewer than 15 heavy atoms and tested on 50 QMugs conformations with 56-60 heavy atoms. For Chignolin, the two models are trained on fragment conformations with all peptide lengths (2-5) and tested on 1,000 Chignolin conformations. The per-atom mean absolute errors (MAEs) in energy for the

training set, the validation set, and energy from density optimization on larger-scale systems (denoted as *energy prediction*) are listed in kcal/mol.

Table 6

Method	QMugs			Chignolin		
	Train	Validation	Energy Prediction	Train	Validation	Energy Prediction
M-PES	0.008	0.017	1.768	0.003	0.003	0.084
M-PES-Den	0.004	0.019	1.824	0.002	0.002	0.079
M-OFDFT	0.044	0.055	0.112	0.015	0.016	0.071

[00297] F.4 Ablation Study

[00298] F.4.1 Multi-Step Data and Gradient Label

[00299] To capture the energy landscape in the density coefficient space, multiple **p** samples were generated for each molecular structure M and compute the gradient label $\nabla_{\mathbf{p}} T_S$ for additional supervision information. An ablation experiment is conducted on the Ethanol dataset to investigate the significance of each type of supervision by excluding the supervision label during the training process. The ablation results are presented in Supplementary Table 7. Removing the gradient label $\nabla_{\mathbf{p}} T_S$ leads to a considerable decrease in energy and HF force accuracy for both density initialization strategies, emphasizing the importance of maintaining the optimization of M-OFDFT on a physical track. Incorporating multi-step **p** samples is also crucial for enhancing the performance of M-OFDFT, particularly for the ProjMINAO initialization, as the accuracy of the density corrector model depends on the size of training densities.

[00300] Table 7 shows an ablation study for various data augmentation strategies. All results are evaluated on test ethanol molecules with the learned $T_{S,res,\theta}$ model. The MAEs in energy and HF force are listed in kcal/mol and kcal/mol/A, respectively.

Table 7

Multi-step $\mathbf{p}$	Gradient $\nabla_{\mathbf{p}}T_S$	Density Initialization	Energy	HF Force
✓	✓	Hückel	5.27	27.73
		ProjMINAO	0.18	1.18
✓		Hückel	166.84	144.78
		ProjMINAO	170.83	143.67
	✓	Hückel	6.43	32.83
		ProjMINAO	20.61	38.56

[00301] F.4.2 Non-locality

[00302] Considering the non-local dependency of KEDF on electron density, it is essential to incorporate non-local calculations in OFDFT, which inspires us to use Graphormer as a backbone architecture. To demonstrate the importance of non-locality, an ablation was conducted by gradually decreasing the receptive field (i.e., the distance cutoff of atom neighborhood) of the M-OFDFT model. As illustrated in Fig. 18, both the energy and HF force prediction accuracy tend to worsen as the distance cutoff decreases, providing empirical evidence of the importance of incorporating non-locality into the KEDF model.

[00303] F.4.3 Density Feature Preprocessing Module

[00304] The geometric variability of input data and the vast range of gradient data make optimizing the KEDF objective challenging. Therefore, a density feature preprocessing module is proposed, which implements several techniques to enable effective optimization. An ablation experiment was conducted to study the importance of the local frame and natural reparameterization techniques. As shown in

Supplementary Table 8, incorporating the local frame results in a considerable performance improvement compared to the baseline model without these techniques, demonstrating the capacity of the local frame to reduce the geometric variability of input data.

[00305] Furthermore, since different basis functions have varying importance in representing a density, some coefficient dimensions can significantly influence the density function and the total energy. This is difficult for M-OFDFT to handle, as the model treats each dimension equally. During density optimization, this influence is amplified when the input density is far from the ground state. This issue was addressed by introducing the natural reparameterization technique to balance the impact of each coefficient dimension. The results in Supplementary Table 8 show that implementing natural reparameterization can significantly improve the performance of the model, especially for the Huckel initialization, which is far from the ground-state" density. Here, the energy MAE is reduced by one or two orders of magnitude, highlighting the powerful capability of natural reparameterization in balancing sensitivities across coefficient dimensions.

[00306] We also attempted to examine the importance of the atomic reference model and the dimension-wise rescaling techniques but found it difficult to optimize the learning objective when removing either technique from the model. Consequently, the unreasonable evaluation metrics are omitted in Supplementary Table 8. The substantial impact on efficient training and the quantitative results for reducing geometric variability, as clarified in Supplementary Sec. B.3.3, strongly support the necessity of these two techniques.

[00307] Table 8 shows results of an ablation study for components in the density preprocessing module. All results are evaluated on test ethanol molecules with the learned $T_{S,res,\theta}$ model. The MAEs in energy and HF force are listed in kcal/mol and kcal/mol/A, respectively.

Table 8

Local Frame	Nat. Reparam.	Density Initialization	Energy	HF Force
		Huckel"	567.69	73.17
		ProjMINAO	0.57	2.02
✓		Huckel"	514.56	36.55
		ProjMINAO	0.30	1.29
	✓	Huckel"	25.41	37.18
		ProjMINAO	0.18	1.81
✓	✓	Huckel"	5.77	27.73
		ProjMINAO	0.18	1.18

[00308] F.4.4 Results using other training strategies

[00309] To provide the model with information on the optimization landscape, the training data was enriched with gradient labels and labels for multiple densities corresponding to the same molecular structure. While there are alternative methods to supervise the optimization behavior of a functional model, experiments were conducted indicate that they are less effective in observed settings. In the context of learning XC functionals, Kirkpatrick et al. introduced an SCF loss to regularize the objective landscape, but effectively acts as a gradient supervision only at the ground state (converged orbitals). In contrast, M-OFDFT utilizes multi-step data and gradient supervision to capture the finer energy landscape, leading to a substantial performance improvement (Supplementary Table 7).

[00310] Some other studies implicitly enforce the energy landscape for optimization by supervising the model optimized energy and density (or orbitals). This considerably complicates the model optimization process. To address this challenge, some works resort to inefficient gradient-free optimization methods or back-propagate the loss through a series of eigenvalue solvers, which is both costly and difficult to generalize. Specifically, the approach by Li et al. was attempted, which involves supervising model-optimized energy and density. Optimizing the model is extremely challenging due to the need to back-propagate the parameter gradient through a series of density-optimization steps. Even after eliminating the grid completely by training an E_{TXC} model to save costs and pretraining the model on data to simplify optimization, the method still cannot optimize the model with more than 8 density optimizing steps. In experiments that were conducted on QM9, the best result could provide a reasonable solution after the optimization steps used in training. However, the energy does not converge if the optimization continues, indicating that the model does not learn a physical functional (not N-representable) and instead resembles an end-to-end predictor by stacking K times, where K is the designated number of optimization steps in training.

[00311] Chen et al. attempt to bypass the costly back-propagation through iterative density optimization by assuming the model-optimized state is exact. In detail, this method uses the optimal density from the current model paired with the true ground-state energy label as a data pair for the next step. Since finding the minimizing density after every model update is expensive, the method opts to optimize the model and density alternately. However, this modification disrupts the property and the optimization process may not perform as expected. Furthermore, this approach still primarily focuses on the landscape near the ground state, which may not effectively guide the initialized density. In an experiment on Chignolin by the inventors, the

alternative optimization was repeated for four iterations but observed only marginal performance improvement. Specifically, the performance gains for each iteration are 0.5%, -0.06%, 1.7%, and -0.8%, respectively, implying that such a strategy is ineffective in capturing the energy landscape.

[00312] F.5 Scalability study on large protein systems

[00313] With the aid of sophisticated atomic orbitals and numerical optimization, KSDFT can be extended to molecular systems of up to approximately 500 atoms using a modern PC cluster. To further showcase the scaling advantage of M-OFDFT, the disclosed model was tested on two large-scale protein systems that are rarely encountered in the context of KSDFT calculations (Results 2.3). Given the considerable scale gap between training molecules and target molecules, the M-OFDFT model was trained on all Chignolin fragments and finetuned on 800 Chignolin conformations as the surrogate KEDF model, which can theoretically provide the most robust extrapolation capacity. The results indicate that M-OFDFT can achieve a 25.6-fold and 27.4-fold speedups on the two test systems, respectively, compared to KSDFT calculations. The per-atom energy MAE of M-OFDFT for the two test systems is 0.23kcal/mol and 0.31kcal/mol, respectively. While higher than the prediction error for Chignolin molecules, the result still demonstrates a significant advantage over M-PES (0.36kcal/mol and 0.63kcal/mol, respectively).

[00314] In some embodiments, the methods and processes described herein may be tied to a computing system of one or more computing devices. In particular, such methods and processes may be implemented as a computer-application program or service, an application-programming interface (API), a library, and/or other computer-program product.

[00315] FIG. 19 schematically shows a non-limiting embodiment of a computing system 1000 that can enact one or more of the methods and processes described above. Computing system 1000 is shown in simplified form. Computing system 1000 may embody the computing system 10 described above and illustrated in FIG. 1B and FIG. 8. Components of computing system 1000 may be included in one or more personal computers, server computers, tablet computers, home-entertainment computers, network computing devices, video game devices, mobile computing devices, mobile communication devices (e.g., smartphone), and/or other computing devices, and wearable computing devices such as smart wristwatches and head mounted augmented reality devices.

[00316] Computing system 1000 includes processing circuitry 1002, volatile memory 1004, and a non-volatile storage device 1006. Computing system 1000 may optionally include a display subsystem 1008, input subsystem 1010, communication subsystem 1012, and/or other components not shown in FIG. 19.

[00317] Processing circuitry typically includes one or more logic processors, which are physical devices configured to execute instructions. For example, the logic processors may be configured to execute instructions that are part of one or more applications, programs, routines, libraries, objects, components, data structures, or other logical constructs. Such instructions may be implemented to perform a task, implement a data type, transform the state of one or more components, achieve a technical effect, or otherwise arrive at a desired result.

[00318] The logic processor may include one or more physical processors configured to execute software instructions. Additionally or alternatively, the logic processor may include one or more hardware logic circuits or firmware devices configured to execute hardware-implemented logic or firmware instructions.

Processors of the processing circuitry 1002 may be single-core or multi-core, and the instructions executed thereon may be configured for sequential, parallel, and/or distributed processing. Individual components of the processing circuitry optionally may be distributed among two or more separate devices, which may be remotely located and/or configured for coordinated processing. For example, aspects of the computing system disclosed herein may be virtualized and executed by remotely accessible, networked computing devices configured in a cloud-computing configuration. In such a case, these virtualized aspects are run on different physical logic processors of various different machines, it will be understood. These different physical logic processors of the different machines will be understood to be collectively encompassed by processing circuitry 1002.

[00319] Non-volatile storage device 1006 includes one or more physical devices configured to hold instructions executable by the processing circuitry to implement the methods and processes described herein. When such methods and processes are implemented, the state of non-volatile storage device 1006 may be transformed—e.g., to hold different data.

[00320] Non-volatile storage device 1006 may include physical devices that are removable and/or built in. Non-volatile storage device 1006 may include optical memory, semiconductor memory, and/or magnetic memory, or other mass storage device technology. Non-volatile storage device 1006 may include nonvolatile, dynamic, static, read/write, read-only, sequential-access, location-addressable, file-addressable, and/or content-addressable devices. It will be appreciated that non-volatile storage device 1006 is configured to hold instructions even when power is cut to the non-volatile storage device 1006.

[00321] Volatile memory 1004 may include physical devices that include random access memory. Volatile memory 1004 is typically utilized by processing circuitry 1002 to temporarily store information during processing of software instructions. It will be appreciated that volatile memory 1004 typically does not continue to store instructions when power is cut to the volatile memory 1004.

[00322] Aspects of processing circuitry 1002, volatile memory 1004, and non-volatile storage device 1006 may be integrated together into one or more hardware-logic components. Such hardware-logic components may include field-programmable gate arrays (FPGAs), program- and application-specific integrated circuits (PASIC / ASICs), program- and application-specific standard products (PSSP / ASSPs), system-on-a-chip (SOC), and complex programmable logic devices (CPLDs), for example.

[00323] The terms "module," "program," and "engine" may be used to describe an aspect of computing system 1000 typically implemented in software by a processor to perform a particular function using portions of volatile memory, which function involves transformative processing that specially configures the processor to perform the function. Thus, a module, program, or engine may be instantiated via processing circuitry 1002 executing instructions held by non-volatile storage device 1006, using portions of volatile memory 1004. It will be understood that different modules, programs, and/or engines may be instantiated from the same application, service, code block, object, library, routine, API, function, etc. Likewise, the same module, program, and/or engine may be instantiated by different applications, services, code blocks, objects, routines, APIs, functions, etc. The terms "module," "program," and "engine" may encompass individual or groups of executable files, data files, libraries, drivers, scripts, database records, etc.

[00324] When included, display subsystem 1008 may be used to present a visual representation of data held by non-volatile storage device 1006. The visual representation may take the form of a graphical user interface (GUI). As the herein described methods and processes change the data held by the non-volatile storage device, and thus transform the state of the non-volatile storage device, the state of display subsystem 1008 may likewise be transformed to visually represent changes in the underlying data. Display subsystem 1008 may include one or more display devices utilizing virtually any type of technology. Such display devices may be combined with processing circuitry 1002, volatile memory 1004, and/or non-volatile storage device 1006 in a shared enclosure, or such display devices may be peripheral display devices.

[00325] When included, input subsystem 1010 may comprise or interface with one or more user-input devices such as a keyboard, mouse, touch screen, camera, or microphone.

[00326] When included, communication subsystem 1012 may be configured to communicatively couple various computing devices described herein with each other, and with other devices. Communication subsystem 1012 may include wired and/or wireless communication devices compatible with one or more different communication protocols. As non-limiting examples, the communication subsystem may be configured for communication via a wired or wireless local- or wide-area network, broadband cellular network, etc. In some embodiments, the communication subsystem may allow computing system 1000 to send and/or receive messages to and/or from other devices via a network such as the Internet.

[00327] "And/or" as used herein is defined as the inclusive or $\vee$, as specified by the following truth table:

A	B	A $\vee$ B

True	True	True
True	False	True
False	True	True
False	False	False

[00328] It will be understood that the configurations and/or approaches described herein are exemplary in nature, and that these specific embodiments or examples are not to be considered in a limiting sense, because numerous variations are possible. The specific routines or methods described herein may represent one or more of any number of processing strategies. As such, various acts illustrated and/or described may be performed in the sequence illustrated and/or described, in other sequences, in parallel, or omitted. Likewise, the order of the above-described processes may be changed.

[00329] The subject matter of the present disclosure includes all novel and non-obvious combinations and sub-combinations of the various processes, systems and configurations, and other features, functions, acts, and/or properties disclosed herein, as well as any and all equivalents thereof.

CLAIMS

1. A computing system, comprising:

processing circuitry and an associated non-volatile storage device storing instructions that upon execution by the processing circuitry cause the processing circuitry to:

at inference time:

receive inference-time molecular structure data for an inference-time molecular structure, the inference time molecular structure data including inference-time atomic basis density expansion coefficients and data indicating atomic attributes of each atom in the inference-time molecular structure;

input the inference-time molecular structure data to a trained kinetic energy density functional machine learning model to thereby generate a predicted value for electronic non-interacting kinetic energy of a density function, wherein the trained machine learning model has been trained on a training data set that includes, as input, atomic attributes of each atom in a training-time molecular structure, and atomic basis density expansion coefficients for the training-time molecular structure, the atomic basis density expansion coefficients indicating values by which electron charge density is represented on an atomic basis, and that includes corresponding ground truth values for non-interacting kinetic energy; and

output the predicted value for non-interacting kinetic energy for the inference-time molecular structure.

2. The computing system of claim 1, wherein the atomic attributes include one or more of atomic position and/or atomic number.

3. The computing system of claim 1, wherein the density function is based on atomic basis density expansion coefficients.

4. The computing system of claim 1,

the training data set further includes ground truth values for kinetic energy density expansion gradients; and

the trained kinetic energy density function machine learning model is further configured to output a predicted value of the kinetic energy density expansion gradient through backpropagation at inference time.

5. The computing system of claim 4, wherein the processing circuitry is further configured to perform an orbital-free Density Functional Theory (DFT) calculation using the predicted value of the kinetic energy density expansion gradient.

6. The computing system of claim 1, wherein

the processing circuitry is configured to train the kinetic energy density function machine learning model on a training data set that is generated at least in part by:

at training time:

based on orbital values computed in intermediate cycles of a self-consistent field (SCF) algorithm evaluation of training-time molecular structure data for the training-time molecular structure, computing the atomic basis density expansion coefficients, non-interacting kinetic energy, and gradients of the non-interacting kinetic energy for the intermediate cycles; and

storing the density expansion coefficients and training-time molecular structure as training-time inputs, and the non-interacting kinetic energy and the kinetic energy density expansion gradients as ground truth output in the training data set.

7. The computing system of claim 6, wherein the kinetic energy density function machine learning model has a graph neural network transformer architecture configured to implement a non-local attention mechanism.

8 The computing system of claim 7, wherein the kinetic energy density function machine learning model includes a transformer neural network with a multi-headed attention mechanism that is trained by backpropagation using the training-time inputs and the ground truth output.

9. The computing system of claim 8, wherein the multi-headed attention mechanism is configured to compute attention and loss for a predictive task of predicting the non-interacting kinetic energy given the density expansion coefficients as input, and is configured to compute loss by backpropagation with respect to the input density expansion coefficients for a predictive task of predicting the kinetic energy density expansion gradient.

10. The computing system of claim 6, wherein

the training-time molecular structure data includes data indicating atomic coordinates, atomic numbers, number of electrons in the training-time molecular structure, and an orbital basis set;

the SCF algorithm is executed at least in part by iterating through a plurality of SCF cycles, and in each SCF cycle:

compute a plurality of orbital coefficients by which the orbital basis set is expanded to determine orbital values for each electron;

based on the orbital coefficients, construct a matrix approximating the single-electron energy operator of a given quantum system;

using the constructed matrix, compute updated orbital coefficients;

based on the updated orbital coefficients, compute non-interacting kinetic energy;

compute a gradient value for the non-interacting kinetic energy;

compute density expansion coefficients under atomic basis based on the orbital coefficients;

store values for the non-interacting kinetic energy density expansion coefficients, non-interacting kinetic energy, and kinetic energy density expansion gradients;

determine whether a total energy and/or the density matrix has converged to within a convergence threshold, and if converged, update a density matrix for the training-time molecular structure based on the orbital coefficients and cease iterating.

11. The computing system of claim 10, wherein the matrix approximating the single-electron energy operator of a given quantum system is a Fock matrix.

12. The computing system of claim 10, wherein the training-time molecular structure data is stored in a graph data structure, the atomic position and atomic number

are represented as node features in the graph data structure, and the atomic positions are used to compute edge values in the graph data structure.

13. The computing system of claim 1, wherein the density coefficients are expressed in a local atomic frame of reference, which is equivariant to global shift and rotation, and makes the density expansion coefficients SE(3)-invariant.

14. The computing system of claim 1, wherein the density coefficients are subject to natural reparameterization to make a Euclidean metric in a reparameterized coefficient space trend towards an L2-metric in density-function space.

15. The computing system of claim 14, wherein the density coefficients are subject to dimension-wise rescaling.

16. The computing system of claim 15, wherein an atomic reference model is adopted for the kinetic energy that is linear in density coefficients, such that density coefficients corresponding to a same atom type, species and/or chemical element share a same linear weight.

17. A computerized method implemented via processing circuitry, the method comprising:

at inference time:

receiving inference-time molecular structure data for an inference-time molecular structure, the inference time molecular structure data including inference-

time atomic basis density expansion coefficients and data indicating atomic attributes of each atom in the inference-time molecular structure;

inputting the molecular structure data to a trained kinetic energy density functional machine learning model to thereby generate a predicted value for electronic non-interacting kinetic energy of a density function, wherein the trained machine learning model has been trained on a training data set that includes, as input, atomic attributes of each atom in the training-time molecular structure, and atomic basis density expansion coefficients for the training-time molecular structure, the atomic basis density expansion coefficients indicating values by which electron charge density is represented on an atomic basis, and that includes corresponding ground truth values for non-interacting kinetic energy; and

outputting the predicted value for non-interacting kinetic energy for the inference-time molecular structure.

18. The computerized method of claim 17, wherein the atomic attributes include one or more of atomic position and/or atomic number.

19. The computerized method of claim 17, wherein the density function is based on atomic basis density expansion coefficients.

20. The computerized method of claim 17, wherein

the training data set further includes ground truth values for kinetic energy density expansion gradients, the method further comprising:

outputting from the trained kinetic energy density function machine learning model a predicted value of the kinetic energy density expansion gradient through backpropagation at inference time.

21. The computerized method of claim 20, further comprising:

performing an orbital-free Density Functional Theory (DFT) calculation using the predicted value of the kinetic energy density expansion gradient.

22. The computerized method of claim 21, further comprising:

generating a training data set at least in part by,

based on orbital values computed in intermediate cycles of a self-consistent field (SCF) algorithm evaluation of training-time molecular structure data for the training-time molecular structure, computing the atomic basis density expansion coefficients, non-interacting kinetic energy, and gradients of the non-interacting kinetic energy for the intermediate cycles; and

storing the density expansion coefficients and training-time molecular structure as training-time inputs, and the non-interacting kinetic energy and the kinetic energy density expansion gradients as ground truth output in the training data set; and

training the trained kinetic energy density function machine learning model on the training data set.

23. The computerized method of claim 22, wherein the kinetic energy density function machine learning model has a graph neural network transformer architecture configured to implement a non-local attention mechanism.

24. The computerized method of claim 23, wherein the non-local attention mechanism is a multi-headed attention mechanism that is trained by backpropagation using the training-time inputs and the ground truth output.

25. The computerized method of claim 24, wherein the multi-headed attention mechanism is configured to compute attention and loss for a predictive task of predicting the non-interacting kinetic energy given the density expansion coefficients as input, and is configured to compute loss by backpropagation with respect to the input density expansion coefficients for a predictive task of predicting the kinetic energy density expansion gradient.

26. The computerized method of claim 22, wherein

the training-time molecular structure data includes data indicating atomic coordinates, atomic numbers, number of electrons in the training-time molecular structure, and an orbital basis set; and

the SCF algorithm is executed at least in part by iterating through a plurality of SCF cycles, and in each SCF cycle:

compute a plurality of orbital coefficients by which the orbital basis set is expanded to determine orbital values for each electron;

based on the orbital coefficients, construct a matrix approximating the single-electron energy operator of a given quantum system;

using the constructed matrix, compute updated orbital coefficients;

based on the updated orbital coefficients, compute non-interacting kinetic energy;

compute a gradient value for the non-interacting kinetic energy;

compute density expansion coefficients under atomic basis based on the updated orbital coefficients;

store values for the non-interacting kinetic energy density expansion coefficients, non-interacting kinetic energy, and kinetic energy density expansion gradients; and

determine whether a total energy and/or the density matrix has converged to within a convergence threshold, and if converged, update a density matrix for the training-time molecular structure based on the orbital coefficients and cease iterating.

27. The computerized method of claim 22, wherein

the training-time molecular structure data is stored in a graph data structure, the atomic position and atomic number are represented as node features in the graph data structure, and the atomic positions are used to compute edge values in the graph data structure;

the density coefficients are expressed in a local atomic frame of reference, which is equivariant to global shift and rotation, and makes the density expansion coefficients SE(3)-invariant;

the density coefficients are subject to natural reparameterization to make a Euclidean metric in a reparameterized coefficient space trend towards an L2-metric in density-function space;

the density coefficients are subject to dimension-wise rescaling; and

an atomic reference model is adopted for the kinetic energy that is linear in density coefficients, such that density coefficients corresponding to a same atom type, species and/or chemical element share a same linear weight.

28. The computerized method of claim 21, further comprising:

iteratively optimizing the density expansion coefficient for an inference-time molecular structure using the trained kinetic energy density function machine learning model, by in each of a plurality of iteration cycles:

inputting the molecular structure data and density expansion coefficients to the trained kinetic energy density functional machine learning model to thereby generate a predicted value for electronic non-interacting kinetic energy of the density function, and then generating the predicted kinetic energy density expansion gradient by backpropagation through the trained kinetic energy density functional machine learning model with respect to the input atomic basis density expansion coefficients;

computing the gradient of the sum of the Hartree energy, the external potential energy, and the exchange-correlation energy, with respect to the input atomic basis density expansion coefficients, and adding it to the computed gradient;

projecting the gradient onto a linear subspace of density expansion coefficients that determine normalized densities;

updating the density expansion coefficients using the projected gradient; and

determining whether a total energy change value from a previous cycle or a projected gradient normalized value is larger than a value in the previous cycle, and if larger, cease iterating.

29. The computerized method of claim 28, further comprising:

initializing the density expansion coefficients by adding upon MINAO initialized density with an increment that is predicted using a trained initialization value machine learning model that takes as input the inference-time molecular structure, and

a standard MINAO initialized density expansion coefficient, wherein the trained initialization value machine learning model has been trained on a training data set that includes, as input, atomic positions and data indicating atomic numbers of each atom in the training-time molecular structure, and atomic basis density expansion coefficients for the training-time molecular structure, the atomic basis density expansion coefficients indicating values by which electron charge density is represented on an atomic basis, and that includes corresponding ground truth values for an increment with respect to the density expansion coefficients at a SCF converged step.

30. A computerized method implemented via processing circuitry, the method comprising:

generating a training data set for a kinetic energy density functional machine learning model at least in part by,

based on orbital values computed in intermediate cycles of a self-consistent field (SCF) algorithm evaluation of training-time molecular structure data for a training-time molecular structure, computing atomic basis density expansion coefficients, non-interacting kinetic energy, and gradients of the non-interacting kinetic energy for intermediate cycles of the SCF algorithm; and

storing the atomic basis density expansion coefficients and training-time molecular structure as training-time inputs, and the non-interacting kinetic energy and the kinetic energy density expansion gradients as ground truth output in the training data set;

training the kinetic energy density function machine learning model on the training data set; and

outputting the trained kinetic energy density function machine learning model.

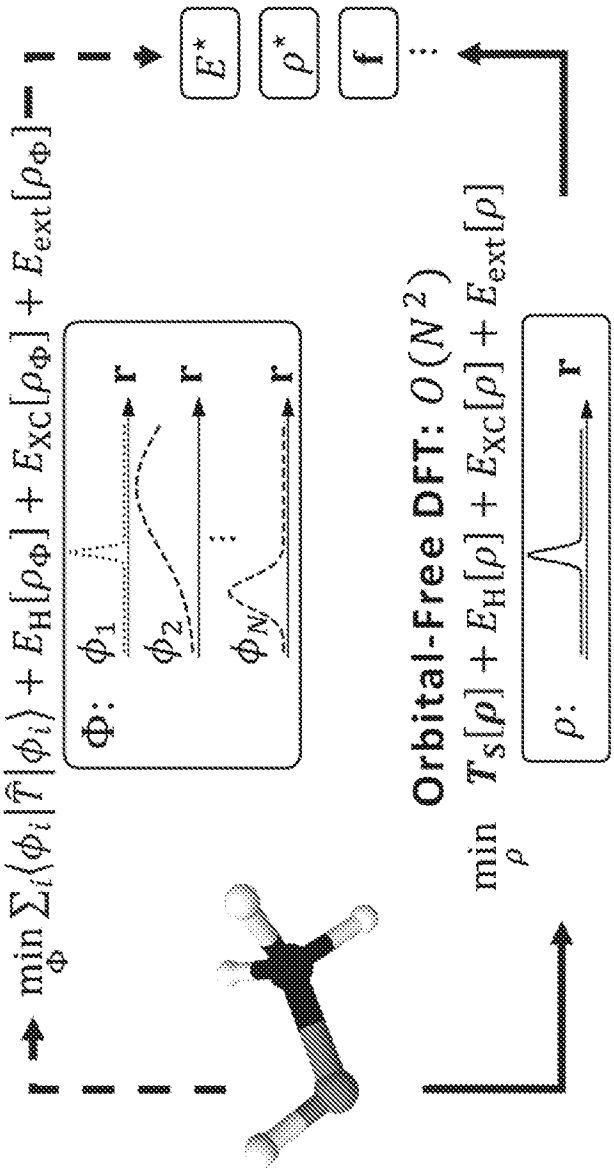


FIG. 1A
(CONVENTIONAL ART)

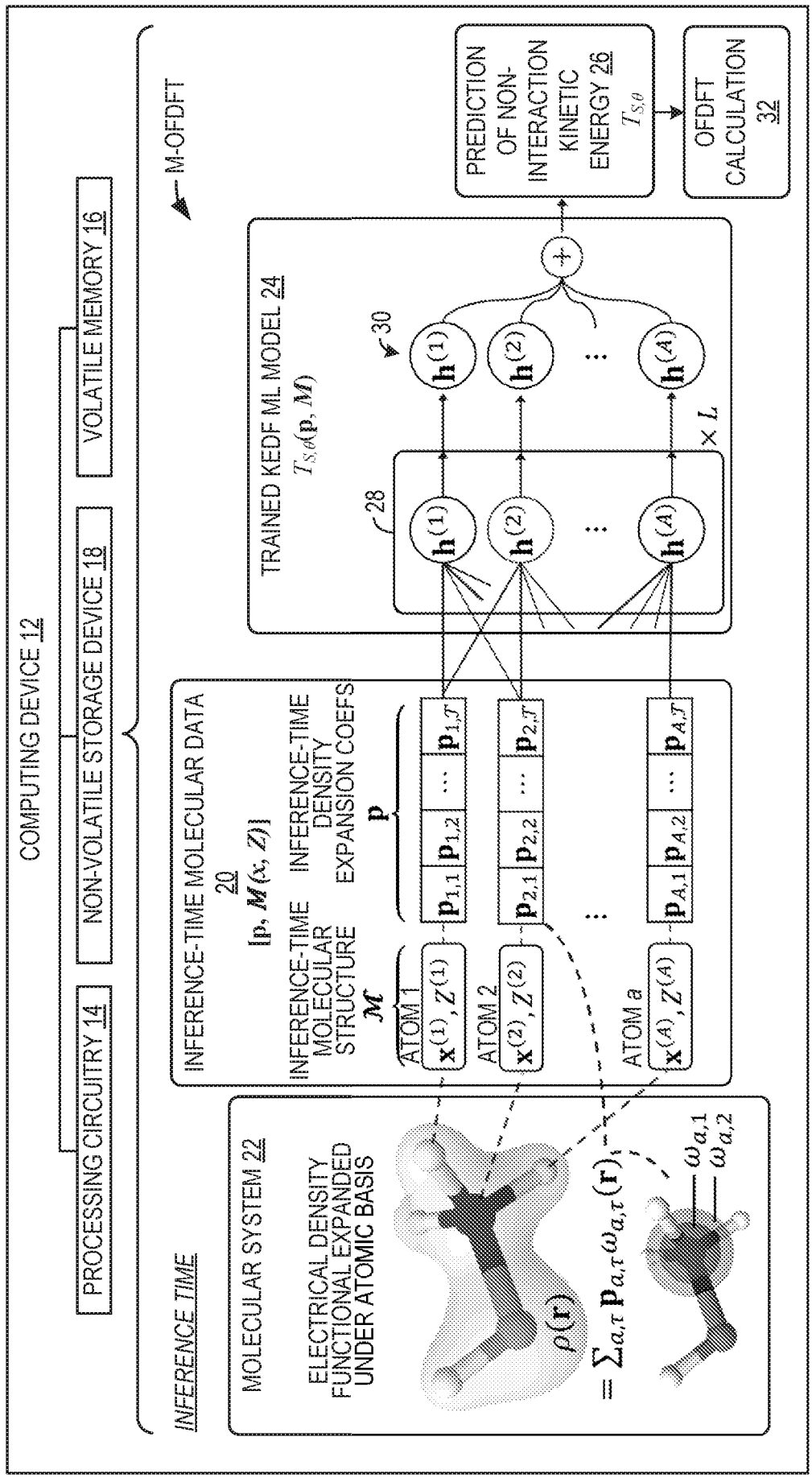

10
COMPUTING DEVICE 12
PROCESSING CIRCUITRY 14
NON-VOLATILE STORAGE DEVICE 18
VOLATILE MEMORY 16
M-OFDFT
INFERENCE TIME
MOLECULAR SYSTEM 22
ELECTRICAL DENSITY FUNCTIONAL EXPANDED UNDER ATOMIC BASIS
$\rho(\mathbf{r}) = \sum_{a,\tau} \mathbf{p}_{a,\tau} \omega_{a,\tau}(\mathbf{r})$
$\omega_{a,1}$
$\omega_{a,2}$
INFERENCE-TIME MOLECULAR DATA 20
$\{\mathbf{p}, \mathcal{M}(x, Z)\}$
INFERENCE-TIME MOLECULAR STRUCTURE $\mathcal{M}$
INFERENCE-TIME DENSITY EXPANSION COEFS $\mathbf{p}$
ATOM 1
ATOM 2
ATOM a
$\mathbf{x}^{(1)}, Z^{(1)}$
$\mathbf{x}^{(2)}, Z^{(2)}$
$\mathbf{x}^{(A)}, Z^{(A)}$
$\mathbf{p}_{1,1}$ $\mathbf{p}_{1,2}$ ⋯ $\mathbf{p}_{1,T}$
$\mathbf{p}_{2,1}$ $\mathbf{p}_{2,2}$ ⋯ $\mathbf{p}_{2,T}$
$\mathbf{p}_{A,1}$ $\mathbf{p}_{A,2}$ ⋯ $\mathbf{p}_{A,T}$
TRAINED KEDF ML MODEL 24
$T_{S\theta}(\mathbf{p}, \mathcal{M})$
28
30
$\mathbf{h}^{(1)}$ $\mathbf{h}^{(2)}$ ⋯ $\mathbf{h}^{(A)}$
× L
PREDICTION OF NON-INTERACTION KINETIC ENERGY 26
$T_{S\theta}$
OFDFT CALCULATION 32

FIG. 1B

FINDING GROUND-STATE ENERGY AND DENSITY FOR A MOLECULAR STRUCTURE USING A TRAINED KEDF MODEL
1
KEDF MODEL IS TRAINED USING DENSITY COEFFICIENTS AND AND KINETIC ENERGY AND GRADIENT COMPUTED FROM EACH SCF CYCLE.
SOLUTION SURFACE
ENERGY GRADIENT
DENSITY COEFFICIENTS
$\mathbf{p}^{(k)}$
$\nabla_{\mathbf{p}} E_\theta(\mathbf{p}^{(k)}, \mathcal{M})$
$\mathbf{p}^{(k+1)}$
2
TRAINED KEDF MODEL CAN BE USED TO ESTIMATE NON-INTERACTING KINETIC ENERGY AND GRADIENT AT INFERENCE TIME, AND GROUND-STATE ENERGY CAN BE COMPUTED
$E_\theta(\mathbf{p}, \mathcal{M}) :=$
$T_{S,\theta}(\mathbf{p}, \mathcal{M}) + E_{\mathrm{H}}(\mathbf{p}, \mathcal{M}) + E_{\mathrm{XC}}(\mathbf{p}, \mathcal{M}) + E_{\mathrm{ext}}(\mathbf{p}, \mathcal{M})$
PREDICTIVE TASK OF KEDF MODEL: PREDICT NON-INTERACTING KINETIC ENERGY GIVEN ($\mathbf{p}$, $\mathcal{M}$)

FIG. 1C

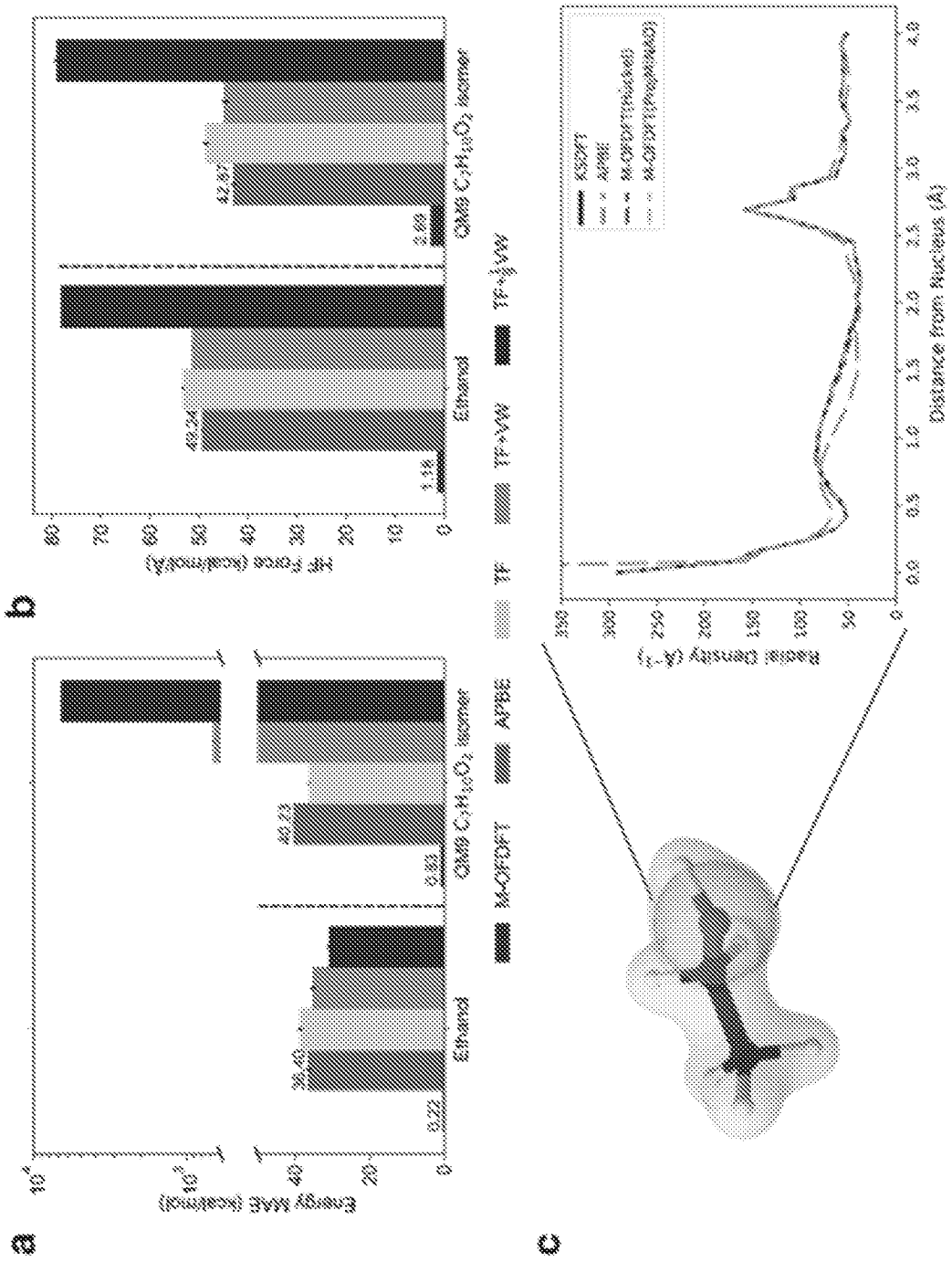

a
b
c
Energy MAE (kcal/mol)
HF force (kcal/mol/Å)
Ethanol
QM9 $C_7H_{10}O_2$ isomer
M-OFDFT
APBE
TF
TF-vW
TF-λvW
KSDFT
Radial Density (Å^{-3})
Distance from Nucleus (Å)

FIG. 2

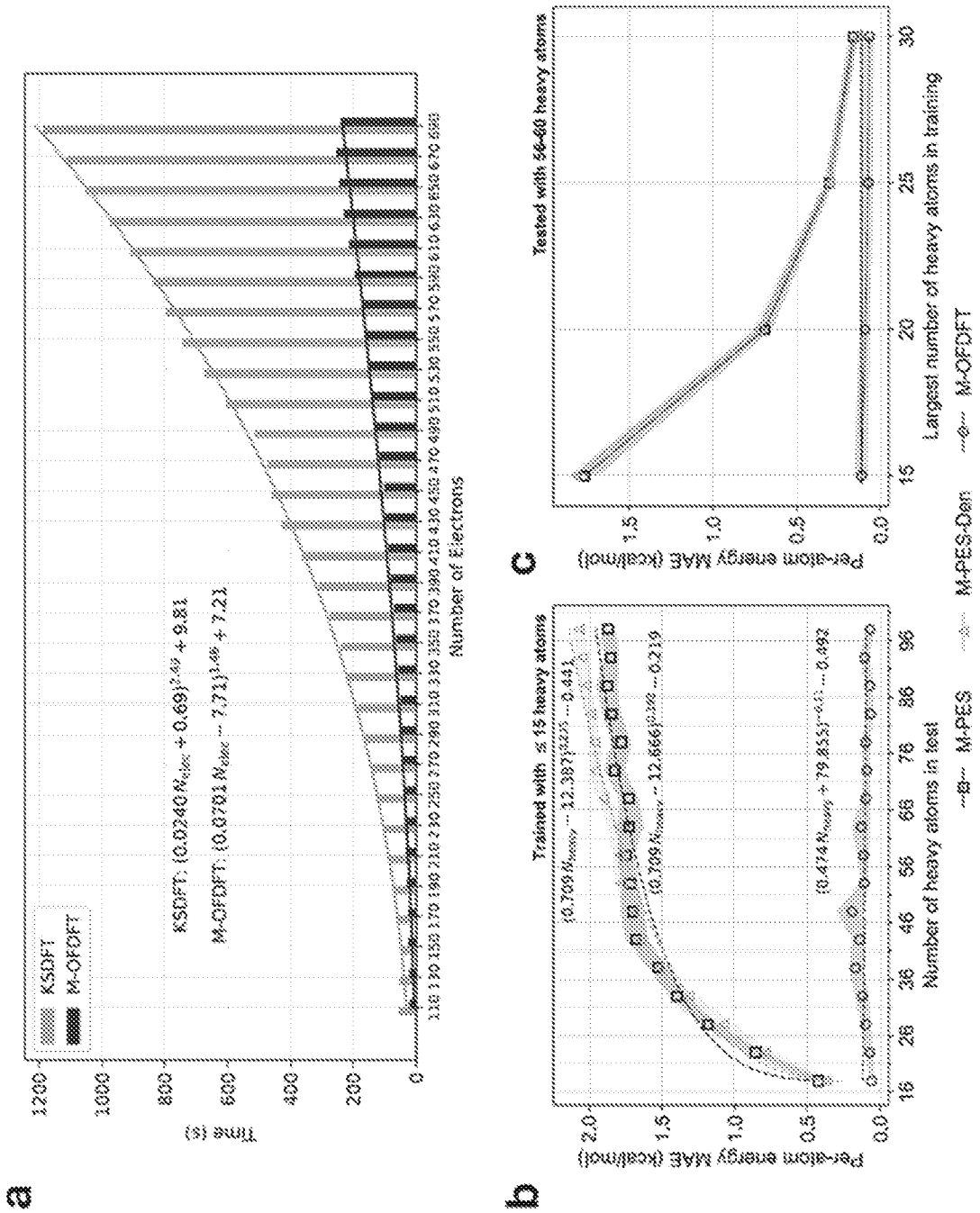

a
KSDFT
M-OFDFT
KSDFT: $(0.0240\,N_{elec} + 0.69)^{2.49} + 9.81$
M-OFDFT: $(0.0701\,N_{elec} - 7.71)^{1.46} + 7.21$
Time (s)
Number of Electrons
b
Trained with ≤ 15 heavy atoms
$(0.709\,N_{heavy} - 12.387)^{0.25} - 0.441$
$(0.709\,N_{heavy} - 12.666)^{0.25} - 0.219$
$(8.474\,N_{heavy} + 79.853)^{-0.5} - 0.492$
Per-atom energy MAE (kcal/mol)
Number of heavy atoms in test
M-PES
M-PES-Den
c
Tested with 56-60 heavy atoms
Per-atom energy MAE (kcal/mol)
Largest number of heavy atoms in training
M-OFDFT

FIG. 3

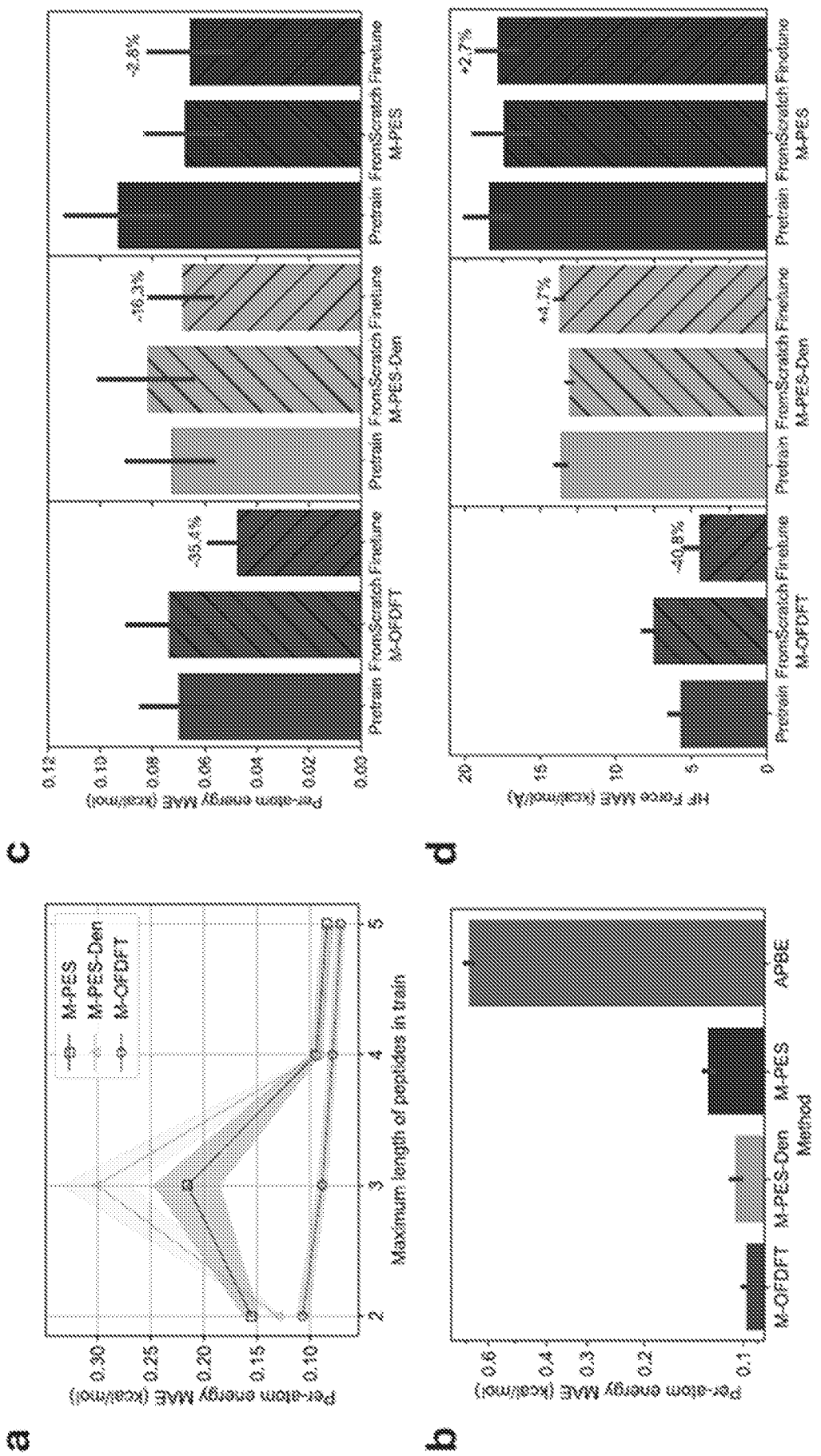

a
Per-atom energy MAE (kcal/mol)
Maximum length of peptides in train
M-PES
M-PES-Den
M-OFDFT
b
Per-atom energy MAE (kcal/mol)
M-OFDFT
M-PES-Den
M-PES
APBE
Method
c
Per-atom energy MAE (kcal/mol)
-35.4%
-16.3%
-9.6%
Pretrain FromScratch Finetune
M-OFDFT
M-PES-Den
M-PES
d
HF Force MAE (kcal/mol/Å)
-40.8%
+4.7%
+2.7%
Pretrain FromScratch Finetune
M-OFDFT
M-PES-Den
M-PES

FIG. 4

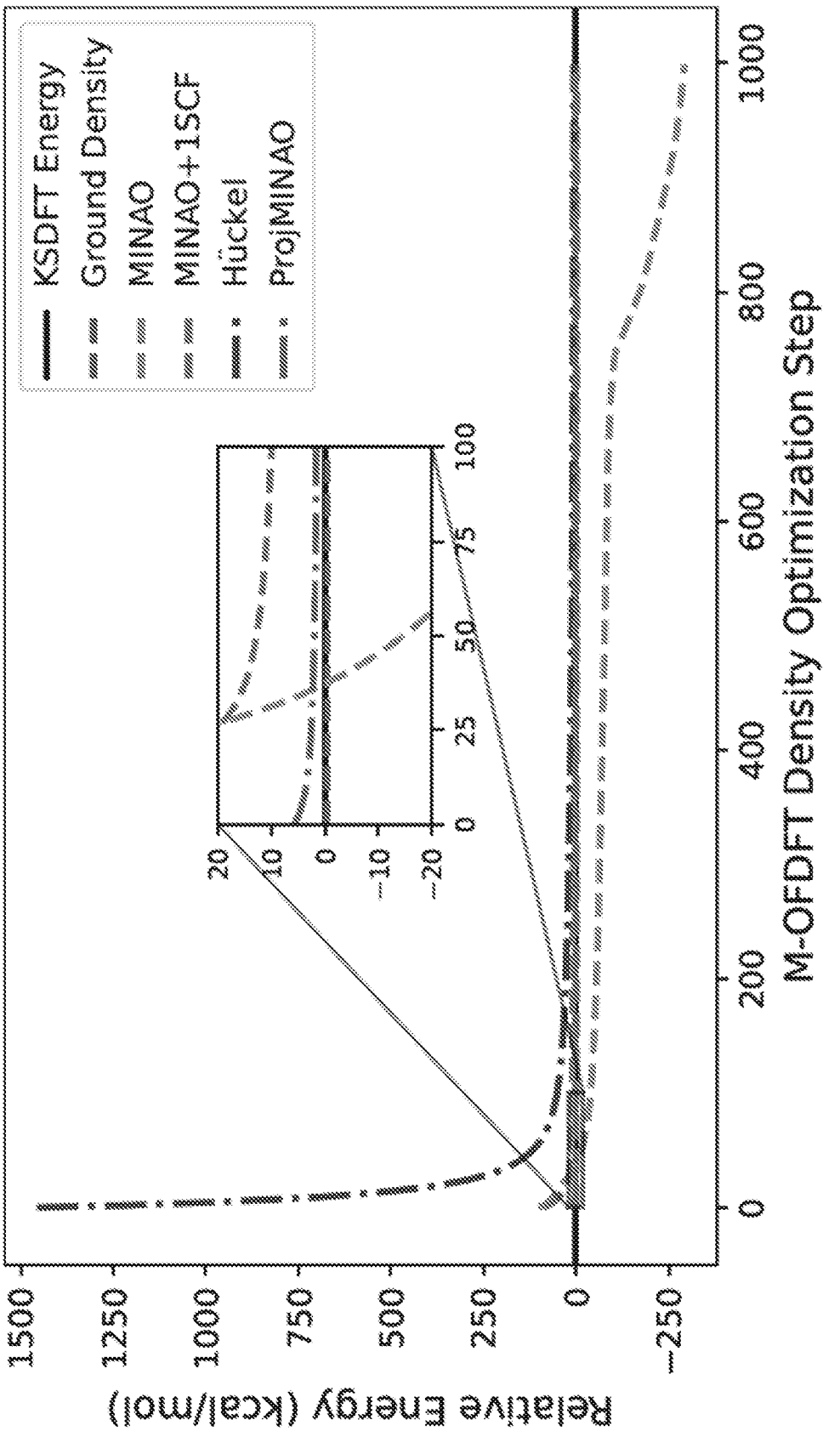

KSDFT Energy
Ground Density
MINAO
MINAO+1SCF
Hückel
ProjMINAO
Relative Energy (kcal/mol)
M-OFDFT Density Optimization Step
1500
1250
1000
750
500
250
0
−250
0
200
400
600
800
1000
20
10
0
−10
−20
0
25
50
75
100

FIG. 5

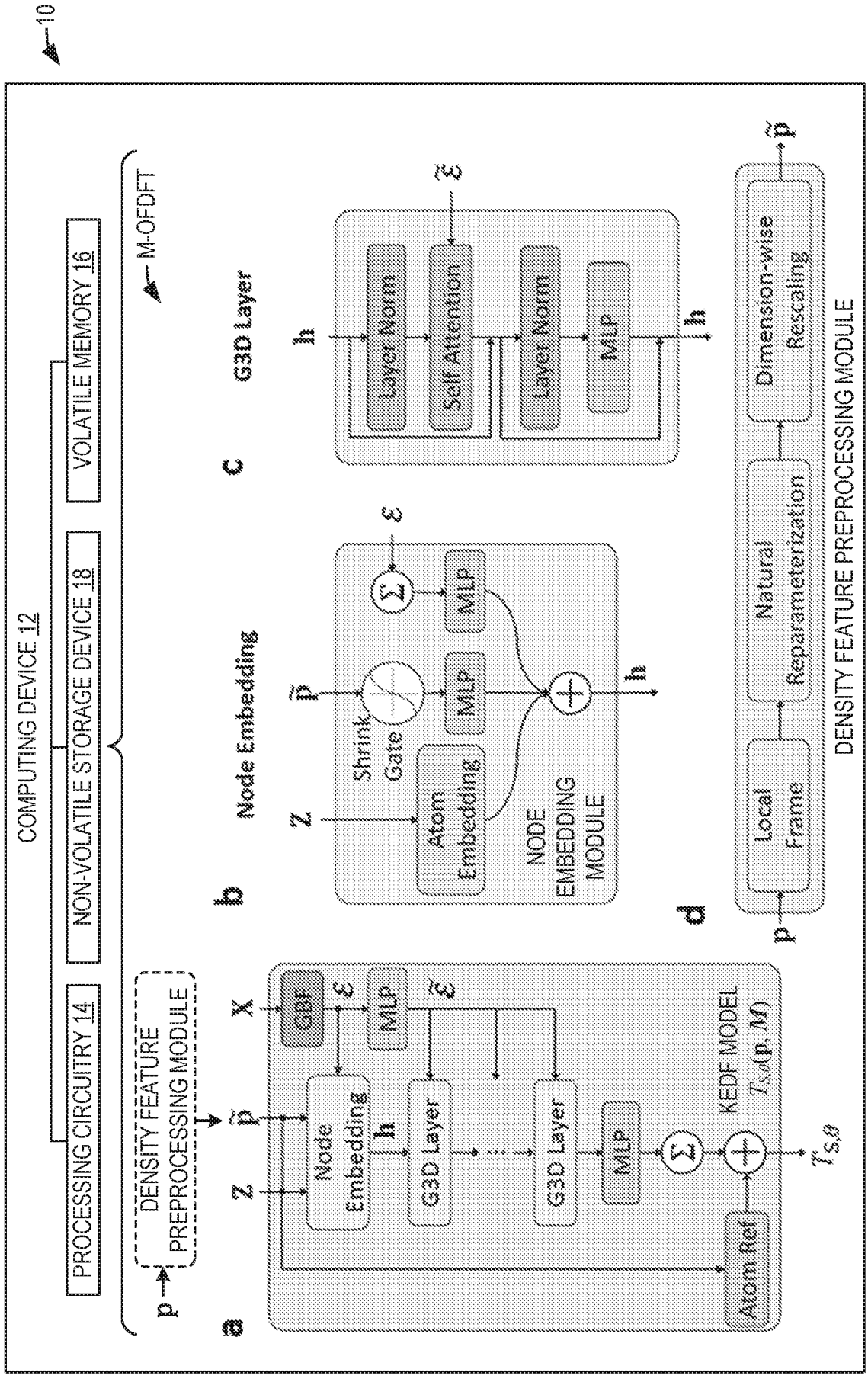

10
COMPUTING DEVICE 12
PROCESSING CIRCUITRY 14
NON-VOLATILE STORAGE DEVICE 18
VOLATILE MEMORY 16
M-OFDFT
DENSITY FEATURE PREPROCESSING MODULE
p
a
Z
p̃
X
GBF
Node Embedding
h
G3D Layer
G3D Layer
MLP
MLP
Atom Ref
KEDF MODEL
$T_{S,\theta}(\mathbf{p}, M)$
$T_{S,\theta}$
b
Node Embedding
Z
p̃
Shrink Gate
Atom Embedding
MLP
MLP
NODE EMBEDDING MODULE
h
c
G3D Layer
h
Layer Norm
Self Attention
Layer Norm
MLP
h
d
Local Frame
Natural Reparameterization
Dimension-wise Rescaling
p̃
DENSITY FEATURE PREPROCESSING MODULE

FIG. 6

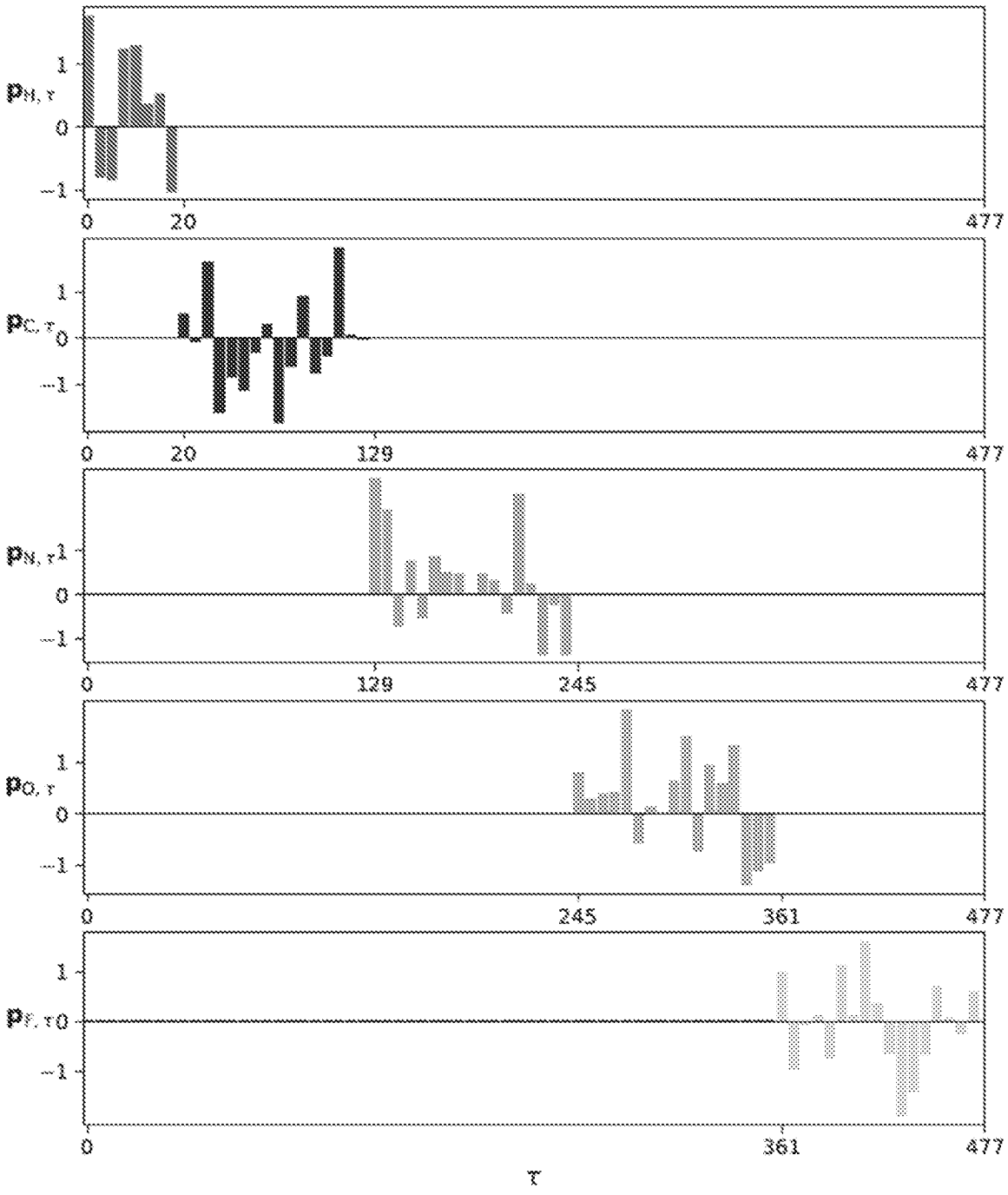

$p_{N,\tau}$
1
0
−1
0
20
477
$p_{C,\tau}$
1
0
−1
0
20
129
477
$p_{N,\tau}$
1
0
−1
0
129
245
477
$p_{O,\tau}$
1
0
−1
0
245
361
477
$p_{F,\tau}$
1
0
−1
0
361
477
τ

FIG. 7

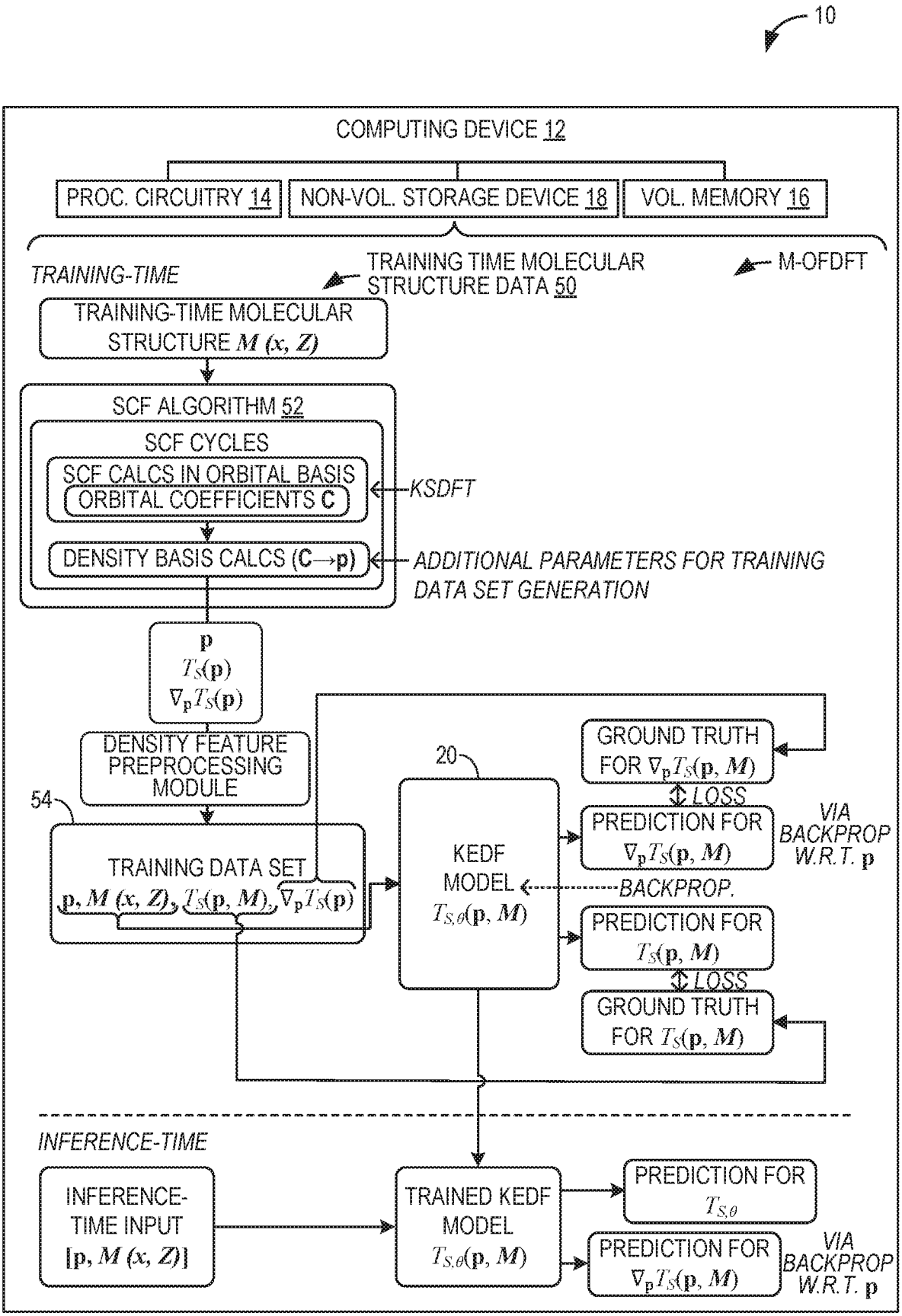

10
COMPUTING DEVICE 12
PROC. CIRCUITRY 14
NON-VOL. STORAGE DEVICE 18
VOL. MEMORY 16
TRAINING-TIME
TRAINING TIME MOLECULAR STRUCTURE DATA 50
M-OFDFT
TRAINING-TIME MOLECULAR STRUCTURE M (x, Z)
SCF ALGORITHM 52
SCF CYCLES
SCF CALCS IN ORBITAL BASIS
ORBITAL COEFFICIENTS C
KSDFT
DENSITY BASIS CALCS (C→p)
ADDITIONAL PARAMETERS FOR TRAINING DATA SET GENERATION
p
T_S(p)
$\nabla_p T_S$(p)
DENSITY FEATURE PREPROCESSING MODULE
54
TRAINING DATA SET
p, M (x, Z), T_S(p, M), $\nabla_p T_S$(p)
20
KEDF MODEL
$T_{S,\theta}$(p, M)
GROUND TRUTH FOR $\nabla_p T_S$(p, M)
LOSS
PREDICTION FOR $\nabla_p T_S$(p, M)
VIA BACKPROP W.R.T. p
BACKPROP.
PREDICTION FOR T_S(p, M)
LOSS
GROUND TRUTH FOR T_S(p, M)
INFERENCE-TIME
INFERENCE-TIME INPUT [p, M (x, Z)]
TRAINED KEDF MODEL
$T_{S,\theta}$(p, M)
PREDICTION FOR $T_{S,\theta}$
PREDICTION FOR $\nabla_p T_S$(p, M)
VIA BACKPROP W.R.T. p

FIG. 8

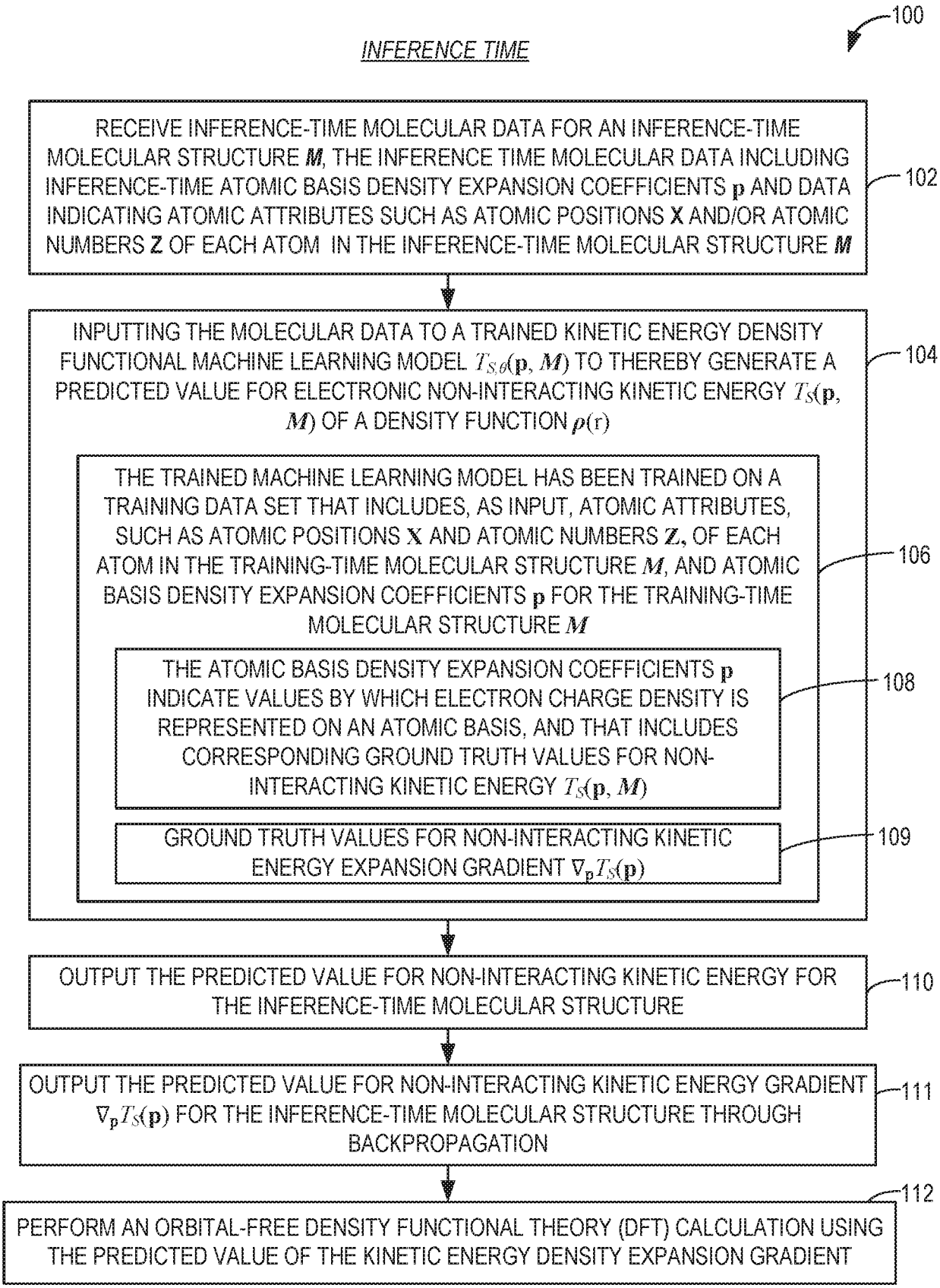

100
INFERENCE TIME
RECEIVE INFERENCE-TIME MOLECULAR DATA FOR AN INFERENCE-TIME MOLECULAR STRUCTURE $\boldsymbol{M}$, THE INFERENCE TIME MOLECULAR DATA INCLUDING INFERENCE-TIME ATOMIC BASIS DENSITY EXPANSION COEFFICIENTS $\mathbf{p}$ AND DATA INDICATING ATOMIC ATTRIBUTES SUCH AS ATOMIC POSITIONS $\mathbf{X}$ AND/OR ATOMIC NUMBERS $\mathbf{Z}$ OF EACH ATOM IN THE INFERENCE-TIME MOLECULAR STRUCTURE $\boldsymbol{M}$
102
INPUTTING THE MOLECULAR DATA TO A TRAINED KINETIC ENERGY DENSITY FUNCTIONAL MACHINE LEARNING MODEL $T_{S,\theta}(\mathbf{p}, \boldsymbol{M})$ TO THEREBY GENERATE A PREDICTED VALUE FOR ELECTRONIC NON-INTERACTING KINETIC ENERGY $T_S(\mathbf{p}, \boldsymbol{M})$ OF A DENSITY FUNCTION $\rho(\mathrm{r})$
104
THE TRAINED MACHINE LEARNING MODEL HAS BEEN TRAINED ON A TRAINING DATA SET THAT INCLUDES, AS INPUT, ATOMIC ATTRIBUTES, SUCH AS ATOMIC POSITIONS $\mathbf{X}$ AND ATOMIC NUMBERS $\mathbf{Z}$, OF EACH ATOM IN THE TRAINING-TIME MOLECULAR STRUCTURE $\boldsymbol{M}$, AND ATOMIC BASIS DENSITY EXPANSION COEFFICIENTS $\mathbf{p}$ FOR THE TRAINING-TIME MOLECULAR STRUCTURE $\boldsymbol{M}$
106
THE ATOMIC BASIS DENSITY EXPANSION COEFFICIENTS $\mathbf{p}$ INDICATE VALUES BY WHICH ELECTRON CHARGE DENSITY IS REPRESENTED ON AN ATOMIC BASIS, AND THAT INCLUDES CORRESPONDING GROUND TRUTH VALUES FOR NON-INTERACTING KINETIC ENERGY $T_S(\mathbf{p}, \boldsymbol{M})$
108
GROUND TRUTH VALUES FOR NON-INTERACTING KINETIC ENERGY EXPANSION GRADIENT $\nabla_{\mathbf{p}} T_S(\mathbf{p})$
109
OUTPUT THE PREDICTED VALUE FOR NON-INTERACTING KINETIC ENERGY FOR THE INFERENCE-TIME MOLECULAR STRUCTURE
110
OUTPUT THE PREDICTED VALUE FOR NON-INTERACTING KINETIC ENERGY GRADIENT $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ FOR THE INFERENCE-TIME MOLECULAR STRUCTURE THROUGH BACKPROPAGATION
111
PERFORM AN ORBITAL-FREE DENSITY FUNCTIONAL THEORY (DFT) CALCULATION USING THE PREDICTED VALUE OF THE KINETIC ENERGY DENSITY EXPANSION GRADIENT
112

FIG. 9A

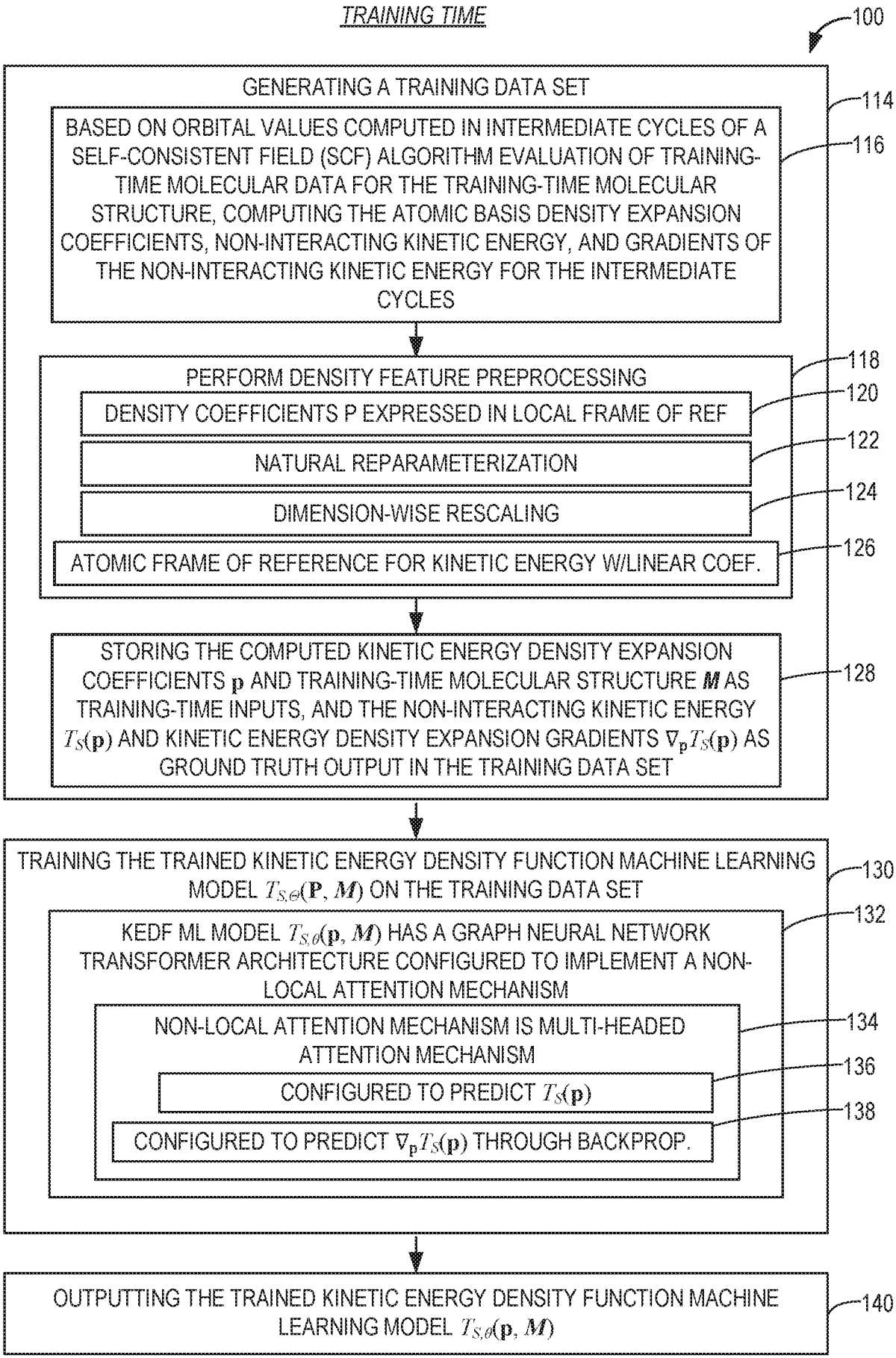

TRAINING TIME
100
GENERATING A TRAINING DATA SET
114
BASED ON ORBITAL VALUES COMPUTED IN INTERMEDIATE CYCLES OF A SELF-CONSISTENT FIELD (SCF) ALGORITHM EVALUATION OF TRAINING-TIME MOLECULAR DATA FOR THE TRAINING-TIME MOLECULAR STRUCTURE, COMPUTING THE ATOMIC BASIS DENSITY EXPANSION COEFFICIENTS, NON-INTERACTING KINETIC ENERGY, AND GRADIENTS OF THE NON-INTERACTING KINETIC ENERGY FOR THE INTERMEDIATE CYCLES
116
PERFORM DENSITY FEATURE PREPROCESSING
118
DENSITY COEFFICIENTS P EXPRESSED IN LOCAL FRAME OF REF
120
NATURAL REPARAMETERIZATION
122
DIMENSION-WISE RESCALING
124
ATOMIC FRAME OF REFERENCE FOR KINETIC ENERGY W/LINEAR COEF.
126
STORING THE COMPUTED KINETIC ENERGY DENSITY EXPANSION COEFFICIENTS $\mathbf{p}$ AND TRAINING-TIME MOLECULAR STRUCTURE $\boldsymbol{M}$ AS TRAINING-TIME INPUTS, AND THE NON-INTERACTING KINETIC ENERGY $T_S(\mathbf{p})$ AND KINETIC ENERGY DENSITY EXPANSION GRADIENTS $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ AS GROUND TRUTH OUTPUT IN THE TRAINING DATA SET
128
TRAINING THE TRAINED KINETIC ENERGY DENSITY FUNCTION MACHINE LEARNING MODEL $T_{S,\theta}(\mathbf{P}, \boldsymbol{M})$ ON THE TRAINING DATA SET
130
KEDF ML MODEL $T_{S,\theta}(\mathbf{p}, \boldsymbol{M})$ HAS A GRAPH NEURAL NETWORK TRANSFORMER ARCHITECTURE CONFIGURED TO IMPLEMENT A NON-LOCAL ATTENTION MECHANISM
132
NON-LOCAL ATTENTION MECHANISM IS MULTI-HEADED ATTENTION MECHANISM
134
CONFIGURED TO PREDICT $T_S(\mathbf{p})$
136
CONFIGURED TO PREDICT $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ THROUGH BACKPROP.
138
OUTPUTTING THE TRAINED KINETIC ENERGY DENSITY FUNCTION MACHINE LEARNING MODEL $T_{S,\theta}(\mathbf{p}, \boldsymbol{M})$
140

FIG. 9B

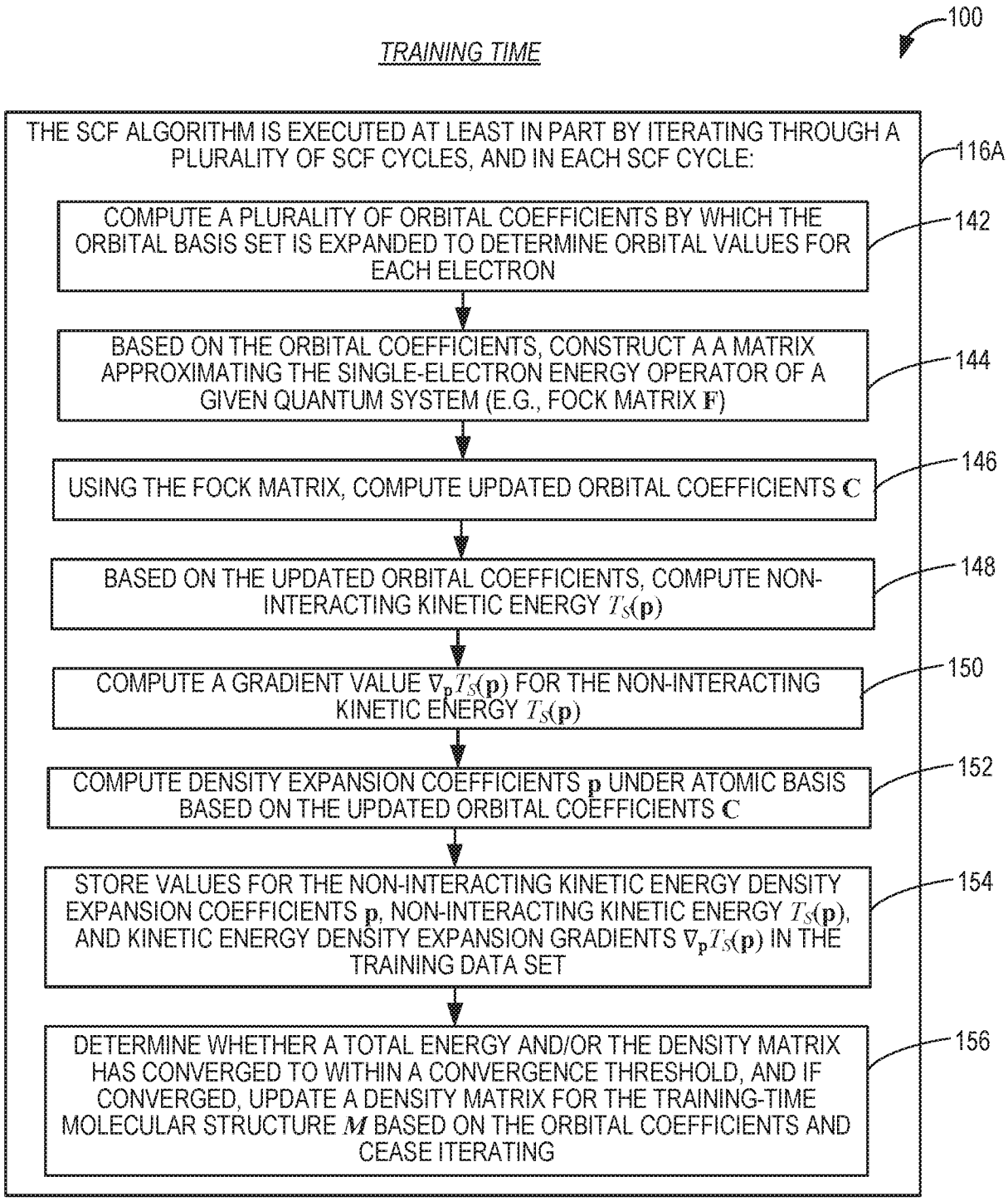

100
TRAINING TIME
THE SCF ALGORITHM IS EXECUTED AT LEAST IN PART BY ITERATING THROUGH A PLURALITY OF SCF CYCLES, AND IN EACH SCF CYCLE:
116A
COMPUTE A PLURALITY OF ORBITAL COEFFICIENTS BY WHICH THE ORBITAL BASIS SET IS EXPANDED TO DETERMINE ORBITAL VALUES FOR EACH ELECTRON
142
BASED ON THE ORBITAL COEFFICIENTS, CONSTRUCT A A MATRIX APPROXIMATING THE SINGLE-ELECTRON ENERGY OPERATOR OF A GIVEN QUANTUM SYSTEM (E.G., FOCK MATRIX $\mathbf{F}$)
144
USING THE FOCK MATRIX, COMPUTE UPDATED ORBITAL COEFFICIENTS $\mathbf{C}$
146
BASED ON THE UPDATED ORBITAL COEFFICIENTS, COMPUTE NON-INTERACTING KINETIC ENERGY $T_S(\mathbf{p})$
148
COMPUTE A GRADIENT VALUE $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ FOR THE NON-INTERACTING KINETIC ENERGY $T_S(\mathbf{p})$
150
COMPUTE DENSITY EXPANSION COEFFICIENTS $\mathbf{p}$ UNDER ATOMIC BASIS BASED ON THE UPDATED ORBITAL COEFFICIENTS $\mathbf{C}$
152
STORE VALUES FOR THE NON-INTERACTING KINETIC ENERGY DENSITY EXPANSION COEFFICIENTS $\mathbf{p}$, NON-INTERACTING KINETIC ENERGY $T_S(\mathbf{p})$, AND KINETIC ENERGY DENSITY EXPANSION GRADIENTS $\nabla_{\mathbf{p}} T_S(\mathbf{p})$ IN THE TRAINING DATA SET
154
DETERMINE WHETHER A TOTAL ENERGY AND/OR THE DENSITY MATRIX HAS CONVERGED TO WITHIN A CONVERGENCE THRESHOLD, AND IF CONVERGED, UPDATE A DENSITY MATRIX FOR THE TRAINING-TIME MOLECULAR STRUCTURE M BASED ON THE ORBITAL COEFFICIENTS AND CEASE ITERATING
156

FIG. 9C

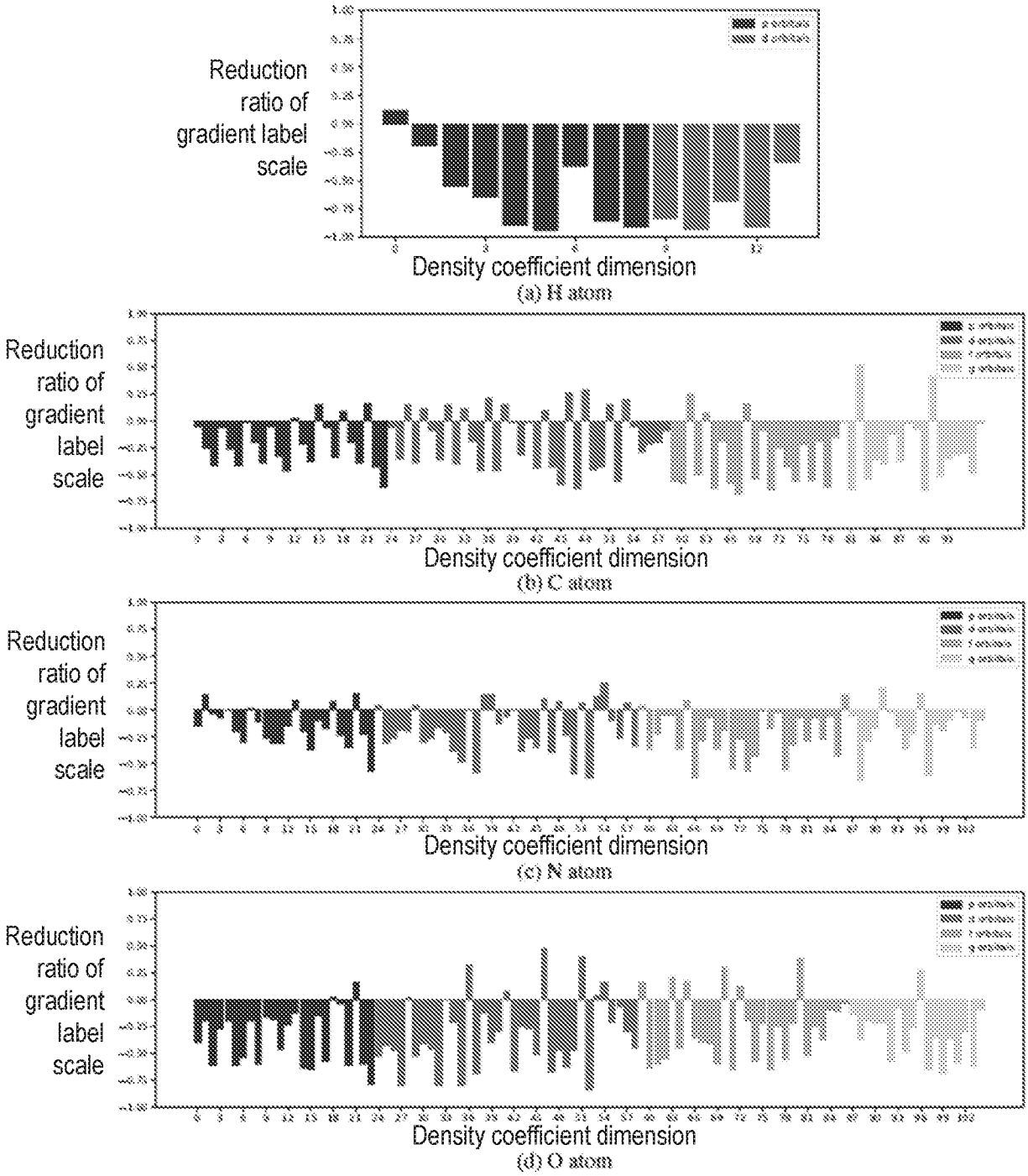

Reduction ratio of gradient label scale
Density coefficient dimension
(a) H atom
Reduction ratio of gradient label scale
Density coefficient dimension
(b) C atom
Reduction ratio of gradient label scale
Density coefficient dimension
(c) N atom
Reduction ratio of gradient label scale
Density coefficient dimension
(d) O atom

FIG. 10

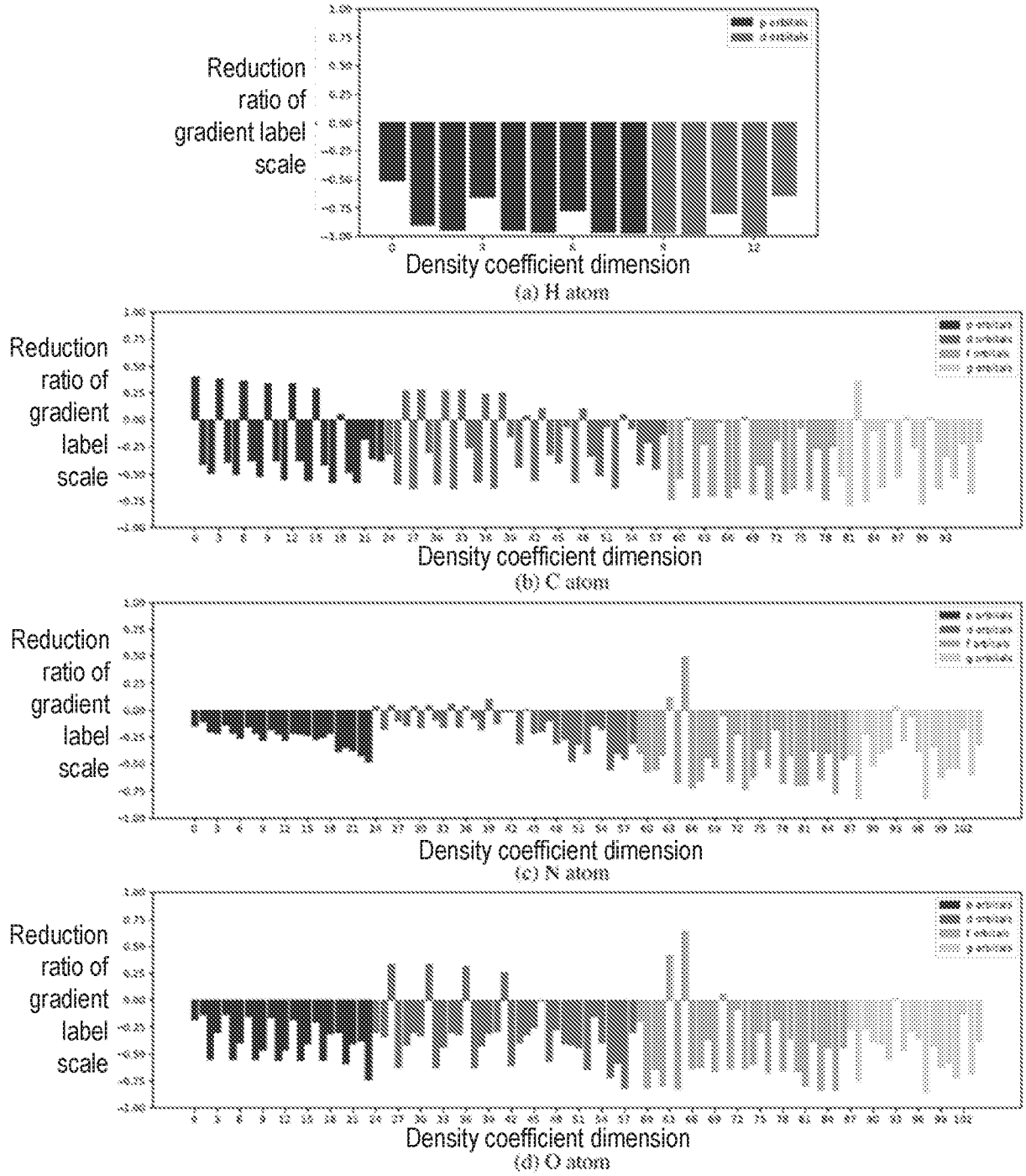
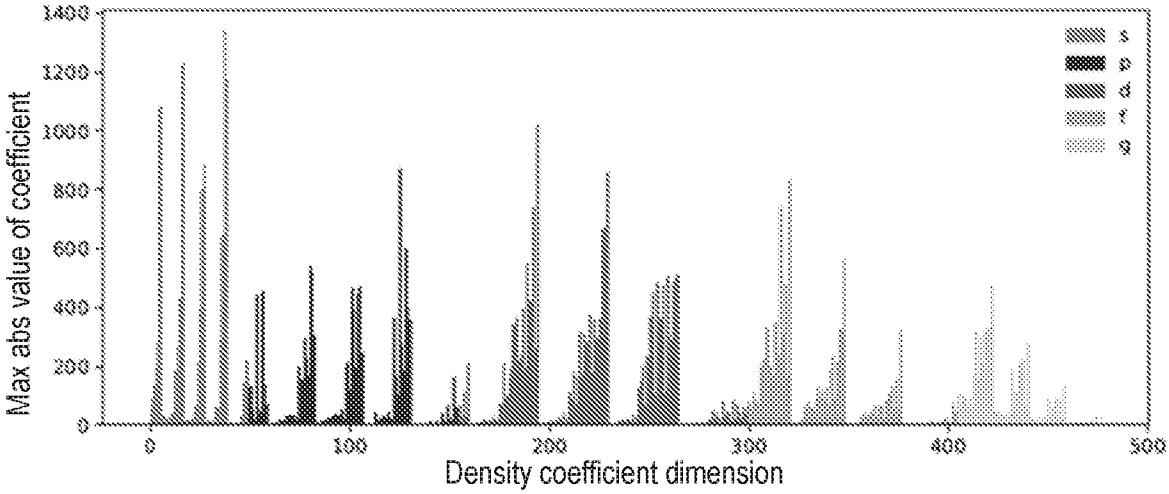
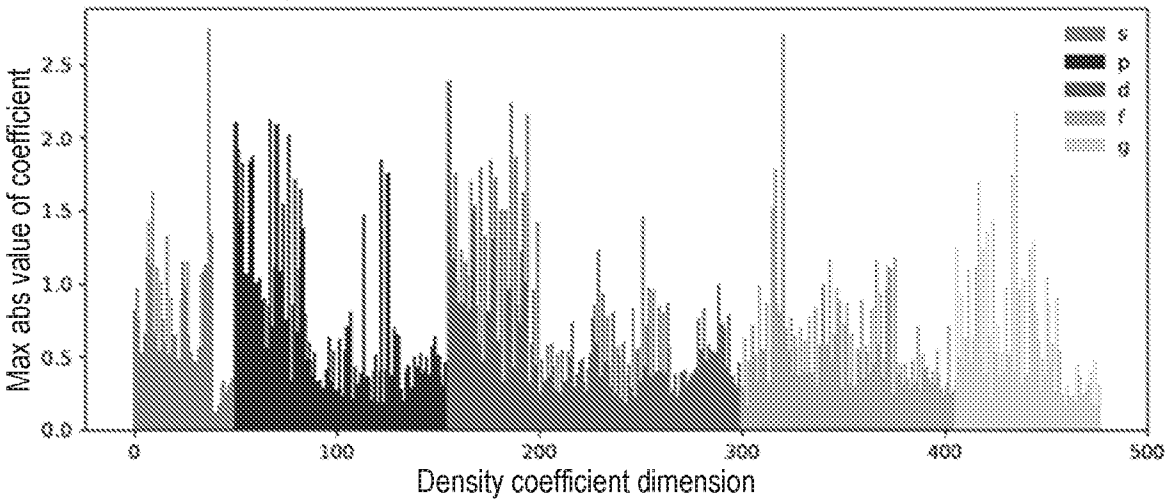

Reduction ratio of gradient label scale
Density coefficient dimension
(a) H atom
Reduction ratio of gradient label scale
Density coefficient dimension
(b) C atom
Reduction ratio of gradient label scale
Density coefficient dimension
(c) N atom
Reduction ratio of gradient label scale
Density coefficient dimension
(d) O atom

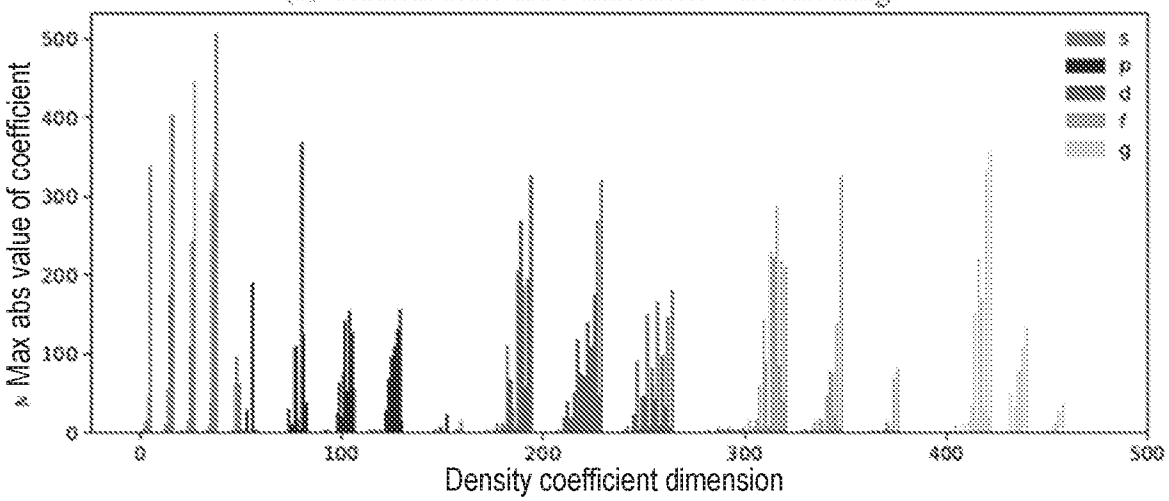
FIG. 11



(a) Gradient scale of different coefficient dimensions.



(b) Gradient scale after dimension-wise rescaling.



(c) Density coefficients scale after dimension-wise rescaling.

FIG. 12

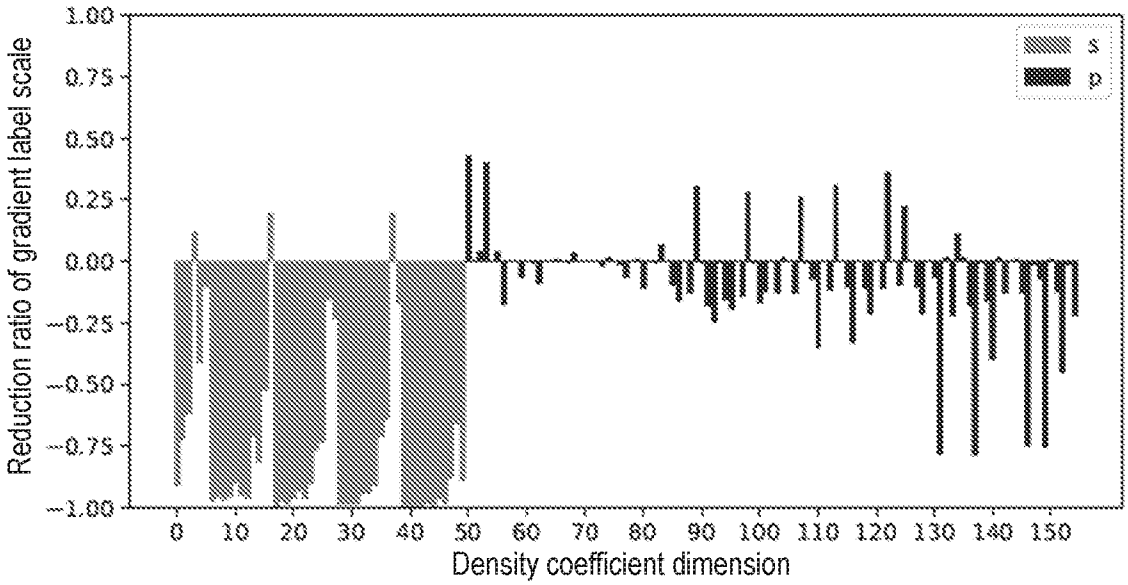
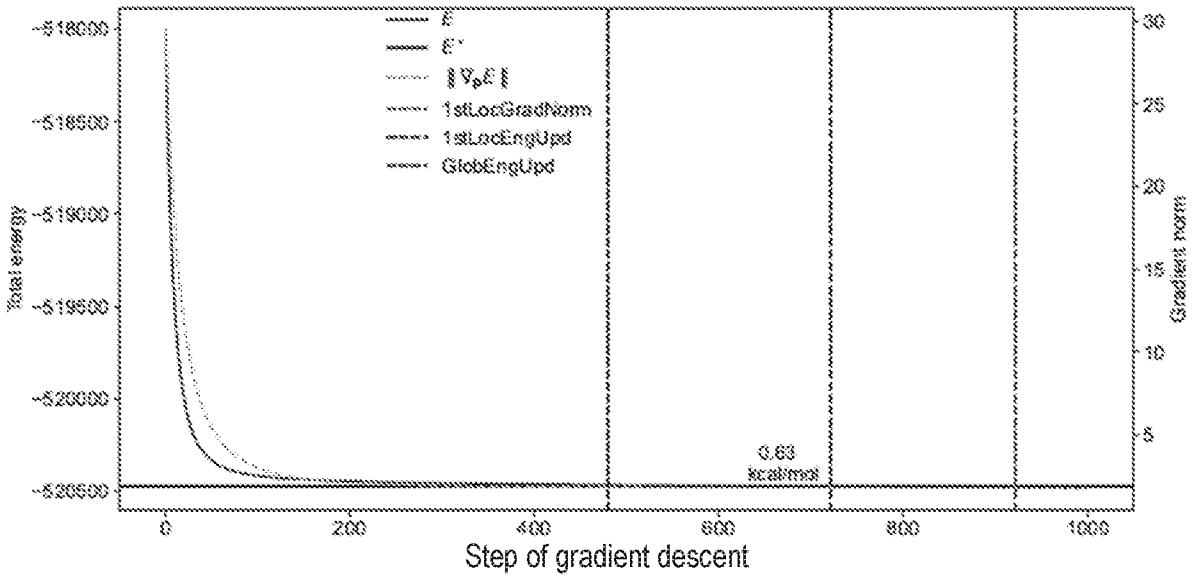
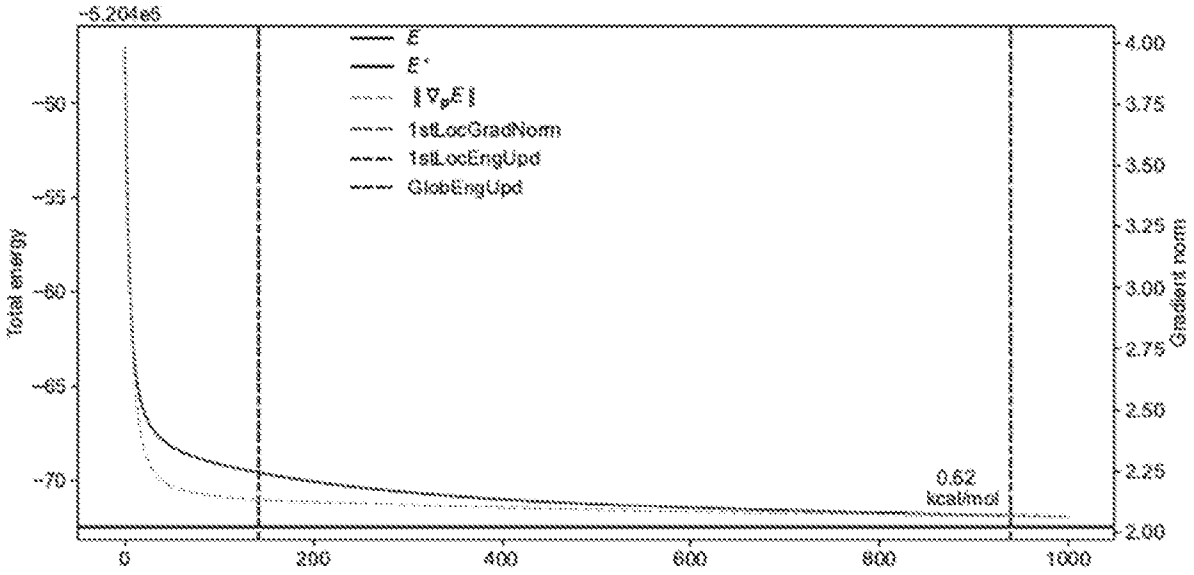

Reduction ratio of gradient label scale
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
Density coefficient dimension
s
p

FIG. 13



(a) Hückel initialization.



(b) ProjMINAO initialization.

FIG. 14

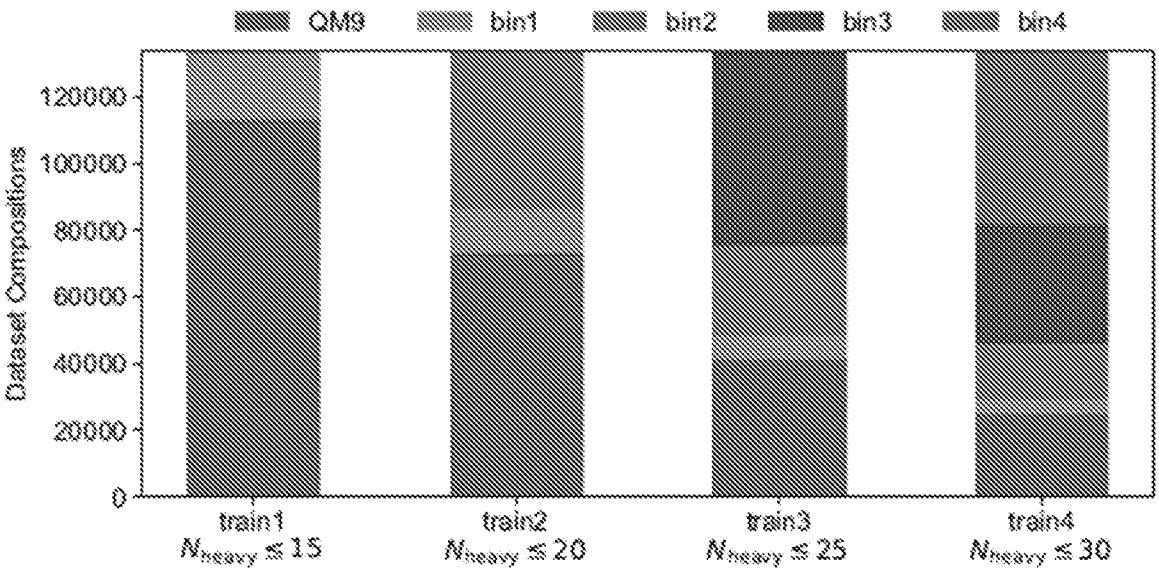

QM9
bin1
bin2
bin3
bin4
Dataset Compositions
120000
100000
80000
60000
40000
20000
0
train1
$N_{heavy} \leq 15$
train2
$N_{heavy} \leq 20$
train3
$N_{heavy} \leq 25$
train4
$N_{heavy} \leq 30$

FIG. 15

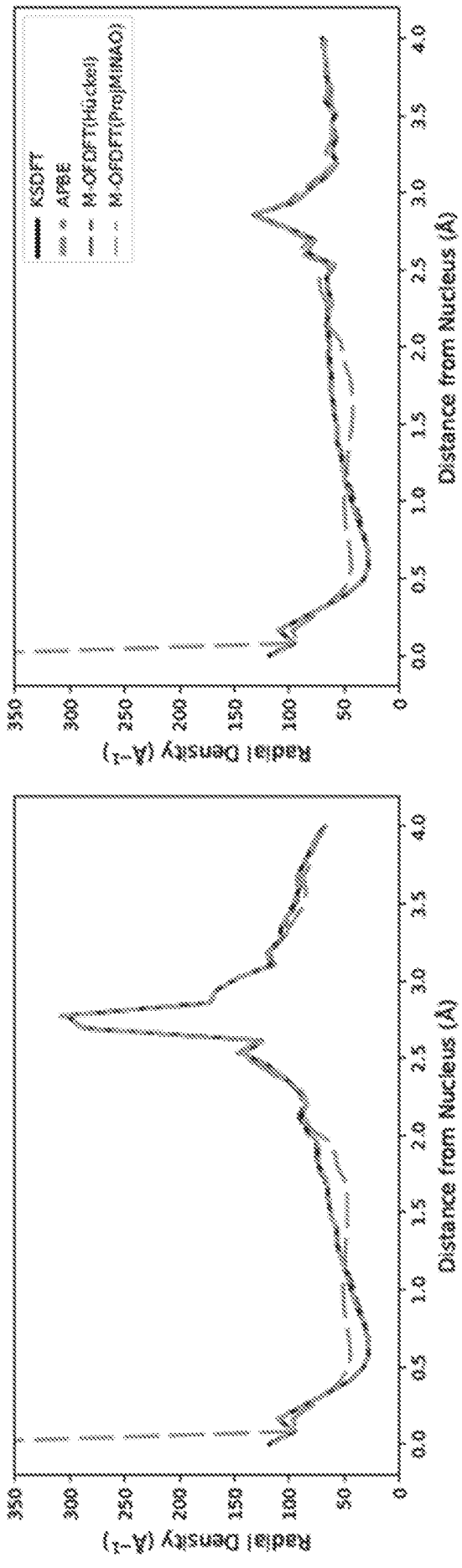

KSDFT
APBE
M-OFDFT(Hackel)
M-OFDFT(PrajapatiMAO)
Radial Density (Å^{-1})
Distance from Nucleus (Å)
(a) The α-carbon
(b) The β-carbon

FIG. 16

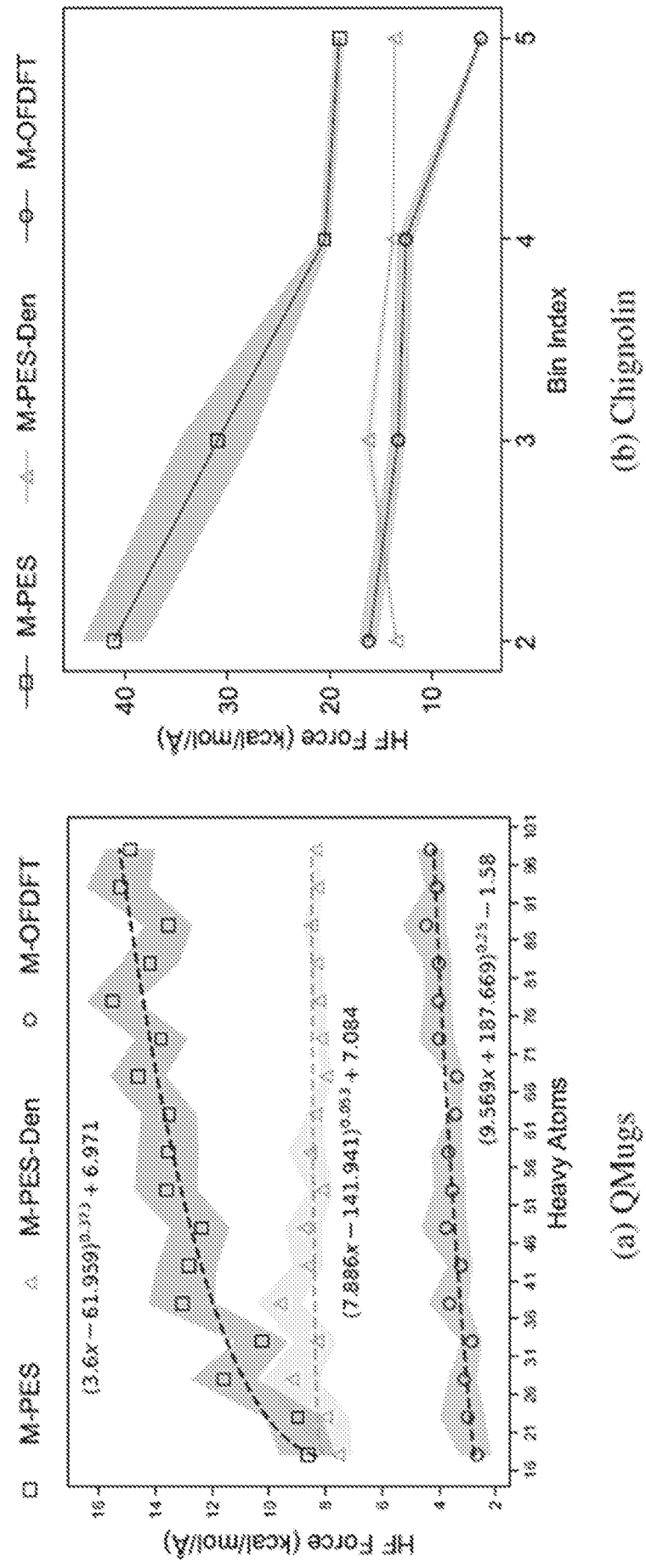

M-PES
M-PES-Den
M-OFDFT
HF Force (kcal/mol/Å)
Heavy Atoms
(3.6x − 61.959)^0.333 + 6.971
(7.886x − 141.941)^0.333 + 7.084
(9.569x + 187.669)^0.333 − 1.58
(a) QMugs
HF Force (kcal/mol/Å)
Bin Index
(b) Chignolin

FIG. 17

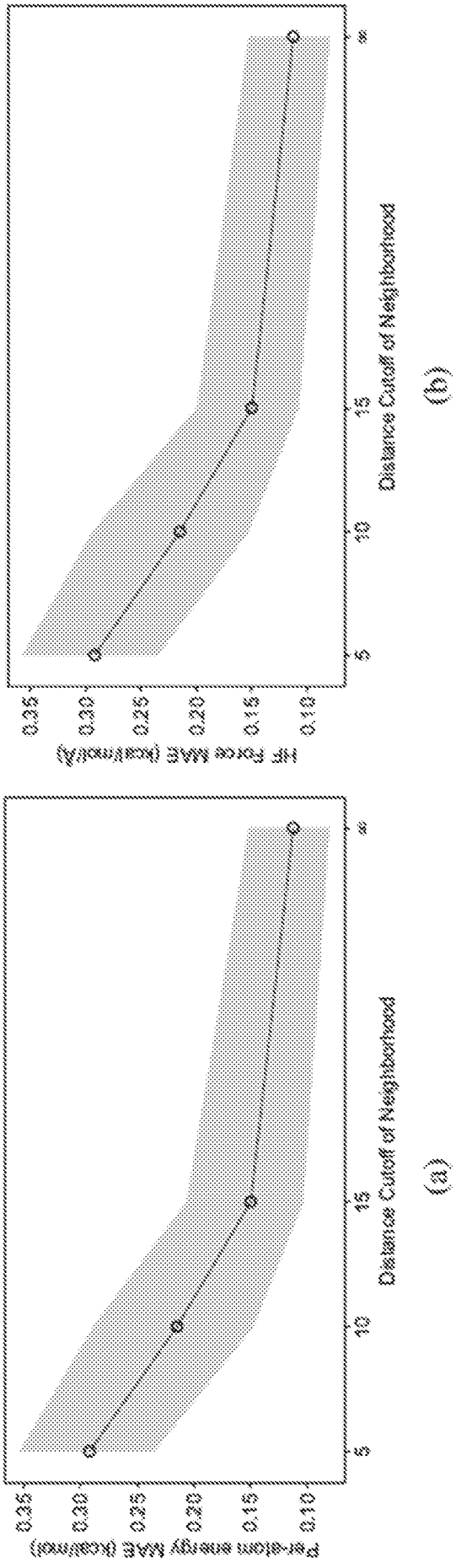

Per-atom energy MAE (kcal/mol)
0.35
0.30
0.25
0.20
0.15
0.10
5
10
15
∞
Distance Cutoff of Neighborhood
(a)
HF Force MAE (kcal/mol/Å)
0.35
0.30
0.25
0.20
0.15
0.10
5
10
15
∞
Distance Cutoff of Neighborhood
(b)

FIG. 18

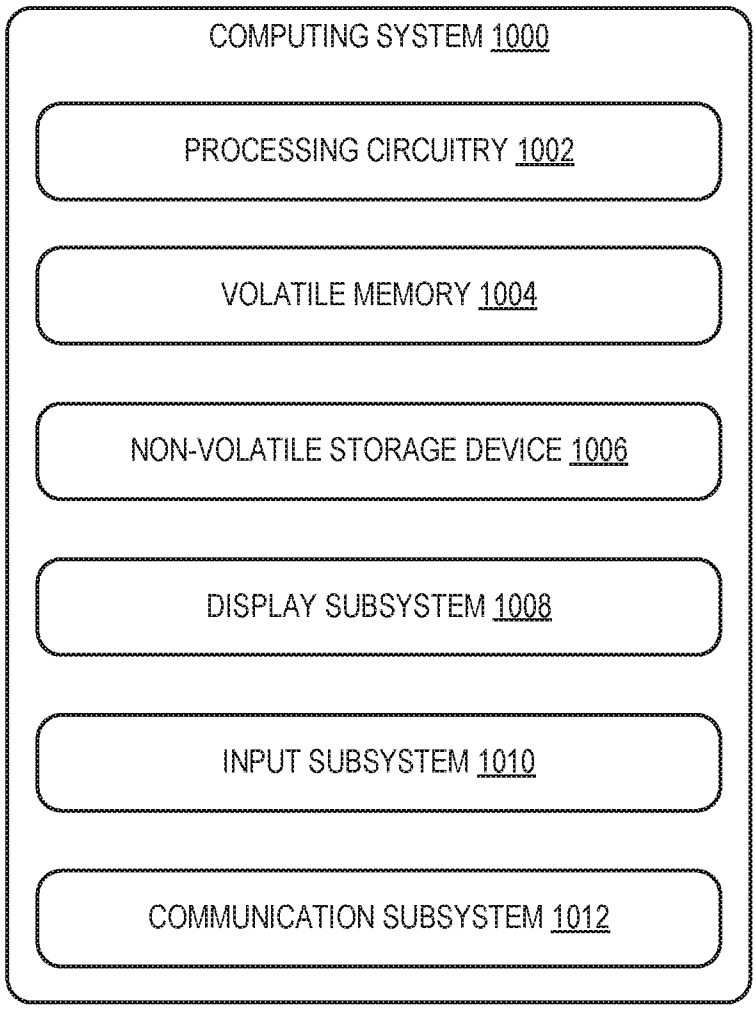

COMPUTING SYSTEM 1000
PROCESSING CIRCUITRY 1002
VOLATILE MEMORY 1004
NON-VOLATILE STORAGE DEVICE 1006
DISPLAY SUBSYSTEM 1008
INPUT SUBSYSTEM 1010
COMMUNICATION SUBSYSTEM 1012

FIG. 19

INTERNATIONAL SEARCH REPORT

International application No
PCT/CN2023/112628

A. CLASSIFICATION OF SUBJECT MATTER
INV. G16C10/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G16C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	FUMIHIRO IMOTO ET AL: "Order-N orbital-free density-functional calculations with machine learning of functional derivatives for semiconductors and metals", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 2 September 2021 (2021-09-02), XP091048122, DOI: 10.1103/PHYSREVRESEARCH.3.033198 abstract ----- -/--	1-30

[x] Further documents are listed in the continuation of Box C.
[ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 January 2024	02/02/2024

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Nurmi, Jussi

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/CN2023/112628

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PAVLO GOLUB ET AL: "Kinetic energy densities based on the fourth order gradient expansion: performance in different classes of materials and improvement via machine learning", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 28 May 2018 (2018-05-28), XP081421880, abstract page 5	1-30
X	ROMAN REMME ET AL: "KineticNet: Deep learning a transferable kinetic energy functional for orbital-free density functional theory", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 8 May 2023 (2023-05-08), XP091515921, abstract	1-30
T	HE ZHANG ET AL: "M-OFDFT: Overcoming the Barrier of Orbital-Free Density Functional Theory for Molecular Systems Using Deep Learning", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 28 September 2023 (2023-09-28), XP091625142, the whole document	

1