EQUIVARIANT FLOW MATCHING FOR ELECTRON DENSITY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning surrogates for density functional theory (DFT) have been increasingly used to reduce the cost of first-principles calculations. In this arena, predicting real-space electron densities offers a scalable and transferable initialization for self-consistent field (SCF) procedures. However, current methods face a clear dilemma. That is, grid-based architectures incur a high computational cost, while basis-set methods fail to capture the structural correlations inherent in the coefficient space. Here, we develop *OrbFlow*, an $\mathrm{SE}(3)$-equivariant generative model that predicts Gaussian-type orbital (GTO) coefficients via flow matching. OrbFlow retains the efficiency of a compact atom-centered basis while replacing pointwise regression with a learned probability path over the full coefficient space. It is trained through a two-phase trajectory curriculum that mitigates discretization drift during numerical integration. OrbFlow achieves state-of-the-art accuracy on QM9, reducing density error by 13.6% relative to the previous best model, and reduces error by 51% to 63% on every molecule of the MD benchmark relative to the strongest prior method sharing its basis. The predicted density also cuts SCF iterations by up to 68% with zero-shot transfer to unseen exchange-correlation functionals and recovers dipole and quadrupole moments to within a few percent of DFT references without any SCF calculation.

1 INTRODUCTION

Density functional theory (DFT) is a cornerstone of computational quantum chemistry, materials science, and solid-state physics (Hohenberg & Kohn, 1964; Kohn & Sham, 1965; Jain et al., 2016; Kirkpatrick et al., 2021; Marzari et al., 2021). It maps the many-body Schrödinger equation onto many one-electron Kohn-Sham equations, in which the total energy is a functional of electron density rather than many-electron wavefunctions, enabling *ab initio* evaluation of molecular energies and electronic structure. The iterative self-consistent field (SCF) calculation that solves the Kohn-Sham equations scales cubically ($\mathcal{O}(N^3)$) with the number of electrons N, making it a bottleneck for large molecular structures, amorphous systems, and long-timescale molecular dynamics.

Recently, rapid advances have been made in developing deep learning surrogates to accelerate or bypass DFT, most commonly by predicting electronic structure operators directly from atomic coordinates (Zhang et al., 2025), such as Kohn-Sham Hamiltonian matrix $\mathbf{H}$ (Tang et al., 2024; Li et al., 2022; Gong et al., 2023; Yu et al., 2023; 2026; Zhong et al., 2023; Gu et al., 2024; Zhouyin et al., 2025; Li et al., 2025; Luo et al., 2025; Kim et al., 2025b) or density matrix $\mathbf{P}$ (Shao et al., 2023; Hazra et al., 2024; Febrer et al., 2025) in localized basis sets. However, these matrix-centric predictions face systemic scalability and transferability bottlenecks. $\mathbf{H}$ is numerically difficult to predict owing to its large, dense entrywise structure (Liu et al., 2026). Models trained on small clusters therefore often fail to generalize to larger, unseen geometries, sometimes causing numerical divergence in the downstream DFT SCF calculations. In contrast, the real-space electron density is a physically smooth, localized, and universally transferable scalar field, and a high-quality initial guess for it can consistently reduce the number of SCF iterations under same convergence threshold.

Existing density prediction methods generally fall into two paradigms (Fu et al., 2024). Probe-based methods evaluate density on dense 3D grids and offer strong expressiveness but incur heavy overhead from cubic spatial scaling. In contrast, basis-set methods are more efficient but rely on pointwise coefficient regression that fails to capture the structural correlations of the coefficient

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

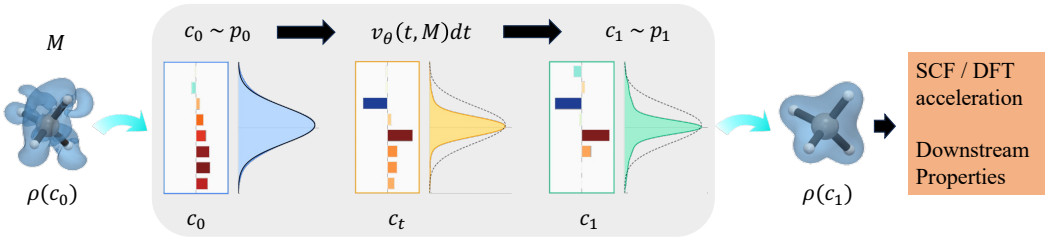


Figure 1: **Overview of OrbFlow.** Given a molecular graph $\mathcal{M}$, OrbFlow initializes Gaussian-type orbital (GTO) coefficients from a structured noise prior $\mathbf{c}_0 \sim p_0$ and integrates a learned, geometry-conditioned vector field $\mathbf{v}_\theta(t, M)$ (equation 4) over $t \in [0, 1]$ to obtain the target coefficients $\mathbf{c}_1 \sim p_1$. The recovered coefficients are converted into a continuous real-space electron density $\rho(\mathbf{r})$ through the auxiliary GTO basis expansion (equation 1), providing a fast surrogate for DFT initialization and enabling direct prediction of downstream physical properties.

space. We propose to address this dilemma by flow matching (FM) (Lipman et al., 2023; Tong et al., 2024), which parameterizes a time-dependent vector field that defines efficient ordinary differential equation (ODE) trajectories and yields rapid convergence while capturing intricate structural dependencies. FM has already achieved state-of-the-art accuracy in predicting quantum mechanical operators such as Kohn-Sham Hamiltonians (Kim et al., 2025b), but its capabilities have not yet been applied to other continuous physical fields such as Gaussian-type orbital (GTO) basis spaces.

Motivated by the above considerations, we introduce *OrbFlow*, a geometry-conditioned, $\mathrm{SE}(3)$-equivariant flow matching framework operating within a compact GTO coefficient space. Rather than a rigid, one-shot regression mapping, OrbFlow models the distribution of optimal basis spaces by learning a probability path that deforms a structured noise prior into the exact target distribution. To eliminate discretization drift during ODE integration rollouts, we propose a two-phase *trajectory curriculum training strategy*. The model is first initialized with an independent flow matching objective to build a stable global vector field. It is then dynamically fine-tuned via an end-to-end differentiable multi-step Euler rollout loss that directly suppresses numerical truncation errors. Figure 1 presents an overview of OrbFlow.

Our primary contributions are summarized below.

- **Few-Step Flow Matching for Basis-Set Density Prediction.** We formulate GTO coefficient prediction as conditional flow matching, and show that its benefit is realized when the flow is trained under its own few-step rollout.
- **Trajectory Curriculum Training.** We introduce a two-phase curriculum that fine-tunes the flow under its own differentiable few-step rollout, requiring no teacher network and matching the accuracy of single-stage alternatives at $2.5\times$ less training compute.
- **Symmetry-Preserving Equivariant Integration.** We pair a high-order equivariant graph transformer with an irrep-isotropic prior, so that symmetry holds at every state along the rollout and the predicted density transforms equivariantly under rigid motions.
- **Accuracy and Downstream Utility.** OrbFlow achieves state-of-the-art accuracy on QM9 and reduces error on every MD molecule relative to the strongest prior model sharing its GTO basis, with SCF acceleration transferring zero-shot across unseen exchange-correlation functionals.

2 Related Work

Electron Density Prediction. Continuous real-space electron density prediction generally falls into two paradigms (Fu et al., 2024). Probe-based models evaluate density as spatial queries on dense 3D grids (Jørgensen & Bhowmik, 2022; Koker et al., 2024; Gong et al., 2019), but cubic scaling with voxel resolution imposes severe memory and throughput costs at scale. Basis-set methods instead predict expansion coefficients of localized analytical functions, and split further by basis choice. Hybrid models combine a plane-wave basis with a localized GTO basis (Kim & Ahn, 2024). Gaussian mixture models such as ELECTRA (Elsborg et al., 2025) instead use a learnable set of Gaussian primitives rather than a fixed atom-centered basis. Auxiliary GTO regression predicts

coefficients directly against a fixed atom-centered basis (Cheng & Peng, 2023; Fu et al., 2024; Qiao et al., 2022; del Rio et al., 2023).

Flow Matching and Exposure Bias. Flow Matching (FM) (Lipman et al., 2023; Tong et al., 2024) has emerged as a powerful alternative to diffusion models for molecular generative tasks (Jing et al., 2024; Miller et al., 2024; Kim et al., 2025a; Stark et al., 2024), most relevantly in QHFlow (Kim et al., 2025b), which frames Hamiltonian matrix regression as an equivariant distribution-matching problem. We extend this paradigm to GTO coefficient spaces, pairing a flow over compact GTO coefficients with a trajectory-regularized multi-step Euler objective. The mismatch between single-step training and the compounding error of multi-step inference is known as exposure bias (Bengio et al., 2015; Ranzato et al., 2015). Diffusion and flow-based models remove it in two ways. Distillation trains a student to reproduce the multi-step behavior of a separately trained, frozen teacher, as in progressive distillation (Salimans & Ho, 2022) and consistency models (Song et al., 2023). Shortcut models (Frans et al., 2025) drop the teacher instead, conditioning the network on the step size so that one model stays accurate at many step budgets. Our curriculum takes a third route and needs neither. It continues training the same network under its own K-step rollout, with identical weights across both phases, so it adds no model capacity, no teacher, and no extra conditioning input.

3 METHODS

3.1 CHARGE DENSITY REPRESENTATION VIA GAUSSIAN-TYPE ORBITALS

To represent the continuous molecular electron density $\rho(\mathbf{r})$ compactly, we adopt an atom-centered Gaussian-type orbital (GTO) basis expansion (Eichkorn et al., 1995), following Fu et al. (2024). Each basis function $\varphi_{\alpha,\ell,m,\mathbf{R}_i}(\mathbf{r})$ is a normalized spherical Gaussian centered at node coordinate $\mathbf{R}_i$, characterized by radial exponent α, angular momentum ℓ, and magnetic quantum number m. The full functional form and the even-tempered scheme used to construct the radial exponents are given in Appendix A.1. Given a molecular geometry $\{\mathbf{R}_i\}_{i=1}^{N}$ containing both nuclear coordinates and bond-midpoint virtual nodes, we parameterize the spatial density field as a linear combination of these localized GTOs,

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} \sum_{j=1}^{N_i^{\mathrm{b}}} \sum_{m=-\ell_{i,j}}^{\ell_{i,j}} c_{i,j,m}\, \varphi_{\alpha_{i,j},\ell_{i,j},m,\mathbf{R}_i}(\mathbf{r}), \tag{1}$$

where N_i^{b} denotes the number of unique radial shells assigned to node i. Following the basis enrichment recipe of Fu et al. (2024), we construct our basis by starting from uncontracted `def2-QZVPPD` elemental configurations (Rappoport & Furche, 2010), embedding $\mathrm{SE}(3)$-equivariant virtual nodes at bond midpoints to capture interatomic shared densities, and applying even-tempered geometric shell augmentations (Bardo & Ruedenberg, 1974) for radial flexibility. Element-specific hyperparameters are detailed in Appendix A.2, and the representation floor of this basis, the error of the best fit it can express, is reported in Appendix A.4.

3.2 EQUIVARIANT FLOW MATCHING

Problem Formulation. Our goal is to predict the target GTO coefficient vector $\mathbf{c}^\star \in \mathbb{R}^d$ from a molecular graph $\mathcal{G} = (\mathbf{R}, \mathbf{H})$, where $\mathbf{R} \in \mathbb{R}^{N\times 3}$ and $\mathbf{H} \in \mathbb{R}^{N\times D}$ denote the node coordinates and features. The density is reconstructed linearly from these coefficients on a discrete grid as

$$\boldsymbol{\rho}(\mathbf{c}) = \mathcal{O}(\mathbf{c}) = s\, \boldsymbol{\Phi}\, \mathbf{c}, \tag{2}$$

where $\Phi_{pj} = \varphi_j(\mathbf{r}_p)$ evaluates the j-th GTO function at grid point p, $s > 0$ is a scale factor, and $\mathcal{O}(\cdot)$ is our differentiable density synthesis operator. We train on $\mathcal{D} = \{(\mathcal{G}, \mathbf{c}^\star)\}$, where target coefficients $\mathbf{c}^\star$ are obtained by projecting ground-truth DFT densities onto the basis via a regularized least-squares fit (Appendix A.3). Instead of mapping $\mathcal{G} \mapsto \mathbf{c}^\star$ in a single step, our model learns a transport from a tractable prior $\mathbf{c}_0 \sim p_0(\cdot \mid \mathcal{G})$ to the target $\mathbf{c}^\star$.

Conditional Flow Matching in GTO Space. We formulate a continuous normalizing flow (CNF) within this compact, auxiliary GTO coefficient space: a time-dependent, geometry-conditioned vector field $\mathbf{v}_\theta(\mathbf{x}_t, t)$ defining an ODE $\mathrm{d}\mathbf{x}_t/\mathrm{d}t = \mathbf{v}_\theta(\mathbf{x}_t, t)$ over $t \in [0, 1]$ that deforms the prior $p_0(\mathbf{c}_0)$

at $t = 0$ into a distribution on $\mathbf{c}^\star$ at $t = 1$. Following Kim et al. (2025b), we use a linear probability bridge,

$$\mathbf{x}_t = (1 - t)\,\mathbf{c}_0 + t\,\mathbf{c}^\star, \tag{3}$$

with conditional field $\mathbf{u}_t(\mathbf{x}_t \mid \mathbf{c}^\star, \mathbf{c}_0) = \mathbf{c}^\star - \mathbf{c}_0$. We parameterize $\mathbf{v}_\theta$ via an endpoint scheme,

$$\mathbf{v}_\theta(\mathbf{x}_t, t) = \frac{f_\theta(\mathcal{G}, \mathbf{x}_t, t) - \mathbf{x}_t}{1 - t}, \tag{4}$$

where $f_\theta(\mathcal{G}, \mathbf{x}_t, t)$ predicts the clean target endpoint $\hat{\mathbf{c}}^\star \approx \mathbf{c}^\star$. Since our physical objective is real-space density accuracy rather than raw coefficient error, we supervise the predicted endpoint through the synthesis operator $\mathcal{O}(\cdot)$, evaluating the MAE over P' subsampled spatial probes:

$$\mathcal{L}_{\text{stage1}}(\theta) = \mathbb{E}_{t,\,\mathcal{G},\,\mathbf{c}_0}\Big[\tfrac{1}{P'}\sum_{p=1}^{P'}\big|\big[\mathcal{O}(f_\theta(\mathcal{G}, \mathbf{x}_t, t))\big]_p - \boldsymbol{\rho}_p^{\text{lab}}\big|\Big], \tag{5}$$

Two-Phase Trajectory Curriculum Learning. Training solely on isolated, random timesteps via equation 5 does not align the vector field with the multi-step trajectories used at inference. We address this with a two-phase curriculum. Phase 1 first establishes a good vector field at arbitrary, isolated timesteps. Phase 2 then fine-tunes that field under the same multi-step rollout used at inference, so training conditions match evaluation conditions directly.

Phase 1 (Field Initialization). We minimize $\mathcal{L}_{\text{stage1}}$ for T_1 steps, teaching the network to map arbitrary perturbed states $(\mathcal{G}, \mathbf{x}_t, t)$ to a target endpoint whose synthesized density matches the true DFT field. This establishes a stable continuous normalizing flow without the overhead of unrolling the ODE, giving a stable warm-start vector field $\mathbf{v}_\theta$.

Phase 2 (Trajectory Regularization). Starting from the Phase-1 weights, we continue training for T_2 steps under a full K-step Euler rollout that mirrors how the model is actually used at inference. Each rollout begins at the pure noise prior $\mathbf{x}_{t_0} = \mathbf{c}_0$ at $t_0 = 0$, rather than at an interpolated point along the bridge, and is unrolled end to end through the network in a differentiable manner. We discretize the interval $[0, 1]$ into K equal steps of size $\Delta t = 1/K$, with $t_k = k \cdot \Delta t$, and advance the state at each step with the Euler update

$$\mathbf{x}_{t_{k+1}} = \mathbf{x}_{t_k} + \Delta t \cdot \mathbf{v}_\theta(\mathbf{x}_{t_k}, t_k), \quad \text{for } k = 0, \ldots, K - 1. \tag{6}$$

The final state $\hat{\mathbf{c}}_1 = \mathbf{x}_{t_K}$ is the model's simulated prediction of the target coefficients, using $K = 2$ in our primary experiments. We train the model by comparing the density reconstructed from this simulated endpoint to the true DFT density, giving the Phase-2 objective

$$\mathcal{L}_{\text{stage2}}(\theta) = \mathbb{E}_{\mathcal{G},\,\mathbf{c}_0}\Big[\tfrac{1}{P'}\sum_{p=1}^{P'}\big|\big[\mathcal{O}(\hat{\mathbf{c}}_1)\big]_p - \boldsymbol{\rho}_p^{\text{lab}}\big|\Big]. \tag{7}$$

In Phase 1, the input state $\mathbf{x}_t$ is built by interpolating with the ground-truth target itself via equation 3, so the network is partially shown the answer during training. In Phase 2, the rollout starts from pure noise with no such guidance, so the model must navigate the full trajectory on its own, and any small per-step error can accumulate into discretization drift. Phase 2 directly penalizes this accumulated drift. Penalizing this drift acts as a trajectory regularizer that straightens the flow field, which is why Phase 2 improves performance at the trained rollout horizon even though it starts from Phase-1 weights that already minimize $\mathcal{L}_{\text{stage1}}$. Phase 2 achieves this by continuing to train the same network under its own rollout, rather than by introducing a separate teacher network or a distillation step.

Equivariant Flow and Isotropic Irrep Priors. For the generated density to be invariant under rigid translations and rotations across both phases, the flow must be strictly $\mathrm{SE}(3)$-equivariant. Atomic coordinates are centered at the origin ($\mathbf{r}_i \leftarrow \mathbf{r}_i - \frac{1}{N}\sum_j \mathbf{r}_j$), removing translational dependence. Under a rotation $R \in \mathrm{SO}(3)$, GTO basis functions (built from spherical harmonics Y_ℓ^m) of angular momentum ℓ transform via the $(2\ell + 1)$-dimensional Wigner D-matrix $D^{(\ell)}(R)$, so the coefficient vector $\mathbf{c}$ transforms covariantly as a direct sum of irreducible representations,

$$\mathbf{c} \xrightarrow{R} \mathbf{D}(R)\mathbf{c} = \bigoplus_\ell D^{(\ell)}(R)\mathbf{c}^{(\ell)}. \tag{8}$$

We therefore require both the network f_θ and the prior $p_0(\mathbf{c}_0)$ to respect this symmetry. We parameterize f_θ with a high-order equivariant graph transformer (Sec. 3.3) satisfying

$$f_\theta\big(R \cdot \mathcal{G},\, \mathbf{D}(R)\mathbf{x}_t,\, t\big) = \mathbf{D}(R) f_\theta(\mathcal{G}, \mathbf{x}_t, t), \tag{9}$$

and require $p_0(\mathbf{c}_0) = p_0(\mathbf{D}(R)\mathbf{c}_0)$ for all $R \in \mathrm{SO}(3)$, satisfied by an *irrep-isotropic Gaussian prior* in which each angular momentum subspace ℓ is drawn independently from a zero-mean isotropic Gaussian, $\mathbf{c}_0^{(\ell)} \sim \mathcal{N}(\mathbf{0}, \sigma_\ell^2 \mathbf{I}_{2\ell+1})$, with σ_ℓ^2 matched to the empirical variance of the target coefficients in subspace ℓ.

Proposition 1. *Let $\mathbf{c}_0 \sim p_0(\mathbf{c}_0)$ be sampled from the irrep-isotropic prior. Under any rotation $R \in \mathrm{SO}(3)$, the prior density is invariant, $p_0(\mathbf{c}_0) = p_0(\mathbf{D}(R)\mathbf{c}_0)$. The proof is in Appendix B.*

Pairing this rotationally invariant prior with our equivariant network f_θ guarantees that integrating the velocity field yields a physically consistent, symmetry-preserving density.

3.3 Model Implementation

We parameterize f_θ using a stack of L EquiformerV3 blocks (Liao et al., 2026), leveraging $\mathrm{SE}(3)$-equivariant multi-head attention and local $\mathrm{SO}(2)$ convolutions (Passaro & Zitnick, 2023) to jointly process the molecular geometry $\mathcal{G}$ and the current flow state $\mathbf{x}_t$. The backbone readout is mapped to per-atom GTO coefficients via a bilinear `e3nn` tensor-product readout (Geiger & Smidt, 2022), serving directly as the flow head f_θ in equation 4. Full architectural details are in Appendix C.

4 Experiments

We evaluate OrbFlow on the QM9 and MD benchmarks, building on the GTO auxiliary basis formulation of SCDP (Fu et al., 2024) and a state-of-the-art EquiformerV3 (Liao et al., 2026) backbone. We evaluate predicted real-space densities $\boldsymbol{\rho}^{\mathrm{pred}} = \mathcal{O}(\hat{\mathbf{c}}_1)$ against ground-truth DFT references $\boldsymbol{\rho}^{\mathrm{lab}}$ on the full spatial grid. All QM9 and MD results are averaged over three independently trained random seeds. We report global reconstruction accuracy via the normalized Mean Absolute Error, $\mathrm{NMAE} = 100 \times \frac{\|\boldsymbol{\rho}^{\mathrm{pred}} - \boldsymbol{\rho}^{\mathrm{lab}}\|_1}{\|\boldsymbol{\rho}^{\mathrm{lab}}\|_1}$.

4.1 QM9 Dataset

Dataset. QM9 (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014) contains ∼134,000 stable organic molecules with up to nine heavy atoms ($\mathrm{C}, \mathrm{O}, \mathrm{N}, \mathrm{F}$) plus H. We use the real-space density files curated by Jørgensen & Bhowmik (2022) (VASP, PBE functional, PAW method), adopting the standard 123,835/50/10,000 train/val/test partition. To compare directly with ChargE3Net (Koker et al., 2024), InfGCN (Cheng & Peng, 2023), and GPWNO (Kim & Ahn, 2024), we also evaluate on "Split 2," the last 1,600 structures of the standard test set, matching these baselines' protocol.

Results. OrbFlow achieves state-of-the-art accuracy on both partitions (Table 1). It reaches 0.152% NMAE on the primary test set, a 13.6% reduction over the strongest baseline ELECTRA (0.176%) and 14.6% over its closest GTO-based predecessor SCDP + VO (0.178%), and 0.165% on Split 2, 13.6% below SCDP + VO and 18.7% below ChargE3Net (0.203%). These margins reflect both the flow matching formulation and the EquiformerV3 backbone. Sec. 4.6 separates the two at fixed architecture and compute, and finds that the flow matching formulation contributes on its own. They are also best read against the representation floor of GTO basis, since a direct least-squares fit of the ground-truth densities reaches 0.061% NMAE (Appendix A.4) and OrbFlow removes 22% of the error SCDP + VO leaves above it. Figure 2 visualizes predicted isosurfaces and spatial error.

Training Details. We train with Adam under a linear warm-up followed by a power-decayed schedule, following the two-phase curriculum of Sec. 3.2 with rollout budget $K = 2$. Full hyperparameters are in Appendix D.1.

4.2 MD Dataset

Dataset. We benchmark on molecular dynamics (MD) trajectories spanning six molecules, ethanol, benzene, phenol, resorcinol, ethane, and malonaldehyde (MDA) (Brockherde et al., 2017; Bogo-

Table 1: QM9 charge density prediction error, grouped by model representation. The first column reports NMAE [%] on the standard 10,000-molecule test set, and the second on "Split 2," the last 1,600 structures of that test set. Results for OrbFlow and ELECTRA were obtained in this work as means over three training seeds, all other entries are quoted from the cited papers. Lower NMAE is better (↓). **Bold** marks the best result per column, underline the second-best.

Model	NMAE [%] ↓	NMAE, Split 2 [%] ↓
Probe-based		
i-DeepDFT (Jørgensen & Bhowmik, 2022)	0.357 ± 0.001	–
e-DeepDFT (Jørgensen & Bhowmik, 2022)	0.284 ± 0.001	–
ChargE3Net (Koker et al., 2024)	0.196 ± 0.001	0.203 ± 0.003
Hybrid PW & GTO		
GPWNO (Kim & Ahn, 2024)	–	0.73
Gaussian Mixture		
ELECTRA (Elsborg et al., 2025)	$\underline{0.176 \pm 0.001}$	–
GTO-based		
InfGCN (Cheng & Peng, 2023)	0.869 ± 0.002	0.93
SCDP, $L = 3$ (Fu et al., 2024)	0.432 ± 0.001	0.438 ± 0.003
SCDP + VO, $L = 6$ (Fu et al., 2024)	0.178 ± 0.001	$\underline{0.191 \pm 0.002}$
OrbFlow (ours)	$\mathbf{0.152 \pm 0.001}$	$\mathbf{0.165 \pm 0.001}$

jeski et al., 2020), with reference densities computed at the PBE level via plane-wave Quantum ESPRESSO calculations, sampled across diverse geometries on uniform 3D grids.

Results. Table 2 reports NMAE across the six MD trajectories. Compared with SCDP, the strongest prior GTO-based method, OrbFlow reduces test error on every molecule by 51% to 63% in relative terms, with the largest gains on benzene (1.13% to 0.421%) and ethane (2.05% to 0.871%). Averaged over the six molecules, NMAE falls from 1.81% to 0.798%, a 2.3× reduction at comparable model size. On benzene and ethane OrbFlow comes within 0.02 and 0.06 percentage points of the basis fitting floor (Appendix A.4), leaving little room for further gains in this basis. Against ELECTRA, a 723M-parameter Gaussian-mixture model with roughly $55\times$ more parameters, OrbFlow remains competitive. OrbFlow is more accurate on benzene and ethane and statistically indistinguishable on phenol (0.575% against 0.573%, well within seed variance), while ELECTRA leads on ethanol, resorcinol, and malonaldehyde, by a small margin on the first two and a clear one on malonaldehyde. Therefore, OrbFlow approaches the accuracy of a much larger Gaussian-mixture model while improving substantially over every prior method operating in the same GTO basis, and it outperforms the hybrid plane-wave method GPWNO on all six molecules by wide margins. Figure 2 visualizes predicted density isosurfaces and spatial error for both benchmarks.

Training Details. Full details are in Appendix D.1.

4.3 DFT Acceleration and Functional Transferability

A key downstream application of machine-learned charge density is to use it as improved initial guess for SCF calculations, thereby reducing the number of SCF iterations and accelerating DFT calculations. Standard Kohn-Sham DFT calculations typically initialize the SCF loop using a superposition of atomic densities (SAD), which may provide a relatively poor approximation to the converged electronic density and therefore require a larger number of computationally expensive SCF iterations. To improve the initialization, we instead use the real-space charge density ρ_{pred} predicted by OrbFlow as the initial density for the SCF calculation.

The resulting SCF acceleration was evaluated using two complementary metrics. The Relative Iteration Count (RIC), defined as $\mathrm{RIC} = (N_{\mathrm{pred}}/N_{\mathrm{SAD}}) \times 100\%$, compares the number of SCF iterations required to reach the specified energy convergence criterion when initialized with the OrbFlow-predicted density and the conventional SAD density, respectively. Here, N_{pred} and N_{SAD} denote

Table 2: NMAE [%] on MD test sets for OrbFlow, SCDP (Fu et al., 2024), InfGCN (Cheng & Peng, 2023), ELECTRA (Elsborg et al., 2025), and GPWNO (Kim & Ahn, 2024), grouped by model representation. **Bold** marks the best result per row, underline marks the second-best. Results for OrbFlow and ELECTRA were obtained in this work and are reported as mean ± std over three training seeds. SCDP, InfGCN and GPWNO numbers are taken from Fu et al. (2024). ELECTRA's ethanol standard deviation is below 5×10^{-4} and rounds to 0.000.

Molecule	*GTO-based*			*Gaussian Mixture*	*Hybrid PW & GTO*
	OrbFlow (ours)	**SCDP**	**InfGCN**	**ELECTRA**	**GPWNO**
# Params	13M	16M	–	723M	–
MD-ethanol	1.078 ± 0.023	2.34 ± 0.25	8.43	$\mathbf{1.030 \pm 0.000}$	4.00
MD-benzene	$\mathbf{0.421 \pm 0.002}$	1.13 ± 0.06	5.11	0.472 ± 0.013	2.45
MD-phenol	$\mathbf{0.575 \pm 0.008}$	1.29 ± 0.07	5.51	$\mathbf{0.573 \pm 0.012}$	2.68
MD-resorcinol	0.663 ± 0.014	1.35 ± 0.08	5.95	$\mathbf{0.619 \pm 0.005}$	2.73
MD-ethane	$\mathbf{0.871 \pm 0.008}$	2.05 ± 0.12	7.01	0.970 ± 0.055	3.67
MD-malonaldehyde	1.177 ± 0.012	2.71 ± 0.60	10.34	$\mathbf{0.826 \pm 0.025}$	5.32

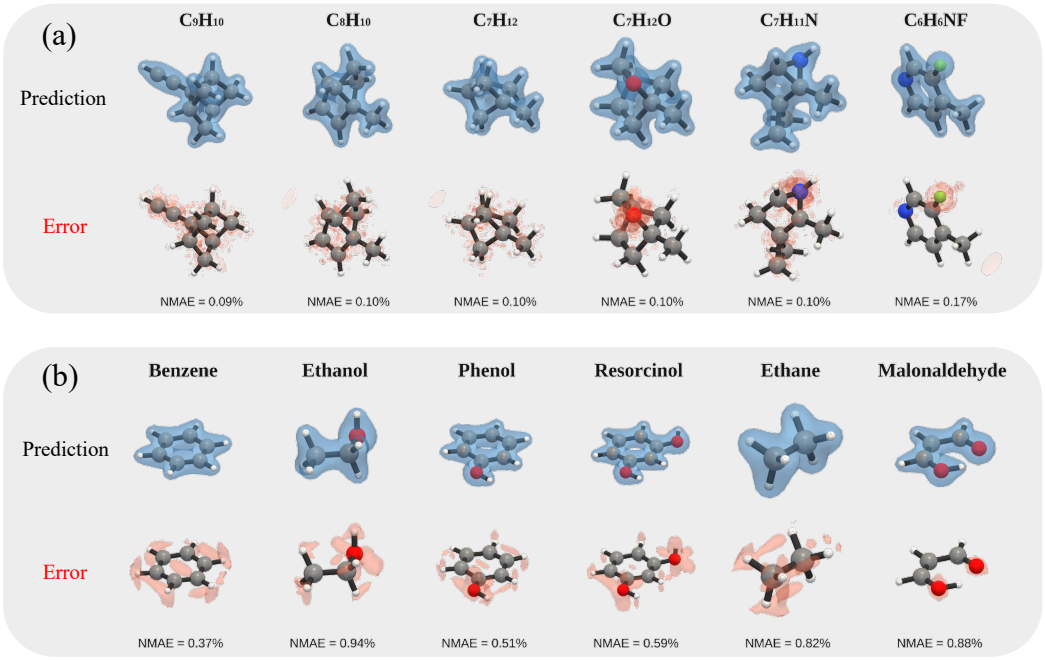


Figure 2: Visualization of OrbFlow-predicted electron densities and prediction error. (a) Six representative QM9 test molecules. (b) Six representative MD test molecules. *Prediction* rows show the predicted density isosurface, rendered at $\rho = 0.69\,\mathrm{e/Å^3}$ for QM9 and $\rho = 1.12\,\mathrm{e/Å^3}$ for MD. *Error* rows show an isosurface of the absolute error $|\rho^{\mathrm{pred}} - \rho^{\mathrm{DFT}}|$, at $|\Delta\rho| = 1.2 \times 10^{-3}\,\mathrm{e/Å^3}$ for QM9 and $|\Delta\rho| = 1.1 \times 10^{-2}\,\mathrm{e/Å^3}$ for MD. Per-example NMAE is reported below for each molecule.

the corresponding numbers of SCF iterations. Similarly, the Relative Time Ratio (RTR), defined as $\mathrm{RTR} = (T_{\mathrm{pred}}/T_{\mathrm{SAD}}) \times 100\%$, compares the corresponding total wall-clock times, where T_{pred} and T_{SAD} denote the wall-clock times for calculations initialized with ρ_{pred} and SAD, respectively.

Functional Transferability. To assess whether OrbFlow learns transferable physical representations rather than memorizing artifacts specific to a single level of theory, we evaluate its zero-shot generalization across unseen exchange-correlation (XC) functionals using held-out QM9 molecules. All reference calculations are performed using VASP (Kresse & Furthmüller, 1996) with a fixed plane-wave basis, with the complete computational settings provided in Appendix D.2. Although trained exclusively at the GGA level, OrbFlow is evaluated across higher rungs of Jacob's ladder to

Table 3: SCF acceleration from OrbFlow initializations, evaluated on held-out QM9 molecules. N_{SAD} and N_{pred} are SCF iterations to convergence from the SAD and OrbFlow initializations, respectively. Lower $\mathrm{RIC} \downarrow$ and lower $\mathrm{RTR} \downarrow$ both indicate greater acceleration. Iteration counts are means over the evaluated molecules and the $\pm$ values are standard deviations across those molecules. RIC and RTR are computed from the means.

Functional (Family: Name)	N_{SAD}	N_{pred}	**RIC [%]** $\downarrow$	**RTR [%]** $\downarrow$
Training-distribution functional				
GGA: PBE	21.33 ± 1.03	11.62 ± 1.65	54.48	89.51
Zero-shot transfer to unseen XC functionals				
Hybrid GGA: B3LYP5	21.60 ± 1.73	16.64 ± 1.42	77.04	83.12
Hybrid GGA: PBE0	43.02 ± 4.77	13.71 ± 2.47	31.87	39.02

hybrid functionals (B3LYP5 Becke (1993); Lee et al. (1988), PBE0 Adamo & Barone (1999); Paier et al. (2005)), without any retraining.

As shown in Table 3, at the training-distribution functional PBE, OrbFlow reduces the required SCF iterations to 54.48% of the SAD baseline and the wall-clock time to 89.51%. The wall-clock saving is smaller because SCF iterations are only one component of a DFT run (Appendix D.3). This acceleration carries over to the two functionals never seen during training, both of which evaluate exact exchange rather than the semi-local exchange-correlation OrbFlow was trained on. PBE0 exhibits the strongest speedup on both metrics, reducing SCF iterations by 68.13% and wall-clock time by 60.98%, though its SAD baseline converges unusually slowly (43.02 iterations against 21.33 for PBE). B3LYP5 also benefits, with a larger wall-clock reduction than at the training-distribution functional (RTR 83.12% against 89.51%) though a higher relative iteration count (RIC 77.04% against 54.48%). Taken together, these results indicate that OrbFlow's predicted density is a genuinely informative initial guess across levels of theory rather than a narrow fit to PBE-specific dynamics.

4.4 Direct Physical Properties from Predicted Densities

Beyond accelerating SCF, a predicted density also serves as a DFT surrogate. Density-derived observables follow directly from $\boldsymbol{\rho}^{\mathrm{pred}}$, with no self-consistent calculation. We evaluate the molecular dipole moment $\boldsymbol{\mu}$ and the traceless quadrupole tensor Q_{ab} on 500 held-out QM9 molecules, against PBE references computed with VASP. Relative MAE is 1.0% on the dipole magnitude and 1.1% on the quadrupole norm, and 1.7% and 2.5% on their individual components (Table 11, Appendix D.4). The accuracy is uniform across molecules rather than an average over well-predicted and poorly-predicted ones (Figure 3, Appendix D.4). Accuracy on both indicates that OrbFlow captures the density's shape at multiple spatial scales. One prediction therefore supports two downstream uses, an optimized SCF initial guess (Sec. 4.3) and a stand-alone property surrogate.

4.5 Inference Efficiency

Table 12 reports per-molecule inference latency. OrbFlow's end-to-end cost of 0.066s is on par with ELECTRA (0.061s), and $1.6\times$ that of the single-shot GTO baseline SCDP, reflecting the two Euler steps of the rollout. The absolute overhead is a few hundredths of a second per molecule, negligible against a single SCF iteration, which typically costs on the order of seconds. Because the network cost scales close to linearly with K, OrbFlow's latency is predictable and directly controllable through the rollout budget. The full protocol is in Appendix D.5.

4.6 Ablation Studies

All ablations use a random 10% subset of QM9, subsampled in both the training and test splits, held fixed across configurations, with three seeds and the protocol of Sec. 4.1. Absolute NMAE is therefore higher than in Table 1, where models train on the full set. We ask three questions, whether the flow formulation contributes beyond the backbone, whether the two-phase curriculum is necessary, and how many Euler steps are needed.

What the Flow Formulation Contributes. Our flow formulation differs from direct regression in two ways. It starts from noise drawn from the prior rather than from a fixed input, and it is supervised on the output of its own multi-step rollout (equation 7) rather than one timestep at a time (equation 5). The four cells in Table 4 run from direct regression, which does neither, to OrbFlow, which does both.

Neither ingredient works alone. Direct regression reaches $0.280 \pm 0.003\%$. Adding noise alone (Phase-1-only training) reaches $0.295 \pm 0.016\%$, slightly worse. Adding rollout alone, a deterministic two-step chain $\hat{\mathbf{c}} = g_\theta\big(\mathcal{G}, g_\theta(\mathcal{G}, \mathbf{0})\big)$, reaches $0.268 \pm 0.003\%$. Both together reach $0.240 \pm 0.005\%$. Starting from noise exposes the network to a wide range of input states, and rollout supervision makes that exposure count by training on the states inference actually visits, the model's own intermediates rather than points interpolated toward the target. All four cells share the backbone, parameter count, and 9×10^4-step budget, both rollout cells use two network evaluations, and the deterministic chain keeps a learned step-index embedding (Appendix E.1), so neither architecture, compute, nor step conditioning explains the gap. That gap is what few-step methods such as consistency (Song et al., 2023) and shortcut (Frans et al., 2025) models exist to close.

Table 4: Ablations as a 2×2 over where the input state comes from and what the loss supervises. NMAE (%), mean $\pm$ std over three seeds. All cells share the backbone, parameter count, basis, real-space loss, and step budget, and all use two network evaluations per prediction except direct regression, which uses one.

Input state	Single-step loss	Rollout loss
Deterministic (from $\mathbf{0}$)	0.280 ± 0.003	0.268 ± 0.003
Stochastic (noise prior)	0.295 ± 0.016	$\mathbf{0.240 \pm 0.005}$

Two-Phase Training Curriculum. Both ingredients are needed, so the question becomes how to train them. Table 5 compares three routes to the same $K{=}2$ objective. Training directly at $K{=}2$ with no warm start reaches $0.250 \pm 0.027\%$, and optimizing the single-step and rollout losses jointly from random initialization reaches $0.241 \pm 0.002\%$, within seed variance of our $0.240 \pm 0.005\%$. Accuracy is therefore not what separates the rollout-trained variants. Cost is, 74 GPU-h for the curriculum against 81 without a warm start and 182 for joint optimization, and the curriculum reuses one Phase-1 checkpoint across rollout horizons while the alternatives retrain for each. Budgets here are matched in optimization steps rather than wall-clock, since a Phase-2 step evaluates the network twice.

Table 5: Curriculum variants on the same subset, with total training compute. The rollout-budget sweep and full compute accounting are in Appendix E.2.

Configuration	NMAE (%) ↓	GPU-h
Phase 1 only	0.295 ± 0.016	35
Phase 2 only	0.250 ± 0.027	81
Joint single-stage	0.241 ± 0.002	182
Full curriculum (Ours)	$\mathbf{0.240 \pm 0.005}$	**74**

How Many Euler Steps. We sweep the step count twice, over the Phase-2 rollout horizon K used during training (Appendix E.3) and over the integration budget used at inference by the resulting $K{=}2$ model (Appendix E.4). In training, $K{=}1$ matches Phase-1-only (0.296% against 0.295%), since under the endpoint parameterization a single step from $t{=}0$ lands where the single-step loss already supervises, while $K{=}2$ and $K{=}3$ tie at a cost that grows with K. We therefore train at $K{=}2$. At inference, integrating more finely degrades accuracy, from 0.240% at two steps to 0.539% at four, because Phase 2 sharpens the network only at the states the two-step rollout visits. The train-matched budget is thus both the most accurate and the cheapest.

5 Conclusion and Limitations

We introduced OrbFlow, an $\mathrm{SE}(3)$-equivariant flow matching framework for electron density prediction. OrbFlow reaches state-of-the-art accuracy on QM9 and lowers error on every MD molecule against the strongest prior method sharing its basis, while matching a Gaussian-mixture model $55\times$ its size in accuracy and latency. Its predicted densities accelerate SCF across unseen exchange-correlation functionals and recover dipole and quadrupole moments without running SCF. The rollout adds a small inference overhead, and accuracy holds only at the trained horizon, so OrbFlow is best understood as a trained K-step map. Supervision across a range of horizons is a natural route to discretization-free behavior.

AI USE STATEMENT

In this work, we used generative AI tools to polish and edit the manuscript for readability and to revise portions of its text, including tightening prose, reorganizing sections, and refining table captions. These tools were also used to check the manuscript for internal consistency, such as agreement between numbers reported in tables and in the surrounding text.

We did not use generative AI tools for the research itself. In particular, they were not used to generate synthetic datasets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to propose or refine hypotheses, to design our methodology or experiments, to implement methods, to clean or reformat datasets, or to interpret experimental results. The model, the training procedure, the experimental design, all reported numbers, and the analysis of what they mean are the authors' own work.

We have reviewed all AI-assisted work. Every AI-edited passage was read and revised by the authors against the underlying results, and all numerical values, claims, and citations in the manuscript were verified by the authors against our own experimental logs and the cited sources. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All settings needed to reproduce our results are specified in the appendices: the GTO basis construction and element-specific exponents in Appendix A, the density-fitting projection used to build training targets in Appendix A.3, the network configuration and flow-matching objective in Appendix C, the per-dataset hyperparameters and split sizes in Appendix D.1, and the DFT solver settings and SCF initialization protocol in Appendix D.2. All reported QM9 and MD numbers are means over three training seeds under identical step budgets, with standard deviations given in every main table. The QM9 and MD reference densities are public datasets (Jørgensen & Bhowmik, 2022; Brockherde et al., 2017; Bogojeski et al., 2020) used with the standard splits of prior work. Source code and trained checkpoints will be released publicly upon acceptance.

ETHICS STATEMENT

This work predicts electron densities for organic molecules from public quantum-chemistry datasets and involves no human subjects, personal data, or sensitive information. The primary societal impact we anticipate is a reduction in the computational cost, and hence the energy consumption, of electronic structure calculations. As with any method that accelerates DFT, the resulting capability is general-purpose and could in principle be applied to harmful as well as beneficial chemistry. We consider this risk to be no greater than that of the underlying DFT methods the model approximates.

REFERENCES

Carlo Adamo and Vincenzo Barone. Toward reliable density functional methods without adjustable parameters: The pbe0 model. *The Journal of Chemical Physics*, 110(13):6158–6170, April 1999. ISSN 1089-7690. doi: 10.1063/1.478522. URL http://dx.doi.org/10.1063/1.478522.

Richard D Bardo and Klaus Ruedenberg. Even-tempered atomic orbitals. vi. optimal orbital exponents and optimal contractions of gaussian primitives for hydrogen, carbon, and oxygen in molecules. *The Journal of Chemical Physics*, 60(3):918–931, 1974.

Axel D. Becke. Density-functional thermochemistry. iii. the role of exact exchange. *The Journal of Chemical Physics*, 98(7):5648–5652, April 1993. ISSN 1089-7690. doi: 10.1063/1.464913. URL http://dx.doi.org/10.1063/1.464913.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

P. E. Blöchl. Projector augmented-wave method. *Phys. Rev. B*, 50:17953–17979, Dec 1994. doi: 10.1103/PhysRevB.50.17953. URL https://link.aps.org/doi/10.1103/PhysRevB.50.17953.

Mihail Bogojeski, Leslie Vogt-Maranto, Mark E Tuckerman, Klaus-Robert Müller, and Kieron Burke. Quantum chemical accuracy from density functional approximations via machine learning. *Nature Communications*, 11:5223, 2020.

Felix Brockherde, Leslie Vogt, Li Li, Mark E Tuckerman, Kieron Burke, and Klaus-Robert Müller. Bypassing the kohn-sham equations with machine learning. *Nature Communications*, 8:872, 2017.

Chaoran Cheng and Jian Peng. Equivariant neural operator learning with graphon convolution. *Advances in Neural Information Processing Systems*, 36:61960–61984, 2023.

Beatriz G del Rio, Brandon Phan, and Rampi Ramprasad. A deep learning framework to emulate density functional theory. *npj Computational Materials*, 9:158, 2023.

Karin Eichkorn, Oliver Treutler, Holger Öhm, Marco Häser, and Reinhart Ahlrichs. Auxiliary basis sets to approximate coulomb potentials. *Chemical Physics Letters*, 240(4):283–290, 1995.

Jonas Elsborg, Luca Thiede, Alán Aspuru-Guzik, Tejs Vegge, and Arghya Bhowmik. ELECTRA: A Cartesian network for 3D charge density prediction with floating orbitals. *Advances in Neural Information Processing Systems*, 38:28092–28121, 2025.

Pol Febrer, Peter Bjørn Jørgensen, Miguel Pruneda, Alberto Garcia, Pablo Ordejón, and Arghya Bhowmik. Graph2mat: universal graph to matrix conversion for electron density prediction. *Machine Learning: Science and Technology*, 6(2):025013, 2025.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Xiang Fu, Andrew Rosen, Kyle Bystrom, Rui Wang, Albert Musaelian, Boris Kozinsky, Tess Smidt, and Tommi Jaakkola. A recipe for charge density prediction. *Advances in Neural Information Processing Systems*, 37:9727–9752, 2024.

Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

Sheng Gong, Tian Xie, Taishan Zhu, Shuo Wang, Eric R Fadel, Yawei Li, and Jeffrey C Grossman. Predicting charge density distribution of materials using a local-environment-based graph convolutional network. *Physical Review B*, 100(18):184103, 2019.

Xiaoxun Gong, He Li, Nianlong Zou, Runzhang Xu, Wenhui Duan, and Yong Xu. General framework for e (3)-equivariant neural network representation of density functional theory hamiltonian. *Nature Communications*, 14:2848, 2023.

Qiangqiang Gu, Zhanghao Zhouyin, Shishir Kumar Pandey, Peng Zhang, Linfeng Zhang, and Weinan E. Deep learning tight-binding approach for large-scale electronic simulations at finite temperatures with ab initio accuracy. *Nature Communications*, 15:6772, 2024.

Suman Hazra, Urvesh Patil, and Stefano Sanvito. Predicting the one-particle density matrix with machine learning. *Journal of Chemical Theory and Computation*, 20(11):4569–4578, 2024.

Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical Review*, 136(3B):B864, 1964.

Anubhav Jain, Yongwoo Shin, and Kristin A Persson. Computational predictions of energy materials using density functional theory. *Nature Reviews Materials*, 1:15004, 2016.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=rs8Sh2UASt.

Peter Bjørn Jørgensen and Arghya Bhowmik. Equivariant graph neural networks for fast electron density estimation of molecules, liquids, and solids. *npj Computational Materials*, 8:183, 2022.

Nayoung Kim, Seongsu Kim, Minsu Kim, Jinkyoo Park, and Sungsoo Ahn. MOFFlow: Flow matching for structure prediction of metal-organic frameworks. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=dNT3abOsLo.

Seongsu Kim and Sungsoo Ahn. Gaussian plane-wave neural operator for electron density estimation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=TvoG41N1Y3.

Seongsu Kim, Nayoung Kim, Dongwoo Kim, and Sungsoo Ahn. High-order equivariant flow matching for density functional theory hamiltonian prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=iFIjNXb0Y5.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

James Kirkpatrick, Brendan McMorrow, David HP Turban, Alexander L Gaunt, James S Spencer, Alexander GDG Matthews, Annette Obika, Louis Thiry, Meire Fortunato, David Pfau, et al. Pushing the frontiers of density functionals by solving the fractional electron problem. *Science*, 374 (6573):1385–1389, 2021.

Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A):A1133, 1965.

Teddy Koker, Keegan Quigley, Eric Taw, Kevin Tibbetts, and Lin Li. Higher-order equivariant neural networks for charge density prediction in materials. *npj Computational Materials*, 10:161, 2024.

G. Kresse and J. Furthmüller. Efficient iterative schemes for $abinitio$ total-energy calculations using a plane-wave basis set. *Physical Review B*, 54:11169–11186, 1996.

Chengteh Lee, Weitao Yang, and Robert G. Parr. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B*, 37:785–789, Jan 1988. doi: 10.1103/PhysRevB.37.785. URL https://link.aps.org/doi/10.1103/PhysRevB.37.785.

He Li, Zun Wang, Nianlong Zou, Meng Ye, Runzhang Xu, Xiaoxun Gong, Wenhui Duan, and Yong Xu. Deep-learning density functional theory hamiltonian for efficient ab initio electronic-structure calculation. *Nature Computational Science*, 2(6):367–377, 2022.

Yunyang Li, Zaishuo Xia, Lin Huang, Xinran Wei, Samuel Harshe, Han Yang, Erpai Luo, Zun Wang, Jia Zhang, Chang Liu, Bin Shao, and Mark Gerstein. Enhancing the scalability and applicability of kohn-sham hamiltonians for molecular systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=twEvvkQqPS.

Yi-Lun Liao, Alexander J Hoffman, Sabrina C Shen, Alexandre Duval, Sam Walton Norwood, and Tess Smidt. Equiformerv3: Scaling efficient, expressive, and general se (3)-equivariant graph attention transformers. *arXiv preprint arXiv:2604.09130*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Zhe Liu, Yuyan Ni, Zhichen Pu, Qiming Sun, Siyuan Liu, and Wen Yan. Towards a transferable acceleration method for density functional theory. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=JNuk3yGDKE.

Erpai Luo, Xinran Wei, Lin Huang, Yunyang Li, Han Yang, Zaishuo Xia, Zun Wang, Chang Liu, Bin Shao, and Jia Zhang. Efficient and scalable density functional theory hamiltonian prediction through adaptive sparsity. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=K3lykWhXON.

Nicola Marzari, Andrea Ferretti, and Chris Wolverton. Electronic-structure methods for materials design. *Nature Materials*, 20(6):736–749, 2021.

Benjamin Kurt Miller, Ricky T. Q. Chen, Anuroop Sriram, and Brandon M Wood. FlowMM: Generating materials with riemannian flow matching. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=W4pB7VbzZI.

Joachim Paier, Robin Hirschl, Martijn Marsman, and Georg Kresse. The perdew–burke–ernzerhof exchange-correlation functional applied to the g2-1 test set using a plane-wave basis set. *The Journal of Chemical Physics*, 122(23), 2005. ISSN 1089-7690. doi: 10.1063/1.1926272. URL http://dx.doi.org/10.1063/1.1926272.

Saro Passaro and C Lawrence Zitnick. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. In *International conference on machine learning*, pp. 27420–27438. PMLR, 2023.

John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996. doi: 10.1103/PhysRevLett.77.3865. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.3865.

Zhuoran Qiao, Anders S Christensen, Matthew Welborn, Frederick R Manby, Anima Anandkumar, and Thomas F Miller III. Informing geometric deep learning with electronic interactions to accelerate quantum chemistry. *Proceedings of the National Academy of Sciences*, 119(31): e2205221119, 2022.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Dmitrij Rappoport and Filipp Furche. Property-optimized gaussian basis sets for molecular response calculations. *The Journal of chemical physics*, 133(13), 2010.

Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Xuecheng Shao, Lukas Paetow, Mark E Tuckerman, and Michele Pavanello. Machine learning electronic structure methods based on the one-electron reduced density matrix. *Nature Communications*, 14:6281, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023.

Hannes Stark, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Harmonic self-conditioned flow matching for joint multi-ligand docking and binding site design. In *Forty-first International Conference on Machine Learning*, 2024.

Zechen Tang, He Li, Peize Lin, Xiaoxun Gong, Gan Jin, Lixin He, Hong Jiang, Xinguo Ren, Wenhui Duan, and Yong Xu. A deep equivariant neural network approach for efficient hybrid density functional calculations. *Nature Communications*, 15:8815, 2024.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Haiyang Yu, Zhao Xu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Efficient and equivariant graph networks for predicting quantum hamiltonian. In *International Conference on Machine Learning*, pp. 40412–40424. PMLR, 2023.

Haiyang Yu, Yuchao Lin, Xuan Zhang, Xiaofeng Qian, and Shuiwang Ji. Efficient prediction of SO(3)-equivariant hamiltonian matrices via SO(2) local frames. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=f9i6VLHT5v.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, ..., and Shuiwang Ji. Artificial intelligence for science in quantum, atomistic, and continuum systems. *Foundations and Trends® in Machine Learning*, 18(4):385–912, 2025.

Yang Zhong, Hongyu Yu, Mao Su, Xingao Gong, and Hongjun Xiang. Transferable equivariant graph neural networks for the hamiltonians of molecules and solids. *npj Computational Materials*, 9:182, 2023.

Zhanghao Zhouyin, Zixi Gan, Shishir Kumar Pandey, Linfeng Zhang, and Qiangqiang Gu. Learning local equivariant representations for quantum operators. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kpq3IIjUD3.

A GTO Basis Set Details and Coefficient Projection

To train our machine learning model to predict the continuous real-space electron density $\rho(\mathbf{r})$, we project the continuous ground-truth density obtained from density functional theory (DFT) onto a localized, auxiliary basis set of Gaussian-Type Orbitals (GTOs). This section details the mathematical formulation of this basis set, the even-tempered scheme used to generate the radial exponents, and the regularized least-squares projection framework used to extract the target coefficient vectors.

A.1 Even-Tempered Gaussian-Type Orbital (GTO) Basis Set

For an atom i located at coordinate $\mathbf{R}_i$, the real-space density $\rho(\mathbf{r})$ is expanded using a linear combination of atom-centered GTO basis functions $\varphi_\mu(\mathbf{r} - \mathbf{R}_i)$. The basis functions are decomposed into radial and angular components,

$$\varphi_{n,\ell,m}(\mathbf{r} - \mathbf{R}_i) = \mathcal{N}_{n,\ell} |\mathbf{r} - \mathbf{R}_i|^\ell \exp\left(-\alpha_{n,\ell} |\mathbf{r} - \mathbf{R}_i|^2\right) Y_\ell^m \left(\frac{\mathbf{r} - \mathbf{R}_i}{|\mathbf{r} - \mathbf{R}_i|}\right) \tag{10}$$

where Y_ℓ^m represents the real spherical harmonics for angular momentum ℓ and magnetic quantum number $m \in [-\ell, \ell]$, and $\mathcal{N}_{n,\ell}$ is the normalization constant given by

$$\mathcal{N}_{n,\ell} = \sqrt{\frac{2^{2\ell+7/2} \alpha_{n,\ell}^{\ell+3/2}}{(2\ell+1)!! \sqrt{\pi}}} \tag{11}$$

where $(2\ell+1)!!$ denotes the double factorial of $2\ell+1$.

To systematically cover the radial coordinate space, the exponents $\alpha_{n,\ell}$ are parameterized using an even-tempered geometric progression. For each angular momentum channel ℓ up to $\ell_{\max}$, we construct N_ℓ radial shells. The exponents are determined by a base exponent $\alpha_{0,\ell}$ and a scaling factor β_ℓ,

$$\alpha_{n,\ell} = \alpha_{0,\ell} \cdot \beta_\ell^{n-1}, \quad \text{for } n = 1, 2, \ldots, N_\ell \tag{12}$$

The hyperparameters $\{\alpha_{0,\ell}, \beta_\ell, N_\ell\}$ are optimized per element to span both the sharp, core-like densities near the nuclei (large α) and the diffuse, bonding valence densities (small α).

A.2 Element-Specific Hyperparameter Configurations

For each chemical element present in our dataset, Table 6 reports the maximum angular momentum $\ell_{\max}$, the number of radial shells N_ℓ, the boundaries of the even-tempered progression ($\alpha_{\min} = \alpha_{0,\ell}$ and $\alpha_{\max} = \alpha_{0,\ell}\beta_\ell^{N_\ell - 1}$), and the geometric progression scaling factor β_ℓ, all instantiated from an uncontracted `def2-QZVPPD` parent basis with a shared even-tempered scale $\beta = 1.3$. QM9 uses the full table across H, C, N, O, F. MD, whose six trajectories span only H, C, O (Sec. 4.2), uses the corresponding subset. Bond-midpoint virtual nodes (Sec. 3.1) are assigned the oxygen basis.

A.3 L2 Least-Squares Coefficient Projection Mechanics

Let $\varphi_p(\mathbf{r})$ represent a shorthand for the set of all active basis functions centered on all atoms (and virtual bond-midpoint nodes) in the system, indexed by a single flattened index $p = (i, n, \ell, m)$. Incorporating the scale factor $s > 0$ defined in the forward continuous density synthesis operator $\mathcal{O}(\mathbf{c}) = s\,\mathbf{\Phi}\,\mathbf{c}$, the continuous electron density $\rho(\mathbf{r})$ is approximated by the expansion

$$\rho(\mathbf{r}) \approx \tilde{\rho}(\mathbf{r}) = s \sum_p c_p \varphi_p(\mathbf{r}) \tag{13}$$

To obtain the optimal target coefficients c_p corresponding to the ground-truth DFT density $\rho_{\text{DFT}}(\mathbf{r})$, we minimize the L^2 error over the continuous space,

$$E(\mathbf{c}) = \int_{\mathbb{R}^3} \left[\rho_{\text{DFT}}(\mathbf{r}) - s \sum_p c_p \varphi_p(\mathbf{r})\right]^2 d\mathbf{r} \tag{14}$$

Table 6: Even-tempered GTO parameters used for density projection (`def2-QZVPPD` parent basis, global $\beta = 1.3$). $\alpha_{\max} = \alpha_{\min}\beta^{N_\ell - 1}$. MD uses the H/C/O rows. QM9 uses the full table. Virtual nodes use O.

Element	**Channel**	N_ℓ	$\alpha_{\min}$	$\alpha_{\max}$	β
H	s ($\ell = 0$)	23	0.1606	51.59	1.3
	p ($\ell = 1$)	18	0.1643	14.22	1.3
	d ($\ell = 2$)	13	0.1681	3.917	1.3
C	s ($\ell = 0$)	42	0.08996	4224	1.3
	p ($\ell = 1$)	32	0.1169	398.2	1.3
	d ($\ell = 2$)	27	0.1176	107.9	1.3
	f ($\ell = 3$)	18	0.1529	13.22	1.3
	g ($\ell = 4$)	13	0.1538	3.583	1.3
N	s ($\ell = 0$)	42	0.1243	5837	1.3
	p ($\ell = 1$)	32	0.1630	555.2	1.3
	d ($\ell = 2$)	27	0.1641	150.5	1.3
	f ($\ell = 3$)	18	0.2151	18.61	1.3
	g ($\ell = 4$)	13	0.2165	5.044	1.3
O	s ($\ell = 0$)	42	0.1393	6540	1.3
	p ($\ell = 1$)	37	0.1102	1394	1.3
	d ($\ell = 2$)	32	0.08720	297.0	1.3
	f ($\ell = 3$)	23	0.1479	47.51	1.3
	g ($\ell = 4$)	14	0.2509	7.600	1.3
F	s ($\ell = 0$)	42	0.1753	8229	1.3
	p ($\ell = 1$)	37	0.1280	1619	1.3
	d ($\ell = 2$)	33	0.09354	414.2	1.3
	f ($\ell = 3$)	23	0.1728	55.51	1.3
	g ($\ell = 4$)	14	0.3194	9.673	1.3

Setting the partial derivatives with respect to each coefficient to zero, $\frac{\partial E}{\partial c_p} = 0$, yields a system of linear equations, the normal equations,

$$\sum_q S_{pq} c_q = y_p \tag{15}$$

where S_{pq} is the analytic overlap matrix between GTO basis functions,

$$S_{pq} = \int_{\mathbb{R}^3} \varphi_p(\mathbf{r}) \varphi_q(\mathbf{r}) d\mathbf{r} \tag{16}$$

and y_p represents the normalized projection of the true DFT density onto the basis functions, factoring out the scale factor s,

$$y_p = \frac{1}{s} \int_{\mathbb{R}^3} \rho_{\text{DFT}}(\mathbf{r}) \varphi_p(\mathbf{r}) d\mathbf{r} \tag{17}$$

In practice, the ground-truth DFT reference density $\boldsymbol{\rho}^{\text{lab}}$ is represented on a discrete numerical grid $\{(\mathbf{r}_k, w_k)\}$, where $\mathbf{r}_k$ is the coordinate of grid point k and w_k is the corresponding integration weight. Incorporating the diagonal spatial mask matrix $\mathbf{W} = \text{diag}(w_1, w_2, \ldots, w_P)$, the discrete least-squares objective formulated on the grid is expressed as

$$\mathbf{c}^\star = \arg\min_{\mathbf{c}} \left\| \mathbf{W}^{1/2} \left(\boldsymbol{\Phi} \mathbf{c} - \mathbf{y} \right) \right\|_2^2 + \lambda \|\mathbf{c}\|_2^2 + \gamma\, \mathbf{c}^\top \text{diag}(\boldsymbol{\Phi}^\top \boldsymbol{\Phi})\, \mathbf{c} \tag{18}$$

where $\mathbf{y} = \boldsymbol{\rho}^{\text{lab}}/s$, $\boldsymbol{\Phi}_{kj} = \varphi_j(\mathbf{r}_k)$ is the forward GTO evaluation design matrix, and $\lambda = 10^{-6}$ is a Tikhonov regularizer to prevent numerical instabilities caused by basis near-linear dependencies. The auxiliary parameter γ regularizes the coefficients scaling with respect to the continuous basis functions' self-overlap.

Solving this regularized objective directly yields the target coefficients $\mathbf{c}^\star$,

$$\mathbf{c}^\star = \left(\boldsymbol{\Phi}^\top \mathbf{W} \boldsymbol{\Phi} + \lambda \mathbf{I} + \gamma\, \text{diag}(\boldsymbol{\Phi}^\top \boldsymbol{\Phi}) \right)^{-1} \boldsymbol{\Phi}^\top \mathbf{W} \mathbf{y} \tag{19}$$

During preprocessing we set $\lambda = 10^{-6}$ and $\gamma = 0$ (isotropic Tikhonov only). This is numerically stable while leaving a small, well-characterized representation residual.

A.4 Empirical Fitting Floor

Because OrbFlow predicts GTO coefficients rather than grid densities, learned NMAE cannot fall below the reconstruction error of $\mathbf{c}^\star$. We measure this floor by synthesizing $\tilde{\rho} = s\,\mathbf{\Phi}\,\mathbf{c}^\star$ from the stored ridge-fitted coefficients, with no neural network in the loop, and scoring NMAE on the same full probe grid used to evaluate the trained models.

Table 7: GTO density fitting floor versus OrbFlow NMAE (%). The floor is the NMAE of ρ reconstructed from stored ridge ground-truth coefficients with no network. OrbFlow is mean $\pm$ std over three training seeds; the MD macro mean is unweighted over the six molecules.

Dataset	n	**GT floor** $\downarrow$	**OrbFlow** $\downarrow$
QM9	10,000	0.061	0.152 ± 0.001
MD benzene	500	0.399	0.421 ± 0.002
MD ethanol	1,000	0.850	1.078 ± 0.023
MD phenol	1,200	0.469	0.575 ± 0.008
MD resorcinol	1,000	0.515	0.663 ± 0.014
MD ethane	201	0.807	0.871 ± 0.008
MD malonaldehyde	400	0.548	1.177 ± 0.012
MD macro mean	6	0.598	0.798

On QM9 the fitting residual is 0.061% while OrbFlow reaches 0.152%, so the remaining gap is due to coefficient prediction rather than incompleteness of the GTO span in Table 6. Relative to SCDP + VO, which uses the same basis construction and reaches 0.178%, OrbFlow removes 22% of the error remaining above this floor. On MD the gap varies by molecule: benzene and ethane lie close to the basis limit, whereas ethanol and malonaldehyde remain well above it, so compact rigid molecules are already well represented by this expansion while flexible oxygenates leave a larger residual for the model to close.

A.5 Electron Count Conservation

Because a spherical Gaussian of angular momentum ℓ integrates to zero over $\mathbb{R}^3$ for every $\ell > 0$, only the $\ell{=}0$ channel of equation 1 contributes to the total electron count,

$$N_{\mathrm{e}}[\mathbf{c}] = \int_{\mathbb{R}^3} \rho(\mathbf{r})\, d\mathbf{r} = \sum_{i,j:\,\ell_{i,j}=0} c_{i,j,0} \int_{\mathbb{R}^3} \varphi_{\alpha_{i,j},0,0,\mathbf{R}_i}(\mathbf{r})\, d\mathbf{r}, \tag{20}$$

so charge conservation is one linear condition on the $\ell{=}0$ coefficients rather than a property of the whole field. We do not impose it during training, because a hard constraint would couple the equivariant readout across atoms. When the predicted density is used as an SCF initial guess we renormalize it to the PAW valence count $N_{\mathrm{e}}^{\mathrm{val}}$. On the official QM9 test set the closed-form count in equation 20 for OrbFlow differs from $N_{\mathrm{e}}^{\mathrm{val}}$ by $0.014\,e$ on average, or 0.030% of $N_{\mathrm{e}}^{\mathrm{val}}$. The mean valence count on this set is $48.28\,e$, so the absolute miss is a few hundredths of an electron. The learned density already carries nearly the right charge, and the renormalization used for an SCF initial guess only has to correct a 0.030% discrepancy.

B Proof of Prior Rotational Invariance

This section provides the full proof of the proposition stated in Sec. 3.2, that the irrep-isotropic prior $p_0(\mathbf{c}_0)$ is invariant under any rotation $R \in \mathrm{SO}(3)$.

Proof. Because Wigner D-matrices $D^{(\ell)}(R)$ are unitary matrices (satisfying $D^{(\ell)}(R)^\top D^{(\ell)}(R) = \mathbf{I}$), the quadratic form in the Gaussian exponent of each subspace ℓ remains invariant under rotation,

$$\|\mathbf{D}(R)\mathbf{c}_0^{(\ell)}\|^2 = \big(D^{(\ell)}(R)\mathbf{c}_0^{(\ell)}\big)^\top \big(D^{(\ell)}(R)\mathbf{c}_0^{(\ell)}\big) = \mathbf{c}_0^{(\ell)\top} D^{(\ell)}(R)^\top D^{(\ell)}(R)\mathbf{c}_0^{(\ell)} = \|\mathbf{c}_0^{(\ell)}\|^2. \tag{21}$$

Thus, the density function is unchanged under any arbitrary 3D rotation, satisfying global $\mathrm{SO}(3)$-invariance. □

C Implementation Details

This appendix consolidates the model architecture, target construction, and optimization procedure that support Sec. 3.3. These settings are shared across QM9 and MD except where noted. Per-dataset hyperparameters and training configuration are given separately in Appendix D.1.

C.1 Network Architecture and Coefficient Parameterization

Graph construction. Each molecule is represented as a graph $\mathcal{G}$ whose nodes comprise nuclear coordinates together with bond-midpoint virtual orbitals (`vnode_elem`=8). Node features are initialized as ℓ=0 spherical-tensor embeddings of atomic numbers, with higher-order channels set to zero. Edges are formed within a cutoff radius r_{cut}=6 Å and parameterized by Gaussian radial bases (300 functions, resolution 0.02 Å) together with $\mathrm{SO}(3)$ edge frames used by equivariant attention.

Equivariant backbone. OrbFlow uses an EquiformerV3 encoder. QM9 employs L=8 blocks with $\ell_{\max}$=6, $m_{\max}$=2, and sphere/hidden/edge channel widths 128/256/128. Each MD molecule uses a compact variant with L=3, $\ell_{\max}$=3, $m_{\max}$=2, and widths 96/192/96 (∼13M parameters). In both settings, attention uses 8 heads with attention-hidden width 64. No backbone dropout is applied.

Flow conditioning and readout. The scalar flow time t is injected into the ℓ=0 channel via a sinusoidal embedding. The continuous coefficient state $\mathbf{x}_t$ is introduced at each layer through an equivariant linear map, using EquiformerV3's SO(2)-convolution-based (eSCN-style) equivariant readout for coefficient entanglement (Passaro & Zitnick, 2023), with no separate velocity MLP. This readout mechanism differs from using eSCN as a full alternative backbone. Latent tensors are mapped to per-atom GTO coefficients with multiplicities matched to the underlying basis. Coefficients are then scaled by $1/\sqrt{N_{\mathrm{orb}}}$ prior to density synthesis so that the training target remains consistent with the normalized coefficient representation.

Basis and density synthesis. We use an uncontracted `def2-QZVPPD` GTO basis with weighted even-tempered augmentation at ETB scale β=1.3 and orbital cutoff 5 Å. Densities are synthesized in real space under the linear combination $\rho(\mathbf{r}) = \sum_i c_i\, \phi_i(\mathbf{r})$.

C.2 Flow-Matching Objective

Training follows conditional flow matching (CFM) with an *endpoint* parameterization. The network predicts the terminal coefficient field, and the CFM velocity is implied by the linear bridge. The initial state is sampled from an irrep-structured Gaussian in normalized coefficient space, $c_0^{\mathrm{norm}} = \sigma\varepsilon + \mu$, where ε is drawn via e3nn irrep noise and the scalar center μ is added only to the ℓ=0 (scalar) coefficient dimension, following the standard convention for additive terms in equivariant architectures, since a nonzero mean is not rotation-invariant for any ℓ>0 irrep. All ℓ>0 channels remain zero-mean, consistent with the irrep-isotropic prior described in Sec. 3.2 and the invariance result of Proposition 1. On QM9 we use $(\mu, \sigma) = (2.5\times10^{-3},\ 0.85)$ for the scalar channel. On MD, (μ, σ) is estimated once as the empirical mean and standard deviation of the scalar-channel training ground-truth coefficients for that molecule (Table 8) and held fixed. Electron-count renormalization of the prior is disabled.

Table 8: MD irrep-Gaussian prior parameters, estimated from each molecule's training-set ground-truth coefficients and held fixed during training.

Molecule	μ	σ
Benzene	8.67×10^{-3}	1.82
Ethanol	5.64×10^{-3}	3.04
Phenol	5.04×10^{-3}	1.98
Resorcinol	4.72×10^{-3}	2.02
Ethane	8.44×10^{-3}	3.66
Malonaldehyde	1.79×10^{-2}	2.79

C.3 Optimization Protocol

All production runs use Adam (Kingma & Ba, 2014) with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\varepsilon{=}10^{-8}$, base learning rate 1×10^{-3}, and zero weight decay. The learning rate follows a linear warm-up for the first $1{,}000$ steps (from a factor of 0.2 to 1.0), then a power decay

$$\eta_t \;=\; \eta_0\, \alpha^{t/\beta}, \qquad \alpha{=}0.96, \quad \beta{=}4433.$$

We maintain an exponential moving average (EMA) of parameters with decay 0.995, apply global-norm gradient clipping at 0.5, and train in FP32 under PyTorch Lightning DDP. Checkpoints are selected by the lowest validation density error after Euler integration ($K{=}2$). All production jobs were run on NVIDIA H200 GPUs.

D Experiment Details

This appendix collects the protocols for the four supporting procedures referenced in the main text, the per-dataset training configuration and hyperparameters underlying every reported result, the DFT solver settings used for the SCF acceleration experiments (Sec. 4.3), the inference latency measurement protocol underlying the computational efficiency comparison (Sec. 4.5), and the multipole moment definitions used to evaluate direct physical property prediction (Sec. 4.4).

D.1 Training Configuration

Table 9 summarizes the primary architecture, coefficient representation, flow-matching, and optimization hyperparameters for both datasets. The training-time sampling range $t \sim \mathrm{Unif}[0.01, 0.99]$ excludes both endpoints of the bridge to avoid numerical instability during training, since the endpoint parameterization in Eq. 4 is singular at $t{=}1$ where the input state degenerates to $\mathbf{x}_t = \mathbf{c}^\star$, while at $t{=}0$ the input reduces to the pure noise prior and carries no target information.

QM9. We train on the standard DeepDFT/SCDP partition ($123{,}835$ / 50 / $10{,}000$ train/val/test) and report both the full test set and Split 2 (last $1{,}600$ test molecules) as in the main text. Training uses $P'{=}60{,}000$ randomly subsampled probe points per molecule ($120{,}000$ for validation/test), per-GPU batch size 4 on 8 GPUs (effective batch size 32), Phase-1 length $T_1{=}7.5 \times 10^4$ steps, and Phase-2 continuation to a total of $T{=}4.5 \times 10^5$ steps.

MD. We train a separate model for each of the six trajectories (benzene, ethanol, phenol, resorcinol, ethane, malonaldehyde), using the official per-molecule splits in Table 10. Training, validation, and test all use $P'{=}125{,}000$ probes per configuration, per-GPU batch size 4 on 4 GPUs (effective batch size 16), Phase-1 length $T_1{=}10^4$ steps, and Phase-2 continuation to a total of $T{=}1.1 \times 10^5$ steps.

D.2 DFT Solver Settings

This section provides full plane-wave DFT solver settings for the SCF acceleration experiments in Sec. 4.3. For unseen XC functionals, the evaluation is performed on a fixed test set of 200 molecules. For training-distribution PBE, we expand the test set to 500 molecules because of the lower computational cost.

DFT Calculations and Exchange-Correlation Functionals. All calculations use VASP (Kresse & Furthmüller, 1996) with the projector-augmented wave (PAW) (Blöchl, 1994) method under the PBE exchange-correlation functional(Perdew et al., 1996) (Sec. 4.3 additionally evaluates B3LYP5(Becke, 1993; Lee et al., 1988) and PBE0 (Adamo & Barone, 1999; Paier et al., 2005) for transferability). The kinetic energy for plane-wave basis set is fixed at 400 eV, and the reciprocal-space sampling is restricted to the Γ-point only, consistent with the isolated-molecule (non-periodic) nature of the QM9 evaluation set. The blocked-Davidson algorithm was used for all SCF calculations except those employing B3LYP5, for which a conjugate gradient-based all-band simultaneous optimization scheme was adopted. This choice was made because blocked-Davidson algorithm exhibited poor SCF convergence for B3LYP5, with pronounced oscillations during electronic iterations

Table 9: Primary architecture and optimization hyperparameters. Shared entries are repeated for readability.

Hyperparameter	**QM9**	**MD (per molecule)**
EquiformerV3 backbone		
# blocks L	8	3
Sphere / hidden / edge channels	128 / 256 / 128	96 / 192 / 96
Attention hidden / # heads	64 / 8	64 / 8
$\ell_{\max}$ ($m_{\max}$)	6 (2)	3 (2)
Cutoff r_{cut} / # radial bases	6 Å / 300	6 Å / 300
Approx. # parameters	$\sim$99M	$\sim$13M
Coefficient / density representation		
GTO basis / ETB β	`def2-QZVPPD` / 1.3	`def2-QZVPPD` / 1.3
GT projection ridge	10^{-6}	10^{-6}
Virtual orbitals	bond midpoints	bond midpoints
Probe subsample P' (train / val)	6×10^4 / 1.2×10^5	1.25×10^5 / 1.25×10^5
Flow matching		
Bridge / loss parameterization	CFM / endpoint	CFM / endpoint
Prior	irrep-Gaussian	irrep-Gaussian
Prior (μ, σ)	$(2.5\times10^{-3}, 0.85)$	Table 8
Train t range / integrate grid	$[0.01, 0.99]$ / uniform	$[0.01, 0.99]$ / uniform
Euler steps K (train / eval)	2 / 2	2 / 2
Optimization		
Optimizer / base LR / weight decay	Adam / 10^{-3} / 0	Adam / 10^{-3} / 0
(β_1, β_2) / warm-up steps	$(0.9, 0.999)$ / 10^3	$(0.9, 0.999)$ / 10^3
LR decay	power (α=0.96, β=4433)	power (α=0.96, β=4433)
EMA decay / grad clip / precision	0.995 / 0.5 / FP32	0.995 / 0.5 / FP32
Per-GPU batch / # GPUs (eff. batch)	4 / 8 (32)	4 / 4 (16)
Phase 1 $\rightarrow$ total steps	$7.5\times10^4 \rightarrow 4.5\times10^5$	$10^4 \rightarrow 1.1\times10^5$
Random seeds	0, 42, 123	0, 42, 505

Table 10: MD trajectory split sizes used in our experiments.

Molecule	Train	Validation	Test
Benzene	1001	500	500
Ethanol	1002	1000	1000
Phenol	1001	1200	1200
Resorcinol	1004	1000	1000
Ethane	2000	201	201
Malonaldehyde	2000	400	400

that often hindered convergence. The all-band optimization scheme provided more robust and stable convergence in these cases.

Convergence Criterion. SCF convergence is reached when the change in total energy between consecutive iterations falls below 10^{-5} eV, matching the convergence criterion used to report RIC and RTR in Sec. 4.3.

Cell Construction. Molecules are placed in an orthorhombic simulation cell with a minimum of 2 Å of vacuum padding on each side of the cell to prevent spurious periodic self-interaction, corresponding to a minimum distance of 4 Å between the atoms from periodic images, same as the original charge density data set (Jørgensen & Bhowmik, 2022). The real-space FFT grid resolution is fixed at approximately 0.15 Å per voxel along each cartesian direction across all systems, ensuring consistent grid density independent of molecule size.

Initialization Protocol. For each test molecule, we run two matched SCF calculations that differ only in initial charge density, an atomic superposition baseline (SAD initialization) and initialization from the OrbFlow-predicted density, re-normalized to conserve total valence electron count and written directly as the solver's input charge density file. All other solver settings, including k-point sampling, cutoff, and convergence threshold, are held identical between the two runs to ensure that any difference in required SCF iterations is attributable solely to the quality of the initial density guess.

Reported Metrics. For each molecule, we record the number of SCF iterations required to reach convergence from each initialization and compute the Relative Iteration Count $\mathrm{RIC} = N_{\mathrm{pred}}/N_{\mathrm{SAD}} \times 100\%$. The Relative Time Ratio additionally accounts for OrbFlow's own inference latency, ensuring that reported acceleration reflects genuine end-to-end speedup rather than SCF iteration count alone.

D.3 Why Iteration Savings Only Partially Translate to Wall-Clock Savings

A DFT run costs more than its SCF iterations. Basis and PAW projector construction, FFT grid setup, symmetry analysis, and file I/O are incurred once per calculation and are unaffected by the quality of the initial density, so a reduction in iteration count can never reduce total runtime proportionally. Writing the total time of a run as a fixed overhead plus a per-iteration cost, $T = T_{\mathrm{fix}} + N\, t_{\mathrm{iter}}$, and letting $f = T_{\mathrm{fix}}/T_{\mathrm{SAD}}$ denote the non-iterative share of a baseline run, the two reported metrics are related by

$$\mathrm{RTR} = f + (1 - f)\,\mathrm{RIC}. \tag{22}$$

Solving equation 22 for f at each functional in Table 3 gives $f = 77.0\%$ for PBE, 26.5% for B3LYP5, and 10.5% for PBE0. Substituting these back reproduces all three reported RTR values exactly. PBE iterations on molecules of this size are cheap, so fixed setup dominates and even a 46% reduction in iterations buys only about 10% of wall-clock time. Hybrid functionals evaluate exact exchange at every iteration, which makes the iterative part far more expensive, shrinks f, and brings RTR close to RIC. Since $\mathrm{RTR} \to \mathrm{RIC}$ as $f \to 0$, the wall-clock benefit of a learned initial density grows with the cost of the underlying method, and the PBE row understates what the same iteration savings are worth at higher levels of theory or on larger systems. Note that T_{pred} includes OrbFlow's own inference latency, so these values of f are mild overestimates.

D.4 Multipole Moments

This section gives the full definitions of the dipole and quadrupole moments evaluated in Sec. 4.4, and supplements the aggregate accuracy reported there (Table 11) with a per-molecule view.

Because the PAW reference densities contain only valence electrons, we pair them with the valence (pseudo-ion) charges Z_A^{val}, so that the total system is charge neutral. The molecular electric dipole moment is then obtained via

$$\boldsymbol{\mu} = \sum_A Z_A^{\mathrm{val}} \mathbf{R}_A - \int \mathbf{r}\, \boldsymbol{\rho}(\mathbf{r})\, d\mathbf{r}, \tag{23}$$

and the traceless electric quadrupole moment tensor is obtained analogously via

$$Q_{ab} = \sum_A Z_A^{\mathrm{val}} \left(3R_{A,a}R_{A,b} - |\mathbf{R}_A|^2 \delta_{ab}\right) - \int \left(3r_a r_b - |\mathbf{r}|^2 \delta_{ab}\right) \boldsymbol{\rho}(\mathbf{r})\, d\mathbf{r}, \tag{24}$$

where $a, b \in \{x, y, z\}$. Both $\boldsymbol{\mu}$ and Q_{ab} are therefore fully determined by $\boldsymbol{\rho}$ once the nuclear geometry is fixed.

Table 11: Dipole and quadrupole MAE (OrbFlow vs. DFT) on held-out QM9. Relative MAE is normalized by the mean DFT-reference magnitude on the same test set.

Quantity	**MAE**	**Rel. MAE [%]**
Dipole, μ_x, μ_y, μ_z	0.026 D	1.7
Dipole magnitude, $\lvert\boldsymbol{\mu}\rvert$	0.031 D	1.0
Quadrupole, Q_{ab}	0.044 B	2.5
Quadrupole norm, $\lVert Q\rVert_F$	0.066 B	1.1

Figure 3 shows the per-molecule prediction accuracy for both moments across the held-out QM9 test set, confirming that the aggregate accuracy in Table 11 is uniform across molecules rather than driven by a favorable average over a subset of poorly-predicted outliers.

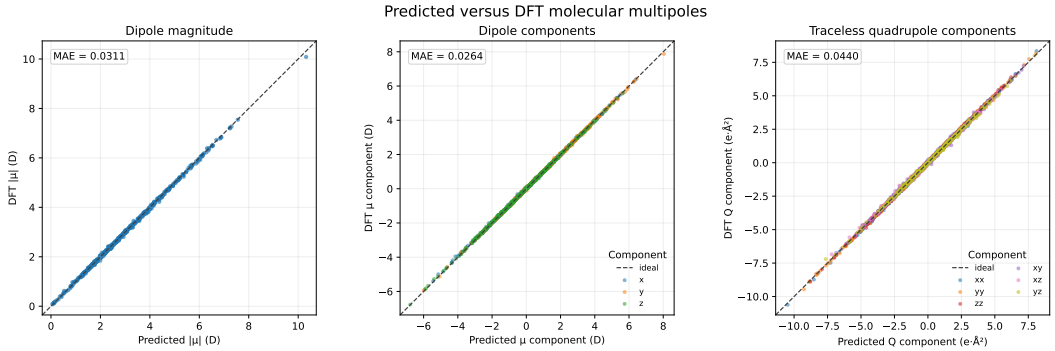


Figure 3: Predicted (OrbFlow) vs. DFT-reference molecular multipoles on held-out QM9 (no SCF), computed via equation 23 and equation 24. Left: dipole magnitude $|\boldsymbol{\mu}|$. Center: individual dipole components μ_x, μ_y, μ_z. Right: individual traceless quadrupole components Q_{ab}. The dashed line is $y = x$. Points close to this line indicate accurate direct property prediction, and the reported MAE in each panel matches the corresponding row of Table 11.

D.5 Efficiency Measurement Protocol

This section details the measurement protocol underlying the inference latency comparison in Sec. 4.5.

Protocol. All timings are measured on a single NVIDIA H200 GPU with batch size 1. We report the mean over the six MD test molecules (Sec. 4.2) and over three independent timing repeats of the same checkpoints, where each repeat covers the full test split after a short CUDA warm-up of three graphs. The network cost is the wall-clock time to produce final GTO coefficients from the molecular graph, which for OrbFlow includes the full K-step Euler rollout in equation 6. The density synthesis cost is the cost of the operator $\mathcal{O}(\cdot)$ in equation 2 mapping coefficients to the real-space grid. OrbFlow evaluates this operator with a fused Triton reduction of the same GTO contraction used by SCDP.

Table 12: Per-molecule inference latency on the MD benchmark, averaged across the six molecules and three independent H200 timing repeats (mean $\pm$ std).

Model	**Network Cost [s]**	**Density Synthesis Cost [s]**	**Total [s]**
ELECTRA	0.058 ± 0.003	0.0025 ± 0.0002	0.061 ± 0.003
SCDP	0.0265 ± 0.0004	0.0150 ± 0.0001	0.0414 ± 0.0005
OrbFlow (ours)	0.054 ± 0.001	0.0126 ± 0.0001	0.066 ± 0.001

Scaling with K. Two patterns in Table 12 are worth expanding on. First, OrbFlow's network cost scales close to linearly with K. Its 0.054s cost at two integration steps is roughly twice SCDP's single-shot cost of 0.026s, consistent with a per-step cost close to a single regression forward pass, around 0.027s per step. OrbFlow's inference cost is therefore predictable and directly controllable through the choice of K (Sec. 4.6) rather than compounding unpredictably. Second, the density synthesis cost differs substantially between ELECTRA (0.003s) and the two GTO-based models, SCDP (0.015s) and OrbFlow (0.013s). This gap reflects the underlying density representation, a 3D Gaussian mixture evaluated with a fused KeOps reduction for ELECTRA against the auxiliary GTO basis expansion shared by SCDP and OrbFlow, rather than the flow matching formulation itself.

E Extended Ablation Results

This appendix expands on the ablations summarized in Sec. 4.6, covering the deterministic two-step baseline, training compute cost, the training-time rollout horizon, and the inference-time integration budget.

E.1 Deterministic Two-Step Baseline

The deterministic baseline of Sec. 4.6 uses the same EquiformerV3 configuration as the QM9 models (Table 9), including the equivariant linear map that injects the coefficient state at each layer, and is trained and evaluated on the same 10% QM9 ablation subset as the other ablation rows. The first evaluation receives an all-zero coefficient state and the second receives the output of the first. No noise is sampled, and the continuous time embedding is replaced by a learned two-valued step-index embedding, so that the baseline retains the ability to behave differently on its first and second evaluation. This isolates the stochastic prior and the bridge interpolation, the components specific to the flow formulation, rather than conflating them with the loss of step conditioning. The chain is trained end to end with the real-space MAE of equation 7 under the same step budget as the corresponding OrbFlow run, with identical optimizer, learning-rate schedule, EMA, probe subsampling, and checkpoint selection.

E.2 Curriculum Variants and Training Compute

Table 13 collects every configuration discussed in Sec. 4.6 together with its total training cost. The curriculum is compute-efficient in practice because the Phase-1 warm start is shared across any number of downstream rollout horizons. Of the 74 GPU-h total cost for the full $K{=}2$ curriculum, only 39 GPU-h come from the Phase-2 fine-tuning stage itself, with the remaining 35 GPU-h of Phase-1 cost shared with $K{=}3$. That incremental fine-tuning cost is roughly half that of training the same $K{=}2$ objective from scratch (81 GPU-h), and the saving widens once the shared Phase-1 cost is amortized across rollout horizons.

Table 13: All ablation configurations on the 10% QM9 subset, with total training compute. NMAE is mean $\pm$ std over three seeds. Rows without a compute entry reuse the shared Phase-1 checkpoint and are reported for accuracy only.

Configuration	**NMAE (%)** $\downarrow$	**GPU-h**
Direct regression (single pass)	0.280 ± 0.003	–
Deterministic two-step	0.268 ± 0.003	–
Phase 1 only	0.295 ± 0.016	35
Phase 2 only, $K{=}2$ (no warm start)	0.250 ± 0.027	81
Joint single-stage, $K{=}2$	0.241 ± 0.002	182
Full curriculum, $K{=}2$ (Ours)	$\mathbf{0.240 \pm 0.005}$	**74**
Full curriculum, $K{=}1$	0.296 ± 0.005	–
Full curriculum, $K{=}3$	0.240 ± 0.002	–

E.3 Training-Time Rollout Horizon

Holding the Phase-1 warm start fixed, we vary the rollout budget K used during Phase 2. $K{=}1$ reaches $0.296 \pm 0.005\%$, statistically indistinguishable from Phase-1-only training at $0.295 \pm 0.016\%$. This follows from the endpoint parameterization. A single Euler step starting at $t_0 = 0$ evaluates f_θ at exactly the point $\mathcal{L}_{\text{stage1}}$ already supervises, so fine-tuning restricted to this boundary case adds essentially no signal beyond Phase 1.

$K{=}2$ and $K{=}3$ converge to the same accuracy, $0.240 \pm 0.005\%$ and $0.240 \pm 0.002\%$. Total training compute nevertheless grows monotonically with K, since each additional Euler step adds one more sequential network evaluation per training example during Phase 2, together with the corresponding backward pass. We therefore adopt $K{=}2$ as the smallest budget at which rollout supervision is informative and past which additional steps yield no measurable benefit at increasing cost.

E.4 Inference-Time Integration Budget

Table 14 reports accuracy and relative cost as a function of the number of Euler steps used at inference for our $K{=}2$-trained model, extending to budgets never seen during training. Accuracy is best exactly at the train-matched budget of two steps. Using more steps than trained does not recover accuracy but degrades it, from 0.240% to 0.539% at four steps, after which the error plateaus in the 0.52–0.54% range while relative cost grows to $3.98\times$ at sixteen steps. The train-matched setting is therefore both the accuracy-optimal and the most compute-efficient inference choice.

Table 14: QM9 NMAE as a function of the number of Euler integration steps used at inference, for a model trained with $K{=}2$. NMAE is mean $\pm$ std over three training seeds (best-val checkpoint). Relative wall-clock inference cost is normalized to the train-matched budget (t_K/t_2), averaged over seeds.

Inference Steps	**2 (train-matched)**	**4**	**8**	**16**
NMAE (%) $\downarrow$	$\mathbf{0.240 \pm 0.005}$	0.539 ± 0.033	0.519 ± 0.044	0.516 ± 0.053
Relative Inference Cost	$1\times$	$1.36\times$	$2.18\times$	$3.98\times$

This lack of generalization beyond the trained horizon follows directly from how Phase 2 supervises the model, and is worth making explicit. Under the endpoint parameterization of equation 4, $\mathbf{v}_\theta$ is derived algebraically from a predicted clean endpoint $f_\theta(\mathcal{G}, \mathbf{x}_t, t)$, and Phase 2 supervises that endpoint prediction only at the two discretization points $t \in \{0, 0.5\}$ visited by the $K{=}2$ rollout. The network therefore receives no training signal at the intermediate timesteps an integrator with $K > 2$ would query, nor at the coarser single interpolation step $K{=}1$ would require. The plateau in Table 14 is consistent with this account. Rather than learning a smooth field valid under arbitrary discretizations, Phase 2 sharpens f_θ into an accurate two-point discrete map, and additional Euler steps subdivide the trajectory into intervals the endpoint predictor was never asked to be accurate on. Removing this restriction, for instance by conditioning on the step size as in shortcut models (Frans et al., 2025), is the clearest direction for future work.