Transferable SCF-Acceleration through Solver-Aligned Initialization Learning

Eike S. Eberhard[*1,2,3], **Viktor Kotsev**[*1,2], **Timm Güthle**[1], **Stephan Günnemann**[1,2,3]

{e.eberhard, viktor.kotsev}@tum.de

[1]Technical University of Munich (TUM) [2]Munich Data Science Institute (MDSI)
[3]Munich Center for Machine Learning (MCML)

Abstract

The cost of Kohn-Sham density functional theory (KS-DFT) calculations scales with the number of required solver iterations, which depend on the quality of the initial guess. Machine learning methods that predict initial guesses from molecular geometry can reduce this cost, but matrix-prediction models fail when extrapolating to larger molecules, degrading rather than accelerating convergence [Liu et al., 2025]. We show that this failure is a supervision problem, not an extrapolation problem: models trained on ground-state targets fit those targets well out of distribution, yet produce initial guesses that slow convergence. Solver-Aligned Initialization Learning (SAIL) resolves this for both Hamiltonian and density matrix models by differentiating through parts of the self-consistent field (SCF) solver end-to-end. We introduce the Effective Relative Iteration Count (ERIC), a correction to the commonly used RIC that accounts for hidden Fock-build overhead. On QM40, which contains molecules up to $4\times$ larger than the training distribution, SAIL reduces ERIC by 40% (PBE), 35% (SCAN), and 32% (B3LYP), more than doubling the previous state-of-the-art reduction on B3LYP. On QMugs molecules $10\times$ larger than the training set, SAIL delivers a $1.35\times$ wall-time speedup at the hybrid level of theory, extending ML SCF acceleration to large drug-like molecules. A reference implementation is available at: `https://github.com/ESEberhard/ML4DFT`.

1 Introduction

Self-consistent field (SCF) solvers are the computational backbone of Hartree-Fock (HF) [Hartree, 1928, Fock, 1930] and KS-DFT calculations [Hohenberg and Kohn, 1964, Kohn and Sham, 1965]. KS-DFT is the most widely used electronic structure method in computational chemistry [Szabo and Ostlund, 1996], reflected in the hundreds of thousands of publications [Haunschild et al., 2019] and the substantial HPC resources devoted to these calculations [Antypas et al., 2014, Zhao et al., 2023]. Any reliable method for speeding up SCF calculations increases resource efficiency and accelerates scientific discovery, both directly and through downstream applications such as machine-learned interatomic potentials (MLIPs), where the cost of generating large reference datasets remains a bottleneck [Kulichenko et al., 2024]. Moreover, unlike machine-learned surrogates that aim to replace DFT entirely, solver acceleration preserves accuracy, allowing practitioners to choose the level of theory *and* its **known limitations** rather than adopt those of a yet-to-be-characterized surrogate.

In recent years, ML initialization methods have been proposed to accelerate SCF calculations [Koker et al., 2024, Yu et al., 2024, Febrer et al., 2025]. These methods produce initial guesses conditioned on the molecular point cloud and typically fall into one of three categories. We jointly refer to Hamiltonian and density matrix prediction as **matrix-based methods**, as both parameterize quadratic

[*]Equal contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

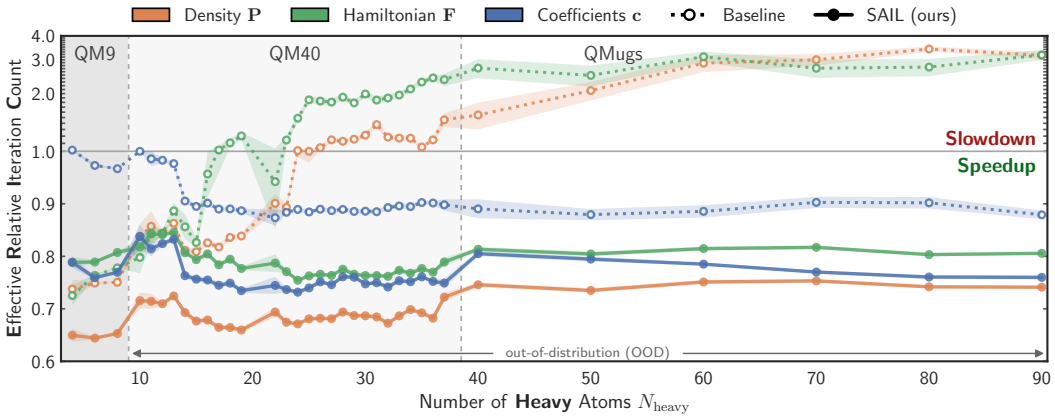


Figure 1: **Size extrapolation of SAIL on B3LYP / def2-SVP.** ERIC of initial-guess models trained on QM9 and evaluated on QM9, QM40, and QMugs, with QMugs molecules up to $10\times$ larger than those seen during training. ERIC < 1 indicates speedup over the traditional initial guess, ERIC > 1 slowdown. Matrix-models trained on ground-state targets degrade out of distribution, whereas SAIL maintains a $\sim 25\%$ speedup across the full range. The coefficient model is robust out of distribution, yet every SAIL-trained model (matrix variants included) outperforms it across the full range.

basis-set expansions with coefficient matrices $\mathbf{X} \in \mathbb{R}^{B\times B}$. The third category, the **coefficient-based** ansatz, predicts the coefficients $\mathbf{c} \in \mathbb{R}^{B_{\text{aux}}}$ of a linear auxiliary basis-set expansion. Liu et al. [2025] have shown that matrix-based approaches *decelerate* SCF calculations when **extrapolating to larger molecules**, a groundbreaking observation given that prior work on matrix-based methods suggested robust out-of-distribution (OOD) performance based on, in hindsight, insufficient extrapolation benchmarks. They therefore proposed adopting coefficient-based approaches [Song and Feng, 2024], which are, by default, limited to local (GGA-type) functionals. Unfortunately, this level of theory is insufficient for many practical applications, particularly in organic chemistry, which is dominated by non-local (range-separated hybrid) functionals [Isert et al., 2021, Eastman et al., 2023, Madushanka et al., 2024, Levine et al., 2025].

In this work, we show that the currently prevailing approach of training initial-guess models by fitting ground-state quantities is suboptimal. We address this with **Solver-Aligned Initialization Learning (SAIL)**, which backpropagates through the SCF algorithm to the ML initial guess, training on solver dynamics rather than ground-state references. This procedure is label-free and requires only molecular geometries. We show that SAIL cures the size-extrapolation issues for matrix-based methods and reveal a **misalignment between ground-state supervision and initial-guess quality**. For the extension of coefficient-based models to non-local functionals proposed by Liu et al. [2025], the constant (non-learnable) treatment of the non-local Hamiltonian contribution makes ground-state supervision suboptimal by construction, *even in distribution*. Training with SAIL enables the model to learn a systematic density error compensating for this fixed approximation.

2 Background

KS-DFT describes the quantum mechanical ground state in terms of its electron density $\rho^* \in \mathcal{D}_{N_e} = \{\mathbb{R}^3 \to [0,\infty) | \int_{\mathbb{R}^3} \rho(\mathbf{r})\, d\mathbf{r} = N_e\}$ and energy $E^* \in \mathbb{R}$ to determine most chemical properties of interest. Molecular DFT codes expand ρ in a finite set $\{\chi_\mu\}_{\mu=1}^{B}$ of atom-centered basis functions

$$\rho(\mathbf{r}) = \sum_{\mu\nu}^{B} \chi_\mu(\mathbf{r})\, P_{\mu\nu}\, \chi_\nu(\mathbf{r})\,, \tag{1}$$

where $\mathbf{P} \in \mathbb{R}^{B\times B}$ is the density matrix. In this representation, the ground-state energy functional $E[\rho]\colon \mathcal{D}_{N_e} \to \mathbb{R}$ reduces to a function $E(\mathbf{P})\colon \mathbb{R}^{B\times B} \to \mathbb{R}$, and finding the ground state becomes an energy minimization problem within the space of valid density matrices [Lehtola et al., 2020]. SCF solvers approach this minimization problem iteratively. Each step computes the energy gradient

$\mathbf{F} = \partial E/\partial \mathbf{P}$, the so-called Fock matrix, and obtains an updated density $\mathbf{P}^{(t+1)}$ by solving a generalized eigenvalue problem (Appendix A):

$$\mathbf{P}^{(t+1)} = \mathrm{SCF}(\mathbf{P}^{(t)}) = (\mathcal{M}_{\mathbf{F}\to\mathbf{P}} \circ \mathcal{M}_{\mathbf{P}\to\mathbf{F}})(\mathbf{P}^{(t)}) \,. \tag{2}$$

This map is applied until *self-consistency* is reached, meaning $\mathbf{P}^* \simeq \mathrm{SCF}(\mathbf{P}^*)$. In practice, convergence accelerators augment this fixed-point iteration and drastically alter its dynamics (Appendix A.1), but the quality of the **initial guess** $\mathbf{P}^{(0)}$ remains critical. It determines *both* whether the solver converges *and* how many iterations it requires [Lehtola, 2019]. The standard initialization in most DFT codes is based on the superposition of atomic densities (SAD) [Van Lenthe et al., 2006].

Since wall-time reduction is the ultimate goal, understanding how the computational costs of an SCF step are distributed is essential. The **Fock build** $\mathcal{M}_{\mathbf{P}\to\mathbf{F}}$ dominates the cost (Fig. 2). Its scaling depends on the choice of the exchange-correlation (XC) functional: $\mathcal{O}(B^3)$ for semi-local functionals such as PBE and SCAN, and $\mathcal{O}(B^4)$ for hybrid functionals like B3LYP [Perdew, 2001].

Density fitting lowers the scaling of the electrostatic contribution to the Fock build from $\mathcal{O}(B^4)$ to $\mathcal{O}(B^3)$ by approximating the electron density as $\rho(\mathbf{r}) \approx \sum_{\mu}^{B_{\text{aux}}} c_\mu \chi_\mu^{(\text{aux})}(\mathbf{r})$, where $\mathbf{c} \in \mathbb{R}^{B_{\text{aux}}}$ are the fitting coefficients and $B_{\text{aux}} \ll B^2$ [Whitten, 1973, Vahtras et al., 1993]. Apart from enabling the $\mathcal{O}(B^3)$ scaling of semi-local functionals, this expansion provides a natural interface between equivariant GNNs and the electron density. Since the auxiliary basis consists of equivariant functions centered on individual nuclei, the coefficients $\mathbf{c}$ are directly predictable from node-level representations [Song and Feng, 2024, Liu et al., 2025], unlike the density matrix representation, which couples basis functions across atom pairs.

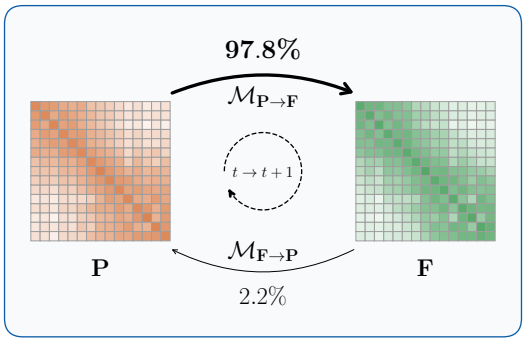


Figure 2: **SCF cycle wall-time analysis** at the B3LYP/def2-SVP level of theory, with jk-density-fitting. Each iteration constructs the Fock matrix $\mathbf{F}^{(t)} = \mathcal{M}_{\mathbf{P}\to\mathbf{F}}(\mathbf{P}^{(t)})$ and updates the density $\mathbf{P}^{(t+1)} = \mathcal{M}_{\mathbf{F}\to\mathbf{P}}(\mathbf{F}^{(t)})$, see Eq. (2). The Fock build dominates at 97.8% of the iteration cost. Cost measured on a QM40 molecule with 30 heavy atoms using GPU4PySCF [Li et al., 2025a].

3 Related Work

Hamiltonian prediction methods are trained to predict the converged Fock matrix[2] $\hat{\mathbf{F}} \approx \mathbf{F}^*$ [Schütt et al., 2019, Yu et al., 2023, 2024, 2025b, Kim et al., 2025, 2026], which can then serve as an SCF initialization $\mathbf{P}^{(0)} = \mathcal{M}_{\mathbf{F}\to\mathbf{P}}(\hat{\mathbf{F}})$. Li et al. [2025b] target the scalability of Hamiltonian prediction to larger molecules and basis sets, introducing an alternative loss (WALoss). Models predicting **density matrices** $\hat{\mathbf{P}} \approx \mathbf{P}^*$ share the same architectural blueprint, only differing in the supervision target [Hazra et al., 2024, Febrer et al., 2025].

Liu et al. [2025] have shown that these matrix-based approaches fail to transfer to larger systems, leading to SCF *deceleration* when used as initial guesses, even when trained with WALoss. To bridge this generalization gap, they suggest **predicting the coefficients** $\mathbf{c} \in \mathbb{R}^{B_{\text{aux}}}$ **of a linear auxiliary basis expansion** [Song and Feng, 2024] of the density $\rho(r) = \sum_\mu c_\mu \chi_\mu^{(\text{aux})}(r)$, as a better learning target. They have demonstrated that this approach extrapolates well to larger molecules, achieving an RIC reduction of 33% and a wall-time speedup of 23% on PBE. However, the auxiliary expansion ansatz is limited to local functionals and cannot represent the density matrix required by hybrid functionals. For hybrid functionals, Liu et al. [2025] fall back to traditional (non-ML) density matrix initialization for the HF-exchange component, resulting in a speedup of 16% in their OOD evaluations on B3LYP. For exchange fractions close to unity (e.g., pure HF), the method is not applicable.

Beyond ML approaches, traditional initial-guess strategies remain relevant baselines. In their systematic assessment of initial guess strategies, Lehtola [2019] note that it is difficult to separate the

[2] In finite-basis-set calculations, the Hamiltonian is commonly referred to as the Fock matrix

effects of the initial guess from those of the SCF dynamics on convergence characteristics. Instead, they proposed using the projection Q of the initial guess orbitals onto the converged ground state $Q(P^{(0)}, P^*) = \sum_{\mu\nu}^{B} \left(\mathbf{P}^{(0)}\mathbf{S}\,\mathbf{P}^*\mathbf{S}\right)_{\mu\nu}$ as a continuous metric. Among the traditional baselines, they found that superposition of atomic potentials (SAP), extended Hückel, and superposition of atomic densities (SAD) performed best, with SAD being the default in most quantum chemistry codes. More recently, Yu et al. [2025a] evaluated basis-set projection and many-body expansion as non-ML initial guess strategies, achieving an average wall-time speedup of less than 3% for B3LYP.

Zhang et al. [2024] exploit the self-consistency condition to train Hamiltonian models without DFT labels, penalizing the Frobenius residual of $\hat{\mathbf{F}} - \mathcal{M}_{\mathbf{P}\to\mathbf{F}}(\mathcal{M}_{\mathbf{F}\to\mathbf{P}}(\hat{\mathbf{F}}))$ and differentiating through the Roothaan-Hall Eq. (11). This residual treats all matrix entries uniformly and penalizes unphysical directions that leave the density unchanged (Appendix A.2) [Eberhard et al., 2026a]. SAIL instead supervises the solver *trajectory*, using an energy gradient loss term (Eq. 5), which directly measures distance to energy stationarity. Zhang and Chan [2022] avoid the initial-guess dependence via the implicit function theorem. We build on the differentiable SCF implementation of Gao et al. [2024], which supports backpropagation through the solver and its convergence accelerator (Appendix A.1), and show that differentiating through a single SCF step ($T = 1$) suffices in practice.

4 Methods

To study the effect of SAIL on different prediction targets, we use the same **backbone** for all approaches. We adapt the NeuralSCF embedding [Song and Feng, 2024], constructing SE(3)-equivariant node features from atomic numbers Z_a and pairwise displacements $\mathbf{r}_{ij}$, and feed them into an EquiformerV2 GNN [Liao et al., 2023] (hyperparameters in Appendix H).

We combine this backbone with **ansatz-dependent readout heads**. For the **coefficient-readout**, we follow Song and Feng [2024] and group the auxiliary basis coefficients c_ν by the atomic number Z_ν of their parent atom and their associated angular momentum l_ν. The node-wise readout $\hat{f}_\nu$ is then elementwise rescaled as $\hat{c}_\nu = \hat{f}_\nu\,\sigma_{(Z_\nu,l_\nu)} + \mu_{(Z_\nu,l_\nu)}$, with mean μ and standard deviation σ computed per group from the training data. Appendix C describes in detail how these predicted auxiliary coefficients are converted into an initial density $\mathbf{P}^{(0)}$. For **matrix prediction**, we adopt the QHNet module [Yu et al., 2023] and predict the target matrix $\mathbf{X} \in \{\mathbf{F}^{(-1)}, \mathbf{P}^{(0)}\}$ block-by-block. Each atom pair (i, j) contributes a submatrix $\mathbf{X}_{ij} \in \mathbb{R}^{B_{Z_i}\times B_{Z_j}}$, built from a self-tensor product of node features (diagonal/node-prediction) or a filtered tensor product of both node features (off-diagonal/edge-prediction). Fock-prediction models additionally require the map $\mathbf{P}^{(0)} = \mathcal{M}_{\mathbf{F}\to\mathbf{P}}(\mathbf{F}^{(-1)})$ to obtain the initial density guess.

The **baseline** models are trained on surrogate ground-state losses, which also serve as the pretraining stage for our fine-tuned SAIL comparisons. We train the coefficient models by minimizing the L^2 density error

$$\mathcal{L}_{\text{coeff}}(\hat{c}, c^*) = \int \left|\rho^*(\mathbf{r}) - \hat{\rho}(\mathbf{r})\right|^2 d\mathbf{r} = \sum_{\mu\nu}^{B_{\text{aux}}} \Delta c_\mu \left(\int \chi_\mu^{(\text{aux})}(\mathbf{r})\,\chi_\nu^{(\text{aux})}(\mathbf{r})\,d\mathbf{r}\right)\Delta c_\nu\,, \tag{3}$$

where $\Delta c_\mu = \hat{c}_\mu - c_\mu^*$. The B_{aux}^2 integrals depend only on the molecular geometry and can be precomputed once per structure. The matrix-based models aim to minimize a mixed Frobenius–L^1 loss on the prediction error $\Delta\mathbf{X} = \hat{\mathbf{X}} - \mathbf{X}^*$ of a matrix target $\mathbf{X} \in \{\mathbf{P}, \mathbf{F}\}$

$$\mathcal{L}_{\mathbf{X}} = \frac{1}{2B}\left(\sqrt{\sum_{\mu\nu}(\Delta X_{\mu\nu})^2} + \sum_{\mu\nu}|\Delta X_{\mu\nu}|\right)\,. \tag{4}$$

Solver-Aligned Initialization Learning (SAIL) differentiates through T SCF cycles end-to-end (Fig. 3), either starting from a pretrained baseline or training directly from scratch in a single label-free stage. The SCF problem minimizes the total energy over the manifold of valid density matrices (Appendix A.2). At a converged solution, the energy is stationary and its gradient $\mathbf{G}^{(t)}$ vanishes [Lehtola et al., 2020]. To reward initial guesses that accelerate convergence, we uniformly

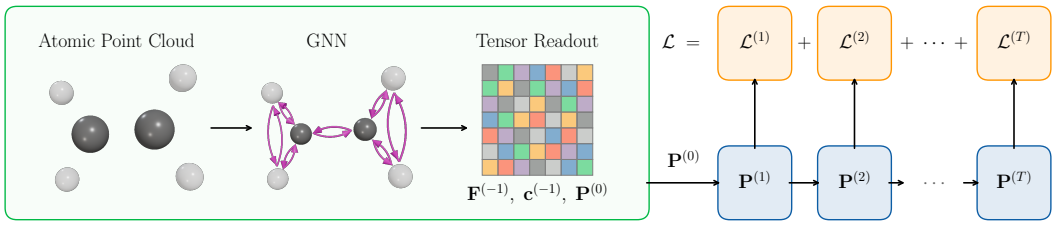


Figure 3: **Illustration of Solver-Aligned Initialization Learning (SAIL).** ML initial-guess models use an SE(3)-equivariant message-passing GNN to map the point cloud of atoms $(\mathbf{Z}, \mathbf{R})$ to an initial tensor that parameterizes a basis-set expansion. If necessary, the output is converted to a density matrix, $\mathbf{P}^{(0)}$, which is used to initialize an SCF calculation. The SCF solver produces a sequence of density matrices $\mathbf{P}^{(0)}, \mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(T)}$, with per-iteration losses $\mathcal{L}^{(t)}$ supervising the trajectory toward self-consistency.

penalize the root-mean-square gradient along the trajectory [Eberhard et al., 2026b]

$$\mathcal{L}_{\text{tot}} = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_{\nabla}^{(t)}, \qquad \mathcal{L}_{\nabla}^{(t)} = \sqrt{\mathbb{E}_{ia}\left[\left|G_{ia}^{(t)}\right|^2\right]}\,. \tag{5}$$

Since the energy error is second order in the density it is only a weak proxy for the distance to the ground state $\nabla E(\mathbf{P}^*) = \mathbf{0}$, so $\Delta E = \mathcal{O}(\|\delta\mathbf{P}\|^2)$, whereas its gradient in the manifold of valid densities is first-order $\|\mathbf{G}\| = \mathcal{O}(\|\delta\mathbf{P}\|)$. Additionally, ΔE collapses into a single scalar, so many different $\delta\mathbf{P}$ yield the same energy error while the gradient is itself an $N_{\text{occ}} \times N_{\text{virt}}$ matrix, so minimizing its root-mean-square puts pressure on every mode separately. Energy supervision is therefore informationally collapsed, whereas gradient supervision preserves the full directional structure (Appendix B). Consistent with this reasoning, preliminary experiments with energy-based trajectory losses did not yield reliable SCF acceleration.

To **measure the resulting acceleration**, prior work heavily relies on the Relative Iteration Count (RIC) [Yu et al., 2024, Zhang et al., 2024, Febrer et al., 2025, Kim et al., 2026]

$$\text{RIC} = \frac{\#\text{SCF iterations from learned initialization}}{\#\text{SCF iterations from } \mathbf{P}_{\text{ref}}^{(\mathbf{0})}}\,, \tag{6}$$

where $\mathbf{P}_{\text{ref}}^{(\mathbf{0})}$ is an established (non-ML) reference initialization. However, even when ignoring inference cost, RIC does not capture the relative wall-time speedup. For example, a Δ-learning approach for Hamiltonian prediction requires computing $\mathbf{F}_{\text{base}} = \mathcal{M}_{\mathbf{P}\to\mathbf{F}}(\mathbf{P}_{\text{base}})$, adding the learned residual $\mathbf{F}^{(-1)} = \mathbf{F}_{\text{base}} + \Delta\mathbf{F}_{\text{learned}}$, and solving $\mathbf{P}^{(0)} = \mathcal{M}_{\mathbf{F}\to\mathbf{P}}(\mathbf{F}^{(-1)})$, a complete additional SCF cycle that does not arise for Δ-learning in density matrix space. For coefficient models, we discuss the Fock build procedure in Appendix C. Based on our SCF wall-time benchmarks in Figure 2, we propose the **Effective Relative Iteration Count (ERIC)**

$$\text{ERIC} = \frac{\#\ \textbf{total}\ \text{Fock builds required for SCF with ML-initialization}}{\#\text{Fock builds when starting from } \mathbf{P}_{\text{ref}}^{(\mathbf{0})}}\,, \tag{7}$$

which counts all Fock builds, including those hidden in the initialization. The correction is not negligible: a typical B3LYP calculation on QM9 converges in $\approx$10 cycles from $\mathbf{P}_{\text{MINAO}}$. A Δ-Hamiltonian model that reduces the solver to 6 iterations achieves a 40% RIC reduction, but the hidden Fock build yields an ERIC of $7/10$, only 30%. This gap is comparable to the RIC differences between non-Δ-learned and Δ-learned Hamiltonian models across the literature, including recent flow-matching approaches (Appendix F), suggesting that their apparent gain may largely stem from the uncounted Fock build (see Table 7).

5 Experiments

We evaluate SAIL on QM9 [Ramakrishnan et al., 2014], QM40 [Madushanka et al., 2024], and QMugs [Isert et al., 2021] for PBE, SCAN, and B3LYP, representing a GGA, a meta-GGA, and a hybrid functional [Perdew et al., 1996, Sun et al., 2015, Becke, 1993, Perdew, 2001]. We split QM9 by molecular size (train: ≤ 20, val: 21–22, test: ≥ 23 atoms). To evaluate size extrapolation beyond QM9, we use QM40 and QMugs as far-out-of-distribution test sets. For a head-to-head comparison of the coefficient-based models proposed by Liu et al. [2025], we evaluate their performance when trained directly on semi-local and hybrid functionals using the authors' proposed extensions. For additional evaluation details, see Appendix D. We compare to prior ML methods in Appendix F and to traditional functional-initialization baselines in Appendix E.

Table 1 reports our **far-out-of-distribution size-extrapolation** results on QM40, together with key **design ablations**. Since SAIL fine-tuning starts from the baseline parameters, we can directly compare surrogate losses before and after fine-tuning. The matrix-based baselines decelerate SCF convergence out of distribution (ERIC > 1), confirming the failure mode reported by Liu et al. [2025]. **The baseline failures are due to misalignment, not extrapolation.** The surrogate losses in Table 1 stay small out of distribution, so the baselines *do* extrapolate, just not the right quantity. Matrix models overfit ground-state targets in directions that decelerate the solver, which is why **SAIL restores acceleration (ERIC < 1) for both matrix-based ansätzes across all three functionals**. The coefficient model separates into two categories. On PBE, it already extrapolates after ground-state training (ERIC 0.65) and shows only minor misalignment, suggesting that the restricted expressivity of the linear auxiliary expansion acts as an implicit regularization. On SCAN and B3LYP, extending the coefficient ansatz beyond GGA requires Fock-build substitutions (Appendix C), since the density alone does not determine the kinetic-energy density τ or the exchange matrix $\mathbf{K}$. Because the ansatz is misspecified, the ground-state coefficient target is a poor learning objective (degrading the baseline to ERIC 0.85 and 0.90, respectively), whereas SAIL lets the model absorb these systematic Fock-build errors into a compensating adjustment that drives the auxiliary surrogate loss up by orders of magnitude but the ERIC down.

Figure 4 shows ERIC as a function of heavy-atom count on far-out-of-distribution molecules from QM40. Without SAIL, matrix-based models degrade with increasing molecule size, confirming the failure mode reported by Liu et al. [2025]. **After SAIL fine-tuning, the ERIC remains flat across all three ansätzes and functionals, with no upward trend even at four times the training-distribution size** (up to 9 heavy vs. up to 37 heavy). The coefficient-based model also benefits from SAIL on SCAN and B3LYP and is on par on PBE (Figure 4). For B3LYP, we extend our size extrapolation experiment to QMugs (Figure 1). SAIL maintains a mean ERIC of 0.74 (density matrix prediction) with **no upward trend even for molecules $10\times$ larger than the training distribution**. **The B3LYP wall-time measurements** for matrix-based methods (Figure 5) show that these **ERIC**

Table 1: **SCF acceleration of baseline and SAIL (fine-tuned) models** across input representations and functionals, evaluated on QM40 (out-of-distribution). Grey rows show ablation variants. **Bold**: best RIC/ERIC per functional across all input representations.

Model		Functional	Ablation	Surrogate Loss (↓)		RIC (↓)		ERIC (↓)	
				base	SAIL	base	SAIL	base	SAIL
Coefficients	**c**	PBE		0.003	0.015	**0.58**	0.59	0.65	0.66
		SCAN		0.004	17.558	0.77	0.67	0.85	0.74
		B3LYP		0.003	53.815	0.82	**0.68**	0.90	0.76
Density Mat.	**P**	PBE		0.281	1.149	1.05	0.60	1.05	**0.60**
		SCAN		0.250	1.100	0.99	**0.65**	0.99	**0.65**
		"	T=4	"	1.101	"	0.65	"	0.65
		"	T=2	"	1.110	"	0.65	"	0.65
		"	T=1	"	1.145	"	0.65	"	0.65
		"	Single-stage	–	1.270	–	0.66	–	0.66
		B3LYP		0.267	0.976	1.02	**0.68**	1.02	**0.68**
Fock	**F**	PBE		0.005	0.277	1.11	0.66	1.18	0.73
		SCAN		0.017	0.284	1.16	0.69	1.23	0.76
		B3LYP		0.026	0.382	1.47	0.70	1.55	0.78

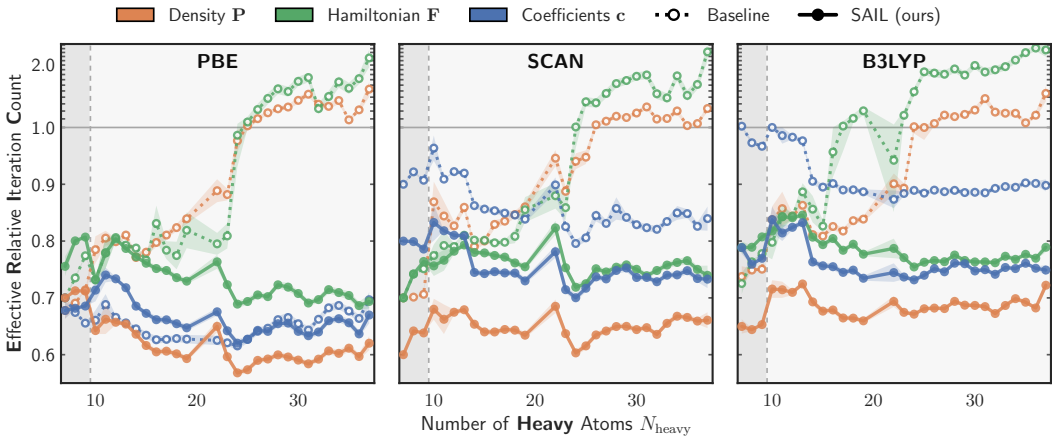


Figure 4: **Effective Relative Iteration Count (ERIC)** for different XC-functionals (from left to right: local, semi-local, hybrid) on far-out-of-distribution molecules, binned by their number of heavy atoms (up to 71 atoms, 37 heavy). All models are **trained on QM9** molecules with at most 20 atoms; we only plot QM9 test molecules with > 23 atoms. The dashed vertical line marks the transition from the QM9 test set to the QM40 test set. For comparison to related work, see Table 6.

reductions translate into comparable speedups, as expected from the Fock-build dominance at the hybrid level (Figure 2), while the per-component breakdown shows that model evaluation is cheap compared to the MINAO initial-guess cost. Additional B3LYP runs with three training seeds show that the SAIL results are reproducible, with seed-to-seed variations of at most 0.01 ERIC, and SAIL- and MINAO-initialized calculations reach the same final energies to within 10^{-7} Ha (Appendix K).

To test whether SAIL requires unrolling the full SCF trajectory or pretraining on ground-state labels, we ablate both components on the SCAN density-matrix model in Table 1. **Truncating backpropagation to a single solver cycle ($T = 1$) matches the ten-cycle default** on QM40 at an ERIC of 0.65, as do $T = 2$ and $T = 4$. At $T = 1$, the gradient still passes through the first SCF update (two Fock builds and one eigensolve) and omits only deeper cycles where DIIS extrapolation acts. Table 2 explains why deeper unrolling adds little. Under DIIS, the per-cycle parameter-gradient norm $\|\mathbf{g}_t\|$ decays by five orders of magnitude by cycle 8, while the direction $\cos(\mathbf{g}_t, \mathbf{g}_1)$ stays aligned with the first cycle across all steps. **Training directly from scratch without ground-state pretraining** (single-stage) reaches an ERIC of 0.66, matching the pretrained model (0.65) and improving over the surrogate-trained baseline (0.99) without any reference DFT labels. Additional ablations of model design choices and training objectives are provided in Appendix I.

Table 2: **Per-cycle parameter gradients** $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\nabla}^{(t)}$ for SCAN density-matrix SAIL on QM40 (60 molecules).

Cycle t	1	2	4	8
$\|\mathbf{g}_t\|$	7e-3	5e-3	2e-4	2e-8
$\cos(\mathbf{g}_t, \mathbf{g}_1)$	1.00	1.00	0.98	0.96

Simultaneous size and basis set generalization. Matrix ansätzes tie predictions to a specific AO basis and require retraining for each. Coefficient models predict in a fixed auxiliary basis and apply unchanged to any compatible target AO basis [Liu et al., 2025]. Table 3 combines size and basis extrapolation. We train on QM9/def2-SVP and evaluate on QM40 using larger basis sets. The baseline coefficient model stays near ERIC 0.90 across the four bases. SAIL reduces ERIC further to 0.82 on the three larger bases. The SAIL gain is smaller outside the training basis set (8 vs. 14 percentage points of the $\mathbf{P}_{\text{MINAO}}$ cost), but SAIL outperforms the baseline at every target.

Table 3: **Simultaneous size and basis set generalization** for coeff. models on B3LYP. Trained on QM9, eval. on QM40.

Basis Set	**RIC** (↓)		**ERIC** (↓)	
	base	SAIL	base	SAIL
def2-SVP (ID)	0.82	0.68	0.90	0.76
def2-TZVP	0.82	**0.74**	0.90	**0.82**
def2-TZVPPD	0.82	**0.74**	0.90	**0.82**
def2-QZVP	0.82	**0.74**	0.90	**0.82**

To test generalization to non-equilibrium geometries we evaluate the robustness of acceleration under geometric distortions for QM9-trained models on 3BPA [Kovács et al., 2021]. This benchmark contains conformations of 3-(benzyloxy)pyridin-2-amine sampled at 300, 600,

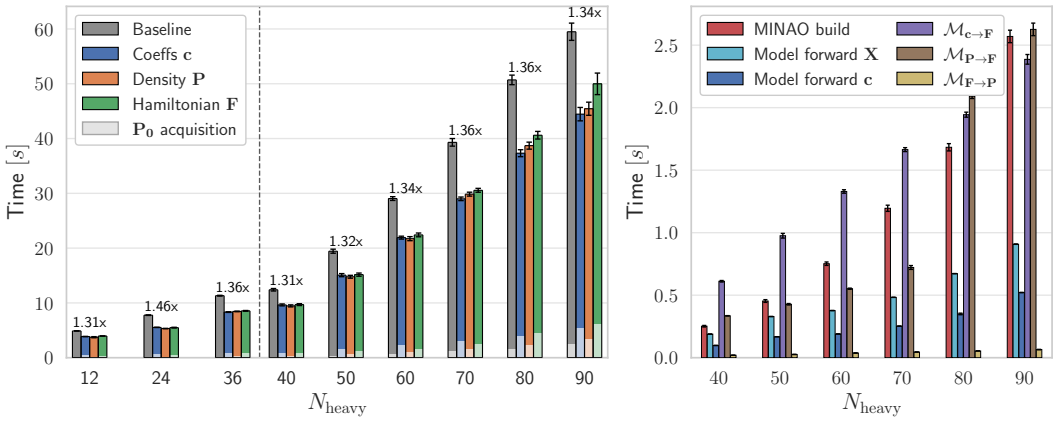


Figure 5: **Left: Wall-time speedup on B3LYP/def2-SVP.** Total SCF wall-time on QM40 (left of dashed line) and QMugs, measured on GPU with GPU4PySCF. Each bar is decomposed into initial-guess acquisition (lighter color) and the subsequent SCF loop. All SAIL-trained models deliver consistent speedups across the full size range, with no degradation on molecules $10\times$ larger than the training distribution. Numbers above the bars report the speedup of the density matrix model, which outperforms the Hamiltonian model at every size. **Right: Initial-guess acquisition breakdown.** Per-component cost of the initial-guess acquisition on QMugs. Matrix-based models use Δ-learning on top of MINAO, so the MINAO build is a fixed cost shared by both ansätzes. The model forward pass adds only a small overhead, while the extra Fock build $\mathcal{M}_{\mathbf{P}\to\mathbf{F}}$ required by Δ-Hamiltonian learning dominates at large sizes, motivating ERIC.

and 1200 K, together with angular dihedral scans. As temperature increases from 300 to 1200 K, its mean ERIC rises modestly from 0.81 to 0.86, whereas the surrogate-trained baseline deteriorates, becoming slower than MINAO. Table 4 shows that **the density-matrix model retains acceleration across all tested temperatures and dihedral scans**. On the dihedral scans, SAIL achieves ERICs of 0.77–0.78, compared with 0.89–0.90 for the baseline. The coefficient model also benefits from SAIL, but its advantage diminishes with temperature, reaching ERIC 0.99 at 1200 K. This contrast supports density-matrix prediction as the more robust choice in these tests. We further evaluate transfer to reactive pathways and transition states from Transition1x [Schreiner et al., 2022] in Appendix J (Tables 10 and 11). SAIL accelerates all frames along an intramolecular hydrogen transfer and through the transition state of a C−C bond dissociation, whereas compressed contact dimers past bond breaking are not accelerated (ERIC = 1.00, likely because mid-dissociation geometries are absent from the training set), though never decelerated.

Table 4: **Transfer to non-equilibrium geometries on 3BPA.** ERIC at B3LYP for models trained on QM9.

Test set	Density P		Coefficients c	
	Base	SAIL	Base	SAIL
300 K	0.94	**0.81**	0.94	0.86
600 K	0.99	**0.84**	0.96	0.91
1200 K	1.08	**0.86**	1.00	0.99
$\beta = 120°$	0.89	**0.77**	0.95	0.86
$\beta = 150°$	0.90	**0.78**	0.95	0.87
$\beta = 180°$	0.90	**0.78**	0.96	0.87

All models up to here have been trained on QM9, which contains only neutral equilibrium molecules across five elements (H, C, N, O, F). To show-case the **robustness of SAIL in a more chemically diverse setting while minimizing up-front costs**, we directly train a density-matrix model on small molecules from SPICEv2 [Eastman et al., 2024] **without pretraining (label-free) for only two epochs, differentiating through a single solver cycle** ($T = 1$). The training set consists of 48,470 structures at B3LYP / def2-SVP level of theory and spans neutral and charged systems across 13 elements (H, B, C, N, O, F, Na, Si, P, S, Cl, K, and Br). Despite this short, single-stage, one-step training, Figure 6 shows that **SAIL accelerates every evaluated chemical category while also extrapolating to larger molecules**. Across all subsets, mean ERIC stays between 0.67 and 0.79 regardless of training share, including zero-shot transfer to solvated systems absent from the training set (0.68). In the size-extrapolation regime, mean ERIC *improves* from 0.75 in distribution ($\leq$ 35 heavy atoms) to 0.67 in the size-extrapolation regime (36-50 heavy atoms). Across all 600 evaluation structures (300 ID and 300 OOD), 599 converge under both initializations, and **SAIL never decelerates a single calculation**, reducing iterations on 596 structures (including all 300 OOD structures), matching MINAO on three, and failing alongside MINAO on one H/N/Si structure.

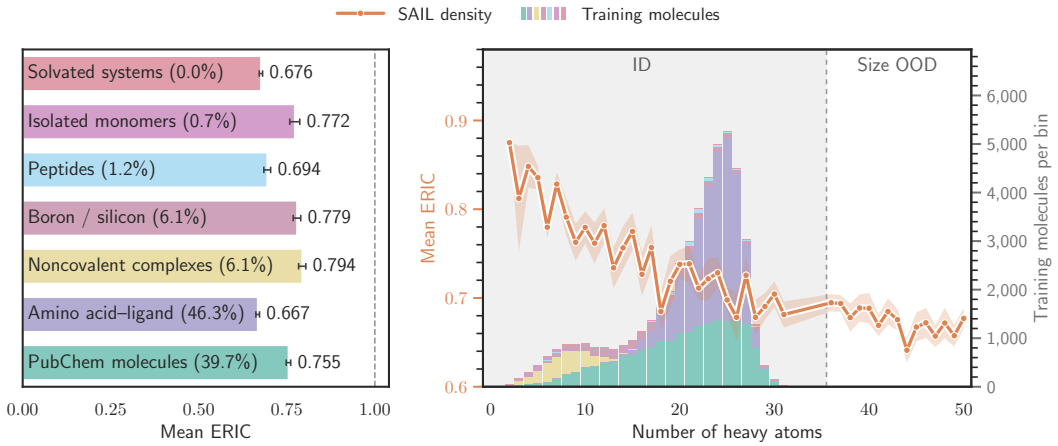


Figure 6: **Chemical coverage and size transfer on SPICEv2 for hybrid-level DFT.** We train a density-matrix model label-free from scratch using the minimal, $T = 1$, version of SAIL without surrogate pretraining. **Left:** Mean ERIC across chemical categories; percentages indicate each category's share of the 48,470 training molecules. **Right:** Mean ERIC versus heavy-atom count (orange, left axis), overlaid on the number of training samples per bin (gray, right axis).

6 Discussion

This work introduces **Solver-Aligned Initialization Learning (SAIL)**, which trains on solver dynamics rather than ground-state targets, and the **Effective Relative Iteration Count (ERIC)**, which corrects RIC for ansatz-dependent Fock-build overhead. We evaluate ML initial-guess architectures across three rungs of Jacob's ladder, a GGA (PBE), a meta-GGA (SCAN), and a hybrid (B3LYP).

On QM9/QM40, we independently confirm the size-extrapolation failure of conventionally trained matrix-based methods reported by Liu et al. [2025]. We show that this failure stems from a **misalignment between ground-state supervision and solver dynamics**. Models trained to minimize a loss with respect to the converged solution do not produce initial guesses that converge fast. SAIL **cures the size-extrapolation failure for matrix methods**, achieving near-identical ERICs on QM40 (**c**: 0.66, **P**: 0.60, **F**: 0.73, Tab. 1) at the PBE/def2-SVP level of theory.

SAIL extends reliable SCF acceleration to mGGA and hybrid functionals. The coefficient and Δ-density matrix ansätzes offer complementary strengths in this regime. The coefficient approach transfers across AO basis sets without retraining (Table 3). It does not apply to pure Hartree–Fock or to functionals with exact-exchange fractions near unity [Liu et al., 2025]. For hybrid functionals, the coefficient ansatz relies on the model to compensate for the fixed exact-exchange treatment by learning a density distortion. Whether this also works on *range-separated* hybrids remains untested. Δ-density matrix prediction requires retraining per basis set. It uses no functional-dependent substitutions and gives the most consistent acceleration across our evaluations, reaching ERIC 0.65 on SCAN (35% reduction over MINAO) and 0.68 on B3LYP (32%). SAIL thus extends reliable ML SCF acceleration to drug-like molecules up to $10\times$ larger than the training distribution at the hybrid level of theory.

The training cost of SAIL amortizes quickly in the label-free, single-stage $T = 1$ setting, since it does not incur the up front labeling cost of surrogate-loss pretraining. On QM9, a $T = 1$ step takes 86 ms per molecule of pure compute on an H200 GPU (14 GPU-hours for five epochs over $\sim$100k structures), and validation curves reach over 90% of the final acceleration within the first epoch ($\sim$3 GPU-hours). On SPICEv2, two epochs over 48,470 molecular groups take 330–370 ms of compute per step ($\sim$9 GPU-hours of compute, or 1.7 days wall-clock including dataloader I/O). In downstream B3LYP evaluations (Table 5), SAIL saves 8 to 42 A100 GPU-hours per 10,000 calculations across molecules of 40 to 90 heavy atoms, so the training investment breaks even after a few thousand to a

Table 5: **NVIDIA-A100 hours saved** per 10,000 B3LYP / def2-SVP calculations for SAIL-trained **P** models.

N_{heavy}	40	50	60	70	80	90
Hours saved	8	13	20	29	37	42

few tens of thousands of large-molecule calculations, a small fraction of a single screening or MLIP dataset generation campaign (e.g., 665k structures in QMugs).

Limitations. Like the baseline methods, SAIL requires a separate model for each functional. Matrix-based ansätzes require retraining per basis set. These costs are amortized over all subsequent calculations at that level of theory. We evaluate three functionals (PBE, SCAN, B3LYP) spanning three rungs of Jacob's ladder. Transfer to range-separated and double hybrids, as well as periodic structures [Li et al., 2022, Gong et al., 2023, Tang et al., 2024], remains untested. Chemical coverage extends to 13 elements up to bromine via SPICEv2, but transition metals, heavier elements, and open-shell systems are not yet covered. Acceleration transfers to non-equilibrium geometries (3BPA, Transition1x), yet compressed contact pairs past bond breaking are not accelerated (ERIC $= 1.00$), suggesting that such geometries need to be represented in training. While SAIL itself is label-free, our strongest results combine it with surrogate-loss pretraining. SAIL training increases total training time by a factor of $\sim 1.7\times$ relative to the baseline, though our ablations suggest that truncating the number of SCF steps during training can reduce this overhead substantially.

ERIC improves on RIC as a wall-time proxy but does not replace wall-time measurements. Wall-time depends on hardware, XC-functional, basis set, and implementation. It does not scale linearly with iteration count, as different algorithms and molecule sizes shift the computational bottleneck [Liu et al., 2025]. ML research code rarely matches the optimization of production DFT codes, making absolute wall-time comparisons across methods unreliable. ERIC partially bridges this gap by providing a hardware- and implementation-independent proxy.

Future work. Since SAIL is label-free, data augmentation through chemical perturbations such as H$\leftrightarrow$F substitutions and random atom position perturbations comes at no additional labeling cost. Adapting SAIL to periodic systems and to alternative solvers, such as direct energy minimization and second-order SCF methods [Sun et al., 2017], is a direct extension of the framework presented here. Finally, initializing the GNN backbone from a pretrained force-field foundation model could be a path toward better generalization across chemical space at lower per-functional training cost.

Broader Impact

Solver-Aligned Initialization Learning accelerates an established numerical method (KS-DFT) without altering the underlying physics or accuracy, so it inherits rather than introduces risk profiles relative to conventional DFT. The expected positive impacts are reduced compute and energy cost per calculation at constant accuracy, as well as faster research iteration in domains that rely heavily on DFT, including drug discovery and the generation of reference data for machine-learned interatomic potentials [Kulichenko et al., 2024]. On the negative side, cheaper computations may induce a rebound effect, in which aggregate energy consumption rises because lower per-calculation costs expand the volume of calculations performed. We are not aware of direct dual-use concerns specific to SCF acceleration beyond those already present for DFT itself.

Acknowledgments

We thank Zhe Liu, Siyuan Liu, Qiming Sun, Arghya Bhowmik, and Pol Febrer for insightful discussions regarding their respective methods, and Nicholas Gao for feedback on the final manuscript.

References

K Antypas, B A Austin, T L Butler, R A Gerber, C L Whitney, N J Wright, W Yang, and Z Zhao. NERSC Workload Analysis on Hopper. Technical report, National Energy Research Scientific Computing (NERSC) Center, 2014.

Axel D. Becke. Density-functional thermochemistry. III. The role of exact exchange. *The Journal of Chemical Physics*, 98(7):5648–5652, 1993.

Peter Eastman, Pavan Kumar Behara, David L. Dotson, Raimondas Galvelis, John E. Herr, Josh T. Horton, Yuezhi Mao, John D. Chodera, Benjamin P. Pritchard, Yuanqing Wang, Gianni De Fabritiis, and Thomas E. Markland. SPICE, A Dataset of Drug-like Molecules and Peptides for Training Machine Learning Potentials. *Scientific Data*, 10(1):11, 2023.

Peter Eastman, Benjamin P. Pritchard, John D. Chodera, and Thomas E. Markland. Nutmeg and SPICE: Models and data for biomolecular machine learning. *Journal of Chemical Theory and Computation*, 20(19):8583–8593, 2024. doi: 10.1021/acs.jctc.4c00794.

Eike S. Eberhard, Xaver Kainz, Viktor Kotsev, Abdulrahman Aldossary, and Stephan Günnemann. Physics-Aligned Electronic Ground-State Learning Improves Generalization, 2026a.

Eike S. Eberhard, Luca A. Thiede, Abdul Aldossary, Andreas Burger, Nicholas Gao, Vignesh Bhethanabotla, Alán Aspuru-Guzik, and Stephan Günnemann. Derivative Informed Learning of Exchange-Correlation Functionals. In *Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea*, 2026b.

Pol Febrer, Peter Bjørn Jørgensen, Miguel Pruneda, Alberto García, Pablo Ordejón, and Arghya Bhowmik. Graph2Mat: Universal graph to matrix conversion for electron density prediction. *Machine Learning: Science and Technology*, 6(2):025013, 2025.

V. Fock. Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems. *Zeitschrift für Physik*, 61(1):126–148, 1930.

Nicholas Gao, Eike Eberhard, and Stephan Günnemann. Learning Equivariant Non-Local Electron Density Functionals. In *The Thirteenth International Conference on Learning Representations*, 2024.

Alberto García, Nick Papior, Arsalan Akhtar, Emilio Artacho, Volker Blum, Emanuele Bosoni, Pedro Brandimarte, Mads Brandbyge, J. I. Cerdá, Fabiano Corsetti, Ramón Cuadrado, Vladimir Dikan, Jaime Ferrer, Julian Gale, Pablo García-Fernández, V. M. García-Suárez, Sandra García, Georg Huhs, Sergio Illera, Richard Korytár, Peter Koval, Irina Lebedeva, Lin Lin, Pablo López-Tarifa, Sara G. Mayo, Stephan Mohr, Pablo Ordejón, Andrei Postnikov, Yann Pouillon, Miguel Pruneda, Roberto Robles, Daniel Sánchez-Portal, Jose M. Soler, Rafi Ullah, Victor Wen-zhe Yu, and Javier Junquera. Siesta: Recent developments and applications. *The Journal of Chemical Physics*, 152 (20):204108, 2020.

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(1):2848, 2023.

Tim Gould. A step toward density benchmarking—The energy-relevant "mean field error". *The Journal of Chemical Physics*, 159(20):204111, 2023.

D. R. Hartree. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1):89–110, 1928.

Robin Haunschild, Andreas Barth, and Bernie French. A comprehensive analysis of the history of DFT based on the bibliometric method RPYS. *Journal of Cheminformatics*, 11(1):72, 2019.

S. Hazra, U. Patil, and S. Sanvito. Predicting the One-Particle Density Matrix with Machine Learning. *Journal of Chemical Theory and Computation*, 20(11):4569–4578, 2024.

P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Physical Review*, 136(3B):B864–B871, 1964.

Clemens Isert, Kenneth Atz, José Jiménez-Luna, and Gisbert Schneider. QMugs: Quantum Mechanical Properties of Drug-like Molecules, 2021.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Seongsu Kim, Nayoung Kim, Dongwoo Kim, and Sungsoo Ahn. High-order Equivariant Flow Matching for Density Functional Theory Hamiltonian Prediction, 2025.

Seongsu Kim, Chanhui Lee, Yoonho Kim, Seongjun Yun, Honghui Kim, Nayoung Kim, Changyoung Park, Sehui Han, Sungbin Lim, and Sungsoo Ahn. Machine Learning Hamiltonians are Accurate Energy-Force Predictors, 2026.

W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review*, 140(4A):A1133–A1138, 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(1):161, 2024.

Dávid Péter Kovács, Cas van der Oord, Jiri Kucera, Alice E. A. Allen, Daniel J. Cole, Christoph Ortner, and Gábor Csányi. Linear atomic cluster expansion force fields for organic molecules: Beyond RMSE. *Journal of Chemical Theory and Computation*, 17(12):7696–7711, 2021. doi: 10.1021/acs.jctc.1c00647.

Maksim Kulichenko, Benjamin Nebgen, Nicholas Lubbers, Justin S. Smith, Kipton Barros, Alice E. A. Allen, Adela Habib, Emily Shinkle, Nikita Fedik, Ying Wai Li, Richard A. Messerly, and Sergei Tretiak. Data Generation for Machine Learning Interatomic Potentials and Beyond. *Chemical Reviews*, 124(24):13681–13714, 2024.

Susi Lehtola. Assessment of Initial Guesses for Self-Consistent Field Calculations. Superposition of Atomic Potentials: Simple yet Efficient. *Journal of Chemical Theory and Computation*, 15(3): 1593–1604, 2019.

Susi Lehtola, Frank Blockhuys, and Christian Van Alsenoy. An Overview of Self-Consistent Field Calculations Within Finite Basis Sets. *Molecules*, 25(5):1218, 2020.

Daniel S. Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G. Taylor, Muhammad R. Hasyim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman, Nathan C. Frey, Xiang Fu, Vahe Gharakhanyan, Aditi S. Krishnapriyan, Joshua A. Rackers, Sanjeev Raja, Ammar Rizvi, Andrew S. Rosen, Zachary Ulissi, Santiago Vargas, C. Lawrence Zitnick, Samuel M. Blau, and Brandon M. Wood. The Open Molecules 2025 (OMol25) Dataset, Evaluations, and Models, 2025.

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.

Rui Li, Qiming Sun, Xing Zhang, and Garnet Kin-Lic Chan. Introducing GPU Acceleration into the Python-Based Simulations of Chemistry Framework. *The Journal of Physical Chemistry A*, 129 (5):1459–1468, 2025a.

Yunyang Li, Zaishuo Xia, Lin Huang, Xinran Wei, Han Yang, Sam Harshe, Zun Wang, Chang Liu, Jia Zhang, Bin Shao, and Mark B. Gerstein. Enhancing the Scalability and Applicability of Kohn-Sham Hamiltonians for Molecular Systems, 2025b.

Yi-Lun Liao, Brandon M. Wood, Abhishek Das, and Tess Smidt. EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhe Liu, Yuyan Ni, Zhichen Pu, Qiming Sun, Siyuan Liu, and Wen Yan. Towards A Universally Transferable Acceleration Method for Density Functional Theory, 2025.

Ayesh Madushanka, Renaldo T. Moura, and Elfi Kraka. QM40, Realistic Quantum Mechanical Dataset for Machine Learning in Molecular Science. *Scientific Data*, 11(1):1376, 2024.

John P. Perdew. Jacob's ladder of density functional approximations for the exchange-correlation energy. In *AIP Conference Proceedings*, volume 577, pages 1–20, Antwerp (Belgium), 2001. AIP.

John P. Perdew and Lucian A. Constantin. Laplacian-level density functionals for the kinetic energy density and exchange-correlation energy. *Physical Review B*, 75(15):155109, 2007.

John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized Gradient Approximation Made Simple. *Physical Review Letters*, 77(18):3865–3868, 1996.

P. Pulay. Improved SCF convergence acceleration. *Journal of Computational Chemistry*, 3(4): 556–560, 1982.

Péter Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical Physics Letters*, 73(2):393–398, 1980.

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

C. C. J. Roothaan. New Developments in Molecular Orbital Theory. *Reviews of Modern Physics*, 23 (2):69–89, 1951.

Mathias Schreiner, Arghya Bhowmik, Tejs Vegge, Jonas Busk, and Ole Winther. Transition1x - a dataset for building generalizable reactive machine learning potentials. *Scientific Data*, 9(1):779, 2022. doi: 10.1038/s41597-022-01870-w.

K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nature Communications*, 10(1):5024, 2019.

Feitong Song and Ji Feng. NeuralSCF: Neural network self-consistent fields for density functional theory, 2024.

Jianwei Sun, Adrienn Ruzsinszky, and John P. Perdew. Strongly Constrained and Appropriately Normed Semilocal Density Functional. *Physical Review Letters*, 115(3):036402, 2015.

Qiming Sun, Jun Yang, and Garnet Kin-Lic Chan. A general second order complete active space self-consistent-field solver for large-scale systems. *Chemical Physics Letters*, 683:291–299, 2017.

Qiming Sun, Xing Zhang, Samragni Banerjee, Peng Bao, Marc Barbry, Nick S. Blunt, Nikolay A. Bogdanov, George H. Booth, Jia Chen, Zhi-Hao Cui, Janus J. Eriksen, Yang Gao, Sheng Guo, Jan Hermann, Matthew R. Hermes, Kevin Koh, Peter Koval, Susi Lehtola, Zhendong Li, Junzi Liu, Narbe Mardirossian, James D. McClain, Mario Motta, Bastien Mussard, Hung Q. Pham, Artem Pulkin, Wirawan Purwanto, Paul J. Robinson, Enrico Ronca, Elvira R. Sayfutyarova, Maximilian Scheurer, Henry F. Schurkus, James E. T. Smith, Chong Sun, Shi-Ning Sun, Shiv Upadhyay, Lucas K. Wagner, Xiao Wang, Alec White, James Daniel Whitfield, Mark J. Williamson, Sebastian Wouters, Jun Yang, Jason M. Yu, Tianyu Zhu, Timothy C. Berkelbach, Sandeep Sharma, Alexander Yu. Sokolov, and Garnet Kin-Lic Chan. Recent developments in the PySCF program package. *The Journal of Chemical Physics*, 153(2):024109, 2020.

Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications, 1996. ISBN 978-0-486-69186-2.

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(1):8815, 2024.

O. Vahtras, J. Almlöf, and M. W. Feyereisen. Integral approximations for LCAO-SCF calculations. *Chemical Physics Letters*, 213(5):514–518, 1993.

J. H. Van Lenthe, R. Zwaans, H. J. J. Van Dam, and M. F. Guest. Starting SCF calculations by superposition of atomic densities. *Journal of Computational Chemistry*, 27(8):926–932, 2006.

Florian Weigend and Reinhart Ahlrichs. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Physical Chemistry Chemical Physics*, 7(18):3297–3305, 2005.

J. L. Whitten. Coulombic potential energy integrals and approximations. *The Journal of Chemical Physics*, 58(10):4496–4501, 1973.

Fiona C. Y. Yu, Christopher Seidl, Elise Palethorpe, and Giuseppe M. J. Barca. Acceleration of Self-Consistent Field Calculations Using Basis Set Projection and Many-Body Expansion as Initial Guess Methods. *Journal of Chemical Theory and Computation*, 21(3):1230–1248, 2025a.

Haiyang Yu, Zhao Xu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Efficient and Equivariant Graph Networks for Predicting Quantum Hamiltonian, 2023.

Haiyang Yu, Meng Liu, Youzhi Luo, Alex Strasser, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. QH9: A Quantum Hamiltonian Prediction Benchmark for QM9 Molecules, 2024.

Haiyang Yu, Yuchao Lin, Xuan Zhang, Xiaofeng Qian, and Shuiwang Ji. Efficient Prediction of SO(3)-Equivariant Hamiltonian Matrices via SO(2) Local Frames, 2025b.

He Zhang, Chang Liu, Zun Wang, Xinran Wei, Siyuan Liu, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Self-Consistency Training for Density-Functional-Theory Hamiltonian Prediction, 2024.

Xing Zhang and Garnet Kin-Lic Chan. Differentiable quantum chemistry with PySCF for molecules and materials at the mean-field level and beyond. *The Journal of Chemical Physics*, 157(20): 204801, 2022.

Zhengji Zhao, Brian Austin, Stefan Maintz, and Martijn Marsman. VASP Performance on HPE Cray EX Based on NVIDIA A100 GPUs and AMD Milan CPUs. Technical report, Cray User Group, 2023.

Contents

A The Self-Consistent Field Cycle **16**

A.1 Direct Inversion of the Iterative Subspace (DIIS) 16

A.2 Energy Gradient on the Density Matrix Manifold 17

B Surrogate Metrics for Initial Guess Quality **17**

C Coefficient-Based Models **19**

D Evaluation Details **20**

D.1 Data Split 20

D.2 Compute Resources 20

D.3 Wall-time measurement procedures 21

E Traditional Baselines **21**

F Related Work and Additional Reference Models **23**

G Convergence Thresholds **25**

H Hyperparameters **26**

I Further Ablations **27**

J Generalization to Transition States **28**

K Robustness of SAIL **29**

A The Self-Consistent Field Cycle

For accessibility, this section is limited to the spin-restricted Kohn-Sham (RKS) framework. In practice, the electron density ρ is expanded in a finite basis set $\{\chi_\mu\}_{\mu=1}^{B}$ of atom-centered functions via a coefficient matrix $\mathbf{C} \in \mathbb{R}^{B\times B}$,

$$\rho(\mathbf{r}) = 2\sum_{\mu\nu}^{B}\sum_{i=1}^{N_e/2} C_{\mu i}\, C_{\nu i}\, \chi_\mu(\mathbf{r})\, \chi_\nu(\mathbf{r}) = \sum_{\mu\nu}^{B} P_{\mu\nu}\, \chi_\mu(\mathbf{r})\, \chi_\nu(\mathbf{r})\,, \tag{8}$$

where $P_{\mu\nu} = 2\sum_{i=1}^{N_{\text{occ}}} C_{\mu i}\, C_{\nu i}$ is the density matrix and the sum runs over the $N_{\text{occ}} = N_e/2$ lowest-energy *occupied* orbitals, with the remaining $N_{\text{virt}} = B - N_{\text{occ}}$ columns of $\mathbf{C}$ termed *virtual* orbitals. In its basis-set discretization, the ground-state energy becomes a *function* of the density matrix

$$E(\mathbf{P}) = \sum_{\mu\nu} H_{\mu\nu}^{(\text{core})} P_{\mu\nu} + \tfrac{1}{2}\sum_{\mu\nu} J_{\mu\nu}(\mathbf{P}) P_{\mu\nu} + E_{\text{xc}}(\mathbf{P})\,, \tag{9}$$

where $\mathbf{H}^{\text{core}}$ contains kinetic and nuclear attraction terms, $\mathbf{J}$ is the Coulomb matrix, and E_{xc} is the exchange-correlation energy. The Fock matrix $\mathbf{F} \in \mathbb{R}^{B\times B}$ is defined as the derivative of the energy w.r.t the density matrix

$$\mathbf{F}(\mathbf{P}) = \frac{\partial E}{\partial \mathbf{P}} = \mathbf{H}^{(\text{core})} + \mathbf{J}(\mathbf{P}) + \mathbf{V}^{\text{xc}}(\mathbf{P})\,. \tag{10}$$

At a minimum of $E(\mathbf{P})$, the coefficients $\mathbf{C}$ satisfy the Roothaan–Hall equations,

$$\big(\mathbf{F}(\mathbf{P})\,\mathbf{C}\big)_{\mu i} = \varepsilon_i\ \big(\mathbf{S}\,\mathbf{C}\big)_{\mu i}\ , \tag{11}$$

a generalized eigenvalue problem, where ε_j are the so-called orbital energies [Roothaan, 1951]. While the overlap matrix $S_{\mu\nu} = \int \chi_\mu(\mathbf{r})\, \chi_\nu(\mathbf{r})\, d\mathbf{r}$ is constant with respect to $\mathbf{C}$, the Fock matrix depends on $\mathbf{C}$ through $\mathbf{P}$, making Eq. (11) nonlinear. The self-consistent field ansatz [Lehtola et al., 2020] linearizes Eq. (11) via the iterative solution approach

$$\begin{aligned} \mathbf{P}^{(t)} &= \mathbf{P}\big(\mathbf{C}^{(t)}\big) \\ \mathbf{F}^{(t)} &= \mathbf{F}\big(\mathbf{P}^{(t)}\big) \\ \mathbf{C}^{(t+1)} &= \text{gEVP}\big(\mathbf{F}^{(t)}, \mathbf{S}\big)\,, \end{aligned} \tag{12}$$

where $\text{gEVP}(\cdot,\cdot)$ returns the eigenvectors of the generalized eigenvalue problem and t is the iteration index (Fig. 2).

A.1 Direct Inversion of the Iterative Subspace (DIIS)

The self-consistent field (SCF) method uses an iterative ansatz to solve Equation (11). The so-called Direct Inversion of the Iterative Subspace (DIIS) formalism accelerates convergence by constructing an optimal linear combination of previous Fock matrices that minimizes a residual error [Pulay, 1980, 1982]. At SCF iteration t, we compute the residual $\mathbf{R}^{(t)} \in \mathbb{R}^{B\times B}$ (commutator error):

$$\mathbf{R}^{(t)} = \mathbf{F}^{(t)}\mathbf{P}^{(t)}\mathbf{S} - \mathbf{S}\mathbf{P}^{(t)}\mathbf{F}^{(t)} \tag{13}$$

where $\mathbf{F}$ is the Fock matrix, $\mathbf{P}$ is the density matrix, and $\mathbf{S}$ is the basis overlap matrix. In practice, one often transforms the residual into an orthonormal basis via $\tilde{\mathbf{R}}^{(t)} = \mathbf{X}^\top \mathbf{R}^{(t)} \mathbf{X}$ with $\mathbf{X} = \mathbf{S}^{-1/2}$, which improves numerical stability. Using the DIIS method, we compute

$$\mathbf{F}_{\text{DIIS}}^{(t)} = \sum_{i=1}^{t} a^{(i)} \mathbf{F}^{(i)}, \tag{14}$$

to construct the next Fock matrix for which the gEVP is solved. The coefficients $\mathbf{a}^{(t)}$ minimize the residual norm subject to the constraint $\sum_{i=1}^{t} a^{(i)} = 1$:

$$\min_{\mathbf{a}\in\mathbb{R}^t} \sum_{jk}^{t}\sum_{\mu\nu}^{B} a^{(j)} a^{(k)}\, R_{\mu\nu}^{(j)}\, R_{\mu\nu}^{(k)}\,. \tag{15}$$

Using Lagrange multipliers, one can derive the Pulay equations [Pulay, 1980]

$$\begin{pmatrix} B_{11} & \cdots & B_{1t} & -1 \\ \vdots & \ddots & \vdots & \vdots \\ B_{t1} & \cdots & B_{tt} & -1 \\ -1 & \cdots & -1 & 0 \end{pmatrix} \begin{pmatrix} a^{(1)} \\ \vdots \\ a^{(t)} \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} \tag{16}$$

where $B_{jk} := \sum_{\mu\nu} R^{(j)}_{\mu\nu} R^{(k)}_{\mu\nu}$.

A.2 Energy Gradient on the Density Matrix Manifold

The SCF fixed-point iteration (Eq. 17) is not the only approach to finding the ground-state density matrix. An alternative is *direct minimization*: treating $E(\mathbf{P})$ as an objective function and optimizing it with gradient-based methods on the constraint set of valid density matrices [Lehtola et al., 2020]. This perspective motivates the loss function used in SAIL.

The set of valid density matrices of rank $N_{\text{occ}} = N_e/2$ forms a smooth manifold (the Grassmannian). Its tangent space at any point is spanned by rotations that mix the N_{occ} occupied with the $N_{\text{virt}} = B - N_{\text{occ}}$ virtual columns of the coefficient matrix $\mathbf{C}^{(t)} \in \mathbb{R}^{B\times B}$. Rotations within the occupied or virtual subspaces leave the density matrix unchanged and are therefore not degrees of freedom of the optimization. The energy gradient with respect to these occupied–virtual rotations is the $N_{\text{occ}} \times N_{\text{virt}}$ matrix

$$G^{(t)}_{ia} = \left(\mathbf{C}^{\text{occ},(t)\top} \mathbf{F}^{(t)} \mathbf{C}^{\text{virt},(t)} \right)_{ia}, \tag{17}$$

where $\mathbf{C}^{\text{occ},(t)} \in \mathbb{R}^{B\times N_{\text{occ}}}$ and $\mathbf{C}^{\text{virt},(t)} \in \mathbb{R}^{B\times N_{\text{virt}}}$ are the occupied and virtual blocks of $\mathbf{C}^{(t)}$, with i indexing occupied and a virtual orbitals. At a converged SCF solution the Fock matrix is diagonal in its own eigenbasis (Eq. (11)), so the off-diagonal block $\mathbf{C}^{\text{occ},\top}\mathbf{F}\,\mathbf{C}^{\text{virt}}$ vanishes — the standard first-order optimality condition $\mathbf{G} = \mathbf{0}$.

This gives a natural convergence measure: any iterate with large $\|\mathbf{G}^{(t)}\|$ is far from a stationary point, regardless of how close $\mathbf{P}^{(t)}$ is to $\mathbf{P}^*$ in Frobenius norm. The per-cycle loss used in SAIL (Eq. (5)) is the root-mean-square of this gradient,

$$\mathcal{L}^{(t)}_{\nabla} = \sqrt{\frac{1}{N_{\text{occ}}\,N_{\text{virt}}} \sum_{i=1}^{N_{\text{occ}}} \sum_{a=1}^{N_{\text{virt}}} \left| G^{(t)}_{ia} \right|^2}. \tag{18}$$

Unlike the DIIS commutator residual $\mathbf{R}^{(t)} = \mathbf{F}^{(t)}\mathbf{P}^{(t)}\mathbf{S} - \mathbf{S}\mathbf{P}^{(t)}\mathbf{F}^{(t)}$ (Appendix A.1), which lives in the full $B \times B$ AO space and mixes all orbital pairs, $\mathbf{G}^{(t)}$ isolates the $N_{\text{occ}} \times N_{\text{virt}}$ degrees of freedom that actually affect the energy. The RMS normalization makes the loss comparable across molecules of different size.

B Surrogate Metrics for Initial Guess Quality

Throughout this section, $\hat{\mathbf{X}}$ denotes a predicted (ML) quantity and $\mathbf{X}^*$ its converged SCF reference. ML models for SCF acceleration are typically trained and evaluated using elementwise error norms on the predicted tensors $\hat{\mathbf{P}}$ or $\hat{\mathbf{F}}$. These Frobenius-based targets are convenient to compute but do not directly characterize what makes an initial guess yield fast and stable SCF convergence. Here, we catalog a broader set of surrogate metrics that probe different aspects of initial-guess quality.

Total energy. The deviation in total electronic energy evaluated at the predicted density:

$$\Delta E = |E(\hat{\mathbf{P}}) - E(\mathbf{P}^*)|, \qquad E(\mathbf{P}) = \sum_{\mu,\nu=1}^{B} H^{(\text{core})}_{\mu\nu} P_{\nu\mu} + \tfrac{1}{2} \sum_{\mu,\nu=1}^{B} J_{\mu\nu}(\mathbf{P})\, P_{\nu\mu} + E_{\text{xc}}[\rho(\mathbf{P})], \tag{19}$$

where $H^{(\text{core})}_{\mu\nu}$ is the one-electron (core) Hamiltonian, $J_{\mu\nu}(\mathbf{P}) = \sum_{\lambda,\sigma=1}^{B} (\mu\nu|\lambda\sigma)\, P_{\lambda\sigma}$ is the Coulomb matrix, and E_{xc} is the exchange-correlation functional including optional HF-exchange

contributions[3]. Since $E(\mathbf{P})$ is stationary at $\mathbf{P}^*$, ΔE is second order in $\|\hat{\mathbf{P}} - \mathbf{P}^*\|$ and therefore insensitive to the direction of the initial guess error.

Mean-field energy. The sum of one-electron and classical Coulomb contributions,

$$E_{\mathrm{MF}}(\mathbf{P}) = \sum_{\mu,\nu=1}^{B} H_{\mu\nu}^{(\mathrm{core})}\, P_{\mu\nu} + \tfrac{1}{2} \sum_{\mu,\nu=1}^{B} J_{\mu\nu}(\mathbf{P})\, P_{\mu\nu}, \tag{20}$$

provides an energetically meaningful measure of density quality that is independent of the choice of exchange-correlation functional [Gould, 2023].

Real-space density norms. The L^p distance between densities:

$$\|\Delta\rho\|_p = \left(\int |\hat{\rho}(\mathbf{r}) - \rho^*(\mathbf{r})|^p \, d\mathbf{r} \right)^{1/p}, \qquad p \in \{1, 2\}. \tag{21}$$

The L^1 norm gives the total absolute electron displacement; the L^2 norm, expressed in the AO basis, involves the four-center overlap tensor $(\mu\nu|\lambda\sigma) = \int \chi_\mu(\mathbf{r})\chi_\nu(\mathbf{r})\chi_\lambda(\mathbf{r})\chi_\sigma(\mathbf{r})\, d\mathbf{r}$. Both weight all spatial regions uniformly and do not distinguish OV errors from redundant OO or VV contributions.

Dipole moment. Unlike the L^p norms, which weight all regions of space equally, the electronic dipole moment weights the density error by the position vector $\mathbf{r}$, making it a physically transparent measure of charge displacement:

$$\Delta\mu = \|\boldsymbol{\mu}(\hat{\rho}) - \boldsymbol{\mu}(\rho^*)\|, \qquad \mu_\alpha(\rho) = -\int r_\alpha\, \rho(\mathbf{r})\, d\mathbf{r}, \tag{22}$$

where the nuclear contribution cancels in the difference $\Delta\mu$ and is therefore omitted. The linear weighting by $\mathbf{r}$ projects out only the $\ell = 1$ multipole component of $\Delta\rho$, so $\Delta\mu$ is insensitive to higher-order error structure, but directly governs long-range electrostatic properties.

Orbital projection. Lehtola [2019] propose the projection of the initial guess onto the converged occupied subspace,

$$Q = \sum_{\mu,\nu,\lambda,\kappa=1}^{B} \hat{P}_{\mu\nu}\, S_{\nu\lambda}\, P^*_{\lambda\kappa}\, S_{\kappa\mu}, \tag{23}$$

as a continuous metric that separates initial guess quality from the dynamics of the SCF algorithm.

DIIS residual. The commutator that measures departure from self-consistency:

$$r_{\mathrm{DIIS}} = \|\mathbf{R}\|_F, \qquad R_{\mu\nu} = \sum_{\lambda,\sigma=1}^{B} \Big(F_{\mu\lambda}(\hat{\mathbf{P}})\, \hat{P}_{\lambda\sigma}\, S_{\sigma\nu} - S_{\mu\lambda}\, \hat{P}_{\lambda\sigma}\, F_{\sigma\nu}(\hat{\mathbf{P}}) \Big). \tag{24}$$

In an orthonormal basis this reduces to $\|[\mathbf{F}(\hat{\mathbf{P}}), \hat{\mathbf{P}}]\|_F$, whose OV block is the orbital gradient $g_{ia} = 2F_{ia}$. This isolates the OV sector but treats all OV pairs uniformly regardless of their gap.

Orbital rotation gradient. The squared norm of the energy gradient with respect to occupied-virtual rotations:

$$\|G\|^2 = \sum_{ia} \Big(\sum_{\mu,\nu=1}^{B} C_{\mu i}^{\mathrm{occ}}\, F_{\mu\nu}(\hat{\mathbf{P}})\, C_{\nu a}^{\mathrm{virt}} \Big)^2, \tag{25}$$

where $\mathbf{C}^{\mathrm{occ}}$ and $\mathbf{C}^{\mathrm{virt}}$ are the occupied and virtual orbital coefficient matrices obtained from $\hat{\mathbf{P}}$. Like r_{DIIS}, $\|G\|$ isolates the OV sector but weights all OV pairs uniformly regardless of their gap.

[3] Hence, E_{xc} is formally a *functional* of ρ and, if the functional is a hybrid, also a *function* of $\mathbf{P}$.

Frobenius norms. Elementwise matrix distances:

$$\|\Delta\mathbf{F}\|_F = \Big(\sum_{\mu,\nu=1}^{B} |\hat{F}_{\mu\nu} - F^*_{\mu\nu}|^2\Big)^{1/2}, \qquad \|\Delta\mathbf{P}\|_F = \Big(\sum_{\mu,\nu=1}^{B} |\hat{P}_{\mu\nu} - P^*_{\mu\nu}|^2\Big)^{1/2}. \tag{26}$$

The Frobenius norm treats OO, OV, and VV blocks identically. For $\mathbf{F}$, core-core diagonal elements can dominate over the small but physically critical OV elements that enter the energy gradient G_{ia}. For $\mathbf{P}$, the set of valid density matrices forms a Grassmannian of dimension $N_{\mathrm{occ}} \times N_{\mathrm{virt}}$, far lower than the B^2 degrees of freedom of the full basis representation. The Frobenius norm penalizes deviations in all B^2 components equally, including OO and VV blocks that correspond to redundant rotations affecting neither observables nor convergence.

C Coefficient-Based Models

The auxiliary-coefficient ansatz predicts a linear expansion of the electron density, $\hat{\rho}(\mathbf{r}) = \hat{c}_P\, \chi_P^{(\mathrm{aux})}(\mathbf{r})$, rather than an AO density matrix [Song and Feng, 2024, Liu et al., 2025]. The Kohn-Sham Fock matrix decomposes as

$$\mathbf{F} \;=\; \mathbf{H}^{(\mathrm{core})} \;+\; \mathbf{J}[\rho] \;+\; \mathbf{V}_{\mathrm{xc}}^{(\mathrm{local/grid})}[\rho, \nabla\rho, \tau] \;+\; \alpha\, \mathbf{K}[\mathbf{P}], \tag{27}$$

where $\alpha \in [0, 1]$ is the exact-exchange fraction, $\mathbf{V}_{\mathrm{xc}}^{(\mathrm{local/grid})}$ collects the semi-local XC contribution, and $\mathbf{K}$ is the Hartree-Fock exchange matrix. $\mathbf{H}^{(\mathrm{core})}$ is density-independent, and $\mathbf{J}$ is evaluated from $\hat{\rho}$ through the standard auxiliary-basis density-fitting route [Vahtras et al., 1993]. The remaining two terms require inputs that $\hat{\rho}$ alone does not determine, namely the kinetic-energy density τ for meta-GGA functionals and the density matrix $\mathbf{P}$ for any functional with $\alpha > 0$. Liu et al. [2025] address each with a substitution derivable from $\hat{\rho}$.

Meta-GGA functionals. The kinetic-energy density

$$\tau(\mathbf{r}) \;=\; \tfrac{1}{2} \sum_i \nabla\psi_i(\mathbf{r}) \cdot \nabla\psi_i(\mathbf{r}) \tag{28}$$

is a sum over occupied molecular orbitals $\{\psi_i\}$ and is not recoverable from ρ. Liu et al. [2025] substitute the von Weizsäcker kinetic-energy density

$$\tau_{\mathrm{vW}}(\mathbf{r}) \;=\; \frac{\nabla\rho(\mathbf{r}) \cdot \nabla\rho(\mathbf{r})}{8\,\rho(\mathbf{r})}, \tag{29}$$

which depends only on ρ and $\nabla\rho$ and is therefore directly evaluable from $\hat{\mathbf{c}}$ [Perdew and Constantin, 2007]. The τ-dependent contribution to the meta-GGA XC matrix is then evaluated on a real-space quadrature grid $\{\mathbf{r}_g\}$ with weights $\{\omega_g\}$ as

$$\big[\mathbf{V}_{\mathrm{xc}}^{(\tau)}\big]_{\mu\nu} = \frac{1}{2} \sum_g \omega_g\, v_\tau(\mathbf{r}_g)\, \nabla\chi_\mu(\mathbf{r}_g) \cdot \nabla\chi_\nu(\mathbf{r}_g), \qquad v_\tau = \left.\frac{\partial[\rho\, \varepsilon_{\mathrm{xc}}(\rho, \nabla\rho, \tau)]}{\partial\tau}\right|_{\rho=\hat{\rho},\, \tau=\tau_{\mathrm{vW}}}. \tag{30}$$

τ_{vW} is exact for single-orbital systems and introduces a systematic approximation error for general molecules [Perdew and Constantin, 2007]. The predicted $\hat{\rho}$ therefore parametrizes an approximate meta-GGA Fock build rather than the exact one.

Hybrid and range-separated functionals. The exchange matrix $\mathbf{K}[\mathbf{P}]$ is a function of the density matrix rather than of ρ. Many density matrices correspond to the same ρ, so reconstructing $\mathbf{P}$ from $\hat{\rho}$ is underdetermined. Liu et al. [2025] replace the learned density matrix with the superposition-of-atomic-densities (SAD) matrix,

$$\mathbf{P}_{\mathrm{SAD}} = \mathrm{blockdiag}(\mathbf{P}_{A_1}, \ldots, \mathbf{P}_{A_M}), \tag{31}$$

where $\mathbf{P}_A$ is the atomic SCF density matrix for atom A, and use $\mathbf{K}[\mathbf{P}_{\mathrm{SAD}}]$ in Eq. (27). $\mathbf{P}_{\mathrm{SAD}}$ coincides with the MINAO initial guess of most quantum-chemistry implementations [Sun et al., 2020]. The $\alpha\,\mathbf{K}$ contribution is therefore independent of the learned prediction and coincides with the exchange contribution from a MINAO/SAD initial density, while the learned coefficients still determine the Coulomb and semi-local XC terms.

ERIC considerations for coefficient models. ERIC charges $\hat{\mathbf{c}} \to \mathbf{F}^{(-1)}$ as one full Fock build, matching the convention for matrix-based Δ-learning. This is always an upper bound: $\mathcal{M}_{\mathbf{c}\to\mathbf{F}}$ skips the forward auxiliary fit $\mathbf{P} \to \mathbf{c}$ that a normal $\mathbf{J}$ build requires, and substitutes τ_{vW} for the MO-dependent τ in meta-GGAs at lower grid cost, so $\mathcal{M}_{\mathbf{c}\to\mathbf{F}}$ is strictly cheaper than $\mathcal{M}_{\mathbf{P}\to\mathbf{F}}$ in any reasonable implementation. The size of the gap depends on which terms dominate per-iteration cost. For semi-local functionals (PBE, SCAN), the iteration is dominated by $\mathbf{J}[\mathbf{P}]$ at $\mathcal{O}(B^4)$ without density fitting and falls to $\mathcal{O}(B^2 B_{\mathrm{aux}})$ with J-fitting, comparable to the grid-based $\mathbf{V}_{\mathrm{xc}}$ contribution; the $+1$ charge therefore overcounts substantially without DF and only modestly with DF. For hybrids, $\mathbf{J}$ and $\mathbf{K}$ each scale as $\mathcal{O}(B^4)$ without DF and contribute comparable wall-time, so the asymptotic initial-guess overhead reduces to roughly half a Fock build because it saves the $\mathbf{J}$ build of a standard iteration. With JK-fitting, $\mathbf{K}$ retains the larger prefactor due to its N_{occ}-scaling and dominates the iteration cost; the coefficient overhead approaches (but stays below) a full Fock build. Across all six functional/DF combinations, the ERIC values we report for coefficient-based models are therefore conservative upper bounds, with the slack largest for semi-local functionals without DF and the smallest for hybrids with JK-fitting.

MINAO overhead. A separate wall-time consideration, independent of ERIC, is the cost of the MINAO (SAD) build itself. Matrix-based Δ-learning requires it at every level of theory to construct the residual base $\mathbf{P}_{\mathrm{base}}$ for density-matrix targets, or $\mathbf{F}_{\mathrm{base}} = \mathcal{M}_{\mathbf{P}\to\mathbf{F}}(\mathbf{P}_{\mathrm{MINAO}})$ for Hamiltonian targets. The coefficient model only references a density matrix through $\mathbf{K}[\mathbf{P}_{\mathrm{SAD}}]$ (Eq. (27)), which drops out at $\alpha = 0$. At the GGA and meta-GGA level the coefficient pipeline therefore avoids the MINAO build entirely. At the hybrid level both pipelines pay it: Δ-learning as the residual base, the coefficient model as $\mathbf{P}_{\mathrm{SAD}}$ for the exchange substitution.

D Evaluation Details

The reference computations and final speedup evaluations are performed using PySCF [Sun et al., 2020] and GPU4PySCF [Li et al., 2025a] with grid level 1. We use the def2-SVP basis set [Weigend and Ahlrichs, 2005] and do exact calculations. The RIC and ERIC are measured relative to PySCF's default (MINAO), using its default convergence threshold of 10^{-9} Ha [Sun et al., 2020] and maximum number of cycles $T = 50$.

D.1 Data Split

We follow Yu et al. [2024] and split QM9 by molecular size, assigning molecules with at most 20 atoms to training, 21–22 to validation, and 23 or more to testing. To evaluate out-of-distribution generalization beyond QM9, we additionally use QM40 and QMugs as far-out-of-distribution test sets. For QM40 we randomly sample up to 100 molecules per heavy-atom count, whereas for QMugs we sample 10 molecules at every tenth heavy-atom count from 40 to 90, drawing at most one conformer per molecule since QMugs provides three conformers each. For both QM40 and QMugs we restrict the sampling to molecules composed only of the elements present in QM9 (H, C, N, O, F), isolating size extrapolation from element extrapolation.

D.2 Compute Resources

All experiments were conducted on a local compute cluster. Pretraining and fine-tuning were performed on single-GPU jobs using NVIDIA H200 GPUs (144 GB GPU memory). Pretraining runs additionally used 16 CPU cores and approximately 100 GB of host RAM, while SAIL fine-tuning used approximately 300 GB of host RAM. Evaluation runs were conducted separately on a single NVIDIA A100 GPU with 40 GB of memory.

A single pretraining run required roughly 2.5–4 days of wall-clock time on one H200 GPU. In our setup, pretraining used less GPU memory than the default JAX preallocation, which is 75% of total GPU memory. In practice, pretraining could also be carried out on A100-class GPUs without difficulty. A single SAIL fine-tuning run required roughly 1.5–3 days of wall-clock time on one H200 GPU and used approximately 87% of the available H200 memory. Fine-tuning therefore required larger-memory GPUs than pretraining. Under the truncation ablations described in the paper, the fine-tuning memory requirement can be reduced.

To reproduce the full set of reported results, we estimate a total budget of approximately $16 \times 3.5 + 22 \times 2.5 = 111$ H200 GPU-days for pretraining and fine-tuning runs, plus approximately 10 A100 GPU-days for evaluation. The full research project required more compute than the totals above, since it also included failed runs, debugging, preliminary experiments, early-stage ablations, and hyperparameter exploration.

D.3 Wall-time measurement procedures

All wall-time measurements were performed on a single NVIDIA A100 GPU. We separate the neural initialization from the SCF loop into two stages rather than running them concurrently, because JAX and GPU4PySCF do not share a single GPU cleanly. JAX assumes it owns the device, tends to pre-allocate aggressively, and does not always release memory back to the driver, which interferes with GPU4PySCF's memory management. Running the two stages back-to-back on the same GPU, gives a clean measurement of each.

In the first stage, the ML model produces an initial density matrix for each molecule. The reported initialization time covers the model forward pass and the construction of the initial density matrix from the model output. For the coefficient-based Fock build, we use the optimized implementation of Liu et al. [2025]. In the second stage, this density matrix is passed to GPU4PySCF for the SCF calculation using the same basis set, exchange-correlation functional, grid level, and density-fitting settings as the baseline. The accelerated wall-time is the sum of the two stages. The baseline is GPU4PySCF initialized with the standard MINAO guess.

We exclude two categories of cost from the ML-initial guess acquisition, both of which a production implementation would avoid. First, we exclude the construction of integral tensors that GPU4PySCF recomputes in the subsequent SCF stage, such as the overlap matrix. A merged implementation would compute these once and share them across both stages. Second, we exclude JAX tracing and compilation, which ahead-of-time compilation removes entirely.

E Traditional Baselines

A perfect prediction of the target ground state would reach RIC$= 0$, so there is no upper bound on what a ground-state-based initial guess can achieve. What we can measure instead is how far a realistic but imperfect ground-state-like guess takes us. A natural way to construct such a guess is to run the target SCF starting from a converged ground-state density of a cheaper functional, where the imperfection comes from the difference between the two theories rather than from an ML approximation. This gives a reference point for the acceleration achievable from ground-state-like guesses that fall short of the true target by a known amount. We run all valid chains across Jacob's

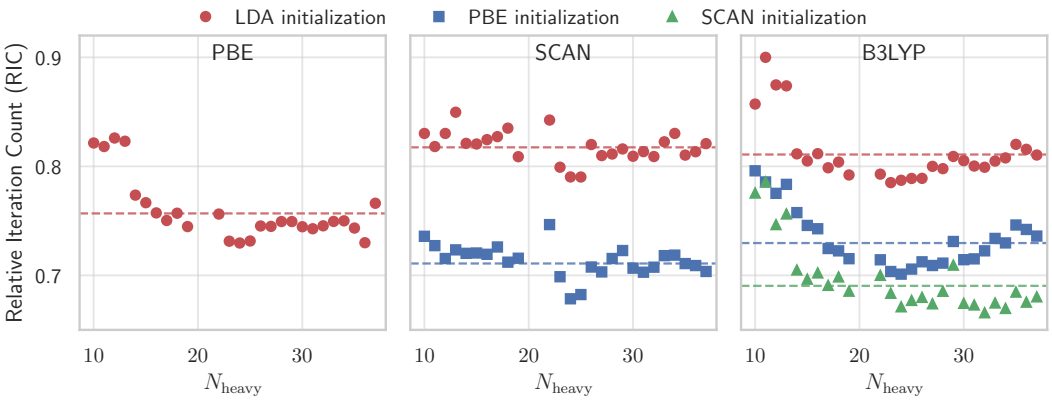


Figure 7: RIC when initializing KS-DFT from converged orbitals of a cheaper functional. Each panel is a different target functional, with initialization sources colored. Dashed lines show mean RIC. SCAN $\rightarrow$ B3LYP reaches the lowest RIC at $\sim$70%.

ladder (LDA → PBE, LDA/PBE → SCAN, and LDA/PBE/SCAN → B3LYP) and report the resulting RIC in Figure 7.

Initializing from a converged cheaper functional reduces iteration counts by 20–30% relative to MINAO, with higher-rung sources yielding progressively lower RIC. The reductions are stable across N_{heavy}. SCAN → B3LYP reaches the lowest RIC at $\sim$70%.

Wall-time reductions do not follow. The pre-run is not free. Even SCAN, despite its formally lower scaling than B3LYP, costs roughly ten Fock builds to converge, which must be offset by iteration savings in the subsequent hybrid SCF. An ML initial guess, by contrast, costs a single forward pass. The RIC reductions reported here are therefore not wall-time competitive with ML initialization, and serve only as a reference point for the iteration savings attainable when a ground-state-like guess approximates the target through a cheaper theory.

Table 6: Relative iteration reduction ($1 - \text{RIC}$) with respect to MINAO as reported in prior work. Values with a star use Δ-Learning for Fock prediction or a coefficient Fock build with similar cost, such that RIC $\neq$ ERIC. OOD values in brackets are limited extrapolation.

Model	Theory	ID	OOD-S	OOD-L	Notes
QHNet					
Yu et al. [2024]	B3LYP/SVP	29%	28%	-	QH9 dataset. For OOD: train and validation on < 22, test on up to 29 atoms
Zhang et al. [2024]	"	-	36%	-	QH9 dataset, with subsampled test set. With additional self-consistency tuning
Zhang et al. [2024]	"	34%*	-	-	Trained on MD22 and QH9 datasets, and evaluated on MD22. With additional self-consistency tuning
Kim et al. [2026]	"	33%	-	-	QH9 dataset. Convergence tol. 10^{-13} Ha
Li et al. [2025b]	"/TZVP	10%	-	-	PubChemQH (40 - 100 Atoms). WALoss
Liu et al. [2025]	PBE/SVP	37%	-	−80%	On SCF-bench (QM9-like)
WANet					
Li et al. [2025b]	B3LYP/TZVP	18%	-	-	PubChemQH (40 - 100 Atoms). WALoss
QHFlow					
Kim et al. [2026]	B3LYP/SVP	40%*	-	-	QH9 dataset. Convergence tol. 10^{-13} Ha
Liu et al. [2025]	PBE/SVP	43%*	-	−47%*	On SCF-bench (QM9-like)
Graph2Mat					
Febrer et al. [2025]	PBE/DZP	40%	-	-	On QM9 molecules with SIESTA defaults [García et al., 2020]. PBE functional confirmed by correspondence with the authors.
aux-coefficients					
Liu et al. [2025]	PBE/SVP	36%*	-	33%*	On SCF-bench (QM9-like)
"	SCAN/ "	12%*	-	14%*	" trained on PBE
"	B3LYP/ "	15%*	-	16%*	" trained on PBE

Table 7: Relative iteration reduction ($1 - \text{RIC}$) with respect to MINAO, measured in this work. Higher is better; negative values indicate deceleration. We use the def2-SVP basis set and PySCF's defaults [Sun et al., 2020]. The Coefficients, Density Matrix, and Fock rows are trained on QM9 with OOD-L evaluated on QM40. Values with a star use Δ-Learning for Fock prediction or a coefficient Fock build with similar cost, such that RIC $\neq$ ERIC.

Model	**Functional**	**OOD-S**		**OOD-L**	
		base	SAIL	base	SAIL
Δ-QHNet					
This work	B3LYP	30%*	-	−94%*	-
QHFlow					
This work	B3LYP	32%*	-	−88%*	-
QDensFlow					
This work	B3LYP	15%	-	−22%	-
Coefficients	PBE	44%*	41%*	42%*	41%*
	SCAN	18%*	31%*	23%*	33%*
	B3LYP	13%*	33%*	18%*	32%*
Density Matrix	PBE	29%	29%	−5%	40%
	SCAN	29%	36%	1%	35%
	B3LYP	25%	35%	−2%	32%
Fock	PBE	32%*	28%*	−11%*	34%*
	SCAN	34%*	33%*	−16%*	31%*
	B3LYP	32%*	29%*	−47%*	30%*

F Related Work and Additional Reference Models

OOD-S vs OOD-L. Here we introduce two out-of-distribution regimes to reconcile our results with prior work. OOD-S follows the standard QH9-stable-ood split of Yu et al. [2024], in which training molecules have up to 20 atoms and test molecules contain up to 29 atoms, corresponding to a mild size extrapolation of less than $1.5\times$ the training distribution. OOD-L corresponds to a more demanding protocol in which a substantial fraction of the test set consists of molecules more than twice the size of the largest training molecule. Since the OOD split of QH9 alone has been reported to be insufficient to expose size-transferability failures of learned Hamiltonian initializations [Liu et al., 2025], we adopt OOD-L as our primary out-of-distribution benchmark and retain OOD-S only for comparability with prior work. We match the metric used in prior work (RIC; Table 6) in our own measurements (Table 7), expressing both as the relative iteration reduction $1 - \text{RIC}$ with respect to MINAO so that readers can compare across the two tables directly.

Reproduction of QHFlow. Using the official QHFlow reference implementation [Kim et al., 2025], we are unable to reproduce the 69% ID SCF iteration reduction originally reported, and our numbers instead align with the independent evaluation of Liu et al. [2025] and with the authors' own re-evaluation in Kim et al. [2026], both of which report substantially lower acceleration. Inspection of the reference implementation reveals two accounting choices that inflate the reported acceleration relative to a like-for-like comparison against the MINAO baseline used elsewhere in the literature [Sun et al., 2020]. First, although the paper describes the 100% reference as conventional DFT initialized with MINAO, the inference pipeline computes the baseline SCF using the `1e` (core-Hamiltonian) guess via `init_guess_by_1e`,[4] which is known to be a strictly worse starting point than MINAO [Lehtola, 2019] and therefore inflates the denominator of every iteration and wall-time ratio. Second, QHFlow is a residual-learning model and requires an initial Fock matrix F_{init} as input; this F_{init} is produced

[4] `https://github.com/seongsukim-ml/QHFlow/blob/b2e8662ee8e6549e2a00b66bb41f0709bf24a4b3/src/pl_module/base_module.py#L356`, commit `b2e8662`, dated 2025-10-25.

by a full Fock build on top of a MINAO density[5] and cached in the preprocessed dataset, so that the per-molecule cost of constructing the model's own input is amortized into offline dataset preparation and is absent from both the inference timer and the SCF timer. A similar gap applies to the generalized eigendecomposition that converts the predicted Hamiltonian into an initial density matrix, which is executed after the inference timer has stopped and before the SCF timer has started. Under an evaluation protocol that charges these costs to the per-molecule budget and uses MINAO as the 100% reference, we obtain the numbers reported in Table 7. We can, however, reproduce the size extrapolation failure mode reported by Liu et al. [2025] for the original QHFlow.

QDensFlow. To disentangle the cost of the extra Fock build required by Hamiltonian-target learning from the acceleration attributable to the learned initialization itself, we introduce QDensFlow, an adapted variant of QHFlow in which the supervision target is swapped from the Fock matrix to the density matrix while keeping the architecture, prior distributions, flow-matching objective, and residual-learning setup otherwise unchanged. The predicted density matrix is passed directly to SCF as `dm0`, and the MINAO density matrix used as the residual reference P_{init} is available at negligible cost, removing the per-molecule Fock build that Hamiltonian-target residual learning requires. We do not expect this swap to resolve the size extrapolation failure and confirm this empirically in Table 7, but it allows a fair accounting of the per-molecule cost of learned initialization within the QHFlow framework.

Consistency with independent evaluations. Our reproduction is consistent with the independent evaluation of Liu et al. [2025], who report a QHFlow ID acceleration of 43% and an OOD-L deceleration of -47% on SCF-bench, well below the 69%/68% originally claimed. The authors of QHFlow themselves appear to have revised these numbers downward in their follow-up work [Kim et al., 2026], where QHFlow is re-evaluated under a shared SCF protocol and reaches only 40% ID acceleration on QH9 (B3LYP/SVP), in line with our measurements and with those of Liu et al. [2025].

[5] https://github.com/seongsukim-ml/QHFlow/blob/b2e8662ee8e6549e2a00b66bb41f0709bf24a4b3/src/dataset_module/ori_dataset.py#L313-L330, same commit.

G Convergence Thresholds

The SCF convergence threshold ϵ_{tol} governs when the self-consistent field iteration is considered converged, and it has a non-trivial effect on both the quality of the resulting electron density and the computational cost incurred. Figure 8 shows this dependence for B3LYP, SCAN, and PBE on the QM40 benchmark at heavy atom count 36, sweeping ϵ_{tol} from 10^{-5} to 10^{-13}.

Effect on SCF cost. The right column of Figure 8 decomposes the mean number of SCF iterations into the baseline contribution and the number of iterations the SAIL coefficient ansatz requires. Both components grow with tighter thresholds, and the total iteration count roughly doubles when moving from $\epsilon_{\text{tol}} = 10^{-5}$ to 10^{-13} (e.g., from ≈ 9 to ≈ 22 for B3LYP).

Choice of threshold and cross-work comparisons. Throughout the main paper we use $\epsilon_{\text{tol}} = 10^{-9}$, which offers a favorable balance between converged densities and computational cost. We stress that comparisons across different works must account for the convergence threshold used: a method evaluated at $\epsilon_{\text{tol}} = 10^{-6}$ will systematically report lower ERIC values than the same method at 10^{-10}, regardless of the underlying functional quality. When reproducing or comparing to our results, we therefore recommend using the same threshold, or reporting results at multiple thresholds as shown here.

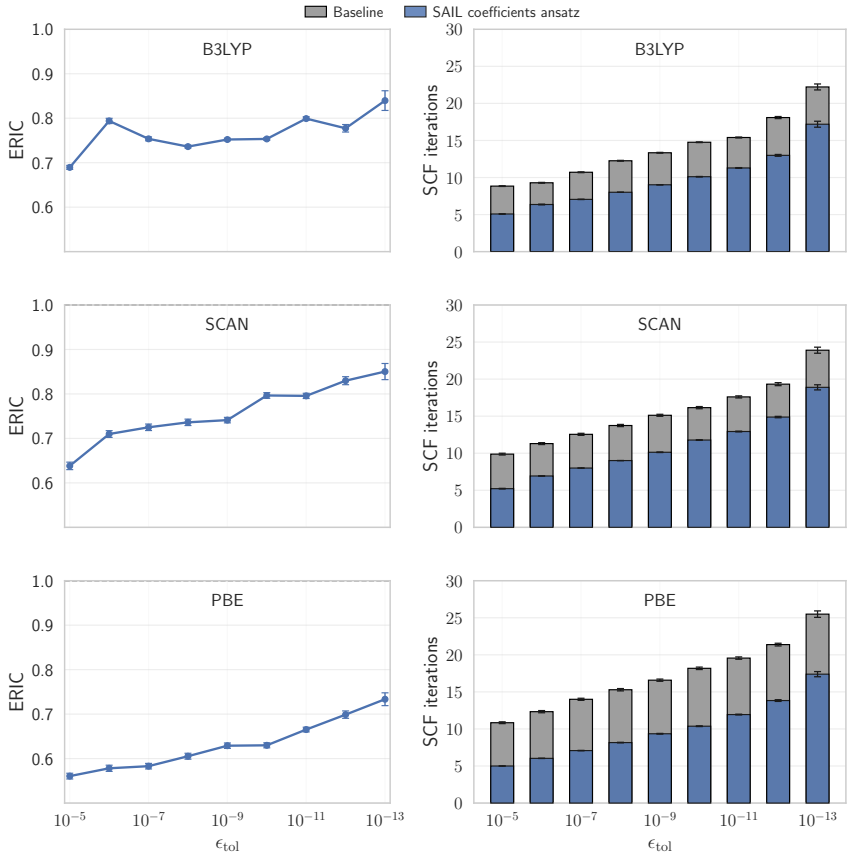


Figure 8: Effect of the SCF convergence threshold ϵ_{tol} on accuracy and computational cost, evaluated on QM40 molecules with heavy atom count 36. **Left column:** ERIC as a function of ϵ_{tol} for B3LYP (top), SCAN (middle), and PBE (bottom). Tighter thresholds yield higher ERIC across all functionals, with the variation between 0.15 and 0.21 ERIC units over the full range. **Right column:** Mean number of SCF iterations, decomposed into the baseline contribution (gray) and the overhead of the SAIL coefficient ansatz (blue). Iteration counts roughly double from $\epsilon_{\text{tol}} = 10^{-5}$ to 10^{-13}. Error bars indicate the standard error over the evaluation set. All main-paper results use $\epsilon_{\text{tol}} = 10^{-9}$ (vertical dashed line in the main text figures).

H Hyperparameters

Table 8: Hyperparameters not listed here follow the defaults of EquiformerV2 (base model), NeuralSCF (embedding and coefficient readout), and QHNet (matrix readout).

Parameter	Value (c/ P/ F)	Notes
Base Model		
Radial cutoff (Å)	5	Typical value for MLIPs, also used by Equiformerv2.
Emb dim encoder	128	Dimensionality of each learned atom-type embedding.
Env hidden encoder	64	Width of the encoder edge MLP.
Gaussian RBF	128	
Resolution of point samples R	9	We use $\beta = 9$, $\alpha = 2\beta + 1$ as opposed to square 18x18. Memory-costly hyperparameter with negligible difference for higher values.
Maximum degree L_{max}	4	Matches highest angular momentum of (auxiliary) basis functions used in our experiments
Baseline (Pretraining)		
Loss	L^2–Overlap/ mixed Frobenius–L^1	We used the proposed loss by Song and Feng [2024] for our coefficient-based models and Yu et al. [2024] for the matrix-based models.
Optimizer	Muon	More stable, less sensitive to LR [Jordan et al., 2024].
Epochs	100	Good balance between performance and training time.
Batch size	32	Tried $\{1, 4, 8, 32, 64\}$.
Warmup schedule	linear	1000 Steps.
Decay	Cosine	
Base LR	5e-3	Tried {2e-2, 1e-2, 5e-3, 2e-3, 1e-3, 5e-4}. LR higher than 5e-3 would sometimes result in divergence.
Min LR	1e-6	
EMA decay	0.995	Exponential moving average of model parameters. Stabilizes training, improves generalization.
Gradient clipping norm	10.0	Reduces impact of outliers.
Weight decay	1e-3	
SAIL (Fine-tuning)		
Loss	Eq. (5)	Commutator Loss can be used as an alternative.
Optimizer	Muon	[Jordan et al., 2024]
Epochs	5	
Batch size	1	Current implementation does not support batching
Warmup schedule	linear	1000 Steps.
Decay	Cosine	
Base LR	5e-4 / 1e-3 / 5e-4	Tried {5e-3, 2e-3, 1e-3, 5e-4, 2e-4, 1e-4, 5e-5, 1e-5}. While lower LR can have better learning curves ID, they end up performing worse OOD. We emphasize the importance of higher LR in combination with EMA and Muon.
Min LR	1e-7	Improvements unnoticeable below this.
EMA decay	0.995	Exponential moving average of model parameters. Stabilizes training, improves generalization.
Gradient clipping norm	1.0	
SCF-cycles T	10	Most molecules converge within 10 cycles. Ablations showed that it can be reduced to $T = 1$ without losing significant performance.
SCF-loss weighting	Uniform	Since early SCF iterations typically have larger errors, this naturally gives them a larger absolute contribution to the total loss. We also tested reweighting schemes that equalize relative contributions across cycles or emphasize later cycles, but both degraded performance.

I Further Ablations

Table 9: **SCF acceleration of baseline and SAIL (fine-tuned) models** across input representations and functionals, evaluated on QM40 (out-of-distribution). Grey rows show ablation variants. **Bold**: best RIC/ERIC per functional across all input representations.

Model		Functional	Ablation	Surrogate Loss (↓)		RIC (↓)		ERIC (↓)	
				base	SAIL	base	SAIL	base	SAIL
Coefficients	**c**	PBE		0.003	0.015	0.58	0.59	0.65	0.66
		"	Embedding	0.007	0.013	0.62	0.60	0.69	0.67
		"	B_{aux}	0.005	0.014	**0.55**	0.61	0.62	0.67
		SCAN		0.004	17.558	0.77	0.67	0.85	0.74
		"	B_{aux}	0.005	21.403	0.76	0.67	0.84	0.74
		B3LYP		0.003	53.815	0.82	**0.68**	0.90	0.76
		"	B_{aux}	0.003	61.490	0.82	**0.68**	0.89	0.76
Density Mat.	**P**	PBE		0.281	1.149	1.05	0.60	1.05	**0.60**
		SCAN		0.250	1.100	0.99	**0.65**	0.99	**0.65**
		"	$\mathcal{L}_{\text{comm}}$	"	1.108	"	0.65	"	0.65
		"	T=4	"	1.101	"	0.65	"	0.65
		"	T=2	"	1.110	"	0.65	"	0.65
		"	T=1	"	1.145	"	0.65	"	0.65
		"	Non-Δ	0.260	0.942	1.09	0.66	1.09	0.66
		"	Single-stage	–	1.270	–	0.66	–	0.66
		B3LYP		0.267	0.976	1.02	**0.68**	1.02	**0.68**
Fock	**F**	PBE		0.005	0.277	1.11	0.66	1.18	0.73
		SCAN		0.017	0.284	1.16	0.69	1.23	0.76
		"	Non-Δ	0.453	3.431	1.99	0.79	1.99	0.79
		B3LYP		0.026	0.382	1.47	0.70	1.55	0.78
		"	Non-Δ	0.185	6.913	2.08	0.74	2.08	0.74

We ablate the density embedding of Song and Feng [2024], which conditions the equivariant node features on the MINAO density. Removing it improves both the baseline and SAIL fine-tuned models. Replacing the default `jfit` with the larger `jkfit` auxiliary basis moves the SAIL-trained ERIC by ≤ 0.02 on every functional while substantially raising the cost of the initial Fock build $\mathcal{M}_{\mathbf{c}\to\mathbf{F}}$, so we keep the cheaper default. As an alternative to the gradient loss $\mathcal{L}_{\nabla}$ (Eq. 5), we have tried a commutator-based loss

$$\mathcal{L}_{\text{comm}}^{(t)} = \frac{1}{B} \left\| \mathbf{F}^{(t)} \mathbf{P}^{(t)} \mathbf{S} - \mathbf{S} \mathbf{P}^{(t)} \mathbf{F}^{(t)} \right\|_F, \tag{32}$$

which corresponds to the root-mean-square of the DIIS error matrix (Appendix A.1), a convergence measure routinely used in quantum chemistry solvers as the extrapolation residual [Pulay, 1980, 1982]. Both losses yield identical ERIC (0.65) on SCAN. Energy-based alternatives, such as the energy change between consecutive iterations or the energy difference to the final iterate, failed to produce any acceleration in our early experiments.

Across functionals, Δ-density matrix prediction emerges as a strong default for SCF acceleration. On PBE, it achieves ERIC 0.60 vs. 0.66 for the coefficient approach while generalizing to all functional classes without modification. For SCAN and B3LYP, the coefficient baseline reaches ERICs of 0.85 and 0.90. SAIL recovers this to 0.74 and 0.76, still behind Δ-density matrix (0.65 and 0.68). Although Δ-Fock prediction achieves a low RIC on B3LYP (0.70), its hidden Fock build yields an ERIC of 0.78, behind the density matrix model at 0.68.

J Generalization to Transition States

To test out-of-distribution generalization to reactive non-equilibrium structures, we evaluate our single-stage ($T = 1$) density-matrix model trained on SPICEv2 at the B3LYP/def2-SVP level of theory on two reaction pathways from the Transition1x dataset [Schreiner et al., 2022]. Along each pathway, $\xi = 0$ denotes the reactant, $\xi = 0.5$ the labeled transition state, and $\xi = 1$ the product, parameterized by the signed RMSD to the transition state and normalized separately on each side.

Table 10 reports results for intramolecular hydrogen transfer in $C_3H_8N_2O$. SAIL accelerates every frame along the reaction coordinate (ERIC 0.64–0.90), with the margin narrowing near the transition point ($\Delta E \approx 48.8\,\mathrm{kcal/mol}$, ERIC 0.82–0.90).

Table 10: **Intramolecular H-transfer** in $C_3H_8N_2O$ from Transition1x [Schreiner et al., 2022] at B3LYP/def2-SVP using the single-stage ($T = 1$) SPICEv2 **P** model.

ξ	**Frame**	ΔE (kcal/mol)	**MINAO** (iter.)	**ERIC** (↓) SAIL
0.00	reactant	0.00	11	0.64
0.33		2.05	11	0.73
...		...	...	...
0.43		6.25	11	0.64
0.45		13.48	12	0.67
0.46		24.30	12	0.67
0.48		48.82	10	0.90
0.49		48.84	11	0.82
0.50	transition state	48.83	11	0.82
0.51		48.83	10	0.90
0.60		37.47	11	0.73
0.64		27.20	10	0.80
...		...	...	...
0.89		20.15	10	0.80
1.00	product	19.78	10	0.80

Table 11 evaluates the C−C bond dissociation of the largest molecule in Transition1x, $C_7H_{16} \rightarrow C_6H_{12} + CH_4$. At the equilibrium bond length (1.53 Å), the structure is in distribution and reaches an ERIC of 0.67. As the C−C bond stretches, ERIC increases to 0.90 by 2.23 Å around the transition state and reaches 1.00 at 2.95–3.13 Å once the bond is broken. Although SPICEv2 contains many multi-fragment systems, it includes no bonds caught mid-dissociation, and at 3.13 Å the fragments remain within the C···C van der Waals contact (≈ 3.4 Å) as a compressed contact pair rather than a separated dimer. Even in this regime, the SAIL initial guess never decelerates the solver relative to MINAO.

Table 11: **Bond dissociation** $C_7H_{16} \rightarrow C_6H_{12} + CH_4$ from Transition1x [Schreiner et al., 2022] at B3LYP/def2-SVP using the single-stage ($T = 1$) SPICEv2 **P** model.

ξ	**Frame**	C−C (Å)	ΔE (kcal/mol)	**MINAO** (iter.)	**ERIC** (↓) SAIL
0.00	C_7H_{16}	1.53	0.00	9	0.67
0.07		1.55	2.00	9	0.67
0.20		1.65	25.96	9	0.78
0.33		2.03	79.14	10	0.80
0.48		2.23	103.15	10	0.90
0.50	transition state	2.27	103.22	10	0.90
0.52		2.27	103.29	10	0.90
0.60		2.42	103.48	10	0.90
0.90		2.95	95.39	9	1.00
1.00	$C_6H_{12} + CH_4$	3.13	94.59	9	1.00

K Robustness of SAIL

Training-seed variability. To assess variability across training seeds, we retrain the B3LYP coefficient and density-matrix models, both the surrogate-loss baselines and their SAIL fine-tunes, with three training seeds each and evaluate them on QM40 and QMugs (Table 12). Across SAIL-trained models, the seed-to-seed standard deviation of the mean ERIC lies between 0.001 and 0.010. SAIL also makes models more reproducible: the density-matrix baseline varies by 0.05 (QM40) and 0.21 (QMugs) across seeds, while its SAIL fine-tuned versions vary by at most 0.01.

Table 12: Mean ERIC over three training seeds for B3LYP/def2-SVP models trained on QM9.

Model		**QM40**		**QMugs**	
		base	SAIL	base	SAIL
Coefficients	$\mathbf{c}$	0.919 ± 0.017	0.756 ± 0.001	0.886 ± 0.005	0.771 ± 0.009
Density Matrix	$\mathbf{P}$	1.065 ± 0.051	0.696 ± 0.010	2.702 ± 0.205	0.745 ± 0.004

Convergence to the same solution. A faster-converging initial guess is only useful if the SCF reaches the same solution. Table 13 compares the final energies of SAIL- and MINAO-initialized B3LYP calculations on QM40. Even the largest deviation, 1.3×10^{-7} Ha, lies four orders of magnitude below chemical accuracy (1.6×10^{-3} Ha). The learned initialization therefore changes the path to the fixed point but not the fixed point itself.

Table 13: Deviation of the final energy of SAIL-initialized from MINAO-initialized SCF calculations on QM40 (B3LYP/def2-SVP).

Model		**MAE** [Ha]	**max. AE** [Ha]
Coefficients	$\mathbf{c}$	6.3×10^{-11}	5.5×10^{-9}
Density Matrix	$\mathbf{P}$	1.4×10^{-10}	1.3×10^{-7}
Fock	$\mathbf{F}$	1.6×10^{-10}	1.3×10^{-7}

Per-molecule robustness. Every SAIL-initialized calculation on QM40 and QMugs converges. For the density-matrix model, every molecule has an ERIC below one (maximum 0.93). For the coefficient and Hamiltonian models, a single QM40 molecule with 16 heavy atoms exceeds one (1.07 and 1.21, respectively); the density-matrix model reaches 0.93 on the same molecule.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   The abstract and introduction state the contributions of this paper precisely and within the scope of the experimental evidence presented, with caveats and assumptions delineated in the Limitations paragraph.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Limitations paragraph of Section 6, covering chemical and functional scope, evaluation regime, the role of ground-state pretraining, and the gap between ERIC and wall-time.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Hyperparameters and optimizer choices for both pretraining and SAIL finetuning are listed in Appendix H. Data splits, basis sets, density-fitting choices, convergence thresholds, and wall-time measurement procedures are documented in Appendix D. A reference implementation is publicly available at https://github.com/ESEberhard/ML4DFT.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The datasets used in our experiments are referenced with their original sources. The reference implementation, including training and evaluation scripts, experiment configurations, and environment specifications, is publicly available at https://github.com/ESEberhard/ML4DFT.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Data splits, basis sets, density-fitting choices, and convergence thresholds are described in Appendix D (Evaluation Details). Hyperparameters and the rational behind their choice for both pretraining and SAIL finetuning, including optimizer, learning rate schedule, and search ranges, are listed in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report variability across evaluation molecules and molecular-size bins. Additionally, Appendix K assesses training-seed variability for B3LYP coefficient and density-matrix models over three independent training seeds, demonstrating consistent acceleration.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: we provide this information in Appendix D.2

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the code of ethics and comply with it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: See the (unnumbered) Broader Impacts section at the end of the paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The presented models and procedures only enable quicker quantum chemical computations, and hence pose no additional risk other than accelerating potentially harmful usage of preexisting computational chemistry methods.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (QM9, QM40, QMugs) and software packages (PySCF, GPU4PySCF, EquiformerV2, QHNet, NeuralSCF) used in this work are publicly available academic releases and are cited at their points of use. We comply with the terms of their respective licenses and use them only for non-commercial research consistent with their stated terms.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper releases code via an anonymized repository included as supplemental material. Training details, hyperparameters, data splits, and evaluation procedures are documented in Appendices D and H. A reference implementation is publicly available at https://github.com/ESEberhard/ML4DFT.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.