STEPWISE REFINEMENT AND LABEL-FREE ERROR RANKING FOR KOHN–SHAM HAMILTONIANS

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine-learned Kohn–Sham Hamiltonians can bypass costly self-consistent density functional theory (DFT) calculations, but their practical use requires identifying predictions with large errors that should be recomputed with DFT. In this work, we introduce HIVE (Hamiltonian Iterative Velocity Estimator), an equivariant Hamiltonian prediction model that also provides an error ranking at no additional cost. During training, the model sees one randomly sampled point on a deterministic path from zero to the target Hamiltonian. At inference, we repeatedly apply the same field to its evolving Hamiltonian prediction, a process we call the rollout. With this strategy, averaging the stepwise estimates gives the final Hamiltonian, while their dispersion provides a label-free score for ranking likely failures without additional evaluations beyond the rollout. We test HIVE across four periodic benchmarks with PBE and HSE06 labels, and the results improve the best reported Hamiltonian mean absolute error on three and match it on the fourth. More steps improve accuracy without retraining, mainly because the finer rollout produces better estimates rather than because averaging cancels noise. On newly DFT-labelled structural shifts, the dispersion ranks per-structure error with Spearman correlations of 0.98–0.99 across all three PBE systems. Overall, HIVE provides both an accurate Hamiltonian prediction and a practical uncertainty signal without training a separate model.

1 INTRODUCTION

Kohn–Sham density functional theory (DFT) is the workhorse of electronic structure calculations for materials and molecules (Hohenberg and Kohn, 1964; Kohn and Sham, 1965; Jones, 2015). Its computational cost is dominated by the self-consistent cycle, which repeatedly builds and diagonalizes the Hamiltonian. This cost limits high-throughput screening and large-scale or long-time simulations. Equivariant machine-learning models provide a faster alternative by predicting the converged Hamiltonian $H^\star(\mathcal{S})$ directly from an atomic structure $\mathcal{S}$ in a localized atomic-orbital basis (Schütt et al., 2019; Unke et al., 2021; Nigam et al., 2022; Li et al., 2022; Gong et al., 2023). A growing number of Hamiltonian prediction methods continuously improve the accuracy on Hamiltonian prediction benchmarks (Yu et al., 2023; Luo et al., 2025; Li et al., 2025; Yu et al., 2026; Wang et al., 2024a; Liang et al., 2026; Qian et al., 2026).

High average accuracy, however, is not sufficient for practical use. A model must also identify which individual predictions are unreliable and should instead be recomputed with DFT. This need becomes more important as Hamiltonian models expand towards broad chemical and structural domains (Wang et al., 2024b; Yin et al., 2026), where an error-ranking score could guide selection of structures for DFT verification or active learning (Zhang et al., 2020). One option is an ensemble of independently trained models, whose disagreement provides an error estimate (Lakshminarayanan et al., 2017), but this approach multiplies both training and inference costs.

In this work, we introduce HIVE (Hamiltonian Iterative Velocity Estimator), a shared-weight equivariant model that repeatedly estimates the same target Hamiltonian. HIVE is trained along a deterministic straight path from zero to the normalized target Hamiltonian. For each training example, it samples one path coordinate and predicts the constant target associated with that path, requiring only one field evaluation and no unrolled trajectory. At inference, the same field is evaluated K times while updating an intermediate Hamiltonian, and each evaluation produces a separate estimate of

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

the same normalized target. Their mean yields the final Hamiltonian prediction, and their dispersion D_M measures how consistently the model predicts that target along its own rollout.

This construction gives HIVE both a refinement mechanism and an intrinsic error-ranking score. Increasing the number of steps can improve the prediction without changing the model weights, while D_M requires no field evaluations beyond those already used to produce that prediction. It is important to note that the error-ranking score is interpreted as a ranking signal rather than a calibrated error estimate, so that a large value indicates inconsistent step estimates, but a small value cannot exclude an error shared by every step.

HIVE is related to flow matching strategies through its use of a straight path, but its purpose and construction differ from generative flow models. QHFlow, for example, learns a conditional distribution from a stochastic source (Kim et al., 2025), whereas HIVE begins from a fixed zero state and targets a deterministic Hamiltonian, making the outputs at different steps directly comparable. HIVE also differs from iterative Hamiltonian predictors based on self-consistency, whether self-consistency is used as a training signal (Zhang et al., 2024) or as a fixed-point iteration (Wang et al., 2024c; Lou et al., 2026). In HIVE, every step estimates the same target, enabling one rollout to produce both the prediction and its error-ranking score.

Our contributions are:

- **A shared-weight prediction and ranking method.** HIVE combines repeated estimates from one equivariant field: their mean produces the Hamiltonian, and their dispersion ranks predictions for DFT verification without an additional model or additional field evaluations.
- **A characterization of stepwise refinement.** We derive exact decompositions that separate disagreement among the step estimates from error shared across them. These results clarify what D_M can observe and distinguish improvement from better intermediate estimates from improvement due to averaging.
- **State-of-the-art accuracy and structural-shift benchmarks.** On four periodic benchmarks with PBE and HSE06 labels, HIVE improves the strongest reported Hamiltonian MAE on three and matches it on the fourth. Additional inference steps improve accuracy without retraining, primarily because the estimates along the finer rollout become more accurate rather than because averaging removes their disagreement. On newly DFT-labelled structural shifts, D_M ranks per-structure prediction error with Spearman correlations of 0.98–0.99 across all three PBE systems and five displacement levels spanning approximately three decades of error.

2 Stepwise Refinement and Consistency

2.1 Hamiltonian target and deterministic path

For an atomic structure $\mathcal{S}$, the prediction target is its converged Kohn–Sham Hamiltonian $H^\star(\mathcal{S})$. In a localized atomic-orbital basis, the Hamiltonian consists of real-space blocks $H^\star_{ij}(\mathbf{R})$, whose matrix elements

$$[H^\star_{ij}(\mathbf{R})]_{\mu\nu} = \langle \phi_{i\mu} | \hat{H}_{\mathcal{S}} | \phi_{j\nu,\mathbf{R}} \rangle \tag{1}$$

couple orbital μ on atom i to orbital ν on the image of atom j displaced by $\mathbf{R}$. The on-site block $H^\star_{ii}(\mathbf{0})$ couples orbitals on one atom and all other blocks couple orbitals between atoms.

These Hamiltonian blocks obey the physical symmetries of the atomic-orbital representation: they transform equivariantly under rotations, remain unchanged under rigid translations, reorder under atom permutations, and satisfy the Hermiticity relation $H_{ji}(-\mathbf{R}) = H_{ij}(\mathbf{R})^\top$. Representation conventions and implementation details are given in Appendix B.2.

To obtain several comparable estimates from one model, we define a deterministic straight path from zero to the normalized target, similar to the straight conditional paths of flow matching (Lipman et al., 2023) but starting from a fixed zero state rather than a noise distribution:

$$Q(\mathcal{S}) = \frac{H^\star(\mathcal{S})}{\sigma_H}, \qquad \widetilde{H}_t = tQ(\mathcal{S}), \qquad \frac{\mathrm{d}\widetilde{H}_t}{\mathrm{d}t} = Q(\mathcal{S}), \tag{2}$$

where σ_H is a fixed normalization scale estimated from the training set and then frozen. The coordinate $t \in [0,1]$ specifies position along the path and it is not physical time. The intermediate Hamiltonian $\widetilde{H}_t$ changes with t, but its derivative is the same normalized target velocity $Q(\mathcal{S})$ everywhere along the path.

We train a field $\boldsymbol{f}_\theta(H, t; \mathcal{S})$ to estimate $Q(\mathcal{S})$ from the atomic structure, the current Hamiltonian iterate, and the path coordinate. Because every point on the path has the same target, field outputs obtained at different coordinates can be directly averaged and compared.

To accommodate species with different numbers of orbitals, we pad the Hamiltonian blocks and introduce a binary mask M, where $M_a = 1$ for physical entries and $M_a = 0$ for padding. The normalized target satisfies $Q = Q \odot M$, and training and evaluation use the normalized masked norm

$$\|x\|_M^2 = \frac{1}{\|M\|_1} \sum_a M_a x_a^2. \tag{3}$$

2.2 Single-evaluation training and shared-weight rollout

For each training example, we sample $t \sim \mathcal{U}(0,1)$, construct the corresponding teacher iterate $tQ(\mathcal{S})$, and minimize

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{S},\, t\sim\mathcal{U}(0,1)} \left\| \boldsymbol{f}_\theta(tQ(\mathcal{S}), t; \mathcal{S}) - Q(\mathcal{S}) \right\|_M^2. \tag{4}$$

With this strategy, training requires one field evaluation per sampled structure and path coordinate. The teacher iterate is available analytically, so the trajectory need not be simulated or differentiated through.

At inference, we instead evaluate the field on its own evolving iterates. Starting from zero, K uniform Euler steps with $t_k = k/K$ give

$$\widetilde{H}^{(k+1)} = \left[\widetilde{H}^{(k)} + \frac{1}{K} \boldsymbol{f}_\theta\left(\widetilde{H}^{(k)}, t_k; \mathcal{S} \right) \right] \odot M, \qquad \widetilde{H}^{(0)} = 0, \tag{5}$$

The final prediction in physical units is

$$\widehat{H} = \sigma_H \widetilde{H}^{(K)}. \tag{6}$$

The same parameters are reused at every step, so the inference step count can be changed without retraining. Each field evaluation produces another estimate of the same target $Q(\mathcal{S})$, rather than a different residual or step-specific target. Consequently, the final normalized iterate is the average of the step estimates. Section 2.3 below uses this observation to define the prediction and its stepwise-consistency score.

We note that the teacher path also admits the formal relation $Q(\mathcal{S}) = \widetilde{H}_t/t$ for $t > 0$, which raises the possibility of an iterate-only rescaling shortcut. Such a shortcut becomes singular near $t = 0$, where the iterate contains vanishing information about the target Hamiltonian and a bounded field must rely on the structure. Appendix D.2 gives the corresponding argument and a swap probe confirming that the trained field remains strongly structure-dependent.

2.3 Prediction by averaging and stepwise consistency

Let q_k denote the normalized-target estimate produced at step k, and let $\overline{q}$ be the mean of the K estimates:

$$q_k = \boldsymbol{f}_\theta\left(\widetilde{H}^{(k)}, t_k; \mathcal{S} \right), \qquad \overline{q} = \frac{1}{K} \sum_{k=0}^{K-1} q_k, \qquad k = 0, \ldots, K-1. \tag{7}$$

Because every q_k estimates the same target $Q(\mathcal{S})$, their disagreement can be measured directly, and we therefore define the masked root-mean-square dispersion

$$D_M^2 = \frac{1}{K} \sum_{k=0}^{K-1} \|q_k - \overline{q}\|_M^2. \tag{8}$$

The mean $\overline{q}$ determines the final Hamiltonian prediction, while the dispersion D_M measures the consistency of the estimates produced along the rollout. Figure 1 illustrates this geometry.

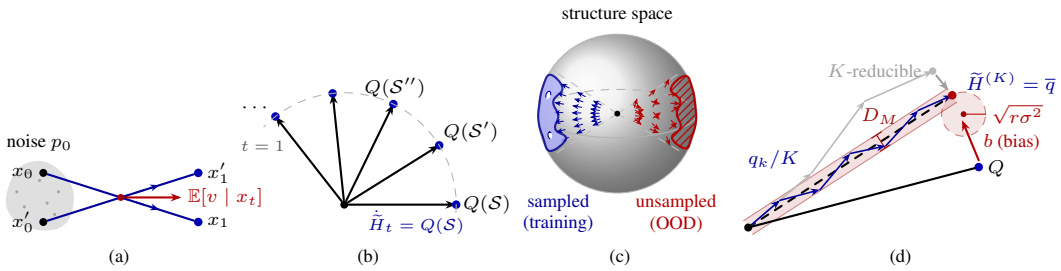

Figure 1: Geometry of the stepwise estimates. **(a)** Generative flow matching: conditional paths from noise to data can meet at the same (x_t, t), where the learned marginal velocity is the expectation of the conditional ones. **(b)** Our setting: every structure leaves the same source $\widetilde{H}_0 = 0$ along a unit-time straight path of constant velocity $Q(\mathcal{S})$, so conditioning on $\mathcal{S}$ leaves no competing target. **(c)** The ball is Hamiltonian space, its centre $\widetilde{H}_0 = 0$; one ray per structure, angle labelling the structure and radius the path coordinate t. Sectors mark the training region (blue), unsampled in-distribution pockets (dashed) and an unsampled out-of-distribution region (hatched). Short arrows are the learned velocity: radial inside the training region, so the K estimates agree and averaging helps; scattered outside it, so the same average exposes their disagreement. **(d)** A K-step rollout ($K = 6$) chains estimates q_k/K and ends at $\overline{q} = \widetilde{H}^{(K)}$; the coarser $K = 3$ rollout (gray) is drawn landing farther from Q. D_M is the spread of one sample's q_k; the disc is a population quantity, centred at $Q + b$ with radius $\sqrt{r\sigma^2}$, the residual that averaging cannot remove (Proposition 2).

Proposition 1 (Per-structure averaging and error decomposition)**.** *For any structure $\mathcal{S}$, the uniform Euler rollout in Eq. (5) satisfies*

$$\widetilde{H}^{(K)} = \overline{q} \odot M, \tag{9}$$

Moreover, the mean squared error of the individual step estimates decomposes exactly as

$$\frac{1}{K}\sum_{k=0}^{K-1} \|q_k - Q\|_M^2 = D_M^2 + \|\widetilde{H}^{(K)} - Q\|_M^2. \tag{10}$$

The proof is given in Appendix A.1. Equation (9) follows by unrolling the equal-step Euler updates from zero: the final normalized iterate is the masked mean of the step estimates. Equation (10) is the corresponding bias–variance identity for a fixed structure. It separates the error of the individual estimates into their disagreement, D_M^2, and the error that remains after averaging.

The identity in Eq. (10) clarifies both the value and the limitation of D_M. A large value means that the model gives inconsistent estimates of the same target along its rollout and therefore provides evidence that the prediction is unreliable. A small value, however, does not imply accuracy, as all steps may agree on the same incorrect target. We therefore use D_M as an error-ranking score, not as a calibrated estimate of the prediction error.

A population-level decomposition further explains when averaging improves the final prediction. For a fixed step count K, let

$$\delta_k = q_k - Q \tag{11}$$

be the error of step k. Let $b = \frac{1}{K}\sum_k \mathbb{E}[\delta_k]$ be the step-averaged bias, let σ^2 be the average squared fluctuation about b, and let r be the average pairwise correlation between those fluctuations. The expected terminal error then satisfies

$$\mathbb{E}\|\widetilde{H}^{(K)} - Q\|_M^2 = \underbrace{\|b\|_M^2}_{\text{common bias}} + \underbrace{\frac{1-r}{K}\sigma^2}_{K\text{-reducible component}} + \underbrace{r\sigma^2}_{\text{cross-step correlation}}, \tag{12}$$

while the expected dispersion is

$$\mathbb{E}D_M^2 = \frac{K-1}{K}\underbrace{(1-r)\sigma^2}_{\text{non-common stepwise error}}. \tag{13}$$

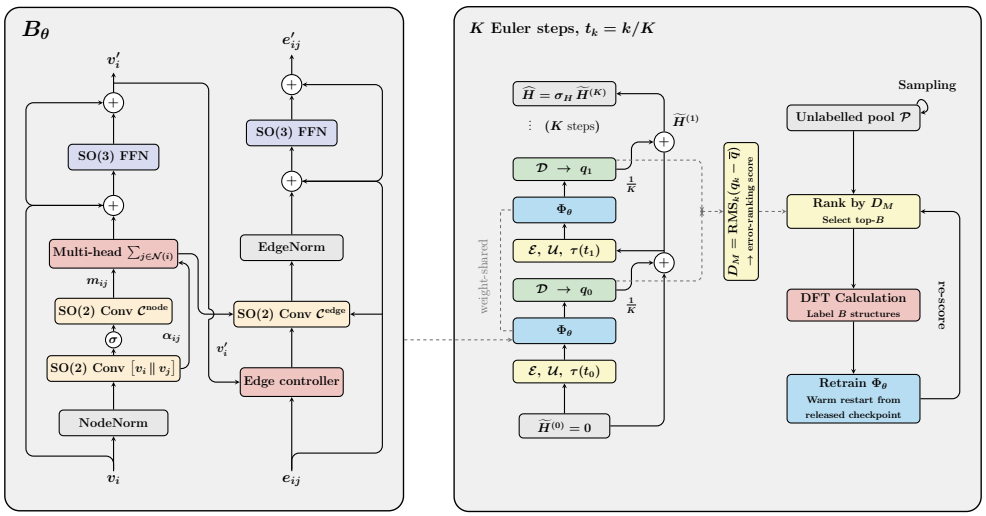

Figure 2: One shared vector field produces both the stepwise estimates and the Hamiltonian prediction. **Left:** one constituent $SO(3)$-equivariant interaction block B_θ; one or more such blocks form the interaction network Φ_θ. **Right:** the complete Φ_θ is reused at every integration step; starting from $\widetilde{H}^{(0)} = 0$, each Euler step decodes a target estimate q_k and updates the Hamiltonian state. Since every q_k estimates the same constant target velocity, their masked dispersion defines the stepwise-consistency score $D_M(\mathcal{S})$ in Eq. (8); it is a label-free uncertainty score, not a calibrated estimate of predictive error.

Proposition 2 (Population error decomposition)**.** *At fixed K, define δ_k, b, σ^2, and r under the masked population measure specified in Appendix A.2. Then Eqs. (12)–(13) hold.*

Equation (12) separates three contributions to the terminal error. The first is bias shared on average across the steps; the second is the non-common component that averaging can suppress; and the third arises from correlated errors and remains after averaging. Equation (13) shows that D_M measures the non-common component: it is large when the step estimates disagree, but it does not reveal bias or error shared across the rollout.

These equations apply separately at each value of K. Increasing K changes both the path coordinates at which the field is evaluated and the intermediate iterates it encounters, so b, σ^2, and r may also change. The decomposition therefore does not imply that a longer rollout must always be more accurate. Instead, it allows us to distinguish improvement caused by averaging from improvement caused by better estimates along the finer rollout.

Finally, we note that the score preserves the physical symmetry of the prediction. If the field is rotation equivariant and the mask contains complete angular-momentum multiplets, then $D_M(g\mathcal{S}) = D_M(\mathcal{S})$ for any rotation g, with the direct verification in Appendix A.1. All q_k are already produced during the prediction rollout, so evaluating D_M requires only a reduction over those outputs and no additional field evaluations or independently trained models.

2.4 Shared Equivariant Field

The field $\boldsymbol{f}_\theta(\widetilde{H}_t, t; \mathcal{S})$ combines three inputs: the atomic geometry $\mathcal{S}$, the current Hamiltonian iterate $\widetilde{H}_t$, and the path coordinate t. We represent the structure as a graph in which on-site Hamiltonian blocks are associated with nodes and off-site blocks with edges. The geometry supplies the initial node $\boldsymbol{n}^0$ and edge $\boldsymbol{e}^0$ features.

The current Hamiltonian blocks are converted into irreducible $SO(3)$ features through a Clebsch–Gordan encoding $\mathcal{E}$, followed by learned projections $\mathcal{U}_\text{n}$ and $\mathcal{U}_\text{e}$ into the node and edge channels (Gong et al., 2023; Geiger and Smidt, 2022). A sinusoidal embedding followed by an MLP produces the path-coordinate features $\tau(t)$. Because t is a scalar, these features are added only to

the $l = 0$ channels:

$$\boldsymbol{n}_t = \boldsymbol{n}^0 + \mathcal{U}_{\mathrm{n}}\mathcal{E}\big(\widetilde{H}_t^{\mathrm{node}}\big) + \tau(t)\boldsymbol{\delta}_{l0}, \qquad \boldsymbol{e}_t = \boldsymbol{e}^0 + \mathcal{U}_{\mathrm{e}}\mathcal{E}\big(\widetilde{H}_t^{\mathrm{edge}}\big) + \tau(t)\boldsymbol{\delta}_{l0}, \tag{14}$$

An equivariant interaction network Φ_θ processes these features together with the fixed geometric information $\mathcal{G}(\mathcal{S})$, and a decoder $\mathcal{D}$ maps its output back to Hamiltonian blocks:

$$\boldsymbol{f}_\theta(\widetilde{H}_t, t; \mathcal{S}) = \mathcal{D}(\Phi_\theta(\boldsymbol{n}_t, \boldsymbol{e}_t; \mathcal{G}(\mathcal{S}))) \,. \tag{15}$$

The interaction network combines the $SO(2)$ convolution used by eSCN (Passaro and Zitnick, 2023) with the equivariant attention mechanism of EquiformerV2 (Liao et al., 2024). Reverse off-site blocks are reconstructed by transposition, $H_{ji}(-\mathbf{R}) = H_{ij}(\mathbf{R})^\top$, which enforces their Hermiticity relation. On-site blocks are predicted directly, and their symmetry is learned. Appendix B.3 describes the interaction blocks and the path embedding, while Appendix B.2 gives the encoder and decoder details.

The entire interaction network is reused with the same parameters at every rollout step. Geometry-dependent features can therefore be cached, while only the Hamiltonian iterate and path coordinate change between evaluations. The network depth controls the capacity of each evaluation, whereas K controls the number of shared-field evaluations performed at inference. Thus, changing K changes the inference budget without introducing new parameters or requiring retraining. Figure 2 summarizes this shared-weight computation.

3 Experiments

We test HIVE with three questions. First, how accurately does it predict Hamiltonians, and how does accuracy change when the same checkpoint is evaluated with more rollout steps? Second, can the label-free dispersion D_M rank prediction errors on both test structures and controlled structural shifts? Third, which part of the rollout accounts for the improvement obtained from additional steps?

3.1 Predictive accuracy and fixed-weight refinement

We evaluate HIVE on four periodic benchmarks: monolayer graphene, monolayer MoS_2, and bilayer graphene with PBE labels (Perdew et al., 1996), and bilayer graphene with HSE06 hybrid-functional labels (Tang et al., 2024). Dataset definitions, splits, and baseline sources are given in Appendix B.1.

Table 1 compares the test-set Hamiltonian MAE at $K = 8$ with previously reported results. HIVE improves the strongest reported value on monolayer MoS_2 and on both bilayer graphene benchmarks, while matching the reported precision on monolayer graphene. Specifically, its MAE is approximately half the Hot-Ham value on MoS_2 and 22% lower on bilayer graphene with PBE labels, and compared to DeepH-hybrid it is 25% lower on bilayer graphene with HSE06 labels.

These results use one checkpoint per system. To isolate the effect of inference-time computation, we hold the model parameters fixed and vary only the number of Euler steps. Increasing K from one to four reduces the MAE on every benchmark, including a 41.6% reduction on monolayer MoS_2. The largest improvements occur within the first few evaluations. Monolayer graphene, MoS_2, and bilayer graphene with HSE06 labels then approach plateaus, whereas bilayer graphene with PBE labels reaches its minimum at an intermediate K and increases slightly afterward. Thus, K provides a test-time accuracy–cost trade-off without retraining, although a longer rollout is not guaranteed to be more accurate. The complete sweep is reported in Table 5.

3.2 Ranking prediction error under structural shift

We next test the main practical use of D_M: identifying predictions that should be verified with DFT. This requires labels that were not available when the model was trained. To build them, for each PBE benchmark we select eighteen validation structures, apply displacements with amplitudes $\varsigma \in \{0, 0.03, 0.1, 0.3, 1\}$ Å, and calculate new OpenMX reference Hamiltonians for all ninety resulting structures. We then compare D_M at $K = 8$ with the per-structure Hamiltonian MAE of the same prediction. The displacement protocol and DFT settings are given in Appendix C.1.

Table 1: Test-set accuracy. Hamiltonian MAE (meV) reported in the literature alongside ours at $K = 8$; the full sweep over K is in Table 5 (Appendix B.3). The error decomposition at the same K is in Table 2. The best result in each column is bold. On monolayer graphene, Hot-Ham reports only two decimals, and at that precision it equals our 0.0665, so both are bold. ‡Third-party retraining; dashes mark unavailable results. Datasets and baseline sources are in Appendix B.1.

Method	Monolayer Graphene	Monolayer MoS_2	Bilayer Graphene (PBE)	Bilayer Graphene (HSE)
DeepH	2.10	1.00	1.9‡	–
DeepH-E3	0.40	0.46	0.41‡	–
HamGNN	0.35‡	1.20‡	–	–
DeePTB	0.14	0.14	–	–
DeepH-2	0.12	0.21	0.17	–
Hot-Ham	**0.07**	0.12	0.15	–
DeepH-hybrid	–	–	–	0.147
HIVE (ours), $K = 8$	**0.0665**	**0.0626**	**0.1172**	**0.1100**

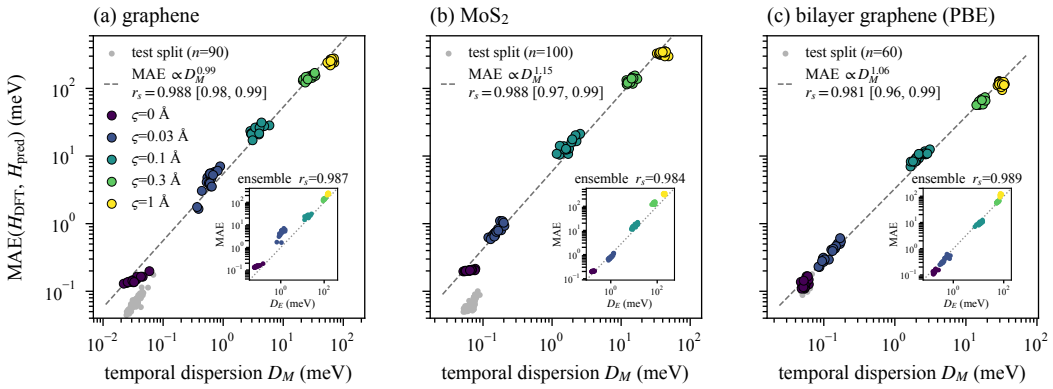

Figure 3: Temporal dispersion against true error. For each PBE/OpenMX benchmark, eighteen validation structures displaced by $\varsigma \in \{0, 0.03, 0.1, 0.3, 1\}$ Å and relabelled with OpenMX: the label-free score D_M ($K = 8$) against the true per-structure MAE of the same predictions. Grey points are the in-distribution test split at the same K; dashed lines are log–log fits; r_s is the Spearman correlation over the 90 relabelled structures, with a bootstrap 95% interval over parent structures. The insets repeat the same comparison for a three-member deep ensemble of independently trained models, using the cross-member dispersion D_E in place of D_M and the error of the ensemble mean, so their absolute scale is not comparable with the main panels. Displacement and relabelling protocol, exact values and selective-prediction results: Appendix C.1; ensemble details: Appendix C.3.

Figure 3 shows that D_M closely tracks the true error under these controlled shifts. Across the five displacement levels, the error spans approximately three orders of magnitude, while the log–log relation between D_M and MAE has a slope close to one. The Spearman correlations are 0.988, 0.988, and 0.981 for monolayer graphene, MoS_2, and bilayer graphene, respectively. Thus, a score computed without reference labels recovers almost the same ordering as the true prediction error in all three systems.

The score also contains information within individual displacement levels. Up to 0.3 Å, the ordering produced by D_M largely agrees with the ordering of the true errors, with within-level rank correlations above 0.7 in most cells (Table 7). This effect is not explained merely by differences in displacement magnitude: within each level, those magnitudes vary by only about 4%, and ranking by displacement is usually less effective than ranking by D_M. At 1 Å, all predictions have errors comparable to the hopping integrals themselves and are already unusable. In this regime, D_M continues to flag these structures but no longer orders their errors reliably.

The newly calculated labels introduce a small offset relative to the released benchmark labels, even for the undisplaced structures. This shifts their measured errors above the original test-set cloud without affecting D_M, which never observes either label set. Appendix C.1 quantifies this offset and reports the complete within-level and selective-prediction results. On the original test splits, where the errors occupy a much narrower range, ranking by D_M recovers 57–94% of the oracle improvement in area under the risk–coverage curve (Geifman and El-Yaniv, 2017).

Equation (10) explains why D_M should be interpreted as a ranking score rather than an error magnitude. It measures disagreement among the step estimates but cannot observe an error shared by all of them. The fraction of stepwise error visible through D_M varies substantially across systems and structures, so no universal conversion from D_M to MAE exists. Nevertheless, the observable component is sufficient to rank the shifted structures with correlations above 0.98.

For comparison, we evaluate a deep ensemble of three independently trained models. Its cross-model dispersion gives similar rank correlations (0.987, 0.984, and 0.989; Fig. 3, insets), but requires three trained models and twenty-four field evaluations for an eight-step prediction. HIVE uses one model and obtains D_M from the eight evaluations already used for its prediction. Because the ensemble was not matched to HIVE under a common compute budget, this comparison concerns error-ranking ability rather than attainable predictive accuracy; further details are given in Appendix C.3.

3.3 Why additional steps improve accuracy

Having established that the rollout improves prediction and provides a useful ranking score, we finally ask where the accuracy gain from a larger K comes from. The population decomposition in Eqs. (12) and (13) distinguishes two mechanisms. First, averaging can suppress the non-common component on which the step estimates disagree. Second, changing K changes the path coordinates and intermediate iterates seen by the field, and can therefore change the bias b, fluctuation scale σ^2, and correlation r of the estimates themselves.

Figure 4 decomposes the reduction in terminal MSE relative to $K = 1$ into three terms: the reduction in σ^2, the component removed by averaging, and the reduction in $\|b\|_M^2$. These contributions sum exactly to the total reduction at each K. These results indicate that averaging is helpful but is never the dominant source of improvement. On monolayer graphene, it accounts for 36% of the gain at $K = 2$ and 21% at $K = 8$, with similar behavior on the other systems. By $K = 8$, the component that further averaging could remove is at most 6.3% of the terminal MSE across the four benchmarks (Table 2). Most of the improvement instead comes from lower stepwise fluctuation and, to a lesser extent, lower common bias. In other words, a finer rollout improves accuracy mainly because the field produces better estimates along the altered sequence of iterates, not simply because more estimates are averaged.

This analysis concerns the error of the field outputs, rather than the distance between each intermediate Hamiltonian and the final target. A reduction in σ^2 is therefore consistent with, but does not by itself prove, that the finer rollout supplies more informative intermediate iterates.

Behavior beyond $K = 8$ is system dependent. MoS_2 saturates by approximately $K = 16$, bilayer graphene with HSE06 labels continues to improve slowly, and bilayer graphene with PBE labels becomes marginally worse as σ^2 increases. One possible explanation is that a long rollout drifts away from the teacher iterates encountered during training, although we have not measured this directly. These results show that additional steps are a useful refinement mechanism, but continued improvement will ultimately require a better field rather than an indefinitely longer rollout.

4 Discussion and limitations

HIVE turns an iterative Hamiltonian predictor into a sequence of comparable estimates of one target. Reusing the same equivariant field provides two benefits: averaging the estimates produces an accurate Hamiltonian, while their disagreement provides a label-free score for ranking likely failures. The refinement analysis shows that most of the accuracy gain from additional steps comes from better estimates along the finer rollout, rather than from averaging alone. The benefit therefore saturates after relatively few steps and is not guaranteed to be monotonic. Increasing K offers a useful

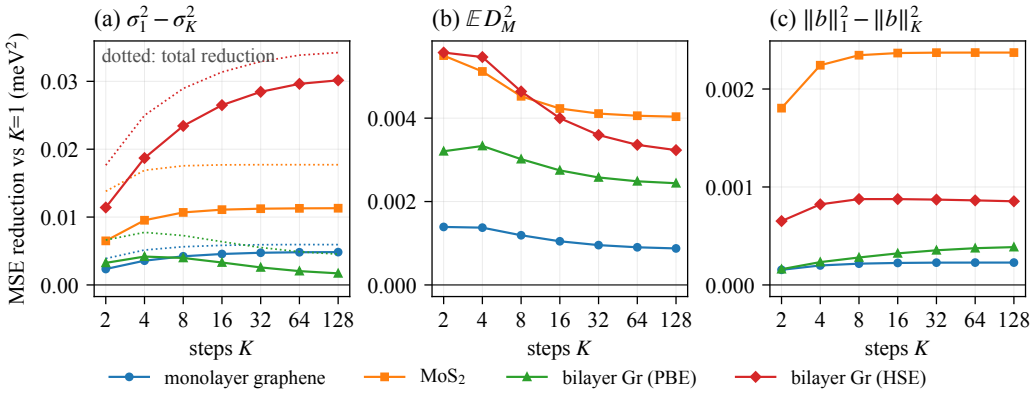

Figure 4: Where the gain from additional steps comes from. For each system the reduction of the test-set terminal MSE relative to $K = 1$ is decomposed exactly, by Eqs. (12)–(13), into three parts, one per panel: **(a)** the part due to a smaller stepwise variance, $\sigma_1^2 - \sigma_K^2$ (dotted lines: the total reduction); **(b)** the part removed by averaging over steps, $(1 - r)\sigma_K^2(1 - 1/K)$; **(c)** the part due to a smaller common bias, $\|b\|_1^2 - \|b\|_K^2$. The three sum to the total reduction. Quantities are the bias-corrected estimates of Appendix A.2, whose sampling correction preserves the decomposition identity exactly.

Table 2: Decomposition of the test-set terminal MSE at $K = 8$ into its sources, following Proposition 2. Components are percentages of the terminal MSE; r is the average pairwise correlation of the step errors. Estimation details are in Appendix A.2.

	Monolayer Graphene	Monolayer MoS_2	Bilayer Graphene (PBE)	Bilayer Graphene (HSE)
Terminal MSE (meV^2)	0.0357	0.0103	0.1250	0.0603
Common bias (%)	1.18	4.22	0.92	1.84
K-reducible component (%)	0.48	6.25	0.34	1.10
Cross-step correlation (%)	98.35	89.53	98.74	97.06
Average correlation r	0.963	0.642	0.973	0.917

test-time accuracy–cost trade-off, but it cannot substitute indefinitely for improving the underlying field.

The interpretation of D_M is deliberately limited. It measures disagreement among the step estimates and cannot detect an error shared by all of them. A large D_M provides evidence that a prediction is unreliable, whereas a small value does not certify accuracy. Its scale also varies across systems and types of shift, so it should be used to rank structures rather than as a calibrated prediction of MAE. Any operational threshold for accepting a prediction or triggering a DFT calculation would require labelled calibration for the intended domain.

Our empirical evidence has a correspondingly defined scope. We evaluate error ranking under isotropic displacements in three PBE/OpenMX datasets, using one model trained on each mature dataset. These experiments do not establish performance under broader changes in composition, bonding, phase, or electronic-structure method. They also do not test the early stages of active learning, where a model trained on a small and scattered set of configurations may not yet produce a smooth field. Evaluating D_M in these settings, and determining how its ranking quality evolves as training data are acquired, are important directions for future work.

The same principle may extend beyond HIVE to other iterative predictors, provided that their intermediate outputs are comparable estimates of a common target. Under that condition, stepwise consistency can supply a failure-ranking signal from computations already required for prediction. HIVE demonstrates this principle for Kohn–Sham Hamiltonians, combining competitive accuracy with an intrinsic mechanism for prioritizing structures for DFT verification.

AI USE STATEMENT

AI assistance was used for code debugging, mathematical derivations, and grammatical refinement. All AI-assisted derivations were independently verified by the authors.

REFERENCES

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/4a8423d5e91fda00bb7e46540e2b0cf1-Paper.pdf.

Mario Geiger and Tess Smidt. e3nn: Euclidean Neural Networks, 2022. URL https://arxiv.org/abs/2207.09453.

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, May 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-38468-8. URL https://doi.org/10.1038/s41467-023-38468-8.

P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964. doi: 10.1103/PhysRev.136.B864. URL https://link.aps.org/doi/10.1103/PhysRev.136.B864.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=DNdN26m2Jk.

R. O. Jones. Density functional theory: Its origins, rise to prominence, and future. *Rev. Mod. Phys.*, 87:897–923, Aug 2015. doi: 10.1103/RevModPhys.87.897. URL https://link.aps.org/doi/10.1103/RevModPhys.87.897.

Semin Kim, Jaehoon Yoo, Jinwoo Kim, Yeonwoo Cha, Saehoon Kim, and Seunghoon Hong. Simulation-free training of neural ODEs on paired data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GOgKhunkfw.

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

W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965. doi: 10.1103/PhysRev.140.A1133. URL https://link.aps.org/doi/10.1103/PhysRev.140.A1133.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

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, Jun 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00265-6. URL https://doi.org/10.1038/s43588-022-00265-6.

Pengfei Li, Xiaohui Liu, Mohan Chen, Peize Lin, Xinguo Ren, Lin Lin, Chao Yang, and Lixin He. Large-scale ab initio simulations based on systematically improvable atomic basis. *Computational Materials Science*, 112:503–517, 2016. ISSN 0927-0256. doi: https://doi.org/10.1016/j.commatsci.2015.07.004. URL https://www.sciencedirect.com/science/article/pii/S0927025615004140.

Yunyang Li, Zaishuo Xia, Lin Huang, Xinran Wei, Samuel Harshe, Han Yang, Erpai Luo, Zun Wang, Jia Zhang, Chang Liu, Bin Shao, and Mark Gerstein. Enhancing the Scalability and Applicability of Kohn-Sham Hamiltonians for Molecular Systems. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 918–962, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/035a73893121b4534bb3314e831050b1-Paper-Conference.pdf.

Zhixin Liang, Yunlong Wang, Chi Ding, Junjie Wang, Hui-Tian Wang, Dingyu Xing, and Jian Sun. Hot-ham: an accurate and efficient e(3)-equivariant machine-learning electronic structures calculation framework. *Chinese Physics Letters*, 43(2):020704, jan 2026. doi: 10.1088/0256-307X/43/2/020704. URL https://doi.org/10.1088/0256-307X/43/2/020704.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 39282–39309, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/ab12e8f3443c1a789f595b18d8c597b4-Paper-Conference.pdf.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yunhong Lou, Xihang Yue, Xinran Wei, Tianqi Deng, and Linchao Zhu. A Fixed-Point Neural Operator for Size- and Functional-Transferable Hamiltonian Prediction, 2026. URL https://arxiv.org/abs/2606.14498.

Erpai Luo, Xinran Wei, Lin Huang, Yunyang Li, Han Yang, Zaishuo Xia, Zun Wang, Chang Liu, Bin Shao, and Jia Zhang. Efficient and Scalable Density Functional Theory Hamiltonian Prediction through Adaptive Sparsity. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 41368–41390. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/luo25l.html.

Jigyasa Nigam, Michael J. Willatt, and Michele Ceriotti. Equivariant representations for molecular hamiltonians and n-center atomic-scale properties. *The Journal of Chemical Physics*, 156(1): 014115, 01 2022. ISSN 0021-9606. doi: 10.1063/5.0072784. URL https://doi.org/10.1063/5.0072784.

T. Ozaki. Variationally optimized atomic orbitals for large-scale electronic structures. *Phys. Rev. B*, 67:155108, Apr 2003. doi: 10.1103/PhysRevB.67.155108. URL https://link.aps.org/doi/10.1103/PhysRevB.67.155108.

Saro Passaro and C. Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient equivariant GNNs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27420–27438. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/passaro23a.html.

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

Chen Qian, Valdas Vitartas, James R. Kermode, and Reinhard J. Maurer. Equivariant electronic hamiltonian prediction with many-body message passing. *npj Computational Materials*, 12(1): 169, Mar 2026. ISSN 2057-3960. doi: 10.1038/s41524-026-02020-1. URL https://doi.org/10.1038/s41524-026-02020-1.

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, Nov 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-12875-2. URL https://doi.org/10.1038/s41467-019-12875-2.

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, Oct 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-53028-4. URL https://doi.org/10.1038/s41467-024-53028-4.

Oliver Unke, Mihail Bogojeski, Michael Gastegger, Mario Geiger, Tess Smidt, and Klaus-Robert Müller. Se(3)-equivariant prediction of molecular wavefunctions and electronic densities. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14434–14447. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/78f1893678afbeaa90b1fa01b9cfb860-Paper.pdf.

Yuxiang Wang, He Li, Zechen Tang, Honggeng Tao, Yanzhen Wang, Zilong Yuan, Zezhou Chen, Wenhui Duan, and Yong Xu. Deeph-2: Enhancing deep-learning electronic structure via an equivariant local-coordinate transformer, 2024a. URL https://arxiv.org/abs/2401.17015.

Yuxiang Wang, Yang Li, Zechen Tang, He Li, Zilong Yuan, Honggeng Tao, Nianlong Zou, Ting Bao, Xinghao Liang, Zezhou Chen, Shanghua Xu, Ce Bian, Zhiming Xu, Chong Wang, Chen Si, Wenhui Duan, and Yong Xu. Universal materials model of deep-learning density functional theory Hamiltonian. *Science Bulletin*, 69(16):2514–2521, August 2024b. ISSN 2095-9273. doi: 10.1016/j.scib.2024.06.011. URL http://dx.doi.org/10.1016/j.scib.2024.06.011.

Zun Wang, Chang Liu, Nianlong Zou, He Zhang, Xinran Wei, Lin Huang, Lijun Wu, and Bin Shao. Infusing self-consistency into density functional theory hamiltonian prediction via deep equilibrium models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024c. URL https://openreview.net/forum?id=PSVkinBs4u.

Shi Yin, Zujian Dai, Xinyang Pan, and Lixin He. Advancing Universal Deep Learning for Electronic-Structure Hamiltonian Prediction of Materials, 2026. URL https://arxiv.org/abs/2509.19877.

Haiyang Yu, Zhao Xu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Efficient and Equivariant Graph Networks for Predicting Quantum Hamiltonian. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 40412–40424. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/yu23i.html.

Haiyang Yu, Yuchao Lin, Xuan Zhang, Xiaofeng Qian, and Shuiwang Ji. Efficient Prediction of SO(3)-Equivariant Hamiltonian Matrices via SO(2) Local Frames, 2026. URL https://arxiv.org/abs/2506.09398.

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. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 59329–59357. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhang24ak.html.

Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and Weinan E. Dp-gen: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253:107206, 2020. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2020.107206. URL https://www.sciencedirect.com/science/article/pii/S001046552030045X.

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

A Mathematical results

This appendix proves the step-averaging identities introduced in Sec. 2.3, establishes their behavior under rotations, and describes the finite-sample correction used in the empirical decomposition.

A.1 Per-structure averaging and rotation invariance

Proof of Proposition 1. Recall that

$$q_k(\mathcal{S}) = \boldsymbol{f}_\theta\left(\widetilde{H}^{(k)}, t_k; \mathcal{S}\right), \qquad \overline{q}(\mathcal{S}) = \frac{1}{K}\sum_{k=0}^{K-1} q_k(\mathcal{S}). \tag{16}$$

Because the rollout starts from $\widetilde{H}^{(0)} = 0$ and the update in Eq. (5) is masked at every step, induction gives

$$\widetilde{H}^{(k)} = \widetilde{H}^{(k)} \odot M \qquad \text{for all } k. \tag{17}$$

Unrolling the K uniform Euler steps therefore yields

$$\widetilde{H}^{(K)} = \frac{1}{K}\sum_{k=0}^{K-1} q_k \odot M = \overline{q} \odot M, \tag{18}$$

which proves Eq. (9).

To obtain the error decomposition, write $q_k - Q = (q_k - \overline{q}) + (\overline{q} - Q)$. Averaging the squared masked norm over the rollout gives

$$\frac{1}{K}\sum_{k=0}^{K-1} \|q_k - Q\|_M^2 = D_M^2 + \|\overline{q} - Q\|_M^2, \tag{19}$$

where the cross-term vanishes because

$$\frac{1}{K}\sum_k \langle q_k - \overline{q}, \overline{q} - Q\rangle_M = \left\langle \frac{1}{K}\sum_k (q_k - \overline{q}), \overline{q} - Q \right\rangle_M = 0. \tag{20}$$

Since the masked norm ignores padded coordinates, $\|x \odot M\|_M = \|x\|_M$ for binary M. Together with $Q = Q \odot M$ and Eq. (9), this gives

$$\|\overline{q} - Q\|_M^2 = \|\overline{q} \odot M - Q\|_M^2 = \|\widetilde{H}^{(K)} - Q\|_M^2, \tag{21}$$

which proves Eq. (10). Thus, the average stepwise error separates exactly into the terminal prediction error and the disagreement among the step estimates. This identity holds for each structure and does not require a probabilistic assumption about the step errors. □

Rotation invariance. Let $\mathcal{T}_g$ denote the blockwise action of a rotation g on the Hamiltonian representation of Eq. (49). Assume that the learned vector field is equivariant, $\boldsymbol{f}_\theta(\mathcal{T}_g\widetilde{H}, t; g\cdot\mathcal{S}) = \mathcal{T}_g\boldsymbol{f}_\theta(\widetilde{H}, t; \mathcal{S})$, and that the mask contains complete angular-momentum multiplets, so that masking commutes with $\mathcal{T}_g$. Induction from $\widetilde{H}^{(0)} = 0$ using Eq. (5) gives

$$\widetilde{H}^{(k)}(g\mathcal{S}) = \mathcal{T}_g\widetilde{H}^{(k)}(\mathcal{S}). \tag{22}$$

It follows that

$$q_k(g\mathcal{S}) = \mathcal{T}_g q_k(\mathcal{S}). \tag{23}$$

Consequently $\overline{q}(g\mathcal{S}) = \mathcal{T}_g \overline{q}(\mathcal{S})$. Because $\mathcal{T}_g$ is an isometry of the masked inner product, preservation of the masked norm in every summand of Eq. (8) proves

$$D_M(g\mathcal{S}) = D_M(\mathcal{S}). \tag{24}$$

The target transforms in the same way, $Q(g\mathcal{S}) = \mathcal{T}_g Q(\mathcal{S})$, because $Q = H^\star/\sigma_H$ and $H^\star$ obeys Eq. (49). Hence, by the equivariance of the rollout shown above and the isometry of $\mathcal{T}_g$,

$$\big\|\widetilde{H}^{(K)}(g\mathcal{S}) - Q(g\mathcal{S})\big\|_M = \big\|\mathcal{T}_g\big(\widetilde{H}^{(K)}(\mathcal{S}) - Q(\mathcal{S})\big)\big\|_M = \big\|\widetilde{H}^{(K)}(\mathcal{S}) - Q(\mathcal{S})\big\|_M, \tag{25}$$

and the same argument applied to each $\|q_k - Q\|_M$ shows that every term of Eq. (10) is independent of the coordinate frame.

A.2 Population decomposition and finite-sample estimation

Proof of Proposition 2. Pooled population measure. Structures may carry different orbital masks, so we pool coordinate occurrences rather than weighting structures equally. Let $M_{\mathcal{S}}$ be the mask of structure $\mathcal{S}$ and $p_a = \Pr(M_{\mathcal{S},a} = 1)$. For fields x, y supported on valid coordinates, define

$$\mathbb{E}\langle x, y\rangle_M := \frac{\mathbb{E}\big[\sum_a M_{\mathcal{S},a}\, x_a y_a\big]}{\mathbb{E}\,\|M_{\mathcal{S}}\|_1}, \qquad \mathbb{E}\|x\|_M^2 := \mathbb{E}\langle x, x\rangle_M, \tag{26}$$

which weights every valid coordinate occurrence equally and reduces to the structure average of Eq. (3) when all structures share one mask. For a deterministic field c, $\|c\|_M^2 := \sum_a p_a c_a^2 / \sum_a p_a$. All expectations of masked norms and inner products in this subsection, including those in Proposition 2, are taken in this sense; the finite-sample estimators below are its empirical counterpart.

Let $\delta_k = q_k - Q$ be the error of step k, and define the step-averaged error by $\delta = \frac{1}{K}\sum_{k=0}^{K-1} \delta_k$. On valid coordinates, $\delta = \widetilde{H}^{(K)} - Q$. We define the step-averaged population bias coordinate-wise, conditionally on the coordinate being present,

$$b_a = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}[\delta_{k,a} \mid M_{\mathcal{S},a} = 1] = \mathbb{E}[\delta_a \mid M_{\mathcal{S},a} = 1], \tag{27}$$

and introduce errors relative to the step-averaged bias

$$\xi_k = \delta_k - b, \qquad \xi = \frac{1}{K}\sum_{k=0}^{K-1} \xi_k. \tag{28}$$

Although the individual ξ_k need not have zero mean, their average satisfies $\mathbb{E}[\xi_a \mid M_{\mathcal{S},a} = 1] = 0$ for every a, and hence $\mathbb{E}\langle c, \xi\rangle_M = 0$ for any deterministic field c.

Let

$$\sigma^2 = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|\xi_k\|_M^2 \tag{29}$$

denote the average centered step error. For $\sigma^2 > 0$, define the mean pairwise correlation by

$$r = \frac{1}{K(K-1)\sigma^2}\sum_{k\neq l} \mathbb{E}\langle \xi_k, \xi_l\rangle_M. \tag{30}$$

Because $\delta = b + \xi$ and $\mathbb{E}\langle b, \xi\rangle_M = 0$,

$$\mathbb{E}\|\widetilde{H}^{(K)} - Q\|_M^2 = \mathbb{E}\|\delta\|_M^2 = \|b\|_M^2 + \mathbb{E}\|\xi\|_M^2. \tag{31}$$

Expanding the second term gives

$$\mathbb{E}\|\xi\|_M^2 = \frac{1}{K^2}\sum_{k,l} \mathbb{E}\langle \xi_k, \xi_l\rangle_M = \frac{K\sigma^2 + K(K-1)r\sigma^2}{K^2} = \frac{1-r}{K}\sigma^2 + r\sigma^2. \tag{32}$$

Therefore,

$$\mathbb{E}\|\widetilde{H}^{(K)} - Q\|_M^2 = \|b\|_M^2 + \frac{1-r}{K}\sigma^2 + r\sigma^2. \tag{33}$$

This expression separates the terminal error into squared bias, a component reduced by averaging, and a component retained because of correlation between steps.

For the disagreement statistic, the standard variance identity gives

$$D_M^2 = \frac{1}{K}\sum_{k=0}^{K-1}\|\xi_k\|_M^2 - \left\|\frac{1}{K}\sum_{k=0}^{K-1}\xi_k\right\|_M^2. \tag{34}$$

Taking expectations and substituting the preceding result yields

$$\mathbb{E}D_M^2 = \frac{K-1}{K}(1-r)\sigma^2. \tag{35}$$

Hence D_M^2 measures precisely the component of the centered step error that is removed by step averaging. When the step errors are strongly correlated, averaging provides little reduction and the disagreement is correspondingly small. When their correlations are weaker, both the averaging benefit and the expected disagreement increase.

Note that no assumption of identical step distributions, equal pairwise correlations, or independence is required. Step-dependent mean offsets are included in σ^2 and r. Positive semidefiniteness of the expected Gram matrix, together with the Cauchy–Schwarz inequality, implies

$$-\frac{1}{K-1} \le r \le 1. \tag{36}$$

If $\sigma^2 = 0$, then $\xi_k = 0$ almost surely on all valid coordinates. In this degenerate case,

$$\mathbb{E}\|\widetilde{H}^{(K)} - Q\|_M^2 = \|b\|_M^2, \qquad D_M^2 = 0, \tag{37}$$

and r is left undefined. □

Finite-sample estimation. The quantities reported in Table 2 and Fig. 4 are estimated by pooling all valid coordinate occurrences in the test set. Let $\mathcal{I}_a$ be the set of test structures containing coordinate a, let $C_a = |\mathcal{I}_a|$, and let

$$N = \sum_a C_a = \sum_s \|M_s\|_1 \tag{38}$$

be the total number of valid occurrences.

For each valid structure-coordinate pair, define the terminal normalized error

$$d_{s,a} = \frac{1}{K}\sum_{k=0}^{K-1}\left(q_{s,k,a} - Q_{s,a}\right), \tag{39}$$

and its coordinate-wise sample mean

$$\widehat{b}_a = \frac{1}{C_a}\sum_{s\in\mathcal{I}_a} d_{s,a}. \tag{40}$$

For $C_a \ge 2$, let

$$s_a^2 = \frac{1}{C_a - 1}\sum_{s\in\mathcal{I}_a}\left(d_{s,a} - \widehat{b}_a\right)^2 \tag{41}$$

be the corresponding unbiased sample variance.

If $b_a = \mathbb{E}[d_{s,a} \mid (M_s)_a = 1]$, then, conditional on C_a and under independent sampling of the test structures,

$$\mathbb{E}\left[C_a\widehat{b}_a^2 \mid C_a\right] = C_a b_a^2 + \mathrm{Var}(d_{s,a} \mid (M_s)_a = 1). \tag{42}$$

Thus, the direct plug-in estimate of the pooled squared bias is upward biased. We estimate this excess by

$$\widehat{\epsilon} = \frac{1}{N} \sum_{a:C_a \geq 2} s_a^2 \tag{43}$$

and use

$$\widehat{\|b\|_M^2} = \frac{1}{N} \sum_a C_a \widehat{b}_a^2 - \widehat{\epsilon}. \tag{44}$$

The same estimation of b_a biases the pooled centered-moment estimates downward. We therefore add $\widehat{\epsilon}$ to the plug-in estimates of both σ^2 and $r\sigma^2$, after centering each coordinate at $\widehat{b}_a$. We denote the resulting corrected estimates by $\widehat{\sigma^2}$ and $\widehat{r\sigma^2}$. When $\widehat{\sigma^2} > 0$, the reported correlation parameter is

$$\widehat{r} = \frac{\widehat{r\sigma^2}}{\widehat{\sigma^2}}. \tag{45}$$

These corrections preserve the empirical decomposition exactly:

$$\frac{1}{N} \sum_a \sum_{s \in \mathcal{I}_a} d_{s,a}^2 = \widehat{\|b\|_M^2} + \frac{\widehat{\sigma^2} - \widehat{r\sigma^2}}{K} + \widehat{r\sigma^2}. \tag{46}$$

Indeed, subtracting $\widehat{\epsilon}$ from the squared-bias estimate and adding it to the cross-step component cancel in this expression, while adding the same correction to $\widehat{\sigma^2}$ and $\widehat{r\sigma^2}$ leaves their difference unchanged. Consequently, the estimated K-reducible component and expected dispersion are also unchanged.

For coordinates with $C_a = 1$, the variance cannot be estimated and no correction is applied. Depending on the dataset, coordinates eligible for correction account for approximately 93%–99% of all valid occurrences.

B Reproducibility Details

This appendix specifies the datasets and evaluation conventions, the Hamiltonian representation used by the equivariant field, and the training and inference settings of the reported checkpoints.

B.1 Datasets and Evaluation Protocol

Datasets and splits. We evaluate HIVE on four periodic Hamiltonian benchmarks. Table 3 summarizes their labels, electronic-structure codes, and data splits. The monolayer datasets are those of DeepH-E3 (Gong et al., 2023). Each graphene frame contains 72 carbon atoms and 936 orbitals, using the OpenMX (Ozaki, 2003) $s^2p^2d^1$ carbon basis. Each MoS_2 frame contains 25 Mo atoms, 50 S atoms, and 1125 orbitals, and the basis is $s^3p^2d^2$ for Mo and $s^2p^2d^1$ for S. The PBE bilayer dataset is the non-twisted bilayer graphene dataset released with DeepH-E3. The HSE06 bilayer dataset is that of DeepH-hybrid (Tang et al., 2024), computed with ABACUS (Li et al., 2016).

For every benchmark, we use the released structures, Hamiltonian labels, and Hamiltonian pair keys. The corresponding electronic-structure settings are given by the source publications: Gong et al. (2023) for the OpenMX/PBE benchmarks and Tang et al. (2024) for the ABACUS/HSE06 benchmark.

Table 3: Datasets used in the experiments, with splits reported as train/validation/test.

Dataset	Labels/code	Frames	Split
Monolayer graphene	PBE/OpenMX	450	270/90/90
Monolayer MoS_2	PBE/OpenMX	500	300/100/100
Bilayer graphene (PBE)	PBE/OpenMX	300	180/60/60
Bilayer graphene (HSE)	HSE06/ABACUS	1000	600/200/200

Baseline provenance. The values in Table 1 are quoted from the literature rather than recomputed. The monolayer DeepH, DeepH-E3, and DeepH-2 values are taken from Table 1 of Wang et al. (2024a), and the Hot-Ham values from Table 1 of Liang et al. (2026). The DeePTB entries are for the 4.5M-parameter SLEM model in Table 1 of Zhouyin et al. (2025); the HamGNN entries are the 2.8M-parameter third-party retrainings reported in the same table. For bilayer graphene with PBE labels, the Hot-Ham value is taken from its abstract, while the DeepH and DeepH-E3 entries marked ‡ are third-party retrainings reported by Wang et al. (2024a). For bilayer graphene with HSE06 labels, the DeepH-hybrid value is the test-set MAE reported by Tang et al. (2024). Thus, ‡ consistently denotes a result obtained by a third party rather than by the original authors.

Canonical blocks and orbital mask. Graph edges correspond to the $(i, j, \boldsymbol{R})$ keys present in the released Hamiltonians. The prediction target contains every on-site block and one canonical orientation of each off-site pair. The retained orientation is selected deterministically from the atom indices and integer cell offset, and is therefore unchanged by a rotation of the structure. For heterogeneous species, all blocks use a common padded layout. The binary mask M selects the physically valid orbital entries. Padding is applied to complete orbital shells, so the mask contains complete angular-momentum multiplets and commutes with the rotation action used below.

Evaluation metric. Dataset-level Hamiltonian MAE is computed by pooling all valid canonical entries,

$$\mathrm{MAE} = \frac{\sum_s \sum_a (M_s)_a \left| \widehat{H}_{s,a} - H^{\star}_{s,a} \right|}{\sum_s \|M_s\|_1}. \tag{47}$$

On-site blocks are included, and each Hermitian off-site pair is counted once. When a per-structure error is required, as in the reliability experiments, the same expression is evaluated for a single structure. The decomposition in Table 2 is evaluated at $K = 8$ using the pooled, finite-sample-corrected estimators of Appendix A.2. The common-bias, K-reducible, and cross-step components are divided by the empirical terminal MSE; the correlation parameter r is reported without this normalization.

Comparison with published metrics. The DeepH-E3 implementation uses the same element-wise masked error but differs in two aggregation conventions. First, it averages per-structure MAEs, whereas Eq. (47) pools all valid entries. On our test sets, the resulting relative differences are 0.000% for graphene, 0.020% for MoS_2, and 0.023% for bilayer graphene with PBE labels. Second, DeepH-E3 retains both directed blocks of an off-site pair, whereas we retain one. Because the two errors are equal under Eq. (48), this leaves each off-site error unchanged but gives off-site entries slightly greater weight relative to on-site entries.

DeepH-E3 reports separate atom-pair-type averages for MoS_2; these are not identical to either pooled convention, so we quote them as published. DeepH-hybrid uses the DeepH-E3 evaluation implementation, and the same qualification applies to its HSE06 result.

B.2 Hamiltonian representation and symmetry constraints

Hermiticity. We use a real orbital basis without spin-orbit coupling. Hermiticity therefore reduces to

$$H_{ji}(-\mathbf{R}) = H_{ij}(\mathbf{R})^{\top}. \tag{48}$$

For an off-site pair, the model predicts only its canonical orientation; the reverse block is obtained by transposition, so Eq. (48) holds exactly. For on-site blocks, their symmetry is not imposed explicitly and is instead learned from the symmetric training targets.

Rotation action. Let $D^{(i)}(g)$ denote the direct sum of the orbital rotation matrices associated with atom i. The action of a rotation g on a Hamiltonian-like block field X is $(\mathcal{T}_g X)_{ij}(\boldsymbol{R}) := D^{(i)}(g)\, X_{ij}(\boldsymbol{R})\, D^{(j)}(g)^{\top}$. Writing $g\mathcal{S}$ for the rotated structure, the target satisfies

$$H^{g\mathcal{S}}_{ij}(g\mathbf{R}) = D^{(i)}(g)\, H^{\mathcal{S}}_{ij}(\mathbf{R})\, D^{(j)}(g)^{\top}. \tag{49}$$

The canonical half-edge rule depends only on atom indices and integer cell offsets, so it is invariant under g. Because the orbital mask contains complete angular-momentum multiplets,

$$\mathcal{T}_g(X \odot M) = (\mathcal{T}_g X) \odot M, \qquad \|\mathcal{T}_g X\|_M = \|X\|_M. \tag{50}$$

These properties are the ones used in the rotation-invariance argument of Appendix A.1.

Component-wise equivariant encoding. The interaction network operates on irreducible $SO(3)$ features rather than directly on LCAO matrix elements. For orbitals with angular degrees l_μ and l_ν, the decoder reconstructs a Hamiltonian block using the Clebsch–Gordan decomposition,

$$H_{\mu\nu}^{(l_\mu l_\nu)} = \sum_{L=|l_\mu - l_\nu|}^{l_\mu + l_\nu} \sum_{M} C^{\,l_\mu l_\nu L}_{m_\mu m_\nu M} f_M^{(L)} \equiv \mathcal{D}(f), \tag{51}$$

where $\mu = (n_\mu, l_\mu, m_\mu)$ and $\nu = (n_\nu, l_\nu, m_\nu)$ are composite orbital indices. Here C denotes the normalized Wigner-3j coefficients implemented in e3nn (Geiger and Smidt, 2022), rather than unit-normalized Clebsch–Gordan coefficients. With this convention,

$$N_L = \sum_{m_\mu m_\nu} \left(C^{\,l_\mu l_\nu L}_{m_\mu m_\nu M}\right)^2 = \frac{1}{2L+1}. \tag{52}$$

The encoder is the corresponding normalized projection,

$$f_M^{(L)} = \frac{1}{N_L} \sum_{m_\mu, m_\nu} C^{\,l_\mu l_\nu L}_{m_\mu m_\nu M} H_{\mu\nu}^{(l_\mu l_\nu)}, \qquad \mathcal{E} \circ \mathcal{D} = \mathrm{id}. \tag{53}$$

For degree l, the encoder produces n_l orbital-pair slots. A learned degree-preserving lift maps these slots to the network channels:

$$[\mathcal{U} f]_{c,m}^{(l)} = \frac{1}{\sqrt{n_l}} \sum_{k=1}^{n_l} W_{ck}^{(l)} f_{k,m}^{(l)}. \tag{54}$$

The factor $n_l^{-1/2}$ keeps the feature scale approximately independent of the number of slots. Separate lifts $\mathcal{U}_\mathrm{n}$ and $\mathcal{U}_\mathrm{e}$ are used for on-site node blocks and off-site edge blocks, respectively.

B.3 Architecture, optimization, and inference

Shared equivariant field. Each evaluation of the vector field uses a fixed-depth equivariant interaction network with edge updates enabled. The network combines the $SO(2)$ convolution of eSCN (Passaro and Zitnick, 2023) with the equivariant attention mechanism of EquiformerV2 (Liao et al., 2024). Its complete parameter set is shared across every rollout step.

The path coordinate is represented by a sinusoidal embedding followed by a two-layer MLP (Jiao et al., 2023),

$$\gamma(t)_{2j} = \sin(\omega_j t), \qquad \gamma(t)_{2j+1} = \cos(\omega_j t), \qquad \tau(t) = \mathrm{MLP}(\gamma(t)). \tag{55}$$

Because t is a scalar, $\tau(t)$ is added only to the $l = 0$ channels. The MLP output layer is initialized to zero. Table 4 gives the architecture and optimization settings of each reported checkpoint.

Target normalization. The scale σ_H in Sec. 2.1 is computed once, before optimization, as the standard deviation of the valid canonical Hamiltonian entries in the first shuffled training batch. Because the batch size is one, this estimate uses one training structure. It is then frozen, stored in the checkpoint, and reused during both training and inference. The exact value for each checkpoint is reported in Table 4; Hamiltonians are stored in Rydberg throughout.

Optimization and checkpoint selection. Training minimizes Eq. (4) with AdamW (Loshchilov and Hutter, 2019), batch size one, initial learning rate 10^{-4}, zero weight decay, and gradient-norm clipping at 0.5. Each training example draws one independent $t \sim \mathcal{U}(0, 1)$ and requires one field evaluation. Validation uses explicit Euler integration with $K = 4$, and the checkpoint with the lowest validation MAE at this step count is retained. The model parameters are then held fixed for every reported inference-time comparison.

Table 4: Architecture and optimization hyperparameters of the final checkpoints used in Table 1.

	Monolayer Graphene	Monolayer MoS_2	Bilayer Graphene (PBE)	Bilayer Graphene (HSE)
Spherical order $l_{\max}$	5	4	5	4
Feature channels	128	192	128	128
Interaction blocks	2	2	2	2
Radial basis functions	128	128	128	128
Radial MLP hidden widths	[64, 64]	[64, 64]	[64, 64]	[64, 64]
Node attention heads	4	4	4	4
Controller layers	2	2	2	2
Edge envelope $r_{\max}$ (Å)	7.0	7.2	7.0	7.0
Node cutoff (Å)	5.0	5.0	5.0	5.0
Hermiticity half-edge constraint	yes	yes	yes	yes
Parameters	105.9M	143.6M	105.9M	64.5M
Optimizer	AdamW	AdamW	AdamW	AdamW
Initial learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}
Weight decay	0	0	0	0
LR schedule	Plateau	Plateau	Plateau	Plateau
Gradient-norm clip	0.5	0.5	0.5	0.5
Batch size / loss	1 / MSE	1 / MSE	1 / MSE	1 / MSE
Max training epochs	3000	3000	3000	3000
σ_H (Ry)	0.08910	0.10013	0.06324	0.06028

Inference protocol and step-count sweep. Inference uses float32 arithmetic, batch size one, and the uniform explicit Euler update in Eq. (5). Table 1 and all error-ranking experiments use $K = 8$ unless stated otherwise. The step-count sweep varies only K; the checkpoint, model parameters, and test structures remain fixed. Table 5 reports all test frames: 90 for monolayer graphene, 100 for monolayer MoS_2, 60 for bilayer graphene with PBE labels, and 200 for bilayer graphene with HSE06 labels.

Table 5: Test-set Hamiltonian MAE (meV) against the number of Euler steps K, at fixed weights. The best value in each column is bold.

K	Monolayer Graphene	Monolayer MoS_2	Bilayer Graphene (PBE)	Bilayer Graphene (HSE)
1	0.0815	0.1119	0.1325	0.1409
2	0.0708	0.0763	0.1209	0.1204
4	0.0675	0.0653	0.1179	0.1129
8	0.0665	0.0626	**0.1172**	0.1100
16	0.0662	0.0620	**0.1172**	0.1086
32	0.0661	**0.0618**	0.1174	0.1079
64	**0.0660**	**0.0618**	0.1175	0.1075
128	**0.0660**	**0.0618**	0.1176	**0.1074**

C Additional reliability analysis

C.1 DFT-labelled structural perturbations

Protocol. We evaluate D_M on controlled structural shifts for which new DFT labels are available. For each PBE benchmark, we select the first eighteen validation structures from the split generated with seed 42. For every parent structure, we draw one isotropic Gaussian displacement pattern and scale it to $\varsigma \in \{0, 0.03, 0.1, 0.3, 1.0\}$ Å, producing 90 structures per system. Using the same pattern at every level isolates the effect of displacement magnitude.

We recompute their Hamiltonians with OpenMX 4.0 (Ozaki, 2003), using the settings of the released datasets (Li et al., 2022). When a displaced atom leaves the unit cell, its unwrapped coordinate is retained while rebuilding the edge list, so the lattice-translation indices remain consistent with the OpenMX output. Predictions use the checkpoints of Table 1 with $K = 8$. None of these displaced structures or their new labels is used for training.

In-distribution reference. On the original test splits, the Spearman correlation between D_M and per-structure MAE is 0.902 for graphene ($n = 90$), 0.705 for MoS_2 ($n = 100$), and 0.442 for bilayer graphene with PBE labels ($n = 60$). We also measure selective-prediction performance using the risk–coverage curve (Geifman and El-Yaniv, 2017). After sorting structures from smallest to largest score, let $R(c)$ be the element-weighted masked MAE of the most confident fraction c. We define

$$\rho_{\mathrm{rec}} := \frac{A_{\mathrm{rand}} - A_{D_M}}{A_{\mathrm{rand}} - A_{\mathrm{oracle}}}, \qquad A = \frac{1}{n}\sum_{k=1}^{n} R(k/n), \tag{56}$$

where A_{oracle} is obtained by sorting on the true MAE and A_{rand} is averaged over 1000 random orderings. Ordering by D_M recovers 93.6%, 77.5%, and 56.9% of the oracle reduction for graphene, MoS_2, and bilayer graphene, respectively.

Ranking under structural shift. Across the five displacement levels, the true error spans approximately three orders of magnitude. The pooled Spearman correlation between D_M and MAE is 0.981–0.988 across the three systems. Thus, the score preserves the broad ordering of prediction difficulty under this controlled shift. However, note that the scale of the score is not calibrated, and Table 6 shows that the ratio of mean MAE to mean D_M varies from 2.4 to 9.4 across systems and displacement levels. We therefore interpret D_M as an error-ranking score, not as a numerical estimate of the prediction error or a confidence interval.

DFT relabelling control. The $\varsigma = 0$ structures are geometrically identical to their released counterparts and provide a direct check of the relabelling pipeline. Without involving the model, the newly computed and released Hamiltonians differ by 0.102 meV for graphene, 0.182 meV for MoS_2, and 0.055 meV for bilayer graphene, pooled over the eighteen structures in each system. These label-to-label differences set the error floor of the $\varsigma = 0$ experiment. They explain why the $\varsigma = 0$ prediction errors (0.147, 0.204, and 0.131 meV) exceed those on the released test splits (0.067, 0.063, and 0.117 meV); at the shifted displacement levels they are negligible.

Table 6: True Hamiltonian errors and stepwise-consistency score D_M on displaced structures with newly computed DFT labels ($K = 8$; mean $\pm$ standard deviation over eighteen structures per level; meV).

System	ς (Å)	MAE	RMSE	D_M	MAE/D_M
Graphene	0	0.147 ± 0.016	0.42 ± 0.03	0.034 ± 0.008	4.4
	0.03	4.53 ± 1.42	13.3 ± 3.5	0.610 ± 0.129	7.4
	0.1	24.4 ± 3.3	82.3 ± 10.9	3.76 ± 0.75	6.5
	0.3	145.4 ± 11.1	426 ± 33	27.2 ± 3.7	5.4
	1.0	248.4 ± 14.8	698 ± 60	61.6 ± 4.3	4.0
MoS_2	0	0.204 ± 0.004	0.38 ± 0.01	0.065 ± 0.007	3.2
	0.03	0.784 ± 0.152	2.03 ± 0.92	0.159 ± 0.023	4.9
	0.1	14.7 ± 3.1	57.4 ± 11.9	1.75 ± 0.36	8.4
	0.3	132.3 ± 12.1	416 ± 45	14.0 ± 1.5	9.4
	1.0	330.7 ± 16.7	1069 ± 75	40.0 ± 3.9	8.3
Bilayer graphene (PBE)	0	0.131 ± 0.017	0.38 ± 0.09	0.054 ± 0.004	2.4
	0.03	0.377 ± 0.115	1.70 ± 0.74	0.125 ± 0.030	3.0
	0.1	9.86 ± 1.45	47.4 ± 10.6	2.27 ± 0.43	4.3
	0.3	63.0 ± 4.7	259 ± 26	16.3 ± 1.3	3.9
	1.0	116.2 ± 8.5	464 ± 52	31.6 ± 2.2	3.7

Geometric-displacement control. Because displacement level and prediction error both increase with ς, a high correlation pooled across all 90 structures does not by itself show that a score captures model-specific error. Indeed, any quantity monotone in ς could produce a similar result. We therefore compare D_M with the realized displacement RMSD, computed from each perturbed structure and its undisplaced parent without alignment. As expected, displacement RMSD has a pooled Spearman correlation of at least 0.97 with MAE in every system. Within a fixed displacement level, however, it provides little ordering information: for an isotropic Gaussian displacement over $3N$ coordinates, the RMSD fluctuates by only approximately $1/\sqrt{6N} \simeq 4\%$ around its mean, whereas

the prediction error varies by 5%–31%. Across the twelve nonzero system–level combinations, D_M has a higher within-level correlation with MAE than displacement RMSD in nine cases, with the clearest separation at $\varsigma = 0.03$ Å. The within-level correlation between RMSD and D_M is below 0.8 in every cell and is typically between 0.1 and 0.7. Therefore, we can conclude that D_M is not simply a proxy for displacement magnitude. These observations are summarized in Fig. 5, which shows the two scatters.

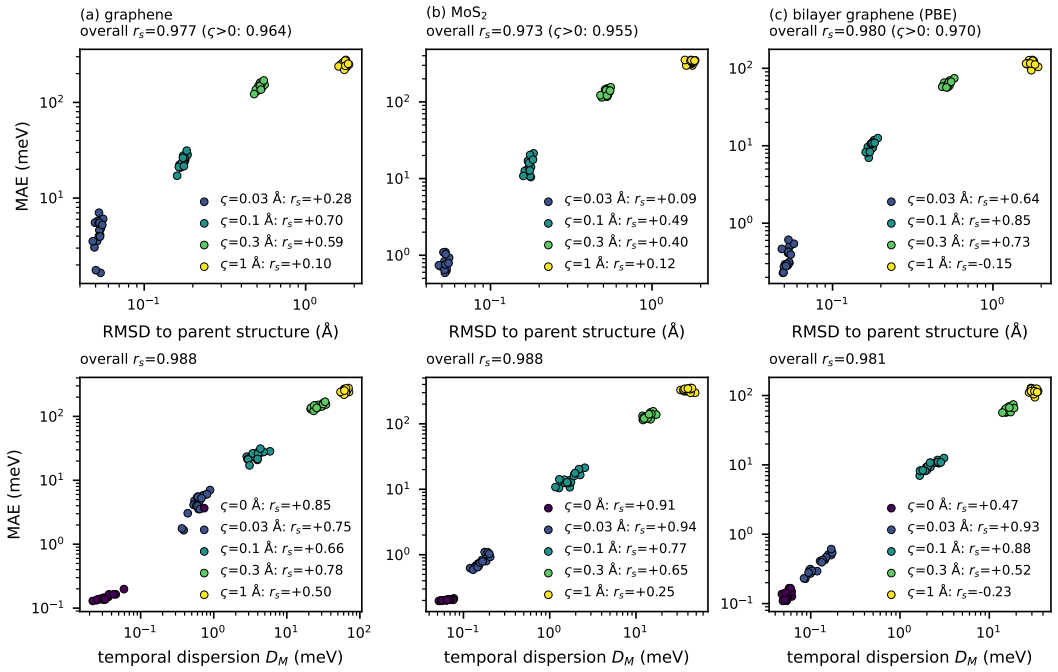

Figure 5: Per-structure MAE against the realised displacement RMSD (top row) and against D_M (bottom row) on the 90 relabelled structures of each system, coloured by displacement level; legends give the within-level Spearman correlation. Both quantities separate the levels; within a level, D_M orders the structures better than RMSD in most cells.

C.2 What the score can and cannot measure

Detectable error component. The dispersion is computed from outputs already produced by the rollout and therefore requires no additional field evaluations. Its limitation follows directly from Proposition 2, $\mathbb{E}D_M^2 = \frac{K-1}{K}(1-r)\sigma^2$. Thus, D_M^2 measures the non-common component removed when the individual step estimates are averaged. At $K = 8$, this quantity corresponds to 3.3% of terminal MSE for monolayer graphene, 43.8% for MoS_2, 2.4% for bilayer graphene with PBE labels, and 7.7% for bilayer graphene with HSE06 labels. These percentages should not be interpreted as error that remains removable by extending the rollout, as averaging has already removed a fraction $(K-1)/K$ of the non-common component. The component retained in the terminal error is $\mathbb{E}[D_M^2]/(K-1)$, which gives the K-reducible row of Table 2: 0.5%, 6.3%, 0.3%, and 1.1% of terminal MSE, respectively. The remaining error is common bias or cross-step-correlated error. Because these contributions affect all step estimates similarly, they are invisible to any score based only on their disagreement. A large D_M therefore signals unreliability, but a small D_M does not guarantee accuracy. Figure 4(b) additionally shows that the observable component changes smoothly over the tested range of K and does not collapse as the rollout becomes longer.

Step-dependent mean errors. Proposition 2 centres all steps on the step-averaged bias b. Consequently, a systematic difference between the mean errors of individual steps is included in σ^2 and r. To quantify this contribution, let $\delta_k = q_k - Q$ be the error of step k and

$$b_k = \mathbb{E}[\delta_k], \qquad b = \frac{1}{K}\sum_k b_k.$$

Table 7: Within-level Spearman correlations on the displaced structures of Fig. 5 (18 structures per cell, $K = 8$). RMSD is the realised displacement of each structure from its undisplaced parent; it is identically zero at $\varsigma = 0$. The last column gives the relative spread (std/mean) of RMSD within the level.

System	ς (Å)	$r_s(\mathrm{RMSD}, \mathrm{MAE})$	$r_s(D_M, \mathrm{MAE})$	$r_s(\mathrm{RMSD}, D_M)$	RMSD std/mean
Graphene	pooled	0.977	0.988	0.976	—
	0	—	0.85	—	—
	0.03	0.28	0.75	0.14	0.037
	0.1	0.70	0.66	0.39	0.037
	0.3	0.59	0.78	0.72	0.037
	1.0	0.10	0.50	0.28	0.037
MoS_2	pooled	0.973	0.988	0.974	—
	0	—	0.91	—	—
	0.03	0.09	0.94	0.23	0.037
	0.1	0.49	0.77	0.43	0.037
	0.3	0.40	0.65	0.53	0.037
	1.0	0.12	0.25	0.12	0.037
Bilayer graphene (PBE)	pooled	0.980	0.981	0.975	—
	0	—	0.47	—	—
	0.03	0.64	0.93	0.53	0.043
	0.1	0.85	0.88	0.76	0.043
	0.3	0.73	0.52	0.44	0.043
	1.0	−0.15	−0.23	−0.30	0.043

Then

$$\mathbb{E}D_M^2 = \frac{1}{K}\sum_k \|b_k - b\|_M^2 + \mathbb{E}\Big[\frac{1}{K}\sum_k \|\delta_k - b_k\|_M^2 - \Big\|\frac{1}{K}\sum_k (\delta_k - b_k)\Big\|_M^2\Big].$$

The first term is the dispersion caused by differences between the stepwise mean errors. Its largest contribution comes from the first evaluation at the uninformative initial iterate $\widetilde{H}^{(0)} = 0$, consistent with the analysis in Appendix D.1. Table 8 shows that this term accounts for approximately 4%–5% of $\mathbb{E}[D_M^2]$ for monolayer and bilayer graphene and 11%–14% for MoS_2, with little dependence on K. Most of the score therefore reflects structure-dependent disagreement remaining after the stepwise mean offsets are removed, rather than a fixed offset between rollout positions.

Table 8: Per-step bias spread $K^{-1}\sum_k \|b_k - b\|_M^2$ as a percentage of $\mathbb{E}D_M^2$, on the test split of each released PBE checkpoint. A value of zero would mean that every step of the rollout has the same mean error.

System	$K = 2$	4	8	16	32	64	128
Graphene	4.8	4.6	4.6	4.5	4.5	4.5	4.5
MoS_2	13.7	13.2	12.2	11.4	11.0	10.8	10.7
Bilayer graphene (PBE)	5.1	4.8	4.6	4.5	4.6	4.6	4.6

C.3 Comparison with an independently trained ensemble

We compare D_M with the disagreement of three independently trained models for each PBE system. The models use seeds 0, 1, and 2, with seed 0 corresponding to the released checkpoint. All members are evaluated with $K = 8$ on the same 90 relabelled structures used above.

Let $\widehat{H}_m$ be the terminal prediction of member m. We define

$$D_E^2 := \frac{1}{3}\sum_{m=1}^{3} \big\|\widehat{H}_m - \overline{\widehat{H}}\big\|_M^2, \qquad \overline{\widehat{H}} := \frac{1}{3}\sum_m \widehat{H}_m, \tag{57}$$

The two scores rank errors similarly. The correlations $r_s(D_E, \mathrm{MAE})$ are 0.987, 0.984, and 0.989 for monolayer graphene, MoS_2, and bilayer graphene, compared with 0.988, 0.988, and 0.981 for D_M.

On bilayer graphene, D_E is numerically closer to the MAE scale: the ratio MAE/D_E ranges from 0.7 to 1.8, compared with 2.4–4.3 for MAE/D_M. Neither score is calibrated, however. Across displacement levels, the former ratio changes by a factor of 2.6, compared with 1.8 for the latter.

The ensemble-mean prediction has 3.3%–4.2% lower MAE than the mean MAE of its individual members. This gain comes at substantially greater cost: the ensemble requires three trained models and 24 field evaluations per structure, compared with one model and eight evaluations for D_M. Because the ensemble members received unequal training budgets, we use this experiment only to compare their error-ranking signals, not their attainable predictive accuracy.

C.4 Active learning pilot

Candidate and held-out sets. We next test whether D_M can guide the selection of structures for labelling. Candidate structures are generated with $\varsigma = 0.03$ Å from validation parents not used in Appendix C.1. Each of 50 parents is displaced twice, producing 100 newly labelled candidates per system.

For bilayer graphene, only 42 unused validation parents remain after the structural-shift experiment, so the candidate parents include eight test structures. The held-out set uses 50 different test parents and is disjoint from these eight. For every system, each held-out parent is displaced once with the same amplitude. All candidate and held-out structures are relabelled with OpenMX. No training structure is displaced, and the held-out labels are never used for selection, early stopping, or hyperparameter tuning. The whole-pool acquisition experiment combines the 100 new candidates with the original training structures. We also evaluate acquisition restricted to the new candidates alone.

Acquisition and fine-tuning. Starting from the released checkpoint, we perform five acquisition rounds. At each round, we select 20 structures, reload the released checkpoint, and fine-tune for 20 epochs using AdamW with a fixed learning rate of 2×10^{-6}. We compare four selection rules:

1. largest D_M;
2. largest true prediction error, used only as a label-informed reference;
3. random selection from the combined training and candidate pool; and
4. random selection from the new candidates only.

The two random rules use three seeds. After each round, D_M and true error are recomputed using the updated model before the next selection. Because whole-pool random selection can choose structures already present in the original training set, Fig. 6 reports the number of genuinely new structures added.

Results. Selection by D_M directs acquisition away from structures that the model has already seen. Its held-out learning curve closely follows that of the label-informed true-error rule, despite requiring no candidate labels for ranking. When acquisition is restricted to unseen structures, however, this pilot does not show a consistent advantage over random selection. At the end of the experiment all new-only methods approach the same training set, so the new-only random curve is stopped after round four and its final convergence is shown only as a dotted segment. These results support D_M as a mechanism for avoiding redundant labels in a mixed pool, but do not yet establish that it is superior to random acquisition within a uniformly novel pool.

C.5 Scope and limitations of the reliability analysis

The experiments above establish that D_M can rank large prediction errors under the tested structural shifts and can identify redundant candidates in a small mixed-pool acquisition problem. They do not establish that D_M is a calibrated uncertainty estimate. In particular, it cannot detect bias or correlated error shared by all rollout steps.

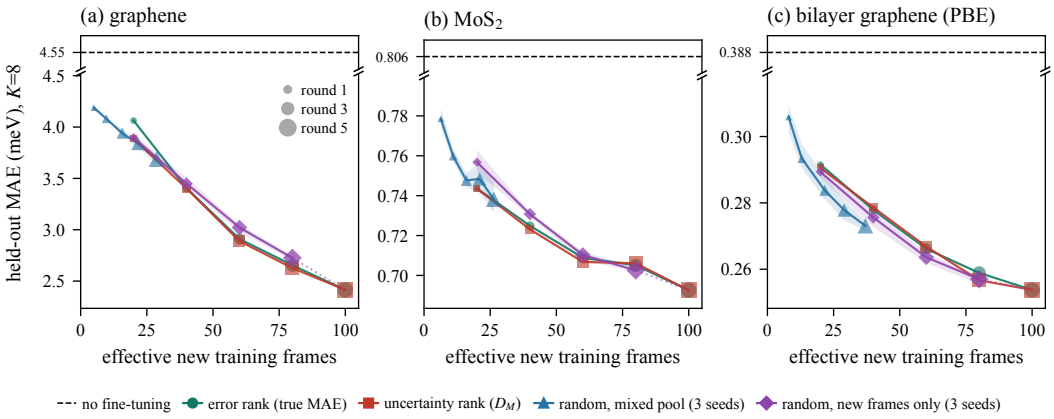

Figure 6: Held-out MAE ($K = 8$) against the number of new structures added. Larger markers are later rounds; random rules show three-seed means with the range shaded. The new-only random rule stops at round four, after which its training set would match the D_M rule (dotted segment). The y axis is broken.

Table 9: Held-out MAE (meV, $K = 8$) per round. Round 0 is the model without fine-tuning; parentheses give the new structures actually added by whole-pool random selection.

System	Round	New	Uncertainty (D_M)	Error rank	Random, pool	Random, new
Graphene	0	0	4.5500	—	—	—
	1	20	3.8854	4.0629	4.1889 (5)	3.8970
	2	40	3.4022	3.4053	4.0815 (10)	3.4499
	3	60	2.8891	2.9099	3.9403 (16)	3.0219
	4	80	2.6264	2.6543	3.8396 (22)	2.7252
	5	100	2.4141	2.4141	3.6865 (29)	—
MoS_2	0	0	0.8062	—	—	—
	1	20	0.7435	0.7436	0.7785 (6)	0.7568
	2	40	0.7231	0.7249	0.7598 (11)	0.7308
	3	60	0.7068	0.7089	0.7477 (16)	0.7102
	4	80	0.7061	0.7051	0.7484 (21)	0.7026
	5	100	0.6926	0.6926	0.7382 (26)	—
Bilayer graphene (PBE)	0	0	0.3875	—	—	—
	1	20	0.2905	0.2915	0.3059 (8)	0.2893
	2	40	0.2786	0.2779	0.2934 (13)	0.2756
	3	60	0.2668	0.2662	0.2838 (22)	0.2636
	4	80	0.2567	0.2589	0.2778 (29)	0.2570
	5	100	0.2538	0.2538	0.2731 (37)	—

The empirical scope is also limited. We do not compare with a learned heteroscedastic uncertainty model. The ensemble experiment contains only three members per PBE system, with unequal training budgets. The active-learning pilot uses one displacement amplitude, 100 candidates, five greedy acquisition rounds, and no diversity term. It is therefore a feasibility study rather than a full evaluation within a large-scale active-learning workflow. Broader conclusions will require larger candidate pools, matched computational budgets, purpose-built uncertainty baselines, and multiple types of distribution shift.

D ADDITIONAL PREDICTION RESULTS

D.1 WHAT CHANGES WHEN THE NUMBER OF INFERENCE STEPS INCREASES?

Each evaluation of the vector field contains the same fixed-depth equivariant interaction network Φ_θ. Increasing K does not change its parameters or the depth of an individual evaluation; it increases only the number of times the shared field is applied. At every rollout step, the structure-dependent node and edge features are initialized from the same cached embeddings $(\boldsymbol{n}^0, \boldsymbol{e}^0)$. Information from earlier evaluations is carried instead by the evolving node- and edge-resolved Hamiltonian iterate $\widetilde{H}^{(k)}$. Thus, later evaluations process the same structure together with a state generated by the preceding steps. Changing K affects both the Euler step size and the sequence of self-generated iterates at which the field is evaluated. These iterates need not coincide with the teacher states used during training. We therefore treat the fixed-checkpoint sweep in Table 5 as an empirical accuracy–computation trade-off rather than as a formal numerical-convergence result.

Accuracy of the individual step estimates. Table 10 resolves the $K = 4$ rollout into its four normalized-target estimates. The first estimate is evaluated at $t_0 = 0$ and $\widetilde{H}^{(0)} = 0$, before the state contains information from an earlier evaluation. It has the largest error on all three PBE benchmarks. The remaining estimates consume self-generated Hamiltonian states and have substantially smaller, mutually similar errors. The terminal prediction is the average of these four estimates. Its MSE is 3%–32% lower than the mean MSE of an individual estimate and lies within 3% of the best single estimate. It is 2.5% better than the best single estimate on MoS_2 and only marginally worse on the two graphene datasets. This behavior is consistent with Proposition 2: averaging removes the non-common part of the stepwise errors but leaves their shared component.

Table 10: Per-step errors of the $K = 4$ rollout. Each entry is the masked MSE (meV^2) against the DFT target when a single step estimate q_k is used as the prediction; "Terminal" is the MSE of the average of the four step estimates, i.e. of $\widetilde{H}^{(4)}$. Test split, all frames, $K = 4$, PBE checkpoints of the K-sweep in Table 5.

System	Step 0	Step 1	Step 2	Step 3	Terminal
Graphene	0.0414	0.0361	0.0361	0.0369	0.0362
MoS_2	0.0279	0.0126	0.0113	0.0128	0.0110
Bilayer graphene (PBE)	0.1323	0.1240	0.1259	0.1291	0.1245

D.2 DOES THE FIELD USE THE STRUCTURE OR RESCALE ITS ITERATE?

Potential shortcut. On the teacher path, $\widetilde{H}_t = tQ(\mathcal{S})$, and therefore

$$Q(\mathcal{S}) = \frac{\widetilde{H}_t}{t}, \qquad t > 0. \tag{58}$$

A field could consequently fit the teacher velocity by dividing its state input by t, without using the structure. A related shortcut was identified by Kim et al. (2024) for simulation-free neural-ODE training on paired data. In our setting, the source state is fixed at zero and the structure is provided as a separate input. The relevant failure mode is therefore not collapse of a structure encoder, but a field that ignores $\mathcal{S}$ whenever $t > 0$ and recovers the target only by rescaling $\widetilde{H}_t$.

An exact rescaler would also make the rollout independent of K. Let $q_0 = \boldsymbol{f}_\theta(0, 0; \mathcal{S})$ be the first output. After the first Euler update, $\widetilde{H}^{(1)} = \frac{q_0}{K}$. If every later evaluation returned $\widetilde{H}^{(k)}/t_k$, then it would return q_0 at every step, and the terminal prediction would equal the $K = 1$ prediction for every K. The substantial dependence on K in Table 5 therefore rules out this exact shortcut, although it does not exclude partial use of state rescaling.

Measurements made only on matched teacher pairs $\big(tQ(\mathcal{S}), \mathcal{S}\big)$ cannot distinguish the intended predictor from the shortcut: both return approximately $Q(\mathcal{S})$ there. We therefore combine an analytic constraint near $t = 0$ with an off-manifold structure-swap probe.

Constraint near the zero state. Assume that, for fixed $(t, \mathcal{S})$, the trained field is locally Lipschitz in its Hamiltonian input on the segment between 0 and $tQ(\mathcal{S})$. Let $L_{t,\mathcal{S}}$ be a corresponding Lipschitz constant, and define the teacher-state residual $\epsilon_{t,\mathcal{S}} := \|\boldsymbol{f}_\theta(tQ, t; \mathcal{S}) - Q(\mathcal{S})\|_M$, and the triangle inequality then gives

$$\left\|\boldsymbol{f}_\theta(0, t; \mathcal{S}) - Q(\mathcal{S})\right\|_M \;\le\; \epsilon_{t,\mathcal{S}} + L_{t,\mathcal{S}}\, t\, \|Q(\mathcal{S})\|_M. \tag{59}$$

The left-hand side evaluates the field with no Hamiltonian information in its state input. If the teacher-state residual is small and $L_{t,\mathcal{S}}$ remains bounded as $t \to 0$, Eq. (59) forces the zero-state output to approach the structure-specific target. Near the source, the field must therefore obtain the target information from $\mathcal{S}$. By contrast, the exact map $\widetilde{H}/t$ has state derivative proportional to $1/t$ and is singular at the source.

This argument is qualitative: we do not estimate a uniform Lipschitz constant, and the training objective controls an average residual rather than its supremum. It consequently rules out an exact bounded rescaler near $t = 0$ but does not quantify how strongly the field uses the state at larger t. The following probe addresses that question empirically.

Structure-swap probe. For two test structures $\mathcal{S}$ and $\mathcal{S}'$ from the same system, we hold the state fixed at the teacher state of $\mathcal{S}$ and change only the structure input. We compare $\boldsymbol{f}_\theta(t\,Q(\mathcal{S}), t; \mathcal{S})$ and $\boldsymbol{f}_\theta(t\,Q(\mathcal{S}), t; \mathcal{S}')$ through

$$\varrho(t) = \frac{\left\|\boldsymbol{f}_\theta(t\,Q(\mathcal{S}), t; \mathcal{S}') - \boldsymbol{f}_\theta(t\,Q(\mathcal{S}), t; \mathcal{S})\right\|_M}{\left\|Q(\mathcal{S}') - Q(\mathcal{S})\right\|_M}. \tag{60}$$

A state-only rescaler is unaffected by the swap and gives $\varrho = 0$. A field that ignores the state and predicts the corresponding target entirely from geometry gives $\varrho = 1$, up to prediction error. We find that $\|\boldsymbol{f}_\theta(t\,Q(\mathcal{S}), t; \mathcal{S}) - Q(\mathcal{S})\|_M$ is 0.0001 in the units of Table 11; this sets the measurement floor.

Results. Table 11 reports the probe for monolayer MoS_2, using the checkpoint of Table 1 and 40 random test-structure pairs. Replacing only the structure input changes the field output by 78%–95% of the frame-to-frame target difference. The swapped output is between 2.7 and 15 times closer to $Q(\mathcal{S}')$ than to $Q(\mathcal{S})$. Thus, the geometry remains the dominant input throughout the tested range of t.

The influence of the injected state nevertheless increases with t. The sum of the distances from the swapped output to the two targets exceeds $\|Q(\mathcal{S}') - Q(\mathcal{S})\|_M$ by only 1.1%–6.1%, indicating that the output remains close to the segment joining them. Projecting onto that segment as $(1 - w)Q(\mathcal{S}) + wQ(\mathcal{S}')$ places approximately 5% of the weight on the state-associated target at $t = 1/8$ and 23% at $t = 7/8$. This projection is only a geometric summary of the output; it should not be interpreted as an internal mixing mechanism.

Taken together, the K-dependence, the near-source constraint, and the swap probe reject the hypothesis that the trained field merely divides its state by t. The field continues to read the structure at every tested path coordinate while using the Hamiltonian iterate as supplementary information.

Table 11: Structure-swap probe on monolayer MoS_2, 40 random test pairs. ϱ is the relative displacement of the velocity output when only the structure input is replaced; 0 corresponds to a pure rescaler and 1 to a state-independent field. The last two columns give the distance from the swapped output to each target as an absolute masked norm; for reference the frame-to-frame target spread is $\|Q(\mathcal{S}') - Q(\mathcal{S})\|_M = 0.0964$, so the first of them approximately equals ϱ once divided by it. The measurement floor is 0.0001, i.e. about 0.1% of that spread.

t	ϱ	to $Q(\mathcal{S})$	to $Q(\mathcal{S}')$
1/8	0.949 ± 0.002	0.0915	0.0060
1/4	0.908 ± 0.004	0.0875	0.0110
1/2	0.844 ± 0.007	0.0813	0.0189
3/4	0.796 ± 0.009	0.0767	0.0250
7/8	0.777 ± 0.010	0.0748	0.0275

Probe scope. The probe uses a teacher state from $\mathcal{S}$, not a self-generated rollout state. When paired with $\mathcal{S}'$, this input is deliberately off the teacher manifold; the experiment therefore tests the field's dependence on its two inputs rather than the distribution of states encountered during inference. Frames from the same system can also have slightly different edge lists. We inject the state only on the intersection of the two edge sets, which covers 98.8% of Hamiltonian rows on average, and set the remaining near-cutoff rows to zero in both branches. The effect of this approximation is included in the 0.0001 measurement floor reported above.