MOLFIELD: REPRESENTING MOLECULES AS NEURAL FIELDS VIA HYPERNETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular representations fundamentally shape how machine learning systems reason about molecular structure and physical properties. Most existing approaches adopt a discrete pipeline: molecules are encoded as sequences, graphs, or point clouds, mapped to fixed-dimensional embeddings, and then used for task-specific prediction. This paradigm treats molecules as discrete objects, despite their intrinsically continuous and field-like physical nature. We argue that molecular learning can instead be formulated as learning in function space. Specifically, we model each molecule as a continuous function over three-dimensional (3D) space and treat this molecular field as the primary object of representation. From this perspective, conventional molecular representations arise as particular sampling schemes of an underlying continuous object. We instantiate this formulation with MolField, a hyper-network-based framework that generates molecular fields. To ensure consistency under rigid motions, these functions are defined over canonicalized coordinates, yielding invariance to global SE(3) transformations. To enable learning directly over functions, we introduce a structured weight tokenization and train a sequence-based hyper-network that generates the parameters of each molecular field. We evaluate MolField on dynamic molecular surface modeling, molecular property prediction, and molecular generation. Our results show that treating molecules as neural fields fundamentally changes how molecular representations generalize and yields downstream behavior that is stable to how molecules are discretized or queried.

1 INTRODUCTION

Representations determine how scientific objects are formalized, compared, and manipulated, and thereby constrain what can be learned about them and transferred across settings (Frigg & Nguyen, 2016; Franklin, 2017). A molecule is a physical entity embedded in three-dimensional (3D) space, whose identity is determined not only by discrete atom and bond types, but also by geometry, shape, and their evolution under physical processes (Crippen et al., 1988). Many fundamental molecular quantities—such as fields, surfaces, and energies—are therefore most naturally described as continuous functions over space (Vanommeslaeghe et al., 2014), a view long established in electronic-structure theory and classical molecular modeling (Hohenberg & Kohn, 1964; Kohn & Sham, 1965; Murray & Sen, 1996; Cramer et al., 1988). Concretely, molecular quantities such as the dipole moment (μ), polarizability (α), and spatial extent (R^2) are molecule-wide, depending on the spatially extended electron distribution rather than on local bonding motifs alone.

Despite this intrinsic continuity, prevailing molecular learning pipelines treat molecules as discrete structures: sequences, graphs, or point clouds are first constructed, mapped to fixed-dimensional embeddings, and then consumed by task-specific predictors (Figure 1). These discretizations act as particular sampling schemes of the underlying molecular object, but discard fine-grained continuity and impose representation-specific inductive biases, coupling the learned representation to individual tasks and discretization choices (Schütt et al., 2017; Gasteiger et al., 2020b).

To this end, we formulate molecular learning directly in function space. Specifically, we represent each molecule as a continuous function defined over 3D space (Sitzmann et al., 2020b; Dupont et al., 2022; Zama Ramirez et al., 2024), while treating the function as the object of representation rather

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

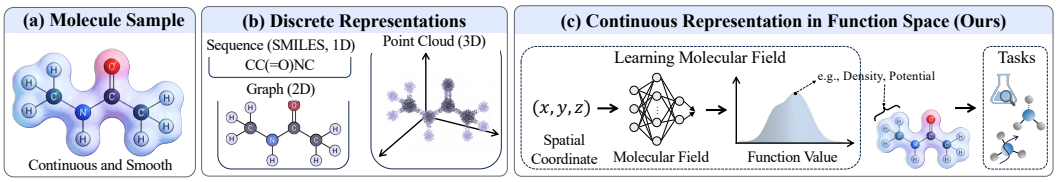

Figure 1: **Molecular representations: discrete versus continuous function-space modeling.** (a) A molecule exhibits inherently continuous and smooth spatial structure. (b) Conventional molecular representations discretize molecules into sequences, graphs, or point clouds, which only partially capture this continuity. (c) In contrast, we model molecules directly in function space by learning a continuous molecular field that maps 3D coordinates to geometric quantities (e.g., surface distance or atom-type density), which can be queried for different downstream tasks.

than as a fixed physical observable or a task-specific computational target. Downstream tasks are then addressed by querying or reading out such functions.

This formulation raises three challenges. *(1) Defining molecules as functions.* While molecular identity is invariant to global translations and rotations (Batzner et al., 2022), continuous functions are sensitive to coordinate choices, so physically identical configurations may correspond to different functions. Related work mitigates this by augmentation or a fixed reference rotation (Kirchmeyer et al., 2024; Dumitrescu et al., 2025), by equivariant field architectures (Wu et al., 2025), or by learned canonicalization (Kaba et al., 2023). But augmentation yields invariance only in expectation, equivariant architectures constrain the field network itself, and no canonical frame on point clouds is everywhere continuous (Dym et al., 2024); a parameter-level representation additionally requires that the generated parameters, not only the field values, be pose-independent. *(2) Learning molecular functions.* Standard molecular representations map discrete structures to fixed-dimensional embeddings, whereas neural fields lack a discrete interface for comparison, interpolation, or generalization. Prior work either fits the parameters per instance and encodes them afterwards (Zama Ramirez et al., 2024), conditions one shared network on a latent code (Park et al., 2019; Dupont et al., 2022), or amortizes the fitting with a hyper-network (Chen & Wang, 2022; Babu et al., 2025). But fitted parameters carry optimization artifacts and are comparable only up to permutations of hidden units, while a latent code fixes the interface before the field is learned and leaves one weight set shared by every molecule. *(3) Using molecular functions in downstream tasks.* If the function itself is the representation, different tasks must be solved by querying or reading out such functions, which requires a common interface through which they can be generated and used across tasks. Existing read-outs take a quantity adjacent to the field: a modulation code (Kirchmeyer et al., 2024), a pre-trained encoder (Lee et al., 2024), or an intermediate hyper-network block (Babu et al., 2025). But whether the generated parameters themselves suffice as the molecular representation, in particular for molecule-wide quantum properties, remains open.

To address these challenges, we propose MolField, a framework that operationalizes molecular learning directly in function space. MolField represents each molecule as a continuous molecular field using a *Canonical Implicit Neural Representation (C-INR)*, which defines molecular functions over canonicalized coordinates invariant to SE(3) transformations, so that the function depends only on intrinsic molecular geometry up to the degeneracy characterized in Section 4.1 (Challenge 1). To expose molecular functions to modern sequence models, we introduce *Structured Weight Tokenization (SWT)*, which converts C-INR parameters into semantically organized tokens while preserving the compositional structure of the underlying neural function (Challenge 2). Building on this interface, MolField employs a *Function Space Hyper-Network (FSHN)* (Ha et al., 2017; Chen & Wang, 2022) that directly generates C-INR parameters from a molecular condition in a single forward pass, so that the generated parameters are read out directly as the molecular representation (Challenge 3). We evaluate MolField on dynamic molecular surface modeling, property prediction, and generation tasks, showing the neural field modeling yields the robust representation for molecules.

2 Related Work

Molecular Representations: Discrete and Continuous. Molecular representation fundamentally shapes what structural and physical information can be learned and how models generalize. Most

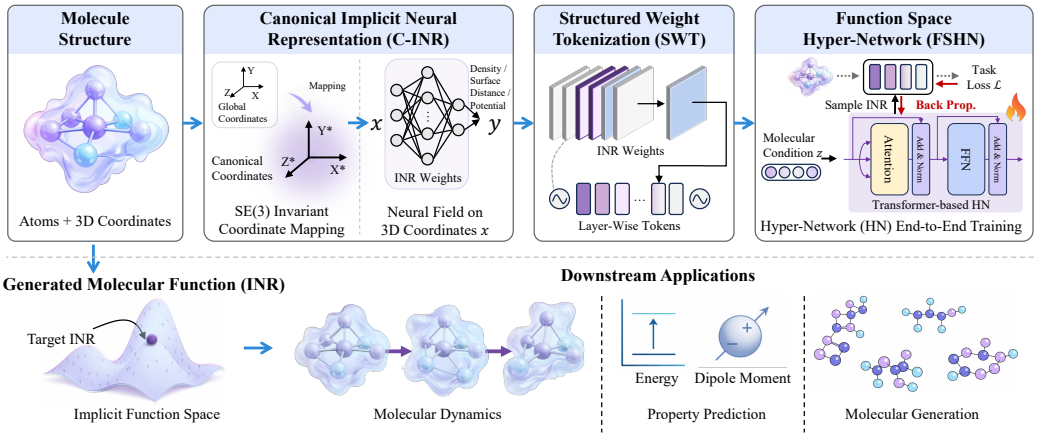

Figure 2: **Overview of MolField.** *Top*: A molecule is mapped into a canonical coordinate system to construct an implicit neural representation (INR) that is invariant to SE(3) transformations. The INR weights are tokenized in a structured, layer-wise manner to enable transformer-based processing, while the model is trained as a hyper-network that learns to generate task-conditioned INRs. *Bottom*: The resulting INR supports diverse molecular learning tasks, including dynamic surface modeling, molecular property prediction, and molecular generation.

existing approaches rely on *discrete surrogates*, including 1D sequence-based encodings such as SMILES or SELFIES (Weininger, 1988; Krenn et al., 2020), graph-based message-passing networks (Gilmer et al., 2017; Schütt et al., 2017; Gasteiger et al., 2020a; Schütt et al., 2021), and 3D coordinate-based equivariant networks (Satorras et al., 2021; Liao & Smidt, 2022; Batatia et al., 2022). Other works adopt continuous but still discretized representations, such as voxel grids (O Pinheiro et al., 2023), which remain resolution-dependent and tied to specific sampling schemes. In contrast, we model the molecule itself as a continuous function over 3D space, and view discrete representations as query schemes over an underlying molecular field.

Implicit Neural Representations for Molecules. Implicit neural representations model continuous signals as coordinate-based neural functions $f_\Theta : \mathbf{x} \mapsto \mathbf{y}$ (Mildenhall et al., 2021; Martin-Brualla et al., 2021; Essakine et al., 2024). Several recent works apply INRs to molecular surface reconstruction and dynamics modeling (Sun et al., 2023; Wu et al., 2025; Lee et al., 2024), complementing learned surface representations of proteins (Sverrisson et al., 2021; Wu & Li, 2024). Another line of work represents molecules as continuous density fields for molecular generation (Kirchmeyer et al., 2024). Closest to us, HyDiF generates per-molecule neural distance fields with a diffusion-trained hyper-network, without explicit handling of rotations (Babu et al., 2025), while MING and MCF define functions on the molecular graph rather than on 3D space (Nguyen et al., 2025; Wang et al., 2024b). In most of these approaches, the implicit function is tied to a single task. In contrast, we treat the implicit function itself as the molecular representation. Rather than conditioning a shared network on a latent code (Park et al., 2019), we produce the parameters of each molecular function with a hyper-network over structured weight tokens (Chen & Wang, 2022), and evaluate the function in a canonical frame. The same architecture is used for dynamic surface modeling, property prediction, and generation, with separately trained models.

Traditional Continuous Representations for Molecules. Continuous molecular representations are not unique to neural representation learning: quantum chemistry modeling have long described molecules through wavefunctions and electron densities, electrostatic potentials and CoMFA-style interaction fields, and smooth atom-density descriptors (Hohenberg & Kohn, 1964; Kohn & Sham, 1965; Murray & Sen, 1996; Cramer et al., 1988; Bartók et al., 2013; Willatt et al., 2019), and machine-learning models increasingly emulate such electronic-structure quantities (Hermann et al., 2020; del Rio et al., 2023). These functions are physical state variables, observables, or fixed descriptors, whereas MolField uses an INR-parameterized geometric field (surface distance or atom-type density) as a learnable representation object whose parameters are generated by a hyper-network. Our contribution is therefore not to introduce functions into molecular science, but to bring neural function-space learning to molecular representation.

3 PRELIMINARIES

Definition 3.1 (Molecular Functions)**.** We model molecules as continuous functions via INRs defined over 3D Euclidean space. Given a molecular configuration described by atomic coordinates $X = \{\mathbf{x}_i \in \mathbb{R}^3\}_{i=1}^N$ and atomic attributes $A = \{a_i\}_{i=1}^N$, a molecular function is defined as

$$f_M : \mathbb{R}^3 \to \mathbb{R}^C, \tag{1}$$

where $f_M(\mathbf{x})$ encodes molecular quantities at spatial location $\mathbf{x}$, such as the signed distance to the molecular surface or atom-type densities (Sun et al., 2023; Kirchmeyer et al., 2024); these are geometric fields, not electron densities. In practice, this function is implemented as an MLP with parameters $\mathbf{\Theta}$, which can be queried at any resolution.

Definition 3.2 (SE(3) Symmetry)**.** Molecular functions admit intrinsic symmetries of 3D space. Specifically, for any rigid-body transformation $g = (R, \mathbf{t}) \in \mathrm{SE}(3)$ with $R \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$, the function transforms as

$$f_M(\mathbf{x}) \mapsto f_M(R^{-1}(\mathbf{x} - \mathbf{t})). \tag{2}$$

Since such transformations preserve molecular identity, the representation must be invariant to the action of $\mathrm{SE}(3)$; we exclude reflections, which alter molecular chirality.

Problem Definition. Given a molecule specified by atomic coordinates X and atomic attributes A, our objective is to learn a continuous molecular function f_M parameterized by an implicit neural representation that is invariant under global rigid-body transformations in $\mathrm{SE}(3)$ and supports diverse downstream learning objectives.

4 METHODOLOGY

We propose MolField, a framework for learning and generating molecular representations directly in implicit functional space (Figure 2). At a high level, MolField decomposes the problem into three stages: constructing a symmetry-consistent implicit representation for individual molecules, exposing this representation through a structured tokenization, and learning a conditional hyper-network over such representations.

4.1 CANONICAL IMPLICIT NEURAL REPRESENTATIONS (C-INR)

We define a C-INR as a coordinate-based neural function that models a molecule as a continuous field in $\mathbb{R}^3$ while ensuring invariance to global rigid-body transformations. Given atomic coordinates $X = \{\mathbf{x}_i\}_{i=1}^N$, a C-INR parameterizes a molecular function $f_{\mathbf{\Theta}} : \mathbb{R}^3 \to \mathbb{R}^C$, whose outputs represent physically meaningful quantities such as signed distance or density. The key property of C-INR is that the function is evaluated in a molecule-specific canonical coordinate system, such that the parameters $\mathbf{\Theta}$ depend only on intrinsic molecular geometry.

Canonical Coordinate Mapping. We remove pose ambiguity by expressing spatial queries in a canonical reference frame. Specifically, let $\bar{\mathbf{x}} = \frac{1}{N}\sum_i \mathbf{x}_i$ denote the molecular centroid and let $Q(X) \in SO(3)$ be a rotation-equivariant canonical frame derived from molecular geometry. We define the coordinate transformation

$$c(\mathbf{x}; X) = Q(X)^\top (\mathbf{x} - \bar{\mathbf{x}}), \tag{3}$$

and evaluate the molecular field as

$$f^{\mathrm{C}}_{\mathbf{\Theta}}(\mathbf{x}; X) = f_{\mathbf{\Theta}}(c(\mathbf{x}; X)). \tag{4}$$

This formulation standardizes the function domain across different global poses without aligning distinct molecules into a shared global frame.

Canonical Frame Construction. The frame $Q(X)$ is constructed from rotation-equivariant vector features extracted from molecular geometry. Let $\Phi(X) = \{\mathbf{v}_i(X)\}_{i=1}^N$ be a mapping that satisfies $\Phi(RX) = \{R\mathbf{v}_i(X)\}_{i=1}^N$ for any $R \in SO(3)$ (see Appendix D for details). We form two global axes via invariant aggregation,

$$\mathbf{u}_k(X) = \sum_{i=1}^N \alpha_i^{(k)}(X)\mathbf{v}_i(X), \quad k \in \{1, 2\}, \tag{5}$$

where $\alpha_i^{(k)}(X)$ are rotation-invariant scalars. An orthonormal basis is then obtained through a modified Gram–Schmidt procedure,

$$Q(X) = [\mathbf{q}_1(X), \mathbf{q}_2(X), \mathbf{q}_3(X)] \in SO(3), \tag{6}$$

with $\mathbf{q}_1 = \mathbf{u}_1/\|\mathbf{u}_1\|$, $\mathbf{q}_2$ the normalized component of $\mathbf{u}_2$ orthogonal to $\mathbf{q}_1$, and $\mathbf{q}_3 = \mathbf{q}_1 \times \mathbf{q}_2$.

SE(3) Invariance. Invariance to global rigid-body transformations then follows by construction, from the rotation equivariance of the frame and the removal of translations by centering.

Proposition 4.1 (Frame equivariance)**.** *For any $R \in SO(3)$ and any X with $\mathbf{u}_1(X) \neq \mathbf{0}$ and $\mathbf{u}_1(X) \times \mathbf{u}_2(X) \neq \mathbf{0}$, the canonical frame satisfies $Q(RX) = R\,Q(X)$.*

Lemma 4.2 (SE(3)-invariant coordinate mapping)**.** *For any rigid-body transformation $g = (R, \mathbf{t}) \in SE(3)$,*

$$c(R\mathbf{x} + \mathbf{t}; RX + \mathbf{t}) = c(\mathbf{x}; X). \tag{7}$$

Lemma 4.2 follows directly from Proposition 4.1 and the definition of the centroid-based coordinate transform. Consequently, for any two rigid-body equivalent molecular configurations, all spatial queries are mapped to identical canonical coordinates. Since the molecular field is evaluated through the canonical coordinate map, the induced function is invariant under SE(3), and the representation parameters $\mathbf{\Theta}$ depend only on intrinsic molecular geometry. The full proofs are provided in Appendix B. The frame is not defined when $\mathbf{u}_1 = \mathbf{0}$ or $\mathbf{u}_1 \parallel \mathbf{u}_2$. Because both vectors are aggregated from O(3)-equivariant features, this necessarily occurs for symmetric molecules whose point group is neither C_1 nor C_s, a known limitation of equivariant canonicalization (Kaba et al., 2023; Dym et al., 2024); in QM9 these molecules account for about 0.8% of the dataset (Appendix B). Appendix G.6 measures the consequence: on the most degenerate 5% of the QM9 test set, ranked by $\|\mathbf{u}_1\|$, a rigid rotation changes the prediction by about 10^{-4} of the model's own error there.

4.2 Structured Weight Tokenization (SWT)

The parameters of a C-INR are high-dimensional tensors with rich internal structure, and treating them as flat vectors discards the architectural organization of the underlying neural function. To enable learning directly over molecular functions, we introduce a structured tokenizer that exposes C-INR weights in a form amenable to sequence models.

A C-INR is parameterized by a collection of layer-wise tensors $\mathbf{\Theta} = \{\theta^{(l)}\}_{l=1}^{L}$, where each layer implements a distinct functional transformation. SWT introduces an ordered sequence of parameter tokens that generate this structured parameter set,

$$\tau_{1:T} \longrightarrow \mathbf{\Theta}, \quad \tau_t \in \mathbb{R}^{d_{\text{model}}}, \tag{8}$$

where each token is a learnable vector that is projected to a localized parameter block associated with a specific layer and parameter type (e.g., weight or bias). The token ordering follows the architectural hierarchy of the C-INR, thereby preserving the causal structure of neural computation.

Since parameter tokens are non-exchangeable and carry explicit architectural semantics, we encode their structural roles using learnable positional embeddings. Each token is indexed by a tuple (l, r) indicating its layer l and parameter role r, and its representation is defined as

$$\tilde{\tau}_t = \tau_t + \mathbf{e}_{\text{layer}}(l) + \mathbf{e}_{\text{role}}(r), \tag{9}$$

where $\mathbf{e}_{\text{layer}}$ and $\mathbf{e}_{\text{role}}$ are trainable embeddings, and $\tilde{\tau}_{1:T}$ is the input to the hyper-network (Section 4.3). This encoding injects inductive bias about network depth and parameter functionality, allowing downstream models to reason over molecular functions while respecting the compositional structure of the implicit representation.

4.3 Function Space Hyper-Network (FSHN)

Under the C-INR formulation, each molecule is represented by a neural function $f_{\mathbf{\Theta}}$ whose parameters specify the corresponding molecular field, so molecular learning can be viewed as learning over such functions. Fitting INRs independently per molecule and then learning over the resulting parameters introduces optimization-dependent artifacts and prevents consistent reasoning in function space. We address this limitation by learning a conditional hyper-network over C-INRs in an end-to-end manner, building on transformer-based INR hyper-networks (Chen & Wang, 2022).

Specifically, we introduce a function-space hyper-network G_ϕ that maps a conditioning variable z to C-INR parameters

$$\mathbf{\Theta} = G_\phi(z). \tag{10}$$

This defines a conditional mapping from z to molecular functions via their parameters, enabling learning directly at the level of functions.

Building on SWT (Section 4.2), the C-INR parameters are represented as a structured token sequence $\tau_{1:T}$. Writing g_ϕ for the transformer body of G_ϕ, the hyper-network processes the embedded tokens $\tilde{\tau}_{1:T}$ with a causal mask conditioned on z and projects each output token to its parameter block,

$$\hat{\tau}_t = g_\phi(\tilde{\tau}_{\le t}, z), \qquad \theta^{(l,r)} = \mathrm{Proj}_{(l,r)}(\hat{\tau}_t), \tag{11}$$

so that each token is conditioned on the preceding ones while respecting the hierarchy of the underlying representation; all tokens are produced in a single forward pass.

During training, the hyper-network generates C-INR parameters $\mathbf{\Theta} = G_\phi(z)$, which instantiate a molecular function $f_{\mathbf{\Theta}}$. Task-specific predictions are obtained by querying this function through $\mathcal{T}(\cdot)$ and optimized using the task loss

$$\mathcal{L}_{\text{task}}(\phi) = \mathbb{E}_{(z,y)\sim\mathcal{D}}\big[\ell(\mathcal{T}(f(\cdot; G_\phi(z))), y)\big]. \tag{12}$$

Gradients propagate through the downstream task loss from INR to update ϕ, yielding a single end-to-end training procedure. At inference time, molecular functions are generated in a single forward pass conditioned on z and queried according to the downstream task, without additional optimization.

4.4 Training Procedure

Conditioning Variables. The conditioning variable z specifies which molecular function should be generated. (1) For *dynamic surface modeling*, z is formed by concatenating a learned embedding of the molecule with the target time step t. The generated C-INR therefore represents the molecular field of the same molecule at time t. (2) For *property prediction*, z is given by the molecular embedding of the static structure. The generated parameters serve as a function-space representation of the molecule for downstream regression. (3) For *molecular generation* (Appendix G.1), z is the molecular embedding during density-field training; at sampling time we consider the unconditional setting and draw $z \sim \mathcal{N}(0, I)$. The hyper-network thus generates molecular functions from the sampled condition alone, without additional semantic constraints.

Training Objectives. The C-INR architecture and canonical coordinate system are shared across tasks. Tasks differ in the conditioning variable, the semantics of the function outputs, the read-out, and the supervision signals used during training. *(1) Dynamic Surface Modeling.* The generated C-INR is instantiated as a signed distance function (SDF) (Sun et al., 2023) $f_{\mathbf{\Theta}} : \mathbb{R}^3 \to \mathbb{R}$, which maps spatial coordinates to distances from the molecular surface. Given sampled spatial locations $\mathbf{x}$, we supervise the field using $\mathcal{L}_{\text{MD}} = \mathcal{L}_{\text{SDF}} + \lambda\mathcal{L}_{\text{eikonal}}$, where $\mathcal{L}_{\text{SDF}}$ enforces accurate reconstruction of surface distances and $\mathcal{L}_{\text{eikonal}} = \mathbb{E}_{\mathbf{x}}(\|\nabla_{\mathbf{x}} f_{\mathbf{\Theta}}(\mathbf{x})\| - 1)^2$ regularizes gradient norms to encourage valid geometric fields. *(2) Property Prediction.* For property prediction, we do not explicitly instantiate the neural field. Instead, the structured token sequence corresponding to the generated C-INR parameters is directly aggregated into a fixed-dimensional embedding. This embedding is supervised by multiple physical property targets using a regression objective $\mathcal{L}_{\text{prop}} = \|\hat{\mathbf{y}} - \mathbf{y}_{gt}\|_1$, where $\hat{\mathbf{y}}$ denotes the predicted properties. The property model is initialized from the density-field model trained for molecular generation (see (3) below), which uses no property labels, and is then fine-tuned on all targets jointly. *(3) Molecular Generation* (Appendix G.1). For molecular generation, the generated C-INR is instantiated as a continuous density field $f_{\mathbf{\Theta}} : \mathbb{R}^3 \to \mathbb{R}^C$, whose values represent the probability of the existence of certain atoms at spatial locations. We sample spatial points $\mathbf{x}$ from the 3D domain and supervise the predicted density against the ground-truth distribution $\rho_{gt}(\mathbf{x})$ using $\mathcal{L}_{\text{gen}} = \mathbb{E}_{\mathbf{x}}\|f_{\mathbf{\Theta}}(\mathbf{x}) - \rho_{gt}(\mathbf{x})\|^2$.

5 Main Results

Dynamic Surface Modeling. Following Sun et al. (2023), we evaluate MolField on dynamic molecular surface modeling from protein molecular dynamics trajectories, assessing its ability to model time-varying molecular structures in a continuous and resolution-independent manner.

Table 1: **Evaluation of dynamic molecular surface modeling.** We report volumetric IoU (IoU ↑), chamfer distance (CD ↓), and normal consistency (NC ↑), where the best performance is **boldface**. MolField achieves the best average performance on all three metrics, and the best IoU and NC on five of the six systems.

	NDF			DSR			CanFields			MolField (Ours)		
Dataset	IoU↑	CD↓	NC↑	IoU↑	CD↓	NC↑	IoU↑	CD↓	NC↑	IoU↑	CD↓	NC↑
DIMS	0.9104	0.0006	0.9512	0.9316	**0.0003**	0.9671	**0.9391**	0.0005	**0.9719**	0.9353	0.0006	0.9645
YiiP	0.8120	0.0006	0.8515	0.8425	**0.0003**	0.8737	0.8404	0.0004	0.8354	**0.9270**	0.0006	**0.9565**
NhaA	0.7968	0.0008	0.8124	0.8265	0.0005	0.8372	0.8362	0.0004	0.8027	**0.8844**	**0.0003**	**0.8949**
I-FABP	0.8729	0.0008	0.9196	0.8907	0.0005	0.9382	0.9135	0.0006	0.9270	**0.9392**	**0.0002**	**0.9721**
FRODA	0.8087	0.0007	0.8893	0.8318	0.0004	0.9098	0.8283	**0.0003**	0.8895	**0.8687**	0.0006	**0.9414**
adk_equi	0.7215	0.0025	0.7620	0.7526	0.0020	0.7914	0.7491	0.0019	0.7977	**0.9131**	**0.0004**	**0.9522**
Average	0.8204	0.0010	0.8643	0.8460	0.0007	0.8862	0.8511	0.0007	0.8707	**0.9113**	**0.0005**	**0.9469**

Table 2: **Evaluation of molecular property prediction on QM9. A.R.** denotes the average ranking across all five properties.

Method	$\varepsilon_{\text{HOMO}}$ meV	μ mD	R^2 ma_0^2	α ma_0^3	ZPVE meV	A.R.
SchNet	41	33	73	235	1.70	17.8
Molformer	25	28	350	41	2.05	15.3
Cormorant	34	38	961	85	2.03	20.8
SEGNN	24	23	660	60	1.62	16.0
EGNN	29	29	106	71	1.55	15.7
PaiNN	28	12	66	45	1.28	10.8
NoisyNodes	20	25	700	52	1.16	12.7
GNS-TAT+NN	17	21	650	47	**1.08**	10.0
DimeNet++	25	30	331	44	1.21	13.3
SphereNet	23	26	292	46	1.12	11.8
MACE	22	15	210	38	1.23	9.5
GA-GNN	21	11	63	45	1.18	7.9
LieConv	30	32	800	84	2.28	20.2
Equiformer v1	15	11	251	46	1.26	9.1
Equiformer v2	**14**	10	186	50	1.47	9.2
TorchMD-NET	20.3	11	33	59	1.84	11.0
ViSNet	17.3	9.5	29.8	41.1	1.56	7.0
Geoformer	18.4	10	27.5	40	1.28	6.0
GotenNet-B	15.2	**7.2**	**27**	32	1.09	**2.2**
FuncMol	46	15	174	36	3.64	13.9
MolField (from scratch)	23	24	106	31	1.32	10.0
MolField (Ours)	**14**	9	62	**21**	1.16	2.8

Table 3: **Ablation study on key components of MolField.** Removing any component degrades performance.

	Dynamics		Property	
Variant	IoU↑	NC↑	μ↓	R^2↓
Full model (MolField)	**0.9113**	**0.9469**	**9**	**63**
(a) C-INR Analysis				
w/o Canon.	0.8560	0.8980	23	146
Rand. Proj.	0.8066	0.8532	35	285
(b) SWT Analysis				
Flatten Weights	0.8925	0.9265	12	77
w/o PE	0.9025	0.9323	14	72
(c) FSHN Analysis				
Transformer → MLP	0.8795	0.9125	14	91
w/o Auto-Regressive	0.9010	0.9350	11	74

Experiments are conducted on protein trajectories from Sun et al. (2023), comparing against NDF (Sun et al., 2022), DSR (Sun et al., 2023), and CanFields (Wang et al., 2025a) with volumetric IoU, Chamfer Distance (CD), and Normal Consistency (NC) as metrics (Table 1). MolField achieves the best average performance on all three metrics, and the best IoU and normal consistency on five of the six systems. The largest gains are on adk_equi, YiiP, and NhaA. Compared to DSR, which fits an implicit network per trajectory, MolField generates the field parameters with a hyper-network conditioned on the molecule and the time step. Results for DSR are those reported by Sun et al. (2023) for reconstruction of training frames, with standard deviations omitted here. The overhead of generating the parameters rather than fitting them is small: on `adk_equi` (3,341 atoms) the encoder, SWT and FSHN add 14.4 ms per inference step against 5,298 ms of INR reconstruction that MolField and DSR both pay (Table 5, Appendix F.3).

Molecular Property Prediction. We evaluate MolField on QM9 (Ramakrishnan et al., 2014), following the standard 100K/18K/13K train/validation/test split of Anderson et al. (2019). Five scalar targets are considered: $\varepsilon_{\text{HOMO}}$, μ, R^2, α, and ZPVE, with mean absolute error (MAE) as the evaluation metric. Baselines include: (1) GNN-based methods, including SchNet (Schütt et al., 2017), Molformer (Wu et al., 2023), Cormorant (Anderson et al., 2019), SEGNN (Brandstetter et al., 2022), EGNN (Satorras et al., 2021), PaiNN (Schütt et al., 2021), NoisyNodes (Godwin et al., 2022), GNS-TAT (Zaidi et al., 2022), DimeNet++ (Gasteiger et al., 2020a), SphereNet (Liu et al., 2022), MACE (Batatia et al., 2022), GA-GNN (Petersen & Schmidt, 2026); (2) CNN and transformer-based architectures, such as LieConv (Finzi et al., 2020), Equiformer (Liao & Smidt, 2022; Liao et al., 2023), TorchMD-NET (Thölke & De Fabritiis, 2022), ViSNet (Wang et al., 2024a), Geoformer (Wang et al., 2023), and GotenNet (Aykent & Xia, 2025); and (3) function-based approaches, represented by FuncMol (Kirchmeyer et al., 2024). As shown in Table 2, MolField achieves the lowest MAE on polarizability (α, 21 ma_0^3) among the compared models and matches the best $\varepsilon_{\text{HOMO}}$ in Table 2 (14 meV). On μ (9 mD), R^2 (62 ma_0^2) and ZPVE (1.16 meV) it is competitive but behind recent equivariant networks such as GotenNet. On this task MolField is initialized from the density-field

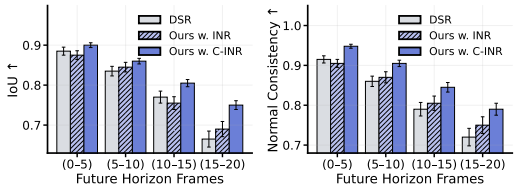

Figure 3: **Extended-horizon prediction of dynamic molecular surfaces.** Quantitative comparison of future surface prediction over increasing time horizons. MolField yields more accurate and stable extended-horizon predictions.

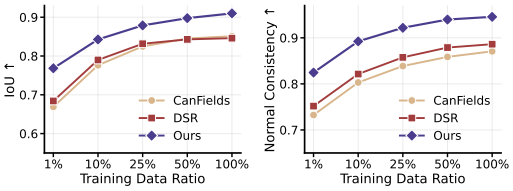

Figure 4: **Data efficiency on dynamic molecular surface modeling.** Performance of different methods under varying training data ratios. MolField attains higher IoU and NC than the compared baselines across the evaluated data ratios.

model trained for molecular generation, which uses no property labels. Table 2 also reports the same model trained from scratch: pretraining improves every target, from 23 to 14 meV on $\varepsilon_{\mathrm{HOMO}}$ and from 31 to 21 ma_0^3 on α, so the reported row is itself a transfer result obtained from a signal carrying no property labels. Because QM9 geometries are relaxed to equilibrium, we also report force prediction on MD17, whose conformations are sampled from molecular dynamics; MolField is on par with Equiformer there but behind TorchMD-NET and NequIP (Appendix G.5), showing the improvement space. Beyond accuracy, MolField also runs $\sim$1.9$\times$ faster than Equiformer at inference ($\sim$560 vs. $\sim$300 samples/sec on QM9), with about 30M parameters compared with 3.5M for Equiformer (Appendix F.2).

Ablation Study. Table 3 analyzes the contribution of each core component in MolField; its dynamics columns average the six trajectories of Table 1, and its property columns report QM9 μ (mD) and R^2 (ma_0^2) on the split of Table 2. Removing any component degrades performance on both tasks, indicating that the framework relies on their joint design. *(1) C-INR.* Removing canonicalization or replacing it with random projections leads to substantial performance drops, confirming the importance of SE(3)-consistent function evaluation. *(2) SWT.* Flattening weights or removing positional encodings degrades accuracy, showing that preserving the structured organization of function parameters is critical. *(3) FSHN.* Replacing the Transformer with an MLP or removing autoregressive generation consistently hurts performance, highlighting the importance of modeling dependencies among function components. Appendix G.4 steps outside it and compares against a latent-as-input INR, which conditions one shared parameter set on the coordinate instead of generating parameters. It reaches the desirable performance yet stays below MolField, indicating that inducing a molecular function is not by itself the same as inducing an effective representation.

Extension to Molecular Generation. The same architecture also admits a generative read-out: a condition drawn from the prior is decoded into a density field, from which a molecular graph is recovered by post-processing. On QM9 this is comparable to baselines in validity and stability but behind dedicated 3D generative models, so we report it as evidence that the formulation extends beyond prediction rather than as a state-of-the-art generative method (Table 6). However, our method degrades less than the compared models in the transferability analysis, as indicated by the QM9-to-Drugs transfer experiment (Table 7). See Appendix G.1 for details.

6 Analysis

We analyze the behavior of MolField across temporal horizons, data and label regimes, reconstruction fidelity, and input corruption.

Extended Horizon Prediction as Function Extrapolation. To assess temporal generalization, we evaluate extended-horizon prediction of molecular surfaces beyond the observed training window. Figure 3 reports IoU and Normal Consistency over increasing future horizons. While DSR fits an implicit network independently for each trajectory, MolField generates the field for a requested time step with the hyper-network. As the prediction horizon increases, the performance gap widens, with canonical implicit representations maintaining higher geometric fidelity for distant future frames.

Data Efficiency from Sampling-Independent Representations. Figure 4 evaluates performance under varying fractions of training data. MolField outperforms the compared baselines across data regimes, with the largest advantages under sparse supervision, and matches DSR trained on the full

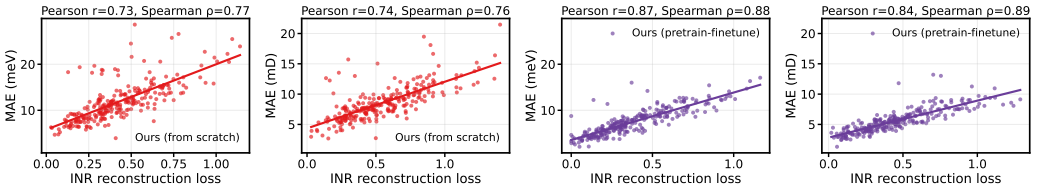

Figure 5: **Correlation between INR reconstruction loss and property prediction error (MAE) of MolField.** Positive Pearson and Spearman correlations indicate that more accurate function-space reconstructions are associated with lower property prediction error. Pretraining the INR on molecular generation further strengthens this relationship and consistently reduces prediction error.

dataset with about 25% of the data. Discarding frames removes part of what a per-trajectory fit represents, whereas for a generated field it only removes queries of a function that remains defined everywhere; because the parameters come from a hyper-network shared across molecules and time steps, the retained frames also constrain the dropped ones.

Label Efficiency in Property Prediction. The same question arises for the property task, where the scarce resource is labels rather than frames. Figure 7 in Appendix G.3 reports QM9 MAE under varying labeled-data fractions. MolField attains lower MAE than FuncMol at every label ratio on all five targets, and initializing from the density-field model improves over training from scratch at every ratio, with the largest margin in the low-label regime. Because that initialization is obtained without property labels, the gain suggests that part of what supervised training would otherwise have to learn is already carried by the generated field parameters.

Function Fidelity Is an Indicator of Representation Quality. Figure 5 analyzes the relationship between INR reconstruction loss on density prediction and property prediction error for $\varepsilon_{\mathrm{HOMO}}$ and μ; the remaining three targets are shown in Figure 6 (Appendix G.2). Across all five targets we observe positive Pearson and Spearman correlations: molecules whose fields are reconstructed more accurately tend to yield lower prediction errors. This trend holds both when training from scratch and under a pretrain–finetune regime, with pretraining further strengthening the correlation and concentrating predictions at lower MAE values. These results indicate that reconstruction fidelity is an observable indicator of the quality of the generated representation. The analysis is correlational and compares across molecules, so molecular complexity may affect both quantities.

Robustness to Input Corruption. We corrupt inputs by replacing the type feature of a random fraction of atoms with a mask symbol while leaving positions intact, and give the same corrupted molecule to three settings: the discrete 3D graph (Orig. 3D), a graph re-read from the generated density field (MolField-3D), and direct prediction from the field parameters (MolField-INR). The first two use an Equiformer backbone (Liao & Smidt, 2022). Table 4 shows the error growing fastest on the original graph, while MolField-INR has the smallest absolute increase at every corruption level. MolField-3D degrades more slowly than Orig. 3D because atom types are recovered as local maxima of the per-type channels, that is, from spatial context rather than propagated through the graph as a mask.

Table 4: **Robustness under input corruption**, measured by MAE on α (ma_0^3; lower is better). Values in parentheses denote the absolute increase in error relative to the clean setting.

Corruption	Orig. 3D*	MolField-3D*	MolField-INR
0%	46	25	21
25%	64 **(+18)**	33 **(+8)**	26 **(+5)**
50%	91 **(+45)**	54 **(+29)**	45 **(+24)**
75%	158 **(+112)**	68 **(+43)**	59 **(+38)**

7 Conclusion

We presented MolField, which formulates molecular learning as learning in function space by neural fields. Across dynamic molecular surface modeling, property prediction, and molecular generation, this formulation yields accurate extended-horizon prediction, higher data efficiency, and smaller degradation under input corruption. We believe this perspective offers a useful bridge between molecular modeling and neural field methods. We position MolField as a data-driven, geometry-level representation that complements physics-driven approaches (Appendix H).

AI USE STATEMENT

In this work, we used generative AI tools only for language polishing, improving the grammar, wording, and clarity of text drafted by the authors. We have not used generative AI tools for research ideation, for the design of the proposed method or the experimental protocol, for implementation, for the generation or analysis of results, for the creation of figures and tables, or for literature discovery and citation; all of these are the original work of the authors. We have reviewed all AI-assisted work, checking every suggested revision against the authors' original text and the underlying results, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All data used in this work are publicly available: QM9 (Ramakrishnan et al., 2014) under the 100K/18K/13K split of Anderson et al. (2019), the protein molecular dynamics trajectories released by Sun et al. (2023), GEOM-Drugs (Axelrod & Gomez-Bombarelli, 2022), and MD17 (Chmiela et al., 2017) under its standard split. Appendix E documents the setup for each task, including the field that is supervised, the evaluation metrics, the hyper-parameters and optimizer settings, and the hardware used; Appendix D specifies the equivariant encoder; Appendix C gives pseudo-code for end-to-end training; and Appendix B contains the full proofs of Proposition 4.1 and Lemma 4.2, together with the conditions under which the canonical frame is degenerate. The processing applied to the trajectory data and the post-processing used to recover discrete molecules from generated fields are described in Appendix E.1 and Appendix G.1. Code will be made publicly available upon publication.

REFERENCES

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Advances in neural information processing systems*, volume 32, 2019.

Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Sudarshan Babu, Phillip Lo, Xiao Zhang, Aadi Srivastava, Ali Davariashtiyani, Jason Perera, Michael Maire, and Aly A. Khan. HyperDiffusionFields (HyDiF): Diffusion-guided hypernetworks for learning implicit molecular neural fields. *arXiv preprint arXiv:2510.18122*, 2025. NeurIPS 2025 Workshop on AI Virtual Cells and Instruments (AI4D3).

Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B—Condensed Matter and Materials Physics*, 87(18):184115, 2013.

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. volume 35, pp. 11423–11436, 2022.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

Nurken Berdigaliyev and Mohamad Aljofan. An overview of drug discovery and development. *Future medicinal chemistry*, 12(10):939–947, 2020.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e (3) equivariant message passing. In *International Conference on Learning Representations (ICLR)*, 2022.

Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pp. 170–187. Springer, 2022.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017.

Richard D Cramer, David E Patterson, and Jeffrey D Bunce. Comparative molecular field analysis (comfa). 1. effect of shape on binding of steroids to carrier proteins. *Journal of the American Chemical Society*, 110(18):5959–5967, 1988.

Gordon M Crippen, Timothy F Havel, et al. *Distance geometry and molecular conformation*, volume 74. Research Studies Press Taunton, 1988.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Beatriz G. del Rio, Brandon Phan, and Rampi Ramprasad. A deep learning framework to emulate density functional theory. *npj Computational Materials*, 9(1):158, 2023.

Alexandru Dumitrescu, Dani Korpela, Markus Heinonen, Yogesh Verma, Valerii Iakovlev, Vikas Garg, and Harri Lähdesmäki. E(3)-equivariant models cannot learn chirality: Field-based molecular generation. In *International Conference on Learning Representations*, 2025.

Emilien Dupont, Hyunjik Kim, S. M. Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning (ICML)*, 2022.

Nadav Dym, Hannah Lawrence, and Jonathan W. Siegel. Equivariant frames and the impossibility of continuous canonicalization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12228–12267. PMLR, 2024.

Amer Essakine, Yanqi Cheng, Chun-Wun Cheng, Lipei Zhang, Zhongying Deng, Lei Zhu, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Where do we stand with implicit neural representations? a technical and performance survey. *arXiv preprint arXiv:2411.03688*, 2024.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to Lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3165–3176. PMLR, 2020.

James Franklin. Discrete and continuous: a fundamental dichotomy in mathematics. *Journal of Humanistic Mathematics*, 7(2):355–378, 2017.

Roman Frigg and James Nguyen. Scientific representation. 2016.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in neural information processing systems*, volume 33, pp. 1970–1981, 2020.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4181–4192, 2021.

Johannes Gasteiger, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020a.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020b.

Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

Jonathan Godwin, Michael Schaarschmidt, Alexander L Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple gnn regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations (ICLR)*, 2022.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

David Ha, Andrew M Dai, and Quoc V Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic schrödinger equation. *Nature Chemistry*, 12(10):891–897, 2020.

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

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Lei Huang, Hengtong Zhang, Tingyang Xu, and Ka-Chun Wong. Mdm: Molecular diffusion model for 3d molecule generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5105–5112, 2023.

Frank Jensen. *Introduction to computational chemistry*. John wiley & sons, 2017.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15546–15566. PMLR, 2023.

Matthieu Kirchmeyer, Pedro O O Pinheiro, and Saeed Saremi. Score-based 3d molecule generation with neural fields. *Advances in Neural Information Processing Systems*, 37:10646–10678, 2024.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

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

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

Youhan Lee, Hasun Yu, Jaemyung Lee, and Jaehoon Kim. Pre-training sequence, structure, and surface features for comprehensive protein representation learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph diffusion transformers for multi-conditional molecular generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8065–8092, 2024.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International conference on learning representations*, 2022.

Johannes T Margraf and Karsten Reuter. Pure non-local machine-learned density functional theory for electron correlation. *Nature communications*, 12(1):344, 2021.

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7210–7219, 2021.

Łukasz Maziarka, Tomasz Danel, Sławomir Mucha, Krzysztof Rataj, Jacek Tabor, and Stanisław Jastrzebski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020.

Kenneth M Merz Jr. Using quantum mechanical approaches to study biological systems. *Accounts of Chemical Research*, 47(9):2804–2811, 2014.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Jane S Murray and Kalidas Sen. *Molecular electrostatic potentials: concepts and applications*. Elsevier, 1996.

Van Khoa Nguyen, Maciej Falkiewicz, Giangiacomo Mercatali, and Alexandros Kalousis. MING: A functional approach to learning molecular generative models. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1279–1287. PMLR, 2025.

Pedro O O Pinheiro, Joshua Rackers, Joseph Kleinhenz, Michael Maser, Omar Mahmood, Andrew Watkins, Stephen Ra, Vishnu Sresht, and Saeed Saremi. 3d molecule generation by denoising voxel grids. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69077–69097, 2023.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

Kasper Helverskov Petersen and Mikkel N Schmidt. Design and evaluation of a geometric algebra-based graph neural network for molecular property prediction. In *Proceedings of the 7th Northern Lights Deep Learning Conference (NLDL)*, 2026.

Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J. Smith, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, volume 30, 2017.

Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International conference on machine learning*, pp. 9377–9388. PMLR, 2021.

Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 10136–10147, 2020a.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. volume 33, pp. 7462–7473, 2020b.

Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 549–568, 2023.

Afnan Sultan, Jochen Sieg, Miriam Mathea, and Andrea Volkamer. Transformers for molecular property prediction: Lessons learned from the past five years. *Journal of Chemical Information and Modeling*, 64(16):6259–6280, 2024.

Daiwen Sun, He Huang, Yao Li, Xinqi Gong, and Qiwei Ye. Dsr: Dynamical surface representation as implicit neural networks for protein. *Advances in Neural Information Processing Systems*, 36: 13873–13886, 2023.

Shanlin Sun, Kun Han, Deying Kong, Hao Tang, Xiangyi Yan, and Xiaohui Xie. Topology-preserving shape reconstruction and registration via neural diffeomorphic flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20845–20855, 2022.

Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15272–15281, 2021.

Philipp Thölke and Gianni De Fabritiis. Torchmd-net: equivariant transformers for neural network based molecular potentials. *arXiv preprint arXiv:2202.02541*, 2022.

Kenno Vanommeslaeghe, Olgun Guvench, et al. Molecular mechanics. *Current pharmaceutical design*, 20(20):3281–3292, 2014.

Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *ICLR*, 2023.

Miaowei Wang, Changjian Li, and Amir Vaxman. Canfields: Consolidating diffeomorphic flows for non-rigid 4d interpolation from arbitrary-length sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28587–28598, 2025a.

Yusong Wang, Shaoning Li, Tong Wang, Bin Shao, Nanning Zheng, and Tie-Yan Liu. Geometric transformer with interatomic positional encoding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55981–55994, 2023.

Yusong Wang, Tong Wang, Shaoning Li, Xinheng He, Mingyu Li, Zun Wang, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Enhancing geometric representations for molecules with equivariant vector-scalar interactive message passing. *Nature Communications*, 15(1):313, 2024a.

Yuyang Wang, Ahmed A. A. Elhag, Navdeep Jaitly, Joshua M. Susskind, and Miguel Ángel Bautista. Swallowing the bitter pill: Simplified scalable conformer generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50400–50418. PMLR, 2024b.

Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, et al. Graph foundation models: A comprehensive survey. *arXiv preprint arXiv:2505.15116*, 2025b.

Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei, and Ming Lei. Machine learning in materials science. *InfoMat*, 1(3):338–358, 2019.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Michael J Willatt, Félix Musil, and Michele Ceriotti. Atom-density representations for machine learning. *The Journal of chemical physics*, 150(15), 2019.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in neural information processing systems*, volume 33, pp. 4697–4708, 2020.

Fang Wu and Stan Z. Li. Surface-VQMAE: Vector-quantized masked auto-encoders on molecular surfaces. In *Forty-first International Conference on Machine Learning*, 2024.

Fang Wu, Dragomir Radev, and Stan Z Li. Molformer: Motif-based transformer on 3d heterogeneous molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5312–5320, 2023.

Fang Wu, Bozhen Hu, and Stan Z Li. Generalized implicit neural representations for dynamic molecular surface modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 877–885, 2025.

Yuning You, Ruida Zhou, Jiwoong Park, Haotian Xu, Chao Tian, Zhangyang Wang, and Yang Shen. Latent 3d graph diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=cXbnGtO0NZ.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.

Pierluigi Zama Ramirez, Luca De Luigi, Daniele Sirocchi, Adriano Cardace, Riccardo Spezialetti, Francesco Ballerini, Samuele Salti, and Luigi Di Stefano. Deep learning on object-centric 3d neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2024.

Chengxi Zang and Fei Wang. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 617–626, 2020.

Jiachi Zhao, Zehong Wang, Yamei Liao, Chuxu Zhang, and Yanfang Ye. Controllable graph generation with diffusion models via inference-time tree search guidance. *arXiv preprint arXiv:2510.10402*, 2025.

A Additional Related Work

A.1 Implicit Neural Representation Generation

Weight space generation studies how to directly model and generate the parameters of neural networks, rather than their outputs or latent embeddings. Early work introduces hypernetworks as conditional generators that map an input code or contextual signal to the weights of a target network (Ha et al., 2017). From this perspective, neural networks are treated as points in a structured weight space, where shared statistical regularities across tasks, instances, or signals can be captured and exploited. This formulation enables generalization at the level of functions themselves, and has been explored in contexts such as meta-learning, conditional computation, and neural function priors (Finn et al., 2017; Wilson & Izmailov, 2020). Within this broader framework, INRs provide a natural and challenging testbed for weight space generation. An INR encodes a continuous signal as a neural function whose parameters fully determine the represented object. Classical INR construction relies on per-instance optimization via gradient descent, which is computationally expensive and often unstable under sparse or noisy observations (Park et al., 2019; Sitzmann et al., 2020b). Weight space generation offers an alternative by amortizing this optimization process: instead of iteratively fitting network parameters, a learned generator predicts the weights of an INR directly from observations, effectively instantiating a neural function in a single forward pass. This approach enables fast inference and highlights a principled connection between INRs, hypernetworks, and learned priors over neural functions (Chen & Wang, 2022; Sitzmann et al., 2020a)

A.2 Molecule Generation

Early molecular generative models primarily focused on 2D representations, treating molecules as SMILES strings or discrete graphs and modeling their distributions using VAEs (Gómez-Bombarelli et al., 2018; Jin et al., 2018), GANs (De Cao & Kipf, 2018), normalizing flows (Zang & Wang, 2020) or more recently discrete diffusion models on graphs (Vignac et al., 2023; Liu et al., 2024; Wang et al., 2025b). These approaches have achieved notable success in topology generation and property-conditioned design, but they do not explicitly model three-dimensional molecular geometry. More recent work has shifted toward 3D molecule generation, where atoms are represented with continuous coordinates and categorical types. Representative approaches include normalizing flows (Rezende & Mohamed, 2015; Köhler et al., 2020; Garcia Satorras et al., 2021), as well as a broad family of diffusion-based methods, encompassing equivariant diffusion models (Hoogeboom et al., 2022; Huang et al., 2023), flow matching formulations (Song et al., 2023), and score-based generative models (O Pinheiro et al., 2023; Zhao et al., 2025). Beyond point-cloud representations, recent work has further extended score-based generation to INRs, modeling molecules as continuous spatial fields rather than discrete atomic sets (Kirchmeyer et al., 2024).

A.3 Molecular Surface Reconstruction

Molecular surfaces provide a compact and physically meaningful interface between biomolecules and their environment, encoding geometric and chemical cues that govern key interactions such as protein–ligand binding and allosteric regulation. Accurately modeling these surfaces as they evolve over time is therefore essential for understanding functional conformational changes and for enabling downstream tasks that rely on differentiable geometry. Recent advances in dynamic molecular surface modeling leverage implicit neural representations to describe protein surfaces as continuous, time-dependent fields. Dynamical Surface Representation (DSR) models a protein surface as the zero-level set of a signed distance function (SDF) parameterized by an implicit neural network, with time explicitly incorporated as an input (Sun et al., 2023). This formulation yields a continuous representation over both space and time, enabling smooth interpolation and extrapolation of surface dynamics. Training is performed directly from surface point clouds, without requiring precomputed SDF supervision, by enforcing manifold consistency and Eikonal regularization, while latent codes are used to capture protein-specific variability. To improve generalization across diverse protein families, subsequent work extends this framework with a mixture-of-experts architecture (MoE-DSR), where multiple implicit experts specialize in different structural distributions (Wu et al., 2025). A learned routing mechanism extracts geometric information from the initial conformation to dynamically select experts, enhancing robustness to unseen proteins.

A.4 Molecular Property Prediction

Molecular property prediction aims to infer physicochemical and quantum-mechanical properties from molecular structure and has been extensively studied under a variety of representation and modeling paradigms. Graph neural networks constitute the dominant framework in this area, representing molecules as atom–bond graphs and learning molecular representations through iterative message passing (Gilmer et al., 2017; Schütt et al., 2017; Gasteiger et al., 2020a). To more faithfully capture three-dimensional molecular geometry, subsequent approaches explicitly incorporate geometric information such as interatomic distances, bond angles, and higher-order features, giving rise to directional message passing architectures exemplified by DimeNet and its variants (Gasteiger et al., 2020b;a). More recent work enforces geometric symmetries via equivariant neural networks, including EGNN and SE(3)-equivariant architectures, which guarantee consistent transformations under rotations and translations (Satorras et al., 2021; Fuchs et al., 2020). In parallel, transformer-based models have also been explored for molecular property prediction, leveraging global attention mechanisms to capture long-range interactions beyond local neighborhoods (Sultan et al., 2024; Maziarka et al., 2020).

A.5 Traditional Molecular functions and the Scope of MolField

Continuous functions are already central to molecular science. Our use of function space should therefore be understood in the sense of neural representation learning: MolField represents each molecule as a learnable INR-parameterized field, rather than as a fixed physical observable or an electronic-structure quantity. In electronic-structure theory, wavefunctions are tied to the Schrödinger equation, while electron densities and density functionals form the basis of density-functional theory (Hohenberg & Kohn, 1964; Kohn & Sham, 1965; Hermann et al., 2020; Margraf & Reuter, 2021). Molecular modeling has also used spatial fields, such as molecular electrostatic potentials and CoMFA-style interaction fields, to characterize interactions with charges, probes, or molecular environments (Murray & Sen, 1996; Cramer et al., 1988). Atomistic machine learning further represents local environments through smooth atom densities, as in SOAP and related density-based descriptors (Bartók et al., 2013; Willatt et al., 2019). The key distinction is the role played by the function. Traditional continuous descriptions are typically physical state variables, observables, interaction fields, or fixed descriptors derived from electronic-structure theory, probe-based modeling, or descriptor design. MolField instead treats the function as a learnable neural representation: the parameters of an INR-parameterized molecular field are generated by a hyper-network and reused across tasks. The learned object is therefore not a value sampled at spatial coordinates, but the parameters of a neural function. This distinction also clarifies the scope of MolField. It is a geometry-level representation framework, not an electronic-structure solver. It does not impose the physical constraints required by wavefunction or density-functional methods, such as antisymmetry, variational energy minimization, or explicit Hamiltonian structure.

B Proofs

B.1 Proof of Proposition 4.1

Proof. The proof proceeds in three stages: the equivariance of learned vector features, the equivariance of aggregated global axes, and the preservation of equivariance under the Gram-Schmidt orthogonalization and cross-product completion.

Equivariance of vector features. The initial features $\mathbf{v}_i^{(0)}$ are constructed from relative coordinates $\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i$. Under a rotation R, the centered coordinates transform as $R\tilde{\mathbf{x}}$. Thus, the relative difference becomes $R(\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i)$. Since the initialization function ψ and subsequent message-passing weights $\mathbf{W}_v, \eta$ act only on the invariant channel dimension (scalars), they commute with the spatial rotation matrix R. By induction, if $\mathbf{v}_i^{(\ell-1)}(RX) = R\mathbf{v}_i^{(\ell-1)}(X)$, then due to the linearity of the update rule, the refined features satisfy $\mathbf{v}_i^{(\ell)}(RX) = R\mathbf{v}_i^{(\ell)}(X)$ for all ℓ.

Equivariance of aggregated axes. The global axes $\mathbf{u}_1, \mathbf{u}_2$ are obtained via weighted linear combinations of $\mathbf{v}_i$. Since the attention weights are derived from invariant scalars, they remain constant under rotation. Because matrix multiplication is a linear operator, the aggregation preserves equivari-

ance:

$$\mathbf{u}_k(RX) = \sum_i \alpha_i^{(k)} R\mathbf{v}_i(X) = R\sum_i \alpha_i^{(k)} \mathbf{v}_i(X) = R\mathbf{u}_k(X), \quad \text{for } k \in \{1,2\}. \tag{13}$$

Equivariance of the orthonormal basis. We verify the construction of the basis vectors $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ step-by-step:

- **First Axis ($\mathbf{q}_1$):** Normalization is scaling by a scalar length, which is rotation-invariant ($\|R\mathbf{u}\| = \|\mathbf{u}\|$). Thus:

$$\mathbf{q}_1(RX) = \frac{R\mathbf{u}_1}{\|R\mathbf{u}_1\|} = R\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} = R\mathbf{q}_1(X). \tag{14}$$

- **Second Axis ($\mathbf{q}_2$ via Gram-Schmidt):** The projection step removes the component of $\mathbf{u}_2$ parallel to $\mathbf{q}_1$. Let $\tilde{\mathbf{u}}_2 = \mathbf{u}_2 - (\mathbf{u}_2^\top \mathbf{q}_1)\mathbf{q}_1$. Under rotation R:

$$\begin{aligned}
\tilde{\mathbf{u}}_2(RX) &= R\mathbf{u}_2 - ((R\mathbf{u}_2)^\top (R\mathbf{q}_1))(R\mathbf{q}_1) && (15)\\
&= R\mathbf{u}_2 - (\mathbf{u}_2^\top R^\top R\mathbf{q}_1)(R\mathbf{q}_1) && (16)\\
&= R\mathbf{u}_2 - (\mathbf{u}_2^\top \mathbf{I}\mathbf{q}_1)R\mathbf{q}_1 && (17)\\
&= R(\mathbf{u}_2 - (\mathbf{u}_2^\top \mathbf{q}_1)\mathbf{q}_1) = R\tilde{\mathbf{u}}_2(X). && (18)
\end{aligned}$$

  Since $\tilde{\mathbf{u}}_2$ rotates equivariantly, its normalized version $\mathbf{q}_2$ also satisfies $\mathbf{q}_2(RX) = R\mathbf{q}_2(X)$.

- **Third Axis ($\mathbf{q}_3$ via Cross Product):** A critical property of the cross product for rotation matrices $R \in SO(3)$ (where $\det(R) = 1$) is $R(\mathbf{a} \times \mathbf{b}) = (R\mathbf{a}) \times (R\mathbf{b})$. Thus:

$$\begin{aligned}
\mathbf{q}_3(RX) &= \mathbf{q}_1(RX) \times \mathbf{q}_2(RX) && (19)\\
&= (R\mathbf{q}_1) \times (R\mathbf{q}_2) && (20)\\
&= R(\mathbf{q}_1 \times \mathbf{q}_2) = R\mathbf{q}_3(X). && (21)
\end{aligned}$$

  Note that this step strictly relies on the right-handed nature of the frame; for reflections, this equality would flip the sign.

Stacking the columns, we obtain:

$$Q(RX) = [R\mathbf{q}_1, R\mathbf{q}_2, R\mathbf{q}_3] = R[\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3] = RQ(X). \tag{22}$$

□

Discussion. The canonical frame construction assumes $\mathbf{u}_1 \neq \mathbf{0}$ and a non-degenerate orthogonalized $\tilde{\mathbf{u}}_2$; Proposition 4.1 holds on this set. These assumptions fail on molecules whose point group is neither C_1 nor C_s, and not merely in pathological cases. Since $\mathbf{u}_1$ and $\mathbf{u}_2$ are aggregated with invariant weights from vector features built only from relative position vectors and invariant gates, these features are O(3)-equivariant; hence any point-group symmetry $S \in \mathrm{O}(3)$ of a molecule ($SX = X$ up to permutation) forces $\mathbf{u}_k = S\,\mathbf{u}_k$, so both vectors lie in the subspace fixed by the point group. For molecules with only a mirror plane (C_s) the fixed subspace is two-dimensional and the frame remains well defined; for molecules with a rotation axis (C_n, C_{nv}, $C_{\infty v}$ with $n \geq 2$) the fixed subspace is the axis itself, so $\mathbf{u}_1 \parallel \mathbf{u}_2$ and $\mathbf{q}_2$ is undetermined; and for all remaining point groups the fixed subspace is trivial, giving $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{0}$. This is a known limitation of equivariant canonicalization rather than an artifact of our construction: for the action of $SO(3)$ on point clouds with at least three points, no continuous equivariant canonicalization exists (Kaba et al., 2023; Dym et al., 2024). Analyzing the point groups of the QM9 geometries indicates that about 0.8% of the molecules fall into the degenerate classes, while the remaining molecules are C_1 or C_s. Near this set the normalization is ill-conditioned: the angular change induced by a perturbation scales as $O(1/\|\mathbf{u}\|)$. In practice, we apply ϵ-regularized normalization for numerical stability. Averaging over a frame set (Puny et al., 2022), and in particular the weighted frames of Dym et al. (2024), are principled remedies that we leave to future work.

B.2 Proof of Lemma 4.2

Proof. Let $X' = g \cdot X$ denote the transformed molecular geometry and $\mathbf{x}' = R\mathbf{x} + \mathbf{t}$ denote the transformed query point. First, observe that the centroid shifts equivariantly: $\bar{\mathbf{x}}' = \frac{1}{N}\sum(R\mathbf{x}_i + \mathbf{t}) = R\bar{\mathbf{x}} + \mathbf{t}$. Second, from Proposition 4.1, the frame rotates only: $Q(X') = RQ(X)$. Substituting these into the query definition:

$$
\begin{aligned}
c(\mathbf{x}'; X') &= Q(X')^\top(\mathbf{x}' - \bar{\mathbf{x}}') && (23)\\
&= (RQ(X))^\top\left((R\mathbf{x} + \mathbf{t}) - (R\bar{\mathbf{x}} + \mathbf{t})\right) && (24)\\
&= Q(X)^\top R^\top\left(R(\mathbf{x} - \bar{\mathbf{x}}) + (\mathbf{t} - \mathbf{t})\right) && (25)\\
&= Q(X)^\top R^\top R(\mathbf{x} - \bar{\mathbf{x}}) && (26)\\
&= Q(X)^\top \mathbf{I}(\mathbf{x} - \bar{\mathbf{x}}) && (27)\\
&= c(\mathbf{x}; X). && (28)
\end{aligned}
$$

This demonstrates that the canonical coordinate mapping implicitly handles translation via relative differences and rotation via frame projection, ensuring the downstream INR receives consistent inputs regardless of the global pose. □

Discussion. This invariance is crucial: the INR weights $\boldsymbol{\Theta}$ only learn to recognize features in the canonical frame $\mathbf{x}'$. During inference (e.g., surface reconstruction), we query the field at $\mathbf{x}$ by passing its canonical counterpart $\mathbf{x}'$ into the network. This allows the network output to track the molecule's rotation without the weights themselves being equivariant.

C Pseudo-Code

End-to-End Training Pipeline. Algorithm 1 summarizes the end-to-end training procedure of MolField in function space. Given a training sample $(\mathcal{I}, y)$, the model first constructs a task-dependent condition z from the input. Conditioned on z, a transformer-based hyper-network with a causal mask produces a structured sequence of weight tokens, which are subsequently detokenized into the parameters $\boldsymbol{\Theta}$ of a C-INR. This process instantiates a continuous molecular field $f_{\boldsymbol{\Theta}}$ in a single forward pass, without per-instance optimization.

To train the generated field, MolField follows a generate-then-query paradigm. A set of spatial query points is sampled in $\mathbb{R}^3$ according to the task and input, and each query point is mapped to a canonical coordinate system defined by the molecular geometry. The implicit field is evaluated at these canonical coordinates, producing field values as the final prediction. A task loss is then computed by comparing the prediction with the supervision signal, and gradients are backpropagated through the C-INR queries and the hyper-network, enabling end-to-end optimization of the hyper-network parameters.

Importantly, in this illustrative single-task setting, the pipeline is fully specified by the condition z. Across tasks, the same pipeline applies with task-specific conditioning defined in the main text. The same formulation applies to dynamic surface modeling, property prediction, and generation, with a separately trained model for each task.

D Equivariant Encoder Details

Our encoder is designed to extract SE(3)-invariant scalar features $H = \{h_i\}_{i=1}^N \in \mathbb{R}^{N\times C}$ and SO(3)-equivariant vector features $V = \{\mathbf{v}_i\}_{i=1}^N \in \mathbb{R}^{N\times C\times 3}$ from the input molecular graph. The architecture follows a dual-stream message passing scheme, ensuring that geometric information propagates efficiently while strictly preserving equivariance.

D.1 Initialization

The input consists of atomic numbers Z and positions X. We perform the following initializations before the first layer:

Algorithm 1 MolField: End-to-End Training in Function Space

Require: Dataset $\mathcal{D} = \{(\mathcal{I}, y)\}$ for a fixed task; condition z associated with $\mathcal{I}$; hyper-network G_ϕ; C-INR with L layers; detokenizers $\textsc{Proj}_{(\ell,r)}(\cdot)$; equivariant encoder $\Phi(\cdot)$; query sampler $S(z, \mathcal{I})$; loss $\ell(\cdot, \cdot)$
Ensure: Trained hyper-network parameters ϕ

Initialize ϕ
while not converged **do**
  Sample mini-batch $\mathcal{B} = \{(\mathcal{I}, y)\} \sim \mathcal{D}$
  for all $(\mathcal{I}, y) \in \mathcal{B}$ **do**
    (1) Generate C-INR parameters via hyper-network
    for $\ell = 1$ **to** L **do**
      for all $r \in \{\texttt{weight}, \texttt{bias}\}$ **do**
        $\hat{\tau}_t \leftarrow G_\phi(\tilde{\tau}_{\le t}, z)$ // causally masked transformer, single forward pass
        $\theta^{(\ell,r)} \leftarrow \textsc{Proj}_{(\ell,r)}(\hat{\tau}_t)$
      end for
    end for
    Instantiate field network $f_{\boldsymbol{\Theta}}$
    (2) Sample one spatial query point
    $\mathbf{x} \leftarrow S(z, \mathcal{I})$ // sample one 3D point (illustration)
    (3) Canonical querying (C-INR)
    Extract molecular geometry (X, A) from $\mathcal{I}$
    $\bar{\mathbf{x}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$
    $\{\mathbf{v}_i\} \leftarrow \Phi(X, A)$
    Build canonical frame $Q(X) \in SO(3)$ from $\{\mathbf{v}_i\}$
    $\mathbf{x}' \leftarrow Q(X)^\top (\mathbf{x} - \bar{\mathbf{x}})$
    $\hat{y} \leftarrow f_{\boldsymbol{\Theta}}(\mathbf{x}')$ // queried field value
    (4) End-to-end optimization
    $L \leftarrow \ell(\hat{y}, y)$
    Update ϕ by backpropagating $\nabla_\phi L$
  end for
end while
return ϕ

Scalar initialization. The scalar features $h_i^{(0)}$ are initialized by embedding the atomic numbers using a learnable lookup table:

$$h_i^{(0)} = \text{Embedding}(Z_i) \in \mathbb{R}^C. \tag{29}$$

Edge embedding. To capture pairwise geometry, we compute the relative distance $d_{ij} = \|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i\|$ and encode it using a set of K Radial Basis Functions (RBF), followed by a linear projection:

$$e_{ij} = \text{Linear}(\text{RBF}(d_{ij})) \in \mathbb{R}^C. \tag{30}$$

This edge embedding e_{ij} is invariant to SE(3) transformations and acts as a geometric bias in message passing.

Vector initialization. We initialize the vector features $\mathbf{v}_i^{(0)}$ by aggregating local bond directions. This explicitly couples the vector stream with the molecular geometry:

$$\mathbf{v}_i^{(0)} = \sum_{j \in \mathcal{N}(i)} \psi(h_i^{(0)}, h_j^{(0)}) \cdot \frac{\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i}{|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i|}, \tag{31}$$

where $\psi : \mathbb{R}^{2C} \to \mathbb{R}^C$ is a shallow MLP that weights the bond directions based on the chemical species of the atom pair.

D.2 Equivariant Message Passing Layer

Each layer ℓ updates both scalar and vector features. The update involves two stages: interaction-aware message passing and channel-wise mixing.

Vector update (equivariant stream). We update the vector features by aggregating information from neighbors. Crucially, we preserve rotation equivariance by restricting operations to linear combinations of existing vectors. The update rule is:

$$\mathbf{v}_i^{(\ell)} = \mathbf{W}_v \mathbf{v}_i^{(\ell-1)} + \sum_{j \in \mathcal{N}(i)} \eta_{ij}^{(\ell)} \cdot \mathbf{v}_j^{(\ell-1)}, \tag{32}$$

where $\mathbf{W}_v \in \mathbb{R}^{C \times C}$ is a learnable channel-mixing matrix. The coefficients $\eta_{ij}^{(\ell)}$ are scalar gating values derived from the invariant stream, allowing scalar information to modulate vector propagation:

$$\eta_{ij}^{(\ell)} = \text{MLP}_{gate}\left(h_i^{(\ell-1)}, h_j^{(\ell-1)}, e_{ij}\right) \in \mathbb{R}^C. \tag{33}$$

This mechanism ensures that the vector aggregation is sensitive to both chemical context and scalar distances.

Scalar update (invariant stream). The scalar features are updated by aggregating messages from neighbors and, importantly, fusing information from the vector stream. This allows angular and directional information (captured by vectors) to influence the scalar chemical properties:

$$m_{ij}^{(\ell)} = \phi_{msg}\left(h_i^{(\ell-1)}, h_j^{(\ell-1)}, e_{ij}\right), \tag{34}$$

$$h_i^{(\ell)} = h_i^{(\ell-1)} + \text{MLP}_{upd}\left(\sum_{j \in \mathcal{N}(i)} m_{ij}^{(\ell)} \oplus \|\mathbf{v}_i^{(\ell)}\|\right). \tag{35}$$

Here, ϕ_{msg} is the message function, and $\|\mathbf{v}_i^{(\ell)}\| \in \mathbb{R}^C$ denotes the channel-wise norm of the updated vector features. By concatenating the vector norms, we allow the scalar features to sense the anisotropy of the local environment without breaking rotation invariance (since the norm of a vector is rotation-invariant).

E Experimental Details

E.1 Dynamic Surface Modeling

Molecular dynamics simulations provide a physically grounded and high-resolution description of protein conformational dynamics, and are therefore widely used as benchmarks for evaluating data-driven models of biomolecular motion. In this work, we follow Sun et al. (2023) to construct a comprehensive benchmark suite by combining multiple publicly available MD datasets, covering diverse protein sizes, structural classes, and dynamical regimes. Specifically, we include soluble enzymes undergoing equilibrium fluctuations and conformational transitions, as well as large membrane proteins with complex collective motions. To further assess generalization across related systems, we also incorporate multiple trajectories from the same protein families simulated under similar protocols. For computational efficiency and consistency, extremely long trajectories are temporally cropped to a fixed maximum length, while shorter trajectories are used in full. This combination of datasets enables evaluation of surface reconstruction accuracy and future-state prediction under realistic and heterogeneous protein dynamics. We follow Sun et al. (2023) for the experimental setup.

E.2 Molecular Property Prediction

In addition to protein dynamics benchmarks, we evaluate our model on the QM9 dataset for molecular property prediction (Ramakrishnan et al., 2014). QM9 is a widely used quantum chemistry benchmark consisting of approximately 134k small organic molecules with up to nine heavy atoms, where molecular geometries and properties are computed using density functional theory. Following standard practice, we consider five representative target properties that capture both electronic and structural characteristics of molecules: the highest occupied molecular orbital energy ($\varepsilon_{\text{HOMO}}$), dipole moment ($\mu$), electronic spatial extent (R^2), isotropic polarizability (α), and zero-point vibrational energy (ZPVE). These properties span different physical scales and units, providing a comprehensive assessment of a model's ability to learn quantum-mechanical observables from molecular structures. Performance is reported using mean absolute error for each target, and an average ranking across all properties is additionally used to summarize overall predictive accuracy. We follow the Anderson et al. (2019) for the experimental setup.

E.3 Molecular Generation

We further evaluate our model on unconditional three-dimensional molecular generation tasks using the QM9 (Ramakrishnan et al., 2014) and Drugs (Axelrod & Gomez-Bombarelli, 2022) datasets. QM9 contains small organic molecules with up to nine heavy atoms and well-optimized equilibrium geometries, serving as a standard benchmark for learning physically valid 3D molecular distributions. The Drugs dataset consists of substantially larger and more diverse drug-like compounds, providing a more challenging setting with increased structural complexity and conformational variability. We adopt the standard unconditional generation protocol, where models are trained to learn the joint distribution of molecular topology and atomic coordinates from the training set without conditioning on molecular properties or target structures. During training, molecules are represented by their atomic types and 3D coordinates, and the model is optimized with the field-reconstruction objective of Section 4.4. At inference time, molecules are generated from random latent samples, and full molecular geometries are produced in a single forward sampling process. We follow Hoogeboom et al. (2022); Kirchmeyer et al. (2024) for the experimental setup.

E.4 Implementation Details of MolField

MolField employs a Transformer-based hyper-network with 6 encoder layers, 8 attention heads, hidden dimension 2048, and dropout rate 0.1 generates weights for the C-INR, which is implemented as an 8-layer MLP with hidden dimensions [512, 512, 512, 512, 512, 512, 512, 512] and skip connections at layer 4. For the equivariant encoder, MolField uses an equivariant Graph Neural Network with 6 message-passing layers, hidden dimension 512, and edge features encoded using radial basis functions with 32 centers. For the dynamic surface modeling task, the time coordinates are encoded using 64-dimensional Fourier features projected to 512 dimensions. We train for 2000 epochs using Adam optimizer with initial learning rates 10^{-4} for the network and 10^{-3} for latent codes, both decayed by factor 0.5 every 200 epochs, with batch size 1 and 16 data loading workers. All experiments are conducted on NVIDIA 4 $\times$ A40 GPUs using PyTorch 1.12.0 and PyTorch Geometric.

F Efficiency Analysis

We analyze the computational footprint of MolField along three axes: per-module time complexity, end-to-end parameter count and inference speed, and component-wise wall-clock breakdown on a protein-scale system.

F.1 Per-Module Time Complexity

MolField consists of three modules whose costs scale differently with input. The equivariant encoder operates on a molecular graph with N atoms and at most k neighbors per atom, with L_{enc} message-passing layers and channel width C, giving a cost of $O(L_{\text{enc}} \cdot N \cdot k \cdot C^2)$. At query time, the C-INR is evaluated at M spatial locations through an L_{INR}-layer MLP of width d_{INR}, costing $O(M \cdot L_{\text{INR}} \cdot d_{\text{INR}}^2)$. The Structured Weight Tokenization (SWT) maps the C-INR parameters to T weight tokens at cost $O(T \cdot d_{\text{model}})$, where T is determined by the C-INR architecture and is independent of N. The Function-Space Hyper-Network (FSHN) then attends over this token sequence at cost $O(T^2 \cdot d_{\text{model}})$. The overall per-instance cost is therefore

$$O\big(\underbrace{L_{\text{enc}} \cdot N \cdot k \cdot C^2}_{\text{scales with molecule size}} + \underbrace{T^2 \cdot d_{\text{model}}}_{\text{fixed by INR architecture}} + \underbrace{M \cdot L_{\text{INR}} \cdot d_{\text{INR}}^2}_{\text{shared by all INR-based methods}}\big), \tag{36}$$

which has an important structural property: the attention sequence length T is governed by the size of the C-INR, not by the number of atoms. This decoupling is the key to scaling MolField to large biomolecular systems where token-per-atom or token-per-residue Transformers would be infeasible.

F.2 Parameter Count and Inference Speed

MolField has approximately 30M parameters in total, about an order of magnitude more than Equiformer (3.5M on QM9). Despite the additional hyper-network, MolField is faster at inference:

on QM9 (NVIDIA A40), MolField achieves ∼560 samples/sec compared with Equiformer's ∼300 samples/sec. The reason is that MolField achieves SE(3) invariance through a single canonical-frame construction, avoiding the expensive equivariant tensor products applied at every layer in Equiformer/EquiformerV2.

F.3 Component-Wise Wall-Clock on adk_equi

To assess the practical overhead introduced by the hyper-network on protein-scale inputs, we benchmark MolField against DSR (Sun et al., 2023) on `adk_equi` (3,341 atoms). DSR uses an INR per trajectory but does not include an encoder or hyper-network, so it isolates the cost of INR forward/backward and reconstruction. Table 5 reports per-step wall-clock times. In training, MolField's total per-step time (46.7 ms) is roughly $2\times$ DSR (22.0 ms), a modest overhead in exchange for generating the field parameters rather than fitting them per frame. In inference, the additional cost from the encoder, SWT, and FSHN (14.4 ms) is negligible relative to INR reconstruction (5,298 ms, 99.7% of total inference time), which is shared by all INR-based methods.

Table 5: **Per-step wall-clock breakdown on `adk_equi` (3,341 atoms).** Times are in milliseconds.

Stage	Component	MolField (ms)	DSR (ms)
Training	Encoder (GNN)	10.28	—
	SWT + FSHN	5.45	—
	INR forward + backward	31.00	22.00
Inference	Encoder (GNN)	10.49	—
	SWT + FSHN	3.93	—
	INR reconstruction	5,298.40	5,298.40

F.4 Scalability to Large Biomolecules

Because the FSHN sequence length T is fixed by the C-INR architecture, MolField's quadratic attention cost does not grow with molecular size. Increasing the C-INR depth L_{INR} adds more weight tokens and lengthens the sequence linearly, but molecular size enters MolField only through the upstream equivariant encoder, whose cost is comparable to standard equivariant GNNs and is not an overhead specific to function-space modeling. This is consistent with the empirical results in Table 1, where MolField remains effective on protein systems including I-FABP (131 residues), AdK (214 residues, used in DIMS, FRODA, and adk_equi), NhaA (388 residues), and YiiP (600 residues as a homodimer).

G Additional Experiments

G.1 Molecule Generation

In this section, we present supplementary results on molecular generation to illustrate that the proposed function-space formulation admits a coherent generative interpretation. We stress that molecular generation is *not* a primary evaluation target of MolField, nor is the framework optimized specifically for this task. Instead, these experiments serve to demonstrate the extensibility of learning molecular representations directly in implicit functional space, beyond the discriminative and predictive settings studied in the main body.

Experimental setting. We follow the experimental protocols of Kirchmeyer et al. (2024) and evaluate molecular generation on QM9 (Ramakrishnan et al., 2014) and Drugs (Axelrod & Gomez-Bombarelli, 2022). Generation in MolField is performed by sampling latent conditions from a Gaussian prior and instantiating continuous molecular fields through the learned hyper-network. Discrete molecular graphs are then recovered via standard post-processing procedures applied to the generated fields. Concretely, following Kirchmeyer et al. (2024), we evaluate the field on a molecule-centered Cartesian grid of 32^3 points, take the local maxima of each atom-type channel as candidate atoms and types, and then optimize each position continuously against the field. The grid

Table 6: **Evaluation on QM9 molecular generation.** We report validity and uniqueness (Valid & Uni), atom-level stability (AtomSta), and molecule-level stability (MolSta).

Method	Valid & Uni (%)	AtomSta (%)	MolSta (%)
ENF	39.4	85.0	4.9
G-SchNet	80.3	95.7	68.1
GDM	–	97.0	63.2
EDM	90.7	**98.7**	82.0
Ours	91.3	97.2	83.7
LDM-3DG	**95.3**	97.6	**86.9**

Table 7: **Transfer stability on Drugs.** Atom-level and molecule-level performance under scratch and transfer settings, together with the performance drop (Drop ↓), where lower is better.

Method	AtomSta			MolSta		
	Scratch	Transfer	Drop ↓	Scratch	Transfer	Drop ↓
EDM	97.8	91.5	**6.3**	40.3	36.7	**3.6**
$\text{FuncMol}_{\text{dec}}$	95.3	87.6	**7.7**	69.7	60.1	**9.6**
MolField	90.1	88.4	**1.7**	56.3	54.7	**1.6**

therefore initializes the search rather than discretizing the output: final coordinates are not snapped to grid points, and the C-INR itself remains queryable at any resolution. What is sampled at generation time is the latent, not the coordinates. We report commonly used generation metrics, including validity and uniqueness (Valid & Uni), atom-level stability (AtomSta), and molecule-level stability (MolSta). Comparisons are made against ENF (Garcia Satorras et al., 2021), G-SchNet (Gebauer et al., 2019), GDM and EDM (Hoogeboom et al., 2022), and LDM-3DG (You et al., 2024), using their reported configurations. The ENF, G-SchNet, GDM and EDM numbers in Table 6 are taken from Hoogeboom et al. (2022) and LDM-3DG from You et al. (2024), under the distance-lookup bonding protocol of Hoogeboom et al. (2022); Table 7 instead uses the protocol of Kirchmeyer et al. (2024) for transferability analysis.

Generation results. Table 6 reports quantitative results on molecular generation. MolField is not designed as a molecular generative model; against the baselines of Table 6 it produces valid, unique and largely stable molecules. This indicates that the learned function-space prior captures non-trivial molecular structure and produces coherent samples rather than degenerate or noisy outputs. Importantly, these results suggest that the same architecture admits a generative read-out, although it is not competitive with dedicated 3D molecular generative models.

Transferability under distribution shift. We further consider a transfer generation setting, where the model is pretrained on QM9 and finetuned on the Drugs dataset. As shown in Table 7, MolField exhibits smaller degradation in atom-level and molecule-level stability under domain shift compared to prior approaches, although from a lower absolute level. This behavior suggests that the learned function-space representation captures structural regularities that are transferable across molecular domains, rather than being tightly coupled to dataset-specific graph statistics. Such transferability is consistent with the view of molecules as continuous functional objects and further supports the generality of the proposed formulation.

G.2 Full Distribution Analysis

Beyond aggregate metrics, we analyze the relationship between function-space reconstruction quality and downstream prediction performance at the level of the full data distribution. Figure 6 visualizes the correlation between the INR reconstruction loss and property prediction error (MAE) for multiple molecular properties, under both training-from-scratch and pretrain-finetune settings. Across all properties, we observe strong positive correlations between reconstruction loss and prediction error, as

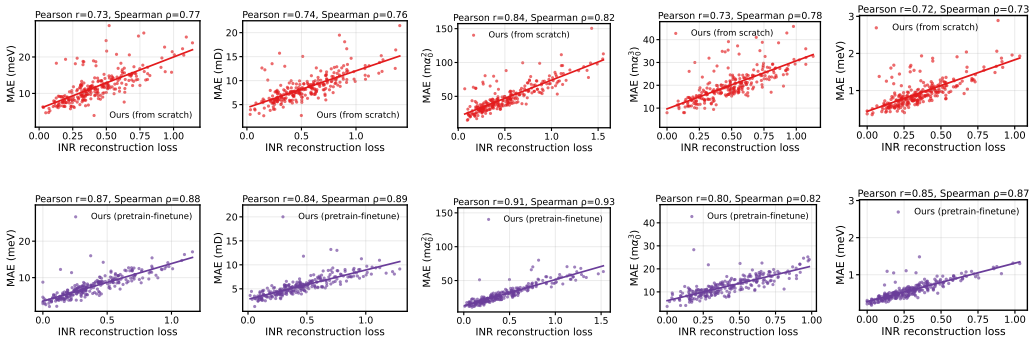

Figure 6: **Additional correlation results between INR reconstruction loss and property prediction error (MAE) for MolField.** Each column corresponds to a molecular property, with the top row showing models trained from scratch and the bottom row showing models with pretraining followed by finetuning. Each point denotes one molecule. Pearson and Spearman correlation coefficients are reported in each subfigure. Consistent positive correlations across properties indicate that more accurate function-space reconstructions are associated with improved property prediction, and that pretraining further reduces both reconstruction loss and prediction error.

quantified by both Pearson and Spearman coefficients. These indicate that improvements in function reconstruction accuracy translate continuously into lower prediction error.

Pretraining the INR further strengthens this relationship. As shown in Figure 6 (bottom row), the pretrain-finetune regime shifts the joint distribution toward lower reconstruction loss and lower prediction error, while preserving the overall monotonic trend. This behavior indicates that pretraining improves the quality of the learned function family rather than altering the mechanism by which downstream tasks are solved.

More broadly, this distribution-level dependence suggests that downstream prediction difficulty is related to the intrinsic complexity of the underlying molecular function. As the implicit representation more faithfully captures the continuous structure of a molecule, multiple physical properties become simultaneously easier to predict. These results are correlational: they indicate an association between reconstruction fidelity and prediction error, and we do not claim a causal role for either quantity.

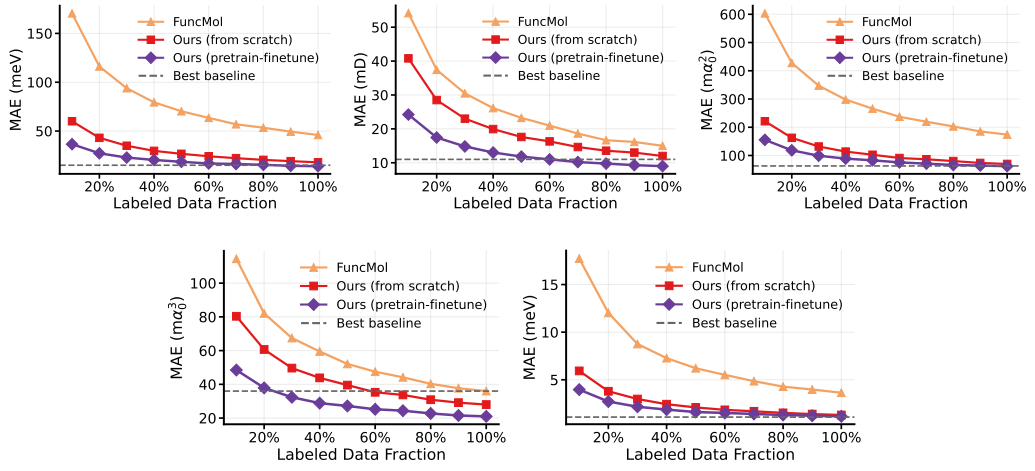

Figure 7: **Property prediction results with different labeled data ratio.** MolField consistently outperforms the primary baseline FuncMol across all label ratios and properties. Pretraining followed by finetuning further improves sample efficiency, especially in low-label regimes. The dashed line indicates the best-performing baseline with full data among prior methods for each property.

G.3 Low data analysis

We evaluate property prediction under varying labeled data fractions to assess data efficiency and robustness. As shown in Figure 7, MolField consistently achieves lower MAE than the primary baseline FuncMol across all labeled-data ratios and all evaluated molecular properties. This difference can be attributed to the distinct inductive biases of the two approaches. While FuncMol is designed to model instance-specific molecular configurations and excels at generation, molecular property prediction primarily depends on capturing structural regularities that are shared across molecules. Rather than requiring exact reconstruction of individual atomic configurations, property prediction benefits from representations that emphasize common geometric and physical patterns, such as chemical bonding structures and relational atomic arrangements. By learning molecular representations in function space via a hyper-network, MolField prioritizes these shared structural characteristics, leading to consistently improved prediction accuracy.

The dashed lines in Figure 7 indicate the best-performing prior baseline for each property. Compared against this reference, MolField demonstrates competitive or superior data efficiency on several properties. For example, on polarizability (α) MolField reaches the dashed reference using a fraction of the labels.

Figure 7 further compares two training strategies of MolField: training from scratch and pretrain-finetune. Across all label ratios, pretrain-finetune consistently outperforms training from scratch, with the largest gains appearing in the low-label regime. Since pretraining is performed without access to target property labels, these improvements suggest that the model learns regularities of molecular structure that reduce the amount of labeled supervision required for downstream prediction.

Taken together, these results indicate that MolField learns more than a task-specific predictor. By capturing structural patterns intrinsic to molecular systems, it yields improved predictive performance and stronger sample efficiency, particularly in low-label regimes.

G.4 Comparison with a Latent-as-Input INR

A continuous function over space can also be obtained without generating parameters, by concatenating a molecular latent to the query coordinate and decoding both with a single network shared across molecules, $f_{\boldsymbol{\Theta}}([c(\mathbf{x}); z])$, which is the conditioning used by DeepSDF-style models (Park et al., 2019). This construction defines a molecular function in the same formal sense that MolField does, so whether it also yields an equally effective *representation* is an empirical question rather than a definitional one. We therefore built it under matched settings: the same equivariant encoder, the same MLP architecture, the same canonical coordinate transform, and the same SDF and eikonal supervision. The only difference is where the molecular information enters. In the variant, z enters through the input channel and one parameter set serves every molecule and every time step; in MolField, $\boldsymbol{\Theta} = G_\phi(z)$ is generated per conditioning state, so each state instantiates its own field.

Table 8: **Latent-as-input INR versus MolField** on two systems of Table 1, under a matched encoder, MLP architecture, canonical coordinate transform, and SDF and eikonal supervision.

System	Model	IoU ↑	CD ↓	NC ↑
I-FABP	Latent-as-input INR	0.8895	0.0006	0.9311
	MolField	**0.9392**	**0.0002**	**0.9721**
FRODA	Latent-as-input INR	0.8254	0.0007	0.9134
	MolField	**0.8687**	**0.0006**	**0.9414**

The variant is not a weak comparison. It performs at the level of published methods on both systems, 0.8895 IoU against DSR's 0.8907 on I-FABP and 0.8254 against CanFields' 0.8283 on FRODA (Table 1), so the gap is not an artifact of an under-trained baseline, and it is trained on the same objective, so the gap does not originate in the supervision either. It does hold fewer parameters, since generating weights is what the FSHN does and this construction has none; we note this for transparency, but a variant that matches published methods is not failing for want of capacity. The two constructions also differ in what they require as input. In the variant the function for a given state is defined only once the coordinates of that state are available, because z is computed from them. In

MolField the conditioning variable for dynamics is the molecule together with a time step, so the field at time t is generated without X_t, which is what extended-horizon prediction requires since X_t is the quantity being predicted.

G.5 Force Prediction on MD17

Property prediction in the main text is evaluated on QM9, whose geometries are relaxed to equilibrium. To test the same read-out on conformations that are not, we evaluate force prediction on MD17 (Chmiela et al., 2017), where conformations are sampled from molecular dynamics trajectories while the task remains regression and therefore stays comparable to our QM9 setting. Under the standard split we report the mean force MAE across all molecules.

Table 9: **Force prediction on MD17**, mean MAE across all molecules under the standard split. Baseline values are those reported in the respective papers.

Method	Force MAE (kcal/mol/Å) ↓
SchNet (Schütt et al., 2017)	0.717
DimeNet (Gasteiger et al., 2020b)	0.306
PaiNN (Schütt et al., 2021)	0.178
MolField	0.175
Equiformer (Liao & Smidt, 2022)	0.172
TorchMD-NET (Thölke & De Fabritiis, 2022)	0.119
NequIP (Batzner et al., 2022)	**0.106**

MolField reaches the level of Equiformer and improves on PaiNN, but does not match TorchMD-NET or NequIP. MD17 molecules contain 9 to 21 atoms and forces are a local quantity, which is the regime where equivariant message passing is strongest and where a continuous field representation has least to offer. This is consistent with the pattern on QM9 in Table 2, where the gains concentrate on targets governed by molecule-wide spatial structure (μ, R^2, α) and are comparable on the more local ones ($\varepsilon_{\mathrm{HOMO}}$, ZPVE). We report the result in these terms rather than selecting a benchmark that favors the method.

G.6 Empirical Stability of the Canonical Frame

Section 4.1 establishes that $Q(X)$ is not uniquely determined on molecules whose point group is neither C_1 nor C_s, and that this is a consequence of equivariance rather than of our particular construction. That argument says nothing about how much the ambiguity matters in practice, so we measure it.

Protocol. We rank the QM9 test set by $\|\mathbf{u}_1\|$ under the trained model and compare the most degenerate 5% against the remaining 95%. We use this ranking rather than a grouping by symmetry class because it also captures near-symmetry, linear molecules, and accidental cancellation in molecules with no symmetry at all. Two perturbations are applied to each input: a random rigid rotation, which tests invariance, and a 0.01 Å coordinate perturbation, the scale of thermal motion, which tests stability. Deviations are reported relative to each group's own MAE, because the dipole moment of a symmetric molecule is close to zero and a measure relative to the true value would be dominated by that. We report μ and α: μ is the most orientation-sensitive target, so a frame error would surface there first, while α is isotropic and does not collapse on symmetric molecules, which makes it a control against a result that looks stable for the wrong reason.

What the measurements show. Predictions are invariant to rotation in practice: even on the most degenerate 5%, a rotation changes the prediction by about 10^{-4} of the model's own error on those molecules. The diagnostic is not insensitive to the effect it is meant to detect, since within that group the rotational deviation is two orders of magnitude larger for μ than for α, which is the asymmetry expected if the residual originates in frame orientation; its magnitude is nonetheless negligible. Under coordinate perturbation the degenerate group degrades by $1.6\times$ and $1.5\times$ relative to the rest for μ and α respectively. Had frame switching been the cause, the increase would have appeared for μ and

Table 10: **Sensitivity of QM9 predictions to the canonical frame.** Change in prediction relative to each group's own MAE, for the most degenerate 5% of the test set ranked by $\|\mathbf{u}_1\|$ and for the remaining 95%.

Target	Random rotation (95% / worst 5%)	0.01 Å perturbation (95% / worst 5%)
μ	10^{-7} / 10^{-4}	5% / 8%
α	10^{-7} / 10^{-6}	2% / 3%

not for α; that it is nearly the same for both points to the geometry of these molecules, which are smaller and more compact, rather than to $Q(X)$.

What the measurements do not establish. The diagnostic probes the property read-out, which consumes pooled weight tokens rather than field values at particular query points, so it bounds the effect of frame ambiguity on molecule-level regression and not on the field itself; a field-level diagnostic comparing $f_{\boldsymbol{\Theta}}$ before and after rotation at matched spatial locations would test the representation more directly. The ranking by $\|\mathbf{u}_1\|$ also does not isolate configurations in which $\mathbf{u}_1$ and $\mathbf{u}_2$ are nearly collinear while both have normal magnitude, which is the second way the construction degenerates. Finally, a 0.01 Å displacement is small relative to the reorientation a frame flip would produce, so this test bounds local sensitivity rather than the cost of a discontinuous switch. The claim we make is correspondingly narrow: on QM9 the non-uniqueness of the frame does not measurably affect predictions, not that it cannot affect predictions in general.

H Impact Statements

This work introduces MolField, a function-space formulation of molecular representation learning that models molecules as continuous fields rather than discrete structures. By shifting the representation object from task-specific embeddings to shared molecular functions, our framework aims to improve generalization, robustness, and transferability across molecular learning tasks. As molecular machine learning increasingly influences scientific discovery and engineering practice, it is important to consider the broader implications of such representational advances.

Potential Benefits. MolField may benefit multiple scientific and engineering domains. In computational chemistry (Jensen, 2017) and drug discovery (Berdigaliyev & Aljofan, 2020), more stable and transferable molecular representations could reduce data requirements, improve long-horizon simulation accuracy, and enable more reliable property prediction and molecular design. In materials science (Wei et al., 2019), function-space representations may facilitate modeling complex spatially continuous phenomena such as charge distributions or reaction fields. More broadly, our formulation contributes to scientific machine learning by providing a principled interface between machine learning models and continuous physical systems, potentially enabling better integration of data-driven and physics-based approaches.

Risks and Misuse Considerations. As with many advances in molecular modeling, improved representation learning may lower barriers to molecular design and optimization. In adversarial settings, such capabilities could be misused to assist in designing harmful chemical compounds or to accelerate unsafe experimentation. While MolField itself does not introduce new generative objectives beyond existing molecular modeling frameworks, its improved generalization and robustness could amplify the effectiveness of downstream applications. We therefore emphasize that responsible deployment should follow existing best practices in chemical safety, regulatory compliance, and dual-use risk assessment, and that access to high-capacity molecular design systems should be carefully managed.

Limitations and Scope. MolField focuses on improving molecular representations and does not directly address questions of chemical validity, synthetic feasibility, or biological safety of generated molecules. Its benefits depend on the quality and diversity of available training data and do not substitute for experimental validation or domain expertise. MolField is a data-driven framework that learns continuous geometric fields from molecular data; it enforces geometric consistency

(SE(3) invariance, field-level supervision) but does not impose first-principles constraints such as energy conservation, Hamiltonian structure, or wavefunction antisymmetry (Merz Jr, 2014). We therefore position MolField as a geometry-level representation that complements—rather than replaces—physics-driven approaches (e.g., DFT, wavefunction methods). These limitations constrain the scope of its applicability and should be considered when interpreting its results.

Future Directions. Future work may explore integrating function-space representations with physically grounded simulation frameworks, such as neural operators or differentiable solvers, to better capture complex molecular processes. Extending the approach to larger biomolecular systems, reactive chemistry, or multimodal molecular data could further broaden its applicability. From a societal perspective, developing technical safeguards, dataset governance strategies, and evaluation protocols for dual-use molecular modeling systems will be important to ensure that advances in representation learning are deployed responsibly and for beneficial scientific purposes.