ATOM-JEPA: JOINT-EMBEDDING PREDICTIVE ARCHITECTURE FOR 3D ATOMISTIC SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale self-supervised pretraining has reshaped modern machine learning, substantially advancing the ability of language and vision models to generalize across downstream tasks. While deep learning has driven considerable progress in modeling atomistic systems in recent years, self-supervised pretraining in this domain has not yet achieved comparable downstream generalization. To address this, we introduce Atom-JEPA, a self-supervised pretraining framework that learns latent representations from unlabeled 3D structures through complementary atom-level and substructure-level objectives inspired by joint-embedding predictive architectures. We pretrain Atom-JEPA on large-scale molecular and crystalline datasets and evaluate its transfer performance by fine-tuning on a diverse set of downstream property prediction tasks. Atom-JEPA achieves state-of-the-art performance on molecular ADMET and quantum-chemical property prediction tasks, and is highly competitive in predicting the physical properties of crystalline materials. These results demonstrate the potential of latent-space predictive pretraining to support broad downstream generalization from structural data alone.

1 INTRODUCTION

In recent years, machine learning has been adopted at an accelerating rate in molecular and materials discovery. Generative models enable inverse design through candidate generation (Hoogeboom et al., 2022; Cornet et al., 2024; Zeni et al., 2025), structure prediction (Abramson et al., 2024; Wohlwend et al., 2025; Chai Discovery Team et al., 2025) and property-guided optimization, complementing traditional workflows reliant on expert-guided design, computational screening, and experimental structure determination. Machine-learning interatomic-potentials (MLIPs) (Schütt et al., 2021; Batatia et al., 2023; Rhodes et al., 2025; Fu et al., 2025; Wood et al., 2026; Kavanagh et al., 2026) serve as surrogates for electronic structure methods such as DFT, while predictive models are increasingly used to screen promising candidates across a wide range of properties (Kellenberger et al., 2007; Zhuang et al., 2014; Pillai et al., 2023; Wong et al., 2024; Sun et al., 2024; Bai et al., 2025) before running costly computational or experimental evaluation. Historically, many of these approaches have relied on tailored models trained from scratch on individual datasets, often using molecular fingerprints (Rogers & Hahn, 2010) or string-based representations (Weininger, 1988). Recently, increasing computational resources and growing availability of structural data has enabled the training of deep neural networks directly on 3D atomistic structures (Schütt et al., 2017; Gilmer et al., 2017), offering the potential to learn representations that capture chemically relevant information directly from atomistic structure. However, a central question that remains unanswered is how to effectively leverage *massive unlabelled* datasets to improve performance on *sparsely labeled* downstream tasks, where less complex approaches such as simple fingerprint baselines often remain competitive.

Foundation models, i.e. neural networks pretrained on large-scale datasets and subsequently adapted to downstream tasks, have been central to recent advances in deep learning. Many modern computer vision models (Siméoni et al., 2025), large-language models (Chiang et al., 2022), and tabular data models (Hollmann et al., 2022; Qu et al., 2025) increasingly rely on large-scale pretraining. In machine learning for atomistic systems, there is a similar growing interest in developing foundation models (Pyzer-Knapp et al., 2025; Yuan et al., 2025) that can exhibit strong generalization across different chemical systems and tasks. Existing approaches either pretrain via supervised regression on quantum-chemical properties such as energies, forces, stress, and electronic characteristics, or through self-supervised structural objectives such as denoising, masking, and environment prediction.

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

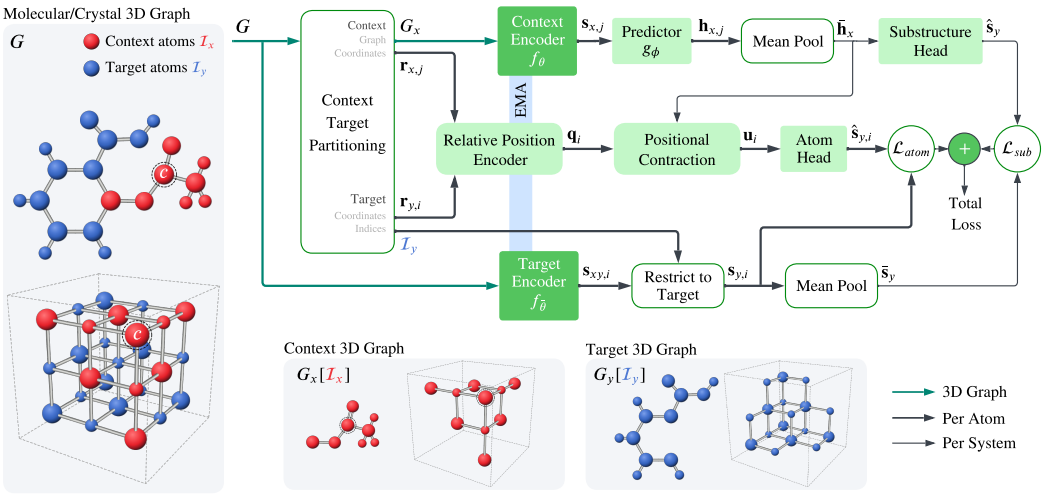


Figure 1: **Overview of Atom-JEPA.** Molecular and crystalline structures are partitioned into complementary context (red) and target (blue) subgraphs by sampling a k-hop neighborhood around an anchor atom. A 3D equivariant graph neural network encodes the context, which is then used to predict target representations at two scales: individual atoms and mean-pooled substructures. Target representations are extracted from the full structure after using a target encoder updated as an exponential moving average of the context encoder. Pretraining minimizes prediction errors in latent space, with the two subgraphs alternating as context and target. The pretrained context encoder is subsequently fine-tuned for downstream property prediction.

Based on the success of self-supervised learning (SSL) in computer vision and language models, we posit that SSL is the way forward for strong foundation models for atomistic systems as well.

In this work, we propose and demonstrate Atom-JEPA, a performant adaptation of the joint-embedding predictive architecture framework (LeCun, 2022) to 3D atomistic systems. Specifically, Atom-JEPA combines complementary atom-level and substructure-level predictive objectives in latent space, using only 3D positions and atom types. In contrast, existing approaches based on 3D atomistic structure primarily formulate self-supervision through prediction or reconstruction in the input space, combine 3D information with other modalities, or employ latent-space predictive objectives only at the substructure level. For an overview of Atom-JEPA, we refer to Fig. 1. We train Atom-JEPA models on both molecular and crystalline structures, analyse their learned embedding spaces, and finally evaluate their performance on supervised downstream tasks spanning ADMET, quantum-chemical, and crystalline material properties. To summarise:

1. We propose Atom-JEPA, to the best of our knowledge, the first JEPA-style loss that operates both on the atom- and substructure level of 3D atomistic systems alone.
2. We show that Atom-JEPA outperforms most existing open-source models and methods across ADMET, quantum-chemical and crystalline material property predictions.
3. We analyse the learned embeddings of Atom-JEPA trained models, illustrating where they do (and do not) separate atomistic systems by higher-order structural and functional groups.

2 Related Work

Supervised pretraining Supervised pretraining trains a neural network on a large *labeled* dataset, before finetuning or adapting it to downstream tasks (Shoghi et al., 2024; Ivković et al., 2024; Kong et al., 2025; Kim et al., 2026). For atomistic systems, models are typically pretrained on energies and forces computed using density functional theory (DFT), and while these labels provide rich supervision about the underlying potential-energy surface, this pretraining objective may not capture all information relevant to diverse downstream properties. Another limitation is scalability: generating

high-quality labels for molecules or crystals is computationally expensive. OMol25, for example, contains labels for 150 million atomistic systems at the ωB97M-V/def2-TZVPD level of theory and required approximately 6.6 *billion* CPU hours to generate (Levine et al., 2026). In contrast, SSL provides a route to exploiting structural data without requiring expensive label simulations.

Self-supervised pretraining Self-supervised learning as pretraining has also seen increasing attention in recent years. One of the earliest methods used VAEs, where the authors learn a latent representation of molecules by training a VAE on SMILES strings (Gómez-Bombarelli et al., 2018). Since then, the development has been wide and varied. Several works have done SSL on 2D graphs, using both masking and contrastive learning losses (Hu et al., 2020; Li et al., 2022; Wang et al., 2023; Xia et al., 2023; Xue et al., 2026) and more recently, similar losses have been proposed for 3D molecular graphs, with the expectation that features incorporating geometry should allow better downstream performance (Zhu et al., 2022; Zhou et al., 2023; Wu et al., 2025). Finally, a recent surge of papers has proposed using positional 3D denoising, sometimes in combination with masking losses, as a pretraining objective (Zaidi et al., 2023; Feng et al., 2023; Liu et al., 2025; Perez & Gomez-Bombarelli, 2026; Morehead et al., 2026).

Joint-Embedding Predictive Architecture Joint-embedding predictive architectures (JEPA) (LeCun, 2022) predict directly in *latent space* rather than in the input domain, and rely on generating *views* of data that teach the model to encode views of the same datapoints to be close in embedding space by using a predictor network. Graph-JEPA (Skenderi et al., 2025) divides 2D graphs into sub-graphs, and uses these sub-graphs as context and target views, but without a node-level objective. Polymer-JEPA (Piccoli et al., 2026) extends the Graph-JEPA framework to 2D polymers by changing the target encoder to encode the full polymer, while only computing the loss over a target subgraph, by pooling the embeddings of the nodes in that subgraph that are not contained within the context subgraph, again with no node-level objective, and on 2D graphs. Crys-JEPA (Liu et al., 2026) operates on sequences (created from the 3D crystal graph) using a Transformer and generates contexts by rotating and translating original target graphs, whilst using formation energy in the loss to produce energy-aware embeddings. C-Free (Ariguib et al., 2026) is most similar to our method, but with significant differences: C-Free operates on 2D and 3D graphs jointly and generates contexts via k-hops along covalent bonds, where the targets are the complements to the context subgraph. In comparison, we sidestep the need for multiple modalities by introducing an atom-level JEPA objective, and generating contexts via k-hops on a cutoff radius based graph. Most recently, Mol-JEPA (Rottach et al., 2026) is a framework in which representations from multiple pretrained models, computed descriptors, and experimental measurements define different molecular modalities, with masked modalities predicted in latent space from the remaining modalities. To summarise, our method is to the best of our knowledge, the first JEPA model for atomistic systems to learn atomic representations solely through latent-space prediction between 3D structural views.

3 METHOD

The core idea of Atom-JEPA is analogous to I-JEPA for images (Assran et al., 2023), but tailored to atomistic systems in 3D space: given part of a molecule or crystal (context view), predict the representation of the remaining part (target view) using context and target encoders coupled through an exponential moving average (EMA). We consider two prediction objectives: (1) an atom-level objective that predicts the representation of each atom in the target view, and (2) a substructure-level objective that predicts the mean-pooled representation of all target atoms. For the context encoder, target encoder, and predictor, we use an SE(3)-equivariant graph neural network (GNN).

We restrict our training data to structures that are close to local minima of the potential energy surface, as including off-equilibrium structures without information about their relative stability (e.g. total energy), could make the learning problem ambiguous. We discuss this choice further in Section A.4.

3.1 3D GRAPH REPRESENTATION AND PARTITIONING

We represent an atomistic system with N atoms by a 3D graph $G = (V, E)$. The node set $V = \{(z_i, \mathbf{r}_i)\}_{i=1}^{N}$ represents atoms by their species z_i and coordinates $\mathbf{r}_i \in \mathbb{R}^3$, while the edge set $E = \{(i, j) \mid i \neq j, \|\mathbf{r}_i - \mathbf{r}_j\| < r_c\}$ encodes spatial proximity defined by a cutoff radius r_c. For

crystalline systems, the edge set is constructed under periodic boundary conditions, including edges to periodic images within the cutoff radius.

Given an atomistic system, we partition atom indices into a k-hop EgoNet $\mathcal{I}_{\mathrm{a}}$ and its complement $\mathcal{I}_{\mathrm{b}}$: we sample an anchor atom $c \sim \mathrm{Unif}\{1, \ldots, N\}$ and a hop count $k \sim \mathrm{Unif}\,\mathcal{K}$ from a predefined set of hop counts $\mathcal{K}$, and define the EgoNet centered at c and its complement as

$$\mathcal{I}_{\mathrm{a}} = \{i \in \{1, \ldots, N\} \mid d(c, i) \leq k\}, \quad \mathcal{I}_{\mathrm{b}} = \{1, \ldots, N\} \setminus \mathcal{I}_{\mathrm{a}}, \tag{1}$$

where $d(c, i)$ denotes the shortest-path distance between atoms c and i, measured in number of edges. The corresponding induced subgraphs are $G_{\mathrm{a}} = G[\mathcal{I}_{\mathrm{a}}]$ and $G_{\mathrm{b}} = G[\mathcal{I}_{\mathrm{b}}]$ where

$$G[\mathcal{I}] = (V_{\mathcal{I}}, E_{\mathcal{I}}), \quad V_{\mathcal{I}} = \{(z_i, \mathbf{r}_i) \mid i \in \mathcal{I}\}, \quad E_{\mathcal{I}} = \{(i, j) \in E \mid i, j \in \mathcal{I}\}. \tag{2}$$

For each partition, we train in both prediction directions, defining the context and target graphs $(G_{\mathrm{x}}, G_{\mathrm{y}})$ as either $(G_{\mathrm{a}}, G_{\mathrm{b}})$ or $(G_{\mathrm{b}}, G_{\mathrm{a}})$.

3.2 Context, Target and Predictor Models

The context and target encoders, f_θ and $f_{\bar{\theta}}$ respectively, share the same 3D equivariant GNN architecture, taking 3D graphs as inputs and producing atom-level features. The parameters of the context encoder, θ, are updated via gradients, whereas the target encoder parameters, $\bar{\theta}$, are an exponential moving average of θ. The context encoder is only given context graph G_x, whilst the target encoder is given the full graph G; the JEPA loss is only computed over the target graph G_y. We label the context encoder output as $\mathbf{s}_x$ and the target encoder output on the G_y graph as $\mathbf{s}_y$.

Given the context representations $\mathbf{s}_x$, the predictor, g_ϕ, predicts representations produced by the target encoder. The predictor consists of a shallow 3D equivariant GNN backbone followed by objective-specific readout heads. We consider two prediction objectives: (1) a substructure objective, $\mathcal{L}_{\mathrm{sub}}$, that targets a global representation of the target subgraph, and (2) an atom-level objective, $\mathcal{L}_{\mathrm{atom}}$, that targets the representation of each target atom. We write the atom features produced by the predictor as $\mathbf{h}_x$ and the mean-pooled features as $\bar{\mathbf{h}}_x$.

3.3 Prediction Objectives & Atom-JEPA Loss

Substructure Objective In the substructure objective, the target is the mean-pooled invariant scalar features produced by the target encoder indexed over the target subgraph, whereas the prediction is constructed by passing the invariant component of the pooled context through a two-layer MLP mapping into the feature space of the target encoder,

$$\bar{\mathbf{s}}_{\mathrm{y}} = \frac{1}{|\mathcal{I}_{\mathrm{y}}|} \sum_{i \in \mathcal{I}_{\mathrm{y}}} \mathbf{s}_{\mathrm{y},i}^{(0)} \quad \text{and} \quad \hat{\bar{\mathbf{s}}}_{\mathrm{y}} = R_\phi^{\mathrm{sub}}(\bar{\mathbf{h}}_{\mathrm{x}}) = \mathrm{MLP}(\bar{\mathbf{h}}_{\mathrm{x}}^{(0)}), \tag{3}$$

where $(\cdot)^{(0)}$ denotes the zero-th degree (invariant) features alone.

Atom-level Objective The atom-level objective instead produces one prediction $\hat{\mathbf{s}}_{\mathrm{y},i}$ for every target atom $i \in \mathcal{I}_{\mathrm{y}}$, which is compared against the corresponding target encoder representation $\mathbf{s}_{\mathrm{y},i}$. In addition to the context features, the atom-level readout head is provided with an equivariant relative positional encoding $\mathbf{q}_i$ of the target atom's position relative to the context subgraph,

$$\hat{\mathbf{s}}_{\mathrm{y},i} = R_\phi^{\mathrm{atom}}(\bar{\mathbf{h}}_{\mathrm{x}}, \mathbf{q}_i) = \mathrm{MLP}(\mathbf{u}_i), \quad i \in \mathcal{I}_{\mathrm{y}}, \tag{4}$$

where $\mathbf{u}_i$ is constructed by computing the inner product between each feature channel in $\mathbf{q}_i$ and $\bar{\mathbf{h}}_{\mathrm{x}}$ and concatenating the individual inner products. We refer to Appendix Sections A.3.2 and A.3.3 for details on the construction of $\mathbf{u}_i$ and $\mathbf{q}_i$ respectively.

Atom-JEPA Loss We use mean squared error between predicted and target representations, $\ell(\mathbf{a}, \mathbf{b}) = \frac{1}{C}\|\mathbf{a} - \mathbf{b}\|_2^2$, where C is the feature dimension as the loss. The substructure and atom-level objectives are thus defined as

$$\mathcal{L}_{\mathrm{sub}}^{x \to y} = \frac{1}{B} \sum_{g=1}^{B} \ell\left(\hat{\bar{\mathbf{s}}}_{\mathrm{y}}^{(g)}, \bar{\mathbf{s}}_{\mathrm{y}}^{(g)}\right) \quad \text{and} \quad \mathcal{L}_{\mathrm{atom}}^{x \to y} = \frac{1}{\sum_{g=1}^{B} |\mathcal{I}_{\mathrm{y}}^{(g)}|} \sum_{g=1}^{B} \sum_{i \in \mathcal{I}_{\mathrm{y}}^{(g)}} \ell\left(\hat{\mathbf{s}}_{\mathrm{y},i}, \mathbf{s}_{\mathrm{y},i}\right), \tag{5}$$

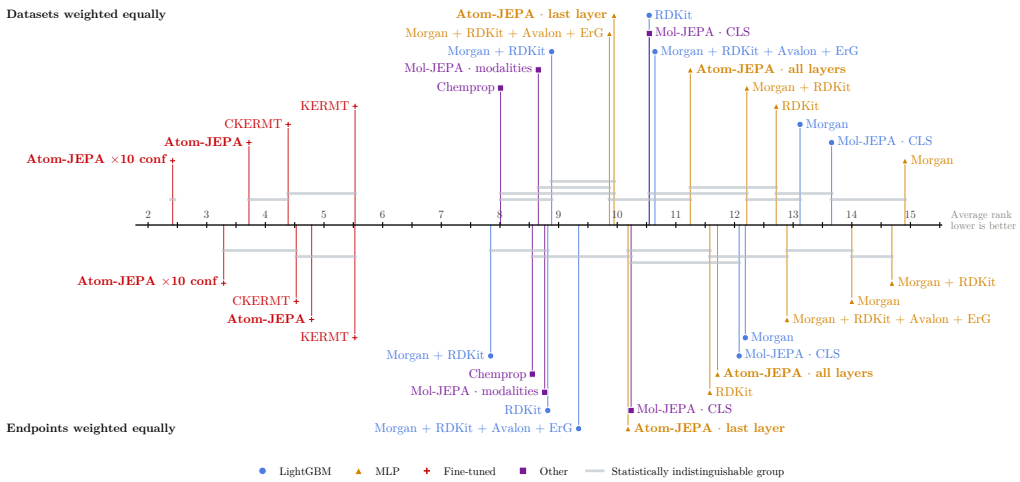


Figure 2: Average rank across 38 endpoints from three benchmarks (Biogen 4 endpoints, ChEMBL-MT 25 endpoints, ExpansionRx 9 endpoints). Top part weights each dataset equally, whereas bottom part weighs each endpoint equally. Bars join models whose paired rank difference could not be distinguished from zero by a stratified seed/fold bootstrap over endpoints (10^6 replicates, resampled within dataset, 95% BCa interval). Tables with individual results can be found in Section E.

where $x \to y$ denotes the prediction direction in which G_x is used as the context graph and G_y as the target graph and B is the batch size. The subgraph loss is a mean over the B graphs in a batch, whilst the atom-level loss is a mean over all the target atoms in a batch B. The final loss is then,

$$\mathcal{L}_{\text{Atom-JEPA}} = \lambda_{\text{atom}}\mathcal{L}_{\text{atom}} + \lambda_{\text{sub}}\mathcal{L}_{\text{sub}}, \tag{6}$$

where $\lambda_{\{\text{atom,sub}\}}$ control the weight of each objective.

4 Experiments

We use EquiformerV3 (Liao et al., 2026) as the SE(3) equivariant GNN architecture for the context encoder, target encoder, and predictor. For molecular pretraining, we use ∼19 million molecules from Uni-Mol (Zhou et al., 2023); for crystal pretraining, we use ∼1.7 million structures from Alexandria (Cavignac et al., 2026) with energies less than 50 meV/atom above the convex hull. Details about datasets and preprocessing are provided in App. C. Pretraining details are provided in App. G. After pretraining, we finetune the context encoder for downstream property prediction across both domains. For molecules, we consider ADMET prediction and quantum-chemical property prediction on QM9 (Ramakrishnan et al., 2014). For crystals, we evaluate materials-property prediction using the Matbench benchmark suite (Dunn et al., 2020). We further conduct ablation studies to assess the effects of pretraining, varying the pretraining prediction objective, and the scale and composition of the pretraining data (Section 4.2).

4.1 Experimental Results

Results on ADMET We extensively investigate the transfer learning for ADMET tasks on three different datasets, spanning various amounts of data, label sparsity, and splitting methods:

1. ChEMBL-MT (Adrian et al., 2025): Multi-task, multiple sources of data, 25 targets, Taylor-Butina clustering on Morgan fingerprints with pre-defined splits.
2. ExpansionRx (MacDermott-Opeskin & Castellanos, 2026): Multi-task, single source for collection, 10 targets, temporal split of training and test data.
3. Biogen ADME (Fang et al., 2023): Multi-task, internal source, 6 targets, Bemis–Murcko scaffold and Taylor-Butina clustering on Morgan fingerprints.

Table 1: **ADMET results.** For each endpoint, the model with the lowest mean error across folds is designated as *Best*, and an unpaired Tukey's test is used to identify additional models that are statistically indistinguishable (*Indist.*) from the best model at the 95% confidence level (see Appendix E.1). *Sum* denotes the total best or indistinguishable endpoints. * reports only the mean error across splits; these values are treated as constant in the statistical tests. † did not report Biogen results for two targets and is therefore excluded from the statistical tests for these targets.

	Biogen (6)		ChEMBL MT (25)		ExpansionRx (9)		
Model	**Best / Indist.**	**Sum**	**Best / Indist.**	**Sum**	**Best / Indist.**	**Sum**	**Total** (40)
Atom-JEPA ×10 conf	3 / 3	6	6 / 10	16	7 / 2	9	31
Atom-JEPA	0 / 6	6	0 / 15	15	0 / 9	9	30
CKERMT (Xue et al., 2026)	0 / 6	6	11 / 8	19	1 / 3	4	29
KERMT* (Adrian et al., 2025)	1 / 1†	2	3 / 9	12	0 / 1	1	15†
Chemprop (Graff et al., 2026)	1 / 3	4	0 / 12	12	0 / 1	1	17
Mol-JEPA · modalities (Rottach et al., 2026)	1 / 2	3	2 / 5	7	0 / 3	3	13
LightBGM Morgan + RDKit	0 / 2	2	3 / 9	12	1 / 2	3	17

For completeness, we also include results on TDC-ADMET (Huang et al., 2021) in Section E, but exclude them from the statistical analysis, due to data acquisition inconsistencies and a questionable leaderboard (Koleiev et al., 2026). As Atom-JEPA requires 3D structures, we use the Uni-Mol pipeline to generate conformers (described in Section H.1.2). To ensure a fair comparison, we tune the hyperparameters using the training data from a cluster split on Biogen from Adrian et al. (2025). We use the average validation MAE over endpoints as our selection criteria (details in Section H.1). We follow the transfer learning setup of multi-head multi-target training strategy used by CKERMT (Xue et al., 2026) with some minor changes (see Section H).

We present the aggregated results from the three main ADMET datasets in Fig. 2 and Table 1. Atom-JEPA reaches on par or significantly better performance than the next best model, CKERMT, and outperforms fingerprint-based models. Notably, Atom-JEPA has no explicitly coded expert chemical knowledge in comparison to all other models, including the self-supervised models of KERMT and CKERMT. This also explains why we find that Atom-JEPA's predictions are the least correlated with other models tested, see Appendix E.2. This is especially useful for the common ADMET practice of ensembling models. We additionally observe significantly improved prediction accuracy when averaging predictions over our 10 generated conformers (Atom-JEPA ×10 conf) for each molecule, suggesting that different conformers can provide different information to the prediction.

Interestingly, Atom-JEPA performs especially well on ExpansionRX, a dataset that differs notably from other benchmarks (Xue et al., 2026), including our pretraining data Uni-Mol, where only a single molecule shares the same scaffold (see Table 6). This suggests that the features learned by Atom-JEPA are more robust at transferring, compared to other models.

Results on QM9 QM9 consists of ∼130k small organic molecules with 12 quantum-chemical target properties. We use the random 110k/10k/11k split most commonly used in the literature; finetuning details are given in Section H.2. The same optimization protocol and hyperparameters are used for all targets, and all readout heads use sum aggregation over per-atom features.

Table 2 compares Atom-JEPA with previous models. 3D-EMGP, DenoiseVAE, TriForces and JMP-L use different random data partitions; their results are shown in grey for reference and excluded from highlighting. Atom-JEPA outperforms all SSL pretraining methods on 11/12 targets and ties on the remaining target C_v with SliDe. Overall, Atom-JEPA achieves the best results on $\varepsilon_{\text{HOMO}}$, $\varepsilon_{\text{LUMO}}$, μ, and ZPVE, and ties for best on α and C_v. It is best or within 5% of the best result on 10/12 targets. Relative to EquiformerV2, the predecessor of our EquiformerV3 backbone, Atom-JEPA reduces MAE on all 12 targets, with a mean relative reduction of 35%.

GotenNet-L, trained from scratch, remains the strongest method on several QM9 targets. Nevertheless, Atom-JEPA matches or outperforms it on 6/12 targets, and on the four thermodynamic energies (G, H, U, U_0) it trails by at most 5%, with a mean gap of 2.4%. We view GotenNet's strong performance as complementary to our contribution, as Atom-JEPA pretraining could, in principle, be applied to a GotenNet backbone. Beyond using a different random split with a smaller test set and correspondingly larger training set, JMP-L is not directly comparable as it is pretrained with supervised DFT labels aligned with several QM9 targets, on molecules that overlap with QM9. Even so, Atom-JEPA achieves lower error on α, μ, and R^2, three targets not aligned with JMP-L's supervised pretraining labels, suggesting stronger transfer.

Table 2: **QM9 results.** The evaluation metric is MAE (↓). **Bold:** best result in each column; underline: within 10% of the best. Grey rows with † use a different random train/validation/test split from 110k/10k/11k and are shown for reference only, excluded from highlighting. "–" denotes unreported targets.

Task Units	α ma_0^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ mD	C_ν $\frac{\text{cal}}{\text{mol K}}$	G meV	H meV	R^2 ma_0^2	U meV	U_0 meV	ZPVE meV
No pretraining												
MACE (Batatia et al., 2023)	38	42	22	19	15	21	5.5	4.7	210	4.1	4.1	1.23
Equiformer$_{V2}$ (Liao et al., 2024)	47	29.0	14.4	13.3	9.9	23	7.57	6.22	186	6.49	6.17	1.47
GotenNet$_L$ (Aykent & Xia, 2025)	**28**	**19.8**	13.4	12.2	6.7	**19**	**4.98**	**3.30**	**24**	**3.41**	**3.37**	1.08
Supervised pretraining												
Transformer-M (Luo et al., 2023)	41	27.4	17.5	16.2	37	22	9.63	9.39	75	9.41	9.37	1.18
JMP-L† (Shoghi et al., 2024)	32	19.1	8.8	8.6	8.0	17	4.3	2.8	163	2.8	2.9	0.9
Self-supervised pretraining only												
3D-EMGP† (Jiao et al., 2023)	57	37.1	21.3	18.2	20	26	9.30	8.70	92	8.60	8.60	1.38
GeoSSL-DDM (Liu et al., 2023)	46	40.2	23.5	19.4	15	24	7.65	7.09	122	6.99	6.92	1.31
Coord (Zaidi et al., 2023; Feng et al., 2023)	52	31.8	17.7	14.3	12	20	6.91	6.45	450	6.11	6.57	1.71
Frad (VRN) (Ni et al., 2024a)	42	27.7	17.9	13.8	11	21	6.03	6.01	354	5.35	5.41	1.63
Frad (RN) (Feng et al., 2023)	37	27.8	15.3	13.7	10	20	6.19	5.55	342	5.62	5.33	1.42
DenoiseVAE† (Liu et al., 2025)	65	26.0	14.2	11.9	7.9	15	5.35	4.19	62	4.03	4.31	1.03
SliDe (Ni et al., 2024b)	37	26.2	13.6	12.3	8.7	**19**	5.37	4.26	341	4.29	4.28	1.52
TriForces† (Ramlaoui et al., 2026)	75	38.8	20.2	20.4	18.0	31	–	–	–	–	8.9	–
Atom-JEPA	**28**	24.7	**12.4**	**11.0**	**6.1**	**19**	4.99	3.37	29	3.50	3.53	**1.05**

Results on Matbench Table 3 reports means and standard deviations across the five official folds for 8 Matbench tasks; training details are in Section H.3. Baseline results are taken from the official leaderboard for CGCNN, ALIGNN, and coGN; from the original papers for Crystal Twins (CT) and JMP-L; and from the Triforces paper for MACE, Orb, eSEN, and both TriForces variants.

Compared with models trained from scratch, Atom-JEPA achieves the lowest error on the four smaller tasks, with improvements of up to 21%. On the larger MP regression tasks, coGN remains stronger, although pretraining improves our encoder by 12–15% on these tasks relative to training from scratch (Section 4.2). Among pretrained models with at most 4M pretraining structures, Atom-JEPA achieves the lowest error on 6/8 tasks, and outperforms Crystal Twins on all tasks. This holds even though MACE, Orb, and eSEN are pretrained on 4M structures with energy and force labels, while Atom-JEPA is pretrained on 1.7M structures without any labels.

Comparison to TriForces (5M structures) highlights the importance of the backbone architecture: Atom-JEPA outperforms TriForces MACE on 7/8 tasks, whereas compared to TriForces eSEN it performs worse, with the exception of MP Is Metal (.908 versus .888 F1) and MP E Form, which is essentially tied (20.3 versus 20.2 meV/atom). JMP-L, pretrained with supervision on 120M structures, achieves lower MAE than Atom-JEPA on all seven regression tasks.

Table 3: **Matbench results.** The evaluation metric is MAE (↓) for all tasks except MP Is Metal, which uses F1 (↑). Values are means over five folds, with standard deviations where available. Pretrained models are ordered by increasing pretraining dataset size, shown in parentheses after each method. **Bold:** best result in each task. "–" denotes targets not reported by the original authors. * results from (Ramlaoui et al., 2026).

Tasks Units #Samples	Phonons cm^{-1} 1,265	Dielectric – 4,764	Log GVRH $\log_{10}$(GPa) 10,987	Log KVRH $\log_{10}$(GPa) 10,987	Perovskites meV 18,928	MP Gap eV 106,113	MP E Form meV/atom 132,752	MP Is Metal F1 106,113
No pretraining								
CGCNN (Xie & Grossman, 2018)	57.8±12.3	.599±.083	.090±.002	.071±.003	45.2±.7	.297±.004	33.7±.6	**.946**±.008
ALIGNN (Choudhary & DeCost, 2021)	29.5±2.1	.345±.087	.072±.001	.057±.003	28.8±.9	.186±.003	21.5±.5	.902±.002
coGN (Ruff et al., 2023)	29.7±2.0	.309±.086	.069±.001	.054±.003	26.9±.8	.156±.002	17.0±.3	.901±.003
With pretraining								
CT$_{\text{SimSiam}}$ (Magar et al., 2022) (0.43M)	48.9±7.7	.417±.079	.087±.003	.067±.003	42.0±1.0	.281±.025	37.0±.0	–
Atom-JEPA (1.7M)	23.3±4.5	.268±.074	.067±.001	.047±.002	29.1±1.0	.162±.002	20.3±.2	.908±.002
Orb* (Neumann et al., 2024) (4M)	26.2	.251	.063	.051	30.7	.194	21.1	.895
MACE* (Batatia et al., 2023)(4M)	36.7	.279	.082	.055	61.4	.370	40.8	.858
eSEN* (Fu et al., 2025) (4M)	57.8	.205	.093	.072	40.1	.392	83.5	.811
TriForces MACE (Ramlaoui et al., 2026) (5M)	27.6	**.159**	.073	.050	35.1	.250	34.4	.876
TriForces eSEN (Ramlaoui et al., 2026) (5M)	**19.5**	.232	**.058**	**.043**	**25.6**	.139	20.2	.888
JMP-L (Shoghi et al., 2024) (120M)	20.6	.249	.059	.045	26.0	**.091**	**10.1**	–

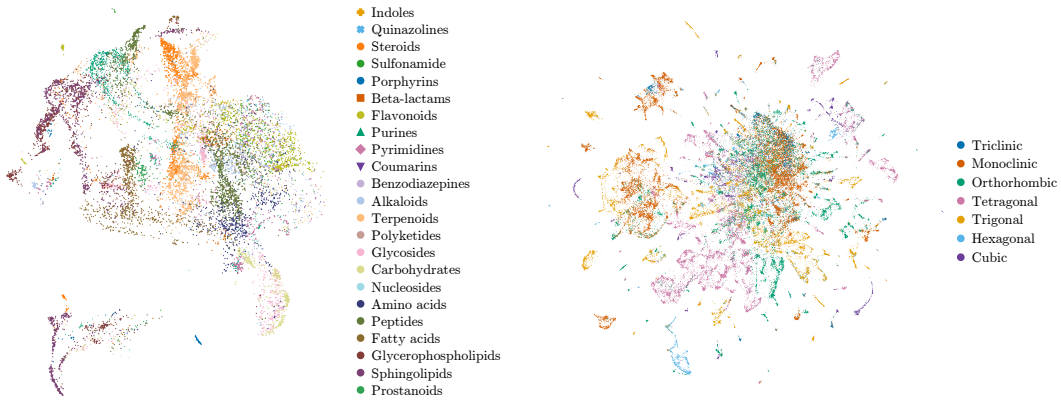


Figure 3: UMAP projections of Atom-JEPA representations for molecules annotated as belonging to one chemical family from the 32,252 molecules in the manually annotated three-star subset of ChEBI (Malik et al., 2026), colored by chemical family, and 50,000 randomly sampled crystals from the Alexandria pretraining dataset, colored by crystal system.

4.2 Ablation Studies

Effect of Atom-JEPA pretraining We compare the Atom-JEPA pretrained encoder to a randomly initialized counterpart on both ADMET and Matbench, using the training procedures described in Sections H.1 and H.3. Atom-JEPA pretraining improves downstream performance for both molecular and crystalline tasks (Table 4). On the molecular ADMET benchmarks, pretraining reduces error by 9.6% on Biogen and 8.5% on ExpansionRx. ChEMBL-MT does not see an improvement, but we suspect that confounded by hyperparameters not generalizing well from Biogen to the much larger dataset size. On Matbench, pretraining improves all eight tasks and all 40 task–fold combinations, yielding mean foldwise relative MAE reductions of 10.1–14.5% across the seven regression tasks and increasing mean classification F1 from 0.9010 to 0.9083. These gains span datasets ranging from 1,265 to 132,752 samples and persist on the largest regression benchmarks: MP Gap and MP E Form improve by 14.5% and 12.0%, respectively. Even without pretraining, the encoder is competitive with established from-scratch models, matching or exceeding their performance on several tasks (Table 3). This confirms that the gains from pretraining do not simply stem from a weak baseline.

Table 4: **Effect of Atom-JEPA pretraining.** The same encoder architecture is compared with and without pretraining on identical official folds; values are means and standard deviations across the five folds or endpoints for ADMET tasks. Δ (rel.) is the mean fold-wise or endpoint relative improvement (a reduction in MAE, or an increase in F1). "Impr." counts folds or endpoints on which pretraining wins; p is a two-sided paired t-test over the five Matbench folds or mean MAE over the folds in the 6/9 endpoints for ADMET tasks.

Task (Units)	#Samples	No pretraining	Atom-JEPA	Δ (rel.)	Impr..	p
ADMET datasets						
Biogen	3,521	.3825	**.3458**	9.6%	6/6	.011
ExpansionRX	7,608	.3673	**.3361**	8.5%	9/9	$<$ .001
ChEMBL-MT	114,112	**.4543**	.4567	$-$.5%	12/25	.57
Matbench						
Phonons (cm^{-1})	1,265	26.9$\pm$4.0	**23.3**$\pm$4.5	13.5%	5/5	.015
Dielectric (–)	4,764	.298$\pm$.084	**.268**$\pm$.074	10.1%	5/5	.011
Log GVRH ($\log_{10}$(GPa))	10,987	.0773$\pm$.0010	**.0670**$\pm$.0010	13.3%	5/5	$<$ .001
Log KVRH ($\log_{10}$(GPa))	10,987	.0539$\pm$.0034	**.0473**$\pm$.0021	12.3%	5/5	.001
Perovskites (meV)	18,928	32.5$\pm$.6	**29.1**$\pm$1.0	10.5%	5/5	$<$ .001
MP Gap (eV)	106,113	.1896$\pm$.0022	**.1622**$\pm$.0022	14.5%	5/5	$<$ .001
MP E Form (meV/atom)	132,752	23.03$\pm$.29	**20.26**$\pm$.19	12.0%	5/5	$<$ .001
MP Is Metal (F1)	106,113	.9010$\pm$.0021	**.9083**$\pm$.0024	.8%	5/5	$<$ .001

We hypothesize that these performance gains arise because Atom-JEPA pretraining encourages representations to organize according to high-level features that transfer well across downstream tasks in both domains. The UMAP visualizations in Fig. 3 suggest such organization by chemical family and crystal system, respectively, providing a qualitative indication consistent with this hypothesis. In contrast, an encoder trained from scratch starts from representations shaped by the inductive biases

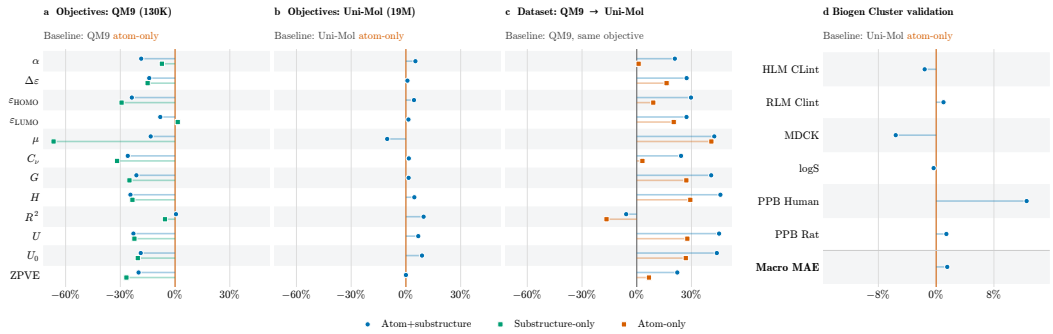


Figure 4: **Effect of pretraining objectives and dataset.** Frozen-feature MLP probes across 12 QM9 targets. The horizontal axis shows relative MAE reduction to the baseline of each panel. Positive values indicate improvement. **(a)** Joint (Atom+substructure) and substructure-only objectives relative to atom-only pretraining on QM9. **(b)** The joint objective relative to atom-only pretraining on Uni-Mol. **(c)** Uni-Mol pretraining relative to QM9 pretraining, with the objective held fixed. **(d)** The joint objective relative to atom-only pretraining on Uni-Mol on Biogen ADME validation objective.

of the backbone architecture, which primarily organize systems according to size and composition initially. We conduct a deeper analysis of the learned representations of Atom-JEPA in Section D.

Pretraining objective Next, we examine how the choice of pretraining prediction objective affects downstream performance. In addition to the joint-objective encoder pretrained on the 19M molecules from Uni-Mol, we pretrain an encoder using only the atom-level objective on the same data. We also pretrain three encoders on the 130K molecules from the QM9 dataset using atom-only, substructure-only, and joint objectives. All encoders are evaluated using all-layer MLP probes on frozen features, on the 12 QM9 properties. With QM9 pretraining, atom-only achieves the lowest MAE on 10/12 properties, while substructure-only performs best on $\varepsilon_{\mathrm{LUMO}}$ and the joint objective performs best on R^2 (Fig. 4a). With Uni-Mol pretraining, the joint objective improves over atom-only on 10/12 properties, ties on ZPVE, and performs worse only on μ (Fig. 4b). We additionally compare the two Uni-Mol-pretrained encoders on the six Biogen ADME tasks. The joint objective performs better on average, although atom-only remains competitive, performing better or similarly on 3/6 tasks (Fig. 4d). These results suggest that the benefit of combining atom-level and substructure-level objectives depends on both the pretraining dataset and the downstream task.

Data scaling Increasing the pretraining dataset from 130k (QM9) to 19M molecules (Uni-Mol) improves both atom-only and joint-objective pretraining on 11/12 QM9 properties, with R^2 as the sole exception (Fig. 4c). The gains are particularly pronounced for the joint objective, including MAE reductions of 27% for $\Delta\varepsilon$, 42.5% for μ, and 45.9% for H. Notably, Uni-Mol pretraining substantially improves performance even though QM9 pretraining already exactly matches the molecular domain of the downstream evaluation. While this comparison changes both dataset size and composition, the results suggest that scaling Atom-JEPA pretraining to larger and more chemically and structurally diverse datasets could further improve downstream performance.

5 Conclusion

In this work, we proposed Atom-JEPA, a self-supervised pretraining framework for 3D atomic systems that combines atom-level and substructure-level prediction in latent space. Using structural data alone without expensive property labels nor molecular attributes as the pretraining signal, Atom-JEPA learns representations that transfer effectively to molecular ADMET, quantum-chemical, and crystalline materials property prediction. Ablation studies demonstrate consistent improvement over training the same encoder from scratch, and indicate potential for further increasing downstream performance by scaling to larger and more diverse pretraining data. Overall, Atom-JEPA establishes latent-space prediction from 3D structure as a promising approach to learning transferable representations for molecules and materials, reducing reliance on property-labeled pretraining data.

AI USE STATEMENT

In this work, we used generative AI tools for sanity checking experimental setups and searching for additional related works. We have not used generative AI tools for: Generating synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results.

Additionally, we used AI for the creation and editing of software code and polishing writing. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 3 authors. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To facilitate full reproducibility, we report all hyperparameters used in our pretraining and downstream fine-tuning experiments. We will publicly release the source code, trained model parameters, and experimental results across all folds to support independent verification of our findings and comparison with future work.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Matthew Adrian, Yunsie Chung, Kevin Boyd, Saee Paliwal, Srimukh Prasad Veccham, and Alan C. Cheng. Multitask finetuning and acceleration of chemical pretrained models for small molecule drug property prediction. *CoRR*, abs/2510.12719, 2025. doi: 10.48550/ARXIV.2510.12719. URL https://doi.org/10.48550/arXiv.2510.12719.

Kumar K Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake Richards. \alpha-req : Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17626–17638. Curran Associates, Inc., 2022. doi: 10.52202/068431-1281. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/70596d70542c51c8d9b4e423f4bf2736-Paper-Conference.pdf.

Boshra Ariguib, Mathias Niepert, and Andrei Manolache. Learning the neighborhood: Contrast-free multimodal self-supervised molecular graph pretraining, 2026. URL https://arxiv.org/abs/2509.22468.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5wxCQDtbMo.

Ruilin Bai, Yu Yao, Qiaosong Lin, Lize Wu, Zhen Li, Huijuan Wang, Mingze Ma, Di Mu, Lingxiang Hu, Hai Yang, Weihan Li, Shaolong Zhu, Xiaojun Wu, Xianhong Rui, and Yan Yu. Preferable single-atom catalysts enabled by natural language processing for high energy density Na-S batteries. *Nature Communications*, 16(1):5827, 7 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-60931-x. URL https://www.nature.com/articles/s41467-025-60931-x.

Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields, 2023. URL https://arxiv.org/abs/2206.07697.

Théo Cavignac, Jonathan Schmidt, Pierre-Paul De Breuck, Antoine Loew, Tiago F. T. Cerqueira, Hai-Chen Wang, Anton Bochkarev, Yury Lysogorskiy, Aldo H. Romero, Ralf Drautz, Silvana Botti, and Miguel A. L. Marques. AI-driven expansion and application of the Alexandria database. *Journal of Physics: Materials*, 9(2):025014, 2026. doi: 10.1088/2515-7639/ae6620. URL https://doi.org/10.1088/2515-7639/ae6620.

Chai Discovery Team, Jacques Boitreaud, Jack Dent, Danny Geisz, Matthew McPartlon, Joshua Meier, Zhuoran Qiao, Alex Rogozhnikov, Nathan Rollins, Paul Wollenhaupt, and Kevin Wu. Zero-shot antibody design in a 24-well plate. *bioRxiv*, 2025. doi: 10.1101/2025.07.05.663018. URL https://www.biorxiv.org/content/early/2025/07/06/2025.07.05.663018.

Cheng-Han Chiang, Yung-Sung Chuang, and Hung-yi Lee. Recent advances in pre-trained language models: Why do they work and how do they work. In Miguel A. Alonso and Zhongyu Wei (eds.), *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pp. 8–15, Taipei, November 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.aacl-tutorials.2. URL https://aclanthology.org/2022.aacl-tutorials.2/.

Kamal Choudhary and Brian DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1), November 2021. ISSN 2057-3960. doi: 10.1038/s41524-021-00650-1. URL http://dx.doi.org/10.1038/s41524-021-00650-1.

François R J Cornet, Grigory Bartosh, Mikkel N. Schmidt, and Christian A. Naesseth. Equivariant neural diffusion for molecule generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=40pE5pFhWl.

Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1), 2020. ISSN 2057-3960. doi: 10.1038/s41524-020-00406-3. URL http://dx.doi.org/10.1038/s41524-020-00406-3.

Cheng Fang, Ye Wang, Richard Grater, Sudarshan Kapadnis, Cheryl Black, Patrick Trapa, and Simone Sciabola. Prospective validation of machine learning algorithms for absorption, distribution, metabolism, and excretion prediction: An industrial perspective. *Journal of Chemical Information and Modeling*, 63(11):3263–3274, 05 2023. ISSN 1549-9596. doi: 10.1021/acs.jcim.3c00160. URL https://doi.org/10.1021/acs.jcim.3c00160.

Shikun Feng, Yuyan Ni, Yanyan Lan, Zhi-Ming Ma, and Wei-Ying Ma. Fractional denoising for 3D molecular pre-training. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9938–9961. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/feng23c.html.

Xiang Fu, Brandon M. Wood, Luis Barroso-Luque, Daniel S. Levine, Meng Gao, Misko Dzamba, and C. Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction, 2025. URL https://arxiv.org/abs/2502.12147.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank, 2023. URL https://arxiv.org/abs/2210.02885.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1263–1272. JMLR.org, 2017.

David E Graff, Nathan K Morgan, Jackson W Burns, Anna C Doner, Brian Li, Shih-Cheng Li, Joel Manu, Angiras Menon, Hao-Wei Pang, Haoyang Wu, et al. Chemprop v2: An efficient, modular machine learning package for chemical property prediction. *Journal of Chemical Information and Modeling*, 66(1):28–33, 2026.

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, January 2018. ISSN 2374-7951. doi: 10.1021/acscentsci.7b00572. URL http://dx.doi.org/10.1021/acscentsci.7b00572.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations*, 2022. URL https://api.semanticscholar.org/CorpusID:252683429.

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d, 2022. URL https://arxiv.org/abs/2203.17003.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks, 2020. URL https://arxiv.org/abs/1905.12265.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf H. Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/4c56ff4ce4aaf9573aa5dff913df997a-Abstract-round1.html.

Žarko Ivković, Jesús Jover, and Jeremy Harvey. Transfer learning based on atomic feature extraction for the prediction of experimental 13c chemical shifts. *Digital Discovery*, 3(11):2242–2251, 11 2024. ISSN 2635-098X. doi: 10.1039/d4dd00168k. URL https://doi.org/10.1039/d4dd00168k.

Rui Jiao, Jiaqi Han, Wenbing Huang, Yu Rong, and Yang Liu. Energy-motivated equivariant pretraining for 3d molecular graphs. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i7.25978. URL https://doi.org/10.1609/aaai.v37i7.25978.

Seán R. Kavanagh, Chuin Wei Tan, Menghang Wang, Marc L. Descoteaux, Gabriel de Miranda Nascimento, Ulrik Unneberg, Laura Zichi, Francesco Libbi, Norma Rivano, Austin Glover, Vivek Bharadwaj, Anders Johansson, William C. Witt, Albert Musaelian, and Boris Kozinsky. Fast and accurate equivariant foundation models for atomistic simulation, 2026. URL https://arxiv.org/abs/2607.28461.

Esther Kellenberger, Jean-Yves Springael, Marc Parmentier, Muriel Hachet-Haas, Jean-Luc Galzi, and Didier Rognan. Identification of Nonpeptide CCR5 Receptor Agonists by Structure-based Virtual Screening. *Journal of Medicinal Chemistry*, 50(6):1294–1303, 3 2007. ISSN 0022-2623. doi: 10.1021/jm061389p. URL https://pubs.acs.org/doi/full/10.1021/jm061389p.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7482–7491. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00781. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.html.

So Yeon Kim, Yang Jeong Park, and Ju Li. Leveraging neural network interatomic potentials for a foundation model of chemistry. *npj Computational Materials*, 2026. doi: 10.1038/s41524-026-02167-x. URL https://doi.org/10.1038/s41524-026-02167-x.

Ihor Koleiev, Roman Stratiichuk, Nazar Shevchuk, Mykola Melnychenko, Alex Nyporko, Daniil Todoryshyn, Vladyslav Husak, Sergii Starosyla, Semen Yesylevskyy, and Alan Nafiiev. An end-user audit of reproducibility, data leakage, and overfitting of the top-ranked admet prediction models in tdc leaderboards. *Journal of Chemical Information and Modeling*, 2026.

Lingyu Kong, Nima Shoghi, Guoxiang Hu, Pan Li, and Victor Fung. Mattertune: an integrated, user-friendly platform for fine-tuning atomistic foundation models to accelerate materials simulation and discovery. *Digital Discovery*, 4(8):2253–2262, 08 2025. ISSN 2635-098X. doi: 10.1039/d5dd00154d. URL https://doi.org/10.1039/d5dd00154d.

Yann LeCun. A path towards autonomous machine intelligence. 2022. URL https://api.semanticscholar.org/CorpusID:251881108.

Daniel S. Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G. Taylor, Muhammad R. Hasyim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman, Nathan C. Frey, Xiang Fu, Vahe Gharakhanyan, Aditi S. Krishnapriyan, Joshua A. Rackers, Sanjeev Raja, Ammar Rizvi, Andrew S. Rosen, Zachary Ulissi, Santiago Vargas, C. Lawrence Zitnick, Samuel M. Blau, and Brandon M. Wood. The open molecules 2025 (omol25) dataset, evaluations, and models, 2026. URL https://arxiv.org/abs/2505.08762.

Han Li, Dan Zhao, and Jianyang Zeng. Kpgt: Knowledge-guided pre-training of graph transformer for molecular property prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 857–867. ACM, August 2022. doi: 10.1145/3534678.3539426. URL http://dx.doi.org/10.1145/3534678.3539426.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations, 2024. URL https://arxiv.org/abs/2306.12059.

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

Nian Liu, Nikita Kazeev, Stephen Gregory Dale, Artem Maevskiy, Yuwei Zeng, Ryoji Kubo, Pengru Huang, Thomas Laurent, Yann LeCun, Kostya S. Novoselov, and Xavier Bresson. Crys-jepa: Accelerating crystal discovery via embedding screening and generative refinement, 2026. URL https://arxiv.org/abs/2605.14759.

Shengchao Liu, Hongyu Guo, and Jian Tang. Molecular geometry pretraining with SE(3)-invariant denoising distance matching. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CjTHVo1dvR.

Yurou Liu, Jiahao Chen, Rui Jiao, Jiangmeng Li, Wenbing Huang, and Bing Su. Denoisevae: Learning molecule-adaptive noise distributions for denoising-based 3d molecular pre-training. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 22078–22101, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/37e9e62294ff6607f6f7c170cc993f2c-Paper-Conference.pdf.

Shengjie Luo, Tianlang Chen, Yixian Xu, Shuxin Zheng, Tie-Yan Liu, Liwei Wang, and Di He. One transformer can understand both 2d & 3d molecular data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vZTp1oPV3PC.

Hugo MacDermott-Opeskin and Maria Castellanos. Announcement 1: Expansionrx-openadmet blind challenge, August 2026. URL https://doi.org/10.5281/zenodo.21784075.

Rishikesh Magar, Yuyang Wang, and Amir Barati Farimani. Crystal Twins: self-supervised learning for crystalline material property prediction. *npj Computational Materials*, 8(1):231, 2022. doi: 10.1038/s41524-022-00921-5.

Adnan Malik, Muhammad Arsalan, Carlos Moreno, Juan Mosquera, Eloy Félix, Tevfik Kizilören, Venkatesh Muthukrishnan, Barbara Zdrazil, Andrew R Leach, and Noel M O'Boyle. Chebi: re-engineered for a sustainable future. *Nucleic Acids Research*, 54(D1):D1768–D1778, 01 2026. ISSN 1362-4962. doi: 10.1093/nar/gkaf1271. URL https://doi.org/10.1093/nar/gkaf1271.

Alex Morehead, Miruna Cretu, Antonia Panescu, Rishabh Anand, Maurice Weiler*, Tynan Perez, Samuel Blau, Steven Farrell, Wahid Bhimji, Anubhav Jain, Hrushikesh Sahasrabuddhe, Pietro Liò, Tommi Jaakkola, Rafael Gómez-Bombarelli, Rex Ying, Ben Erichson, and Michael Mahoney. Zatom-1: Towards a multimodal foundation model for 3d molecules and materials. *arXiv*, 2026. URL https://arxiv.org/abs/2602.22251. * denotes equal contribution.

Mark Neumann, James Gin, Benjamin Rhodes, Steven Bennett, Zhiyi Li, Hitarth Choubisa, Arthur Hussey, and Jonathan Godwin. Orb: A fast, scalable neural network potential, 2024. URL https://arxiv.org/abs/2410.22570.

Yuyan Ni, Shikun Feng, Xin Hong, Yuancheng Sun, Wei-Ying Ma, Zhi-Ming Ma, Qiwei Ye, and Yanyan Lan. Pre-training with fractional denoising to enhance molecular property prediction. *Nature Machine Intelligence*, 6(10):1169–1178, 2024a.

Yuyan Ni, Shikun Feng, Wei-Ying Ma, Zhi-Ming Ma, and Yanyan Lan. Sliced denoising: A physics-informed molecular pre-training method. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=liKkG1zcWq.

Tynan Perez and Rafael Gomez-Bombarelli. Self-conditioned denoising for atomistic representation learning, 2026. URL https://arxiv.org/abs/2603.17196.

Francesco Piccoli, Gabriel Vogel, and Jana M. Weber. Joint embedding predictive architecture for self-supervised pretraining on polymer molecular graphs. *Digital Discovery*, 5(2):819–834, 02 2026. ISSN 2635-098X. doi: 10.1039/d5dd00308c. URL https://doi.org/10.1039/d5dd00308c.

Hemanth Somarajan Pillai, Yi Li, Shih-Han Wang, Noushin Omidvar, Qingmin Mu, Luke E K Achenie, Frank Abild-Pedersen, Juan Yang, Gang Wu, and Hongliang Xin. Interpretable design of Ir-free trimetallic electrocatalysts for ammonia oxidation with graph neural networks. *Nature Communications*, 14(1):792, 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36322-5. URL https://doi.org/10.1038/s41467-023-36322-5.

Edward O. Pyzer-Knapp, Matteo Manica, Peter Staar, Lucas Morin, Patrick Ruch, Teodoro Laino, John R. Smith, and Alessandro Curioni. Foundation models for materials discovery –current state and future directions. *npj Computational Materials*, 11(1):61, 2025. doi: 10.1038/s41524-025-01538-0. URL https://doi.org/10.1038/s41524-025-01538-0.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=0VvD1PmNzM.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014. doi: 10.1038/sdata.2014.22.

Ali Ramlaoui, Alexandre Duval, Hannah Bull, Victor Schmidt, Hugues Talbot, Fragkiskos D. Malliaros, and Joseph Musielewicz. Triforces: Augmenting atomistic GNNs for transferable representations. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=dVgMUSGPN3.

Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale, 2025. URL https://arxiv.org/abs/2504.06231.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 05 2010. ISSN 1549-9596. doi: 10.1021/ci100050t. URL https://doi.org/10.1021/ci100050t.

Florian Rottach, Sebastian Schieferdecker, William Rudman, Randall Balestriero, and Carsten Eickhoff. Mol-jepa: A multimodal joint embedding predictive architecture for molecules, 2026. URL https://arxiv.org/abs/2608.22642.

Robin Ruff, Patrick Reiser, Jan Stühmer, and Pascal Friederich. Connectivity optimized nested graph networks for crystal structures, 2023. URL https://arxiv.org/abs/2302.14102.

K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller. Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 992–1002, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of chemical physics*, 148(24), 2018.

Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra, 2021. URL https://arxiv.org/abs/2102.03150.

Nima Shoghi, Adeesh Kolluru, John R. Kitchin, Zachary W. Ulissi, C. Lawrence Zitnick, and Brandon M. Wood. From molecules to materials: Pre-training large generalizable models for atomic property prediction, 2024. URL https://arxiv.org/abs/2310.16802.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Geri Skenderi, Hang Li, Jiliang Tang, and Marco Cristani. Graph-level representation learning with joint-embedding predictive architectures. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=v47f4DwYZb.

Jikai Sun, Rui Tu, Yuchun Xu, Hongyan Yang, Tie Yu, Dong Zhai, Xiuqin Ci, and Weiqiao Deng. Machine learning aided design of single-atom alloy catalysts for methane cracking. *Nature Communications*, 15(1):6036, 2024. URL https://www.nature.com/articles/s41467-024-50417-7.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Tamás Lajos Tompa, Eszter Varga-Umbrich, Ilyes Batatia, Alin M Elena, Noam Bernstein, and Gábor Csányi. Fine-tuning mlip foundation models: strategies for accuracy and transferability. *arXiv preprint arXiv:2606.12704*, 2026.

Zhen Wang, Zheng Feng, Yanjun Li, Bowen Li, Yongrui Wang, Chulin Sha, Min He, and Xiaolin Li. Batmannet: bi-branch masked graph transformer autoencoder for molecular representation. *Briefings in Bioinformatics*, 25(1), November 2023. ISSN 1477-4054. doi: 10.1093/bib/bbad400. URL http://dx.doi.org/10.1093/bib/bbad400.

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, 02 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005. URL https://doi.org/10.1021/ci00057a005.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1 democratizing biomolecular interaction modeling. *bioRxiv*, 2025. doi: 10.1101/2024.11.19.624167. URL https://www.biorxiv.org/content/early/2025/05/06/2024.11.19.624167.

Felix Wong, Erica J. Zheng, Jacqueline A. Valeri, Nina M. Donghia, Melis N. Anahtar, Satotaka Omori, Alicia Li, Andres Cubillos-Ruiz, Aarti Krishnan, Wengong Jin, Abigail L. Manson, Jens Friedrichs, Ralf Helbig, Behnoush Hajian, Dawid K. Fiejtek, Florence F. Wagner, Holly H. Soutter, Ashlee M. Earl, Jonathan M. Stokes, Lars D. Renner, and James J. Collins. Discovery of a structural class of antibiotics with explainable deep learning. *Nat.*, 626(7997):177–185, 2024. doi: 10.1038/S41586-023-06887-8. URL https://doi.org/10.1038/s41586-023-06887-8.

Brandon Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John Kitchin, Daniel Levine, et al. Uma: A family of universal models for atoms, 2026.

Chang Wu, Zhiyuan Liu, Wen Shu, Liang Wang, Yanchen Luo, Wenqiang Lei, Yatao Bian, Junfeng Fang, and Xiang Wang. 3d-GSRD: 3d molecular graph auto-encoder with selective re-mask decoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=PZ7YLONKiI.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning, 2018. URL https://arxiv.org/abs/1703.00564.

Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z. Li. Mole-BERT: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jevY-DtiZTR.

Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters*, 120(14), April 2018. ISSN 1079-7114. doi: 10.1103/physrevlett.120.145301. URL http://dx.doi.org/10.1103/PhysRevLett.120.145301.

Yifan Xue, Srimukh Prasad Veccham, Saee Paliwal, Tyler Shimko, and Micha Livne. Probabilistic contrastive pretraining for multi-task ADME property prediction. *CoRR*, abs/2606.11508, 2026. doi: 10.48550/ARXIV.2606.11508. URL https://doi.org/10.48550/arXiv.2606.11508.

LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International conference on machine learning*, pp. 2825–2834. PMLR, 2018.

Eric C. Y. Yuan, Yunsheng Liu, Junmin Chen, Peichen Zhong, Sanjeev Raja, Tobias Kreiman, Santiago Vargas, Wenbin Xu, Martin Head-Gordon, Chao Yang, Samuel M. Blau, Bingqing Cheng, Aditi Krishnapriyan, and Teresa Head-Gordon. Foundation models for atomistic simulation of chemistry and materials, 2025. URL https://arxiv.org/abs/2503.10538.

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. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tYIMtogyee.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake

Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025. doi: 10.1038/s41586-025-08628-5. URL https://doi.org/10.1038/s41586-025-08628-5.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6K2RM6wVqKu.

Jinhua Zhu, Yingce Xia, Lijun Wu, Shufang Xie, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Unified 2d and 3d pre-training of molecular representations. In *KDD 2022*, July 2022. URL https://www.microsoft.com/en-us/research/publication/unified-2d-and-3d-pre-training-of-molecular-representations/.

Chunlin Zhuang, Sreekanth Narayanapillai, Wannian Zhang, Yuk Yin Sham, and Chengguo Xing. Rapid Identification of Keap1–Nrf2 Small-Molecule Inhibitors through Structure-Based Virtual Screening and Hit-Based Substructure Search. *Journal of Medicinal Chemistry*, 57(3):1121–1126, 2 2014. ISSN 0022-2623. doi: 10.1021/jm4017174. URL https://pubs.acs.org/doi/full/10.1021/jm4017174.

APPENDIX CONTENTS

A Detailed Methodology **19**

A.1 Atomic 3D Graph Representation and Partitioning 19

A.2 Context and Target Encoders 20

A.3 Prediction 20

A.4 Data Curation 22

B Limitations **22**

C Datasets **23**

C.1 Structural Overlap 23

D Analysis of Learned Representations **26**

D.1 Structure and Property Organization in the Embedding Space 26

D.2 Effect of Pretraining on Molecular Representations 35

E ADMET Full Results **38**

E.1 Statistical Tests Setup 38

E.2 Model Prediction Correlation 38

E.3 Biogen ADME 39

E.4 ExpansionRX and ChEMBL-MT 39

E.5 TDC ADMET 42

F Effect of Pretraining Objectives and Data Scale Full Results **45**

G Pretraining Implementation Details **45**

G.1 Hyperparameters 45

G.2 Representation Collapse Metrics 45

H Finetuning Implementation Details **48**

H.1 ADMET Finetuning 48

H.2 QM9 Finetuning 52

H.3 Matbench Finetuning 53

A Detailed Methodology

The overall idea of Atom-JEPA is analogous to I-JEPA for images (Assran et al., 2023), but redesigned for atomic systems in 3-dimensional space: Given part of a molecule or crystal (context view), predict the representation of the remaining part (target view). We consider two prediction objectives: (1) an atom-level objective that predicts the feature representation of each atom in the target view, and (2) a substructure-level objective that predicts the mean-pooled representation of all target atoms. For the context encoder, target encoder and predictor, we use a 3D geometric graph neural network (GNN). We only use data that are either in ground-state or close to local minima of the potential energy surface. We posit that using data in off-equilibrium states (either rattled systems or from MD trajectories) without explicitly telling the model something about the stability of the system (e.g. via total energy of the system) could make the learning problem much more ambigious.

A.1 Atomic 3D Graph Representation and Partitioning

We represent atomic systems as 3D graphs and partition each system into complementary context and target subgraphs. An atomic system containing N atoms can be represented by a 3D graph $G = (V, E)$ with the node set $V = \{(z_i, \mathbf{r}_i)\}_{i=1}^{N}$, where z_i denotes the atom type of atom i and $\mathbf{r}_i \in \mathbb{R}^3$ its 3D position. The edge set E encodes spatial proximity between atoms within a cutoff radius r_c. Given a partition of the atomic indices into a context set $\mathcal{I}_x$ and a target set $\mathcal{I}_y$, we define the corresponding induced subgraphs as

$$G_x = G[\mathcal{I}_x] = (V_x, E_x), \quad G_y = G[\mathcal{I}_y] = (V_y, E_y),$$

where $V_x = \{(z_i, \mathbf{r}_i) : i \in \mathcal{I}_x\}$ and $V_y = \{(z_i, \mathbf{r}_i) : i \in \mathcal{I}_y\}$. The two index sets form a partition of the atoms in the atomic system, such that $\mathcal{I}_x \cap \mathcal{I}_y = \emptyset$ and $\mathcal{I}_x \cup \mathcal{I}_y = \{1, \dots, N\}$.

Molecular Graphs. For a molecule containing N atoms, the neighborhood of atom i is defined by all atoms within the cutoff radius r_c,

$$\mathcal{N}(i) = \{j \in \{1, \dots, N\} \setminus \{i\} \mid \|\mathbf{r}_j - \mathbf{r}_i\| < r_c\}.$$

The edge set E is obtained from these neighborhoods. For each neighboring pair (i, j), we define the relative displacement vector $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$, together with the interatomic distance $\|\mathbf{r}_{ij}\|$ and unit direction $\hat{\mathbf{r}}_{ij} = \frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|}$.

Crystal Graphs. Crystalline systems are represented analogously, with the additional requirement that periodic images of atoms in neighboring unit cells are taken into account. Let the lattice be specified by three lattice vectors $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$, collected in the lattice matrix $\mathbf{L}$. A periodic image of atom i, indexed by the integer translation vector $\mathbf{n} \in \mathbb{Z}^3$, has position $\mathbf{r}_i + \mathbf{n}\mathbf{L}$. The periodic neighborhood of atom i is therefore

$$\mathcal{N}(i) = \{(j, \mathbf{n}) \in \{1, \dots, N\} \times \mathbb{Z}^3 \;\big|\; 0 < \|\mathbf{r}_j + \mathbf{n}\mathbf{l} - \mathbf{r}_i\| < r_c\}.$$

Although the underlying crystal is infinite, the graph contains only the N atoms of the reference unit cell as nodes. Periodicity is represented through edges connecting these atoms to neighboring periodic images within the cutoff radius r_c. Similarly to molecules, for each such edge, we define the periodic displacement vector $\mathbf{r}_{ij\mathbf{n}} = \mathbf{r}_j + \mathbf{n}\mathbf{L} - \mathbf{r}_i$, along with the distance $\|\mathbf{r}_{ij\mathbf{n}}\|$ and unit direction $\hat{\mathbf{r}}_{ij\mathbf{n}}$.

Context and Target Partitioning. Given an atomic system, we partition it into the context and target index sets $\mathcal{I}_x$ and $\mathcal{I}_y$ introduced above by partitioning the atoms using a k-EgoNet subgraph. We first sample an anchor atom $c \sim \text{Unif}(\{1, \dots, N\})$ and a hop count $k \sim \text{Unit}(K)$ from a predefined set of hop counts K. The k-hop EgoNet centered at c is then defined by the index set

$$\mathcal{N}_k(c) = \{i \in \{1, \dots, N\} \mid d(c, i) \leq k\},$$

where $d(c, i)$ denotes the shortest-path distance between atoms c and i, measured in number of edges. We take the atoms of the sampled EgoNet $\mathcal{N}_k(c)$ as the context graph G_x, and the atoms in the complementary part as the target graph G_y. If the sampled value of k produces an EgoNet containing all atoms in the system, we instead use the largest value in K for which the target graph

remains non-empty. During training, the two subgraphs are used symmetrically: each subgraph acts once as the context G_x and once as the target G_y, and the loss is evaluated in both directions. The anchor atom c and hop count k are resampled at each training epoch, yielding different partitions of the atomic system throughout training. To encourage spatially local partitions, the graph used to determine the k-hop EgoNet is constructed using a smaller cutoff radius $r_k \ll r_c$. This auxillary graph is used only to determine the partition $(\mathcal{I}_x, \mathcal{I}_y)$. Once the partition has been obtained, the full graph G, context graph G_x, and target graph G_y are constructed using the larger interaction cutoff r_c. Thus, r_k controls the locality of the sampled partition, whereas r_c determines the atomic interactions represented in the graphs provided to the models.

A.2 Context and Target Encoders

The context encoder f_θ and target encoder $f_{\bar{\theta}}$ are both graph neural networks that take a 3D graph as input and produce atom-level feature representations. We use an equivariant architecture and write $\mathbf{s}_i = \{\mathbf{s}_i^{(l)}\}_{l=0}^{L_{\text{max}}}$ to denote atomic features, where $\mathbf{s}_i^{(0)}$ denote the invariant scalar features and $\mathbf{s}_i^{(l)}, l > 0$ denote the higher-order equivariant features from $l = 1, \ldots, L_{\text{max}}$. The parameters of the context encoder θ are updated via gradient based optimization during training, while the parameters of the target encoder $\bar{\theta}$ are an exponential moving average of the context encoder parameters: $\bar{\theta} \leftarrow \tau\bar{\theta} + (1-\tau)\theta$, where $\tau \in [0, 1]$ controls the rate of the exponential moving average. The context encoder is given only the context graph G_x and produces a representation for each context atom,

$$\mathbf{s}_x = f_\theta(G_x) = \{\mathbf{s}_{x_i}\}_{i \in \mathcal{I}_x}.$$

In contrast, the target encoder operates on the full atomic graph G. It therefore produces representations for all N atoms,

$$\tilde{\mathbf{s}}_y = f_{\bar{\theta}}(G) = \{\tilde{\mathbf{s}}_{y_i}\}_{i=1}^{N}.$$

After encoding the full graph, only the representations corresponding to atoms in the target subgraph are retained as prediction targets,

$$\mathbf{s}_y = \{\tilde{\mathbf{s}}_{y_i}\}_{i \in \mathcal{I}_y}.$$

Thus, although the learning objective is defined only over atoms belonging to the target graph G_y, their target representations are computed from the full atomic system. Each target representation can therefore incorporate information from the surrounding atomic environment through the message-passing operations of the target encoder, rather than being restricted to information contained within G_y alone. We argue that this provides richer target representations that encode the target atoms in the context of the complete atomic system.

A.3 Prediction

Given the context representations $\mathbf{s}_x$, a shallow graph neural network g_ϕ predicts representations produced by the target encoder. The predictor consists of a GNN body followed by objective-specific readout heads. Since the GNN body of the predictor operates on the atom features produced by the context encoder it does not contain an embedding layer. We consider two prediction objectives: a substructure objective that predicts a global representation of the target subgraph and an atom-level objective that predicts the representation of each individual target atom. First, the GNN body of the predictor produces further refined context-atom features $\mathbf{h}_x$, which are then mean-pooled and passed to the respective readout heads:

$$\mathbf{h}_x = g_\phi^{\text{GNN}}(G_x, \mathbf{s}_x), \quad \bar{\mathbf{h}}_x = \frac{1}{|\mathcal{I}_x|} \sum_{i \in \mathcal{I}_x} \mathbf{h}_{x_i}.$$

A.3.1 Substructure Target Prediction

In the substructure objective, the target is the mean-pooled invariant scalar features produced by the target encoder indexed over the target subgraph

$$\bar{\mathbf{s}}_y = \frac{1}{|\mathcal{I}_y|} \sum_{i \in \mathcal{I}_y} \mathbf{s}_{y_i}^{(0)},$$

and the substructure readout head is a two-layer MLP with SiLU activation applied to the invariant component of the pooled context

$$\hat{\mathbf{s}}_y = R_\phi^{\text{sub}}(\bar{\mathbf{h}}_x) = \text{MLP}(\bar{\mathbf{h}}_x^{(0)}),$$

mapping into the feature space of the target encoder.

A.3.2 Atom-level Target Prediction

The atom-level objective instead produces one prediction $\hat{\mathbf{s}}_{y_i}$ for every atom $i \in \mathcal{I}_y$, which is compared against the corresponding target encoder representation $\mathbf{s}_{y_i}$. In addition to the context features, the atom-level readout head is provided with a positional encoding $\mathbf{q}_i$ of the target atom's position relative to the context subgraph:

$$\hat{\mathbf{s}}_{y_i} = R_\phi^{\text{atom}}(\mathbf{h}_x, \mathbf{q}_i), \quad i \in \mathcal{I}_y.$$

We construct the positional encoding so that it has the same degree structure as the atom features, with each $\mathbf{q}_i^{(l)}$ transforming under rotation exactly as $\mathbf{h}_x^{(l)}$ does. For each degree $l \geq 1$ we then take the inner product of the two over the $2l+1$ components seperately in each feature channel and concatenate them with the invariant parts to form

$$\mathbf{u}_i = \left[\bar{\mathbf{h}}_x^{(0)} \| \mathbf{q}_i^{(0)} \| \langle \bar{\mathbf{h}}_x^{(1)}, \mathbf{q}_i^{(1)} \rangle \| \cdots \| \langle \bar{\mathbf{h}}_x^{(L_{\max})}, \mathbf{q}_i^{(L_{\max})} \rangle \right] \in \mathbb{R}^{C(L_{\max}+2)},$$

where $\cdot\|\cdot$ denotes concatenation and $\langle \bar{\mathbf{h}}_x^{(l)}, \mathbf{q}_i^{(l)} \rangle \in \mathbb{R}^C$ denote the channel-wise inner product, with c-th entry

$$\langle \bar{\mathbf{h}}_x^{(l)}, \mathbf{q}_i^{(l)} \rangle_c = \sum_{m=-l}^{l} \bar{h}_{x,mc}^{(l)} q_{i,mc}^{(l)}.$$

The invariant term $\bar{\mathbf{h}}_i^{(0)}$ carries only the content of the context, the term $\mathbf{q}_i^{(0)}$ depends only on the distance to the target atom, and the remaining terms couple the directional components of the context to the direction of the query. A two-layer MLP maps this feature into the target encoder's feature space

$$\hat{\mathbf{s}}_{y_i} = \text{MLP}(\mathbf{u}_i).$$

A.3.3 Equivariant Relative Positional Encoding

We encode target positions relative to the context using a radial-spherical-harmonic construction following Tensor Field Networks (Thomas et al., 2018), with distances expanded in a Gaussian radial basis as in SchNet (Schütt et al., 2017).

For molecules, we define the position of each target atom i relative to the context graph. First, the centroid of the context atoms is computed as $\mathbf{r}_x = \frac{1}{|\mathcal{I}_x|} \sum_{j \in \mathcal{I}_x} \mathbf{r}_j$. The relative displacement of target atom i from the context centroid is then $\mathbf{r}_{xi} = \mathbf{r}_i - \mathbf{r}_x$. We expand the distance $\|\mathbf{r}_{xi}\|$ in a Gaussian radial basis $\varphi(\|\mathbf{r}_{xi}\|) \in \mathbb{R}^K$, with

$$\varphi_k(\|\mathbf{r}_{xi}\|) = \exp\left(\frac{-(\|\mathbf{r}_{xi}\| - \mu_k)^2}{2\sigma^2}\right),$$

where the centers μ_k are spaced over $[0, r_q]$. The direction $\hat{\mathbf{r}}_{xi} = \mathbf{r}_{xi}/\|\mathbf{r}_{xi}\|$ is encoded using spherical harmonics, giving

$$q_{i,mc}^{(l)} = Y_m^{(l)}(\hat{\mathbf{r}}_{xi})(\mathbf{W}^{(l)} \varphi(\|\mathbf{r}_{xi}\|) + \mathbf{b}^{(l)})_c,$$

where $l = 0, \ldots, L_{\max}, |m| \leq l, c = 1, \ldots C$. The parameters $\mathbf{W}^{(l)} \in \mathbb{R}^{C \times K}, \mathbf{b}^{(l)} \in \mathbb{R}^C$ are a separate learned linear layer for each degree.

For crystals, the relative displacement $\mathbf{r}_{xi}$ is defined differently to account for periodic boundary conditions. The centroid of the context atoms is not well defined from the coordinates of a unit cell, since atoms that are spatially close may lie on opposite sides of a periodic boundary. We therefore define a reference atom $c_x \in \mathcal{I}_x$ for the context graph. When the context graph is the sampled k-EgoNet $c_x = c$, while when the complementary subgraph forms the context, c_x is sampled uniformly

from $\mathcal{I}_x$. The relative displacement of target atom i is then defined using its closest periodic image with respect to c_x

$$\mathbf{r}_{xi} = \mathbf{r}_i + \mathbf{n}^*\mathbf{L} - \mathbf{r}_{c_x}, \quad \mathbf{n}^* = \arg\min_{n\in\mathbb{Z}^3} \|\mathbf{r}_i + \mathbf{n}\mathbf{L} - \mathbf{r}_{c_x}\|.$$

Thus, for crystals $\mathbf{r}_{xi}$ is the minimum-image displacement from the context reference atom c_x to target atom i. The resulting displacement is then encoded in the same way as in the molecular case.

A.3.4 Loss

We use mean squared error between predicted and target representations, $\ell(\mathbf{a}, \mathbf{b}) = \frac{1}{C}\|\mathbf{a} - \mathbf{b}\|_2^2$, where C is the feature dimension. For the substructure objective, the loss is averaged over the B graphs in the batch

$$\mathcal{L}_{\text{sub}}^{x\to y} = \frac{1}{B}\sum_{g=1}^{B} \ell\left(\hat{\mathbf{s}}_y^{(g)}, \bar{\mathbf{s}}_y^{(g)}\right),$$

where $x \to y$ denotes the prediction direction in which G_x is used as the context graph and G_y as the target graph. For the atom-level objective, the loss is averaged over all target atoms across all graphs in the batch

$$\mathcal{L}_{\text{atom}}^{x\to y} = \frac{1}{\sum_{g=1}^{B}|\mathcal{I}_y^{(g)}|}\sum_{g=1}^{B}\sum_{i\in\mathcal{I}_y^{(g)}} \ell\left(\hat{\mathbf{s}}_{y_i}, \mathbf{s}_{y_i}\right).$$

Both objectives are evaluated symmetrically in both directions, with the two complementary subgraphs alternating as context and target. Thus, for $o \in \{\text{sub}, \text{atom}\}$,

$$\mathcal{L}_o = \frac{1}{2}\left(\mathcal{L}_o^{x\to y} + \mathcal{L}_o^{y\to x}\right),$$

and the complete loss is $\mathcal{L} = \lambda_{\text{sub}}\mathcal{L}_{\text{sub}} + \lambda_{\text{atom}}\mathcal{L}_{\text{atom}}$. We train models with each objective individually and jointly, setting $\lambda_{\text{pool}} = \lambda_{\text{atom}} = 1$.

A.4 Data Curation

We use structures at or near local energy minima because we expect their geometry to help constrain the identity of the target sub-structure and atom. The intuition is that, in a relaxed structure, the atomic arrangement must be compatible with the interactions between the constituent atom types. Strong distortions of the system can weaken this connection, making the target embedding of a specific atom *significantly* harder to predict from the context $\mathcal{I}_x$. Restricting the data to near-minimum structures should reduce such ambiguity, even though several atom types could still be compatible with the same context. An alternative to this approach would be to include off-equilibrium structures and then condition the model on the system's potential energy, as this would likely also help constrain the learning problem.

B Limitations

Throughout our experiments, we have used the Equiformer V3 architecture for all tasks. While we do observe a strong effect from using our SSL objective, we do not have a direct comparison to other SSL objectives with the same architecture. For ADMET datasets, other SSL objectives are not directly translatable due to their reliance on chemical bond encoding, whereas our SSL objective relies on the 3D structure of molecules. An additional nuance to this is that Atom-JEPA is the only ADMET method tested that explicitly uses a 3D structure as input from conformers. We hypothesise that in addition to the Atom-JEPA loss, the ADMET performance could be driven by the strong implicit biases present in Equiformer V3 such as rotational equivariance or the the data augmentation benefit of generating multiple conformers.

We found a strong influence from the conformer choices as highlighted in Appendix H.1.3. Notably, the number of conformers used during training and averaging the model prediction over multiple conformers improve prediction accuracy. At the same time the quality of the conformer (as judged by the energy), seems to not have an impact on the performance. This suggest that it is rather the

diversity of the conformers that improve the model not better conformers. This could be a problem when using weak conformer generation methods such as ours (see Appendix H.1.2), as their strong biases restrict the true conformer space. We have not investigated more accurate conformer generation strategies through higher order theory here.

ADMET tasks are notoriously difficult to benchmark correctly, due to their heavy generalization requirements from non-uniformly random splits, small test samples, and noisy labels. To asses optimal performance for each model, their hyperparameters would have been tuned through 2-fold cross-validation. It would have been computationally infeasible to do this for finetuning of Atom-JEPA and costly to do for all the MLP baselines. While it would be possible to do 2-fold cross-validation for the LGBT models, we opted to use our shared hyperparameter setup consistent across all models to make cross-model comparisons more fair.

C Datasets

Table 5 summarizes the datasets used for large-scale pretraining and downstream evaluation. In addition to its use as a downstream benchmark, we use QM9 for a separate set of pretraining ablations of the loss objectives.

We subsample the Alexandria PBE dataset by retaining only near-stable relaxed crystal structures with an energy above the convex hull of at most 0.05 eV/atom and containing between 2 and 100 atoms per unit cell. These filtering criteria reduce the dataset from 5,952,515 to 1,696,344 structures.

The Uni-Mol dataset contains 10 3D conformers per molecule. We use only a single conformer per molecule (the first), resulting in 18,837,320 training samples.

Table 5: **Summary of datasets.** Uni-Mol and Alexandria are used for large-scale pretraining. The remaining datasets are used for downstream evaluation. QM9 is additionally used for pretraining in the separate ablation experiments.

Dataset	Samples	Tasks	Description
Large-scale pretraining datasets			
Uni-Mol (Zhou et al., 2023)	18,837,320	–	3D molecular conformers
Alexandria PBE (Cavignac et al., 2026)	1,696,344	–	DFT-relaxed inorganic crystals
Downstream evaluation datasets			
ChEMBL-MT (Adrian et al., 2025)	114,000	25	Multi-task ADMET
ExpansionRx (MacDermott-Opeskin & Castellanos, 2026)	7,608	10	Multi-task ADMET
Biogen ADME (Fang et al., 2023)	3,521	6	Multi-task ADME
TDC-ADMET (Huang et al., 2021)	475–13,130	22	Individual-task ADMET
MoleculeNet (Wu et al., 2018)	1,513–93,097	8	Individual-task ADMET
QM9 (Ramakrishnan et al., 2014)	130,831	12	Quantum-chemical properties
Matbench (Dunn et al., 2020)	1,265–132,752	8	Materials-property prediction

C.1 Structural Overlap

Due to the size of Unimol and Alexandria, some of the structures used for pretraining of Atom-JEPA are also present in datasets of ADMET, QM9, and Matbench datasets. While self-supervised training has not been observed to bias accuracy noticeably (Xue et al., 2026), we still report the overlaps for transparency of the test sets for ADMET tasks in Table 6, full datasets of QM9 in Table 7, and full dataset of Matbench in Table 8. For most ADMET endpoints, Unimol contains somewhere around $35\% - 75\%$ of the structures of molecules seen in the endpoints. Due to the temporal split of ExpansionRX, only 1 test point is found in Unimol and at most 2% of the molecules have the same scaffold. We found that Atom-JEPA's gap to other models is the largest on specifically ExpansionRX, but that the accuracy cannot be attributed to structural overlap between the pretraining data and ExpansionRX. For QM9 the overlap is also quite small at 4%. For Matbench the overlap to Alexandria varies for the endpoints.

Table 6: **Uni-Mol pretraining-set overlap in downstream evaluation data.** Dataset rows are unions over their evaluation test membership. "Unique molecules" and "Found in Uni-Mol" use the Exact definition and with an observed label for that endpoint. Exact denotes identical RDKit canonical isomeric molecular graphs and retains stereochemistry, isotopes, formal charge, and disconnected fragments. Parent compares cleaned fragment parents, ignoring salts and solvents while retaining stereochemistry. Broad parent also normalizes charge and isotopes, removes stereochemistry, and uses a heteroatom-tautomer hash. Murcko scaffold is the percentage of unique Bemis-Murcko scaffolds observed in Uni-Mol and measures structural relatedness. Percentage rounded to nearest percentage.

Evaluation dataset	Unique molecules	Found in Uni-Mol	Overlap (%)			
			Exact	Parent	Broad parent	Murcko scaffold
ChEMBL-MT	23,213	11,964	52	52	73	90
CL Microsome Human	968	632	65	67	73	88
CL Microsome Mouse	136	84	62	66	69	92
CL Microsome Rat	341	226	66	66	69	87
CL Total Dog	90	54	60	71	94	97
CL Total Human	241	150	62	71	94	97
CL Total Monkey	42	21	50	55	95	97
CL Total Rat	124	72	58	65	94	95
CYP2C8 Inhibition	53	28	53	55	60	77
CYP2C9 Inhibition	515	264	51	53	60	81
CYP2D6 Inhibition	449	252	56	60	67	88
CYP3A4 Inhibition	1,084	615	57	58	64	79
Fraction unbound plasma (Dog)	50	23	46	62	96	95
Fraction unbound plasma (Human)	603	274	45	49	83	74
Fraction unbound plasma (Monkey)	24	10	42	50	96	95
Fraction unbound plasma (Rat)	70	40	57	67	96	95
Papp Caco2	1,216	495	41	42	74	90
Pgp (Human)	419	296	71	73	77	93
hERG binding	757	322	43	44	75	90
LogD pH 7.4	763	516	68	72	79	94
LogSaq Kinetic	11,350	6,528	58	58	82	94
LogSaq Thermo	6,750	2,835	42	43	59	86
VDss Dog	87	53	61	71	94	97
VDss Human	245	152	62	71	94	97
VDss Monkey	41	20	49	54	95	97
VDss Rat	111	61	55	63	95	95
ExpansionRx	2,282	1	0	0	0	2
LogD	2,270	1	0	0	0	2
KSOL	2,170	1	0	0	0	1
HLM CLint	782	0	0	0	0	1
MLM CLint	1,170	0	0	0	0	1
Caco-2 Permeability Papp A>B	1,616	0	0	0	0	2
Caco-2 Permeability Efflux	1,616	0	0	0	0	2
MPPB	454	0	0	0	0	1
MBPB	451	0	0	0	0	1
MGMB	209	0	0	0	0	0
Biogen ADME	1,830	907	50	50	63	83
Log HLM CLint (mL/min/kg)	1,620	809	50	50	63	83
Log MDR1-MDCK	1,384	703	51	51	63	83
Log SOLUBILITY PH 6.8 (ug/mL)	1,097	508	46	47	60	83
Log Plasma Protein Binding Human	120	53	44	44	64	75
Log Plasma Protein Binding Rat	93	42	45	45	67	81
Log RLM CLint (mL/min/kg)	1,596	804	50	51	64	84
TDC-ADMET	13,816	7,406	54	56	71	89
AMES	1,453	513	35	36	51	73
BBB Martins	394	248	63	72	84	93
Bioavailability Ma	128	89	70	83	94	99
Caco2 Wang	181	67	37	39	66	86
Clearance Hepatocyte AZ	203	142	70	73	82	94
Clearance Microsome AZ	221	155	70	72	76	94
CYP2C9 Substrate Carbon-Mangels	135	89	66	69	88	96
CYP2C9 Veith	2,419	1,541	64	66	82	93
CYP2D6 Substrate Carbon-Mangels	134	93	69	77	92	100
CYP2D6 Veith	2,626	1,707	65	67	84	94
CYP3A4 Substrate Carbon-Mangels	134	90	67	75	92	97
CYP3A4 Veith	2,467	1,603	65	67	82	93
DILI	96	59	61	70	96	99
Half-life Obach	134	84	63	78	93	96
hERG	128	35	27	32	73	74
HIA Hou	117	60	51	55	84	97
LD50 Zhu	1,466	538	37	38	53	84
Lipophilicity AstraZeneca	840	601	72	74	78	90
Pgb Broccatelli	245	137	56	60	82	90
PPBR AZ	343	241	70	71	79	96
Solubility AqSolDB	1,995	762	38	41	57	84
VDss Lombardo	221	45	20	25	89	67

Table 7: **Full QM9 overlap with Uni-Mol pretraining data.** Exact denotes an identical canonical isomeric molecular graph. Parent removes salts and solvents while retaining stereochemistry. Broad parent additionally normalizes charge, isotopes, stereochemistry, and common heteroatom tautomers. Murcko scaffold measures structural relatedness rather than molecule identity and excludes acyclic molecules from its denominator. All QM9 properties share this molecular inventory. The graph-based profiles exclude 1,819 PyG QM9 entries whose explicit-hydrogen SMILES cannot be sanitized by RDKit; Unique molecules therefore denotes unique valid graph identities.

Evaluation dataset	Unique molecules	Found in Uni-Mol	Overlap (%)			
			Exact	Parent	Broad parent	Murcko scaffold
QM9	128,822	4,742	4	4	5	9

Table 8: **Matbench overlap with Alexandria PBE pretraining data.** Strict requires the same elemental composition and a tight primitive-cell match without lattice rescaling. Relaxed uses lattice rescaling and moderate geometric tolerances to accommodate relaxation differences. Prototype permits element substitution but requires the same canonical anonymous formula, Pearson symbol, space group, and Wyckoff sequence. Composition requires only the same reduced elemental composition. The latter two columns measure structural or chemical relatedness, not identical materials.

Evaluation dataset	Unique structures	Found in Alexandria	Overlap (%)			
			Strict	Relaxed	Prototype	Composition
Matbench	151,731	59,032	39	44	76	59
Phonons	1,265	1,190	94	96	100	99
Dielectric	4,764	3,399	71	80	93	92
Log GVRH	10,987	8,697	79	83	99	93
Log KVRH	10,987	8,697	79	83	99	93
Perovskites	18,928	26	0	0	64	5
MP Gap	106,113	58,659	55	63	84	78
MP E Form	132,752	58,959	44	51	78	67
MP Is Metal	106,113	58,659	55	63	84	78

D Analysis of Learned Representations

Here we examine the learned representation space of Atom-JEPA using visualizations, nearest-neighbor analysis and probes on frozen features. First, for each of the two main models trained for molecules and crystals respectively, we randomly sample 50,000 structures from the corresponding pretraining dataset and mean-pool over the final-layer scalar atom features. For all embedding visualizations, we standardize the features and apply principal component analysis (PCA), retaining components that explain at least 90% of the variance, followed by UMAP with 60 neighbors and a minimum distance of 0.4 to obtain a two-dimensional projection. We color crystals by crystallographic symmetry and molecules by physicochemical properties and structural features. We further assess structural similarity within embedding neighborhoods and use probes to quantify the structural, geometric and physical-property information captured beyond what can be predicted from atomic composition, system size and geometry descriptor baselines. For the molecular model, we extend the analysis to 32,252 molecules from the manually annotated three-star subset of ChEBI (Chemical Entities of Biological Interest) (Malik et al., 2026), coloring their embeddings by chemical family. Finally, on QM9 molecules, we compare the pretrained molecular model with a randomly initialized counterpart through visualizations and probes of each QM9 target using final-layer and all-layer features, assessing Atom-JEPA pretraining's contribution beyond the 3D equivariant GNN backbone's inductive biases.

D.1 Structure and Property Organization in the Embedding Space

D.1.1 Crystal Representations

Figures 5 and 6 visualize the Atom-JEPA features of 50,000 random samples from the Alexandria pretraining dataset. Figure 5 colors each crystal by its corresponding crystal system and Figure 6 colors by the six most frequent space groups in the sample.

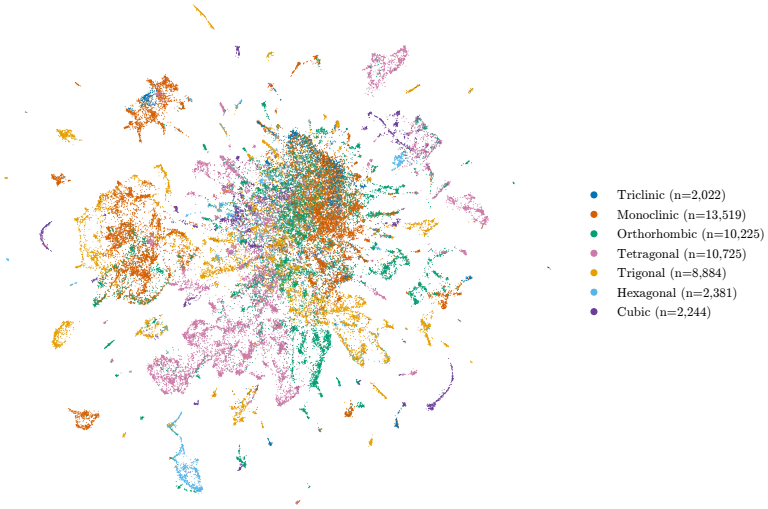
To quantify structural organization in the embedding space, we evaluate the 10 nearest neighbors of 1,000 randomly selected samples using Euclidean distance in Table 9. We compare these neighborhoods with nearest neighbors measured by composition (118-dimensional vector of elemental fractions), random crystals with matching atom count, and unrestricted random crystals. Embedding neighborhoods exhibit substantially greater space-group agreement than all three controls. Only 0.05% of embedding-neighbor pairs in the sample share the same reduced composition (elemental counts divided by their greatest common divisor), indicating that this symmetry organization extends across different chemical compositions.

Table 9: Structural neighborhood agreement.

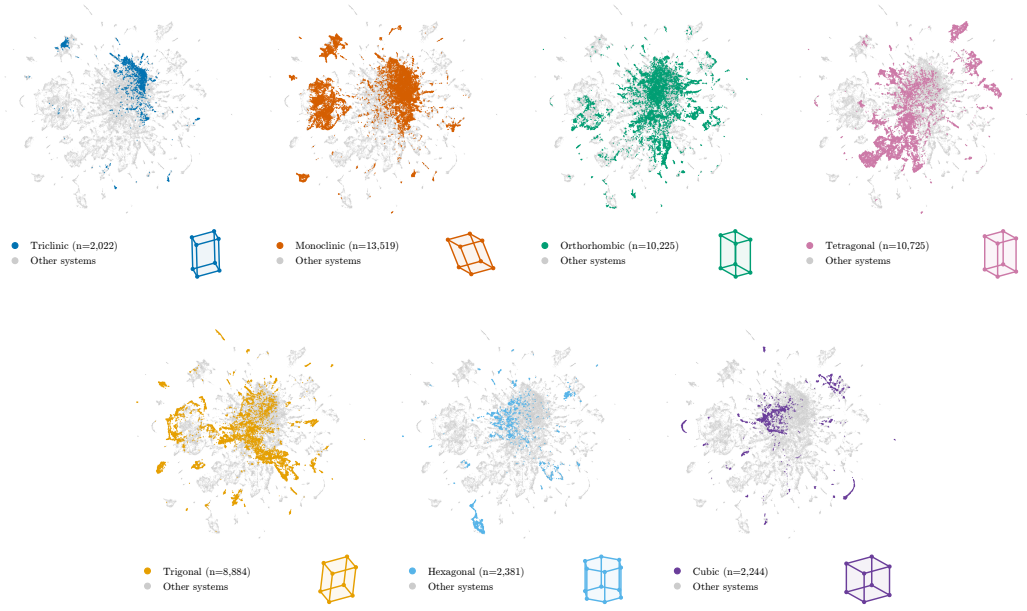
Neighbor selection	Same space group
Embedding neighbors	69.7%
Composition-only neighbors	14.4%
Random crystals with matching atom count	12.4%
Unmatched random crystals	4.9%

Next, we assess the information accessible through linear probes on the frozen features by predicting geometric quantities and physical properties. We partition the 50,000 sample into training, validation, and test sets containing approximately 70%, 15%, and 15% of the structures, respectively, with reduced compositions kept disjoint across partitions. We then fit ridge-regression probes with regularization selected by validation mean squared error. Features are standardized using training-set statistics, and performance is reported as R^2, the coefficient of determination on the test set.

Table 10 compares embeddings with a composition-and-size baseline for predicting log-volume per atom $\ln(V/N)$, where V is the unit-cell volume and N is the number of atoms in the cell as well as predicting mean coordination within 3 Å. The baseline represents each crystal using 122 features: 118 elemental fractions and four size features given by N, $\ln N$, N^2 and N^3 allowing the regression to model nonlinear relationships between cell size and the target. Coordination is calculated using periodic neighbours, including qualifying self-images. Combining embeddings with composition and size improves prediction of both quantities, indicating that the learned features include complementary

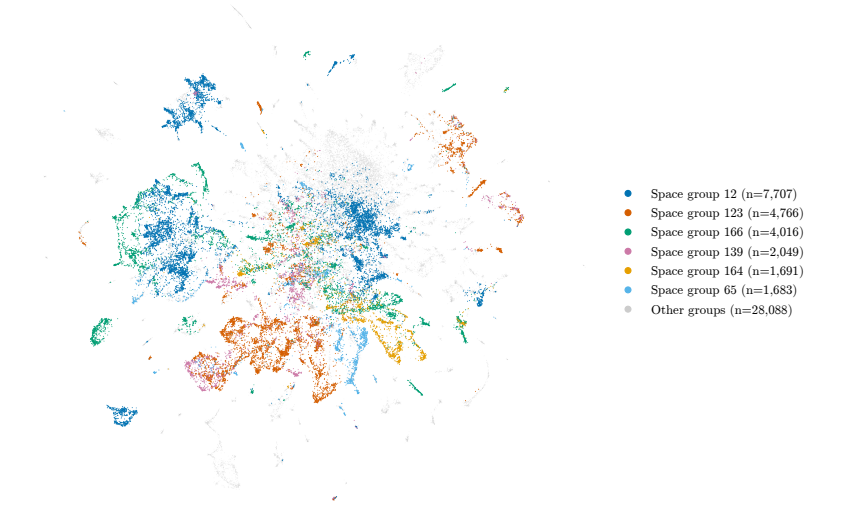


(a) Combined crystal systems.

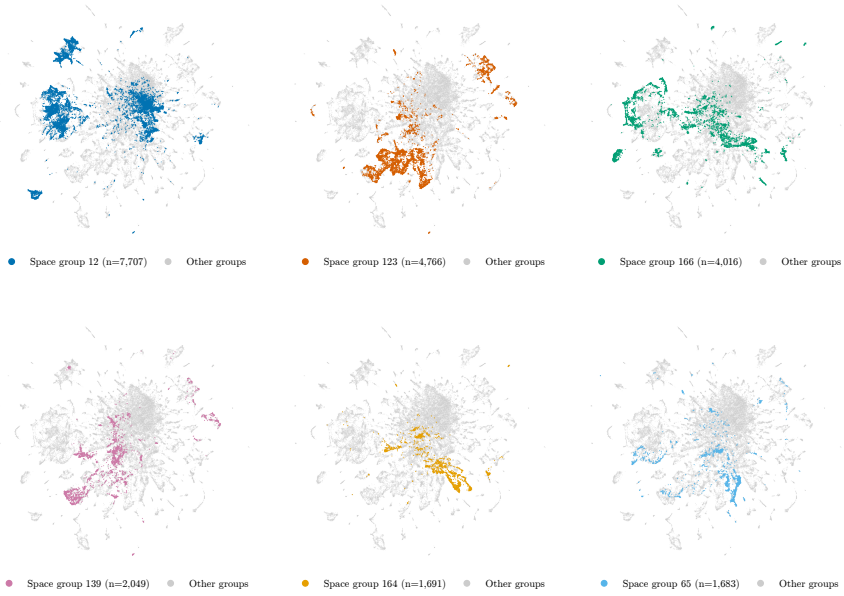


(b) Each crystal system highlighted against the others in gray.

Figure 5: UMAP visualization of crystal representations by crystal system.



(a) Combined space groups.



(b) Each of the six most frequent space groups highlighted against the others in gray.

Figure 6: UMAP visualization of crystal representations by space group.

geometric information that can be accessed by linear probes. While the embeddings alone slightly outperform the baseline for coordination, they perform worse for log-volume per atom.

Table 10: Test-set coefficient of determination R^2 for geometric quantities.

Predictor	Log-volume-per-atom	Coordination
Composition + atom count	.942	.794
Embeddings	.854	.814
Both together	**.973**	**.894**

Table 11 reports linear-probe performance for energy above hull, formation energy, and indirect band gap with targets taken from the corresponding Alexandria record. We compare embeddings with composition-and-size baselines, additionally incorporating nine geometric descriptors that summarize cell geometry and local atomic environments. Embeddings alone outperform these baselines for energy above hull but underperform for formation energy and band gap. However, combining embeddings with composition, size, and geometry yields the highest test R^2 for all three targets. This suggests that the embeddings provide complementary predictive features, rather than consistently replacing explicit descriptors in the setting of linear probing.

Formation energy is predicted most strongly, with the combined model improving R^2 from 0.861 to 0.908 over the baseline including geometry. Energy-above-hull prediction remains weak, despite achieving the largest relative improvement over that baseline: the combined model reaches $R^2 = 0.136$, which corresponds to an MAE of 11.7 meV/atom, compared with 13.2 meV/atom for a training-mean predictor. These results concern the near-stable subset used for pretraining, restricted to relaxed structures with energy above hull at most 0.05 eV/atom. The probe therefore assesses fine differences within the narrow 0–50 meV/atom interval, rather than discrimination across a broad range of crystal stabilities.

The nine geometric descriptors that we use are log-volume per atom; the ratio of the longest to shortest lattice-vector length; the root-mean-square cosine of the cell angles; the mean, standard deviation, minimum, and maximum coordination within 3 Å; and the mean and standard deviation of nearest-neighbor distances.

Table 11: Test-set coefficient of determination R^2 for physical properties.

Predictor	Energy above hull	Formation energy	Indirect band gap
Composition + size	.072	.821	.493
Composition + size + geometry	.079	.861	.507
Embeddings	.109	.712	.446
Embeddings + composition + size + geometry	**.136**	**.908**	**.542**

D.1.2 Molecular Representations

Next, we examine molecular representations using 50,000 randomly sampled molecules from the Unimol pretraining dataset. Figures 7, 8, and 9 shows visualizations colored by molecular weight, net formal charge, ring count, aromatic heavy-atom fraction, the six most frequent functional groups, and molecular scaffolds. At the low level, we observe limited separation according to molecular properties and no clear clustering by functional group. A small exception is negatively charged molecules, which are concentrated around two localized islands, whereas neutral and positively charged molecules are distributed broadly. At the mid-level, molecular scaffolds show clearer separation, with some distinguishable local clusters, although many scaffolds remain scattered. Thus, scaffold identity is more apparent in the organization of the representations than individual low-level features, but does not fully determine it.

To assess whether low-level information remains accessible despite this limited visual separation, table 12 evaluates linear ridge-regression probes for ring count, aromatic heavy atom fraction and radius of gyration, and compare to a composition-and-size baseline. We use a 70/15/15% train/val/test split of scaffold groups defined by Murcko scaffold, with acyclic molecules grouped by canonical SMILES. Again features are standardized using training-set statistics, and regularization is selected

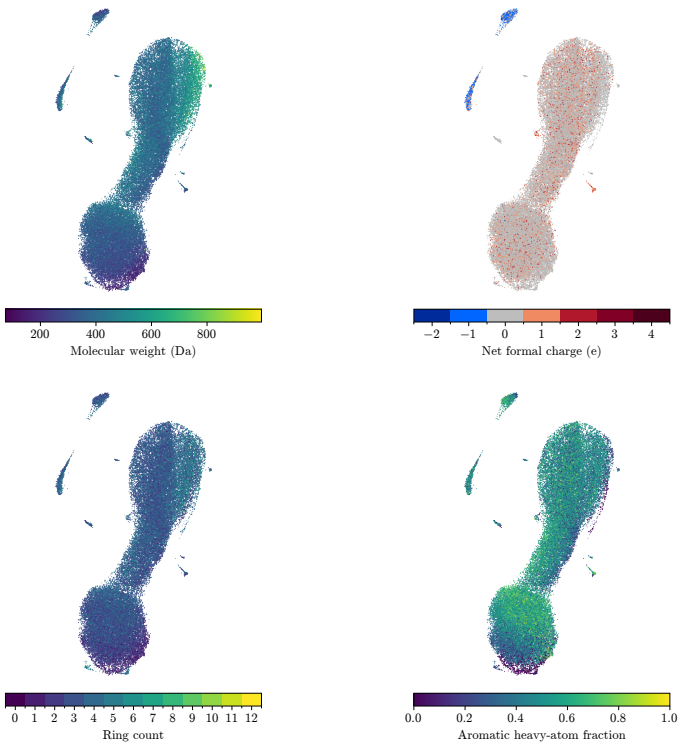
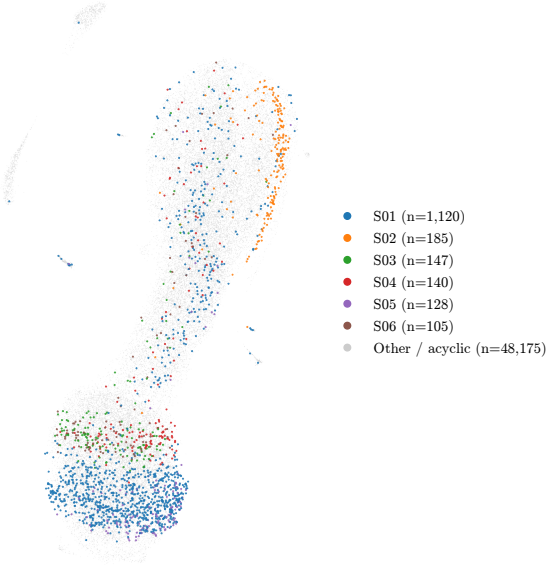


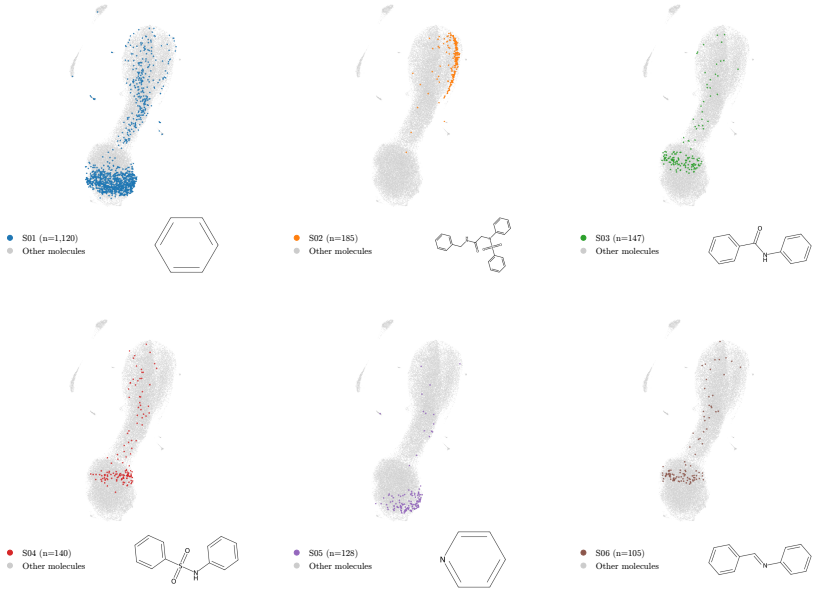
Figure 7: Uni-Mol representations colored by molecular weight, formal charge, ring count, and aromatic heavy-atom fraction.



Figure 8: Uni-Mol representations with six functional groups highlighted individually.



(a) 6 most frequent scaffolds combined



(b) 6 most frequent scaffolds highlighted individually

Figure 9: Uni-Mol representations with the 6 most frequent molecular scaffolds highlighted, shown combined and individually (remaining points in grey).

for each target by validation mean squared error, and we report the test-set coefficient of determination. Embeddings outperform composition and size for all three targets, with the largest improvement for radius of gyration. Combined features achieve the highest scores, although gains over embeddings alone are small for aromatic fraction and radius of gyration. Thus, despite limited visual separation in the UMAP projections, the representations make molecular structural and geometric properties accessible to linear prediction across scaffold-disjoint partitions.

Table 12: Uni-Mol structural probe results.

Input	Ring count	Aromatic fraction	Radius of gyration
Composition + size	.706	.642	.615
Embeddings	.909	.848	.964
Combined	**.958**	**.856**	**.969**

Next, we encode and plot approximately 32,252 molecules from the manually annotated three-star subset of ChEBI Malik et al. (2026) and color them by chemical family in figures 10 and 11. Here, the projections exhibit clearer clustering than the low-level property and mid-level scaffold colorings of the Uni-Mol sample. Table 13 quantifies this using linear ridge scores fitted to binary membership labels. We use the same scaffold-group splitting procedure and training-set standardization, and regularization as above, selecting regularization separately for each family by validation average precision. Evaluation includes families with at least 50 annotated positives in training, 20 in validation, and 20 in testing, together with at least 20 unannotated examples in each of validation and testing. We report average precision and AUROC averaged equally across eligible families. The outputs are ranking scores, and unannotated membership is treated as the negative class for evaluation. Embeddings substantially outperform composition and size in recovering annotated chemical-family membership, while their combination yields only small further gains. Together, these results suggest a hierarchy in the organization of the learned representations: low-level properties and functional groups show no or very limited visual separation, mid-level scaffolds exhibit partial clustering, and high-level chemical families show the clearest organization. This pattern is consistent with the hypothesized effect of Atom-JEPA pretraining, with latent-space prediction encouraging representations to prioritize higher-level chemical features while reducing the influence of low-level details. The strong structural probe results further suggest that this emphasis does not require discarding low-level information entirely; such information remains recoverable.

Table 13: ChEBI family probe results.

Input	Mean average precision	Mean AUROC
Composition + size	.186	.874
Embeddings	.547	.964
Combined	**.554**	**.966**

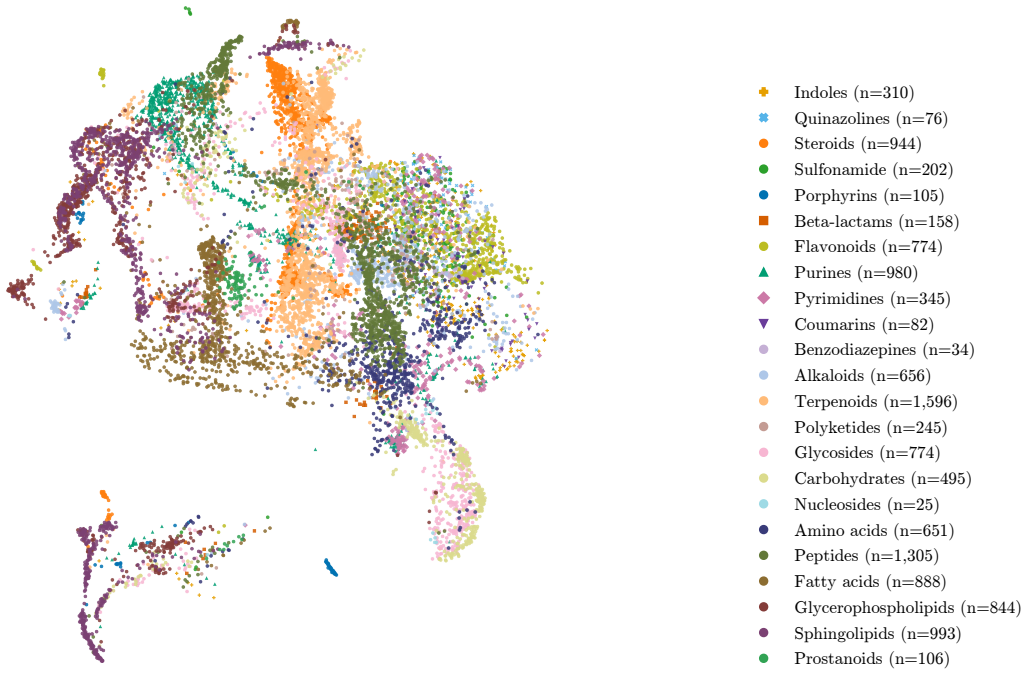


Figure 10: UMAP visualization of ChEBI representations, with exclusive chemical families highlighted.



Figure 11: Visualization of ChEBI representations, with chemical families highlighted individually and the rest in grey.

D.2 Effect of Pretraining on Molecular Representations

Finally, to assess how Atom-JEPA pretraining shapes molecular representations beyond the inductive biases of the GNN backbone, we compare pretrained and randomly initialized encoders with identical architectures on the QM9 dataset. For each, we train frozen-backbone probes using the dataset split specified in H and visualize representations of the test set molecules to relate their organization to predictive performance.

Figure 12 highlights the three most frequent chemical formulas in the test set. The randomly initialized encoder separates these formulas into 3 distinct clusters, suggesting that without pretraining representations are organized entirely by molecular size and elemental composition. In contrast, as found above Atom-JEPA representations are not separated by low-level features such as size and composition. Coloring the embeddings by the twelve QM9 properties (Figure 13) also reveals clear patterns for several targets in the random representations. These patterns are consistent with a simple organization by size and composition, which already accounts for substantial variation in several properties. Again, the Atom-JEPA representations do not exhibit the same apparent patterns.

To assess how these visual differences translate to downstream performance, we train all-layer MLP probes on frozen per-atom features from both encoders and report results in Table 14. All probes outperform the training-set mean predictor on every target, confirming that the backbone's inductive biases alone provide useful predictive features. Nevertheless, Atom-JEPA achieves substantially lower MAE across all twelve targets with both probe types. With all-layer features, pretraining reduces MAE by approximately three- to fourfold for polarizability, the HOMO–LUMO gap, HOMO and LUMO energies, and heat capacity, and by seven- to eightfold for the atom-referenced energies G, H, U, and U_0. Gains are smaller for dipole moment, where random features already perform comparatively well. An additional finding is that all-layer probes outperform last-layer probes on every target for Atom-JEPA and on eleven of twelve targets for the random encoder, with particularly pronounced gains for Atom-JEPA energy predictions.

These results show that clear visual separation by target value does not indicate greater predictive utility. The patterns in the random representations reflect a simple organization by molecular size and elemental composition, whereas Atom-JEPA supports substantially more accurate prediction without comparable formula-based clustering. Together with the chemical-family analysis, these findings are consistent with latent-space predictive pretraining shifting representations beyond these basic attributes toward higher-level, property-relevant molecular information that is not readily apparent in two-dimensional projections, but nevertheless improve downstream performance.

Table 14: QM9 frozen-backbone MLP probe test MAE results using all-layer or last-layer features for a pretrained encoder vs random initialization, with a training-set mean predictor as a baseline (atom-referenced for U, U_0, H, G).

Model	α a_0^3	$\Delta\varepsilon$ eV	$\varepsilon_{\text{HOMO}}$ eV	$\varepsilon_{\text{LUMO}}$ eV	μ D	C_v $\frac{\text{cal}}{\text{mol K}}$	G eV	H eV	$\langle R^2 \rangle$ a_0^2	U eV	U_0 eV	ZPVE eV
Training-set mean	6.343	1.077	.434	1.055	1.165	3.195	7.511	8.304	203.364	8.244	8.170	.720
All layers												
Atom-JEPA	**.091**	**.095**	**.061**	**.064**	**.032**	**.031**	**.013**	**.012**	**.419**	**.012**	**.012**	**.001**
Random initialization	.371	.279	.171	.221	.039	.108	.091	.089	1.850	.089	.091	.004
Last layer												
Atom-JEPA	.217	.141	.094	.099	.034	.071	.061	.070	1.579	.068	.072	.004
Random initialization	.479	.327	.192	.246	.038	.135	.114	.110	2.444	.114	.110	.005

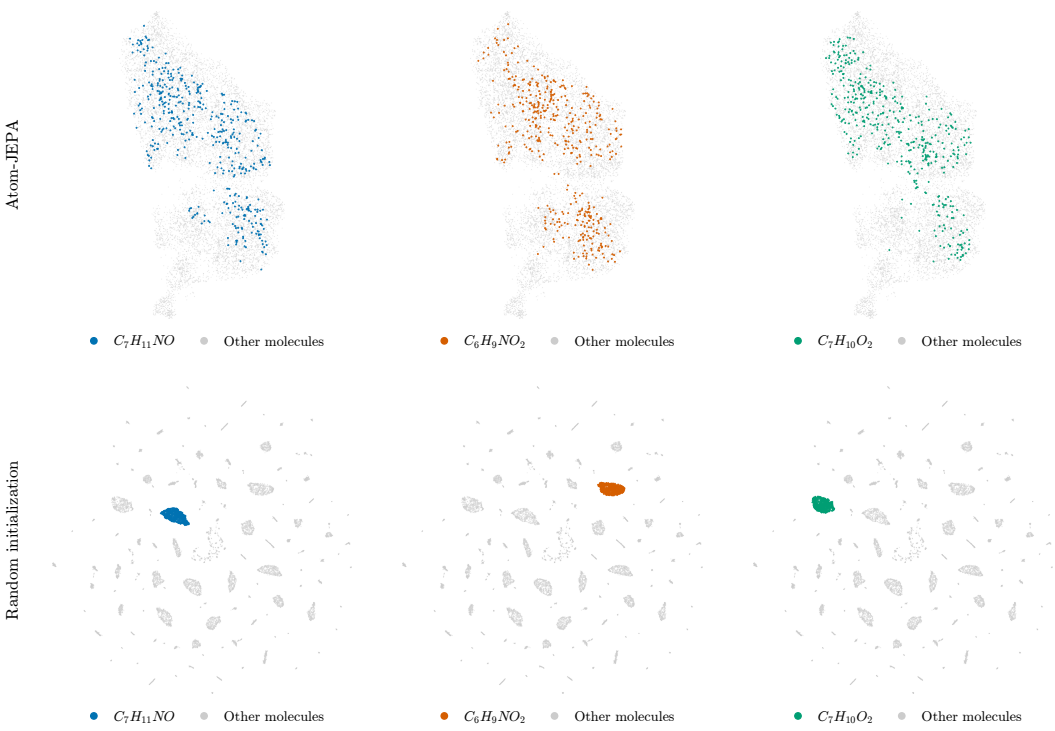


Figure 12: The three most frequent size/composition profiles highlighted in the randomly initialized and the Atom-JEPA pretrained QM9 representation, respectively

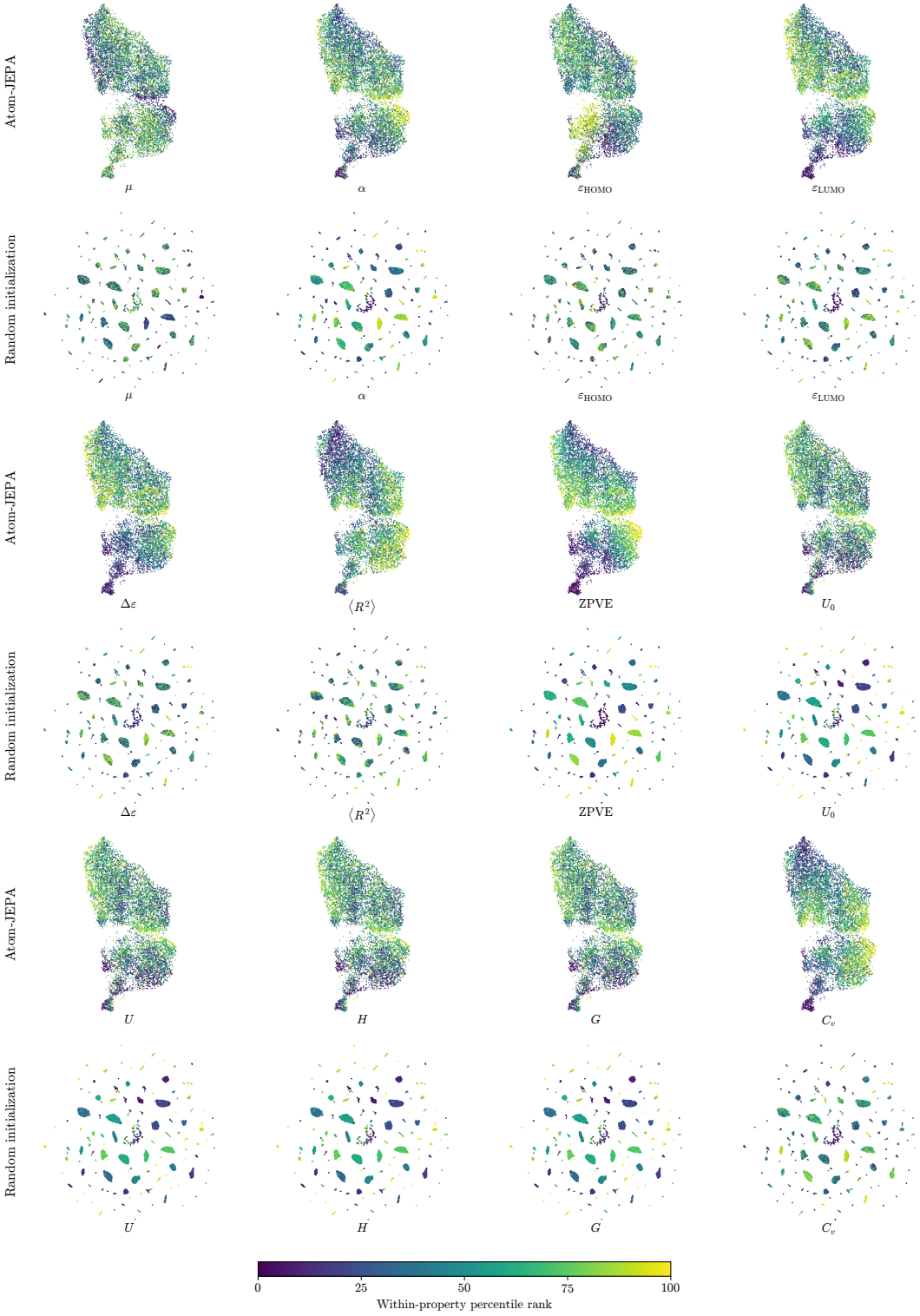


Figure 13: Pretrained and randomly initialized QM9 representations colored by within-property percentile rank for all twelve targets.

E ADMET FULL RESULTS

E.1 STATISTICAL TESTS SETUP

To compute statistical indistinguishable models in Table 1, we compute an ANOVA test over each endpoint. For Biogen and ChEMBL the design is blocked across folds, where as ExpansionRX is unpaired due to using the same test set each run. The best model that is compared against is the model with the lowest mean error over the respective endpoint folds. For runs where we have the individual fold errors, we check if each models computed 95% confidence intervals don't overlap using a pairwise Tukey test. Because we only have the mean run error for KERMT, we instead compare using a single-sided t-test with Holm–Bonferroni correction.

E.2 MODEL PREDICTION CORRELATION

We present the full analysis of the differences between the ADMET models. We quantify difference between the prediction of models by their spearman correlation (r). We use residuals instead of direct predictions to reduce model strength as a confounder. In the interest of ensembling model predictions, we also control against the case where one model happens to be weakly correlated with other models, but strongly correlated with the sum of the other models. For completeness we calculate the uniqueness score as follows: For endpoint t and run s, let $e_m \in \mathbb{R}^n$ denote model m's residual vector $\hat{y} - y$ over the n test molecules with an observed label for t, and let $\mathcal{P}(m)$ be one representative of each of the four model families other than m's own. We regress e_m on $\{e_k\}_{k\in\mathcal{P}(m)}$ with an intercept by ordinary least squares and define

$$U_m(t,s) = 1 - R^2_{\text{adj}} = \left(1 - R^2\right) \frac{n-1}{n-p-1}, \tag{7}$$

where $p = |\mathcal{P}(m)| = 4$ for our for model comparisons, R^2 is the coefficient of determination of that fit. U_m is the share of model m's squared error that no linear combination of the other families reproduces,. We do not include endpoints with less than 30 test datapoints for robustness. We only compare other models to Atom-JEPA averaged over 10 conformers to not bias the results by also including the the non-conformer averaged Atom-JEPA model. The single and $10\times$ conformer predictions correlate with over 0.99. The results can be seen in Table 15.

Table 15: **Model distinctiveness.** How reproducible each model's errors are from the other four model families, measured two ways on the same per-molecule test residuals. emphresidual correlation is the averaged *pairwise* correlation between other models, averaged over test data-points, *Uniqueness* is the $1 - R^2_{\text{adj}}$ from regressing a model's residuals on all four peers' residuals at once. Per-benchmark columns are means over that benchmark's endpoints. *Mean rank* ranks the families within each benchmark and endpoint cell and averages with equal weight per benchmark, with rank 1 = most distinct. The brackets are a stratified cluster bootstrap over endpoints. *Group* shows the group of models that cannot separated by the paired 95% bootstrap interval on their mean-rank difference.

Model	Biogen	ChEMBL-MT	ExpansionRx	Mean rank across endpoints	Group
Residual correlation $\bar{r}$ to peers *(lower is more distinct)*					
Atom-JEPA $\times$10 conf	.723	**.791**	.730	**1.98** [1.66, 2.34]	a
Chemprop	**.719**	.803	**.681**	2.53 [2.02, 2.88]	ab
Mol-JEPA · modalities	.738	.801	.749	2.78 [2.42, 3.14]	b
Morgan + RDKit · LGBM	.724	.814	.761	2.98 [2.59, 3.31]	b
CKERMT	.779	.829	.787	4.73 [4.48, 4.86]	c
Uniqueness $1 - R^2_{\text{adj}}$ *(higher is more distinct)*					
Atom-JEPA $\times$10 conf	**.327**	**.253**	.283	**1.97** [1.65, 2.32]	a
Chemprop	.325	.221	**.392**	2.53 [2.02, 2.89]	ab
Mol-JEPA · modalities	.285	.221	.246	2.89 [2.52, 3.23]	b
Morgan + RDKit · LGBM	.312	.192	.228	2.99 [2.57, 3.39]	b
CKERMT	.214	.171	.187	4.63 [4.39, 4.78]	c

E.3 BIOGEN ADME

Table 16: Test MAE mean ± standard deviation across scaffold splits across tasks for Biogen dataset. †: Data from Figure 13 in Xue et al. (2026).

Model	HLM CLint	RLM CLint	MDCK	logS	PPB Human	PPB Rat	Mean
GBT models							
RDKit	.3773±.0190	.4347±.0268	.3493±.0116	.3725±.0204	.4713±.1293	.4872±.0594	.4154±.0298
Morgan	.4115±.0242	.4616±.0182	.3682±.0131	.3838±.0230	.5085±.1062	.6293±.0828	.4605±.0260
Morgan + RDKit	.3712±.0140	.4136±.0239	.3362±.0103	.3663±.0143	.4462±.1109	.5125±.0984	.4077±.0339
Morgan + RDKit + Avalon + ErG	.3863±.0186	.4160±.0196	.3497±.0154	.3882±.0252	.5454±.1507	.5686±.0660	.4424±.0247
Mol-JEPA CLS	.4051±.0218	.4523±.0187	.3689±.0096	.3887±.0262	.5302±.1025	.5264±.0835	.4453±.0317
Atom-JEPA Last Layer	.4300±.0221	.4869±.0206	.3791±.0127	.3813±.0201	.5182±.0828	.5090±.0991	.4507±.0213
Atom-JEPA All Layer	.4021±.0134	.4613±.0198	.3563±.0039	.3691±.0229	.4798±.1242	.4753±.0763	.4240±.0266
MLP models							
RDKit	.3772±.0249	.4400±.0239	.3383±.0192	.3798±.0272	.4112±.0802	.4227±.0932	.3949±.0292
Morgan	.4323±.0271	.5055±.0343	.3917±.0215	.4159±.0221	.4793±.0778	.4794±.0793	.4507±.0218
Morgan + RDKit	.3469±.0197	.3998±.0202	.3344±.0238	.3772±.0133	.4034±.0778	.4412±.0778	.3838±.0194
Morgan + RDKit + Avalon + ErG	.3336±.0161	.3892±.0139	.3197±.0193	.3679±.0241	.4538±.0758	.4651±.0205	.3882±.0137
Mol-JEPA CLS	.3710±.0209	.4148±.0205	.3357±.0172	.3660±.0176	.4000±.0441	.3810±.0375	.3781±.0137
Mol-JEPA modalities	.3423±.0115	.3932±.0221	.3237±.0143	.3293±.0174	.4048±.0585	**.3782**±.0409	.3619±.0141
Atom-JEPA Last Layer	.3754±.0244	.4304±.0144	.3383±.0086	.3594±.0238	.4297±.0637	.4516±.0503	.3975±.0181
Atom-JEPA All Layer	.3757±.0230	.4322±.0257	.3153±.0115	.3518±.0197	.4282±.0883	.4320±.0772	.3892±.0270
Finetuning							
No Pretrain	.3232±.0170	.3781±.0082	.3322±.0114	.3274±.0091	.4696±.1024	.4623±.0711	.3821±.0281
Atom-JEPA	.2950±.0146	.3548±.0185	.2816±.0107	.3246±.0110	.4221±.0922	.3970±.0747	.3458±.0260
Atom-JEPA (10-conf. ensemble)	**.2931**±.0147	**.3523**±.0186	.2796±.0110	**.3231**±.0112	.4199±.0940	.3932±.0759	**.3435**±.0267
Literature models							
Chemprop	.3133±.0058	.3683±.0141	.2954±.0070	.3305±.0143	**.3968**±.1017	.4062±.1136	.3518±.0317
CKERMT Our runs	.2996±.0095	.3646±.0194	.2892±.0136	.3265±.0104	.4093±.0820	.4451±.0953	.3557±.0290
KERMT base†	.320	.402	.289	.344	–	–	–
KERMT task-specific†	.309	.402	.278	.339	–	–	–
CKERMT base†	.302	.389	.275	.329	–	–	–
CKERMT task-specific†	.294	.389	**.268**	.333	–	–	–

Table 17: Test Pearson r^2 ± standard deviation across cluster splits across tasks for Biogen dataset. All literature model data from Adrian et al. (2025).

Model	HLM CLint	RLM CLint	MDCK	logS	PPB Human	PPB Rat	Mean
GBT models							
RDKit	.3846±.0170	.4337±.0041	.3921±.0071	.2143±.0069	.3666±.0005	.3733±.0807	.3608±.0134
Morgan	.3256±.0244	.3722±.0079	.3635±.0093	.1569±.0032	.3720±.0551	.0372±.0293	.2712±.0103
Morgan + RDKit	.4411±.0020	.4950±.0055	.4397±.0073	.2346±.0007	.4152±.0089	.3111±.0075	.3895±.0008
Morgan + RDKit + Avalon + ErG	.4469±.0229	.4800±.0083	.3881±.0075	.2428±.0125	.3454±.0115	.2428±.0730	.3577±.0146
Atom-JEPA Last Layer	.2940±.0205	.3840±.0111	.2995±.0118	.2293±.0027	.3661±.0463	.2613±.0799	.3057±.0057
Atom-JEPA All Layer	.3598±.0168	.4226±.0057	.4018±.0167	.2429±.0378	.4674±.0436	.4564±.0267	.3918±.0008
MLP models							
RDKit	.4821±.0022	.5182±.0073	.4707±.0073	.2141±.0092	.4479±.0370	.4347±.0364	.4279±.0141
Morgan	.2489±.0071	.3001±.0031	.2284±.0060	.2256±.0144	.3418±.0352	.2555±.0316	.2667±.0119
Morgan + RDKit	.4223±.0139	.4911±.0116	.4131±.0040	.1898±.0021	.4922±.0395	.4088±.0663	.4029±.0123
Morgan + RDKit + Avalon + ErG	.4639±.0092	.5312±.0098	.5027±.0168	.2164±.0070	.4828±.0407	.3682±.0311	.4275±.0135
Atom-JEPA Last Layer	.4515±.0043	.4706±.0036	.4180±.0013	.3240±.0147	.4224±.0350	.3639±.0450	.4084±.0159
Atom-JEPA All Layer	.4566±.0053	.4836±.0078	.5208±.0054	.3518±.0005	.5498±.0067	.5042±.0059	.4778±.0011
Finetuning							
Atom-JEPA	.6384±.0048	.6417±.0053	.6021±.0104	.3728±.0154	.5808±.0187	.4332±.0532	.5448±.0146
Atom-JEPA (10-conf. ensemble)	.6444±.0048	.6465±.0052	.6070±.0105	.3777±.0153	.5881±.0190	.4380±.0534	.5503±.0146
Literature models							
MolFormer ST	.510±.004	.570±.015	.415±.007	.181±.010	.269±.035	.119±.023	.344
Chemprop ST	.566±.022	.596±.014	.475±.009	.296±.027	.190±.012	.121±.085	.374
KPGT $ST_{complex}$	.505±.004	.548±.013	.520±.007	.337±.003	.291±.031	.269±.043	.412
KPGT ST	.580±.009	.597±.013	.538±.012	.369±.012	.358±.062	.284±.064	.454
KERMT ST	.600±.019	.577±.017	.525±.009	.343±.009	.381±.018	.314±.064	.457
Chemprop MT	.558±.019	.602±.011	.519±.012	.380±.012	.572±.059	.501±.046	.522
KPGT MT	.562±.007	.580±.020	.529±.007	.318±.013	.362±.019	.315±.020	.444
KERMT MT	.634±.010	.602±.011	.535±.009	.365±.008	.481±.040	.427±.038	.507

E.4 EXPANSIONRX AND CHEMBL-MT

Table 18: Test MAE mean ± standard deviation across cluster splits across tasks for ExpansionRX dataset. †: Data from Xue et al. (2026)

Model	log D	KSOL	HLM CLint	MLM CLint	Caco-2 Papp A→B	Caco-2 Efflux	MPPB	MBPB	MGMB	Macro test
GBDT models										
Morgan †	.517±.011	.490±.028	.382±.007	.406±.011	.511±.013	.518±.032	.313±.020	.317±.007	.396±.012	.428±.004
RDKit	.567±.009	.535±.007	.363±.004	.401±.008	.468±.005	.494±.010	.346±.011	.322±.005	.353±.014	.428±.002
Morgan + RDKit	.523±.009	.540±.010	.355±.009	.409±.008	.458±.009	.514±.012	.332±.006	.277±.008	.357±.013	.418±.003
Morgan + RDKit + Avalon + ErG	.511±.019	.477±.010	.371±.009	.445±.017	.464±.022	.482±.032	.333±.012	.311±.021	.328±.022	.414±.005
Mol-JEPA cls	.586±.005	.521±.007	.381±.006	.435±.007	.530±.005	.524±.004	.337±.015	.376±.018	.331±.007	.447±.005
Atom-JEPA Last Layer	.613±.017	.495±.004	.389±.004	.426±.003	.586±.020	.556±.008	.328±.018	.327±.014	.374±.010	.455±.004
Atom-JEPA All Layers	.577±.015	.509±.004	.357±.008	.431±.008	.533±.017	.522±.009	.354±.007	.331±.009	.309±.009	.436±.003
MLP models										
Morgan	.635±.014	.464±.032	.417±.013	.448±.006	.630±.021	.579±.019	.261±.003	.306±.014	.292±.011	.448±.006
RDKit	.647±.010	.598±.008	.406±.006	.440±.006	.516±.004	.548±.004	.437±.016	.391±.011	.414±.013	.489±.006
Morgan + RDKit	.494±.004	.418±.014	.373±.011	.454±.020	.558±.012	.522±.010	.259±.009	.263±.009	.300±.006	.405±.006
Morgan + RDKit + Avalon + ErG	.476±.010	.389±.004	.396±.004	.419±.019	.512±.011	.454±.007	.229±.008	.244±.004	.282±.003	.378±.004
Mol-JEPA CLS	.573±.013	.509±.020	.374±.004	.456±.007	.457±.005	.483±.010	.338±.017	.372±.021	.394±.014	.440±.009
Mol-JEPA Modalities	.545±.014	.486±.010	.396±.011	.434±.018	.477±.010	.457±.012	.356±.007	.369±.007	.337±.003	.428±.003
Atom-JEPA All Layers	.624±.015	.522±.012	.361±.009	.445±.007	.520±.010	.551±.012	.337±.011	.265±.005	.283±.004	.434±.006
Finetuning										
No Pretrain	.390±.007	.376±.005	.347±.007	.411±.005	.529±.005	.486±.007	.258±.007	.227±.007	.282±.004	.367±.004
Atom-JEPA	.363±.006	.346±.003	.316±.006	.403±.006	.475±.006	.439±.006	.218±.006	.188±.003	.259±.004	.334±.002
Atom-JEPA (10-conf. ensemble)	.363±.006	.344±.003	.316±.006	.403±.006	.475±.006	.439±.006	.218±.005	.188±.003	.259±.004	.334±.002
Literature models										
Chemprop	.449±.009	.404±.006	.552±.105	.487±.025	.528±.032	.456±.024	.251±.016	.339±.019	.371±.031	.426±.012
CKERMT Our runs	.407±.009	.406±.008	.350±.010	.385±.009	.472±.009	.466±.007	.250±.003	.273±.011	.333±.023	.372±.003
KERMT base‡	.393	.417	.429	.458	.472	.489	.247	.225	.287	.380
KERMT task-specific‡	.396	.396	.400	.423	.469	.479	.263	.251	.299	.375
CMIM only task-specific‡	.461	.477	.418	.420	.468	.479	.284	.247	.293	.394
CKERMT base	–	–	–	–	–	–	–	–	–	–
CKERMT task-specific‡	.390	.387	.405	.396	.470	.471	.252	.232	.286	.365
CKERMT upscale task-specific†	.382	.385	.401	.394	.463	.455	.259	.233	.288	.362
CKERMT Biogen task-specific†	.388	.395	.393	.406	.457	.460	.257	.234	.289	.364
CKERMT Biogen sampled-30k task-specific†	.393	.373	.413	.410	.453	.447	.228	.220	.290	.359
CKERMT Biogen+OA+CM task-specific†	.377	.375	.397	.395	.460	.459	.255	.226	.300	.360
CKERMT upscale+Biogen task-specific†	.388	.375	.411	.403	.452	.458	.249	.225	.288	.361

Table 19: Test MAE on the ChEMBL-MT benchmark. Values marked with ‡ are taken from Figure 15 in Xue et al. (2026). Mic-H, Mic-M, and Mic-R denote microsomal clearance in human, mouse, and rat, respectively. Tot-D, Tot-H, Tot-M, and Tot-R denote total clearance in dog, human, monkey, and rat. fu-D, fu-H, fu-M, and fu-R denote the fraction unbound in plasma for the corresponding species. VD-D, VD-H, VD-M, and VD-R denote steady-state volume of distribution. Kin-S and Therm-S denote kinetic and thermodynamic $\log S_{\mathrm{aq}}$. Mean is the Overall value reported in Figure 15. The figure reports means to three decimal places without numerical standard deviations. Lower is better.

Model	Mic-H	Mic-M	Mic-R	Tot-D	Tot-H	Tot-M	Tot-R	2C8	2C9	2D6	3A4	fu-D	fu-H	log D	fu-M	Papp	P-gp	fu-R	VD-D	VD-H	VD-M	VD-R	hERG	Kin-S	Therm-S	Mean
LBGM																										
Morgan	.421	.452	.502	.463	.512	.347	.470	.506	.570	.617	.649	.504	.458	.766	.537	.476	.456	.504	.430	.466	.362	.488	.514	.379	1.135	.519
RDKit	.400	.406	.506	.421	.472	.313	.459	.513	.554	.589	.660	.434	.392	.630	.539	.467	.436	.512	.389	.403	.356	.444	.530	.369	.813	.480
RDKit + Morgan	.399	.385	.513	.411	.464	.335	.456	.496	.556	.576	.633	.443	.390	.614	.549	.457	.433	.504	.377	.409	.425	.443	.509	.352	.809	.478
Morgan + RDKit + Avalon + ErG	.401	.442	.498	.453	.508	.317	.454	.523	.551	.608	.634	.433	.402	.616	.507	.480	.431	.516	.421	.424	.416	.468	.506	.361	.840	.488
Mol-JEPA CLS	.431	.415	.542	.417	.506	.304	.449	.515	.567	.605	.665	.483	.449	.772	.573	.534	.438	.468	.442	.445	.392	.477	.545	.389	1.149	.519
Atom-JEPA Last Layers	.455	.436	.531	.413	.506	.333	.445	.509	.602	.635	.729	.479	.436	.868	.563	.536	.467	.518	.419	.463	.459	.465	.581	.407	.932	.527
Atom-JEPA All Layers	.425	.431	.502	.413	.496	.339	.450	.519	.574	.606	.673	.454	.399	.751	.600	.508	.445	.491	.428	.436	.420	.457	.541	.382	.895	.505
MLP																										
Morgan	.443	.503	.467	.451	.548	.385	.476	.480	.597	.672	.669	.460	.478	.842	.475	.545	.495	.468	.465	.459	.488	.471	.598	.442	1.365	.550
RDKit	.413	.556	.566	.437	.481	.320	.421	.585	.591	.651	.688	.460	.408	.666	.391	.502	.430	.477	.365	.430	.336	.450	.559	.398	.893	.499
RDKit + Morgan	.513	.485	.560	.467	.553	.346	.479	.528	.599	.677	.730	.598	.681	1.040	.581	.613	.507	.558	.542	.514	.526	.556	.672	.554	1.818	.628
Morgan + RDKit + Avalon + ErG	.510	.480	.559	.459	.548	.347	.476	.521	.599	.675	.729	.554	.623	.981	.598	.606	.508	.512	.517	.491	.506	.521	.672	.485	1.391	.595
Mol-JEPA cls	.428	.460	.553	.388	.513	.394	.446	.564	.560	.619	.652	.410	.382	.682	.363	.543	.451	.389	.383	.427	.414	.461	.550	.378	1.019	.497
Mol-JEPA Modalities	.405	.444	.545	.395	.510	.358	.428	.536	.533	.584	.647	.477	.388	.619	.445	.505	.444	.434	.379	.411	.381	.460	.521	.369	.979	.488
Atom-JEPA Last Layers	.443	.482	.559	.390	.478	.337	.435	.492	.597	.614	.716	.437	.415	.737	.523	.544	.457	.495	.435	.476	.416	.454	.572	.395	.861	.510
Atom-JEPA All Layers	.451	.459	.577	.402	.505	.359	.447	.502	.597	.617	.711	.437	.412	.746	.471	.556	.446	.462	.490	.482	.411	.468	.578	.405	.948	.518
Finetuning																										
No Pretrain	.376	.417	.473	.414	.488	.301	.443	.472	.565	.575	.585	.442	.382	.442	.384	.457	.443	.461	.561	.281	.755	.406	.407	.400	.427	.454
Atom-JEPA	.365	.447	.472	.402	.472	.319	.436	.451	.534	.607	.592	.420	.378	.409	.390	.471	.459	.484	.560	.279	.781	.422	.416	.423	.427	.457
Atom-JEPA (10-conf. ensemble)	.360	.443	.468	.401	.470	.316	.433	.447	.527	.599	.585	.419	.374	.408	.388	.466	.453	.477	.552	.274	.774	.420	.414	.420	.426	.453
Literature models																										
Chemprop	.368	.490	.476	.427	.483	.377	.465	.494	.534	.621	.626	.429	.368	.536	.438	.409	.449	.431	.498	.401	.452	.472	.484	.271	.920	.477
CKERMT Our runs	.361	.464	.459	.420	.484	.317	.463	.492	.509	.586	.582	.395	.350	.401	.443	.408	.414	.470	.516	.266	.869	.422	.398	.354	.463	.452
KERMT base†	.357	.441	.439	.454	.482	.414	.482	.486	.509	.576	.604	.409	.376	.514	.442	.426	.430	.449	.417	.408	.369	.454	.480	.287	.959	.466
KERMT task-specific†	.360	.435	.455	.460	.473	.353	.488	.495	.512	.573	.598	.413	.361	.525	.436	.428	.424	.418	.405	.387	.371	.460	.476	.276	.910	.460
CMIM only task-specific†	.369	.433	.467	.439	.456	.324	.423	.514	.512	.593	.606	.388	.376	.576	.420	.448	.432	.432	.402	.385	.354	.447	.491	.293	.856	.458
CKERMT task-specific†	.362	.442	.432	.423	.471	.326	.453	.499	.506	.592	.586	.414	.348	.515	.394	.424	.427	.415	.409	.383	.360	.455	.475	.267	.878	.450
CKERMT upscale task-specific†	.354	.433	.439	.425	.479	.338	.442	.490	.495	.568	.582	.402	.348	.507	.401	.412	.420	.410	.406	.393	.370	.448	.469	.265	.834	.445
CKERMT Biogen task-specific†	.360	.446	.436	.424	.466	.330	.445	.499	.508	.578	.580	.414	.348	.516	.388	.413	.425	.417	.404	.388	.366	.442	.474	.269	.885	.449
CKERMT Biogen sampled-30k task-specific †	.361	.471	.436	.421	.474	.332	.447	.494	.500	.576	.571	.427	.343	.510	.399	.407	.429	.397	.404	.387	.363	.444	.459	.263	.816	.445
CKERMT Biogen+OA+CM task-specific †	.355	.443	.432	.420	.473	.345	.446	.480	.499	.572	.569	.409	.349	.506	.370	.410	.420	.402	.400	.386	.356	.452	.469	.262	.814	.442
CKERMT upscale+Biogen task-specific †	.365	.443	.436	.413	.467	.345	.456	.490	.496	.582	.588	.417	.353	.506	.394	.422	.426	.412	.392	.383	.347	.448	.467	.265	.841	.446

E.5 TDC ADMET

Table 20: Results on the TDC-ADMET benchmark: absorption. Columns correspond, from left to right, to `caco2_wang`, `hia_hou`, `pgp_broccatelli`, `bioavailability_ma`, `lipophilicity_astrazeneca`, and `solubility_aqsoldb`. Numbers from Koleiev et al. (2026).

Model Unit	Caco2 MAE↓	HIA AUROC↑	P-gp AUROC↑	Bioav. AUROC↑	Lipo. MAE↓	Solub. MAE↓
GBDT models						
Morgan	.3530±.0370	.8170±.0300	.8650±.0230	.5690±.0730	.6060±.0170	.9700±.0060
RDKit	.3020±.0110	.8900±.0500	.8690±.0190	.5800±.0430	.5690±.0110	.7630±.0110
Morgan + RDKit	.2950±.0060	.8940±.0400	.8590±.0080	.5960±.0550	.5420±.0050	.7760±.0280
Morgan + RDKit + Avalon + ErG	.3290±.0150	.9410±.0170	.8690±.0160	.5790±.0770	.5400±.0110	.8030±.0050
Atom-JEPA Last Layer	.3390±.0040	.9380±.0330	.8760±.0110	.6100±.0560	.7640±.0190	.9270±.0150
Atom-JEPA All Layers	.3090±.0150	.9410±.0280	.8670±.0210	.5850±.0550	.6490±.0050	.8620±.0130
MLP models						
Morgan	.4410±.0050	.8940±.0050	.8900±.0240	.5990±.0540	.6720±.0080	1.1700±.0080
RDKit	.3590±.0100	.9790±.0050	.8990±.0050	.6890±.0080	.5640±.0080	.7930±.0380
RDKit + Morgan	.3160±.0100	.9400±.0180	.9120±.0130	.6730±.0240	.5310±.0020	.8320±.0060
RDKit + Morgan + Avalon + ErG	.3620±.1170	.9490±.0040	.9280±.0030	.6060±.0540	.6800±.2390	.9390±.2730
Atom-JEPA Last Layer	.2970±.0050	.9440±.0140	.8850±.0090	.6370±.0330	.6310±.0100	.8000±.0130
Atom-JEPA All Layers	.2970±.0010	.9600±.0190	.8810±.0070	.6040±.0150	.5430±.0230	.7720±.0190
Other models						
MapLight	.276±.005	.983±.001	.932±.001	.741±.007	.542±.003	.787±.006
MapLight+GNN	.290±.008	.989±.001	.938±.0	.740±.01	.525±.003	.795±.004
CaliciBoost	.256±.006	–	–	–	–	–
Atom-JEPA	.2910±.0100	.9800±.0100	.9180±.0100	.6900±.0370	.3970±.0030	.7800±.0180

Table 21: Results on the TDC-ADMET benchmark: distribution. Columns correspond, from left to right, to `bbb_martins`, `ppbr_az`, and `vdss_lombardo`. Numbers from Koleiev et al. (2026). RDKit + Morgan and RDKit + Morgan + Avalon + ErG MLP diverged during training on VDSS Lombardo

Model Unit	BBB AUROC↑	PPBR MAE↓	VDss Spear↑
GBDT models			
Morgan	.8300±.0110	8.6780±.1380	.6070±.0230
RDKit	.8770±.0150	7.5460±.1110	.6220±.0090
Morgan + RDKit	.8600±.0140	7.5100±.0980	.6250±.0260
Morgan + RDKit + Avalon + ErG	.8600±.0260	8.1450±.3010	.5750±.0230
Atom-JEPA Last Layer	.8540±.0080	8.8830±.1440	.5800±.0260
Atom-JEPA All Layers	.8500±.0210	8.1380±.0690	.5840±.0400
MLP models			
Morgan	.8380±.0110	8.5710±.0550	.5070±.0460
RDKit	.8910±.0140	7.6240±.0730	.5550±.0110
RDKit + Morgan	.9040±.0060	8.1030±.0880	nan±nan
RDKit + Morgan + Avalon + ErG	.9020±.0090	9.9660±1.4280	nan±nan
Atom-JEPA Last Layer	.8690±.0150	8.1870±.1250	.5370±.0430
Atom-JEPA All Layers	.8920±.0230	7.9370±.0390	.5770±.0460
Other models			
MapLight	.920±.002	7.622±.065	.716±.005
MapLight+GNN	.916±.002	7.573±.163	.716±.006
Atom-JEPA	.8960±.0150	7.6690±.2530	.6800±.0360

Table 22: Results on the TDC-ADMET benchmark: metabolism. Columns correspond, from left to right, to `cyp2c9_veith`, `cyp2d6_veith`, `cyp3a4_veith`, `cyp2c9_substrate_carbonmangels`, `cyp2d6_substrate_carbonmangels`, and `cyp3a4_substrate_carbonmangels`. Numbers from Koleiev et al. (2026).

Model Unit	2C9-I AUPRC↑	2D6-I AUPRC↑	3A4-I AUPRC↑	2C9-S AUPRC↑	2D6-S AUPRC↑	3A4-S AUROC↑
GBDT models						
Morgan	.6730±.0590	.5300±.0100	.7800±.0050	.2810±.0000	.4660±.0960	.5990±.0190
RDKit	.7110±.0090	.5880±.0150	.7840±.0100	.2970±.0250	.5420±.1280	.6170±.0280
Morgan + RDKit	.7120±.0100	.6000±.0110	.8090±.0060	.2810±.0090	.5560±.0500	.6310±.0090
Morgan + RDKit + Avalon + ErG	.6930±.0120	.5630±.0290	.8100±.0110	.2930±.0150	.5140±.1230	.6210±.0300
Atom-JEPA Last Layer	.6450±.0750	.3360±.0790	.7740±.0100	.3090±.0370	.5330±.0780	.5880±.0390
Atom-JEPA All Layers	.7000±.0180	.5100±.0050	.7950±.0040	.2940±.0170	.5170±.1090	.6130±.0340
MLP models						
Morgan	.7110±.0050	.6040±.0050	.8320±.0050	.4320±.0330	.6420±.0370	.6240±.0130
RDKit	.7390±.0130	.6260±.0030	.8350±.0050	.3900±.0140	.6680±.0290	.5990±.0160
RDKit + Morgan	.7510±.0040	.6520±.0050	.8700±.0020	.4310±.0280	.7200±.0210	.6290±.0470
RDKit + Morgan + Avalon + ErG	.7460±.0200	.5660±.2060	.8450±.0640	.4220±.0560	.6280±.1530	.5840±.0500
Atom-JEPA Last Layer	.7560±.0060	.6480±.0060	.8400±.0050	.3680±.0190	.5870±.0530	.5950±.0230
Atom-JEPA All Layers	.7620±.0040	.6470±.0050	.8480±.0020	.4150±.0710	.6420±.0300	.5900±.0120
Other models						
MapLight	.786±.004	.720±.002	.881±.001	.411±.005	.718±.008	.649±.006
MapLight+GNN	.859±.001	.790±.001	.916±.00	.446±.015	.716±.007	.642±.004
Atom-JEPA	.7720±.0060	.6610±.0180	.8680±.0050	.4560±.0270	.6840±.0550	.5790±.0230

Table 23: Results on the TDC-ADMET benchmark: excretion. Columns correspond, from left to right, to `half_life_obach`, `clearance_microsome_az`, and `clearance_hepatocyte_az`. Numbers from Koleiev et al. (2026).

Model Unit	Half-life Spear↑	Cl-Mic Spear↑	Cl-Hep Spear↑
GBDT models			
Morgan	.4130±.0430	.4670±.0290	.3690±.0250
RDKit	.3520±.0350	.5710±.0310	.4030±.0480
Morgan + RDKit	.4150±.0370	.5510±.0190	.4270±.0330
Morgan + RDKit + Avalon + ErG	.3810±.0440	.4620±.0680	.3750±.0190
Atom-JEPA Last Layer	.4120±.1510	.4560±.0470	.2500±.0390
Atom-JEPA All Layers	.4310±.0740	.4980±.0280	.2480±.0760
MLP models			
Morgan	.3440±.0360	.4990±.0160	.2910±.0520
RDKit	.3350±.0700	.5890±.0330	.3670±.0180
RDKit + Morgan	.2550±.1480	.5570±.0150	.3840±.0090
RDKit + Morgan + Avalon + ErG	.2400±.1120	.4090±.2030	.2290±.2250
Atom-JEPA Last Layer	.4560±.0480	.5650±.0310	.3460±.0250
Atom-JEPA All Layers	.3830±.1100	.5450±.0270	.3020±.0320
Other models			
MapLight	.559±.011	.625±.009	.469±.016
MapLight+GNN	.563±.009	.635±.007	.500±.012
CaliciBoost	–	–	–
Atom-JEPA	.4910±.0900	.6020±.0210	.4990±.0230

Table 24: Results on the TDC-ADMET benchmark: toxicity. Columns correspond, from left to right, to `herg`, `ames`, `dili`, and `ld50_zhu`. Numbers from Koleiev et al. (2026).

Model **Unit**	**hERG** **AUROC↑**	**AMES** **AUROC↑**	**DILI** **AUROC↑**	**LD50** **MAE↓**
GBDT models				
Morgan	.8100±.0180	.7690±.0170	.8640±.0170	.6750±.0170
RDKit	.7910±.0220	.8050±.0050	.8920±.0230	.6370±.0120
Morgan + RDKit	.8120±.0170	.8160±.0100	.8870±.0180	.6500±.0090
Morgan + RDKit + Avalon + ErG	.7680±.0530	.8050±.0130	.8300±.0470	.6400±.0130
Atom-JEPA Last Layer	.7600±.0210	.7660±.0110	.8830±.0480	.6970±.0090
Atom-JEPA All Layers	.8080±.0160	.8050±.0050	.9130±.0390	.6540±.0100
MLP models				
Morgan	.7770±.0150	.7920±.0140	.8520±.0180	.6670±.0120
RDKit	.8360±.0040	.8160±.0070	.8480±.0310	.6750±.0130
RDKit + Morgan	.8170±.0300	.8380±.0190	.8590±.0100	.6030±.0150
RDKit + Morgan + Avalon + ErG	.6460±.2500	.8330±.0080	.8290±.0210	.5530±.0210
Atom-JEPA Last Layer	.7930±.0080	.8190±.0070	.9270±.0090	.6520±.0110
Atom-JEPA All Layers	.8250±.0250	.8320±.0020	.9070±.0020	.6230±.0210
Other models				
MapLight	.873±.004	.865±.001	.883±.003	.618±.005
MapLight+GNN	.880±.003	.872±.001	.918±.005	.563±.009
Atom-JEPA	.8450±.0190	.8400±.0120	.8980±.0180	.5590±.0100

F Effect of Pretraining Objectives and Data Scale Full Results

Table 25 shows the full results for the MLP probe evaluation of each pretrained encoder as described in Section 4.2. The pretraining configurations differ to some degree between QM9 and Uni-Mol to accommodate differences in dataset size and molecular composition, as detailed in Table 26. These differences include architectural, graph-partitioning, and optimization hyperparameters. Within QM9, the three objective variants share the exact same pretraining hyperparameters. Within Uni-Mol, the atom-only encoder was pretrained using multiple nodes to speed up training time, with the global batch size increased from 128 to 384 and the maximum learning rate increased from 1.4×10^{-4} to 2.42×10^{-4}, approximately following square-root scaling with batch size. This adjustment accommodates the distributed training configuration.

All encoders are evaluated using the same all-layer MLP probing protocol on frozen features. The results therefore support the observed performance trends across the evaluated configurations, including the stronger relative performance of the joint objective with Uni-Mol pretraining and the broad improvements from pretraining on the larger dataset.

Table 25: All-layer MLP probes on frozen encoder features pretrained on the QM9 dataset and Unimol dataset respectively across combinations of the pretraining objectives. The evaluation metric is MAE (↓). **Bold:** best result within each pretraining group; **bold and underline**: best overall. "atom" and "sub" denote atom-level and substructure-level objectives, respectively.

Task Units	α ma_0^3	$\Delta\varepsilon$ meV	$\varepsilon_{\text{HOMO}}$ meV	$\varepsilon_{\text{LUMO}}$ meV	μ mD	C_ν $\frac{\text{mcal}}{\text{mol K}}$	G meV	H meV	R^2 ma_0^2	U meV	U_0 meV	ZPVE meV
Pretrained on QM9 (130k)												
atom+sub	115	130.6	86.1	88.0	56.0	40.8	22.3	22.9	**<u>396</u>**	22.6	22.1	1.8
atom-only	**97**	**114.4**	**69.6**	81.4	**49.4**	**32.4**	**18.4**	**18.4**	398	**18.4**	**18.6**	**1.5**
sub-only	104	131.6	90.0	**80.2**	82.3	42.7	23.0	22.7	420	22.5	22.4	1.9
Pretrained on Uni-Mol (19M)												
atom+sub	**<u>91</u>**	**<u>94.9</u>**	**<u>60.5</u>**	**<u>64.0</u>**	32.2	**<u>30.9</u>**	**<u>13.2</u>**	**<u>12.4</u>**	**419**	**<u>12.4</u>**	**<u>12.4</u>**	**<u>1.4</u>**
atom-only	96	95.7	63.3	64.9	**<u>29.2</u>**	31.4	13.4	13.0	464	13.3	13.6	**<u>1.4</u>**

G Pretraining Implementation Details

G.1 Hyperparameters

Table 26 contains the hyperparameters used for the pretraining on Uni-Mol and Alexandria as well as the hyperparameters used for pretraining the three encoders on QM9, which we use for the ablation studies on prediction objectives and data scale. We use EquiformerV3 Liao et al. (2026) for the context encoder, target encoder, and predictor throughout, following the notation of Equiformer V3 for the architectural hyperparameters. The context and target encoders use identical architectures. The predictor uses the same configuration as the encoders except for the hyperparameters explicitly listed in the predictor section of the table. In particular, the predictor GNN body is more shallow with 3 transformer blocks instead of 8 and a feature dimension of 128 rather than 256.

Pretraining was performed on 8 NVIDIA H100 80GB HBM3 GPUs per run. The Uni-Mol run took approximately 309 hours (2,475 GPU-hours) for 1,364,000 training steps, and the Alexandria run took approximately 268 hours (2,147 GPU-hours) for 1,321,000 training steps.

G.2 Representation Collapse Metrics

To monitor representation collapse during pretraining, we track RankMe Garrido et al. (2023) and α-reQ Agrawal et al. (2022) computed on the pooled representations of the target encoder. Both metrics operate on an embedding matrix $D \in \mathbb{R}^{B \times d_{embed}}$ formed by stacking the pooled representations across the batch, including both evaluation directions of the symmetric loss (i.e., each partition acting once as G_y against the other as context, and vice versa), gathered across all 8 GPUs used for pretraining. With a per-GPU batch size of 16, this gives $B = 256$, matching the target encoder's feature dimension $d_{embed} = 256$. RankMe is an estimate of the effective rank of D via the Shannon entropy of its singular value distribution, bounded by $\min(B, d_{embed}) = 256$ in our setting. A RankMe value collapsing towards 1 indicates that the target encoder is producing near-constant or low-dimensional outputs, which would mean representation collapse. α-ReQ instead fits a power-law

Table 26: Hyperparameters used for pretraining on Uni-Mol, Alexandria, and QM9. We use EquiformerV3 for the context encoder, target encoder, and predictor. The context and target encoders use identical architectures. Predictor hyperparameters not explicitly listed are identical to those of the corresponding encoder.

Hyperparameter	**Uni-Mol**	**Alexandria**	**QM9**
Encoder			
Maximum degree $L_{\max}$	2	2	6
Maximum order $M_{\max}$	2	2	2
Number of Transformer blocks	8	8	8
Embedding dimension d_{embed}	$(2, 256)$	$(2, 256)$	$(6, 256)$
Attention hidden dimension $d_{\text{attn-hidden}}$	$(2, 64)$	$(2, 64)$	$(6, 64)$
Number of attention heads h	8	8	8
Attention alpha dimension $d_{\text{attn-}\alpha}$	$(0, 64)$	$(0, 64)$	$(0, 64)$
Attention value dimension $d_{\text{attn-value}}$	$(2, 16)$	$(2, 16)$	$(6, 16)$
FFN hidden dimension d_{ffn}	$(2, 512)$	$(2, 512)$	$(6, 512)$
Attention grid resolution (R_ϕ, R_θ)	$(8, 14)$	$(8, 14)$	$(8, 20)$
FFN grid resolution (R_ϕ, R_θ)	$(14, 14)$	$(14, 14)$	$(20, 20)$
Radial basis function	Gaussian	Gaussian	Gaussian
Number of radial bases	128	128	128
Radial hidden scalar dimension d_{edge}	$(0, 128)$	$(0, 128)$	$(0, 128)$
Predictor			
Number of Transformer blocks	3	3	3
Embedding dimension d_{embed}	$(2, 128)$	$(2, 128)$	$(6, 128)$
Encoder-to-predictor projection	$(2, 256) \to (2, 128)$	$(2, 256) \to (2, 128)$	$(6, 256) \to (6, 128)$
Pooling readout hidden dimension	128	128	128
Pooling readout output dimension	256	256	256
Number of positional radial bases	128	128	128
Positional radial range (Å)	12	12	12
Positional embedding dimension	$(2, 128)$	$(2, 128)$	$(6, 128)$
Atom-head input dimension	512	512	1024
Atom-head hidden dimension	128	128	128
Atom-head output dimension	256	256	256
Graph Construction and Partitioning			
Radius-graph cutoff r_c (Å)	6	6	6
k-EgoNet cutoff r_k (Å)	1.8	3.5	1.8
k-EgoNet hops	$\{1, 3, 5, 10\}$	$\{1, 2, 3, 5\}$	$\{1, 2\}$
Optimization			
Optimizer	AdamW	AdamW	AdamW
Learning-rate schedule	Cosine + linear warmup	Cosine + linear warmup	Cosine + linear warmup
Warmup fraction	.15	.15	.15
Maximum learning rate	1.4×10^{-4} (atom+pool) 2.42×10^{-4} (atom-only)	1.4×10^{-4}	1.0×10^{-4}
Batch size	128 (atom+pool) 384 (atom-only)	128	64
Number of epochs	10	100	200
Weight decay	.04	.04	.04
Dropout rate	.0	.0	.0
Stochastic depth	.05	.05	.05
Gradient clipping norm	100	100	100
Target EMA decay τ	.995 $\to$ 1.0 (linear)	.995 $\to$ 1.0 (linear)	.995 $\to$ 1.0 (linear)

exponent α to the decay of D's covariance eigen spectrum, such that a large α is an indication of a small number of directions dominating the representation, and small α indicates an overly flat, noise-dominated spectrum.

Figure 14 shows both metrics over the course of pretraining on the Uni-Mol and Alexandria datasets, for the joint-objective encoder. In both cases, we observe that the RankMe metric rises sharply from the initial value and then plateaus at around ($\sim$80 for Uni-Mol, $\sim$130–140 for Alexandria). On Uni-Mol α-ReQ falls from its initial value and then stabilizes in a moderate range. For Alexandria it also falls from its initial value, and then slowly increases throughout training, remaining still within a

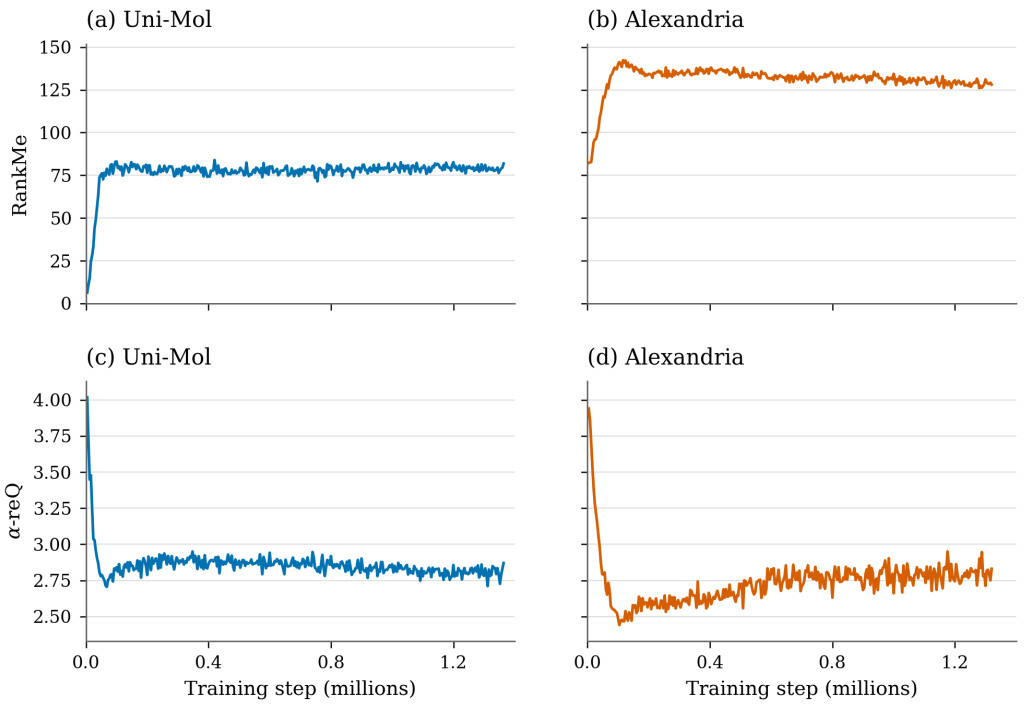


Figure 14: Evolution of RankMe (top) and α-reQ (bottom) during pretraining for the Uni-Mol (left) and Alexandria (right) encoders.

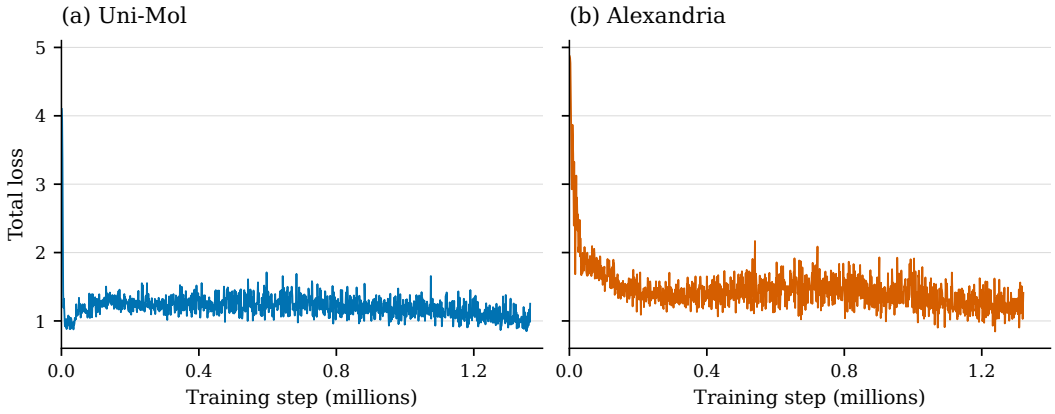


Figure 15: Total training loss during pretraining for the Uni-Mol (left) and Alexandria (right) encoders.

moderate range. Overall, both metrics remain stable throughout training, indicating that the context and target encoders do not undergo representation collapse over the course of pretraining.

Figure 15 shows the total loss for both runs during training. In both cases we observe a pattern of a sharp initial decline as the predictor and encoders leave initialization, followed by a period of gradual increase as the EMA target encoder drifts away form the context encoder, and finally a convergence phase towards the end of training as the target and context encoders are brought back into closer alignment, allowing the loss to decrease again.

H FINETUNING IMPLEMENTATION DETAILS

H.1 ADMET FINETUNING

Atom-JEPA Due to the label imbalances and objective differences between targets present in the multi-task datasets, we follow CKERMT (Xue et al., 2026) and employ two tricks; an individual MLP head for each of the individual tasks, and homoscedastic uncertainty task weighting of the loss (Kendall et al., 2018). We follow the learning from Tompa et al. (2026), relying on freezing the pretrained model for better generalisation, and otherwise only tune the learning rate. We do deviate from their recommendations of avoiding weight decay on the pre-trained model by instead using L2-SP (Xuhong et al., 2018) to penalize the model from changing too much of the pretrained weights.

LGBT and MLP Baselines : For baselines, we implement a gradient boosted tree (LGBT) and a 3-layer MLP model using different combination of features: Morgan fingerprints, RDKit features, ErG, and Avalon features. In addition, we also experiment with adding frozen Atom-JEPA features.

CKERMT As Xue et al. (2026) does not provide test MAE of individual folds, we reran their experiments to get the uncertainty for statistical tests. We use their model with the largest self-supervised dataset called "CKERMT Biogen+OA+CM task-specific". Our results agree closely with the published mean MAEs across datasets.

Mol-JEPA We also impelement Mol-JEPA with the same MLP and LGBT setup either using their pooled molecule level embedding (CLS) or their full 12 modality output (modalities).

Chemprop For the Chemprop baseline we use the v2 version, with Biogen tuned hyperparameters from Adrian et al. (2025). We also tried default parameters but found them to perform worse.

For all models except Atom-JEPA and Chemprop, we run a hyper-parameter tuning setup with Optuna using a budget of 500 iterations.

H.1.1 HYPERPARAMETER TUNING

We take the training split of one of the first split in Adrian et al. (2025) on Biogen split by clusters, which we part into a training and validation split. We select the validation macro MAE as our optimisation target. over all targets using Optuna for hyperparameter tuning. For all models except , we use a budget of 500 iterations with the TPE optimizer and hyperband pruning. All with a warm start of 15 iterations. For each feature extract(s) + feature processor, we run the full hyperparameter pipeline. If multiple feature extractors are used in the same model, all their hyperparameters are tuned at the same time. Some of the considerations for design choices. For the LGBM models we fixed the number of estimators high at 2000, as the complexity of the model is already protected by early stopping. The ranges for the LGBM was based on the values from Fang et al. (2023) and Optuna's LightGBMTuner, but by design left slightly larger. We use CDF-normalized RDKit features.

We allow a higher amount of optimisation iterations for compared to Atom-JEPA due to the difference in computational overhead. Instead we reuse the hyperparameters found by CKERMT for the architectures. Instead perform tuning in two sweeps: first the L2SP weight decay of the encoder $\{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{-0}\}$ versus L_2 weight decay of the other parameters $\{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{-0}\}$, for a total of 49 iterations. Then the learning rate ($\{1 \cdot 10^{-5}, 2.5 \cdot 10^{-5}, 5 \cdot 10^{-5}, 1 \cdot 10^{-4}, 2.5 \cdot 10^{-4}, 5 \cdot 10^{-4}, 1 \cdot 10^{-3}\}$), the number of epochs $\{100, 50, 25, 10\}$ and number of epochs the Atom-JEPA encoder is frozen for $\{20, 10, 5, 0\}$ for a total of 98 iterations excluding impossible combinations. We accommodate the much larger data size of ChEMBL-MT by setting the encoder freezing and encoder L2SP to 0.

For the baselines and Atom-JEPA we do not use early stopping on a validation set, instead we train on the full training set as we have already selected the number of epochs.. For the literature models Chemprop, CKERMT, and Mol-JEPA we follow their respective described setups. Using a single set of hyperparameters for all datasets reduces model performance, but significantly reduces the computational burden.

Table 27: Intervals and priors for the hyperparameters of the specific feature processor tuned over using Optuna.

Processor Hyperparameter	Interval	Prior
MLP		
Learning rate	$(10^{-4}, 10^{-2})$	Log-uniform
Epochs	$(5, 500)$	Log-uniform
Weight decay	$(10^{-4}, 10^{-1})$	Log-uniform
Head dropout	$(0, 0.5)$	Uniform
Head hidden units	$(32, 256)$	Log-uniform
Trunk hidden units	$\{128, 256, 512, 1024\}$	Categorical
Trunk layers	$\{2, 3, 4, 5\}$	Categorical
Trunk dropout	$(0, 0.5)$	Uniform
Trunk activations	{ReLU, GELU. SiLU}	Categorical
LGBM		
Learning rate	$(5 \cdot 10^{-3}, 3 \cdot 10^{-1})$	Log-uniform
Number of leaves	$(8, 128)$	Log-uniform
Minimum child sample	$(5, 100)$	Log-uniform
Subsample fraction	$(0.5, 1.)$	Uniform
Colsample bytree	$(0.4, 1)$	Uniform
Reguralizer α	$(10^{-8}, 10^{1})$	Log-uniform
Reguralizer λ	$(10^{-8}, 10^{1})$	Log-uniform

Table 28: Intervals and priors for the hyperparameters of the specific feature extractor tuned over using Optuna.

Feature Hyperparameter	Interval	Prior
Morgan		
Radius	$\{2, 3\}$	Categorical
Number of bits	$\{1024, 2048\}$	Categorical
Use Counts	(False, True)	Boolean
RDKit		
–	–	–
ErG		
Fuzz Increase	$(0.1, 0.5)$	Uniform
Avelon		
Use Counts	(False, True)	Boolean
Atom-JEPA		
Node Aggregation	(Mean, Sum)	Categorical
Higher Order Features	(False, True)	Boolean
Feature Scaling	(None, Standard, Quantile)	Categorical
Mol-JEPA		
Feature Scaling	(None, Standard, Quantile)	Categorical

H.1.2 Conformer Generation Details

For datasets where the 3D structure of the compound is not given, we generate conformers following the same setup as Unimol. For completeness we detail the setup here. To convert a SMILE string to a 3D conformer, we import the SMILE string in RDKit. If the importing fails with sanitation we retry without sanitation and afterwards sanitise it.

To get the 3D coordinates we use ETKDGv3 to generate up to 10 conformers. First with chirality enforced, but in case of failure we start from random coordinates. Since Atom-JEPA uses the SE(3) equivariant GNN EquiformerV3, the model can in theory learn to distinguish between enantiomers. The conformers is only used during training as a form of training augmentation.

For several of the ADMET-TDC datasets, the SMILE string contains multiple fragmented components, such as salt forms of CN(C)C(...)Cc1ccc(Cl)c(Cl)c1.[Cl-].[H+]. This is a problem during conformer

Table 29: Molecules in the TDC ADMET benchmark group that RDKit parses as more than one connected component (salts, hydrates, mixtures), over the de-duplicated train_val + test SMILES of each dataset. "Extra atoms" is the mean, over fragmented molecules only, of the share of atoms outside the largest fragment.

Dataset	N	**Fragmented**	**Fragmented** (%)	**Extra atoms** (%)
Caco2 Wang	906	9	1.0	4.6
HIA Hou	578	0	0.0	0.0
Pgb Broccatelli	1212	0	0.0	0.0
Bioavailability Ma	640	1	0.2	52.5
Lipophilicity AstraZeneca	4200	1	0.0	4.9
Solubility AqSolDB	9982	1098	11.0	30.2
BBB Martins	1975	105	5.3	11.7
PPBR AZ	1797	0	0.0	0.0
VDss Lombardo	1111	0	0.0	0.0
CYP2D6 Veith	13130	549	4.2	16.6
CYP3A4 Veith	12328	549	4.5	16.8
CYP2C9 Veith	12092	538	4.4	17.0
CYP2D6 Substrate Carbon-Mangels	664	0	0.0	0.0
CYP3A4 Substrate Carbon-Mangels	667	0	0.0	0.0
CYP2C9 Substrate Carbon-Mangels	666	0	0.0	0.0
Half-life Obach	665	10	1.5	13.2
Clearance Microsome AZ	1102	0	0.0	0.0
Clearance Mepatocyte AZ	1020	0	0.0	0.0
hERG	648	12	1.9	20.8
AMES	7255	0	0.0	0.0
DILI	475	18	3.8	30.0
LD50 Zhu	7342	0	0.0	0.0

relaxation, as the disconnected components can arbitrarily be moved around, thus geometrically ill-defined.

H.1.3 Conformer Scaling

Generation of conformers can be seen as a way of generating multiple 3D views of the same molecule. Previous work (Zhou et al., 2023) generates multiple conformers, specifically 11, as a type of augmentation for training. In this section we investigate the impact on performance these parameters have. To avoid leakage of data, we use the same data for the train and validation split as the hyperparameter tuning. First, we investigate the impact of the number of conformers available during training and test time. Some of the targets have only 7 datapoints in the validation set, so instead of averaging the over the tasks, we have the error over the datapoints. For this experiment we generate up to 50 conformers (from 10) which matches a new conformer seen each epoch. In addition to training conformers, we also investigate the impact ensembling the predictions over conformers can have. We average over 10 seeds for each run. The results are shown in Figure 16. In the figure, we see a large drop in validation MAE when going from training on 1 to 2 conformers, while the run to run variance dominates higher conformer numbers. This shows the importance of having multiple (>1) conformer available during training. Generating many conformers is a good trade-off for many ADMET tasks as cost of generating multiple conformers is comparatively cheap compared to the training time. In addition to training conformers, the figure reveals moderate gains from averaging multiple model predictions from conformers. This can be though of as a way of performing test-time augmentations or ensemble averaging. Curiously , the figure reveals a weak trend of increasing error when ensembling conformer predictions when training on more conformers. An explanation could be that the variance of the model's predictions over conformers is reduced when trained with more conformers, which could negatively impact the ensembles prediction.

We also investigate the impact of the conformer generation quality. Lower potential energy of conformers are typically associated with higher quality as they are more probable. To avoid a co-founding generation artifacts from the relaxation process of MMFF94 and using it to judge the same energies, we instead use the inter-atomic potential model UMA-s-1p2 (Wood et al., 2026) model to predict energy. The results are shown in Figure 17. Surprisingly, the figures shows no correlation

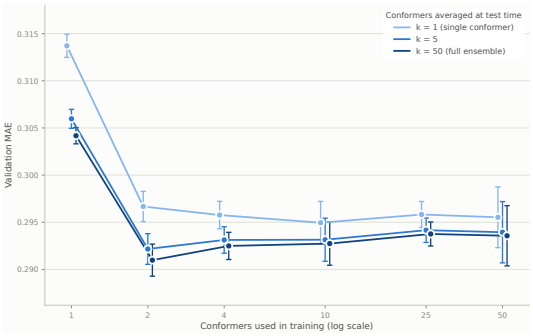


Figure 16: The validation error on the validation error (MAE) against the number of conformers used during training. The different curves represent different numbers of predictions from conformers averaged over during test time (test time augmentations).

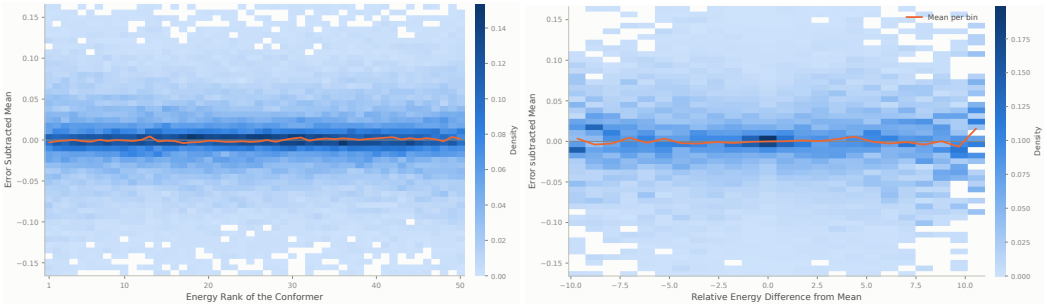


Figure 17: Histogram of the prediction error of validation datapoints versus the energy of the conformer. Left plot is the relative energy of the conformer from the mean, right plot is the rank of the conformer. For none of the plots a strong correlation between error and conformer energy was observed.

between the error of the conformer prediction versus the conformer quality. Only for conformers of very high energy compared to the mean ($\sim 10\%$) we see some small increase in error.

H.1.4 ADMET Parameters for Atom-JEPA

The found hyperparameters from hyperparameter tuning can be found in Table 30.

Table 30: Hyperparameters used for Atom-JEPA fine-tuning on Biogen ADME, ChEMBL-MT, ExpansionRX, and ADMET-TDC. All three start from the same Uni-Mol-pretrained context encoder; encoder architecture is unchanged from Table 26. Every task gets its own MLP head on top of a shared trunk that takes the atom-pooled representation.

Hyperparameter	**Biogen ADME**	**ChEMBL-MT**	**ExpansionRX**
Prediction Head			
Number of tasks	6	25	9
Trunk Architecture	Linear $(768, 256) \to$ SiLU $\to$ Dropout		
Per Task Head Architecture	Linear $(256, 64) \to$ SiLU $\to$ Linear $(64, 1)$		
Shared Trunk dropout	.1		
Pooling readout	Mean		
Higher-order invariants	Yes		
Conformers			
Conformers generated per molecule	10		
Conformers used in training	10		
Objective			
Per-task loss	Huber		
Label standardization	Per-task train mean/std		
Log-transformed targets	None	None	All but LogD
Regularization			
L2-SP coefficient λ	1×10^{-5}	0	1×10^{-5}
Shared Trunk weight decay	.5		
MTL log-σ weight decay	.1		
Encoder stochastic depth	.05		
Encoder normalization layers	Eval mode		
Optimization			
Optimizer	AdamW		
Learning-rate schedule	Cosine + linear warmup		
Encoder learning rate	5×10^{-5}		
Head learning-rate multiplier	10		
MTL log-σ learning rate	5×10^{-5}		
Minimum learning rate	1×10^{-7}		
Warmup epochs	10		
Encoder-only warmup epochs	10		
Frozen-encoder epochs	10	0	10
Batch size	8		
Number of epochs	50		
Gradient clipping norm	10		
Weight EMA decay	.99		

H.2 QM9 FINETUNING

Table 31 summarizes the hyperparameters used for QM9 finetuning. We use three target-specific readouts operating on the encoder's final atom representations following SchNet/SchNetPack-style atom-wise and electronic spatial-extent constructions (Schütt et al., 2018) and a PaiNN-style gated equivariant dipole construction (Schütt et al., 2021), adapted from the GotenNet implementation (Aykent & Xia, 2025). Let C denote the encoder feature width and $\mathbf{r}_i$ the position of atom i. All readouts use sum aggregation of atom-wise contributions.

Dipole moment (μ). Two gated equivariant blocks combine the last layer scalar ($\ell = 0$) and vector ($\ell = 1$) features, with SiLU activations, to predict an atomic scalar q_i and dipole vector $\mathbf{d}_i$. The

molecular dipole magnitude is then

$$\hat{\mu} = \left\| \sum_i (\mathbf{d}_i + q_i \mathbf{r}_i) \right\|_2 .$$

Electronic spatial extent (R^2). A two-layer atomwise MLP with dimensions $C \to C \to 1$ and shifted-softplus activation predicts scalar weights a_i. The molecular prediction is

$$\hat{R}^2 = \sum_i a_i \left\| \mathbf{r}_i - \mathbf{r}_{\mathrm{CM}} \right\|_2^2 ,$$

where $\mathbf{r}_{\mathrm{CM}}$ is the molecular center of mass.

Remaining targets. A two-layer atomwise MLP with dimensions $C \to C \to 1$ and SiLU activation maps scalar atom features to scalar contributions, which are summed to obtain the molecular prediction. For U, U_0, H, and G, summed atomic reference energies are subtracted during training and restored for evaluation, following standard practice. No targets are standardized.

Table 31: Hyperparameters used for QM9 fine-tuning. All targets share the same optimization settings, with target-specific readout architectures for μ and R^2. Encoder architecture hyperparameters are inherited from the pretrained model.

Hyperparameter	All targets
Optimization	
Optimizer	AdamW
Adam ϵ	10^{-7}
Maximum training epochs	1,000
Batch size	32
Encoder base learning rate	10^{-4}
Readout base learning rate	10^{-4}
Weight decay	0
Learning-rate schedule	Linear warmup + ReduceLROnPlateau
Global learning-rate warmup (optimizer steps)	10,000
Learning-rate reduction factor	.8
Learning-rate reduction patience (epochs)	15
Minimum scheduler learning rate	10^{-7}
Initial encoder freezing (epochs)	10
Encoder ramp after unfreezing (epochs)	10
Gradient clipping norm	5
Objective and Model Selection	
Training loss	MSE
Scheduler and early-stopping metric	Validation MSE
Early-stopping patience (epochs)	150
Checkpoint-selection metric	Validation MAE
Reproducibility	
Training seed	42
Split seed	1

H.3 MATBENCH FINETUNING

We use a common training configuration for the five small-to-medium tasks (phonons, dielectric, Log GVRH, Log KVRH and Perovskites), and a second configuration for the three large tasks (MP Gap, MP E Form, and MP Is Metal) as summarized in Table 32. MP Is Metal is the only classification task.

All tasks use an atom-wise readout consisting of a two-layer MLP with SiLU activation with one output for regression or two logits for classification. The final layer has a bias only for classification, and no dropout is used. Atom-wise outputs are mean-pooled, except for phonons, which uses max pooling. Classification probabilities are obtained by applying softmax after pooling.

We standardize targets for all regression tasks and not for classification. Training uses separate encoder and readout learning rates, with linear warmup followed by cosine decay. After the initial encoder-freezing phase, the encoder learning rate is ramped over ten epochs for small-to-medium tasks and 1 epoch for large tasks. We do not use L2-SP regularization for Matbench.

For each official Matbench fold, we reserve 10% of the training fold for validation using split seed 42 and set the PyTorch random seed to 0. Regression normalization statistics are computed from the training fold. We select EMA weights by the lowest validation MAE for regression or highest validation F1 for classification. Scratch baselines use the same architecture and settings, but initialize the encoder randomly and train it from the first epoch without freezing.

Table 32: Hyperparameters used for Matbench fine-tuning. Small-to-medium tasks comprise phonons, dielectric, Log GVRH, Log KVRH and Perovskites; large tasks comprise MP Gap, MP E Form, and MP Is Metal. Classification settings apply only to MP Is Metal. Encoder freezing applies only to pretrained runs; encoders trained from scratch are trainable from the first epoch.

Hyperparameter	**Small-to-medium tasks**	**Large tasks**
Optimization		
Optimizer	AdamW	AdamW
Adam coefficients (β_1, β_2)	$(.9, .95)$	$(.9, .95)$
Adam ϵ	10^{-8}	10^{-8}
Number of epochs	100	12
Training batch size	4	32
Evaluation batch size	32	32
Encoder base learning rate	5×10^{-5}	1.5×10^{-4}
Readout learning-rate multiplier	10	10
Readout base learning rate	5×10^{-4}	1.5×10^{-3}
Learning-rate schedule	Linear warmup + cosine decay	Linear warmup + cosine decay
Global learning-rate warmup (epochs)	10	2
Initial warmup factor	.1	.5
Encoder learning-rate floor	10^{-7}	10^{-7}
Readout learning-rate floor	10^{-6}	10^{-6}
Initial encoder freezing (epochs)	10	2
Encoder ramp after unfreezing (epochs)	10	1
Encoder / readout weight decay	$0/0$	$0/0$
Gradient clipping norm	10	10
Model EMA decay	.99	.99
Objectives and Aggregation		
Regression training loss	MAE	Huber
Classification training loss	—	Weighted cross-entropy
Aggregation of atomwise outputs	Mean max for phonons	Mean