Monroe: A Molecular Foundation Model for In-Context Probabilistic Inference

Błażej Banaszewski
blazej@banaszewski.pl
Graphcore, Bristol, United Kingdom
Max Planck Institute of Biochemistry, Martinsred, Germany

Andrew W. Fitzgibbon
awf@fitzgibbon.ie
Graphcore, Cambridge, United Kingdom

Abstract

Bioassay activity prediction is often data-limited because drug-discovery datasets rely on time-consuming and expensive wet-lab experiments for data generation and evaluation. This challenge has inspired recent research into molecular foundation models (MFMs), which aim to encode general-purpose chemical knowledge into molecular representations that generalize well in data-constrained scenarios. This paper presents Monroe, a new MFM with several innovations over the existing state of the art: increased scale allowing pre-training on over 81 million molecules from the PM6 quantum chemistry dataset; improved graph representation of stereochemistry; improved training losses including conformer denoising and embedding decorrelation; improved multi-task learning; and the use of a prior-data-fitted model (TabPFNv3) for downstream in-context prediction. Our evaluations use a principled pairwise comparison framework that measures statistically significant performance differences. Across established Polaris benchmarks, Monroe matches or exceeds existing MFMs, while on activity cliff benchmarks, designed to assess utility for molecular discovery, it achieves significant improvements over prior methods. Finally, ablation and transfer experiments show that PFN-based downstream predictors also substantially improve two leading existing models, MiniMol and CheMeleon, yielding new state-of-the-art variants we call MiniMol$_{\mathsf{PFN}}$ and CheMeleon$_{\mathsf{PFN}}$, suggesting that our downstream adaptation strategy generalizes beyond Monroe. Source code is at github.com/blazejba/monroe.

1 Introduction

In silico biological property prediction for small molecules is typically performed using wet-lab bioassay datasets whose scales are inherently limited, as biological experiments are expensive, time-consuming, and can only be conducted at scale in a small number of specialized facilities. Traditional machine learning approaches fit a task-specific, low-capacity model directly to small datasets, while more recent approaches seek to learn molecular representations that transfer across tasks, enabling accurate prediction in data-scarce settings. This paradigm, often referred to as *molecular foundation models* (MFMs), aims to extract generalizable chemical and structural knowledge from large, readily available data sources, which can then be adapted to downstream biological tasks. In drug discovery, this is particularly appealing for applications such as virtual screening, where models are repeatedly used to prioritize compounds before committing to further rounds of costly experimental evaluation.

In this paper, we introduce a new MFM, called Monroe,[1] based on a number of innovations over

[1] Named after the computational quantum chemistry pioneer Elizabeth Monroe Boggs.

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

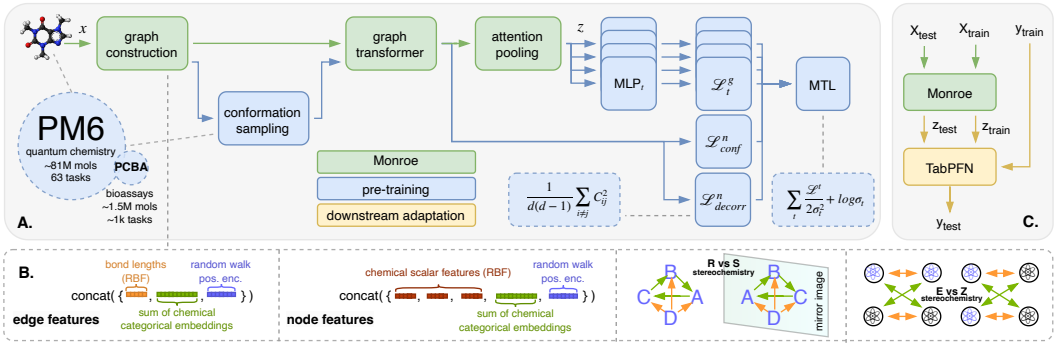

Figure 1: A. Monroe multi-task pretraining with structural and decorrelation losses under uncertainty weighting. B. Graph construction including featurization and stereochemistry-aware rewiring. C. In-context probabilistic inference: embeddings generated by attention pooling of the pretrained model together with training labels are passed to TabPFN, which outputs an approximate posterior-predictive distribution for the test molecule in a single forward pass.

the current state of the art: 1. improved multitask learning using uncertainty-weighted (UW) loss balancing; 2. improved training losses including conformer denoising and embedding decorrelation; 3. improved graph representation of stereochemistry; 4. replacing standard fine-tuning with downstream in-context prediction using a prior-data-fitted model; and 5. scaling training to allow pretraining on the PM6 dataset, comprising semi-empirical quantum properties and quantum chemistry (QC) conformers, and the PCBA dataset, consisting of high-throughput bioassay activity readouts.

We follow recent best practices in model evaluation [1] to make statistically sound comparisons between models, and show that this combination of innovations produces a model which exceeds state-of-the-art performance on standard Polaris benchmarks, and on newer activity-cliff benchmarks [33]. In addition to standard ablations over the design choices, we also augment existing methods with TabPFN-based downstream prediction and obtain two new state-of-the-art variants: $\text{MiniMol}_{\text{PFN}}$ (augmenting MiniMol [13]) and $\text{CheMeleon}_{\text{PFN}}$ (augmenting CheMeleon [5]). $\text{MiniMol}_{\text{PFN}}$ is the closest competitor to Monroe on Polaris but significantly weaker on MoleculeACE, and the opposite is true for $\text{CheMeleon}_{\text{PFN}}$, which is significantly behind on Polaris and closer on MoleculeACE.

2 Related Work

Molecular property prediction. Training objectives for pretraining molecular foundation models span a wide range of approaches: unsupervised [24, 29], multi-task [31, 13, 5], and hybrids that combine molecular property prediction with denoising [22]. Relevant training data sources used by these methods cover a wide range of modalities: bioassays [32], quantum chemical properties [26], geometrical and molecular descriptors [25] and proprietary data like phenomics [31]. Architectures vary as well, including SMILES language models [29], graph transformers [39], and message-passing networks [38, 13, 31]. Our approach derives from GRIT [23], which augments a transformer with a range of inductive biases developed in graph learning, including random-walk positional encodings [6], graph attention mechanisms [34], attention-based global pooling [16, 40], and the use of a virtual node for global information propagation [7]. We rely on TabPFN [8], a prior-data-fitted model that performs amortized in-context prediction and can be interpreted as approximating Bayesian posterior-predictive inference under its training prior.

Multi-task learning. Previous MFM training has cast the problem as multi-task learning (MTL), but has typically used only the simplest MTL methods. The wide variety of task types and losses used in this work requires deeper consideration of MTL techniques, beyond Equal Weighting (EW), namely Uncertainty Weighting (UW) [12], Dynamic Weight Average (DWA) [21], Random Loss Weighting (RLW) [18], and Smooth Tchebycheff (STCH) [20]. Beyond these *scalarization* methods are gradient-based methods that manipulate per-task gradients, but which require T backward passes for T tasks, which becomes prohibitive for T of the order of a thousand as in this work.

Conformational prediction. Uni-Mol+ [22] couples conformation refinement with QC property prediction: it starts from a low-cost RDKit conformer, iteratively updates coordinates toward a DFT

equilibrium structure, and predicts properties from the refined geometry. To reduce the train–test gap, it samples intermediate conformers along a pseudo-linear trajectory between the raw and equilibrium conformations during training. LoQI [27] targets stereochemistry explicitly by augmenting molecular graphs with auxiliary edges for E/Z double bonds and R/S chiral centers, enabling stereochemistry-aware generation.

In our work, we leverage the strengths of both of these approaches, namely: conformation refinement and stereochemistry-awareness, and we explain why the latter is important despite the apparently present stereochemically-disambiguating structural information. We do not adopt an SE(3)-equivariant backbone, reasoning that benchmarks such as TDC [9] are dominated by non-equivariant models, and sequence-only methods like MolFormer [29] remain competitive on the same Polaris tasks we evaluate on, indicating that coordinate-level equivariance is not a prerequisite for strong downstream performance. Stereochemistry, however, clearly matters: Nikitin et al. [27] report that an SE(3)-equivariant baseline distinguishes R/S configurations only 38% of the time without auxiliary stereo edges, rising to 96% once they are added—coordinates alone are not sufficient despite the architecture in principle handling handedness. Coordinate-level equivariance is therefore neither necessary for our downstream targets nor sufficient for stereochemical disambiguation, motivating our choice of an invariant encoder augmented with explicit stereo edges.

3 Methods

Monroe follows established multi-task pretraining practice [3, 31, 13, 5]. As illustrated in Figure 1, the architecture comprises a trunk f_ψ, which is applied to input graph G to produce a "fingerprint" vector $\mathbf{z} = f_\psi(G)$. Each pre-training task t has a dedicated lightweight MLP head on top of the shared encoder. These heads are used only for the pre-training objectives and are discarded thereafter, retaining only the encoder f_ψ for downstream tasks. Pretraining also includes a node-level task of structure denoising, explained later.

3.1 Architecture

We build on GRIT [23], a graph transformer that augments self-attention with random-walk positional encodings and per-edge keys and values; the full layer update is given in Section B.1. Our variant differs from the original in three ways. First, attention is evaluated only on observed edges, rather than on the full $|V| \times |V|$ pair set, and the base graph is symmetrized so message passing is undirected. This yields sparse, $O(|E|)$ attention that scales to molecular graphs. Second, for graph-level representations we use multi-head attention pooling [15], where each head learns to weight nodes differently before aggregation. Third, a virtual node connected to all atoms enables global all-to-all information exchange at each layer, which has been shown to be valuable for accurate structure prediction [10, 22].

Featurization. Node and edge features combine categorical and scalar chemical features from RDKit, input conformers from RDKit, and random-walk positional encodings (see Figure 1B).

Let a molecule be a graph $G = (V, E)$. For each node $i \in V$, let $\{c_{i,k}\}_{k=1}^{K_{\text{node}}}$ denote K_{node} categorical codes, $\mathbf{f}_i \in \mathbb{R}^{F_{\text{node}}}$ continuous RDKit features, and $\mathbf{p}_i \in \mathbb{R}^3$ atom coordinates. Let $P \in \mathbb{R}^{|V|\times|V|}$ be the random-walk transition matrix on the undirected graph with self-loops, and let R be the maximum walk length. Using embedding tables $\text{Emb}_k^{\text{node}} : \mathcal{C}_k \to \mathbb{R}^{d_{\text{emb}}}$, we form

$$\mathbf{u}_i = \sum_{k=1}^{K_{\text{node}}} \text{Emb}_k^{\text{node}}(c_{i,k}), \quad \mathbf{r}_i = \left[(P^1)_{ii}, \ldots, (P^R)_{ii}\right] \in \mathbb{R}^R, \quad \mathbf{x}_i^{(0)} = W_I \left[\mathbf{u}_i \,\|\, \mathbf{f}_i \,\|\, \mathbf{r}_i\right] + W_{Ib}, \tag{1}$$

where $W_I \in \mathbb{R}^{d_h \times (d_{\text{emb}} + F_{\text{node}} + R)}$ and $W_{Ib} \in \mathbb{R}^{d_h}$ project the concatenated input to the encoder hidden dimension d_h.

Edges use categorical bond features together with radial basis encodings of bond lengths $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|_2$ from the input conformer:

$$\phi_m(d_{ij}) = \exp\big(-\gamma(d_{ij} - \mu_m)^2\big), \qquad \boldsymbol{\phi}_{ij} = \left[\phi_1(d_{ij}), \ldots, \phi_M(d_{ij})\right] \in \mathbb{R}^M. \tag{2}$$

Here M is the number of radial basis functions, $\{\mu_m\}_{m=1}^M$ are fixed RBF centers (uniformly spaced over the bond-length range), and γ controls the bandwidth; we set γ based on the spacing between

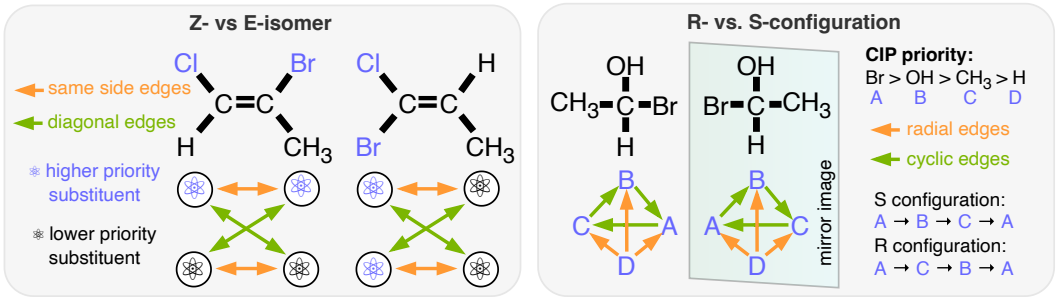

Figure 2: Stereochemical E/Z isomers and R/S configurations.

centers so each basis function covers a comparable length scale. Combining with random walk and edge-categoricals:

$$r_{ij} = \tfrac{1}{2}\left[P^1_{ij} + P^1_{ji}, \ldots, P^R_{ij} + P^R_{ji}\right], \quad u_{ij} = \sum_{k=1}^{K_{\text{edge}}} \text{Emb}^e_k(c_{ij,k}), \quad \mathbf{e}^{(0)}_{ij} = W_e\left[u_{ij} \,\|\, \boldsymbol{\phi}_{ij} \,\|\, r_{ij}\right] + b_e, \tag{3}$$

with K_{edge} edge categorical codes ($K_{\text{edge}} = 4$), $W_e \in \mathbb{R}^{d_e \times (d_{\text{emb}}+M+R)}$ and $b_e \in \mathbb{R}^{d_e}$ projecting to the edge hidden dimension d_e. For more details on featurization see Appendix B.1.

Stereochemistry-aware representation. Our encoder consumes only rotation- and reflection-invariant features (bond lengths and graph attributes). This design choice, while ensuring geometric consistency, creates a fundamental problem: *enantiomers* (mirror-image molecules) and *E/Z isomers* (geometric isomers around double bonds) can collapse to identical inputs, despite having drastically different biological behaviors. Following LoQI [27], we augment the molecular graph with auxiliary edges that encode E/Z and R/S configurations (see Section B.2 for background).

For each double bond with substituents prioritized by CIP rules, let (s_1, s_2) denote the higher-priority substituent on each carbon and (t_1, t_2) the lower-priority ones. We add "parallel" edges connecting same-side substituents and "diagonal" edges connecting opposite-side substituents:

$$\mathcal{E}_{\text{EZ}} = \{(s_1, s_2), (t_1, t_2)\}, \quad \mathcal{E}^{\perp}_{\text{EZ}} = \{(s_1, t_2), (t_1, s_2)\}. \tag{4}$$

The observed E/Z label is attached to $\mathcal{E}_{\text{EZ}}$ and its opposite to $\mathcal{E}^{\perp}_{\text{EZ}}$, ensuring the message-passing network can distinguish both isomers regardless of input ordering (see Figure 2A).

For a tetrahedral stereocenter with CIP-ordered neighbors (a, b, c, d) (highest to lowest priority), we add edges that encode the chiral arrangement:

$$\mathcal{E}_{\text{RS}} = \underbrace{\{(a, d), (b, d), (c, d)\}}_{\text{radial}} \cup \underbrace{\{(a, b), (b, c), (c, a)\}}_{\text{cyclic (S; reversed for R)}}. \tag{5}$$

The radial edges connect the lowest-priority substituent d to all others. The cyclic edges follow the $a \to b \to c \to a$ direction, which is counterclockwise when viewed from d and therefore encodes the S configuration; the reversed cycle $a \to c \to b \to a$ encodes R. Flipping the cycle direction in the graph encoding distinguishes enantiomers (see Figure 2B).

3.2 Training

Task weighting. We balance task losses $\{\mathcal{L}_t\}_{t=1}^T$ via Uncertainty Weighting [12], which assigns each task a learnable standard deviation $\sigma_t > 0$ and minimizes

$$\mathcal{L}_{\text{UW}} = \sum_{t=1}^{T} \left[\frac{\mathcal{L}_t}{2\sigma_t^2} + \log \sigma_t\right]. \tag{6}$$

Tasks with high observation noise (large σ_t) are downweighted automatically via the precision term, while the $\log \sigma_t$ regularizer prevents the trivial solution $\sigma_t \to \infty$. For numerical stability, we

parameterize $\sigma_t = e^{s_t/2}$, giving the equivalent objective $\frac{1}{2}\sum_t [e^{-s_t}\mathcal{L}_t + s_t]$ and we learn $\{s_t\}$ jointly with the encoder.

Conformer denoising and augmentation. Inspired by Uni-Mol+ [22], we train the model to recover high-quality equilibrium structures from noisy estimates (see Figure 1A). Input positions $\mathbf{p}^{\text{in}}$ are sampled by interpolating between a reference $\mathbf{p}^{\text{ref}}$ (PM6-optimized) and a low-cost approximation $\mathbf{p}^{\text{rdk}}$, generated using RDKit's ETKDGv3 [14] followed by MMFF94s force field minimization: $\mathbf{p}_i^{\text{in}} = \eta\,\mathbf{p}_i^{\text{ref}} + (1-\eta)\,\mathbf{p}_i^{\text{rdk}}$. We use $\eta = 1$ with probability 0.1, $\eta = 0$ with probability 0.8, and $\eta \sim \mathcal{U}(0.4, 0.6)$ with probability 0.1.

The model predicts atomic coordinates $\hat{\mathbf{p}}_i \in \mathbb{R}^3$ from last-layer node representations $\mathbf{x}_i^{(L)}$. To handle the rotational and translational invariance of molecular systems, we calculate the loss based on the "best fit" alignment. We analytically compute the optimal rotation matrix R^* and translation vector t^* that align the predicted coordinates to the reference structure $\mathbf{p}^{\text{ref}}$ by minimizing the L_2 error [11]. The loss is then computed on the aligned coordinates:

$$\mathcal{L}_{\text{conf}} = \frac{1}{|V|}\sum_{i=1}^{|V|}\sum_{c=1}^{3}\ell_\delta\big(\big[(R^*\hat{\mathbf{p}}_i + t^*) - \mathbf{p}_i^{\text{ref}}\big]_c\big), \quad R^*, t^* = \underset{R\in\text{SO}(3),\, t\in\mathbb{R}^3}{\text{argmin}} \sum_{i=1}^{|V|}\big\|(R\hat{\mathbf{p}}_i + t) - \mathbf{p}_i^{\text{ref}}\big\|_2^2, \tag{7}$$

where ℓ_δ is the (scalar) Huber loss applied per coordinate component $c \in \{x, y, z\}$. This approach provides a more direct supervision signal than internal distance matrices while maintaining invariance to the prediction reference frame.

Embedding decorrelation. The pooled graph fingerprint $\mathbf{z} \in \mathbb{R}^d$ feeds all T task heads as well as the downstream TabPFN predictor. In order to make the embeddings more tabular-like, we add a covariance penalty that drives the off-diagonal entries of the batch fingerprint covariance to zero, in the spirit of VICReg [2]. For a batch of fingerprints $Z \in \mathbb{R}^{B\times d}$ centered to zero mean, with sample covariance $C = (B-1)^{-1}Z^\top Z$, the loss is

$$\mathcal{L}_{\text{decorr}} = \frac{1}{d(d-1)}\sum_{i\neq j} C_{ij}^2, \tag{8}$$

i.e., the mean squared off-diagonal covariance. This differs from the original VICReg normalization of $1/d$ by an extra factor of $1/(d-1)$, which only rescales the effective weight w. We add $w\,\mathcal{L}_{\text{decorr}}$ to $\mathcal{L}_{\text{UW}}$ outside the UW scalarization, since it acts as a regularizer.

Downstream adaptation. We use the pre-trained encoder f_ψ as a fixed feature extractor (see Figure 1C). For a downstream task with training data $\mathcal{D}_{\text{train}}^{\text{raw}} = \{(G_i, y_i)\}$, we compute embeddings $z_i = f_\psi(G_i) \in \mathbb{R}^d$ to form the feature support set $\mathcal{D}_{\text{train}} = \{(z_i, y_i)\}$. These embeddings are passed to TabPFN, a prior-data-fitted model trained to approximate posterior-predictive inference on tabular datasets. Given a test molecule G_{test} with embedding $z_{\text{test}} = f_\psi(G_{\text{test}})$, TabPFN outputs an approximate posterior-predictive distribution $\tilde{q}(y \mid z_{\text{test}}, \mathcal{D}_{\text{train}})$ in a single forward pass. Since TabPFN is pretrained, this is amortized in-context prediction rather than explicit Bayesian inference performed at test time. This allows for efficient in-context learning: the model adapts to the task solely by conditioning on $\mathcal{D}_{\text{train}}$, eliminating the need for gradient-based fine-tuning or hyperparameter optimization (see Figure 1C).

Datasets. We train on two complementary data sources. The first is the PM6 dataset [26] with labels from Beaini et al. [3], which spans roughly 81 million molecules (see Section B.3); each molecule supplies 62 graph-level semi-empirical quantum-chemical properties, plus the conformer denoising objective described in §3.2. The second is PCBA [3], a panel of binary PubChem bioassays covering 1.56 M molecules: while PM6 supplies dense, large-scale physical signal, PCBA injects sparse but biologically aligned supervision closer in distribution to the downstream Polaris and MoleculeACE tasks, which the ablation in Section 4.4 shows is essential for the model's downstream performance. We thus train on $T = 1152$ tasks: 62 PM6 graph-level targets, 1089 PCBA per-assay binary classifications (see PCBA head below), and 1 conformer denoising objective. Each task loss $\mathcal{L}_t$ is chosen to match the target's domain and natural error structure: we employ Huber loss for signed real-valued quantities, log-Huber for strictly positive scale-like descriptors, Beta NLL for

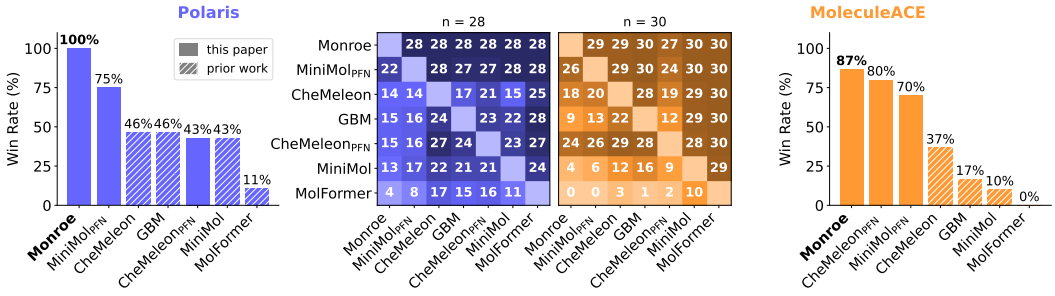

Figure 3: Comparison between {Monroe, MiniMol$_{\text{PFN}}$, CheMeleon$_{\text{PFN}}$ } (methods new to this paper) and {MiniMol, CheMeleon, MolFormer}, the leading previous models, plus a classical QSAR, on 28 tasks from Polaris (left) and 30 tasks from MoleculeACE (right). The pairwise matrices in the middle show the number of tasks that the model on the y-axis won against the model on the x-axis. The barplots show the win rates (see Section 4) in % when all seven methods are taken into account.

bounded ratios, ordinal cross-entropy for integer counts (see Section B.3.1 for full details), and focal binary cross-entropy for PCBA bioassays.

We filter PCBA assays to those with at least 100 positive, 100 negative, and 1000 total non-NaN training labels, retaining 1089 of the original 1328 assays (per-assay statistics and the cutoffs are shown in Figure 7). The PCBA head is a shared linear probe (a single $\mathrm{Linear}(d, K)$ layer with $K = 1089$ logits, one per surviving assay), but each assay is registered as its own task with its own learnable noise scale σ_t, so PCBA contributes K independent terms to the UW sum rather than one. Per-assay class imbalance is handled by focal loss [19] with $\gamma = 2$, $\alpha = 0.25$.

We remove, by InChIKey, all molecules from the pretraining set that overlap with the test set of *any* dataset used for downstream evaluation. Although there is no chance of label leakage, as the tasks differ, we want to avoid a situation where any of the pretraining labels happens to be closely correlated with a downstream one.

Implementation details are given in Section B. Pretraining took 26 hours on four H100 GPUs.

4 Experiments

We evaluate Monroe on two primary problem classes: unseen bioactivity tasks from Polaris (Section 4.1) and activity-cliff benchmarks from MoleculeACE (Section 4.2).

Our evaluation compares seven methods: MolFormer, CheMeleon, MiniMol, and Monroe, as well as novel TabPFN variants of CheMeleon and MiniMol, and GBM, a classical baseline that uses no pretraining: count-based ECFP4 fingerprints with RDKit 2D descriptors, given to a LightGBM tuned per task by randomised search cross-validated on the training split. This is the classical QSAR baseline that remains competitive on the leaderboards, and Praski et al. [28] find that nearly all pretrained embedding models fail to improve on plain ECFP fingerprints. We do not consider MolE [24], Chemprop [38] or MolCLR [36] in our comparisons because the first was shown to be weaker than MiniMol, while the latter two were shown to be weaker than CheMeleon in their respective publications.

Following the evaluation protocol of Ash et al. [1] and Burns et al. [5], we incorporate model variability into the evaluation in order to allow statistically significant evaluation of the models' relative performance, even under multiple sources of variance. This protocol performs simultaneous pairwise comparisons of all models, where a model is considered a "winner" on a specific task if its performance is statistically indistinguishable from the best-performing model (significance level $\alpha = 0.05$, used for all statistical tests across the paper). We aggregate these results to report the total win count and win rate across all tasks. For pairwise comparisons we adopt the Tukey Honestly Significant Difference (HSD) test, as recommended by Ash et al. [1], and apply the Benjamini–Hochberg (BH) procedure [4] to control the False Discovery Rate (FDR) at $\alpha = 0.05$ independently for each benchmark. The same Tukey HSD + BH procedure is used for every pairwise comparison in this paper, including the ablation studies. Figure 3 summarizes the pairwise win counts across both benchmarks.

Additionally we compile a large "leaderboard"-like table for 28 tasks in the Polaris benchmark, where we rank solely based on the mean result, ignoring variance (see Table 8). Although less statistically sound, this may be of interest as an alternative assessment with a wider range of methods.

4.1 Generalization to unseen biological tasks

To evaluate whether the model can generalize from pre-training to unseen biological contexts, we employ a subset of 28 bioactivity tasks from Polaris [37, 9]. These tasks cover ADMET and bioactivity properties, which lets us estimate the model's ability to extrapolate to novel chemical spaces and biological tasks, particularly in the low-data regime.

Monroe is the best method, or is statistically indistinguishable from the best method, on 100% of Polaris tasks (Table 1) and achieves the lowest mean rank (3.71 over the full leaderboard, see Table 8). In addition, MiniMol$_{\text{PFN}}$, a method which is also new to this paper, shows strong performance.

Table 1: Model performance on Polaris and MoleculeACE. Error bars report the mean standard deviation across seeds.

	Polaris		MoleculeACE		
Model	**Win %**	**Rank ↓**	**Win %**	**RMSE ↓**	**Cliff RMSE ↓**
MolFormer	10.7	17.57	0.0	0.748 ± 0.018	0.854 ± 0.027
CheMeleon	46.4	13.25	36.7	0.662 ± 0.016	0.760 ± 0.026
MiniMol	42.9	11.39	10.0	0.703 ± 0.018	0.794 ± 0.031
CheMeleon$_{\text{PFN}}$	42.9	11.46	80.0	0.641 ± 0.013	0.741 ± 0.020
GBM	46.4	10.04	16.7	0.680 ± 0.020	0.769 ± 0.023
MiniMol$_{\text{PFN}}$	75.0	5.82	70.0	0.642 ± 0.004	0.760 ± 0.008
Monroe	**100.0**	**3.71**	**86.7**	**0.629** ± 0.004	**0.737** ± 0.006

4.2 Sensitivity to activity cliffs

MoleculeACE [33] specifically assesses the model's performance on activity cliffs—structurally similar molecules with divergent potencies. The primary metrics are both the overall Root Mean Squared Error (RMSE) and the RMSE on "cliff" compounds (those marking a steep change in activity). The dataset partitioning facilitates this by distributing molecular series across splits, allowing for a targeted evaluation of whether a model can transcend simple determinants of activity.

On MoleculeACE Monroe leads on every metric: lowest overall RMSE (0.629), lowest Cliff RMSE (0.737), and the highest Win % (86.7%, computed on overall RMSE). CheMeleon$_{\text{PFN}}$ and MiniMol$_{\text{PFN}}$ are next, with the original baselines trailing further behind. Figure 4 shows the per-task Cliff RMSE for each method, while Figure 6 gives the per-task pairwise comparisons. The small gap to MiniMol$_{\text{PFN}}$ on Polaris widens substantially on this harder benchmark, suggesting that Monroe's structure-aware pretraining and stereochemistry rewiring may resolve local activity changes that fingerprint-only models miss.

4.3 Downstream adaptation for baseline models

We seek to establish the most effective downstream adaptation strategy for each compared model on 28 Polaris benchmarks. For each model, we compare three distinct strategies:

Native: We use the adaptation method originally proposed for each model (typically a task-specific MLP head trained from scratch or finetuned with gradient descent).

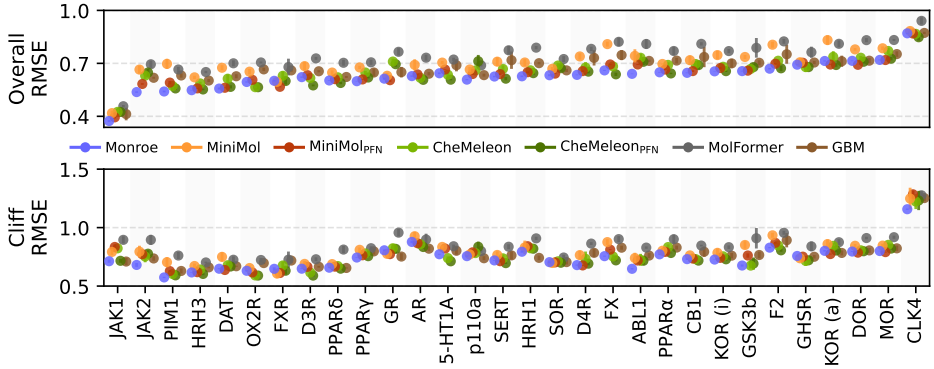

Figure 4: Per-task Cliff RMSE on MoleculeACE for each method.

Pretrained embeddings ($^{\text{PT}}_{\text{PFN}}$): We use the frozen pretrained encoder to extract embeddings for TabPFN without any task-specific weight updates.

Finetuned embeddings ($^{\text{FT}}_{\text{PFN}}$): We finetune the encoder on the downstream task before extracting embeddings for TabPFN. We hypothesize that while finetuning may adapt representations to the specific domain, it may also risk overfitting and potentially degrade the quality of signal for subsequent TabPFN-based inference.

Table 2 reports the within-model win rate for each strategy. The best strategy is model-dependent: MiniMol prefers frozen pretrained embeddings ($^{\text{PT}}_{\text{PFN}}$, 92.9%), CheMeleon prefers finetuned embeddings ($^{\text{FT}}_{\text{PFN}}$, 96.4%), and MolFormer prefers its native MLP head (92.9%).

For MiniMol, using pretrained embeddings with TabPFN substantially outperforms both finetuned embeddings and native adaptation; for CheMeleon, finetuned embeddings with TabPFN edge out the already-strong native adaptation. In both cases the TabPFN-based variant is the best for that model, and we carry the two forward as new methods, MiniMol$_{\text{PFN}}$ (MiniMol $^{\text{PT}}_{\text{PFN}}$) and CheMeleon$_{\text{PFN}}$ (CheMeleon $^{\text{FT}}_{\text{PFN}}$), throughout the remainder of this paper, finding that they become strong additional baselines. MolFormer is the exception: its native MLP-head adaptation outperforms both PFN variants, so we keep MolFormer in its native form.

Table 2: Downstream adaptation comparison on Polaris. In this test we compare within variants of each method, not across methods.

Model	$^{\text{PT}}_{\text{PFN}}$	$^{\text{FT}}_{\text{PFN}}$	Native
MiniMol	**92.9**	85.7	57.1
CheMeleon	75.0	**96.4**	92.9
MolFormer	39.3	71.4	**92.9**

4.4 Ablations

Table 3 shows controlled ablation studies to isolate the contribution of individual components of Monroe. In each instance, alternatives were compared to the finally selected model, again under the Tukey HSD + BH pairwise evaluation protocol.

Table 3: Ablation study on Polaris and MoleculeACE. Pairwise testing is performed within each group separately. Error bars report the mean standard deviation across 3 seeds.

	Polaris		MoleculeACE	
Variant	**Win %**	**Rank ↓**	**RMSE ↓**	**Cliff RMSE ↓**
Multi-task balancing				
UW	**75.0**	**3.71**	0.629 ± 0.003	**0.737 ± 0.005**
EW	67.9	3.71	0.633 ± 0.003	0.743 ± 0.006
RLW	42.9	3.96	0.634 ± 0.004	0.746 ± 0.006
STCH	39.3	4.32	**0.627 ± 0.003**	0.738 ± 0.006
DWA	7.1	10.71	0.680 ± 0.003	0.788 ± 0.005
Conformer denoising				
✓	**92.9**	**3.71**	**0.629 ± 0.003**	**0.737 ± 0.005**
×	60.7	4.61	0.638 ± 0.003	0.749 ± 0.005
Stereochemistry augmentation				
✓	**78.6**	**3.71**	**0.629 ± 0.003**	**0.737 ± 0.005**
×	53.6	3.75	0.633 ± 0.004	0.741 ± 0.007
Decorrelation loss				
✓ **(w=8)**	**92.9**	**3.71**	0.629 ± 0.003	**0.737 ± 0.005**
× (w=0)	67.9	4.21	**0.626 ± 0.004**	0.739 ± 0.005
Dataset				
All states + PCBA	**89.3**	**3.71**	**0.629 ± 0.003**	**0.737 ± 0.005**
No PCBA	32.1	5.36	0.641 ± 0.004	0.750 ± 0.005
S0 only (no excited states)	28.6	5.57	0.651 ± 0.003	0.759 ± 0.004

Multi-task balancing. We compare five scalarization methods: EW, UW, DWA, RLW, and STCH. DWA is clearly harmful, whereas the rest land within noise of one another on both benchmarks. The UW is slightly better, whose per-task learnable variance down-weights noisy or already-saturated tasks, so we choose it for the final variant.

Conformer denoising. We evaluate the contribution of the auxiliary coordinate denoising objective ($\mathcal{L}_{\text{conf}}$) by comparing the full model against a variant trained without it. Removing the objective collapses the Polaris win rate and degrades MoleculeACE on both overall and cliff RMSE. We hypothesize that the 3D refinement signal forces the encoder to internalize geometric constraints that transfer to ADMET prediction.

Stereochemistry augmentation. We assess the importance of our stereochemistry-aware graph rewiring by comparing it against a standard graph representation that ignores E/Z and R/S edges. Its inclusion confers a clear advantage on the Polaris tasks, consistent with observations that many biological tasks are stereoselective, but is equivalent up to experimental variation on MoleculeACE.

Decorrelation loss. We compare the full model with the embedding-decorrelation penalty against a variant without. The penalty confers a decisive win on Polaris, but within-noise impact on MoleculeACE. Our best hypothesis is that TabPFN on the classification tasks in Polaris is enhanced by decorrelation, while regression tasks in general are unaffected.

Dataset. We assess the contribution of the two largest pretraining sources by training (i) on PM6 ground state only (excluding excited and charged-species tasks) and (ii) without the PCBA bioactivity assays. Removing either source degrades both Polaris win rate and MoleculeACE RMSE substantially, indicating that the chosen data sources are critical factor of the achieved performance.

Downstream adaptation strategy. We compare three ways of adapting pretrained encoder to per-task predictions (Table 4). *TabPFN* (the default): the encoder is held fixed and TabPFN performs in-context inference over the embeddings. *MLP decoders*: a task-specific multi-layer perceptron is trained from scratch on the frozen embeddings (the MiniMol recipe). *LoRA→TabPFN*: low-rank adapters in the encoder's attention projections and pooling are finetuned per task through a linear head with a supervised loss; at inference, the head is discarded and TabPFN takes its place.

Using TabPFN with a frozen encoder is a clear winner: adding per-task LoRA adapters on top of the same encoder does not improve over plain in-context inference, and a trained MLP head trails by a wide margin. We read this as evidence that in-context inference with TabPFN absorbs the role that downstream finetuning would otherwise play.

Table 4: Downstream adaptation strategies on Monroe. Win % is computed pairwise within this group.

	Polaris		MoleculeACE	
Strategy	**Win %**	**Rank ↓**	**RMSE ↓**	**Cliff RMSE ↓**
TabPFN(v2)	**100.0**	**3.93**	**0.637** ± 0.002	**0.740** ± 0.003
MLP decoders	28.6	9.04	0.723 ± 0.001	0.806 ± 0.007
LoRA→TabPFN	89.3	4.93	0.653 ± 0.001	0.749 ± 0.003

4.5 Scaffold-split and test-partition evaluation

We test sensitivity of the methods to test-partitioning and generalization to unseen scaffolds. Polaris withholds its test labels, so the split is constructed inside the official training split: molecules are grouped by Bemis–Murcko scaffold and whole groups are assigned to one side, at a fixed 80/20 ratio with five independent repeats per task, with every method evaluated on identical partitions. Two of the 28 tasks (Fang RPPB and Fang HPPB) are excluded because splitting makes their test folds too small to measure. Roughly two thirds of Polaris molecules carry a unique scaffold, so the grouping constrains the assignment less than it might elsewhere. The achieved shift, as the mean nearest-neighbour Tanimoto from a test molecule to the training set, falls from 0.532 under random assignment to 0.443 under scaffold assignment.

Table 5: Polaris under repeated Bemis–Murcko scaffold splits grouped by metric. $1 - \text{MAE}/\sigma_y$ is used for meaningful aggregation across scales. Scores are not comparable to Table 1, which uses the official benchmark split.

Metric	n **tasks**	Monroe	$\text{MiniMol}_{\text{PFN}}$	$\text{CheMeleon}_{\text{PFN}}$
AUROC	8	**0.853** ± 0.034	0.847 ± 0.032	0.811 ± 0.049
AUPRC	5	**0.706** ± 0.035	0.701 ± 0.045	0.625 ± 0.048
$1 - \text{MAE}/\sigma_y$	5	**0.610** ± 0.029	0.583 ± 0.032	0.547 ± 0.038
Pearson r	4	**0.762** ± 0.022	0.741 ± 0.019	0.642 ± 0.026
Spearman r	4	**0.579** ± 0.064	0.558 ± 0.079	0.541 ± 0.078

Monroe obtains the best mean score in every metric group (Table 5), and under the Tukey HSD + BH protocol it is the best method, or statistically indistinguishable from it, on all 26 tasks.

4.6 Downstream data regimes

We also vary how much labelled data is given as support for in-context inference. We subsample the support set to 1, 5, 10, 25 and 50% of the labelled data, holding the test sets at full size. Subsampling is stratified for classification, with a floor of eight molecules.

Table 6: Downstream performance as a function of labelled support-set size.

	Polaris Rank ↓			MoleculeACE RMSE ↓			MoleculeACE Cliff RMSE ↓		
Support	Monroe	$\text{MiniMol}_{\text{PFN}}$	$\text{CheMeleon}_{\text{PFN}}$	Monroe	$\text{MiniMol}_{\text{PFN}}$	$\text{CheMeleon}_{\text{PFN}}$	Monroe	$\text{MiniMol}_{\text{PFN}}$	$\text{CheMeleon}_{\text{PFN}}$
1%	**23.29**	23.79	24.36	1.173 ± 0.019	**1.162** ± 0.030	1.247 ± 0.021	1.195 ± 0.036	**1.181** ± 0.043	1.244 ± 0.032
5%	**20.64**	21.86	23.68	**1.014** ± 0.010	1.015 ± 0.010	1.116 ± 0.006	**1.048** ± 0.016	1.050 ± 0.016	1.140 ± 0.011
10%	**17.54**	19.39	22.96	**0.914** ± 0.008	0.920 ± 0.006	1.022 ± 0.009	**0.968** ± 0.013	0.977 ± 0.007	1.052 ± 0.012
25%	**11.93**	16.29	21.57	**0.786** ± 0.008	0.794 ± 0.006	0.854 ± 0.010	**0.867** ± 0.014	0.880 ± 0.010	0.924 ± 0.009
50%	**7.29**	10.57	18.50	**0.707** ± 0.005	0.717 ± 0.004	0.747 ± 0.004	**0.809** ± 0.008	0.822 ± 0.006	0.851 ± 0.004
100%	**3.71**	5.82	11.46	**0.629** ± 0.001	0.642 ± 0.000	0.641 ± 0.002	**0.737** ± 0.001	0.760 ± 0.001	0.741 ± 0.004

Monroe has the lowest Polaris mean rank at every regime (Table 6). On MoleculeACE it leads on both RMSE and cliff RMSE from 10% upwards, while at 1 and 5% the methods fall within each other's error bars, i.e. below roughly a hundred labelled molecules per task the support set is too small to separate the encoders.

4.7 Model ensembling

The top-ranked entries on the Polaris leaderboard are ensembles rather than single models [31], hence we ask how far ensembling carries Monroe. We pool the per-molecule TabPFN predictions of several models by averaging regressions and class probabilities. We ensemble only models introduced in this paper: members are added by individual Polaris rank, first the variants from the ablation study (k=$\{2, 4, 8\}$, excluding DWA), then our two PFN-augmented baselines MiniMol$_{\mathsf{PFN}}$ (k=9) and CheMeleon$_{\mathsf{PFN}}$ (k=10), where k is the number of models in the ensemble.

Table 7: Prediction-level ensembling of Monroe. Monroe$_{\mathsf{ENS}}(k)$ pools TabPFN predictions from the k strongest models. MiniMol$_{\mathsf{PFN}}$ added at k=9 and CheMeleon$_{\mathsf{PFN}}$ added at k=10.

Method	Polaris Rank ↓	MACE RMSE ↓	MACE Cliff RMSE ↓
Monroe	3.71	0.629	0.737
Monroe$_{\mathsf{ENS}}(2)$	3.18	0.623	0.733
Monroe$_{\mathsf{ENS}}(4)$	<u>2.86</u>	0.620	<u>0.730</u>
Monroe$_{\mathsf{ENS}}(8)$	3.00	0.621	0.732
Monroe$_{\mathsf{ENS}}(9)$	2.71	<u>0.619</u>	0.732
Monroe$_{\mathsf{ENS}}(10)$	**2.68**	**0.615**	**0.726**
MiniMol$_{\mathsf{PFN}}$	5.82	0.642	0.760
CheMeleon$_{\mathsf{PFN}}$	11.46	0.641	0.741

Table 7 reports the result. Pooling a few Monroe variants lowers the mean rank from 3.71 to 2.86 at k=4, after which adding weaker variants gives nothing back: k=8 is no better than k=4. What helps again are the two PFN-augmented baselines, individually much weaker than Monroe but wrong in different places, and the best rank (2.68) and the best activity-cliff RMSE (0.726) are both reached at k=10. The gain is therefore driven by error diversity rather than member quality.

5 Discussion

This paper has introduced a new model for molecular property prediction, with fully open-source training and inference code (see App. B). While it combines several individual innovations rather than a single "big idea", we contend that it is useful in practice, and represents considerable step forward in the incorporation of prior-data-fitted models into the MFM pipeline.

The evaluation follows best current practices, but it is certainly worth noting that several flaws are known [35] in widely used benchmarks such as TDC. These include inconsistent experimental conditions arising from aggregating data across multiple labs, unrealistic dynamic ranges, and varying classification cutoffs. Datasets often suffer from curation errors, such as duplicates with conflicting labels, or contain high rates of assay artifacts. Specific to TDC, we observe instances of chemically incorrect or irrelevant structures, stereochemical ambiguity in SMILES strings, and inconsistent handling of salts. These issues are non-trivial to address and persist even in aggregated benchmarks. Polaris exposes subsets of TDC, making these criticisms directly relevant to parts of our evaluation.

Conformer denoising helps downstream accuracy, and removing it hurts both benchmarks (Section 4.4). Still, Monroe is not a good conformation predictor, and under our training setup its predicted geometries are not better than the input conformers. We don't consider this a shortcoming, since structure determination is not a goal of the model. It arises from the task weighting: equal weighting does learn to denoise, but uncertainty weighting, which we use because it is far stronger downstream, lets the property and bioassay tasks take the priority over denoising.

Separately, Monroe has the lowest RMSE of any method we evaluate on MoleculeACE, on both the full test set and the activity-cliff subset (Table 1). Yet the cliff error still sits well above the overall error, as it does for every method, indicating that the benchmark is not solved. Predicting the sharp, local changes in activity that define cliffs remains an open problem.

6 Societal impact

This work develops machine learning methods for learning molecular representations that can be applied to scientific and industrial tasks such as molecular property prediction and early-stage drug discovery. By improving representation quality, evaluation rigor, and downstream adaptation, the methods introduced here may reduce the cost and time required for exploratory molecular design.

As with many general-purpose modeling tools, these methods could also be applied in other domains involving molecular data, with impacts that depend on the specific context of use. The work does not introduce new biological agents, chemical processes, or experimental protocols, and its potential societal effects arise primarily through downstream applications chosen by practitioners.

References

[1] Jeremy R. Ash, Cas Wognum, Raquel Rodríguez-Pérez, Matteo Aldeghi, Alan C. Cheng, Djork-Arné Clevert, Ola Engkvist, Cheng Fang, Daniel J. Price, Jacqueline M. Hughes-Oliver, and W. Patrick Walters. Practically significant method comparison protocols for machine learning in small molecule drug discovery. *Journal of Chemical Information and Modeling*, 65(18): 9398–9411, 2025. doi: 10.1021/acs.jcim.5c01609. URL https://pubs.acs.org/doi/10.1021/acs.jcim.5c01609.

[2] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

[3] Dominique Beaini, Shenyang Huang, Joao Alex Cunha, Zhiyi Li, Gabriela Moisescu-Pareja, Oleksandr Dymov, Samuel Maddrell-Mander, Callum McLean, Frederik Wenkel, Luis Müller, Jama Hussein Mohamud, Ali Parviz, Michael Craig, Michał Koziarski, Jiarui Lu, Zhaocheng Zhu, Cristian Gabellini, Kerstin Kläser, Josef Dean, Cas Wognum, Maciej Sypetkowski, Guillaume Rabusseau, Reihaneh Rabbany, Jian Tang, Christopher Morris, Ioannis Koutis, Mirco Ravanelli, Guy Wolf, Prudencio Tossou, Hadrien Mary, Therence Bois, Andrew W. Fitzgibbon, Błażej Banaszewski, Chad Martin, and Dominic Masters. Towards foundational models for molecular learning on large-scale multi-task datasets. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Zc2aIcucwc.

[4] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

[5] Jackson W. Burns, Akshat S. Zalte, and William H. Green. Descriptor-based foundation models for molecular property prediction, 2025. URL https://arxiv.org/abs/2506.15792. Introduces CheMeleon.

[6] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations, 2022. URL https://openreview.net/forum?id=wTTjnvGphYj.

[7] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 7 2017. URL http://proceedings.mlr.press/v70/gilmer17a.html.

[8] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

[9] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf 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 *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[10] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure predic-

tion with AlphaFold. *Nature*, 596(7873):583–589, 7 2021. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

[11] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976. doi: 10.1107/S0567739476001873. URL https://doi.org/10.1107/S0567739476001873.

[12] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/papers/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.pdf.

[13] Kerstin Kläser, Błażej Banaszewski, Samuel Maddrell-Mander, Callum McLean, Luis Müller, Ali Parviz, Shenyang Huang, and Andrew Fitzgibbon. MiniMol: A Parameter-Efficient Foundation Model for Molecular Learning. 4 2024. doi: 10.48550/arxiv.2404.14986. URL http://arxiv.org/abs/2404.14986.

[14] Greg Landrum. RDKit: Open-source cheminformatics, 2016. URL http://www.rdkit.org.

[15] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for Attention-based Permutation-Invariant Neural Networks, 10 2018. URL https://arxiv.org/abs/1810.00825.

[16] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph sequence neural networks, 11 2015. URL https://arxiv.org/abs/1511.05493.

[17] Baijiong Lin and Yu Zhang. LibMTL: A Python library for multi-task learning. *Journal of Machine Learning Research*, 24(209):1–7, 2023.

[18] Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W. Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research (TMLR)*, 2022. URL https://openreview.net/forum?id=jjtFD8A1Wx.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[20] Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth tchebycheff scalarization for multi-objective optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 30479–30509. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lin24y.html.

[21] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/papers/Liu_End-To-End_Multi-Task_Learning_With_Attention_CVPR_2019_paper.pdf.

[22] Shuqi Lu, Zhifeng Gao, Di He, Linfeng Zhang, and Guolin Ke. Highly Accurate Quantum Chemical Property Prediction with Uni-Mol+. *arXiv (Cornell University)*, 3 2023. doi: 10.48550/arxiv.2303.16982. URL http://arxiv.org/abs/2303.16982.

[23] Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23321–23337. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ma23c.html.

[24] Oscar Méndez-Lucio, Christos A. Nicolaou, and Berton Earnshaw. MolE: a foundation model for molecular graphs using disentangled attention. *Nature Communications*, 15(1):9431,

2024. doi: 10.1038/s41467-024-53751-y. URL https://www.nature.com/articles/s41467-024-53751-y.

[25] Hirotomo Moriwaki, Yu-Shi Tian, Naoaki Kawashita, and Toshihisa Takagi. Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*, 10(1):4, 2018. doi: 10.1186/s13321-018-0258-y.

[26] Maho Nakata, Tomomi Shimazaki, Masatomo Hashimoto, and Toshiyuki Maeda. PubChemQC PM6: Data sets of 221 million molecules with optimized molecular geometries and electronic properties. *Journal of Chemical Information and Modeling*, 60(12):5891–5899, 2020. doi: 10.1021/acs.jcim.0c00740. URL https://pubs.acs.org/doi/10.1021/acs.jcim.0c00740.

[27] Filipp Nikitin, Dylan M. Anstine, Roman Zubatyuk, Saee Gopal Paliwal, and Olexandr Isayev. Scalable Low-Energy Molecular Conformer Generation with Quantum Mechanical Accuracy, 8 2025. URL https://doi.org/10.26434/chemrxiv-2025-k4h7v.

[28] Mateusz Praski, Jakub Adamczyk, and Wojciech Czech. Benchmarking pretrained molecular embedding models for molecular representation learning, 2025. URL https://arxiv.org/abs/2508.06199.

[29] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022. doi: 10.1038/s42256-022-00580-7. URL https://www.nature.com/articles/s42256-022-00580-7.

[30] James J. P. Stewart. Optimization of parameters for semiempirical methods V: modification of NDDO approximations and application to 70 elements. *Journal of Molecular Modeling*, 13(12):1173–1213, 2007. doi: 10.1007/s00894-007-0233-4. URL https://link.springer.com/article/10.1007/s00894-007-0233-4.

[31] Maciej Sypetkowski, Frederik Wenkel, Farimah Poursafaei, Nia Dickson, Karush Suri, Philip Fradkin, and Dominique Beaini. On the scalability of GNNs for molecular graphs, 2024. URL https://arxiv.org/abs/2404.11568.

[32] Viet-Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. LIT-PCBA: An unbiased data set for machine learning and virtual screening. *Journal of Chemical Information and Modeling*, 60(9):4263–4273, 2020. doi: 10.1021/acs.jcim.0c00155.

[33] Derek van Tilborg, Alisa Alenicheva, and Francesca Grisoni. Exposing the limitations of molecular machine learning with activity cliffs. *Journal of Chemical Information and Modeling*, 62(23):5938–5951, 2022. doi: 10.1021/acs.jcim.2c01073. URL https://pubs.acs.org/doi/10.1021/acs.jcim.2c01073.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1710.10903.

[35] Pat Walters. We need better benchmarks for machine learning in drug discovery, August 2023. URL https://practicalcheminformatics.blogspot.com/2023/08/we-need-better-benchmarks-for-machine.html. Blog post, Practical Cheminformatics.

[36] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022. doi: 10.1038/s42256-022-00447-x. URL https://doi.org/10.1038/s42256-022-00447-x.

[37] Cas Wognum, Lu Zhu, Hadrien Mary, Julien St-Laurent, Larissa, Andrew Quirke, Honoré Hounwanou, roselyn!, Jack Li, Kun Zhu, Shawn Whitfield, Daniel, Jackson Burns, Kira

Howe, Kira Howe (McLean), and felix. polaris-hub/polaris: Version 0.13.0, 2025. URL https://doi.org/10.5281/zenodo.15610218.

[38] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237. URL https://pubs.acs.org/doi/10.1021/acs.jcim.9b00237.

[39] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform bad for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pages 28877–28888, 2021. URL https://proceedings.neurips.cc/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf.

[40] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Neural Information Processing Systems*, 31:4805–4815, 12 2018. URL http://papers.nips.cc/paper/7729-hierarchical-graph-representation-learning-with-differentiable-pooling.pdf.

A Detailed results

A.1 Benchmark details

We compile the leaderboard in Table 8 by joining the results from two separate resources: the Therapeutic Data Commons (TDC) ADMET leaderboard and the Polaris benchmarks. Some methods present on the Polaris boards have no reference to a codebase or paper; we filter those out and consider only methods referenced to an implementation. TDC provides standard deviation numbers across seeds, whereas Polaris only provides a single number per metric, without any error bars. For that reason, in order to consolidate them, we drop the standard deviations from the table and present only the mean results. We replace as-reported results with our own evaluations for Monroe, MiniMol$_{\text{PFN}}$, CheMeleon$_{\text{PFN}}$, MiniMol, MolFormer, and CheMeleon where available.

Table 8: Consolidated leaderboard from Polaris and TDC ADMET benchmarks. **Monroe**$_{\text{ENS}}(10)$ (black) is the ensemble of Table 7. We replace as-reported results with our own evaluations for **Monroe**, **MiniMol**, **MiniMol**$_{\text{PFN}}$, **CheMeleon**, **CheMeleon**$_{\text{PFN}}$ and **MolFormer**.

Dataset	Size	Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CYP2D6 Veith	13,130	AUPRC (↑)	0.801	0.790	**0.749**	0.739	0.739	0.737	**0.726**	0.723	0.721	0.718	**0.716**	**0.705**	**0.698**	0.685	0.673
CYP3A4 Veith	12,328	AUPRC (↑)	0.919	0.916	0.904	0.902	**0.901**	0.895	0.895	**0.884**	0.884	0.881	0.876	0.875	**0.874**	**0.873**	0.867
CYP2C9 Veith	12,092	AUPRC (↑)	0.895	0.859	0.839	**0.835**	0.833	0.829	**0.826**	**0.825**	**0.816**	**0.796**	0.789	0.786	0.783	0.777	**0.776**
Solubility AqSolDB	9,982	MAE (↓)	**0.657**	**0.681**	**0.719**	0.726	0.743	**0.744**	0.761	0.775	0.776	0.789	0.792	0.796	**0.796**	0.827	0.828
LD50 Zhu	5,907	MAE (↓)	**0.533**	**0.547**	0.552	**0.555**	0.573	**0.576**	0.588	**0.593**	0.606	**0.612**	0.614	0.614	0.621	0.622	0.625
Ames	5,821	AUROC (↑)	**0.881**	**0.875**	0.871	0.870	0.869	0.868	0.868	**0.866**	0.866	**0.865**	**0.862**	0.852	0.850	**0.847**	0.847
Lipophilicity AZ	3,360	MAE (↓)	0.392	**0.397**	**0.399**	0.411	0.424	**0.444**	**0.463**	**0.467**	0.467	0.470	0.479	**0.492**	**0.508**	0.512	0.515
PPBR AZ	2,231	MAE (↓)	**6.589**	**6.801**	7.004	7.026	**7.098**	**7.235**	7.440	7.526	**7.599**	7.660	7.788	7.914	7.990	8.200	8.288
Fang HCLint	2,229	Pearson (↑)	**0.791**	**0.784**	0.778	**0.742**	**0.727**	0.720	0.717	**0.710**	**0.707**	**0.695**	0.674				
Fang RClint	2,218	Pearson (↑)	0.798	**0.786**	0.784	**0.775**	**0.746**	**0.735**	**0.729**	**0.720**	0.714	**0.709**	0.698	0.660			
Fang Perm	1,919	Pearson (↑)	0.879	**0.866**	0.860	**0.851**	**0.824**	**0.808**	**0.795**	**0.791**	**0.773**	0.772	0.762	0.714			
BBB Martins	1,624	AUROC (↑)	**0.936**	0.933	**0.932**	0.928	**0.922**	0.920	**0.920**	0.916	0.915	0.913	0.912	0.910	0.908	0.906	0.905
Fang Solu	1,578	Pearson (↑)	0.770	0.764	**0.734**	**0.730**	0.729	**0.716**	**0.672**	0.654	0.651	**0.651**	0.632	**0.609**	**0.580**		
Pgp Broccatelli	973	AUROC (↑)	0.954	**0.946**	**0.944**	**0.942**	0.942	**0.941**	0.938	0.935	0.930	0.929	0.929	0.928	0.926	0.923	0.922
Clearance Hepatocyte AZ	970	Spearman (↑)	0.538	0.536	0.525	**0.503**	0.498	0.466	**0.463**	**0.452**	0.447	0.440	0.440	0.439	0.431	**0.431**	0.430
VDss Lombardo	904	Spearman (↑)	0.713	0.707	**0.700**	**0.700**	**0.684**	**0.680**	**0.639**	0.637	0.628	0.627	**0.615**	0.609	0.600	0.582	0.561
Clearance Microsome AZ	881	Spearman (↑)	0.683	**0.680**	**0.678**	0.653	0.630	**0.628**	0.626	0.625	0.625	**0.620**	**0.607**	0.599	0.597	0.586	0.585
Caco2 Wang	728	MAE (↓)	0.256	**0.257**	**0.274**	0.274	0.276	0.282	0.285	0.287	0.287	0.289	0.297	**0.302**	0.316	0.319	0.321
HIA Hou	578	AUROC (↑)	**0.996**	**0.994**	0.990	0.989	0.988	**0.987**	0.986	**0.985**	0.985	0.981	0.981	0.978	**0.978**	0.975	0.974
CYP3A4 Substrate	535	AUROC (↑)	0.696	**0.691**	0.685	**0.685**	0.667	**0.664**	0.662	0.655	**0.655**	0.650	0.647	0.640	**0.640**	0.639	0.633
CYP2C9 Substrate	534	AUPRC (↑)	0.478	**0.462**	**0.457**	**0.444**	0.441	0.437	0.437	0.433	**0.432**	**0.428**	0.417	0.417	0.415	0.407	0.400
CYP2D6 Substrate	532	AUPRC (↑)	0.736	**0.735**	0.731	**0.727**	**0.726**	0.720	0.713	0.711	0.704	0.704	0.686	0.685	0.680	0.677	**0.673**
Half Life Obach	532	Spearman (↑)	**0.639**	0.625	**0.617**	**0.604**	0.576	0.573	0.562	0.557	0.547	0.544	**0.532**	0.485	**0.469**	0.438	0.392
hERG	523	AUROC (↑)	0.891	0.880	0.875	0.874	0.871	**0.864**	0.856	**0.856**	0.855	**0.845**	0.841	0.840	0.836	0.825	**0.823**
Bioavailability Ma	512	AUROC (↑)	0.748	0.742	0.730	0.729	0.728	**0.715**	0.714	**0.714**	0.706	**0.699**	**0.684**	0.675	0.673	0.672	0.667
DILI	379	AUROC (↑)	**0.952**	**0.951**	0.933	**0.931**	0.925	0.925	**0.924**	0.919	0.919	0.917	0.909	0.906	**0.901**	0.899	0.887
Fang HPPB	126	Pearson (↑)	0.888	0.884	**0.853**	**0.822**	0.815	**0.787**	0.774	**0.774**	**0.765**	**0.736**	**0.656**	0.636	0.635		
Fang RPPB	111	Pearson (↑)	0.908	0.886	**0.860**	**0.847**	**0.776**	**0.774**	0.747	**0.730**	**0.709**	0.680	0.614	**0.487**			

A.2 Per-task pairwise comparisons

The win-rate counts reported in the main text aggregate the same pairwise tests applied per task. For visual inspection of those underlying tests, we provide one panel per task showing each method's mean and 95% confidence interval (5 seeds per method). Confidence intervals use the pooled within-group variance from the equal-variance assumption, so within a panel all error bars share the same length. The dashed vertical lines mark the 95% CI of the best method (the practical-equivalence band). The set of methods shown in blue, gray, or amber on a given task is exactly the set that contributes to that task's win count.

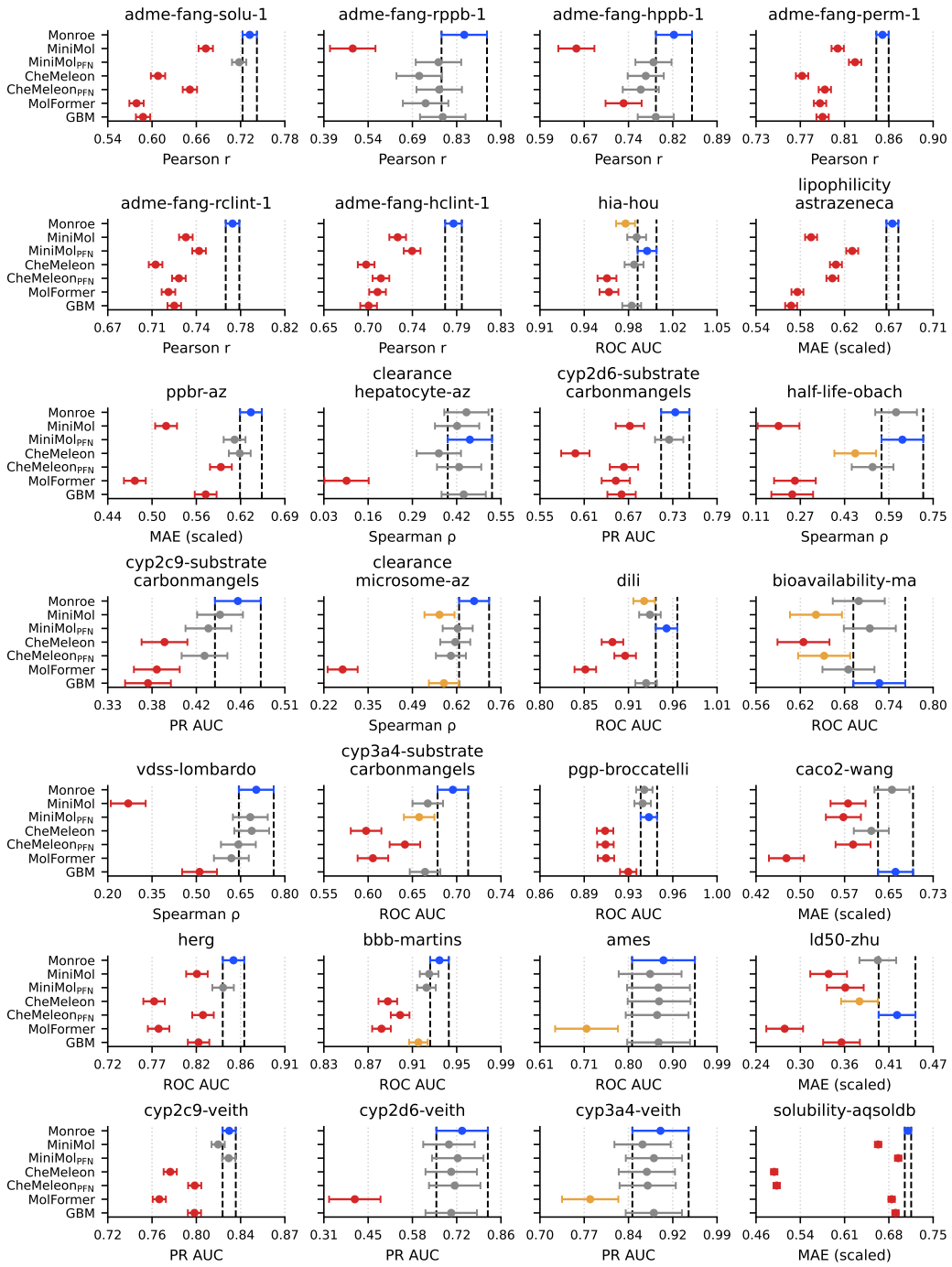

Figure 5: Per-task pairwise comparisons on 28 ADMET tasks from Polaris. Error bars are 95% confidence intervals; interval overlap corresponds to an *uncorrected* pairwise comparison. The colour reflects the multiplicity-corrected verdict against the best method (Tukey HSD followed by Benjamini–Hochberg FDR): **blue** marks the best-mean method on the task (a winner); **gray** marks methods tied with the best (also winners, counted as ties for first) whose interval overlaps the best's; **amber** marks methods that are also counted as winners under FDR correction but whose interval does *not* overlap the best's, meaning the intervals separate but the correction still classifies them as tied; **red** marks methods significantly worse than the best (losers). Dashed vertical lines mark the best method's 95% CI; all error bars in a panel are equal-length under the pooled-variance assumption. MAE-based tasks are reported as scaled MAE ($1 - \mathrm{MAE}/\mathrm{scale}$, see [1]) so that higher is better across all panels.

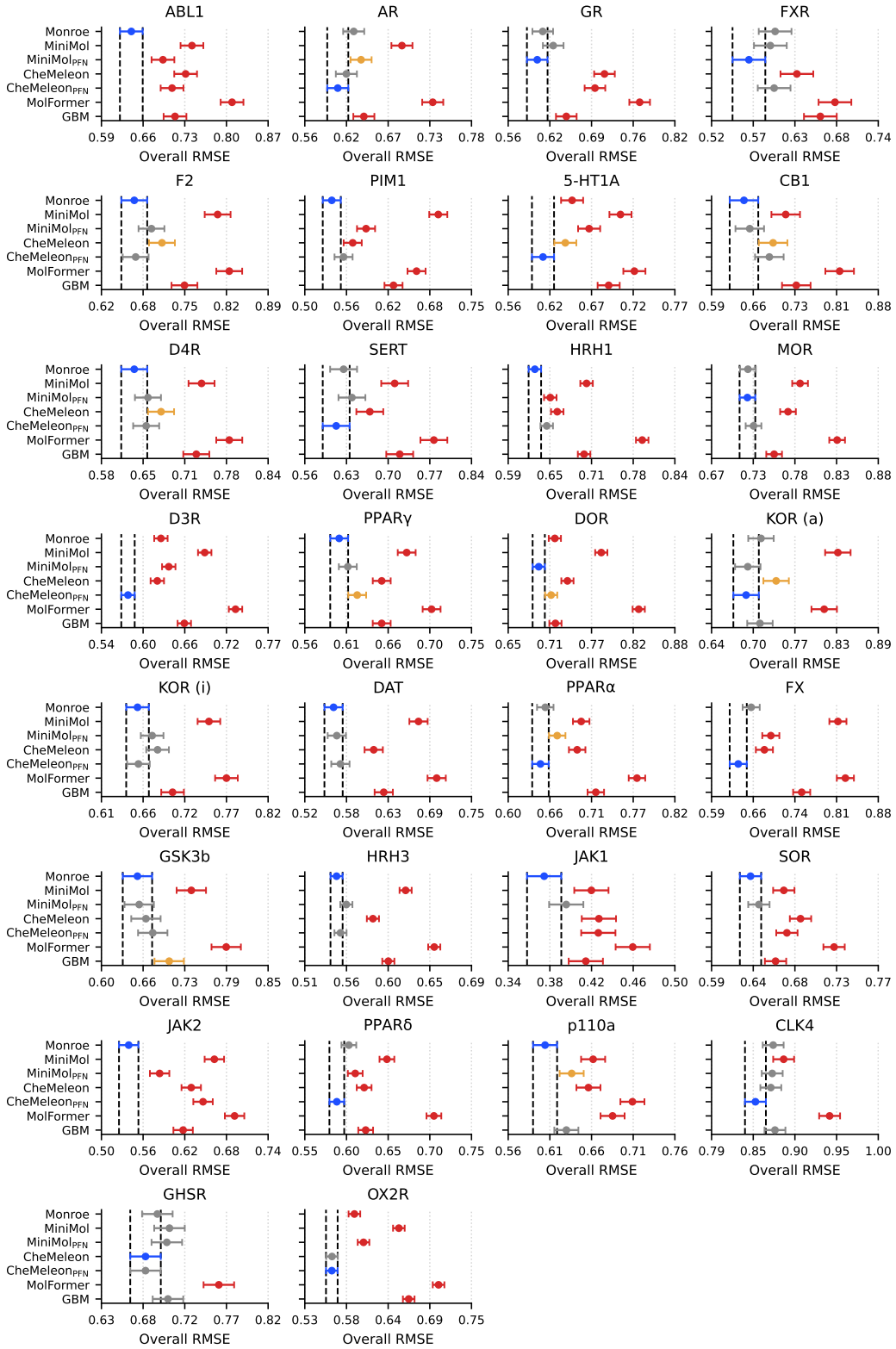

Figure 6: Per-task pairwise comparisons on 30 activity-cliff tasks from MoleculeACE. Same color and CI conventions as Figure 5: **blue** = best on this task, **gray** = tied with the best and interval overlaps, **amber** = tied with the best under FDR correction but interval does not overlap, **red** = significantly worse than the best. Each panel reports the overall test RMSE on one dataset (lower is better, so the best-mean method has the smallest RMSE).

B Implementation details

The architecture of the encoder is summarized below:

```
================================================================================================================================
ENCODER ARCHITECTURE
================================================================================================================================
  GritTransformer(
  (edge_rbf): RBF()
  (node_oh): ModuleDict(
    (valence_implicit): Embedding(14, 128)
    (valence_total): Embedding(14, 128)
    (hybridization): Embedding(9, 128)
    (atomic_number): Embedding(44, 128)
    (total_num_hs): Embedding(10, 128)
    (is_aromatic): Embedding(2, 128)
    (is_in_ring): Embedding(2, 128)
    (chirality): Embedding(3, 128)
    (period): Embedding(8, 128)
    (degree): Embedding(14, 128)
    (group): Embedding(20, 128)
  )
  (edge_oh): ModuleDict(
    (conjugation): Embedding(2, 128)
    (is_in_ring): Embedding(2, 128)
    (bond_type): Embedding(5, 128)
    (stereo): Embedding(9, 128)
  )
  (node_proj): Linear(in_features=156, out_features=720, bias=True)
  (edge_proj): Linear(in_features=180, out_features=720, bias=True)
  (rwpe_node_bias_proj): Linear(in_features=20, out_features=720, bias=False)
  (rwpe_edge_bias_proj): Linear(in_features=20, out_features=720, bias=False)
  (grit_layers): ModuleList(
    (0-9): 10 x GritTransformerLayer(in_channels=720, out_channels=720, heads=10)
    [GritTransformerLayer(
      (attention): MultiHeadAttentionLayerGritSparse(
        (dropout): Dropout(p=0.0, inplace=False)
        (Q): Linear(in_features=720, out_features=720, bias=True)
        (K): Linear(in_features=720, out_features=720, bias=True)
        (E): Linear(in_features=720, out_features=1440, bias=True)
        (V): Linear(in_features=720, out_features=720, bias=True)
      )
      (O_h): Linear(in_features=720, out_features=720, bias=True)
      (O_e): Linear(in_features=720, out_features=720, bias=True)
      (batch_norm1_h): BatchNorm1d(720, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (batch_norm1_e): BatchNorm1d(720, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (batch_norm2_h): BatchNorm1d(720, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (FFN_h_layer1): Linear(in_features=720, out_features=1440, bias=True)
      (FFN_h_layer2): Linear(in_features=1440, out_features=720, bias=True)
    )]
  )
  (pooling): MultiHeadAttentionPooling(
    (gate_dropout): Dropout(p=0.05, inplace=False)
    (gate): Linear(in_features=720, out_features=10, bias=True)
    (value_proj): Linear(in_features=720, out_features=720, bias=False)
  )
)
================================================================================================================================
PARAMETER COUNT
================================================================================================================================
Node embeddings:            17.9K
Edge embeddings:             2.3K
Input projections:         272.2K
Transformer layers:        57.68M
Pooling:                   525.6K
--------------------------------
TOTAL:                     58.50M
```

Monroe is implemented in PyTorch with minimal dependencies, and adapts methods from LibMTL [17]. Optimization of the MTL objective using Adam on 81 million molecules took 26 hours on four H100 GPUs. The entire preprocessing, training, and evaluation pipelines are publicly available. A focus on efficiency of implementation means that it is practical for other researchers to consider adaptations of the pretraining losses and architectures in future research. The code is available at: github.com/blazejba/monroe.

B.1 Architectural details

GRIT layer update. For each layer and head h, given nodes $\mathbf{x}_i \in \mathbb{R}^{d_h}$ and edges $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$, our

sparse GRIT layer produces $\mathbf{x}_i^+$ and $\mathbf{e}_{ij}^+$ via

$$\begin{aligned}
\mathbf{q}_i^h,\ \mathbf{k}_j^h,\ \mathbf{v}_j^h &= W_Q^h \mathbf{x}_i,\ \ W_K^h \mathbf{x}_j,\ \ W_V^h \mathbf{x}_j, \\
\mathbf{s}_{ij}^h &= \varphi\big((\mathbf{q}_i^h \odot \mathbf{k}_j^h) \odot W_E^h \mathbf{e}_{ij}\big) + W_{Eb}^h \mathbf{e}_{ij}, \\
\boldsymbol{\alpha}_{ij}^h &= \mathrm{softmax}_{j\in\mathcal{N}(i)}\Big(d_h^{-1/2} \cdot (\mathbf{w}_{\text{attn}}^h)^\top \mathbf{s}_{ij}^h\Big), \\
\mathbf{m}_i^h &= \sum_{j\in\mathcal{N}(i)} \boldsymbol{\alpha}_{ij}^h \big(\mathbf{v}_j^h + W_{ev}^h \mathbf{s}_{ij}^h\big), \\
\mathbf{x}_i^+,\ \mathbf{e}_{ij}^+ &= W_O\big[\|_{h=1}^H \mathbf{m}_i^h\big],\ \ W_{Eo}\big[\|_{h=1}^H \mathbf{s}_{ij}^h\big],
\end{aligned} \tag{9}$$

with $\varphi(z) = \mathrm{sign}(z)\sqrt{|z| + \epsilon}$ and learned attention vectors $\mathbf{w}_{\text{attn}}^h \in \mathbb{R}^{d_h}$. We then apply degree-based scaling $\mathbf{x}_i^+ \leftarrow \lambda_1 \mathbf{x}_i^+ + \lambda_2 \mathbf{x}_i^+ \log(\deg_i + 1)$, residual connections, batch normalization, and a two-layer FFN. The neighborhood $\mathcal{N}(i)$ is restricted to observed (and stereochemistry-augmented) edges plus the virtual-node edge, giving $O(|E|)$ attention rather than the dense $O(|V|^2)$ used in the original GRIT.

Node features. Each atom v has two groups of features: (i) eight continuous scalars: number of radical electrons, electronegativity, first ionization energy, covalent radius, melting point, formal charge, van der Waals radius, and atomic weight; and (ii) eleven categorical codes: implicit valence, total valence, hybridization, element symbol, total number of attached hydrogens, aromaticity, ring membership, chirality (R/S), period, degree, and group. Each categorical feature has a fixed vocabulary, with unseen values mapped to a dedicated *other* category.

Edge features. Each bond (u, v) is assigned four categorical codes: conjugation, ring membership, bond type (single/double/triple/aromatic), and stereochemistry (none/any/E/Z/cis/trans). The base molecular graph is treated as undirected (each bond is inserted in both directions). When stereochemical augmentation is enabled, additional edges are inserted to encode E/Z double bonds and R/S chiral centers; these edges use two auxiliary stereo categories and map the remaining bond attributes to the *other* category. Unlike base bonds, the R/S radial and cyclic edges (Eq. 5) are inserted in a single direction so that the cycle orientation $a{\to}b{\to}c{\to}a$ vs. $a{\to}c{\to}b{\to}a$ is preserved as topology, allowing the encoder to distinguish enantiomers.

Virtual node and degree scaling. Beyond the node and edge inputs constructed in Eqs. (1)–(3), each graph receives one extra virtual node with a dedicated categorical token and zero continuous/positional features, plus edges connecting every atom to this virtual node. Within each layer, $\log(\deg + 1)$ is additionally used as a degree-based scaling signal applied to the GRIT layer output.

B.2 Stereochemistry details

Stereochemistry refers to the three-dimensional arrangement of atoms in molecules. Two molecules with identical connectivity but different spatial arrangements are *stereoisomers*. This section explains the two key types of stereoisomerism that our model encodes.

CIP priorities. The Cahn–Ingold–Prelog (CIP) rules provide a consistent ordering of substituents: (1) rank by atomic number of the directly attached atom (higher = higher priority); (2) for ties, compare next atoms outward until a difference is found; (3) treat double bonds as duplicated single bonds (e.g., C=O counts as C bonded to two oxygens). This ordering underlies both R/S and E/Z nomenclature.

R/S chirality (tetrahedral stereocenters). A carbon with four *distinct* substituents is a stereocenter. After ranking substituents as $a{>}b{>}c{>}d$ by CIP priority, orient the molecule so that the lowest-priority group d points away from you (like looking down an axle toward the back wheel of a car). Trace $a{\to}b{\to}c$: clockwise defines **R** (Latin *rectus*, right), counterclockwise defines **S** (Latin *sinister*, left). Enantiomers—non-superimposable mirror images—have opposite R/S configurations at every stereocenter.

E/Z configuration (geometric isomers). Double bonds prevent free rotation, so substituents on each carbon can be locked on the same or opposite sides. For each double-bond carbon, identify the

higher-priority substituent via CIP. **Z** (German *zusammen*, together): both higher-priority substituents lie on the *same* side. **E** (German *entgegen*, opposite): higher-priority substituents lie on *opposite* sides. E and Z isomers are diastereomers—stereoisomers that are *not* mirror images.

Why invariant encoders need augmentation. A rotation-invariant encoder maps 3D coordinates to scalar distances, erasing chirality: a molecule and its mirror image produce identical distance matrices. Similarly, E/Z bond geometry may be noisy or absent in approximate conformers. Our auxiliary edges convert stereochemical labels into *graph topology*:

- **E/Z:** "parallel" edges link same-side substituents (stereo-defining pair); "diagonal" edges link opposite-side substituents (contrastive pair). Labeling one set with the observed configuration and the other with its complement ensures the network receives a unique signal for each isomer.
- **R/S:** radial edges from the lowest-priority substituent d to the others, plus a directed cycle $a{\to}b{\to}c{\to}a$ that encodes the S configuration (counterclockwise when viewed from d); the reversed cycle $a{\to}c{\to}b{\to}a$ encodes R.

Because these signals are discrete edge labels rather than continuous distances, they survive rotation invariance and let message passing distinguish stereoisomers that would otherwise be indistinguishable.

B.3 PM6 dataset details

PM6 (Parametric Method 6) is a semi-empirical quantum chemistry method that solves an approximation of the Schrödinger equation [30]. Unlike *ab initio* Density Functional Theory (DFT) methods such as B3LYP, which calculate electron interactions explicitly, PM6 uses empirical parameters derived from experimental data to simplify the Hamiltonian. This approximation makes PM6 orders of magnitude faster than DFT, enabling calculations on the scale of tens of millions of molecules—a scale that would be computationally prohibitive with higher-level theory. Despite its lower precision, Nakata et al. [26] showed that PM6-optimized geometries and electronic properties (e.g., HOMO/LUMO energies) exhibit strong correlation ($R^2 \approx 0.9$) with those obtained from B3LYP/6-31G* calculations, providing a high-quality signal for large-scale pre-training.

We use the PM6_83M dataset derived from the PubChemQC project [26], as curated by Beaini et al. [3] for the UltraLarge benchmark. The raw release contains 83 million unique molecules; after deduplication, removal of molecules whose InChIKey overlaps the test split of any downstream benchmark, and discarding featurization failures, we retain roughly 81 million molecules for pretraining—the figure cited throughout the paper. Since calculations are performed for four different molecular states—ground state (S0), lowest energy triplet excited state (T0), cation, and anion—the raw dataset comprises a total of 221M semi-empirical PM6 computations. This represents the largest available dataset for molecular representation learning in terms of unique chemical structures.

The pre-training objective uses 62 graph-level targets per molecule. These include quantum properties like Alpha HOMO, Alpha HOMO-LUMO gap, Beta HOMO, and Total energy across the available states (S0, T0, cation, anion). Additionally, we predict 3D descriptors computed from the PM6-optimized conformations, including Spherocity, Plane of best fit, and Principal length along the three main axes. Node-level targets (atomic Mulliken charges and electronic spins across the four states) are also available in the source data, but we observed that including them degraded downstream performance under our training conditions, so they are excluded from training.

B.3.1 Loss functions for heterogeneous PM6 targets

The PM6 objective spans heterogeneous quantities with different supports and error structures: some targets are signed and live on $\mathbb{R}$ (e.g., energies, orbital levels, Mulliken charges), others are strictly positive scale parameters (e.g., lengths and molecular size descriptors), others are bounded ratios in $[0, 1]$ (e.g., spherocity, QED, FSP3), and several are small bounded integer counts (e.g., ring counts). A single regression loss is therefore a poor inductive bias; instead, we choose the loss *per target* to match (i) the target's natural domain constraints and (ii) the most appropriate notion of error (absolute vs. relative, additive vs. multiplicative, continuous vs. discrete).

Huber loss for signed real-valued targets. For signed targets where an *absolute* deviation is meaningful, we use the Huber loss. Let $r = \hat{y} - y$ denote the residual. With threshold $\delta > 0$,

$$\mathcal{L}_{\text{Huber}}(y, \hat{y}; \delta) = \begin{cases} \frac{1}{2} r^2, & |r| \le \delta, \\ \delta \left(|r| - \frac{1}{2}\delta \right), & |r| > \delta. \end{cases} \tag{10}$$

PM6-derived targets can contain rare heavy-tailed errors. Huber preserves the benefits of MSE near the optimum while limiting the influence of outliers, which is particularly important in a multi-target setting.

Log-Huber loss for strictly positive scale-like targets. For strictly positive quantities where *relative* error is the natural scale (e.g., lengths and size descriptors), we apply Huber in log-space. Using a small $\varepsilon > 0$, the network output $\hat{y}$ is interpreted as a log-domain prediction and compared to the log-transformed target $y^{(\log)} = \log(y + \varepsilon)$:

$$\mathcal{L}_{\text{log-Huber}}(y, \hat{y}; \delta, \varepsilon) = \mathcal{L}_{\text{Huber}}\left(y^{(\log)}, \hat{y}; \delta \right). \tag{11}$$

Log-space converts multiplicative/scale-dependent noise into approximately additive noise, making optimization more uniform across magnitudes and ensuring positivity after exponentiation.

Log-ratio squared loss for spherocity. Spherocity is a bounded shape descriptor ($[0, 1]$) defined through ratios of geometric quantities. We therefore penalize squared error in log-ratio space:

$$\mathcal{L}_{\text{log-ratio}}(s, \hat{s}; \varepsilon) = \left(\log(\hat{s} + \varepsilon) - \log(s + \varepsilon) \right)^2, \qquad s \in [0, 1]. \tag{12}$$

The log transform makes multiplicative deviations symmetric (e.g., $\times 2$ and $\times \frac{1}{2}$), which aligns with the ratio nature of geometric descriptors and yields stable gradients.

Beta negative log-likelihood for bounded ratios/scores. For bounded ratios/scores in $(0, 1)$, we use a Beta likelihood and clip targets to avoid boundary issues:

$$\tilde{y} = \text{clip}(y, \varepsilon, 1 - \varepsilon). \tag{13}$$

We parameterize $\text{Beta}(\alpha, \beta)$ via mean $\mu \in (0, 1)$ and concentration $\kappa > 0$:

$$\begin{aligned} &\mu = \text{clip}(\sigma(z_\mu), 0.01, 0.99), \qquad \kappa = \text{clip}(\text{softplus}(z_\kappa) + 2, 2, 200), \\ &\alpha = \mu\kappa, \qquad \beta = (1 - \mu)\kappa, \end{aligned} \tag{14}$$

and compute the negative log-likelihood

$$\mathcal{L}_{\text{BetaNLL}}(\tilde{y}; \alpha, \beta) = -(\alpha - 1) \log \tilde{y} - (\beta - 1) \log(1 - \tilde{y}) + \log \text{B}(\alpha, \beta), \tag{15}$$

where $\log \text{B}(\alpha, \beta) = \log \Gamma(\alpha) + \log \Gamma(\beta) - \log \Gamma(\alpha + \beta)$. Since Beta models a *density* on $(0, 1)$, $\mathcal{L}_{\text{BetaNLL}}$ can be negative when the predicted distribution is highly concentrated near $\tilde{y}$. For numerical stability and to ensure a bounded loss range, we clamp the concentration parameters,

$$\alpha \leftarrow \text{clip}(\alpha, \alpha_{\min}, \alpha_{\max}), \qquad \beta \leftarrow \text{clip}(\beta, \beta_{\min}, \beta_{\max}), \tag{16}$$

with $\alpha_{\min} = \beta_{\min} = 1$, $\alpha_{\max} = \beta_{\max} = 100$, and target-clipping threshold $\varepsilon = 10^{-4}$. Clamping bounds the most negative achievable NLL under our parameterization. To obtain a positive task loss for uncertainty weighting, we apply softplus to the mean Beta NLL over the observed labels in the batch:

$$\mathcal{L}_{\text{Beta}} = \text{softplus}\left(\overline{\mathcal{L}}_{\text{BetaNLL}} \right). \tag{17}$$

Ordinal loss for bounded integer counts. For bounded counts, we use an ordinal formulation that preserves ordering. Let $y \in \{m, m + 1, \ldots, M\}$ and define $y' = y - m \in \{0, 1, \ldots, K\}$ with $K = M - m$. For $k = 1, \ldots, K$, define $t_k = \mathbb{1}[y' \ge k]$. The model outputs logits a_k and predicts

$p_k = \sigma(a_k) \approx \Pr(y' \geq k)$. The ordinal loss sums binary cross-entropies:

$$\mathcal{L}_{\text{ord}}(y, \{a_k\}_{k=1}^{K}) = \sum_{k=1}^{K} \text{BCE}(t_k, \sigma(a_k)) \,. \tag{18}$$

These targets are discrete and bounded, and the ordinal construction respects their ordering, penalizing near-miss errors less than large deviations while avoiding the mismatch of unconstrained real-valued regression.

B.4 PCBA dataset details

PCBA [3] is the standard PubChem BioAssay benchmark, comprising 1,328 binary high-throughput-screen assays measured on 1.56 M small molecules. Each (molecule, assay) cell is either active (1), inactive (0), or missing (NaN); the matrix is sparse—most molecules have been screened against only a handful of the 1,328 assays.

Preprocessing and benchmark contamination. We canonicalize each PCBA SMILES to InChI, deduplicate, and route any molecule whose InChI also appears in a Polaris or MoleculeACE *test* split into our validation set so it cannot leak into pretraining. The remaining molecules form the PCBA training pool.

Per-assay filtering. Computed over the training rows only, we drop any assay that does not satisfy *all three* of the following: at least 100 positive labels, at least 100 negative labels, and at least 1,000 total non-NaN labels. Figure 7 shows the three distributions and the cutoffs as dashed lines. The thresholds remove assays that are either too small or too class-skewed to provide a useful learning signal, leaving 1089 of the 1328 input assays. The dominant cause of rejection is the positive-count threshold (235 assays below the cutoff of 100 positives), reflecting the strong class imbalance of high-throughput-screen data.

Loss and head architecture. Even after filtering, surviving assays have an average positive rate of roughly $1:100$. A naive binary cross-entropy is dominated by trivially-classified negatives, so we use focal loss [19] with $\gamma = 2$, $\alpha = 0.25$, which downweights well-classified examples and concentrates gradient on hard ones. The PCBA head is a single linear probe ($\text{Linear}(d, K)$ with $K = 1089$, no hidden layers) that emits one logit per surviving assay from the pooled fingerprint. The forward pass slices this $[B, K]$ output into K per-assay views; each view is registered as its own task in the MTL bookkeeping with its own focal-BCE loss (averaged over non-NaN cells of that assay) and its own learnable σ_t in UW. PCBA therefore contributes $K = 1089$ independent terms to the UW sum, yielding a total of $T = 1152$ tasks (62 PM6 + 1 conformer denoising + 1089 PCBA). The head weights themselves are shared, but the task-level weighting is per-assay, which lets UW automatically downweight assays whose loss is dominated by noise.

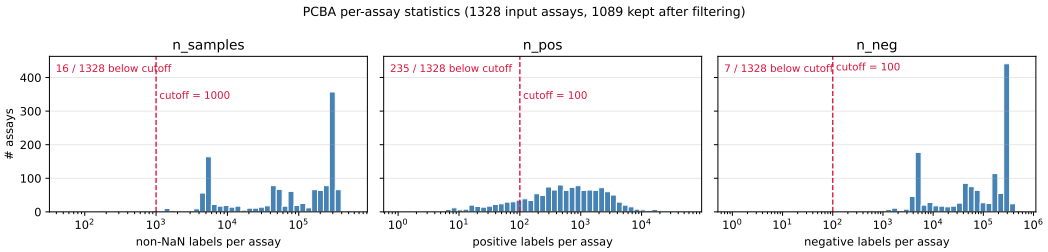

Figure 7: PCBA per-assay distributions of the three filter signals: number of non-NaN labels (left), positive labels (middle), and negative labels (right) per assay. Dashed crimson lines mark the training-time cutoffs (1000 / 100 / 100). An assay is dropped if it fails any single threshold; this leaves 1089 of the 1328 input assays. The positive-count threshold accounts for the bulk of the rejected assays (235 below the cutoff), reflecting strong class imbalance in high-throughput screens.

C MoleculeACE details

Figure 8: The activity distributions for cliff and non-cliff molecules in MoleculeACE.

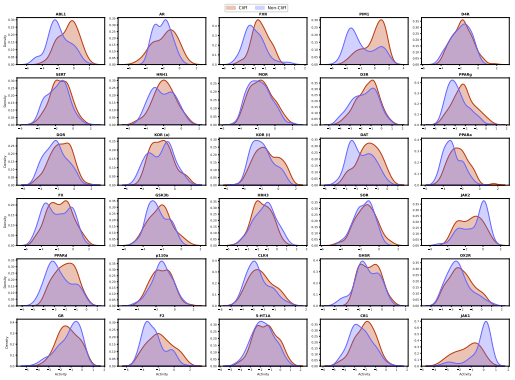

Table 9: Overview of MoleculeACE dataset statistics, highlighting the prevalence of stereochemical features in the test set.

Target Name	ID	Cliff Mols	Non-Cliff Mols	R/S Stereo (%)	E/Z Stereo (%)
Serotonin 1a receptor	5-HT1A	245	421	35.1	2.3
Tyrosine-protein kinase ABL1	ABL1	67	94	21.1	1.2
Androgen Receptor	AR	32	102	67.9	6.7
Cannabinoid receptor 1	CB1	76	132	33.7	10.1
Dual specificity protein kinase CLK4	CLK4	13	136	14.8	0.0
Dopamine D3 receptor	D3R	320	413	27.3	4.2
Dopamine D4 receptor	D4R	148	226	29.4	5.1
Dopamine transporter	DAT	55	158	72.8	0.9
Delta opioid receptor	DOR	204	317	82.1	5.8
Thrombin	F2	220	333	54.8	3.3
Coagulation factor X	FX	296	325	49.6	5.5
Farnesoid X receptor	FXR	50	78	30.5	21.9
Ghrelin receptor	GHSR	73	66	100.0	5.8
Glucocorticoid receptor	GR	52	100	83.6	12.5
Glycogen synthase kinase-3 β	GSK3b	32	141	7.5	6.4
Histamine H1 receptor	HRH1	48	149	40.6	1.0
Histamine H3 receptor	HRH3	239	335	36.6	1.6
Janus kinase 1	JAK1	13	113	24.6	0.8
Janus kinase 2	JAK2	34	163	20.8	1.0
Kappa opioid receptor (agonism)	KOR (a)	91	102	89.1	11.9
Kappa opioid receptor (inhibition)	KOR (i)	223	299	84.3	5.2
μ-opioid receptor	MOR	259	371	82.1	6.2
Orexin receptor 2	OX2R	160	137	79.8	1.3
Serine/threonine-protein kinase PIM1	PIM1	116	178	37.8	0.3
Peroxisome proliferator-activated receptor α	PPARα	141	206	57.1	13.0
Peroxisome proliferator-activated receptor δ	PPARδ	94	132	45.6	8.0
Peroxisome proliferator-activated receptor γ	PPARγ	178	294	54.0	15.3
Serotonin transporter	SERT	127	215	59.6	5.6
Sigma opioid receptor	SOR	103	164	40.4	1.5
PI3-kinase p110-alpha subunit	p110a	81	112	13.0	0.0

D Comparison models

In our evaluation, we consider three baseline methods: MiniMol, CheMeleon, and MolFormer. For each method, we evaluate performance using two strategies: (1) extracting embeddings from the pre-trained backbone (without task-specific modification) and using them with a downstream predictor, and (2) finetuning the model on each specific task.

When employing the finetuning strategy, we repeat the process over 5 random seeds for each task, using an 80/20 train/validation split. This variation in finetuning seeds serves as a primary source of experimental variance. Additionally, when using TabPFN as the downstream predictor (applied to embeddings from either the frozen or finetuned models), we account for its internal variance source (`random_state`) and the number of estimators (`n_estimators`). Because TabPFN splits data among estimators, the seed determines this split. This introduces a second source of variance for finetuned methods and a primary source for frozen methods (including MiniMol$_{\mathsf{PFN}}$ and Monroe). These multiple sources of variance, combined with the diverse pre-training objectives of the baselines, are accounted for by the pairwise testing protocol described in the main text.

For the results reported in Table 1 and 3, the error bars represent the *mean of standard deviations*: we first calculate the standard deviation across seeds for each task, and then report the average of these per-task deviations. We chose this metric over the standard error of the mean because aggregating scores across nearly 30 tasks causes independent noise terms to average out, artificially dampening the variance. This masking effect obscures meaningful differences in stability between methods. With our chosen metric, methods with only one source of variance (e.g., MiniMol$_{\mathsf{PFN}}$ and Monroe, which rely on fixed pretrained embeddings and only have variance from TabPFN's internal sampling) are correctly shown to have significantly lower standard deviations compared to methods involving end-to-end finetuning where variance arises from optimization, data splits, and initialization.

For each baseline we adopt the finetuning strategy and implementation provided in the CheMeleon repository. Below we provide a breakdown of the finetuning strategies.

Feature	CheMeleon	MiniMol	MolFormer
Strategy	Full finetuning	Frozen features + Head	Full finetuning
Task head	chemprop default FFN	Residual MLP	Linear on CLS token
Epochs	20	Max 50 (patience=5)	5
Model selection	Best val loss	Best val loss	Final epoch
Batch size	64	64	8
Optimizer	Adam	Adam	AdamW
Learning rate	1e-4	3e-4	5e-5

Table 10: Model-specific finetuning details.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: The abstract and introduction state the main contributions: Monroe, a molecular foundation model; PM6-scale pretraining; stereochemistry-aware graph representation; conformer denoising and embedding decorrelation losses; uncertainty-weighted multi-task learning; and TabPFN-based downstream in-context prediction. The claims are evaluated on Polaris and MoleculeACE benchmarks and are supported by the experiments and ablations in Sections 4 and 4.4.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: Section 5 discusses limitations of widely used molecular benchmarks, including issues in TDC/Polaris-style datasets such as inconsistent experimental conditions, dynamic ranges, assay artifacts, curation errors, stereochemical ambiguity, and salt handling. The discussion also identifies future work around incorporating TabPFN into pretraining and using distributional predictors.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete proof?
   Answer: [N/A]
   Justification: The paper is primarily empirical and methodological. It introduces architecture, training losses, and evaluation protocols, but does not present formal theorems requiring proofs.
4. **Experimental result reproducibility**
   Question: Does the paper fully disclose the information needed to reproduce the main experimental results?
   Answer: [Yes]
   Justification: The paper describes the model architecture, featurization, stereochemistry rewiring, training objectives, downstream adaptation procedure, datasets, filtering criteria, contamination removal, evaluation protocol, statistical testing, and ablation settings. Further implementation details are provided in the appendix and associated code release.
5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results?
   Answer: [Yes]
   Justification: The paper provides an anonymized source-code repository, which will be released after the anonymous review process, and the experiments use public or cited datasets and benchmarks including PM6/UltraLarge, PCBA, Polaris, and MoleculeACE.

6. **Experimental setting/details**

   Question: Does the paper specify all training and test details, including data splits, hyperparameters, and how they were chosen?

   Answer: [Yes]

   Justification: Sections 3 and 4, together with Appendix B, describe the architecture, graph construction, stereochemistry augmentation, training losses, datasets, PCBA filtering, test-set contamination removal, downstream adaptation strategies, evaluation protocol, and ablation settings.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Section 4 describes statistically principled pairwise comparisons using Tukey HSD with Benjamini–Hochberg correction. The experiments report seed variability, including standard deviations across seeds in the main result and ablation tables.

8. **Experiments compute resources**

   Question: Does the paper provide sufficient information on the compute resources needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Implementation section describes the compute resources, and we are confident that our code release will enable other researchers to exactly duplicate our entire pipeline in four to five H100-days. Current estimates place this at about 500 US dollars, which we hope is within the resources of a significant proportion (although we appreciate not all) of the NeurIPS community. We further note that downstream applications would typically not run the pre-training pipeline, but use Monroe as a fingerprint generator, which is measured in milliseconds of GPU time.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The research described in the paper conforms with the NeurIPS Code of Ethics. The work develops and evaluates a molecular representation model using existing molecular datasets and benchmarks. It does not involve human subjects, crowdsourcing, participant data, private personal data, new biological agents, new chemical synthesis procedures, wet-lab experiments, surveillance, deception, or weapons development. We avoid benchmark contamination by removing pretraining molecules that overlap with downstream test sets. The main foreseeable ethical consideration is downstream dual use in molecular discovery, but the paper presents a predictive representation model rather than a generative model, so is not directly producing new molecules.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the potential for improved molecular representation learning to reduce the cost and time of exploratory molecular design, while noting that downstream societal effects depend on practitioners' chosen applications.

11. **Safeguards**

    Question: Does the paper describe safeguards for responsible release of data or models that have a high risk of misuse?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, scraped internet dataset, or model such as a pretrained language model with obvious direct misuse pathways.

The released asset is a molecular representation model and associated training/inference code.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited, and are licenses and terms of use respected?

    Answer: [Yes]

    Justification: The paper cites the datasets, benchmarks, software, and prior models it builds on or compares against, including PM6/PubChemQC, PCBA, Polaris, MoleculeACE, RDKit, TabPFN, MiniMol, CheMeleon, MolFormer, and related molecular-learning benchmarks, all of which are licensed freely for research purposes as embodied in this paper.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces Monroe and states that training and inference code will be released. The model and code release will include documentation covering training data, preprocessing, intended use, limitations, license, and reproduction instructions. For initial submission, released assets are anonymized.

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

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions, screenshots, and details about compensation?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject experiments.

15. **Institutional review board approvals**

    Question: Does the paper describe potential participant risks and obtain Institutional Review Board approval, if applicable?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or participant data, so IRB approval is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods?

    Answer: [N/A]

    Justification: The core method does not use an LLM as an important, original, or non-standard component. TabPFN is used as a prior-data fitted tabular model for in-context prediction, not as a language model.