Supporting Information: Deep Learning Foundation Models from Classical Molecular Descriptors

Jackson W. Burns[1†], Akshat Shirish Zalte[1†], Charlles R. A. Abreu[1], Jochen Sieg[2], Christian Feldmann[2], Miriam Mathea[2], William H. Green[1*]

[1]Department of Chemical Engineering, MIT, Cambridge, Massachusetts, USA.

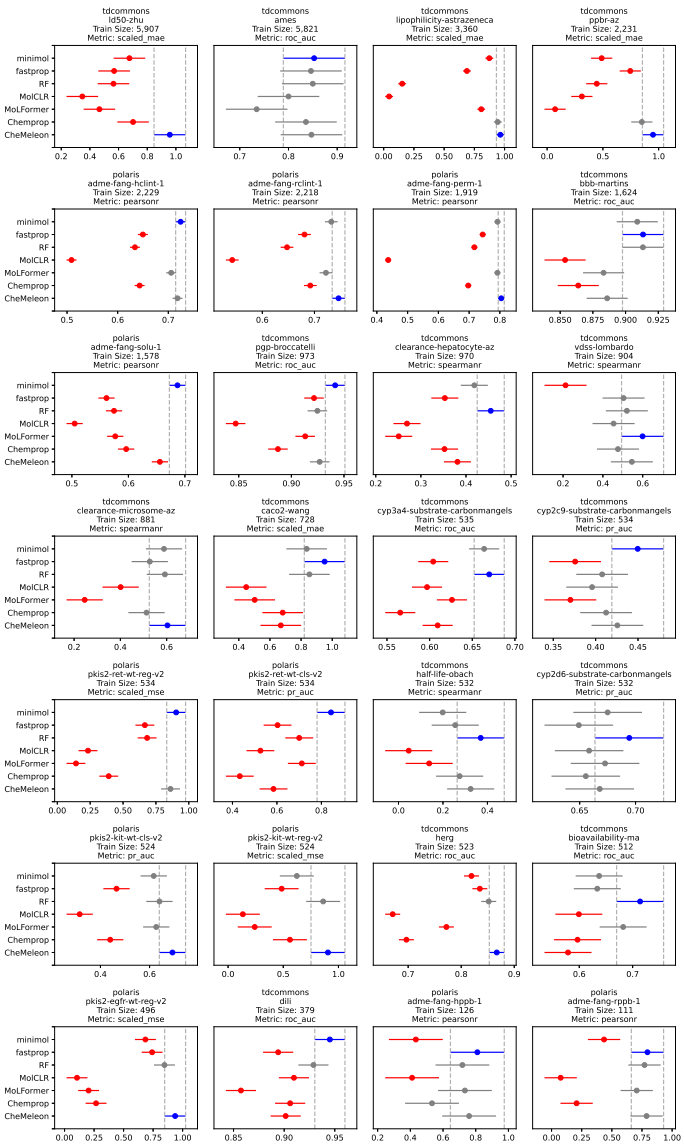
[2]BASF SE, Ludwigshafen, Germany.

*Corresponding author(s). E-mail(s): whgreen@mit.edu;

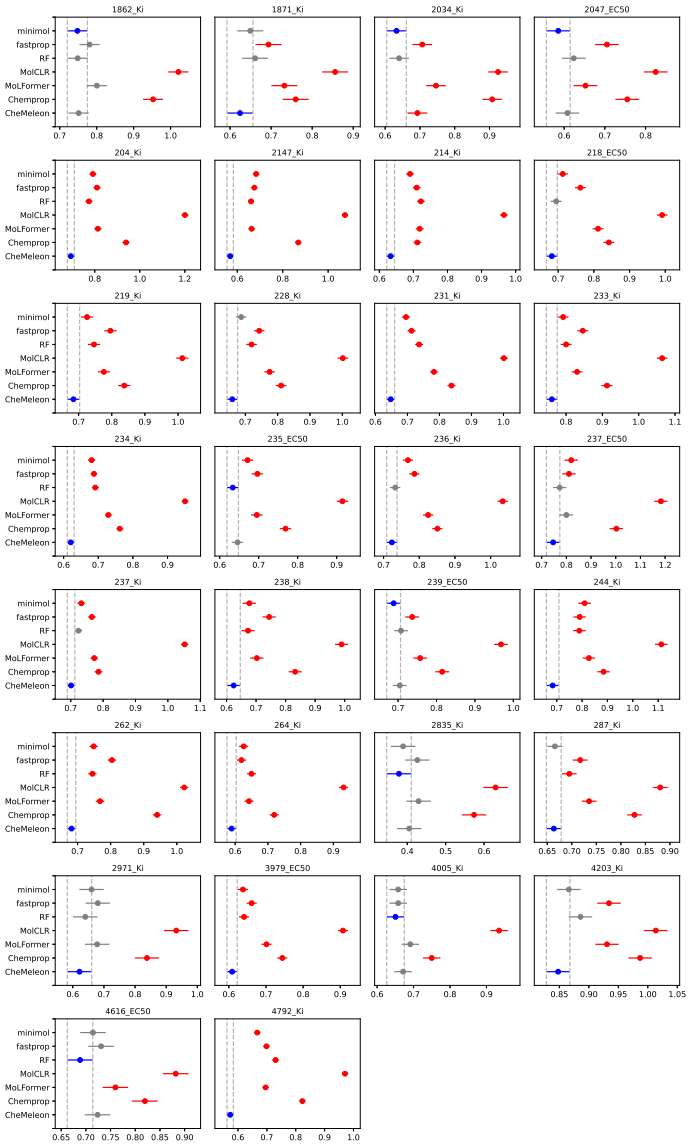
[†]These authors contributed equally to this work.

S1 Detailed Experimental Results

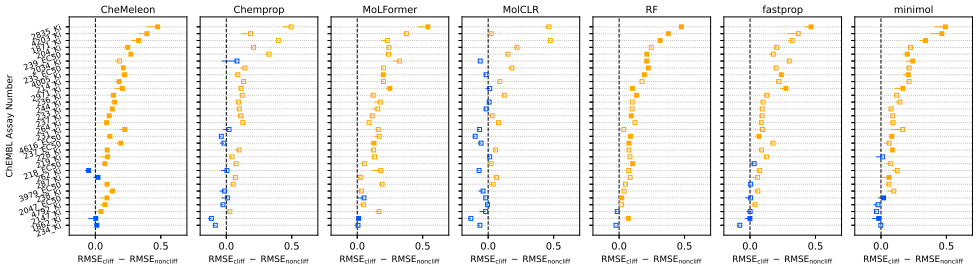
The complete set of MoleculeACE-style diagrams and Honestly Significant Difference diagrams following the conventions described in the main text (Figures 2 and 3, respectively) are provided in Figures S1, S2, and S3. A high-quality version of these images is also available in the GitHub repository described in Section 3.5.



Supplementary Figure S1: Performance of all of the tested models across all of the tested Polaris benchmark tasks. The origin of each benchmark set is shown as the first line of each subplot title, followed by the name of the dataset (which indicates the task), the size of the training data, and the metric used to evaluate model performance. Benchmarks are sorted by training size in decreasing order. Models shown in blue are the absolute highest performers on the given benchmark, while models shown in gray are not practically different from the best performer according to the Tukey Honestly Significant Difference test based on the variance in test set performance across five repetitions, as laid out in Section 3.2. Models shown in red *are* practically worse performers and are considered to have "lost" on the indicated benchmark.



Supplementary Figure S2: Performance of all of the tested models on the MoleculeACE study in terms of Root Mean Square Error (RMSE). The ChEMBL assay and target corresponding to each benchmark is shown as the title of each subplot. Models shown in blue are the absolute highest performers on the given benchmark, while models shown in gray are not practically different from the best performer according to the Tukey Honestly Significant Difference test based on the variance in test set performance across five repetitions, as laid out in Section 3.2. Models shown in red *are* practically worse performers and are considered to have "lost" on the indicated benchmark.



Supplementary Figure S3: Performance of all tested models across the ChEMBL assays [1] curated as part of the MoleculeACE study [2]. Each assay is one row along the horizontal axis and the plotted value on the x-axis is the difference in root mean squared error for predictions of molecules in the cliff set ("cliff") and those not in the cliff set ("noncliff"). Dots shown in blue are not practically different from zero, a positive result indicating that the model performance on the two sets is indistinguishable. Filled dots indicating that the given model was statistically the best or indistinguishable from the best performer and hollow dots indicating that it was significantly worse than the best model for that dataset.

S1.1 MoleculeNet Benchmarks

This section extends our evaluation of **CheMeleon** to include all molecular property prediction benchmarks originally used to assess **Chemprop** [3], in particular the MoleculeNet benchmarks [4]. These were excluded from the main text because more relevant benchmarks have since been devised [5]. We used the same datasets and data splits as in the original **Chemprop** paper. These benchmarks include a mix of regression tasks (UV/Vis, SAMPL, QM9, PCQM4Mv2) and classification tasks (HIV, PCBA). No hyperparameter tuning was performed for either **CheMeleon** or **Chemprop** models.

Table S1: Comparison of model performance on the test set, trained with **CheMeleon** and **Chemprop**, for regression tasks, including UV/Vis peak absorption wavelength, logP for the SAMPL challenge (SAMPL 6, 7, and 9), and HOMO-LUMO gap for the PCQM4Mv2 dataset.

	CheMeleon			**Chemprop**		
Task	MAE	RMSE	R^2	MAE	RMSE	R^2
UV/Vis	16.86	31.36	0.911	20.00	34.2	0.894
SAMPL6	0.28	0.31	0.784	0.30	0.39	0.662
SAMPL7	0.48	0.70	-0.102	0.36	0.55	0.301
SAMPL9	0.85	1.04	0.786	0.94	1.09	0.763
PCQM4Mv2	0.09	0.15	0.982	0.10	0.16	0.982

Table S2: Comparison of model performance on the test set, trained with **CheMeleon** and **Chemprop**, for different targets of QM9. The tasks in the top group were trained together in a single multitask model. The bottom results are for single-task models.

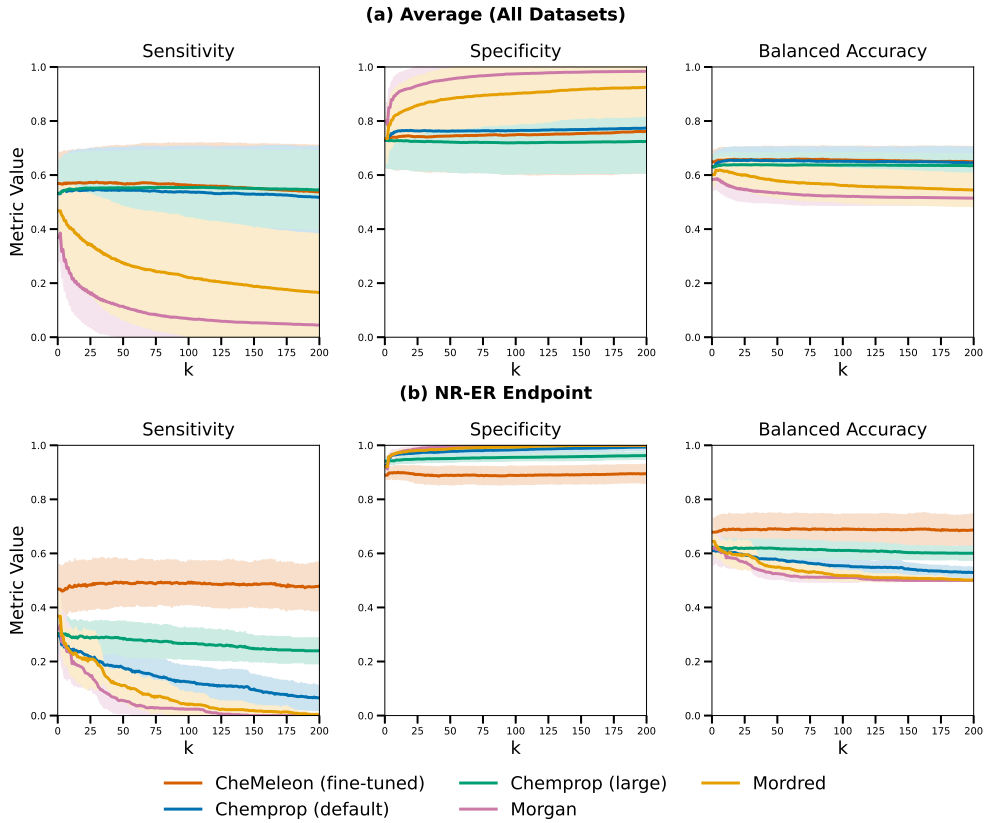
Model	Target	**CheMeleon**		**Chemprop**	
		MAE	RMSE	MAE	RMSE
multitask	mu	0.347	0.607	0.408	0.655
	alpha	0.294	0.824	0.371	0.680
	HOMO	0.00249	0.00426	0.00322	0.00499
	LUMO	0.00247	0.00417	0.00350	0.00519
	gap	0.00338	0.00600	0.00454	0.00709
	r2	18.1	36.0	24.6	40.3
	ZPVE	0.000463	0.000153	0.000416	0.000551
	Cv	0.134	0.331	0.181	0.288
	U0	3.73	15.64	3.55	5.37
	U298	3.75	15.75	3.54	5.38
	H298	3.77	15.84	3.54	5.38
	G298	3.53	14.51	3.50	5.36
individual	gap	0.00322	0.00588	0.00390	0.00653
	U0	2.17	14.13	1.45	2.90

Table S3: Comparison of model performance on the test set, trained with **CheMeleon** and **Chemprop**, for HIV and PCBA classification tasks.

	Split Type	**CheMeleon**			**Chemprop**		
		ROC-AUC	PRC-AUC	AP	ROC-AUC	PRC-AUC	AP
HIV	Random	0.7681	0.3318	0.3369	0.7990	0.3080	0.3057
PCBA	Random	0.9098	0.2104	0.2151	0.9045	0.1971	0.2023
PCBA (None)	Random	0.9055	0.3693	0.3755	0.9004	0.3614	0.3565
PCBA	Scaffold	0.8895	0.2875	0.2913	0.8812	0.265	0.2705

S2 k-Nearest Neighbors Representation Probing

Following the conventions of main text Section 1.4, Figure S4 presents results for the kNN probing study of the **CheMeleon** learned representation using an agglomerative clustering-based split.



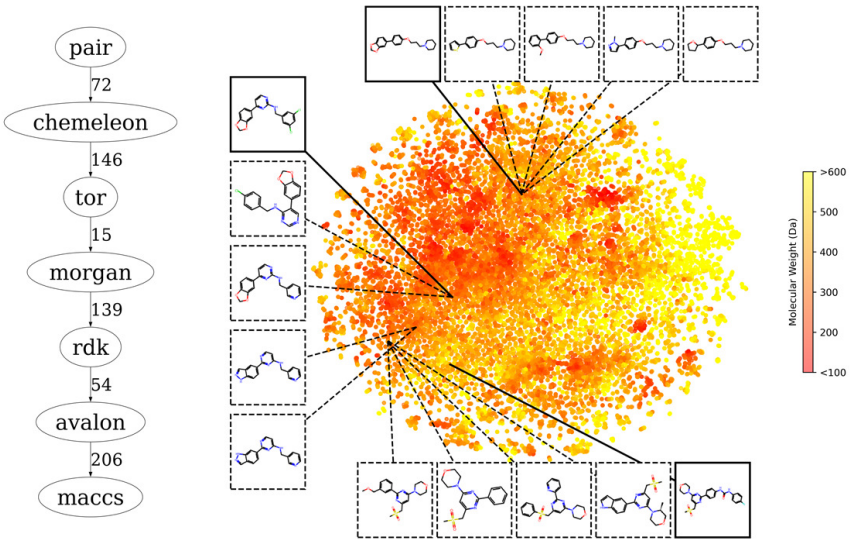
Supplementary Figure S4: kNN probing of fixed and learned molecule representations for the agglomerative clustering split. Upper row shows average results over 20 ToxCast endpoints. Lower row shows the results on the NR-ER endpoint, which was also used in the comparison by Ball et al. [6]. The results were obtained with a 5-fold cross validation using an agglomerative clustering split.

These results reinforce the conclusions of the main text, again showing that **CheMeleon** offers favorable performance improvements and trade-offs relative to standard models. The ablation model further reiterates that performance improvements are attributable to descriptor-based pre-training rather than an architectural bias.

S3 Foundation Fingerprint

Adapting the provided source code from the O'Boyle and Sayle [7] study, we generated the learned representation (LR) for each of the molecules provided in the Single Assay benchmark. We calculated the cosine distance between the embeddings within each series to arrive at a **CheMeleon**-based sorting order. This was compared with

the sort order derived from Atom Pair fingerprints [8], Topological-torsion fingerprints [9], Morgan fingerprints [10], RDKit fingerprints [11], Avalon fingerprints [12], and MACCS keys [13] as they collectively constitute the standard set of molecular fingerprints used for molecular representation, such as for feeding the Random Forest model used here. This provides insights into how the model has learned to separate common chemical moieties. The same **CheMeleon** fingerprints are then subjected to the methods of Orlov et al. [14] to generate a two-dimensional projection, and a few of the series present in the benchmark are highlighted. An overview of the complete fingerprint analysis is presented in Figure S5.



Supplementary Figure S5: (left) Hasse diagram showing the difference in performance on the O'Boyle and Sayle [7] Single Assay benchmark when using **CheMeleon** as a fingerprint in comparison to the traditional Atom Pair fingerprints [8] ("pair"), Topological-torsion fingerprints [9] ("tor"), Morgan fingerprints [10] ("morgan"), RDKit fingerprints [11] ("rdk"), Avalon fingerprints [12] ("avalon"), and MACCS keys [13] ("maccs"). A directed edge indicates that a practically significant difference in correctness of sorting order exists between the two nodes, with edge weight indicating the number of series in the entire benchmark set for which a practical difference was observed. See the original study for a more detailed discussion on the formulation and interpretation of this Hasse diagram [7]. (right) t-SNE projection computed using `scikit-learn` [15] with perplexity hyperparameter tuned such that highly similar neighborhoods in the original feature space are preserved in the projected space following the procedure laid out by Orlov et al. [14]. Three of the series present in the benchmark data have been highlighted with the "lead" molecule shown in bold and the consecutively more dissimilar molecules shown after it.

The t-SNE projection shows that **CheMeleon** can separate large and small molecules in its feature space and that the three highlighted assays are far apart from each other. Each assay also provides further chemical insight. At the top, all five species are small modifications of the original structure at the same site and are thus all projected into similar locations in chemical space. The assay on the left shows that modifications removing the methylenedioxy group cause two subsets of the series to be projected into *different* regions of chemical space. The final assay (shown at the bottom) follows a similar trend. In this case, however, the lead compound is not projected into the same chemical space as the other four compounds in the series, reflecting the extent of the initial structural modification.

S4 Training Details

All models were trained using PyTorch Lightning [16] and associated open-source machine learning and cheminformatics packages. We used eight Nvidia 2080 Ti GPUs for foundation model training and an Nvidia Quadro RTX 4000 laptop GPU for fine-tuning. The architecture for **CheMeleon** was as described in the main text with hyperparameters chosen on the basis of heuristics.

To pre-train **CheMeleon**, a set of 1 million molecules was randomly selected from the PubChem database. They were further pre-processed to remove SMILES strings that were longer than 150 characters, had multiple fragments (such as salts), or could not be converted into a valid molecular graph using the RDKit [11]. From there, descriptors were calculated and saved to disk, as described in the main text. Note that we used mordred-community, a community-maintained fork of the original Mordred descriptor calculator, which is no longer maintained.

Unfortunately, many of the reportedly best models from the literature, such as those referenced in the main text, are not available for us to run, preventing direct, fair, and statistically rigorous comparisons between models. **GROVER** [17], one of the first foundation models in this space, no longer provides model weights and has not been maintained since publication. **MolE** [18] provides weights for a toy model used to demonstrate reproducibility, but not the full model. **MolGPS** [19] and Beaini et al. [20] pre-trained foundation models following the concatenated datasets approach discussed previously, and released only the training datasets. **GraphQPT** [21] leveraged QM descriptors as targets during pre-training but has not yet made model checkpoints available.

Among the models which were available, we selected a representative set of the best performers. Every reference model benchmarked as part of this study used the following architecture and training approach, which were decided in deference to the suggested configuration from each of the original studies where such an indication was given, and held consistent otherwise:

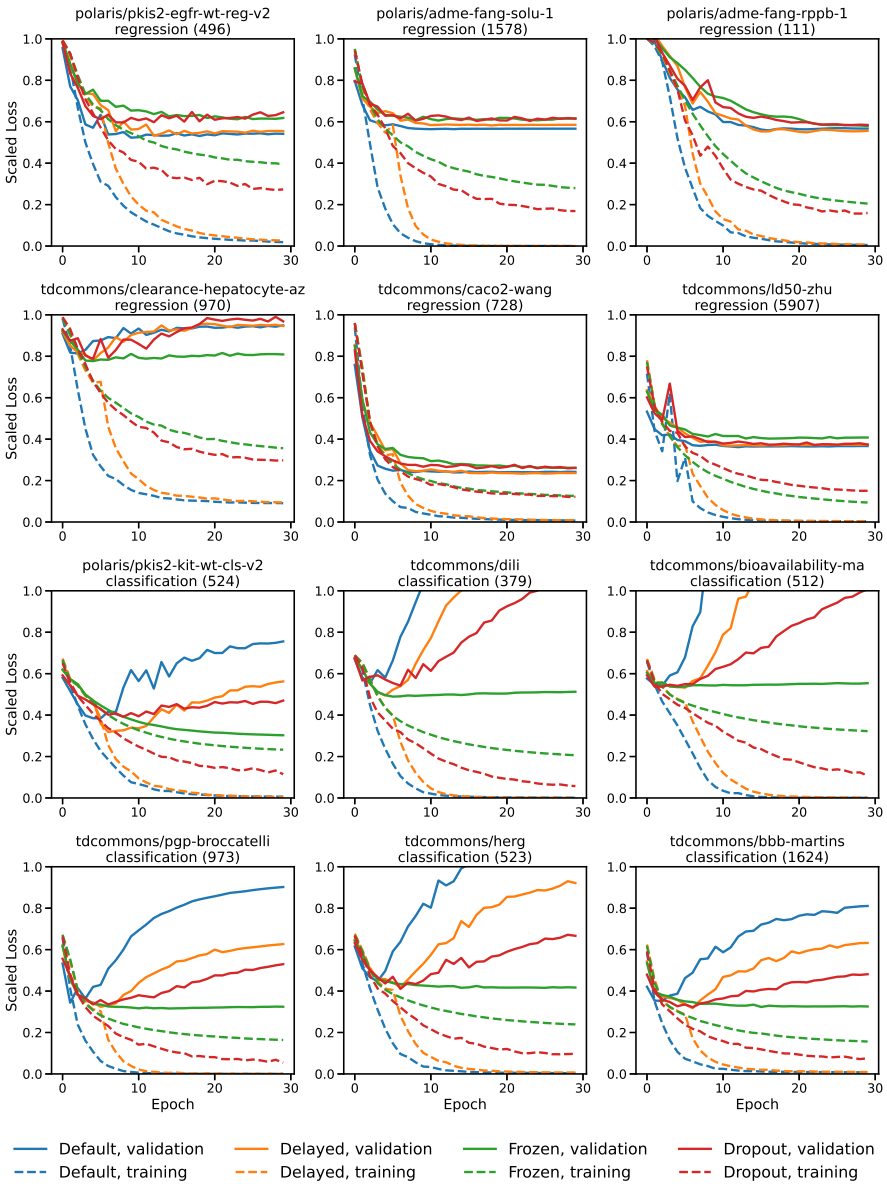
- **Random Forest**: increased number of estimators to 500 as suggested in Svetnik et al. [22] and set the maximum number of features to log_2 of their total number
- **MoLFormer**: tuned the entire network with a single additional readout layer
- **fastprop**: default settings as set by Burns and Green [23]
- **Chemprop**: default settings as set by Heid et al. [3]

- **minimol**: learned representation held constant, tuned a multi-layer skip connection MLP with regularization and hidden size of 512 as described by Kläser et al. [24]

When using **CheMeleon** for a benchmark, an equivalent directed message-passing neural network (D-MPNN) is initialized with the pre-trained weights and biases (≈ 8.7 million parameters) and followed by a feedforward neural network (FNN) initialized from scratch (single hidden layer, $\approx$ 525 thousand parameters). The entire model is then fine-tuned end-to-end using stochastic gradient descent. Despite the distinct initialization and disparate parameter counts, treating the two submodels equally during fine-tuning is typically the most effective approach, as we demonstrate in Section S5.

S5 Analysis of Fine-Tuning Strategies

The choice of fine-tuning strategy can significantly impact the performance of a pre-trained model on downstream tasks [25, 26]. We evaluated several fine-tuning strategies across the 28 Polaris benchmarks. These include the default end-to-end fine-tuning, delayed unfreezing of the D-MPNN, permanently freezing the D-MPNN, and applying dropout regularization. Averaged learning curves obtained by applying these strategies to 12 representative benchmarks from the Polaris benchmark set are shown in Figure S6.



Supplementary Figure S6: Learning curves for the fine-tuning of **CheMeleon** on selected Polaris benchmarks. Four different strategies were tested: 1. *Default*: fine-tuning the entire model using default **Chemprop** settings, 2. *Delayed*: freezing the D-MPNN weights for the first 5 epochs, 3. *Frozen*: freezing the D-MPNN weights for the entire process, and 4. *Dropout*: applying a 20% dropout rate to both the D-MPNN and FNN layers. The curves are averaged over 5 runs with different random splits (train:validation = 4:1). The subplot titles indicate the benchmark name, the task type, and the training set size.

The first two rows of Figure S6 correspond to regression tasks, while the third and fourth rows correspond to classification tasks. Under the default fine-tuning strategy, the validation loss decreases rapidly in all cases, typically reaching a minimum within fewer than 10 epochs. Overall, **CheMeleon**'s pretrained representations appear particularly well-suited for regression tasks, with the default strategy yielding the best or near-best performance across all regression benchmarks.

Notably, for regression tasks we observe little to no degradation in validation loss after the minimum is reached, even as the training loss continues to decrease, often approaching zero. Although this behavior may appear counterintuitive, it has been reported previously in large, overparameterized neural networks trained with stochastic gradient descent, particularly when initialized from pretrained representations and applied to small datasets [27, 28]. One possible explanation is an effective separation between network components that capture underlying structure and those that primarily memorize label noise, such that memorization does not substantially impair generalization to unseen inputs.

In contrast, classification tasks exhibit more pronounced overfitting, making early stopping necessary to prevent degradation in validation performance. In these cases, freezing the D-MPNN for the first few epochs sometimes lowers the minimum validation loss, although the effect is inconsistent across benchmarks. The transition from pre-training to fine-tuning likely induces a more severe change in the effective loss landscape for classification tasks than for regression tasks, which may reduce transfer efficiency. However, further investigation is required to test this hypothesis.

Permanently freezing the backbone consistently underperforms in regression tasks. In classification tasks, it typically reaches a minimum validation loss comparable to the other strategies, but requires substantially longer training time to do so. Finally, introducing a 20% dropout rate has little effect on performance for either regression or classification tasks.

Overall, although delayed unfreezing can be beneficial in specific classification scenarios, the default strategy of treating the pretrained D-MPNN and the newly initialized FNN uniformly during fine-tuning emerges as a robust and effective approach for **CheMeleon**. This approach is used throughout the present study and suggested for all downstream user applications of **CheMeleon**.

References

[1] Gaulton, A., Bellis, L.J., Bento, A.P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., Overington, J.P.: Chembl: a large-scale bioactivity database for drug discovery. Nucleic Acids Research **40**(D1), 1100–1107 (2011) https://doi.org/10.1093/nar/gkr777

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

[3] Heid, E., Greenman, K.P., Chung, Y., Li, S.-C., Graff, D.E., Vermeire, F.H., Wu, H., Green, W.H., McGill, C.J.: Chemprop: A machine learning package for chemical property prediction. Journal of Chemical Information and Modeling **64**(1), 9–17 (2023) https://doi.org/10.1021/acs.jcim.3c01250

[4] Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.: Moleculenet: a benchmark for molecular machine learning. Chemical Science **9**(2), 513–530 (2018) https://doi.org/10.1039/c7sc02664a

[5] Ash, J.R., Wognum, C., Rodríguez-Pérez, R., Aldeghi, M., Cheng, A.C., Clevert, D.-A., Engkvist, O., Fang, C., Price, D.J., Hughes-Oliver, J.M., Walters, W.P.: Practically significant method comparison protocols for machine learning in small molecule drug discovery. Journal of Chemical Information and Modeling **65**(18), 9398–9411 (2025) https://doi.org/10.1021/acs.jcim.5c01609

[6] Ball, N., Madden, J., Paini, A., Mathea, M., Palmer, A.D., Sperber, S., Hartung, T., Ravenzwaay, B.: Key read across framework components and biology based improvements. Mutation Research/Genetic Toxicology and Environmental Mutagenesis **853**, 503172 (2020) https://doi.org/10.1016/j.mrgentox.2020.503172

[7] O'Boyle, N.M., Sayle, R.A.: Comparing structural fingerprints using a literature-based similarity benchmark. Journal of Cheminformatics **8**(1) (2016) https://doi.org/10.1186/s13321-016-0148-0

[8] Carhart, R.E., Smith, D.H., Venkataraghavan, R.: Atom pairs as molecular features in structure-activity studies: definition and applications. Journal of Chemical Information and Computer Sciences **25**(2), 64–73 (1985) https://doi.org/10.1021/ci00046a002

[9] Nilakantan, R., Bauman, N., Dixon, J.S., Venkataraghavan, R.: Topological torsion: a new molecular descriptor for sar applications. comparison with other descriptors. Journal of Chemical Information and Computer Sciences **27**(2), 82–85 (1987) https://doi.org/10.1021/ci00054a008

[10] Morgan, H.L.: The generation of a unique machine description for chemical

structures-a technique developed at chemical abstracts service. Journal of Chemical Documentation **5**(2), 107–113 (1965) https://doi.org/10.1021/c160017a018

[11] Landrum, G.: rdkit. Zenodo (2025). https://doi.org/10.5281/ZENODO.591637 . https://zenodo.org/doi/10.5281/zenodo.591637

[12] Gedeck, P., Rohde, B., Bartels, C.: Qsar - how good is it in practice? comparison of descriptor sets on an unbiased cross section of corporate data sets. Journal of Chemical Information and Modeling **46**(5), 1924–1936 (2006) https://doi.org/10.1021/ci050413p

[13] Durant, J.L., Leland, B.A., Henry, D.R., Nourse, J.G.: Reoptimization of mdl keys for use in drug discovery. Journal of Chemical Information and Computer Sciences **42**(6), 1273–1280 (2002) https://doi.org/10.1021/ci010132r

[14] Orlov, A.A., Akhmetshin, T.N., Horvath, D., Marcou, G., Varnek, A.: From high dimensions to human insight: Exploring dimensionality reduction for chemical space visualization. Molecular Informatics **44**(1) (2024) https://doi.org/10.1002/minf.202400265

[15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python (2012) https://doi.org/10.48550/ARXIV.1201.0490

[16] Falcon, W., The PyTorch Lightning team: PyTorch Lightning. https://doi.org/10.5281/zenodo.3828935 . https://github.com/Lightning-AI/lightning

[17] Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., Huang, J.: Self-supervised graph transformer on large-scale molecular data (2020) https://doi.org/10.48550/ARXIV.2007.02835

[18] Méndez-Lucio, O., Nicolaou, C.A., Earnshaw, B.: MolE: a foundation model for molecular graphs using disentangled attention. Nature Communications **15**(1) (2024) https://doi.org/10.1038/s41467-024-53751-y

[19] Sypetkowski, M., Wenkel, F., Poursafaei, F., Dickson, N., Suri, K., Fradkin, P., Beaini, D.: On the scalability of GNNs for molecular graphs. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024). https://openreview.net/forum?id=klqhrq7fvB

[20] Beaini, D., Huang, S., Cunha, J.A., Li, Z., Moisescu-Pareja, G., Dymov, O., Maddrell-Mander, S., McLean, C., Wenkel, F., Müller, L., Mohamud, J.H., Parviz, A., Craig, M., Koziarski, M., Lu, J., Zhu, Z., Gabellini, C., Klaser, K., Dean, J., Wognum, C., Sypetkowski, M., Rabusseau, G., Rabbany, R., Tang, J., Morris, C., Ravanelli, M., Wolf, G., Tossou, P., Mary, H., Bois, T., Fitzgibbon, A.W.,

Banaszewski, B., Martin, C., Masters, D.: Towards foundational models for molecular learning on large-scale multi-task datasets. In: The Twelfth International Conference on Learning Representations (2024). https://openreview.net/forum?id=Zc2aIcucwc

[21] Fallani, A., Nugmanov, R., Arjona-Medina, J., Wegner, J.K., Tkatchenko, A., Chernichenko, K.: Pretraining graph transformers with atom-in-a-molecule quantum properties for improved admet modeling. Journal of Cheminformatics **17**(1) (2025) https://doi.org/10.1186/s13321-025-00970-0

[22] Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P.: Random Forest: A classification and regression tool for compound classification and qsar modeling. Journal of Chemical Information and Computer Sciences **43**(6), 1947–1958 (2003) https://doi.org/10.1021/ci034160g

[23] Burns, J.W., Green, W.H.: Generalizable, fast, and accurate DeepQSPR with fastprop. Journal of Cheminformatics **17**(1) (2025) https://doi.org/10.1186/s13321-025-01013-4

[24] Kläser, K., Banaszewski, B., Maddrell-Mander, S., McLean, C., Müller, L., Parviz, A., Huang, S., Fitzgibbon, A.: `MiniMol`: A parameter-efficient foundation model for molecular learning (2024) https://doi.org/10.48550/ARXIV.2404.14986

[25] Peters, M.E., Ruder, S., Smith, N.A.: To tune or not to tune? adapting pretrained representations to diverse tasks. In: Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP@ACL), pp. 7–14. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/W19-4302

[26] Kumar, A., Raghunathan, A., Jones, R., Ma, T., Liang, P.: Fine-tuning can distort pretrained features and underperform out-of-distribution (2022) https://doi.org/10.48550/ARXIV.2202.10054

[27] Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences **116**(32), 15849–15854 (2019) https://doi.org/10.1073/pnas.1903070116

[28] Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021) https://doi.org/10.1145/3446776