MOLPFN: GRAPH-LEVEL PRIOR-DATA FITTED NETWORKS FOR MOLECULAR PROPERTY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular property prediction is essential across chemical, biological, and materials applications, from molecular discovery to materials design. However, experimentally measured labels are often scarce, whereas existing molecular models often require large labeled datasets and task-specific training. Tabular foundation models based on Prior-Data Fitted Networks (PFNs) offer a promising alternative: they achieve strong predictive performance in small-data regimes and can adapt to new tasks through in-context learning without task-specific parameter updates. Yet, applying these models to molecules requires bridging the gap between graph-structured molecular inputs and the tabular representations expected by PFNs. We introduce MOLPFN, a framework for in-context molecular property prediction with a learnable molecular graph adapter, that transforms molecular graphs into PFN-compatible representations, via a synthetic graph prior designed for PFN pretraining. MOLPFN thereby combines molecular graph structure and atomic information with the in-context learning capabilities of PFNs, while supporting optional auxiliary molecule-level information, such as learned embeddings and molecular fingerprints, without task-specific retraining. We evaluate MOLPFN on nine molecular benchmarks covering regression and classification tasks. When trained only on tasks sampled from the synthetic graph prior, MOLPFN matches or outperforms established tabular foundation models consistently improving results across tasks. We further fine-tune MOLPFN across thousands of experimentally measured small-molecule bioactivity tasks, substantially improving few-shot prediction for previously unseen protein targets and outperforming most established few-shot molecular learning baselines, even when using only standard atomic features. Together, these results show how domain-specific adapters and structured priors can extend tabular foundation models to molecular graphs, enabling flexible in-context prediction across molecular tasks and representations.

1 INTRODUCTION

Molecular property prediction plays a fundamental role in drug discovery and materials science, powering applications such as virtual screening and toxicity assessment (Mayr et al., 2016). Molecules are naturally represented as graphs, enabling Graph Neural Networks (GNNs) that directly model atomic features and interactions (Zhou et al., 2020), while geometry-aware models can additionally exploit three-dimensional structure (Satorras et al., 2021). However, training such models for individual prediction tasks can be challenging when labelled data are scarce, as molecular labels often rely on costly experiments or simulations (Stanley et al., 2023; Eraqi et al., 2025).

To reduce this dependence on task-specific labelled data, molecular representation learning pretrains graph encoders on large molecular collections using self-supervised objectives such as molecular contrastive learning (Wang et al., 2022; Xia et al., 2023). The representations can then be transferred to downstream tasks, typically by extracting fixed-dimensional embeddings and training a separate predictor for each target property (Oliveira & Saeed, 2026). While pretraining reduces reliance on labelled data, adaptation to new molecular properties still typically requires task-specific training.

Few-shot molecular property prediction poses the chance of predicting properties from only a small labelled support set. Meta-learning approaches use a meta-training set of related tasks to learn transferable representations, adaptation strategies, or decision rules. These include gradient-based meth-

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

ods, prototypical networks, and multitask-pretrained GNNs (Finn et al., 2017; Snell et al., 2017). The FS-Mol dataset (Stanley et al., 2021) provides meta-training tasks derived from biological assays and evaluates few-shot performance on hold-out tasks.

Prior-Data Fitted Networks (PFNs) provide an alternative to task adaptation using in-context learning(ICL), performing inference on new tasks from labeled context examples without task-specific parameter updates (Hollmann et al., 2023; Guan et al., 2026). Recent tabular in-context learners, including TabICL and LimiX, extend this paradigm to larger datasets and broader task distributions (Qu et al., 2025; Zhang et al., 2025). However, these models assume fixed-dimensional tabular inputs and therefore cannot directly process molecular graphs.

GraphPFN extends PFNs to graph-structured data using learnable message-passing adapters (Eremeev et al., 2025; 2026), but focuses primarily on node-level prediction rather than settings in which each input is an entire molecular graph. For graph-level molecular prediction, Monroe model instead encodes each molecule into a fixed-dimensional representation for a downstream TabPFN (Banaszewski & Fitzgibbon, 2026), decoupling molecular representation learning from PFN inference and compressing the graph structure before it reaches the model.

In this work, we introduce MOLPFN, a novel end-to-end PFN-based framework designed specifically for graph-level molecular property prediction. MOLPFN augments a tabular PFN with learnable molecular graph adapters that propagate information through molecular topology and produce PFN-compatible graph-level representations. This allows for molecular representation learning and in-context prediction to be optimized jointly, rather than relying on representations from separately pretrained molecular encoders. We additionally develop a synthetic molecular graph prior for PFN pretraining that generates diverse graph-level prediction tasks reflecting relationships among atomic attributes, molecular topology, and graph-level targets. Trained solely on this prior, MOLPFN consistently improves upon tabular PFN baselines and achieves competitive performance against established graph-learning and molecular representation-learning methods.

Beyond synthetic pretraining, we further train the shared parameters of MOLPFN across thousands of experimentally measured FS-Mol training assays. At evaluation time, the model predicts on held-out FS-Mol tasks in a single forward pass conditioned on each task's labeled support set, without gradient-based adaptation. This enables a direct comparison between PFN-style in-context learning and established few-shot molecular learning methods under the same evaluation protocol.

2 BACKGROUND AND RELATED WORK

Prior-data fitted networks. Prior-data fitted networks (PFNs) are pretrained over a distribution of supervised-learning tasks to learn an inference procedure shared across tasks (Müller et al., 2022). Each pretraining task consists of a labeled context set and query examples, with the model learning to predict query labels conditioned on the context. This training objective can be interpreted as approximating posterior predictive inference under the task prior. At test time, inference on a new task is performed by conditioning on its labeled context set, without updating model parameters.

TabPFN applies the PFN framework to tabular prediction by pretraining a Transformer on synthetic datasets (Hollmann et al., 2023). Its task prior includes structural causal models (SCMs), in which variables are generated from their parents in a directed acyclic graph through nonlinear mechanisms (Pearl, 2009; Peters et al., 2017). Varying the causal structure, mechanisms, and noise distributions generates diverse feature–target relationships. Subsequent models, including TabICL and LimiX, further develop this synthetic-task pretraining paradigm (Qu et al., 2025; Zhang et al., 2025).

LimiX represents a tabular dataset as a two-dimensional grid of tokens, with rows corresponding to examples and columns to groups of input features and the target (Zhang et al., 2025). Its input encoder partitions the feature columns into groups and maps each group to a token embedding. Each row contains a set of feature-group tokens and a target token, which encodes the observed label for context examples or a masked label for queries. Transformer blocks processes this grid along two axes: *feature-level attention* (FLA) operates within each row, allowing feature-group and target tokens to interact, while *sample-level attention* (SLA) operates across rows, incorporating information from labeled context examples. The final query target-token representation is decoded into a regression prediction or class probabilities.

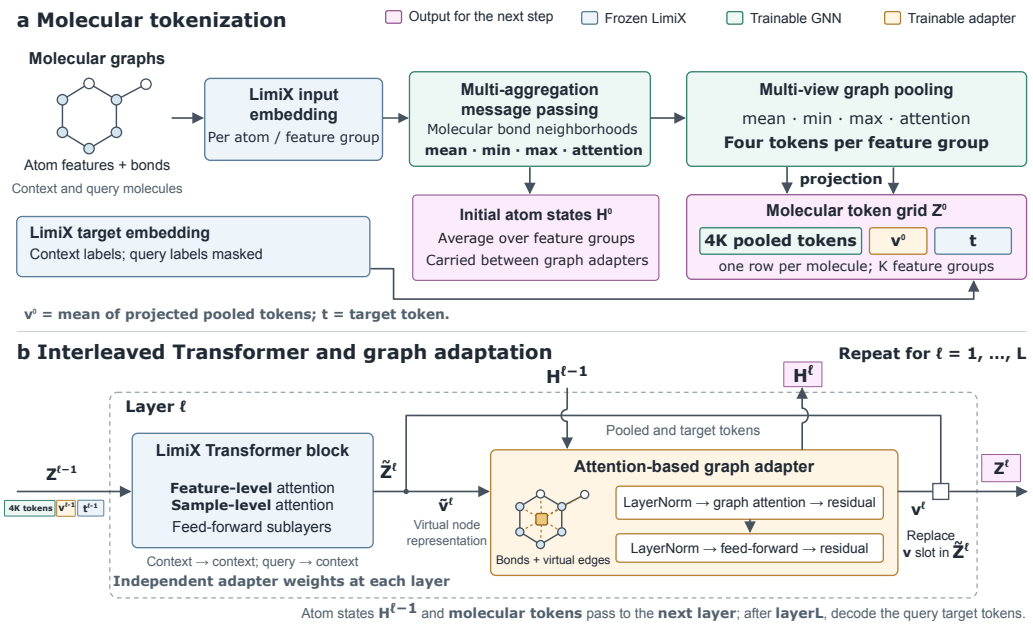


Figure 1: **MOLPFN architecture.** (a) Frozen atom embeddings are refined by a trainable GNN and pooled into four tokens per feature group. A virtual-node token and a target token complete each molecular row. (b) Frozen LimiX Transformer blocks alternate with trained graph adapters that update persistent atom states and replace the virtual-node token.

GraphPFN extends pretrained LimiX to node-level prediction through attention-based message-passing adapters (Eremeev et al., 2026). Its synthetic prior generates graph topologies using a two-level construction based on degree-corrected stochastic block models (Karrer & Newman, 2011), and node features and tergets through a graph-aware SCM combining randomly initialized MLP transformations with graph aggregation. GraphPFN treats individual nodes as prediction examples, whereas our setting requires prediction over collections of graphs, with each complete graph constituting one example.

3 MOLPFN

MOLPFN is a graph foundation model for in-context molecular property prediction. It extends the pretrained LimiX backbone (Zhang et al., 2025) with a molecular tokenization module and the attention-based graph adapters introduced by GraphPFN (Eremeev et al., 2026). The tokenization module converts each molecular graph into a row of tokens, allowing molecules to occupy LimiX's sample dimension. Between Transformer blocks, graph adapters exchange information across the molecular tokens and the underlying atom representations. All pretrained LimiX parameters remain frozen; only the molecular tokenization module and graph adapters are trained.

3.1 MOLECULAR TOKENIZATION

Molecular tokenization proceeds in three steps: embedding atom features, refining them along molecular bonds, and pooling the resulting atom states into molecule-level tokens, Figure 1a.

Atom encoding and local message passing. Let $G_m = (V_m, E_m)$ denote molecule m, with atom features $X_m \in \mathbb{R}^{|V_m| \times F}$, the frozen LimiX input encoder groups atom features in pairs, padding the final group with a zero when necessary. For $K = \lceil F/2 \rceil$ feature groups, the encoder creates K embeddings of dimension d, giving a tensor of shape $|V_m| \times K \times d$ per molecule. We use LimiX-16M, for which $d = 192$.

A trainable message-passing layer processes each embedded feature group with shared weights. For each atom, it combines the atom's transformed embedding with three arithmetic summaries of its neighbors: mean, minimum, and maximum. An attention mechanism provides an additional learned neighbor aggregation. The resulting five vectors are concatenated and projected back to dimension d, followed by layer normalization and ReLU. The result is one refined state $h_{m,a,g} \in \mathbb{R}^d$ for each atom a and feature group g.

From atom states to molecular tokens. Whereas message passing aggregates neighboring atoms, pooling aggregates all atoms in a molecule. Following atom encoding and message passing, each molecule is represented by a tensor of shape $|V_m| \times K \times d$. Pooling aggregates the atom dimension separately for each feature group, replacing its $|V_m|$ atom embeddings with four molecular tokens, each summarizing one pair of input features, embedded and then aggregated by one of the four pooling operations. The resulting representation therefore contains $4K$ tokens, each of dimension d.

Specifically, we compute mean, minimum, maximum, and learned attention readouts, then apply a shared linear projection:

$$p_{m,g,r} = W_{\mathrm{p}}\ \mathrm{Pool}_r\big(\{h_{m,a,g} : a \in V_m\}\big) + b_{\mathrm{p}}, \qquad r \in \{\mathrm{mean}, \mathrm{min}, \mathrm{max}, \mathrm{attn}\}. \tag{1}$$

Collecting these tokens gives a molecular representation $P_m \in \mathbb{R}^{4K \times d}$, which forms the pooled-token portion of a single molecular row in LimiX's token grid. Thus, each molecule occupies one sample position while retaining multiple tokens to describe its features.

Virtual-node and target tokens. Two additional tokens complete the molecular row. A virtual-node token connects the Transformer representation to subsequent graph processing, while a target token encodes the observed context label or a masked query label. The virtual-node token is initialized by averaging all pooled molecular tokens. We also retain one state per atom, obtained by averaging its refined embeddings across feature groups:

$$v_m^0 = \frac{1}{4K} \sum_{g=1}^{K} \sum_{r} p_{m,g,r}, \qquad h_{m,a}^0 = \frac{1}{K} \sum_{g=1}^{K} h_{m,a,g}. \tag{2}$$

LimiX feature embeddings are added to the pooled and virtual-node tokens before the target token is appended.

The complete molecular row thus contains $4K + 2$ tokens, each of dimension d: four pooled tokens per feature group, one virtual-node token, and one target token. This token count depends on the number of input features, not the number of atoms. Together, the tokens represent one molecule, with the target token providing its prediction readout. The retained atom states are stored separately for graph processing.

3.2 Interleaved In-Context and Graph Processing

The model alternates between two operations: the frozen Transformer incorporates labeled context, and the graph adapters refine representations using molecular connectivity, Figure 1b.

Within each Transformer block, FLA mixes tokens within a molecular row, while SLA incorporates information across molecules. Context molecules attend to context molecules, and query molecules attend to context molecules without attending to other queries.

Between successive Transformer blocks, a graph adapter $\mathcal{A}_\ell$ combines the updated virtual-node token with the retained atom states. It operates on an augmented graph $\widetilde{G}_m$ containing the molecular bonds and bidirectional edges between every atom and its virtual node. Writing $\widetilde{v}_m^\ell$ for the virtual-node token produced by Transformer block ℓ, the update is

$$\big(H_m^\ell, v_m^\ell\big) = \mathcal{A}_\ell\big(\widetilde{G}_m, H_m^{\ell-1}, \widetilde{v}_m^\ell\big), \tag{3}$$

where $H_m^{\ell-1}$ contains the incoming atom states. Each adapter uses sparse multi-head graph attention followed by a node-wise feed-forward network.

The updated virtual-node state replaces its token before the next Transformer block, and the updated atom states are retained for the next graph adapter. Pooled tokens and target tokens remain in the

Transformer stream and bypass the graph adapters. Thus, molecular pooling occurs once at input, while the virtual-node token enables repeated exchange between atom-level structure and context-conditioned representations.

3.3 Pre-Training and Prediction

We pretrain the graph modules on synthetic graph-level regression tasks. Each task is divided into context and query molecules. Targets are standardized using context statistics, and query labels are masked. The frozen regression head decodes the final normalized query target-token states, and training minimizes

$$\mathcal{L} = \frac{1}{|\mathcal{Q}|} \sum_{m \in \mathcal{Q}} \left(\widehat{\widetilde{y}}_m - \widetilde{y}_m \right)^2, \tag{4}$$

where $\mathcal{Q}$ is the query set and $\widetilde{y}_m$ denotes a standardized target. Gradients pass through the frozen backbone to optimize the molecular tokenization module and graph adapters. Architecture and optimization details are provided in Appendix A.1.

For binary classification, we retain the regression-trained graph modules and use LimiX's frozen classification target encoder and decoder. Context labels are encoded as class identities, and temperature-scaled softmax converts the two output logits into class probabilities. This setting evaluates classification transfer without classification-specific adapter training.

4 Graph-Level Prior

Our prior follows the two-stage formulation introduced by GraphPFN (Eremeev et al., 2026), adapting it to graph-level molecular prediction. Each synthetic dataset contains a collection of molecule-like graphs, with observed attributes at the atom level and one regression target per graph. We first sample molecular connectivity from a topology prior and then generate atom attributes and graph-level targets using a graph-aware structural causal model (SCM).

The topology prior uses a spanning-tree backbone augmented with ring-closing edges. This construction separates the generation of connected, branched structures from the introduction of cycles, allowing molecular size, connectivity, and cycle lengths to be controlled explicitly. The attribute prior subsequently assigns a shared, randomly sampled data-generating mechanism to the molecules within each dataset. A virtual readout node converts the node-level SCM output into the graph-level target y_m, as illustrated in Figure 2.

4.1 Structure Generation

Our structure generation module combines common chemoinformatic ideas to generate graphs reflecting molecular structures. We construct molecular skeletons as spanning trees (Diestel, 2017) augmented with ring-closing edges, a decomposition also used in SMILES (Weininger, 1988). Ring closure performed to approximately replicate the 2.7 cycles per molecule we find in the ZINC training subset (Irwin & Shoichet, 2005; Dwivedi et al., 2023) and a PubChem-derived collection (Kim et al., 2019).

We sample heavy atoms from $\{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{F}\}$, and use element-dependent valence budgets to guide branching. We grow a tree and add ring-closing bonds using ZINC-derived cycle-length weights that favor five- and six-membered cycles. Remaining positive valence is completed with explicit hydrogen leaves. Element identities and bond orders guide connectivity but are discarded before attribute generation. Appendix B.1 provides more details about the method including detailed prior statistics, also in comparison to the evaluated molecular dataset.

4.2 Attribute and Target Generation

To construct a new dataset, we first generate a set of molecular structures. We then sample a mechanism that establishes a common feature–target relationship across these structures, from which the dataset examples are generated. The SCM computation starts from random input variables, which serve as initial causes, together with optional structural descriptors. The aggregation operator and the degree of reliance on message passing are varied across datasets.

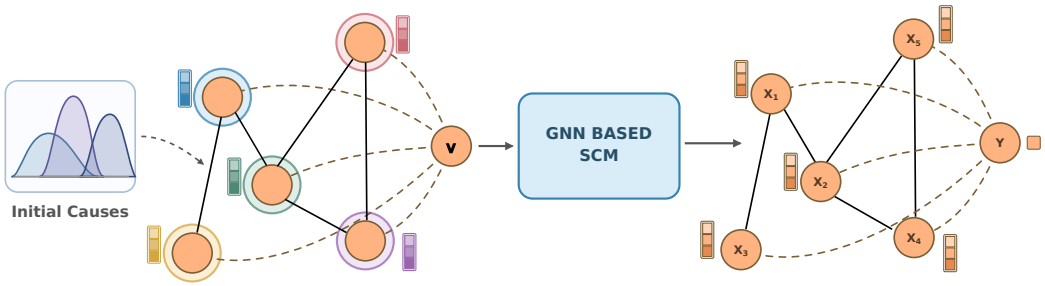


Figure 2: Graph-level synthetic prior for MOLPFN. Initial causes are latent input vectors, not observed atom features. Each coordinate is sampled independently from a standard normal distribution for atom nodes and set to zero for the virtual readout node. The diagram traces their transformation into observed features X and a molecular target Y.

To obtain one target per molecule, we add a virtual readout node connected bidirectionally to every atom. Its cause vector is set to zero, while atom nodes receive independent random causes. The SCM generates observed atom features from hidden activations and the molecular target from its final output at the virtual node. Thus, message passing can couple atom attributes and graph-level targets through molecular connectivity. This readout node is part of the data-generating prior and is distinct from the virtual-node representation used by the prediction model. Further information of the SCM configuration is provided in Appendix B.2.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and features. We evaluate MOLPFN on datasets spanning diverse molecular structures, atom types, dataset sizes, and target properties. Our conventional ICL evaluation includes AqSolDB (Sorkun et al., 2019), ZINC (Irwin & Shoichet, 2005), and seven MoleculeNet benchmarks (Wu et al., 2017): FreeSolv, ESOL, Lipophilicity, BBBP, BACE, ClinTox, and SIDER. FreeSolv, ESOL, Lipophilicity, AqSolDB, and ZINC are used for regression, whereas BBBP, BACE, ClinTox, and SIDER are used for classification. We employ an 80/10/10 train–validation–test partition, using the established splits for AqSolDB and ZINC and, following Wang et al. (2022), scaffold-based splits for the MoleculeNet benchmarks. For each dataset, we additionally augment the atom features with extended atom-level descriptors derived from each molecule referenced in Table A5. During evaluation, only labelled examples from the training split are supplied to MOLPFN as context, while test examples are treated as queries with their labels withheld. We additionally evaluate MOLPFN in the low-data regime using the few-shot tasks of FS-Mol (Stanley et al., 2021). Further dataset statistics, target descriptions, split details, and definitions of atom features are provided in Appendix C.2.

5.2 MOLECULAR PROPERTY PREDICTION BENCHMARKS.

Methods. We compare MOLPFN with conventional and PFN-based predictors. Because tabular methods cannot directly process variable-sized molecular graphs, we use separately pretrained molecular encoders to map each molecule to a fixed-dimensional representation. Specifically, we consider the graph-based Mole-BERT (Xia et al., 2023) and the Transformer-based MolDeBERTa (Oliveira & Saeed, 2026). Their molecular representations are supplied to either a k-nearest-neighbour predictor (k-NN)(Cover & Hart, 1967), with hyperparameters selected by grid search (Pedregosa et al., 2011) (see Appendix A.3), or to the tabular in-context learners TabICLv2 (Qu et al., 2026) and TabPFNv3 (Grinsztajn et al., 2026). This comparison contrasts MOLPFN's direct processing of molecular topology and atom features with pipelines operating on fixed representations from separately pretrained molecular encoders, paired with either a simple non-parametric k-NN predictor or more expressive tabular in-context learners.

Table 1: Regression performance on molecular property prediction datasets. Results are reported as mean ± standard deviation over ensemble runs. For deterministic NN-baseline, no error is reported. FreeSolv, ESOL, and Lipo use RMSE; AqSol and ZINC use MAE. Lower is better. Best results are shown in bold.

Method *Split*	**FreeSolv** *Scaffold*	**ESOL** *Scaffold*	**Lipo** *Scaffold*	**AqSol** *Predefined*	**ZINC** *Predefined*
Nearest-neighbour baseline					
Mole-BERT	3.18	1.45	0.978	1.400	0.960
MolDeBERTa	4.03	1.85	0.965	1.405	0.766
TabPFN v3					
Mole-BERT	2.79 ± 0.06	1.27 ± 0.02	0.747 ± 0.003	1.287 ± 0.003	0.837 ± 0.003
MolDeBERTa	2.88 ± 0.07	0.93 ± 0.01	0.751 ± 0.003	1.080 ± 0.002	0.464 ± 0.001
TabICL v2					
Mole-BERT	2.53 ± 0.05	1.26 ± 0.02	0.754 ± 0.004	1.297 ± 0.005	0.838 ± 0.004
MolDeBERTa	2.87 ± 0.09	0.95 ± 0.01	0.783 ± 0.003	1.148 ± 0.004	0.457 ± 0.004
LimiX					
Mole-BERT	2.72 ± 0.02	1.24 ± 0.01	0.815 ± 0.005	1.299 ± 0.003	0.889 ± 0.002
MolDeBERTa	2.51 ± 0.02	0.95 ± 0.01	0.807 ± 0.003	1.098 ± 0.004	0.481 ± 0.005
MOLPFN *(ours)*					
No molecular encoder	2.13 ± 0.02	1.114 ± 0.004	0.760 ± 0.001	1.092 ± 0.003	0.316 ± 0.001
+ Mole-BERT	**1.94** ± 0.01	1.016 ± 0.001	**0.730** ± 0.001	1.108 ± 0.002	0.343 ± 0.001
+ MolDeBERTa	2.00 ± 0.01	**0.848** ± 0.001	0.732 ± 0.002	**1.017** ± 0.001	**0.282** ± 0.003

Evaluation Results. For each ICL method, we average predictions from 8 forward passes with different random seeds controlling the permutation of feature columns and, for classification tasks, of class labels before each pass. We cap the in-context set at 2000 examples and additionally average over 10 independently sampled contexts, so that datasets exceeding this threshold are evaluated over 80 total forward passes.

On the regression benchmarks (Table 1), MOLPFN performs strongly using only atom-level features and molecular topology, without a pretrained molecular encoder that is required by the baseline methods. It outperforms the k-NN baseline on all five datasets, exceeds every ICL-based baseline on FreeSolv and ZINC with a particularly large improvement on ZINC. On Lipo and AqSol, it nearly matches the strongest ICL baseline, with only 1.7% and 1.1% higher error, respectively. The main exception is ESOL, where baselines using pretrained MolDeBERTa representations retain a clear advantage.

We next ask whether these pretrained molecular representations can also complement MOLPFN's graph-based input. Leveraging LimiX's ability to accept an arbitrary number of input features, we augment MOLPFN's atom-level feature grid with molecule-level representations from Mole-BERT or MolDeBERTa. With this additional information, MOLPFN achieves the best performance on all five regression benchmarks. Relative to the strongest TabPFNv3 baseline, it reduces error by 22.7% on FreeSolv, 8.8% on ESOL, 2.3% on Lipo, 5.8% on AqSol, and 38.3% on ZINC. We further isolate the contribution of atom-feature richness itself in an ablation over the number of atom-level feature dimensions (Appendix C.3)

Classification additionally tests transfer across task types. Although MOLPFN is pretrained only on synthetic regression tasks (Section 4.2), it transfers strongly to classification without classification-specific training (Table A2, section C.1). Using atom features alone, it remains within 1.9-3.3 ROC-AUC points of the strongest baseline across all four datasets. Incorporating MolDeBERTa representations further improves performance: MOLPFN achieves the best result on BBBP and is within 0.5–1.6 points of the strongest baseline on the remaining datasets.

5.3 FEW-SHOT EVALUATION ON FS-MOL

Evaluation protocol. FS-Mol evaluates adaptation to previously unseen biological assays from small labelled support sets (Stanley et al., 2021). Assays are partitioned into training, validation, and

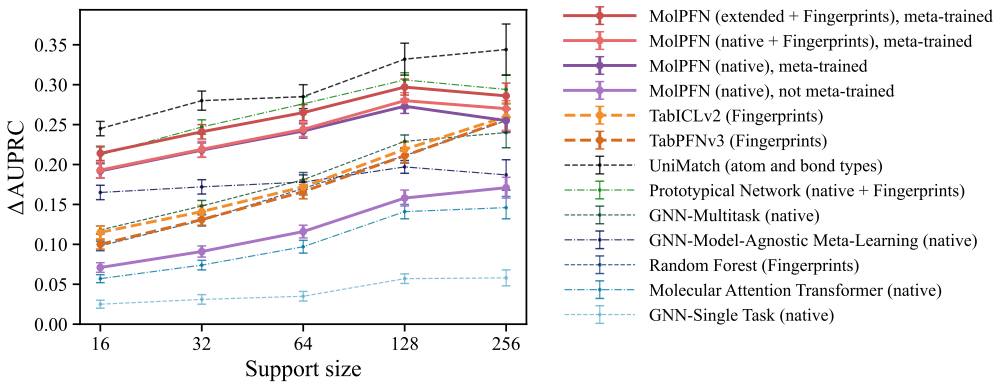


Figure 3: Average precision above the query-set positive rate ΔAUPRC averaged over the FS-Mol test tasks for support sizes $\{16, 32, 64, 128, 256\}$. We compare MOLPFN with established FS-Mol baselines (thin lines), the SOTA UniMatch and the frozen TabPFN v3 and TabICL v2 models (dash-dotted). $(\cdot)$ denotes the model input, where *native* denotes FS-Mol's 32-dimensional atom features, *extended* denotes the 40-dimensional atom representation described in section A5. Error bars show the standard error across tasks. The lines are sorted by their performance at support size 16.

test tasks, used respectively for learning shared model parameters, model selection, and evaluation. For each test assay, the model receives a labelled support set and is evaluated on a separate query set whose labels remain hidden during prediction.

Methods. We compare MOLPFN with the established FS-Mol baselines, which differ in how they use the labelled support set adapt to a test assay. Multitask GNNs are pretrained jointly across the training assays and subsequently fine-tuned on each test assay's support set, while GNN-MAML, based on Model-Agnostic Meta-Learning, learns across the training assays an initialization that is designed for rapid adaptation through a small number of gradient steps (Finn et al., 2017). Both therefore update model parameters when adapting to a new test assay. Prototypical networks (PN) instead embed the support examples and construct task-specific class prototypes, classifying queries by their distance to these prototypes (Snell et al., 2017). UniMatch uses attention-based matching between support and query representations obtained from a GNN, combined with task-specific gradient adaptation of its matching and output modules (Li et al., 2025). We additionally compare against TabPFNv3 (Grinsztajn et al., 2026) and TabICLv2 (Qu et al., 2026), which use the support set for perform frozen in-context prediction without task-specific parameter updates.

We evaluate MOLPFN in two settings. First, we use the synthetically pretrained model as is, with the native FS-mol atom features and no further training. Second, we train the graph adapters across the FS-Mol training assays while keeping the Limix backbone frozen (Appendix A.2), and compare three input configurations: native atom features, native atom features combined with molecular fingerprints used in FS-Mol, fixed-length representations encoding molecular substructures, and custom atom-level features defined in Table A5 combined with molecular fingerprints .

At evaluation time, all MOLPFN parameters are frozen. For each held-out test assay, MOLPFN jointly processes the labelled support set and query molecules in a single forward-pass, using the support examples solely for in-context conditioning, without the task-specific gradient updates as in GNN-MAML and Unimatch or prototype construction as in PN.

Evaluation Results. Figure 3 reports ΔAUPRC, defined as the area under the precision–recall curve (AUPRC) minus the query-set positive rate, across support sizes. Even without any FS-Mol-specific training, the synthetically pretrained MOLPFN extracts a positive few-shot signal from native atom features ($\Delta\text{AUPRC} = 0.071$ at support 16), showing that in-context adaptation transfers beyond the synthetic pretraining distribution without exposure to real FS-Mol assays. Training the graph adapters on the FS-Mol training assays substantially improves performance across support sizes (e.g. ΔAUPRC increases from 0.071 to 0.193 at support 16 using native features), showing that FS-Mol's training assays provide learnable prior information that can be captured through the graph

adapters alone while the LimiX backbone remains frozen. Adding molecular fingerprints provides a further improvement, with the meta-trained model reaching $\Delta\text{AUPRC} = 0.280$ at support 128.

Replacing FS-Mol's native atom features with our custom atom-level features (Appendix C.2) yields the strongest MOLPFN configuration throughout, reaching $\Delta\text{AUPRC} = 0.214$ at support 16 and 0.297 at support 128. At support 16, this is nearly twice the performance of the frozen fingerprint-based TabPFNv3 (0.101) and TabICLv2 (0.115), despite all three methods conditioning on the same fingerprint representation. This comparison highlights the contribution of FS-Mol-specific training of the graph adapters: providing a molecular fingerprint to a general-purpose frozen tabular foundation model alone does not recover the performance achieved when molecular representations and in-context prediction are connected through adapters trained across the FS-Mol assays.

Among the molecular few-shot baselines, UniMatch achieves the strongest performance, although MOLPFN remains close despite requiring no task-specific adaptation. MOLPFN is competitive with Prototypical Networks and outperforms GNN-MAML, GNN-Multitask, Random Forest, MAT, and GNN-ST across support sizes. At the lowest support size, MOLPFN slightly exceeds PN (0.214 vs. 0.211), while at larger support sizes PN remains marginally ahead (< 0.011 difference), with the two methods statistically indistinguishable given standard errors across support sizes. Unlike UniMatch, GNN-MAML, and PN, MOLPFN requires no per-assay optimization or task-specific predictor construction, performing inference through a single frozen in-context forward pass.

6 CONCLUSION

In this paper we introduced MOLPFN, to the best of our knowledge the first prior fitted network for molecular property prediction that operates directly on molecular graphs. MOLPFN augments a frozen LimiX backbone with graph adapters that integrate topology and atom-level information, while retaining the flexibility to incorporate arbitrary molecule-level features. As a PFN, it adapts to new molecular prediction tasks through in-context learning, without updating model parameters.

Our results show that graph-based molecular prediction can be learned from synthetic priors, that combine valence-aware molecular structures with structure-dependent atom features and graph-level regression targets, and transferred effectively to experimental data. Pretrained solely on synthetic regression tasks, MOLPFN matches or outperforms several strong baselines across the regression benchmarks using only atom-level features and molecular topology. Incorporating representations from pretrained molecular encoders as molecule-level features further improves performance, with MOLPFN outperforming all baselines across all five regression datasets. Despite being pretrained exclusively for regression, the same model transfers to classification without classification-specific training, performing competitively with or outperforming strong molecular and tabular baselines.

In the few-shot FS-Mol setting, synthetic pretraining alone transfers to unseen experimental assays, while training the graph adapters across FS-Mol training tasks substantially improves performance. Incorporating molecular fingerprints yields further gains, with MOLPFN approaching the strongest few-shot method, UniMatch, and remaining competitive with or outperforming the other established baselines. Notably, this performance is achieved without the per-assay optimization of UniMatch and GNN-MAML or the task-specific prototype construction of PN: MOLPFN requires only a single frozen in-context forward pass at deployment. Thus, new assays require neither per-assay fine-tuning nor construction of a task-specific predictor.

Overall, our results demonstrate the potential of prior-data fitted networks and in-context learning for molecular property prediction: MOLPFN requires training only the graph-specific components on top of a frozen foundation model, while adapting to a new property requires only providing labelled molecules as context rather than training a new predictor. With fewer than 4M trainable parameters and adaptation-free inference, MOLPFN offers a parameter-efficient approach to molecular prediction in data-scarce settings. An important next step is to exploit the probabilistic foundations of PFNs to obtain calibrated uncertainty estimates. More broadly, the ability of ICL to assess how predictions and uncertainty change as labelled molecules are added to the context could ultimately support experimental acquisition by identifying compounds whose measurement is expected to be most informative for the property of interest.

REPRODUCIBILITY STATEMENT

Code and documentation for reproducing the results of this paper are available at https://anonymous.4open.science/r/MolPFN/.

AI USE STATEMENT

In this work, we used generative AI tools to propose and refine hypotheses, design and provide feedback on research methodology and experiments, implement research methods, assist with translation, and interpret research results. We also used generative AI tools for finding related work, brainstorming and research ideation, editing and polishing the writing, and creating or editing software code. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for or assist with mathematical proofs, clean or reformat datasets, or conduct qualitative or thematic data analysis. We have reviewed all AI-assisted work. Relevant content in related work identified by AI was thoroughly read and evaluated by the authors. Research ideas suggested by LLMs were checked for soundness and originality through manual literature review. AI-assisted code was examined, tested for correctness, and reviewed by at least two authors, with changes tracked using version control. AI-drafted content was manually revised. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Blazej Banaszewski and Andrew W. Fitzgibbon. Monroe: A molecular foundation model for in-context probabilistic inference, 2026. URL https://arxiv.org/abs/2608.18982.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

Reinhard Diestel. *Graph Theory*, volume 173. Springer-Verlag, Berlin, Heidelberg, 2017.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

Hossam Eraqi et al. Molecular property prediction in the ultra-low data regime. *Communications Chemistry*, 8, 2025. doi: 10.1038/s42004-025-01592-1.

Dmitry Eremeev, Gleb Bazhenov, Oleg Platonov, Artem Babenko, and Liudmila Prokhorenkova. Turning tabular foundation models into graph foundation models. *arXiv preprint arXiv:2508.20906*, 2025.

Dmitry Eremeev, Oleg Platonov, Gleb Bazhenov, Artem Babenko, and Liudmila Prokhorenkova. Graphpfn: A prior-data fitted graph foundation model, 2026. URL https://arxiv.org/abs/2509.21489.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-3: Technical report, 2026. URL https://arxiv.org/abs/2605.13986.

Davy Guan, Lu Zhang, Asiri Wijesinghe, Allen Zhu, He Zhao, Helen Power, F. Hafna Ahmed, Andrew Warden, Cheng Soon Ong, and Daniel M. Steinberg. Can tabular in-context learners generalize to biomolecular property prediction?, 2026. URL https://arxiv.org/abs/2606.31126.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

John J. Irwin and Brian K. Shoichet. Zinc—a free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, 45(1):177–182, 2005. doi: 10.1021/ci049714+. URL https://doi.org+.

Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), January 2011. ISSN 1550-2376. doi: 10.1103/physreve.83.016107. URL http://dx.doi.org/10.1103/PhysRevE.83.016107.

Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jian Chen, Gang Fu, Asta Gindulyte, Lonqi Han, Jiyao He, Siqi He, Benjamin A. Shoemaker, Jiyao Wang, Bo Wang, Jianqiang Yu, Jian Zhang, Qingzhou Zhang, David C. Nickle, and Stephen H. Bryant. Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, 2019. doi: 10.1093/nar/gky1033. URL https://doi.org.

Ruifeng Li, Mingqian Li, Wei Liu, Yuhua Zhou, Xiangxin Zhou, Yuan Yao, Qiang Zhang, and Hongyang Chen. UniMatch: Universal matching from atom to task for few-shot drug discovery. In *International Conference on Learning Representations*, 2025.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net.

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

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in environmental science*, 3:80, 2016.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Gabriel Bianchin de Oliveira and Fahad Saeed. Moldeberta: Foundational model for physicochemical and substructure-informed molecular representation learning. *bioRxiv*, 2026. doi: 10.64898/2026.02.15.706011. URL https://doi.org/10.64898/2026.02.15.706011.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2nd edition, 2009.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, Cambridge, MA, 2017.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data, 2025. URL https://arxiv.org/abs/2502.05564.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model. *ArXiv*, abs/2602.11139, 2026. URL https://api.semanticscholar.org/CorpusID:285470584.

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

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 4077–4087. Curran Associates, Inc., 2017. URL https://neurips.cc.

Murat Cihan Sorkun, Abhishek Khetan, and Sener Er. Aqsoldb, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. *Scientific Data*, 6(1):1–8, 2019. doi: 10.1038/s41597-019-0151-1.

Megan Stanley, John F Bronskill, Krzysztof Maziarz, Hubert Misztela, Jessica Lanini, Marwin Segler, Nadine Schneider, and Marc Brockschmidt. FS-mol: A few-shot learning dataset of molecules. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=701FtuyLlAd.

Megan Stanley, John F. Bronskill, Krzysztof Maziarz, Hubert Misztela, Jessica Lanini, Marwin H. S. Segler, Gisbert Schneider, and Marc Brockschmidt. Limitations of representation learning in small molecule property prediction. *Nature Communications*, 14:7043, 2023. doi: 10.1038/s41467-023-41967-3.

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.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical Science*, 9:513–530, 2017.

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

Xingxuan Zhang, Gang Ren, Han Yu, et al. Limix: Unleashing structured-data modeling capability for generalist intelligence. *arXiv:2509.03505*, 2025.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

A Appendix

A.1 Pretraining details

Optimization. We optimize with AdamW (Loshchilov & Hutter, 2019) using a weight decay of 0.1 and gradient-norm clipping at 1.0. The learning rate follows a cosine annealing schedule (Loshchilov & Hutter, 2017) with linear warmup over the first 1000 steps and a peak value of 10^{-3}. Training runs for $3{,}364$ optimizer steps, each accumulating gradients over 20 micro-steps, with one synthetic dataset per micro-step per GPU; on 4 NVIDIA A100 GPUs forward and backward passes are performed under bfloat16 autocast

Synthetic tasks and objective. Each micro-step samples a new dataset from the graph-level prior. Its molecules are randomly split into 80% context and 20% query examples. Regression targets are standardized using the context mean and standard deviation; the loss is the query mean squared error. Gradient-based pretraining uses exclusively synthetic tasks. The use of real molecular structures for topology calibration is specified in Appendix B;

Model selection. Every 20 steps we evaluate the EMA weights in the in-context learning regime on a held-out real molecular dataset — the ZINC penalized-logP regression benchmark (Irwin & Shoichet, 2005; Dwivedi et al., 2023) — 500 pairing query molecules from its validation split with 2000 context molecules from its training split, and we retain the checkpoint with the lowest validation error. ZINC validation labels thus inform model selection but are not used for gradient updates in synthetic pretraining. No test split is used for checkpoint selection.

A.2 Meta Training Details

We meta-train the graph pooler on the FS-Mol training assays while the LimiX-16M backbone stays frozen. The trained module is the small graph pooler placed in front of the backbone (one message-passing layer, multi-view pooling and the per-block graph adapters, 3.97M parameters). All three variants below share the same procedure and differ only in the input features. Each is initialised from the synthetically pretrained pooler used in the first setting.

Episodes. Each training step uses episodes built from the 4,938 FS-Mol training assays. An episode samples one assay uniformly at random and a support size from $\{16, 32, 64, 128\}$ uniformly at random. It then draws a stratified support/query split, in the same way as at test time, and caps the query set at 128 molecules. Support and query molecules are passed to the model as one in-context sequence. The loss is the cross-entropy of the query predictions, and the labels of the query molecules are hidden from the model. Assays (fewer than four molecules or a single class) are never sampled.

Optimization. We use AdamW with peak learning rate 10^{-4}, weight decay 0.1, gradient-norm clipping at 1.0 and bfloat16 autocast. The learning rate is warmed up linearly over the first 10% of the steps and then decays with a cosine schedule. We keep an exponential moving average of the pooler weights with decay 0.98, and this average is used for validation and for all reported results. Training runs for at most $20{,}000$ optimiser steps. Each step accumulates gradients over 20 episodes on each of 4 GPUs (80 episodes per step); gradients are averaged across GPUs, and every GPU samples its own episodes.

Model selection. Every 200 steps we evaluate the averaged weights in context on the 40 FS-Mol validation assays, with support sizes 16 and 128 and 5 random splits each. We keep the checkpoint with the highest mean ΔAUPRC at support size 16 and stop early if it has not improved for 25 consecutive evaluations. The validation assays are disjoint from both the training and the test assays.

Variants.

- **Native features.** The 32 atom features stored in FS-Mol and its bond graph.
- **Native features + fingerprint.** The same inputs, plus a per-molecule ECFP count fingerprint folded from 2048 to 128 bins. It is encoded by the frozen LimiX feature encoder and added as extra feature groups to the pooled molecule representation, before the transformer.
- **RDKit features + fingerprint.** The 40-dimensional atom features of Table A5, computed from the SMILES.

A.3 The KNN Baseline

Model. We use the scikit-learn implementations `KNeighborsClassifier` and `KNeighborsRegressor` (Pedregosa et al., 2011). Each feature is first standardized to zero mean and unit variance, with the scaler fitted on the training split only. For the cosine distance, the standardized vectors are additionally ℓ_2-normalized and compared with the Euclidean distance, which gives the same neighbour ranking as cosine distance. Neighbours are weighted uniformly.

Table A1: Selected checkpoint per variant (validation assays only).

Variant	Selected step	Validation ΔAUPRC (support 16)
Native features	17,000	0.225
Native features + fingerprint	14,000	0.224
RDKit features + fingerprint	10,200	0.232

For classification, the predicted positive-class probability is the fraction of positive labels among the k nearest training molecules. For regression, the prediction is the mean target of those neighbours.

Hyperparameter search. For each dataset and embedding model we run an grid search over

$$k \in \{1, 3, 5, 7, 9, 11, 15, 21, 31, 51\}, \qquad \text{distance} \in \{\text{Euclidean}, \text{cosine}\},$$

which gives 20 configurations; values of k larger than the training set are dropped. Each configuration is fitted on the full training split and scored on the fixed validation split (a single hold-out split, without cross-validation). We keep the configuration with the best validation score: the highest ROC-AUC for the classification datasets (BACE, BBBP, ClinTox, SIDER), the lowest RMSE for ESOL, FreeSolv and Lipophilicity, and the lowest MAE for ZINC and AQSOL. We report its score on the held-out test split. The test split is never used for selection.

For the multi-task datasets (ClinTox, SIDER), each task is treated as an independent binary problem with its own hyperparameter search, and we report the mean of the per-task test scores.

B Graph-Level Prior: Implementation Details

B.1 Topology Sampling

Dataset and graph sizes. For the training configuration considered here, the number of molecules is sampled log-uniformly between $1{,}000$ and $2{,}000$. Molecular skeletons are sampled independently conditional on the topology parameters and combined by disjoint union. The nominal size is 55 nodes, independently perturbed for each molecule by a factor sampled uniformly from $[0.85, 1.15]$. The heavy-atom count is $\max(2, \mathrm{round}(0.42n))$, where n is the perturbed nominal size. The final total size depends on the hydrogen-completion step.

Elements and bond orders. Heavy-atom identities are sampled with probabilities

$$(p_{\mathrm{C}}, p_{\mathrm{N}}, p_{\mathrm{O}}, p_{\mathrm{F}}) = \frac{1}{48}(32, 7, 8, 1), \tag{5}$$

and nominal valence budgets $(4, 3, 2, 1)$, respectively. These probabilities are fixed design choices of the prior. Bond orders are sampled from $\{1, 2, 3\}$ with base weights $(0.8, 0.15, 0.05)$, restricted and renormalized according to the remaining endpoint budgets.

Tree growth. Each new heavy atom attaches to a uniformly selected earlier atom with available valence. The sampled bond order is subtracted from both endpoint budgets. If no eligible earlier atom remains, a single-bond fallback preserves connectivity but can exceed the nominal valence budget. The sampler is therefore valence-aware rather than strictly valence-constrained.

Ring closure. For n_{h} heavy atoms, we target

$$q = \max\left(0,\ \mathrm{round}\left(\frac{2.15 n_{\mathrm{h}}}{2}\right) - (n_{\mathrm{h}} - 1)\right) \tag{6}$$

additional edges beyond the initial tree, corresponding to a target average heavy-atom degree of 2.15. Eligible endpoints have positive remaining valence and a shortest-path distance between two and six in the current graph; connecting them introduces a cycle of length three to seven.

We weight cycle-length proposals using a graph-theoretic cycle-basis analysis of the $10{,}000$-molecule ZINC training split (Irwin & Shoichet, 2005; Dwivedi et al., 2023). The observed percentages for lengths three through seven are approximately $(2.28, 0.60, 30.71, 65.00, 1.18)$. For each

proposed source atom, eligible partners are grouped by shortest-path distance. We sample a group according to these weights, renormalized over the available groups, then select a partner uniformly within that group. Sampling stops after q successful additions or $20q + 50$ attempts.

An additional analysis of $9{,}999{,}861$ successfully parsed molecules from a PubChem-derived collection (Kim et al., 2019) supports this preference for short cycles. Using the same cycle-basis method, five- and six-membered cycles account for 94.81% of basis cycles, and lengths three through seven account for 99.20%. The mean number of independent cycles per molecule is 2.79 ± 1.70. These statistics provide broader empirical support for the construction; the sampler retains the ZINC-derived proposal weights.

The weights guide individual closure proposals rather than enforce a particular output cycle-length distribution. All cycle-length statistics refer to the computed graph cycle bases, which need not coincide with chemically perceived rings.

Hydrogen completion. Each positive remaining unit of valence is filled by a degree-one hydrogen node. Element identities and bond orders are then discarded, leaving a connected, simple graph for attribute generation.

B.2 SCM Configuration

Mechanism and inputs. A new neural SCM is instantiated for each dataset and shared across all molecules. It uses three transformation layers, $\tanh$ nonlinearities, and additive Gaussian noise with standard deviation 0.02 in the nonlinear blocks. The configuration specifies a hidden width of 32; causal-mode capacity requirements increase the effective width to 33 for 16 observed features and one target.

Ordinary nodes receive standard-normal exogenous causes. The number of causes follows GraphPFN's log-scaled truncated-normal hyperparameter distribution, with mean hyperparameter between 1 and 12, standard-deviation hyperparameter between 0.01 and 1, a lower bound of one, and integer rounding. Log-degree and log-PageRank descriptors of the augmented graph are each enabled independently with probability 0.5.

Graph-dependent transformations. Each layer applies a linear transformation and computes a graph aggregation of the transformed values. A dimension-wise Bernoulli mask selects between local and aggregated outputs. The selection probability is sampled per dataset from $\{0, 0.1, \ldots, 1\}$. The aggregation operator is sampled from GCN, GraphSAGE-mean, GraphSAGE-min, GraphSAGE-max, and graph-transformer attention. At probability zero, message passing is disabled, although enabled structural descriptors can still introduce topology dependence.

Feature and target extraction. Before evaluating the SCM, each molecule receives a virtual node with bidirectional edges to all its atoms. The virtual node's exogenous causes are zeroed; structural inputs and additive noise remain active. In causal mode, the 16 observed feature coordinates are selected from a contiguous window of concatenated hidden activations, and the target coordinate is taken from the final layer. We retain features at ordinary atom nodes and the target at the virtual node. Generated features undergo the feature-postprocessing pipeline, while regression targets are standardized using context statistics during training.

Fixed and resampled quantities. Graph realizations, SCM weights, causes, noise, feature-selection indices, cause counts, aggregation operators, mixing probabilities, and structural-feature inclusion vary between datasets or realizations. The nominal molecular size, element probabilities, valence budgets, ring-closure weights, and SCM architecture settings remain fixed.

C Experimental Setup Details

C.1 Classification ICL

Table A2: Classification performance measured by ROC-AUC (%). Results are reported as mean ± standard deviation. Higher is better. Best results are shown in bold.

Method	**BBBP**	**BACE**	**ClinTox**	**SIDER**
Split	*Scaffold*	*Scaffold*	*Scaffold*	*Scaffold*
Number of tasks	*1*	*1*	*2*	*27*
Nearest-neighbour baseline				
MolBERT	65.3 ± 0.00	80.2 ± 0.00	68.4 ± 1.0	65.8 ± 8.8
MolDeBERTa	70.3± 0.00	83.7 ± 0.00	91.4 ± 1.9	62.7 ± 6.2
TabPFN v3				
Mole-BERT	70.5 ± 1.1	83.7 ± 0.6	70.7 ± 0.3	67.2 ± 7.4
MolDeBERTa	75.8 ± 0.5	83.9 ± 0.4	93.7 ± 1.0	67.7 ± 7.6
TabICL v2				
Mole-BERT	70.0 ± 0.4	83.6 ± 0.5	66.6 ± 1.3	**68.9** ± 6.3
MolDeBERTa	74.9 ± 0.9	82.1 ± 1.1	**95.6** ± 0.5	67.5 ± 7.5
LimiX				
Mole-BERT	69.4 ± 0.3	**84.6** ± 0.4	69.8 ± 2.6	66.3 ± 6.1
MolDeBERTa	74.2 ± 0.5	82.8 ± 0.4	94.1 ± 0.9	66.9 ± 7.6
MolPFN *(ours)*				
No molecular encoder	74.3 ± 0.4	82.7 ± 0.2	93.6 ± 0.4	66.0 ± 7.5
+ Mole-BERT	71.6 ± 0.1	82.7 ± 0.1	90.8 ± 0.8	67.1 ± 6.0
+ MolDeBERTa	**76.0** ± 0.1	82.9 ± 0.1	94.3 ± 0.5	67.3 ± 7.6

Classification inference. For binary classification, we retain the regression-pretrained molecular adapters and use LimiX's frozen classification target encoder and decoder. Context labels are encoded as class identities, while query labels remain masked. The final query target-token representations are decoded into two class logits, which are converted to probabilities by softmax. No classification-specific parameter updates are performed for these evaluations.

Transfer to classification. Although the molecular adapters are pretrained using only synthetic regression tasks, the same checkpoint supports classification through LimiX's frozen classification encoder and prediction head. Without an auxiliary pretrained molecular encoder, MolPFN achieves ROC-AUC scores within 1.9–3.3 percentage points of the strongest reported baseline across the four classification datasets. Adding MolDeBERTa representations narrows these gaps to 0.5–1.7 points.

These results demonstrate transfer of regression-pretrained molecular adapters to classification, but do not establish consistent superiority over embedding-based tabular predictors. Auxiliary representations are not uniformly beneficial: Mole-BERT augmentation reduces ClinTox ROC-AUC from 93.6 to 90.8. Classification-specific adapter pretraining is therefore a natural direction for further investigation.

C.2 Datasets and Representations

Table A3 identifies the prediction targets and reported metrics. The MoleculeNet evaluations use scaffold partitions; AqSolDB and ZINC use the established partitions adopted in the experiments. Labeled training molecules provide context, and test molecules provide queries. These downstream atom representations need not have the same feature count or semantics as the synthetic SCM attributes.

Atom representations. The nine-column atom representation contains atomic number, chirality, total degree, formal charge, total hydrogen count, radical-electron count, hybridization, aromaticity, and ring membership. The extended representation appends 31 descriptors computed from molecular structure, for a total of 40 columns. Table A5 lists the added descriptors. The one-, nine-, and forty-column variants are compared in Appendix C.3. The FS-Mol figure additionally reports the benchmark's native 32-dimensional atom representation.

Table A3: Prediction tasks in the nine-dataset evaluation. Endpoint counts are shown for classification datasets.

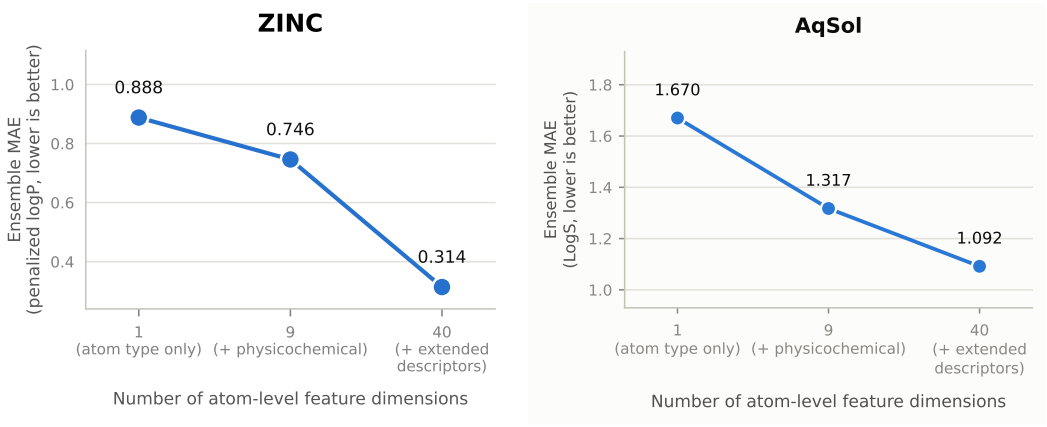
Dataset	Target	Task	Metric
FreeSolv	Hydration free energy	Regression	RMSE
ESOL	Aqueous solubility	Regression	RMSE
Lipophilicity	Octanol/water logD	Regression	RMSE
AqSolDB	Aqueous solubility	Regression	MAE
ZINC	Penalized logP	Regression	MAE
BBBP	Blood–brain barrier penetration	Binary	ROC-AUC
BACE	BACE inhibition	Binary	ROC-AUC
ClinTox	Clinical toxicity / approval	2 binary endpoints	ROC-AUC
SIDER	Adverse drug reactions	27 binary endpoints	ROC-AUC

Table A4: Topology statistics for the nine molecular datasets and the synthetic prior. N denotes the number of molecules; the prior statistics pool $5{,}334$ molecules from four synthetic dataset draws. Node and edge counts refer to heavy atoms and heavy-atom bonds. Independent cycles are counted as $|E|-|V|+c$, where c is the number of connected components. Entries report ranges and means $\pm$ standard deviations. Distinct atoms denotes the number of unique element types across each dataset.

Dataset	N	Nodes (min–max, mean$\pm$std)	Edges (min–max, mean$\pm$std)	Rings (min–max, mean$\pm$std)	Distinct atoms
FreeSolv	642	1–24 (8.72 ± 4.19)	0–25 (8.39 ± 4.78)	0–5 (0.67 ± 0.81)	9
ESOL	1,128	1–55 (13.29 ± 6.88)	0–62 (13.68 ± 7.92)	0–8 (1.39 ± 1.31)	9
Lipophilicity	4,200	7–115 (27.04 ± 7.46)	7–118 (29.50 ± 8.24)	0–13 (3.46 ± 1.10)	12
AqSolDB	9,980	1–388 (17.38 ± 12.24)	0–400 (17.63 ± 13.10)	0–36 (1.50 ± 1.59)	65
ZINC	12,000	9–37 (23.16 ± 4.48)	8–42 (24.92 ± 5.27)	0–7 (2.76 ± 1.00)	9
BBBP	2,039	2–132 (24.06 ± 10.59)	1–145 (25.95 ± 11.71)	0–14 (2.97 ± 1.51)	13
BACE	1,513	10–97 (34.09 ± 8.52)	10–101 (36.86 ± 8.76)	0–7 (3.77 ± 0.88)	8
ClinTox	1,479	1–136 (26.25 ± 15.79)	0–143 (27.99 ± 17.07)	0–12 (2.75 ± 1.78)	29
SIDER	1,427	1–492 (33.64 ± 47.88)	0–505 (35.36 ± 49.21)	0–46 (2.97 ± 2.41)	40
Prior (ours)	5,334	20–26 (23.12 ± 1.99)	22–28 (25.12 ± 1.99)	3–3 (3.00 ± 0.00)	—

Table A5: Atom descriptors totaling 40-dimensional atom features.

Category	Descriptors	Count
Atom identity	Atomic number	1
Stereochemistry	Chirality	1
Connectivity	Total degree; total hydrogen count	2
Electronic properties	Formal charge; number of radical electrons; hybridization	3
Aromaticity and rings	Aromaticity indicator; ring-membership indicator	2
Valence	Total valence	1
Bond environment	Numbers of single, double, triple, aromatic, conjugated, and ring bonds	6
Ring environment	Membership in 3-, 4-, 5-, and 6-membered rings; smallest ring size; number of rings containing the atom	6
Chemical descriptors	Gasteiger charge; Crippen logP and molar refractivity contributions; polar surface-area contribution; electrotopological-state index	5
Chemical roles	Hydrogen-bond donor and acceptor indicators	2
Element properties	Atomic mass, covalent radius, van der Waals radius, and number of outer-shell electrons	4
Neighbor composition	Counts of carbon, nitrogen, oxygen, sulfur, phosphorus, halogen, and heteroatom neighbors	7
Total		40



(a) ZINC (b) AqSolDB

Figure A1: Performance of MOLPFN as additional atom features are included in the input: (a) ZINC and (b) AqSolDB.

C.3 ATOM FEATURE ABLATION EVALUATION

An ablation on ZINC and Aqsol isolates the effect of atom-feature richness directly: replacing the native single-column atom-type encoding with a 9-column MoleculeNet-style atom encoding (atomic number, chirality, degree, formal charge, hydrogen count, radical-electron count, hybridization, aromaticity, ring membership) reduces MAE from 0.888 to 0.746, a 16.0% relative improvement, with nothing else in the pipeline changed. This confirms that MOLPFN's regression gains are substantially driven by the atom-level features themselves: lowering the number of atom features available to the model degrades performance accordingly, Figure A1.