Representing Molecules as Random Walks Over Interpretable Grammars

Michael Sun [1] **Minghao Guo** [1] **Weize Yuan** [2] **Veronika Thost** [3] **Crystal Elaine Owens** [1]
Aristotle Franklin Grosz [4] **Sharvaa Selvan** [5] **Katelyn Zhou** [6] **Hassan Mohiuddin** [5] **Benjamin J Pedretti** [4]
Zachary P Smith [4] **Jie Chen** [3] **Wojciech Matusik** [1]

Abstract

Recent research in molecular discovery has primarily been devoted to small, drug-like molecules, leaving many similarly important applications in material design without adequate technology. These applications often rely on more complex molecular structures with fewer examples that are carefully designed using known substructures. We propose a data-efficient and interpretable model for representing and reasoning over such molecules in terms of graph grammars that explicitly describe the hierarchical design space featuring motifs to be the design basis. We present a novel representation in the form of random walks over the design space, which facilitates both molecule generation and property prediction. We demonstrate clear advantages over existing methods in terms of performance, efficiency, and synthesizability of predicted molecules, and we provide detailed insights into the method's chemical interpretability. Code is available at `https://github.com/shiningsunnyday/polymer_walk`.

1. Introduction

Property-driven molecular discovery represents a challenging application with great potential benefits for society, and this is reflected in the large amount of research conducted in the machine learning community on this topic in recent years (Sawlani, 2024). Yet, most of the research focuses on small, drug-like molecules, while many classes of more complex molecules have been largely neglected. Materials designed for applications such as gas-separation membranes or photovoltaics, which are critical for a sustainable future, often have specific distributions of molecule structure that differ significantly from typical drug-like molecules. In addition, the specificity of the designs and use cases, and the considerable cost of practical experiments, make it often a scenario that is scarce in both data and labels; for example, datasets of ≈ 300 molecules or less are not uncommon (Wang et al., 2018; Lopez et al., 2016; Helma et al., 2001). As a consequence, materials science has not yet fully exploited the potential of machine learning methods (Karande et al., 2022; Wang & Wu, 2023). We focus on such challenging datasets that feature complex molecules containing functional groups and structural motifs which are applied in multiple diverse, real-world application scenarios.

Our goal is to represent and reason about molecules in a data-efficient and interpretable way. Domain-specific datasets typically exhibit distinct motifs and functional groups, which serve as structural priors in our molecular representation. Previous works show that structural priors are highly advantageous for applications that require data efficiency (Rogers & Hahn, 2010; Xia et al., 2023a; Shui & Karypis, 2020; Jiang et al., 2022; Yang et al., 2022). We propose a novel approach to molecular discovery that is tailored to more complex molecules and low-data scenarios and builds upon the above insights. The idea is to start from a set of expert-defined motifs[1] and learn a context-sensitive grammar over the space of motifs. The *novelty of this work lies in our representation and learning of this grammar*.

We define a *motif graph* – a hierarchical abstraction of the molecular design space induced by the given data, where each node is a motif and each edge represents a possible attachment between a pair of motifs. Our main technical contribution is an efficient and interpretable parameterization over the context-sensitive grammar induced by the design space, and the description of a molecule as a random walk of context-sensitive transition rules. Our representation of molecules combines the quality of representation learning with the interpretability of a rule-based grammar.

In terms of quality, we demonstrate our grammar representa-

[1]MIT CSAIL [2]MIT Chemistry [3]MIT-IBM Watson AI Lab, IBM Research [4]MIT Chemical Engineering [5]MIT [6]Wellesley. Correspondence to: Michael Sun <msun415@csail.mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] Note that our method works with any given set of motifs (e.g., we can apply one of the more simple algorithms used in existing works), but our evaluation shows that certain applications benefit from high-quality domain knowledge.

tion suits applications characterized by designer molecules. We select datasets that reflect real-world settings of experimentally curated designs of molecules with complex, modular sub-structures characterized by functional groups known or hypothesized to yield high target properties.

In terms of interpretability, our grammar representation is special in two ways. As an indirect consequence of supervised learning, our model produces visually discernible clusters according to distinctive structural features within the dataset. More importantly, our compact, context-sensitive *grammar allows for discovering design rules* that reveal the design principles used during the creation of the dataset.

- Our method largely outperforms pretrained and traditional methods for molecular property prediction. It is competitive with a state-of-the-art graph grammar system for chemistry (Guo et al., 2023b) in terms of quality while being an order of magnitude more runtime efficient.
- Our method's interpretable representations reveal deeper insights into relationships implicit in the data, explain the model's reasoning, and lead to novel scientific insights.
- Our method produces promising molecule generations, in particular, producing diverse designs that are synthesizable at a significantly higher rate than the state-of-the-art data-efficient generative model, DEG (Guo et al., 2023a).
- Finally, made possible by our method's interpretability, our approach enables close collaboration with domain experts. In particular, we devised and executed feasible, practical, and semi-automated workflows with experts for fragmenting molecules, constructing the design space, and interpreting the results.

2. Related Works

Motif-based molecular property prediction. ECFP embeddings (Rogers & Hahn, 2010), which capture relevant ego-graphs present in a molecule in bit vectors, represent a motif-based encoding. ECFP embeddings in combinations with simple predictors (e.g., XGBoost) have been competitive on small datasets (Xia et al., 2023a). In our evaluation, we show that our model is similarly *data-efficient* but delivers a better predictive performance, owing to the use of graph-based representations. In light of the good performance of ECFPs, it is not surprising that the recently developed subgraph graph neural networks (GNNs) report competitive performance in molecular property prediction when using ego-graphs as subgraphs (Frasca et al., 2022); we consider ESAN (Bevilacqua et al., 2022) in our evaluation. However, existing models usually apply subgraphs rooted at all individual nodes rather than a set of more coarse-grained, potentially complex, domain-specific subgraphs. Other recent work that integrates motifs to improve out-of-distribution detection similarly lacks this dimension of modeling (Yang et al., 2022).

A few closely related works have recently proposed molecular graph representations where the relations between motifs are explicitly represented, together with corresponding models (Shui & Karypis, 2020; Jiang et al., 2022). Our work is different from theirs in two aspects. First, we show that commonly used automatic approaches for motif extraction are not sufficient for property prediction over several kinds of more complex molecules, and that custom motifs given by domain experts yield better performance. It allows for biasing the model towards known structure-activity relationships or the expert's hypotheses (e.g., fragments known or assumed to be critical for the property under consideration). Second, to the best of our knowledge, their motif graph representations do not model the context sensitivity explicitly (e.g., HM-GNN's motif graph (Shui & Karypis, 2020) connects two motifs based on co-occurence in a molecule only).

Molecule representation by grammars. Recent work has shown that such grammars represent a data-efficient way for representing molecules and yield SOTA results (Guo et al., 2023a;b). In a nutshell, this is achieved by explicitly representing the training data's design space in terms of learnt motifs, in the form of a graph grammar. *Grammars naturally allow for generating novel molecules in the given design space*. Yet, obtaining production rules involves either manual definition (Krenn et al., 2020; Guo et al., 2022; Nigam et al., 2021) or a significant complexity to automatically learn (Guo et al., 2023a; Kajino, 2019), where the training times for downstream tasks are considerable (see Figure 5). Further, the learnt substructures sometimes lack a chemical interpretation, and grammar derivations often produce chemically invalid structures (Guo et al., 2023a), so the natural potential of symbolic methods for interpretability and validity is lost, although such elements are critical for expert validation and for gaining scientific insights. *We propose a novel way for representing and learning such context-sensitive grammars*, over a design space informed by chemical motifs. This approach results in order-of-magnitude differences in runtime and enhances chemical interpretability.

Other works for molecular representation learning. There are various other non-motif based approaches that we compare to in our evaluation, including (pre-trained) GNNs (Hu et al., 2020; Xia et al., 2023b), motif-based pre-training approaches designed for semi- or unsupervised learning (Xia et al., 2023b), and molecular few-shot learning including the SOTA, which relies on modeling the domain expert's reasoning in terms of related molecule contexts using associative memories (Schimunek et al., 2023). Central to our method is the connection between random walks and graph diffusion, established methods that have been particularly effective to model graph structures through physics-inspired processes (Thanou et al., 2017). Other related works and more detailed discussions can be found in Appendix C.

3. An Interpretable, Grammar-based Molecule Representation and Efficient Learning

Our method employs a *graph grammar*, which is composed of a set of predefined molecular motifs and a set of transition rules. The motifs are devised either through automatic generation or manual curation and are interconnected following the transition rules to assemble into a complete molecular structure. Following (Guo et al., 2023a;b), a grammar $\mathcal{G} = (\mathcal{N}, \Sigma, \mathcal{P}, \mathcal{X})$ contains a set $\mathcal{N}$ of non-terminal nodes, a set Σ of terminal nodes representing chemical atoms, and a starting node $\mathcal{X}$. The generation of molecular graphs is described using a set of production rules, $\mathcal{P} = \{p_i | i = 1, \ldots, k\}$. Each rule, p_i, is defined by a transformation from a left-hand side (LHS) to a right-hand side (RHS), with both sides being graphs. The process starts from an initial empty graph $\mathcal{X}$, and a molecule is constructed by iteratively applying a rule from $\mathcal{P}$, where the LHS of the selected rule matches a subgraph within the current graph. This selected subgraph is then replaced by the corresponding RHS of the rule.

Random Walk Grammar. We introduce random walk grammar, characterized by a specific condition where the LHS of each rule differs from its RHS by exactly a motif. Such a design ensures that the generation of a molecule is a progressive process, where in each step, a new subgraph is attached to the existing graph. We implement the grammar using a compact *motif graph* G (Fig. 1 (b)); the nodes are the motifs and each edge describes the application of a transition rule.

We highlight two novelties of this work:

1. Molecules are represented as random walks over connected subgraphs of G (Fig. 1 (a)). This representation is explicit, compact, and interpretable.
2. The context-sensitive grammar over G is learnable from a given training dataset by optimizing parameters that determine the prior and adjusted edge weights of G. These weights parameterize the transition probabilities, thereby influencing the molecular representation and facilitating the learning of context-sensitive rules, which we elucidate in our analysis section.

We demonstrate the utility of our grammar-based representation for both molecular generation and property prediction tasks. The main steps of our workflow are as follows.

Motif-based Molecule Fragmentation. Our method builds upon a given molecule fragmentation. More specifically, given a dataset $D = \{M^{(i)} := (V_{M^{(i)}}, E_{M^{(i)}})\}_{i=1}^{|D|}$, a fragmentation of $M^{(i)}$ is a collection of disjoint molecular graphs $\{F_j^{(i)}\} := \{(V_j^{(i)}, E_j^{(i)})\}$ such that $\sqcup_j V^{(i)} = V_{M^{(i)}}$. Letting $g(v)$ denote the node-induced subgraph of g by v, $F_j^{(i)}$ is the subgraph of $M^{(i)}$ induced by $V_j^{(i)}$. When $F_j^{(i)}$ is a chemical motif, it is essential to know the possible contexts within which $F_j^{(i)}$ occur, because the behavior of one substructure is often influenced by neighboring structures[2]. Specifically, given neighboring fragments j_1, j_2, i.e. $\exists e \in E_{M^{(i)}}$ s.t. $e \notin E_{j_1}^{(i)}, e \notin E_{j_2}^{(i)}$ and $e \in M^{(i)}(V_{j_1}^{(i)} \cap V_{j_2}^{(i)})$, then we can use automatic rules R_D to infer the "context" of j_1: $c_{j_2}^{(j_1)} := R_D(V_{j_1}^{(i)}, V_{j_2}^{(i)})$ s.t. $c_{j_2}^{(j_1)} \subseteq V_{j_2}^{(i)}$ and $M^{(i)}(V_{j_1}^{(i)} \sqcup c_{j_2}^{(j_1)})$ is connected. The same rule is applied in reverse to obtain $c_{j_1}^{(j_2)}$. The descriptions and examples for dataset-specific rules are given in Appendix A.1.

There are various automated methods to obtain such a fragmentation (e.g., (ChemAxon; Degen et al., 2008; Jin et al., 2020)); some are integrated in the commonly used RDKit package (Landrum, 2016). Nevertheless, we found that complex molecular datasets often benefit from fragmentations and rules tailored to the application domain, in the sense that they may better capture known domain knowledge and provide a strong structural prior. For this reason, we also designed and executed feasible, practical workflows for annotating molecules and extracting the motifs.

Motif Graph Construction. Given a set of motifs, V, we describe our hierarchical abstraction over V. $G = (V, E)$ is a directed multigraph. Each $v \in V$ contains both the motif graph g_v and $\{v_{r_l}\}$, denoting the possible "contexts" for attaching g_v to another motif; that is, $\forall l, v_{r_l} \subseteq N(g_v)$, with $N(g)$ denoting the set of atom nodes of graph g. $v_R := \cup_l v_{r_l}$, and $\emptyset \neq v_B := N(g_v) \setminus v_R$. Denoting $\sim$ to be the isomorphism relation, we construct E by matching every pair of motifs u, v and their contexts (l_1, l_2) by finding corresponding subgraphs in u_B and v_B to match $u_{r_{l_1}}$ and $v_{r_{l_2}}$, as shown in Fig. 1 (c). Specifically, $(u, v, e_{l_1,l_2}) \in E \iff \exists b_2 \subseteq u_B, b_1 \subseteq v_B$ such that:

$$g_u(u_{r_{l_1}}) \sim g_v(b_2) \tag{1}$$

$$g_v(v_{r_{l_2}}) \sim g_u(b_1) \tag{2}$$

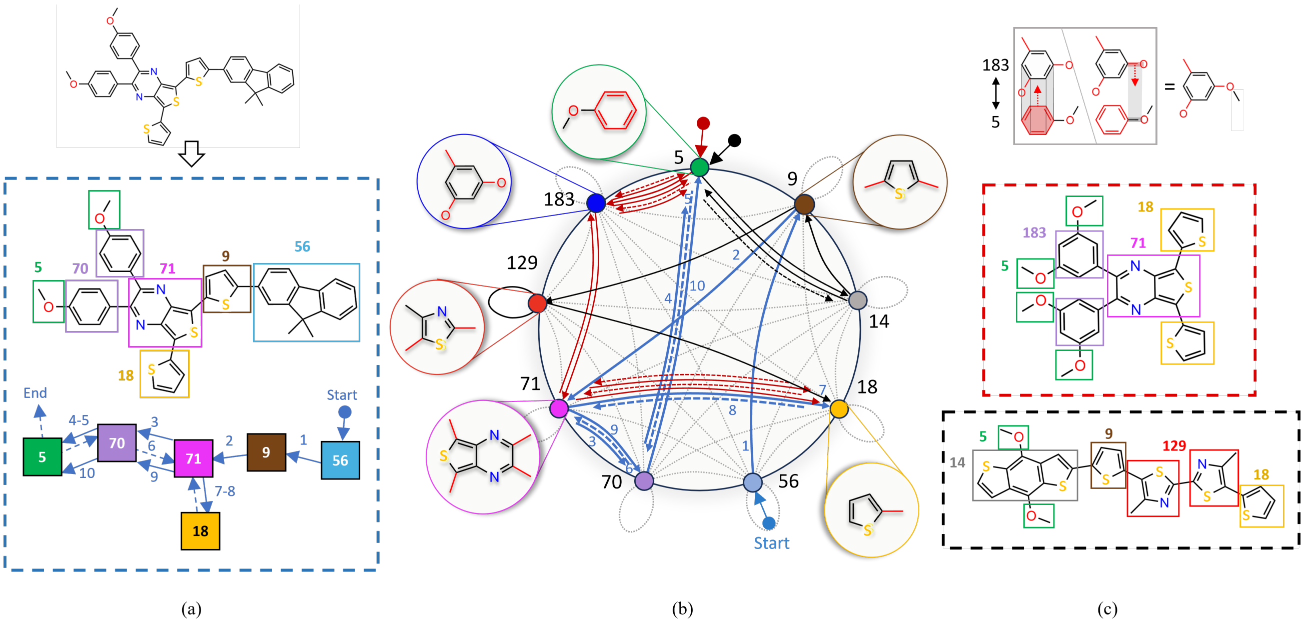
$$g_u(u_{r_{l_1}} \cup b_1) \sim g_v(b_2 \cup v_{r_{l_2}}) \tag{3}$$

$$g_u(u_{r_{l_1}} \cup b_1) \text{ is connected} \tag{4}$$

e_{l_1,l_2} is attributed with $u_{r_{l_1}}, v_{r_{l_2}}, b_1, b_2$.

The construction of the motif graph G is in practice very efficient. For example, for the datasets we study, it is done under a minute when parallelized across 100 CPU cores. Details are given in Appendix C.

[2] For materials applications that rely in particular on electrophilicity, polarity, and extended aromaticity, longer-range combinations and patterns of motifs are often more influential than any individual one.



(a)

(b)

(c)

Figure 1. Illustration of our random walk representation: (a) (top) molecule M, number 33 (middle) H_M as a connected subgraph of G (bottom) $\hat{H}_M$ as a random walk over H_M; (b) the motif graph G, each node is a motif v that contains both the molecular fragment v_B (black molecule sections) and the contexts for attachment (v_R, red molecule sections), each gray line indicates a possible attachment between nodes. Directed edges of $\hat{H}_M$ use the same color as the dashed border of the corresponding figure of M; (c) (top) demonstration of motif matching criteria eq 1-4 (183 $\leftrightarrow$ 5), another example is in Fig. 10 (bottom) two more examples of H_M.

3.1. The Molecular Design Space as Derivations of a Context-Sensitive Grammar Over Motif Graph

We now define our context-sensitive grammar over G. We use the notations defined in the previous section to enumerate the set of production rules, $\mathcal{P}$, in our grammar. There is one initial rule $p_v \in \mathcal{P}$ for each motif v in G, where the LHS is $\mathcal{X}$, and the RHS is the molecular graph g_v with u_B being the base atoms and $\{(u_{r_l})\}$ being the red atom sets that become "options" for attachment. Then, there is exactly one production rule $p_{u,v,l_1,l_2} \in \mathcal{P}$ for each edge $(u, v, e_{l_1,l_2}) \in G$. This edge was attributed with $(u_{r_{l_1}}, v_{r_{l_2}}, b_1, b_2)$ during the construction of G. The application of the production rule then equates to attaching the fragment of v to the fragment of u, at the attachment options keyed with l_1, l_2. In the language of graph grammars, the context of this production rule is hence the molecular graph $g_u(u_B \cup u_{r_{l_1}})$, with the requirement that the matched atoms for $u_{r_{l_1}}$ are red. Applying this production rule replaces the matched atoms for $u_{r_{l_1}}$ within the LHS by $g_v(N(g_v) \setminus v_{r_{l_2}})$, where the red atom sets $\{v_{r_l} \mid v_{r_l} \cap v_{r_{l_2}} \neq \emptyset\}$ in v are introduced as new options for attachment in the RHS. The random walk characterization arises out of the fact that if the LHS molecule contains the context $g_u(u_B \cup u_{r_{l_1}})$, *any* edge $(u, v, e_{l_1,l_2}) \in E$ can be traversed, possibly including self-loops and parallel edges since G is a directed multidigraph.

3.2. Molecules as Random Walks in the Design Space

Intuitively, our representation of a molecule M captures a derivation in the above-defined context-sensitive grammar. While prior work has modeled such derivations in large and complex tree structures (e.g., with auxiliary nodes for partial derivations) (Guo et al., 2023a;b), we model it compactly in terms of a random walk over the bidirectionally connected subgraph $H_M = (V_M, E_M)$ of G given by the fragmentation of M[3]; see Fig. 1 (a). Observe that G is a strong prior for constraining the design space and sufficient for describing the molecular structure of M, but H_M misses the global distribution of which it is a sample of.

Our learnable component models this distribution and, at the same time, captures the features that characterize a specific molecule in terms of a random walk. More specifically, our final molecule representation is a directed-acyclic multigraph $\hat{H}_M = (V_M, \hat{E}_M, w_M)$ that linearizes H_M into a random walk such that (1) $\hat{E}_M \subseteq E_M$, (2) $\hat{H}_M$ remains connected, and (3) there is an Euler path[4]. (i.e., each edge is used exactly once) $v_0, v_1, \ldots, v_\ell$ over $(V_M, \hat{E}_M)$ with $\hat{E}_M := \cup_i\{(v_i, v_{i+1})\}$; this path can be generated via a pre-order traversal that adds a reversed duplicate of the

[3] Refer to Appendix B.3 for how and why we augment G with duplicates of the same motif.

[4] In the case of monomers, the Euler path needs to be closed as monomers have the property of self-loops.

sub-trajectory when the stack contracts. The last component, w_M is the sequence $w_M := p_0, p_1, \ldots, p_{\ell-1}$ of probabilities given by the random walk; that is, p_i represents the probability with which the edge between v_i and v_{i+1} was traversed in the presence of all nodes visited thus far, as shown in Fig. 2. w_M is parameterized by our learnable grammar, and $\hat{H}_M$ is explainable as a random walk of context-sensitive grammar rule applications.

3.3. Learning Motif-based, Context-sensitive Grammars

3.3.1. Parameter Estimation

For parameter estimation, we formulate the process of random walk as a graph heat diffusion process,

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = L(\Phi, t)x_t, \tag{5}$$

where $x_t \in \mathbb{R}^{|V|}$ represents the probability distribution of sampling motifs and $L(\Phi, t) \in \mathbb{R}^{|V|\times|V|}$ is a time-dependent graph Laplacian parameterized by Φ. Here the initial condition of the diffusion process x_0 is a one-hot vector with the root of $\hat{H}_M$ as one. At every time step, the ground-truth x_t follows the transition state of a random walk. In our implementation, $L(\Phi, t)$ is calculated as

$$L(\Phi, t) = D - \hat{W}(t), \hat{W}(t) = W + h(c_t; \phi) \tag{6}$$

where $D \in \mathbb{R}^{|V|\times|V|}$ is the in-degree matrix of G, $h(\cdot; \phi)$ is a memory-sensitive adjustment layer, and c_t is a set-based memory of all nodes visited thus far. If $p^{(t)}$ is the current state of the random walk, the set-based memory, $c^{(t+1)}$, is updated as follows: $c^{(t+1)} \leftarrow \frac{t}{t+1} \cdot c^{(t)} + \frac{1}{t+1} \cdot p^{(t)}$. This set-based memory mechanism has precedents in graph theory literature. The learnable parameters are $\Phi = (W, \phi)$. Further motivation of the set-based memory mechanism is in Appendix C.3 and the full training algorithm can be found in Appendix D.

3.3.2. Training for a Downstream Task

Property Prediction Our grammar-induced molecular graph representation $\hat{H}_i$ allows for applying an off-the-shelf graph neural network $\mathcal{F}_\Theta$ to solve a given prediction task; in our evaluation, we used GIN (Xu et al., 2019). Given a property value $y^{(i)} \in \mathbb{R}$ with each molecule $M^{(i)}$, we apply a linear head f_θ and application-specific loss function $\mathcal{L}$ (e.g., MSE for regression or cross-entropy for classification).

End to End Training Our grammar-based representation can further be optimized via end-to-end training of Φ. Typically, we first train Φ to convergence under our MC-based objective, then train Θ to convergence under eq 7 on the representations induced by Φ. Finally, we freeze Θ and finetune Φ to convergence. Alternatively, we train Φ and Θ together, end to end, by using the following differentiable objective,

$$\tilde{\mathcal{L}}(D; \Theta, \theta, \Phi) = \mathbb{E}_{\hat{H}_M(\cdot;\Phi)}[\mathcal{L}(f_\theta(\mathcal{F}_\Theta(\hat{H}_M, y)] \tag{7}$$

$$= \frac{1}{|D|}\sum_{i=1}^{|D|} \mathcal{L}(f_\theta(\mathcal{F}_\Theta(\hat{H}_M^{(i)})), y^{(i)}), \tag{8}$$

where we estimate the expectation using the training samples from training dataset D.

3.3.3. Molecular Generation

To generate a molecule M, we apply the learned grammar forward to sample edges to traverse during the random walk, as depicted in Fig. 2. Each sampled edge either attaches a new motif to the current M, or backtracks to a previous motif. Details on the algorithm are given in Appendix F.

4. Results & Analysis

Our experiments quantitatively answer the following questions: 1) How well does our method perform on property prediction for our setting of interest? 2) How well does our representation work for the generation of novel molecules, compared with both SOTA symbolic and deep molecular generative models? We also include three ablations to answer: 3) How important are expert motifs, compared to heuristic-based motifs? 4) How data-efficient and runtime-efficient is our method? 5) How does our method compare with other motif-based predictors? Our qualitative analysis answers the following questions: 6) How interpretable is our learnt grammar to an expert? 7) How can we analyze the model's learnt representations?

4.1. Datasets and Baselines

Table 1. Average size of our hierarchical representation H_M over each dataset, with expert vs heuristic motifs.

Dataset	GC		HOPV		PTC	
Expert	Yes	No	Yes	No	Yes	No
Avg. $\|$H_M$\|$	7.3 ± 2.8	3.8 ± 2.2	5.4 ± 1.9	6.5 ± 2.9	3.6 ± 2.4	2.1 ± 1.4

Group Contribution (GC) (Wang et al., 2018; Park & Paul, 1997; Wu et al., 2021)**.** 114 molecules, characterized in terms of gas separation. Their functional groups contribute significantly to maintaining the structures and properties of 3D scaffold building blocks in polymer self-assembly, which in turn play a significant role in gas separation processes, important in gas and oil industry. We used existing monomers (Wang et al., 2018) and compilations of groups (Park & Paul, 1997; Wu et al., 2021) for inferring the fragmentations.

The Harvard organic photovoltaic dataset (HOPV) (Lopez et al., 2016)**.** 316 molecules, applied to aid in the

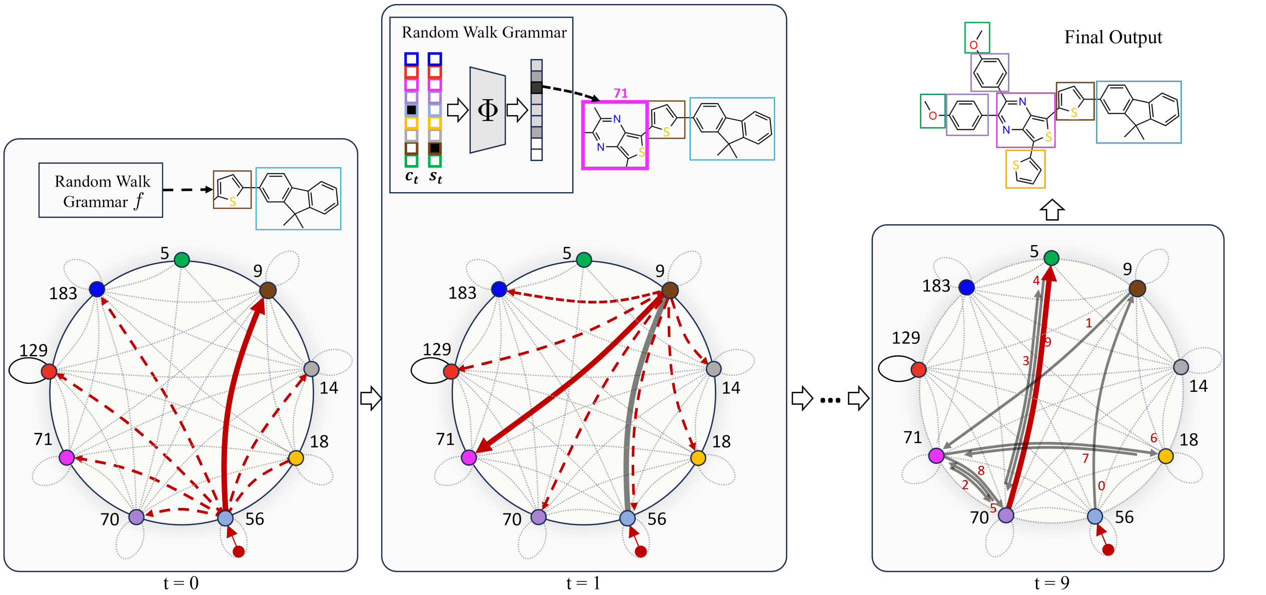


Figure 2. Illustration of our generation procedure: (t=1) our learnable grammar parameterized by Φ samples a state transition $56 \to 9$; (t=2) with the memory of having visited $\{56\}$, our grammar samples a state transition $\to 71$; (t=10) (bottom) our grammar samples a final transition 5, which determines the molecular structure (top); our program's notation is $56 \to 9 \to 71[\to 70 \to 5] \to 70 : 1 \to 5 : 1$

design of organic solar cells, with detailed information pertinent to organic photovoltaic performance metrics. The molecules contain motifs which are among the most significant functional groups for conducting/electroactive materials (Swager, 2017) and photovoltaic properties (Yuan et al., 2022). We extracted motifs important for high HOMO values and enhanced electron delocalization, which are critical for photovoltaic efficiency; see Appendix G for details.

Predictive Toxicology Challenge (PTC) (Helma et al., 2001)**.** 344 small chemical compounds characterized by very distinct functional groups known for their carcinogenic properties or liver toxicity (Miller et al., 1949), with reported values for rats. We specifically segmented it into functional groups that majorly contribute to the improvement of compounds' toxicity (Hughes et al., 2015). Examples from each dataset are shown in Figure 3.

Baselines. To address question 1), we compare with pretrained GNNs (PN (Stanley et al., 2021) and Pre-trained GIN (Hu et al., 2020)), a specialized GNN model for property prediction (wD-MPNN (Aldeghi & Coley, 2022)), two SOTA pretrained models for molecular representation learning (MolCLR (Wang et al., 2022) and UniMol (Zhou et al., 2023)) and two SOTA subgraph-based methods (ESAN (Bevilacqua et al., 2022) and HM-GNN (Shui & Karypis, 2020)). To address question 2), we compare with both Geo-DEG, the SOTA on small dataset property prediction, and its generative variant, DEG, for molecular generation.

4.2. Results

We report the mean absolute error (MAE) and coefficient of determination (R^2) over normalized prediction values for GC and HOPV, and the accuracy and AUC for PTC. For each (dataset, property) pair, we perform an 80-20 train-test split over 3 random seeds and report the mean and standard deviation. For molecular generation, we report commonly used metrics (Polykovskiy et al., 2020; Guo et al., 2023a): Validity/Uniqueness/Novelty: Percentage of chemically valid/unique/novel molecules; Diversity: Average pairwise molecular distance among generated molecules; Retro* Score (RS): Success rate of Retro* (Chen et al., 2020) which was trained to find a retrosynthesis path to build a molecule from a list of commercially available ones. We add the metric of Membership, which tests whether certain motif(s) characteristic of membership to the chemical class are present, primarily as a sanity check. Our method, by design, can achieve 100% if the random walk initializes at the characteristic motif. See A.1 for further discussion.

4.2.1. Property Prediction

To answer question 1), we see in Table 2 that our method, with expert motifs, achieves the best R^2 by a wide margin of 0.10 and 0.06 over the second best method on regression datasets GC and HOPV and the highest accuracy on PTC. With heuristic motifs, our method remains competitive to Geo-DEG, achieving higher R^2 on both regression datasets and accuracy within standard deviation on PTC. In-

Table 2. Results on property prediction (best result **bolded**, second-best underlined). The datasets we include have expert-annotated motifs. We also report Ours (w/o expert) as an ablation without expert motifs.

Datasets \ Methods		wD-MPNN	ESAN	HM-GNN	PN (finetuned)	Pre-trained GIN (finetuned)	MolCLR	Unimol	Geo-DEG	**Ours**	**Ours** (w/o expert)
Group	**MAE** ↓	0.47 ± 0.09	0.51 ± 0.06	0.34 ± 0.12	0.76 ± 0.30	0.68 ± 0.05	$\underline{0.26 \pm 0.10}$	0.38 ± 0.13	$\underline{0.26 \pm 0.11}$	$\mathbf{0.25 \pm 0.09}$	0.27 ± 0.08
	$\mathbf{R^2}$ ↑	0.41 ± 0.12	-0.39 ± 0.62	0.56 ± 0.20	-7.56 ± -7.71	0.19 ± 0.09	0.68 ± 0.20	0.47 ± 0.25	0.70 ± 0.20	$\mathbf{0.80 \pm 0.15}$	$\underline{0.74 \pm 0.15}$
HOPV	**MAE** ↓	0.36 ± 0.03	0.37 ± 0.02	0.40 ± 0.02	0.42 ± 0.02	0.38 ± 0.02	0.34 ± 0.03	0.31 ± 0.03	$\underline{0.30 \pm 0.02}$	$\underline{0.30 \pm 0.05}$	$\mathbf{0.22 \pm 0.15}$
	$\mathbf{R^2}$ ↑	0.69 ± 0.04	0.66 ± 0.06	0.65 ± 0.05	0.65 ± 0.04	0.66 ± 0.03	0.68 ± 0.03	0.70 ± 0.02	$\underline{0.74 \pm 0.03}$	$\mathbf{0.80 \pm 0.06}$	$\underline{0.77 \pm 0.12}$
PTC	**Acc** ↑	0.67 ± 0.06	0.64 ± 0.08	0.66 ± 0.07	0.61 ± 0.08	0.62 ± 0.09	0.60 ± 0.03	0.57 ± 0.05	$\underline{0.69 \pm 0.07}$	$\mathbf{0.70 \pm 0.01}$	0.67 ± 0.02
	AUC ↑	$\underline{0.70 \pm 0.05}$	0.68 ± 0.06	0.69 ± 0.06	0.65 ± 0.07	0.66 ± 0.07	0.66 ± 0.05	0.67 ± 0.06	$\mathbf{0.71 \pm 0.07}$	0.69 ± 0.03	0.66 ± 0.05

terestingly, using heuristic-based motifs in HOPV, achieves significantly (27%) lower MAE than expert-based motifs and Geo-DEG. To answer question 3), we see that the ablation suggests expert motifs are generally better, but may be more sensitive to outliers than heuristic-based motifs. We observe experts are generally better at identifying special cases that heuristics are unaware of, but heuristics are more consistent. This reflects how R^2 is generally more sensitive to outliers than MAE. We describe our experts' annotation criteria in Appendix A and do an in-depth case study on HOPV in Appendix G.

4.2.2. Molecular generation

To answer question 2), we see in Table 3 that our method produces comparably more diverse molecules than the training dataset (+0.03 on HOPV, -0.01 on PTC) and significantly more synthesizable molecules than the previous SOTA, DEG (+39% on HOPV, +22% on PTC). On HOPV, our retrosynthetic planner finds synthesis paths at a 14% *higher* rate on our novel molecules than the *original* dataset, a careful collation of experimental photovoltaic designs (Lopez et al., 2016). This is encouraging to experimentalists whose work is contingent on the designs' feasibility for synthesis. We also compare our methods with established VAE-based molecular generative models such as (Jin et al., 2018) and its follow-up work (Jin et al., 2020) which includes larger structural motifs. Furthermore, we modified the implementation of Hier-VAE to incorporate our epert motifs. For all three cases, we follow the default settings, train until convergence, and use the checkpoint with the lowest loss to sample 1000 molecules. We observe that both VAE-based methods struggle to generate sufficiently unique molecules, with only 11%-43% (HOPV) and 8%-28% (PTC) of the 1000 generated molecules being unique. This is despite sampling from a Gaussian noise distribution. Meanwhile, our model generates 100% unique and novel molecules, while ensuring a high internal diversity second only to DEG. For reference, (Jin et al., 2018; 2020) trained and evaluated their methods on 250K molecules extracted from ZINC (Sterling & Irwin, 2015) and a polymer dataset containing 86K polymers. Meanwhile, our datasets contain only 100-300 molecules and, in the case of HOPV, feature much larger molecules. Rather than using an encoder-decoder setup which requires significantly more data to learn the mapping to and from a latent space, our generative model explicitly captures the transition probabilities over traversing the symbolic space of structural motifs. Our grammar derivation can easily be conditioned by a set-based memory to apply a diverse set of transition rules. This leads to more unique, diverse, and, most importantly, synthesizable structures.

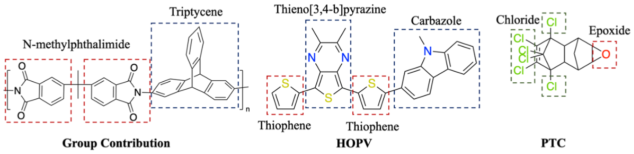


Figure 3. Example molecules from GC, HOPV, and PTC. These datasets are characterized by modular substructures that correspond to meaningful chemical functional groups.

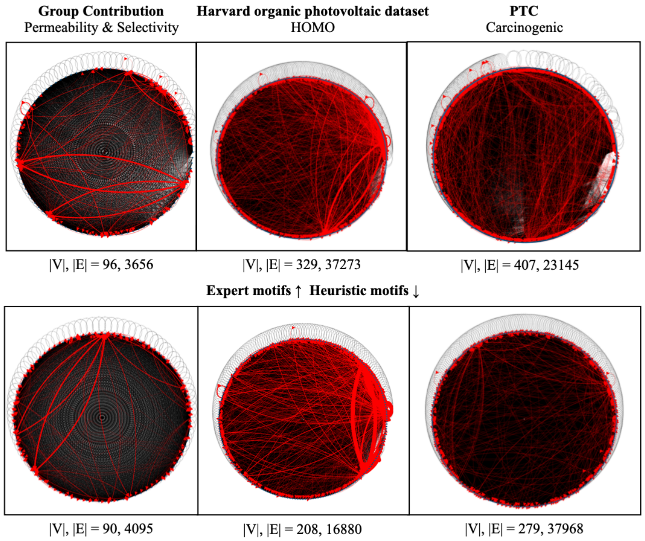


Figure 4. Visualization of our motif graph G; black edges indicate matched motif pairs, thickness of red edges correspond to the numbers of H_M that traverse it.

4.3. Ablations

4.3.1. Ablation: vary training dataset size

To answer question 4), we conduct an ablation study in Figure 5 over the training split size to study how data and

Table 3. Results on molecular generation for HOPV (top) and PTC (bottom); for both datasets, we generate 1000 novel molecules. Refer to Appendix A.1 for more details on Membership.

Datasets	Methods	Valid	Unique	Novel	Diversity	RS	Memb.
HOPV	Train Data	100%	100%	N/A	0.86	51%	100%
	DEG	100%	98%	99%	0.93	19%	46%
	JT-VAE	100%	11%	100%	0.77	99%	84%
	Hier-VAE	100%	43%	96%	0.87	79%	76%
	Hier-VAE (+expert)	100%	29%	92%	0.86	84%	82%
	Ours	100%	100%	100%	0.89	58%	71%
PTC	Train Data	100%	100%	N/A	0.94	87%	30%
	DEG	100%	88%	87%	0.95	38%	27%
	JT-VAE	100%	8%	80%	0.83	96%	27%
	Hier-VAE	100%	20%	85%	0.91	92%	25%
	Hier-VAE (+expert)	100%	28%	75%	0.93	90%	17%
	Ours	100%	100%	100%	0.93	60%	22%

runtime-efficient our method is in comparison with Geo-DEG. Our method performs strictly better on MAE as the training set is reduced from 70% to 10%. This is in addition to the method running an order of magnitude faster, highlighting gains in both data efficiency and runtime efficiency.

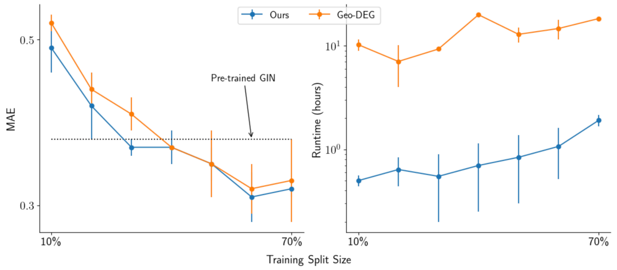


Figure 5. Varying the training dataset size from 10-70%.

4.3.2. Comparison with motif-based baselines

To answer question 5), we compare with two baselines. The first, Bag-of-Motifs, ablates our hierarchical information and retains only the motif co-occurrence information. For each molecule, we obtain a feature vector that concatenates a) the occurrence counts of all motifs and b) the Morgan fingerprint of the molecule. We train an XGBoost regressor/classifier on top of these features (details in Appendix E). As shown in Table 4, this baseline has enough capacity to overfit the training data but fails to generalize. This allows us to conclude in the absence of a proper representation, motif occurrence information is not sufficient for generalization. Interestingly, expert-level motifs are not superior to heuristic-based motifs in this featurization. This suggests that the quality of motifs are not relevant in the absence of a hierarchical representation that incorporates the fine-grained features of each individual motif. The second, HM-GNN (Shui & Karypis, 2020), is a SOTA motif-based property predictor that explicitly models motif-molecule and motif-motif relationships using a hetereogenous graph. Furthermore, we endowed the method with our expert motifs since the vanilla version only considers bonds and rings. On both regression datasets, HM-GNN avoids overfitting but does not catch up to our method's generalization capability. Endowing HM-GNN with our expert motifs enables better fitting of the training data but further hinders generalization. On PTC, HM-GNN is competitive with our method in accuracy but shows a discrepancy in terms of AUC. This is concerning as a lower AUC may imply higher sensitivity to class imbalance (in PTC, there are 45% more negatives than positives) and classification thresholds. Meanwhile, our method can both 1) completely fit the training data (> 0.99 R^2, $> 99\%$ Acc/AUC), and 2) generalize to the test data, with further regularization potentially leading to even better results. We believe our motif-based representation carries better inductive biases, integrates better with expert motifs, and demonstrates stronger empirical performance.

4.4. Analysis

4.4.1. Rule extraction from grammar

To answer question 6), we extract context-sensitive grammar rules from our trained model. We perform best-first search over random walk trajectories, beginning with base trajectories corresponding to each group $v \in G$. We only expand trajectories with transition probabilities above a minimum threshold. Each trajectory that reaches a transition with probability of 1 is extracted as a "hard" context-sensitive rule. We depict two such rules in Figure 6, with a more exhaustive compilation in Appendix F.1.

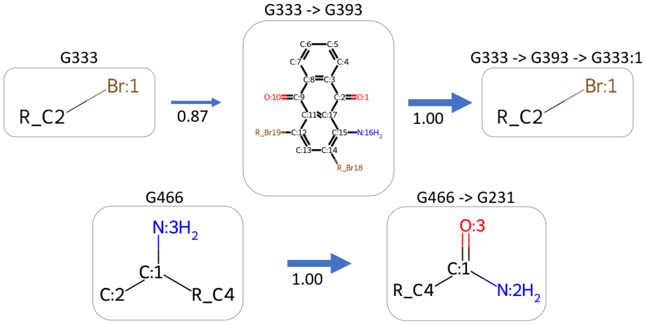


Figure 6. We visualize two hard context-sensitive rules on PTC that correspond to design principles of the addition of halogen groups to further improve molecular toxicity.

Figure 6 shows how our model recovers a set of design principles used to facilitate the synthesis of functional molecules and grounded in the structure-property relationship of PTC. Consider the transformation of the triple benzene derivative molecule (labeled as ['G333', 'G393']) with the addition of bromide moiety (labeled as ['G333', 'G393', 'G333:1']). In this instance, the central moiety, G393, is characterized by two symmetrical ketone groups and two bromides adjoined to the aromatic ring. This configuration markedly enhances the molecule's toxicity. Moreover, by strategically positioning additional binding sites on the aromatic ring, the software augments the molecule with two extra bromide groups, G333, thereby exacerbating its hepatotoxicity. In

Table 4. Ablation study on overfitting and generalization, vs other motif-based baselines, with and w/o expert motifs. Best result is **bolded**.

Ablation/Dataset	HOPV				PTC				Group Contribution			
	Train MAE ↓	Train R^2 ↑	Test MAE ↓	Test R^2 ↑	Train Acc ↑	Train AUC ↑	Test Acc ↑	Test AUC ↑	Train MAE ↓	Train R^2 ↑	Test MAE ↓	Test R^2 ↑
Bag-of-Motifs	0.014± 0.002	0.997± 0.001	0.486± 0.025	0.489± 0.062	0.996± 0.000	**1.000± 0.000**	0.529± 0.031	0.609± 0.031	**0.000± 0.000**	**1.000± 0.000**	0.481± 0.174	0.257± 0.453
Bag-of-Motifs (+expert)	**0.011± 0.004**	**1.000± 0.000**	0.521± 0.031	0.446± 0.125	0.996± 0.000	**1.000± 0.000**	0.581± 0.018	0.612± 0.029	**0.000± 0.000**	**1.000± 0.000**	0.493± 0.143	0.214± 0.404
HM-GNN	0.366± 0.035	0.686± 0.066	0.473± 0.019	0.441± 0.065	0.915± 0.033	0.966± 0.016	**0.710± 0.023**	0.678± 0.040	0.281± 0.064	0.717± 0.137	0.362± 0.113	0.592± 0.202
HM-GNN (+expert)	0.201± 0.009	0.895± 0.019	0.451± 0.025	0.408± 0.095	**0.999± 0.002**	**1.000± 0.000**	0.681± 0.024	0.587± 0.075	0.185± 0.016	0.926± 0.039	0.345± 0.149	0.547± 0.295
Ours (-expert)	0.075± 0.003	0.990± 0.001	**0.288± 0.048**	0.765± 0.146	0.994± 0.001	0.999± 0.000	0.671± 0.020	0.659± 0.047	0.044± 0.015	0.995± 0.004	0.268± 0.084	0.738± 0.148
Ours	0.045± 0.003	0.996± 0.001	0.295± 0.049	**0.796± 0.105**	0.996± 0.000	**1.000± 0.000**	0.705± 0.007	**0.711± 0.018**	0.028± 0.007	0.998± 0.002	**0.222± 0.079**	**0.819± 0.137**

another example, the molecule with an ammonia group (labeled as [‘G466’]) transforms with an additional ketone group (labeled as [‘G466’, ‘G231’]). Here, the presence of a C=O double bond within an acetamide group is a key contributor to hepatotoxicity.

4.4.2. Two-dimensional T-SNE on H_M versus pretrained representations

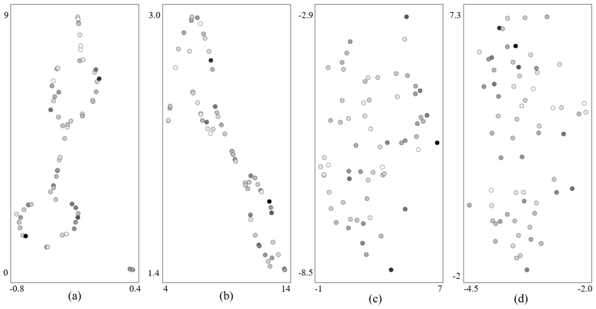


Figure 7. Final layer representations from: a) Our method b) Our method (-expert) c) Pre-trained GIN d) HM-GNN. We apply a grayscale coloring map using the normalized value of the desired property (the darker the dot, the higher the HOMO).

To answer question 7), we analyze the 2D t-SNE embeddings of various methods’ final layer representations of 64 test set molecules on HOPV. As shown in Figure 7, our method is unique in extracting visually meaningful representations. High HOMO molecules were identified from the visual clusters for structural analysis. Molecules in the upper cluster as illustrated in Figure 8 often have structures promoting electron delocalization, like carbonohydrazonoyl dicyanide, while those in the lower cluster have electron-donating groups or structures increasing steric hindrance to boost HOMO values as shown in Figure 8. These two structural features correspond to the two primary ways to design molecules with high HOMO values. These findings aid the search for novel molecules with desirable photovoltaic properties. As 2D t-SNE is not a universal way to analyze representations, we also visualize the agreement between embedding similarity and structural similarity using a 64 × 64 grid. This is can be found in Appendix G.3, as part of an in-depth case study on HOPV.

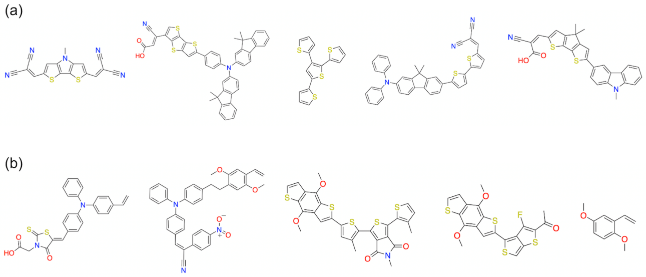


Figure 8. Examples of top HOMO value compounds with group (a) from the top cluster and group (b) from the bottom cluster.

4.5. Conclusion & Future Work

We represent molecules as random walks over an interpretable context-sensitive grammar over the motif graph, a hierarchical abstraction over the design space. We devise and execute a practical workflow that invites experts in the loop to enhance our design basis and representations by fragmenting molecules into well-established functional groups, creating a synergy between expert feedback and the quality of our representations. Our evaluation on downstream property prediction and molecular generation tasks shows our representation combines quantitative advantages in performance and efficiency with qualitative advantages of simplicity and enhanced interpretability. One promising avenue of future research is improving the autonomous extraction of interpretable grammar rules through learnable and/or human-guided approaches with Large Language Models.

Acknowledgements

This work is supported by the MIT-IBM Watson AI Lab and its member companies Evonik and Shell.

Impact Statement

This paper presents work whose goal is to concurrently and conjointly advance the fields of Machine Learning and Chemical Discovery. The application of our method can have consequences for real-world discovery workflows. There are no ethical aspects which we foresee and feel must be discussed here.

References

Aldeghi, M. and Coley, C. W. A graph representation of molecular ensembles for polymer property prediction. *Chemical Science*, 13(35):10486–10498, 2022.

Bevilacqua, B., Frasca, F., Lim, D., Srinivasan, B., Cai, C., Balamurugan, G., Bronstein, M., and Maron, H. Equivariant subgraph aggregation networks. *ICLR*, 2022.

Blunk, D., Bierganns, P., Bongartz, N., Tessendorf, R., and Stubenrauch, C. New speciality surfactants with natural structural motifs. *New J. Chem.*, 30:1705–1717, 2006. doi: 10.1039/B610045G.

Bronstein, H., Nielsen, C. B., Schroeder, B. C., and McCulloch, I. The role of chemical design in the performance of organic semiconductors. *Nature Reviews Chemistry*, 4(2):66–77, jan 2020. ISSN 2397-3358. doi: 10.1038/s41570-019-0152-9.

Cai, C., Wang, D., and Wang, Y. Graph coarsening with neural networks. *ICLR*, 2021.

ChemAxon. Fragmenter. URL `http://www.chemaxon.com/`.

Chen, B., Li, C., Dai, H., and Song, L. Retro*: Learning retrosynthetic planning with neural guided a* search. 2020.

Chen, J., Saad, Y., and Zhang, Z. Graph coarsening: from scientific computing to machine learning. *SeMA Journal*, 79(1):187–223, 2022.

Chen, Y., Yao, R., Yang, Y., and Chen, J. A gromov–wasserstein geometric view of spectrum-preserving graph coarsening. *ICML*, 2023.

Degen, J., Wegscheid-Gerlach, C., Zaliani, A., and Rarey, M. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 2008.

Fang, G., Samorodnitsky, G., and Xu, Z. A cover time study of a non-markovian algorithm. *arXiv preprint arXiv:2306.04902*, 2023.

Frasca, F., Bevilacqua, B., Bronstein, M. M., and Maron, H. Understanding and extending subgraph gnns by rethinking their symmetries. *NeurIPS*, 2022.

Gasieniec, L. and Radzik, T. Memory efficient anonymous graph exploration. In *Graph-Theoretic Concepts in Computer Science: 34th International Workshop, WG 2008, Durham, UK, June 30–July 2, 2008. Revised Papers 34*, pp. 14–29. Springer, 2008.

Guo, M., Shou, W., Makatura, L., Erps, T., Foshey, M., and Matusik, W. Polygrammar: Grammar for digital polymer representation and generation. *Advanced Science*, 9(23):2101864, 2022. doi: https://doi.org/10.1002/advs.202101864. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.202101864`.

Guo, M., Thost, V., Song, S., Balachandran, A., Das, P., Chen, J., and Matusik, W. Grammar-induced geometry for data-efficient molecular property prediction. 2023a.

Guo, M., Thost, V., Song, S. W., Balachandran, A., Das, P., Chen, J., and Matusik, W. Hierarchical grammar-induced gemoetry for data-efficient molecular property prediction. *ICML*, 2023b.

Helma, C., King, R. D., Kramer, S., and Srinivasan, A. The Predictive Toxicology Challenge 2000–2001 . *Bioinformatics*, 17(1):107–108, 2001.

Hu, W., Liu, B., Gomes, J., Marinka Zitnik, P. L., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. *ICLR*, 2020.

Hughes, T. B., Miller, G. P., and Swamidass, S. J. Modeling epoxidation of drug-like molecules with a deep machine learning network. *ACS Central Science*, 1(4):168–180, 2015.

IUPAC. *Compendium of Chemical Terminology*. 1997.

Jiang, J., Zhang, R., Zhao, Z., Ma, J., Liu, Y., Yuan, Y., and Niu, B. Multigran-smiles: multi-granularity smiles learning for molecular property prediction. *Bioinformatics*, 38 (19):4573–4580, 2022.

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

Jin, W., Barzilay, R., and Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. *ICML*, 2020.

Kajino, H. Molecular hypergraph grammar with its application to molecular optimization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3183–3191. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/kajino19a.html`.

Karande, P., Gallagher, B., and Han, T. Y.-J. A strategic approach to machine learning for material science: How to tackle real-world challenges and avoid pitfalls. *Chemistry of Materials*, 2022.

Krenn, M., Häse, F., Nigam, A., Friederich, P., and Aspuru-Guzik, A. Self-referencing embedded strings (selfies): A 100 *Machine Learning: Science and Technology*, 1(4):045024, oct 2020. doi: 10.1088/2632-2153/aba947. URL `https://dx.doi.org/10.1088/2632-2153/aba947`.

Landrum, G. Rdkit: Open-source cheminformatics software. 2016.

Lee, W. J., Kwak, H. S., Lee, D.-r., Oh, C., Yum, E. K., An, Y., Halls, M. D., and Lee, C.-W. Design and synthesis of novel oxime ester photoinitiators augmented by automated machine learning. *Chemistry of Materials*, 34(1):116–127, jan 2022. ISSN 0897-4756. doi: 10.1021/acs.chemmater.1c02871.

Leung, L., Kalgutkar, A. S., and Obach, R. S. Metabolic activation in drug-induced liver injury. *Drug metabolism reviews*, 44(1):18–33, 2012.

Li, J., Wang, J., Zhao, Y., Zhou, P., Carter, J., Li, Z., Waigh, T. A., Lu, J. R., and Xu, H. Surfactant-like peptides: From molecular design to controllable self-assembly with applications. *Coordination Chemistry Reviews*, 421:213418, 2020. ISSN 0010-8545. doi: https://doi.org/10.1016/j.ccr.2020.213418.

Li, Y., Vinyals, O., Dyer, C., Pascanu, R., and Battaglia, P. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

Lopez, S. A., Pyzer-Knapp, E. O., Simm, G. N., Lutzow, T., Li, K., Seress, L. R., Hachmann, J., and Aspuru-Guzik, A. The harvard organic photovoltaic dataset. *Sci Data*, 3, 2016.

Ma and Chen. Unsupervised learning of graph hierarchical abstractions with differentiable coarsening and optimal transport. *AAAI*, 2021.

Masuda, N., Porter, M. A., and Lambiotte, R. Random walks and diffusion on networks. *Physics reports*, 716: 1–58, 2017.

Miller, J. A., Sapp, R. W., and Miller, E. C. The Carcinogenic Activities of Certain Halogen Derivatives of 4-Dimethylaminoazobenzene in the Rat*. *Cancer Research*, 9(11):652–660, 1949.

Nigam, A., Pollice, R., Krenn, M., Gomes, G. D. P., and Aspuru-Guzik, A. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chem. Sci.*, 12(20):7079–7090, April 2021.

Park, J. and Paul, D. R. Correlation and prediction of gas permeability in glassy polymer membrane materials via a modified free volume based group contribution method. *Journal of Membrane Science*, 125(1):23–39, 1997.

Pemantle, R. A. *Random processes with reinforcement*. PhD thesis, Massachusetts Institute of Technology, Dept. of Mathematics, 1988.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Johansson, S., Chen, H., Nikolenko, S., Aspuru-Guzik, A., and Zhavoronkov, A. Molecular sets (moses): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 2020.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 2010.

Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D., and Lambiotte, R. Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications*, 5(1):4630, 2014.

Sawlani, N. Drug discovery informatics market set to surge at 10.9*Transparency Market Research, Inc*, 2024.

Schimunek, J., Seidl, P., Friedrich, L., Kuhn, D., Rippmann, F., Hochreiter, S., and Klambauer, G. Context-enriched molecule representations improve few-shot drug discovery. 2023. URL `https://openreview.net/pdf?id=XrMWUuEevr`.

Shui, Z. and Karypis, G. Heterogeneous molecular graph neural networks for predicting molecule properties. *ICDM*, 2020.

Stanley, M., Bronskill, J. F., Krzysztof Maziarz, H. M., Lanini, J., Segler, M., Schneider, N., and Brockschmidt, M. Fs-mol: A few-shot learning dataset of molecules. *NeurIPS*, 2021.

Sterling, T. and Irwin, J. J. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Swager, T. M. 50th anniversary perspective: Conducting/semiconducting conjugated polymers. a personal perspective on the past and the future. *Macromolecules*, 50 (13):4867–4886, 2017.

Thanou, D., Dong, X., Kressner, D., and Frossard, P. Learning heat diffusion graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):484–499, 2017.

Türel, T., Dağlar, Ö., Eisenreich, F., and Tomović, Ž. Epoxy thermosets designed for chemical recycling. *Chemistry – An Asian Journal*, 18(15), aug 2023. ISSN 1861-4728. doi: 10.1002/asia.202300373.

Wang, S. and Wu, X. The mechanical performance prediction of steel materials based on random forest. *Frontiers in Computing and Intelligent Systems*, 2023.

Wang, S., Shi, K., Tripathi, A., Chakraborty, U., Parsons, G. N., and Khan, S. A. Designing intrinsically microporous polymer (pim-1) microfibers with tunable morphology and porosity via controlling solvent/nonsolvent/polymer interactions. *ACS Applied Polymer Materials*, 2(6):2434–2443, 2020.

Wang, Y., Ma, X., Ghanem, B., Alghunaimi, F., Pinnau, I., and Han, Y. Polymers of intrinsic microporosity for energy-intensive membrane-based gas separations. *Materials Today Nano*, 3:69–95, 2018.

Wang, Y., Wang, J., Cao, Z., and Farimani, A. B. Molecular contrastive learning of representations via graph neural networks. *nature machine intelligence*, 2022.

Wu, A. X., Lin, S., Rodriguez, K. M., Benedetti, F. M., Joo, T., Grosz, A. F., Storme, K. R., Roy, N., Syar, D., and Smith, Z. P. Revisiting group contribution theory for estimating fractional free volume of microporous polymer membranes. *Journal of Membrane Science*, 636, 2021.

Xia, J., Zhang, L., Liu, Y., Gao, Z., Hu, B., Tan, C., Zheng, J., Li, S., and Li, S. Z. Understanding the limitations of deep models for molecular property prediction: Insights and solutions. *NeurIPS*, 2023a.

Xia, J., Zhu, Y., Du, Y., Liu, Y., and Li, S. A systematic survey of chemical pre-trained models. IJCAI, 2023b.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *ICLR*, 2019.

Yang, N., Zeng, K., Qitian Wu, X. J., and Yan, J. Learning substructure invariance for out-of-distribution molecular representations. *NeurIPS*, 2022.

You, J., Liu, B., Ying, Z., Pande, V., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

Yuan, W., Vijayamohanan, H., Luo, S.-X. L., Husted, K., Johnson, J. A., and Swager, T. M. Dynamic polypyrrole core–shell chemomechanical actuators. *Chemistry of Materials*, 34(7):3013–3019, 2022.

Zhou, G., Gao, Z., Ding, Q., Zheng, H., Xu, H., Wei, Z., Zhang, L., and Ke, G. Uni-mol: A universal 3d molecular representation learning framework. *ICLR*, 2023.

A. Motif Collection Strategy

Motifs are used to construct our motif graph G, which forms the design basis for both our generative and predictive methods. The complexity of our grammar as conveyed by the size of the motif graph G for different motif collection strategies we tried are summarized in Table 5. For the remainder of this section, we describe the Expert Annotation strategy which is our primary workflow. The strategies for obtaining motifs from literature and heuristic-based fragmentation are described in B.1 and F.1, respectively.

Grammar Complexity $(\|V\|, \|E\|)$	HOPV	PTC	GC
Literature	N/A	N/A	$(96, 3656)$
Expert	$(329, 37273)$	$(407, 23145)$	N/A
Heuristic	$(208, 16880)$	$(279, 37968)$	$(90, 4095)$

Table 5. Number of nodes and edges of motif graph G constructed using different annotation strategies

A.1. Expert Annotation Workflow

The workflow consists of two steps: molecule segmentation, and extracting the negative groups for pairwise attachments. Step 1 involves cooperation from an expert, and we detail our polished workflow to facilitate that process, which we have attempted with multiple experts. Step 2 can become automated after the expert identifies 1) governing rules for a particular dataset, and 2) important exceptions to the rule. On average, each dataset takes less than one working day for one expert to fully annotate and process. The annotated datasets for Group Contribution, HOPV, and PTC will be released upon publication.

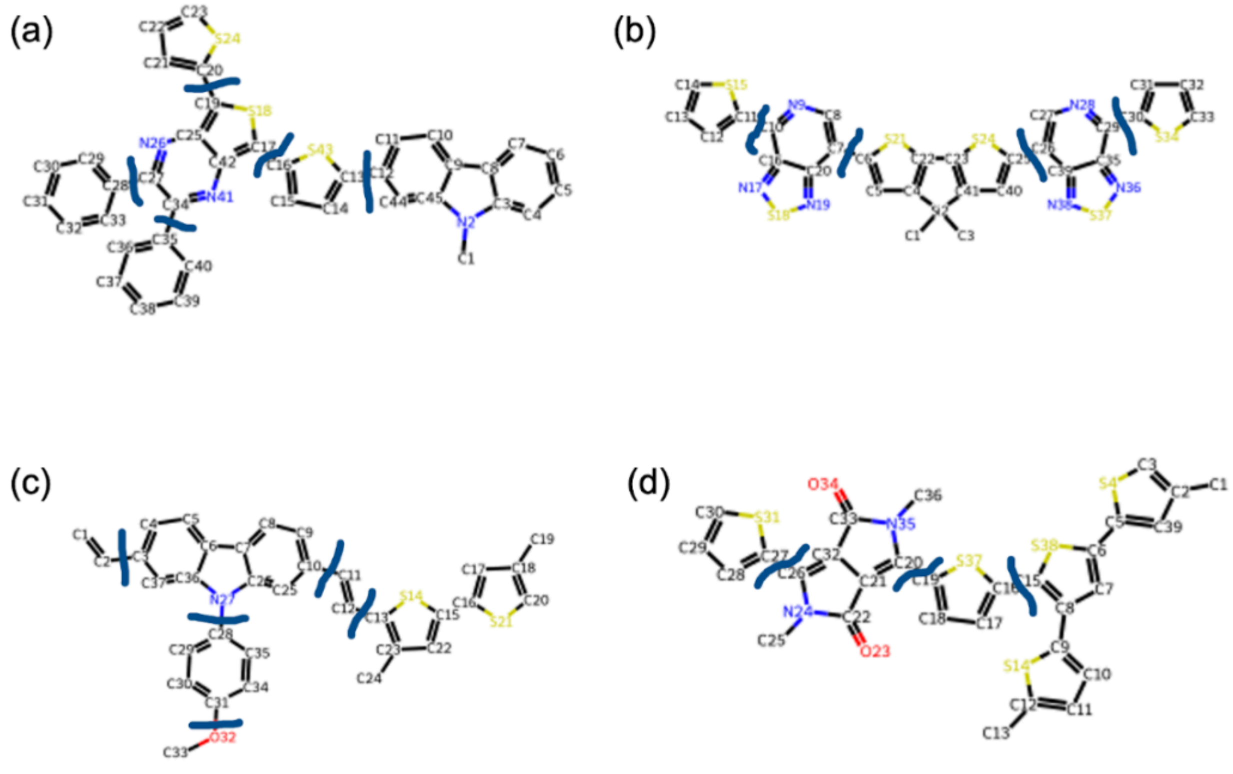


Figure 9. Example segmentations for four molecules in the HOPV dataset. Segmentation locations are marked by the dark blue (teal) line.

STEP 1: EXPERT SEGMENTATION

First, experts view figures of the molecules, and indicate the bonds to break in order to segment the molecule into coherently chosen sub-fragments, shown in Figure 9. We provide a brief description of the example datasets we show here and elaborate on the rationale behind the experts' segmentation strategy:

Table 6. Segmentation of the molecules (a) to (d) in 9. *Bonds to break* indicates the chemical bonds to cut to create black fragments, while the *black groups* and *red groups* listed for each molecules correspond to one another, respectively.

Structures	Bonds to Break	Black Groups	Red Groups
(a)	(12,13) (16,17) (19,20) (27,28) (34,35)	(1,2,3,4,5,6,7,8,9,10,11,12,44,45) (13,14,15,16,43) (17,18,19,25,26,27,34,41,42) (20,21,22,23,24) (28,29,30,31,32,33) (35,36,37,38,39,40)	(13) (12,17) (16,20,28,35) (19) (27) (34)
(b)	(10,11) (6,7) (25,26) (29,30)	(11,12,13,14,15) (7,8,9,10,16,17,18,19,20) (1,2,3,4,5,6,21,22,23,24,25,40,41) (26,27,28,29,35,36,37,38,39) (30,31,32,33,34)	(10) (11,6) (7,26) (25,30) (29)
(c)	(2,3) (11,10) (12,13) (27,28) (31,32)	(1,2) (3,4,5,6,7,8,9,10,24,25,27,36,37) (28,29,30,31,34,35) (11,12) (32,33) (13,14,15,16,17,18,19,20,21,22,23,24)	(3) (2,11) (27,32) (10,13) (31) (12)
(d)	(15,16) (19,20) (26,27)	(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,38,39) (16,17,18,19,37) (20,21,22,23,24,25,26,32,33,34,35,36) (27,28,29,30,31)	(16) (15,20) (19,27) (26)

Predictive Toxicology Challenge (PTC) (Helma et al., 2001) The small molecules are characterized by distinct functional groups known for their carcinogenic properties or liver toxicity (Miller et al., 1949; Helma et al., 2001). These groups comprise a rich variety of elements such as halides, alkylating agents, epoxides, and furan rings. (Figure 3) Therefore, we specifically segmented it into functional groups and sub-structures that contribute most to the toxicity of the compounds (Hughes et al., 2015).

The Harvard organic photovoltaic dataset (HOPV) (Lopez et al., 2016) The process of segmenting the Harvard Organic Photovoltaic Dataset (HOPV15) demonstrates a methodical and efficient approach to categorizing photovoltaic data. This dataset contains a comprehensive collection of experimental photovoltaic data from literature coupled with quantum-chemical calculations across various conformers. The criteria for the extraction of the black group are clearly defined and systematically applied. Functional groups like vinyl, alcohol, ketone, aldehyde, amine, ester, and amide are separated as individual black fragments. Similarly, distinct black fragments are used for individual rings including benzene, pyrrole, and thiophene, in acknowledgement of their Pi-orbital electron delocalization. Complex structures with multiple consecutive rings, known for their distinctive HOMO-LUMO bandgaps and electrochemical properties, such as thieno[3,4-b]pyrazine, carbazole, and 2,5-dimethyl-3,6-dioxo-2,3,5,6-tetrahydropyrrolo[3,4-c]pyrrole, are also segmented as individual entities. Moreover, for groups of 2-3 consecutive symmetrical thiophene or pyrrole units, the methodology captures the significance of maintaining them as a complete black group because these consecutive groups sustain the electron cloud delocalization between repeating units, strongly influencing optical and electronic properties not limited to light absorption, charge transport, and luminescent properties in photovoltaic applications. Meanwhile, this method of segmentation enhances utility and understanding of the results by clearly basing predictions on existing important structures.

Defining Membership. The Membership metric is reported in 4.2.2 after further consultation with experts, who identify the presence of Thiophene as a proxy for Membership to HOPV, and the presence of Chloride/Bromide Halides (a key indicator of toxicity) for PTC. In the case of both datasets, the Membership metric is only a sanity check that the method can produce a non-trivial number of characteristic compounds. Here's our justification for the criteria on each dataset:

1. A chloroalkane (Cl-C) is the most common motif in the PTC dataset. Yet, it is still not present in a majority of structures, making the broader class of alkyl halides (Cl-C, Br-C-C) the best choice for a membership criterion. Their prevalence is attributed to their reactivity and ability to undergo metabolic activation (Leung et al., 2012), leading to

the formation of highly reactive intermediates that can interact with DNA and other cellular components, potentially initiating carcinogenic processes. Although not all carcinogenic compounds will necessarily contain this class of motifs, their presence contributes a strong likelihood.

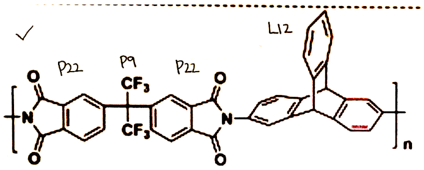
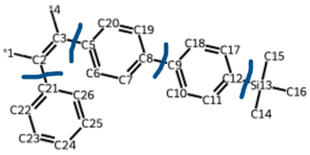
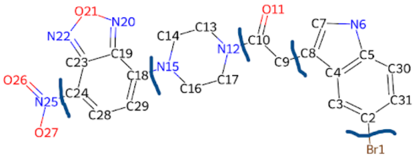
2. Thiophene, a 5-member ring with one sulfur group, is the most common motif in the HOPV dataset, making it the best choice for a single-motif membership criterion for HOPV. More broadly, thiophene and its derivatives are arguably the most common chemical substructure in photovoltaics due to their ability to donate electrons, resulting in particularly high highest occupied molecular orbital (HOMO) levels, along with stability, tunability of energy levels, and compatibility with film forming techniques. While not every suitable organic photovoltaic compound will contain it, the vast majority will.

In both cases, our method can easily achieve 100% membership with a slight modification to the sampling procedure: instead of iterating through every possible starting motif node, we always initialize our random walk at the membership motif. We choose not to modify our sampling procedure, and instead include this metric in Table 3 for completeness, since it is still a good sanity check for other methods to show they generate a non-trivial fraction of candidates with those motif(s).

Step 2: Extracting Red Groups

Key to the definition of our motif graph is the specification of red groups ($v_R \subset v$) that define the possible pairwise attachments between motifs. There are no hard rules, but generally red groups should be minimally necessary. It should be: 1) consistent, for enabling more attachments, hence making the motif graph denser; 2) small, for enabling fast isomorphism checking during the precomputation of the motif graph, and 3) necessary, ensuring only valid attachments. Failure to follow 3) can generate chemically disallowed structures.

Table 7. Context determination rules and examples on datasets Group Contribution, HOPV and PTC.

Dataset	Rule	Example
Group Contribution	We directly use the released groups in (Park & Paul, 1997; Wu et al., 2021).	 ✓ P22 P9 P22 L12 O CF₃ O N CF₃ N O O n
HOPV	- For groups of a single atom – pick ring of neighbor fragment if possible - For groups of multiple atoms – pick only the connected atoms in the neighbor fragment	 *4 C3 C5 C20 C19 *1 C2 C6 C7 C8 C9 C18 C17 C15 C12 Si13 C16 C10 C11 C14 C21 C26 C22 C25 C23 C24
PTC	- Same as HOPV	 O21 N20 N22 C19 C23 C18 O26 N25 C24 O27 C28 C29 C14 C13 N12 N15 C16 C17 O11 C10 C9 C7 N6 C8 C5 C4 C30 C3 C31 C2 Br1

B. Representing Existing Molecules as Walks on This Graph

B.1. Extracting Walks from Segmentation (HOPV, PTC)

During segmentation, we use the workflow in Appendix A.1 to segment a molecule into fragments. In doing so, we also obtain the molecule's representation H_M as a directed subgraph over the motif graph. The pseudocode is found in Algorithm 1.

Algorithm 2 linearizes the molecule into a directed acyclic graph (DAG). This procedure begins by finding the longest path,

```
Algorithm 1: function extract_walk(D,B)
Input: $D = [M_i \mid i = 1, \ldots, |D|]$ // dataset of molecules
$B = [B_i \mid i = 1, \ldots, |D|]$; // annotation, i.e.  bonds to break, for each molecule
$D_G = []; V = \{\}; H = []$;
for $i$ in range (len($D$)) do
    $F_i \leftarrow$ break_bonds($M_i, B_i$); // break bonds and form fragments
    $G_i \leftarrow$ form_graph($F_i, B_i$); // graph of motifs, edges preserving connections
    for $f_1$ in $F_i$ do
        for $f_2$ in $N_{F_i}(f_1)$ do
            b $\leftarrow$ $G_i$.edges[(f1,f2)]; // bond(s) connecting $f_1, f_2$
            rule $\leftarrow$ apply_rule($f_1, f_2, b$);
            V.add(rule);
    $D_G$.append($G_i$);
for $G_i$ in $D_G$ do
    $H_i \leftarrow$ traverse_dag($G_i, G$);
    H.append($H_i$);
Out: H,G
```

```
Algorithm 2: function traverse_dag($G_i, G$)
Input: $G_i$, $G$, $N_{G_i}$
// graph of fragments, motif graph, neighbor iterator
paths $\leftarrow$ all_pairs_shortest_paths($G_i$);
path_len = 0;
for src in paths do
    for dest in paths[src] do
        if len(paths[src][dest]) $>$ path_len then
            path_len $\leftarrow$ paths[src][dest];
            longest_path $\leftarrow$ paths[src][dest];
visited $\leftarrow \{\}$;
visited[src] $\leftarrow$ True;
root $\leftarrow$ Node(src, main = (src in longest_path));
Q $\leftarrow$ queue([(root,src)]);
while !Q.empty() do
    prev_node, prev $\leftarrow$ Q.dequeue();
    for cur in $N_{G_i}$(prev) do
        if visited[cur] then
            continue
        $e \leftarrow G_i$.edges[(prev, cur)];
        e_index $\leftarrow$ find_edge(e, G.edges[(prev,cur)]);
        cur_node $\leftarrow$ Node(cur, main = (cur in longest_path));
        prev_node.add_child(cur_node, e_index);
        vis[cur] $\leftarrow$ True;
        Q.enqueue((cur_node, cur))
Out: root
```

and choosing a consistent ordering over neighbors (N_{G_i}) to determine the random walk sequence. We elaborate on the reasoning behind this canonicalization in Appendix B.4. The DAG constraint enables our graph diffusion process to become a generator of new molecules (as will be discussed in Appendix D), in addition to capturing the distribution of existing ones.

Thus, we specifically ask experts to create segmentations that are acyclic, which they naturally do in nearly all cases anyway. In the case of monomers, this canonicalization is consistent with the IUPAC nomenclature(IUPAC, 1997) of linearizing a monomer via its longest (main) chain, where N_{G_i} should iterate over neighbor fragments that descend side chains before the consecutive fragment on the backbone of the main chain. More specifically, src and dest in Algorithm 2 correspond to the first and last group of the main chain.

B.2. Extracting Walks From Literature (Case Study of Group Contribution)

The Group Contribution dataset includes a compilation of motifs characterized for gas separation, including common organic chemical functional groups as well as important scaffold functional groups such as Triptycene and its derivatives, dioxin and its derivatives, and N-methylphthalimide and PIM-1 and its derivatives (Wang et al., 2018; 2020). These functional groups contribute significantly to maintaining the structures and properties of 3D scaffold building blocks in polymer self-assembly, which in turn play a significant role in gas separation processes, i.e. the separation of $H_2, H_2/N_2, O_2, O_2/N_2, CO_2, CO_2/CH_4$ which are common separation tasks important in gas and oil industry. The steps we take for compiling this dataset of segmented monomers are as follows:

1. We obtain an established compilation of groups (Park & Paul, 1997; Wu et al., 2021) for microporous polymers.
2. We visually segment the monomers in (Wang et al., 2018) into random walks over the groups identified in Step 1.
3. We collect experimental permeability and separation performances for 114 of the monomers identified in Step 2.

In addition to the motifs used here, the concept of such segmentation arises naturally across other application domains in chemical design. Within synthetic organic chemistry, molecular design plays a governing role in advancing new technology (Bronstein et al., 2020). Understanding of the behavior of a molecule or polymer in an application is commonly described by experts using the function of key subparts, particularly key functional groups, scaffold structures, and backbone architectures within a molecule or monomer, and their arrangement relative to each other, rather than considering atom-by-atom or a molecule as a whole. In chemical design, new molecules can be complex and, when designed by hand in traditional ways, are built from these relatively modular subcomponents. This approach naturally takes advantage of the physical laws by which molecules are built by synthesis reactions, where a discrete set of additions and substitutions are allowed to finally construct a desired target structure. Such methods of chemical design find broader application in drug discovery for pharmaceuticals, surfactant and detergent design (Blunk et al., 2006; Li et al., 2020), organic semiconductors (Bronstein et al., 2020), photoinitiators (Lee et al., 2022), and more recyclable plastics (Türel et al., 2023), among other uses, in each of which chemists fine-tune properties of such components by adjusting the selection and arrangement of these sub-structures, or otherwise use them as a guide for understanding performance.

Utilizing groups from existing structures as well as discovery of new and novel structures, researchers can predict performance, find new uses for existing molecules, discover new molecules, and further optimize structures for better performance. Utilizing machine learning models has been show to drastically decrease the time and cost of such methods while simultaneously improving throughput by creating and screening novel structures in a single step and providing researchers with predictions of target molecules that have higher potential for success, which are then verified by experts. As presented by (Wu et al., 2021), different structural elements and functional groups present in effective drug molecules can be identified and recombined in new architectures. These novel structures can then be tested using computer models to benchmark likely efficacy given new targets or modifications to binding sites.

B.3. Graph Augmentation

The motif graph is the directed, multi-edge graph $G = (V, E)$. When traversing to a previously seen motif v, there is ambiguity in whether the random walk forms a cycle vs creating a copy of the previous motif and appending to the trajectory. To remove this ambiguity, the random walk traverses a duplicate node, v_k for the latter case. A dataset of molecules and their representations, $D := \{(M, H_M)\}$ thus induces “an augmented version of G” =: G'. For each $v \in V$, let

$K_v = \max(\text{count}(v, H_M)$ for M). We create duplicates for v and the in/out-edges of v:

$$V' \leftarrow V \cup \bigcup_{v \in V} \{v_k \mid k = 0, \ldots, K_v - 2\} \tag{9}$$

$$E' \leftarrow E \cup \bigcup_{v \in V} \{(v_k, v', e) \mid (v, v', e) \in E, \forall k = 0, \ldots, K_v - 2\} \tag{10}$$

$$E' \leftarrow E \cup \bigcup_{v \in V} \{(v', v_k, e) \mid (v', v, e) \in E, \forall k = 0, \ldots, K_v - 2\}. \tag{11}$$

Molecule $M = (V_M, E_M)$ is then a rooted subgraph of G'. In the main text, we refer to G as its augmented version, to simplify the notation.

B.4. Data Augmentation

Like the Simplified molecular-input line-entry system (SMILES), our description, $\hat{H}_M$, of a molecule is not unique. We tried, to varying extents, balancing between canonicalizing the description vs applying data augmentation during the grammar training phase.

As described in Algorithm 2, we linearize a molecule by first setting its “main chain” – the longest shortest path of H_M. If this happens to be part of a cycle, we disregard one edge. If there are multiple longest shortest paths, we choose the one whose first differing node comes first in our canonical ordering over the nodes of G.

We tried two types of data augmentation:

1. Reversing the direction of the main chain.
2. For each node, trying every permutation over the side chains descending from it.

However, we noticed no practical improvements in training loss or downstream task performance when either of the two types of augmentation were applied. We believe that, given our parameter estimation procedure, the consistently applied canonicalization over the nodes of G improves data-efficiency by significantly reducing the hypothesis space.

C. Building the Motif Graph & More Related Works

Expanding on Section 3.1, we apply a subgraph-matching algorithm with pseudocode in Algorithm 3 over all pairs of motifs v_1, v_2. This algorithm is embarrassingly parallel and runtime-efficient as the subgraph v_{R_l} is, unless specified otherwise, a few atoms or a ring. RDKit(Landrum, 2016) provides out-of-the-box implementations for subgraph matching optimized for molecular sub-fragments like rings, enabling a significant speedup in runtime.

C.1. Connection to Dual Graph of Geo-DEG's Meta-Grammar

Our proposed directed multi-digraph also conceptualizes the dual graph of the Geo-DEG meta-geometry. The essence of the Geo-DEG meta-geometry lies in its completeness, a characteristic inherently inherited by our proposed digraph. A significant advantage of our approach is the substantial reduction in complexity. To elucidate this process, consider the construction of our multi-digraph from the Geo-DEG meta-geometry, denoted as $G_g = (V_g, E_g)$. The initial step involves replacing each node in V_g, which represents a junction tree, with all feasible molecule structures derived from motifs that maintain the same junction tree structure. Subsequently, we augment E_g with fully connected edges between these sets of molecule structures. The dual graph, G_dg, is then derived from G_g, where each node from V_g is transformed into an edge, and each edge from E_g becomes a node. This dual graph not only preserves the completeness of the original graph but also provides an intuitive representation of molecular assembly. Each node in the dual graph symbolizes a motif, and traversing this graph illustrates the process of assembling a molecule by adding motifs. To refine this representation, we eliminate duplicate nodes in the dual graph, ensuring each node's uniqueness.

The representation's completeness is maintained because every possible molecule structure derivable from the motifs is accounted for in the dual graph. Each pathway through the graph represents a unique assembly sequence of motifs, translating into a distinct molecular structure. The reduction in complexity arises from the transformation process. By converting the original graph into its dual form, we reduce the granularity of representation. Instead of representing every possible molecular structure as a separate node, we represent them as pathways through the dual graph. This approach

Algorithm 3: function build_motif_graph(V)

```
Input: V
// motifs
G = graph(V);
for $v_1$ in $V$ do
    for $v_2$ in $V$ do
        for $l1$ in $v_{1_R}$ do
            sub_2$_r$ ← extract_subgraph$(v_2, v_{1_{R_l 1}})$;
            b2_sub ← substruct_matches$(v_2$, sub_2$_r$;
            b2_all ← isomorphisms_iter$(v_2$, b2_sub);
            conn_b1 = []; for b1 in b1_all do
                if connected$(v_1(v_{1_{R_l 1}} + b1))$ then
                    conn_b1.append(b1);
            for l2 in $v_{2_R}$ do
                sub_1$_r$ ← extract_subgraph$(v_1, v_{2_{R_l 2}})$;
                b1_sub ← substruct_matches$(v_1$, sub_1$_r$;
                b1_all ← isomorphisms_iter$(v_1, b1$_sub);
                conn_b2 = [];
                for b2 in b2_all do
                    if connected$(v_2(b2 + v_{2_{R_l 2}}))$ then
                        conn_b2.append(b2);
                for b2 in conn_v2 do
                    for b1 in conn_b1 do
                        sub_1 ← $v_1(v_{1_{R_l 1}} + b1)$;
                        sub_2 ← $v_2(v_{2_{R_l 2}} + b2)$; if isomorphic(sub_1, sub_2) then
                            $e_{l1,l2}$ ← $(v_1, v_2$, r_grp_1: $v_{1_{R_l 1}}$, r_grp_2: $v_{2_{R_l 2}}$, $b_1$: $b_2$: $b_2$);
                            G.add_edge(e_$\{l1, l2\}$);
Out: G
```

significantly decreases the number of nodes and edges required, leading to a more manageable yet complete representation of the molecular structures.

C.2. Connection to Graph Coarsening

Mathematically, the motif graph advocated in this work is the *quotient graph* of the molecular graph, under the equivalence relation defined as $u \equiv v$ if nodes u and v belong to the same motif. As our motifs do not overlap and jointly cover all nodes of the molecular graph, they define a partitioning of the graph. In scientific computing, collapsing each partition into a single node and retaining edges crossing partitions is called *graph coarsening*, which is a commonly used technique to solve large-scale problems, notably solving sparse linear systems of equations (Chen et al., 2022). Working on the coarsened version of the graph (i.e., the quotient graph) is computationally attractive as the graph size is much smaller. Moreover, when applied to machine learning problems such as graph classification, it is demonstrated that the representation learned from the quotient graph can be as predictive as that learned from the original graph (Chen et al., 2023; Ma & Chen, 2021; Cai et al., 2021). Favorably, a unique scenario of this work is that all concerned (molecular) graphs share the same set of motifs, which brings in the potential benefit of learning better molecule representations based on motif representations that form the basis of all molecules.

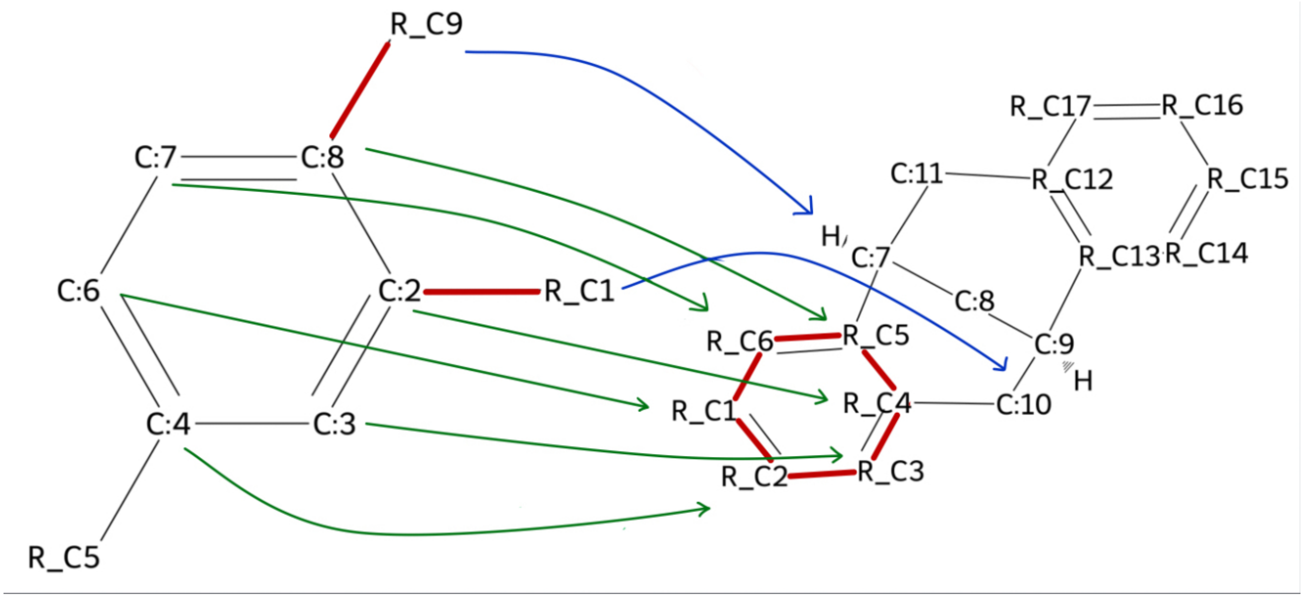


Figure 10. Following notations of main text, $\{v_1\}_{r_1} = \{1, 9\}, \{v_1\}_{r_2} = \{1\}, \{v_1\}_{r_3} = \{9\}, \{v_1\}_{r_4} = \{5\}$. $\{v_2\}_{r_1} = \{1, 2, 3, 4, 5, 6\}, \{v_2\}_{r_2} = \{12, 13, 14, 15, 16, 17\}$. We annotate $e_{1,1}$, where $b_1 = \{6, 4, 3, 2, 8, 7\}, b_2 = \{10, 7\}$ provides the "certificate" of a successful match.

C.3. Connection to Random Walk Literature

Our parameterization of the random walk is by learning a graph heat diffusion process over the motif graph G. The relationship between graph heat diffusion and random walk has been studied before (Masuda et al., 2017), but we integrate two new ideas: 1) making the Laplacian (edge weights) learnable and dynamically adjustable, and 2) conditioning the adjustment on an order-invariant memory. The justification as to why we don't just use autoregressive models is part of a larger discussion on the respective merits of autoregressive models vs grammar-based approaches. In data-efficient settings, previous works (Guo et al., 2023a;b) show grammar (esp. context-free grammar) work well due to the relatively small (tens/hundreds) number of examples needed to learn valid rules and derivation sequences. Meanwhile, the number of possible hidden states that autoregressive models (Li et al., 2018; You et al., 2018; Liu et al., 2018) are parameterized to learn is exponential (to the length of the sequence), and learning a good parameterization is difficult (Jin et al., 2018; 2020). We take a middle ground, combining the data-efficient advantages of context-free grammar and the expressivity of autoregressive models, by introducing a context-sensitive grammar which utilizes a set-based memory during the random walk. The set-based memory mechanism $c^{(t)}$ keeps an order-invariant memory of the nodes visited so far. Without the memory mechanism, our model becomes an order 1 Markov process. Previous literature show that higher-order random walks are required to capture temporal correlations in edge activations (Rosvall et al., 2014; Masuda et al., 2017), with a tradeoff of complexity and practicality. In the design of complex and modular structures, the order 1 Markov assumption is not sufficient (see footnote 2 in the paper). Meanwhile, higher-order models make it difficult to scale our grammar to larger motif graphs. We take a middle ground by introducing a set-based memory state, replacing the entire visit history with a summary of node visit counts. In particular, prior works study how memory mechanisms in random walks affect exploration efficiency (Fang et al., 2023; Gasieniec & Radzik, 2008) and enable negative/positive feedback (Fang et al., 2023; Pemantle, 1988). Our results in Section 4.2.2 demonstrate the efficacy of this approach.

D. Grammar Learning

D.1. Graph Diffusion Strategy

Our strategy is to encode the dataset of walk trajectories by training the parameters of our graph diffusion process to recover the ground-truth state of a particle being diffused over the motif graph. We use stochastic gradient descent and choose between a "forcing" approach (where a single particle transitions from one state to another) and a "split" approach (where a single particle splits its mass equally along the out-edges of its current state). See the pseudocode in Algorithm 6.

```
Algorithm 4: function re_order(childs)
Input: childs // children
ordered_childs ← sorted(childs, key = λ c: (c.main, c.id));
Out: ordered_childs // re-ordered children, with side-chain descendants first
```

```
Algorithm 5: function dfs_walk(cur, traj)
Input: cur, traj // children
traj.append(cur);
childs ← re_order(cur.children);
for c in childs do
    cur_len ← len(traj);
    dfs_walk(c, traj);
    if !c.main then
        traj ← traj + reverse(traj[cur_len:]);
```

D.2. Visualizing Learning Process

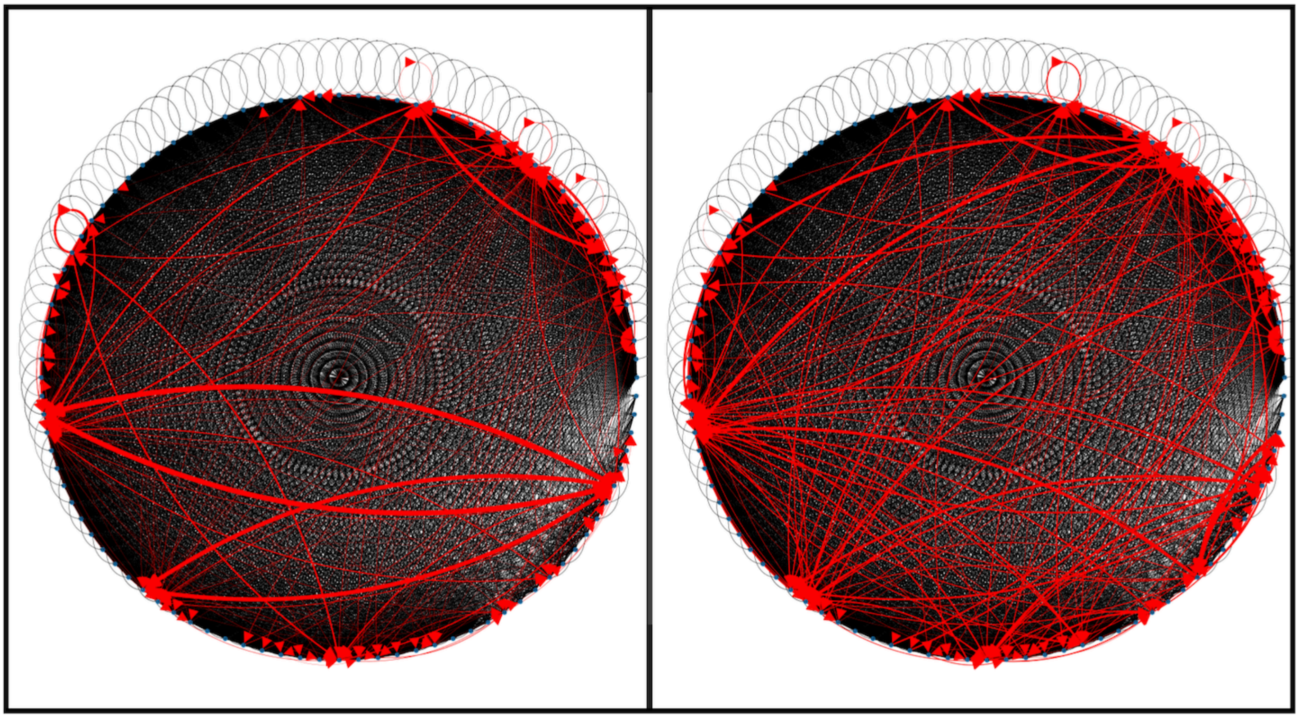


Figure 11. (Left) The raw data of Group Contribution. The edge thickness is proportional to the number of monomers whose random walk representations traverse the edge. (Right) The learned parameter matrix of E after training converges The grammar both retains essential nodes and edges and smoothens the distribution of edge weights.

In Figure 11 and Figure 12, we see our grammar's capacity to estimate the prior edge weights, E, through training, as well as correct the edge weights via a memory-sensitive adjustment during the random walk. Weights of edges that are commonly traversed after $G14$ will be amplified during training, and weights of edges that visit $G14$ from another state will be diminished.

Algorithm 6: function algo-diffusion

```
Input: T, G, alpha, strategy // number of time-steps, motif graph, learning rate, either 'split'
       or 'forcing'
E ← rand($|G| \times |G|$);
W ← rand($|G|, |G| * |G|$);
b ← zeros($|G|$);
for $(H_M, E_M)$ in D do
    $c^{(0)}$ ← [0 for v in G];
    $x^{(0)}$ ← [1 if v==$H_M$.root else 0 for v in G];
    $p^{(0)}$ ← [1 if v==$H_M$.root else 0 for v in G];
    if strategy == 'forcing' then
        traj ← [];
        dfs_walk($H_M$.root, traj);
    for $t = 0, \dots, T-1$ do
        $c^{(t+1)} = \frac{t}{t+1} \cdot c^t + \frac{1}{t+1} \cdot p^t$;
        $W^{(t+1)} = E + f(c^{(t+1)})$;
        $x^{(t+1)} = x^t + (D - W^{(t+1)})x^t$;
        if strategy == 'forcing' then
            $p^{(t+1)}$ ← [1 if v == traj[(t+1)%len(traj)] else 0 for v in G];
        else
            for i in G do
                $p_i^{(t+1)} \leftarrow \sum_{(j,i) \in E_M} \frac{p_j^t}{d_j}$;
        Loss ← MSE($x_t$, $p^t$);
        E ← E - $\frac{\text{dLoss}}{\text{dE}}$;
        W ← W - $\alpha * \frac{\text{dLoss}}{\text{dE}}$;
        b ← b - $\alpha * \frac{\text{dLoss}}{\text{db}}$;
Out = E,W,b
```

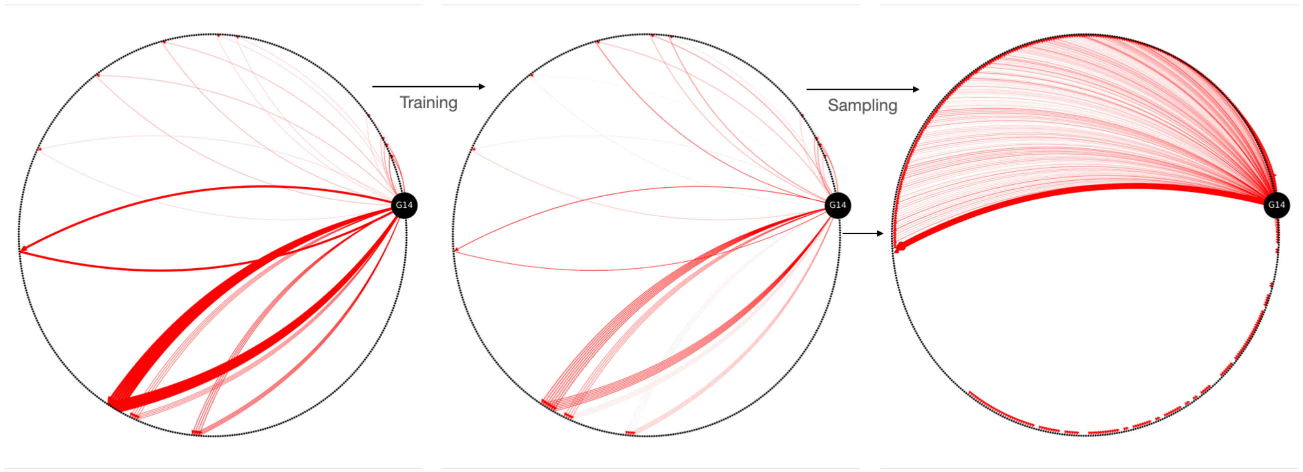


Figure 12. We show the weight evolution of the edges incidental to $G14$ on HOPV. (Left) After processing the raw dataset into random walks, we visualize the empirical distribution of edge traversals. (Middle) After learning our context-sensitive grammar, we plot the prior edge weights, i.e. the memory-free parameter E. (Right) We plot the transition probabilities starting at $G14$ during the random walk generation process.

E. Property Prediction

E.1. Graph Neural Network Design Choices

Table 8. Hyperparameter settings for property prediction

Hyperparameter	**Value**
Number of layers	5
Activation	ReLU
Hidden dimension	16
Motif featurization	Morgan fingerprint
Motif feature dimension	2048
Input feature dimension [5]	2048 + 2048 + $\|G\|$
Batch Size	1
Learning Rate	1e-3

We apply a Graph Isomorphism Network (Xu et al., 2019) with hyperparameters in Table 8. For molecule M with representation H_M, the node-level features include: a) the Morgan fingerprint of the motif v_i (dimension 2048), b) the memory-free weights of its out-edges (dimension $|G|$), i.e. E[i]. We also concatenate the Morgan fingerprint of M.

E.2. Bag-of-Motifs Design Choices

We obtain $|G|$-dimension motif-occurrence feature vector for each M. Similar to Ours, we concatenate the Morgan fingerprint of M to it. We use XGBoost with 16 estimators (boosting rounds) and maximum tree depth of 10.

E.3. Optimization Design Choices

We apply the Adam optimizer with stochastic gradient descent. To mitigate noisy training dynamics, we report the mean and standard deviation over 3 runs, corresponding to 3 random seeds during data splitting. We initialize weights using the Gaussian distribution.

F. Generating Novel Random Walks

We illustrate the generation of the random walk with notation $G81 \rightarrow G82 \rightarrow G274 \rightarrow G82 : 1$ in Figure 13. Our graph resolves any ambiguity of whether to revisit $G82$ or attach a new copy of $G82$ to the molecule by attaching a colon for each newly attached motif that has a naming conflict. This is possible after augmenting G with duplicates of the motif (see Appendix B.3), which, in practice, has a negligible increase the complexity of G.

Our implementation in Algorithm 7 handles the distinction between revisiting a previous node vs adding a new duplicate of the same motif as a previous node through mask_attach (new nodes which can be attached) vs mask_return (the node which the random walker can backtrack to). This distinction is done by creating duplicates of nodes for each revisit (see B.3).

We guarantee 100% validity since we can explicitly check the possible motifs which can be attached to M at each step. When there are neither new motifs to attach, nor existing motifs to return to, the generation terminates with the current M being the final generation output. Please refer to the GitHub for details of the implementation.

As shown in Figure 14, applying our generation method produces artifacts of learning that invite further scrutiny: "rules" of consecutive motifs. The second example in Figure 14 shows there are only two possible motifs (green) that can be attached to the $G297$ end of a molecule with the $G239$ and $G297$ functional groups (ignoring the return back to G239, which transitions the state but does not attach a new motif). In the first example, the distribution of possible new motifs to attach to the $G305$ end of a molecule with $G262$ and $G305$ appears more uniform.

[5] We concatenate the molecule's 2048-dimensional morgan fingerprint to the input features. We concatenate the edge-weighted adjacency matrix to the input features.

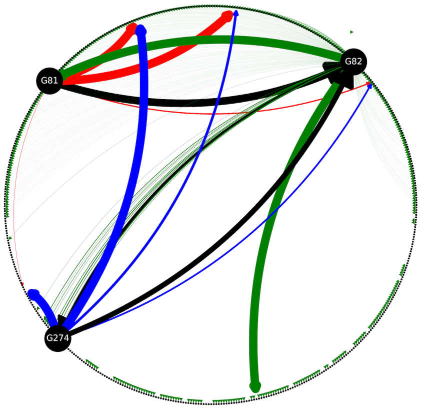


Figure 13. Generation of a random walk $G81 \rightarrow G82 \rightarrow G274 \rightarrow G82 : 1$; The possible transitions from $G81$, $G82$ and $G274$ are in Red, Green, and Blue (with thickness proportional to probability).

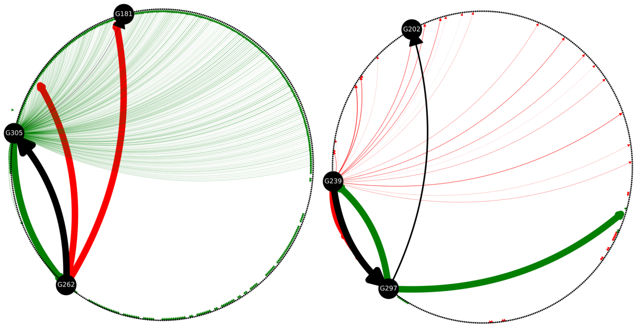


Figure 14. Generation process of novel random walks on HOPV: (Left) $G262 \rightarrow G305 \rightarrow G181$ and (Right) $G239 \rightarrow G297 \rightarrow G202$; The possible transitions from the first and second visited nodes are in Red and Green.

F.1. Extracting Context-Sensitive Grammar Rules

One side product of our generation and verification procedure is the ability to extract "hard" rules. A hard rule is when a certain edge *must* be traversed (probability of 1) under a certain memory and at a certain state. Although our memory is invariant to the order of visited nodes thus far, we search for hard rules by using a best-first algorithm to store all promising trajectories. Table 9 is a compilation of hard rules learned by our model on the PTC dataset.

G. Detailed Case Study: Harvard Organic Photovoltaic Dataset

The Harvard Organic Photovoltaic Dataset (HOPV15) is a comprehensive collection that bridges experimental photovoltaic data with quantum-chemical calculations, serving as a crucial resource in the field of organic photovoltaics. This dataset includes experimental results from literature and corresponding quantum chemical data for a wide range of molecular conformers. These are analyzed using various density functionals and basis sets, including both generalized-gradient approximation and hybrid designs. A key feature of HOPV15 is its utility in calibrating quantum chemical results with experimental observations, aiding in the development of new semi-empirical methods, and benchmarking model chemistries for organic electronic applications. The dataset employs the Scharber model to compute the maximum percent conversion efficiencies for 350 studied molecules, focusing on their HOMO (Highest Occupied Molecular Orbital) values.

G.1. Segmentation Strategy

Our segmentation approach involved systematically categorizing molecules based on their functional groups and ring structures. We separated standard functional groups (e.g., vinyl, alcohol) and individual rings (e.g., benzene, thiophene,

Algorithm 7: function generate

```
Input: G // motif graph
root_M ∼ V; // can sample according to prior
loop_back; // whether to loop back (applies for monomers)
root ← Node(root_M);
H ← root;
M ← molecule(root_M); // initialize the molecule
t ← 0;
$c^{(t)}$ ← [0 for v in V];
terminate ← False;
while !terminate do
    $p^{(t)}$ ← [1 for v in V if v==H.val else 0];
    $c^{(t+1)} \leftarrow \frac{t}{t+1} \cdot c^{(t)} + \frac{1}{t+1} \cdot p^{(t)}$;
    $W^{(t+1)}$ ← E + f($c^{(t+1)}$);
    $x^{(t+1)} \leftarrow x^{(t)}$+ (D-$W^{(t+1)} x^{(t)}$);
    mask_attach, mask_return ← mask_possible(M,G,H);
    mask ← mask_attach | mask_return;
    $x^{(t+1)} \leftarrow \frac{\text{mask}*x^{(t+1)}}{(\text{mask}*x^{(t+1)}).\text{sum}()}$;
    cur ← sample($x^{(t+1)}$);
    if cur is not None then
        if loop_back and cur == root_M then
            terminate ← True;
        else
            if mask_attach[cur] then
                M ← attach(M, molecule(cur));
                H.child ← Node(cur);
                H ← H.child;
            else
                H ← H.parent;
            else
                if loop_back then
                    return M, root, False;
                else
                    break;
return M, root, True;
Out: molecule, representation of molecule, boolean indicating validity
```

pyrrole) to understand their unique contributions to photovoltaic properties. Additionally, we paid special attention to complex structures with consecutive rings, acknowledging their impact on the optical and electronic characteristics of the materials. These parameters are impactful to the molecular's HOMO value, which are essential for calculating the open circuit potential and short circuit current density, leading to an understanding of percent conversion efficiency.

The segmentation strategy is particularly focused on the differentiation and categorization of molecular structures based on their photovoltaic properties and electronic configurations. This includes the separation of standard functional groups such as vinyl, alcohol, ketone, aldehyde, amine, ester, and amide, each identified as individual black fragments. This separation is critical in analyzing their distinct contributions to photovoltaic efficiency and electronic properties.

Moreover, the dataset and segmentation emphasize the unique characteristics of individual rings like benzene, pyrrole, and thiophene by treating them as separate black fragments. This distinction is vital due to their specific Pi-orbital electron

Table 9. Under our string-based implementation, A[→B] encodes a random walk trajectory of A→B→A. All rules shown are valid, as verified to correspond to valid molecules that can be constructed following the random walk trajectory.

Trajectory A	⇒ Trajectory B
['G4']	['G4', 'G2']
['G27']	['G27', 'G6']
['G115']	['G115', 'G6']
['G218']	['G218', 'G6']
['G283']	['G283', 'G6']
['G290']	['G290', 'G6']
['G301']	['G301', 'G6']
['G335']	['G335', 'G6']
['G368']	['G368', 'G6']
['G466']	['G466', 'G231']
['G272']	['G272', 'G271']
['G362']	['G362', 'G361']
['G205']	['G205', 'G202']
['G435']	['G435', 'G434']
['G167']	['G167', 'G166']
['G436']	['G436', 'G166']
['G224']	['G224', 'G225']
['G2', 'G4']	['G2[->G4]']
['G202', 'G205']	['G202[->G205]']
['G434', 'G435']	['G434[->G435]']
['G361', 'G362']	['G361[->G362]']
['G333', 'G393']	['G333', 'G393', 'G333:1']
['G224', 'G225', 'G224:1']	['G224', 'G225[->G224:1]']

delocalization, which plays a crucial role in the photovoltaic properties of the molecules. The segmentation method goes a step further in dealing with complex structures possessing multiple consecutive rings, such as thieno[3,4-b]pyrazine, carbazole, and 2,5-dimethyl-3,6-dioxo-2,3,5,6-tetrahydropyrrolo[3,4-c]pyrrole. These structures are treated as individual entities to accurately reflect their unique HOMO-LUMO bandgaps and electrochemical characteristics, which are central to their functionality in organic photovoltaics.

The segmentation strategy also pays special attention to groups of 2-3 consecutive symmetrical thiophene or pyrrole units, maintaining these as a single black group. This decision is based on the understanding that the electron cloud delocalization across these repeating units significantly influences the optical and electronic properties of the molecules, impacting factors such as light absorption, charge transport, and luminescence. Such approach is essential for advancing the understanding of molecular alignment and stability, thereby optimizing the functional properties of photovoltaic materials.

Meanwhile, all the red group along with the segmented black group are chosen to be either single atom, or closet conjugated rings if the black group is too small (just one or two atoms). This method helps reduce the redundancy and computational resources of the red group.

G.2. Heuristic Based Fragmentation

We adopted a heuristic-based, deterministic algorithm to segment molecules across all datasets for our ablation study. Below, we analyze its segmentation quality on the HOPV dataset. We cleave on any bond that satisfies one or either of these conditions:

1. Bond connecting two rings.
2. Bond connecting a ring and an atom with degree greater than 1.

This algorithm works for molecules with rings, but tends to not capture functional groups consistently. It either fails to sufficiently segment groups attached to ring like A2 in Figure 15 or cleaves on every ring even when they should be kept together like B1 in Figure 15.

G.3. Analysis of Learnt Representations

In this section, we perform an alternate and more accepted means of analysis than the 2D t-SNE analysis done in Section 4.4.2. We seek to understand the agreement between our property predictor's learnt representation and the structural similarity over HOPV's test set molecules. Since the final layer embedding is used for prediction, we expect molecules with similar properties to have similar embeddings.

In Figure 16, several groups of trends stand out. Particularly, highlighted in green are cases where the embedding similarity is high despite dissimilar HOMO property values; blue marks cases where the embedding similarity is low, and red marks sections that are similar in property, structure, and embedding. We detail each of these, basing comparison against molecule 50 for illustration:

- For the topmost green section, molecules in the range 1-4 have similar components as those with higher HOMO values, though are much smaller in size and relatively disordered. For instance, molecules 3 and 4 each share key subcomponents (thiophene groups) with molecule 50, despite having quite different overall structure. The embedding similarity between (50, 4) and (50, 3) is thus medium-low and medium-high.
- For the red sections along the diagonal, molecules in the ranges 14-16 and 17-26 cluster together. These tend to have an over-representation of electron-withdrawing groups in in non-symmetric locations in the structure, particularly methoxy, cyano, and carbonyl groups, without sufficient electron donating groups. Molecules 15 and 20 are shown as examples, and their embedding similarity is high. Blue outlines mark similar sub-groups between 15 and 20.
- For the second-from-top green section, we again consider molecules in the range 18-26, where they show high similarity to the highest band in the range 47-63. These share many component structures, for instance thiophenes groups and derivatives. Molecule 23 is shown as an example, and has a barbituric acid core on one side, an electron withdrawing group, with methoxy groups on benzene rings on the other side, with a nitrogen atom between benzene rings, contributing to electron delocalization. The most likely explanation is that similar high-sterics groups have developed similar embeddings in this case.
- For the bottom-right red section, molecules in the range 47-63 generally cluster together, reflecting the model's ability to agree on both structural and property similarity. They tend to have an alternating pattern of electron-donating and electron-withdrawing groups which can increase the HOMO and provide a more direct pathway for charge transport. Yellow outlines mark matching and similar groups with molecule 50. In these cases, more than simply thiophene shows similar or the same structure. The embedding similarity between (50, 52), (50, 57), (52, 57) are all medium-high.

These insights show how complex molecule structure affects the measured property in this application, and how both structure and property are captured in the embedding. We hope the analysis provides more insights into how structural priors in our representation facilitates learning and generalization.

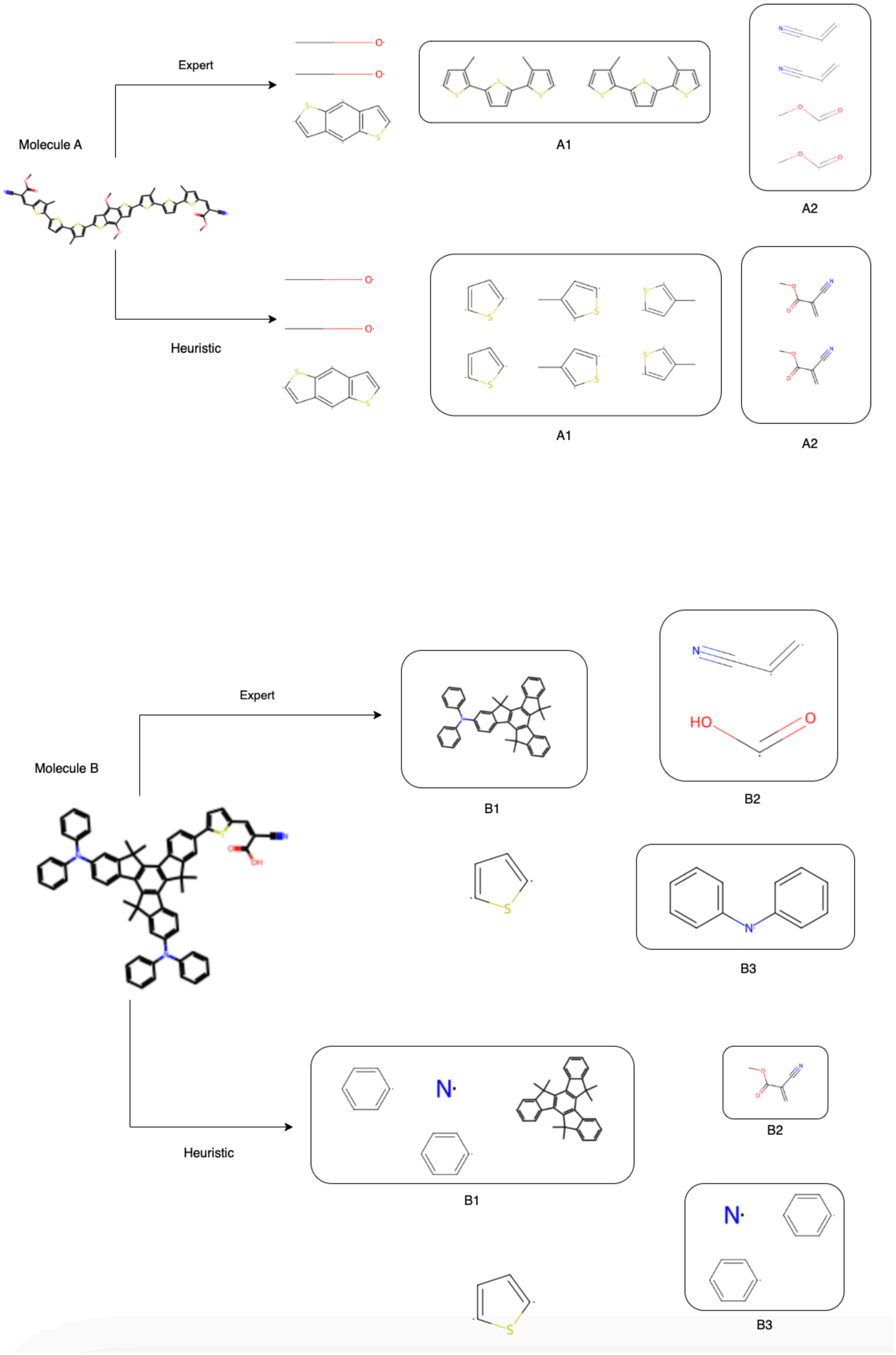


Figure 15. We detail the difference between the expert and heuristic segmentations, highlighting how the heuristics are not sufficiently capable. For example, the expert segmentation keeps the 3 thiophene rings together in A1, while the heuristic breaks them up. Similarly, in B1, the expert treats the consecutive rings as one fragment, whereas the heuristic cleaves on bonds connecting them.

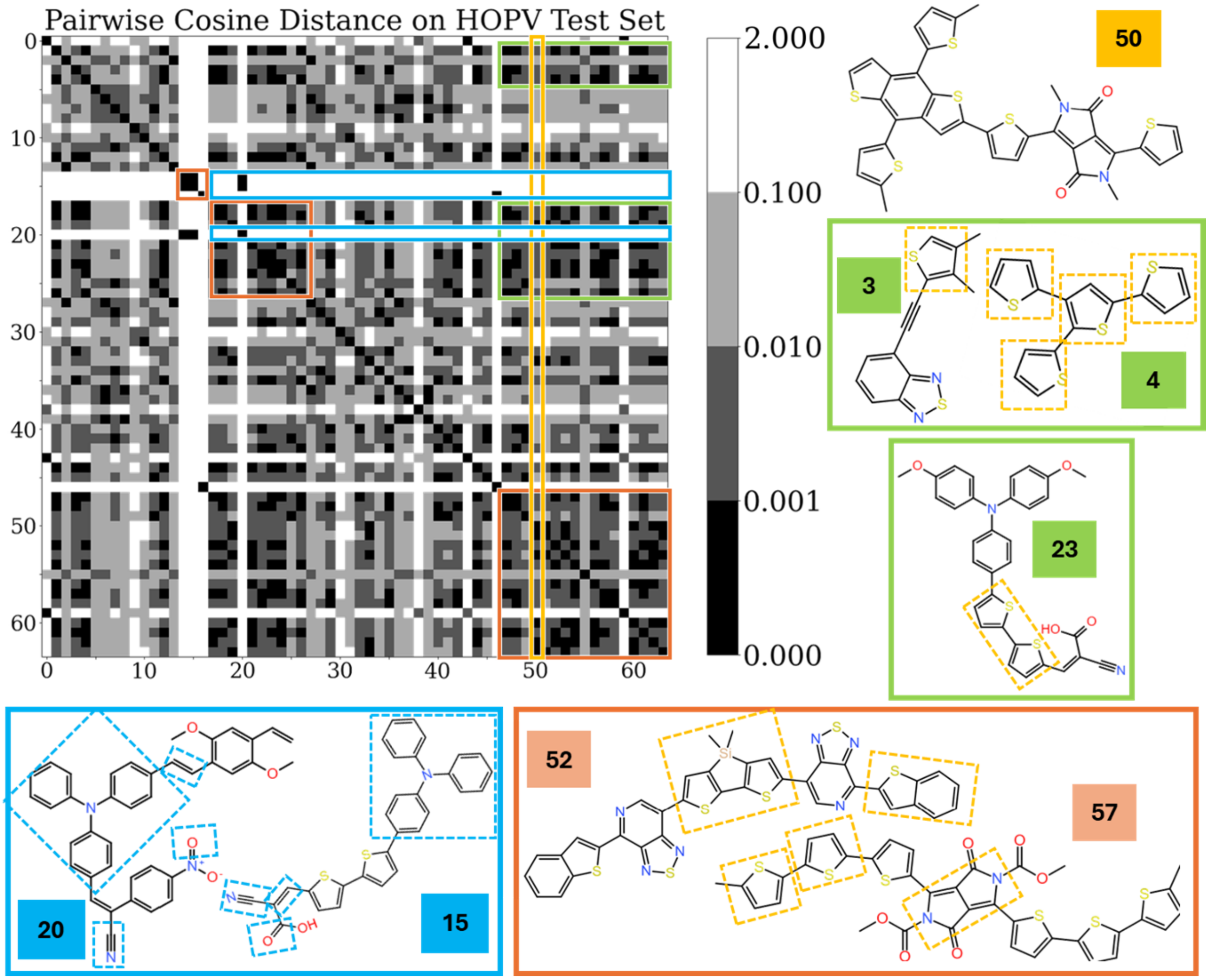


Figure 16. There are 64 molecules in this test set indexed from lowest to highest HOMO value. The above grid visualizes the distance between each pair of molecules as a cosine distance between the final layer embeddings of our model, with darker color representing lower distance (higher similarity). We use 4 quantiles, and refer to their ranges as low, medium-low, medium-high, and high similarity.