Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Molecular Property Prediction from SMILES and SELFIES Sequences

Rokas Garbacauskas

rokas.garbacauskas@ff.stud.vu.lt

Vilnius University

Stepas Toliautas

Vilnius University

Research Article

Keywords: molecular property prediction, SMILES, SELFIES, transformer, multi-task learning, molecular representations

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10221092/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Molecular Property Prediction from SMILES and SELFIES Sequences

Rokas Garbacauskas[1*] and Stepas Toliautas[1]

[1*]Institute of Chemical Physics, Vilnius University, Faculty of Physics, Vilnius, Lithuania.

*Corresponding author(s). E-mail(s): rokas.garbacauskas@ff.stud.vu.lt;

Abstract

Predicting molecular properties directly from string representations avoids the computational cost of explicit 3D structure generation, yet the relative merits of different representations remain poorly understood. We systematically evaluated SMILES (Simplified Molecular Input Line Entry System) and SELFIES (SELF-referencIng Embedded Strings) molecular representations across 50 transformer model configurations on seven properties—five classification tasks (molecular dimension, ring count, chirality, symmetry planes, point group) and two regression tasks (planar fit error, ring plane angles)—derived from 3.9 million density functional theory-optimized structures. SMILES consistently outperforms SELFIES across all properties; model depth improves performance on globally complex properties without overfitting; and multi-task learning benefits classification but degrades regression, with the best configurations achieving combined scores above 0.85 on a normalized seven-task benchmark.

Encoder-only transformer models (2, 4, and 8 encoder blocks; d_{model}) were trained using PyTorch on NVIDIA A100 GPUs (Leonardo cluster, CINECA). SMILES strings were generated with RDKit; SELFIES representations with the SELFIES library. Molecular geometries originate from the Molecule3D benchmark, computed at the B3LYP/6-31G* level of theory via PubChemQC. Point group symmetries were computed using the SOFI library. Three architectural variants were compared: single-task, parallel multi-task, and hierarchical multi-task with cross-attention between property prediction heads.

Keywords: molecular property prediction, SMILES, SELFIES, transformer, multi-task learning, molecular representations

1 Introduction

Predicting molecular properties is central to drug discovery, materials design, and computational chemistry [1]. The standard approach computes properties from 3D molecular geometries, which must first be obtained through conformer generation and quantum mechanical optimization—a process that can require minutes per molecule using density functional theory (DFT) [2]. SMILES (Simplified Molecular Input Line Entry System) [3] and SELFIES (SELF-referencIng Embedded Strings) [4] encode molecular graphs as text sequences, enabling inference at millisecond timescales. Transformer architectures trained on these representations have achieved strong results on various molecular benchmarks [5–7], yet basic questions about representation choice and architectural design remain open.

The first question concerns which string representation encodes molecular structure more effectively. SELFIES uses a context-free grammar that guarantees every string decodes to a valid molecule [4]—a property useful for generative models. SMILES lacks such guarantees but encodes stereochemistry explicitly and maps more directly to chemical concepts. Whether these differences affect property prediction accuracy has not been systematically examined across diverse property types.

The second question concerns how transformer capacity should scale given the limited size of quantum chemistry datasets. Unlike language modeling, where datasets routinely contain billions of tokens [8, 9], accurate molecular property labels require expensive DFT calculations. Our dataset of 3.9 million molecules is large by quantum chemistry standards but small relative to typical transformer training sets. It is unclear whether deeper models (8 vs. 2–4 encoder blocks) will overfit or whether the structural diversity of chemical space provides sufficient complexity to benefit from additional capacity. A related question is whether predicting multiple properties jointly improves performance through shared representations, or whether task-specific models trained independently perform better.

We address these questions through a systematic evaluation spanning 50 model configurations: two representations (SMILES, SELFIES), three architectural variants (single-task, parallel multi-task, hierarchical multi-task with cross-attention), and three depths (2, 4, 8 encoder blocks), evaluated on seven molecular properties. The properties range from trivial (ring count) to demanding (point group symmetry, requiring global structural reasoning). This study compares SMILES and SELFIES for property prediction, characterizes how performance scales with model depth for properties of varying complexity, and quantifies the benefits and limitations of multi-task learning across classification and regression targets. The results yield practical guidelines for representation choice, model sizing, and architecture selection in molecular property prediction.

2 Related Work

2.1 Molecular Representations

SMILES [3] remains the dominant text-based molecular representation, encoding molecular graphs as ASCII strings through a linear traversal with branch and ring-closure annotations. The format is compact and human-readable, but random string manipulation often produces invalid molecules—a limitation for generative applications. SELFIES [4] addresses this through a context-free grammar where every string maps to a valid structure, at the cost of less intuitive token semantics. Fixed-length fingerprints such as Extended Connectivity Fingerprints (ECFP) [10] compress molecular structure into vectors suitable for traditional machine learning but discard the sequential structure that transformers exploit.

2.2 Transformer-Based Molecular Models

Transformers [11] have been adapted to molecules primarily through SMILES-based language modeling. ChemBERTa [5, 12] pre-trains on millions of SMILES strings using masked token prediction, then fine-tunes for property prediction. MolBERT [13] augments this approach with auxiliary tasks encoding chemical knowledge. The Molecular Attention Transformer [14] incorporates graph adjacency directly into attention weights, while Schwaller et al. [6] applied transformers to reaction prediction. Ross et al. [7] demonstrated that scaling SMILES language models improves property prediction.

These architectures typically attach prediction heads only at the final encoder layer, treating all properties equivalently. Multi-task molecular models [15, 16] share representations across tasks but usually through a single shared encoder followed by task-specific heads. Our architecture permits heads at different encoder depths, allowing properties to access features at varying levels of abstraction.

2.3 Graph and 3D Molecular Methods

Graph neural networks (GNNs) process molecules as graphs with atoms as nodes and bonds as edges. Message Passing Neural Networks [17] form the basis for most molecular GNNs, aggregating neighbor information through learned message functions. AttentiveFP [18] and Chemprop [19] extend this framework with attention mechanisms and directed edge messages. MoleculeNet [20] and the Open Graph Benchmark [21] provide standard benchmarks for these methods.

Three-dimensional architectures incorporate spatial coordinates directly. SchNet [22] applies continuous convolutions over interatomic distances; DimeNet [23] adds angular information; PaiNN [24] and SphereNet [25] achieve equivariance to rotations and translations. These methods excel at geometric properties but require 3D structures as input—either from DFT optimization or approximate conformer generators [26]—limiting throughput to thousands of molecules per hour rather than millions. String-based approaches trade some geometric accuracy for the ability to screen chemical libraries at scale without explicit 3D structure generation.

3 Dataset

3.1 Data Source and Preprocessing

The dataset derives from the Molecule3D benchmark [27], which collects ground-state 3D geometries computed at the B3LYP/6-31G* level from the PubChemQC database [28]. We use a curated subset after removing single-atom molecules, fragmented structures, and molecules with non-standard valences, yielding 3.9 million molecules. RDKit [29] generated SMILES strings; the SELFIES library [4] produced SELFIES representations; the SOFI library [30] computed point group symmetries with manual corrections for edge cases. Molecules average 29 atoms (14 heavy atoms) and 30 bonds. The full dataset occupies approximately 3.4 GB on disk across 8 HDF5 files (approximately 1 GB RAM footprint when loaded).

3.2 Molecular Representations

SMILES tokenization uses a regex-based approach yielding a vocabulary of 166 tokens, including single-character atoms (C, N, O), bond symbols (=, #), branch markers, and stereochemistry annotations such as `[C@@H]`. Sequences average 24.9 tokens. SELFIES tokenization produces 109 tokens, all in bracket notation (`[C]`, `[=C]`, `[Branch1]`), with sequences averaging 21.5 tokens—13% shorter than SMILES at the token level despite appearing longer as character strings (Table 1). Both representations include special tokens for padding, unknown symbols, sequence start, and sequence end. Over 99.9% of sequences fall under 50 tokens, well within the 512-token sequence length limit (a model hyperparameter chosen to cover all sequences with margin).

Table 1: Token count statistics for SMILES and SELFIES (1M sampled molecules).

Representation	Mean	Median	Max	Std	95%ile	99%ile
SMILES	24.9	25	111	5.9	33	37
SELFIES	21.5	22	90	4.8	28	31

3.3 Prediction Targets

We predict seven molecular properties: five classification tasks and two regression tasks. These properties were selected to span a range of computational complexity—from tasks where the answer is nearly explicit in the input representation to those requiring global reasoning about molecular structure.

The classification targets span structural and symmetry properties. *Ring count* (7 classes) records the number of rings; this information is largely explicit in SMILES ring closure digits, making it the simplest prediction task. *Molecular dimension* (5 classes) categorizes molecules as linear, almost linear, planar, almost planar, or three-dimensional based on principal component analysis (PCA) of atomic coordinates.

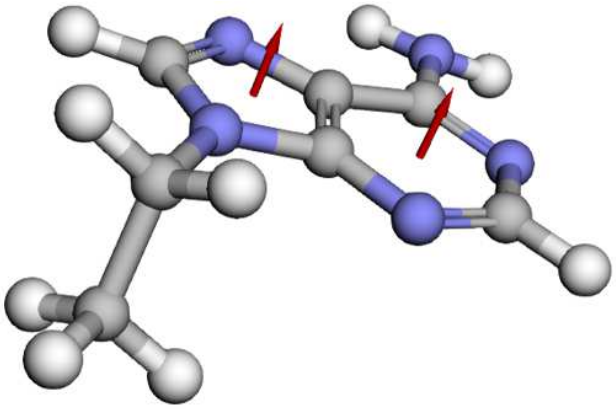
Chirality (binary) indicates whether a molecule lacks mirror symmetry—a property critical for drug activity, as enantiomers often have distinct biological effects [31]. Chirality prediction requires reasoning about 3D spatial arrangements from sequential input. *Symmetry planes* (4 classes) counts molecular mirror planes, considering only planes that permute atomic positions—the trivial in-plane symmetry of planar molecules and the infinite axial planes of linear molecules are excluded. *Point group* classifies molecules by their full symmetry using group-theoretic labels (C_1, C_s, C_{2v}, etc.); this is the most demanding classification task, requiring integration of all symmetry elements across the entire molecule.

The regression targets characterize ring geometry. *Planar fit error* measures the root mean square deviation of ring atoms from their best-fit plane, quantifying how far rings deviate from perfect planarity—relevant for π-conjugation and aromaticity; for molecules with multiple rings, this is averaged across all rings. *Ring plane angles* records dihedral angles between adjacent ring systems, computed from the angle between their best-fit plane normals; these are predicted as a variable-length sequence of pairwise values padded to a fixed maximum length. These angles affect electronic coupling: coplanar rings (0°) maximize π-conjugation while perpendicular arrangements (90°) minimize it.

4 Label Computation and Class Imbalance

All labels were computed from DFT-optimized 3D geometries and pre-stored in HDF5 files. The dataset exhibits severe class imbalance [32]: three-dimensional molecules dominate (90.6%), with planar structures comprising 7.6% and linear structures under 0.01%; similarly, 91.8% of molecules belong to the C_1 point group (no symmetry), and 94% have zero symmetry planes. Ring count derives from cycle analysis of the molecular graph. Molecular dimension uses PCA on atomic coordinates, with class boundaries based on explained variance ratios (Section C). Point groups were assigned using the SOFI library [30], which identifies symmetry operations from 3D coordinates; manual corrections handled edge cases in high-symmetry molecules. Symmetry plane counts follow directly from point group assignments. Chirality labels combine dimensionality and symmetry: a molecule is chiral if it is three-dimensional and has no mirror planes. The regression targets—planar fit error and ring plane angles—measure deviations from planarity using least-squares plane fitting.

Table 2 summarizes class distributions. Imbalance ratios exceed 1000:1 for most properties, reaching 20,183:1 for dimension (almost-linear molecules are rare). We bin tail classes to ensure each category contains at least several hundred training examples: molecules with 6+ rings form a single class (1,320 samples), molecules with 3+ symmetry planes are grouped (1,384 samples), and 21 rare point groups collapse into an "Other" class (398 samples). The 14 retained point groups include C_1, C_s, C_2, C_{2v}, C_{2h}, C_i, C_{3v}, D_{2h}, C_3, C_{nv}, D_2, D_{3h}, D_{2d}, and T_d. Focal loss with class weighting addresses the remaining imbalance (Section 6).



Point symmetry group: C1

Molecule is 3D
Standard deviation out of line/plane: 0.659 Å
Relative non linearity/flatness: 14.7%

The molecule has 0 vertical mirror plane(s)
The molecule is chiral

The molecule is comprised of 2 rings:
Normal of ring 1 [0.74 0.04 0.67]
Out of plane standard deviation of the ring: 0.00 Å
Normal of ring 2 [0.75 0.03 0.66]
Out of plane standard deviation of the ring: 0.00 Å
The angle between ring 1 and ring 2 is 0.8°

Fig. 1: Example geometric and symmetry properties computed from DFT-optimized molecular geometries. These properties serve as prediction targets for the transformer model: molecular dimension (3D), ring count, chirality, number of symmetry planes, point group classification, planar fit error, and inter-ring dihedral angles

Table 2: Class distribution across 3,897,897 molecules. Imbalance ratios computed after binning; class weights capped at 20:1 during training.

Property	Classes	Min Samples	Max Samples	Imbalance Ratio
Dimension	5	175	3,531,980	20,183:1
Ring Count	7	1,320 (6+)	1,670,057 (1 ring)	1,265:1
Chirality	2	302,554	3,595,343	11.88:1
Symmetry Planes	4	1,384 (3+)	3,663,766 (0)	2,647:1
Point Group	15	398 (Other)	3,579,586 (C1)	8,993:1

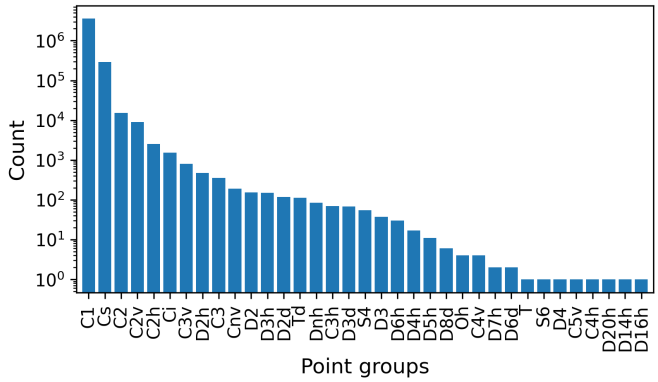


Fig. 2: Distribution of point group classifications across the dataset. The C_1 group (no symmetry) dominates with over 91% of molecules, while most other point groups are severely underrepresented. To address this extreme imbalance, rare groups were binned into an "Other" category for classification, though this necessarily limits the model's ability to distinguish between chemically distinct symmetry types

5 Model Architecture

5.1 Overview

All model architectures were implemented using PyTorch [33]. The architecture consists of a transformer encoder with property-specific prediction heads (Figure 3). Standard multi-task transformers attach all heads at the final layer; we additionally evaluate a hierarchical variant that attaches heads at different encoder depths, allowing properties to access features at varying levels of abstraction.

5.2 Encoder

The encoder stacks $N \in \{2, 4, 8\}$ transformer blocks [11], each containing multi-head self-attention and a feed-forward network with residual connections and layer normalization. We use $d_{\text{model}} = 512$, 8 attention heads ($d_k = 64$), feed-forward dimension 2048, and dropout 0.1. Input tokens are embedded via a learned embedding matrix and summed with learned positional encodings [34]:

$$\mathbf{X}_0 = \text{Embed}(\text{tokens}) + \text{PosEnc}(\text{positions}) \tag{1}$$

Each block applies two sublayers with residual connections and layer normalization:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{X}_{i-1}\mathbf{W}_Q, \mathbf{X}_{i-1}\mathbf{W}_K, \mathbf{X}_{i-1}\mathbf{W}_V \tag{2}$$

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{3}$$

$$\mathbf{X}'_i = \text{LayerNorm}(\mathbf{X}_{i-1} + \text{Dropout}(\text{MultiHead}(\mathbf{X}_{i-1}))) \tag{4}$$

$$\mathbf{X}_i = \text{LayerNorm}(\mathbf{X}'_i + \text{Dropout}(\text{FFN}(\mathbf{X}'_i))) \tag{5}$$

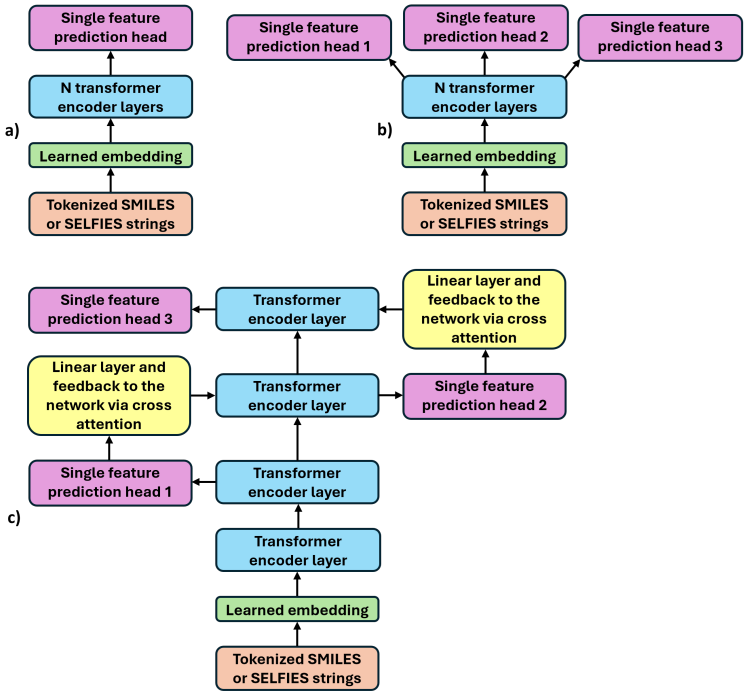


Fig. 3: Architecture configurations. (a) Single-task: one property per model. (b) Parallel multi-task: all heads at the final layer. (c) Hierarchical multi-task: heads at different depths with cross-attention feedback. The hierarchical variant uses 8 blocks (1 initial + 7 property-specific)

The feed-forward network applies two linear transformations with a ReLU activation:

$$\text{FFN}(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{6}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ with $d_{\text{ff}} = 2048$ (4× expansion). Padding masks prevent attention to pad tokens.

5.3 Prediction Heads

Classification heads apply a linear projection to class logits:

$$\mathbf{y}_{\text{logits}} = \mathbf{W}_{\text{out}}\mathbf{h}_0 + \mathbf{b}_{\text{out}} \tag{7}$$

where $\mathbf{h}_0$ is the encoder output at the first token position (used as the sequence representation), $\mathbf{W}_{\text{out}}$ is the output weight matrix, and $\mathbf{b}_{\text{out}}$ is the bias vector. These logits yield class probabilities after applying the softmax function:

$$\mathbf{P} = \text{softmax}(\mathbf{y}_{\text{logits}}) \tag{8}$$

Since the number of rings varies across molecules (up to 15 rings and 105 pairwise angles for molecules with 15 rings), the regression heads project to a fixed-size output and mask invalid positions when computing loss:

$$\mathcal{L}_{\text{seq}} = \frac{\sum_i m_i(\hat{y}_i - y_i)^2}{\sum_i m_i} \tag{9}$$

where $\hat{y}_i$ is the predicted value, y_i is the target value, and $m_i \in \{0, 1\}$ is the mask indicating valid positions.

5.4 Initialization

Weight initialization follows PyTorch defaults: the transformer encoder layers use Xavier uniform initialization [35] for their linear projections, token embeddings are initialized from $\mathcal{N}(0, 1)$, and learned positional encodings from $\mathcal{N}(0, 0.02)$. Prediction heads use Kaiming uniform initialization [36].

5.5 Hierarchical Cross-Attention

In the hierarchical architecture, each encoder block produces one property prediction, and subsequent blocks can attend to earlier predictions through cross-attention. The queries derive from the encoder state $\mathbf{X}_i$; keys and values derive from the embedded prediction at the previous block:

$$\mathbf{Q} = \mathbf{X}_i\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{E}_j\mathbf{W}_K, \quad \mathbf{V} = \mathbf{E}_j\mathbf{W}_V \tag{10}$$

$$\mathbf{X}'_i = \text{LayerNorm}(\mathbf{X}_i + \text{Dropout}(\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}))) \tag{11}$$

Classification predictions are embedded via a learned lookup table; regression predictions via linear projection. Each block attends only to the immediately preceding prediction, keeping memory linear in sequence length. Because keys and values derive from a single embedding vector rather than a sequence, the cross-attention adds negligible forward-pass cost. However, the additional gradient flow through prediction embeddings approximately doubled training time.

Cross-attention should help when properties have hierarchical dependencies—symmetry plane counts, for instance, directly inform point group classification. We evaluate whether this explicit inter-property communication improves over parallel architectures in Section 7.

6 Training

6.1 Setup

Models were trained on NVIDIA A100 40GB GPUs using PyTorch [33]. The data split is 80/10/10 for train/validation/test. Training uses bfloat16 (BF16) mixed precision [37] with batch size 256, AdamW optimizer [38], and a OneCycle learning rate schedule

[39]. The base learning rate is 10^{-4} for 2-block models, scaled by

$$(\text{batch_size}/64)/\sqrt{N_{\text{blocks}}} \tag{12}$$

for deeper models [40]. Epoch budgets were scaled with model depth (50, 75, and 100 epochs for 2-, 4-, and 8-block models) to accommodate slower convergence in deeper networks; early stopping with patience 10 terminated training when validation metrics plateaued. Training an 8-block multi-task model required approximately 25 GB VRAM and 50 GB RAM, with each epoch taking roughly 45 minutes on a single A100 GPU. The full dataset is preloaded into RAM for throughput. All experiments used a fixed random seed (42) for reproducibility; results represent single training runs.

6.2 Loss Functions

Classification tasks use focal loss [41] to handle class imbalance:

$$\text{FL}(p_t) = -\alpha_c(1 - p_t)^\gamma \log(p_t) \tag{13}$$

where p_t is the model's predicted probability for the true class c, $\gamma = 2.0$ is the focusing parameter that down-weights easy examples, and α_c is the class-specific weight. Weights are computed from training set frequencies:

$$\alpha_c = \frac{N}{n_c \cdot C} \tag{14}$$

where N is the total number of training samples, n_c is the number of samples in class c, and C is the number of classes. Weights are capped at a 20:1 ratio relative to the minimum weight to prevent excessive gradient imbalance from extremely rare classes.

Regression tasks use mean squared error. For planar fit error (a scalar target), standard MSE applies directly. Ring plane angles require special handling: molecules have variable numbers of ring pairs, and angles are periodic. We use a masked circular loss:

$$\mathcal{L}_{\text{angle}} = \frac{\sum_i m_i \cdot \delta_i^2}{\sum_i m_i}, \quad \delta_i = \text{atan2}(\sin(\hat{\theta}_i - \theta_i), \cos(\hat{\theta}_i - \theta_i)) \tag{15}$$

where $\hat{\theta}_i$ and θ_i are predicted and target angles, δ_i is the wrapped angular difference (handling periodicity), and $m_i \in \{0, 1\}$ masks invalid positions for molecules with fewer ring pairs.

Multi-task models sum property losses with task-specific weights calibrated empirically to produce approximately equal loss magnitudes across tasks. These weights were determined once and held constant across all multi-task experiments to ensure comparability.

Preliminary experiments confirmed focal loss improves macro F1 (see Eq. 17) by 1–8% over cross-entropy for imbalanced classification tasks, with the $\gamma = 2.0$ setting selected based on [41].

6.3 Evaluation Metrics

We report Matthews Correlation Coefficient (MCC) for binary classification (chirality), macro F1 for multi-class classification, and root mean square error (RMSE) for regression. To compare configurations across all seven tasks on a common scale we additionally define a combined score that maps each task to $[0, 1]$ with higher values indicating better performance.

For binary classification, MCC [42, 43] accounts for all four confusion matrix elements and remains informative under class imbalance:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{16}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives. MCC ranges from -1 (perfect inverse prediction) to $+1$ (perfect prediction), with 0 indicating random performance.

For multi-class classification, macro F1 averages the F1 score across all classes, weighting each class equally regardless of frequency:

$$\text{Macro F1} = \frac{1}{C} \sum_{c=1}^{C} \text{F1}_c, \quad \text{where} \quad \text{F1}_c = \frac{2 \cdot \text{precision}_c \cdot \text{recall}_c}{\text{precision}_c + \text{recall}_c} \tag{17}$$

Here C is the number of classes, and precision_c and recall_c are computed for class c in a one-vs-rest manner. This ensures performance on minority classes contributes equally to the metric.

To aggregate performance across all seven tasks into a single figure of merit, we define a combined score:

$$S_{\text{combined}} = \frac{1}{7} \sum_{i=1}^{7} s_i, \tag{18}$$

where each per-task score $s_i \in [0, 1]$ equals the macro F1 for multi-class classification, $(\text{MCC} + 1)/2$ for binary classification (chirality), or $1/(1 + \text{RMSE})$ for regression, mapping all tasks to a higher-is-better scale. The same $1/(1 + \text{RMSE})$ transformation is applied to regression axes in Figure 4 to give all seven properties a consistent higher-is-better orientation; raw RMSE values are reported in Table 3.

7 Results

Table 3 summarizes test set performance across all 50 configurations: 2 representations $\times$ (21 single-task + 3 parallel + 1 hierarchical) = 50 models. The table reports MCC for chirality, macro F1 for multi-class classification, and RMSE for regression. Three patterns emerge: SMILES outperforms SELFIES across nearly all properties, deeper models improve performance on complex properties without overfitting, and multi-task learning benefits classification but not regression. Section 7.5 revisits the primary representation comparison and the hierarchical architecture with targeted follow-up experiments using multiple random seeds and greater model depth.

Table 3: Comprehensive test set performance comparison across model depths, architectures, and representations. Primary metrics shown: MCC for chirality (binary), macro F1 for multiclass classification, RMSE for regression. Higher is better for MCC and F1; lower is better for RMSE. Best result per property in bold.

Property	Depth	SMILES			SELFIES		
		Single	Parallel	Hier.	Single	Parallel	Hier.
Classification: Chirality (MCC)							
Chirality	2	0.620	0.656	—	0.581	0.636	—
	4	0.652	0.747	—	0.676	0.709	—
	8	0.697	**0.782**	0.566	0.689	0.742	0.571
Classification: Dimension (Macro F1)							
Dimension	2	0.773	0.798	—	0.771	0.790	—
	4	0.772	0.836	—	0.806	0.806	—
	8	0.824	**0.882**	0.862	0.819	0.807	0.851
Classification: Ring Count (Macro F1)							
Ring Count	2	0.941	0.970	—	0.951	0.955	—
	4	0.850	**0.975**	—	0.965	0.964	—
	8	0.948	0.967	0.963	0.955	0.949	0.953
Classification: Symmetry Planes (Macro F1)							
Symmetry Planes	2	0.589	0.671	—	0.596	0.665	—
	4	0.584	0.756	—	0.626	0.726	—
	8	0.635	0.766	**0.769**	0.670	0.742	0.731
Classification: Point Group (Macro F1)							
Point Group	2	0.455	0.537	—	0.467	0.524	—
	4	0.499	0.602	—	0.522	0.565	—
	8	0.493	**0.615**	0.604	0.509	0.557	0.578
Regression: Planar Fit Error (RMSE, Å)							
Planar Fit	2	0.01319	0.01667	—	0.01479	0.01795	—
	4	0.01047	0.01287	—	0.01224	0.01484	—
	8	**0.00967**	0.01195	0.01261	0.01145	0.01446	0.01494
Regression: Ring Plane Angles (RMSE, rad)							
Ring Plane Angles	2	0.271	0.291	—	0.286	0.308	—
	4	**0.248**	0.256	—	0.272	0.279	—
	8	0.249	0.252	0.253	0.290	0.288	0.281

7.1 Representation Comparison: SMILES vs. SELFIES

SMILES outperforms SELFIES on all classification tasks (Figure 4). The gap varies by property: point group improves by 10%, dimension by 9%, symmetry planes by 3%, and ring count by 2%. Chirality shows a 5% improvement in MCC. Regression tasks follow the same pattern, with SMILES achieving 16% lower RMSE on planar fit (Table 3). Section 7.5 confirms these differences hold across multiple random seeds, with the ranking preserved in every replicate.

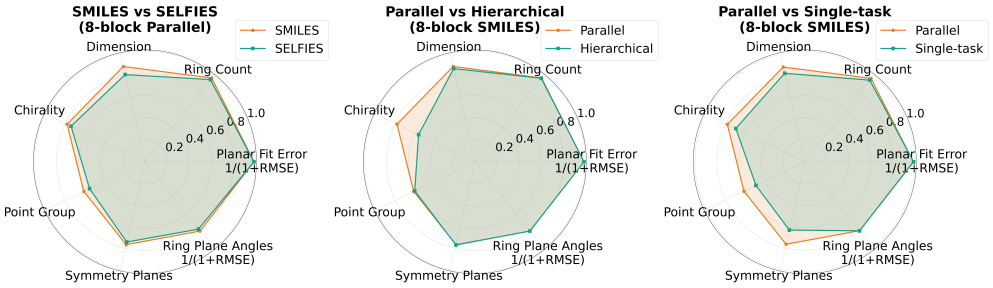


Fig. 4: Comparison of model configurations across seven molecular properties (8-block models). Left: SMILES vs. SELFIES (parallel multi-task). Center: parallel vs. hierarchical multi-task (SMILES). Right: parallel multi-task vs. single-task (SMILES). Higher values indicate better performance for all properties; regression tasks (planar fit error, ring plane angles) are displayed as $1/(1 + \text{RMSE})$ so that all axes share a higher-is-better orientation. Raw RMSE values are reported in Table 3.

7.2 Scaling with Model Depth

Performance scaling with depth depends on property complexity (Figure 5). Properties requiring global molecular reasoning benefit substantially from increased depth: dimension improves by 11%, chirality by 19%, and point group by 14% when moving from 2 to 8 encoder blocks (SMILES parallel). Point group gains are limited by severe class imbalance (8,993:1).

Ring count behaves differently, achieving near-peak performance at just 2 blocks with no improvement beyond. Ring counting requires only local pattern recognition—identifying cyclic substructures in the token sequence—which completes in early encoder layers.

The two regression tasks scale differently. Planar fit error improves by 27% from 2 to 8 blocks, benefiting from maximum depth. Ring plane angles saturates at 4 blocks with slight degradation at 8. This difference likely reflects task complexity: planar fit error aggregates information across all rings, while pairwise ring angles are more locally determined.

These patterns suggest that the dataset provides sufficient training signal for 8-block models on properties requiring global integration. The absence of overfitting on sufficiently complex tasks indicates that task complexity, not dataset size, determines optimal depth.

7.3 Multi-Task Learning

Multi-task learning affects classification and regression differently (Figure 5). Classification tasks benefit from shared representations: symmetry planes improve by 13.8% when moving from single-task to parallel multi-task at 2 blocks, and by 29.3% at 4 blocks. Chirality gains 5.8% in MCC at 2 blocks. These improvements suggest that classification properties share useful intermediate features that joint training helps extract.

Regression tasks show the opposite pattern. Planar fit error degrades by 26% and ring plane angles by 7% in multi-task mode at 2 blocks. The best regression results come from single-task models, indicating that the feature representations optimal for classification do not serve regression equally well. This asymmetry may reflect different abstraction requirements: classification benefits from categorical structural features, while regression needs continuous geometric information.

The hierarchical architecture with cross-attention shows minimal advantage over parallel multi-task. Only symmetry planes improves marginally; point group, chirality, and dimension all degrade. The sharp chirality degradation arises because the hierarchical design attaches the chirality head after only one shared encoder block, which is insufficient for this globally-dependent property. Cross-attention between property heads was intended to let later predictions condition on earlier ones, but shared encoder representations apparently capture these dependencies without explicit communication, and the additional complexity introduces instability. The hierarchical architecture was evaluated only at 8 blocks because it requires one attachment point per property (7 properties at blocks 1–7 plus a base block); fewer than 8 blocks would force multiple properties to share depths, negating the hierarchical design.

7.4 Error Analysis

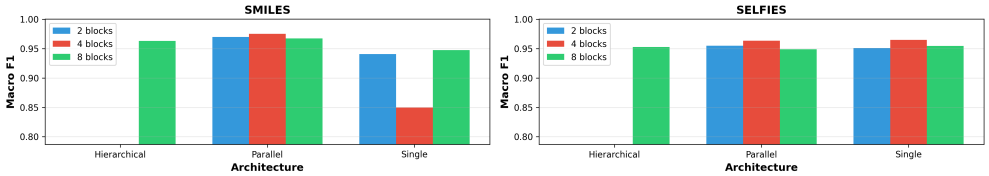
Confusion matrices (Figures 6 and 7) illustrate how task complexity affects prediction quality, comparing SMILES and SELFIES representations on the same test set.

Chirality prediction (Figure 6) achieves strong performance with both representations: SMILES correctly identifies 84% of achiral molecules and 98% of chiral molecules, while SELFIES achieves 80% and 97% respectively. The binary nature of this task and explicit stereochemical markers in SMILES make chirality relatively tractable. Errors concentrate on achiral molecules misclassified as chiral.

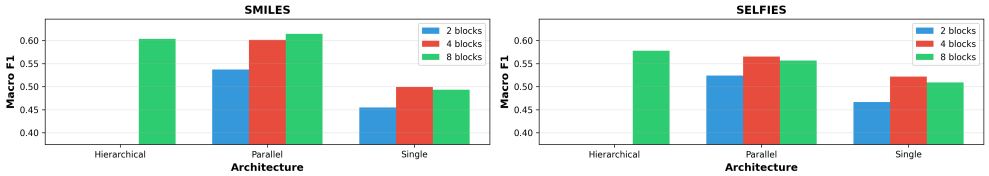
Point group classification (Figure 7) presents a harder challenge. With 15 classes and extreme imbalance (C_1 comprises 91.8% of the dataset), the model achieves high overall accuracy but low macro F1 due to poor performance on minority classes. The confusion matrices reveal systematic patterns: underrepresented symmetry groups are frequently misclassified as C_1 (no symmetry) or C_s (single mirror plane), as expected given the class imbalance. Several point groups have fewer than 100 training examples (Figure 2), leaving the model with insufficient data to correctly identify them. SMILES modestly outperforms SELFIES across most point groups, consistent with the overall representation comparison.

7.5 Representation Robustness and Deeper Hierarchy

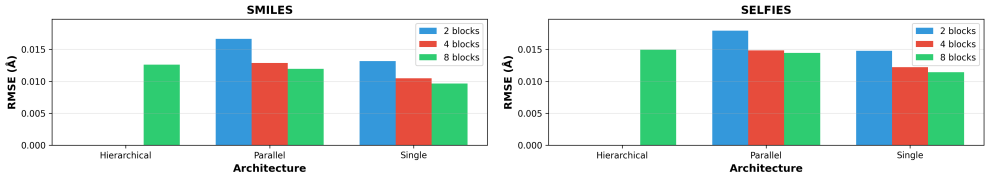
The 50-configuration grid trained each model with a fixed random seed, so apparent differences between configurations could partly reflect sensitivity to weight initialization and data ordering rather than true systematic effects. Two follow-up experiments addressed this directly. Rather than fixed epoch budgets, these runs used convergence-based early stopping: training halted when the combined validation score failed to improve by $\geq 10^{-4}$ over 20 consecutive epochs (plateau patience), or when validation



(a) Ring count



(b) Point group



(c) Planar fit error

Fig. 5: (a) Ring count performance saturates at two encoder blocks. (b) Point group and other non-trivial classification tasks benefit from deeper models. (c) Regression tasks, evaluated via error minimization, do not benefit from task parallelism

degraded while training loss continued to improve for 10 consecutive epochs (overfitting patience). A hard cap of 500 epochs was set but never reached in any run. All other hyperparameters matched the exploratory grid.

To compare configurations with a single figure of merit, we use the combined score defined in Eq. 18.

7.5.1 SMILES vs. SELFIES with Three Random Seeds

Three independent seeds were run for the 8-block parallel architecture on both representations. Table 4 reports per-property metrics averaged over seeds, with standard deviations.

SMILES achieves a combined score of 0.851 ± 0.005 versus 0.838 ± 0.002 for SELFIES. The seed ranges do not overlap, confirming that the advantage is not attributable to random initialization. Property-level gaps are largest for dimension (+4.6% F1) and smallest for ring count (+0.6%), preserving the ordering seen in the exploratory

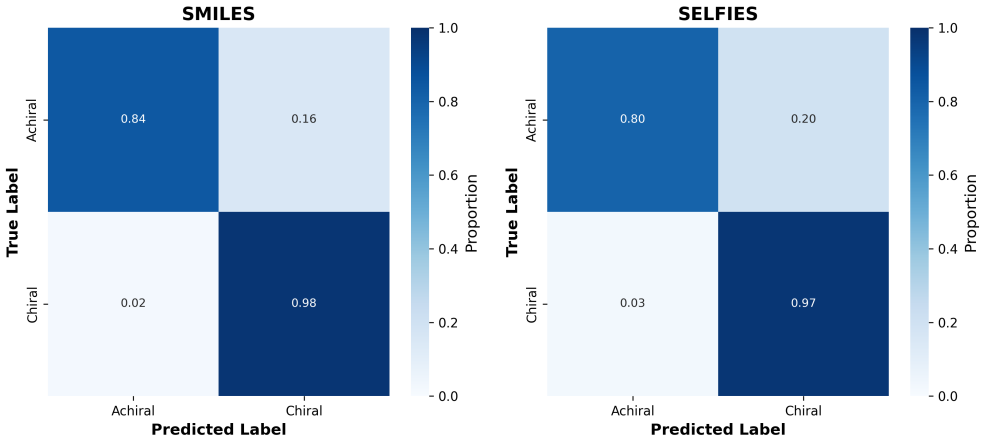


Fig. 6: Confusion matrices for chirality prediction comparing SMILES (left) and SELFIES (right) representations (8-block parallel model). Values show row-normalized proportions. Both representations achieve high recall on the majority chiral class (98–97%) with SMILES showing slightly better recall on the minority achiral class (84% vs. 80%)

experiments. The magnitudes are smaller than some single-seed estimates from the exploratory grid; the multi-seed comparison provides more reliable effect sizes.

SMILES models converged after 106–122 epochs (mean 112), while SELFIES models stopped earlier at 87–96 epochs (mean 91). The faster convergence of SELFIES likely reflects its shorter, more regular token sequences, which may produce a smoother loss landscape. Despite this training efficiency, the resulting models are uniformly weaker.

7.5.2 10-Block Hierarchical Architecture

The original 8-block hierarchical model attached properties at encoder blocks 1–7, leaving no shared encoding base before the first prediction head. The 10-block variant shifts all attachment points three layers forward: chirality at block 3, ring count at block 4, planar fit error at block 5, ring plane angles at block 6, dimension at block 7, symmetry planes at block 8, and point group at block 9. Blocks 0–2 form a shared encoding base that all prediction heads process before any property-specific branch connects.

One seed per representation was run. SMILES scores 0.843 versus the 8-block parallel mean of 0.851; SELFIES scores 0.837 versus 0.838. The deeper shared base provides no aggregate benefit over the parallel architecture. Its most informative result concerns chirality: MCC rises from 0.566 in the 8-block hierarchical (Table 3) to 0.723 here, because three shared encoder layers—rather than one—precede the chirality head. This confirms that the sharp chirality degradation in the original hierarchical design was an artifact of insufficient pre-head encoding depth. The parallel architecture nonetheless remains superior for chirality and matches or exceeds the 10-block

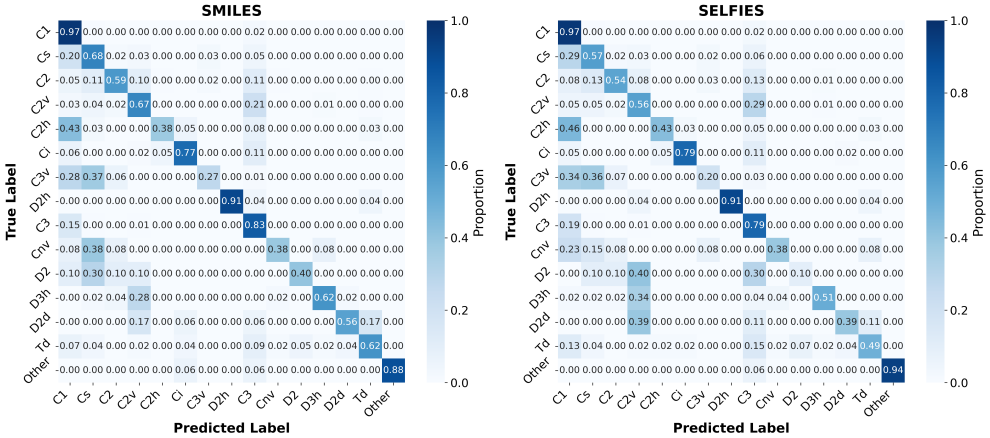


Fig. 7: Confusion matrices for point group classification comparing SMILES (left) and SELFIES (right) representations (8-block parallel model). Values show row-normalized proportions. The dominant C_1 class achieves high recall (97%) but underrepresented symmetry groups show poor recall, frequently misclassified as C_1 or C_s

Table 4: Test set performance for targeted follow-up experiments. Mean ± standard deviation over three random seeds for the 8-block parallel architecture (SMILES: epochs 106, 122, 107; SELFIES: 91, 96, 87); single runs for the 10-block hierarchical (SMILES: 111 epochs, SELFIES: 138 epochs). Primary metrics: MCC for chirality, macro F1 for classification, RMSE for regression. Combined score defined by Eq. 18.

Property	8-block parallel		10-block hierarchical	
	SMILES ($n = 3$)	SELFIES ($n = 3$)	SMILES	SELFIES
Classification (Macro F1, except MCC for chirality)				
Dimension	0.870 ± 0.013	0.832 ± 0.007	0.852	0.869
Ring count	0.971 ± 0.004	0.966 ± 0.004	0.975	0.966
Chirality (MCC)	0.787 ± 0.005	0.762 ± 0.005	0.723	0.706
Symm. planes	0.786 ± 0.014	0.772 ± 0.005	0.778	0.781
Point group	0.645 ± 0.011	0.631 ± 0.012	0.644	0.613
Regression (RMSE ↓)				
Planar fit error (Å)	0.0116 ± 0.0004	0.0128 ± 0.0003	0.0120	0.0132
Ring plane angles (rad)	0.243 ± 0.004	0.259 ± 0.003	0.245	0.265
Combined score	$\mathbf{0.851 \pm 0.005}$	$\mathbf{0.838 \pm 0.002}$	**0.843**	**0.837**

hierarchical on every other property. Explicit ordering of prediction heads provides no substitute for the shared representations that parallel multi-task training develops.

8 Discussion

8.1 Why SMILES Outperforms SELFIES

The SMILES advantage appears to stem from how each representation encodes chemical information. SMILES tokens map directly to atoms, bonds, and stereochemistry markers (`@`, `@@`); ring closures use numeric labels that preserve topological relationships. SELFIES tokens encode grammar operations rather than chemical concepts, potentially diluting the structural signal. This difference matters less for tasks where any consistent encoding suffices, but more for tasks requiring fine-grained structural discrimination.

The validity guarantee that motivates SELFIES provides no benefit for property prediction, where inputs are already valid molecules. For generative applications involving string manipulation, validity guarantees remain valuable—practitioners should choose representations based on their use case.

8.2 Scaling Behavior and Feature Hierarchy

Large Language Model (LLM) scaling laws suggest that compute-optimal training requires roughly 20 tokens per parameter [9]. With an 80/10/10 train/validation/test split, our training set of 3.12 million molecules yields approximately 78 million training tokens for SMILES (24.9 tokens per molecule) or 67 million for SELFIES (21.5 tokens per molecule). The 2-block model (7.7M parameters) operates at 10.1 tokens per parameter for SMILES—half the optimal ratio. The 4-block model (16.1M parameters) drops to 4.8 tokens per parameter (4× below optimal), while the 8-block model (32.9M parameters) reaches only 2.4 tokens per parameter, over 8× below the Chinchilla-optimal ratio.

This data constraint shapes the observed performance patterns. Properties that can be extracted from local token patterns—like ring count, where structures such as `C1CCCCC1` are directly visible—saturate at 2 blocks because even a data-limited shallow model captures these features. Properties requiring global integration across the sequence, such as dimension and point group, continue improving with bigger model size and more computational power used despite the unfavorable tokens-per-parameter ratio. The additional depth helps, but gains are modest compared to what might be achieved with sufficient data.

This analysis suggests practitioners should match model depth to both task complexity and data availability. For locally-extractable features, shallow models suffice regardless of dataset size. For complex properties requiring compositional reasoning, deeper models help but their potential might be constrained by data.

8.3 Multi-Task Trade-offs

The divergence between classification and regression tasks under multi-task learning suggests these task types rely on different feature representations. Classification

benefits from categorical structural features that transfer well across related tasks. Regression requires continuous geometric information that may conflict with the discrete boundaries classification prefers. Joint training forces a compromise that benefits classification at regression's expense.

The failure of hierarchical cross-attention to improve over parallel multi-task is informative. We hypothesized that later predictions (e.g., point group) could benefit from conditioning on earlier ones (e.g., symmetry planes). Shared encoder representations appear to capture these dependencies implicitly, making explicit inter-head communication redundant.

8.4 Future Directions

Self-supervised pre-training on large unlabeled molecular databases [44, 45] could improve generalization, particularly for rare scaffolds underrepresented in labeled data. Combining string encodings with graph neural networks [17] might capture both sequential and topological information. Attention visualization could reveal which substructures drive predictions, providing chemical interpretability beyond accuracy metrics.

8.5 Limitations

String representations encode molecular connectivity, not 3D geometry. Properties that depend on precise bond angles or conformational flexibility cannot be fully recovered from topology alone—the boundary errors in dimension classification illustrate this limitation. Furthermore, SMILES and SELFIES each encode a single molecular graph without specifying a conformation; for molecules with multiple accessible conformers, property labels were derived from a single DFT-optimized geometry. This is most consequential for symmetry properties, where conformational variation can alter the point group entirely. Three-dimensional methods [22, 23] can achieve higher accuracy on geometric tasks by using explicit coordinates, at the cost of requiring conformer generation.

The dataset, while large by quantum chemistry standards, covers a small fraction of drug-like chemical space [1]. Performance may degrade on molecules dissimilar to the training distribution. Point group prediction remains the most challenging task because rare symmetry groups have few training examples—groups with fewer than 100 samples were combined into an "Other" category, and class weighting alone cannot fully compensate for this imbalance.

The 50-configuration exploratory grid and both hierarchical models were run with a single random seed, so differences within that grid cannot be tested for statistical significance. The primary representation comparison (8-block parallel) was replicated with three independent seeds (Section 7.5), confirming that SMILES outperforms SELFIES with non-overlapping seed ranges (0.851 ± 0.005 vs. 0.838 ± 0.002 combined score). Other configuration differences in the exploratory grid remain single-seed estimates. We did not compare against pre-trained molecular language models such as ChemBERTa [5] or MolBERT [13] as they were trained on entirely different features of molecules like bioavailability or toxicity.

9 Conclusions

This study compared SMILES and SELFIES representations across 50 model configurations for seven molecular properties, with targeted follow-up experiments confirming the key findings across multiple random seeds. Three findings have practical implications.

SMILES outperforms SELFIES on all seven properties. A three-seed replication of the primary comparison quantifies the gap as 0.851 ± 0.005 versus 0.838 ± 0.002 in combined score, with property-level differences ranging from 0.6% (ring count) to 4.6% F1 (dimension). The advantage stems from SMILES' direct encoding of chemical structure rather than grammar operations. For property prediction, SMILES is the better choice; SELFIES remains preferable for generative applications where validity guarantees matter.

Model depth should match task complexity. Properties requiring global molecular reasoning—dimension, chirality, point group—benefit from 8 encoder blocks. Local structural features like ring count saturate at 2 blocks. The dataset of 3.9 million molecules supports deeper models without overfitting on complex tasks, indicating that task difficulty, alongside dataset size, sets its own limitations on the optimal model size.

Multi-task learning improves classification (6–29% relative gains depending on property) but degrades regression. Classification properties share useful intermediate features; regression requires different representations that joint training compromises. The hierarchical architecture with cross-attention offers no improvement over simpler parallel multi-task designs. A deeper 10-block variant with a three-layer shared encoding base confirms this finding while revealing that the chirality degradation in the original design was an artifact of insufficient pre-head depth, not a fundamental incompatibility.

The best configurations achieve high accuracy across all properties (Table 3). String-based inference enables high-throughput screening at scales impractical for methods requiring explicit 3D structure generation. These results demonstrate that string-based transformers can predict diverse molecular properties from 1D representations when architectural choices are matched to task requirements.

Acknowledgements. We gratefully acknowledge CINECA (Consorzio Interuniversitario per il Calcolo Automatico dell'Italia Nord-Orientale) for providing access to the Leonardo HPC cluster under project EUHPC_D21_046. All training was conducted on Leonardo's GPU partition equipped with NVIDIA A100 40GB GPUs, which enabled the large-scale experiments presented in this work.

We thank the developers of the Molecule3D dataset [27] and the PubChemQC database [28] for providing high-quality DFT-optimized molecular geometries and properties that formed the foundation of this research. We acknowledge the computational chemistry community for maintaining essential open-source libraries: RDKit for molecular informatics [29], the SELFIES library [4] for robust molecular representations, and the SOFI library [30] for symmetry analysis.

This work was implemented using PyTorch [33]. We thank the PyTorch team and the broader deep learning community for developing and maintaining the frameworks that enabled this research.

Statements and Declarations

Funding This work used HPC resources of CINECA under project EUHPC_D21_046. No external funding was received.

Competing interests The authors declare no competing interests.

Author contributions R.G. conceived the study, designed and implemented the model architecture and training pipeline, conducted all experiments, performed data analysis, and drafted the manuscript. S.T. contributed to the research direction, discussed results, and critically revised the manuscript. Both authors approved the final version.

Data availability The curated dataset used in this work is available at https://drive.google.com/drive/folders/194NMd0PC7auKNxHdCTW2ddZ0h08lNzf6?usp=drive_link. This dataset is a fork of the Molecule3D benchmark [27], originally licensed under GPL-3.0. The dataset comprises approximately 3.9 million molecular structures organized into 8 HDF5 files (4 structure files, 4 feature files) containing SMILES and SELFIES representations, 3D atomic coordinates, and computed molecular properties. All data processing scripts, model implementation, training code, and experimental results are available on GitHub at https://github.com/Nr-1000101/hidra.

References

[1] Bohacek, R.S., McMartin, C., Guida, W.C.: The art and practice of structure-based drug design: a molecular modeling perspective. Medicinal Research Reviews **16**(1), 3–50 (1996) https://doi.org/10.1002/(SICI)1098-1128(199601)16:1⟨3::AID-MED1⟩3.0.CO;2-6

[2] Becke, A.D.: Density-functional thermochemistry. iii. the role of exact exchange. The Journal of Chemical Physics **98**(7), 5648–5652 (1993) https://doi.org/10.1063/1.464913 . B3LYP functional

[3] Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences **28**(1), 31–36 (1988) https://doi.org/10.1021/ci00057a005

[4] Krenn, M., Häse, F., Nigam, A., Friederich, P., Aspuru-Guzik, A.: Self-referencing embedded strings (selfies): A 100% robust molecular string representation. Machine Learning: Science and Technology **1**(4), 045024 (2020) https://doi.org/10.1088/2632-2153/aba947

[5] Chithrananda, S., Grand, G., Ramsundar, B.: Chemberta: Large-scale self-supervised pretraining for molecular property prediction. arXiv preprint arXiv:2010.09885 (2020)

[6] Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C.A., Bekas, C., Lee, A.A.: Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. ACS Central Science **5**(9), 1572–1583 (2019) https://doi.org/10.1021/acscentsci.9b00576

[7] Ross, J., Belgodere, B., Chenthamarakshan, V., Padhi, I., Mroueh, Y., Das, P.: Large-scale chemical language representations capture molecular structure and properties. Nature Machine Intelligence **4**(12), 1256–1264 (2022) https://doi.org/10.1038/s42256-022-00580-7

[8] Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)

[9] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D.d.L., Hendricks, L.A., Welbl, J., Clark, A., *et al.*: Training compute-optimal large language models. In: Advances in Neural Information Processing Systems, vol. 35, pp. 30016–30030 (2022). Chinchilla scaling laws. https://arxiv.org/abs/2203.15556

[10] Rogers, D., Hahn, M.: Extended-connectivity fingerprints. Journal of Chemical Information and Modeling **50**(5), 742–754 (2010) https://doi.org/10.1021/ci100050t . ECFP/Morgan fingerprints

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[12] Ahmad, W., Simon, E., Chithrananda, S., Grand, G., Ramsundar, B.: Chemberta-2: Towards chemical foundation models. arXiv preprint arXiv:2209.01712 (2022)

[13] Fabian, B., Edlich, T., Gaspar, H., Segler, M., Meyers, J., Fiscato, M., Ahmed, M.: Molecular representation learning with language models and domain-relevant auxiliary tasks. arXiv preprint arXiv:2011.13230 (2020). MolBERT

[14] Maziarka, Ł., Danel, T., Mucha, S., Rataj, K., Tabor, J., Jastrzebski, S.: Molecule attention transformer. arXiv preprint arXiv:2002.08264 (2020). MAT - Molecular Attention Transformer

[15] Caruana, R.: Multitask learning. Machine Learning **28**(1), 41–75 (1997) https://doi.org/10.1023/A:1007379606734

[16] Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)

[17] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning, pp. 1263–1272 (2017). PMLR. MPNN - Message Passing Neural Networks

[18] Xiong, Z., Wang, D., Liu, X., Zhong, F., Wan, X., Li, X., Li, Z., Luo, X., Chen, K., Jiang, H., Zheng, M.: Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. Journal of Medicinal Chemistry **63**(16), 8749–8760 (2019) https://doi.org/10.1021/acs.jmedchem.9b00959 . AttentiveFP

[19] Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., *et al.*: Analyzing learned molecular representations for property prediction. Journal of Chemical Information and Modeling **59**(8), 3370–3388 (2019) https://doi.org/10.1021/acs.jcim.9b00237 . Chemprop

[20] Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.: Moleculenet: a benchmark for molecular machine learning. Chemical Science **9**(2), 513–530 (2018) https://doi.org/10.1039/C7SC02664A

[21] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. Advances in Neural Information Processing Systems **33**, 22118–22133 (2020)

[22] Schütt, K., Kindermans, P.-J., Sauceda Felix, H.E., Chmiela, S., Tkatchenko, A., Müller, K.-R.: Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In: Advances in Neural Information Processing Systems, pp. 991–1001 (2017). https://proceedings.neurips.cc/paper/2017/hash/303ed4c69846ab36c2904d3ba8573050-Abstract.html

[23] Klicpera, J., Groß, J., Günnemann, S.: Directional message passing for molecular graphs. In: International Conference on Learning Representations (2020). DimeNet. https://openreview.net/forum?id=B1eWbxStPH

[24] Schütt, K.T., Unke, O.T., Gastegger, M.: Equivariant message passing for the prediction of tensorial properties and molecular spectra. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 9377–9388. PMLR, Virtual (2021). PaiNN. https://proceedings.mlr.press/v139/schutt21a.html

[25] Liu, Y., Wang, L., Liu, M., Lin, Y., Zhang, X., Oztekin, B., Ji, S.: Spherical message passing for 3d molecular graphs. In: International Conference on Learning Representations (2022). SphereNet. https://openreview.net/forum?id=givsRXsOt9r

[26] Riniker, S., Landrum, G.A.: Better informed distance geometry: Using what we know to improve conformation generation. Journal of Chemical Information and Modeling **55**(12), 2562–2574 (2015) https://doi.org/10.1021/acs.jcim.5b00654 . ETKDG conformer generation

[27] Xu, Z., Luo, Y., Zhang, X., Xu, X., Xie, Y., Liu, M., Dickerson, K., Deng, C., Nakata, M., Ji, S.: Molecule3d: A benchmark for predicting 3d geometries from molecular graphs. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2021). https://openreview.net/forum?id=WnaHbA4jCN

[28] Nakata, M., Shimazaki, T.: Pubchemqc project: A large-scale first-principles electronic structure database for data-driven chemistry. Journal of Chemical Information and Modeling **57**(6), 1300–1308 (2017) https://doi.org/10.1021/acs.jcim.7b00083

[29] RDKit: RDKit: Open-source cheminformatics. http://www.rdkit.org. Accessed: 2025-12-16

[30] Gunde, M., Salles, N., Grisanti, L., Martin-Samos, L., Hemeryck, A.: Sofi: Finding point group symmetries in atomic clusters as finding the set of degenerate solutions in a shape-matching problem. The Journal of Chemical Physics **161**(6), 062503 (2024) https://doi.org/10.1063/5.0215689

[31] Blaschke, G., Kraft, H., Fickentscher, K., Köhler, F.: Chromatographische racemattrennung von thalidomid und teratogene wirkung der enantiomere. Arzneimittel-Forschung **29**(10), 1640–1642 (1979). First demonstration that (S)-thalidomide enantiomer is teratogenic

[32] Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018) https://doi.org/10.1016/j.neunet.2018.07.011

[33] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., *et al.*: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, pp. 8026–8037 (2019). https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[34] Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1243–1252. PMLR, Sydney, Australia (2017). Learned positional encodings. https://proceedings.mlr.press/v70/gehring17a.html

[35] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward

neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 9, pp. 249–256. PMLR, Sardinia, Italy (2010). Xavier/Glorot initialization. http://proceedings.mlr.press/v9/glorot10a.html

[36] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015). https://doi.org/10.1109/ICCV.2015.123

[37] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., Wu, H.: Mixed precision training. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=r1gs9JgRZ

[38] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019). AdamW optimizer. https://openreview.net/forum?id=Bkg6RiCqY7

[39] Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, vol. 11006, pp. 369–386 (2019). https://doi.org/10.1117/12.2520589 . SPIE. OneCycle learning rate scheduler

[40] Smith, L.N.: A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820 (2018)

[41] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.324

[42] Matthews, B.W.: Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure **405**(2), 442–451 (1975) https://doi.org/10.1016/0005-2795(75)90109-9

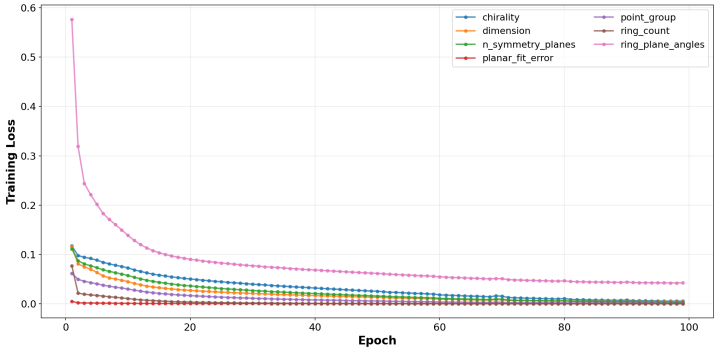
[43] Chicco, D., Jurman, G.: The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC Genomics **21**(1), 1–13 (2020) https://doi.org/10.1186/s12864-019-6413-7

[44] Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., Leskovec, J.: Strategies for pre-training graph neural networks. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=HJlWWJSFDH

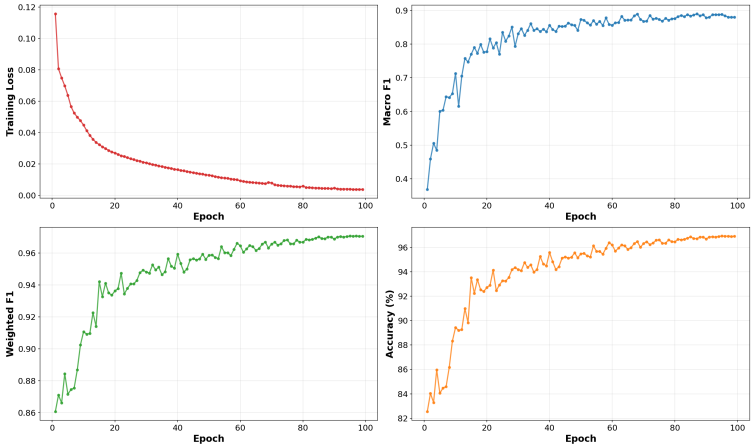
[45] Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., Huang, J.: Self-supervised graph transformer on large-scale molecular data. Advances in Neural

Appendix A Training Dynamics

Figures A1 and A2 provide supplementary information about model training behavior.



(a) Per-property training losses



(b) Validation metrics for dimension

Fig. A1: Training dynamics for multi-task model (parallel architecture, SMILES, 8 encoder blocks, 100 epochs). (a) Per-property training losses weighted to contribute similar magnitude gradients during backpropagation; all properties converge smoothly without overfitting. (b) Validation metrics for dimension classification showing loss, macro F1 (primary metric), weighted F1, and accuracy improving steadily throughout training

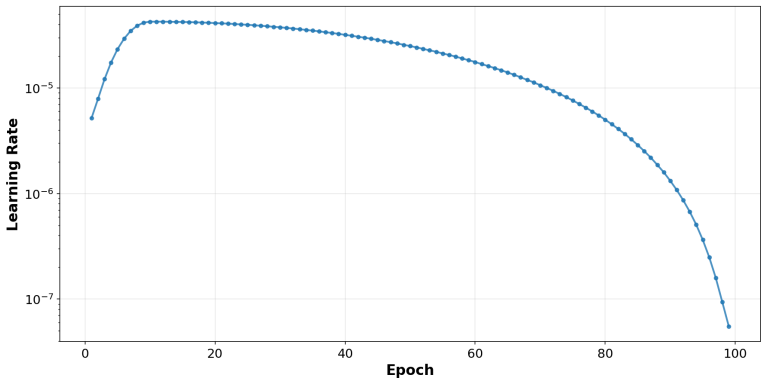


Fig. A2: OneCycle learning rate schedule with warmup. Learning rate increases linearly to peak over 10% of training, then decays cosine-annealed over the remaining epochs

Appendix B Hyperparameter Details

Table B1 summarizes all hyperparameters used across experiments.

B.1 Architecture Modes

- **Single-task**: One property per model, head at last encoder block
- **Parallel multi-task**: All 7 properties, all heads at last block, no cross-attention
- **Hierarchical multi-task**: All 7 properties, heads at different blocks (chirality:1, ring_count:2, planar_fit_error:3, ring_plane_angles:4, dimension:5, n_symmetry_planes:6, point_group:7), with cross-attention enabled

Appendix C Property Computation Details

This appendix describes the algorithms used to compute molecular properties from DFT-optimized 3D geometries. Properties were pre-computed using RDKit [29] for graph analysis and SOFI [30] for symmetry detection, then stored in HDF5 files.

C.1 Ring Count

Ring count uses RDKit's Smallest Set of Smallest Rings (SSSR) algorithm. The count is binned to $\min(n_{\text{rings}}, 6)$, mapping molecules with ≥ 6 rings to class 6.

C.2 Molecular Dimensionality

Dimensionality classification applies PCA to atomic coordinates. For N atoms with centered coordinates $\mathbf{X}' \in \mathbb{R}^{N\times 3}$, we compute the covariance matrix eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and their ratios $r_i = \lambda_i / \sum_j \lambda_j$.

Classification uses adaptive thresholds that scale with molecule size: $\tau_{\text{strict}} = 1 - 10^{-3}/\sqrt{N}$ and $\tau_{\text{relaxed}} = 1 - 10^{-1}/\sqrt{N}$. A molecule is classified as: **linear** if $r_1 >$

Table B1: Comprehensive hyperparameter configuration for all experiments

Category	Parameter	Value(s)
Model Architecture		
	Model dimension (d_{model})	512
	Attention heads	8 ($d_k = d_v = 64$)
	Feed-forward dimension	2048 (4× expansion)
	Encoder blocks	2, 4, or 8 (varied by experiment)
	Dropout rate	0.1
	Maximum sequence length	512 tokens
	Positional encoding	Learned embeddings
Training Configuration		
	Batch size	256 (optimized for A100 40GB)
	Precision	BF16 mixed precision
	Epochs	50 (2-block), 75 (4-block), 100 (8-block)
	Hardware	NVIDIA A100 40GB GPU
	Data split	80% train / 10% val / 10% test
	Data loading	4 workers, pin_memory=True, persistent_workers=True
	RAM preloading	Full dataset (˜1GB RAM footprint; 3.4GB on disk)
Optimization		
	Optimizer	AdamW [38]
	Base learning rate	1×10^{-4} (for 2-block model)
	LR scaling	$(base_lr \times batch_size/64)/\sqrt{n_blocks}$
	Weight decay	0.01 (AdamW default)
	Gradient clipping	None (BF16 provides stability)
Learning Rate Scheduler		
	Scheduler type	OneCycleLR [39]
	Warmup percentage	10% of total epochs
	Annealing strategy	Cosine
	Initial divisor	10.0 (start_lr = max_lr / 10)
	Final divisor	100.0 (end_lr = max_lr / 100)
Loss Functions		
	Classification loss	Focal Loss [41]
	Focal γ	2.0 (focus on hard examples)
	Focal α	Auto (inverse class frequency, capped at 20:1)
	Regression loss	Mean Squared Error (MSE)
	Multi-task weighting	Balanced (w_p chosen for equal loss contribution)
Early Stopping		
	Patience	10 epochs (no improvement threshold)
	Min delta	0.0 (any improvement counts)
	Metric	MCC (binary), Macro F1 (multiclass), RMSE (regression)

τ_{strict}; **almost linear** if $r_1 > \tau_{\text{relaxed}}$; **planar** if $r_1 + r_2 > \tau_{\text{strict}}$; **almost planar** if $r_1 + r_2 > \tau_{\text{relaxed}}$; **3D** otherwise.

C.3 Ring Geometry

C.3.1 Planar Fit Error

For each ring, we fit a plane via PCA on ring atom coordinates. The planar fit error is $\sqrt{\lambda_3}$ where λ_3 is the smallest eigenvalue (variance perpendicular to the best-fit plane). Output is the mean error across all rings.

C.3.2 Ring Plane Angles

For each pair of rings, we compute the dihedral angle between their best-fit planes: $\theta_{ij} = \arccos(|\mathbf{n}_i \cdot \mathbf{n}_j|)$ where $\mathbf{n}_i, \mathbf{n}_j$ are plane normal vectors. The absolute value ensures $\theta \in [0°, 90°]$. Output is a variable-length sequence of pairwise angles, padded to 105 elements (maximum for 15 rings).

C.4 Symmetry Properties

C.4.1 Point Group

The SOFI library [30] detects molecular symmetry operations by testing candidate rotations, reflections, and inversions. A transformation is accepted if it maps the molecule onto itself within an adaptive tolerance $\epsilon = 0.015 \times N$ Å, where N is the number of atoms. Linear molecules are post-processed to assign C_{nv} or D_{nh} groups. Output is binned to the 14 most common point groups plus "Other."

C.4.2 Symmetry Plane Count

We count reflection operations (σ) from the SOFI output, excluding trivial cases: for planar molecules, the molecular plane is not counted; for linear molecules, axial planes are excluded. The count is binned into four classes: 0, 1, 2, and ≥ 3 planes.