CS 273A Final Project - Group 23

Edward Finkelstein
efinkel1@uci.edu
Tyler Collins
tcollin1@uci.edu

December 2025

Abstract

This project uses machine-learning methods to predict molecular atomization energy, a quantum-mechanical measure of bond strength and molecular stability. Using the QM9 dataset from Kaggle, each molecule's SMILES string is converted into Morgan fingerprints using RDKit. Multiple regression models, such as Random Forests, Support Vector Regression, a feed-forward neural network, and symbolic regression via `PySR`, are trained and compared using the same fingerprint representation to determine which algorithm best captures structure–property relationships. Model performance is evaluated through RMSE, MAE, and parity plots. After identifying the best model, we also compare fingerprint-based and descriptor-based feature sets. We expect nonlinear models to outperform linear ones, with fingerprint features providing higher accuracy than global descriptors.

1 Introduction

Machine learning provides an efficient alternative to quantum-chemical calculations for predicting molecular properties such as atomization energy, a key measure of molecular stability. In this project, we use the QM9 dataset and convert each molecule's SMILES string into Morgan fingerprints, yielding a high-dimensional structural representation suitable for regression. We evaluate several learning methods; Random Forests, Support Vector Regression, feed-forward neural networks, and symbolic regression, to compare their ability to model structure–property relationships. Our goal is to determine which approaches and feature representations most effectively predict atomization energy.

2 Methods

2.1 Data Preparation

The QM9 dataset was obtained using the PyTorch Geometric QM9 loader with RDKit enabled, allowing extraction of canonical SMILES strings and all quantum-mechanical target properties. For each molecule, the `U0` atomization energy was selected as the prediction target. A base table containing only SMILES and `U0` was constructed to ensure a consistent starting point for all feature representations.

Two molecular representations were generated from the SMILES strings. First, Morgan fingerprints (ECFP; radius = 2, 2048 bits) were computed using RDKit to capture local substructural patterns and functional groups. Second, RDKit 2D descriptors were calculated to provide a global, interpretable feature set. Descriptor columns that were all-NaN or constant were removed, and remaining missing values were median-imputed.

Both feature tables were checked to ensure identical molecule sets, SMILES strings, and `U0` values, guaranteeing a fair comparison between representations. We partitioned the dataset into an 80/10/10 train/validation/test split to maximize training data while reserving held-out subsets for hyperparameter tuning and final model evaluation. These datasets were then used to train and evaluate several regression models, described in the following sections.

2.2 Random Forest Regression

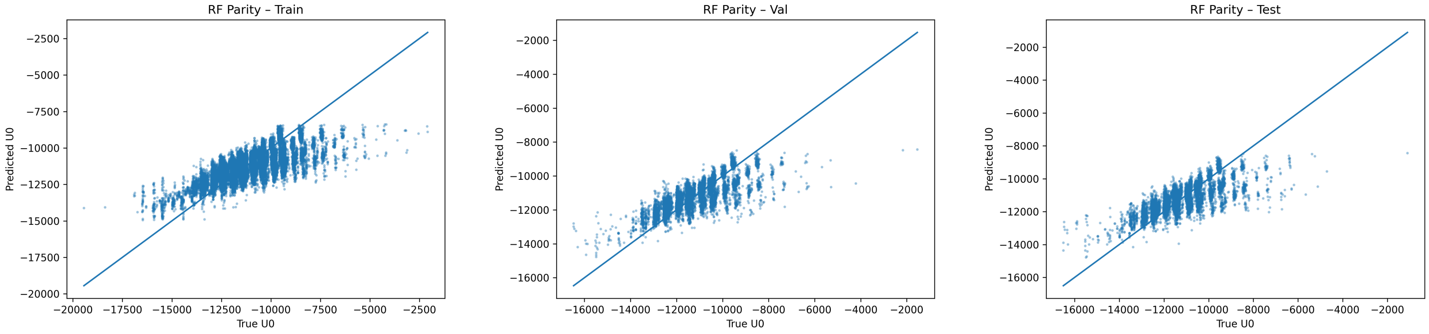
We first trained a baseline Random Forest (RF) model on the Morgan-fingerprint representation using default hyperparameters. This baseline exhibited heavy overfitting, with training (RMSE, MAE) $\approx$ (251, 176) but validation (RMSE, MAE) $\approx$ (664, 470) and test (RMSE, MAE) $\approx$ (678, 478), suggesting that the unconstrained bagged trees were simply memorizing local structures.

To control model variance, we performed a coarse hyperparameter search over tree depth, leaf size, and feature subsampling using `RandomizedSearchCV` with `n_iter = 10` sampled parameter settings from the grid shown in Table 1. Because the full dataset contains over 10^5 molecules, we used only 10% of the training set for hyperparameter tuning to reduce runtime while still preserving relative rankings of parameter settings. The best configuration found used increased leaf sample-sizes and restricted feature sampling. This model generalized better as it achieved nearly identical error across train, validation, and test splits (Table 2), indicating that constraining tree growth mitigates the previously observed overfitting.

Hyperparameter	**Search Space**
Number of trees (`n_estimators`)	{100, 200}
Maximum tree depth (`max_depth`)	{None, 20, 40}
Minimum samples per leaf (`min_samples_leaf`)	{1, 2, 4}
Minimum samples to split (`min_samples_split`)	{2, 5, 10}
Feature subsampling (`max_features`)	{`"sqrt"`, 0.25}

Table 1: Hyperparameter search space used in `RandomizedSearchCV` for Random Forest tuning.

Figure 1 shows parity plots comparing predicted and true `U0` values. The RF correctly captures the overall monotonic relationship between structure and atomization energy, but predictions cluster into vertical bands reflecting the discrete nature of the Morgan fingerprints: many molecules share identical or near-identical bit patterns, causing the RF to average their corresponding energies. The model also displays a mild regression-to-the-mean effect, with high-energy molecules under-predicted and low-energy molecules over-predicted.



Train

Validation

Test

Figure 1: Parity plots for the Random Forest model on the training, validation, and test splits. Each subplot shows predicted `U0` versus the ground-truth value, with the diagonal line corresponding to perfect agreement.

Feature-importance values (Figure 2) reveal that only a small number of fingerprint bits contribute meaningfully to the prediction. These bits likely correspond to specific substructures that strongly correlate with molecular atomization energy (e.g., carbonyl groups or triple bonds), whereas the majority of bits have near-zero importance, which appears to be consistent with their sparse structure.

Overall, the Random Forest provides a stable baseline model with moderate predictive accuracy, potentially limited primarily by the limited non-geometric nature of the Morgan fingerprint representation. This result motivates comparing RF performance to more flexible models such as Support Vector Regression, neural networks, and symbolic regression.

Split	**RMSE**	**MAE**
Train	673.4	455.6
Val	679.0	460.6
Test	698.3	467.9

Table 2: Train, validation, and test RMSE and MAE stats for the best Random-Forest parameter combo found by , namely 100 trees (`n_estimators = 100`), unrestricted depth (`max_depth = None`), minimum leaf size of 4 (`min_samples_leaf = 4`), minimum split size of 2 (`min_samples_split = 2`), and feature sub-sampling ratio of 0.25 (`max_features=0.25`).

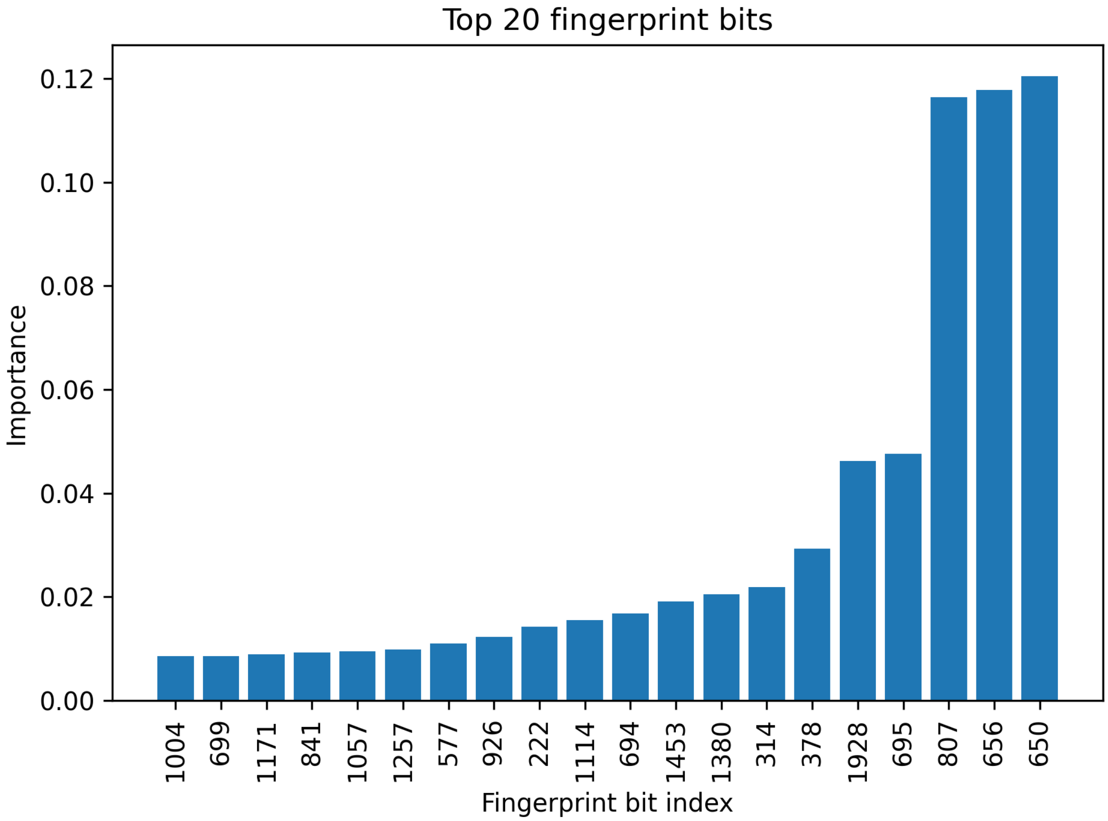


Figure 2: Top 20 most important fingerprint bits for the cross-validated Random Forest model 1, obtained with `RandomForestRegressor’s` `feature_importances_` attribute.

2.3 Support-Vector Regression

In this project, we used the Support Vector Regression (SVR) from the scikit-learn machine-learning library. SVR extends support vector machines to continuous targets and is well-suited for high-dimensional, sparse input features such as Morgan fingerprints. The model uses the radial basis function (RBF) kernel, which allows it to learn nonlinear relationships between molecular fingerprints and atomization energy. Data preprocessing is handled through scikit-learn’s Pipeline, which standardizes features with StandardScaler before training the SVR model. This combination provides a stable and flexible workflow for fitting large molecular datasets and evaluating predictive performance on held-out test data.

SVR with an RBF kernel scales poorly with dataset size, with training time growing approximately quadratically to cubically in the number of samples, it was not computationally feasible to train on the full QM9 fingerprint dataset. Even with parallelization on a modern CPU, a full fit would require many hours or longer. To keep the workflow manageable, the project uses representative random subsets for both hyperparameter tuning and final model training. A subset of about 8,000 samples was used during grid search to reduce tuning time from hours to minutes, and a larger 25,000-sample subset was used for the final model. This approach preserves the statistical structure of the data while keeping computational cost reasonable, allowing the SVR model to be trained and evaluated effectively within practical time limits.

A structured grid search was used to select effective hyperparameters for the RBF SVR model (Table 3). The search focused on the three parameters that most strongly influence SVR performance: the penalty parameter C, the tolerance band epsilon, and the kernel width controlled by gamma. The values of C were chosen to span different model capacities, from more regularized to more flexible fits. The epsilon values were selected to adjust how closely the model follows the training data, with smaller values encouraging a tighter fit. For gamma, the search included both the default “scale” setting and several small numeric values, since high dimensional fingerprint features often benefit from smoother kernels. The search grids were kept compact to maintain reasonable computation time while still covering the most important regions of the parameter space. This provided an efficient balance between runtime and thorough exploration of the model’s key hyperparameters.

The best performing SVR model used the RBF kernel with C = 900, epsilon = 0.005, and gamma = 3e-4. These hyperparameters provided an effective balance for the high dimensional fingerprint data and produced the validation and test results summarized in Table 4. The larger value of C allowed the model to fit nonlinear patterns more closely without becoming

Hyperparameter	**Search Space**
Kernel (`kernel`)	{rbf}
Penalty parameter (`C`)	{300, 600, 900}
Epsilon-insensitive band (ϵ)	{0.005, 0.01}
Kernel width (`gamma`)	{scale, 1e-3, 3e-4}
Feature scaling	`StandardScaler`
Training subset size	{8000 for search, 25000 for final model}

Table 3: Hyperparameter search space used for Support Vector Regression with the RBF kernel.

Split	**RMSE**	**MAE**
Training (best subset)	531.08	223.69
Validation (best subset)	764.19	420.00
Test (final 25k model)	628.54	401.02

Table 4: Training, validation, and test performance for the selected RBF SVR hyperparameters. The training row corresponds to a 4000 sample subset used to assess model fit capacity, the validation row corresponds to the best subset-based hyperparameter search model, and the test row reports performance of the final model trained on 25,000 samples.

too sensitive to noise. The small epsilon value encouraged tighter adherence to the training data, which is appropriate for smoothly varying atomization energies. The small gamma value produced a smoother RBF kernel and prevented overly sharp decision functions, which is important for sparse 2048 bit fingerprints where overly large gamma values can lead to overfitting. The resulting model demonstrated reasonable generalization, as reflected by the validation RMSE of 764.19 and the test RMSE of 628.54 in Table 4. This performance trend is also visible in the parity plot shown in Figure 3, where the predictions follow the overall diagonal trend but exhibit systematic underprediction and increasing spread at larger magnitudes.

2.4 Symbolic Regression

To obtain an interpretable analytic model for atomization energy, we applied symbolic regression using the `PySR` package. Unlike Random Forests or neural networks, which offer strong accuracy but limited interpretability, symbolic regression searches directly for closed-form expressions relating structure to property.

Our initial use of `PySR` is a *baseline experiment* intended only to test feasibility. Because a full search over the 2048-bit Morgan fingerprint is intractable, we restricted `PySR` to the 20 most informative bits identified by the default hyperparametrized Random Forest (Section 2.2). `PySR` was trained on a 5% subsample of the training data ($\approx$ 2000 molecules) with a small evolutionary budget (`niterations = 100`), and with unary-operator-set $\{\texttt{sin}, \texttt{cos}, \texttt{exp}\}$.

This baseline model attains higher errors than the Random Forest, as expected under these constraints, namely, (RMSE, MAE) $\approx$ (881, 647), validation (RMSE, MAE) $\approx$ (896, 653), and test (RMSE, MAE) $\approx$ (917, 663). Nevertheless, `PySR` produces a compact and interpretable expression:

$$\hat{U}_0 = -793.0\ (\texttt{fp}_{650} + \texttt{fp}_{656} + \texttt{fp}_{807}) - 10238.5,$$

where each $\texttt{fp}_j$ marks the presence of a specific molecular substructure. The model indicates that any of these fragments contributes a large, approximately additive decrease in atomization energy.

Split	**RMSE**	**MAE**
Train (subset)	891.1	653.0
Validation	895.5	653.4
Test	916.7	663.7

Table 5: Tuned `PySR` performance using the top 30 fingerprint bits and a 10% training subset. This configuration prioritizes speed and interpretability over accuracy.

2.4.1 Improved Symbolic Regression

To assess whether relaxing the baseline constraints could yield more expressive formulas, we increased the PySR training subset from 5% to 10%, expanded the feature set from 20 to 30 RF-selected fingerprint bits, added `tanh` to the unary-operator

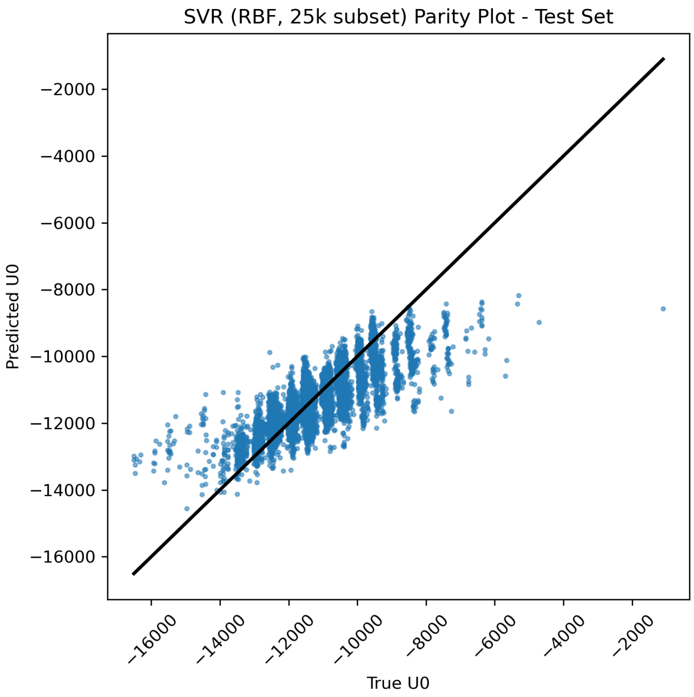


Figure 3: Parity plot comparing the predicted and true atomization energies for the final SVR model trained on the 25,000 sample subset. The diagonal line represents perfect agreement. The distribution of points shows that the model captures the overall trend in the data, but exhibits consistent underprediction at higher energies and increasing spread as the magnitude of the true values increases. This pattern reflects the nonlinear structure and wide dynamic range of the QM9 atomization energies.

set, and raised the evolutionary budget to `niterations = 1000`. From the resulting pareto-front, PySR chose the “best” expression as essentially the same functional form as in the baseline experiment, namely:

$$\hat{U}_0 = -788.5488\,(\texttt{fp}_{650} + \texttt{fp}_{656} + \texttt{fp}_{807}) - 10249.428 \tag{1}$$

again relying on the three dominant fragments $\texttt{fp}_{650}$, $\texttt{fp}_{656}$, and $\texttt{fp}_{807}$. This indicates that, under binary Morgan fingerprints, the symbolic optimum is intrinsically low-dimensional and largely unaffected by additional data or search depth. The performance metrics for equation (1) are shown in table 5, and Fig. 5 shows the parity plots for equation (1), which exhibit the characteristic banding and strong regression-to-the-mean behavior expected from a symbolic model built on binary fingerprint bits.

To understand the trade-off between expression complexity and prediction error, we evaluated all PySR equations on the train/validation/test splits and plotted RMSE/MAE versus algebraic complexity (Fig. 4). The best test performance occurs at complexity$\sim$ 30, for which PySR returns

$$\begin{aligned}\hat{U}_0 = -1370.17\,\texttt{fp}_{1928}\,e^{\texttt{fp}_{1453}} - 813.53\big(&\texttt{fp}_{650} + \texttt{fp}_{656} + \texttt{fp}_{807} \\ &+ \tanh(\texttt{fp}_{378} + \texttt{fp}_{695} + \tanh(\texttt{fp}_{1171} + \texttt{fp}_{577} - \texttt{fp}_{974}))\big) - 10047.79,\end{aligned} \tag{2}$$

with (RMSE, MAE) ≈ (782, 561) on train, (772, 554) on validation, and (788, 563) on test. Although this more complex expression improves RMSE by roughly 15% over the simple three-bit model, both symbolic expressions remain far less accurate than the RF, SVR, or neural-network models.

2.5 Feed-Forward Neural Network

The neural-network models were implemented using PyTorch, an open-source deep learning framework widely used for scientific computing and GPU-accelerated machine learning. Training was performed with the Adam optimizer, and GPU acceleration was enabled through the CUDA backend.

To understand how architectural choices and training dynamics affect prediction accuracy on Morgan fingerprints, we performed a structured grid search over several key neural-network hyperparameters. Fingerprints are high-dimensional,

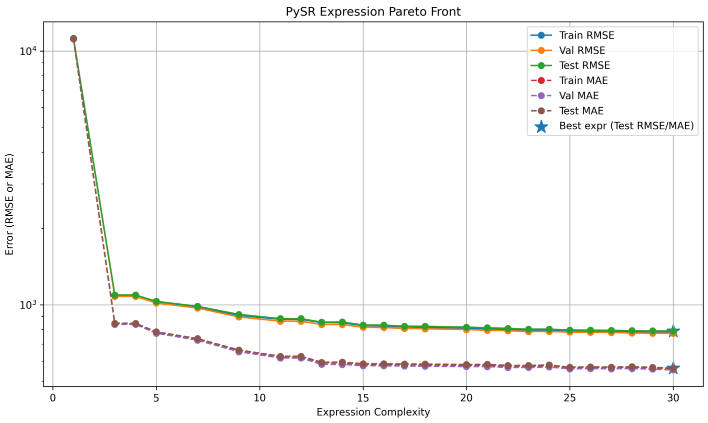
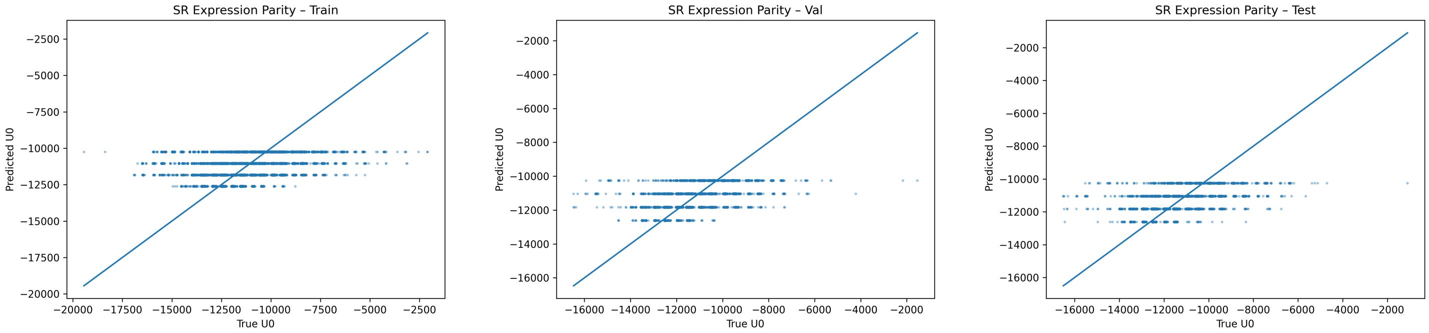


Figure 4: Error–complexity Pareto front for PySR expressions. Each point corresponds to one candidate equation fitted on the RF-selected fingerprint bits; curves show train/validation/test RMSE and MAE as a function of symbolic complexity. The starred point marks the expression with the lowest test RMSE (equation 2).



Train

Validation

Test

Figure 5: Parity plots for the Symbolic Regression model (1) on the training, validation, and test splits. Each subplot shows predicted `U0` versus the ground-truth value, with the diagonal line corresponding to perfect agreement.

sparse binary vectors (2048 bits), so model performance depends heavily on the network's ability to extract meaningful nonlinear combinations while avoiding overfitting to rare substructures. We varied the network width and depth by testing two architectures, [512, 256] and [1024, 512, 256], to evaluate whether a deeper hierarchy provides additional representational power or simply adds unnecessary capacity. Activation functions (ReLU and GELU) were included because GELU's smoother nonlinearities can improve optimization on sparse inputs, while ReLU provides a baseline that is computationally simple and widely used. We also explored dropout levels (0.0 and 0.1) to examine whether stochastic regularization improves generalization by preventing large networks from memorizing sparse fingerprint patterns.

Several training-level hyperparameters were also systematically varied to study their influence on generalization. We tested two weight-decay strengths (1e-5 and 1e-4) because L2 regularization is crucial when working with noisy, redundant fingerprint features; stronger regularization encourages smoother solutions and reduces reliance on individual bits. The learning rate (1e-3 and 5e-4) was varied to capture the trade-off between fast convergence and stable optimization, especially when paired with dropout and deeper architectures. Finally, testing two batch sizes (1024 and 2048) allowed us to probe how gradient-noise levels affect generalization; moderate batch sizes can introduce beneficial stochasticity, while overly large batches typically converge to sharper, less stable minima. Together, these hyperparameter sweeps were designed to map how architectural depth, regularization, optimization dynamics, and gradient noise interact when learning from ECFP fingerprints, enabling the identification of the most stable and generalizable neural-network configuration.

The best-performing neural network used a deep architecture with hidden layers [1024, 512, 256], GELU activations, dropout of 0.1, weight decay of 1×10^{-4}, a learning rate of 5×10^{-4}, and a batch size of 1024. When retrained on the full training split with early stopping, the model converged at epoch 34, achieving the performance metrics summarized in Table 7. In terms of predictive accuracy on unseen data, the model obtained an RMSE of 419.7 kcal/mol, MAE of 254.6 kcal/mol, and $R^2 = 0.852$ on the held-out test set, with qualitative behavior shown in the parity plot in Figure 6.

This deeper architecture was only effective when paired with strong regularization—including dropout and L_2 weight decay—and the smoother GELU activation function, which together helped prevent overfitting on the sparse, high-dimensional

Hyperparameter	**Search Space**
Hidden layer sizes (`hidden`)	{[512, 256], [1024, 512, 256]}
Activation function (`act`)	{`relu`, `gelu`}
Dropout rate (`dropout`)	{0.0, 0.1}
Weight decay (`weight_decay`)	{1e-5, 1e-4}
Learning rate (`lr`)	{1e-3, 5e-4}
Batch size (`batch_size`)	{1024, 2048}
Maximum epochs (`epochs`)	{100}
Early stopping patience (`patience`)	{10}

Table 6: Hyperparameter search space used for neural network tuning on Morgan fingerprints.

Morgan fingerprint inputs. The moderate batch size of 1024 also introduced beneficial gradient noise that improved generalization relative to larger batches. Across the full hyperparameter grid, this configuration achieved the best balance between model capacity and regularization, making it the optimal neural architecture for predicting atomization energies from molecular fingerprints.

Split	**RMSE**	**MAE**	R^2
Train	45.88	24.56	0.9982
Validation	417.30	251.40	0.8516
Test	419.74	254.55	0.8522

Table 7: Performance of the best neural network model trained on the different training splits. The network uses hidden layers [1024, 512, 256], GELU activations, dropout 0.1, weight decay 1×10^{-4}, learning rate 5×10^{-4}, and batch size 1024, with early stopping at epoch 34 based on validation RMSE.

2.6 Fingerprints vs. Descriptors

We trained a fully connected neural network to predict molecular atomization energies using RDKit descriptor features. Because the descriptor matrix is dense, lower dimensional, and more information rich than Morgan fingerprints, we performed a targeted grid search over the architectural and optimization hyperparameters most likely to influence performance (Table 8). The search varied hidden layer sizes (two vs. three layers with 512–1024 units), activation functions (ReLU vs. GELU), dropout (0 to 0.1), weight decay (10^{-5} to 10^{-4}), learning rate (0.001 to 0.0005), and batch size (512 to 1024). Each configuration was trained with early stopping on the validation set to avoid overfitting and reduce runtime. The results consistently showed that deeper and wider models with GELU activation, no dropout, modest weight decay, and a moderate batch size achieved the lowest validation error. The best model, with hidden layers [1024, 512, 256], GELU activation, weight decay 10^{-4}, learning rate 0.001, and batch size 512, generalized extremely well, substantially outperforming the fingerprint-based network (Table 10). It reached a test RMSE of 5.58 kcal/mol, MAE of 3.45 kcal/mol, and $R^2 = 0.999974$, and its predictions fall tightly along the ideal $y = x$ line in the parity plot (Figure 7). These results demonstrate that descriptor features allow the network to learn a smooth, highly accurate mapping from molecular structure to atomization energy.

Table 8: Best hyperparameters identified for the descriptor-based neural network.

Hyperparameter	Value
Hidden layers	[1024, 512, 256]
Activation function	GELU
Dropout	0.0
Weight decay	10^{-4}
Learning rate	0.001
Batch size	512
Epochs (max)	100
Early stopping patience	10
Best epoch	99

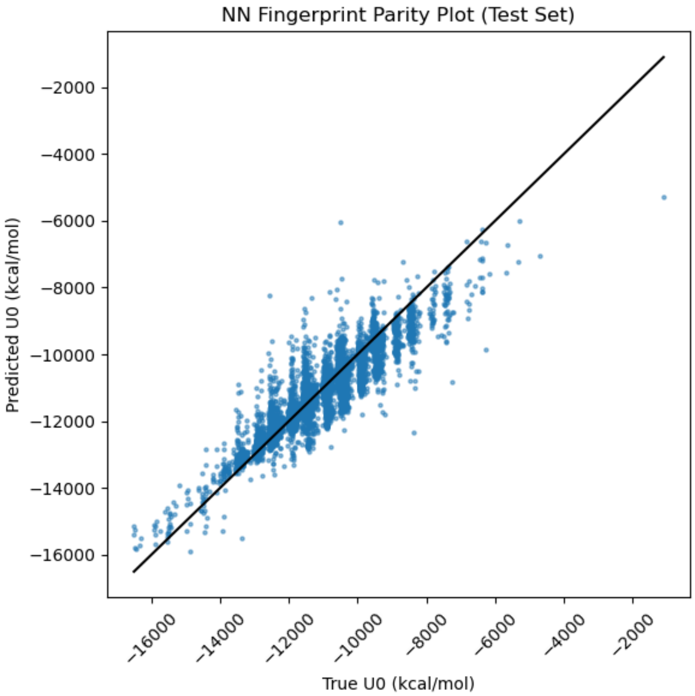


Figure 6: Parity plot comparing predicted and true atomization energies for the test set. The diagonal reference line indicates perfect agreement. The neural network model shows strong alignment with ground-truth U0 values, demonstrating good generalization performance on unseen molecules.

Table 9: Training, validation, and test metrics for the descriptor–based neural network (best configuration).

Split	RMSE (kcal/mol)	MAE (kcal/mol)	R^2
Train	3.79	2.86	0.999988
Validation	6.06	3.48	0.999969
Test	5.58	3.45	0.999974

Table 10: Comparison of neural network test performance using descriptor features vs. fingerprint features.

Model	RMSE (kcal/mol)	MAE (kcal/mol)	R^2
NN (Descriptors)	5.58	3.45	0.999974
NN (Fingerprints)	419.74	254.55	0.8522

3 Remarks

Across all models, performance was strongly influenced by both the feature representation and the model class. The Random Forest baseline demonstrated that while tree ensembles can capture coarse structure–property trends, the fingerprint representation induces vertical banding in the parity plots (Fig. 1) due to repeated bit patterns. This led to moderate accuracy and clear limits in expressive power.

SVR with an RBF kernel (Tables 3–4) improved upon the Random Forest by enabling nonlinear relationships in the high-dimensional fingerprint space. However, the cubic scaling of RBF SVR with dataset size required training on subsets, which constrained achievable accuracy. Even at optimized hyperparameters, SVR showed noticeable underfitting relative to the neural models.

Symbolic Regression (PySR) offers interpretability but is heavily constrained by the binary fingerprint representation. Even after expanding the feature set and search budget, the optimized expression (1) depends on just three fingerprint bits, leading to large errors (Table 5) and highly quantized predictions (Fig. 5). These results suggest that meaningful analytic structure–property relationships cannot be recovered from sparse fingerprints without richer geometric or electronic descriptors.

The fingerprint-based neural network offered a large increase in capacity, but the sparse and highly redundant Morgan fingerprints still posed difficulty. Despite deep architectures and strong regularization, performance plateaued, as seen in

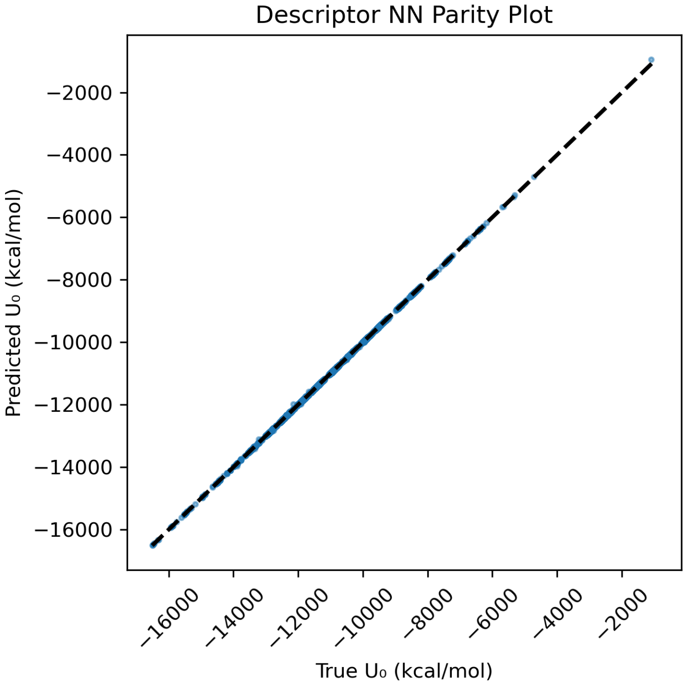


Figure 7: Parity plot comparing predicted atomization energies from the descriptor based neural network with the true test set values. Alignment along the 45° line indicates strong predictive performance, with a test RMSE of approximately 5.58 kcal/mol and $R^2 \approx 0.99997$. The tight clustering demonstrates that the model captures the structure–property relationships with very high fidelity.

the parity plot (Fig. 6), where systematic deviations from the $y = x$ line remain. This suggested that the fingerprint representation itself was the fundamental limitation.

In contrast, the descriptor-based neural network (Tables 8 and 9) achieved the best results by a substantial margin. The descriptor features are dense, continuous, and information rich, enabling the model to learn a smooth structure–property mapping. The resulting parity plot (Fig. 7) shows extremely tight alignment with the ideal diagonal, and the descriptor-based network significantly outperforms the fingerprint-based version in all metrics (Table 10). This confirms that model performance was limited less by neural architecture and more by the expressiveness of the feature representation.

4 Conclusion

This study evaluated multiple machine learning approaches for predicting molecular atomization energies in the QM9 dataset, focusing on how model choice and feature representation influence predictive accuracy. Random Forests and Support Vector Regression provided useful baselines, but both were ultimately limited by the sparse Morgan fingerprint representation. Their performance metrics and parity plots revealed consistent underfitting, indicating that the fingerprint space lacks the continuity needed for smooth energy prediction.

Symbolic regression, while providing more interpretability in its $X \rightarrow y$ mapping, sacrificed quantitative performance (5). Increasing the complexity helps to a degree (4) but at the expense of interpretability (2). Exploring a richer basis-set of operators such as aggregate operations $\mathbb{R}^{N>1} \rightarrow \mathbb{R}^N$ might provide better interpretability and performance in the larger-feature-space arena.

Neural networks improved substantially upon these baselines, demonstrating the benefits of learned nonlinear transformations. However, when trained on fingerprint features, even deep architectures with carefully tuned hyperparameters showed systematic deviations from ideal predictions. This highlighted a central conclusion of the project: model accuracy is constrained not only by algorithmic capacity but more fundamentally by feature expressiveness. The descriptor-based neural network overcame this barrier by leveraging dense, chemically meaningful RDKit descriptors. These features enabled the network to learn a highly accurate structure–property mapping, achieving exceptional test performance (Table 9) and nearly perfect alignment in the parity plot (Fig. 7). In direct comparison, the descriptor model outperformed the fingerprint-based network by nearly two orders of magnitude in RMSE (Table 10), demonstrating that the choice of representation is the dominant factor governing accuracy in this task.

Overall, the results show that while traditional models and sparse fingerprints capture coarse structural trends, high-quality continuous descriptors paired with a suitably expressive neural network provide state-of-the-art performance. This suggests that future work in molecular energy prediction should prioritize improved feature engineering or learned representations (e.g., graph neural networks) to unlock even greater accuracy and generalizability.

5 Contributions

Edward worked on the random forest and symbolic regression models, and Tyler worked on the support vector regression and neural network models. Both contributed in the project conceptualization.

6 Code

The code for this report can be found here.