Machine learning for HOMO-LUMO-gap Prediction and Inverse Molecular Design

Mattice Criel

Yarno De Jaeger

Thibo Van Eeckhoorn

Abstract

In the abstract of your paper, briefly summarize your research in about 150 to 250 words. Briefly explain the problem statement, the techniques you used, and your general results. You can't go into deep detail here, of course, but you don't have to. Think of this as a kind of written "elevator pitch": explain your research to someone who has 1-2 minutes time to listen.

1 Introduction

Within the spectrum of discrete energy levels (molecular orbitals) in a molecule that can be filled with electrons, the HOMO-lUMO gap is the energy difference between the highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO). This is a fundamental descriptor of the electronic structure of a molecule and is strongly influenced by the specific molecular structure and its functional groups. The HOMO-LUMO gap governs key properties such as reactivity, optical absorption and charge transport characteristics and is directly related to the band-gap in conductivity, determining whether a material behaves as a conductor, insulator or semi-conductor(Dwivedi et al., 2025). The HOMO-LUMO gap also influences the effiency of organic photovoltaics (solar cells based on organic molecules) and organic light-emitting devices (OLED-technology) (Liu et al., 2015). As a consequence, accurate determination of the HOMO-LUMO gap and design of molecular materials with a specified HOMO-LUMO gap is essential for both materials science and molecular engineering.

Determining the HOMO-LUMO gap, by approximation, is typically done in an experimental setting through optical spectroscopy or voltammetry(Costa et al., 2016; Sworakowski, 2018). However, experimental measurements require synthesized, purified materials making large-scale exploration of hypothetical chemical structures cost expensive and time consuming. As an alternative, methods in computational quantum chemistry, such as Hartree-Fock and Density functional theory (DFT) are used to simulate approximations. Despite their accuracy, these quantum chemical methods are computationally expensive and scale poorly with molecular size and dataset volume, also restricting the capacity of HOMO-LUMO gap exploration and molecular design.

Machine learning (ML) provides a strategy to overcome these computational limitations(Hasan et al., 2025). After training, ML models can predict electronic properties with negligible computational cost compared to traditional quantum chemistry. Crucially, large-scale quantum-chemical datasets like QM9, contain extensive information about molecular structures and properties of organic molecules, enabling the development and benchmarking of ML models with the data volume required for robust learning. Beyond property prediction, ML also enables generative molecular design, making it possible to explore hypothetical chemical structures, optimize properties in silico, and drastically reduce the number of costly experiments. This accelerates molecular discovery pipelines, supports safer design exploration, and opens chemical regions that would be impractical to investigate experimentally.

In this study, we pursue a two-fold objective. First, we build predictive models for HOMO-LUMO gap estimation using both Light Gradient Boosting Model (LightGBM), a high-performance gradient boosting framework and a graph convolutional network (GCN) that learns molecular representations directly from graph topology. Second, we integrate the predictive GCN into a conditional variational autoencoder (cVAE) to generate novel molecular structures with user-specified target HOMO-LUMO gaps. This combined framework enables both accurate property prediction and inverse molecular design, supporting accelerated discovery workflows in molecular design and thereby potentially facilitating a more directed control of specific chemical reactions and increasing the relevance of organic based technology (like OPVs and OLED).

NEED TO ADD A PART ABOUT OUR RESULTS

2 Related work

In recent years, significant progress has been made in predicting HOMO-LUMO gaps using various ML techniques, such as deep learning, ensemble methods, and graph-based models.

...

In the context of generative models, variational autoencoders (VAEs) have emerged as a standard framework for de novo molecular design(Walters & Barzilay, 2021). One of the more recent influencial works is done by Gómez-Bombarelli et al. (2018), which adopted a variational autoencoder to optimize the molecular properties in a latent space and uses a Gaussian process to optimize a chemical structure with the desired properties. Lim et al. (2018) later demonstrated a conditional VAE framework that enables property-controlled molecular generation and serves as a foundational inspiration for our work. Both studies highlight limitations arising from SMILES-based representations and suggest that graph-based encoders and decoders could better capture molecular structure and improve validity, motivating the graph-based generative approach adopted in our research.

TO BE CHANGED

3 Methodology

3.1 Dataset

For all models in this study, we used the QM9 dataset(Team, 2024), a widely adopted quantum chemistry dataset comprising 130,831 small organic molecules with associated properties, including the HOMO–LUMO gap. Only molecules with valid SMILES representations were retained, leaving 129,012 molecules, can be seen in Figure 1. SMILES (Simplified Molecular Input Line Entry System) is a text-based notation that encodes chemical structures as ASCII strings, enabling efficient computational processing; for example, benzene is represented as `C1=CC=CC=C1`. While SMILES strings can be interpreted by computers, they are not directly suitable as input for most machine learning models, which require numerical representations. Consequently, preprocessing is necessary to transform molecular structures into informative, fixed-length numerical features suitable for model training.

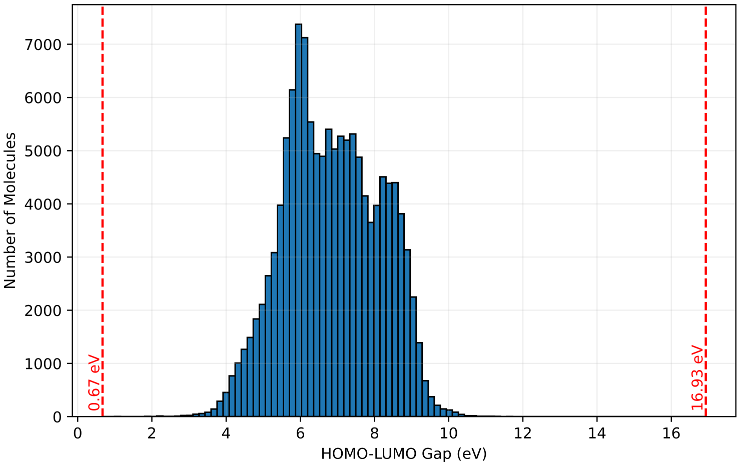

Figure 1: Illustration of the bandgap distribution in the QM9 dataset where only the valid molecules are kept. The vertical dashed lines indicates the minimum and maximum bandgap values in the dataset.

3.2 Molecular Representations and Preprocessing

In this study, we focus on two different models for HOMO-LUMO gap prediction: LightGBM, a classical machine learning model and GCN, a deep learning model, which require different types of molecular representations. Therefore, two types of molecular representations were generated from the SMILES strings.

3.2.1 Descriptor-based features for LightGBM

Using RDKit, a set of 217 molecular descriptors was generated for each molecule from the SMILES representation(RDKit: Open-Source Cheminformatics, 2025). These descriptors capture various aspects of molecular structure and properties, including molecular size, shape, electronic characteristics, surface area, and the presence of functional groups. Together, they provide a fixed-length numerical representation of each molecule suitable for machine learning models such as LightGBM.

3.2.2 Graph-based representation for GCN:

For graph neural networks, molecules are represented as graphs where atoms are nodes and bonds are edges. As the QM9 dataset represents molecules as SMILES, all molecule SMILES were translated to a node and edge matrix using the *Chem* package of RDKit(RDKit Developers, 2025). Node features include atomic number, formal charge, hybridization, aromaticity, and whether the atom is in a ring structure. The edge matrix was defined as an adjacency matrix where the diagonal elements were set to 1, indicating a self-connection, which makes the matrix amenable to convolutions(Deshmukh, 2023). The distance of each bond was included in the edge matrix as 1 over the bond distance. The representation of a molecule as a node and edge matrix allows the GCN to learn structural features directly from molecular graphs without relying on precomputed descriptors.

3.3 Data Splitting

To ensure unbiased evaluation, the dataset was split into training, validation, and test sets. For all models, nested cross-validation was used:

- **Outer loop:** 5-fold cross-validation to estimate generalization performance

- **Inner loop:** 10-fold cross-validation for hyperparameter tuning

The splits were stratified based on the distribution of the HOMO-LUMO gap values into 10 quantile bins to preserve this distribution across the different folds.

3.4 Evaluation metrics

Since both the LightGBM and GCN models are used to predict HOMO-LUMO gaps and are directly compared, the same evaluation metrics are applied to both. Commonly used metrics for HOMO-LUMO gap prediction include the mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination (R^2) between the predicted and true HOMO-LUMO gaps.
The cVAE model performance is evaluated using the average character-level reconstruction accuracy ($\overline{CRA}$), the chemical validity of newly generated SMILES strings using RDKit, and the agreement between the target HOMO-LUMO gap and the GCN-predicted gap of the generated molecules, quantified using MAE. Each metric is calculated as

$$MAE = \frac{1}{N}\sum^{N}|y_{pred} - y_{true}| \tag{1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum^{N}(y_{pred} - y_{true})^2} \tag{2}$$

$$R^2 = 1 - \frac{\sum^{N}(y_{true} - y_{pred})^2}{\sum^{N}(y_{true} - \bar{y})^2} \tag{3}$$

$$\overline{CRA} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{T_n}\sum_{t=1}^{T_n}\mathbf{1}(\hat{x}_{n,t} = x_{m,t}) \tag{4}$$

with N the total number of test molecules and T_n the number of characters per SMILES string.

3.5 LightGBM

LightGBM is a gradient-boosted decision tree algorithm optimized for speed and high-dimensional data. It grows trees in a leaf-wise rather than level-wise manner, allowing it to capture complex, non-linear relationships more efficiently than many traditional boosting methods. This makes it well-suited for tabular molecular descriptor data, as it is robust to feature scaling, multicollinearity, and missing values.
Hyperparameter optimization for the LightGBM model was carried out using Optuna. The search space included parameters controlling model complexity and regularization: the number of leaves (num_leaves, 16–256), the learning rate (learning_rate, 1×10^{-3}–0.1, log-scaled), feature subsampling (feature_fraction, 0.7–1.0), sample bagging (bagging_fraction, 0.7–1.0; bagging_freq, 1–5), and the minimum number of samples per leaf (min_data_in_leaf, 10–100). Other LightGBM settings, such as the boosting type (gbdt) and objective function (regression_l1), were kept fixed throughout. Optuna selected the optimal hyperparameter configuration by minimizing the mean absolute error (MAE) obtained from the inner loop of the nested cross-validation, using a total of 50 trials.

After nested cross-validation, the best-performing hyperparameters were selected (either by averaging over outer folds or by majority vote). The final LightGBM model was then trained on the entire dataset to produce predictions for comparison with the GCN model.

3.6 GCN

The implementation of the GCN model was based on a tutorial about creating a simple Pytorch-based GCN(Deshmukh, 2023). Some edits to the tutorial code were made, for example the addition of node features, the addition of an R^2 metric, and correcting a mistake with the standardization of the loss.

Starting from the node and edge matrix representations of a graph, the GCN model first applied a graph convolution using multiple convolution layers. Each convolution layer gives a node information about its neighbors using the matrix multiplication shown in Figure 2. After the convolution layers *pooling* is applied, which turns the 2-dimensional matrix into a 1-dimensional vector that contains the means of every column of the node matrix. This vector can then be passed to the neural network, which is a simple multilayer perceptron (MLP), a fully connected feedforward neural network. For more details on the implementation, see the GitHub repository add ref to repo.

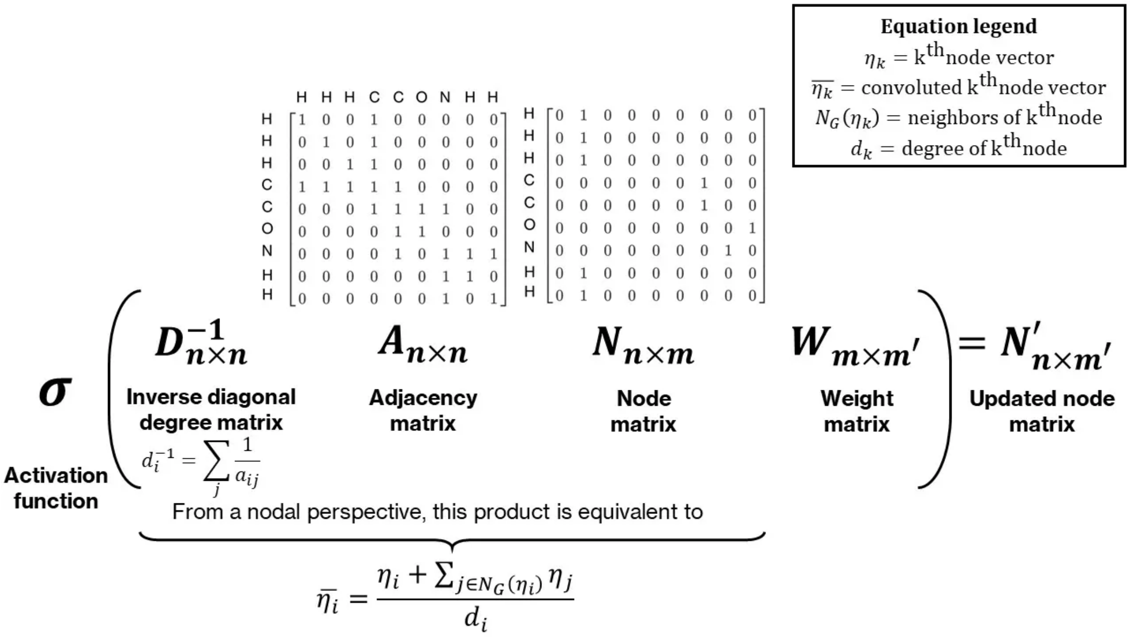
Equation legend
η_k = k$^{\text{th}}$node vector
$\overline{\eta_k}$ = convoluted k$^{\text{th}}$node vector
$N_G(\eta_k)$ = neighbors of k$^{\text{th}}$node
d_k = degree of k$^{\text{th}}$node
σ
Activation function
$D^{-1}_{n\times n}$
Inverse diagonal degree matrix
$d_i^{-1} = \sum_j \frac{1}{a_{ij}}$
$A_{n\times n}$
Adjacency matrix
$N_{n\times m}$
Node matrix
$W_{m\times m'}$
Weight matrix
$= N'_{n\times m'}$
Updated node matrix
H H H C C O N H H
From a nodal perspective, this product is equivalent to
$\overline{\eta_i} = \frac{\eta_i + \sum_{j\in N_G(\eta_i)} \eta_j}{d_i}$

Figure 2: Graph convolution for an acetamide molecule(Deshmukh, 2023).

Hyperparameter optimization for the GCN model was performed using a simple grid search, as this allowed easy parallelization of hyperparameter tuning. The parameters and their possible values included in the grid search are: batch_size (64, 128, 256, 384), hidden_nodes (64, 96, 128), n_conv_layers (1-5), n_hidden_layers (1-3), learning_rate (0.001, 0.003, 0.005, 0.007, 0.01). Other parameters such as the maximum dimensions of the node and edge matrix, and the number of epochs were kept constant. After nested cross-validation, the same best-performing hyperparameters were obtained for each outer fold: batch_size = 256, hidden_nodes = 128, n_conv_layers = 4, n_conv_layers = 2, learning_rate = 0.003, the number of epochs was kept constant at 50.

3.7 cVAE

The conditional variational autoencoder (cVAE) is chosen for its probabilistic nature, which enables sampling of new molecules with higher validity than a standard autoencoder (Kingma & Welling, 2019). Our implementation is partially inspired by (Gómez-Bombarelli et al., 2018; Lim et al., 2018), except for the encoder. The model consists of a graph-based GCN encoder, a continuous conditioned latent space, and a GRU sequence decoder (Cho et al., 2014; Yuan et al., 2020), combining the richer information from graph representations with the simplicity of SMILES decoding.

Training maximizes the conditional ELBO (equivalently, minimizes the negative ELBO), composed of a categorical cross-entropy reconstruction loss and a Kullback-Leibler divergence regularizing the latent space(Beckham, 2023):

$$\mathcal{L}_{cVAE} = -\mathbb{E}[\log\ p(x|z, y)] + \beta D_{KL}(q(z|x, y) \| \mathcal{N}(0, I)) \tag{5}$$

where $q(z|x, y)$ is the encoder's approximate posterior, mapping a molecule and its HOMO-LUMO gap to a latent distribution, and $p(x|z, y)$ is the decoder's likelihood, giving the probability of reconstructing a SMILES sequence from a latent vector and conditioning value.

Hyperparameters are optimized using Bayesian optimization with a TPE sampler (Watanabe, 2023), although only a limited search is feasible, consisting of only 15 trials, due to the large parameter space and time constraints. For the same reason, five inner folds are used for cross-validation instead of ten, but given the large size of the QM9 dataset, this reduction is not expected to significantly affect the results. An overview of the selected optimal hyperparameters is provided in Table 3.

The best set of hyperparameters is then used to generate new molecules in two ways: by randomly sampling latent vectors from the latent space and by perturbing latent vectors of dataset molecules, which are both then decoded conditioned on target HOMO-LUMO gaps. Generated molecules were evaluated by sampling for both experimental pathways 10 different molecules and for each molecule doing a 1000 generations, for which the average chemical validity and MAE are calculated per molecule, as described in Section 3.4.

4 Results

4.1 GCN

Just as for the LightGBM model, the mean absolute error (MAE), root mean square error (RMSE), and R^2 between the predicted and true HOMO-LUMO gap were determined for the GCN model. All evaluation metrics of the GCN model (Table 1) have slightly higher values than those of the LightGBM model (**??**).

Table 1: Mean absolute error (MAE), root mean square error (RMSE), and R^2 between predicted and target HOMO–LUMO gaps per outer fold. Mean values across all five folds are also reported.

	MAE	RMSE	R^2
Outer Fold 1	0.1677	0.2403	0.9646
Outer Fold 2	0.1585	0.2280	0.9682
Outer Fold 3	0.1567	0.2255	0.9690
Outer Fold 4	0.1586	0.2231	0.9697
Outer Fold 5	0.1587	0.2299	0.9679
Mean	0.16	0.23	0.97

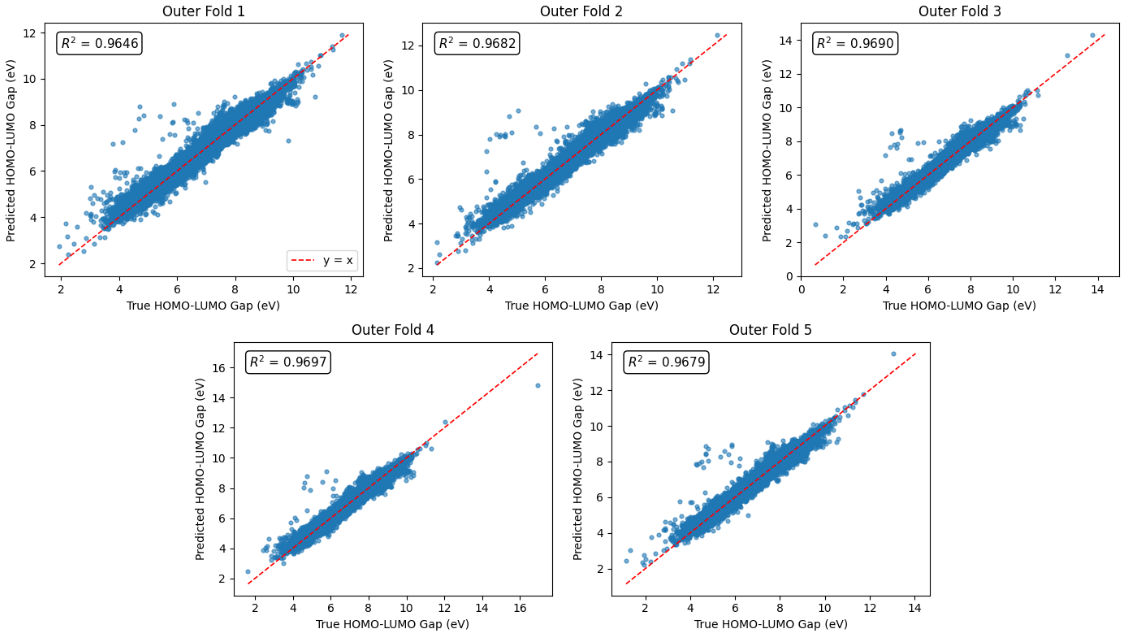

Figure 3: Predicted vs true HOMO-LUMO gap for all 5 outer folds of the GCN model.

4.2 cVAE

Across the five outer folds, the selected hyperparameters were largely consistent, three out of the five fold ended on the same set of parameters, while fold 2 and 5 managed to get a different set (Table 3). Between the five sets, the major differences lie in the size

The final hyperparameter configurations for each outer fold are summarized in Table X.

Overall the nested cross-validation strategy provided a robust estimate of the model performance and reduced the risk of overfitting to a single validation split. However the limited amount of hyperparameter tuning that was able to be done suggests that further gains may be achievable with a more extensive optimization strategy.

The reconstruction accuracy of the model, using $\overline{CRA}$ (Equation (4)), across the five outer folds are reported in **tab:CRA**. The model achieved a mean reconstruction accuracy of 0.852, indicating it is capable of reconstructing specific molecules.

Table 2: The average character-wise reconstruction accuracy, as calculated by Equation (4), of the test set per outer fold of the cross-validation.

	$\overline{CRA}$
Outer fold 1	0.842
Outer fold 2	0.840
Outer fold 3	0.871
Outer fold 4	0.846
Outer fold 5	0.863
Mean	0.852

Beyond reconstruction, the more important metric of interest is the generative capability of the cVAE. The chemical validity of the generated molecules can be found in **??** for both the random selected latent vectors and the perturbed latent vectors.

5 Discussion

5.1 LightGBM vs GCN

Here I would split intro three paragraphs:

1. Interpretation of LightGBM results and how it relates to literature
2. Interpretation of GCN results and how it relates to literature
3. Results of LightGBM vs GCN and what this could mean for the decision between traditional vs deep learning HOMO-LUMO gap prediction

6 Conclusion

7 Disclaimers

Appendix A. Hyperparameters

Table 3: The optimized hyperparameters for each outer fold achieved after running 15 trials in a Bayesian optimization algorithm using a TPE sampler. The number of epochs and batch size were fixed to narrow the searching space of the Bayesian algorithm.

	fold 1	fold 2	fold 3	fold 4	fold 5
learning rate	0.00133	0.00133	0.00548	0.00133	0.00548
latent dimension	62	62	50	62	50
#hidden nodes	79	79	66	79	66
GRU dimension	42	42	64	42	64
#convolution layers	1	1	1	1	1
#hidden layers	1	1	2	1	2
#GRU layers	1	1	1	1	1
#FC layers	3	3	2	3	2
embedding dimension	18	18	14	18	14
teacher forcing ratio	0.876	0.876	0.888	0.876	0.888
β	4.62	4.62	1.88	4.62	1.88
batch size	1000	1000	1000	1000	1000
#epochs	15	15	15	15	15

References

Beckham, C. (2023). Conditional vaes. https://beckham.nz/2023/04/27/conditional-vaes.html

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. https://arxiv.org/abs/1406.1078

Costa, J. C., Taveira, R. J., Lima, C. F., Mendes, A., & Santos, L. M. (2016). Optical band gaps of organic semiconductor materials. *Optical Materials*, *58*, 51–60. https://doi.org/https://doi.org/10.1016/j.optmat.2016.03.041

Deshmukh, G. (2023). *Building a graph convolutional network for molecular property prediction* [Accessed: 2025-12-15]. Medium – TDS Archive. https://medium.com/data-science/building-a-graph-convolutional-network-for-molecular-property-prediction-978b0ae10ec4

Dwivedi, A. D. D., Maji, D., & Chakrabarti, P. (2025). Introductory Chapter: Organic Electronics – From Fundamentals to Applications [Section: 1]. In A. D. D. Dwivedi, D. Maji, & P. Chakrabarti (Eds.), *Organic Electronics - From Fundamentals to Applications*. Section: 1. London, IntechOpen. https://doi.org/10.5772/intechopen.1011119

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., & Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules [PMID: 29532027]. *ACS Central Science*, *4*(2), https://doi.org/10.1021/acscentsci.7b00572, 268–276. https://doi.org/10.1021/acscentsci.7b00572

Hasan, M. M., Tarkhaneh, O., Bungay, S. D., Poirier, R. A., & Islam, S. M. (2025). Predicting homo-lumo gaps using hartree-fock calculated data and machine learning models [Epub 2025 Sep 10, PMID: 40929702]. *J. Chem. Inf. Model.*, *65*(18), 9497–9515. https://doi.org/10.1021/acs.jcim.5c01412

Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *CoRR*, *abs/1906.02691* arXiv 1906.02691. https://arxiv.org/abs/1906.02691

Lim, J., Ryu, S., Kim, J. W., & Kim, W. Y. (2018). Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of Cheminformatics*, *10*(1), 31. https://doi.org/10.1186/s13321-018-0286-7

Liu, X., Chen, H., & Tan, S. (2015). Overview of high-efficiency organic photovoltaic materials and devices. *Renewable and Sustainable Energy Reviews*, *52*, 1527–1538. https://doi.org/10.1016/j.rser.2015.08.032

RDKit Developers. (2025). *Rdkit.chem module documentation* [Accessed: 2025-12-15]. RDKit. https://www.rdkit.org/docs/source/rdkit.Chem.html

RDKit: Open-Source Cheminformatics. (2025). Rdkit documentation and python tutorials [Accessed December 6, 2025].

Sworakowski, J. (2018). How accurate are energies of homo and lumo levels in small-molecule organic semiconductors determined from cyclic voltammetry or optical spectroscopy? *Synth. Met.*, *235*, 125–130. https://doi.org/10.1016/j.synthmet.2017.11.013

Team, P. (2024). Torch_geometric.datasets.qm9 — pytorch geometric documentation [Accessed: 2025-12-09].

Walters, W. P., & Barzilay, R. (2021). Applications of deep learning in molecule generation and molecular property prediction [PMID: 33370107]. *Accounts of Chemical Research*, *54*(2), https://doi.org/10.1021/acs.accounts.0c00699, 263–270. https://doi.org/10.1021/acs.accounts.0c00699

Watanabe, S. (2023). Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint arXiv:2304.11127*. https://doi.org/10.48550/arXiv.2304.11127

Yuan, Q., Santana-Bonilla, A., Zwijnenburg, M. A., & Jelfs, K. E. (2020). Molecular generation targeting desired electronic properties via deep generative models. *Nanoscale*, *12*, 6744–6758. https://doi.org/10.1039/C9NR10687A