Advanced Methods in Applied Statistics

Written Project: An Exploration of Physics Based Constraints in Graph Neural Network Loss Functions for Molecular Property Predictions

Alexandra Haslund-Gourley and Søren Jefsen

March 23, 2025

Abstract

We explore the use of Graph Neural Networks (GNNs) to predict total energy, material band-gap, and Kohn-Sham gap (KS-gap) for Porphyrin-Based Dyes (PBDs). Specifically, we investigate how modifying the loss function to enforce known physical and statistical relationships affects predictive accuracy. We compare a standard mean squared error (MSE) loss with two alternative approaches: (I) incorporating a physical constraint that enforces band-gap values to exceed KS-gap values, and (II) modeling correlations among all three properties using a multivariate normal distribution, optimizing predictions via negative log-likelihood. Our results show that enforcing physical constraints (I) consistently improves prediction accuracy, whereas the statistical correlation approach (II) slightly degrades performance, highlighting the need to validate distributional assumptions for effective implementation.

1 Introduction

Materials science holds tremendous promise for discovering revolutionary materials that could transform energy storage, production, transmission, and myriad other scientific fields. However, the combinatorial space of possible atomic configurations for candidate materials is extraordinarily vast, making exhaustive experimental exploration infeasible. While traditional physics methods like Density Functional Theory (DFT) can predict critical material properties such as electronic structure in solids, such calculations are computationally expensive, particularly for large or complex molecular systems. As a result, there has been recent excitement about applying machine learning (ML) to bypass DFT calculations and offer quicker predictions based solely on molecular structure.

Due to the graph-like nature of molecules, Graph Neural Networks (GNNs) have emerged as a popular ML technique in materials science (Schütt et al. (2017), Reiser et al. (2022)). In GNNs, atoms are typically encoded as nodes on a graph, with properties like atomic number, mass, and charge serving as node features. The relationships between atoms (Coulomb potential, bond type, distance, etc.) can be encoded as features of the edges between nodes. Computation proceeds through message passing, where information is exchanged between neighboring nodes and updated iteratively, allowing information to be distributed across the graph. After several iterations of message passing (also referred to as graph convolution), the resulting embedded graph feature values can be passed through linear neural networks to make predictions about specific node, edge, or full graph properties. For a deeper dive into GNNs, please see the Appendix.

In this project, we investigate how incorporating physical constraints into the loss function of Graph Neural Networks (GNNs) affects their ability to predict molecular properties. We focus on a class of molecules called Porphyrin-Based Dyes (PBDs), which exhibit strong visible light absorption, making them valuable for applications in photovoltaic cells, photodynamic cancer therapy, and fluorescent imaging and sensing Li et al. (2024). The data for this study comes from the PBD Database developed at DTU Ornsø et al. (2013), which contains computed properties of various zinc-based porphyrins with different electron-donating side groups.

Our goal is to jointly predict the total energy, material band-gap, and Kohn-Sham gap (KS-gap) based on the atomic structure of PBDs. The molecular structures are provided in the Atomic Simulation Environment (ASE) format, detailing the spatial coordinates and atomic numbers of all constituent atoms. The three target properties, sourced from the PBD database, each capture different aspects of the electronic structure: total energy represents the system's overall energy, the band-gap quantifies the range of forbidden electronic states and influences electrical conductivity, and the KS-gap, derived from quantum mechanical calculations, differs from the band-gap, which incorporates empirical models and experimental data. Figure 4 in the Appendix presents histograms and correlations among these parameters, including total mass.

To improve predictive accuracy, we first analyze statistical relationships between total energy, band-gap, and KS-gap. Based on these insights, we introduce physics-based constraints into the GNN's loss function and assess their impact on model performance.

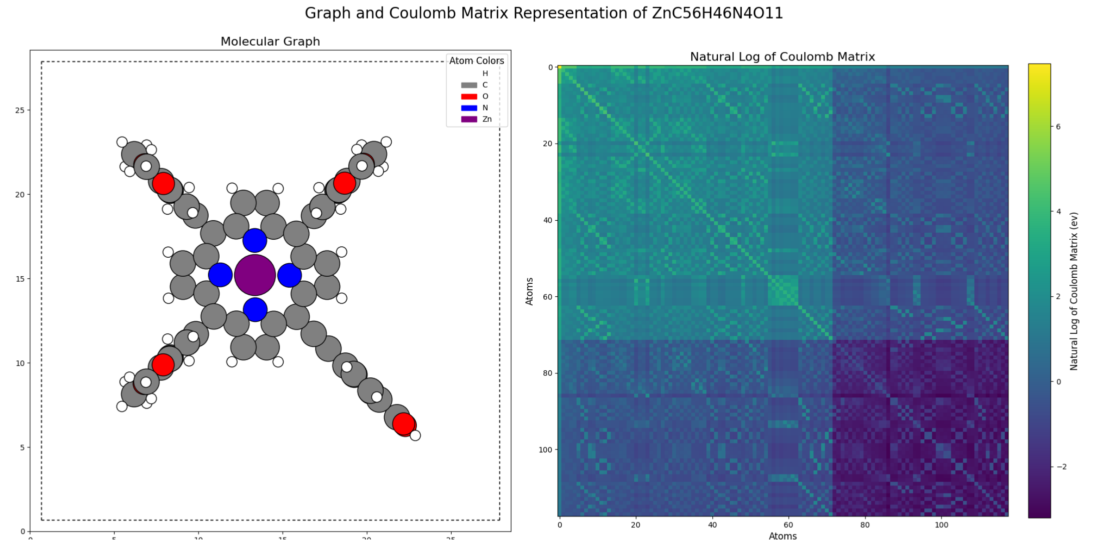

Figure 1: An example of PBD molecular structure and the natural logarithm of its associated Coulomb Matrix. (The natural logorithm is used purely for visulization purposes and is not part of our modeling process.)

2 Methods

GNN Implementation

Data Preparation

We convert each PBD (Poly-Benzimidazole Derivative) molecule to a graph by first constructing the Coulomb Matrix (CM), in which every element, M_{ij}, represents the mutual Coulomb interaction between every pair of nuclei, in units of electron volts (eV). It takes the form of Equation 1.

$$M_{ij} = \begin{cases} \frac{1}{2} Z_i^{2.4} & \text{for } \; i = j \\ \frac{Z_i Z_j}{R_{ij}} & \text{for } \; i \neq j \end{cases} \tag{1}$$

Here Z_i is the atomic number and R_{ij} is the distance between nuclei i and j in Angstroms. The diagonal of the Coulomb matrix governs the *self-interaction* for each nucleus, while the off-diagonal entries govern Coulomb repulsion between pairs of nuclei. This representation encodes similar information to the Hamiltonian used in electronic structure calculations, providing a compact description of molecular geometry and composition. Furthermore, it's invariant under translation, rotation, and permutation of atoms, while uniquely encoding the system Rupp et al. (2012), making it an ideal representation for machine learning models. In this project, the CM is generated using the DScribe library Himanen et al. (2020) Laakso et al. (2023). Figure 1 illustrates a molecule in real space and its corresponding CM.

To increase efficiency in model training and to highlight relevant interactions, we exclude any M_{ij} elements with absolute values less than 1 eV, as these represent weak interactions with minimal contribution to molecular properties. We use the remaining values in the lower triangular portion of the CM as the edge attributes in each molecule's graph representation.

Next, we standardize the target features (energy, band gap, and KS-gap) individually to have zero mean and unit standard deviation. Each feature's rescaling factors are stored to later convert predictions back to the original feature space.

Model Architecture

Our GNN model is composed of three successive layers of graph convolutions, each followed by batch normalization and a rectified linear unit (ReLU) activation function. The output from these layers goes through average pooling, which yields a vector (h_G) for each molecule. Finally, h_G is sent to three separate two-layer neural networks in order to arrive at predictions ($\hat{y}_E, \hat{y}_{BG}, \hat{y}_{KS}$) for the energy, band gap and KS-gap. We include the complete code in the submission.

Physics-Informed Loss Function

Enforcing physical constraints in neural network loss functions can significantly improve model performance Nagai et al. (2022). Due to the simultaneous prediction of energy, band gap, and KS-gap by a single model, we have the opportunity to incorporate physics priors into the loss function to discourage physically unrealistic predictions. We test two adaptations of the standard loss function.

Loss Function Constraint I: Band Gap Bounding KS-gap

As shown in Figure 2, the band gap (E_{gap}) and KS-gap are correlated, with the band gap value consistently larger than the KS-gap, as indicated by the linear fit with slope less than 1. To embed this relationship into the loss function, we rescale the raw predictions (which are standardized) back to the original feature space, then apply Equation 2, which is added to the total loss. Here, $\lambda_{BG,KS}$ is a scalar hyperparameter that controls the impact of this constraint and $\hat{y}_{KS}, \hat{y}_{BG}$ are the model's scaled predictions. This equation utilizes the ReLU function, which outputs 0 for negative values and returns the input value otherwise.

$$\text{loss}_{BG,KS} = \lambda_{BG,KS} \cdot \text{ReLU}(\hat{y}_{KS} - \hat{y}_{BG}) \tag{2}$$

Loss Function Constraint II: Multivariate Normal

If the target variables are normally distributed with covariance matrix Σ, the entire dataset can be modeled according to a multivariate normal probability density function $\mathcal{N}(\vec{y},\ \vec{\mu},\ \Sigma)$. Here $\vec{\mu}$ is a vector of the means of the individual variables found from a Gaussian fit. The predictions $\vec{y}_{pred}$ made by the model can then be further constrained based on known correlations by minimizing the negative log-likelihood of this multivariate normal distribution Zheng and Sun (2025). This is implemented by adding the following term to the loss function:

$$\mathcal{L}_{Normal} = -\lambda_n \ \ LLH(\mathcal{N}(\vec{y}_{pred},\ \vec{\mu},\ \Sigma)). \tag{3}$$

Here λ_n is another hyperparameter that controls the strength of this constraint and $\vec{y}_{pred} = [\hat{y}_E, \hat{y}_{BG}, \hat{y}_{KS}]$. To implement this method, prior to training, we calculate the multivariate distribution parameters ($\vec{\mu},\ \Sigma$) using the training data. During training, we evaluate the predictions with Equation 3 and add this term to the regular loss function.

Model Training

Our final loss function (4), can then can be tailored to enforce the desired coinstraints by changing the $\lambda_{BG,KS}$ and λ_n parameters. Here, the first term is the standard Mean Squared Error (MSE) function used to enforce accuarate predictions. During training, we used the Adaptive Moment Estimation (Adam) optimizer and a learning rate scheduler to help stabilize the optimization process. Additionally, we performed a brief hyperparameter search to obtain reasonable results, and settled on a node-embedding hidden dimension of 64, a learning rate of 0.005, 150 training epochs, and a batch size of 32.

We divided the PBD dataset such that 75% was used for training (n=9,072 samples), 12.5% for testing (n=1,512), and 12.5% for validation (n=1,512). Training and test loss were monitored during optimization to prevent overfitting. An example training loss plot is provided in the Appendix.

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \sum_{f\in\{E,BG,KS\}} (y_{i,f} - \hat{y}_{i,f})^2 + \lambda_{BG,KS} \cdot \text{ReLU}(\hat{y}_{KS} - \hat{y}_{BG}) - \lambda_n \cdot LLH(\mathcal{N}(\vec{y}_{\text{pred}}, \vec{\mu}, \Sigma)) \tag{4}$$

	Energy (MSE)	Energy R^2	Band-Gap (MSE)	Band-Gap R^2	KS-Gap (MSE)	KS-Gap R^2
Base Model	766.33	0.967	0.0344	0.821	0.0282	0.721
Constraint I	754.77	0.971	0.0309	0.844	0.0262	0.742
Constraint II	832.99	0.968	0.0332	0.832	0.0290	0.715

Table 1: Comparison of Model Performance

3 Model Performance

We ran each model with the same train, test, and validation data, using a learning rate of 0.005, batch size of 32, and 150 training epochs. The only deviation came from the loss function configuration, where we set $\lambda_{BG,KS} = 0, \lambda_n = 0$ for the base model, $\lambda_{BG,KS} = 0.2, \lambda_n = 0$ for Constraint I, and $\lambda_{BG,KS} = 0, \lambda_n = 0.2$ for Constraint II.

We present the three models' predictions in Figure 3 as well as the mean square error (MSE) and the coefficient of determination (R^2) for our three models in Table 1. The comprehensive loss function used across all models is shown in Equation 4, with different constraint terms activated based on the model variant.

Our results indicate that incorporating the physical constraint between band gap and KS-gap (Constraint I) yielded a consistent improvement across all target variables, with reductions in MSE and increases in R^2 values. The most significant improvements were observed for the band gap prediction, where the R^2 increased from 0.821 to 0.844, representing a 2.8% improvement in explained variance.

In contrast, the multivariate normal-based approach (Constraint II) showed slightly decreased performance compared to the base model. We hypothesize that this might result from the complex model requiring longer training time to converge properly. Additionally, it may suggest that the multivariate Gaussian distribution poorly captures the true underlying data distribution and thus hinders performance.

Discussion

3.1 Enforcing Physical Correlations

In a standard neural network, each target variable x_i has an associated loss function $\mathcal{L}_i$, and the model minimizes the sum of these losses: $\mathcal{L} = \mathcal{L}_1(x_1) + \mathcal{L}_2(x_2) + (\cdots)$. Taking the gradient of this expression yields a sum of individual derivatives: $\nabla\mathcal{L} = \partial_{x_1}\mathcal{L}_1(x_1) + \partial_{x_2}\mathcal{L}_2(x_2) + (\cdots)$. While neural networks can typically identify strong correlations using this approach, weaker correlations may be missed due to the independent treatment of target variables.

Our physics-informed constraint between band gap and KS-gap demonstrates how domain knowledge can be effectively incorporated into the loss function. By explicitly enforcing known physical relationships, we improved prediction accuracy across all target variables. This suggests that even when training a multi-target neural network, explicitly modeling physical constraints provides valuable inductive bias that guides the optimization process toward more physically plausible solutions.

3.2 Gaussian Assumption

Using the multivariate normal constraint relies on the assumption that target variables follow a multivariate normal distribution. From the histogram fits in Figure 4 in the appendix, we observed that a standard Gaussian model achieves p-values only slightly above 0.9. Using a skewed Gaussian distribution (Equation 8 in the Appendix) yields p-values > 0.95, indicating that the target variables deviate from perfect normality.

This deviation from normality may explain the underperformance of the multivariate normal constraint. The skew could result from non-random sampling of the dataset, where molecules with

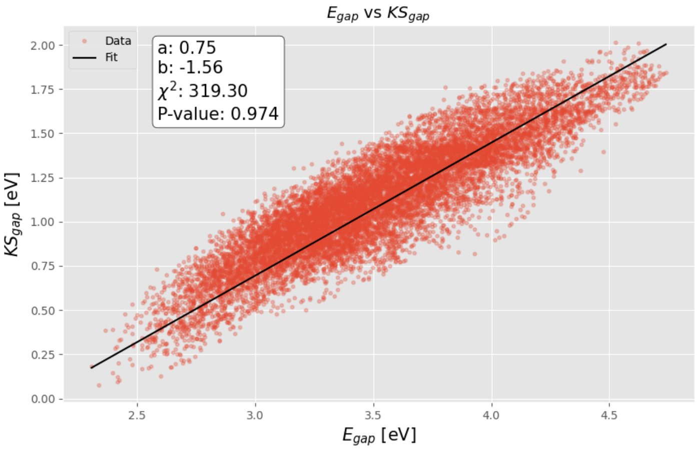

Figure 2: Relation between KS_{gap} and E_{gap} for the data. Fitted with a linear function using Pearson χ^2.

certain property distributions were preferentially selected. When the underlying assumptions of a statistical model are violated, enforcing them can lead to suboptimal performance, as observed in our results.

4 Conclusion

In this work, we have developed a Graph Neural Network (GNN) model for predicting multiple quantum mechanical properties of PBD molecules simultaneously. Our approach focused on incorporating physical and statistical constraints into the loss function to improve model performance. We demonstrated that the explicit incorporation of physical relationships between band gap and KS-gap consistently improved model performance, with MSE reductions across all target variables and R^2 increases of up to 2.8% for band gap predictions. Further, our results indicate that physics-informed constraints provide valuable inductive bias that guides the optimization process toward more physically plausible solutions, even when the relationships might be partially captured by standard neural network training. Statistical constraints based on multivariate normal distributions were less effective, likely due to the slight deviations from normality in our target variables, highlighting the importance of validating distributional assumptions before applying such constraints.

Future work could explore alternative distribution models that better capture the non-Gaussian nature of quantum mechanical properties, more complex physical constraints incorporating additional domain knowledge, adaptive weighting schemes for constraint terms, and including node features in our molecular graphs. The improved accuracy in predicting molecular properties demonstrated here could potentially accelerate materials discovery by reducing the need for expensive computational simulations or experimental validation.

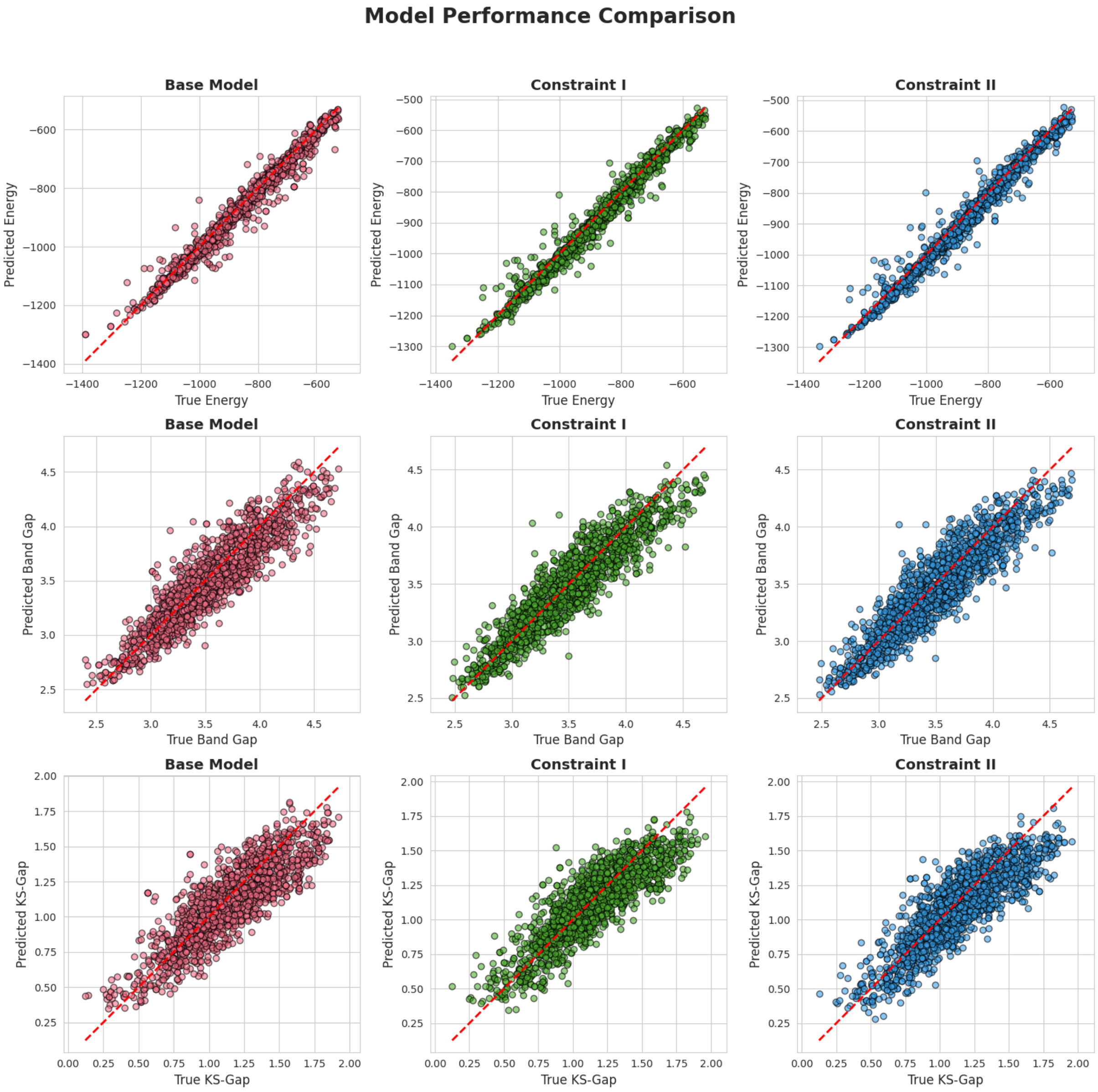

Figure 3: Comparison of Model Performance with Various Constraints in Loss Function

5 Appendix

6 Graph Neural Network Background

Graph Neural Networks (GNNs) extend deep learning methods to graph-structured data by iteratively updating node representations through message passing. A typical graph convolution operation incorporating edge features can be represented as Equation 5 Gilmer et al. (2017).

$$h_i^{(l+1)} = \sigma\left(W^{(l)}h_i^{(l)} + \sum_{j\in\mathcal{N}(i)} \frac{1}{c_{ij}} W_{\text{neighbor}}^{(l)} h_j^{(l)} \odot \phi(e_{ij})\right) \tag{5}$$

Here, $h_i^{(l)}$ is the feature vector of node i at layer l, $\mathcal{N}(i)$ represents the set of neighbors of node i, and c_{ij} is a normalization constant. The terms $W^{(l)}$ and $W_{\text{neighbor}}^{(l)}$ are learnable weight matrices, while e_{ij} denotes the edge features between nodes i and j. The function ϕ transforms edge features, $\odot$ represents element-wise multiplication, and σ is a nonlinearity function (e.g., ReLU, sigmoid, or hyperbolic tangent). This message-passing mechanism allows information to propagate between neighboring nodes, incorporating both node and edge attributes into the learned representation.

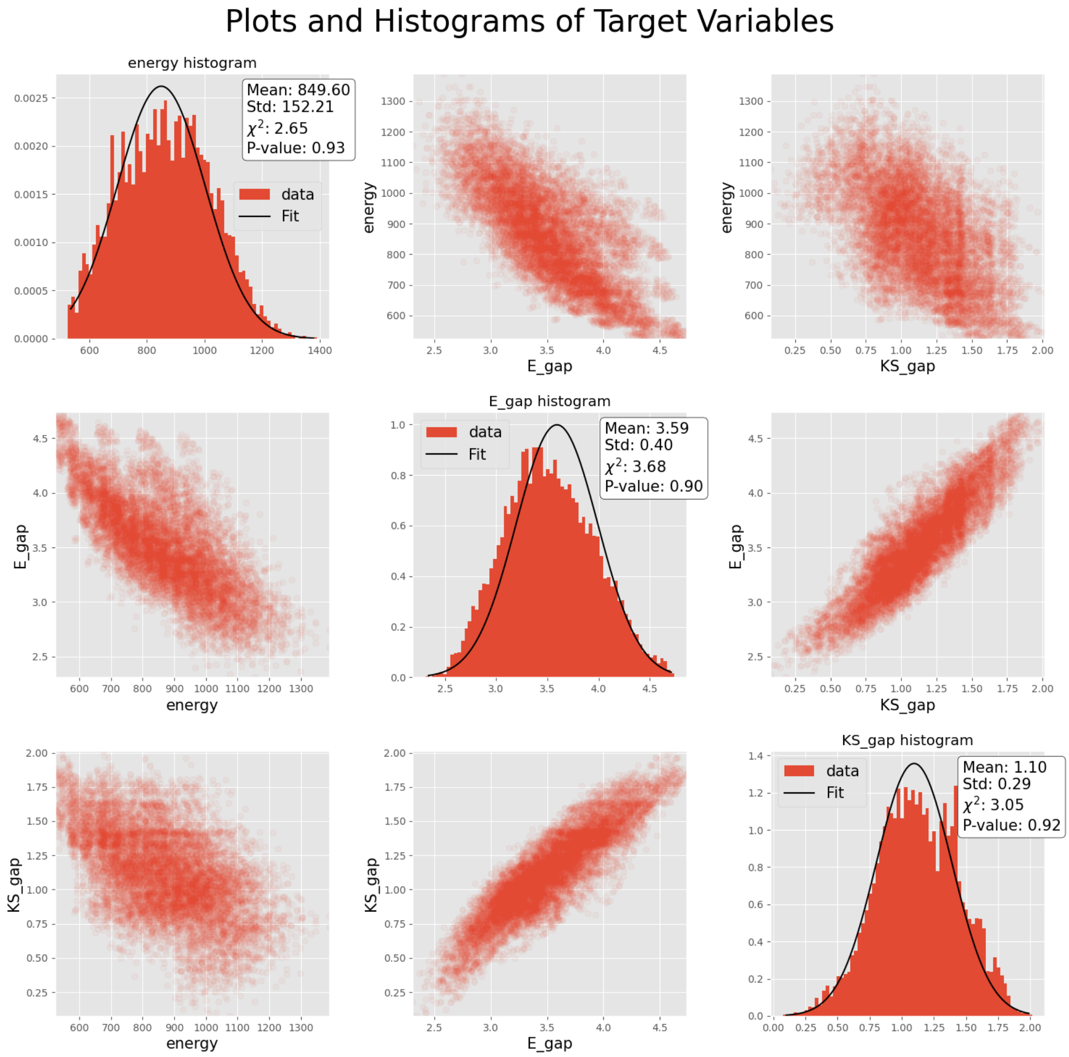

Figure 4: Overview of target variables distributions. The diagonal shows a normalized histogram of each variable, fitted with a normal distribution to get mean and standard deviation. The off-diagonals shows the correlations between different target variables.

Similar to convolutional kernels in Convolutional Neural Networks (CNNs), this operation refines node embeddings layer by layer, enabling deeper network representations. To obtain a graph-level prediction, the final-layer node embeddings ($h_i^{(L)}$) are aggregated through a pooling function—commonly mean, max, or a learnable transformation—yielding a single vector representation of the graph, h_G. This aggregated representation is then passed through a Multi-Layer Perceptron (MLP) to produce the final prediction, as expressed in Equation 6.

$$\hat{y} = \text{MLP}(h_G) = W^{(out)} \cdot \text{ReLU}(W^{(hidden)} h_G + b^{(hidden)}) + b^{(out)} \tag{6}$$

GNN models are often trained by minimizing the mean squared error (MSE) loss between the predicted values $\hat{y}_i$ and the true target values y_i, as defined in Equation 7, where N represents the total number of training samples.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{7}$$

This formulation enables the GNN to learn meaningful graph representations while preserving structural and relational information from the input molecular graphs.

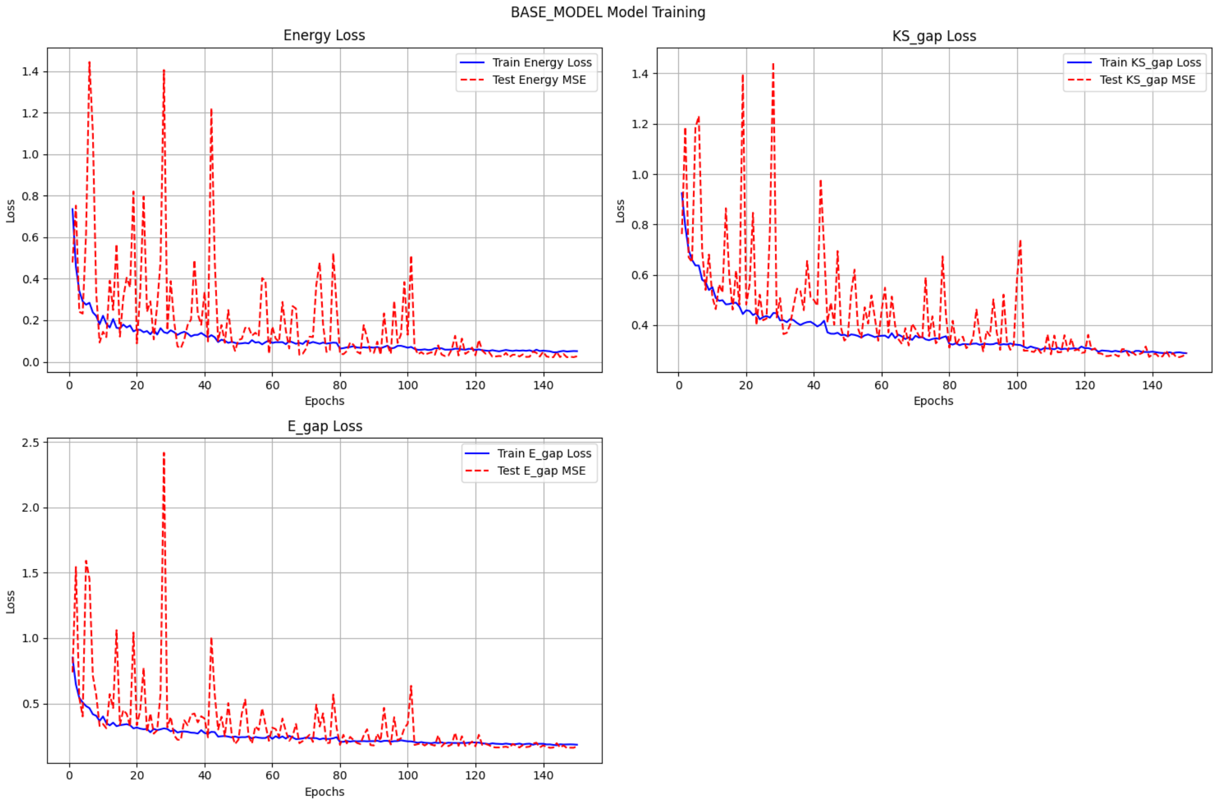

Figure 5: Example of Standard GNN Training Loss.

6.1 Skewed Gaussian

The skewed gaussian distribution is defined in Equation 8. Here p_{norm} is a normal gaussian PDF, and $\Phi(x)$ is the cumulative distribution function of p_{norm}. μ and σ are the mean and standard deviation of the normal gaussian, and α is a parameter controlling the amount of *skew*.

$$p_{skew}(x, \alpha, \mu, \sigma) = 2p_{norm}(x, \mu, \sigma) \cdot \Phi(\alpha x) \tag{8}$$

References

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1263–1272. PMLR.

Himanen, L., Jäger, M. O. J., Morooka, E. V., Federici Canova, F., Ranawat, Y. S., Gao, D. Z., Rinke, P., and Foster, A. S. (2020). DScribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949.

Laakso, J., Himanen, L., Homm, H., Morooka, E. V., Jäger, M. O., Todorović, M., and Rinke, P. (2023). Updates to the dscribe library: New descriptors and derivatives. *The Journal of Chemical Physics*, 158(23).

Li, Z., Li, Q., Li, C., and Xie, Y. (2024). Panchromatic porphyrin-based dye-sensitized solar cells: from cosensitization to concerted companion dye approaches. *Materials Chemistry Frontiers*, 8(3):652–680.

Nagai, R., Akashi, R., and Sugino, O. (2022). Machine-learning-based exchange correlation functional with physical asymptotic constraints. *Phys. Rev. Res.*, 4:013106.

Ornsø, K., García-Lastra, J. M., and Thygesen, K. (2013). Computational screening of functionalized zinc porphyrins for dye sensitized solar cells. *Physical chemistry chemical physics : PCCP*, 15.

Reiser, P., Neubert, M., Eberhard, A., et al. (2022). Graph neural networks for materials science and chemistry. *Communications Materials*, 3:93.

Rupp, M., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. (2012). Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108:058301.

Schütt, K. T., Kindermans, P.-J., Sauceda, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. (2017). Schnet: A continuous-filter convolutional neural network for modeling quantum interactions.

Zheng, V. Z. and Sun, L. (2025). Mvg-crps: A robust loss function for multivariate probabilistic forecasting.