

Article

Optimizing GNN Architectures Through Nonlinear Activation Functions for Potent Molecular Property Prediction

Areen Rasool¹, Jamshaid Ul Rahman¹  and Quaid Iqbal^{2,*}

¹ Abdus Salam School of Mathematical Sciences, GC University, Lahore 54600, Pakistan; areen_rasool_22@sms.edu.pk (A.R.); jamshaid@sms.edu.pk (J.U.R.)

² Department of Mathematics and Statistics, Binghamton University—State University of New York, Binghamton, NY 13902, USA

* Correspondence: qiqbal@binghamton.edu

Abstract: Accurate predictions of molecular properties are crucial for advancements in drug discovery and materials science. However, this task is complex and requires effective representations of molecular structures. Recently, Graph Neural Networks (GNNs) have emerged as powerful tools for this purpose, demonstrating significant potential in modeling molecular data. Despite advancements in GNN predictive performance, existing methods lack clarity on how architectural choices, particularly activation functions, affect training dynamics and inference stages in interpreting the predicted results. To address this gap, this paper introduces a novel activation function called the Sine Linear Unit (SLU), aimed at enhancing the predictive capabilities of GNNs in the context of molecular property prediction. To demonstrate the effectiveness of SLU within GNN architecture, we conduct experiments on diverse molecular datasets encompassing various regression and classification tasks. Our findings indicate that SLU consistently outperforms traditional activation functions on hydration free energy (FreeSolv), inhibitory binding of human β secretase (BACE), and blood brain barrier penetration (BBBP), achieving the superior performance in each task, with one exception on the GCN model using the QM9 data set. These results underscore SLU's potential to significantly improve prediction accuracy, making it a valuable addition to the field of molecular modeling.

Keywords: chemical graph theory; graph neural networks; molecular graph; molecular property prediction; Sine Linear Unit



Citation: Rasool, A.; Rahman, J.U.; Iqbal, Q. Optimizing GNN Architectures Through Nonlinear Activation Functions for Potent Molecular Property Prediction.

Computation **2024**, *12*, 212.

<https://doi.org/10.3390/computation12110212>

Academic Editor: Paul Popelier

Received: 28 September 2024

Revised: 15 October 2024

Accepted: 18 October 2024

Published: 22 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Chemistry relies heavily on the creation of accurate molecular models. Traditionally, this has been achieved through methods like IUPAC names, formulas, and various structural drawings. Glycerol, for instance, has the IUPAC name propane-1,2,3-triol. Figure 1 illustrates its skeletal, structural, and molecular formulas. However, these representations cater primarily to human comprehension and are not directly usable by machine learning algorithms. To bridge this gap and empower computers to understand and utilize molecular data, Molecule Representation Learning (MRL) emerges. MRL tackles this challenge by transforming molecules into a low-dimensional space, represented as dense vectors (embedding). These learned embeddings unlock a vast array of applications, including organic reaction prediction [1,2], predicting molecular property [3], molecule generation [4], drug detection [5], planning chemical syntheses [6], chemical text mining [7], and knowledge graph prediction [8].

Several MRL methods have been proposed by researchers, such as MolBERT [9], ChemBERTa [10], SMILESTransformer [11], SMILES-BERT [12], Molecule-Transformer [13], and SABiLSTM [14]. These methods predominantly utilize SMILES strings as input and employ natural language models like Transformers [15] or BERT [16] as their foundation model. SMILES, or Simplified Molecular-Input Line-Entry System, is a powerful tool for

chemists. It allows them to represent complex molecular structures using short, easy-to-read strings of characters. For example, the SMILES string "OCC(O)CO" accurately describes the structure of glycerol.

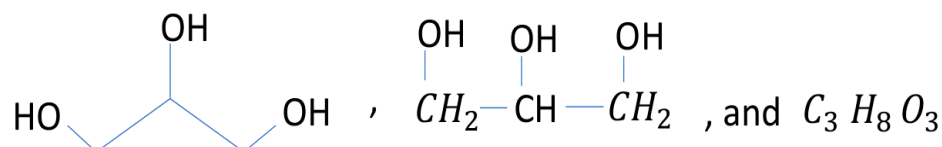


Figure 1. Skeletal, structural, and molecular formulas of glycerol.

While powerful language models excel at processing sequential data, their performance suffers when dealing with SMILES strings, a 1D representation of molecular structures. This linear format limits their ability to capture the inherent complexities of molecular graphs. Further details can be found in [17]. Presently, a number of computationally favorable deep learning [18] techniques have been introduced for processing data and representing molecules as graphs. Deep learning architectures have become the foundation for many state-of-the-art MRL methods. This is exemplified by recent work in simulating oscillators and biochemical systems using deep neural networks [19,20]. Understanding these deep learning architectures is vital for researchers in MRL. Various MRL methods leverage GNNs [21] for processing molecules as graphs [22,23].

GNNs are revolutionizing how we understand molecules. Unlike traditional methods, GNNs treat molecules as graphs, where atoms are nodes and bonds are edges. This approach effectively captures the complex, non-Euclidean nature of molecular structures. GNNs boast exceptional flexibility, working seamlessly with various molecule representations, including chemical graphs, 3D structures, and more. Fueled by research across various domains, GNNs have found a valuable application in chemistry, particularly for analyzing molecules [24]. GNN models have consistently surpassed traditional methods in a critical task for drug discovery: predicting molecular properties [25–28]. This success highlights the immense potential of GNNs in accelerating drug development. A major challenge in drug discovery is predicting a molecule's properties. This challenge has attracted growing attention in recent years [29–32], leading to the development of various GNN architectures specifically designed to excel at molecular property prediction tasks in the field of chemistry.

In the realm of MRL, there isn't a single "one-size-fits-all" GNN architecture that can effectively predict all types of molecular properties. Different tasks require specific GNN architectures to achieve the best performance. The search for the ideal GNN architecture for property prediction in MRL is an ongoing challenge. The goal is not just to create a single architecture that outperforms all others, but to develop GNNs that are inherently flexible and capable of addressing the diverse nature of molecular properties.

While standard GNN architectures possess significant capabilities, their effectiveness can be limited by design choices that restrict their ability to capture the nuances of various molecular properties. Additionally, while the architecture of a model is important, GNN performance is also heavily influenced by hyperparameters. Unfortunately, these hyperparameters are often not thoroughly optimized, which can introduce variability in performance that is independent of the architecture itself. In GNN architectures, hyperparameters play a crucial role, and the choice of activation function is especially significant. Understanding how different activation functions affect the learning process and the complexity they can capture is essential for optimizing GNN performance in molecular property prediction tasks. By selecting an activation function that encourages the learning of a wide range of molecular features, captures non-linear relationships between atoms, and regulates information flow, researchers can create GNNs that are more adaptable to various property prediction tasks. Traditional activation functions like Rectified Linear Unit (ReLU) [33], LeakyReLU [34], and Amplifying Sine Unit (ASU) [35] are effective in many situations, but

they may not be ideal for capturing complex relationships within molecules. For example, ReLU neurons can become inactive during training when they output zero for all inputs. Moreover, LeakyReLU poses challenges in determining the optimal slope for the negative input region. To address the issues related to dying ReLU and other limitations of existing activation functions, we introduce a novel activation function called the Sine Linear Unit (SLU). Our aim in developing this new activation function is to significantly enhance the capabilities of GNNs in managing the complexities associated with various molecular property prediction tasks.

Our main contributions in this paper are the introduction of the Sine Linear Unit (SLU), a novel activation function specifically designed to enhance the performance of GNNs for predicting molecular properties. We provide a rigorous mathematical analysis of SLU, demonstrating its advantageous characteristics that make it particularly effective for GNN architectures. Furthermore, we integrate SLU into various GNN architectures and conduct a comprehensive performance comparison against traditional activation functions using publicly available molecular datasets. Figure 2 presents a visual overview of our contributions.

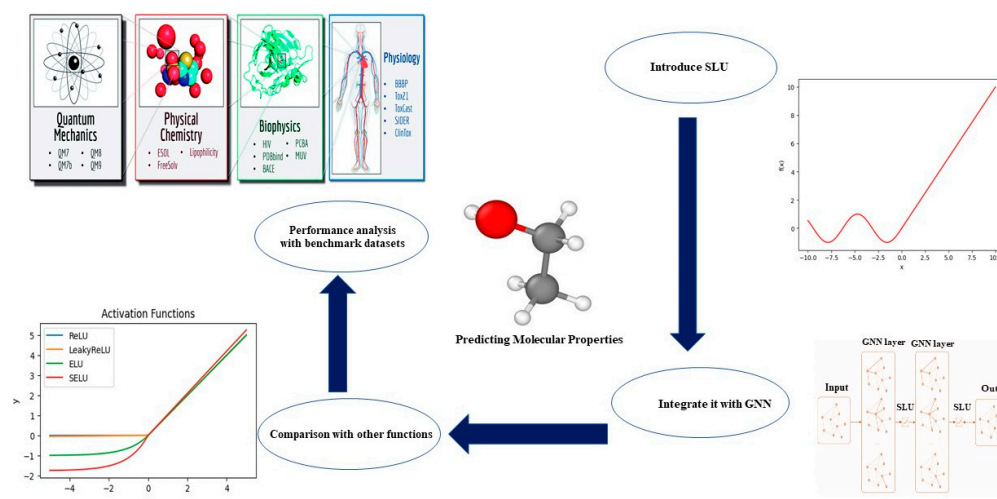


Figure 2. Graphical representation of key contributions.

2. Preliminaries on Molecular Representations and GNNs

In mathematical chemistry, chemical graph theory has been concerned with representing the arrangement of compounds using graphs since 1874 [36]. Modern chemical graph theory goes beyond just representing structure. It explores various mathematical properties of the graph, such as connectivity, cycles, and shortest paths, to gain insights into the molecule's behavior and properties. Typically, an undirected graph depicts the molecular structure, with atoms as nodes and chemical bonds as edges. Molecules can be easily visualized in this way, as shown in Figure 3. Considering a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where \mathcal{V} indicates the vertices or nodes, with each node $v_i \in \mathbb{R}^{d_v}$ representing an atom in molecular graphs. These nodes may possess various characteristics, including atomic number and chirality. The edges typically signify covalent bonds between the atoms. Each edge $e_{ij} \in \mathbb{R}^{d_e}$ is defined by a certain number of attributes, commonly indicating the type of bond.

GNNs are highly proficient in addressing tasks involving the categorization or estimation of graph properties across diverse hierarchical levels, showcasing modern efficiency.

- Graph-level tasks encompass a multitude of applications, such as forecasting distinct features across the entirety of the graph. This task can enclose activities like determining toxicity or carrying out regressions. Within this guide, we will focus on implementing regression tasks aimed at anticipating molecular properties. Another vital graph-level endeavor involves predicting entirely novel graphs or molecules, a

pivotal aspect, especially within the realm of drug discovery, which revolves around identifying new drug candidates.

- Tasks at the node level involve the ability to make predictions about a specific node within a graph, such as determining the atomic charges associated with each atom. Another aspect is the ability to foreseeing the addition of a new node into the graph. This is commonly seen in the context of molecule creation, where the goal is to sequentially incorporate multiple atoms to generate novel molecules.
- Edge-level tasks entail predicting edge characteristics, such as molecular forces or the formation of new connections within a graph. In the context of generating molecules, the goal is to speculating potential links between atoms. Edge forecasting also serves to deduce relationships or interactions, like those in a genetic regulatory network.

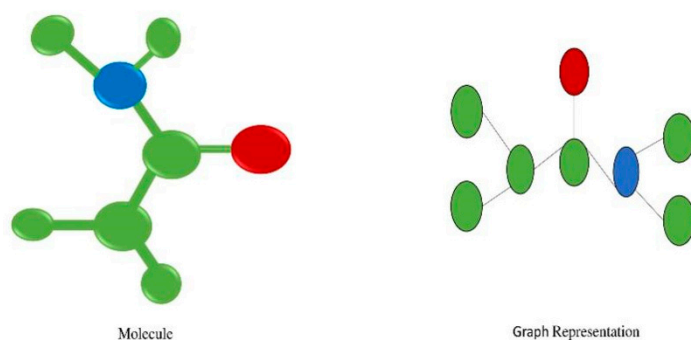


Figure 3. Graphical representation of a molecule as a graph.

In the realm of chemistry, most tasks revolve around projecting outcomes at the graph level. We will emphasize the prediction of molecular properties through the analysis of molecular structure. Molecular property prediction deals with two types of target properties: categorical properties (e.g., toxicity [37]) for classification and continuous properties (e.g., energy [38]) for regression. The predictive task aims to unravel the valuable properties of a molecule. Molecular property prediction takes a molecule c and its graph representation \mathcal{G}_c . GNNs leverage molecular graphs as input data, effectively capturing the essential features that govern the prediction of specific target properties.

GNN networks leverage both the molecular structure and atom attributes to generate a vector representation for each atom as well as the entire molecule. In GNNs, a prevalent approach is neighborhood aggregation, where an atom's representation is iteratively refined by integrating information from its neighboring atoms and its own features. Equation (1) formally defines the k -th layer of a GNN as follows.

$$h_i^k = \text{AGGREGATE} \left(\left\{ h_j^{k-1} \right\}_{j \in \mathfrak{N}(i) \cup \{i\}} \right), k = 1, \dots, K \quad (1)$$

The vector representation of atom a_i at the k -th layer is symbolized as h_i^k (where h_i^0 set as the initial feature x_i of a_i). The set of atoms directly connected to a_i is represented as $\mathfrak{N}(i)$, and K denotes the quantity of layers within the GNN architecture. The careful selection of the AGGREGATE function stands as a cornerstone in the architecture of GNNs, a fact underscored by the diverse array of GNN designs that have emerged.

Equation (2) defines the final step. This step involves utilizing a readout function to combine all node representations produced by the last GNN layer. This ultimately derives the complete representation of the entire molecule as h_G :

$$h_G = \text{READOUT} \left(\left\{ h_i^K \right\}_{a_i \in V} \right) \quad (2)$$

An illustration of the GNN encoder alongside molecular structure is presented in Figure 4.

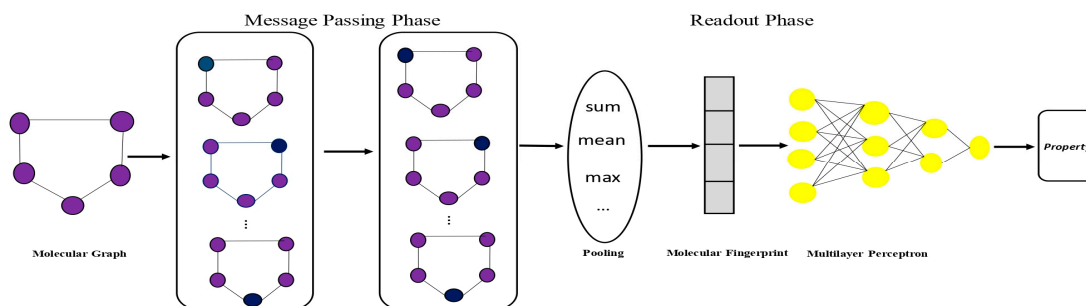


Figure 4. Overview of GNN framework along its core components.

With their widespread popularity and versatility, a diverse range of GNN architectures have emerged, including the versatile MPNN framework [22] and the SchNet architecture designed for quantum interactions [39]. Wieder et al. [40] provide an in-depth analysis of various GNN variants and their real-world applications across multiple fields.

The following section introduces a new activation function with a compelling set of characteristics. Notably, it overcomes the shortcomings of existing functions and offers the potential for superior performance in molecular property prediction, which we will discuss in detail later.

3. Sine Linear Unit

This section introduces a new activation function known as Sine Linear Unit (SLU), which is especially well-suited for deep learning networks. Mathematically, the function is formally defined as,

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \sin(x) & \text{if } x < 0 \end{cases} \quad (3)$$

aiming to offer superior properties compared to other activation functions. The function is straightforward, acting as the identity function when given positive input and as $\sin(x)$ when given negative input. The graphical representation of the SLU and its derivative can be found in Figure 5.

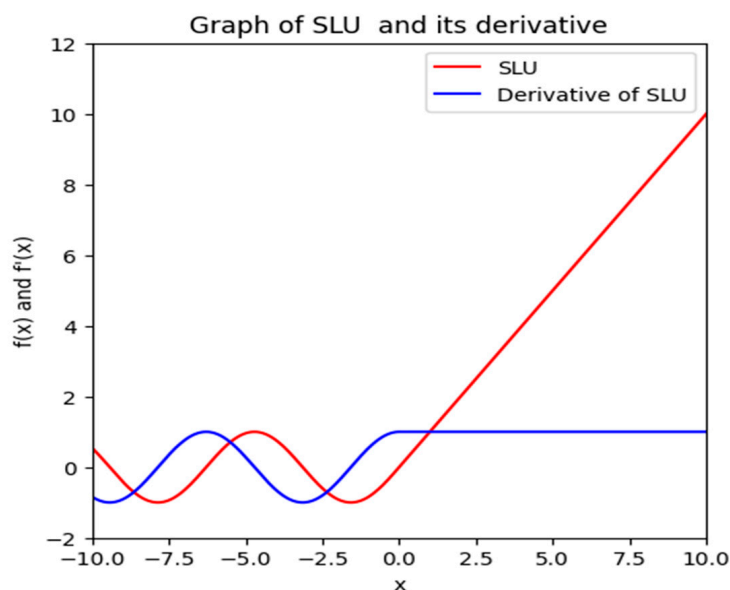


Figure 5. Graphical representation of SLU function.

Next, we examine certain characteristics of SLU that a suitable activation function should possess.

I. **Continuity:** In Figure 5, SLU is portrayed as a continuous activation function with a unique characteristic compared to traditional activation functions.

For $x \geq 0$:

- The function is defined as $f(x) = x$ for $x \geq 0$.
- The identity function $f(x) = x$ is continuous everywhere, including at $x \geq 0$.

For $x < 0$:

- The function is defined as $f(x) = \sin(x)$ for $x < 0$.
- The sine function $f(x) = \sin(x)$ is continuous for all real numbers, including $x < 0$.

As $f(x)$ is defined as continuous functions in both cases, it implies that $f(x)$ maintains continuity across all real numbers. Therefore, the function $f(x)$ is continuous throughout its domain, encompassing all real numbers \mathbb{R} .

II. **Differentiability:** The differentiability of $f(x)$ across its domain requires examining its differentiability at each point.

For $x > 0$:

- The function is $f(x) = x$, which is a polynomial function. Polynomials are differentiable everywhere, so $f(x)$ is differentiable for $x > 0$.

For $x < 0$:

- The function is $f(x) = \sin(x)$, which is also differentiable for all real numbers. The derivative of $f(x) = \sin(x)$ with respect to x is $\cos(x)$, and since $\cos(x)$ is defined for all real numbers, $f(x) = \sin(x)$ is differentiable for $x < 0$.

At $x = 0$:

- We need to check the differentiability at $x = 0$. To do this, we need to verify if the left-hand derivative and the right-hand derivative at $x = 0$ match.
- Left-hand derivative at $x = 0$:

$$\lim_{h \rightarrow 0^-} \frac{f(0+h) - f(0)}{h} = \lim_{h \rightarrow 0^-} \frac{\sin(h) - 0}{h} = \lim_{h \rightarrow 0^-} \frac{\sin(h)}{h} = 1 \text{ (by the Squeeze theorem)}$$

- Right-hand derivative at $x = 0$:

$$\lim_{h \rightarrow 0^+} \frac{f(0+h) - f(0)}{h} = \lim_{h \rightarrow 0^+} \frac{h - 0}{h} = 1$$

- Since both left-hand and right-hand derivatives at $x = 0$ exist and are equal, $f(x)$ is differentiable at $x = 0$.

Consequently, $f(x)$ exhibits differentiability throughout its entire domain. Expression (4) captures the first derivative of SLU.

$$f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ \cos(x), & \text{if } x < 0 \end{cases} \tag{4}$$

III. **Nonmonotonicity:** The function defined in (3) is non-monotonic because it doesn't consistently increase or decrease across its entire domain.

For $x \geq 0$:

$$f(x) = x$$

is a linear function. As x increases, $f(x)$ also increases. This portion of the function is monotonic.

For $x < 0$:

$$f(x) = \sin(x)$$

is a periodic function. It oscillates between -1 and 1 as x changes. Therefore, the function is not monotonic in this region because it doesn't consistently increase or decrease.

The function exhibits a shift in behavior at $x = 0$, transitioning from an increasing trend for $x \geq 0$ to an oscillatory pattern for $x < 0$. This change violates monotonicity across the entire domain. So, SLU exhibits a non-monotonic behavior, fluctuating in a more intricate way.

IV. **Boundedness:** It is noteworthy that SLU has a lower bound and an unbounded upper limit within the range of approximately -1 to infinity.

For Bounded Below:

- For $x \geq 0$, $f(x) = x$, which is non-negative for all $x \geq 0$.
- For $x < 0$, $f(x) = \sin(x)$, which ranges between -1 and 1 , inclusive.
- Therefore, $f(x) \geq -1$ for all x in its domain, indicating that $f(x)$ is bounded below by -1 .

For Not Bounded Above:

- For $x \geq 0$, $f(x) = x$, which increases without bound as x approaches positive infinity.
- For $x < 0$, $f(x) = \sin(x)$, which oscillates between -1 and 1 , and there is no upper limit as x approaches negative infinity.
- Therefore, $f(x)$ is not bounded above because it can take arbitrarily large positive values for $x \geq 0$ and oscillates between -1 and 1 for $x < 0$.

SLU, an unbounded above activation function, has the advantage of avoiding saturation which can slow down training by causing near-zero gradients. This can have a drastic impact on the efficiency of the learning process. Additionally, SLU being bounded below also brings benefits as it leads to strong regularization effects. This means that using SLU can help prevent overfitting and improve the model's generalization abilities.

V. **Vanishing gradient:** A good activation function is crucial for deep learning networks because it should not cause the gradient to vanish, as this can hinder the training process. SLU fulfills this requirement by guaranteeing the gradient's stability. This makes SLU a solid choice for deep learning networks, as it allows for effective training without facing the issue of vanishing gradients.

VI. **Non linearity:** Unlike traditional activation functions like ReLU or LeakyReLU, SLU offers a unique method for introducing non-linearity into neural networks.

Overall, SLU unique properties make it a valuable tool in training deep learning models effectively. This innovative approach aims to improve the efficacy of GNNs to accurately predicting a molecule's properties based on its structure.

4. Experiments

We perform experiments on four molecular property benchmarks from MoleculeNet [28], comprising small datasets that lack spatial information (like FreeSolv, BACE, and BBBP) and big datasets (like QM9) with spatial information. Following, we provide a concise description of each tasks.

- The FreeSolv dataset is designed for regression applications, specifically for estimating the hydration free energy of 642 small molecules in water. The SMILES format used to depict the molecules is devoid of spatial information.
- The QM9 dataset, which includes 133,885 molecules with 12 target attributes, is used for problems involving the regression of quantum mechanical parameters. The dataset's exact spatial information for each molecule makes it perfect for applications like chemical property prediction based on quantum chemistry computations. Out

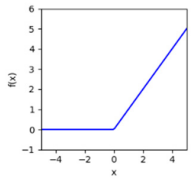
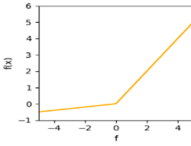
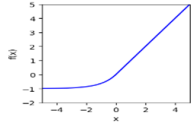
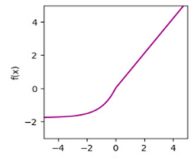
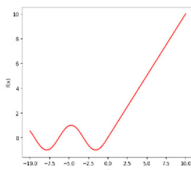
of the diverse regression targets within the dataset, we will specifically address the prediction of the dipole moment μ .

- BACE, a binary classification task for predicting the inhibition of human β -secretase (BACE), an enzyme linked to Alzheimer's disease. The dataset contains 1513 molecules represented in SMILES format and is used to evaluate potential inhibitors for drug discovery.
- BBBP (Blood-Brain Barrier Penetration) is used for binary classification tasks, predicting whether a compound can penetrate the blood-brain barrier. It contains 2039 molecules labeled as either permeable or non-permeable to the blood-brain barrier.

The dataset was split into three subsets to enable robust model evaluation: 80% for training, which lets the model learn from most of the data; 10% for validation, which helps to fine-tune and choose the best model; and the remaining 10% for testing, which evaluates the model's generalization performance.

To leverage the power of the SLU function for predicting properties, a key factor in molecular interactions, this study explores its integration with GNN models. The success of an existing methods, Graph Isomorphism Network (GIN) [41] and Graph Convolutional Network (GCN) [21], will be used to evaluate how well the SLU function performs within these architectures in predicting molecular properties. A variety of activation functions, listed in Table 1, are used to evaluate the SLU function's performance. Our final evaluation is based on the lowest RMSE for FreeSolv, the lowest MAE for QM9, and the highest ROC-AUC for BACE and BBBP, where RMSE, MAE, and ROC-AUC refer to the root mean square error, mean absolute error, and receiver operating characteristic area under the curve, respectively.

Table 1. List of activation functions used in evaluation.

Activation Function	Mathematical Expression	Graphical Representation
ReLU	$\begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$	
LeakyReLU	$\begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases}$	
Exponential Linear Unit (ELU)	$\begin{cases} x & \text{if } x \geq 0 \\ \alpha(\exp(x) - 1) & \text{if } x < 0 \end{cases}$	
Scaled Exponential Linear Unit (SELU)	$\lambda \begin{cases} x & \text{if } x \geq 0 \\ \alpha(\exp(x) - 1) & \text{if } x < 0 \end{cases}$	
Sine Linear Unit (SLU)	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \sin(x) & \text{if } x < 0 \end{cases}$	

4.1. Experimental Results on FreeSolv

Table 2 presents the regression outcomes of GNN models applied to the FreeSolv dataset, highlighting their performance with different activation functions. SLU stands out as the clear leader, achieving the lowest RMSE scores of 1.8433 in GIN and 1.9702 in GCN. While SLU may not have achieved the absolute lowest score, it significantly outperformed traditional activation functions with notable improvements. SLU surpassed ReLU with improvements of 0.7847 for GIN and 0.8311 for GCN, demonstrating a considerable reduction in error. These results underscore SLU’s effectiveness in accurately predicting hydration energy.

Table 2. RMSE score of GNN models on FreeSolv.

Model	GIN	GCN
ReLU	2.6280	2.8013
LeakyReLU	2.5732	2.2138
ELU	2.4384	2.3141
SELU	2.0215	2.2992
SLU	1.8433	1.9702

Figure 6 illustrates how the GNN model’s training loss curves with various activation functions show diverse performance patterns. As can be seen in Figure 6, the SLU activation function is better at learning from the training data on both model evidence as its constantly reduced training loss over all epochs. Unlike other activation functions, SLU efficiently reduces the loss and exhibits exceptional stability across the training process. There are some discrepancies in other functions with noticeable increases in training loss at the conclusion. As seen in Figure 7, SLU stands out in the validation phase with a consistent decline in validation loss, exhibiting good generalization to unseen data, making it more reliable than alternatives. Overall, SLU shows strong performance in both models trained on the FreeSolv dataset.

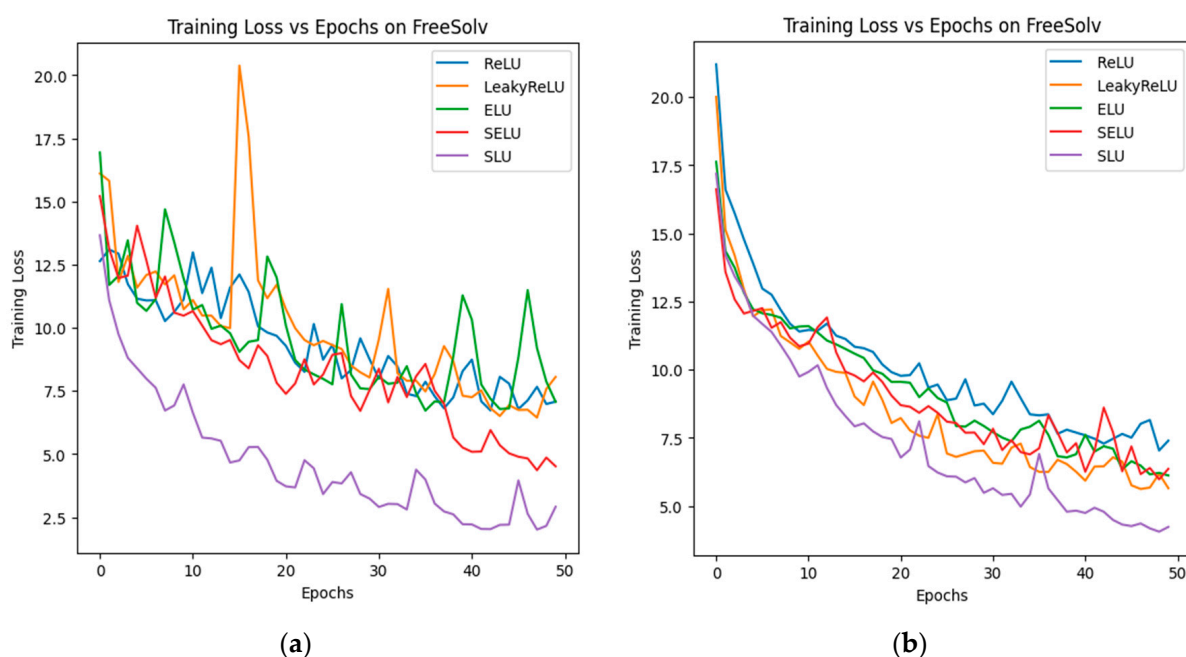


Figure 6. Training loss curves achieved via different activation functions on FreeSolv. (a) Training loss vs. epochs in GIN. (b) Training loss vs. epochs in GCN.

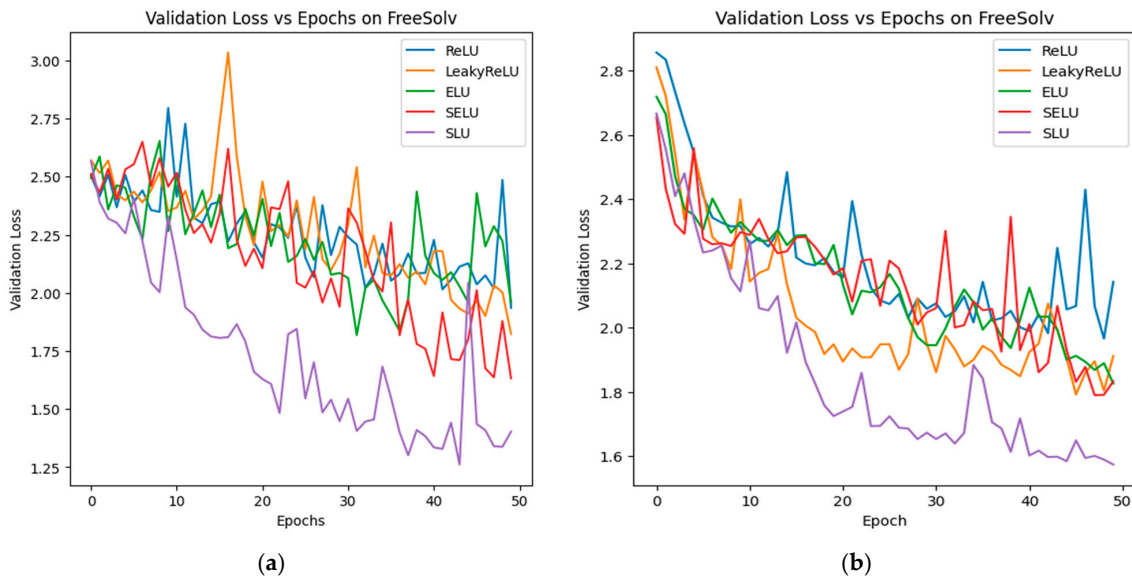


Figure 7. Validation loss curves achieved via different activation functions on FreeSolv. (a) Validation loss vs. epochs in GIN. (b) Validation loss vs. epochs in GCN.

4.2. Experimental Results on QM9

Figure 8 provides important insights into how different activation functions perform in the GIN and GCN models. In Figure 8a, the SLU activation function shows a clear advantage, with its training loss curve consistently decreasing across all epochs, indicating effective learning. In contrast, the training loss curves for other activation functions, especially ReLU, show smaller improvements or fluctuating patterns, suggesting they may have difficulty optimizing learning. Figure 8b presents the GCN model’s training curves, where SLU demonstrates a steady decrease in error with each epoch. However, despite this steady progress, SLU performs worse than ReLU and LeakyReLU in both training and validation loss. This is particularly evident in Figure 9b, where the validation loss curves show SLU struggling to generalize as well as ReLU and LeakyReLU. Even though SLU’s loss decreases, it does not outperform these functions, indicating that SLU might face challenges in optimizing GCN models as effectively as its counterparts. On the other hand, the validation loss curves in Figure 9a illustrate how well SLU performs in the GIN model and its ability to generalize to new data.

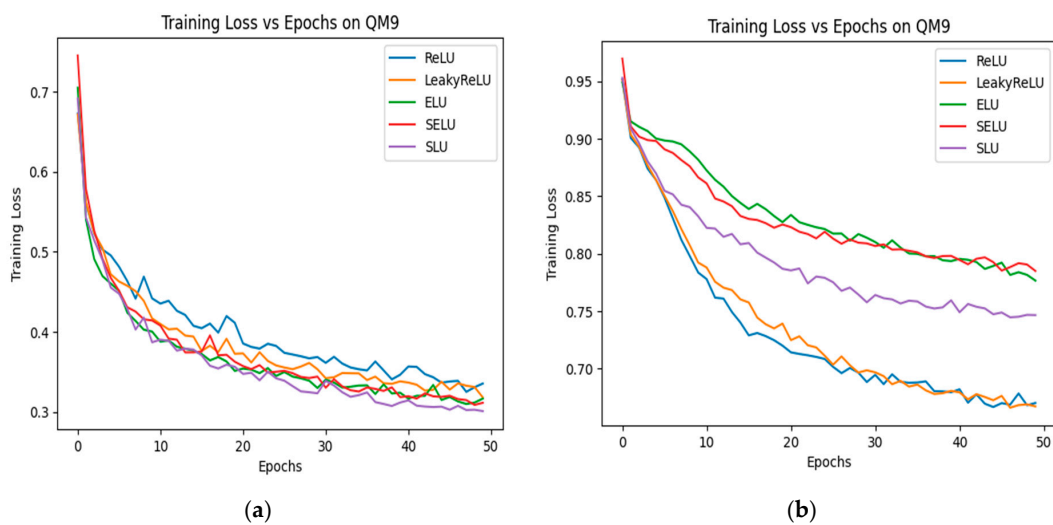


Figure 8. Training loss curves achieved via different activation functions on QM9. (a) Training loss vs. epochs in GIN. (b) Training loss vs. epochs in GCN.

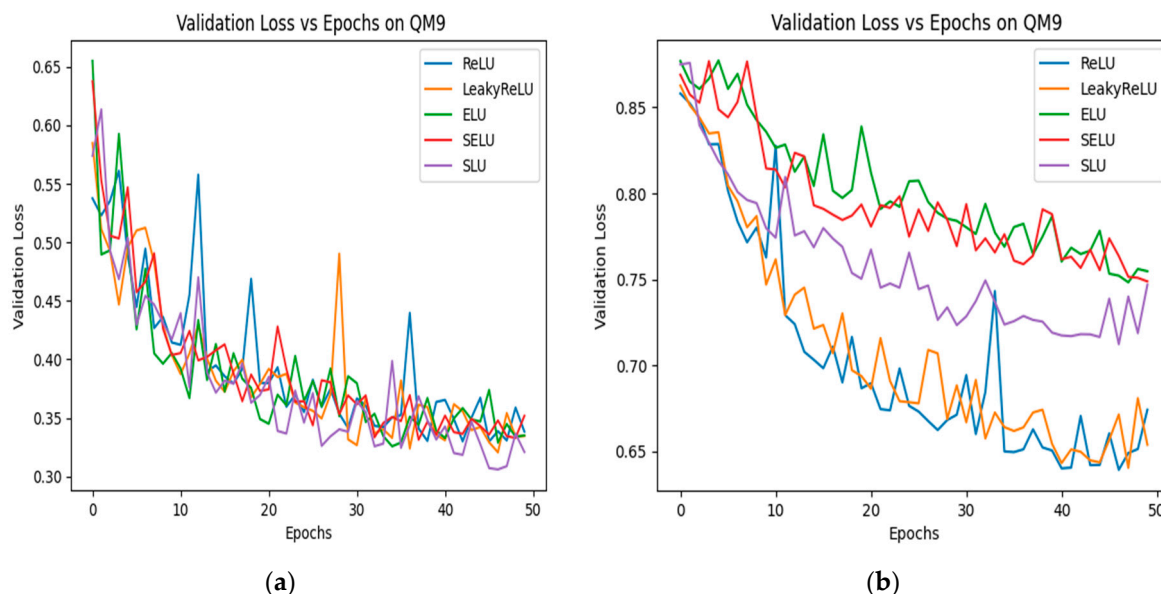


Figure 9. Validation loss curves achieved via different activation functions on QM9. (a) Validation loss vs. epochs in GIN. (b) Validation loss vs. epochs in GCN.

Table 3 shows the MAE scores achieved by GNN models using different activation functions (ReLU, LeakyReLU, ELU, SELU, and SLU). With an MAE score of 0.3315 in the GIN model, the SLU activation function performs better than ReLU (0.3741), LeakyReLU (0.3738), ELU (0.3558), and SELU (0.3652). On the other hand, the GCN model performs better with LeakyReLU (0.6614) but shows a higher MAE score of 0.7598 with SLU compared to all others except ELU. This suggests that the SLU function helps the GIN model learn complex quantum mechanical properties more effectively, while the GCN struggles to capture the finer details of the dataset.

Table 3. MAE score of GNN models on QM9.

Model	GIN	GCN
ReLU	0.3741	0.6742
LReLU	0.3738	0.6614
ELU	0.3558	0.7648
SELU	0.3652	0.7542
SLU	0.3315	0.7598

Now we visualize the effectiveness of SLU on the regression task by comparing the actual target values with the predicted values using the testing data. By utilizing the GIN model trained with SLU, we created two scatter plots, shown in Figure 10, illustrate the predicted and target values for the FreeSolv and QM9 datasets. In these scatter plots, the *y*-axis represents the actual target values, while the *x*-axis shows the predicted values. When points are above the diagonal line, it means that the predicted values are higher than the actual target values. This suggests that the model is overestimating the results for those particular samples. Conversely, points below the diagonal suggest that the predicted values are lower than the actual targets, indicating inaccuracies in the model’s predictions. An ideal scenario would be a balanced distribution of points, suggesting that the model is reliable and effective, with most points clustering closely to the diagonal and only a few significantly above or below. As we can see in Figure 10, the scatter plots reveal a strong correlation between the actual and predicted values, as demonstrated by the concentration of points near the diagonal line. This close clustering illustrates how well the

GIN model trained with SLU captures the relationships in the FreeSolv and QM9 datasets. The few outliers that deviate from the diagonal further affirm the model's performance and reliability. The alignment of points with the diagonal indicates the model's good generalization, showcasing SLU's ability to optimize predictions across these molecular datasets. The consistent clustering of samples around the diagonal line in both scatter plots highlights SLU's effectiveness in generating accurate and dependable predictions in various molecular scenarios.

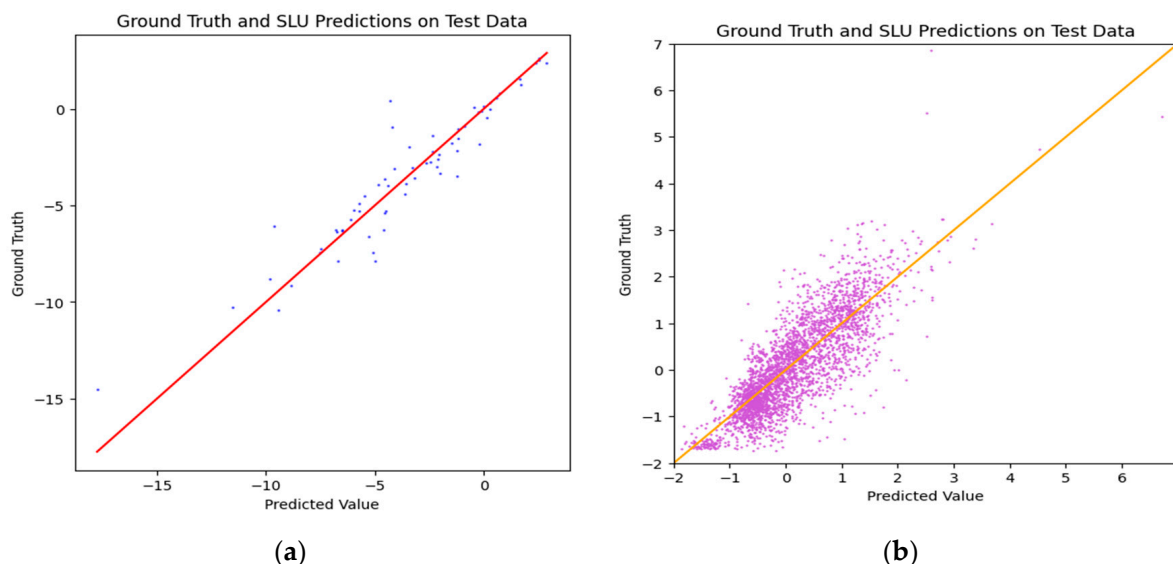


Figure 10. Analysis of ground truth vs. predictions utilizing SLU on testing data. (a) Scatter plot of GIN on FreeSolv. (b) Scatter plot of GIN on QM9.

4.3. Experimental Results on BACE

Table 4 highlights the ROC-AUC scores of five activation functions used with GNN models on the BACE dataset. SLU stands out with scores of 0.8243 for the GIN model and 0.8264 for the GCN model, showing strong predictive potential for BACE inhibition. In comparison, ReLU and LeakyReLU trailed SLU by 0.0483 and 0.0529 in GIN, and by 0.0678 and 0.0661 in GCN, respectively, indicating significant performance gaps. SELU, while closer to SLU, still showed improvements of 0.0241 in GIN and 0.0244 in GCN. Overall, the data reveal a clear trend where SLU significantly enhances ROC-AUC scores, making it a promising activation function for improving model performance on the BACE dataset.

Table 4. ROC-AUC score of GNN models on BACE.

Model	GIN	GCN
ReLU	0.7760	0.7585
LReLU	0.7744	0.7603
ELU	0.8121	0.7706
SELU	0.8002	0.8020
SLU	0.8243	0.8264

As shown in Figures 11 and 12, SLU consistently achieved the lowest training and validation losses than its competitors, while also avoiding the chaotic behavior that other activation functions display.

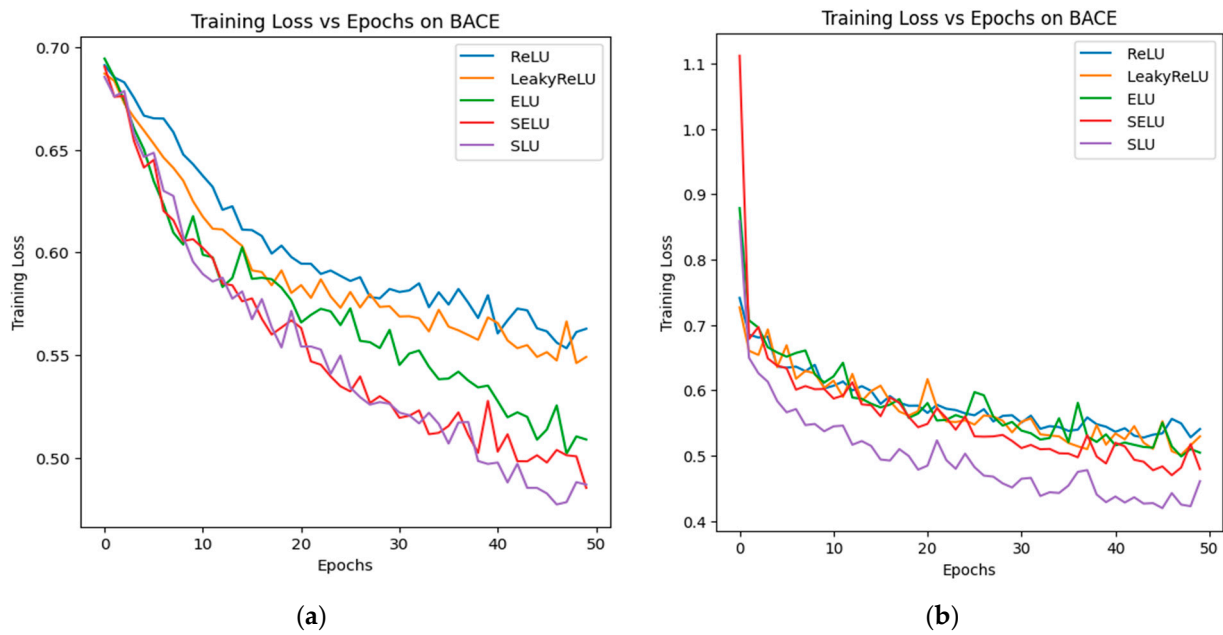


Figure 11. Training loss curves achieved via different activation functions on BACE. (a) Training loss vs. epochs in GIN. (b) Training loss vs. epochs in GCN.

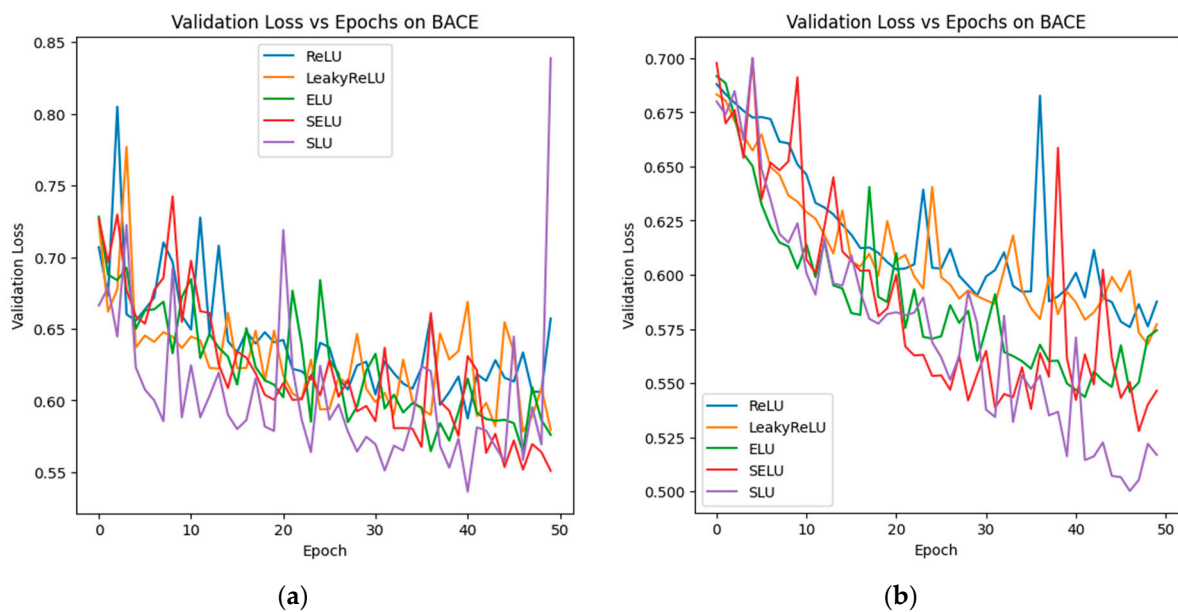


Figure 12. Validation loss curves achieved via different activation functions on BACE. (a) Validation loss vs. epochs in GIN. (b) Validation loss vs. epochs in GCN.

4.4. Experimental Results on BBBP

Table 5 discloses the classification results on BBBP that were obtained via the activation functions ReLU, LeakyReLU, ELU, SELU, and SLU. Evidently, SLU consistently outperformed the other four activation functions, achieving the highest ROC-AUC value for both models. In GCN, SLU demonstrated better predictive performance compared to ReLU and LeakyReLU. Although the differences with ELU and SELU were not statistically significant, SLU still outperformed them, achieving improvements of 0.3409 and 0.1303, respectively. Surprisingly, SLU exhibited remarkable performance in the GIN model as well, with a ROC-AUC value of 0.9683, demonstrating its potential for use in complex models compared to other activation functions.

Table 5. ROC-AUC score of GNN models on BBBP.

Model	GIN	GCN
ReLU	0.9209	0.9056
LeakyReLU	0.9211	0.9049
ELU	0.9410	0.9257
SELU	0.9396	0.9292
SLU	0.9683	0.9376

The training and validation loss curves for the BBBP dataset, shown in Figures 13 and 14, highlight the impressive performance of the SLU activation function in both GIN and GCN models. SLU consistently achieves the lowest loss values, showing its effectiveness in reducing prediction errors related to blood-brain barrier penetration. It demonstrates rapid convergence during training and lower validation loss, indicating strong generalization and reduced overfitting.

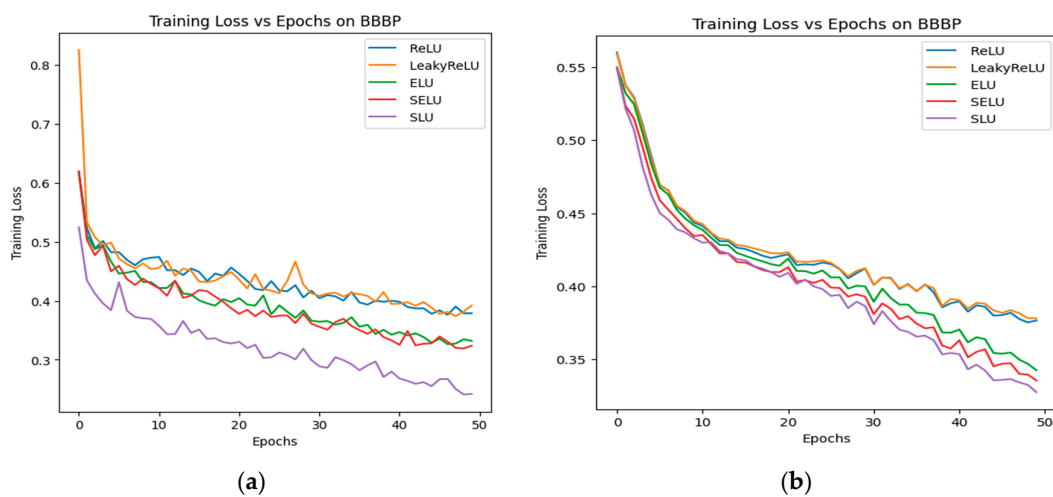


Figure 13. Training loss curves achieved via different activation functions on BBBP. (a) Training loss vs. epochs in GIN. (b) Training loss vs. epochs in GCN.

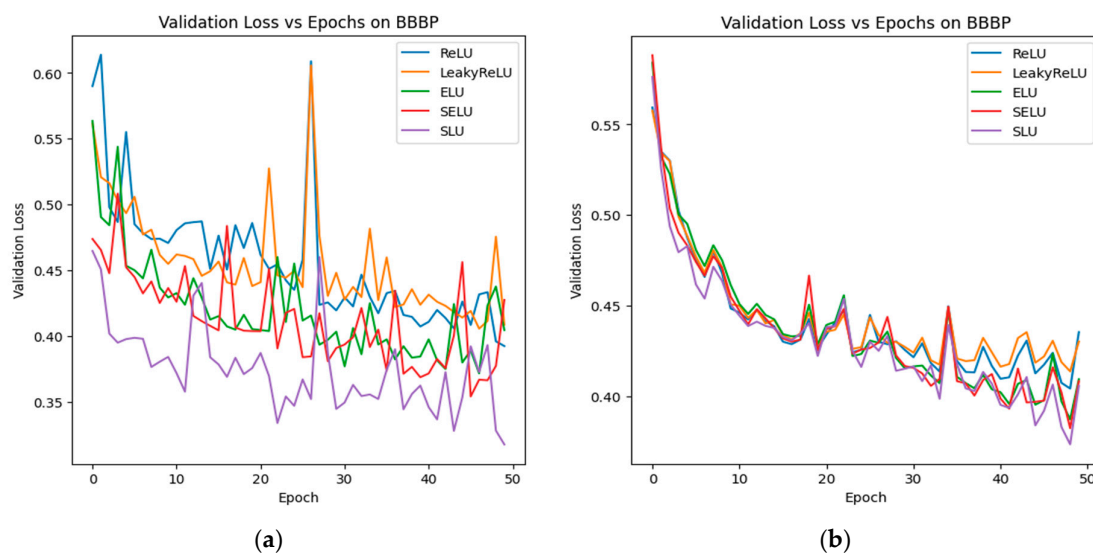


Figure 14. Validation loss curves achieved via different activation functions on BBBP. (a) Validation loss vs. epochs in GIN. (b) Validation loss vs. epochs in GCN.

The bar charts in Figure 15 visually compare the overall performance in both regression and classification tasks. In the regression tasks using the FreeSolv dataset, SLU is clearly distinguished, showing better performance than the other activation functions. It has the lowest bars in both the GIN and GCN models, indicating that it effectively minimizes error values. For the QM9 dataset, SLU continues to perform well, achieving the lowest error in the GIN model, yet it performs less effectively in the GCN model.

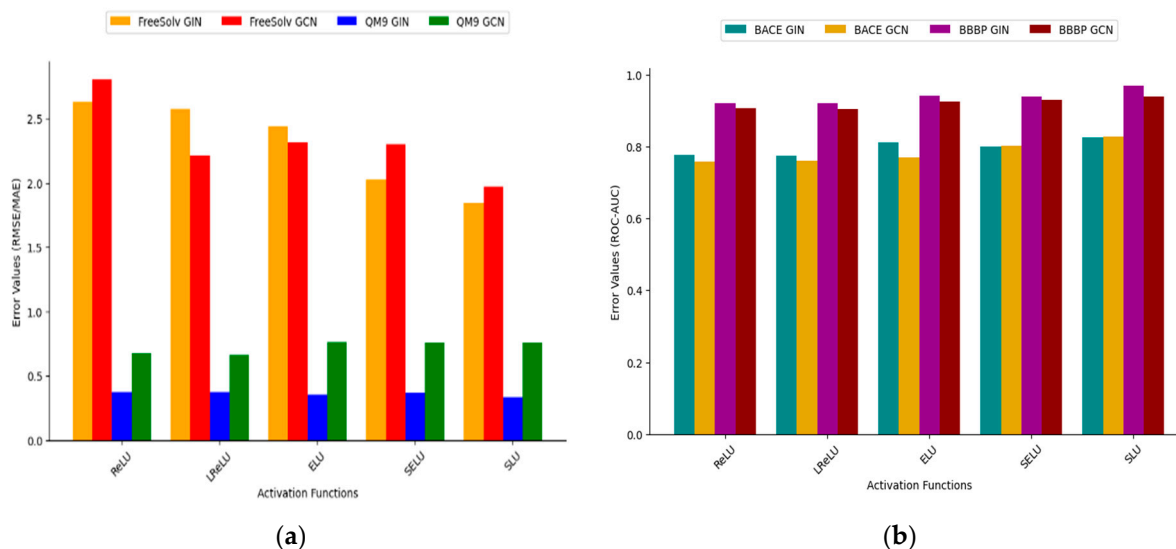


Figure 15. Analysis of SLU via different activation functions on Benchmark datasets. (a) Comparison of GIN and GCN models on FreeSolv and QM9 datasets. (b) Comparison of GIN and GCN models on BACE and BBBP datasets.

In the classification tasks, SLU demonstrates its strength on the BACE and BBBP datasets. The bars representing SLU in these tasks are significantly higher than those for the other activation functions, highlighting its exceptional predictive ability. The tall bars in the ROC-AUC scores for the BBBP dataset reflect SLU's strong performance, while the bars for the BACE dataset further confirm its superiority.

Overall, these findings underscore the effectiveness of SLU in both regression and classification tasks, especially when it comes to predicting molecular properties. Its strong performance in the molecular domain establishes SLU as an essential tool for future research in molecular modeling.

5. Conclusions and Future Directions

In conclusion, our study highlights the advancements achieved through the introduction of the Sine Linear Unit (SLU) activation function in GNNs for predicting molecular properties. We assessed the performance of two GNN models using five activation functions: ReLU, LeakyReLU, ELU, SELU, and SLU. To do this, we utilized four benchmark datasets, which included two regression tasks (FreeSolv and QM9) and two classification tasks (BACE and BBBP). This approach allowed us to compare the effectiveness of each activation function in various predictive scenarios. Our results show that SLU consistently outperforms traditional activation functions across all tasks, with one exception—the GCN model on the QM9 dataset. The impressive performance of SLU demonstrates its potential to significantly improve prediction accuracy in molecular modeling. However, the GCN model's performance on the QM9 dataset remains an exception, indicating a potential area for further research. Overall, SLU emerges as a valuable tool for enhancing the capabilities of GNNs in the complex field of molecular property prediction, paving the way for future research and applications in chemical analysis and molecular engineering.

Author Contributions: Conceptualization, A.R. and J.U.R.; methodology, J.U.R.; software, A.R.; validation, J.U.R. and Q.I.; formal analysis, A.R.; investigation, A.R. and J.U.R.; resources, A.R.; writing—original draft preparation, A.R.; writing—review and editing, J.U.R. and Q.I.; visualization, All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The molecular SMILES strings and corresponding data sets are available <https://moleculenet.org/datasets-1>.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jin, W.; Coley, C.; Barzilay, R.; Jaakkola, T. Predicting organic reaction outcomes with weisfeilerlehman network. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
2. Schwaller, P.; Gaudin, T.; Lanyi, D.; Bekas, C.; Laino, T. “Found in Translation”: Predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chem. Sci.* **2018**, *9*, 6091–6098. [[CrossRef](#)] [[PubMed](#)]
3. Zhang, X.C.; Wu, C.K.; Yang, Z.J.; Wu, Z.X.; Yi, J.C.; Hsieh, C.Y.; Hou, T.J.; Cao, D.S. MGBERT: Leveraging unsupervised atomic representation learning for molecular property prediction. *Brief. Bioinform.* **2021**, *22*, bbab152. [[CrossRef](#)] [[PubMed](#)]
4. Mahmood, O.; Mansimov, E.; Bonneau, R.; Cho, K. Masked graph modeling for molecule generation. *Nat. Commun.* **2021**, *12*, 3156. [[CrossRef](#)] [[PubMed](#)]
5. Rathi, P.C.; Ludlow, R.F.; Verdonk, M.L. Practical high-quality electrostatic potential surfaces for drug discovery using a graph-convolutional deep neural network. *J. Med. Chem.* **2019**, *63*, 8778–8790. [[CrossRef](#)]
6. Segler, M.H.; Preuss, M.; Waller, M.P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604–610. [[CrossRef](#)]
7. Krallinger, M.; Rabal, O.; Lourenco, A.; Oyarzabal, J.; Valencia, A. Information retrieval and text mining technologies for chemistry. *Chem. Rev.* **2017**, *117*, 7673–7761. [[CrossRef](#)]
8. Bean, D.M.; Wu, H.; Iqbal, E.; Dzahini, O.; Ibrahim, Z.M.; Broadbent, M.; Stewart, R.; Dobson, R.J.B. Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records. *Sci. Rep.* **2017**, *7*, 16416. [[CrossRef](#)]
9. 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* **2020**, arXiv:2011.13230.
10. Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction. *arXiv* **2020**, arXiv:2010.09885.
11. Honda, S.; Shi, S.; Ueda, H.R. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv* **2019**, arXiv:1911.04738.
12. Wang, S.; Guo, Y.; Wang, Y.; Sun, H.; Huang, J. Smiles-bert: Large scale unsupervised pre-training for molecular property prediction. In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Niagara Falls, NY, USA, 7–10 September 2019; pp. 429–436.
13. Shin, B.; Park, S.; Kang, K.; Ho, J.C. Self-attention based molecule representation for predicting drug-target interaction. In Proceedings of the Machine Learning for Healthcare Conference, Ann Arbor, MI, USA, 8–10 August 2019; PMLR; pp. 230–248.
14. Zheng, S.; Yan, X.; Yang, Y.; Xu, J. Identifying structure-property relationships through smiles syntax analysis with self-attention mechanism. *J. Chem. Inf. Model.* **2019**, *59*, 914923. [[CrossRef](#)] [[PubMed](#)]
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Wang, H.; Li, W.; Jin, X.; Cho, K.; Ji, H.; Han, J.; Burke, M.D. Chemical-reaction-aware molecule representation learning. *arXiv* **2021**, arXiv:2109.09888.
18. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
19. Rahman, J.U.; Danish, S.; Lu, D. Oscillator Simulation with Deep Neural Networks. *Mathematics* **2024**, *12*, 959. [[CrossRef](#)]
20. Ul Rahman, J.; Danish, S.; Lu, D. Deep Neural Network-Based Simulation of Sel'kov Model in Glycolysis: A Comprehensive Analysis. *Mathematics* **2023**, *11*, 3216. [[CrossRef](#)]
21. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
22. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR; pp. 1263–1272.
23. Ishida, S.; Miyazaki, T.; Sugaya, Y.; Omachi, S. Graph neural networks with multiple feature extraction paths for chemical property estimation. *Molecules* **2021**, *26*, 3125. [[CrossRef](#)]
24. Duvenaud, D.K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R.P. Convolutional networks on graphs for learning molecular fingerprints. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
25. Gasteiger, J.; Groß, J.; Günnemann, S. Directional message passing for molecular graphs. *arXiv* **2020**, arXiv:2003.03123.

26. Schütt, K.; Unke, O.; Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; PMLR; p. 93779388.
27. Schütt, K.T.; Sauceda, H.E.; Kindermans, P.J.; Tkatchenko, A.; Müller, K.R. Schnet—a deep learning architecture for molecules and materials. *J. Chem. Phys.* **2018**, *148*, 241722. [[CrossRef](#)] [[PubMed](#)]
28. 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. *Chem. Sci.* **2018**, *9*, 513–530. [[CrossRef](#)] [[PubMed](#)]
29. Rollins, Z.A.; Cheng, A.C.; Metwally, E. MolPROP: Molecular Property prediction with multimodal language and graph fusion. *J. Cheminform.* **2024**, *16*, 56. [[CrossRef](#)] [[PubMed](#)]
30. Zhang, Q.; Mao, D.; Tu, Y.; Wu, Y.Y. A New Fingerprint and Graph Hybrid Neural Network for Predicting Molecular Properties. *J. Chem. Inf. Model.* **2024**, *64*, 5853–5866. [[CrossRef](#)]
31. Ren, G.P.; Yin, Y.J.; Wu, K.J.; He, Y. Force field-inspired molecular representation learning for property prediction. *J. Cheminform.* **2023**, *15*, 17. [[CrossRef](#)]
32. Tang, B.; Kramer, S.T.; Fang, M.; Qiu, Y.; Wu, Z.; Xu, D. A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. *J. Cheminform.* **2020**, *12*, 16. [[CrossRef](#)]
33. Hahnloser, R.H.; Sarpeshkar, R.; Mahowald, M.A.; Douglas, R.J.; Seung, H.S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **2000**, *405*, 947–951. [[CrossRef](#)]
34. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
35. Rahman, J.U.; Makhdoom, F.; Lu, D. Amplifying Sine Unit: An Oscillatory Activation Function for Deep Neural Networks to Recover Nonlinear Oscillations Efficiently. *arXiv* **2023**, arXiv:2304.09759.
36. Cayley, F.R.S. LVII. On the mathematical theory of isomers. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1874**, *47*, 444–447. [[CrossRef](#)]
37. Richard, A.M.; Judson, R.S.; Houck, K.A.; Grulke, C.M.; Volarath, P.; Thillainadarajah, I.; Yang, C.; Rathman, J.F.; Martin, M.T.; Wambaugh, J.F.; et al. ToxCast chemical landscape: Paving the road to 21st century toxicology. *Chem. Res. Toxicol.* **2016**, *29*, 1225–1251. [[CrossRef](#)] [[PubMed](#)]
38. Blum, L.C.; Raymond, J.L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **2009**, *131*, 8732–8733. [[CrossRef](#)] [[PubMed](#)]
39. 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. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
40. Wieder, O.; Kohlbacher, S.; Kuenemann, M.; Garon, A.; Ducrot, P.; Seidel, T.; Langer, T. A compact review of molecular property prediction with graph neural networks. *Drug Discov. Today Technol.* **2020**, *37*, 1–12. [[CrossRef](#)]
41. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? *arXiv* **2018**, arXiv:1810.00826.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.