# Panel Performance Prediction Using Domain Knowledge-Guided Deep Learning

***Yiyeon Hwang, SeungWan Cho, ChangSeung Woo, JunTak Oh, and ByungSeung Lee***
**LG Display Co., Ltd., Seoul, Korea**

## Abstract

***A robust deep learning model for predicting the performance of display panel is proposed by integrating domain knowledge into the framework. The model combines a simple learning part, which captures domain-specific principles, with a deep learning part that identifies complex pattern, enabling accurate predictions even for unseen inputs. Experimental results show that the proposed method achieves high accuracy on both trained and untrained datasets while significantly reducing simulation runtime. The proposed model is efficient and interpretable, enabling rapid evaluation of multiple design variations.***

## Author Keywords

Deep Learning; Machine Learning; Circuit Simulation

## 1. Introduction

Simulation has a crucial role in circuit and layout design. Before manufacturing a product, simulations are used to ensure that the circuit or layout design performs as intended. In the case of display panel design, various performance metrics—such as luminance, power consumption, and signal uniformity—can be evaluated using simulations, typically with a Simulation Program with Integrated Circuit Emphasis (SPICE). However, since display panel circuits consist of numerous circuit arrays with diverse components, this often results in significantly increased runtimes. [1]

To handle this runtime issue, machine learning offers an effective solution. By exploiting machine learning techniques, computational costs can be drastically reduced, especially when evaluating multiple design variations within a short period. In particular, deep learning provides a powerful approach, as it autonomously learns the relationships between input features and their corresponding outputs. [2] Nevertheless, a key limitation of deep learning is that its predictive performance is reliable only within the range covered by the training dataset. This limitation may lead to inaccurate predictions for inputs outside the trained dataset range. Consequently, additional data generation or model retraining is required to ensure reliable performance for such inputs.

In this paper, we proposed a robust deep learning model that predicts the performance of panel designs. By integrating a domain-knowledge learning component into the deep neural network, the proposed model can provide predictions even for untrained input features. First, the main features, which are highly related to the target performance metric, are extracted. These main features are then employed in a simple learning part, referred to as the domain-knowledge learning part, to capture domain knowledge related to the performance metric. Additionally, a deep neural network is combined with the simple learning part to consider the effects of other input features, which are not included in the main features. Experimental results show that the proposed model provides reliable predictions even for inputs outside the trained dataset range.

The remainder of this paper is organized as follows. Section 2 introduces the basic model structure used as a substitute for the SPICE simulator. Section 3 describes the proposed method for integrating domain knowledge into the deep neural network. Section 4 presents the experimental results and discussions. Finally, Section 5 concludes the paper.

## 2. Basic Model Structure

To construct a deep learning model that represents the simulator, the model's inputs and outputs must be clearly defined. A SPICE simulation requires several types of inputs: the netlist, which describes the circuit components and their connections; design parameters, which define the behaviors of the components; and simulation conditions, etc. All these parameters must be provided as inputs to the model, as they can influence the simulation outputs.

In this paper, the input vector primarily consists of various design parameters, such as transistor sizes, their characteristics, resistances, capacitances, and input signal levels. These parameters can take either continuous or discrete values, and the input vector typically has hundreds of dimensions. The output is the target performance metric, which can be obtained from the simulator using the corresponding input parameters.

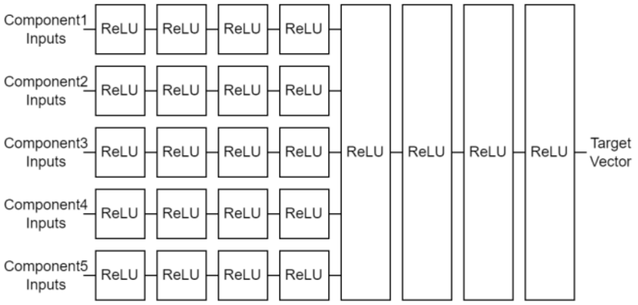


**Figure 1.** Target prediction using a deep neural network.

$$a^{l+1} = f(W^l a^l + b^l) \tag{1}$$

Once the inputs and outputs are generated for various design parameters, the input vectors are fed into a feed-forward neural network, as illustrated in Figure 1. The model learns the interactions between inputs and outputs by minimizing a loss function during training. The deep neural network contains several hidden layers, with each layer computed as shown in Equation (1). In this equation, $W^l$ represents the weights at the $l$-th layer, $a^l$ is the activation output using the rectified linear unit (ReLU), and $b^l$ is the bias at the $l$-th layer.

Additionally, as depicted in Figure 1, grouping related input parameters from the same components in the schematic helps the model learn more meaningful interactions. [3] Once trained, the model can predict outputs directly from input vectors without requiring additional simulations. However, because the model learns the interactions between inputs and outputs only within the scope of the training dataset, its predictions may become less accurate for inputs outside the range of the trained dataset.

***ISSN 0097-996X/25/5603-1685-$1.00 © 2025***

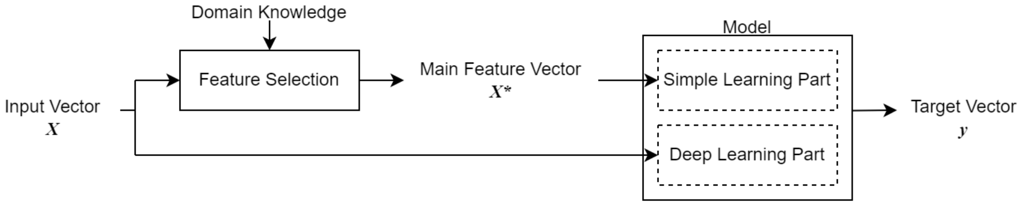


**Figure 2.** Overall flow of the proposed method.

## 3. Proposed Methods

An overview of the proposed method is illustrated in Figure 2. The method consists of two main steps. First, the main feature vector $X^*$ is extracted from the input vector $X$ based on domain knowledge. Then, $X^*$ is passed through the simple learning part of the model, while $X$ is fed into the deep learning part. The final output $y$ is predicted after the feature-guided deep learning process.

### *3.1. Feature Selection*

#### ***(a) Filtering***

The filtering step selects candidate features relevant to the target output. If certain input parameters are already known to have a strong correlation with the output, they can be directly utilized. In this paper, we identify candidate subsets of main features by statistically analyzing the relationships between inputs and outputs. The marginal effects [4] of each input parameter are evaluated to determine their individual contributions to the target output.

#### ***(b) Wrapping***

The candidate feature subsets are further tested using a simple deep neural network to assess their predictive capability. Subsets that can reliably predict the output using only their included parameters are considered for final selection. The final subset is chosen based on two criteria: robust predictive performance and minimal size. If the target output can be accurately predicted with only a few highly correlated parameters, the resulting model is not only simpler and more interpretable but also likely to align with underlying physical principles.

### *3.2. Feature-Guided Deep Learning*

The structure of the proposed model is inspired by the *wide & deep* framework. [5] It comprises two components: a simple learning part, analogous to the wide component in [5], and a deep learning part, which corresponds to the deep component in [5]. The simple learning part is effective and interpretable because it captures human-defined rules or straightforward interactions. Meanwhile, the deep learning part introduces diversity by identifying complex, latent patterns across all input parameters.

The main feature vector $X^*$ is fed into the simple learning part of the model to capture the relationship between the main features and the output. If the main features exhibit a linear relationship with the output, these features can be directly connected to the final layer without the use of hidden layers. By leveraging the simple learning part, the model can make predictions for inputs beyond the range of the training dataset by applying domain knowledge.

The deep learning part complements the simple learning part. While the output is dominantly influenced by the main features, it can also be affected by additional input parameters. By incorporating all input parameters into the deep learning part, the model can learn more complicated patterns, thereby enhancing predictive accuracy. This combination of both parts allows the model to balance interpretability and complexity.

Finally, the outputs from the simple learning part and the deep learning part integrated through the final layer, as illustrated in Figure 3. As described in Equation (2), the final output is computed as a weighted sum of the outputs from both parts. ($\mathbf{y}$ is the final target vector, $\mathbf{w}_{simple}^T$ and $\mathbf{w}_{deep}^T$ are the weights from the simple learning and deep learning part, $a^{l1}$ and $a^{l2+l3}$ are activation outputs from each part, and $b$ is the bias term.) The model is trained using backpropagation to minimize a loss function. The number of layers ($l1$, $l2$, $l3$) can be adjusted to control the contributions of each part. Additionally, dropout can be applied to the deep learning part to mitigate overfitting.

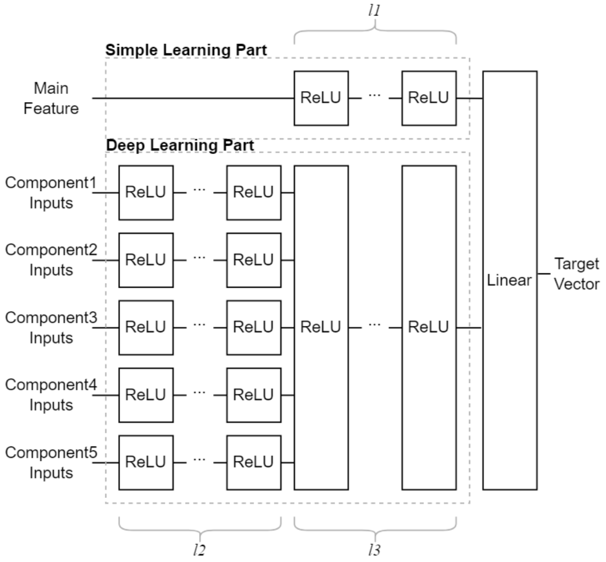


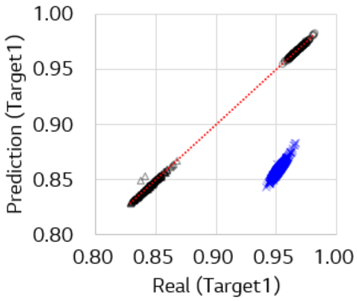
**Figure 3.** A proposed model structure.

$$\mathbf{y} = \mathbf{w}_{simple}^T a^{l1} + \mathbf{w}_{deep}^T a^{l2+l3} + b \qquad (2)$$

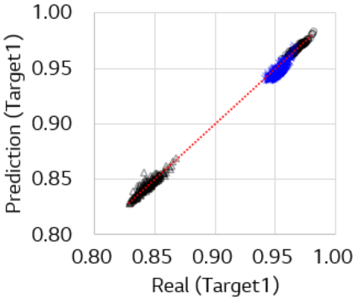
## 4. Experimental Results

To evaluate the predictive performance of the proposed method on untrained input vectors, we generate three distinct datasets with different panel sizes. Each dataset contains various design parameters corresponding to a specific panel size. The input-output pairs were obtained using Smart SPICE simulation. Among these datasets, two (Panel A and C) were used for training, while the remaining one (Panel B) was reserved for evaluation. For Panels A and C, 800 samples were used for training, and 200 samples were allocated as test sets. Panel B, containing 1100 untrained input samples, was used to evaluate the model's predictive performance. In this experiment, we used the mean squared error (MSE) as the loss function and Adam [6] as the optimizer. The model predicted three different target performance metrics. (Target 1, 2, and 3)

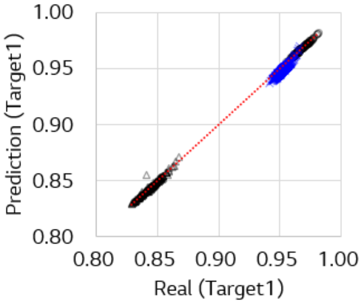
The predictive results for Target 1 are shown in Figure 4. In this figure, the x-axis represents the true data from SPICE simulation, while the y-axis shows the predictions made by the model. The ○, △, and × markers indicate test results for Panels A, C, and B, respectively. The performances of the basic model, the simple learning part-only model, and the proposed model are compared in Figure 4.



(a) Results from the basic model



(b) Results from the simple learning part-only model



(c) Results from the proposed model

**Figure 4.** Predictive results for Target 1

As illustrated in Figure 4, the basic model demonstrates strong predictive performance for the trained datasets (Panels A and C). However, its predictions for the untrained dataset (Panel B) are inaccurate. This limitation arises because the basic model learns the input-output interactions only within the scope of the training dataset, without incorporating domain knowledge.

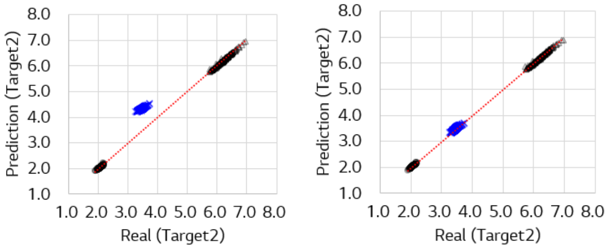
**Table 1.** Predictive Performance Comparison for Untrained Dataset (Panel B); R-squared Value and Mean Absolute Percentage Error (MAPE)

| Model | $R^2$ | MAPE |
|---|---|---|
| Basic Model | 0.970 | 9.71% |
| Simple Leaning Part-Only Model | 0.858 | 0.54% |
| Proposed Model | 0.973 | 0.38% |

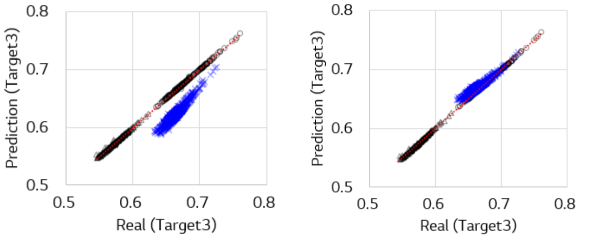
In contrast, the simple learning part-only model can predict outputs for the untrained dataset (Panel B), as shown in Figure 4. This improvement is attributed to the model's ability to learn the domain knowledge linking the main features and the output. However, the R-square value is lower than that of the basic model. This discrepancy occurs because additional input parameters, beyond the main features, also contribute to the output.

The proposed method enhances prediction accuracy even for inputs outside the training data range. By integrating simple learning part and the deep learning part, the model primarily learns domain knowledge while also capturing variations in the other influenced by other input parameters. Consequently, the proposed model achieves robust performance across both trained and untrained datasets by leveraging domain knowledge.

Figure 5 presents predictions for Target 2 and 3. Target 2 exhibits significant variability depending on the panel size, while Target 3 maintains a consistent range across different panel sizes. These results demonstrate that the proposed model delivers robust predictive performance for untrained datasets, even for varying target distributions.



(a) Predictive results for Target 2; the basic model (left) and the proposed model (right)



(b) Predictive results for Target 3; the basic model (left) and the proposed model (right)

**Figure 5.** Predictive results for Target 2 and 3

## 5. Conclusions

In this paper we proposed a robust deep learning model for predicting the performance of panel designs by leveraging domain knowledge. The main features were extracted based on domain-specific insights, and the integration of a simple learning part with a deep learning part allowed the model to learn both straightforward principles and complex patterns simultaneously.

Experimental results demonstrated that the proposed model achieves high predictive accuracy, even for inputs beyond the range of the training dataset. Furthermore, the proposed method significantly reduces simulation runtime to under 5 minutes, compared to over 120 minutes required by traditional SPICE tools for the same tasks. This makes the proposed approach an efficient solution, particularly for evaluating multiple design variations within a short time.

Due to its simplicity and flexibility, this framework is expected to generalize to other domains that require domain-aware prediction models. Future work will explore the application of diverse datasets to enhance model's generality and accuracy, making it even more robust for broader applications.

## 6. Acknowledgements

This work was supported by LG Display Co., Ltd.

## 7. References

1. J. Lee, S. Park, M. Kweon, and S. Kang, "Machine Learning-based Fast Circuit Simulation for Analog Circuit Array," *2023 ACM/IEEE 5th Workshop on Machine Learning for CAD (MLCAD)*, Sep., 2023.
2. A. L. Georgevici, and M. Terblanche, "Neural Networks and Deep Learning: a Brief Introduction," *Intensive Care Medicine*, Feb., 2019, 45, pp. 712-714
3. P Cui, X. Wang, J. Pei, and W. Zhu, "A Survey on Network Embedding," *IEEE Transactions on Knowledge and Data Engineering*, May, 2019, 31(5), pp. 833-852
4. S. Leong, and Y. Shoham, "Marginal Contribution Nets: A Compact Representation Scheme for Coalitional Games," *Proceedings of the 6th ACM Conference on Electronic Commerce*, Jun., 2005, pp. 193-202
5. H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye et al., "Wide & Deep Learning for Recommender Systems," *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS)*, Sep., 2016, pp. 7-10
6. D. P. Kingma, and J. L. Ba, "ADAM: a Method for Stochastic Optimization," *3rd International Conference on Learning Representations (ICLR)*, May, 2015.