

# Optimizing Deep Learning Models with Hybrid Nonlinear Loss Functions: Integrating Heterogeneous Nonlinear Properties for Enhanced Performance

Seoung-Ho Choi

Faculty of Basic Liberal Art, College of Liberal Arts, Hansung University, Seoul, Korea

## Abstract

Current deep learning models underperform when using loss functions characterized by single properties. Therefore, optimizing these models with a combination of multiple attributes is essential to enhance performance. We propose a novel hybrid nonlinear loss function technique incorporating heterogeneous nonlinear properties to achieve optimal performance. We evaluated the proposed method using six analytical techniques: contour map visualization, loss error frequency analysis, scatter plot visualization, loss function visualization, gradient descent analysis, and covariate Analysis with convex and linear coefficients. Our experiments on semantic segmentation tasks using the Pascal visual object classes and automatic target recognition datasets demonstrated superior optimization performance compared to existing loss functions.

**Keywords:** Hybrid nonlinear loss function, Heterogeneous properties, Optimization, High performance

## 1. Introduction

A loss function is employed to mitigate overfitting during optimization and training. Various types of loss functions exist, including the L1 loss function [1], L2 loss function [1], nonlinear exponential loss function, and exponential moving average loss function.

The exponential moving average is a type of loss function designed to capture the dynamic properties of a model by optimizing dynamic information. Notably, nonlinear properties can vary dynamically. However, the nonlinear exponential loss function ensures stable training of the model by continuously reflecting these nonlinear fixed properties.

Existing loss functions have been investigated to structure single properties and enhance model performance effectively. However, training models with single properties do not achieve optimal performance compared to their optimum state. Although models trained with single properties consistently reach a particular value, their performance remains suboptimal. Moreover, optimizing models with single properties is unlikely to achieve the best performance, as the reliability of such performance is limited.

Existing loss functions have been examined to structure single properties and enhance model performance effectively. However, training deep learning models with single properties do not achieve optimal performance compared to their ideal state. Models trained with single properties tend to accumulated consistent values, reaching certain thresholds. Moreover, such

Received: Jul. 7, 2024  
 Revised : Aug. 17, 2024  
 Accepted: Dec. 11, 2024

Correspondence to: Seoung-Ho Choi  
 (jcn99250@naver.com)  
 ©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

models are unlikely to achieve peak performance, as their reliability remains limited. To optimize large and complex models, combinatorial optimization methods are necessary. Given that deep learning models are large and complex nonlinear models systems, combinatorial optimization can yield higher performance than traditional single optimization techniques. Therefore, combining various properties is essential to achieve the optimal state of the deep learning model. The contributions of our study are as follows.

- We propose a novel hybrid loss function technique that achieves optimal model performance by incorporating heterogeneous nonlinear properties.
- We evaluate the effectiveness of the proposed method using three analytical techniques: contour map visualization, loss error frequency analysis, and scatter plot visualization of the loss error map.
- We conduct a relational analysis to evaluate the impact of the proposed technique on the loss function. We examine the relationships between the loss function and existing loss functions using the activation function.
- We employ various methods to influence changes in covariates of the loss function on model optimization. Additionally, we analyze the relationship between these covariates using convex and linear coefficients.
- We utilized gradient descent analysis to demonstrate that unique combinations of loss and activation functions enhance the learning performance of deep learning models.
- Using 3D surface plots and heatmap visualizations of the formulas, we visually demonstrated that different combinations of loss and activation functions improve the stability and responsiveness of the deep learning model during optimization.
- We initialized seed values for five classes to examine the impact of loss function properties on model training.
- We confirmed the importance of reflecting significant properties when considering heterogeneous nonlinear properties.

## 2. Related Works

### 2.1 Loss Function

Loss functions have evolved to address various learning problems, Liu et al. [2] developed an algorithm to propose a new loss

function by exploring variations within existing loss functions using a search space comprising 21 mathematical operators, three constant inputs, and three variable inputs. The algorithm efficiently identified optimal combinations of loss functions, and experiments conducted on the common object in context (COCO), VOC2017, and Berkeley deep drive (BDD) datasets demonstrated that these new combinations outperformed existing ones.

Wen et al. [3] introduced a novel loss function designed to mitigate performance degradation due to insufficient sample diversity without relying on sampling procedures. This approach incorporates adjustable parameters to widen the loss range, diminish the impact of easily classified samples, and replace traditional sampling functions. Derived from the cross-entropy loss, the proposed formula integrates the adjustable parameters  $\beta$  and  $c$  with the SoftMax function for multi-class image classification. Experimental results indicated optimal performance at  $\beta = 4$  and  $c = 5$ , highlighting improved classification capabilities and reduced training complexity compared to traditional loss functions.

Xie et al. [4] addressed the challenge of accurate image segmentation of surface defects in convolution neural network (CNN)-based models by proposing a balanced loss function. This study identified loss imbalance as a critical issue affecting segmentation accuracy, including label imbalance, easy-to-hard example imbalance, and boundary imbalance. The proposed balanced loss function introduced dynamic class weights, truncated cross-entropy loss, and strategies to suppress label confusion, significantly enhancing segmentation accuracy across various benchmarks and CNN segmentation models. Experimental findings consistently showed performance improvements ranging from 5% to 30% over commonly used loss functions.

### 2.2 Hybrid Loss Function

Dickson et al. [5] evaluated and compared various loss functions for training artificial neural networks (ANNs). This study analyzed a hybrid loss function that combines entropy and squared error, using the squared error for initial bootstrap network training and entropy loss for subsequent refinement. The results indicate that this approach leverages the initial optimal state achieved by the squared error while the entropy loss provides further enhancements. This strategy offers a novel method for improving the training efficiency of ANNs, contributing to advancements in neural network-based models.

Yu and Wu [6] introduced a hybrid loss function, SSIM+L1

integrating the structural similarity index measure (SSIM) and L1 norms. Their study incorporated an attention mechanism that effectively leverages global information during network training. When applied to a CNN for reconstructing missing traces, experiments on synthetic and field data demonstrated that the SSIM+L1 loss function, combined with the attention mechanism, significantly improved interpolation results compared to networks lacking attention.

In medical imaging, researchers have tailored hybrid loss functions for encoder-decoder convolutional neural networks to address underutilization and image information loss [7]. This study proposed a low-dose computed tomography (LDCT) image-denoising network that employs residual multiscale feature extraction and a hybrid loss function comprising mean square error (MSE), SSIM, and perceptual losses. Experimental results demonstrated that this approach enhanced image quality and improved computational efficiency compared to state-of-the-art techniques, offering promising advancements in medical imaging applications.

Deep-learning techniques have been employed to enhance LDCT image quality using a hybrid loss function that integrates adversarial, perceptual, sharpness, and structural similarity losses [8]. Experimental results demonstrated that the SS-WGAN with the hybrid loss function outperformed traditional denoising methods in preserving image details and structure, which are crucial for accurate medical diagnosis.

### 3. Proposed Method

Figure 1 visualizes the proposed method. Figure 1(a) illustrates hybrid loss function case 1, and Figure 1(b) depicts hybrid loss function case 2. The proposed method is detailed as follows.

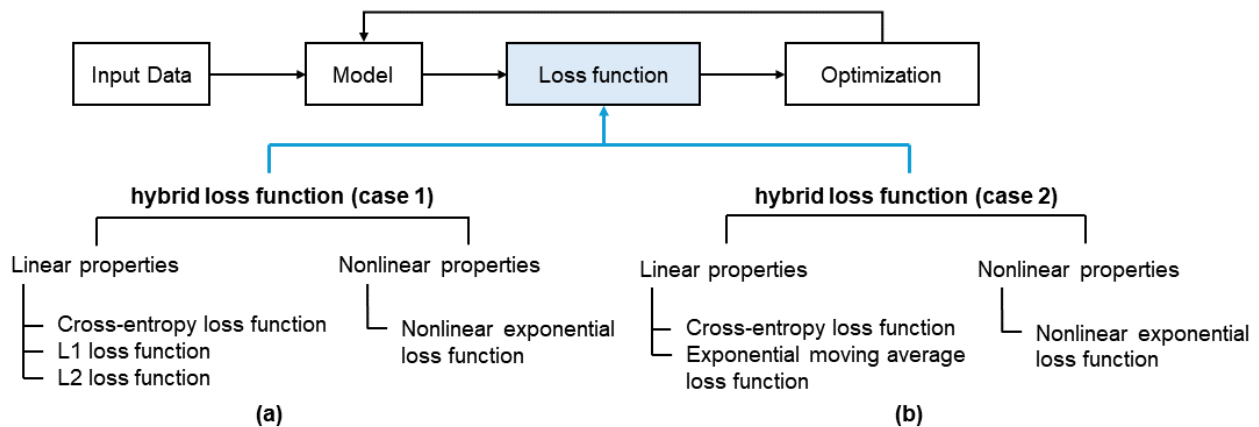


Figure 1. Proposed method: (a) hybrid loss function case 1 and (b) hybrid loss function case 2.

**Hybrid loss function:** We propose a hybrid loss function to capture heterogeneous properties. The concept of this hybrid loss function is illustrated below. We define a hybrid loss function that reflects the heterogeneous properties.

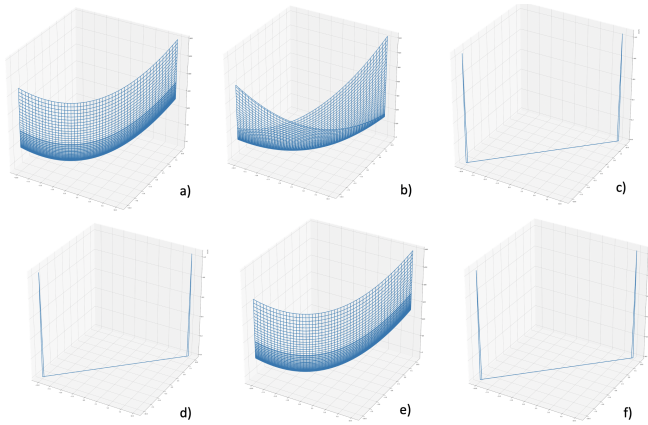
We test this function using two cases: a combination of nonlinear chromatistic and linear properties and a combination of nonlinear fixed and nonlinear dynamic properties.

We optimize deep learning models using the loss function of single properties, which limits our scope. We identified the cause of this limitation and aim to address it through heterogeneous properties. Although this might not be the optimal point for a deep learning model, expanding the range by reflecting various properties is necessary to find the optimal point. To verify this, we propose a novel hybrid loss function that incorporates these properties. We intend to validate the rationale for incorporating heterogeneous properties using three analyses.

Figure 2 visualizes the employed loss function methods using the formula  $x^2 + 4 \times x + y^2 - 6 \times y$ . This represents the hyperplane form of deep learning models. Figure 2(a) shows the experiment with no loss function. Figure 2(b) depicts the L1 and L2 loss functions. Figure 2(c) illustrates the nonlinear exponential loss function. Figure 2(d) combines the nonlinear exponential loss function with the L1 and L2 loss functions. Figure 2(e) presents the exponential moving average loss function. Figure 2(f) shows both the exponential moving average loss function and the nonlinear loss function.

Based on these results, the hyperplane space defined by the loss function may assume a narrow valley shape. This shape allows the deep learning model to converge rapidly and reach the optimal point.

**Comparison of linear and nonlinear properties in deep learning model optimization within loss functions:** Accord-



**Figure 2.** Visualization of used loss function: (a) no loss function, (b) L1 and L2 loss functions, (c) nonlinear exponential loss function, (d) nonlinear exponential loss function and L1/L2 loss functions, (e) exponential moving average loss function, and (f) exponential moving average loss function and nonlinear loss function.

ing to [9], the comparison between linear and nonlinear properties shows that nonlinear properties generally enhance deep learning model performance. This enhancement is due to the nonlinear properties being less sensitive and having fewer divergence points.

**Nonlinear dynamic properties:** Traditional neural networks are recognized as nonlinear systems, and numerous studies, such as [10], have explored methods to optimize their nonlinearity. The dynamic programming methods discussed in [11] were employed to manage deep learning model optimization effectively. Incorporating nonlinear dynamic properties is concluded to be an efficient approach for optimizing models within nonlinear systems.

Reflecting nonlinear dynamic properties within such systems proves to be an efficient approach for deep learning model optimization.

## 4. Experiment Method

**Pseudo code:** The pseudo-code for applying the proposed hybrid loss function to two cases is presented below. Algorithms 1 and 2 demonstrate the application of the proposed method for training deep learning models.

**Hybrid loss function experiment setting:** Figure 3 illustrates the experimental method configuration. The experiment comprised five stages. Step 0 involved the impact analysis of seed values. Step 1 used the seed values to generate initial model values. The second stage involved employing fully

convolution network (FCN) and U-Net. Step 3 analyzed the relationship between the loss and the loss function. Step 4 evaluated the prediction results of the deep learning model.

This experiment was applied to U-Net [12] and FCN [13] for semantic segmentation. We conducted experiments on the proposed method using Keras in an Ubuntu environment, recording the average value over five iterations. Experiments with FCN and U-Net were conducted using visual object classes (VOC) and automatic target recognition (ATR) datasets for the image segmentation task, employing cross-entropy loss. Results were evaluated using dice similarity coefficient (DSC), F1-score, Intersection over Union (IOU), loss, precision, and recall. The DSC measures the similarity of two images to evaluate the accuracy of the segmented image. The F1-score, the harmonic mean of precision and recall, evaluates binary classification model performance. IOU measures the overlap between predicted and actual segmented regions, evaluating segmentation accuracy. Loss indicates the difference between the prediction and actual value of the deep learning model. Precision shows the percentage of correctly predicted positive classes, while recall indicates the percentage of true-positive classes correctly identified by the deep learning model. We analyzed the experiment results using five seed values: 1, 250, 500, 777, and 999. The hybrid loss function, where A and B denote the covariates, is presented below.

**Relation analysis:** We employed five functions to analyze the relationship between the loss and the loss function, specifically square root (Sqrt), rectified linear unit (ReLU) function, exponential linear unit (eLU), Bi-ReLU, and Bi-eLU. Identifying the significant effects of the loss function on the loss during deep learning model optimization is crucial. The exper-

---

### Algorithm 1. Hybrid loss function case 1.

---

```

while Epoch  $\neq$  0 do
    Loss  $\leftarrow$  Crossentropy loss function
        + Nonlinear exponential loss function
        + L1 loss function + L2 loss function
    Optimization model using Loss error
end while

```

---



---

### Algorithm 2. Hybrid loss function case 2.

---

```

while Epoch  $\neq$  0 do
    Loss  $\leftarrow$  Crossentropy loss function
        + Exponential moving average
        + Nonlinear exponential loss function
    Optimization model using Loss error
end while

```

---

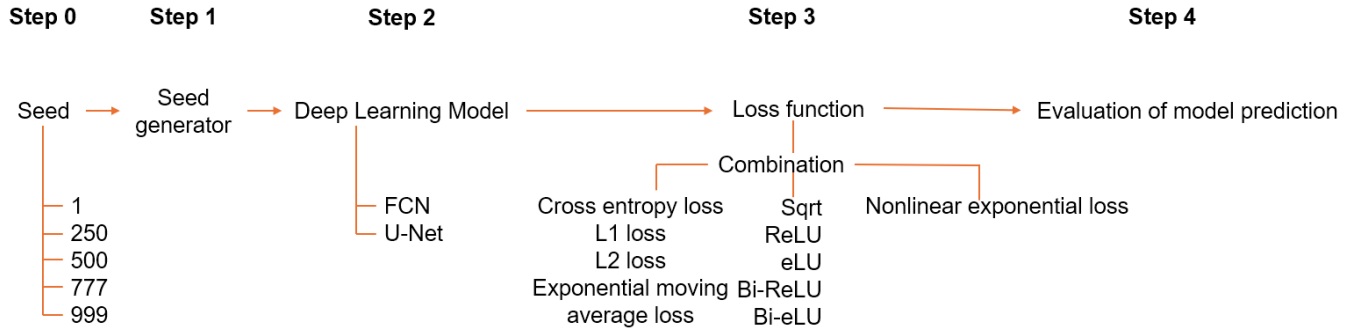


Figure 3. Visualization of experiment method configuration.

iment was conducted using these five functions: sqrt, ReLU, eLU, Bi-ReLU, and Bi-eLU. These functions were applied to four loss-function methods: exponential moving average with a linear coefficient, exponential moving average with a convex coefficient, hybrid v2 with a linear coefficient, and hybrid v2 with a convex coefficient. Our objective is to confirm the effectiveness of deep learning models in reflecting meaningful information through an analysis of the relationship between loss and existing loss functions.

**Seed analysis:** We assessed the loss function performance using five seed values. This evaluation is critical because, during hyperparameter optimization, the initial settings significantly influence the optimization outcome of deep learning models. The experiment was conducted with five seed values: 1, 250, 500, 777, and 999. Our findings confirmed that the quality expressed in the deep learning model is affected by the seed value.

**Impact analysis method of the proposed method:** We analyzed the relationship between the loss functions. Reflecting meaningful information in the loss function is essential for optimizing the deep learning model, as it promotes rapid converges along the shortest path. Additionally, this approach prevents deviations in the wrong direction.

To analyze the relationship between the covariates of the loss function, two analyses were performed: convex and linear coefficients. Deep learning models typically exhibit hyperplanes with convex properties. During optimization, the learning and learned position oscillate. This analysis determines whether the loss function should possess convex properties or whether the deep learning model should incorporate linear properties to enhance optimization.

Another analysis was conducted using five seed values with a loss function. As the deep learning model was trained, the initial distribution varied according to the seed value. Finally, we aimed to confirm the stability of the training by analyzing

the loss function based on the initial state of the model.

To understand the influence of the hyperplane on the deep learning model, we employed a contour map. This approach visualizes the impact on the hyperplane shape on the deep learning model. If the model is acquired stably, it indicates that a broad area is achieved without complex contour patterns.

Additionally, a frequency visualization analysis using loss errors was performed. Frequency analysis is challenging when the results for real and fake images are highly similar. Therefore, similar data were selected by analyzing the frequency of occurrence. The loss error can be used to analyze the state of the deep learning model by indicating a specific pattern of error convergence or emission. Both frequency and spread analyses were conducted.

Moreover, we conducted a spread analysis using the loss error. If the loss is concentrated at a single value, the expressed points cluster and appear more robustly compressed. This confirms that the deep learning model exhibits consistent errors, with similar errors recurring.

*Combination L1 and L2 lossfunction*

*with convex coefficient*

$$:= A * L1 + (1 - A) * L2, \quad (1)$$

*Combination L1 and L2 lossfunction*

*with linear coefficient*

$$:= A * L1 + B * L2, \quad (2)$$

*Nonlinear exponential lossfunction*

*with linear coefficient (static)*

$$:= A * L1e^{(1-A)*L2}, \quad (3)$$

*Exponential moving average lossfunction*

*with linear coefficient (dynamic)*

$:=$

$$Step1 := A * L1 - (1 - A) * L1 - B * L2,$$



$$\begin{aligned}
 & \text{Step2} := A * \text{Step1} + B * L2, \\
 & \text{Exponential moving average lossfunction} \\
 & \text{with convex coefficient (dynamic)} \\
 & := \\
 & \text{Step1} := A * L1 - (1 - A) * L1 - B * L2, \\
 & \text{Step2} := A * \text{Step1} + ((1 - A) * L2), \\
 & \text{Hybrid v2 lossfunction} \\
 & \text{with convex coefficient} \\
 & \text{(static and dynamic)} \\
 & := \\
 & \text{Step1} := A * L1 - (1 - A) * L1 - B * L2, \\
 & \text{Step2} := \sqrt{A * \text{Step1} + ((1 - A) * L2)} \\
 & \quad + A * L1e^{(1-A)*L2}, \\
 & \text{Hybrid v2 lossfunction} \\
 & \text{with linear coefficient} \\
 & \text{(static and dynamic)} \\
 & := \\
 & \text{Step1} := A * L1 - B * L1 - B * L2, \\
 & \text{Step2} := A * \text{Step1} + (B * L2) \\
 & \quad + A * L1e^{B*L2}, \\
 & \text{Hybrid v2 lossfunction} \\
 & \text{with linear coefficient adopted ReLU} \\
 & \text{(relation analysis using linear filtering)} \\
 & := \\
 & \text{Step1} := A * L1 - B * L1 - B * L2, \\
 & \text{Step2} := \text{ReLU}(A * \text{Step1} + (B * L2)) \\
 & \quad + A * L1e^{B*L2}, \\
 & \text{Hybrid v2 lossfunction} \\
 & \text{with linear coefficient adopted Bi - eLU} \\
 & \text{(relation analysis using bipolar filtering)} \\
 & := \\
 & \text{Step1} := A * L1 - B * L1 - B * L2, \\
 & \text{Step2} := eLU(A * \text{Step1} + (B * L2)) \\
 & \quad - eLU(A * \text{Step1} + (B * L2)) \\
 & \quad + A * L1e^{B*L2}.
 \end{aligned}$$

The equation used in the experiment is as follows:

Eq. (1) denotes the convex combination of the L1 and L2 loss functions, while Eq. (2) denotes their linear combination. These

formulations elucidate the convex and linear impacts on the interplay between the L1 and L2 loss functions as demonstrated in Eqs. (1) and (2).

Eq. (3) denotes the nonlinear exponential loss function, which validates the static effect of this function.

Eq. (4) describes an exponential moving average with a linear coefficient, whereas Eq. (5) involves a convex coefficient. Both are nonlinear exponential loss functions, elucidating the dynamic effects of convex and linear relationships on these functions.

Eq. (6) outlines an exponential moving average and nonlinear exponential loss function with a convex coefficient, and Eq. (7) outlines the same but with a linear coefficient.

Eqs. (6) and (7), respectively, confirm the exponential moving average and nonlinear exponential loss functions, demonstrating the static and dynamic effects of convex and linear relationships.

Eq. (8) denotes the exponential moving average combined with a nonlinear exponential loss function featuring a linear coefficient employing ReLU. Eq. (9) denotes an exponential moving average with a linear coefficient hybrid employing Bi-eLU. Eqs. (8) and (9) signify exponential moving averages. The experimental results are summarized as follows:

The combination of an exponential moving average with the L1 and L2 loss functions is termed a hybrid loss function.

The combination of an exponential moving average with a nonlinear exponential loss function is termed a hybrid v2 loss function.

Table 1 presents a summary of the index utilized in the experiment. The experiment involved 27 distinct loss function equations. The rationale for the experiment design is as follows: we investigate the impact of two terms on the optimization of the deep learning model by analyzing the coefficients of the loss functions. This analysis employs both linear and convex coefficients for model optimization. Additionally, we examine the trade-off relationship between these two terms and its effect on the performance of the deep learning model.

The conventional method involves a straightforward linear combination of the L1 and L2 loss functions. In contrast, the nonlinear exponential loss function is designed to capture nonlinear properties rather than the characteristics of linear combinations in the learning process of the deep learning model. Certain attributes are consistently reflected within the deep learning model. The exponential moving average loss function dynamically integrates information, including historical data, in the loss function.

Table 1. The index definition of hybrid loss function

Name	Index A
None	Experiment 1
Combination L1 and L2 loss function with convex coefficient	Experiment 2
Combination L1 and L2 loss function with linear coefficient	Experiment 3
Nonlinear exponential loss function with convex coefficient	Experiment 4
Nonlinear exponential loss function with linear coefficient	Experiment 5
Exponential hybrid loss function with convex coefficient	Experiment 6
Exponential hybrid loss function with linear coefficient	Experiment 7
Exponential moving average loss function with convex coefficient adopted no filter	Experiment 8
Exponential moving average loss function with convex coefficient adopted ReLU	Experiment 9
Exponential moving average loss function with convex coefficient adopted eLU	Experiment 10
Exponential moving average loss function with convex coefficient adopted Bi-ReLU	Experiment 11
Exponential moving average loss function with convex coefficient adopted Bi-eLU	Experiment 12
Exponential moving average loss function with linear coefficient adopted no filter	Experiment 13
Exponential moving average loss function with linear coefficient adopted ReLU	Experiment 14
Exponential moving average loss function with linear coefficient adopted eLU	Experiment 15
Exponential moving average loss function with linear coefficient adopted Bi-ReLU	Experiment 16
Exponential moving average loss function with linear coefficient adopted Bi-eLU	Experiment 17
Exponential hybrid v2 loss function with convex coefficient adopted no filter	Experiment 18
Exponential hybrid v2 loss function with convex coefficient adopted ReLU	Experiment 19
Exponential hybrid v2 loss function with convex coefficient adopted eLU	Experiment 20
Exponential hybrid v2 loss function with convex coefficient adopted Bi-ReLU	Experiment 21
Exponential hybrid v2 loss function with convex coefficient adopted Bi-eLU	Experiment 22
Exponential hybrid v2 loss function with linear coefficient adopted no filter	Experiment 23
Exponential hybrid v2 loss function with linear coefficient adopted ReLU	Experiment 24
Exponential hybrid v2 loss function with linear coefficient adopted eLU	Experiment 25
Exponential hybrid v2 loss function with linear coefficient adopted Bi-ReLU	Experiment 26
Exponential hybrid v2 loss function with linear coefficient adopted Bi-eLU	Experiment 27

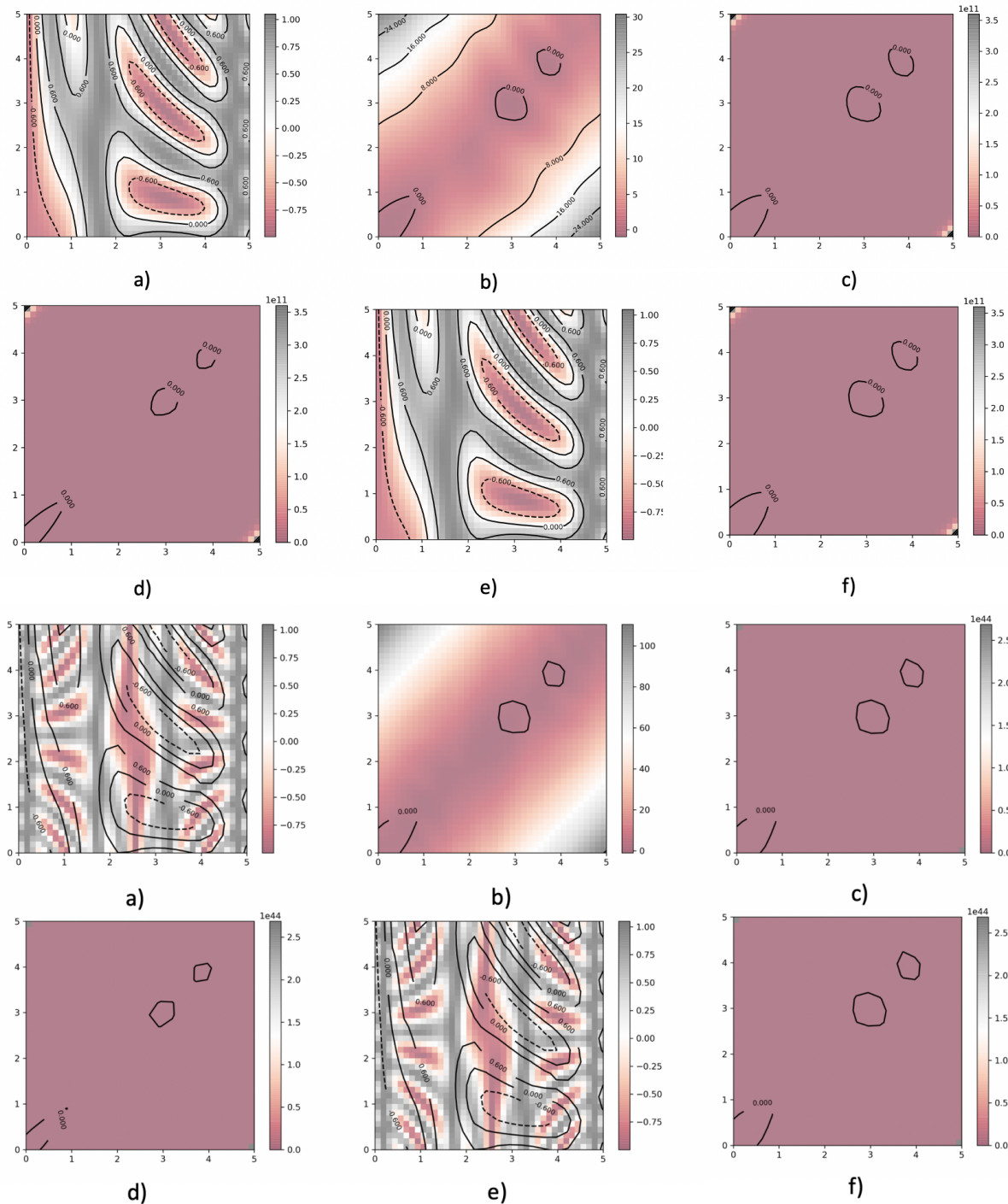
By introducing a loss function that embodies fixed properties alongside one that captures dynamic properties, we developed and evaluated a heterogeneous loss function encompassing both characteristics.

These two methods validated the impact of heterogeneous feature loss functions on deep-learning model optimization and performance.

We investigated the impact of two heterogeneous feature loss functions on deep learning model loss. This analysis aimed to enhance model learning by selectively filtering significant properties, as not all properties are effectively learned by the model.

The ReLU function, combined with linear filtering, was applied to eliminate values below zero and linearly reflect values above zero. This approach leverages the vanishing property of the ReLU function to improve model performance.

Additionally, node filtering is applied to reflect only sparse information, optimizing the learning process of the deep learning model by emphasizing meaningful information and eliminating redundancy.



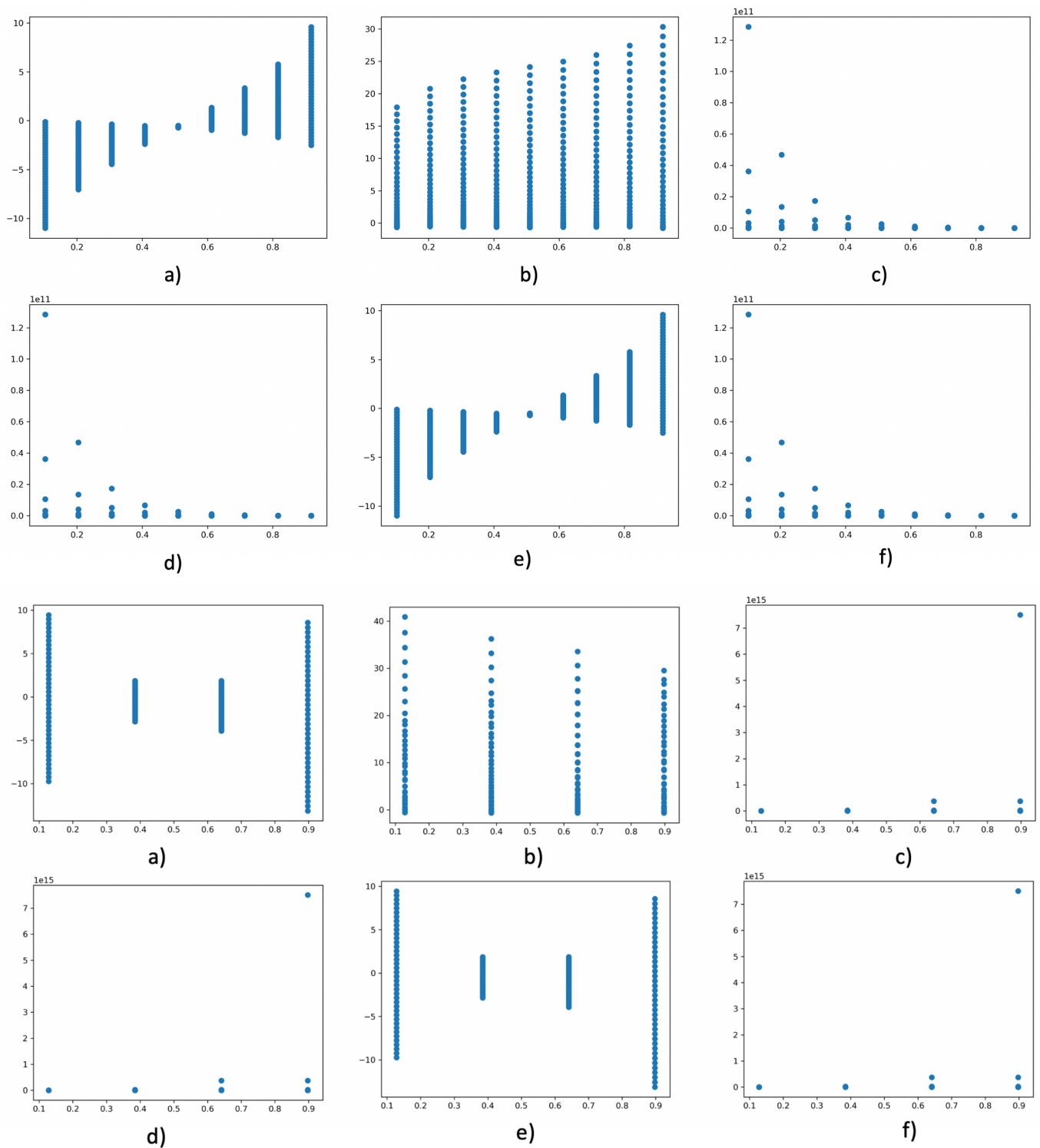
**Figure 4.** Visualization of the analysis of the proposed method using contour maps of 6 different loss functions with balanced and unbalanced input data: (a) existing experimental equation, (b) L1 and L2 loss function addition formulas, (c) nonlinear exponential loss function, (d) nonlinear exponential loss function and linear combination L1 & L2 loss functions, (e) exponential moving average loss function, and (f) exponential moving average loss function and nonlinear exponential loss function.

## 5. Experiment Result

Figure 4 presents the contour lines analysis, where the generated values are evaluated using the formula  $\sin^{10}(x) + \cos(10 +$

$y \times x) \times \cos(x)$ . For unbalanced values,  $x$  generated 40 values ranging from 0 to 5, and  $y$  was analyzed with 40 values from 0 to 5, as shown in the top six of Figure 4. For balanced values,  $x$  generated 40 values from -5 to 5, and  $y$  was analyzed with 40



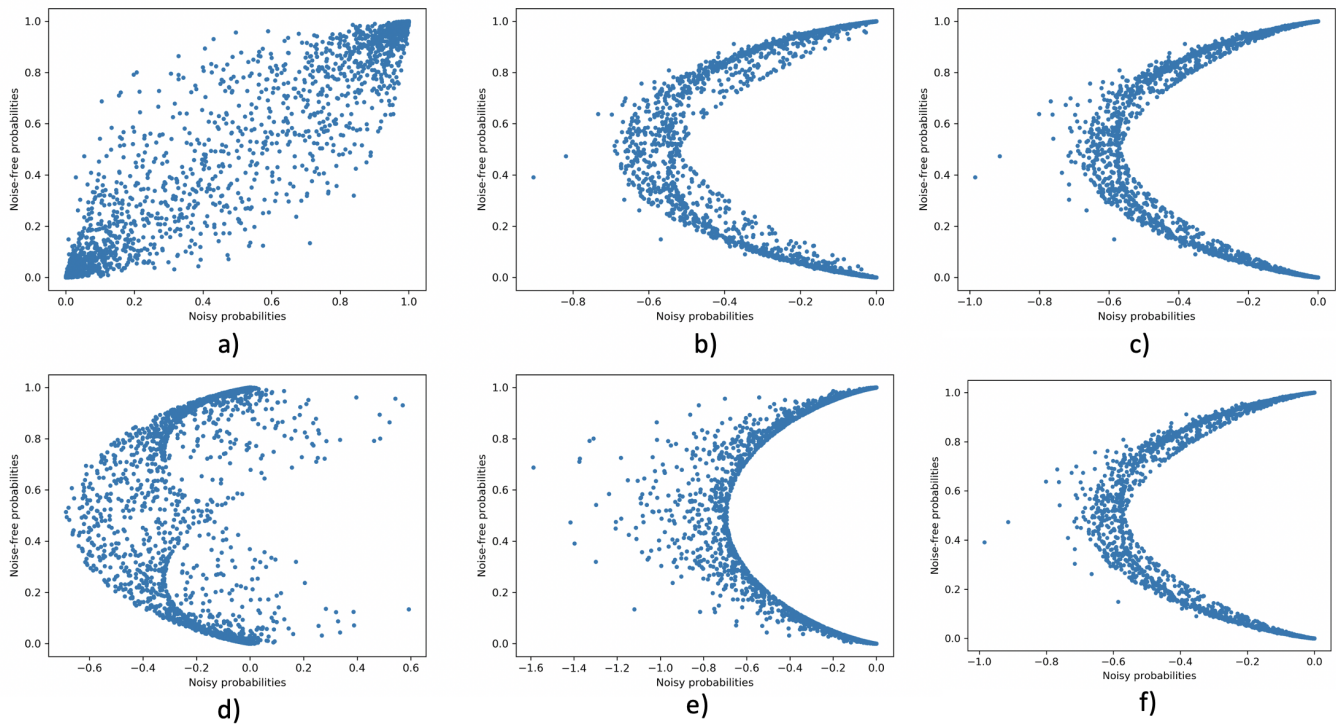


**Figure 5.** Visualization of the impact of 6 different loss functions on the frequency distribution of generated values with unbalanced and balanced input data: (a) existing experimental equation, (b) L1 and L2 loss function addition formulas, (c) nonlinear exponential loss function, (d) nonlinear exponential loss function and linear combination L1 & L2 loss functions, (e) exponential moving average loss function, and (f) exponential moving average loss function and nonlinear exponential loss function

values from 0 to 5, as depicted in the bottom six of Figure 4.

Figure 4 illustrates the frequency of value occurrence as con-

tour lines for various experimental conditions: (a) the existing experimental equation; (b) the experimental formula with L1



**Figure 6.** Visualization of scatter plots using LogisticGroupLasso model to analyze the model and sparsity values of 6 different loss functions: (a) existing experimental equation, (b) L1 and L2 loss function addition formulas, (c) nonlinear exponential loss function, (d) nonlinear exponential loss function and linear combination L1 & L2 loss functions, (e) exponential moving average loss function, and (f) exponential moving average loss function and nonlinear exponential loss function.

and L2 loss function addition; (c) the experimental formula with the nonlinear exponential loss function; (d) the experimental formula with the nonlinear exponential and linear combination L1 and L2 loss functions; (e) the experimental formula with the exponential moving average loss function, and (f) the experimental formula with both the exponential moving average and nonlinear exponential loss functions.

Figure 4 demonstrates that when the loss function is applied, numerous zero values are generated compared to previous instances. In Figure 4(b), non-zero values cluster in the upper left and lower right, indicating a phenomenon that maximizes the zero margin as non-zero values spread to the peripheries. Additionally, Figure 4(c), 4(d), and 4(f) display contour lines that may present local minima within the hyperplane space of the neural network model. When zero values occurs frequently, expanding the range of these local minima(indicated by the contour line) or identifying an optimizable range increases the information the deep learning model can learn, leading to improved optimization. As shown in the results, this process progresses to the branch.

Figure 4 presents an analysis of balanced and unbalanced data generation. For balanced data, values near zero are more

clustered than the contour lines, suggesting multiple local minima within the hyperplane of a neural network. Additionally, when both positive and negative values are generated, the contour lines appears as straight lines indicating that extreme values form a more defined boundary compared to a soft boundary.

The analysis using the frequency of occurrence is depicted in Figure 5, where each loss function method was applied to the cross-entropy loss. For unbalanced values,  $x$  generated 40 values from 0 to 5, and  $y$  was analyzed using 40 values from 0 to 5. For balanced values,  $x$  generated 40 values from -5 to 5, and  $y$  was analyzed using 40 values from -5 to 5. The resulting values are illustrated in Figure 5. The upper panel shows the case using unbalanced data, with generated values between 0 and 1, dominated by nine specific values. In Figure 5(c), 5(d), and 5(f), the frequency of occurrence largely clusters at zero. In Figure 5(a) and 5(e), the generated values are symmetrically distributed around 0.5 on the x-axis with opposite signs. The bottom panel shows the case using balanced data, with four values generated on the x-axis. In Figure 5(a), and 5(e), the generated values are symmetrically distributed around the y-axis. In Figure 5(c), 5(d), and 5(f), the values cluster at the bottom right.

Finally, the results from the experimental formula were validated by analyzing specific outcomes. This confirmed that the deep learning model continuously learned from a uniform set of values. Therefore, past research should have focused on optimizing generalization by incorporating a variety of values rather than continuously relying on specific ones.

The analysis employing the LogisticGroupLasso model is depicted in Figure 6, which presents the outcomes derived from this model. The x-axis denotes noisy probabilities, while the y-axis represents noise-free probabilities. The resultant distribution exhibits a convex function shape. Moreover, the degree of clustering varied across different loss function methods. Notably, disparate loss function methods produced non-clustered values at varying frequencies, where such non-clustered values could be identified as outliers during the training process of the deep learning model. These outliers potentially lead to suboptimal learning pathways. Hence, meticulously designing the loss function to mitigate the occurrence of sparse values is crucial.

$$w_{\text{new}} = w_{\text{old}} - a \left( A \frac{\partial L1}{\partial w} + (1 - A) \frac{\partial L2}{\partial w} \right), \quad (10)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( A \frac{\partial L1}{\partial w} + B \frac{\partial L2}{\partial w} \right), \quad (11)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( A L1 e^{(1-A)L2} (1-A) \frac{\partial L2}{\partial w} \right), \quad (12)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( A \left( A \frac{\partial L1}{\partial w} - (1-A) \frac{\partial L1}{\partial w} - B \frac{\partial L2}{\partial w} \right) + B \frac{\partial L2}{\partial w} \right), \quad (13)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( A \left( A \frac{\partial L1}{\partial w} - (1-A) \frac{\partial L1}{\partial w} - B \frac{\partial L2}{\partial w} \right) + (1-A) \frac{\partial L2}{\partial w} \right), \quad (14)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( \frac{A}{2\sqrt{A \cdot \text{Step1} + (1-A) \cdot L2}} \left( A \frac{\partial L1}{\partial w} - (1-A) \frac{\partial L1}{\partial w} - B \frac{\partial L2}{\partial w} \right) + \frac{(1-A)}{2\sqrt{A \cdot \text{Step1} + (1-A) \cdot L2}} \frac{\partial L2}{\partial w} \right), \quad (15)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( A \left( A \frac{\partial L1}{\partial w} - B \frac{\partial L1}{\partial w} - B \frac{\partial L2}{\partial w} \right) + B \frac{\partial L2}{\partial w} \right), \quad (16)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( A \cdot \frac{\partial \text{ReLU}(A \cdot \text{Step1} + B \cdot L2)}{\partial w} \cdot \frac{\partial \text{Step1}}{\partial w} + B \cdot \frac{\partial L2}{\partial w} \cdot \frac{\partial \text{ReLU}(A \cdot \text{Step1} + B \cdot L2)}{\partial w} \right), \quad (17)$$

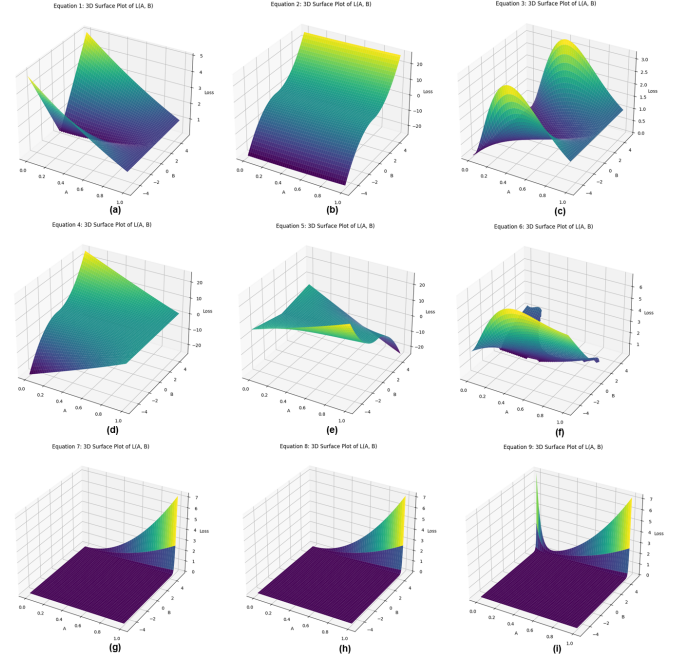


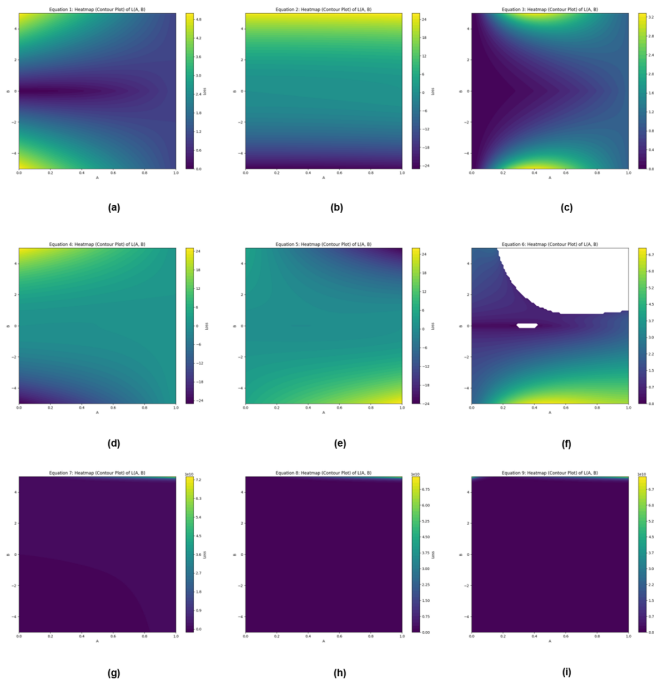
Figure 7. Visualization of 3D surface for 9 equations: (a) Eq. (1), (b) Eq. (2), (c) Eq. (3), (d) Eq. (4), (e) Eq. (5), (f) Eq. (6), (g) Eq. (7), (h) Eq. (8), and (i) Eq. (9).

$$+ B \cdot \frac{\partial L2}{\partial w} \cdot \frac{\partial \text{ReLU}(A \cdot \text{Step1} + B \cdot L2)}{\partial w} \right), \quad (17)$$

$$w_{\text{new}} = w_{\text{old}} - a \left( A \cdot \frac{\partial \text{ReLU}(A \cdot \text{Step1} + B \cdot L2)}{\partial w} \cdot \frac{\partial \text{Step1}}{\partial w} + B \cdot \frac{\partial L2}{\partial w} \cdot \frac{\partial \text{ReLU}(A \cdot \text{Step1} + B \cdot L2)}{\partial w} \right). \quad (18)$$

Eqs. (10) to (18) analyze the gradient descent of the equations in Eqs. (1) through (9). Utilizing the expression  $w \leftarrow w - a \frac{\partial L}{\partial w}$ , we obtained  $w_{\text{new}}$  from  $w_{\text{old}}$ . Each weight-update formula aimed to enhance the learning performance of the deep learning model through a specific combination of the loss and activation functions. Our findings indicate that the interplay of linear and nonlinear coefficients, static and dynamic factors, and ReLU and Bi-eLU activation functions can further refine the optimization process of the deep learning model.

Figures 7 and 8 illustrate the equations utilized in the experiment through 3D surface plots and heat maps. In Figures 7(a) and 8(a) depict a smooth transition between L1 and L2 losses, signifying a well-behaved loss function that smoothly interpolates between L1 and L2 losses based on the value of A. This smooth transition is crucial for optimization, as it mitigates abrupt changes in the loss landscape. Figures 7(b) and 8(b) demonstrate a linear relationship and exhibits a weighted combination of L1 and L2 losses, providing flexibility in adjusting



**Figure 8.** Visualization of heatmap for 9 equations: (a) Eq. (1), (b) Eq. (2), (c) Eq. (3), (d) Eq. (4), (e) Eq. (5), (f) Eq. (6), (g) Eq. (7), (h) Eq. (8), and (i) Eq. (9).

the influence of each component. Figures 7(c) and 8(c) reveal exponential growth, indicating high sensitivity to changes in L2, which can generate significant gradients beneficial for escaping flat regions in optimization, though it also poses a risk of instability. Figures 7(d) and 8(d) display dynamic behavior with a mix of linear terms, allowing for adaptive changes in the loss function and enhancing optimization by focusing on different components at various stages. Figures 7(e) and 8(e) resemble Figures 7(d) and 8(d) but with convex coefficients. The application of convex coefficients aims to ensure stability and smoother transitions in the loss landscape, which is essential for consistent optimization. Figures 7(f)-7(i) and 8(f)-8(i) depict the equations employed in hybrid loss functions. Figure 8(f) illustrates a more complex landscape due to the hybrid nature and exponential terms, capturing more intricate data patterns. Figure 8(g) shows both linear combinations and exponential growth, leveraging the strengths of both terms for balanced stability and responsiveness. Figure 8(h) and 8(i) pertain to ReLU and Bi-eLU equations, respectively. Figure 8(h) exhibits nonlinearity and sparsity, aiding in handling nonlinearity and ensuring sparsity in gradients, thus facilitating efficient optimization. Figure 8(i) demonstrates the activation of Bi-eLU, revealing complex nonlinear behavior that helps capture more intricate data patterns, thereby providing a more nuanced optimization

landscape.

**Qualitative analysis:** Visualizing the effect of the nonlinear exponential loss function weight for each loss function revealed distinct decision boundary. The shape of these decision boundaries supports stable optimization in deep learning models. Our findings indicate that a hybrid loss function, combining both fixed and dynamic properties, can slightly alleviate optimization challenges. However, when employing hybrid properties, the model exhibited correlated properties that aided in optimization. Conversely, performance of the model deteriorated when the heterogeneous properties were ineffective in learning. Experimental results confirm the importance of identifying an adaptive loss function suitable for the deep learning model.

The experimental results of the first component, the hybrid loss function, are as follows: The standard deviation is illustrated on a bar graph in Figure 9 to depict the dispersion degree of each method. The loss exhibited the most stable error performance when convex coefficients were utilized in hybrid v2, reflecting both nonlinear static and nonlinear dynamic information. Regarding deep learning model performance, hybrid v1, which combines the exponential moving average with L1 and L2 loss functions, demonstrated the best performance. This indicates that the deep learning model was trained stably and achieved high performance by incorporating hybrid properties. The reflection of nonlinear dynamic and static properties confirmed that the model could learn most stably and perform better compared to existing loss function methods.

Consequently, the deep learning model achieved stable learning by incorporating nonlinear dynamic and static properties. Our findings confirmed that reflecting nonlinear dynamics and linear fixed properties led to superior performance compared to existing loss function methods.

Table 2 presents the experimental performance of the proposed method. The results in Table 2 indicate that the hybrid loss function enhanced the segmentation performance of the deep learning model by 0.1%. Regarding loss reduction, the best experiment demonstrated an error decrease of 3% or more compared to the previous one. The hybrid loss function v2 exhibited the lowest loss, attributed to the nonlinear fixed and dynamic properties achieved through bi-directional nonlinear filtering. This suggests that the nonlinear properties were effectively incorporated into the model learning process within a consistent nonlinear environment.

Analysis of the relationship between loss and the loss function in Table 2 confirmed that the loss value is slightly higher when simple addition is employed. This addition suggests that



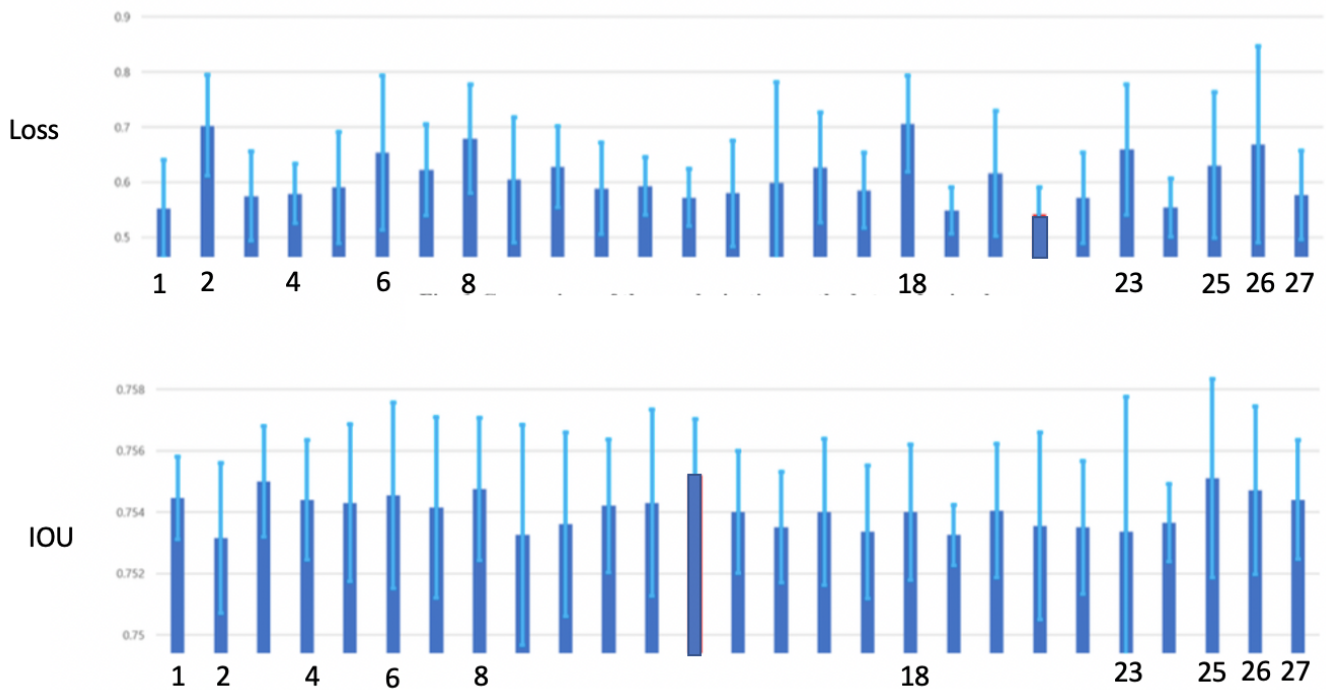


Figure 9. Visualization of comparison of the loss function methods: (a) loss values for different training epochs and (b) IOU values for different training epochs.

the complexity of information increases through superposition, resulting in a higher loss value. However, utilizing ReLU, eLU, Bi-ReLU, and Bi-eLU to reflect the degree of the loss function shows different effects. ReLU filtering reflects linear properties, reducing the loss compared to conventional addition. Filtering with eLU demonstrates that nonlinear properties are more stably reflected than linear properties. Bi-activation was then applied to learn by extracting essential properties through bipolar information reflection. The Bi-ReLU linear filtering effect shows a learning loss value similar to a single nonlinear filtering results. Furthermore, it was confirmed that Bi-eLU presents a simple linear effect when the nonlinear filtering effect is inadequate in both directions but shows a lower error value when essential properties are identified through nonlinear filtering.

Additionally, the performance of the loss function according to the change in seed value was analyzed. The analysis indicated that the performance of the loss function varied depending on the initial seed occurrence, confirming the necessity for the deep learning model to establish a stable initial state. Studying a robust loss function that performs well even in this initial state is essential. Hybrid test results showed better outcomes when dynamically reflected properties were effectively mixed, though not in all cases. To further develop the hybrid loss function, identifying an optimal combination of fixed and dynamic

properties is necessary. For Bi-eLU, the deletion of shared information and nonlinear sparse information reflected in deep learning model optimization was confirmed. The hybrid loss function found a lower error and slightly improved performance in the seed value experiment. Seed test results indicated that performance varied significantly depending on the initial state of the model, underscoring the need for robust loss function studies in the initial model state.

In Figure 9, the bar graph visualizes the mean values from the five-seed value experiments. The highest IOU test performance and the lowest learning error rate were achieved by incorporating nonlinear static and dynamic properties. The application of the activation function demonstrates a performance change as significant features are integrated into the model. When eLU and Bi-eLU, which reflect nonlinear properties, are applied, heterogeneous nonlinear features exhibit low error rates and maximum IOU performance in the deep learning model.

As shown in Table 2, similar performance metrics were observed across the various experiments of the U-Net model. Metrics such as DSC, F1-score, and IOU were consistent across experiments, indicating that the experimental conditions and hyperparameter settings were comparable. However, there were some variations in loss values between experiments. In the initial experiments, loss values were above 1.0 but gradually de-



Table 2. Experiment result of loss function using U-Net on ATR dataset with Seed 250

Loss function	DSC	F1-score	IOU	Loss	Precision	Recall
Experiment1	0.724	0.724	0.768	1.072	0.724	0.724
Experiment2	0.723	0.723	0.767	1.525	0.723	0.723
Experiment3	0.724	0.724	0.768	1.398	0.724	0.724
Experiment4	0.724	0.724	0.768	1.364	0.724	0.724
Experiment5	0.724	0.724	0.767	1.287	0.724	0.724
Experiment6	0.723	0.723	0.767	1.222	0.723	0.723
Experiment7	0.724	0.724	0.768	1.278	0.724	0.724
Experiment8	0.724	0.724	0.768	1.255	0.724	0.724
Experiment9	0.724	0.724	0.768	1.228	0.724	0.724
Experiment10	0.724	0.724	0.768	1.237	0.724	0.724
Experiment11	0.724	0.724	0.768	1.195	0.724	0.724
Experiment12	0.724	0.724	0.768	1.164	0.724	0.724
Experiment13	0.724	0.724	0.768	1.136	0.724	0.724
Experiment14	0.724	0.724	0.768	1.112	0.724	0.724
Experiment15	0.724	0.724	0.768	1.115	0.724	0.724
Experiment16	0.724	0.724	0.768	1.095	0.724	0.724
Experiment17	0.724	0.724	0.768	1.08	0.724	0.724
Experiment18	0.725	0.724	0.768	1.075	0.724	0.724
Experiment19	0.725	0.725	0.768	1.09	0.725	0.725
Experiment20	0.725	0.725	0.768	1.077	0.725	0.725
Experiment21	0.725	0.725	0.768	1.069	0.725	0.725
Experiment22	0.725	0.725	0.768	1.07	0.725	0.725
Experiment23	0.725	0.725	0.768	1.077	0.725	0.725
Experiment24	0.725	0.725	0.768	1.07	0.725	0.725
Experiment25	0.725	0.725	0.768	1.052	0.725	0.725
Experiment26	0.725	0.725	0.768	1.044	0.725	0.725
Experiment27	0.725	0.725	0.768	1.042	0.725	0.725

creased to below 1.0 in later experiments. This trend indicates a reduction in loss as the deep learning model is trained. The decrease in loss values was accompanied by improvements in accuracy metrics such as DSC, F1-score, and IOU. Thus, as the deep learning model training progressed, loss decreased and segmentation performance improved, which can be attributed to the progress of the model training.

The experimental results summarized the performance of various loss function as follows: no existing loss function < linear combination of L1 and L2 loss functions < exponential moving average to nonlinear exponential loss function < heterogeneous nonlinear loss function. It was confirmed that

incorporating nonlinear properties into deep learning models enhances learning and test performance.

## 6. Conclusion

We proposed a novel hybrid nonlinear loss function technique to encapsulate the semantic fixed and dynamic properties of a deep learning model during training. The effectiveness of the proposed method was demonstrated using five methods of analysis. Further analysis examined the relationship between the proposed loss function and existing loss functions, considering the reflection of linear and nonlinear properties on the covari-

ate coefficients of loss functions. Visualization and gradient descent analysis of the applied loss function were conducted, along with an assessment of the effect of seed value on hyperparameter optimization. Consequently, the results indicate that the proposed technique facilitated reliable convergence and optimization of the deep learning model, fully reflecting static and dynamic characteristics. Analysis of the relationship between the proposed and existing loss functions that only meaningful information must be filtered. Additionally, seed value analysis confirmed a slight increase in deviation; however, the proposed technique demonstrated greater robustness to changes due to small standard deviations compared to existing methods.

In future studies, the energy functions of the loss function will be defined to visualize their effects on model training. Additionally, visualization methods should be employed to identify optimal points by illustrating possible relationships within the model. The learning direction should also be optimized to filter and output only meaningful information, visualizing the complexity between information derived from loss and optimization areas.

## Conflict of Interest

No potential conflict of interest relevant to this article was reported.

## Acknowledgement

This study was conducted with support from Hansung University.

## References

- [1] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *Proceedings of the 21st International Conference on Machine Learning (ICML)*, Banff, Canada, 2004. <https://doi.org/10.1145/1015330.1015435>
- [2] P. Liu, G. Zhang, B. Wang, H. Xu, X. Liang, Y. Jiang, and Z. Li, "function discovery for object detection via convergence-simulation driven search," 2021 [Online]. Available: <https://arxiv.org/abs/2102.04700>
- [3] C. Wen, X. Yang, K. Zhang, and J. Zhang, "Improved loss function for image classification," *Computational Intelligence and Neuroscience*, vol. 2021, article no. 6660961, 2021. <https://doi.org/10.1155/2021/6660961>
- [4] Z. Xie, C. Shu, Y. Fu, J. Zhou, and D. Chen, "Balanced loss function for accurate surface defect segmentation," *Applied Sciences*, vol. 13, no. 2, article no. 826, 2023. <https://doi.org/10.3390/app13020826>
- [5] M. C. Dickson, A. S. Bosman, and K. M. Malan, "Hybridised loss functions for improved neural network generalization," in *Pan-African Artificial Intelligence and Smart Systems*. Cham, Switzerland: Springer, 2021, pp. 169-181. [https://doi.org/10.1007/978-3-030-93314-2\\_11](https://doi.org/10.1007/978-3-030-93314-2_11)
- [6] J. Yu and B. Wu, "Attention and hybrid loss guided deep learning for consecutively missing seismic data reconstruction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, article no. 5902108, 2021. <https://doi.org/10.1109/TGRS.2021.3068279>
- [7] L. Jia, A. Huang, X. He, Z. Li, and J. Liang, "A residual multi-scale feature extraction network with hybrid loss for low-dose computed tomography image denoising," *Signal, Image and Video Processing*, vol. 18, no. 2, pp. 1215-1226, 2024. <https://doi.org/10.1007/s11760-023-02809-3>
- [8] Z. Li, W. Shi, Q. Xing, Y. Miao, W. He, H. Yang, and Z. Jiang, "Low-dose CT image denoising with improving WGAN and hybrid loss function," *Computational and Mathematical Methods in Medicine*, vol. 2021, article no. 2973108, 2021. <https://doi.org/10.1155/2021/2973108>
- [9] O. Krakovska, G. Christie, A. Sixsmith, M. Ester, and S. Moreno, "Performance comparison of linear and non-linear feature selection methods for the analysis of large survey datasets," *PLOS One*, vol. 14, no. 3, article no. e0213584, 2019. <https://doi.org/10.1371/journal.pone.0213584>
- [10] W. Hu, "Nonlinear optimization in machine learning," 2020 [Online]. Available: <https://github.com/huwenqing0606/Nonlinear-Optimization-in-Machine-Learning>.
- [11] B. Luo, D. Liu, and H. N. Wu, "Adaptive dynamic programming for optimal control of nonlinear distributed parameter systems," in *Adaptive Learning Methods for Nonlinear System Modeling*. Oxford, UK: Butterworth-Heinemann, 2018, pp. 335-359. <https://doi.org/10.1016/B978-0-12-812976-0.00019-1>
- [12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation,"

in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*. Cham, Switzerland: Springer, 2015, pp. 234-241. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)

- [13] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” 2014 [Online]. Available: <https://arxiv.org/abs/1411.4038v1>.



**Seoung Ho Choi** received his B.S degree from Hansung University, Korea, in 2018 and M.S. degrees from Hansung University, in 2020. He joined the Faculty of Basic Liberal Art, College of Liberal Arts, at the Hansung University in 2021, where

he is an assistant professor.