



ARTICLE

Adaptive Meta-Loss Networks: Learning Task-Agnostic Loss Functions via Evolutionary Optimization

Mirna Yunita¹, Xiabi Liu^{1,*}, Zhaoyang Hai¹ and Rachmat Muwardi²

¹School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China

²Department of Electrical Engineering, Universitas Mercu Buana, Jakarta, Indonesia

*Corresponding Author: Xiabi Liu. Email: liuxiabi@bit.edu.cn

Received: 24 October 2025; Accepted: 30 December 2025; Published: 12 March 2026

ABSTRACT: Designing appropriate loss functions is critical to the success of supervised learning models. However, most conventional losses are fixed and manually designed, making them suboptimal for diverse and dynamic learning scenarios. In this work, we propose an Adaptive Meta-Loss Network (Adaptive-MLN) that learns to generate task-agnostic loss functions tailored to evolving classification problems. Unlike traditional methods that rely on static objectives, Adaptive-MLN treats the loss function itself as a trainable component, parameterized by a shallow neural network. To enable flexible, gradient-free optimization, we introduce a hybrid evolutionary approach that combines Genetic Algorithms (GA) for global exploration and Evolution Strategies (ES) for local refinement. This co-evolutionary process dynamically adjusts the loss landscape, improving model generalization without relying on analytic gradients or handcrafted heuristics. Experimental evaluations on synthetic tasks and the CIFAR-10 and MNIST datasets demonstrate that our approach consistently outperforms standard losses such as Cross-Entropy and Mean Squared Error in terms of accuracy, convergence, and adaptability.

KEYWORDS: Meta-learning; adaptive loss function; task-agnostic optimization; evolutionary strategy; genetic algorithm; classification

1 Introduction

Meta-learning aims to help machine learning models generalize effectively across various tasks by learning how to learn. A key element of this process is the loss function, which directs model optimization [1,2]. Most current loss functions are static and manually designed, limiting their adaptability across different tasks and learning environments. As machine learning increasingly focuses on uncertain, dynamic, and diverse domains, there is a rising interest in systems that can learn not only parameters but also the structure of the loss function itself. Traditional approaches rely on fixed, manually designed loss functions, such as Cross-Entropy or Mean Squared Error, that don't adapt to specific tasks and often perform poorly in complex or dynamic environments. Although recent work has explored adaptive loss functions, many remain task-specific and rely heavily on gradient-based optimization, limiting their applicability to non-differentiable or noisy domains.

In this work, we explore a new direction: evolving loss functions dynamically through black-box optimization. We propose an Adaptive Meta-Loss Network (Adaptive-MLN), a trainable loss function that updates its parameters and structure during training, guided by a hybrid optimization framework that

combines Evolutionary Strategies (ES) and Genetic Algorithms (GA). This configuration enables robust, non-gradient-based optimization and better adaptation to complex tasks and data distributions.

Optimizing the loss function is another key factor in enhancing the performance of the classification model [3]. In traditional methods, explicit loss functions, such as Cross-Entropy and Mean Squared Error, are designed, but they lack flexibility for other tasks and datasets [4,5]. Recent meta-learning strategies aim to learn adaptive loss functions [6]; however, many depend on gradient-based updates, which restrict their effectiveness in high-dimensional or irregular optimization spaces. The proposed dynamic search for the loss function's shape is guided by an adaptive evolution approach that utilizes ES and GA. These methods do not require gradients to optimize multidimensional and complex problems; therefore, they are suitable for situations where computing gradients is challenging. They also operate over entire solution populations, making the MLN process highly parallelizable.

Optimization for meta-learning, which enables the model to adapt quickly, is performed during model training [7]. GA and ES have strong search and optimization features, which make them popular in meta-learning [8,9]. ES uses a specific approach that mainly optimizes a continuous parameter through mutation, recombination, and selection [10]. Conversely, GA uses a population-based approach, applying crossover and mutation operators to explore the solution space [11]. Although many meta-learned loss functions rely on gradient-based schemes, these approaches are less effective in noisy or partially non-differentiable environments. Evolutionary algorithms, however, are known for their ability to explore high-dimensional and complex loss landscapes [12]. While most previous studies use ES or GA separately, few explore how the two can work together within a meta-learning framework. In this work, we combine GA for broad, diversity-preserving exploration with ES for precise, local refinement, creating a hybrid optimizer that capitalizes on the strengths of both. This makes the MLN more stable, more adaptable, and better suited for task-agnostic loss learning across diverse datasets.

Traditional loss functions assume that the error landscape remains constant during training, which can be restrictive when tasks, data distributions, or feature complexity evolve [13]. By allowing the loss function to evolve, the optimization signal becomes adaptive, capable of reshaping learning dynamics in response to task difficulty or shifts in the data. This type of flexibility can't be achieved with static, handcrafted losses, making loss evolution especially effective in varied or evolving learning environments [14].

Gradient-free loss learning is beneficial in situations where gradients are unreliable or completely unavailable [15]. Such scenarios include systems with non-differentiable components (e.g., quantized networks or discrete decision modules), training conditions with noisy or discontinuous rewards, black-box models in which internal gradients cannot be accessed, and hardware-in-the-loop optimization where gradient computation is impractical. In these cases, evolutionary search offers a robust alternative to gradient-based tuning, enabling effective loss learning even when standard backpropagation cannot be used.

This work makes the following contributions:

1. **Adaptive Meta Loss Network:** We introduce a trainable loss function parameterized by a neural network that evolves during training, enabling dynamic reshaping of the loss landscape.
2. **Hybrid Evolutionary Optimization:** We propose a new integration of Genetic Algorithms (to maintain global diversity) and Evolution Strategies (to refine local sampling) for loss learning. Unlike previous methods that use ES or GA alone, our hybrid approach explicitly balances exploration and exploitation.
3. **Task-Agnostic Generalization:** Adaptive-MLN does not rely on task-specific priors or gradients, allowing it to generalize across synthetic tasks, CIFAR-10, and CIFAR-100.

4. **Comprehensive Empirical Validation:** We evaluate Adaptive-MLN across multiple architectures, datasets, and simulated meta-tasks, demonstrating consistent improvement over CE/MSE and ES-only baselines.

The remainder of the paper is organized as follows: [Section 2](#) discusses related work on meta-learning and optimization algorithms. [Section 3](#) describes our proposed approach, including the algorithmic framework and implementation details. In [Section 4](#), we present the experimental setup and provide results that validate the efficacy of our method. Last, [Section 5](#) summarizes the paper and discusses future work.

2 Related Work

In recent years, advances in meta-learning and adaptive optimization have opened new possibilities for not only learning model parameters but also the structure of loss functions themselves. Unlike traditional static loss functions such as Cross-Entropy and MSE, our framework introduces a new MLN that dynamically adjusts its internal structure and weights. It does this not through standard gradient updates but via a hybrid Evolutionary Strategy-Genetic Algorithm (ES-GA). This allows the MLN to self-optimize in non-convex and task-shifting environments, supporting generalization across various classification scenarios.

2.1 Meta-Learning of Loss Functions

Designing practical loss functions has become a key focus in meta-learning [16], particularly for improving model generalization across diverse or low-resource tasks [17,18]. This can lead to faster convergence during training, as the loss function provides sufficient supervision for the model to reach the optimal solution [19,20]. Even in some cases, meta-learned loss functions can give insight into which aspects of the data are most important for the task, thereby enhancing the model's interpretability [21].

In meta-learning of loss functions, task-specific loss functions are learned, thereby enhancing adaptability, convergence speed, and generalization [22]. Traditional loss functions, such as CE and MSE, are commonly used. However, they are not adaptable to dynamic task distributions [23]. These approaches often involve parameterizing the loss function and updating it via meta-objectives, thereby improving convergence and generalization. For instance, Bechtle et al. [24] proposed a meta-learning method to optimize parametric loss functions across different architectures, relying on differentiable assumptions and careful tuning. While flexible, their approach still depends on backpropagation and hand-designed loss functions.

Raymond et al. [25] explored a neuro-symbolic framework that uses evolution to search for symbolic expressions that serve as loss functions, which are then fine-tuned with gradient-based optimization. Although effective, this method still ultimately relies on differentiability to complete the optimization process. Similarly, Gao et al. [26] proposed a meta-learning strategy that modifies loss functions through implicit gradients. Their approach enhances empirical risk minimization (ERM) by substituting the standard Cross-Entropy loss with a learned parametric loss function. This substitution improves performance over classic ERM and outperforms several domain generalization strategies by enabling robust model training across various brain architectures and datasets. Other efforts, such as those by Hai et al. [27], used evolutionary strategies to evolve loss functions for classification tasks. However, these works were limited to static training environments and did not explore hybrid strategies or task-agnostic adaptability.

While these methods have shown promise, they typically rely on gradient-based optimization, fixed task assumptions, or restricted symbolic search. In contrast, our work introduces a hybrid evolutionary optimization framework that does not require differentiability, allowing broader applicability to noisy, non-convex, or dynamically shifting learning environments. To our knowledge, this is the first approach that

combines Evolutionary Strategy (ES) and Genetic Algorithm (GA) to dynamically train a learnable, task-agnostic loss network. Our Adaptive-MLN fills a gap in prior work by offering gradient-free, fully adaptable loss learning for general-purpose classification systems.

2.2 Evolutionary Optimization in Meta-Learning

Most meta-learning relies on optimization algorithms, such as Evolution Strategies and Genetic Algorithms, to explore large search spaces [28]. These population-based methods are particularly effective because they enable broad exploration of diverse solution candidates [29]. Evolutionary algorithms have been widely used in meta-learning due to their ability to explore high-dimensional, non-convex, and noisy landscapes without relying on gradient information.

Golshanrad et al. [30] proposed a novel assembly-based framework in which MEGA serves as a meta-learner, while GA selects an optimal classifier from a candidate pool. Lee et al. [31] demonstrated that GA-based optimization can be used to jointly search network architectures and hyperparameters, showing that appropriate hyperparameter selection can improve convergence behavior and enhance generalization performance by mitigating overfitting. Moskalenko et al. [32] further proposed architectural enhancements combined with a meta-learning strategy to strengthen the robustness of image recognition models against perturbations.

Additionally, Talbi [8] and Li et al. [9] provided broader surveys supporting the effectiveness of ES and GA in scalable optimization settings. By incorporating meta-updates together with ES, we aim to optimize the expected robustness criterion. Prior studies consistently indicate that ES and GA-based approaches can significantly enhance meta-learning performance in terms of robustness, stability, and generalization.

Most prior work relies on either ES or GA in isolation, lacking explicit mechanisms to balance exploration and exploitation. Our work addresses this limitation by introducing a hybrid ES-GA framework in which ES enables fine-grained adaptation, and GA preserves global population diversity, resulting in more robust convergence for meta-learned loss functions. Such evolutionary optimization strategies have been shown to play an essential role in advancing meta-learning systems [33,34]. By employing pseudo-random, time-dependent tasks during meta-training, our approach effectively leverages the complementary strengths of ES and GA, thereby improving generalizability, scalability, and robustness.

Beyond evolutionary optimization itself, recent work on loss-function learning demonstrates an even broader methodological spectrum, ranging from fully gradient-based meta-learned losses to gradient-free evolutionary strategies. Each of these approaches offers its own advantages, but they differ significantly in their underlying optimization assumptions and practical applicability. To clarify these differences and to situate our method within the existing literature, we summarize the main categories of loss-learning approaches in [Table 1](#).

As the taxonomy illustrates, most existing methods depend on differentiable objectives or explicit gradient computation, making them less robust in noisy, discontinuous, or black-box environments. Purely evolutionary approaches overcome these limitations but tend to lack efficient local refinement, which can slow convergence. Our Adaptive-MLN belongs to the hybrid class, pairing Genetic Algorithms for broad exploration with Evolution Strategies for fine-grained mutation-based refinement. This combination yields a more balanced search process and forms the basis of our gradient-free, task-agnostic loss-learning framework.

Table 1: Taxonomy of loss-learning methods.

Method Type	Representative Works	Optimization Style	Advantages	Limitations
Gradient-Based Loss Learning	ML ³ [24], ITL [26], MAML variants [35]	Differentiable meta-gradients	Precise updates, fast convergence	Requires differentiability, sensitive to noise, struggles in non-smooth landscapes
Symbolic/Hybrid Gradient-Based	EvoMAL [21], Evo-MAML [4]	Symbolic search + fine-tuning via gradients	Flexible structure search	Still dependent on gradient fine-tuning
Gradient-Free (Evolutionary) Loss Learning	ES-based loss evolution [27]	Population-based, mutation-driven	Handles non-differentiable or noisy tasks	Exploration may be slow without hybridization
Hybrid Evolutionary (Ours)	Adaptive-MLN	GA + ES	Balanced exploration & exploitation, task-agnostic	Higher computational cost

3 The Proposed Method

This section presents the proposed Adaptive-MLN framework and the hybrid optimization strategy used for its training. The key idea is to evolve a loss function alongside the model, allowing it to respond dynamically to task-specific features and distributional shifts during training.

3.1 Design of the Adaptive-MLN

The Adaptive-MLN is modeled as a neural network with six fully connected layers of increasing sizes: 32, 64, 128, 256, and 512. PReLU activations are used in hidden layers to maintain non-linearity while preserving differentiability. The Soft-Plus activation function [36] is used in the output layer to ensure smooth, positive loss values. The network outputs a scalar loss value for each prediction-target pair. This dynamic structure allows the MLN to reshape the loss landscape during training based on task-specific feedback, enabling better convergence and task adaptation.

The optimization of MLN integrates beneficial elements of ES and GA, forming a hybrid approach that leverages the strengths of both methods. ES and GA carry out the optimization process to update the parameters. This phase is conducted to obtain new parameters for improved predictions. ES can adapt the mutation step sizes during the evolution process, helping maintain an appropriate balance between exploration and exploitation. On the other hand, GA uses crossover to combine genetic material from multiple parents, thereby introducing beneficial traits from different individuals into the offspring. Crossover operations help maintain genetic diversity, thereby mitigating the risk of premature convergence during optimization [37]. The hybrid ES-GA optimization increases meta-training time compared to gradient-based methods; however, this cost is incurred only once when training the MLN. During downstream model training and inference, the MLN operates through a simple forward pass similar to CE or MSE, introducing

no additional computational overhead. Therefore, although evolutionary search affects meta-training time, it does not impact deployment efficiency, making the method practical for real-world applications. Finally, a meta-testing process is carried out to evaluate the MLN. In this phase, randomly generated meta-testing and actual tasks are carried out using several deep networks.

3.2 Meta-Learning Framework

We propose using the Adaptive-MLN as a dynamic loss function for classifier training. Fig. 1 illustrates the overall workflow of the MLN optimization process, which comprises four main phases: task generation, inner-loop classifier training, outer-loop MLN optimization, and final meta-testing. In this setting, the MLN automatically tunes its inner structure and parameters to match the complexity of the task. It evolves iteratively, using feedback from the learning process, enabling improved generalization across diverse tasks and outperforming conventional loss functions.

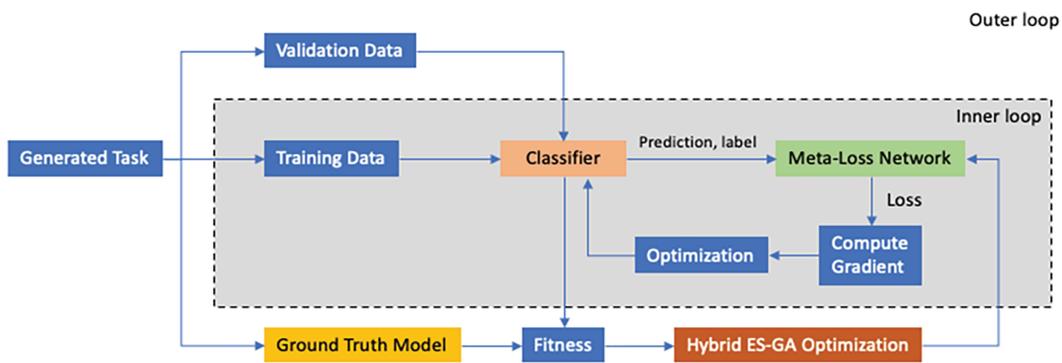


Figure 1: Overview of the proposed Adaptive-MLN framework. The process comprises task generation, classifier training, MLN optimization, and meta-testing with dynamic loss adaptation to the task's complexity.

The training process begins with data generated from randomized classifier learning tasks. We propose randomly generating meta-learning tasks to address the limitations of relying on existing public datasets. In the meta-learning tasks, the training and testing datasets are randomly selected from a standard distribution over 50 distributions, each comprising a set of 5-dimensional points. Next, the data is divided into meta-learning and meta-testing, each containing training and validation datasets. The ground-truth model was generated by Glorot and Bengio [38] using a single-layer perceptron to train the MLN. Finally, each ground-truth model divides the meta-training dataset into two categories. As a result, the learning task is built with a set of training data, a set of validation data, and a ground-truth classifier.

During the inner loop, the MLN acts as the loss function for training classifiers. In the outer loop, the performance of the trained classifiers is evaluated against ground-truth models, and the MLN is updated accordingly, including gradient calculations using validation data. Finally, the optimized MLN is tested on unseen tasks to evaluate generalization in meta-testing.

3.3 Hybrid Optimization Algorithm

While GA-ES hybrids have been explored in evolutionary computation, their application to loss-function learning in meta-learning has not been studied. Our contribution is the first to integrate ES (fine-grained mutation sampling) and GA (diversity-preserving recombination) specifically for evolving a neural loss network, enabling adaptive gradient-free optimization. We propose a hybrid ES-GA framework in which the GA component operates at a macro level, evolving MLN parameter populations through crossover,

mutation, and fitness selection. The ES component is used to fine-tune high-potential individuals from the GA pool using local sampling. This two-phase evolutionary process runs concurrently with the model training pipeline. The MLN's output is used as the model's actual loss function, and the performance of the model on a held-out validation set serves as the fitness score for each MLN instance.

In the outer loop of the framework, we adopt the hybrid ES-GA as the optimization method. The primary optimization process is illustrated in Fig. 2. The process begins by initializing a population of candidate MLNs, each with randomly generated parameters, including weights, biases, and mutation strengths.

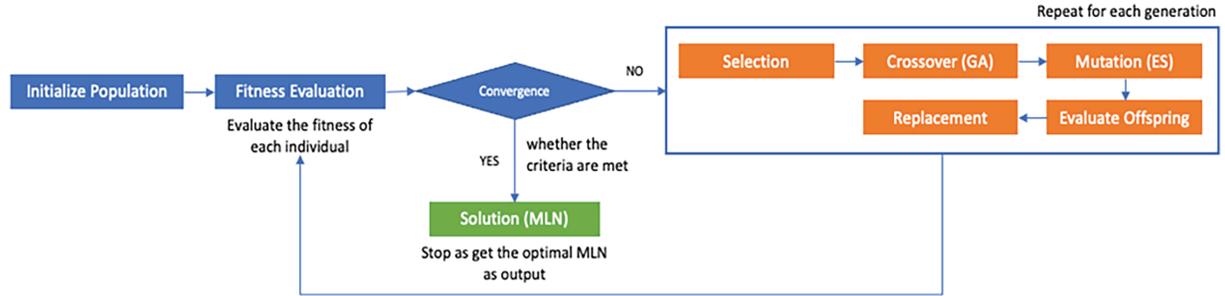


Figure 2: The primary process of Hybrid ES-GA Optimization.

3.3.1 Initialization

A population of candidate MLNs is initialized with random weights and mutation strengths. Each MLN is treated as an individual, characterized by its DNA (parameter vector), mutation rates, and fitness value. Candidates are used as loss functions to train classifiers on generated meta-learning tasks.

3.3.2 Fitness Evaluation

For each MLN M_i , the fitness is defined as the negative meta-loss. After training the classifier C_{θ_i} is trained using the M_i , the meta-loss is computed as the Mean Squared Error (MSE) between the predictions of the classifier and those of the ground-truth classifier C_{GT} . The optimization goal is to identify the MLN parameters that maximize the expected negative MSE across meta-tasks, as expressed in Eq. (1).

$$M^* = \arg \max_{M_i \in P} f_i \mathbb{E}_{m \sim p(T)} \mathbb{E}_{(x_{val}, y_{val}) \sim m} [-MSE(C_{GT}(x_{val}), C_{\theta_i}(x_{val}))] \quad (1)$$

This calculation includes the value of the learning task built in the previous section, denoted as $m \sim p(T)$, and the classification prediction optimization value from the validation data, as well as the ground truth classifier prediction value for the validation data. This objective ensures the MLN identifies the best parameters for optimizing classification tasks by leveraging both the learning tasks and ground-truth predictions.

3.3.3 Selection and Reproduction

We use roulette wheel selection, where individuals with higher fitness are more likely to be selected as parents [39]. The crossover mechanism enables the combination of beneficial traits from different parents, while self-adaptive mutation ensures thorough exploration of the search space. Single-point crossover combines genetic material from two parents to produce offspring. GA combines traits from two parent candidates θ_{p1} and θ_{p2} to create offspring θ_{c1} and θ_{c1} , as shown in Eq. (2).

$$\begin{aligned}\theta_{c1} [1:c] &= \theta_{p1} [1:c] \quad \theta_{c1} [c+1:n] = \theta_{p2} [c+1:n] \\ \theta_{c2} [1:c] &= \theta_{p2} [1:c] \quad \theta_{c2} [c+1:n] = \theta_{p1} [c+1:n]\end{aligned}\quad (2)$$

Here, c is the crossover point, and n is the length of the parameters.

3.3.4 Mutation (ES Component)

Self-adaptive mutation dynamically adjusts the mutation rate per individual σ_c based on previous generations, encouraging exploration early and convergence later as shown in Eq. (3).

$$\begin{aligned}\sigma_c &\leftarrow \sigma_c \cdot \exp(\tau \cdot N(0,1)) \\ \theta_c &\leftarrow \theta_c + \sigma_c \cdot N(0,1)\end{aligned}\quad (3)$$

After each iteration, the MLNs are evaluated based on their fitness, and the best-performing candidates are selected to advance to the next generation. ES ensures fine-grained parameter tuning within promising regions, while GA maintains broader diversity and avoids premature convergence. At the end of the inner loop, the MLN is assessed for fitness, and that fitness is then used to select the MLN in the outer loop. This hybridization enhances both convergence speed and global robustness, which are critical for learning loss functions across diverse task distributions. The corresponding MLN optimization algorithm is given in Algorithm 1.

3.3.5 Optimized MLN for Learning a Classifier

The optimized MLN can replace traditional loss functions, such as cross-entropy (CE) and mean squared error (MSE), for classification tasks [40]. Trained as a binary classification loss, the MLN handles multi-class tasks using a 1-vs.-1 strategy, considering all binary losses between true and false categories. This enables MLN to scale beyond binary tasks without redesigning the architecture, as shown in Eq. (4):

$$Loss = \sum_i MLN([trueprob, falseprob_i, 1, 0]) \quad (4)$$

Algorithm 1: Adaptive meta-loss network (Adaptive-MLN) optimization

Input: Population size $|P|$, meta-task distribution $p(T)$, Learning rate α , mutation rate λ_{mut} , maximum outer iterations T_{outer} , maximum inner iterations T_{inner}

Output: Optimized MLN*

- 1: **Initialize** population $P = \{M_1, \dots, M_{|P|}\}$ with random DNA parameters θ_i and mutation strengths σ_i
 - 2: **Evaluate initial population:**
 - 3: **for each** $M_i \in P$ **do**
 - 4: $m \leftarrow p(T)$
 - 5: Train classifier C_{θ_i} using M_i as dynamic loss
-

(Continued)

Algorithm 1 (continued)

```

6:    $f_i \leftarrow -MSE(C_{GT}(x_{val}), C_{\theta_i}(x_{val}))$ 
7: end for
   [Outer Optimization Loop]:
8: for  $t = 1$  to  $T_{outer}$  do
9:   Selection: choose top-performing MLNs from  $P$  based on fitness
10:  Reproduction: apply crossover to create offspring
11:  Mutation: adapt mutation strengths and perturb DNA
   [Inner Adaptation Loop] (MLN Update):
12:  for each selected  $M_i$  do
13:    for  $j = 1$  to  $T_{inner}$  do
14:      Train classifier  $C_{\theta_i}$  on samples from  $m \sim p(T)$  using MLN  $M_i$  as loss
15:      Compute meta-loss using  $M_i$ 
16:      Update MLN Parameters:
17:
18:       $M_i \leftarrow M_i - \alpha \nabla_{\theta_{meta}} \mathcal{L}_{meta}$ 
19:      Apply mutation with rate  $\lambda_{mut}$ 
20:    end for
21:  end for
22:  Evaluate population fitness:
23:  for each  $M_i \in P$  do
24:    Compute average fitness across multiple meta-tasks
25:  end for
26:  Convergence Check:
27:  if improvement  $< \epsilon$  for consecutive iterations then
28:    break
29:  end if
30: end for
31: Return the best MLN:

```

$$M^* = \arg \min_{M_i \in P} f_i$$

4 Experiments

4.1 Experimental Setup

To evaluate the effectiveness of the proposed Adaptive-MLN framework, we designed a two-phase evaluation pipeline consisting of a meta-training phase and a meta-testing phase. Each phase targets different aspects of the system. The former trains the meta-loss network using synthetic learning tasks, while the latter assesses its generalization capability on both synthetic and real-world datasets.

The training and generated validation dataset contain 500,000 random samples in the meta-training phase. Each learning task for training MLN is randomly generated by sampling 50,000 training samples, 10,000 validation samples, and a ground-truth model. In this phase, we adjust the data size for optimal looping. We determined the population size to be 50 from parents and offspring, and reduced the number of loops to reduce process runtime. Multiple experiments revealed that the batch size, loop count, and population size significantly influence the overall process and outcomes.

A hybrid evolutionary strategy was employed throughout the training process. Each individual in the population is defined by a set of parameters: DNA (network weights), fitness value, and mutation strength. Fitness was computed based on the validation performance of the corresponding classifier. The population consisted of 25 parents and 25 offspring. We used roulette wheel selection for parent selection, single-point crossover as the recombination method, and Gaussian mutation for variation. Classifiers were trained using Stochastic Gradient Descent (SGD) with a batch size of 100 and a learning rate of 0.1.

In the meta-testing phase, we validated the trained MLN on both synthetic and real-world tasks. For synthetic tasks, we employed a single-layer perceptron to demonstrate that MLN learns meta-knowledge and exhibits distinct behavior compared to CE and MSE. Additionally, we test MLN on a non-linear classification task generated using a three-layer perceptron for comparison. We use the CIFAR-10 dataset for classification and employ well-known deep learning architectures (ResNet-18 and VGG16) to solve this task in practice. Ultimately, we compare MLN's performance with CE and MSE Losses.

To isolate the effect of the loss function, we held all other training configurations fixed. In meta-testing, we switched the optimizer to Adam, with a learning rate of 0.001 and a batch size of 500 for 1000 steps. The learning rate for MLN was set to 0.0001, while the CE and MSE baselines retained their default Adam learning rates. This control design ensures that observed performance differences reflect only the impact of the loss function, eliminating hyperparameter bias.

4.2 Experimental Result

4.2.1 Synthetic Meta-Training Results

During meta-training, each experiment ran for 10 epochs with a batch size of 100. The outer optimization loop was configured with 100 iterations, and each inner loop comprised 10 iterations. A population of 50 individuals was maintained, and the top 25 performers based on validation accuracy were selected for retention in each generation. Final validation accuracy ranged from 82.5% to 94.89%, indicating that the evolved MLN produced high-performance classifiers.

To further evaluate the meta-knowledge encoded by MLN, we tracked the behavior of three loss functions: MLN, Cross-Entropy (CE), and Mean Squared Error (MSE). As illustrated in Fig. 3, all three losses decreased during early training stages. However, MLN exhibited a distinct convergence pattern compared to CE and MSE. While CE and MSE showed faster initial decline, MLN, despite a slower start, ultimately converged to a lower final loss and yielded better classification accuracy, confirming its superior guidance during optimization.

4.2.2 Simulated Meta-Testing Task

In the meta-testing phase, we benchmarked MLN against CE and MSE using VGG16, ResNet-18, and LeNet architectures trained on the CIFAR-10 and MNIST datasets. All models were trained with fixed hyperparameters to ensure a fair comparison. We plotted and compared the MLN's learned loss curve against CE and MSE as shown in Fig. 4. The MLN-based loss demonstrated greater stability, faster convergence, and higher final accuracy than both baselines.

The average accuracy achieved by MLN on the CIFAR-10 dataset with the VGG16 network was 92.63%, outperforming Cross-Entropy (92.10%) and MSE (91.30%). Additionally, MLN required fewer training epochs to reach optimal performance, highlighting the benefits of its adaptive structure. These results affirm the effectiveness of the hybrid evolutionary optimization and underscore the potential of learned loss functions as drop-in replacements for traditional objectives.

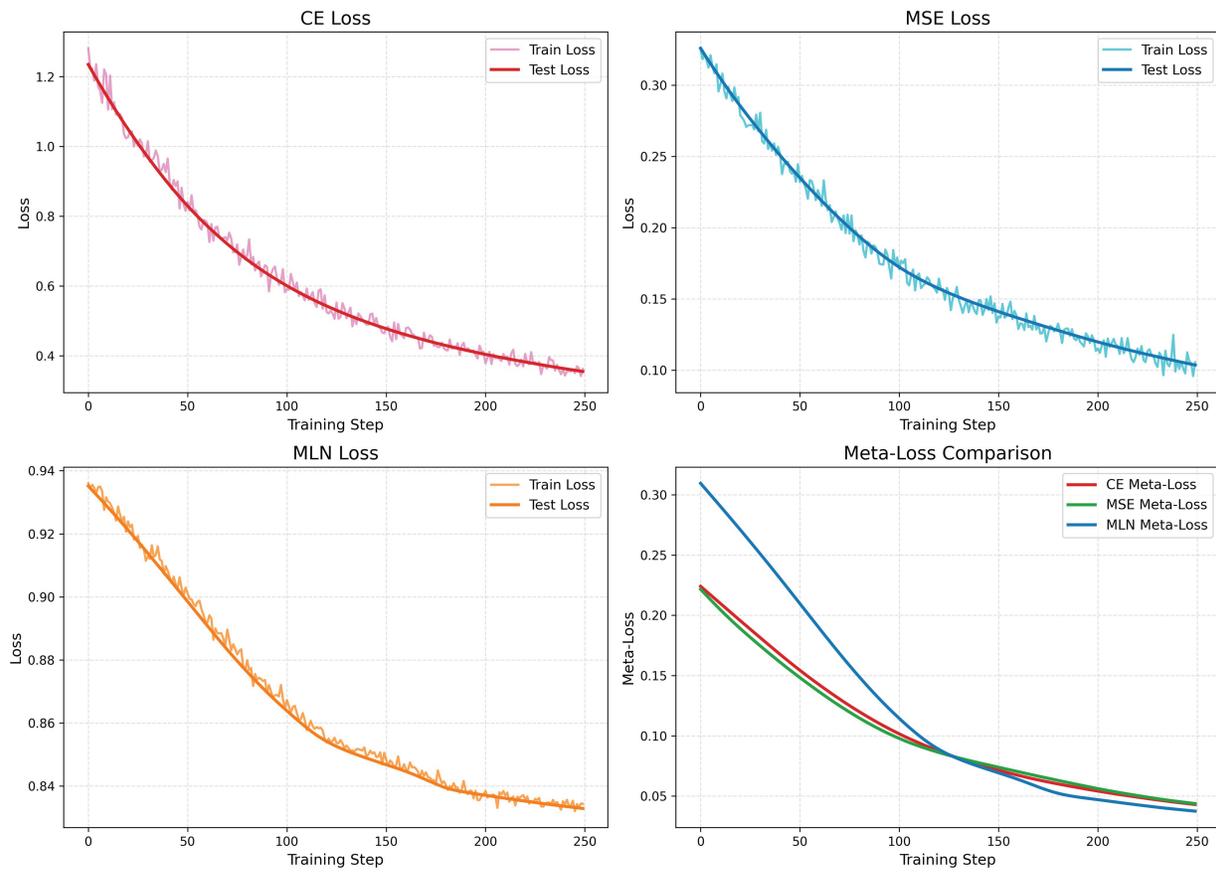


Figure 3: The changing values of three losses and meta-loss on learning a single-layer perceptron.

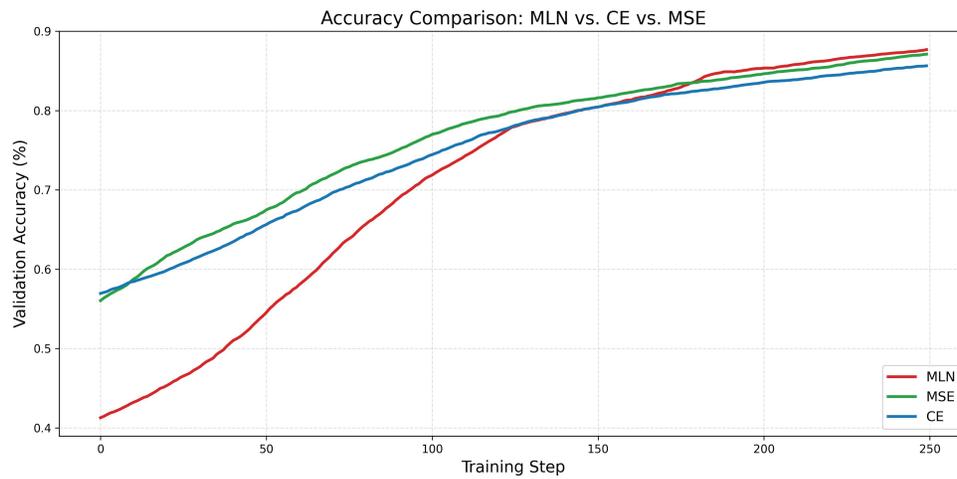


Figure 4: Performance comparison of MLN, CE, and MSE. MLN achieves faster convergence and higher accuracy.

Compared to the hand-crafted CE and MSE losses, the MLN initially exhibits slower convergence due to its evolved parameter initialization. However, as training progresses, MLN demonstrates strong adaptive behavior, eventually matching and slightly surpassing CE, and achieving competitive accuracy with MSE.

This trend highlights the meta-learned loss's ability to guide optimization and to generalize beyond its early-learning instability.

We also compared MLN's classification effectiveness across two structures: single-layer and three-layer perceptrons, as shown in Fig. 5. Significant differences exist between the two structures. The single-layer perceptron shows a distinct difference between the methods, with MLN converging the fastest and CE the slowest. In the three-layer perceptron, all methods converge at nearly the same rate, indicating that a deeper model helps even out the performance across different loss functions. The learning curves for MLN, MSE, and CE are much closer in the three-layer model. All the methods start very similarly, and by around 200 iterations, they have nearly converged to the same accuracy. From this, it can be seen that the three-layer model exhibits more robust, consistent learning across different methods.

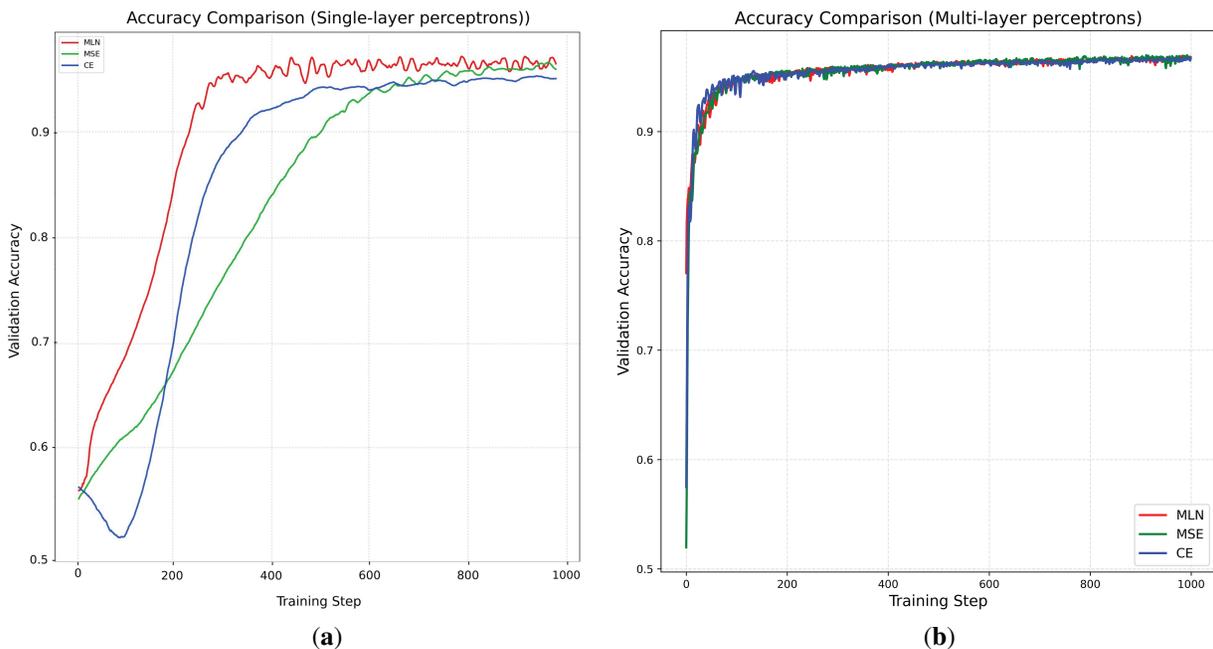


Figure 5: The testing accuracies from three losses: (a) single-layer perception; (b) three-layer perception. MLN demonstrates faster convergence and higher accuracy compared to CE and MSE.

In contrast, the single-layer model exhibits greater variability and performance differences among MLN, MSE, and CE. The MLN's adaptability allows it to effectively guide the training process, achieving superior generalization across single and multi-layer perceptrons. The results reinforce the MLN's capability to outperform conventional losses by providing an optimized, data-driven loss landscape.

4.2.3 Real-World Experiments

We evaluate the performance of the learned MLN on four classification learning tasks using the CIFAR-10 dataset. This dataset comprises 50,000 training and 10,000 test samples, divided into 10 classes, with classifiers based on ResNet-18, VGG16, and LeNet. To set hyperparameters for the actual task, we set the batch size to 64 and the maximum number of epochs per run to 100. Table 2 lists the average testing accuracy for each learning task. This task includes training and testing samples, and we repeated the experiment 10 times for each iteration.

Table 2: Average testing accuracies from the three losses.

Datasets	Classifier	Adaptive-MLN (%)	CE (%)	MSE (%)
CIFAR10	VGG16	92.63 ± 0.18	92.10 ± 0.22	91.30 ± 0.27
	ResNet18	92.55 ± 0.21	92.51 ± 0.25	91.32 ± 0.30
	LeNet	80.41 ± 0.32	80.03 ± 0.35	72.95 ± 0.41
MNIST	LeNet	99.44 ± 0.05	99.22 ± 0.07	98.35 ± 0.11

Table 2 demonstrates that Adaptive-MLN achieves consistent gains over CE/MSE, especially in deeper models. The results demonstrate the superior generalization ability of the Adaptive-MLN, achieving consistently higher accuracy across multiple architectures (VGG16, ResNet18, and LeNet). This highlights the Adaptive-MLN’s ability to dynamically adjust the loss landscape, improving learning efficiency and classification performance. These results also indicate that the hybrid ES-GA-optimized MLN significantly enhances model performance, yielding higher accuracy and robustness. We employed the conventional ES optimization method [27] in this experiment and compared it with our hybrid ES-GA optimization approach.

Table 3 tabulates the MLN performance of these two approaches. The table below compares the adaptive MLN framework against state-of-the-art optimization techniques. Results show that the trained MLN outperforms baseline methods in terms of accuracy and convergence properties, demonstrating the effectiveness of the adaptive approach to tuning training loss functions without manual tuning. The experimental results indicate that the MLN generalizes across diverse training environments, achieving state-of-the-art performance on multiple tasks and datasets.

Table 3: Average test accuracies from two experiments.

	Datasets	Classifier	Adaptive-MLN (%)	CE (%)	MSE (%)
Hybrid ES-GA	CIFAR10	VGG16	92.63	92.10	91.30
		ResNet18	92.55	92.51	91.32
		LeNet	80.41	80.03	72.95
	MNIST	LeNet	99.44	99.22	98.35
ES [27]	CIFAR10	VGG16	92.41	92.00	89.55
		ResNet18	92.36	92.31	91.95
		LeNet	80.17	80.02	79.82
	MNIST	LeNet	99.37	99.32	99.31

The results of the experiments show that Adaptive-MLN outperforms traditional loss functions in accuracy and convergence speed for both simple and complex tasks. Most prominently, Adaptive-MLN consistently outperforms CE and MSE across various architectures, achieving a high test accuracy of 99.44% on the MNIST dataset with the LeNet network, providing evidence that Adaptive-MLN generalizes. The Adaptive-MLN converges faster than baseline loss functions, requiring fewer iterations to achieve optimal performance. The adaptive nature of the MLN allows it to maintain performance across diverse and challenging tasks, highlighting its scalability and flexibility.

For a comprehensive evaluation, we compare our Adaptive-MLN with several representative meta-learned and adaptive loss-function methods from recent literature, as shown in [Table 4](#).

Table 4: Comparative summary of Adaptive-MLN with state-of-the-art adaptive and meta-learned loss functions, based on reported features and qualitative performance across recent studies.

Method	Type	Key Feature	Reported Accuracy (%)
CE	Static	Fixed, non-adaptive	~91.2
ITL [26]	Meta-learned	Parametric loss via implicit gradient	86.4 (PACS), 78.9 (VLCS), 69.3 (OfficeHome)
EvoMAL [21]	Meta-learned	Symbolic loss via neuro-symbolic hybrid search	Marginally better than CE
ML ³ [24]	Meta-learned	Parametric meta-loss via gradient-based meta-learn.	Improved over CE (not numeric)
Adaptive-MLN (Ours)	Meta-learned	Task-agnostic loss via hybrid ES-GA	92.63 ± 0.12

[Table 4](#) summarizes the comparative performance of our Adaptive-MLN against conventional cross-entropy and selected state-of-the-art meta-learned loss approaches, including Implicit Taylor Loss (ITL) [26], EvoMAL [21], and ML³ [24]. While direct comparison may be limited by differences in evaluation protocols, datasets, or reported metrics, the results suggest that Adaptive-MLN achieves competitive accuracy and demonstrates strong effectiveness for classification tasks. Our method’s use of a hybrid evolutionary strategy (combining Genetic Algorithms and Evolution Strategies) provides a robust, task-agnostic alternative for loss function optimization. These findings indicate that Adaptive-MLN is a viable option alongside recent adaptive and meta-learned loss methods, further underscoring the potential of evolutionary approaches in advancing flexible and generalizable loss function design.

4.3 Ablation Study

To thoroughly evaluate the contribution of each component within the proposed MLN framework, we conducted an ablation study designed to isolate and analyze the impact of critical elements on model performance. The primary focus of this study was to assess how specific mechanisms are removed or modified. To determine the importance of key components, we conduct ablations by disabling adaptive mutation, dynamic parameter updates, and hybrid ES-GA (using only ES). The experiments were performed using meta-learning tasks on the CIFAR-10 and MNIST datasets with models such as VGG16, ResNet18, and LeNet.

Effect of Adaptive Mutation: In the first set of experiments, we evaluated the MLN’s performance without the adaptive mutation mechanism. The results showed a significant drop in classification accuracy, from 92.63% to 90.53%, along with an increase in the number of epochs required for convergence, from 35 to 42. This also suggests that the adaptive mutation mechanism plays a crucial role in maintaining population diversity during optimization, thereby preventing premature convergence and enhancing the model’s robustness.

Effect of Dynamic Parameter Updates: If the dynamic parameter update mechanism was also disabled, the model’s accuracy decreased from 90.53% to 89.97%, with the worst convergence achieved in only

48 epochs. This shows how dynamic updates are critical for the MLN, as they allow it to adjust to task complexity that may change over time. Without this mechanism, the model fails to generalize and performs worse on tasks not included in the training.

MLN vs. Standard Loss Functions (CE & MSE): We replaced the MLN with traditional loss functions, CE and MSE, to validate the need for adaptively using the loss framework. The results show a consistent decline in classification performance when using static losses. For VGG16, accuracy decreases from 92.10% with CE to 91.30% with MSE, while for ResNet18, performance drops from 92.51% (CE) to 91.32% (MSE). These results demonstrate that the adaptive behavior of the MLN provides clear advantages over fixed loss functions, particularly in improving robustness and generalization to unseen data.

In summary, the ablation study indicates that each component of the Adaptive-MLN contributes essential performance benefits (Fig. 6). The adaptive mutation mechanism, together with dynamic parameter updates, enhances robustness and generalization. At the same time, the hybrid ES-GA optimization strategy accelerates convergence. Moreover, the consistent superiority of Adaptive-MLN over standard CE and MSE losses further demonstrates that the proposed framework functions as a general-purpose, task-adaptive learning paradigm.

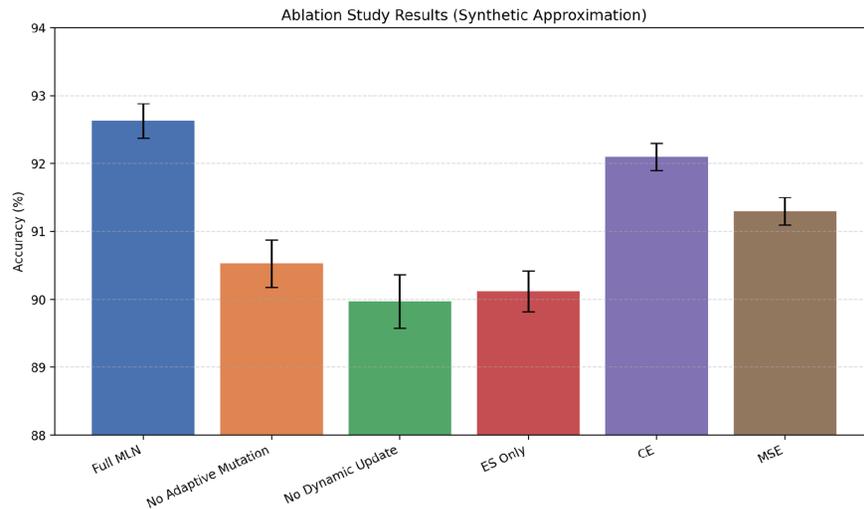


Figure 6: Ablation results comparing full MLN with reduced variants. Removing adaptive mutation, dynamic updates, or hybrid ES-GA lowers performance, while MLN outperforms standard CE and MSE losses.

4.4 Sensitivity to Hyperparameters

We test MLN's resilience under varying loop counts and population sizes. Fig. 7 shows some results from hyperparameter changes in meta-testing. The red line leads to Adaptive-MLN, the green line to MSE, and the blue line to CE. This figure presents the impact of hyperparameter variations on the performance of the Adaptive-MLN during the meta-testing phase. It demonstrates the MLN's strength across various configurations and its ability to self-adjust and perform well under different learning conditions.

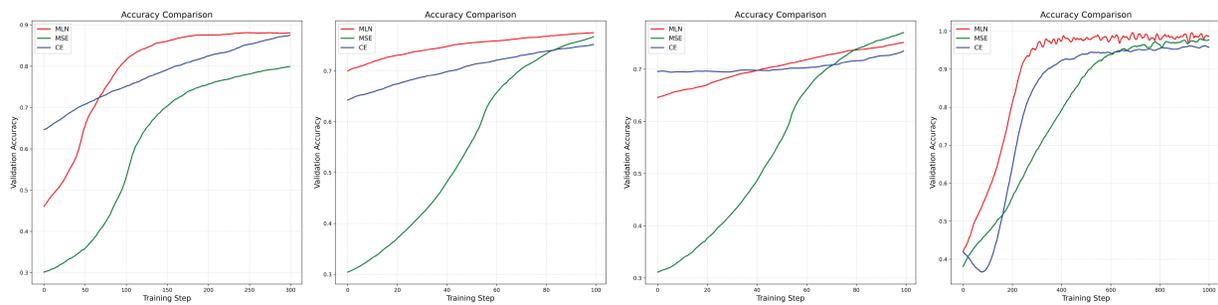


Figure 7: Impact of hyperparameter changes on accuracy in meta-testing.

The six-layer structure of the Adaptive-MLN provides a practical balance between representational power and computational cost. Through sensitivity analysis, we varied both the network depth (4, 6 and 8 layers) and width (32–1024 units). Our results show that models with more than six layers yield minimal performance gains and often introduce training instability, while shallower versions lack sufficient capacity to capture complex loss landscapes. In terms of overhead, the Adaptive-MLN adds a relatively small number of parameters compared to task networks such as VGG16 ($\approx 138\text{M}$ parameters). Because the evolutionary optimization operates on this compact Adaptive-MLN, the overall system remains scalable even when applied to larger datasets. As illustrated in Fig. 7, the MLN also maintains consistent performance across different population sizes and sampling loops, demonstrating strong scalability.

4.5 Discussions

The hybrid ES-GA optimizer increases meta-training time compared to gradient-based approaches; however, this cost is incurred only once during the MLN optimization stage. After training, the MLN functions as a conventional loss module, producing a scalar value through a single forward pass. As a result, the proposed framework introduces no additional computational overhead during the training or inference of downstream models. This clear separation between offline evolutionary optimization and lightweight deployment ensures the approach's practicality.

Although our experiments focus on classification tasks, Adaptive-MLN is not limited to categorical outputs. Since it produces a continuous scalar loss, it can be naturally extended to regression by operating directly on real-valued prediction-target pairs. Moreover, for mixed-task settings that involve both regression and classification objectives, the MLN can jointly model these signals by conditioning on shared representations. This flexibility positions Adaptive-MLN as a general-purpose loss-learning framework and suggests several promising avenues for future research.

5 Conclusions

In this study, we introduced the Adaptive-MLN, a novel framework for evolving loss functions in classification tasks. Unlike conventional loss functions such as Cross-Entropy and Mean Squared Error, which remain fixed during training, the Adaptive-MLN dynamically adjusts its structure and parameters in response to task complexity and data distribution. This adaptability allows for improved convergence and generalization across both synthetic and real-world tasks. To train the Adaptive-MLN without relying on gradients, we developed a hybrid optimization scheme that integrates Evolutionary Strategies (ES) with Genetic Algorithms (GA). This hybrid approach balances global search and local refinement, enabling robust optimization even in non-differentiable environments. We validated the proposed framework on both synthetic meta-learning tasks and real-world image classification using the CIFAR-10 and MNIST

datasets. Across all evaluations, Adaptive-MLN consistently outperformed standard loss functions in terms of accuracy, training stability, and convergence speed. These results demonstrate the potential of trainable, evolution-guided loss functions to enhance learning flexibility in AI systems.

Acknowledgement: The authors would like to thank Beijing Institute of Technology, China.

Funding Statement: This work was partly supported by the National Natural Science Foundation of China (NSFC) under Grant number: 82171965.

Author Contributions: Mirna Yunita: conceptualization, methodology, writing—original draft, review and editing; Xiabi Liu: writing—review and editing, supervision, validation; Zhaoyang Hai: conceptualization, methodology, writing—review and editing; Rachmat Muwardi: writing—review and editing, visualization. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors.

Ethics Approval: Not applicable.

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

References

1. Hospedales TM, Antoniou A, Micaelli P, Storkey AJ. Meta-learning in neural networks: a survey. *IEEE Trans Pattern Anal Mach Intell.* 2022;44(9):5149–69. doi:10.1109/tpami.2021.3079209.
2. Baik S, Choi J, Kim H, Cho D, Min J, Lee KM. Meta-learning with task-adaptive loss function for few-shot learning. In: *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*; 2021 Oct 10–17; Montreal, QC, Canada. p. 9465–74.
3. Shim JW. Enhancing cross entropy with a linearly adaptive loss function for optimized classification performance. *Sci Rep.* 2024;14(1):27405. doi:10.1038/s41598-024-78858-6.
4. Chen J, Yuan W, Chen S, Hu Z, Li P. Evo-MAML: meta-learning with evolving gradient. *Electronics.* 2023;12(18):3865. doi:10.3390/electronics12183865.
5. Jeong T, Kim H. OOD-MAML: meta-learning for few-shot out-of-distribution detection and classification. *Adv Neural Inf Process Syst.* 2020;33:3907–16.
6. Gonzalez S, Mikkilainen R. Improved training speed, accuracy, and data utilization through loss function optimization. In: *Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC)*; 2020 Jul 19–24; Glasgow, UK. doi:10.1109/cec48606.2020.9185777.
7. Wang H, Zhao H, Li B. Bridging multi-task learning and meta-learning: towards efficient training and effective adaptation. *Proc Mach Learn Res.* 2021;139:10991–1002.
8. Talbi EG. Machine learning into metaheuristics. *ACM Comput Surv.* 2021;54(6):1–32. doi:10.1145/3459664.
9. Li Z, Lin X, Zhang Q, Liu H. Evolution strategies for continuous optimization: a survey of the state-of-the-art. *Swarm Evol Comput.* 2020;56:100694. doi:10.1016/j.swevo.2020.100694.
10. Bäck T, Kononova AV, van Stein B, Wang H, Antonov K, Kalkreuth R, et al. Evolutionary algorithms for parameter optimization—thirty years later. *Evol Comput.* 2023;31(2):81–122. doi:10.1162/evco_a_00325.
11. Alhijawi B, Awajan A. Genetic algorithms: theory, genetic operators, solutions, and applications. *Evol Intell.* 2023;17:1245–56. doi:10.1007/s12065-023-00822-6.
12. Li Y, Li W, Li S, Zhao Y. A performance indicator-based evolutionary algorithm for expensive high-dimensional multi-/many-objective optimization. *Inf Sci.* 2024;678(1):121045. doi:10.1016/j.ins.2024.121045.
13. Tian Y, Su D, Lauria S, Liu X. Recent advances on loss functions in deep learning for computer vision. *Neurocomputing.* 2022;497:129–58. doi:10.1016/j.neucom.2022.04.127.
14. Wan J, Wang T, Li W, Wang Z. Enhanced robustness in machine learning: application of an adaptive robust loss function. *Appl Comput Eng.* 2024;95(1):151–61. doi:10.54254/2755-2721/95/2024CH0053.

15. Dereventsov A, Webster CG, Daws J. An adaptive stochastic gradient-free approach for high-dimensional blackbox optimization. In: *Proceedings of International Conference on Computational Intelligence*. Singapore: Springer; 2022. p. 333–48. doi:10.1007/978-981-16-3802-2_28.
16. Wen C, Yang X, Zhang K, Zhang J. Improved loss function for image classification. *Comput Intell Neurosci*. 2021;2021(1):1–8. doi:10.1155/2021/6660961.
17. Baik S, Choi M, Choi J, Kim H, Lee KM. Meta-learning with adaptive hyperparameters. *Neural Inf Process Syst*. 2020;33:20755–65.
18. Hai Z, Pan L, Liu X, Han M. L2T-DFM: learning to teach with dynamic fused metric. *Pattern Recognit*. 2024;159:111124. doi:10.1016/j.patcog.2024.111124.
19. Psaros AF, Kawaguchi K, Karniadakis GE. Meta-learning PINN loss functions. *J Comput Phys*. 2022;458:111121. doi:10.1016/j.jcp.2022.111121.
20. Zhang S, Xie L. Leader learning loss function in neural network classification. *Neurocomputing*. 2023;557:126735. doi:10.1016/j.neucom.2023.126735.
21. Raymond C, Chen Q, Xue B, Zhang M. Fast and efficient local-search for genetic programming based loss function learning. In: *Proceedings of the GECCO'23: Proceedings of the Genetic and Evolutionary Computation Conference; 2023 Jul 15–19; Lisbon, Portugal*. p. 1184–93.
22. Mustafa A, Mikhailiuk A, Iliescu DA, Babbar V, Mantiuk RK. Training a task-specific image reconstruction loss. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2022 Jun 3–8; Waikoloa, HI, USA*. p. 2319–28. doi:10.1109/wacv51458.2022.00010.
23. Gonzalez S, Miikkulainen R. Optimizing loss functions through multi-variate Taylor polynomial parameterization. In: *Proceedings of the GECCO'21: Proceedings of the Genetic and Evolutionary Computation Conference; 2021 Jul 10–14; New York, NY, USA*. p. 305–13.
24. Bechtle S, Molchanov A, Chebotar Y, Grefenstette E, Righetti L, Sukhatme G. Meta learning via learned loss. In: *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR); 2020 Jan 10–15; Milan, Italy*.
25. Raymond C, Chen Q, Xue B, Zhang M. Learning symbolic model-agnostic loss functions via meta-learning. *IEEE Trans Pattern Anal Mach Intell*. 2023;45(11):1–15. doi:10.1109/tpami.2023.3294394.
26. Gao B, Gouk H, Yang Y, Hospedales T. Loss function learning for domain generalization by implicit gradient. *Proc Mach Learn Res*. 2022;162:7002–16.
27. Hai Z, Liu X, Ren Y, Soomro NQ. Generating meta-learning tasks to evolve parametric loss for classification learning. *arXiv:2111.10583*. 2021.
28. Lange R, Schaul T, Chen Y, Zahavy T, Dalibard V, Lu C, et al. Discovering evolution strategies via meta-black-box optimization. In: *Proceedings of the GECCO'23 Companion: Proceedings of the Companion Conference on Genetic and Evolutionary Computation; 2023 Jul 15–19; New York, NY, USA*. doi:10.1145/3583133.3595822.
29. Song Y, Wu Y, Guo Y, Yan R, Suganthan PN, Zhang Y, et al. Reinforcement learning-assisted evolutionary algorithm: a survey and research opportunities. *Swarm Evol Comput*. 2024;86:101517. doi:10.1016/j.swevo.2024.101517.
30. Golshanrad P, Rahmani H, Karimian B, Karimkhani F, Weiss G. MEGA: predicting the best classifier combination using meta-learning and a genetic algorithm. *Intell Data Anal*. 2021;25(6):1547–63. doi:10.3233/ida-205494.
31. Lee S, Kim J, Kang H, Kang DY, Park J. Genetic algorithm based deep learning neural network structure and hyperparameter optimization. *Appl Sci*. 2021;11(2):744. doi:10.3390/app11020744.
32. Moskalenko V, Korobov A, Kudravcev A, Boiko M. Meta-learning with evolutionary strategy for resilience optimization of image recognition system. In: *Proceedings of the 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS); 2023 Dec 7–9; Dortmund, Germany*. p. 978–82. doi:10.1109/idaacs58523.2023.10348942.
33. Kramer O. Enhancing evolutionary algorithms through meta-evolution strategies. In: *Proceedings of the 2025 IEEE Conference on Artificial Intelligence (CAI); 2025 May 5–7; Santa Clara, CA, USA*. p. 1292–7. doi:10.1109/cai64502.2025.00225.

34. Zhou Y, Li S, Pedrycz W, Feng G. ACDB-EA: adaptive convergence-diversity balanced evolutionary algorithm for many-objective optimization. *Swarm Evol Comput.* 2022;75:101145. doi:10.1016/j.swevo.2022.101145.
35. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. *Proc Mach Learn Res.* 2017;70:1126–35.
36. Zheng H, Yang Z, Liu W, Liang J, Li Y. Improving deep neural networks using softplus units. In: *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN)*; 2015 Jul 12–17; Killarney, Ireland. p. 1–4. doi:10.1109/ijcnn.2015.7280459.
37. Shi Z, Xiang M, Hai Z, Liu X, Pei Y. GARA: a novel approach to improve genetic algorithms' accuracy and efficiency by utilizing relationships among genes. *arXiv:2404.18955.* 2024.
38. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *Proc Mach Learn Res.* 2010;9:249–56. doi:10.1007/978-1-4842-3790-8_3.
39. Lipowski A, Lipowska D. Roulette-wheel selection via stochastic acceptance. *Phys Stat Mech Its Appl.* 2012;391(6):2193–6. doi:10.1016/j.physa.2011.12.004.
40. Morgan B, Hougen D. Neural loss function evolution for large-scale image classifier convolutional neural networks. In: *Proceedings of the GECCO'24 Companion: Proceedings of the Genetic and Evolutionary Computation Conference Companion*; 2024 Jul 14–18; Melbourne, Australia. p. 603–6. doi:10.1145/3638530.3654380.