Distribution-Free Pretraining of Classification Losses via Evolutionary Dynamics

Xiang Meng* and Yan Pei†
*Graduate School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, 965-8580 Japan.
∗ d8242104@u-aizu.ac.jp
†Computer Science Division, University of Aizu, Aizuwakamatsu, 965-8580 Japan.
† corresponding author: peiyan@u-aizu.ac.jp

***Abstract*—We propose Evolutionary Dynamic Loss (EDL), a framework that learns a transferable classification loss in the probability space using unlimited synthetic prediction-label pairs, without accessing real samples during the main loss pretraining stage. EDL parameterizes the loss as a lightweight network and is trained with a semantics-free ranking-consistency objective that assigns larger penalties for more erroneous predictions. To robustly explore the space of loss functions, we optimize EDL via an evolutionary strategy and introduce chaotic mutation to improve exploration under noisy fitness evaluations. Experiments on CIFAR-10 with ResNet backbones show that EDL can serve as a drop-in replacement for cross-entropy and achieves competitive or improved accuracy, while ablation studies confirm that chaotic mutation yields faster convergence and better synthetic pretraining metrics than standard Gaussian mutation.**

***Index Terms*—neural networks, dynamic loss function, evolutionary computation, chaotic mutation, adaptive optimization.**

I. Introduction

Loss functions play a central role in supervised classification. They convert predicted probabilities into learning signals and thereby shaping optimization dynamics, training stability, and generalization. Despite their broad success, widely used objectives such as cross-entropy and its variants take the form of fixed analytic functions. When training conditions change, e.g., the hardness distribution, sampling bias, model capacity, label noise, or class imbalance, a fixed loss may no longer provide well-calibrated gradients or robust performance. This raises a fundamental question: can we learn a loss function that does not depend on any particular dataset yet remains scalable and transferable across downstream settings?

Most existing approaches to learned losses or meta-learned objectives still rely on real task data and training trajectories for outer-loop supervision. Consequently, they face three limitations: supervision is bounded by dataset size and privacy constraints; dataset-specific biases can be absorbed into the learned objective and harm cross-task transfer; and learning a loss is closer to searching for a function shape than fitting conventional parameters, making optimization noise-sensitive and prone to undesirable shapes that destabilize downstream training.

In this work, we propose a loss-learning framework centered on two principles: distribution-free synthetic supervision in the probability space and evolutionary search for robust loss shapes. Our key observation is that, for classification, a loss depends on the relationship between the predictive distribution and the label rather than on the raw input modality. Therefore, we learn a parametric Evolutionary Dynamic Loss (EDL) directly in the probability space by generating unlimited synthetic pairs (p, y) without accessing any real samples, thereby enabling scalable and data-independent loss learning.

Fig. 1 summarizes the overall pipeline. To obtain meaningful supervision without semantics, we impose a simple monotonicity principle: harder predictions should incur larger penalties. Concretely, we sample synthetic pairs, define a deterministic hardness proxy (e.g., based on the true-class probability), and train EDL using a smooth pairwise ranking-consistency objective that encourages it to preserve the same ordering. Moreover, we adopt a controllable mixture sampling scheme on the probability simplex to emphasize critical regimes that dominate training dynamics, including extreme-confidence cases and boundary-confusing predictions, thereby helping to constrain the loss shape across the full confidence spectrum.

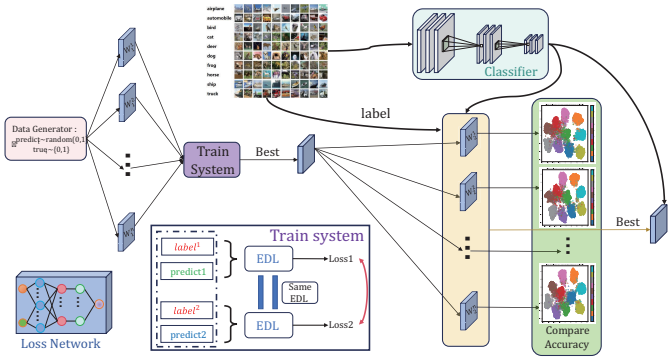

Fig. 1. Overview of the proposed EDL pipeline. We learn a transferable loss prior from unlimited synthetic prediction and label pairs (p, y) in the probability space using a ranking-consistency signal and evolutionary search with chaotic mutation. The learned loss can then be used as a drop-in objective for standard downstream classifier training, optionally with a lightweight validation-based selection or calibration step on the target dataset.

Optimizing such a shape-search objective can be brittle when using purely gradient-based methods under noisy pairwise estimates. We therefore employ an evolutionary strategy (ES) to explore the EDL parameter space globally. In addi-

tion, we introduce a logistic-map-driven chaotic mutation to modulate mutation amplitudes in a bounded yet non-periodic manner, thereby improving population diversity and reducing premature convergence in the early stages of the search. The outcome of this stage is a transferable loss prior learned without real data. For practical use, the learned loss can be integrated into standard classifier training as a drop-in objective, optionally followed by a lightweight validation-based selection step to better match the target dataset.

There are three contributions of this study. We propose a distribution-free loss-learning formulation that learns a parametric loss directly from synthetic prediction and label pairs in the probability space via a ranking-consistency objective. We further develop an evolutionary training strategy for loss-shape search, incorporating logistic-map-driven chaotic mutation to enhance global exploration and robustness under noisy pairwise supervision. Finally, we demonstrate that the resulting loss prior can be seamlessly integrated into standard training pipelines, yielding stable and competitive generalization across downstream settings.

The remainder of this paper is organized as follows. Section II reviews the relevant literature on loss learning and evolutionary optimization. Section III presents the proposed method in detail. Section IV reports the experimental results and analysis. Finally, Section V concludes the paper and discusses future work.

II. Related Work

In supervised classification, performance can be improved by optimizing multiple components of the learning pipeline, including initialization [1], learning-rate schedules [2], activation functions [3], data augmentation [4], and network architectures. Among these factors, the loss function is particularly influential: it defines the training signal and error geometry, thereby shaping optimization dynamics, training stability, and generalization. Accordingly, recent work has increasingly focused on learning and adapting loss functions to better handle task diversity and distributional shifts [5].

A. Static Loss Functions

Static loss functions are specified prior to training and remain fixed throughout the optimization process. Classical objectives such as cross-entropy and hinge loss are widely adopted due to their simplicity and interpretability [6], [7]. A substantial body of research refines static losses by reshaping penalty profiles or introducing auxiliary terms to improve robustness and enhance discriminative learning. However, because their functional forms remain fixed, static losses can be suboptimal when training conditions vary across stages or when the data distribution shifts, as they cannot dynamically reallocate learning pressure across different error regimes.

B. Dynamic and Learnable Loss Functions

To address the limited adaptability of static objectives, dynamic and learnable loss functions aim to adapt error evaluation during training based on the data distribution or the model state. Representative approaches [8] include learning adaptive coefficient structures that modulate the relationship between predictions and labels, meta-models [9] that stochastically select and combine handcrafted losses, and meta-networks [10] that generate transformation parameters for updating a loss network. Hai [11] further propose an LSTM-based meta-optimizer to update a loss network online, enabling stage-wise adaptation during training. In addition, margin-based Softmax losses have been studied under unified formulations, where random search and policy-gradient methods [12] are employed to tune key hyperparameters for improved discriminative learning. Overall, these methods highlight the promise of learning the optimization objective itself; however, they typically rely on real datasets and task-specific training trajectories for outer-loop supervision.

C. Evolutionary and Gradient-Free Loss Optimization

Complementary to trajectory-driven, gradient-based meta-learning, another line of work explores loss shapes using gradient-free optimization. In particular, evolutionary computation can parameterize the loss and optimize it via mutation and selection, enabling global exploration without differentiability requirements. Representative examples include Genetic Loss-function Optimization (GLO) [5] and its continuous parameterizations such as TaylorGLO [13], as well as evolutionary loss-function frameworks such as ELF [14]. These perspectives motivate our use of evolutionary search to learn a transferable loss prior from probability-space supervision. However, existing evolutionary approaches still rely on real data or task-specific evaluation, motivating our distribution-free formulation.

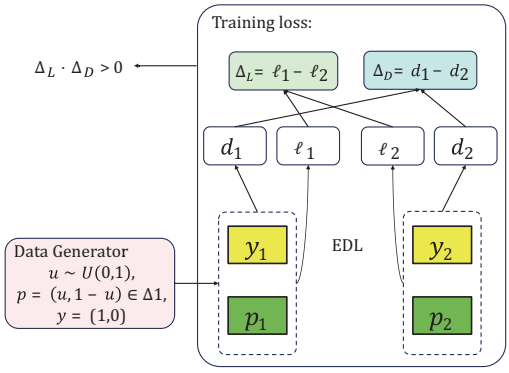

Fig. 2. Synthetic supervision for pretraining EDL via ranking consistency in probability space. Two synthetic prediction–label pairs $(\mathbf{p}_1, \mathbf{y}_1)$ and $(\mathbf{p}_2, \mathbf{y}_2)$ are sampled on the probability simplex. EDL outputs scalar losses ℓ_1 and ℓ_2, and we form $\Delta_L = \ell_1 - \ell_2$. A proxy hardness difference Δ_D is computed from the same pairs, and pretraining encourages the ordering agreement $\Delta_L \Delta_D > 0$.

III. Distribution-Free Pretraining of Classification Losses Using Evolutionary Dynamics

We propose to learn a parametric loss function directly in the probability space. Specifically, we model the loss as an

Evolutionary Dynamic Loss that takes a classifier's predictive distribution and the ground-truth label as input and outputs a non-negative scalar value. To make the learned loss scalable and dataset-agnostic, we construct synthetic supervision by sampling prediction–label pairs on the probability simplex and optimize EDL parameters via an evolutionary strategy under a ranking-consistency objective. During evolution, we adopt chaotic mutation driven by a logistic map to diversify the population and improve global exploration. After training, EDL can be integrated into standard training pipelines as the optimization objective for downstream classifiers. This design enables data-independent and scalable loss learning without relying on any real samples.

Fig. 2 visualizes the core supervision signal used to train EDL without real data. Note that the condition $\Delta L \cdot \Delta D > 0$ is not optimized as a discrete constraint; in practice, we minimize a smooth pairwise ranking surrogate (e.g., softplus applied to $-\Delta L \cdot \Delta D$) to obtain stable gradients. The proxy hardness is computed deterministically from $(\mathbf{p}, \mathbf{y})$ (e.g., using the true-class probability or a distance between $\mathbf{p}$ and $\mathbf{y}$), and serves only to define a relative difficulty ordering between two synthetic samples. This construction provides scalable supervision for shaping a monotone loss behavior across diverse confidence regimes, while remaining agnostic to any specific dataset.

A. Evolutionary Dynamic Loss Network

Let z denote the logits produced by a classifier, and $p = \mathrm{softmax}(z) \in \Delta^{C-1}$ be the corresponding predictive distribution over C classes. Given a label $y \in \{1, \ldots, C\}$ with one-hot encoding $e_y \in \{0,1\}^C$, EDL defines a learned loss

$$L_\phi(p, y) \;=\; \mathrm{EDL}_\phi\big([p, e_y]\big), \tag{1}$$

where $[p, e_y]$ denotes concatenation and ϕ denotes the parameters of EDL. We enforce non-negativity by applying a softplus output layer, ensuring $L_\phi(p, y) \geq 0$.

B. Synthetic Data Construction in the Probability Space

To train EDL without using real samples, we generate synthetic training pairs (p, y). We sample labels uniformly as $y \sim \mathrm{Unif}(1, \ldots, C)$ and draw p from a controllable mixture on Δ^{C-1} to cover diverse confidence regimes. In particular, we emphasize two critical regions that frequently dominate training dynamics: (i) extreme-confidence predictions with probabilities close to 0 or 1, and (ii) boundary-confusing predictions where the true-class probability is close to that of a competing class. The resulting synthetic probabilities exhibit concentrated mass near the extremes while still maintaining broad coverage over intermediate-confidence ranges (see Fig. 3),providing scalable supervision for shaping the loss function behavior across the full spectrum of prediction patterns.

C. Ranking-Consistency Objective and Fitness

Our supervision is based on a monotonicity principle: more incorrect predictions should incur larger loss values. For a synthetic sample (p, y), we define an error-severity (hardness) score

$$D(p, y) \;=\; 1 - p_y, \tag{2}$$

where p_y is the predicted probability assigned to the true class. We then sample pairs (p_i, y_i) and (p_j, y_j) and compare the ordering induced by D and by EDL outputs. Let

$$\begin{aligned} \Delta D_{ij} &= D(p_i, y_i) - D(p_j, y_j), \qquad s_{ij} = \mathrm{sign}(\Delta D_{ij}), \\ \Delta L_{ij} &= L_\phi(p_i, y_i) - L_\phi(p_j, y_j). \end{aligned} \tag{3}$$

The desired ranking consistency condition is $s \cdot \Delta L > 0$. To obtain a stable and continuous training signal, we define the fitness as a smooth pairwise ranking loss:

$$\mathcal{F}(\phi) = \mathbb{E}_{(i,j)}\Big[\mathrm{softplus}\big(-s \cdot \Delta L\big)\Big], \tag{4}$$

where the expectation is over randomly sampled pairs. Minimizing $\mathcal{F}(\phi)$ encourages EDL to assign larger loss to examples with larger error severity. For monitoring, we also report a trend-consistency accuracy:

$$\mathrm{Acc}(\phi) = \mathbb{P}_{(i,j)}\big[s \cdot \Delta L > 0\big]. \tag{5}$$

D. Evolutionary Optimization with Chaotic Mutation

We optimize EDL parameters ϕ using a population-based evolutionary strategy over candidates $\mathcal{P} = \{\phi^{(k)}\}_{k=1}^{K}$. Since $\mathcal{F}(\phi)$ in Eq. 4 is defined as an expectation over randomly sampled pairs, we evaluate each candidate by a Monte Carlo estimate computed on multiple synthetic batches:

$$\begin{aligned} \widehat{\mathcal{F}}\Big(\phi^{(k)}\Big) &= \frac{1}{B}\sum_{b=1}^{B}\frac{1}{|\mathcal{S}_b|}\sum_{(i,j)\in\mathcal{S}_b}\mathrm{softplus}\Big(-s_{ij}\,\Delta L_{ij}^{(k)}\Big), \\ \Delta L_{ij}^{(k)} &= L_{\phi^{(k)}}(p_i, y_i) - L_{\phi^{(k)}}(p_j, y_j). \end{aligned} \tag{6}$$

At each generation, we keep the top K_e candidates with the lowest $\widehat{\mathcal{F}}$ as elites and generate the remaining candidates by chaotic mutation.

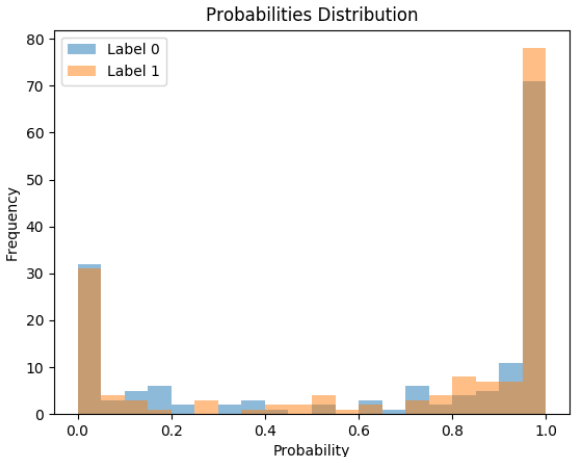

Fig. 3. Histogram of synthetic probabilities used for EDL training. We group samples by label and plot the distribution of predicted probabilities. The mixture sampling intentionally covers both near-extreme regions (close to 0 or 1) and intermediate-confidence regions, providing diverse supervision for learning a ranking-consistent loss shape in probability space.

a) Logistic Map Chaos: We use thelogistic map to generate a chaotic coefficient:

$$x_{t+1} = r\, x_t(1 - x_t), \qquad r = 4.0,\ x_t \in (0, 1). \tag{7}$$

b) Chaotic Mutation Operator: Let θ_k denote the k-th learnable parameter in ϕ. Given a mutation scale σ, we mutate each parameter by

$$\theta'_k = \theta_k + \sigma\, d_k\, x_{t+1}\, \epsilon_k, \qquad d_k \in \{-1, +1\},\ \epsilon_k \sim \mathcal{N}(0, 1), \tag{8}$$

where d_k controls mutation direction and the chaotic coefficient x_{t+1} adaptively modulates the step size in a bounded yet non-linear manner. Compared with a fixed-scale Gaussian mutation, chaotic mutation yields richer perturbation patterns and improves early-stage exploration of diverse loss shapes.

We adopt an adaptive mutation scale (switching between σ_{high} and σ_{low} based on the best Acc) to balance exploration and refinement. The overall procedure is summarized in Algorithm 1.

E. Algorithm Explanation

Algorithm 1 repeats evaluation–selection–chaotic mutation: it evaluates candidates by $\widehat{\mathcal{F}}$, keeps the best K_e elites, and refills the population using Logistic-map-driven chaotic mutation with an adaptive σ. The final output is the EDL with the lowest estimated fitness.

Algorithm 1 Training for EDL

Require: Population size K, elite size K_e, generations G, batches per evaluation B, mutation scales $(\sigma_{\text{high}}, \sigma_{\text{low}})$, threshold τ, max attempts A
Ensure: Best EDL parameters $\phi^\star$

Initialize population $\mathcal{P} = \{\phi^{(k)}\}_{k=1}^{K}$; sample $x_0 \sim \mathrm{Unif}(0, 1)$; set $t \leftarrow 0$
for $g = 1$ to G **do**
  Evaluate $\widehat{\mathcal{F}}(\phi^{(k)})$ for all k using B synthetic batches
  Select elites $\mathcal{E} \subset \mathcal{P}$ as the K_e candidates with lowest $\widehat{\mathcal{F}}$; $\phi_{\text{best}} = \arg\min_{\phi \in \mathcal{E}} \widehat{\mathcal{F}}(\phi)$
  $\sigma \leftarrow \sigma_{\text{high}}$ if $\mathrm{Acc}(\phi_{\text{best}}) < \tau$ else σ_{low}
  $\mathcal{P}' \leftarrow \mathcal{E}$
  while $|\mathcal{P}'| < K$ **do**
    Sample parent $\phi \sim \mathrm{Unif}(\mathcal{E})$
    for $a = 1$ to A **do**
      $x_{t+1} \leftarrow 4x_t(1 - x_t)$; $t \leftarrow t + 1$
      Sample $d \in \{-1, +1\}^{|\phi|}$ and $\epsilon \sim \mathcal{N}(0, I)$
      $\phi' \leftarrow \phi + \sigma\, x_{t+1}\, (d \odot \epsilon)$ ▷ Chaotic mutation
      if $\widehat{\mathcal{F}}(\phi') \le \widehat{\mathcal{F}}(\phi)$ **then break**
      end if
    end for
    Add ϕ' to $\mathcal{P}'$
  end while
  $\mathcal{P} \leftarrow \mathcal{P}'$
end for
return $\phi^\star = \arg\min_{\phi \in \mathcal{P}} \widehat{\mathcal{F}}(\phi)$

F. Theoretical Analysis

a) Contrastive Supervision in the Probability Space: EDL is trained in the probability space using synthetic pairs (p, y), which is sufficient because a classification loss depends only on the relation between predictive probabilities and labels. We reuse the hardness score $D(p, y)$ in Eq. 2, the pairwise ordering variables $(\Delta D_{ij}, s_{ij}, \Delta L_{ij})$ in Eq. 3, and the ranking-consistency objective $\mathcal{F}(\phi)$ in Eq. 4. Minimizing Eq. 4 enforces the monotonicity principle that harder predictions (larger D) should incur larger loss values (larger L_ϕ). Moreover, if there exists a strictly increasing $g(\cdot)$ such that $L^\star(p, y) = g(D(p, y))$, then $\mathrm{sign}(\Delta L^\star_{ij}) = \mathrm{sign}(\Delta D_{ij})$ for any (i, j); hence the objective in Eq. 4 directly promotes a monotone loss shape with a positive ordering margin.

b) Extreme Coverage via Mixture Sampling: Uniform sampling on the simplex may under-represent extreme-confidence regimes ($p_y \approx 0$ or 1), which are often the most sensitive regions for gradient magnitudes and training dynamics. Our synthetic construction therefore draws (p, y) from a controllable mixture distribution (see Fig. 3), which increases the fraction of pairwise constraints involving near-extreme samples. This reduces extrapolation uncertainty at the boundaries and improves ordering consistency across the full confidence spectrum induced by $D(p, y)$.

c) Stability under Stochastic Fitness Evaluation: Because $\mathcal{F}(\phi)$ is an expectation over randomly sampled pairs, the ES fitness estimate $\widehat{\mathcal{F}}(\phi)$ in Eq. 6 is noisy. Averaging over B independent batches reduces estimator variance approximately inversely with B:

$$\widehat{\mathcal{F}}(\phi) = \frac{1}{B}\sum_{b=1}^{B} \widehat{\mathcal{F}}_b(\phi), \qquad \mathrm{Var}\Big[\widehat{\mathcal{F}}(\phi)\Big] \approx \frac{1}{B}\mathrm{Var}\Big[\widehat{\mathcal{F}}_1(\phi)\Big]. \tag{9}$$

Elitism further protects the current best candidate, and the optional non-degradation acceptance rule

$$\widehat{\mathcal{F}}(\phi') \le \widehat{\mathcal{F}}(\phi) \tag{10}$$

reduces random drift caused by unlucky perturbations under noisy evaluation.

d) Why Chaotic Mutation Helps (Bounded, Intermittent Steps): Standard mutation uses fixed-scale Gaussian perturbations, while our chaotic mutation in Eq. 8 introduces a bounded, time-varying scale factor generated by the Logistic map in Eq. 7. Equivalently, each mutation step can be viewed as

$$\Delta\theta = \sigma\, x\, \epsilon, \qquad x \in (0, 1),\ \epsilon \sim \mathcal{N}(0, 1), \tag{11}$$

where x follows the chaotic dynamics induced by Eq. 7. In the canonical chaotic regime, x is ergodic on $(0, 1)$ with an invariant distribution that places substantial mass near both 0 and 1, yielding an intermittent step pattern: most mutations are strongly down-scaled (small steps) while a non-negligible fraction are near unscaled (large steps). This creates a natural "many-small, few-large" exploration behavior: small steps stabilize selection by reducing sensitivity to noisy fitness estimates, whereas rare large steps help escape poor local loss

shapes. Combined with elitist selection and the acceptance rule in Eq. 10, chaotic mutation provides a simple mechanism to balance exploitation and exploration during loss-shape search.

IV. Experimental Evaluations

A. Experimental Setup

a) Datasets: We evaluate EDL on CIFAR-10 [15], which contains 50000 training images and 10000 test images of size 32×32.

b) Implementation Details: Our method contains two learning components: (1) synthetic pretraining of the parametric loss in probability space, and (2) downstream classifier training using the learned loss. Table I summarizes the shared architectures and hyper-parameters. For the classifier, we use a standard ResNet backbone. For EDL, we employ a lightweight MLP that maps the concatenated input $[p; e_y]$ to a non-negative scalar via a Softplus head, following the same loss-network architecture as [14]. In the evolutionary setting, we maintain a population of candidate losses, keep elites, and generate offspring by mutation; chaotic mutation uses a Logistic map to modulate the mutation amplitude. All experiments are conducted on a single NVIDIA GPU (RTX 3060).

c) Evaluation Metric: Downstream performance is evaluated by Top-1 accuracy (%) on the CIFAR-10 test set. For the synthetic pretraining process, we report the best (lowest) fitness $\widehat{\mathcal{F}}$ and the trend-consistency accuracy Acc.

TABLE I
IMPLEMENTATION DETAILS (EDL).
STAGE 1 SYNTHETIC PRETRAINING IS THE FOCUS.

Downstream (CIFAR-10)	
Backbone / epochs	ResNet / 200
Optimiser / batch	SGD (momentum 0.9, wd 5×10^{-4}) / 128
LR schedule	init 10^{-2}, step $\times 0.1$ at $\{120, 160\}$
Loss	CE or EDL
Stage 1: Synthetic pretraining (probability space)	
Classes / input	$C = 10$; input: predicted prob. vector + one-hot label
Loss network	MLP (2C-10-20-20-1) + Softplus
Ranking supervision	Pairwise ranking on synthetic samples (Softplus ranking loss)
Synthetic budget	$A = 8192$ samples, $B = 4096$ pairs per generation
Optimisation (ES)	
Population / elites	$K = 6$, $K_e = 2$
Generations	80
Mutation (Normal)	Gaussian perturbation
Mutation (Chaotic)	Logistic-modulated perturbation (chaos factor $x_0 \sim U(0, 1)$)
Noise schedule	threshold $\tau = 0.95$; $\sigma_{\text{high}} = 0.20$, $\sigma_{\text{low}} = 0.01$

B. Results and Analyses

a) Synthetic Pretraining Behaviour: We first examine whether synthetic supervision can shape a meaningful loss prior. Across runs, EDL trained with evolutionary strategy exhibits steadily decreasing fitness and increasing trend-consistency accuracy, indicating that the learned loss preserves the desired monotonic ordering (harder predictions incur larger loss). Compared with Gaussian mutation, chaotic mutation improves population diversity and typically finds better loss candidates with more stable convergence.

TABLE II
RESULTS ON CIFAR-10 [15] FOR THE CLASSIFICATION TASK. ALL EXPERIMENTS ARE CONDUCTED UNDER THE SAME SETTINGS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	ResNet8	ResNet20	ResNet32
CE	87.6	91.3	92.5
Smooth [16]	87.9	91.5	92.6
L-M Softmax [17]	88.7	92.0	93.0
L2T-DLF [8]	89.2	92.4	93.1
GLO [5]	87.7	91.3	-
TaylorGLO [13]	-	91.6	-
ARLF [18]	89.5	91.5	92.2
SLF [9]	89.8	93.0	93.9
ALA [19]	-	-	93.2
L2T-DLN [11]	90.7	93.4	93.8
ELF [14]	90.4	92.9	93.0
EDL-GD	90.6 $\pm$ 0.18	91.9 $\pm$ 0.08	92.3 $\pm$ 0.24
EDL-ES-Chaotic	**91.1** $\pm$ **0.14**	**93.4** $\pm$ **0.17**	**93.9** $\pm$ **0.12**

b) Downstream CIFAR-10 Classification: We plug the pretrained EDL into a standard ResNet training pipeline on CIFAR-10. Table II summarizes the test accuracy. EDL-ES-Chaos achieves the best accuracy among learned-loss variants and outperforms over EDL-ES-Normal, demonstrating the effectiveness of chaotic exploration during loss pretraining. In contrast, EDL-GD, while achieving high synthetic ranking consistency, exhibits weaker transfer performance on CIFAR-10, suggesting that directly optimizing EDL by gradient descent on synthetic pairs may lead to less robust loss shapes for real-data optimization.

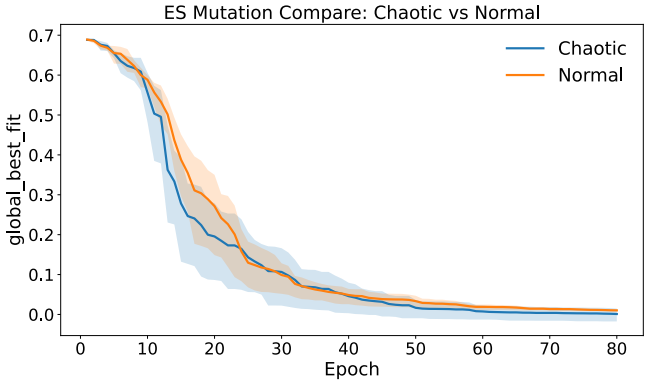

Fig. 4. Ablation on the mutation operator in ES-based EDL pretraining. We plot global_best_fit (best-so-far synthetic ranking fitness; lower is better) over epochs, shown as mean $\pm$ std across seeds. Compared with normal Gaussian mutation, chaotic mutation reduces the fitness faster in the early-to-mid stage and achieves a slightly better final global-best fitness under the same ES budget, suggesting improved exploration and more robust loss-shape search.

C. Chaotic vs. Normal Mutation Ablation Experiments

To isolate the effect of the mutation operator in synthetic EDL pretraining, we compare chaotic mutation (Eq. 7-Eq. 8) with standard Gaussian mutation under identical evolutionary settings. We keep the ES population size, elite selection, evaluation budget (generations and Monte Carlo batches), acceptance rule, and the adaptive mutation-scale schedule

fixed; the only difference is whether the mutation amplitude is modulated by a chaotic coefficient (Chaotic) or uses the standard Gaussian perturbation with the same scale (Normal).

Fig. 4 shows the trajectory of *global_best_fit* during synthetic pretraining, where *global_best_fit* denotes the best-so-far ranking-consistency fitness $\widehat{\mathcal{F}}$ found up to each epoch (lower is better); curves are reported as mean $\pm$ std across seeds. Both variants steadily reduce $\widehat{\mathcal{F}}$, confirming that probability-space supervision can shape a meaningful loss prior. Notably, chaotic mutation attains lower fitness earlier in the search (early-to-mid epochs), while both methods converge to similar performance. This early advantage translates into a lower average fitness over epochs and a better best-so-far solution overall.

Table III quantitatively corroborates this observation. Compared with normal mutation, chaotic mutation achieves a significantly lower final global-best fitness (0.01994 vs. 0.02810) and a much lower mean fitness across epochs, while maintaining saturated ranking accuracy (max best_acc $= 100\%$) with improved stability (lower std. best_acc) and slightly faster convergence to the best solution.

TABLE III
NORMAL VS. CHAOTIC MUTATION IN ES-BASED EDL SYNTHETIC PRETRAINING (FITNESS ↓, BEST_ACC ↑).

Mutation	Final best ↓	Mean fit ↓	Max acc ↑	Mean acc ↑	Std acc ↓	Epoch@ best ↓
Normal	0.02810	0.25742	100.00	99.18	1.70	80
Chaotic	**0.01994**	**0.15528**	100.00	**99.60**	**1.44**	**78**
Gain	**-29.05**%	**-39.68**%	+0.00%	+0.43	**-15.32**%	−2

V. CONCLUSION AND FUTURE WORK

We have presented Evolutionary Dynamic Loss, a distribution-free loss pretraining framework that learns a transferable loss prior from unlimited synthetic prediction–label pairs (p, y) in probability space, without accessing real samples during the main pretraining stage. The loss is identified via a semantics-free ranking-consistency objective that enforces a monotone penalty ordering with respect to prediction hardness, which is reflected in steadily improved synthetic ranking fitness and consistency.

To make the shape search robust under stochastic fitness evaluation, we optimize EDL with an evolutionary strategy and a Logistic-map-driven chaotic mutation that accelerates early-to-mid-stage convergence and improves aggregated pretraining metrics under the same ES budget; the resulting loss can then be integrated into standard CIFAR-10 training pipelines as a drop-in replacement and achieves competitive or improved Top-1 accuracy compared with strong learned-loss baselines. These results suggest that learning loss functions in probability space provides a promising direction for dataset-independent optimization design.

Future work will extend the proposal beyond classification to regression and generative modeling, where loss shaping in probability or distribution space may further improve robustness and flexibility across diverse learning paradigms. In addition, we plan to investigate theoretical foundations from a statistical learning and optimization perspective, including generalization properties, identifiability of learned loss functions, and convergence behavior under stochastic evolutionary search.

REFERENCES

[1] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. Schoenholz, and J. Pennington, "Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5393–5402.

[2] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[3] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[4] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[5] S. Gonzalez and R. Miikkulainen, "Improved training speed, accuracy, and data utilization through loss function optimization," in *2020 IEEE congress on evolutionary computation (CEC)*. IEEE, 2020, pp. 1–8.

[6] C. Gentile and M. K. K. Warmuth, "Linear hinge loss and average margin," in *Advances in Neural Information Processing Systems*, M. Kearns, S. Solla, and D. Cohn, Eds., vol. 11. MIT Press, 1998, pp. 225–231.

[7] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *The Annals of Statistics*, vol. 32, no. 1, pp. 56–85, 2004.

[8] L. Wu, F. Tian, Y. Xia, Y. Fan, T. Qin, L. Jian-Huang, and T.-Y. Liu, "Learning to teach with dynamic loss functions," *Advances in neural information processing systems*, vol. 31, pp. 6467–6478, 2018.

[9] Q. Liu and J. Lai, "Stochastic loss function," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4884–4891.

[10] S. Baik, J. Choi, H. Kim, D. Cho, J. Min, and K. M. Lee, "Meta-learning with task-adaptive loss function for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9465–9474.

[11] Z. Hai, L. Pan, X. Liu, Z. Liu, and M. Yunita, "L2t-dln: Learning to teach with dynamic loss network," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023, pp. 43 084–43 096.

[12] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[13] S. Gonzalez and R. Miikkulainen, "Improved training speed, accuracy, and data utilization through loss function optimization," in *2020 IEEE congress on evolutionary computation (CEC)*. IEEE, 2020, pp. 1–8.

[14] X. Meng, Z. Hai, X. Liu, and Y. Pei, "Optimization design of adaptive loss function using evolutionary neural networks," in *International Conference on Neural Information Processing*. Springer, 2025, pp. 321–335.

[15] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[16] T. Nguyen and S. Sanner, "Algorithms for direct 0–1 loss optimization in binary classification," in *International conference on machine learning*. PMLR, 2013, pp. 1085–1093.

[17] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," *arXiv preprint arXiv:1612.02295*, 2016.

[18] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4331–4339.

[19] C. Huang, S. Zhai, W. Talbott, M. B. Martin, S.-Y. Sun, C. Guestrin, and J. Susskind, "Addressing the loss-metric mismatch with adaptive loss alignment," in *International conference on machine learning*. PMLR, 2019, pp. 2891–2900.