Research Article

Alexandr Udeneev, Petr Babkin, and Oleg Bakhteev

Surrogate assisted diversity estimation in neural ensemble search

https://doi.org/10.1515/sample-YYYY-XXXX
Received Month DD, YYYY; revised Month DD, YYYY; accepted Month DD, YYYY

Abstract: Since most Neural Architecture Search (NAS) methods are computationally expensive, extending them to neural ensemble search (NES), which involves the joint optimization of both individual architectures and their ensemble configurations, can lead to a combinatorial increase in the search space, which often leads to prohibitively high computational costs. To address this, we introduce a dual-surrogate ensemble construction method: candidate architectures are represented as graphs, and two surrogate models are trained separately to predict accuracy and ensemble diversity. Their combined estimates guide an NES framework that efficiently identifies architectures that are both individually strong and collectively diverse. Our final ensemble achieves near-state-of-the-art accuracy on FashionMNIST, CIFAR-10, and CIFAR-100, surpassing standard baselines such as Deep Ensembles and random architecture search, while matching the performance of current top-performing methods.

Keywords: neural ensemble search, ensemble diversity, surrogate function, triplet loss.

1 Introduction

Neural network ensembles often demonstrate better accuracy, improved robustness, and more reliable uncertainty estimation compared to single models, especially in classification and regression tasks [1, 2]. This fact gives rise to the problem of constructing an efficient ensemble of models, often referred to as Neural Ensemble Search (NES) [3, 4]. NES, in turn, relies on Neural Architecture Search (NAS) methods, which are extensively studied and applied to search for individual neural network architectures, such as evolutionary algorithms [5, 6], reinforcement learning [7, 8], and Bayesian optimization [9, 10]. Selecting an optimal architecture for even a single model is a challenging task, particularly when considering data-specific constraints and computational limitations [11, 12].

One of the pioneer works for NES is DeepEns [13], which trains the same neural network architecture multiple times with different random initializations and combines their predictions. While this method is simple to implement and tune, its effectiveness is limited because there is no guarantee that the resulting ensemble will be sufficiently diverse. More sophisticated adaptations of NAS techniques are presented in recent works [4, 14, 15], which are designed to efficiently combine multiple networks into an ensemble. However, these methods often lack explicit, structured control over ensemble diversity: diversity either emerges implicitly through optimization objectives [4] or is guided by heuristic similarity measures [15], without a dedicated mechanism to balance it against individual model accuracy.

Our research also adapts ideas from NAS for NES, specifically utilizing surrogate functions to bypass the expensive evaluation of numerous candidate architectures [16, 17]. Modern NAS methods extensively employ surrogate models to estimate architecture performance without requiring full training, which significantly reduces computational overhead [16, 18]. For instance, [17] demonstrates the effectiveness of surrogate-assisted evolutionary algorithms in multi-objective search spaces. In the context of specialized tasks, [19] utilizes a Surrogate-assisted Multiobjective Evolutionary-based Algorithm (SaMEA) for medical image segmentation, proving the robustness of surrogates in high-dimensional domains. Furthermore, recent developments in similarity-based surrogates [20] provide a foundation for using such models to estimate not only accuracy but also structural relationships between architectures.

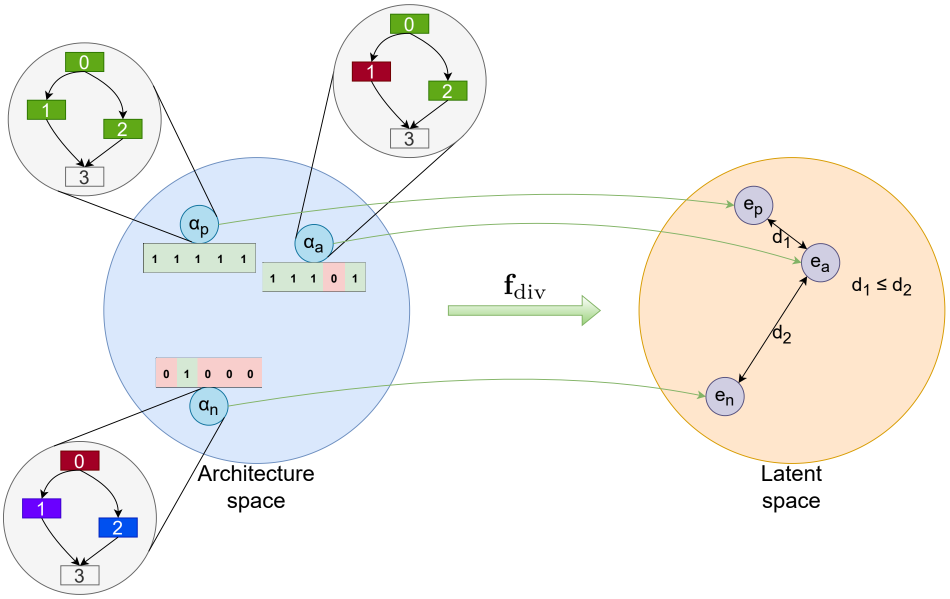

Fig. 1: Visualization of the surrogate diversity function. Architecture space showing different DARTS-like architectures represented as graphs, the vectors below them are the predictions of these models on fixed dataset. A surrogate diversity function transforms this space into latent space. In latent space where similar architectures (e.g. $\mathbf{e}_a$ and $\mathbf{e}_p$) are mapped close together, while dissimilar architectures (e.g., $\mathbf{e}_a$ and $\mathbf{e}_n$) are mapped farther apart, satisfying $d_1 \leq d_2$. Geometric organization in a latent space makes it possible to effectively evaluate diversity without the need for pre-training models with these architectures.

To address the challenges of NES, we propose a dual-surrogate framework that separates the search into two predictive objectives: accuracy estimation for individual architectures and diversity modeling via prediction of architecture embeddings in a latent space[20], enabling diversity assessment through embedding-space geometry. Inspired by surrogate-assisted NAS [12, 19], we encode candidate architectures as graphs and train lightweight surrogate models to capture these objectives. As illustrated in Fig. 1, the surrogate diversity function maps architectures from the discrete search space into a continuous latent embedding. In this space, architectures with correlated predictions are embedded close to each other, while dissimilar ones are separated by larger distances. This geometric organization enables efficient, gradient-friendly, and explicit estimation of ensemble diversity without requiring costly model training. To implement this, we employ Graph Attention Networks (GATs) [21] as the backbone for surrogate modeling [16], utilizing a Triplet Loss objective [22] to structure the latent space according to architectural diversity.

Main Contributions:

1. We introduce a novel surrogate-based methodology for training diversity predictors that map discrete neural architectures to a structured latent space.
2. We present the first surrogate-assisted framework for Neural Ensemble Search that jointly optimizes predictive performance and architectural diversity via dual-objective guidance.

3. We rigorously validate that our method matches or exceeds the performance of both standard baselines (Deep Ensembles, random search) and state-of-the-art NES approaches across FashionMNIST, CIFAR-10, and CIFAR-100, while significantly reducing computational overhead.

2 Problem statement

2.1 Neural Architecture Search

We define the fundamental components of the search space as:

$$\mathcal{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_N\} \subset \mathbb{R}^d, \quad \mathcal{O} = \{\boldsymbol{o}_1, \dots, \boldsymbol{o}_K\}, \boldsymbol{o}_i : \mathbb{R}^d \to \mathbb{R}^d,$$

where $\mathcal{V}$ is the ordered set of node feature-vectors (implying a topological order), and $\mathcal{O}$ is a finite set of elementwise operations (e.g., convolutions, poolings).

We then define the space of feasible architectures $\mathcal{A}$ as the set of all labeled directed graphs where edges strictly follow the topological order to ensure acyclicity:

$$\mathcal{A} = \{(\mathcal{V}, E) \mid E \subseteq \{(\boldsymbol{v}_i \to \boldsymbol{v}_j, \boldsymbol{o}) \mid 1 \le i < j \le N, \, \boldsymbol{o} \in \mathcal{O}\}\}.$$

Here, the condition $i < j$ restricts edges to forward connections only, intrinsically guaranteeing that every architecture in $\mathcal{A}$ is a Directed Acyclic Graph (DAG).

Consequently, a specific architecture $\boldsymbol{\alpha} \in \mathcal{A}$ can be represented as:

$$\boldsymbol{\alpha} = \big(\mathcal{V}, \{(\boldsymbol{v}_i \to \boldsymbol{v}_j, \boldsymbol{o}_{i,j}) \mid (\boldsymbol{v}_i \to \boldsymbol{v}_j) \in \mathcal{E}_\alpha\}\big),$$

where $\mathcal{E}_\alpha$ is the set of active edges satisfying $1 \le i < j \le N$, and each edge is uniquely labeled by an operation $o_{i,j} \in \mathcal{O}$.

Denote $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ as the training and validation losses, respectively. The NAS problem can then be formulated as the search for an optimal architecture $\boldsymbol{\alpha}^*$ that minimizes $\mathcal{L}_{val}(\boldsymbol{\alpha}^*, \boldsymbol{\omega}^*)$, under the constraint that the weights are obtained by minimizing the training loss:

$$\boldsymbol{\omega^*} = \arg\min_{\boldsymbol{\omega} \in \mathcal{W}} \mathcal{L}_{train}(\boldsymbol{\alpha}^*, \boldsymbol{\omega}).$$

This can be expressed as the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{\alpha} \in \mathcal{A}} \mathcal{L}_{val}(\boldsymbol{\omega}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}), \\ &\text{s.t.} \quad \boldsymbol{\omega}^*(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\omega} \in \mathcal{W}} \mathcal{L}_{train}(\boldsymbol{\omega}, \boldsymbol{\alpha}). \end{aligned} \tag{1}$$

The primary challenge in this optimization lies in the immense search space of possible architectures (e.g., in DARTS [23], it is approximately 10^{25}).

2.2 Neural Ensemble Search

The primary objective of NES is to find an optimal ensemble of neural networks whose architectures lie within the NAS search space.

As before, we denote $\boldsymbol{\alpha} \in \mathcal{A}$ as a network architecture and $\omega(\alpha)$ as its corresponding parameters. The action of this network on an input x is denoted by $f_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{\omega}(\boldsymbol{\alpha}))$. Let $S \subset \mathcal{A}$ be a subset of architectures. Then, the NES problem can be formally described as follows:

$$\begin{aligned} &\min_{\mathcal{S}} \mathcal{L}_{val}\left(\frac{1}{|S|}\sum_{\boldsymbol{\alpha} \in S} f_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{\omega}^*(\boldsymbol{\alpha}))\right), \\ &\text{s.t.} \quad \forall \boldsymbol{\alpha} \in \mathcal{S} : \ \boldsymbol{\omega}^*(\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\omega}(\boldsymbol{\alpha})} \mathcal{L}_{train}(f_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{\omega}(\boldsymbol{\alpha}))). \end{aligned} \tag{2}$$

Thus, in addition to searching over a vast number of architectures, we now also need to find the optimal ensemble composition.

3 Method

In this work, we consider the transformation of the architecture space proposed in DARTS [23] for application in Graph Attention Networks (GAT) (see Section 3.1). In Section 3.2, we present the architecture of the surrogate functions and describe its working principle, while Section 3.3 provides a detailed discussion of the ensemble construction method based on this surrogate functions, explaining how surrogate predictions are used to ensure architectural diversity within the ensemble.

The proposed approach enables ensembles that optimally trade off between predictive accuracy and architectural diversity.

3.1 Architecture Search Space

Following the conventions of [23], we instantiate the search space $\mathcal{A}$ using a cell-based approach. Each architecture $\boldsymbol{\alpha} \in \mathcal{A}$ is composed of two functional units: a *normal cell* $\boldsymbol{\alpha}_{\text{norm}}$ and a *reduction cell* $\boldsymbol{\alpha}_{\text{red}}$, such that $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_{\text{norm}}, \boldsymbol{\alpha}_{\text{red}})$. Both units are DAGs adhering to the formalisms established in Section 2.1.

Unlike the original DARTS, which utilizes continuous relaxation, our method explores $\mathcal{A}$ via discrete random sampling. We consider cells with $n = 5$ nodes and $m = 10$ edges. With an operation set size $|\mathcal{O}| = 7$, the cardinality of the cell-level search space is:

$$|\mathcal{A}_{\text{cell}}| = \prod_{k=2}^{n-1} \binom{k}{2} \cdot |\mathcal{O}|^m \approx 10^9. \tag{3}$$

Consequently, the full architecture space $|\mathcal{A}| = |\mathcal{A}_{cell}|^2 \approx 10^{18}$ remains computationally intractable for exhaustive search, necessitating surrogate-guided exploration.

To train the dual-surrogate models, we construct a dataset $\mathcal{D}_{\text{train}}$ by evaluating N sampled architectures:

$$\mathcal{D}_{\text{train}} = \{(\boldsymbol{\alpha}_i, \mathbf{y}_i, \text{acc}_i)\}_{i=1}^{N}, \tag{4}$$

where $\mathbf{y}_i = f_{\boldsymbol{\alpha}_i}(\mathcal{X}_{val}, \boldsymbol{\omega}^*(\boldsymbol{\alpha}_i))$ is the vector of model predictions on a fixed validation subset $\mathcal{X}_{val}$, and acc_i is the corresponding validation accuracy. These observations serve as the ground truth for our surrogate functions.

3.2 Surrogate Function

In order to construct the ensemble described in Section 3.3, we need to predict both the performance and the diversity of candidate architectures. Due to the enormous size of the architecture space, obtaining these characteristics via full training is infeasible. To address this, we employ surrogate models: given an architectural representation, they predict key properties of neural networks. Formally, we define

$$\mathbf{f} : \mathcal{A} \to \mathbb{R}^d,$$

where d is the dimension of the latent space. In particular, we consider two surrogates:

$$f_{\text{acc}}^{\theta} : \mathcal{A} \to \mathbb{R}, \quad \mathbf{f}_{\text{div}}^{\theta} : \mathcal{A} \to \mathbb{R}^d.$$

Each architecture is represented as a directed acyclic graph (DAG), where nodes correspond to latent representations and edges to operations, similar to NAS-Bench-201 [24]. For surrogate modeling, we adopt an alternative graph representation, inspired by NAS-Bench-101 [25]: nodes represent operations and edges their corresponding latent embeddings. Conversion is done as follows:

1. Each edge of the original graph is transformed into a node in the new graph, labeled by the operation present on that edge.
2. An oriented edge is drawn from operation $\mathbf{o}_i$ to $\mathbf{o}_j$ if in the original graph $\mathbf{o}_i$ acts from node $\mathbf{x}$ to $\mathbf{y}$ and $\mathbf{o}_j$ acts from $\mathbf{y}$ to $\mathbf{z}$.

Operations are encoded as one-hot vectors. The surrogates themselves are Graph Attention Networks (GATs) with N sequential layers, residual connections, GraphNorm, a global pooling aggregation, and lightweight output heads (Figure 2).

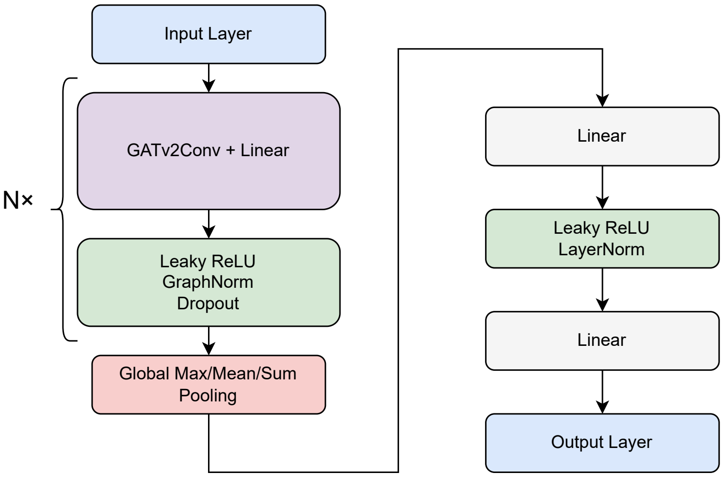

Fig. 2: Surrogate architecture: N GAT layers with residual connections and GraphNorm, global pooling, and output heads for accuracy and similarity embeddings.

The accuracy surrogate f_{acc} is trained via supervised regression to predict the performance of a given architecture. Given the training dataset $\mathcal{D}_{\text{train}} = \{(\boldsymbol{\alpha}_i, \mathbf{y}_i, \text{acc}_i)\}_{i=1}^{N}$, we optimize the parameters θ_{acc} by minimizing the mean squared error (MSE) relative to the ground-truth validation accuracy:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} \left(f_{\text{acc}}^{\theta}(\boldsymbol{\alpha}_i) - \text{acc}_i\right)^2, \tag{5}$$

In contrast, the diversity surrogate $\mathbf{f}_{\text{div}}^{\theta}$ can not be trained directly with supervised learning, since no ground-truth latent embeddings exist. Instead, we optimize it using a triplet loss on architecture embeddings.

Let us consider a set of N trained models $\{M_1, \ldots, M_N\}$ and a fixed validation dataset of K examples. Denote the predictions of model M_i on the validation dataset by the vector

$$\mathbf{y}^{(i)} = \left(y_1^{(i)}, \ldots, y_K^{(i)}\right).$$

We construct a diversity matrix $\mathbf{C} \in [0,1]^{N \times N}$, where each entry

$$c_{ij} = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\left(y_k^{(i)} = y_k^{(j)}\right)$$

measures the fraction of matching predictions between models M_i and M_j.

For the purposes of training with triplet loss, $\mathbf{C}$ is discretized into a matrix $\mathbf{D} \in \{-1, 0, 1\}^{N \times N}$ according to upper and lower quantile thresholds q_p and q_n:

$$\mathbf{D}_{ij} = \begin{cases} 1, & \text{if } c_{ij} > q_p, \\ -1, & \text{if } c_{ij} < q_n, \\ 0, & \text{otherwise.} \end{cases}$$

Algorithm 1: Training the diversity surrogate function

```
Input: $\mathcal{D}_{train}$: training dataset of architectures $\{(\boldsymbol{\alpha}_i, \mathbf{y}_i, \text{acc}_i)\}_{i=1}^{N}$;
$\mathbf{D}$: diversity matrix where $\mathbf{D}_{jk} \in \{1, -1\}$ (positive/negative);
$m$: triplet loss margin;
$\eta$: learning rate;
$B$: batch size.
Output: Optimized parameters $\theta_{div}$ for surrogate model $\mathbf{f}_{div}$
Initialize parameters $\theta_{div}$
for epoch $e \leftarrow 1$ to $n$ do
    for step $t \leftarrow 1$ to $N/B$ do
        Sample a minibatch of anchor indices $\mathcal{J} \subset \{1, \ldots, N\}$, where $|\mathcal{J}| = B$
        for each $j \in \mathcal{J}$ do
            Identify candidate sets: $\mathcal{P}_j = \{k \mid \mathbf{D}_{jk} = 1\}$ and $\mathcal{N}_j = \{k \mid \mathbf{D}_{jk} = -1\}$
            Sample positive $k_p \sim \text{Uniform}(\mathcal{P}_j)$ and negative $k_n \sim \text{Uniform}(\mathcal{N}_j)$
            Compute embeddings:
            $\mathbf{e}_a, \mathbf{e}_p, \mathbf{e}_n \leftarrow \mathbf{f}_{div}^{\theta_{div}}(\boldsymbol{\alpha}_j), \mathbf{f}_{div}^{\theta_{div}}(\boldsymbol{\alpha}_{k_p}), \mathbf{f}_{div}^{\theta_{div}}(\boldsymbol{\alpha}_{k_n})$
            Compute triplet loss for instance $j$:
            $\ell_j \leftarrow \max\left(0, ||\mathbf{e}_a - \mathbf{e}_p||_2^2 - ||\mathbf{e}_a - \mathbf{e}_n||_2^2 + m\right)$
        end
        Compute batch loss: $\mathcal{L} = \frac{1}{B} \sum_{j \in \mathcal{J}} \ell_j$
        Update parameters: $\theta_{div} \leftarrow \theta_{div} - \eta \nabla_{\theta_{div}} \mathcal{L}$        // e.g., via Adam
    end
end
return $\theta_{div}$
```

The training procedure for the diversity surrogate $\mathbf{f}_{\text{div}}$ is detailed in Algorithm 1. For each training epoch, we sample a batch of triplets $(\boldsymbol{\alpha}_a, \boldsymbol{\alpha}_p, \boldsymbol{\alpha}_n)$ where the relationships are defined by the discretized diversity matrix $\mathbf{D}$. Specifically, for a given anchor architecture $\boldsymbol{\alpha}_a$, the positive $\boldsymbol{\alpha}_p$ and negative $\boldsymbol{\alpha}_n$ examples are sampled uniformly from the precomputed sets:

$$\mathcal{P}_a = \{k \mid \mathbf{D}_{ak} = 1\}, \quad \mathcal{N}_a = \{k \mid \mathbf{D}_{ak} = -1\}. \tag{6}$$

The architectures are subsequently mapped into a d-dimensional latent space via the surrogate function $\mathbf{f}_{\text{div}}$ with parameters θ_{div}:

$$\mathbf{e}_a = \mathbf{f}_{\text{div}}^{\theta_{\text{div}}}(\alpha_a), \quad \mathbf{e}_p = \mathbf{f}_{\text{div}}^{\theta_{\text{div}}}(\alpha_p), \quad \mathbf{e}_n = \mathbf{f}_{\text{div}}^{\theta_{\text{div}}}(\alpha_n). \tag{7}$$

The parameters θ_{div} are optimized by minimizing the triplet loss objective:

$$\mathcal{L}_{\text{div}} = \sum_{(a,p,n)} \max\left(\|\mathbf{e}_a - \mathbf{e}_p\|_2^2 - \|\mathbf{e}_a - \mathbf{e}_n\|_2^2 + m, 0\right), \tag{8}$$

where $m > 0$ is a margin hyperparameter that enforces a minimum separation between dissimilar architectures in the embedding space. This optimization scheme encourages the surrogate to learn a manifold where Euclidean distances reflect architectural diversity.

The ground-truth matrix $\mathbf{D}$ used to guide the triplet sampling process is constructed by thresholding pairwise diversity metrics calculated on model outputs. In this work, we primarily utilize the fraction of identical predictions as the diversity measure, although the proposed framework is inherently compatible with other metrics, such as the Jensen-Shannon divergence, Hellinger distance, or the correlation of prediction vectors $\mathbf{y}_i$. These metrics yield an $N \times N$ diversity matrix which is subsequently discretized to facilitate stable triplet sampling and the formation of a structured latent space.

3.3 Ensemble Construction

Once the surrogate models $f_{\text{acc}}^{\theta_{\text{acc}}}$ and $\mathbf{f}_{\text{div}}^{\theta_{\text{div}}}$ are trained, they enable an efficient search for the optimal ensemble without requiring the prohibitive computational cost of training intermediate candidate architectures. The proposed ensemble construction (Algorithm 2) proceeds in two distinct phases: *candidate pool filtering* and *diversity-driven greedy selection*.

In the first phase, we generate a large set of random candidate architectures $\mathcal{A}_{\text{cand}}$ via discrete sampling from the search space. We then use the dual-surrogate framework to predict the accuracy s and the latent diversity embedding $\mathbf{e}$ for each candidate. To ensure the base quality of the ensemble, we apply an accuracy threshold α, forming a pruned candidate pool $\mathcal{P}$ that contains only high-performing models.

Algorithm 2: Surrogate-Assisted Ensemble Construction

```
Input: $K$: target ensemble size;
$N$: number of initial candidates;
$\alpha$: accuracy threshold;
$f_{\text{acc}}, \mathbf{f}_{\text{div}}$: trained surrogate models.
Output: $\mathcal{A}_{\text{best}}$: selected ensemble of $K$ architectures.
// Phase 1: Candidate Pool Construction
$\mathcal{A}_{\text{cand}} \leftarrow \text{RandomSample}(N)$
$\mathcal{P} \leftarrow \big\{(\alpha, s, \mathbf{e}) \mid \alpha \in \mathcal{A}_{\text{cand}}, s = f_{\text{acc}}(\alpha), \mathbf{e} = \mathbf{f}_{\text{div}}(\alpha), s \geq \alpha\big\}$
// Phase 2: Diversity-driven Greedy Selection
Select $\alpha^* \in \mathcal{P}$ with highest predicted accuracy: $\alpha^* = \arg\max s$
$\mathcal{A}_{\text{best}} \leftarrow \{\alpha^*\}$;   $\mathcal{E}_{\text{sel}} \leftarrow \{\mathbf{e}^*\}$;   $\mathcal{P} \leftarrow \mathcal{P} \setminus \{(\alpha^*, s^*, \mathbf{e}^*)\}$
while $|\mathcal{A}_{best}| < K$ and $\mathcal{P} \neq \emptyset$ do
    For each candidate $i \in \mathcal{P}$, compute average distance to the current ensemble:
        $$d_i = \frac{1}{|\mathcal{E}_{\text{sel}}|} \sum_{\mathbf{e}_{\text{sel}} \in \mathcal{E}_{\text{sel}}} \|\mathbf{e}_i - \mathbf{e}_{\text{sel}}\|_2 \qquad (9)$$
    Select architecture with maximum diversity: $i^* = \arg\max_{i \in \mathcal{P}} d_i$
    Update ensemble: $\mathcal{A}_{\text{best}} \leftarrow \mathcal{A}_{\text{best}} \cup \{\alpha_{i^*}\}$;   $\mathcal{E}_{\text{sel}} \leftarrow \mathcal{E}_{\text{sel}} \cup \{\mathbf{e}_{i^*}\}$
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{(\alpha_{i^*}, s_{i^*}, \mathbf{e}_{i^*})\}$
end
return $\mathcal{A}_{best}$
```

The second phase employs a greedy forward-selection strategy to maximize architectural diversity. Starting with the most accurate model as an anchor, the algorithm iteratively selects subsequent architectures that exhibit the maximum average Euclidean distance from the currently selected set in the latent space. By leveraging the geometric organization of the latent space provided by $\mathbf{f}_{\text{div}}$, this selection process explicitly favors architectures with dissimilar prediction patterns. This approach effectively identifies a diverse set of high-performing models, balancing individual predictive power with collective ensemble robustness.

4 Computational Experiment

In this section we present the experimental results as well as the metrics used for comparison. In Section 4.1, we describe the dataset collection procedure. Section 4.2 details the training setup for the surrogate functions. Finally, in Sections 4.3, 4.4, 4.5, we compare our proposed method against DeepEnsemble [13] and Random Search [26] for FashionMNIST, CIFAR10 and CIFAR100, respectively.

4.1 Dataset Construction

The models in our dataset were trained according to the following algorithm:

1. Sample architecture from the DARTS search space.
2. Split the dataset into training and validation subsets with an 20%/80% ratio.
3. Train each sampled architecture on the training subset.
4. For each trained model, record:
   - its architectural description,
   - its predictions on the validation subset,
   - its validation accuracy.

For training the models, we adopt the hyperparameter configuration from [23], varying only the number of epochs, the number of cells, and the channel width.

Table 1 summarizes the training hyperparameters and results for all sampled models:

Tab. 1: Training Hyperparameters and Performance per Dataset

Dataset	Number of Cells	Initial Width	Num. Epochs	Avg. Accuracy (%)	Avg. Diversity
FashionMNIST	**3**	**16**	**125**	89.6 ± 0.5	0.900 ± 0.004
CIFAR-10	**8**	**16**	**200**	75.8 ± 0.6	0.693 ± 0.006
CIFAR-100	**8**	**16**	**200**	37.6 ± 1.1	0.324 ± 0.008

To enhance the efficiency of the candidate generation process, we implement an iterative surrogate-assisted refinement strategy. Initially, a preliminary set of $N_1 = 1000$ architectures is sampled to perform the baseline training of the accuracy surrogate $f_{\text{acc}}^{\theta_{\text{acc}}}$. Following this, we filter the search space by retaining only the top 10% of candidates based on their predicted accuracy.

This is followed by the generation of an additional $N_2 = 2000$ architectures within the high-performance regions identified by the surrogate. Notably, on datasets of lower complexity, such as FashionMNIST and CIFAR-10, this procedure results in a characteristic bi-modal distribution of model properties (see Figures 4, 6).

4.2 Training of the Surrogate Functions

In our experiment, each cell contains $n = 5$ nodes, with two outgoing edges per node assigned random operations from $\mathcal{O}$, yielding $m = 10$ edges per cell. The operation set $\mathcal{O}$ includes:

- Separable convolutions (3×3, 5×5)
- Dilated separable convolutions (3×3, 5×5)
- 3×3 max pooling
- 3×3 average pooling
- Skip connection

We employed a graph attention network (GAT) with four convolutional layers and two fully connected layers at the output, i.e. $N = 4$ (see Figure 2). The training of the surrogate functions used a cosine annealing learning rate scheduler.

Full models are constructed by stacking normal and reduction cells in a 2:1 ratio, starting from a predefined number of initial channels. Each model combines one normal and one reduction cell sampled independently from $\mathcal{A}$.

4.3 FashionMNIST

For constructing the diversity matrix, we used $q_p = 0.9$ and $q_n = 0.1$. In the diversity regime, the GAT surrogate has an output dimensionality of 16, employs 16 attention heads without dropout, and is trained for 50 epochs with a cosine-annealed learning rate from 5×10^{-4} to 1×10^{-5}. For accuracy prediction, the surrogate outputs a single scalar, uses 16 attention heads, and is trained for 125 epochs with a cosine-annealed learning rate from 5×10^{-3} to 1×10^{-4}. Both models are optimized with Adam on a 2400/600 train/validation split.

Subsequently, the models obtained via the surrogate functions were trained for 125 epochs using stochastic gradient descent (SGD) with a cosine-annealed learning rate, starting from 0.025 and decreasing to 0.001. Training was performed with a weight decay of 3×10^{-4}, an auxiliary loss weight of 0.4, a network composed of 3 cells with width 16, and a batch size of 96.

Tab. 2: FashionMNIST ensemble results (Updated DeepEns).

Metric	**Surrogate Ensemble**	**DeepEns**	**Random Search**
Top-1 Accuracy (%)	$\mathbf{95.3 \pm 0.1}$	95.02 ± 0.35	95.01 ± 0.19
Average Model Accuracy (%)	$\mathbf{94.7 \pm 0.1}$	94.55 ± 0.43	94.32 ± 0.24
NLL	$\mathbf{0.263 \pm 0.003}$	0.266 ± 0.006	0.265 ± 0.003
Oracle NLL	0.199 ± 0.010	0.206 ± 0.003	$\mathbf{0.191 \pm 0.010}$
Brier Score	$\mathbf{0.089 \pm 0.001}$	0.091 ± 0.006	0.092 ± 0.002
ECE	0.124 ± 0.002	0.121 ± 0.001	$\mathbf{0.120 \pm 0.004}$
Ambiguity	0.0058 ± 0.0009	0.0047 ± 0.0009	$\mathbf{0.0069 \pm 0.0013}$
Normalized Disagreement	0.065 ± 0.008	0.062 ± 0.008	$\mathbf{0.077 \pm 0.009}$
Predictive Disagreement	0.130 ± 0.016	0.125 ± 0.016	$\mathbf{0.154 \pm 0.018}$
Number of models	3	3	3
Total samples	10000	10000	10000

The performance results, summarized in Table 2, demonstrate that the proposed *Surrogate Ensemble* achieves the highest classification performance with a Top-1 accuracy of 95.3%. An analysis of the relationship between individual model performance and ensemble efficacy yields several insights.

While the individual models identified by the DARTS-based DeepEns outperform those from Random Search on average (94.55% vs. 94.32%), both ensembles yield statistically equivalent Top-1 accuracies (95.02% and 95.01%, respectively). This phenomenon underscores the critical role of ensemble diversity. Specifically, Random Search leads across all evaluated diversity metrics, including Ambiguity, Normalized Disagreement, and Predictive Disagreement.

Furthermore, Random Search achieves the lowest Oracle NLL (0.191), suggesting that increased diversity compensates for lower individual model accuracy. These results provide empirical evidence that ensemble diversity remains a key performance driver even in primitive structured datasets like FashionMNIST.

Robustness analysis under adversarial attacks (Figure 3) shows comparable degradation across all methods. No single approach demonstrates a statistically significant advantage, suggesting that the intrinsic vulnerability of the search space on FashionMNIST outweighs the impact of ensemble selection strategies.

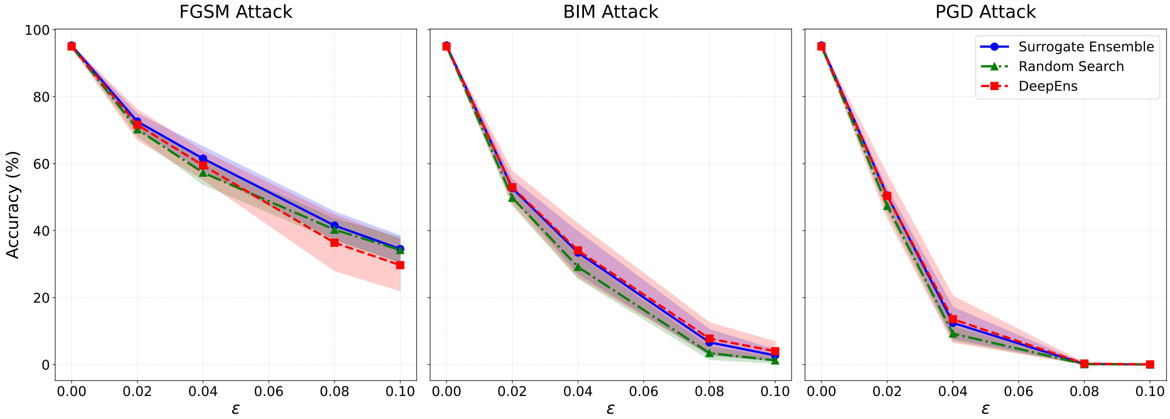

Fig. 3: Accuracy of the surrogate ensemble under FGSM, BIM, and PGD attacks across increasing ε for FashionMNIST.

Finally, the effectiveness of the diversity surrogate is confirmed by the latent space analysis in Figure 4. The strong negative monotonic relationship (Spearman's $\rho = -0.70$) for 1,000 architecture pairs from a test set validates that $\mathbf{f}_{\text{div}}$ generalizes well to unseen models. However, the high density of pairs within a narrow embedding region confirms significant architectural redundancy on this dataset, which limits the potential for diversity-driven selection to further enhance ensemble performance.

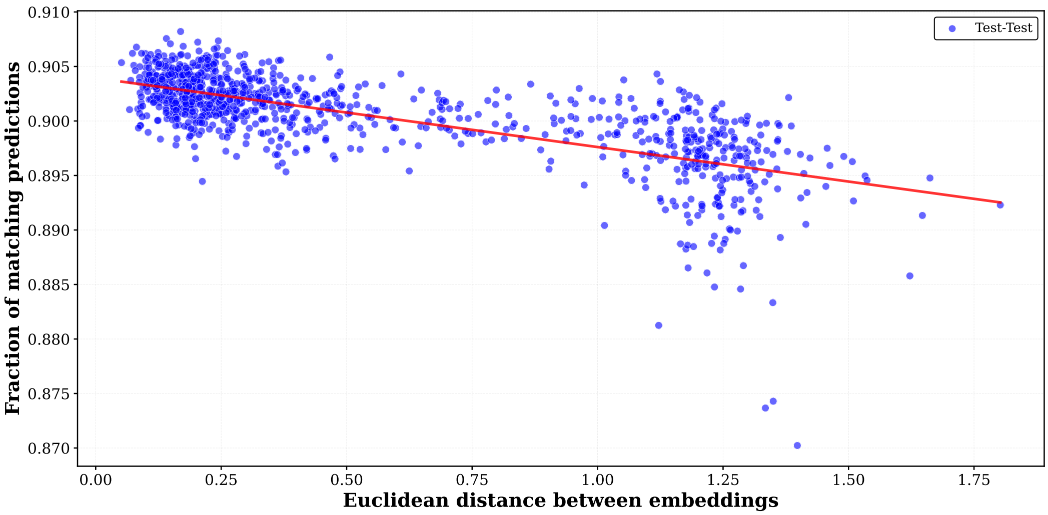

Fig. 4: Relationship between embedding distance and prediction similarity for test-test model pairs on FashionMNIST. Each point corresponds to a pair of models.

4.4 CIFAR10

Diversity was measured using the overlap metric with thresholds $q_p = 0.9$ and $q_n = 0.1$. The GAT surrogates (16 dim/heads) were optimized for 20 and 100 epochs for the diversity and accuracy tasks, respectively, using cosine annealing. The selected architectures were subsequently trained for 600 epochs (20 cells, width 36) following the protocol in Section 4.3, while keeping other hyperparameters unchanged.

The results in Table 3 demonstrate that the Surrogate Ensemble achieves superior predictive performance, securing the highest Top-1 Accuracy (97.80%) and lowest NLL (0.208).

Tab. 3: CIFAR10 ensemble results (Updated DeepEns)

Metric	Surrogate Ensemble	DeepEns	Random Search
Top-1 Accuracy (%)	$\mathbf{97.80 \pm 0.08}$	97.64 ± 0.09	97.56 ± 0.11
Average Model Accuracy (%)	$\mathbf{96.73 \pm 0.10}$	96.71 ± 0.14	96.48 ± 0.09
NLL	$\mathbf{0.208 \pm 0.003}$	0.210 ± 0.003	0.212 ± 0.002
Oracle NLL	0.156 ± 0.002	$\mathbf{0.154 \pm 0.004}$	0.158 ± 0.003
Brier Score	$\mathbf{0.055 \pm 0.001}$	0.056 ± 0.001	0.057 ± 0.001
ECE	0.136 ± 0.002	$\mathbf{0.135 \pm 0.002}$	0.136 ± 0.001
Ambiguity	$\mathbf{0.011 \pm 0.001}$	0.009 ± 0.001	$\mathbf{0.011 \pm 0.001}$
Normalized Disagreement	0.044 ± 0.001	0.046 ± 0.003	$\mathbf{0.046 \pm 0.001}$
Predictive Disagreement	0.087 ± 0.003	$\mathbf{0.093 \pm 0.007}$	0.092 ± 0.003
Number of models	5	5	5
Total samples	10000	10000	10000

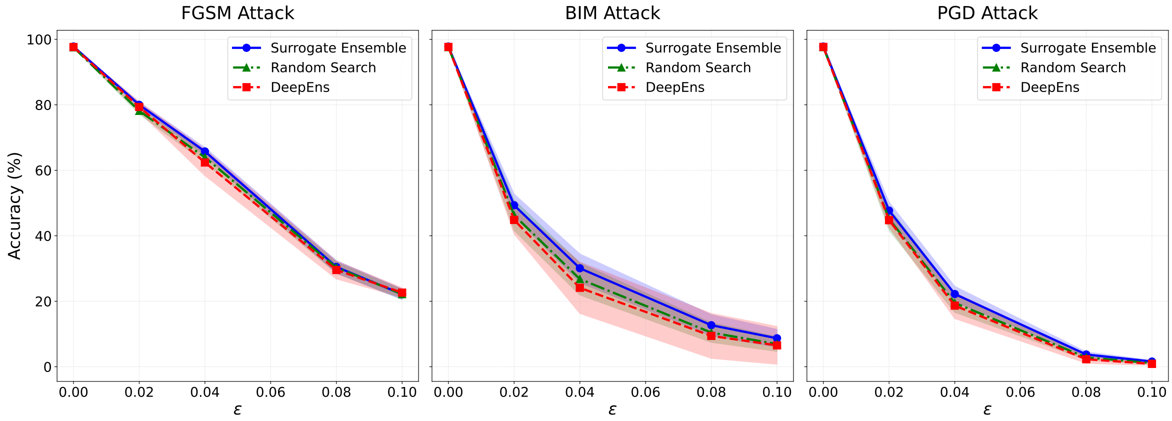

Fig. 5: Accuracy of the surrogate ensemble and DeepEns under FGSM, BIM, and PGD attacks across increasing ε on CIFAR10.

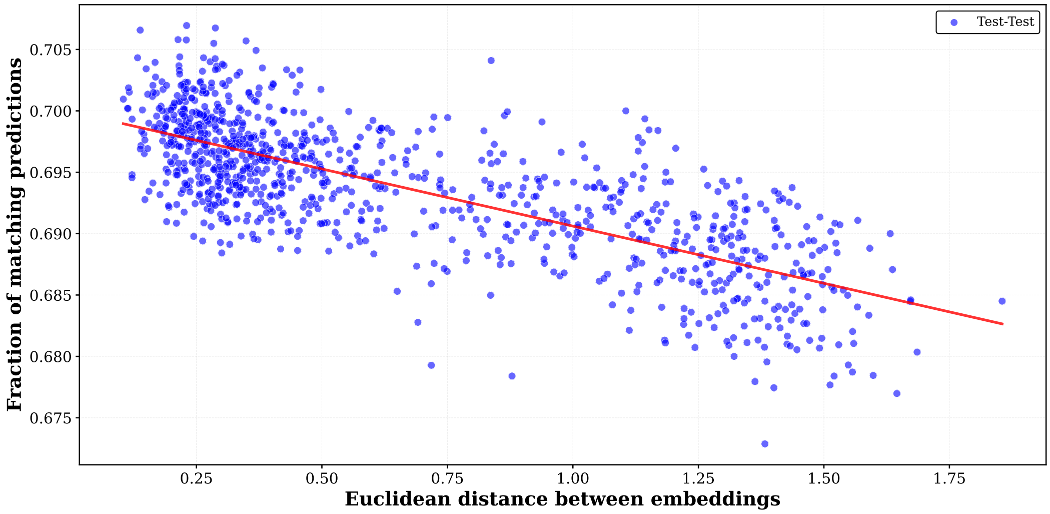

Fig. 6: Relationship between embedding distance and prediction similarity on CIFAR-10 ($\rho = -0.73$).

Crucially, while our individual models share comparable accuracy with DeepEns ($\approx 96.7\%$), the ensemble benefits from enhanced diversity—evidenced by high Ambiguity (0.011) matching Random Search. This confirms that our dual-surrogate framework successfully balances high individual performance with the architectural heterogeneity required for complex datasets.

The benefit of this diversity extends to adversarial robustness (Figure 5), where our method maintains a consistent advantage under FGSM, BIM, and PGD attacks. Furthermore, latent space analysis (Figure 6) reveals a strong negative correlation ($\rho = -0.73$) between embedding distance and prediction similarity. The wider dispersion in CIFAR-10's embedding space allows the surrogate to effectively navigate a rich candidate pool, bridging the gap between the accuracy of DeepEns and the diversity of Random Search to achieve state-of-the-art performance.

4.5 CIFAR-100

Using the identical surrogate configuration as in Section 4.4, the results in Table 4 highlight the efficacy of the proposed method on complex datasets. The Surrogate Ensemble achieves a dominant Top-1 Accuracy of 85.17%, surpassing both DeepEns (84.16%) and Random Search (84.41%), while also securing the best probabilistic metrics (NLL 0.692, Brier 0.234).

Tab. 4: CIFAR100 ensemble results.

Metric	Surrogate Ensemble	DeepEns	Random Search
Top-1 Accuracy (%)	$\mathbf{85.17 \pm 0.16}$	84.16 ± 0.00	84.41 ± 0.09
Average Model Accuracy (%)	$\mathbf{80.50 \pm 0.12}$	79.54 ± 0.00	79.57 ± 0.19
NLL	$\mathbf{0.692 \pm 0.007}$	0.735 ± 0.000	0.718 ± 0.006
Oracle NLL	$\mathbf{0.403 \pm 0.004}$	0.441 ± 0.000	0.418 ± 0.006
Brier Score	$\mathbf{0.234 \pm 0.002}$	0.247 ± 0.000	0.244 ± 0.002
ECE	0.140 ± 0.004	$\mathbf{0.135 \pm 0.000}$	0.141 ± 0.002
Ambiguity	0.047 ± 0.001	0.046 ± 0.000	$\mathbf{0.048 \pm 0.002}$
Normalized Disagreement	0.206 ± 0.002	0.207 ± 0.000	$\mathbf{0.215 \pm 0.002}$
Predictive Disagreement	0.411 ± 0.004	0.414 ± 0.000	$\mathbf{0.430 \pm 0.004}$
Number of models	5	5	5
Total samples	10000	10000	10000

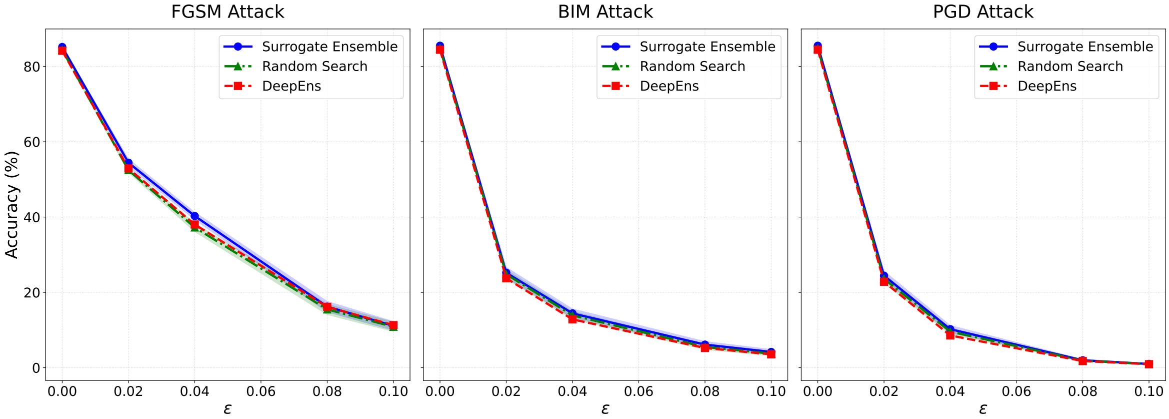

Fig. 7: Adversarial robustness on CIFAR-100.

Crucially, the results illustrate the limitations of diversity without quality. While Random Search maximizes diversity metrics (e.g., Ambiguity 0.048), its ensemble performance is hampered by the stochastic inclusion of sub-optimal architectures. In contrast, our method prioritizes high-quality individual models (Average Accuracy 80.50% vs. 79.57% for RS) while leveraging the latent space structure—characterized by a strong correlation ($\rho = -0.71$) between embedding distance and prediction similarity (Figure 8)-to select complementary candidates. Regarding adversarial robustness (Figure 7), all methods exhibit comparable degradation, suggesting that on the complex CIFAR-100 manifold, the search space's intrinsic vulnerability outweighs the benefits of ensemble selection strategies.

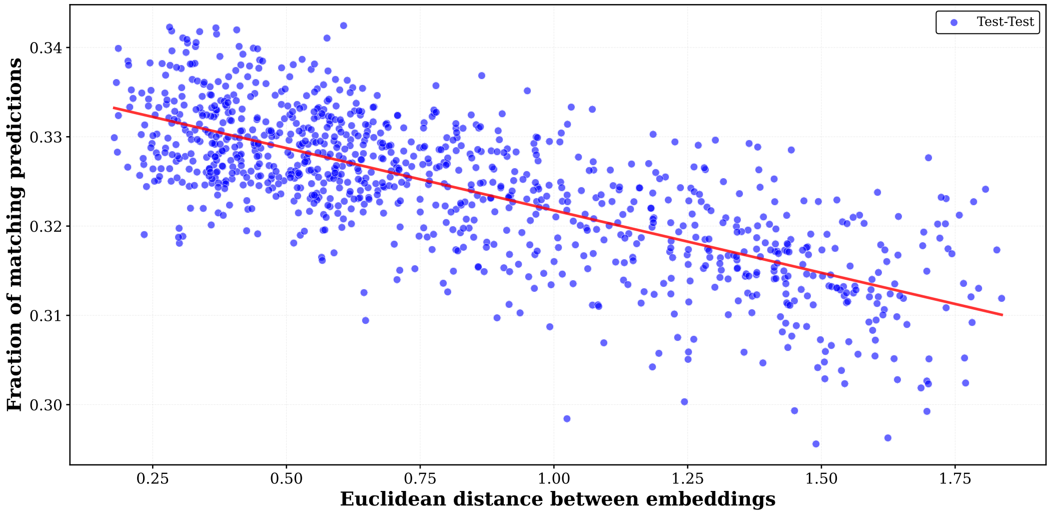

Fig. 8: Latent space correlation ($\rho = -0.71$) on CIFAR-100.

5 Conclusion

In this study, we demonstrated that surrogate diversity functions can be effectively learned through embedding distances in a latent space. Our experiments revealed consistently strong negative correlations between the Euclidean distance of model embeddings and the proportion of identical predictions across FashionMNIST, CIFAR-10, and CIFAR-100 (Spearman's ρ of -0.70, -0.73, and -0.71, respectively). Building on these findings, we proposed an ensemble construction algorithm that simultaneously leverages accuracy and diversity surrogates to select high-performing, complementary models. Our results indicate that the efficacy of this approach is closely tied to the structural complexity of the dataset: while simple datasets favor ensembles composed of the most accurate individual models, more complex datasets allow diversity-driven selection to yield significant performance gains.

The primary limitation of the current approach is the substantial computational overhead required to train a large pool of base models for surrogate dataset generation. Future research should focus on mitigating these costs, perhaps through more efficient sampling or weight-sharing techniques.

Furthermore, we identify the investigation of the square root of Jensen-Shannon divergence as a promising alternative similarity metric. Unlike discrete response-proportion measures, a continuous metric based on Jensen-Shannon distance may provide a smoother optimization landscape, potentially refining diversity estimates. Additionally, situating these empirical findings within the formal framework of the bias-variance-covariance decomposition could offer deeper theoretical insights into the fundamental interplay between model accuracy and architectural diversity in ensemble performance.

References

[1] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990. 10.1109/34.58871.

[2] Ye Ren, Le Zhang, and P. N. Suganthan. Ensemble classification and regression–recent developments, applications and future directions. *IEEE Computational Intelligence Magazine*, 11(1):41–53, February 2016. ISSN 1556-603X. 10.1109/MCI.2015.2471235.

[3] Corinna Cortes, Xavier Gonzalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. AdaNet: Adaptive structural learning of artificial neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 874–883. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/cortes17a.html.

[4] Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris C. Holmes, Frank Hutter, and Yee Whye Teh. Neural ensemble search for uncertainty estimation and dataset shift. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34*, pages 7898–7911, 2021.

[5] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *AAAI*, volume 33, pages 4780–4789. AAAI Press, 2019.

[6] Yuqiao Liu, Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, and Kay Chen Tan. A survey on evolutionary neural architecture search. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):550–570, 2023. 10.1109/TNNLS.2021.3100554.

[7] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*. OpenReview.net, 2017.

[8] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: Stochastic neural architecture search. In *ICLR*. OpenReview.net, 2019.

[9] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P. Xing. Neural architecture search with bayesian optimisation and optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, volume 31, pages 201–211, 2018.

[10] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *KDD*, pages 1946–1956. ACM, 2019. 10.1145/3292500.3330648.

[11] Chetan Swarup, Kamred Udham Singh, Ankit Kumar, Saroj Kumar Pandey, Neeraj Varshney, and Teekam Singh. Brain tumor detection using cnn, alexnet and googlenet ensembling learning approaches. *Electronic Research Archive*, 31(5): 2900–2924, 2023. ISSN 2688-1594. 10.3934/era.2023146.

[12] Zhichao Lu, Ran Cheng, Shihua Huang, Haoming Zhang, Changxiao Qiu, and Fan Yang. Surrogate-assisted multi-objective neural architecture search for real-time semantic segmentation. *CoRR*, abs/2208.06820, 2022. 10.48550/ARXIV.2208.06820.

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*, volume 30, pages 6405–6416, 2017.

[14] Yao Shu, Yizhou Chen, Zhongxiang Dai, and Bryan Kian Hsiang Low. Neural ensemble search via bayesian sampling. In C. C. Aggarwal and Z. Zhou, editors, *UAI*, volume 180, pages 1803–1812, 2022.

[15] Minghao Chen, Jianlong Fu, and Haibin Ling. One-shot neural ensemble architecture search by diversity-guided search space shrinking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16530–16539, June 2021.

[16] Wei Wen, Hanxiao Liu, Yiran Chen, Hai Li, Gabriel Bender, and Pieter-Jan Kindermans. Neural predictor for neural architecture search. In A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, editors, *ECCV*, pages 660–676. Springer, 2020. 10.1007/978-3-030-58580-8_39.

[17] Zhichao Lu, Kalyanmoy Deb, Erik Goodman, Wolfgang Banzhaf, and Vishnu Naresh Boddeti. Nsganetv2: Evolutionary multi-objective surrogate-assisted neural architecture search. In *European conference on computer vision*, pages 35–51. Springer, 2020.

[18] Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10293–10301, 2021.

[19] Maria G. Baldeon Calisto and Susana K. Lai-Yuen. Emonas-net: Efficient multiobjective neural architecture search using surrogate-assisted evolutionary algorithm for 3d medical image segmentation. *Artificial Intelligence in Medicine*, 119: 102154, 2021. 10.1016/j.artmed.2021.102154.

[20] Yu Xue, Zhenman Zhang, and Ferrante Neri. Similarity surrogate-assisted evolutionary neural architecture search with dual encoding strategy. *Electronic Research Archive*, 32(2):1017–1043, 2024. ISSN 2688-1594. 10.3934/era.2024050.

[21] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2017. 10.48550/arxiv.1710.10903.

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 10.1109/CVPR.2015.7298682.

[23] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*. OpenReview.net, 2018.

[24] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. *CoRR*, 2020. 10.48550/arXiv.2001.00326.

[25] Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. *CoRR*, abs/1902.09635, 2019. 10.48550/arxiv.1902.09635.

[26] Liam Li and Ameet Talwalkar. Random search and reproducibility for neural architecture search. In J. Peters and D. Sontag, editors, *UAI*, volume 124, pages 367–377. PMLR, 2020.

A NAS-Bench formats

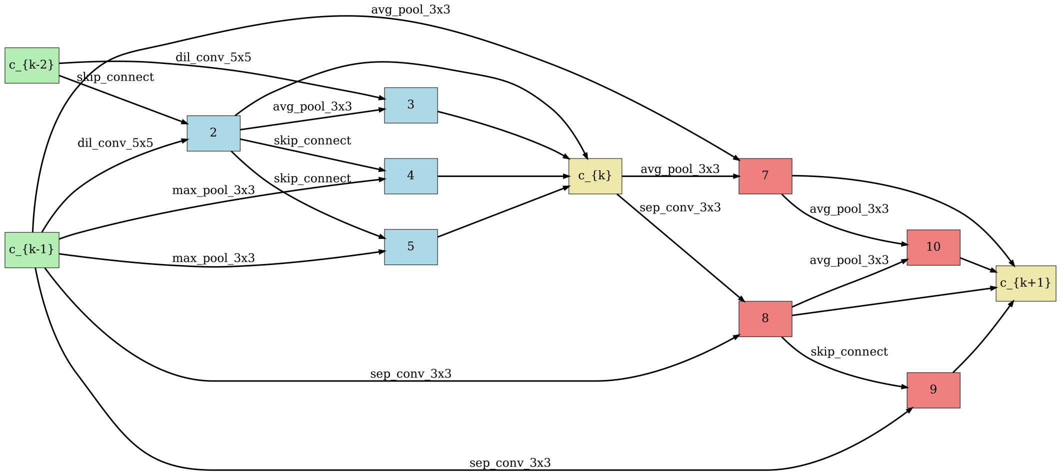

Fig. 9: Combined normal and reduced cells. The red vertices belong to the reduction cell; the blue vertices belong to the normal cell.

B Distributions of models in dataset

The main distributions of trained models you can see on Figure 11 and Figure 12:

C Training curve for surrogate funcitons

You can see the process of training accuracy surrogate function on Figure 14 and simularity surrogate function on Figure 13

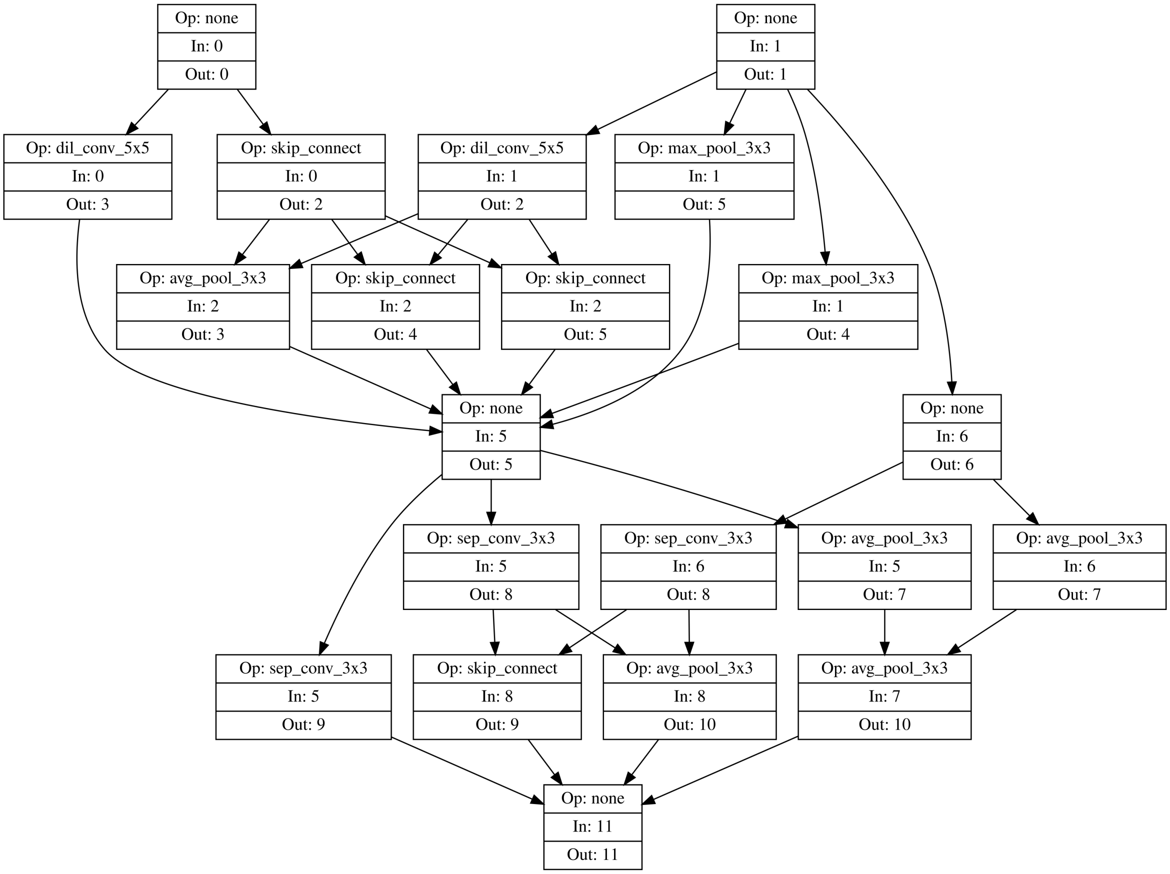

Fig. 10: Conversion of an architecture to the NAS-Bench-101 format.

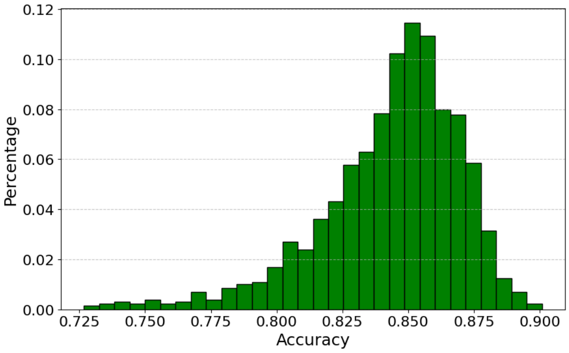

Fig. 11: Distribution of model accuracies

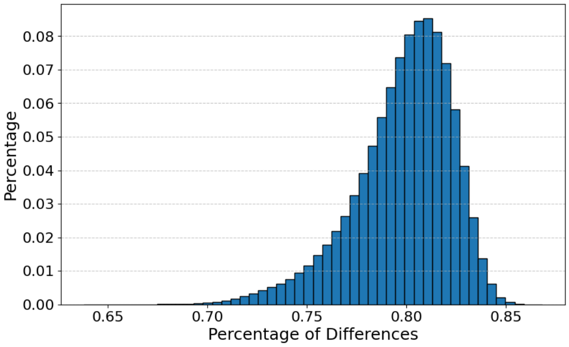

Fig. 12: Distribution of inter-model diversity

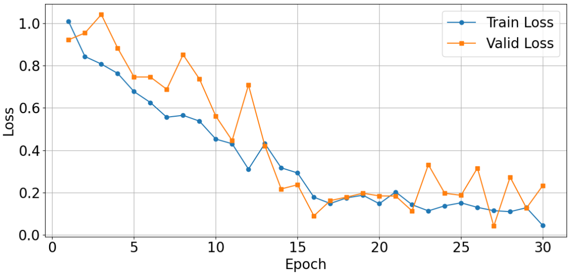

Fig. 13: Training curve of the surrogate model for diversity

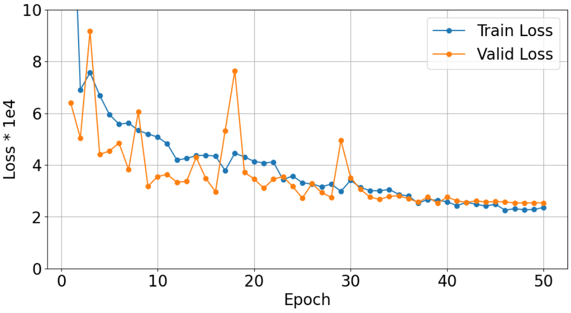

Fig. 14: Training curve of the surrogate model for accuracy