SN Computer Science (2025) 6:631
https://doi.org/10.1007/s42979-025-04056-4

ORIGINAL RESEARCH

Improving Novelty Search with a Surrogate Model and Accuracy Objectives to Build High-Performing Ensembles of Classifiers

Rui P. Cardoso[1] · Emma Hart[2] · Jeremy V. Pitt[1]

Received: 2 September 2022 / Accepted: 13 May 2025

Abstract

Neuroevolution combined with Novelty Search to promote behavioural diversity is capable of constructing high-performing ensembles for classification. However, using gradient descent to train evolved architectures during the search can be computationally prohibitive. We have proposed a method to overcome this limitation by using a *surrogate* model which estimates the behavioural distance between two neural network architectures, required to calculate novelty scores. This has demonstrated a speedup of 10 times over previous work, significantly improving on previous reported results on three benchmark datasets from Computer Vision—CIFAR-10, CIFAR-100, and SVHN. This method makes an explicit search for diversity considerably more tractable *for the same bounded resources*. Here we investigate a range of search methods that span the full spectrum of favouring accuracy, diversity, or different combinations of both. Surprisingly, we show that multiple unique combinations between a diversity metric and accuracy give rise to similar results. This enables us to posit the existence of a diversity-accuracy duality in ensembles of classifiers, which suggests that there might not be a need to find a trade-off between the two.

Keywords Diversity · Ensemble · Novelty search · Surrogate · Local competition

Introduction

A typical approach to defining a classifier ensemble requires two phases: creating a large set of potential classifiers, then selecting an appropriate subset to form an ensemble. Ensemble performance is fundamentally dependent on both the accuracy of individual base learners and the diversity between them [1]. However, techniques to promote diversity are typically only implicit, such as training the models on different subsets of the data or starting from different random initialisations. In previous work [2], we proposed a method that *explicitly* searched for diversity amongst a set of base learners by making use of metrics for measuring *behavioural* diversity. However, a fundamental limitation of this approach was its computational complexity, with a costly step of training all the neural network models in the population at each step of the search. Such time and computational demands compromise the goal of our approach, which is to develop learning algorithms which scale horizontally, namely with models which can be distributed across many low-cost machines.

In order to overcome the costly step of training each model, in [3] we have introduced a surrogate model [4] into our Novelty Search (NS) method. Novelty Search [5] is a type of evolutionary algorithm which rewards behavioural novelty, in a search space defined by the user, rather than objective fitness. We combine a NS algorithm with a surrogate model, pretrained on a sample drawn from the search space of neural network architectures, to get an estimate of the *error* distance between two neural networks given architectural descriptors, *without* training these networks. These distance estimates are required in order to calculate novelty scores during the NS. Whereas this calculation had previously been a very costly step, this technique renders it essentially instantaneous. This produces a speedup of 10 times compared to the previous approach when the same parameters are used, *without loss of performance*. By

✉ Rui P. Cardoso
ruipepeicar@gmail.com

Emma Hart
e.hart@napier.ac.uk

Jeremy V. Pitt
j.pitt@imperial.ac.uk

[1] Department of Electrical and Electronic Engineering, Imperial College London, London, UK

[2] Department of Computing, Edinburgh Napier University, Edinburgh, UK

Published online: 11 July 2025

changing the parameters to expand the search space of neural network architectures we have considerably improved on previous results reported on three benchmark datasets from Computer Vision—CIFAR-10, CIFAR-100, and SVHN. We choose these datasets because training up a neural network architecture on them is straightforward and well understood [6]. The surrogate model allows exploration of a broader range of architectures and supports longer search durations.

In [7], we take the first steps towards extending this augmented NS procedure with accuracy objectives in order to study the trade-off between diversity and accuracy in ensemble learning. This has been the subject of extensive research, e.g. [8–10]. In this paper, we consider several strategies for combining diversity and accuracy objectives along the two phases mentioned above, ranging the full spectrum between favouring only explicit diversity and only explicit individual model accuracy, with different combinations in between. We measure diversity by a number of *diversity metrics*, always employing the surrogate technique first proposed in [3] to reduce computational burden, which facilitates an extensive search of potential architectures and, therefore, ensembles. The surrogate models are used to estimate (1) the distance between neural network architectures, which is required to drive the NS method, and (2) the accuracy of a network. In this way, we are able to conduct a thorough study to investigate whether there is indeed a fundamental tension between accuracy and diversity.

Figure 1 illustrates the workflow of this paper w.r.t. its contributions. Its major contribution is that it brings together previous work with the aim of describing a complete framework for implementing horizontal scaling of learning algorithms. It also extends previous work with additional methods and results. "Novelty Search Augmented with a Surrogate Model" section describes the NS method with a surrogate model first presented in [3] and "Generic Search Method with Accuracy Objectives" section details the new extension of this method to a generic search method that can incorporate accuracy objectives, which was briefly outlined in [7]. Additional experiments are described in "Test Set 3: Introducing Accuracy Objectives" section and new results are discussed in "Results for the Novelty Search Extended with Accuracy Objectives" and "The Diversity-Accuracy Duality" section. Explicitly creating diversity amongst the members of an ensemble establishes a sound criterion for distributing these models. By improving the method with a surrogate model in the way described above, our approach makes an explicit search for diversity considerably more tractable *for the same bounded resources*. Our experimental results on three problems from Computer Vision (CV)—CIFAR-10, CIFAR-100 [11], and SVHN [12]—also show that incorporating accuracy objectives significantly improves ensemble accuracy for the worst-performing diversity metrics, but not for the best ones.

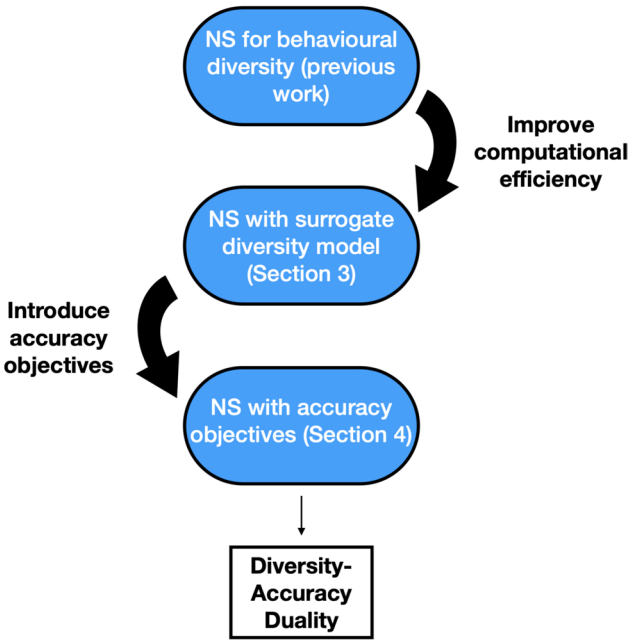


Fig. 1 Workflow of this paper w.r.t. its contributions

Building on from the preliminary observations made in [7], this paper presents and discusses additional results which shed light on the surprising result that there are multiple equivalent ways of combining the best diversity metrics with accuracy objectives that lead to ensembles of similar measured diversity and average individual accuracy. This includes even a method which only makes use of explicit accuracy objectives, with diversity being generated in an implicit manner by the evolutionary procedure. This means that, in the cases we study here, there is no dichotomy between diversity and accuracy in ensembles of classifiers; each contributes to final performance without detriment to the other and weighting one more does not impact negatively upon the other. For the problem domains we have tackled, there appears to be a fundamental *equivalence* between searching for diversity and searching for individual model accuracy. This equivalence between utilising diversity or accuracy objectives suggests that the two are interchangeable in some conditions and, therefore, enables us to posit the existence of a diversity-accuracy duality in ensembles of classifiers. This is a highly significant result as it suggests that there might not be a need to find a trade-off between accuracy and diversity.

Background

In previous work [2, 3], we proposed methods that *explicitly* search for behavioural diversity amongst a set of base learners. They use a Novelty Search (NS) [5] approach

in conjunction with Neuroevolution. Floreano et al. [13] provides an overview of Neuroevolution. In this section, we focus on surrogate modelling, ensemble diversity, and the trade-off between diversity and individual accuracy in ensembles.

In the method we proposed in [2], novelty was determined by novel metrics that explicitly measured behavioural diversity. The evolved behaviourally diverse ensembles outperformed both their individual learners and ensembles created with techniques that only implicitly promote diversity. This work also enabled us to study and compare different definitions of diversity. However, a fundamental limitation was its computational complexity. In order to calculate the behavioural distance between two models, we need to compare the classification errors that they make on a validation data set. This requires first training each member of the current population of neural network models on a training data set with gradient descent at each iteration of the NS. If computational resources are limited, this very time-consuming step can be prohibitive. This poses significant challenges because it restricts the search to only a few iterations at best and renders the problem intractable at worst. In [3], upon which this paper builds, we overcome this difficulty by augmenting the NS with a *surrogate model*.

Combining an evolutionary algorithm (EA) with a surrogate modelling function has been common in the literature for many years, e.g. in single-objective optimisation [14], multi-objective optimisation [15], and particularly in expensive optimisation [16]. A first surrogate model for neural network optimisation was introduced by Gaier et al. [17] and used in conjunction with the NEAT [18] algorithm for evolving the weights and topology of a neural network. This paper used a surrogate distance-based model, employing a genotypic compatibility distance metric that is part of NEAT. The approach has been quickly adopted in the literature using a range of surrogates and a variety of methods to evolve networks. There are several examples of approaches that use surrogates to estimate the performance of an architecture. For example, in 2017 Deng et al. proposed the Peephole algorithm [19], which predicted the performance of a convolutional neural network based on its architecture information: a long-short term memory (LSTM) neural network was used to train the model. Stork et al. [20] extended a Cartesian Genetic Programming method called CPGANN to evolve neural networks using surrogate-based optimisation to reduce the number of fitness evaluations required. They used a Kriging model [21] as the surrogate. In [22], a Random Forest algorithm (RF) was used as a surrogate to predict the performance of a CNN architecture—the authors proposed a method for describing a CNN as a set of features which were used as input to the RF. In [4], the authors use a surrogate benchmark for neural architecture search (NAS).

In contrast, Hagg et al. [23] introduce a more flexible method for building a surrogate model that is independent of network topology: rather than describing the neural network architecture, they introduce a *phenotypic* metric which measures the difference in output between two neural networks given the same input sequence. The difference is used in a Kriging surrogate model. Our proposed approach, first described in [3], is conceptually closest to that of Hagg. For a given neural network, we calculate a behavioural vector that describes its behaviour on a dataset (see "Diversity Metrics" section). We then propose a RF surrogate model that is used to estimate the distance between the behavioural vectors produced by any two neural networks, as this value is required to drive a NS algorithm.

Dietterich [1] explains how the performance of an ensemble depends crucially on both the individual accuracy of base learners and the diversity between them. Krogh and Vedelsby [24] formalise this by defining the generalisation error of an ensemble as $E = \bar{E} - \bar{A}$, where $\bar{E}$ is a weighted average of the generalisation errors of individual models and $\bar{A}$ is the weighted average of their ambiguities, which expresses their *disagreement*. Therefore, the more accurate and diverse the learners, the more accurate the predictions made by the ensemble. The question is often posed regarding the trade-off between diversity and individual accuracy in ensemble learning. This has been the subject of extensive research.

Chandra et al. [25] present a review of the use of multi-objective evolutionary algorithms to find this trade-off. Zhu et al. [10] propose an artificial resampling method which groups the training set into crossed training sets. They claim that this method provides the best trade-off between diversity and accuracy and show that it outperforms Boosting [26] and Bagging [27] on several classification problems. Gu and Jin [8] propose a multi-objective evolutionary algorithm which maximises accuracy and diversity together. The Pareto-optimal solutions are analysed as trade-offs between diversity and accuracy. Özöğür-Akyüz et al. [9] propose an ensemble pruning method which utilises accuracy and diversity information simultaneously and show that it outperforms alternative approaches. Bhowan et al. [28] employ a multi-objective GP approach to evolve classifier ensembles that are both accurate and diverse in order to tackle the problem of unbalanced data; they refine their approach in [29].

Sheng et al. [30] propose a niching evolutionary algorithm with adaptive negative correlation learning in which the adaptation strategy controls the diversity-accuracy trade-off. Hart and Sim [31] study ensembles of optimisation algorithms, which have otherwise received little attention, and investigate the accuracy-diversity trade-off in that context. They apply their approach to the domain of bin-packing as an example. Tsakonas [32] analyses this trade-off with a multi-objective evolutionary system that combines partially

trained learners, utilising a ranking formula which incorporates both diversity and accuracy. In [33, 34], we proposed an algorithm for building an ensemble using MAP-Elites [35] to maximise both the diversity and the accuracy of its members. Many other approaches explore the balance between diversity and accuracy in ensembles, e.g. [36–40].

Here we extend upon the work presented in [3] by incorporating accuracy objectives into the explicit search for diversity, so as to investigate whether this could lead to a performance gain. A preliminary version of this study was published in [7]. We propose multiple ways of combining diversity and accuracy objectives and study the effect of each of them upon ensemble accuracy.

Novelty Search Augmented with a Surrogate Model

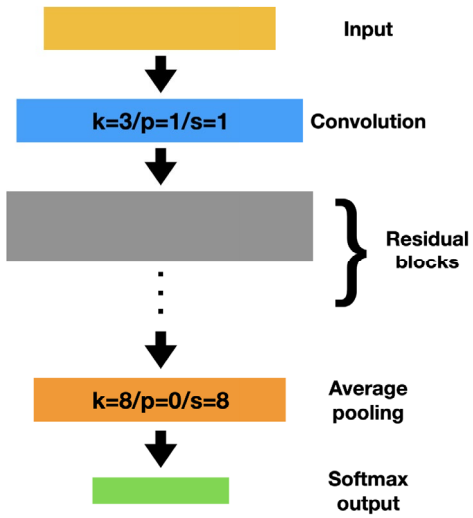
We use NS to evolve an ensemble of behaviourally diverse neural network models, as described in "Novelty Search Algorithm" section. NS operates over a space of architectures defined by a set of hyperparameters. It starts with a random population of neural network architectures and iteratively evolves a set of models, with the search being driven by novelty scores. Novelty is defined by a set of diversity metrics, as explained in "Diversity Metrics" section. Unlike the first version of this procedure [2], where the neural networks in a generation were trained with gradient descent at each iteration of the NS in order to calculate the behavioural distances between each pair of architectures—needed to calculate novelty scores—these distances are now *estimated* by a surrogate model which is pretrained on a sample drawn from the space of neural network architectures. The most diverse models are added to the final ensemble, which is then trained on the input data. We evaluate the method against our previous method and compare the performance obtained with different diversity metrics. The following subsections go into detail about each of these steps.

Neural Network Architectures

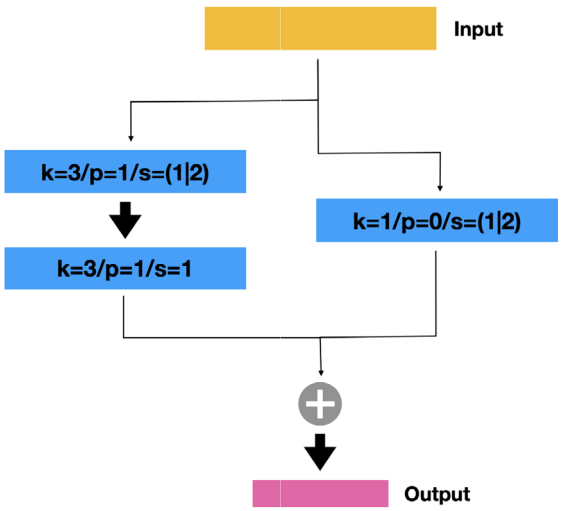
The architectures evolved by our procedure are *residual neural networks* [41] based on the wide architectures proposed by [6]. They are of the same kind as those we used in previous work [2]. Figure 2a shows a generic neural network and Fig. 2b illustrates a generic residual block. Please refer to [2] for a more detailed description of these architectures.

The *hyperparameters* of each network are evolved by NS. Each individual in the population is defined by a variable-length vector, depending on the number of blocks r: $[J, C, O^1, \ldots, O^r, D^1, \ldots, D^r]$, where J is a Boolean value indicating whether the network should be trained jointly or separately if it is in the final ensemble, C is the output size of the first convolution, O^i is the output size of block i, and D^i its dropout probability. Each individual is mapped to a Pytorch module [42] for implementation purposes. The *parameters* of each network are randomly initialised and then optimised by a standard gradient descent procedure.

In order to preprocess the input to the surrogate model, we *normalise* the representation described above in the



(a) Generic residual neural network (k = kernel size; p = padding; s = stride).



(b) Generic residual block.

Fig. 2 Generic topology of individual neural networks

following way: we first *rescale* the elements in all positions so that they lie between 0 and 1. This is necessary for regularising training. The first element is the Boolean value indicating whether the neural network should be trained jointly or separately, so it need not be normalised. Then, given that the representations have variable length depending on the number of residual blocks in each neural network, we *pad* the vector so that it has fixed length, corresponding to the maximum possible number of residual blocks, by adding an appropriate number of elements equal to 0 before the sequence of block output sizes and before the sequence of dropout probabilities. Therefore, if the number of residual blocks in the network is r and the maximum number of blocks is R; if the maximum and minimum sizes of the first convolution in the network are $C_{\max}$ and $C_{\min}$, respectively; if the maximum and minimum sizes of each residual block are $O_{\max}$ and $O_{\min}$, respectively; and if the maximum and minimum dropout probability of each block are $D_{\max}$ and $D_{\min}$; then the normalised representation of neural network m_i is:

$$\begin{aligned}\textbf{norm}\_\text{rep}_i = \Bigg[& J_i, \frac{C_i - C_{\min}}{C_{\max} - C_{\min}}, \\ & 0, \ldots, 0, \frac{O_i^{R-r} - O_{\min}}{O_{\max} - O_{\min}}, \ldots, \frac{O_i^{R} - O_{\min}}{O_{\max} - O_{\min}}, \\ & 0, \ldots, 0, \frac{D_i^{R-r} - D_{\min}}{D_{\max} - D_{\min}}, \ldots, \frac{D_i^{R} - D_{\min}}{D_{\max} - D_{\min}} \Bigg]\end{aligned} \tag{1}$$

where there are $R - r$ elements equal to 0 before the sequence of block output sizes and before the sequence of dropout probabilities.

Diversity Metrics

In order to calculate novelty scores, which are used as the objective function by the NS, we have considered six different diversity metrics, five of which we have defined ourselves. This has enabled us to observe how final performance is affected by the choice of diversity metric. These metrics are calculated between each pair of individual neural network architectures. We have used three of these metrics in the first version of our procedure [2], whereas the remaining ones were introduced in [3].

Let $\boldsymbol{y}_i$ be the vector of predictions for model m_i with each prediction y_i^n for data point $\boldsymbol{x}^n$ being a class label in $\{1 \ldots C\}$. Let $\boldsymbol{p}_i$ be a binary vector where $p_i^n = 1$ if the prediction y_i^n is correct and $p_i^n = 0$ otherwise. Let N^{11}, N^{00}, N^{01}, and N^{10}, respectively, be the total number of test instances where two models are both correct, both incorrect, and when one is correct and the other is not. The first diversity metric we consider is the *proportion of different errors* between two models when at least one of them is *correct*. We expect it to provide insight into the divergence between the errors made by two models. It is defined as:

$$\text{prop}_{i,j}^1 = \frac{N^{01} + N^{10}}{N^{11} + N^{01} + N^{10}} \tag{2}$$

The second diversity metric we consider is very similar and is the *proportion of different errors* between two models when at least one of them is *incorrect*. We have defined it as:

$$\text{prop}_{i,j}^2 = \frac{N^{01} + N^{10}}{N^{00} + N^{01} + N^{10}} \tag{3}$$

The third metric we propose is the *harmonic mean* between these two proportion metrics. This is a sound way of averaging the two proportion metrics into a single metric so that they are both taken into account. It is defined as:

$$\text{prop}_{i,j}^{\text{harm}} = \frac{2 \cdot \text{prop}_{i,j}^1 \cdot \text{prop}_{i,j}^2}{\text{prop}_{i,j}^1 + \text{prop}_{i,j}^2} \tag{4}$$

We also consider a widely used metric (e.g. [43–45]) defined as the *disagreement* between two models, i.e. the proportion of test instances where one is correct and the other is not. We take this metric into account since it expresses how commonly two models disagree on any test instance. It is defined as:

$$\text{dis}_{i,j} = \frac{N^{01} + N^{10}}{N^{00} + N^{01} + N^{10} + N^{11}} \tag{5}$$

Consider now the *two's complement* of the binary vector of correct predictions $\boldsymbol{p}_i$, $\boldsymbol{w}_i$, i.e. the binary vector of *wrong* predictions. The next metric we propose is the *cosine distance* between the binary vectors of wrong predictions made by two models m_i and m_j. Like $\text{prop}_{i,j}^1$ and $\text{prop}_{i,j}^2$, we consider this metric because it is a measure of the distance between the errors made by two models. We have defined it as:

$$\text{cos\_dist}_{i,j} = 1 - \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|} \tag{6}$$

Lastly, we consider a metric of *architectural* diversity. Take the *normalised* vector which represents each individual neural network, as described in "Neural Network Architectures" section. Let its size be L. To obtain an architectural representation, we simply remove the first element from the normalised representation, i.e. the Boolean value indicating whether or not the neural network should be trained separately or jointly. Thus, referring to Eq. 1, the architectural representation of model m_i is:

$$\textbf{arch}\_\text{rep}_i = \textbf{norm}\_\text{rep}_i^{\{1 \ldots L-1\}} \tag{7}$$

We then define *architectural distance* between neural networks m_i and m_j as the cosine distance between their normalised architectural representations:

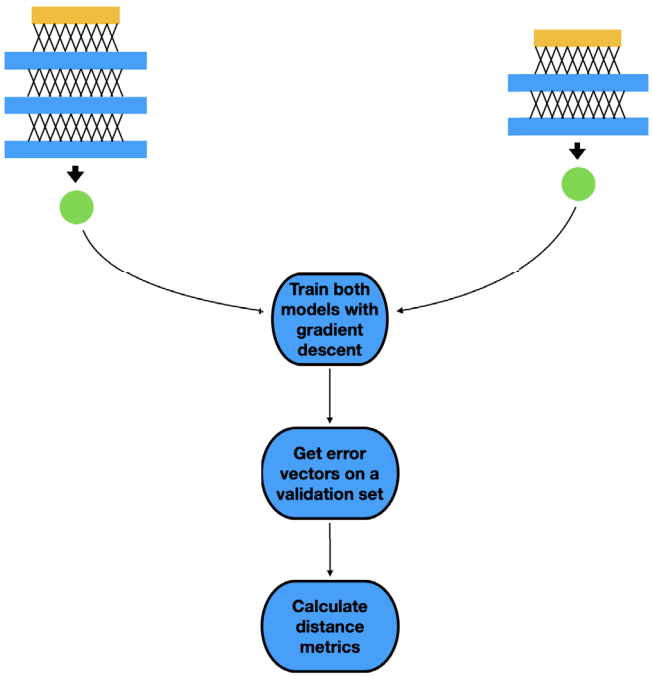
$$\text{arch\_dist}_{i,j} = 1 - \frac{\mathbf{arch}\_\text{rep}_i \cdot \mathbf{arch}\_\text{rep}_j}{\|\mathbf{arch}\_\text{rep}_i\| \|\mathbf{arch}\_\text{rep}_j\|} \quad (8)$$

These metrics determine the *behavioural distance* between two neural network models, which is used to calculate the novelty scores that guide the NS procedure, as explained in "Novelty Search Algorithm" section. Note that the metrics $\text{prop}^2_{i,j}$ and $\text{cos\_dist}_{i,j}$ focus more closely on the instances where the models made a *prediction error*. In our first work [2], these two metrics have led to better performance than the others. We have observed the same pattern with our improved version of the NS procedure [3].

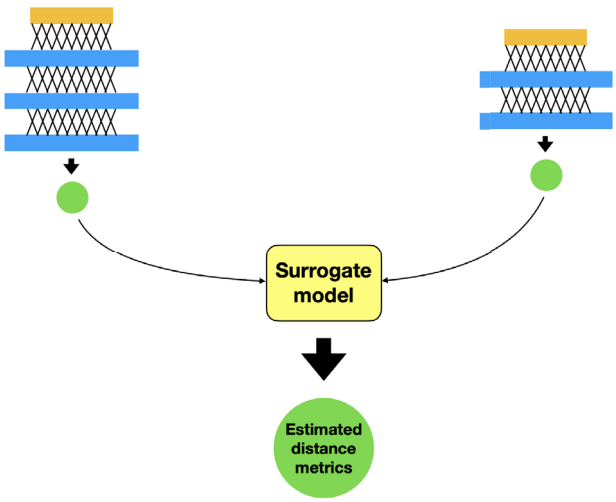
Surrogate Model to Estimate Distances

The NS requires novelty scores to be determined, which in turn require the distances between pairs of neural networks in the current population to be calculated. However, calculating the exact distance values between two neural network models entails first training the models on the input data with gradient descent and then evaluating them on a validation dataset, as we did in previous work [2]. This can be a very costly step if computational resources are limited, which constrains the NS to only a few iterations and the population to a small size—as the neural networks have to be trained in parallel for efficiency. Here we overcome this limitation by pretraining a Random Forest [46] surrogate model which estimates the behavioural distances between a pair of neural network models.

Note that the estimates of behavioural distances produced by the surrogate model do not need to be very accurate. This is because, when calculating the novelty score of a particular individual neural network, we only need to know relative distances in order to determine nearest neighbours. This means that the surrogate model need only capture the general trends of growth of the distance values, even if the actual values are not very precise. This makes the use of a surrogate model very appropriate with no need for a very complex model. Figure 3 shows the differences between the previous method for calculating exact distance values, shown in Fig. 3a, and the current method using a surrogate model, shown in Fig. 3b. Calculating exact distance values is a very costly step, potentially requiring several GPU hours depending on the length of training. In contrast, estimating these distances by means of the surrogate model is an instantaneous process, once the surrogate model has been trained on sample data beforehand.



(a) Steps required to calculate exact distance values between two neural networks.



(b) Estimating the distance values between two neural networks with a surrogate model.

Fig. 3 Difference between calculating and estimating distance values

Pretraining the Surrogate Model

The surrogate model must be trained beforehand so that it can be used effectively during the NS to estimate the distance values between two neural network models. To do this, we draw a sample of neural networks from the search space of architectures defined by the set of hyperparameters used with the NS method. We first train each of these neural networks with gradient descent and calculate their error vectors on a validation data set. We then build random pairs of neural networks and calculate the exact distance values, for all six metrics considered, between them as a function of either their error vectors or their architectural descriptors, as explained in "Diversity Metrics" section. Finally, we construct a data set on which we fit a Random Forest regressor [46] which takes as input the normalised representations of two neural network architectures, as per "Neural Network Architectures" section, and has six outputs: the estimates of the distance values for all six metrics considered. We have selected a Random Forest model due to its low complexity and because we expect it to generalise well on new data, given that it is an ensemble model. Algorithm 1 describes the process of training this surrogate model in pseudocode.

Algorithm 1 Pretraining the surrogate model on sample architectures

draw a sample S of neural networks from the search space defined by J, $[C_{\min}, C_{\max}]$, $[O_{\min}, O_{\max}]$, and $[D_{\min}, D_{\max}]$;
train(S); ▷ Models trained jointly or separately according to the value of J_i
for neural network model $m_i \in S$ **do**
  get error vector e_i on validation set $\mathcal{D}_{\text{val}}$;
end for
build $\frac{\|S\|^2 - \|S\|}{2}$ unique pairs of neural networks;
initialise dataset $\mathcal{D}_{\text{dists}} \leftarrow \emptyset$;
for each pair m_i, m_j **do**
  $\boldsymbol{d} \leftarrow$ all 6 distance values; ▷ calculated as per Section 3.2
  norm_rep$_i$, norm_rep$_j$ are the normalised representations of m_i and m_j;
  add data point $x \leftarrow \{\text{norm\_rep}_i, \text{norm\_rep}_j, \boldsymbol{d}\}$ to $\mathcal{D}_{\text{dists}}$;
end for
train random forest model rf on $\mathcal{D}_{\text{dists}}$;
return random forest model rf;

and one for validation. The training set is used to train the final ensemble; it is also used to train the sample of neural network architectures drawn from the search space that is in turn used to pretrain the surrogate model. Whereas training each of these sample neural networks makes use of the entire training set, pretraining the surrogate model only requires the validation set, which is used to calculate exact distance values between pairs of neural networks.

Selection in NS is driven by the novelty score, which computes the sparseness at any point in the behavioural space. This sparseness is defined by one of the distance metrics of "Diversity Metrics" section. Areas with denser clusters of visited points are considered less novel and therefore rewarded less. This is defined as the average distance to the K-nearest neighbours of a point, calculated with respect to the other individuals in the current generation and to a stored *archive* of previously sampled solutions. Hence, the novelty score is calculated as:

$$NS_i = \frac{1}{k} \sum_{k=0}^{K} \text{div\_metric}(m_i, \mu_k) \tag{9}$$

where μ_k is the kth-nearest neighbour of m_i with respect to the diversity metric div_metric$_{i,j}$, selected from the metrics defined in "Diversity Metrics" section.

Novelty Search Algorithm

Our algorithm for building an ensemble implements NS as described by [5], applying it to our problem domain. Algorithm 2 presents the pseudocode for this procedure. The original training data is split into two sets, one for training

Individuals are selected for reproduction on the basis of their novelty scores using a tournament selection procedure. In the interests of promoting divergence and avoiding convergence, reproduction only uses mutation. Mutation either adds or removes a randomly chosen residual block from an individual, modifying input/output sizes at the

mutation point as necessary; changes the output size and dropout probability of a random block; or swaps two consecutive blocks chosen at random.

After evaluating the entire population, n_A randomly chosen individuals are added to the *archive*, following the method suggested in [47]. In addition, the individual from the population with the highest *elite score*, calculated in a similar fashion to the novelty score, is added to an *elite archive*. After running the NS for the specified number of iterations, a subset of this elite archive is selected as the final ensemble. This subset is chosen so as to maximise the average distance amongst its members. The final ensemble is then trained by gradient descent, the *only time* when this parameter optimisation takes place.

Algorithm 2 Ensemble evolution through NS

randomly initialise population pop;
$archive \leftarrow \emptyset$;
$elite\_archive \leftarrow \emptyset$;
draw $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ from training set $\mathcal{D}$;
set evolution iterations $epochs$;
set archive sample size n_A;
set final ensemble size $ensemble\_size$;
select diversity $\text{div\_metric}_{i,j}$ from Section 3.2;
surrogate model s_{div} pretrained as per Section 3.3;
for $epochs$ **do**
  for $m_i, m_j \in pop \times pop \cup archive : m_i \neq m_j$ **do**
    $\text{div\_metric}_{i,j} \approx s_{\text{div}}(m_i, m_j)$
  end for
  for $m_i, m_j \in pop \times elite\_archive$ **do**
    $\text{div\_metric}'_{i,j} \approx s_{\text{div}}(m_i, m_j)$
  end for
  for $m_i \in pop$ **do**
    $NS_i \leftarrow \frac{1}{k} \sum_{k=0}^{K} \text{div\_metric}(m_i, \mu_k)$ ▷ Equation 9
    $NS'_i \leftarrow \sum_{m_j \in elite\_archive} \text{div\_metric}'(m_i, m_j)$
  end for
  $sample \leftarrow \text{random\_sample}(pop, n_A)$
  $archive \leftarrow archive \cup sample$
  $el\_best \leftarrow \max(pop, NS'_i)$
  $elite\_archive \leftarrow elite\_archive \cup \{el\_best\}$
  $s \leftarrow \text{tournament\_select}(pop, NS_i)$
  $pop \leftarrow \text{mutate}(s)$
end for
for $m_i, m_j \in elite\_archive \times elite\_archive : m_i \neq m_j$ **do**
  $\text{div\_metric}^*_{i,j} \approx s_{div}(m_i, m_j)$
end for
for $m_i \in elite\_archive$ **do**
  $NS^*_i \leftarrow \sum_{m_j \in elite\_archive : m_i \neq m_j} \text{div\_metric}^*(m_i, m_j)$
end for
$ensemble \leftarrow \max(elite\_archive, NS^*_i, ensemble\_size)$
$\text{train}(ensemble)$; ▷ Models trained jointly or separately according to the value of J_i

Evaluation of Evolved Ensembles

In order to evaluate the performance of the evolved ensemble, we use the stacking technique [48], which trains a linear model to weight the predictions of each individual learner. This linear model is trained for a configurable number of iterations on the validation set mentioned in "Novelty Search Algorithm" section. This is to avoid overfitting the test set.

Baseline: Previous Method

The approach we present here is an improvement of the method that we first proposed in [2]. We use this as a baseline against which we compare the new method. In its main

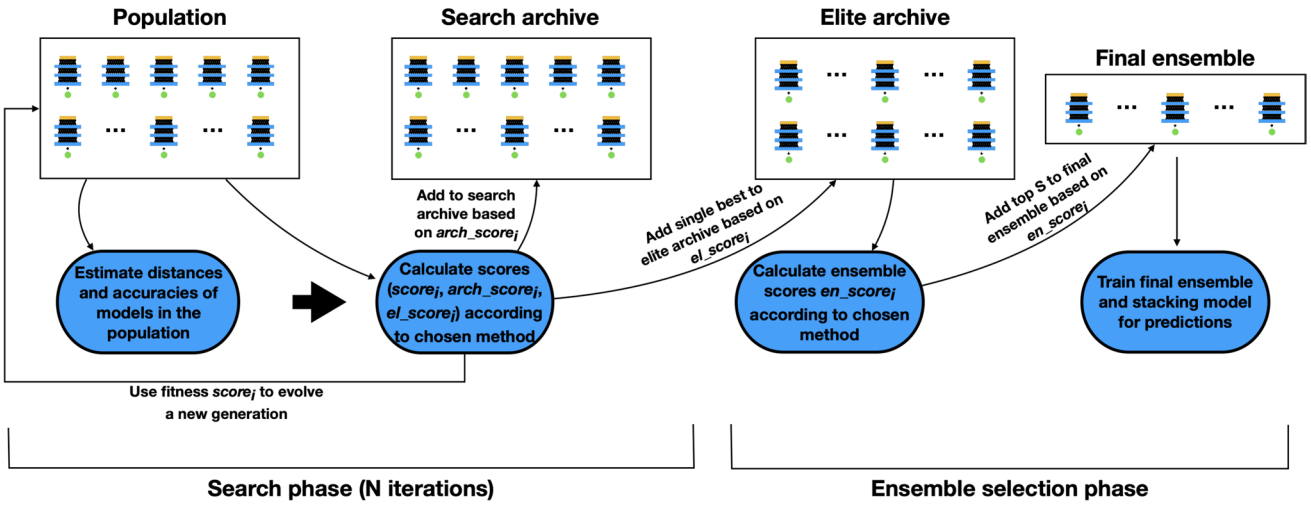


Fig. 4 A high-level view of the two phases of a generic search method for constructing a classifier ensemble. First, a large elite archive is created as a result of a population-based evolutionary search. Then a subset of the elite archive is selected to form an ensemble

aspects, the previous method is similar to the new method, with the notable difference that it does not make use of a surrogate model to estimate the distance between two neural network models. As discussed previously, this original method calculates exact distances between each pair of neural networks by first training all the models in the current generation with gradient descent and then getting their error vectors by evaluating them on a validation set. As an additional difference, the previous method would calculate at each iteration an *ensemble selection metric* for each member of the population and then add to the final ensemble the single best-scoring neural network in each generation. The new method maintains an *elite archive*, to which a sample of neural networks from each generation with highest novelty score with respect to this archive, which we call *elite score*, is added at each iteration; novelty scores with respect to the final elite archive are calculated at the end for each of its members and this *ensemble score* is used to select a subset of neural networks which will make up the final ensemble.

We compare the new method, first proposed in [3], with our previous approach along two main lines. Firstly, we seek to understand whether there is a speedup with the new method as a result of increased efficiency when looking for solutions of similar complexity to those found with the previous method. Secondly, we investigate whether the new method can be used to produce better solutions, i.e. solutions of higher complexity and leading to better final performance. This means that we are interested in investigating whether there is both a quantitative improvement, i.e. being able to do *more of* what could be done with the previous method thanks to a more efficient use of computational resources, and a qualitative improvement, i.e. being able to do *more than* what could be done with the previous method by tackling solutions that were previously unfeasible or intractable.

Generic Search Method with Accuracy Objectives

In [7], we extend the NS procedure that we originally proposed in [3], which constructs an ensemble of behaviourally diverse neural networks by evolving their architectures. We augment this method with accuracy objectives and make use of the same technique which employs a surrogate model for estimating novelty scores, as this reduces the computational burden of the *explicit* search for diversity. In addition, to preserve this greater efficiency, we deploy another surrogate model for *estimating* what the accuracy of a neural network architecture will be if it is trained with gradient descent on the input data. This surrogate accuracy model is pretrained on a sample of architectures drawn from the search space. We then propose multiple ways to incorporate explicit accuracy objectives into the NS.

Conceptual Model

Figure 4 shows a high-level view of a generic population-based search method for constructing ensembles of classifiers. This method has two phases: a search phase and an ensemble selection phase. During the search phase, a large set of candidate classifiers is created. Firstly, the distance between each pair of neural network models in the population, and between each member of the population and each member of the *search archive*, is estimated by a *surrogate diversity* model. The accuracy of each neural

network architecture is also estimated by a *surrogate accuracy* model. These distance and accuracy estimates are then used to calculate three scores: (1) a fitness $score_i$, which is used to evolve a new generation at each iteration to replace the current population; (2) an archive score $arch\_score_i$, which is used to select a sample of models to be added to the search archive at each iteration; and (3) an elite score el_score_i, which determines a single neural network to be added to an *elite archive* at each iteration. The way these scores are calculated is determined by the particular method from "Novelty Search Extended with Accuracy Objectives" section with which this generic search procedure is instantiated.

In the ensemble selection phase, a subset of the neural networks in the elite archive is selected to become the final ensemble. These are the top S networks which score the highest ensemble score en_score_i. Again, the way this score is calculated depends on the particular search method. The methods we propose in this paper represent different combinations of diversity and accuracy objectives along the two phases of the generic search algorithm. Those different combinations are reflected in the four scores mentioned above. As the last step, the final ensemble is trained on a training set with stochastic gradient descent (SGD) and a linear stacking model [48] is trained on a validation set to weight the predictions of each individual learner.

Surrogate Models for Estimating Distance and Accuracy

In [3], we employed a surrogate model that produces estimates of the values of the diversity metrics of "Diversity Metrics" section between two neural network architectures, as described in "Surrogate Model to Estimate Distances" section. This reduces the computational burden of the search, as, in order to calculate exact values for these diversity metrics, our original NS procedure in [2] required exact error vectors to be determined by evaluating the architectures on a validation set. This involved a costly step of training all the neural networks in the population with gradient descent at every iteration. Following this approach, for each dataset we pretrain a Random Forest regressor [46] to be the *surrogate diversity* model.

In addition, we also pretrain a *surrogate accuracy* model for each of the three datasets considered. This is so that accuracy objectives can be incorporated in the search for an ensemble of neural network architectures without having to train these architectures at every step of the procedure and calculate exact accuracy values on a validation set. Instead, estimates are produced for the expected accuracy given the normalised representation of each neural network individual, described in "Neural Network Architectures" section. The surrogate accuracy model is pretrained by first drawing a sample of 3200 neural networks—so as to be large and diverse enough—from the search space of architectures, training them on the input data, and calculating their exact accuracy values on a validation set. We then construct a dataset on which we fit a Random Forest regressor. This model takes as input the normalised representation of a neural network architecture, as per "Neural Network Architectures" section, and outputs its estimated accuracy. Although a different regression model could have been used, the reason for selecting a Random Forest model lies in its low complexity and expected good generalisation (due to low variance in predictions), as it is an ensemble model. Algorithm 3 provides a pseudocode description of the procedures for training both the surrogate diversity and the surrogate accuracy models.

Algorithm 3 Pretraining the surrogate accuracy model on sample architectures

draw a sample S of neural networks from the search space defined by J, $[C_{\min}, C_{\max}]$, $[O_{\min}, O_{\max}]$, and $[D_{\min}, D_{\max}]$;
train(S); ▷ Models trained jointly or separately according to the value of J_i
for neural network model $m_i \in S$ **do**
    calculate accuracy a_i on validation set $\mathcal{D}_{\text{val}}$;
end for
initialise dataset $\mathcal{D}_{\text{accuracies}} \leftarrow \emptyset$;
for each model m_i **do**
    norm_rep$_i$ is the normalised representation of m_i;
    add data point $x \leftarrow \{\text{norm\_rep}_i, a_i\}$ to $\mathcal{D}_{\text{accuracies}}$;
end for
train random forest model rf on $\mathcal{D}_{\text{accuracies}}$;
return random forest model rf;

Novelty Search Extended with Accuracy Objectives

In this paper, we propose a number of *alternative* methods which extend the NS method of [3] by combining diversity and accuracy objectives in different ways across the two phases outlined in Fig. 4. We refer to the generic search method described in "Conceptual Model" section, which is *instantiated* according to each of the selected search methods proposed herein and described below, namely by calculating *in different alternative ways* the scores described before: the fitness $score_i$, the archive score $arch\_score_i$, and the elite score el_score_i, all three during the *search phase*; and the ensemble score en_score_i, during the *ensemble selection phase*. These scores affect the way individuals are selected for both the search and the elite archives, whose purpose is described in "Conceptual Model" section, and to the final ensemble. A pseudocode description of this generic search method is given in Algorithm 4.

Algorithm 4 Generic search method for constructing a classifier ensemble

randomly initialise population pop;
$search\_archive \leftarrow \emptyset$;
$elite\_archive \leftarrow \emptyset$;
draw $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ from training set $\mathcal{D}$;
set evolution iterations $epochs$;
set search archive sample size S_A
set final ensemble size S;
surrogate diversity model s_{div} pretrained as per Algorithm 1;
surrogate accuracy model s_{acc} pretrained as per Algorithm 3;
select diversity $\text{div\_metric}_{i,j}$ from Section 3.2;
$score_i$ is the *fitness score* of model m_i;
$arch\_score_i$ is the *archive score* of model m_i;
el_score_i is the *elite score* of model m_i;
en_score_i is the *ensemble score* of candidate model m_i;
for $epochs$ **do**
  for $m_i, m_j \in pop \times pop \cup search\_archive : m_i \neq m_j$ **do**
    $\text{div\_metric}_{i,j} \approx s_{\text{div}}(m_i, m_j)$
  end for
  for $m_i \in pop$ **do**
    $acc_i \approx s_{\text{acc}}(m_i)$
  end for
  calculate $score_i$, $arch\_score_i$, and el_score_i according to the selected search method;
  $sa\_sample \leftarrow \text{sample}(pop, arch\_score_i, S_A)$
  $search\_archive \leftarrow search\_archive \cup sa\_sample$
  $el\_best \leftarrow \max(pop, el\_score_i)$
  $elite\_archive \leftarrow elite\_archive \cup \{el\_best\}$
  $s \leftarrow \text{tournament\_select}(pop, score_i)$
  $pop \leftarrow \text{mutate}(s)$
end for
for $m_i \in elite\_archive$ **do**
  calculate en_score_i according to the selected search method;
end for
$ensemble \leftarrow \max(elite\_archive, en\_score_i, S)$
$\text{train}(ensemble)$;

Combining Method 1 (CM1): Local Competition

Local Competition (LC) [49] extends NS [5] by adding fitness objectives, which in our case are explicit accuracy objectives. It weights diversity and accuracy according to a parameter α. We expect variations of this parameter to produce different results. The distance between two models, div_metric(m_i, m_k), and the accuracy acc_i of model m_i are *estimated* by surrogate models as described before. In addition to the novelty score NS_i, a local competition score LC_i is calculated as the proportion of neighbours that a model outperforms:

$$NS_i = \frac{1}{k}\sum_{k=0}^{K} \text{div\_metric}(m_i, m_k) \tag{10}$$

$$LC_i = \frac{1}{k}\sum_{k=0}^{K} c(m_i, m_k) \tag{11}$$

For all K neighbours m_k of m_i. The diversity metric $\text{div\_metric}_{i,j}$ is selected from the metrics defined in "Diversity Metrics" section. Note that, while the paper which originally proposed Novelty Search with Local Competition (NSLC) [49] defines LC_i as a count, we define it as a *proportion* calculated w.r.t. the number of neighbours of m_i. This is to ensure that a single score can be appropriately calculated that mixes both LC_i and the novelty score NS_i, which will be of the same order of magnitude due to the fact that distances are scaled to lie between 0 and 1 when pretraining the surrogate diversity model. $c(m_i, m_k)$ is defined thus:

$$c(m_i, m_k) = \begin{cases} 1 & \text{if } acc_i > acc_k \\ 0 & \text{if otherwise} \end{cases} \tag{12}$$

The *fitness score* of model m_i is then calculated by mixing NS_i and LC_i according to the mixing parameter α:

$$score_i = (1-\alpha) \times NS_i + \alpha \times LC_i. \tag{13}$$

No archive score $arch\_score_i$ is calculated for LC since a *random sample*, of size S_A, of the individuals in the current population is added to the search archive, as in [3]. Consider now a novelty score for model m_i calculated w.r.t. *all the individuals in the elite archive*, NS_i^*. Consider the equivalent local competition score, LC_i^*. The elite score el_score_i is calculated in a similar fashion to $score_i$, with the same parameter α but using these two scores instead. The individual in each generation with the highest el_score_i is added to the elite archive. At the end of the procedure, an ensemble score en_score_i is calculated in a very similar way for all the neural network models in the elite archive—w.r.t. all other individuals in this archive. The top S individuals with the highest ensemble scores will make up the final ensemble.

Combining Method 2 (CM2): Search for Diversity with Accuracy in Archives

This variant uses a novelty score to guide the search procedure, namely the selection of individuals from the population for reproduction, while storing the neural network models in the archives, including the search archive, according to their accuracy. Thus, we expect this method to maintain diverse populations whilst selecting the most accurate models in an elitist fashion. The scores are defined in the following way:

$$score_i = NS_i \tag{14}$$

$$arch\_score_i = el\_score_i = en\_score_i = acc_i. \tag{15}$$

While the single individual with highest el_score_i is added to the elite archive at every step of the search, the top S_A individuals with highest $arch\_score_i$ are added to the search archive.

Table 1 Novelty search parameters for both test sets

Parameter	Test set 1: runtime comparison (both methods)	Test set 2: expanded search space (new method only)
Iterations	10	100
Final ensemble size	11	40
Population size	30	100
Diversity metric	$\text{cos\_dist}_{i,j}$	All from "Diversity Metrics" section
Number of blocks	2:6	2:6
Number of channels in the first convolution	4:16	4:16
Number of channels in residual blocks	24:32	16:64
Dropout probability in residual blocks	0.1:0.4	0.1:0.9
Number of neighbours K	3	15
Size n_A of archive sample	5	10
Size of tournament for selection	10	50

Combining Method 3 (CM3): Search for Accuracy with Diversity in Archives

This variant does the opposite of the previous combining method, i.e. it uses accuracy to guide the search procedure, but stores individuals in the elite archive based on how novel/diverse they are. The final ensemble is also selected based on novelty. We thus expect it to maintain accurate populations and the final ensemble to be a set of diverse models selected from a high-performing region of the search space. Recall that NS_i^* is the novelty score calculated for model m_i in the current generation w.r.t. all individuals in the elite archive. Let NS_i^{**} be a similar novelty score calculated for model m_i in the elite archive w.r.t. all other models in this archive. The scores are then defined as:

$$score_i = acc_i \tag{16}$$

$$el\_score_i = NS_i^* \tag{17}$$

$$en\_score_i = NS_i^{**} \tag{18}$$

This method does not keep a search archive as the search is guided by accuracy only and, therefore, there is no need to keep an archive of past solutions with respect to which a novelty score is calculated.

Explicit Accuracy with Implicit Diversity

The last method we consider uses only an implicit definition of diversity. The search is guided by accuracy and individuals are stored in the elite archive and selected for the final ensemble also based exclusively on their estimated accuracy. Diversity is generated implicitly by the evolutionary procedure, namely the mutation operator applied in the reproduction step. We expect this method to produce accurate but not very diverse ensembles. The scores are then determined as:

$$score_i = el\_score_i = en\_score_i = acc_i \tag{19}$$

As in the case of the previous method, no search archive is kept, since the purpose of such an archive is for novelty scores to be calculated w.r.t. past solutions.

Experiments

This section describes three sets of experiments. The first two compare the new NS method of [3], which makes use of a surrogate model to estimate the distance between models as described previously, with the previous method of [2], which instead calculates exact distance values by first training all the models in the population with gradient descent and then determining their error vectors on a validation set. We compare the methods based on resource usage, namely runtime, for similar parameter settings and model complexity, as well as on their ability to scale to larger search spaces and search for more complex models. The last test set concerns the generic search method of [7], which extends the NS with accuracy objectives. We instantiate this generic method with the four methods of "Novelty Search Extended with Accuracy Objectives" section and compare their resulting performance.

Test Set 1: Resource Usage for Similar Complexity

In this set of tests, we investigate the total time required to run each of the two methods when they are looking for *solutions of the same complexity* and running for the same number of iterations. We wish to determine the speedup that can be gained with the new method, which makes use of a surrogate model to overcome the need for training all the models in the current generation with gradient descent in order to calculate novelty scores. We run both the new and the previous methods on CIFAR-10 and fix the parameters, as shown in the second column of Table 1. We conjecture that in these conditions our new method not only results in a speedup due to the use of a Random Forest surrogate model, but also outputs ensembles of similar performance. This is expressed by Hypotheses 1 and 2.

Hypothesis 1 (Runtime of previous NS method vs new method enhanced with a surrogate model) *Enhancing the NS procedure with a Random Forest surrogate model pretrained to estimate the distance between models, and thereby their novelty scores, results in a speedup compared with our previous method, which calculates exact distance values and novelty scores, when constructing ensembles of the same complexity.*

Hypothesis 2 (Performance achieved with the previous NS method vs the new method with a surrogate model) *When looking for solutions of the same complexity, the new NS procedure, using a surrogate model, outputs ensembles which do not perform worse than those constructed by our previous method, even though the new method only estimates distance values and novelty scores.*

Test set 2: Expanding the Search Space

Using a surrogate model to speed up the procedure has enabled us to both search for solutions of higher complexity

and run the NS for longer. In this set of experiments, we apply the new method to three benchmark datasets from the Computer Vision (CV) literature—CIFAR-10, CIFAR-100, and SVHN—and test it with all diversity metrics previously defined in "Diversity Metrics" section. We also compare the results achieved with the new method to the best results observed with the previous method. The parameters that we use with the new method are shown in the third column of Table 1; they correspond to the *expanded* search space made possible by the use of a surrogate model. We expect to see further evidence of what we observed in previous work [2] regarding *error diversity* metrics, namely that those diversity metrics which focus more closely on the instances where the models make prediction errors lead to higher-performing ensembles. This is expressed by Hypothesis 3. We also expect the new method to lead to higher-performing ensembles than those constructed with the previous method, since the use of a surrogate model makes it feasible to expand the search space and run the NS for longer. This is expressed by Hypothesis 4.

Hypothesis 3 (Better performance with metrics that focus on error instances) *In a similar fashion to what we have observed with our previous method, running the NS procedure with the distance metrics that focus more closely on the instances where the models make prediction errors leads to higher-performing ensembles than when more generic diversity metrics are employed.*

Hypothesis 4 (Performance achieved with the previous NS method vs the new method with a surrogate model) *The new NS method enhanced with a surrogate model makes it possible to search a larger space of more complex neural network architectures and, therefore, outputs higher-performing ensembles than the best ones constructed by our previous method.*

Test Set 3: Introducing Accuracy Objectives

Here we describe the experiments carried out on three datasets—CIFAR-10, CIFAR-100, and SVHN. We compare the results reported for the NS method of [3] with the four modified methods that we present in "Novelty Search Extended with Accuracy Objectives" section. Running each of these methods to construct an ensemble whose performance is then evaluated requires four steps: (1) running the modified NS procedure using the surrogate diversity and surrogate accuracy models, *without training* the neural network architectures during the search; (2) training the ensemble of neural network architectures resulting from the previous step on the training set $\mathcal{D}_{train}$, using a standard stochastic gradient descent (SGD) procedure; (3) training a stacking model [48] on the validation set $\mathcal{D}_{val}$, so as to learn a weighted average of the predictions made by each member of the ensemble; and (4) calculating the classification accuracy of the ensemble on a test set. The surrogate diversity and surrogate accuracy models are pretrained as described before. Table 2 shows the parameters used throughout the experiments. Each experiment, i.e. each sequence of the steps (1)-(4) described above, is run 10 times in order to ensure statistical significance in our observations. The following four hypotheses are tested as part of our empirical analysis.

Table 2 Common parameters fixed throughout the experiments for all the NS methods extended with accuracy objectives

Parameter	Value
Iterations	100
Final ensemble size S	40
Population size	100
Number of residual blocks	2:6
Number of channels in the first convolution	4:16
Number of channels in residual blocks	16:64
Dropout probability in residual blocks	0.1:0.9
Number of neighbours K	15
Size n_A of archive sample	10
Size of tournament for selection	50

Hypothesis 5 (Performance Gain by Adding Accuracy Objectives) *Taking individual model accuracy into account as an objective, by means of the methods presented in "Novelty Search Extended with Accuracy Objectives" section, leads to better ensemble accuracy than what can be achieved with a plain NS method.*

This hypothesis expresses the expectation that accuracy objectives can improve the results of a plain search for explicit diversity alone. We test it by comparing the results achieved by the NS method of [3] with the final ensemble accuracy resulting from the three methods from "Novelty Search Extended with Accuracy Objectives" section which combine diversity and accuracy objectives: local competition, search for diversity with accuracy in the archives, and search for accuracy with diversity in the archives.

Hypothesis 6 (Different Performance with Different Combinations of Diversity and Accuracy) *Selecting different combinations of diversity and individual model accuracy, along the spectrum that ranges from favouring only diversity to favouring only accuracy, results in different ensemble accuracy.*

This hypothesis expresses the notion that multiple ways of mixing diversity and accuracy objectives lead to

different ensemble accuracy and that, therefore, an optimal middle ground between searching only for one or the other can be found. We test it by comparing the ensemble performance resulting from varying the mixing weight α when deploying the LC approach (see Eq. 13), as well as from the other methods of "Novelty Search Extended with Accuracy Objectives" section.

Hypothesis 7 (Diversity and Accuracy Must Be Balanced) *Assigning greater importance to diversity in an ensemble leads to worse individual model accuracy. Conversely, weighting individual accuracy more leads to less diverse ensembles. There is a trade-off to be found between the two.*

This hypothesis claims that there is a fundamental tension between diversity and individual model accuracy and that one can only be improved at the expense of the other. We test it by calculating the values of diversity metrics and average individual model accuracy for the ensembles resulting from applying both the previous NS method and the various methods of combining diversity and accuracy objectives.

Hypothesis 8 (Worse Performance Without Explicit Diversity) *Removing explicit diversity objectives, keeping only individual accuracy objectives when searching for an ensemble, leads to a decrease in ensemble diversity and, consequently, worse ensemble accuracy.*

This hypothesis expresses the importance of explicit diversity objectives for constructing a high-performing classifier ensemble. This follows from the results of Cardoso et al. [2, 3]. We test it by comparing the performance resulting from the last method presented in "Novelty Search Extended with Accuracy Objectives" section, which considers only accuracy objectives with diversity being generated implicitly, with that resulting from the previous NS approach and the methods which combine both diversity and accuracy objectives.

Results for the Novelty Search Augmented with a Surrogate Distance Model

In this section, we present the results of the first two sets of experiments described in "Experiments" section. We then discuss these results and whether the hypotheses formulated above can be rejected.

Hypothesis 1

Hypothesis (Runtime of previous NS method vs new method enhanced with a surrogate model) *Enhancing the NS procedure with a Random Forest surrogate model pretrained to estimate the distance between models, and thereby their novelty scores, results in a speedup compared with our previous method, which calculates exact distance values and novelty scores, when constructing ensembles of the same complexity.*

Table 3 Median results over 10 runs of the previous NS method and the new NS method with a surrogate model on CIFAR-10 (test set 1). Training a sample of architectures and the random forest surrogate model are one-off costs

Runtime of NS	48,760.5 s
Runtime of NS with surrogate model	4871 s
Training a sample of architectures	28,970.5 s
Building a dataset and training the Random Forest surrogate model	18,113.5 s
Accuracy achieved by NS	82.245%
Accuracy achieved by NS with surrogate model	83.885%

Table 3 shows the median value, calculated after 10 independent runs, of the time required to run both the previous NS method [2] and the new method [3], which makes use of a surrogate model, with the same parameters. These results show that the new method is about 10 times faster than the original NS method. A Mann-Whitney significance test shows that this difference is significant at the 1% level. This supports the claim of Hypothesis 1 that enhancing the NS method with a Random Forest surrogate model to estimate the distances between models speeds up the search for diverse models and the construction of a diverse ensemble. For reference, we also report in Table 3 the median time, over 10 runs, required to train a sample of 40 neural network architectures on CIFAR-10, as well as to build a dataset and train the Random Forest surrogate model as per Algorithm 1. Note that these two runtimes are a *one-off cost* and that, in order to pretrain the surrogate model for our experiments, we have trained a total of 3200 sample architectures by running several processes in parallel on a cluster, each training 40 architectures.

Hypothesis 2

Hypothesis (Performance achieved with the previous NS method vs the new method with a surrogate model) *When looking for solutions of the same complexity, the new NS procedure, using a surrogate model, outputs ensembles which do not perform worse than those constructed by our previous method, even though the new method only estimates distance values and novelty scores.*

Table 3 also shows the median accuracy, calculated after 10 independent runs, achieved by ensembles constructed by both the previous NS method and the new method, when these are executed with the same parameters. The results show that the ensembles constructed by the new method do not perform worse than those constructed by the original method, which calculates exact values for the distance metrics and novelty scores. In fact, we observe that the new method leads to slightly better performance. A Mann–Whitney significance test shows that this difference is significant at the 1% level. This corroborates Hypothesis 2, which claims that there is no loss in performance when using the new method and its surrogate estimates. Besides the use of surrogate models, the major difference between the previous and the new method is the way a subset of all the models is selected to be in the final ensemble. As explained before, the previous method applies an *ensemble selection metric* at each iteration of the NS, whereas the new method keeps an *elite archive*, from which the final ensemble is selected in an additional step at the end of the procedure. It seems that the ensemble selection procedure of the new method is the cause behind the better performance achieved by its ensembles.

Hypothesis 3

Hypothesis (Better performance with metrics that focus on error instances) *In a similar fashion to what we have observed with our previous method, running the NS procedure with the distance metrics that focus more closely on the instances where the models make prediction errors leads to higher-performing ensembles than when more generic diversity metrics are employed.*

Table 4 shows the median accuracy, after 10 runs, of ensembles evolved by the new NS procedure extended with a surrogate model, for all six diversity metrics of "Diversity Metrics" section and all three datasets considered. We observe that on CIFAR-10 and SVHN, the metrics $\text{prop}^2_{i,j}$ and $\text{cos\_dist}_{i,j}$ lead to the highest-performing ensembles. Mann–Whitney tests show that the difference to the other metrics is statistically significant. On CIFAR-100, this is observed additionally with the metrics $\text{prop}^{\text{harm}}_{i,j}$ and $\text{dis}_{i,j}$.

The metrics $\text{prop}^2_{i,j}$ and $\text{cos\_dist}_{i,j}$ are the two that focus more closely on the instances where the two models being compared make prediction errors. Additionally, the metric $\text{prop}^{\text{harm}}_{i,j}$ depends on the value of $\text{prop}^2_{i,j}$. These observations back the claim of Hypothesis 3 that error diversity metrics lead to better-performing ensembles compared to more generic diversity metrics. This confirms what we observed in previous work [2].

Table 4 Median accuracy over 10 runs of ensembles constructed by the new method (test set 2). Best results with the original NS are shown for comparison. Results that outperform the original NS at the 1% level are highlighted in bold

Dataset	Diversity metric	Final ensemble accuracy (%)	Best accuracy with original NS (from [2]) (%)
CIFAR-10	$\text{prop}^1_{i,j}$	67.295	83.51
	$\mathbf{prop^2_{i,j}}$	**90.605**	
	$\text{prop}^{\text{harm}}_{i,j}$	83.975	
	$\text{dis}_{i,j}$	86.28	
	$\mathbf{cos\_dist_{i,j}}$	**90.11**	
	$\text{arch\_dist}_{i,j}$	80.4	
CIFAR-100	$\text{prop}^1_{i,j}$	28.725	45.42
	$\mathbf{prop^2_{i,j}}$	**63.05**	
	$\mathbf{prop^{harm}_{i,j}}$	**63.41**	
	$\mathbf{dis_{i,j}}$	**63.18**	
	$\mathbf{cos\_dist_{i,j}}$	**63.035**	
	$\text{arch\_dist}_{i,j}$	49.83	
SVHN	$\text{prop}^1_{i,j}$	78.825	91.435
	$\mathbf{prop^2_{i,j}}$	**94.8**	
	$\text{prop}^{\text{harm}}_{i,j}$	89.775	
	$\text{dis}_{i,j}$	90.675	
	$\mathbf{cos\_dist_{i,j}}$	**94.79**	
	$\text{arch\_dist}_{i,j}$	90.68	

Hypothesis 4

Hypothesis (Performance achieved with the previous NS method vs. the new method with a surrogate model) *The new NS method enhanced with a surrogate model makes it possible to search a larger space of more complex neural network architectures and, therefore, outputs higher-performing ensembles than the best ones constructed by our previous method.*

The last column of Table 4 shows the best performance achieved by ensembles evolved with our previous NS method. These results show very clearly that the new method constructs higher-performing ensembles than our previous procedure, with the most considerable difference being observed on CIFAR-100 and CIFAR-10. Mann–Whitney tests reveal that, for each dataset, the difference between the best results achieved by the new method and the best achieved by the previous method is indeed statistically significant. This difference results from the fact that the new method, thanks to its use of a surrogate model, is able to *search a wider space of neural network architectures*, even though it runs on *the same bounded resources*. We conclude that this supports Hypothesis 4.

Table 5 Median accuracy over 10 runs for the previous NS method and all the methods extending it with accuracy objectives. Results significantly better than NS at the 1% level for the respective diversity metric are highlighted in bold

Method		Diversity Metric	Accuracy CIFAR-10 (%)	Accuracy CIFAR-100 (%)	Accuracy SVHN (%)
NS (from [3])		$prop^1_{i,j}$	67.295	28.725	78.825
		$prop^2_{i,j}$	90.605	63.05	94.8
		$prop^{harm}_{i,j}$	83.975	63.41	89.775
		$dis_{i,j}$	86.28	63.18	90.675
		$cos\_dist_{i,j}$	90.11	63.035	94.79
		$arch\_dist_{i,j}$	80.4	49.83	90.68
CM1: LC	$\alpha = 0.1$	$prop^1_{i,j}$	**80.485**	**34.605**	**89.975**
		$prop^2_{i,j}$	90.655	63.46	94.98
		$prop^{harm}_{i,j}$	86.12	63.935	91.635
		$dis_{i,j}$	84.615	63.63	91.63
		$cos\_dist_{i,j}$	90.26	63.415	94.87
		$arch\_dist_{i,j}$	**87.745**	**53.865**	**93.27**
	$\alpha = 0.5$	$prop^1_{i,j}$	**90.67**	**63.45**	**94.565**
		$prop^2_{i,j}$	90.715	63.18	94.87
		$prop^{harm}_{i,j}$	86	63.755	92.67
		$dis_{i,j}$	86.62	63.99	93.76
		$cos\_dist_{i,j}$	90.005	63.685	94.925
		$arch\_dist_{i,j}$	**88.635**	**58.615**	**94.25**
	$\alpha = 0.9$	$prop^1_{i,j}$	**90.295**	**63.695**	**94.82**
		$prop^2_{i,j}$	90.735	63.47	94.99
		$prop^{harm}_{i,j}$	85.31	63.755	93.78
		$dis_{i,j}$	86.145	63.835	92.23
		$cos\_dist_{i,j}$	89.9	63.265	94.955
		$arch\_dist_{i,j}$	**88.635**	**59.045**	**94.455**
CM2: search for div., acc. in archives		$prop^2_{i,j}$	**90.83**	63.125	94.915
		$cos\_dist_{i,j}$	**90.83**	62.545	94.91
CM3: search for acc., div. in archives		$prop^2_{i,j}$	90.73	**64.16**	94.865
		$cos\_dist_{i,j}$	90.565	63.93	94.89
Explicit accuracy with implicit diversity			90.895	63.755	94.915

Results for the Novelty Search Extended with Accuracy Objectives

Referring now to the last set of experiments ("Test Set 3: Introducing Accuracy Objectives" section), Table 5 shows the results of running each of the methods proposed in "Novelty Search Extended with Accuracy Objectives" section, as well as the NS method of [3] (reported above). This is the mean final ensemble accuracy over 10 runs for each method, parameter setting, and diversity metric, as applicable. As observed above, the metrics that lead to the best results are $prop^2_{i,j}$ and $cos\_dist_{i,j}$. While for LC we run the method with all diversity metrics for a direct comparison with the previous NS approach, for the other two combining methods we confine ourselves to these two metrics in the interests of clarity. The method that only utilises explicit accuracy objectives does not make use of any diversity metric as diversity is generated implicitly.

Hypothesis 5

Hypothesis (Performance Gain by Adding Accuracy Objectives) *Taking individual model accuracy into account as an objective, by means of the methods presented in "Novelty Search Extended with Accuracy Objectives" section, leads to better ensemble accuracy than what can be achieved with a plain NS method.*

Table 5 shows accuracy results both for the NS method of [3] and the four methods we propose herein. The cells highlighted in bold correspond to results which are

significantly better at the 1% level, determined by Mann-Whitney statistical significance tests over 10 runs, than the NS method for the respective diversity metric (not applicable to the last method). We can see that both LC and the other two combining methods of "Novelty Search Extended with Accuracy Objectives" section improve on the NS for some of the metrics considered, but that the only consistent improvement on all three datasets is observed for LC with metrics $\text{prop}^1_{i,j}$ and $\text{arch\_dist}_{i,j}$, which are the ones that tend to perform the worst in the NS. We note that, on CIFAR-10, the second combining method produces statistically significant improvements over NS, but that these improvements are not only too small to be considered relevant, but also inconsistent as they are not observed on CIFAR-100 or on SVHN. A similar inconsistent improvement is observed on CIFAR-100 for the third method, with metric $\text{prop}^2_{i,j}$. These results therefore only partially support Hypothesis 5, since introducing accuracy objectives only improves on the results of the NS for the worst-performing diversity metrics. The hypothesis must be rejected since no significant improvement is observed for the best-performing ones, which suggests that the choice of a good diversity metric plays a more important role and can make a more considerable difference than explicit accuracy objectives.

Hypothesis 6

Hypothesis (Different Performance with Different Combinations of Diversity and Accuracy) *Selecting different combinations of diversity and individual model accuracy, along the spectrum that ranges from favouring only diversity to favouring only accuracy, results in different ensemble accuracy.*

As observed above, introducing accuracy objectives only considerably improves on the NS results for the two worst-performing metrics. To analyse the influence of different combinations of diversity and accuracy objectives, we now focus more closely on the results achieved by LC, the other two combining methods, and the explicit accuracy search method. For LC, we see that there is an improvement for the two worst-performing metrics, $\text{prop}^1_{i,j}$ and $\text{arch\_dist}_{i,j}$, when increasing α from 0.1 to 0.5 or 0.9. Mann–Whitney tests confirm that this improvement is indeed statistically significant: for both metrics on CIFAR-100 and SVHN; and for $\text{prop}^1_{i,j}$ on CIFAR-10. However, no statistically significant difference is observed for any of the other metrics.

If we look at the second combining method of "Novelty Search Extended with Accuracy Objectives" section, we observe a slight improvement, which is nonetheless statistically significant, on CIFAR-10 over LC for the metric $\text{cos\_dist}_{i,j}$, but this is not observed on CIFAR-100 or SVHN and, therefore, not a consistent result. And finally, if we look at the final method we propose, the explicit accuracy search with implicit diversity, we again see that, although improving on the worst metrics, there is no significant improvement observed consistently on all three datasets over NS or any of the combining methods for the two best diversity metrics, $\text{prop}^2_{i,j}$ and $\text{cos\_dist}_{i,j}$. Statistical significance is observed in particular cases—e.g. on CIFAR-10 and CIFAR-100 for NS with the metric $\text{cos\_dist}_{i,j}$ or on CIFAR-10 for LC with that same metric—but in any case the improvements in accuracy are very small. We therefore reject Hypothesis 6 since the results do not consistently back the claim that different combinations of diversity and accuracy objectives lead to significantly different ensemble accuracy. As observed before, the choice of diversity metric seems to play a more crucial role than the choice of a particular combination between diversity and accuracy.

Hypothesis 7

Hypothesis (Diversity and Accuracy Must Be Balanced) *Assigning greater importance to diversity in an ensemble leads to worse individual model accuracy. Conversely, weighting individual accuracy more leads to less diverse ensembles. There is a trade-off to be found between the two.*

Table 6 shows the average individual accuracy and the values of distance metrics for the final ensemble, measured on the test data for each of the datasets considered, with different methods and parameter settings. In the interests of clarity and due to limitations of space we only include results for some of the methods and diversity metrics, since other results do not contribute with any additional insight. The values of different diversity metrics are scaled *for the same dataset* so that the magnitude of variations across rows may be directly compared, hence the negative values. We have utilised the same min-max scalers that are fitted on the training data when pretraining the surrogate diversity model for each dataset.

The first thing we note is the clear correspondence between similar average individual accuracy and similar values for each diversity metric. The more similar the accuracy values, the more similar the diversity values. This is observed across different methods on all three datasets. For example, if we compare the rows for the NS method with metrics $\text{prop}^2_{i,j}$ and $\text{cos\_dist}_{i,j}$, we find no statistically significant difference in average individual accuracy or the values of diversity metrics, measured on the test set. If we take a closer look on the results of both the NS and LC, we see that

Table 6 Average individual accuracy and diversity metrics for the final ensemble after training, calculated on test sets. Median values over 10 runs. Distance values are scaled to facilitate comparison

	Method		Diversity metric	Avg. ind. acc. (%)	prop^1	prop^2	$\text{prop}^{\text{harm}}$	dis	cos_dist	arch_dist
CIFAR-10	NS		$\text{prop}^1_{i,j}$	19.53	0.895	0.203	0.309	0.234	0.132	0.015
			$\text{prop}^2_{i,j}$	**85.51**	**− 0.195**	**0.569**	**− 0.119**	**− 0.075**	**0.437**	**0.018**
			$\text{cos\_dist}_{i,j}$	**85.50**	**− 0.195**	**0.571**	**− 0.120**	**− 0.076**	**0.439**	**0.028**
	CM1: LC	$\alpha = 0.1$	$\text{prop}^1_{i,j}$	23.78	0.872	0.283	0.396	0.312	0.190	0.115
			$\text{cos\_dist}_{i,j}$	**85.89**	**− 0.201**	**0.570**	**− 0.129**	**− 0.081**	**0.438**	**0.010**
		$\alpha = 0.5$	$\text{prop}^1_{i,j}$	84.35	− 0.173	0.580	− 0.085	− 0.054	0.447	0.038
			$\text{cos\_dist}_{i,j}$	**85.41**	**− 0.196**	**0.567**	**− 0.122**	**− 0.077**	**0.435**	**0.023**
		$\alpha = 0.9$	$\text{prop}^1_{i,j}$	**85.05**	**− 0.191**	**0.563**	**− 0.113**	**− 0.072**	**0.431**	**0.054**
			$\text{cos\_dist}_{i,j}$	**85.59**	**− 0.198**	**0.569**	**− 0.123**	**− 0.078**	**0.436**	**0.022**
	Explicit Acc., Implicit Div.			**85.87**	**− 0.199**	**0.574**	**− 0.126**	**− 0.079**	**0.442**	**0.025**
CIFAR-100	NS		$\text{prop}^1_{i,j}$	4.85	0.923	0.086	0.159	0.113	0.059	0.027
			$\text{prop}^2_{i,j}$	**53.75**	**− − 0.806**	**0.536**	**0.474**	**0.378**	**0.434**	**0.040**
			$\text{cos\_dist}_{i,j}$	**53.76**	**− 0.801**	**0.538**	**0.473**	**0.377**	**0.435**	**0.029**
	CM1: LC	$\alpha = 0.1$	$\text{prop}^1_{i,j}$	7.07	0.892	0.143	0.245	0.186	0.100	0.073
			$\text{cos\_dist}_{i,j}$	**54.41**	**− 0.812**	**0.544**	**0.476**	**0.380**	**0.442**	**0.047**
		$\alpha = 0.5$	$\text{prop}^1_{i,j}$	52.61	− 0.730	0.552	0.497	0.402	0.448	0.045
			$\text{cos\_dist}_{i,j}$	**54.52**	**− 0.810**	**0.544**	**0.476**	**0.380**	**0.442**	**0.050**
		$\alpha = 0.9$	$\text{prop}^1_{i,j}$	**54.88**	**− 0.816**	**0.549**	**0.477**	**0.380**	**0.447**	**0.044**
			$\text{cos\_dist}_{i,j}$	**53.81**	**− 0.809**	**0.537**	**0.474**	**0.378**	**0.435**	**0.033**
	Explicit Acc., Implicit Div.			**54.68**	**− 0.818**	**0.544**	**0.475**	**0.379**	**0.441**	**0.041**
SVHN	NS		$\text{prop}^1_{i,j}$	16.23	0.851	0.126	0.144	0.138	0.087	0.055
			$\text{prop}^2_{i,j}$	**92.10**	**− 0.173**	**0.521**	**− 0.292**	**− 0.171**	**0.386**	**0.036**
			$\text{cos\_dist}_{i,j}$	**92.07**	**− 0.172**	**0.520**	**− 0.291**	**− 0.171**	**0.385**	**0.039**
	CM1: LC	$\alpha = 0.1$	$\text{prop}^1_{i,j}$	23.15	0.867	0.266	0.307	− 0.274	0.187	0.134
			$\text{cos\_dist}_{i,j}$	**92.24**	**− 0.174**	**0.525**	**− 0.294**	**− 0.172**	**0.390**	**0.029**
		$\alpha = 0.5$	$\text{prop}^1_{i,j}$	86.02	− 0.017	0.642	− 0.057	− 0.002	0.495	0.081
			$\text{cos\_dist}_{i,j}$	**92.39**	**− 0.177**	**0.516**	**− 0.300**	**− 0.175**	**0.381**	**0.020**
		$\alpha = 0.9$	$\text{prop}^1_{i,j}$	**92.23**	**− 0.175**	**0.514**	**− 0.297**	**− 0.174**	**0.379**	**0.062**
			$\text{cos\_dist}_{i,j}$	**92.40**	**− 0.176**	**0.519**	**− 0.298**	**− 0.174**	**0.384**	**0.027**
	Explicit Acc., Implicit Div.			**92.49**	**− 0.178**	**0.516**	**− 0.302**	**− 0.176**	**0.381**	**0.025**

for the metric $\text{prop}^1_{i,j}$, one of the worst-performing ones as discussed previously, increasing the weight α of the local competition score LC_i (Eq. 13) leads to a clear increase in the average individual accuracy of the models in the ensemble and a decrease in the observed value for this metric w.r.t. the test set. This is observed consistently on all three datasets and Mann–Whitney tests confirm that this difference is statistically significant at the 1% level between rows corresponding to different values of α and w.r.t. to plain NS. However, the exact opposite is observed for the rows corresponding to the metric $\text{cos\_dist}_{i,j}$, with the measured individual accuracy and diversity values remaining approximately constant for different values of α and even for the method that only uses accuracy objectives; statistically significant differences are not observed. This counter-example allows us to reject Hypothesis 7 since it is not the case in general that weighting diversity more will lead to worse individual accuracy or vice-versa, which is a surprising result. The observations suggest this is highly dependent on the choice of a diversity metric, rather than being a general rule. *Running the methods with a high-performing diversity metric does not seem to require a trade-off with individual model accuracy.*

Hypothesis 8

Hypothesis (Worse Performance Without Explicit Diversity) *Removing explicit diversity objectives*, *keeping only individual accuracy objectives when searching for an ensemble*, *leads to a decrease in ensemble diversity and*, *consequently*, *worse ensemble accuracy*.

Looking at Table 5, we can see that the last method, which searches only for explicit accuracy, with implicit diversity being generated by the evolutionary procedure, does not do worse than the best amongst the other methods. In fact, our analysis reveals that in some cases it achieves better accuracy than some of these other methods in a statistically significant way, although this is not observed consistently—i.e. for all methods and diversity metrics across all three datasets—and at any rate the differences are small. This means that Hypothesis 8 must be rejected, as removing explicit diversity objectives does not lead to worse ensemble accuracy. This is a surprising result given the findings of Cardoso et al. [2, 3], regarding the explicit search for diversity when compared to common methods that only promote it implicitly.

The Diversity-Accuracy Duality

For an explanation of these surprising results, we now turn again to Table 6 and focus on the last row for each dataset, corresponding to this explicit accuracy search method. We have already noted that a trade-off between accuracy and diversity is not required when the best metrics are utilised. We can also see that, for the last method, the values for the individual accuracy and each of the diversity metrics are very similar to those in the rows corresponding to the best diversity metrics, $\text{prop}^2_{i,j}$ and $\text{cos\_dist}_{i,j}$. The difference between the values for this explicit accuracy search and the rows corresponding to the worst metric, $\text{prop}^1_{i,j}$, is naturally statistically significant in most cases, with some exceptions observed for LC with $\alpha = 0.9$. For the other cases, a statistically significant difference is at times observed, as this method tends to result in slightly higher average individual accuracy, probably a result of *only* favouring accuracy during the search. However, these differences are very small and the key observation is that the average individual accuracy and the diversity values are very similar when comparing this method with all the other ones using the two best metrics, including the NS, *which only favours diversity* explicitly.

These results suggest that, contingent on the choice of a high-performing diversity metric—in this case, $\text{prop}^2_{i,j}$ or $\text{cos\_dist}_{i,j}$—there is an *equivalence* between searching for diversity and searching for accuracy. Regardless of the importance assigned to each of these two properties, the resulting ensembles will have similar average individual accuracy and diversity, and it is for this reason that their accuracy on test data is similar. We therefore hypothesise that, for these two diversity metrics, there is an *accuracy-diversity duality*, in the sense that these two properties appear to be interchangeable by means of an underlying process which is not yet understood, but which our methods nevertheless seem to approximate. This is highly significant because it suggests, in contrast with the literature, that there might not be a need to find a trade-off between diversity and accuracy in ensemble learning.

Conclusions

This paper has extended previous work [2], which proposed an innovative NS method to build behaviourally diverse ensembles of classifiers. The previous method had signposted an innovative way to construct high-performing

ensembles by explicitly searching for diversity. However, its application in practice had been hampered by limitations in the amount of available computational resources, since it involved a time-consuming step of training all networks in each generation of the NS with gradient descent. In [3], we propose a new method which overcomes this limitation by using a pretrained surrogate model to estimate the distance between neural network architectures, necessary to calculate novelty scores, without the need to train them. In this way, we can obtain an approximate speedup of 10 times w.r.t. the previous method when running them both with the same parameters, *without loss of classification accuracy*. We can also construct better-performing ensembles thanks to the expanded architecture search space facilitated by using a surrogate. We have confirmed previous observations that error diversity metrics lead to better-performing ensembles than more generic metrics.

We also build upon the preliminary results of [7] to describe an extension of this NS method. This extension incorporates accuracy objectives when searching for behaviourally diverse ensembles, so as to investigate the relationship and trade-offs between diversity and classification accuracy. Our initial research question was whether these accuracy objectives could lead to a performance gain in terms of final ensemble accuracy. We investigated a range of search methods that span the full spectrum of favouring only accuracy, only diversity, or different combinations of both. We found that accuracy objectives lead to significant improvements in ensemble accuracy, but only for the worst-performing diversity metrics. For the best metrics, performance was not improved upon regardless of the importance/weight assigned to accuracy objectives. But the most surprising result was the observation that there is an equivalence between searching for diversity—when defined by the two best metrics, $\text{prop}^2_{i,j}$ and $\text{cos\_dist}_{i,j}$—and searching for accuracy, with multiple ways of combining these two objectives leading to ensembles of similar diversity and average individual accuracy. When we considered the highest-performing metrics, there was no dichotomy between diversity and accuracy; each contributed to ensemble performance without detriment to the other and weighting one more did not impact negatively upon the other.

The augmented NS method thus represents an improved paradigm for implementing horizontal scaling of learning algorithms. It makes an explicit search for diversity considerably more tractable than our original approach *for the same bounded resources*. The observed equivalence between utilising diversity or accuracy objectives potentially means that the two are interchangeable and correlated in some conditions. This is a rather counter-intuitive result which suggests the existence of a diversity-accuracy duality in ensembles of classifiers. While further investigation of this equivalence is required so that stronger conclusions may be drawn, this result is significant because it challenges widespread assumptions about the need to trade off diversity for accuracy. An implication of this is the possibility of designing better algorithms which evolve diverse ensembles *without detriment to their accuracy*, since it is implicitly ensured.

Appendix: Additional Figures

We provide here additional material regarding the performance of the surrogate distance and surrogate accuracy models (Figs. 5, 6, 7, 8, 9, 10).

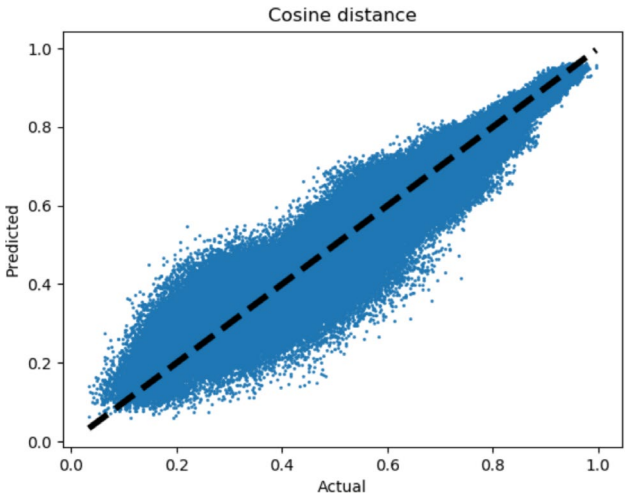


Fig. 5 Performance of surrogate model for predicting the cosine distance on CIFAR-10 ("Novelty Search Augmented with a Surrogate Model" section)

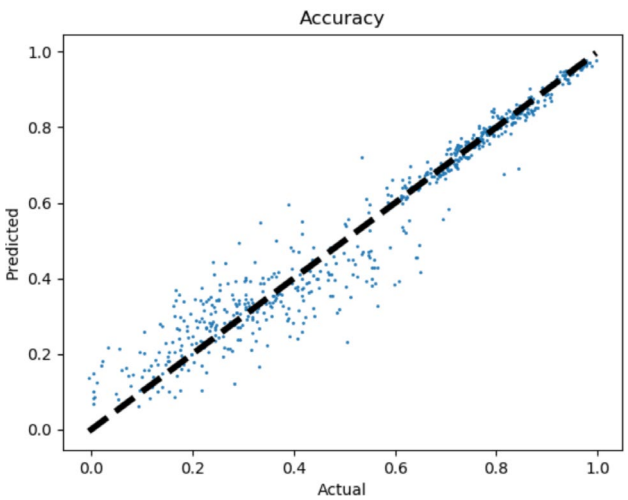


Fig. 6 Performance of surrogate model for predicting classification accuracy on CIFAR-10 ("Generic Search Method with Accuracy Objectives" section)

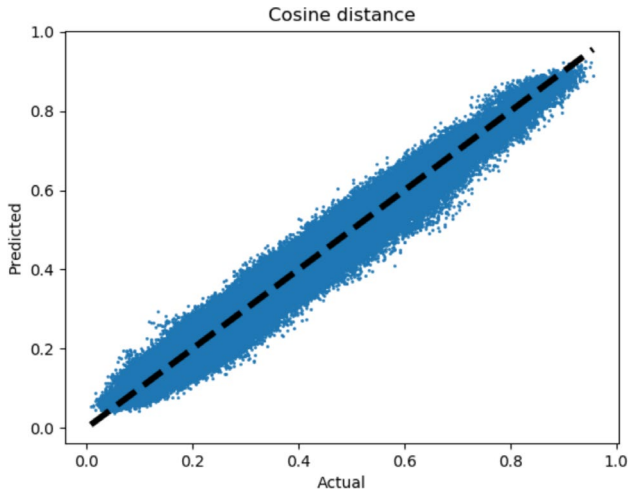


Fig. 7 Performance of surrogate model for predicting the cosine distance on CIFAR-100 ("Novelty Search Augmented with a Surrogate Model" section)

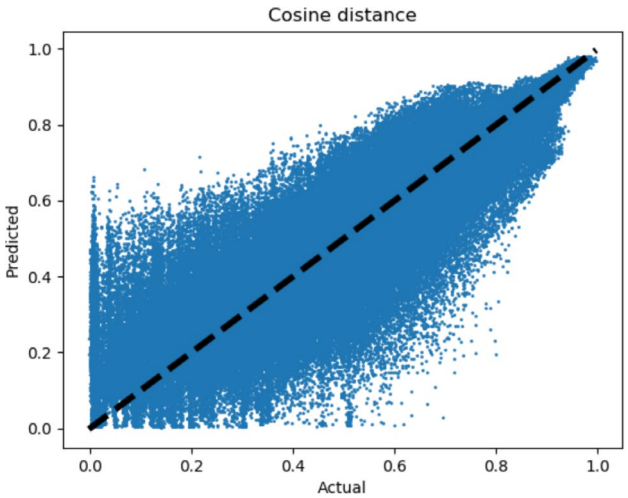


Fig. 9 Performance of surrogate model for predicting the cosine distance on SVHN ("Novelty Search Augmented with a Surrogate Model" section)

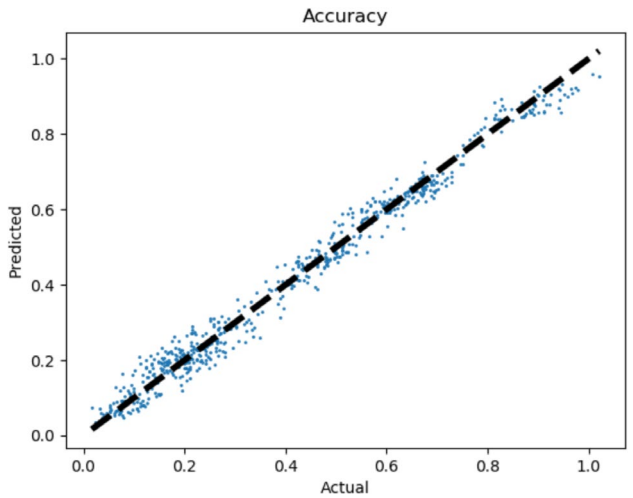


Fig. 8 Performance of surrogate model for predicting classification accuracy on CIFAR-100 ("Generic Search Method with Accuracy Objectives" section)

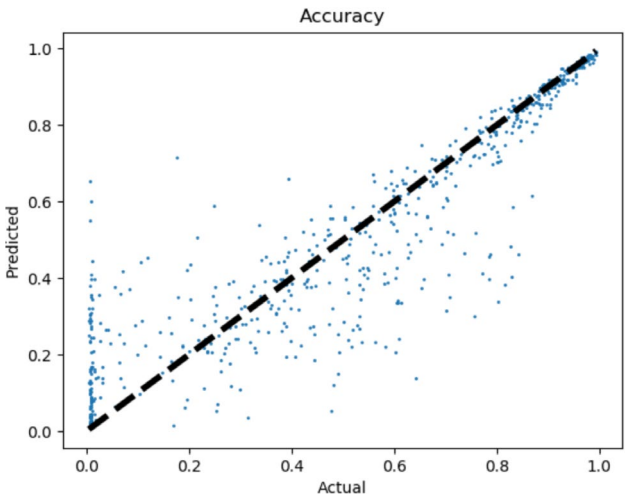


Fig. 10 Performance of surrogate model for predicting classification accuracy on SVHN ("Generic Search Method with Accuracy Objectives" section)

Author Contributions Not applicable.

Funding Not applicable.

Data Availability Not applicable.

Declarations

Conflict of interest Not applicable.

Research Involving Human and/or Animals Not applicable.

Informed Consent Not applicable.

References

1. Dietterich TG, Ensemble methods in machine learning. In: International workshop on multiple classifier systems. Berlin: Springer; 2000. p. 1–15.
2. Cardoso RP, Hart E, Kurka DB, Pitt JV. Using novelty search to explicitly create diversity in ensembles of classifiers. In: Proceedings of the Genetic and Evolutionary Computation Conference. GECCO '21. New York: Association for Computing Machinery; 2021. p. 849–857.
3. Cardoso RP, Hart E, Kurka DB, Pitt J. Augmenting novelty search with a surrogate model to engineer meta-diversity in ensembles of classifiers. In: Jiménez Laredo JL, Hidalgo JI, Babaagba KO, editors. Applications of evolutionary computation. Cham: Springer; 2022. p. 418–34.
4. Siems J, Zimmer L, Zela A, Lukasik J, Keuper M, Hutter F. Nas-bench-301 and the case for surrogate benchmarks for neural architecture search. CoRR https://arxiv.org/abs/2008.097772008.09777; 2020.
5. Lehman J, Stanley KO. Abandoning objectives: evolution through the search for novelty alone. Evol Comput. 2011;19:189–223.
6. Zagoruyko S, Komodakis N. Wide residual networks. In: Wilson RC, Hancock ER, Smith WAP, editors. Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016. BMVA Press; 2016. http://www.bmva.org/bmvc/2016/papers/paper087/index.html
7. Cardoso RP, Hart E, Kurka DB, Pitt JV. The diversity-accuracy duality in ensembles of classifiers. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. GECCO '22. New York: Association for Computing Machinery; 2022. p. 627–630. https://doi.org/10.1145/3520304.3528914.
8. Gu S, Jin Y. Generating diverse and accurate classifier ensembles using multi-objective optimization. In: 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM); 2014. p. 9–15 . https://doi.org/10.1109/MCDM.2014.7007182
9. Özöğür-Akyüz S, Windeatt T, Smith R. Pruning of error correcting output codes by optimization of accuracy-diversity trade off. Mach Learn. 2015;101(1–3):253–69. https://doi.org/10.1007/s10994-014-5477-5.
10. Zhu X, Zhong J, Zhuo L. Optimization of the trade-off by artificially re-sampling for ensemble learning. In: Third International Conference on Natural Computation (ICNC 2007), vol. 5; 2007. p. 49–53 https://doi.org/10.1109/ICNC.2007.527
11. Krizhevsky A. Learning Multiple Layers of Features from Tiny Images. ... Science Department, University of Toronto, Tech. ... (2009) arXiv: 1011.1669v3
12. Netzer Y, Wang T. Reading digits in natural images with unsupervised feature learning. Nips; 2011.
13. Floreano D, Dürr P, Mattiussi C. Neuroevolution: from architectures to learning. Evol Intell. 2008;1(1):47–62. https://doi.org/10.1007/s12065-007-0002-4.
14. Tong H, Huang C, Minku LL, Yao X. Surrogate models in evolutionary single-objective optimization: a new taxonomy and experimental study. Inf Sci. 2021;562:414–37.
15. Ruan X, Li K, Derbel B, Liefooghe A Surrogate assisted evolutionary algorithm for medium scale multi-objective optimisation problems. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference; 2020. p. 560–568.
16. Zhou Z, Ong YS, Nair PB, Keane AJ, Lum KY. Combining global and local surrogate models to accelerate evolutionary optimization. IEEE Trans Syst Man Cybern Part C (Appl Rev). 2006;37(1):66–76.
17. Gaier A, Asteroth A, Mouret J-B. Data-efficient neuroevolution with kernel-based surrogate models. In: Proceedings of the genetic and evolutionary computation conference; 2018. p. 85–92.
18. Stanley KO, Miikkulainen R. Evolving neural networks through augmenting topologies. Evol Comput. 2002;10(2):99–127.
19. Deng B, Yan J, Lin D. Peephole: Predicting network performance before training. CoRR https://arxiv.org/abs/1712.033511712.03351; 2017.
20. Stork J, Zaefferer M, Bartz-Beielstein T. Improving neuroevolution efficiency by surrogate model-based optimization with phenotypic distance kernels. In: International conference on the applications of evolutionary computation (part of EvoStar). Berlin: Springer; 2019. p. 504–519.
21. Chilès J-P, Desassis N. In: Daya Sagar BS, Cheng Q, Agterberg F, editors. Fifty years of kriging. Cham: Springer; 2018. p. 589–612. https://doi.org/10.1007/978-3-319-78999-6_29.
22. Sun Y, Wang H, Xue B, Jin Y, Yen GG, Zhang M. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. IEEE Trans Evol Comput. 2019;24:350–64.
23. Hagg A, Zaefferer M, Stork J, Gaier A. Prediction of neural network performance by phenotypic modeling. In: Proceedings of the genetic and evolutionary computation conference companion. GECCO '19. New York: Association for Computing Machinery; 2019. p. 1576–1582
24. Krogh A, Vedelsby J. Neural network ensembles, cross validation and active learning. In: Proceedings of the 7th international conference on neural information processing systems. NIPS'94. Cambridge: MIT Press; 1994. p. 231–238.
25. Chandra A, Chen H, Yao X. Trade-off between diversity and accuracy in ensemble generation. Stud Comput Intell. 2006;16:429–64. https://doi.org/10.1007/11399346_19.
26. Schapire RE. The strength of weak learnability. Mach Learn. 1990;5(2):197–227. https://doi.org/10.1007/BF00116037.
27. Breiman L. Bagging predictors. Mach Learn. 1996;24(2):123–40. https://doi.org/10.1007/BF00058655.
28. Bhowan U, Johnston M, Zhang M, Yao X. Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Trans Evol Comput. 2012;17(3):368–86.
29. Bhowan U, Johnston M, Zhang M, Yao X. Reusing genetic programming for ensemble selection in classification of unbalanced data. IEEE Trans Evol Comput. 2013;18(6):893–908.
30. Sheng W, Shan P, Chen S, Liu Y, Alsaadi FE. A niching evolutionary algorithm with adaptive negative correlation learning for neural network ensemble. Neurocomputing. 2017;247:173–82. https://doi.org/10.1016/j.neucom.2017.03.055.
31. Hart E, Sim K. On constructing ensembles for combinatorial optimisation. Evol Comput. 2018;26(1):67–87. https://doi.org/10.1162/EVCO_a_00203.
32. Tsakonas A. An analysis of accuracy-diversity trade-off for hybrid combined system with multiobjective predictor selection. Appl Intell. 2014;40(4):710–23. https://doi.org/10.1007/s10489-013-0507-8.
33. Cardoso RP, Hart E, Pitt JV. Diversity-driven wide learning for training distributed classification models. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. GECCO '20. New York: Association for Computing Machinery; 2020. p. 119–120
34. Cardoso RP, Hart E, Kurka DB, Pitt J. WILDA: wide learning of diverse architectures for classification of large datasets. In: Castillo PA, Laredo JLJ, editors. Applications of Evolutionary Computation—24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12694. Berlin: Springer; 2021. p. 649–664.

35. Mouret J, Clune J. Illuminating search spaces by mapping elites. CoRR https://arxiv.org/abs/1504.049091504.04909; 2015.
36. Mukhriya A, Kumar R. Building outlier detection ensembles by selective parameterization of heterogeneous methods. Pattern Recogn Lett. 2021;146:126–33. https://doi.org/10.1016/j.patrec.2021.03.008.
37. Brown G, Wyatt JL, Tiño P. Managing diversity in regression ensembles. J Mach Learn Res. 2005;6:1621–50.
38. Mellor A, Boukir S. Exploring diversity in ensemble classification: applications in large area land cover mapping. ISPRS J Photogrammetry Remote Sens. 2017;129:151–61. https://doi.org/10.1016/j.isprsjprs.2017.04.017.
39. Wang S, Yao X. Diversity analysis on imbalanced data sets by using ensemble models. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009, Part of the IEEE Symposium Series on Computational Intelligence 2009, Nashville, TN, USA, March 30, 2009—April 2, 2009. IEEE; 2009. p. 324–331. https://doi.org/10.1109/CIDM.2009.4938667.
40. Trawinski K, Quirin A, Cordón O. On the combination of accuracy and diversity measures for genetic selection of bagging fuzzy rule-based multiclassification systems. In: Ninth International Conference on Intelligent Systems Design and Applications, ISDA 2009, Pisa, Italy , November 30-December 2, 2009. IEEE Computer Society; 2009. p. 121–127. https://doi.org/10.1109/ISDA.2009.123.
41. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.
42. Paszke A, Gross S, Chintala S, Chanan G, Yang E, Facebook ZD, Research AI, Lin Z, Desmaison A, Antiga L, Srl O, Lerer A. Automatic differentiation in PyTorch. In: Advances in neural information processing systems, vol. 32; 2019.
43. Van Krevelen R. Error diversity in classification ensembles. PhD thesis; 2005.
44. Kuncheva LI, Whitaker CJ. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Mach Learn. 2003;51:181–207.
45. Pasti R, De Castro LN, Coelho GP, Von Zuben FJ. Neural network ensembles: immune-inspired approaches to the diversity of components. Nat Comput. 2010;9(3):625–53.
46. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.
47. Gomes J, Mariano P, Christensen AL. Devising effective novelty search algorithms: a comprehensive empirical study. In: GECCO 2015—proceedings of the 2015 genetic and evolutionary computation conference; 2015.
48. Wolpert DH. Stacked generalization. Neural Netw. 1992;5:241–59.
49. Lehman J, Stanley KO. Evolving a diversity of virtual creatures through novelty search and local competition. In: Proceedings of the 13th annual conference on genetic and evolutionary computation. ACM; 2011. p. 211–218.