Beyond Numerical Features: CNN-Driven Algorithm Selection via Contour Plots for Continuous Black-Box Optimization

Yiliang Yuan*, Xiang Shi†, Mustafa Misir†
* Mohamed bin Zayed University of Artificial Intelligence, Masdar City, United Arab Emirates
yiliang.yuan@mbzuai.ac.ae
† Duke Kunshan University, Suzhou, China
{xiang.shi, mustafa.misir}@dukekunshan.edu.cn

***Abstract*—The present paper introduces a new representation-driven approach to per-instance algorithm selection, applied to black-box optimization, for automatically choosing the most promising solver from a fixed portfolio. Prior work in continuous optimization largely relies on numerical descriptors, including Exploratory Landscape Analysis features and learned embeddings such as Deep-ELA. This work studies a complementary representation: contour-map visualizations of probed landscapes. A CNN regressor takes multiple instance-specific contour views (stacked or encoded per view and aggregated) and predicts per-solver performance, enabling selection by the predicted best value. On the standard BBOB 2009 single-objective protocol, the resulting selectors significantly outperform the single best solver (SBS) and are competitive with feature-based baselines. A subsequent bi-objective evaluation under the DeepELA setting further indicates that the same image-based principle can be competitive when using windowed contour views. Overall, the results suggest that simple vision models can exploit spatial structure in probed landscapes for algorithm selection without handcrafted ELA features.**

***Index Terms*—Algorithm selection; continuous black-box optimization; convolutional neural networks; deep learning**

I. Introduction

Continuous black-box optimization (BBO) is a common setting in science and engineering where only function evaluations are available and gradient information is inaccessible [1]. In such regimes, derivative-free solvers like CMA-ES variants offer strong performance, yet the No Free Lunch theorem implies that no single algorithm universally outperforms across all instances [2]. Consequently, the potential to improve performance by selecting a solver tailored to each problem instance rather than committing to a static choice, motivates Automated Algorithm Selection (AAS) [3], [4].

AAS in continuous BBO typically relies on instance representations derived from probing evaluations. Exploratory Landscape Analysis (ELA) constructs numerical feature vectors intended to summarize properties such as curvature, separability, and modality [5]. Recent works replace handcrafted ELA features with learned embeddings, reducing feature correlation and improving robustness under the same benchmark protocol [6]. However, both types of studies rely on abstract numerical descriptors that may fail to capture the spatial structure evident in a visualized landscape. In parallel, CNN-based AAS has been explored in discrete domains where direct instance encodings are available [7]; the continuous BBO setting differs in that a representation must be constructed through probing rather than read off from an explicit input.

This paper studies a probing-based, image-centered representation for continuous BBO. This paper mainly investigates whether 2-D contour maps of probed landscapes contain useful signal for AAS in continuous BBO. Each problem instance is rendered as a grayscale contour map from objective evaluations on a fixed grid. Then, a CNN model is built to predict per-solver performance and to choose the best algorithm. Specifically, two variants are considered: a combined-view model that stacks multiple instance views as channels, and a separate-view model that encodes each view and aggregates the resulting features. Evaluation on BBOB [8] under the established protocol shows that our methods substantially improve over the single best solver and are competitive with ELA/Deep-ELA [6] baselines. A follow-up bi-objective study following [6] further examines the same principle.

The main contributions are:

1) a probing-based AAS formulation for continuous BBO that uses contour-map renderings as instance representations and trains CNN selectors without handcrafted landscape features
2) an empirical study on single- and multi-objective optimization problems under standard AAS protocols, analyzing the effects of view aggregation and input resolution, and positioning contour-based selection relative to ELA and Deep-ELA baselines.

The remainder of this article is structured as follows. Section II discusses optimization and AAS. Our proposed CNN-based AAS setup and the representation it utilizes are detailed in Section III. Section IV reports the empirical results. Finally, Section V summarizes the work, presents the key findings, and outlines future research directions.

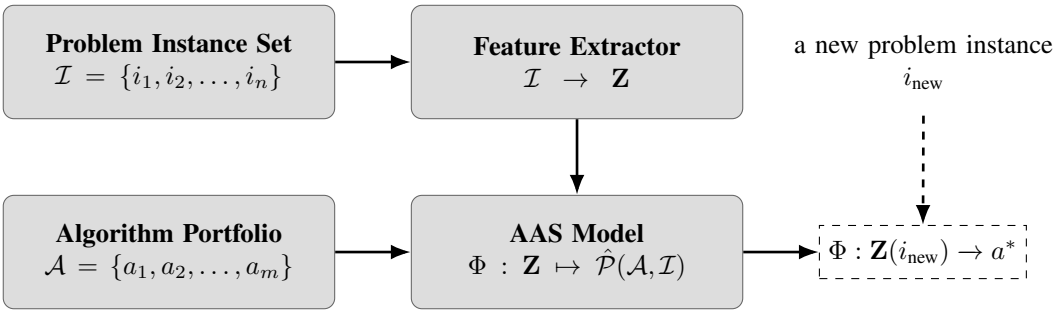

Fig. 1. A traditional performance prediction based Automated Algorithm Selection (AAS) process. $\mathcal{I}$ denotes the instance set, $\mathcal{A}$ the algorithm portfolio, and $\mathbf{Z}$ the feature extractor, yielding a predicted solver a^*.

II. Background

A. Optimization Problems

An optimization problem requires determining values for decision variables that minimize or maximize an objective function, possibly subject to constraints [9]. Problems are commonly categorized by variable domain (discrete vs. continuous) and by the number of objectives (single- vs. multi-objective). Discrete optimization involves integer, binary, or categorical variables and typically yields combinatorial search spaces. Continuous optimization, in contrast, deals with real-valued variables and induces continuous objective landscapes. Single-objective optimization targets one scalar objective, whereas multi-objective optimization considers multiple, often conflicting, objectives and evaluates solutions via Pareto dominance and the Pareto front.

B. Automated Algorithm Selection for Optimization

AAS largely aims to specify and apply the most suitable algorithm from a portfolio for solving a given problem instance. Typical AAS pipelines extract instance descriptors and combine them with empirical performance data, such as runtime or solution quality, to train predictive models that map instances to algorithms [10], as depicted in Figure 1.

Beyond handcrafted numerical descriptors, representation learning has been explored for AAS. For example, applying CNNs to image-like encodings of combinatorial problem instances, constructed from instance descriptions, enables selection without manual feature engineering [7]. This paradigm is *feature-free* in the sense of avoiding explicit engineered descriptors, but it does not rely on black-box probing nor on visualizing fitness landscapes concerning solution spaces.

In continuous BBO, AAS has been widely studied under benchmarked settings such as BBOB [8]. A dominant approach is **Exploratory Landscape Analysis (ELA)**, which computes numerical landscape features such as curvature, convexity, and level-set structure, from a limited sample of evaluations and uses these features to train AAS models [5], [11]. More recently, **Deep-ELA** replaces handcrafted ELA vectors with embeddings learned by pretrained transformers, trained via self-supervision on a large portfolio of synthetic optimization problems, yielding lower-correlation and higher-signal-to-noise-ratio representations for downstream selection [6]. Within this line, the central design choice is the representation of an instance under a probing budget. The present work follows the same AAS setting but considers a complementary representation that preserves spatial structure by mapping probed evaluations to contour-plot images, enabling CNN-based selection from visual landscape information.

III. Methodology

A. Benchmarks and Performance Metrics

This study uses two benchmarks, respectively for single-objective optimization (SOO) and multi-objective optimization (MOO), both mirroring the evaluation settings of Deep-ELA [6].

Single-objective optimization. The primary evaluation dataset is the widely used COCO/BBOB 2009 noiseless single-objective benchmark suite [8]. It consists of 24 scalable continuous functions grouped by properties such as separability, conditioning, and multimodality. It additionally supports multiple instances via random shifts in both decision space and objective space. Following [6] (originally built on [5]), a fixed portfolio of 12 representative and complementary optimizers is adopted from COCO [12]: deterministic baselines, multi-level/model-based solvers, CMA-ES variants, and OQNLP, enumerated in [5]. The raw benchmark comprises 480 instances (24 functions $\times$ 4 dimensions in $\{2,3,5,10\}$ $\times$ 5 instances). A *problem configuration* refers to a function-dimension pair (f, d), yielding 96 configurations, each associated with five instance-specific views.

Performance is measured by the Expected Running Time (ERT) to a fixed target precision $\varepsilon = 10^{-2}$:

$$\mathrm{ERT}(\varepsilon) = \frac{\sum_i \mathrm{FE}_i(\varepsilon)}{\sum_i \mathrm{Success}_i(\varepsilon)},$$

where $\mathrm{FE}_i(\varepsilon)$ is the number of evaluations used on instance i to reach the target, and $\mathrm{Success}_i(\varepsilon) \in \{0, 1\}$ is the binary success indicator. To compare across heterogeneous difficulties, relative ERT (relERT) is used. It normalizes ERT over the algorithm portfolio $\mathcal{A}$ per (f, d) pair by the minimal ERT:

$$\mathrm{relERT}_{f,d,a} = \frac{\mathrm{ERT}_{f,d,a}}{\min\limits_{a' \in \mathcal{A}} \mathrm{ERT}_{f,d,a'}}.$$

ERT is aggregated across the five instances per (f, d) before computing relERT, giving 96 relERT values per algorithm. Undefined values due to missing/failed runs are imputed using a PAR10-style penalty (10× the maximum finite relERT observed, 36,690.3) [5], to preserve comparability while penalizing non-performing algorithms.

Multi-objective optimization. Downstream evaluation additionally considers bi-objective problems following the DeepELA evaluation setting [6] (itself akin to [13]). The problem set contains 33 bi-objective instances with two-dimensional decision and objective spaces ($d = 2, m = 2$ for objective dimension m): ZDT [14], DTLZ [15], and MMF [16] instances (excluding ZDT5 and MMF13), plus Bi-BBOB [17] instances f_{46}, f_{47}, and f_{50}. The algorithm portfolio follows that setting: NSGA-II [18], SMS-EMOA [19], MOEA/D [20], and four additional optimizers (Omni-Optimizer [21], MOLE [22], MOGSA [23], HIGA-MO [24]).

Performance is measured by the hypervolume (HV) attained within a budget of 20,000 function evaluations. For most instances, the HV reference point is pre-specified; when unavailable, it is derived per instance from the least favorable solution among all algorithms. To compare across instances, HV is normalized using the best HV identified from additional runs of all algorithms with a budget of 100,000 evaluations. Following Deep-ELA, normalized HV is then converted into relative HV by contrasting against the VBS-SBS gap:

$$\mathrm{relHV}_{i,a} = \frac{\mathrm{HV}_{i,a} - \mathrm{HV}_{\mathrm{SBS}} + \epsilon}{\mathrm{HV}_{\mathrm{VBS}} - \mathrm{HV}_{\mathrm{SBS}} + \epsilon},$$

with $\epsilon = 10^{-8}$. Values near 0 correspond to SBS-level performance, values near 1 correspond to VBS-level performance, and negative values indicate performance worse than SBS.

B. Visual Representations via Contour Plots

SOO contour plots. Each SOO instance is represented as a 2D grayscale contour plot generated by evaluating the objective on a fixed uniform grid of resolution 300×300. For 2D problems, this grid is naturally defined on the ambient $[-5, 5]^2$ space. For higher-dimensional problems ($d \in \{3, 5, 10\}$), a 2D slice-based representation is used. Two coordinates are selected uniformly at random to span a 2D cross-sectional subspace $[-5, 5]^2 \subset [-5, 5]^D$ where the grid is situated, while all remaining coordinates are fixed to zero to leverage BBOB's symmetric domain. Each grid is evaluated using the official COCO implementation, and the resulting scalar field is normalized to $[0, 1]$ per plot and rendered as a filled contour plot. This yields a consistent 2D representation while preserving a partial view of the full d-dimensional landscape and informative geometric structures.

To study the effect of input fidelity on learning, alongside the original input resolution $r = 300$, generated plots are resized by interpolation to coarser CNN inputs with $r \in \{64, 128\}$ for additional experiments. Each SOO configuration (f, d) provides five plots (one per instance), which are concatenated to an input tensor $X \in \mathbb{R}^{5\times r\times r}$ for that configuration.

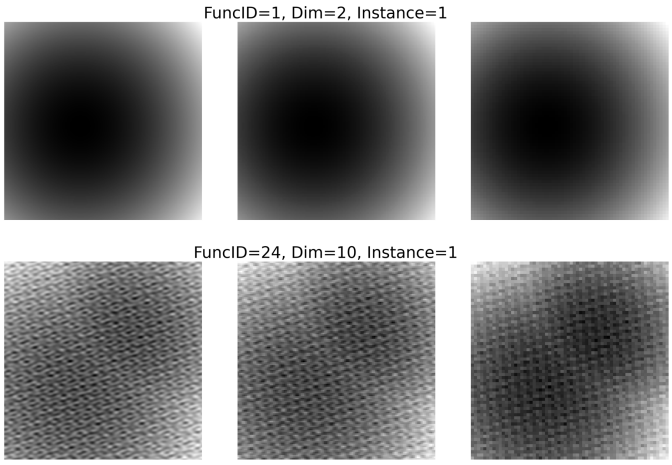

Fig. 2. Contour plots for different problems. From left to right, the *input* resolution decreases: 300×300, 128×128, and 64×64 pixels.

TABLE I
AVERAGE WALL-CLOCK TIME (SECONDS) TO GENERATE A SINGLE 300×300 CONTOUR PLOT, BY FUNCTION GROUP AND DIMENSION.

Function Group	Dimension	Avg. time per plot (s)
F1–F5	2D	0.18
	3D	0.21
	5D	0.21
	10D	0.27
F6–F9	2D	0.17
	3D	0.19
	5D	0.19
	10D	0.22
F10–F14	2D	0.17
	3D	0.19
	5D	0.21
	10D	0.25
F15–F19	2D	0.20
	3D	0.24
	5D	0.29
	10D	0.40
F20–F24	2D	0.31
	3D	0.28
	5D	0.29
	10D	0.47

The contouring step defines the probing cost: each contour plot is generated by evaluating the objective on a fixed 300×300 grid (i.e., 90,000 evaluations per plot), resulting in 450,000 evaluations per configuration. Figure 2 illustrates examples at different resized input resolutions. While the probing budget is large in evaluation-count terms, the measured preprocessing overhead on the BBOB benchmark implementation is modest in elapsed time, as Table I shows sub-second average generation times per plot. This makes the current design acceptable for offline proof-of-concept analysis, even though it is not intended for strict low-budget online deployment.

MOO sub-contour sampling. In the bi-objective setting considered here, the decision space is two-dimensional, so each objective can be visualized without slicing. Since each

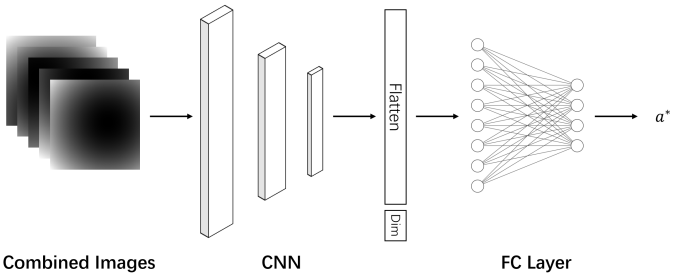

Fig. 3. Illustration of the combined model. The stacked contour tensor is encoded into feature vector z, which is passed to the regression head to predict per-algorithm performance and select the best solver a^*

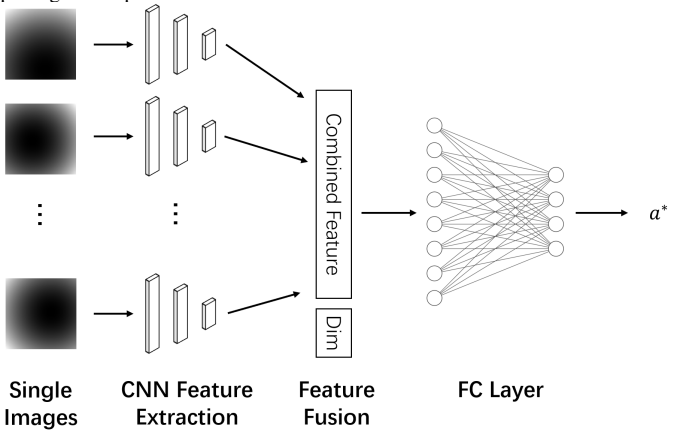

Fig. 4. Illustration of the separate model. Each contour view is encoded separately into a feature vector, and the resulting features are concatenated into z, which is passed to the regression head to predict per-algorithm performance and select the best solver a^*

MOO configuration represents one problem instance, to provide multiple informative views per instance and to reduce reliance on a single global rendering, a sub-contour-plot sampling strategy is adopted. For each instance, five axis-aligned rectangular regions are sampled uniformly within the domain; each region is a window proportionally shrunk by a scaling coefficient λ_{MOO} from the instance's domain (with empirically chosen $\lambda_{\text{MOO}} = 0.1$). Similar to SOO, within each region, two objectives are evaluated on a 300×300 grid, interpolated to obtain additional plots with $r \in \{64/128\}$, and normalized per plot, yielding two contour-plot views over the same window. With five maps per objective, each input consists of two multi-channel tensors $X_{\text{obj1}}, X_{\text{obj2}} \in \mathbb{R}^{5\times r\times r}$. This sampling process is repeated 20 times per instance, consistent with the evaluation design of Deep-ELA [6].

C. CNN Architectures for Algorithm Selection

The architecture consists of a vision encoder and a regression head that predicts per-algorithm performance. Given 5 views $X \in \mathbb{R}^{5\times r\times r}$, two variants are considered:

1) *Combined model*: The five views are stacked along the channel dimension, with random view ordering to avoid spurious dependence on stacking order. The encoder operates on the stacked view directly to extract the feature vector z. (Figure 3)
2) *Separate model*: Each view $x_i \in \mathbb{R}^{r\times r}$ is individually processed by a shared encoder, and the resulting features are concatenated to form z. (Figure 4)

Algorithm 1 CNN-Based Algorithm Selection (Combined vs. Separate)

Input: Contour-plot tensor $X \in \mathbb{R}^{k\times r\times r}$ (SOO), dimension d, and k=5; portfolio $\mathcal{A}$=$\{a_1, \ldots, a_m\}$.
Output: Selected algorithm a^*.
Combined model:
$z \leftarrow \Phi_{\text{cnn}}(X)$
$\hat{\mathcal{P}} \leftarrow \Psi(z, d)$ ▷ $\hat{\mathcal{P}} \in \mathbb{R}^m$
$a^* \leftarrow \arg\min_{\alpha\in\{1,\ldots,m\}} \hat{\mathcal{P}}_\alpha$ ▷ SOO (relERT); use $\arg\max$ for MOO (relHV)
Separate model:
$z_j \leftarrow \Phi_{\text{cnn}}(x^{(j)})$ for $j = 1, \ldots, k$ ▷ Φ shared
$z \leftarrow \text{Concat}(z_1, \ldots, z_k)$
$\hat{\mathcal{P}} \leftarrow \Psi(z, d)$ ▷ $\hat{\mathcal{P}} \in \mathbb{R}^m$
$a^* \leftarrow \arg\min_{\alpha\in\{1,\ldots,m\}} \hat{\mathcal{P}}_\alpha$ ▷ SOO; $\arg\max$ for MOO
MOO note: if using two objectives, replace X by $(X^{(1)}, X^{(2)})$; encode each to $z^{(1)}, z^{(2)}$ and concatenate before Ψ.

For SOO, the encoder is a three-layer CNN; the resulting feature vector is appended with the dimension d and fed to a regression head predicting per-solver relERT. For MOO, ResNet-18 is used as the encoder due to the higher representational demands observed empirically. The same encoder is applied to X_{obj1} and X_{obj2} with shared weights, producing embeddings $z_{\text{obj1}}, z_{\text{obj2}}$ that are concatenated and passed to an analogous regression head predicting per-solver relHV.

Input resolutions $r \in \{64, 128, 300\}$ are evaluated for both models on both SOO and MOO benchmarks. The only exception is the separate model at $r = 300$ under SOO evaluation, which is omitted due to training-cost considerations.

The overall procedure is summarized in Algorithm 1 and follows the same selection principle, i.e., predict performance over the portfolio and select by $\arg\min_a \mathrm{relERT}_a$ or $\arg\max_a \mathrm{relHV}_a$.

IV. Computational Analysis

A. Experiment Setting

SOO evaluation. Following the evaluation setting of [5], leave-one-out cross-validation is performed over the 96 SOO configurations: each fold holds out one (f, d) configuration and trains on the remaining 95.

MOO evaluation. Consistent with the evaluation design of Deep-ELA [6], for each bi-objective instance, 20 repetitions of the window-sampling procedure are generated, where 15 repetitions are used for training and the remaining 5 for testing.

Implementation detail. Model training uses the same implementation across folds, with hyper-parameters selected via a lightweight grid search on a validation split within each fold.

TABLE II

relERT VALUES OF DIFFERENT MODELS. SBS IS THE HCMA ALGORITHM, THE DEEP-ELA MEDIUM MODEL CANNOT HANDLE DATA MORE THAN 6 DIMENSIONS, RESULTING IN EMPTY SPACE FOR D=10. THE LAST FIVE COLUMNS ARE THE RESULTS OF OUR CNN-BASED AAS MODELS.

D	F. Group	SBS	ELA (50d)*		Large (25d)*		Medium (25d)*		Combined CNN			Separate CNN	
			RF	MLP	kNN	RF	kNN	RF	64×64	128×128	300×300	64×64	128×128
2	1	**3.71**	10.41	10.59	9.24	10.30	17.57	7.57	14.65	14.65	4.12	14.89	14.65
	2	5.80	8.51	3.72	2.65	2.87	2.69	3.37	**1.66**	1.74	11.52	**1.66**	**1.66**
	3	6.29	1473.16	4.72	3.52	3.12	3.91	4.40	1.27	**1.0**	**1.0**	**1.0**	**1.0**
	4	25.34	**3.89**	9.25	6.45	9.76	5.73	5.99	8.08	6.86	6.20	8.08	8.08
	5	44.95	148.39	**3.32**	3.69	5.06	3.79	8.08	3.61	5.60	4.09	3.87	3.87
	all	17.69	342.22	6.43	5.21	6.36	6.91	5.99	6.03	6.15	**5.13**	6.08	6.03
3	1	356.10	1480.68	11.87	19.94	66.76	11.72	53.23	**4.80**	5.54	6.22	5.54	6.84
	2	4.46	8.33	3.50	2.73	3.02	2.67	2.87	5.32	1.90	3.50	**1.83**	**1.83**
	3	4.98	7.07	3.82	2.69	3.48	2.59	3.53	3.06	**2.29**	2.92	4.98	2.85
	4	**2.63**	441.96	5.06	5.15	4.63	5.36	4.83	8.90	5.21	3.95	**2.63**	8.90
	5	66.81	**1.22**	2.54	30.75	12.02	2.70	4.99	66.30	51.57	2.83	51.39	2.73
	all	90.43	403.67	5.44	12.65	18.61	5.10	14.35	18.19	13.78	**3.90**	13.75	4.75
5	1	11.99	14.14	**11.97**	17.01	17.31	17.50	17.85	16.47	11.99	11.99	11.99	11.99
	2	3.90	369.26	2.62	2.40	4.27	2.48	2.45	5.06	**1.76**	3.90	3.90	3.90
	3	4.21	150.44	3.97	3.70	4.87	3.76	3.69	3.31	**3.07**	4.21	4.21	4.21
	4	4.29	1470.28	6.81	**1.87**	4.07	3.95	3.51	4.29	3.68	4.29	4.29	4.29
	5	7.67	1.13	1.83	**1.08**	4.73	1.72	1.43	2.99	7.67	7.67	7.67	7.67
	all	6.52	402.38	5.56	**5.47**	7.17	6.02	5.93	6.48	5.80	6.52	6.52	6.52
10	1	**2.74**	14.64	15.27	9.54	9.53	-	-	**2.74**	3.44	**2.74**	**2.74**	**2.74**
	2	2.16	**1.62**	1.76	2.40	2.43	-	-	2.16	2.16	2.16	2.16	2.16
	3	**2.76**	2.87	4.35	3.54	3.62	-	-	**2.76**	**2.76**	**2.76**	**2.76**	**2.76**
	4	2.02	442.01	**1.96**	2.06	2.05	-	-	2.02	2.02	2.02	2.02	2.02
	5	23.64	148.01	3.25	**1.73**	11.33	-	-	23.64	23.64	23.64	23.64	23.64
	all	6.85	126.84	5.46	**3.91**	5.93	-	-	6.85	7.00	6.85	6.85	6.85
all	1	93.63	379.97	12.43	14.10	25.97	15.60	26.22	9.67	8.90	**6.27**	8.79	9.05
	2	4.08	96.93	2.90	2.54	3.15	2.62	2.90	3.55	**1.89**	5.27	2.39	2.39
	3	4.56	408.38	4.21	3.36	3.77	3.42	3.88	2.60	**2.28**	2.72	3.24	2.71
	4	8.57	589.54	5.77	3.88	5.13	5.01	4.78	5.82	4.44	**4.12**	4.25	5.82
	5	35.77	74.69	**2.74**	9.31	8.29	**2.73**	4.83	24.14	22.12	9.56	21.64	9.48
	all	30.37	318.78	5.72	6.81	9.52	6.01	8.76	9.39	8.18	**5.60**	8.30	6.04

* The results of ELA-based models, ELA (50d), are taken from [4] and those of Deep-ELA based models, Large (25d) and Medium (25d), are taken from [6].

All models are trained on a workstation equipped with 2 AMD EPYC 7763 64-Core CPUs, 8 NVIDIA RTX 3090 GPUs, and 256 Gigabytes of RAM, using PyTorch for implementation. Due to the modest model size, each model is trained on a single GPU.

B. Baselines

Performance is compared against the single best solver (SBS), defined as the portfolio algorithm with the lowest mean relERT over the 96 configurations, representing the best static choice. In the current setup, SBS corresponds to HCMA. Feature-based AAS baselines include two ELA-based selectors, with multilayer-perceptron (MLP) and random forest (RF) heads respectively, from [4] and Deep-ELA variants, with RF and kNN heads, from [6]. For SOO, the $25d$ Deep-ELA variants are reported, as they exhibit more consistent performance across function groups and dimensions than their $50d$ counterparts, while for MOO all variants are included. Deep-ELA Medium is constrained by its combined dimensionality limit $d{+}m \leq 6$, and therefore SOO results for $d{=}10$ ($m{=}1$) are outside of its scope.

C. Single-objective optimization

Table II reports relERT aggregated by dimension and function group. Across all configurations, the combined CNN improves substantially over SBS (mean relERT $30.37 \rightarrow 5.60$ at $r{=}300$), while remaining competitive with the strongest feature-based baselines (ELA-MLP: 5.72). The separate CNN also improves over SBS (e.g., 6.04 at $r{=}128$), though its gains are less consistent across groups.

Performance varies by dimension and function group. The most pronounced gains occur at $d{=}3$, where the combined CNN at $r{=}300$ attains mean relERT 3.90, outperforming the reported ELA and Deep-ELA baselines in the same slice of the table. Across function groups, CNN-based selectors perform strongly on groups 1–4 (F1–F19), while performance on group 5 (F20–F24) remains mixed, where some feature-based methods retain an advantage.

Input fidelity matters: for the combined model, increasing r typically reduces mean relERT (e.g., overall $9.39 \rightarrow 8.18 \rightarrow 5.60$ for $r{=}64, 128, 300$). This trend is consistent with the boxplot in Figure 5, where the combined CNN at $r{=}300$

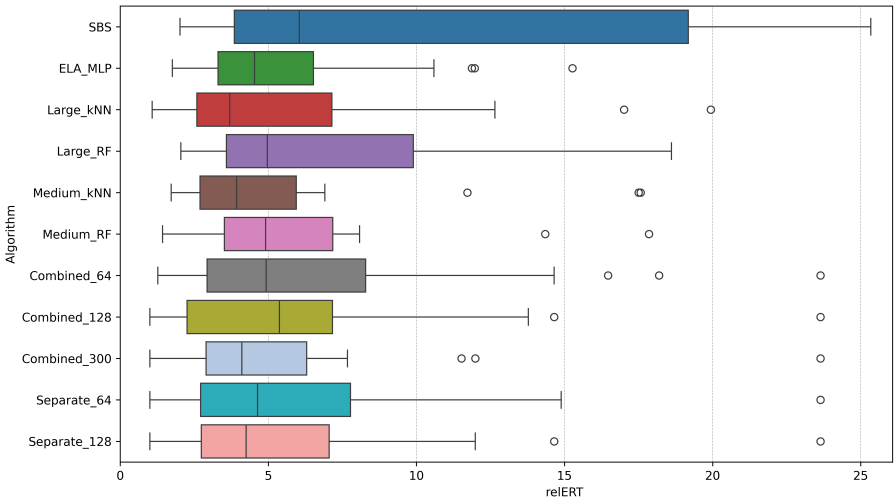

Fig. 5. relERT distribution per algorithm across all function groups and dimensions.

exhibits a lower median and a tighter distribution than its lower-r counterparts. These gains come with increased training cost; hence r represents a practical accuracy–cost trade-off rather than a change in probing budget.

To assess robustness across aggregated settings, a paired Wilcoxon signed-rank test is conducted over the 20 aggregated relERT entries (4 dimensions $\times$ 5 function groups) comparing the combined CNN (r=300) against selected baselines. The combined CNN significantly improves over SBS ($p = 0.037$). Against ELA-MLP and Deep-ELA Medium-kNN, there is no statistically significant difference over all 20 entries ($p = 0.286$ and $p = 0.598$, respectively), while restricting to groups 1–4 yields stronger evidence against ELA-MLP ($p = 0.05$) and remains inconclusive against Deep-ELA Medium-kNN ($p = 0.339$). Given the small number of aggregated test points, these tests are best interpreted as supportive rather than definitive.

D. Multi-objective optimization

Table III reports relHV on 33 bi-objective instances grouped by benchmark family, where values near 1 indicate closing the SBS–VBS gap and negative values indicate worse-than-SBS performance. Overall, both CNN variants are competitive and outperform the feature-based Deep-ELA baselines in the aggregate. The best mean relHV scores are achieved by the Separate CNN, ranging from 0.971–0.974. For the Combined CNN, r=128 performs best (0.964), with r=64 and r=300 slightly lower (0.933 and 0.946). Among the feature-based baselines, Deep-ELA Medium (kNN; 25d/50d) is strongest (0.904), while the remaining variants perform worse overall.

The breakdown by benchmark family highlights where gains arise. BiBBOB is saturated (all methods at 1.0), leaving little discriminative room under this grouping. On ZDT, CNN-based selectors essentially solve the subset (Separate: 1.0 for all r; Combined: 0.960 at r=64 and 1.0 at $r \geq 128$), whereas some Deep-ELA settings are near-zero or negative (e.g., Large-25d), indicating failures to outperform SBS. On DTLZ and MMF, CNN models remain strong but do not uniformly match the best Deep-ELA settings: Separate CNN reaches up to 0.981 on DTLZ (vs. 1.0 for Deep-ELA kNN, 50d) and up to 0.969 on MMF (vs. 1.0 for Deep-ELA Medium kNN, 50d).

Input resolution effects are modest in the overall averages but not monotone across families. The Separate CNN benefits from higher r on DTLZ (up to 0.981 at r=300) yet degrades on MMF at r=300 (0.917), suggesting that finer-scale window detail can be informative for some landscapes but less stable under stochastic window sampling for others. Given the near-identical mean performance across r, r=64 is a reasonable practical default in MOO, with $r \in \{128, 300\}$ serving as sensitivity checks rather than a uniformly better setting.

V. Discussion and Conclusion

This work studies automated algorithm selection for continuous black-box optimization using an image-based representation of problem instances. Each configuration is probed by generating contour maps from objective evaluations on a fixed 300×300 grid (five instance-specific views per (f, d)), and a CNN-based regressor predicts per-solver performance for selection. Across the BBOB 2009 SOO benchmark under LOOCV, the best-performing variant (combined CNN with r=300 input) reduces mean relERT from 30.37 (SBS) to 5.60, and is comparable in aggregate to the strongest feature-based baseline reported in Table II (ELA-MLP: 5.72). Input resizing from the same probed field indicates that higher input fidelity tends to improve selection quality, at the cost of higher training-time and memory demands.

The same contour-based principle also extends to the bi-objective setting under the Deep-ELA protocol. Under this inherited evaluation setting, apart from BiBBOB which is saturated across all methods, CNN-based selectors remain competitive on relHV and perform particularly strongly on ZDT, while DTLZ and MMF remain closer to the strongest Deep-ELA variants. These results suggest that the image-based representation is not confined to the SOO setting.

TABLE III
relHV OF FUNCTION GROUPS (ROWS) OVER DIFFERENT METHODS/SETTINGS (COLUMNS).

Function group	Large (25d)		Large (50d)		Medium (25d)		Medium (50d)		Combined CNN			Separate CNN		
	kNN	RF	kNN	RF	kNN	RF	kNN	RF	64	128	300	64	128	300
BiBBOB	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
DTLZ	0.914	0.457	**1.000**	0.651	**1.000**	0.651	**1.000**	0.822	0.819	0.914	0.876	0.924	0.924	0.981
MMF	0.892	0.463	0.931	0.609	0.929	0.764	**1.000**	0.933	0.886	0.941	0.907	0.969	0.960	0.917
ZDT	-0.079	-0.087	0.041	0.209	0.688	0.185	0.616	0.745	0.960	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
all	0.682	0.458	0.743	0.617	0.904	0.650	0.904	0.875	0.933	0.964	0.946	0.973	0.971	**0.974**

Several limitations bound the scope of the claims. First, the SOO probing budget is substantial (five 300×300 maps per configuration), which is acceptable for offline dataset construction but mismatched to strict online probing regimes. This study focuses on representational feasibility rather than probing-budget optimality. Therefore, budget-aware probing and direct low-resolution evaluation are natural next steps. Second, for $d > 2$, the representation is only a partial visual probe of the landscape. A single axis-aligned 2D slice with fixed coordinates may miss important high-dimensional structure, which plausibly contributes to the weaker performance on the hardest function group. Third, both SOO and MOO evaluations are tied to a fixed portfolio and protocol (including window sampling for MOO), and the MOO study is limited to d=2, m=2. Future work should therefore prioritize cost-sensitive and multi-view probing designs, stronger high-dimensional representations such as multiple slices or hybrid visual–numerical features, and validation across alternative portfolios as well as optimization budgets.

ACKNOWLEDGMENT

We thank Shiya Ye for assistance with the initial collection and organisation of the BBOB-related data used in this study.

REFERENCES

[1] F. X. Long, D. Vermetten, B. van Stein, and A. V. Kononova, "Bbob instance analysis: Landscape properties and algorithm performance across problem instances," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 2023, pp. 380–395.

[2] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[3] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, "Automated algorithm selection: Survey and perspectives," *Evolutionary computation*, vol. 27, no. 1, pp. 3–45, 2019.

[4] R. P. Prager, M. V. Seiler, H. Trautmann, and P. Kerschke, "Automated algorithm selection in single-objective continuous optimization: a comparative study of deep learning and landscape analysis methods," in *International Conference on Parallel Problem Solving from Nature (PPSN)*. Springer, 2022, pp. 3–17.

[5] P. Kerschke and H. Trautmann, "Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning," *Evolutionary Computation*, vol. 27, no. 1, pp. 99–127, 2019.

[6] M. V. Seiler, P. Kerschke, and H. Trautmann, "Deep-ela: Deep exploratory landscape analysis with self-supervised pretrained transformers for single-and multi-objective continuous optimization problems," *Evolutionary Computation*, pp. 1–27, 2025.

[7] A. Loreggia, Y. Malitsky, H. Samulowitz, and V. Saraswat, "Deep learning for algorithm portfolios," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[8] S. Finck, N. Hansen, R. Ros, and A. Auger, "Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions," INRIA, Tech. Rep. RR-6829, 2009, updated version as of February 2019. [Online]. Available: https://inria.hal.science/inria-00362633v2/document

[9] C. Daskalakis, S. Skoulakis, and M. Zampetakis, "The complexity of constrained min-max optimization," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021, pp. 1466–1478.

[10] M. Alissa, K. Sim, and E. Hart, "Automated algorithm selection: from feature-based to feature-free approaches," *Journal of Heuristics*, vol. 29, no. 1, pp. 1–38, 2023.

[11] M. Seiler, U. Škvorc, G. Cenikj, C. Doerr, and H. Trautmann, "Learned features vs. classical ela on affine bbob functions," in *International Conference on Parallel Problem Solving from Nature (PPSN)*. Springer, 2024, pp. 137–153.

[12] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff, "COCO: A platform for comparing continuous optimizers in a black-box setting," *Optimization Methods and Software*, vol. 36, pp. 114–144, 2021.

[13] J. Rook, H. Trautmann, J. Bossek, and C. Grimme, "On the potential of automated algorithm configuration on multi-modal multi-objective optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2022, pp. 356–359.

[14] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[15] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization: theoretical advances and applications*. Springer, 2005, pp. 105–145.

[16] C. Yue, B. Qu, K. Yu, J. Liang, and X. Li, "A novel scalable test problem suite for multimodal multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 48, pp. 62–71, 2019.

[17] D. Brockhoff, A. Auger, N. Hansen, and T. Tušar, "Using well-understood single-objective functions in multiobjective black-box optimization test suites," *Evolutionary computation*, vol. 30, no. 2, pp. 165–193, 2022.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[19] N. Beume, B. Naujoks, and M. Emmerich, "Sms-emoa: Multiobjective selection based on dominated hypervolume," *European journal of operational research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[20] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[21] K. Deb and S. Tiwari, "Omni-optimizer: A procedure for single and multi-objective optimization," in *International conference on evolutionary multi-criterion optimization*. Springer, 2005, pp. 47–61.

[22] L. Schäpermeier, "An r package implementing the multi-objective landscape explorer (mole)," 2022.

[23] C. Grimme, P. Kerschke, and H. Trautmann, "Multimodality in multi-objective optimization–more boon than bane?" in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2019, pp. 126–138.

[24] H. Wang, A. Deutz, T. Bäck, and M. Emmerich, "Hypervolume indicator gradient ascent multi-objective optimization," in *International conference on evolutionary multi-criterion optimization*. Springer, 2017, pp. 654–669.