Learning Improvement Representations to Accelerate Evolutionary High-Dimensional Multiobjective Optimization

Songbai Liu, Zeyi Wang, Qiuzhen Lin, Jianqiang Li, and Kay Chen Tan

Supplementary File

Abstract

We provide more discussion, details on the proposed methods, experimental results and analysis in this supplementary file.

1 Learnable MOEAs

The conventional evolutionary generator and discriminator (or selector) in a typical MOEA are constructed using fixed genetic operators, such as crossover, mutation, and selection. Consequently, they lack the capability to learn and adapt to the specific characteristics of the optimization problem they are tasked with. As a result, they cannot effectively respond to the potential challenges posed by solving such a black-box optimization problem. In the context of evolutionary computation, research studies focused on learnable Multi-Objective Evolutionary Algorithms (MOEAs) have garnered substantial attention [Liu, 2022]. Machine learning (ML) techniques are leveraged to support and enhance various modules within learnable MOEAs, including the generator, and discriminator (or selector).

Specifically, the MOEA's generator iteratively explores the variable space, generating a significant volume of data comprising feasible solutions. ML techniques systematically analyze this data to gain insights into the search behavior and enhance its future search capabilities. By traversing promising directions learned within the search space, the MOEA's generator efficiently identifies solutions with high potential [Michalski, 2000].

The MOEA's discriminator benefits from online predictions of Pareto Front (PF) shapes, enabling it to adeptly filter out underperforming solutions when dealing with MOPs featuring irregular PFs. Prior to tackling these problems, dimension reduction and spatial transformation techniques simplify both the objective and search spaces.

Furthermore, reinforcement learning (RL) techniques come into play in determining suitable evolutionary operators (i.e., actions) based on the current parent state. These actions guide the generator in producing high-quality offspring. Domain adaptation techniques are employed to learn domain-invariant feature representations across different optimization problems, enabling the analysis of distribution divergence. This knowledge transfer facilitates the sequential or simultaneous solution of these problems.

Bayesian optimization is an optimization technique that uses probabilistic models, like Gaussian Processes, to efficiently optimize complex, expensive functions. It sequentially samples the function, updates the surrogate model, and selects the next sampling point using an acquisition function. This process balances exploration and exploitation to find the global optimum. However, Bayesian optimization has limitations: it can be computationally expensive, struggles with high-dimensional spaces, assumes smooth functions, depends on the quality of the initial model, and may converge to local optima for multimodal functions.

2 Learnable Evolutionary Generators

Learnable Evolutionary Generators (LEGs) represent a unique category of Evolutionary Algorithms (EAs) that blend the heuristic essence of EAs with the deterministic attributes of Machine Learning (ML) methods. The distinctive feature of LEGs lies in their ability to expedite the convergence of an evolutionary process, achieving optimal solutions in fewer iterations. The synergy between ML and EAs becomes feasible due to the substantial number of individuals generated in an evolutionary process. This abundance enables ML models to discern between high and low-performance individuals, formulating hypotheses on generating new individuals with enhanced performance. This dynamic interplay enhances the overall search for optimal solutions by introducing diversity into the population and identifying performance-improved solutions.

2.1 Competitive Learning-based LEGs

To enhance the search capability when addressing HMOPs, certain MOEAs have developed competitive learning-based generators. In these generators, the population is divided into two groups: winners and losers, and the search process is guided so that the losers move closer to the winners. Typically, these competitions occur within the objective space. The effectiveness of these strategies heavily relies on the quality of the winners, as losers are guided towards them through competitive learning. If the winners encounter difficulties, such as falling into local optima, the entire evolutionary population may experience slow convergence. Furthermore, winners, often subjected to genetic operators, may only exhibit slight improvements. Consequently, the challenge lies in determining how these higher-quality winners can further evolve with a faster convergence rate.

2.2 Innovization-based LEGs

To address this challenge and expedite the search process, efforts have been made through the design of online innovization generators. The term "innovization" is derived from "innovation via optimization" and was originally defined as a post-optimality analysis of optimal solutions to provide valuable design principles to engineering designers. In innovization-based generators, offspring generated by genetic operators are further refined by innovization operators to progress along the learned directions of performance improvement [Gaur and Deb, 2017]. In the innovization process, various data-mining and machine learning techniques are employed to automatically uncover innovative and crucial design principles within optimal solutions. These principles may include inter-variable relationships, commonalities among optimal solutions, and distinctions that set them apart from one another [Mittal *et al.*, 2020]. These operators enable the population to converge faster without consuming additional function evaluations compared to traditional local search methods.

Expanding on the concept of innovization, the notion of knowledge-driven optimization is introduced. In this approach, MOEAs assimilate knowledge, such as latent patterns shared by high-quality solutions, learned from intermediate solutions to guide their search away from mediocre solutions and toward potentially promising regions. This constitutes an online learning process that involves deciphering what makes one solution optimal (or near-optimal) within the final solution set and understanding what causes one solution to outperform another during the optimization process. Online innovization aims to accelerate evolutionary search and enhance the efficiency of generators.

In an online innovization generator, a supervised learning model is typically constructed and trained online with the aim of implicitly learning historical directional improvements, such as transitioning from dominated to non-dominated solutions, within the previously explored search space. A solution $x = (x_1, x_2, \dots, x_n)$ is identified as performing poorly, while x^* represents a high-quality solution. Throughout the evolutionary process, various solution pairs (x, x^*) can be collected. Subsequently, the selected model, which can be a multilayer perceptron, a random forest, or a deep neural network, is trained using this labeled data. In this context, x serves as the input, and x^* serves as its label or expected target output. The trained model is believed to have the capacity to capture underlying patterns that reflect the directional improvement of solutions within the search space.

Ideally, a newly generated offspring solution, x^{new}, produced by genetic operators can be enhanced (or progressed) by inputting it into the well-trained model to obtain an improved version, y^{new}, as the output. This process of repairing offspring is also referred to as innovized progress. It holds the potential to enhance the search capability of generators when tackling scalable MOPs, primarily due to the following four merits: 1) Incorporation of all conflicting objective information during the learning process. 2) Elimination of the need for additional function evaluations during the innovized progress of solutions. 3) Adaptability of the learned directional improvement of solutions as generations progress. 4) The potential for a substantial leap in the objective space when transitioning from x^{new} to y^{new} in the search space, which can expedite the search process. However, it's crucial to consider four key considerations when customizing such an innovization progress:

Selection of the learning model: The choice of the learning model is flexible and can align with available supervised learning models that meet the requirements. However, it's essential to take into account the computational cost associated with training the selected model.

Collection of training data: The process of collecting training data involves gathering paired data based on the performance of available solutions, either from previous generations or the current one. Therefore, when selecting a pair (x, x^*), it is crucial to consider that x^* outperforms x. For example, in a Pareto-based MOEA, x^* should dominate x, in a decomposition-based MOEA, the aggregation fitness of x^* should be superior to that of x, or x^* should have the potential to guide x towards rapid convergence.

Training of the adopted model: The process of training a model is itself an optimization problem, involving numerous hyperparameters such as model architecture, learning rate, and training epochs, which often require manual tuning. This can lead to various challenges, including the risk of overfitting the model. Additionally, it's essential to investigate whether the model should be updated regularly, such as every generation, and whether training should be conducted online or offline.

Advancement of the search capability with the learned model: The expectation is that the generator's search capability can be enhanced with the assistance of this well-trained model. Specifically, subpar and average solutions within the population can be repaired, facilitating rapid convergence in the learned promising direction. Simultaneously, high-quality solutions can be further improved to explore a broader range of elite solutions. However, two important considerations arise: Is it necessary to repair all newly generated solutions? Is it necessary to perform the innovization progress in every generation?

2.3 Autoencoder-based LEGs

The concept of autoencoder-based representation learning has been previously introduced in MOEA/PSL, where it focuses on acquiring a compressed representation of the input solution, referred to as Pareto Subspace Learning (PSL). This approach involves training a Denoise Autoencoder (DAE), a specific type of artificial neural network with equal nodes in the input and output layers. The DAE comprises two main components: an encoder and a decoder. The encoder learns to create a representation (or code) of the partially corrupted input, while the decoder maps this representation back to a reconstructed input. Through this process, the DAE can extract higher-level features from the input distribution.

During training, the DAE iteratively minimizes the reconstruction error, which measures the disparity between the output and the input. This training process resembles that of a feedforward neural network. To elaborate, each input undergoes perturbation via the mutation operator, and the non-dominated solutions within the current population serve as the training data.

Following training, each solution can be mapped between the original search space and the compressed code representation space. Subsequently, a new offspring, denoted as y, can be generated through the following process:

Step 1: select two random parents x^1 and x^2 from the current population.

Step 2: map x^1 and x^2 to the code space to get their corresponding representations c^1 and c^2. The value of each $c_i^1 \in c^1$ (the same as c^1) in the code layer can be computed as follows,

$$c_i = \sigma\left(b_i + \sum_i x_i w_{ij}\right) \tag{1}$$

Step 3: run evolutionary search (e.g., SBX, DE, PM) on c^1 and c^2 to generate a new code c;

Step 4: map c back to the original space for getting the new solution y and the value of each $y_j \in y$ can be computed by

$$y_j = \sigma\left(b_j' + \sum_j c_i w_{ji}'\right) \tag{2}$$

where b and w are respectively the bias and the weight of this DAE with only one hidden layer, while σ represents the sigmoid function. In this work, $c = \Theta_{encode}(x)$ and $y = \Theta_{decode}(c)$ follow the similar process as introduced above.

The autoencoder is primarily employed to reconstruct explored non-dominated solutions, lacking the ability to enhance solution quality, thus falling short in accelerating the convergence of the evolutionary search. The innovization progress model is mainly designed for repairing newly generated solutions [Mittal *et al.*, 2021], and may not fully exploit the potential of evolutionary search. Moreover, their reliance on relatively large models necessitates a substantial amount of training data, which can be inefficient and less adaptable as the optimization progresses. Typically, they draw data from extensive past populations. However, as the optimization progresses, the promising directions of improvement change, and past populations may mislead model training. Therefore, contemporary populations often provide a more accurate reflection of the path towards optimal future solutions. Building upon these insights, this study aims to train a lightweight MLP model that effectively leverages the current population. This trained model is then iteratively stacked to create a larger model, with the goal of capturing improvement representations of solutions. Subsequently, an evolutionary search is conducted within this learned representation space to maximize the potential for discovering high-quality solutions.

3 Pseudocode for this to work

Here is the pseudo-code of the algorithms designed in this paper, including the general MOEA algorithm framework and our three proposed improved learnable MOEA algorithm frameworks.

Algorithm 1 is the basic MOEA algorithm framework. It differs from Algorithms 2-4 in the way solutions are generated.

Algorithm 2 corresponds to our proposed algorithms LNSGAV1, LNSGAV4, LMOEADV1, LMOEADV4.

Algorithm 3 corresponds to our proposed algorithms LNSGAV2, LNSGAV3, LNSGAV5, LMOEADV2, LMOEADV3, LMOEADV5. Among these variants, LNSGAV2 and LMOEADV2 utilize the learned compressed code space for searching, employing SBX and DE, respectively. Variants (LNSGAV1, LNSGAV3) and (LMOEADV1, LMOEADV3) leverage the learned improvement representation space for their search operations, employing SBX and DE, respectively. Finally, variants (LNSGAV4, LNSGAV5) and (LMOEADV4, LMOEADV5) utilize the learned deep improvement representation space for their search, employing SBX and DE, respectively. The detailed configuration is listed in Table 3.

Algorithm 4 corresponds to our proposed algorithms VLNSGA and VLMOEAD.

In this study, the process of using SBX to generate a child solution is as follows:

Step 1: Randomly select two different parent solutions: $x^1 = \left(x_1^1, \cdots, x_n^1\right)$ and $x^2 = \left(x_1^2, \cdots, x_n^2\right)$ from the current population P;

Step 2: generate a child solution $c = (c_1, \cdots, c_n)$, where c_i is computed as follows:

$$c_i = 0.5 \times \left[(1+\beta)\cdot x_i^1 + (1-\beta)\cdot x_i^2\right] \tag{3}$$

where β is dynamically computed as follows:

$$\beta = \begin{cases} (\text{ rand } \times 2)^{1/(1+\eta)} & \text{rand } \leq 0.5 \\ (1/(2 - \text{ rand } \times 2))^{1/(1+\eta)} & \text{otherwise.} \end{cases} \tag{4}$$

where η is a hyperparameter (the spread factor distribution index), which is set as 20. The greater the value of η, the greater the probability that the resulting child solution will be close to the parent.

The DE/rand/1 operator is used in this study. For each solution $x \in P$, the process of using DE to generate a child solution of x is as follows:

Step 1: Pick two solutions x^1 and x^2 from the population P at random, they must be distinct from each other as well as from the base vector x.

Step 2: The mutated individual v is obtained according to the following formula:

$$v_i = x_i + F(x_i^1 - x_i^2) \tag{5}$$

Step 3: The final individual c is obtained by crossover according to the following formula

$$c_i = \begin{cases} v_i & \text{if rand}_i[0,1] \leq CR \text{ or } i = k \\ x_i & \text{Otherwise} \end{cases} \tag{6}$$

In this study, we set $F = 0.5$ and $CR = 0.75$.

The $y = SBX(x^1, x^2)$ by the SBX operator and $y = DE(x^1, x^2, x^3)$ by the DE operator follow the process introduced above.

Algorithm 1: The general framework of an MOEA

Input: the LMOP with m objectives and n variables, the function evaluation budget FE_{max}
Output: the final population P to approximate the PF/PS
initialize P with N random solutions;
initialize the function evaluation counter $FE = 0$;
while $FE \leq FE_{max}$ **do**
    Q = Generator(P); *//evolutionary search in variable space to find new offspring solutions.*
    P = Selector(P, Q); *//environmental selection in objective space to filter poor solutions.*
    $FE = FE + N$;
end
return P

Algorithm 2: The general framework of the proposed LMOEA-V1

Input: the LMOP with m objectives and n variables, the function evaluation budget FE_{max}
Output: the final population P to approximate the PF/PS
initialize P with N random solutions;
initialize a set of N uniformly distributed reference vectors $R = (r_1, r_2, \ldots, r_N)$;
initialize the function evaluation counter $FE = 0$;
while $FE \leq FE_{max}$ **do**
    initialize an MLP model $M(A^*)$ with random parameters;
    D^* = TrainingDataPreparation(P, R); *//PBI subproblem-guided pairing of solutions.*
    update the parameters of $M(A^*)$ via backpropagation with gradient descent;
    for $i = 1$ *to* N **do**
        search in the original variable space to generate an offspring solution;
        if $rand > FE/FE_{max}$ **then**
            repair the new generated solution by $M(A^*)$ to be its improvement representation;
        end
        mutate the new solution by the polynomial mutation operator; add the new generated offspring solution into Q;
    end
    P = Selector(P, Q); *//environmental selection in the objective space.*
    $FE = FE + N$;
end
return P

4 Time-Varying Ratio Error Estimation

The precise estimation of voltage transformers' (VTs) ratio error (RE) holds significant importance in modern power delivery systems. Existing RE estimation methods predominantly revolve around periodic calibration, disregarding the time-varying aspect. This oversight presents challenges in achieving real-time VT state estimation. To address this concern, the formulation of a time-varying RE estimation (TREE) problem as a large-scale multiobjective optimization problem is proposed in [He *et al.*, 2020]. Multiple objectives and inequality constraints are defined based on statistical and physical rules extracted from power delivery systems. Additionally, a benchmark test suite is systematically created, encompassing various TREE problems from different substations to depict their distinct characteristics. This formulation not only transforms a costly RE estimation task into a more economical optimization problem but also contributes to the advancement of research in large-scale multiobjective optimization by providing a real-world benchmark test suite featuring intricate variable interactions and objective correlations. The source code for these optimization problems can be found on the PlatEMO.

5 Supplementary Experimental Studies

Due to space limitations, a supplement of some experimental data from this work is provided here. Mainly are the average IGD and HV results of each algorithm in solving the DTLZ and WFG problems with different settings.

Algorithm 3: The general framework of the proposed LMOEA-V2 to LMOEA-V5

```
Input: the LMOP with m objectives and n variables, the function evaluation budget FE_max
Output: the final population P to approximate the PF/PS
initialize P with N random solutions;
initialize a set of N uniformly distributed reference vectors R = (r_1, r_2, ..., r_N);
initialize the function evaluation counter FE = 0;
while FE ≤ FE_max do
    initialize an MLP model M(A*) with random parameters;
    D* = TrainingDataPreparation(P, R); //PBI subproblem-guided pairing of solutions.
    update the parameters of M(A*) via backpropagation with gradient descent;
    for i = 1 to N do
        if rand < FE/FE_max then
            search in the original variable space to generate an offspring solution;
        end
        else
            search in the learned representation space by M(A*) to generate a solution;
        end
        mutate the new solution by the polynomial mutation operator; add the new generated offspring solution into Q;
    end
    P = Selector(P, Q); //environmental selection in the objective space.
    FE = FE + N;
end
return P
```

Algorithm 4: The general framework of the proposed VLNSGA and VLMOEAD

```
Input: the LMOP with m objectives and n variables, the function evaluation budget FE_max
Output: the final population P to approximate the PF/PS
initialize P with N random solutions;
initialize a set of N uniformly distributed reference vectors R = (r_1, r_2, ..., r_N);
initialize the function evaluation counter FE = 0;
while FE ≤ FE_max do
    initialize an MLP model M(A*) with random parameters;
    D* = TrainingDataPreparation(P, R); //PBI subproblem-guided pairing of solutions.
    update the parameters of M(A*) via backpropagation with gradient descent;
    for i = 1 to N do
        if rand < FE/FE_max then
            search in the original variable space to generate an offspring solution z;
        end
        else
            if rand < 0.5 then
                search in the learned improvement representation space to generate a solution z;
            end
            else
                search in the learned compressed code space to generate a solution z;
            end
        end
        if rand < FE/FE_max then
            mutate the new solution by the polynomial mutation operator, i.e., y = PM(z);
        end
        else
            mutate the new solution by the y = Θ(z);
        end
        add the new generated offspring solution into Q;
    end
    P = Selector(P, Q); //environmental selection in the objective space.
    FE = FE + N;
end
return P
```

Table 1: The search and selection strategy configuration of our proposed algorithms

Algorithms	Evolutionary search of the Generator	Environmental selection
NSGA-II	SBX in the original space	Pareto-based selection
LNSGA-V1	SBX-based search in the original space followed by reparing part of offspring with MLP	Pareto-based selection
LNSGA-V2	SBX-based search in the original space + SBX in the compressed representation space	Pareto-based selection
LNSGA-V3	SBX-based search in the original space + SBX in the improvement representation space	Pareto-based selection
LNSGA-V4	SBX-based search in the original space followed by repairing part of offspring with stacked MLP	Pareto-based selection
LNSGA-V5	SBX-based search in the original space + SBX in the deep improvement representation space	Pareto-based selection
MOEA/D	DE in the original space	Decomposition-based selection
LMOEAD-V1	DE-based search in the original space followed by reparing part of offspring with MLP	Decomposition-based selection
LMOEAD-V2	DE-based search in the original space + DE in the compressed representation space	Decomposition-based selection
LMOEAD-V3	DE-based search in the original space + DE in the improvement representation space	Decomposition-based selection
LMOEAD-V4	DE-based search in the original space followed by repairing part of offspring with stacked MLP	Decomposition-based selection
LMOEAD-V5	DE-based search in the original space + DE in the deep improvement representation space	Decomposition-based selection

Table 2: Average IGD and HV results of NSGA-II and its five learnable variant versions on DTLZ1-4 with $m = 2, n = 1000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	NSGA-II	LNSGA-V1	LNSGA-V2	LNSGA-V3	LNSGA-V4	LNSGA-V5
IGD n=1000 m=2	DTLZ1	4.477e+3(5.3e+1)	**7.366e+0(9.6e+0)**	6.010e+1(5.8e+1)	4.143e+2(1.2e+2)	1.957e+1(2.0e+1)	1.913e+3(5.3e+3)
	DTLZ2	2.004e+0(1.8e-1)	1.291e-2(4.7e-2)	9.554e-2(3.2e-2)	4.820e-2(2.6e-2)	4.985e-3(7.8e-3)	**4.849e-3(3.0e-3)**
	DTLZ3	1.144e+4(3.9e+2)	**7.236e-1(7.6e-1)**	**5.786e-1(4.5e-1)**	1.767e+2(1.2e+2)	2.453e+2(7.4e+2)	1.258e+3(5.0e+3)
	DTLZ4	2.935e+0(1.2e-1)	1.652e-1(3.4e-1)	7.123e-1(2.9e-1)	8.984e-2(1.9e-1)	1.286e-2(2.6e-2)	**6.248e-3(8.4e-3)**
HV n=1000 m=2	DTLZ1	0.00e+0(0.0e+0)	**2.938e-1(3.0e-1)**	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)
	DTLZ2	0.00e+0(0.0e+0)	3.230e-1(1.6e-1)	1.830e-1(4.6e-2)	2.760e-1(1.4e-1)	3.429e-1(1.2e-1)	**3.457e-1(1.0e-1)**
	DTLZ3	0.00e+0(0.0e+0)	9.807e-2(4.4e-2)	**1.197e-1(6.3e-1)**	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)
	DTLZ4	0.00e+0(0.0e+0)	1.716e-1(1.2e-1)	9.751e-2(1.2e-1)	2.134e-1(1.4e-1)	1.659e-1(1.4e-1)	**3.046e-1(1.4e-1)**

Table 3: Average IGD and HV results of NSGA-II and its five learnable variant versions on DTLZ1-4 with $m = 2, n = 5000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	NSGA-II	LNSGA-V1	LNSGA-V2	LNSGA-V3	LNSGA-V4	LNSGA-V5
IGD n=5000 m=2	DTLZ1	7.274e+4(1.7e+3)	**1.823e+0(4.4e+0)**	5.372e+0(7.2e+0)	1.149e+2(9.3e+3)	1.647e+2(4.0e+3)	2.706e+2(3.3e+3)
	DTLZ2	1.514e+2(6.3e+0)	1.919e-2(3.2e-2)	7.376e+0(1.7e+1)	1.403e-2(2.9e-2)	1.017e-2(4.0e-2)	**8.503e-3(6.2e-3)**
	DTLZ3	1.931e+5(6.6e+3)	**3.447e+3(8.0e+3)**	**9.285e+3(2.0e+4)**	1.261e+4(2.5e+4)	4.499e+3(9.5e+3)	3.675e+3(8.4e+3)
	DTLZ4	1.712e+2(1.1e+1)	2.107e-1(2.9e-1)	2.799e-1(3.7e-1)	1.304e-1(2.0e-1)	1.980e-1(2.5e-1)	**7.511e-2(1.8e-1)**
HV n=5000 m=2	DTLZ1	0.00e+0(0.0e+0)	**4.607e-1(2.3e-1)**	1.916e-1(2.9e-1)	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)
	DTLZ2	0.00e+0(0.0e+0)	3.253e-1(4.6e-2)	2.302e-1(1.7e-1)	3.150e-1(1.7e-1)	3.264e-1(1.7e-1)	**3.465e-1(1.6e-1)**
	DTLZ3	0.00e+0(0.0e+0)	1.153e-1(1.7e-1)	**5.784e-2(1.4e-1)**	1.153e-1(1.7e-1)	9.413e-2(1.8e-1)	1.141e-1(1.3e-1)
	DTLZ4	0.00e+0(0.0e+0)	2.181e-1(1.4e-1)	2.198e-1(1.4e-1)	2.692e-1(1.8e-1)	2.472e-1(1.3e-1)	**2.851e-1(1.4e-1)**

Table 4: Average IGD and HV results of NSGA-II and its learnable variant versions on DTLZ1-4 with $m = 2, n = 10^4, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	NSGA-II	LNSGA-V1	LNSGA-V2	LNSGA-V3	LNSGA-V4	LNSGA-V5
IGD	DTLZ1	1.993e+5(3.8e+3)	**3.055e+1(6.8e+1)**	**2.068e+1(2.3e+1)**	7.163e+1(1.5e+2)	4.098e+1(9.7e+1)	4.805e+1(8.5e+1)
n=	DTLZ2	4.251e+2(8.1e+0)	8.651e-3(8.7e-3)	9.888e-3(6.2e-3)	1.481e-2(4.2e-2)	8.039e-3(8.2e-3)	**7.789e-3(6.2e-3)**
10000	DTLZ3	5.616e+5(8.7e+3)	**5.040e+0(1.1e+1)**	**6.319e+2(1.5e+3)**	1.415e+2(1.6e+2)	1.115e+3(2.7e+3)	2.994e+2(4.6e+3)
m=2	DTLZ4	4.414e+2(5.7e+0)	1.686e-1(2.9e-1)	2.682e-1(4.1e-1)	1.060e-1(2.5e-1)	6.481e-2(1.0e-2)	**1.045e-2(1.4e-2)**
HV	DTLZ1	0.00e+0(0.0e+0)	**1.934e-1(3.0e-1)**	8.803e-2(2.1e-1)	7.176e-2(1.7e-1)	**3.269e-1(2.8e-1)**	1.514e-1(2.4e-1)
n=	DTLZ2	0.00e+0(0.0e+0)	3.407e-1(1.3e-2)	3.425e-1(4.4e-3)	3.152e-1(1.9e-1)	3.385e-1(1.2e-2)	**3.438e-1(1.6e-1)**
10000	DTLZ3	0.00e+0(0.0e+0)	**1.762e-1(1.7e-1)**	**6.261e-2(1.4e-1)**	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)	5.786e-2(1.4e-1)
m=2	DTLZ4	0.00e+0(0.0e+0)	2.598e-1(1.3e-1)	2.306e-1(1.7e-1)	2.645e-1(1.1e-1)	2.820e-1(1.8e-1)	**3.076e-1(1.6e-1)**

Table 5: Average IGD and HV results of MOEA/D and its five learnable variant versions on DTLZ1-4 with $m = 2, n = 5000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	MOEA/D	LMOEADV1	LMOEADV2	LMOEADV3	LMOEADV4	LMOEADV5
IGD	DTLZ1	2.645e+4(8.4e+3)	**3.449e+0(7.5e+0)**	5.419e+2(1.3e+2)	1.126e+2(1.2e+2)	1.072e+2(2.6e+2)	5.725e+2(1.3e+3)
n=5000	DTLZ2	2.106e+1(4.1e+0)	9.524e-3(7.7e-4)	8.157e-3(1.8e-4)	1.151e-2(1.9e-2)	6.612e-3(6.4e-3)	**6.442e-3(4.9e-3)**
m=2	DTLZ3	7.016e+4(1.1e+4)	1.703e+2(4.1e+2)	**4.303e+1(5.3e+1)**	3.896e+3(4.5e+3)	2.942e+3(4.4e+3)	3.054e+3(4.7e+3)
	DTLZ4	1.433e+1(3.5e+0)	5.325e-1(6.6e-1)	5.145e+0(8.6e+0)	4.159e-1(1.7e+0)	3.105e-1(3.5e-1)	**2.053e-1(4.4e-1)**
HV	DTLZ1	0.00e+0(0.0e+0)	**2.036e-1(1.3e-1)**	0.00e+0(0.0e+0)	0.00e+0(0.0e+0)	9.560e-2(1.4e-1)	1.508e-2(3.7e-2)
n=5000	DTLZ2	0.00e+0(0.0e+0)	3.041e-1(1.6e-2)	3.149e-1(4.9e-2)	2.269e-1(1.7e-1)	**3.471e-1(8.4e-2)**	3.449e-1(1.7e-1)
m=2	DTLZ3	0.00e+0(0.0e+0)	**8.548e-2(6.2e-2)**	2.126e-2(5.1e-2)	0.00e+0(0.0e+0)	4.454e-2(6.4e-2)	0.00e+0(0.0e+0)
	DTLZ4	0.00e+0(0.0e+0)	1.996e-1(1.5e-1)	2.153e-1(1.7e-1)	2.670e-1(1.3e-1)	2.749e-1(1.4e-1)	**2.815e-1(1.2e-1)**

Table 6: Average IGD and HV results of MOEA/D and its five learnable variant versions on DTLZ1-4 with $m = 2, n = 10^4, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	MOEA/D	LMOEADV1	LMOEADV2	LMOEADV3	LMOEADV4	LMOEADV5
IGD	DTLZ1	5.122e+4(1.1e+4)	**1.692e-1(1.8e-2)**	1.670e+0(3.4e+0)	6.430e+1(1.2e+2)	1.294e+1(2.3e+1)	2.858e+1(3.1e+1)
n=	DTLZ2	4.815e+1(6.8e+0)	**5.108e-3(2.2e-4)**	6.582e-2(1.4e-1)	4.342e-2(1.0e-2)	2.127e-2(3.2e-1)	7.635e-3(9.5e-3)
10000	DTLZ3	1.517e+5(6.2e+4)	2.812e+1(6.7e+1)	3.389e+2(8.3e+2)	**2.201e+1(6.5e+1)**	6.353e+2(8.6e+2)	5.255e+2(8.2e+2)
m=2	DTLZ4	3.573e+1(3.3e+0)	3.554e-1(3.7e-1)	1.276e+0(1.2e+0)	2.646e-1(7.3e-1)	3.396e-1(4.9e-1)	**1.796e-1(4.0e-1)**
HV	DTLZ1	0.00e+0(0.0e+0)	**3.050e-1(1.7e-2)**	2.452e-1(2.9e-1)	1.434e-1(2.4e-1)	1.384e-1(1.5e-1)	1.909e-1(2.2e-1)
n=	DTLZ2	0.00e+0(0.0e+0)	**3.452e-1(2.3e-4)**	2.919e-1(1.2e-1)	2.840e-1(1.3e-1)	3.008e-1(1.7e-1)	3.229e-1(1.7e-1)
10000	DTLZ3	0.00e+0(0.0e+0)	1.062e-1(5.2e-2)	4.207e-2(6.5e-2)	**1.209e-1(1.2e-1)**	2.087e-2(5.1e-2)	6.192e-2(6.8e-2)
m=2	DTLZ4	0.00e+0(0.0e+0)	2.347e-1(1.3e-1)	1.278e-1(1.6e-1)	2.703e-1(1.3e-1)	2.360e-1(1.5e-1)	**2.713e-1(1.7e-1)**

Table 7: Average IGD results of LNSGAV4-5, LMOEADV4-5 and their five state-of-the-art LMOEA competitors on DTLZ1 to DTLZ7 with $m = 3, n \in (1000, 5000, 10000), FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Problems	n	CCGDE3	LMOCSO	DGEA	FDV	MOEAPSL	LNSGAV4	LNSGAV5	LMOEADV4	LMOEADV5
DTLZ1	1000	1.7217e+4(1.00e+3)	3.2312e+3(7.24e+2)	2.3344e+3(1.53e+3)	1.6134e+2(8.56e+0)	7.2991e+3(5.32e+1)	1.5451e+2(1.64e+2)	3.8213e+1(1.00e+2)	**3.2785e+1(5.45e+1)**	8.3810e+1(1.88e+2)
	5000	9.0465e+4(6.36e+3)	1.5374e+4(3.71e+3)	1.3147e+4(3.22e+3)	1.1790e+3(1.85e+2)	3.6479e+4(3.18e+2)	4.4894e-1(8.61e-1)	3.1124e-1(2.44e-1)	**2.3059e-1(5.23e-2)**	1.1470e+0(7.73e+0)
	10000	1.8414e+5(1.15e+4)	3.1759e+4(9.53e+3)	1.7238e+4(1.79e+4)	2.3949e+3(2.05e+2)	7.2611e+4(3.76e+2)	3.0943e-1(4.22e-1)	3.7318e+0(4.19e+0)	**2.0836e-1(9.05e-11)**	5.1845e+0(7.19e+0)
DTLZ2	1000	3.5489e+1(5.96e+0)	4.3729e+0(3.32e-1)	9.7221e+0(1.67e+0)	5.4715e+0(7.88e-1)	1.3163e+0(4.83e-1)	5.9351e-1(1.21e+0)	**1.0305e-1(4.80e-2)**	2.1142e-1(2.34e-1)	3.2014e-1(3.01e-1)
	5000	2.0355e+2(1.08e+1)	2.5686e+1(2.33e+0)	4.8258e+1(6.36e+0)	3.9753e+1(2.40e+0)	1.3535e+0(3.44e-1)	**6.2989e-2(2.07e-3)**	3.8928e+0(9.30e+0)	1.9077e+0(4.45e+0)	2.4242e+0(5.76e+0)
	10000	4.0705e+2(1.58e+1)	4.7210e+1(4.39e+0)	8.5965e+1(1.36e+1)	7.8984e+1(8.24e+0)	1.7844e+0(3.13e-1)	**6.1272e-2(3.48e-3)**	1.8469e-1(2.60e-1)	8.6939e-2(2.61e-2)	7.3235e-2(1.65e-2)
DTLZ3	1000	5.8810e+4(5.99e+3)	9.9866e+3(1.51e+3)	6.6308e+3(3.05e+3)	5.7306e+2(4.30e+1)	2.4370e+4(5.20e+2)	1.3516e+3(2.39e+3)	1.3609e+3(2.65e+3)	**2.6458e+2(4.31e+2)**	2.7038e+2(3.13e+2)
	5000	2.9041e+5(9.79e+3)	4.3509e+4(1.27e+4)	3.1891e+4(3.22e+4)	3.9859e+3(2.17e+2)	1.2453e+5(2.24e+2)	2.1861e+3(3.10e+3)	1.5139e+3(1.68e+3)	**6.1260e-1(2.00e-1)**	5.0168e+0(5.79e+0)
	10000	5.9058e+5(1.07e+4)	6.7224e+4(1.33e+4)	4.9663e+4(4.35e+4)	8.1286e+3(5.26e+2)	2.4917e+5(8.08e+2)	**2.7059e-1(4.77e-1)**	4.0025e-1(9.64e-2)	5.6419e-1(4.02e-3)	2.8193e-1(5.35e-1)
DTLZ4	1000	3.6751e+1(6.56e+0)	8.7989e+0(4.90e+0)	1.3120e+1(5.38e+0)	2.7978e+0(1.52e+0)	1.7943e+0(1.38e+0)	2.9849e-1(3.52e-1)	3.0249e-1(2.69e-1)	**1.2255e-1(5.39e-2)**	3.4578e-1(6.39e-1)
	5000	2.0298e+2(2.29e+1)	8.0450e+1(2.45e+1)	7.7825e+1(1.12e+1)	8.2570e+1(2.04e+1)	3.1396e+0(2.48e+0)	3.8238e-1(3.72e-1)	**8.4489e-2(2.91e-2)**	1.1712e-1(2.69e-2)	1.3242e-1(4.05e-2)
	10000	3.8442e+2(8.39e+0)	1.5148e+2(2.89e+1)	1.3450e+2(1.96e+1)	1.8845e+2(3.83e+1)	7.4236e+1(1.51e+2)	7.2282e-2(1.89e-2)	**6.3001e-2(5.25e-3)**	1.1355e-1(3.30e-2)	1.1971e-1(2.15e-2)
DTLZ5	1000	3.5126e+1(6.65e+0)	3.6676e+0(5.48e-1)	9.0203e+0(1.74e+0)	4.8166e+0(4.32e-1)	9.7212e-1(5.92e-1)	3.2991e-1(7.55e-1)	1.5579e-1(3.27e-1)	**2.2791e-2(2.55e-2)**	5.1728e-1(6.25e-1)
	5000	1.9598e+2(7.66e+0)	2.6307e+1(2.98e+0)	4.1106e+1(5.84e+0)	4.1093e+1(4.26e+0)	1.7579e+0(7.64e-1)	**6.3200e-3(3.44e-3)**	1.6563e-2(2.83e-2)	4.9498e-1(1.52e-1)	3.5783e+1(6.54e+0)
	10000	4.0984e+2(1.33e+1)	5.1802e+1(5.67e+0)	8.5696e+1(6.97e+0)	7.7378e+1(6.35e+0)	2.0465e+0(9.38e-1)	5.1561e-3(1.08e-3)	**4.5976e-3(5.98e-4)**	1.2996e-2(1.00e-2)	5.2693e-3(7.55e-4)
DTLZ6	1000	6.7254e+2(3.56e+1)	2.6919e+2(6.47e+1)	3.2103e+2(1.87e+2)	4.4008e+2(4.92e+1)	8.0847e-3(6.14e-3)	8.0427e-3(3.07e-3)	**3.5509e-3(1.61e-4)**	4.5964e-2(9.10e-3)	4.5587e-2(1.35e-3)
	5000	3.3458e+3(6.70e+1)	2.2042e+3(1.67e+2)	1.7773e+3(6.03e+2)	1.9879e+3(4.67e+2)	4.3210e-3(2.55e-3)	4.3188e-3(3.22e-3)	**3.4626e-3(1.61e-4)**	2.5415e-2(1.67e-2)	2.5420e-2(1.94e-2)
	10000	6.8175e+3(2.26e+2)	4.8254e+3(6.47e+1)	3.3037e+3(1.63e+3)	4.6517e+3(2.71e+2)	8.6914e-3(1.24e-3)	8.6981e-3(1.55e-3)	**3.4565e-3(6.14e-5)**	4.9226e-2(3.30e-3)	4.9357e-2(3.25e-2)
DTLZ7	1000	7.6884e+0(1.14e+0)	9.5440e+0(3.35e-1)	9.9196e+0(2.93e-1)	7.8822e+0(5.09e-1)	**1.1498e-1(1.15e-1)**	2.0752e-1(8.20e-2)	2.0377e-1(5.91e-2)	6.5215e-1(1.70e-1)	6.6534e-1(1.93e-1)
	5000	9.6590e+0(8.82e-1)	1.1118e+1(7.16e-2)	1.0916e+1(3.29e-2)	1.0017e+1(3.73e-1)	**2.5499e-1(1.44e-1)**	6.2057e-1(6.31e-2)	6.2603e-1(9.97e-2)	1.1193e+0(1.59e-1)	1.1121e+0(9.66e-2)
	10000	1.0748e+1(3.13e-2)	1.1279e+1(1.04e-1)	1.1170e+1(8.66e-2)	1.0490e+1(1.73e-1)	**4.0907e-1(3.34e-1)**	8.0830e-1(3.39e-2)	8.0574e-1(5.30e-2)	1.0420e+0(1.14e-1)	1.0410e+0(8.64e-2)

6 Future Research Directions

Enhancing Evolutionary Selectors or Discriminators Through Machine Learning: In the context of Many-Objective Optimization Problems (MaOPs), the application of machine learning techniques serves as a subtle yet powerful augmentation to the environmental selection process. This augmentation proves invaluable when confronted with the escalating complexity of objective spaces within MaOPs. As the number of objectives in MaOPs increases, the efficacy of traditional environmental selection strategies in distinguishing subpar solutions from elite ones diminishes significantly. More precisely, the convergent pressure of the discriminator falls short, and its ability to maintain solution diversity becomes inadequate. Consequently, it becomes imperative to explore methods for augmenting the discriminative capabilities of environmental selection strategies when tackling MaOPs.

Empowering Evolutionary Generators with Machine Learning: Within the realm of Large-Scale Multi-Objective Problems (LMOPs), the integration of machine learning techniques plays a strategic role in augmenting the evolutionary search process. This augmentation enables a dynamic response to the formidable challenges posed by the expansive search spaces characteristic of LMOPs. In such vast search spaces, the effectiveness of conventional genetic operators markedly declines, resulting in the unfortunate consequence of generating suboptimal offspring by the generator. Hence, it becomes imperative to delve into methods aimed at elevating the search prowess of these generators when tackling LMOPs.

Advancing Evolutionary Modules Through Learnable Transfer Techniques: In the realm of multi-objective optimization problems (MOPs), we introduce evolutionary modules employing transfer learning principles to facilitate the exchange of valuable optimization insights between source and target MOPs. In essence, this approach serves as a shortcut to solving target MOPs by leveraging the knowledge acquired from the optimization processes of related source problems. The optimization of source MOPs can be accomplished either concurrently with or prior to addressing the target MOPs, leading to two distinct forms of transfer optimization: sequential and multitasking. In the sequential form, the target MOPs are tackled one after another, benefiting from the cumulative wisdom gleaned from prior optimization exercises. This approach ensures that the experiences garnered from solving earlier problems are effectively applied to optimize subsequent ones. In contrast, the multitasking form involves the simultaneous optimization of all MOPs from the outset, with each problem drawing upon the knowledge cultivated during the optimization of other MOPs. This collaborative optimization approach maximizes the utility of learned knowledge, significantly enhancing the efficiency of solving multiple MOPs simultaneously.

References

[Gaur and Deb, 2017] Abhinav Gaur and Kalyanmoy Deb. Effect of size and order of variables in rules for multi-objective repair-based innovization procedure. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2177–2184. IEEE, 2017.

[He *et al.*, 2020] Cheng He, Ran Cheng, Chuanji Zhang, Ye Tian, Qin Chen, and Xin Yao. Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers. *IEEE Transactions on Evolutionary Computation*, 24(5):868–881, 2020.

[Liu, 2022] Songbai Liu. A survey on learnable evolutionary algorithms for scalable multiobjective optimization. *arXiv preprint arXiv:2206.11526*, 2022.

[Michalski, 2000] Ryszard S Michalski. Learning and evolution: An introduction to non-darwinian evolutionary computation. In *International Symposium on Methodologies for Intelligent Systems*, pages 21–30. Springer, 2000.

[Mittal *et al.*, 2020] Sukrit Mittal, Dhish Kumar Saxena, and Kalyanmoy Deb. Learning-based multi-objective optimization through ann-assisted online innovization. In *Proceedings of the 2020 genetic and evolutionary computation conference companion*, pages 171–172, 2020.

[Mittal *et al.*, 2021] Sukrit Mittal, Dhish Kumar Saxena, Kalyanmoy Deb, and Erik D Goodman. A learning-based innovized progress operator for faster convergence in evolutionary multi-objective optimization. *ACM Transactions on Evolutionary Learning and Optimization (TELO)*, 2(1):1–29, 2021.

Table 8: Average IGD and HV results of five learnable NSGA-II variant versions and VLNSGA on WFG1-9 with $m = 2, n = 1000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	LNSGA-V1	LNSGA-V2	LNSGA-V3	LNSGA-V4	LNSGA-V5	VLNSGA
IGD	WFG1	1.1862e+0(7.99e-3)	1.1861e+0(8.46e-3)	1.1999e+0(1.93e-2)	1.1956e+0(1.09e-2)	1.1925e+0(8.30e-3)	**1.0925e+0(1.11e-2)**
	WFG2	1.7520e-1(6.15e-3)	1.6920e-1(1.00e-2)	1.7422e-1(3.28e-3)	1.7366e-1(6.83e-3)	1.7517e-1(1.20e-2)	**1.6918e-1(9.16e-3)**
	WFG3	**2.0928e-1(3.80e-4)**	2.0950e-1(3.23e-4)	2.0959e-1(3.44e-4)	2.0946e-1(4.13e-4)	2.0951e-1(4.13e-4)	2.0936e-1(2.87e-4)
	WFG4	6.2603e-2(4.74e-4)	6.2299e-2(3.33e-4)	6.2779e-2(4.60e-4)	6.3031e-2(5.43e-4)	6.3152e-2(7.04e-4)	**6.1558e-2(5.36e-4)**
	WFG5	**1.7087e-1(8.94e-3)**	1.7385e-1(6.78e-3)	1.8153e-1(6.01e-3)	1.9364e-1(1.50e-2)	1.9233e-1(1.72e-2)	1.9261e-1(1.74e-2)
	WFG6	1.6213e-2(7.55e-4)	1.6294e-2(8.49e-4)	1.6471e-2(6.42e-4)	1.5812e-2(6.28e-4)	1.6077e-2(6.55e-4)	**1.5295e-2(8.98e-4)**
	WFG7	1.7152e-1(7.15e-3)	1.7604e-1(7.87e-3)	1.8720e-1(9.11e-3)	2.0474e-1(1.21e-2)	1.9033e-1(1.45e-2)	**1.6712e-1(6.13e-3)**
	WFG8	8.5441e-2(8.20e-3)	8.7716e-2(6.01e-3)	**8.1181e-2(7.62e-3)**	8.6814e-2(1.34e-2)	9.6029e-2(1.04e-2)	8.2051e-2(1.03e-2)
	WFG9	1.7627e-2(1.28e-3)	1.8022e-2(1.22e-3)	1.8549e-2(1.00e-3)	1.7812e-2(7.19e-4)	1.7607e-2(6.18e-4)	**1.7455e-2(7.94e-4)**
HV	WFG1	1.7831e-1(8.48e-3)	1.7739e-1(6.92e-3)	1.6669e-1(1.43e-2)	1.7204e-1(1.01e-2)	1.7507e-1(6.53e-3)	**1.7925e-1(8.91e-3)**
	WFG2	5.3182e-1(3.46e-3)	5.3518e-1(5.65e-3)	5.3234e-1(1.84e-3)	5.3267e-1(3.83e-3)	5.3185e-1(6.75e-3)	**5.3622e-1(5.13e-3)**
	WFG3	4.7804e-1(2.31e-4)	4.7793e-1(2.54e-4)	4.7794e-1(1.38e-4)	4.7796e-1(1.77e-4)	4.7799e-1(2.00e-4)	**4.7805e-1(2.22e-4)**
	WFG4	3.1433e-1(2.53e-4)	3.1455e-1(2.35e-4)	3.1429e-1(2.52e-4)	3.1419e-1(2.67e-4)	3.1412e-1(3.52e-4)	**3.1542e-1(1.16e-4)**
	WFG5	**2.5527e-1(4.77e-3)**	2.5373e-1(3.59e-3)	2.4964e-1(3.28e-3)	2.4337e-1(7.79e-3)	2.4399e-1(8.97e-3)	2.4383e-1(9.13e-3)
	WFG6	3.4510e-1(4.08e-4)	3.4524e-1(4.38e-4)	3.4506e-1(5.96e-4)	3.4514e-1(5.30e-4)	3.4502e-1(6.90e-4)	**3.4614e-1(2.47e-4)**
	WFG7	2.5487e-1(3.82e-3)	2.5236e-1(4.15e-3)	2.4651e-1(4.86e-3)	2.3723e-1(6.40e-3)	2.4483e-1(7.73e-3)	**2.5648e-1(3.18e-3)**
	WFG8	3.0204e-1(4.45e-3)	3.0059e-1(3.29e-3)	**3.0427e-1(4.06e-3)**	3.0153e-1(7.66e-3)	2.9633e-1(5.60e-3)	3.0046e-1(6.07e-3)
	WFG9	3.4363e-1(9.62e-4)	3.4328e-1(7.58e-4)	3.4324e-1(9.21e-4)	3.4345e-1(6.21e-4)	3.4354e-1(6.28e-4)	**3.4477e-1(4.96e-4)**

Table 9: Average IGD and HV results of five learnable NSGA-II variant versions and VLNSGA on WFG1-9 with $m = 2, n = 5000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	LNSGA-V1	LNSGA-V2	LNSGA-V3	LNSGA-V4	LNSGA-V5	VLNSGA
IGD	WFG1	1.2816e+0(8.14e-3)	1.2816e+0(8.79e-3)	1.2878e+0(7.81e-3)	1.2821e+0(1.39e-2)	1.2882e+0(1.14e-2)	**1.2715e+0(2.32e-2)**
	WFG2	2.0156e-1(8.14e-3)	2.0434e-1(7.07e-3)	1.9158e-1(5.21e-3)	1.9454e-1(8.18e-3)	2.1289e-1(2.08e-2)	**1.9071e-1(3.55e-3)**
	WFG3	2.0847e-1(4.75e-4)	2.0867e-1(4.02e-4)	**2.0770e-1(6.27e-4)**	2.0869e-1(7.27e-4)	2.0890e-1(6.09e-4)	2.0863e-1(4.86e-4)
	WFG4	6.1773e-2(2.24e-4)	6.1916e-2(6.06e-4)	6.1947e-2(4.66e-4)	6.1806e-2(4.48e-4)	6.1921e-2(4.03e-4)	**6.1722e-2(3.72e-4)**
	WFG5	3.2606e-1(6.63e-3)	3.2569e-1(4.85e-3)	3.3173e-1(4.65e-3)	3.4200e-1(9.31e-3)	**3.2497e-1(1.15e-2)**	3.3434e-1(3.45e-3)
	WFG6	1.5943e-2(5.40e-4)	1.5921e-2(7.61e-4)	1.5802e-2(7.40e-4)	1.5911e-2(4.07e-4)	1.6273e-2(1.07e-3)	**1.5055e-2(7.33e-4)**
	WFG7	2.4881e-1(3.71e-3)	2.4831e-1(6.51e-3)	2.5685e-1(4.40e-3)	2.6132e-1(7.48e-3)	2.5723e-1(6.81e-3)	**2.4267e-1(2.00e-3)**
	WFG8	**1.0440e-1(6.46e-3)**	1.1324e-1(7.39e-3)	1.0785e-1(9.61e-3)	1.1735e-1(1.54e-2)	1.1563e-1(1.63e-2)	1.1383e-1(4.25e-3)
	WFG9	1.7070e-2(1.22e-3)	1.7308e-2(1.23e-3)	1.7775e-2(1.68e-3)	1.7065e-2(1.22e-3)	1.6914e-2(7.72e-4)	**1.6345e-2(8.92e-4)**
HV	WFG1	1.5411e-1(7.00e-3)	1.5370e-1(7.73e-3)	1.4937e-1(6.95e-3)	1.5406e-1(1.22e-2)	1.4776e-1(9.89e-3)	**1.5723e-1(1.77e-2)**
	WFG2	5.1712e-1(4.55e-3)	5.1556e-1(3.93e-3)	5.2265e-1(2.89e-3)	5.2100e-1(4.59e-3)	5.1089e-1(1.14e-2)	**5.2594e-1(1.98e-3)**
	WFG3	**4.7852e-1(2.81e-4)**	4.7844e-1(2.11e-4)	4.7815e-1(3.92e-4)	4.7839e-1(4.36e-4)	4.7825e-1(3.55e-4)	4.7830e-1(3.90e-4)
	WFG4	3.1482e-1(1.71e-4)	3.1480e-1(2.29e-4)	3.1478e-1(2.66e-4)	3.1476e-1(2.60e-4)	3.1472e-1(1.57e-4)	**3.2175e-1(1.92e-4)**
	WFG5	1.7958e-1(2.84e-3)	**1.7981e-1(1.97e-3)**	1.7721e-1(2.07e-3)	1.7301e-1(3.73e-3)	1.7167e-1(4.39e-3)	1.7666e-1(1.28e-3)
	WFG6	3.4616e-1(2.71e-4)	3.4614e-1(2.32e-4)	3.4589e-1(4.14e-4)	3.4605e-1(3.71e-4)	3.4614e-1(5.07e-4)	**3.4909e-1(3.77e-4)**
	WFG7	2.1432e-1(1.90e-3)	2.1441e-1(3.31e-3)	2.1041e-1(2.17e-3)	2.0823e-1(3.61e-3)	2.1026e-1(3.34e-3)	**2.2241e-1(1.04e-3)**
	WFG8	2.8346e-1(3.54e-3)	2.8629e-1(4.02e-3)	**2.8938e-1(5.27e-3)**	2.8421e-1(8.20e-3)	2.8519e-1(8.94e-3)	2.8588e-1(2.22e-3)
	WFG9	3.4465e-1(5.48e-4)	3.4433e-1(7.23e-4)	3.4458e-1(3.10e-4)	3.4471e-1(5.27e-4)	3.4484e-1(4.66e-4)	**3.4978e-1(2.78e-4)**

Table 10: Average IGD and HV results of five learnable MOEA/D variant versions and VLMOEAD on WFG1-9 with $m = 2, n = 1000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	LMOEAD-V1	LMOEAD-V2	LMOEAD-V3	LMOEAD-V4	LMOEAD-V5	VLMOEAD
IGD	WFG1	1.2774e+0(5.29e-3)	1.2768e+0(4.08e-3)	1.2771e+0(5.33e-3)	1.2760e+0(5.39e-3)	1.2755e+0(6.21e-3)	**1.2673e+0(4.19e-3)**
	WFG2	1.9618e-1(6.64e-3)	1.9942e-1(4.76e-3)	1.9577e-1(3.10e-3)	2.0393e-1(6.65e-3)	2.0398e-1(9.05e-3)	**1.9368e-1(3.45e-3)**
	WFG3	2.0702e-1(8.68e-5)	2.0694e-1(1.17e-4)	2.0699e-1(7.28e-5)	2.0701e-1(7.25e-5)	2.0696e-1(1.14e-4)	2.0576e-1(5.65e-5)
	WFG4	7.7038e-2(3.46e-3)	7.8502e-2(5.51e-3)	7.8763e-2(4.40e-3)	7.7892e-2(3.04e-3)	7.9284e-2(5.34e-3)	**7.5692e-2(2.17e-3)**
	WFG5	6.8097e-2(2.53e-3)	**6.7094e-2(6.31e-4)**	6.7786e-2(1.15e-3)	6.7505e-2(8.92e-4)	6.7525e-2(1.98e-3)	6.7407e-2(2.01e-3)
	WFG6	1.7445e-2(6.02e-4)	1.7620e-2(1.49e-4)	1.7711e-2(5.89e-4)	1.7584e-2(5.15e-4)	1.8622e-2(6.09e-4)	**1.7404e-2(3.83e-4)**
	WFG7	2.0861e-1(2.82e-3)	2.1024e-1(2.25e-3)	2.0811e-1(2.69e-3)	2.1004e-1(2.78e-3)	2.1401e-1(3.48e-3)	**2.0369e-1(2.32e-3)**
	WFG8	6.7560e-2(4.82e-3)	6.9462e-2(5.57e-3)	**6.5243e-2(4.50e-3)**	7.3786e-2(8.12e-3)	8.4424e-2(1.01e-2)	7.1542e-2(4.87e-3)
	WFG9	2.1445e-2(1.04e-3)	2.1757e-2(7.29e-4)	2.2399e-2(2.10e-3)	2.1878e-2(2.07e-3)	2.2767e-2(8.20e-4)	**2.1389e-2(4.84e-4)**
HV	WFG1	1.5741e-1(3.40e-3)	1.5778e-1(3.63e-3)	1.5681e-1(2.96e-3)	1.5809e-1(3.64e-3)	1.5712e-1(3.71e-3)	**1.6250e-1(2.22e-3)**
	WFG2	5.1908e-1(3.82e-3)	5.1706e-1(2.69e-3)	5.1905e-1(1.77e-3)	5.1445e-1(3.81e-3)	5.1444e-1(5.14e-3)	**5.2551e-1(2.30e-3)**
	WFG3	4.7968e-1(8.47e-5)	4.7971e-1(8.66e-5)	4.7967e-1(7.47e-5)	4.7967e-1(7.54e-5)	4.7969e-1(1.26e-4)	**4.8072e-1(6.49e-5)**
	WFG4	3.0622e-1(1.97e-3)	3.0560e-1(2.99e-3)	3.0570e-1(2.65e-3)	3.0609e-1(1.84e-3)	3.0544e-1(2.84e-3)	**3.1027e-1(1.48e-3)**
	WFG5	3.1046e-1(2.32e-3)	**3.1151e-1(6.82e-4)**	3.1085e-1(1.25e-3)	3.1067e-1(9.53e-4)	3.1093e-1(1.91e-3)	3.1085e-1(1.90e-3)
	WFG6	3.4297e-1(3.04e-4)	3.4298e-1(2.03e-4)	3.4300e-1(2.27e-4)	3.4296e-1(2.12e-4)	3.4213e-1(3.62e-4)	**3.4683e-1(1.53e-4)**
	WFG7	2.3562e-1(1.53e-3)	2.3480e-1(1.21e-3)	2.3591e-1(1.38e-3)	2.3485e-1(1.29e-3)	2.3287e-1(1.74e-3)	**2.4305e-1(1.14e-3)**
	WFG8	3.1159e-1(2.86e-3)	3.1061e-1(3.10e-3)	**3.1294e-1(2.44e-3)**	3.0818e-1(4.52e-3)	3.0233e-1(5.69e-3)	3.0929e-1(2.85e-3)
	WFG9	3.3951e-1(8.43e-4)	3.3943e-1(3.69e-4)	3.3896e-1(1.45e-3)	3.3921e-1(1.53e-3)	3.3890e-1(7.06e-4)	**3.4246e-1(4.12e-4)**

Table 11: Average IGD and HV results of five learnable MOEA/D variant versions and VLMOEAD on WFG1-9 with $m = 2, n = 5000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	LMOEAD-V1	LMOEAD-V2	LMOEAD-V3	LMOEAD-V4	LMOEAD-V5	VLMOEAD
IGD	WFG1	1.2943e+0(5.02e-3)	1.2906e+0(4.30e-3)	1.2944e+0(3.48e-3)	1.2925e+0(5.57e-3)	1.2913e+0(3.06e-3)	**1.2012e+0(4.89e-3)**
	WFG2	2.1435e-1(5.35e-3)	2.0907e-1(5.54e-3)	2.0327e-1(7.76e-3)	2.1053e-1(6.86e-3)	2.1863e-1(1.70e-2)	**2.0573e-1(5.09e-3)**
	WFG3	2.0702e-1(2.57e-5)	2.0702e-1(3.58e-5)	2.0701e-1(1.72e-5)	2.0703e-1(3.98e-5)	2.0701e-1(3.94e-5)	2.0703e-1(1.81e-5)
	WFG4	6.9476e-2(1.89e-3)	6.8713e-2(1.04e-3)	6.9658e-2(1.60e-3)	6.9266e-2(1.26e-3)	6.9513e-2(1.06e-3)	**6.7862e-2(1.71e-3)**
	WFG5	6.6721e-2(1.59e-3)	6.5940e-2(4.81e-4)	6.6194e-2(8.18e-4)	**6.5931e-2(4.97e-4)**	6.6975e-2(1.78e-3)	6.6807e-2(9.36e-4)
	WFG6	1.5831e-2(4.40e-4)	1.5863e-2(5.14e-4)	1.6030e-2(4.69e-4)	1.5832e-2(3.43e-4)	1.6703e-2(7.86e-4)	**1.5243e-2(4.01e-4)**
	WFG7	2.6543e-1(1.02e-3)	2.6553e-1(1.17e-3)	2.6471e-1(8.45e-4)	2.6637e-1(9.04e-4)	2.6779e-1(1.21e-3)	**2.5630e-1(1.04e-3)**
	WFG8	8.5191e-2(4.51e-3)	9.0146e-2(6.66e-3)	**8.2585e-2(5.66e-3)**	9.2711e-2(5.68e-3)	9.4459e-2(5.31e-3)	8.9001e-2(4.75e-3)
	WFG9	1.8053e-2(7.80e-4)	1.7533e-2(7.65e-4)	1.8042e-2(1.03e-3)	1.7852e-2(7.66e-4)	1.8640e-2(6.95e-4)	**1.6166e-2(1.96e-3)**
HV	WFG1	1.5332e-1(4.21e-3)	1.5582e-1(3.20e-3)	1.5299e-1(2.40e-3)	1.5410e-1(4.21e-3)	1.5518e-1(2.17e-3)	**1.6023e-1(3.33e-3)**
	WFG2	5.0865e-1(3.04e-3)	5.1177e-1(3.02e-3)	5.1505e-1(4.37e-3)	5.1114e-1(3.85e-3)	5.0658e-1(9.19e-3)	**5.2933e-1(2.77e-3)**
	WFG3	4.7968e-1(6.35e-5)	4.7968e-1(9.69e-5)	4.7971e-1(8.60e-5)	4.7966e-1(1.21e-4)	4.7971e-1(7.29e-5)	**4.8066e-1(7.06e-5)**
	WFG4	3.1050e-1(1.15e-3)	3.1088e-1(6.95e-4)	3.1065e-1(9.22e-4)	3.1070e-1(6.55e-4)	3.1047e-1(5.43e-4)	**3.1201e-1(9.29e-4)**
	WFG5	3.1147e-1(1.57e-3)	3.1203e-1(6.51e-4)	3.1193e-1(9.81e-4)	**3.1220e-1(4.63e-4)**	3.1128e-1(1.60e-3)	3.1107e-1(1.05e-3)
	WFG6	3.4431e-1(1.09e-4)	3.4425e-1(2.67e-4)	3.4423e-1(2.23e-4)	3.4445e-1(1.82e-4)	3.4379e-1(2.50e-4)	**3.4619e-1(2.24e-4)**
	WFG7	2.0655e-1(4.76e-4)	2.0648e-1(5.74e-4)	2.0686e-1(3.97e-4)	2.0605e-1(4.27e-4)	2.0538e-1(6.06e-4)	**2.1204e-1(5.36e-4)**
	WFG8	3.0177e-1(2.46e-3)	2.9900e-1(3.69e-3)	**3.0329e-1(3.13e-3)**	2.9755e-1(3.12e-3)	2.9658e-1(2.84e-3)	2.9953e-1(2.60e-3)
	WFG9	3.4217e-1(5.61e-4)	3.4221e-1(7.63e-4)	3.4184e-1(7.76e-4)	3.4217e-1(8.74e-4)	3.4173e-1(6.10e-4)	**3.5064e-1(1.55e-3)**

Table 12: Average IGD and HV results of five learnable MOEA/D variant versions and VLMOEAD on WFG1-9 with $m = 2, n = 10000, FE_{max} = 10^5$. The standard deviation indicated in parentheses following.

Metric	Problem	LMOEAD-V1	LMOEAD-V2	LMOEAD-V3	LMOEAD-V4	LMOEAD-V5	VLMOEAD
IGD	WFG1	1.2943e+0(5.02e-3)	1.2906e+0(4.30e-3)	1.2944e+0(3.48e-3)	1.2925e+0(5.57e-3)	1.2913e+0(3.06e-3)	**1.2012e+0(4.89e-3)**
	WFG2	2.1435e-1(5.35e-3)	2.0907e-1(5.54e-3)	2.0327e-1(7.76e-3)	2.1053e-1(6.86e-3)	2.1863e-1(1.70e-2)	**2.0573e-1(5.09e-3)**
	WFG3	2.0702e-1(2.57e-5)	2.0702e-1(3.58e-5)	2.0701e-1(1.72e-5)	2.0703e-1(3.98e-5)	2.0701e-1(3.94e-5)	2.0703e-1(1.81e-5)
	WFG4	6.9476e-2(1.89e-3)	6.8713e-2(1.04e-3)	6.9658e-2(1.60e-3)	6.9266e-2(1.26e-3)	6.9513e-2(1.06e-3)	**6.7862e-2(1.71e-3)**
	WFG5	6.6721e-2(1.59e-3)	6.5940e-2(4.81e-4)	6.6194e-2(8.18e-4)	**6.5931e-2(4.97e-4)**	6.6975e-2(1.78e-3)	6.6807e-2(9.36e-4)
	WFG6	1.5831e-2(4.40e-4)	1.5863e-2(5.14e-4)	1.6030e-2(4.69e-4)	1.5832e-2(3.43e-4)	1.6703e-2(7.86e-4)	**1.5243e-2(4.01e-4)**
	WFG7	2.6543e-1(1.02e-3)	2.6553e-1(1.17e-3)	2.6471e-1(8.45e-4)	2.6637e-1(9.04e-4)	2.6779e-1(1.21e-3)	**2.5630e-1(1.04e-3)**
	WFG8	8.5191e-2(4.51e-3)	9.0146e-2(6.66e-3)	**8.2585e-2(5.66e-3)**	9.2711e-2(5.68e-3)	9.4459e-2(5.31e-3)	8.9001e-2(4.75e-3)
	WFG9	1.8053e-2(7.80e-4)	1.7533e-2(7.65e-4)	1.8042e-2(1.03e-3)	1.7852e-2(7.66e-4)	1.8640e-2(6.95e-4)	**1.6166e-2(1.96e-3)**
HV	WFG1	1.5332e-1(4.21e-3)	1.5582e-1(3.20e-3)	1.5299e-1(2.40e-3)	1.5410e-1(4.21e-3)	1.5518e-1(2.17e-3)	**1.6023e-1(3.33e-3)**
	WFG2	5.0865e-1(3.04e-3)	5.1177e-1(3.02e-3)	5.1505e-1(4.37e-3)	5.1114e-1(3.85e-3)	5.0658e-1(9.19e-3)	**5.2933e-1(2.77e-3)**
	WFG3	4.7968e-1(6.35e-5)	4.7968e-1(9.69e-5)	4.7971e-1(8.60e-5)	4.7966e-1(1.21e-4)	4.7971e-1(7.29e-5)	**4.8066e-1(7.06e-5)**
	WFG4	3.1050e-1(1.15e-3)	3.1088e-1(6.95e-4)	3.1065e-1(9.22e-4)	3.1070e-1(6.55e-4)	3.1047e-1(5.43e-4)	**3.1201e-1(9.29e-4)**
	WFG5	3.1147e-1(1.57e-3)	3.1203e-1(6.51e-4)	3.1193e-1(9.81e-4)	**3.1220e-1(4.63e-4)**	3.1128e-1(1.60e-3)	3.1107e-1(1.05e-3)
	WFG6	3.4431e-1(1.09e-4)	3.4425e-1(2.67e-4)	3.4423e-1(2.23e-4)	3.4445e-1(1.82e-4)	3.4379e-1(2.50e-4)	**3.4619e-1(2.24e-4)**
	WFG7	2.0655e-1(4.76e-4)	2.0648e-1(5.74e-4)	2.0686e-1(3.97e-4)	2.0605e-1(4.27e-4)	2.0538e-1(6.06e-4)	**2.1204e-1(5.36e-4)**
	WFG8	3.0177e-1(2.46e-3)	2.9900e-1(3.69e-3)	**3.0329e-1(3.13e-3)**	2.9755e-1(3.12e-3)	2.9658e-1(2.84e-3)	2.9953e-1(2.60e-3)
	WFG9	3.4217e-1(5.61e-4)	3.4221e-1(7.63e-4)	3.4184e-1(7.76e-4)	3.4217e-1(8.74e-4)	3.4173e-1(6.10e-4)	**3.5064e-1(1.55e-3)**