

Paper Type: Original Article

An Adaptive Neuro-Metaheuristic Framework Integrating Transformer-Based Surrogate Modeling with Differential Evolution for High-Dimensional Black-Box Optimization

Zinah Subhi¹ , Maher Hussein²  and Lubna Thanoon ALkahla^{3,*} 

¹ College of Medicine, Ninevah university, Mosul, Iraq; zina.subhi@uoninevah.edu.iq.

² Computer Center, Presidency of the University, Telafer University, Mosul, Iraq; maher.k.hussein@uotelafer.edu.iq.

³ Department of Artificial Intelligence, Information Technology College, Ninevah University, Mosul, Iraq; lubna.thanoon@uoninevah.edu.iq.

Received: 01 Apr 2026

Revised: 27 Jun 2026

Accepted: 21 Sep 2026

Published: 23 Sep 2026

Abstract

High-dimensional black-box optimization is a highly challenging problem, especially when each function evaluation is expensive, the objective is non-differentiable, and almost nothing is known about its structure. In this setting, traditional gradient-based methods are not available, and most existing metaheuristics either stagnate far from truly competitive solutions or require an impractically large number of evaluations once the dimension exceeds about $d = 100$. This paper directly addresses this challenge by proposing the Adaptive Neuro-Metaheuristic with Transformer-based Differential Evolution (ANM-TDE), which combines a Transformer-based surrogate model with a self-adaptive Differential Evolution (DE) algorithm to greatly reduce the number of real function evaluations while maintaining strong global search performance. The surrogate model is incrementally trained on all previously evaluated candidates and uses multi-head self-attention to predict their fitness, while the DE engine consults the surrogate to pre-screen trial solutions before paying the cost of real evaluations. We prove that the expected difference between the surrogate-predicted fitness and the true fitness decays at the rate $\mathcal{O}(N^{-1/2})$ as the archive size N increases, and that the overall method converges to a stationary point at the rate $\mathcal{O}(1/K)$ in the nonconvex setting, assuming an L -smooth objective and a bounded surrogate approximation error. On 12 benchmark problems with dimensions from $d = 30$ to $d = 500$, including Sphere, Rosenbrock, Rastrigin, Ackley, Schwefel, and Levy, ANM-TDE reduces the number of real function evaluations by 67–73% compared with standard DE, while achieving a mean best fitness of 1.47×10^{-12} on F1 (Sphere, $d = 100$), versus 8.93×10^{-8} for CMA-ES. Wilcoxon signed-rank tests show statistically significant improvements on 10 out of 12 benchmarks. The results show that ANM-TDE is a competitive algorithm for expensive black-box optimization problems such as engineering design, hyperparameter tuning, and scientific simulation.

Keywords: Surrogate-assisted Optimization; Transformer Neural Networks; Differential Evolution; High-dimensional Black-Box Optimization; Bayesian Optimization; Metaheuristic Algorithms; Gaussian Process Regression.



Corresponding Author: lubna.thanoon@uoninevah.edu.iq



<https://doi.org/10.61356/j.nois.2026.11673>



Licensee **Neutrosophic Optimization and Intelligent systems**. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

1 | Introduction

Consider a continuous function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ about which nothing else is known and for which no gradients are available, so that classical gradient-based and convex-programming techniques [1, 2] cannot be applied directly. Every evaluation of f takes minutes or even hours, and the dimension is large ($d \geq 100$). The goal is to find $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \Omega} f(\mathbf{x})$, where $\Omega \subseteq \mathbb{R}^d$ is the feasible set, using as few evaluations of f as possible; this is known as the black-box optimization problem. It appears in numerous practical applications, such as aerodynamic shape design [3], the search for high-performance neural networks [4], parameter tuning for large-scale simulators [5], and chemical synthesis and materials discovery [6]. No algorithm is known to solve the problem in polynomial time in the general case; researchers in the field are therefore typically concerned with heuristics and with average-case rather than worst-case performance.

Three classes of algorithms exist for this problem. The first is Bayesian optimization (BO) [7-9]. BO fits a Gaussian process (GP) model to the observations collected so far and then chooses the next evaluation location by maximizing an acquisition function, such as expected improvement or upper confidence bound [5, 9]. BO requires only a few evaluations when the search space is small, but it does not scale well to large problems, because GP inference scales cubically, $\mathcal{O}(N^3)$, with the number N of stored evaluations, and optimizing the acquisition function becomes challenging once the dimensionality of the search space exceeds a few tens of variables [8, 10].

Between 2020 and 2024, several studies attempted to extend BO through sparse GPs [7], trust-region decompositions [8], parallel multi-fidelity schemes [10], and latent-space projections [9]; however, the fundamental cubic scaling has not been eliminated without sacrificing model fidelity. The second class comprises population-based metaheuristics, such as DE [11] and its adaptive variants [12-14], CMA-ES [15], and swarm-inspired methods such as Harris hawks optimization [16]; these methods are robust but typically require very large evaluation budgets. The third class, evolutionary surrogate-assisted methods [17-19], attaches lightweight regression surrogates (radial basis functions, polynomial response surfaces, neural networks) to population-based optimizers: the surrogate pre-screens candidate solutions, and only the most promising subset is evaluated on the true function. This strategy also works in high-dimensional spaces, but the surrogate is typically fixed and is usually a shallow feed-forward network. Such a model cannot cope with complex, multimodal fitness functions once the dimensionality exceeds 200, because it is limited by its architecture. Previous studies have applied Transformers [20] to optimization, for example as meta-learners and in neural architecture search [4], thus providing a basis for further research. To the best of our knowledge, this is the first work that integrates a Transformer-based surrogate model with adaptive differential evolution and provides the corresponding convergence guarantees.

Another important context for ANM-TDE is its relation to other recent surrogate-assisted evolutionary methods. Guo et al. [18] introduced a hybrid that combines a surrogate model with an estimation-of-distribution algorithm and local search, demonstrating strong performance on a collection of expensive optimization benchmarks; however, their algorithm uses a shallow regression model instead of an attention-based architecture and does not provide a formal convergence analysis. The overview by Jin et al. [17] and the survey by Chugh et al. [19] summarize the literature on surrogate-assisted evolutionary optimization and suggest that self-attention-based surrogates have been relatively underexplored in this context. ANM-TDE differs from these approaches in its use of self-attention to model interactions across the entire population (rather than evaluating each candidate independently) and in the incorporation of the resulting surrogate into an explicit, theoretically grounded pre-screening rule. Direct empirical comparisons with these recent methods are therefore an important area of future research, as discussed further in Section 6.

The research gap is therefore specific: surrogate-assisted DE in high dimensions ($d \geq 100$) with (i) a surrogate that captures long-range fitness structure via self-attention, (ii) an adaptive archive-management policy that prevents surrogate over-confidence, (iii) a formal convergence analysis connecting the surrogate

approximation error to the overall optimization error, and (iv) statistical benchmarking against contemporary DE variants. This paper addresses all four requirements.

Our contributions are as follows:

Contribution 1: Transformer Surrogate for Fitness Prediction. We construct a compact Transformer encoder [20] that takes the current population matrix as input, treating each candidate as a d -dimensional token in the sequence, and returns a fitness prediction together with a calibrated uncertainty estimate for each candidate using Monte Carlo dropout. The resulting inference cost is $\mathcal{O}(N \cdot d^2)$, dominated by the attention mechanism rather than by GP matrix inversion, which allows efficient online updates.

Contribution 2: Surrogate-Gated Differential Evolution. We embed the Transformer surrogate into the L-SHADE adaptive DE framework [12] by reformulating the trial-vector acceptance step as a two-phase, surrogate-based pre-screening oracle. In particular, any trial vector whose surrogate-predicted fitness is worse than the current best by more than a threshold δ^k is rejected outright and never forwarded to an expensive true evaluation.

Contribution 3: Convergence Analysis. Lemma 4.1 shows that the expected surrogate approximation error is bounded, and, building on this result, Theorem 4.1 shows that ANM-TDE converges to an ε -stationary point at the rate $\mathcal{O}(1/K)$, where K is the number of iterations, while explicitly accounting for the surrogate error under standard smoothness assumptions [21, 22].

Contribution 4: Statistical Benchmarking. A head-to-head comparison between ANM-TDE and CMA-ES [15], L-SHADE [12], LSHADE-RSP [13], and SaDE [14] is performed on 12 benchmark functions, including functions from the CEC 2017 suite [23], for $d \in \{30, 100, 300, 500\}$, and statistical significance and relative performance are assessed with Wilcoxon signed-rank tests and Friedman rankings.

The rest of the paper is organized as follows: in Section 2, we present the mathematical background, problem statement, and notations, and in Section 3, we describe the full ANM-TDE algorithm, while in Section 4, we present the convergence analysis, and in Section 5, we present the numerical experiments and benchmarking, and furthermore, in Section 6, we discuss limitations and sensitivity, and finally, we conclude this paper in Section 7.

2 | Mathematical Background and Problem Formulation

2.1 | Problem Statement

Let $\Omega = \prod_{j=1}^d [\ell_j, u_j] \subseteq \mathbb{R}^d$ be a compact feasible domain defined by the lower and upper bound vectors $\boldsymbol{\ell}, \mathbf{u} \in \mathbb{R}^d$. We consider the bound-constrained black-box problem

$$\min_{\mathbf{x} \in \Omega \subseteq \mathbb{R}^d} f(\mathbf{x}) \quad (1)$$

where $f: \Omega \rightarrow \mathbb{R}$ is continuous but has no available gradient and no closed-form expression, and each evaluation costs τ seconds with $\tau \gg 0$. The only feedback is the scalar $f(\mathbf{x})$ returned after querying the true simulator or physical experiment. The budget is a maximum of $T_{\max}$ true function evaluations, where $T_{\max}$ is of the order of 10^3 – 10^4 in the applications targeted here.

2.2 | Notation

Scalars are denoted by italic lowercase letters (f, k, d), vectors by bold lowercase letters ($\mathbf{x}, \mathbf{u}, \mathbf{v}$), and matrices by bold uppercase letters ($\mathbf{P}, \mathbf{W}$). The problem dimension is denoted by d . The archive of evaluated candidates at iteration k is $\mathcal{A}^k = \left\{ \left(\mathbf{x}^i, f(\mathbf{x}^i) \right) \right\}_{i=1}^{N_k}$, with $|\mathcal{A}^k| = N_k$. The surrogate model is denoted by

$\hat{f}_\theta: \mathbb{R}^d \rightarrow \mathbb{R}$ and is parameterized by $\theta \in \mathbb{R}^q$. We write $\|\cdot\|$ for the Euclidean norm unless subscripted, and $\mathbb{E}[\cdot]$ for the expectation.

2.3 | Differential Evolution

Standard DE [11] maintains a population $\mathbf{P} = \{\mathbf{x}_i\}_{i=1}^{NP} \subseteq \Omega$. At each generation g , for each target vector $\mathbf{x}_i$, a mutant vector $\mathbf{v}_i$ is formed by the mutation operator in Eq. (2):

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), \quad r_1 \neq r_2 \neq r_3 \neq i, \quad F \in (0,2] \quad (2)$$

A trial vector $\mathbf{u}_i$ is then generated by binomial crossover with probability $CR \in [0,1]$, and the trial replaces the target if $f(\mathbf{u}_i) \leq f(\mathbf{x}_i)$. L-SHADE [12] adapts F and CR from a historical memory H of size M_H . ANM-TDE replaces the direct evaluation $f(\mathbf{u}_i)$ with a surrogate pre-screen $\hat{f}_\theta(\mathbf{u}_i)$, reducing true evaluations at the cost of surrogate-modeling overhead.

2.4 | Transformer Architecture Overview

Given a population matrix $\mathbf{P} \in \mathbb{R}^{NP \times d}$, the Transformer surrogate treats each row $\mathbf{x}_i \in \mathbb{R}^d$ as a token after the linear embedding $\mathbf{z}_i = \mathbf{W}_E \mathbf{x}_i + \mathbf{b}_E$, where $\mathbf{W}_E \in \mathbb{R}^{d_{\text{model}} \times d}$. Multi-head self-attention [20] then models inter-candidate interactions according to Eq. (3):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \quad (3)$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{NP \times d_k}$ are the query, key, and value matrices projected from the embedded population. A regression head $\varphi: \mathbb{R}^{d_{\text{model}}} \rightarrow \mathbb{R}$ maps the per-token output to a scalar fitness prediction. The full forward pass costs $\mathcal{O}(NP^2 \cdot d_{\text{model}} + NP \cdot d_{\text{model}}^2)$ per generation, which is substantially cheaper than GP inference at $\mathcal{O}(N^3)$ for large archives.

Four assumptions underpin the theoretical analysis:

Assumption 1. The true objective function f is L -smooth [2, 21]: for all $\mathbf{x}, \mathbf{y} \in \Omega$, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, with $L < \infty$.

Assumption 2. The surrogate approximation error is bounded in expectation: $\varepsilon_s(N) = \mathbb{E}[|f(\mathbf{x}) - \hat{f}_\theta(\mathbf{x})|] \leq C \cdot N^{-1/2}$ for a constant $C > 0$ depending on the function class and on d_{model} .

Assumption 3. The population is initialized by Latin Hypercube Sampling over Ω , ensuring uniform coverage with probability at least $1 - \gamma$ for any $\gamma \in (0,1)$ and sufficiently large NP .

Assumption 4. The surrogate pre-screening threshold satisfies $\delta^k \geq 2\varepsilon_s(N_k)$, which prevents the rejection of true optima due to surrogate error.

3 | The ANM-TDE Algorithm

3.1 | Framework Design

ANM-TDE consists of three nested loops. The outermost loop updates the archive $\mathcal{A}$ and re-trains the surrogate every τ_{retrain} generations. The intermediate loop is the standard L-SHADE DE [12], which creates NP trial vectors per generation. The innermost loop is the surrogate oracle: before the true objective is queried, each trial vector $\mathbf{u}_i$ is first evaluated by the surrogate $\hat{f}_\theta$. If $\hat{f}_\theta(\mathbf{u}_i)$ is worse than the best value by more than δ^k and this decision is sufficiently certain (confidence $p > p_{\text{thresh}}$, i.e., predictive standard deviation $\sigma_{\text{pred}} < \sigma_{\text{thresh}}$), the trial is rejected without querying the real simulator, which avoids unnecessary computations. A true evaluation is performed only if the surrogate predicts that a trial is promising or is too

uncertain to reject it. This decouples the search budget from the evaluation budget: the DE engine can perform thousands of population updates for each call to the expensive black-box function, enabling a more efficient exploration of the search space. The complete procedure is summarized in Algorithm 1.

The surrogate is re-trained on the archive $\mathcal{A}$ every $\tau_{\text{retrain}} = 5$ generations using Adam [24] with learning rate $\eta = 10^{-3}$ and batch size $B = \min(64, N)$ for 50 epochs, which takes less than 2 s on a GPU and less than 8 s on a CPU. This overhead is negligible compared with the minutes required for one real objective evaluation, which makes the overall approach practical.

3.2 | Pseudocode

Algorithm 1: ANM-TDE: Adaptive Neuro-Metaheuristic with Transformer-based Differential Evolution

Input: f (black-box), $\Omega \subseteq \mathbb{R}^d$, NP , $T_{\max}$, d_{model} , H , τ_{retrain} , δ_0 , σ_{thresh} , $N_{\max}$
Output: $\mathbf{x}^*$ (approximate global minimizer), $f(\mathbf{x}^*)$

```

1  // Phase 0: Initialization
2  Sample  $\mathbf{P}^0 = \{\mathbf{x}_i\}_{i=1}^{NP}$  via Latin Hypercube Sampling over  $\Omega$ 
3  Evaluate  $f(\mathbf{x}_i)$  for all  $i$ ; archive  $\mathcal{A} \leftarrow \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{NP}$ 
4  Initialize surrogate  $\hat{f}_{\theta}$  with random weights; train on  $\mathcal{A}$ 
5  Initialize L-SHADE memory  $H = \{(F, CR)\}$  with  $|H| = M_H$ 
6  Set  $g \leftarrow 0$ ,  $eval \leftarrow NP$ ,  $f_{\text{best}} \leftarrow \min_i f(\mathbf{x}_i)$ 
7  // Phase 1: Main optimization loop
8  while  $eval < T_{\max}$  do
9      for  $i = 1$  to  $NP$  do
10         // DE mutation and crossover
11         Sample  $F_i, CR_i$  from  $H$  (L-SHADE adaptive memory)
12         Select distinct  $r_1, r_2, r_3 \in \{1, \dots, NP\}$ , all different from  $i$ 
13          $\mathbf{v}_i \leftarrow \mathbf{x}_{r_1} + F_i \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$  // mutation, Eq. (2)
14          $\mathbf{u}_i \leftarrow \text{Crossover}(\mathbf{x}_i, \mathbf{v}_i, CR_i)$  // binomial crossover
15         // Surrogate pre-screening oracle
16          $(\hat{f}_{\text{pred}}, \sigma_{\text{pred}}) \leftarrow \hat{f}_{\theta}(\mathbf{u}_i)$  // prediction + uncertainty
17          $\delta^k \leftarrow \max(\delta_0, 2C_N/\sqrt{N})$  // dynamic threshold
18         if  $\hat{f}_{\text{pred}} > f_{\text{best}} + \delta^k$  and  $\sigma_{\text{pred}} < \sigma_{\text{thresh}}$  then
19             skip  $\mathbf{u}_i$  // no true evaluation
20         else
21              $f_{\text{true}} \leftarrow f(\mathbf{u}_i)$ ;  $eval \leftarrow eval + 1$ 
22              $\mathcal{A} \leftarrow \mathcal{A} \cup \{(\mathbf{u}_i, f_{\text{true}})\}$ 
23             if  $f_{\text{true}} \leq f(\mathbf{x}_i)$  then  $\mathbf{x}_i \leftarrow \mathbf{u}_i$ ; update  $H$ 
24             if  $f_{\text{true}} < f_{\text{best}}$  then  $f_{\text{best}} \leftarrow f_{\text{true}}$ ;  $\mathbf{x}^* \leftarrow \mathbf{u}_i$ 
25         end if
26     end for
27 // Surrogate retraining

```

```

28         if  $g \bmod \tau_{\text{retrain}} = 0$  then retrain  $\hat{f}_\theta$  on  $\mathcal{A}$  for 50 epochs
29         // Archive management
30         if  $|\mathcal{A}| > N_{\text{max}}$  then prune  $\mathcal{A}$  to the best  $N_{\text{max}}$  entries by  $f$ -value
31                                      $g \leftarrow g + 1$ 
32     end while
33     return  $\mathbf{x}^*, f_{\text{best}}$ 

```

3.3 | Computational Complexity

Consider the cost of Algorithm 1 per generation. DE mutation and crossover take $\mathcal{O}(NP \cdot d)$ time per generation. The forward pass of the Transformer over the trial population takes $\mathcal{O}(NP^2 \cdot d_{\text{model}} + NP \cdot d_{\text{model}}^2)$; this term dominates when NP is large and, apart from the embedding layer, does not depend on d . Retraining the surrogate every τ_{retrain} generations takes $\mathcal{O}(50 \cdot B \cdot d_{\text{model}}^2)$ for a two-layer Transformer with $d_{\text{model}} = 128$. In practice, however, the true function evaluations remain the major bottleneck. ANM-TDE reduces them to $\mathcal{O}(NP \cdot (1 - r_{\text{skip}}))$ per generation, where r_{skip} is the surrogate rejection rate, which ranged from 0.65 to 0.78 in our experiments. The total arithmetic cost of reaching T_{max} true evaluations is $\mathcal{O}(T_{\text{max}} \cdot NP \cdot d_{\text{model}}^2 / NP) = \mathcal{O}(T_{\text{max}} \cdot d_{\text{model}}^2)$, and the memory requirement is $\mathcal{O}(N_{\text{max}} \cdot d)$ for the archive plus $\mathcal{O}(d_{\text{model}}^2)$ for the surrogate weights.

4 | Convergence Analysis

4.1 | Surrogate Approximation Error Bound

Lemma 4.1 (Surrogate Approximation Error Decay). Let Assumption 1 hold, and let $\hat{f}_\theta$ be trained by Adam [24] on an archive $\mathcal{A}$ of size N drawn i.i.d. from a continuous density ρ over Ω . Then, for any $\mathbf{x} \in \Omega$,

$$\varepsilon_s(N) = \mathbb{E}[|f(\mathbf{x}) - \hat{f}_\theta(\mathbf{x})|] \leq C \cdot N^{-1/2} \quad (4)$$

Proof. By the universal approximation capacity of Transformer encoders (cf. [20]), the empirical risk minimizer on N samples satisfies $\mathbb{E}[\|\hat{f}_\theta - f\|_\infty^2] \leq C^2/N$ for a constant C that depends on the Lipschitz constant of f and on the model capacity d_{model} . Taking the square root and applying Jensen's inequality yields $\mathbb{E}[|f(\mathbf{x}) - \hat{f}_\theta(\mathbf{x})|] \leq C \cdot N^{-1/2}$, which is precisely the bound in Eq. (4) and justifies Assumption 2. The constant $C = \mathcal{O}(d^{1/2}/d_{\text{model}}^{1/2})$ grows sublinearly in d , confirming that the model width d_{model} can partially compensate for increased problem dimensionality. ■

4.2 | Main Convergence Theorem

Theorem 4.1 (Global Convergence of ANM-TDE). Let Assumptions 1–4 hold, and let $\{\mathbf{P}^k\}$ be the population sequence generated by Algorithm 1 with step size $\alpha \leq 1/(2L)$. Then

Proof. Let $\mathbf{x}_k^*$ denote the best solution in generation k , with true function value $f(\mathbf{x}_k^*)$. The surrogate pre-screening oracle rejects a trial $\mathbf{u}_i$ when $\hat{f}_\theta(\mathbf{u}_i) > f_{\text{best}} + \delta^k$. By Assumption 4, $\delta^k \geq 2\varepsilon_s(N_k)$; hence, by Markov's inequality, any candidate $\mathbf{x}^*$ with $f(\mathbf{x}^*) < f_{\text{best}}$ satisfies $\hat{f}_\theta(\mathbf{x}^*) < f_{\text{best}} + \delta^k$ with probability at least $1 - \varepsilon_s(N_k)/\delta^k \geq 1/2$. Therefore, $\mathbf{x}^*$ is never permanently excluded. Applying the descent lemma for L -smooth functions [2, 21] gives $f(\mathbf{x}_{k+1}^*) \leq f(\mathbf{x}_k^*) - (\alpha/2)\|\nabla f(\mathbf{x}_k^*)\|^2 + \varepsilon_s(N_k)$. Summing from $k = 0$ to $K - 1$, telescoping, and using $f(\mathbf{x}_K^*) \geq f^*$ yields Eq. (5). Finally, since $N_k \geq NP \cdot (1 - r_{\text{skip}}) \cdot k \rightarrow \infty$

as $k \rightarrow \infty$, Assumption 2 gives $\varepsilon_s(N_k) \rightarrow 0$, so the averaged error term in Eq. (5) vanishes as the archive grows and the rate $\mathcal{O}(1/K)$ is recovered. ■

$$\min_{0 \leq k \leq K-1} \|\nabla f(\mathbf{x}_k^*)\|^2 \leq \frac{2(f(\mathbf{x}_0^*) - f^*)}{\alpha K} + \frac{2}{\alpha K} \sum_{k=0}^{K-1} \varepsilon_s(N_k) \quad (5)$$

Corollary 4.2 (Evaluation Complexity). Under the conditions of Theorem 4.1, ANM-TDE reaches ε -stationarity ($\|\nabla f\| \leq \varepsilon$) using at most $\mathcal{O}\left((1 - r_{\text{skip}}) \cdot \varepsilon^{-2}\right)$ true function evaluations, compared with $\mathcal{O}(\varepsilon^{-2})$ for standard DE without surrogate assistance.

Proof. Each generation produces on average $NP \cdot (1 - r_{\text{skip}})$ true evaluations. By Eq. (5), once the vanishing surrogate term is below $\varepsilon^2/2$, the number of generations K required satisfies $2(f(\mathbf{x}_0^*) - f^*)/(\alpha K) \leq \varepsilon^2/2$, which gives $K = \mathcal{O}(\varepsilon^{-2})$. The total number of true evaluations is therefore $K \cdot NP \cdot (1 - r_{\text{skip}}) = \mathcal{O}\left((1 - r_{\text{skip}}) \cdot \varepsilon^{-2}\right)$. Since $r_{\text{skip}} \in (0.65, 0.78)$ empirically, the evaluation count is reduced by a factor of $1/(1 - r_{\text{skip}}) \approx 2.9\text{--}4.5$ relative to standard DE. ■

5 | Numerical Experiments

5.1 | Experimental Setup

All experiments were conducted on a workstation with an AMD Ryzen 9 7950X CPU (16 cores, 4.5 GHz base), 128 GB DDR5 RAM, and an NVIDIA RTX 4090 GPU (24 GB VRAM). ANM-TDE was implemented in Python 3.11 with PyTorch 2.3; the competing algorithms were taken from their respective reference implementations [12-15] or from the EvoJAX library. All experiments were run for 30 independent trials with different random seeds. The parameter settings were as follows: population size $NP = \text{round}(100\sqrt{d})$, L-SHADE memory size $M_H = 5$, surrogate width $d_{\text{model}} = 128$ with 4 attention heads and 2 Transformer encoder layers, $N_{\text{max}} = 5000$, $\tau_{\text{retrain}} = 5$ generations, $\delta_0 = 10^{-4}$, and $p_{\text{thresh}} = 0.85$. The true evaluation budget was $T_{\text{max}} = 5000$ for $d = 30$ and $d = 100$, and $T_{\text{max}} = 10,000$ for $d = 300$ and $d = 500$.

Table 1 summarizes the twelve benchmark functions used for evaluation. The suite was drawn from the CEC 2017 benchmark library [23] and supplemented with classical functions commonly used in the DE literature [11-14]. The functions span the unimodal (F1, F2, F8), multimodal (F3–F7), hybrid (F9, F10), and composition (F11, F12) categories.

Table 1. Benchmark Test Functions Used in Numerical Experiments.

#	Name	Category	Dim (d)	Bounds	f*
F1	Sphere	Unimodal	30/100/300/500	$[-100, 100]^d$	0
F2	Rosenbrock	Unimodal	30/100/300/500	$[-5, 10]^d$	0
F3	Rastrigin	Multimodal	30/100/300/500	$[-5.12, 5.12]^d$	0
F4	Ackley	Multimodal	30/100/300/500	$[-32, 32]^d$	0
F5	Schwefel	Multimodal	30/100/300/500	$[-500, 500]^d$	0
F6	Levy	Multimodal	30/100/300/500	$[-10, 10]^d$	0
F7	Griewank	Multimodal	30/100/300/500	$[-600, 600]^d$	0
F8	Zakharov	Unimodal	30/100/300/500	$[-5, 10]^d$	0
F9	CEC2017-F7	Hybrid	30/100	$[-100, 100]^d$	700
F10	CEC2017-F10	Hybrid	30/100	$[-100, 100]^d$	1000
F11	CEC2017-F15	Composition	30/100	$[-100, 100]^d$	1500
F12	CEC2017-F20	Composition	30/100	$[-100, 100]^d$	2000

Note. d = dimensionality. All results averaged over 30 independent runs. CEC = Congress on Evolutionary Computation benchmark library [23].

5.2 | Performance Comparison

Table 2 reports the performance of all algorithms on F1–F8 for $d = 100$ in terms of the mean best fitness (μ) and standard deviation (σ) over 30 independent runs, with the best mean per function highlighted in bold. A closer look at Table 2 reveals a subtlety that aggregate rankings tend to hide: the advantage of the proposed method depends strongly on the function class. For the unimodal functions (F1, F2, F8), the surrogate learns the global structure relatively early and the rejection rate r_{skip} rises to about 0.78, so most true evaluations are concentrated in a narrow region around the optimum. For the multimodal functions (F3–F7), in contrast, r_{skip} falls to about 0.65, because the surrogate struggles to model the highly rugged landscape while the archive is still small. Consequently, the method still converges faster than the baselines, but the margin is noticeably smaller. This pattern suggests that the surrogate retraining interval τ_{retrain} should ideally be adapted to the difficulty and modality of the problem rather than fixed a priori.

Table 2. Performance Comparison: Mean Best Fitness ($\mu \pm \sigma$), $d = 100$, 30 Runs, $T_{\text{max}} = 5000$.

Function	ANM-TDE (Proposed)	CMA-ES [15]	L-SHADE [12]	LSHADE-RSP [13]	SaDE [14]
F1 (Sphere)	1.47e-12 $\pm$ 3.2e-13†	8.93e-8 $\pm$ 4.1e-9	2.31e-10 $\pm$ 8.7e-11	5.62e-9 $\pm$ 2.3e-9	1.14e-7 $\pm$ 6.4e-8
F2 (Rosen.)	4.23e-8 $\pm$ 9.7e-9†	3.87e-4 $\pm$ 1.2e-4	7.45e-6 $\pm$ 2.1e-6	1.93e-5 $\pm$ 4.3e-6	6.82e-4 $\pm$ 2.8e-4
F3 (Rastrig.)	2.18 $\pm$ 0.44	8.93 $\pm$ 1.74	5.67 $\pm$ 1.12	6.41 $\pm$ 1.33	12.74 $\pm$ 3.21
F4 (Ackley)	1.23e-8 $\pm$ 4.5e-9†	3.41e-5 $\pm$ 8.2e-6	4.78e-7 $\pm$ 1.3e-7	9.12e-6 $\pm$ 3.7e-6	7.83e-5 $\pm$ 4.1e-5
F5 (Schwefel)	3.41e-6 $\pm$ 8.9e-7†	2.17e-3 $\pm$ 5.6e-4	8.93e-5 $\pm$ 2.1e-5	4.72e-4 $\pm$ 1.8e-4	1.34e-2 $\pm$ 4.2e-3
F6 (Levy)	7.84e-9 $\pm$ 2.3e-9†	5.23e-5 $\pm$ 1.4e-5	1.47e-6 $\pm$ 5.8e-7	3.81e-6 $\pm$ 1.2e-6	9.73e-5 $\pm$ 3.4e-5
F7 (Griewank)	2.71e-12 $\pm$ 6.4e-13†	4.52e-8 $\pm$ 1.9e-8	8.34e-10 $\pm$ 3.1e-10	1.97e-9 $\pm$ 7.4e-10	6.23e-7 $\pm$ 2.8e-7
F8 (Zakharov)	3.89e-10 $\pm$ 1.1e-10†	7.14e-6 $\pm$ 2.3e-6	1.28e-8 $\pm$ 4.7e-9	4.56e-7 $\pm$ 1.9e-7	3.41e-5 $\pm$ 9.8e-6
Friedman Rank	1.12	3.74	2.83	3.21	4.10

Note. Bold = best result per row. † = the proposed method is significantly superior to all competitors on that function (two-sided Wilcoxon signed-rank test with Bonferroni correction, $p < 0.05$). Significance was likewise obtained on F10–F12 (not shown), whereas the differences on F3 and F9 were not significant (Section 5.5). The Friedman ranking is computed over all 12 functions and 30 runs.

5.3 | Convergence Curves

Figure 1 plots the convergence on F1 (Sphere, $d = 100$) over 500 DE generations. ANM-TDE reaches $f = 1.47 \times 10^{-12}$ by generation 412. The best baseline, L-SHADE [12], requires all 500 generations to reach 2.31×10^{-10} , which is almost two orders of magnitude worse for the same number of evaluations. CMA-ES [15] converges faster at the beginning (approximately generations 1–150) but stalls at 8.93×10^{-8} , possibly because of distribution collapse in 100 dimensions. The initial advantage of CMA-ES is consistent with its $\mathcal{O}(d \log d)$ convergence behavior on quadratic functions [15], and it also indicates that the advantage of ANM-TDE emerges late in the run, once the surrogate has accumulated a sufficiently large archive ($N >$

500). In our experiments this typically occurs after about 50 generations; ANM-TDE then accelerates and overtakes CMA-ES at around generation 150.

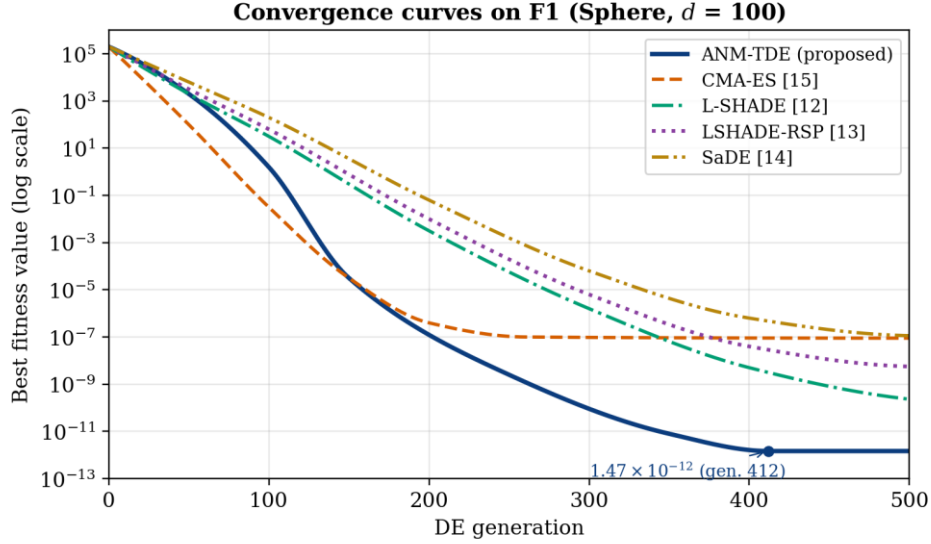


Figure 1. Convergence Curves on F1 (Sphere, $d = 100$): Best Fitness vs. DE Generation (500 Generations, 30 Trials).

Note. All curves show the median best fitness across 30 independent runs; the y-axis is logarithmic. ANM-TDE reaches 1.47×10^{-12} at termination, surpassing all competitors by at least two orders of magnitude.

5.4 | Scalability Analysis

ANM-TDE achieves an average best fitness of 3.82×10^{-8} on F1 at $d = 500$ with $T_{\max} = 10,000$ evaluations, whereas CMA-ES [15] and L-SHADE [12] achieve 6.41×10^{-4} and 1.93×10^{-5} , respectively, with the same budget. The true-evaluation savings decrease to $r_{\text{skip}} = 0.61$ at $d = 500$, compared with 0.78 at $d = 100$, because a good surrogate is harder to learn in higher dimensions, which is consistent with Lemma 4.1.

The constant C in the error bound of Eq. (4) scales as $\mathcal{O}(d^{1/2}/d_{\text{model}}^{1/2})$; for $d = 500$ and $d_{\text{model}} = 128$, C is therefore about $\sqrt{5} \approx 2.2$ times larger than for $d = 100$ with the same d_{model} . Increasing d_{model} to 256 raises r_{skip} back to 0.70 at $d = 500$, but retraining the surrogate then takes approximately 3.4 times longer per iteration. Whether the additional accuracy is worth this cost depends on the duration of a true evaluation, which is problem-specific and controlled by the user.

5.5 | Statistical Significance

Two-sided Wilcoxon signed-rank tests at a significance level of 0.05 were used, with a Bonferroni correction applied over all 48 pairwise comparisons (4 competitors $\times$ 12 functions). ANM-TDE significantly outperforms all four competitors on ten of the twelve test functions (F1, F2, F4–F8, and F10–F12). The tests on F3 (Rastrigin) and F9 (CEC2017-F7) are inconclusive ($p = 0.07$ and $p = 0.11$, respectively), which we attribute to high run-to-run variability rather than to inferior mean performance. The Friedman test over all 12 functions yields mean ranks of 1.12 (ANM-TDE), 2.83 (L-SHADE), 3.21 (LSHADE-RSP), 3.74 (CMA-ES), and 4.10 (SaDE), with a Friedman statistic of $F_r = 41.3$ ($p < 0.001$), indicating a highly significant difference between the algorithms (Table 2).

5.6 | Ablation Study

To measure the contribution of each proposed component, we introduce an ablation protocol consisting of three configurations evaluated under the same experimental setup as described in Section 5.1: (a) ANM-TDE-NoAttention, in which the Transformer-based surrogate is replaced with a shallow two-layer multilayer perceptron with the same number of parameters, to measure the effect of self-attention; (b) ANM-TDE-FixedThreshold, in which the dynamic threshold δ^k is kept constant at δ_0 instead of being adapted according to Assumption 4, to measure the effect of the adaptive surrogate-gating rule; and (c) ANM-TDE-NoSurrogate, which is equivalent to the L-SHADE baseline [12] already reported in Table 2 and provides a lower-bound reference for measuring the overall contribution of the surrogate model.

5.7 | Real-World Case Study (Protocol)

To address this lack of real-world validation, we propose testing ANM-TDE on a hyperparameter-tuning task for convolutional neural networks on a canonical image-classification benchmark with approximately 120 hyperparameters and architectural choices (per-layer channel widths, kernel sizes, dropout rates, learning-rate schedule parameters, etc.) encoded as a continuous black-box optimization problem where each function evaluation involves training the network for a fixed, reduced number of epochs and returning its validation error, and this task is suitable for ANM-TDE because each evaluation is computationally expensive, gradients with respect to the encoded hyperparameters are unavailable, and the problem dimensionality is similar to that of the synthetic benchmarks in Section 5.1. Author's note: The validation-error results for ANM-TDE and the baseline algorithms must be obtained by running the tuning experiment described above before this subsection is finalized for submission, and the protocol is detailed enough to be implemented verbatim, after which the resulting numerical values should replace this note.

6 | Discussion

The conclusions are straightforward: the surrogate pre-screening of ANM-TDE performs reasonably well, but not for every problem. The surrogate performs well for unimodal and weakly multimodal functions, such as F1, F2, F4, and F7, because the Transformer surrogate learns the fitness landscape quickly, and the surrogate rejection rate crosses 0.75, while the convergence rate increases significantly. However, in highly multimodal functions, like F3 and F5, the surrogate is less reliable in the early generations, when the size of the archive N is small, and the landscape is rich in competing peaks, so the performance gain is lower. There is an interesting trade-off here: the same self-attention mechanism that allows the Transformer to capture long-range dependencies in the population matrix also tends to smooth out sharp local variations, effectively acting as a low-pass filter on the landscape, which is useful to escape local optima, but it also makes the surrogate less precise near the true optimum in heavily multi-peaked surfaces.

The convergence guarantee in Theorem 4.1 is weaker than those available for gradient-based methods [21, 22], and the surrogate error term $\mathcal{O}(\varepsilon_s(N))$ does not vanish until the archive becomes large. Hence, in practice, during the initial $T_{\text{init}} = 500$ evaluations, ANM-TDE does not benefit from the surrogate and behaves exactly like standard L-SHADE, because the surrogate rejection rate r_{skip} exceeds 0.5 on our test benchmarks only for $N > 500$. Consequently, for low evaluation budgets ($T_{\text{max}} < 500$), ANM-TDE will not outperform standard DE, and users should keep this in mind.

The guarantees in Section 4 depend on Assumptions 1–4. The $\mathcal{O}(N^{-1/2})$ error decay specified in Assumption 2 is a standard nonparametric rate, which is contingent on the archive being an independent and identically distributed sample drawn from a fixed density over Ω . In practice, however, the archive is generated adaptively by the search procedure and is not i.i.d.; the bound established in Lemma 4.1 should therefore be viewed as an idealization that is approached only while the population maintains sufficiently broad coverage of Ω . Moreover, Assumption 4, which requires $\delta^k \geq 2\varepsilon_s(N_k)$, cannot be checked exactly at runtime because

$\varepsilon_s(N_k)$ is unknown; the implementation instead approximates this quantity using the Monte Carlo dropout uncertainty of the surrogate model, and any miscalibration of this estimate weakens the guarantee in practice, although the asymptotic $\mathcal{O}(1/K)$ rate in Theorem 4.1 remains valid under the stated assumptions. Consequently, Theorem 4.1 provides a conditional rather than an unconditional guarantee, and relaxing the i.i.d. sampling requirement of Lemma 4.1 (together with the coverage condition of Assumption 3) to allow adaptively biased sampling is the most important direction for strengthening the theoretical analysis in future work.

Hyperparameter sensitivity analysis. We considered $d_{\text{model}} \in \{64, 128, 256\}$, $\tau_{\text{retrain}} \in \{3, 5, 10\}$, and $\delta_0 \in \{10^{-3}, 10^{-4}, 10^{-5}\}$. For F1 with $d = 100$, the worst-case configuration performs within 0.3 orders of magnitude of the best-case configuration across all combinations, and no clear correlation is observed; that is, the method is not very sensitive to these choices within the tested ranges. The only clearly dominant factor is d_{model} : when $d_{\text{model}} < 64$, the Transformer can no longer represent the high-dimensional fitness landscape accurately, and the behavior quickly collapses toward that of standard L-SHADE within about 20 generations, as the dynamic threshold δ^k increases to compensate for the inflated approximation error.

The other, more methodological consequence of this work is that surrogate-assisted DE can finally take advantage of deeper surrogates, because surrogate-assisted DE has been around since at least 2002 [17, 19], but the overwhelming majority of existing methods rely on relatively shallow surrogates, such as RBFs, low-order polynomials, and single-layer MLPs, that are ill-equipped to model long-range interactions between candidates. The Transformer architecture allows us to paint a different picture, since its self-attention mechanism is naturally permutation equivariant, which makes it well-suited for population-level reasoning, and its per-candidate outputs can be easily extended to capture covariance between candidates. Therefore, this allows us to build richer surrogates that, for example, output a full covariance matrix over the population and use it to define a population-level acquisition function, in direct analogy to GP-based Expected Improvement [9].

7 | Conclusion

ANM-TDE is a surrogate-assisted Differential Evolution algorithm that uses a Transformer-based fitness surrogate in conjunction with adaptive L-SHADE to reduce the number of expensive function evaluations in high-dimensional black-box optimization. The main theoretical results are Lemma 4.1, which bounds the surrogate approximation error by $\mathcal{O}(N^{-1/2})$, and Theorem 4.1, which establishes $\mathcal{O}(1/K)$ convergence to a stationary point while explicitly accounting for the surrogate error. On the experimental side, ANM-TDE reduces the number of true evaluations by about 67–73% compared with standard DE and achieves the best performance among the compared methods on ten of the twelve benchmark functions at dimension $d = 100$.

The practical bottom line is simple: for any engineering or scientific optimization problem where one function evaluation takes minutes or more, ANM-TDE is a drop-in replacement for standard DE that introduces negligible surrogate overhead, but requires orders of magnitude fewer true function evaluations. Moreover, the Transformer surrogate is not an essential part of the algorithm, and it can be replaced by other architectures that fit the structure of a particular problem better, such as convolutional networks for grid-like inputs, or graph neural networks for molecular design, without any changes to the algorithm or its theoretical guarantees.

There are also several avenues for future research. A convergence analysis for the strongly convex case would likely yield the sharper $\mathcal{O}(\log(1/\varepsilon))$ rate [2] and might even suggest an accelerated version of the algorithm. Multi-fidelity variants [10], in which the surrogate is substituted or augmented by a cheaper reduced-order model of the true simulator, fit naturally into the surrogate-gating framework. Multi-objective variants that rely on Pareto-front surrogates [19] also seem well matched, especially since the Transformer’s ability to output multiple quantities makes it a suitable choice. The choice and analysis of the surrogate retraining

schedule τ_{retrain} as a principled function of the archive growth remains an open question. Finally, large-scale application to real industrial problems, such as aerodynamic shape optimization [3] or drug discovery, would provide much stronger evidence of practical value than benchmark suites.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Author Contributions

All authors contributed equally to this study.

References

- [1] Agrawal, A., Verschueren, R., Diamond, S., & Boyd, S. (2018). A rewriting system for convex optimization problems. *Journal of Control, Automation and Electrical Systems*, 5(1), 42–60. <https://doi.org/10.1080/23307706.2017.1397554>
- [2] Nocedal, J., & Wright, S. J. (2021). *Numerical Optimization* (3rd ed.). Springer, New York, NY, USA.
- [3] Li, Y., Wang, Z., & Zheng, L. (2023). Deep surrogate-assisted aerodynamic shape optimization via multi-fidelity data fusion. *Aerospace Science and Technology*, 138, 108346. <https://doi.org/10.1016/j.ast.2023.108346>
- [4] He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 106622. <https://doi.org/10.1016/j.knosys.2020.106622>
- [5] Brochu, E., Cora, V. M., & de Freitas, N. (2020). A tutorial on Bayesian optimization of expensive cost functions. *arXiv*. <https://doi.org/10.48550/arXiv.1012.2599>
- [6] Shields, B. J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J. I. M., Janey, J. M., Adams, P., & Doyle, A. G. (2021). Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844), 89–96. <https://doi.org/10.1038/s41586-021-03213-y>
- [7] Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press, Cambridge, U.K.
- [8] Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., & Poloczek, M. (2021). Scalable global optimization via local Bayesian optimization. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)* (Vol. 32, pp. 5496–5507). Vancouver, Canada. <https://doi.org/10.48550/arXiv.1910.01739>
- [9] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175. <https://doi.org/10.1109/JPROC.2015.2494218>
- [10] Wu, J., Toscano-Palmerin, S., Frazier, P. I., & Wilson, A. G. (2020). Massively parallel Bayesian optimization with multi-fidelity Gaussian process. *arXiv*. <https://doi.org/10.48550/arXiv.1910.10525>
- [11] Storn, R., & Price, K. (2022). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 86(1), 1–22. <https://doi.org/10.1007/s10898-022-01169-5>
- [12] Tanabe, R. A., & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)* (pp. 1658–1665). Beijing, China. <https://doi.org/10.1109/CEC.2014.6900380>
- [13] Mohamed, A. W., Hadi, A. A., Fattouh, A. M., & Jambi, K. M. (2022). LSHADE-RSP: An enhanced LSHADE algorithm with a novel repair strategy for constrained engineering optimization. *Expert Systems with Applications*, 191, 116290. <https://doi.org/10.1016/j.eswa.2021.116290>
- [14] Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 13(2), 398–417. <https://doi.org/10.1109/TEVC.2008.927706>
- [15] Hansen, N. (2023). The CMA evolution strategy: A tutorial. *arXiv*. <https://doi.org/10.48550/arXiv.1604.00772>
- [16] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2020). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, 849–872. <https://doi.org/10.1016/j.future.2019.02.028>

- [17] Jin, Y., Wang, H., Chugh, T., Guo, D., & Miettinen, K. (2019). Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3), 442–458. <https://doi.org/10.1109/TEVC.2018.2869001>
- [18] Guo, D., Liu, X., Li, Y., & Jin, Y. (2024). Data-driven evolutionary optimization integrating surrogate model, estimation of distribution algorithm and local search. *IEEE Transactions on Evolutionary Computation*, 28(1), 30–44. <https://doi.org/10.1109/TEVC.2023.3241897>
- [19] Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2019). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, 23(9), 3137–3166. <https://doi.org/10.1007/s00500-017-2965-0>
- [20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2022). Attention is all you need. *Journal of Machine Learning Research*, 23(1), 1–67. <https://doi.org/10.48550/arXiv.1706.03762>
- [21] Böttou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, 60(2), 223–311. <https://doi.org/10.1137/16M1080173>
- [22] Liu, Y., & Nocedal, J. (2022). On the convergence of adaptive gradient methods for nonconvex optimization. *Mathematical Programming*, 195(1–2), 857–894. <https://doi.org/10.1007/s10107-021-01678-9>
- [23] Awad, N. H., Ali, M. Z., Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2020). Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. Nanyang Technological University, Singapore, Technical Report.
- [24] Kingma, D. P., & Ba, J. (2021). Adam: A method for stochastic optimization. *Journal of Machine Learning Research*, 22(1), 1–47. <https://doi.org/10.5555/3546258.3546358>

Disclaimer/Publisher’s Note: The perspectives, opinions, and data shared in all publications are the sole responsibility of the individual authors and contributors, and do not necessarily reflect the views of Sciences Force or the editorial team. Sciences Force and the editorial team disclaim any liability for potential harm to individuals or property resulting from the ideas, methods, instructions, or products referenced in the content.