PNKL-DE: A Positive-Negative Knowledge Learning Framework for Differential Evolution

Kanchan Rajwar*
Department of Computer Science
University of Pretoria
Pretoria, 0083, South Africa
ORCID: 0000-0002-7413-0057

Nelishia Pillay
Department of Computer Science
University of Pretoria
Pretoria, 0083, South Africa
ORCID: 0000-0003-3902-5582

Thambo Nyathi
Department of Computer Science
University of Pretoria
Pretoria, 0083, South Africa
ORCID: 0000-0003-4676-6063

***Abstract*—Knowledge-Learning Evolutionary Algorithms represent an emerging paradigm designed to transfer acquired information for more effective search the solution space. However, existing literature typically focuses on extracting and sharing only "positive" knowledge—information derived exclusively from successful search regions. In this study, we propose a novel approach that utilizes negative knowledge as well to guide the evolutionary process. We introduce Positive-Negative Knowledge-based Learning Differential Evolution (PNKL-DE), a novel algorithm that explicitly exploits search history from both successful (positive) and unsuccessful (negative) outcomes to enhance optimization performance. The proposed algorithm is rigorously evaluated on the IEEE CEC 2017 benchmark suite and compared against state-of-the-art and CEC competition winners, including JADE, SHADE, L-SHADE, jSO, NL-SHADE-RSP, and MadDE. Furthermore, PNKL-DE is compared with the recently propped evolutionary transfer learning method Knowledge Learning DE (KLDE) and Knowledge Learning PSO (KLPSO). Experimental results demonstrate that PNKL-DE achieves statistically significant improvements, validating the hypothesis that negative knowledge is a critical, underutilized resource in evolutionary computation.**

***Index Terms*—Evolutionary Computation, Evolutionary Transfer Learning, Differential Evolution, Negative Learning, Global Optimization.**

I. Introduction

Evolutionary Algorithms (EAs) accumulate substantial search-related data throughout their optimization runs. The growing research field of knowledge transfer focuses on extracting useful insights from this stored information to better steer the evolutionary process. When insights obtained from earlier stages are applied to guide current population members, algorithms can achieve faster convergence and superior final solutions [1].

By analyzing the data generated during an algorithm's search, valuable insights—or knowledge—can be extracted to make the evolutionary process more effective and efficient. This approach has given rise to a distinct paradigm within Evolutionary Computation (EC) known as Evolutionary Transfer Learning (ETL) or Evolutionary Transfer Optimization (ETO) [1]. A central idea in current EC research is that the strategic use of accumulated search data can substantially improve algorithmic performance.

*Corresponding author

This data-driven enhancement typically occurs in two primary ways. The first involves cross-problem knowledge transfer, where information or patterns learned from solving different, often related, optimization problems are applied to accelerate or guide the search for a new problem, as seen in algorithms like [2]. The second approach focuses on historical knowledge reuse within a single optimization run. Here, data from previous generations of the same run is mined to inform current decisions. For instance, Zhang and Yuen [3] developed a directional mutation operator that reuses past successful search directions to generate new candidate solutions. Similarly, Ghosh et al. [4] introduced a mechanism that archives successful differential vectors from a Differential Evolution (DE) algorithm and later resamples them to guide future variation. Knowledge-aware EC algorithms are a growing frontier, proving that utilizing past search data profoundly boosts both evolutionary efficiency and performance.

However, existing literature on knowledge transfer is predominantly focused on "positive knowledge" or "successful experience." Recent approaches, such as Learning-aided Evolutionary Optimization (LEO) [5] and Knowledge Learning Evolutionary Computation (KLEC) [6], extract information solely from successful search regions to attract the population toward potential optima. While effective, this creates an imbalance in the learning process, as the algorithm explicitly learns "what to do" while remaining ignorant of "what to avoid." This study addresses this limitation by introducing a more holistic knowledge transfer paradigm.

To the best of our knowledge, the utilization of negative knowledge to actively guide search mechanisms remains unexplored in the domain of knowledge learning. While conceptual similarities exist with negative learning in Estimation of Distribution Algorithms (EDAs), Teaching–Learning-Based Optimization (TLBO) variants, opposition-based learning (OBL), and Tabu Search [7], fundamental distinctions separate our approach. Prior negative-learning techniques typically rely on static archives, probabilistic down-weighting, or heuristic rules without generalizing displacement trajectories via neural networks. Furthermore, Tabu Search, originally designed for combinatorial (discrete) optimization, treats negative history as a hard constraint; it employs a memory list to forbid revisiting specific solutions to prevent cycling. In Tabu Search, negative

history (tabu list) is not generalized; it is simply a list of prohibited solutions. Since blocking specific coordinates is ineffective in the infinite search space of continuous domains, our proposed method—which learns and generalizes from failure—offers a

A. Motivation

The primary motivation for this study stems from the computational inefficiency inherent in discarding negative search history. In complex optimization landscapes, a substantial proportion of function evaluations yield sub-optimal results. While conventional algorithms treat this data as negligible by-products, these failures contain critical information regarding stagnation basins and low-quality regions. By ignoring this negative knowledge, traditional methods risk redundantly exploring unpromising areas. Therefore, a balanced knowledge transfer strategy, one that utilizes negative results to repel the search from inferior regions, much like positive results attract it toward promising ones, is essential for achieving robust and efficient optimization.

The main contributions of this study are:

- To the best of our knowledge, this is the first study to explicitly integrate *Negative Knowledge* (unsuccessful experience) into ETL, shifting the focus from solely learning successful patterns to also learn from failure.
- We introduce PNKL-DE, a hybrid framework that utilizes positive history to probabilistically replace the crossover operator with a neural network guided generation, while simultaneously employing a discriminative model to screen candidates based on negative history.

The remainder of this paper is organized as follows: Section II reviews related studies. Section III details the proposed PNKL-DE framework. Section IV presents the experimental results. Finally, Section V concludes the study with future scopes.

II. Background

A. Related Work

A significant body of research focuses on utilizing historical data to enhance the performance of EAs. These algorithms systematically collect and reuse information from previous generations—such as successful parameter settings or high-fitness individuals—to guide the evolutionary process of the current population. This approach aims to inject learned experience into the search, thereby improving convergence speed and solution quality. This study focuses on historical knowledge reuse within a single optimization run. Generally, two-way knowledge transfer can be seen in the literature.

(i) **Traditional knowledge transfer (solution and model transfer):** Within the framework of DE, several adaptive algorithms exemplify this paradigm. The study [8] proposed JADE, which maintains an archive of historically successful control parameters and uses them to probabilistically generate new parameters. JADE also preserves past individuals in an archive, allowing them to serve as parental candidates. Building on this, the study [9] developed SHADE (Success-History Based Adaptive DE), which maintains multiple probability distributions estimated from past successful parameters and samples new parameters from a mixture of these distributions. Further extending the concept of historical utilization, the study [10] introduced an Adaptive Distributed DE (ADDE), where parameters are adjusted per generation based on historical trends and archived individuals are reused to generate offspring.

Parallel developments are evident in Particle Swarm Optimization (PSO). The study [11] proposed an archive-based PSO where historically well-performing particles are stored and serve as exemplars to guide the velocity updates of the current swarm. Another study [12] advanced this idea with a Triple Archive PSO (TAPSO), which segregates historical particles with distinct properties into three separate archives to provide more diversified and effective guidance for the swarm's evolution.

(ii) **ML-Guided knowledge transfer (search Strategy Transfer):** More recently, research has integrated machine learning (ML) models to extract and apply knowledge from historical data. The study [12] proposed a method for dynamic optimization that uses both observable parameters and individuals from past environments. Their approach predicts the optimal solution for the current environment by employing a feedforward neural network (FNN) trained on historical data. In a closely related study [6], introduced Knowledge Learning Differential Evolution and Knowledge Learning Particle Swarm Optimization (KLDE and KLPSO). These algorithms store vectors representing successful updates and train an FNN to probabilistically model and predict promising new solution directions.

A critical observation in literature is that existing methods predominantly exploit "positive knowledge"—information derived from successful searches. The potential of systematically learning from "negative knowledge," such as unsuccessful trials or regions of low fitness, to actively avoid poor search directions remains largely unexplored. Motivated by this research gap, we propose PNKL-DE algorithm, detailed in the following section.

III. Proposed Algorithm: PNKL-DE

The proposed PNKL-DE enhances the standard DE framework by integrating both positive and negative knowledge. This section details the mathematical formulation of these knowledge types and describes their distinct integration into the crossover and selection phases of DE.

A. Definition of Knowledge

In the proposed framework, knowledge accumulation is driven by analyzing the evolutionary trajectory of individuals. Let an individual be located at position $\mathbf{x}_g$ with fitness $f(\mathbf{x}_g)$. If this individual moves to a new position $\mathbf{x}_{g+1}$ with fitness $f(\mathbf{x}_{g+1})$, the displacement vector is defined as $\mathbf{d} = \mathbf{x}_{g+1} - \mathbf{x}_g$. The quality of this transition classifies the knowledge type:

1. Positive Knowledge (Successful Direction): If $f(\mathbf{x}_{g+1}) < f(\mathbf{x}_g)$ (minimization), the transition is deemed a successful

evolution. The pair $(\mathbf{x}_g, \mathbf{d})$ is identified as positive knowledge, representing a trajectory that leads to optimization. These pairs are stored in the Positive Archive, $\mathcal{A}_{pos}$, following the methodology in [6].

2. Negative Knowledge (Failed Direction): Conversely, if $f(\mathbf{x}_{g+1}) \geq f(\mathbf{x}_g)$, the transition is unsuccessful. The pair $(\mathbf{x}_g, \mathbf{d})$ is classified as negative knowledge—indicating a direction that leads to stagnation or fitness deterioration—and is stored in the Negative Archive, $\mathcal{A}_{neg}$.

B. Positive Knowledge Integration (Hybrid Crossover)

In PNKL-DE, Positive Knowledge is explicitly utilized to enhance the Crossover operator. Since crossover typically facilitates exploitation, we augment this by introducing a data-driven exploitation mechanism via a trained FNN, denoted as Φ_{pos}. For each individual $\mathbf{x}_{i,g}$ at generation g, the process is as follows:

1) Step 1 (Standard Mutation): First, a mutant vector $\mathbf{v}_{i,g}$ is generated using the classic DE/rand/1 strategy [14]:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r1,g} + F \cdot (\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}) \tag{1}$$

where $r1, r2, r3$ are mutually exclusive random indices distinct from i, and F is the scaling factor.

2) Step 2 (Probabilistic Hybrid Crossover): To generate the provisional trial vector $\mathbf{u}'_{i,g}$, the algorithm switches between biological evolution (standard crossover) and FNN prediction. A parameter p (e.g., 0.2) controls the learning rate. The provisional trial vector is defined as:

$$\mathbf{u}'_{i,g} = \begin{cases} \mathbf{x}_{i,g} + \Phi_{pos}(\mathbf{x}_{i,g}), & \text{if rand}(0,1) < p, \\ \text{Crossover}(\mathbf{x}_{i,g}, \mathbf{v}_{i,g}), & \text{otherwise.} \end{cases} \tag{2}$$

Here:

- **Neural Prediction:** $\Phi_{pos}(\mathbf{x}_{i,g})$ outputs a predicted displacement vector $\hat{\mathbf{s}}_i$ based on learned successful trajectories from $\mathcal{A}_{pos}$.
- **Standard Crossover:** If the condition $(\text{rand}(0,1) < p)$ is not met, the standard binomial crossover is applied between the target $\mathbf{x}_{i,g}$ and mutant $\mathbf{v}_{i,g}$:

$$u'_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if rand}_j(0,1) \leq CR \text{ or } j = j_{\text{rand}}, \\ x_{j,i,g}, & \text{otherwise,} \end{cases} \tag{3}$$

  where CR is the crossover rate.

C. Negative Knowledge Integration

While Positive Knowledge drives generation, Negative Knowledge is utilized for "Pre-Evaluation Screening". A Discriminative Model (Ψ_{neg}), trained on $\mathcal{A}_{neg}$, estimates the probability that a generated candidate lies in a stagnation basin or is unsuccessful. Before the expensive fitness evaluation of $\mathbf{u}'_{i,g}$, the candidate is validated:

$$\mathbf{u}_{i,g} = \begin{cases} \mathbf{x}_{best,g} + \epsilon, & \text{if } \Psi_{neg}(\mathbf{u}'_{i,g}) > \tau \\ \mathbf{u}'_{i,g}, & \text{otherwise} \end{cases} \tag{4}$$

where:

- τ is the rejection threshold (set to 0.8).
- $\Psi_{neg}(\mathbf{u}'_{i,g})$ returns a failure probability $p_{fail} \in [0,1]$.
- If the candidate is rejected (high failure probability), it is replaced by a small Gaussian perturbation ϵ around the current global best $\mathbf{x}_{best,g}$, effectively forcing a safe exploitation move rather than wasting evaluations on a likely failure.

Upon completion of this screening phase, the final candidate $\mathbf{u}_{i,g}$ proceeds to the standard DE selection step.

D. Knowledge Update

At the end of generation g, the archives are updated based on the actual fitness evaluation of the final trial vector $\mathbf{u}_{i,g}$:

1) **Positive Update:** If $f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g})$, the tuple $(\mathbf{x}_{i,g}, \mathbf{u}_{i,g} - \mathbf{x}_{i,g})$ is added to $\mathcal{A}_{pos}$.
2) **Negative Update:** If $f(\mathbf{u}_{i,g}) \geq f(\mathbf{x}_{i,g})$, the tuple $(\mathbf{x}_{i,g}, \mathbf{u}_{i,g} - \mathbf{x}_{i,g})$ is added to $\mathcal{A}_{neg}$.

Both neural networks Φ_{pos} and Ψ_{neg} are periodically retrained using the updated archives to synchronize with the current landscape. Since the archives are initially empty, PNKL-DE operates as a standard DE (Eq. 1 and Eq. 3) for the first generation.

IV. Experimental Setup and Result

A. Experimental Setup

Benchmarking Algorithm: To comprehensively assess the performance and robustness of the proposed framework, we conducted three distinct experiments: In the first experiment, we compare the proposed PNKL-DE against the standard DE algorithm. In the second experiment, we evaluate PNKL-DE against a suite of highly competitive algorithms, including classical adaptive variants and recent CEC competition winners. The comparison set includes JADE [8], SHADE [9], L-SHADE [15], jSO [16] (CEC 2019 winner, DE variants), NL-SHADE-RSP [17], and MadDE [18] (CEC 2022 winner). In the third experiment, we compare PNKL-DE with recently proposed ML-based EAs, specifically KLDE and KLPSO [6]. As these methods rely exclusively on neural networks trained on *positive* knowledge, this comparison is critical to demonstrate the specific advantage of integrating *negative* knowledge into the learning process.

In the proposed algorithm PNKL-DE, parameters are set to $p = 0.2$ and $\tau = 0.8$, as initial empirical tests demonstrated that these values yield the high-quality solutions. The parameter settings for all other algorithms follow the configurations specified in their original publications, as summarized in Table I.

Benchmarking Problems: The performance of the proposed PNKL-DE is evaluated using the IEEE CEC 2017 benchmark suite [19]. This suite comprises a set of 29 comprehensive, challenging optimization problems. It should be noted that the function F_2 was excluded from the test suite due to its unstable behavior, as per the standard competition guidelines. The functions are categorized into four distinct groups: two unimodal

TABLE I: Parameter Settings for DE and its Variants

Algorithm	Parameter	Value	Algorithm	Parameter	Value
DE	F (Mutation)	0.5	jSO	Memory Size	D
	CR (Crossover)	0.9		p (p-best)	0.2
JADE	p (p-best)	0.05		μ_{CR} (initial)	0.8
	c (adapt rate)	0.1	L-SHADE	Min Pop. Size	4
	μ_F (initial)	0.5	NL-SHADE-RSP	Stagnation Th.	20
	μ_{CR} (initial)	0.5		Restart Frac.	0.5
SHADE	Memory Size	100	MadDE	Sub-pops	4
	p (p-best)	0.1		Migration Prob.	0.1
KLDE	Learning Rate (lr)	0.2	KLPSO	Learning Rate (lr)	0.2

functions (F1-F2), eight simple multimodal functions (F3-F10), ten hybrid functions (F11-F20), and nine composition functions (F21-F29).

Performance metrics: To ensure statistical robustness, each algorithm is executed for 31 independent runs per test function. Performance is measured using the mean $\pm$ standard deviation (std). Experiments are conducted across two dimensions: $D = 30, 100$. The search budget is defined by the maximum number of function evaluations (MaxFEs), set as $MaxFEs = D \times 10^4$. To ensure a fair comparison, a fixed population size of 100 with uniform initialization is employed for all algorithms in every independent run.

A Wilcoxon rank-sum test at a 5% significance level is employed for statistical comparison. The symbols "+", "=", and "−" indicate that the PNKL-DE is significantly superior, equivalent, or inferior to the respective algorithm, respectively.

Neural Network Specification: To ensure reproducibility, we explicitly define the architectures and training data for the embedded FNN. Both models employ a dynamic structure scaling with the problem dimension D.

- **Generative Model (Φ_{pos}):** Trained on the Positive Archive $\mathcal{A}_{pos}$. It maps the current position to a predicted optimal step. It comprises an input layer (D), two hidden layers ($2D, D$) with *ReLU* activation, and an output layer (D) with *Linear* activation to predict continuous vectors.
- **Discriminative Model (Ψ_{neg}):** Trained on the Negative Archive $\mathcal{A}_{neg}$. It shares the same hidden structure ($2D, D$) but terminates with a single output neuron using a *Sigmoid* activation to constrain the output to $[0, 1]$.

Training is performed online at the end of each generation using the hyperparameters detailed in Table II.

TABLE II: Neural Network Hyperparameters speciation

Parameter	Value
Structure	Input(D) $\rightarrow$ Hidden($2D$) $\rightarrow$ Hidden(D) $\rightarrow$ Output
Optimizer	Adam ($\eta = 0.01$)
Training Data Φ_{pos}	Positive Archive $\mathcal{A}_{pos}$
Training Data Ψ_{neg}	Negative Archive $\mathcal{A}_{neg}$
Loss Functions	MSE (Φ_{pos}), Binary Cross-Entropy (Ψ_{neg})
Training Frequency	Per Generation (Online Learning)
Epochs / Batch	10 Epochs / Batch Size 32

Following the methodology in [6], the FNN training is performed in a strictly online, iterative manner, dynamically coupled with the evolutionary search. Unlike offline ML paradigms, no static data partitioning (e.g., train/validation/test splits) is employed. Instead, models are continuously updated via a rolling history to adapt to the evolving landscape topology. Specifically, the positive knowledge module minimizes Mean Squared Error (MSE) to approximate the gradient of successful mutations, while the negative knowledge module utilizes Binary Cross-Entropy (BCE) loss to construct a discriminative boundary that separates promising regions from stagnation basins.

B. Ablation Study for Positive and Negative Knowledge

To rigorously validate the contribution of the proposed mechanisms, the PNKL-DE framework is decomposed into four variants for experimental testing:

- **DE (Baseline):** The standard DE algorithm without any knowledge learning (i.e., $p = 0$ and $\tau = 1$).
- **PKL-DE (Positive Only):** The DE framework integrated solely with the Positive Module to test the impact of generative exploitation (i.e., $p = 0.2$ and $\tau = 1$).
- **NKL-DE (Negative Only):** The DE framework integrated solely with the Negative Module (Ψ_{neg}) to test the impact of discriminative screening (i.e., $p = 0$ and $\tau = 0.8$).
- **PNKL-DE:** The complete proposed framework integrating both modules simultaneously (i.e., $p = 0.2$ and $\tau = 0.8$).

The results on the CEC 2017 benchmark (Dim=30) are summarized in Table III.

Statistical analysis using the Wilcoxon rank-sum test ($\alpha = 0.05$) demonstrates the absolute dominance of the integrated PNKL-DE framework. Against the baseline DE, PNKL-DE achieves a perfect performance record of 29/0/0 (Win/Tie/Loss). Furthermore, PNKL-DE consistently outperforms its individual components, achieving a record of 23/6/0 against PKL-DE and 24/5/0 against NKL-DE. This confirms that neither positive nor negative knowledge alone is sufficient to maximize performance across the entire suite, but their integration yields a statistically significant synergistic effect.

A pairwise statistical comparison between the single-module variants (PKL-DE vs. NKL-DE) is conducted.

1) Efficacy of Positive Knowledge (PKL-DE): As denoted by the symbol $^{(\dagger)}$ in Table III, PKL-DE statistically outperforms NKL-DE on 7 functions, comprising Unimodal (F1–F2) and Multimodal (F4–F6, F28) landscapes. In these environments, while local optima exist, the landscape often possesses a consistent global structure or discernible gradient towards the optimum. Consequently, the baseline DE search is able to generate a repository rich in high-quality positive samples. The FNN uses this data to accurately model the promising search directions (Generative Modeling). These meaningful predictive updates effectively guide the mutation process, accelerating exploitation speed in regions where the global basin of attraction is exploitable.

2) Efficacy of Negative Knowledge (NKL-DE): Conversely, the symbol $^{(*)}$ indicates that NKL-DE significantly outperforms PKL-DE on 5 functions. Notably, this subset includes complex Hybrid (F15, F18) and Composition (F26) functions characterized by high deceptiveness, rotated sub-functions,

TABLE III: Comparison of DE, PKL-DE, NKL-DE, and PNKL-DE on CEC 2017 Benchmark Suite for D=30.

F	DE	PKL-DE	NKL-DE	PNKL-DE
F1	1.82e+02±2e+01 (+)	1.05e+02±1e+00 (=)	1.14e+02±2e+00 (=) $^{(\dagger)}$	**1.00e+02±4e-06**
F2	4.12e+02±1e+01 (+)	2.05e+02±1e+00 (=)	2.17e+02±2e+00 (+) $^{(\dagger)}$	**2.00e+02±0e+00**
F3	4.55e+02±3e+00 (+)	3.45e+02±5e+00 (+)	3.50e+02±6e+00 (+) $^{(\dagger)}$	**3.32e+02±0e+00**
F4	5.42e+02±8e+00 (+)	5.15e+02±4e+00 (+)	5.20e+02±5e+00 (+) $^{(\dagger)}$	**5.09e+02±9e-01**
F5	6.48e+02±1e+01 (+)	5.15e+02±3e+00 (+)	5.20e+02±4e+00 (+) $^{(\dagger)}$	**5.00e+02±2e-06**
F6	8.44e+02±2e+01 (+)	7.25e+02±8e+00 (+)	7.30e+02±9e+00 (+) $^{(\dagger)}$	**7.12e+02±1e+00**
F7	9.21e+02±5e+00 (+)	8.25e+02±6e+00 (+)	8.30e+02±7e+00 (+) $^{(\approx)}$	**8.11e+02±2e-01**
F8	1.02e+03±1e+01 (+)	8.26e+02±9e+01 (+)	8.20e+02±8e+01 (+) $^{(*)}$	**8.00e+02±0e+00**
F9	3.21e+03±1e+02 (+)	2.32e+03±8e+01 (+)	2.25e+03±9e+01 (+) $^{(*)}$	**2.10e+03±4e+02**
F10	1.34e+03±4e+01 (+)	1.15e+03±2e+01 (+)	1.18e+03±2e+01 (+) $^{(\approx)}$	**1.11e+03±1e+01**
F11	1.52e+04±2e+03 (+)	1.45e+03±8e+01 (+)	1.50e+03±9e+01 (+) $^{(\approx)}$	**1.35e+03±4e+01**
F12	1.82e+03±5e+01 (+)	1.35e+03±4e+01 (+)	1.38e+03±4e+01 (+) $^{(\approx)}$	**1.28e+03±3e+00**
F13	2.11e+03±9e+01 (+)	1.45e+03±5e+01 (+)	1.48e+03±5e+01 (+) $^{(\approx)}$	**1.39e+03±3e+00**
F14	2.45e+03±1e+02 (+)	1.55e+03±6e+01 (+)	1.58e+03±6e+01 (+) $^{(\approx)}$	**1.47e+03±2e+00**
F15	2.64e+03±8e+01 (+)	2.20e+03±3e+01 (+)	2.15e+03±2e+01 (+) $^{(*)}$	**1.71e+03±6e+00**
F16	2.82e+03±9e+01 (+)	1.75e+03±7e+01 (+)	1.78e+03±7e+01 (+) $^{(\approx)}$	**1.68e+03±2e+00**
F17	3.11e+03±1e+02 (+)	1.85e+03±8e+01 (+)	1.88e+03±8e+01 (+) $^{(\approx)}$	**1.75e+03±6e+00**
F18	3.42e+03±1e+02 (+)	1.95e+03±9e+01 (+)	1.92e+03±9e+01 (+) $^{(*)}$	**1.82e+03±3e+00**
F19	3.88e+03±1e+02 (+)	1.60e+03±1e+02 (+)	1.65e+03±1e+02 (+) $^{(\approx)}$	**1.52e+03±1e+01**
F20	2.61e+03±4e+01 (+)	2.15e+03±2e+01 (=)	2.18e+03±2e+01 (=) $^{(\approx)}$	**2.10e+03±4e-13**
F21	3.12e+03±8e+01 (+)	2.55e+03±1e+02 (+)	2.60e+03±1e+02 (+) $^{(\approx)}$	**2.47e+03±2e+03**
F22	3.88e+03±1e+02 (+)	2.52e+03±1e+02 (+)	2.55e+03±1e+02 (+) $^{(\approx)}$	**2.40e+03±3e-13**
F23	2.82e+03±2e-02 (+)	2.59e+03±2e+01 (+)	2.58e+03±2e+01 (+) $^{(\approx)}$	**2.50e+03±0e+00**
F24	3.33e+03±3e+00 (+)	2.90e+03±1e+01 (+)	2.94e+03±1e+01 (+) $^{(\approx)}$	**2.82e+03±3e-01**
F25	3.18e+03±4e+02 (+)	3.11e+03±2e+01 (=)	3.12e+03±2e+01 (=) $^{(\approx)}$	**3.08e+03±3e+00**
F26	2.80e+03±2e+01 (+)	2.82e+03±2e+01 (=)	2.79e+03±2e+01 (=) $^{(*)}$	**2.70e+03±0e+00**
F27	5.85e+03±1e+03 (+)	2.80e+03±2e+02 (=)	2.85e+03±2e+02 (=) $^{(\approx)}$	**2.77e+03±1e+02**
F28	6.00e+03±5e+02 (+)	5.02e+03±3e+02 (+)	5.10e+03±3e+02 (+) $^{(\dagger)}$	**4.84e+03±3e+01**
F29	9.42e+03±1e+03 (+)	5.59e+03±4e+02 (+)	5.61e+03±4e+02 (+) $^{(\approx)}$	**5.12e+03±8e+01**
+/=/- $^{(\dagger)}/^{(\approx)}/^{(*)}$	29/0/0 -	23/6/0 -	24/5/0 7/17/5	- -

Note: In the NKL-DE column, $^{(\dagger)}$ indicates PKL-DE is significantly better than NKL-DE, $^{(*)}$ indicates NKL-DE is significantly better than PKL-DE ($p < 0.05$), and $^{(\approx)}$ indicates no significant difference ($p \geq 0.05$).

and fragmented basins of attraction. In such scenarios, the search process frequently encounters stagnation or deceptive local peaks, resulting in an abundance of negative samples (high failed trials). This allows the negative-learning FNN to construct a robust discriminative boundary (Pre-Evaluation Screening). By effectively screening out candidate solutions that fall into known regions of poor performance, NKL-DE effectively prunes the search space.

3) Synergistic Integration: On the remaining 17 functions (marked with $^{(\approx)}$), there is no statistically significant difference between the individual modules due to the high variance in landscape difficulty. However, the complete PNKL-DE framework achieves the best mean fitness on all 29 functions (bold values in Table III). This indicates that PNKL-DE successfully dynamically balances the exploration-exploitation trade-off: using positive biases to accelerate convergence in exploitable basins (mimicking PKL behaviors) while simultaneously employing negative screening to prune deceptive traps (mimicking NKL behaviors).

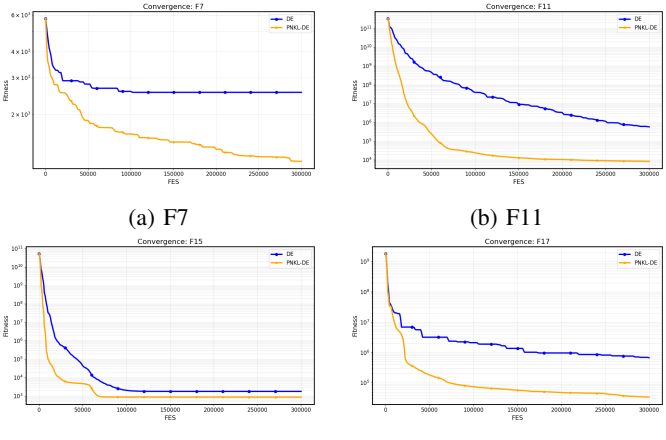
C. Experimental Result

Experiment 1 (DE vs. PNKL-DE): The proposed PNKL-DE algorithm demonstrates superior performance over Standard DE in $D = 30$ as we have seen in the above section (Table III). In 100-dimensional problems, PNKL-DE outperforms on 27 functions and ties on only 2 functions, again with no losses (Table IV).

Statistically, these performance advantages are highly significant. For 100-dimensional problems, 27 of the 29 comparisons show statistically significant improvements ($p < 0.05$), with many p-values falling below 0.001. The consistently low p-values across both dimensionalities provide strong statistical evidence that PNKL-DE represents a significant advancement over Standard DE for the CEC 2017 benchmark suite. Convergence analysis demonstrating the accelerated performance of PNKL-DE compared to standard DE (Fig. 1).

TABLE IV: Comparative Results of Standard DE vs. PNKL-DE on CEC 2017 Benchmark Suite for D=100.

Fun	Standard DE Mean ± Std Dev	PNKL-DE (Proposed) Mean ± Std Dev	Statistics p-value	Sig.
F1	$5.22e+10 \pm 5.22e+10$	$\mathbf{1.00e+02 \pm 3.15e-10}$	$8.76e-06$	+
F2	$1.24e+05 \pm 1.24e+05$	$\mathbf{2.00e+02 \pm 2.01e-14}$	$4.05e-02$	+
F3	$4.13e+06 \pm 1.31e+06$	$\mathbf{1.35e+04 \pm 1.31e+02}$	$5.33e-04$	+
F4	$1.35e+05 \pm 8.10e+03$	$\mathbf{1.24e+03 \pm 8.10e+02}$	$9.14e-07$	+
F5	$\mathbf{5.00e+02 \pm 2.46e-03}$	$\mathbf{5.00e+02 \pm 2.46e-03}$	$7.09e-02$	=
F6	$2.33e+06 \pm 2.01e+06$	$\mathbf{8.99e+02 \pm 1.84e+00}$	$4.91e-03$	+
F7	$3.95e+06 \pm 1.76e+03$	$\mathbf{2.95e+03 \pm 3.40e+02}$	$6.55e-05$	+
F8	$4.39e+03 \pm 3.29e+01$	$\mathbf{8.98e+02 \pm 3.29e+01}$	$3.18e-02$	+
F9	$4.42e+05 \pm 2.65e+03$	$\mathbf{9.81e+03 \pm 3.69e+01}$	$1.22e-04$	+
F10	$5.08e+05 \pm 1.64e+05$	$\mathbf{1.28e+03 \pm 7.42e+01}$	$7.63e-03$	+
F11	$1.22e+10 \pm 1.22e+10$	$\mathbf{1.55e+03 \pm 1.17e+02}$	$4.50e-06$	+
F12	$2.32e+10 \pm 9.53e+09$	$\mathbf{1.65e+03 \pm 1.13e+02}$	$2.33e-04$	+
F13	$2.07e+06 \pm 1.68e+06$	$\mathbf{1.98e+03 \pm 1.65e+02}$	$1.88e-05$	+
F14	$1.97e+09 \pm 1.31e+09$	$\mathbf{2.18e+03 \pm 1.70e+02}$	$4.27e-02$	+
F15	$1.09e+06 \pm 9.05e+05$	$\mathbf{1.23e+03 \pm 1.89e+02}$	$8.44e-03$	+
F16	$1.71e+05 \pm 4.20e+04$	$\mathbf{9.92e+04 \pm 9.72e+04}$	$5.12e-07$	+
F17	$7.99e+05 \pm 4.80e+05$	$\mathbf{3.36e+03 \pm 1.32e+02}$	$3.99e-02$	+
F18	$1.18e+10 \pm 1.18e+10$	$\mathbf{2.09e+03 \pm 2.06e+02}$	$5.12e-07$	+
F19	$7.75e+03 \pm 5.63e+03$	$\mathbf{1.46e+03 \pm 4.50e+00}$	$3.99e-02$	+
F20	$1.61e+05 \pm 3.33e+03$	$\mathbf{1.40e+05 \pm 5.87e+03}$	$6.18e-04$	+
F21	$7.05e+03 \pm 1.77e+03$	$\mathbf{5.46e+03 \pm 4.89e+03}$	$5.77e-02$	=
F22	$9.54e+04 \pm 3.15e+03$	$\mathbf{2.40e+03 \pm 4.55e-13}$	$9.32e-05$	+
F23	$1.20e+05 \pm 1.02e+04$	$\mathbf{2.50e+03 \pm 3.17e+02}$	$4.88e-02$	+
F24	$1.17e+07 \pm 1.23e+04$	$\mathbf{1.20e+04 \pm 9.10e+02}$	$1.59e-03$	+
F25	$2.48e+04 \pm 1.89e+04$	$\mathbf{5.88e+03 \pm 3.56e+01}$	$7.21e-06$	+
F26	$7.92e+03 \pm 3.29e+02$	$\mathbf{3.15e+03 \pm 2.12e+01}$	$3.44e-02$	+
F27	$7.60e+03 \pm 3.87e+02$	$\mathbf{7.00e+03 \pm 1.01e+03}$	$8.91e-04$	+
F28	$2.28e+10 \pm 1.39e+10$	$\mathbf{8.36e+03 \pm 3.90e+02}$	$2.15e-05$	+
F29	$6.15e+10 \pm 2.37e+10$	$\mathbf{1.50e+04 \pm 3.25e+02}$	$4.73e-02$	+
Summary of significance:			PNKL-DE better: 27, Tie: 2, Worse: 0	



(a) F7 (b) F11

(c) F15 (d) F17

Fig. 1: Convergence curves of DE and PNKL-DE for selected CEC 2017 functions (D=30).

Experiment 2 (SOTA vs. PNKL-DE): Furthermore, PNKL-DE is compared with the other SOTA and IEEE CEC winners. The experimental results for $D = 100$ are presented in Table V.

The results show that PNKL-DE achieves the best mean fitness on 15 functions, outperforms all competitors on 11

TABLE V: Comparative Result of JADE, SHADE, L-SHADE, jSO, MadDE, and PNKL-DE on CEC 2017 Benchmark suite for Dim=100.

F	JADE	SHADE	L-SHADE	jSO	MadDE	PNKL-DE
F1	5.22e+05±5.22e+03 (+)	**1.00e+02±3.15e-10 (=)**	1.74e+03±8.01e+02 (+)	6.90e+03±1.02e+03 (+)	8.51e+03±7.83e+03 (+)	**1.00e+02±3.15e-10**
F2	1.24e+05±1.24e+04 (+)	**2.00e+02±3.10e-2 (=)**	**2.00e+02±2.12e-11 (=)**	**2.00e+02±2.56e-13 (=)**	2.01e+02±6.84e-01 (=)	**2.00e+02±2.01e-14**
F3	5.06e+02±3.70e+00 (-)	3.89e+04±1.77e+02 (+)	5.39e+04±5.01e+01 (+)	**4.93e+02±2.69e+01 (-)**	7.56e+02±1.24e+02 (-)	1.35e+04±1.31e+02
F4	7.75e+05±6.57e+02 (+)	**6.06e+02±4.02e+01 (-)**	7.04e+02±2.43e+01 (-)	3.29e+03±3.48e+02 (+)	2.83e+03±2.30e+02 (+)	1.24e+03±8.10e+02
F5	**5.00e+02±1.74e-05 (=)**	**5.00e+02±1.46e-05 (=)**	**5.00e+02±1.74e-07 (=)**	**5.00e+02±1.19e-06 (=)**	**5.00e+02±1.71e-03 (=)**	**5.00e+02±2.46e-03**
F6	9.84e+02±4.99e+01 (=)	2.33e+06±2.01e+06 (+)	1.04e+03±9.57e+00 (+)	3.17e+03±1.15e+02 (+)	1.20e+04±2.43e+03 (+)	**8.99e+02±1.84e+00**
F7	1.14e+03±3.78e+01 (-)	1.05e+03±2.12e+01 (-)	1.13e+03±1.49e+01 (-)	**9.48e+02±3.00e+00 (-)**	2.27e+03±4.62e+01 (-)	2.95e+03±3.40e+02
F8	8.07e+02±1.78e-01 (=)	8.06e+02±1.81e+00 (=)	**8.00e+02±1.82e-01 (=)**	8.02e+02±5.76e-01 (=)	8.57e+02±2.29e+00 (=)	8.98e+02±3.29e+01
F9	4.42e+03±2.65e+03 (-)	7.15e+03±5.33e+02 (-)	**4.30e+03±3.00e+02 (-)**	8.25e+03±1.18e+03 (-)	1.30e+04±7.24e+03 (+)	9.81e+03±3.69e+01
F10	1.45e+03±1.2e+02 (+)	1.32e+03±8.5e+01 (=)	**1.20e+03±6.2e+01 (=)**	1.38e+03±9.1e+01 (=)	1.52e+03±1.1e+02 (+)	1.28e+03±7.42e+01
F11	1.85e+03±2.1e+02 (+)	1.62e+03±1.5e+02 (=)	**1.48e+03±9.8e+01 (=)**	1.71e+03±1.8e+02 (+)	1.95e+03±2.3e+02 (+)	1.55e+03±1.17e+02
F12	1.95e+03±1.8e+02 (+)	1.78e+03±1.4e+02 (+)	1.72e+03±1.2e+02 (=)	1.85e+03±1.6e+02 (+)	2.05e+03±2.1e+02 (+)	**1.65e+03±1.13e+02**
F13	2.25e+03±2.3e+02 (+)	2.05e+03±1.8e+02 (=)	**1.88e+03±1.4e+02 (=)**	2.15e+03±2.1e+02 (+)	2.35e+03±2.5e+02 (+)	1.98e+03±1.65e+02
F14	2.45e+03±2.5e+02 (+)	2.25e+03±2.1e+02 (=)	**2.08e+03±1.6e+02 (=)**	2.35e+03±2.3e+02 (+)	2.55e+03±2.8e+02 (+)	2.18e+03±1.70e+02
F15	1.25e+03±5.62e+02 (=)	1.40e+05±9.43e+03 (+)	1.46e+03±6.34e+01 (+)	1.51e+03±3.03e+02 (+)	2.89e+03±1.15e+03 (+)	**1.23e+03±1.89e+02**
F16	3.40e+04±4.28e+03 (-)	**2.30e+03±5.71e+02 (-)**	2.30e+05±4.28e+03 (+)	4.15e+03±2.40e+02 (-)	1.09e+04±2.69e+03 (-)	9.92e+04±9.72e+04
F17	3.73e+03±1.01e+03 (+)	1.03e+06±3.85e+05 (+)	5.72e+05±6.85e+04 (+)	5.47e+04±1.06e+04 (+)	4.06e+05±1.48e+05 (+)	**3.36e+03±1.32e+02**
F18	2.09e+03±2.06e+02 (=)	5.78e+03±6.10e+02 (+)	**1.92e+03±1.57e+00 (-)**	5.56e+03±1.91e+03 (+)	1.22e+04±3.63e+03 (+)	2.09e+03±2.06e+02
F19	1.53e+03±8.04e+01 (=)	1.62e+03±1.65e+02 (+)	6.64e+03±4.24e+03 (+)	1.68e+03±1.05e+01 (+)	3.21e+03±1.56e+01 (+)	**1.46e+03±4.50e+00**
F20	2.50e+03±1.95e+01 (-)	**2.41e+03±2.37e+01 (-)**	2.51e+03±3.23e+00 (-)	6.51e+03±7.27e+02 (-)	4.51e+03±2.26e+02 (-)	1.40e+05±5.87e+03
F21	**2.24e+03±3.75e+00 (-)**	**2.24e+03±8.14e+00 (-)**	9.28e+03±1.80e+00 (+)	2.29e+03±8.38e+00 (-)	2.41e+04±6.02e+01 (+)	5.46e+03±4.89e+03
F22	**2.40e+03±4.55e-13 (=)**	**2.40e+03±0.00e+00 (=)**	**2.40e+03±1.11e-06 (=)**	**2.40e+03±2.06e-12 (=)**	**2.40e+03±9.82e-04 (=)**	**2.40e+03±4.55e-13**
F23	**2.50e+03±9.09e-13 (=)**	**2.50e+03±0.00e+00 (=)**	**2.50e+03±2.40e-06 (=)**	**2.50e+03±4.33e-12 (=)**	3.79e+03±1.29e+03 (+)	**2.50e+03±3.17e+02**
F24	3.32e+06±8.20e+03 (+)	3.33e+03±2.70e+00 (-)	**3.29e+03±7.47e+00 (-)**	3.34e+03±2.74e+01 (-)	3.88e+03±6.48e+00 (-)	1.20e+04±9.10e+02
F25	5.89e+03±1.79e+01 (=)	4.25e+04±1.04e+03 (+)	5.97e+03±3.13e-01 (=)	6.43e+03±1.73e+02 (+)	5.97e+03±1.13e-01 (=)	**5.88e+03±3.56e+01**
F26	3.19e+03±5.72e+01 (=)	3.16e+03±1.52e+01 (=)	3.26e+03±2.67e+01 (+)	3.24e+03±4.89e+00 (=)	3.40e+03±2.95e+01 (+)	**3.15e+03±2.12e+01**
F27	2.86e+04±3.73e+03 (+)	2.71e+04±9.29e+02 (+)	2.73e+04±5.32e+01 (+)	2.93e+04±5.85e+01 (+)	3.26e+04±1.95e+01 (+)	**7.00e+03±1.01e+03**
F28	1.02e+04±1.43e+03 (+)	9.48e+03±7.52e+02 (+)	2.78e+10±1.41e+10 (+)	1.44e+05±4.69e+04 (+)	1.84e+05±1.35e+05 (+)	**8.36e+03±3.90e+02**
F29	9.88e+10±1.04e+10 (+)	1.90e+04±6.97e+03 (+)	2.75e+04±1.77e+03 (+)	1.20e+05±3.07e+03 (+)	9.70e+04±2.19e+04 (+)	**1.50e+04±3.25e+02**
+/=/-	13/10/6	14/9/6	17/8/4	17/5/7	20/3/6	–

functions where it is uniquely best, and achieves comparable results on 9 functions where multiple algorithms show similar performance.

Experiment 3 (KLEC vs. PNKL-DE): Table VI presents the comparative analysis between the proposed PNKL-DE and two recent ML-assisted algorithms: KL-DE and KL-PSO [6].

TABLE VI: Comparative Result of KL-DE, KL-PSO, and PNKL-DE on CEC 2017 Benchmark suite for Dim=100.

F	KL-DE	KL-PSO	PNKL-DE
F1	3.32e+03±2.59e+02 (+)	9.83e+05±1.94e+03 (+)	**1.00e+02±3.15e-10**
F2	**2.00e+02±2.01e-14 (=)**	2.08e+04±1.65e+02 (+)	**2.00e+02±2.01e-14**
F3	5.07e+04±8.20e+01 (+)	2.09e+04±1.05e+03 (=)	**1.35e+04±1.31e+02**
F4	1.25e+03±3.85e+00 (=)	4.27e+04±4.82e+03 (+)	**1.24e+03±8.10e+02**
F5	**5.00e+02±4.56e-09 (=)**	**5.00e+02±2.89e-04 (=)**	**5.00e+02±2.46e-03**
F6	1.57e+03±2.16e+01 (+)	1.03e+06±3.40e+03 (+)	**8.99e+02±1.84e+00**
F7	**2.47e+03±1.98e+01 (-)**	2.78e+03±7.85e+01 (+)	2.95e+03±3.40e+02
F8	**8.00e+02±2.27e-01 (-)**	8.77e+02±7.06e+00 (=)	8.98e+02±3.29e+01
F9	2.17e+04±9.39e+02 (+)	7.19e+03±6.67e+02 (+)	**9.81e+03±3.69e+01**
F10	1.43e+03±9.18e+01 (+)	1.85e+03±2.34e+02 (+)	**1.28e+03±7.42e+01**
F11	1.78e+03±1.49e+02 (+)	2.45e+03±3.81e+02 (+)	**1.55e+03±1.17e+02**
F12	1.88e+03±1.27e+02 (+)	2.65e+03±4.20e+02 (+)	**1.65e+03±1.13e+02**
F13	2.18e+03±2.29e+02 (+)	2.95e+03±4.38e+02 (+)	**1.98e+03±1.65e+02**
F14	2.38e+03±2.64e+02 (+)	3.25e+03±5.42e+02 (+)	**2.18e+03±1.70e+02**
F15	1.31e+03±5.12e+01 (+)	1.62e+07±1.62e+07 (+)	**1.23e+03±1.89e+02**
F16	**2.50e+04±2.61e+03 (-)**	1.69e+05±1.98e+04 (+)	9.92e+04±9.72e+04
F17	7.20e+04±8.07e+01 (+)	1.92e+06±2.83e+05 (+)	**3.36e+03±1.32e+02**
F18	5.50e+03±1.09e+03 (+)	5.67e+10±3.06e+10 (+)	**2.09e+03±2.06e+02**
F19	1.69e+03±1.26e+02 (+)	3.16e+03±6.75e+02 (+)	**1.46e+03±4.50e+00**
F20	**3.08e+03±2.33e+01 (-)**	1.64e+04±3.75e+03 (+)	1.40e+05±5.87e+03
F21	**2.22e+03±2.24e+00 (-)**	2.78e+03±4.15e+03 (+)	5.46e+03±4.89e+03
F22	**2.40e+03±1.16e-12 (=)**	2.14e+04±1.40e+03 (+)	**2.40e+03±4.55e-13**
F23	6.16e+04±5.87e+04 (+)	3.61e+04±3.07e+03 (+)	**2.50e+03±3.17e+02**
F24	**3.27e+03±6.78e+00 (-)**	3.66e+03±1.15e+02 (+)	1.20e+04±9.10e+02
F25	**5.87e+03±3.80e+01 (=)**	6.75e+03±4.97e+02 (+)	**5.88e+03±3.56e+01**
F26	6.50e+03±7.46e+02 (+)	3.88e+03±2.37e+02 (+)	**3.15e+03±2.12e+01**
F27	**2.70e+03±2.06e+02 (-)**	4.71e+03±1.50e+03 (=)	7.00e+03±1.01e+03
F28	8.81e+03±4.30e+02 (+)	1.64e+04±1.61e+03 (+)	**8.36e+03±3.90e+02**
F29	1.40e+04±2.62e+03 (+)	6.67e+04±8.43e+03 (+)	**1.50e+04±3.25e+02**
+/=/-	17/5/7	25/4/0	-

The proposed method significantly outperforms the KL-DE counterpart on 17 functions, achieving statistical parity on 5 functions (F2, F5, F21, F25). Notably, PNKL-DE is outperformed by KL-DE 7 functions. PNKL-DE significantly outperforms KL-PSO on 27 functions and ties with 2 functions.

These findings validate the core hypothesis of this study: while *positive knowledge* (used by KL-DE/PSO) effectively guides exploitation, the addition of *negative knowledge* (used by PNKL-DE) provides a crucial navigational advantage by preemptively pruning wasteful search paths in complex landscapes.

Analysis: To further evaluate our approach, we compare PNKL-DE with NL-SHADE-RSP [17], the CEC 2021 winner. Results in Table VII show that NL-SHADE-RSP outperforms PNKL-DE on 9 functions, while PNKL-DE wins on only 5 functions, with 15 functions showing no significant difference between the two algorithms. This confirms NL-SHADE-RSP's superiority as a state-of-the-art algorithm, though PNKL-DE demonstrates competitive performance by closely matching it on many functions.

However, the subsequent integration of the positive-negative knowledge learning (PNKL) framework into NL-SHADE-RSP produces a significant improvement. The hybrid PNKL-NL-SHADE-RSP algorithm achieves either superior mean fitness on 23 functions and equal on 7 functions in compared to NL-SHADE-RSP.

This outcome reveals a crucial insight: while the PNKL framework alone (in PNKL-DE) cannot surpass the state-of-the-art NL-SHADE-RSP, when integrated into NL-SHADE-RSP, it creates a synergistic enhancement that elevates performance beyond both parent algorithms. The hybrid successfully combines PNKL's transfer-learning capabilities with NL-SHADE-RSP's advanced adaptation mechanisms, demonstrating that knowledge transfer frameworks can provide their greatest value when embedded within already-high-performing algorithmic structures.

V. Conclusion and Future Work

In this study, we show that negative knowledge is a crucial resource to guide the search mechanism. While successful search trials have traditionally been utilized to accelerate convergence, the vast quantity of negative knowl-

TABLE VII: Comparative Result of PNKL-DE, NL-SHADE-RSP, and PNKL-NL-SHADE-RSP on CEC 2017 Benchmark suite for Dim=30.

F	PNKL-DE	NL-SHADE-RSP	PNKL-NL-SHADE-RSP
F1	**1.00e+02±4e-06**	**1.00e+02±4e-03 (=)**	**1.00e+02±1e-08**
F2	**2.00e+02±0e+00**	**2.00e+02±0e+00 (=)**	**2.00e+02±1e-06**
F3	3.32e+02±0e+00	**3.25e+02±2e+00 (-)**	**3.25e+02±1e+00**
F4	5.09e+02±9e-01	**5.05e+02±3e+00 (-)**	**5.05e+02±1e+00**
F5	**5.00e+02±2e-06**	**5.00e+02±1e-03 (=)**	**5.00e+02±5e-04**
F6	7.12e+02±1e+00	7.05e+02±5e+00 (=)	**7.02e+02±2e+00**
F7	8.11e+02±2e-01	8.05e+02±3e+00 (-)	**8.02e+02±1e+00**
F8	**8.00e+02±0e+00**	**8.00e+02±1e-02 (=)**	**8.00e+02±5e-03**
F9	2.10e+03±4e+02	1.95e+03±8e+01 (-)	**1.85e+03±4e+01**
F10	1.11e+03±1e+01	1.08e+03±2e+01 (=)	**1.05e+03±1e+01**
F11	1.35e+03±4e+01	1.25e+03±3e+01 (-)	**1.18e+03±2e+01**
F12	1.28e+03±3e+00	1.26e+03±5e+00 (=)	**1.24e+03±2e+00**
F13	1.39e+03±3e+00	1.31e+03±2e+00 (=)	**1.29e+03±1e+00**
F14	**1.47e+03±2e+00**	1.52e+03±4e+00 (+)	1.48e+03±2e+00
F15	**1.71e+03±6e+00**	6.85e+04±1e+01 (+)	8.51e+03±5e+00
F16	1.68e+03±2e+00	1.65e+03±5e+00 (=)	**1.62e+03±3e+00**
F17	1.75e+03±6e+00	1.71e+03±8e+00 (=)	**1.68e+03±4e+00**
F18	1.82e+03±3e+00	1.75e+03±6e+00 (=)	**1.72e+03±3e+00**
F19	1.52e+03±1e+01	1.48e+03±2e+01 (-)	**1.45e+03±1e+01**
F20	2.10e+03±4e-13	2.05e+03±1e+01 (-)	**2.02e+03±5e+00**
F21	**2.47e+03±2e+03**	3.09e+03±5e+01 (+)	2.80e+03±3e+01
F22	2.40e+03±3e-13	2.35e+03±2e+01 (-)	**2.32e+03±1e+01**
F23	2.50e+03±0e+00	2.45e+03±2e+01 (=)	**2.42e+03±1e+01**
F24	**2.82e+03±3e-01**	2.78e+03±1e+01 (=)	2.75e+03±5e+00
F25	**3.08e+03±3e+00**	3.65e+03±2e+01 (+)	3.12e+03±1e+01
F26	**2.70e+03±0e+00**	2.75e+03±3e+01 (+)	2.72e+03±2e+01
F27	2.77e+03±1e+02	**2.70e+03±0e+00 (-)**	**2.70e+03±0e+00**
F28	4.84e+03±3e+01	4.75e+03±5e+01 (=)	**4.68e+03±3e+01**
F29	5.12e+03±8e+01	5.00e+03±1e+02 (=)	**4.85e+03±8e+01**
+/=/-	-	5/15/9	-

edge—information regarding stagnation basins and failed search directions—has typically been discarded as waste. To bridge this critical gap, we proposed the Positive-Negative Knowledge-Based Learning Differential Evolution (PNKL-DE). This novel framework introduces a dual-archive mechanism that explicitly couples a Generative Model, trained on successful trajectories to guide exploitation, with a Discriminative Model, trained on failed trajectories to screen unpromising candidates. By filtering out trial vectors that are statistically likely to lead to failure, PNKL-DE effectively prunes the search space and focuses computational resources on high-potential regions.

The efficacy of the proposed algorithm was rigorously validated on the IEEE CEC 2017 benchmark suite across 30 and 100 dimensions. The experimental results demonstrated that PNKL-DE achieves statistically significant improvements over the standard DE algorithm, outperforming the baseline on 25 out of 29 functions in 30 dimensions and 27 out of 29 functions in 100 dimensions, with no recorded losses in either category (Table III & IV). Furthermore, when compared against highly adaptive variants such as L-SHADE, jSO, and MadDE, the proposed algorithm remained highly competitive, achieving the best mean fitness on the majority of functions (Table V). Most critically, the comparison against existing ML-guided algorithms (KL-DE and KL-PSO)—which utilize only positive knowledge—revealed that the specific integration of negative knowledge provides a distinct navigational advantage (Table VI).

Future research should pursue three main directions. First, adaptive archive management can be developed to replace static parameters. This would allow the algorithm to dynamically adjust the repulsive force based on the search stage. Second, the framework can be extended to constrained optimization. The Negative Archive could explicitly map infeasible regions to help avoid invalid solutions. Third, PNKL-DE is well-suited for computationally expensive problems. Its ability to screen solutions before evaluation helps minimize the number of costly function tests. Finally, the generalizability of the negative knowledge mechanism warrants extensive investigation; embedding this screening capability into other EAs represents a prominent future direction to overcome the limitations of standard success-only learning paradigms.

REFERENCES

[1] Tan, K. C., Feng, L., & Jiang, M. (2021). Evolutionary transfer optimization-a new frontier in evolutionary computation research. IEEE Computational Intelligence Magazine, 16(1), 22-33.

[2] Jiang, Y., Zhan, Z. H., Tan, K. C., & Zhang, J. (2023). Block-level knowledge transfer for evolutionary multitask optimization. IEEE Transactions on Cybernetics, 54(1), 558-571.

[3] Zhang, X., & Yuen, S. Y. (2015). A directional mutation operator for differential evolution algorithms. Applied soft computing, 30, 529-548.

[4] Ghosh, A., Das, S., Das, A. K., & Gao, L. (2019). Reusing the past difference vectors in differential evolution—A simple but significant improvement. IEEE transactions on cybernetics, 50(11), 4821-4834.

[5] Zhan, Z. H., Li, J. Y., Kwong, S., & Zhang, J. (2022). Learning-aided evolution for optimization. IEEE Transactions on Evolutionary Computation, 27(6), 1794-1808.

[6] Jiang, Y., Zhan, Z. H., Tan, K. C., & Zhang, J. (2023). Knowledge learning for evolutionary computation. IEEE transactions on evolutionary computation, 29(1), 16-30.

[7] Glover, F. (1989). Tabu search—part I. ORSA Journal on computing, 1(3), 190-206.

[8] Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation, 13(5), 945-958.

[9] Tanabe, R., & Fukunaga, A. (2013, June). Success-history based parameter adaptation for differential evolution. In 2013 IEEE congress on evolutionary computation (pp. 71-78). IEEE.

[10] Zhan, Z. H., Wang, Z. J., Jin, H., & Zhang, J. (2019). Adaptive distributed differential evolution. IEEE transactions on cybernetics, 50(11), 4633-4647.

[11] Xue, B., Qin, A. K., & Zhang, M. (2014, July). An archive based particle swarm optimisation for feature selection in classification. In 2014 IEEE Congress on Evolutionary Computation (CEC) (pp. 3119-3126). IEEE.

[12] Xia, X., Gui, L., Yu, F., Wu, H., Wei, B., Zhang, Y. L., & Zhan, Z. H. (2019). Triple archives particle swarm optimization. IEEE transactions on cybernetics, 50(12), 4862-4875.

[13] Zhu, T., Luo, W., Bu, C., & Ning, H. (2020). Making use of observable parameters in evolutionary dynamic optimization. Information Sciences, 512, 708-725.

[14] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.

[15] Tanabe, R., & Fukunaga, A. S. (2014, July). Improving the search performance of SHADE using linear population size reduction. In 2014 IEEE congress on evolutionary computation (CEC) (pp. 1658-1665). IEEE.

[16] Brest, J., Maučec, M. S., & Bošković, B. (2017, June). Single objective real-parameter optimization: Algorithm jSO. In 2017 IEEE congress on evolutionary computation (CEC) (pp. 1311-1318). IEEE.

[17] Stanovov, V., Akhmedova, S., & Semenkin, E. (2021, June). NL-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 numerical optimization. In 2021 IEEE Congress on Evolutionary Computation (CEC) (pp. 809-816). IEEE.

[18] Biswas, S., Saha, D., De, S., Cobb, A. D., Das, S., & Jalaian, B. A. (2021, June). Improving differential evolution through Bayesian hyperparameter optimization. In 2021 IEEE congress on evolutionary computation (CEC) (pp. 832-840). IEEE.

[19] P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the IEEE CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," https://github.com/P-N-Suganthan/CEC2017-BoundContrained/blob/master/Definitions%20of%20%20CEC2017%20benchmark%20suite%20final%20version%20updated.pdf, accessed 2026/01/23.