Physics-supervised deep learning–based optimization (PSDLO) with accuracy and efficiency

Xiaowen Li[a,b], Lige Chang[a,b], Yajun Cao[a,b], Junqiang Lu[c], Xiaoli Lu[d], and Hanqing Jiang[a,b,e,1]

Edited by Yonggang Huang, Northwestern University, Glencoe, IL; received May 31, 2023; accepted July 21, 2023

Identifying efficient and accurate optimization algorithms is a long-desired goal for the scientific community. At present, a combination of evolutionary and deep-learning methods is widely used for optimization. In this paper, we demonstrate three cases involving different physics and conclude that no matter how accurate a deep-learning model is for a single, specific problem, a simple combination of evolutionary and deep-learning methods cannot achieve the desired optimization because of the intrinsic nature of the evolutionary method. We begin by using a physics-supervised deep-learning optimization algorithm (PSDLO) to supervise the results from the deep-learning model. We then intervene in the evolutionary process to eventually achieve simultaneous accuracy and efficiency. PSDLO is successfully demonstrated using both sufficient and insufficient datasets. PSDLO offers a perspective for solving optimization problems and can tackle complex science and engineering problems having many features. This approach to optimization algorithms holds tremendous potential for application in real-world engineering domains.

physics-supervise | deep learning | evolutionary algorithm | accuracy | efficiency

Optimization algorithms that are both efficient and accurate are an area of research of considerable value and interest (1–6). Evolutionary methods, such as genetic algorithms (GA) (7), particle swarm optimization (PSO) (8), and ant colony optimization (ACO) (9) are commonly used global optimization algorithms. These methods simulate biological evolution processes and the associated collective behaviors and can address complex nonlinear problems that require global search with multiple objectives (e.g., fitness functions) and constraints. During the evolution of the optimization problem that use these methods, the fitness function needs to be calculated at every iteration step to evaluate the status of the evolution. This is achieved by solving physics-based governing equations through various means, such as finite element method (FEM) or first-principles calculations, depending on the nature of the problem (10–16). These calculations are often time-consuming and result in a relatively slow convergence for the entire evolutionary optimization process, even though the results are accurate. To address the efficiency of the physics-based solving methods, deep learning–based neural networks (NN) have been utilized to replace the physics-based governing equations (17–23). NNs have shown to be highly efficient and accurate in predicting specific problems, leading to a discussion about whether NN can, in fact, replace physics-based simulations (24). However, different from the single-problem prediction using NN methods, evolutionary algorithms–based optimization problems combined with NN methods involve many iterative steps, and thus many executions of the NN calculations in which certain problem-specific factors would be amplified/diminished and inherited (25). Consequently, a significant question to ask is: Can a well-trained deep learning-based NN method that has been tested successfully for single-problem prediction maintain its accuracy and be harnessed when applied to an evolutionary optimization method to achieve simultaneous efficiency and accuracy?

In this paper, we examined three cases based on different physics, namely solid mechanics, acoustics, and solid-state physics, at different length scales and demonstrated that even a very well-trained deep learning-based NN method with a large amount of training data (squared correlation coefficient $R^2 = 0.99$), and capable of accurately predicting many individual problems can still lead to a significant deviation from the optimal result (calibrated by the physics-based simulations) when applied for evolutionary optimization algorithms. We identified that small inaccuracies from the NN methods that are insignificant for individual predictions would be inherited in the evolutionary methods, and the species that favors the fitness functions would be amplified and gradually dominate throughout generations of optimization, while at the same time disfavor the species that are opposite to the fitness function, thereby resulting in substantial discrepancy. This natural tendency is independent of the accuracy of the NN methods but is inevitable to the evolutionary algorithms.

Significance

The scientific and engineering field has long sought an optimization method that is both efficient and accurate. While combining evolutionary algorithms with deep-learning methods offers a viable solution for complex problems, the simple combination does not achieve accurate and efficient optimization due to the nature of evolutionary principles, even with well-trained deep-learning models. We introduce a physics-supervised deep-learning optimization (PSDLO) algorithm that significantly improves convergence speed while maintaining optimization accuracy. PSDLO's mechanism and demonstration are clearly explained and verified, presenting an ideal optimization method for various science and engineering applications.

Author contributions: X. Li and H.J. designed research; X. Li and L.C. performed research; X. Li, J.L., X. Lu, and H.J. contributed new reagents/analytic tools; X. Li, Y.C., and H.J. analyzed data; and X. Li and H.J. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: hanqing.jiang@westlake.edu.cn.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2309062120/-/DCSupplemental.

Published August 21, 2023.

To correct the inaccuracy problem by simply combining evolutionary algorithms and NN methods, a physics-supervised deep-learning algorithm for efficient and accurate optimization (PSDLO) was developed (Fig. 1*A*). In this method, physics is not only used to provide data to the NN model, but also plays a crucial role in supervising the evolutionary process during each iteration. Quantitative comparison between the present PSDLO and the nonphysics-supervised NN combined with evolutionary methods (Fig. 1*B*) clearly showed that by supervising the NN-based optimization method, erroneous data or features (i.e., species) can be identified and properly treated, thereby achieving a balance between accuracy and efficiency (Fig. 1*C*). The present physics-supervised deep-learning method for evolutionary optimization is generic and can be applied to many problems that involve inheritance, mutation, and swarm behavior over the generations.

Results

Problem Statement. Fig. 1*A* illustrates three scenarios: a) a pure evolutionary method ("EM") in which the fitness function is evaluated by solving physics-based governing equations in every iteration step; b) an evolutionary method combined with deep-learning method ("EM+DL") in which physics-based simulations provide data to train the deep-learning method, which is followed by the trained deep-learning method that evaluates the fitness function in the evolutionary method; and c) the present PSDLO method in which physics provides data to train the deep-learning method and supervises the evolutionary process based on the evaluation results from the deep-learning method. Here, the "physics" term is not limited to a specific subject and can be, for example, solid mechanics, acoustics, or solid-state physics, ranging from meters to nanometers. For demonstration purposes, three cases, namely, designing a bistable structure to achieve optimal snap-through behavior, optimizing the sound barrier performance, and maximizing the piezoelectric coupling coefficient using strain engineering are presented (Fig. 1*B*). The computational methods used can be FEM or density functional theory (DFT). The fitness functions for the three cases are the ratio between the backward and forward snapping forces $|F_2/F_1|$, average transmission loss TL_{avg}, and the out-of-plane piezoelectric coupling coefficient e_{33}, respectively. Details can be found in *Materials and Methods* section.

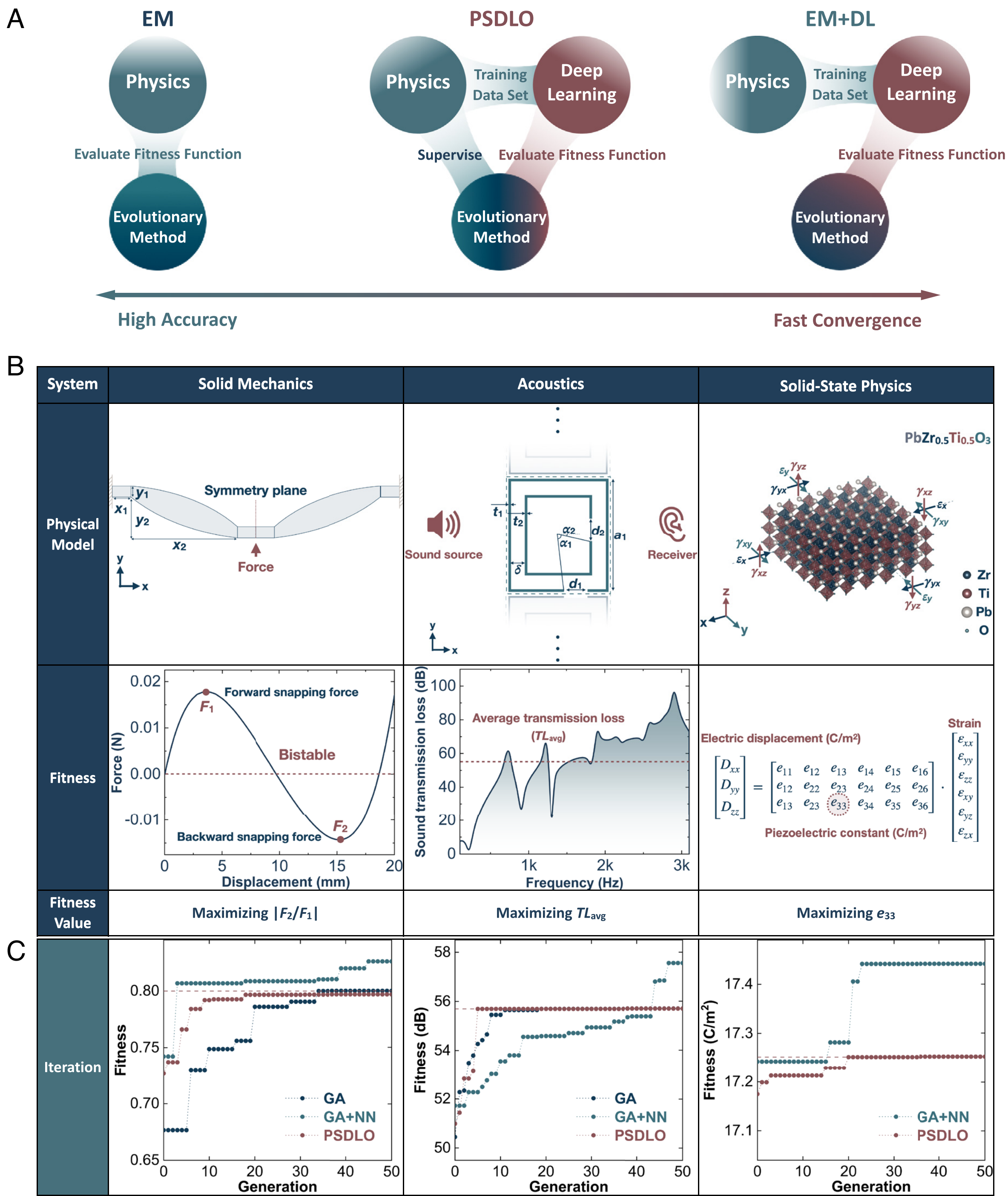
Though DL methods include many options [e.g., convolutional neural networks (CNN), recurrent neural networks (RNN), and generative adversarial networks (GAN)] (26), for the sake of discussion, only fully connected neural network (NN) method, in which the input layer communicates with the physical model by extracting its features (e.g., geometrical and physical parameters), was used in this study (*SI Appendix*, Fig. S1). Then, the hidden layers capture the complex relationships and interactions between the input features, and the output layer, which is also the fitness function for the evolutionary algorithm, reflects the results of this physical problem.

Similarly, the evolutionary methods also include many algorithms, such as GA, PSO, ACO. Here, we present only GA as an example in the main text although another method (i.e., PSO) was also used in the study. GA is an optimization method based on the principles of natural selection and genetics (27). As shown in *SI Appendix*, Fig. S2, by treating physical features as genes, GA involves cross-over and mutation to create individuals with different features. By evaluating and then sorting the performance of individuals based on the fitness function, the ones with better performance (i.e., higher value of fitness function) have higher chances to pass their features to their offspring individuals. PSO is a population-based heuristic optimization algorithm, where each "particle" in the PSO algorithm represents a potential solution, and particles move through the search space to find the optimal solution. Similarly, particles with higher fitness have a greater influence on the overall population behavior. The main steps of the PSO algorithm can be found in *Materials and Methods* section.

Deep Learning–Based Evolutionary Method without Physics Supervision. To apply an NN-based GA method (i.e., GA+NN), we first need to train the NN model based on the data generated from the physics-based simulations. *SI Appendix*, Fig. S4 shows the distribution and preparation time of datasets (row 1), and the performance and training time of the NNs (row 2) (*Materials and Methods*). The dataset preparation times, ranging from several hours to hundreds of days indicate the varying complexity of these cases. The coefficient of determination [R^2(test)] of the test set is used to measure the predictive capability of the NN model, which ranges from 0.99 to 0.76, representing NN models with different quality. Given that GA with fitness function evaluated by solving physics-based governing equations provides the most accurate results, we consider "GA" as the accurate and optimal benchmark in these cases.

To process GA+NN, the initial population (i.e., parent population) is obtained by randomly sampling the feasible parameter space of the physical model's features. The number of populations depends on the feature size. Practically, the population size normally takes 10 to 50 times more than the feature sizes (28). For this bistable problem, there are four features (x_1, x_2, y_1, and y_2 in Fig. 1*B*) and 50 individuals in each population. For the acoustic problem, there are eight geometrical features (Fig. 1*B*) and thus more individuals (i.e., 200) in each generation. Using the NN acquired in the first step, the fitness function of the parent population is calculated. Subsequently, we apply the GA's selection [via roulette wheel method (28)], cross-over, and mutation processes to obtain the initial offspring population. Their performance is then evaluated using the trained NN model for fast prescreening. The same process repeats until reaching convergence. As shown in Fig. 1*C*, GA+NN (in teal) demonstrates a significant advantage in computational efficiency, e.g., about 200 to 1,000 times faster than GA (in dark blue) depending on different physics and feature sizes. For problems with more feature sizes (e.g., the acoustic problem), the gain in computational efficiency is more obvious, from 2.75 d using the GA method reduced to 3.45 min using GA+NN, i.e., close to 1,150 times faster. However, the results predicted by the GA+NN method clearly deviate from the actual values. The more feature sizes, the larger the deviation. For the acoustic problem, the optimal TL_{avg} predicted by GA+NN is 57.56 dB, which seems very close to the optimal result predicted by the GA model (55.70 dB). However, the actual TL_{avg} for the configuration given by GA+NN is just 47.89 dB, more than 16.8% deviation from its predicted value. Clearly, GA+NN is efficient by sacrificing accuracy.

Fig. 2 analyzes the underlying mechanism of why GA+NN leads to deviation. Fig. 2*A* depicts the change in the fitness function of the bistable problem for all individuals using the GA+NN during the iterative process. The *y* axis shows their fitness predicted by the NN model, while the colors of these dots show the values of these fitness functions obtained by FEM, as the benchmark, with the darker color for larger value and the lighter color for smaller value. It is clear now that due to the inevitable deviation of the NN-predicted fitness from the ones obtained from physics-based simulations, many individuals display color artifacts, e.g., dots taking a higher position in *y* axis do not show darker color as they are supposed to do. There are two types of artifacts or deviations,



Algorithm	GA	GA+NN	PSDLO	GA	GA+NN	PSDLO	GA	GA+NN	PSDLO
Prediction	0.80	0.83	0.80	55.70 dB	57.56 dB	55.70 dB	Very Long	17.44 C/m²	17.25 C/m²
Actual		0.76			47.89 dB			17.06 C/m²	
Time	3.72 hours	52.37 seconds	22.33 minutes	2.75 days	3.45 minutes	1.65 hours		52.48 seconds	54.31 hours

Fig. 1. Overview of three methods for optimization with and without deep-learning methods involved. (*A*) Comparative illustration of three optimization approaches: the pure evolutionary method ("EM"), the evolutionary method combined with a deep-learning model ("EM+DL"), and the present physics-supervised deep learning–based optimization (PSDLO) method, highlighting their differences in fitness function evaluation and the role of physics. (*B*) Illustration of three case studies: bistable structure design for optimal snap-through behavior, sound barrier performance optimization, and piezoelectric coupling coefficient maximization via strain engineering. The fitness functions for these three cases are the snapping force ratio $|F_2/F_1|$, average transmission loss (TL_{avg}), and out-of-plane piezoelectric coupling coefficient (e_{33}), respectively. (*C*) Comparisons of convergence time and accuracy among the GA, the neural-network (NN)-based GA method (GA + NN), and the present PSDLO algorithms.

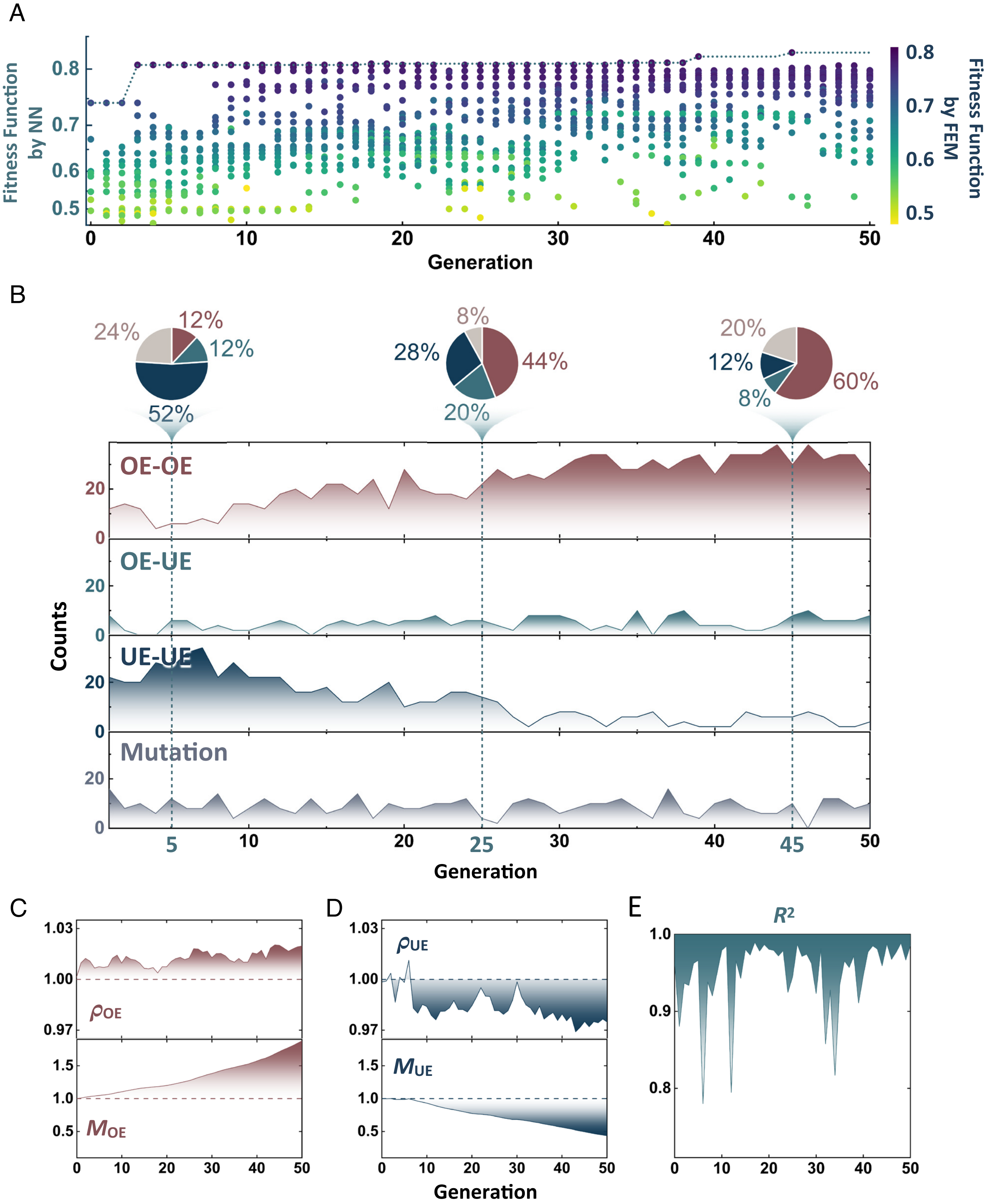


Fig. 2. Detailed analysis of the GA+NN algorithm for 50 generations using the bistable problem. (*A*) Fitness functions predicted by the NN (left vertical axis) and those supervised by physics (right vertical axis color band) for each iteration. (*B*) Genetic inheritance of individuals with overestimated (OE) and underestimated (UE) genes, forming offspring with OE-OE, OE-UE, UE-UE, and mutated genes; pie charts show the proportions of these combinations in the 5th, 25th, and 45th generations. The amplification/diminishing effects of (*C*) OE, and (*D*) UE cases in the population, based on Eqs. **1** and **2**, plotted against the number of generations. (*E*) The R^2 value for the population plotted against the number of generations.

with certain individuals having NN-predicted fitness larger than the fitness from FEM, known as "overestimated" individuals (i.e., OE), and others having NN-predicted fitness smaller than that from FEM, characterized as "underestimated" individuals (i.e., UE). Fig. 2*B* shows how the genes (i.e., features) of these individuals (OE and UE) pass through generations, in which four types can be categorized, namely, OE-OE individuals (whose genes come from both OE parents), UE-UE individuals (whose parents are UEs), OE-UE individuals (mixed OE and UE parents), and mutations. Two apparent trends can be observed in Fig. 2*B*. 1) OE-OE populations increase significantly through generations. Specifically, at the 5th generation, there are only 12% OE-OE individuals, and this number increases to 44% at the 25th generation and dominates with 60% at the 45th generation. 2) UE-UE populations decrease significantly through evolution, from 52% at the 5th generation, to 28% at the 25th generation, and to 12% at the 45th generation. The other two populations (i.e., OE-UE and mutations) fluctuate and remain at a fairly stable level through generations. The obvious increasing trend of the OE genes and decreasing trend of the UE genes are the natural result of GA, namely, the fitness function prefers the genes that produce larger fitness (i.e., OE's) and disfavor the conservative ones (i.e., UE's), which indeed leads to an overall bias and overestimation of the fitness. As shown in Fig. 1*C*, the GA+NN method's prediction overestimates the fitness, e.g., 0.83 (prediction by NN) vs. 0.76 (actual by FEM) for the bistable problem, 57.56 dB (prediction) vs. 47.89 dB (actual) for the acoustic problem, and 17.44 C/m^2 (prediction) vs. 17.06 C/m^2 (actual) for the piezoelectric coupling coefficient problem. This trend is physics-independent.

Since the value of a fitness function of an individual relates linearly to the survival probability of its features to the next generation (i.e., roulette wheel method), OE individuals are amplified, and UE ones are suppressed. To quantify the amplification/diminishing effects of OE and UE species, the following two quantities are defined,

$$\rho_{OE} = \frac{\frac{\sum_{i=1}^{OE} f_{NN}}{\sum_{i=1}^{total} f_{NN}}}{\frac{\sum_{i=1}^{OE} f_p}{\sum_{i=1}^{total} f_p}}, \rho_{UE} = \frac{\frac{\sum_{i=1}^{UE} f_{NN}}{\sum_{i=1}^{total} f_{NN}}}{\frac{\sum_{i=1}^{UE} f_p}{\sum_{i=1}^{total} f_p}}, \quad [1]$$

where "OE" is the counts of OE individuals for a given generation, "total" is the total individuals in one generation, and f_{NN} and f_p are fitness functions predicted by the NN model and FEM, respectively. The denominators of ρ_{OE} and ρ_{UE} are the probabilities of these OE and UE individuals passing to the next generation based on FEM (which is accurate), and the numerators are the inherit probability of passing based on the NN-model. While ρ_{OE} and ρ_{UE} describe the amplification/diminishing effects for each generation, respectively, it is more important to note that the evolutionary algorithm is iterative, with effects compounding over generations. Therefore, we further introduce a cumulative multiplication expression to capture this iterative influence:

$$M_{OE}(n) = \prod_{j=0}^{n} \rho_{OE}(j), \; M_{UE}(n) = \prod_{j=0}^{n} \rho_{UE}(J), \quad [2]$$

Here, *n* and *j* denote the generation numbers.

Fig. 2 *C* and *D* show that the ρ_{OE} increases from 1.00 to 1.02 with iteration, while the ρ_{UE} decreases from 1.00 to 0.97. At first glance, it seems that the values of ρ_{OE} and ρ_{UE} do not deviate significantly from the ideal value 1. In fact, they cannot deviate too much from 1 because of the quality of the NN as characterized by R^2 (=0.99). Yet, it should be noted that this is an evolutionary process, in which small tendencies would cause huge effects after generations. The cumulative effects do show that the multiplication factors, the M_{OE} significantly increases from 1.00 to 1.88, and the M_{UE} significantly drops from 1.00 to 0.44. This demonstrates that the survival rate of OE individuals in the population is gradually and notably increased, while the survival rate of UE individuals is notably decreased.

It should be noted as well that the NN used in this case study performs exceptionally well on the test set, with an R^2 = 0.99 (*SI Appendix*, Fig. S4) and continues performing well through generations (Fig. 2*E*). The eventual deviation from the actual fitness is a natural and inevitable consequence because of the principles of evolutionary algorithms: Individuals showing larger fitness (though maybe incorrect) have a higher survival rate in the next generation, and thus exert the greatest influence on the offspring. Using an analogy from sociology, the entire population tends to "ingratiate" the fitness function and evolve in a direction favored by it, which is the underlying cause of the inaccuracy of GA+NN optimization. Neither GA nor NN is considered at fault, but this simple combination leads to deviation.

Note that only GA+NN was discussed in this section, we also studied PSO+NN for the bistable problem. Similar to the GA+NN method, the PSO+NN method also overestimated the fitness value. For instance, the predicted value was 0.83, while the actual value was 0.76 for the bistable problem, as shown in *SI Appendix*, Figs. S5 and S6. The same conclusion holds.

Deep Learning–Based Evolutionary Method with Physics Supervision – PSDLO. Knowing the intrinsic problem of GA+NN, the core rationale of PSDLO is to establish a balance between accuracy and efficiency in a deep-learning algorithm powered by the evolutionary method. The detailed process for PSDLO is given in Fig. 3. To make this process more focused, we just present GA as a representative evolutionary algorithm with other methods provided in *SI Appendix* and employ NN for the DL method. The first step is generating initial population, which forms a parent population by sorting the NN-calculated fitness function, and the first offspring is the same as that in the GA+NN method. Now, to avoid the intrinsic problem of GA+NN, physics-based supervision is employed here as a critical step to reevaluate the top performers of this population and then re-sort them, i.e., physics-supervised deep-learning optimization. The number of top performers is an empirical, which balances accuracy and efficiency of the algorithm. For more information on selecting the number of supervised individuals and meeting a balance between computational complexity and optimization performance, one can refer to *SI Appendix*, section 3. The same process repeats until it reaches convergence. *SI Appendix*, Fig. S5 presents the process of PSDLO using PSO as the evolutionary method. During the optimization process, PSDLO can be combined with PSO to guide the movement of particles in the search space. Specifically, PSDLO can improve the search capability of PSO by supervising the objective function (physical quantities predicted by the NN). Though the specific steps are different, the essential rationale is the same, i.e., to supervise the fitness function calculated from the NN model.

Now let us examine the performance of PSDLO method by studying the three problems in Fig. 1*B*. Here, only the first top five performers in each generation are supervised by physics-based simulations. Take the piezoelectric problem, for example, to explain the PSDLO process. For this problem, there are five features, and each generation has 200 individuals. At the 5th generation, the top five (i.e., top 2.5% individuals) predicted e_{33} (C/m^2) by the NN model are 17.24, 17.23, 17.20, 17.20, and 17.20 with each subjected to different strain fields characterized by five parameters (e.g., five features), named as NN_1 to NN_5 for the sake of discussion. We then used first-principal calculation to reevaluate these five cases (i.e., NN_1 to NN_5); the corresponding e_{33} (C/m^2) are given by 17.17, 17.09, 17.16, 17.03, 17.01. It can be seen that the values are obviously different from that predicted by the NN model. More importantly,

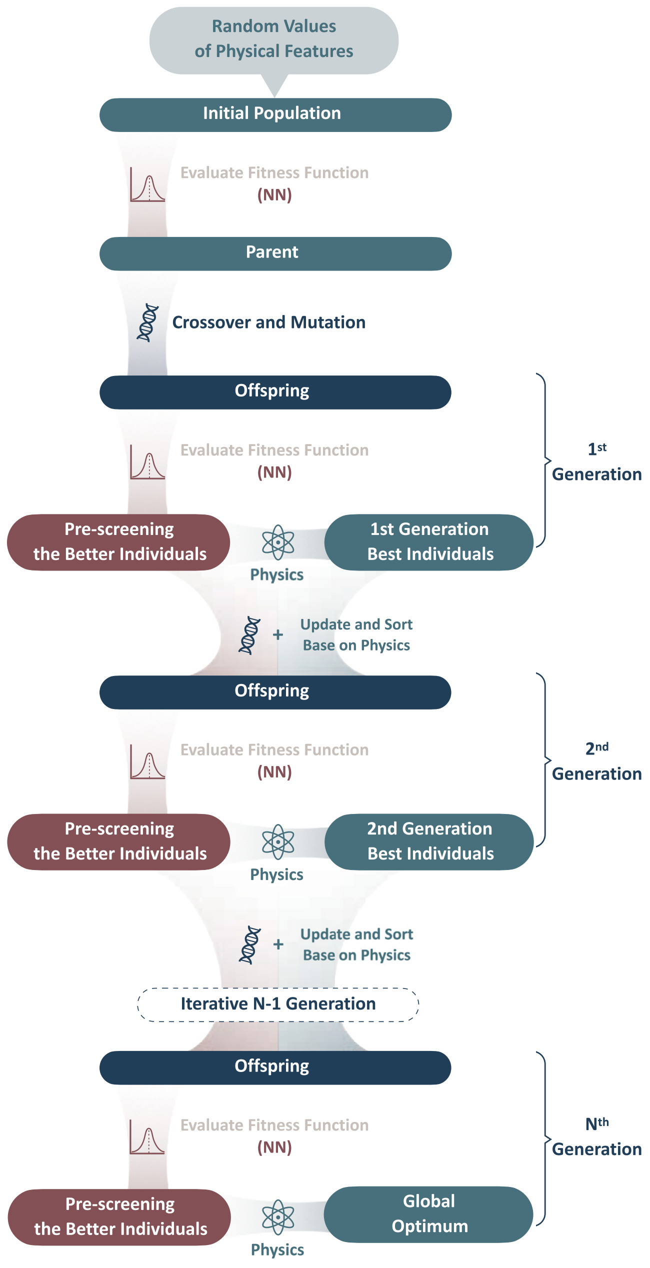

Random Values of Physical Features
Initial Population
Evaluate Fitness Function (NN)
Parent
Crossover and Mutation
Offspring
Evaluate Fitness Function (NN)
Pre-screening the Better Individuals
Physics
1st Generation Best Individuals
1st Generation
Update and Sort Base on Physics
Offspring
Evaluate Fitness Function (NN)
Pre-screening the Better Individuals
Physics
2nd Generation Best Individuals
2nd Generation
Update and Sort Base on Physics
Iterative N-1 Generation
Offspring
Evaluate Fitness Function (NN)
Pre-screening the Better Individuals
Physics
Global Optimum
Nth Generation

Fig. 3. Overall workflow of iterative evolution of PSDLO method. Icons represent different components: the function graph symbolizes the NN, the gene icon represents the GA, and the atom icon denotes physics.

the orders are different. Specifically, NN_3 has a larger e_{33} than NN_2. Now, we re-sort them as 17.17, 17.16, 17.09, 17.03, 17.01 (C/m^2) to make sure the accurate fitness functions are adopted to generate the next generation. This process repeats until reaching a convergence. As shown in Fig. 1*C*, the PSDLO can produce results as accurate as those from the GA method, but is more computationally efficient, e.g., 10 to 40 times faster than the GA method. We also studied the bistable problem using PSDLO using the top four individuals (*SI Appendix*, Fig. S7).

We further used the bistable problem to understand why PSDLO can reach a balance of accuracy and efficiency (Fig. 4). As shown in Fig. 4*A*, the individuals with top fitness exhibit even color changes and no color noise, meaning that after physics-based supervision, the color artifacts (in GA+NN) have been resorted, which ensures that the features having higher probability to be passed to the next generation do have higher fitness. With the physics-based supervision, as one can observe in Fig. 4*B*, the evolutionary processes do not show bias preference, i.e., although OE-OE, UE-UE, and OE-UE offspring fluctuate, they do not show clear lopsided trends. Specifically, the OE-OE populations are 32%, 20%, and 24% at 5th, 25th, and 45th generations, respectively; the UE-UE populations are 52%, 64%, and 36% at 5th, 25th, and 45th generations, respectively; the OE-UE populations are 12%, 12%, and 16% at 5th, 25th, and 45th generations, respectively, which is remarkably different from the case for GA+NN (Fig. 2*B*).

Fig. 4 *C* and *D* demonstrates that ρ_{OE} and ρ_{UE} oscillate around 1.01 and 1.00, respectively, throughout the iterations, without any notable increase or decrease. The cumulative effects of these iterations, represented by M_{OE} and M_{UE}, show a slight increase to 1.31 and a decrease to 0.83, respectively. These results indicate that unlike the GA+NN case, the survival probabilities of the OE and UE populations in the PSDLO method do not undergo significant amplification or reduction. Due to the supervision of physics, the population does not significantly deviate from the correct evolutionary direction and can ensure accurate estimation of the top few individuals with high fitness.

Additionally, we implemented the PSDLO using the PSO approach for the bistable case. The result obtained from the PSDLO algorithm (*SI Appendix*, Fig. S6) is identical to the outcome derived from the purely physics based PSO algorithm. It further corroborates our conclusion.

PSDLO under Inadequate Data. In previous cases, the optimal solution is within the dataset that trains the NN model, which may not be the case for practical physical problems. Thus, the accuracy of the NN model would be significantly hampered outside the training dataset, and in turn the GA+NN method would perform even worse. For PSDLO, can physics supervision be able to correct inaccurate NN model? In this comparison, the results obtained from GA using the fitness evaluated by solving physics-based governing equations are still the benchmark.

Table 1 compares the performance of GA, GA+NN, and PSDLO algorithms studying three problems (i.e., bistable problem, sound barrier, and piezoelectric coefficient) with inadequate dataset to train the NN model. As expected, the deviation from the GA and GA+NN is more severe as compared to Fig. 1*C* where the dataset contains the optimal results. Specifically, for the acoustic problem having eight features, GA+NN predicted TL_{avg} as 46.97 dB as the optimal results, which has 15.7% error compared with 55.70 dB from GA; even more severe, the optimal configuration predicted by GA+NN actually can only provide 38.52 dB average transmission loss, which has 30.8% deviation from the optimal configuration. As a comparison, PSDLO algorithm still maintains minimal error (less than 1.6% even for the acoustic problem with 8 features), while improving computation speed by 10 to 40 times. This case clearly shows that the supervisory role of physics in PSDLO ensures that the entire population evolves in the correct direction even when NN learning is insufficient.

We also studied the dependence of the predication capability of PSDLO on the size of training data. Using the solid mechanics case as an example, we have tested the predictive performance of PSDLO with training data ranging from 100% to 10% of the raw data, as well as its comparison with GA+NN (*SI Appendix*, Tables S4 and S5). We found that as the training data decreased, the predictive performance of NN gradually worsened, with errors eventually reaching as high as 10.09%. However, the predictive performance of PSDLO remained consistently good, even for a training set comprising only 10% of the raw data, maintaining a relatively small error of 1.52% (*SI Appendix*, Fig. S10).

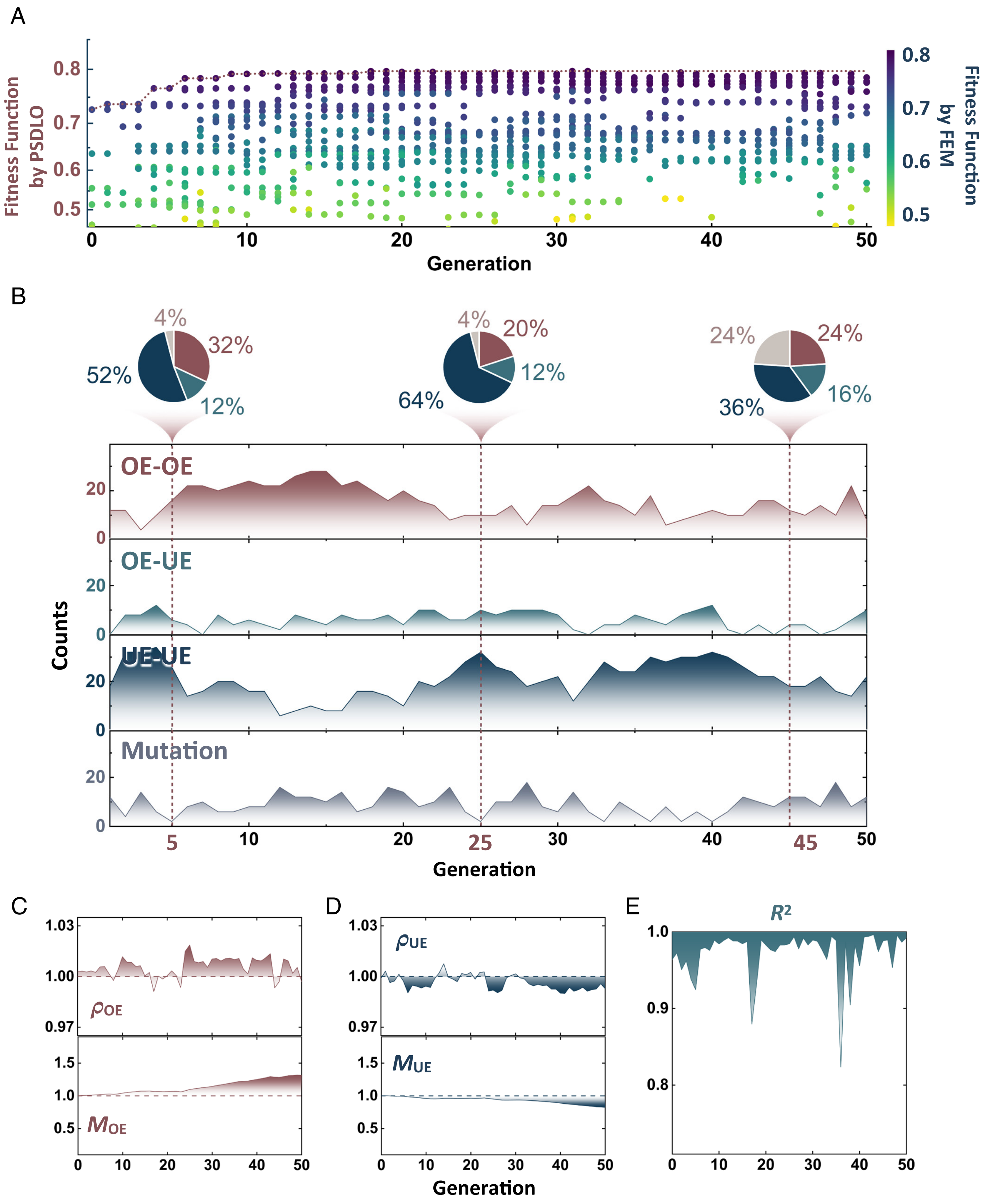
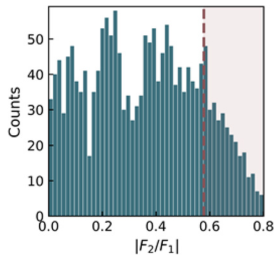
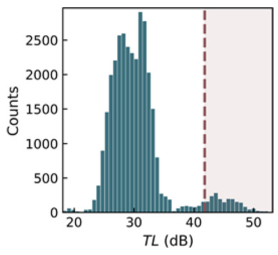
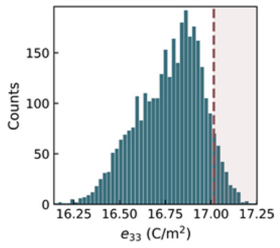
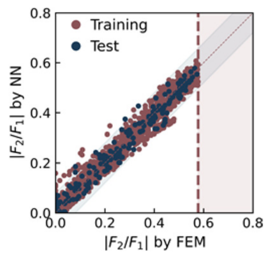
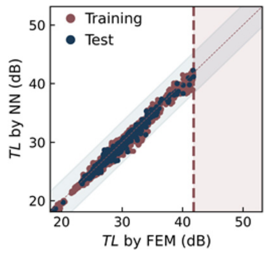
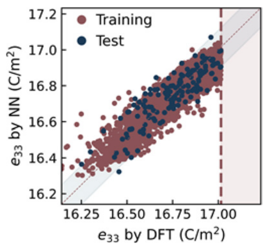
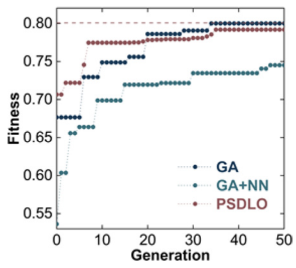
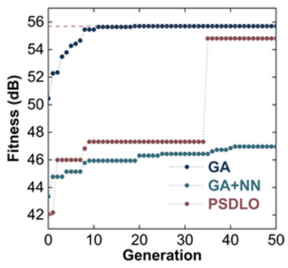
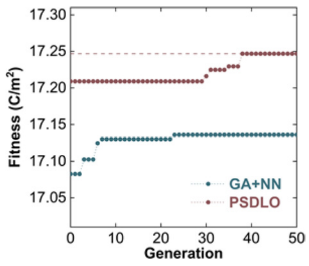


Fig. 4. Detailed analysis of the PSDLO method for 50 generations using the bistable problem. (*A*) Fitness functions predicted by the NN (left vertical axis) and those supervised by physics (right vertical axis color band) for each iteration. (*B*) Genetic inheritance of individuals with overestimated (OE) and underestimated (UE) genes, forming offspring with OE-OE, OE-UE, UE-UE, and mutated genes; pie charts show the proportions of these combinations in the 5th, 25th, and 45th generations. The amplification/diminishing effects of (*C*) OE, and (*D*) UE cases in the population, based on Eqs. **1** and **2**, plotted against the number of generations. (*E*) The R^2 value for the population plotted against the number of generations.

Table 1. Comparison among GA, GA+NN, and PSDLO algorithms studying three problems with inadequate data

System	Solid mechanics			Acoustics			Solid-state physics		
Training set									
Cutoff	0.58			41.83 dB			17.02 C/m^2		
Amount	9,680			38,046			3,098		
NN									
R^2 (test)	0.98			0.98			0.73		
Time	11.58 min			24.60 min			13.80 min		
Iteration									
Algorithm	GA	GA+NN	PSDLO	GA	GA+NN	PSDLO	GA	GA+NN	PSDLO
Prediction	0.80	0.75	0.79	55.70 dB	46.97 dB	54.81 dB	Very Long	17.14 C/m^2	17.25 C/m^2
Actual		0.77			38.52 dB			16.92 C/m^2	
Time	3.72 h	52.85 s	21.77 min	2.87 d	3.70 min	2.12 h		43.20 s	51.72 h

Cutoff value represents a threshold used to remove data points greater than this value from the dataset. This threshold is determined by taking the average of the dataset's mean and maximum values. The estimation for time cost is based on one CPU Core (Intel Core i9-12900 KS) and one GPU (NVIDIA GeForce RTX 3080 Ti).

Discussion

We successfully developed and validated a PSDLO algorithm that achieves a balance between efficiency and accuracy in the evolutionary optimization process. We concluded that despite the high accuracy of deep-learning NN in single-problem predictions, their accuracy would be seriously compromised when applied in evolutionary optimization algorithms due to the intrinsic and inevitable nature of the evolutionary method, i.e., overestimated individuals seize more favorable positions through generations and eventually incorrectly dominate the evolutionary process. To address this issue, we designed a physics-supervised approach, where physics supervises the evolutionary process at each iteration, thereby achieving a balance between accuracy and efficiency. Three cases of different physics using sufficient or insufficient dataset were demonstrated. A concrete conclusion was achieved, i.e., even a very well-trained deep-learning model cannot replace physics-based simulations and proper physics-based supervision is indispensable to reach accurate results.

The present PSDLO method can be extended to optimization problems involving multiscale features, such as the macroscopic instability of beams with microscopic defects, by replacing the current NN methods with more suitable models, such as the PINN model (29) that is capable of predicting macroscopic material properties from atomistic features. Furthermore, the PSDLO method can also be extended to multi-objective optimization problems. One can simply conduct individual prediction by straightforwardly using the current PSDLO methods for each objective. Alternatively, simultaneous predictions can be achieved by combining multi-task NNs with the output layer predicting multiple physical objectives and multi-objective optimization methods, such as NSGA-II (30), SPEA2 (31), MOEA/D (32), or NGA (33).

It is noted that the problems considered in this study involve at most eight features. For more complex physical problems with additional features, the evolutionary process would require a larger population, and the PSDLO algorithm would demonstrate even more significant advantages than those problems. We believe that PSDLO algorithm provides a perspective for addressing multi-scale and multiphysics field issues and will have significant impact on the study of optimization problems in the fields of science and engineering.

Materials and Methods

FEM. We employed the finite element analysis software COMSOL Multiphysics 5.6 for both the solid mechanics problem (i.e., bistable problem) and the acoustic problem. By incorporating the solid mechanics module and the steady-state solver, we successfully established a simulation model and calculated the force-displacement curve. During the modeling process, we discretized the structural domain using triangular elements, ultimately generating a model consisting of 222 mesh elements. To evaluate the performance characteristics of the bistable structure, we extracted the snap-through force as a key indicator from the force-displacement curve. This approach provided an effective means to analyze the bistable structure, contributing to the understanding of its performance behavior.

By utilizing the pressure acoustics module, we calculated the transmission loss (34). In the model, we applied Floquet periodic boundary conditions in the direction perpendicular to the incident sound wave. The structural domain was discretized using triangular mesh elements, with a total mesh count of 27,092. To calculate the transmission loss of the finite element lattice array, we set a background pressure field of 1 Pa outside the acoustic metamaterials. Additionally, we created a perfectly matched layer to simulate an infinitely open region (*SI Appendix*, Fig. S8).

DFT. In this study, we employed the first-principles calculation software VASP (Vienna Ab initio Simulation Package) to compute the piezoelectric coefficients. VASP is a high-performance first-principles calculation software based on DFT capable of accurately simulating the electronic structure and crystal structure of materials (35). Initially, we used the standard library in VASP to perform structural optimization, obtaining the optimal lattice parameters for $PbZr_{0.5}Ti_{0.5}O_3$ (*SI Appendix*, Fig. S9). Subsequently, within the optimized lattice, we simulated the strain field and computed the corresponding piezoelectric coefficients. To ensure the accuracy and reliability of the calculation results, the exchange correlation functional adopted the meta generalized gradient approximation (meta-GGA) of the strongly constrained and appropriately normed (SCAN) semilocal density functional (36). The calculations were converged with an energy tolerance of 10^{-5} eV, and a force tolerance of 0.01 eV/Å under a cutoff energy of 500 eV. All calculations were carried out with automatic *k*-mesh generators with $l = 0.03$, where l is the *k*-points resolved value between adjacent *k*-points in reciprocal cell, and the unit is 2π/Å. The number N of *k*-points is further determined from

$$N = max\left(1, \frac{\left|\vec{b}\right|}{l}\right), \qquad [3]$$

where $\left|\vec{b}\right|$ is the reciprocal lattice vector in the specific direction.

NN. We designed our NN architecture based on the number of features in the physical problems. The NN consists of multiple hidden layers, which serves to capture the complex relationships and interactions among the input features. Each hidden layer comprises a varying number of neurons, depending on the problem's complexity and desired level of abstraction. For problems with four or fewer features, we chose a network with five hidden layers, with the number of neurons in each layer increasing in powers of 2, ranging from 2^4 to 2^8. For physical problems with more than four features, we utilized a network with six hidden layers, with the number of neurons in each layer also increasing in powers of 2, ranging from 2^4 to 2^9.

To introduce nonlinearity, we chose the ReLU (Rectified Linear Unit) activation function for the hidden layers. For the output layer, since the results of the physical problems in this study are nonnegative, we employed the ReLU activation function to obtain the predicted outcomes. We used the TensorFlow (v2.11.0) framework to implement our NN model. To evaluate the discrepancy between the model's predictions and the actual values, we used the mean squared error (MSE) as the loss function. In terms of the optimizer, we selected the Adam optimizer with adaptive learning rate adjustment capabilities. We divided the collected dataset into training and test sets, with the test set comprising 10% of the total dataset. We preprocessed the data, including standardization and normalization, to improve the model's training performance and generalization capabilities. During the training process, we employed batch gradient descent to update the weights and biases at each iteration. We trained our model for 10,000 epochs to ensure adequate learning and convergence of the network weights. To prevent overfitting, we implemented early stopping, where training would terminate prematurely if the validation loss did not show significant improvement over consecutive iterations. After training, we evaluated the model using the test set. The primary evaluation metric was the coefficient of determination (R^2), expressed as:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum\left(\hat{y}_i - y_i\right)}{\sum\left(y_i - \bar{y}\right)}, \qquad [4]$$

where SS_{res} is the sum of squared residuals (the difference between the actual and predicted values $\hat{y}_i$, and SS_{tot} is the total sum of squares (the difference between the actual values and the mean of the actual values $\bar{y}$.

GA. The GA is an optimization method based on the principles of natural selection and genetics (27). As shown in *SI Appendix*, Fig. S2, we first initialized a population according to the characteristics of the physical problem, consisting of Y individuals. Each individual possesses X features. The more features the physical problem has, the larger the population. For physical models with fewer than five features, we set the population size to 50 individuals; for physical models with more than five features, we set the population size to 200 individuals. Next, we evaluate the performance of each individual based on their fitness function. Based on fitness values, we sort and perform roulette wheel selection operations on the population, choosing individuals with higher fitness to proceed to the next generation. After the selection process, we apply cross-over and mutation operations to generate new individuals. The cross-over operation generates new offspring individuals by combining parts of the features of two parent individuals. The mutation operation introduces new variations by randomly changing some features of an individual. To avoid premature convergence and being trapped in local optima, we set the cross-over and mutation probabilities to 90% and 20%, respectively.

Particle Swamp Optimization (PSO). We also investigate the PSDLO algorithm based on PSO to explore the universality of the PSDLO algorithm in different evolutionary algorithms. The PSO algorithm is inspired by the regularity of bird flocking behavior and is a simplified model established by utilizing swarm intelligence. The PSO algorithm leverages the sharing of information among individuals within the swarm, enabling the entire swarm's motion in the problem-solving space to undergo an evolutionary process from disorder to order, thus obtaining the optimal solution (8). From time t to $t + 1$, the change in the position of each particle can be described as follows:

$$X_{ij}(t+1) = X_{ij}(t) + v_{ij}(t+1), \qquad [5]$$

Here, X represents the position, v represents the velocity, the subscript i denotes the ith particle within the swarm, and the subscript j represents the jth feature of the current particle, as illustrated in *SI Appendix*, Fig. S5. At time $t + 1$, the velocity of each particle consists of three components: the inertial direction of the particle's velocity at time t, the direction of the current particle's historical best position (Xp), and the direction of the historical best position (Xg) of all particles. The particle velocity at time $t + 1$ can be expressed as:

$$\begin{aligned} v_{ij}(t+1) = {} & c_0 v_{ij}(t) + c_1 r_1(t)\left[Xp_{ij}(t) - X_{ij}(t)\right] \\ & + c_2 r_2(t)\left[Xg_{ij}(t) - X_{ij}(t)\right], \end{aligned} \qquad [6]$$

Here, $r_1(t)$ and $r_1(t)$ represent random numbers within the range of 0 to 1, which vary with time t. c_0, c_1, and c_2 denote the learning factors. Based on empirical studies, the values for c_0, c_1, and c_2 are set to 0.8, 2, and 2, respectively.

Data, Materials, and Software Availability. All the analysis and result data are included in the article and/or *SI Appendix*. The code for the PSDLO algorithm based on GA and PSO algorithms, as well as the training sets, have been deposited in the GitHub repository (https://github.com/SkyRiverMoon/PSDLO.git) (37).

ACKNOWLEDGMENTS. We thank the Research Center for Industries of the Future at Westlake University and Westlake Education Foundation for supporting this work. L.C. acknowledges support from the China Postdoctoral Science Foundation (Grants 2022M722837).

Author affiliations: [a]School of Engineering, Westlake University, Hangzhou, Zhejiang 310030, China; [b]Westlake Institute for Advanced Study, Hangzhou, Zhejiang 310024, China; [c]Department of Physics, Shaoxing University, Shaoxing 312000, China; [d]Department of Physics, Zhejiang Normal University, Jinhua 321000, China; and [e]Research Center for Industries of the Future, Westlake University, Hangzhou, Zhejiang 310030, China

1. J. A. Ruffolo, L.-S. Chu, S. P. Mahajan, J. J. Gray, Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nat. Commun.* **14**, 2389 (2023).
2. R. Dey *et al.*, Efficient and accurate frailty model approach for genome-wide survival association analysis in large-scale biobanks. *Nat. Commun.* **13**, 5437 (2022).
3. S. Feng *et al.*, Dense reinforcement learning for safety validation of autonomous vehicles. *Nature* **615**, 620–627 (2023).
4. B. Mikulak-Klucznik *et al.*, Computational planning of the synthesis of complex natural products. *Nature* **588**, 83–88 (2020).
5. M. Hutson, AI shortcuts speed up simulations by billions of times. *Science* **367**, 728–728 (2020).
6. P. Gainza *et al.*, De novo design of protein interactions with learned surface fingerprints. *Nature* **617**, 176–184 (2023). 10.1038/s41586-023-05993-x.
7. S. Mirjalili, "Genetic algorithm" in *Evolutionary Algorithms and Neural Networks: Theory and Applications*, S. Mirjalili, Ed. (Springer International Publishing, Cham, 2019), pp. 43–55. 10.1007/978-3-319-93025-1_4.
8. J. Kennedy, R. Eberhart, "Particle swarm optimization" in *Proceedings of ICNN'95 - International Conference on Neural Networks* (IEEE Conference, Perth, WA, Australia, 1995), **vol. 1944**, pp. 1942–1948.
9. M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2006).
10. L. Wu *et al.*, Modular design for acoustic metamaterials: Low-frequency noise attenuation. *Adv. Funct. Mater.* **32**, 2105712 (2021).
11. L. Wu *et al.*, A machine learning-based method to design modular metamaterials. *Extreme Mech. Lett.* **36**, 100657 (2020).
12. S. Lysgaard, J. S. G. Mýrdal, H. A. Hansen, T. Vegge, A DFT-based genetic algorithm search for AuCu nanoalloy electrocatalysts for CO2 reduction. *Phys. Chem. Chem. Phys.* **17**, 28270–28276 (2015).
13. M. L. Paleico, J. Behler, Global optimization of copper clusters at the ZnO(10$\bar{1}$0) surface using a DFT-based neural network potential and genetic algorithms. *J. Chem. Phys.* **153**, 054704 (2020).
14. P. C. Jennings, S. Lysgaard, J. S. Hummelshøj, T. Vegge, T. Bligaard, Genetic algorithms for computational materials discovery accelerated by machine learning. *NPJ Comput. Mater.* **5**, 46 (2019).
15. S. Han *et al.*, Unfolding the structural stability of nanoalloys via symmetry-constrained genetic algorithm and neural network potential. *NPJ Comput. Mater.* **8**, 121 (2022).
16. Y. Wang, Y.-Q. Su, E. J. M. Hensen, D. G. Vlachos, Finite-temperature structures of supported subnanometer catalysts inferred via statistical learning and genetic algorithm-based optimization. *ACS Nano* **14**, 13995–14007 (2020).
17. Z. Yang, C.-H. Yu, M. J. Buehler, Deep learning model to predict complex stress and strain fields in hierarchical composites. *Sci. Adv.* **7**, eabd7416 (2021).
18. E. Khare *et al.*, Discovering design principles of collagen molecular stability using a genetic algorithm, deep learning, and experimental validation. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2209524119 (2022).
19. B. Deng *et al.*, Inverse design of mechanical metamaterials with target nonlinear response via a neural accelerated evolution strategy. *Adv. Mater.* **34**, 2206238 (2022).
20. C. Kim, R. Batra, L. Chen, H. Tran, R. Ramprasad, Polymer design using genetic algorithm and machine learning. *Comput. Mater Sci.* **186**, 110067 (2021).
21. B. D. Lee *et al.*, Discovery of lead-free hybrid organic/inorganic perovskites using metaheuristic-driven dft calculations. *Chem. Mater.* **33**, 782–798 (2021).
22. J.-H. Bastek, S. Kumar, B. Telgen, R. N. Glaesener, D. M. Kochmann, Inverting the structure–property map of truss metamaterials by deep learning. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2111505119 (2022).
23. G. E. Karniadakis *et al.*, Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
24. M. J. Colbrook, V. Antun, A. C. Hansen, The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and Smale's 18th problem. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2107151119 (2022).
25. Y. Xin, Evolving artificial neural networks. *Proc. IEEE Inst. Electr. Electron. Eng.* **87**, 1423–1447 (1999).
26. R. Batra, L. Song, R. Ramprasad, Emerging materials intelligence ecosystems propelled by machine learning. *Nat. Rev. Mater.* **6**, 655–678 (2021).
27. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (University of Michigan Press, 1975).
28. A. E. Eiben, J. E. Smith, "Evolutionary programming" in *Introduction to Evolutionary Computing* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003), pp. 89–99.
29. G. P. P. Pun, R. Batra, R. Ramprasad, Y. Mishin, Physically informed artificial neural networks for atomistic modeling of materials. *Nat. Commun.* **10**, 2339 (2019).
30. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T. Evolut. Comput.* **6**, 182–197 (2002).
31. M. Kim, T. Hiroyasu, M. Miki, S. Watanabe, "SPEA2+: Improving the performance of the strength pareto evolutionary algorithm 2" in *Parallel Problem Solving from Nature - PPSN VIII*, X. Yao *et al.*, Eds. (Springer, Berlin Heidelberg, Berlin, Heidelberg, 2004), pp. 742–751.
32. Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE T. Evolut. Comput.* **11**, 712–731 (2007).
33. M. Sefrioui, J. Perlaux, "Nash genetic algorithms: examples and applications" in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)* (IEEE Conference, La Jolla, CA, USA, 2000), **vol. 501**, pp. 509–516.
34. A.-L. Chen, Y.-S. Wang, Y.-F. Wang, H.-T. Zhou, S.-M. Yuan, Design of acoustic/elastic phase gradient metasurfaces: Principles, functional elements, tunability, and coding. *Appl. Mech. Rev.* **74**, 020801 (2022).
35. G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169–11186 (1996).
36. J. Sun, A. Ruzsinszky, J. P. Perdew, Strongly constrained and appropriately normed semilocal density functional. *Phys. Rev. Lett.* **115**, 036402 (2015).
37. X. Li, Physics-supervised Deep Learning–based Optimization (PSDLO) with Accuracy and Efficiency. GitHub. https://github.com/SkyRiverMoon/PSDLO.git. Deposited 10 August 2023.