

# Octopus Inspired Optimization (OIO): A Hierarchical Framework for Navigating Protein Fitness Landscapes

Xu Wang<sup>a,d,\*</sup>, Yiquan Wang<sup>b,\*</sup>, Tin-Yeh Huang<sup>c</sup>, Yuhua Dong<sup>a</sup>, Jia Deng<sup>a</sup>, Longji Xu<sup>a</sup>, Xiang Li<sup>a</sup>, and Rui He<sup>a</sup>

<sup>a</sup>College of Communication Engineering, Jilin University, Changchun, China

<sup>b</sup>College of Mathematics and System Science, Xinjiang University, Urumqi, China

<sup>c</sup>Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China

<sup>d</sup>Tsinghua University-Peking University Joint Center for Life Sciences, Tsinghua University, Beijing, China

Email: wangxu2020@mails.jlu.edu.cn, ethan@stu.xju.edu.cn

\*These authors contributed equally to this work and are co-corresponding authors.

**Abstract**—Navigating vast, rugged biological fitness landscapes to discover high-value functional patterns—such as optimal protein sequences—is a central challenge in health informatics. However, conventional algorithms often struggle with the exploration-exploitation dilemma, failing to synergize global search with deep local refinement, which leads to entrapment in suboptimal solutions. To overcome this barrier, we introduce Octopus Inspired Optimization (OIO), a novel hierarchical metaheuristic that mimics the octopus’s unique neural architecture to intrinsically unify centralized global exploration and parallelized local exploitation. We validated OIO on a real-world protein engineering benchmark, where it surpassed 15 competing metaheuristics. This success is underpinned by OIO’s architectural suitability for protein-like landscapes, confirmed by its top ranking on the NK-Landscape benchmark, and its powerful optimization engine, demonstrated by its first-place performance on the gold-standard CEC2022 benchmark. OIO thus provides a robust, nature-inspired computational tool for complex optimization problems in drug discovery and personalized medicine.

**Index Terms**—Octopus Inspired Optimization (OIO), Pattern Mining, Metaheuristic, Protein Engineering, Hierarchical Optimization

## I. INTRODUCTION

In the multi-omics era, the proliferation of technologies such as genomics and proteomics has revolutionized health informatics but also generated massive and complex high-dimensional data. A central challenge is to effectively extract meaningful biological insights from this data deluge, which is fundamentally a pattern mining (PM) problem. Recent approaches have demonstrated the efficacy of sequential pattern mining in this domain, successfully applying it to the classification of macromolecule genome sequences [1], the identification of frequent patterns in heat shock proteins [2], and the analysis of viral spike protein structures [3]. Broadly, these challenges range from analytical tasks, like discovering patterns of genetic variation that lead to disease, to design tasks, such as engineering novel protein molecules with specific therapeutic functions. The essence of these tasks is a search and optimization problem, for which heuristic algorithms (HAs) are

critical tools, set within a vast and rugged ‘fitness landscape’ [4], [5]—a high-dimensional space where each point represents a biological configuration and its altitude corresponds to its functional value. Protein engineering serves as a classic and critical example of this challenge in proteomics, a key domain within multi-omics, where the goal is to find an optimal amino acid sequence from a combinatorially astronomical search space [6]–[8]. Navigating this landscape to find the global optimum is exceptionally difficult, and existing HAs often struggle with the “exploration-exploitation” dilemma [9], [10]. Local search methods, like a mountaineer who can only see their immediate surroundings, are efficient at climbing the nearest hill (exploitation) but are easily trapped. Conversely, global search methods, such as classic HAs like Genetic Algorithms, act like reconnaissance aircraft, identifying many potential peaks (exploration) but lacking the tools to efficiently ascend each one [11]–[14]. What this demanding PM task truly requires is a system that synergistically combines reconnaissance and mountaineering—a strategy that can both globally survey the landscape for promising regions and locally mine those regions for the highest peaks in parallel.

In recent years, deep learning methods, such as Protein Language Models (PLMs), have revolutionized protein engineering by serving as powerful surrogate fitness models or efficient sequence generators [15]–[18]. However, they are not optimization algorithms themselves; they leave a critical gap, as the fundamental challenge remains: how to efficiently guide the sequence search to find optimal solutions in the vast sequence space [18]–[20]. This is precisely where a more powerful search strategy is needed. Nature, through millennia of evolution, provides a perfect blueprint for such a strategy in the octopus. An octopus’s brain formulates a global foraging plan, while its eight semi-autonomous arms act as independent, coordinated teams to explore, sense, and capture prey in parallel [21]–[23]. This remarkable fusion of *centralized global planning* and *decentralized parallel execution* provides an ideal model for overcoming the exploration-exploitation dilemma inherent in protein fitness landscapes.

Inspired by this biological blueprint, we introduce the Octopus Inspired Optimization (OIO), a novel metaheuristic algorithm that mimics this unique neural architecture. Our core contributions are twofold: 1) We propose a new algorithmic paradigm whose structure is naturally suited for solving complex combinatorial problems like protein sequence optimization. 2) We comprehensively validate OIO’s performance through a three-tiered experimental framework: demonstrating its real-world efficacy on a protein design benchmark, confirming its structural fit with a classical landscape model, and verifying its robust underlying optimization engine on a standard benchmark suite. This work positions OIO as a powerful new tool not just for protein engineering, but for a broader range of optimization challenges in computational **drug discovery** and **personalized medicine**. Its advantages are demonstrated by systematically outperforming a comprehensive and diverse suite of algorithms, including both time-tested classic strategies and state-of-the-art metaheuristics from the last two years.

## II. RELATED WORKS

### A. Computational Approaches for Protein Sequence Optimization

The computational search for optimal protein sequences is fundamentally constrained by the exploration-exploitation dilemma. Global strategies like conventional Evolutionary Algorithms (EAs) [24]–[28] excel at surveying the vast fitness landscape but are often slow to converge, while Local Search and Heuristics like hill climbing rapidly ascend the nearest fitness peak but are easily trapped in suboptimal solutions [29]–[31]. Each of these classes represents a point on the spectrum, forcing a trade-off. While hybrid approaches (e.g., Hybrid PSO-GWO [32]), often known as memetic algorithms, have sought to combine these strengths, they typically bolt on a local searcher to a global algorithm [33]–[37]. A framework that intrinsically and structurally unifies high-level strategic exploration with deep, parallelized local exploitation remains a critical unmet need [8], [38]. OIO is proposed not merely to integrate these mechanisms, but to embody this synergy within its core hierarchical architecture.

### B. The Structural Limitations of General-Purpose Metaheuristics

To rigorously benchmark OIO, we compare it against a comprehensive suite of 15 metaheuristics. This suite was intentionally designed to be diverse, comprising 7 well-established, classic algorithms [29], [39]–[44] and 8 state-of-the-art optimizers [32], [45]–[51] proposed within the last two years. While this diverse set spans foundational paradigms from swarm intelligence (e.g., Particle Swarm Optimization, PSO [40]) to evolutionary methods (e.g., Differential Evolution, DE [42]), they largely share a critical structural limitation: a “flat” population structure. In this paradigm, all search agents operate at the same conceptual level, contributing to a shared global search [52]–[54]. They lack a native, hierarchical framework that explicitly decouples high-level strategic direction (like an octopus’s brain) from parallel, semi-autonomous local

refinement (like its tentacles). This structural flaw often forces the entire population to converge prematurely or maintain diversity at the cost of deep exploitation, highlighting the critical gap for an algorithm whose very architecture is designed for the multi-level search required by complex protein fitness landscapes [55]–[58].

## III. OIO: A HIERARCHICAL FRAMEWORK FOR PROTEIN OPTIMIZATION

### A. Biological Principles and the Hierarchical Synergy Framework

The design of OIO is a direct abstraction of the octopus’s neurological and morphological features. We map this biological architecture to a hierarchical algorithmic framework, as shown in Fig. 1. This framework operates on three synergistic levels, each with a distinct role in navigating the protein fitness landscape.

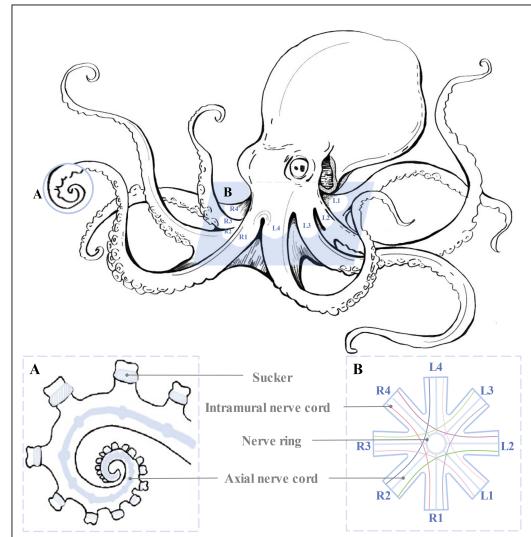


Fig. 1: The octopus nervous system, illustrating the centralized brain and decentralized arm ganglia that inspire OIO’s hierarchical architecture.

This hierarchical structure, inspired by the octopus’s nervous system, is visualized in Fig. 2. The framework’s power lies in its clear mapping of biological roles to algorithmic components and, most critically, in the dynamic, bidirectional flow of information that drives the optimization. This feedback loop—where local data is aggregated upward and global strategy is disseminated downward—creates a synergistic system that intrinsically balances exploration and exploitation.

### B. Core Optimization Mechanisms

Two core mechanisms, illustrated in Fig. 3, drive OIO’s search process and ensure a robust balance between exploration and exploitation.

**Co-evolution and Adaptive Regeneration.** OIO employs a group co-evolution strategy where some tentacles act as ‘masters’ to deeply exploit known high-fitness regions, while

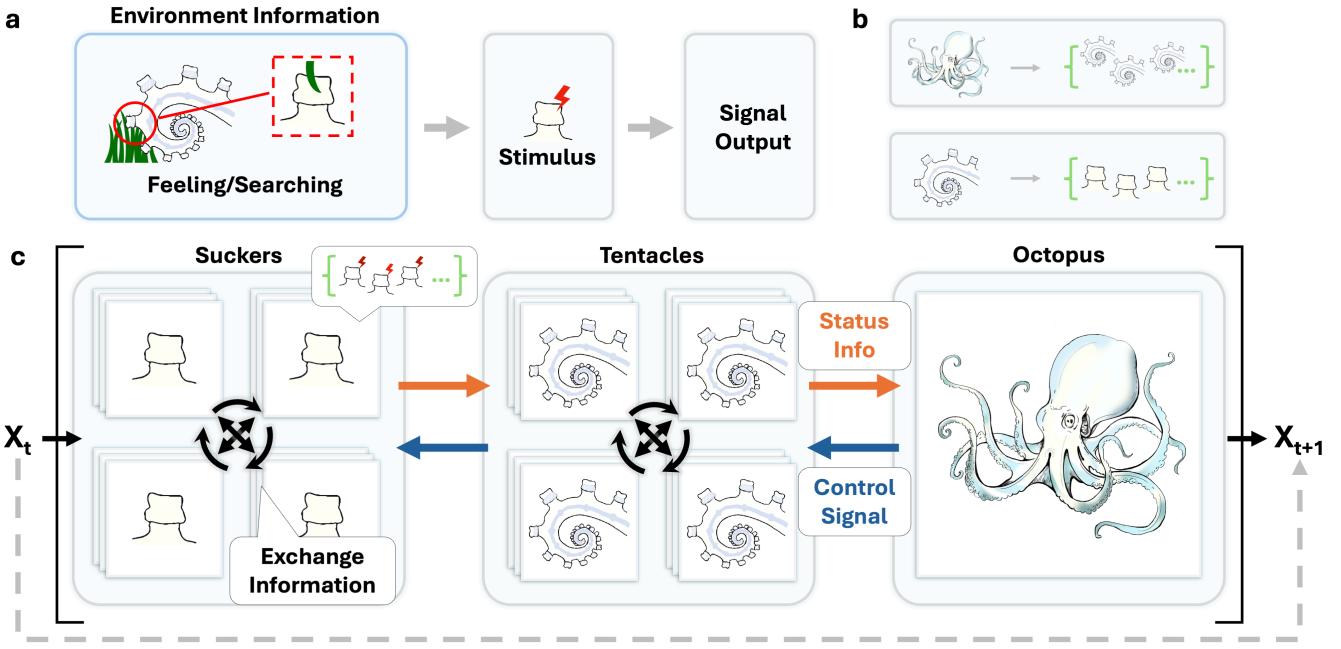


Fig. 2: The simplified framework of OIO, integrating signal generation, hierarchical mapping, and information interaction. Panel (a) illustrates how suckers perceive the environment, generate stimuli, and output signals as the initial source of information. Panel (b) presents the biological hierarchy of the octopus abstracted into the algorithm, where the octopus corresponds to the global individual, tentacles act as regional agents, and suckers serve as local units. Panel (c) demonstrates the bidirectional flow of information across the three levels, where suckers exchange information and report status upward, tentacles aggregate and transmit results to the octopus, and the octopus integrates feedback and issues control signals downward to guide the next iteration.

others act as 'slaves' to explore new areas (Fig. 3(a)). Critically, an underperforming tentacle can be adaptively regenerated to a new random location. This is a key mechanism for escaping the numerous local optima characteristic of protein fitness landscapes.

**The Iterative Optimization Process.** The entire optimization unfolds as a dynamic, iterative process (Fig. 3(b)). In each cycle, a multi-level feedback loop occurs: suckers perform local mutations, tentacles aggregate this information to adjust their regional search strategy, and the individual oversees the global picture, potentially reassigning tentacles. This process allows OIO to dynamically allocate computational resources to the most promising areas of the fitness landscape.

### C. Mathematical Formulation of Core Mechanisms

To formally define the optimization dynamics within the OIO framework, we present the core mathematical equations governing the behavior of the Suckers and Tentacles.

At the heart of OIO's local exploitation is the position update for each Sucker. Let  $\vec{S}_{ij}(t)$  be the position of the  $j$ -th Sucker of the  $i$ -th Tentacle at iteration  $t$ . Its new position  $\vec{S}_{ij}(t+1)$  is determined by a hybrid strategy that adaptively switches between exploration and exploitation. This transition is governed by a dynamic energy factor  $E$ . At iteration  $t$  (up

to a maximum  $T_{\max}$ ),  $E$  is calculated using a random seed  $E_0 \in [-1, 1]$ :

$$E = 2E_0(1 - t/T_{\max}) \quad (1)$$

The primary update rule depends on the magnitude of  $E$ :

$$\vec{S}_{ij}(t+1) = \begin{cases} \text{Exploration Strategy} & \text{if } |E| \geq 1 \\ (\vec{S}_{ij}(t), \vec{S}_{\text{rand}}, \dots) & \\ \text{Exploitation Strategy} & \text{if } |E| < 1 \\ (\vec{S}_{ij}(t), \vec{T}_{i,\text{best}}, \vec{G}_{\text{best}}) & \end{cases} \quad (2)$$

where  $\vec{T}_{i,\text{best}}$  is the best-known position within Tentacle  $T_i$ , and  $\vec{G}_{\text{best}}$  is the global best position found by the entire individual. The exploitation strategy synthesizes information from the Sucker's personal best, the Tentacle's best, and the global best, resembling a multi-level social learning structure inspired by Particle Swarm Optimization (PSO):

$$\begin{aligned} \vec{v}_{ij}(t+1) &= w\vec{v}_{ij}(t) + c_1 r_1(\vec{P}_{ij,\text{best}} - \vec{S}_{ij}(t)) \\ &\quad + c_2 r_2(\vec{T}_{i,\text{best}} - \vec{S}_{ij}(t)) \\ &\quad + c_3 r_3(\vec{G}_{\text{best}} - \vec{S}_{ij}(t)) \end{aligned} \quad (3)$$

$$\vec{S}_{ij}(t+1) = \vec{S}_{ij}(t) + \vec{v}_{ij}(t+1) \quad (4)$$

Here,  $\vec{v}_{ij}$  is the velocity vector,  $w, c_1, c_2, c_3$  are inertial and acceleration coefficients, and  $r_1, r_2, r_3$  are random numbers.

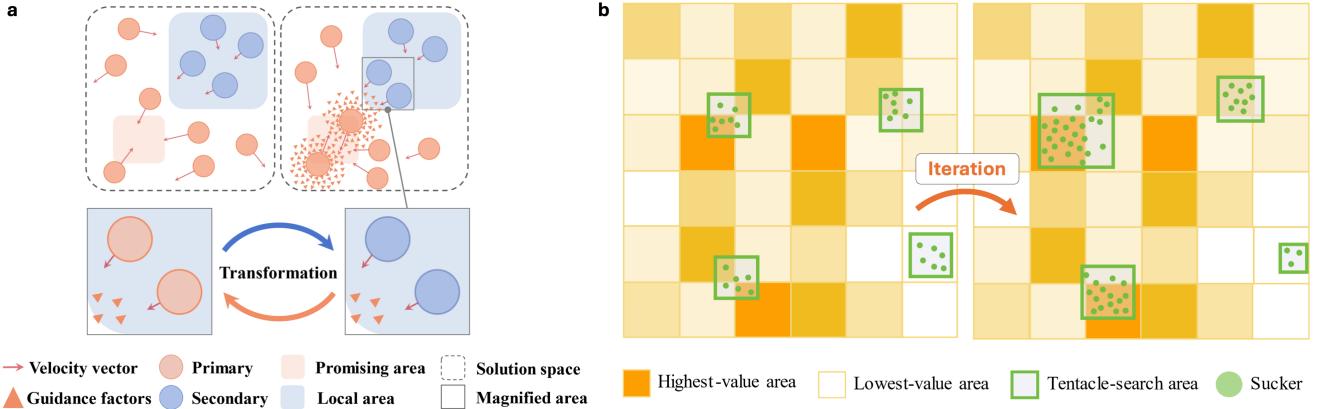


Fig. 3: Core dynamic mechanisms of OIO. Panel (a) illustrates the Group Co-evolution strategy, which balances exploration and exploitation using master and slave tentacles and includes an adaptive regeneration mechanism to escape local optima. Panel (b) shows the Iterative Optimization Process, a dynamic feedback loop where information flows from the suckers up to the individual, enabling adaptive resource allocation.

#### Algorithm 1 The Octopus Inspired Optimization (OIO)

```

1: Input: Fitness function  $f(x)$ , Max Iterations  $T_{\max}$ 
2: Initialize: Individual  $I$ , Tentacles  $\{T_1, \dots, T_m\}$ , Suckers  $\{S_{ij}\}$ 
3: Evaluate all Suckers' initial fitness
4: while  $t < T_{\max}$  do
5:   Calculate Energy Factor  $E$  via Eq. (1)
6:   for all Tentacle  $T_i$  in  $\{T_1, \dots, T_m\}$  do
7:     for all Sucker  $S_{ij}$  in  $T_i$  do
8:       Update position via Eq. (2)-(4) based on  $E$ 
9:       Evaluate fitness and update personal best  $\vec{P}_{ij,\text{best}}$ 
10:    end for
11:    Update Tentacle's best  $\vec{T}_{i,\text{best}}$ 
12:   end for
13:   Update global best solution  $\vec{G}_{\text{best}}$ 
14:   Perform regeneration (Eq. 5) if  $\text{Stag}_i > \tau_{\text{stag}}$ 
15:    $t \leftarrow t + 1$ 
16: end while
17: Return: The best solution  $\vec{G}_{\text{best}}$ 

```

This formulation extends canonical PSO by incorporating the Tentacle's collective intelligence ( $\vec{T}_{i,\text{best}}$ ), creating a more structured and efficient local search.

The adaptive regeneration mechanism is triggered for a Tentacle  $T_i$  when its stagnation counter,  $\text{Stag}_i$ , exceeds a predefined threshold  $\tau_{\text{stag}}$ . Regeneration involves repositioning the Tentacle's search center  $\vec{C}_i$ :

$$\text{if } \text{Stag}_i > \tau_{\text{stag}} \text{ then } \vec{C}_i(t+1) = \vec{L} + \text{rand}() \cdot (\vec{U} - \vec{L}) \quad (5)$$

where  $\vec{L}$  and  $\vec{U}$  define the lower and upper bounds of the search space. The complete procedure is outlined in Algorithm 1.

#### D. Encoding Strategy for Protein Sequence Optimization

A critical component for applying OIO to protein engineering is the strategy for encoding a discrete protein sequence within

the algorithm's continuous internal search space. We adopt a probabilistic matrix encoding scheme. For a protein of length  $L$ , an internal solution in OIO is represented by a one-dimensional vector of length  $L \times 20$ . During fitness evaluation, this vector is reshaped into an  $L \times 20$  matrix  $\mathbf{P}$ , where the element  $P_{ij}$  represents a preference score for the  $j$ -th amino acid (out of 20) appearing at the  $i$ -th position in the sequence.

Subsequently, a Softmax function is applied to each row of this matrix, converting the scores into a valid probability distribution. The final protein sequence is then generated by sampling an amino acid for each position based on these probabilities (or by deterministically selecting the amino acid with the highest probability). This encoding allows OIO to perform smooth optimization in a continuous internal space while effectively exploring the discrete space of protein sequences.

## IV. EXPERIMENTS AND ANALYSIS

### A. Experimental Design

To comprehensively validate the OIO algorithm, we designed a rigorous three-tiered experimental framework intended to dissect its performance from practical application down to its core mechanics. This framework is guided by three progressively deeper questions. First, we present a case study of HA application in proteomics by assessing OIO's real-world efficacy on the demanding DMS-GFP protein optimization task [59]. Second, we investigate its structural affinity for protein-like fitness landscapes using the classic NK-Landscape model to understand the architectural reasons for its success [60]. Finally, we evaluate its core engine performance using the gold-standard CEC2022 benchmark suite to confirm the fundamental power and robustness of its optimization mechanisms [61], [62].

Across all three benchmarks, OIO was compared against a comprehensive and challenging suite of 15 metaheuristic algorithms, comprising both classic methods and state-of-the-art optimizers. To ensure reproducibility, the specific parameter

settings for OIO are detailed in Table I. This rigorous approach ensures that OIO's performance is validated against a diverse and contemporary set of optimization strategies, confirming its novelty and effectiveness.

TABLE I: Hyperparameter settings for the OIO algorithm.

Hyperparameter	Value
Number of Tentacles	5
Suckers per Tentacle	40
Iterations per Tentacle	100
Stagnation Threshold for Regeneration	5 iterations
Diversity Threshold	0.005
Elite Memory Size	8

### B. Core Validation: Real-World GFP Optimization

The primary validation of OIO was conducted on the GFP sequence optimization task, where the goal is to find the amino acid sequence with the highest fluorescence intensity. The results, visualized in Fig. 4, are highly illuminating.

The most striking result is that OIO achieved the second-best overall rank among 16 algorithms, surpassed only by Hill Climbing (HC). Crucially, OIO was the top-performing algorithm among all general-purpose metaheuristics. This outcome is significant and requires careful interpretation. Hill Climbing is a pure local search method—a specialized "exploiter." Its success confirms that the GFP fitness landscape contains steep, climbable peaks. However, HC's performance is entirely dependent on its starting point; from a poor initial sequence, it would be irreversibly trapped in a suboptimal region.

OIO, by contrast, is a general-purpose optimizer designed to balance exploration and exploitation. The fact that it achieved a performance nearly identical to a specialized exploiter, without sacrificing its global search capability, is a powerful testament to the efficiency of its "tentacle-sucker" mechanism for deep, parallel local refinement. This makes OIO far more robust for unknown and potentially more complex fitness landscapes where a good starting point is not guaranteed. As illustrated by the tighter high-fitness distribution in Fig. 4, OIO consistently achieved a higher mean fitness than 14 out of the 15 other algorithms, demonstrating a superior balance of global robustness and local efficiency—a critical requirement for practical protein engineering tasks. Furthermore, the search trajectory indicates OIO rapidly approaches the high-fitness region found by the specialist HC algorithm. This complementary nature suggests a promising future research direction: employing OIO for robust global exploration to identify high-potential basins of attraction, followed by a rapid HC-like local search for final, efficient refinement. This hybrid strategy could potentially achieve the best of both worlds.

### C. Structural Affinity Analysis: The NK-Landscape Benchmark

The NK-Landscape is a tunable mathematical model designed specifically to replicate the "rugged" and correlated nature of protein fitness landscapes. An algorithm that performs well on this benchmark is likely to have a search strategy that

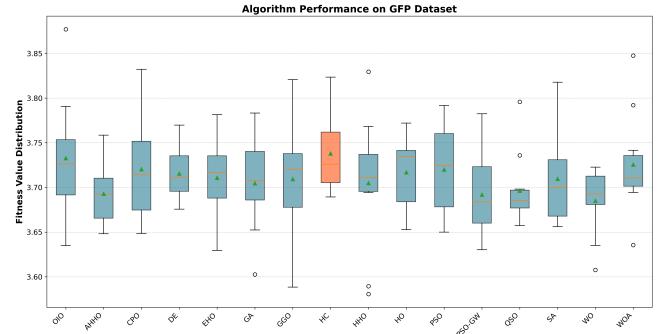


Fig. 4: Performance comparison of OIO and 15 baseline algorithms on the GFP optimization task. The boxplot illustrates the distribution of final fitness values across 10 independent runs. OIO (highlighted) demonstrates a superior median fitness and a consistently high-performing distribution, second only to the specialist Hill Climbing algorithm.

is intrinsically compatible with this type of problem structure. To ensure a rigorous evaluation, we constructed a test suite of five NK configurations with increasing difficulty, from Simple ( $N=20, K=2$ ) to Complex ( $N=100, K=5$ ). For each configuration, every algorithm was run 10 times with a budget of 20,000 function evaluations, and all continuous-space optimizers used a unified Sigmoid transfer function for a fair comparison.

The results, summarized in Fig. 5, are decisive. OIO achieved the number one overall rank among all 16 algorithms. The boxplot in Fig. 5 visually confirms this dominance, showing that OIO's performance distribution is clearly superior to all other competitors, with a consistently higher median and lower variance. Furthermore, quantitative analysis reveals that OIO consistently obtained a higher mean fitness than the entire cohort of baseline algorithms across all five levels of landscape complexity. This commanding performance provides strong evidence that OIO's hierarchical, parallel search architecture is naturally suited to navigating the high-dimensional, multi-modal, and rugged landscapes characteristic of protein optimization, explaining its ability to effectively map and search the complex GFP fitness landscape. This structural advantage translates into superior search efficiency, where OIO not only finds a better final solution but also converges significantly faster than the other top-performing algorithms.

### D. Core Engine Validation: The CEC2022 Benchmark

While OIO operates on discrete protein sequences, its internal search process, guided by the probabilistic encoding matrix, occurs in a continuous space. The CEC2022 benchmark suite is the standard for evaluating the core performance of continuous optimizers, rigorously testing their ability to balance global exploration and local exploitation. For this validation, all 16 algorithms were tested on the 12 standard 10-dimensional problems (F1-F12). Each was run 10 times with a maximum budget of 20,000 function evaluations (FES).

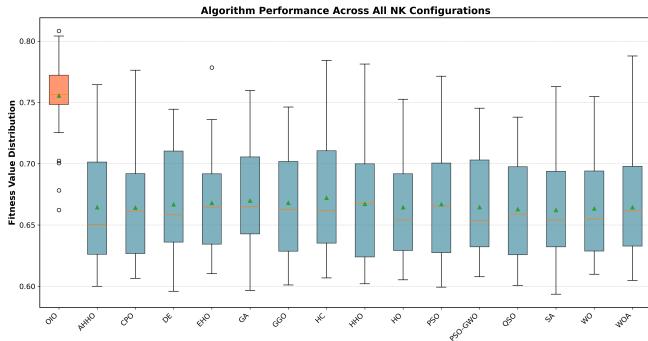


Fig. 5: Comprehensive performance comparison on the NK-Landscape benchmark. The boxplot shows the distribution of final fitness values from 10 independent runs across all five NK configurations. OIO (highlighted) exhibits a markedly superior distribution, with a higher median and less variance compared to the 15 baseline algorithms.

The results unequivocally demonstrate the superiority of OIO’s core engine. Quantitative ranking analysis confirms that OIO achieved the number one overall rank, significantly outperforming the second-place CPO. This commanding lead confirms OIO’s world-class optimization power.

The performance gap is visually evident in Fig. 6. OIO’s results (highlighted) not only exhibit a lower median, indicating better solutions, but also a much tighter interquartile range, signifying exceptional stability. Table II further quantifies this, showing that on multimodal problems like F6, OIO finds solutions orders of magnitude better than the best baseline, showcasing its critical ability to escape local optima. In terms of convergence behavior, OIO exhibits a rapid descent towards superior solutions across unimodal, multimodal, and composition functions, starkly contrasting with the slower convergence or premature stagnation observed in other leading algorithms.

TABLE II: Mean Fitness Comparison on Representative CEC2022 Functions.

Function	OIO Mean Fitness	Best Baseline Mean	Avg. of Baselines
F1 (Unimodal)	$4.10 \times 10^3$	$7.99 \times 10^4$ (CPO)	$1.12 \times 10^5$
F6 (Multimodal)	$6.57 \times 10^4$	$2.31 \times 10^9$ (CPO)	$2.90 \times 10^9$
F12 (Composition)	$2.93 \times 10^3$	$2.91 \times 10^3$ (CPO)	$2.95 \times 10^3$

Beyond solution quality, OIO excels in computational efficiency. As shown in Table III, OIO consistently finds its superior solutions in a significantly shorter timeframe. On every representative function, OIO’s average execution time is substantially lower than even the fastest baseline algorithm. For example, on the complex F6 function, OIO is over 30% faster than the quickest competitor (34.3s vs 48.8s) and nearly twice as fast as the baseline average. This demonstrates that OIO’s performance gains do not come at the cost of computational overhead; rather, its intelligent search strategy leads to a more

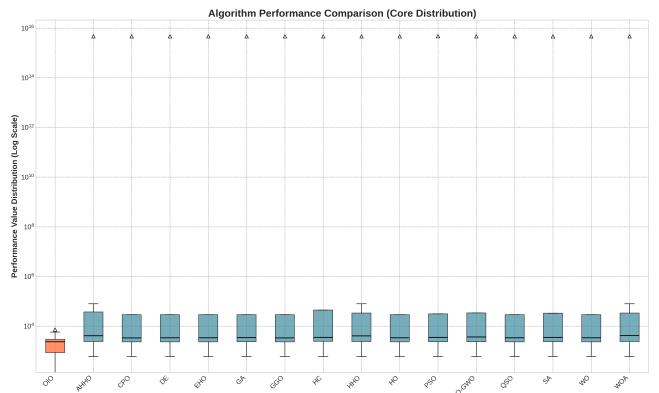


Fig. 6: Performance comparison on the CEC2022 benchmark suite. The boxplot shows the distribution of final fitness values from 10 independent runs across all 12 functions. OIO (highlighted) demonstrates a substantially lower median and a more compact distribution, indicating superior performance and higher stability.

efficient discovery of high-quality solutions. This combination of accuracy, stability, and speed solidifies OIO’s position as a powerful and practical optimization tool.

TABLE III: Comparison of Mean Execution Time (seconds) on Representative CEC2022 Functions.

Function	OIO Avg. Time	Fastest Baseline	Avg. Baseline	Slowest Baseline
F1	<b>22.49s</b>	32.33s	43.60s	61.25s
F6	<b>34.26s</b>	48.82s	63.87s	86.85s
F12	<b>148.77s</b>	207.46s	257.57s	332.03s

## V. DISCUSSION & CONCLUSION

In this work, we addressed the significant challenge of computational protein engineering by proposing the Octopus Optimizer (OIO), a novel metaheuristic algorithm inspired by the sophisticated neural system of the octopus. Through a three-tiered experimental framework, we have demonstrated that: 1) OIO exhibits top-tier practical performance on a real-world GFP protein design task; 2) its success is rooted in a fundamental architectural alignment with the nature of protein fitness landscapes, as validated on the NK-Landscape benchmark; and 3) its performance is powered by a state-of-the-art optimization engine, confirmed by its number one ranking on the CEC2022 benchmark suite.

The core advantage of OIO lies in its intrinsic, structured synergy between exploration and exploitation. This makes it more robust and efficient than traditional methods for complex black-box problems, such as protein optimization, which demand both a global perspective and deep local refinement. Furthermore, its architecture with multiple semi-autonomous tentacles is naturally suited for distributed and parallel HA frameworks, addressing the critical need for scalability when tackling large-scale multi-omics data.

The significance of this work, however, extends beyond outperforming other algorithms; it lies in establishing OIO's strategic role within modern health informatics. We identify two core ecological niches for OIO. First, it can form powerful hybrid approaches combining HA with Machine Learning (ML), a key theme of this workshop. In such a synergy, OIO acts as an intelligent 'Optimization Driver' proposing candidates for a DL-based 'Evaluator' (e.g., a PLM), combining OIO's search power with the domain knowledge of DL models. Second, in many 'black-box' scenarios central to health informatics—such as early-stage drug discovery where wet-lab experiments are costly, or designing therapies for rare diseases where training data is scarce—DL models are not viable [63]–[65]. In these critical situations, an efficient, gradient-free optimizer like OIO is indispensable [66].

This positioning opens up new avenues for computational biology. Future work will focus on four key directions: 1) extending OIO to handle multi-objective protein design [30], [67]–[69]; 2) applying it to *de novo* drug discovery and molecular design [70]–[72]; 3) adapting the OIO framework to other multi-omics optimization problems, such as identifying optimal biomarker combinations from transcriptomics data for clinical decision support, or optimizing microbial compositions in microbiome analysis; and 4) pursuing the deep integration of OIO with PLMs to create practical, high-performance optimization systems. Ultimately, OIO provides a powerful new tool, demonstrating that nature-inspired computational strategies are vital for solving humanity's most complex scientific challenges.

#### ACKNOWLEDGMENT

The source code for this paper is available on GitHub: [https://github.com/JLU-WangXu/Octopus\\_Inspired\\_Optimization\\_OIO](https://github.com/JLU-WangXu/Octopus_Inspired_Optimization_OIO).

#### REFERENCES

- [1] M. S. Nawaz, P. Fournier-Viger, S. Nawaz, H. Zhu, and U. Yun, "Spm4gac: Spm based approach for genome analysis and classification of macromolecules," *International Journal of Biological Macromolecules*, vol. 266, p. 130984, 2024.
- [2] M. S. Nawaz, P. Fournier-Viger, S. Nawaz, W. Gan, and Y. He, "Fsp4hsp: Frequent sequential patterns for the improved classification of heat shock proteins, their families, and sub-types," *International Journal of Biological Macromolecules*, vol. 277, p. 134147, 2024.
- [3] M. S. Nawaz, P. Fournier-Viger, and Y. He, "S-pdb: Analysis and classification of sars-cov-2 spike protein structures," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, December 2022, pp. 2259–2265.
- [4] P. A. Romero and F. H. Arnold, "Exploring protein fitness landscapes by directed evolution," *Nature reviews Molecular cell biology*, vol. 10, no. 12, pp. 866–876, 2009.
- [5] P. A. Romero, A. Krause, and F. H. Arnold, "Navigating the protein fitness landscape with gaussian processes," *Proceedings of the National Academy of Sciences*, vol. 110, no. 3, pp. E193–E201, 2013.
- [6] S. B. Ebrahimi and D. Samanta, "Engineering protein-based therapeutics through structural and chemical design," *Nature communications*, vol. 14, no. 1, p. 2411, 2023.
- [7] G. O. Ndochinwa, Q.-Y. Wang, N. O. Okoro, O. C. Amadi, T. N. Nwagu, C. I. Nnamchi, A. N. Moneke, and A. S. Odiba, "New advances in protein engineering for industrial applications: Key takeaways," *Open life sciences*, vol. 19, no. 1, p. 20220856, 2024.
- [8] K. I. Albanese, S. Barbe, S. Tagami, D. N. Woolfson, and T. Schiex, "Computational protein design," *Nature Reviews Methods Primers*, vol. 5, no. 1, p. 13, 2025.
- [9] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
- [10] O. Berger-Tal, J. Nathan, E. Meron, and D. Saltz, "The exploration-exploitation dilemma: a multidisciplinary framework," *PLoS one*, vol. 9, no. 4, p. e95693, 2014.
- [11] T. T. Hills, P. M. Todd, D. Lazer, A. D. Redish, and I. D. Couzin, "Exploration versus exploitation in space, mind, and society," *Trends in cognitive sciences*, vol. 19, no. 1, pp. 46–54, 2015.
- [12] H. Taylor, B. Fernandes, and S. Wraith, "The evolution of complementary cognition: humans cooperatively adapt and evolve through a system of collective cognitive search," *Cambridge Archaeological Journal*, vol. 32, no. 1, pp. 61–77, 2022.
- [13] T. Dokeroğlu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers & Industrial Engineering*, vol. 137, p. 106040, 2019.
- [14] J. Jerebic, M. Mernik, S.-H. Liu, M. Ravber, M. Baketarić, L. Mernik, and M. Črepinské, "A novel direct measure of exploration and exploitation based on attraction basins," *Expert Systems with Applications*, vol. 167, p. 114353, 2021.
- [15] S. Gelman, B. Johnson, C. Freschlin, A. Sharma, S. D'Costa, J. Peters, A. Gitter, and P. A. Romero, "Biophysics-based protein language models for protein engineering," *bioRxiv*, pp. 2024–03, 2025.
- [16] Y. Qiu and G.-W. Wei, "Artificial intelligence-aided protein engineering: from topological data analysis to deep protein language models," *Briefings in Bioinformatics*, vol. 24, no. 5, p. bbad289, 2023.
- [17] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial, "Proteinbert: a universal deep-learning model of protein sequence and function," *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, 2022.
- [18] Y. Xu, D. Verma, R. P. Sheridan, A. Liaw, J. Ma, N. M. Marshall, J. McIntosh, E. C. Sherer, V. Svetnik, and J. M. Johnston, "Deep dive into machine learning models for protein engineering," *Journal of chemical information and modeling*, vol. 60, no. 6, pp. 2773–2790, 2020.
- [19] E.-M. Nikolados and D. A. Oyarzun, "Deep learning for optimization of protein expression," *Current opinion in biotechnology*, vol. 81, p. 102941, 2023.
- [20] Z. Wu, K. E. Johnston, F. H. Arnold, and K. K. Yang, "Protein sequence design with deep generative models," *Current opinion in chemical biology*, vol. 65, pp. 18–27, 2021.
- [21] L. Zullo, H. Eichenstein, F. Maiole, and B. Hochner, "Motor control pathways in the nervous system of octopus vulgaris arm," *Journal of Comparative Physiology A*, vol. 205, no. 2, pp. 271–279, 2019.
- [22] D. M. Sivitilli, T. Strong, W. Weertman, J. Ullmann, J. R. Smith, and D. H. Gire, "Mechanisms of octopus arm search behavior without visual feedback," *Bioinspiration & Biomimetics*, vol. 18, no. 6, p. 066017, 2023.
- [23] B. Hochner, T. Shomrat, and G. Fiorito, "The octopus: a model for a comparative analysis of the evolution of learning and memory mechanisms," *The biological bulletin*, vol. 210, no. 3, pp. 308–317, 2006.
- [24] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [25] A. N. Sloss and S. Gustafson, "2019 evolutionary algorithms review," *Genetic programming Theory and practice XVII*, pp. 307–344, 2020.
- [26] P. A. Vikhar, "Evolutionary algorithms: A critical review and its future prospects," in *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*. IEEE, 2016, pp. 261–265.
- [27] X. Kong, Y. Yang, Z. Lv, J. Zhao, and R. Fu, "A dynamic dual-population co-evolution multi-objective evolutionary algorithm for constrained multi-objective optimization problems," *Applied Soft Computing*, vol. 141, p. 110311, 2023.
- [28] H. Kneiding and D. Balcells, "Augmenting genetic algorithms with machine learning for inverse molecular design," *Chemical Science*, vol. 15, no. 38, pp. 15 522–15 539, 2024.
- [29] B. Selman and C. P. Gomes, "Hill-climbing search," *Encyclopedia of cognitive science*, vol. 81, no. 333–335, p. 10, 2006.
- [30] L. Hong and T. Kortemme, "An integrative approach to protein sequence design through multiobjective optimization," *PLoS computational biology*, vol. 20, no. 7, p. e1011953, 2024.

- [31] F. Xue, A. Kubaney, Z. Guo, J. K. Min, G. Liu, Y. Yang, and D. Baker, "Improving protein sequence design through designability preference optimization," *arXiv preprint arXiv:2506.00297*, 2025.
- [32] M. S. Shaikh, H. Lin, S. Xie, X. Dong, Y. Lin, C. K. Shiva, and W. F. Mbasso, "An intelligent hybrid grey wolf-particle swarm optimizer for optimization in complex engineering design problem," *Scientific Reports*, vol. 15, no. 1, p. 18313, 2025.
- [33] P. Moscato *et al.*, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," *Caltech concurrent computation program, C3P Report*, vol. 826, no. 1989, p. 37, 1989.
- [34] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.
- [35] W. E. Hart, N. Krasnogor, and J. E. Smith, "Recent advances in memetic algorithms," 2006.
- [36] K. G. Reddy and D. Mishra, "Advances in feature selection using memetic algorithms: A comprehensive review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 15, no. 2, p. c70026, 2025.
- [37] J. Zhou, Z. Ji, Z. Zhu, and S. He, "Compression of next-generation sequencing quality scores using memetic algorithm," *BMC bioinformatics*, vol. 15, no. Suppl 15, p. S10, 2014.
- [38] A. Blanco-Gonzalez, A. Cabezon, A. Seco-Gonzalez, D. Conde-Torres, P. Antelo-Riveiro, A. Pineiro, and R. Garcia-Fandino, "The role of ai in drug discovery: challenges, opportunities, and strategies," *Pharmaceuticals*, vol. 16, no. 6, p. 891, 2023.
- [39] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [40] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. ieee, 1995, pp. 1942–1948.
- [41] S. Forrest, "Genetic algorithms," *ACM computing surveys (CSUR)*, vol. 28, no. 1, pp. 77–80, 1996.
- [42] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [43] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.
- [44] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849–872, 2019.
- [45] A. Manzoor, M. A. Judge, S. u. Islam, N. Neggaz, L. Abulaigh, and I. Ahmad, "Ahho: Arithmetic harris hawks optimization algorithm for demand side management in smart grids," *Discover Internet of Things*, vol. 3, no. 1, p. 3, 2023.
- [46] M. Abdel-Basset, R. Mohamed, and M. Abouhawwash, "Crested porcupine optimizer: A new nature-inspired metaheuristic," *Knowledge-Based Systems*, vol. 284, p. 111257, 2024.
- [47] E.-S. M. El-Kenawy, N. Khodadadi, S. Mirjalili, A. A. Abdelhamid, M. M. Eid, and A. Ibrahim, "Greylag goose optimization: nature-inspired optimization algorithm," *Expert Systems with Applications*, vol. 238, p. 122147, 2024.
- [48] M. A. Al-Betar, M. A. Awadallah, M. S. Braik, S. Makhadmeh, and I. A. Doush, "Elk herd optimizer: a novel nature-inspired metaheuristic algorithm," *Artificial Intelligence Review*, vol. 57, no. 3, p. 48, 2024.
- [49] M. H. Amiri, N. Mehrabi Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm," *Scientific Reports*, vol. 14, no. 1, p. 5032, 2024.
- [50] W. J. AL-kubaisy and B. AL-Khatib, "Quokka swarm optimization: A new nature-inspired metaheuristic optimization algorithm," *Journal of Intelligent Systems*, vol. 33, no. 1, p. 20240051, 2024.
- [51] M. Han, Z. Du, K. F. Yuen, H. Zhu, Y. Li, and Q. Yuan, "Walrus optimizer: A novel nature-inspired metaheuristic algorithm," *Expert Systems with Applications*, vol. 239, p. 122413, 2024.
- [52] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13 187–13 257, 2023.
- [53] T. Cai, S. Zhang, Z. Ye, W. Zhou, M. Wang, Q. He, Z. Chen, and W. Bai, "Cooperative metaheuristic algorithm for global optimization and engineering problems inspired by heterosis theory," *Scientific Reports*, vol. 14, no. 1, p. 28876, 2024.
- [54] I. Dagal, A. Harrison, W. F. Mbasso, F. N. Kemdoum, M. Khishe, P. Jangir, A. Smerat, S. F. Al-Gahtani, Z. Elbarbary, E. F. Donfack *et al.*, "Logarithmic mean optimization a metaheuristic algorithm for global and case specific energy optimization," *Scientific Reports*, vol. 15, no. 1, p. 18155, 2025.
- [55] V. Osuna-Enciso, E. Cuevas, and B. M. Castañeda, "A diversity metric for population-based metaheuristic algorithms," *Information Sciences*, vol. 586, pp. 192–208, 2022.
- [56] W. H. Lim and N. A. M. Isa, "Two-layer particle swarm optimization with intelligent division of labor," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2327–2348, 2013.
- [57] T. Dokerooglu, D. Canturk, and T. Kucukyilmaz, "A survey on pioneering metaheuristic algorithms between 2019 and 2024," *arXiv preprint arXiv:2501.14769*, 2024.
- [58] O. E. Turgut, M. S. Turgut, and E. Kirtepe, "A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems," *Neural Computing and Applications*, vol. 35, no. 19, pp. 14 275–14 378, 2023.
- [59] K. S. Sarkisyan, D. A. Bolotin, M. V. Meer, D. R. Usmanova, A. S. Mishin, G. V. Sharonov, D. N. Ivankov, N. G. Bozhanova, M. S. Baranov, O. Soylemez *et al.*, "Local fitness landscape of the green fluorescent protein," *Nature*, vol. 533, no. 7603, pp. 397–401, 2016.
- [60] S. A. Kauffman, "Origins of order in evolution: self-organization and selection," in *Understanding origins: Contemporary views on the origin of life, mind and society*. Springer, 1992, pp. 153–181.
- [61] A. Kumar, K. V. Price, A. W. Mohamed, A. A. Hadi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2022 special session and competition on single objective bound constrained numerical optimization," Nanyang Technological University, Tech. Rep., December 2021, available at: <https://github.com/P-N-Suganthan/2022-SO-BO>.
- [62] P. N. Suganthan, "Cec 2022 special session / competition on single objective bound constrained optimization," [https://www3.ntu.edu.sg/home/epnsugan/index\\_files/CEC2022/CEC2022.htm](https://www3.ntu.edu.sg/home/epnsugan/index_files/CEC2022/CEC2022.htm), 2022, accessed: 10-Aug-2025.
- [63] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *Journal of the royal society interface*, vol. 15, no. 141, p. 20170387, 2018.
- [64] N. Sapoval, A. Aghazadeh, M. G. Nute, D. A. Antunes, A. Balaji, R. Baraniuk, C. Barberan, R. Dannenfelser, C. Dun, M. Edrisi *et al.*, "Current progress and open challenges for applying deep learning across the biosciences," *Nature Communications*, vol. 13, no. 1, p. 1728, 2022.
- [65] D. van Tilborg, H. Brinkmann, E. Criscuolo, L. Rossen, R. Özçelik, and F. Grisoni, "Deep learning for low-data drug discovery: Hurdles and opportunities," *Current Opinion in Structural Biology*, vol. 86, p. 102818, 2024.
- [66] C. Audet and M. Kokkolaras, "Blackbox and derivative-free optimization: theory, algorithms and applications," *Optimization and Engineering*, vol. 17, no. 1, pp. 1–2, 2016.
- [67] N. Tokuriki, F. Stricher, L. Serrano, and D. S. Tawfik, "How protein stability and new functions trade off," *PLoS computational biology*, vol. 4, no. 2, p. e1000002, 2008.
- [68] P. Mistani and V. Mysore, "Preference optimization of protein language models as a multi-objective binder design paradigm," *arXiv preprint arXiv:2403.04187*, 2024.
- [69] J. A. Stern, T. J. Free, K. L. Stern, S. Gardiner, N. A. Dalley, B. C. Bundy, J. L. Price, D. Wingate, and D. Della Corte, "A probabilistic view of protein stability, conformational specificity, and design," *Scientific Reports*, vol. 13, no. 1, p. 15493, 2023.
- [70] G. Schneider, "Automating drug discovery," *Nature reviews drug discovery*, vol. 17, no. 2, pp. 97–113, 2018.
- [71] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *nature*, vol. 596, no. 7873, pp. 583–589, 2021.
- [72] X. Zeng, F. Wang, Y. Luo, S.-g. Kang, J. Tang, F. C. Lightstone, E. F. Fang, W. Cornell, R. Nussinov, and F. Cheng, "Deep generative molecular design reshapes drug discovery," *Cell Reports Medicine*, vol. 3, no. 12, 2022.