

Research Article

Practical Topology Optimization with Deep Reinforcement Learning and Genetic Algorithms

Giorgi Tskhondia* 

Independent Researcher, Tbilisi, Georgia

Abstract

Despite the rapid adoption of Large Language Models across many domains, identifying artificial intelligence applications that deliver clear, high-impact value in industrial engineering remains a significant challenge. This work addresses that gap by presenting a comprehensive topology optimization framework tailored for engineering design problems, integrating deep reinforcement learning (DRL), genetic algorithms (GA), and finite element methods (FEM). The proposed framework makes several key contributions. First, it demonstrates substantial performance improvements by achieving a 12×12 grid resolution in two-dimensional topology optimization using a purely DRL-based approach, enabled by carefully designed smart reward shaping strategies. Second, it introduces a hybrid DRL–GA methodology that leverages the complementary strengths of learning-based exploration and evolutionary optimization, resulting in consistently improved solutions compared to standalone DRL. Third, the framework addresses dimensionality scaling challenges by proposing a novel “pseudo-3D” Euclidean space formulation based on a multi-objective optimization strategy. Through a systematic 2D deconstruction approach, this enables effective optimization at a 15×15 grid resolution while mitigating the computational burden of full 3D simulations. Finally, the practical relevance of the framework is validated through an applied engineering case study: the design and optimization of a rocket nozzle. This application highlights the framework’s potential for real-world, high-value industrial use cases.

Keywords

Topology Optimization, Reinforcement Learning, Genetic Algorithms, Finite Element Methods, Hybrid, Pseudo 3D, Nozzle Design

1. Introduction

Topology optimization (TO) is a computational approach aimed at determining the optimal material layout within a given design space to maximize performance under specified loads, boundary conditions, and constraints. It is widely used in various engineering domains, including structural, fluid, acoustic, and thermal design. Several methodologies exist for implementing TO, such as homogenization-based methods,

density-based methods, evolutionary algorithms, and boundary variation techniques. These approaches typically rely on the finite element method (FEM), which is well-suited for modelling complex geometries and nonlinear behaviours.

In my work, I introduce a topology optimization approach that integrates FEM with deep reinforcement learning, producing novel and efficient topologies. Reinforcement learning

*Corresponding author: gigatskhondia@gmail.com (Giorgi Tskhondia)

Received: 6 January 2026; **Accepted:** 14 January 2026; **Published:** 27 January 2026



Copyright: © The Author(s), 2026. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

is a global, gradient free, learning based, generalizable alternative to classical optimisation methods. RL can deal with non-linear non-convex design spaces.

A critical challenge in this approach is scalability. Applying RL for topology optimization in isolation may not be practical due to the substantial computational resources required for common compute availabilities. To address this, I have proposed several techniques to make this method more practical for industrial needs. Namely, I have investigated:

For 2D pure DRL case: sparse rewards; smart skipping (where I optimize the number of FE calculations required); “one island” strategy (where I incentivize topology to be simply connected); “smooth shape” strategy (where I incentivize topology to be of a smooth shape); smart post-processing (removing elements with less than 2 neighbors i.e. isolated elements). Applying these strategies led to considerable improvements on benchmarks for TO with pure DRL. In particular, I was able to get topology for 12 by 12 grid (compared to 6 by 6 grids in previous works, [1, 2]).

I propose an innovative hybrid approach that combines genetic algorithms (GA) with RL to accelerate the optimization process for larger grids (e.g., 25x25 grid). My methods typically involve using GA to generate an initial design outline, followed by RL for fine-tuning and refinement.

For 3D space: I explore multi-objective 2D projection-based strategies as a computationally efficient alternative for achieving 3D topology optimization (which I call “Pseudo 3D”). The key advantage is that, instead of directly optimizing in 3D (which is computationally intensive), we can perform three parallel 2D optimizations. During training, a 3D structure is reconstructed on-the-fly from the three 2D outputs and evaluated for compliance. To reconstruct the 3D structure, you simply take the intersection of the extrusions from the three 2D projections. Remarkably, by using this strategy for 2D case, I was able to get 15 by 15 grid by pseudo 3D deconstruction into 2D plane. I also consider multi-objective optimization experiments that led to *Pseudo 3D* strategy.

Finally, some initial results on a rocket nozzle design by DRL are presented.

2. Methodology

2.1. Reinforcement Learning Applied to Topology Optimization

We are looking at a standard reinforcement learning scenario where an *agent* interacts with an *environment* in discrete time steps. The agent receives a *state* (s_t), selects an *action* (a_t) using its *policy* (π) and then gets a new state (s_{t+1}) along with a *reward* (r_t). This cycle repeats until a terminal state is reached. The agent's goal is to maximize the *total discounted return* (R_t), aiming for the highest expected return from each state.

The *action value* $Q^\pi(s, a)$ tells us the expected total reward for taking action a in state s and continuing with policy π . The *optimal action value* $Q^*(s, a)$ is simply the best possible expected reward for a given state and action.

In the context of topology optimization, the finite element model acts as the environment. The agent, powered by a neural network, takes actions that modify the topology. Each modification triggers a finite element analysis (FEA), which then provides the new state's information back to the neural network (alternatively, FEA is triggered after some accumulation of actions). This loop continues, with the agent receiving rewards for minimizing *compliance* (a common optimization objective). The final output of this process, after the inference stage, is an optimized topology. During inference, the agent applies changes to the topology based solely on its current observations in a greedy manner.

The conceptualization of engineering design as a complete reinforcement learning problem is more thoroughly discussed in [1].

2.2. Proximal Policy Optimization (PPO)

PPO is a reinforcement learning algorithm that boosts training stability by restricting policy updates from becoming too large. It does this by calculating the ratio between the current and old policy probabilities and then clipping this ratio within a specific interval, such as $[1-\epsilon, 1+\epsilon]$.

PPO's strength lies in its ability to deliver the data efficiency and consistent performance of *Trust Region Policy Optimization* (TRPO) while solely relying on first-order optimization methods. Its core innovation is a novel objective function that uses clipped probability ratios to create a conservative (lower-bound) estimate of how well the policy is performing. To optimize policies, PPO iteratively samples data from the current policy and then performs several optimization steps on that sampled data, [3].

2.3. Genetic Algorithm (GA)

According to Mitchell (1996) and Whitley (1994), a Genetic Algorithm is a type of metaheuristic in computer science and operations research. It is inspired by natural selection and falls under the broader category of evolutionary algorithms (EA).

GAs are widely used to find high-quality solutions for optimization and search problems. They do this by mimicking biological processes like mutation, crossover, and selection. In a GA, a group of potential solutions, called “individuals” or “phenotypes,” are gradually improved over time. Each solution has a set of characteristics, or “chromosomes,” that can be changed or “mutated.” While solutions are often represented as binary strings (0s and 1s), other formats, like continuous variables, are also possible.

The evolution typically begins with a random set of indi-

viduals and proceeds iteratively, with each step called a "generation." In each generation, the "fitness" of every individual is assessed. This fitness is usually determined by how well the solution performs against the problem's objective. Fitter individuals are then chosen to create the next generation, where their "genomes" are modified through recombination and potential random mutations. This new generation then becomes the input for the next iteration. The algorithm typically stops when a set number of generations has passed or when the population achieves a satisfactory fitness level.

3. Focus and Contributions of This Article

I explore several techniques to improve benchmarks for 2D topology optimization by pure deep reinforcement learning. Previously reported benchmark was for a 6 by 6 grid [1, 2], whereas I was able to obtain a 12 by 12 grid after applying my methods as I experimented with: (a) sparse rewards; (b) smart skipping (where I optimize the number of FE calculations required); (c) "one island" strategy (where I incentivize topology to be simply connected); (d) "smooth shape" strategy (where I incentivize topology to be of a smooth shape); (e) smart post-processing (removing elements with less than 2 neighbors i.e. isolated elements). Because of sparse rewards, I was able to run my methods under 35 min wall-clock time on my Mac M1 CPU. The number of states that can be optimized by these improvements is more than there are stars in the observable Universe, which opens the door for practical application of reinforcement learning to industrial topology optimization.

To further improve on dimensions of accessible design spaces, a practical imperative of hybrid reinforcement learning (RL) approaches for scalability in topology optimization is a crucial theme emerging from my research. While my work and that of by other researchers demonstrates the fundamental feasibility of RL for 2D topology optimization [1, 2], my explicit statement regarding the impracticality of standalone RL for topology optimization due to computational demands, especially for larger design spaces, highlights a significant bottleneck, [1, 4, 5]. This observation is not merely a limitation but points to a strategic direction for the field. The computational cost and sample efficiency challenges associated with RL, [6], become particularly acute as the complexity and dimensionality of engineering design problems increase. My work directly addresses this by advocating for and demonstrating *hybrid approaches* (combining GA with RL) and exploring *hierarchical RL* (HRL), [1]. This is not simply an incremental improvement; it represents a strategic response to a fundamental computational constraint. Prominence in this area stems from my focus on *practicality* and *accelerating*

learning in real-world engineering contexts. This suggests a broader trend: while pure RL possesses immense theoretical power, its successful deployment in practical engineering applications will likely necessitate its integration within a hybrid computational ecosystem, often involving synergistic combinations with other optimization techniques or the implementation of hierarchical structures to manage complexity and reduce computational overhead. This points to a future where RL is a powerful component of a larger, integrated design automation framework.

For 3D space, in Computer-Aided Design (CAD) programs, it's possible to extract 3D surfaces into 2D projections, typically resulting in three orthogonal views. This inspired an idea: perform topology optimization using reinforcement learning on the three 2D projections separately, but in a coordinated, multi-objective manner — where each 2D agent is aware of the others. The key advantage is that, instead of directly optimizing in 3D (which is computationally intensive), we can perform three parallel 2D optimizations. During training, a 3D structure could be reconstructed on-the-fly from the three 2D outputs and evaluated for *compliance* or other objectives. To reconstruct the 3D structure, you simply take the intersection of the extrusions from the three 2D projections. If we apply node by node intersection for the extrusions, many types of complexities for the resulting 3D topology should be possible. This strategy was informed by my experiments with multi-objective optimisation where I wanted to frame hierarchical approach as simultaneous optimization of several RL environments, [4].

Finally, some initial experiments of applying 2D TO with pure DRL for a rocket nozzle design are considered. Here, a good perception of the results can be obtained by a concept of 'comprehensive levels' i.e. the depth in learning and applying Finite Element Analysis while moving from basic principles and linear analysis (like analytical or 2D case) to advanced 3D analysis, [7]. Namely, I utilize analytical solution for a rocket engine's thrust and my algorithm to get a basic outline of 2D nozzle.

4. Experiments

In my experiments, the topology was pinned in the upper left and right corners, the downward force was applied in the lower right corner (I also experimented with other boundary conditions) as in [1].

I first applied PPO reinforcement learning to optimize a 12 by 12 grid with pure deep reinforcement learning with the following tricks: (a) sparse rewards; (b) smart skipping; (c) "one island" strategy; (d) "smooth shape" strategy; (e) smart post-processing. As a result, I was able to obtain topologies on [Figure 1](#).

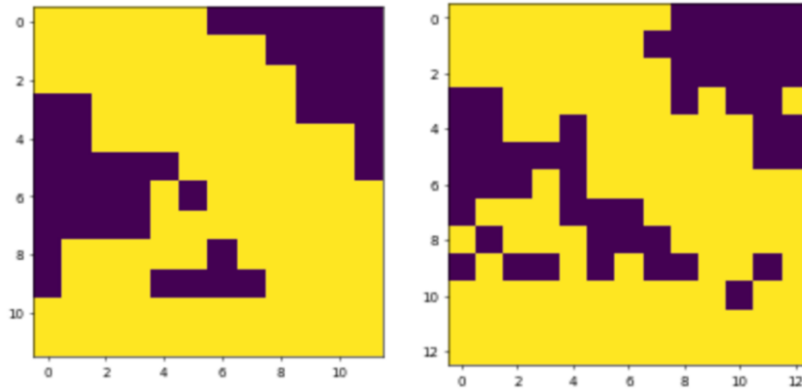


Figure 1. Topologies obtained by pure deep reinforcement learning with sparse rewards, requesting “one island”, requesting “smooth” geometry, and smart post-processing for 12 by 12 (left) and 13 by 13 (right) grids (wall-clock time is ~35 min; 4M iterations).

Based on experiments, which showed enormous potential of pure reward engineering to improve the state of the art of TO with DRL, most actionable tricks were sparse rewards, “one island” and “smooth” topology.

By applying the results from these experiments, and hybrid GA pretraining and RL refinement strategy, [1], I was able to get topology for 25 by 25 grid, Figure 2, which is a considerable improvement over my prior work, [1].

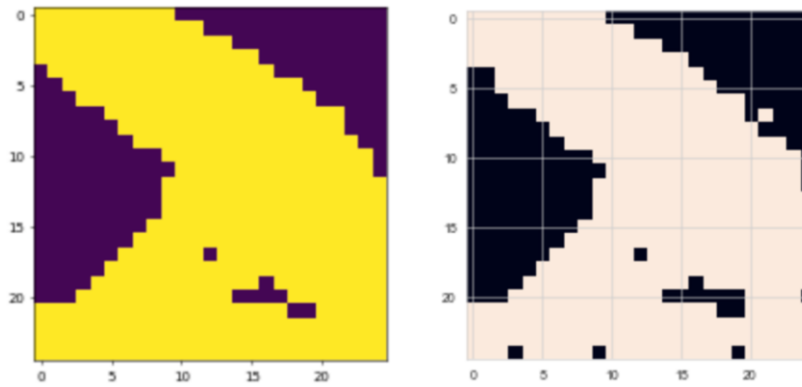


Figure 2. Topology obtained by genetic algorithms pre-training (on the left) and reinforcement learning refinement (on the right) with sparse rewards for 25 by 25 grid (total wall-clock time is ~ 2hr).

For 3D space, I optimized a 3x3x5 grid in the pseudo 3D manner, [4]. The obtained topology can be seen in Figure 3.

The 3x3x5 grid was fixed at the four corners of X-Y plane and downward force was applied at a single tip node of Z-axis.

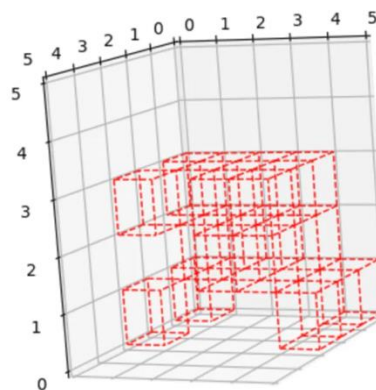


Figure 3. The 3x3x5 cantilever obtained by pseudo-3D topology optimisation with reinforcement learning (1M iterations; ~3hr wall-clock run time).

My experimental approach tackles the computationally demanding problem of 3D topology optimization by leveraging a coordinated multi-objective reinforcement learning (RL) framework across three separate 2D projections. Instead of directly optimizing a complex 3D structure, I employ three independent 2D RL agents, each responsible for optimizing a distinct 2D projection (e.g., X-Y, X-Z, Y-Z planes). The crucial aspect of this setup is that each 2D agent operates with awareness of the others' progress and objectives, fostering a coordinated optimization effort.

During the training phase, a full 3D structure is dynamically reconstructed from the three 2D outputs. This reconstruction is achieved by simply taking the intersection of the extrusions generated by each of the 2D projections. Specifically, a node-by-node intersection of these extrusions is performed. This method allows for the generation of a wide array of complex 3D topologies. The reconstructed 3D structure is then immediately evaluated for performance metrics, such as *compliance*, providing feedback to the individual 2D agents and guiding their learning. This parallel 2D optimization strategy offers a significant computational advantage over direct 3D optimization, making the process more efficient and scalable.

An interesting result was obtained for 15 by 15 grid by to applying pseudo 3D method, [4], for 2D case by deconstructing 3D topology into 2D plane layer by layer (see the codebase and Figure 4). This result bolsters validity of pseudo 3D approach, [4].

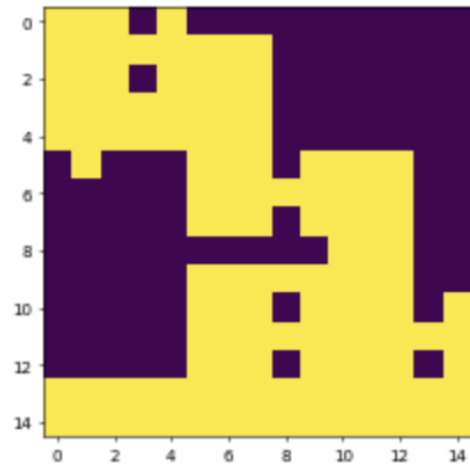


Figure 4. Topology obtained by pure DRL pseudo 3D deconstruction into 2D plane for 15 by 15 grid (total wall-clock time is ~ 1hr).

Finally, as to practical application of my method, some experiments for design of a rocket nozzle, where I utilize analytical solution for the engine's thrust and my algorithm, can be see on Figure 5. The shapes represent a loose analogy with a real nozzle (with a low comprehensive level as per [7]).

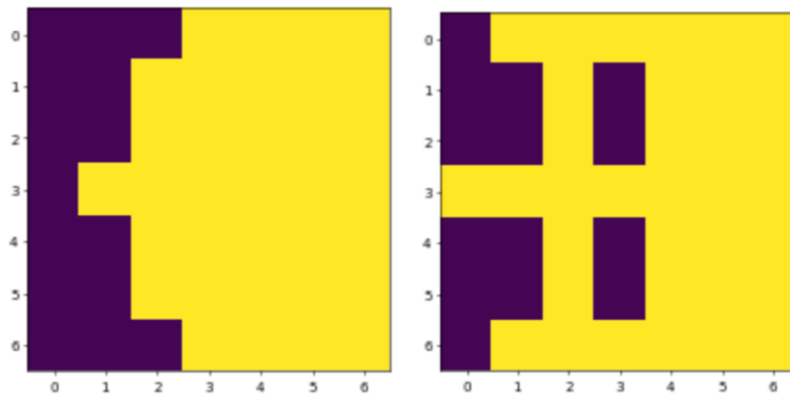


Figure 5. Early experiments on a rocket nozzle design by TO with DRL: Bell Nozzle (on the left) and Spike Nozzle (on the right); (cherry picking).

I trained all my models on Apple M1 Pro. Details on my models can be found in my codebase. 2D FEM model was

taken from [8]. 3D Space Frame Element FEM model was taken from [9].

Table 1. below includes summary of the benchmarks.

Method	(Prior) DRL	Pure DRL	Hybrid DRL-GA	Pseudo 3D	Pseudo 3D deconstruction into 2D plane
Grid dimensions	6x6	12 x 12	25 x 25	3 x 3 x 5	15 x 15

Method	(Prior) DRL	Pure DRL	Hybrid DRL-GA	Pseudo 3D	Pseudo 3D deconstruction into 2D plane
Wall-clock run time	85 min (1M iterations)	35 min (4M iterations)	2hr	3hr (1M iterations)	1hr

5. Discussion

This work serves as a proof of concept, demonstrating a novel approach to generating 3D topologies from 2D designs through the application of a clever heuristic: performing TO on three separate 2D projections of a 3D structure in a coordinated, multi-objective manner. This involves optimizing, for example, the XY, XZ, and YZ views of a design simultaneously. During the training phase, a full 3D structure is dynamically reconstructed by taking the intersection of the extrusions from the three optimized 2D outputs. This reconstructed 3D model is then evaluated for its structural performance, such as *compliance*, providing feedback to guide the ongoing 2D optimizations.

The presented pseudo 3D topology optimisation strategy could significantly reduce computational requirements while still achieving effective 3D topology optimization (for example, a full 3D optimization totally contains 2^{1000} states on a 10 by 10 by 10 grid, my method would contain only 2^{90} states in the same design space. Also for the full 3D optimization you would need 1000 actions, but for my method the three independent environments would only need 100 actions each).

To address a limitation of the *Pseudo 3D* approach on the assumption that a 3D structure can be adequately represented and optimized through coordinated 2D projections, which might not fully capture all intricate 3D topological features as accurately as direct 3D methods, we could combine the pseudo 3D topology optimisation by RL with the enhancement by a direct 3D optimisation via genetic algorithms (GA) on top of the topology obtained by pseudo 3D TO by RL. First, we do RL and then we refine the obtained topology by GA, which is usually faster than RL and as a direct 3D method should allow us obtaining as intricate topology as we possible can. This hybrid strategy (combining RL and GA) can be seen in the lens of my previous work, [1], but contrary to that work here I propose to apply RL first and enhance it with GA, second. Interestingly, the above mentioned limitation might turn out to be an advantage of some kind as 3D shapes obtained by intersection of the extrusions might be easier to manufacture. Alternatively, to address the 3D intricacy limitation instead of 'binary action' we could use a 'triple action': void, material, dashed line. Where, dashed line typically represents hidden features or edges of an object that are not directly visible in the current view.

Beyond the core methods, this work introduces several

novel tricks designed to potentially significantly speed up topology optimization using reinforcement learning: (a) sparse rewards; (b) smart skipping (where I optimize the number of FE calculations required); (c) “one island” strategy (where I incentivize topology to be simply connected); (d) “smooth shape” strategy (where I incentivize topology to be of a smooth shape); (e) smart post-processing (removing elements with less than 2 neighbors i.e. isolated elements).

Also by breaking topology optimisation task to “scale hierarchies”, in hierarchical reinforcement learning (HRL) terms, you can have “workers” designing sub-topologies and a “manager” assembling these sub-topologies together in a bigger topology. In this way, at each abstraction level, your RL agent would take only “lower dimension” actions compared to the case of designing the entire topology in one go, [1].

Another intriguing speculative idea struck me while watching the NVIDIA Keynote at COMPUTEX 2025, where Jensen Huang mentioned that “we are rendering only one out of ten pixels and guessing the remaining nine” using a technology called DLSS. This concept of selectively computing and inferring the rest inspired me to think on applying a similar principle to topology optimization. Instead of calculating the entire topology, what if we compute only a subset of key “anchor” nodes and use a neural network to infer the remaining structure?! The combined layout — consisting of anchor and predicted nodes — could then be evaluated using finite element analysis (FEA) to assess structural compliance, providing feedback to guide the optimization process. This approach would have the potential to significantly accelerate learning and improve computational efficiency.

To further improve the method, in finite element analysis (FEM), mesh size often varies depending on the degree of geometric nonlinearity in different regions. This gave me the idea: why not apply the same principle to the action space in topology optimization!? Instead of using a uniform grid where each action places material in a single square, we could allow actions of varying sizes — some affecting one square, others spanning two, three, or more. This would reduce the total number of actions needed by enabling multi-scale actions. To maintain accuracy, we would adjust the “action mesh” to ensure convergence with respect to mesh resolution (the actual FE mesh could, however, stay uniform). A fast algorithm like SIMP (Solid Isotropic Material with Penalization) could be used initially to identify high-density regions, which would then receive larger actions. Meanwhile, sparse or low-density areas would require smaller or no actions. This strategy could

significantly reduce the size of the action space while preserving optimization quality.

I also completed some multi-scale and multi-objective topology optimisation by RL, Figure 6, that led me to the *pseudo 3D* strategy discussed above.

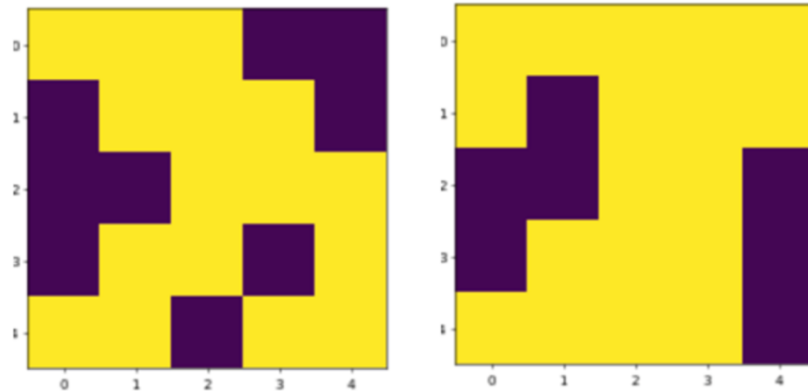


Figure 6. Experiments on multi env topology optimisation with reinforcement learning, [4].

Notably, current boom in artificial intelligence was informed by rapid developments in hardware (like GPUs) as well as aggregation of humongous amounts of data. The volume of data generated each year is bigger than the amount of data generated in all the previous years combined. These data powers AI solutions in various domains from Gen AI to industry application to medicine and beyond. However, despite a big success of LLMs, many companies still struggle to find actionable use cases to really boost the economic growth with AI. In addition, processing all that data in a supervised manner might be prohibitively expensive in terms of the compute required.

On the contrary, my work's intention is twofold. First, topology optimization by reinforcement learning comprises a good use case for applying artificial intelligence to classical engineering design. Second, it provides SOTA method for global, gradient free (in a sense that there are no gradients of *compliance* in the objective function), non-convex, learning based, generalizable topology optimization suitable for practical needs. Sequential nature of reinforcement learning also makes it applicable to technological processes where it can provide manufacturing steps, and to the design of complex machinery with moving parts (for example, imagine there is a robotic manipulator in one position, then because of a sequential nature of RL, you can "erase" the manipulator in that position, and "redraw" it in a different position, thus "optimizing topologies for dynamical events").

By applying deep reinforcement learning (RL), genetic algorithms (GA) and finite element methods (FEM), my approach does not require data, or to put it more precisely, the data is generated in physics simulator (finite elements) on the fly during the training of the RL/GA agent.

The philosophy on my method is that all my algorithms (which I implement in Python programming language – an ideal programming language for rapid prototyping of smaller

to medium topologies) are free, open-source and have practical performance on personal computers, hence could result in a considerable reduction in the ecological footprint in the future. With my approach, I have novelty of current developments in artificial intelligence and safety of classical engineering methods.

The list of my previous contributions (starting from 2018) includes:

- 1) Applying Proximal Policy Optimization (PPO) algorithm for the task of topology optimization (TO) with reinforcement learning
- 2) Applying Hierarchical Reinforcement Learning (HRL) for topology optimization
- 3) Applying a hybrid strategy of "genetic algorithms pre-training and reinforcement learning refinement" for the task of topology optimization
- 4) Framing structural design as a competitive/cooperative game and outlining the application of AlphaZero's algorithm to this task
- 5) Proposing a novel strategy which I called "Pseudo 3D topology optimization with reinforcement learning and genetic algorithms"

Whereas, in this work, I push on the benchmark by some smart techniques which allows to get an improvement of 108 orders of magnitude over previous work, [1, 2].

With pure DRL and applied techniques, I was able to optimize more states than there are stars in the observable Universe. And all of this was done under 35 min wall-clock run time on Mac M1 CPU. With this results, for 12 by 12 grid, I push the benchmark by 108 orders of magnitude over previous work.

With hybrid GA + DRL approach and sparse rewards, I was able to optimize more states than there are atoms in the observable Universe (and even more states than Googol) for 25 by 25 grid. It was done under ~ 2 hr total wall-clock run time on Mac M1 CPU.

Pseudo TO with RL: a multi-objective 2D projection-based strategy that replaces full 3D optimization with coordinated optimization of three 2D views, reconstructing the 3D topology from their intersecting extrusions, was proposed. Applying this method, for 2D case by deconstructing 3D topology into 2D plane layer by layer, a 15 by 15 grid was obtained. The result can be further improved if we apply 'triple action' discussed above.

Finally, some early experiments on a rocket nozzle design by my method were considered.

Potential practical application of topology optimization with RL can be microscopic topology optimization, [10], and metamaterials design, [11]. Or in circuit design, [12-14].

The results obtained in this work open possibly for more practical utilization of deep reinforcement learning for industrial topology optimization with potential applications to space, automotive, offshore and beyond.

Abbreviations

DRL	Deep Reinforcement Learning
GA	Genetic Algorithms
FEM	Finite Element Methods
DRL-	Hybrid Strategy of Deep Reinforcement
GA	Learning and Genetic Algorithms
TO	Topology optimization
FE	Finite Elements
FEA	Finite Element Analysis
PPO	Proximal Policy Optimization
TRPO	Trust Region Policy Optimization
EA	Evolutionary Algorithms
RL	Reinforcement Learning
HRL	Hierarchical Reinforcement Learning
CAD	Computer-Aided Design
DLSS	Deep Learning Super Sampling
SIMP	Solid Isotropic Material with Penalization
GPU	Graphics Processing Unit
Gen AI	Generative Artificial Intelligence

Acknowledgments

I would like to acknowledge all artificial intelligence researchers and structural engineers on top of whose ideas I have built my work. Also, I would like to acknowledge my family who provided me with immense support throughout all these years.

Author Contributions

Giorgi Tskhondia is the sole author. The author read and approved the final manuscript.

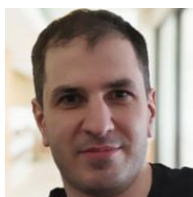
Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] Reinforcement Learning Guided Engineering Design: from Topology Optimization to Advanced Modelling, G. Tskhondia, March 2025, Journal of Next-Generation Research 5 0, 1(3): 1-17, <https://doi.org/10.70792/jngr5.0.v1i3.95>
- [2] Deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains, Nathan K. Brown et al, <https://doi.org/10.1016/j.matdes.2022.110672>
- [3] Proximal Policy Optimization Algorithms, Schulman et al, 2017. <https://arxiv.org/abs/1707.06347>
- [4] [Preprint] Pseudo 3D Topology Optimisation with Reinforcement Learning, G. Tskhondia, June 2025, <https://doi.org/10.13140/RG.2.2.15594.22729>
- [5] [Preprint] Practical topology optimization with deep reinforcement learning, G. Tskhondia, December 2025, <https://doi.org/10.13140/RG.2.2.16548.95369>
- [6] [2502.09417] A Survey of Reinforcement Learning for Optimization in Automation - arXiv, <https://arxiv.org/abs/2502.09417>
- [7] Bathe, Klaus-Jürgen. Finite Element Procedures. 1st ed. Upper Saddle River, NJ: Prentice Hall; 1996, pp. 1-1037.
- [8] [2205.08966] Greydanus, Sam. A Tutorial on Structural Optimization - arXiv, <https://arxiv.org/abs/2205.08966>
- [9] Kattan, Peter I. MATLAB Guide to Finite Elements. 2nd ed. Berlin: Springer; 2008, pp. 1-429.
- [10] Microscopic stress-constrained two-scale topology optimisation for additive manufacturing, Xiaopeng Zhang et al, January 2025 Virtual and Physical Prototyping 20(1) Advanced Modelling, <https://doi.org/10.1080/17452759.2025.2450276>
- [11] Deep reinforcement learning for the design of mechanical metamaterials with tunable deformation and hysteretic characteristics, Nathan K. Brown et al, Materials & Design Volume 235, November 2023, 112428, <https://doi.org/10.1016/j.matdes.2023.112428>
- [12] A graph placement methodology for fast chip design, Azalia Mirhoseini et al, Nature volume 594, pages 207-212 (2021), <https://doi.org/10.1038/s41586-021-03544-w>
- [13] Reinforcement Learning Guided Detailed Routing for Custom Circuits, Hao Chen et al, (NVIDIA, 2023), <https://dl.acm.org/doi/10.1145/3569052.3571874>
- [14] Reinforcement Learning For Automated Chip Floorplanning And Routing Optimization, Hameed Ul Hassan Mohammed , Deng Ying, Volume 21, Number 2, 2024.

Biography



Giorgi Tskhondia (aka Georgy A. Tskhondiya): Independent researcher, PhD in mathematical modeling, ex subsea pipelines installation engineer, data science professional. He maintains Gigala and Ocean Intella open-source software (<https://github.com/gigatskhondia>), and scientific blog <https://gigatskhondia.medium.com/>