

Deep Learning Evidence for Global Optimality of Gerber's Sofa

Kuangdai Leng , Jia Bi, Jaehoon Cha, Samuel Pinilla * and Jeyan Thiagalingam 

Scientific Computing Department, Science and Technology Facilities Council—Rutherford Appleton Laboratory, Didcot OX11 0QX, UK; kuangdai.leng@stfc.ac.uk (K.L.); jia.bi@stfc.ac.uk (J.B.); jaehoon.cha@stfc.ac.uk (J.C.); t.jeyan@stfc.ac.uk (J.T.)

* Correspondence: samuel.pinilla@stfc.ac.uk

Abstract: The moving sofa problem, introduced by Leo Moser in 1966, seeks to determine the maximal area of a 2D shape that can navigate an L -shaped corridor of unit width. Joseph Gerber's 1992 solution, providing a lower bound of approximately 2.2195, is the best known, though its global optimality remains unproven. This paper leverages neural networks' approximation power and recent advances in invexity optimization to explore global optimality. We propose two approaches supporting Gerber's conjecture that his sofa is the unique global maximum. The first approach uses continuous function learning, discarding assumptions about the monotonicity, symmetry, and differentiability of sofa movements. The sofa area is computed as a differentiable function using our "waterfall" algorithm, with the loss function incorporating both differential terms and initial conditions based on physics-informed machine learning. Extensive training with diverse network initialization consistently converges to Gerber's solution. The second approach applies discrete optimization to the Kallus–Romik upper bound, improving it from 2.37 to 2.3337 for five rotation angles. As the number of angles increases, our model asymptotically converges to Gerber's area from above, indicating that no larger sofa exists.

Keywords: moving sofa problem; optimality; deep learning; physics-informed neural network



check for updates

Citation: Leng, K.; Bi, J.; Cha, J.; Pinilla, S.; Thiagalingam, J. Deep Learning Evidence for Global Optimality of Gerber's Sofa. *Symmetry* **2024**, *16*, 1388. <https://doi.org/10.3390/sym16101388>

Academic Editor: Zhixun Su

Received: 29 September 2024

Revised: 12 October 2024

Accepted: 16 October 2024

Published: 18 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Among the unsolved problems in geometry, the moving sofa problem stands out for its apparent simplicity. Formally proposed by Leo Moser [1], it asks for the two-dimensional shape of the largest area that can be maneuvered through an L -shaped corridor with unit width (see Figure 1). Beyond trivial shapes such as the square and half disk, the best-known lower bounds are those found by John Hammersley [2] and Joseph Gerber [3], as shown in Figure 1. Hammersley's sofa, presented in a problem set for school and college students, has an area of $\pi/2 + 2/\pi \approx 2.2074$. It is the largest among the family whose touch point with the inner corner of the corridor forms a semi-circle; he also showed an upper bound of $2\sqrt{2} \approx 2.8284$. With an area slightly larger than 2.2195, Gerber's sofa is the largest known to date, composed of 18 curve sections (as partitioned by their analytical expressions). In a reasonably implicit way, Gerber proved that his solution is a local maximum and conjectured that it is the unique global maximum.

Dan Romik [4] advanced Gerber's analysis by introducing a set of six differential equations along with their general solutions. These solutions, under prescribed contact mechanisms through five phases of rotation, land on Gerber's sofa in a natural and explicit way. Using the same method, Romik also discovered a lower bound of the *ambidextrous sofa* that can pass through a double-cornered corridor. A numerical study [5] was carried out based on Romik's formulation. Yoav Kallus and Dan Romik [6] provided a new perspective on the problem. They proved an upper bound of the maximum area through discrete optimization of the rotation center at a finite sequence of angles. Under a seven-angle setup, they discovered a new upper bound of the area of 2.37, along with a new lower bound of

the least-required rotation angle for maximum area production, approximately 81.2° . We will delve into their findings in greater detail in Section 2.3.

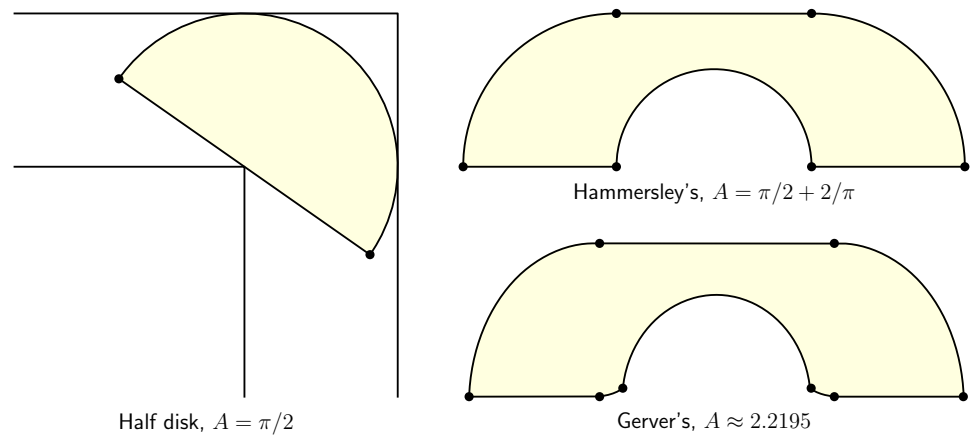


Figure 1. The moving sofa problem proposed by Leo Moser [1] in 1966 and some lower bounds. The best-known two are found, respectively, by John Hammersley [2] and Joseph Gerver [3]. The point markers separate the sections formed by different contact mechanisms (i.e., with either the inner corner, the walls, or the wall envelopes of the corridor).

The past decade has witnessed groundbreaking advancements in deep learning, transforming numerous fields of human activity (e.g., computer vision [7], natural language processing [8], healthcare [9], and environmental sciences [10]). This impact extends to mathematical research, a domain increasingly referred to as *AI4Math*. In the realm of constructing examples or counterexamples—such as discovering new configurations for the moving sofa problem—AI has produced results that surpass traditional human efforts. Notable achievements include the development of faster algorithms for matrix multiplication [11], the discovery of larger cap sets in high dimensions [12], and the identification of more complex knots with specific topological properties [13]. Here, we aim to explore the moving sofa problem through the lens of deep learning. One of the core methodologies we will employ is physics-informed neural networks (PINNs). Originating in the 1990's [14,15] and flourishing within the contemporary deep learning landscape [16,17], PINNs are primarily designed for solving and inverting differential equations. The central concept involves training neural networks not only with data but also by incorporating the target equations along with their initial and boundary conditions. This approach enhances accuracy and generalization while reducing the amount of training data required. We will adhere to the principles of PINNs to address the differential terms involved in sofa area calculation, and the initial conditions of sofa movements. By doing so, we aim to achieve a more precise and efficient solution to the moving sofa problem.

In this paper, we approach the moving sofa problem from an optimization standpoint, employing neural networks to parameterize a function space to identify optimal sofa movements or upper bounds. This method harnesses the universal approximation capability of neural networks and modern optimizers' enhanced, albeit still evolving, proficiency to avoid local minima. Specifically, we take advantage of recent developments in invexity optimization [18,19] to ensure that the employed neural networks will map only global optima. Here, an invex function is a mapping where any critical point is a global minimizer. While preliminary attempts from the community exist (e.g., on GitHub and Stack Overflow), these efforts often fail to consider geometric constraints accurately and lack comprehensive insights into optimality. A significant challenge we have overcome is the computation of the sofa area. Our solution ensures that the area is calculated not only with accuracy and robustness but also in a manner that supports differentiation, as required for backpropagation.

The remainder of the paper is organized as follows. In Section 2, we present two problem formulations—one optimizing sofa area through corridor movement and the other approaching the Kallu-s-Romik upper bound [6]. Section 3 describes our deep learning methodology, highlighting the generality of the function space and diverse initialization to reduce inductive bias. Finally, Section 4 presents our findings, reinforcing the global optimality of Gerver’s sofa. To the best of our knowledge, this is the first study to leverage deep learning in addressing the moving sofa problem. It represents a substantial advancement by integrating geometric exactness with differentiable optimization techniques.

2. Problem Definition and Formulations

2.1. Sofa Formation

The moving sofa problem can be understood as determining the movement of the L-shaped corridor in \mathbb{R}^2 to maximize the area not swept by its four walls. We describe its movement using the trajectory of its inner corner, denoted by p , and the rotation angle α , as illustrated in Figure 2. This representation allows us to describe the corridor’s position and orientation at any moment. An interactive visualization, assuming an elliptical trajectory for p , has been contributed by the community [20].

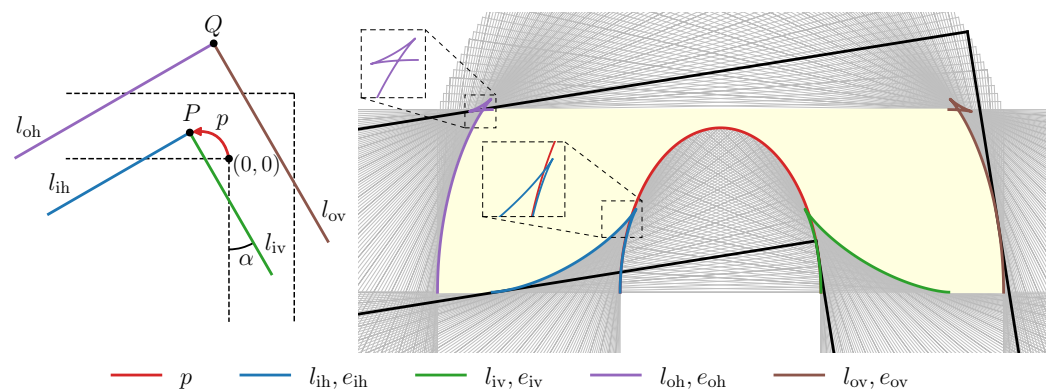


Figure 2. Geometry of the moving sofa problem. The movement of the corridor is described by the trajectory of its inner corner P , as denoted by p , and the rotation angle α . The four walls form the four families of lines: l_{ih} , l_{iv} , l_{oh} and l_{ov} , with the subscripts showing their initial positions (i for inner, o for outer, h for horizontal and v for vertical). Their envelopes are, respectively, e_{ih} , e_{iv} , e_{oh} and e_{ov} . In this example, p is a semi-ellipse with its major and minor lengths being, respectively, 1.8 and 1.1, leading to some complexities in the envelopes, as highlighted by the magnified windows.

Previous studies using continuous optimization have considered (x_p, y_p) as functions of α [3–5], thereby restricting the movement to a monotonic rotation. To overcome this limitation, we introduce a pseudo-time $t \in [0, 1]$, which allows us to describe the movement as follows:

$$\begin{cases} x_p = x_p(t), \\ y_p = y_p(t), \\ \alpha = \alpha(t); \end{cases} \quad \text{satisfying} \quad \begin{cases} x_p(0) = 0, \\ y_p(0) = 0, \\ \alpha(0) = 0. \end{cases} \quad (1)$$

The above initial conditions at $t = 0$ are trivial, addressed by the translation and rotation invariance of \mathbb{R}^2 . A non-trivial condition is

$$\alpha(1) > \arcsin \frac{84}{85} \approx 81.2^\circ, \quad (2)$$

that is, any moving sofa shape of the largest area must undergo rotation by an angle of at least 81.2° [6]. We will later use this condition in our loss function. The parameterization in Equation (1) only requires that $x_p(t)$, $y_p(t)$ and $\alpha(t)$ are of class C^0 , allowing for independent translation and rotation that are non-monotonic, non-symmetric, and non-differentiable.

The movement of the corridor will determine the following four families of lines, named by their initial position at $t = 0$ (see Figure 2):

$$\begin{aligned}
 \text{Inner horizontal, } l_{ih} : & \quad (x - x_p) \sin \alpha - (y - y_p) \cos \alpha = 0; \\
 \text{Inner vertical, } l_{iv} : & \quad (x - x_p) \cos \alpha + (y - y_p) \sin \alpha = 0; \\
 \text{Outer horizontal, } l_{oh} : & \quad (x - x_p) \sin \alpha - (y - y_p) \cos \alpha + 1 = 0; \\
 \text{Outer vertical, } l_{ov} : & \quad (x - x_p) \cos \alpha + (y - y_p) \sin \alpha - 1 = 0.
 \end{aligned} \tag{3}$$

Based on simple calculus, their envelopes can be shown as

$$\begin{aligned}
 \text{Inner horizontal, } e_{ih} : & \quad \begin{cases} x_{ih} = x_p + \cos \alpha \left(x'_p \sin \alpha - y'_p \cos \alpha \right) (\alpha')^{-1}, \\ y_{ih} = y_p + \sin \alpha \left(x'_p \sin \alpha - y'_p \cos \alpha \right) (\alpha')^{-1}; \end{cases} \\
 \text{Inner vertical, } e_{iv} : & \quad \begin{cases} x_{iv} = x_p - \sin \alpha \left(x'_p \cos \alpha + y'_p \sin \alpha \right) (\alpha')^{-1}, \\ y_{iv} = y_p + \cos \alpha \left(x'_p \cos \alpha + y'_p \sin \alpha \right) (\alpha')^{-1}; \end{cases} \\
 \text{Outer horizontal, } e_{oh} : & \quad \begin{cases} x_{oh} = x_{ih} - \sin \alpha, \\ y_{oh} = y_{ih} + \cos \alpha; \end{cases} \\
 \text{Outer vertical, } e_{ov} : & \quad \begin{cases} x_{ov} = x_{iv} + \cos \alpha, \\ y_{ov} = y_{iv} + \sin \alpha, \end{cases}
 \end{aligned} \tag{4}$$

where the primes denote derivatives with respect to t . We assume that $x_p(t)$, $y_p(t)$, and $\alpha(t)$ are piecewise differentiable (piecewise C^1) so that any non-differentiable points can be excluded from the aforementioned envelopes. This assumption is justified, as it is unlikely that the largest sofa could contain some fractal sections. If it does indeed, such sections are inherently undetectable by methods based on analytical parameterization, including deep learning approaches.

2.2. Differentiable Area Calculation

The resultant area is bounded from below by p , e_{ih} , e_{iv} , and $y = 0$, depending on which one is on top at any given point. Similarly, it is bounded from above by e_{oh} , e_{ov} , and $y = 1$, depending on which one is at the bottom. However, it cannot be formulated as a simple integral because these curves can be self-intersecting and intersect at multiple points, as indicated by the zoomed-in windows in Figure 2. Note that Figure 2 only shows a simple case where the trajectory is elliptical; the envelopes can become intricate for arbitrary movements. For example, Figure 3 shows the envelopes created by an irregular movement. This complexity contributes to why the moving sofa problem remains unsolved.

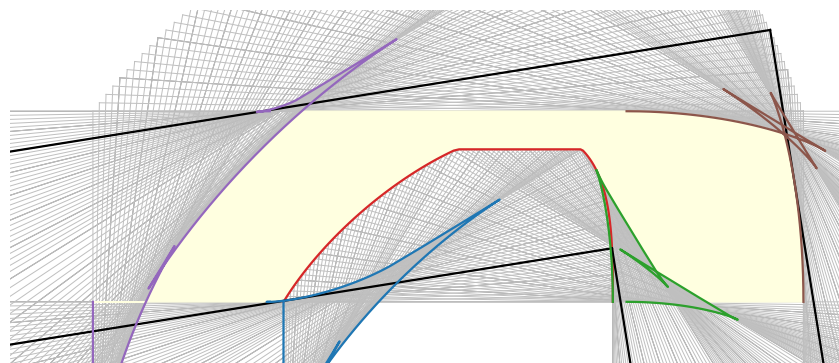


Figure 3. Wall envelopes and sofa shape generated by an irregular corridor movement (red curve). Refer to Figure 2 for detailed descriptions of the geometric elements. The envelope's complex geometry requires a robust algorithm to ensure precise area calculation.

To compute the area robustly for any movement, we developed the *waterfall* algorithm, inspired by the watershed algorithm for image segmentation [21,22]. The watershed algorithm is a region-based image segmentation method that treats the image as a topographic surface, where pixels represent elevation; it identifies “watershed lines” to separate distinct regions. We adapt the concept of using “water” to identify surfaces from above and below. Intuitively, a dense series of water sources aligned horizontally are placed high above, generating a *waterfall* that touches the intersecting curves to form the upper limit for area quadrature. The lower limit is determined similarly, but with the water flowing upward from below. Our implementation ensures differentiability of the resultant area with respect to the curve positions, which is essential for backpropagation. We also parallelize the water source through sensitization to achieve faster computation. Some examples are given in Figure 4.

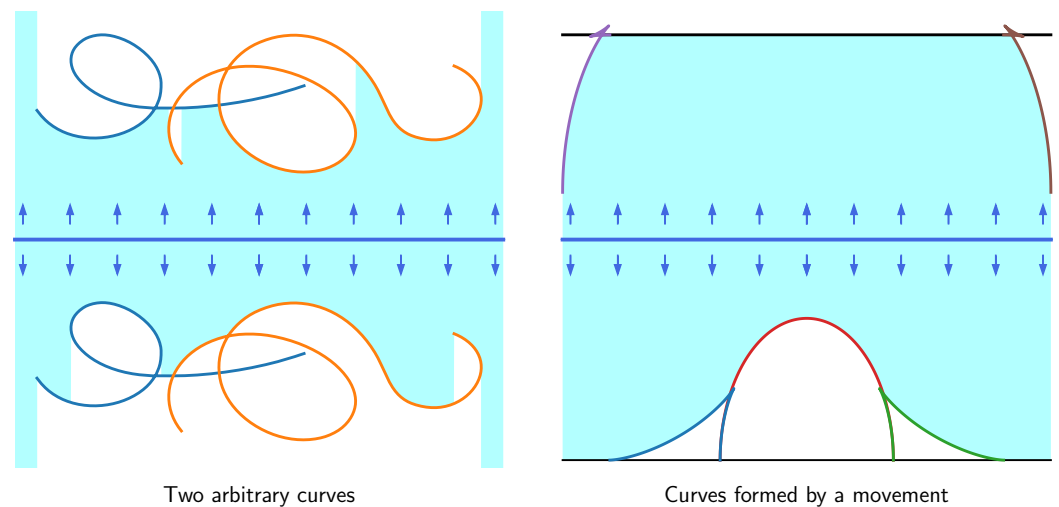


Figure 4. The *waterfall* algorithm for area calculation. Left: The *waterfall* is tested on two hand-drawn curves from both below and above. Right: The *waterfall* is applied to the curves formed by the corridor movement in Figure 2. Water sources are placed along the horizontal line in the middle, with the arrows indicating the falling direction.

We optimize the calculated sofa area as a function of corridor movement, where the targets $x_p(t)$, $y_p(t)$, and $\alpha(t)$ are modeled using neural networks, as detailed in Section 3. The corresponding results are presented in Section 4.1.

2.3. Kallus–Romik Upper Bound

Yoav Kallus and Dan Romik [6] established an upper bound for the maximum sofa area by optimizing the rotation center at a finite sequence of rotation angles. We provide a brief review of their theory and discuss some limitations of their numerical algorithm.

Let $L_\alpha(\mathbf{u})$ denote the set formed by rotating the corridor by an angle α around the center $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$:

$$\begin{aligned}
 L_\alpha(\mathbf{u}) = & \left\{ (x, y) \in \mathbb{R}^2 : u_1 \leq x \cos \alpha + y \sin \alpha \leq u_1 + 1 \right\} \cap \\
 & \left\{ (x, y) \in \mathbb{R}^2 : -x \sin \alpha + y \cos \alpha \leq u_2 + 1 \right\} \cup \\
 & \left\{ (x, y) \in \mathbb{R}^2 : x \cos \alpha + y \sin \alpha \leq u_1 + 1 \right\} \cap \\
 & \left\{ (x, y) \in \mathbb{R}^2 : u_2 \leq -x \sin \alpha + y \cos \alpha \leq u_2 + 1 \right\}.
 \end{aligned} \tag{5}$$

Let $B(\beta_1, \beta_2)$ denote the set formed by rotating the vertical strip by angles β_1 and β_2 , respectively, and taking the union of these two rotations, resulting in a butterfly-shaped region:

$$\begin{aligned}
 B(\beta_1, \beta_2) = & \left\{ (x, y) \in \mathbb{R}^2 : 0 \leq x \cos \beta_1 + y \sin \beta_1 \right\} \cap \\
 & \left\{ (x, y) \in \mathbb{R}^2 : x \cos \beta_2 + y \sin \beta_2 \leq 1 \right\} \cup \\
 & \left\{ (x, y) \in \mathbb{R}^2 : x \cos \beta_1 + y \sin \beta_1 \leq 1 \right\} \cap \\
 & \left\{ (x, y) \in \mathbb{R}^2 : 0 \leq x \cos \beta_2 + y \sin \beta_2 \right\}.
 \end{aligned}
 \tag{6}$$

Let λ denote the area measure on \mathbb{R}^2 , and $\lambda^*(\mathcal{X})$ denote the maximal area of any connected component $\mathcal{X} \subset \mathbb{R}^2$. Then, given a finite sequence of angles, $(\alpha_1, \alpha_2, \dots, \alpha_k)$, and their corresponding rotation centers, $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k)$, and (β_1, β_2) with $\beta_1 \leq \beta_2$, Kallus and Romik [6] first defined the following area:

$$g_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \beta_2}(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k) = \lambda^* \left(H \cap \bigcap_{j=1}^k L_{\alpha_j}(\mathbf{u}_j) \cap B(\beta_1, \beta_2) \right),
 \tag{7}$$

where $H = \mathbb{R} \times [0, 1]$ is the horizontal strip. Taking the supremum of the above area with respect to the centers, they further obtained

$$G_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \beta_2} = \sup \left\{ g_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \beta_2}(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k) : \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k \in \mathbb{R}^2 \right\},
 \tag{8}$$

which they ultimately proved to be an upper bound of the maximum sofa area. Note that $H \cap B(\beta_1, \pi/2) \equiv H$; namely, when $\beta_2 = \pi/2$, the value of β_1 does not affect the resultant area. Therefore, the end case of $G_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \pi/2}$ can simply be denoted by $G_{\alpha_1, \alpha_2, \dots, \alpha_k}$. The above-defined sets, $L_\alpha(\mathbf{u})$ and $B(\beta_1, \beta_2)$, and their induced $g_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \beta_2}$ are illustrated in Figure 5, based on a small number of angles.

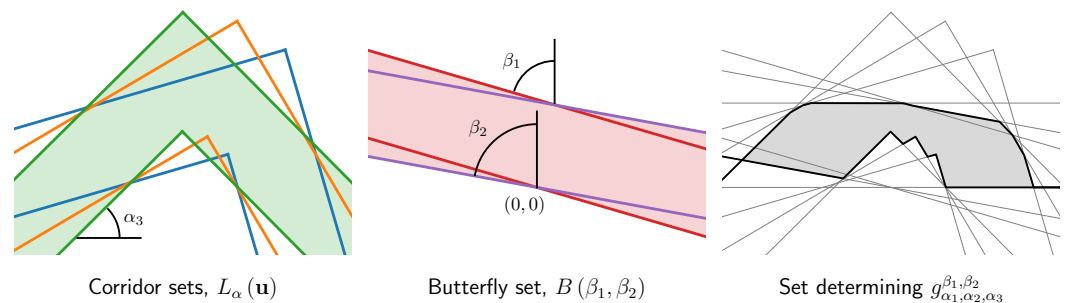


Figure 5. Sets defined for Kallus–Romik upper bound. In this example, the angles are $\alpha_1 \approx 16.26^\circ$, $\alpha_2 \approx 30.51^\circ$, $\alpha_3 \approx 44.76^\circ$, $\beta_1 \approx 73.74^\circ$ and $\beta_2 \approx 79.61^\circ$, corresponding to Equation (28) in [6].

For a sofa shape that moves around the corner while rotating continuously and monotonically from 0 to some angle $\beta \in [0, \pi/2]$, define its maximum area as $A(\beta)$. The global maximum area is then $A_{\max} = \sup_{0 \leq \beta \leq \pi/2} A(\beta)$. The most relevant results from Kallus and Romik [6] are summarized as follows:

- for any $(\alpha_1, \alpha_2, \dots, \alpha_k)$ and (β_1, β_2) satisfying $\alpha_1 < \alpha_2 < \dots < \alpha_k \leq \beta_1 \leq \beta \leq \beta_2 \leq \pi/2$, one has [6] [Proposition 4 (iii)]

$$A(\beta) \leq G_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \beta_2}
 \tag{9}$$

and [6] [Proposition 4 (ii)]

$$A_{\max} \leq G_{\alpha_1, \alpha_2, \dots, \alpha_k}
 \tag{10}$$

2. given an integer $n \geq 3$, let $\gamma_j = \frac{j}{n} \pi$, $j = 1, 2, \dots, n$; [6] [Theorem 5] asserts that

$$A_{\max} = \lim_{n \rightarrow \infty} \max_{k=1, 2, \dots, \lceil n/3 \rceil} G_{\gamma_1, \gamma_2, \dots, \gamma_{n-k-1}}^{\gamma_{n-k}, \gamma_{n-k+1}} \quad (11)$$

which provides a way for constraining A_{\max} asymptotically from above.

The above results are rigorous and elegant from a theoretical perspective. However, the numerical algorithm they present for discrete optimization of the rotation centers, $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k)$, seems to have two shortcomings:

1. Their algorithm does not yield $G_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \beta_2}$ but instead provides another upper bound whose tightness is unproven; though this computed one is also an upper bound of $A(\beta)$, there can be space for improving the tightness if $G_{\alpha_1, \alpha_2, \dots, \alpha_k}^{\beta_1, \beta_2}$ is evaluated directly.
2. Based on rational programming, their algorithm is computationally expensive and unscalable, which can only tackle a small number of angles. Consequently, Equation (11) or their Theorem 5 remains unexploited for constraining the actual value of A_{\max} .

These shortcomings motivate our work reported in Sections 4.2 and 4.3.

3. Deep Learning Methodology

3.1. Fully Connected Networks

Neural networks, particularly fully connected networks (FCNs), play a central role in many machine learning tasks due to their ability to approximate complex functions. An FCN consists of multiple layers of nodes, where each node applies a weighted sum to its inputs, followed by an activation function to introduce non-linearity. This structure allows FCNs to act as universal approximators, capable of representing a broad range of continuous functions [23,24]. In deep learning, such models are trained to optimize parameters that minimize a loss function, guiding the model towards accurate predictions. The rectified linear unit (ReLU) activation function, commonly used in FCNs, introduces piecewise linearity [25]. This property makes ReLU-based FCNs ideal for exploring a broad function class, as required by the optimization problems discussed here.

In Appendix A, we provide a theoretical review of FCNs, including their formulation, universal approximation ability, and characterization of invexity. Universal approximation increases the likelihood that the global maximum is included in the solution space. Invexity enhances the probability that the first-order, gradient-based optimizers used in deep learning, featuring some capability to escape from local minima, can successfully locate the global minima. Together, these properties reinforce the appeal of our conclusions from extensive function search.

3.2. Network Parameterization of Corridor Movement

Distinct from typical deep learning tasks focused on data fitting, our application is not primarily concerned with the accuracy or generalization of a single best model. Instead, our objective is to support the global optimality of Gerver's sofa by demonstrating that multiple models from our over-parameterized function space converge to Gerver's lower bound. Consequently, we want our function space, or hypothesis class, to be as general as possible. We parameterize $x_p(t)$, $y_p(t)$, and $\alpha(t)$ by three independent FCNs with ReLU as the activation function. FCNs are chosen for their universal approximation capabilities [23,24], and ReLU is selected for its ability to provide piecewise linear parameterization [25,26]. It is worth mentioning that using C^∞ activation functions such as \tanh and softplus instead of ReLU, would limit the search space to C^∞ functions. Though they can lead to faster convergence to Gerver's sofa, our approach favors piecewise C^1 functions for broader generality. As explained, piecewise C^1 is likely the most general class of functions that can be explored using non-fractal methods.

Specifically, we mathematically show that our proposed neural networks map to global optima (see Appendix A). We remark that this theoretical result plays a crucial role in addressing the moving sofa problem. It ensures that any critical point corresponds to a

global optimum (invexity definition [27]), a necessary condition for achieving the global optimality of Gerver’s sofa. In Section 4, we will provide numerical validation for this optimality by illustrating the landscapes of our network parameterization.

In general, neural network optimization, while gradient-based, does not guarantee convergence to a global minimum. However, contemporary optimizers like Adam [28,29], which incorporate adaptive learning rates and momentum, enhance the probability of escaping local minima. Additionally, the likelihood of attaining the global minimum increases if multiple approximators, initialized diversely, converge to the same local minimum. In this context, “diverse initialization” extends beyond mere variation in weights and biases; it implies that approximators approach their resultant minima from distinct trajectories in the parameter space. This requires meticulous scaling of the weight sampling distribution. For instance, our target function $\alpha(t)$ lies within $[0, \pi/2]$, whose approximators should hence be initialized to safely cover $[0, \pi/2]$. This consideration diverges from typical machine learning practices. Figure 6 illustrates a subset of our initialization samples.

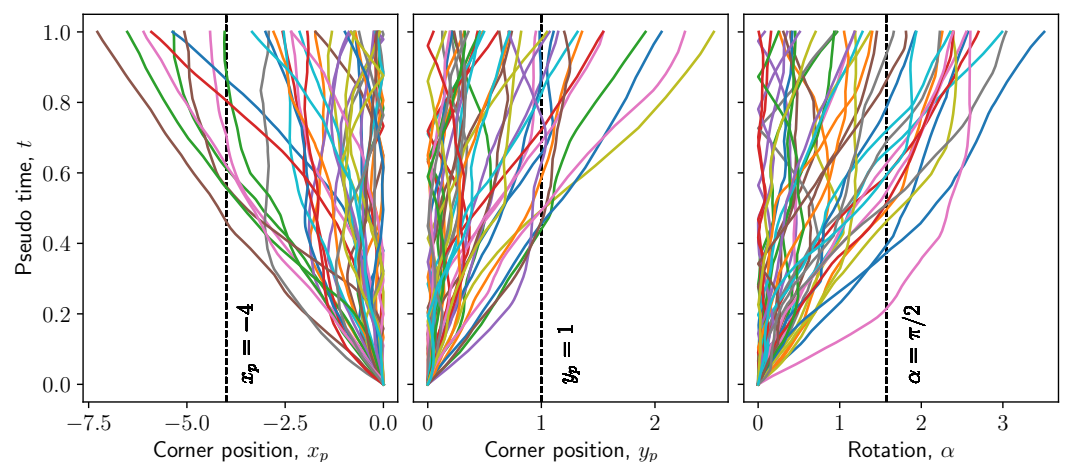


Figure 6. Function initialization with network weights sampled from scaled uniform distributions. Each function is plotted in a different color to distinguish the individual curves. We sample weights and biases from $\mathcal{U}(-s\sqrt{k}, s\sqrt{k})$, where k is the reciprocal of input size and s a number we obtain for each target function by trial and error until its admissible range is safely covered from above, as indicated by the dashed lines. Note that $s = 1$ is PyTorch default.

3.3. Physics-Informed Architecture

Our neural networks are physics informed, leveraging principles outlined in the recent literature [16,17]. Firstly, as indicated in Equation (4), the computation of the envelopes—and consequently the area—depends on the derivatives of the learned functions (x'_p , y'_p , and α'), which are efficiently calculated via automatic differentiation (AD). Secondly, we adhere to the initial conditions specified in Equation (1) directly within the network architecture, while the non-trivial condition in Equation (2) is incorporated into the loss function due to its unequal nature. Additionally, it is straightforward to see that a movement involving $x_p > 0$, $y_p < 0$, or $\alpha < 0$ is forbidden by geometry. Combing all of these considerations, the functions describing a corridor movement are approximated by

$$\begin{cases} x_p(t) = -|\mathcal{F}_{x_p}(t) - \mathcal{F}_{x_p}(0)|, \\ y_p(t) = |\mathcal{F}_{y_p}(t) - \mathcal{F}_{y_p}(0)|, \\ \alpha(t) = |\mathcal{F}_{\alpha}(t) - \mathcal{F}_{\alpha}(0)|, \end{cases} \quad (12)$$

where \mathcal{F}_{x_p} , \mathcal{F}_{y_p} , and \mathcal{F}_α are unconstrained FCNs. With the area A computed by our *waterfall* algorithm based on $x_p(t)$, $y_p(t)$, and $\alpha(t)$, the following loss function is adopted for backpropagation:

$$L = -A + \text{ReLU}(81.2^\circ - \alpha(1)), \quad (13)$$

where the ReLU deactivates the penalizing term once $\alpha(1)$ exceeds 81.2° . The complete architecture is summarized in Figure 7.

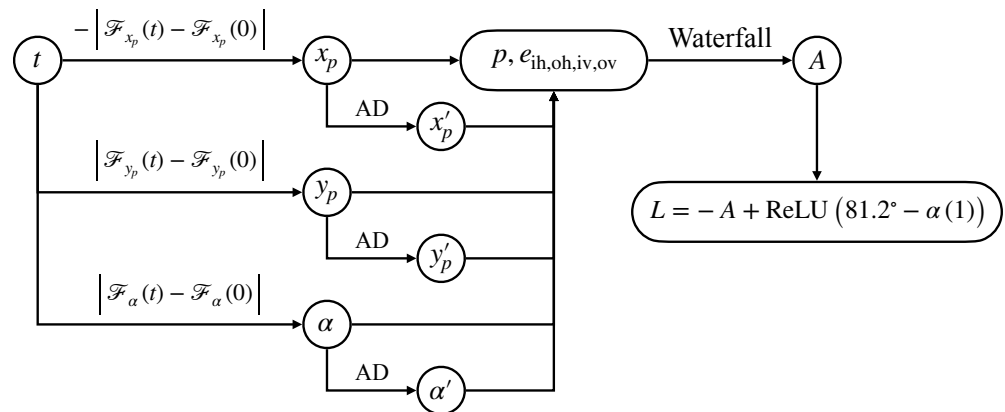


Figure 7. Schematic of our physics-informed network architecture and loss function. The \mathcal{F} 's represent unconstrained ReLU-based FCNs. Once $x_p(t)$, $y_p(t)$, and $\alpha(t)$ are computed, automatic differentiation (AD) is applied to obtain their derivatives, which are then used in Equation (4) to calculate the envelopes, and subsequently, the sofa area via our *waterfall* algorithm.

For the Kallus–Romik upper bound, we use unconstrained FCNs to parameterize the functions $u_1(\alpha)$ and $u_2(\alpha)$; that is, $u_{1,2}(\alpha) = \mathcal{F}_{u_{1,2}}(\alpha)$, with no initial conditions and sign enforcement. The *waterfall* algorithm is applied directly on the four rays in $L_\alpha(\mathbf{u})$ and the two lines in $B(\beta_1, \beta_2)$, instead of on envelopes. Note that envelopes do not exist because the upper bound is defined with discrete angles. These two differences indicate that our neural networks are no longer informed by any “physics”, which is consistent with the notion that the Kallus–Romik upper bound is not necessarily associated with a valid sofa movement [6].

4. Results

4.1. Physics-Informed Area Optimization

We discretize the interval $t \in [0, 1]$ into 2000 points, with 10,000 sources allocated to the waterfall. This configuration is determined through iterative refinement, ensuring that the area converges to the fourth decimal place when tested on Gerver’s shape. We conduct a comprehensive exploration of 6000 runs, incorporating 1000 random seeds for weight initialization, 3 initial learning rates (10^{-3} , 10^{-4} , and 10^{-5} , halved every 2000 epochs), and 2 network architectures (2 hidden layers of size 256 and 3 hidden layers of size 128). These architectures are selected to practically apply Theorem A2 (Appendix A) under the full row-rank hypothesis. Moreover, since we use ReLU activation, which is strictly increasing in its active domain, the assumptions of Theorem A2 are inherently satisfied. Each run consists of 10,000 epochs, using the Adam optimizer [28]. To ensure high-precision area calculations, we utilize `f10at64`, resulting in a runtime of approximately two hours on the CPU, with minimal performance gain from GPU acceleration. The use of `f10at64` is critical for reliable statistical results, as `f10at32` occasionally produces areas that slightly exceed 2.2195.

In our analysis of the 6000 experimental runs, approximately 8.5% resulted in a vanishing area, primarily occurring when the learning rate is highest. After excluding these outliers, the remaining trials exhibit strong convergence, consistently approaching a small neighborhood around Gerver’s solution, with an area close to 2.219532 [30]. The observed areas follow a normal distribution, $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 2.219482$ and $\sigma = 0.000011$. The

area values range from 2.219459 to 2.219502. Further refinement with 1000 epochs of fine tuning using the L-BFGS algorithm [31] increases the maximum area to 2.219515. Due to resource constraints, L-BFGS is not applied to other models.

We further extend our experiments to include deeper and wider architectures, C^∞ activation functions, and diverse weight sampling distributions for rigorous verification. The models using the tanh activation function yield a strikingly similar distribution while exhibiting accelerated convergence in the initial phases of training, before the area approaches the vicinity of 2.219.

Finally, to visually represent our findings, Figure 8 illustrates the Hessian-based landscape of the sofa area for one of our representative models. This visualization offers compelling evidence for the global optimality of Gerver’s sofa within our network-parameterized function space.

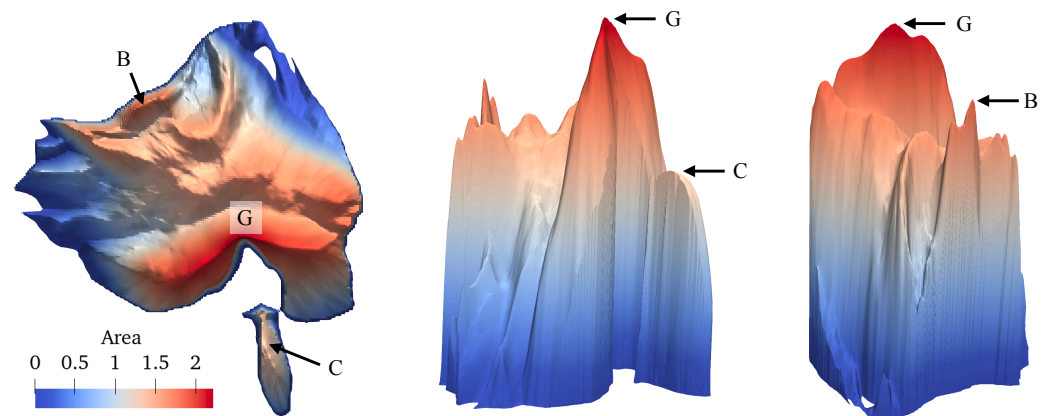


Figure 8. Landscape of sofa area. In this visualized model, the FCNs have three hidden layers of size 128, trained with a learning rate of 10^{-4} . The landscape is anchored by the dominant eigenvectors of the Hessian matrix with respect to model weights [32,33], normalized to unit length. The range of the plot is $[-1.5, 1.5] \times [-1.5, 1.5]$, centered at Gerver’s area (the summit). There are no other peaks in the landscape across $[-10, 10] \times [-10, 10]$. Point G represents Gerver’s solution, the global maximum. Points B and C are local maxima, marked in each viewport to help associate perspectives across the different views.

4.2. Kallus–Romik Upper Bound with Many Angles

The computational cost of our deep learning workflow (*waterfall* and backpropagation) scales with the number of rotation angles in the Kallus–Romik upper bound, allowing us to directly explore the convergence theorem in Equation (11). We increase the number of angles (n) from 10 to 10,000, incremented by 10 between $[0, 100]$, by 100 between $(100, 3000]$ and by 500 afterward, making a total of 53 values for n . For each n , Equation (11) requires $\lceil n/3 \rceil$ training instances, which are still computationally expensive for large n ’s. While Kallus and Romik [6] have shown that 81.2° is the minimum angle, the corridor must rotate for maximum area production; all current evidence suggests that a rotation of $\pi/2$ should take place [3–6], including our results reported in Section 4.1. Therefore, we will vary model initialization only for $k = 1$ (rotation by $\pi/2$) with 20 random seeds (i.e., using 1 seed for $k > 1$). The other hyper-parameters (learning rates, architectures, and epochs) remain the same as those in Section 4.1. The most expensive run for $n = 10,000$, with 10,000 sources for the *waterfall* and 10,000 epochs, takes approximately 10 CPU hours to complete.

Our training results show that, for all values of n ’s, the maximum $G_{\gamma_1, \gamma_2, \dots, \gamma_{n-k-1}}^{\gamma_{n-k}, \gamma_{n-k+1}}$ is achieved when $k = 1$, supporting of the conjecture that the largest sofa necessitates a full rotation by $\pi/2$. We can then focus our attention on $G_{\gamma_1, \gamma_2, \dots, \gamma_{n-2}}^{\gamma_{n-1}, \gamma_n}$ (i.e., $k = 1$). As shown in Figure 9, $G_{\gamma_1, \gamma_2, \dots, \gamma_{n-2}}^{\gamma_{n-1}, \gamma_n}$ found by our models converges asymptotically to Gerver’s area from above. The relative error becomes smaller than 1%, 0.1%, and 0.01% respectively at $n = 30$, 300, and 2100, and finally reaches 0.003% at $n = 10,000$. This result provides another piece

of evidence for the global optimality of Gerver’s sofa, along with the findings presented in Section 4.1.

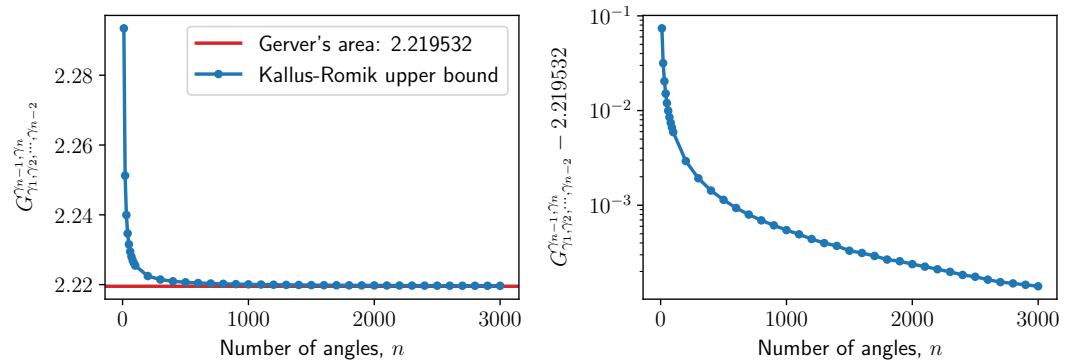


Figure 9. Convergence of $G_{\gamma_1, \gamma_2, \dots, \gamma_{n-2}}^{\gamma_{n-1}, \gamma_n}$ to Gerver’s area, where $\gamma_j = \frac{j}{n} \pi, j = 1, 2, \dots, n$. **(Left)** $G_{\gamma_1, \gamma_2, \dots, \gamma_{n-2}}^{\gamma_{n-1}, \gamma_n}$ in linear scale. **(Right)** $G_{\gamma_1, \gamma_2, \dots, \gamma_{n-2}}^{\gamma_{n-1}, \gamma_n} - 2.219532$ in logarithmic scale.

4.3. Kallus–Romik Upper Bound with Five Angles

An outstanding advantage of the Kallus–Romik upper bound is that it allows for exploring the problem at a low dimension by considering only a few angles. Kallus and Romik [6] defined the following five angles (note that they originally defined seven, but the other two are irrelevant to the sofa area):

$$\begin{aligned}
 \alpha_1 &= \arcsin \frac{7}{25} \approx 16.26^\circ, \\
 \alpha_2 &= \arcsin \frac{33}{65} \approx 30.51^\circ, \\
 \alpha_3 &= \arcsin \frac{119}{169} \approx 44.76^\circ, \\
 \alpha_4 &= \arcsin \frac{56}{65} = \frac{\pi}{2} - \alpha_2 \approx 59.59^\circ, \\
 \alpha_5 &= \arcsin \frac{24}{25} = \frac{\pi}{2} - \alpha_1 \approx 73.74^\circ.
 \end{aligned}
 \tag{14}$$

These angles do not have a specific geometric meaning; they are derived from arcsin functions because the numerical algorithm of Kallus and Romik [6] is rational.

As we have explored the Kallus–Romik upper bound with up to 10,000 rotation angles in Section 4.2, the five-angle case holds less mathematical significance. However, its smaller scale enables us to conduct a more exhaustive model search and visualize the results in terms of the landscape with respect to the rotation centers. Both further reinforce the conclusions drawn in Section 4.2 based on many angles.

The inequalities below combine the optimization results of their algorithm with those of our neural networks for $G_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5}$ and $G_{\alpha_1, \alpha_2, \alpha_3}^{\alpha_4, \alpha_5}$, with ours in the middle:

$$\begin{aligned}
 G_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5} &\leq 2.3337 \pm 0.0000 < 2.37, \\
 G_{\alpha_1, \alpha_2, \alpha_3}^{\alpha_4, \alpha_5} &\leq 1.9259 \pm 0.0000 < 2.21.
 \end{aligned}
 \tag{15}$$

Our results come from 6000 runs, using the same hyper-parameters (learning rates, architectures, seeds and epochs) as described in Section 4.1. The training converges so well that the variances almost vanish (i.e., ± 0.0000 in the inequalities in Equation (15)).

Based on Equation (9), it can be shown that $A_{\max} \leq \max(G_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5}, G_{\alpha_1, \alpha_2, \alpha_3}^{\alpha_4, \alpha_5})$. Therefore, our results suggest a tighter upper bound of 2.3337, corresponding to the set shown in Figure 10. The landscape of $G_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5}$ is also shown in Figure 10, verifying the global optimality of $G_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5}$ in our network-parameterized function space.

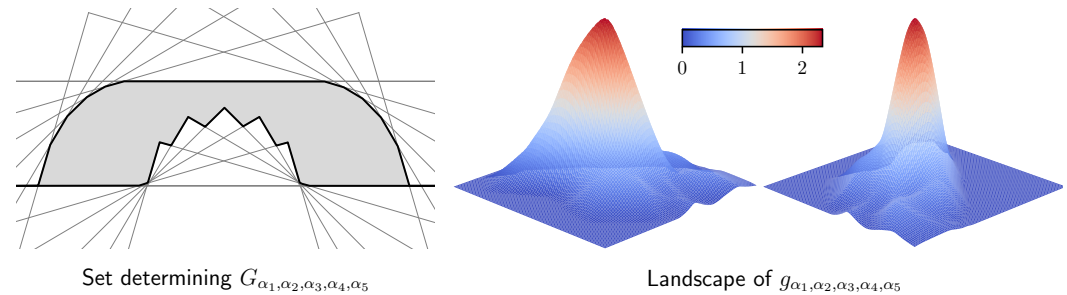


Figure 10. Set determining $G_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5}$ (≈ 2.3337) and landscape of $g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5}$. The landscape uses the rotation centers (u_1, u_2, \dots, u_5) (instead of the network weights) as the high-dimensional variable for directional perturbation, based on the discrete nature of the Kallus–Romik upper bound (namely, the weights are less relevant since the α -sequence is prescribed). The two surface plots show the same peak from different viewpoints, with a range of $[-1, 1] \times [-1, 1]$ centered at $G_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5}$ (the summit). There are no other peaks in the landscape across $[-10, 10] \times [-10, 10]$.

5. Discussion

We have provided two lines of evidence supporting Gerver’s conjecture that his 18-section sofa, with an area of approximately 2.2195, is the unique shape of maximum area that can navigate through a unit-width L -shaped corridor. First, we optimized the sofa area using PINNs, focusing on minimizing inductive bias by ensuring generality in the function space and diversity in function initialization. After training more than 6000 models, over 90% converged on Gerver’s solution. Second, we refined the Kallus–Romik upper bound [6], achieving a tighter upper bound of around 2.3337 compared to their previous estimate of 2.37. By increasing the number of rotation angles to 10,000, our training results showed that the upper bound asymptotically approaches Gerver’s area, further supporting the conclusion that no larger sofa exists.

If a larger sofa does exist, how has it remained undiscovered for nearly 60 years? We propose two plausible explanations in light of all previous attempts, including ours. First, the shape may be fractal or include a fractal section. If so, its discovery would be highly challenging, as we still struggle to estimate areas of known fractals. Second, the global maximum might have such a small catchment basin that it is almost undetectable by optimization-based methods, whether gradient-based or non-gradient-based. In either case, any parameterized function space such as neural networks, would be insufficient. Instead, geometric insights revealing a new movement pattern with contact mechanisms distinct from Gerver’s (such as those described in [4]) would be more critical than further algorithmic advancements.

Author Contributions: Conceptualization, K.L. and J.T.; methodology, K.L., J.B., J.C. and S.P.; software, K.L. and J.C.; validation, J.B. and S.P.; formal analysis, K.L. and J.B.; writing—original draft preparation, K.L. and S.P.; writing—review and editing, K.L., J.B., J.C., S.P. and J.T.; visualization, K.L. and J.C.; funding acquisition, J.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the EPSRC grant, Blueprinting for AI for Science at Exascale (BASE-II, EP/X019918/1).

Data Availability Statement: Our code and experiments are available open-source at <https://github.com/kuangdai/sofa> (accessed on 15 October 2024). The experiments can be reproduced without HPC resources by reducing the number of seeds. All figures presented in this paper can be fully reproduced using our repository.

Acknowledgments: Computations are performed on SCARE, an HPC cluster run by Scientific Computing Department, STFC.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Fully Connected Networks

Appendix A.1. Architecture

Fully connected networks (FCNs) are the backbone of deep learning, forming the most fundamental building blocks of neural network architectures. In this paper, we use FCNs to model univariate functions, $\mathcal{F} : \mathbb{R} \rightarrow \mathbb{R}$, where the input is either the pseudo-time t (as discussed in Section 2) or the rotation angle α (as described in Section 2.3). The output represents one of the degrees of freedom in the movement of the corridor. For simplicity, we demonstrate a two-hidden-layer architecture with a fixed width M , formulated as

$$g(t) = \mathbf{w}_3^T \sigma(\mathbf{W}_2 \sigma(t\mathbf{w}_1 + \mathbf{b}_1) + \mathbf{b}_2) + b_3, \quad (\text{A1})$$

where the learnable parameters include $\mathbf{w}_1, \mathbf{w}_3, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^M$, $\mathbf{W}_2 \in \mathbb{R}^{M \times M}$, and $b_3 \in \mathbb{R}$, and σ denotes the ReLU activation function (applied entry-wise), $\sigma(x) = \max(0, x)$. Such ReLU-activated FCNs form a piecewise linear function space with a vanished Hessian ($g''(t) \equiv 0$) [26].

Appendix A.2. Universal Approximation

The universal approximation properties of FCNs enable them to capture complex patterns in target functions, which is crucial for the success and versatility of deep learning. Early universal approximation theorems [34,35] focused on networks with a single hidden layer, while more recent studies have extended such theorems to deeper architectures [36,37], including those applied to physics-informed machine learning [24].

For our two-layer case in Equation (A1), the following universal approximation theorem can be proved based on either [36] or [37]:

Theorem A1. *Given any continuous function $f : \mathbb{R} \rightarrow \mathbb{R}$ defined on a compact subset of \mathbb{R} , and any $\epsilon > 0$, there exists a two-hidden-layer FCN of the form given in Equation (A1), with its width M being sufficiently large and its weights and biases being appropriately chosen, such that*

$$|g(x) - f(x)| < \epsilon \quad (\text{A2})$$

for all x in the compact subset.

Appendix A.3. Invexity

The notion of *invex functions* was first introduced in [38,39] with the main aim of generalizing the concept of convexity while retaining beneficial analytical properties. Specifically, invex functions exhibit the remarkable property that their *critical points coincide with the global minima* [27]. As such, convex functions are a special case of invex functions. One of the most significant findings has been that invexity enables zero duality gap for certain optimization problems [38], thereby matching an excellent attribute of convex optimization. Motivated by the elegant properties of invex functions and their applicability for solving complex optimization problems efficiently and robustly, we name a few works from the state of the art [40–42]. Below, we present the notion of invexity, along with few needed concepts, starting with the definition of a locally Lipschitz continuous function.

Definition A1. *A function $f : \mathbb{R}^M \rightarrow \mathbb{R}$ is locally Lipschitz continuous at a point $\mathbf{x} \in \mathbb{R}^M$ if there exist scalars $K > 0$ and $\epsilon > 0$ such that for all $\mathbf{y}, \mathbf{z} \in B(\mathbf{x}, \epsilon)$ we have*

$$|f(\mathbf{y}) - f(\mathbf{z})| \leq K \|\mathbf{y} - \mathbf{z}\|_2,$$

where $B(\mathbf{x}; r) = \{\mathbf{w} \in \mathbb{R}^M : \|\mathbf{w} - \mathbf{x}\|_2 < r\}$.

Based on this, the concept of invex function is presented as follows.

Definition A2. Let $f : \mathbb{R}^M \rightarrow \mathbb{R}$ be locally Lipschitz; then f is invex if there exists a function $\eta : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^M$ such that

$$f(\mathbf{x}) - f(\mathbf{y}) \geq \nabla f(\mathbf{y})^T \eta(\mathbf{x}, \mathbf{y}),$$

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^M.$$

By using the recent invexity results in [18], here we mathematically show that our proposed network parameterization as in Equation (A1) are invex. To this end, we remark that Equation (A1) can be equivalently, for analysis, seen as

$$g(\mathbf{x}) = \mathbf{w}_3^T \sigma(\mathbf{W}_2 \sigma(\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + b_3, \quad (\text{A3})$$

because the product $t\mathbf{w}_1$ in Equation (A1) can be encapsulated as a global vector input $\mathbf{x} \in \mathbb{R}^M$ since t is variable and \mathbf{w}_1 in Equation (A1) is a learnable parameter. We remark that proving (A3) is invex plays a crucial role in addressing the problem discussed in this paper. It introduces a parameterization for $x_p(t)$, $y_p(t)$, and $\alpha(t)$ through neural networks, which ensures that any critical point corresponds to a global optimum (invexity definition [27]), a necessary condition for global optimality of Gerver's sofa. This result is summarized in the following theorem.

Theorem A2. Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be any strictly increasing smooth function. Then the neural network parameterization $g : \mathbb{R}^M \rightarrow \mathbb{R}$, of the form

$$g(\mathbf{x}) = \mathbf{w}_3^T \sigma(\mathbf{W}_2 \sigma(\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + b_3, \quad (\text{A4})$$

is invex, when $\mathbf{W}_2 \in \mathbb{R}^{M \times M}$ is full row-rank with M the width of the neural network, $\mathbf{w}_3, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^M, b_3 \in \mathbb{R}$ and $\mathbf{w}_3 \neq \mathbf{0}$.

Proof. To prove the above theorem we analyze the gradient of $g(\mathbf{x})$. Specifically, observe that $\nabla g(\mathbf{x})$ is given as

$$\nabla g(\mathbf{x}) = \left(\mathbf{W}_2^T (\mathbf{w}_3 \odot \sigma'(\mathbf{b}_2 + \mathbf{W}_2 \sigma(\mathbf{x} + \mathbf{b}_1))) \right) \odot \sigma'(\mathbf{x} + \mathbf{b}_1). \quad (\text{A5})$$

From hypothesis we know that \mathbf{W}_2 is a full row-rank matrix, and the learnable parameter \mathbf{w}_3 satisfies $\mathbf{w}_3 \neq \mathbf{0}$. This implies that $\nabla g(\mathbf{x}) \neq \mathbf{0}$, because $\sigma' > 0$ since $\sigma(\cdot)$ was assumed to be strictly increasing smooth function. Therefore, by appealing to [27] [Page 14] $g(\mathbf{x})$ is invex. Thus the result holds. \square

References

1. Moser, L. Problem 66-11, Moving furniture through a hallway. *SIAM Rev.* **1966**, *8*, 381. [[CrossRef](#)]
2. Hammersley, J.M. On the enfeeblement of mathematical skills by modern mathematics and by similar soft intellectual trash in schools and universities. *Educ. Stud. Math.* **1968**, *1*, 17. [[CrossRef](#)]
3. Gerver, J.L. On moving a sofa around a corner. *Geom. Dedicata* **1992**, *42*, 267–283. [[CrossRef](#)]
4. Romik, D. Differential equations and exact solutions in the moving sofa problem. *Exp. Math.* **2018**, *27*, 316–330. [[CrossRef](#)]
5. Batsch, M. A numerical approach for analysing the moving sofa problem. *Symmetry* **2022**, *14*, 1409. [[CrossRef](#)]
6. Kallus, Y.; Romik, D. Improved upper bounds in the moving sofa problem. *Adv. Math.* **2018**, *340*, 960–982. [[CrossRef](#)]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NE, USA, 3–6 December 2012; Curran Associates, Inc.: New York, NY, USA, 2012; Volume 25, pp. 1097–1105.
8. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30, pp. 5998–6008.
9. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med.* **2019**, *25*, 24–29. [[CrossRef](#)]
10. Chen, Y.; Weng, Q.; Tang, L.; Wang, L.; Xing, H.; Liu, Q. Developing an intelligent cloud attention network to support global urban green spaces mapping. *ISPRS J. Photogramm. Remote Sens.* **2023**, *198*, 197–209. [[CrossRef](#)]

11. Fawzi, A.; Balog, M.; Huang, A.; Hubert, T.; Romera-Paredes, B.; Barekatin, M.; Novikov, A.; R Ruiz, F.J.; Schrittwieser, J.; Swirszcz, G.; et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature* **2022**, *610*, 47–53. [CrossRef]
12. Romera-Paredes, B.; Barekatin, M.; Novikov, A.; Balog, M.; Kumar, M.P.; Dupont, E.; Ruiz, F.J.; Ellenberg, J.S.; Wang, P.; Fawzi, O.; et al. Mathematical discoveries from program search with large language models. *Nature* **2024**, *625*, 468–475. [CrossRef]
13. Gukov, S.; Halverson, J.; Manolescu, C.; Ruehle, F. Searching for ribbons with machine learning. *arXiv* **2023**, arXiv:2304.09304.
14. Dissanayake, M.; Phan-Thien, N. Neural-network-based approximations for solving partial differential equations. *Commun. Numer. Methods Eng.* **1994**, *10*, 195–201. [CrossRef]
15. Lagaris, I.E.; Likas, A.; Fotiadis, D.I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* **1998**, *9*, 987–1000. [CrossRef] [PubMed]
16. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
17. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [CrossRef]
18. Pinilla, S.; Thiyaalingam, J. Global Optimality for Non-linear Constrained Restoration Problems via Invexity. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
19. Pinilla, S.; Mu, T.; Bourne, N.; Thiyaalingam, J. Improved imaging by invex regularizers with global optima guarantees. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 10780–10794.
20. Esporeto. Moving Sofa. Online Math Tools, GeoGebra. 2016. Available online: <https://www.geogebra.org/m/vemEtGyj> (accessed on 15 May 2016).
21. Vincent, L.; Soille, P. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 583–598. [CrossRef]
22. De Bock, J.; De Smet, P.; Philips, W. A fast sequential rainfalling watershed segmentation algorithm. In Proceedings of the Advanced Concepts for Intelligent Vision Systems: 7th International Conference, ACIVS 2005, Antwerp, Belgium, 20–23 September 2005; Proceedings 7; Springer: Berlin/Heidelberg, Germany, 2005; pp. 476–482.
23. Park, J.; Sandberg, I.W. Universal approximation using radial-basis-function networks. *Neural Comput.* **1991**, *3*, 246–257. [CrossRef]
24. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. [CrossRef]
25. Tao, Q.; Li, L.; Huang, X.; Xi, X.; Wang, S.; Suykens, J.A. Piecewise linear neural networks and deep learning. *Nat. Rev. Methods Prim.* **2022**, *2*, 42. [CrossRef]
26. Leng, K.; Thiyaalingam, J. On the compatibility between neural networks and partial differential equations for physics-informed learning. *arXiv* **2022**, arXiv:2212.00270.
27. Mishra, S.K.; Giorgi, G. *Invexity and Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008; Volume 88.
28. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015.
29. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
30. Post, J.V. Sequence A128463 in the On-Line Encyclopedia of Integer Sequences. 2007. Available online: <https://oeis.org/A128463> (accessed on 5 May 2007).
31. Liu, D.C.; Nocedal, J. On the limited memory BFGS method for large scale optimization. *Math. Program.* **1989**, *45*, 503–528. [CrossRef]
32. Li, H.; Xu, Z.; Taylor, G.; Studer, C.; Goldstein, T. Visualizing the loss landscape of neural nets. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 6391–6401.
33. Böttcher, L.; Wheeler, G. Visualizing high-dimensional loss landscapes with Hessian directions. *J. Stat. Mech. Theory Exp.* **2024**, *2024*, 023401. [CrossRef]
34. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
35. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251–257. [CrossRef]
36. Maiorov, V.; Pinkus, A. Lower bounds for approximation by MLP neural networks. *Neurocomputing* **1999**, *25*, 81–91. [CrossRef]
37. Lu, Z.; Pu, H.; Wang, F.; Hu, Z.; Wang, L. The expressive power of neural networks: A view from the width. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6232–6240.
38. Hanson, M.A. On sufficiency of the kuhn-tucker conditions. *J. Math. Anal. Appl.* **1981**, *80*, 545–550. [CrossRef]
39. Craven, B.D.; Glover, B.M. Invex functions and duality. *J. Aust. Math. Soc.* **1985**, *39*, 1–20. [CrossRef]
40. Barik, A.; Honorio, J. Fair sparse regression with clustering: An invex relaxation for a combinatorial problem. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 23245–23257.
41. Syed, M.; Pardalos, P.; Principe, J. Invexity of the minimum error entropy criterion. *IEEE Signal Process. Lett.* **2013**, *20*, 1159–1162. [CrossRef]
42. Chen, B.; Xing, L.; Zhao, H.; Zheng, N.; Pri, J.C. Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **2016**, *64*, 3376–3387. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.