Published for SISSA by Springer

Received: March 12, 2024
Accepted: April 23, 2024
Published: May 16, 2024

Reconstructing S-matrix Phases with Machine Learning

Aurélien Dersy[a,b] **Matthew D. Schwartz**[a,b] **and Alexander Zhiboedov**[c]

[a] *Department of Physics, Harvard University, 02138 Cambridge, MA, U.S.A.*
[b] *NSF Institute for Artificial Intelligence and Fundamental Interactions, Cambridge, MA 02139, U.S.A.*
[c] *CERN, Theoretical Physics Department, CH-1211 Geneva 23, Switzerland*

E-mail: adersy@g.harvard.edu, schwartz@g.harvard.edu, alexander.zhiboedov@cern.ch

Abstract: An important element of the S-matrix bootstrap program is the relationship between the modulus of an S-matrix element and its phase. Unitarity relates them by an integral equation. Even in the simplest case of elastic scattering, this integral equation cannot be solved analytically and numerical approaches are required. We apply modern machine learning techniques to studying the unitarity constraint. We find that for a given modulus, when a phase exists it can generally be reconstructed to good accuracy with machine learning. Moreover, the loss of the reconstruction algorithm provides a good proxy for whether a given modulus can be consistent with unitarity at all. In addition, we study the question of whether multiple phases can be consistent with a single modulus, finding novel phase-ambiguous solutions. In particular, we find a new phase-ambiguous solution which pushes the known limit on such solutions significantly beyond the previous bound.

Keywords: Nonperturbative Effects, Scattering Amplitudes

ArXiv ePrint: 2308.09451

https://doi.org/10.1007/JHEP05(2024)200

Contents

1 Introduction **1**

2 Machine Learning implementation **5**
2.1 Implementation details 6

3 Single phase determination **7**
3.1 Warmup: simple examples 8
3.2 Scanning the loss landscape 9
3.3 Extremal amplitudes 14

4 Phase ambiguities: finite partial waves **16**
4.1 Classical solutions 16
4.2 Machine learning with repulsive loss 18

5 Infinite partial wave ambiguities **21**
5.1 Classical solutions 22
5.2 Machine learning complex functions 25
5.3 Resolving the z_1 landscape with ML 26
5.4 Extensions beyond one zero 30

6 Conclusions **33**

A Finite partial wave decomposition **35**

B Phase shift ambiguities with finite partial waves **37**

C Scaled and non-scaled losses **38**

D Simple dual bounds **39**

1 Introduction

A crucial ingredient in the S-matrix bootstrap program is the relation between the modulus of an amplitude $F(z)$ and its phase, as constrained by unitarity. An important question is, is it always possible to find a phase given the magnitude of a scattering amplitude, and, if so, is that phase unique? If an algorithm were known to find the phase, then it could be applied to physical data, from the differential cross section, to reconstruct the underlying quantum mechanical amplitude. It is generally believed that the amplitude is uniquely fixed, up to the trivial ambiguity, $F(z) \to -F(z)^\star$, provided one has access to the differential cross section across all energies and all angles [1, 2].[1] In a more realistic setting such information is never

[1]The proof [1] assumes that the external particles are scalars, that the amplitude is symmetric under crossing, and finally that there are no bound states.

JHEP05(2024)200

known, so we can ask how much information about the phase can be deduced from scattering data in a limited range of energies, or even at fixed energy. Even at fixed energy, and even in the elastic scattering regime where only $2 \to 2$ scattering is possible, the problem of finding the amplitude from the differential cross section is a hard one [3–5]. For inelastic scattering with an infinite number of partial waves, there can be continuous families of phases with the same modulus [6–8]; for inelastic scattering with L partial waves there are at most 2^L phases for a given modulus [9]; for elastic scattering, the number of phase-ambiguous solutions is expected to be at most *two* [10]. The question of how to constrain the phase given the cross section has been around since the 1960s, but not much progress has been made since the 1970s. Given the revitalization of the S-matrix bootstrap program [11], partly inspired by modern computational techniques, we propose to revisit some of the questions using modern tools such as machine learning. We focus here on a clear well-defined problem: given a differential cross-section of a scalar $2 \to 2$ scattering process in the elastic region (energy below the first inelastic threshold), under what circumstances does an underlying complex amplitude producing it exists and under what circumstances is the amplitude unique?

We focus on the elastic scattering regime at fixed energy. Since energy is fixed, a $2 \to 2$ amplitude is a function only of the scattering angle θ and we use $z \equiv \cos\theta$ throughout. We write $F(z)$ for the amplitude, $B(z) \equiv |F(z)|$ for its modulus, and $\phi(z)$ for its phase. The partial wave decomposition of the amplitude is

$$F(z) = B(z)e^{i\phi(z)} = \sum_{\ell=0}^{\infty}(2\ell+1)f_\ell P_\ell(z), \tag{1.1}$$

where $P_\ell(z)$ are the standard Legendre polynomials of spin ℓ. In this notation, unitarity requires $\mathrm{Im} f_\ell = |f_\ell|^2$ for all ℓ.[2] Writing the partial waves as $f_\ell = \sin\delta_\ell e^{i\delta_\ell}$, unitarity is equivalent to all the phase shifts δ_ℓ being real. The differential cross section depends only on $|F(z)|^2 = B(z)^2$, so up to trivial kinematic factors the differential cross section and modulus are equivalent.

Although the partial-wave decomposition is general, if there are an infinite number of partial waves it may not be so useful. One can instead phrase the unitarity condition as an integral equation for the modulus $B(z)$ and phase $\phi(z)$ [14, 15]:

$$\sin\phi(z) = \int_{-1}^{1} dz_1 \int_0^{2\pi} d\phi_1 \frac{B(z_1)B(z_2)}{4\pi\, B(z)} \cos\left[\phi(z_1) - \phi(z_2)\right] \tag{1.2}$$

where

$$z_2(z, z_1, \phi_1) \equiv z z_1 + \sqrt{1-z^2}\sqrt{1-z_1^2}\cos\phi_1\,. \tag{1.3}$$

[2]This unitarity relation holds for elastic scattering of non-identical scalar particles $AB \to AB$. The relationship between $F(z)$ and the standard amplitude $\langle p_3, p_4|\hat{T}|p_2, p_1\rangle = (2\pi)^4\delta^4(p_1+p_2-p_3-p_4)T(s,t)$ at fixed s, see e.g. [12, 13], is $F(z) \equiv \frac{1}{16\pi}\sqrt{\frac{s-4m^2}{s}}T\Big(s, -\frac{s-4m^2}{2}(1-z)\Big)$, where in writing this formula we assumed for simplicity that all particles have equal mass m. For scattering of identical particles $AA \to AA$ only even spin partial waves appear in the sum (1.1), so that $F(z) = F(-z)$, and the relationship to the standard scattering amplitude becomes $F(z) \equiv \frac{1}{32\pi}\sqrt{\frac{s-4m^2}{s}}T\Big(s, -\frac{s-4m^2}{2}(1-z)\Big)$.

JHEP05(2024)200

By evaluating eq. (1.2) at $z = 1$ so that $z_2 = z_1$ we immediately get a *necessary* condition on $B(z)$ to be a valid modulus, namely

$$\int_{-1}^{1} dz_1 \frac{B(z_1)^2}{2B(1)} \leq 1. \tag{1.4}$$

This "dual" bound already severely restricts the space of allowable $B(z)$.

Given $B(z)$ it is in general very difficult to solve eq. (1.2) to find $\phi(z)$. Indeed, there are two closely related, but unanswered, questions we can ask

1. For which $B(z)$ is there a solution to eq. (1.2)? That is, which elastic-scattering cross sections can conceivably be realized in a unitary quantum field theory?

2. For which $B(z)$ can there be more than one solution to eq. (1.2)? That is, when is the phase unique?

Both of these questions were studied some time ago and only partially answered, as we now review.

The sharpest statements so far have been made by applying the contraction mapping principle to the unitarity equation, where the search for a phase solution can be recast as a problem of finding the mapping's associated fixed point [14–16]. The current bounds on both existence and uniqueness have been derived based on the integrated form of the kernel in eq. (1.2)

$$K(z) \equiv \int_{-1}^{1} dz_1 \int_{0}^{2\pi} d\phi_1 \, \frac{B(z_1)B(z_2)}{4\pi \, B(z)} \,. \tag{1.5}$$

The maximum of this function was denoted by Martin as

$$\sin\mu \equiv \max_{-1\leq z\leq 1} K(z) \,. \tag{1.6}$$

To motivate this, we note that since $|\cos[\phi(z_1) - \phi(z_2)]| \leq 1$ eq. (1.2) implies

$$|\sin\phi(z)| \leq \sin\mu \,. \tag{1.7}$$

If the phase $\phi(z)$ is constant then by eq. (1.2) $B(z)$ must be constant as well and $\sin\phi = \sin\mu = B$, so this bound is saturated. Furthermore, since ϕ must be real we have that for constant phases, $\sin\mu \leq 1$ and $B \leq 1$. It has also been proven that as long as $\sin\mu \leq 1$ for any given $B(z)$, a corresponding phase always exists. The proof treats the eq. (1.2) as a non-linear operation $\phi_{n+1} = O(\phi_n)$, and applies the Leray-Schauder principle to argue for the existence of a fixed point [3]. We discuss this approach more in section 3.1. Conversely, there exist differential cross sections with $\sin\mu > 1$ for which no phase exists, the simplest example being constant $B > 1$. So one cannot hope to push this sufficient criterion for the existence of a phase further. If $\sin\mu > 1$ there is no known test to determine whether or not a phase exists for a given $B(z)$ beyond (1.4).

Regarding the question of uniqueness, the contraction mapping principle was applied to demonstrate that any solution with $\sin\mu < \sqrt{\frac{\sqrt{5}-1}{2}} \approx 0.79$ is unique [3, 14], while further refinements [17] pushed the bound up to $\sin\mu < 0.86$. For polynomial amplitudes (finite

JHEP05(2024)200

number of partial waves) $\sin\mu \leq 1$ is enough to ensure both existence and uniqueness [3]. For polynomial amplitudes, it has also been shown that if the average modulus $\sigma_T = \frac{1}{2}\int_{-1}^{1} dz B^2(z)$ satisfies $\sigma_T < 1.38$ then uniqueness is guaranteed. For general amplitudes with an infinite number of partial waves, it has been conjectured [3, 15] but not proven or disproven that uniqueness should still hold if $\sin\mu < 1$. In the elastic scattering region that we consider, any nontrivial (i.e. excluding $F(z) \to -F(z)^\star$) phase ambiguity is expected to be twofold at most, as has been proven for genuine entire functions (i.e not a polynomial) [18], see also [10].

A modulus with two corresponding phases was found by Crichton in 1966 [19]. Crichton's solution has only the $L = 2$ partial waves and $\sin\mu = 3.2$. Shortly after, the complete set of $L = 2$ phase ambiguous solutions was characterized [20]. The lowest value of $\sin\mu$ among these was 2.6. Solutions with $L = 3$ and $L = 4$ have also been studied [21, 22]. These solutions are discussed in section 4.

Phase-ambiguous solutions have also been found with an infinite number of partial waves in [23]. The lowest published value of $\sin\mu$ among these, to our knowledge, is $\sin\mu \approx 2.15$. These results are reviewed in more detail in section 5. Applying modern numerical methods we are able to find a phase-ambiguous solution with $\sin\mu \approx 1.67$.

In this paper, we revisit some of these old questions about phase determination in the elastic regime using modern numerical methods and machine learning. Recent advances in machine learning have given rise to a multitude of applications in physics, from jet tagging algorithms [24], to fast detector simulators [25] or AI-driven symbolic regression [26, 27] and give us the perfect tool for tackling hard numerical problems for which classical algorithms are challenging to design. In particular, machine learning has already been useful for studying some of the structure of the S-matrix relevant to hadron physics. There, classifiers have been trained to predict the pole structure of scattering amplitudes and identify the associated physical states [28–30]. It is well known that neural networks are universal function approximators [31] and as such are ideal candidates for solving integro-differential equations. In particular Physics-informed neural networks have been shown to be able to resolve multi-dimensional differential equations [32], where they act as a functional ansatz and have a loss function given by the differential equation of interest. The extension to integral equations usually involves a discretization scheme for the actual integral and has been studied and implemented in various libraries [33–36]. In the following, we will explore how similar techniques can be applied to study and solve the unitarity integral equation eq. (1.2), demonstrating how to numerically recover the phase corresponding to a given input differential cross section. We will highlight the interesting duality between the convergence properties of the machine learning algorithm and the kernel function, making the link with bounds derived in the literature. Finally, we will deploy various neural networks and impose a repulsive loss in order to probe the uniqueness of the recovered solution.

We begin in section 2 by describing the machine learning setup and approach we take to establishing consistency between a modulus and a phase. The unitarity constraint is encoded in eq. (1.2). There are different ways to solve this equation. Given a known modulus $B(z)$, for example from experimental cross-section data or some S-matrix-bootstrap computation, one can then search for a phase $\phi(z)$ consistent with unitarity. To find moduli with phase ambiguities, one can alternatively search for 3 functions $B(z)$, $\phi_1(z)$ and $\phi_2(z)$

JHEP05(2024)200

all consistent with unitarity. To guarantee that $\phi_1(z)$ and $\phi_2(z)$ are not trivially equivalent (e.g. by $\phi_1(z) = \pi - \phi_2(z)$), we will also need to add a repulsive loss to keep the solutions apart. Section 3 discusses how to use our machine learning set-up to find $\phi(z)$ given $B(z)$. Section 4 discusses the case of phase-ambiguous solutions for amplitudes with a finite number of partial waves and section 5 the infinite partial-wave case. By exploring this space through a combination of a machine learning search and a refinement using a classical algorithm capable of high precision, we find a large number of new phase-ambiguous solutions. The lowest $\sin\mu$ among these is $\sin\mu \approx 1.67$, a value significantly closer to $\sin\mu = 1$ than the best previously known example $\sin\mu \approx 2.15$. A summary and conclusions are in section 6.

2 Machine Learning implementation

Finding a unitary amplitude for a given input differential cross section boils down to solving the eq. (1.2). Solving differential or integral equations with machine learning is a problem that has already been tackled efficiently in the literature, through the use of Physics-Informed Neural Networks (PINNs) [32]. In a PINN the idea is to use a neural network $u_\sigma(\vec{x})$ as a surrogate of the solution $u(\vec{x})$. Here u_σ is a neural network that takes as input the data $\vec{x}$ and has parameters (weights and biases) described by σ. The precise architecture of u_σ can be fine-tuned given the problem at hand but typical setups consider simple feed-forward neural networks. Having the neural network ansatz allows one to take derivatives efficiently with respect to the inputs, a desirable property for solving differential equations. Indeed the loss function for PINNs is usually taken to be the differential equation itself, evaluated at a set of collocation points.

In our problem, we have a few notable particularities that depart from the typical PINN use case:

1. We are solving an integral equation as opposed to a differential equation. In practice, this is done by approximating the integral, for instance with Gaussian quadrature or a trapezoidal rule, which will inevitably lead to some numerical errors.

2. We will be interested in understanding whether the phase solution is unique. Probing this property could be done by training different neural networks that are initialized with different random seeds. Another approach, which is one that we will prefer, is to add a repulsion term in the loss function. This allows us to simultaneously train different neural networks, each corresponding to a distinct solution of the integral equation.

3. We are not always guaranteed to find a solution for any given $B(z)$, so our networks are not always expected to converge to low loss values. This will lead us to study the loss landscape in more detail for simple input differential cross sections.

4. We are interested in parsing through the space of differential cross sections to probe existence and uniqueness criteria. When explicitly looking for ambiguous solutions we will see that after parameterizing the amplitude in some simple way we are able to learn both the phase $\phi(z)$ and the modulus $B(z)$.

JHEP05(2024)200

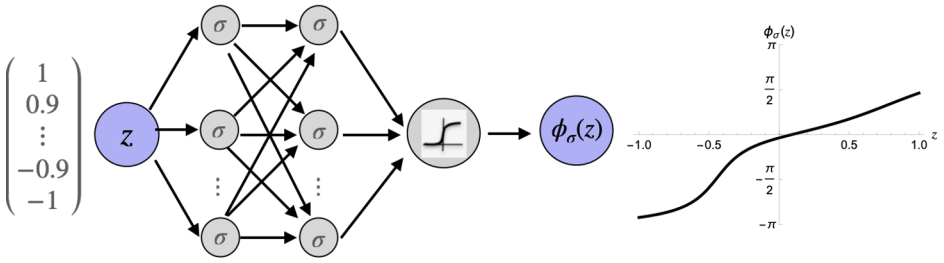


Figure 1. We utilize a neural network ansatz for parametrizing the phase solution. The feedforward neural network has a series of layers with learnable parameters σ ending with a final *Tanh* activation function, constraining the outputs to lie within the range $[-\pi, \pi]$.

Similarly to PINNs however we will parametrize the phase $\phi(z)$ by a neural network ϕ_σ and ask for it to solve the eq. (1.2). The parameterized phase $\phi_\sigma(z)$ shown in figure (1) is a network that takes in a single input and depends on a set of neural network parameters σ. These parameters are to be updated and optimized to satisfy a given objective or loss function, typically given by the unitarity integral equation. Contrarily to PINNs, we will not have any specific boundary condition to satisfy, rather we will force the output of $\phi_\sigma(z)$ to lie within the range $[-\pi, \pi]$. This is done by adding a scaled sigmoid or tanh activation function at the end of the network. Since we are interested in solving a formal equation we are free to take any z point as part of our training data, provided $z \in [-1, 1]$.

2.1 Implementation details

Following the discussion of the previous section we parametrize $\phi(z)$ by a network with a simple feed-forward architecture, which we implement with *PyTorch* [37]. We will restrict ourselves to small architectures, typically 4 layers with 64 nodes each using Rectified Linear Unit (ReLU) activation functions. The outputs are constrained in the $[-\pi, \pi]$ range by adding a scaled hyperbolic tangent function after the final layer.[3] The loss function is taken to be the Mean Squared Error (MSE) of the integral equation, averaged over a set of N_c randomly sampled collocation points, namely:

$$\mathcal{L}_E = \mathbb{E}_z \left|\left| B(z)\sin\phi(z) - \frac{1}{4\pi}\int_{-1}^{1} dz_1 \int_0^{2\pi} d\phi_1 B(z_1)B(z_2)\cos\left(\phi(z_1) - \phi(z_2)\right)\right|\right|^2 . \tag{2.1}$$

In practice, the expectation value $\mathbb{E}_z$ appearing in the loss is estimated by averaging over the set of collocation points $\{z_c\}$ as $\mathbb{E}_z \sim N_c^{-1}\sum_{z\in\{z_c\}}$. The two-dimensional integral is also estimated, discretizing the integrand over a grid in (z_1, ϕ_1) space. For training, we will also consider a scaled version of this loss defined as

$$\mathcal{L}_E^S = \mathbb{E}_z \left|\left| \sin\phi(z) - \frac{1}{4\pi B(z)}\int_{-1}^{1} dz_1 \int_0^{2\pi} d\phi_1 B(z_1)B(z_2)\cos\left(\phi(z_1) - \phi(z_2)\right)\right|\right|^2 \tag{2.2}$$

which has the desirable feature of having terms of order 1. When learning $B(z)$ this loss will also discourage the network from learning arbitrarily small moduli. One could be tempted

[3] If we are in the $\sin\mu < 1$ regime, where existence is guaranteed, we can further restrict the range to $[-\pi/2, \pi/2]$ in order to eliminate the trivial ambiguity relating $\phi(z) \to \pi - \phi(z)$.

JHEP05(2024)200

Parameter class	Parameter type	Value
Architecture	Activation	ReLU
	Layers	[64,64,64,64]
	Final layer	$\pi \tanh(z)$
Optimizer	Optimizer name	Adam
	β_1	0.9
	β_2	0.999
	Learning rate	$3 \cdot 10^{-3}$
	Scheduler	MultiplicativeLR with 0.999 decay
Training loss	Batch size	64
	Integral approximator	Trapezoidal rule
	Integral sampling points	25×25

Table 1. Model architecture and hyperparameters used for the parameterization of the phase $\phi_\sigma(z)$ with a single neural network.

to further normalize and divide the loss function by $\sin\phi(z)$, but this leads to numerical instabilities if the expected phase value nears 0.

During training the expectation value in the loss function is approximated by averaging over a batch of 64 randomly sampled $\{z_c\}$ collocation points. Random sampling ensures that the entire angle range is properly resolved and not overfitted. The network parameters are updated at the end of each epoch, defined here by the complete processing of a single batch. The parameter update is done via the Adam optimizer [38] where we set the coefficients $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to their default values. These coefficients correspond to the exponential decay rates of respectively the first and second moment estimates of the gradient. In order to calculate the loss function and the relevant two-dimensional integrals we have to resort to a numerical approximation. We use the trapezoidal rule,[4] where at each fixed z value we pick out 25×25 reference points, linearly spaced out in the z_1 and ϕ_1 directions, giving us an evaluation grid for approximating the two-dimensional integral. In general the trapezoidal rule for a function $f(x)$ will give a numerical error scaling as KN^{-2}, where N is the number of points picked along a single direction and $|f''(x)| < K$. In our problem of interest, with our choice of points, we expect the trapezoidal rule to give errors in the range of $10^{-4} - 10^{-6}$ at different z values. This implies that as $\mathcal{L}_E \sim 10^{-8}$ the loss becomes of the order of the numerical precision that we operate at. Our default model choice and hyperparameters are summarized in table 1 and any deviation from those in the numerical experiments will be mentioned explicitly. Different choices of hyperparameters have been considered but those listed ended up giving the best performance after a brief optimization search.

3 Single phase determination

We start our analysis by probing the question of existence of $\phi(z)$ given $B(z)$. That is, we will be interested in training a single neural network to recover a phase $\phi(z)$, which is a

[4]The trapezoidal rule is implemented in *PyTorch* directly which ensures that we will end up with a final loss that is fully differentiable and supports backpropagation.

JHEP05(2024)200

solution to eq. (1.2), assuming that the modulus $B(z)$ is a known function. We will start by verifying that the network can be trained to recover solutions in the regime where $\sin\mu < 1$, where we have guarantees on the existence of the function. Focusing on simple polynomial differential cross sections, we will illustrate that the loss landscape is sensitive to the value of $\sin\mu$ and that the existence bounds are respected. We will also demonstrate that our method is able to recover solutions when $\sin\mu > 1$, taking examples where the amplitudes are parameterized by an either finite or infinite partial wave decomposition.

3.1 Warmup: simple examples

To get started, we first consider simple polynomial forms for the modulus. We consider a linear function $B(z) = (z+4)/10$ and a quadratic function $B(z) = (z^2+1)/2$. Both moduli are positive across the z range and have $\sin\mu$ values that are respectively $\sin\mu_1 = \frac{47}{90} \approx 0.522$ and $\sin\mu_2 = \frac{13}{15} \approx 0.867$, guarantying the existence of a solution. Their integrated kernels $K(z)$ (cf. eq. (1.5)) whose maximum gives $\sin\mu$ are shown in figure 2. We implement different neural networks following the setup described in section 2 and let them run for 5000 epochs. The final performance is evaluated on a test set of 100 linearly spaced out z points. We show at the bottom of figure 2 the predicted phases for both cases considered. The final evaluation losses are both of the order of $\mathcal{L}_E^S \sim 10^{-8}$, thus at the order of the numerical precision which is supported by the numerical integration scheme. For moduli satisfying $\sin\mu < 1$ such as these, the numerical fixed point iteration [15] applies and we have verified with a classical algorithm that the solutions agree with the ones found by our framework.

In figure 2, we can observe that for both the linear and quadratic cases, the phases $\phi(z)$ look a lot like the integrated kernel $K(z)$. This is straightforward to understand. When $\phi(z) \ll 1$, one can expand the unitarity equation eq. (1.2) using $\sin\phi(z) \approx \phi(z)$ and $\cos[\phi(z_1) - \phi(z_2)] \approx 1$ to see that $\phi(z) = K(z)$ to first order in $\phi(z)$. Indeed, one can then expand to second order in $\phi(z)$ giving

$$\phi(z) = K(z) + \int_{-1}^{1} dz_1 \int_0^{2\pi} d\phi_1 \frac{B(z_1)B(z_2)}{4\pi\, B(z)} \frac{1}{2}[K(z_1) - K(z_2)]^2 + \cdots \tag{3.1}$$

and so on. This is actually just a version of the fixed-point iteration scheme starting with $\phi(z) = 0$. One can in principle use this procedure even if $\phi(z)$ is not small. There the iterative scheme takes $\phi_{n+1} = \Phi(\phi_n)$ where

$$\Phi(\phi_n) = \arcsin\left(\frac{1}{4\pi}\int_{-1}^{1} dz_1 \int_0^{2\pi} d\phi_1 \frac{B(z_1)B(z_2)}{B(z)} \cos[\phi(z_1) - \phi(z_2)]\right) \tag{3.2}$$

and aims at finding the fixed point $\phi^\star = \Phi(\phi^\star)$. However, if $\sin\mu > 1$ then $K(z) > 1$ for some z and $\sin\phi(z) = K(z)$ has no solution for real phases $\phi(z)$. This is one reason we only expect phase ambiguities for $\sin\mu > 1$. Generally, we find that for $\sin\mu < 1$ the iteration tends to converge fairly quickly (although we cannot prove it is independent of the initial condition for the iteration), but when $\sin\mu > 1$ it often does not converge at all.

The unitarity equation also implies the bound $\sin\phi(z) \leq K(z)$, which is respected in our results, giving an important cross-check. For these simple polynomial amplitudes, we can see from figure 2 that $K(z) \approx B(z)^{-1}$. This is once again expected in the regime where the

JHEP05(2024)200

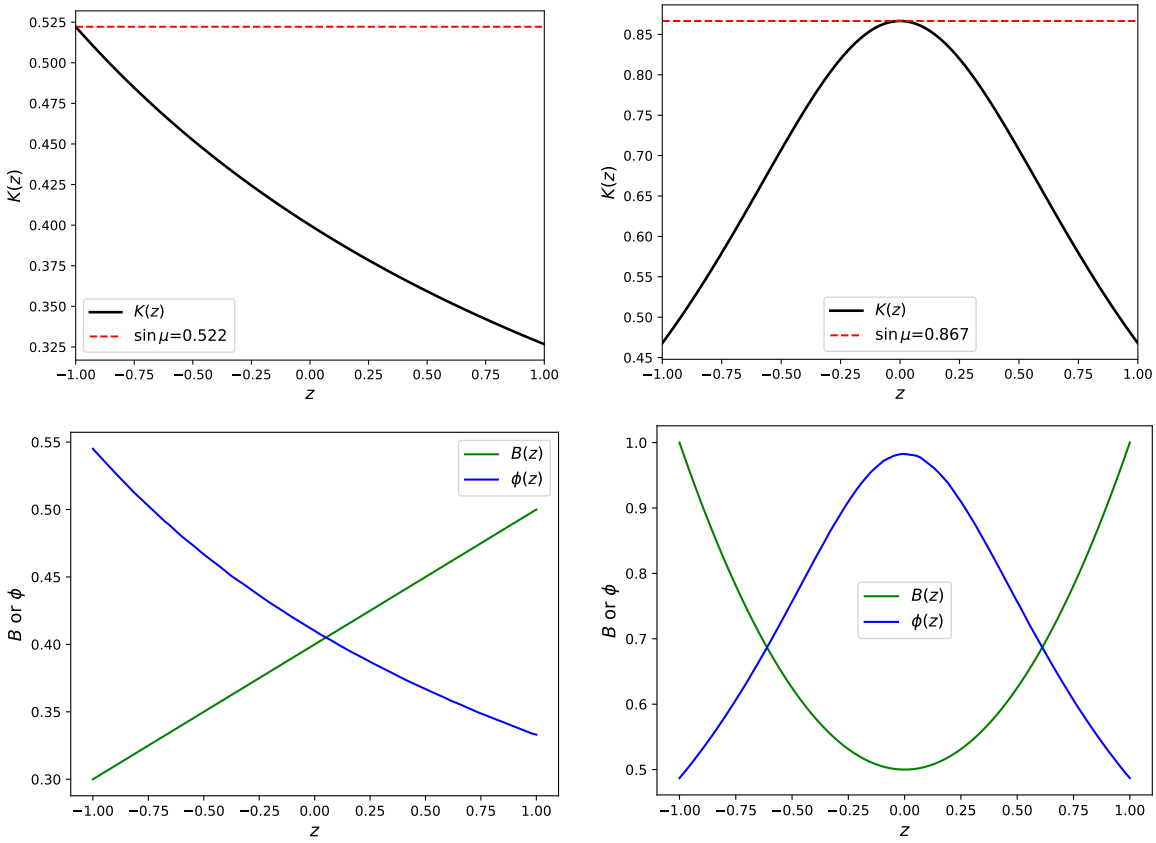


Figure 2. We warm up by determining the phase $\phi(z)$ for a linear modulus $B(z) = \frac{z+4}{10}$ (left) and a quadratic modulus $B(z) = \frac{z^2+1}{2}$ (right). Top panels show the integrated kernel $K(z)$ and its maximum $\sin\mu$. Bottom panels show $B(z)$ and the phase $\phi(z)$ found with machine learning.

integral appearing in eq. (1.5) is slowly varying. In particular, for the linear modulus, we have

$$K(z)B(z) = \int_0^{2\pi} d\phi_1 \int_{-1}^{1} \frac{B(z_1)B(z_2)}{4\pi} = \frac{z+48}{300} \tag{3.3}$$

Since $-1 < z < 1$ this function is essentially constant and $K(z) \approx \frac{c}{B(z)}$ for some c results.

As a side note, when $\phi(z)$ is a solution then so is $\pi - \phi(z)$. These solutions are said to be trivially related and either could have been expected. Changing the initialization seed of the networks is a way to recover such alternative solutions.

3.2 Scanning the loss landscape

Having validated our method on two simple polynomial examples, we can now look into the performance of our implementation on families of $B(z)$. We consider two families: a linear one, $B(z) = az+b$ and a quadratic one $B(z) = cz^2+d$. For each value of a and b or c and d we can search for a phase. Although the network cannot tell us for sure whether unitarity is exactly satisfied, the loss of the neural network provides a good proxy for satisfaction. We thus explore the loss landscape and compare it to other indicators of whether unitarity can be satisfied.

JHEP05(2024)200

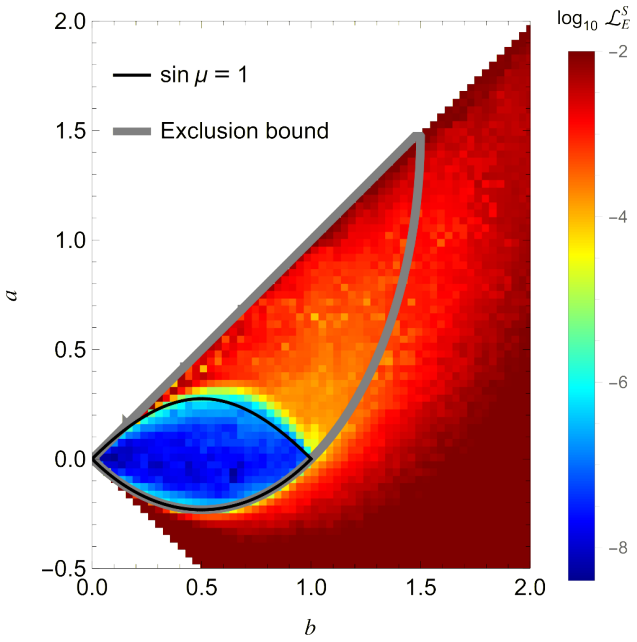


Figure 3. Scaled loss landscape for the two-parameter family of moduli $B(z) = az + b$. For each a and b we find a phase $\phi(z)$. Red regions indicate that no solution is likely. The black curve is $\sin\mu = 1$ and delimits the region within which we are guaranteed the existence of a solution. The grey curve delimits the bound of eq. (1.4) and solutions cannot be found outside of its enclosed region.

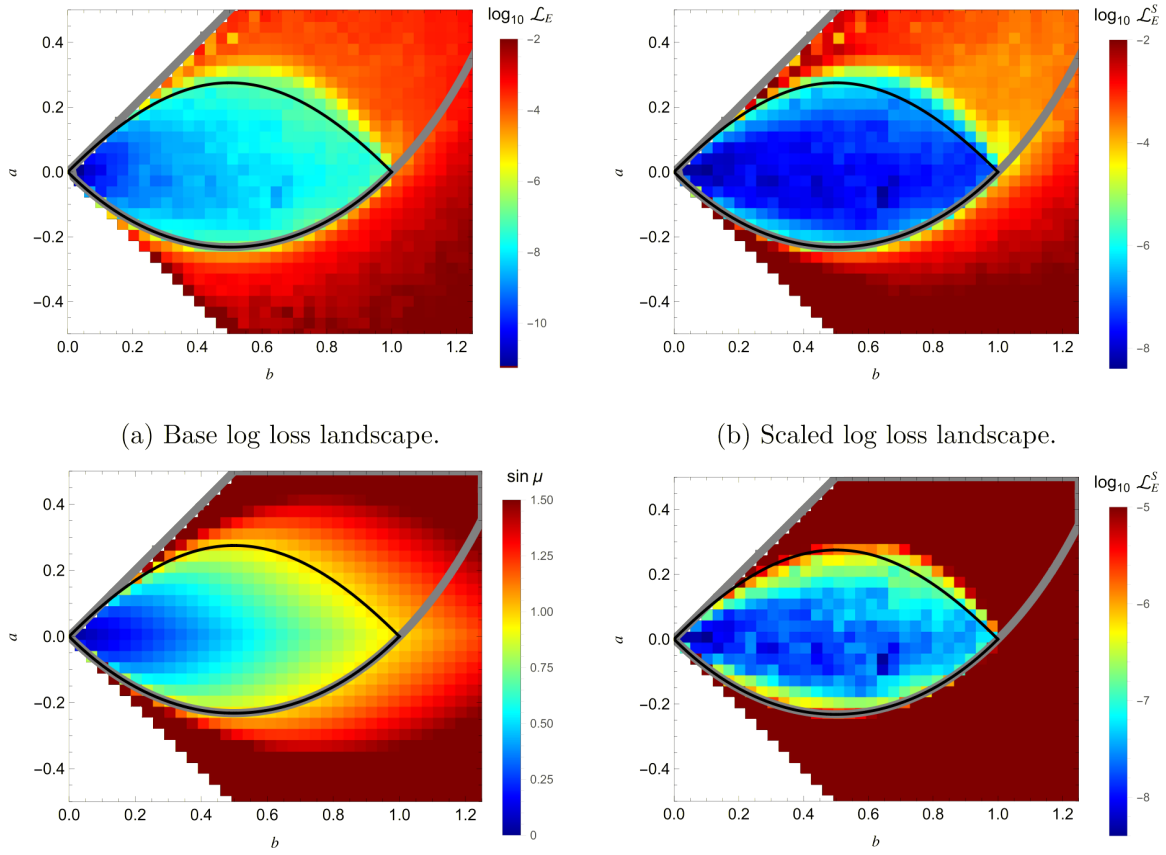
3.2.1 Linear functions

We consider the family $B(z) = az + b$ with a and b real and $b > |a|$, which ensures positivity of $B(z)$ for all z values. Although $B(z)$ is a polynomial, the amplitude $F(z) = B(z)e^{i\phi(z)}$ will generally not be. Indeed, this parameterization for $B(z)$ is not compatible with any unitary polynomial $F(z)$ (see appendix A). As such, any numerical solution for $\phi(z)$ has to be understood as possessing an infinite partial wave decomposition.

We conduct a scan over this family of $B(z)$ by taking a 75×60 grid over different a, b values, with $a \in [-0.5, 2.0]$ and $b \in [0, 2.0]$. For every parameter pair, we train a new neural network using the scaled loss function of eq. (2.2) for 2000 epochs. We then evaluate the base and scaled losses, eqs. (2.1) and (2.2), on the resulting solutions. In figure 3 we show the complete scaled loss landscape, along with the $\sin\mu = 1$ and $\int_{-1}^{1} B(z_1)^2 = 2B(1)$ contours. Those are contours for respectively guaranteeing and excluding solutions. A zoomed-in perspective on the regions of low losses is shown in figure 4. $\sin\mu < 1$ seems to give a good indication that a solution exists or not. This is non-trivial — the ML algorithm knows nothing about $\sin\mu$ and there could equally well have been an entirely different functional of $B(z)$ which characterized the existence of a solution. The correspondence of $\sin\mu = 1$ with the boundary of the allowed region is further explored in the bottom panels. There we also show the values of $\sin\mu$ across this linear $B(z)$ family.

On the bottom right we show that the boundaries of $\sin\mu \sim 1$ and $\mathcal{L}_E^S \sim 10^{-5}$ almost perfectly overlap. As the loss crosses this threshold it rapidly grows by many orders of magnitude, up to $10^{-4} - 10^{-2}$, indicative of a regime where the network is unable to find a

JHEP05(2024)200



(a) Base log loss landscape.

(b) Scaled log loss landscape.

(c) $\sin\mu$ landscape.

(d) Cut on the scaled log loss landscape.

Figure 4. Zoom on the loss landscapes for the two-parameter family of moduli $B(z) = az + b$. Black and grey curves follow figure 3. Top panels show the base loss and scaled loss. Panel (c) shows a heat map of $\sin\mu$ values over the family $B(z) = az + b$ (no phase is determined or needed). Right shows the scaled loss landscape with a hard cut of $\mathcal{L}_E^S \sim 10^{-5}$. This loss boundary agrees very well with the $\sin\mu = 1$ boundary.

solution outside of $\sin\mu > 1$ for linear moduli. In particular, the bottom boundary of $\sin\mu = 1$ overlaps with the dual exclusion bound, across which the network has $\mathcal{L}_E^S > 10^{-5}$ and does not find any solutions. However, near the top $\sin\mu = 1$ boundary, we have a very thin region of potential solutions with $\sin\mu > 1$. All of these solutions fall within the grey region and are not forbidden by the dual exclusion bound. Additional dual bounds constraints are explored in appendix D but do not go beyond the simple exclusion bound we have considered.

3.2.2 Quadratic functions

Next, we consider the family of symmetric quadratic moduli with $B(z) = c + dz^2$. To keep $B(z)$ positive we restrict to $c > |d|$. We proceed in a similar fashion as in the previous section, constructing a grid of 45×180 points for $c \in [0, 1.5]$ and $d \in [-0.5, 5.5]$, where we train a new neural network at each point for 2000 epochs.

In figures 5–6 we repeat the same plots as for the linear function: the loss landscapes for the base and scaled losses along with the $\sin\mu$ values and the $\mathcal{L}_E^S$ cut. Within the $\sin\mu < 1$

JHEP05(2024)200

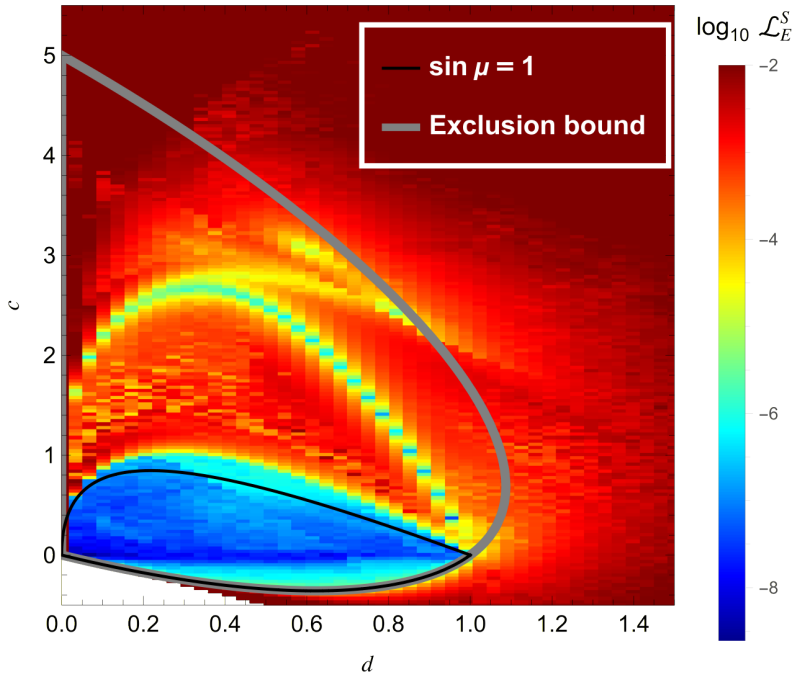


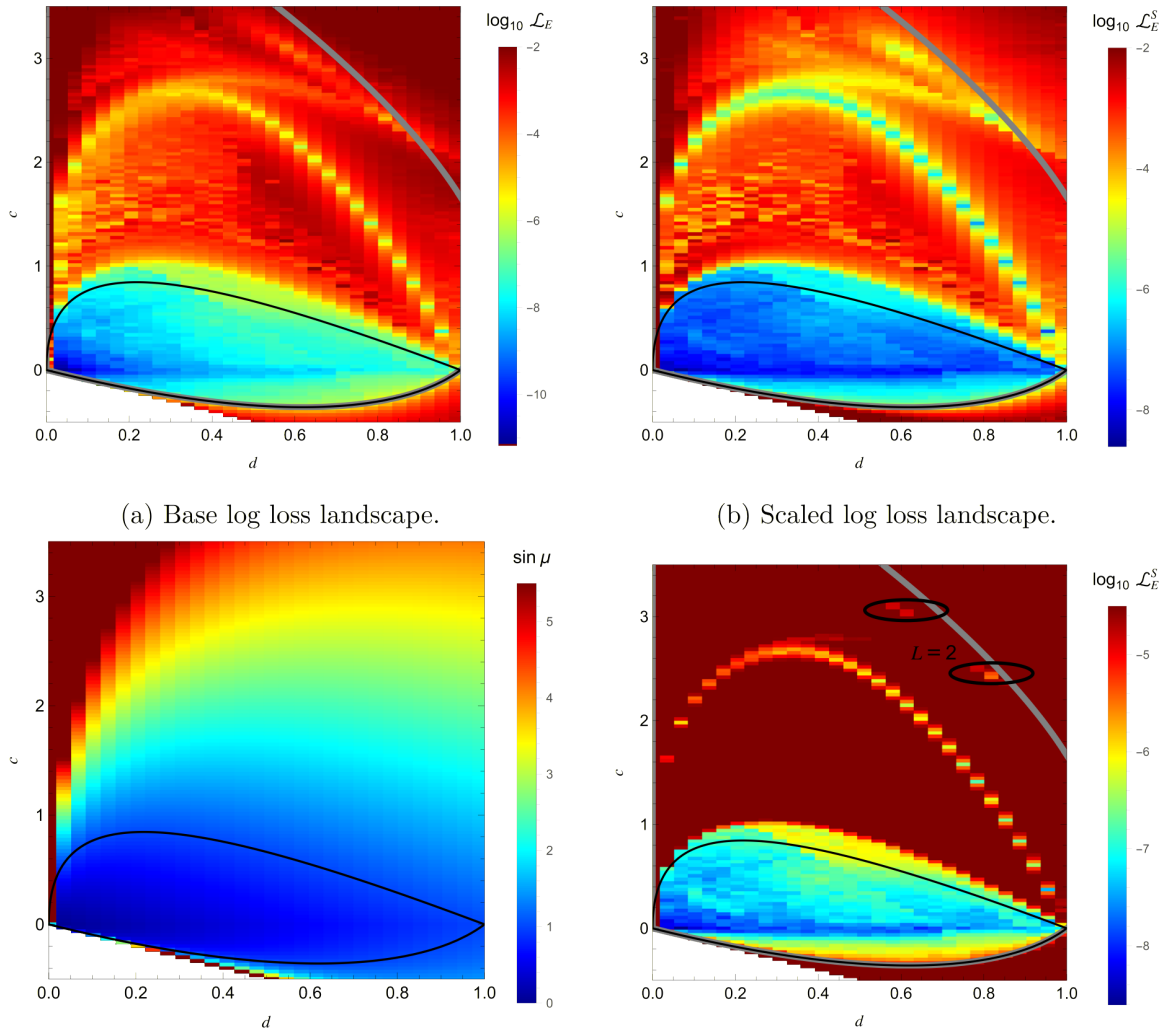
Figure 5. Scaled loss landscape for the two-parameter family of moduli $B(z) = cz^2 + d$. For each c and d we find a phase $\phi(z)$. Red regions indicate that no solution is likely. The black curve is $\sin\mu = 1$ and delimits the region within which we are guaranteed the existence of a solution. The grey curve delimits the bound of eq. (1.4) and solutions cannot be found outside of its enclosed region.

region, where a solution is guaranteed, the loss is generally small and $\mathcal{L}_E^S < 10^{-5}$ is always satisfied. This is a strong sign that our network is able to properly find the expected solutions in this region. We notice however that the network still finds approximate numerical solutions with $\mathcal{L}_E^S \sim 10^{-5}$ up to $\sin\mu \sim 1.1 - 1.2$.

In figure 6 we can see two regions of low loss away from $\sin\mu < 1$: two small islands circled in panel d and an additional one-dimensional curve along which the loss is around $\mathcal{L}_E^S \sim 10^{-6} - 10^{-5}$. Both regions are significantly outside of the $\sin\mu = 1$ boundary.

As shown in appendix A the two islands correspond to the genuine finite partial wave solutions of order $L = 2$, which have quadratic differential cross sections. Although genuine solutions, their associated loss is around $\mathcal{L}_E^S \sim 10^{-5}$ as their corresponding $\sin\mu$ values are quite high, around 2.95 and 3.67 respectively. Indeed, as $\sin\mu$ grows the networks become harder to train as can be understood from the structure of the loss function of eq. (2.2). For $\sin\mu > 1$ the cosine term in the integral needs to precisely modulate the kernel function in order to have a value that can be matched with $|\sin\phi(z)| < 1$. This difficulty is intrinsic to the unitarity loss and is responsible for the failure of the classical fixed point iterative scheme as $\sin\mu > 1$. Since our neural networks converge slower in that regime, we thus require either a higher number of training epochs or a further fine-tuning of hyperparameters to resolve these high $\sin\mu$ solutions with better accuracy. Alternatively, provided one knows that a finite partial wave solution is expected, it is also possible to first parametrize the unitarity amplitude as in eq. (1.1). One can then fit the phase shifts δ_ℓ by ensuring that

JHEP05(2024)200



(a) Base log loss landscape.

(b) Scaled log loss landscape.

(c) $\sin\mu$ landscape.

(d) Cut on the scaled log loss landscape.

Figure 6. Loss landscapes for the two-parameter family of moduli $B(z) = cz^2 + d$. Black and grey curves follow figure 3. For each c and d we find a phase $\phi(z)$. Red regions indicate that no solution is likely. Top panels show the base loss and scaled loss. Panel (c) shows a heat map of $\sin\mu$ values over the family $B(z) = cz^2 + d$. Right shows the scaled loss landscape with a hard cut of $\mathcal{L}_E^S \sim 10^{-4.5}$, along with the $L = 2$ finite partial wave solutions circled in black. The 1D curve corresponds to finite $L > 2$ solutions.

modulus $|F(z)|$ matches the $B(z)$ given as input. In that context, one can go back and forth between the machine learning scans and the classical fitting algorithm in order to fully characterize the low loss landscape.

Similarly, as described in appendix A, the 1D curve of $\sin\mu > 1$ solutions is associated with finite partial wave solutions of order $L \neq 2$. Their corresponding moduli are numerically well approximated by a quadratic $B(z)$ in the $z \in [-1, 1]$ region. Thus, even though their moduli are not quadratic per see, these spurious solutions show up in our scans and are associated with low loss values.

JHEP05(2024)200

3.3 Extremal amplitudes

Up until now, we have considered only toy amplitudes where the modulus is a low order polynomial. Here we demonstrate that the same technique can also be applied when using differential cross sections obtained in the context of the nonperturbative S-matrix bootstrap program, see e.g. [39–41]. One class of amplitudes of interest are "extremal" amplitudes that maximize or minimize the value of the amplitude at the crossing-symmetric point $\lambda = \frac{1}{32\pi}T(\frac{4m^2}{3}, \frac{4m^2}{3})$, which intuitively measures the strength of the interaction between particles. One can place bounds on this coupling by combining the so-called primal and dual methods [42]. The primal method consists of optimizing for the value of the coupling while satisfying all relevant constraints such as analyticity, crossing and unitarity. Any coupling within the primal bounds is defined to be in the allowed region. This approach is complementary to the dual method which places bounds on the excluded region. There one optimizes for the coupling while breaking the constraints as softly as possible. Since any coupling in the excluded region yields an unphysical theory, the extremal coupling is asserted to be between the primal and dual bounds. Numerically the maximal value of the coupling was obtained to be

$$2.66 \leq \max \lambda \leq 2.73, \tag{3.4}$$

whereas for the minimal coupling [41], the current bound is

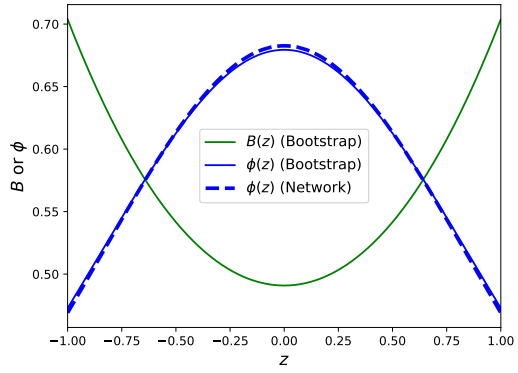
$$-8.02 \leq \min \lambda \leq -7.0. \tag{3.5}$$

It was also found that when maximizing/minimizing various couplings, while elastic unitarity was not imposed it effectively emerged with a good precision in the extremization process. This structure was further explored in [43]. In the context of the present work, it is therefore interesting to consider the differential cross-sections produced by the extremal amplitudes and check that elastic unitarity can indeed be satisfied by finding the appropriate $\phi(z)$.

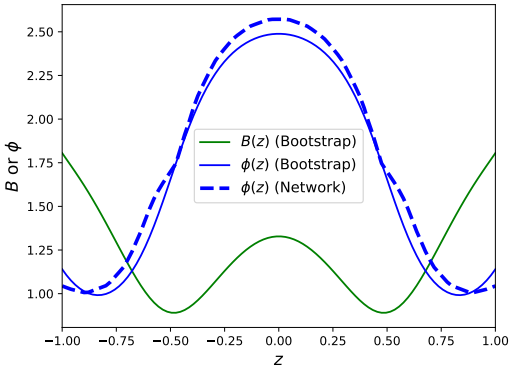
To analyze this case in more detail, we take the numerical results from [40] and use them to compute $B(z)$ that we then use as input. The functions $B(z)$ computed in this way are more physical than our toy functions in that they come from amplitudes that satisfy analyticity, unitarity and crossing at all energies. Due to the fact that they describe the scattering of identical particles, they obey $B(z) = B(-z)$. This symmetry is also respected by the expected phase, and we force our networks to output a symmetric result by considering $[\phi_\sigma(z) + \phi_\sigma(-z)]/2$ as the final output. In order to expedite the numerical evaluations of the input $B(z)$ functions, we will fit them by symmetric polynomials of finite order.

We find that more interesting results arise from the amplitudes that are closer to minimizing the coupling. Using the primal bootstrap results of [40] for $N_{\max} = 26$, we get $\lambda \approx -6.48$. We choose energies to be $s = 4.5m^2$ and $s = 8m^2$ and extract the appropriate $B(z)$ functions. Those are fitted by symmetric polynomials of order 14 and 40 respectively, which are then fed as inputs to our networks. Training is done over 5000 epochs, minimizing the scaled unitarity loss and using the default architecture and hyperparameters. The input $B(z)$ and associated phases are plotted on the top panels of figure 7, where we also include the phase predictions from the bootstrap amplitudes. At $s = 4.5m^2$ the modulus has $\sin\mu = 0.64$, while at $s = 8m^2$ the modulus is associated with $\sin\mu = 1.64$.

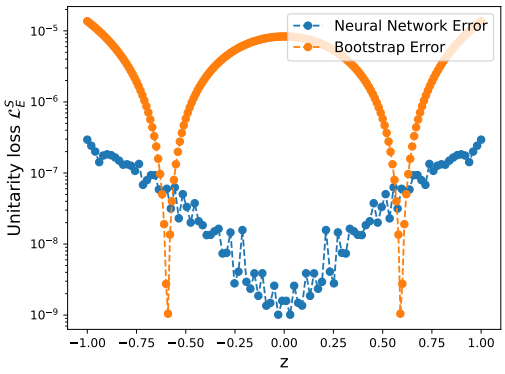
JHEP05(2024)200



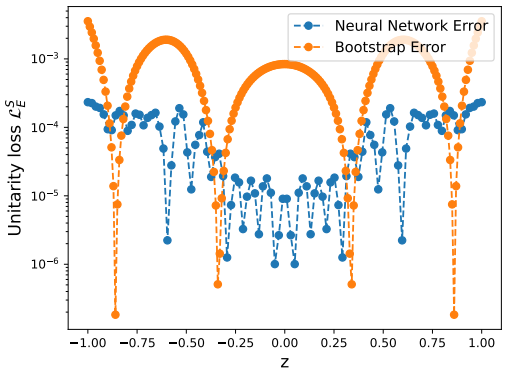
(a) Amplitudes at $s = 4 + 1/2m^2$.



(b) Amplitudes at $s = 8m^2$.



(c) Unitarity loss at $s = 4 + 1/2m^2$.



(d) Unitarity loss at $s = 8m^2$.

Figure 7. (Top pannels) $B(z)$ moduli and associated $\phi(z)$ phases coming from the bootstrap program of [40] for the amplitude with $\lambda \approx -6.48$ and $N_{\max} = 26$. We extract $B(z)$ at different s energies and use a neural network to predict the associated phase. (Bottom panels) Evaluation of the unitarity loss of eq. (2.2) on the predicted amplitudes coming from the bootstrap program and the trained neural networks.

To verify the accuracy of our results we also plot the evaluation of the unitarity loss of eq. (2.2) on the bottom panels of figure 7. This is done for both the trained networks and the bootstrap predictions and allows us to verify to which extent unitarity is broken in both cases. We can immediately notice that at $s = 4.5m^2$ the bootstrap and the neural network phases agree with one another and that in both cases unitarity is well respected. The unitarity loss is lower for the phase coming from the neural network as it has been specifically trained to minimize this quantity, while in the case of the bootstrap program elastic unitarity is only an emergent property. At $s = 8m^2$ the difference between the neural network and bootstrap phases is more pronounced. Unitarity is not respected with very good accuracy, in particular for the bootstrap phase. At the $z = \pm 1$ edges the loss reaches the order of 10^{-2} and indicates that numerically the emergent elastic unitarity property does not hold accurately.[5]

[5]This is further confirmed by numerically calculating $K(z)$ of eq. (1.5). For unitarity to hold we must have $\sin\phi(z) < K(z)$, which is respected by the neural network prediction. However, for the bootstrap phase, this

JHEP05(2024)200

4 Phase ambiguities: finite partial waves

So far we considered the existence question: given a modulus $B(z)$ when does there exist a phase $\phi(z)$ so that the amplitude $F(z) = B(z)e^{i\phi(z)}$ is unitary? A related question is: when are there two possible phases for the same $B(z)$? These phases should not be related by the redefinition, $\phi(z) \to \pi - \phi(z)$ (such redefinition is called a trivial ambiguity). The value of $\sin\mu$ is often considered as an indicator of whether ambiguous solutions may exist. The best bound in the literature [17] guarantees uniqueness for $\sin\mu < 0.89$, and it has been conjectured [3] that uniqueness should hold up to $\sin\mu = 1$. In practice, however, most known examples have $\sin\mu$ values much higher (above 2.0) and require a dedicated construction. In the following sections, we will see how machine learning can be used in conjunction with classical algorithms in order to study these ambiguous solutions. In this section, we focus on finite partial wave phase ambiguities and consider the infinite partial wave case in section 5.

In the finite L case, we both review known results and then discuss how machine learning can help. For finite L, the first phase-ambiguous solution was found by Crichton in 1966 and has $L = 2$, so the amplitude is quadratic in z. As we will see, for finite L there are an infinite number of phase-ambiguous solutions which decompose into 1d curves in the space of phase-shifts. The low dimensionality of the solution space makes the machine learning approach challenging. In the infinite L case, the solution space is higher-dimensional and easier to explore with gradient descent.

4.1 Classical solutions

We first review what is known about the finite L phase-ambiguous solutions classically, i.e. without machine learning. When there are a finite number of partial waves in an amplitude, the question of whether there are multiple phases for the same amplitude reduces to whether a finite set of equations can be solved simultaneously. For finite L we write the amplitude as

$$F(z) = \sum_{\ell=0}^{L} (2\ell+1) e^{i\delta_\ell} \sin(\delta_\ell) P_\ell(z) \tag{4.1}$$

where $P_\ell(z)$ are Legendre polynomials and $\delta_\ell \in \mathbb{R}$ are the phase shifts. This parameterization in terms of real phases guarantees that the amplitude is unitary. We are looking for another amplitude

$$\widetilde{F}(z) = \sum_{\ell=0}^{L} (2\ell+1) e^{i\widetilde{\delta}_\ell} \sin(\widetilde{\delta}_\ell) P_\ell(z) \tag{4.2}$$

with the same norm as $F(z)$. We are interested in non-trivial ambiguities.

To find non-trivial ambiguities we need two sets of phases δ_ℓ and $\widetilde{\delta}_\ell$ not all equal (and not all opposite) for which $B(z) = |F(z)|^2 = |\widetilde{F}(z)|^2$ is the same. Since $P_\ell(z)$ is a polynomial of degree ℓ, $|F(z)|^2$ is a polynomial of degree $2L$. So setting the coefficients of z^j from $|F(z)|^2$ equal to those of $|\widetilde{F}(z)|^2$ gives $2L+1$ equations for the $2L+2$ real phase shifts δ_ℓ and $\widetilde{\delta}_\ell$. This generically leads to a 1-dimensional solution space. Indeed, the finite L solutions for every L correspond to a set of 1D curves in $2L+1$ dimensions.

is broken near the $z = \pm 1$ edge of the angle range.

JHEP05(2024)200

The term associated with z^{2L} in $|F(z)|^2$ is $(2L+1)^2|P_L(z)|^2\sin^2\delta_L$. For this to be the same with δ_ℓ and $\widetilde{\delta}_\ell$ requires $\delta_L = \widetilde{\delta}_L$ so that the highest phase shift for the two solutions must be equal. There may or may not be one of the 1D curves which has a given value of δ_L. However, if we find a point on one of these curves with a given δ_L we can then move along the curve unambiguously to determine all the other connected solutions.

For example, with $L = 1$ equating the expression for $|F(z)|^2$ using the two sets of phase shifts gives

$$|F(z)|^2 = \sin^2\delta_0 - 6z\cos(\delta_0-\delta_1)\sin\delta_0\sin\delta_1 + 9z^2\sin\delta_1^2 \tag{4.3}$$

$$= \sin^2\widetilde{\delta}_0 - 6z\cos(\widetilde{\delta}_0-\widetilde{\delta}_1)\sin\widetilde{\delta}_0\sin\widetilde{\delta}_1 + 9z^2\sin\widetilde{\delta}_1^2 \tag{4.4}$$

Matching the coefficients of z^0, z^1 and z^2 gives $2L+1 = 3$ equations for the 4 phase shifts $\delta_0, \delta_1, \widetilde{\delta}_0$ and $\widetilde{\delta}_1$. The z^2 term forces $\delta_1 = \widetilde{\delta}_1$ and the z^0 forces $\delta_0 = \widetilde{\delta}_0$ so that there are no nontrivial solutions with $L = 1$.

The $L = 2$ case is already fairly complicated. A non-trivial solution with $L = 2$ was found by Crichton [19]:

$$\text{Set 1}: \begin{cases} \delta_0 = -\frac{7}{54}\pi \\ \delta_1 = -\frac{869}{3600}\pi \\ \delta_2 = \frac{1}{9}\pi \end{cases} \quad \text{and} \quad \text{Set 2}: \begin{cases} \widetilde{\delta}_0 = \frac{659}{1200}\pi \\ \widetilde{\delta}_1 = -\frac{59}{400}\pi \\ \widetilde{\delta}_2 = \frac{1}{9}\pi \end{cases} \tag{4.5}$$

that give rise to two amplitudes $F(z)$ and $\widetilde{F}(z)$ which share the same differential cross section.

Shortly after Crichton's paper, Atkinson, Johnson, Mehta and de Roo found the complete set of $L = 2$ solutions [20]. These form a 1D curve in the space of phase shifts, as expected. The value of $\sin\mu$ for these solutions is shown in figure 8 with Crichton's point indicated. Crichton's solution has $\sin\mu = 3.2$. For $L = 3$ the complete space of solutions is also known. For $L = 4$ only a handful of solutions are known.

Rather than attempting to improve these analytic results, we simply take a brute-force approach to finding solutions. To do so we first pick a random $\delta_L = \widetilde{\delta}_L$. Then we search for a solution to the remaining $2L$ equations and $2L$ unknows close to random seed points for the other δ_ℓ and $\widetilde{\delta}_\ell$. Once the equations are solved, we then confirm that the solutions are not trivially related. By sampling enough points one can see the emergence of a set of curves (one can also move along the curves to find connected solutions if desired).

Results for $L = 2, 3, 4$ and 5 are shown in figure 8. Interestingly we observe that the minimum value of $\sin\mu$ with $\delta_L \neq 0$ does not seem to decrease with L.[6] For $L = 2$, the lowest value has $\sin\mu \approx 2.63$. For $L = 3$ it is $\sin\mu \approx 3.41$. The lowest value for $L = 3$ is when $\delta_3 = 0$ which reduces it to $L = 2$. Such points do not show up in our search since they must have $\delta_L = 0$ *exactly* which will never occur in a random scan. For $L = 4$ and $L = 5$, the lowest values we found with $\delta_L \neq 0$ are $\sin\mu \approx 3.58$ and $\sin\mu \approx 3.83$ respectively. Based on these observations, we do not believe that going to higher finite L will reveal ambiguous solutions with $\sin\mu$ values smaller than those from $L = 2$.

[6] We have tentatively explored up to $L = 10$. For $L > 4$ there are so many curves that an enormous number of samples would be required to resolve them.

JHEP05(2024)200

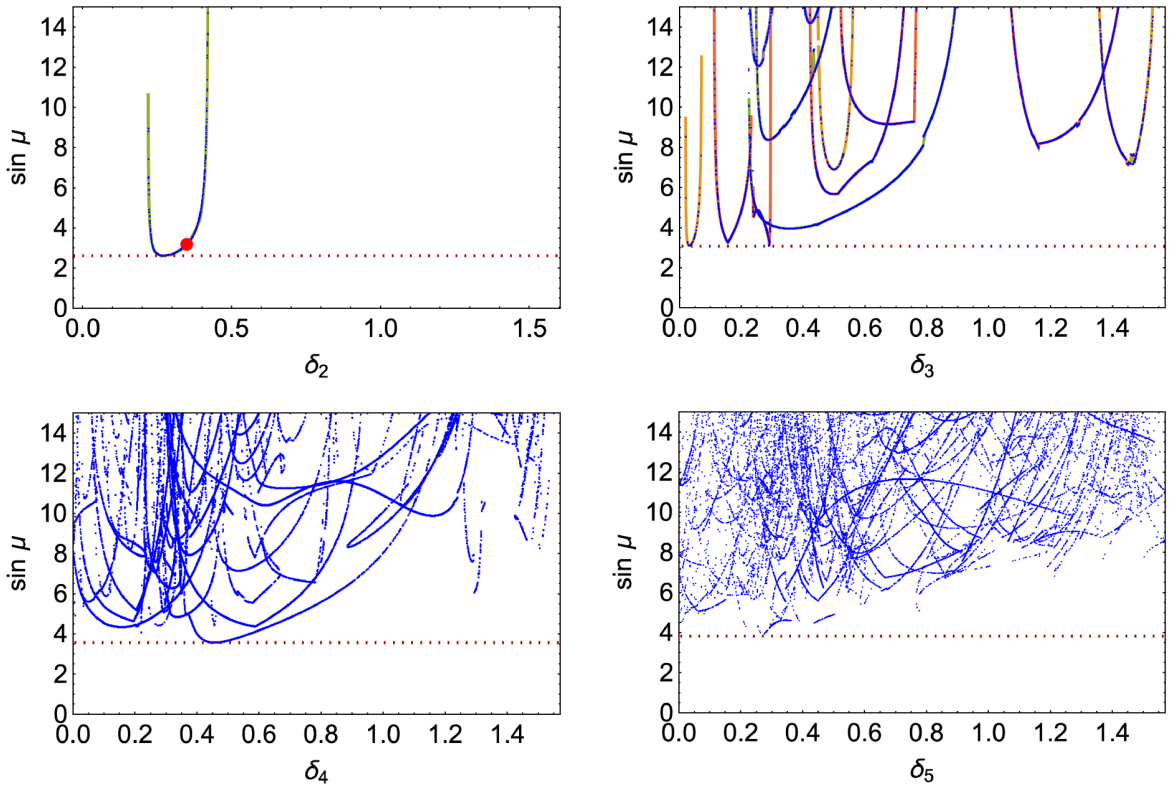


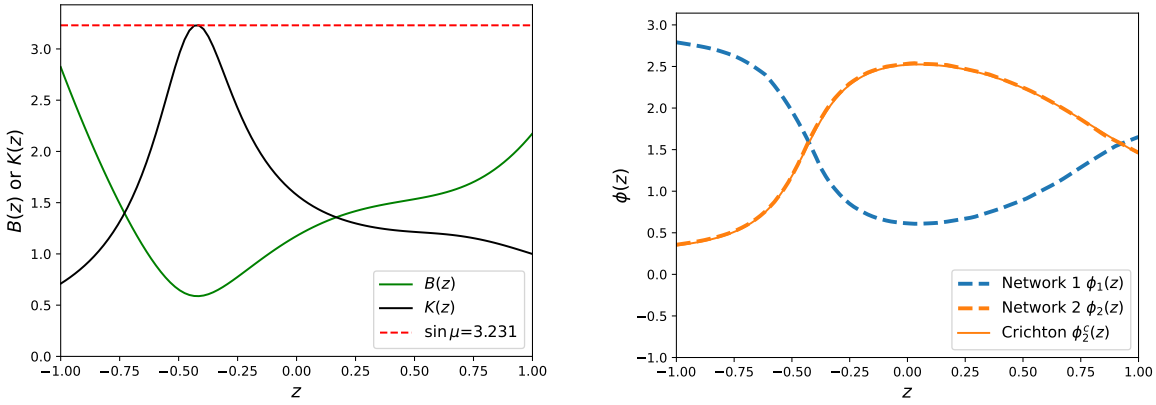
Figure 8. Finite partial-wave amplitudes with phase ambiguities separate into non-intersecting 1D curves in the $L+1$ dimensional space of phase shifts. For $L=2$ there is only one curve. These plots show a projection of these curves into the $\sin\mu, \delta_L$ plane for $L=2,3,4,5$. The point on the $L=2$ plot is Crichton's original ambiguity with $\delta_2 = \frac{\pi}{9}$. The solid curves are analytic solutions when known. The blue dots are a random scan. The red line gives the minimum $\sin\mu$ in each case.

4.2 Machine learning with repulsive loss

One might hope to use machine learning to find lower values of $\sin\mu$ for finite L. This becomes difficult because the solution space is one-dimensional. Thus, if one starts on a particular curve and does gradient descent in $\sin\mu$, one will only find the local minimum of that curve and never be able to jump to other curves. Fortunately, this is only a problem for finite L. Ambiguous solutions with infinite L fill higher dimensional regions which are easier to explore. The finite L case is still useful for exploring the machine learning approach as we have some exact phase ambiguous solutions, such as the Crichton one. So we will use finite L as a testbed for constructing a neural network capable of finding phase ambiguities. The lessons learned from these examples can then be applied to the more promising infinite L case.

We first attempt to recover both of Crichton's $L=2$ solutions. To begin, we simply try to find the phase multiple times and hope to get different answers based on different initialization seeds. We fix $B(z) = |F(z)|$ from Crichton's solution, as shown in figure 9(a). We first simply define two independent neural networks ($\phi_1(z)$ $\phi_2(z)$) and train them according to the principles of section 3 with different random initializations. We let the networks run for 5000 epochs using the loss of eq. (2.1). We do find two phases this way, as displayed in figure 9(b) alongside the theoretical solution coming from using the second set of phase

JHEP05(2024)200



(a) Crichton's modulus and integrated kernel.

(b) Redundant neural network phase solutions.

Figure 9. Crichton ambiguity: naively training using two independently initialized neural networks. On the right panel, we notice that the two phases are trivially ambiguous.

shifts $\widetilde{\delta}$. Unfortunately, the phases are trivially related: $\phi_1(z) = \pi - \phi_2(z)$. As we randomly initialize new neural networks we can end up recovering either solution (or the second Crichton solution). What is apparent from this simple experiment is that we need a way to avoid trivial ambiguities.

In order to study the uniqueness property associated with a given differential cross section we devise a methodology for consistently recovering different solutions with our neural networks. The setup is almost identical to the one used in section 3, where we start by independently initializing various neural networks that aim to solve the unitarity equation, minimizing the losses of either eq. (2.1) or eq. (2.2). The main deviation from this simple setup follows previous work in the literature [44] and consists in introducing a new repulsive term in the loss function. The role of this term will be to push apart the various neural network solutions and ensure that they do not overlap.

To introduce the repulsion term we must first define a measure for the closeness of two solutions $\phi_1(z)$ and $\phi_2(z)$. This measure should account for the periodicity properties of the phases to avoid boundary effects. One choice is to first define

$$d(\phi_1, \phi_2) = \mathbb{E}_z \left|\left| \left[\cos\phi_1(z) - \cos\phi_2(z)\right]^2 + \left[\sin\phi_1(z) - \sin\phi_2(z)\right]^2 \right|\right| \tag{4.6}$$

as a distance between two phase solutions.

For the repulsive loss itself, we will consider two alternatives. The first one, the *kick repulsion*, follows [44] and consists in introducing the pairwise loss

$$\mathcal{L}_{R1}^{(1,2)} = \left[d(\phi_1, \phi_2)\right]^{-p} + \left[d(\phi_1, \pi - \phi_2)\right]^{-p} \tag{4.7}$$

where p is some hyperparameter to be fixed. The first term in this loss ensures that we do not get exactly the same solutions, while the second term pushes us away from the trivial ambiguity. Since this repulsive term is always non-null, it will only be included in the loss function at intermediate epochs. More precisely, we let the networks first train for e_i epochs, then activate the repulsion and then turn it off after a total of e_f epochs.

JHEP05(2024)200

Repulsive loss type	Parameter	Parameter description
Kick repulsion	p	Strength of the inverse power law repulsion
	e_i, e_f	Start and end epochs where the repulsion is active
	λ_R	Repulsion loss to unitarity loss relative strength
Decaying repulsion	c_0	Initial repulsive factor
	s_f	Total scaling of the repulsive factor
	e_f	Epoch timescale where the repulsive factor grows
	λ_R	Repulsion loss to unitarity loss relative strength

Table 2. Hyperparameters to be tuned for the different repulsive losses considered.

In that way, this repulsive term in the loss function acts like a kick that pushes the two solutions apart but doesn't prevent the network from achieving arbitrary low loss even when the solutions are similar.

At a given epoch t the full loss function for N networks now reads

$$\mathcal{L}_K = \begin{cases} \sum_i^N \mathcal{L}_E^{(i)} + \lambda_R \sum_{i<j}^N \mathcal{L}_{R1}^{(i,j)} & \text{for } e_i < t < e_f \\ \sum_i^N \mathcal{L}_E^{(i)} & \text{for } t < e_i \quad \text{or } t > e_f \end{cases} \tag{4.8}$$

where λ_R is another hyperparameter defining the relative repulsion strength and $\mathcal{L}_E$ is the unitarity loss. At the end of the training run, we can then evaluate the repulsion term to check if the two solutions found are indeed not identical or correspond to the trivial ambiguity.

The second loss that we considered is a *decaying repulsion*. It consists of an interaction term that is not singular even when the solutions overlap. This is done by adding the loss term

$$\mathcal{L}_{R2}^{(1,2)} = 2 - \tanh[c(t)d(\phi_1, \phi_2)] - \tanh[c(t)d(\phi_1, \pi - \phi_2)] \tag{4.9}$$

where $c(t)$ is some hyperparameter. In order to precisely fit the different solutions we want the repulsion to be inconsequential as the training nears the end. This is achieved by making the parameter $c(t)$ epoch dependent, increasing throughout the training. In particular, we will take

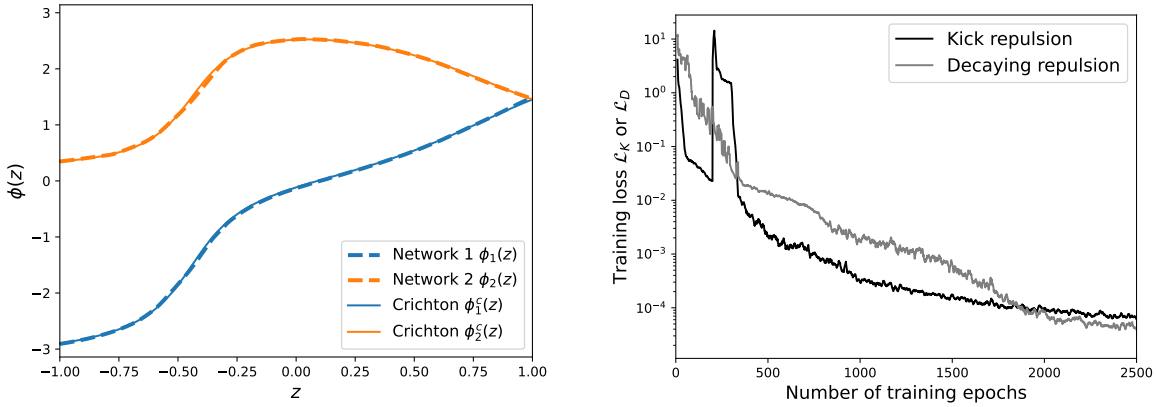
$$c(t) = \Delta_1 \tanh\left(\frac{t}{a} - b\right) + \Delta_2 \tag{4.10}$$

where the parameters are chosen such that $c(t)$ starts at $c(0) = c_0$ and reaches 99% of its maximal final value c_f after e_f epochs, with $c(e_f) = 0.99 c_0 s_f$ where s_f is the total scale factor.[7] At a given epoch t the full loss function for N networks now reads

$$\mathcal{L}_D = \sum_i^N \mathcal{L}_E^{(i)} + \lambda_R \sum_{i<j}^N \mathcal{L}_{R2}^{(i,j)} \tag{4.11}$$

[7] Additionally we ask for $c'(t)$ to be maximum at $e_f/2$. These constraints fix the parameters of $c(t)$ to be $a = e_f/(2\,\mathrm{arctanh}\,[(100 - 99s_f)/(100 - 101s_f)])$, $b = e_f/(2a)$, $\Delta_1 = c_0(100 + 99s_f)/200$ and $\Delta_2 = c_0(-100 + 101s_f)/200$.

JHEP05(2024)200



(a) Learned neural network phases.

(b) Loss functions for the different repulsion types.

Figure 10. Recovering the Crichton ambiguity with the addition of a repulsive loss. On the left panel, we show the two distinct ambiguous phases recovered by our networks. On the right panel, we compare the evolution of the two repulsive losses introduced in eq. (4.8) and eq. (4.11).

where the repulsive loss term is always included.[8] We summarize the hyperparameters for the two types of repulsive losses in table 2.

Adding the repulsive loss $\mathcal{L}_{R1}$, we train a neural network for 10000 epochs using the unitarity loss of eq. (2.1). After a quick search of the hyperparameter space, we use $p = 2$, $e_i = 200$, $e_f = 300$ and $\lambda_R = 1.0$, although other values could be considered with additional fine-tuning. The phases recovered by the networks are shown in figure 10(a) and are plotted against Crichton's prediction of the finite partial wave parameterization. The final losses for each phase are around $\mathcal{L}_E^S \sim 10^{-5}$, in good agreement with the expected answer. We note that for the solid blue curve in figure 10(a) we represent the phase coming from using Crichton's phase shifts $-\delta_l$, which corresponds to representing the trivial ambiguity associated with Crichton's first solution.

We also train another neural network with the $\mathcal{L}_{R2}$ repulsive loss using $c_0 = 2, s_f = 16, e_f = 1000$ and $\lambda_R = 50.0$. The resulting phases also have a final loss around $\mathcal{L}_E^S \sim 10^{-5}$. We compare the loss functions using the kick and decaying repulsion in figure 10(b), where we use a moving average window of 10 epochs to smooth the plot. Whereas the decaying repulsion has a smoothly decreasing loss we can indeed verify that the kick repulsion sharply boosts the loss across a given window.

5 Infinite partial wave ambiguities

Looking for amplitudes with ambiguous phases that have an infinite number of partial waves has also been explored classically. We first review some results in this direction, then apply machine learning to the infinite L case.

[8] In practice when the repulsive loss starts to be smaller than a tenth of the unitarity loss we will discard it to facilitate training. This check is to be performed throughout training and the repulsive loss can be reactivated as soon as we breach that threshold.

JHEP05(2024)200

5.1 Classical solutions

One approach to finding phase ambiguities with infinite L is based on partially factorizing the amplitude and conjugating some of its zeros. The following discussion follows [23].

Any amplitude can be written (non-uniquely) as

$$F(z) = g(z) \prod_{\ell=1}^{L} \frac{z - z_\ell}{1 - z_\ell} \tag{5.1}$$

for some L and some $g(z)$. When $g(z) = 1$ the amplitude is a polynomial with a finite number of phase shifts. For $L = 0$, this is just $F(z) = g(z)$ an arbitrary function. This form is still useful, since if we conjugate any number of z_ℓ the amplitude will have the same norm. In particular, the amplitude

$$\widetilde{F}(z) = g(z) \prod_{\ell=1}^{L} \frac{z - z_\ell^\star}{1 - z_\ell^\star} \tag{5.2}$$

has $|F(z)| = |\widetilde{F}(z)|$. Constructing amplitudes in this way guarantees they have the same norm but does not guarantee unitarity. And moreover, even if $F(z)$ is unitary, $\widetilde{F}(z)$ generally will not be.

In order to make progress, one needs to restrict the class of functions searched. Ref. [23] focused exclusively on the simplest case where two amplitudes differ by a single zero. Doing a partial wave decomposition of $g(z)$ we can write

$$\begin{aligned} F(z) &= \frac{z - z_1}{1 - z_1} \sum_{\ell=0}^{\infty} (2\ell + 1) \left(\frac{\gamma_\ell - 1}{2i} \right) P_\ell(z) \\ \widetilde{F}(z) &= \frac{z - z_1^\star}{1 - z_1^\star} \sum_{\ell=0}^{\infty} (2\ell + 1) \left(\frac{\gamma_\ell - 1}{2i} \right) P_\ell(z) = \frac{z - z_1^\star}{z - z_1} \frac{1 - z_1}{1 - z_1^\star} F(z) \end{aligned} \tag{5.3}$$

This is similar to a normal partial wave decomposition

$$F(z) = \sum_{\ell=0}^{\infty} (2\ell + 1) \left(\frac{S_\ell - 1}{2i} \right) P_\ell(z) \tag{5.4}$$

where $S_\ell = e^{2i\delta_\ell}$ so that the unitarity condition $\delta_\ell \in \mathbb{R}$ is equivalent to $|S_\ell| = 1$. Because of the $\frac{z - z_1}{1 - z_1}$ prefactor the unitarity condition is not $|\gamma_\ell| = 1$ but rather $|S_\ell| = 1$. Solving for the S_ℓ in terms of the γ_ℓ gives

$$S_\ell = \frac{1}{1 - z_1} \left[\frac{(\ell + 1)\gamma_{\ell+1} + \ell \gamma_{\ell-1}}{2\ell + 1} - z_1 \gamma_\ell \right] \tag{5.5}$$

The condition that $|S_\ell| = 1$ then gives a recursion relation among the γ_ℓ. Writing

$$\gamma_\ell = 1 - \epsilon_\ell \tag{5.6}$$

this relation can be written in descending form:

$$\epsilon_{\ell-1} = \frac{2\ell + 1}{\ell} \epsilon_\ell \operatorname{Re}(z_1) - \frac{\ell + 1}{\ell} \epsilon_{\ell+1} + \frac{2\ell + 1}{\ell} (1 - \operatorname{Re} z_1) \left[1 \pm \sqrt{1 + \frac{(\operatorname{Im} z_1)^2}{(\operatorname{Re} z_1 - 1)^2} \epsilon_\ell (2 - \epsilon_\ell)} \right] \tag{5.7}$$

or equivalently in ascending form:

$$\epsilon_{\ell+1} = -\frac{\ell}{\ell+1}\epsilon_{\ell-1} + \frac{2\ell+1}{\ell+1}\epsilon_\ell \operatorname{Re}(z_1) + \frac{2\ell+1}{\ell+1}(1-\operatorname{Re} z_1)\left[1 \pm \sqrt{1 + \frac{(\operatorname{Im} z_1)^2}{(\operatorname{Re} z_1 - 1)^2}\epsilon_\ell(2-\epsilon_\ell)}\right] \tag{5.8}$$

Note the sign ambiguity: there are generally two solutions at each step leading to 2^n sign choices. Generally, only one of them will give finite amplitudes, with $\varepsilon_\ell \to 0$ as $\ell \to \infty$. There is nevertheless no clear criterion for deciding which sign to choose at each recursion step.

As discussed by [23] an additional necessary condition for the unitarity of $F(z)$ following from $|S_\ell| = 1$ and finiteness is that

$$\operatorname{Im}(\gamma_\ell^\star \gamma_{\ell-1}) = 0 \tag{5.9}$$

This means that each pair of successive γ_ℓ must have the same phase. The phase can only change if $\gamma_L = 0$ for some L, in which case the γ_{L-1} and γ_{L+1} can have different phases. So overall the series of γ_ℓ comprises sequences with the same phase separated by zeros. Moreover, for the amplitude to be finite $\gamma_\ell \to 1$ as $\ell \to \infty$, which means that there has to be some L beyond which all of the γ_ℓ are real. Since some γ_ℓ has to be complex (or else $F(z)$ and $\widetilde{F}(z)$ are complex conjugates), we conclude that $\gamma_{L-1} = 0$ for some L and $\gamma_\ell \in \mathbb{R}$ for $\ell \geqslant L$. Atkinson et al. call such solutions *class L*.

Class 2 amplitudes have $\gamma_0 \in \mathbb{C}$, $\gamma_1 = 0$ and $\gamma_\ell \in \mathbb{R}$ for $\ell > 1$. One way to find such solutions for a given z_1 is by guessing a $\gamma_2 \in \mathbb{R}$ and recursing upwards. Given z_1 and γ_2 we can solve for γ_0 using the downward recursion relations. This gives

$$|\gamma_0| = \left|\frac{1-z_1}{z_1}\right|, \quad \operatorname{Re}(\gamma_0) = \frac{1}{4\gamma_2}\left(9|1-z_1|^2 - \left|\frac{1-z_1}{z_1}\right|^2 - 4\gamma_2^2\right) \tag{5.10}$$

From here, one can iterate upwards from γ_2 demanding that γ_ℓ be real for $\ell > 2$. It is only possible for γ_ℓ to be real if the upward-iterating discriminant is real, which implies

$$1 - \sqrt{\frac{(z_1-1)(z_1^\star-1)}{(\operatorname{Im} z_1)^2}} \leq \varepsilon_\ell \leq 1 + \sqrt{\frac{(z_1-1)(z_1^\star-1)}{(\operatorname{Im} z_1)^2}} \tag{5.11}$$

This must hold for all ℓ so, in particular, it gives an allowed range for $\varepsilon_2 = 1-\gamma_2$. For a given value of γ_2 and z_1 as one iterates upwards one may find that some higher γ is imaginary. If this happens for all choices of signs in the recursion relation then that value of γ_2 is disallowed. This approach is called the ascending iteration. Although ref. [23] found the ascending iteration inefficient, we find it can actually work quite well.

An alternative search procedure is the descending iteration. There one starts at large ℓ where $|\varepsilon_\ell| \ll 1$ and one can linearize the recursion relation. Solving the linearized version exactly gives $\varepsilon_\ell = CQ_\ell\left(\operatorname{Re} z_1 + \frac{\operatorname{Im}^2 z_1}{\operatorname{Re} z_1 - 1}\right)$ with $Q_\ell(x)$ the Legendre polynomial of the second kind and C a constant. One can then take this form for ε_L and ε_{L+1} for some large L and iterate downwards. Only for some value of C will $\gamma_1 = 0$. Thus one can search for $\gamma_1 = 0$ via the shooting method, as one might search for eigenfunctions of a differential equation.

We find that both the ascending and descending solutions are computationally extremely intense, sometimes requiring 60+ digits of precision to converge to a trustworthy solution.

JHEP05(2024)200

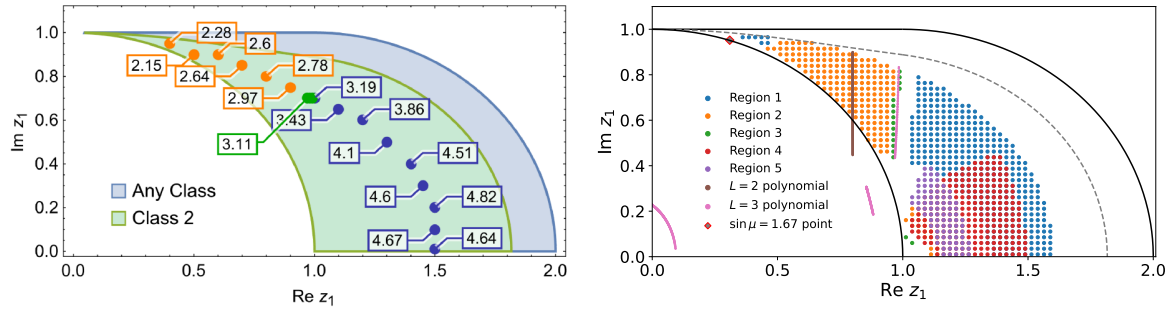


Figure 11. The left panel shows the region in z_1 space where non-polynomial amplitude pairs with a single conjugated zero are possibly allowed. The smaller green region is the possibly-allowed region for class 2 amplitudes where $\gamma_1 = 0$. Points are solutions found in [23] labelled by their values of $\sin\mu$. Different colours correspond to different choices of signs in the descending iteration. The right panel shows solutions we found using the descending iterative scheme of ref. [23]. Different colours are different regions corresponding to different sign choices at various steps in the iteration. Class 2, $L = 2, 3$ partial waves polynomial amplitudes are also represented.

The choice of which sign to pick on each step of the recursion is problematic as the search tree grows exponentially. The descending solution is inferior in the sense that it can never produce an exact solution since the asymptotic form is assumed to be reached at some finite L; the ascending solution can give an exact answer if the seed at low γ_ℓ is exact. In addition, we find that in some cases, the asymptotic behaviour at large ℓ is approached very slowly: even at $\ell = 100$, the partial wave coefficients are not exponentially small. In particular, this happens for solutions that are strongly peaked at $z = \pm 1$ requiring large numbers of modes in their partial wave decomposition. Such solutions happen to be the ones which we found with low values of $\sin\mu$ (see figure 15 below).

In ref. [23], Atkinson et al. focused exclusively on parameterizations of the form of eq. (5.3) with one zero conjugated. They showed that non-polynomial solutions are only possible if $|z_1| > 1$, $|\mathrm{Im}\, z_1| < 1$ and if in the region where $|\mathrm{Re}\, z_1| > 1$ the additional constraint $\sqrt{|z_1|^2 - 2|\mathrm{Re}\, z_1| + 1} \leq 1$ holds. The majority of the examples were class 2 (which has $\gamma_1 = 0$) for which one can impose the stronger constraints $|1 - z_1| \geq \frac{3}{2}\left|\frac{1-z_1}{z_1}\right|(|z_1| - \frac{1}{3})|\mathrm{Im}\, z_1|$ and $|1 - z_1| \geq \frac{3}{2}\left|\frac{1-z_1}{z_1}\right|(|z_1| - \frac{1}{3})\sqrt{|z_1|^2 - 2|\mathrm{Re}\, z_1| + 1}$ in the $|\mathrm{Re}\, z_1| > 1$ region. These regions and the explicit points Atkinson et al. found are shown in figure 11. The lowest value of $\sin\mu$ listed was $\sin\mu = 2.15$.

A challenge with the iterative approach is that there is no clear prescription for which sign to choose at each step in the iteration. When using the descending iteration ref. [23] defined Region I to have all positive signs in the recursion relation in eq. (5.7). Region II is defined by choosing $-$ for $\ell = 2$ and $+$ for all other ℓ in eq. (5.7). Region III has a minus sign for $\ell = 2, 3$ only, while region IV has minus signs at $\ell = 2, 3, 4$. We also define a new region V which has a minus sign for $\ell = 3$ only. The points belonging to these respective regions are coloured differently in figure 11. We extend this study by searching for solutions in a 80×40 grid in z_1 space. Our results are shown on the right side of figure 11. Although not clear from the figure, we find that the regions overlap: some points have solutions for the same z_1 but different sign choices in the iteration. Note that such pairs of solutions have different moduli so there are still only at most two phases for a given modulus (it is only z_1 which the solutions share).

JHEP05(2024)200

Although Atkinson and collaborators found a number of phase-ambiguities at infinite L, they could only consider a small class of functions. They restricted to a single zero conjugated, as in eq. (5.3), and moreover looked exclusively at class 2 amplitudes where $\gamma_1 = 0$ in their numerical work. Even with these assumptions, searching through the different regions is tedious, and trying to generalize to other classes or multiple conjugated zeros would be a herculean task. We next explore how machine learning can search more efficiently for solutions.

5.2 Machine learning complex functions

For the machine learning approach, we start with the class of functions in eq. (5.3) with a single conjugated zero. However, we do not need to do a partial wave decomposition. So we write simply

$$F(z) = \frac{z - z_1}{1 - z_1} f(z), \qquad \widetilde{F}(z) = \frac{z - z_1^\star}{1 - z_1^\star} f(z) \tag{5.12}$$

where $f(z)$ is the function to be learned and z_1 will be treated as a fixed input. The function $f(z)$ is parametrized by a neural network, closely following the implementation of section 2, however for this application we need $f(z)$ to be a complex function instead of a real phase. In order to avoid complex numbers we have two obvious choices: have the network learn the modulus and phase of $f(z)$ or have it learn the real and imaginary parts of $f(z)$. We tried both approaches but found that learning the real and imaginary parts was the most promising so we restrict to that choice in the following discussion.

For our neural network implementation, we modify the neural network depicted in figure 1 by removing the final *Tanh* layer and having two final outputs instead of one. Removing the *Tanh* layer is justified since we are now predicting the real and the imaginary part of the amplitude directly and thus do not need to constrain them within a given finite range. The two amplitudes are then reconstructed following the eq. (5.12) and training is done using the unitarity loss of eq. (2.2) and the decaying repulsion of eq. (4.9).

Although the iterative algorithm described in section 5.1 and the machine learning implementation try to find similar solutions, the approaches differ in several key points:

1. The iterative algorithm requires the choice of a particular coefficient γ_L that is to be set to zero, yielding a *class* L amplitude. In the machine learning implementation, the same neural network is used to recover any type of solution. For the same z_1 there can be phase-ambiguous solutions of different classes. Whereas these different class solutions can be individually picked out by the classical algorithm, the machine learning implementation will naturally tend to yield the one that is easiest to train, typically the one that has the lowest $\sin\mu$ value.

2. The classical algorithm requires the choice of signs for the discriminant at each step in the iteration. So for L steps, there are 2^L choices, leading to 2^L "regions". Most of these choices will not yield solutions, but there is no clear way to restrict the search. In the machine learning approach, the regions play no role: the algorithm will find solutions in all regions automatically.

JHEP05(2024)200

3. In the iterative algorithm unitarity is automatically enforced by the partial wave decomposition. However, the iteration may yield divergent results. A shooting method is used to find a sensible boundary condition for the iteration. In the machine learning implementation unitarity is enforced by minimizing a loss function. In practice one has to impose a cutoff on the loss $\mathcal{L}_E$ in order to assess whether the learned amplitudes do indeed respect unitarity.

4. In the iterative algorithm, it is easy to see if the iterative solution yields a trivial ambiguity since the complex coefficients are known. In the machine learning approach, a repulsive loss has to be used to avoid trivial ambiguities. This makes the resolution of ambiguous solutions that are naturally close to one another more difficult.

5. Finding a solution with the classical algorithm can require high numerical precision. For example, we found that to confirm solutions close to the boundary of allowed z_1 values (where $\sin\mu$ is minimized) one can require 60 digits of precision or more. On the machine learning side, the precision is limited by the numerical integration scheme required when calculating the unitarity constraint of eq. (2.1). One generally cannot expect to have more than 5-10 digits of precision at best. However, to explore the space of solutions, high numerical precision is not required, as we could see in previous examples in section 3 or section 4.2.

In summary, the ML algorithm has the advantage of not needing a bunch of discrete choices and special cases to search. Thus it has the potential to search for a much broader class of solutions than the classical algorithm. On the other hand, its numerical precision is limited: you can never know if it actually finds a solution or not. An optimal approach may be to combine the two approaches: exploring the landscape of solutions with machine learning and then using a classical algorithm to refine particular solutions we find.

5.3 Resolving the z_1 landscape with ML

As a warm-up, we take one of the points and solutions found in [23] which belongs to region I: $z_1 = \frac{6}{5} + \frac{3}{5}i$. With this z_1 value, we train a network for 5000 epochs with the unitarity loss of eq. (2.2) and using the decaying repulsion with $c_0 = 2$, $s_f = 16$, $e_f = 1000$ and $\lambda_R = 2.0$. We extract the phases corresponding to the amplitudes of eq. (5.12) along with their respective moduli (which are identical by construction). We show these in figure 12, along with the solutions recovered by the iterative algorithm. The final loss values are around $\mathcal{L}_E^S \sim 10^{-6}$ for both solutions, and we observe good agreement with the answer derived from the classical approach.

We note that the amplitudes and phases for this solution are similar to the ones that we obtained when solving for the Crichton ambiguity in section 4.1. This observation will hold for a major part of the z_1 plane (where solutions are expected to be found) and prompts us to implement a better initialization for our neural network. When searching for another solution along the z_1 plane, it will be advantageous to initialize the network with a solution from a neighbouring point. The main training run will then be done over a smaller number of epochs and will not consider any repulsion term for the loss function. Since the phase solutions will

JHEP05(2024)200

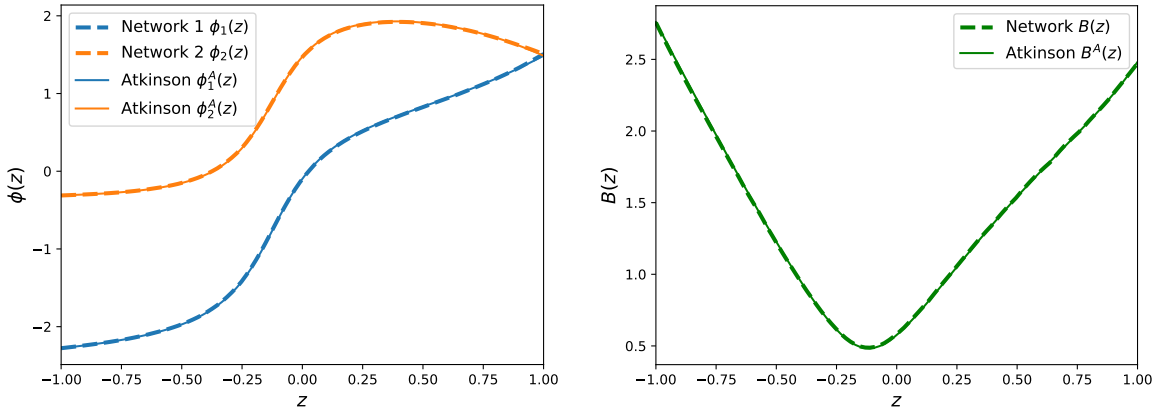


Figure 12. Phase-ambiguous solutions rediscovered with machine learning compared to previous results from [23]. This solution has the form of eq. (5.12) with $z_1 = \frac{6}{5} + \frac{3}{5}i$.

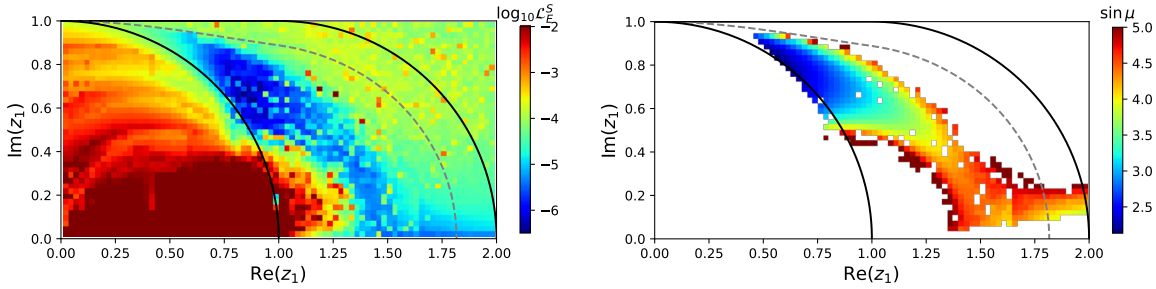


Figure 13. Loss landscape and $\sin\mu$ values in the search for phase ambiguities in amplitude pairs with a single conjugated root. The solid black lines delimit the region reachable with the descending algorithm of [23]. Class 2 amplitudes (those with $\gamma_1 = 0$) are only allowed between the grey dashed line and the lower black line $|z_1| = 1$. The low-loss points outside of the lower black curve ($|z_1| = 1$) correspond to finite-L solutions. Right panel shows the $\sin\mu$ landscape for low losses $\mathcal{L}_E^S < 10^{-4.5}$ and non degenerate solutions $\mathcal{L}_{R2} < 0.99$.

be seeded at initialization as being properly distinct, we do not expect further training to modify them drastically and, instead, we will recover the two genuine ambiguous solutions.[9]

Next, we explore the space of z_1 values with phase ambiguities. To do so, we create a grid of 80×40 points in the complex z_1 plane with $\mathrm{Re}\, z_1 \in [0, 2]$ and $\mathrm{Im}\, z_1 \in [0, 1]$. This region is motivated by the known bound on the allowed range of z_1 (see figure 11), but as we will see, that region will be rediscovered independently by the network. At each point, we train for 500 epochs where the new neural networks are initialized with the trained networks of a nearest neighbor.[10]

[9] One can view this property as starting the training run near the correct minimum of the loss landscape, as opposed to near a spurious minimum corresponding to a trivial ambiguity. In practice, after training, we will explicitly verify the nature of the solutions, for example by computing the value of the repulsion loss at evaluation.

[10] We start the procedure at the point $z_1 = \frac{6}{5} + \frac{3}{5}i$, which we associate with the trained networks shown in figure 12. A point is only trained if one of its neighbours has been previously resolved.

JHEP05(2024)200

The loss landscape from our scan is displayed on the left in figure 13. The black curves here are an analytical bound on possible solutions: all possible infinite-L single-conjugated-root solutions must lie within those curves. That does not mean that the entire region is allowed. The bounds also do not apply to the finite-L polynomial solutions which can (and do) have $|z_1| < 1$, as discussed in appendix B. From the figure, we can clearly see a region of low loss within the allowed region. In addition we recover a small domain with $|z_1| < 1$ that has low loss, which is associated with polynomial solutions. We also note that not all of the allowed region has low loss. Indeed, for $\mathrm{Re}(z_1) \in [1, 1.2]$ and $\mathrm{Im}(z_1) \in [0, 0.3]$ the networks all have high evaluation losses with $\mathcal{L}_E^S > 10^{-3}$, indicating a failure to recover ambiguous solutions in that area. However, the iterative classical algorithm shows that some of the points should correspond to genuine solutions (see regions IV, V in figure 11). Upon inspection we see that the solutions in this region are highly oscillatory — they do not resemble the functions obtained in the region I. Since these solutions are sufficiently dissimilar they cannot hope to be resolved by neural networks that have been initialized following another class of solutions and would require independent training and optimization trials in order to be recovered. This is certainly possible. But these solutions also have $\sin\mu$ values that are much higher than in the other domains of the z_1 plane. Because of the higher $\sin\mu$ values, this region is of no particular interest and we have not pursued its exploration further.

One feature distinguishing the machine learning loss landscape from the solution space of the classical algorithm is its continuity. The loss landscape does not suffer from the sharp boundaries that the classical algorithm experiences and is smooth across the domain in the z_1 plane that it resolves. Deforming z_1 slightly will result in another solution with a similar phase and differential cross section and the machine learning algorithm can do that interpolation easily. This is to be contrasted with the landscape emerging from the descending algorithm which struggles at the boundaries of the different regions, requiring enormous precision and fine-tuning to find solutions there. This is seen most clearly in the region near $\mathrm{Re}\, z_1 = 1$ where regions I, II and III can possibly overlap (see figure 11). The machine learning algorithm has no trouble in this region whereas the classical descending algorithm requires either an increasing number of partial waves or a broader search for its shooting method.

On the right of figure 13 we show the $\sin\mu$ values for the points in our scans where we only retain points that have $\mathcal{L}_E^S < 10^{-4.5}$. Additionally, we discard points corresponding to identical or trivial solutions. These points are characterized by having either of the terms in the loss of eq. (4.9) above 0.99. We note that possible phase-ambiguous solutions which happen to be very similar, such as those with $\mathrm{Re}\, z_1 > 1.5$ and small $\mathrm{Im}\, z_1$, are discarded by this second cut.

From this study, we see that the smallest $\sin\mu$ values tend to be close to the $|z_1| = 1$ curve. The lowest value of $\sin\mu$ that we found with this initial scan is $\sin\mu = 2.13$ at $z_1 = 0.56 + 0.84i$ and is located in the region where the loss of the network starts to near $\mathcal{L}_E^S \sim 10^{-4.5}$. To get a lower value we can use the fact that the $\sin\mu$ landscape is continuous, allowing one to do a constrained gradient descent on it. The constraint that we have to respect here is one where the descent does not take us into regions of high loss. To implement the gradient descent we numerically estimate $\nabla_{z_1} \sin\mu(z_1)$ by using a central difference. This is done by training four networks at $z_1 \pm h$ and $z_1 \pm ih$ for 500 epochs with $h = 10^{-3}$ and calculating the respective

JHEP05(2024)200

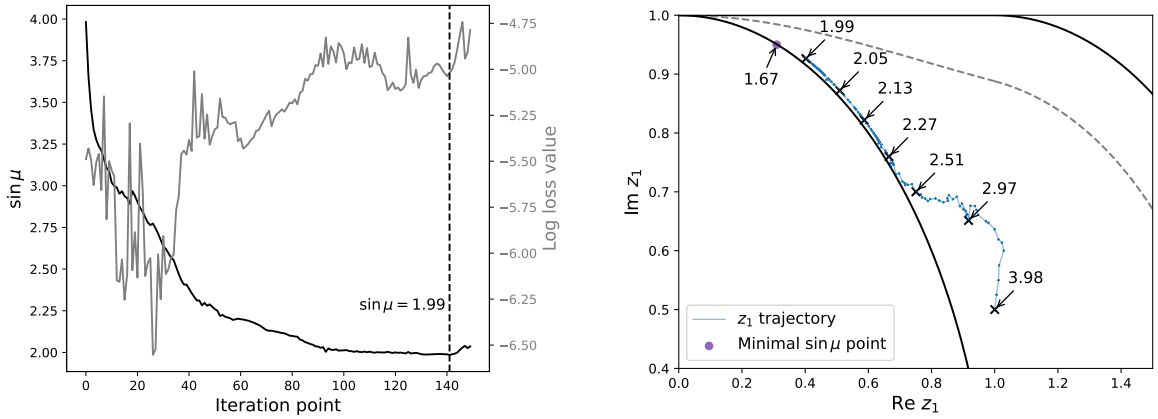


Figure 14. Gradient descent in the z_1 plane for minimizing $\sin\mu$. The left panel shows the value of $\sin\mu$ along the trajectory, along with the scaled loss $\mathcal{L}_E^S$ at each point. The right panel displays the trajectory in the z_1 plane. The point with the minimal $\sin\mu$ is found with the classical ascending algorithm by extending the gradient descent trajectory.

$\sin\mu$ value at those points. In order to accelerate convergence each network is initialized with the solved network at the z_1 point considered. We then take a step in the z_1 plane following[11]

$$z_1^{\text{new}} = \begin{pmatrix} \text{Re}\, z_1^{\text{new}} \\ \text{Im}\, z_1^{\text{new}} \end{pmatrix} = \begin{pmatrix} \text{Re}\, z_1^{\text{old}} \\ \text{Im}\, z_1^{\text{old}} \end{pmatrix} - \frac{\lambda_r}{2h} \begin{pmatrix} \sin\mu(z_1^{\text{old}} + h) - \sin\mu(z_1^{\text{old}} - h) \\ \sin\mu(z_1^{\text{old}} + ih) - \sin\mu(z_1^{\text{old}} - ih) \end{pmatrix} . \tag{5.13}$$

A new network is then trained at z_1^{new} and the process is iterated. In figure 14 we show such a gradient descent trajectory of 150 points where we used the learning rate $\lambda_r = 0.005$. The minimal value of $\sin\mu$ along the trajectory is 1.99, noticeably lower than in the initial scan. This is due to the fact that the region with the low $\sin\mu$ values is better resolved. The gradient descent takes small steps in the problematic region and the networks are trained to lower loss values, remaining under $\mathcal{L}_E^S \sim 10^{-5}$ as can be verified in the left panel of figure 14.

Following this procedure, we can then use the classical ascending algorithm to further refine the lowest $\sin\mu$ point in a systematically improvable way. Following the gradient descent curve we are led to a point at $z_1 = 1.001e^{0.4\pi i} = 0.31 + 0.95i$. When implementing the ascending algorithm we choose the Region II solution. The amplitude and phases for this point are shown in figure 15. It has $\sin\mu \approx 1.67$. This is the lowest known $\sin\mu$ value with phase ambiguous solutions. The phase shifts for this solution are given in table 3. At large ℓ the phase shifts in this solution oscillate and decay exponentially $\delta_\ell \sim (-1)^\ell \ell^a e^{-b\ell}$ with $b \approx 0.076$ and $a \approx 0.45$.

Looking at this solution it appears that $B(z)$ is peaked near $z = \pm 1$ and the phases are roughly linear going between $\frac{\pi}{2}$ at $z = 1$ to either π or 2π at $z = -1$. By exploring this structure, we can consider a toy amplitude of the form

$$F(z) = ae^{i\phi_a}\delta(z-1) + be^{i\phi_b}\delta(z+1) \tag{5.14}$$

[11] If $|z_1^{\text{new}}| < 1$ we project out of the unit circle by considering $1.01\, z_1^{\text{new}}/|z_1^{\text{new}}|$ instead. This allows us to remain in regions of relatively low loss where we can trust our gradient descent.

JHEP05(2024)200

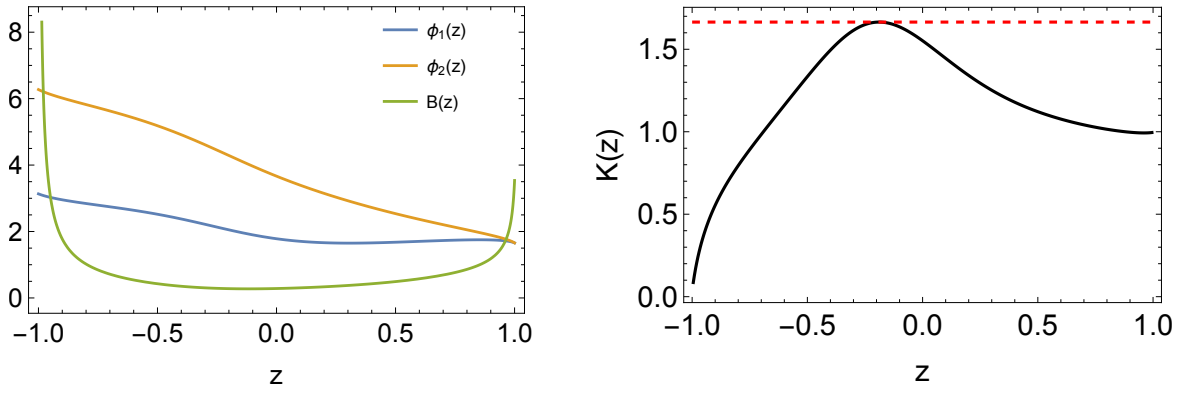


Figure 15. Left panel shows the modulus and unitary phases for the amplitude with $\sin\mu \approx 1.67$. Notice that it is peaked both in the forward, $z = 1$, and backwards, $z = -1$, directions with $B(1) \approx 3.53$ and $B(-1) \approx 44.2$. For $z = 1$ both phases approach $\frac{\pi}{2}$, whereas for $z = -1$ they approach π and 2π correspondingly. Curiously, the same features are exhibited by the simple toy model (5.14). Right panel shows the integrated kernel whose maximum is $\sin\mu$. This is the lowest known value of $\sin\mu$ for which a phase ambiguity exists.

with a, b, ϕ_a and ϕ_b real numbers. Unitarity then implies

$$\sin\phi_a = \frac{a^2+b^2}{2a}, \quad \sin\phi_b = a\cos(\phi_a - \phi_b), \quad \sin\mu = \max\left(\frac{a^2+b^2}{2a}, a\right) \tag{5.15}$$

Recall that the expected lower bound on $\sin\mu$ for which there are ambiguous solutions is $\sin\mu = 1$. If $a < 1$ then the condition $\sin\mu = 1$ automatically leads to the value $\phi_a = \frac{\pi}{2}$ and then the second constraint in eq. (5.14) becomes $\sin\phi_b = a\sin\phi_b$. Since we assumed $a < 1$ we must have $\phi_b = \pi, 2\pi$. Remarkably, these are exactly the $z = \pm 1$ endpoints of the phases we found, see figure 15. That is, we find two solutions

$$F_\pm(z) = ai\delta(z-1) \pm \sqrt{2a-a^2}\,\delta(z+1) \tag{5.16}$$

with $a < 1$. Unfortunately, these two solutions are related by $F_+ = -F_-^\star$ which is a trivial ambiguity. Based on this feature, it is interesting to contemplate a scenario in which any family of phase-ambiguous solutions will approach the trivial ambiguity as $\sin\mu \to 1$. It would be very interesting to explore this further.

5.4 Extensions beyond one zero

So far we have concentrated our efforts on finding solutions where a single zero z_1 is complex conjugated. This is only a small subset of all possible ambiguous solutions. In general, one could consider probing ambiguous solutions constructed from complex conjugating multiple different zeros. With the classical approach, one could try to construct an iterative algorithm in the style of [23]. With more zeros, there is a larger space to search and many more discrete choices to make. While progress is possible, it seems extraordinarily tedious to search this way. The complexity of the classical approach is to be contrasted with the flexibility of the machine learning implementation, where one simply needs to modify the parametrization of eq. (5.12) by taking out the appropriate number of zeros $z_1, \ldots, z_n$. Training can then proceed in the exact same way with no additional conceptual work.

JHEP05(2024)200

δ_0	-0.7435785523	δ_6	-0.12099793	δ_{16}	-0.037063302
δ_1	0.3847784634	δ_7	0.10455316	δ_{17}	0.033422548
		δ_8	-0.091873270	δ_{18}	-0.030227979
$\widetilde{\delta}_0$	-0.1982113969	δ_9	0.080754325	δ_{19}	0.027337227
$\widetilde{\delta}_1$	0.5583871796	δ_{10}	-0.071709663	δ_{20}	-0.024780662
		δ_{11}	0.063677489	δ_{21}	0.022462722
δ_2	-0.314656996	δ_{12}	-0.056937764	δ_{22}	-0.020400075
δ_3	0.20231948	δ_{13}	0.050906239	δ_{23}	0.018527050
δ_4	-0.16811695	δ_{14}	-0.045741950	δ_{24}	-0.016852096
δ_5	0.14018563	δ_{15}	0.041096977	δ_{25}	0.015329178

Table 3. Here we present first 25 phase shifts for the $\sin\mu \approx 1.67$ solution with two possible phases. Only the first two phase shifts differ between the two amplitudes. To find the results we ran the ascending algorithm with $\sim 100 - 200$ modes and checked that the significant figures quoted above are convergent, in that they are not sensitive to how many modes are included.

As a warm-up with multiple zeros, we show that a known ambiguous $L = 3$ solution can be reproduced with the same ML construction. Ref. [21] considered cases where one root z_3 is held fixed and the other two roots z_1 and z_2 are conjugated:

$$F(z) = a\frac{(z-z_1)(z-z_2)(z-z_3)}{(1-z_1)(1-z_2)(1-z_3)} \tag{5.17}$$

$$\widetilde{F}(z) = a\frac{(z-z_1^\star)(z-z_2^\star)(z-z_3)}{(1-z_1^\star)(1-z_2^\star)(1-z_3)} \tag{5.18}$$

For the machine learning setup, we parameterize

$$F(z) = \frac{(z-z_1)(z-z_2)}{(1-z_1)(1-z_2)}f(z) \tag{5.19}$$

$$\widetilde{F}(z) = \frac{(z-z_1^\star)(z-z_2^\star)}{(1-z_1^\star)(1-z_2^\star)}f(z) \tag{5.20}$$

we can then take z_1 and z_2 as inputs and learn a single complex function $f(z)$ as we did in the case of one zero.

To be concrete, one example solution found in [21] had $a \approx 2.80+4.67i$, $z_1 \approx -0.74-0.06i$, $z_2 \approx 0.19+0.03i$ and $z_3 \approx 1.32+0.95i$. We take only z_1 and z_2 from this known solution and then train to find $f(z)$. We train the network for 5000 epochs using the decaying repulsion with the parameters of section 5.3. To speed up training we also seed the neural network for $f(z)$ by the one that has been trained on the $z_1 = \frac{6}{5} + \frac{3}{5}i$ point. The result is shown in figure 16. We find that the phases and moduli of the two solutions agree with those found in [21]. The agreement is robust, with an evaluation loss around $\mathcal{L}_E^S \sim 10^{-5}$ indicating that we indeed recovered the expected solution. We note that although the phases in the left panel of figure 16 appear to be discontinuous, the functions predicted by the network, $\mathrm{Re}f(z)$ and $\mathrm{Im}f(z)$, are themselves continuous. This translates into the real and imaginary parts of the amplitude $F(z)$ being continuous, as displayed on the right panel of figure 16. The modulus and integrated kernel $K(z)$ are also shown. The $\sin\mu$ value comes from the maximum of $K(z)$, which gives $\sin\mu \approx 18$ for this example.

JHEP05(2024)200

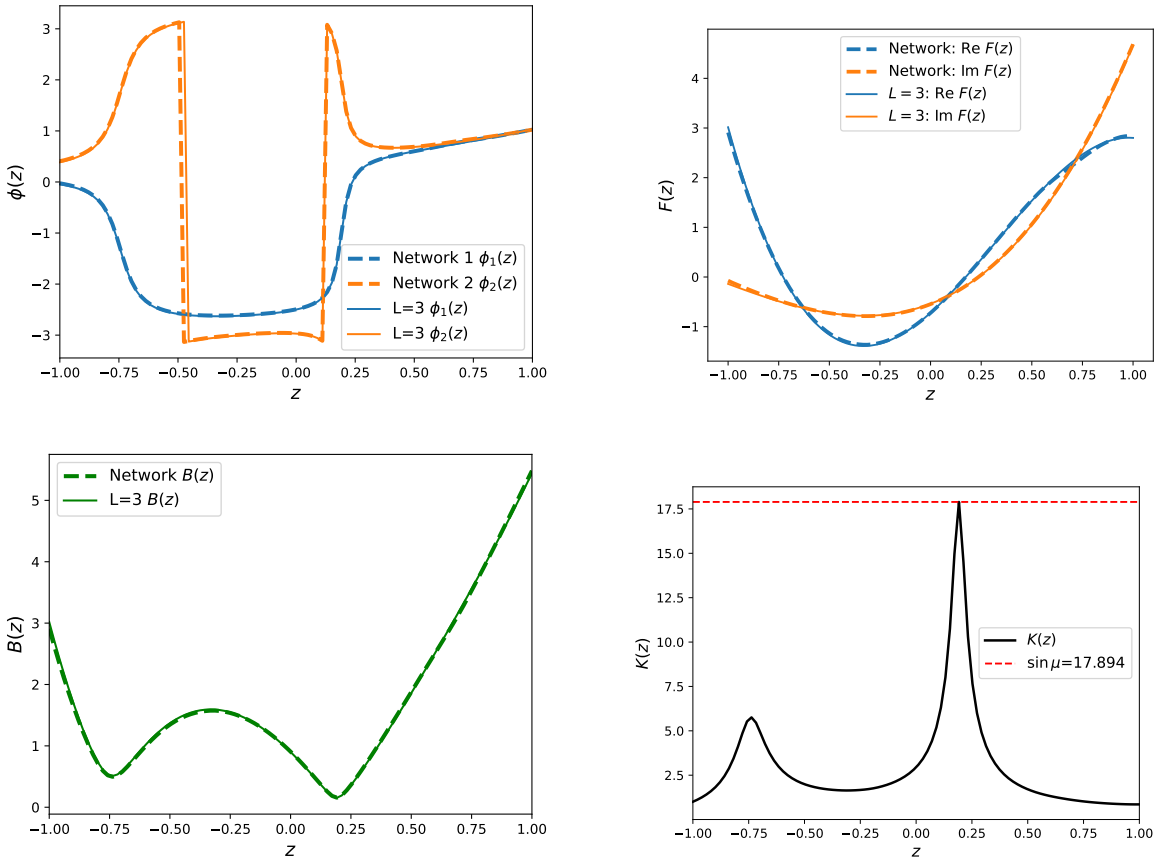


Figure 16. Resolving the ambiguous solutions associated with a finite $L = 3$ partial wave differential cross section. The two solutions differ by complex conjugation of two of their zeros. We display the phase outputs of the neural network compared to the exact solutions (top left panel) and the prediction for the real and imaginary parts of the first solution's amplitude $F(z)$ (top right panel). The bottom panels show the learned modulus $B(z)$ and the corresponding kernel $K(z)$.

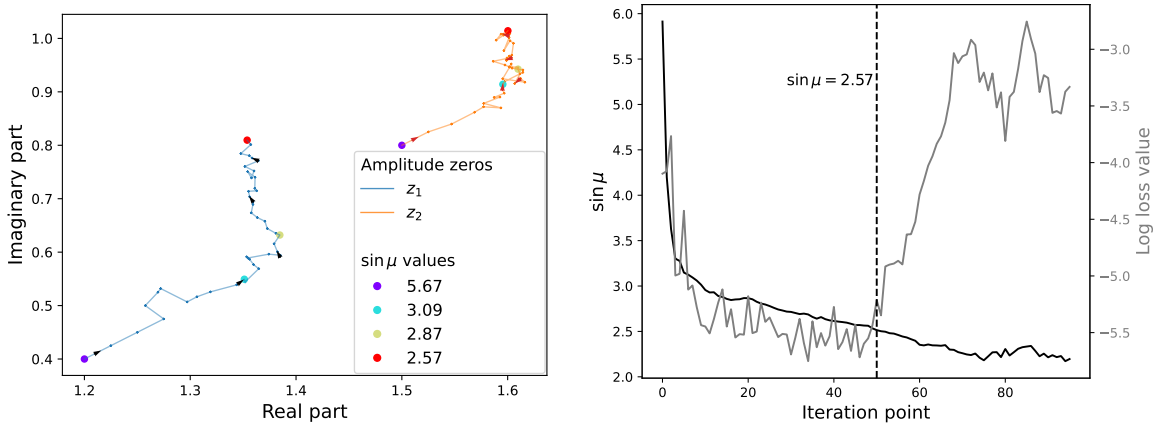


Figure 17. Gradient descent in the z_1, z_2 space for minimizing $\sin\mu$. The left panel displays the trajectories of both the z_1 and z_2 roots for the points along the descent where $\mathcal{L}_E^S$ remains small. The right panel shows the value of $\sin\mu$ along the trajectory, along with the scaled loss $\mathcal{L}_E^S$ at each point. The gradient descent trajectory cannot be trusted once the evaluation loss blows up.

JHEP05(2024)200

Following the procedure outlined in section 5.3 one can also proceed to do gradient descent in the z_1, z_2 space in order to minimize $\sin\mu$. An example of a gradient descent trajectory is displayed in figure 17 where both roots are simultaneously updated at each point of the descent. The main difficulty we encounter is ensuring that the gradient descent follows a trajectory of low loss. As can be observed in the right panel of figure 17, the evaluation loss associated with our trained networks starts to blow up at a given point along the trajectory, after which we cannot trust that viable ambiguous solutions are being recovered. The last value that can be trusted yields an ambiguous solution with $\sin\mu \approx 2.57$. In order to reach a trustworthy lower value of $\sin\mu$ one could envision modifying the gradient descent update of eq. (5.13) by forbidding updates that increase $\mathcal{L}_E^S$ substantially. Given this constraint or a modification of the loss function used, along with a more powerful neural network architecture, one can then imagine searching through the $z_1, \cdots, z_n$ space and resolving an ambiguous solution with a smaller $\sin\mu$ value than the one of table 3. While finding low $\sin\mu$ phase-ambiguous solutions in this way is conceivable, our initial assessment suggests that the amount of oversight required for this approach outweighs its probability for success, so we have not pursued this direction further.

6 Conclusions

In this paper, we have explored the problem of determining the phase of an amplitude from its modulus using modern machine learning. In the elastic scattering regime, the modulus and phase of an amplitude are constrained by a non-linear integral equation which enforces unitarity. Although the equation is difficult to solve analytically or with traditional numerical methods, it is easily solved with machine learning. Given a modulus $B(z)$ with z the cosine of the scattering angle, a phase $\phi(z)$ for the amplitude can be parameterized as a neural network, then determined from unitarity through gradient descent. Using this technique we were able to reproduce known results for finite partial wave amplitudes, infinite partial wave amplitudes, and amplitudes determined by other S-matrix bootstrap principles. A few obvious extensions of our work include focusing on the scattering of identical particles, considering elastic scattering of spinning and/or flavored particles which would lead to a coupled system of unitarity equations, as well as doing the computation in the general number of spacetime dimensions d.

More generally, it would be very interesting to apply machine learning to explore the full amplitude in both energy and angle, as in the classic analysis of pion scattering in [45], or more recent explorations of the space of nonperturbative amplitudes starting from [39]. This would require imposing in addition to unitarity the constraints of analyticity and crossing. In fact, methods very similar to the ones used to analyse elastic scattering at fixed energy were developed for the full amplitude by Atkinson, see e.g. [46–49], and were recently successfully implemented numerically [43]. They are based on iterations of unitarity and are expected to converge only for a small subset of admissible amplitudes. More powerful gradient-descent type methods to construct the full amplitude have not been developed yet and it is to be seen if machine learning could be useful to tackle this problem.

Coming back to the present paper, there are two important open questions in S-matrix theory which we have shown machine learning approaches can address. The first is whether a

JHEP05(2024)200

phase $\phi(z)$ exists at all for a given differential cross section, or equivalently, a given modulus $B(z)$ of the scattering amplitude. This problem is solvable in a straightforward manner with machine learning. Code to find $\phi(z)$ from $B(z)$ is available here https://github.com/aurelienndersy/S-Matrix-Bootstrap. It has been proposed that a functional $\sin\mu$ which involves a non-linear integral over $B(z)$ (see eq. (1.6)) is a good criterion for whether a solution exists. It has been shown that for $\sin\mu < 1$ a solution always exists. If $\sin\mu > 1$ there are three possibilities 1) no phase may exist 2) a unique phase may exist 3) two non-trivially related phases may exist. Examples are known in all three cases. No clear criterion is known however to determine which case applies for a given $B(z)$ in general. Only options 2) and 3) are possible for $\sin\mu < 1$ but no criterion is known to decide which. The analytical bound is that uniqueness must hold if $\sin\mu < 0.86$ or the average modulus $\frac{1}{2}\int_{-1}^{1} dz B^2(z)$ is less than 1.38. In the literature, ambiguous solutions are known with at best $\sin\mu \approx 2.15$. Using machine learning, we have found $B(z)$ with ambiguous phases with $\sin\mu \approx 1.67$. This is the first improvement on this bound in 50 years.

The machine learning approach offers several distinct advantages over classical approaches. The framework we have developed for solving the unitarity integral equation is general and recovering a phase can be attempted for any input modulus. This is to be contrasted with classical fixed point iteration schemes, which only converge if $\sin\mu < 1$. This straightforwardness has enabled us in section 3.1 to extensively explore various polynomial moduli and determine which ones are consistent with unitarity. We confirmed the existence bounds set by $\sin\mu < 1$ and identified the region in moduli space where $\sin\mu > 1$ solutions could be expected. Extending the setup to probe the uniqueness of the solution space was equally conceptually simple and only required the addition of a repulsive term to the loss function. Classical approaches have instead focused only on analytically solvable cases, such as finite partial waves of low order, or on specific parametrizations for the amplitudes. One such parametrization proposed in [23] was reviewed in section 5.1 and contrasted with a machine-learning solution of the same problem. Whereas the classical algorithm required multiple discrete choices of parameters, carving out separate solution regions, the machine learning algorithm was able to smoothly interpolate across the whole solution landscape. This flexibility comes at a cost, lack of numerical precision, but allows a complementarity approach to traditional numerical methods. It is in that spirit that we have demonstrated in section 5.3 that one can utilize the smoothness of the machine learning loss landscape to perform gradient descent and find ambiguous solutions with low $\sin\mu$. There the inflexible, but powerful, classical algorithms allowed further refinement in order to obtain the lowest possible solution precisely. Extensions to other amplitude parameterizations are immediate with our machine learning framework whereas developing the corresponding classical iterative schemes (if possible) would require considerable amounts of effort.

Although here we focused on the narrow problem of the relationship between the modulus of an amplitude and its phase in the elastic scattering regime, a similar methodology can be used for much broader questions. The S-matrix bootstrap approach attempts to apply a set of general constraints such as unitarity, analyticity, and crossing to constrain the form of amplitudes. Implementing these constraints directly[12] is a nontrivial task and the subject

[12]The so-called primal approach, see e.g. the discussion in [42].

JHEP05(2024)200

of many ongoing works, see e.g. [39–41, 43]. Machine learning offers the potential to search through a broad class of functions and perform the gradient descent efficiently, as we have seen here for phase-ambiguities in the elastic regime. In addition, a similar methodology could help in the inelastic regime, where given particle production one can ask whether it is possible to reconstruct the associated scattering amplitude by solving a similar unitarity equation [50]. Finally, the methods developed in this paper could also be useful to the analytic S-matrix bootstrap which applies constraints such as collinear limits or possible locations of singularities to perturbative scattering amplitudes. The classical approach has already been very successful, bootstrapping the 6-point amplitude in $\mathcal{N} = 4$ super-Yang-Mills theory to 6 loops this way [51]. Additional constraints are known, such as those on sequential discontinuities [52], but have not been incorporated. Machine learning could make it easier to apply additional constraints and it would be very interesting to explore the potential of machine learning for the S-matrix bootstrap further. This paper represents just a small first step into a field with enormous possibilities.

Acknowledgments

We would like to thank Filip Niewinski for their collaboration in the early stages of this work. We also thank Zohar Komargodski, Piotr Tourkine, and Jiaxin Qiao for useful discussions. AD and MDS are supported in part by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement number 949077).

A Finite partial wave decomposition

In our study of polynomial amplitudes, we were mainly concerned with recovering solutions that admitted an infinite partial wave decomposition. In order to verify this point we characterize here the space of unitary amplitude solutions that admit a finite partial wave decomposition and a corresponding polynomial $B(z)$. We start our analysis by looking for solutions that could have a corresponding linear $B(z) = az + b$. Since $B(z)^2$ is a polynomial of order 2, our finite partial wave solution must be of order 1 and parameterized as

$$F_1(z) = e^{i\delta_0} \sin\delta_0 + 3e^{i\delta_1} \sin\delta_1 z \tag{A.1}$$

where we used the explicit representation for the first two Legendre polynomials. Equating $|F_1(z)|^2 = (az + b)^2$ we find the system

$$\begin{cases} a^2 &= 9\sin^2\delta_1 \\ b^2 &= \sin^2\delta_0 \\ 2ab &= 6\sin\delta_0 \sin\delta_1 \cos(\delta_0 - \delta_1) \end{cases} \tag{A.2}$$

which can only be realized for $a = 3b$ with $\delta_0 = \delta_1$. Then we have $F_1(z) = \pm e^{i\delta_0}\left(az + \frac{a}{3}\right)$ that satisfies the unitarity constraint and $|F_1(z)|^2 = |a|^2|z + \frac{1}{3}|^2$. The corresponding modulus is

JHEP05(2024)200

$B(z) = |az + \frac{a}{3}|$ where the absolute value is necessary to ensure positivity for $-1 < z < 1$. Thus we do not have a simple finite partial wave amplitude that is associated with $B(z) = az + b$ where $b > |a|$ as was considered in section 3.2.1.

For the quadratic modulus $B(z) = az^2 + c$ we can proceed in a similar fashion, parameterizing

$$F_2(z) = e^{i\delta_0} \sin\delta_0 + 3e^{i\delta_1} \sin\delta_1 z + \frac{5}{2} e^{i\delta_2} \sin\delta_2 (3z^2 - 1) \tag{A.3}$$

and deriving a similar system of equations

$$\begin{cases} a^2 &= \frac{225}{4} \sin^2\delta_2 \\ 0 &= \sin\delta_1 \sin\delta_2 \cos(\delta_1 - \delta_2) \\ 2ac &= 9\sin^2\delta_1 - \frac{2}{3}a^2 + 15 \sin\delta_0 \sin\delta_2 \cos(\delta_0 - \delta_2) \\ 0 &= \sin\delta_1 \sin\delta_0 \cos(\delta_0 - \delta_1) \\ c^2 &= \sin^2\delta_0 + \frac{a^2}{9} - 5\sin\delta_2 \sin\delta_0 \cos(\delta_0 - \delta_2) \end{cases} \tag{A.4}$$

We have 3 different solution sets. The first one with $\delta_1 = 0$ does not lead to a valid solution with both $a > 0$ and $c > 0$. The second solution set has $\delta_0 = 0$ and $\delta_1 = \delta_2 \pm \frac{\pi}{2}$. For $\delta_1 = \delta_2 - \frac{\pi}{2}$ we can have $a = \frac{15}{4}\sqrt{\frac{3}{7}}$ with $a = 3c$ that leads to a valid quadratic differential cross section. The third solution set has $\delta_1 = \delta_2 \pm \frac{\pi}{2}$ and $\delta_0 = \delta_1 \pm \frac{\pi}{2}$. For $\delta_1 = \delta_2 - \frac{\pi}{2}$ and $\delta_0 = \delta_1 - \frac{\pi}{2}$ we can have $a = \frac{5}{2}\sqrt{\frac{3}{2}}$ with $a = 5c$ that leads to a valid solution. Summarizing, for $B(z) = az^2 + c$ we have two valid solutions with $a > 0$ and $c > 0$:

$$F_2^a(z) = -\frac{15z}{2\sqrt{7}} e^{-i\sin^{-1}\left(\frac{5}{2\sqrt{7}}\right)} + \frac{5}{4}\sqrt{\frac{3}{7}}\left(3z^2 - 1\right) e^{i\sin^{-1}\left(\frac{1}{2}\sqrt{\frac{3}{7}}\right)} \tag{A.5}$$

$$F_2^b(z) = \frac{1}{\sqrt{6}} e^{i\sin^{-1}\left(\frac{1}{\sqrt{6}}\right)} - \sqrt{\frac{15}{2}} z e^{-i\sin^{-1}\left(\sqrt{\frac{5}{6}}\right)} + \frac{5}{2\sqrt{6}}\left(3z^2 - 1\right) e^{i\sin^{-1}\left(\frac{1}{\sqrt{6}}\right)} \tag{A.6}$$

which are respectively associated with the moduli

$$B_2^a(z) = \frac{5}{4}\sqrt{\frac{3}{7}}\left(3z^2 + 1\right) \tag{A.7}$$

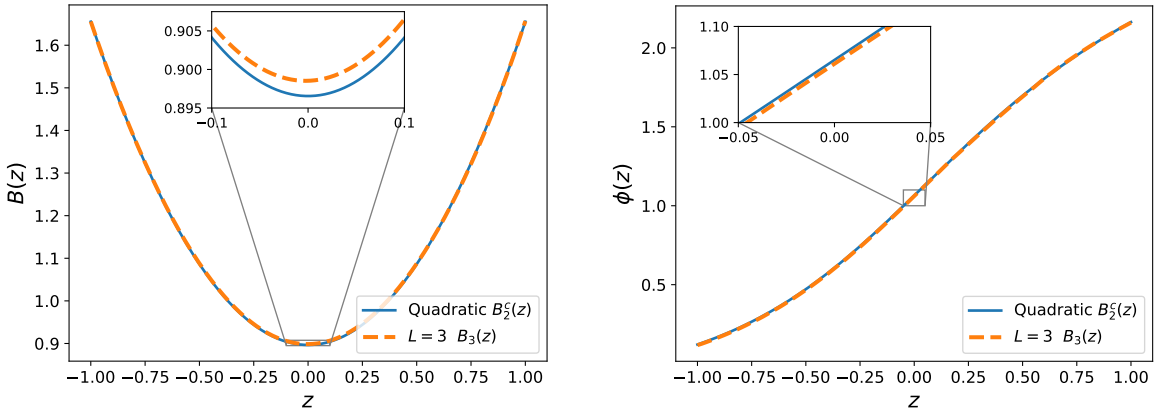
$$B_2^b(z) = \sqrt{\frac{3}{8}}\left(5z^2 + 1\right) \tag{A.8}$$

As discussed in section 3.2.2, scans over quadratic moduli revealed a 1D curve of low loss, whose corresponding $B(z)$ do not match any of the ones of eq. (A.7)–(A.8). Upon closer inspection, the low loss values are explained by the numerical closeness of the input moduli with ones corresponding to finite partial wave solutions with $L \neq 2$. For instance the modulus $B_2^c(z) = \frac{22}{29}z^2 + \frac{26}{29}$ has a loss $\mathcal{L}_S^E < 10^{-6}$ and is numerically within 0.3% of another modulus corresponding to the $L = 3$ solution

$$F_3(z) = \sin\delta_0 e^{i\delta_0} + 3z \sin\delta_1 e^{i\delta_1} + \frac{5}{2}(3z^2 - 1)\sin\delta_2 e^{i\delta_2} + \frac{7}{2}(5z^3 - 3z)\sin\delta_3 e^{i\delta_3}\,. \tag{A.9}$$

For the best fit values of $\delta_0 = 2.051, \delta_1 = 0.4578, \delta_2 = -3.131, \delta_3 = -3.128$, we have an associated $B_3(z)$ that is numerically close to $B_2^c(z)$ which we display on the figure 18(a). We refine the phase learned during our quadratic scans by letting the network run for 5000

JHEP05(2024)200



(a) $B(z)$ moduli.

(b) $\phi(z)$ phase.

Figure 18. Comparison between the exact $L = 3$ finite partial wave solution and the machine-learned one associated with the quadratic input modulus $B_2^c(z)$. On the left panel, we display the numerical closeness of the two moduli and on the right panel we compare the exact $L = 3$ phase and the machine learned one associated with the input $B_2^c(z)$.

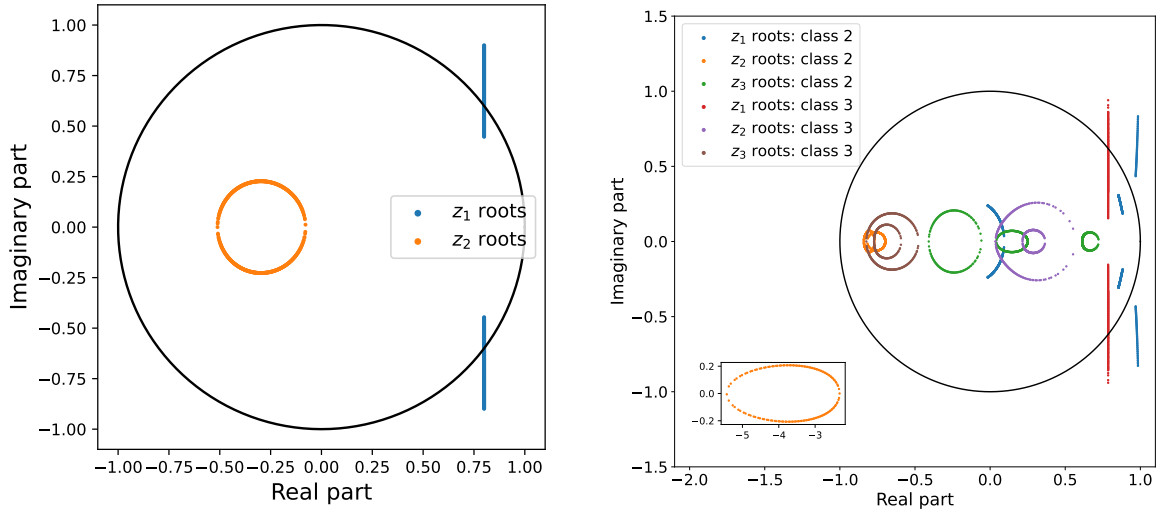
epochs with $B_2^c(z)$ as input. We then compare the resulting phase to the exact $L = 3$ solution as shown in figure 18(b). The agreement between the learned phase and the finite $L = 3$ solution explains why the associated loss value is so low for this spurious solution. Along the 1D curve of low loss, all of the factious solutions found have the same property in that their quadratic modulus is well approximated by a finite partial wave solution with $L \neq 2$. It is to be noted that the finite partial wave solutions of figure 18(b) look qualitatively different from the infinite partial wave solutions found in the $\sin\mu < 1$ region, with one example displayed in the bottom right panel of figure 2.

B Phase shift ambiguities with finite partial waves

Whereas the main convergence region of the algorithm by Atkinson et al. is limited to $|z_1| > 1$ for solutions with a large number of non-zero partial waves, it can also resolve a select few solutions within the unit circle, corresponding to true finite partial wave amplitudes. Finite partial wave ambiguities can be resolved exactly when the number of partial waves is small, as has been described in the literature for $L = 2, 3, 4$ [20–22]. In particular, the $L = 2$ ambiguous solutions are all of the type described by ref. [23]. At $L = 2$ the two ambiguous amplitudes $F(z)$ and $\tilde{F}(z)$ are polynomials of order 2 and possess one common root z_2. The other root is distinct and is respectively z_1 and $z_1^\star$. Since the difference $F(z) - (-\tilde{F}^\star(z))$ is a linear polynomial in z, the amplitudes correspond to genuine class 2 solutions. We represent in figure 19(a) the roots associated with these ambiguous amplitudes, noticing that we have the real part of z_1 that precisely equates $\frac{4}{5}$. As expected we have recovered solutions both outside the z_1 unit circle but also inside of it. All of these solutions can be recovered by both the classical descending algorithm and our machine learning implementation.

At $L = 3$ we have two different families of ambiguous amplitude solutions which share two roots z_2, z_3 and differ only by a single root, z_1 and $z_1^\star$. One family has $\mathrm{Im}(z_2 + z_3) = 0$

JHEP05(2024)200



(a) $L = 2$. (b) $L = 3$.

Figure 19. Roots of the ambiguous polynomial ambiguities at $L = 2, 3$. We only represent solutions where the two ambiguous amplitudes have a single distinct root, respectively z_1 and $z_1^\star$. At $L = 2$ the solutions are all of Ref's. [23] class 2, as described in section 5.1. At $L = 3$ the solutions can be class 2 or class 3.

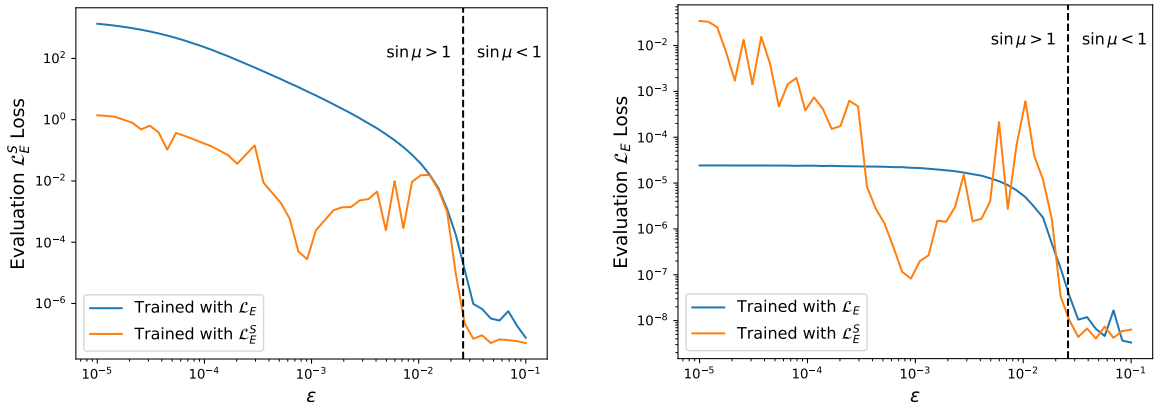
and the other family has $\mathrm{Im}(z_2 + z_3) \neq 0$. In the first case the difference $F(z) - (-\tilde{F}^\star(z))$ is a linear polynomial in z, a class 2 solution, while in the second case, the same difference is a quadratic polynomial, hence a class 3 solution. The first family is resolved by the original implementation of the descending algorithm while the second family would require a shooting method that aims at finding $\gamma_2(C) = 0$ for γ_2 appearing in eq. (5.3). Both family and their respective roots are displayed in figure 19(b), where notably we have a plethora of points lying within $|z_1| < 1$.

C Scaled and non-scaled losses

The difference between using the non-scaled or scaled losses of respectively eq. (2.1) and eq. (2.2) becomes apparent when studying edge cases. One example of interest concerns differential cross sections that are almost vanishing at a particular z value, making $\sin\mu$ blow up. The $B(z)$ term in the denominator of eq. (2.2) makes the whole loss expression close to singular in that case. A simple example to probe this is to take $B(z) = z^2/2+\epsilon$. The differential cross section is positive but almost vanishes at $z = 0$, such that $\sin\mu = (60\epsilon^2 + 20\epsilon + 1)/(60\epsilon)$. For $\epsilon < (10 - \sqrt{85})/30$ we have $\sin\mu > 1$ and the existence of a solution is not guaranteed. In particular, the classical fixed point iterative scheme of [15] does not converge.

We can compare how the choice of the loss function plays a role in this edge case scenario, by training different neural networks using either eq. (2.1) or eq. (2.2) as a loss function. We create a series of different ϵ values, distributed in $\epsilon \in [10^{-5}, 10^{-1}]$, and train a neural network for 2000 epochs at each point, using either loss function. We show in figure 20(a) the scaled loss $\mathcal{L}_E^S$ at evaluation and in figure 20(b) the non-scaled loss $\mathcal{L}_E$ at evaluation. In the $\sin\mu < 1$ region the networks trained using the scaled loss (orange curves) perform better on

JHEP05(2024)200



(a) Scaled loss of eq. (2.2) at evaluation.

(b) Non-scaled loss of eq. (2.1) at evaluation.

Figure 20. Training on the input modulus $B(z) = az^2 + \epsilon$ using either a scaled (orange) or a non-scaled (blue) loss function. The panels compare the different loss metrics at evaluation time. The dashed black line indicates the transition between moduli with $\sin\mu < 1$ and moduli with $\sin\mu > 1$.

both evaluation metrics and accurately recover the phase solutions. In the $\sin\mu > 1$ region the networks trained with a scaled loss also perform better overall. In particular, we observe a dip in the evaluation losses around $\epsilon \sim 10^{-3}$, where $\mathcal{L}_E^S < 10^{-4}$ and $\mathcal{L}_E < 10^{-7}$. For the networks trained with the non-scaled loss (blue curves), no such dip is observed and instead, the scaled loss at evaluation blows up when $\epsilon < 10^{-2}$, as shown in figure 20(a).

To understand this point better we can plot the learned phases at a few specific values of ϵ. In figure 21(a) and figure 21(b) we plot the phases learned at respectively $\epsilon = 0.04715$ and $\epsilon = 0.00091$ where the former corresponds to a point in the $\sin\mu < 1$ region and the latter to the dip in the $\sin\mu > 1$ region. For $\sin\mu < 1$, both phases are identical and both networks properly resolve the phase solution. However, as $\sin\mu > 1$, the phases learned by the two networks become drastically different. The networks trained on the base loss $\mathcal{L}_E$ will learn a simple deformation of the phase solution while the networks trained on $\mathcal{L}_E^S$ will learn a brand new phase shape. The dip in the evaluation losses that we observed can be explained by the numerical proximity of the phase to a genuine finite partial wave solution, as discussed in appendix A. This feature will only be able to be captured by the networks trained using the scaled loss. It is important to mention however that in most circumstances (including for physical differential cross sections) we do not expect $B(z)$ to be close to vanishing and thus both training losses will perform similarly.

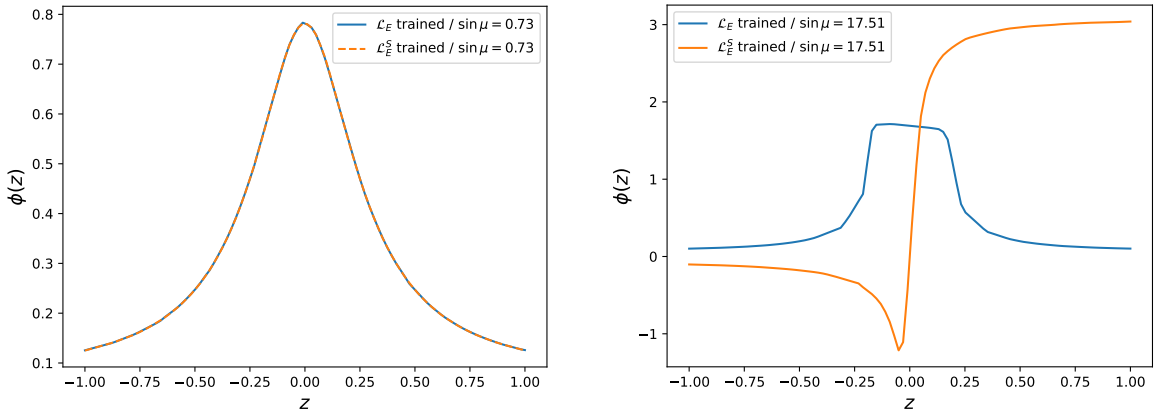
D Simple dual bounds

Let us consider the following problem: can a given function $B(z)$ be an elastic differential cross-section? One obvious requirement is that $B(z) \geq 0$ but there are more constraints.

Let us show that not any $B(z)$ can arise as a differential cross-section. We consider the following integral

$$\int_{-1}^{1} dz F(z)F^*(z) = \int_{-1}^{1} dz B(z)^2. \tag{D.1}$$

JHEP05(2024)200



(a) Low $\sin\mu$ value.

(b) High $\sin\mu$ value.

Figure 21. Learned phases for the input modulus $B(z) = z^2/2 + \epsilon$ where networks are trained using either a scaled (orange) or non-scaled (blue) loss function. On the left panel we use $\epsilon \sim 0.047$ where $\sin\mu < 1$ and on the right panel we use $\epsilon \sim 9.1 \times 10^{-4}$ where $\sin\mu > 1$. On the right panel, the network trained with the non-scaled loss does not lead to a physical phase.

By plugging the partial wave expansion for $F(z)$ into the integral we get

$$\int_{-1}^{1} dz F(z)F^*(z) = 2\sum_{\ell=0}^{\infty}(2\ell+1)|f_\ell|^2 = 2\sum_{\ell=0}^{\infty}(2\ell+1)\mathrm{Im}f_\ell = 2\mathrm{Im}F(1), \tag{D.2}$$

where we used elastic unitarity $\mathrm{Im}f_\ell = |f_\ell|^2$. Using the fact that $B(1) = |F(1)| \geq \mathrm{Im}F(1)$ we thus get the simplest constraint

$$2B(1) \geq \int_{-1}^{1} dz B(z)^2, \tag{D.3}$$

which also immediately follows from considering elastic unitarity equation at $z = 1$.

Consider next the next to simplest integral with spin one Legendre polynomial

$$\int_{-1}^{1} dz P_1(z)F(z)F^*(z) = \int_{-1}^{1} dz P_1(z)B(z)^2. \tag{D.4}$$

This time we get a product of three Legendre polynomials which produces the Wigner 3j-symbol. It is only non-zero when $\ell - \ell' = \pm 1$.

Plugging the explicit expression we get

$$\begin{aligned}\int_{-1}^{1} dz P_1(z)F(z)F^*(z) &= \sum_{\ell=0}^{\infty} 2(\ell+1)\left(f_{\ell+1}f_\ell^* + f_{\ell+1}^* f_\ell\right) \\ &= 2\sum_{\ell=0}^{\infty} 2(\ell+1)\left(\mathrm{Im}f_{\ell+1}\mathrm{Im}f_\ell + \mathrm{Re}f_{\ell+1}\mathrm{Re}f_\ell\right).\end{aligned} \tag{D.5}$$

Consider next the following sum

$$\sum_{\ell=0}^{\infty} 2(\ell+1)\left(|f_{\ell+1}|^2 + |f_\ell|^2 - 2(\mathrm{Im}f_{\ell+1}\mathrm{Im}f_\ell + \mathrm{Re}f_{\ell+1}\mathrm{Re}f_\ell)\right) \geq 0\ . \tag{D.6}$$

JHEP05(2024)200

We then notice that using elastic unitarity

$$\sum_{\ell=0}^{\infty}(\ell+1)\left(|f_{\ell+1}|^2+|f_\ell|^2\right)=\sum_{\ell=0}^{\infty}(2\ell+1)\mathrm{Im}f_\ell=\mathrm{Im}F(1). \tag{D.7}$$

As before, using the fact that $B(1)=|F(1)|\geq \mathrm{Im}F(1)$, we thus get from (D.6) (and its analog where we flip sign in front of the second term)

$$2B(1)\geq\int_{-1}^{1}dzP_1(z)B(z)^2\geq -2B(1). \tag{D.8}$$

This bound is correct but it is trivially satisfied given (D.3).

To get better bounds we need to put more constraints. Imagine that we know that $B(z)^2$ is a polynomial of a maximal degree N. We can then consider zero projections

$$\int_{-1}^{1}dzP_{\ell>N}(z)B(z)^2=0. \tag{D.9}$$

We can try to add these zero projections to the argument above. Consider for example the spin three zero projection

$$\int_{-1}^{1}dzP_3(z)B(z)^2=0 \tag{D.10}$$

$$\sum_{\ell=0}^{\infty}d_{1,\ell}\left(f_{\ell+1}f_\ell^*+f_{\ell+1}^*f_\ell\right)+\sum_{\ell=0}^{\infty}d_{3,\ell}\left(f_{\ell+3}f_\ell^*+f_{\ell+3}^*f_\ell\right)=0, \tag{D.11}$$

where in the second line we rewrote it in terms of partial waves.

We can now derive bounds by considering the following positive semi-definite problem, see e.g. [53] for a similar analysis in the case of dispersion relations,

$$c_\pm\,\mathbb{I}\pm T+\sum_i n_iN_i\succcurlyeq 0, \tag{D.12}$$

where N_i are zero projection matrices $f^*N_if=0$, T is the target quantity that we want to bound $t=f^*Tf$, and $\mathbb{I}$ is the diagonal matrix with elements being $2\ell+1$.

If we have found $c_\pm$ and n_i such that the matrix above is positive semi-definite, we get the bound

$$-c_-B(1)\leq t\leq c_+B(1)\ . \tag{D.13}$$

Numerically, this can be done first by truncating in spin, and then extrapolating the cut-off to infinity.

Implementing the spin-3 zero projection constraint we get that

$$\text{Spin 3:}\quad 1.56B(1)\geq\int_{-1}^{1}dzP_1(z)B(z)^2\geq -1.56B(1), \tag{D.14}$$

which is an improvement of the previous bound. Similarly, we can consider the spin-four zero projection

$$\int_{-1}^{1}dzP_4(z)B(z)^2=0, \tag{D.15}$$

JHEP05(2024)200

and repeat the derivation above. This time we get the following bound

$$\text{Spin 4:} \quad 1.24B(1) \geq \int_{-1}^{1} dz P_2(z) B(z)^2 \geq -0.67B(1). \tag{D.16}$$

For polynomial cross-sections we have infinitely many null constraints which we could try to use to derive the dual bounds. These bounds must be satisfied and they do not depend on the existence of the actual solution.

JHEP05(2024)200

References

[1] D. Bessis and A. Martin, *A theorem of uniqueness*, *Nuovo Cim. A Series 10* **52** (1967) 719.

[2] R.F. Alvarez-Estrada, *On the construction of scattering amplitudes from experimental data and analyticity*, *Annals Phys.* **68** (1971) 196 [INSPIRE].

[3] A. Martin, *Construction of the scattering amplitude from the differential cross-sections*, *Nuovo Cim. A* **59** (1969) 131 [INSPIRE].

[4] A. Martin, *Reconstruction of Scattering Amplitudes From Differential Cross-Section*, *Les rencontres physiciens-mathématiciens de Strasbourg — RCP25* **20** (1974).

[5] K. Chadan, P.C. Sabatier and R.G. Newton, *Inverse Problems in Quantum Scattering Theory*, Springer Berlin Heidelberg (1989) [DOI:10.1007/978-3-642-83317-5].

[6] D. Atkinson, P.W. Johnson and R.L. Warnock, *Determination of the scattering amplitude from the differential cross-section and unitarity*, *Commun. Math. Phys.* **28** (1972) 133 [INSPIRE].

[7] G.R. Bart, P.W. Johnson and R.L. Warnock, *Continuum ambiguity in the construction of unitary analytic amplitudes from fixed-energy-scattering data*, *J. Math. Phys.* **14** (1973) 1558 [INSPIRE].

[8] D. Atkinson, G. Mahoux and F.J. Yndurain, *Construction of a unitary analytic scattering amplitude (i). scalar particles*, *Nucl. Phys. B* **54** (1973) 263 [INSPIRE].

[9] A. Gersten, *Ambiguities of complex phase-shift analysis*, *Nucl. Phys. B* **12** (1969) 537 [INSPIRE].

[10] A. Martin and J.-M. Richard, *New result on phase shift analysis*, *Phys. Rev. D* **101** (2020) 094014 [arXiv:2004.11156] [INSPIRE].

[11] M. Kruczenski, J. Penedones and B.C. van Rees, *Snowmass White Paper: S-matrix Bootstrap*, arXiv:2203.02421 [INSPIRE].

[12] M. Correia, A. Sever and A. Zhiboedov, *An analytical toolkit for the S-matrix bootstrap*, *JHEP* **03** (2021) 013 [arXiv:2006.08221] [INSPIRE].

[13] A. Martin, *Scattering Theory: Unitarity, Analyticity and Crossing*, vol. 3, Springer Berlin Heidelberg (1969) [DOI:10.1007/BFb0101043] [INSPIRE].

[14] R.G. Newton, *Determination of the amplitude from the differential cross section by unitarity*, *J. Math. Phys.* **9** (1968) 2050 [INSPIRE].

[15] D. Atkinson, *Introduction to the use of non-linear techniques in s-matrix theory*, *Acta Phys. Austriaca Suppl.* **7** (1970) 32 [INSPIRE].

[16] J.E. Bowcock and H. Burkhardt, *Principles and Problems of Phase Shift Analysis*, *Rept. Prog. Phys.* **38** (1975) 1099 [INSPIRE].

[17] A.D. Gangal and J. Kupsch, *Determination of the scattering amplitude*, *Commun. Math. Phys.* **93** (1984) 333 [INSPIRE].

[18] C. Itzykson and A. Martin, *Phase-shift ambiguities for analytic amplitudes*, *Nuovo Cim. A* **17** (1973) 245 [INSPIRE].

[19] J.H. Crichton, *Phase-shift ambiguities for spin-independent scattering*, *Nuovo Cim. A Series 10* **45** (1966) 256.

[20] D. Atkinson, P.W. Johnson, N. Mehta and M. De Roo, *Crichton's phase-shift ambiguity*, *Nucl. Phys. B* **55** (1973) 125 [INSPIRE].

[21] F.A. Berends and S.N.M. Ruijsenaars, *Examples of phase-shift ambiguities for spinless elastic scattering*, *Nucl. Phys. B* **56** (1973) 507 [INSPIRE].

[22] H. Cornille and J.M. Drouffe, *Phase-shift ambiguities for spinless and 4>=l(max) elastic scattering*, *Nuovo Cim. A* **20** (1974) 401 [INSPIRE].

[23] D. Atkinson, L.P. Kok and M. de Roo, *Crichton Ambiguities with Infinitely Many Partial Waves*, *Phys. Rev. D* **17** (1978) 2492 [INSPIRE].

[24] A. Butter et al., *The Machine Learning landscape of top taggers*, *SciPost Phys.* **7** (2019) 014 [arXiv:1902.09914] [INSPIRE].

[25] M. Paganini, L. de Oliveira and B. Nachman, *CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks*, *Phys. Rev. D* **97** (2018) 014021 [arXiv:1712.10321] [INSPIRE].

[26] S.-M. Udrescu and M. Tegmark, *AI Feynman: a Physics-Inspired Method for Symbolic Regression*, *Sci. Adv.* **6** (2020) eaay2631 [arXiv:1905.11481] [INSPIRE].

[27] P.-A. Kamienny, G. Lample, S. Lamprier and M. Virgolin, *Deep Generative Symbolic Regression with Monte-Carlo-Tree-Search*, arXiv:2302.11223.

[28] D.L.B. Sombillo, Y. Ikeda, T. Sato and A. Hosaka, *Classifying the pole of an amplitude using a deep neural network*, *Phys. Rev. D* **102** (2020) 016024 [arXiv:2003.10770] [INSPIRE].

[29] Joint Physics Analysis Center and JPAC collaborations, *Deep learning exotic hadrons*, *Phys. Rev. D* **105** (2022) L091501 [arXiv:2110.13742] [INSPIRE].

[30] C. Chen, H. Chen, W.-Q. Niu and H.-Q. Zheng, *Identifying hadronic molecular states with a neural network*, *Eur. Phys. J. C* **83** (2023) 52 [arXiv:2205.03572] [INSPIRE].

[31] K. Hornik, M. Stinchcombe and H. White, *Multilayer feedforward networks are universal approximators*, *Neural Networks* **2** (1989) 359 [INSPIRE].

[32] M. Raissi, P. Perdikaris and G.E. Karniadakis, *Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations*, *J. Comput. Phys.* **378** (2019) 686 [arXiv:1711.10561] [INSPIRE].

[33] K. Zubov et al., *NeuralPDE: Automating Physics-Informed Neural Networks (PINNs) with Error Approximations*, arXiv:2107.09443.

[34] L. Lu, X. Meng, Z. Mao and G.E. Karniadakis, *DeepXDE: A Deep Learning Library for Solving Differential Equations*, *SIAM Review* **63** (2021) 208.

JHEP05(2024)200

[35] L. Yuan, Y.-Q. Ni, X.-Y. Deng and S. Hao, *A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations*, *J. Comput. Phys.* **462** (2022) 111260.

[36] G. Pang, M. D'Elia, M. Parks and G.E. Karniadakis, *nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator. Algorithms and applications*, *J. Comput. Phys.* **422** (2020) 109760.

[37] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, *Adv. Neural Inf. Process. Syst.* **32** (2019) 8024 [arXiv:1912.01703] [INSPIRE].

[38] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [INSPIRE].

[39] M.F. Paulos et al., *The S-matrix bootstrap. Part III: higher dimensional amplitudes*, *JHEP* **12** (2019) 040 [arXiv:1708.06765] [INSPIRE].

[40] H. Chen, A.L. Fitzpatrick and D. Karateev, *Nonperturbative bounds on scattering of massive scalar particles in $d \geq 2$*, *JHEP* **12** (2022) 092 [arXiv:2207.12448] [INSPIRE].

[41] J. Elias Miro, A. Guerrieri and M.A. Gumus, *Bridging positivity and S-matrix bootstrap bounds*, *JHEP* **05** (2023) 001 [arXiv:2210.01502] [INSPIRE].

[42] A. Guerrieri and A. Sever, *Rigorous Bounds on the Analytic S Matrix*, *Phys. Rev. Lett.* **127** (2021) 251601 [arXiv:2106.10257] [INSPIRE].

[43] P. Tourkine and A. Zhiboedov, *Scattering amplitudes from dispersive iterations of unitarity*, *JHEP* **11** (2023) 005 [arXiv:2303.08839] [INSPIRE].

[44] M.D. Giovanni, D. Sondak, P. Protopapas and M. Brambilla, *Finding multiple solutions of odes with neural networks*, in *AAAI Spring Symposium: MLPS*, (2020).

[45] B. Ananthanarayan, G. Colangelo, J. Gasser and H. Leutwyler, *Roy equation analysis of $\pi\pi$ scattering*, *Phys. Rept.* **353** (2001) 207 [hep-ph/0005297] [INSPIRE].

[46] D. Arkinson, *A proof of the existence of functions that satisfy exactly both crossing and unitarity: I. Neutral pion-pion scattering. No subtractions.*, *Nucl. Phys. B* **7** (1968) 375 [INSPIRE].

[47] D. Atkinson, *A proof of the existence of functions that satisfy exactly both crossing and unitarity (ii) charged pions. no subtractions*, *Nucl. Phys. B* **8** (1968) 377 [INSPIRE].

[48] D. Atkinson, *A proof of the existence of functions that satisfy exactly both crossing and unitarity (iii). subtractions*, *Nucl. Phys. B* **13** (1969) 415 [INSPIRE].

[49] D. Atkinson, *A proof of the existence of functions that satisfy exactly both crossing and unitarity. iv. nearly constant asymptotic cross-sections*, *Nucl. Phys. B* **23** (1970) 397 [INSPIRE].

[50] P. Tourkine and A. Zhiboedov, *Scattering from production in 2d*, *JHEP* **07** (2021) 228 [arXiv:2101.05211] [INSPIRE].

[51] S. Caron-Huot et al., *Six-Gluon amplitudes in planar $\mathcal{N} = 4$ super-Yang-Mills theory at six and seven loops*, *JHEP* **08** (2019) 016 [arXiv:1903.10890] [INSPIRE].

[52] H.S. Hannesdottir, A.J. McLeod, M.D. Schwartz and C. Vergu, *Constraints on sequential discontinuities from the geometry of on-shell spaces*, *JHEP* **07** (2023) 236 [arXiv:2211.07633] [INSPIRE].

[53] S. Caron-Huot and V. Van Duong, *Extremal Effective Field Theories*, *JHEP* **05** (2021) 280 [arXiv:2011.02957] [INSPIRE].

JHEP05(2024)200