Published for SISSA by Springer

Received: *December 18, 2023*
Accepted: *March 8, 2024*
Published: *April 3, 2024*

Characterizing 4-string contact interaction using machine learning

Harold Erbin[a,b,c] **and Atakan Hilmi Fırat**[a,b]

[a]*Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge MA 02139, U.S.A.*
[b]*NSF AI Institute for Artificial Intelligence and Fundamental Interactions, U.S.A.*
[c]*Université Paris Saclay, CEA, LIST, Gif-sur-Yvette, F-91191, France*

E-mail: erbin@mit.edu, firat@mit.edu

Abstract: The geometry of 4-string contact interaction of closed string field theory is characterized using machine learning. We obtain Strebel quadratic differentials on 4-punctured spheres as a neural network by performing unsupervised learning with a custom-built loss function. This allows us to solve for local coordinates and compute their associated mapping radii numerically. We also train a neural network distinguishing vertex from Feynman region. As a check, 4-tachyon contact term in the tachyon potential is computed and a good agreement with the results in the literature is observed. We argue that our algorithm is manifestly independent of number of punctures and scaling it to characterize the geometry of n-string contact interaction is feasible.

Keywords: String Field Theory, Differential and Algebraic Geometry, Bosonic Strings

ArXiv ePrint: 2211.09129

https://doi.org/10.1007/JHEP04(2024)016

JHEP04(2024)016

Contents

1 Introduction

Despite possibly being one of the most intricate quantum field theory, our practical knowledge of closed string field theory (CSFT) and its possible solutions, yet alone our understanding of its quantum effects, is still limited after almost three decades from its initial formulation (for reviews see [1–4]). Although there were some attempts towards understanding the classical tachyon potential in the case of bosonic CSFT in the past [5–17], the closed string tachyon vacuum (or lack thereof) hasn't revealed itself entirely yet. This is primarily due to CSFT being a non-polynomial theory. For the lack of better formulation and the absence of analytical techniques that can overcome these difficulties,[1] the most straightforward way to progress appears to truncate the theory to some order/level and perform numerical computations. It is possible such approach will yield results sufficiently close to the results of the full theory, just like it does in open string field theory [25–34].

In the past, Nicolas Moeller has taken this approach to CSFT first by truncating classical bosonic CSFT (in the minimal-area parametrization) up to quartic order [8], then up to quintic order [13–15], and he numerically calculated the tachyon potential and its minimum by level truncation. Even though there has been some progress this way, the fate of closed

[1] Although see the recent developments in hyperbolic string field theory [18–24].

JHEP04(2024)016

string tachyon condensation in the bosonic CSFT remains unclear; indicating including terms from higher orders/levels (and understanding their interplay) may be necessary to produce precise results.

In order to truncate the classical bosonic CSFT up to sextic and higher orders, it is necessary to solve the geometry of string contact interactions on six-and higher-punctured spheres. This involves finding Strebel quadratic differentials,[2] obtaining their associated local coordinates to calculate off-shell string amplitudes and finally finding the relevant sub-region (so-called *vertex region*) in the moduli space of the punctured spheres where the moduli integration has to be performed.

Performing all of these numerically was feasible for four-and five-punctured spheres using classical numerical methods, such as Newton's method. However, they seem to fall short and become unfeasible when there are six or more punctures. The basic roadblock is that the equations one needs to solve begin depending on the shape of the so-called critical graph of Strebel differential, which is impossible to obtain without knowing the Strebel differential itself. This informs us that the numerical methods have to be modified in a way that the algorithm should produce Strebel differentials without referring to their critical graphs a priori.

In this paper we precisely do this by representing Strebel differentials on four-punctured spheres as a *artificial neural network*. We obtain such a network by performing *unsupervised learning* using a custom-built loss function which gets minimized when a quadratic differential is Strebel. Such loss function is, by construction, agnostic of the critical graph and this is the reason it overcomes the hurdle mentioned above. Machine learning algorithms, such as the one we use, have already found their place in string theory, from exploring the landscape of string theory [36–39] to obtaining information regarding Calabi-Yau manifolds [40–48] and its overall relation to quantum field theory [49–51]. In particular, we would like to point out that our algorithm has been partially inspired by the methods described in [46, 47]. For a recent review on the applications of data science to string theory see [52].

We note that four-punctured sphere is the first case leading to a non-trivial string contact interaction and reemphasize that it has already been solved by Moeller in [8]. Here we just use it as a test ground for our ideas. Even so, since we obtain Strebel differentials as a neural network, our approach is an improvement as once the network is properly trained it can be used to find the Strebel differential for any four-punctured sphere practically *immediately*. It is also philosophically different from Moeller's approach as we are solving for the function itself, instead of just finding the solutions at specific moduli. While one can use a (polynomial) fit to approximate the function, neural networks are more flexible and expressive. In particular, they may be used for non-parametric regression [53, 54] and they can extrapolate outside the training region [55, 56]. Let us further note that Moeller stressed that he did not succeed in finding a simple fit in the case of 5-punctured spheres [13].

We make sure every step of our algorithm is manifestly independent of the number of punctures. In subsequent work [57], we plan to characterize the string contact interactions for higher-punctured spheres, where benefits of constructing the algorithm this fashion would be apparent. Once the Strebel differential is obtained, we can find the local coordinates by expanding it around the punctures. This alone doesn't specify the so-called mapping radii, but

[2]Quadratic differentials are reviewed in section 2. For a comprehensive mathematical account, see [35].

JHEP04(2024)016

one can easily solve it for by numerically evaluating a specific integral [5, 8]. In this step we make an observation which renders this calculation independent of the critical graph as well.

On top of the local coordinates, we also need to solve for the region $\mathcal{V}_{0,n} \subset \mathcal{M}_{0,n}$ where the moduli integration has to be performed. Here $\mathcal{M}_{0,n}$ is the moduli space of n-punctured spheres, while $\mathcal{V}_{0,n}$ is so-called *vertex region*, implicitly determined by taking the lengths of *all* non-contractible curves greater than or equal to certain value in the metric associated with Strebel differential [1, 58–61], which we take 2π by convention.[3] These lengths can be computed given Strebel differential and once computed, we can generate a dataset to train a neural network to distinguish punctured-spheres in $\mathcal{V}_{0,n}$ from those outside. That is, we can train a network for *the indicator function* that outputs 1 if the surface is part of the vertex region and 0 otherwise. In this work we obtain the indicator function $\Theta_{0,4} : \mathcal{M}_{0,4} \simeq \mathbb{C} \setminus \{0,1\} \to \{0,1\}$ defined by,

$$\Theta_{0,4}(\xi,\xi^*) = \begin{cases} 1 & \text{if} \quad \xi \in \mathcal{V}_{0,4} \\ 0 & \text{if} \quad \xi \notin \mathcal{V}_{0,4} \end{cases}, \tag{1.1}$$

as a neural network. Here ξ denotes the moduli. This allows us to replace

$$\int_{\mathcal{V}_{0,4}} \to \int_{\mathcal{M}_{0,4}} \Theta_{0,4}(\xi,\xi^*)\,, \tag{1.2}$$

and it simplifies the moduli integration in practical terms by eliminating the need for describing the region $\mathcal{V}_{0,4}$ explicitly. We argue an analogous construction for the indicator function would work for higher-punctured spheres and it would be particularly superior, especially in the view of Monte-Carlo integration may be required for higher dimensional moduli integrals.

Lastly, we test our algorithm by computing the coefficient of the 4-tachyon contact interaction. In the conventions of [8], we report this value to be $v_4 = 72.396$. Comparing with values obtained based on various different techniques, we observe that our results are consistent with those in the literature, see table 1 for summary. This supports the validity of our algorithm.

The rest of the paper is organized as follows. In section 2, we review Strebel differentials, local coordinates, mapping radii, and the indicator function. In particular, we analytically solve Strebel differentials on 4-punctured spheres when all punctures are real and introduce a loss function for Strebel differentials which forms the central part of our algorithm. In section 3, we describe the specifics of our neural networks and their training. In particular we show that the first trained network has learned the relevant symmetries of the Strebel differential and produced the analytic results correctly. Moreover, we show that our results are consistent with the fits provided by Moeller in [8]. In the last section we conclude our paper and discuss possible future directions, especially we argue that scaling the algorithm to higher-punctured spheres is expected to be feasible. In appendix A and B, we provide some details on numerical evaluations in our work.

[3] We only focus on surfaces of genus 0, i.e. punctured spheres, so that this definition for vertex region can be used. In other words, we only consider *classical* CSFT.

JHEP04(2024)016

Average NN (Trapezoid)	72.320 ± 0.146
Best NN (Trapezoid)	72.396
Best NN (Monte-Carlo)	72.366 ± 0.096
Belopolsky (1994) [6]	72.39
Moeller (2004) [8]	72.390 ± 0.003
Yang & Zwiebach (2005) [12]	72.414

Table 1. Values for the 4-tachyon contact interaction v_4. The first line ("Average NN") represents the value obtained by 4 networks (of each type) and computing the integral using the trapezoid method. The second and third lines ("Best NN") show the value for the best result using the trapezoid and Monte Carlo methods. We also add the results in the literature obtained by various authors. The computations are explained in section 3.

2 The geometry of string contact interaction

In this section we review Strebel differentials, local coordinates, mapping radius, and the indicator function. For more details, reader can refer to [5, 6, 8, 35]. The novel features are complete analytic characterization of Strebel differentials for 4-punctured sphere when all punctures lie on a great circle and introduction of a loss function for Strebel differentials as well as few simplifying observations on the calculation of mapping radii.

2.1 Strebel quadratic differential

Imagine a n-punctured sphere $\Sigma_{0,n}$, with punctures placed at $P = \{\xi_1, \ldots, \xi_n\}$ assuming none of the punctures are at infinity. We always fix the positions of the last three punctures to pre-determined positions by appropriate PSL(2,$\mathbb{C}$) transformations. We are interested in quadratic differentials that have a double pole at each puncture with residue equal to -1.[4] In general they can be written as

$$\varphi = \phi(z)dz^2 = \sum_{i=1}^{n} \left[\frac{-1}{(z-\xi_i)^2} + \frac{c_i}{z-\xi_i} \right] dz^2 \,, \tag{2.1}$$

where $c_i \in \mathbb{C}$, $i = 1, \ldots n$ are a priori undetermined variables which we call *accessory parameters*. These are unconstrained and they are only going to be fixed upon demanding (2.1) to be a special type of quadratic differential, Strebel differential. The double pole structure with residues equal to -1 can be motivated by demanding that the metric associated with the quadratic differential $ds = \sqrt{|\phi(z)|}|dz|$ (so-called $\varphi-$metric) is that of a flat cylinder of circumference 2π when sufficiently close to a puncture [35]. The flat cylinders here correspond to external strings.

Given we have the punctures at $P = \{\xi_1, \ldots, \xi_n\}$, the point at infinity $z = \infty$ has to be regular in general for string contact interactions. Inverting the coordinates by $w = 1/z$ it is easy to see that the quadratic differential φ takes the following form around $z = \infty$ (or $w = 0$)

$$\varphi = \frac{1}{w^4}\phi\left(\frac{1}{w}\right) dw^2 = \sum_{i=1}^{n} \left[\frac{c_i}{w^3} + \frac{-1 + c_i\xi_i}{w^2} + \frac{-2\xi_i + c_i\xi_i^2}{w} + \mathcal{O}(w^0) \right] dw^2 \,. \tag{2.2}$$

[4] The residues can be chosen different from each other in principle as long as they are still negative, but this situation is not relevant for the current formulation of CSFT.

JHEP04(2024)016

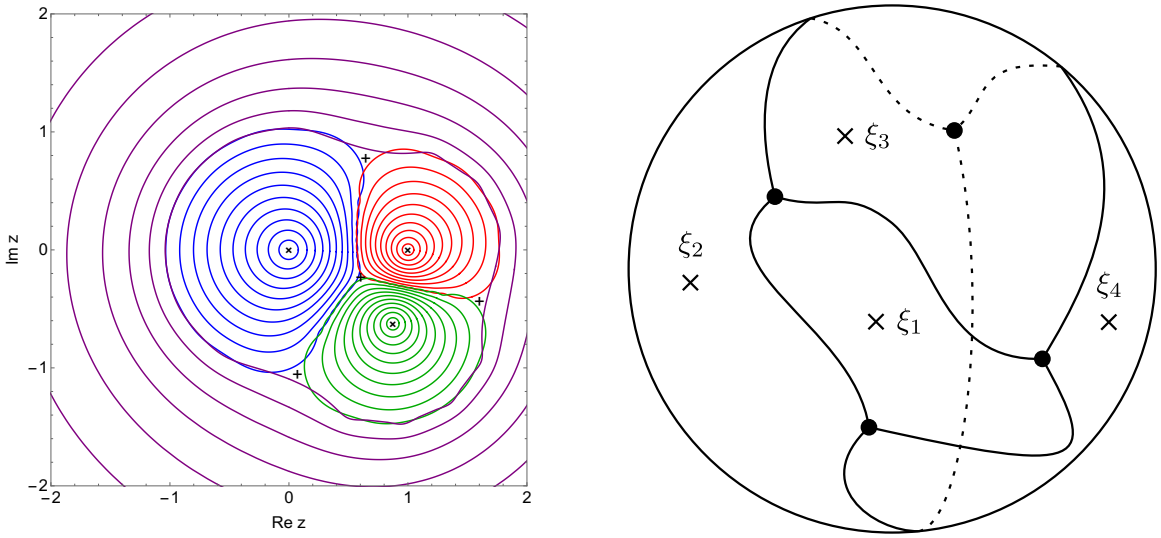

Figure 1. The trajectory structure of the Strebel differential when punctures are at $P = \{0, 1, 0.8734 - 0.6242i, \infty\}$ (left). We marked the positions of punctures and zeros by crosses and plusses respectively. The inaccuracies around the zeros are due to evaluating the trajectories as an expansion after (2.18). The critical graph is a tetrahedron whose sketch on $\mathbb{CP}^1$ is shown on the right.

This leads to following three linear conditions among accessory parameters c_i

$$\sum_{i=1}^{n} c_i = 0\,, \qquad \sum_{i=1}^{n}(-1 + c_i\xi_i) = 0\,, \qquad \sum_{i=1}^{n}(-2\xi_i + c_i\xi_i^2) = 0\,. \tag{2.3}$$

Notice this still leaves us with $n-3$ undetermined accessory parameters c_i. Also notice this explains why we haven't included regular terms in (2.1): it just leads to more singular terms around $z = \infty$.

Notice there are no undetermined accessory parameters when $n = 3$. If we place the punctures at $P = \{0, 1, \infty\}$, we find the quadratic differential (2.1) to be

$$\phi(z) = \left[\frac{-1}{z^2} + \frac{-1}{(z-1)^2} + \frac{-1}{z} + \frac{1}{z-1}\right] = -\frac{z^2 - z + 1}{z^2(z-1)^2} = -\frac{(z - (-1)^{1/3})(z - (-1)^{-1/3})}{z^2(z-1)^2}\,. \tag{2.4}$$

It is well known that such differential leads to Witten's vertex for closed strings [62, 63].

Before we introduce Strebel differentials, let us introduce some nomenclature. Define a *horizontal trajectory* as a path such that $\varphi > 0$ along it. A *critical trajectory* is a horizontal trajectory that begins and ends on a zero of φ. It is easy to argue that $n+2$ critical trajectories would emanate from a n-th order zero, and that the orders of the zeros would add up to $2n-4$ [35]. The union of critical trajectories of φ, together with their endpoints, is called the *critical graph*. A *Strebel differential* is then defined as a quadratic differential with double poles of residue -1 whose critical graph forms a non-empty measure zero set and is connected. Horizontal trajectories of such a differential foliate the entire surface [35]. An example of the trajectory structure of a Strebel differential is shown in figure 1.

The Strebel differential exists and is unique for every punctured sphere (see Theorem 23.5 in [35]). From this, and the fact that double poles with negative residues give closed

JHEP04(2024)016

horizontal trajectories sufficiently close to the punctures, there exists a set of accessory parameters, unique up to relations given in (2.3), such that the quadratic differential (2.1) is Strebel given punctures at P. Our goal is to find such accessory parameters c_i as a function of the position of punctures.

We remark that $\varphi-$metric associated with Strebel differential φ is the metric of minimal-area and it looks like flat cylinders grafted to each other dictated by the critical graph of φ [1, 35, 58, 59]. However, CSFT in minimal-area parametrization actually demands solving for minimal-area metrics for which the lengths of *all* non-contractible curves are greater than or equal to 2π for consistency. While the latter condition certainly fails for Strebel differentials when surfaces are sufficiently close to degeneration, for surfaces that are part of the classical elementary interaction (that is, those in the vertex region $\mathcal{V}_{0,n}$) this condition is satisfied by definition. So after finding Strebel differential everywhere on the moduli space $\mathcal{M}_{0,n}$ (*or* in any region containing $\mathcal{V}_{0,n}$), it is possible to map out the vertex region $\mathcal{V}_{0,n}$ by checking the lengths of non-contractible curves. Furthermore, once Strebel differential is known it is possible to find the local coordinates characterizing the geometry of n-string contact interaction. We explain these in subsequent subsections.

We finally note that for the quadratic differentials on the surfaces outside the vertex region, that is, those in the so-called *Feynman region* $\mathcal{F}_{0,n} = \mathcal{M}_{0,n} \setminus \mathcal{V}_{0,n}$, Strebel differential is not the right type of quadratic differential from the perspective of CSFT: we have to use a quadratic differential of the form (2.1) whose associated metric is of minimal-area under the condition that the lengths of all of its non-contractible curves is greater than or equal to 2π.[5] We call such differentials *Zwiebach differentials* (see Theorem 3.2 in [59]). In the vertex region $\mathcal{V}_{0,n}$ the definition of Zwiebach differentials coincides with the definition of Strebel differential, but in the Feynman region $\mathcal{F}_{0,n}$ they differ: the critical graph of Zwiebach differential becomes disconnected [59]. Geometrically this corresponds having internal cylinders corresponding to string propagators. In fact, Zwiebach differentials are examples of more general type of differentials called *Jenkins-Strebel (JS) differential* for which the critical graph forms a non-empty measure zero set but not necessarily connected. Zwiebach differentials can be shown to exist and be unique (see Theorem 5.1 in [59]). We emphasize that the accessory parameters for Strebel and Zwiebach differentials are distinct functions of the moduli in the Feynman region $\mathcal{F}_{0,n}$. In this study, we only consider Strebel differentials.[6]

2.2 Complex length and loss function

It is hard to work with the definition of Strebel differential given in previous subsection. Here we provide an equivalent characterization more amenable to analytical and numerical investigations. Begin with defining *the complex length* between zeros z_i, z_j of φ as

$$\ell(z_i, z_j) \equiv \int_{z_i}^{z_j} \sqrt{\phi(z)} dz \,. \tag{2.5}$$

[5] Even though Strebel differentials in Feynman region do not seem to be relevant for CSFT, we would like to point out that it has found applications in worldsheet approaches to AdS/CFT correspondence, see [64–67].

[6] Although see section 4 for the discussion on how ideas here can be extended to obtain Zwiebach differential as a neural network.

JHEP04(2024)016

The path of integration here is chosen so that it avoids any branch cuts. Since the branch structure of $\sqrt{\phi(z)}$ is hard to keep track numerically, we are going to replace the square root with the *continuous square root* $\sqrt[\pm]{}$, just like in [8]. That is, we are going to define the domain of square root in the double cover of complex plane without the origin, where it is holomorphic. This would make the overall sign of the continuous square root, hence complex length, ambiguous. But, as we shall see, the problems of this sort would be of technical natural and they can be easily overcome. More details on numerical evaluation of continuous square root are given in appendix A.

With such replacement the complex length is now taken to be

$$\ell(z_i, z_j) \equiv \int_{z_i}^{z_j} \sqrt[\pm]{\phi(z)} dz \,. \tag{2.6}$$

Assuming the integrand doesn't vanish on the path of integration and the path is non self-intersecting, the integrand is holomorphic around some neighborhood of the path and the integral is equal to (2.5) up to overall sign (with appropriate choice of branch cut for $\sqrt{}$). Hence we can deform the path of integration for (2.6) freely without changing the value of the integral as long as the endpoints are fixed, the path doesn't cross any punctures and/or intersects itself. Therefore, for convenience, we evaluate the integral in (2.6) always on the straight line

$$z(t) = \frac{z_j + z_i}{2} + \frac{z_j - z_i}{2} t \quad \text{for} \quad t \in [-1, 1] \,. \tag{2.7}$$

The details on numerical evaluations the complex length can be found in appendix A.

But notice, regardless of where the path lies relative to the punctures, the (absolute value of) imaginary part of the complex length would be the same. To see this, just note that deforming the path of integration over the puncture would pick up the residue of $\sqrt[\pm]{\phi(z)}$ and this is always purely imaginary by having a residue equal to -1 for the double poles (2.1). This makes the shift real and equal to 2π, leading to no change for the imaginary part of the complex length.

This reasoning implies the complex length is real for Strebel differentials, as we can deform the path of integration (2.7) to a critical trajectory between z_i, z_j, which has $\varphi \geq 0$. This makes the integrand equal to the line element $ds = \sqrt{|\phi(z)|}|dz|$ up to sign and manifestly real. We remark that the (absolute value of) complex length may not always give the geodesic distance between zeros z_i, z_j in the $\varphi-$metric for Strebel differentials due to the sign ambiguity and the placement of the punctures relative to the path of integration.

Above we essentially provide a necessary condition for a quadratic differential (2.1) to be Strebel: if φ is Strebel then the complex length is real between all zeros of φ. In fact the other direction is true as well. That is, if z_i, z_j are zeros of a quadratic differential of the form (2.1), $\phi(z_i) = \phi(z_j) = 0$, then we have

$$\varphi \text{ is Strebel} \iff \forall\, z_i, z_j,\ \mathrm{Im}(\ell(z_i, z_j)) = 0 \,. \tag{2.8}$$

In order to argue for the sufficient condition, we just have to show that the integrand of the complex length, $\sqrt[\pm]{\phi(z)}dz$, is real throughout some path among all zeros and then its square, $\phi(z)dz^2$ would define a Strebel differential as it is going to be a single-valued quadratic differential of the form (2.1) and its critical graph would be measure zero and connected.

JHEP04(2024)016

This is easy to accomplish, as we can deform the path between each zeros to the path that would make the integrand real by beginning from one zero and moving in the direction that sets the imaginary part to zero. Since $\mathrm{Im}(\ell(z_i, z_j)) = 0$ for all i, j, we would be guaranteed to hit another zero after this procedure and this makes $\phi(z)dz^2$ a Strebel differential.

The condition (2.8) gives an alternative formulation for Strebel differentials. In fact this is the condition solved by Moeller using Newton's method [8, 13, 15]. Observe that the existence and uniqueness of Strebel differentials translate to the existence and uniqueness of the solution of the equations in the right-hand side of (2.8) in terms of accessory parameters up to the relations in (2.3). Note that there are $\binom{2n-4}{2}$ distinct equations in the right-hand side of (2.8). However, it is actually sufficient to demand vanishing of $\dim(\mathcal{M}_{0,n}) = 2n-6$ imaginary parts of complex lengths by dimensional counting. This shows the set of equations in the right-hand side of (2.8) is in fact over-determined.

Now, define the following function of quadratic differentials of the form (2.1) motivated by the conditions in the right-hand side of (2.8)

$$\mathcal{L}_{0,n}(\varphi) \equiv \binom{2n-4}{2}^{-1} \sum_{i<j} (\mathrm{Im}\ \ell(z_i, z_j))^2 \,. \tag{2.9}$$

Here $i, j = 1, \ldots, 2n-4$ runs over the zeros of φ (accounting degeneracy) and the overall factor is for normalization. We call this function *loss function* for reasons that is going to be apparent in section 3. Observe this function can be *unambiguously* evaluated using the integral (2.6) and taking the path of integration to be the straight line (2.7), as we square each imaginary part such that their sign ambiguity becomes irrelevant. By construction we have $\mathcal{L}_n \geq 0$.

The loss function (2.9) has a *unique* global minimum as a function of accessory parameters c_i (up to relations (2.3)) at fixed positions of punctures ξ_i by the existence and uniqueness of Strebel differentials and its value is equal to zero. So, it is in principle possible to obtain Strebel differentials by minimizing the loss function (2.9) in the space of accessory parameters given the positions of punctures. This optimization problem is perfectly suited to machine learning and it is how we are going to approach finding Strebel differentials in section 3. In particular, the advantage of this approach is clear from the fact that the loss function constructed out of (2.1) is totally agnostic of the shape of critical graphs, which made the previous approaches to solving Strebel differential slightly convoluted as we mentioned.

It is an interesting question whether the loss function (2.9) has another extremum. Our experimental investigation in the case of 4-punctured sphere informs us that even if there is, it hasn't made an appearance in our algorithm. So we *assume* there is no other extremum of the loss function (2.9) for all intents and purposes. It may be interesting to rigorously establish this is the case.

2.3 Strebel differential on 4-punctured sphere

Since we are going to test our algorithm for 4-string contact interaction, let us focus on Strebel differentials on 4-punctured spheres more. Begin with placing punctures at $P = \{0, 1, \xi, \infty\}$ by performing $\mathrm{PSL}(2, \mathbb{C})$ transformation. We see ξ here is the moduli. Since there is single accessory parameter after solving the modified version of conditions in (2.2) when one of

JHEP04(2024)016

the punctures is at $z = \infty$, it can be shown that the quadratic differential (2.3) can be put into the following form:

$$\phi(z) = \frac{-z^4 + az^3 + (2\xi - (1+\xi)a)z^2 + a\xi z - \xi^2}{z^2(z-1)^2(z-\xi)^2} = -\frac{(z-z_1)(z-z_2)(z-z_3)(z-z_4)}{z^2(z-1)^2(z-\xi)^2}\,, \tag{2.10}$$

where $a = a(\xi, \xi^*)$ is the single accessory parameter and z_i, $i = 1, 2, 3, 4$ are the zeros of the quadratic differential.[7] As we have emphasized earlier, finding Strebel differential is equivalent to finding the function $a = a(\xi, \xi^*)$.

There are certain symmetries the accessory parameter $a = a(\xi, \xi^*)$ enjoys. These are:

- The involution symmetry. The complex conjugate of Strebel differential on the surface $\Sigma_{0,4}$ would be the Strebel differential for the conjugated Riemann surface $\Sigma^*_{0,4}$. That is, if the accessory parameter corresponding to the moduli ξ is a, then the accessory parameter corresponding to the moduli ξ^* is a^*. In particular this shows that we have $a \in \mathbb{R}$ for $\xi \in \mathbb{R}$. Clearly, similar symmetry holds for n-punctured spheres.

- PSL$(2, \mathbb{C})$ symmetries permuting $\{0, 1, \infty\}$. There are 6 such transformations but they are generated by the following two transformations

$$z \to z' = 1 - z \quad \text{and} \quad z \to z' = \frac{1}{z}\,, \tag{2.11}$$

  and the position of the puncture at $z = \xi$ as well as the quadratic differential changes accordingly. These transformations shouldn't change a quadratic differential being Strebel, so it can be shown performing (2.11) induces following transformations for the moduli and the accessory parameter of a Strebel differential

$$\xi \to 1 - \xi \implies a \to -a + 4 \quad \text{and} \quad \xi \to \frac{1}{\xi} \implies a \to \frac{a}{\xi}\,. \tag{2.12}$$

  These symmetries can be generalized to higher-punctured spheres in an obvious fashion.

We can use these symmetries to solve for the accessory parameter for certain values of ξ. For example, for $\xi = 1/2 + (\sqrt{3}/2)i = e^{i\pi/3}$ we have $1/\xi = 1 - \xi$, which shows

$$-a + 4 = \frac{a}{\xi} \implies a = \frac{4\xi}{1+\xi} = 2 + \frac{2i}{\sqrt{3}}\,. \tag{2.13}$$

In fact the symmetries fix the critical graph to be a regular tetrahedron whose sides have lengths equal to $2\pi/3$ in this case [6]. We can also find the Strebel differential when $\xi = 1/2$. The moduli satisfies $\xi = 1 - \xi$ so we have $a = -a + 4$, which shows $a = 2$. This subsequently shows $a = 4, 0$ for $\xi = 2, -1$ respectively. Furthermore, it is actually possible to find the Strebel differentials for the moduli between $0 < \xi < 1$. Our claim is $a = 4\xi$ for $0 < \xi < 1$.

In order to argue for this, recall that the critical graph of a Strebel differential on 4-punctured sphere has to be topologically a planar tetrahedron on z-plane in general [6, 60]–one

[7] In this work, we used analytical expressions for the zeros of quadratic differentials. Obviously, this wouldn't work in higher-punctured sphere, but we think this is just a technical issue which we aim to confront in our future work [57].

JHEP04(2024)016

example has been already shown in figure 1. Now recall that the critical graph in the complex conjugated surface has to be the mirror image of the original graph around the real axis by the involution symmetry. Combining these two facts and taking $\xi \in \mathbb{R}$, we see the mirror image of the critical graph has to be itself. However for planar tetrahedral graphs this is impossible: the graph should degenerate. That means at least two of the zeros coincide and the other two zeros coincide as well by the involution symmetry. Pair of double zeros zero emanates 4 critical trajectories each now.

In the language of quadratic differentials that means we now have a pair of double zeros that are complex conjugates of each other. That is, Strebel differential has to take the form

$$\varphi = -\frac{(z-z_1)^2(z-\bar{z}_1)^2}{z^2(z-1)^2(z-\xi)^2}dz^2\,, \tag{2.14}$$

when $\xi \in \mathbb{R}$. Comparing with the form in (2.10), solving for a, and using the fact that $a = 2$ when $\xi = 1/2$ it can be shown that $a = 4\xi$ for $0 < \xi < 1$ after some algebra. Using the symmetries in (2.12) we can further show $a = 4$ for $\xi > 1$ and $a = 0$ for $\xi < 0$.

The situation when punctures collide is more subtle. Demanding continuity of a suggests we should have $a = 0, 4$ for $\xi = 0, 1$. Indeed, if this is the case, we see the quadratic differentials become

$$\varphi = -\frac{dz^2}{(z-1)^2} \quad \text{and} \quad \varphi = -\frac{dz^2}{z^2}\,, \tag{2.15}$$

for $\xi = 0, 1$ respectively. But notice these are the quadratic differentials describing an infinite flat cylinders whose punctures are at $z = 1, \infty$ and $z = 0, \infty$ respectively. This is exactly what it should be expected from the degeneration of Strebel differentials when we take $\xi \to 0, 1$: the residue condition forces having a single cylinder. The situation at $\xi = \infty$ is similar to $\xi = 0$, only difference being that we have to perform this calculation after inversion $z \to 1/z$. In particular, it can be shown that the limit doesn't depend on which value of a we use (0 or 4). Summarizing, we find the following expression for the accessory parameter a when the moduli is real:

$$a(\xi = \xi^*) = \begin{cases} 0 & \xi \leq 0 \\ 4\xi & 0 \leq \xi \leq 1 \\ 4 & 1 \leq \xi \end{cases}. \tag{2.16}$$

The plot of this function is shown in figure 8.

With the accessory parameter is available for Strebel differentials when $\xi \in \mathbb{R}$, we can find the lengths of the sides of the critical graph as a function of the moduli. In order to do that recall the length of the geodesics homotopic to a puncture is always equal to 2π. Then, since the critical graph is degenerated in the way described above, we only need to calculate the length of a single side, ℓ, and the lengths of the other sides would be either equal to this or $2\pi - \ell$. We can find ℓ by carefully evaluating (2.6) using (2.10) with the

JHEP04(2024)016

accessory parameter a given in (2.16) for $\xi \in \mathbb{R}$. The result is

$$\ell\left(\xi=\xi^*\right)=\begin{cases}2\pi-4\arctan(\sqrt{-\xi}) & \xi\leq 0\\ 4\arctan\left(\sqrt{-1+\frac{1}{\xi}}\right) & 0\leq\xi\leq 1\\ 4\arctan(\sqrt{\xi-1}) & 1\leq\xi\end{cases}\quad\text{or}\quad\xi=\begin{cases}-\cot^2\left(\frac{\ell}{4}\right) & \xi\leq 0\\ \cos^2\left(\frac{\ell}{4}\right) & 0\leq\xi\leq 1\\ \sec^2\left(\frac{\ell}{4}\right) & 1\leq\xi\end{cases}. \tag{2.17}$$

It is clear that the limits make sense from this expression. For example for $\xi = 1/2$, we have $\ell = \pi$ and this can be alternatively argued by the symmetry of this case. In fact, notice 2ℓ and $4\pi - 2\ell$ are the lengths of the non-contractible geodesics non-homotopic to punctures and they are equal to 2π only when $\xi = -1, 1/2, 2$ and less than 2π otherwise. In other words, these are the only real moduli that are in the vertex region $\mathcal{V}_{0,4}$. This result is consistent with [60].

Before closing off this subsection, we note that our argument for degenerate Strebel differentials on four-punctured spheres may not generalize to *every* type of degeneration of higher-punctured spheres. This is mostly due to the symmetry of the critical graph when all (or some) moduli taken to be real may not be as restrictive as it does for the four-punctured spheres. Still, it may be possible to find exact solutions for specific type of degenerations. Since we don't need them immediately, we plan to investigate the degeneration behavior in more detail in our upcoming work [57].

2.4 Local coordinates, mapping radii, and vertex region

Calculating off-shell string amplitudes on any Riemann surface requires a choice of local coordinates up to an overall phase around the punctures [1]. Our case of interest, the local coordinates for the n-string contact interactions, can be obtained using Strebel differential on n-punctured spheres as they are described through how n flat semi-infinite cylinders are grafted at the critical graph of Strebel differential [5, 8, 58]. Following the conventions of [8], this means one can find n analytic maps h_i of the form ($i = 1, 2, \cdots, n$)

$$\begin{aligned} z = h_i(w_i) &= \xi_i + \rho_i w_i + \sum_{\alpha=2}^{\infty} d_{i,\alpha-1}(\rho_i w_i)^\alpha \\ &= \xi_i + (\rho_i w_i) + d_{i,1}(\rho_i w_i)^2 + d_{i,2}(\rho_i w_i)^3 + \cdots, \quad \text{with} \quad \rho_i,\, d_{i,\alpha} \in \mathbb{C},\, \alpha = 1, 2, \cdots, \end{aligned} \tag{2.18}$$

from the punctured disks $D_i = \{w_i \in \mathbb{C} \mid 0 < |w_i| < 1\}$ to the n-punctured sphere for which the Strebel differential takes the form it takes for the flat cylinders in w_i coordinates

$$\varphi = -\frac{dw_i^2}{w_i^2}, \tag{2.19}$$

and the unit circles $|w_i| = 1$ are mapped to its critical graph. Here w_i would be the local coordinates for the string contact interactions (sometimes called *natural coordinates*) for which vertex operators are inserted. It can be shown that such coordinates always exist [35].

Notice how we have organized the expansion (2.18). This was because of convenience: as the overall phase of the local coordinates is irrelevant for CSFT we chose $\rho_i \in \mathbb{R}$ without loss of generality and we defined the rest of the coefficients accordingly. Here $\rho_i = |dh_i(w_i)/dw_i|_{w_i=0}$ is called *mapping radius* associated with the puncture ξ_i.

JHEP04(2024)016

Our primary goal is to find the maps (2.18), i.e. to find $d_{i,\alpha}$ and ρ_i. Except for the mapping radius, the coefficients in the expansion (2.18) can be found by expanding the Strebel differential around the puncture $z = \xi_i$

$$\varphi = \left[-\frac{1}{(z-\xi_i)^2} + \frac{b_{i,-1}}{z-\xi_i} + b_{i,0} + b_{i,1}(z-\xi_i) + \cdots \right] dz^2 , \tag{2.20}$$

and setting it equal to (2.19), along with using the expansion for $z = h_i(w_i)$ in (2.18). Comparing term by term in w, we can solve the d coefficients in terms of the b coefficients. First few terms are

$$d_{i,1} = \frac{1}{2} b_{i,-1} , \tag{2.21a}$$

$$d_{i,2} = \frac{1}{16} (7 b_{i,-1}^2 + 4 b_{i,0}) , \tag{2.21b}$$

$$d_{i,3} = \frac{1}{48} (23 b_{i,-1}^3 + 28 b_{i,-1} b_{i,0} + 8 b_{i,1}) , \tag{2.21c}$$

Note that the b coefficients, therefore d coefficients, are determined by the accessory parameters c_i, so knowing the latter is sufficient to construct the maps $z = h_i(w_i)$ up to mapping radius. For example, we see $b_{i,-1} = c_i$, hence $d_{i,1} = c_i/2$.

Finding mapping radii associated with the punctures takes more effort. To that end, begin by writing the local coordinate around $z = \xi_i$ by equating (2.1) and (2.19) as [5]

$$w_i(z) = \exp\left(-i \int_{z_c}^{z} \sqrt[\pm]{\phi(z')} dz' \right) , \tag{2.22}$$

where z_c is some point on the critical trajectory *surrounding* the puncture $z = \xi_i$ and the path of integration here is the straight line from the puncture to $z = z_c$. The lower bound of the integral makes sure $|w_i| = 1$ when z lies on the critical trajectory surrounding the puncture, as the integral just evaluates to real number in this case by repeating the arguments made below (2.7). We implicitly adjust the choice of sign of the exponent to guarantee $|w_i| \leq 1$.

Here we make a small observation that has apparently gone unnoticed in the literature. We demanded z_c to lie on the critical trajectory *surrounding* the puncture $z = \xi_i$ above, but actually this can be relaxed and one can choose z_c to be lying *anywhere* on the critical graph. To argue for this, let z_c to be any point on the critical graph and notice the integral in the exponent of (2.22) can be deformed as shown in figure 2. But then the contribution to the integral from this "outside" part becomes real following a similar argument made below (2.7) and this just results in an irrelevant phase for the local coordinate. In particular, notice that we can take z_c to be *any* zero of the quadratic differential. With this choice, the dependence of the local coordinates (2.22) (and in extension, the mapping radii (2.23)) to the shape of the critical graph drops out.

Using (2.22) we can obtain an integral expression for the mapping radii. It is [5]

$$\log \rho_i = \lim_{\epsilon \to 0} \left(\mathrm{Im} \int_{\xi_i + \epsilon}^{z_c} \sqrt{\phi(z')} dz' + \log |\epsilon| \right) . \tag{2.23}$$

Note that such limit exists with our choice of sign in the exponent of (2.22) and assuming $\xi + \epsilon$ lies on the straight path from the puncture at $z = \xi$ to $z = z_c$. Details on numerical

JHEP04(2024)016

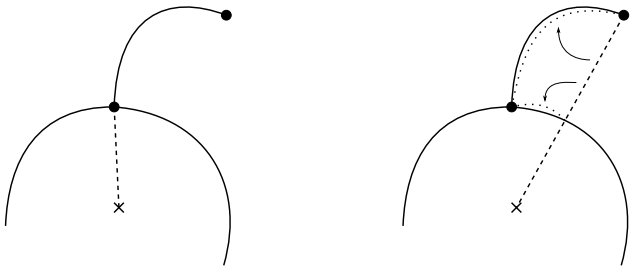

Figure 2. Part of the path of integration in (2.22) can be deformed to the dotted path. The integration over the dotted path produces a real number, resulting in an irrelevant phase for the local coordinates (2.22). In extension, this part also doesn't contribute to the mapping radii (2.23) below.

evaluation of such an integral are relegated to appendix A. In passing we note that it is possible to obtain local coordinates for 4-punctured spheres analytically when $\xi \in \mathbb{R}$. However, as we have stated earlier, these surfaces are not relevant from the perspective of CSFT, so we opt out reporting them here.

Calculating off-shell string amplitudes on Riemann surfaces not only requires a choice of local coordinates around the punctures, but also suitable choice of vertex region in the associated moduli space [1]. Instead of trying to describe the vertex region explicitly we can consider its associated indicator function. Such function is already defined in the introduction in (1.1) for the case of 4-punctured spheres. Here we give a general definition for the case of n-punctured sphere

$$\Theta_{0,n}(\xi,\xi^*) = \begin{cases} 1 & \text{if} \quad \xi \in \mathcal{V}_{0,n} \\ 0 & \text{if} \quad \xi \notin \mathcal{V}_{0,n} \end{cases} . \tag{2.24}$$

The criteria for $\xi \in \mathcal{V}_{0,n}$ is having *all* non-contractible curves in the φ-metric to be greater than or equal to 2π [1, 58–60]. For the case of φ-metric this means that it is sufficient to check the lengths of the critical trajectories separating 2 or more punctures from the rest, i.e. geodesics that are not homotopic to a puncture.

These lengths can be computed by finding the geodesic lengths between each zero of Strebel differential then combining them up suitably. Since the critical graph of a Strebel differential is an undirected graph and we can assign geodesic length to an edge of such graph, it is useful to arrange the associated data into (weighted) adjacency matrix M as follows:

$$M_{ij} = \begin{cases} \ell_{ij} & \text{when there is an edge of length } \ell_{ij} \text{ begins at } i\text{th zero and ends at } j\text{th zero} \\ 0 & \text{otherwise} \end{cases} . \tag{2.25}$$

Here M is a symmetric $(2n-4)\times(2n-4)$ matrix as there are $2n-4$ zeros of φ (including degeneracy). Elements in the diagonal are zero (hence M is traceless) and each row and column contains at most 3 non-zero elements as zeros of φ emit 3 critical trajectories for a generic moduli. Notice there are certain nonzero co-dimension loci in the moduli space where some zeros of φ may coincide. In this case we would set the elements of M corresponding to their connections to zero.

JHEP04(2024)016

Once such matrix is constructed, it is a simple matter to extract the length of all non-contractible curves: this is what we have done in the case of $n = 4$. But notice for the purposes of the indicator function (2.24) we just need to check the length of the shortest non-contractible geodesic and this can be found with relative ease given M, such as using Dijkstra's shortest path algorithm [68].

So the only really contention here is to find the lengths associated with each edge. This can be done by solving the critical trajectories and calculating their lengths. Recall a critical trajectory is a horizontal trajectory begins and ends on a zero of φ. So given a zero, it is possible to construct a critical trajectory emanating from it by taking small steps so that $\varphi > 0$ at each step, until we hit another zero. While we do this we can add the line elements $ds = \sqrt{|\phi(z)|}|dz|$ and that would generate the lengths, hence the adjacency matrix M, we are looking for.

In passing, we note that the condition of having all non-contractible curves greater than or equal to 2π is equivalent the length of each edge of the critical graph to be smaller than π for the case of 4-and 5-punctured spheres [60]. This fact has been exploited in previous works by Moeller [8, 13]. However this condition is not sufficient for higher-punctured spheres, so we opt out to use the generic method described above to make the algorithm manifestly independent of the number of punctures.

3 Neural networks for accessory parameter and indicator function

In this section, we describe the neural networks for the accessory parameter $a = a(\xi, \xi^*)$ and the indicator function $\Theta(\xi, \xi^*)$ in the case of 4-punctured sphere. We show the accessory parameter neural network has successfully learned the analytic behavior for the real moduli described in (2.16) and the symmetry properties described in (2.12). We emphasize these behaviors haven't been explicitly programmed into our neural network — they appear as a consequence of the training process. We additionally compare our result for the accessory parameter with the polynomial fit provided by Moeller [8] and observe a good agreement between our results.

Similarly, we test the indicator function neural network by plotting the vertex region $\mathcal{V}_{0,4}$ in the moduli space $\mathcal{M}_{0,4}$ and show our results are consistent with those in the literature [6, 8]. In particular, even though the indicator function neural network outputs values between 0 and 1, we observe a sharp transition from Feynman region to vertex region and it is almost always 1 or 0 — as it should be the case for the actual indicator function in (1.1).

Lastly, we compute the 4-tachyon contact term in the closed string tachyon potential by performing moduli integration over the vertex region using both trapezoid and Monte-Carlo methods. As mentioned in introduction, we get a good agreement with the results in the literature, providing extra support for our method based on machine learning.

3.1 Accessory parameter neural network

Artificial neural networks are computing systems inspired by biological neural networks. They consist of number of *layers* and each layer consists of number of *nodes*. A node in a given layer is connected to the nodes in the previous and subsequent layer. An example of a neural network we consider in this work is shown in figure 3.

JHEP04(2024)016

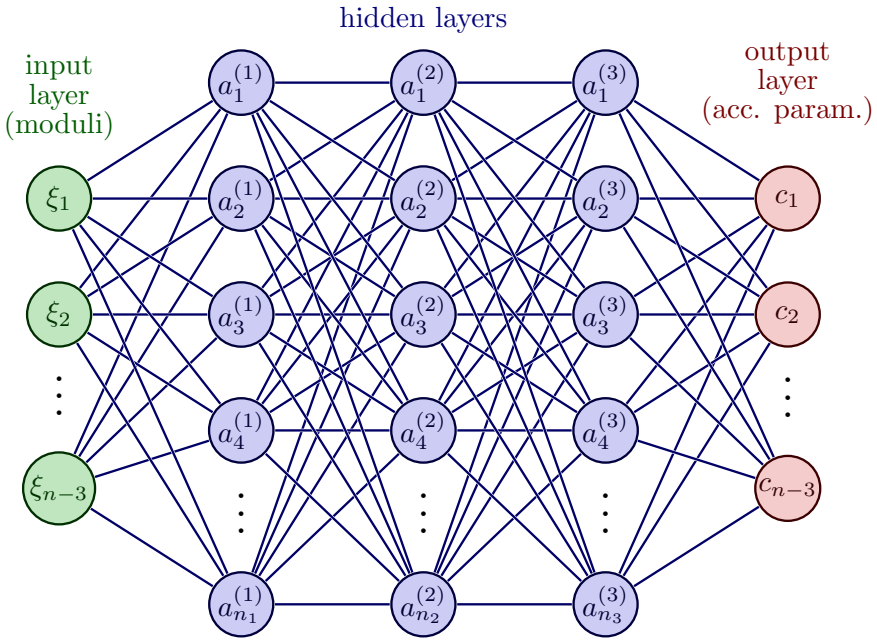

Figure 3. An example of an artificial neural network with 3 hidden layers containing n_i nodes each. It inputs the position of unfixed punctures (moduli) and outputs the (independent) accessory parameters.

At each node, based on the input received from the nodes in the previous layer, a mathematical operation is performed. More specifically, if we denote the collection of inputs received from the $(i-1)$-th layer containing n_{i-1} nodes to a node in the i-th layer containing n_i nodes as a column vector $a^{(i-1)}$ of length n_{i-1}, the nodes in the i-th layer would perform *non-linear* transformation

$$a^{(i-1)} \to a^{(i)} = \sigma(W^{(i)} a^{(i-1)} + b^{(i)})\,, \tag{3.1}$$

and transmit this to the nodes in the $(i+1)$-th layer. Here $W^{(i)}$ is a $n_i \times n_{i-1}$ matrix, $b^{(i)}$ is column vector of length n_i, and the function σ is some non-linear function called *activation function*. In the operation above the function σ acts on column vectors element-wise. The collection of all $W^{(i)}$ and $b^{(i)}$ for all layers is called *weights* and *biases* respectively and we collectively denote them by $\mathbf{W}$ and $\mathbf{b}$. Figure 4 summarizes this procedure. It can be shown that artificial neural networks can approximate class of arbitrarily complicated continuous functions [69–73] for which accessory parameters as a function of moduli are expected to belong.

We are interested to approximate the collection of accessory parameters $c_1, \cdots, c_{n-3}$ uniquely specifying the Strebel differentials on n-punctured spheres as a function of moduli $\xi_1, \cdots, \xi_{n-3}$ using neural networks such as the one shown in figure 3. Since this problem is inherently about complex numbers, we take weights and biases to be complex numbers and use complex neural networks [74–76]. For similar reasons, we take the activation function σ to be complex exponential linear unit ($\mathbb{C}ELU$), which is defined for $u \in \mathbb{C}$ by

$$\mathbb{C}ELU(u) \equiv ELU(\mathrm{Re}(u)) + i\, ELU(\mathrm{Im}(u))\,, \tag{3.2}$$

JHEP04(2024)016

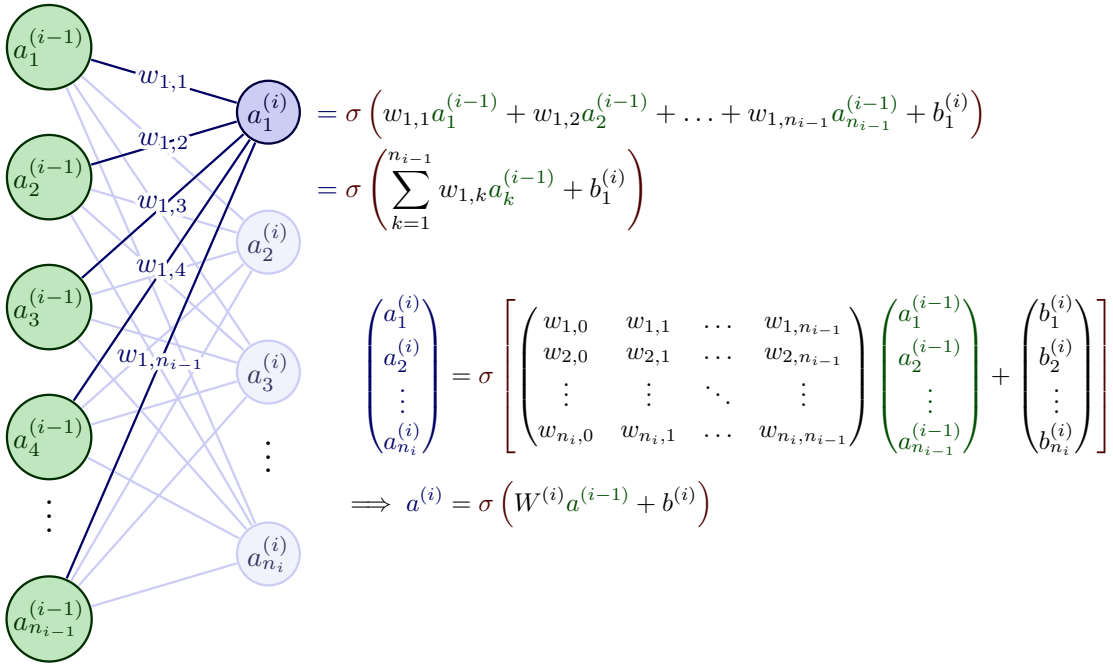

Figure 4. The summary of mathematical operations performed by artificial neural networks.

where ELU is the usual exponential linear unit activation function defined for $x \in \mathbb{R}$ as

$$ELU(x) = \begin{cases} x & \text{for } x > 0 \\ \alpha(\exp(x) - 1) & \text{for } x \leq 0 \end{cases}. \tag{3.3}$$

Here α is a hyperparameter of the network, and the activation function becomes ReLU for $\alpha = 0$. See appendix B for more details on the architecture of the network.

We have implicitly fix the positions of 3 punctures $(\xi_{n-2}, \xi_{n-1}, \xi_n)$ using $\mathrm{PSL}(2,\mathbb{C})$ transformation already. We pick these fixed punctures to be at $\xi_{n-2} = 0$, $\xi_{n-1} = 1$, and $\xi_n = \infty$. Moreover, we solved for three accessory parameters in terms of other parameters and the moduli using (2.3). We did these out of convenience for numerical calculations and in order to have a unique answer after training. In terms of 4-punctured sphere, this means we can use the parametrization given in (2.10). In this case the network inputs are the position of the unfixed punctures (moduli) ξ and outputs are the accessory parameters $a = a(\xi, \xi^*)$.

In order to approximate accessory parameters using neural networks we need to adjust the weights $\mathbf{W}$ and biases $\mathbf{b}$ of the network appropriately. Per usual in machine learning, this can be done using iterative (stochastic) gradient descent in the space of weights and biases based on an appropriate averaged loss function. Averaging here is made over finite number of points in the moduli space called *training set* $\mathcal{S}$, an example of which is shown in figure 5. At the end of the gradient descent we end up in some *local* minima of the averaged loss function in the space of weights and bias. The hope is that the resulting network from such local minima would be generic enough to approximate the behavior of accessory parameters, not only for the points in $\mathcal{S}$, but *everywhere* on a subset of $\mathcal{M}_{0,n}$ containing the training set $\mathcal{S}$. If this is the case, we say the neural network *learned* the accessory parameters.

JHEP04(2024)016

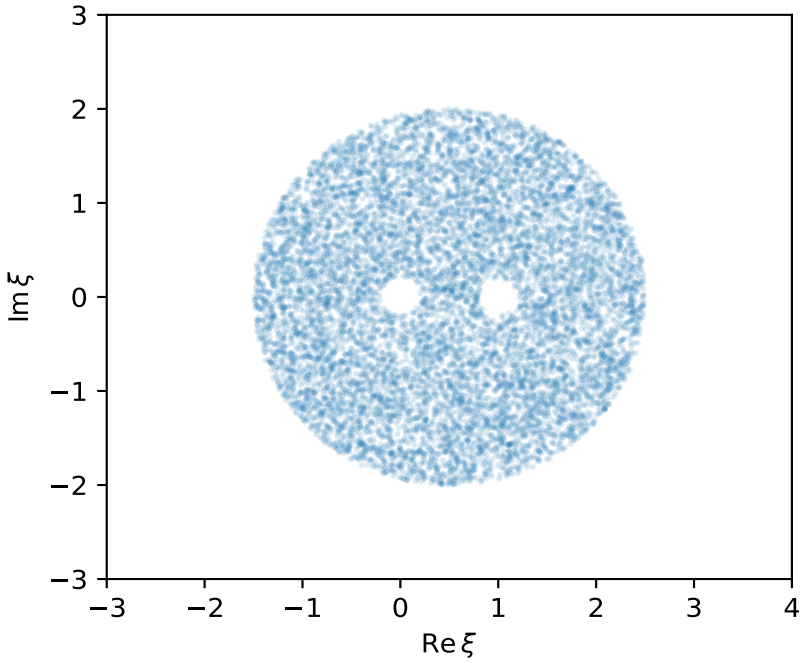

Figure 5. An example of training set $\mathcal{S}$ for 4-puncture spheres with $|\mathcal{S}| = 10^5$. Notice we have excluded small circles centered at $0, 1, \infty$ where 4-punctured sphere is close to degeneration and only sampled points from the remaining triply-connected region (training region).

Before we delve into the specifics of the averaged loss function, let us describe the training set $\mathcal{S}$. For us, the set $\mathcal{S}$ consists of random collection of points uniformly sampled over the moduli space $\mathcal{M}_{0,n}$ excluding the regions where punctures are about to collide. We call this region *training region*. The exclusion condition here is to make sure that the learned behavior for the accessory parameters doesn't get affected by the degeneration behavior of surfaces as our numerical evaluations get unreliable for them.[8] Recall that we are *not* interested in Strebel differentials on surfaces arbitrarily close to degeneration in the view of CSFT, as it is sufficient to obtain Strebel differentials on the vertex region $\mathcal{V}_{0,n}$ which doesn't contain such surfaces by construction. Hence, as long as we guarantee the training region to cover the vertex region $\mathcal{V}_{0,n}$ and successfully train, we are supposed to be able to get all the geometric data relevant to CSFT.

As mentioned earlier, gradient descent should be performed based on an appropriate averaged loss function. In the case at hand, this is constructed by averaging the function (2.9) over $\mathcal{S}$

$$L_{0,n}(\mathbf{W}, \mathbf{b}; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{\xi \in \mathcal{S}} \mathcal{L}_{0,n} \left(\varphi \left(c(\mathbf{W}, \mathbf{b}), \xi \right) \right) . \tag{3.4}$$

It is useful to comment on the dependence of indices here. As we have indicated in (2.9), the loss function depends on the quadratic differential φ, which in turn determined by the

[8] This numerical failure can be understood as follows. As the surface degenerates, some zeros and punctures get closer to each other. This makes evaluating the integral (2.6) numerically unstable and leads to failure in the training.

JHEP04(2024)016

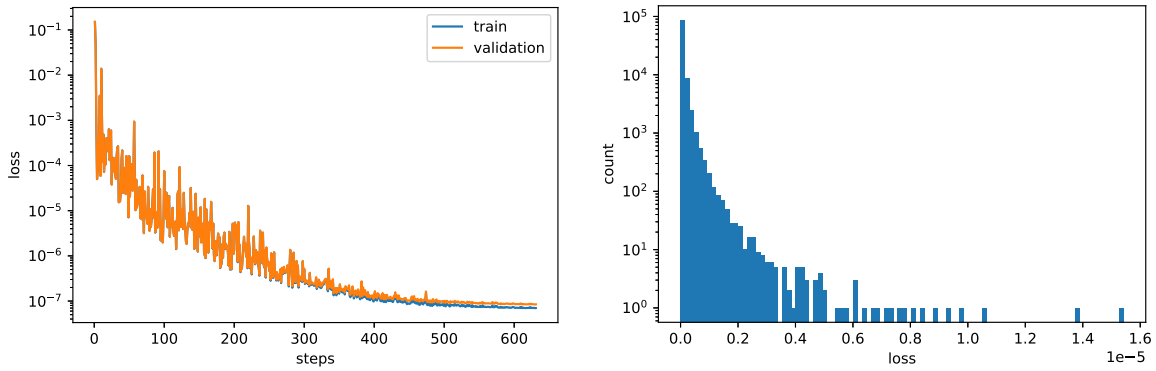

Figure 6. Training curve (left) and the distribution of loss over the test points in the training region (right) for "best NN". As one can observe from the curves on left training was successful and the curve on right informs most points have relatively small losses, except for few outliers.

choice of accessory parameters (collectively denoted as c) and moduli (collectively denoted as ξ), that is $\varphi = \varphi(c, \xi)$. But notice that the accessory parameters c are determined by the parameters of the network (weights and bias), hence we have $c = c(\mathbf{W}, \mathbf{b})$. When we average over the points in $\mathcal{S}$, we see the averaged loss function indeed has the dependence shown in (3.4). Note that we have not provided any labels for the points in the training set $\mathcal{S}$. So in essence we perform *unsupervised learning* for the accessory parameters using the loss (3.4).

Now we have all the ingredients to train an accessory parameter neural network for 4-punctured spheres. Before we give an example of a training run, let us summarize our strategy to confirm our results. We can test how well the network performs by investigating the loss function $\mathcal{L}_{0,4}$ over the training region. This involves sampling two new sets of points over the training region, called *validation set* if it is used during training or hyperparameter optimization and *test set* for subsequent calculations. If we observe the loss is small not only for the training set, but also over the validation/test sets, we conclude the network interpolates and declare it has learned the accessory parameter successfully over the training region. We evaluate the loss function to be 8.5×10^{-14} for the exact solution for $\xi = e^{i\pi/3}$ given in (2.13), so we see there is a scale associated with the loss function and its smallness indeed characterizes how close we are to Strebel differential.

An example of a training based on (3.4) is shown in figure 6 and some of its statistics shown in tables 2 and 3.[9] This particular network has 3 hidden layers with $[512, 128, 1028]$ nodes each respectively. Training was performed in `Python` using Google Jax [77] by sampling 10^5 points in the training region shown in figure 5. We confirm our results with the test set, but also evaluated the loss on the training and validation sets for comparison, and find that they are small and have the same order of magnitude, indicating that the network doesn't overfit and interpolates other points in the training region. Expanded details on the training and the architecture of network can be found in appendix B.

Observe from figure 6 that almost all points in the training region have relatively small

[9]We characterize the stochastic error of our training procedure after we calculate the off-shell 4-tachyon contact term v_4. For clarity of presentation, we just focus on "best NN" as a *generic* example. This particular choice is explained at the end of this subsection.

JHEP04(2024)016

Mean	8.86×10^{-8}	Min	1.32×10^{-11}
Median	3.81×10^{-8}	Max	1.54×10^{-5}

Table 2. The mean, median, minimum, and maximum values for the training loss of the "best NN".

Moduli	Acc. Param.	Loss	NN Acc. Param.	NN Loss
$\xi = e^{i\pi/3}$	$a = 2 + 2/\sqrt{3}i$	8.5×10^{-14}	$1.9997 + 1.1548i$	6.1×10^{-8}
$\xi = 1/2$	$a = 2$	7.8×10^{-12}	$1.9995 - 0.0001i$	6.6×10^{-7}
$\xi = -0.2 + 1.5i$	$a = 1.12245 + 1.2394i$	1.7×10^{-10}	$1.12250 + 1.23965i$	7.4×10^{-9}

Table 3. Comparison of the losses and accessory parameters for previously known solutions with the results "best NN" produced. The last point is solved using Newton's Method and is taken from [8].

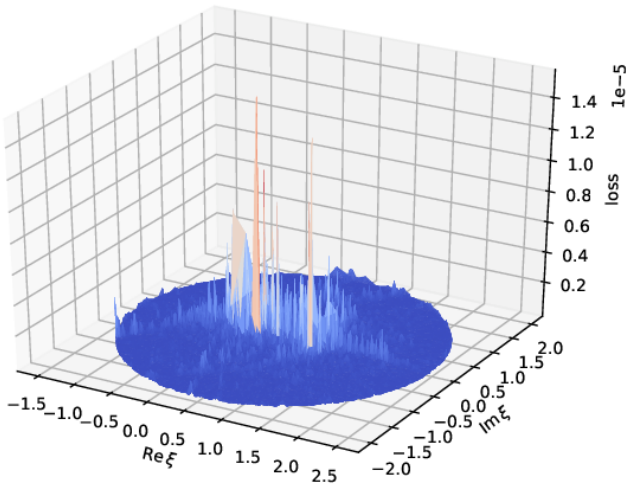

Figure 7. The distribution of the loss over the training region. Network underperforms close to the real line.

loss except for few outliers. In fact, by plotting the behavior of the loss function over the training region (figure 7) we see these outlier points primarily lie close to the real line — the region we don't need for CSFT. The reason for this behavior is actually clear: when the moduli is real, some of the terms in the sum given in (2.9) becomes zero and some of them imposes the equivalent condition as the critical graph degenerates. So relative to other points on the moduli space, the loss function close to the real line is *less* constrained leading to relatively larger loss. Even if the network relatively underperforms for the real moduli, it correctly generates the analytic behavior described in (2.16). This, along with the behavior of the accessory parameter over the training region, is shown in figure 8.

Already from figure 8, it is apparent that the network has learned the involution symmetry of a. We can quantify this, along with the shift and inversion symmetries given in (2.12) by comparing network's result for a pair of moduli related by symmetry. In order to do

JHEP04(2024)016

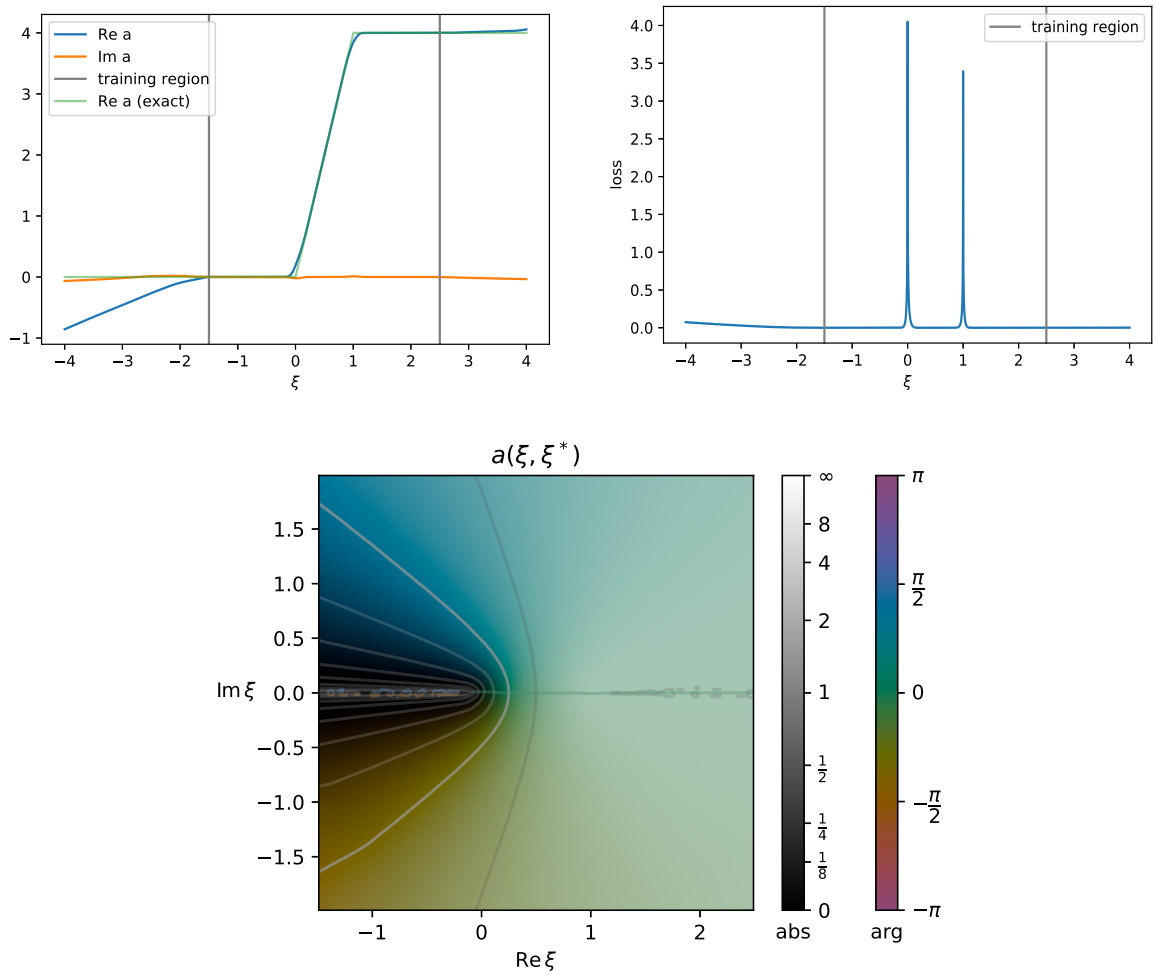

Figure 8. The behavior of the accessory parameter compared with the exact solution (top left) and its loss (top right) for the real moduli. Notice the exact and trained behaviors are almost indistinguishable and differ slightly only when the moduli is close to 0 or 1 or outside of the training region. Even if this is the case, we see the network was still able to extrapolate away from the training region. The overall behavior of the accessory parameter $a = a(\xi, \xi^*)$ is plotted below.

that, define the error by

$$\epsilon_g(\xi, \xi^*) \equiv |a(g(\xi), g(\xi^*)) - g(a)(\xi, \xi^*)|\,, \tag{3.5}$$

where g represents the symmetry transformations. For example, $g(\xi) = \xi^*$ and $g(a) = a^*$ for the involution symmetry. Figure 9 shows the distribution of ϵ_g for points sampled over the training region for three distinct symmetries. As one can see, the errors are quite small and we can conclude the network has learned the symmetries of the accessory parameter *without being explicitly programmed*.

Finally we compare our network's result for the accessory parameter with the polynomial fit provided for $a = a(\xi, \xi^*)$ by Moeller [8]. Again, we define the error between our results as

$$\epsilon_{\text{Moeller}}(\xi, \xi^*) \equiv |a(\xi, \xi^*) - a_{\text{Moeller}}(\xi, \xi^*)|\,, \tag{3.6}$$

JHEP04(2024)016

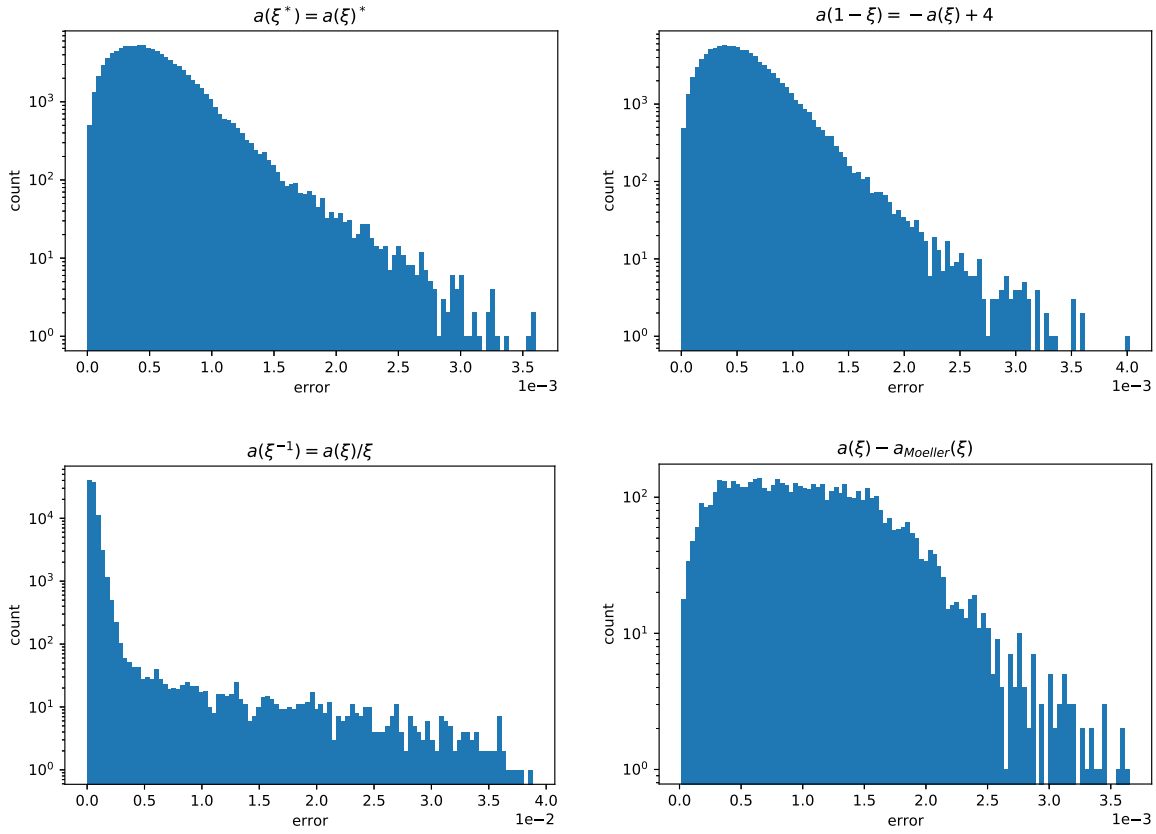

Figure 9. Distributions of errors corresponding to involution (top left), shift (top right), and inversion (bottom left) symmetries as well as the comparison of our results with the fit provided by Moeller in [8] (bottom right) over relevant regions.

where a_{Moeller} is given by the equation (6.9) in [8]. Since the fit in [8] is provided for a subset of the vertex region, we only consider the errors for the points sampled in this subset.[10] Again, we see our results and Moeller's fit are consistent with each other from figure 9.

We have listed various evidences for our approach above and they show that using machine learning to solve for accessory parameters is sound and the results one gets this way are consistent with the exact solutions as well as with the literature. We have worked with 4-punctured sphere, but we emphasize everything in this subsection admits a trivial generalization to higher-punctured spheres in principle. Thus the results here should be viewed as a *proof of principle*.

Before closing off this subsection, we note that the trained network always interpolates in the training region, but one can ask whether it *extrapolates* to outside of the training region. We observed extrapolation of our networks is not quite as good as their interpolations as it is already somewhat evident from figures 8. However we also observed that the better the network extrapolates, the better our results become. So if we would like to specialize to networks among the trained networks, it seems reasonable to us doing this based on how

[10] We used the indicator function neural network to do this, see next subsection for the details.

JHEP04(2024)016

well the network extrapolates for the real moduli and discard the rest of the runs.[11] This further defines the "best NN" for us: it is the network that extrapolates farthest away on the positive real line and it was the one we chose to report here. In higher-punctured spheres an analogous procedure can be repeated by investigating specific degeneration limits.

3.2 Indicator function neural network

As we have described in the previous subsection, it is possible to obtain accessory parameters as a neural network. Once such a representation is in our possession we can solve for the local coordinates and mapping radii as described in section 2. So all it remains for constructing classical CSFT action is to solve for the explicit description of the vertex region $\mathcal{V}_{0,n}$ over which the moduli integration has to be performed. In this subsection, we train a neural network for the indicator function for the vertex region $\mathcal{V}_{0,4}$, which has already been defined in (1.1). This provides an explicit characterization after (1.2). Again, we emphasize that the methods here can be trivially extended to the situation in higher-punctured spheres.

We train the indicator function neural network by performing *supervised learning*. In order to do that, we begin by uniformly sampling points over the training region. However, unlike before, we label these points based on whether they are in the vertex region or not. Remember, a point in the moduli space is in the vertex region if and only if all non-contractible curves in its associated φ-metric has length greater than or equal 2π. Since the accessory parameter is known, we can compute these lengths using the method described in section 2 and use them to label points: 1 if all such lengths are greater than or equal to 2π and 0 otherwise.

Randomly sampled points in the training region, together with their labels, would form the training set $\mathcal{S}'$ is given by

$$\mathcal{S}' = \{(\xi, \Theta^{(\text{true})}) \,|\, \xi \in \mathcal{M}_{0,n}, \Theta^{(\text{true})} \in \{0,1\}\}\,. \tag{3.7}$$

Now the problem of solving for the indicator function (2.24) transforms into a binary classification problem. In the view of this, let us call the indicator function neural network $\Theta^{(NN)}_{0,n}$. Its inputs are the moduli and it outputs some value *between* 0 and 1, i.e. $\Theta^{(NN)}_{0,n} : \mathcal{M}_{0,n} \to [0,1]$. So strictly speaking $\Theta^{(NN)}_{0,n}$ is a probability distribution for a given point in $\mathcal{M}_{0,n}$ to be element of the vertex region $\mathcal{V}_{0,n}$. In any case, we are going to observe transition from 0 to 1 is relatively sharp when $n = 4$. We comment more on this below.

The network $\Theta^{(NN)}_{0,n}$ can be trained to learn the indicator function for $\mathcal{V}_{0,n}$ after performing gradient descent in the space of its weights and biases using the *cross-entropy loss*

$$L'_{0,n} = \frac{-1}{|\mathcal{S}'|} \sum_{\xi \in \mathcal{S}'} \left[\Theta^{(\text{true})}(\xi, \xi^*) \log \Theta^{(NN)}_{0,n}(\xi, \xi^*) + (1 - \Theta^{(\text{true})}(\xi, \xi^*)) \log(1 - \Theta^{(NN)}_{0,n}(\xi, \xi^*)) \right]. \tag{3.8}$$

Like in the accessory parameter neural network, we are going to focus only on four-punctured spheres as a testing ground.

[11] We quantify this by investigating the behavior when the real moduli between $2 \geq \xi \geq 4$, comparing to the solution (2.16) and keeping the networks that has order of magnitude smaller average loss. See section 3.3 for more details.

JHEP04(2024)016

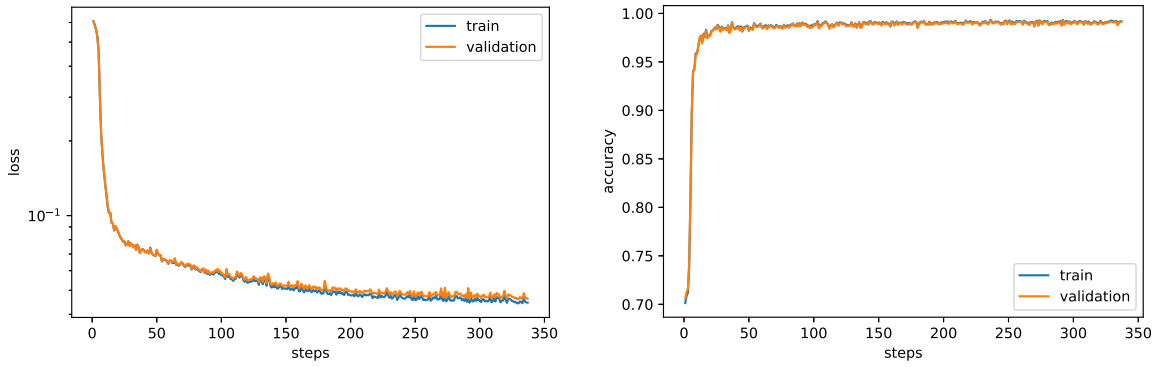

Figure 10. The training curve (left) and the progression of accuracy during training (right). High degree of accuracy is achieved for both training and validation sets.

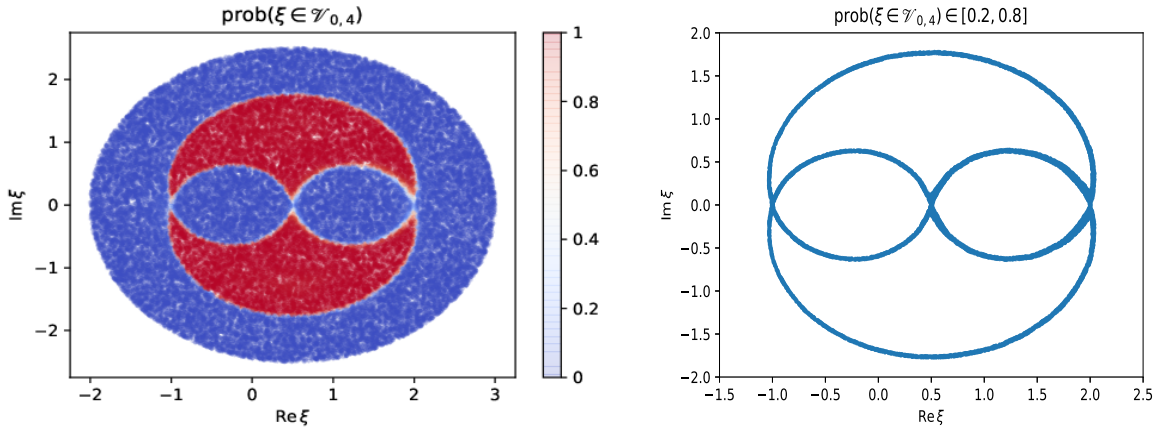

Figure 11. The probabilities obtained from the neural network $\Theta^{(NN)}_{0,4}$.

The training curve, along with the progression of accuracy during the training, for $\Theta^{(NN)}_{0,4}$ is shown in figure 10. For this particular network, we have used the training set $\mathcal{S}'$ constructed using the best NN. Again, the behavior we obtain was generic and such a choice was purely for the presentation purposes. We used 10^5 points for training. The weights and biases of this networks was chosen to be *real* and we input the complex moduli as a two-dimensional vector. We confirmed our results by checking the loss, as well as accuracy, for both training and validation sets. We have achieved 99.34% accuracy for the training set, 99.27% for the validation set, and 99.68% for the test set.[12] We observed no overfitting as is evident from figure 11. These results show the training was successful. Expanded details on the training and the architecture can be found in appendix B.

Figure 11 shows the probabilities produced by $\Theta^{(NN)}_{0,4}$. Note that the shape of the region $\mathcal{V}_{0,4}$ shown in 11 is consistent with the literature [6, 8]. Moreover, we see the transition from 0 to 1 is quite sharp: $\Theta^{(NN)}_{0,4}$ provides a good approximation for the indicator function. Since this is the case, we declare a point in $\mathcal{M}_{0,4}$ is an element of $\mathcal{V}_{0,n}$ when the generated

[12] It is not common for the test accuracy to be higher than the training and validation accuracy. This may indicate an *underfit*. However, the test loss is much higher, see table 6.

JHEP04(2024)016

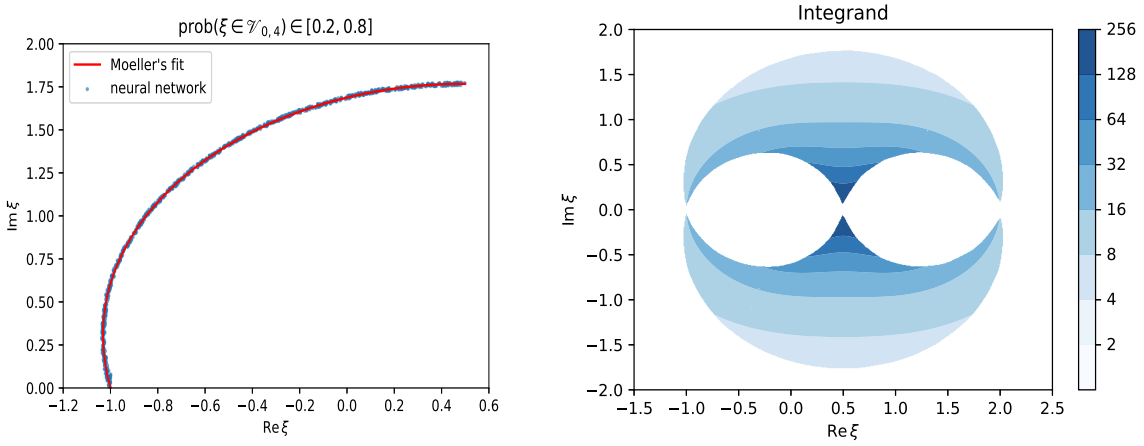

Figure 12. The comparison of $\partial\mathcal{V}_{0,4}$ restricted to $\mathrm{Re}(\xi) \leq 1/2$, $\mathrm{Im}(\xi) \geq 0$, and $|\xi| \geq 1$ between our results and the fit provided by Moeller in [8] (left) and the integrand of the integral (3.11) (right).

probability is greater than or equal to 1/2. That is, we declare our indicator function to be

$$\Theta_{0,n}(\xi, \xi^*) \equiv H\left(\Theta_{0,n}^{(NN)}(\xi, \xi^*) - \frac{1}{2}\right), \tag{3.9}$$

where $H(x)$ is Heaviside step function.

More quantitatively, we can compare the fit provided by Moeller for the boundary of the vertex region $\partial\mathcal{V}_{0,4}$ restricted to $\mathrm{Re}(\xi) \leq 1/2$, $\mathrm{Im}(\xi) \geq 0$, and $|\xi| \geq 1$ (equation (6.5) in [8]) with the corresponding curve we obtain, i.e. $\Theta_{0,4}^{(NN)}(\xi, \xi^*) \approx 1/2$ restricted in similar way. This is shown in figure 12. Again we see a perfect agreement between our results.

3.3 Off-shell 4-tachyon contact term

With an explicit description for the vertex region in terms of (3.9), we now have all the ingredients to find terms in the classical closed string tachyon potential. This is given by [6, 8]

$$V(t, \cdots) = -\frac{1}{2}t^2 - \sum_{n=3}^{\infty} v_n \frac{t^n}{n!} + \cdots . \tag{3.10}$$

Here dots represents the terms involving fields other than the tachyon t. We are only interested in v_4 whose expression can be read from

$$v_n = (-1)^n \frac{2}{\pi^{n-3}} \int_{\mathcal{M}_{0,n}} d^2\xi \; \Theta_{0,n}(\xi, \xi^*) \prod_{i=1}^{n} \frac{1}{\rho_i^2}, \tag{3.11}$$

where $\Theta_{0,n}$ is the indicator function for $\mathcal{V}_{0,n}$ and ρ_i's are the mapping radii associated with punctures. The convention for the measure is $d^2\xi = d(\mathrm{Re}\xi)d(\mathrm{Im}\xi)$. Derivation of (3.11) can be found in [6]. Note that everything in the integrand of (3.11) can be expressed in terms of two neural networks of previous subsections, so all we need to do at this point is to perform this integration over the moduli space. The integrand of (3.11) for $n = 4$ is shown in figure (12). Our results are already presented in table (1).

The integral is first computed using the trapezoid method along the imaginary and real dimensions, with a grid of 700×700 points for $\mathrm{Re}(\xi) \in [-1.1, 2.1]$ and $\mathrm{Im}(\xi) \in [-2.1, 2.1]$.

JHEP04(2024)016

In order to assess the stochastic deviations, we have run the full pipeline (training of the accessory parameter and indicator function networks, and computations of v_4) 10 times. As we mentioned earlier, we observed that networks which perform best are those which generalize well outside the training region on the real axis (as in figure 8). As a consequence, we kept only the 4 networks whose mean loss for $\xi \in [2.5, 4]$ is below 0.1. This allows determining how v_4 varies when we change the random seed, which consists of using different training points, network initializations, and stochastic gradient descent. From the scale of uncertainty of our result for this ($v_4 = 72.320 \pm 0.146$) we see our algorithm is sufficiently stable and produced results consistent with the literature. For the best NN we report $v_4 = 72.396$ using trapezoid method.

For the best NN, we additionally perform a Monte Carlo integration with 2×10^6 points in the vertex region. We report its mean and standard deviation to be $v_4 = 72.366 \pm 0.096$ by evaluating it 5 times. We stress that we use the sharpened indicator function (3.9) in both methods. Of course, trapezoid method provides a deterministic result for v_4 so one may question the point of using Monte-Carlo integration here. However, we note that the moduli integration would be higher dimensional for higher-punctured spheres for which Monte-Carlo integration would be superior to any deterministic method. Here we would like to imitate that case and see how large the resulting errors was due to integration, while still having a baseline for the expected result. We observe our result for v_4 is still sufficiently precise even if we use Monte-Carlo integration. The convergence of this integral can be further improved using more points or employing importance sampling.

4 Discussion and future directions

In this paper we have characterized 4-string string contact interaction using machine learning by constructing neural networks for the accessory parameter for Strebel differentials and the indicator function for the vertex region in the moduli space. Doing so allowed us to construct the local coordinates associated with the 4-string contact interaction. We tested our pipeline by computing the off-shell 4-tachyon contact term in the tachyon potential. We obtained a good agreement with the results in the literature.

We would like to emphasize few advantages of using machine learning over traditional numerical methods to characterize the local coordinates in CSFT:

1. The algorithm presented here is manifestly independent of the number of punctures. So it is in principle possible to repeat the similar construction for n-punctured spheres.

2. Having a neural network representation for the accessory parameters would help to explore properties of Strebel differentials and uncover hidden patterns as neural networks are just approximations to the actual function.

3. Building string interactions using machine learning would eventually simplify the technical aspects of CSFT calculations, as all the geometric data needed for this is encoded in the neural networks. The complete code will be made public in the future.[13]

[13] See https://github.com/melsophos/pysft-nn for the architecture and weights. The code will be made available as a package together with [57].

JHEP04(2024)016

Of course, providing fits for the relevant functions, like Moeller did in [8], achieves the same goal. However, even in the case of 5-punctured sphere, there is no polynomial fits for all of these functions [13], and even if we are able to construct one, it is a general wisdom that neural network representation of functions are superior to fits.

We want to briefly comment on the precision of our results. Even though we get quite close to the results in the literature [6, 8], they differ in the third significant digit and there is still room for improvement. We think this is primarily due to the training for the accessory parameter being not sufficiently precise. Even though we have reached quite low losses during the training, as is evident in figure 6, it is not as low as one would get using Newton's method. This indicates the training precision has to be improved, at least until we reach the same order for the loss as Newton's method [8]. Indeed, the accessory parameters are used in rest of the computations so it is crucial to minimize its error as much as possible.

This type of problem is unconventional from the perspective of machine learning since high precision is rarely needed (although see the recent work [78]). The tendency is to *decrease* the float precision. Having a loss plateau around 10^{-7} in the training shown in figure (8) motivates that we need to use double precision `float64` instead of simple precision `float32`. Still, this is not sufficient as the usual optimization techniques have not been designed to handle such scenarios. For instance, both the gradients and learning rate are around 10^{-7} at the end of the training which implies that weight updates are effectively frozen. Another instance is that the use of regularization (such as L_2) can dominate the loss (2.9) in later epochs. One may try to circumvent these problems by turning off/decaying the regularization and/or using learning rate restart, that is increasing the learning rate to counter-balance the vanishing gradients for later epochs. We have made a preliminary study on using some of these techniques which resulted in smaller loss. We note that the hyperparameter optimization is difficult for these techniques as one needs to train networks over extended periods to find the long-time effects.

In any case, we think such level of precision, at least for the purpose of establishing existence of closed string tachyon vacuum by level/order truncation, won't be needed. Because if the tachyon vacuum happens to be finely-tuned and requires terms in the tachyon potential to be evaluated precisely, the whole procedure for searching the vacuum by truncating the theory won't work; as it would presumably require all orders of CSFT to be considered.[14]

There are numerous natural directions one can take in future. Here we list some of them seem to us of utmost importance:

1. An obvious direction is to generalize this approach to higher-order string contact interactions which we are currently working on [57]. As we have emphasized numerous times throughout the paper, there are no conceptual obstruction doing so, as long as the algorithm, especially the training for the accessory parameters, scales favorably.

   Observe that the number of distinct integrals in (2.9) scales as $\mathcal{O}(n^2)$ and evaluating all of them may slow down the training for higher-punctured spheres. To remedy this

[14] We thank Martin Schnabl for this argument.

JHEP04(2024)016

problem, consider following modification to the loss function[15]

$$\mathcal{L}_{0,n}(\varphi) \to \mathcal{L}_{0,n}(\varphi) = \frac{1}{2n-5}\sum_{(ij)}(\mathrm{Im}(\ell(z_i,z_j)))^2\,. \tag{4.1}$$

Here sum over (ij) means the following. We first construct an ordered list of zeros and only compute the complex lengths between zeros adjacent to each other in this list. It is easy to argue this new function preserves the properties of (2.9). However, compared to (2.9), we use just enough condition to specify Strebel differential at the function's global minimum while being agnostic of the shape of the critical graph. The number of integrals in (4.1) scales as $\mathcal{O}(n)$ and this may speed up the training, possibly at the expense of precision due to imposing less condition on the differential.

Scaling the algorithm for higher-punctured spheres may also require using more advanced architectures, for example, by including equivariant layers for the unitary group $\mathrm{U}(n)$ and complex translations $\mathbb{C}^n$ [79–81]. Furthermore we may also want to represent the local coordinates themselves by a new neural network (and more generally, it would be interesting to understand how to compute conformal maps as neural networks) or use graph neural networks [82] to extract properties from the critical graphs as they become more complicated with increasing number of punctures.

2. We have trained a neural network for the indicator function distinguishing the vertex region from the Feynman region. It is also possible to train a network that would distinguish distinct type of degeneration of punctured-spheres from each other. For 4-punctured spheres, this means we can train a network that takes different values for s,t,-and u-type degenerations.

   Such networks allow us to sample points just from the relevant parts of the Feynman region and based on these points it may be possible to train a network for the Zwiebach differentials using the following modified loss function

$$\begin{aligned}\mathcal{L}_Z(\varphi) = {}& \mathrm{Im}(\ell(z_1,z_2))^2 + \mathrm{Im}(\ell(z_3,z_4))^2 \\ & + (\mathrm{Re}(\ell(z_1,z_2))^2-\pi^2)^2 + (\mathrm{Re}(\ell(z_3,z_4))^2-\pi^2)^2\,,\end{aligned} \tag{4.2}$$

   and variations thereof (i.e. $z_2 \leftrightarrow z_3$ and $z_2 \leftrightarrow z_4$). Remember Zwiebach differentials have a disconnected critical graph in Feynman region (that is, zeros (z_1,z_2) and (z_3,z_4) no longer connected to each other by a critical trajectory) and this is reflected by eliminating terms such as $\mathrm{Im}(\ell(z_1,z_3))$ and replacing them with terms such as $(\mathrm{Re}(\ell(z_1,z_2))^2-\pi^2)^2$.

   Assuming such network can be trained, we can find the local coordinates associated with Feynman diagrams as well. This is interesting for few reasons. First, *all* off-shell string amplitudes will be characterized using neural networks. But more interestingly, this gives an alternative way to plumbing fixture to obtain such local coordinates. It might be interesting to cross-compare these two methods.

3. Since we approximated functions relevant to the geometry of 4-string contact interaction as neural networks, we can try performing *symbolic regression* to get an analytic insight

[15] We thank Barton Zwiebach for suggesting this modification.

JHEP04(2024)016

into the nature of these functions. In particular, it may be interesting to search for closed form expressions for the accessory parameter and $\partial\mathcal{V}_{0,4}$ shown in figure 12.

4. Generalizing the ideas in this paper to the case of higher genera, especially using minimal-area vertices, would possibly take a non-trivial effort: it is known not all minimal-area metrics in higher genera arises from Strebel differentials while our loss function is intrinsically about the latter. Still, it may be possible to exploit convex optimization approach to minimal-area problem [83–85] to construct a suitable loss function.

   Better yet, one may try solving a version of accessory parameter problem appearing in the case of hyperbolic string vertices [22] using machine learning to construct quantum CSFT. In this case the loss function ought to impose the solutions of Fuchsian equation to have a real monodromy on each non-contractible cycle of a Riemann surface at its minimum. We expect the natural loss function for this problem to be independent of the number of punctures and genera. This approach may even provide novel insights to Fuchsian uniformization considering these two problems are continuously connected.

Acknowledgments

We would like to thank Barton Zwiebach for endless discussions and introducing us to the world of quadratic differentials. We also would like to thank Riccardo Finotello, Manki Kim, Fabian Ruehle, and Siddharth Mishra-Sharma for discussions. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics of U.S. Department of Energy under grant Contract Number DE-SC0012567. This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 891169.

A Some details on numerical evaluations

In this appendix we give some details on numerical evaluations. In particular we describe the implementation of continuous square root $\sqrt[\pm]{}$, and the numerical evaluations of the integrals for the complex length (2.6) as well as the mapping radii (2.23).

A.1 Continuous square root

Square root $\sqrt{z}$ has a branch cut on z-plane and it usually placed along the negative real axis. However, for numerical evaluations of the integrals, such as the one given in (2.5), it is advantageous to define a *continuous square root* (denoted by $\sqrt[\pm]{}$) by taking the domain of the complex square root to be the double cover of $\mathbb{C}\setminus\{0\}$ rather than the complex plane with a branch cut. In this domain $\sqrt[\pm]{}$ is a holomorphic function. This way of evaluating square root is inspired by [8].

Note that the global sign of $\sqrt[\pm]{}$ is ambigious. That is for $z_1 = z_2$, we have $\sqrt[\pm]{z_1} = \pm\sqrt[\pm]{z_2}$ depending on which sheet of the domain we are evaluating $\sqrt[\pm]{}$. As we have stated in the main text, this only leads to technical issues in our numerical evaluations which we overcome by simply squaring the resulting expressions and/or correcting the global sign a posteriori.

JHEP04(2024)016

In practice, the continuous square root $\sqrt[\pm]{z}$ is evaluated on a given path $z(t)$ by discretizing it as $\{z_i\}_{i=1}^k$ and comparing the complex square roots $\sqrt{z_i}$ and $\sqrt{z_{i+1}}$. More precisely, one is instructed to compute the difference $|\sqrt{z_{i+1}} - \sqrt{z_i}|$ at each step and choose the sign for $\sqrt[\pm]{z_{i+1}} = \pm\sqrt{z_{i+1}}$ such that the said difference is minimum. This way we compensate any branch crossing for complex square root by flipping its sign and get continuous square root instead. As long as the step size is small relative to the distances between z_i and the branch points, the resulting sequence $\{\sqrt[\pm]{z_i}\}_{i=1}^n$ provides a good approximation to $\sqrt[\pm]{z(t)}$.

A.2 The integral for the complex length

The complex length (2.6) in the case of n-punctured sphere takes the following form

$$\ell(z_i, z_j) = \int_{z_1 i}^{z_j} \sqrt[\pm]{\phi(z)} = \int_{z_i}^{z_j} \frac{\sqrt[\pm]{-(z-z_1)\times\cdots\times(z-z_{2n-4})}}{z(z-1)(z-\xi_1)\times\cdots\times(z-\xi_{n-3})} dz\,. \tag{A.1}$$

Here $z_1, \cdots z_{2n-4}$ are (not-necessarily distinct) zeros of the quadratic differential determined by the accessory parameters c_i and the moduli $\xi_1, \cdots \xi_{n-3}$. For convenience we have fixed the positions of three punctures to $\{0, 1, \infty\}$ and solved their accessory parameters using a version of (2.3).

We take the integration path to be the straight line between the zeros z_i, z_j (2.7) and this gives

$$\ell(z_i, z_j) = \left(\frac{z_j - z_i}{2}\right)^2 \int_{-1}^{1} \frac{\sqrt[\pm]{p_{ij}(z(t))}}{q(z(t))} \sqrt{1-t^2} dt\,. \tag{A.2}$$

Above we defined the following polynomials

$$p_{ij}(z) \equiv \prod_{\substack{k=1 \\ k\neq i,j}}^{2n-4} (z - z_k), \qquad q(z) \equiv \prod_{k=1}^{n-1} (z - \xi_k)\,. \tag{A.3}$$

In order to make the discussion uniform we have further defined $\xi_{n-2} \equiv 0$ and $\xi_{n-1} \equiv 1$ above. Note that we have taken out the factors $(z - z_i)$ and $(z - z_j)$ from the product in $p(z)$ and we have used the usual square root for $\sqrt{1-t^2}$ as $1 - t^2 \geq 0$ for $-1 \leq t \leq 1$.

The goal now is to numerically evaluate the integral (A.2). In order to do this accurately we have to subtract possible pole contributions and add their analytic expressions. That is, we have to evaluate

$$\begin{aligned}\ell(z_i, z_j) = \left(\frac{z_j - z_i}{2}\right)^2 \left[\int_{-1}^{1} \left(\frac{\sqrt[\pm]{p_{ij}(z(t))}}{q(z(t))} - \sum_{k=1}^{n-1} \frac{s_k r_k}{z(t) - \xi_k}\right) \sqrt{1-t^2} dt\right] \\ + \left(\frac{z_j - z_i}{2}\right)^2 \sum_{k=1}^{n-1} s_k r_k \int_{-1}^{1} \frac{\sqrt{1-t^2} dt}{z(t) - \xi_k}\,,\end{aligned} \tag{A.4}$$

where

$$r_k \equiv p_{ij}(\xi_k) \prod_{\substack{j=1 \\ j\neq k}}^{n-1} \frac{1}{\xi_k - \xi_j}\,, \tag{A.5}$$

JHEP04(2024)016

and s_k's are signs that should be chosen so that

$$\max_{-1\le t\le 1}\left[\frac{\sqrt[\pm]{p_{ij}(z(t))}}{q(z(t))}-\sum_{k=1}^{n-1}\frac{s_k r_k}{z(t)-\xi_k}\right], \tag{A.6}$$

is minimal. This additional step was necessary: the sign of the integral (A.2) is ambigious due to continuous square root. Notice there are 2^{n-1} combinations to check.

Once s_k's are determined the first line of the integral (A.4) can be evaluated by trapezoid method using 201 grid points. It is possible to have cancellation errors for this integral, as we may need to subtract two large quantities close to the punctures, but we have observed this has not caused any problems during the training of the accessory parameter neural network.

The second line of the integral (A.4) can be analytically evaluated as

$$\int_{-1}^{1}\frac{\sqrt{1-t^2}}{z(t)-\xi_k}dt=\frac{2}{z_j-z_i}\int_{-1}^{1}\frac{\sqrt{1-t^2}}{t+u}=\frac{2\pi}{z_j-z_i}(u-i\sqrt{u-1}\sqrt{u+1}), \tag{A.7}$$

where

$$u\equiv\frac{z_i+z_j-2\xi_k}{z_j-z_i}. \tag{A.8}$$

In these expressions the square root is taken on the principal branch. After doing these one has to take the imaginary part of the complex length, square it, and sum over all pairs of zeros for the loss function (2.9).

A.3 The integral for the mapping radii

The mapping radii integral (2.23) isn't amenable to numerical evaluations. In order to put it into a better form let us first consider the following identity

$$\lim_{\epsilon\to 0}\left[\log|\epsilon|-\left(-\frac{1}{\sqrt{z_i-\xi_k}}\int_{\xi_k+\epsilon}^{z_i}\frac{\sqrt{z_i-z}}{z-\xi_k}dz+2(\log 2-1)+\log|z_i-\xi_k|\right)\right]=0. \tag{A.9}$$

Here $\epsilon\in\mathbb{C}$ is assumed lying on the straight line from the puncture at $z=\xi_k$ to the zero at $z=z_i$ and the branch of square root is adjusted so that this limit exists. Using this in (2.23) drops out any need for $\epsilon\to 0$ limit as

$$\begin{aligned}\log\rho_k&=\lim_{\epsilon\to 0}\left[\mathrm{Im}\int_{\xi_k+\epsilon}^{z_i}\left(\sqrt[\pm]{\phi(z)}-\frac{i}{\sqrt{z_i-\xi_k}}\frac{\sqrt{z_i-z}}{z-\xi_k}\right)dz\right]+2(\log 2-1)+\log|z_i-\xi_k|\\&=\mathrm{Im}\int_{\xi_k}^{z_i}\left(\sqrt[\pm]{\phi(z)}-\frac{i}{\sqrt{z_i-\xi_k}}\frac{\sqrt{z_i-z}}{z-\xi_k}\right)dz+2(\log 2-1)+\log|z_i-\xi_k|.\end{aligned} \tag{A.10}$$

Parametrizating the straight line by $z(t)=z_i+t(\xi_k-z_i)$ for $0\le t\le 1$ and exponentiating we get

$$\rho_k=\frac{4}{e^2}|z_i-\xi_k|\exp\left[\mathrm{Im}\left((z_i-\xi_k)\int_0^1\left(\frac{\sqrt[\pm]{v_{k,i}(z(t))}}{q(z(t))}-\frac{i}{z(t)-\xi_k}\right)\sqrt{t}dt\right)\right]. \tag{A.11}$$

The polynomial $q(z)$ is defined in (A.3) and we further define

$$v_{k,i}(z)\equiv-(\xi_k-z_i)\prod_{\substack{j=1\\ j\neq i}}^{2n-4}(z-z_j). \tag{A.12}$$

JHEP04(2024)016

Map. Rad.	$z_{c,1}$	$z_{c,2}$	$z_{c,3}$	$z_{c,4}$	Mean	Std. dev.
ρ_1	0.8139	0.8126	0.8132	0.8120	0.8129	0.0007
ρ_2	0.4832	0.4825	0.4821	0.4829	0.4827	0.0004
ρ_3	0.5176	0.5184	0.5188	0.5180	0.5182	0.0004
ρ_4	0.7575	0.7569	0.7581	0.7563	0.7572	0.0007

Table 4. The mapping radii when punctures are placed at $P = \{0, 1, 0.8734 - 0.6242i, \infty\}$.

With the choice of sign for continuous square root made below (2.22) the integral (A.11) is finite. In practice though, we would impose this choice by *demanding* the mapping radii has to be finite as it is more convenient to do numerically.

The integral (A.11) computes the mapping radii associated wit the finite punctures. For the mapping radius associated with the puncture at $z = \infty$, we invert the position of punctures to $1/\xi_k$, adjust the accessory parameters (akin to (2.12)) and calculate the mapping radius associated with the puncture at the origin. Since the mapping radii associated with the origin is invariant under inversion, we obtain the mapping radii associated with the puncture at $z = \infty$.

The integral (A.11) numerically evaluated using Chebyshev-Gauss method [86] after changing variables $t = 1 - x^2$ using 500 grid points. Notice that given ρ_k, different choices for z_i must give the same result as explained below (2.22). We observed this is indeed the case. For example, the mapping radii and their uncertainties are reported in table 4 when punctures are placed at $\{0, 1, 0.8734 - 0.6242, \infty\}$. The behavior of the (mean) mapping radii is shown in figure (13).

It was possible to have cancellation errors in the numerical evaluation of the integral (A.11), but having small standard deviation for the results above indicates that it does not actually pose a threat to the accuracy of our computation. We indeed observed that it doesn't cause any issue during our evaluations. In fact, we have used the mean value for the mapping radii in our evaluations and since their associated uncertainties are small we opt to not include them into our final result for v_4.

B Machine learning details

In this appendix, we present more details on our neural networks and their training. We build the training, validation and test sets by randomly sampling 10^5 points on the complex plane, restricted to the disk of radius 2 centered at 1/2 with two disks of radius 0.2 centered at 0 and 1 excised. We use different sets for both networks and each run we sampled new sets. The test sets are left aside until the end of the training to evaluate the performance. The integrals for v_4 are computed using yet another set, a grid for trapezoid method and random points in the vertex region for Monte-Carlo.

Both neural networks are fully connected. Training is performed using the eponymous set employing AdamW and we use early stopping: metrics are evaluated on the validation set at the end of each epoch, and training stops when there is no improvement of the loss (resp. accuracy) using the accessory neural network (resp. indicator function neural network) after

JHEP04(2024)016

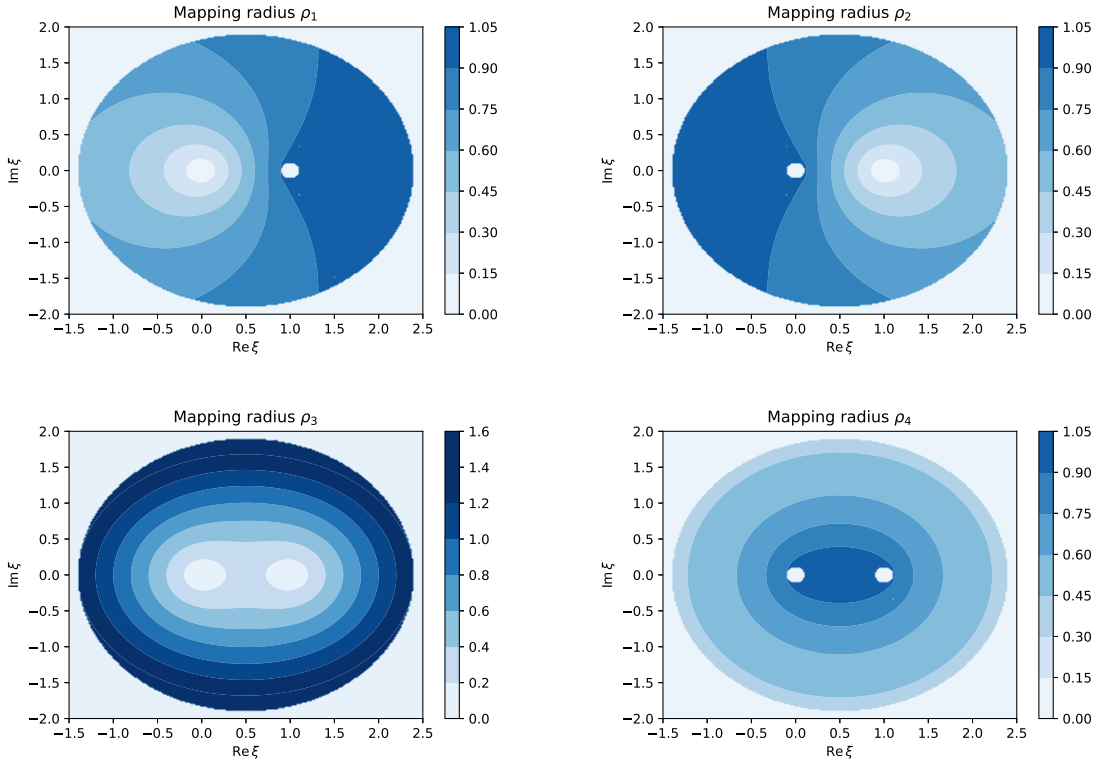

Figure 13. The behavior of the mapping radii ρ_i associated with the punctures $\{0, 1, \xi, \infty\}$. Small regions around $\{0, 1, \infty\}$ excised as the evaluation gets unreliable when punctures are about to collide.

Hyperarameter	Accessory parameter NN	Indicator function NN
hidden layers	$[512, 128, 1028]$	$[512, 32, 8, 8]$
activation function	$\mathbb{C}\text{ELU}(\alpha = 0)$	$\text{ELU}(\alpha = 0.25)$
batch size	32	64
clip	6	4
L_2 regularization	8×10^{-6}	4×10^{-2}
weight decay	8×10^{-5}	1.5×10^{-3}
base learning rate	4×10^{-4}	7×10^{-5}
warm-up steps	20	5
decay factor	0.94	0.99
decay period steps	5	5

Table 5. Hyperparameters of our neural networks.

100 epochs. Gradients are clipped by the global norm. We employ the following learning rate scheduler: a warm-up period increases the learning rate linearly from 0 to the base learning rate during a given number of epochs; then, the learning rate is decayed exponentially with some period. The hyperparameters are presented in table 5 and were found using BOHB hyperparameter optimization [87]. The statistics for the loss of our runs are shown in table 6.

JHEP04(2024)016

Set	Metric	Mean	Std
Train	loss (accessory)	7.11×10^{-8}	6.21×10^{-9}
	loss (vertex)	5.05×10^{-2}	5.89×10^{-3}
	accuracy (vertex)	99.05%	0.20%
Validation	loss (accessory)	8.63×10^{-8}	6.19×10^{-9}
	loss (vertex)	5.16×10^{-2}	4.86×10^{-3}
	accuracy (vertex)	98.99%	0.18%
Test	loss (accessory)	9.39×10^{-8}	6.12×10^{-9}
	loss (vertex)	3.29×10^{-1}	1.27×10^{-2}
	accuracy (vertex)	99.34%	0.18%

Table 6. Mean and standard deviations of the metrics for the different datasets, averaged over 4 runs. The full pipeline takes 294 ± 29 minutes to run on Google Colab.

References

[1] B. Zwiebach, *Closed string field theory: Quantum action and the B-V master equation*, *Nucl. Phys. B* **390** (1993) 33 [hep-th/9206084] [INSPIRE].

[2] C. de Lacroix et al., *Closed Superstring Field Theory and its Applications*, *Int. J. Mod. Phys. A* **32** (2017) 1730021 [arXiv:1703.06410] [INSPIRE].

[3] T. Erler, *Four Lectures on Closed String Field Theory*, *Phys. Rept.* **851** (2020) 1 [arXiv:1905.06785] [INSPIRE].

[4] H. Erbin, *String Field Theory: A Modern Introduction*, Springer Cham (2021) [DOI:10.1007/978-3-030-65321-7] [INSPIRE].

[5] A. Belopolsky and B. Zwiebach, *Off-shell closed string amplitudes: Towards a computation of the tachyon potential*, *Nucl. Phys. B* **442** (1995) 494 [hep-th/9409015] [INSPIRE].

[6] A. Belopolsky, *Effective Tachyonic potential in closed string field theory*, *Nucl. Phys. B* **448** (1995) 245 [hep-th/9412106] [INSPIRE].

[7] Y. Okawa and B. Zwiebach, *Twisted tachyon condensation in closed string field theory*, *JHEP* **03** (2004) 056 [hep-th/0403051] [INSPIRE].

[8] N. Moeller, *Closed bosonic string field theory at quartic order*, *JHEP* **11** (2004) 018 [hep-th/0408067] [INSPIRE].

[9] H.-T. Yang and B. Zwiebach, *Testing closed string field theory with marginal fields*, *JHEP* **06** (2005) 038 [hep-th/0501142] [INSPIRE].

[10] H. Yang and B. Zwiebach, *Dilaton deformations in closed string field theory*, *JHEP* **05** (2005) 032 [hep-th/0502161] [INSPIRE].

[11] H. Yang and B. Zwiebach, *Rolling closed string tachyons and the big crunch*, *JHEP* **08** (2005) 046 [hep-th/0506076] [INSPIRE].

JHEP04(2024)016

[12] H. Yang and B. Zwiebach, *A closed string tachyon vacuum?*, *JHEP* **09** (2005) 054 [hep-th/0506077] [INSPIRE].

[13] N. Moeller, *Closed Bosonic String Field Theory at Quintic Order: Five-Tachyon Contact Term and Dilaton Theorem*, *JHEP* **03** (2007) 043 [hep-th/0609209] [INSPIRE].

[14] N. Moeller and H. Yang, *The nonperturbative closed string tachyon vacuum to high level*, *JHEP* **04** (2007) 009 [hep-th/0609208] [INSPIRE].

[15] N. Moeller, *Closed Bosonic String Field Theory at Quintic Order. II. Marginal Deformations and Effective Potential*, *JHEP* **09** (2007) 118 [arXiv:0705.2102] [INSPIRE].

[16] N. Moeller, *A tachyon lump in closed string field theory*, *JHEP* **09** (2008) 056 [arXiv:0804.0697] [INSPIRE].

[17] L. Schlechter, *Closed Bosonic String Tachyon Potential from the $\mathcal{N} = 1$ Point of View*, arXiv:1905.09621 [INSPIRE].

[18] S.F. Moosavian and R. Pius, *Hyperbolic geometry and closed bosonic string field theory. Part I. The string vertices via hyperbolic Riemann surfaces*, *JHEP* **08** (2019) 157 [arXiv:1706.07366] [INSPIRE].

[19] S.F. Moosavian and R. Pius, *Hyperbolic geometry and closed bosonic string field theory. Part II. The rules for evaluating the quantum BV master action*, *JHEP* **08** (2019) 177 [arXiv:1708.04977] [INSPIRE].

[20] K. Costello and B. Zwiebach, *Hyperbolic string vertices*, *JHEP* **02** (2022) 002 [arXiv:1909.00033] [INSPIRE].

[21] M. Cho, *Open-closed Hyperbolic String Vertices*, *JHEP* **05** (2020) 046 [arXiv:1912.00030] [INSPIRE].

[22] A.H. Fırat, *Hyperbolic three-string vertex*, *JHEP* **08** (2021) 035 [arXiv:2102.03936] [INSPIRE].

[23] P. Wang, H. Wu and H. Yang, *Connections between reflected entropies and hyperbolic string vertices*, *JHEP* **05** (2022) 127 [arXiv:2112.09503] [INSPIRE].

[24] N. Ishibashi, *The Fokker-Planck formalism for closed bosonic strings*, *PTEP* **2023** (2023) 023B05 [arXiv:2210.04134] [INSPIRE].

[25] A. Sen, *Tachyon condensation on the brane anti-brane system*, *JHEP* **08** (1998) 012 [hep-th/9805170] [INSPIRE].

[26] A. Sen, *Universality of the tachyon potential*, *JHEP* **12** (1999) 027 [hep-th/9911116] [INSPIRE].

[27] A. Sen and B. Zwiebach, *Tachyon condensation in string field theory*, *JHEP* **03** (2000) 002 [hep-th/9912249] [INSPIRE].

[28] N. Moeller and W. Taylor, *Level truncation and the tachyon in open bosonic string field theory*, *Nucl. Phys. B* **583** (2000) 105 [hep-th/0002237] [INSPIRE].

[29] N. Berkovits, A. Sen and B. Zwiebach, *Tachyon condensation in superstring field theory*, *Nucl. Phys. B* **587** (2000) 147 [hep-th/0002211] [INSPIRE].

[30] N. Moeller, A. Sen and B. Zwiebach, *D-branes as tachyon lumps in string field theory*, *JHEP* **08** (2000) 039 [hep-th/0005036] [INSPIRE].

[31] A. Sen and B. Zwiebach, *Large marginal deformations in string field theory*, *JHEP* **10** (2000) 009 [hep-th/0007153] [INSPIRE].

[32] L. Rastelli, A. Sen and B. Zwiebach, *Classical solutions in string field theory around the tachyon vacuum*, *Adv. Theor. Math. Phys.* **5** (2002) 393 [hep-th/0102112] [INSPIRE].

JHEP04(2024)016

[33] W. Taylor, *A perturbative analysis of tachyon condensation*, *JHEP* **03** (2003) 029 [hep-th/0208149] [INSPIRE].

[34] D. Gaiotto and L. Rastelli, *Experimental string field theory*, *JHEP* **08** (2003) 048 [hep-th/0211012] [INSPIRE].

[35] K. Strebel, *Quadratic Differentials*, in *Quadratic Differentials*, Springer Berlin Heidelberg (1984), p. 16–26 [DOI:10.1007/978-3-662-02414-0_2].

[36] F. Ruehle, *Evolving neural networks with genetic algorithms to study the String Landscape*, *JHEP* **08** (2017) 038 [arXiv:1706.07024] [INSPIRE].

[37] J. Halverson, B. Nelson and F. Ruehle, *Branes with Brains: Exploring String Vacua with Deep Reinforcement Learning*, *JHEP* **06** (2019) 003 [arXiv:1903.11616] [INSPIRE].

[38] M. Larfors and R. Schneider, *Explore and Exploit with Heterotic Line Bundle Models*, *Fortsch. Phys.* **68** (2020) 2000034 [arXiv:2003.04817] [INSPIRE].

[39] A. Ashmore and F. Ruehle, *Moduli-dependent KK towers and the swampland distance conjecture on the quintic Calabi-Yau manifold*, *Phys. Rev. D* **103** (2021) 106028 [arXiv:2103.07472] [INSPIRE].

[40] M.R. Douglas, S. Lakshminarasimhan and Y. Qi, *Numerical Calabi-Yau metrics from holomorphic networks*, arXiv:2012.04797 [INSPIRE].

[41] L.B. Anderson et al., *Moduli-dependent Calabi-Yau and* SU(3)*-structure metrics from Machine Learning*, *JHEP* **05** (2021) 013 [arXiv:2012.04656] [INSPIRE].

[42] H. Erbin and R. Finotello, *Inception neural network for complete intersection Calabi-Yau 3-folds*, *Mach. Learn. Sci. Tech.* **2** (2021) 02LT03 [arXiv:2007.13379] [INSPIRE].

[43] H. Erbin and R. Finotello, *Machine learning for complete intersection Calabi-Yau manifolds: a methodological study*, *Phys. Rev. D* **103** (2021) 126014 [arXiv:2007.15706] [INSPIRE].

[44] H. Erbin, R. Finotello, R. Schneider and M. Tamaazousti, *Deep multi-task mining Calabi-Yau four-folds*, *Mach. Learn. Sci. Tech.* **3** (2022) 015006 [arXiv:2108.02221] [INSPIRE].

[45] A. Ashmore, L. Calmon, Y.-H. He and B.A. Ovrut, *Calabi-Yau Metrics, Energy Functionals and Machine-Learning*, *International Journal of Data Science in the Mathematical Sciences* **1** (2023) 49 [arXiv:2112.10872] [INSPIRE].

[46] M. Larfors, A. Lukas, F. Ruehle and R. Schneider, *Learning Size and Shape of Calabi-Yau Spaces*, arXiv:2111.01436 [INSPIRE].

[47] M. Larfors, A. Lukas, F. Ruehle and R. Schneider, *Numerical metrics for complete intersection and Kreuzer-Skarke Calabi-Yau manifolds*, *Mach. Learn. Sci. Tech.* **3** (2022) 035014 [arXiv:2205.13408] [INSPIRE].

[48] V. Jejjala, W. Taylor and A. Turner, *Identifying equivalent Calabi-Yau topologies: A discrete challenge from math and physics for machine learning*, in the proceedings of the *Nankai Symposium on Mathematical Dialogues: In celebration of S.S.Chern's 110th anniversary*, China, August 02–13 (2021) [arXiv:2202.07590] [INSPIRE].

[49] J. Halverson, A. Maiti and K. Stoner, *Neural Networks and Quantum Field Theory*, *Mach. Learn. Sci. Tech.* **2** (2021) 035002 [arXiv:2008.08601] [INSPIRE].

[50] J. Halverson, *Building Quantum Field Theories Out of Neurons*, arXiv:2112.04527 [INSPIRE].

[51] H. Erbin, V. Lahoche and D.O. Samary, *Non-perturbative renormalization for the neural network-QFT correspondence*, *Mach. Learn. Sci. Tech.* **3** (2022) 015027 [arXiv:2108.01403] [INSPIRE].

JHEP04(2024)016

[52] F. Ruehle, *Data science applications to string theory*, *Phys. Rept.* **839** (2020) 1 [INSPIRE].

[53] J. Schmidt-Hieber, *Nonparametric regression using deep neural networks with ReLU activation function*, arXiv:1708.06633 [DOI:10.1214/19-AOS1875].

[54] K. Zhang and Y.-X. Wang, *Deep Learning meets Nonparametric Regression: Are Weight-Decayed DNNs Locally Adaptive?*, arXiv:2204.09664.

[55] R. Balestriero, J. Pesenti and Y. LeCun, *Learning in High Dimension Always Amounts to Extrapolation*, arXiv:2110.09485.

[56] K. Xu et al., *How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks*, arXiv:2009.11848.

[57] H. Erbin and A.H. Fırat, *Characterizing n-string contact interactions using machine learning*, to appear.

[58] B. Zwiebach, *Consistency of Closed String Polyhedra From Minimal Area*, *Phys. Lett. B* **241** (1990) 343 [INSPIRE].

[59] B. Zwiebach, *How covariant closed string theory solves a minimal area problem*, *Commun. Math. Phys.* **136** (1991) 83 [INSPIRE].

[60] M. Saadi and B. Zwiebach, *Closed String Field Theory from Polyhedra*, *Annals Phys.* **192** (1989) 213 [INSPIRE].

[61] T. Kugo, H. Kunitomo and K. Suehiro, *Nonpolynomial Closed String Field Theory*, *Phys. Lett. B* **226** (1989) 48 [INSPIRE].

[62] E. Witten, *Noncommutative Geometry and String Field Theory*, *Nucl. Phys. B* **268** (1986) 253 [INSPIRE].

[63] H. Sonoda and B. Zwiebach, *Closed String Field Theory Loops With Symmetric Factorizable Quadratic Differentials*, *Nucl. Phys. B* **331** (1990) 592 [INSPIRE].

[64] R. Gopakumar, *From free fields to AdS: III*, *Phys. Rev. D* **72** (2005) 066008 [hep-th/0504229] [INSPIRE].

[65] M.R. Gaberdiel, R. Gopakumar, B. Knighton and P. Maity, *From symmetric product CFTs to AdS_3*, *JHEP* **05** (2021) 073 [arXiv:2011.10038] [INSPIRE].

[66] F. Bhat, R. Gopakumar, P. Maity and B. Radhakrishnan, *Twistor coverings and Feynman diagrams*, *JHEP* **05** (2022) 150 [arXiv:2112.05115] [INSPIRE].

[67] B. Knighton, *Classical geometry from the tensionless string*, *JHEP* **05** (2023) 005 [arXiv:2207.01293] [INSPIRE].

[68] E.W. Dijkstra, *A Note on Two Problems in Connexion with Graphs*, in *Edsger Wybe Dijkstra*, ACM (2022), p. 287–290 [DOI:10.1145/3544585.3544600].

[69] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, *Math. Control Signals Syst.* **2** (1989) 303 [INSPIRE].

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

[71] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, *Neural Networks* **4** (1991) 251 [INSPIRE].

[72] M. Leshno, V.Y. Lin, A. Pinkus and S. Schocken, *Multilayer feedforward networks with a nonpolynomial activation function can approximate any function*, *Neural Networks* **6** (1993) 861.

JHEP04(2024)016

[73] B.C. Csáji, *Approximation with Artificial Neural Networks*, Ph.D. Thesis, Eötvös Loránd University, H-1053 Budapest, Hungary (2001).

[74] C. Trabelsi et al., *Deep Complex Networks*, arXiv:1705.09792 (2018).

[75] S. Scardapane, S.V. Vaerenbergh, A. Hussain and A. Uncini, *Complex-valued Neural Networks with Non-parametric Activation Functions*, arXiv:1802.08026.

[76] J. Bassey, L. Qian and X. Li, *A Survey of Complex-Valued Neural Networks*, arXiv:2101.12249.

[77] J. Bradbury et al., *JAX: composable transformations of Python+NumPy programs*, (2018).

[78] E.J. Michaud, Z. Liu and M. Tegmark, *Precision Machine Learning*, arXiv:2210.13447 [DOI:10.3390/e25010175].

[79] T.S. Cohen and M. Welling, *Group Equivariant Convolutional Networks*, arXiv:1602.07576 [INSPIRE].

[80] T. Cohen, M. Geiger and M. Weiler, *A General Theory of Equivariant CNNs on Homogeneous Spaces*, arXiv:1811.02017.

[81] J.E. Gerken et al., *Geometric Deep Learning and Equivariant Neural Networks*, arXiv:2105.13926 [INSPIRE].

[82] J. Zhou et al., *Graph Neural Networks: A Review of Methods and Applications*, arXiv:1812.08434 [INSPIRE].

[83] M. Headrick and B. Zwiebach, *Minimal-area metrics on the Swiss cross and punctured torus*, *Commun. Math. Phys.* **377** (2020) 2287 [arXiv:1806.00450] [INSPIRE].

[84] M. Headrick and B. Zwiebach, *Convex programs for minimal-area problems*, *Commun. Math. Phys.* **377** (2020) 2217 [arXiv:1806.00449] [INSPIRE].

[85] U. Naseer and B. Zwiebach, *Extremal isosystolic metrics with multiple bands of crossing geodesics*, *Adv. Theor. Math. Phys.* **26** (2022) 1273 [arXiv:1903.11755] [INSPIRE].

[86] M. Abramowitz and I.A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, US Government printing office **55** (1964).

[87] S. Falkner, A. Klein and F. Hutter, *BOHB: Robust and efficient hyperparameter optimization at scale*, in *Proceedings of the 35th International Conference on Machine Learning*, Stockholmsmässan, Stockholm Sweden, July 10–15 (2018) [J. Dy and A. Krause eds., PMLR **80** (2018), p.1437–1446].

JHEP04(2024)016