Building string field theory using machine learning

Harold Erbin

CEA-LIST (France), MIT & IAIFI (USA)

2022-2023

In collaboration with:

– Atakan Hilmi Fırat (MIT, IAIFI)

arXiv: 2211.09129

Massachusetts Institute of Technology

Funded by the European Union (Horizon 2020)

Outline: 1. Introduction

Talk highlights

- string field theory (SFT)
  - 2nd quantized formulation of string theory
  - amplitude $= 2d$ conformal field theory (CFT) correlation function integrated over moduli space of Riemann surfaces

Talk highlights

- string field theory (SFT)
  - 2nd quantized formulation of string theory
  - amplitude = $2d$ conformal field theory (CFT) correlation function integrated over moduli space of Riemann surfaces
- interactions = string vertices
  - characterized by 1) local coordinates, 2) moduli subspace
  - Feynman diagram sum = moduli space covering

Talk highlights

- string field theory (SFT)
  - 2nd quantized formulation of string theory
  - amplitude = $2d$ conformal field theory (CFT) correlation function integrated over moduli space of Riemann surfaces
- interactions = string vertices
  - characterized by 1) local coordinates, 2) moduli subspace
  - Feynman diagram sum = moduli space covering
- minimal area vertices (classical level)
  - built from Strebel quadratic differential
  - parametrized by accessory parameters (uniformization)
  - compute mapping radii $\rightarrow$ local coordinates
  - extract vertex region

Talk highlights

- string field theory (SFT)
  - 2nd quantized formulation of string theory
  - amplitude = $2d$ conformal field theory (CFT) correlation function integrated over moduli space of Riemann surfaces
- interactions = string vertices
  - characterized by 1) local coordinates, 2) moduli subspace
  - Feynman diagram sum = moduli space covering
- minimal area vertices (classical level)
  - built from Strebel quadratic differential
  - parametrized by accessory parameters (uniformization)
  - compute mapping radii $\rightarrow$ local coordinates
  - extract vertex region
- use neural networks to parametrize accessory parameters and vertex region

From worldsheet string theory to string field theory (1)

- usual formulation: worldsheet
  - 1st-quantized (dynamics of a few strings)
  - various problems (on-shell, perturbative. . .)

From worldsheet string theory to string field theory (1)

- usual formulation: worldsheet
  - 1st-quantized (dynamics of a few strings)
  - various problems (on-shell, perturbative...)
- 2nd quantization $\rightarrow$ string field theory (SFT)
  - modern language and tools of field theory
  - constructive, symmetries manifest
  - prove consistency (unitarity, analyticity, finiteness...)
  - study backgrounds (independence, fluxes...)

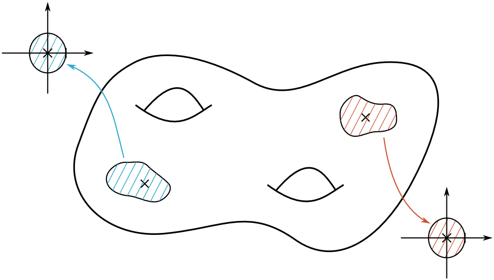
From worldsheet string theory to string field theory (1)

- usual formulation: worldsheet
  - 1st-quantized (dynamics of a few strings)
  - various problems (on-shell, perturbative. . .)
- 2nd quantization $\to$ string field theory (SFT)
  - modern language and tools of field theory
  - constructive, symmetries manifest
  - prove consistency (unitarity, analyticity, finiteness. . .)
  - study backgrounds (independence, fluxes. . .)
- problems
  - action: non-local, non-polynomial, ∞ number of fields
  - general properties known, but not explicit form

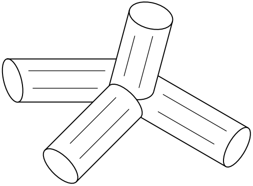
From worldsheet string theory to string field theory (2)

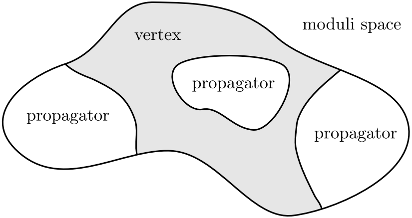
$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\begin{array}{c}\text{Riemann surface}\\ \text{moduli spaces}\end{array} \arrow[r, "2d\ \text{CFT}"] \arrow[d] & \begin{array}{c}\text{on-shell}\\ \text{string amplitudes}\end{array} \arrow[d] \\
\begin{array}{c}\text{Riemann surface}\\ \text{moduli spaces}\\ +\ \text{local coordinates}\end{array} \arrow[r] \arrow[d, "\begin{array}{c}\text{decomposition}\\ \text{by sewing}\end{array}"'] & \begin{array}{c}\text{off-shell}\\ \text{string amplitudes}\end{array} \arrow[d, "\begin{array}{c}\text{decomposition}\\ \text{in Feynman diagrams}\end{array}"'] \\
\begin{array}{c}\text{geometric}\\ \text{BV algebra}\end{array} \arrow[r, "\text{BV morphism}"'] & \begin{array}{c}\text{string field theory}\\ \text{(quantum BV)}\end{array}
\end{tikzcd}
$$

Local coordinates and moduli space decomposition

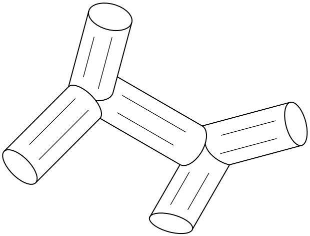


fundamental vertex





graph with propagator



Building string vertices with machine learning

Objective (physics)

Construct action using machine learning in order to extract numbers from SFT (in particular, closed string tachyon vacuum).

Building string vertices with machine learning

Objective (physics)

Construct action using machine learning in order to extract numbers from SFT (in particular, closed string tachyon vacuum).

Objective (math)

Construct functions on and subspaces of moduli space of Riemann surfaces using machine learning.

Tachyon vacuum

- main application: study closed string tachyon vacuum (settle existence or not)
- method
  - perform level-truncation (keep fields up to some mass)
  - compute potential up to some order in g_s
  - integrate out other fields (except dilaton)
  - extrapolate in level and order of interaction

Tachyon vacuum

- main application: study closed string tachyon vacuum (settle existence or not)
- method
  - perform level-truncation (keep fields up to some mass)
  - compute potential up to some order in g_s
  - integrate out other fields (except dilaton)
  - extrapolate in level and order of interaction
- truncated tachyon potential

$$V(t) = -t^2 + \sum_{n\geq 3} \frac{v_n}{n!}\, t^n, \qquad v_4 \approx 72.32 \pm 0.15$$

  previous results: $v_4 \approx 72.39$

  [hep-th/9412106, Belopolsky; hep-th/0408067, Moeller]
- other backgrounds: twisted tachyons on $\mathbb{C}/\mathbb{Z}_N \ldots$

  [hep-th/0111004, Dabholkar; hep-th/0403051, Okawa-Zwiebach]

Outline: 2. String field theory

Classical string field theory action

- string background: $2d$ conformal field theory (CFT)
- string field $\Psi \in \mathcal{H}$ (1st-quantized CFT Hilbert space)
- classical action

$$S = \frac{1}{2} \langle \Psi, Q_B \Psi \rangle + \sum_{n \geq 3} \frac{g^{n-2}}{n!} \mathcal{V}_n(\Psi^n)$$

- 1st-quantized BRST operator $Q_B : \mathcal{H} \to \mathcal{H}$
- string vertices $\mathcal{V}_n : \mathcal{H}^{\otimes n} \to \mathbb{C}$ (contact interactions)

Example: ϕ^4 scalar field

- action

$$
\begin{aligned}
S = \frac{1}{2} \int \mathrm{d}^d k \, \phi(-k)(k^2 + m^2)\phi(k) \\
\quad + \frac{\lambda}{4!} \int \mathrm{d}^d k_1 \cdots \mathrm{d}^d k_4 \, \delta^{(d)}(k_1 + \cdots + k_4)\, \phi(k_1) \cdots \phi(k_4) \\
= \frac{1}{2} \, \langle \phi, K\phi \rangle + \frac{\lambda}{4!} \, \mathcal{V}_4(\phi^4)
\end{aligned}
$$

- 1st-quantized momentum state basis $\{|k\rangle\}$

$$
|\phi\rangle = \int \mathrm{d}^d k \, \phi(k) \, |k\rangle \, , \qquad \langle k, k' \rangle = \delta^{(d)}(k + k')
$$

- Klein–Gordon operator $K = (p^2 + m^2)$
- quartic vertex

$$
\mathcal{V}_4(\phi^4) = \int \mathrm{d}^d k_1 \cdots \mathrm{d}^d k_4 \, V_4(k_1, \ldots, k_4)\, \phi(k_1) \cdots \phi(k_4)
$$

$$
V_4(k_1, \ldots, k_4) = \delta^{(d)}(k_1 + \cdots + k_4)
$$

String amplitudes

- off-shell n-point string amplitude with external states $A_i \in \mathcal{H}$

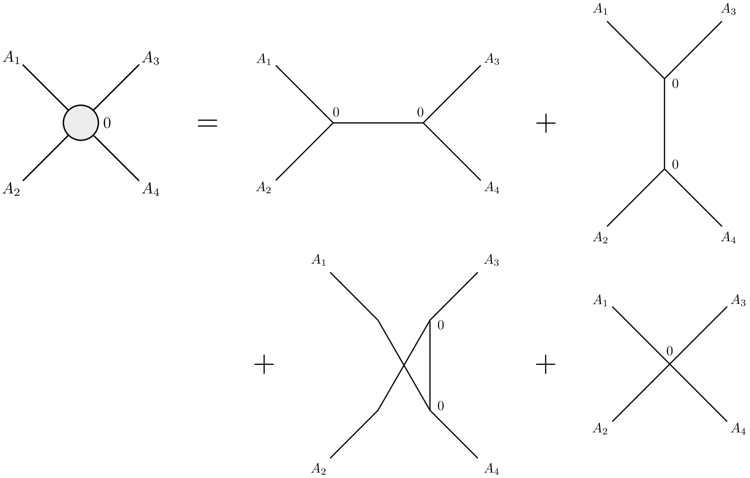
$$\mathcal{A}_n(A_1, \dots, A_n) = \int_{\mathcal{M}_n} \mathrm{d}^{n-3}\xi \, \left\langle \text{ghosts} \times \prod_i f_{n,i} \circ A_i(0) \right\rangle_{\Sigma_n}$$

  - $\langle \cdots \rangle$ CFT correlation function
  - sum over topologically inequivalent spheres Σ_n with n punctures at $(\xi_1, \dots, \xi_n)$
  - can fix 3 points $(\xi_{n-2}, \xi_{n-1}, \xi_n) = (0, 1, \infty)$
  - $\xi_\lambda \in \mathcal{M}_n \sim \mathbb{C}^{n-3}$ $(\lambda = 1, \dots, n-3)$ moduli space
  - ghosts: 1) measure over $\mathcal{M}_n$, 2) needed for BRST invariance
  - $f_{n,i}(w_i; \xi_\lambda)$ local coordinates = conformal maps

$$f_{n,i}(0; \xi_\lambda) := \xi_i, \qquad f_{n,i} \circ A_i(0) := |f'_{n,i}(0)|^{2h_i} A_i\big(f_{n,i}(0)\big)$$

  if A_i is primary with weight (h_i, h_i)

Amplitude and Feynman diagrams



$$\mathcal{A}_4 \qquad = \qquad \mathcal{F}_4^{(s)} + \mathcal{F}_4^{(t)} + \mathcal{F}_4^{(u)} \quad + \qquad \mathcal{V}_4$$

String vertices

- string vertex

$$\mathcal{V}_n(A_1, \ldots, A_n) = \int_{\mathcal{V}_n} \mathrm{d}^{n-3}\xi \left\langle \text{ghosts} \times \prod_i f_{n,i} \circ A_i(0) \right\rangle_{\Sigma_n}$$

String vertices

- string vertex

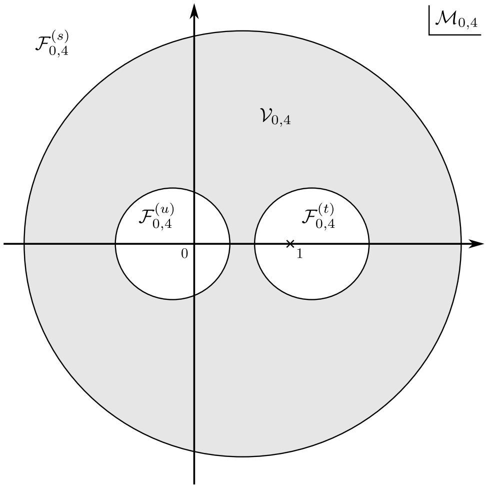
$$\mathcal{V}_n(A_1, \dots, A_n) = \int_{\mathcal{V}_n} \mathrm{d}^{n-3}\xi \left\langle \text{ghosts} \times \prod_i f_{n,i} \circ A_i(0) \right\rangle_{\Sigma_n}$$

- defined such that

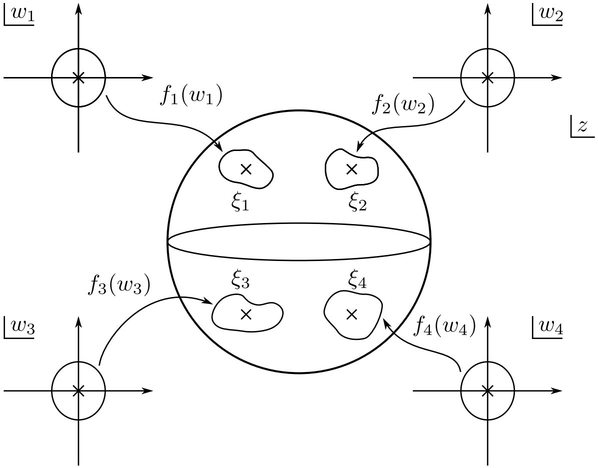
$$\mathcal{A}_n(A_1, \dots, A_n) = \mathcal{F}_n(A_1, \dots, A_n) + \mathcal{V}_n(A_1, \dots, A_n)$$

  $\mathcal{F}_n$ contributions from Feynman diagrams (Riemann surfaces) containing propagators (long tubes) and $\mathcal{V}_{n'}$ with $n' < n$
- $\mathcal{V}_n \subset \mathcal{M}_n$: vertex region $\subset$ moduli space
- constraints between all $\{f_{n,i}\}$ (“gluing compatibility”)

Moduli space covering



Local coordinates



Motivation: restore $\mathrm{SL}(2,\mathbb{C})$ invariance, broken by punctures (transformation between patches)

How to build vertices

- $\mathrm{SL}(2,\mathbb{C})$ vertices
  - $n = 3$ sphere: simplest vertex
  - $n = 4$ sphere: analytical $\mathcal{V}_4$ boundary, no explicit coordinates [HE, in progress]
  - $n = 1$ torus: analytical vertex boundary, no explicit coordinates [1704.01210, Erler-Konopka-Sachs]
- hyperbolic vertices [1706.07366, Moosavian-Pius; 1909.00033, Costello-Zwiebach; 2102.03936, Fırat]
- minimal area string vertices: optimal representation [Zwiebach '91; hep-th/9206084, Zwiebach]

How to build vertices

- $\mathrm{SL}(2,\mathbb{C})$ vertices
  - $n = 3$ sphere: simplest vertex
  - $n = 4$ sphere: analytical $\mathcal{V}_4$ boundary, no explicit coordinates [HE, in progress]
  - $n = 1$ torus: analytical vertex boundary, no explicit coordinates [1704.01210, Erler-Konopka-Sachs]
- hyperbolic vertices [1706.07366, Moosavian-Pius; 1909.00033, Costello-Zwiebach; 2102.03936, Fırat]
- minimal area string vertices: optimal representation [Zwiebach '91; hep-th/9206084, Zwiebach]
- note: superstring vertices can be obtained by dressing bosonic vertices [hep-th/0409018, Berkovits-Okawa-Zwiebach; 1403.0940, Erler-Konopka-Sachs]

Outline: 3. Minimal area vertices

Minimal area vertex

- vertex constructed from minimal area metric with bounds on length of shortest closed geodesic (systole) and heights of internal foliation [Zwiebach '90]
- *n*-punctured sphere vertex: construct metric from Strebel quadratic differential [Saadi-Swiebach '89]
  - fixed up to moduli-dependent parameters
  - lead to contact interactions (internal foliation height $= 0$)

Minimal area vertex

- vertex constructed from minimal area metric with bounds on length of shortest closed geodesic (systole) and heights of internal foliation [Zwiebach '90]
- n-punctured sphere vertex: construct metric from Strebel quadratic differential [Saadi-Swiebach '89]
  - fixed up to moduli-dependent parameters
  - lead to contact interactions (internal foliation height $= 0$)
- goal: $\forall n \geq 3$ obtain $f_{n,i}$ and $\mathcal{V}_n$
- state-of-the-art:
  - analytic solution for $n = 3$, numerical for $n = 4, 5$ [Moeller, hep-th/0408067, hep-th/0609209]
  - convex program for any genus and n, but not implemented and not restricted to vertex region [1806.00449, Headrick-Zwiebach]

Quadratic differential

- quadratic differential $\varphi = \phi(z)\mathrm{d}z^2$ [Strebel, '84]

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

$$0 = \sum_{i=1}^{n} c_i = \sum_{i=1}^{n} (-1 + c_i \xi_i) = \sum_{i=1}^{n} (-2\xi_i + c_i \xi_i^2)$$

- $c_i(\xi_i, \bar{\xi}_i)$ accessory parameters
  (limit from Liouville accessory parameters)
- constraints: regularity at $z = \infty$

Quadratic differential

- quadratic differential $\varphi = \phi(z)\mathrm{d}z^2$ [Strebel, '84]

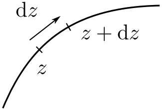
$$\phi(z) = \sum_{i=1}^{n} \left[\frac{-1}{(z-\xi_i)^2} + \frac{c_i}{z-\xi_i} \right]$$

$$0 = \sum_{i=1}^{n} c_i = \sum_{i=1}^{n} (-1 + c_i \xi_i) = \sum_{i=1}^{n} (-2\xi_i + c_i \xi_i^2)$$

- $c_i(\xi_i, \bar{\xi}_i)$ accessory parameters
  (limit from Liouville accessory parameters)
- constraints: regularity at $z = \infty$
- φ induces metric with semi-infinite flat cylinders around punctures (= external strings)

$$\mathrm{d}s^2 = |\phi(z)|^2 |\mathrm{d}z|^2, \qquad \mathrm{d}s^2|_{w_i} = \frac{|\mathrm{d}w_i|^2}{|w_i|^2}$$

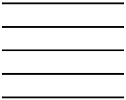
Critical trajectory



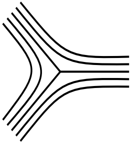
Definitions:

- $\{z_i(c_i, \xi_i)\}$ zeros of $\phi(z)$
- horizontal trajectory = path with $\varphi = \phi(z)\mathrm{d}z^2 > 0$

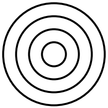
$\mathrm{d}z^2$



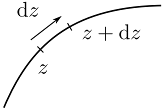
$z\,\mathrm{d}z^2$



$-\dfrac{\mathrm{d}z^2}{z^2}$

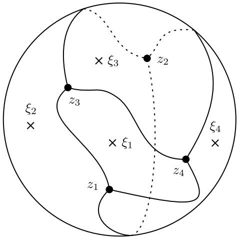


Critical trajectory

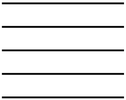


Definitions:

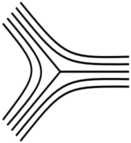
- $\{z_i(c_i, \xi_i)\}$ zeros of $\phi(z)$
- horizontal trajectory = path with $\varphi = \phi(z)\mathrm{d}z^2 > 0$
- critical trajectory = horizontal trajectory with ends at $\phi(z) = 0$
- critical graph = {critical trajectories}



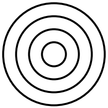
$\mathrm{d}z^2$



$z\,\mathrm{d}z^2$



$-\frac{\mathrm{d}z^2}{z^2}$

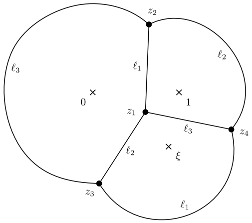


Strebel quadratic differential

Strebel quadratic differential

Quadratic differential such that its critical graph is:

1. a polyhedron (measure zero):
   - vertices = zeros
   - edges = critical trajectories
   - faces = punctures
2. connected (no propagator = long tube)



$$\xi = 0.87 - 0.62\mathrm{i}$$

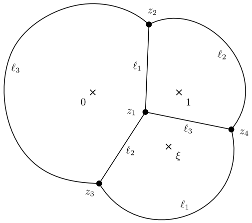
Strebel quadratic differential

Strebel quadratic differential

Quadratic differential such that its critical graph is:

1. a polyhedron (measure zero):
   - vertices = zeros
   - edges = critical trajectories
   - faces = punctures
2. connected (no propagator = long tube)

- unique given ξ_λ
- define minimal area metric
- defines string vertices
  - provide local coordinates
  - allow determining vertex region



$\xi = 0.87 - 0.62\mathrm{i}$

Computing the accessory parameter

- hard mathematical problem (related to Fuchsian uniformization, Liouville theory...)
- complex length between two points

$$\ell(a, b) = \int_a^b \mathrm{d}z \sqrt{\phi(z)}$$

Computing the accessory parameter

- hard mathematical problem (related to Fuchsian uniformization, Liouville theory...)
- complex length between two points

$$\ell(a, b) = \int_a^b \mathrm{d}z \sqrt{\phi(z)}$$

- Strebel differential: necessary and sufficient condition

$$\forall (z_i, z_j): \quad \mathrm{Im}\, \ell(z_i, z_j) = 0$$

Computing the accessory parameter

- hard mathematical problem (related to Fuchsian uniformization, Liouville theory. . .)
- complex length between two points

$$\ell(a, b) = \int_a^b \mathrm{d}z \sqrt{\phi(z)}$$

- Strebel differential: necessary and sufficient condition

$$\forall (z_i, z_j) : \quad \operatorname{Im} \ell(z_i, z_j) = 0$$

- for fixed ξ_i, give equations on c_i
- [Moeller, hep-th/0408067, hep-th/0609209]: solve point by point using Newton method for $n = 4, 5$ (and fit for $n = 4$)

Local coordinates

- Strebel critical graph defines local coordinates
  $\rightarrow$ map $|w_i| = 1$ to critical trajectory around ξ_i

Local coordinates

- Strebel critical graph defines local coordinates
  $\rightarrow$ map $|w_i| = 1$ to critical trajectory around ξ_i
- series expansion

$$z = f_{n,i}(w_i) = \xi_i + \rho_i w_i + \sum_{k \geq 2} d_{i,k-1} (\rho_i w_i)^k$$

$$\varphi \sim_{\xi_i} \left(-\frac{1}{(z-\xi_i)^2} + \sum_{k \geq -1} b_{i,k} (z - \xi_i)^k \right) \mathrm{d}z^2 = -\frac{\mathrm{d}w_i^2}{w_i^2}$$

  where $b_{i,k} = b_{i,k}(c_i, \xi_i)$, e.g. $b_{i,-1} = c_i$
- match coefficients

$$d_{i,1} = \frac{b_{i,-1}}{2}, \qquad d_{i,2} = \frac{1}{16}\left(7 b_{i,-1}^2 + 4 b_{i,0}\right), \qquad \ldots$$

Local coordinates

- Strebel critical graph defines local coordinates
  $\rightarrow$ map $|w_i| = 1$ to critical trajectory around ξ_i
- series expansion

$$z = f_{n,i}(w_i) = \xi_i + \rho_i w_i + \sum_{k\geq 2} d_{i,k-1}(\rho_i w_i)^k$$

$$\varphi \sim_{\xi_i} \left(-\frac{1}{(z-\xi_i)^2} + \sum_{k\geq -1} b_{i,k}(z-\xi_i)^k \right) \mathrm{d}z^2 = -\frac{\mathrm{d}w_i^2}{w_i^2}$$

  where $b_{i,k} = b_{i,k}(c_i, \xi_i)$, e.g. $b_{i,-1} = c_i$
- match coefficients

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

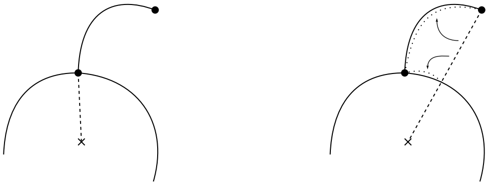
- remaining unknown: mapping radii $\rho_i \in \mathbb{R}$

Mapping radii

- mapping radius for ξ_i (conformal invariant)

$$\ln \rho_i = \ln \left| \frac{\mathrm{d}f_i}{\mathrm{d}w_i} \right|_{w_i=0} = \lim_{\epsilon \to 0} \left[\operatorname{Im} \int_{\xi_i+\epsilon}^{z_c} \mathrm{d}z \sqrt{\phi(z)} + \ln \epsilon \right]$$

- z_c is any point on critical graph (path after crossing closest trajectory does not contribute to imaginary part)
  $\rightarrow$ compute $z_c = z_i \, \forall i$, then average



Vertex region

- vertex region $=$ lengths of non-contractible curves $\geq 2\pi$

Vertex region

- vertex region = lengths of non-contractible curves $\geq 2\pi$
- determine shape (zeros on trajectory around each ξ_i) and distances of critical graph
- example: $n = 4$

$$\xi \in \mathcal{V}_4 \quad \Longleftrightarrow \quad \ell_1, \ell_2, \ell_3 \geq \pi$$

Vertex region

- vertex region $=$ lengths of non-contractible curves $\geq 2\pi$
- determine shape (zeros on trajectory around each ξ_i) and distances of critical graph
- example: $n = 4$

$$\xi \in \mathcal{V}_4 \quad \Longleftrightarrow \quad \ell_1, \ell_2, \ell_3 \geq \pi$$

- indicator function

$$\int_{\mathcal{V}_n} \cdots = \int_{\mathcal{M}_n} \Theta(\xi) \cdots, \qquad \Theta(\xi) := \begin{cases} 1 & \text{if } \xi \in \mathcal{V}_n \\ 0 & \text{if } \xi \notin \mathcal{V}_n \end{cases}$$

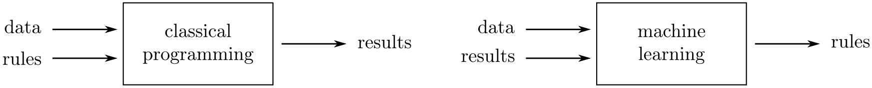
Outline: 4. Machine learning

Machine learning

Definition (Samuel)

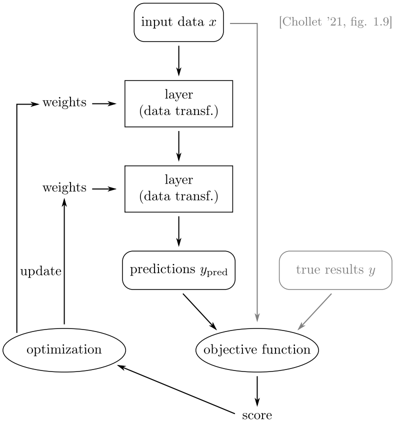
The field of study that gives computers the ability to learn without being explicitly programmed.

- approximate function $y = F(x)$ by some structure (neural network, decision tree. . .)
- agreement measured by some metric (distance, constraint. . .)
- tune structure parameters to improve approximation



Neural network

- neural network
  $=$ sequence of layers
  implementing computations
- layer
  - output $=$ different data representation
  - transformation parametrized by weights
- goal: find weights such that the network reproduces the target fonction $y = F(x)$
- comparison: objective function
- optimization by gradient descent
- general architecture defined by hyperparameters (number of layers...)



Why neural networks?

- generically outperform other machine learning approaches
- flexible inputs (complex numbers, graphs. . .)
- neural network = differentiable function
  - solve for the full function, not points one by one
  - better expressivity than fit
  - may extrapolate outside training region
  - classication task provides (probabilistic) measure
- transfer learning
- compact representation of the result, easily reused and shared

Learning the accessory parameter

- idea : $c_\lambda(\xi_\lambda) =$ complex neural network $C_\lambda(\xi_\lambda; \boldsymbol{W}, \boldsymbol{b})$
- $\boldsymbol{W}$ weights (complex matrices), $\boldsymbol{b}$ biases (complex vectors)

Learning the accessory parameter

- idea : $c_\lambda(\xi_\lambda) =$ complex neural network $C_\lambda(\xi_\lambda; \boldsymbol{W}, \boldsymbol{b})$
- $\boldsymbol{W}$ weights (complex matrices), $\boldsymbol{b}$ biases (complex vectors)
- unsupervised training with loss

$$\mathcal{L}(C_\lambda, \xi_\lambda) = \binom{2n-4}{2}^{-1} \sum_{i \geq j} \Big(\operatorname{Im} \ell(z_i, z_j) \Big)^2 \Big|_{c_\lambda = C_\lambda}$$

  $\rightarrow$ minimize with gradient descent
- for fixed ξ_λ, global minimum for any n with c_λ given by Strebel differential

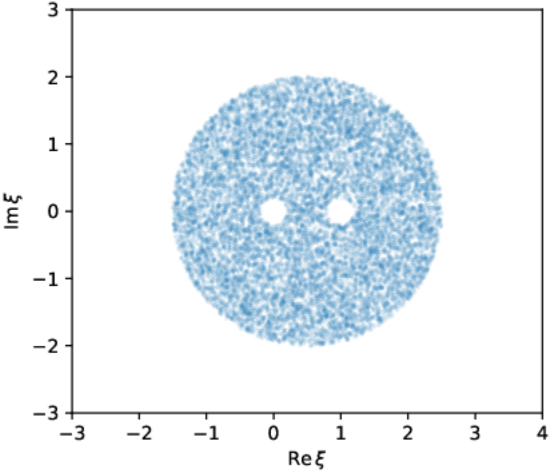
Learning the accessory parameter

- idea : $c_\lambda(\xi_\lambda) =$ complex neural network $C_\lambda(\xi_\lambda; \boldsymbol{W}, \boldsymbol{b})$
- $\boldsymbol{W}$ weights (complex matrices), $\boldsymbol{b}$ biases (complex vectors)
- unsupervised training with loss

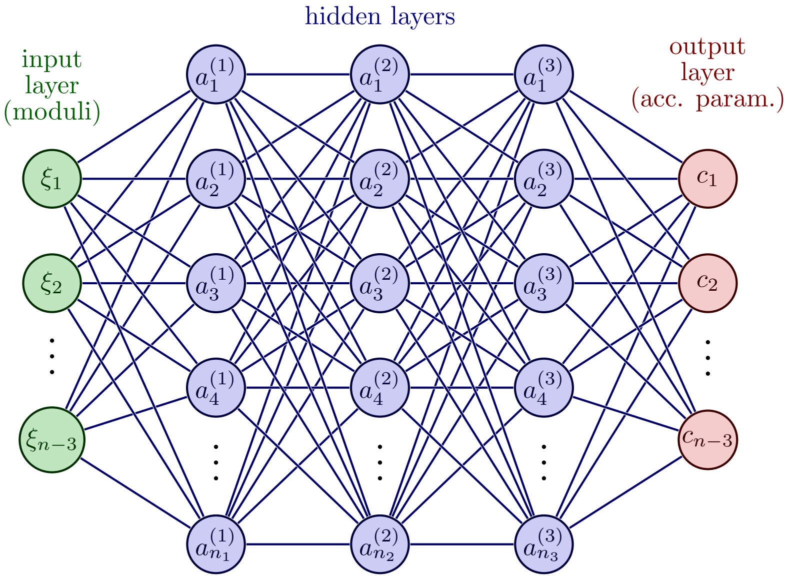
$$\mathcal{L}(C_\lambda, \xi_\lambda) = \binom{2n-4}{2}^{-1} \sum_{i \geq j} \Big(\operatorname{Im} \ell(z_i, z_j) \Big)^2 \Big|_{c_\lambda = C_\lambda}$$

  $\rightarrow$ minimize with gradient descent
- for fixed ξ_λ, global minimum for any n with c_λ given by Strebel differential
- training set: uniform sampling in $\mathcal{M}_n$ minus disks around fixed punctures $(\xi_{n-2}, \xi_{n-1}, \xi_n) = (0, 1, \infty)$

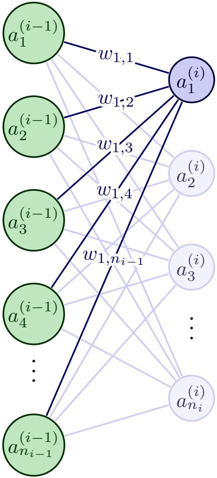
Data



Neural network architecture



Neural network architecture



$$= \sigma\left(w_{1,1}a_1^{(i-1)} + w_{1,2}a_2^{(i-1)} + \ldots + w_{1,n_{i-1}}a_{n_{i-1}}^{(i-1)} + b_1^{(i)}\right)$$

$$= \sigma\left(\sum_{k=1}^{n_{i-1}} w_{1,k}a_k^{(i-1)} + b_1^{(i)}\right)$$

$$\begin{pmatrix} a_1^{(i)} \\ a_2^{(i)} \\ \vdots \\ a_{n_i}^{(i)} \end{pmatrix} = \sigma\left[\begin{pmatrix} w_{1,0} & w_{1,1} & \ldots & w_{1,n_{i-1}} \\ w_{2,0} & w_{2,1} & \ldots & w_{2,n_{i-1}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_i,0} & w_{n_i,1} & \ldots & w_{n_i,n_{i-1}} \end{pmatrix}\begin{pmatrix} a_1^{(i-1)} \\ a_2^{(i-1)} \\ \vdots \\ a_{n_{i-1}}^{(i-1)} \end{pmatrix} + \begin{pmatrix} b_1^{(i)} \\ b_2^{(i)} \\ \vdots \\ b_{n_i}^{(i)} \end{pmatrix}\right]$$

$$\implies a^{(i)} = \sigma\left(W^{(i)}a^{(i-1)} + b^{(i)}\right)$$

4-punctured sphere

- notations for $n = 4$

$$\xi_1 := \xi \in \mathbb{C}, \qquad c_1 := a \in \mathbb{C},$$
$$\ell(z_1, z_2) := \ell_1, \qquad \ell(z_1, z_3) := \ell_2, \qquad \ell(z_1, z_4) := \ell_3$$

- analytic solutions

$$a(1/2) = 2, \qquad a\left(Q = \frac{1}{2} \pm \mathrm{i}\frac{\sqrt{3}}{2}\right) = 2 + \mathrm{i}\,\frac{2}{\sqrt{3}} \approx 2 + 1.1547\mathrm{i}$$
$$a(\xi \in \mathbb{R}) = \begin{cases} 0 & \xi \leq 0 \\ 4\xi & 0 \leq \xi \leq 1 \\ 4 & \xi \geq 1 \end{cases}$$

Results: 4-punctured sphere (1)

- neural network (Jax)
  - fully connected, 3 layers $(512, 128, 1028)$, $\mathbb{C}$ReLU activation

$$\mathbb{C}\mathrm{ReLU}(z) := \mathrm{ReLU}(\operatorname{Re} z) + \mathrm{i}\,\mathrm{ReLU}(\operatorname{Im} z)$$

  - training: 10^5 points, Adam, ℓ_2 regularization, weight decay, early stopping (~ 1000 epochs)

Results: 4-punctured sphere (1)

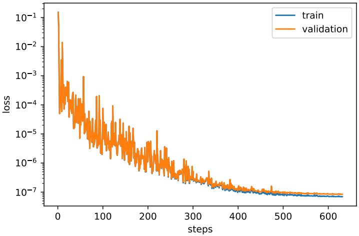
- neural network (Jax)
  - fully connected, 3 layers $(512, 128, 1028)$, $\mathbb{C}$ReLU activation

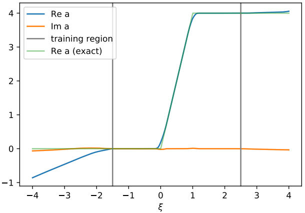
$$\mathbb{C}\mathrm{ReLU}(z) := \mathrm{ReLU}(\operatorname{Re} z) + \mathrm{i}\,\mathrm{ReLU}(\operatorname{Im} z)$$

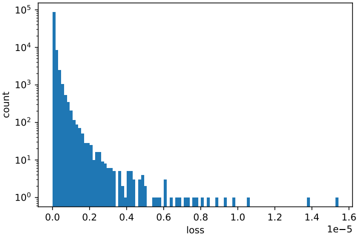
  - training: 10^5 points, Adam, ℓ_2 regularization, weight decay, early stopping (~ 1000 epochs)
- loss statistics (exact solution $\sim 10^{-12}$)
  - mean: $8.9 \cdot 10^{-8}$
  - median: $3.8 \cdot 10^{-8}$
  - min: $1.3 \cdot 10^{-11}$
  - max: $1.5 \cdot 10^{-5}$

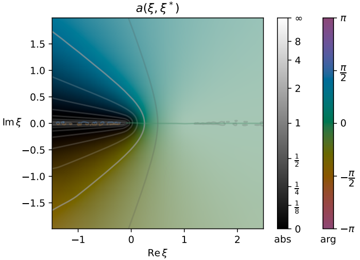
  note: already good performance with 10^3 points, 100 epochs (e.g. mean loss $= 2.7 \cdot 10^{-5}$)
- mean error compared to Moeller's fit: $5.5 \cdot 10^{-3}$

Results: 4-punctured sphere (2)





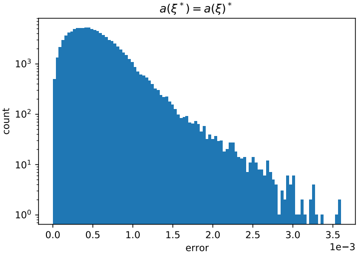


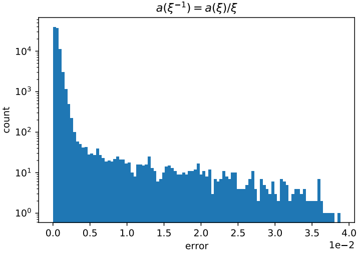


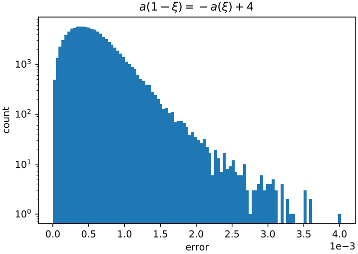
$$a(1/2) = 1.9995 + 0.0001\mathrm{i}, \qquad a(Q) = 1.9997 + 1.1548\mathrm{i}$$

$$\mathcal{L}(1/2) = 6.6 \cdot 10^{-7} \qquad \mathcal{L}(Q) = 6.1 \cdot 10^{-8}$$

Results: symmetries



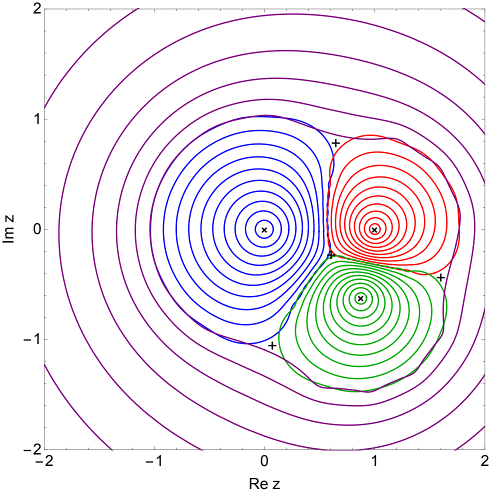




complex conjugation and permutation of fixed punctures

$$a(\xi^*) = a(\xi)^*$$
$$a(1-\xi) = 4 - a(\xi)$$
$$a(\xi^{-1}) = \frac{a(\xi)}{\xi}$$

Strebel differential



$\xi = 0.87 - 0.62\mathrm{i}$

Learning the vertex region

- idea: $\Theta(\xi) =$ neural network $\theta(\xi)$
  - $\theta(\xi)$ becomes probability distribution
  - useful for Monte Carlo integration
  - easily find boundary, e.g. $\theta(\xi) \in [0.2, 0.8]$

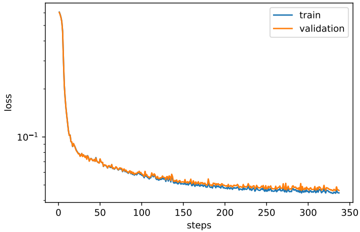
Learning the vertex region

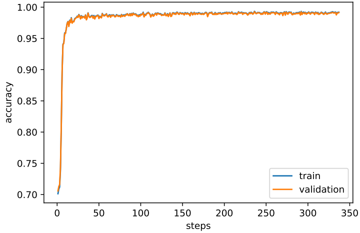
- idea: $\Theta(\xi)$ = neural network $\theta(\xi)$
  - $\theta(\xi)$ becomes probability distribution
  - useful for Monte Carlo integration
  - easily find boundary, e.g. $\theta(\xi) \in [0.2, 0.8]$
- supervised classification, binary cross-entropy loss

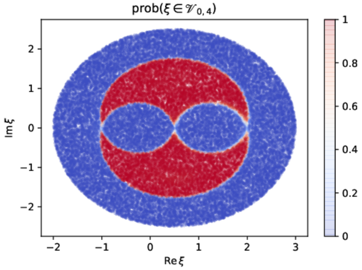
$$\mathcal{L}(\xi) = -\Theta(\xi) \ln \theta(\xi) - \big(1 - \Theta(\xi)\big) \ln \big(1 - \theta(\xi)\big)$$

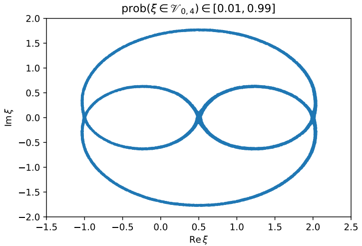
- neural network (Jax)
  - fully connected, 4 layers $(512, 32, 8, 8)$, ELU activation
  - training: 10^5 points, Adam, ℓ_2 regularization, weight decay, early stopping (~ 800 epochs)

Results: vertex region









Accuracy: 99.34 % (train set), 99.27 % (validation set), 99.68 % (test set)

Tachyon potential

Truncated tachyon potential (ignore other fields)

$$V(t) = -t^2 + \frac{v_3}{3!}\, t^3 - \frac{v_4}{4!}\, t^4 + \cdots$$

$$v_n := \mathcal{V}_n(T^n) = (-1)^n \,\frac{2}{\pi^{n-3}} \int_{\mathcal{V}_n} \mathrm{d}^{n-3}\xi \, \prod_{i=1}^{n} \frac{1}{\rho_i^2}$$

- mapping radii

$$\rho_i := \left| \frac{\mathrm{d}f_i}{\mathrm{d}w_i}(0) \right|$$

- $v_3 = -3^9/2^{11} \approx -9.61$
  [hep-th/9409015, Belopolsky-Zwiebach]

Tachyon potential

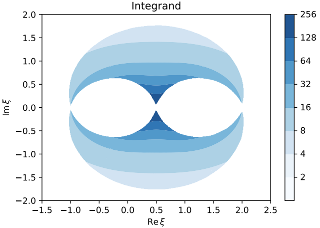
Truncated tachyon potential (ignore other fields)

$$V(t) = -t^2 + \frac{v_3}{3!}\,t^3 - \frac{v_4}{4!}\,t^4 + \cdots$$

$$v_n := \mathcal{V}_n(T^n) = (-1)^n \frac{2}{\pi^{n-3}} \int_{\mathcal{V}_n} \mathrm{d}^{n-3}\xi \prod_{i=1}^{n} \frac{1}{\rho_i^2}$$

- mapping radii

$$\rho_i := \left| \frac{\mathrm{d}f_i}{\mathrm{d}w_i}(0) \right|$$



- $v_3 = -3^9/2^{11} \approx -9.61$
  [hep-th/9409015, Belopolsky-Zwiebach]

Results

method	v_4
[hep-th/9412106, Belopolsky]	72.39
[hep-th/0408067, Moeller]	72.390
[hep-th/0506077, Yang-Zwiebach]	72.414
trapezoid (mean)	72.320 ± 0.146
trapezoid (best)	72.396
Monte Carlo (best)	72.366 ± 0.096

- ML statistics: train 10 neural networks, keep the ones (4) extrapolating well
- error in potential coefficient: $\sim 10^{-3}$
  $\rightarrow$ expect sufficiently precise for determining vacuum
- full pipeline: $\sim$ 4 hours

Outline: 5. Conclusion

Results and outlook

Results:

- new method to construct n-point string vertices
- implementation for $n = 4$ reproduces known results
- general method to compute functions extremizing some property

Results and outlook

Results:

- new method to construct n-point string vertices
- implementation for $n = 4$ reproduces known results
- general method to compute functions extremizing some property

Outlook:

- increase precision (note: difficult and non-standard ML problem!)
- generalize to $n \geq 5$
- compute closed string tachyon vacuum
- compute quadratic differentials for Feynman regions
- generalize to hyperbolic vertices
- generalize higher-genus surfaces (loop corrections) (compute mass renormalization and vacuum shift)