Building string field theory using machine learning

Harold Erbin

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

Ferdowsi university of Mashhad – 19 July 2023

In collaboration with:

- Atakan Hilmi Fırat (MIT, IAIFI)

arXiv: 2211.09129

Massachusetts Institute of Technology

Funded by the European Union (Horizon 2020)

list

Outline

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

Outline: 1. Introduction

From worldsheet string theory to string field theory (1)

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

From worldsheet string theory to string field theory (1)

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

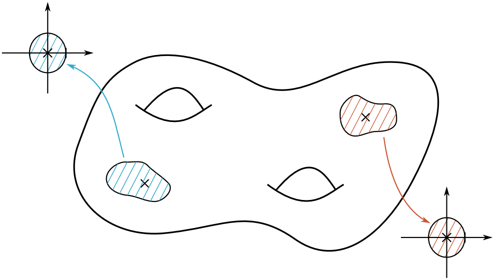
From worldsheet string theory to string field theory (1)

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

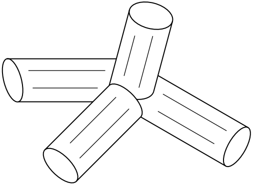
From worldsheet string theory to string field theory (2)

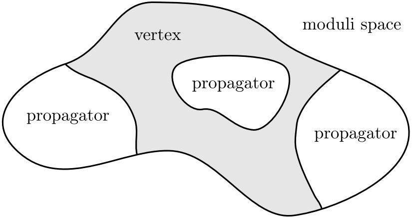
$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\begin{array}{c}\text{Riemann surface}\\ \text{moduli spaces}\end{array} \arrow[r, "2d\ \text{CFT}"] \arrow[d] & \begin{array}{c}\text{on-shell}\\ \text{string amplitudes}\end{array} \arrow[d] \\
\begin{array}{c}\text{Riemann surface}\\ \text{moduli spaces}\\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

Local coordinates and moduli space decomposition

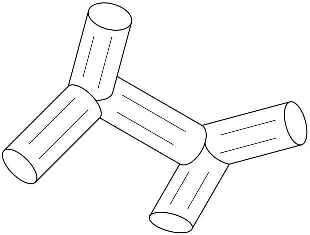


fundamental vertex





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 $\mathbb{C}/\mathbb{Z}_N \dots$

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

Outline: 2. String field theory

String background

- $2d$ conformal field theory (CFT)
  - conformally invariant non-linear sigma model of D non-compact scalar fields X^μ
    $\rightarrow$ spacetime with metric $G_{\mu\nu}$ and D non-compact dimensions
  - a generic internal matter CFT with central charge c_{int}
  - (b, c) anti-commuting ghosts with central charge $c_{\text{gh}} = -26$ from worldsheet reparametrizations
- string coupling g_s

[hep-th/9411047, Bergman-Zwiebach]

String background

- $2d$ conformal field theory (CFT)
  - conformally invariant non-linear sigma model of D non-compact scalar fields X^μ
    $\to$ spacetime with metric $G_{\mu\nu}$ and D non-compact dimensions
  - a generic internal matter CFT with central charge c_{int}
  - (b, c) anti-commuting ghosts with central charge $c_{\text{gh}} = -26$ from worldsheet reparametrizations
- string coupling g_s

[hep-th/9411047, Bergman-Zwiebach]

Notes

- total central charge: $D + c_{\text{int}} + c_{\text{gh}} = 0$
- free scalar fields $\Rightarrow$ flat spacetime

$$G_{\mu\nu} = \eta_{\mu\nu} = \text{diag}(\pm 1, \underbrace{1, \dots, 1}_{D-1}).$$

String field theory action

- string field $\Psi \in \mathcal{H}$ (1st-quantized CFT Hilbert space)
- level-matching constraints

$$b_0^- |\Psi\rangle = L_0^- |\Psi\rangle = 0$$

$$L_0^\pm = L_0 \pm \bar{L}_0, \quad b_0^\pm = b_0 \pm \bar{b}_0, \quad c_0^\pm = (c_0 \pm \bar{c}_0)/2$$

String field theory action

- string field $\Psi \in \mathcal{H}$ (1st-quantized CFT Hilbert space)
- level-matching constraints

$$b_0^- \left|\Psi\right\rangle = L_0^- \left|\Psi\right\rangle = 0$$

$$L_0^\pm = L_0 \pm \bar{L}_0, \quad b_0^\pm = b_0 \pm \bar{b}_0, \quad c_0^\pm = (c_0 \pm \bar{c}_0)/2$$

- quantum BV master action (prime: omit $g = 0, n = 1, 2, 3$)

$$S = \frac{1}{2} \left\langle \Psi, Q_B \Psi \right\rangle + \sum_{g,n \geq 0}{}' \frac{\hbar^g g_s^{2g-2+n}}{n!} \mathcal{V}_{g,n}(\Psi^n)$$

- $\langle \cdot, \cdot \rangle := \langle \cdot | c_0^- | \cdot \rangle$ (BPZ product)
- 1st-quantized BRST operator $Q_B : \mathcal{H} \to \mathcal{H}$
- string vertices $\mathcal{V}_{g,n} : \mathcal{H}^{\otimes n} \to \mathbb{C}$ (“contact” interactions)

Example: ϕ^4 scalar field

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

- 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

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

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

Gauge fixing and Feynman rules

- Siegel gauge

$$b_0^+ |\Psi\rangle = 0$$

- kinetic term

$$S_{\text{free,gf}} = \frac{1}{2} \langle\Psi| c_0^- c_0^+ L_0^+ |\Psi\rangle$$

Gauge fixing and Feynman rules

- Siegel gauge

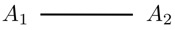
$$b_0^+ |\Psi\rangle = 0$$

- kinetic term

$$S_{\text{free,gf}} = \frac{1}{2} \langle\Psi| c_0^- c_0^+ L_0^+ |\Psi\rangle$$

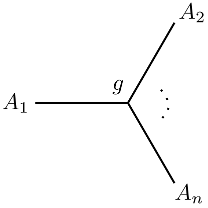
- propagator

$$\langle A_1| \frac{b_0^+}{L_0^+} b_0^- |A_2\rangle \quad =$$



- fundamental g-loop n-point vertex

$$\mathcal{V}_{g,n}(A_1, \ldots, A_n) \quad =$$



Momentum representation (1)

- string field Fourier expansion

$$|\Psi\rangle = \sum_A \int \frac{\mathrm{d}^D k}{(2\pi)^D}\, \phi_A(k)\, |A, k\rangle$$

k: D-dimensional momentum
A: discrete labels (Lorentz indices, group repr., KK modes...)

- 1PI action

$$S = \frac{1}{2}\int \mathrm{d}^D k\, \phi_A(k) K_{AB}(k) \phi_B(-k) + \sum_n \int \mathrm{d}^D k_1 \cdots \mathrm{d}^D k_n\, V^{(n)}_{A_1,\ldots,A_n}(k_1,\ldots,k_n)\phi_{A_1}(k_1)\cdots\phi_{A_n}(k_n)$$

Momentum representation (2)

Propagator

$$K_{AB}(k)^{-1} = \frac{-\mathrm{i}\, M_{AB}}{k^2 + m_A^2}\, Q_A(k)$$

- M_{AB} mixing matrix for states of equal mass
- Q_A polynomial in momentum

Momentum representation (3)

Vertices

$$-\,\mathrm{i}V^{(n)}_{A_1,\ldots,A_n}(k_1,\ldots,k_n) = -\mathrm{i}\int \mathrm{d}t\,\mathrm{e}^{-g^{\{A_a\}}_{ij}(t)\,k_i\cdot k_j - c\sum\limits_{a=1}^{n} m_a^2} \times P_{A_1,\ldots,A_n}(k_1,\ldots,k_n;t)$$

- t moduli parameters
- $P_{\{A_a\}}$ polynomial in $\{k_i\}$
- $c > 0 \to$ damping in sum over states
- g_{ij} positive definite
- no singularity for $k_i \in \mathbb{C}$ (finite)
- $\lim\limits_{k^0\to\pm\mathrm{i}\infty} V^{(n)} = 0$
- $\lim\limits_{k^0\to\pm\infty} V^{(n)} = \infty$

Green functions

Truncated Green function = sum of Feynman diagrams of the form

$$\mathcal{F}(p_1, \ldots, p_n) \sim \int \mathrm{d}\, T \prod_s \mathrm{d}^D \ell_s \, \mathrm{e}^{-G_{rs}(T)\, \ell_r \cdot \ell_s - 2H_{ra}(T)\, \ell_r \cdot p_a - F_{ab}(T)\, p_a \cdot p_b} \times \prod_i \frac{1}{k_i^2 + m_i^2} \mathcal{P}(p_a, \ell_r; T)$$

T, moduli parameters, $\mathcal{P}$, polynomial in (p_a, ℓ_r)

- momenta:
  - external $\{p_a\}$
  - internal $\{k_i\}$
  - loop $\{\ell_s\}$

  k_i = linear combination of $\{p_a, \ell_s\}$
- G_{rs} positive definite
  - integrations over spatial loop momenta $\boldsymbol{\ell}_r$ converge
  - integrations over loop energies ℓ_r^0 diverge

Momentum integration

Prescription = generalized Wick rotation [1604.01783, Pius-Sen]

1. define Green function for Euclidean internal/external momenta
2. analytic continuation of external energies + integration contour s.t.
    - keep poles on the same side
    - keep ends at $\pm i\infty$

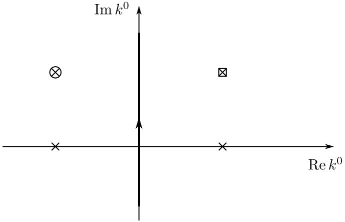
$\rightarrow$ analyticity for $\boldsymbol{p}_a \in \mathbb{R}$, p_a^0 in first quadrant $\operatorname{Im} p_a^0 > 0, \operatorname{Re} p_a^0 \geq 0$

Momentum integration

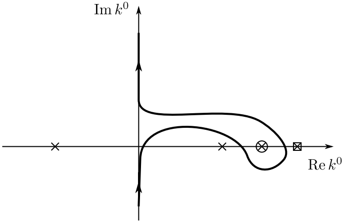
Prescription = generalized Wick rotation [1604.01783, Pius-Sen]

1. define Green function for Euclidean internal/external momenta
2. analytic continuation of external energies + integration contour s.t.
    - keep poles on the same side
    - keep ends at $\pm \mathrm{i}\infty$

$\rightarrow$ analyticity for $\boldsymbol{p}_a \in \mathbb{R}$, p_a^0 in first quadrant $\operatorname{Im} p_a^0 > 0, \operatorname{Re} p_a^0 \geq 0$



$\longrightarrow$



SFT in a nutshell

SFT = standard QFT s.t.:

- infinite number of fields (of all spins)
- infinite number of interactions
- non-local interactions $\propto \mathrm{e}^{-\# k^2}$
- reproduce worldsheet amplitudes (if well-defined)

Reviews: [1703.06410, de Lacroix-HE-Kashyap-Sen-Verma; 1905.06785, Erler; 2301.01686, HE]

SFT in a nutshell

SFT $=$ standard QFT s.t.:

- infinite number of fields (of all spins)
- infinite number of interactions
- non-local interactions $\propto \mathrm{e}^{-\# k^2}$
- reproduce worldsheet amplitudes (if well-defined)

Consequences of non-locality:

- cannot use position representation
- cannot use assumptions from local QFT (micro-causality...)

Reviews: [1703.06410, de Lacroix-HE-Kashyap-Sen-Verma; 1905.06785, Erler; 2301.01686, HE]

Consistency properties

- background independence [hep-th/9311009, Sen-Zwiebach; hep-th/9411047, Bergman-Zwiebach; 1711.08468, Sen]
- Cutkosky rules, unitarity [1604.01783, Pius-Sen; 1607.08244, Sen]
- spacetime and moduli space $\mathrm{i}\epsilon$-prescriptions [1610.00443, Sen]
- primitive analyticity, crossing symmetry [1810.07197, de Lacroix-HE-Sen]
- soft theorems [1702.03934, Sen, 1703.00024, Sen]
- locality? causality? CPT?

Note: also shows consistency of timelike Liouville theory [1905.12689, Bautista-Dabholkar-HE]

Off-shell tree-level string amplitudes

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

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

  - $\langle \cdots \rangle$ CFT correlation function
  - sum over topologically inequivalent spheres Σ_n with n punctures at $(\xi_1, \ldots, \xi_n)$
  - can fix 3 points $(\xi_{n-2}, \xi_{n-1}, \xi_n) = (0, 1, \infty)$
  - $\xi_\lambda \in \mathcal{M}_{0,n} \sim \mathbb{C}^{n-3}$ $(\lambda = 1, \ldots, n-3)$ moduli space

Off-shell tree-level string amplitudes

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

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

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

Off-shell tree-level string amplitudes

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

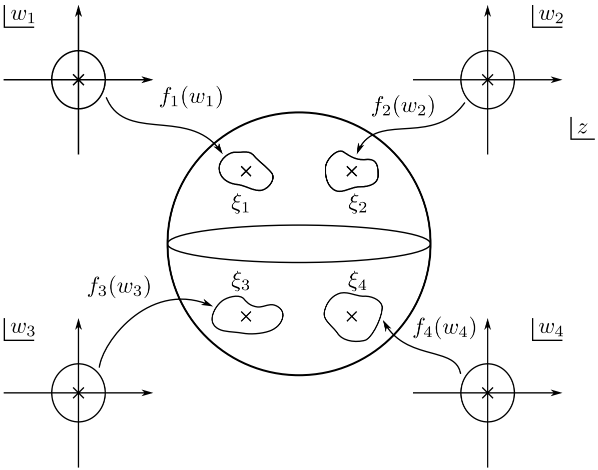
$$\mathcal{A}_{0,n}(A_1, \dots, A_n) = \int_{\mathcal{M}_{0,n}} \mathrm{d}^{n-3}\xi \, \left\langle \text{ghosts} \times \prod_i f_{n,i} \circ A_i(0) \right\rangle_{\Sigma_n}$$

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

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

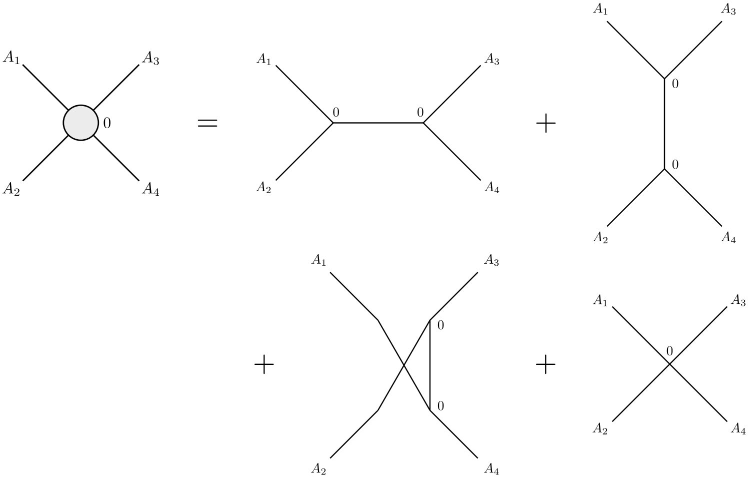
    if A_i is primary with weight (h_i, h_i)

Local coordinates



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

Amplitude and Feynman diagrams



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

Classical string vertices

- string vertex

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

Classical string vertices

- string vertex

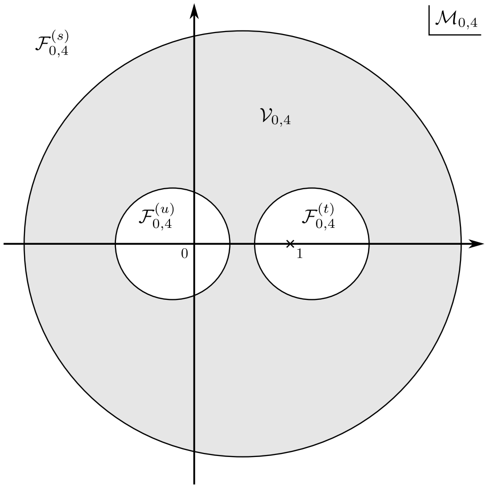
$$\mathcal{V}_{0,n}(A_1, \ldots, A_n) = \int_{\mathcal{V}_{0,n}} \mathrm{d}^{n-3}\xi \left\langle \text{ghosts} \times \prod_i f_{n,i} \circ A_i(0) \right\rangle_{\Sigma_n}$$

- defined such that

$$\mathcal{A}_{0,n}(A_1, \ldots, A_n) = \mathcal{F}_{0,n}(A_1, \ldots, A_n) + \mathcal{V}_{0,n}(A_1, \ldots, A_n)$$

  $\mathcal{F}_{0,n}$ contributions from Feynman diagrams (Riemann surfaces) containing:
  - propagators (long tubes)
  - surfaces in $\mathcal{V}_{0,n'}$ with $n' < n$
- $\mathcal{V}_{0,n} \subset \mathcal{M}_{0,n}$: vertex region $\subset$ moduli space
- constraints between all $\{f_{n,i}\}$ ("gluing compatibility")

Moduli space covering



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 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 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. Machine learning

Machine learning

Definition (Samuel)

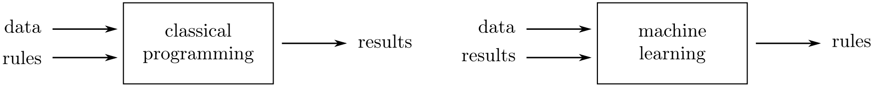
The field of study that gives computers the ability to learn without being explicitly programmed.

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. . .) $\Rightarrow$ new data representation
- agreement measured by some metric (distance, constraint. . .)
- tune structure parameters to improve approximation



Approaches to machine learning

Learning approaches (task: $x \longrightarrow y$)

- supervised: learn a map from a set or pairs $(x_{\text{train}}, y_{\text{train}})$, then predict y_{data} from x_{data}
- unsupervised: give x_{data} and let the machine find structure (i.e. appropriate y_{data})
- reinforcement: give x_{data}, let the machine choose output y_{data} following some rules, reward good and/or punish bad results, iterate

Applications

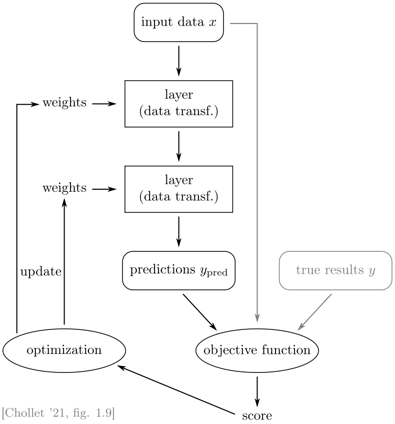
General idea = pattern recognition

- classification / clustering
- regression (prediction)
- transcription / translation
- structuring
- anomaly detection
- denoising
- synthesis and sampling
- density estimation
- symbolic regression

Applications in industry: computer vision, language processing, medical diagnosis, fraud detection, recommendation system, autonomous driving...

Neural network

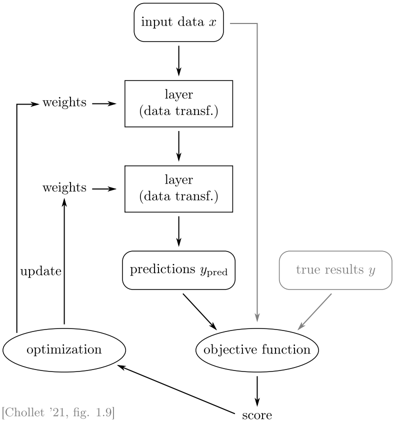
- neural network
  = sequence of layers
  implementing computations
- layer
  - output = different data representation
  - transformation parametrized by weights



[Chollet '21, fig. 1.9]

Neural network

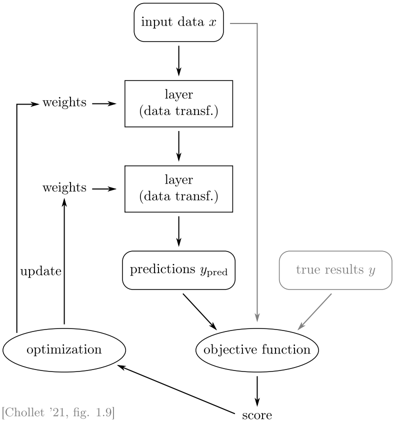
- neural network
  $=$ sequence of layers implementing computations
- layer
  - output $=$ different data representation
  - transformation parametrized by weights
- goal: find weights such that the network reproduces the target fonction $y = F(x)$
- comparison: objective function
- optimization by gradient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



[Chollet '21, fig. 1.9]

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. . .)



[Chollet '21, fig. 1.9]

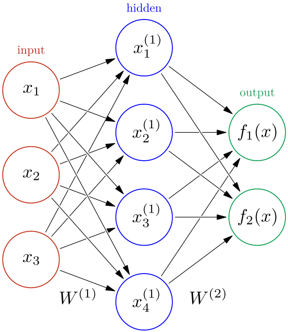
Why neural networks?

- 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

Fully connected neural network

$$x^{(0)}_{i_0} := x_{i_0}$$
$$x^{(1)}_{i_1} = g^{(1)}\left(W^{(1)}_{i_1 i_0} x^{(0)}_{i_0} + b^{(1)}_{i_1}\right)$$
$$f_{i_2}(x_{i_0}) := x^{(2)}_{i_2} = g^{(2)}\left(W^{(2)}_{i_2 i_1} x^{(1)}_{i_1} + b^{(2)}_{i_2}\right)$$

$$i_0 = 1, 2, 3;\ i_1 = 1, \ldots, 4;\ i_2 = 1, 2$$
$$K = 1;\ d_{\text{in}} = 3;\ d_{\text{out}} = 2;\ N^{(1)} = 4$$



- input $x^{(0)} := x \in \mathbb{R}^{d_{\text{in}}}$
- $K \geq 1$ hidden layers, $n \in \{1, \ldots, K\}$
  - layer n: $N^{(n)}$ neurons (units) $x^{(n)} \in \mathbb{R}^{N^{(n)}}$
  - learnable weights $W^{(n)} \in \mathbb{R}^{N^{(n)} \times N^{(n-1)}}$
  - learnable biases $b^{(n)} \in \mathbb{R}^{N^{(n)}}$ (not displayed)
  - fixed activation functions $g^{(n)}$ (element-wise)
- output $x^{(K+1)} := f(x) \in \mathbb{R}^{d_{\text{out}}}$

Training

Method:

1. fix architecture (number of layers, activation functions. . .)
2. learn weights $W^{(n)}$ from gradient descent

Training

Method:

1. fix architecture (number of layers, activation functions...)
2. learn weights $W^{(n)}$ from gradient descent

Gradient descent:

- loss function L: overall error to be decreased

$$L = \sum_{i=1}^{N_{\text{train}}} \text{distance}\left(y_i^{(\text{train})}, y_i^{(\text{pred})}\right) + \text{regularization}$$

  common choices: mean squared error, cross-entropy...
- optimizer and its parameters (learning rate, momentum...)
- ℓ_1 and ℓ_2 weight regularization (penalize high and redundant weights)
- training protocol: early stopping, learning rate decay...

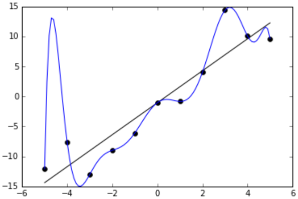
Training cycle

- hyperparameter tuning
  - adapt architecture and optimization for better results
  - search methods: trial-and-error, grid, random, Bayesian, genetic...

Training cycle

- hyperparameter tuning
  - adapt architecture and optimization for better results
  - search methods: trial-and-error, grid, random, Bayesian, genetic…
- main risk: overfitting (= cannot generalize to new data)

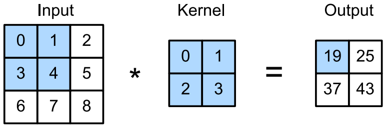
1. split data in training, validation and test sets
2. train several models on the training set
3. compare performances on validation set
4. evaluate performance of the best model on test set



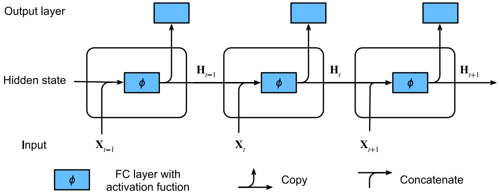
- consider n models in parallel (bagging) to get statistics

Neural network components (1)

- convolutional layer: move window over data, combining values with a kernel (to be learned)
  $\rightarrow$ translation covariance, locality, weight sharing

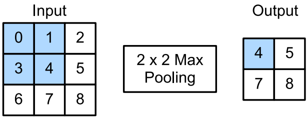


- recurrent layer (LSTM, GRU): keep memory of past information in a sequence
  $\rightarrow$ temporal processing

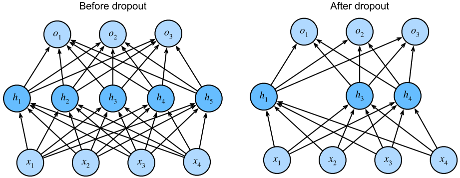


Neural network components (2)

- pooling layer: coarse-graining
  $\rightarrow$ reduce internal data size, translation/rotation/scale invariances



- dropout layer: deactivate neurons randomly with probability p
  $\rightarrow$ improve generalization, regularization



- batch normalization layer: normalize data, then scale and shift (learnable parameters)
  $\rightarrow$ keep stable internal data, regularization

Images: d2l.ai

ML workflow

"Naive" workflow

1. get raw data
2. write neural network with many layers
3. feed raw data to neural network
4. get nice results (or give up)

xkcd.com/1838

ML workflow

Real-world workflow

1. understand the problem
2. exploratory data analysis
   - feature engineering
   - feature selection
3. baseline model
   - full working pipeline
   - lower-bound on accuracy
4. validation strategy
5. machine learning model(s)
6. ensembling

Pragmatic ref.: `coursera.org/learn/competitive-data-science`

Outline: 4. Minimal area string 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]

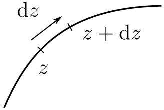
$$\phi(z) = \sum_{i=1}^{n} \left[\frac{-1}{(z-\xi_i)^2} + \frac{c_i}{z-\xi_i} \right]$$

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

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

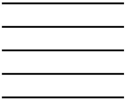
$$\mathrm{d}s^2 = |\phi(z)|^2 |\mathrm{d}z|^2, \qquad \mathrm{d}s^2|_{w_i} = \frac{|\mathrm{d}w_i|^2}{|w_i|^2}$$

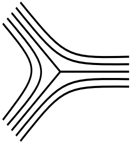
Critical trajectory

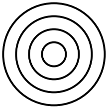


Definitions:

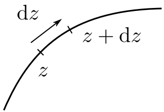
- $\{z_i(c_i, \xi_i)\}$ zeros of $\phi(z)$
- horizontal trajectory = path with $\varphi = \phi(z)\mathrm{d}z^2 > 0$

$\mathrm{d}z^2$ 

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

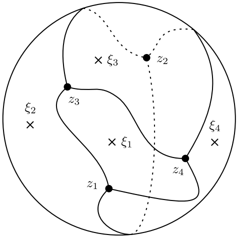
$-\dfrac{\mathrm{d}z^2}{z^2}$ 

Critical trajectory

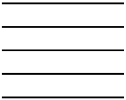


Definitions:

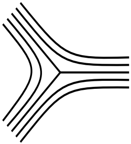
- $\{z_i(c_i, \xi_i)\}$ zeros of $\phi(z)$
- horizontal trajectory = path with $\varphi = \phi(z)\mathrm{d}z^2 > 0$
- critical trajectory = horizontal trajectory with ends at $\phi(z) = 0$
- critical graph = {critical trajectories}



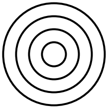
$\mathrm{d}z^2$



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



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

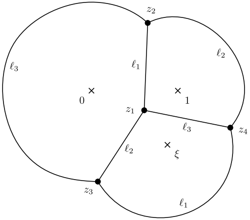


Strebel quadratic differential

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)



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

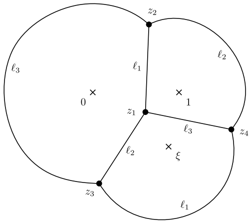
Strebel quadratic differential

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

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

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

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

- 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

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

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

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

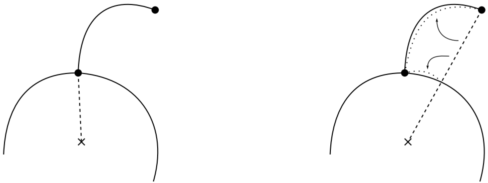
- remaining unknown: mapping radii $\rho_i \in \mathbb{R}$

Mapping radii

- mapping radius for ξ_i (conformal invariant)

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

- 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: 5. Machine learning for string field theory

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

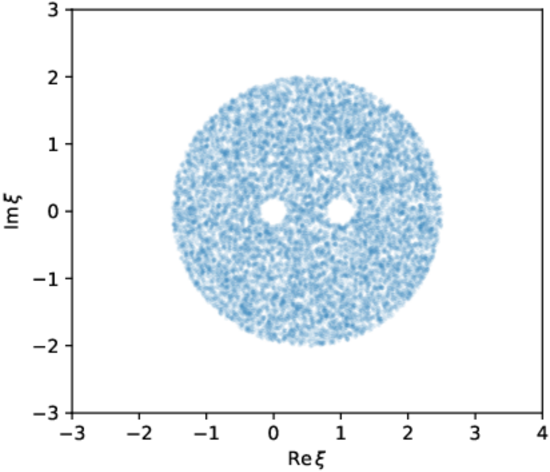
- idea : $c_\lambda(\xi_\lambda) =$ complex neural network $C_\lambda(\xi_\lambda; \boldsymbol{W}, \boldsymbol{b})$
- $\boldsymbol{W}$ weights (complex matrices), $\boldsymbol{b}$ biases (complex vectors)
- unsupervised training with loss

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

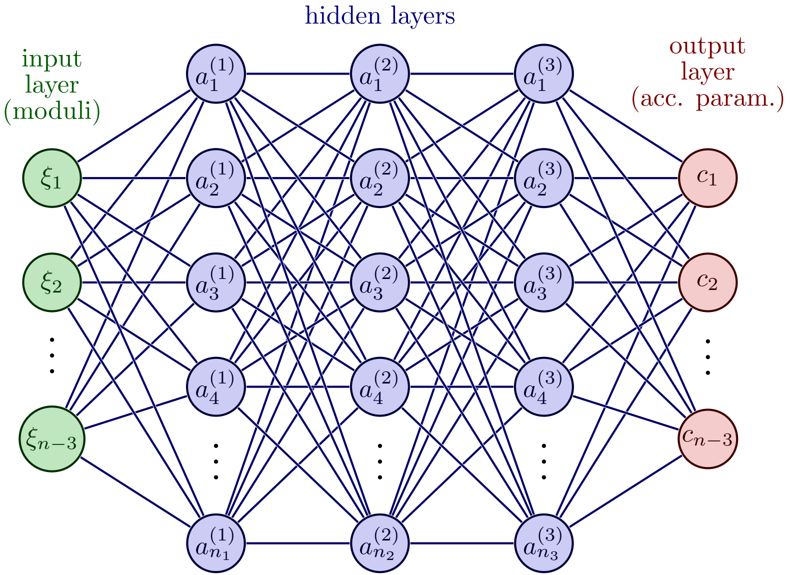
$\rightarrow$ minimize with gradient descent

- 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



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

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

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

Results: 4-punctured sphere (1)

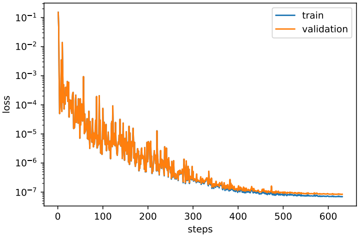
- neural network (Jax)
  - fully connected, 3 layers $(512, 128, 1028)$, $\mathbb{C}\mathrm{ReLU}$ activation

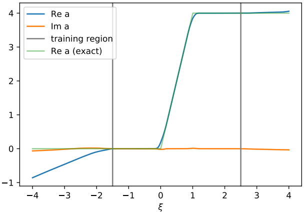
$$\mathbb{C}\mathrm{ReLU}(z) := \mathrm{ReLU}(\operatorname{Re} z) + \mathrm{i}\,\mathrm{ReLU}(\operatorname{Im} z)$$

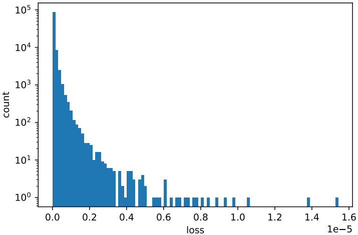
  - training: 10^5 points, Adam, ℓ_2 regularization, weight decay, early stopping (~ 1000 epochs)
- loss statistics (exact solution $\sim 10^{-12}$)
  - mean: $8.9 \cdot 10^{-8}$
  - median: $3.8 \cdot 10^{-8}$
  - min: $1.3 \cdot 10^{-11}$
  - max: $1.5 \cdot 10^{-5}$

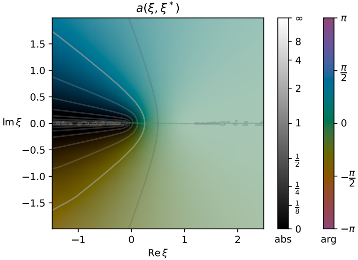
  note: already good performance with 10^3 points, 100 epochs (e.g. mean loss $= 2.7 \cdot 10^{-5}$)
- mean error compared to Moeller's fit: $5.5 \cdot 10^{-3}$

Results: 4-punctured sphere (2)





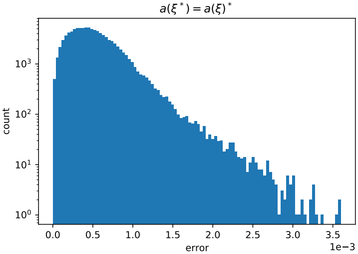


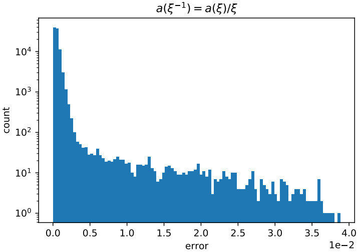


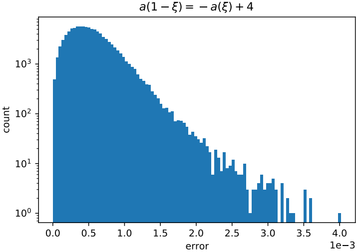
$$a(1/2) = 1.9995 + 0.0001\mathrm{i}, \qquad a(Q) = 1.9997 + 1.1548\mathrm{i}$$

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

Results: symmetries



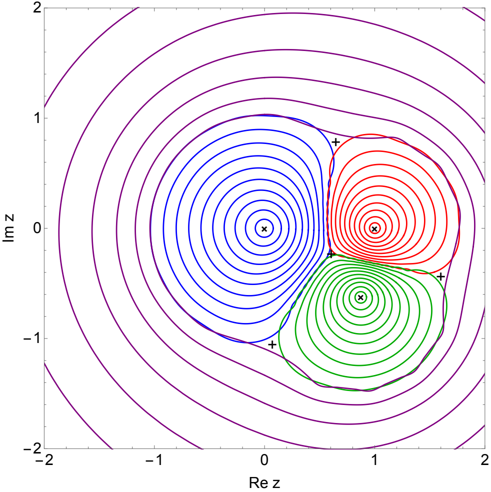




complex conjugation and permutation of fixed punctures

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

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. $\theta(\xi) \in [0.2, 0.8]$

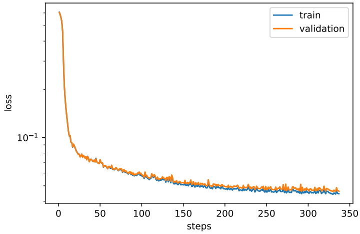
Learning the vertex region

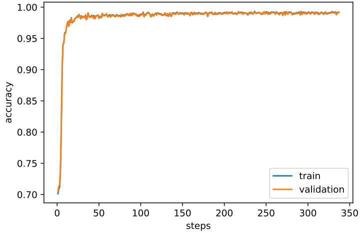
- idea: $\Theta(\xi)$ = neural network $\theta(\xi)$
  - $\theta(\xi)$ becomes probability distribution
  - useful for Monte Carlo integration
  - easily find boundary, e.g. $\theta(\xi) \in [0.2, 0.8]$
- supervised classification, binary cross-entropy loss

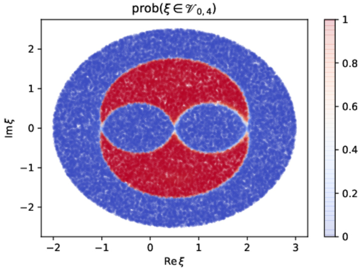
$$\mathcal{L}(\xi) = -\Theta(\xi) \ln \theta(\xi) - \big(1 - \Theta(\xi)\big) \ln \big(1 - \theta(\xi)\big)$$

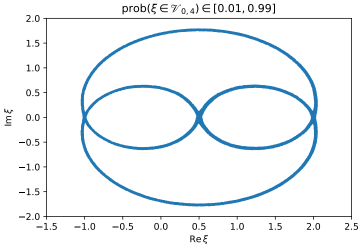
- neural network (Jax)
  - fully connected, 4 layers $(512, 32, 8, 8)$, ELU activation
  - training: 10^5 points, Adam, ℓ_2 regularization, weight decay, early stopping ($\sim$ 800 epochs)

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)

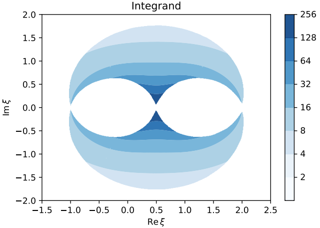
$$V(t) = -t^2 + \frac{v_3}{3!}\, t^3 - \frac{v_4}{4!}\, t^4 + \cdots$$

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

- 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: $\sim 10^{-3}$
  $\to$ expect sufficiently precise for determining vacuum
- full pipeline: $\sim$ 4 hours

Outline: 6. 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)