Bootstrapping String-Like Models via Entanglement Minimization and Machine Learning

Work with: Debapriyo Chowdhury, Arnab Priya Saha, Aninda Sinha

Non-perturbative methods in QFT,

Kyushu University IAS-iTHEMS

Faizan Bhat, Centre for High-Energy Physics,
Indian Institute of Science, Bangalore, India

Based on: arXiv: 2409.18259 [hep-th] and 2401.05733 [hep-th]

Plan for the talk

- **Dual Resonance Models**
- **Field Theory Representations of Dual Resonance/String Models**
- **String Bootstrap: Set up**
- **Entanglement in scattering**
- **String Bootstrap: Results**
- **Bootstrap using Machine Learning**
- **Summary**

Dual Resonance Models

- Data from p-p scattering and heavy-ion collision experiments shows families of higher-spin hadronic resonances that seem to lie on **linear Regge trajectories**, $J \sim \alpha_0 + \alpha' M^2$.

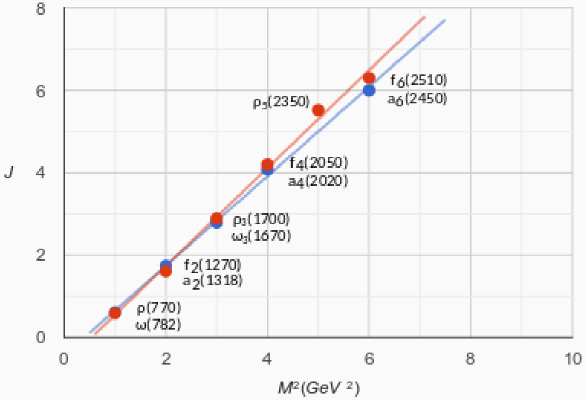


Figure 1: Chew-Frautschi plot for ρ and ω mesons

Dual Resonance Models

- **Puzzle**: The tree-level exchange of a spin J particle of mass m_J leads to the following behaviour in the t-channel

$$\mathcal{M}(s,t) = -\sum_{J}^{J_{max}} g_J^2 \frac{(-s)^J}{t - m_J^2}, \quad \Longrightarrow \mathcal{M}(s,t) \sim s^{J_{max}}$$

- **Veneziano Amplitude**:

$$\boxed{\mathcal{M}^{(Ven)}(s,t) = \frac{\Gamma(-\alpha' s)\Gamma(-\alpha' t)}{\Gamma(1 - \alpha' s - \alpha' t)}}$$

- **Duality**: For $Re(s) < 1$,

$$\boxed{\mathcal{M}^{(Ven)}(s,t) = \frac{1}{st} + \sum_{n=1}^{\infty} \frac{(s+1)(s+2)...(s+n-1)}{n!} \frac{1}{t-n}}$$

- **Regge Trajectories**: Residue at $t = n$ goes as s^{n-1}.
  So, highest-spin exchanged particle with mass n has spin $J = n - 1$.
- **Regge Behavior**: For fixed t and large s, $\mathcal{M}^{(Ven)}(s,t) \sim s^{-1+t}$.

Field Theory Representations

Field Theory Representations

- A function that satisfies $\mathcal{M}(s,t) = \mathcal{M}(t,s)$ and has the Regge behaviour

$$\lim_{|s|\to\infty,\text{fixed } t} |\mathcal{M}(s,t)| = 0$$

can be expressed via a crossing symmetric dispersion relation

$$\boxed{\mathcal{M}(s,t) = \frac{1}{\pi}\int_{s_0}^{\infty} \mathrm{d}\sigma \left[\frac{1}{\sigma - s} + \frac{1}{\sigma - t} - \frac{1}{\sigma + \lambda}\right] \mathcal{A}^{(s)}\left(\sigma, \frac{(s+\lambda)(t+\lambda)}{\sigma + \lambda} - \lambda\right)}$$

- σ_0 is the location of the first singularity along the s-channel and $\mathcal{A}^{(s)}(s,t)$ is the corresponding discontinuity

$$\mathcal{A}^{(s)}(s,t) = \lim_{\epsilon\to 0}\left(\mathcal{M}(s+i\epsilon,t) - \mathcal{M}(s-i\epsilon,t)\right) = \frac{1}{2i}\text{Im}\mathcal{M}(s,t)$$

- Dual resonance models belong to this class.

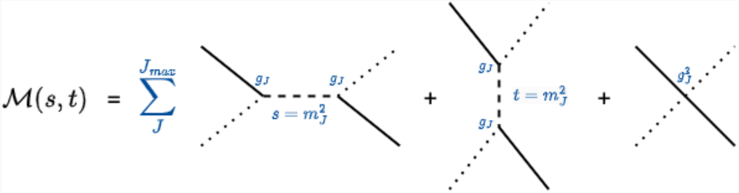


Figure 2: Field Theory Representation: Poles in all channels + contact terms

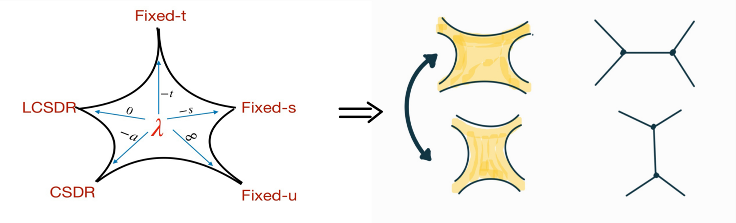
- Open superstring amplitude: Since $\mathcal{M}^{OS}(s,t) \sim s^{-1+t}$, we can apply the CSDR. $\mathcal{A}^{(s)}(s,t) = -\pi\delta(s-n) \operatorname*{Res}_{s=n} \mathcal{M}(s,t)$.

$$\boxed{\frac{\Gamma(-s)\,\Gamma(-t)}{\Gamma(1-s-t)} = \frac{1}{st} + \sum_{n=1}^{\infty} \frac{1}{n!}\left(\frac{1}{s-n} + \frac{1}{t-n} + \frac{1}{\lambda+n}\right) \times \left(1-\lambda+\frac{(s+\lambda)(t+\lambda)}{\lambda+n}\right)_{n-1}}$$

- This is called the "field-theory representation" [A.Sinha, A.P.Saha].

Field Theory Representations

- **Salient features**:
  - Crossing symmetric at each mass level.
  - It converges everywhere except at the poles for $Re(\lambda) > -1$.
  - Allows truncation without losing unitarity.
  - Bootstrap: Allows for a larger domain in s, t to impose constraints.
- **A Unified Dispersion Relation**



Field Theory Representations

- Lorentz invariance implies a partial wave expansion of the residues:

$$\text{Res}_{s=n}\mathcal{M}(s,t) = -\pi \sum_{\ell} c_\ell^{(n)} \mathcal{C}_\ell^{\frac{D-3}{2}}\left(z = 1 + \frac{2t}{s}\right)$$

where $\mathcal{C}_\ell^{\frac{D-3}{2}}(z)$ are the Gegenbauer polynomials in D dimensions.

- The general formula looks like

$$\boxed{\mathcal{M}(s,t) = \sum_{n=1}^{\infty}\sum_{\ell=0}^{\ell_{max}} \left[\frac{1}{s-n} + \frac{1}{t-n} + \frac{1}{\lambda+n}\right] c_\ell^{(n)} \mathcal{C}_\ell^{\left(\frac{D-3}{2}\right)} \left[1 + \frac{2}{n}\left(\frac{(s+\lambda)(t+\lambda)}{\lambda+n} - \lambda\right)\right]}$$

- Tree-Level Unitarity: $c_\ell^{(n)} \geq 0$
- λ- Independence/ Null Constraints: $\partial_\lambda^k \mathcal{M}_\lambda(s,t) = 0, \text{Re}\lambda > -1.$ These are exactly the same as crossing symmetry constraints from the fixed-t dispersion relation.

String Bootstrap
Set up

String Bootstrap: Set up

- Bootstrap Approach: Bound the space of consistent scattering amplitudes by imposing physical constraints such as unitarity, crossing symmetry, Lorentz invariance, analyticity, etc.
- **Q: In the space of (bootstrap) consistent scattering amplitudes that satisfy duality, is the open superstring amplitude special?**
- Wilson Coefficients: Expansion around $s+t=0$ and $s\ t=0$

$$\mathcal{M}_{low}(s,t) = W_{00} + W_{10}(s+t) + W_{01}st + \cdots$$

- Bootstrap Constraints: In $D=10$,
  - Crossing Symmetry: $\mathcal{M}(s,t)=\mathcal{M}(t,s)$
  - Analyticity: Only simple poles at $s=n, \forall n \in \mathbb{Z}_{\geq 0}$.
  - Residues at Poles: Residues are polynomials of order $\ell_{max}=n-1$.
  - Unitarity: At tree-level, $c_\ell^{(n)} \geq 0$
  - λ-Independence/Null constraints: $\partial_\lambda^k \mathcal{M}_\lambda(s,t)=0$, for $k \in \mathbb{Z}_{\geq 1}, \lambda \geq -1, (s,t) \in \mathcal{D}_\lambda$

String Bootstrap: Set Up

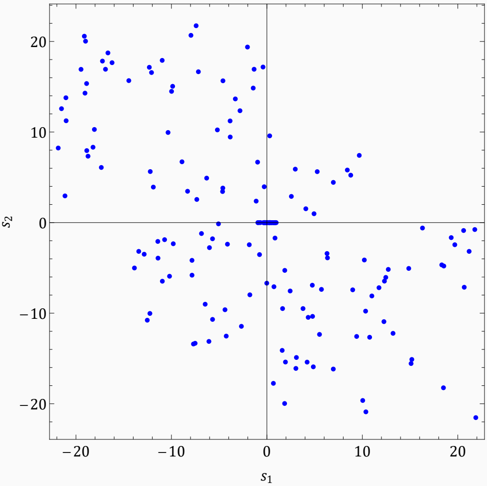


Figure 3: Example of $\mathcal{D}_\lambda$ we use for bootstrap

Entanglement in scattering

Entanglement in scattering

- **Linearized Entropy**: For a 2-2 scattering process is defined as

$$\mathcal{E}[\Omega] = 1 - \mathrm{Tr}_A[\hat{\rho}_A^2], \quad \hat{\rho}_A = \mathrm{Tr}_B \hat{\rho}_{AB}, \quad \hat{\rho}_{AB} = \hat{S}|\Omega\rangle\langle\Omega|\hat{S}^\dagger$$

- $|\Omega\rangle = \sum_{AB} \Omega_{AB} |k_1, a\rangle_A \otimes |k_2, b\rangle_B$ is a general state in $\mathcal{H}_A \otimes \mathcal{H}_B$.
- Density matrix is computed via

$$\hat{\rho}_A = \frac{1}{N} \sum_c \int_u \langle u, c|_B \hat{S} |\Omega\rangle\langle\Omega| \hat{S}^\dagger |u, c\rangle_B.$$

- The initial state is considered to be unentangled.

$$\Omega^{prod} = |k_1, a\rangle_A \otimes |k_2, b\rangle_B, \quad \Longrightarrow \mathcal{E}[\Omega^{prod}] = 0$$

- **Entanglement Power**: Defined as $\Delta\mathcal{E}[\Omega] = \mathcal{E}^f[\Omega] - \mathcal{E}^i[\Omega]$. $\Delta\mathcal{E}[\Omega]$ can be computed perturbatively in some small coupling g.
- For an unentangled initial state, up to leading order in g^2 [R. Aoude, G. Elor, G. N. Remmen, O. Sumensari]

$$\Delta\mathcal{E}[\Omega^{\mathrm{prod}}] = 4\mathcal{N}\, \mathrm{Im}\, \mathcal{M}_{\alpha\beta}^{\alpha\beta}(k_1 k_2 \to k_1 k_2).$$

- Only elastic, forward ($t = 0$) scattering contributes.

Entanglement in scattering

- In open superstring theory, the tree-level, color-ordered, four-gluon scattering amplitude is given by

$$\mathcal{M}_4(1234) = g^2 \mathcal{F}^4 \frac{\Gamma(-s)\,\Gamma(-t)}{\Gamma(1-s-t)},$$

$$\mathcal{F}^4 = F_{1,\mu\nu} F_2^{\mu\nu} F_{3,\alpha\beta} F_4^{\alpha\beta} + F_{1,\mu\nu} F_3^{\mu\nu} F_{4,\alpha\beta} F_2^{\alpha\beta} + F_{1,\mu\nu} F_4^{\mu\nu} F_{2,\alpha\beta} F_3^{\alpha\beta} - 4\left\{ F_{1\mu\nu} F_2^{\nu\alpha} F_{3,\alpha\beta} F_4^{\beta\mu} + F_{1\mu\nu} F_3^{\nu\alpha} F_{4,\alpha\beta} F_2^{\beta\mu} + F_{1\mu\nu} F_4^{\nu\alpha} F_{2,\alpha\beta} F_3^{\beta\mu} \right\},$$

- $F_{i,\mu\nu} = p_{i,\mu}\varepsilon_{i,\nu} - p_{i,\nu}\varepsilon_{i,\mu}$ for the i-th gluon.
- In the forward limit, $\lim_{t\to 0} \mathcal{F}^4 = -2s^2 \varepsilon_1 \cdot \varepsilon_4 \, \varepsilon_2 \cdot \varepsilon_3$
- So we can define

$$\Delta\mathcal{E}_{massive} = 8\pi N s^2 \varepsilon_1 \cdot \varepsilon_4 \, \varepsilon_2 \cdot \varepsilon_3 \sum_{n=1}^{\infty} \mathrm{Res}_{s=n} \mathcal{M}(s,0)\delta(s-n).$$

- We are interested in the first finite moment of entangling power

$$\boxed{EPM = \int_1^{\infty} ds \frac{\Delta\mathcal{E}_{massive}}{s^2} = \sum_{n=1}^{\infty} \sum_{\ell=0}^{\ell_{max}} \frac{1}{n} c_\ell^{(n)} \mathcal{C}_\ell^{\frac{D-3}{2}}(1) = -W_{0,0}}$$

String Bootstrap Results

- **Maximize $W_{0,0}$/Minimize Entanglement**

$$N_{max} = 30, \epsilon = 10^{-9}, k_{max} = 6, \lambda = 14.6.$$

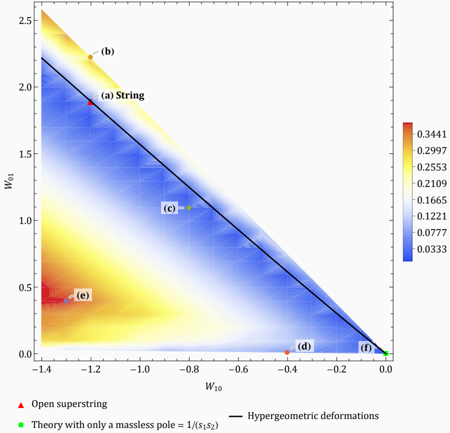


Figure 4: Heat map for duality violation for entanglement minimizing amplitudes in the allowed $W_{1,0} - W_{0,1}$ plane

String Bootstrap: Results

- **Maximize $W_{0,0}$/Minimize Entanglement**

$$(N_{max} = 30,\ \epsilon = 10^{-9},\ k_{max} = 6,\ \lambda = 14.6)$$

n	Exact	Bootstrap
1	1	1.000
2	0.0714	0.0717
3	0.0119	0.0121
4	0.00289	0.00297
5	0.000867	0.000898
6	0.000300	0.000311
7	0.000114	0.000119
8	0.0000469	0.0000486
9	0.0000204	0.0000211
10	9.26×10^{-6}	9.59×10^{-6}
11	4.37×10^{-6}	4.52×10^{-6}
12	2.12×10^{-6}	2.19×10^{-6}
13	1.06×10^{-6}	1.09×10^{-6}
14	5.41×10^{-7}	5.57×10^{-7}
15	2.81×10^{-7}	2.90×10^{-7}

(a) Leading Regge trajectory

n	Exact	Bootstrap
3	0	0
4	0.00108	0.000367
5	0.000636	0
6	0.000325	0.000123
7	0.000163	0.000118
8	0.0000831	0.0000752
9	0.0000430	0.0000427
10	0.0000227	0.0000235
11	0.0000122	0.0000130
12	6.64×10^{-6}	7.20×10^{-6}
13	3.67×10^{-6}	4.01×10^{-6}
14	2.05×10^{-6}	2.25×10^{-6}
15	1.16×10^{-6}	1.27×10^{-6}

(b) Subleading Regge trajectory

String Bootstrap: Results

- **Maximize $\mathbf{W_{0,0}}$/Minimize Entanglement**

$$(N_{max} = 30,\ \epsilon = 10^{-9},\ k_{max} = 6,\ \lambda = 14.6)$$

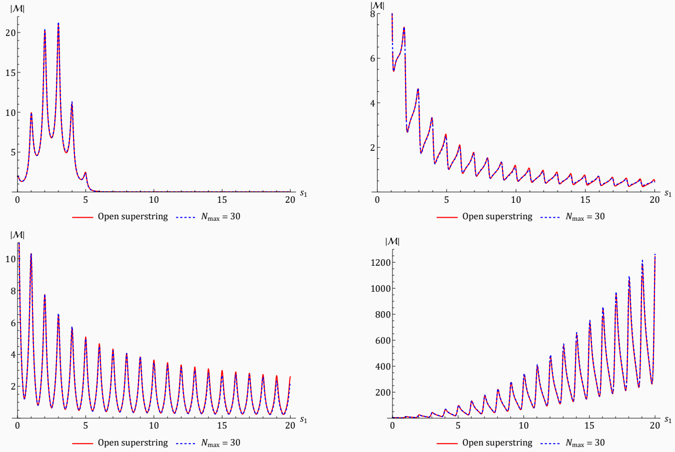


Figure 5: We plot $\left|\mathcal{M}\left(s + \frac{i}{10}, t\right)\right|$ versus s for $t = \{-5.1, -0.1, 0.5, 3.2\}$.

Bootstrap using Neural Networks

Bootstrap using PINNs

- **Why use PINNs for bootstrap?**
- The bootstrap problems we discussed till now and in fact, all bootstrap problems are either Linear optimization problems, or Semi-definite optimization problems.
- These can be handled well via traditional methods like SDPB.
- Caveat: We used $\epsilon \sim 10^{-9}$ while imposing the constraint

$$-\epsilon \leq \partial_\lambda^k \mathcal{M}_\lambda \left(s, t, c_\ell^{(n)}\right) \leq \epsilon, \text{ for } (s,t) \in \mathcal{D}_\lambda \text{ and } 1 \leq k \leq k_{max}$$

  However, truncated to some N_{max}, it is not guaranteed that these constraints will be satisfied to some $\epsilon << 1$.
- It is more appropriate to impose ratio constraints

$$\left|1 - \frac{\mathcal{M}_{\lambda_1}(s,t)}{\mathcal{M}_{\lambda_2}(s,t)}\right| \leq \epsilon, \qquad \left|\frac{\partial_\lambda^k \mathcal{M}_\lambda(s,t)}{\mathcal{M}_\lambda(s,t)}\right| \leq \epsilon$$

- These are non-linear in the parameters $c_\ell^{(n)}$. Traditional methods like SDPB are not useful. This is why we use PINNs.

Bootstrap using PINNs

- Neural networks: Maps with several tunable parameters. In our case,

$$\mathrm{NN}\left(\ell, n, \theta_j\right) \equiv c_\ell^{(n)}$$

- Neuron Input-Output: $y_j^M = \sigma\left(\sum_{k=1}^{k_{max}} w_{j,k}^M y_k^{M-1} + b_j^M\right)$

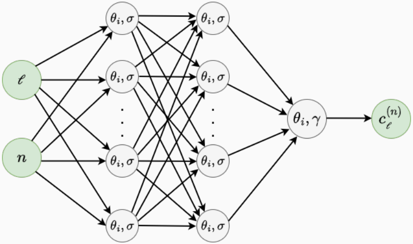


Figure 6: Architecture: Input layer with 2 neurons for (ℓ, n), 2 hidden layers with 64 neurons, output layer wih 1 neuron for $c_\ell^{(n)}$. Every neuron has the ReLU activation function $\sigma(x) = \max(0, x)$. Final layer has the SoftPlus activation function $\gamma(x) = \log(1 + e^x)$ for positive $c_\ell^{(n)}$s.
Total Parameters $= [2(64) + 64] + [64(64) + 64] + [64(1) + 1] = 4417$.

Bootstrap using PINNs

- Typically, neural networks learn the "best fit curve" to a given data by minimizing a loss function that measures the error in the fit.
- PINNs are a sub-class of neural networks where the loss function includes physical constraint terms.
- True solution to the constraint equations is learned as the PINN updates its parameters every epoch) via the gradient descent method to approach the minimum of the loss function

$$\theta_{n+1} = \theta_n - \eta \nabla_{\theta_n} \mathcal{L}_{\theta_n}, \quad \eta = \text{learning rate}$$

- We define the following loss function

$$\boxed{\begin{aligned}\mathcal{L}\left(\theta_j^M\right) = -W_{0,0} + \beta_1 \Big(W_{10} - (-\zeta(3))\Big)^2 + \beta_2 \left(W_{01} - \frac{7\zeta(4)}{4}\right)^2 \\ + \beta_3 \frac{1}{N} \sum_{s,t \in \mathcal{D}_\lambda} \tilde{\mathcal{L}}(s,t), \qquad W_{0,0} = -EPM\end{aligned}}$$

- Hyperparameters β_i set the tolerance for constraint violation. Bigger β_i means smaller tolerance $\implies \min(EPM) = \min\left(\mathcal{L}\left(\theta_j^M\right)\right)$

Bootstrap using PINNs

- We implement PINN using the Python library PyTorch.
- **Case 1:** When $\boxed{\tilde{\mathcal{L}}(s,t) = \left(1 - \frac{\mathcal{M}_{\lambda_1}(s,t)}{\mathcal{M}_{\lambda_2}(s,t)}\right)^2}$,
  - For $N_{max} = 13, \lambda = 5.6, \lambda_1 = 5.1, \lambda_2 = 6.1$,
  - We pick $\beta_i = 10^4, N_{epochs} = 2 \times 10^5$ and $\mathcal{D}_\lambda = \{(s,t) \,|\, -5.5 \leq s \leq 5.5\,, -0.2 \leq t \leq 0.2\,, \Delta_s = 1, \Delta_t = 0.4\}$
  - **Solution**:

$$max(W_{00}) = -1.506\,, \quad W_{10} - (-\zeta(3)) = 2.8 \times 10^{-6}$$

$$W_{01} - \frac{7\zeta(4)}{4} = -5.2 \times 10^{-5}\,, \quad \tilde{\mathcal{L}}_{mean} = 8.5 \times 10^{-4}.$$

  - **Leading Regge Trajectory**:

	$c_0^{(1)}$	$c_1^{(2)}$	$c_2^{(3)}$	$c_3^{(4)}$	$c_4^{(5)}$	$c_5^{(6)}$
Open String	1	0.0714	0.0119	0.00289	0.000867	0.000300
PINN	0.999	0.0715	0.0121	0.00300	0.000922	0.000332

Bootstrap using PINNs

- **Case 2:** When $\boxed{\tilde{\mathcal{L}}(s,t) = \sum_{k=1}^{k_{max}} \left(\frac{1}{\mathcal{M}_\lambda(s,t)} \frac{\partial^k \mathcal{M}_\lambda(s,t)}{\partial \lambda^k} \right)^2}$,
  - For $k_{max} = 1$, $N_{max} = 20$, $\lambda = 14.6$, $\lambda_1 = 5.1$, $\lambda_2 = 6.1$,
  - We pick $\beta_1 = \beta_2 = 10^6$,$\beta_3 = 10^8$, $N_{epochs} = 4 \times 10^5$ and $\mathcal{D}_\lambda = \{(s,t)\,|\,0.4 \le s \le 10.4\,, t = 10.1\,, \Delta_s = 1\}$
  - **Solution**:

$$max(W_{00}) = -1.426\,, \quad W_{10} - (-\zeta(3)) = -6.7 \times 10^{-7}$$

$$W_{01} - \frac{7\zeta(4)}{4} = -4.8 \times 10^{-7}\,, \quad \tilde{\mathcal{L}}_{mean} = 4.2 \times 10^{-5}.$$

  - **Leading Regge Trajectory**:

	$c_0^{(1)}$	$c_1^{(2)}$	$c_2^{(3)}$	$c_3^{(4)}$	$c_4^{(5)}$	$c_5^{(6)}$
Open String	1	0.0714	0.0119	0.00289	0.000867	0.000300
PINN	0.991	0.0574	0.0057	0.00131	0.000473	0.000277

- For open string, at $(s,t) = (10.4, 10.1)$, $\mathcal{M}(s,t) \approx 1.34 \times 10^5$ and $\partial_\lambda \mathcal{M}_\lambda(s,t) \approx -2.51$. $\Longrightarrow$ Only PINN method can work!.

Bootstrap using PINNS

- Regge Pole from PINNs

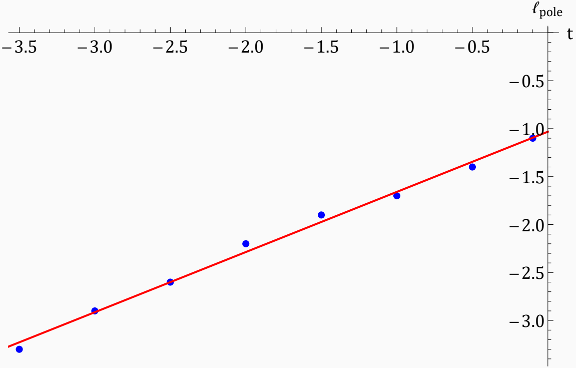


Figure 7: The blue dots denote the location of the Regge pole ℓ_{pole} of $c(\ell, t)$ as a function t. We find that these blue dots lie on the red straight line obtained by making a fit with our data. The intercept ≈ -1 matches the open superstring value.

Summary

Summary

- We present a new way to set up the numerical S-Matrix bootstrap using a parametric crossing symmetric dispersion relation.
- The dispersion relation can be continuously deformed to the fixed-s, fixed-t and CSDRs suggesting an underlying "worldsheet" picture.
- We minimize the first finite moment of the entangling power (EPM) and find that the optimal solution is an excellent approximation to the open superstring amplitude.
- We initiate the use of PINNs for the bootstrap to perform non-linear, constrained optimization, and the complex ℓ Regge pole analysis.
- We also study closed string-like amplitudes and find Dual resonance models there also minimize the first finite moment of the entangling power (EPM).

THANK YOU

- Parametric series representation for the hypergeometric deformed amplitudes

$$\frac{1}{s_1 s_2} - \frac{\Gamma(1-s_1)\Gamma(1-s_2)}{(r+1)\Gamma(-s_1-s_2+2)} \, {}_3F_2\left(r+1, 1-s_1, 1-s_2; r+2, -s_1-s_2+2; 1\right)$$

- Parametric representation for Closed-String amplitude

$$\frac{\Gamma(-s_1)\Gamma(-s_2)\Gamma(-s_3)}{\Gamma(1+s_1)\Gamma(1+s_2)\Gamma(1+s_3)} = -\frac{1}{s_1 s_2 s_3} +$$
$$\sum_{n=1}^{\infty} \frac{1}{(n!)^2} \left[\frac{1}{s_1-n} + \frac{1}{s_2-n} + \frac{1}{s_3-n} + \frac{1}{\lambda+n}\right] \times$$
$$\left(1 - \frac{n}{2} + \frac{n-2\lambda}{2}\sqrt{1 - \frac{4(s_1+\lambda)(s_2+\lambda)(s_3+\lambda)}{(n+\lambda)(n-2\lambda)^2}}\right)_{n-1}$$

Extra

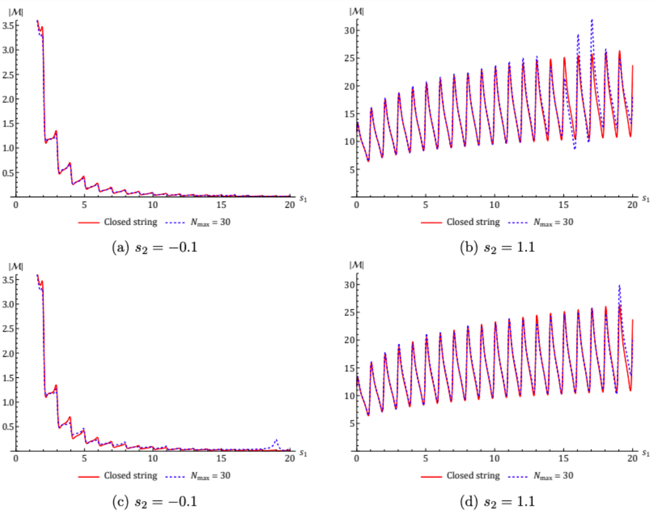


Figure 8: We plot $\left|\mathcal{M}\left(s_1 + \frac{i}{10}, s_2\right)\right|$ versus s_1 for the optimal closed-string like amplitude