# Bootstrability in line-defect CFTs with improved truncation methods

V. Niarchos,[1,*] C. Papageorgakis,[2,†] P. Richmond,[2,‡] A. G. Stapleton,[2,§] and M. Woolley[2,‖]

[1]*ITCP and CCTP, Department of Physics, University of Crete, 71003 Heraklion, Greece*

[2]*Centre for Theoretical Physics, Department of Physics and Astronomy, Queen Mary University of London, London E1 4NS, United Kingdom*

(Received 16 July 2023; accepted 2 November 2023; published 30 November 2023)

We study the conformal bootstrap of 1D CFTs on the straight Maldacena–Wilson line in 4D $\mathcal{N}=4$ super-Yang–Mills theory. We introduce an improved truncation scheme with an "OPE tail" approximation and use it to reproduce the "bootstrability" results of Cavaglià *et al.* for the OPE-coefficients squared of the first three unprotected operators. For example, for the first OPE-coefficient squared at 't Hooft coupling $(4\pi)^2$, linear-functional methods with two sum rules from integrated correlators give the rigorous result $0.294014873 \pm 4.88 \times 10^{-8}$, whereas our methods give with machine-precision computations $0.294014228 \pm 6.77 \times 10^{-7}$. For our numerical searches, we benchmark the reinforcement learning soft actor-critic algorithm against an interior point method algorithm (IPOPT) and comment on the merits of each algorithm.

DOI: 10.1103/PhysRevD.108.105027

## I. INTRODUCTION AND SUMMARY

In two recent papers, the authors of Refs. [1,2] initiated a nonperturbative study of 1D defect conformal field theories (CFTs) in the planar 't Hooft limit, combining methods from integrability and the numerical conformal bootstrap program. The analysis of [1,2] focused on the 1D defect CFT of the $\frac{1}{2}$-BPS infinite Maldacena–Wilson line in 4D $\mathcal{N}=4$ super-Yang Mills (SYM) theory. It produced high-precision numerics for three different (non-supersymmetric) three-point functions involving two protected and one unprotected operator. To obtain these results, the usual constraints of crossing symmetry were combined with information about spectral data from integrability[1] and two sum rules arising from integrated correlation functions [5,6].

The combination of powerful techniques from integrability and the conformal bootstrap, dubbed *bootstrability* in [1], aims to blend two methods that have played a leading role in nonperturbative studies of quantum field theories. On the one hand, integrability has proven very successful in the analytic computation of scaling dimensions in the planar limit of gauge theories but less efficient in computations of correlation functions. On the other hand, the conformal bootstrap[2] has shown great promise in yielding rigorous results for generic CFT data (including both scaling dimensions and correlation functions) but has difficulty in navigating toward arbitrary theories of interest. The input of integrability guides a conformal bootstrap search toward a desired solution.

In this work, we continue the bootstrability study of the 1D CFT on the straight, $\frac{1}{2}$-BPS Maldacena–Wilson line, by employing a different methodology on the conformal bootstrap side compared to Refs. [1,2]. Instead of the commonly used *linear functional* method [12], which produces rigorous inequalities for CFT data by making use of the positivity constraints from unitarity, we introduce an *improved truncation scheme* to directly solve approximate crossing equations and sum rules.

Truncation methods within the conformal bootstrap program are not new, having previously yielded outcomes with varying degrees of success. Earlier work includes [13–20], while more recently [21–24] have implemented truncation methods in the context of the four- and five-point bootstrap, respectively. In particular, Refs. [21,22] highlighted the significance of selecting an appropriate truncation strategy as a means of guiding the search within the conformal bootstrap framework. These studies also aimed to devise

*niarchos@physics.uoc.gr
†c.papageorgakis@qmul.ac.uk
‡p.richmond@qmul.ac.uk
§a.g.stapleton@qmul.ac.uk
‖mitchell.woolley@qmul.ac.uk

[1]The input of exact spectral data into the conformal bootstrap has been considered for 2D CFTs in [3,4].

[2]For reviews, see [7–9], while for a recent state-of-the art, see [10,11].

TABLE I. Sample results for three OPE-coefficients squared in the 1D line-defect CFT of the Maldacena–Wilson line in planar 4D $\mathcal{N}=4$ SYM theory at 't Hooft coupling $\lambda=(4\pi)^2\simeq 157.91$ or $g=1$ in the notation of the main text. The precise definitions appear in Sec. III. The first line presents results from [2] using the linear-functional methods implemented with the semidefinite programming approach of SDPB [29], and the errors reflect rigorous upper/lower bounds. The second line presents a sample of our new results based on the use of an improved truncation method that employed an interior point optimization algorithm implemented with IPOPT [30]. The errors are statistical in this case and reflect $1\sigma$ deviation. In the main text, we present further results obtained using alternative optimizers. Both lines combine the crossing equations with two sum rules from integrated correlation functions. The full list of results is available in Table VII.

| Method | $C_1^2$ | $C_2^2$ | $C_3^2$ |
|---|---|---|---|
| SDPB in [2] | $0.294014873 \pm 4.88\times 10^{-8}$ | $0.039788 \pm 4.10\times 10^{-4}$ | $0.146757 \pm 5.82\times 10^{-4}$ |
| Improved truncation | $0.294014228 \pm 6.77\times 10^{-7}$ | $0.041832 \pm 1.86\times 10^{-3}$ | $0.144100 \pm 2.39\times 10^{-3}$ |

methodologies capable of handling substantial truncations, encompassing hundreds or even thousands of operators. This represents a notable improvement over earlier works, which employed more drastic truncations limited to $O(10)$ operators.

One of the main disadvantages of truncation methods is that they are subject to systematic errors that are hard to quantify, rendering them nonrigorous. Furthermore, determining the appropriate truncation can be a nontrivial task, and the interpretation of results obtained through such approaches may not always be straightforward. Conversely, truncation methods have advantages such as flexibility and computational cost effectiveness. They do not rely on positivity constraints, making them well suited for exploring the landscape of CFT data, including the cases of nonunitary theories, defect CFTs and analyses arising from bootstrapping higher-point functions. Therefore, truncation-based searches could be creatively employed in guiding targeted searches for specific theories, extracting dynamically viable gap assumptions and other information that a more rigorous method could employ at a later stage. We find this aspect particularly interesting and worthy of further exploration.

In this paper, we expand upon the bootstrability investigation of the 1D defect CFT but also introduce several improvements to previously employed truncation methods. These improvements allow us to accurately reproduce the findings of [1,2] with exceptional precision. This precision extends to the seventh decimal place for certain data, all the way from the weak to the strong coupling regimes. To obtain these results, we employed both reinforcement learning (as done in [21–23]), as well as more conventional nonlinear optimization algorithms. We used the same information from integrability methods (the scaling dimensions of 10 nonprotected operators) as developed in [25,26] and employed for bootstrability in [2].

Our main results can be summarized as follows:

(i) We present a substantially improved truncation scheme by introducing approximations for the "tail" of the OPE expansion, as well as "effective" operators.

(ii) We obtain numerical results for the OPE-coefficients squared of the first three unprotected multiplets of the 1D line-defect CFT, with/without utilizing the sum rules of integrated correlators and compare with [1,2]. A sample of these results is presented in Table I, with the full list available in Table VII.

(iii) We benchmark the soft actor-critic (SAC) reinforcement learning algorithm [27], used as a nonlinear optimizer, against the IPOPT implementation of the interior point method optimization algorithm. In this context, we provide specific evidence for the effectiveness of the average of statistical runs with the SAC algorithm and comment on the motivation to explore more advanced reinforcement learning algorithms. We highlight the fact that truncation methods using either of the above algorithms are computationally cheaper than the (rigorous) linear-functional methods. Our computations were performed using machine precision.

(iv) We significantly improve our Python implementation BootSTOP to: (a) use the improved truncation scheme in 1D, (b) work with CFTs in 1D, 2D and 6D, and (c) switch between SAC and the Python Parallel Global Multiobjective Optimizer (PyGMO) [28], which includes a host of deterministic and stochastic optimization algorithms, including IPOPT.

The rest of this paper is organized as follows. In Sec. II, we give a detailed account of our improved truncation scheme. In Sec. III, we provide a brief review of the necessary background needed to set up our bootstrap problem. In Secs. IV and V, we list our main results by reproducing the recently obtained values for the OPE-coefficients squared of the first three unprotected multiplets in the 1D defect CFT [1,2] using our SAC/IPOPT optimization protocols. We round off in Sec. VI by discussing the merits of different optimization algorithms, as well as presenting some predictions for additional data in the 1D defect CFT, before concluding in Sec. VII. Appendix A includes a full list of our results, while Appendix B presents an alternative analysis for the adiabatic variation

of the crossing equations, which could be used in future studies.[3]

## II. IMPROVED TRUNCATION SCHEMES

One of our objectives in this paper is to investigate enhancements to truncation methods. Specifically, we aim to develop a novel framework for interpreting the operators and associated CFT data within a truncation scheme. We will attempt a systematic treatment in contexts where the CFT is a member of a parametric family (with continuous or discrete parameters), by assuming that the theory can be defined/solved in a specific corner of that family. Let us call this corner of parameter space the *defining corner*. That corner typically reflects a weak coupling formulation and could be a free fixed point or a generalized free point (possibly captured by a dual supergravity description). The known spectrum of the defining corner forms the starting point for an informed truncation of the spectrum. The goal of the program is to explore how the CFT data evolve (adiabatically) across the parameter space.[4] This strategy fits well within a more general approach that prioritizes the exploration of specific theories, compared to an exploration of general properties in the space of CFTs.

The ensuing discussion will be kept generic and applies to CFTs in any number of spacetime dimensions. For concreteness, we will focus on a single crossing equation, but similar methods can also be applied to the multi-correlator bootstrap or to the bootstrap with the addition of extra sum rules. The typical bootstrap problem involves an algebraic crossing equation of the general form

$$\sum_n \mathcal{C}_n F_n(x) + r(x) = 0, \tag{1}$$

where the index $n$ runs over the infinite number of operators that appear in the conformal block expansion (in multiple channels). In two and higher spacetime dimensions, $n$ enumerates operators of different spin and scaling dimension. In one dimension, where there is no concept of spin, $n$ simply labels operators at different scaling dimensions. $\mathcal{C}_n$ denotes the OPE-coefficients squared, and $F_n$ is shorthand notation for the (crossed) conformal blocks. The variable $x$ represents the single cross ratio present in 1D CFTs or collectively the pair of complex cross-ratios $(z, \bar{z})$ in higher-dimensional CFTs. The function $r(x)$ is an in-homogeneous contribution, which is assumed to be explicitly known.[5] This term may depend on external continuous or discrete parameters. The CFT data encoded in (1) are the OPE-coefficients squared $\mathcal{C}_n$ and the corresponding scaling dimensions $\Delta_n$.

The cross-ratio dependence of the crossing equation can be discretized, either by evaluating it on a grid of $x$ points [33] or by applying a finite number of linear functionals. A popular basis of linear functional in the conformal bootstrap literature consists of derivatives at the crossing-symmetric point [7–9]. For the 1D applications of the upcoming sections, we used (even) derivatives at $x = \frac{1}{2}$. This discretization reduces the continuous character of the algebraic equation (1) to a finite subset of equations, which we collect in a finite-dimensional vector. Accordingly, we recast Eq. (1) into the vector form:

$$\sum_n \mathcal{C}_n \vec{F}_n + \vec{r} = 0. \tag{2}$$

Let us now split the full set of operators appearing in (2) into a finite subset, call it $\mathcal{S}$, and its complement. The selection of $\mathcal{S}$ can be based on various criteria, which we do not have to specify at the moment. Typically, we are interested in a subset of "the most significant" operators. In the Euclidean bootstrap around the crossing symmetric point, where the conformal block expansion converges exponentially fast [34], these are operators with relatively low scaling dimensions.[6] Consequently, we can now recast Eq. (2) into the more refined form

$$\sum_{n\in\mathcal{S}} \mathcal{C}_n \vec{F}_n + \vec{T} + \vec{r} = 0, \tag{3}$$

where $\vec{T}$ captures the contribution of the operators in the *complement* of $\mathcal{S}$, which we will call the "tail," and the sum over $n$ now involves only a finite number of terms and corresponding CFT data. Thus far, (3) is exact.

The main premise of truncation methods, up to this point in the literature, involves dropping the tail contribution $\vec{T}$ completely and analyzing the resulting equation,

$$\vec{E} \coloneqq \sum_{n\in\mathcal{S}} \mathcal{C}_n \vec{F}_n + \vec{r} \simeq 0, \tag{4}$$

which can only be satisfied approximately. In [13], the analysis of the truncated equations proceeds via the method of determinants. In [20], and subsequently in [21,22], one formulates a positive semidefinite function $\mathcal{L}$ of the vector $\vec{E}$ and tries to minimize the "cost" function

$$\mathrm{Cost}[\{\Delta_n, \mathcal{C}_n\}_{n\in\mathcal{S}}] \coloneqq \mathcal{L}[\vec{E}]. \tag{5}$$

[3]After a preprint of this work appeared on the arXiv, we learned of similar explorations of the 1D defect CFT using BootSTOP in the master's thesis [31]. We thank P. Ferrero for communication on this point.

[4]A recent study of the Ising CFT using a different adiabatic deformation in spacetime dimension appeared in [32].

[5]In the upcoming Eq. (31), $r(x) = \mathcal{G}_{\mathrm{simple}}(g, x)$.

[6]Higher-dimension operators also play a significant role in our approach, and how we incorporate them into $\mathcal{S}$ is part of our discussion. Summarizing remarks related to this aspect appear in Sec. VI. More generally, we expect that higher-dimension operators will eventually become increasingly important in hybrid numerical/analytical bootstrap methods; see, e.g., [35] for a recent discussion.

$\mathcal{L}$ quantifies the deviation of $\vec{E}$ from the zero vector and should therefore vanish at zero by definition. A typical choice of $\mathcal{L}$ is the root mean square, but other options can also be explored.

We want to depart, slightly, from this logic by keeping the tail $\vec{T}$ with a suitable approximation and obtaining a better understanding of what the data $\{\Delta_n, \mathcal{C}_n\}_{n\in\mathcal{S}}$ represent in an approximate scheme, with or without $\vec{T}$. Part of the problem relates to the fact that in truncations with many operators, the higher-dimension CFT data can be redistributed by the optimization algorithm in many different ways to collectively capture a similar overall, approximate contribution to the cost function. This includes configurations where operators are grouped together in narrow bands of scaling dimensions, effectively reducing the number of active, independent CFT data in the truncation. There is also an interplay between this freedom and the dynamics of the tail that affects the complexity of the optimization problem and the interpretation of the results for a given truncation scheme.

### A. Tail approximation

As mentioned in the beginning of this section, we will assume that the theory of interest is part of a family of CFTs and that there is at least one corner in parameter space where it can be solved explicitly with traditional methods. In order to set up a convenient language, let us collectively denote the external parameters $\lambda$ and their value at the defining corner $\lambda^*$. The parameters $\lambda$ could be discrete (e.g., the rank of a gauge group) or continuous (e.g., the value of an exactly marginal coupling). In general, the notation $\lambda$ is shorthand for a multiparameter vector.

The existence of an explicit solution at $\lambda^*$ has several useful consequences. First, the solution at $\lambda^*$ can be used to inform the choice of truncation, namely the set of operators $\mathcal{S}$ in the crossing equation (3). For instance, this can be done by picking a cutoff on the scaling dimension or twist, so as to specify the number of operators that we want to include in $\mathcal{S}$ for each spin. In [21,22], this choice informed a corresponding "spin-partition," With the number of data appearing in $\mathcal{S}$ specified, our goal takes the following form:

"Solve (3) to determine how the data $\{\Delta_n, \mathcal{C}_n\}_{n\in\mathcal{S}}$ vary across the parameter space from $\lambda^*$ to a generic value of $\lambda$."

The quality of the results can depend nontrivially on the tail $\vec{T}$, which includes the value of the CFT data of all the hidden operators in the complement of $\mathcal{S}$. As a first step toward a better approximation of $\vec{T}$ (compared to simply setting $\vec{T}=0$), we propose the following approach: At $\lambda^*$, we assume having access to the CFT data $\{\Delta_n^*, \mathcal{C}_n^*\}_{n\in\mathcal{S}}$ inside the truncation, and the Eq. (3) can be satisfied exactly:

$$\sum_{n\in\mathcal{S}} \mathcal{C}_n^* \vec{F}_n^* + \vec{T}^* + \vec{r}^* = 0. \tag{6}$$

Most importantly, we can use this equation to determine the exact value of the tail at $\lambda^*$:

$$\vec{T}^* = -\sum_{n\in\mathcal{S}} \mathcal{C}_n^* \vec{F}_n^* - \vec{r}^*. \tag{7}$$

If the tail $\vec{T}$ does not vary significantly as a function of $\lambda$, then a first approximation of the tail consists of setting

$$\vec{T}(\lambda) \simeq \vec{T}^*. \tag{8}$$

In such a case, the exact Eq. (3) is approximated by

$$\sum_{n\in\mathcal{S}} \mathcal{C}_n \vec{F}_n + \vec{r} - \sum_{n\in\mathcal{S}} \mathcal{C}_n^* \vec{F}_n^* - \vec{r}^* \simeq 0, \tag{9}$$

and leads to the minimization of the modified cost function

$$\widetilde{\mathrm{Cost}}[\{\Delta_n, \mathcal{C}_n\}_{n\in\mathcal{S}}] \coloneqq \mathcal{L}[\vec{E} - \vec{E}^*], \tag{10}$$

with $\vec{E}$ defined in (4).

The approximate assumption in (8) is not unrealistic (for sufficiently large truncations) in the vicinity of $\lambda^*$ and certainly improves the drastic truncation ansatz $\vec{T}=0$. Indeed, there is now at least one point in parameter space where the crossing equations are satisfied exactly by construction. The assumption (8) is also motivated by the fact that high-dimension operators have minimal contribution to the conformal block expansion around the crossing-symmetric point, and that in the limit of high spin, CFT states behave asymptotically as generalized free fields [36,37]. Nevertheless, whether this approximation holds for a finite deformation away from $\lambda^*$ (and to what degree) is not obvious and is certainly critical. In general, one can imagine various ways in which (8) can break down. For example, as one deforms away from $\lambda^*$ and the spectrum rearranges itself, some operators from the tail can become increasingly important. The tail contribution can also be affected when the scaling dimensions of the external operators are $\lambda$ dependent.[7]

### B. Effective operators and degeneracies

Setting the approximation of the tail aside for the moment, another issue that affects the complexity and efficiency of a truncation scheme relates to the presence of large accidental degeneracies. This usually occurs in the defining corner at $\lambda^*$ that involves a (generalized) free field description; the weak coupling regimes of gauge

[7]For example, one can explicitly write down the crossing equations in the 2D $S^1$ CFT for scalar, charged primaries [21,22] and check the value of the tail for a fixed truncation as a function of the external operator dimensions. As one moves on the conformal manifold, the value of the external dimension changes and the tail exhibits significant variations. We would like to thank A. Stratoudakis for working out specific examples of this type.

theories are typical examples. Large degeneracies are challenging for two reasons: First, they can grow very rapidly as functions of the scaling dimension. In that case, a complete description of the degenerate spectrum would force a truncation with high dimensionality. Second, away from $\lambda^*$, the accidental degeneracies are typically lifted, and tracking the precise splitting across the parameter space can be a very complicated task. One might therefore ask: Is it possible to alleviate the problems that arise in such situations?

To isolate the effects of nearly degenerate operators, let us assume that part of the sum $\sum_{n\in\mathcal{S}} \mathcal{C}_n \vec{F}_n$ in (9) involves a relatively narrow band of $\mathfrak{N}_{\mathtt{band}}$ operators (at the same spin) with scaling dimensions $\Delta \in \mathtt{band}$, where $\mathtt{band} \equiv [\Delta_{\min}, \Delta_{\max}]$ and $\Delta_{\max} > \Delta_{\min}$. We will denote their contribution to the crossing equation as

$$\vec{\mathfrak{E}} = \sum_{\Delta\in\mathtt{band}\subset\mathcal{S}} \mathcal{C}_n \vec{F}_n. \tag{11}$$

For an exact solution to the crossing equations, this vector takes a specific value

$$\vec{\mathfrak{E}}^{(\mathrm{exact})} = \sum_{\Delta\in\mathtt{band}\subset\mathcal{S}} \mathcal{C}_n^{(\mathrm{exact})} \vec{F}_n^{(\mathrm{exact})}. \tag{12}$$

We want to explore the possibility of approximating the exact vector $\vec{\mathfrak{E}}^{(\mathrm{exact})}$ with an effective sum

$$\vec{\mathfrak{E}}^{(\mathrm{eff})} = \sum_{\mathcal{O}_{\mathrm{eff}}} \mathcal{C}_{\mathcal{O}_{\mathrm{eff}}} \vec{F}_{\mathcal{O}_{\mathrm{eff}}}, \tag{13}$$

over a reduced number $\mathfrak{N}_{\mathtt{eff}}$ of operators. Crucially, the CFT data of these operators do not capture the exact data of the CFT in the band. They are meant to provide an effective description that approximates the contribution $\vec{\mathfrak{E}}^{(\mathrm{exact})}$ inside the crossing equation.

A special instance where this effective description is exact is that of exact degeneracies. In that case, there may be a possibly large number of distinct operators, $\mathfrak{N}_{\mathtt{band}} > 1$, that contribute to the sum $\vec{\mathfrak{E}}^{(\mathrm{exact})}$ in (12). However, since all of them have the same scaling dimension $\Delta$ (and the same corresponding conformal block $\vec{F}_\Delta^{(\mathrm{exact})}$), the vector $\vec{\mathfrak{E}}^{(\mathrm{exact})}$ is effectively encoding the contribution of a single operator ($\mathfrak{N}_{\mathrm{eff}} = 1$) with OPE-coefficient squared equal to the sum of the OPE-coefficients squared of the individual degenerate operators:

$$\vec{\mathfrak{E}}^{(\mathrm{exact})} = \left(\sum_{\Delta_n=\Delta} \mathcal{C}_n^{(\mathrm{exact})}\right) \vec{F}_\Delta^{(\mathrm{exact})}. \tag{14}$$

Therefore, from this single effective operator, only the scaling dimension $\Delta$ and total OPE-coefficient squared $(\sum_{\Delta_n=\Delta} \mathcal{C}_n^{(\mathrm{exact})})$ can be read off.

More generally, in a band of finite size, one can write

$$\vec{\mathfrak{E}} = \sum_{\Delta\in\mathtt{band}} \mathcal{C}_n \vec{F}_n = \bar{\mathcal{C}} \sum_{\Delta\in\mathtt{band}} c_n \vec{F}_n, \tag{15}$$

where we defined

$$\bar{\mathcal{C}} \coloneqq \sum_{\Delta\in\mathtt{band}} \mathcal{C}_n, \qquad c_n \coloneqq \frac{\mathcal{C}_n}{\bar{\mathcal{C}}}. \tag{16}$$

With this definition, and assuming that $\mathcal{C}_n \geq 0$ by unitarity, the new coefficients $c_n$ are by construction numbers inside the interval [0, 1] with the property $\sum_n c_n = 1$. Consequently, the vector $\vec{\mathfrak{e}}^{(\mathrm{exact})} = \sum_{\Delta\in\mathtt{band}} c_n^{(\mathrm{exact})} \vec{F}_n^{(\mathrm{exact})}$ of the exact solution is inside the convex combination $\mathbf{\Sigma}_{(\mathrm{exact})}$ of the $\mathfrak{N}_{\mathtt{band}}$ vectors $\vec{F}_n^{(\mathrm{exact})}$. Moreover, the convex combination $\mathbf{\Sigma}_{(\mathrm{exact})}$ is inside the convex hull $\boldsymbol{H}[\Delta_{\min}, \Delta_{\max}]$ of the segment of the curve $\vec{F}_\Delta$ for $\Delta \in [\Delta_{\min}, \Delta_{\max}]$. The latter is a set that characterizes the conformal blocks independently of the details of the exact solution of the $\mathfrak{N}_{\mathtt{band}}$ operators in the band. To summarize, the exact contribution of the band to the crossing equation is the vector

$$\vec{\mathfrak{E}}^{(\mathrm{exact})} = \bar{\mathcal{C}} \vec{\mathfrak{e}}^{(\mathrm{exact})} \tag{17}$$

with

$$\vec{\mathfrak{e}}^{(\mathrm{exact})} \in \mathbf{\Sigma}_{(\mathrm{exact})} \subset \mathbf{H}[\Delta_{\min}, \Delta_{\max}]. \tag{18}$$

When the $\mathfrak{N}_{\mathtt{band}}$ operators are replaced by $\mathfrak{N}_{\mathtt{eff}} < \mathfrak{N}_{\mathtt{band}}$ operators, the quality of the approximation will depend on the minimal distance between the convex combination $\mathbf{\Sigma}_{(\mathrm{eff})}$ of the $\mathfrak{N}_{\mathtt{eff}}$ vectors $\vec{F}_n^{(\mathrm{eff})}$ and the convex combination $\mathbf{\Sigma}_{(\mathrm{exact})}$ as the scaling dimensions of the effective operators vary. Assuming the latter vary inside the same band $[\Delta_{\min}, \Delta_{\max}]$ as the scaling dimensions of the exact configuration, both convex combinations $\mathbf{\Sigma}_{(\mathrm{eff})}$ and $\mathbf{\Sigma}_{(\mathrm{exact})}$ are subsets of the same convex hull $\mathbf{H}[\Delta_{\min}, \Delta_{\max}]$. This puts an indirect upper bound on the error of the approximation of the exact configuration.

It is not easy to promote these observations into specific quantitative predictions in generic situations or to use them to develop a concrete strategy for the selection of the effective operators. We wanted, however, to highlight these features for two reasons.

First, we believe that an effective description of a complicated spectrum can be an important tool that can be used to reduce the complexity of the problem. For relatively narrow bands of nearly degenerate operators, one might expect reasonable results with cheap effective descriptions. Moreover, parametrizing ignorance with an effective description may lead to a better interpretation of the output of a computation. For instance, if there is confidence in the existence of a nearly degenerate band for a given problem, then instead of trying to interpret

specific numbers as individual predictions for actual CFT data, it may be more appropriate to interpret those results as features of an effective description. In that case, from the spread of the scaling dimensions of the effective operators, one may want to distill a prediction for the size of the band, and from the overall coefficient $\bar{C}$ in (15), one may want to distill an approximate sum rule for the total OPE-coefficient squared in the band.

A second related motivation for the above discussion is that sometimes, during the optimization steps in a high-dimensional truncation, an algorithm (or two separate algorithms) may identify two distinct high-reward configurations with one of them having rendered several operators nearly degenerate. Rather than interpreting these two configurations as results corresponding to two distinct theories with a different number of operators, it may be more appropriate to view them as different effective representations of the same theory.

### C. Implementation and a soft extension of the tail

In the last subsection, we attempted to isolate effects inside some relatively narrow band of operators. Let us now return to the complete problem and the approximate truncation scheme (9).

In the defining corner at $\lambda^*$, we understand the structure of the spectrum and how it is captured by our chosen truncation. As we deform the theory away from $\lambda^*$, we can now envision the emergence of the following complications: Nearly degenerate bands (possibly captured by a reduced set of effective operators) can develop significant splits, operator scaling dimensions can cross and the naive approximation of the tail at $\lambda^*$ may cease to be accurate. The latter will force the operators inside our truncation set $\mathcal{S}$ to readjust appropriately. How one proceeds at this point depends on the situation and will typically require additional external input in order to extract confident results. For example, such an input could arise by considering the simultaneous information from multiple correlators, the combination of a truncation scheme with a navigator method based on the linear-functional approach [38] and/or input from OPE inversion formulas [39–41]. We plan to explore all these possibilities in future work. In the present paper, the external input that we use are the exact scaling dimensions for 10 operators from the quantum spectral curve.

Accordingly, our results in Secs. IV and V are obtained with a truncation of 62 operators, further split into 10 operators, the scaling dimensions of which we can track explicitly, and the remaining 52 operators that we treat as effective. We will not attempt to make any predictions for actual CFT data based on these effective operators. We sum their contribution to the crossing equation and treat it as part of a soft extension of the initial tail approximation $\vec{T}^*$ at zero 't Hooft coupling. We will provide concrete evidence that this soft extension of the tail is a valid approximation at all values of the coupling. We will also see that different algorithms treat the 52 effective operators in different ways.

## III. REVIEW OF DEFECT CFT IN $\mathcal{N}=4$ SYM

Before delving into the details of our numerical computations, we begin with a lightning summary of the 1D line-defect CFT, highlighting only the aspects that are necessary for our discussion. For a complete account, we refer the reader to [6] and references therein. The line-defect CFT resides on a straight, infinite Maldacena–Wilson line

$$\mathcal{W} = \mathrm{Tr\,P}\,\exp\int_{-\infty}^{+\infty}(A_t + \Phi_{||})dt, \tag{19}$$

which preserves an $\mathfrak{osp}(4^*|4)$ subalgebra of the full superconformal algebra of the parent $\mathcal{N}=4$ super Yang–Mills theory in four dimensions [42] and inherits its integrable structure in the planar limit [43,44]. In (19), $A_t$ and $\Phi_{||}$ are gauge and real scalar-field components, respectively. The maximal bosonic subalgebra involves the 1D conformal algebra, the $\mathfrak{sp}(2)_R$ $R$ symmetry and the algebra of $\mathfrak{so}(3)$ rotations transverse to the line defect in four-dimensional spacetime (sometimes referred to as "spin"). All states in the line CFT fall into unitary irreducible representations of this superconformal algebra, the superconformal primaries of which are scalars under the $\mathfrak{so}(3)$ global symmetry. The irreducible representations include short $\mathcal{B}_k$ (protected) representations, the dimension of which is fixed by their $\mathfrak{sp}(2)_R$ quantum numbers $[0,k]$, $\Delta = k$, and long $\mathcal{L}^{\Delta}_{[0,0]}$ (unprotected) representations, which are $\mathfrak{sp}(2)_R$ scalars.

We are interested in four-point functions arising from local-operator insertions along the Maldacena–Wilson line. More specifically, we are interested in identical insertions of one of the real scalars of $\mathcal{N}=4$ SYM $\Phi^i_\perp$, $i=1,\ldots,5$, not appearing in (19):

$$\begin{aligned}&\langle\!\langle \Phi^1_\perp(t_1)\Phi^1_\perp(t_2)\Phi^1_\perp(t_3)\Phi^1_\perp(t_4)\rangle\!\rangle \\ &\quad := \langle \mathrm{Tr} W^{t_1}_{-\infty}\Phi^1_\perp(t_1)W^{t_2}_{t_1}\Phi^1_\perp(t_2)W^{t_3}_{t_2}\Phi^1_\perp(t_3)W^{t_4}_{t_3}\Phi^1_\perp(t_4)W^{+\infty}_{t_4}\rangle.\end{aligned} \tag{20}$$

The $\Phi^1$ component is the superconformal primary of the $\mathcal{B}_1$ multiplet, known as the displacement multiplet, the OPEs of which obey the following selection rules:

$$\mathcal{B}_1 \times \mathcal{B}_1 = \mathcal{I} + \mathcal{B}_2 + \sum_{\Delta>1}\mathcal{L}^{\Delta}_{[0,0]}. \tag{21}$$

A crossing equation arises from (20) due to the invariance of the four-point function under a cyclic relabelling of the insertion points, which can be recast as

$$x^2 f(1-x) + (1-x)^2 f(x) = 0. \tag{22}$$

In this expression, $x$ is the single conformal cross-ratio in 1D,

$$x \coloneqq \frac{x_{12}x_{34}}{x_{13}x_{24}}, \qquad x_{ij} \coloneqq x_i - x_j. \tag{23}$$

As a consequence of (21), the function $f(x)$ admits a superconformal-block decomposition of the form

$$f(x) = F_{\mathcal{I}}(x) + C^2_{\mathrm{BPS}} F_{\mathcal{B}_2}(x) + \sum_n C_n^2 F_{\Delta_n}(x), \tag{24}$$

with the specific blocks given by

$$F_{\mathcal{I}}(x) = x, \tag{25}$$

$$F_{\mathcal{B}_2}(x) = x - x_2 F_1(1,2,4;x), \tag{26}$$

$$F_{\Delta_n}(x) = \frac{x^{\Delta_n+1}}{1-\Delta_n} {}_2F_1(\Delta_n+1, \Delta_n+2, 2\Delta_n+4; x), \tag{27}$$

involving standard hypergeometric functions.

In addition to fixing the dimension of the $\mathcal{B}_2$ primary from superconformal representation theory for all values of the 't Hooft coupling $\lambda$ of $\mathcal{N}=4$ SYM, one can also determine the value of the corresponding OPE-coefficients squared, $C^2_{\mathrm{BPS}}$, with the help of supersymmetric localization [45,46] or integrability methods [2]. The latter vary with the 't Hooft coupling and, when expressed as a function of $g \coloneqq \frac{\sqrt{\lambda}}{4\pi}$, read

$$C^2_{\mathrm{BPS}} = 1 - \mathbb{F}(g), \tag{28}$$

where

$$\mathbb{F}(g) = \frac{3(g^2 - \mathbb{B}(g))}{\pi^2 (\mathbb{B}(g))^2}, \tag{29}$$

and the Bremsstrahlung function is

$$\mathbb{B}(g) = \frac{g}{\pi} \frac{I_2(4\pi g)}{I_1(4\pi g)}, \tag{30}$$

involving modified Bessel functions of the first kind.

Using this information, the crossing equations (22) can be recast into the following compact form

$$\sum_n C_n^2 \mathcal{G}_{\Delta_n}(x) + \mathcal{G}_{\mathrm{simple}}(g,x) = 0, \tag{31}$$

where

$$\mathcal{G}_{\mathrm{simple}}(g,x) \coloneqq \mathcal{G}_{\mathcal{I}}(x) + C^2_{BPS}(g) \mathcal{G}_{\mathcal{B}_2}(x) \tag{32}$$

is a now a known function, with $\mathcal{G}_{\mathcal{I}}$ encoding the crossed superconformal blocks:

$$\mathcal{G}_{\mathcal{I},\mathcal{B}_2,\Delta_n}(x) \coloneqq (1-x)^2 F_{\mathcal{I},\mathcal{B}_2,\Delta_n}(x) + x^2 F_{\mathcal{I},\mathcal{B}_2,\Delta_n}(1-x). \tag{33}$$

Therefore, the undetermined quantities appearing in the superconformal-block expansion of (31) will involve the dimensions of the unprotected operators along with their corresponding OPE-coefficients squared. The lowest dimension long primary is $\Phi_{||}$. The long CFT data vary as a function of $g$, but one can make use of the quantum spectral curve (QSC) to numerically determine the evolution of the conformal dimensions of long operators from weak to strong coupling. These results arise from a 1D adaptation of a numerical QSC implementation for $\mathcal{N}=4$ SYM developed in [47,48]. The dimensions of the first 35 long operators for $g \in [0,2]$ were provided in [1], while those of the first 13 long operators for $g \in [0,4]$ were found using methods developed in [25,26] and used in [2].

In [1], the dimensions of the first two unprotected superconformal primaries were used as external input to the linear-functional method to determine bounds for the OPE coefficient of the first long multiplet entering (31). Preliminary results with the input of additional scaling dimensions from the QSC were also reported. In [2], the dimensions of only the first 10 unprotected superconformal primaries were used, but the crossing symmetry conditions were supplemented by two sum rules arising from integrated correlators, which were derived in [5,6], respectively. The incorporation of these sum rules was observed to yield significantly sharper bounds and better accuracy for the first three OPE-coefficients squared. These two additional constraints from integrated correlators can be brought to the following form:

$$\sum_{\Delta_n} C_n^2 \mathrm{Int}_i[F_{\Delta_n}(x)] + \mathrm{RHS}_i = 0 \quad \text{for } i = 1,2, \tag{34}$$

where

$$\mathrm{Int}_1[F_{\Delta_n}(x)] \coloneqq -\int_0^{\frac{1}{2}} (x - 1 - x^2) \frac{F_{\Delta_n}(x)}{x^2} \partial_x \log(x(1-x)) dx, \tag{35}$$

$$\mathrm{Int}_2[F_{\Delta_n}(x)] \coloneqq \int_0^{\frac{1}{2}} F_{\Delta_n}(x) \frac{(2x-1)}{x^2} dx, \tag{36}$$

with $F_{\Delta_n}(x)$ given in (27) and

$$\mathrm{RHS}_1 = \frac{\mathbb{B} - 3\mathbb{C}}{8\mathbb{B}^2} + \left(7\log(2) - \frac{41}{8}\right)(\mathbb{F} - 1) + \log(2), \tag{37}$$

$$\mathrm{RHS}_2 = \frac{1-\mathbb{F}}{6} + (2-\mathbb{F})\log(2) + 1 - \frac{\mathbb{C}}{4\mathbb{B}^2}. \quad (38)$$

The curvature function $\mathbb{C}(g)$ has analytic expansions at weak and strong coupling, while a numerical evaluation with high precision, which we used in our implementation, was provided in [6].

## IV. RESULTS WITHOUT INTEGRAL CONSTRAINTS

We now move on to present the main results of this paper. We begin with the analysis of the crossing symmetry conditions (31) without any input from the two integral constraints (34). We will fix the scaling dimensions of the first 10 long multiplets using the QSC and compare with the corresponding results in [2], which implemented the linear-functional method with SDPB. The inclusion of the integral constraints will be discussed separately in Sec. V.

Our goal is to extract information about the scaling dimensions and corresponding OPE-coefficients squared for operators in unprotected (long) multiplets that appear in the conformal block expansion of the four-point function (20). An adiabatic sequence of runs was performed on the HPC cluster at Queen Mary University of London (QMUL) starting at $g = 0.2$ up to $g = 4$ with step $\delta g = 0.2$. We analyzed derivatives of the crossing equations at the crossing symmetric point (as per usual practice in the conformal bootstrap program) and chose to normalize each derivative with a factor of $1/(2^p p!)$ at order $p$. Because of symmetry, only the even derivatives are nontrivial. In most of the reported results, we included all even derivatives up to order $N_{\mathrm{der}} = 260$, but we also performed runs with $N_{\mathrm{der}} = 700$.

### A. Implementation of algorithms

In this section, we will report independent results using two optimization methods. The first is based on the SAC algorithm, deployed as a stochastic optimizer and implemented through PyTorch. This is a reinforcement learning algorithm based on the concept of Markov decision processes, first introduced in this context in [21,22].[8] The second is the interior point optimization (IPOPT) algorithm [30], which is deterministic. We note that IPOPT is now accessible from within our coding framework for numerical bootstrap, BootSTOP [67], which incorporates all the algorithms of the Python Parallel Global Multiobjective Optimizer PyGMO [28,68].[9] Access to this library allows the user to choose from a large suite of standard deterministic and stochastic algorithms. In our problem, IPOPT seems to outperform other algorithms available in PyGMO (such as simulated annealing, particle swarm optimization and differential evolution); we have not, however, performed a full, systematic, comparative study of all the PyGMO options.

We used the highest precision possible in the PyTorch and PyGMO packages: floating point precision, which on 64-bit machines, corresponds to 16 decimal places. In order to improve the run-time of the algorithms, we pregenerated the values of the differentiated conformal blocks. We found a closed form expression for the $p^{\mathrm{th}}$ derivative and verified this formula in *Mathematica* up to order $p = 20$. Beyond this order, *Mathematica* became extremely slow so we opted to evaluate the $N_{\mathrm{der}} = 700$ derivatives in Python using the MPMATH package. While this package allows for arbitrary precision, we chose 20 decimal places for all intermediate calculations before reducing to 16 for the final output. Cross-checks with *Mathematica* were performed when this was possible. Each derivative was evaluated on a lattice of conformal weights starting from 0 and ending at 20 with an increase of $10^{-4}$ between lattice sites. Conformal blocks on scaling dimensions in-between the points of the grid were evaluated with linear interpolation. The pregeneration of conformal blocks and our setup within PyTorch and PyGMO are some of the main obstacles toward arbitrary numerical precision in our implementation and the reason why we restricted our computations to machine precision. It is encouraging that this compromise did not significantly affect the quality of our final results.

Both SAC and IPOPT algorithms were employed to minimize the $L^2$-norm cost function of the difference between the value of the crossing equations (31) at each coupling $g$ and its corresponding value at $g = 0$. This approach implements the improved truncation scheme of Sec. II with the tail approximated by its value at $g = 0$, as set up in (10). We will also be frequently referring to the corresponding "reward" of a configuration, defined as the inverse of the $L^2$-norm cost function.

### B. Choice of truncation

The set $\mathcal{S}$ of long operators that we included in this truncation was informed by the structure of the spectrum at $g = 0$. At this free point, the scaling dimensions of long operators are integer, $\Delta \equiv J = 1, 2, \ldots$. With the exception of $J = 1$, all other levels are degenerate, with degeneracies

[8]There have been many recent applications of machine learning techniques to high-energy theoretical physics. An incomplete list of references includes the exploration of string vacua [49–51], integrability [52,53], the construction of numerical Calabi–Yau metrics for string compactifications [54–56], interplays with Wilsonian renormalization in quantum field theory [57–61], string field theory [62] and lattice quantum field theory [63,64]. For a recent review, see [65]. For an introduction to reinforcement learning and the SAC algorithm, see [66].

[9]BootSTOP currently contains libraries for conformal blocks in 1D, 2D, and 6D, necessary for attacking CFTs in the corresponding spacetime dimensions with truncation methods. We intend to make further updates with 3D and 4D conformal blocks in the near future.

TABLE II. The number of operators included in our truncation. The operators are allocated in groups characterized by the $g = 0$ value of their scaling dimension, $J$. For the low-lying operators at $J = 1, 2, 3$, this exactly matches the known $g = 0$ degeneracy. For $J \geq 4$, the number of chosen operators is smaller than the expected $g = 0$ degeneracy. We used more operators at $J = 4$ (3 above the reported 19 operators in [1,2]) and gradually less at higher values of $J$. All the operators in the groups $J \geq 4$ are therefore effective. In total, our truncation involves 62 operators with 124 corresponding CFT data.

| $J$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| # of operators in truncation | 1 | 2 | 6 | 22 | 8 | 8 | 5 | 4 | 3 | 3 |

that can be determined in principle. Our choice is detailed in Table II. The operators were grouped according to their $g = 0$ scaling dimension, $J$. Only the first nine states at $J = 1, 2, 3$ match the exact degeneracy at $g = 0$. All the states at higher values of $J$ are effective in the sense of Sec. II B.

As the coupling $g$ is increased, the spectrum rearranges itself and the free-theory degeneracies are lifted. At each search cycle, we opted to reorder the operators within the same $J$ family according to their scaling dimension, as we would normally do in higher-dimensional CFTs for the tower of states at each spin [23]. Such a choice allowed us to track how the groups of nearly degenerate states evolved with increasing coupling. However, we note that $J$ is not a spin quantum number. All the operators within the truncation contribute with the same type of conformal block, and at higher values of $g$, the mixing between different $J$ sectors is significant. This is a characteristic difficulty of 1D CFTs that does not exist in higher dimensions.

Therefore, in all our runs, we included 62 operators in the truncation, which amounts to 124 CFT data (62 scaling dimensions and 62 corresponding OPE-coefficients squared), although technically our numerical algorithms can also handle efficiently many more operators. Input from the QSC was used for the scaling dimensions of the first nine operators at $J = 1, 2, 3$ and for the lowest one at $J = 4$. Our main results refer to the OPE-coefficients squared of the $J = 1, 2$ operators, denoted, respectively, as $C_1^2, C_2^2, C_3^2$.

### C. Specifics of the SAC runs

The SAC searches were implemented with $N_{\text{der}} = 260$ derivative constraints. We employed 200 agents, each allowed to run on the QMUL HPC cluster for a maximum of 23 hours.[10] The search windows (`guess_sizes_deltas` and `guess_sizes_opes` in the code) were set to 0.4 for the conformal dimensions of the 52 unfixed long multiplets, $2 \times 10^{-2}$ for the OPE-coefficients squared of the first 47 long multiplets and $2 \times 10^{-3}$ for the OPE-coefficients squared of the last 15 long multiplets. Each run was performed around the (reward-weighted) average of the solutions at the previous value of $g$.

In the SAC implementation, the scaling dimensions of the 10 "fixed" long multiplets were not completely fixed. They were allowed to vary with a small $10^{-3}$ search window around the solution at the previous value of $g$. We stress that in SAC this value does not control the size of the box inside which the search is performed. Instead, it controls the maximum size of the next action. In this manner, if the starting value of a datum is not in the vicinity of a local minimum, the algorithm can eventually wander off significantly, even with a small search window.

At the end of the runs for the 200 SAC agents at each $g$, we also did an independent search with IPOPT inside a $4\sigma$ area around the average of the SAC result. Two sets of independent IPOPT runs were executed here, one with $N_{\text{der}} = 260$ and another with $N_{\text{der}} = 700$ derivative constraints. These follow-up IPOPT runs were performed with $2 \times 10^8$ agents (subdivided on the cluster into 2000 jobs, each with a population of 100 000 in the PyGMO architecture). They increased the reward significantly by a couple of orders of magnitude, but did not move the SAC averages. The addition of more derivative constraints did not lead to significant improvement either. We will make additional comments regarding these features in Sec. VI.

### D. Specifics of the IPOPT runs

The IPOPT algorithm was employed with $4 \times 10^8$ agents. These runs were subdivided into groups with a population of 100 000 within the PyGMO architecture. Each group was run 4000 times on the QMUL HPC cluster with an approximate 20 minutes run-time. Our final statistics for this approach comprise the 200 HPC cluster runs with the highest reward. The results we report in Table III were obtained with $N_{\text{der}} = 260$ derivative constraints. The box of the overall search was fixed within the hyperparameters of the algorithm. We chose $\pm 1$ for the scaling dimensions and $\pm 0.2$ for the OPE coefficients, around the solution at the previous value of $g$. We also enforced as extra lower bounds the free-limit value for the scaling dimension in each $J$ family and 0 for the OPE-coefficients squared. In contrast to the SAC runs, the first 10 long operators had their dimensions completely fixed to the results of the QSC. To further assist the search, we imposed on both algorithms (SAC and IPOPT) the additional constraints $C_2^2 < 0.1$ and $C_3^2 > 0.1$.

### E. Results

A sample of the results obtained with SAC and IPOPT (from $g = 0.2$ to $g = 1$ with step $\delta g = 0.2$) appears

[10] We observed that most agents were approaching their final configuration roughly within the first 12 hours. We did not attempt to optimize the scheduling of the algorithm, opting to allow for a longer search. In BootSTOP, SAC was implemented with parameters: `faff_max` = 5000, `pc_max` = 6, `window_rate` = 0.7, `max_window_exp` = 30.

TABLE III. Partial list of results (for $g \in [0.2, 1]$) from IPOPT and SAC runs with no integral constraints imposed. The errors in our results encode one standard deviation around the statistical reward-weighted average. For quick reference, we have also included at each value of $g$ the results for $C_1^2$ from Ref. [1]. In that case, the errors are rigorous and have a distinctly different meaning.

| Method | $g$ | $C_1^2$ | $C_2^2$ | $C_3^2$ |
|---|---|---|---|---|
| [1] | 0.2 | $0.0663 \pm 1.9 \times 10^{-3}$ | | |
| IPOPT | 0.2 | $0.06607342 \pm 4.18 \times 10^{-5}$ | $0.04708 \pm 2.04 \times 10^{-3}$ | $0.1630 \pm 2.69 \times 10^{-3}$ |
| SAC | 0.2 | $0.06733947 \pm 1.26 \times 10^{-3}$ | $0.06506 \pm 1.05 \times 10^{-2}$ | $0.1384 \pm 1.47 \times 10^{-2}$ |
| [1] | 0.4 | $0.1684 \pm 1.9 \times 10^{-3}$ | | |
| IPOPT | 0.4 | $0.16944584 \pm 8.35 \times 10^{-5}$ | $0.02659 \pm 3.45 \times 10^{-3}$ | $0.17965 \pm 4.90 \times 10^{-3}$ |
| SAC | 0.4 | $0.16824002 \pm 1.00 \times 10^{-3}$ | $0.06380 \pm 1.37 \times 10^{-2}$ | $0.14198 \pm 1.80 \times 10^{-2}$ |
| [1] | 0.6 | $0.2329 \pm 9 \times 10^{-4}$ | | |
| IPOPT | 0.6 | $0.233574606 \pm 1.32 \times 10^{-4}$ | $0.02533 \pm 6.78 \times 10^{-3}$ | $0.17382 \pm 7.68 \times 10^{-3}$ |
| SAC | 0.6 | $0.232721152 \pm 3.24 \times 10^{-4}$ | $0.06151 \pm 5.46 \times 10^{-3}$ | $0.13363 \pm 6.77 \times 10^{-3}$ |
| [1] | 0.8 | $0.2701 \pm 5 \times 10^{-4}$ | | |
| IPOPT | 0.8 | $0.270632286 \pm 6.67 \times 10^{-5}$ | $0.020165 \pm 6.29 \times 10^{-3}$ | $0.17218 \pm 7.06 \times 10^{-3}$ |
| SAC | 0.8 | $0.270121362 \pm 2.93 \times 10^{-4}$ | $0.05776 \pm 5.00 \times 10^{-3}$ | $0.13110 \pm 5.35 \times 10^{-3}$ |
| [1] | 1.0 | $0.29388 \pm 2.7 \times 10^{-4}$ | | |
| IPOPT | 1.0 | $0.294177967 \pm 6.79 \times 10^{-5}$ | $0.023344 \pm 9.64 \times 10^{-3}$ | $0.163302 \pm 1.04 \times 10^{-2}$ |
| SAC | 1.0 | $0.293941106 \pm 3.03 \times 10^{-4}$ | $0.05658 \pm 5.81 \times 10^{-3}$ | $0.127135 \pm 6.26 \times 10^{-3}$ |

in Table III. The full list of results can be found in Table VII of Appendix A. In Fig. 1, we plot the full set of results against the background of Fig. 6 from Ref. [2], which contains the rigorous upper/lower bounds derived with the linear-functional method and SDPB. The averages and statistical errors of the CFT data from the SAC and IPOPT runs are defined as averages weighted by the square of the ratio of current reward to the best reward.

Furthermore, in the first data row for each value of $g$ in Table III, we have also included for quick comparison the results for $C_1^2$ from Ref. [1]. These were obtained using Algorithm 1 in Ref. [2], which employed only

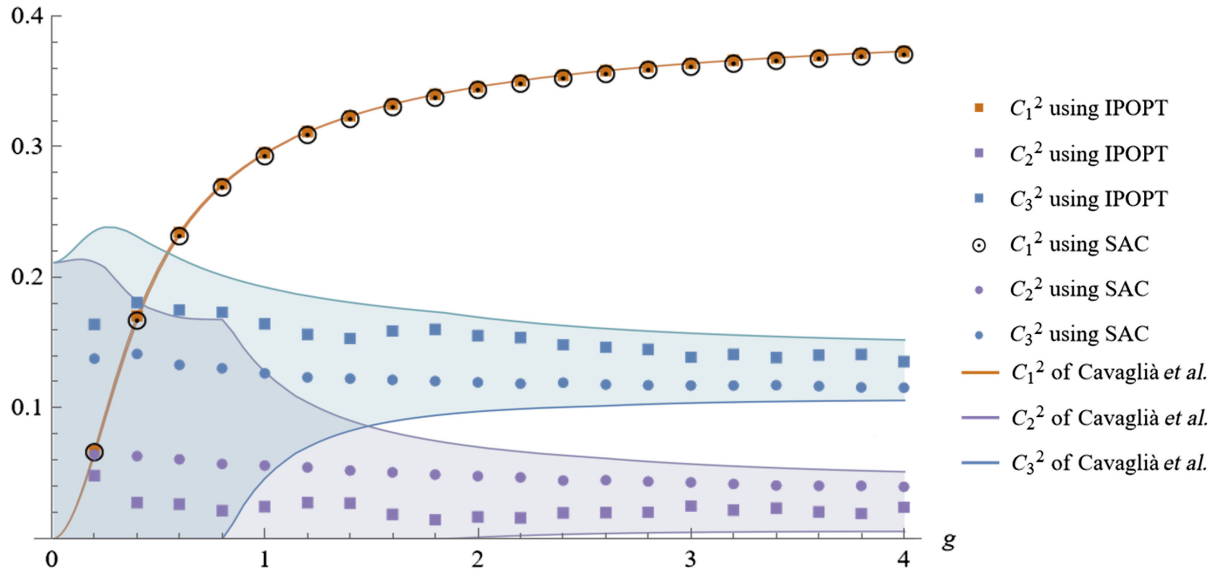


FIG. 1. Results for the OPE-coefficients squared of the first three long operators with no integral constraints. The solid lines indicate the rigorous bounds presented in Fig. 6 of [1], reprinted here with permission from the authors. Same-colored circles and squares indicate our results from the SAC and IPOPT runs, respectively. The corresponding statistical errors are too small to display on this plot but can be found in Table VII.

two scaling dimensions from the QSC. Incorporating the QSC data of 10 scaling dimensions with Algorithm 2 in Ref. [2] yields slightly improved upper/lower bounds.

The most characteristic features of our results are the following:

(1) The results for the OPE-coefficient squared $C_1^2$ are directly comparable with the corresponding results in Refs. [1,2], with agreement at the level of the third decimal point or higher.

(2) As is apparent from Fig. 1, our results for the OPE-coefficients squared $C_2^2$ and $C_3^2$ are always well within the bounds of [2]. Interestingly, SAC and IPOPT have not produced the same average configurations exploiting the effective operators in different ways. toward strong coupling, the spread between the SAC and IPOPT averages seems to be an indirect probe of the rough size of the rigorous allowed regions obtained with the linear-functional method.

(3) The linear-functional method can be quite sensitive to the choice of spectral data imported from the QSC. It was observed in [2] that their algorithms ceased to converge if the spectrum deviates significantly from the QSC answer. For example, at $g = 3$, it is sufficient to introduce an error of the order of $5 \times 10^{-7}$ in the spectrum, for the method that sets bounds for $C_1^2$ to no longer converge [2]. In light of this, the fact that the SAC runs did not alter the values of the "fixed" conformal dimensions of the first 10 operators (even with the relatively large $10^{-3}$ window) and converged on a result with good accuracy showcases that the method is robust and managed to locate the theory. For an example of the variation in the scaling dimensions $\Delta_1$, $\Delta_2$, $\Delta_3$ in the SAC runs, see Table VI below.

(4) SAC and IPOPT are producing configurations of comparable rewards.[11] Qualitatively, the SAC curve in Fig. 1 is smoother compared to the IPOPT curve, but the numbers in Table VII do not declare a clear winner. Besides $C_1^2$, a more specific datum that one can check is the sum of the $C_2^2$ and $C_3^2$ coefficients. Toward the strong coupling region the scaling dimensions of the $J = 2$ operators, $\Delta_2$ and $\Delta_3$, converge toward 4. As a result, the two operators remain nearly degenerate throughout the flow from weak to strong coupling. This feature complicates the search, as was already noted in [1]. In Fig. 4 of [1], narrow bounds were reported at $g = 1$ that place $C_2^2$ and $C_3^2$ on the line $C_3^2 + 1.13C_2^2 = 0.19$. By inserting the average results from the SAC and IPOPT runs into this expression, we find:

$$\begin{aligned} \text{SAC}&: \; C_3^2 + 1.13C_2^2 = 0.19107, \\ \text{IPOPT}&: \; C_3^2 + 1.13C_2^2 = 0.18968. \end{aligned} \tag{39}$$

In the upcoming section, we will see that the most accurate results of [2], based on also using the integral constraints, yield $C_3^2 + 1.13C_2^2 = 0.19171$.

[11]Strictly speaking, for SAC, this is true after running IPOPT around the SAC average configuration. As we discuss in more detail in Sec. VI, this has minuscule effects on the average configuration.

## V. RESULTS WITH INTEGRAL CONSTRAINTS

We proceed to discuss the results obtained by incorporating the two integral constraints (34). Anticipating a more pronounced minimum in this case, we exclusively employed IPOPT. The search parameters closely resembled those used in the IPOPT runs of Sec. V. The integral constraints were supplied as separate equations using the corresponding functionality of PyGMO. Our runs involved $4 \times 10^8$ agents. These runs were subdivided into groups with a population of 100k within the PyGMO architecture. Each group was run 4k times on the QMUL HPC cluster, with an approximate run-time of 20 minutes. Statistics were collected from the 200 runs with the highest reward. We imposed $N_{\text{der}} = 260$ derivative constraints but, unlike the previous section, did not enforce the additional restrictions $C_2^2 < 0.1$ and $C_2^3 > 0.1$. The search windows were set to $\pm 1$ for the scaling dimensions and $\pm 0.2$ for the OPE coefficients, centered around the average solution obtained at the previous value of $g$. To ensure proper results, we incorporated lower bounds. The lower bound for each $J$ family was set to the free-limit value for the scaling dimensions, while the lower bound for the OPE-squared coefficients was set to 0. Additionally, the dimensions of the first 10 long operators were fixed completely according to the results of the QSC.

Our results are plotted in Fig. 2 against the background of the rigorous allowed regions in Fig. 10 of Ref. [2]. A partial list of the specific numbers with statistical errors appears in Table IV and the full list in Table VII of Appendix A.

We observe that the presence of the integral constraints significantly narrows down the statistical errors for the first OPE-coefficient squared and enhances the accuracy of our statistical IPOPT runs, which align closely with the findings of [2]. This alignment is particularly noticeable for lower values of $g$. At higher values of $g$, the IPOPT results for $C_2^2$ and $C_3^2$ become less accurate, while the linear-functional method results become sharper. We believe this is due to the near degeneracy of the corresponding operators, which makes the search more demanding. We observed that by increasing the number of parallel agents, there is improvement in these numbers.

One might wonder whether our approximation scheme, which includes the tail evaluated in the free limit, remains

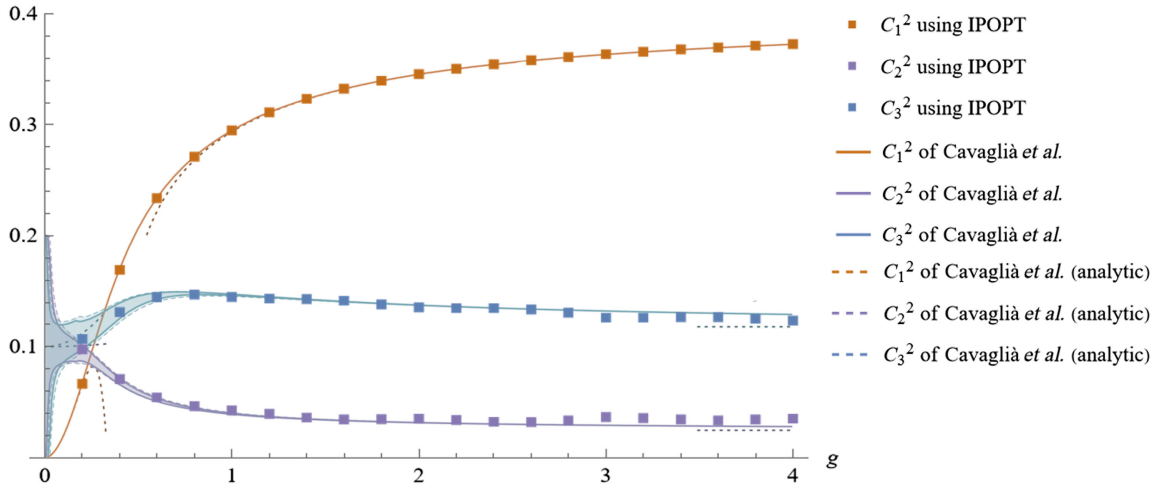


FIG. 2. Results for the OPE-coefficients squared of the first three long operators after the incorporation of two integral constraints. The solid lines indicate the rigorous bounds presented in Fig. 10 of [2], reprinted here with permission from the authors. Same-colored squares indicate our results from the IPOPT runs. The corresponding statistical errors are too small to display on this plot but can be found in Table VII.

valid for all values of $g$. We have checked this point by performing cursory runs, where besides fixing the dimensions of the first 10 long operators to the results of the QSC, we also fixed the values of $C_1^2, C_2^2$ to the values of [2]. IPOPT then recovered, for all $g$, the value of $C_3^2$ in [2] with at least third decimal point accuracy (and often fifth decimal point). We expect that full-fledged statistical runs would improve this even further. This provides favorable evidence that our (soft) tail approximation scheme works well in this specific problem for a large region of parameter space from weak to strong coupling.

Moreover, at $g = 1$, we computed the sum $C_3^2 + 1.13C_2^2$, which is expected to come in at 0.19 from [1], and found:

IPOPT with constraints:

$$C_3^2 + 1.13C_2^2 = 0.19137, \tag{40}$$

Cavaglià *et al.* [2]:

$$C_3^2 + 1.13C_2^2 = 0.19171. \tag{41}$$

We also computed the sum $C_2^2 + C_3^2$ of our IPOPT results for all values of $g$ (up to $g = 4$) and found it to agree with [2] to at least three decimal points. This is further evidence for the validity of the tail approximation.

## VI. ANALYSIS AND DISCUSSION

We would now like to discuss some of the most informative features of the approximate solutions of Secs. IV and V. These properties are not apparent from the table and figure representation of the results for the first three OPE-coefficients squared. First, we will comment on the OPE coefficients of higher excited states predicted by

TABLE IV. Partial list of results (for $g \in [0.2, 1]$) from IPOPT runs with two integral constraints and comparison with [2]. The errors for [2] encode the rigorous upper and lower bounds about the indicated mean. The errors for IPOPT encode one standard deviation around the statistical reward-weighted average.

| Method | $g$ | $C_1^2$ | $C_2^2$ | $C_3^2$ |
|---|---|---|---|---|
| [2] | 0.2 | $0.065679029 \pm 6.95 \times 10^{-7}$ | $0.09452 \pm 7.25 \times 10^{-3}$ | $0.1101 \pm 1.27 \times 10^{-2}$ |
| IPOPT | 0.2 | $0.06567873 \pm 1.55 \times 10^{-7}$ | $0.09683 \pm 1.41 \times 10^{-3}$ | $0.1063 \pm 2.42 \times 10^{-3}$ |
| [2] | 0.4 | $0.16838882 \pm 1.29 \times 10^{-6}$ | $0.06925 \pm 2.80 \times 10^{-3}$ | $0.13196 \pm 7.16 \times 10^{-3}$ |
| IPOPT | 0.4 | $0.16838814 \pm 6.13 \times 10^{-7}$ | $0.07010 \pm 1.06 \times 10^{-3}$ | $0.13026 \pm 2.58 \times 10^{-3}$ |
| [2] | 0.6 | $0.233041731 \pm 4.49 \times 10^{-7}$ | $0.05246 \pm 1.47 \times 10^{-3}$ | $0.14546 \pm 2.99 \times 10^{-3}$ |
| IPOPT | 0.6 | $0.233041064 \pm 8.18 \times 10^{-7}$ | $0.05347 \pm 1.30 \times 10^{-3}$ | $0.14376 \pm 2.37 \times 10^{-3}$ |
| [2] | 0.8 | $0.270286735 \pm 1.32 \times 10^{-7}$ | $0.044285 \pm 7.18 \times 10^{-4}$ | $0.14798 \pm 1.17 \times 10^{-3}$ |
| IPOPT | 0.8 | $0.270286201 \pm 8.53 \times 10^{-7}$ | $0.045597 \pm 1.60 \times 10^{-3}$ | $0.14607 \pm 2.27 \times 10^{-3}$ |
| [2] | 1.0 | $0.294014873 \pm 4.88 \times 10^{-8}$ | $0.039788 \pm 4.10 \times 10^{-4}$ | $0.146757 \pm 5.82 \times 10^{-4}$ |
| IPOPT | 1.0 | $0.294014228 \pm 6.77 \times 10^{-7}$ | $0.041832 \pm 1.86 \times 10^{-3}$ | $0.144100 \pm 2.39 \times 10^{-3}$ |

TABLE V. Preliminary results for the OPE-coefficients squared of the operators with scaling dimensions $\Delta_4$, $\Delta_5$, $\Delta_6$, $\Delta_8$ at $g = 4$ from the searches of Secs. IV and V. The "other" contributions come from effective operators of the $J = 5$ family that have comparable scaling dimensions.

| Method | $C_4^2 \times 10^3$ | $C_5^2 \times 10^3$ | $C_6^2 \times 10^3$ | $C_8^2 \times 10^3$ | other $\times 10^3$ |
|---|---|---|---|---|---|
| IPOPT + cons | $5.57 \pm 5.83$ | $3.74 \pm 2.35$ | $3.76 \pm 2.31$ | $4.12 \pm 4.23$ | $11.18 \pm 15.44$ |
| IPOPT | $3.74 \pm 0.23$ | $3.58 \pm 0.21$ | $3.63 \pm 0.22$ | $3.39 \pm 0.20$ | $17.58 \pm 0.93$ |
| SAC | $7.50 \pm 3.30$ | $3.17 \pm 1.85$ | $3.96 \pm 2.67$ | $2.68 \pm 2.16$ | $10.74 \pm 5.14$ |

our searches. Second, we will compare the performance of the SAC and IPOPT algorithms. In particular, we would like to address the questions:

"What have we learned about nonconvex optimizers in the context of our truncation schemes? Is reinforcement learning a useful tool for future studies?"

### A. Higher CFT data

We remind the reader that, in addition to the first three lowest-lying operators in the $J = 1$ and $J = 2$ families, our searches also had the scaling dimensions of six operators in the $J = 3$ family and the leading operator in the $J = 4$ family fixed using the QSC. These operators acquire anomalous dimensions, and their scaling dimensions can cross with other operators as functions of the coupling. The mixing of contributions from different families in our crossing equations prevents us from extracting clear results for individual operators. However, this mixing is minimal, or altogether absent, for the four lowest-lying operators in the $J = 3$ family; their scaling dimensions (labeled $\Delta_4$, $\Delta_5$, $\Delta_6$, $\Delta_8$ in the language of [2]) are tracked with the QSC. As a preliminary result, we have included in Table VIII of Appendix A the corresponding values of the OPE-coefficients squared $C_4^2$, $C_5^2$, $C_6^2$, $C_8^2$ for $g \in [0.2, 1]$, obtained by independently using SAC without integral constraints, IPOPT without and with integral constraints. These are the same runs already reported with $N_{\text{der}} = 260$.

We observe that the statistical errors are now more significant, which aligns with the observations of [2]. Setting this aside, the values of all three methods are close, giving some confidence that they are in the neighborhood of the exact result. Comparing with the unpublished rigorous bounds of the authors of Ref. [2][12] supports the same conclusion.

It is interesting to ask how our results compare with known expectations at strong coupling. Before delving into the numbers, we must address an issue that affects our data at strong coupling. Throughout the whole range of $g$ values that we explored, both SAC and IPOPT have opted to keep the leading $J = 5$ scaling dimensions close to their weak coupling values. At strong coupling (specifically $g = 4$), this puts the scaling dimensions of some operators in the $J = 5$ family close to the scaling dimensions of the nearly degenerate $J = 3$ operators and obscures the interpretation of our results. This effect is more pronounced in the IPOPT runs. In the SAC runs, only two operators are low enough to be nearly degenerate with the $J = 3$ operators. We expect that this issue can be remedied by using additional information from the QSC for the leading operator in the $J = 5$ family, using an appropriate modification of the method recently developed in [48] along the lines of [1].

In Table V, we present the results of our three runs at $g = 4$. In the column "other," we have included the OPE-coefficient squared contribution of operators in the $J = 5$ family with scaling dimensions close to the $J = 3$ dimensions of interest.[13] For comparison, Ref. [2] reports the upper bounds

$$C_4^2 < 0.0079, \qquad C_5^2 < 0.0123. \tag{42}$$

In addition, Ref. [69] has computed the strong coupling limit of the total OPE-coefficient squared of the four $J = 3$ degenerate operators at $10/429 \simeq 0.023$. Adding up the contributions in Table V, we obtain:

$$\begin{aligned} \text{IPOPT with constraints}&: \ 0.028 \pm 0.030, \\ \text{IPOPT w/o constraints}&: \ 0.032 \pm 0.002, \\ \text{SAC}&: \ 0.028 \pm 0.015. \end{aligned} \tag{43}$$

At the current stage, we do not want to read too much into these numbers, as the statistical errors are significant, but they appear to indicate that our approach is on the right track. We believe that with further improvements, such as fixing the dimension of the first $J = 5$ operator from the

[12]We would like to thank the authors of [1,2] for communication on this point.

[13]At $g = 4$, the nearly degenerate operators of interest in the $J = 3$ family have scaling dimensions $\Delta_4 = 5.504295213$, $\Delta_5 = 5.521481452$, $\Delta_6 = 5.516492991$, $\Delta_8 = 5.539940361$. In SAC, the two interfering $J = 5$ operators come out at $\Delta = 5.226 \pm 0.075$ with $C^2 = 0.00742 \pm 2.88 \times 10^{-3}$ and $\Delta = 5.508 \pm 0.078$ with $C^2 = 0.00332 \pm 2.26 \times 10^{-3}$. The next $J = 5$ operator, which was not included in Table V, has $\Delta = 5.875 \pm 0.072$. For IPOPT, the $J = 5$ effective operators are more densely spread around $\Delta = 5.5$. In "other," we included $J = 5$ operators within the band $\Delta \in [5.25, 5.7]$, which involved six and seven operators for the constrained/unconstrained search, respectively.

QSC, one will eventually be able to obtain more accurate and reliable results for these CFT data as well.

### B. The unreasonable effectiveness of the SAC average

Our results show that the (reward-weighted) average of the SAC runs is particularly accurate and frequently much closer to the actual result (compared to that of the maximum-reward agent in the population). If SAC can efficiently locate a basin of attraction, as already anticipated and partially observed in [23], then perhaps this is a natural expectation. However, whether and how this actually happens is not at all obvious for several reasons. Most notably, unlike other typical (deterministic or stochastic) optimization algorithms, where one observes a high-reward-driven distribution of configurations exploiting the microstructure of the search landscape, in SAC, individual agents are comparatively reward-underachievers. The reward of the average configuration is not remarkable either. Moreover, since we have been running a tiny population of 200 parallel agents—when IPOPT was operated with $4 \times 10^8$ agents—one may also question the quality of the statistics we obtained.

In order to test the quality of the SAC average, and whether SAC was able to identify a genuine basin of attraction, we performed the following exercise at the end of all of our 200-agent SAC runs. We set a search box around the SAC average, $\overline{\text{SAC}}$, with bounds $[\overline{\text{SAC}}_i - 4\sigma_i, \overline{\text{SAC}}_i + 4\sigma_i]$. The index $i$ denotes the $i$th CFT datum and $\sigma_i$ its corresponding $1\sigma$ uncertainty in the SAC runs. Inside this box, we ran $2 \times 10^8$ IPOPT agents (subdivided into 2k groups each with 100k population). We repeated these "IPOPT-on-SAC" runs for all values of $g$. The results at $g = 1$ are plotted in Fig. 3. In Table VI, we present the corresponding values of the reward-weighted averages for the SAC and IPOPT runs with $N_{\text{der}} = 260$, as well as results from IPOPT runs with $N_{\text{der}} = 700$, which are not plotted in Fig. 3. For reference, we also included the QSC values of the scaling dimensions.

For all values of $g$, we observed the following features. First, the algorithms optimize the scaling dimensions in the vicinity of the QSC values. This is a satisfying minimal check of the method against the QSC expectations. Second, increasing the number of derivatives from 260 to 700 in the IPOPT runs does not appear to yield any significant improvements. Third, and most important, the IPOPT averages reproduce consistently and with great accuracy the SAC averages despite the spread that the IPOPT agents exhibit. It is truly striking that the SAC runs with only 200 agents and relatively low reward have managed to capture well a local basin of attraction. For the first operator, we also notice an intriguing feature of Fig. 3: The IPOPT results are arranged linearly along the diagonal on the $(\Delta_1, C_1^2)$ plane. We have observed this

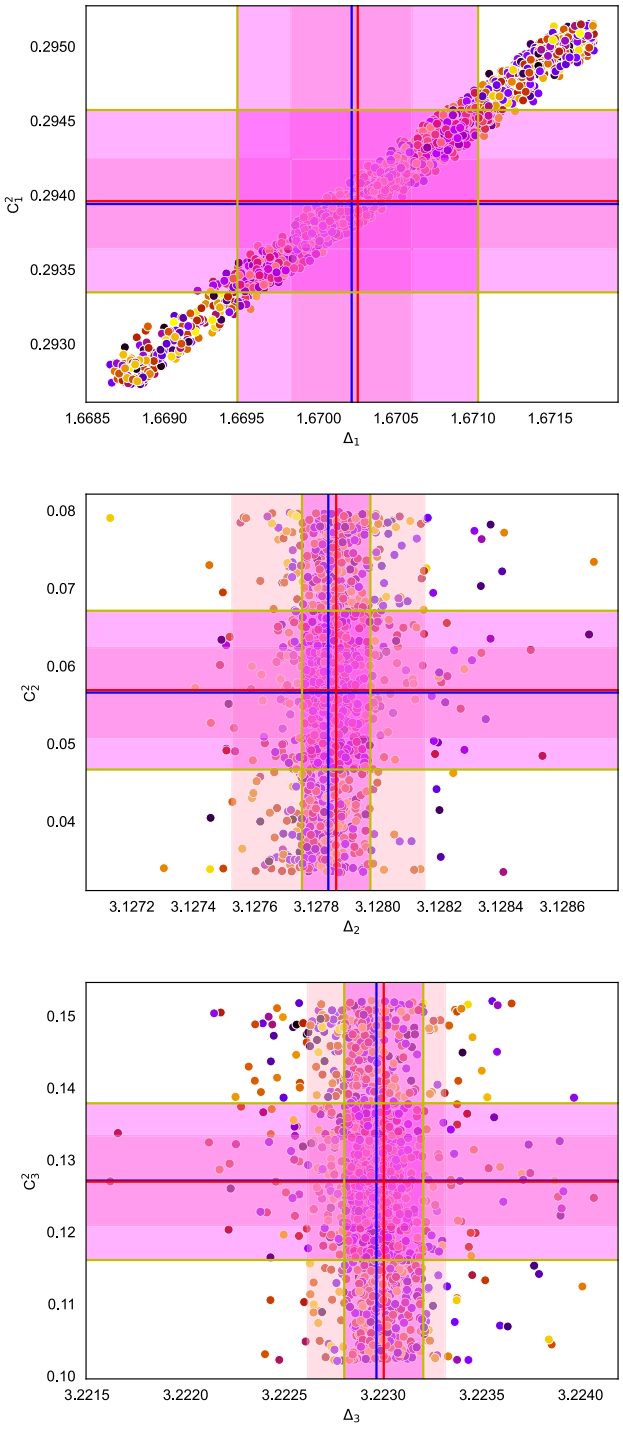


FIG. 3. Plots of SAC and IPOPT results with $N_{\text{der}} = 260$ w/o integral constraints at $g = 1$. For SAC, only the average (blue lines) and the $1\sigma$ regions (pink) appear. For IPOPT, we plot the average (red lines), the $1\sigma$ region (magenta) and the results of the best run for each of the $2k$ jobs on the HPC cluster (points).

configuration at all values of $g$ but do not have a clear explanation for it.

For the $g = 1$ results in Fig. 3 and Table VI, we obtained the following rewards:

$$200\,\mathrm{SAC}_{260}\,\text{agents}:\ \min = 1.76\times10^{4},\quad \text{median} = 4.46\times10^{4},\quad \max = 1.82\times10^{6},$$
$$2\mathrm{k}\,\mathrm{IPOPT}_{260}\,\text{agents}:\ \min = 2.52\times10^{3},\quad \text{median} = 1.56\times10^{6},\quad \max = 2.66\times10^{7},$$
$$2\mathrm{k}\,\mathrm{IPOPT}_{700}\,\text{agents}:\ \min = 1.17\times10^{3},\quad \text{median} = 1.57\times10^{6},\quad \max = 2.22\times10^{7}.$$

We observe similar features at all other values of $g$. Typically, the median and maximum rewards of the IPOPT-on-SAC runs are 2 orders of magnitude larger than those of the SAC runs. The above values of the rewards are also typical for all the IPOPT runs (independent of SAC) at all values of $g$, with or without the integral constraints. In conclusion, we notice that, as a powerful deterministic algorithm, IPOPT gives a visible enhancement of the reward with only small modifications to the average SAC configuration.

### C. Role of effective operators

In Sec. II we introduced and highlighted the significance of effective operators in truncation schemes. The presence (or absence) of higher-dimension effective operators can affect the quality of the results for the low-dimension data, and the freedom to rearrange them at high reward can affect the inherent uncertainties of the search. In that sense, it is not unreasonable to anticipate some correlation between the latter and the size of allowed regions in the linear-functional method. The results in Fig. 1 appear to support this expectation.

In the same context, it is interesting to ask how different algorithms manipulate the effective operators and, correspondingly, how they learn the landscape onto which they are optimizing. In Fig. 4, we present the scaling dimensions of all 62 operators in our truncation as obtained by SAC (dark blue for $g = 0.2$ and light blue for $g = 4$) and IPOPT with the integral constraints (dark red for $g = 0.2$ and light red for $g = 4$). Both plots contain the same information with different orderings of the operators. We observe the following features.

From the top plot in Fig. 4, we notice that both SAC (in blue) and IPOPT (in red) have chosen to minimally vary the scaling dimensions of the $J \geq 5$ families of operators from $g = 0.2$ to $g = 4$. The main variation occurs for the 22 operators of the $J = 4$ family and is more dramatic in the case of IPOPT. There is another significant difference between the two spectra in Fig. 4. IPOPT exhibits a clear tendency (at all values of $g$) to keep operators within the same family nearly degenerate. In this manner, it effectively reduces the number of active operators in the truncation. This was a feature that was also present in other non-SAC algorithms. In contrast, SAC prefers to keep the operators more distinct, effectively smearing them across scaling dimensions, as is apparent in the bottom plot of Fig. 4. It would be interesting to understand why this occurs, whether it is linked to reinforcement learning mechanisms and if it contains some significance that can be used constructively in future searches.

We note in passing that, in addition to the runs reported in this paper, we also performed SAC and IPOPT runs by fixing only nine scaling dimensions that did not include the first of the 22 operators in the $J = 4$ family. In those results, the $22J = 4$ operators received much smaller anomalous dimensions yielding qualitatively worse results both for SAC and IPOPT. It may be useful to explore if exact QSC information for the scaling dimension of the lowest $J = 5$ operator, obtainable from the recent work [48], can similarly lead to further improvements to the results presented here. Naively viewing $J$ as an analog of spin in higher dimensions, it is tempting to consider whether this is analogous to fixing the leading Regge trajectory. That would lead to an interesting hybrid numerical bootstrap analysis in higher-dimensional CFTs, capitalizing on external information from analytics, e.g., from OPE-inversion formulas methods.

TABLE VI. The average and $1\sigma$ values of the plotted CFT data for the first three long operators. The subscripts in SAC and IPOPT denote the number of maximum derivatives used. The row of each table also includes the QSC value of the corresponding scaling dimension.

| Algorithm | $\Delta_1$ | $C_1^2$ |
|---|---|---|
| QSC | 1.670227842 | |
| $\mathrm{IPOPT}_{260}$ | $1.6702536 \pm 7.80\times10^{-4}$ | $0.2939600 \pm 6.12\times10^{-4}$ |
| $\mathrm{IPOPT}_{700}$ | $1.6702387 \pm 8.05\times10^{-4}$ | $0.2939429 \pm 6.34\times10^{-4}$ |
| $\mathrm{SAC}_{260}$ | $1.6702139 \pm 3.92\times10^{-4}$ | $0.2939411 \pm 3.03\times10^{-4}$ |

| Algorithm | $\Delta_2$ | $C_2^2$ |
|---|---|---|
| QSC | 3.127846278 | |
| $\mathrm{IPOPT}_{260}$ | $3.1278644 \pm 1.11\times10^{-4}$ | $0.0569002 \pm 1.02\times10^{-2}$ |
| $\mathrm{IPOPT}_{700}$ | $3.1278716 \pm 1.41\times10^{-4}$ | $0.0582812 \pm 1.00\times10^{-2}$ |
| $\mathrm{SAC}_{260}$ | $3.1278389 \pm 3.13\times10^{-4}$ | $0.0565833 \pm 5.81\times10^{-3}$ |

| Algorithm | $\Delta_3$ | $C_3^2$ |
|---|---|---|
| QSC | 3.222893829 | |
| $\mathrm{IPOPT}_{260}$ | $3.2230032 \pm 2.00\times10^{-4}$ | $0.1269864 \pm 1.09\times10^{-2}$ |
| $\mathrm{IPOPT}_{700}$ | $3.2229707 \pm 2.86\times10^{-4}$ | $0.1254095 \pm 1.05\times10^{-2}$ |
| $\mathrm{SAC}_{260}$ | $3.2229662 \pm 3.49\times10^{-4}$ | $0.1271357 \pm 6.26\times10^{-3}$ |

### D. On the choice of optimization algorithms

Over the course of this work we compared the performance of several different algorithms—some deterministic, some stochastic. We observed that an algorithm like IPOPT was very efficient and could be deployed with $4\times10^{8}$

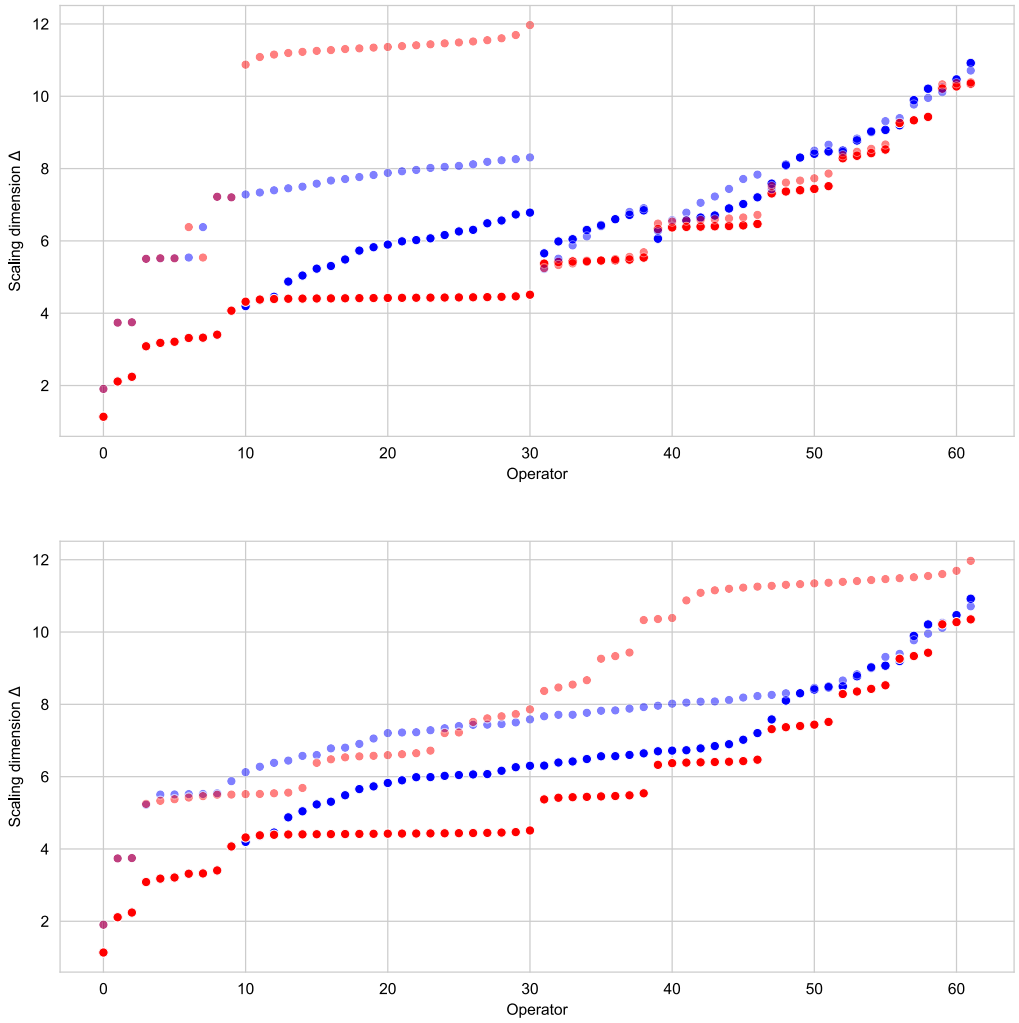


FIG. 4. The scaling dimensions of the operators in our truncation at $g = 0.2$ and $g = 4$. On the $x$ axis, different integer values parametrize different operators. In the upper plot, the operators are ordered separately within their respective $J$ family. In the lower plot, the operators are ordered globally in ascending scaling dimension. The blue dots denote SAC results: dark blue for $g = 0.2$ and light blue for $g = 4$. The red dots denote IPOPT results with integral constraints: dark red for $g = 0.2$ and light red for $g = 4$. We did not include the results of IPOPT without integral constraints, as they are very similar to the data including the constraints. The results of both SAC and IPOPT for the first 10 operators overlap.

agents, producing very quickly (within minutes per 100k population grouping) accurate results. In contrast, SAC produced less accurate results with lower rewards in runs that typically took 12 to 24 hours per agent, depending on the chosen scheduling for the quenching of the search-window sizes. One could therefore ask whether SAC is a useful nonconvex optimization tool within the conformal bootstrap and whether there is sufficient motivation to explore even more advanced machine learning or reinforcement learning algorithms in this context. Based on our current understanding, we would like to argue in the affirmative for the following reasons.

In a typical study of a (truncated) crossing symmetry condition, one formulates a corresponding cost function that is then minimized by varying scaling dimensions and OPE-coefficients squared. The complicated landscape of minima may contain multiple local minimum configurations, some of which correspond to different physical theories. Therefore, when optimizing, one is not necessarily looking for the global minimum. Moreover, in attempting to discover a specific local minimum, one needs the ability to perform a local search without having specific information about the bounds of the search. As a result, typical optimization algorithms that are efficient in global searches or are efficient in local searches but require specific bounds, like interior point methods, stochastic gradient descent and others, are not in general appropriate tools. This feature was not particularly pronounced in the 1D problem that we analyzed in this paper (at least for some CFT data), but it is generally an important issue. For example, in the context of the 6D (2,0) superconformal bootstrap that was analyzed in [23], there are at least two minima of physical interest

corresponding to the $A$ and $D$ series CFTs. In that case, setting up a global search (or a local search with prescribed bounds) is not an efficient approach. On the contrary, Markov-chain algorithms can be a useful alternative, because one can drop agents within a region of interest, and the algorithm will dynamically decide how to explore the local search landscape. SAC (and other similar stochastic algorithms) play this game especially well. SAC is particularly appealing within this class, because it is built upon a process of policy optimization, which adapts its dynamics on the given landscape. Indeed, we have already highlighted several features within this section that support the use of algorithms like SAC. These features include the effectiveness of the average, the qualitative features of the solution and the smooth evolution across parameter space. The collection of these observations also motivates looking for improvements to our reinforcement learning approach, ranging from refining our SAC implementation, as well upgrading the algorithm itself by deploying Constrained and/or multiagent reinforcement learning algorithms, where the agents collaborate to achieve higher rewards.

Irrespective of the above debate, we believe that the present work, and the inherent complexity of the physical problem we are trying to solve, strongly suggest building an arsenal of diverse algorithms. These algorithms can possess traits, which can be used either separately or in combination, to learn a particular problem from qualitatively different perspectives. It is not necessarily desirable to isolate a single algorithm based on a sole feature like speed of performance or quality of the reward. This is perhaps a philosophy that departs from the standard perspective in many optimization problems. We have updated BootSTOP to make more algorithms directly accessible to the bootstrap practitioner in service to this philosophy.

## VII. OUTLOOK

In this paper, we studied the four-point function bootstrap in the 1D CFT of the $\frac{1}{2}$-BPS straight Maldacena-Wilson line in 4D $\mathcal{N}=4$ SYM theory. Importing information for the scaling dimensions of 10 long operators from the quantum spectral curve [70,71], we analyzed the crossing symmetry conditions with or without the inclusion of two additional sum rules arising from integrated correlation functions [5,6]. Unlike Refs. [1,2], which employed linear-functional methods, we introduced an improved truncation scheme with a tail approximation that requires the solution of a nonlinear, nonconvex optimization problem. We attacked this problem using stochastic (reinforcement learning) and deterministic (interior point method) algorithms, producing numerical results for three (nonprotected) OPE-coefficients squared that aligned very close to the rigorous results of [1,2]. The main novelties and contributions of this paper are summarized at the end of Sec. I. Our method is relatively computationally cheap; it does not rely on positivity constraints and offers many future opportunities when combined with other analytic and rigorous numerical methods. Our Python implementation, BootSTOP, now contains a library of pregenerated conformal blocks in diverse spacetime dimensions and, besides SAC, also direct access to all the deterministic and stochastic algorithms within PyGMO [28].

One can envision further applications in several different directions. It would be interesting to further explore the 1D CFT of the $\frac{1}{2}$-BPS Maldacena–Wilson line aiming at the computation of more CFT data. We expect that the judicious use of additional input from the QSC and the simultaneous study of multiple correlators can lead to significant progress. Analogous progress may be possible also in other higher-dimensional CFTs. It would be interesting to revisit the 6D $\mathcal{N}=(2,0)$ bootstrap [23,72–74] and 4D $\mathcal{N}=4$ superconformal bootstrap [75–77] using improved truncation schemes and to explore in that context the improvements in accuracy and rigor that can arise with the combination of other numerical methods (e.g., navigator methods) and analytic methods (e.g., methods based on the light-cone bootstrap and the OPE-inversion formula).

## ACKNOWLEDGMENTS

We would like to thank Andreas Stergiou for suggesting IPOPT and the PyGMO package, and for participation at the early stages of this project. We would also like to thank Alexandros Stratoudakis for collaboration on related work. We are grateful to the authors of [1,2], and especially Nikolay Gromov, for communication and giving us access to unpublished results. C. P. and P. R. were supported by the Science and Technology Facilities Council (STFC) Consolidated Grant ST/T000686/1 "Amplitudes, strings & duality." A. G. S. acknowledges support from Pierre Andurand. M. W. is supported by an STFC Research Studentship. This research utilized the Apocrita HPC facility, supported by QMUL Research-IT [78].

## APPENDIX A: EXPLICIT NUMERICAL RESULTS

In this appendix, we list our complete results for runs without integral constraints (both SAC and IPOPT), as well as for IPOPT runs with both integral constraints implemented. In Table VII, we list the values for the OPE-coefficients squared of the first three unprotected operators from the $J=1,2$ families, $C_1^2, C_2^2, C_3^2$, when $g\in[0.2,4]$ along with the results of [1,2] for reference.

In Table VIII, we list the results of the OPE-coefficients squared for the next four unprotected operators from the $J=3$ family, $C_4^2, C_5^2, C_6^2, C_8^2$ when $g\in[0.2,1]$. The unpublished rigorous bounds of the authors of [2] support the expectation that these results are on the right track. The operators with scaling dimensions $\Delta_7$, $\Delta_9$ acquire large anomalous dimensions at strong coupling and are no longer almost degenerate with $\Delta_4$, $\Delta_5$, $\Delta_6$, $\Delta_8$. The OPE-coefficient squared values for the full list of

TABLE VII. Explicit numerical results for OPE-coefficients squared $C_1^2, C_2^2, C_3^2$ from: (a) [2], (b) IPOPT with two integral constraints imposed (c) [1], (d) IPOPT with no integral constraints imposed, and (e) SAC with no integral constraints imposed. The errors for (a) and (c) encode the rigorous upper and lower bounds about the indicated mean. For (b), (d), (e), the errors encode one standard deviation from the statistical average.

| Method | $g$ | $C_1^2$ | $C_2^2$ | $C_3^2$ |
|---|---|---|---|---|
| [2] | 0.2 | $0.06567902 \pm 6.95 \times 10^{-7}$ | $0.09452 \pm 7.25 \times 10^{-3}$ | $0.1101 \pm 1.27 \times 10^{-2}$ |
| IPOPT w/ cons | 0.2 | $0.06567873 \pm 1.55 \times 10^{-7}$ | $0.09683 \pm 1.41 \times 10^{-3}$ | $0.1063 \pm 2.42 \times 10^{-3}$ |
| [1] | 0.2 | $0.0663 \pm 1.9 \times 10^{-3}$ | | |
| IPOPT w/o cons | 0.2 | $0.06607342 \pm 4.18 \times 10^{-5}$ | $0.04708 \pm 2.04 \times 10^{-3}$ | $0.1630 \pm 2.69 \times 10^{-3}$ |
| SAC w/o cons | 0.2 | $0.06733947 \pm 1.26 \times 10^{-3}$ | $0.06506 \pm 1.05 \times 10^{-2}$ | $0.1384 \pm 1.47 \times 10^{-2}$ |
| [2] | 0.4 | $0.16838882 \pm 1.29 \times 10^{-6}$ | $0.06925 \pm 2.80 \times 10^{-3}$ | $0.13196 \pm 7.16 \times 10^{-3}$ |
| IPOPT w/ cons | 0.4 | $0.16838814 \pm 6.13 \times 10^{-7}$ | $0.07010 \pm 1.06 \times 10^{-3}$ | $0.13026 \pm 2.58 \times 10^{-3}$ |
| [1] | 0.4 | $0.1684 \pm 1.9 \times 10^{-3}$ | | |
| IPOPT w/o cons | 0.4 | $0.16944584 \pm 8.35 \times 10^{-5}$ | $0.02659 \pm 3.45 \times 10^{-3}$ | $0.17965 \pm 4.90 \times 10^{-3}$ |
| SAC w/o cons | 0.4 | $0.16824002 \pm 1.00 \times 10^{-3}$ | $0.06380 \pm 1.37 \times 10^{-2}$ | $0.14198 \pm 1.80 \times 10^{-2}$ |
| [2] | 0.6 | $0.233041731 \pm 4.49 \times 10^{-7}$ | $0.05246 \pm 1.47 \times 10^{-3}$ | $0.14546 \pm 2.99 \times 10^{-3}$ |
| IPOPT w/ cons | 0.6 | $0.233041064 \pm 8.18 \times 10^{-7}$ | $0.05347 \pm 1.30 \times 10^{-3}$ | $0.14376 \pm 2.37 \times 10^{-3}$ |
| [1] | 0.6 | $0.2329 \pm 9 \times 10^{-4}$ | | |
| IPOPT w/o cons | 0.6 | $0.233574606 \pm 1.32 \times 10^{-4}$ | $0.02533 \pm 6.78 \times 10^{-3}$ | $0.17382 \pm 7.68 \times 10^{-3}$ |
| SAC w/o cons | 0.6 | $0.232721152 \pm 3.24 \times 10^{-4}$ | $0.06151 \pm 5.46 \times 10^{-3}$ | $0.13363 \pm 6.77 \times 10^{-3}$ |
| [2] | 0.8 | $0.270286735 \pm 1.32 \times 10^{-7}$ | $0.044285 \pm 7.18 \times 10^{-4}$ | $0.14798 \pm 1.17 \times 10^{-3}$ |
| IPOPT w/ cons | 0.8 | $0.270286201 \pm 8.53 \times 10^{-7}$ | $0.045597 \pm 1.60 \times 10^{-3}$ | $0.14607 \pm 2.27 \times 10^{-3}$ |
| [1] | 0.8 | $0.2701 \pm 5 \times 10^{-4}$ | | |
| IPOPT w/o cons | 0.8 | $0.270632286 \pm 6.67 \times 10^{-5}$ | $0.020165 \pm 6.29 \times 10^{-3}$ | $0.17218 \pm 7.06 \times 10^{-3}$ |
| SAC w/o cons | 0.8 | $0.270121362 \pm 2.93 \times 10^{-4}$ | $0.05776 \pm 5.00 \times 10^{-3}$ | $0.13110 \pm 5.35 \times 10^{-3}$ |
| [2] | 1.0 | $0.294014873 \pm 4.88 \times 10^{-8}$ | $0.039788 \pm 4.10 \times 10^{-4}$ | $0.146757 \pm 5.82 \times 10^{-4}$ |
| IPOPT w/ cons | 1.0 | $0.294014228 \pm 6.77 \times 10^{-7}$ | $0.041832 \pm 1.86 \times 10^{-3}$ | $0.144100 \pm 2.39 \times 10^{-3}$ |
| [1] | 1.0 | $0.29388 \pm 2.7 \times 10^{-4}$ | | |
| IPOPT w/o cons | 1.0 | $0.294177967 \pm 6.79 \times 10^{-5}$ | $0.023344 \pm 9.64 \times 10^{-3}$ | $0.163302 \pm 1.04 \times 10^{-2}$ |
| SAC w/o cons | 1.0 | $0.293941106 \pm 3.03 \times 10^{-4}$ | $0.05658 \pm 5.81 \times 10^{-3}$ | $0.127135 \pm 6.26 \times 10^{-3}$ |
| [2] | 1.2 | $0.310433307 \pm 2.16 \times 10^{-8}$ | $0.036979 \pm 2.62 \times 10^{-4}$ | $0.144696 \pm 3.40 \times 10^{-4}$ |
| IPOPT w/ cons | 1.2 | $0.310433 \pm 3.07 \times 10^{-7}$ | $0.038659 \pm 1.19 \times 10^{-3}$ | $0.142616 \pm 1.44 \times 10^{-3}$ |
| [1] | 1.2 | $0.31033 \pm 1.7 \times 10^{-4}$ | | |
| IPOPT w/o cons | 1.2 | $0.31050381 \pm 4.45 \times 10^{-5}$ | $0.026543 \pm 1.04 \times 10^{-2}$ | $0.155286 \pm 1.14 \times 10^{-2}$ |
| SAC w/o cons | 1.2 | $0.310378551 \pm 2.11 \times 10^{-4}$ | $0.055333 \pm 4.36 \times 10^{-3}$ | $0.124209 \pm 4.74 \times 10^{-3}$ |
| [2] | 1.4 | $0.322466863 \pm 1.08 \times 10^{-8}$ | $0.035063 \pm 1.79 \times 10^{-4}$ | $0.142594 \pm 2.20 \times 10^{-4}$ |
| IPOPT w/ cons | 1.4 | $0.322466925 \pm 1.47 \times 10^{-7}$ | $0.035350 \pm 7.74 \times 10^{-4}$ | $0.142205 \pm 8.95 \times 10^{-4}$ |
| [1] | 1.4 | $0.32239 \pm 1.2 \times 10^{-4}$ | | |
| IPOPT w/o cons | 1.4 | $0.322494228 \pm 1.64 \times 10^{-5}$ | $0.026081 \pm 5.01 \times 10^{-3}$ | $0.152136 \pm 5.33 \times 10^{-3}$ |
| SAC w/o cons | 1.4 | $0.322459863 \pm 2.73 \times 10^{-4}$ | $0.052775 \pm 6.01 \times 10^{-3}$ | $0.123285 \pm 5.96 \times 10^{-3}$ |
| [2] | 1.6 | $0.331663291 \pm 5.97 \times 10^{-9}$ | $0.033675 \pm 1.30 \times 10^{-4}$ | $0.140664 \pm 1.52 \times 10^{-4}$ |
| IPOPT w/ cons | 1.6 | $0.331663391 \pm 1.42 \times 10^{-7}$ | $0.033591 \pm 9.48 \times 10^{-4}$ | $0.140715 \pm 1.06 \times 10^{-3}$ |
| [1] | 1.6 | $0.33160 \pm 9 \times 10^{-5}$ | | |
| IPOPT w/o cons | 1.6 | $0.331696148 \pm 7.15 \times 10^{-6}$ | $0.017362 \pm 2.96 \times 10^{-3}$ | $0.157958 \pm 3.13 \times 10^{-3}$ |
| SAC w/o cons | 1.6 | $0.331515801 \pm 2.68 \times 10^{-4}$ | $0.051282 \pm 3.94 \times 10^{-3}$ | $0.122229 \pm 3.99 \times 10^{-3}$ |
| [2] | 1.8 | $0.338918478 \pm 3.53 \times 10^{-9}$ | $0.0326214 \pm 9.75 \times 10^{-5}$ | $0.138948 \pm 1.11 \times 10^{-4}$ |
| IPOPT w/ cons | 1.8 | $0.338918374 \pm 1.22 \times 10^{-7}$ | $0.0339949 \pm 1.05 \times 10^{-3}$ | $0.137417 \pm 1.15 \times 10^{-3}$ |
| [1] | 1.8 | $0.33887 \pm 6 \times 10^{-5}$ | | |
| IPOPT w/o cons | 1.8 | $0.338952186 \pm 1.03 \times 10^{-5}$ | $0.0133279 \pm 4.15 \times 10^{-3}$ | $0.159124 \pm 4.30 \times 10^{-3}$ |
| SAC w/o cons | 1.8 | $0.338753189 \pm 1.99 \times 10^{-4}$ | $0.0496961 \pm 3.67 \times 10^{-3}$ | $0.121366 \pm 3.98 \times 10^{-3}$ |

*(Table continued)*

TABLE VII. *(Continued)*

| Method | $g$ | $C_1^2$ | $C_2^2$ | $C_3^2$ |
|---|---|---|---|---|
| [2] | 2.0 | $0.344787161 \pm 2.21 \times 10^{-9}$ | $0.0317952 \pm 7.56 \times 10^{-5}$ | $0.1374382 \pm 8.44 \times 10^{-5}$ |
| IPOPT w/ cons | 2.0 | $0.344786958 \pm 1.12 \times 10^{-7}$ | $0.0343623 \pm 1.22 \times 10^{-3}$ | $0.1346474 \pm 1.32 \times 10^{-3}$ |
| [1] | 2.0 | $0.34475 \pm 5 \times 10^{-5}$ | | |
| IPOPT w/o cons | 2.0 | $0.344812637 \pm 6.30 \times 10^{-6}$ | $0.0154283 \pm 4.13 \times 10^{-3}$ | $0.1543773 \pm 4.30 \times 10^{-3}$ |
| SAC w/o cons | 2.0 | $0.344739867 \pm 1.67 \times 10^{-4}$ | $0.0484264 \pm 3.37 \times 10^{-3}$ | $0.1202422 \pm 3.48 \times 10^{-3}$ |
| [2] | 2.2 | $0.349631253 \pm 1.45 \times 10^{-9}$ | $0.0311296 \pm 6.02 \times 10^{-5}$ | $0.1361104 \pm 6.60 \times 10^{-5}$ |
| IPOPT w/ cons | 2.2 | $0.34963113 \pm 1.16 \times 10^{-7}$ | $0.0332158 \pm 1.55 \times 10^{-3}$ | $0.1338673 \pm 1.65 \times 10^{-3}$ |
| [1] | 2.2 | $0.34960 \pm 4 \times 10^{-5}$ | | |
| IPOPT w/o cons | 2.2 | $0.349651805 \pm 4.90 \times 10^{-6}$ | $0.0147950 \pm 2.78 \times 10^{-3}$ | $0.1529520 \pm 2.84 \times 10^{-3}$ |
| SAC w/o cons | 2.2 | $0.349528529 \pm 1.69 \times 10^{-4}$ | $0.04762628 \pm 3.27 \times 10^{-3}$ | $0.1193224 \pm 3.36 \times 10^{-3}$ |
| [2] | 2.4 | $0.353696925 \pm 9.9 \times 10^{-10}$ | $0.0305818 \pm 4.90 \times 10^{-5}$ | $0.1349397 \pm 5.30 \times 10^{-5}$ |
| IPOPT w/ cons | 2.4 | $0.353696905 \pm 1.32 \times 10^{-7}$ | $0.0314842 \pm 2.08 \times 10^{-3}$ | $0.1339678 \pm 2.21 \times 10^{-3}$ |
| [1] | 2.4 | $0.353669 \pm 3.2 \times 10^{-5}$ | | |
| IPOPT w/o cons | 2.4 | $0.353713043 \pm 9.54 \times 10^{-6}$ | $0.0184944 \pm 4.62 \times 10^{-3}$ | $0.1472742 \pm 4.66 \times 10^{-3}$ |
| SAC w/o cons | 2.4 | $0.353608797 \pm 1.34 \times 10^{-4}$ | $0.0452261 \pm 3.98 \times 10^{-3}$ | $0.1201249 \pm 4.05 \times 10^{-3}$ |
| [2] | 2.6 | $0.357157434 \pm 7.0 \times 10^{-10}$ | $0.0301230 \pm 4.06 \times 10^{-5}$ | $0.1339028 \pm 4.34 \times 10^{-5}$ |
| IPOPT w/ cons | 2.6 | $0.357157405 \pm 1.57 \times 10^{-7}$ | $0.0312744 \pm 2.82 \times 10^{-3}$ | $0.1326779 \pm 2.97 \times 10^{-3}$ |
| [1] | 2.6 | $0.357134 \pm 2.7 \times 10^{-5}$ | | |
| IPOPT w/o cons | 2.6 | $0.357170084 \pm 6.24 \times 10^{-6}$ | $0.0187927 \pm 3.21 \times 10^{-3}$ | $0.1454316 \pm 3.27 \times 10^{-3}$ |
| SAC w/o cons | 2.6 | $0.357024241 \pm 1.56 \times 10^{-4}$ | $0.0453255 \pm 2.97 \times 10^{-3}$ | $0.1187001 \pm 2.93 \times 10^{-3}$ |
| [2] | 2.8 | $0.360138240 \pm 5.0 \times 10^{-10}$ | $0.0297329 \pm 3.42 \times 10^{-5}$ | $0.1329800 \pm 3.63 \times 10^{-5}$ |
| IPOPT w/ cons | 2.8 | $0.360138136 \pm 1.18 \times 10^{-7}$ | $0.0327207 \pm 2.68 \times 10^{-3}$ | $0.1298450 \pm 2.80 \times 10^{-3}$ |
| [1] | 2.8 | $0.360118 \pm 2.2 \times 10^{-5}$ | | |
| IPOPT w/o cons | 2.8 | $0.360146985 \pm 3.74 \times 10^{-6}$ | $0.0190562 \pm 3.74 \times 10^{-6}$ | $0.1438591 \pm 5.46 \times 10^{-3}$ |
| SAC w/o cons | 2.8 | $0.360062267 \pm 1.26 \times 10^{-4}$ | $0.0445569 \pm 3.08 \times 10^{-3}$ | $0.1180638 \pm 3.08 \times 10^{-3}$ |
| [2] | 3.0 | $0.362732415 \pm 3.7 \times 10^{-10}$ | $0.0293973 \pm 2.92 \times 10^{-5}$ | $0.1321546 \pm 3.07 \times 10^{-5}$ |
| IPOPT w/ cons | 3.0 | $0.362732181 \pm 8.27 \times 10^{-8}$ | $0.0358479 \pm 2.28 \times 10^{-3}$ | $0.1254397 \pm 2.38 \times 10^{-3}$ |
| [1] | 3.0 | $0.362715 \pm 1.9 \times 10^{-5}$ | | |
| IPOPT w/o cons | 3.0 | $0.362739018 \pm 1.46 \times 10^{-6}$ | $0.0237969 \pm 2.40 \times 10^{-3}$ | $0.1377698 \pm 2.45 \times 10^{-3}$ |
| SAC w/o cons | 3.0 | $0.362574889 \pm 1.71 \times 10^{-4}$ | $0.0436489 \pm 2.77 \times 10^{-3}$ | $0.1180333 \pm 2.88 \times 10^{-3}$ |
| [2] | 3.2 | $0.365010449 \pm 2.8 \times 10^{-10}$ | $0.0291054 \pm 2.52 \times 10^{-5}$ | $0.1314126 \pm 2.64 \times 10^{-5}$ |
| IPOPT w/ cons | 3.2 | $0.365010269 \pm 1.11 \times 10^{-7}$ | $0.0348757 \pm 3.37 \times 10^{-3}$ | $0.1254316 \pm 3.49 \times 10^{-3}$ |
| [1] | 3.2 | $0.364995 \pm 1.6 \times 10^{-5}$ | | |
| IPOPT w/o cons | 3.2 | $0.365018232 \pm 2.83 \times 10^{-6}$ | $0.0206917 \pm 4.75 \times 10^{-3}$ | $0.1398753 \pm 4.85 \times 10^{-3}$ |
| SAC w/o cons | 3.2 | $0.364922694 \pm 1.43 \times 10^{-4}$ | $0.0425634 \pm 2.43 \times 10^{-3}$ | $0.1179391 \pm 2.43 \times 10^{-3}$ |
| [2] | 3.4 | $0.367026704 \pm 2.2 \times 10^{-10}$ | $0.0288492 \pm 2.20 \times 10^{-5}$ | $0.1307425 \pm 2.30 \times 10^{-5}$ |
| IPOPT w/ cons | 3.4 | $0.367026579 \pm 1.47 \times 10^{-7}$ | $0.0335769 \pm 4.71 \times 10^{-3}$ | $0.1258575 \pm 4.86 \times 10^{-3}$ |
| [1] | 3.4 | $0.367013 \pm 1.4 \times 10^{-5}$ | | |
| IPOPT w/o cons | 3.4 | $0.36703306 \pm 2.72 \times 10^{-6}$ | $0.0221385 \pm 1.63 \times 10^{-3}$ | $0.1374627 \pm 1.65 \times 10^{-3}$ |
| SAC w/o cons | 3.4 | $0.366989521 \pm 1.28 \times 10^{-4}$ | $0.0413939 \pm 2.98 \times 10^{-3}$ | $0.1180792 \pm 2.99 \times 10^{-3}$ |
| [2] | 3.6 | $0.368823769 \pm 1.7 \times 10^{-10}$ | $0.0286224 \pm 1.94 \times 10^{-5}$ | $0.1301347 \pm 2.02 \times 10^{-5}$ |
| IPOPT w/ cons | 3.6 | $0.368823674 \pm 2.03 \times 10^{-7}$ | $0.0327516 \pm 7.18 \times 10^{-3}$ | $0.1258821 \pm 7.38 \times 10^{-3}$ |
| [1] | 3.6 | $0.368812 \pm 1.2 \times 10^{-5}$ | | |
| IPOPT w/o cons | 3.6 | $0.368834396 \pm 7.83 \times 10^{-6}$ | $0.0193471 \pm 4.10 \times 10^{-3}$ | $0.1393921 \pm 4.18 \times 10^{-3}$ |
| SAC w/o cons | 3.6 | $0.368792133 \pm 1.50 \times 10^{-4}$ | $0.0411893 \pm 3.09 \times 10^{-3}$ | $0.1173746 \pm 3.11 \times 10^{-3}$ |
| [2] | 3.8 | $0.370435484 \pm 1.4 \times 10^{-10}$ | $0.0284203 \pm 1.73 \times 10^{-5}$ | $0.1295811 \pm 1.79 \times 10^{-5}$ |
| IPOPT w/ cons | 3.8 | $0.370435381 \pm 2.20 \times 10^{-7}$ | $0.0335783 \pm 8.74 \times 10^{-3}$ | $0.1242843 \pm 8.96 \times 10^{-3}$ |

*(Table continued)*

TABLE VII. *(Continued)*

| Method | $g$ | $C_1^2$ | $C_2^2$ | $C_3^2$ |
|---|---|---|---|---|
| [1] | 3.8 | $0.370425 \pm 1.1 \times 10^{-5}$ | | |
| IPOPT w/o cons | 3.8 | $0.370445095 \pm 7.19 \times 10^{-6}$ | $0.0180924 \pm 5.66 \times 10^{-3}$ | $0.1399174 \pm 5.75 \times 10^{-3}$ |
| SAC w/o cons | 3.8 | $0.370403920 \pm 1.67 \times 10^{-4}$ | $0.0411710 \pm 2.97 \times 10^{-3}$ | $0.1165761 \pm 2.99 \times 10^{-3}$ |
| [2] | 4.0 | $0.371889072 \pm 1.1 \times 10^{-10}$ | $0.0282390 \pm 1.55 \times 10^{-5}$ | $0.1290748 \pm 1.60 \times 10^{-5}$ |
| IPOPT w/ cons | 4.0 | $0.371888949 \pm 2.07 \times 10^{-7}$ | $0.0343581 \pm 8.98 \times 10^{-3}$ | $0.1228095 \pm 9.18 \times 10^{-3}$ |
| [1] | 4.0 | $0.371880 \pm 1 \times 10^{-5}$ | | |
| IPOPT w/o cons | 4.0 | $0.371890446 \pm 5.79 \times 10^{-6}$ | $0.0230225 \pm 8.59 \times 10^{-4}$ | $0.1343690 \pm 8.23 \times 10^{-4}$ |
| SAC w/o cons | 4.0 | $0.371940952 \pm 2.1 \times 10^{-4}$ | $0.0404640 \pm 2.60 \times 10^{-3}$ | $0.1162919 \pm 2.86 \times 10^{-3}$ |

TABLE VIII. Explicit numerical results for OPE-coefficients squared $C_4^2, C_5^2, C_6^2, C_8^2$ from: (a) IPOPT with two integral constraints imposed, (b) IPOPT with no integral constraints imposed, and (c) SAC with no integral constraints imposed. The errors encode one standard deviation from the statistical average.

| Method | $g$ | $C_4^2$ | $C_5^2$ | $C_6^2$ | $C_8^2$ |
|---|---|---|---|---|---|
| IPOPT w/ cons | 0.2 | $0.04110 \pm 1.17 \times 10^{-2}$ | $0.02602 \pm 7.65 \times 10^{-3}$ | $0.02292 \pm 9.63 \times 10^{-3}$ | $0.01618 \pm 5.63 \times 10^{-3}$ |
| IPOPT w/o cons | 0.2 | $0.03257 \pm 3.60 \times 10^{-3}$ | $0.02464 \pm 2.03 \times 10^{-3}$ | $0.02251 \pm 1.91 \times 10^{-3}$ | $0.01609 \pm 1.19 \times 10^{-3}$ |
| SAC w/o cons | 0.2 | $0.02186 \pm 2.17 \times 10^{-2}$ | $0.03303 \pm 2.77 \times 10^{-2}$ | $0.01103 \pm 1.26 \times 10^{-2}$ | $0.02534 \pm 1.41 \times 10^{-2}$ |
| IPOPT w/ cons | 0.4 | $0.02596 \pm 1.04 \times 10^{-2}$ | $0.02397 \pm 1.19 \times 10^{-2}$ | $0.02249 \pm 1.11 \times 10^{-2}$ | $0.01203 \pm 6.46 \times 10^{-3}$ |
| IPOPT w/o cons | 0.4 | $0.03601 \pm 5.49 \times 10^{-3}$ | $0.01670 \pm 2.00 \times 10^{-3}$ | $0.01478 \pm 1.99 \times 10^{-3}$ | $0.00883 \pm 1.59 \times 10^{-3}$ |
| SAC w/o cons | 0.4 | $0.02191 \pm 1.65 \times 10^{-2}$ | $0.01655 \pm 1.24 \times 10^{-2}$ | $0.01546 \pm 1.35 \times 10^{-2}$ | $0.01987 \pm 1.73 \times 10^{-2}$ |
| IPOPT w/ cons | 0.6 | $0.01376 \pm 8.94 \times 10^{-3}$ | $0.02165 \pm 1.35 \times 10^{-2}$ | $0.01875 \pm 1.09 \times 10^{-2}$ | $0.01365 \pm 6.03 \times 10^{-3}$ |
| IPOPT w/o cons | 0.6 | $0.02907 \pm 5.97 \times 10^{-3}$ | $0.01232 \pm 1.93 \times 10^{-3}$ | $0.01143 \pm 1.94 \times 10^{-3}$ | $0.00769 \pm 2.30 \times 10^{-3}$ |
| SAC w/o cons | 0.6 | $0.02211 \pm 9.74 \times 10^{-3}$ | $0.01587 \pm 7.10 \times 10^{-3}$ | $0.01923 \pm 1.08 \times 10^{-2}$ | $0.00809 \pm 8.86 \times 10^{-3}$ |
| IPOPT w/ cons | 0.8 | $0.01083 \pm 6.21 \times 10^{-3}$ | $0.01914 \pm 1.00 \times 10^{-2}$ | $0.01228 \pm 6.44 \times 10^{-3}$ | $0.01735 \pm 8.83 \times 10^{-3}$ |
| IPOPT w/o cons | 0.8 | $0.02839 \pm 5.43 \times 10^{-3}$ | $0.01005 \pm 3.40 \times 10^{-3}$ | $0.01043 \pm 3.28 \times 10^{-3}$ | $0.00548 \pm 2.58 \times 10^{-3}$ |
| SAC w/o cons | 0.8 | $0.02026 \pm 6.00 \times 10^{-3}$ | $0.01912 \pm 7.46 \times 10^{-3}$ | $0.01510 \pm 6.27 \times 10^{-3}$ | $0.00606 \pm 6.04 \times 10^{-3}$ |
| IPOPT w/ cons | 1.0 | $0.01273 \pm 6.68 \times 10^{-3}$ | $0.01011 \pm 7.32 \times 10^{-3}$ | $0.01140 \pm 8.15 \times 10^{-3}$ | $0.01965 \pm 9.77 \times 10^{-3}$ |
| IPOPT w/o cons | 1.0 | $0.02194 \pm 6.32 \times 10^{-3}$ | $0.01017 \pm 3.30 \times 10^{-3}$ | $0.01050 \pm 3.13 \times 10^{-3}$ | $0.00649 \pm 3.70 \times 10^{-3}$ |
| SAC w/o cons | 1.0 | $0.01712 \pm 5.97 \times 10^{-3}$ | $0.01688 \pm 6.40 \times 10^{-3}$ | $0.01628 \pm 8.17 \times 10^{-3}$ | $0.00519 \pm 4.66 \times 10^{-3}$ |

$J = 3$ unprotected operators for $g \in [0.2, 4]$ are available upon request.

## APPENDIX B: FLOW EQUATIONS

In this appendix, we present an alternative analysis for the adiabatic variation of the crossing equations, which aims to determine the infinitesimal vector of the CFT data in parameter space. The approach is similar in spirit to the flowing method of Ref. [79], except that it is not specific to extremal solutions and works within a general truncation scheme. We did not use this method to derive any of the results reported in the main part of this work, but we present it here as an idea that may be useful in future studies. Preliminary analysis shows that this method can produce high reward curves, at least within a finite region of parameter space. An interesting technical aspect of this approach is that it leads to a convex optimization problem, which is significantly simpler compared to the nonconvex problems we had to solve in the main text.

As always in this paper, we assume that we have a set of crossing equations (possibly supplemented with additional sum rules) for CFT data on a parameter space. The parameters are collectively denoted as $\lambda$ and can be continuous or discrete. If discrete, we compute finite differences instead of derivatives. Our crossing equations take the form (3), repeated here as

$$\sum_n \mathcal{C}_n(\lambda)\vec{F}_n(\lambda) + \vec{T}(\lambda) + \vec{r}(\lambda) = 0. \tag{B1}$$

We assume to have an exact or approximate solution of the CFT data at $\lambda$ and are interested in determining the solution at $\lambda + \delta\lambda$ for infinitesimal (or small finite) $\delta\lambda$.

Varying (B1) with respect to $\lambda$, we obtain the flow equation:

$$\sum_n \left[\vec{F}_n(\lambda)\delta\mathcal{C}_n + \mathcal{C}_n(\lambda)\frac{\partial\vec{F}_n}{\partial\Delta_n}(\lambda)\delta\Delta_n\right] + \delta\vec{T} + \delta\vec{r} = 0. \tag{B2}$$

We know by construction $\vec{r}$ as a function of $\lambda$ and can approximate $\delta\vec{r} \simeq \frac{\partial\vec{r}}{\partial\lambda}\delta\lambda$. In addition, by implementing our basic approximating assumption in the main text that the tail $\vec{T}$ has a weak $\lambda$ dependence, we can set $\delta\vec{T} \simeq 0$. These assumptions lead to the simplified approximate flow equation

$$\sum_n \left[\vec{F}_n(\lambda)\delta\mathcal{C}_n + \mathcal{C}_n(\lambda)\frac{\partial\vec{F}_n}{\partial\Delta_n}(\lambda)\delta\Delta_n\right] + \frac{\partial\vec{r}}{\partial\lambda}\delta\lambda \simeq 0, \tag{B3}$$

which is linear in the flow-vector unknowns $\delta\mathcal{C}_n, \delta\Delta_n$. An obvious strategy for proceeding is to try and solve for these unknowns by minimizing a quadratic cost function. This is a linear regression problem that can be handled with one's favorite deterministic optimizer. Unfortunately, this strategy is not optimal because it relies on the assumption that at every step one flows away from a good quality solution of the CFT data. As a result, even when starting from an initial point with knowledge of the exact solution, the errors will pile up quickly, and the quality of the search will decrease.

A more promising alternative is to consider the linearized version of the full crossing equation, not just its differential. Indeed, at $\lambda + \delta\lambda$, one could aim to solve the approximate equation

$$\sum_n \mathcal{C}_n(\lambda)\vec{F}_n(\lambda) + \vec{T}(\lambda) + \vec{r}(\lambda) + \sum_n \left[\vec{F}_n(\lambda)\delta\mathcal{C}_n + \mathcal{C}_n(\lambda)\frac{\partial\vec{F}_n}{\partial\Delta_n}(\lambda)\delta\Delta_n\right] + \frac{\partial\vec{r}}{\partial\lambda}\delta\lambda \simeq 0. \tag{B4}$$

The unknowns are again the flow-vector data $\delta\mathcal{C}_n, \delta\Delta_n$, and the problem is a linear regression problem. Since we included the zeroth order part in the first three terms, this approach has a chance to self-correct as one flows away from an exact solution.

We have noticed in preliminary applications that this adiabatic flow can be run quickly with a very small step $\delta\lambda$ to produce high-reward curves, but we have not studied it systematically. At finite distance in the flow, one could also combine it, or compare it, with a full nonconvex optimization run (like the ones reported in the main text). On the technical side, we noticed that a quick implementation of the linear regression in this context was possible using the SLSQP algorithm from NLopt, which is available within PyGMO. SLSQP is a sequential quadratic programming algorithm for nonlinearly constrained gradient-based optimization. We also noticed that the near degeneracies caused instabilities in the gradient descent, which could be addressed with the Ridge-regression regularization scheme common in machine learning; for a review, see [80].


[1] A. Cavaglià, N. Gromov, J. Julius, and M. Preti, Integrability and conformal bootstrap: One dimensional defect conformal field theory, Phys. Rev. D **105**, L021902 (2022).

[2] A. Cavaglià, N. Gromov, J. Julius, and M. Preti, Bootstrability in defect CFT: Integrated correlators and sharper bounds, J. High Energy Phys. 05 (2022) 164.

[3] M. Picco, S. Ribault, and R. Santachiara, A conformal bootstrap approach to critical percolation in two dimensions, SciPost Phys. **1**, 009 (2016).

[4] Y. He, J. L. Jacobsen, and H. Saleur, Geometrical four-point functions in the two-dimensional critical $Q$-state Potts model: The interchiral conformal bootstrap, J. High Energy Phys. 12 (2020) 019.

[5] N. Drukker, Z. Kong, and G. Sakkas, Broken global symmetries and defect conformal manifolds, Phys. Rev. Lett. **129**, 201603 (2022).

[6] A. Cavaglià, N. Gromov, J. Julius, and M. Preti, Integrated correlators from integrability: Maldacena-Wilson line in $\mathcal{N} = 4$ SYM, J. High Energy Phys. 04 (2023) 026.

[7] D. Simmons-Duffin, The conformal bootstrap, in *Theoretical Advanced Study Institute in Elementary Particle Physics: New Frontiers in Fields and Strings* (World Scientific, Singapore, 2017), https://doi.org/10.1142/9789813149441_0001.

[8] D. Poland, S. Rychkov, and A. Vichi, The conformal bootstrap: Theory, numerical techniques, and applications, Rev. Mod. Phys. **91**, 015002 (2019).

[9] S. M. Chester, Weizmann lectures on the numerical conformal bootstrap, Phys. Rep. **1045**, 1 (2023).

[10] T. Hartman, D. Mazac, D. Simmons-Duffin, and A. Zhiboedov, Snowmass white paper: The analytic conformal bootstrap, in *Snowmass 2021* (2022), arXiv:2202.11012.

[11] D. Poland and D. Simmons-Duffin, Snowmass white paper: The numerical conformal bootstrap, in *Snowmass 2021* (2022),arXiv:2203.08117.

[12] R. Rattazzi, V. S. Rychkov, E. Tonni, and A. Vichi, Bounding scalar operator dimensions in 4D CFT, J. High Energy Phys. 12 (2008) 031.

[13] F. Gliozzi, More constraining conformal bootstrap, Phys. Rev. Lett. **111**, 161602 (2013).

[14] F. Gliozzi and A. Rago, Critical exponents of the 3d Ising and related models from conformal bootstrap, J. High Energy Phys. 10 (2014) 042.

[15] F. Gliozzi, P. Liendo, M. Meineri, and A. Rago, Boundary and interface CFTs from the conformal bootstrap, J. High Energy Phys. 05 (2015) 036.

[16] F. Gliozzi, Truncatable bootstrap equations in algebraic form and critical surface exponents, J. High Energy Phys. 10 (2016) 037.

[17] I. Esterlis, A. L. Fitzpatrick, and D. Ramirez, Closure of the operator product expansion in the nonunitary bootstrap, J. High Energy Phys. 11 (2016) 030.

[18] S. Hikami, Conformal bootstrap analysis for the Yang–Lee edge singularity, Prog. Theor. Exp. Phys. **2018**, 053I01 (2018).

[19] S. Hikami, Conformal bootstrap analysis for single and branched polymers, Prog. Theor. Exp. Phys. **2018**, 123I01 (2018).

[20] W. Li, New method for the conformal bootstrap with OPE truncations, arXiv:1711.09075.

[21] G. Kántor, V. Niarchos, and C. Papageorgakis, Conformal bootstrap with reinforcement learning, Phys. Rev. D **105**, 025018 (2022).

[22] G. Kántor, V. Niarchos, and C. Papageorgakis, Solving conformal field theories with artificial intelligence, Phys. Rev. Lett. **128**, 041601 (2022).

[23] G. Kántor, V. Niarchos, C. Papageorgakis, and P. Richmond, 6D (2,0) bootstrap with the soft-actor-critic algorithm, Phys. Rev. D **107**, 025005 (2023).

[24] D. Poland, V. Prilepina, and P. Tadić, The five-point bootstrap, J. High Energy Phys. 10 (2023) 153.

[25] D. Grabner, N. Gromov, and J. Julius, Excited states of one-dimensional defect CFTs from the quantum spectral curve, J. High Energy Phys. 07 (2020) 042.

[26] J. Julius, Modern techniques for solvable models, Doctoral thesis, King's Coll. London, 2021, https://kclpure.kcl.ac.uk/portal/en/studentTheses/modern-techniques-for-solvable-models.

[27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, arXiv:1801.01290.

[28] F. Biscani and D. Izzo, A parallel global multiobjective framework for optimization: PAGMO, J. Open Source Software **5**, 2338 (2020).

[29] D. Simmons-Duffin, A semidefinite program solver for the conformal bootstrap, J. High Energy Phys. 06 (2015) 174.

[30] A. Wächter and L. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming, Math. Program. **106**, 25 (2006).

[31] A. Trenta, Reinforcement learning for conformal field theories, Master thesis, University of Pisa, 2023, https://etd.adm.unipi.it/theses/available/etd-04122023-120235/unrestricted/Trenta_master_thesis.pdf.

[32] J. Henriksson, S. R. Kousvos, and M. Reehorst, Spectrum continuity and level repulsion: The Ising CFT from infinitesimal to finite $\epsilon$, J. High Energy Phys. 02 (2023) 218.

[33] A. Castedo Echeverri, B. von Harling, and M. Serone, The effective bootstrap, J. High Energy Phys. 09 (2016) 097.

[34] D. Pappadopulo, S. Rychkov, J. Espin, and R. Rattazzi, OPE convergence in conformal field theory, Phys. Rev. D **86**, 105043 (2012).

[35] N. Su, The hybrid bootstrap, arXiv:2202.07607.

[36] Z. Komargodski and A. Zhiboedov, Convexity and liberation at large spin, J. High Energy Phys. 11 (2013) 140.

[37] A. L. Fitzpatrick, J. Kaplan, D. Poland, and D. Simmons-Duffin, The analytic bootstrap and AdS superhorizon locality, J. High Energy Phys. 12 (2013) 004.

[38] M. Reehorst, S. Rychkov, D. Simmons-Duffin, B. Sirois, N. Su, and B. Van Rees, Navigator function for the conformal bootstrap, SciPost Phys. **11**, 072 (2021).

[39] S. Caron-Huot, Analyticity in spin in conformal theories, J. High Energy Phys. 09 (2017) 078.

[40] D. Simmons-Duffin, D. Stanford, and E. Witten, A spacetime derivation of the Lorentzian OPE inversion formula, J. High Energy Phys. 07 (2018) 085.

[41] D. Carmi and S. Caron-Huot, A conformal dispersion relation: Correlations from absorption, J. High Energy Phys. 09 (2020) 009.

[42] N. Drukker and S. Kawamoto, Small deformations of supersymmetric Wilson loops and open spin-chains, J. High Energy Phys. 07 (2006) 024.

[43] N. Drukker, Integrable Wilson loops, J. High Energy Phys. 10 (2013) 135.

[44] D. Correa, J. Maldacena, and A. Sever, The quark anti-quark potential and the cusp anomalous dimension from a TBA equation, J. High Energy Phys. 08 (2012) 134.

[45] S. Giombi, R. Roiban, and A. A. Tseytlin, Half-BPS Wilson loop and $AdS_2/CFT_1$, Nucl. Phys. **B922**, 499 (2017).

[46] P. Liendo, C. Meneghelli, and V. Mitev, Bootstrapping the half-BPS line defect, J. High Energy Phys. 10 (2018) 077.

[47] N. Gromov, F. Levkovich-Maslyuk, and G. Sizov, Quantum spectral curve and the numerical solution of the spectral problem in AdS5/CFT4, J. High Energy Phys. 06 (2016) 036.

[48] N. Gromov, A. Hegedus, J. Julius, and N. Sokolova, Fast QSC solver: Tool for systematic study of $N = 4$ super-Yang-Mills spectrum, arXiv:2306.12379.

[49] S. Abel, A. Constantin, T. R. Harvey, and A. Lukas, String model building, reinforcement learning and genetic algorithms, in *Nankai Symposium on Mathematical Dialogues: In celebration of S. S. Chern's 110th Anniversary* (2021), arXiv:2111.07333.

[50] S. Abel, A. Constantin, T. R. Harvey, and A. Lukas, Evolving heterotic gauge backgrounds: Genetic algorithms versus reinforcement learning, Fortschr. Phys. **70**, 2200034 (2022).

[51] A. Cole, S. Krippendorf, A. Schachner, and G. Shiu, Probing the structure of string theory vacua with genetic algorithms and reinforcement learning, in *35th Conference on Neural Information Processing Systems* (2021), arXiv:2111.11466.

[52] S. Krippendorf, D. Lust, and M. Syvaeri, Integrability ex machina, Fortschr. Phys. **69**, 2100057 (2021).

[53] S. Lal, S. Majumder, and E. Sobko, The R-mAtrIx Net, arXiv:2304.07247.

[54] A. Ashmore, Y.-H. He, and B. A. Ovrut, Machine learning Calabi–Yau metrics, Fortschr. Phys. **68**, 2000068 (2020).

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

[56] P. Berglund, G. Butbaia, T. Hübsch, V. Jejjala, D. Mayorga Peña, C. Mishra, and J. Tan, Machine learned Calabi-Yau metrics and curvature, arXiv:2211.09801.
[57] K. Hashimoto, AdS/CFT correspondence as a deep Boltzmann machine, Phys. Rev. D **99**, 106017 (2019).
[58] J. Halverson, A. Maiti, and K. Stoner, Neural networks and quantum field theory, Mach. Learn. Sci. Tech. **2**, 035002 (2021).
[59] H. Erbin, V. Lahoche, and D. O. Samary, Nonperturbative renormalization for the neural network-QFT correspondence, Mach. Learn. Sci. Tech. **3**, 015027 (2022).
[60] D. S. Berman and M. S. Klinger, The inverse of exact renormalization group flows as statistical inference, arXiv:2212.11379.
[61] D. S. Berman, M. S. Klinger, and A. G. Stapleton, Bayesian renormalization, Mach. Learn. Sci. Tech. **4**, 045011 (2023).
[62] H. Erbin and A. H. Fırat, Characterizing 4-string contact interaction using machine learning, arXiv:2211.09129.
[63] P. de Haan, C. Rainone, M. C. N. Cheng, and R. Bondesan, Scaling up machine learning for quantum field theory with equivariant continuous flows, arXiv:2110.02673.
[64] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, Learning lattice quantum field theories with equivariant continuous flows, arXiv:2207.00283.
[65] Y.-H. He, E. Heyes, and E. Hirst, Machine learning in physics and geometry, arXiv:2303.12626.
[66] G. Kántor, Exact and numerical methods in (super)conformal field theories, https://qmro.qmul.ac.uk/xmlui/bitstream/handle/123456789/79600/Thesis_Gergely_Kantor.pdf.
[67] https://github.com/vniarchos/BootSTOP.
[68] https://github.com/esa/PyGMO2.
[69] P. Ferrero and C. Meneghelli, Bootstrapping the half-BPS line defect CFT in $N = 4$ supersymmetric Yang-Mills theory at strong coupling, Phys. Rev. D **104**, L081703 (2021).
[70] N. Gromov, V. Kazakov, S. Leurent, and D. Volin, Quantum spectral curve for planar $\mathcal{N} = 4$ Super-Yang-Mills theory, Phys. Rev. Lett. **112**, 011602 (2014).
[71] N. Gromov, V. Kazakov, S. Leurent, and D. Volin, Quantum spectral curve for arbitrary state/operator in $AdS_5/CFT_4$, J. High Energy Phys. 09 (2015) 187.
[72] C. Beem, M. Lemos, L. Rastelli, and B. C. van Rees, The (2, 0) superconformal bootstrap, Phys. Rev. D **93**, 025016 (2016).
[73] M. Lemos, B. C. van Rees, and X. Zhao, Regge trajectories for the (2, 0) theories, J. High Energy Phys. 01 (2022) 022.
[74] L. F. Alday and S. M. Chester, Pure anti-de Sitter supergravity and the conformal bootstrap, Phys. Rev. Lett. **129**, 211601 (2022).
[75] C. Beem, L. Rastelli, and B. C. van Rees, The $\mathcal{N} = 4$ superconformal bootstrap, Phys. Rev. Lett. **111**, 071601 (2013).
[76] C. Beem, L. Rastelli, and B. C. van Rees, More $\mathcal{N} = 4$ superconformal bootstrap, Phys. Rev. D **96**, 046014 (2017).
[77] S. M. Chester, R. Dempsey, and S. S. Pufu, Bootstrapping $\mathcal{N} = 4$ super-Yang-Mills on the conformal manifold, J. High Energy Phys. 01 (2023) 038.
[78] T. King, S. Butcher, and L. Zalewski, Apocrita—High performance computing cluster for Queen Mary University of London, 10.5281/zenodo.438045.
[79] S. El-Showk and M. F. Paulos, Extremal bootstrapping: Go with the flow, J. High Energy Phys. 03 (2018) 148.
[80] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R* (Springer, New York, 2014).