Bootstrability with Improved Truncation Methods

Vasilis Niarchos

University of Crete

based on 2108.08859, 2108.09330, 2209.02801, **2306.15730**

with **G. Kantor, C. Papageorgakis, P. Richmond, A.G. Stapleton, M. Woolley** (QMUL)

This talk is about:

alternative strategies in the (numerical) **conformal bootstrap**

1. **Physics (QFT)** part:

alternative **non-perturbative** solution methods in CFTs **[CFT problem]**

The standard **linear functional method** vs other options

2. **Computational** part:

non-convex, large-scale optimization problems

Is **Machine Learning (ML) / Artificial Intelligence (AI)** a useful tool?

[numerical, Computer Science problem]

Example: Bootstrability [Cavaglia et al. '22]

bootstrap → integrability

in 1d CFT on 1/2-BPS Wilson line in planar N=4 SYM theory

Sample of results for 3 squared (unprotected) OPE-coefficients

SDPB (~1000 digit precision)

C_1^2	C_2^2	C_3^2
$0.294014873 \pm 4.88{\cdot}10^{-8}$	$0.039788 \pm 4.10{\cdot}10^{-4}$	$0.146757 \pm 5.82{\cdot}10^{-4}$
$0.294014228 \pm 6.77{\cdot}10^{-7}$	$0.041832 \pm 1.86{\cdot}10^{-3}$	$0.144100 \pm 2.39{\cdot}10^{-3}$

$$\lambda = (4\pi)^2 \simeq 157.91$$

improved truncation (16 digit precision)

- For the **first time** a non-rigorous method, not relying on positivity, competes so directly with the so-far standard rigorous tools in numerical conformal bootstrap!

That opens up many possibilities…

- **AI** algos could play an interesting new role in theoretical problems

I will comment on a comparison between ML and non-ML algorithms in a specific example

Some motivation

- Why Conformal Field Theories (CFTs)?

  — UV/IR behaviour of QFTs

  — Phase transitions

  — Quantum Gravity via the AdS/CFT correspondence

  & via the worldsheet description of strings…

- Why bootstrap?

The non-perturbative structure of QFT is rich, but poses a hard conceptual and computational problem

— Real-world physical systems with strong interactions

— QFT-QFT (strong-weak) dualities

— QFT-gravity dualities (holography)

— New exotic non-Lagrangian QFTs from String Theory…

('kills' Lattice)

We need a (new) powerful framework for all these cases…

- The **conformal bootstrap programme**

  [Ferrara-Grillo-Gatto '73, Polyakov '74] + [Rattazzi-Rychkov-Tonni-Vichi '08]

  aims to solve CFTs non-perturbatively leveraging general principles of

  symmetry (*without using the path-integral*)

— Caution: Symmetry alone cannot not be enough!

Combine: analytical, numerical, exact, perturbative…

at the center of this talk

- Recent progress in many fronts:

  analytical/numerical conformal bootstrap, new exact methods in

  SUSY gauge theories (e.g. integrability, localization) …

How symmetry helps in CFTs

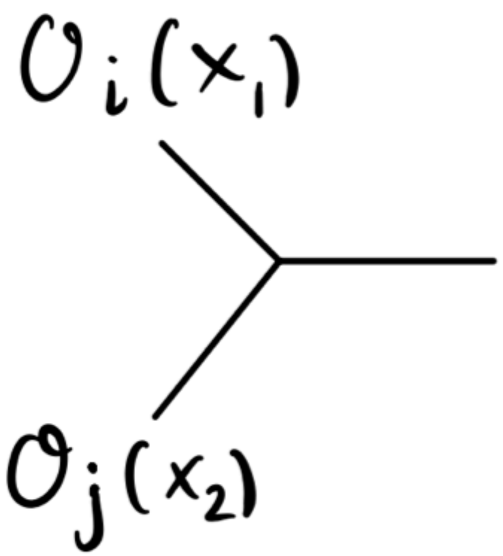
Local CFT data: operators $\mathcal{O}(x)$ with some quantum numbers under global symmetries [scaling dimension Δ, spin s, charges Q…]

2-point correlation functions: $$\langle \mathcal{O}^{(1)}_{\Delta}(x_1)\mathcal{O}^{(2)}_{\Delta}(x_2)\rangle = \frac{G_{12}}{|x_1 - x_2|^{2\Delta}}$$

3-point functions: $$\langle \mathcal{O}_{\Delta_1}(x_1)\mathcal{O}_{\Delta_2}(x_2)\mathcal{O}_{\Delta_3}(x_3)\rangle = \frac{C_{123}}{|x_{12}|^{\Delta_{12,3}}|x_{23}|^{\Delta_{23,1}}|x_{13}|^{\Delta_{13,2}}}$$

$$\Delta_{ij,k} := \Delta_i + \Delta_j - \Delta_k$$

Operator Product Expansion (OPE)

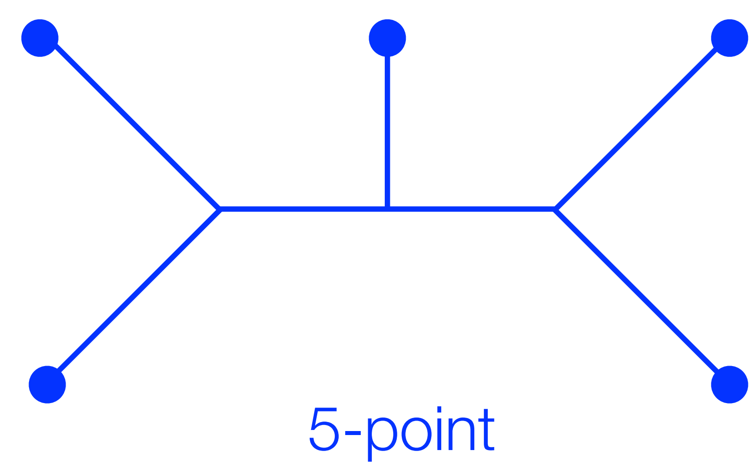


$$\sum_k \boxed{C_{ij}^{k}}\, \hat{f}_{ij}^{k}(x_1, x_2, \partial_{x_2})\, \mathcal{O}_k(x_2)$$

$\langle \mathcal{O}_i \mathcal{O}_j \mathcal{O}_k \rangle$

from Δ_i, C_{ij}^{k} → all N-point functions $\langle \mathcal{O}_1(x_1)\mathcal{O}_2(x_2)\ldots\mathcal{O}_N(x_N)\rangle$

completely fixed by conformal symmetry



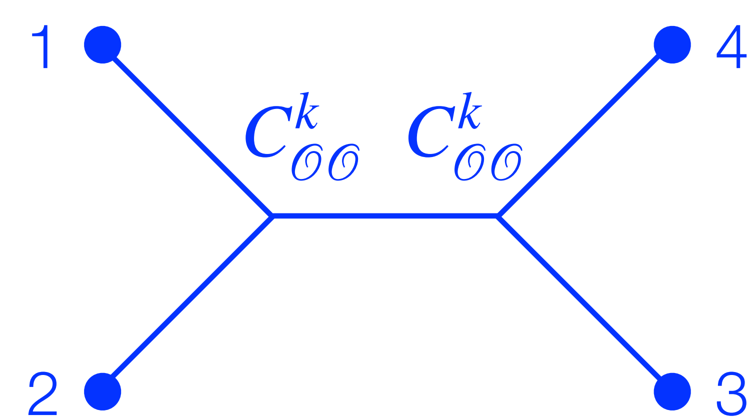
Example: 4-point functions (identical operators)

$$\langle \mathcal{O}(x_1)\mathcal{O}(x_2)\mathcal{O}(x_3)\mathcal{O}(x_4)\rangle = \frac{g(u,v)}{|x_{12}|^{2\Delta}|x_{34}|^{2\Delta}}$$

$$u := z\bar{z}$$

$$v := (1-z)(1-\bar{z})$$

$$u = \frac{x_{12}^2 x_{34}^2}{x_{13}^2 x_{24}^2}, \quad v = \frac{x_{23}^2 x_{14}^2}{x_{13}^2 x_{24}^2}$$ (cross ratios)



Using the (12) and (34) OPEs we get an

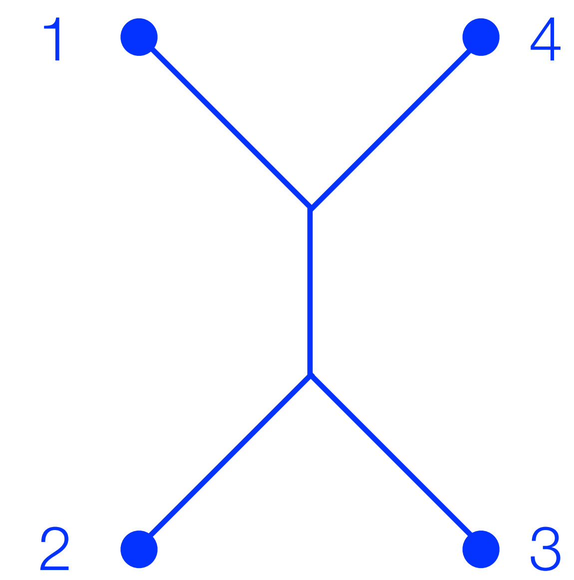
expansion of the form:

$$g(u,v) = \sum_k \left(C^k_{\mathcal{O}\mathcal{O}}\right)^2 g_{\Delta_k,\ell_k}(u,v)$$

conformal blocks
known functions

The same function $g(u,v)$ can be obtained with the (14)-(23) OPE

$$g(u,v) = \left(\frac{u}{v}\right)^{\Delta} \sum_k \left(C^k_{\mathcal{O}\mathcal{O}}\right)^2 g_{\Delta_k,\ell_k}(v,u)$$



Yields a crossing equation of the form

$$\sum_k \left(C^k_{\mathcal{O}\mathcal{O}}\right)^2 F_{\Delta_k}(u,v) = 0$$ to be solved for the CFT data $\left(\Delta_k, C^k_{\mathcal{O}\mathcal{O}}\right)$

Converted to an optimization problem:

— linear functional method (standard, convex optimization)

— we will discuss alternatives (non-convex opt) ➨ ML/AI comes in

Solving the crossing equations

$$\mathfrak{C}_{\mathcal{O}} := (C^{k}_{\mathcal{O}\mathcal{O}})^2$$

$$\sum_{\mathcal{O}} \mathfrak{C}_{\mathcal{O}_{\Delta,s}} F_{\mathcal{O}_{\Delta,s}}(z,\bar{z}) = 0$$

linear in $\mathfrak{C}$ non-linear in Δ

- functional dependence on $z, \bar{z}$
- infinite unknowns $\mathfrak{C}, \Delta$
- infinite number of 4-point functions (& similar sum-rules)

Linear functional method [Rattazi-Rychkov-Tonni-Vichi, '08]

- Act on crossing eq. with linear functionals α

$$\sum_{\mathcal{O}} \underbrace{C_{\mathcal{O}_{\Delta,s}}}_{\geq 0} \, \underbrace{\alpha\left(F_{\mathcal{O}_{\Delta,s}}\right)}_{\geq 0} = 0 \qquad \left[\alpha\left(F_{\mathcal{O}_{0,0}}\right) = 1\right]$$

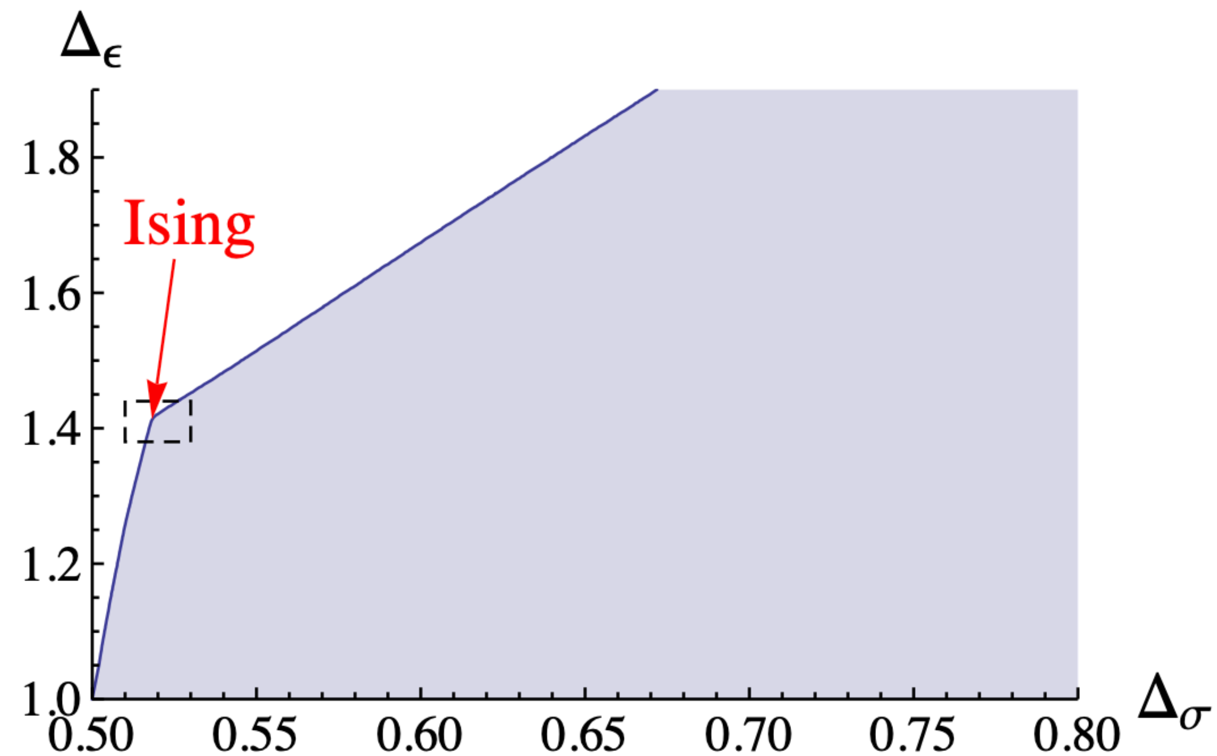
- Oracle assumptions, e.g. $\Delta_s \geq \Delta_{\min}$ for some operator

- Search for α :

$$\alpha\left[F_{\mathcal{O}_{\Delta,s}}\right] \geq 0 \qquad \forall \, \Delta \neq 0$$

- If α exists, the assumption is eliminated

This is a **classification algorithm**: assumption → NO / maybe



El-Showk et al, '12

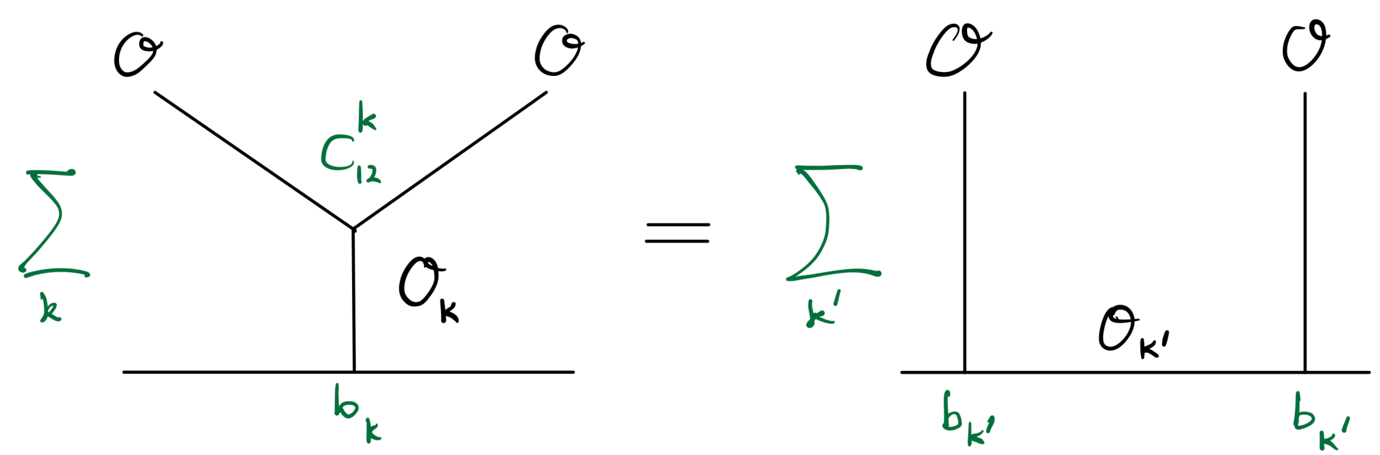
Pros:

- Rigorous
- Generic statements for CFTs
- Sometimes cusps/islands
- Powerful Semi-Definite Programming algorithm **SDPB** [Simmons-Duffin '15] **(convex optimization)**

Cons:

- Typically, the assumptions are blind. Difficult to:
  - Scan high-dimensional parameter spaces
  - Explore
  - "*Solve my theory*"

- Positivity $\mathfrak{C}_k \geq 0$ is not available in many contexts of interest
  - higher-point bootstrap
  - boundary CFT bootstrap
  - non-unitary theories



Cons:

- SDPB is powerful but expensive
  (100s of digits precision required)

- Multiple-correlator bootstrap is a challenge

$\Rightarrow$

desirable to supplement the linear functional method with other methods

Truncation methods offer an alternative but are notoriously messy…

Let's examine the issues. Our exact crossing eq. reads:

$$\sum_n C_n F_n(z,\bar{z}) + r(z,\bar{z}) = 0 \quad (*)$$

F_n: known functions of Δ_n

unknown C_n, Δ_n

$r(z,\bar{z})$: includes exactly known contributions

Step 1: discretize the $z, \bar{z}$ dependence

e.g., evaluation on a lattice or $z, \bar{z}$-derivatives on $z = \bar{z} = \frac{1}{2}$

Yields a finite system of equations (still exact)

$$\sum_n \mathbb{C}_n \vec{F}_n + \vec{r} = 0$$

Step 2: Identify a subset $\mathcal{S}$ of "*most significant*" operators in $\sum_n$

spin-partition: consider spins up to some max spin & for each spin s assume N_s operators

(don't need to know their Δ's)

Then (✶) becomes (still exact)

$$\sum_{n \in S} \mathbb{C}_n \vec{F}_n + \vec{T} + \vec{r} = 0$$

where $\vec{T} = \sum_{n \notin S} \mathbb{C}_n \vec{F}_n$

(Common) Step 3: Set $\overrightarrow{T} = 0$ for the `tail' and solve!

[Gliozzi '13] Pretend $\sum_{n \in \mathcal{S}} \mathfrak{C}_n \overrightarrow{F}_n + \vec{r} = 0$ is an exact over-constrained systems and try to solve it.

[Technicalities: drastic truncation, typically O(10) operators]

• [Li ’17] Formulate a cost function $\mathscr{L}$ (*e.g.*, L_2-norm of $\vec{E}$)

$$\text{cost}\left[\{\Delta_n, \mathbb{C}_n\}_{n\in S}\right] := \mathcal{L}(\vec{E})$$

and minimize it

$$\vec{E} \equiv \sum_{n\in S} \mathbb{C}_n \vec{F}_n + \vec{r}$$

This is now a highly non-trivial **non-convex** optimization problem

In [2108.08859, 2108.09330] we started thinking about this as a

large scale optimization problem (thinking that the larger $\mathcal{S}$ the better!)

This framework is still not good enough!

[1] It is not algorithmic

Q: How do you select the spin-partition?

[2] **Q**: What is the effect of the infinite number of operators in the dropped `tail' contribution $\overrightarrow{T}$? (across parameter spaces?)

[3] Near-degeneracies can grow very quickly with scaling dimension (and spectrum can be chaotic)

Q: How do you track this complexity?

[4] It is <u>not</u> a straightforward optimization problem!

- Multi-nodal high-dimensional landscape of $\mathscr{L}$

- Multiple basins of minima, complicated microstructure

  (& many configurations with comparable cost value)

  You may not be interested in the global minimum!

  The fastest optimizer may not be the right one!

Efficiency in 'local guided search'
Markov-chain algorithms
may be more suitable! Precisely
where AI/ML comes in for us…

[5] Related to the complexities of [3]+[4]:

adding more operators does not necessarily improve the accuracy

A viewpoint shift and proposed improvements

We now focus on families of CFTs: CFT[λ]

- Consider adiabatic deformations of a CFT solved at some λ^*

  [improves [1], consistent with the "*solve my CFT*" perspective]

- **DO NOT drop** the tail $\vec{T}$, approximate it!

  [improves [2]]

Static approximation: At the known solution point λ^*

$$\sum_n C_n^* \vec{F}_n^* + \vec{T}^* + \vec{r}^* = 0 \qquad \text{(exact)}$$

Allows us to determine the λ^*-tail from a finite number of CFT data:

$$\vec{T}^* = -\sum_{n\in S} C_n^* \vec{F}_n - \vec{r}^*$$

Assume that $\overrightarrow{T}(\lambda) \simeq \overrightarrow{T}(\lambda^*)$ (at least in some regime of λ)

Then,

$$\sum_{n\in S} C_n \vec{F}_n + \vec{r} - \sum_{n\in S} C_n^* \vec{F}_n^* - \vec{r}^* \simeq 0$$

$$\vec{E} - \vec{E}^* \simeq 0$$

to be solved by minimizing a new cost function $\mathscr{L}(\overrightarrow{E} - \overrightarrow{E}^*)$

- The contribution of nearly-degenerate operators can be approximated by effective operators [addresses [3]]

$$\sum_{\Delta \in \text{band} \subset S} C_n^{(exact)} \vec{F}_n^{(exact)} \simeq \sum_{n_{eff}} C_{n_{eff}} \vec{F}_{n_{eff}}$$

reduced # of ops

- Effective high-Δ operators can also be used to [address [5]]

  → "dynamical soft tail"

Different optimization algorithms may treat the soft tail differently!!

THE COMPUTATIONAL PROBLEM

Determine (local) minima of $\mathscr{L}(\overrightarrow{E} - \overrightarrow{E}*)$.

This is a non-convex optimization problem for the CFT data (actual+effective) $\{\Delta_n, \mathfrak{C}_n\} \in \mathscr{S}$ (order 100-1000 unknowns)

BootSTOP (Bootstrap STochastic OPtimiser)

From BootSTOP repository README

- Contents
  - Overview
  - Installation
  - Running the code
  - References

PyGMO (Python Parallel Global Multiobjective Optimizer
Izzo-Biscani — European Space Agency

SAC (Soft-Actor-Critic)
developed for task control in robotics

Overview

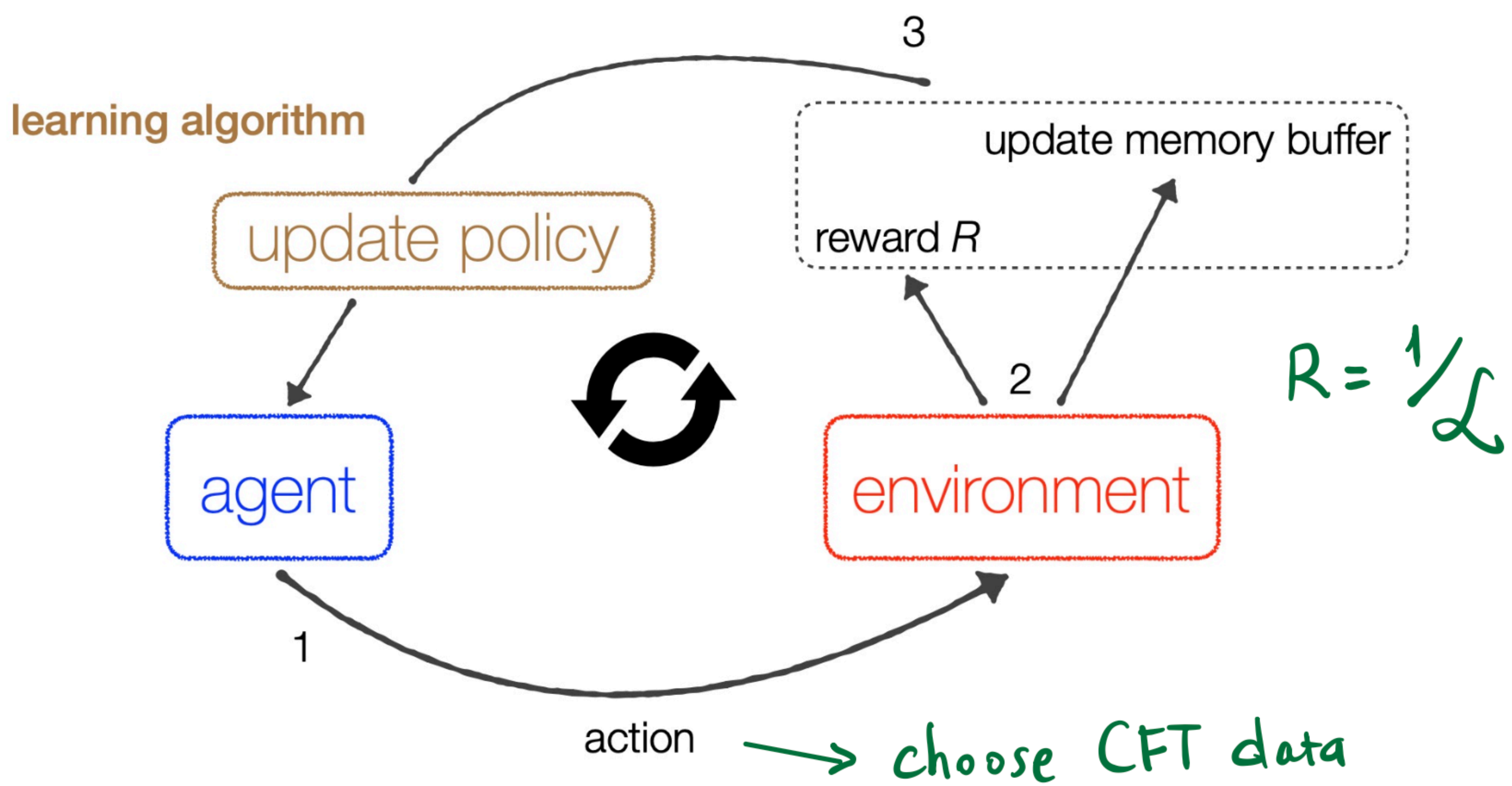
BootSTOP is a Python package for determining CFT data (OPE-coefficients squared and scaling dimensions) which minimise a theory's truncated crossing equation. To do this the code can apply either a custom PyTorch implementation of the Soft-Actor-Critic algorithm or one of the algorithms within the PyGMO package (information about PyGMO can be found on the PyGMO website).

At present the crossing equation for each of the following CFTs is coded within BootSTOP: 1D defect CFT (see [4]),

Soft Actor-Critic (SAC) algorithm

[Harnooja, Zhou, Abbeel, Levine ‘18]

- Stochastic optimization as a **Markov Decision Process**
- Handles continuous actions and state spaces



<u>Technical features of the 1d CFT application that follows</u>

- 2 sets of runs on QMUL HPC Apocrita
  - → IPOPT $\subset$ PyGMO (4×10^{8} agents, ~ 15mins)
  - → SAC (200 parallel agents, ~12hrs)

- 124 CFT data

- Crossing eqs with up to 260 or 700 derivatives (preloaded in BootSTOP)

- Machine precision (16 digits)

Application: 1d defect CFT on Wilson lines

Infinite straight (1/2-BPS) Wilson line in 4d N=4 SYM

$$\mathcal{W} = \mathrm{Tr} P \exp \int_{-\infty}^{\infty} dt \left(iA_t + \Phi_{||} \right)$$

Local operators $\mathcal{O}(t)$ inserted on Wilson line

captured by a 1d CFT

Interested in correlation functions

$$\langle\langle \mathcal{O}_1(t_1) \cdots \mathcal{O}_2(t_N) \rangle\rangle := \langle \mathrm{Tr} W_{-\infty}^{t_1} \mathcal{O}_1(t_1) W_{t_1}^{t_2} \mathcal{O}_2(t_2) \cdots \mathcal{O}_n(t_n) W_{t_n}^{+\infty} \rangle$$

Focus on $\langle\langle \Phi^1_\perp(x_1)\Phi^1_\perp(x_2)\Phi^1_\perp(x_3)\Phi^1_\perp(x_4)\rangle\rangle$

- $\Phi^1_\perp$ is one of the 5 transverse scalars of N=4 SYM
- Planar limit, dependence on the 't Hooft coupling $\lambda = g^2_{YM} N$

  Definition: $g := \dfrac{\sqrt{\lambda}}{4\pi}$

- Dependence of the 4-point function on the single cross-ratio

$$\chi = \frac{x_{12}x_{34}}{x_{13}x_{24}}$$

- Operators are organized in superconformal multiplets

  – protected $\mathscr{B}_k\,,\; k = 1,2,\ldots \quad \textbf{with}\; \Delta_k = k \qquad \left(\Phi_\perp^1 \in \mathscr{B}_1\right)$

  – long $\mathscr{L}_\Delta$

- OPE:

$$\mathscr{B}_1 \times \mathscr{B}_1 = 1 + \mathscr{B}_2 + \sum_n \mathscr{L}_{\Delta_n}$$

- Crossing equation …

known, depends on λ

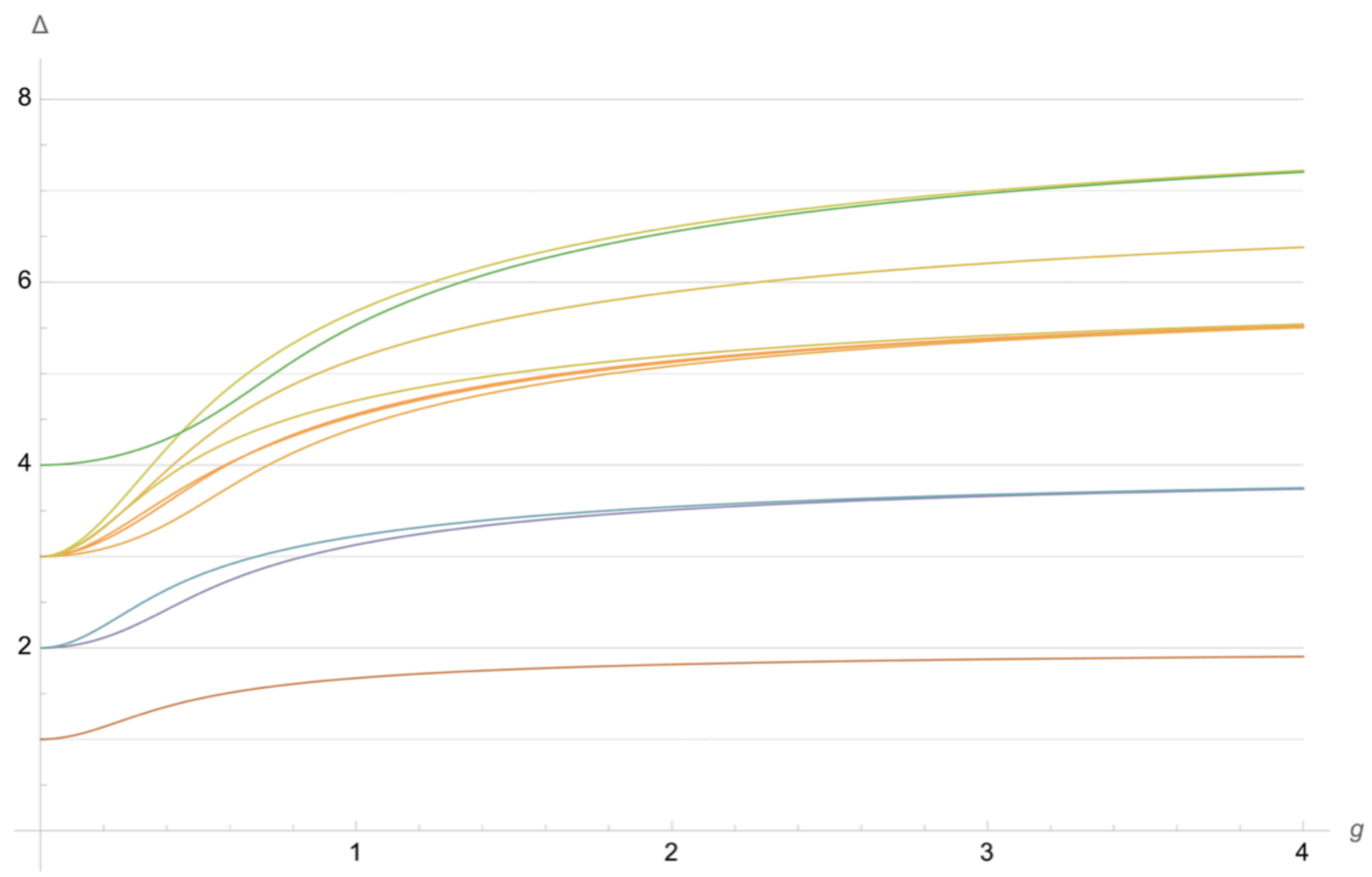
$$\sum_n C_n^2\, G_{\Delta_n}(\chi) = H(\chi)$$

unknown

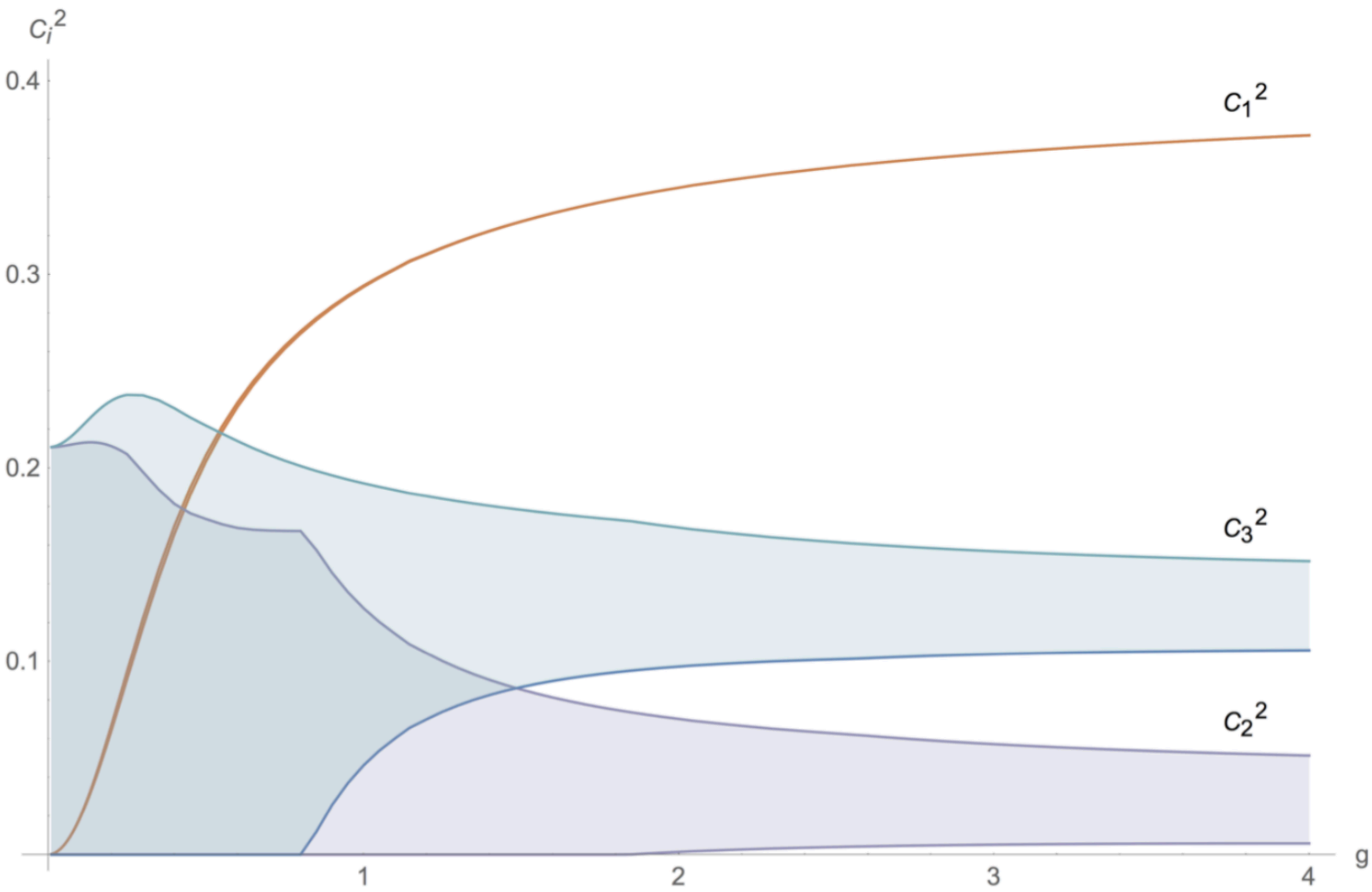
Bootstrability [Cavaglia-Gromov-Julius-Preti, '21]

- Use integrability (QSC) to fix (some of the) planar Δ_n
- Use bootstrap to determine C_n^2 $\left(C_n := C_{\mathscr{B}_1 \mathscr{B}_1 \mathscr{L}_{\Delta_n}} \right)$

Fix 10 long Δ_n's and determine $C_1^2,\ C_2^2,\ C_3^2$



Strategy 1: Linear functional method [Cavaglia et al, '21]



Strategy 2: Traditional truncation

- Truncate to 10 operators with fixed scaling dimensions

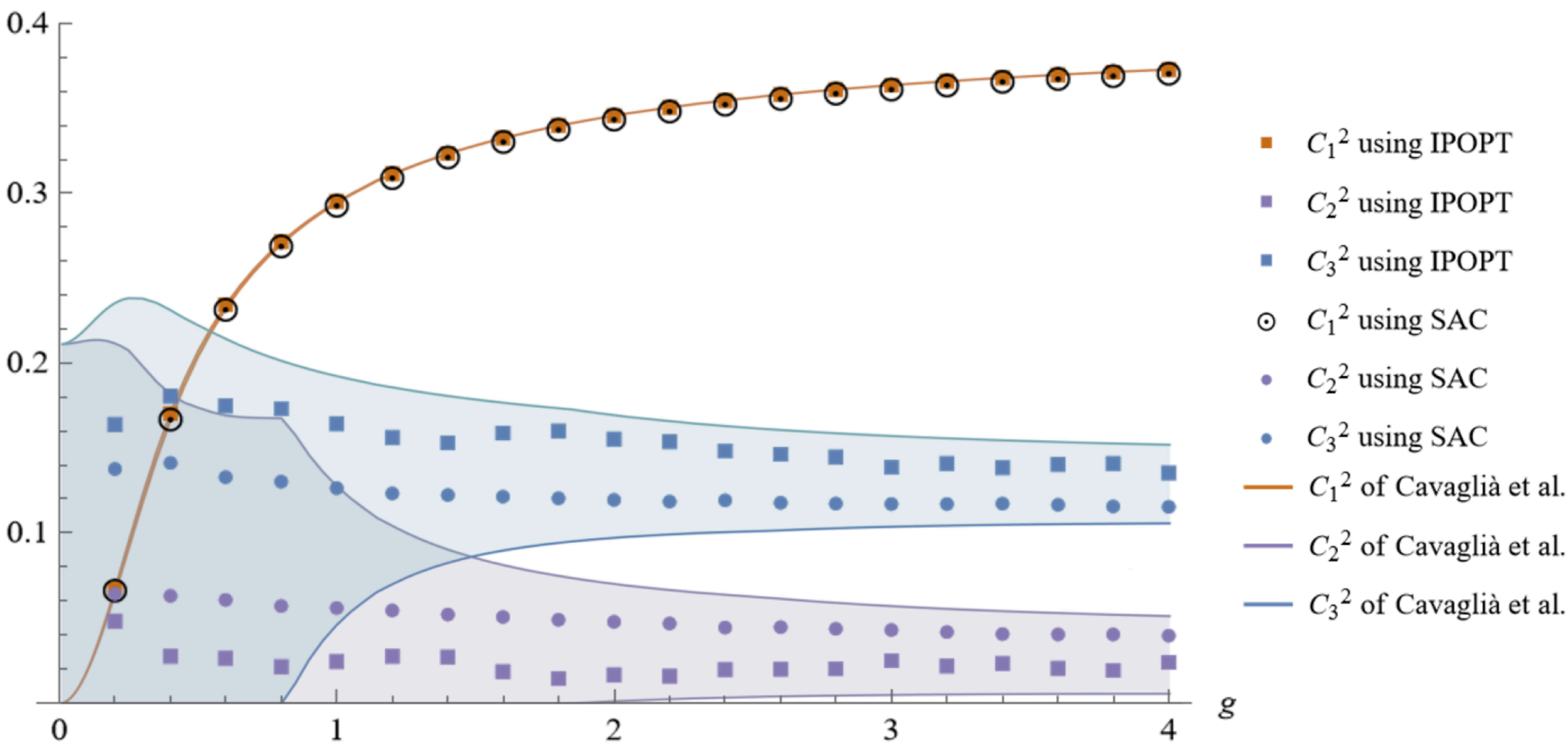
- Minimize the root mean square cost function

With linear dependence on the unknowns $\mathfrak{C}_n := C_n^2$ this is a linear-regression problem

The results are **not** impressive

Strategy 3: Improved Truncation

[VN-Papageorgakis-Richmond-Stapleton-Woolley, '23]



- C_1^2 results reproduced quite well

e.g., at $g = 0.2$

SDPB	0.0663 ± 1.9 x 10^{-3}
IPOPT	0.066073 ± 4.2 x 10^{-5}
SAC	0.067339 ± 1.26 x 10^{-3}

rigorous

statistical

- IPOPT & SAC treat the effective spectrum differently. Spread between IPOPT-SAC shows correlation with size of SDPB allowed regions

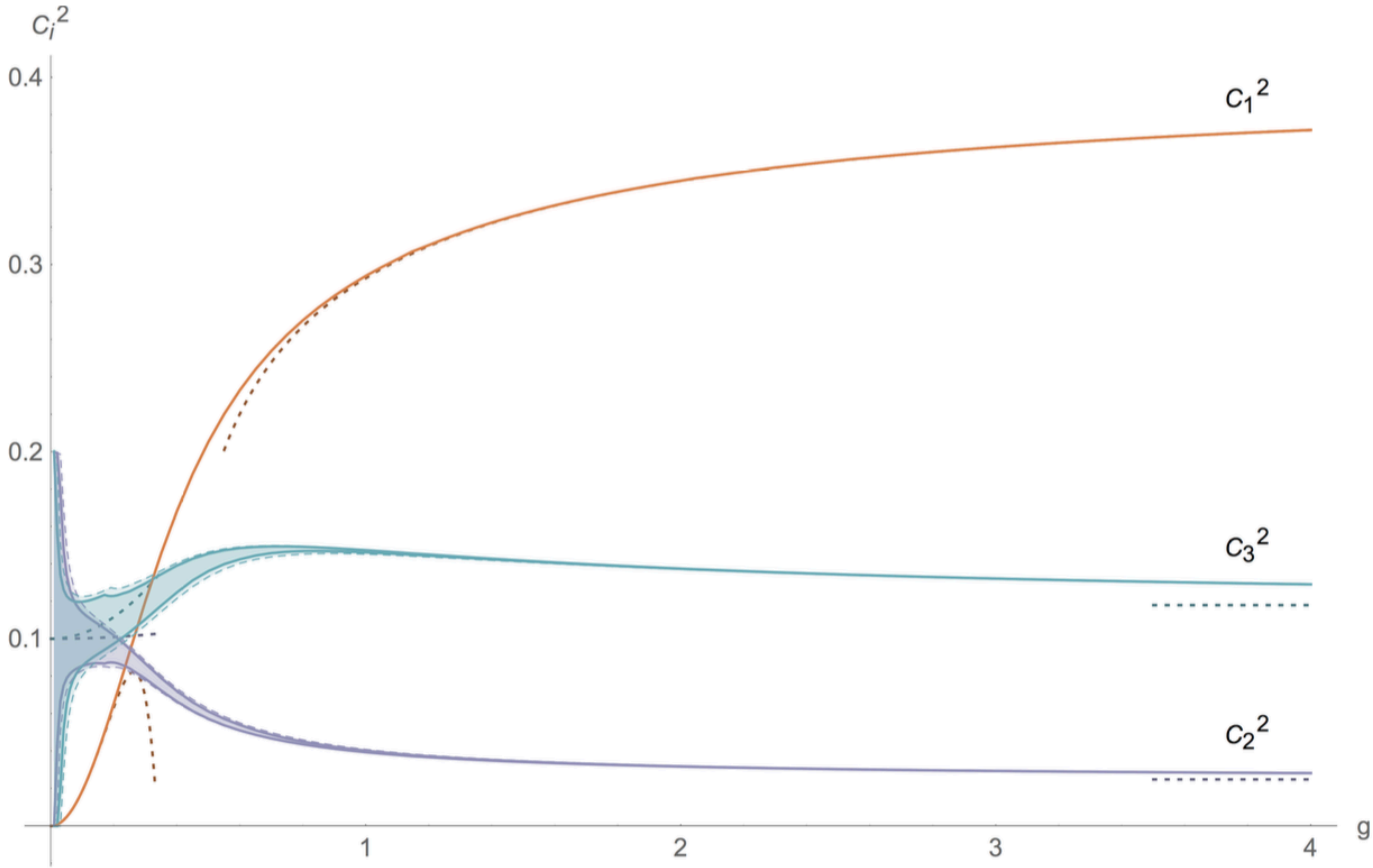
- SAC mean is surprisingly accurate !!

Bootstrability with integrated constraints [Cavaglia et al, ’22]

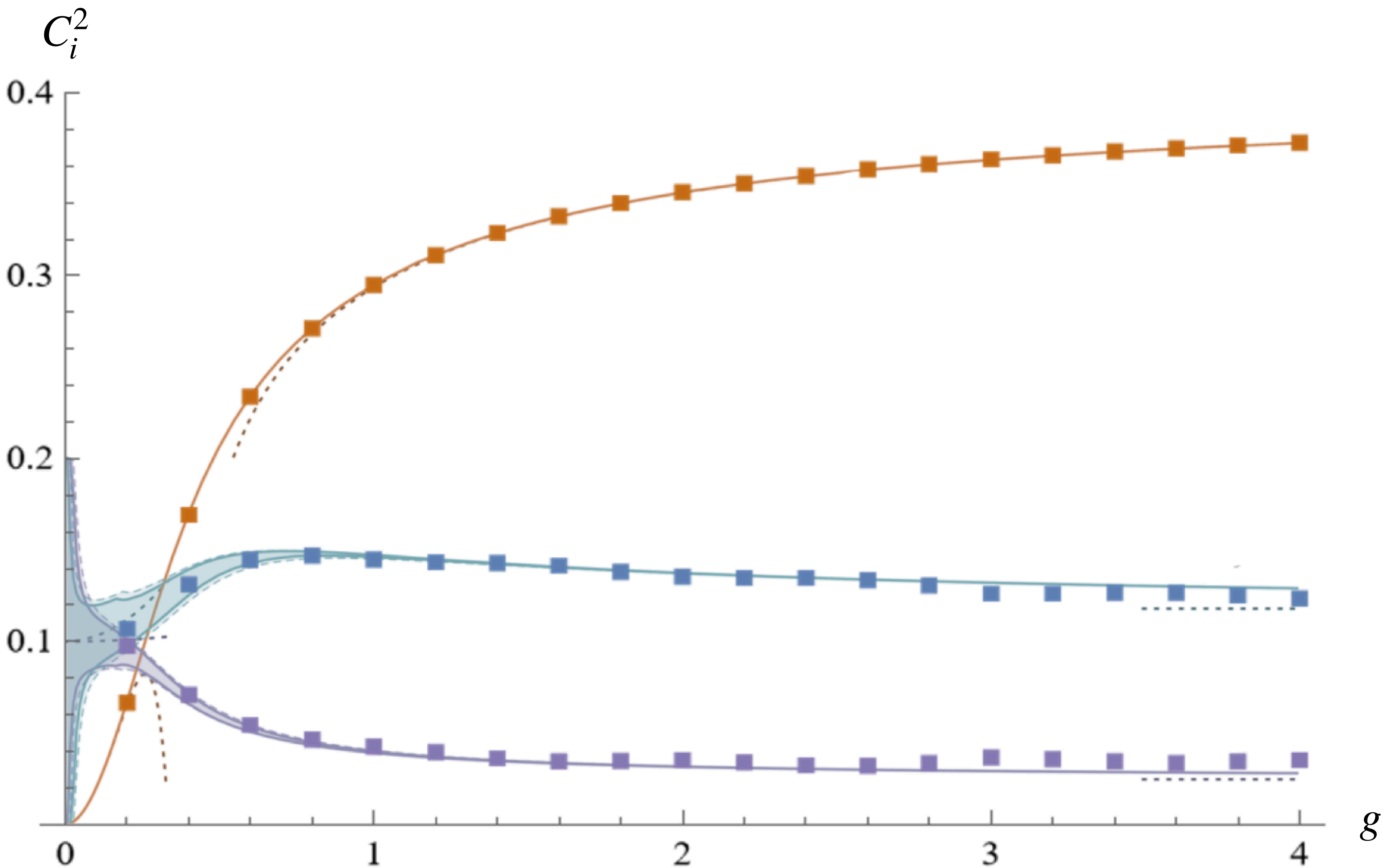
Integrated correlators [Drukker et al, ’22] [Cavaglia et al, ’22] yield 2 extra sum rules for the defect CFT data **that depend on g**.

Use them to tighten the bootstrap bounds !

Linear functional method [Cavaglia et al, ’22]



Improved Truncation [VN-Papageorgakis-Richmond-Stapleton-Woolley, ’23]



- Improved truncation performs as well (sometimes even better!) with much less demand on numerical precision (**16 vs 1000 digits**)

Method	g	C_1^2	C_2^2	C_3^2
[2]	0.2	$0.065679029 \pm 6.95 \cdot 10^{-7}$	$0.09452 \pm 7.25 \cdot 10^{-3}$	$0.1101 \pm 1.27 \cdot 10^{-2}$
IPOPT	0.2	$0.06567873 \pm 1.55 \cdot 10^{-7}$	$0.09683 \pm 1.41 \cdot 10^{-3}$	$0.1063 \pm 2.42 \cdot 10^{-3}$
[2]	0.4	$0.16838882 \pm 1.29 \cdot 10^{-6}$	$0.06925 \pm 2.80 \cdot 10^{-3}$	$0.13196 \pm 7.16 \cdot 10^{-3}$
IPOPT	0.4	$0.16838814 \pm 6.13 \cdot 10^{-7}$	$0.07010 \pm 1.06 \cdot 10^{-3}$	$0.13026 \pm 2.58 \cdot 10^{-3}$
[2]	0.6	$0.233041731 \pm 4.49 \cdot 10^{-7}$	$0.05246 \pm 1.47 \cdot 10^{-3}$	$0.14546 \pm 2.99 \cdot 10^{-3}$
IPOPT	0.6	$0.233041064 \pm 8.18 \cdot 10^{-7}$	$0.05347 \pm 1.30 \cdot 10^{-3}$	$0.14376 \pm 2.37 \cdot 10^{-3}$

- The soft tail works well everywhere from weak to strong coupling!

- New predictions for higher states are possible
(improvements expected with more QSC input)

- Multiple-correlator bootstrap is cheap and technically possible

Take home messages

- Improved truncation methods are flexible, cheap & can be efficient and accurate!

- We can use them to attack physically interesting situations where positivity is absent

- AI-powered methods may play a useful role
  - SAC mean is impressively accurate considering the numbers we ran (200 SAC vs 4×10^8 IPOPT agents)
  - Better RL implementations? Collaborative AI, e.g. MARL?

Some (preliminary results) on the 6d N=(2,0) superconformal bootstrap

(without improvements in truncation method)

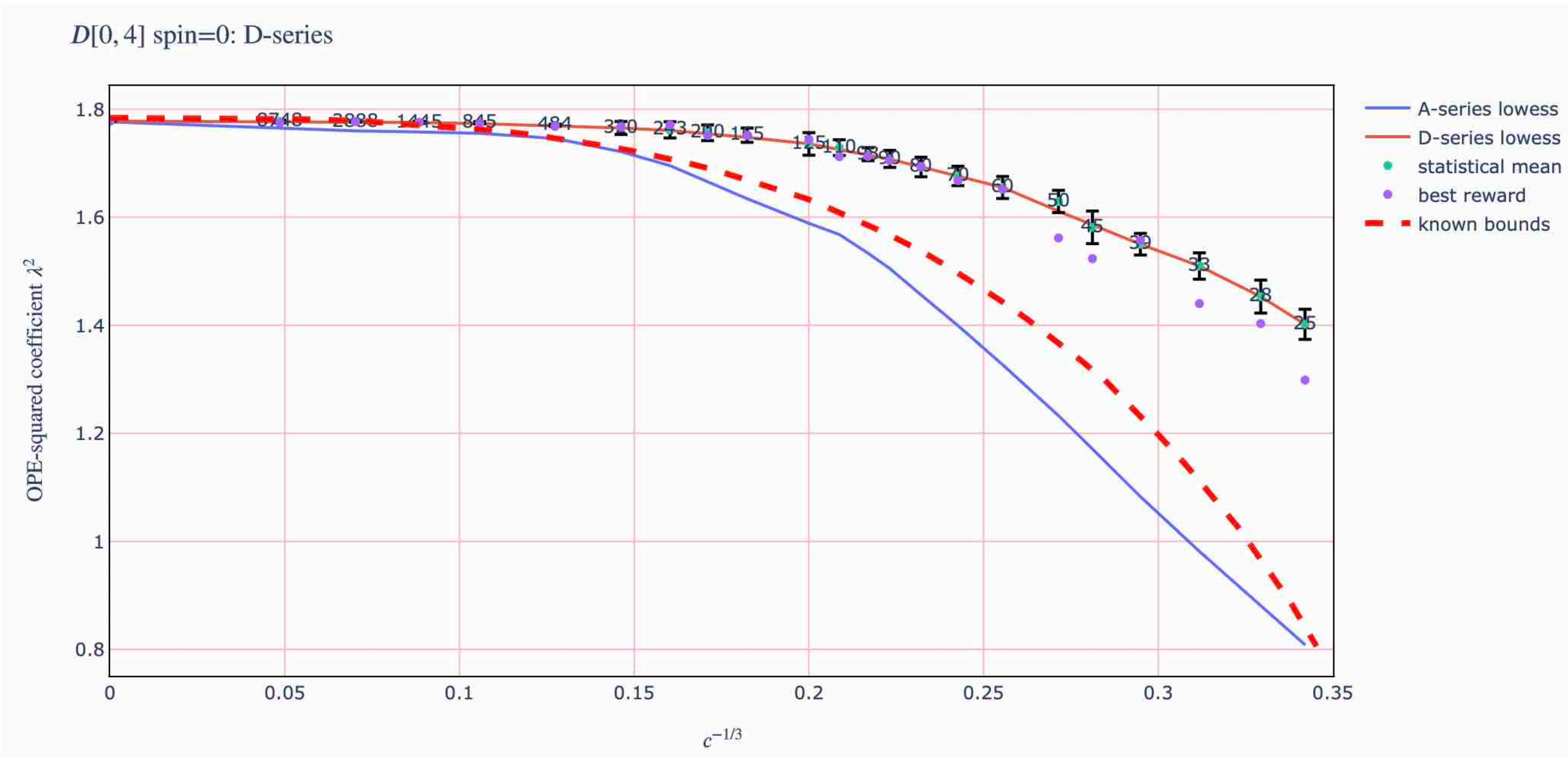
[Kantor-VN-Papageorgakis-Richmond]

- No Lagrangian —only AdS/CFT at $c \to \infty$
- A- and D-series theories. Can bootstrap distinguish them?

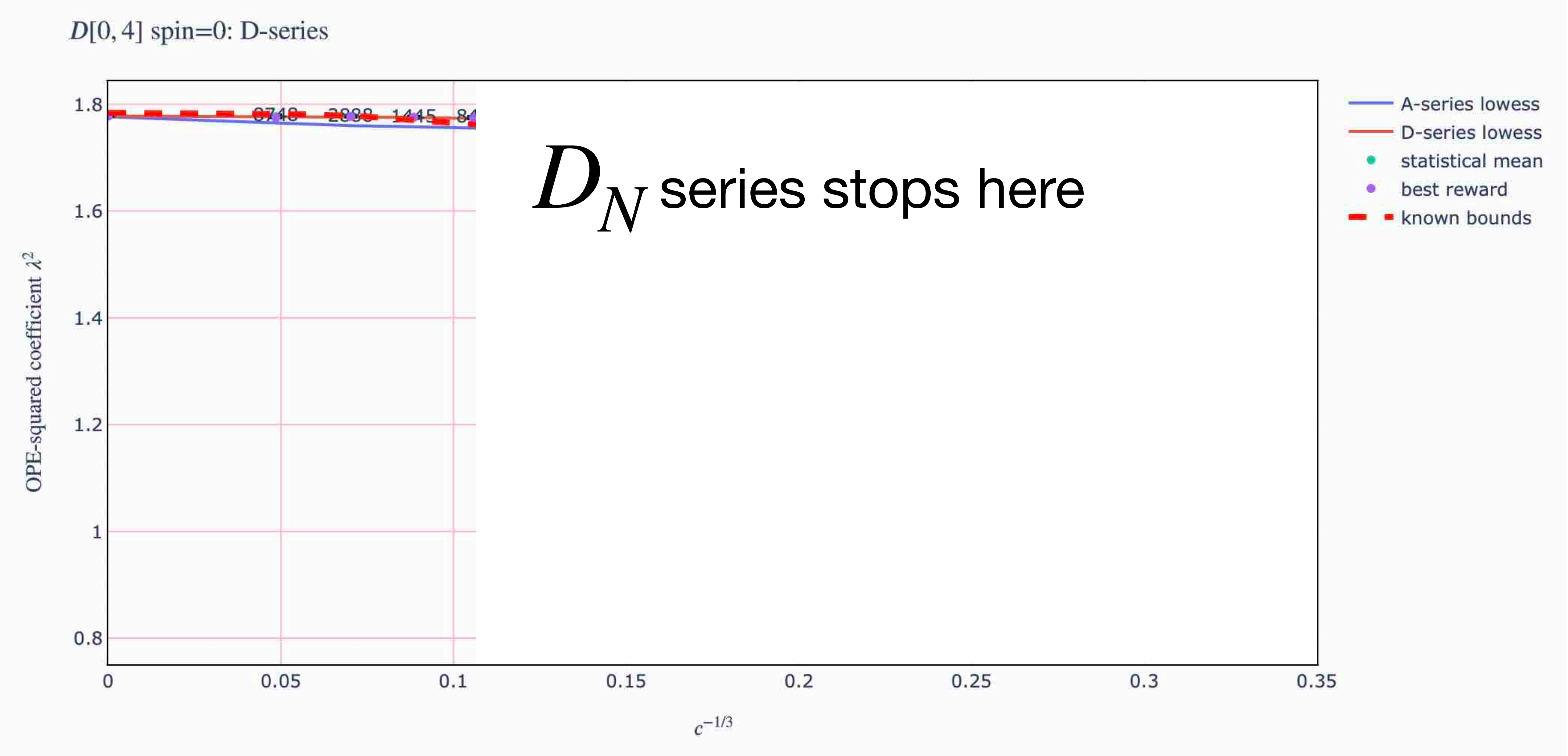
  A-series: $c = 25$ to $c = \infty$

  D-series: $c = 676$ to $c = \infty$

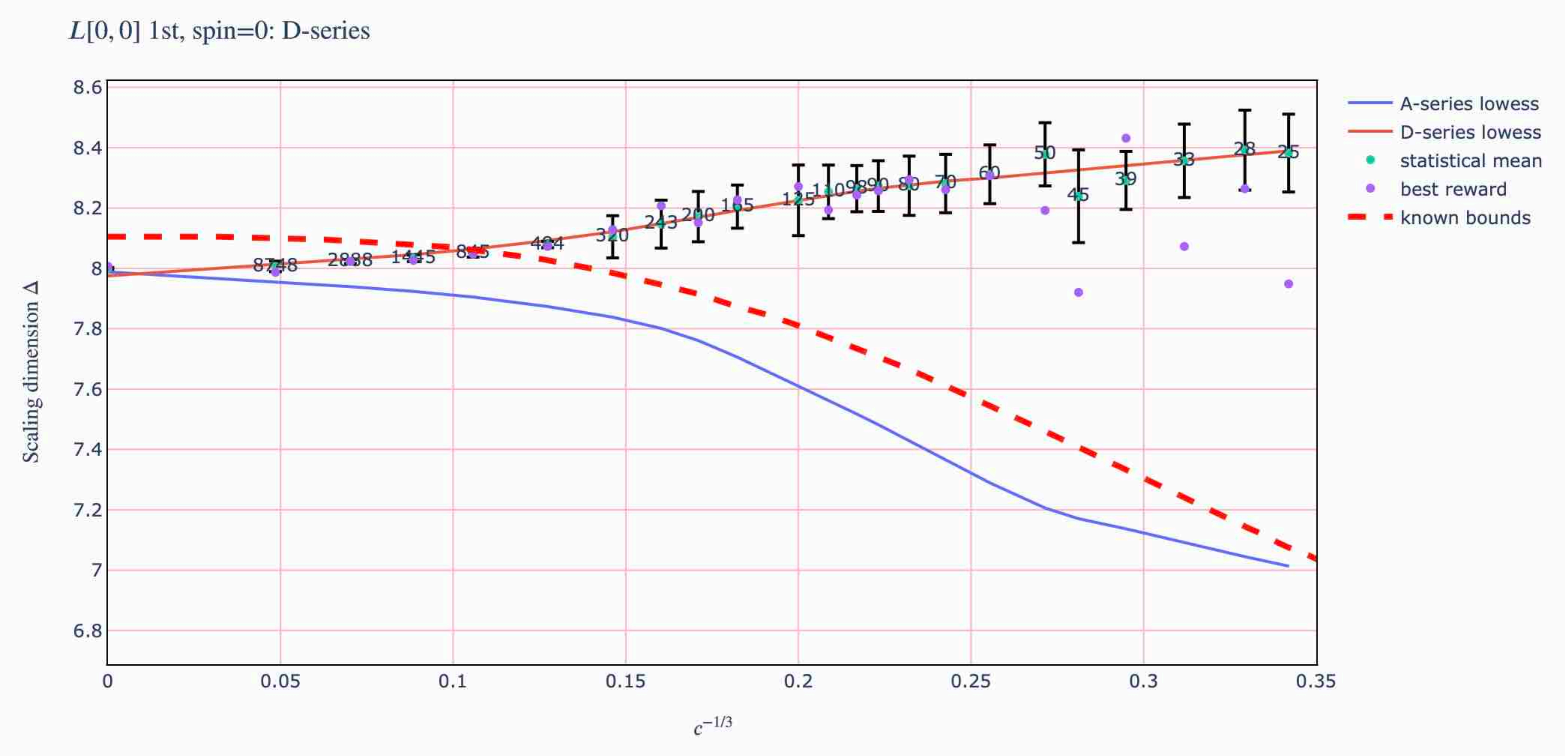
Results (lowest protected multiplet “D series”)



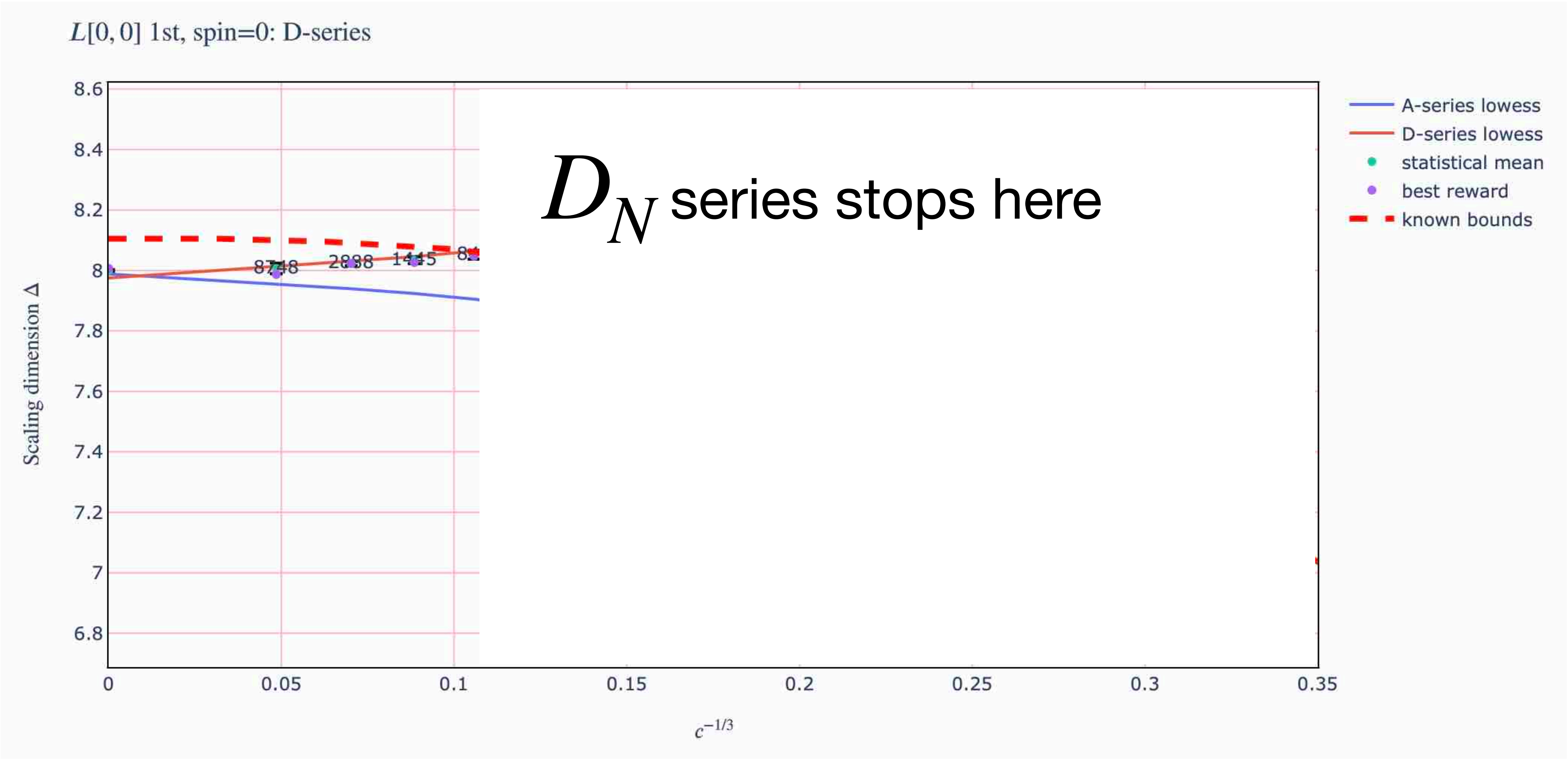
Results (lowest protected multiplet "D series")



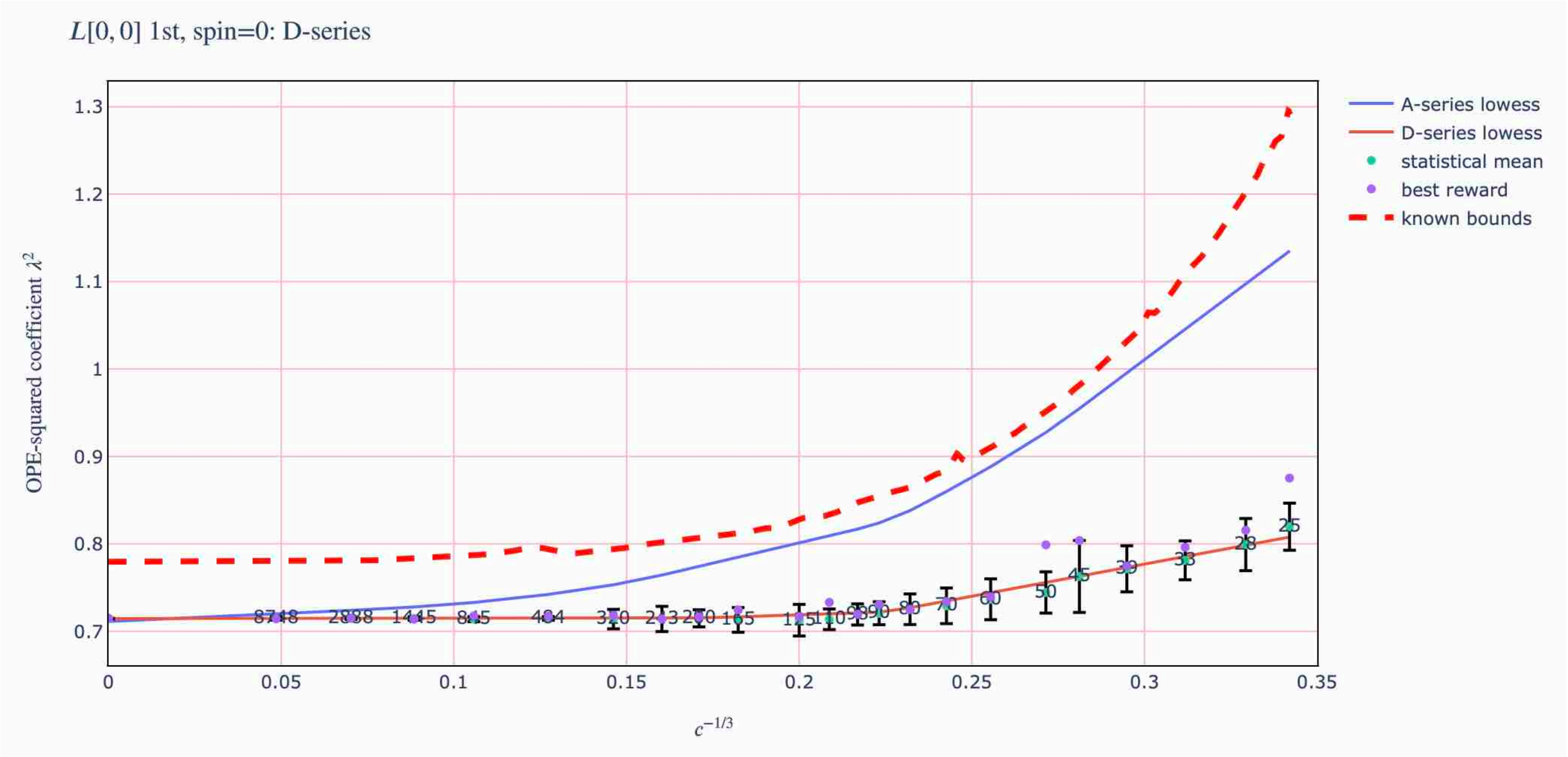
Results (lowest unprotected multiplet “D series”)



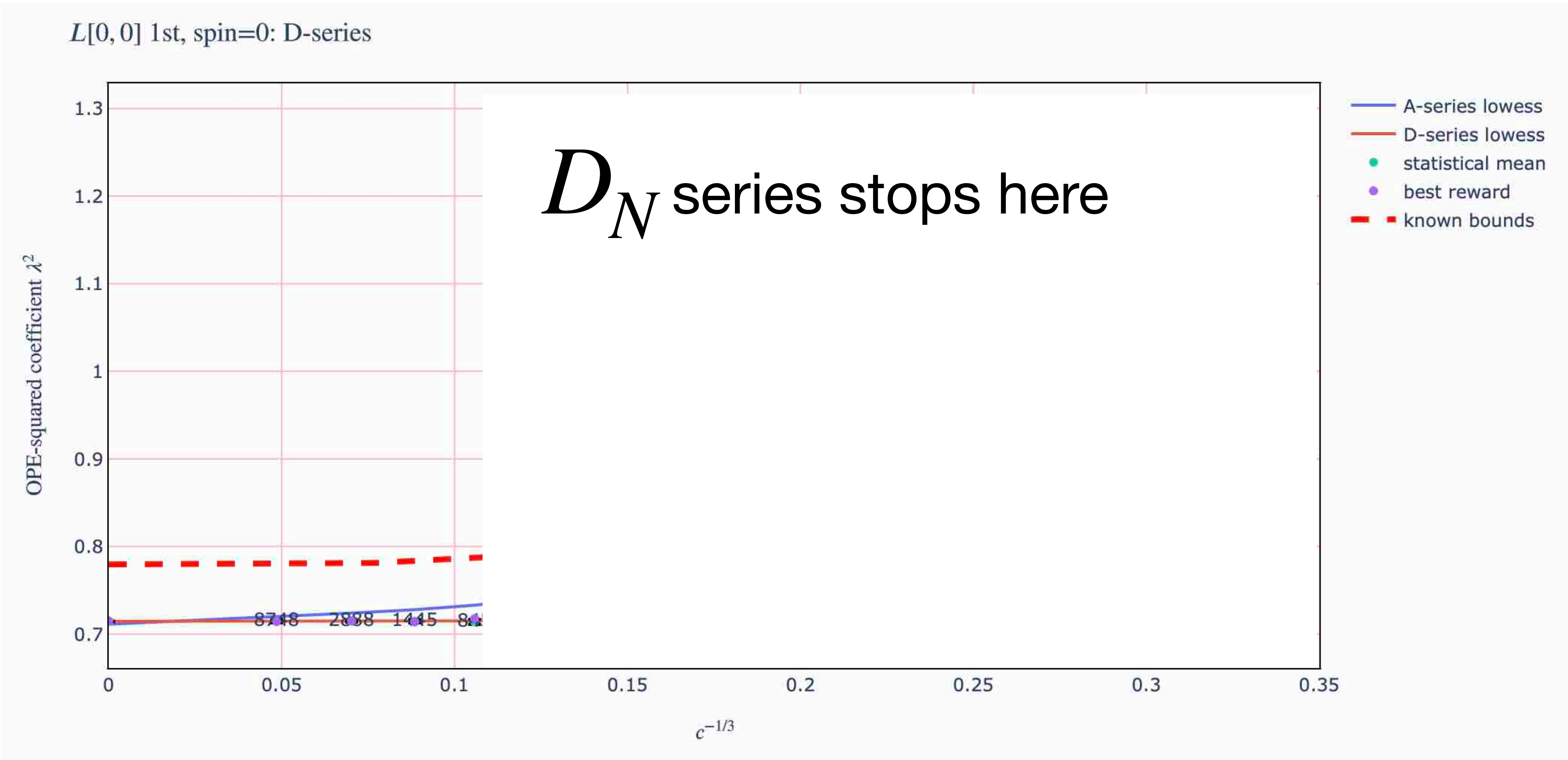
Results (lowest unprotected multiplet “D series”)



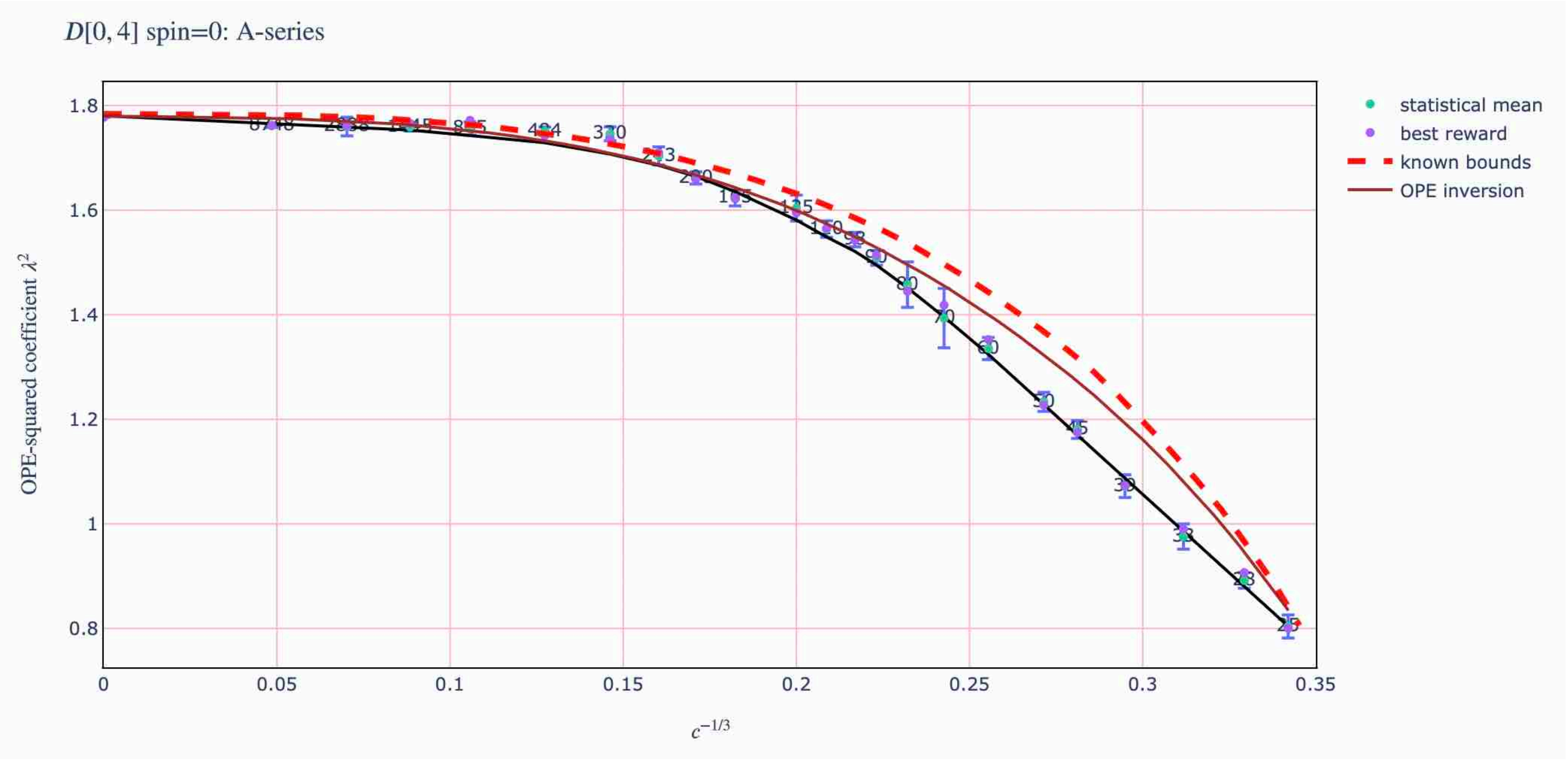
Results (lowest unprotected multiplet “D series”)



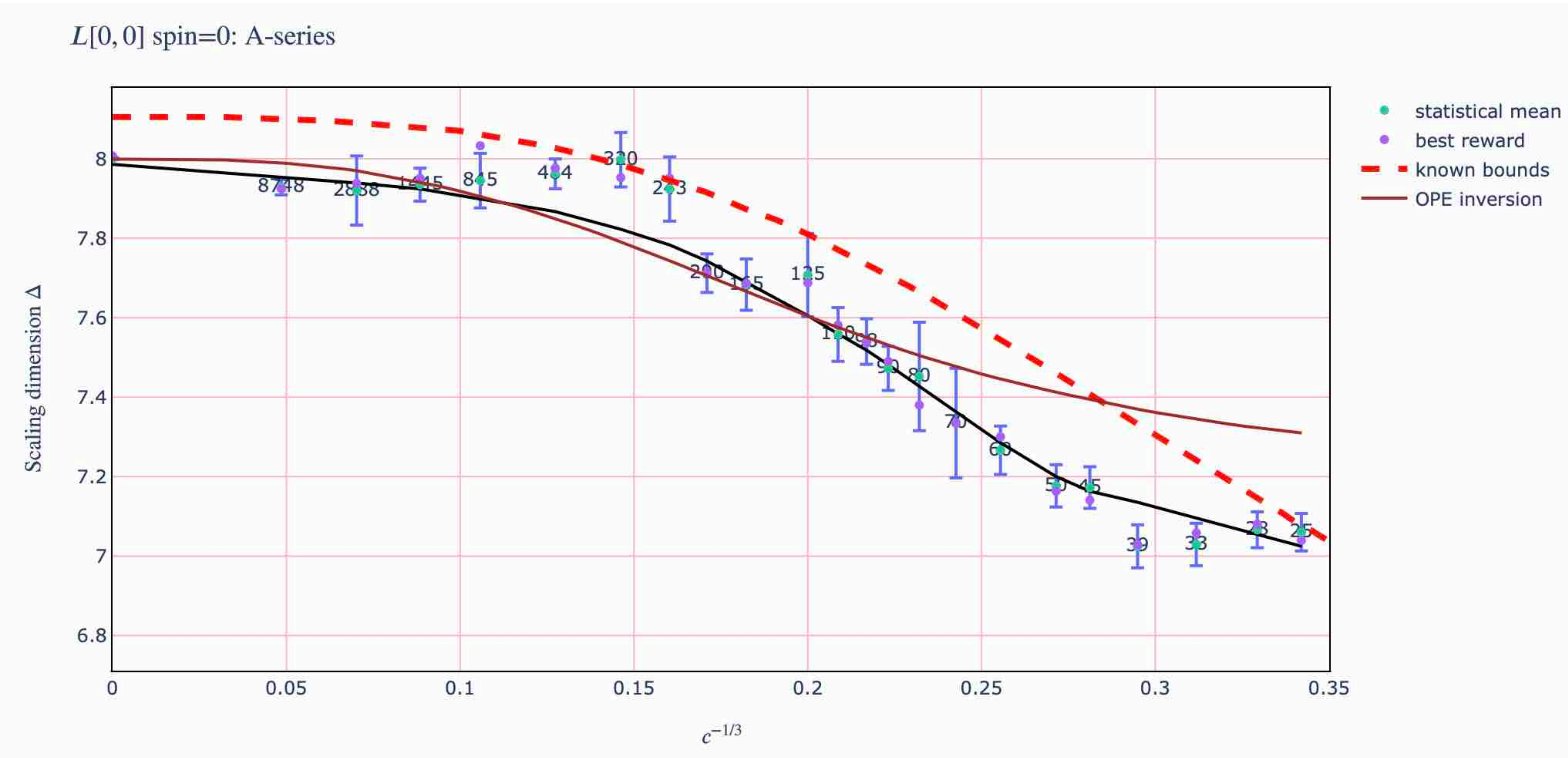
Results (lowest unprotected multiplet “D series”)



Results (lowest protected multiplet “A series”)



Results (lowest unprotected multiplet "$\mathcal{A}$ series")



Results (lowest unprotected multiplet “A series”)