Challenge 01:

AdS$_3$ Pure Gravity via the Modular Bootstrap

Comprehensive Technical Report

Domain:	Quantum Gravity & Particle Physics
Difficulty:	High
Timeline:	6–12 months
Prerequisites:	Conformal field theory, modular forms, semidefinite programming

Pure Thought AI Challenges
January 19, 2026

Contents

1 Executive Summary

This challenge addresses one of the most profound open questions in quantum gravity: **the existence of extremal 2D conformal field theories** at central charges $c = 24k$ for $k > 1$. These theories, if they exist, would provide consistent quantum gravity theories in three-dimensional Anti-de Sitter space (AdS_3) and potentially reveal new connections between number theory, sporadic groups, and physics—extending the famous Monstrous Moonshine correspondence.

Analysis Note

The AdS/CFT correspondence, discovered by Maldacena in 1997, remains one of theoretical physics' most powerful tools. This challenge leverages the special tractability of the AdS_3/CFT_2 case, where infinite-dimensional Virasoro symmetry provides extraordinary computational control. The "modular bootstrap" approach requires *no phenomenological input*—it derives constraints purely from mathematical consistency.

2 Scientific Context and Motivation

2.1 The AdS/CFT Correspondence

The **AdS/CFT correspondence** posits an exact duality between:

- Quantum gravity in $(d + 1)$-dimensional Anti-de Sitter space
- A d-dimensional conformal field theory on the boundary

For $\mathbf{AdS_3/CFT_2}$, this correspondence is particularly tractable due to the infinite-dimensional Virasoro symmetry of 2D CFTs.

Research Direction

Why AdS_3 is special: In three spacetime dimensions, the graviton has no local propagating degrees of freedom—pure gravity is topological. This dramatically constrains the dual CFT, making explicit construction potentially achievable.

2.2 Pure Gravity and Extremal CFTs

Pure gravity in AdS_3 contains only the graviton—no additional matter fields. According to AdS/CFT, such a theory should be dual to an **extremal CFT**: a 2D conformal field theory where the spectrum is *maximally sparse*, containing only:

1. The vacuum state (identity operator)
2. The stress tensor and its Virasoro descendants
3. Primary operators only above a large conformal dimension gap Δ_{gap}

2.3 The Monster CFT and Monstrous Moonshine

For central charge $c = 24$, the extremal theory is **unique** and corresponds to the celebrated **Monster CFT**:

- Partition function given by the modular j-invariant
- Symmetry group is the **Monster group** $\mathbb{M}$—the largest sporadic finite simple group

- Degeneracies encode dimensions of Monster representations: $d(2) = 196884 = 196883 + 1$

Analysis Note

The connection between the j-invariant, the Monster group, and string theory is known as **Monstrous Moonshine**. Discovering extremal CFTs at higher c could reveal *new moonshine phenomena*, potentially connecting to other sporadic groups or novel mathematical structures.

2.4 The Core Question

Central Research Question

Do extremal 2D CFTs exist for central charge $c = 24k$ (with $k > 1$) having only Virasoro primaries below the gap $\Delta_{\mathrm{gap}} \approx c/12$?

- For $k = 1$ ($c = 24$): The Monster CFT provides an explicit example with $\Delta_{\mathrm{gap}} = 2$.
- For $k \geq 2$: **No such theories are known.** Numerical evidence suggests they may not exist for certain gaps.

2.5 Why This Matters

(1) **Existence:** Constructing explicit extremal CFTs would prove pure AdS_3 gravity theories exist at these central charges and potentially reveal new moonshine phenomena.

(2) **Impossibility:** Rigorous no-go theorems would constrain the landscape of quantum gravity theories and support **Swampland conjectures** about which effective theories can be UV-completed.

(3) **Methodology:** The modular bootstrap uses only fundamental axioms (modular invariance, unitarity, integrality)—no phenomenological input required.

(4) **Mathematical Discovery:** Connections between modular forms, sporadic groups, and physics have led to profound discoveries; extremal CFTs at higher c might reveal new instances.

3 Mathematical Formulation

3.1 The Virasoro Algebra

A 2D CFT with central charge c is characterized by its **Virasoro algebra**:

$$\boxed{[L_m, L_n] = (m-n)L_{m+n} + \frac{c}{12}m(m^2-1)\delta_{m+n,0}} \tag{1}$$

Definition 3.1 (Primary Operators)**.** Primary operators $|h\rangle$ satisfy:

$$L_0 |h\rangle = h |h\rangle \quad \text{(conformal dimension } h) \tag{2}$$
$$L_m |h\rangle = 0 \quad \text{for } m > 0 \tag{3}$$

3.2 Virasoro Characters

The **Virasoro character** at conformal dimension h is:

$$\chi_h(q) = \mathrm{Tr}_{\mathcal{V}_h}\, q^{L_0 - c/24} = \frac{q^{h-c/24}}{\prod_{n=1}^{\infty}(1-q^n)} = \frac{q^{h-c/24}}{\eta(\tau)} \tag{4}$$

where:

- $q = e^{2\pi i\tau}$ with τ the modular parameter
- $\eta(\tau)$ is the **Dedekind eta function**

Definition 3.2 (Dedekind Eta Function)**.**

$$\eta(\tau) = q^{1/24}\prod_{n=1}^{\infty}(1-q^n), \quad q = e^{2\pi i\tau} \tag{5}$$

satisfying the modular transformation:

$$\eta\left(-\frac{1}{\tau}\right) = \sqrt{-i\tau}\,\eta(\tau) \tag{6}$$

3.3 The Torus Partition Function

The torus **partition function** is:

$$Z(\tau,\bar{\tau}) = \sum_h d(h)\,|\chi_h(\tau)|^2 \tag{7}$$

where $d(h)$ is the degeneracy of primary operators at conformal dimension h.

For **holomorphic CFTs** (no anti-holomorphic dependence):

$$Z(\tau) = \chi_0(\tau) + \sum_{h>0} d(h)\chi_h(\tau) \tag{8}$$

3.4 Modular Invariance

Theorem 3.1 (Modular Invariance Constraint)**.** The partition function must be invariant under the modular group $\mathrm{PSL}(2,\mathbb{Z}) = \mathrm{SL}(2,\mathbb{Z})/\{\pm I\}$, generated by:

$$S : \tau \mapsto -\frac{1}{\tau} \tag{9}$$

$$T : \tau \mapsto \tau + 1 \tag{10}$$

This requires $Z(\tau) = Z(-1/\tau)$ and $Z(\tau) = Z(\tau+1)$.

The **modular S-transformation** relates characters via:

$$\chi_h\left(-\frac{1}{\tau}\right) = \sum_{h'} S_{h,h'}\chi_{h'}(\tau) \tag{11}$$

For Virasoro characters at large c:

$$S_{h,h'} \approx i\exp\left(-2\pi i\sqrt{hh'}\right) \tag{12}$$

Critical Consideration

The S-matrix approximation above is valid for large c. For exact results, one must compute the S-matrix by evaluating characters at multiple τ points and solving a linear system. Verify that $SS^\dagger = I$ (unitarity) as a consistency check.

3.5 Extremality Condition

Definition 3.3 (Extremal CFT)**.** An extremal CFT has **no primaries in the gap** $(0, \Delta_{\text{gap}})$ except the vacuum:

$$
\begin{aligned}
d(0) &= 1 \quad \text{(vacuum)} && (13)\\
d(h) &= 0 \quad \text{for } 0 < h < \Delta_{\text{gap}} && (14)\\
d(h) &\geq 0 \quad \text{for } h \geq \Delta_{\text{gap}} && (15)
\end{aligned}
$$

For $c = 24k$, a natural gap choice is $\Delta_{\text{gap}} = c/12 = 2k$.

3.6 Optimization Problem Formulation

The modular bootstrap can be cast as a **linear programming (LP)** or **semidefinite programming (SDP)** feasibility problem.

3.6.1 Primal Problem

$$
\begin{aligned}
\text{Find:} \quad & \{d(h) \in \mathbb{Z}_{\geq 0}\} \text{ for } h \geq \Delta_{\text{gap}}\\
\text{Subject to:} \quad & Z(\tau) - Z\left(-\frac{1}{\tau}\right) = 0 \quad \text{(modular invariance)}\\
& d(h) \geq 0 \quad \text{(unitarity)}\\
& d(h) \in \mathbb{Z} \quad \text{(integrality)}
\end{aligned}
\tag{16}
$$

In practice, we truncate the spectrum at some large $h_{\max}$ and solve:

$$
\begin{aligned}
\text{Minimize:} \quad & 0 \quad \text{(feasibility problem)}\\
\text{Variables:} \quad & d(h) \text{ for } h \in \{\Delta_{\text{gap}}, \Delta_{\text{gap}} + 1, \ldots, h_{\max}\}\\
\text{Constraints:} \quad & \text{Modular invariance equations} + d(h) \geq 0
\end{aligned}
\tag{17}
$$

3.6.2 Dual Problem and Certificates

If the primal is infeasible, the LP dual provides a **certificate of impossibility**: a functional $\alpha(h)$ such that:

$$
\sum_h \alpha(h) \cdot [\text{modular constraint}]_h < 0, \quad \alpha(h) \geq 0 \text{ for allowed } h \tag{18}
$$

This proves *mathematically* that no solution exists.

Research Direction

Certificate Verification Strategy: Export dual certificates in SMT-LIB format and verify with Z3 or other SMT solvers. For maximum rigor, formalize the proof in Lean 4 or Isabelle/HOL.

4 Implementation Approach

4.1 Phase 1: Virasoro Characters and Modular Forms (Months 1–2)

Goal: Build a high-precision calculator for Virasoro characters and modular transformations.

4.1.1 Dedekind Eta Function Implementation

```python
from mpmath import mp, exp, pi, sqrt

mp.dps = 150  # 150 decimal places precision

def dedekind_eta(tau: complex) -> complex:
    """
    Compute eta(tau) = q^{1/24} * prod_{n=1}^infty (1 - q^n)

    Uses q-series truncation with error control.
    Error bound: ~ q^{N_max} for truncation at N_max terms.

    Args:
        tau: Modular parameter with Im(tau) > 0

    Returns:
        eta(tau) to mp.dps precision
    """
    q = mp.exp(2 * mp.pi * 1j * tau)

    # Product truncation (error ~ q^{N_max})
    N_max = 100
    product = mp.mpf(1)

    for n in range(1, N_max + 1):
        product *= (1 - q**n)

    eta = q**(mp.mpf(1)/24) * product
    return complex(eta)
```

Listing 1: High-precision Dedekind eta function

Analysis Note

Precision Requirements: The problem requires at least 100 decimal digits of precision to reliably verify modular invariance. Using `mpmath` with `mp.dps = 150` provides a safety margin. The truncation at $N_{\max} = 100$ is justified because $|q|^{100} < 10^{-50}$ for $\mathrm{Im}(\tau) > 0.1$.

4.1.2 Modular Transformation Test

```python
def test_eta_modular():
    """
    Verify eta(-1/tau) = sqrt(-i*tau) * eta(tau)
    """
    tau = 0.3 + 0.5j

    eta_tau = dedekind_eta(tau)
    eta_S_tau = dedekind_eta(-1/tau)

    # Expected relation from modular transformation
    expected = mp.sqrt(-1j * tau) * eta_tau

    error = abs(eta_S_tau - expected)
    assert error < 1e-50, f"Modular check failed: error = {error}"
    print(f"eta modular check PASSED: error = {error:.2e}")
```

Listing 2: Verification of eta modular transformation

4.1.3 Virasoro Character

```python
def virasoro_character(c: float, h: float, tau: complex) -> complex:
    """
    Compute chi_h(tau) = q^{h - c/24} / eta(tau)

    Args:
        c: Central charge
        h: Conformal dimension
        tau: Modular parameter (Im(tau) > 0)

    Returns:
        Character value chi_h(tau)
    """
    q = mp.exp(2 * mp.pi * 1j * tau)
    eta_tau = dedekind_eta(tau)

    chi = q**(h - c/24) / eta_tau
    return complex(chi)
```

Listing 3: Virasoro character computation

4.1.4 Partition Function

```python
from typing import Dict

def partition_function(c: float, spectrum: Dict[float, int],
                       tau: complex) -> complex:
    """
    Compute Z(tau) = chi_0(tau) + sum_h d(h) * chi_h(tau)

    Args:
        c: Central charge
        spectrum: Dictionary {h: d(h)} of degeneracies
        tau: Modular parameter

    Returns:
        Partition function Z(tau)
    """
    # Vacuum contribution
    Z = virasoro_character(c, 0, tau)

    # Sum over primaries
    for h, dh in spectrum.items():
        if h > 0:
            Z += dh * virasoro_character(c, h, tau)

    return Z
```

Listing 4: Partition function from spectrum

4.2 Phase 2: Modular S-Matrix and Constraints (Months 2–3)

Goal: Compute the S-matrix $S_{h,h'}$ and formulate modular invariance as linear equations.

4.2.1 Deriving the Constraint System

Modular invariance $Z(\tau) = Z(-1/\tau)$ gives:

$$\chi_0\left(-\frac{1}{\tau}\right) + \sum_h d(h)\chi_h\left(-\frac{1}{\tau}\right) = \chi_0(\tau) + \sum_h d(h)\chi_h(\tau) \tag{19}$$

Using the S-matrix expansion:

$$\sum_{h'} S_{0,h'}\chi_{h'}(\tau) + \sum_{h} d(h) \sum_{h'} S_{h,h'}\chi_{h'}(\tau) = \chi_0(\tau) + \sum_{h} d(h)\chi_h(\tau) \tag{20}$$

Matching coefficients of $\chi_{h'}(\tau)$ for each h':

$$S_{0,h'} + \sum_{h} d(h)S_{h,h'} = \delta_{h',0} + d(h') \tag{21}$$

Rearranging yields a **linear system**:

$$\boxed{\sum_{h} \left[S_{h,h'} - \delta_{h,h'}\right] d(h) = \delta_{h',0} - S_{0,h'}} \tag{22}$$

```python
import numpy as np

def setup_modular_constraints(c: float, gap: float, h_max: float,
                              h_values: list) -> tuple:
    """
    Set up Ax = b for modular invariance.

    Variables: x = [d(h_1), d(h_2), ..., d(h_N)]
    where h_i in [gap, h_max]

    Constraints: one equation per h' in h_values
    """
    N = len(h_values)

    # Compute S-matrix
    S = compute_s_matrix(c, h_values)

    # Build constraint matrix A and RHS b
    A = np.zeros((N, N), dtype=complex)
    b = np.zeros(N, dtype=complex)

    for i, hp in enumerate(h_values):
        # Equation for h' = hp
        for j, h in enumerate(h_values):
            A[i, j] = S[j, i] - (1 if h == hp else 0)

        # Right-hand side
        b[i] = (1 if hp == 0 else 0) - S[0, i]

    return A, b
```

Listing 5: Setting up modular constraint matrix

4.3 Phase 3: Linear Programming and Optimization (Months 3–4)

Goal: Solve for non-negative integer degeneracies or certify infeasibility.

```python
import cvxpy as cp

def solve_modular_bootstrap_lp(c: float, gap: float,
                               h_max: float) -> dict:
    """
    Solve modular bootstrap as linear program.

    Minimize: 0  (feasibility problem)
    Subject to: A @ d = b, d >= 0
```

```
    """
    h_values = np.arange(gap, h_max + 1, 1.0)
    N = len(h_values)

    A, b = setup_modular_constraints(c, gap, h_max, h_values)

    # Convert to real system (separate real/imaginary parts)
    A_real = np.vstack([A.real, A.imag])
    b_real = np.hstack([b.real, b.imag])

    # Define variables
    d = cp.Variable(N, nonneg=True)

    # Constraints
    constraints = [A_real @ d == b_real]

    # Solve
    problem = cp.Problem(cp.Minimize(0), constraints)
    problem.solve(solver=cp.SCS, verbose=True)

    if problem.status == cp.OPTIMAL:
        spectrum = {h: d.value[i] for i, h in enumerate(h_values)}
        return {'status': 'feasible', 'spectrum': spectrum}
    elif problem.status == cp.INFEASIBLE:
        dual = constraints[0].dual_value
        return {'status': 'infeasible', 'dual_certificate': dual}
    else:
        return {'status': 'unknown'}
```

Listing 6: LP solver for modular bootstrap

Critical Consideration

Integer Constraints: The LP relaxation provides continuous solutions. For physical spectra, degeneracies must be non-negative integers. Use MILP solvers or rounding with verification for exact results.

4.4 Phase 4: Extremal CFT Search at $c = 24k$ (Months 4–6)

Goal: Systematically search for extremal CFTs at $c = 48, 72, 96, \ldots$

4.4.1 Monster CFT Validation ($c = 24$, $k = 1$)

```
def test_monster_cft():
    """
    Verify we recover the Monster CFT at c=24.

    Known spectrum:
    d(1) = 0 (gap at Delta=2)
    d(2) = 196884
    d(3) = 21493760
    d(4) = 864299970
    """
    c = 24
    gap = 2
    h_max = 10

    result = solve_modular_bootstrap_milp(c, gap, h_max)
    assert result['status'] == 'feasible'

```

```python
    # Check first few degeneracies
    monster_spectrum = {
        2: 196884,
        3: 21493760,
        4: 864299970
    }

    for h, d_expected in monster_spectrum.items():
        d_computed = result['spectrum'][h]
        assert abs(d_computed - d_expected) < 1
        print(f"d({h}) = {d_computed} (expected {d_expected})")

    print("Monster CFT validation: PASSED")
```

Listing 7: Validation against Monster CFT

Analysis Note

Why Monster Validation is Critical: The Monster CFT provides a known solution against which to test all numerical infrastructure. If the solver fails to reproduce $d(2) = 196884$ exactly, there is a bug in the implementation. Do not proceed to $k \geq 2$ until this passes.

4.5 Phase 5: Dual Certificates and Impossibility Proofs (Months 6–8)

Goal: Extract machine-verifiable certificates when no solution exists.

```python
def extract_dual_certificate(c: float, gap: float,
                             h_max: float) -> np.ndarray:
    """
    Solve dual LP to get impossibility certificate.

    Dual problem:
    Maximize: b^T y
    Subject to: A^T y <= 0

    If dual is unbounded, primal is infeasible.
    """
    h_values = np.arange(gap, h_max + 1, 1.0)
    A, b = setup_modular_constraints(c, gap, h_max, h_values)

    A_real = np.vstack([A.real, A.imag])
    b_real = np.hstack([b.real, b.imag])

    M = A_real.shape[0]
    y = cp.Variable(M)

    objective = cp.Maximize(b_real @ y)
    constraints = [A_real.T @ y <= 0]

    problem = cp.Problem(objective, constraints)
    problem.solve()

    if problem.status == cp.OPTIMAL and problem.value > 1e-6:
        return y.value
    return None
```

Listing 8: Dual certificate extraction

4.6 Phase 6: Formal Verification (Months 8–12)

Goal: Formalize results in Lean 4 for machine-checked proofs.

```
import Mathlib.Analysis.Complex.Basic
import Mathlib.LinearAlgebra.Matrix.Spectrum

-- Define Virasoro character
def virasoro_character (c h : Real) (tau : Complex) : Complex := sorry

-- Modular invariance axiom
axiom modular_invariance (c : Real) (Z : Complex -> Complex) :
  (forall tau, Z tau = Z (-1/tau)) -> ModularInvariant Z

-- Extremal CFT theorem
theorem no_extremal_cft_c48_gap4 :
  forall (spectrum : Real -> Nat),
    (forall h, 0 < h and h < 4 -> spectrum h = 0) ->  -- gap
    (forall h, spectrum h >= 0) ->                    -- unitarity
    not (ModularInvariant (partition_function 48 spectrum)) := by
  intro spectrum h_gap h_unit
  -- Proof using dual certificate
  sorry
```

Listing 9: Lean 4 formalization template

5 Detailed Research Directions

5.1 Direction 1: Systematic k-Scan

Research Direction

Approach: For each $k = 2, 3, 4, \ldots, 10$, solve the modular bootstrap at $c = 24k$ with gap $\Delta_{\text{gap}} = c/12 = 2k$. Record feasibility status and extract certificates.
Expected Outcome: A phase diagram in (c, Δ_{gap}) space showing regions of existence vs. impossibility.
Novel Contribution: If any extremal CFT is found for $k \geq 2$, this would be a major discovery potentially revealing new moonshine phenomena.

5.2 Direction 2: Gap Variation Study

The natural gap $\Delta_{\text{gap}} = c/12$ is not the only interesting choice:

- **Smaller gaps** ($\Delta < c/12$): May be more feasible; would correspond to CFTs with additional light operators
- **Larger gaps** ($\Delta > c/12$): Stronger constraint; if possible, would give "super-extremal" CFTs

Research Direction

Study: For fixed $c = 48$, scan over gaps $\Delta \in \{2, 2.5, 3, 3.5, 4, 4.5, 5\}$ and map the feasibility boundary.

5.3 Direction 3: Symmetry Enhancement

If an extremal CFT is found, investigate its symmetry group:

1. Compute the graded dimension $\dim_h = d(h)$ for low-lying states
2. Check if dimensions match irreducible representations of known groups
3. Look for sporadic group candidates (Conway groups, Baby Monster, etc.)

5.4 Direction 4: Holographic Interpretation

For any found extremal CFT, interpret in the AdS_3 gravity dual:

- Gap Δ_{gap} corresponds to the mass of the lightest primary operator
- In AdS units: $m^2L^2 = \Delta(\Delta - 2)$ for $\Delta > 1$
- Compare with BTZ black hole threshold and cosmic censorship bounds

5.5 Direction 5: Connection to Quantum Error Correction

Recent work connects extremal CFTs to quantum error-correcting codes:

Research Direction

Investigation: If an extremal CFT exists at $c = 24k$, can its partition function be interpreted as a weight enumerator of a quantum stabilizer code? This could provide a "code-theoretic" construction of the CFT.

6 Success Criteria

6.1 Minimum Viable Result (3–4 months)

✓ Virasoro character calculator accurate to 100+ decimal digits

✓ Modular S-transformation verified numerically with error $< 10^{-50}$

✓ Monster CFT spectrum reproduced: $d(2) = 196884$, $d(3) = 21493760$, $d(4) = 864299970$

✓ **One new rigorous result** at $c = 48$: either feasible spectrum or impossibility certificate

6.2 Strong Result (6–8 months)

✓ Complete results for $k = 2, 3, 4$ ($c = 48, 72, 96$)

✓ All certificates in machine-verifiable format (JSON/SMT-LIB)

✓ Independent verification by Z3 or external LP solver

✓ Phase diagram initiated with gap scans

6.3 Publication-Quality Result (9–12 months)

✓ Results for k up to 10 (c up to 240)

✓ Formal verification in Lean 4 of key impossibility theorems

✓ Novel extremal CFTs discovered (if they exist) with symmetry analysis

✓ ArXiv preprint with public certificate repository

7 Verification Protocol

7.1 For Claimed Feasibility (Extremal CFT Exists)

```python
def verify_extremal_cft(c: float, gap: float, spectrum: dict,
                        tau_samples: list = None) -> dict:
    """
    Comprehensive verification of extremal CFT spectrum.
    """
    if tau_samples is None:
        tau_samples = [0.05 + 0.5j, 0.2 + 0.8j, 0.5 + 1.0j,
                       -0.4 + 0.7j, 0.3 + 1.2j]

    results = {
        'integrality_passed': True,
        'unitarity_passed': True,
        'gap_passed': True,
        'modular_invariance_passed': True,
        'max_error': 0.0
    }

    # 1. Check integrality
    for h, d in spectrum.items():
        if not isinstance(d, int) or d < 0:
            results['integrality_passed'] = False

    # 2. Check gap condition
    if any(0 < h < gap for h in spectrum.keys()):
        results['gap_passed'] = False

    # 3. Check modular invariance
    for tau in tau_samples:
        Z_tau = partition_function(c, spectrum, tau)
        Z_S_tau = partition_function(c, spectrum, -1/tau)
        error = abs(Z_tau - Z_S_tau)
        results['max_error'] = max(results['max_error'], error)
        if error > 1e-30:
            results['modular_invariance_passed'] = False

    results['status'] = 'VERIFIED' if all([
        results['integrality_passed'],
        results['gap_passed'],
        results['modular_invariance_passed']
    ]) else 'FAILED'

    return results
```

Listing 10: Comprehensive verification function

7.2 For Claimed Infeasibility

1. Verify dual certificate satisfies $A^T y \leq 0$
2. Verify $b^T y > 0$ (proves primal infeasibility via Farkas lemma)
3. Export to SMT-LIB and verify with Z3: `z3 certificate_c48_gap4.smt2`
4. Formalize in Lean 4 for machine-checked proof

8 Common Pitfalls and Mitigations

8.1 Numerical Precision Issues

Critical Consideration

Problem: Modular invariance appears satisfied due to rounding errors, leading to false positives.
Solution:

- Use `mpmath` with `mp.dps = 150` or higher
- Verify modular invariance to at least 50 decimal digits
- Cross-check with multiple τ points in fundamental domain

8.2 Non-Integer Solutions from LP Relaxation

Critical Consideration

Problem: LP solution has $d(h) = 123.7$ (non-integer) accepted as valid.
Solution:

- Always enforce strict integrality using MILP
- If rounding LP solution, verify all constraints still satisfied
- Export only exact integer degeneracies

8.3 Truncation Effects

Critical Consideration

Problem: Setting $h_{\max}$ too small misses important high-dimension operators.
Solution:

- Start with $h_{\max} = c$ (usually sufficient)
- Check sensitivity: increase $h_{\max}$ and verify solution stability
- Use asymptotic bounds on degeneracies to justify truncation

9 Resources and References

9.1 Essential Papers

1. Hartman, Keller, Stoica (2014): "Universal Spectrum of 2d Conformal Field Theory in the Large c Limit" [arXiv:1405.5137]

2. Afkhami-Jeddi, Cohn, Hartman, Tajdini (2020): "Free Partition Functions and an Averaged Holographic Duality" [arXiv:2006.04839]

3. Collier, Lin, Yin (2019): "Modular Bootstrap Revisited" [arXiv:1608.06241]

4. Hellerman (2011): "A Universal Inequality for CFT and Quantum Gravity" [arXiv:0902.2790]

5. Friedan, Keller, Yin (2013): "A Remark on AdS/CFT for the Extremal Virasoro Partition Function" [arXiv:1312.1536]

9.2 Code Libraries

- **mpmath:** Arbitrary precision arithmetic — `pip install mpmath`
- **SymPy:** Symbolic mathematics — `pip install sympy`
- **CVXPY:** Convex optimization with SDP/LP solvers — `pip install cvxpy`
- **SciPy:** Scientific computing including MILP — `pip install scipy`
- **Lean 4:** Proof assistant — `https://lean-lang.org`

9.3 Mathematical Background

- **Modular Forms:** Serre's "A Course in Arithmetic"; Diamond & Shurman "A First Course in Modular Forms"
- **Virasoro Algebra:** Di Francesco et al. "Conformal Field Theory" (Yellow Book)
- **Optimization:** Boyd & Vandenberghe "Convex Optimization" (LP duality chapter)
- **AdS/CFT:** Aharony et al. "Large N Field Theories, String Theory and Gravity" [arXiv:hep-th/9905111]

10 Milestone Checklist

10.1 Infrastructure (Months 1–2)

- ☐ Dedekind eta function $\eta(\tau)$ implemented with 100+ digit precision
- ☐ Modular transformation $\eta(-1/\tau) = \sqrt{-i\tau}\,\eta(\tau)$ verified
- ☐ Virasoro character $\chi_h(\tau)$ calculator tested
- ☐ Partition function $Z(\tau)$ builder with arbitrary spectrum
- ☐ Modular S-matrix computed and verified for unitarity

10.2 Validation (Month 2)

- ☐ Monster CFT ($c = 24$, gap$= 2$) reproduced exactly:
  - ☐ $d(2) = 196884$
  - ☐ $d(3) = 21493760$
  - ☐ $d(4) = 864299970$
- ☐ Modular invariance verified to 10^{-50}

10.3 Optimization Solvers (Months 2–3)

- ☐ LP relaxation solver (cvxpy) working
- ☐ MILP solver enforcing integrality
- ☐ Dual certificate extraction implemented
- ☐ Certificate verification functions tested

10.4 New Results (Months 3–6)

- ☐ $c = 48$, gap$= 4$: Result obtained
- ☐ $c = 48$ result independently verified
- ☐ $c = 72$, gap$= 6$: Result obtained
- ☐ $c = 96$, gap$= 8$: Result obtained
- ☐ All certificates exported

10.5 Classification (Months 6–9)

- ☐ Phase diagram for $k = 1$ to 5 complete
- ☐ Gap scan: multiple Δ values tested
- ☐ Database of spectra and impossibility proofs
- ☐ Visualization: (c, Δ) phase diagram

10.6 Formal Verification (Months 9–12)

- ☐ Lean 4 formalization begun
- ☐ First impossibility theorem formalized
- ☐ All certificates verified in proof assistant
- ☐ Publication draft prepared

11 Conclusion

The modular bootstrap approach to extremal CFTs represents a frontier research problem combining deep mathematics (modular forms, sporadic groups) with modern computational techniques (semidefinite programming, formal verification). Success would either:

1. **Discover new extremal CFTs**, potentially revealing new moonshine phenomena and confirming the existence of pure AdS_3 gravity at higher central charges; or
2. **Prove impossibility theorems**, constraining the Swampland and demonstrating which effective theories cannot be UV-completed into quantum gravity.

Either outcome constitutes a significant contribution to theoretical physics and mathematics.