A new perspective on flux vacua

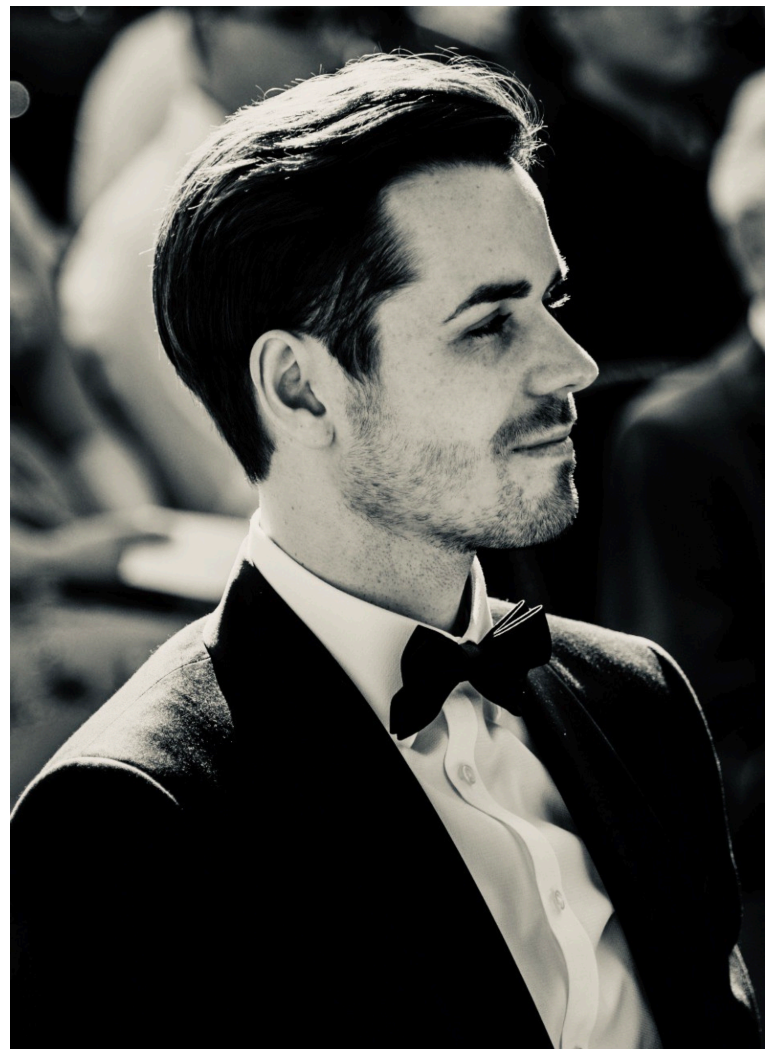
3.7.2023, String Phenomenology 2023 Korea
Sven Krippendorf (sven.krippendorf@physik.uni-muenchen.de, @krippendorfsven)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Based on: 2306.06160, 2307.xxxxx, 2307.xxxxx
Work in collaboration with:

Abhishek Dubey

Julian Ebelt



Andreas Schachner

Talk on Thursday

20 years ago …

20 years ago …

The statistics of string/M theory vacua

Michael R. Douglas

Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08855-0849 USA

I.H.E.S.†, Le Bois-Marie, Bures-sur-Yvette, 91440 France

We discuss systematic approaches to the classification of string/M theory vacua, and physical questions this might help us resolve. To this end, we initiate the study of ensembles of effective Lagrangians, which can be used to precisely study the predictive power of string theory, and in simple examples can lead to universality results. Using these ideas, we outline an approach to estimating the number of vacua of string/M theory which can realize the Standard Model.

20 years ago …

The statistics of string/M theory vacua

Michael R. Douglas

Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08855-0849 USA

I.H.E.S.[†], Le Bois-Marie, Bures-sur-Yvette, 91440 France

We discuss systematic approaches to the classification of string/M theory vacua, and physical questions this might help us resolve. To this end, we initiate the study of ensembles of effective Lagrangians, which can be used to precisely study the predictive power of string theory, and in simple examples can lead to universality results. Using these ideas, we outline an approach to estimating the number of vacua of string/M theory which can realize the Standard Model.

Profound results with assumptions, e.g.:

$$\left\langle W(z_1)W^*(\bar{z}_2)\right\rangle = e^{-K(z_1,\bar{z}_2)},$$

Although the ensembles we considered are somewhat crude, we can progress by formulating better ones which try to reflect more of the structure of the problem, and test our hypothesized ensembles against statistics of sample sets of string vacua constructed either systematically, or by choosing random examples and doing detailed model-by-model analysis. By finding better ensembles, we will be improving our understanding of the distribution of string/M theory vacua in a relatively concrete way. One might think of the structure of a good ensemble as capturing a "stringy" concept of naturalness, which could improve on traditional ideas of naturalness in guiding string phenomenology.

20 years ago …

The statistics of string/M theory vacua

Michael R. Douglas

Department of Physics and Astronomy, Rutgers University, Piscataway, NJ 08855-0849 USA

I.H.E.S.†, Le Bois-Marie, Bures-sur-Yvette, 91440 France

We discuss systematic approaches to the classification of string/M theory vacua, and physical questions this might help us resolve. To this end, we initiate the study of ensembles of effective Lagrangians, which can be used to precisely study the predictive power of string theory, and in simple examples can lead to universality results. Using these ideas, we outline an approach to estimating the number of vacua of string/M theory which can realize the Standard Model.

Profound results with assumptions, e.g.:

$$\left\langle W(z_1)W^*(\bar{z}_2)\right\rangle = e^{-K(z_1,\bar{z}_2)},$$

Today: *NEW* route to such ensembles of string solutions. *We can get actual ensembles of flux vacua in CY constructions such as Kreuzer-Skarke.*

Although the ensembles we considered are somewhat crude, we can progress by formulating better ones which try to reflect more of the structure of the problem, and test our hypothesized ensembles against statistics of sample sets of string vacua constructed either systematically, or by choosing random examples and doing detailed model-by-model analysis. By finding better ensembles, we will be improving our understanding of the distribution of string/M theory vacua in a relatively concrete way. One might think of the structure of a good ensemble as capturing a "stringy" concept of naturalness, which could improve on traditional ideas of naturalness in guiding string phenomenology.

“10^{500}” what’s their physics?

- Which relevant scales (masses, string coupling, scale of supersymmetry breaking, regimes for successful uplifts [cf. Hebecker’s talk])?
- Universal features (distribution of physical properties in a single geometry and in ensembles of geometries)?
- Do minima satisfying all UV constraints actually exist?

We actually know very little about this even after 20 years, because we have not been able to look.

Status of actual ensembles

No systematic analysis due to methodological limitations

Few examples:

- Small $|W_0|$ (Demirtas et al. 1912.10047, Alvarez et al. 2009.03325)
- P11226 (Conlon et al. 0502058), P11169 (Martinez-Pedrera et al. 1212.4530), some of our examples using homotopy methods (Cicoli et al. 1312.0014)
- Analytic approximations: Coudarchet et al. (2212.02533)
- …

What do we need?

Easily obtain flux vacua relevant for our physics questions:

- Many geometries, different regions of moduli space
- Different questions (e.g. SUSY, non SUSY vacua)
- Many samples (statistics, dedicated search algorithms)

$\rightarrow$ **JAXvacua**

2306.06160

Which flux solutions?

- CY orientifold compactifications of Type IIB to 4D $\mathcal{N} = 1$ supergravity concentrating on
  - $h_-^{1,2}$ complex structure moduli Z^i ,
  - axio-dilaton τ.
- Prepotentials at large complex structure (LCS) (Hosono et al. hep-th/9403096,hep-th/9406055); moduli values in Kähler cone

$$F = -\frac{1}{6}\kappa_{ijk} Z^i Z^j Z^k + \frac{1}{2} a_{ij} Z^i Z^j + b_i Z^i + \frac{\mathrm{i}}{2}\tilde{\xi} + F_{\text{inst}}(Z) ,$$

$$F_{\text{inst}}(Z) = -\frac{1}{(2\pi\,\mathrm{i})^3} \sum_{q\in\mathcal{M}(\tilde{X}_3)} n_q^{(0)} \mathrm{Li}_3\left(\exp^{2\pi\,\mathrm{i}\, q_i Z^i}\right) , \quad \mathrm{Li}_3(x) = \sum_{m=1}^{\infty} \frac{x^m}{m^3}$$

- Kähler potential and flux superpotential:

$$K = -\log[-i(\tau-\bar{\tau})] - \log(-\mathrm{i}\,\Pi^\dagger\cdot\Sigma\cdot\Pi) , \quad W = \left(f-\tau h\right)^T \cdot\Sigma\cdot\Pi(Z) ,$$

$$\Pi = \left(2F - Z^i F_i, F_i, 1, Z^i\right)^T , \quad F_i = \partial_{Z^i} F , \quad \Sigma = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} .$$

- $f, h \in \mathbb{Z}^{2(h^{1,2}+1)}$ are choices of integer 3-form fluxes satisfying tadpole constraint.
- The 4D F-term scalar potential for the fluxes reads

$$V_{\text{Flux}} = e^K \left(K^{\tau\bar{\tau}} D_\tau W D_{\bar{\tau}} \overline{W} + K^{i\bar{j}} D_i W D_{\bar{j}} \overline{W}\right) , \quad D_I W = \partial_I W + W\partial_I K$$

We are interested in minima of this system:

A) SUSY: $D_I W = 0$

B) Non-SUSY: $D_I W \neq 0$

Which models?

CY orientifold data — interfacing with previous work (CYTOOLS)

We work with mirror pairs of CY_3 hypersurfaces $X_3, \tilde{X}_3$

in toric varieties $V_4, \tilde{V}_4$

obtained from triangulations of 4D polytopes Δ°, Δ

473,800,776 reflexive polytopes in 4D

Kreuzer, Skarke (KS) `[hep-th/0002240]`

CY data — our input

$\kappa_{ijk}\,, c_2\,, GVs$

We construct smooth orientifolds with $h^{1,1}_- = h^{1,2}_+ = 0$ following `[Jefferson, Kim 2211.00210, Moritz 2305.06363]`

We compute GV data for models with $h^{1,2}(X_3) \leq 25$ using the algorithm of `[Demirtas et al. 2303.00757]`

Computations performed with

CY Tools

http://cy.tools

`Demirtas, Rios-Tascon, McAllister 2211.03823`

Model construction
CY orientifold data from computational tools

↓

EFT module
Auto-diff to construct SUGRA equations from prepotential

↓

Sampling module
Choice of initial guesses for moduli and fluxes

↓

Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

↓

Filter module
Check for vacua and consistent LCS truncation

JAXvacua

EFT module

- Flexible code (i.e. re-use for different CY data) for EFT properties with JAX

CY data $\kappa_{ijk}\,, c_2\,, GVs$ → **Prepotential** F (Only hardcoded input!) → auto-diff → **Periods** Π, K, W → auto-diff → **EFT properties** $V\,, \partial_I\partial_J V\,, M_{3/2}\,, \ldots$

- Auto-diff: machine precision derivatives; easy to implement and adapt to different properties.

Code example:

```
def prepot(moduli):

    return np.einsum("ijk,i,j,k",kappa,moduli,moduli,moduli) + ...

gradient_prepot = jax.grad(prepot,holomorphic=True)(moduli)
```

On paper: $F = -\frac{1}{6}\kappa_{ijk}\, Z^i\, Z^j\, Z^k + \ldots\ , \quad \partial_i F = -\frac{1}{2}\kappa_{ijk}\, Z^j\, Z^k + \ldots$

Model construction
CY orientifold data from computational tools

↓

EFT module
Auto-diff to construct SUGRA equations from prepotential

↓

Sampling module
Choice of initial guesses for moduli and fluxes

↓

Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

↓

Filter module
Check for vacua and consistent LCS truncation

JAXvacua

Making things fast: compiled and vectorised code

- Jit automatically generates C++ code during the first evaluation, executable on CPU/GPU/TPU

Code example:

```
import jax
import jax.numpy as jnp

def selu(x, alpha=1.67, lambda_=1.05):
  return lambda_ * jnp.where(x > 0, x, alpha * jnp.exp(x) - alpha)

x = jnp.arange(1000000)
%timeit selu(x).block_until_ready()
```

```
100 loops, best of 5: 2.05 ms per loop
```

```
selu_jit = jax.jit(selu)

# Warm up
selu_jit(x).block_until_ready()

%timeit selu_jit(x).block_until_ready()
```

```
10000 loops, best of 5: 150 µs per loop
```

Model construction
CY orientifold data from computational tools

EFT module
Auto-diff to construct SUGRA equations from prepotential

Sampling module
Choice of initial guesses for moduli and fluxes

Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

Filter module
Check for vacua and consistent LCS truncation

JAXvacua

Making things fast: compiled and vectorised code

- Jit automatically generates C++ code during the first evaluation, executable on CPU/GPU/TPU
- Vmap automatically vectorises and parallelises code (CPU/GPU)

Advantages:

- Avoids manual rewriting of functions which is typically rigid and messy
- Can be used flexibly depending on the purpose
- Huge speed up

Code example:

```
hessian_values = [model.scalar_potential_hessian(moduli, tau, flux) for moduli,tau,flux in enumerate(data)]
```

```
Time to compute Hessian values sequentially:  7.26
```

```
hessian_vmap = vmap(model.scalar_potential_hessian,axis=(0,0,0))
hessian_values = hessian_vmap(moduli, tau, flux)
```

```
Time to compute Hessian values with vmap:  0.07
```

Difference between evaluating the Hessian for $O(10^4)$ solutions in sequence or by using vmap and evaluate on all solutions at once.

Model construction
CY orientifold data from computational tools

EFT module
Auto-diff to construct SUGRA equations from prepotential

Sampling module
Choice of initial guesses for moduli and fluxes

Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

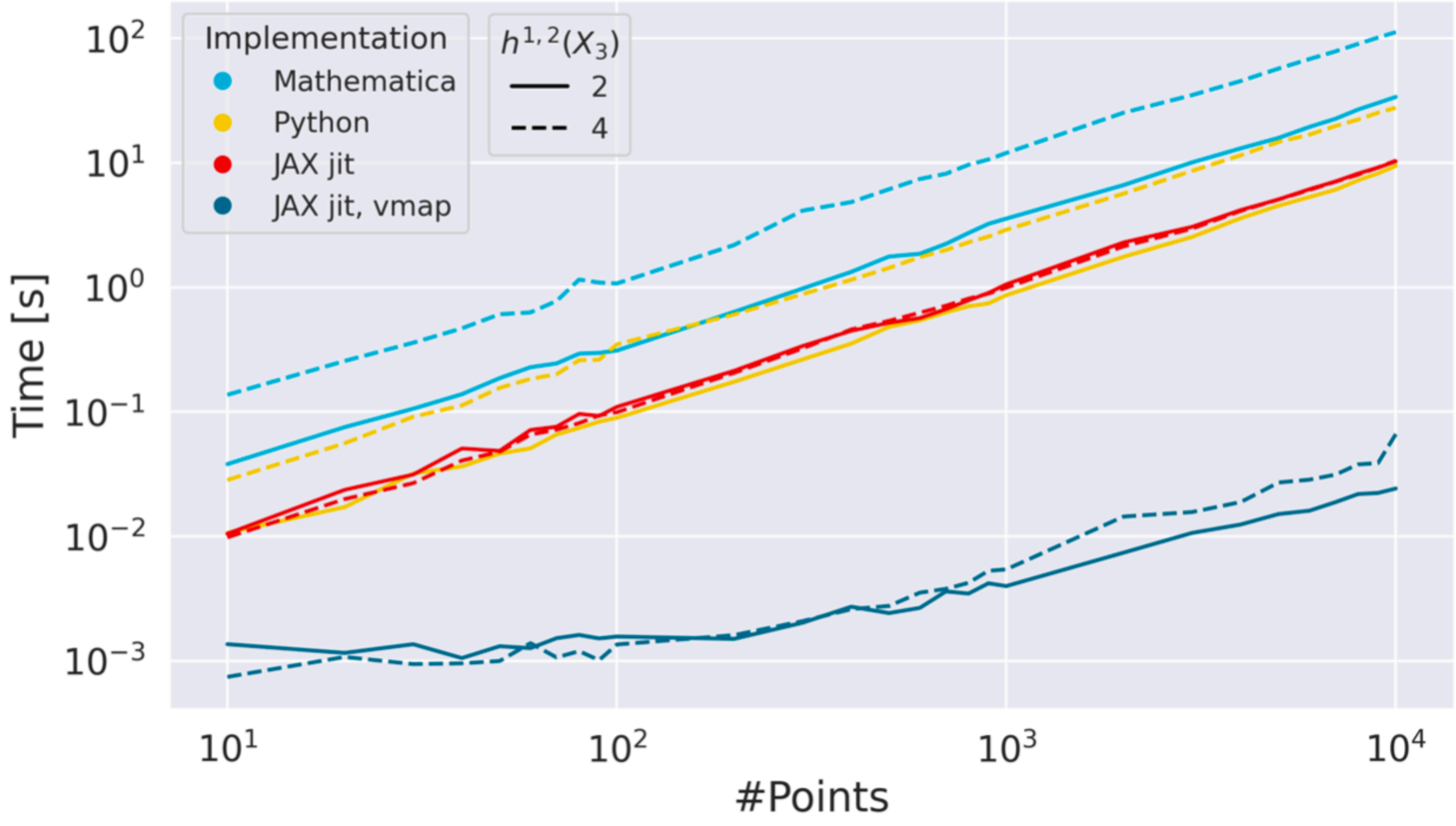
Filter module
Check for vacua and consistent LCS truncation

JAXvacua

Making things fast: compiled and vectorised code

- Jit automatically generates C++ code during the first evaluation, executable on CPU/GPU/TPU
- Vmap automatically vectorises and parallelises code (CPU/GPU)

Timing for evaluating $D_I W$



Orders of magnitude speed improvements!

Implementation is

- **completely modular:** approach easily generalisable
  - ✳ to more general flux vacua including e.g. conifolds (wip)
  - ✳ to include Kähler moduli (wip)

Model construction
CY orientifold data from computational tools

↓

EFT module
Auto-diff to construct SUGRA equations from prepotential

↓

Sampling module
Choice of initial guesses for moduli and fluxes

↓

Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

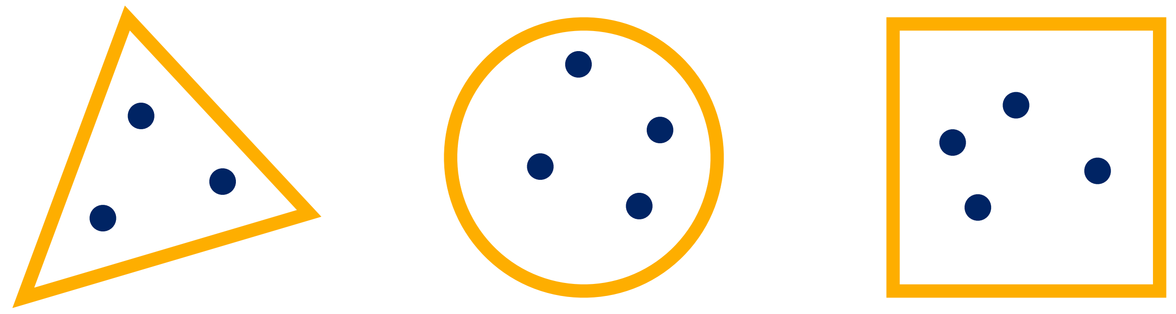
↓

Filter module
Check for vacua and consistent LCS truncation

JAXvacua

Sampling module

Initial guesses for moduli VEVs



Geometry:	Cone	Sphere	Box
Use-cases:	LCS	Conifold	Axions

Flux choices

Fluxes f, h uniformly sampled from

$$f, h \in [-L, L]^{2(h^{1,2}+1)}$$

(in practice $L \lesssim 10$) and we impose typically

$$N_{flux} \leq Q_{D3}$$

ISD sampling:

We sample **half of the fluxes** plus initial points $Z_0^i\,,\,\tau_0$ together.

Then, the remaining fluxes are fixed by the ISD condition

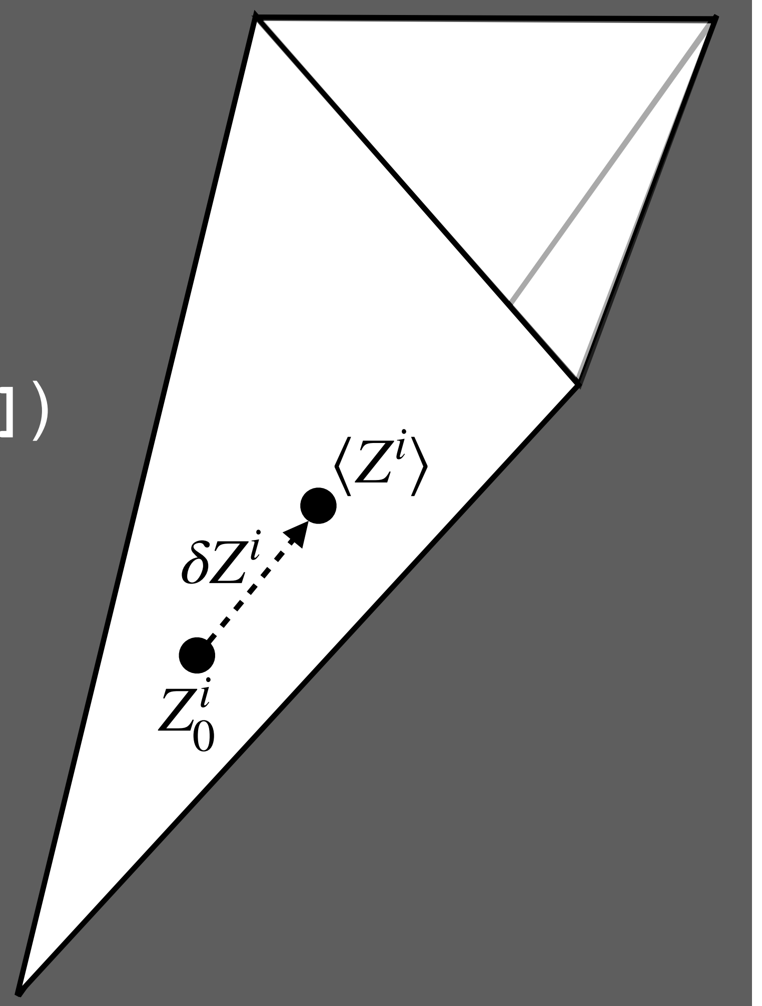
$$ISD_+ : \quad \tilde{m}_J - \tau_0\,\tilde{n}_J = \overline{\mathcal{N}}_{JI}\left(m^I - \tau_0\,n^I\right)$$ (also [Tsagkaris, Plauschinn 2207.13721])

$$ISD_- : \quad \tilde{m}^I - \tau_0\,\tilde{n}^I = \overline{\mathcal{N}}^{IJ}\left(m_J - \tau_0\,n_J\right)$$

where we sample the RHS. In general, rounding is necessary

$$\tilde{m}\,,\tilde{n} \in \mathbb{R} \xrightarrow{\text{rounding}} m\,,n \in \mathbb{Z}$$ (ISD no longer satisfied → optimisation module)

Effective sampling method with generically higher success rate.



Model construction
CY orientifold data from computational tools

↓

EFT module
Auto-diff to construct SUGRA equations from prepotential

↓

Sampling module
Choice of initial guesses for moduli and fluxes

↓

Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

↓

Filter module
Check for vacua and consistent LCS truncation

JAXvacua

Optimisation module

Optimisation targets:

$\partial_I V = 0$

$D_I W = 0$
Dubey, SK, Schachner [2306.06160]

$D_I W \neq 0$
SK, Schachner [2307.XXXX]

- Currently, we employ `scipy.optimize.root` together with multiprocessing!
- We are implementing new optimisers that are adapted to JAX parallelisation scheme (wip)

Model construction
CY orientifold data from computational tools

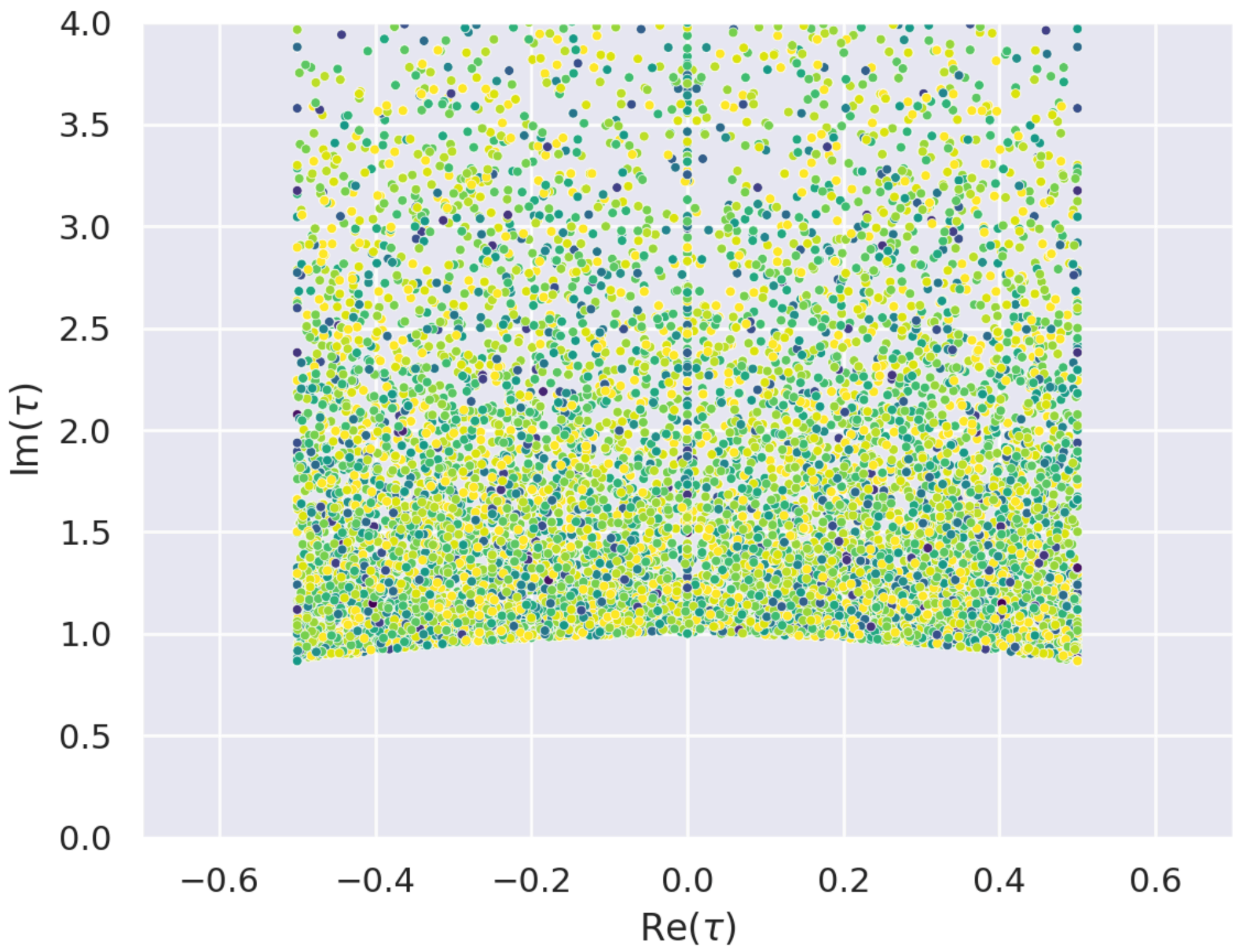
EFT module
Auto-diff to construct SUGRA equations from prepotential

Sampling module
Choice of initial guesses for moduli and fluxes

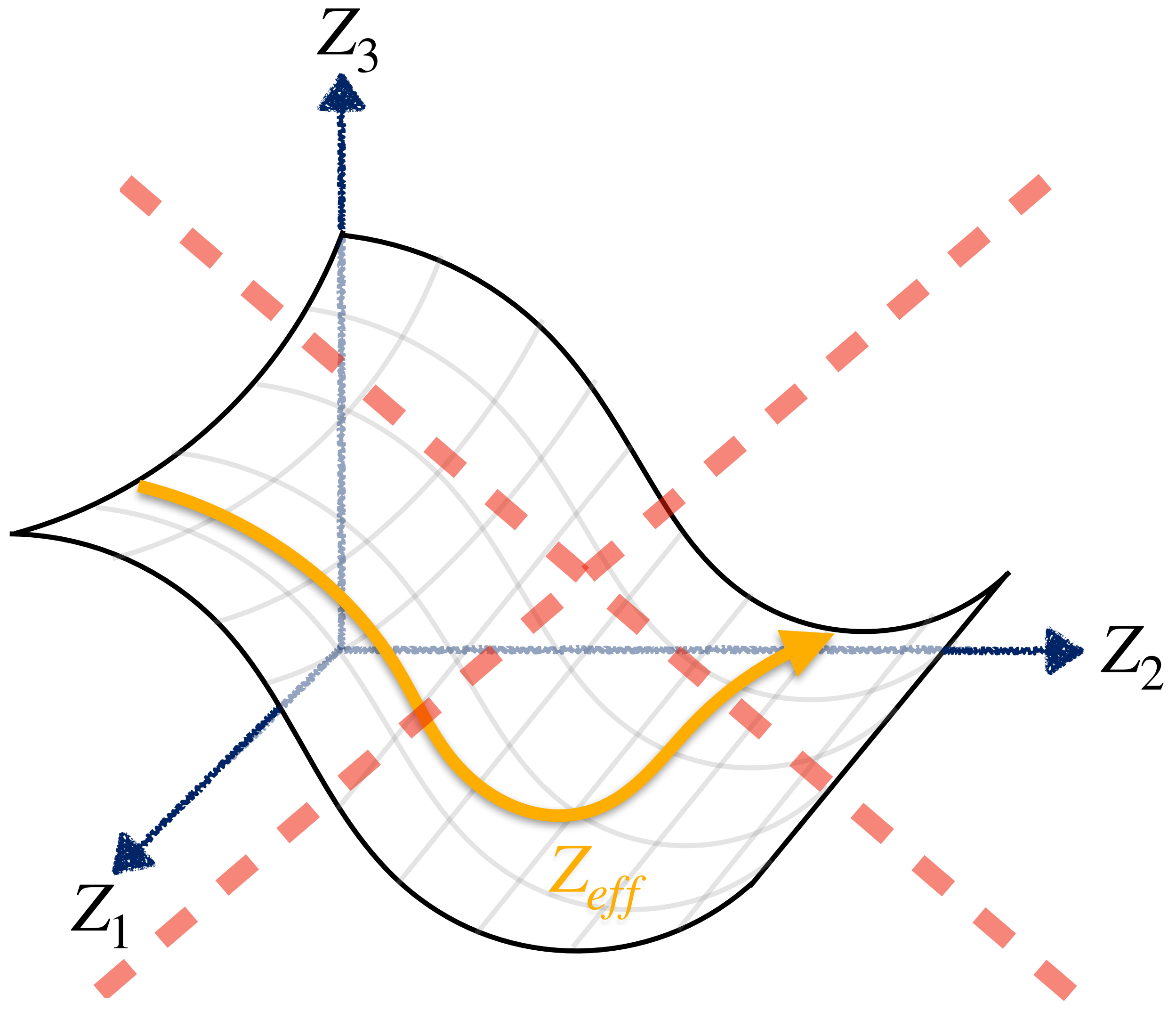
Optimisation module
Find minima by solving $\partial_I V = 0$ using scipy.optimize.root

Filter module
Check for vacua and consistent LCS truncation

JAXvacua
Filter module



1.) Positive string coupling and gauge inequivalence under $SL(2,\mathbb{Z})$



2.) Extremum is minimum (positivity of Hessian) and absence of flat directions

3.) Validity of LCS expansion (VEVs within radius of convergence of instanton sum):

$$\varepsilon = \frac{|F_{inst}|}{|F_{pert}|} \ll 1$$

in practice: $\varepsilon \leq 0.01$

General technique to determine radius of convergence are currently unavailable, but can easily be added to the module once available.

JAXvacua

h11=2: Results

Benchmarking our performance:

Generating solutions at fixed $N_{flux} = 34$

100 nodes each with 32 cores to find
24,882 solutions in **75,000 hours**

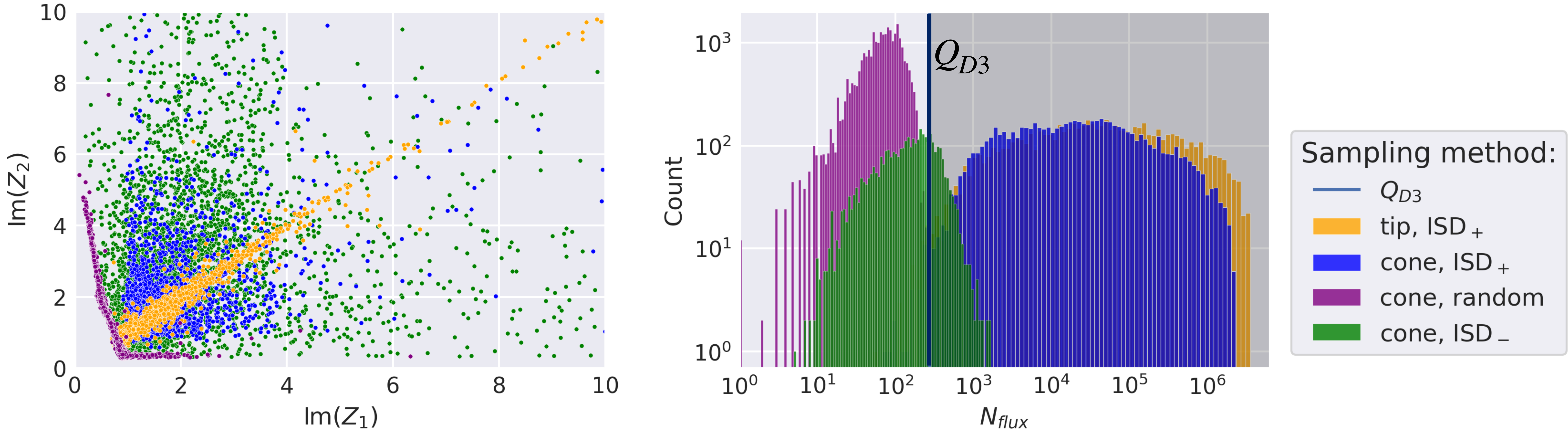
[Martinez-Pedrera et al. 1212.4530]

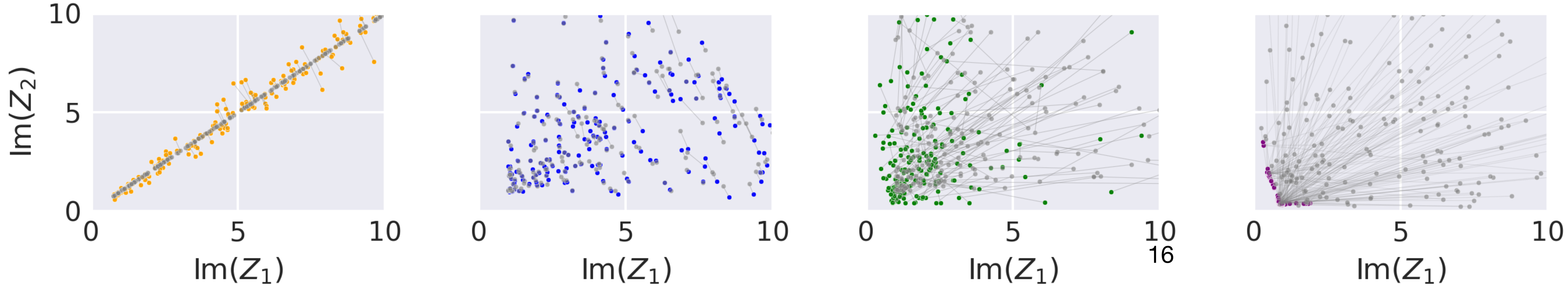
vs.

4 cores with 5GB of memory
33,019 solutions in **45 minutes**

[Dubey, SK, Schachner 2306.06160]

To quantify our sampling biases, compare different samplings for optimisation





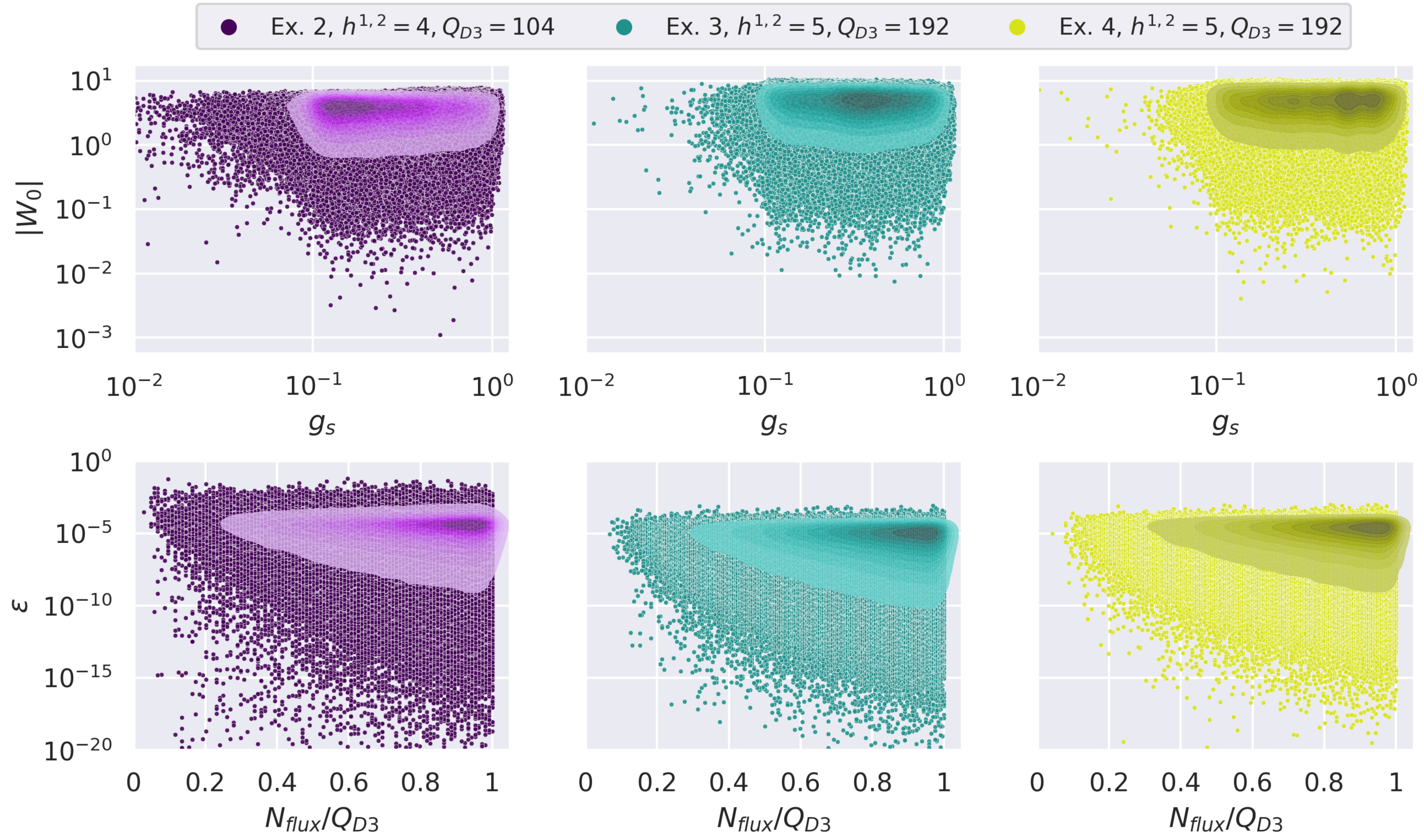
$$\langle Z^i \rangle = Z_0^i + \delta Z^i \,, \quad \langle \tau \rangle = \tau_0 + \delta\tau$$

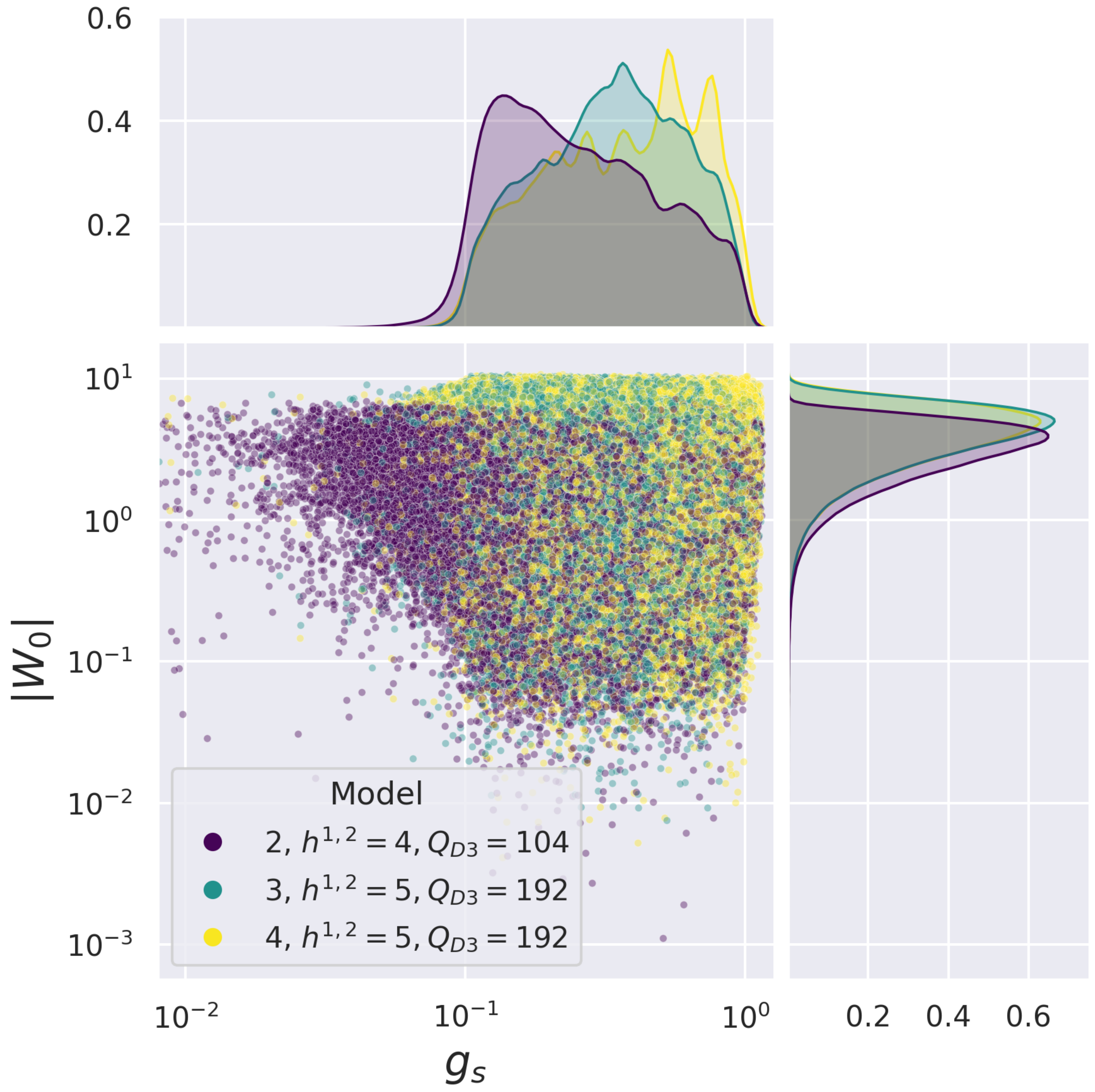
JAXvacua

h12=4,5 — previously hard to access. Now easy

With similar computational resources of 10 CPUs:

$\mathcal{O}(1)$ minima [Cicoli, Klevers, SK et al. 1312.0014] $\rightarrow \mathcal{O}(10^6)$ minima





JAXvacua

Numerical results — $h^{1,2} \leq 25$

Scaling behaviour at larger $h^{1,2}$

$h^{1,1}$	$h^{1,2}$	Q_{D3}	success rate	♯vacua
213	5	220	50%	1,370,842
244	10	256	16%	498,545
399	15	416	7%	168,291
350	20	372	$< 1\%$	36
245	25	272	$< 1\%$	1

Success rate decreases rapidly because

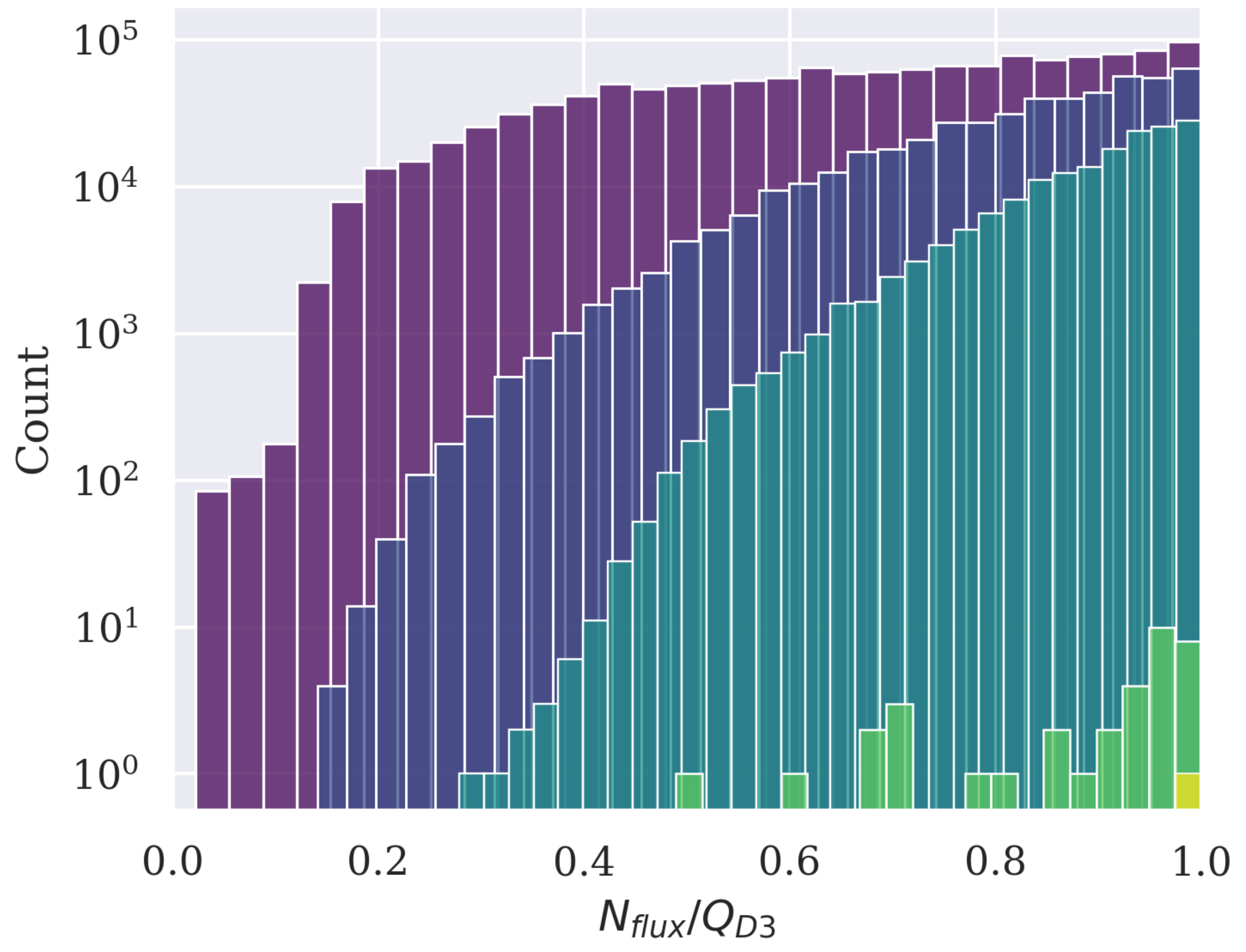
- high dimensionality means slower evaluation time
- harder to perform numerical optimisation
- phase of Kähler cone becomes narrower `[Demirtas et al. 1808.01282]`

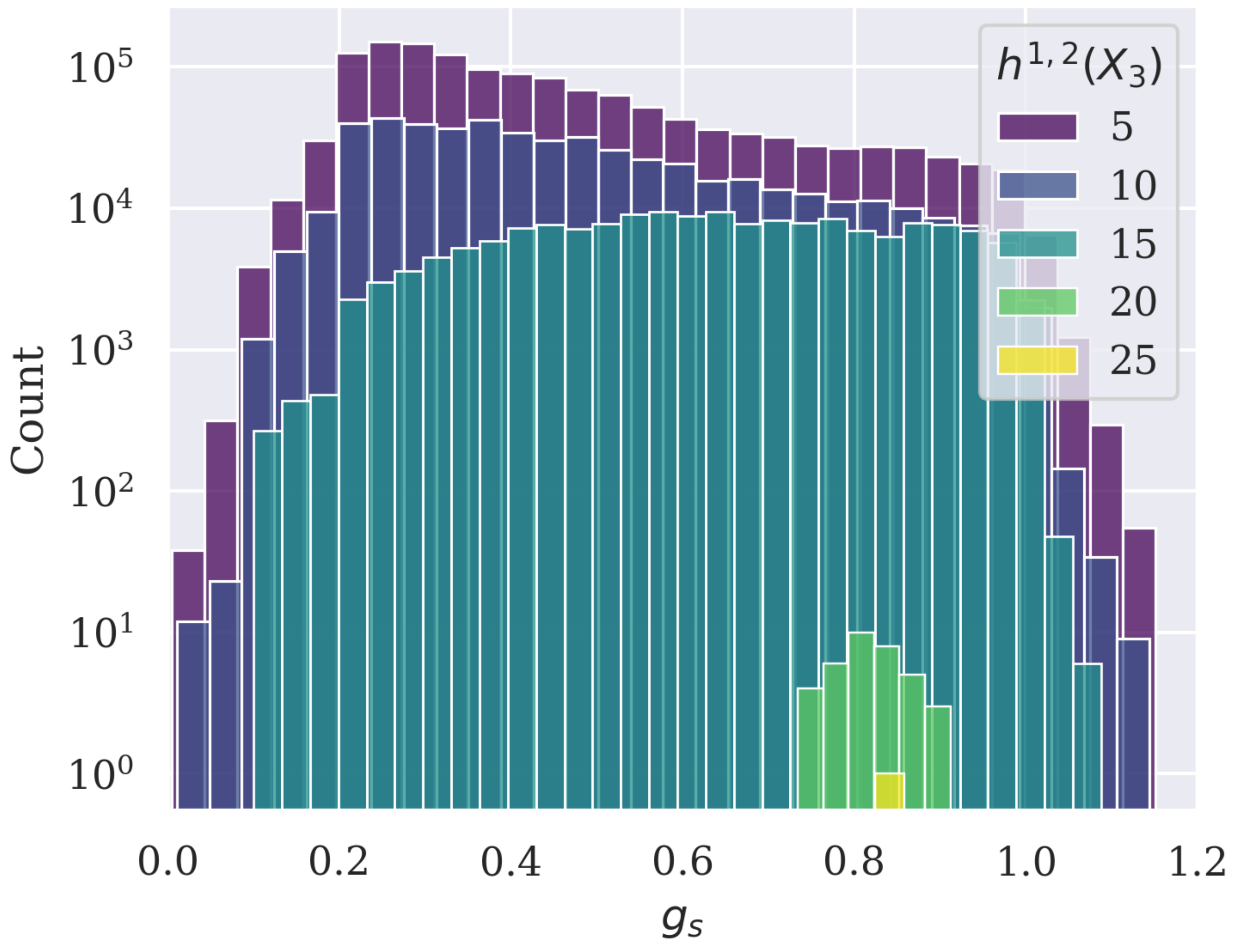
Important to stress: sampling with $N_{flux} \leq Q_{D3}$

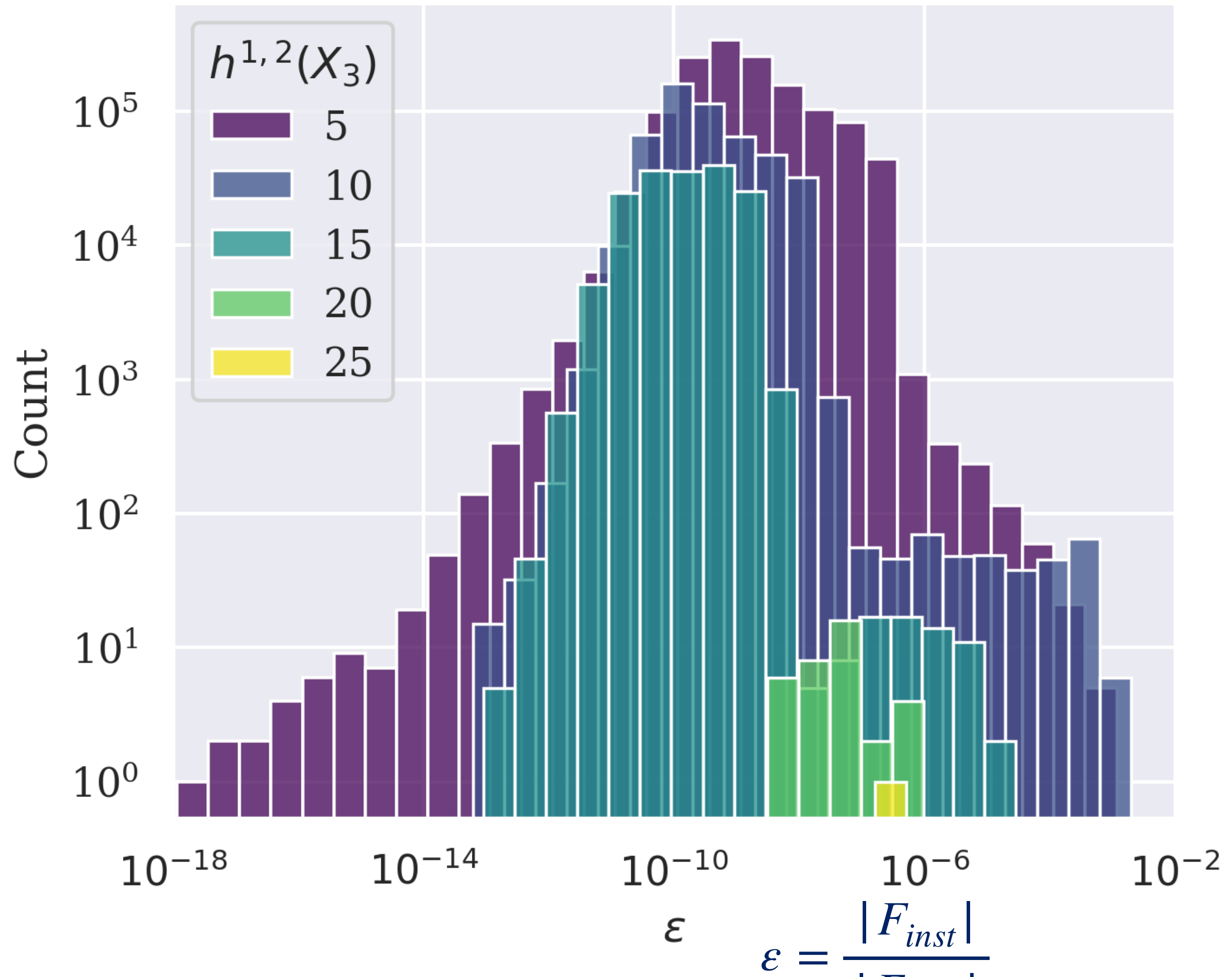
much harder than allowing $N_{flux} \to \infty$.

We actually looked at examples with $h^{1,2} > 100$

and found solutions with $N_{flux} \gg Q_{D3}$

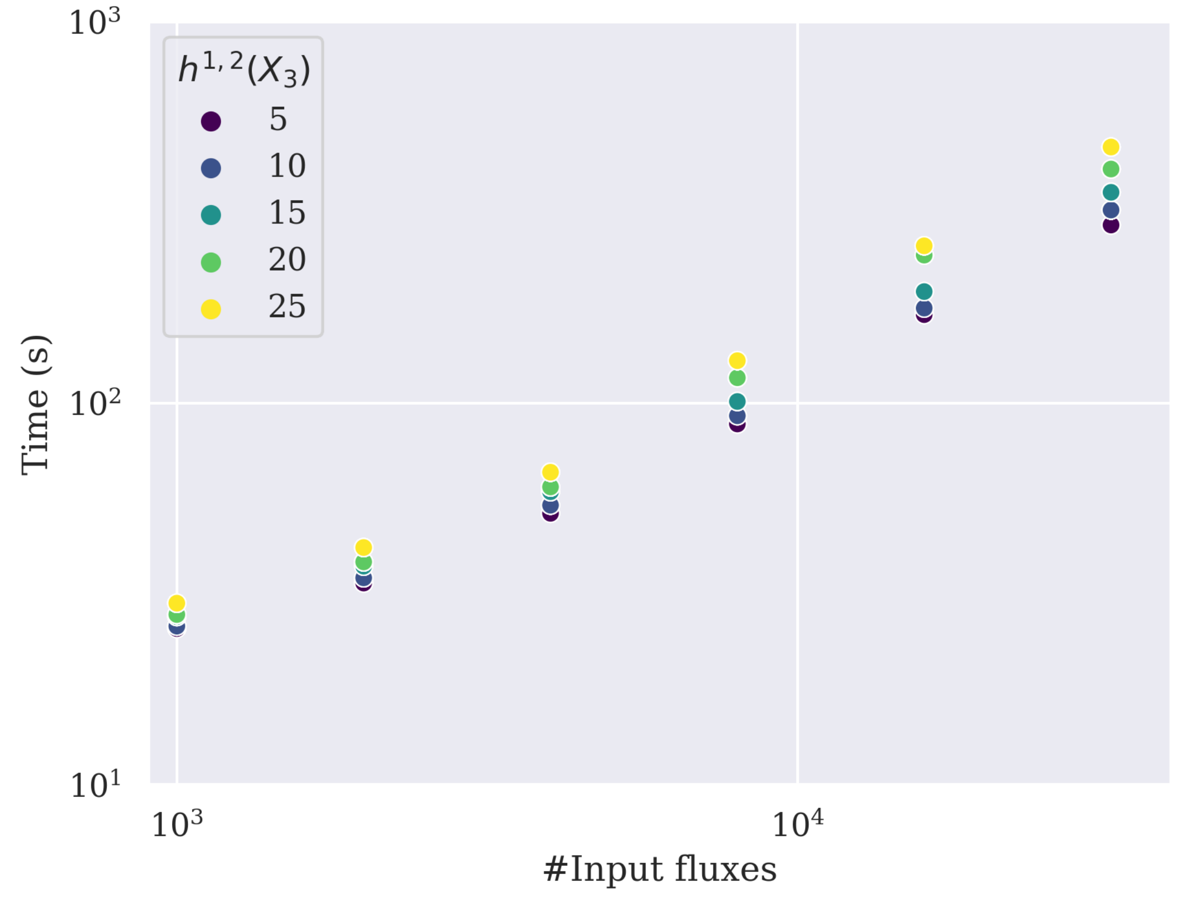






$\epsilon = \frac{|F_{inst}|}{|F_{pert}|}$

Timing Minimisation:



Let’s do some interesting physics…

What can we say about W_0?

Work with J. Ebelt (Master student), A. Schachner

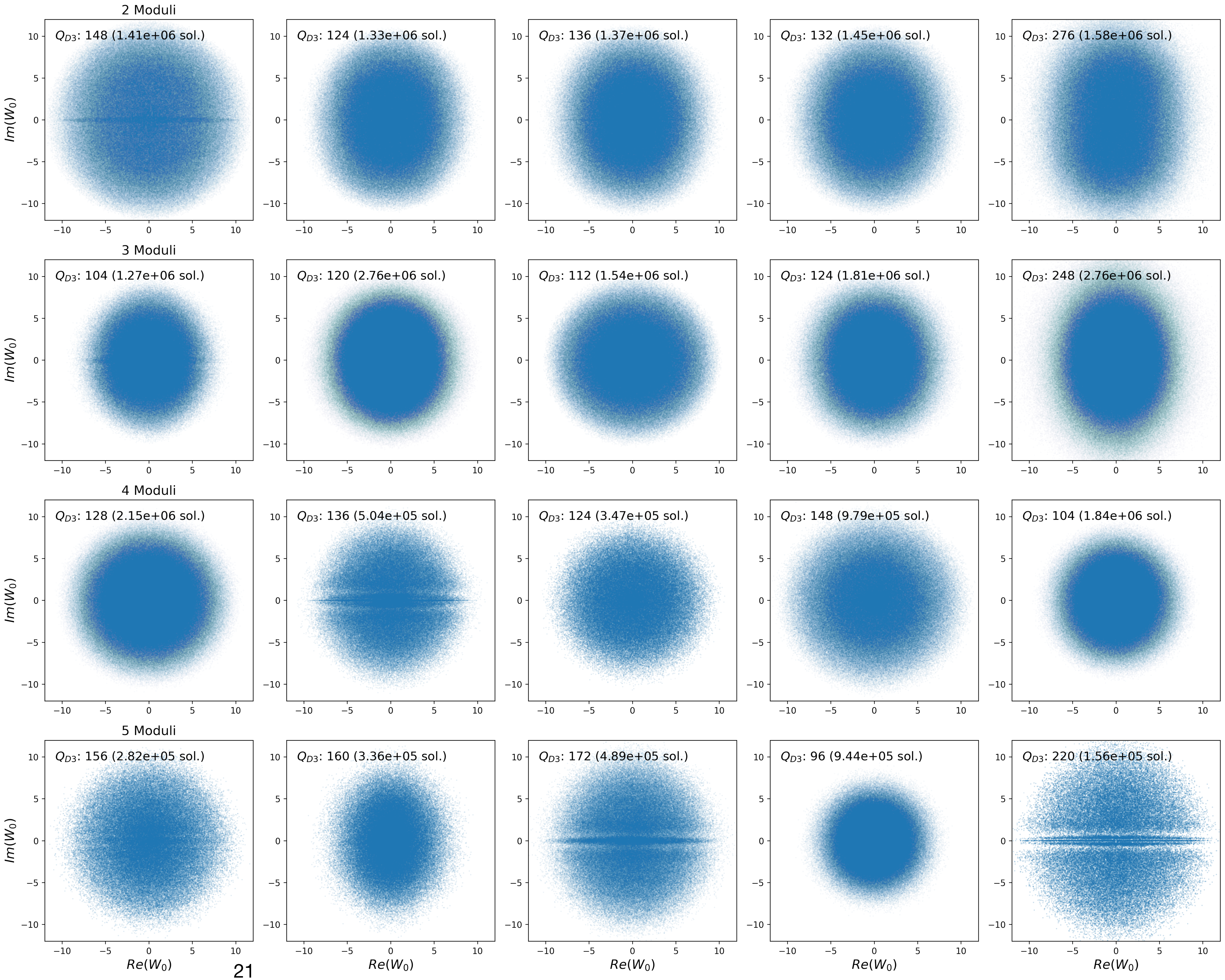
What can we say about W_0?

$$W_0 = \sqrt{2/\pi}\; e^{K/2}\; W$$

Universal behaviour

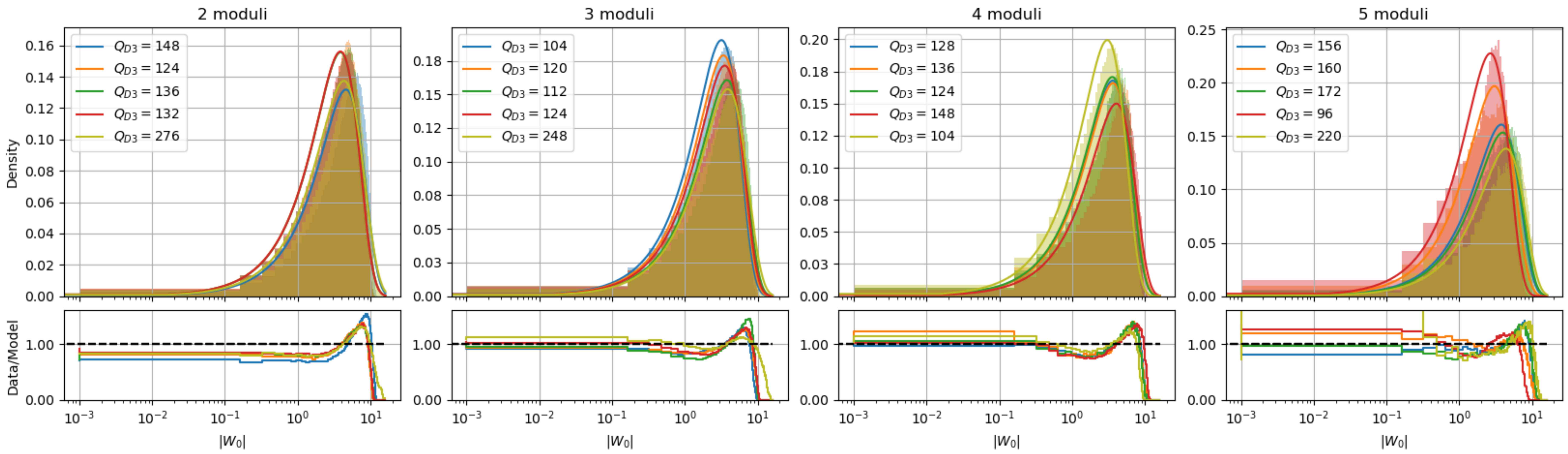
Looks Gaussian?

(Near origin: Gaussian ≈ Uniform [Douglas])



What can we say about W_0?

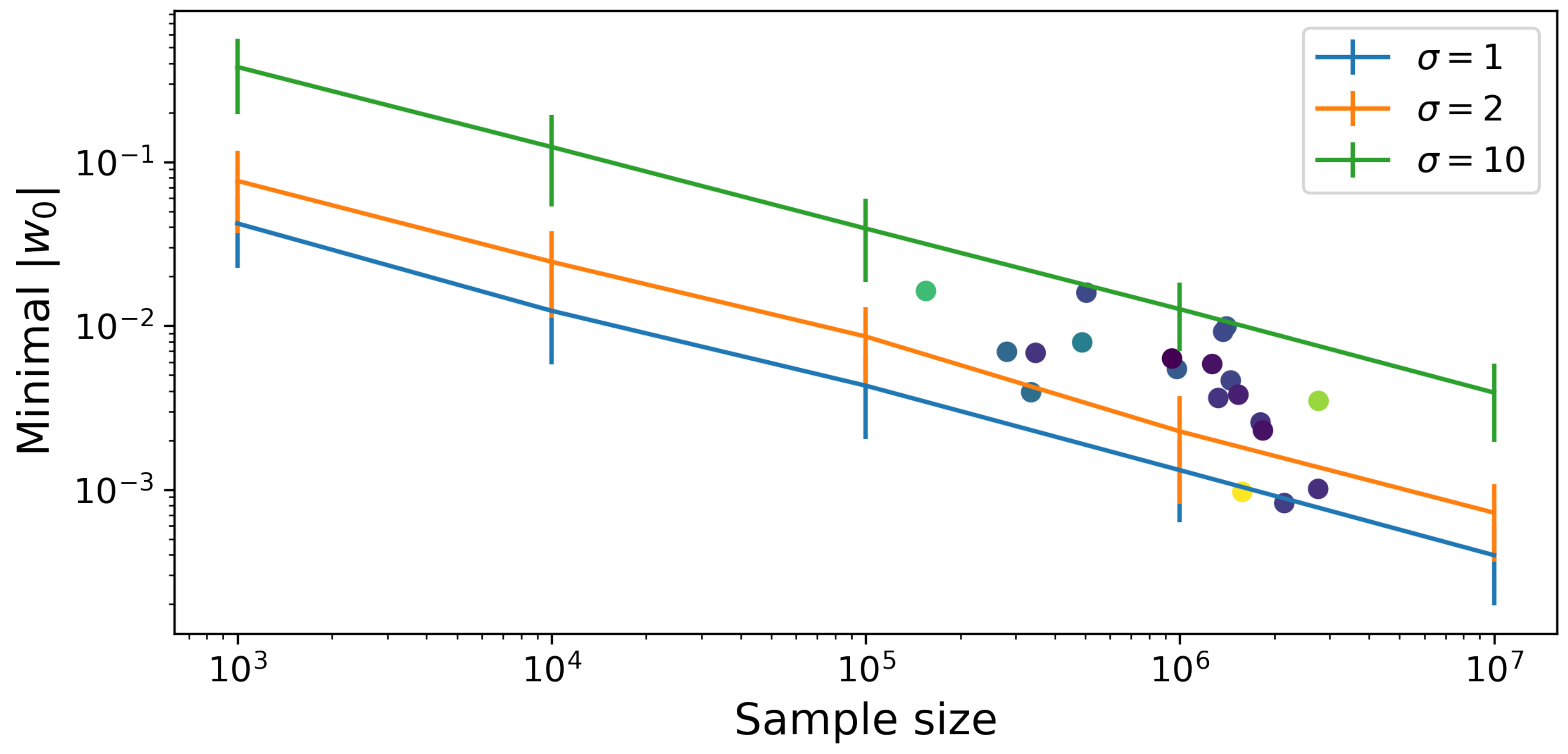
Gaussian distribution is a reasonable fit

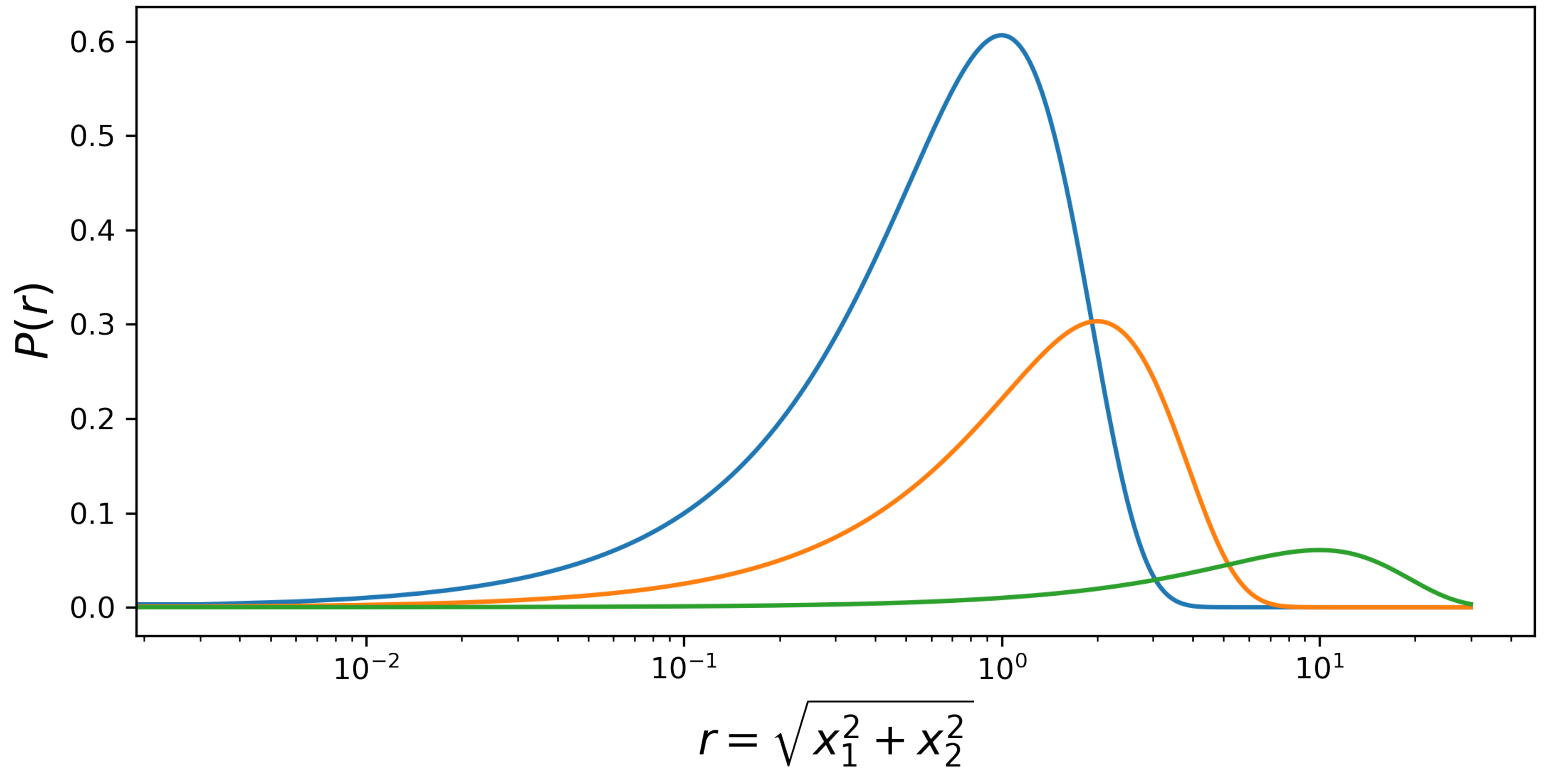


Gaussian approximation → Expectation for smallest value for $|W_0|$ for a given sample.

Standard deviation + sample size relevant!

Benchmark for dedicated search algorithms.





Can we construct SUSY breaking minima $D_I W \neq 0$?

SK, Schachner WIP

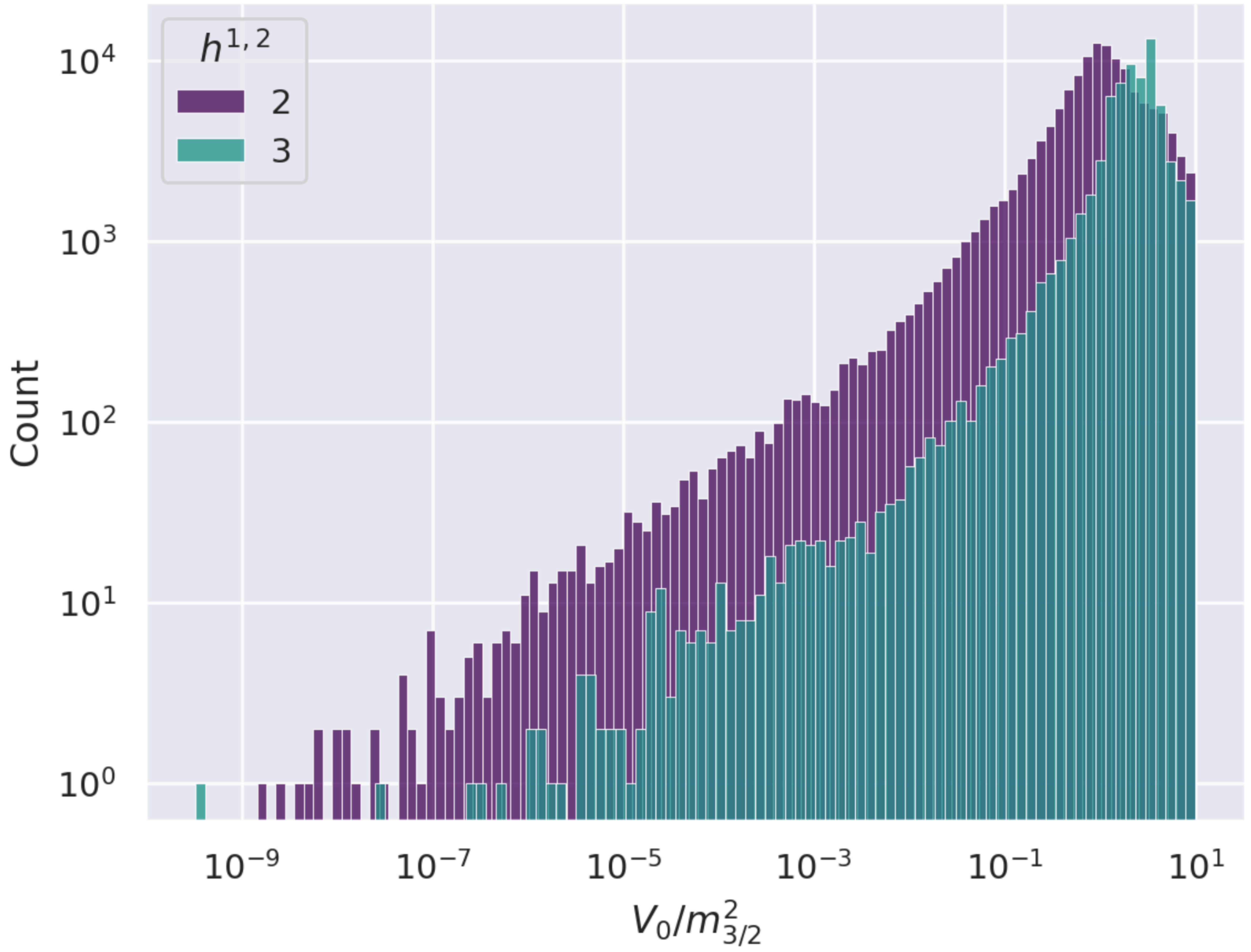
New SUSY breaking vacua

Ensemble with $D_I W \neq 0$

Non-SUSY solutions with $D_I W \neq 0$:

- interesting for their potential application to de Sitter model building **[Saltman, Silverstein hep-th/0402135]**
- almost nothing known about solution space apart from e.g. for continuous fluxes **[Gallego et al. 1707.01095]**
- can be searched for easily our framework by using $\partial_I V = 0$ for the optimisation target

We see some interesting hierarchical suppressions!

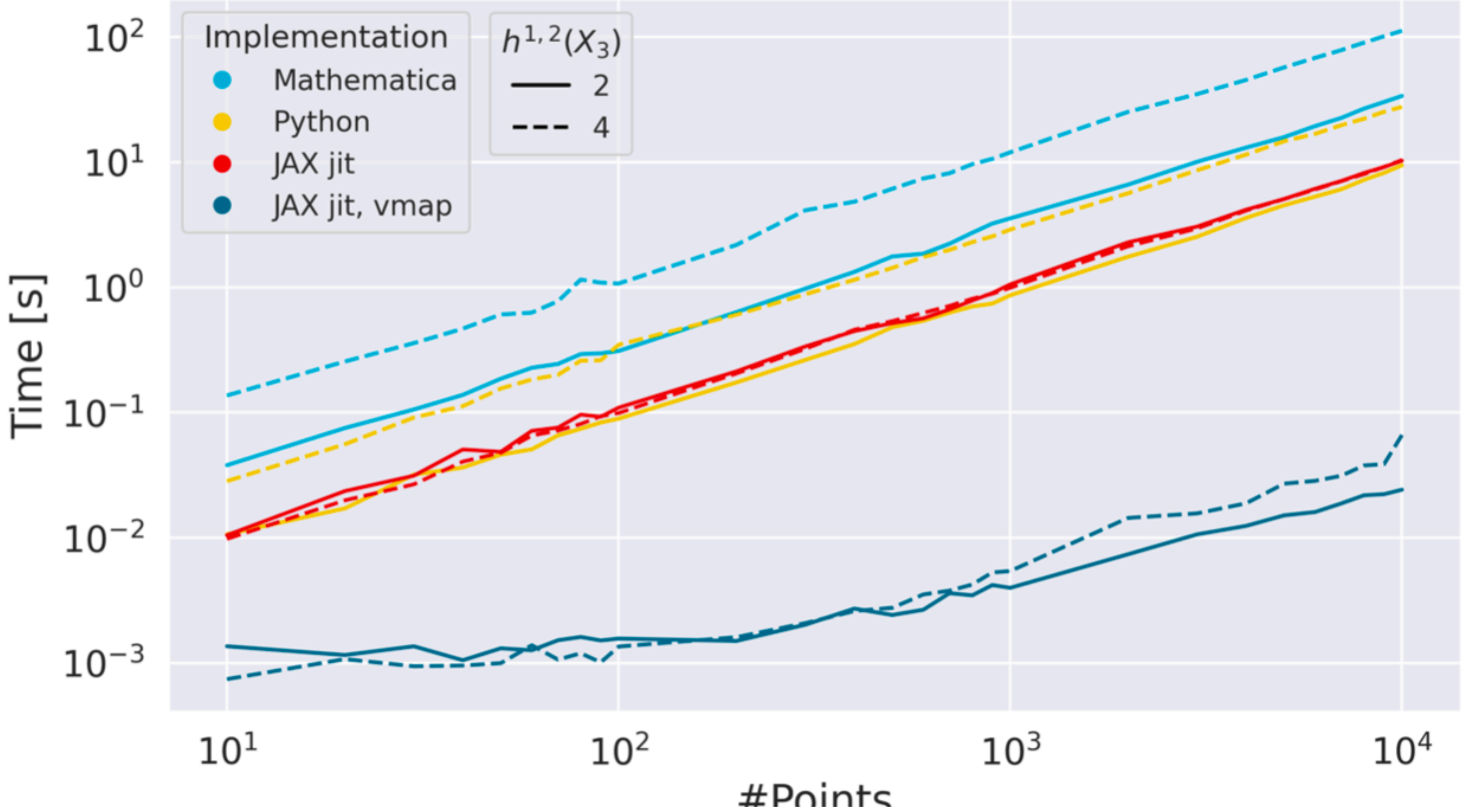


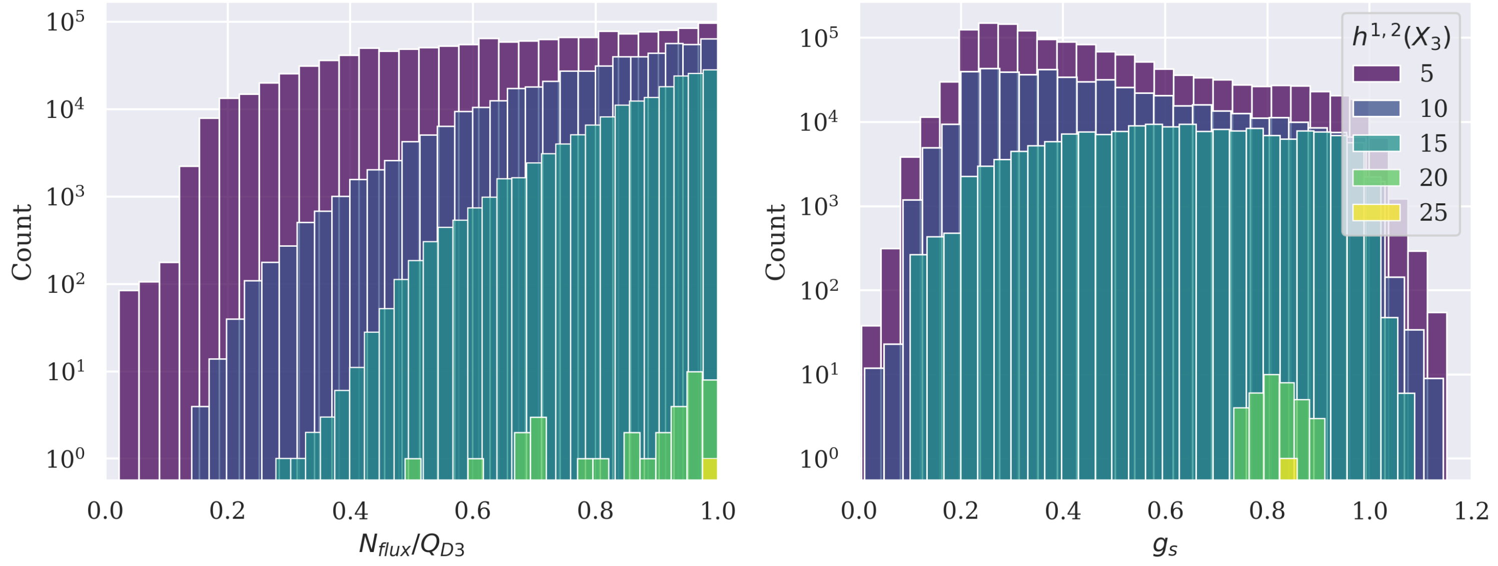
$h^{1,2} = 2 : 171167$ solutions, $h^{1,2} = 3 : 79068$ solutions

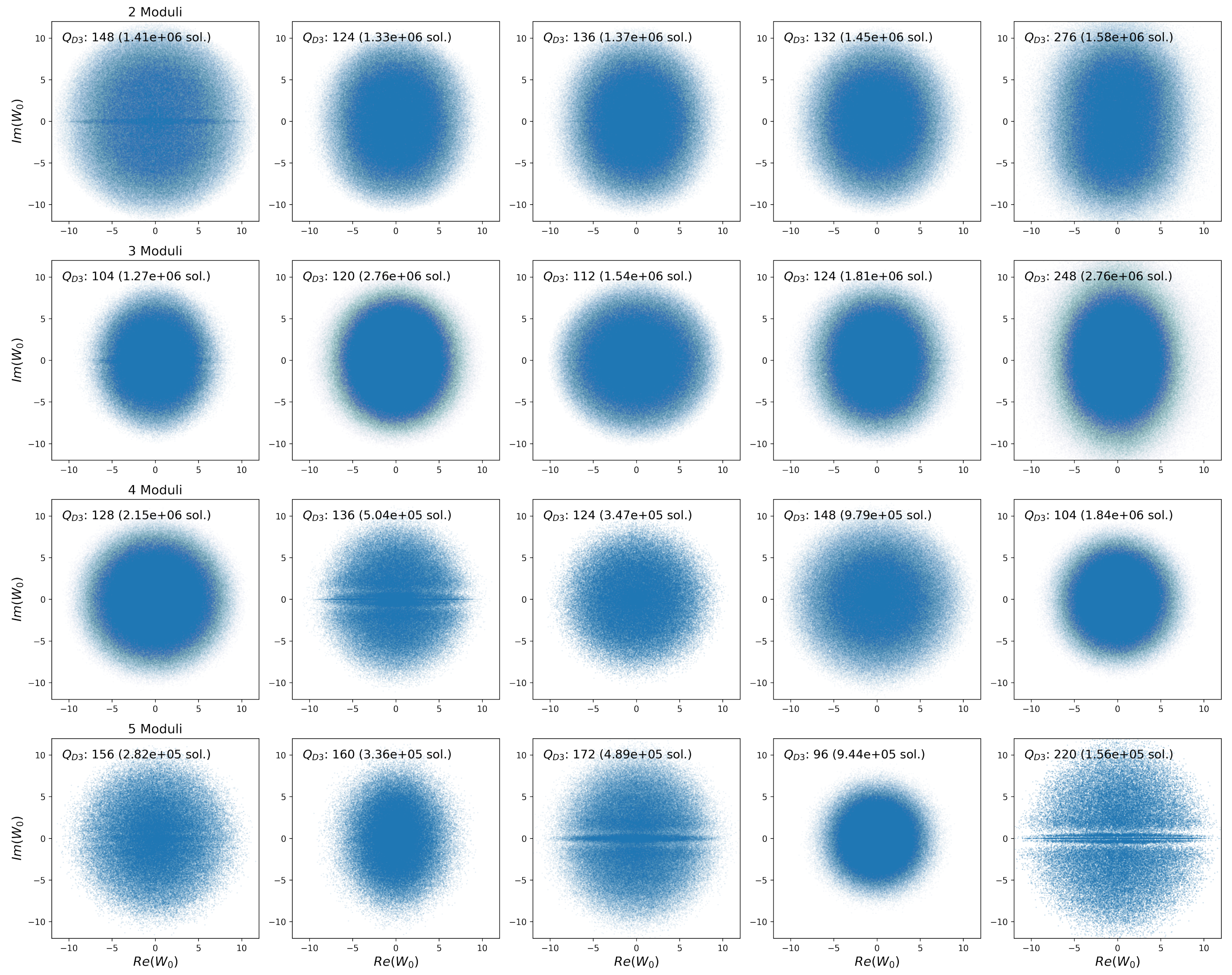
Conclusions

A new look on flux vacua

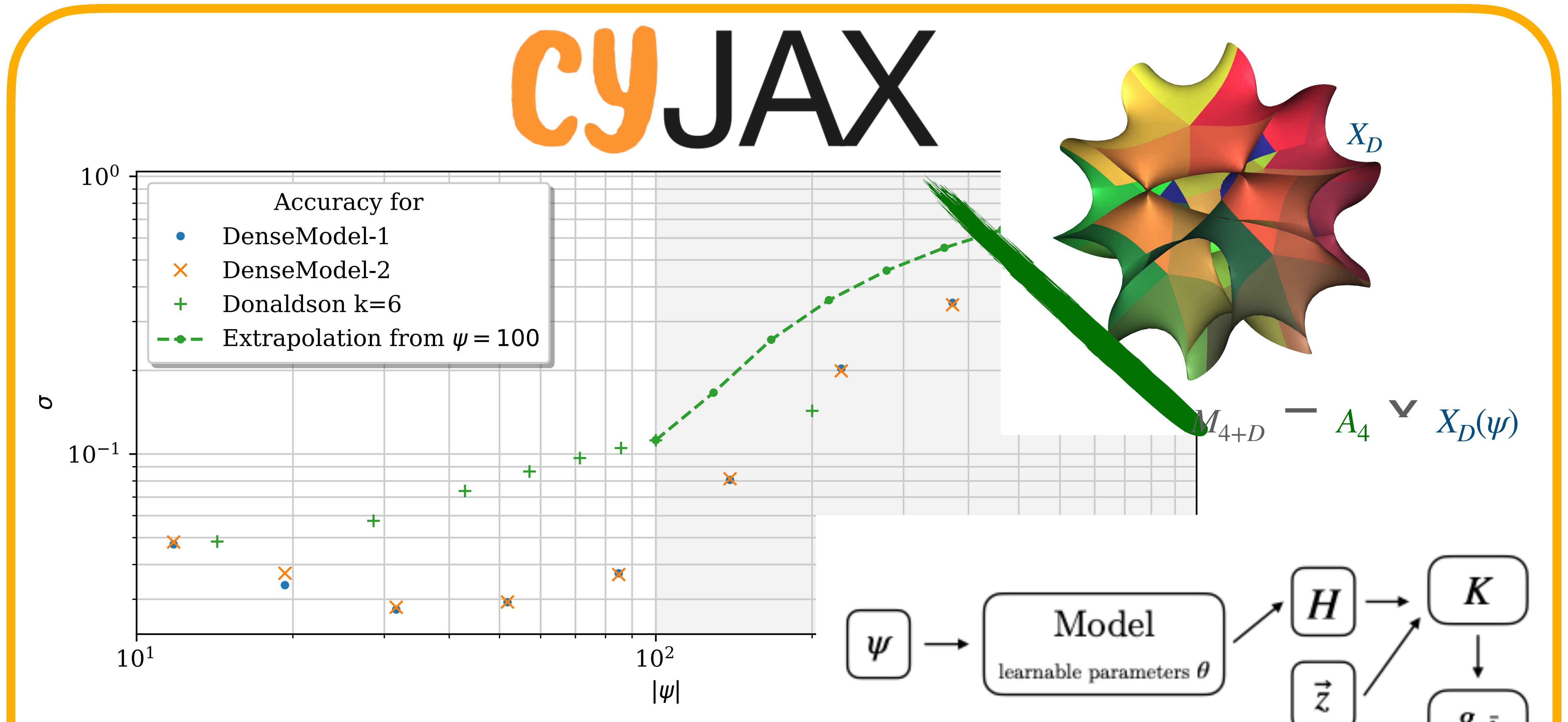
- Efficient code to generate ensembles of flux vacua (autodiff, just in time compilation, vectorisation)
- First large ensembles beyond 1 & 2 moduli cases. Ensembles are necessary to demonstrate/disprove non-existence of certain types of solutions.
- Universal Gaussian behaviour across geometries
- Flexible code allows to search for different type of solutions. Here: Ensemble of SUSY breaking solutions. Hint for hierarchies
- 20 years after Douglas et al. seminal works we are now ready to compare with meaningful ensembles. Let's look at their physics!
  Which imprint to discreteness and finiteness leave? Can we achieve interesting parameter regimes.







Thank you!



We learn the metric of extra-dimensional spaces

Gerdes, Krippendorf: 2211.12520

Highly recommended: Andreas Schachner's talk on Thursday