DMC Parameter Tuning Using the *Autotune* Package

*Mike Kelsey, **Dylan Monteiro**, Warren Perry, David Toback*

DMC sub-WG meeting 12/17/2025

SUPER CDMS

Physics and Astronomy

Outline

1. Review of Tuning Motivation and Goals
2. Example of Manual Tuning Procedure
3. Motivation for Automation
4. Autotune Design
5. Initial Results
6. Future Plans

Review of Tuning Motivation and Goals

SUPER CDMS
Physics and Astronomy

Big Picture:

- Simulation models billions of microphysical interactions and processes using parameterized equations, with some parameters determined phenomenologically.
- Want to be confident that simulation predicts an acceptable Standard Model response for crucial quantities, that is well-enough understood to be used in dark matter search analyses.

Current Status: Simulation does well with some qualitative descriptions, but does not quantitatively reproduce experimental data.

Several possible explanations:

1. Our models are inaccurate or incomplete.
2. The models' parameters have values too far from being a useful description of real data. ← Tuning
3. Our choices for configuring and analyzing simulation are inconsistent with real life.

Simulation tuning: Methodically varying parameters to find the values which result in the best match of simulation to experimental data.

Goal of this talk is to describe methods for automating portions of simulation tuning. See Warren's Confluence page if interested in more on tuning philosophy.

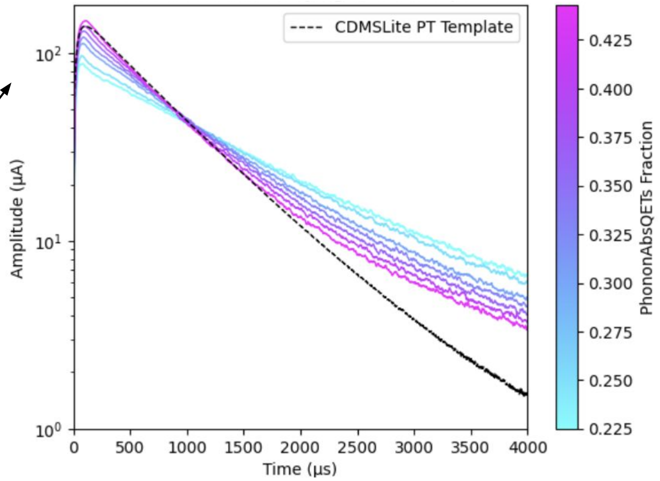
Example of Manual Tuning Procedure

Physics and Astronomy

See here for more tuning examples.

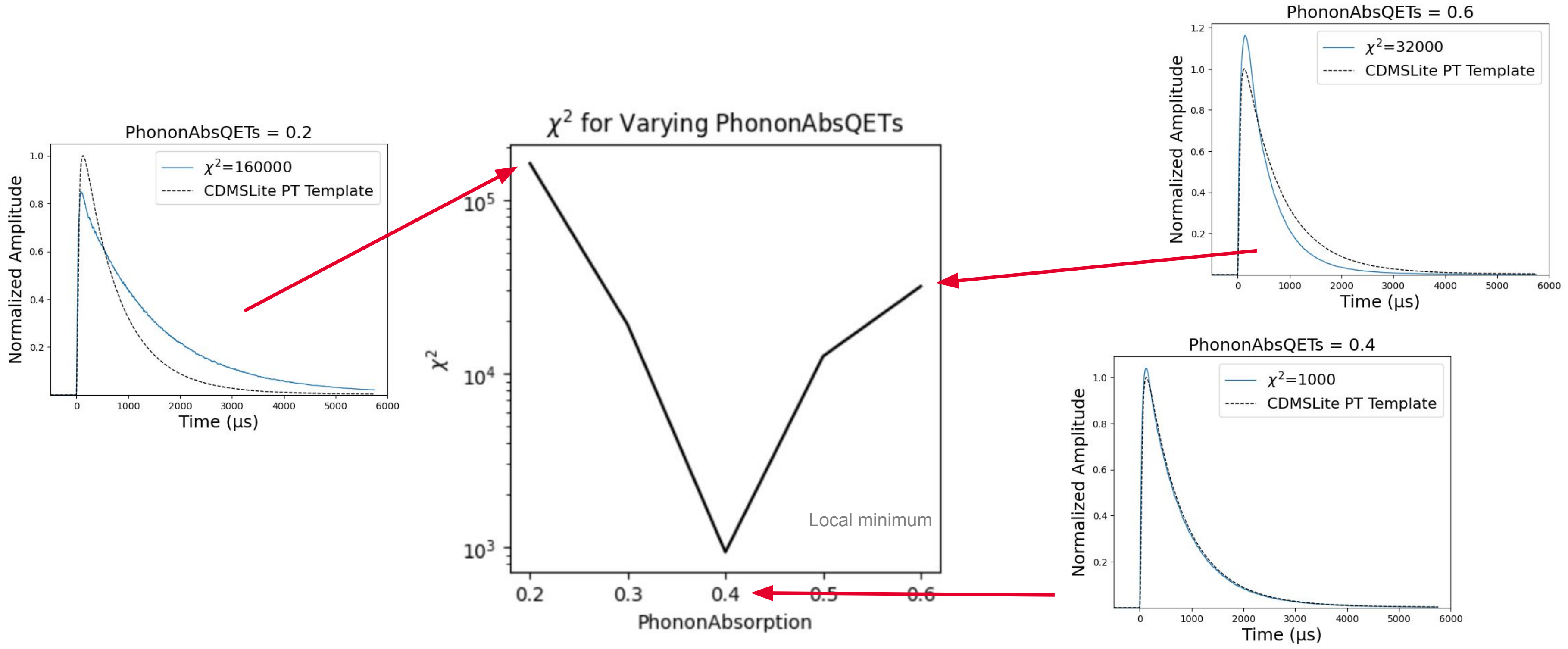
1. Define tuning metric of interest. For example, in comparing simulation and experiment, minimizing:
    - χ^2 of simulated pulse or set of pulses to template. (Just like CDMSBats OptimalFilter)
    - Difference between means and/or standard deviations of some distribution.
2. Identify a set of relevant DMC parameters to adjust.
3. Run simulation many times to create samples of events, each varying the value of a single parameter.
4. Calculate the metric for each parameter value, locate parameter value that produces best output*.
5. Fix value of previous parameter, switch to new parameter and repeat process.

Example Output of Varying Parameter Values



*See next page

Example of Manual Tuning Step 4



Autotune Motivation and Goals

SUPER CDMS

Physics and Astronomy

- **Tuning by hand is…**
  - Time consuming.
  - Computationally intensive.
  - Difficult to control for correlated parameters and those with degenerate effects.
  - Prone to input errors
- **Automatic tuning would…**
  - Reduce drawbacks of manual tuning.
  - Introduce a reproducible system for validating tuning results.
- ***Autotune*** was created to accomplish the above goals.

From the Autotune README:

"Autotune is a python-based package designed to determine the values of constants and parameters that produce the most optimal SuperSim Package output (i.e. what some might call simulation tuning). It utilizes an n-dimensional gradient descent algorithm recursively connected to a SuperSim batch job submitter to minimize a user-defined quantity."

Automated Tuning Procedure

SuperCDMS | Texas A&M Physics and Astronomy

Manual Tuning	Using Autotune
1. Define tuning metric. 2. Construct simulation setup through SuperSim macros. 3. Identify parameters to vary. 4. Create set of simulation samples, varying a single parameter only. 5. Calculate metric for all samples. 6. Fix parameter at minimum, switch to new parameter. 7. Repeat steps 4-6 until a minimum is found in all parameters.	1. Define tuning metric (as integrated python function). 2. Construct simulation setup through SuperSim macros. 3. Identify parameters to vary (in Autotune config). 4. Run Autotune to find parameter set that produces best metric value. 5. Scan local region to confirm not in "local hole".

Algorithm Design

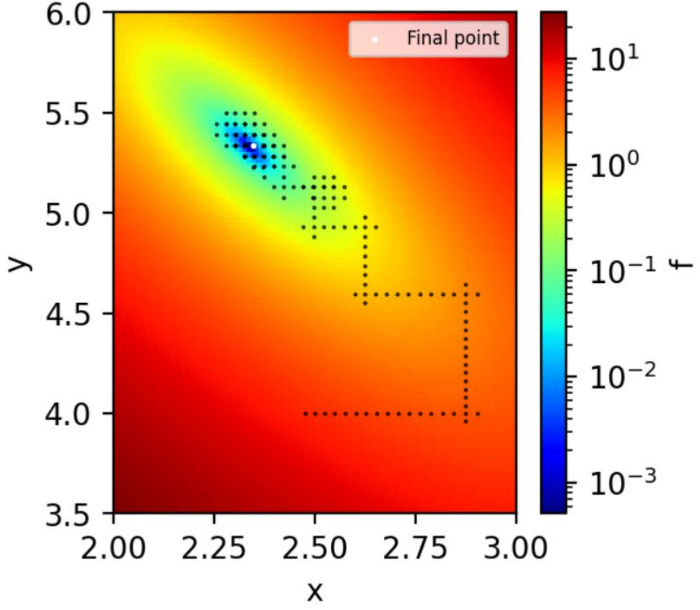
Physics and Astronomy

Algorithm structure:

- N-dimensional for any set of simulation parameters.
- Descends each axis until finding local minimum, then switches to new axis.
- Recursively interfaces with simulation package to create event samples, analyze data, and descend gradient.

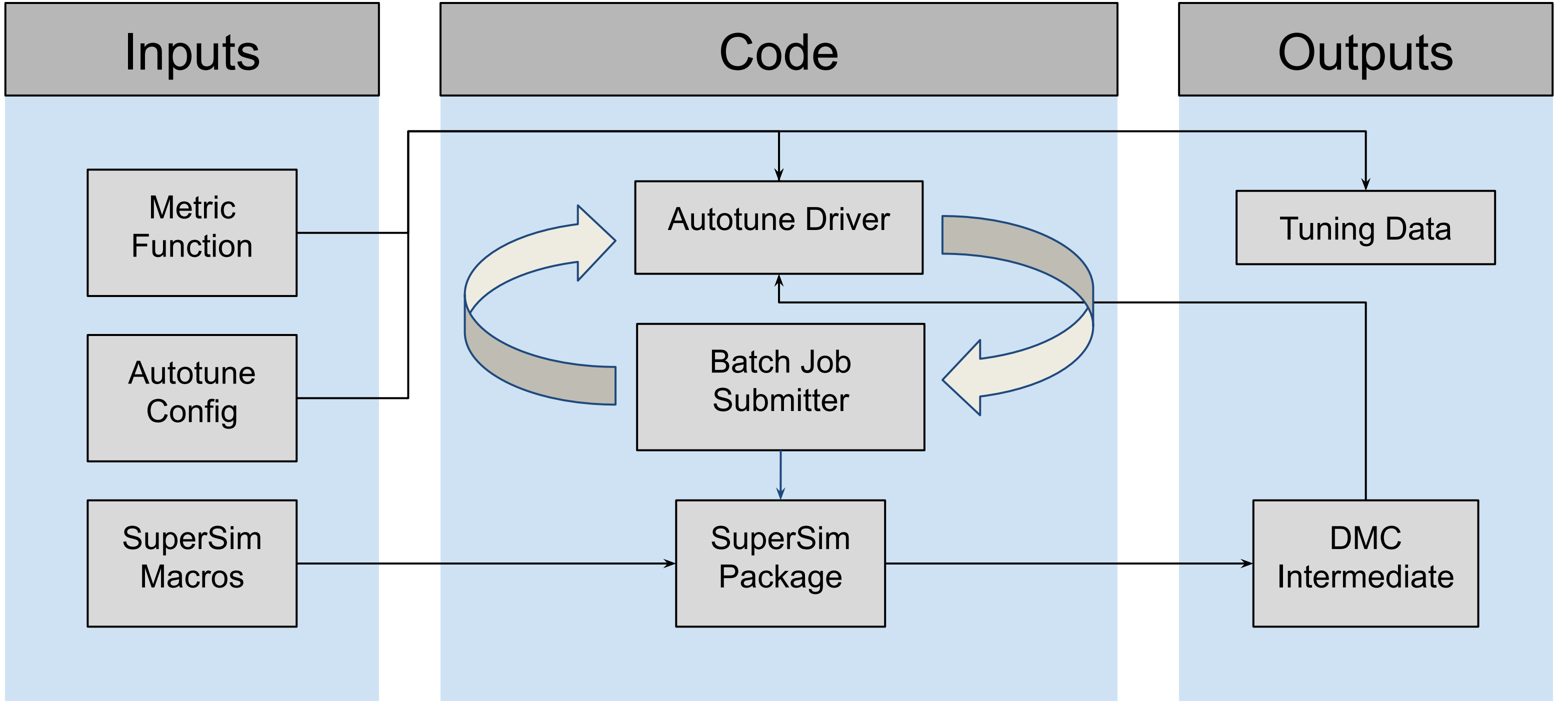
Possible to fall into a local min rather than more-optimal global min. Post-tuning parameter scans mitigate this risk.

Descent path on “dummy” 2D test data



Autotune Infrastructure

Physics and Astronomy



Example Results Using CDMSLite Data

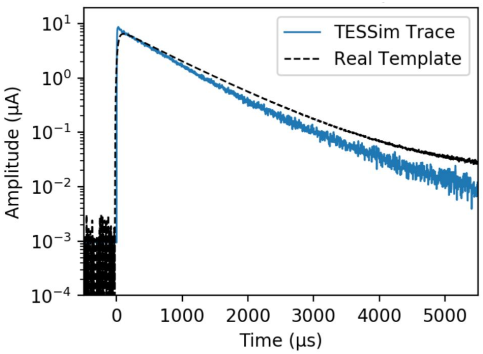
Tuning Statistics

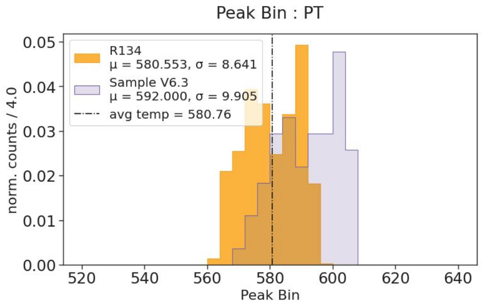
- **Parameters varied:** PhononAbsQET, TESsubgapAbs, Tc, Tw, Tsubst, W_gamma, W_sigma, L
- 40 BulkER events/sample
- ~50 tuning samples to reach minimum
- ~8hr total wall-clock time

R134 = 2018 CDMSLite run 134
Sample V6 = Simulated CDMSlite response

Pre-Tuning Shapes

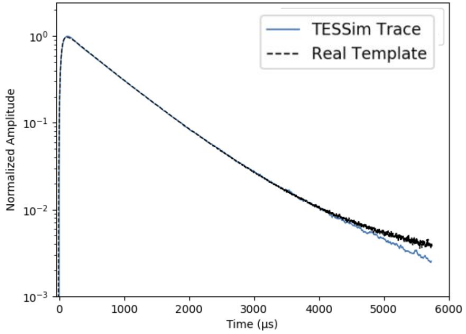
CDMSLite PT Trace

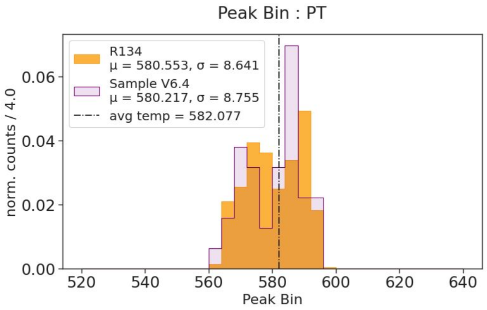




Rik Bhattacharyya, 2025

Post-Tuning Shapes





Rik Bhattacharyya, 2025

Future Plans

Physics and Astronomy

- Finish development of code into a “proper” python package to support directory-agnostic use by others.
- Allow simultaneous jobs with different macros/metrics to enable multiple tuning campaigns in parallel
- Improve documentation to make it easily accessible to new users.
- Improve validation and diagnostic notebooks.
- Address minor bugs.

Conclusions

Physics and Astronomy

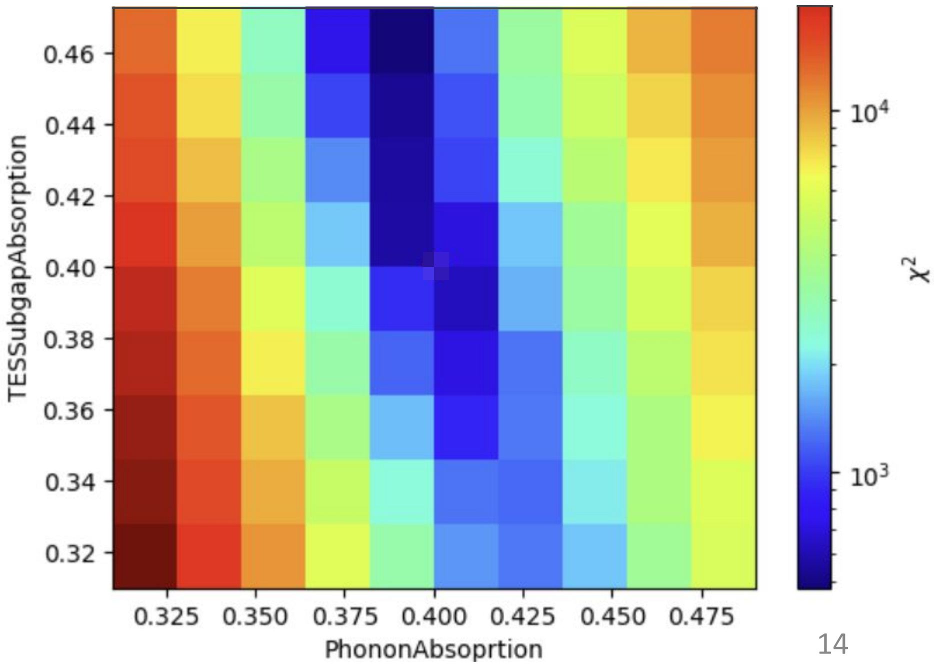
- **Portions of a simple, multi-parameter simulation tuning procedure have been coded** into a software package called Autotune to assist in the tuning process and handle correlated parameters.
- This package takes as input a set of parameters to vary and the calculation method of a metric, and finds a best fit minimum.
- Autotune **has successfully completed several tuning campaigns**.
- In principle it could be used for multiple purposes, but **future development is still needed** before it is ready to be released for broader collaboration use.

Backups

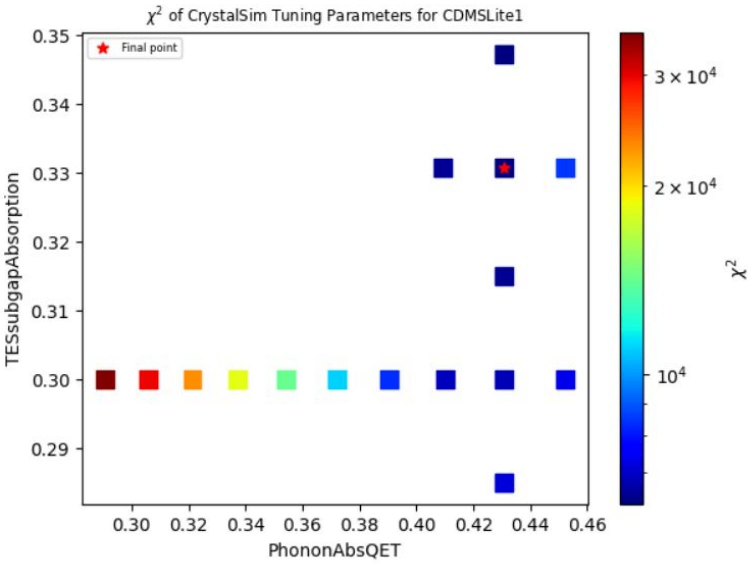
Multi-Dimensional χ^2 Minimization

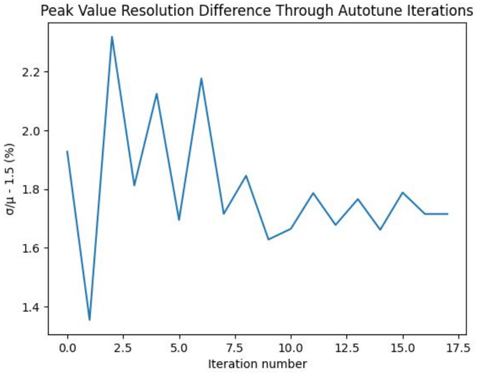
When minimizing χ^2 over multiple parameters, correlation and degeneracies are revealed.

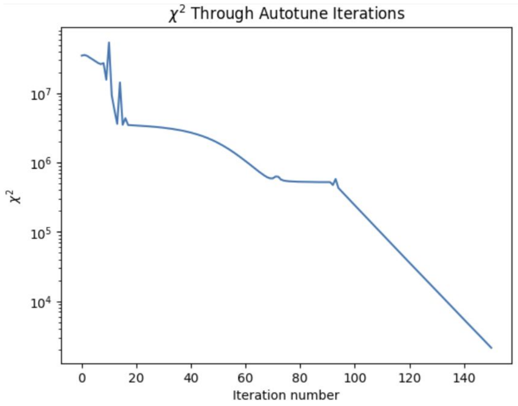
χ^2 Minimum in 2D Region



Metric Minimization Steps







Diagnostic Plots from Autotune