# Efficient CI Estimation for Neutrino Oscillation Parameters with Gaussian Process

**Lingge Li, Nitish Nayak, Jianming Bian, Pierre Baldi**

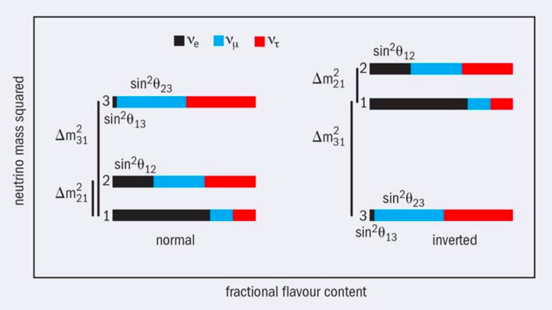
**UC-Irvine**

**NPML 2020**

# Neutrino Oscillations

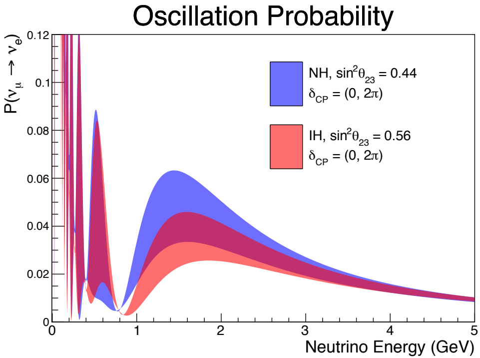
- Neutrino oscillations between flavor states occur with a well defined probability which depends on the $U_{PMNS}$ mixing matrix
- LBL experiments (focus of this talk) measure $P(\nu_\mu \rightarrow \nu_\mu)$ and $P(\nu_\mu \rightarrow \nu_e)$ to infer :

- $\Delta m^2_{32} > 0$ or $< 0$? (Normal or Inverted)
  - Identifying mass hierarchy (NH or IH) has implications for neutrino mass measurements
- Octant of $\theta_{23}$ or $\theta_{23} = 45°$?
- $sin\delta_{CP} \neq 0$?
  - Lepton sector CP-violation. Gives us a clue towards explaining matter-antimatter asymmetry

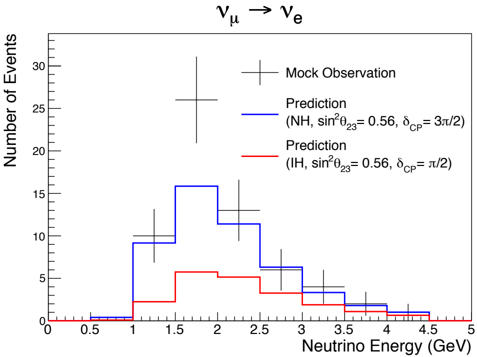


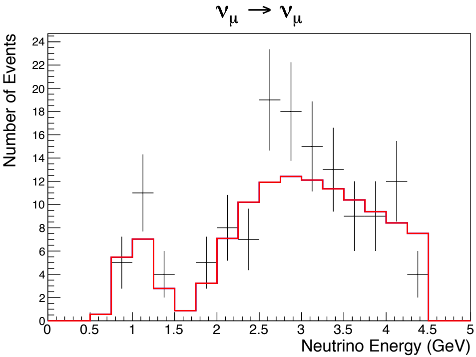
# Statistical Issues

- Experiments collect only a handful of statistics. $\mathcal{O}(10 - 100)$ over years of operation for the $\nu_\mu \to \nu_e$ channel
- Complicated interplay between different parameters $\implies$ difficult to delineate
- Confidence Intervals are hard to find as Likelihood ratios don't satisfy asymptotic properties.
- Let's illustrate this with a toy experiment..



# Toy Experiment



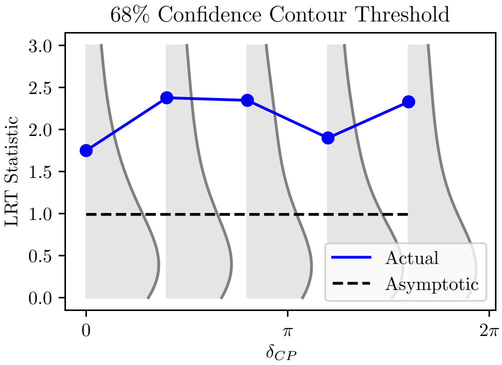

- Create a toy NOvA-like experiment. Data ($\vec{x}$) generated from Poisson variations at some chosen oscillation parameters.
- With $(\theta, \delta)$ denoting list of oscillation and nuisance (flux and xsec errors) parameters,
- Best-fit $(\hat{\theta}, \hat{\delta})$ found by minimizing negative log-likelihood over energy bins, $i$

$$-2 \log L(\theta, \delta) = -2 \sum_{i \in I} \log Pois(x_i; v(\theta, \delta)_i) - \sum_{i \in I} x_i + \sum_{i \in I} v(\theta, \delta)_i + \delta^2$$

# Confidence Intervals

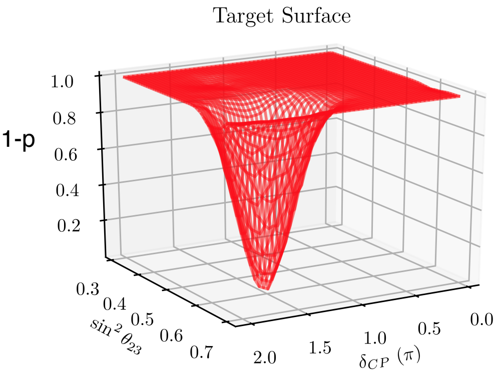
- Likelihood Ratio Tests (LRT) ($\Delta\chi^2$ from global best fit) typically used for estimating confidence intervals.
- In asymptotic case, test statistic : $\Delta\chi^2 \sim \chi^2_k \implies$ look up significance from PDG (Wilks' Theorem)
- In others $\implies$ Feldman-Cousins, i.e

- Explicitly simulate $\Delta\chi^2$ distribution using lots of pseudo-experiments
- Find p-value associated with $\Delta\chi^2_{data}$
- Gather all parameter values for which, say, *percentile* $= 1 - p < 0.68$ to get 1-$\sigma$ interval
- Correct coverage by construction
- Very heavy computational burden, often millions of CPU-hours needed!



# A more efficient FC

- In practice, FC proceeds via a grid search, for eg, simulating $\Delta\chi^2$ distributions for every point in $\sin^2\theta_{23} - \delta_{CP}$ space to find the 1-$\sigma$ contour
- If you had perfect foresight however, only the 1-$\sigma$ boundaries are needed, but obviously not known apriori




- Can we get an idea of how this surface looks like with a few pseudo-experiment throws?
- Can we then use this approximate surface to tell us where those boundaries lie?

# Gaussian Process

- Special case of Bayesian Learning. Assume p-value approximation is a random variable with a multivariate gaussian distribution
- We say, $f \sim \mathcal{GP}(\mu, k(\cdot,\cdot))$ if

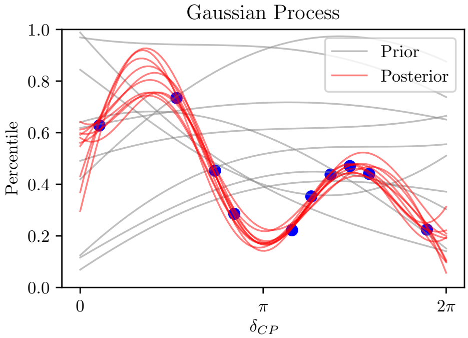
$$\begin{pmatrix} f(x) \\ f(x') \end{pmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu(x) \\ \mu(x') \end{bmatrix}, \begin{bmatrix} k(x,x) & k(x,x') \\ k(x,x') & k(x',x') \end{bmatrix}).$$

- Intuitively, we can picture each draw from a $\mathcal{GP}(\mu, k(\cdot,\cdot))$ giving us a different $f(x)$ with the average result being $\mu(x)$
- The kernel encodes the correlation between nearby points. A commonly used kernel is the radial basis function, $k(x,x') = \exp(-(x-x')^2/l^2)$
- A RBF kernel tells us that $\mathcal{GP}$ results at nearby points are highly influenced by observations at a given point while further out, they aren't.

# Why $\mathcal{GP}$s?

- Enormously flexible! Can basically approximate any well behaved function with an appropriate choice of the kernel.
- Predictions at new data points are posterior distributions calculated with basic linear algebra, i.e for $\mathcal{GP}(\mathbf{0}, k(\cdot,\cdot))$ :

$$f(x')|f(x) \sim \mathcal{N}(\frac{k(x,x')}{k(x,x)}f(x), k(x',x') - \frac{k(x,x')^2}{k(x,x)})$$



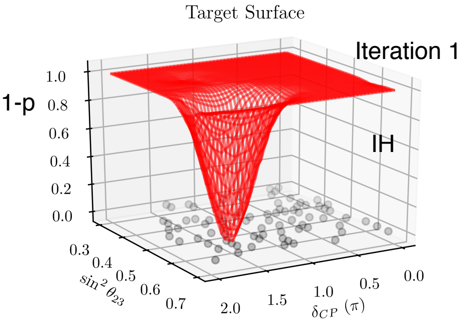

- Kernel hyperparameters can be learned and updated iteratively as well

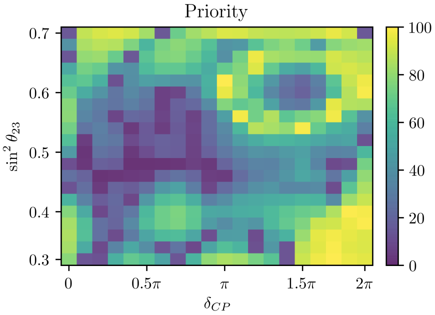
# Optimised Confidence Interval Search

- Use a priority score that guides the CI search in $\theta$-space based on $\mathcal{GP}$ approximated p-value surface.

$$a(\theta) = \sum_{\alpha_i} \left| \frac{\sigma_{\hat{q}(\theta)}}{\hat{q}(\theta) - \alpha_i} \right|$$

- Here, $\hat{q}(\theta)$ is $\mathcal{GP}$ mean, $\sigma_{\hat{q}(\theta)}$ is $\mathcal{GP}$ uncertainty, $\alpha_i$ is chosen to be $(0.68, 0.90)$
- $a(\theta)$ balances between exploration, i.e MC experiments at new points and exploitation, i.e reducing $\mathcal{GP}$ error



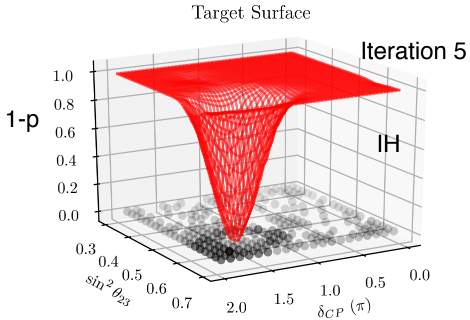



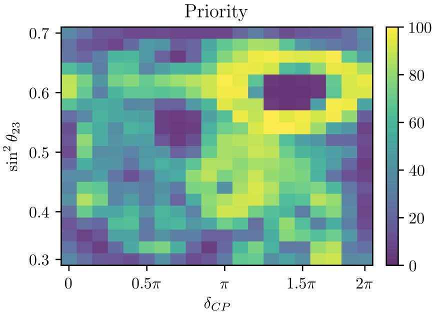
# Optimised Confidence Interval Search

- Use a priority score that guides the CI search in $\theta$-space based on $\mathcal{GP}$ approximated p-value surface.

$$a(\theta) = \sum_{\alpha_i} \left| \frac{\sigma_{\hat{q}(\theta)}}{\hat{q}(\theta) - \alpha_i} \right|$$

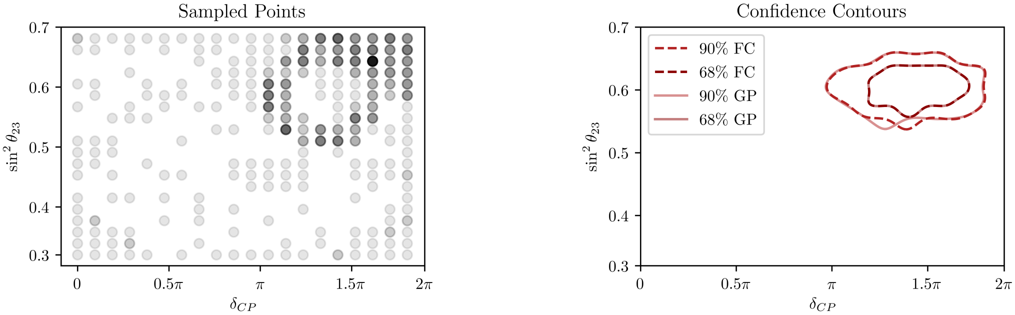
- Here, $\hat{q}(\theta)$ is $\mathcal{GP}$ mean, $\sigma_{\hat{q}(\theta)}$ is $\mathcal{GP}$ uncertainty, $\alpha_i$ is chosen to be $(0.68, 0.90)$
- $a(\theta)$ balances between exploration, i.e MC experiments at new points and exploitation, i.e reducing $\mathcal{GP}$ error



# Results

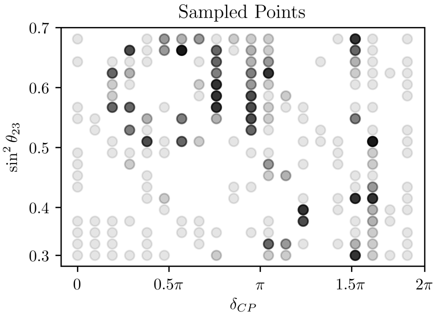
- "Real" data similar to latest best-fit estimate from NOvA. ($sin^2\theta_{23} = 0.56$, $\Delta m^2_{32} = 2.44 \times 10^{-3} eV^2$, $\delta_{CP} = 1.5\pi$)
- $sin^2\theta_{23} - \delta_{CP}$ 68% and 90% CI for IH after 5 iterations

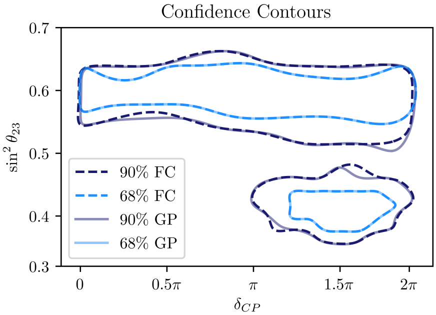



- Grayscale denotes number of experiments thrown in relation to FC (2000)
- Algorithm does a good job of finding the FC contour edge!

# Results

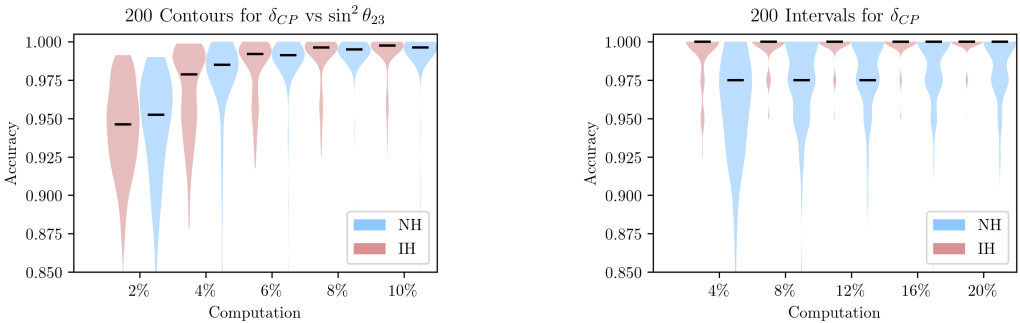
- "Real" data similar to latest best-fit estimate from NOvA. ($sin^2\theta_{23} = 0.56$, $\Delta m^2_{32} = 2.44 \times 10^{-3} eV^2$, $\delta_{CP} = 1.5\pi$)
- $sin^2\theta_{23} - \delta_{CP}$ 68% and 90% CI for NH after 5 iterations



# Results

- 200 different runs for "real" data at the same point as before.
- Use classification accuracy of all grid points, taking FC result as truth, to evaluate performance.
- Progress shows the search algorithm converges to the FC value $\sim 10\times$ faster for 2D case and $\sim 5\times$ for 1D case




- Median Accuracies for 1D is 100%, for 2D is $> 99.5\%$ (both NH, IH)
- Mean Accuracies for 1D is 98.5% (99.8%) for NH (IH), for 2D is $> 99\%$ (both NH, IH)

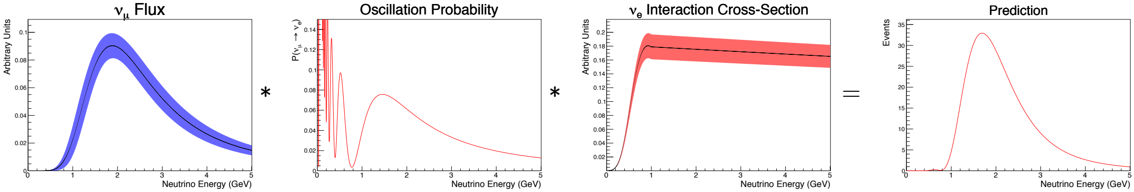
# Summary and Conclusions

- Neutrino oscillation experiments provide interesting test case for estimating frequentist confidence intervals
- LBL experiments typically proceed via Feldman-Cousins
- However, simulating $\Delta\chi^2$ distributions across multi-dimensional parameter space requires huge computational resources
- We've studied a Bayesian approach using Gaussian processes on a toy LBL set-up
- Helps us estimate frequentist contour edges to quite a high accuracy without having to sample the entire parameter space!
- Order of magnitude gain in computation!
- See PRD publication for more details : Phys.Rev.D 101 (2020) 1, 012001
- All code with illustrative notebooks here : `https://github.com/nitish-nayak/ToyNuOscCI`, maintained by Lingge (linggeli7@gmail.com) and myself (nayakb@uci.edu)

# Backup

# Toy Experiment

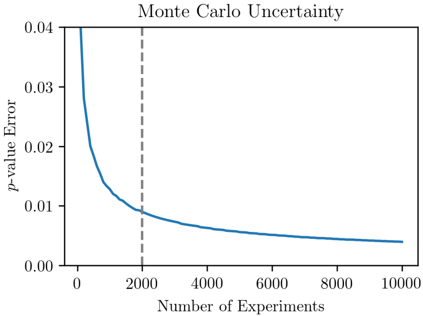
- Modelled on NOvA. Baseline, $L = 810$km with $\nu_\mu$ flux peaking at 2GeV
- $\nu_\mu \rightarrow \nu_e$ by multiplying toy shapes for flux, cross-section and oscillation probability.
- 10% normalisation errors on flux and xsec model




- $P(\nu_\mu \rightarrow \nu_e)$ using 3-flavor PMNS with MSW corrections added for matter propagation.
- Similar setup for $\nu_\mu \rightarrow \nu_\mu$ to constrain $sin^2(2\theta_{23})$ and $|\Delta m^2_{32}|$ but with 2-flavor approximation
- $P(\nu_\mu \rightarrow \nu_\mu) \sim 1 - sin^2(2\theta_{23})sin^2(\Delta m^2_{32} L/4E)$

# $\mathcal{GP}$s for FC

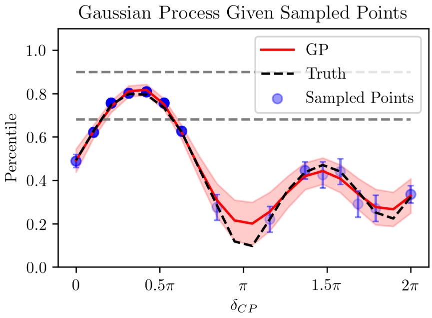
- Fitting a GP to target p-value surface for a given contour. (Stochasticity of the target surface)
- "Observation" at a given point in parameter space, $\theta$ means simulating the LRT distribution and finding the p-value of $crit(\theta)$
- Choose a RBF Kernel with an additional term incorporating variance of p-value estimate at $\theta$.

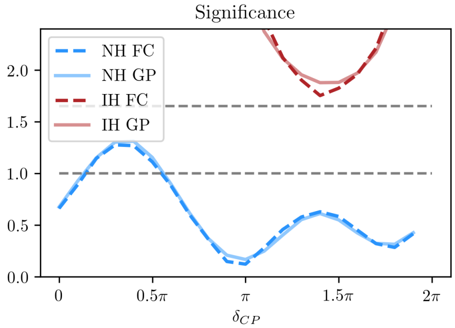



- $k(\cdot,\cdot) = k_{RBF}(\cdot,\cdot) + \sigma_p^2 I$
- The additional variance encodes the binomial error resulting from throwing finite number of experiments to simulate the LRT distribution at $\theta$
- Allows us to incorporate varying number of experiments thrown into the CI search, reducing computational burden further.

# Results

- "Real" data similar to latest best-fit estimate from NOvA. ($sin^2\theta_{23} = 0.56$, $\Delta m^2_{32} = 2.44 \times 10^{-3} eV^2$, $\delta_{CP} = 1.5\pi$)
- Significance of rejecting $\delta_{CP}$ only after 5 iterations. (p-value converted to Z-score significance)



# $\mathcal{GP}$ Technical Details

- Rasmussen and Williams has a good discussion about convergence to true functions in regression settings (typically using squared loss functions) : `http://www.gaussianprocess.org/gpml/chapters/RW7.pdf`
- Well behaved $\implies$ expressible as a generalised fourier series of kernel eigenfunctions
- If kernel is non-degenerate, approximation is guaranteed to converge to true function
- If degenerate, convergence towards an $L_2$ approximation of the true function
- Rates of convergence typically depends on mean and kernel smoothness as well as smoothness of the true function

# $\mathcal{GP}$ Fitting

- Hyperparameters (**w**) learned via maximising log marginal likelihood :

$$p(\mathbf{y}|\mathbf{X},\mathbf{w}) = \int p(\mathbf{y}|\mathbf{X},\mathbf{w},\mathbf{f})p(\mathbf{f}|\mathbf{X},\mathbf{w})d\mathbf{f}$$

- Clearly,

$$\mathbf{f}|\mathbf{X},\mathbf{w} \sim \mathcal{N}(\mathbf{0}, K(\mathbf{X},\mathbf{w}))$$

- Some algebra gives us :

$$-2\log p(\mathbf{y}|\mathbf{X},\mathbf{w}) = \mathbf{y}^T K^{-1}\mathbf{y} + \log|K| + n\log 2\pi$$

- Minimising above equation gives us a good choice for **w**
- $log|K|$ acts as a penalty term for complexity and therefore reduces overfitting to data

# $\mathcal{GP}$ for FC

- "Gaussian" not a statement of the underlying distribution of the test statistic, which can still be heavily non-Gaussian
- Rather, "Gaussianity" for a stochastic process generating the test statistic distributions. Stochasticity mostly from finite FC grid resolution or finite number of MC experiments for simulating the test statistic distribution
- Assumption we're making for this stochasticity is that it can be parameterised by a kernel describing the relationship between the distributions at neighbouring points $\implies$ multi-variate gaussian

- Also important to note, no real statement about FC coverage or handling of nuisance parameters. Assumes FC gives desired level of coverage
- Confidence Intervals still with frequentist interpretation
- Bayesian interpretation for "classification probability" of points in parameter space for desired confidence regions
- A good summary would be "Accelerating Frequentist CI search by estimating CI edges through Bayesian ML"

# $\mathcal{GP}$s in Literature

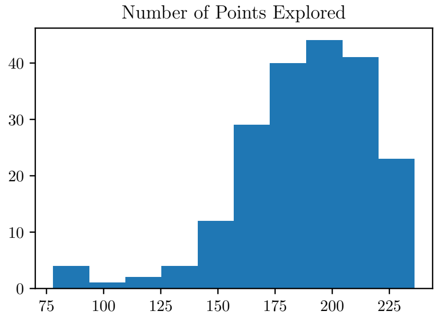
- $\mathcal{GP}$s in HEP : arXiv:1709.05681, M. Frate, K. Cranmer et al. Using $\mathcal{GP}$s to describe background spectra in dijet resonance searches at the LHC non-parametrically.
- Used in Astrophysics for modelling stochasticity of light yields in stars, active galactic nuclei etc
- Many other fields!

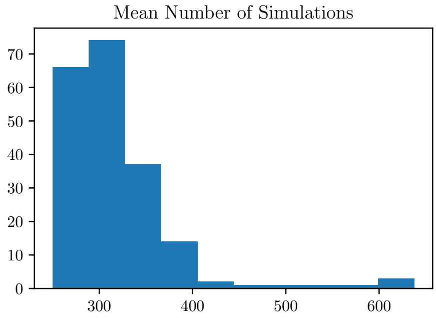
# Pseudo-code

**Algorithm 1** $\mathcal{GP}$ iterative confidence contour finding

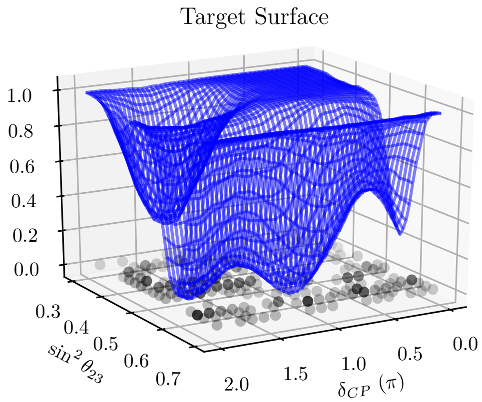
```
for each iteration t = 1, 2, ... do
    Propose new points in parameter space arg max_θ a(θ)
    for each point θ' do
        Simulate likelihood ratio distribution
        for k = 1, 2, ... do
            Perform a pseudo experiment
            Maximize the likelihood with respect to (θ, δ)
            Maximize the likelihood with constraint θ = θ'
        end for
        Obtain critical value c(θ')
    end for
    Update GP approximation ĉ(θ)
    Update confidence contours
end for
```

# Results : NH, $sin^2\theta_{23} - \delta_{CP}$

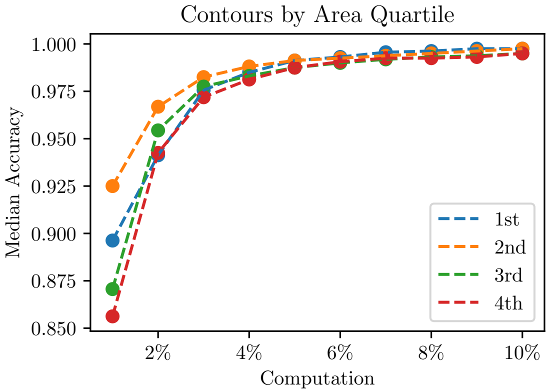


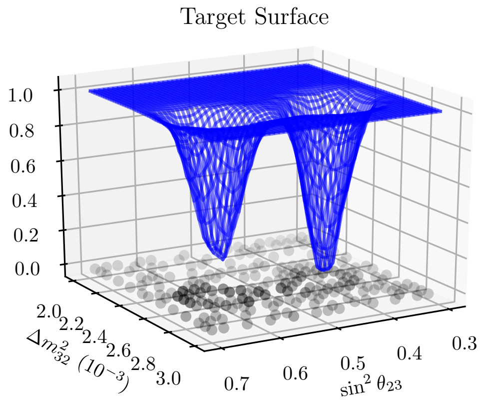
# NH, $sin^2\theta_{23} - \delta_{CP}$



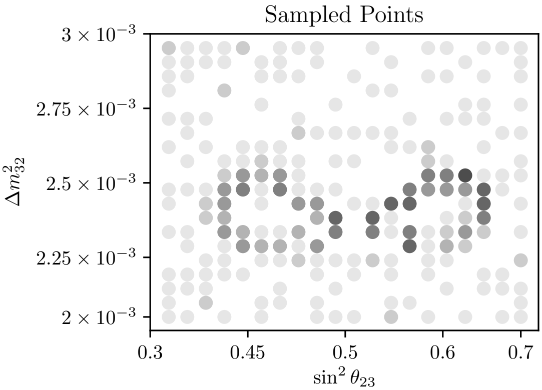
# NH, $sin^2\theta_{23} - \delta_{CP}$



# NH, $sin^2\theta_{23} - \Delta m^2_{32}$



# NH, $sin^2\theta_{23} - \Delta m^2_{32}$



# NH, $sin^2\theta_{23} - \Delta m^2_{32}$