Nested sampling neural ratio estimation (NSNRE)

*On-going work / work in progress

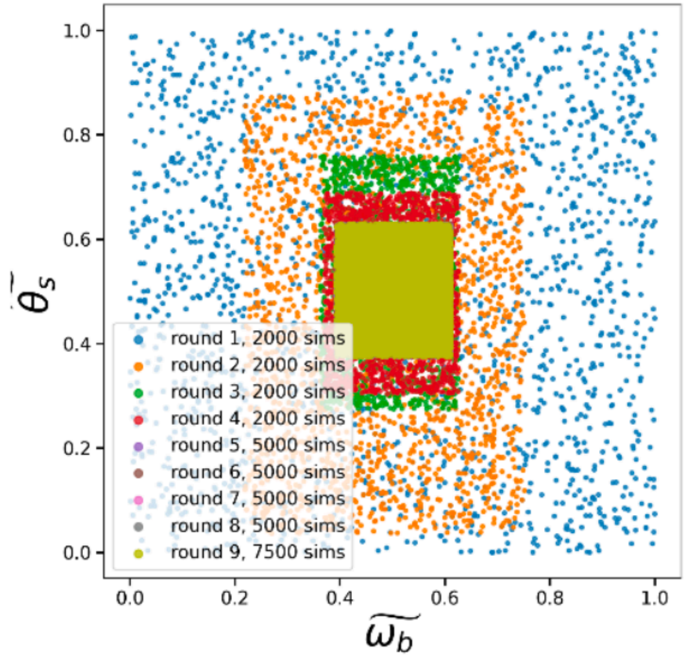
Kilian Hikaru Scheutwinkel, 3rd year Physics Ph.D, University of Cambridge

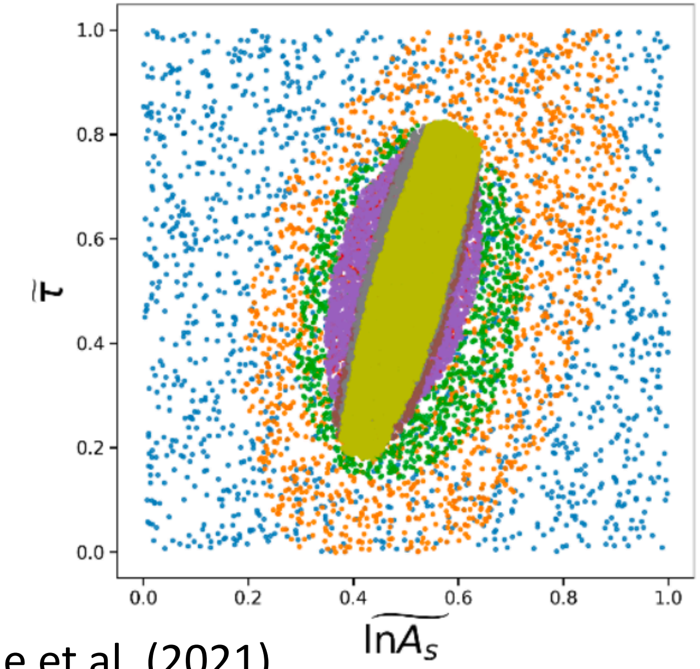
Supervised by Eloy de Lera Acedo & Will Handley

So far….

We saw what the swyft algorithm does:
an implementation of (TM)NRE by creating "boxes" within the prior space, that define the truncated prior region

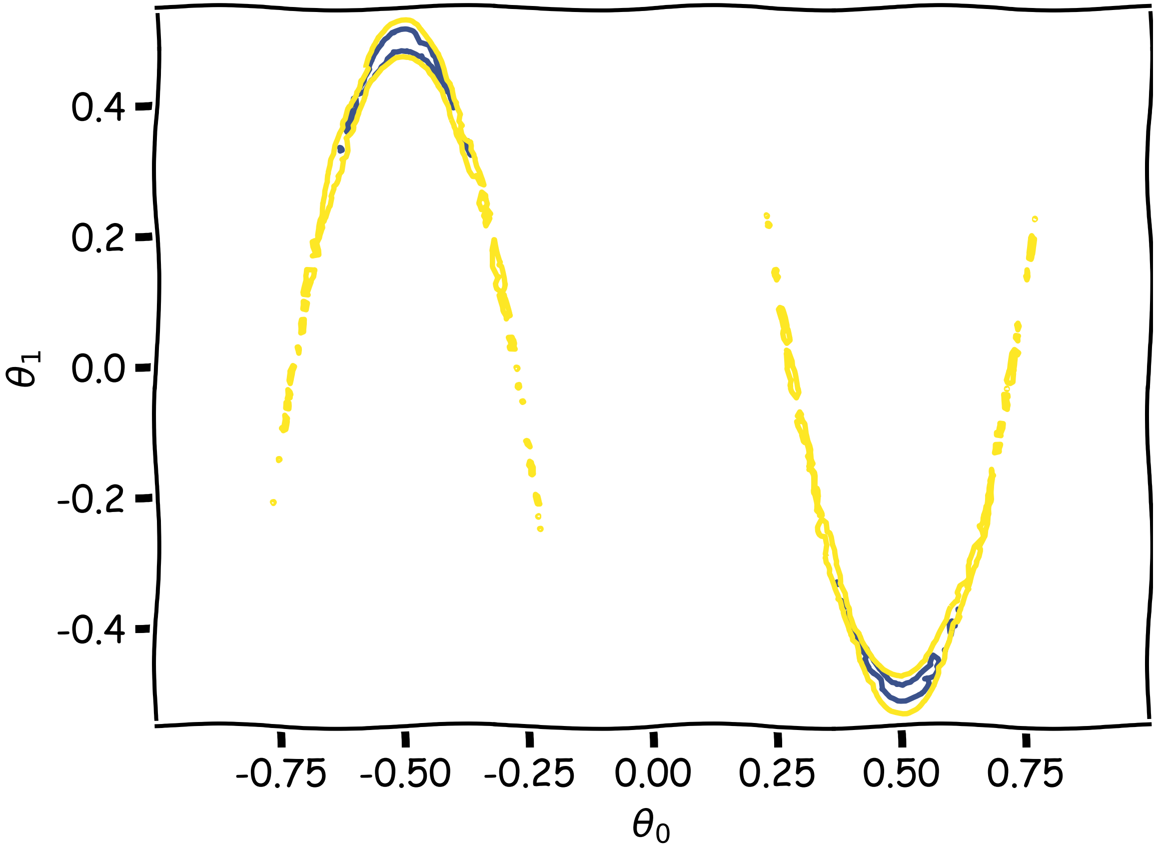
The truncated region allows concentrated sampling i.e. "zooming in" to parameter spaces that are more useful for our observation x_0

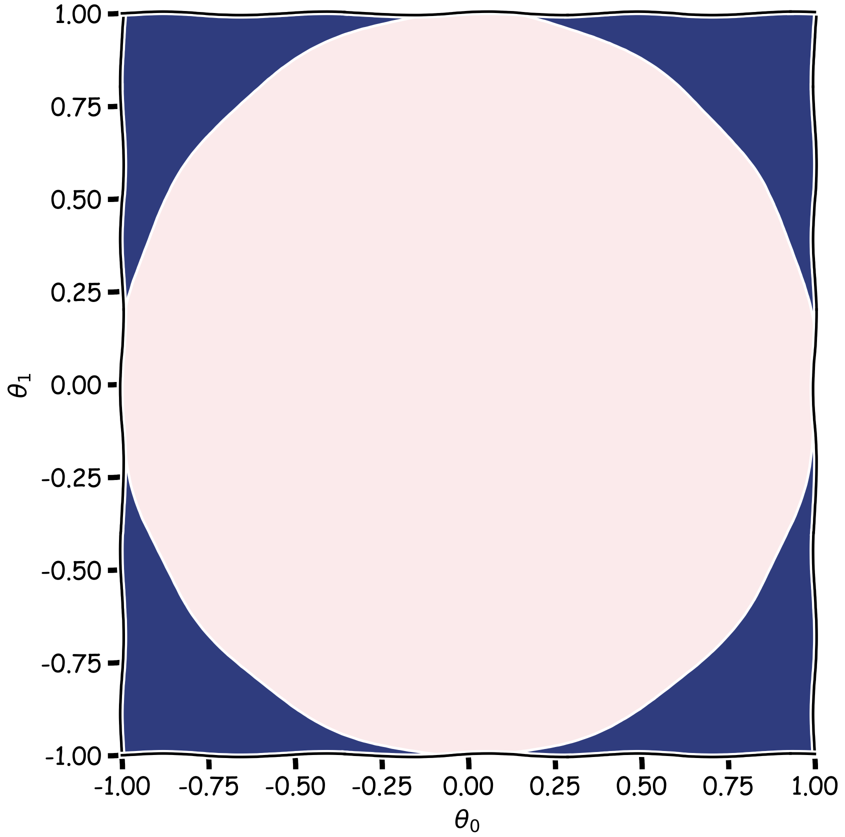




Cole et al. (2021)

Limits of the prior truncation scheme





1) Multimodality

2) Curse of dimensionality $\lim_{n\to\infty} \frac{V_n r^n}{(2r)^n} \to 0$

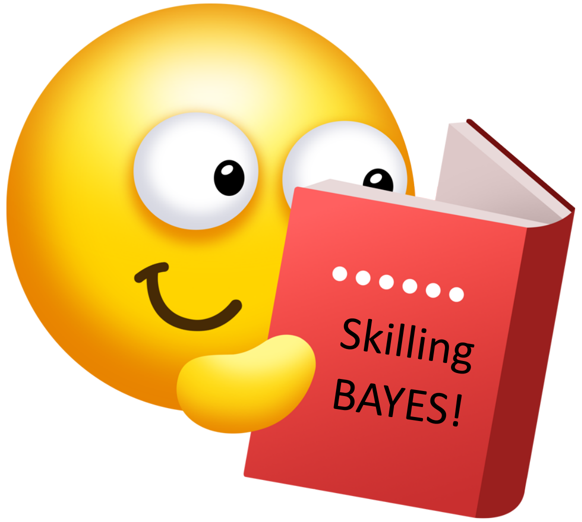
Rising questions regarding TMNRE

1) Can we define a truncation scheme that does not explicitly truncate the prior?

Yes! We can explore the likelihood-over-evidence "weights" space instead!

2) Has something similar already been done?

Yes, we need to take a look at nested sampling!



The idea of nested sampling (NS)

NS solves:

$$Z = \int_{\Theta} L(\theta)\pi(\theta)d\theta = \int_0^1 \mathscr{L} dX \text{ or more in general: } \int f(\theta)dV$$

by creating prior mass shells $dX = \pi(\theta)d\theta$

$dX :=$ fraction of prior mass $L > L^*$

Skilling (2006)

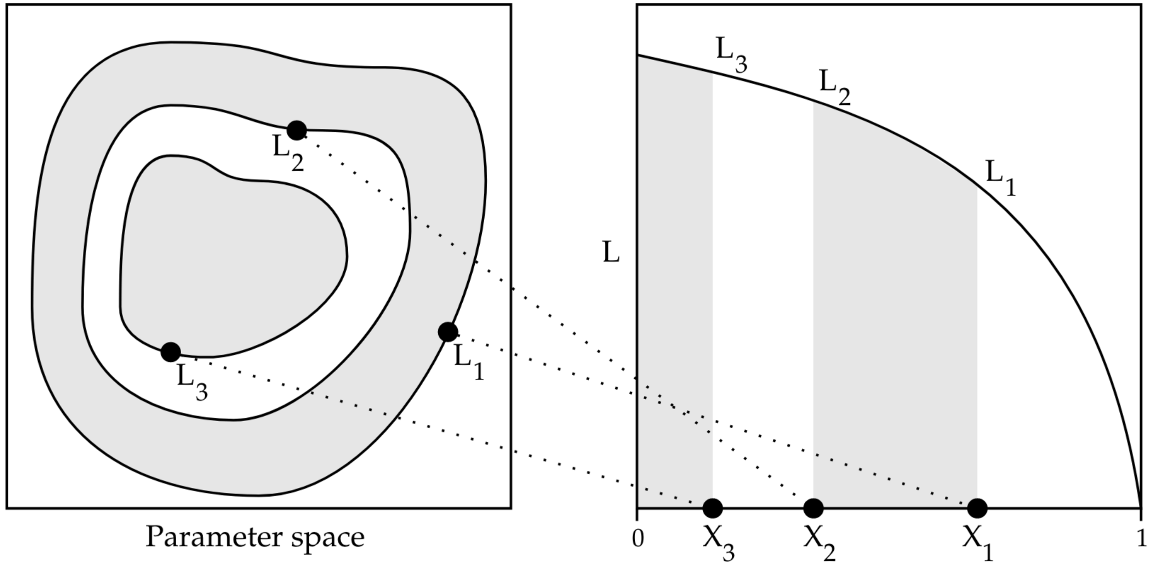


Figure 3: Nested likelihood contours are sorted to enclosed prior mass X.

UNIVERSITY OF CAMBRIDGE

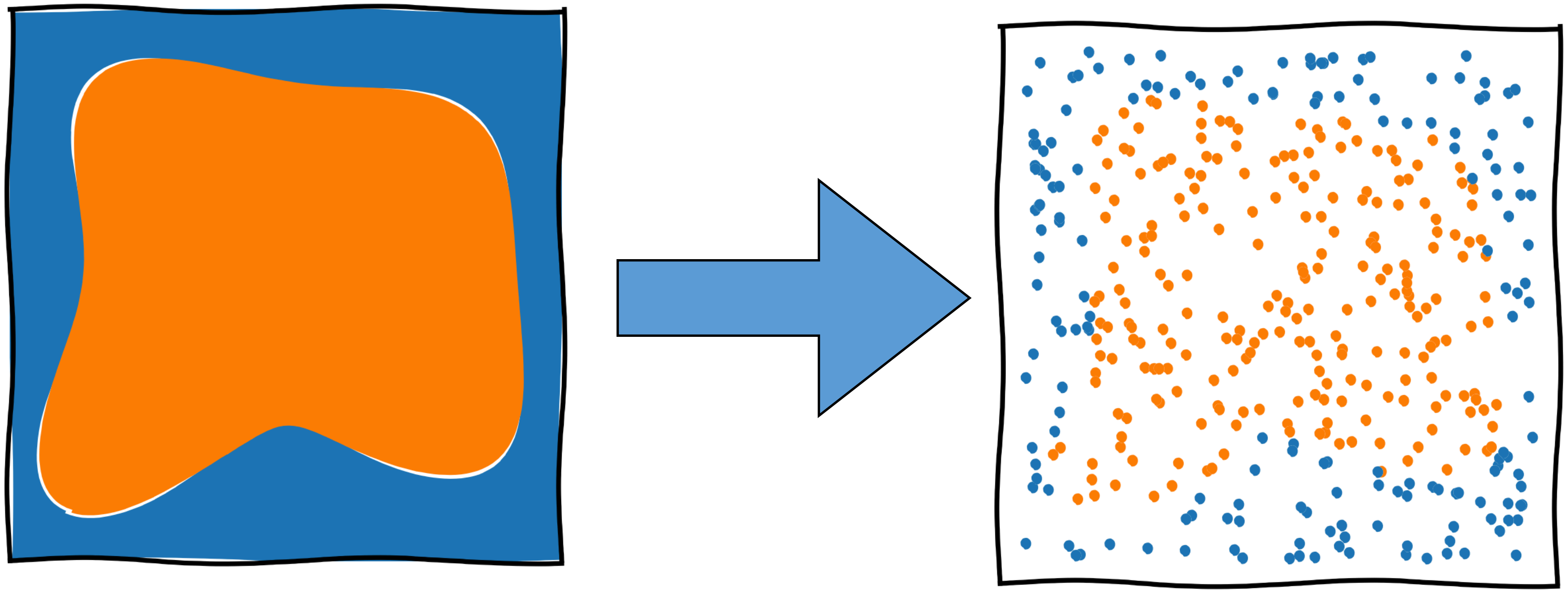
In practice

NS finds samples $\{\theta_i\}$ from prior subject to $L(\{\theta_i\}) > L^*_{i-1}$ using some sampler e.g. Metropolis-Hastings or Slice sampling
This dynamic likelihood constraint define the contours (shells) within the prior!

In standard NS: $L^*_i = \min(L(\{\theta_i\}))$, the lowest likelihood point and replaces it with a new sample θ_{i+1} $\rightarrow$ "1 out and 1 in"
Repeat until some convergence criterion is fulfilled e.g. accumulated evidence $\log Z$ or stop after n_{iter} rounds

Popular nested samplers: PolyChord (arXiv:1502.01856, arXiv:1506.00171), MultiNest (arXiv:0809.3437) and others

A nice feature: volume estimation in high d through counting !



$$\frac{V_{\text{orange}}}{V_{\text{blue}}} \approx \frac{n_{\text{orange}}}{n_{\text{orange}} + n_{\text{blue}}}$$

Figures by Will Handley

NS and TMNRE similarity comparison

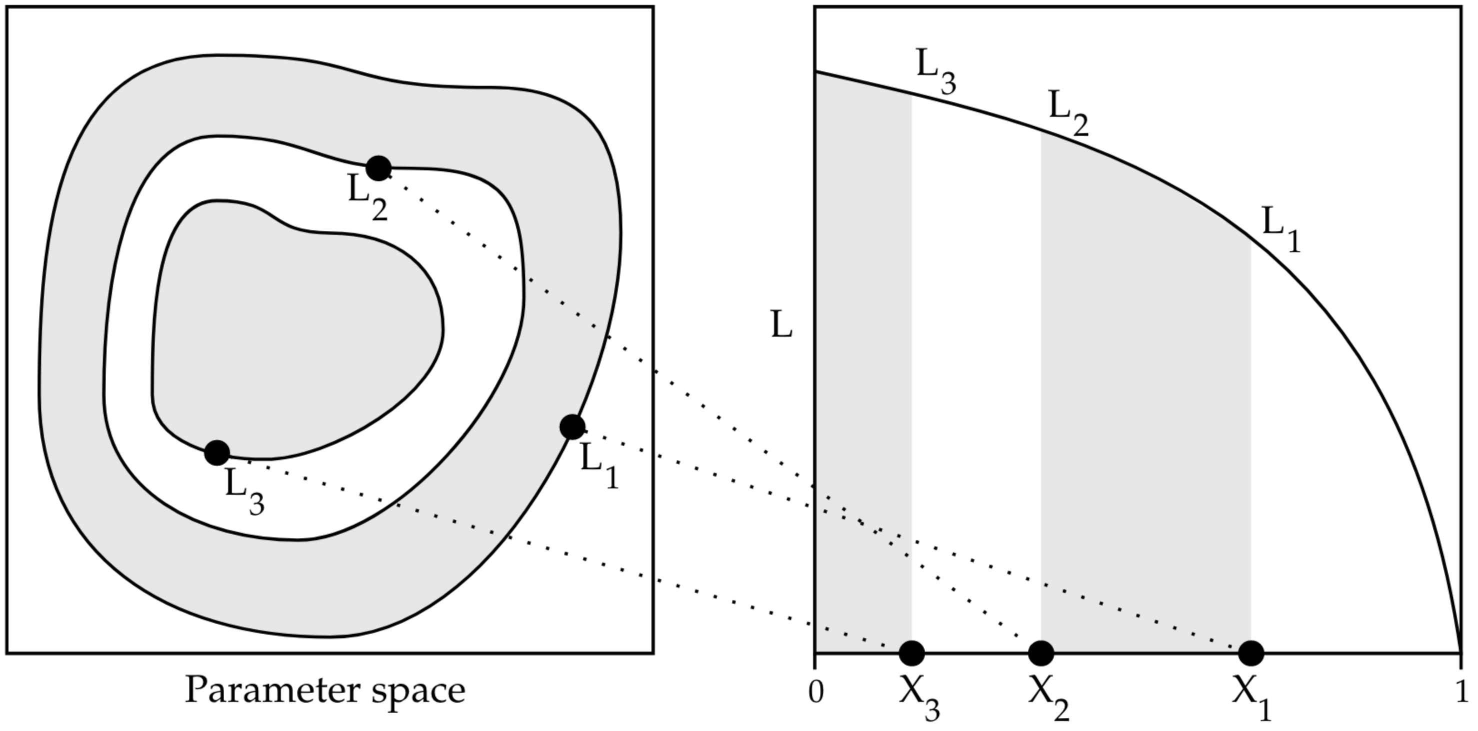
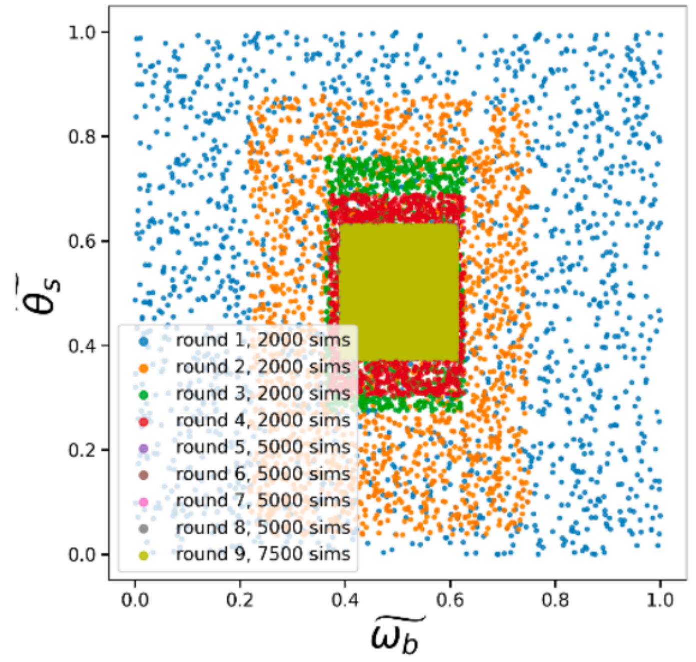
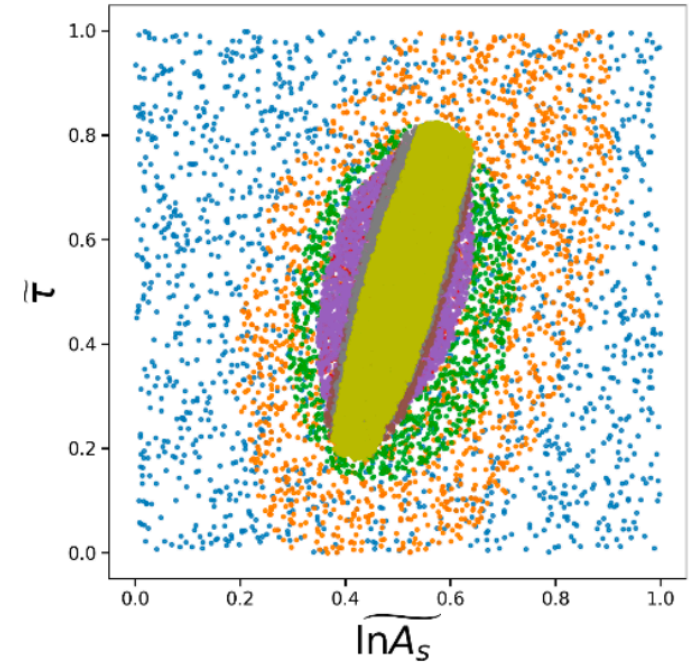


Figure 3: Nested likelihood contours are sorted to enclosed prior mass X.





TMNRE → NSNRE meta-algorithm (putting it together)

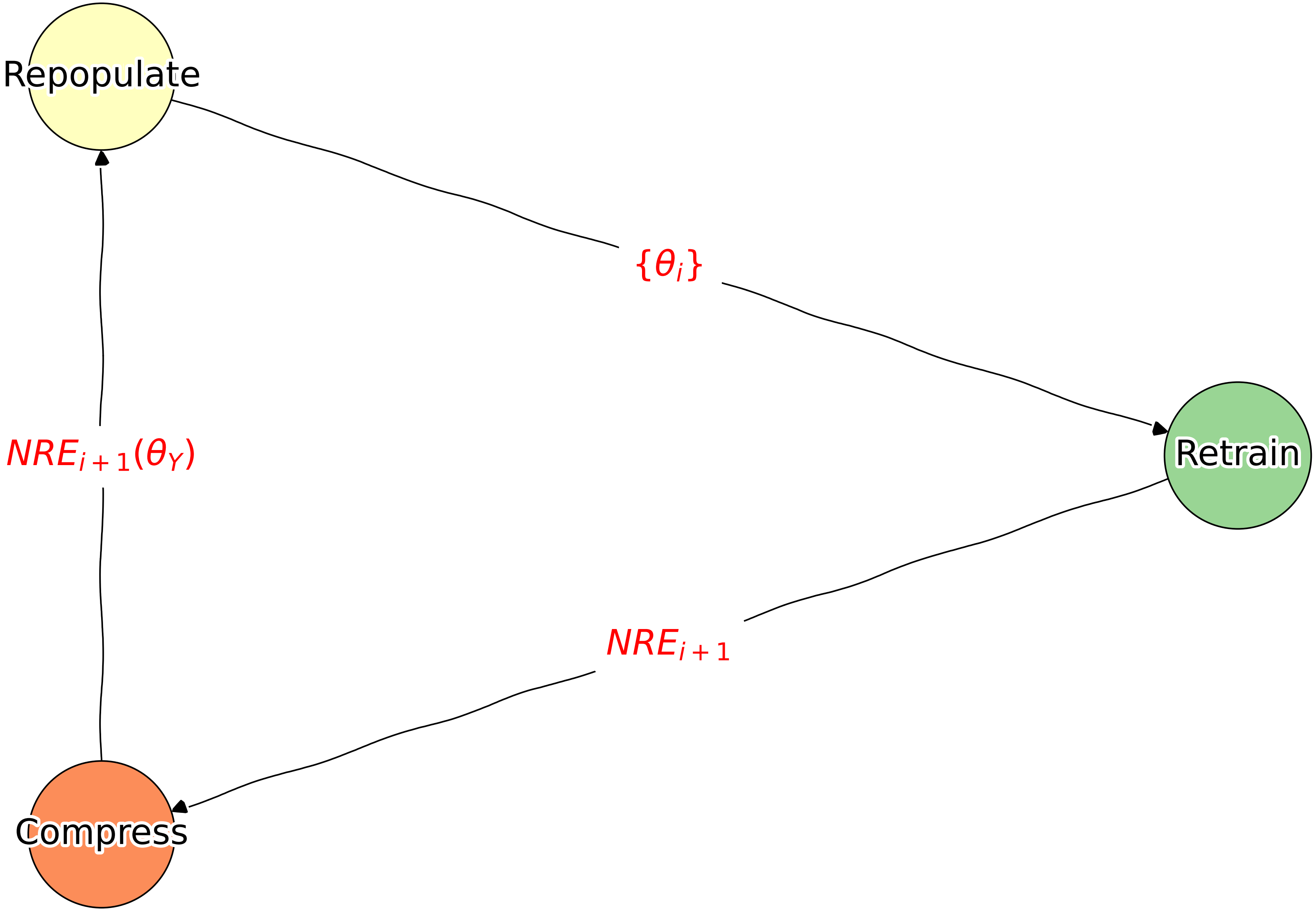
A cycle involving three key steps:

1: Repopulate (NS)

2: Retrain (NRE)

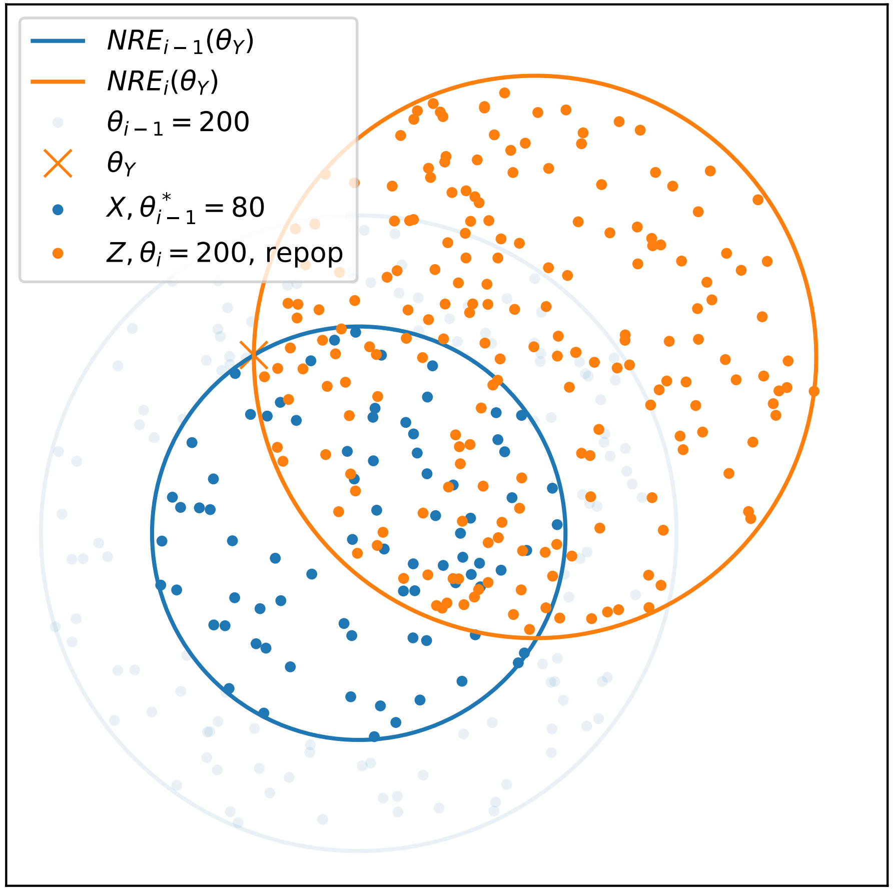
3: Compress (NRE)

Repeat



Expansive nested sampling (including a counting step within the cycle)

Goal: Keeping track of contour shifts due to retraining a new NRE



Available at any retrained step $i > 0$:

samples $\{\theta_{i-1}\}$

NRE_i and NRE_{i-1}

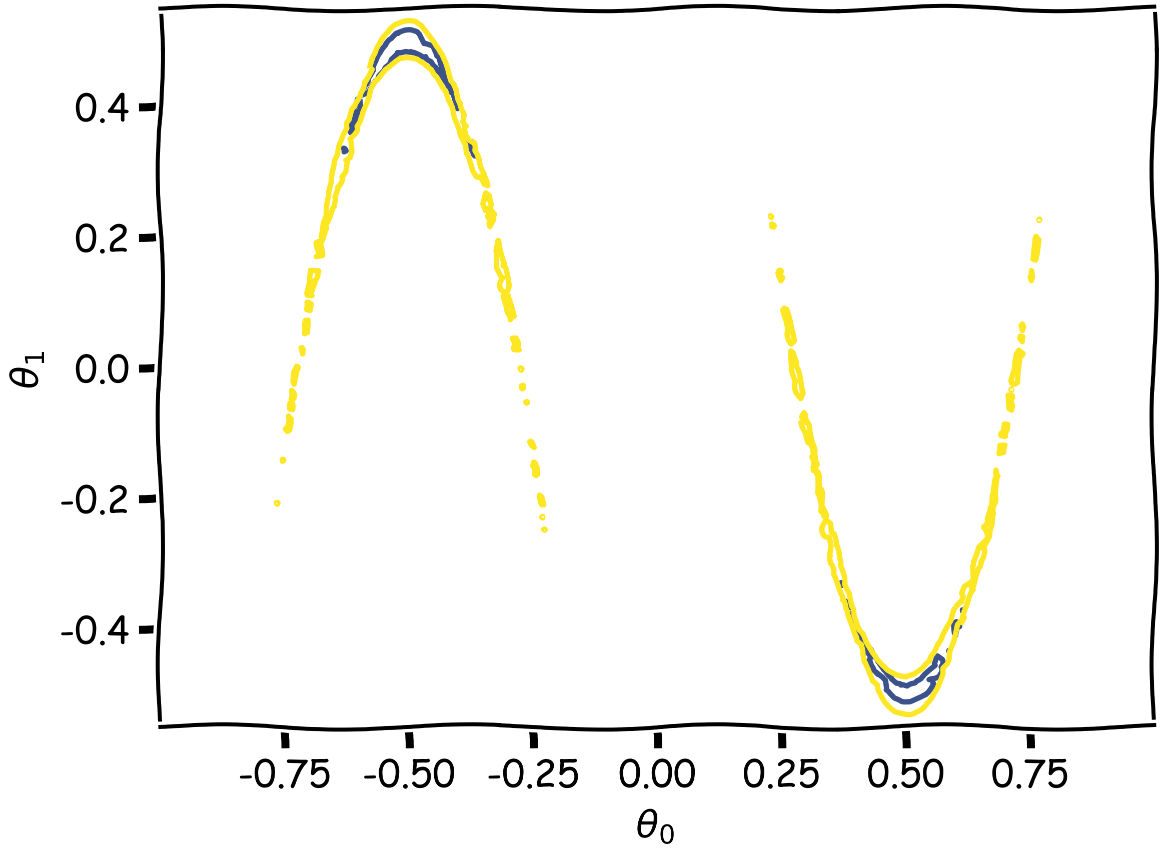
Select boundary sample θ_Y from $\{\theta_{i-1}\}$ using NRE_i

Repopulate Z using $NRE_i(\theta_Y)$

Count samples n_{i-1}, n_i in volumes X, Z and their intersection $X \cap Z$ for <u>volume estimation</u>

Some actual results

The problem to learn:



Problem statement:

given an observation x_0 and a simulator $x, \theta \sim p(x \mid \theta)p(\theta)$, use NSNRE to learn the posterior $p(\theta \mid x_0)$ on the left

We are using swyft as an NRE (note: $TMNRE \rightarrow NRE$) and (for now) a simple Slice sampling nested sampler

Later: implement PolyChord instead

How to choose boundary θ_Y?

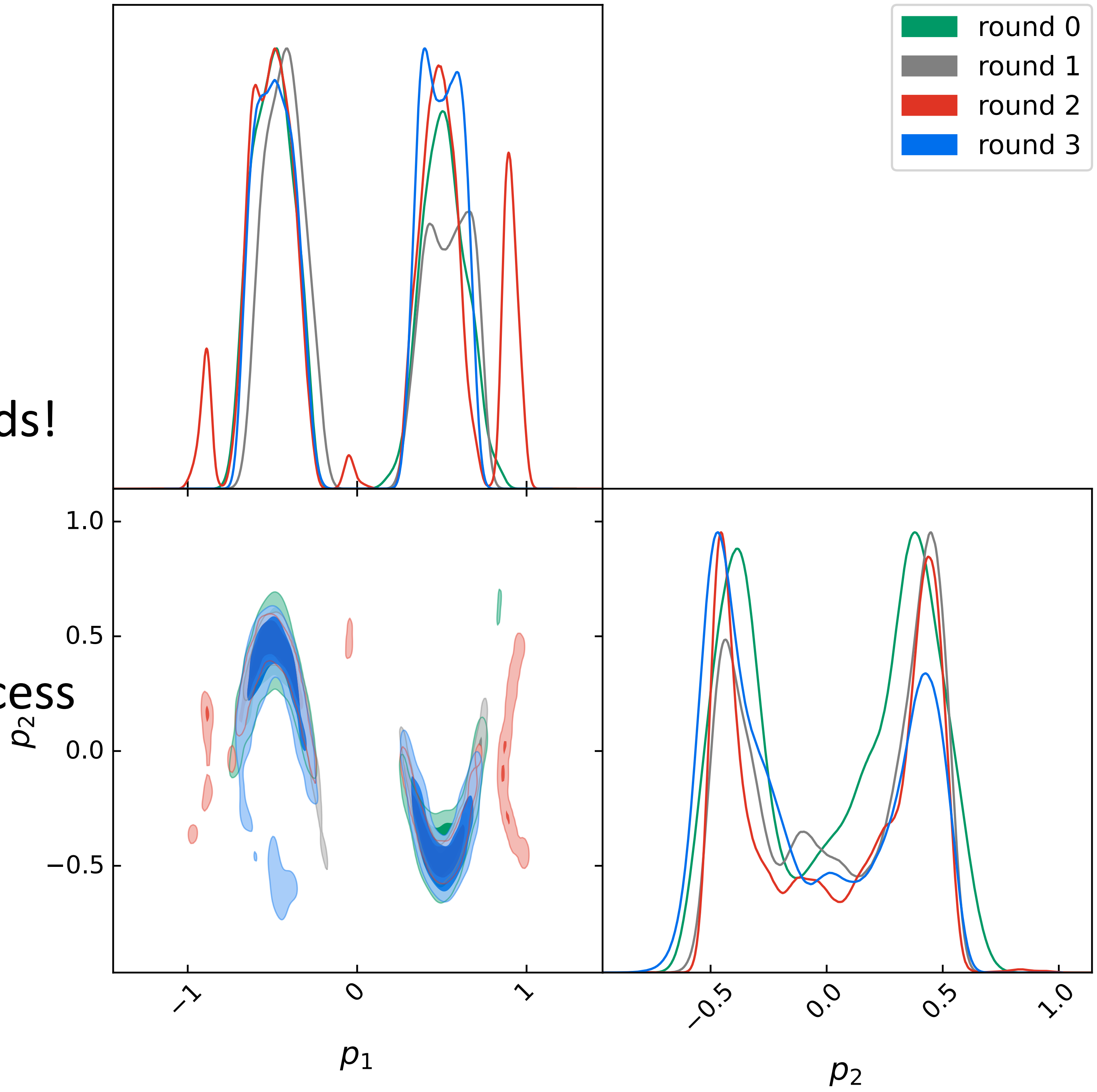
For now, we truncate using the $\theta_Y = median(NRE_{i+1}(\{\theta_i\}))$

NSNRE results

Using a manual $n_{\text{rounds}} = 3$ setup:

In principle can “zoom-in” more rounds!

Note: Counting mechanism
has not been included yet
into the meta-algorithm decision process



UNIVERSITY OF CAMBRIDGE

Open research questions to think about

1) What could be an appropriate truncation scheme to find θ_Y?
2) How to define the convergence criterion of the algorithm? Explicit round based, or volume based using counting mechanism?
3) Should we include all samples in a final retraining step to mitigate these odd contour effects? Or just omit sample evaluations outside the trained sample space?

Research supported by