Symbolic Density Models of One-in-a-Billion Statistical Tails via Importance Sampling and Genetic Programming

Trent McConaghy

Solido Design Automation Inc., Canada
Genetic Programming Theory and Practice
Ann Arbor, MI, May 2010

solido
DESIGN AUTOMATION

Tails

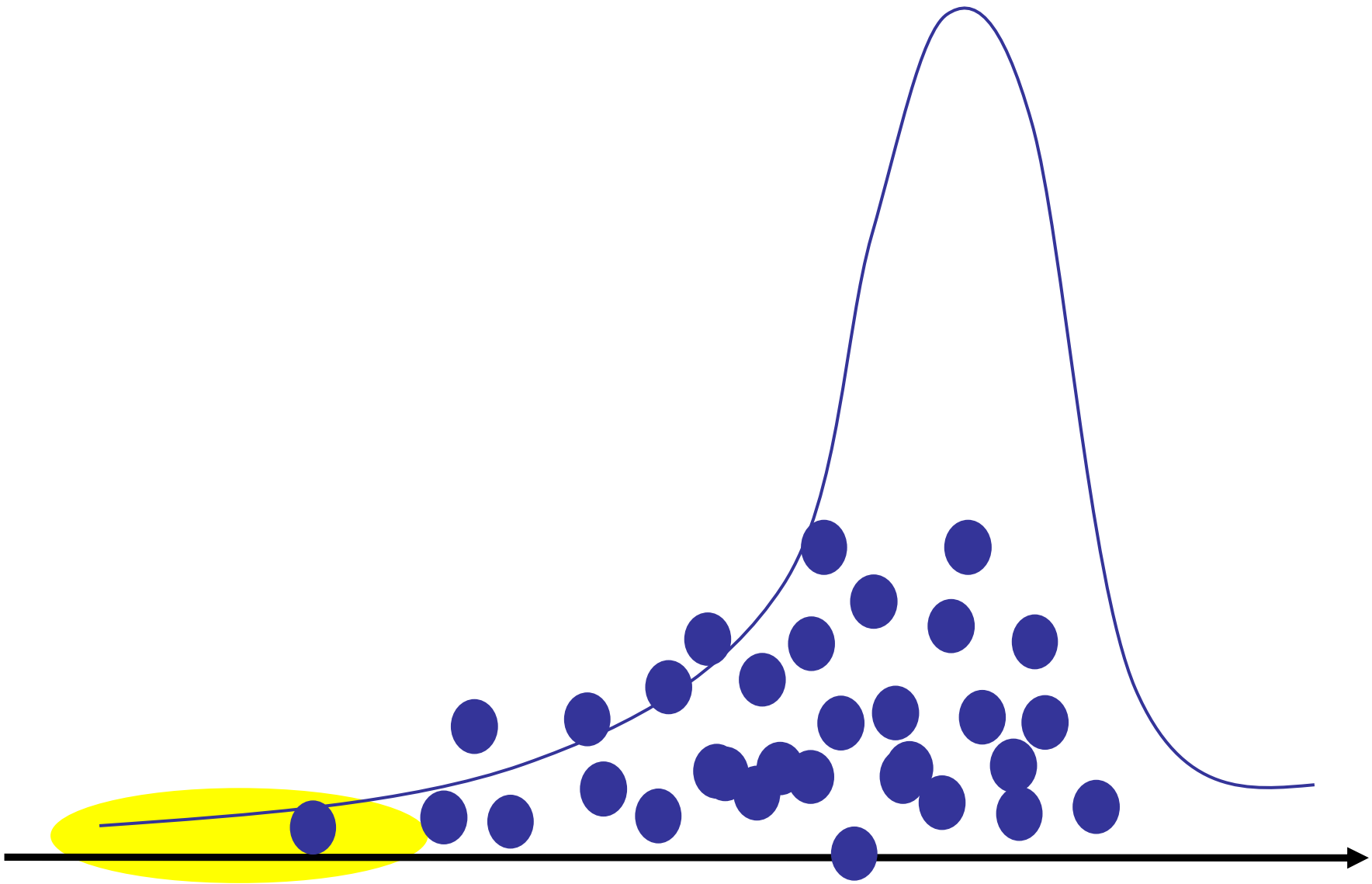
Tails

Tails

Tails

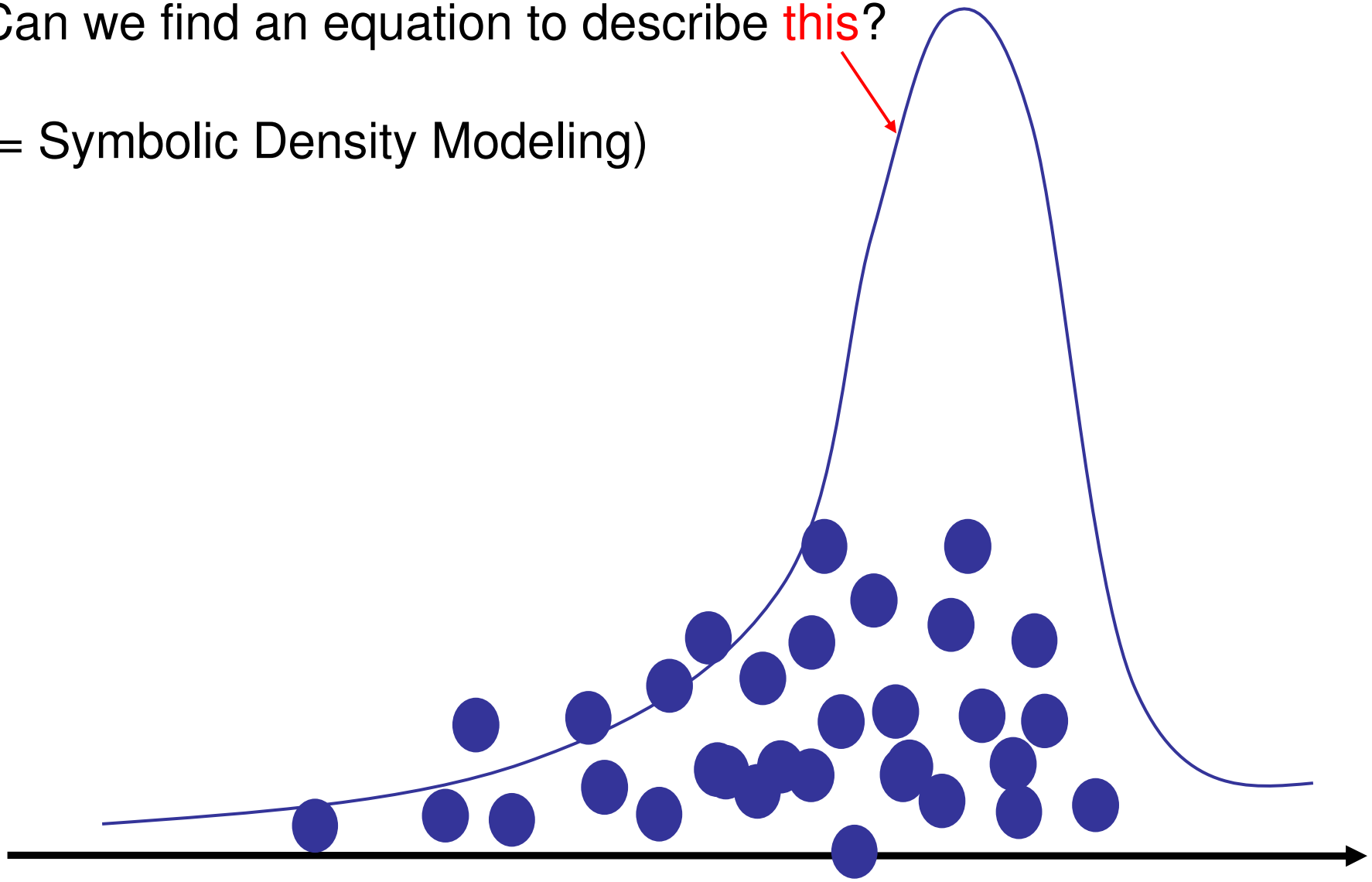
Tails

Tails



Can we find an equation to describe this?

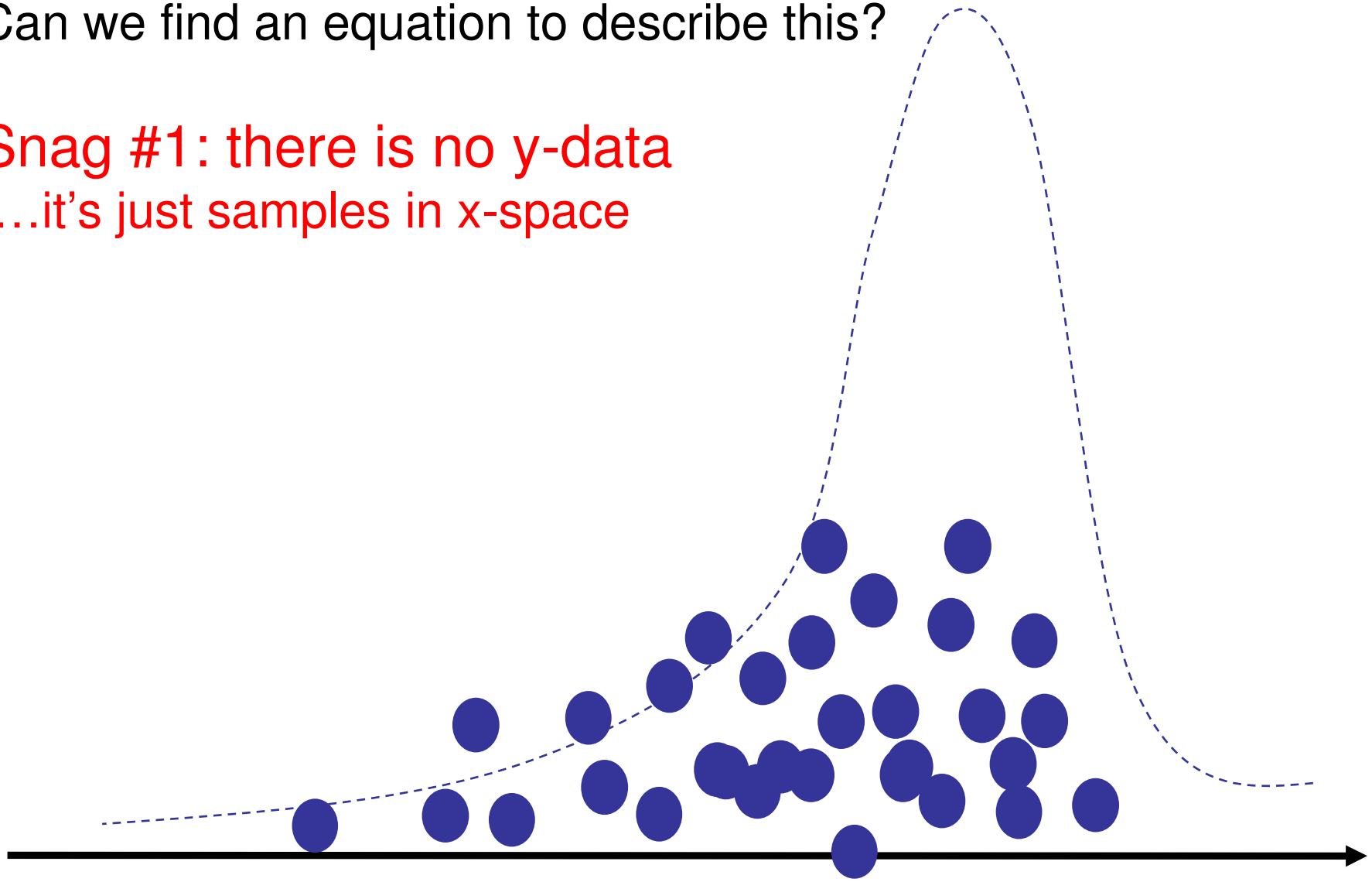
(= Symbolic Density Modeling)



Can we find an equation to describe this?

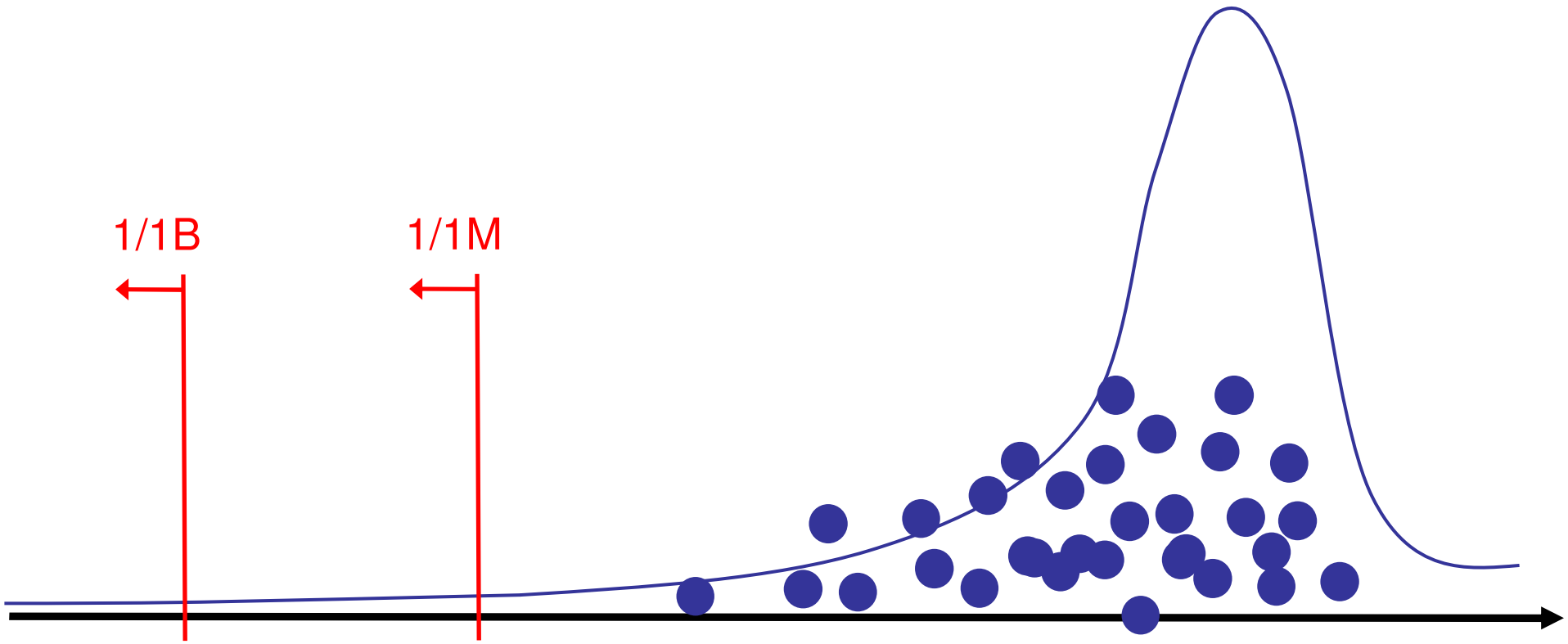
Snag #1: there is no y-data

…it's just samples in x-space



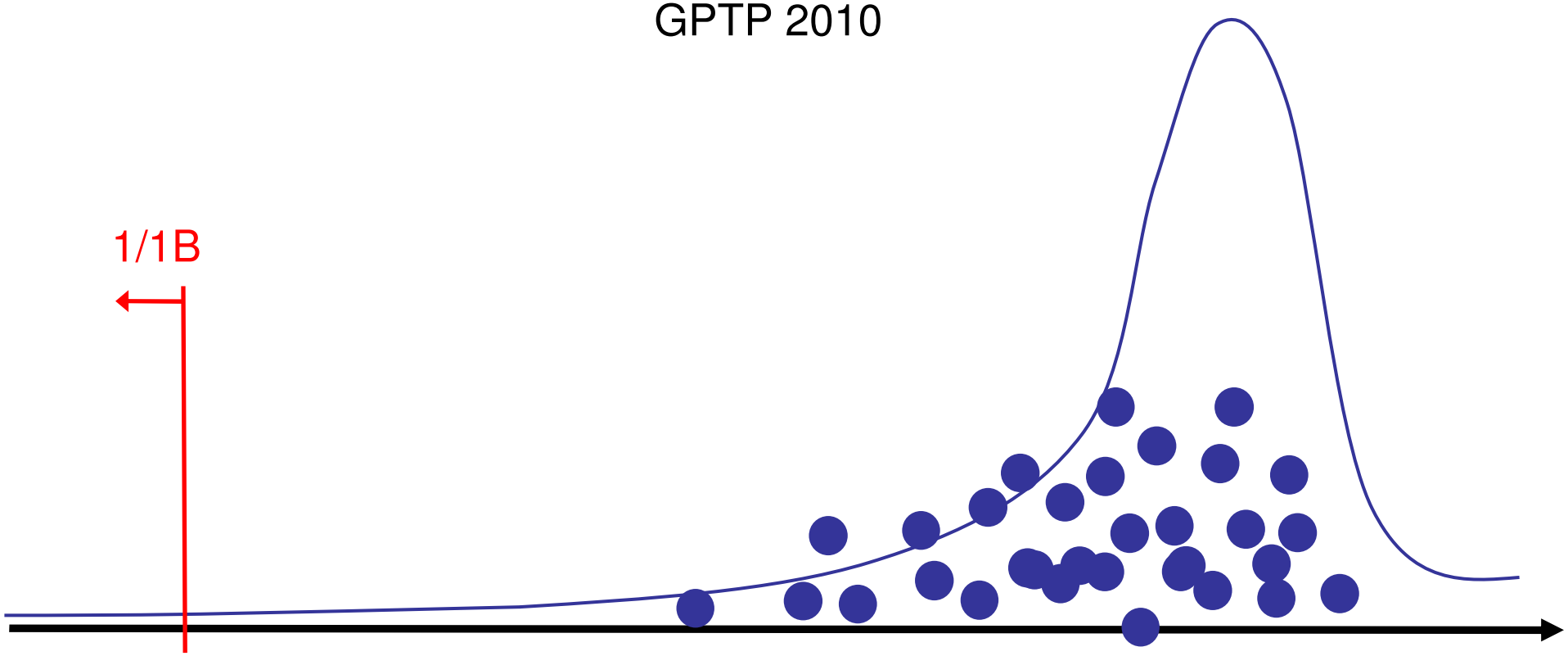
Snag #2: by definition, statistical tails are improbable

- can we generate 1B samples?
- can we model-fit 1B samples?



Symbolic Density Models
Of One-in-a-Billion Statistical Tails

Trent McConaghy
GPTP 2010



(Why) Does Anyone Care

About Statistical Tails?


iPod
Music
Videos
Photos
Podcasts
Extras
Settings
Shuffle Songs
Now Playing
MENU

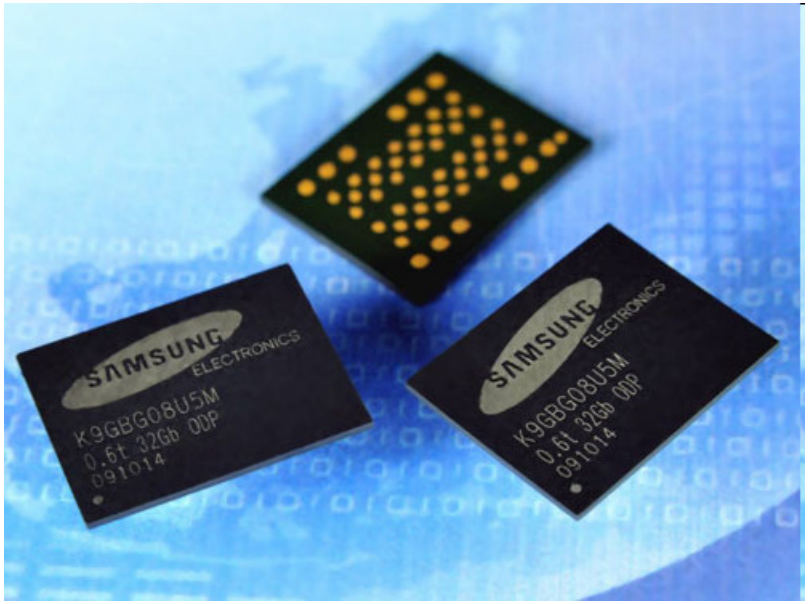

cingular
9:41 AM
Text
Calendar
Photos
Camera
Calculator
Stocks
Maps
Weather
Notes
Clock
Settings
Phone
Mail
Web
iPod


SONY
Cyber-shot
8.1
MEGA PIXELS

8GB memory

32GB memory

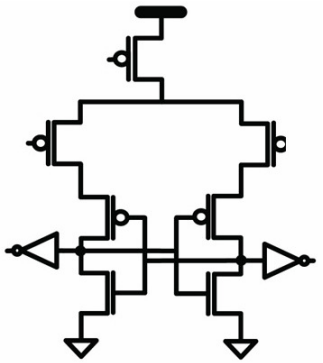
32GB memory (SD card)


SAMSUNG
ELECTRONICS
K9GBG08U5M
0.6t 32Gb 0DP
091014
SAMSUNG
ELECTRONICS
K9GBG08U5M
0.6t 32Gb 0DP
091014


iPod
Music
Videos
Photos
Podcasts
Extras
Settings
Shuffle Songs
Now Playing
MENU

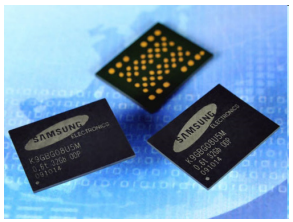

cingular
9:41 AM
Text
Calendar
Photos
Camera
Calculator
Stocks
Maps
Weather
Notes
Clock
Settings

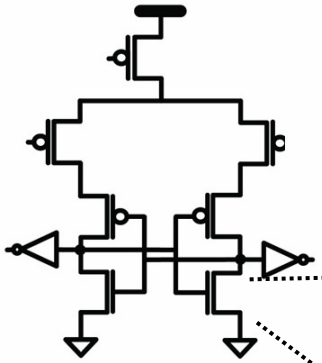

SONY
Cyber-shot
8.1

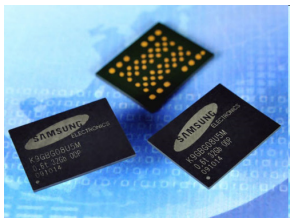


1-bit memory element (bitcell)
x 32 billion !! (for 32GB)

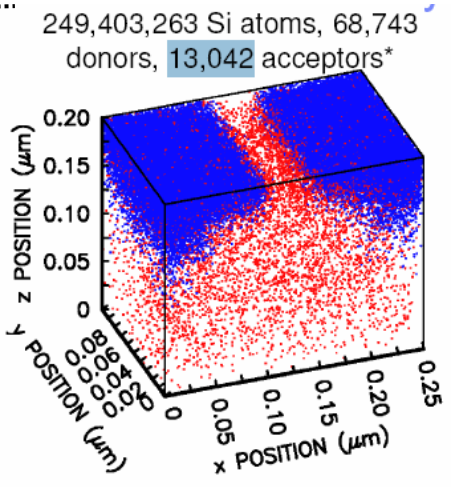
fab





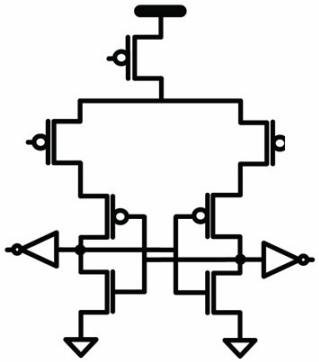


One transistor, in silicon



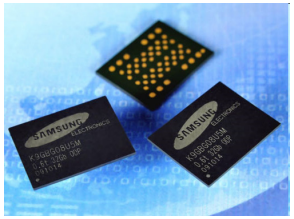
Random dopants cause performance variability, which affects yield

How to get fired!

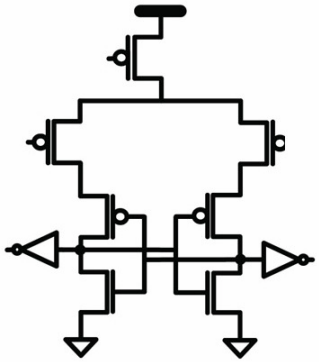


10% yield

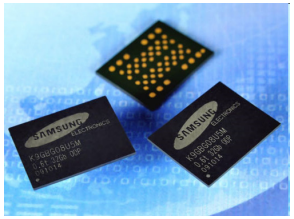
(10% of chips meet power and timing specifications)



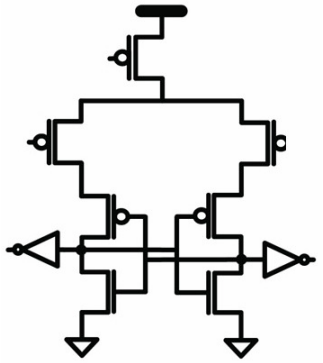
How to keep your job!

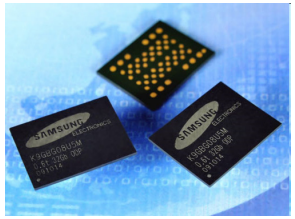


95% yield
(95% of chips meet specs)



How to keep your job!





95% yield
(95% of chips meet specs)

Therefore each bitcell must have yield of 99.999999% or higher (>6 sigma)

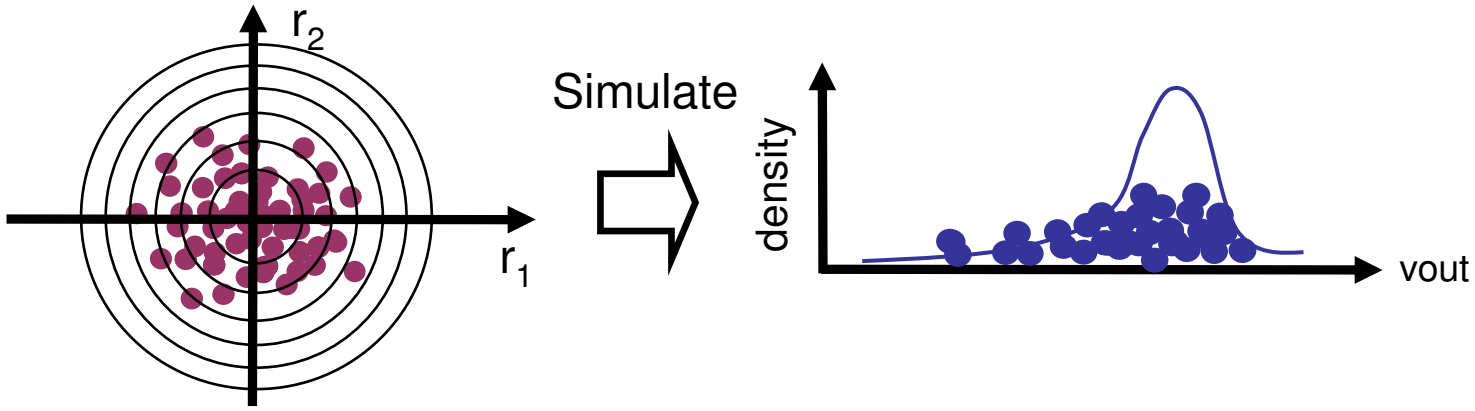
…How??

Ideal: not only measure yield, but
analyze tradeoff between yield & spec
i.e. symbolic density models

Samples of Tails

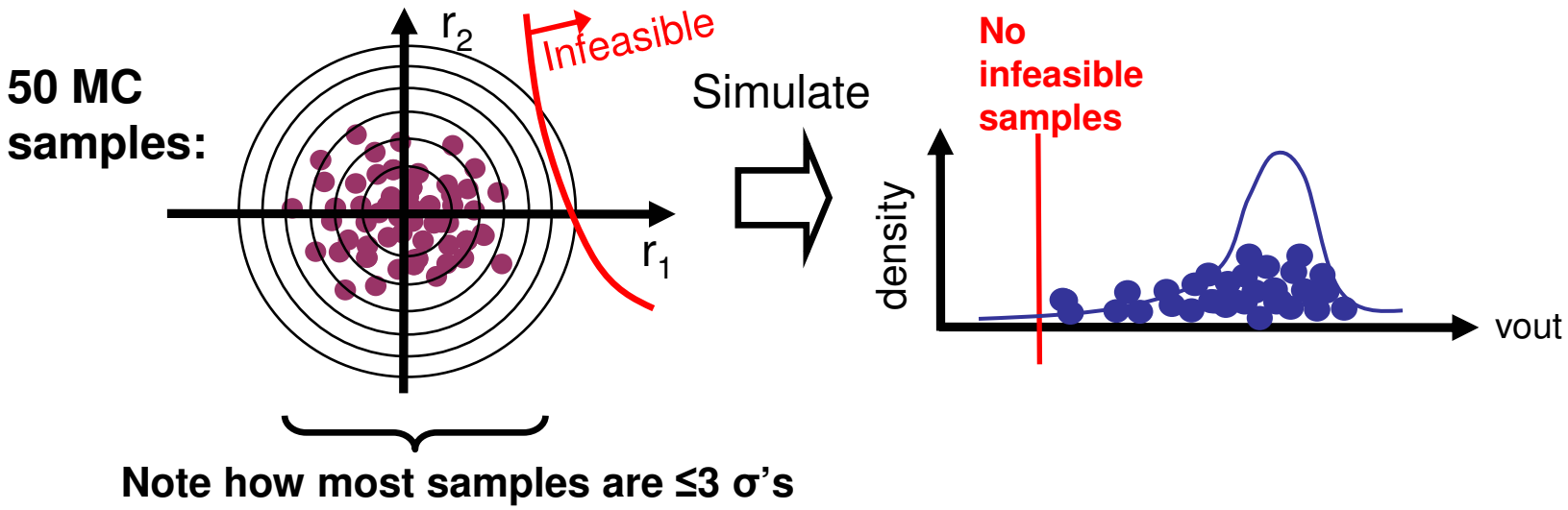
MC Sampling

- We have statistical models for the transistors
- We can simulate a bitcell circuit (using SPICE)
- Can combine these for “Monte Carlo” (MC) sampling



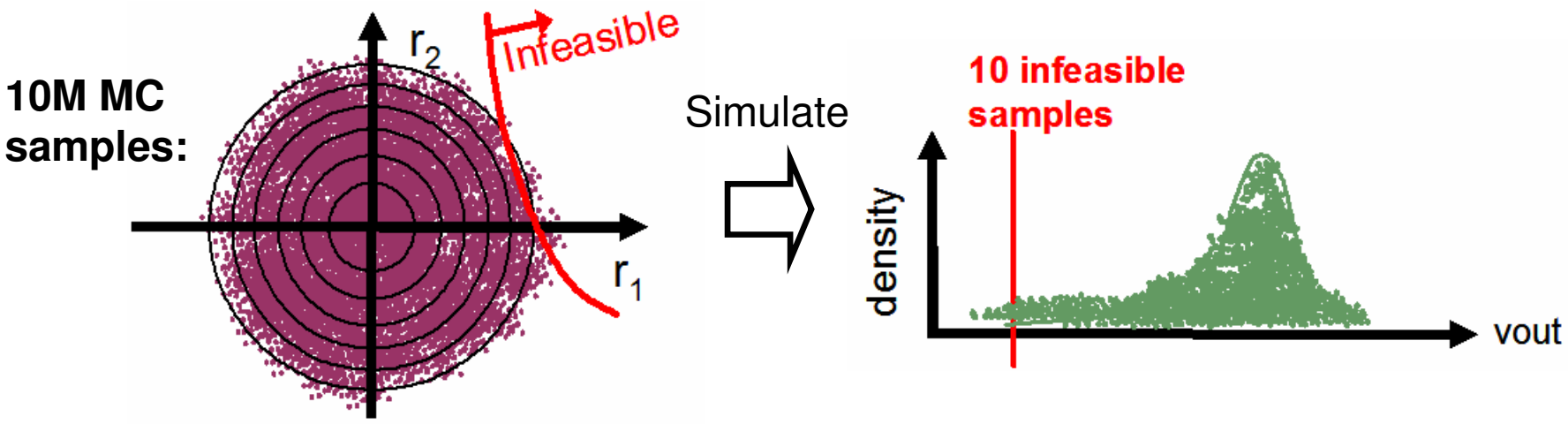
MC Sampling

- If 1/1M samples fails, then with a moderate number of samples we typically will see 0 failures
- E.g. 50/50 samples means 100% of simulations passed
- But we know yield is not 100%! (Argh!)



MC Sampling

- MC sampling would need on average 1M samples just to get 1 failure
- And for a decent yield estimate, want ≥10 failures
- Therefore would need ≥10M samples
  - Argh!!

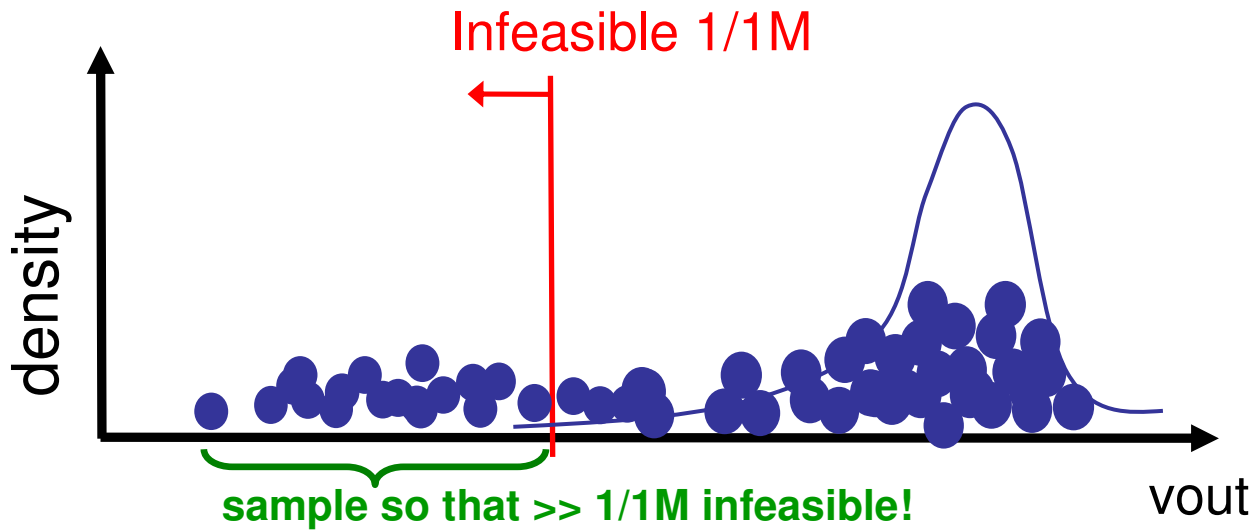


- ***Is there a practical alternative?***

Importance Sampling

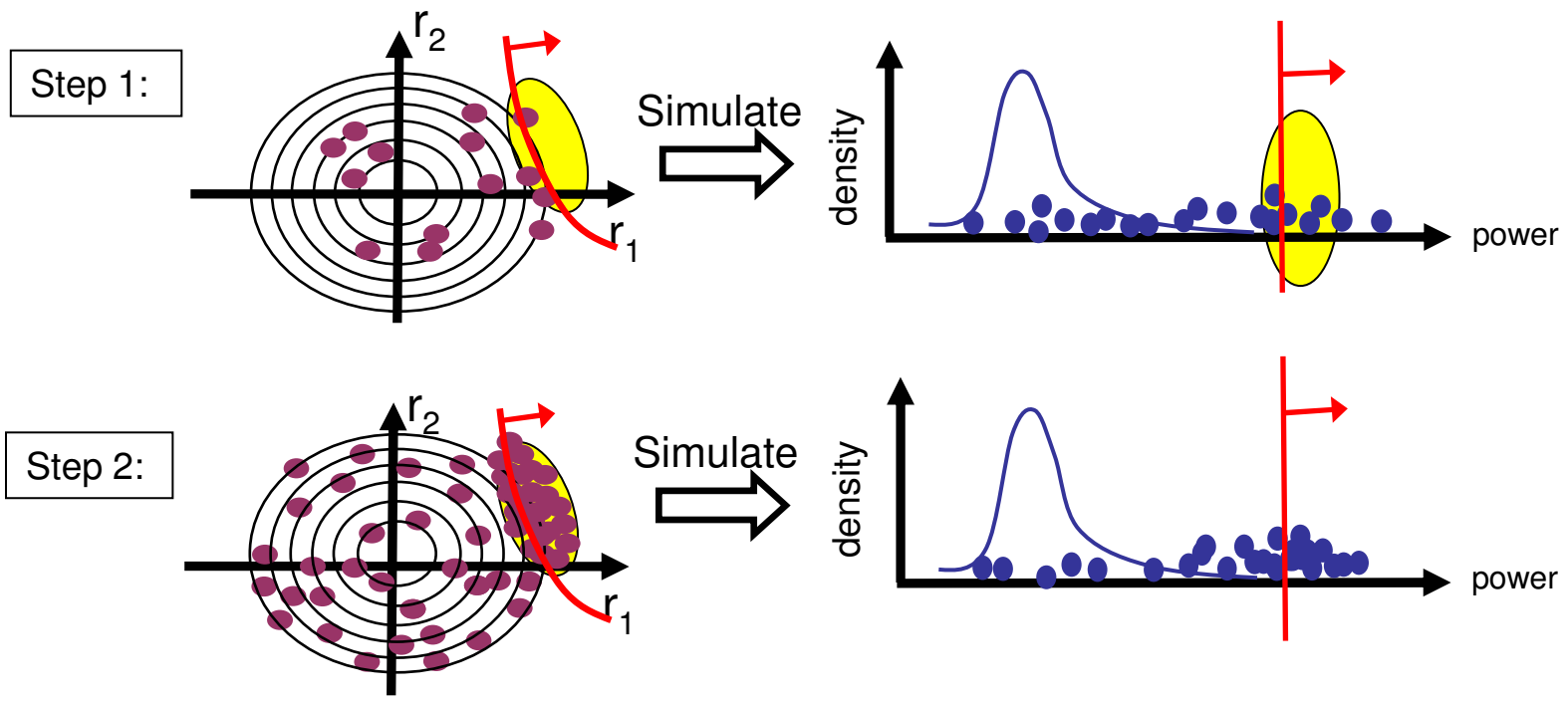
- We don't need to draw samples directly from the (true) distribution of process variations
- We just need to be *aware* of the distribution

- Instead, we can sample "wherever we want"
  - E.g. sample points with extremely low probability

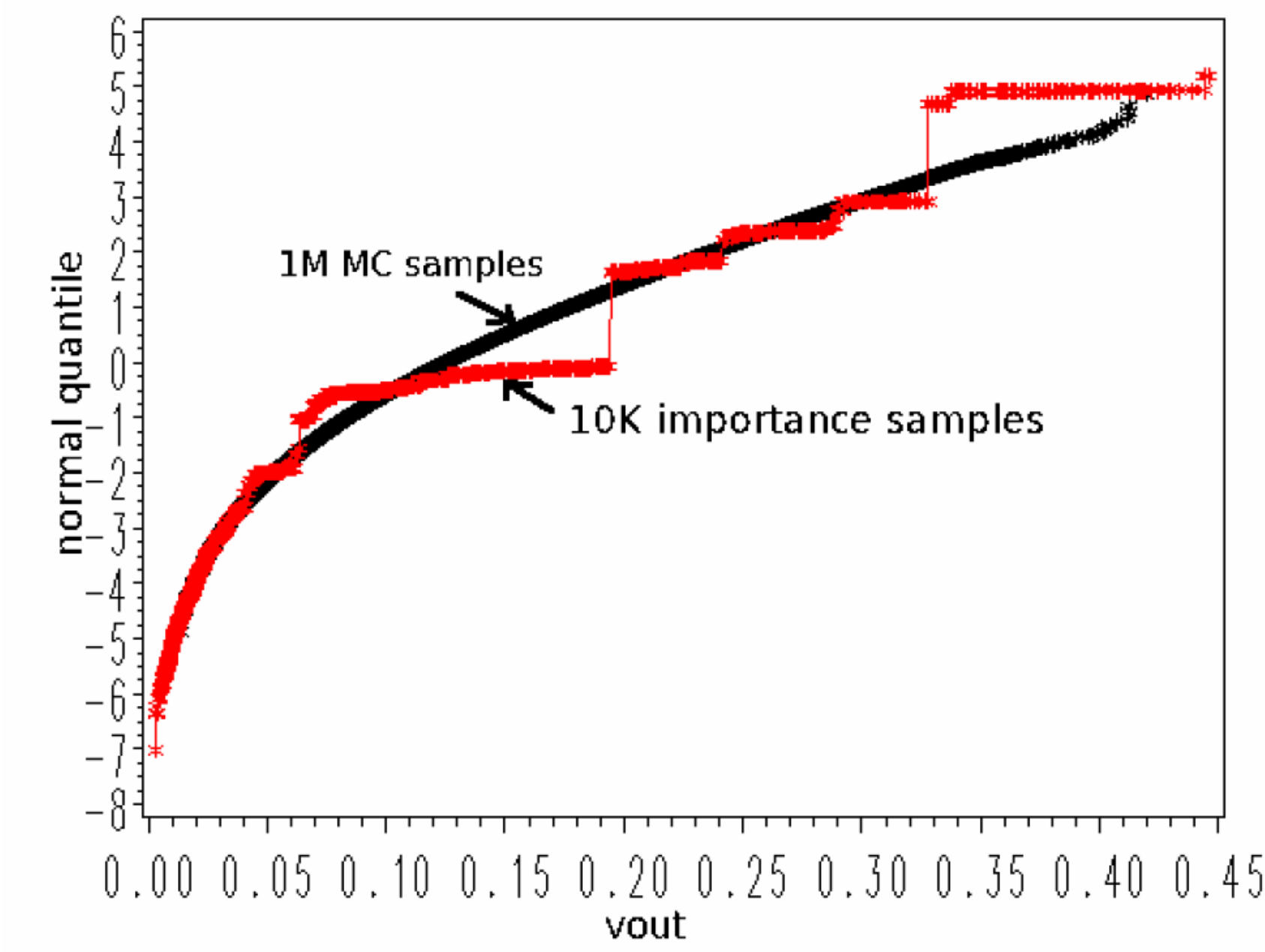


Importance Sampling

1. Find highest-probability regions of process space that cause infeas.
   - E.g. with evolutionary programming and SPICE-in-loop
2. Do a statistical sampling, but with sample pdf *biased to those regions*
   - Remember each sample's weight (= truepdf(r) / samplepdf(r))
3. Compute statistical estimates (e.g. extract pdf *somehow…*)



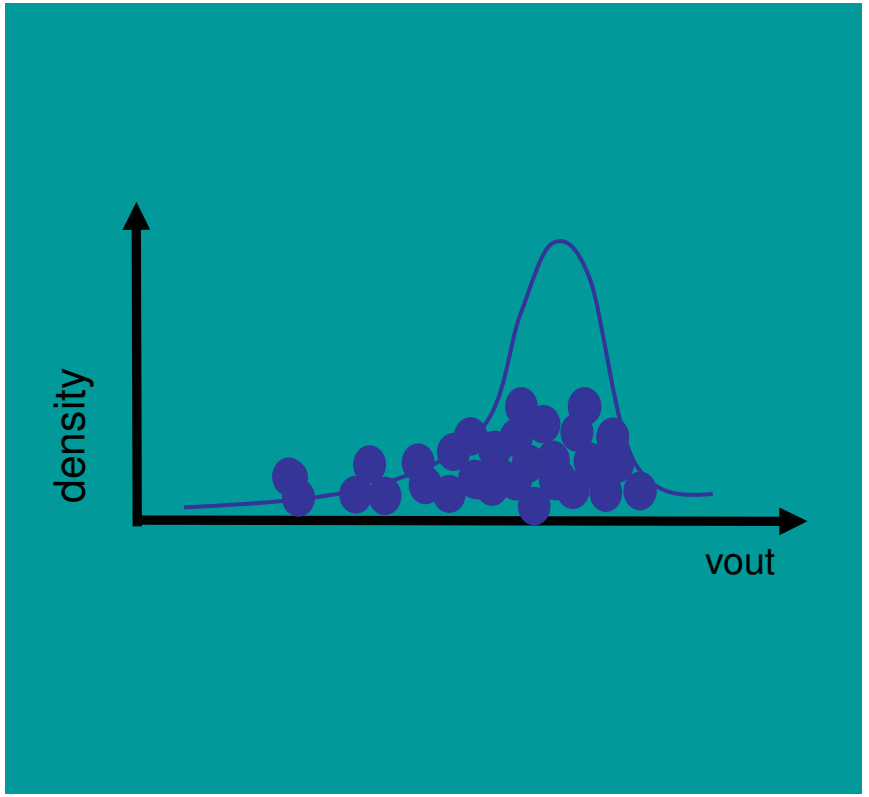
10K Importance samples vs 1M MC samples (bitcell)



Density Models

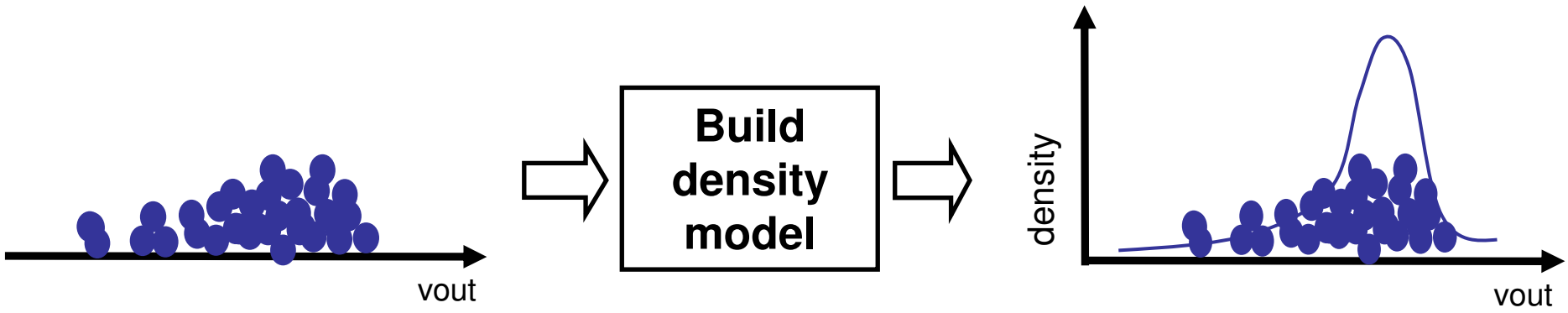
Disambiguation: Dense Models vs. Density Models

Dense model



Density model

Density Estimation

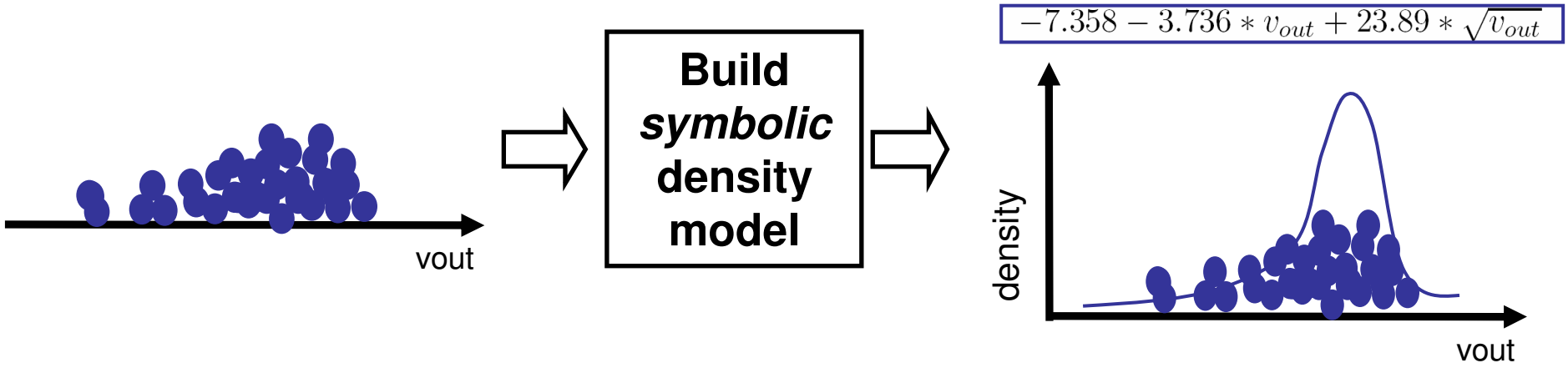


- Aims to maximize the likelihood of the input sample points
- Each point is treated equally (i.e. weights are the same, implicitly)

But…

- Not typically easy-to-analyze closed-form expressions
- How to handle importance-sampled data?

Symbolic Density Estimation



Aims:

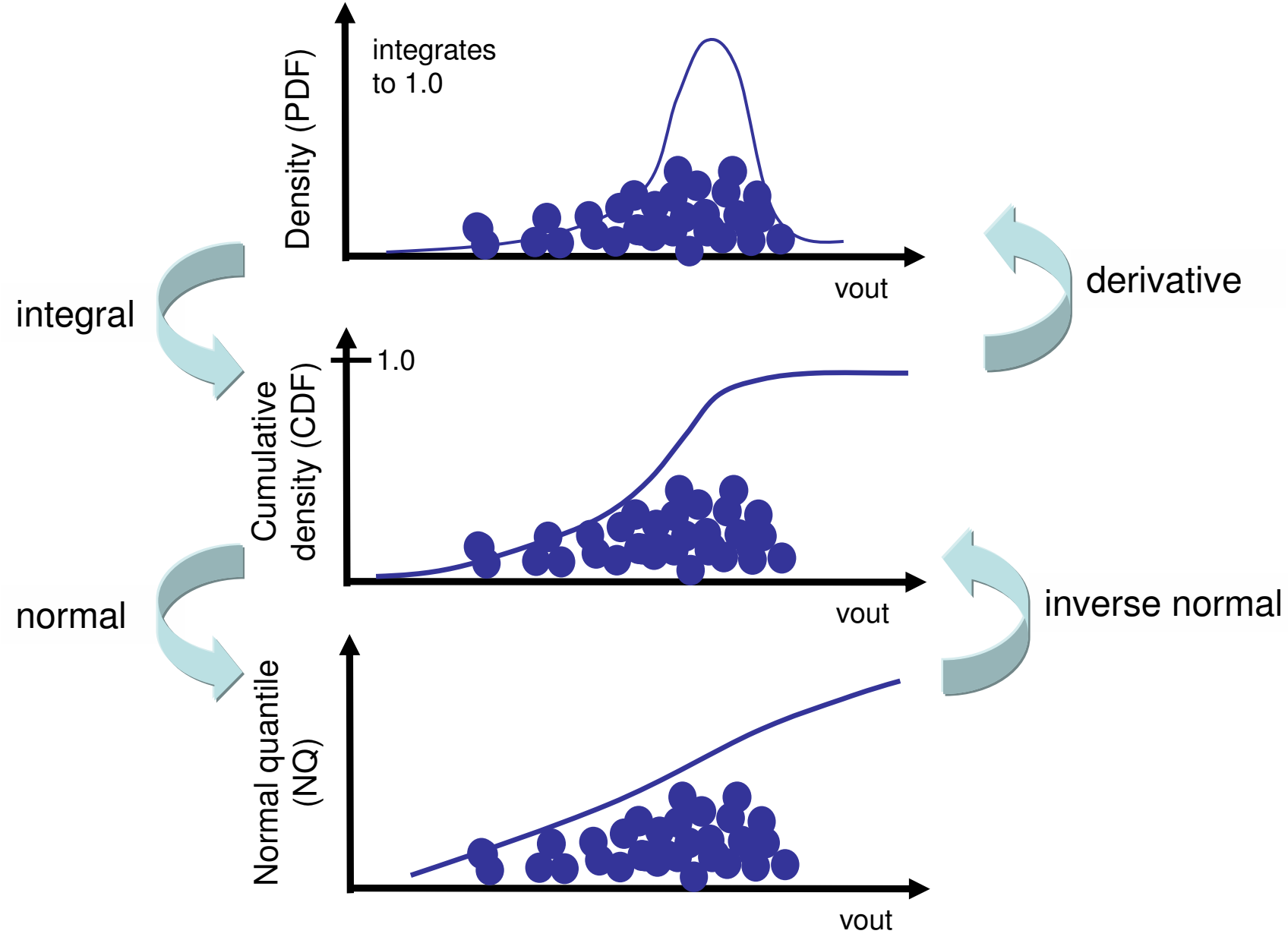
- Explicitly outputs easy-to-analyze closed-form expressions
- Handles importance-sampled data (i.e. weighted samples)

Key approach:

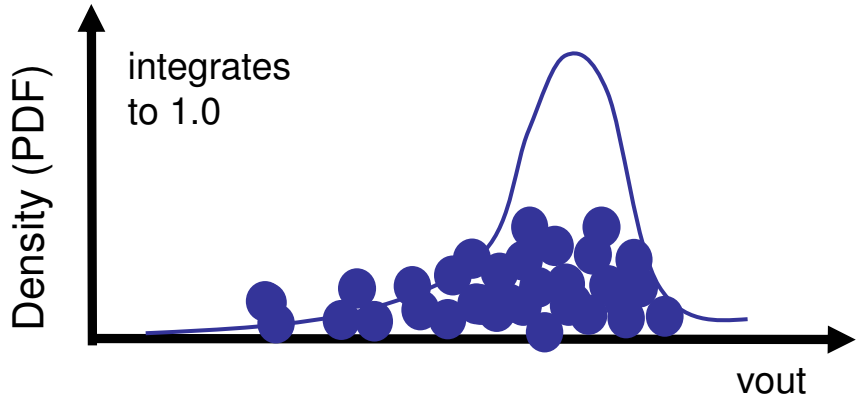
- Cast into a symbolic regression problem
- Apply genetic programming

Refresher: PDF, CDF, NQ

- Three different views of the same distribution



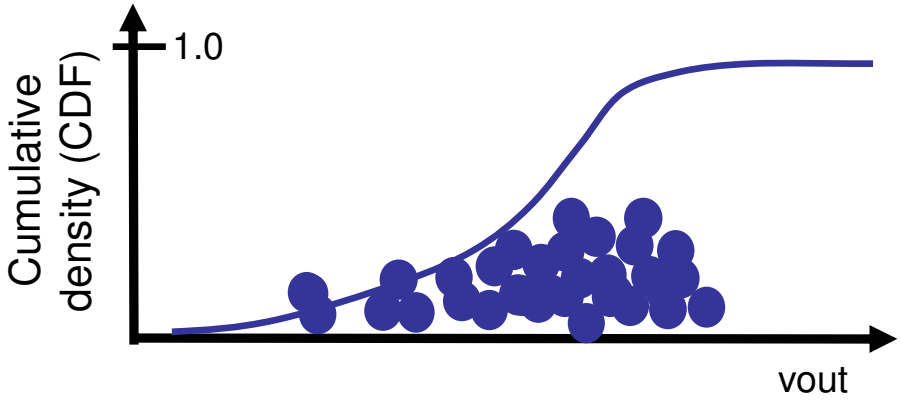
Model in PDF vs CDF vs NQ?



Constraint: integrate to exactly 1.0.

Strongly non-linear model

Constraint: model always ≥ 0.0

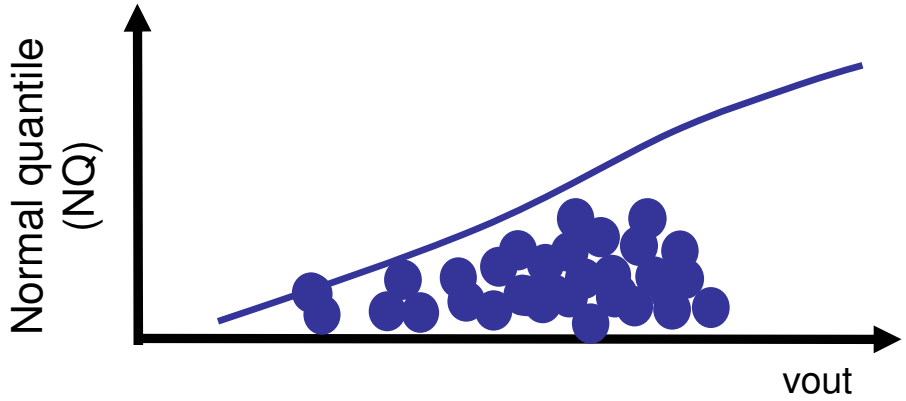


Constraint: model starts at 0.0, ends at 1.0. How to taper tails?

Weakly-nonlinear model

Constraint: model always ≥ 0.0

Constraint: monotonically increasing



Constraint: monotonically increasing

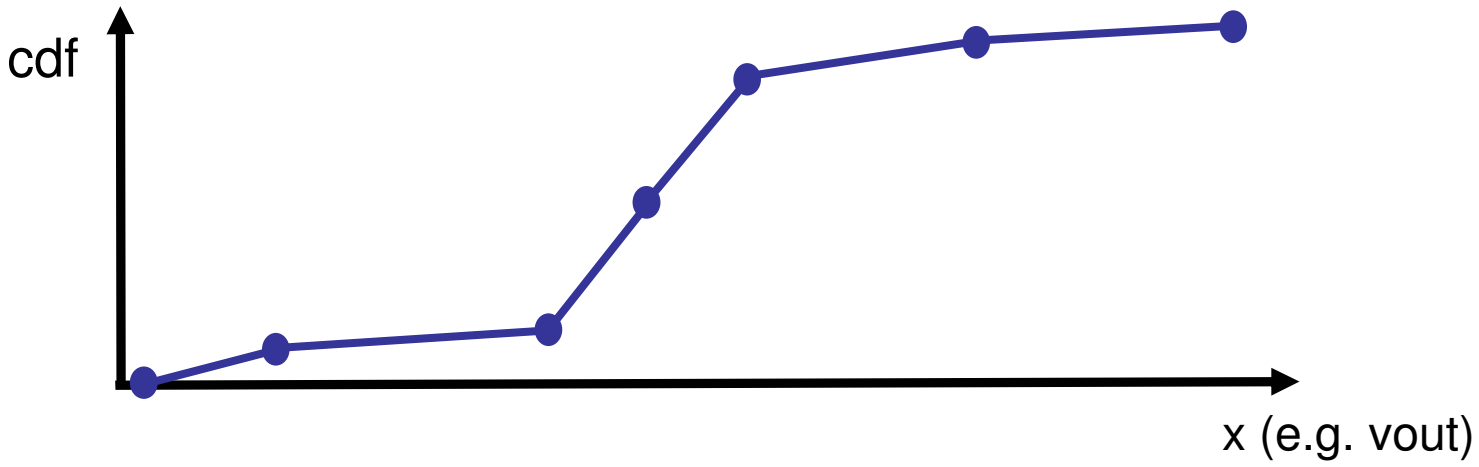
Nearly-linear model

Extrapolates nicely at tails

Symbolic Density Estimation

- Given importance sampled data: $\{x_i, w_i\}$
  - or plain MC data, just let w_i=1
- Assume w normalized

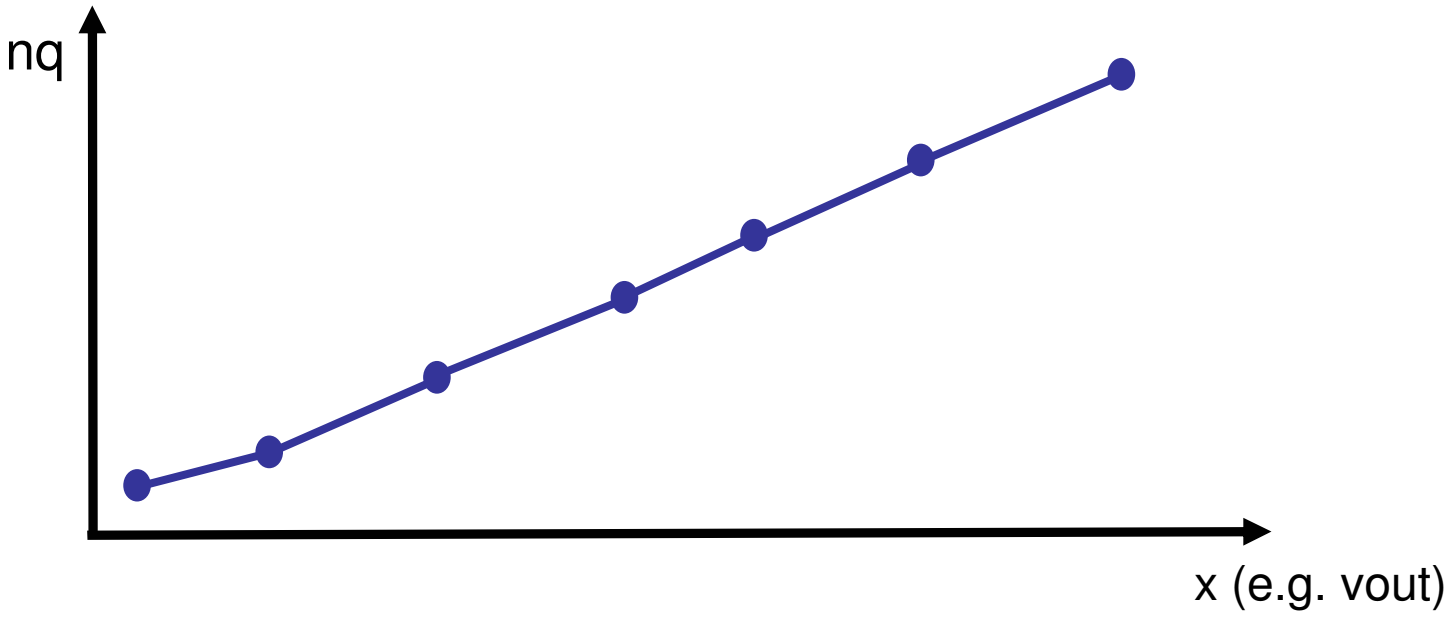
1. Sort x and w in order of ascending x
2. Compute numerical cdf: x→cdf
   - $cdf_i = \sum^{i}_{k=1} w_k$



Symbolic Density Estimation

Given IS or MC data (x_i, w_i)

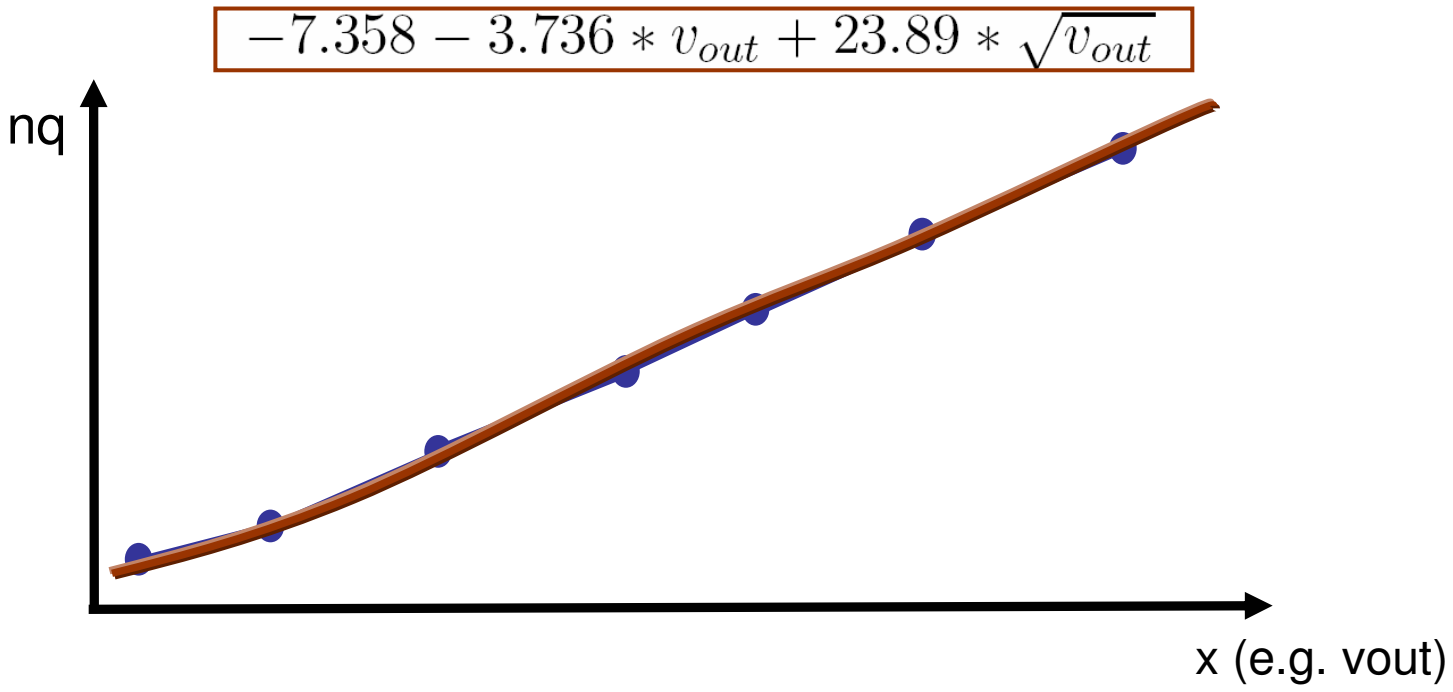
1. Sort in ascending x
2. Compute numerical cdf: x→cdf
3. Compute numerical normal-quantile: x→nq



Symbolic Density Estimation

- Given IS or MC data

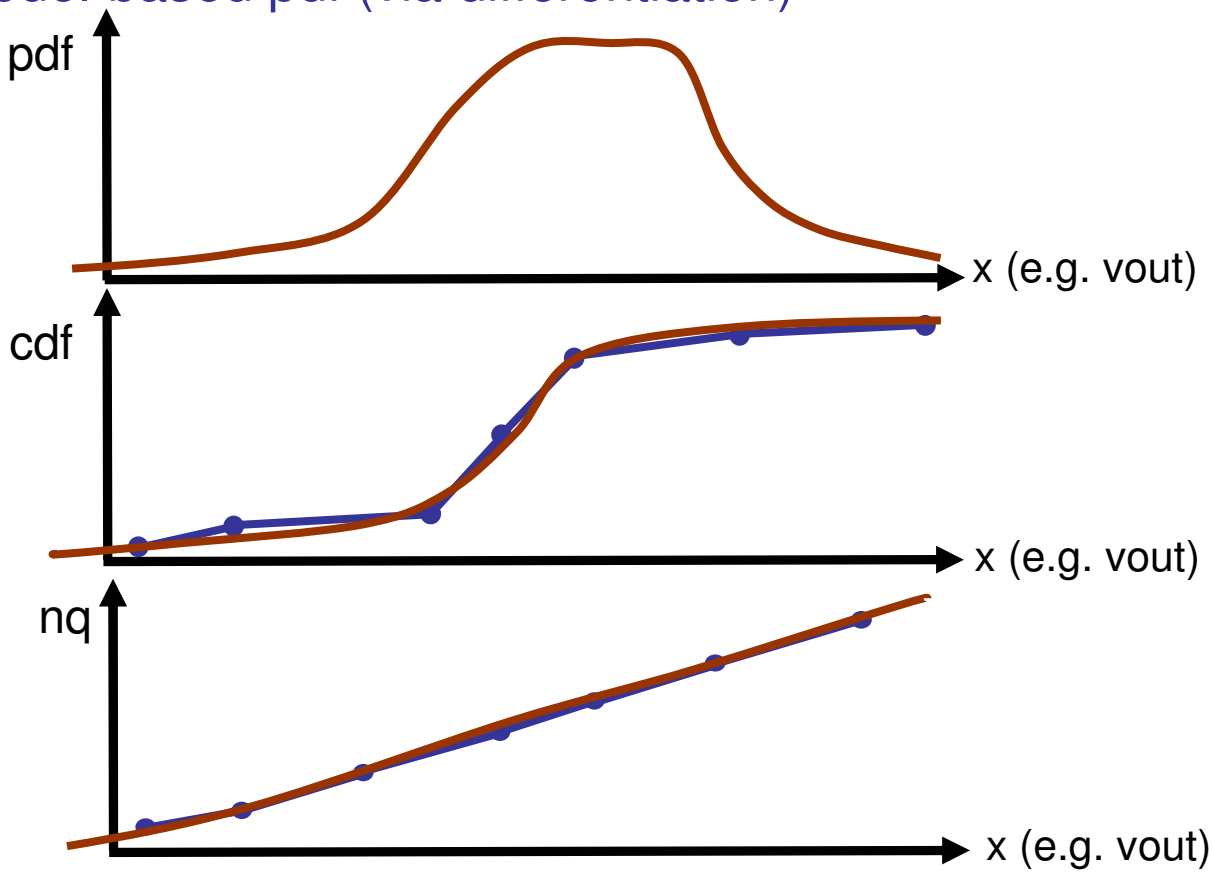
1. Sort in ascending x
2. Compute numerical cdf: x→cdf
3. Compute numerical normal-quantile: x→nq
4. With GP, find symbolic density model of x→nq



Symbolic Density Estimation

- Given IS or MC data

1. Sort in ascending x
2. Compute numerical cdf: x→cdf
3. Compute numerical normal-quantile: x→nq
4. With GP, find symbolic density model of x→nq
5. Compute model-based cdf (via inverse normal)
6. Compute model-based pdf (via differentiation)



Some GP Implementation Details

- Overfitting not a big issue because have 5000-50,000 samples and just 1d input. ("Trivial")
- Can use any GP symbolic regression system
- Can make it multi-objective, for a tradeoff between model complexity and model error

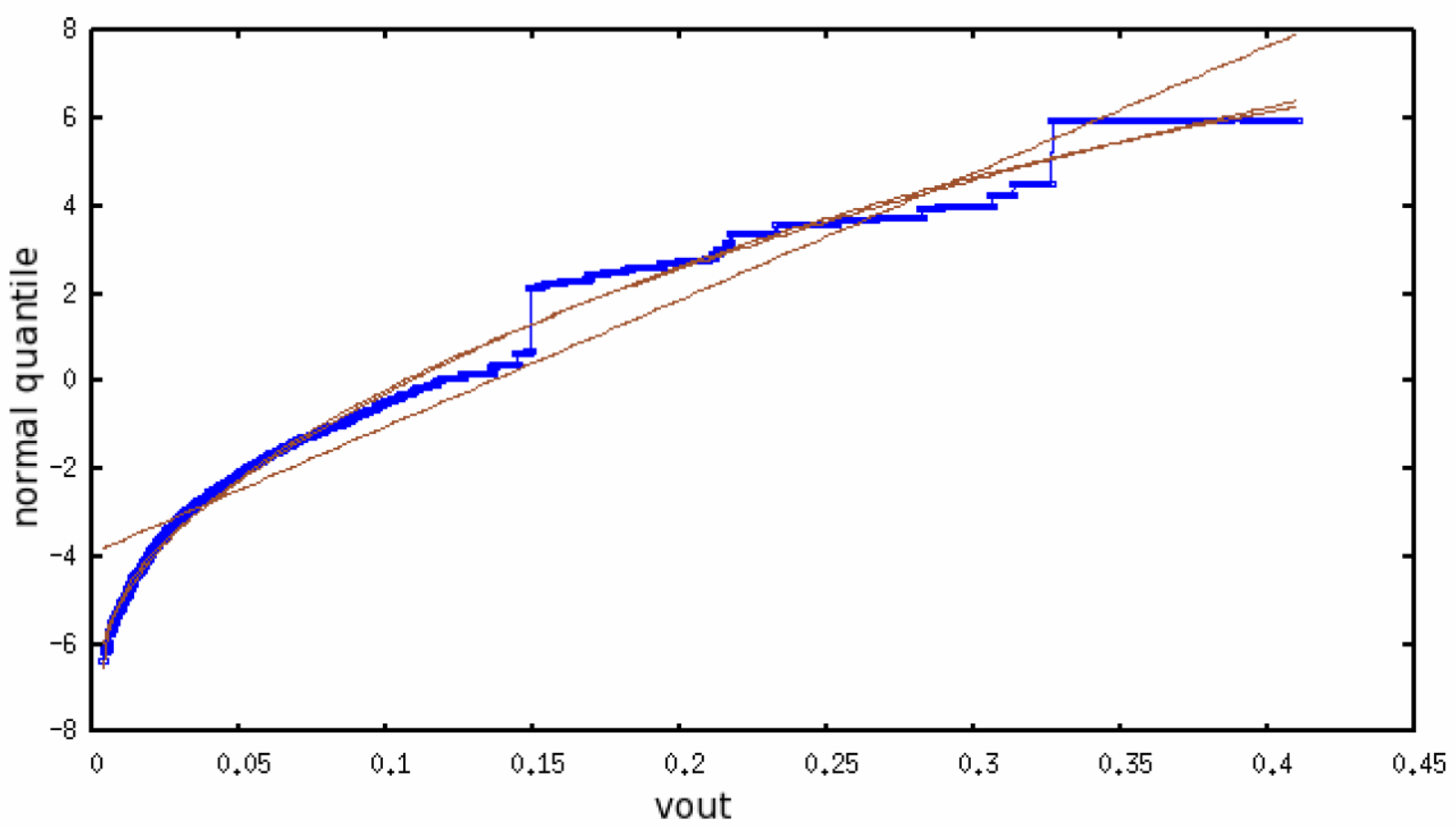
- I used CAFFEINE because:
  - It returns easy-to-interpret equations because search is constrained to canonical-form functions. Bloat not an issue.
  - It has built-in bias to working off linear models (optional)
  - It's multi-objective
  - It was convenient!

- For extra speed, I pre-pruned the data to 50 points:
  1. Took every n^{th} sample to get to 250 points
  2. Then applied SMITS balancing procedure to get to 50 points.
     - At each iteration, remove the sample that has lowest "deviation from linearity". Calculate deviation via local linear fits.

- I actually post-pruned the Pareto Front using SMITS too

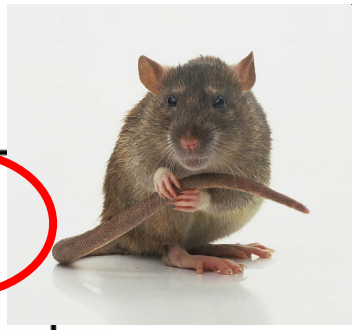
Results

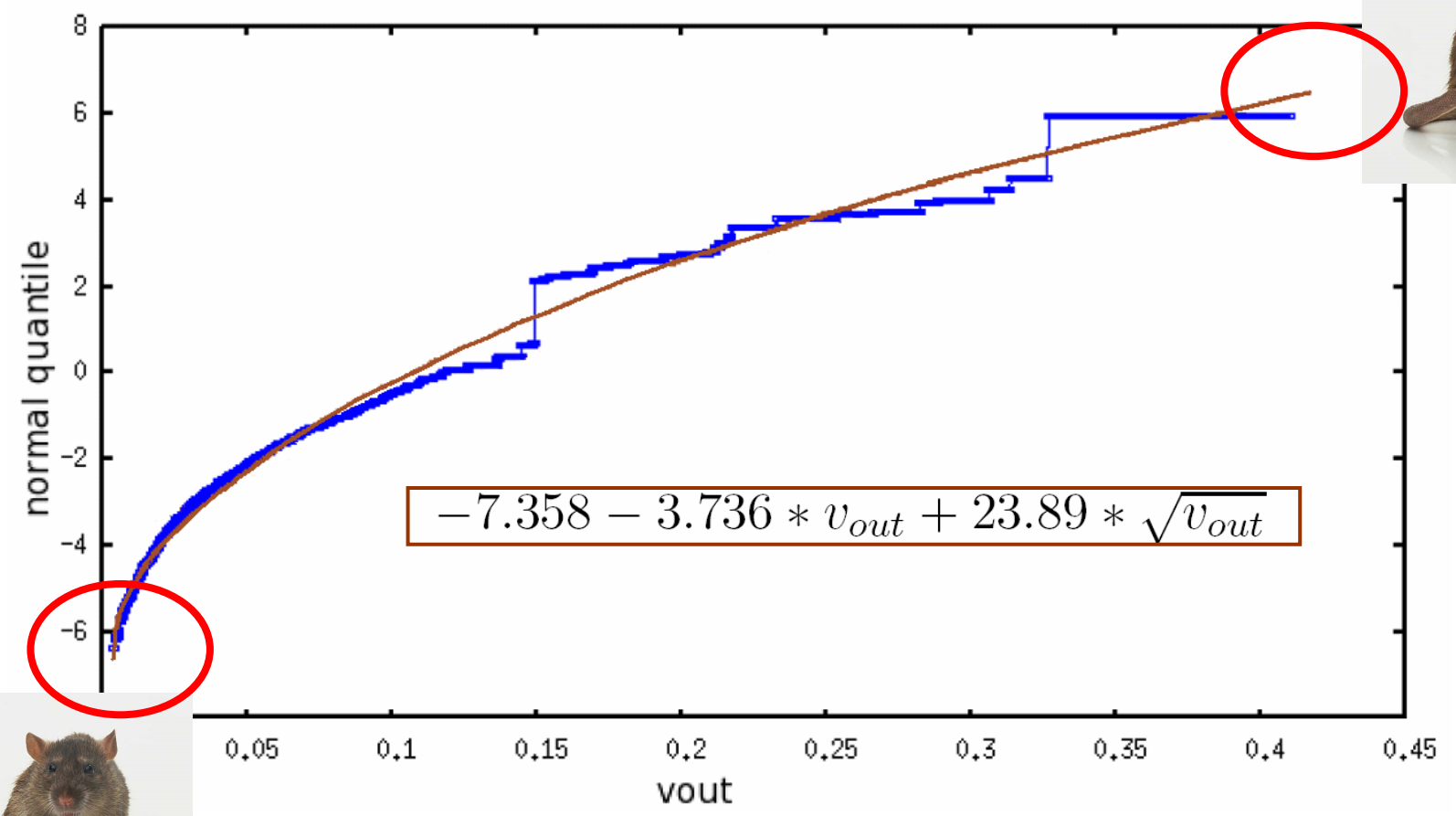
Results on bitcell: nq



Train error (%)	***nq* Expression**
26.3	$-3.903 + 28.90 * v_{out}$
11.42	$-7.358 - 3.736 * v_{out} + 23.89 * \sqrt{v_{out}}$
11.01	$368.1 - 46.482 * v_{out} + 170.959 * \sqrt{v_{out}}$ $+41.0435 * log10(1.07285e - 7 * \sqrt{v_{out}})$ $-0.002472/\sqrt{max(0, 1.562e - 10/\sqrt{v_{out}})}$

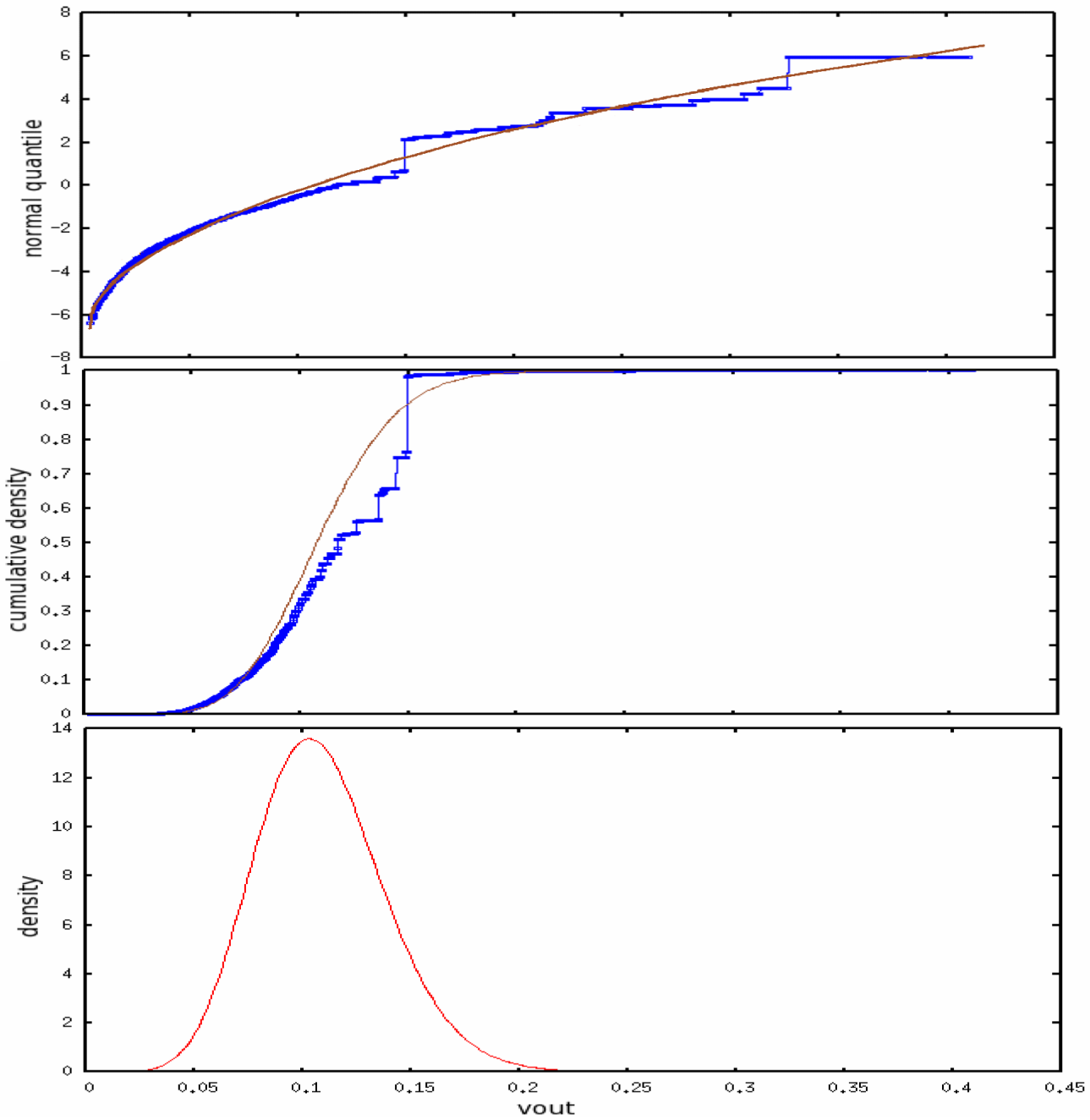
Note the tails – extrapolation looks good





This is a valuable result for memory circuit designers.

Results: nq, cdf, pdf



Conclusion

Take-Away Lessons for GPers

1. Symbolic Density Modeling

- We can apply SR to density modeling: symbolic density modeling
  - Modeling in nq space greatly simplifies the problem
  - Importance sampling enables models of one-in-a-billion tails
  - Demonstrated this on a real-world problem (memory design)
  - Many other real-world problems in density modeling…

2. On GP Real-World Applications

- One recipe for successful industrial application is: give GP problems that are trivial for GP!
  - 1-D SR and 50 samples is trivial. This is a good thing because…
  - …GP can then quickly & reliably get high-quality results!!
  - Just because problems are trivial for GP doesn't mean the overall value is trivial – my memory design example means $$.

3. Tails

- Are funny☺

That's All Folks!