Recursive Statistical Blockade
An Enhanced Technique for Rare Event Simulation with
Application to SRAM Circuit Design

Amith Singhee[1], Jiajing Wang[2], Benton H. Calhoun[2], Rob A. Rutenbar[1]

[1]Dept. of ECE, Carnegie Mellon University

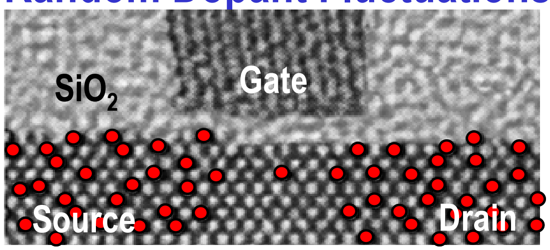
[2]ECE dept., University of Virginia

Jan 6, 2008

New Problem with Nanoscale Circuits...

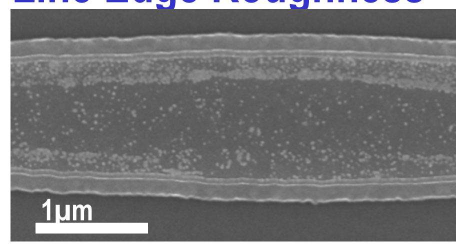
- Process-induced *random variations* in devices

Random Dopant Fluctuations



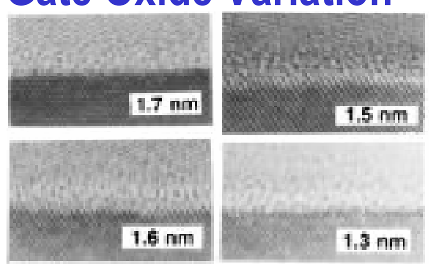
A. Brown *et al.*, *IEEE Trans. Nanotechnology*, p. 195, 2002

Line Edge Roughness



K. Shepard, U. Columbia

Gate Oxide Variation

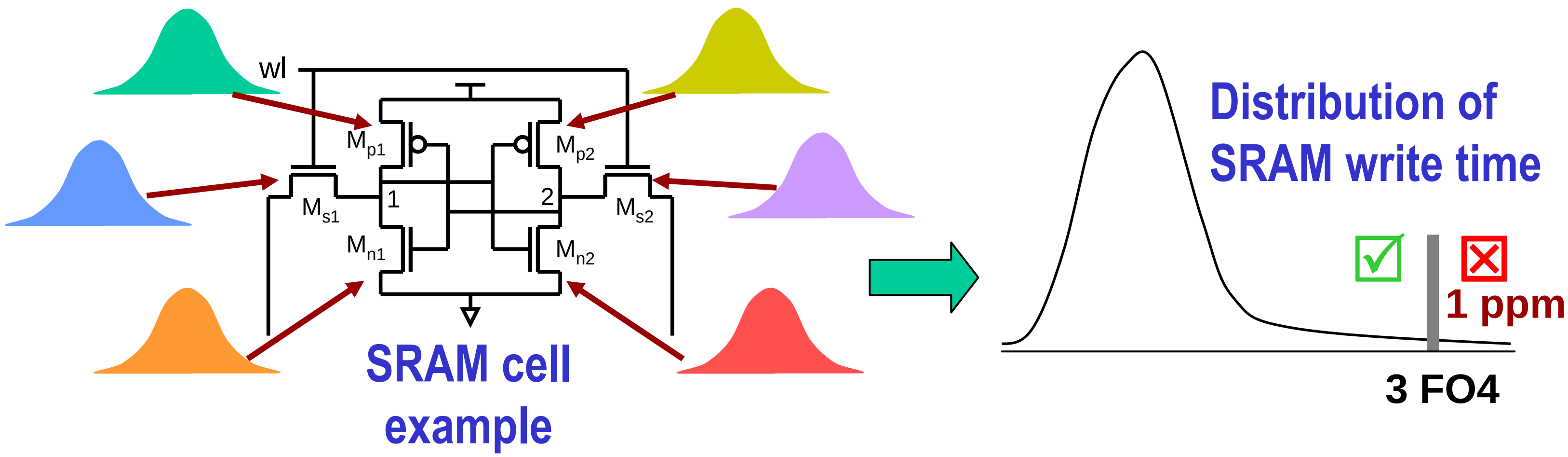


Momose et al, *IEEE Trans. Electron Devices*, 45(3), 1998

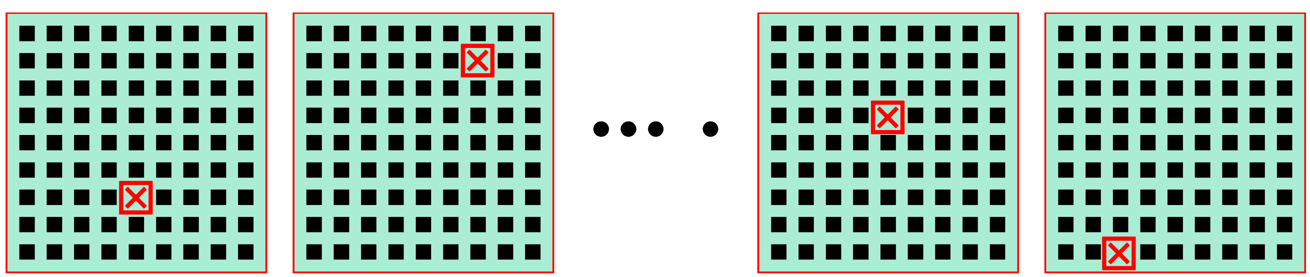
- Concerns:
  - **Independent**, possibly **large** variations **per device**
  - **High-replication** circuits (e.g. SRAMs, flip flops) are extremely **vulnerable**

High-Replication Cells

- High replication = Millions of "identical" cells per chip



- Very good, high yield design …right?
- 1 Mb cache = 1M cells in one chip = *every* chip fails (on average)!



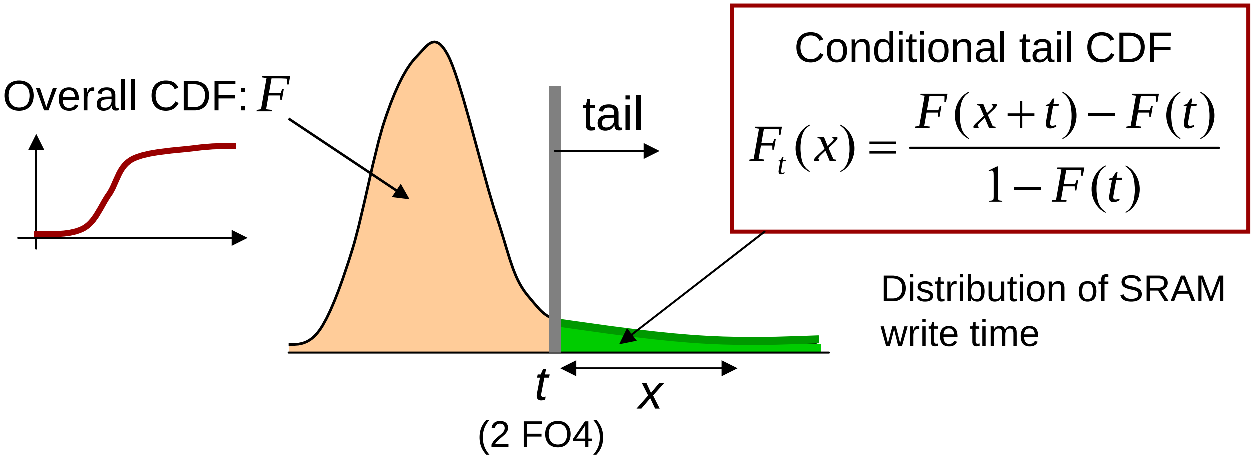
High-Replication Cells: Rare Events

- **How to *estimate* the statistics of such rare events?**
  - Standard Monte Carlo approach is too **s l o w**
  - Maybe **millions of samples** to model these rare, critical events

- **Statistical Blockade [Singhee et al, DATE 2007]:**
  1. Focus the MC to **simulate only (mainly) the critical, rare events**
  2. **Maintain the statistics** of the simulated rare events
  3. **Model the statistics** of the rare events

Modeling the Statistics

- We want to fit only the tail here
  - Example: **Prob[SRAM write-time > 2 FO4 delay]** – the *bad* values

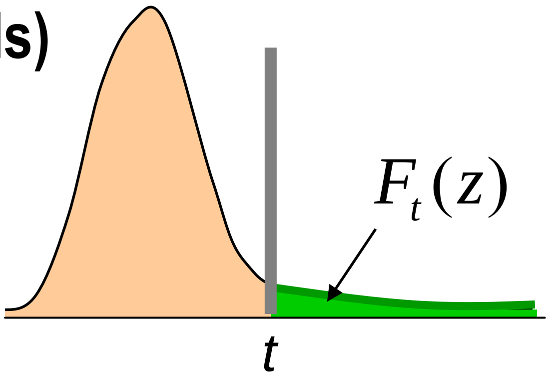


We want: **A simple analytical form for the *conditional* distribution ($F_t(x)$) of the events in the tail**

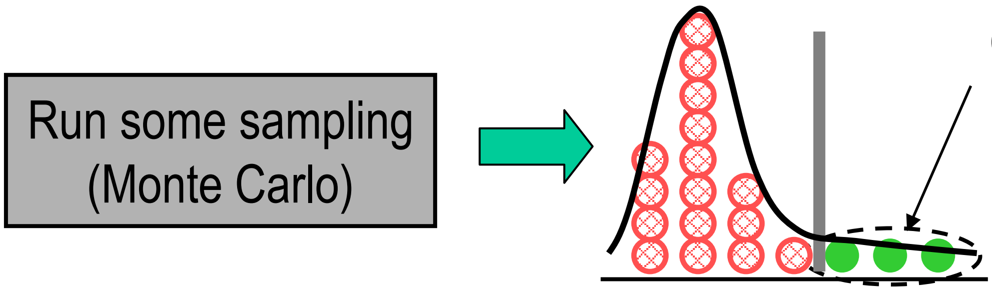
Solution: Extreme Value Theory

- **Recall Central Limit Theorem:** **Σ(i.i.d. samples) → Gaussian**
  - Question: Is there a similar result for these tails of "extreme" results ...?
  - Answer: YES – **Extreme Value Theory** (EVT)

- **Theorem (Balkema / deHaan, and Pickands)**
  - As $t \to \infty$, $F_t(z) \to$ **Generalized Pareto Distrib**
  - Under certain conditions: shown to be satisfied for large classes of distributions



Using EVT: We Fit a GPD to Tail



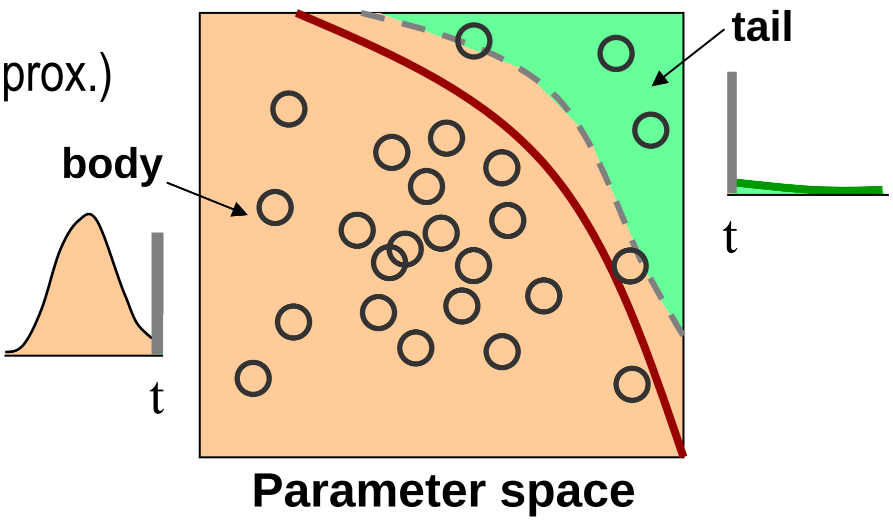
Can fit a **GPD** to these tail points

$$G_{a,k}(z) = \begin{cases} 1 - \left(1 + \dfrac{kz}{a}\right)^{1/k} & , k \neq 0 \\ 1 - e^{-z/a} & , k = 0 \end{cases}$$

- **To fit a GPD means to estimate parameters a and k**
- **Several methods**
  - Moment matching (MM) $\longrightarrow M_p = E[z^p]$
  - Maximum Likelihood Estimation $\longrightarrow \max_{a,k} P(z \mid a, k)$
    - Convergence issues due to optimization formulation
  - **Probability Weighted Moment Matching** $\rightarrow M_{p,r,s} = E[z^p F(z)^r (1 - F(z))^s]$
    - Studies show lower bias than MM
    - No convergence issues

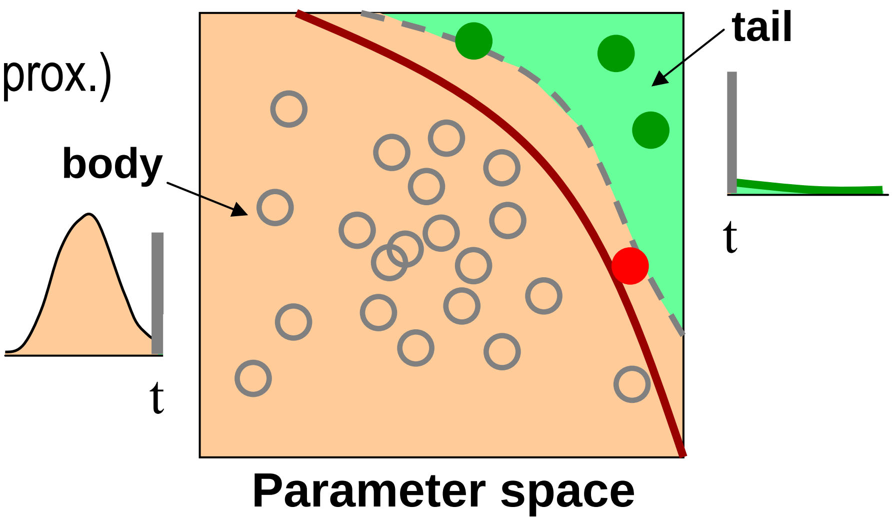
Next Problem: Efficiently Sampling *Just* the Tail

- **MC is too slow: How do we actually generate the tail samples?**
  - How to do this **fast**, following the **true statistics**, to enable fitting the GPD
- **Observe: Generating MC samples is cheap, *simulating* is costly**
- **Idea:**
  1. **Learn** tail region boundary (approx.)
  2. **Generate** MC points

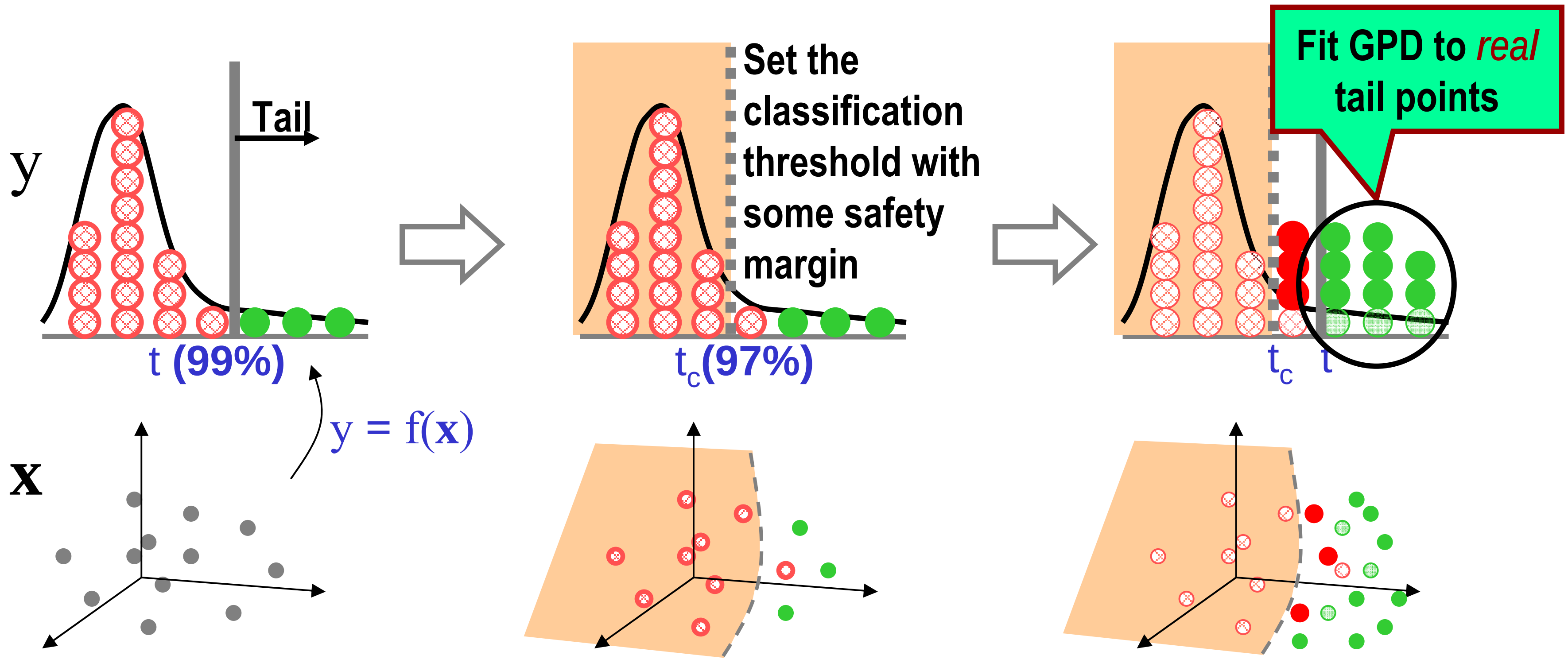


Next Problem: Efficiently Sampling *Just* the Tail

- **MC is too slow: How do we actually generate the tail samples?**
  - How to do this **fast**, following the **true statistics**, to enable fitting the GPD
- **Observe: Generating MC samples is cheap, *simulating* is costly**
- **Idea:**
  1. **Learn** tail region boundary (approx.)
  2. **Generate** MC points
  3. **Block** "body" points (most)
  4. **Simulate** "tail" points (few)



We Call the Idea: *Statistical Blockade* [DATE '07]



Simulate starting set (few points, *fast*)

Build classifier (*fast*)
(data structure to label points as body/tail)

Generate MC samples (*fast*)
Classify sample points (*fast*)
Block nontail points (*fast*)
Simulate the rest (*slow*)

Problem #1: Conditionals in Performance Metrics

- **Example: Data Retention Voltage – minimum supply voltage with no loss of data**

- **Computed using a conditional, max( )**

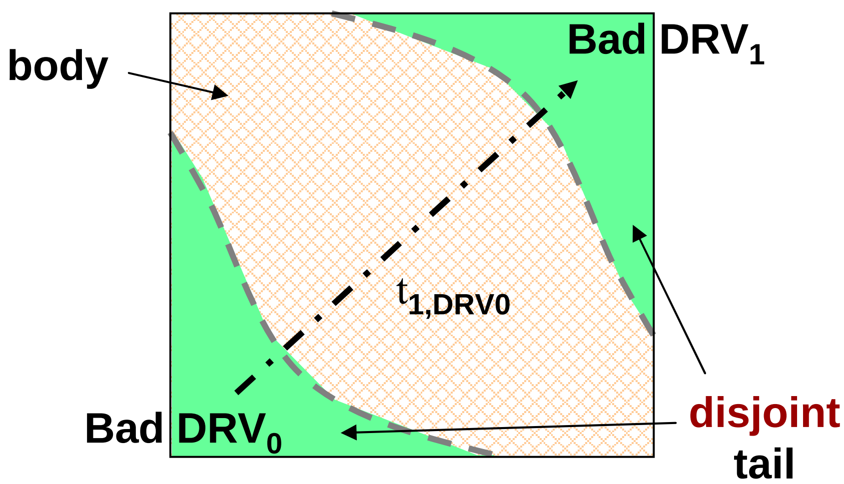
$$DRV = \max(DRV_0, DRV_1)$$

  - DRV_0 – when the cell is storing a 0
  - DRV_1 – when the cell is storing a 1

- **In general, DRV_0 is small when DRV_1 is high, and vice versa**
  - Hard to store 0 → easy to store 1, vice versa

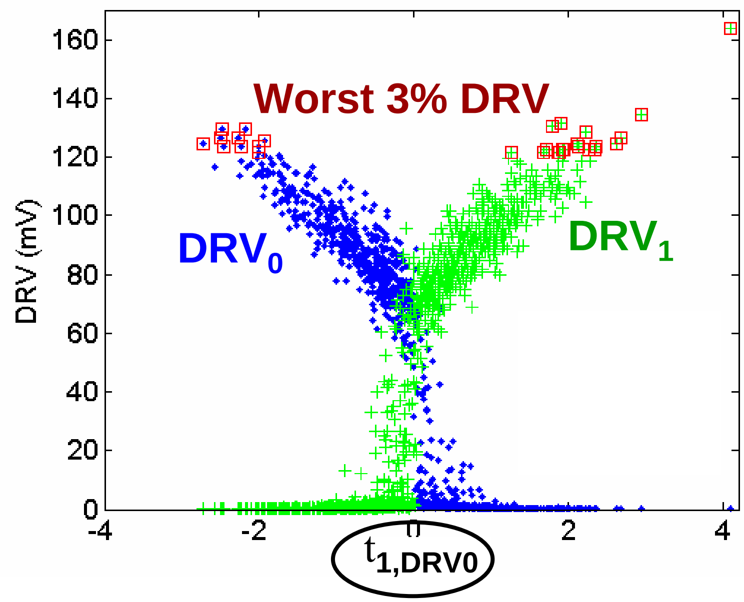
Conditionals in Parameter Space: Disjoint Tail Regions

- **Hence, we get a disjoint tail**
  - A single boundary cannot separate tail and body regions



Statistical parameter space

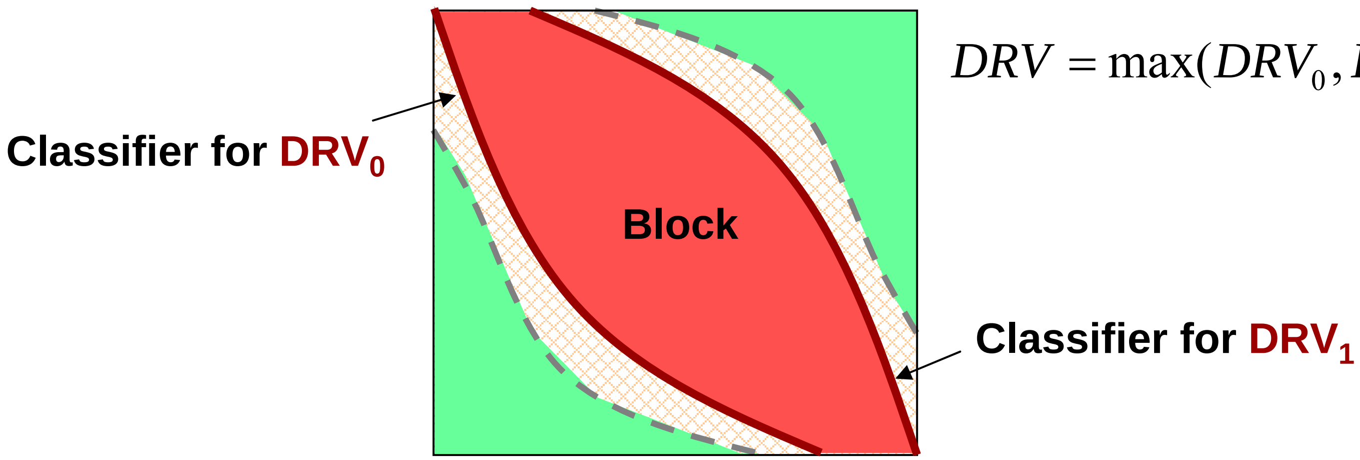
$t_{1,DRV0}$ **: direction of maximum variation in DRV$_0$**



Using SiLVR [Singhee, Rutenbar DAC07] (data from Monte Carlo)

Solution: Multiple Classifiers

- **Build multiple classifiers – one for each arg. of the conditional**
- **Use the same training points**
  - No extra simulations for training, only extra classifier training

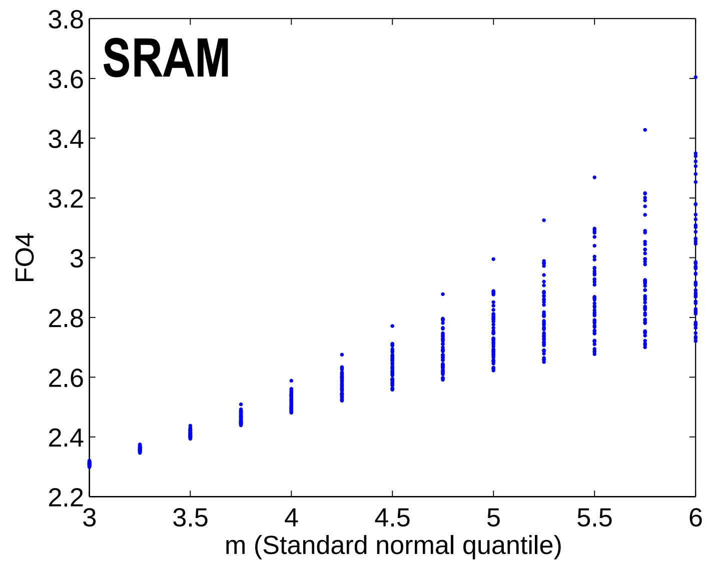


$$DRV = \max(DRV_0, DRV_1)$$

- **Block using both classifier– simulate if not blocked by *both***
  - At worst, two times the simulation cost – still much cheaper than full MC

Problem #2: Confidence in Tail Model

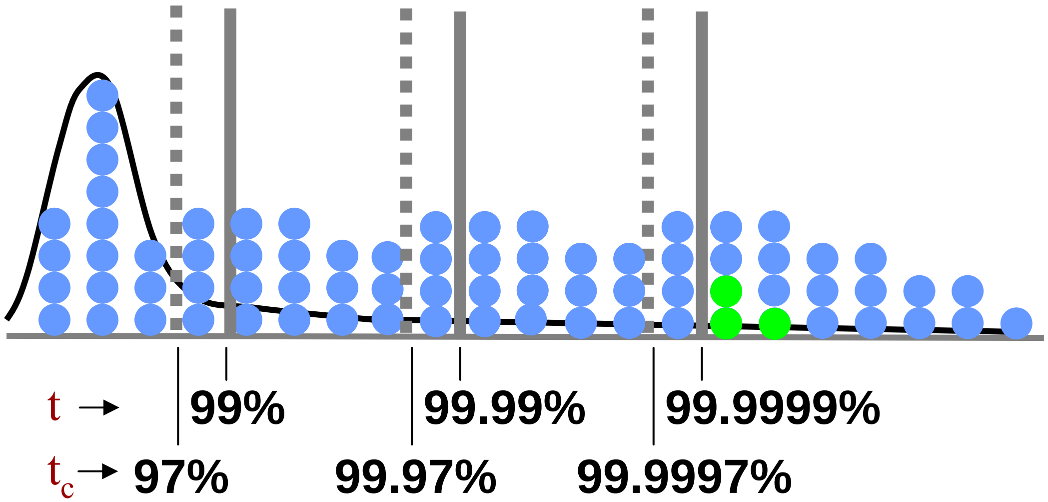
- Confidence decreases as we extrapolate far beyond data
- 50 MC runs of 100K points – 50 GPD models of the tail



- For 6σ, want tail model at t = 99.9999%
- How do we build a classifier at t_c = 99.9997%: need training points

Solution: Recursive Statistical Blockade

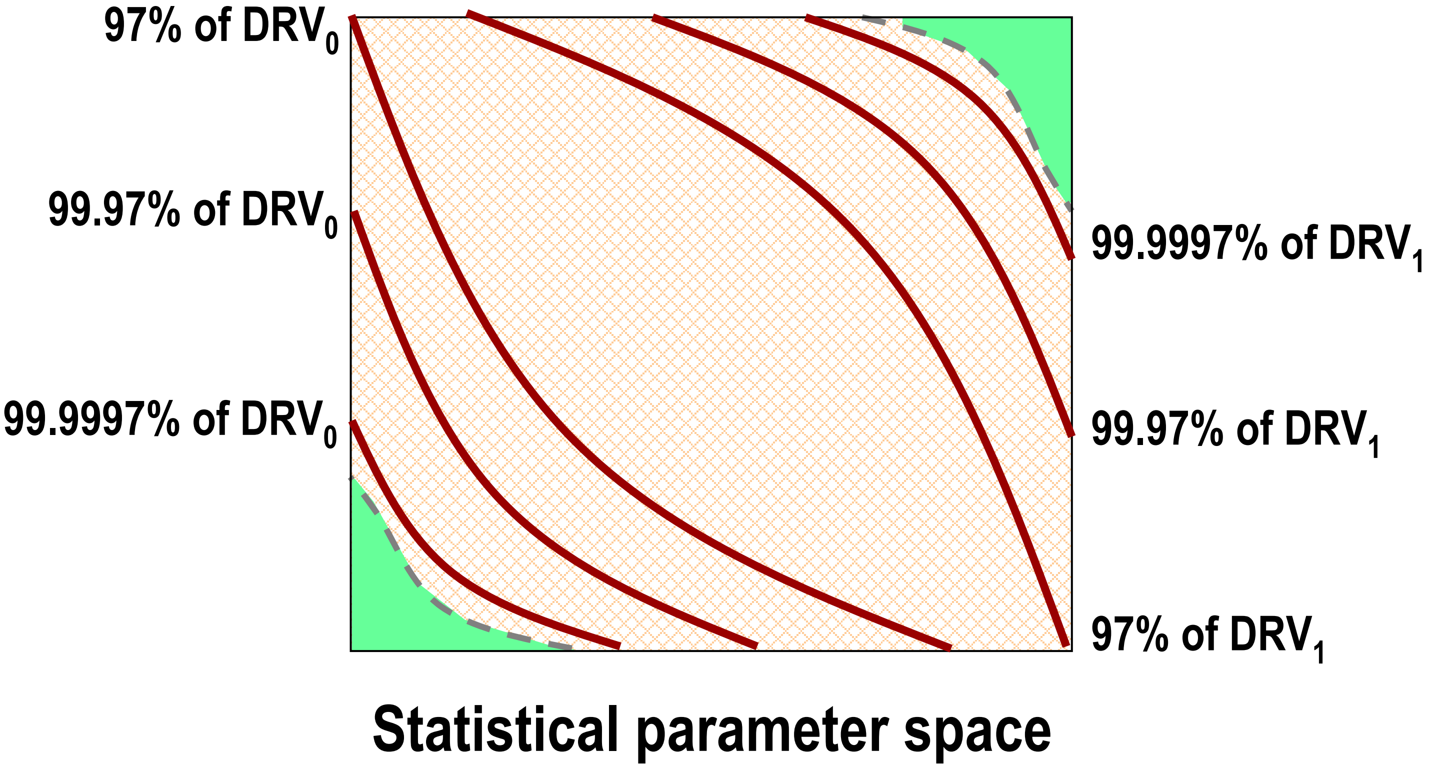
- Classifier at 99.9997% needs training points beyond 99.9997%!
- To generate points beyond 99.9997%, use SB at 99.99%



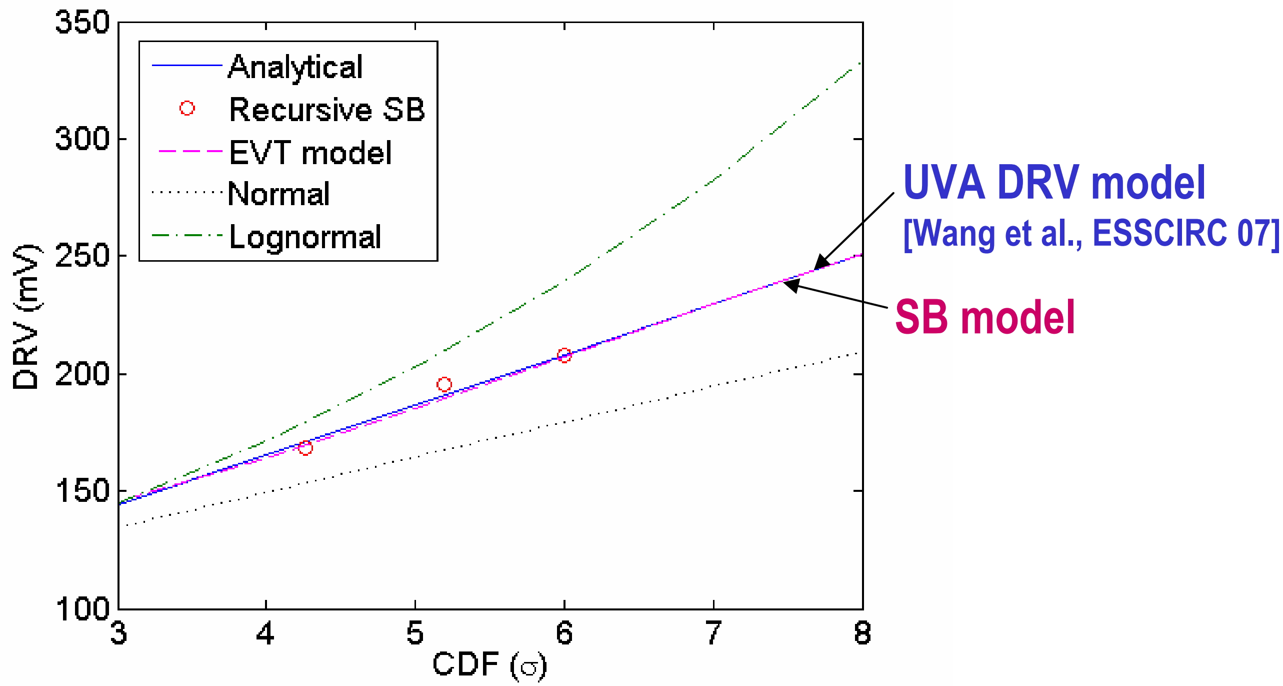
- Fit GPD to final tail points (shown in green)

- Extending this to disjoint tails due to conditionals …

Recursive SB for Disjoint Tail Regions



Comparison with Known Analytical Model



- Comparison out to 8σ shows good match
- Recursive SB required 3 recursion stages – 3 tail thresholds
  - Filtering 100,000, 10 million and 1 billion samples
  - Worst DRV from each stage estimates the 4.26σ, 5.2σ and 6σ DRVs, resp.

Speed Comparison

- Simulation costs for simulating upto the 6σ DRV

Recursive SB	
Recursion Stage	Num. Simulations
Init	1,000
1	11,032
2	14,184
3	15,505
Total	41,721

	Num. simulations	Speedup by Recursive SB
Monte Carlo	1 billion	23,969x
Statistical Blockade	30 million	719x

Conclusions

- **Recursive Statistical Blockade**
  - Extended and “practicalized” SB
  - Handle disjoint tail regions due to conditionals
  - Sample extremely rare events efficiently

- **Flexible approach: almost any distribution with a long tail**
  - Different circuits
  - Different performance metrics

- **Tested on DRV of SRAM cell in 90nm industrial process – speedups of 4 orders of magnitude over Monte Carlo**

Recursive Statistical Blockade
An Enhanced Technique for Rare Event Simulation with Application to SRAM Circuit Design

Amith Singhee[1], Jiajing Wang[2], Benton H. Calhoun[2], Rob A. Rutenbar[1]

[1]Dept. of ECE, Carnegie Mellon University

[2]ECE dept., University of Virginia

Jan 6, 2008

Carnegie Mellon