ASPLOS '21

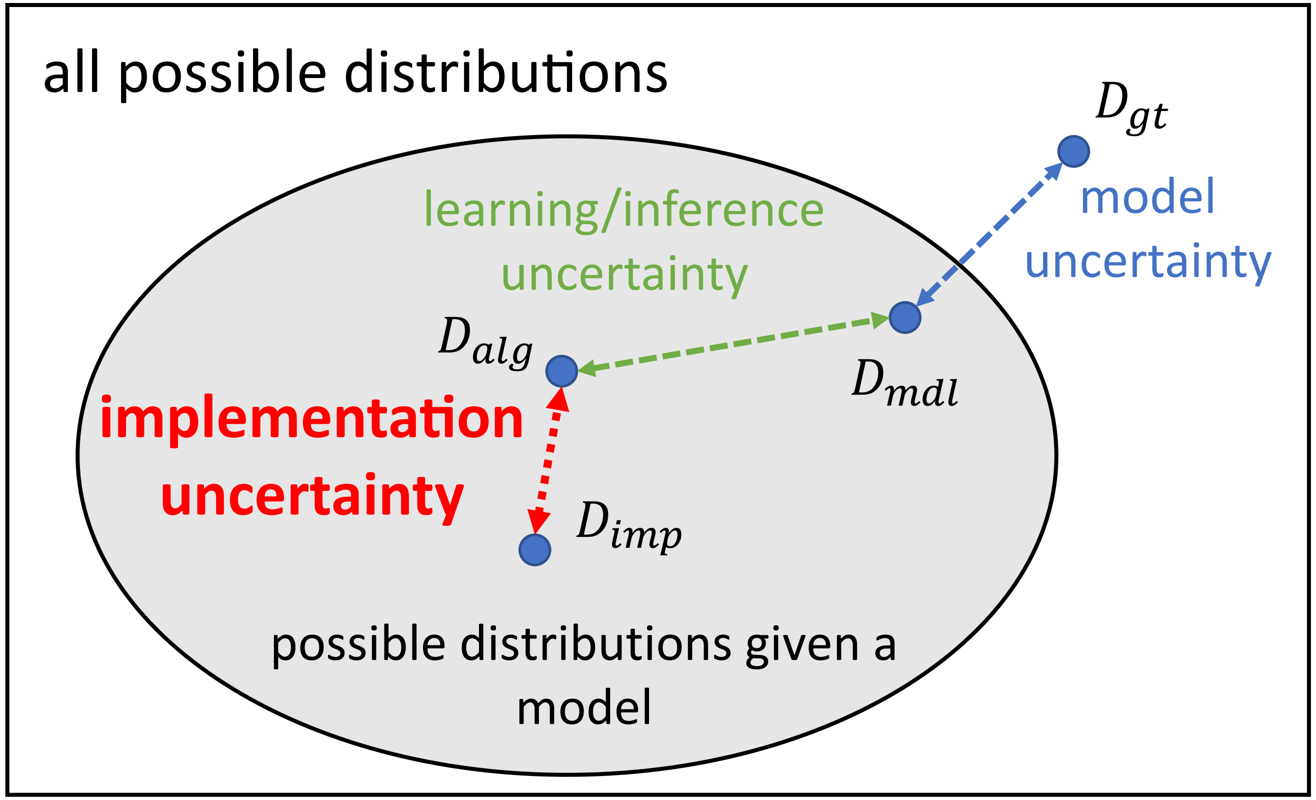
STATISTICAL ROBUSTNESS OF MARKOV CHAIN MONTE CARLO ACCELERATORS

Xiangyu (Mike) Zhang, Ramin Bashizade, Yicheng Wang,
Sayan Mukherjee, Alvin R. Lebeck

Duke University

Supported by intel Semiconductor Research Corporation NSF

Sources of Uncertainty



[Hullermeier and Waegeman 2019]

D_{gt} = Ground truth

D_{mdl} = Optimal from model

D_{alg} = Learned/inferred

D_{imp} = Implementation

Difficult to directly quantify implementation uncertainty…

Solution: Statistical Robustness

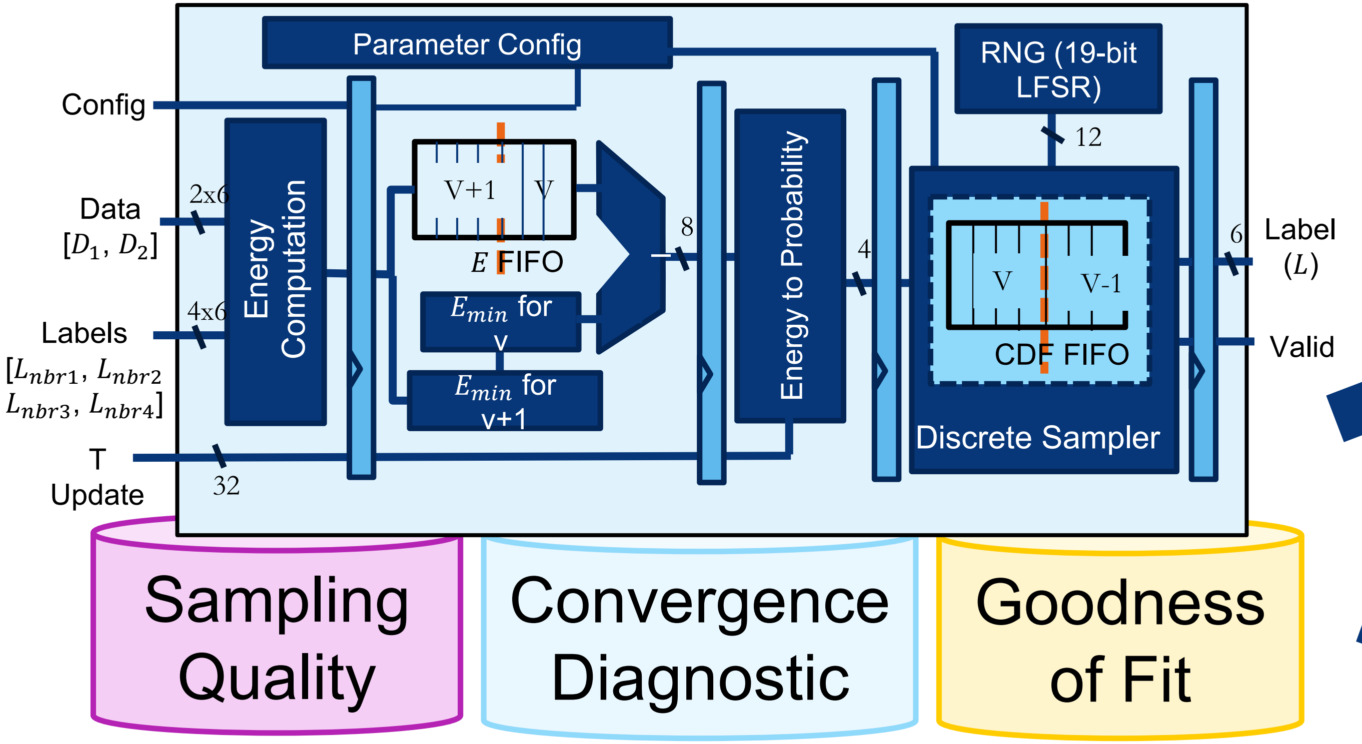
Claim: "*A probabilistic architecture should provide some measure (or guarantee) of statistical robustness.*"

"Accuracy" Evaluation **+** Sampling Quality: ESS; Convergence Diagnostic: $\hat{R}$, CP; Goodness of Fit: RMSE, JSD **=** **Comprehensive Correctness**

- Modified methods for:
  1) high dimensionality
  2) zero empirical variance
- No need to access ground truth
  - comparing with target quality (e.g. FP64)

Using the Three Pillars

[Zhang et al., ISCA 2018]



<u>Inform User</u>

Characterize Existing Hardware

<u>Inform HW Designer</u>

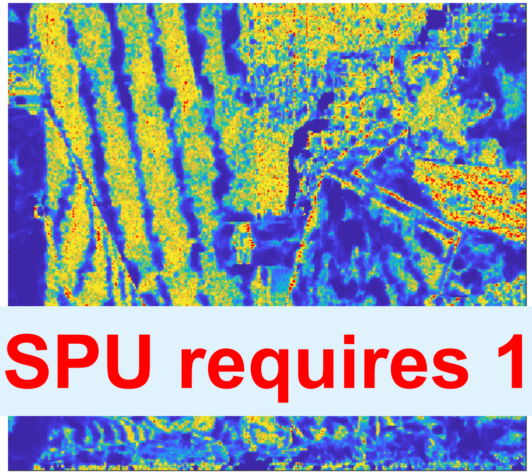
Design space exploration

<u>Methodology</u>

2 applications: Stereo Vision, Motion Estimation

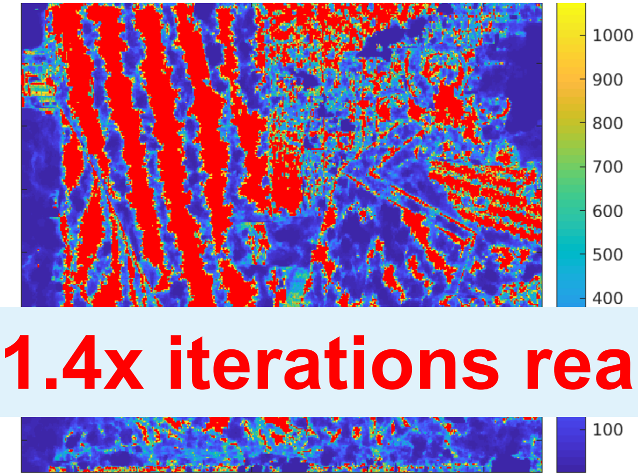
2 Modes: Sampling, optimization

Pillar 1: Sampling Quality (Effective Sample Size)

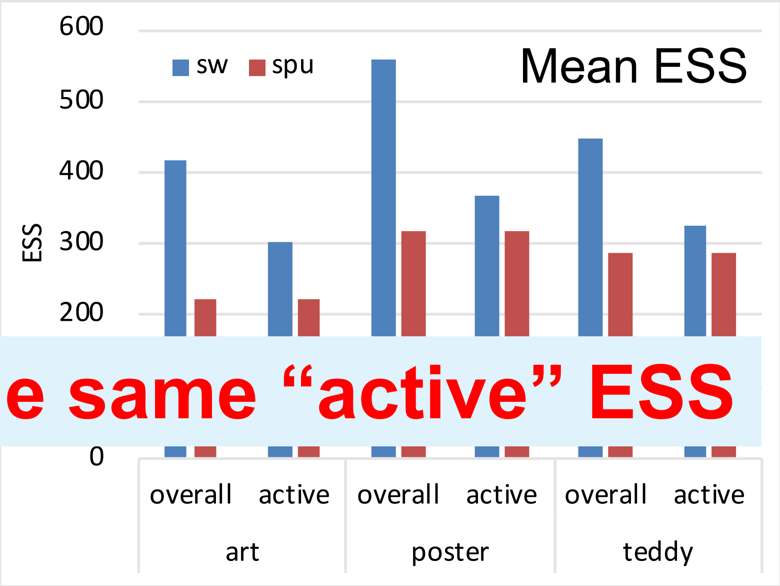


Software FP64

ESS per pixel:
Stereo Vision (*teddy*)



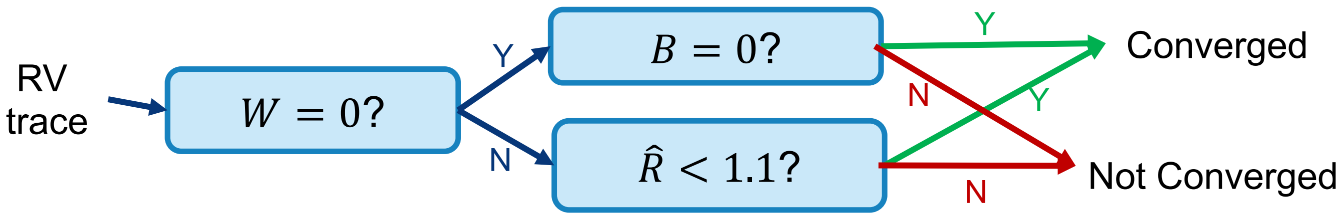
SPU



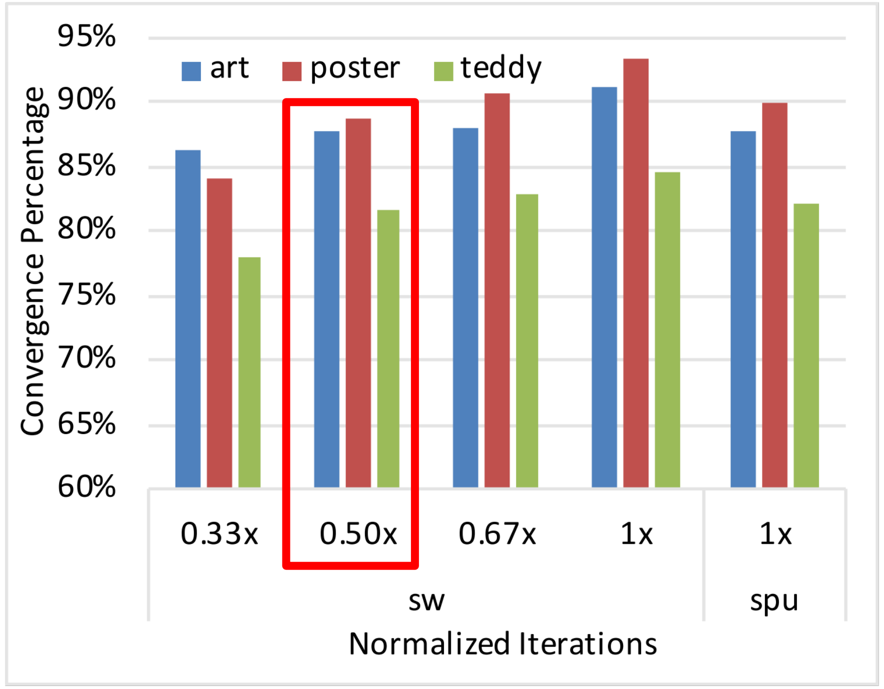
SPU requires 1.1-1.4x iterations reach the same "active" ESS

- Architectural optimizations: possibly reduce the independent samples
- ESS: number of independent samples drawn (lower value -> more iterations)
- Red pixels have var=0: no defined ESS -> Can't directly apply existing metrics
  - "Overall" ESS: omits var=0 pixels in software and the SPU, respectively, bias to software (yellow high ESS)
  - **"Active" ESS: pixels with meaningful ESS (var> 0 in both software and SPU)**

Pillar 2: Convergence Diagnostic

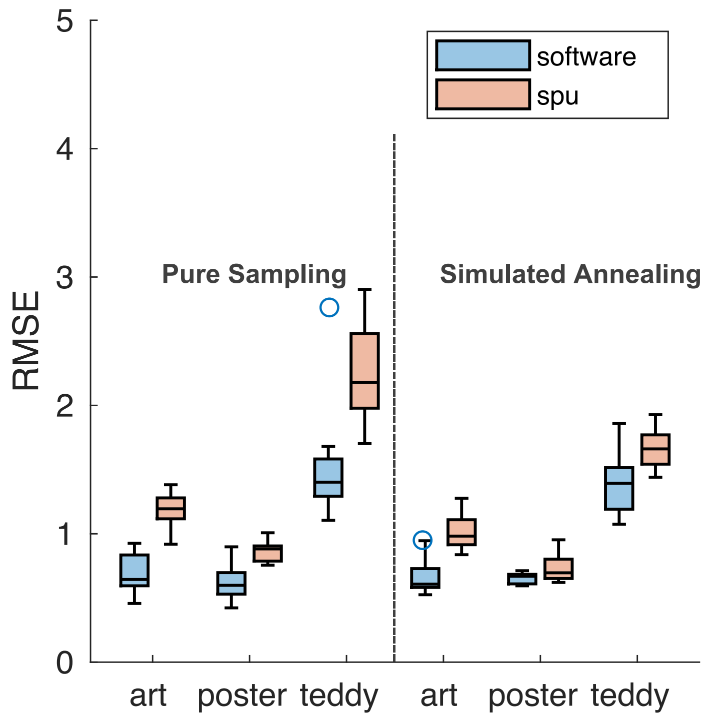


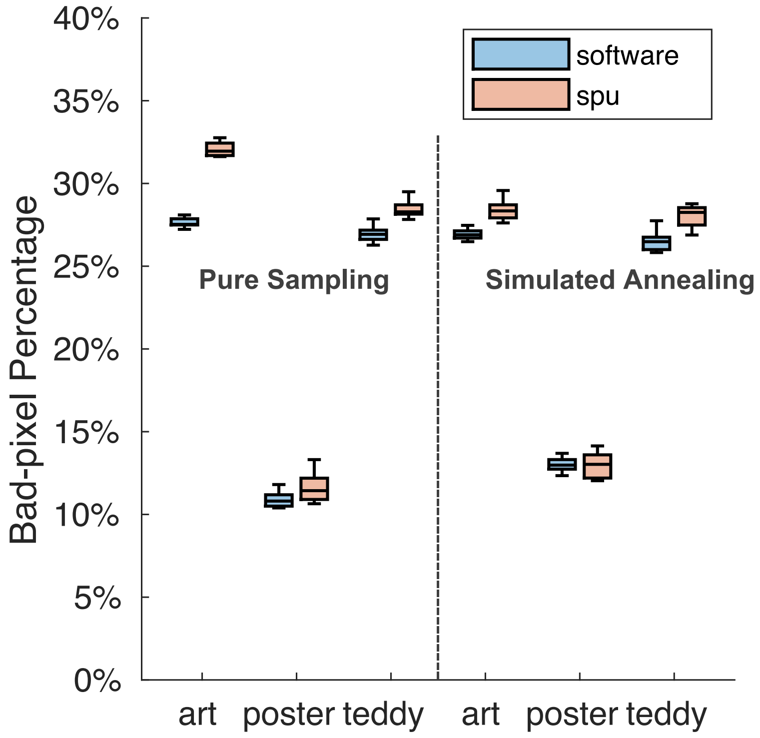
Our process to determine convergence



- How many iterations to converge?
  - Gelman-Rubin's $\hat{\mathrm{R}}$: variance Within (W) vs. Between (B) MCMC runs
  - Rule of thumb: < 1.1 is good (converged).
  - But, W=0 no definition -> Can't directly apply existing metrics
- **New metric, Convergence Percentage based on Gelman-Rubin's $\hat{\mathrm{R}}$**
- **SPU design needs 2x iterations**

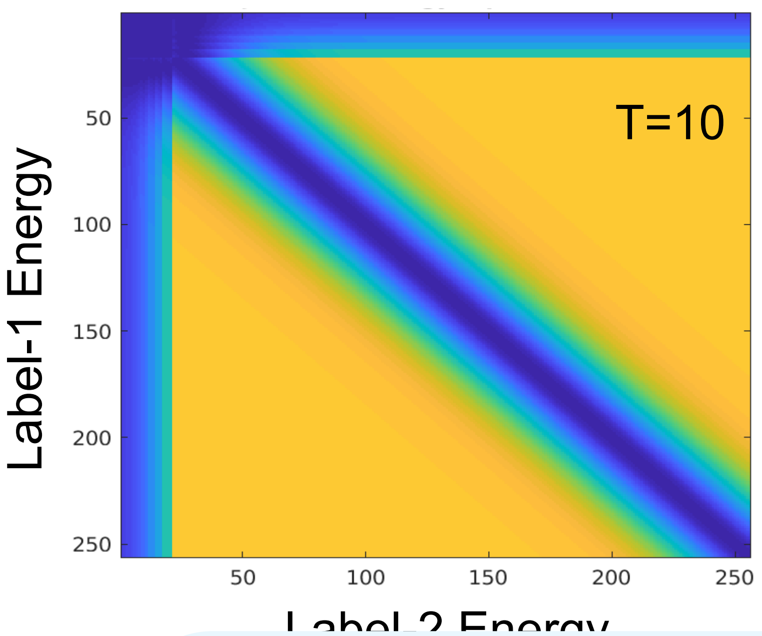
Pillar 3: Goodness of Fit (RMSE) / End-Point Results

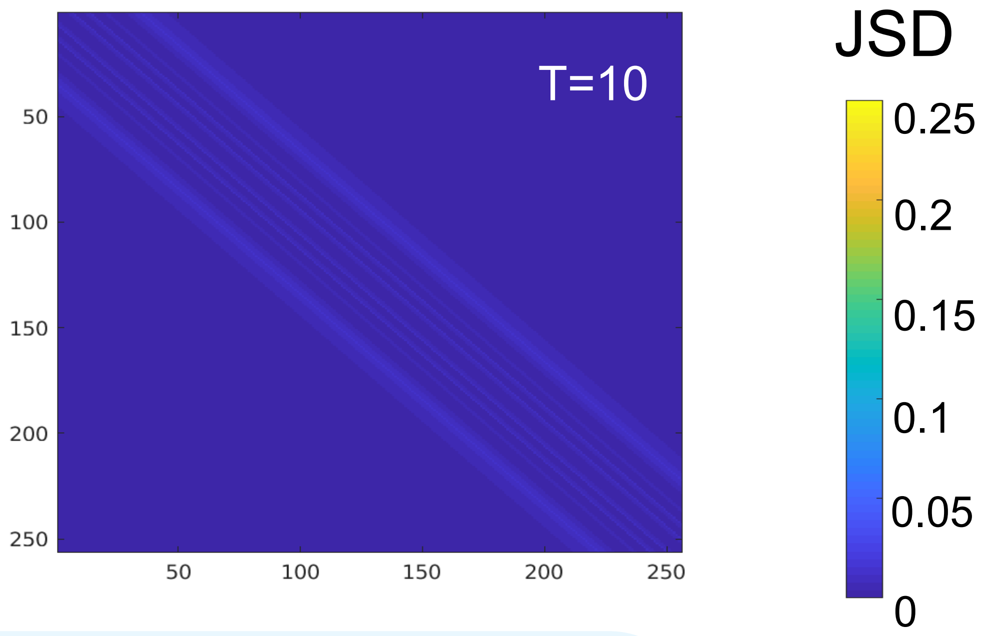




- Reference (per pixel) for RMSE: Mode of 10 software FP64 runs
- RMSE, End-point result quality (BP): **comparable results** (most whiskers overlap)
  - Confirms single-run result quality in slide 5.
  - FP64 **not** always same/better than SPU

Goodness of Fit: Jensen-Shannon Divergence (2-label case)





al., ISCA 2018]

• Ir

• Co

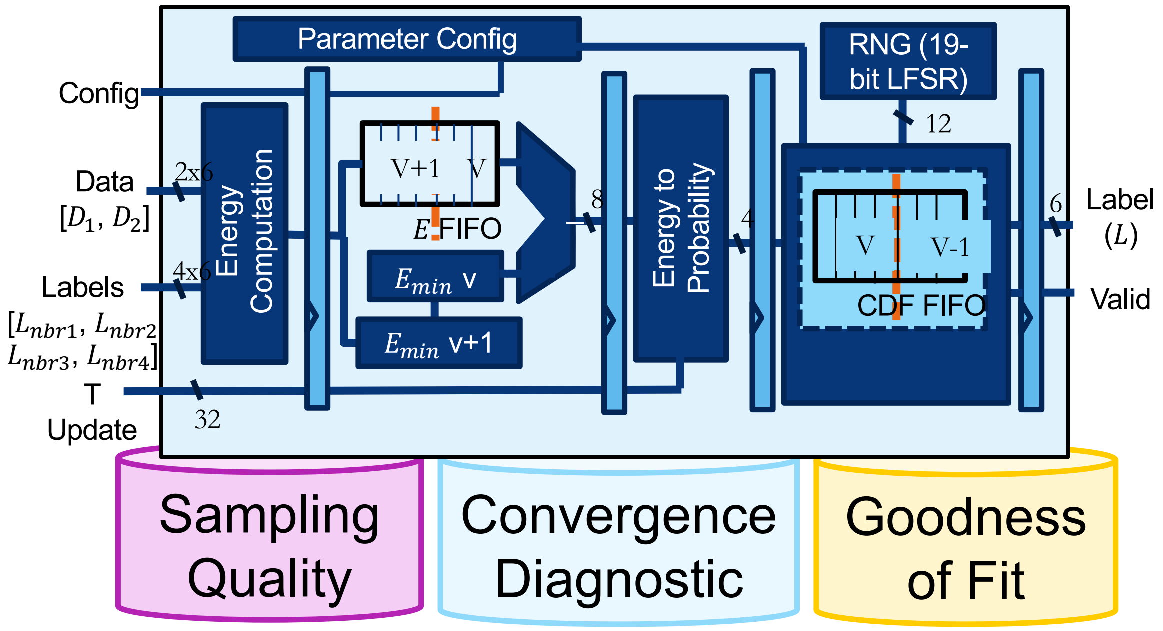
Architectural optimizations:

+ good end-point results

- compromised statistical robustness
- reducing effective speedups by 2x

Using the Three Pillars

[Zhang et al., ISCA 2018]



Inform User

Characterize Existing Hardware

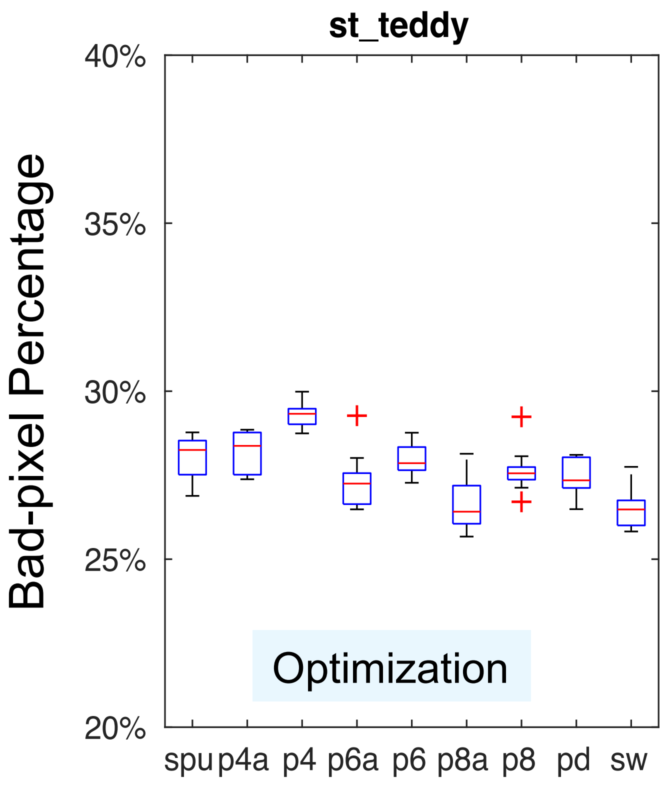
Inform HW Designer

Design space exploration

How to remove the 2x overhead?

- with minimum area/power overheads

What If Only Using End-point Result Quality…

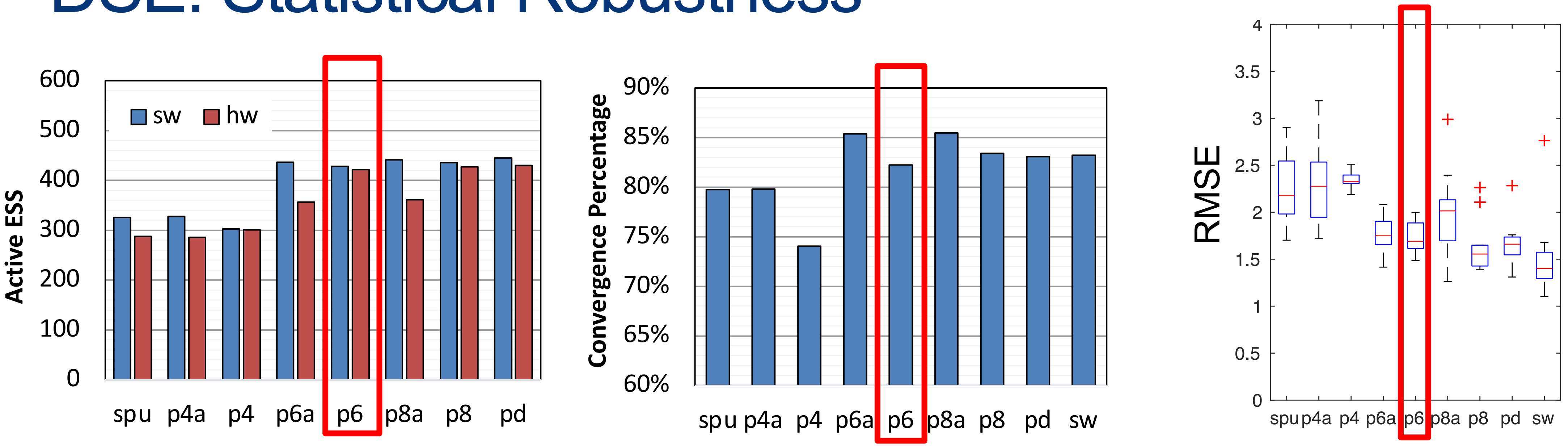


Design Alternatives

- Difficult to tell which is the best design
- Need three pillars to provide insights

Precision ↑ Area ↑ Power ↑

DSE: Statistical Robustness

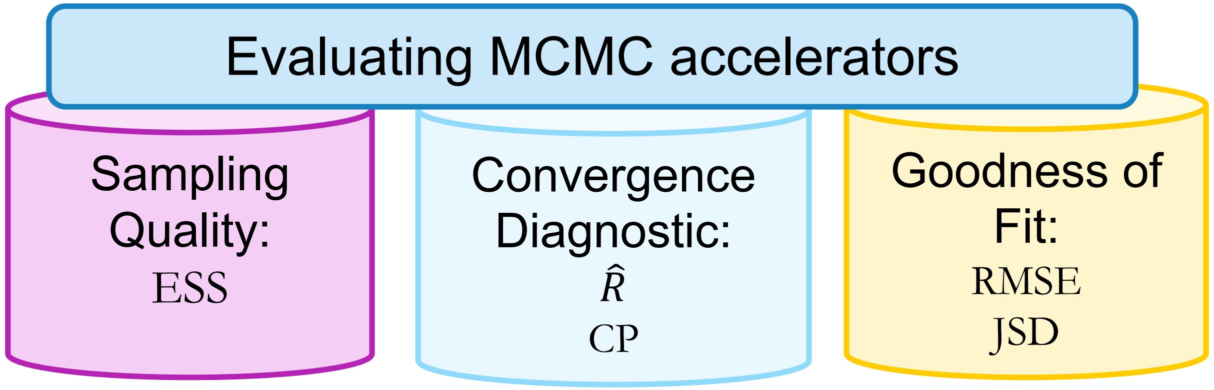


Sampling Quality | Convergence Diagnostic | Goodness of Fit: RMSE (sampling)

Stereo vision: teddy

- **Achieves statistical robustness comparable to FP64**
  - Probability bits 4->6, remove 2^n approximation
- 19-bit LFSR is good.
- Hardware resources ("p6"):
  - Modest increases: 20% area and 10% power
  - **much lower area/power vs. FP HW**

Conclusion



- We claim correctness is defined by more than end-point results.
- We propose three pillars to quantitatively evaluate statistical robustness.
  - Inform user: characterize existing hardware
  - Inform HW designer: design space exploration
- A design might have good end-point results but compromised statistical properties.
- Slight increase in precision achieves FP64 results.
- For broader use: appropriately address **uncertainty**.