SAND2011-1109C

Fast Generation of Nested Space-filling Latin Hypercube Sample Designs

Keith Dalbey, Ph.D.
Sandia National Labs, Dept 1411, Optimization and Uncertainty Quantification
George N. Karystinos, Ph.D.
Technical University of Crete, Depart. of Electronic and Computer Engineering

Feb. 28 – Mar. 4, 2011

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Outline

- Sampling: Why & What's Good?
- Sample Design Quality Metrics
- "Binning Optimality," a New Space-filling Metric
- Latin Hypercube Sampling (LHS)
- Jittered Sampling
- Binning Optimal Symmetric Latin Hypercube Sampling (BOSLHS)
- Conclusions
- Current Work

Sandia National Laboratories

Sampling: Why & What's Good?

Problem: generate a M dimensional sample design with N points at which to evaluate a simulator

Why sample simulator input?

- To calculate statistics of outputs with uncertain inputs
- To optimize e.g., guess several times and pick best guess
- To construct meta-models (fast surrogates for slow simulators)

What qualities do we want in a sample design?

- Design should be **space-filling**
- **Low-dimensional projections** of points should be **well spaced**
- Sample point locations should be uncorrelated with each other
- **Regularity is bad**, leads to biased results
- **Nesting**: want a SEQUENCE of designs that inherit all points from earlier members in the sequence

Sample Design Quality Metrics

- Lots of metrics; fortunately one of them is almost always the most important
- **"Discrepancy"** (some norm of difference between points per sub-volume and uniform density)**: lower is better**
  - "Koksma-Hlawka-like inequality" bounds error in a computed mean in terms of discrepancy
  - **Centered L2 Discrepancy (usually most important metric)**
  - Wrap-Around L2 Discrepancy (important for periodic variables)
- Unfortunately, discrepancy is expensive ($O(M N^2)$ ops) to calculate for designs with large numbers of points, N, so...
- Can't guess a large number of designs & pick the best
- **WARNING: Regularity is easy way to get low discrepancy**

Sample Design Quality Metrics
Other "partial" metrics

- **"Coverage"** (fraction of hypercube's volume filled by convex hull of points, VERY expensive for even moderately high dimensions)**: higher coverage is better**
- **Condition number of** sample design's **correlation matrix** (can be evaluated in $O(M^2N)$ ops)**: lower is better**
- **"t" quality metric** when design is considered to be a **tms-net** (quasi-Monte Carlo; metric moderately expensive $O({}_{(m-t+1+s)}C_s\ s\ b^m)$ ops where $s=M$, $b^m=N$)**: lower "t" is better**
- **NEW!** ***degree*** **of Binning Non-Optimality** (can be evaluated in $O(N \log(N))$ time)**: lower is better**

"Binning Optimality" a New Space-filling Metric

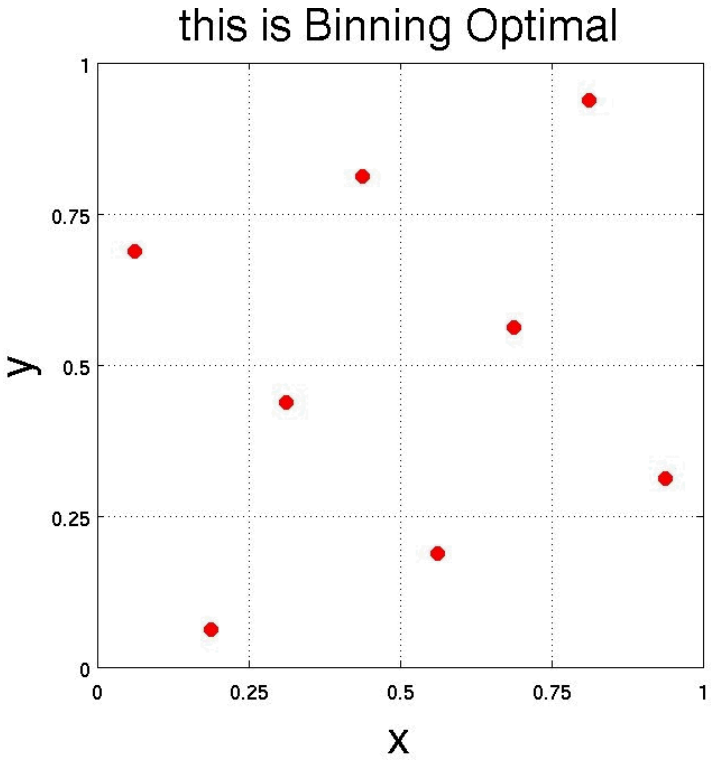
A sample design is "Binning Optimal" (in base 2) **if**

Short answer:

Every sub-bin that should contain a point does

Long answer:

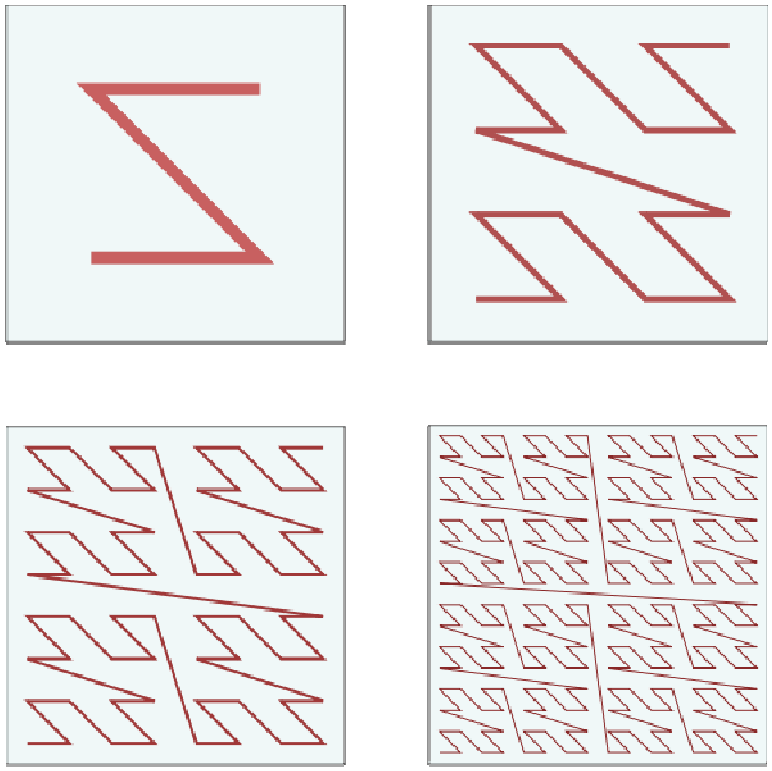
- When you **recursively** subdivide M-dimensional hypercube into 2^M disjoint congruent sub-cube bins, all bins of same generation contain same number of points
- The above must hold true until bins are so small that they each contain either 0 or 1 points



Sandia National Laboratories

"Binning Optimality" Can Be Evaluated in O(N log(N)) Ops

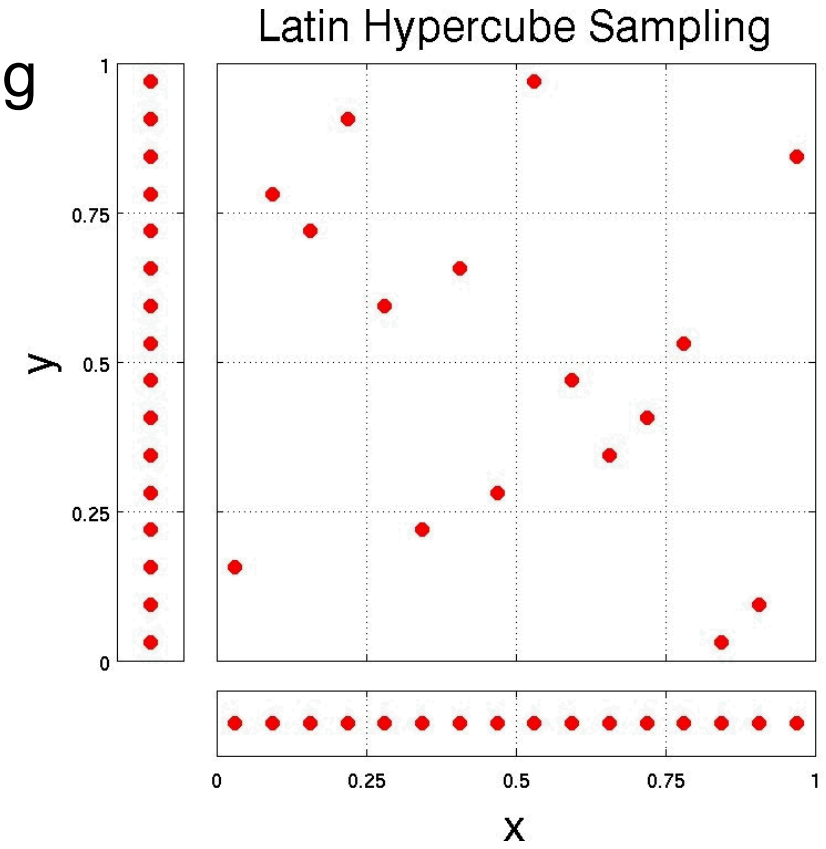
- Generate bin ids as indices into a Morton space-filling curve, also known as a "Z-curve" O(N log(N))+O(N M) work
- Quicksort bin ids O(N log(N)) work
- Tally bins ids: O(N) work



- A FFT of difference of sequential sorted Z-curve bin Ids reveals regularity (cyclic patterns)

Latin Hypercube Sampling (LHS)

- Form of stratified random sampling that converges with fewer points than Monte Carlo Sampling
- Each column contains 1 point
- Each row contains 1 point
- **Quality** of design **depends on pairing of dimensions** used to form points (tough problem)
- Cell-centered LHS with randomly paired dimensions
  - **gets 1D projections "perfect"**
  - **Is NOT space-filling**

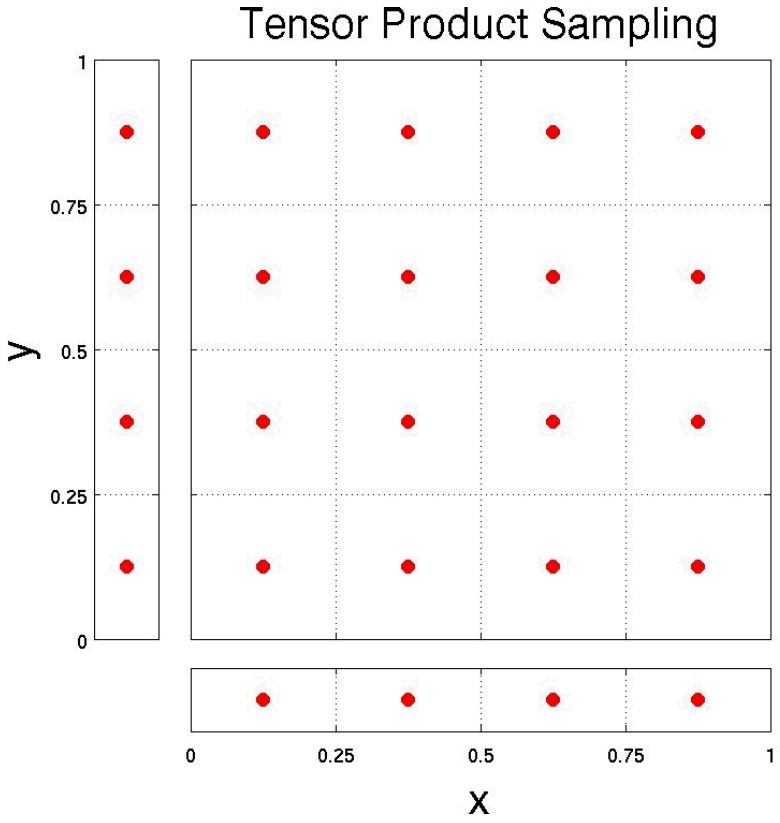


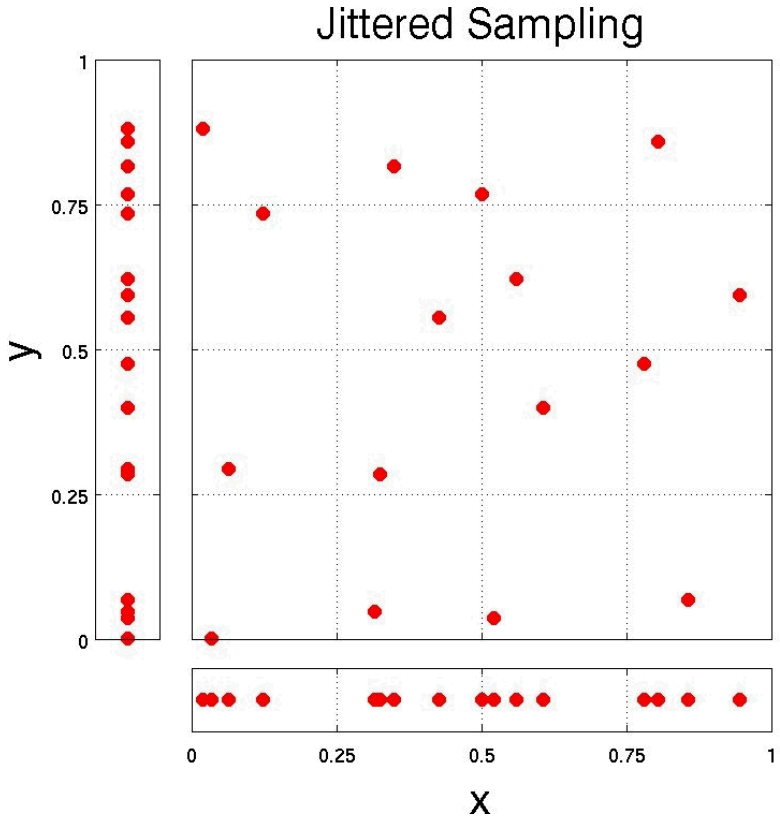
This is not Binning Optimal

Sandia National Laboratories

Jittered Sampling

- Jittered Sampling = Tensor product sampling + random offset
- Better 1D projections than Tensor Product sampling
- **Worse 1D projections than LHS**
- Each cell contains a point $\Rightarrow$ **space-filling** as cell size $\rightarrow 0$



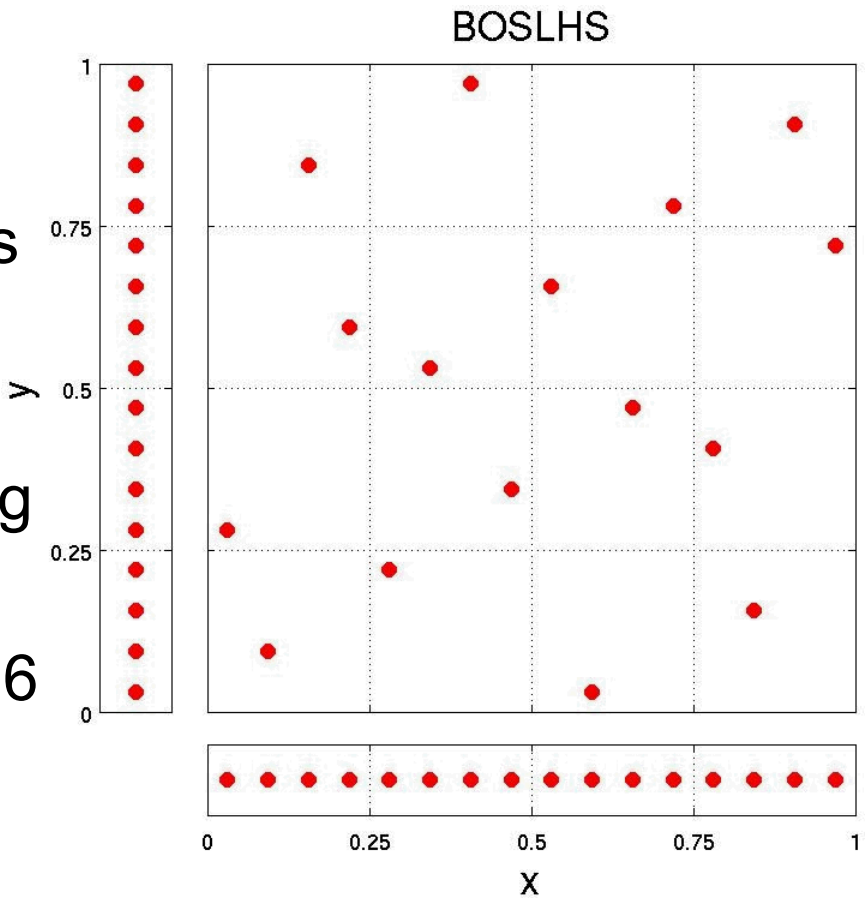


These are Binning Optimal

Sandia National Laboratories

Binning Optimal Symmetric Latin Hypercube Sampling (BOSLHS)

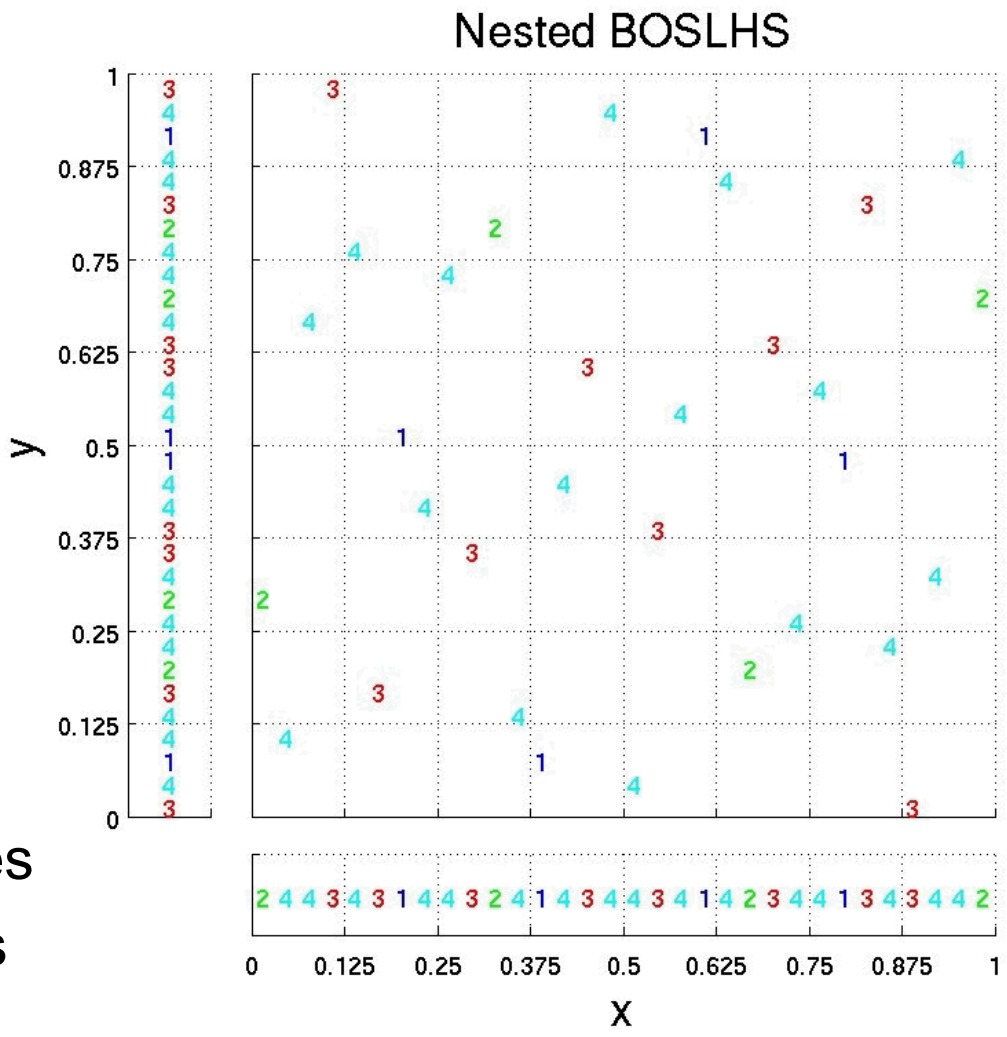
- Gets 1D projections right
- Is space-filling
- Combines most of best features of LHS and Jittered sampling
- Design quality is better than regular LHS or Jittered sampling
- Is **very** fast: generated **Nested** BOSLHS M=8 dim, $N=2^{16}=65536$ points design in 8.21 seconds
- Currently limited to $M=2^p \leq 16$ dimensions (low degree of binning non-optimality for non integer p, working on extending to $M > 16$)



Sandia National Laboratories

Nested BOSLHS Algorithm

1. Start with (lower Z half of) small BOSLHS design
2. Choose new set of bins that are maximally spaced from old bins
3. Generate a new BOSLHS by randomly filling new bins 1 dimension at a time
4. Combine old & new designs, split each row/column/etc. in half, & randomly send each half 1 of duplicate coordinates
5. Repeat steps 2 through 4 as many times as desired.



Sandia National Laboratories

Higher Dimensions Are More Complex

- Need log2(N)/1 bits to uniquely identify each 1D bin
- Binning Optimality in M-D sets first log2(N)/M bits per dimension (BPD)
- BOSLHS matches first log2(N)/M bits of 1D designs to M-D design; “random” matching of remaining bits (step 3 of previous slide)
- But can use Binning Optimality in subsets of dimensions to match bits log2(N)/M+1$\rightarrow$log2(N)/2
- First cut: randomly match first log2(N)/M BPD of M/2 2D BOSLHS designs to M-D design

Higher Dimensions Are More Complex

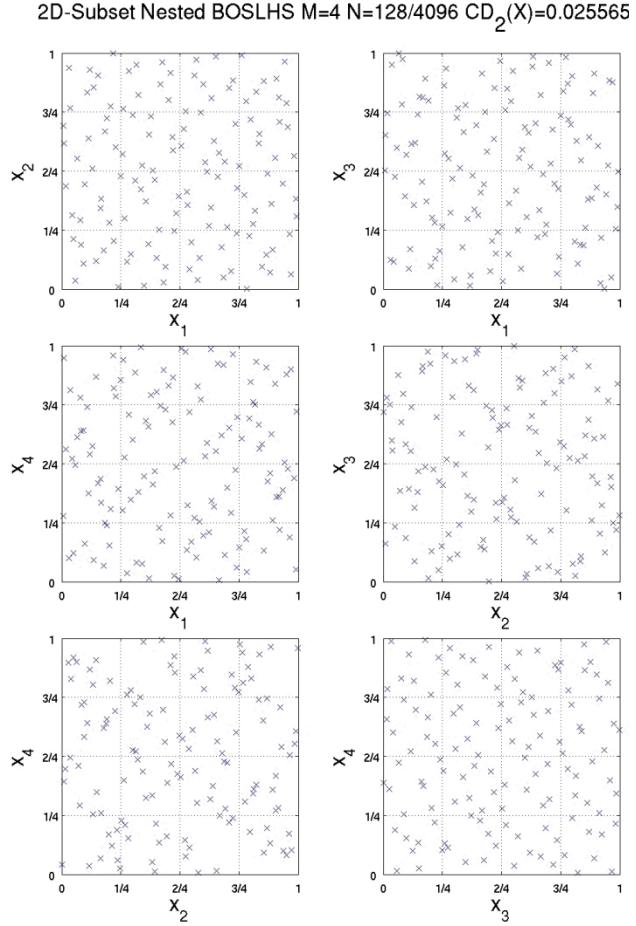
- Bit # ceil(log2(N)/M) “tricky” when log2(N)/M not integer
- Bins/Octants for that bit must be max spaced in M-D; solution is endpoints of max spaced rotated orthogonal axes (see next slide), but getting max spaced subsets of dimensions is “trickier”
- Nesting makes bit # ceil(log2(N)/M) “trickier”
- Ensuring symmetry makes things “trickier”

Higher Dimensions Need a Maximally Spaced List of Octants

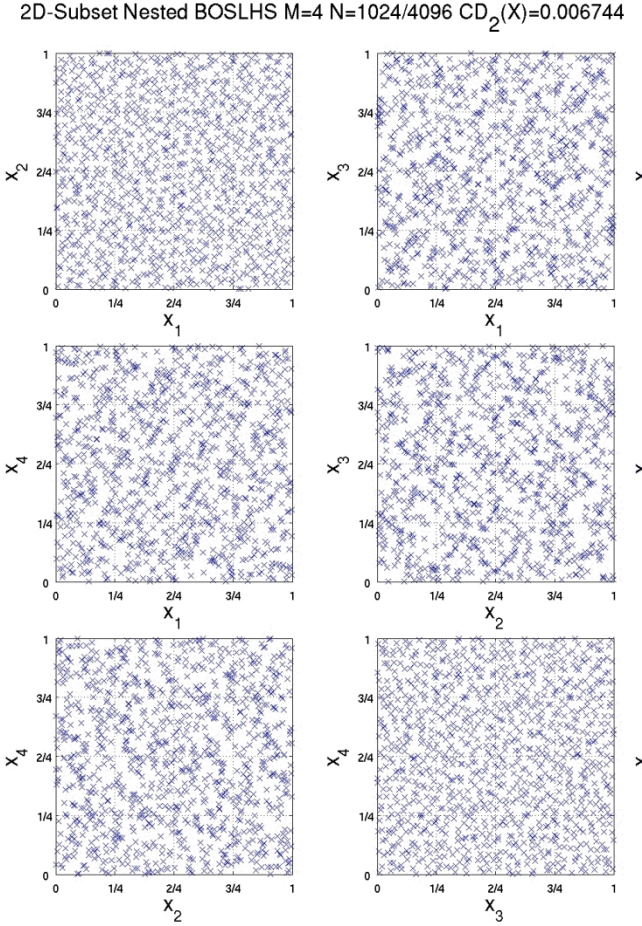
- Generating list is simple for up to M=8 dimensions. It's difficult beyond that BUT…
- It's similar to digital communication problems
- Collaborator, Professor George N. Karystinos of Technical University of Crete (Department of Electronic & Computer Engineering), found a group theory solution for arbitrarily large dimensions
- But… memory requirements prevent even listing the octants for $M \geq 32$
- Working on generating maximally spaced partial list

Results: Eyeball Metric M=4D

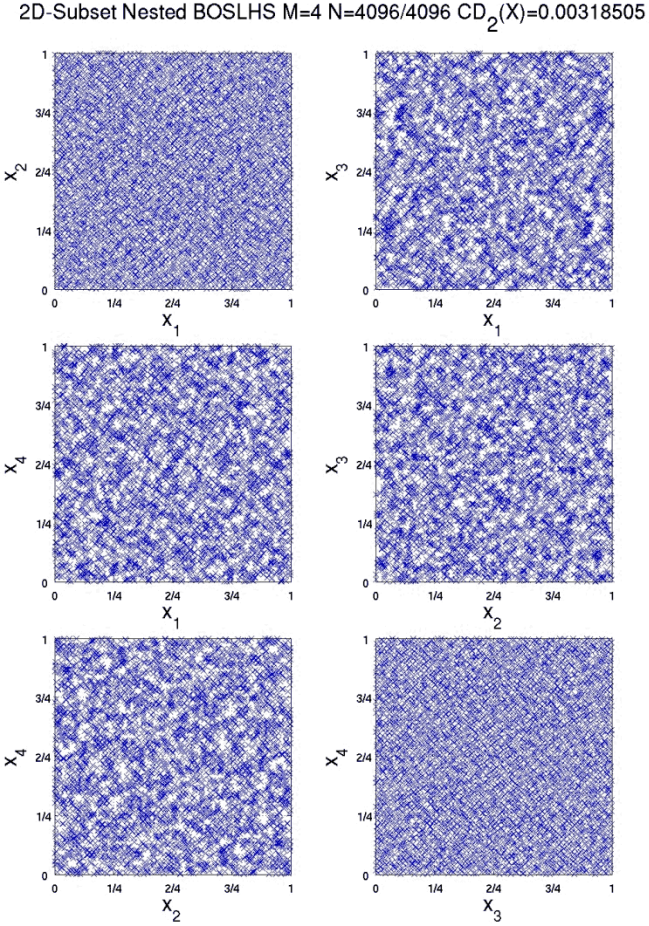
N=128



N=1024



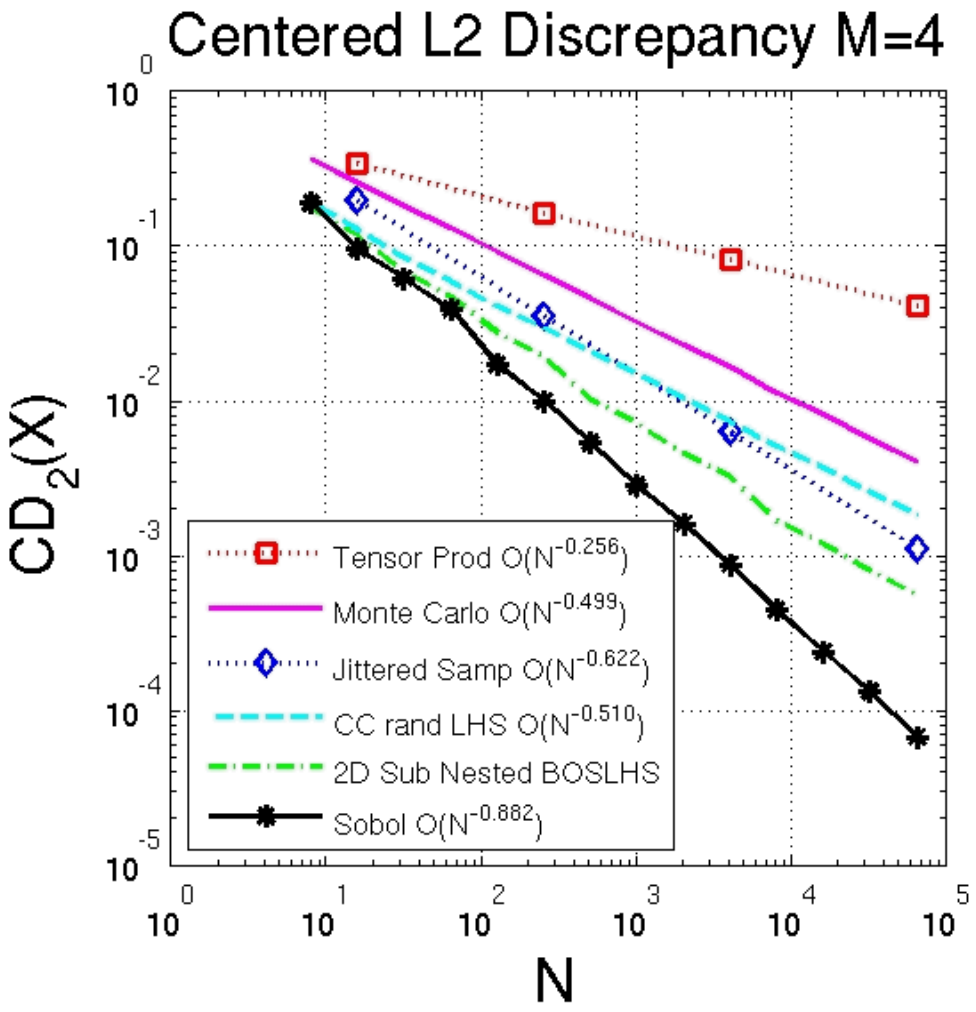
N=4096

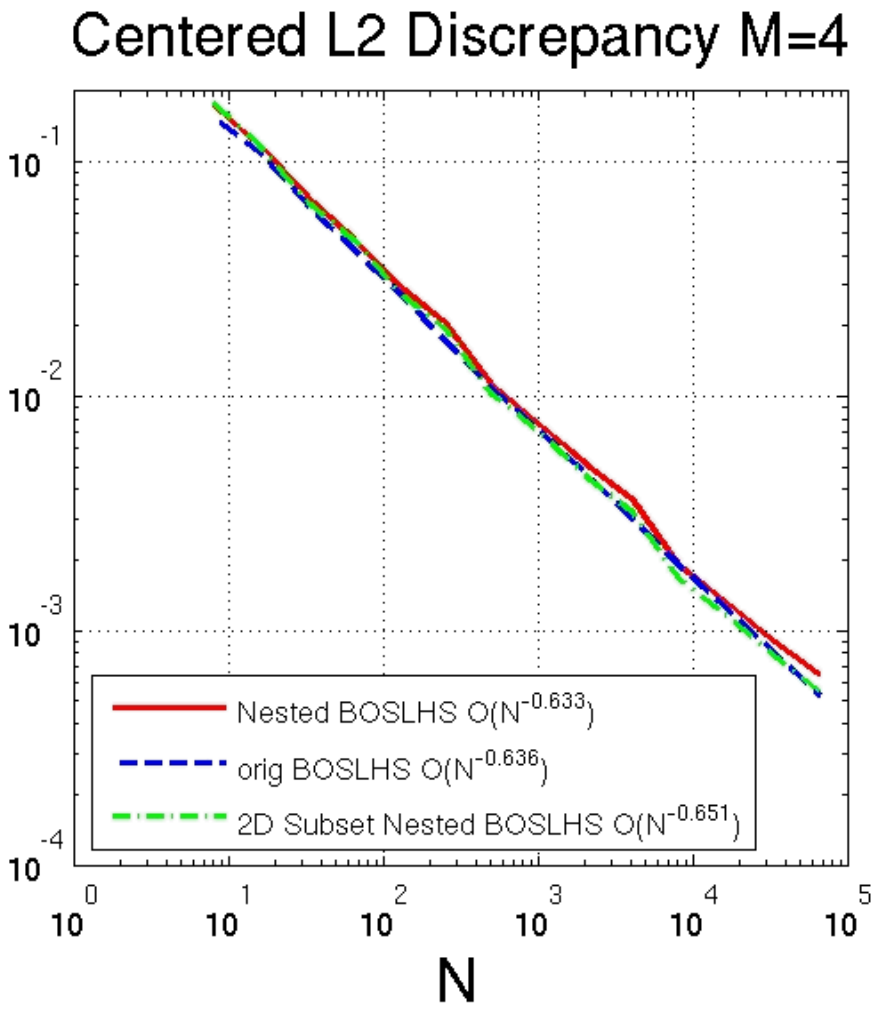


- Plotted all 6 combinations of 2 out of M=4 dimensions
- BOSLHS is visibly space-filling!

Sandia National Laboratories

Results: Centered L2 Discrepancy (Lower is Better)

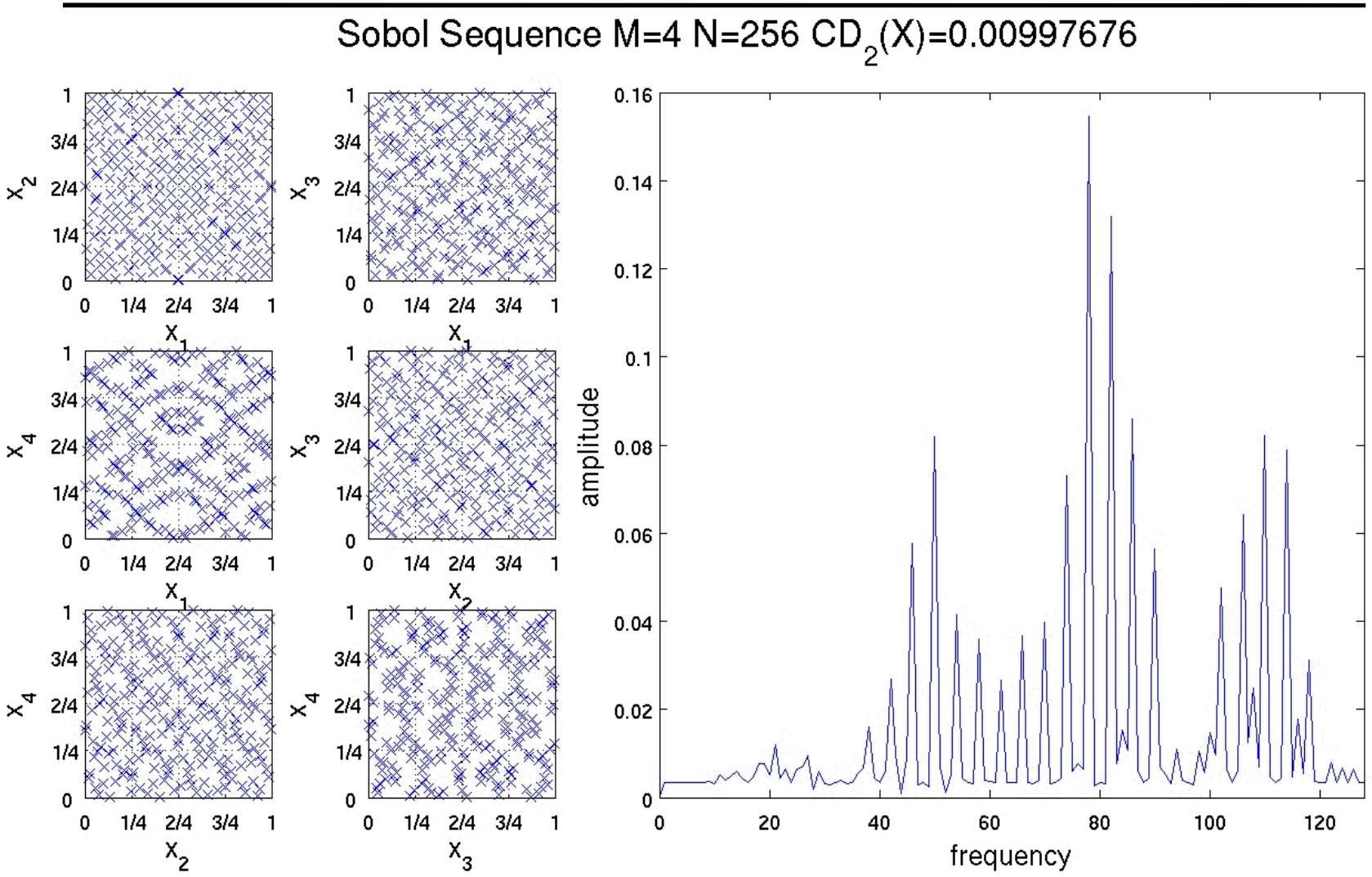




Plots are for average of 40 random designs

Sandia National Laboratories

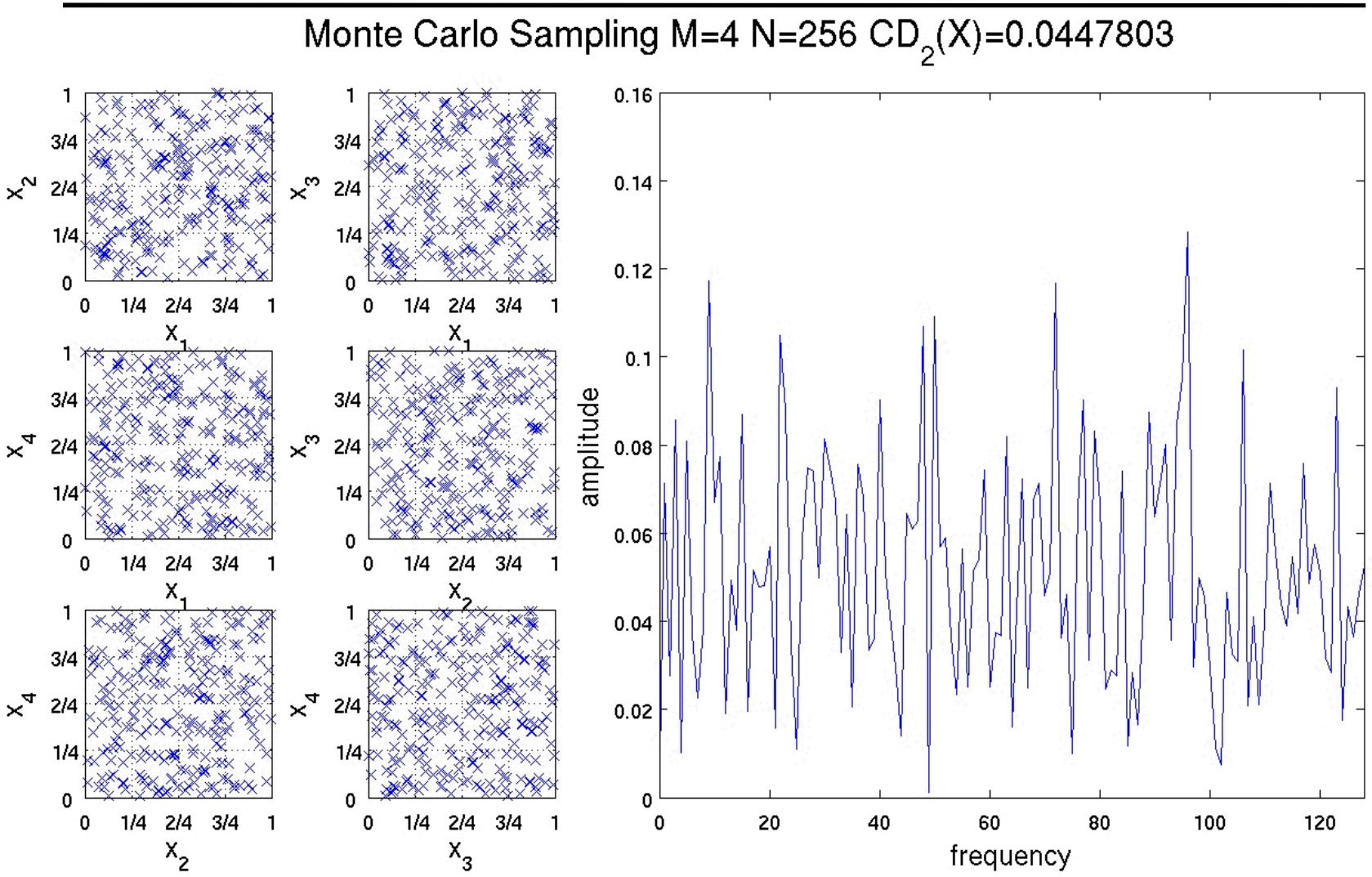
Results: Sobol Sequence Has Lower Discrepancy But Is Regular



Regularity in sample designs results in biased statistics

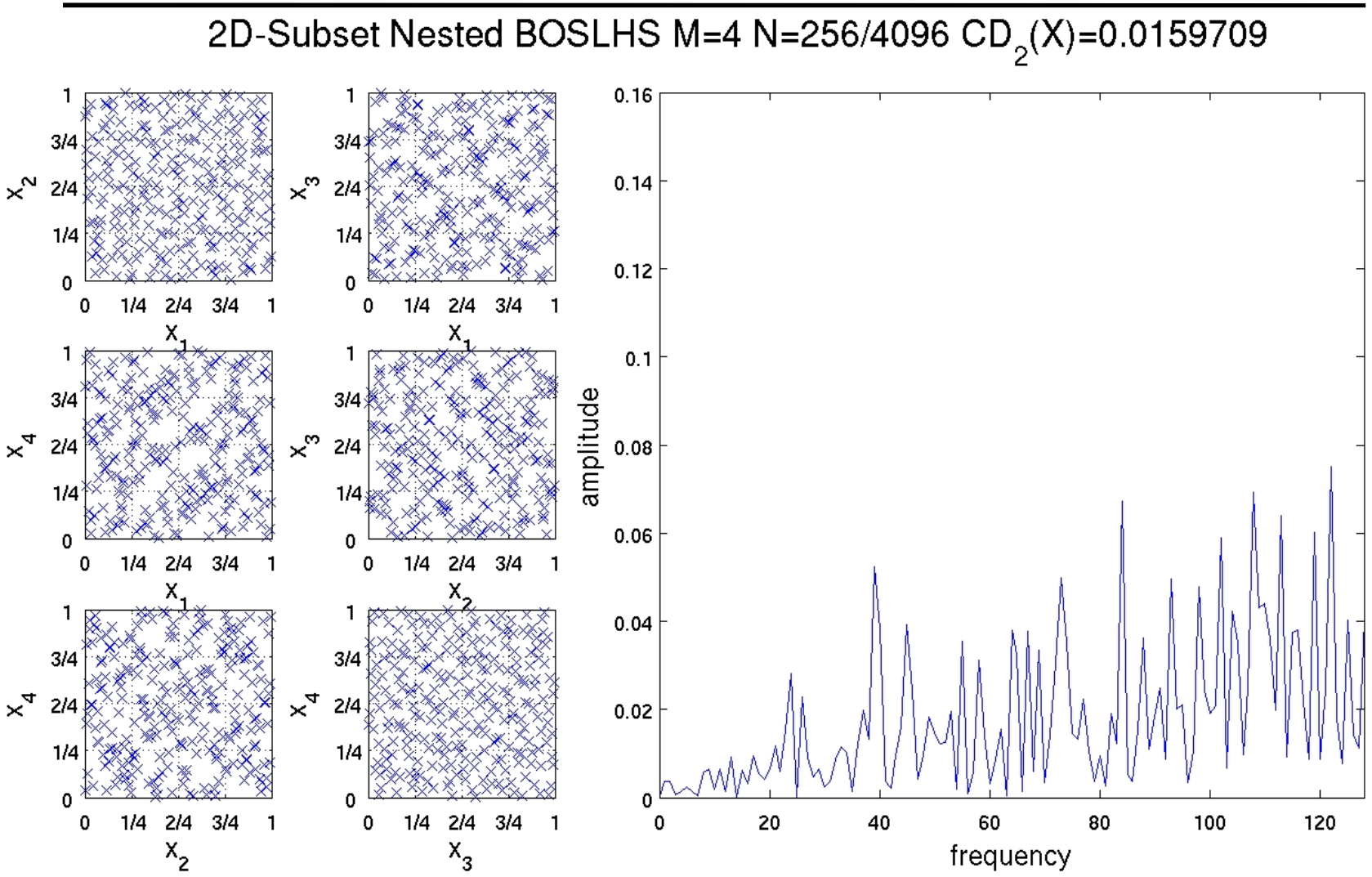
Sandia National Laboratories

Results: What Complete Irregularity (Monte Carlo Sampling) Looks Like



Sandia National Laboratories

Results: Nested BOSLHS Is Not Regular



Sandia National Laboratories

Results

- BOSLHS has low discrepancy without being regular
- BOSLHS also scores well in other metrics: it has high "coverage," low correlations between dimensions, and a low (t,m,s)-net rating
- **VERY fast**: MATLAB generated a $N=2^{16}$ point M=8 dimensional space-filling **nested** BOSLHS design in ~8.21 seconds on an Intel 2.53 GHz processor (algorithms reported in literature take "minutes" for **non-nested** space-filling N = 100 point designs)
- By comparison, it took ~298.2 seconds ($O(N^2M)$ ops) to evaluate discrepancy for same design

Conclusions

- Defined new space-filling metric "Binning Optimality" that evaluates in O(N log(N)) time
- Found related way to detect regularity in sample designs
- Developed fast algorithm for **Nested** Binning Optimal Symmetric Latin Hypercube Sampling (BOSLHS) that
  - **is also Binning Optimal in some 2D subsets**
  - combines best features of LHS & Jittered Sampling

Ongoing Work

- Current BOSLHS algorithm is space-filling in full M dimensional space and 1 dimensional projections and some 2D subsets. Want to be space-filling in more 2D and other larger subsets of dimensions.
- Extension to larger (> 16) and arbitrary (non power of 2) numbers of dimensions.
- How well does BOSLHS do in other design quality metrics?
- Better numerical quantification of “regularity”
- ?Induce correlations between dimensions?

References

1. K. R. Dalbey and G. N. Karystinos, “Fast Generation of Space-filling Latin Hypercube Sample Designs,” *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2010.*

2. K. R. Dalbey and G. N. Karystinos, “Generating a Maximally Spaced Set of Bins to Fill for High Dimensional Space-filling Latin Hypercube Sampling,” *International Journal for Uncertainty Quantification*, (to appear), 2011.

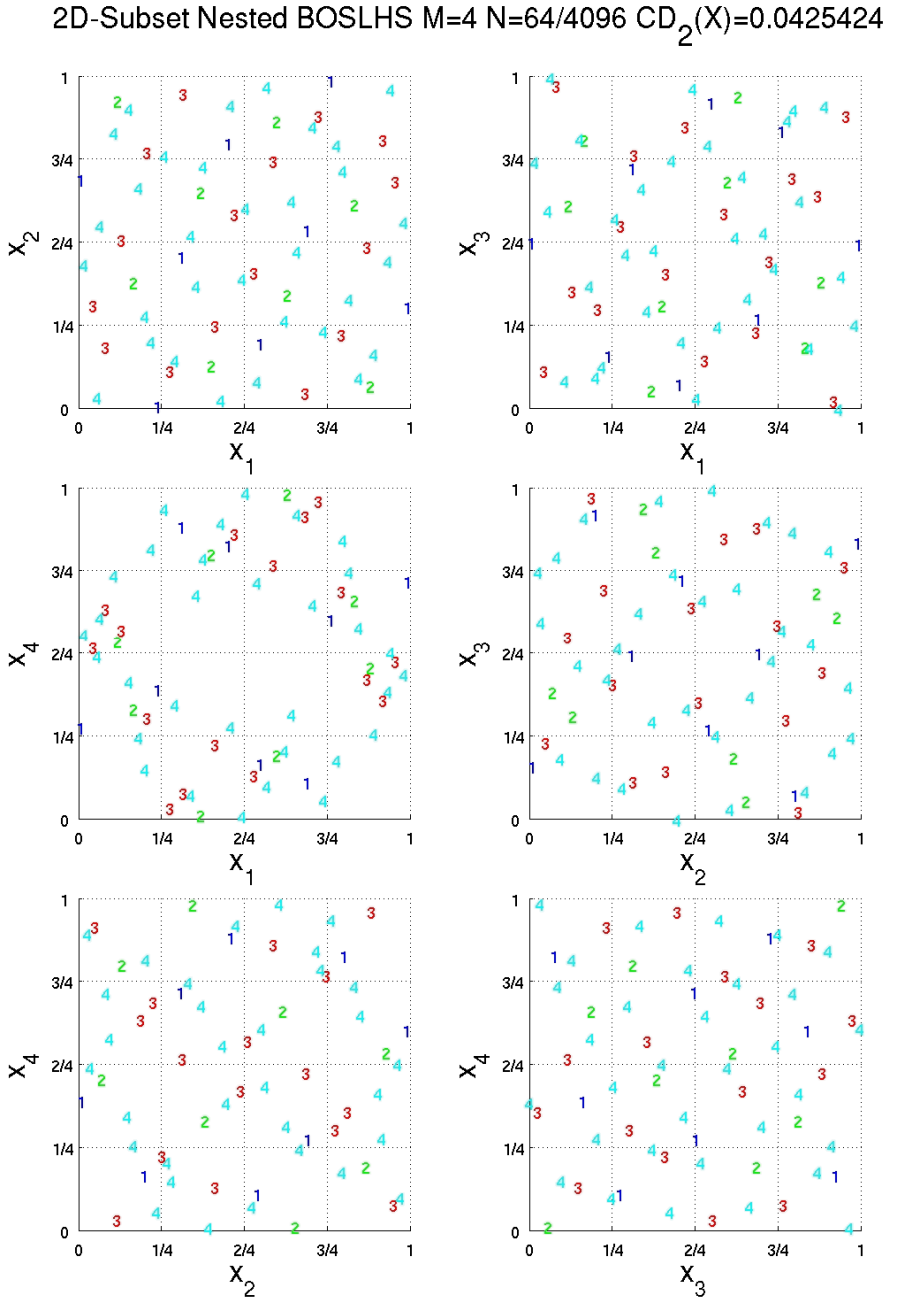
Sandia National Laboratories

Bonus Slides Start Here

Sandia National Laboratories

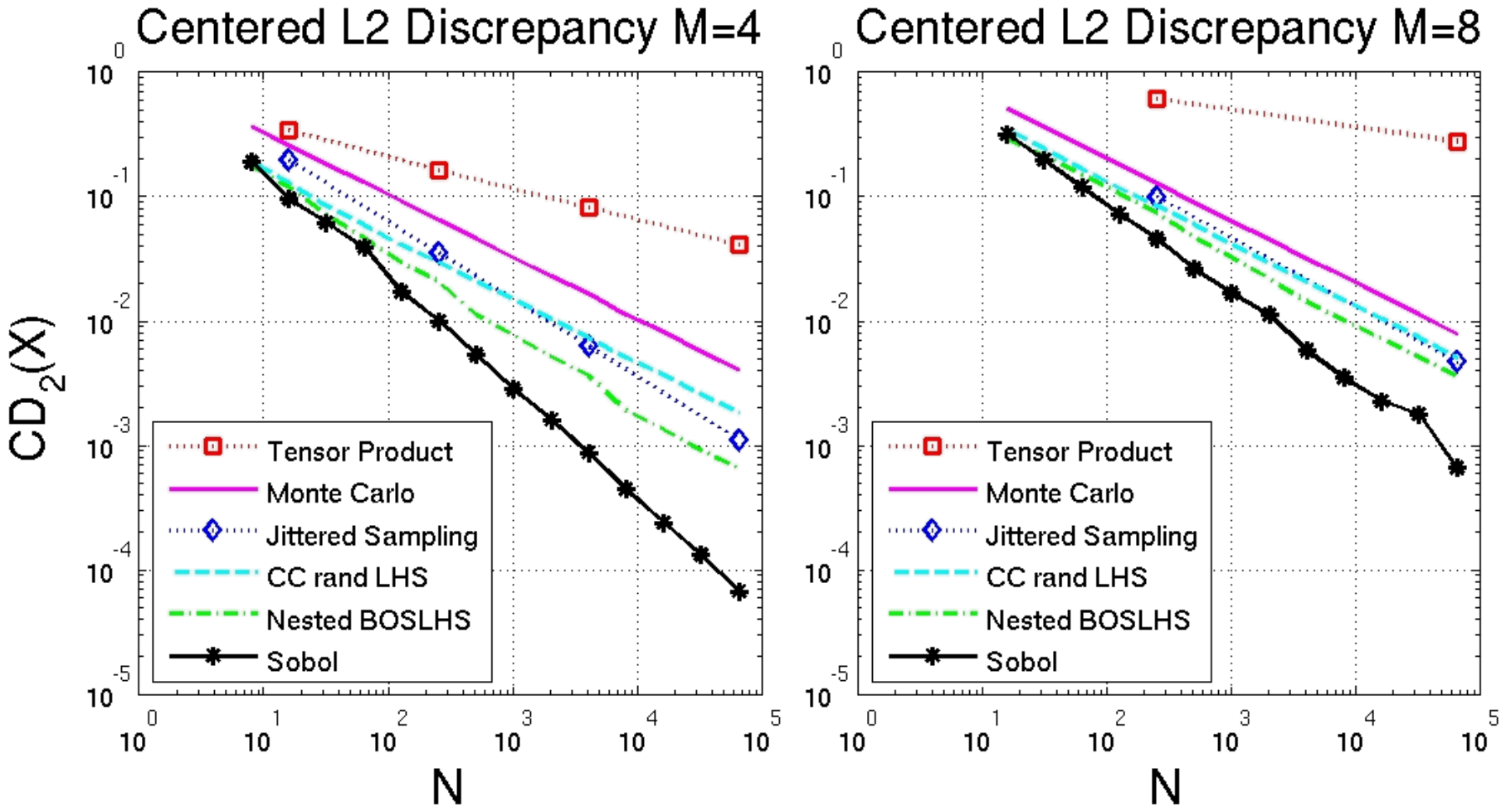
4-D Example

- Difference in 4 dimensions is in choosing maximally spaced bins
- In 2D, only 2^2=4 sub-bins per level, the 2*2=4 end points of 1 "orientation" (rotated set of orthogonal axes)
  - If 1 point in bin, new sub-bin is opposite old one
  - If 2 points (1 axis), 2 new sub-bins are other axis
  - Then go 1 bin deeper
- In 4D, 2^4=16 sub-bins per level, 2 orientations with 2*4=8 bins each
  - After first axis, randomly select order of other axes in same orientation
  - Then choose other orientation
  - Then go 1 bin deeper



Sandia National Laboratories

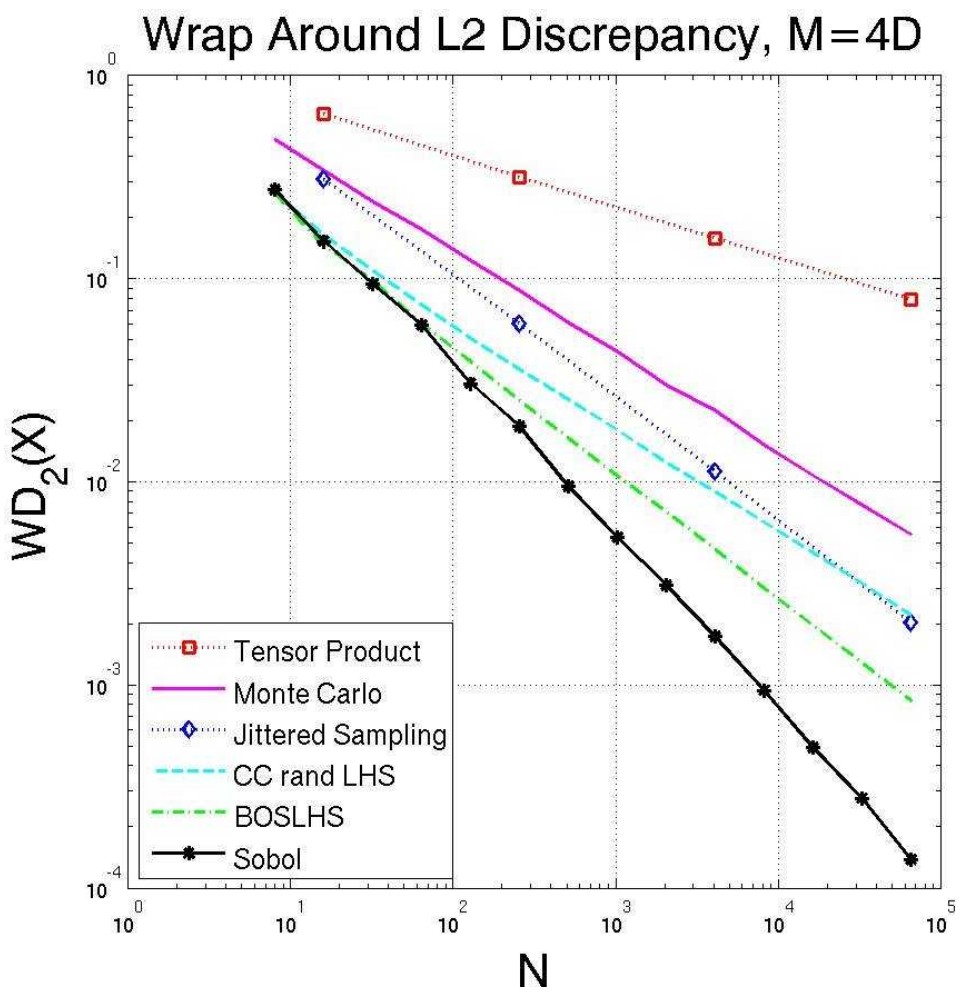
Results: Centered L2 Discrepancy (Lower is Better)

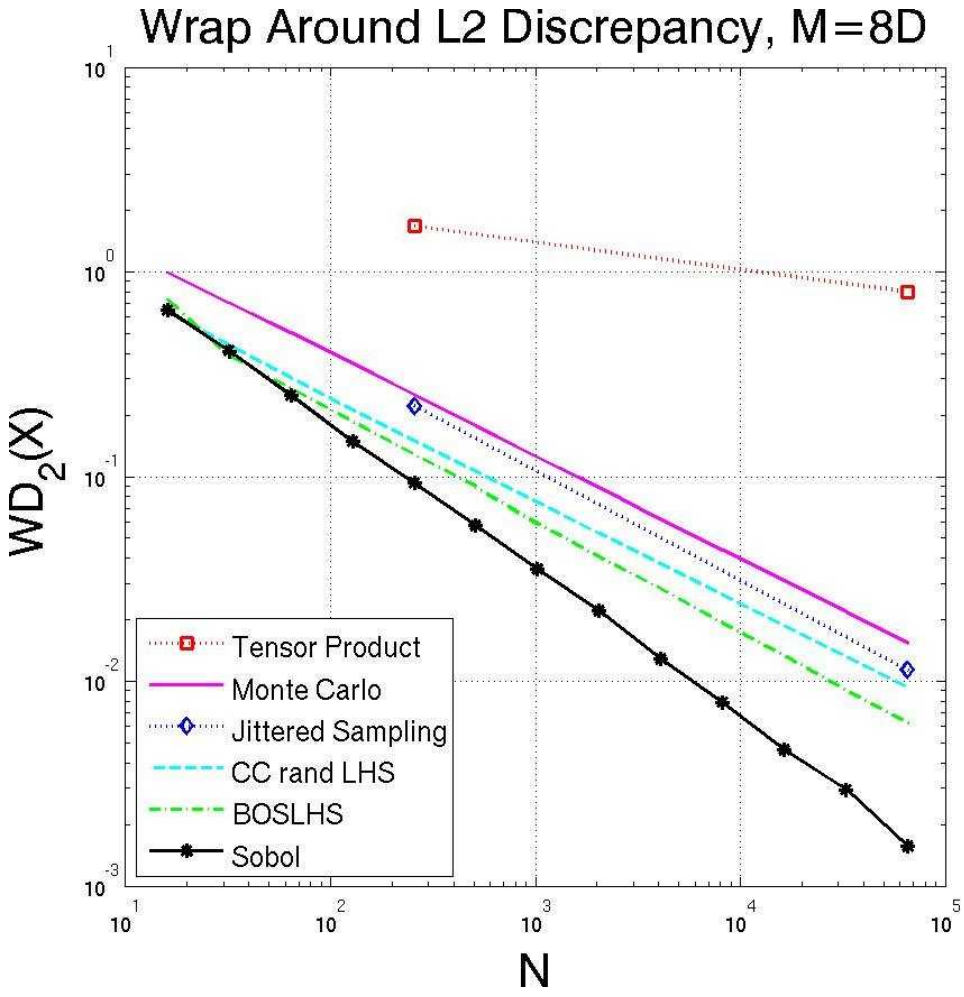


Plots are for average of 40 random designs

Sandia National Laboratories

Results: Wrap Around L2 Discrepancy (Lower is Better)

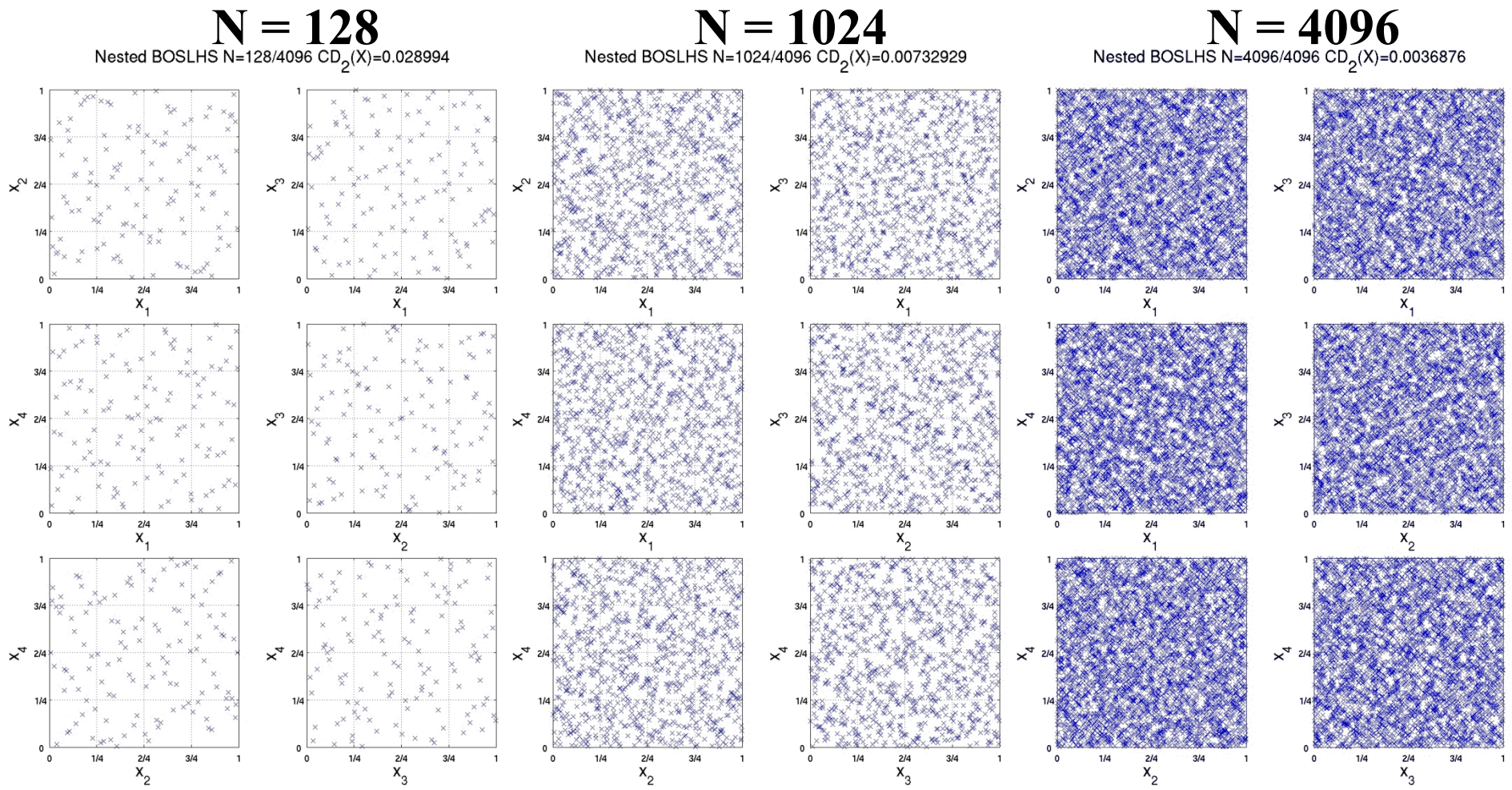




Plots are for average of 40 random designs

Sandia National Laboratories

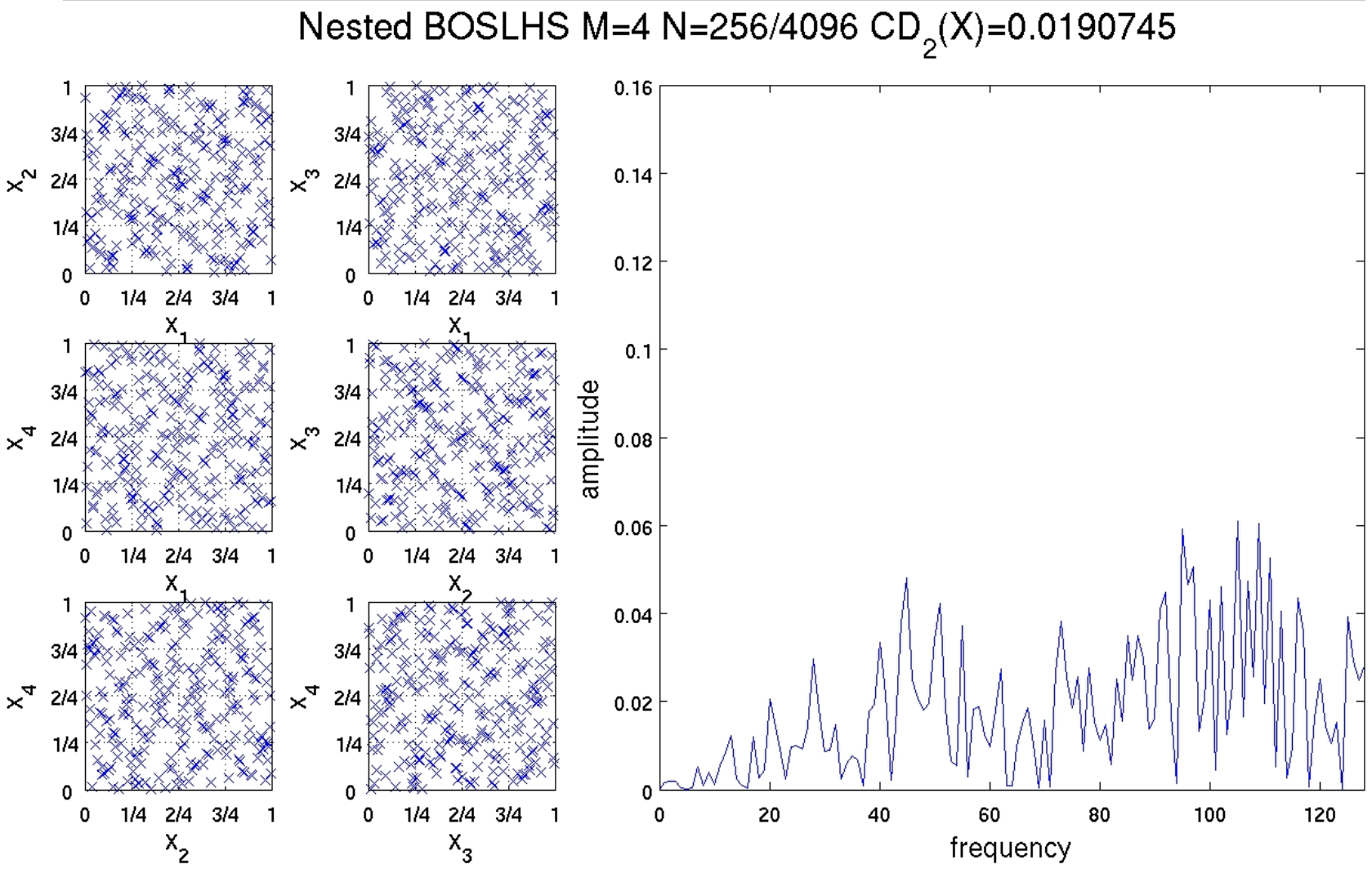
Results: Eyeball Metric M=4D



- Plotted all 6 combinations of 2 out of M=4 dimensions
- BOSLHS is visibly space-filling!

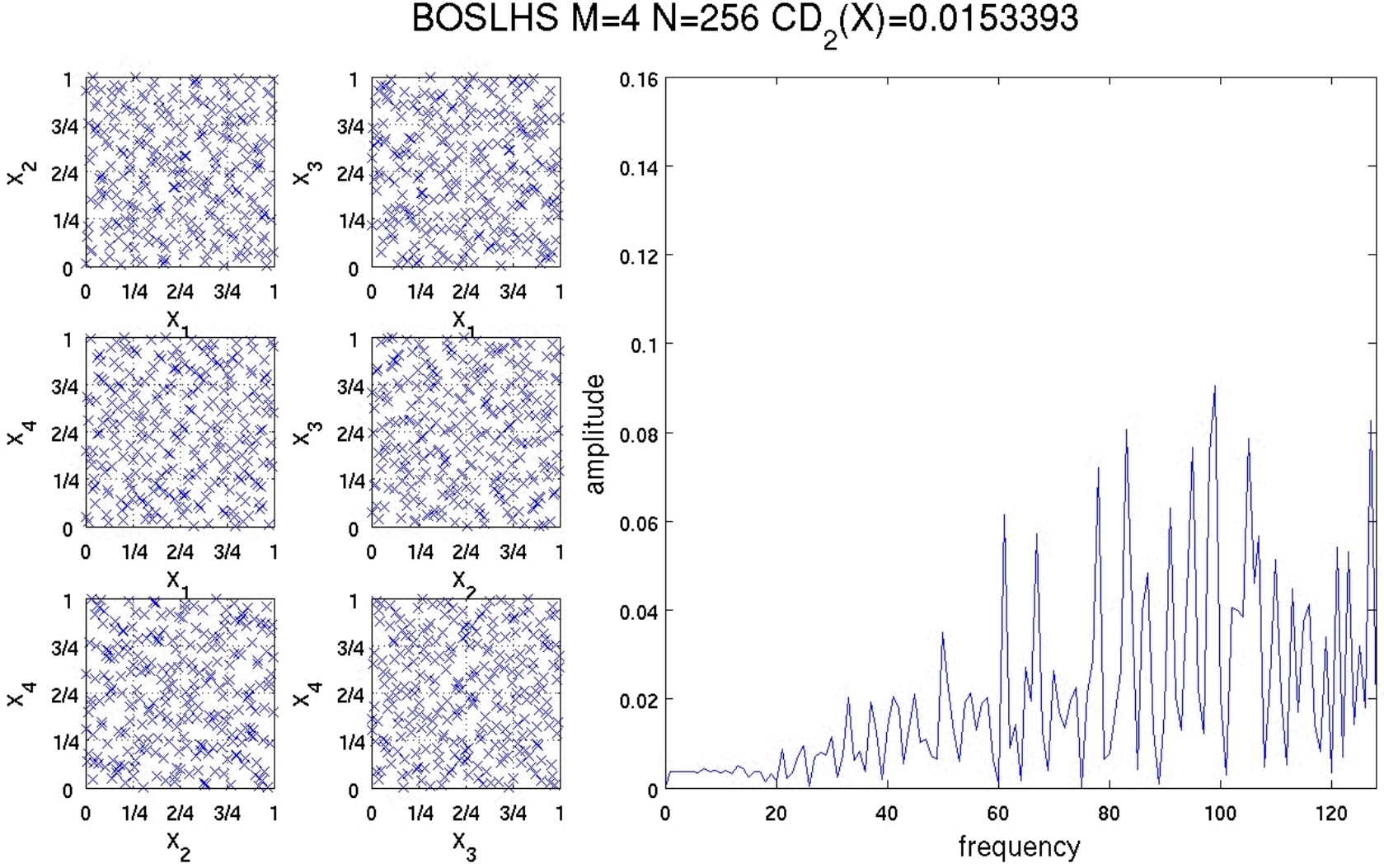
Sandia National Laboratories

Results: Nested BOSLHS Is Not Regular



Sandia National Laboratories

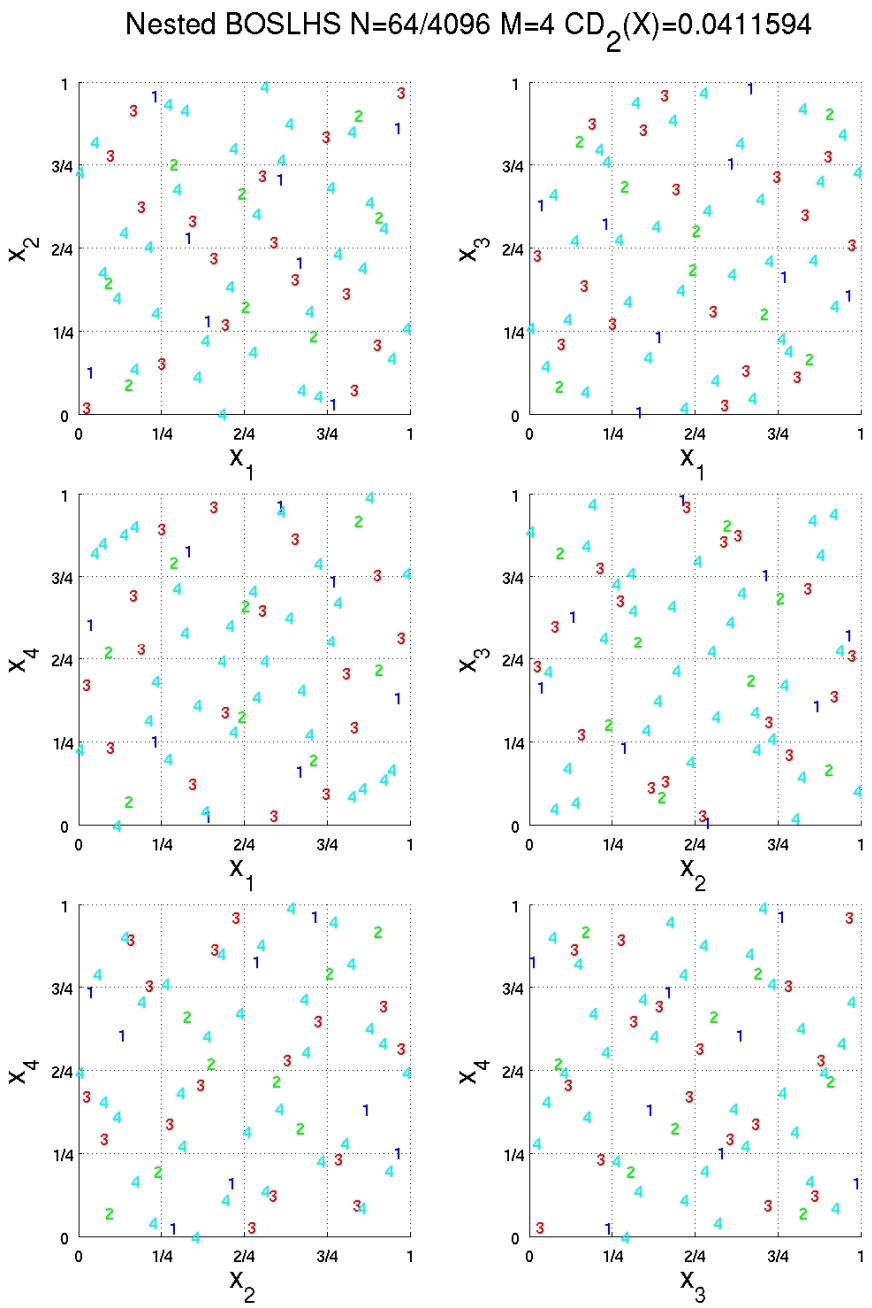
Compared To Original, Nested BOSLHS Less Regular But Higher Discrepancy



Sandia National Laboratories

4-D Example

- Difference in 4 dimensions is in choosing maximally spaced bins
- In 2D, only 2^2=4 sub-bins per level, the 2*2=4 end points of 1 "orientation" (rotated set of orthogonal axes)
  - If 1 point in bin, new sub-bin is opposite old one
  - If 2 points (1 axis), 2 new sub-bins are other axis
  - Then go 1 bin deeper
- In 4D, 2^4=16 sub-bins per level, 2 orientations with 2*4=8 bins each
  - After first axis, randomly select order of other axes in same orientation
  - Then choose other orientation
  - Then go 1 bin deeper



Sandia National Laboratories

Results: Coverage (higher is better)

"Coverage" for $M = 4$ Dimensions: Average of 40 runs

N	Binning Optimal Symmetric LHS	Cell Centered Random LHS	Monte Carlo Sampling	Jittered Sampling	Tensor Product Sampling
8	0.0717773	0.027062	0.0178996		
16	0.135135	0.104126	0.0916156	0.128427	0.0625
32	0.285717	0.233105	0.219465		
64	0.417035	0.372359	0.361626		
128	0.56022	0.522201	0.511982		
256	0.678416	0.647304	0.645049	0.667668	0.316406
512	0.773748	0.754804	0.749725		
1024	0.843177	0.832896	0.831007		
2048	0.896093	0.890245	0.886593		
4096	0.932229	0.928693	0.927748	0.929509	0.586182
8192	0.956723	0.954248	0.953466		
16384	0.97319	0.97129	0.971217		
32768	0.983415	0.982499	0.982312		
65536	0.989815	0.989387	0.98926	0.98965	0.772476

Results: Condition # of Correlation Matrix (lower is better)

Condition Number of the Correlation Matrix for $M = 4$ Dimensions: Average of 40 runs

N	Binning Optimal Symmetric LHS	Cell Centered Random LHS	Monte Carlo Sampling	Jittered Sampling	Tensor Product Sampling
8	1	14.6273	8.23719		
16	3.2505	4.14988	3.75258	2.39394	1
32	1.49974	2.27709	2.15406		
64	1.37672	1.76306	1.82367		
128	1.2064	1.4508	1.49656		
256	1.11022	1.32572	1.33407	1.10916	1
512	1.05589	1.21341	1.2108		
1024	1.0368	1.1546	1.14725		
2048	1.02121	1.09974	1.09939		
4096	1.01246	1.07576	1.07075	1.01254	1
8192	1.00717	1.04643	1.04922		
16384	1.00403	1.03608	1.03365		
32768	1.0027	1.02297	1.02461		
65536	1.00166	1.01872	1.01742	1.00145	1

Results: (t,m,s)-net, "t" quality metric (lower is better)

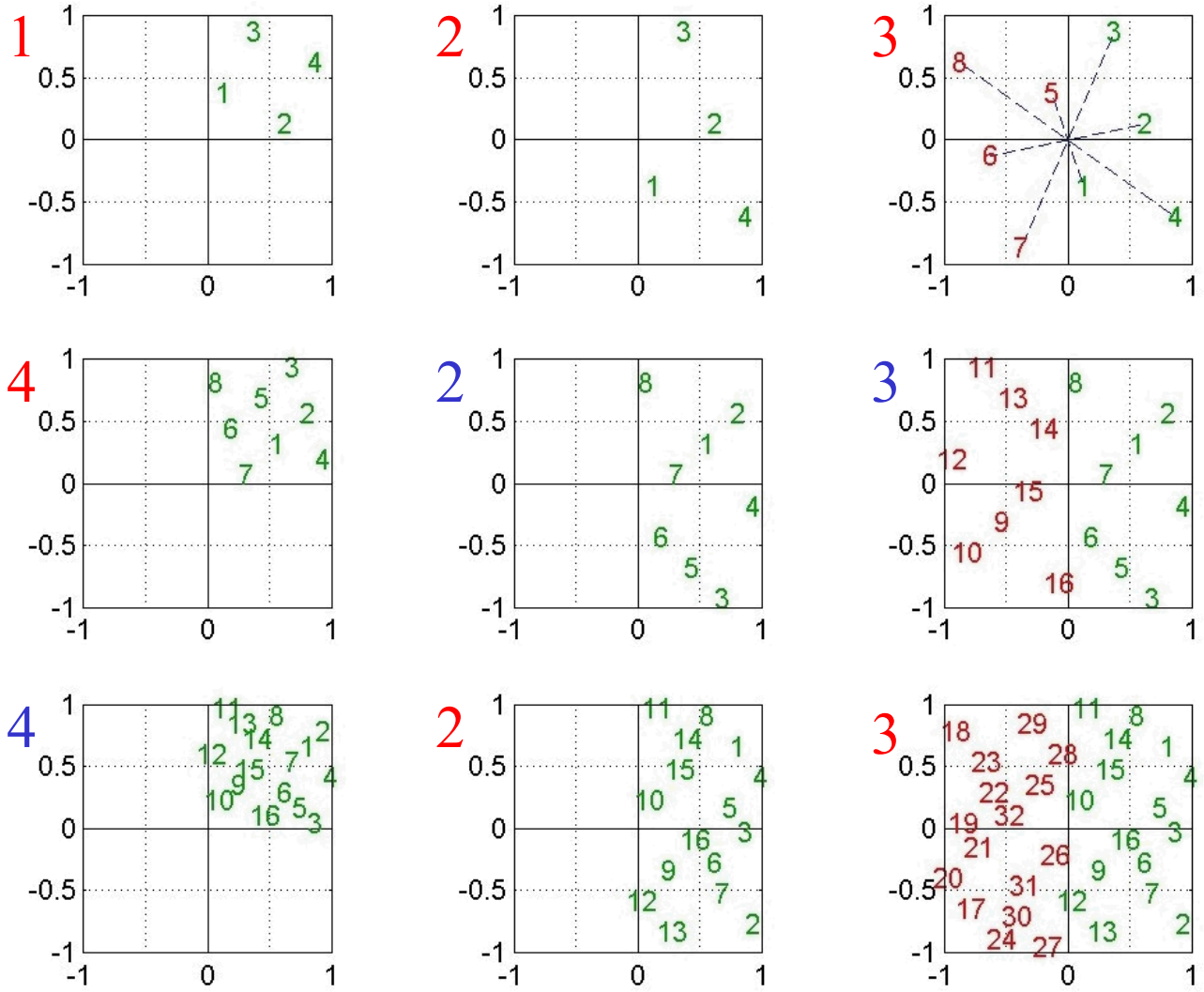
(t, m, s)-net Rating for $M = 4$ Dimensions: Average of 40 runs

N	Binning Optimal Symmetric LHS	Cell Centered Random LHS	Monte Carlo Sampling	Jittered Sampling	Tensor Product Sampling
8	(1 , 3 , 4)	(2 , 3 , 4)	(3 , 3 , 4)		
16	(2 , 4 , 4)	(3 , 4 , 4)	(4 , 4 , 4)	(3 , 4 , 4)	(3 , 4 , 4)
32	(2 , 5 , 4)	(4 , 5 , 4)	(5 , 5 , 4)		
64	(3 , 6 , 4)	(5 , 6 , 4)	(6 , 6 , 4)		
128	(4 , 7 , 4)	(6 , 7 , 4)	(7 , 7 , 4)		
256	(5 , 8 , 4)	(7 , 8 , 4)	(8 , 8 , 4)	(6 , 8 , 4)	(6 , 8 , 4)
512	(5 , 9 , 4)	(8 , 9 , 4)	(9 , 9 , 4)		
1024	(6 , 10 , 4)	(9 , 10 , 4)	(10 , 10 , 4)		
2048	(7 , 11 , 4)	(10 , 11 , 4)	(11 , 11 , 4)		
4096	(8 , 12 , 4)	(11 , 12 , 4)	(12 , 12 , 4)	(9 , 12 , 4)	(9 , 12 , 4)
8192	(8 , 13 , 4)	(12 , 13 , 4)	(13 , 13 , 4)		
16384	(9 , 14 , 4)	(13 , 14 , 4)	(14 , 14 , 4)		
32768	(10 , 15 , 4)	(14 , 15 , 4)	(15 , 15 , 4)		
65536	(11 , 16 , 4)	(15 , 16 , 4)	(16 , 16 , 4)	(12 , 16 , 4)	(12 , 16 , 4)

O(N log(N)) BOSLHS Algorithm

1. Start with n = 2M points that are well distributed in $(0, 1)^M$.
2. Select n/2 of the coordinates in each dimension other than the first to negate in such a way as to obtain n points that are well distributed in $(0, 1) \otimes (-1, 1)^{M-1}$.
3. Reflect the current n points through the origin to create n additional mirror points; this ensures that the design is symmetric.
4. Translate the 2n points from $(-1, 1)^M$ to $(0, 2)^M$, scale them to $(0, 1)^M$, and then set n = 2n.
5. Repeat steps 2 through 4 until the desired number of points has been obtained, i.e. until n = N.

O(N log(N)) BOSLHS Algorithm



Sandia National Laboratories

O(N log(N)) BOSLHS Algorithm

The tough part is step 2

Select n/2 of the coordinates in each dimension other than the first to negate in such a way as to obtain n points that are well distributed in $(0, 1) \otimes (-1, 1)^{M-1}$.

The easy (fast) answer is to recast the problem...

- Don't try change signs of dimensions individually
- **Send nearby points to octants that are far apart**

The Z-order quicksort will put nearby points in sequential order in O(N log(N)) ops

We just need a listing of octants in maximally spaced order