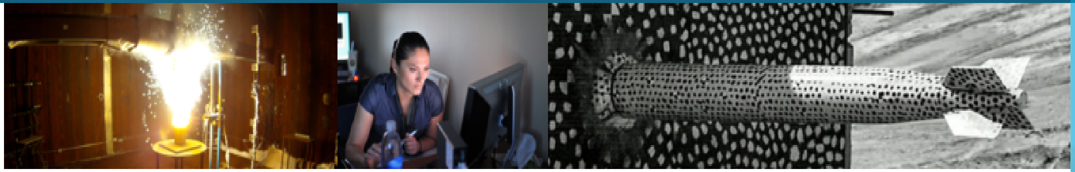
SAND2019-5203C

Group-Based Latin Hypercube Sample Construction for Improved Convergence and Cross-Validation Estimates



PRESENTED BY

Justin Winokur, PhD

Verification & Validation, Uncertainty Quantification, and Credibility Processes Dept

SAND2019XXXX UUR

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation

Expensive computational models with moderate-to-high dimensionality

Limited sampling budget

Unknown *a priori* convergence criteria make it difficult to budget model evaluations

Iterated LHS doubling provides a means to determine convergence with lower up-front investment

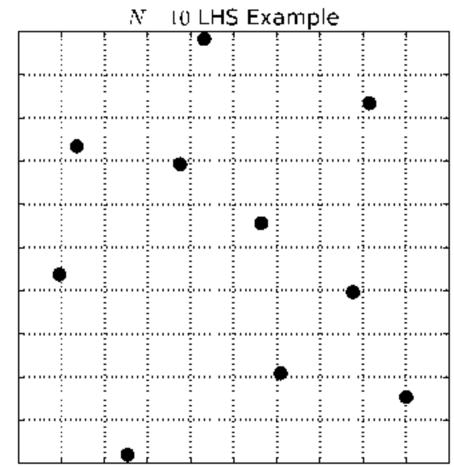
- Doubling sample size may be prohibitively expensive

Samples will be used for surrogate modeling. Best assessment of surrogate quality of via cross-validation

Goal: Develop a sampling scheme to provide for sample enrichment and more robust cross-validation estimates

Approach: Group-Based LHS with enrichment

Latin Hypercube Samples (LHS)

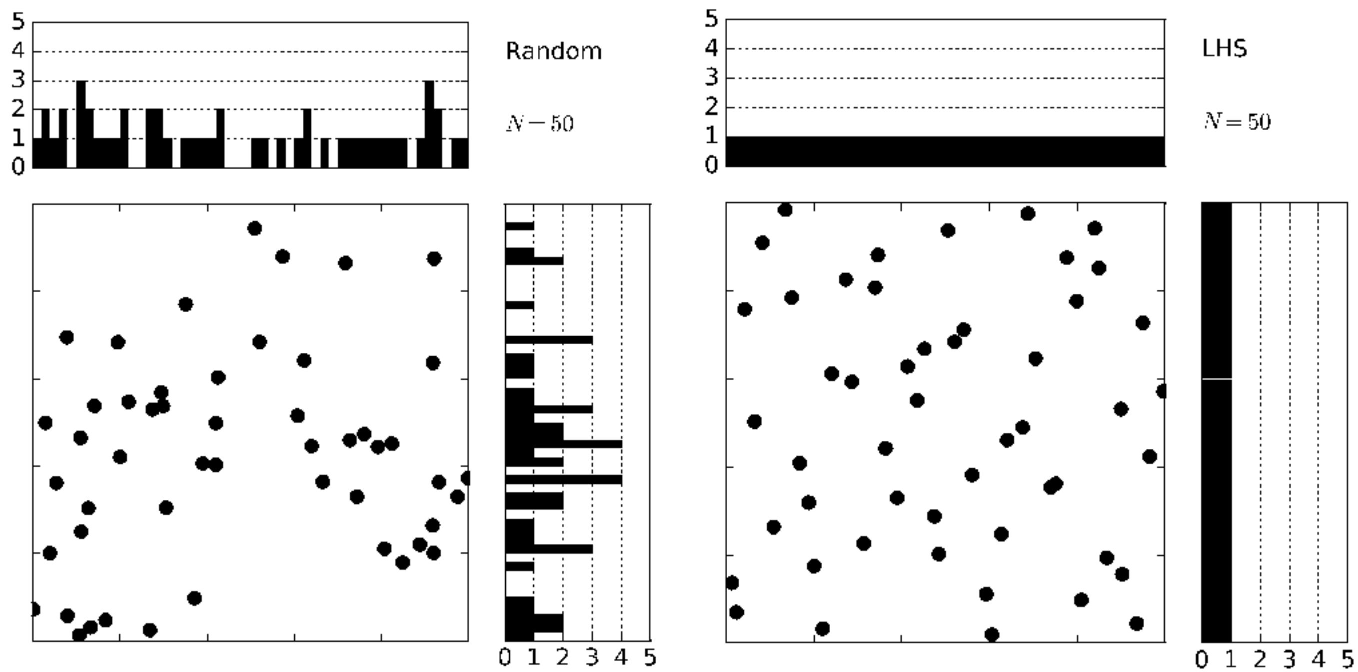


Latin Hypercube Samples provide robust, Monte-Carlo sample for design-of-experiments.

- No (asymptotic) dimensional dependence
- Moderate-to-good space-filling properties
- Non-collapsing marginal samples

Key idea: Divide space into N-hypercubes. Randomly place one sample per row/column/…

- Use inverse-CDF transform to map to other marginal densities
- Advanced methods to add additional correlations, etc



Building Latin Hypercube Samples

At its core, LHS designs are built on a per-dimension basis. Build from “ranks” of 1D permutations of {1,…,N}

```
for d in {1,...,n_d}
    Let R be a random permutation of {0,...,N-1}
    X_d = {(R + u~U[0,1])/N for i = {0,...,N-1} }
```

Commonly, LHS is optimized by taking a large number of realizations of X and choosing the one that minimizes a metric. Alternative Methods:

- Simulated Annealing – Stochastic discrete optimization
- “Improve Distributed” – Choose new samples that are the average separation distance away with a reduced set of candidates
- Translational propagation – Tile a smaller LHS with shifted bins
- “Latinized” Centroidal Voronoi Tessellation – Use CVT to jointly determine ranks then Latinize

LHS Metrics: (non-exhaustive)

$$\phi_{p,q} = \left(\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\frac{1}{d_{ij}^{(q)}} \right)^{p} \right)^{1/p}, \qquad d_{ij}^{(q)} = \left(\sum_{k=1}^{n_d} \left| x_i^{(k)} - x_j^{(k)} \right|^{q} \right)^{1/q}$$

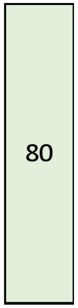
- Larger p values increase the importance on the closest samples.
- Maximin (Maximize the minimum distance) is $\lim_{p \to \infty} \phi_q$ (when minimized)

Enriched and Group-Enriched Samples

Traditional LHS samples cannot be decomposed into smaller groups while still maintaining the LHS property

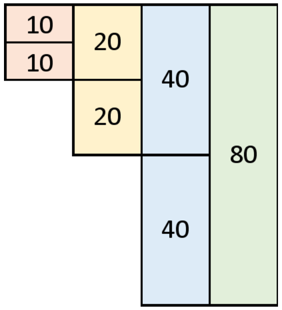
Default LHS samples cannot be decomposed

1x80



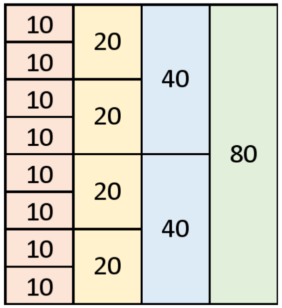
Starting at the smallest size, enriched samples can be decomposed only by half.

2x10, 2x20, 2x40, 1x80



Group enriched samples maintain the LHS property for all groupings starting at the initial size

8x10, 4x20, 2x40, 1x80



Grouped-Enriched LHS Construction: 1D

Construction of enriched LHS is performed independently in each dimension with an enrichment by a factor of two. Example: Consider two N=3 1D sample shuffled *ranks:*

$$R_a^{(0)} = \{1,0,2\} \qquad R_b^{(0)} = \{0,2,1\}$$

Generate offset vector $i_a = \{0,1,1\}$, a vector that is randomly 0 or 1. Let $i_b = 1 - i_a = \{1,0,0\}$

$$R_a^{(0*)} = 2R_a^{(0)} + i_a = \{2,1,5\}$$
$$R_b^{(0*)} = 2R_b^{(0)} + i_b = \{0,4,3\}$$
$$\Rightarrow \{2,1,5,0,4,3\}$$

Repeat the process again with N=3 to generate another N=6 grouped array and combine for N=12. Repeat again for N=24,…

Always begin with N=3 (smallest group)

	4 rank-groups of 3	2 rank-groups of 6	1 rank-group of 12
0	1A	2A	5A
1	0A	1A	2A
2	2A	5A	10A
3	0B	0B	1B
4	2B	4B	9B
5	1B	3B	6B
6	0C	1C	3C
7	2C	4C	8C
8	1C	3C	7C
9	2D	5D	11D
10	0D	0D	0D
11	1D	2D	4D

Randomness comes from original ordering at the smallest group and the offset vectors

Grouped-Enriched LHS Construction: D ≥ 2

Independently follow the the same process as 1D for each dimensions with new, randomly chosen offset vectors

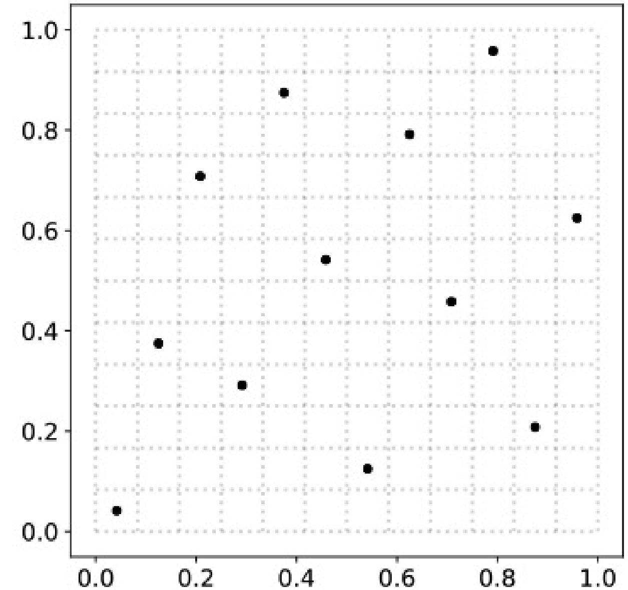
X_1

0	1A	2A	5A
1	0A	1A	2A
2	2A	5A	10A
3	0B	0B	1B
4	2B	4B	9B
5	1B	3B	6B
6	0C	1C	3C
7	2C	4C	8C
8	1C	3C	7C
9	2D	5D	11D
10	0D	0D	0D
11	1D	2D	4D

X_2

0	1A	3A	6A
1	2A	4A	8A
2	0A	1	2A
3	1B	2B	4B
4	2B	5B	11B
5	0B	0B	1B
6	0C	1C	3C
7	1C	2C	5C
8	2C	4C	9C
9	1D	3D	7D
10	0D	0D	0D
11	2D	5D	10D

	X_1	X_2
0	5A	6A
1	2A	8A
2	10A	2A
3	1B	4B
4	9B	11B
5	6B	1B
6	3C	3C
7	8C	5C
8	7C	9C
9	11D	7D
10	0D	0D
11	4D	10D



The is the same for *any* number of dimensions

The final sample can again be doubled with another enriched-doubled sample

Random optimization on the final and/or subsamples can still be performed

Correlations should be added at the *smallest group* size

K-Fold Cross Validation

Robust, black-box method to assess surrogate performance

Many surrogates are exact interpolants so you can't assess error based on the build points. Alternative: Cross-Validation

K-Fold Cross Validation:

- Make K-partitions of the data. Build a new surrogate (S) on training data ($x_{train}^{(k)}$) and evaluate on validation data ($x_{val}^{(k)}$)

$$\epsilon_{cv}^2 = \frac{1}{N}\sum_{k=1}^{K}\left[\sum_{j\in x_{val}^{(k)}}\left(f_j - S_{\{x_{train}^{(k)}\}}(x_j)\right)^2\right]$$

- Relative CV: $\epsilon_{cv,rel} = \sqrt{\frac{\epsilon_{cv}^2}{\sum_{i=1}^{N} f_i^2}}$
- Be careful with notation. Some articles refer to ϵ_{cv}^2 as the error and not the squared error.
- Common K Values
  - 5 - 10 (Reasonable cost)
  - K = N, Leave-One-Out Cross Validation [LOO] (Very Expensive)
- Can use tricks to speed up general K-fold or approximate LOO. Or use brute-force on end-to-end construction for most robust estimate

k=1	k=2	k=3	k=4	k=5
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20

	Training
	Validation

Design should be shuffled in the general case. Not shuffled for group-based LHS

Surrogate Form: Radial Thin-Plate Splines

Ideally, would use Kriging surrogate but they are require $O(n_d)$ dimensional non-convex optimization

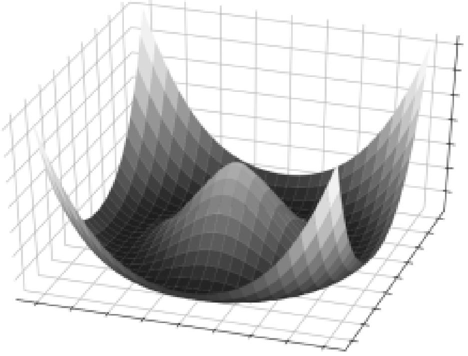
$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{matrix} \mu(\boldsymbol{x}) \\ \mu(\boldsymbol{x}^*) \end{matrix}, \begin{bmatrix} K(\boldsymbol{x},\boldsymbol{x}) + \sigma_n^2 I_N & K(\boldsymbol{x},\boldsymbol{x}^*) \\ K(\boldsymbol{x}^*,\boldsymbol{x}) & K(\boldsymbol{x}^*,\boldsymbol{x}^*) \end{bmatrix} \right)$$

Alternative surrogate is to use Polyharmonic Radial Thin-Plate Splines

$$f(x) = \sum_{j=1}^{N} c_j \Phi(r_j) + \sum_{k \in \mathcal{P}} g_k p_k(x)$$

Where $\Phi(r) = \mathrm{r}^2 \log(\mathrm{r})$ and $\mathrm{r_j} = \left\| x_j - x \right\|_2$. $\mathcal{P}$ is a polynomial index (usually linear terms) and $p_k(x) = \prod_{d=1}^{n_d} x^{\mathcal{P}_d^k}$ (a multi-dimensional). Solve by enforcing $f(x_j) = f_j$ and $\sum_{j=1}^{N} c_j p_k(x_j) = 0 \forall k \in \mathcal{P}$

2D Radial Thin-Plate



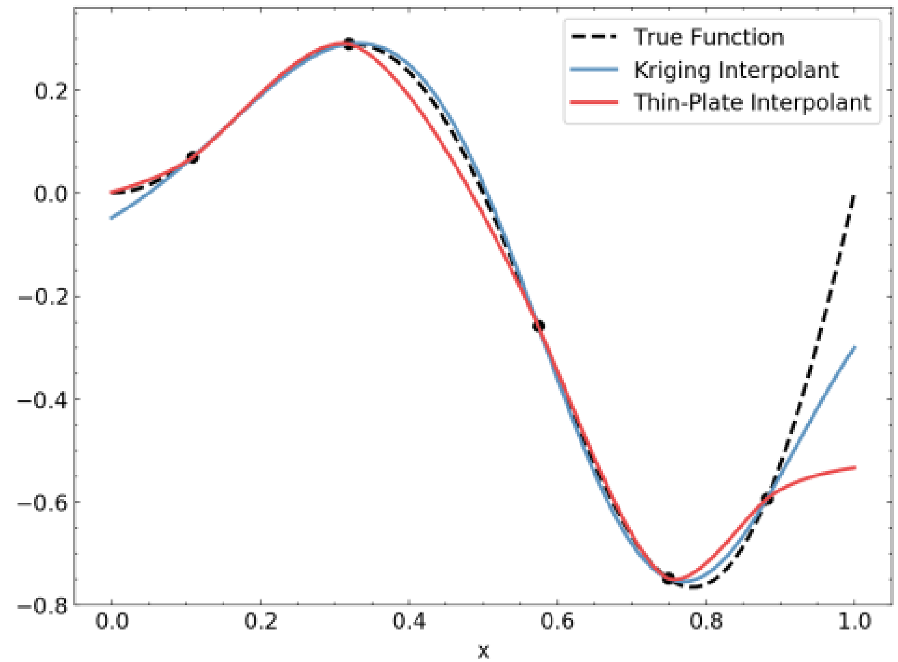
$$\left[\begin{array}{c|c} \Phi(r_{ij}) + \sigma_n^2 I_N & p_k(x_j) \\ \hline p_k(x_j)' & 0 \end{array}\right] \left[\begin{array}{c} c \\ \hline g \end{array}\right] = \left[\begin{array}{c} f \\ \hline 0 \end{array}\right]$$

Surrogate Form: Radial Thin-Plate Splines

Radial splines:

- **completely non-parametric** unlike Kriging (correlation lengths) and polynomials (polynomial basis and coefficients).
- dimensionally independent complexity
- Require only a single matrix N^2 matrix inversion is needed for each experimental design
- Can include data noise (smoothing) but do not provide a statistical basis to determine the appropriate value (unlike Kriging)
- Generally less accurate than Kriging
  - Especially for extrapolation

In practice we would use Kriging but it is too expensive for the repeated runs used in this study



Initial Studies & Results

Study Outline

These are preliminary results

Goals:

- Better sampling on the divisions as compared to a non-grouped LHS
- More accurate cross-validation estimates

Tests

- 3 enrichments for k=8 fold with grouped LHS
  - 2D, 8D, varied dimensions
- Non-LHS enrichment from LHS groups (LHS metrics only)

2D Varying Size

Test Problem

$$f(x,y) = \tanh(2x(2y-1)) + \frac{\cos(\pi(2y-1)) + 5}{e^{-40x+10} + 1}$$

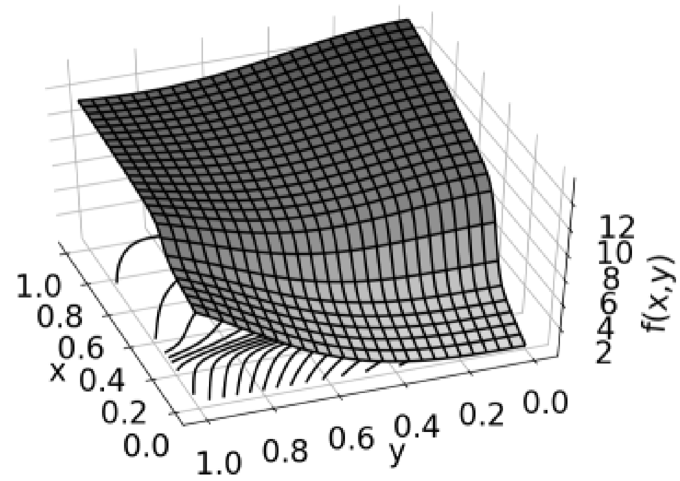
$$x, y \in [0,1]$$

Build,

N0 → N1 = 2*N0 →

N2 = 2*N1 = 4*N0 →

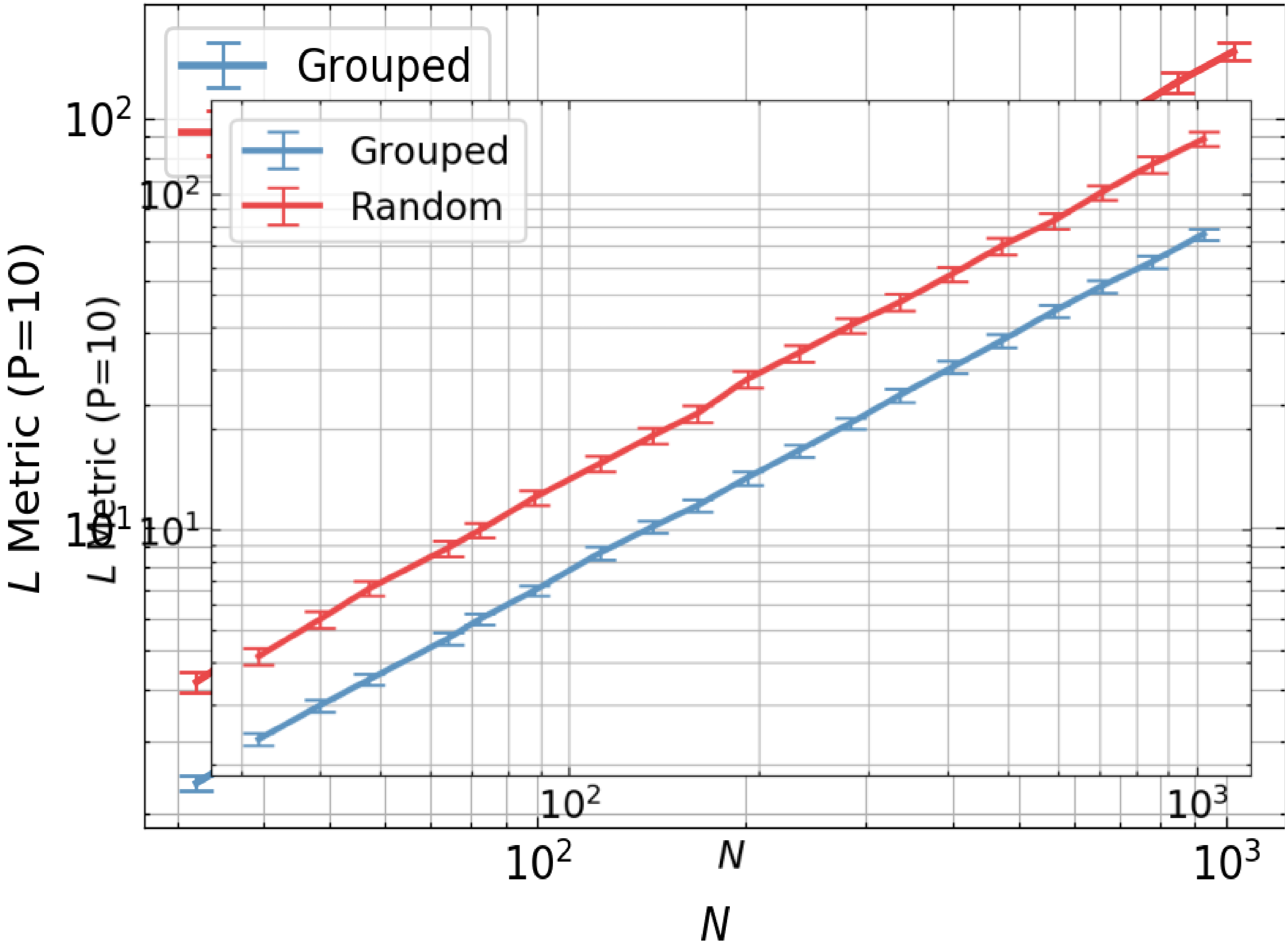
N3 = 2*N2 = 4*N1 = 8*N0



LHS ensemble

- N0 from 4 to 128
- Build to with random optimization over 50 iterations with $\phi_{10,2}$
- For each N0 and trial compute CV error with Grouped LHS and Regular LHS with 8*N0
  - Random shuffle of regular LHS
- Repeat for 150 trials
- Estimate “True” Error with 250,000 LHS (simple) sample

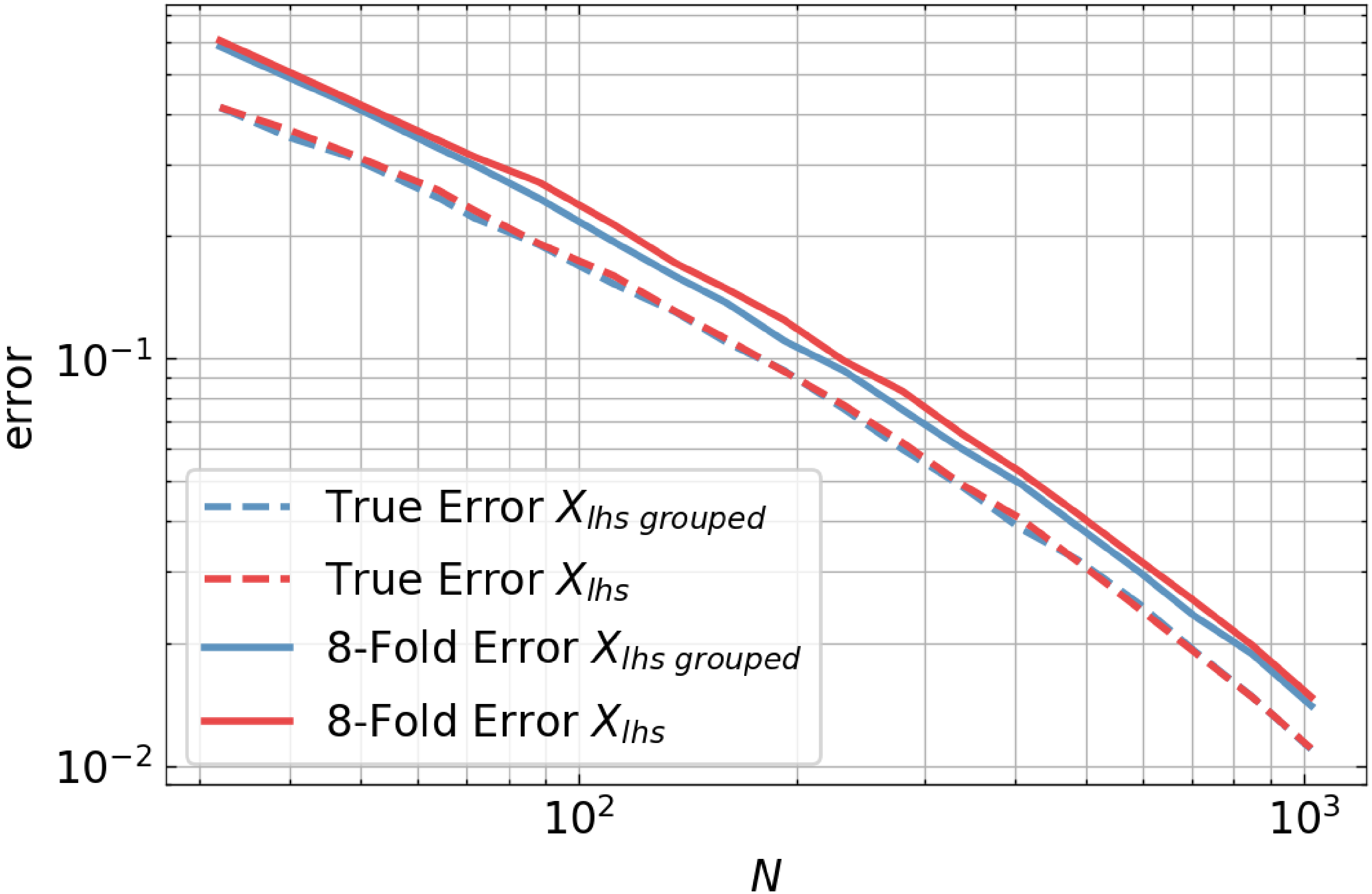
2D Varying Size: LHS Metrics of Divisions



Grouped
L Metric (P=10)
10^1
10^2
10^3
N

2D Varying Size: CV Error Estimate

Use 8-Fold Cross-Validation. Compare to “True” error



Grouped LHS error is consistently less than random LHS error and closer to ”True” error

8D Varying Size

8D: Borehole Function

$$f(r_w, r, T_u, H_u, T_l, H_l, L, K_w) = \frac{2\pi T_u (H_u - H_l)}{\log(r/r_w)\left(1 + \frac{2LT_u}{\log(r/r_w)\, r_w^2 K_w} + \frac{T_u}{T_l}\right)}$$

Build,

N0 → N1 = 2*N0 →

N2 = 2*N1 = 4*N0 →

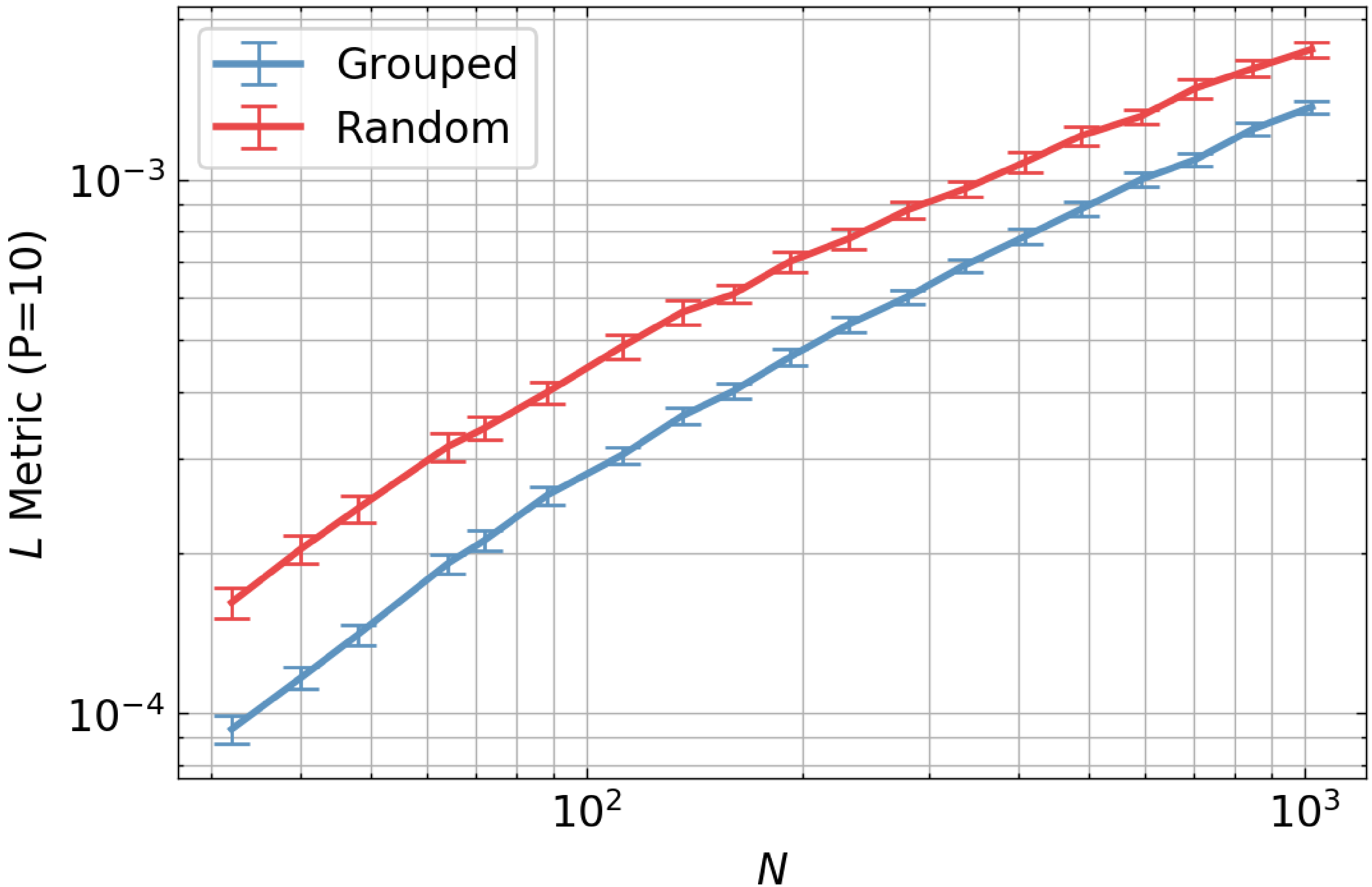
N3 = 2*N2 = 4*N1 = 8*N0

Name	Distribution
	N(μ=0.10, σ=0.0161812)
	Lognormal(μ=7.71, σ=1.0056)
	Uniform[63070, 115600]
	Uniform[990, 1110]
	Uniform[63.1, 116]
	Uniform[700, 820]
	Uniform[1120, 1680]
	Uniform[9855, 12045]

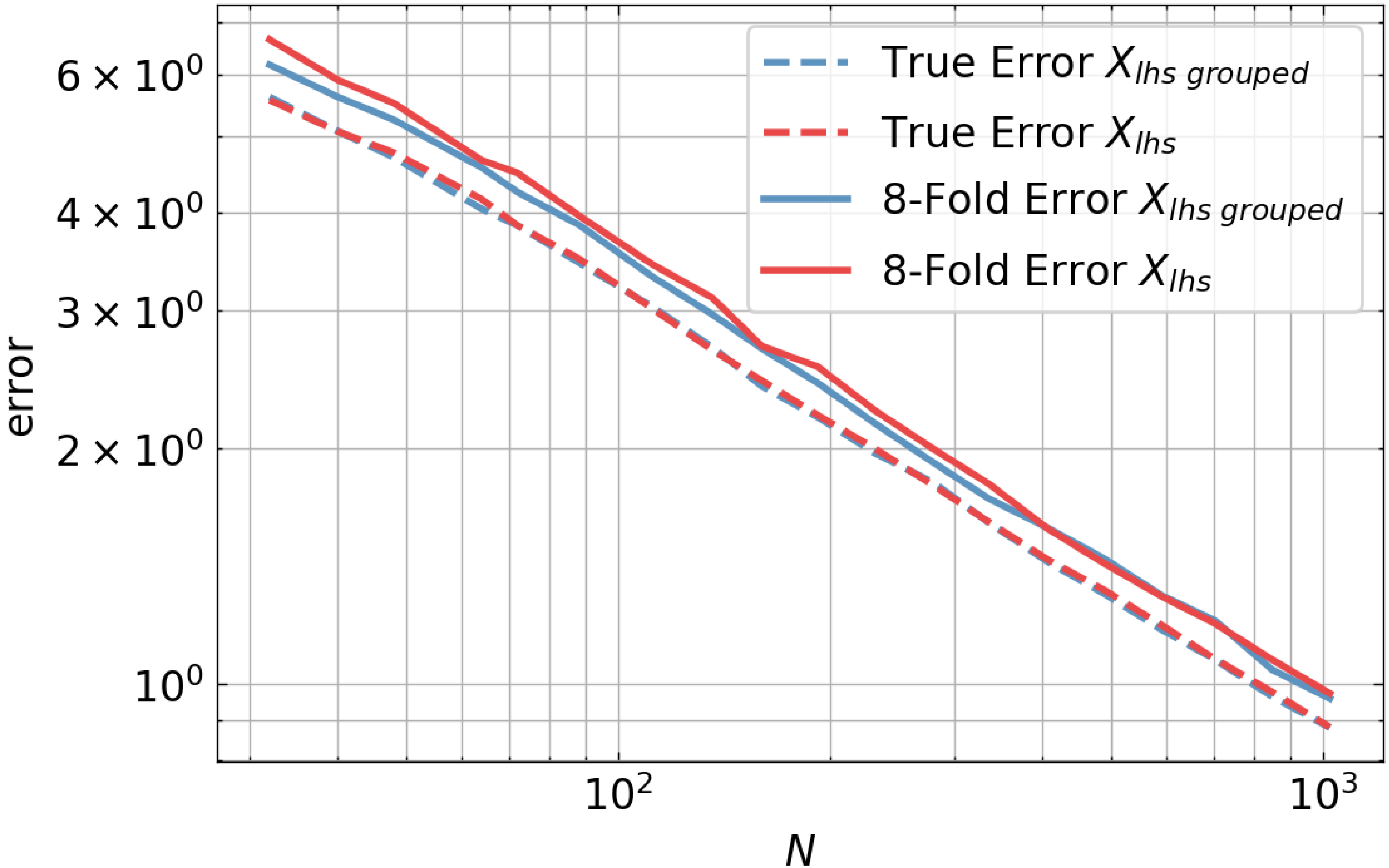
LHS ensemble

- N0 from 4 to 128
- Build to with random optimization over 50 iterations with $\phi_{10,2}$
- For each N0 and trial compute CV error with Grouped LHS and Regular LHS with 8*N0
  - Random shuffle of regular LHS
- Repeat for 150 trials
- Estimate "True" Error with 250,000 LHS (simple) sample

8D Varying Size: LHS Metrics



8D Varying Size: Errors



Less profound improvement in the error but grouped LHS was still more accurate than random LHS

Fixed Sample Size, Varying Dimension

Any Dimension

$$f(\boldsymbol{x}) = \frac{1}{\frac{1}{10} + \sum_{d=1}^{n_d} \left(x_d - \frac{1}{2}\right)^2} \quad , \qquad x_d \in [0,1]$$

Build

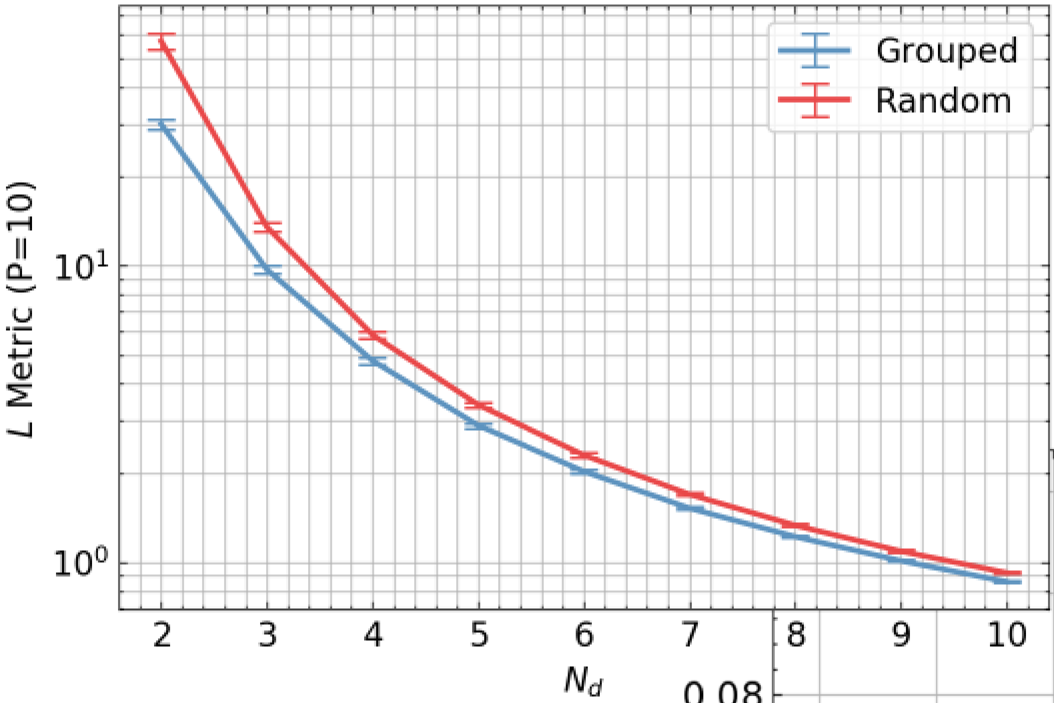
- Same 8-Fold construction with N0 = 50 and N3 = 400
- Build with random optimization over 50 iterations with $\phi_{10,2}$
- For each N0 and trial compute CV error with Grouped LHS and Regular LHS with 8*N0
  - Random shuffle of regular LHS
- Repeat for 150 trials
- Estimate "True" Error with 250,000 LHS (simple) sample
- 2 to 10 dimensions

Variance Based Decomposition (Sobol Indices) values for groups of dimensions. Problem is isotropic

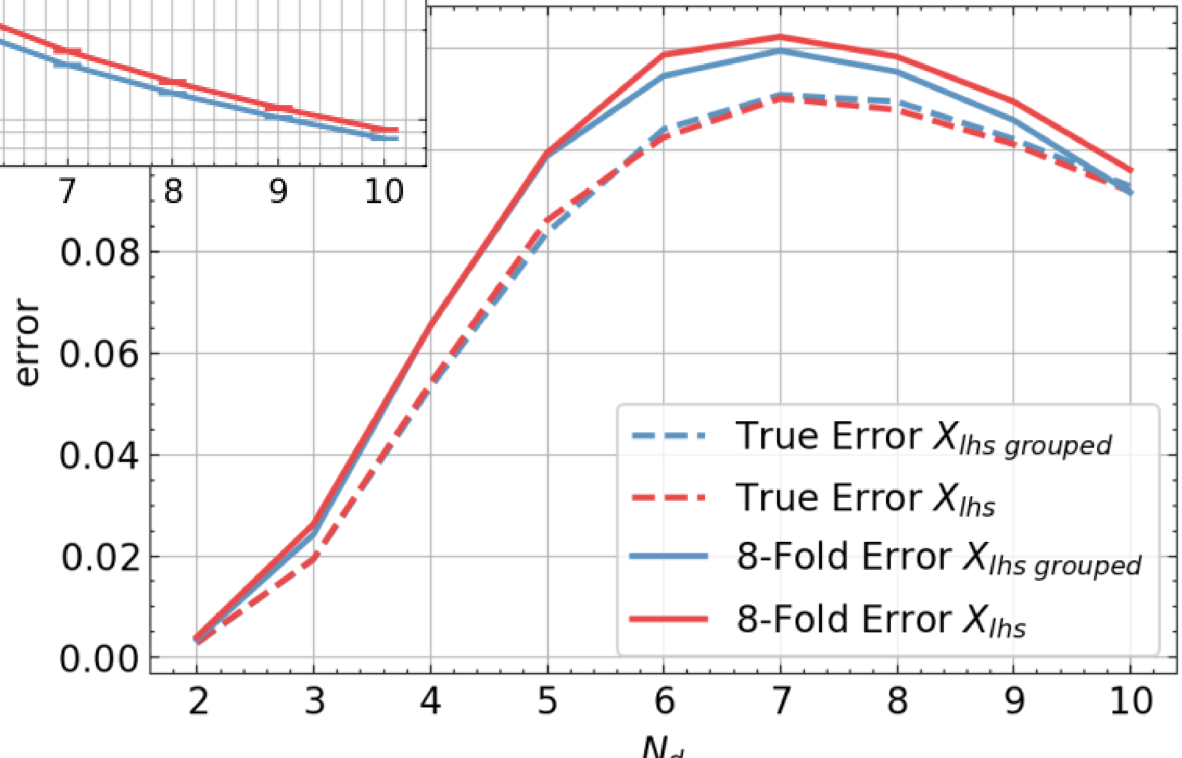
Any Grouping of # dimensions

Dimensions	1	2	3	4	5	6	7	8	9	10
1	1									
2	0.4	0.2								
3	0.1	0.1	0.2							
4	0.08	0.05	0.08	0.07						
5	0.05	0.02	0.04	0.03	0.03					
6	0.03	0.01	0.02	0.01	0.01	0.01				
7	0.03	0.006	0.008	0.007	0.006	0.005	0.005			
8	0.02	0.003	0.004	0.003	0.003	0.003	0.002	0.002		
9	0.02	0.003	0.002	0.002	0.001	0.001	0.001	0.001	0.0009	
10	0.02	0.002	0.001	0.0008	0.0006	0.0005	0.0005	0.0004	0.0004	0.0004

Fixed Sample Size, Varying Dimension



Grouped LHS maintains a lower LHS metric value as dimensions scale



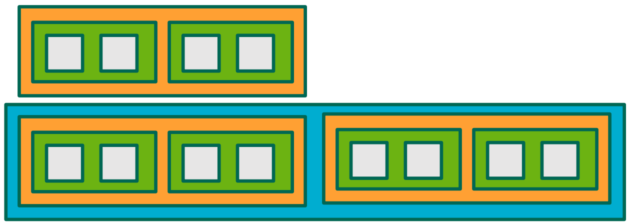
CV Errors are similar for low dimensions (likely due to having 400 samples) but as they decay, the Grouped LHS CV Error reduces and is more representative of the True error

Non-LHS Enrichment

In practice, LHS doubling is prohibitive; grouped or traditional.

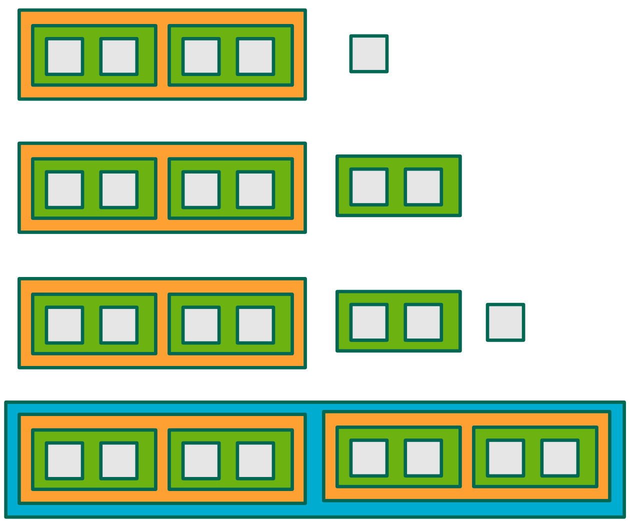
Grouped LHS enrichment can be used to enrich the sample even if the final result is not a true LHS. If all groups are added, final sample retains LHS properties

Goal: Better sampling; not necessarily LHS sampling



Consider a grouped-4 enriched LHS

Performed grouped enrichment



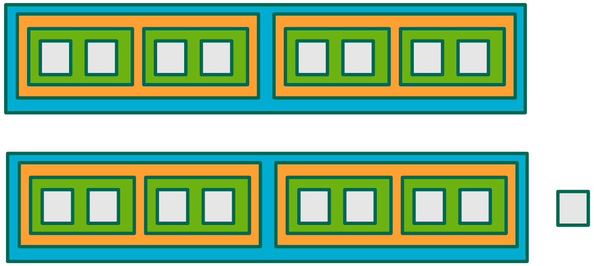
Successively add the next groups (N_0 samples)

These enrichments will *not* be an LHS but will maintain good sampling properties

Final enrichment is fully LHS

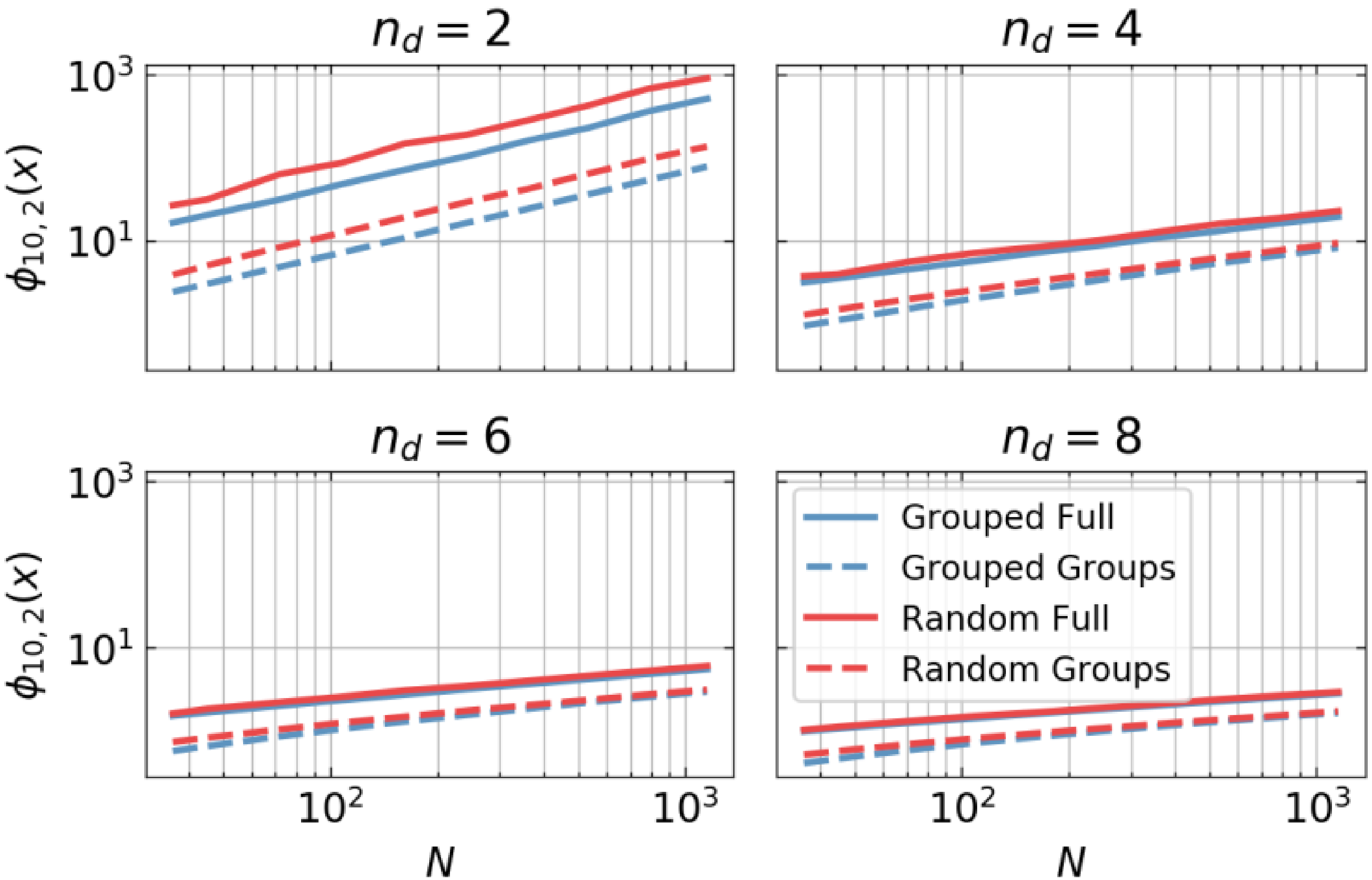
Only tested on LHS metrics; not CV performance

Single Enrichment, Vary N, n_d



Examine how LHS metrics change as a single additional group is added.

- 150 Trials
- 9 total groups

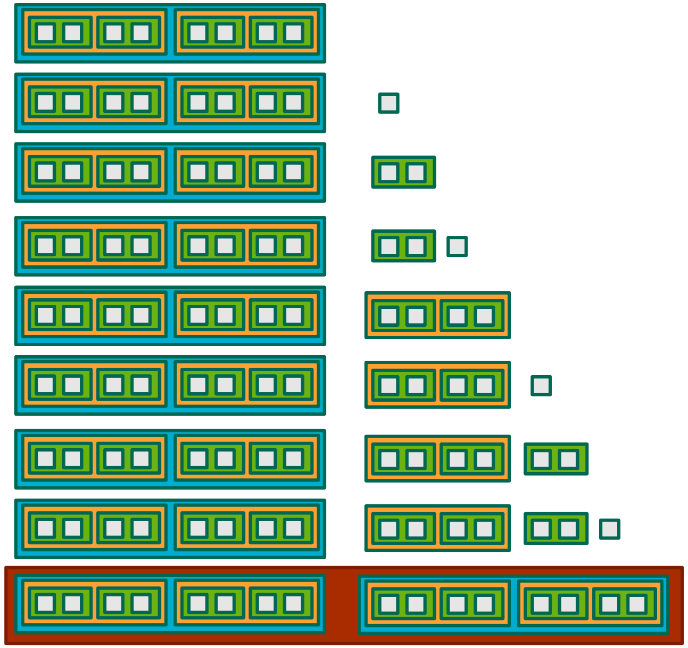


Adding Groups: Metrics

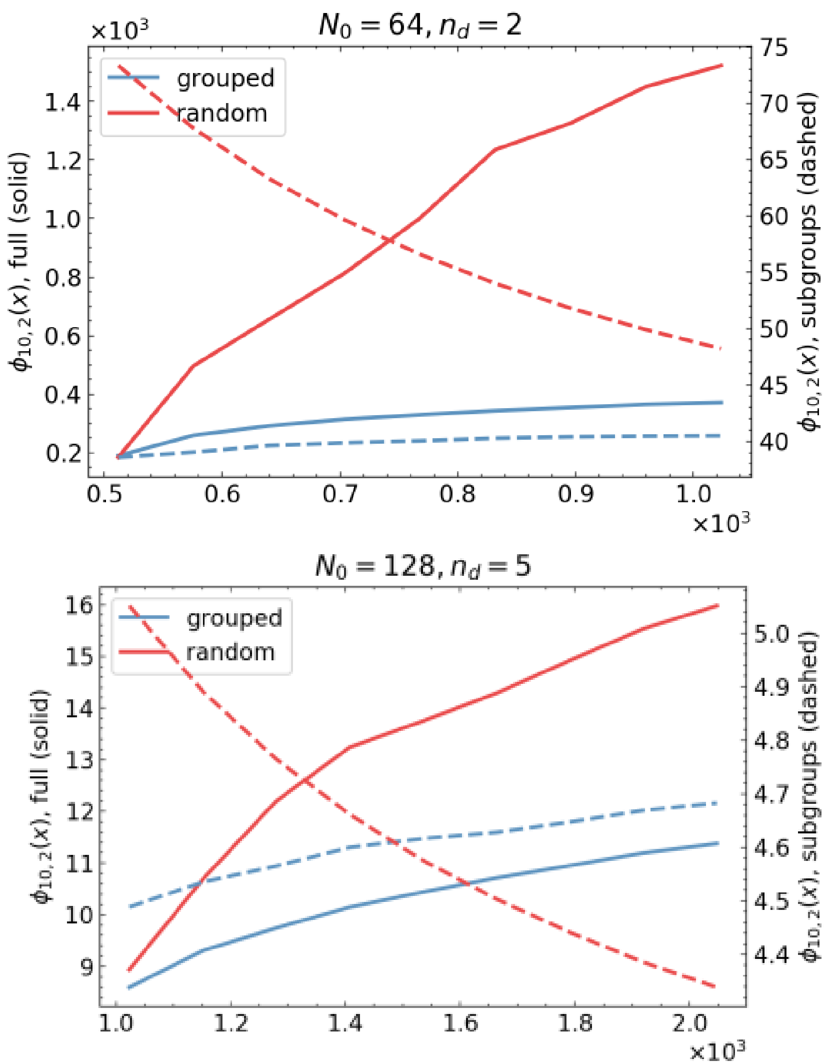
Again consider adding groups from a larger enrichment compared to adding LHS samples:

- Grouped: Generate 8-group LHS sample. Enrich to 16 groups but add one group at a time
- Regular: Generate 1 large LHS sample. Add smaller samples one at a time

Compare metrics on full sample and each group with 150 trials



Adding Groups: Metrics



Enriching a sample with future groups *dramatically* improves LHS sample quality while providing a linear-sized, N_0 enrichment

Full Sample:

- Random LHS and enrichment continues to be a poor sample since more LHSs are being concatenated
- Despite *not* maintaining the LHS property, group enrichment's metric grows only due to increased N
- Note, ϕ will *naturally* grow for the full sample as N increases

Subgroups:

- Random improves since only smaller LHSs are added to the full sample and those smaller LHSs are made without regard to the overall LHS design
- Grouped LHS suffers since it must maintain full LHS property
  - Trivial in comparison to full-sample improvement

Conclusions

Grouped Enriched LHS provides a framework to iteratively construct LHS that provide for better subdivided samples

Clear improvement over random LHS for sample quality metrics

Cross-Validation provided a more accurate result, especially for small sample sizes

For these tests, a better sample did not result in a significantly higher quality surrogate

- Make be surrogate dependent (Full Kriging surrogates may perform better)
- Less-smooth test problems may be more susceptible to better

Alternatively, results suggest that improving sample quality may have diminishing returns

Future Work:

- Examine CV performance with enriched samples
- Test with less-smooth test functions more sensitive to sample quality

Backup Material

Saltelli Method for Global Sensitivity Analysis

Sobol Global Sensitivity Analysis requires solving $\mathbb{V}\left(\mathbb{E}(f|x_{\{d\}})\right)$ which would directly require a nested loop (conditional expectation and variance). The Saltelli algorithm simplifies this:

- Evenly divide an LHS into two samples (or take two samples), $X^{(1)}, X^{(2)}$.
- Let $X^{\{d\}}$ Be built from $X^{(2)}$ with dimensions (columns) $\{d\}$ be from $X^{(1)}$
- $\mathbb{E}(f) = \frac{1}{N}\sum_{i=1}^{N} f(X_i^{(1)})$
- $\mathbb{V}\left(\mathbb{E}(f|x_{\{d\}})\right) = \frac{1}{N-1}\sum_{i=1}^{N}\left(f\left(X_i^{\{d\}}\right) - \mathbb{E}(f)\right)\left(f\left(X_i^{(1)}\right) - \mathbb{E}(f)\right)$

(1)

0.02	0.34	0.22
0.95	0.51	0.49
…	…	…
0.52	0.48	0.87

(2)

0.03	0.65	0.74
0.45	0.86	0.85
…	…	…
0.10	0.05	0.09

$X^{\{1\}}$

0.02	0.65	0.74
0.95	0.86	0.85
…	…	…
0.52	0.05	0.09

$X^{\{2,3\}}$

0.03	0.34	0.22
0.45	0.51	0.49
…	…	…
0.10	0.48	0.87

$X^{\{2\}}$

0.03	0.34	0.74
0.45	0.51	0.85
…	…	…
0.10	0.48	0.09

$X^{\{1,3\}}$

0.02	0.65	0.22
0.95	0.86	0.49
…	…	…
0.52	0.05	0.87

$X^{\{3\}}$

0.03	0.65	0.22
0.45	0.86	0.49
…	…	…
0.10	0.05	0.87

$X^{\{1,2\}}$

0.02	0.34	0.74
0.95	0.51	0.85
…	…	…
0.52	0.48	0.09

When $X^{(1)}, X^{(2)}$ are from splitting an enriched LHS sample, all $X^{\{d\}}$ are *still* LHS samples even with the swapped columns