Constrained Sampling

A Study on Methods for Sampling from Constraint Manifolds

February, 25th 2022
Philip Oedi
340896

Prof. Marc Toussaint
Prof. Jörg Krüger
Ph.D. Jung-Su Ha
M.Sc. Joaquim Ortiz-Haro

Technische Universität Berlin
Learning and Intelligent Systems

Outline

1. Problem
2. Research Questions
3. Methods
4. Experiments
    1. Results
    2. Conclusions

Problem - Definition

Find a set $S \subset \mathcal{X}$ for $\mathcal{X} = \{x \in \mathbb{R}^n : g(x) \leq 0,\ h(x) = 0\}$ that solves the following constrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) \text{ s.t. } g(x) \leq 0,\ h(x) = 0$$

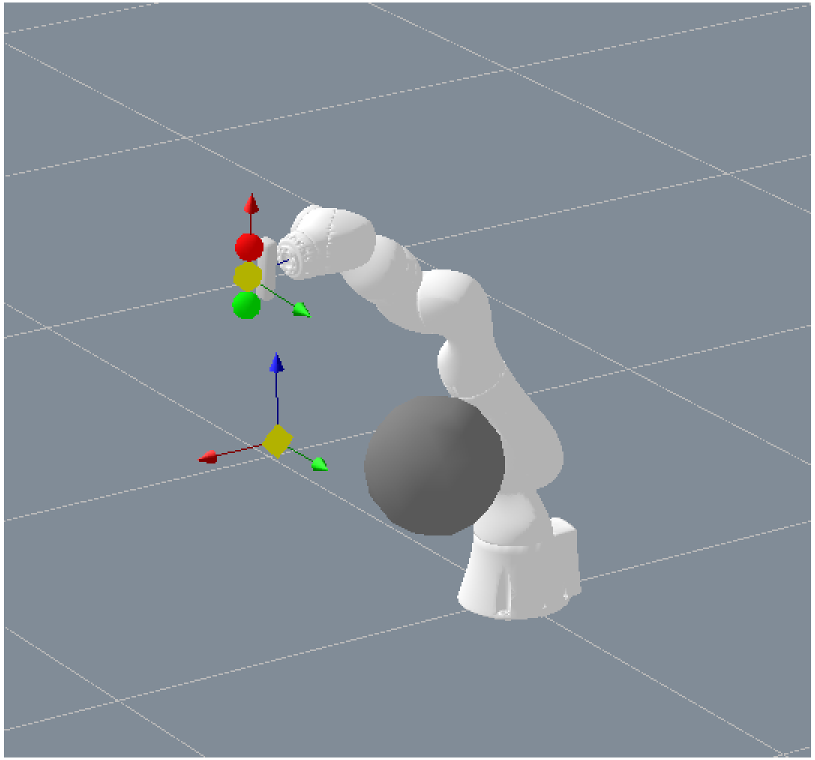
with $f = 0$. S is the set of samples and $\mathcal{X}$ the feasible set. The functions $g(x)$ are inequality constraints and $h(x)$ equality constraints.

Requirements

- We require that S
  - is diverse (Uniformity)
  - Covers $\mathcal{X}$ well (Coverage)
  - Generated with reasonable computational effort (Computational Efficiency)

Problem - Application

- Application in Robotics
- Sampling Goal States in Motion Planning
- Feasible Goal States are on a Constraint Manifold



Problem – Approaches

- Standard approaches like *rejection sampling fail*
- The manifold can be *disconnected* and *arbitrary shape*

- Projection
  - Project infeasible points onto the manifold by solving an optimization problem

\+ *“Walking of the Tangent Space”*

Research Purpose

- Study algorithms for sampling from constraint manifolds and the resulting sampling sets
  - How are the sampling sets distributed?
  - What are causes for non-uniformity and lack in coverage?
- Improve the algorithms
  - Can we add functionality to improve uniformity and coverage?

Methods

1. Biased Optimization
2. i.i.d.-Biased-Samplers
3. Grid-Walk-On-Tangent
4. RRT-On-Tangent
5. Composite Samplers

Biased Optimization

Find feasible x that minimizes the distance to a bias x_{bias} by solving:

$$\min_{x \in \mathbb{R}^n} ||x - x_{\mathrm{bias}}||^2 \text{ s.t. } g(x) \leq 0, \;\; h(x) = 0$$

Choose x_{bias} as the initial guess/seed.

Biased Optimization

Algorithm 7 i.i.d. Biased Sampler

Input: $n \in \mathbf{N},\ b_{\text{low}},\ b_{\text{up}},\ g(x),\ h(x),$

$i \leftarrow 1$
while $i \leq n$ **do**
  $x_{\text{bias}} \sim U(b_{\text{low}},\ b_{\text{up}})$
  $x_i \leftarrow BIASED\_OPTIMIZATION(x_{\text{bias}},\ g(x),\ h(x))$
  $i \leftarrow i + 1$
end while

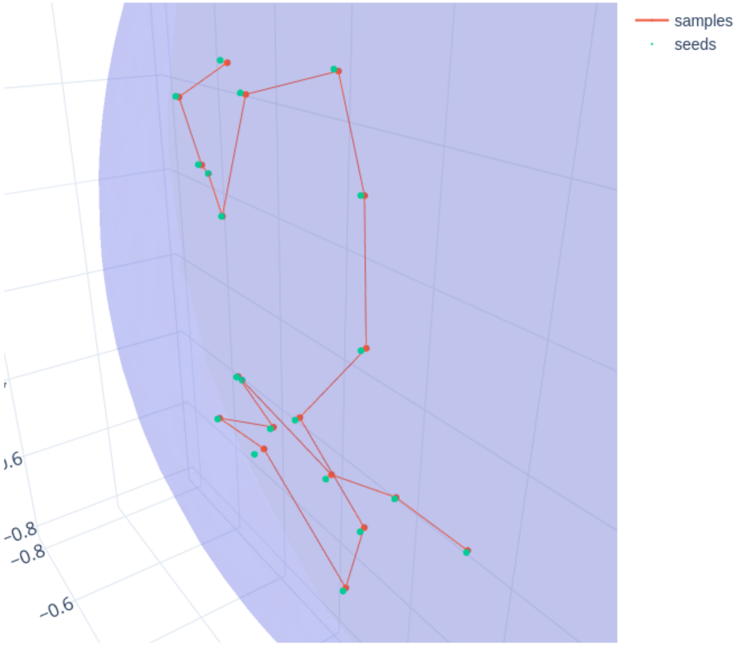
Grid-Walk-on-Tangent

- A random walk over the feasible set

Algorithm 8 Grid Walk on the Tangent Space

Input: $n \in \mathbf{N},\ b_{\text{low}},\ b_{\text{up}},\ g(x),\ h(x),\ w_{\text{GW}}$

$i \leftarrow 0$
$x_{\text{rand}} \sim U(b_{\text{low}},\ b_{\text{up}})$
$x_0 \leftarrow BIASED\_OPTIMIZATION(x_{\text{rand}},\ g(x),\ h(x))$
$i \leftarrow i + 1$
while $i < n$ **do**
    $\mathbf{\Theta} \leftarrow FIND\_TANGENT(x_{i-1}, h(x))$
    $x_{\text{rand}} \leftarrow SAMPLE\_TANGENT(x_{i-1}, \mathbf{\Theta}, w_{\text{GW}})$
    $x_i \leftarrow BIASED\_OPTIMIZATION(x_{\text{rand}},\ g(x),\ h(x))$
    $i \leftarrow i + 1$
end while



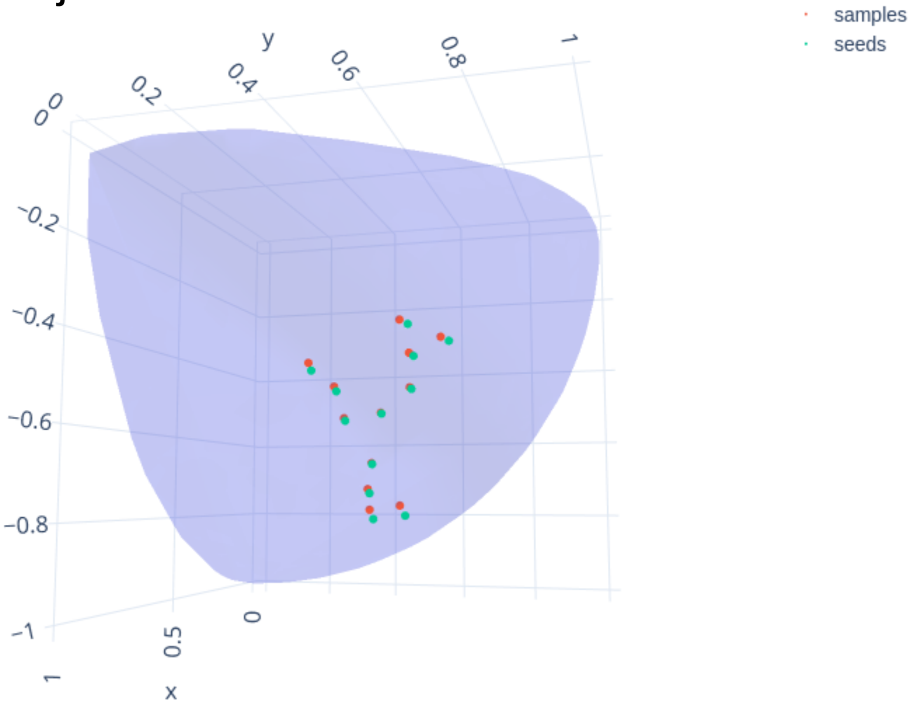
RRT-on-Tangent

- Build a RRT on the tangent space and project it onto manifold

Algorithm 9 RRT on the Tangent Space

Input: $n \in \mathbf{N},\ b_{\text{low}},\ b_{\text{up}},\ g(x),\ h(x),\ w_{\text{RRT}},\ G,\ \alpha,\ x_0$

$i \leftarrow 0$
$x_{\text{rand}} \sim U(b_{\text{low}},\ b_{\text{up}})$
$x_i \leftarrow BIASED\_OPTIMIZATION(x_{\text{rand}},\ g(x),\ h(x))$
$G.init(x_0)$
$\mathbf{\Theta} \leftarrow FIND\_TANGENT(x_0, h(x))$
$i \leftarrow i + 1$
while $i < n$ **do**
  $x_{\text{rand}}^{\text{tangent}} \leftarrow SAMPLE\_TANGENT(x_0, \mathbf{\Theta}, w_{\text{RRT}})$
  $x_{\text{nearest}}^{\text{tangent}} \leftarrow NEAREST\_VERTEX(x_{\text{rand}}^{\text{tangent}}, G)$
  $x_i^{\text{tangent}} \leftarrow NEW\_CONF(x_{\text{nearest}}^{\text{tangent}}, \alpha)$
  $G.add\_vertex(x_i^{\text{tangent}})$
  $G.add\_edge(x_i^{\text{tangent}}, x_{\text{nearest}}^{\text{tangent}})$
  $x_{\text{bias}} \leftarrow x_i^{\text{tangent}}$
  $x_i \leftarrow BIASED\_OPTIMIZATION(x_{\text{bias}},\ g(x),\ h(x))$
  $i \leftarrow i + 1$
end while



Composite Sampler

1. Global Stage

2. Local Stage

i.i.d-Biased-Sampler

GW/RRT-on-Tangent
GW/RRT-on-Tangent
GW/RRT-on-Tangent
GW/RRT-on-Tangent
GW/RRT-on-Tangent
GW/RRT-on-Tangent

S

Use samples generated in global stage as starting points for local stage

Experiments

1. 2-d Manifold (sphere) embedded in 3-d
    1. Center-Connected
    2. Off-Center-Connected
    3. Off-Center-Disconnected
2. Robotics Examples
    1. 3-DOF
    2. 7-DOF

From Requirements to Metrics

- We require that S
  - is diverse (Uniformity)
    - *Variance* Var
    - *Entropy* $\hat{H}$
  - Covers $\mathcal{X}$ well (Coverage)
    - *Average Distance to Sample Set* (ADTS)
  - Generated with reasonable computational effort (Computational Efficiency)
    - *Average Iterations per Sample* (AIPS)

Uniformity - Entropy

- Use KDE to get an estimate of the probability density
- Estimate the entropy:

$$\hat{H}(X) = -\frac{1}{n}\sum_{i=1}^{n} \ln\ \hat{f}_{n,i}(x_i)$$

$\hat{f}_{n,i}$ - KDE at location *i* trained on all samples except *i*-th

Uniformity - Variance

- If samples are distributed uniform
  - Variance of KDEs at sample locations should be small
- Evaluate $\mathrm{Var}(Y)$ over a set of KDEs

$$Y = \{y_i \in \mathbb{R} : \hat{f}(x_i) = y_i \text{ for all } x_i \in S \;\; i \in [1; n_{\text{samples}}]\}$$

Coverage

Average Distance to Sample set (ADTS)

$$\mathrm{ADTS}(R, S) = \sum_{i=1}^{n_{\mathrm{ref}}} d(x_{\mathrm{ref},i})$$

$$d(x_{\mathrm{ref},i}) = \min_j ||x_{\mathrm{ref},i} - x_{s,j}||$$

With a uniform Reference Set $R = \{x_{\mathrm{ref},i} \in \mathcal{X} : i \in \{1, ..., n_{\mathrm{ref}}\}\}$.

(In our experiments Reference are uniform samples over a sphere.)

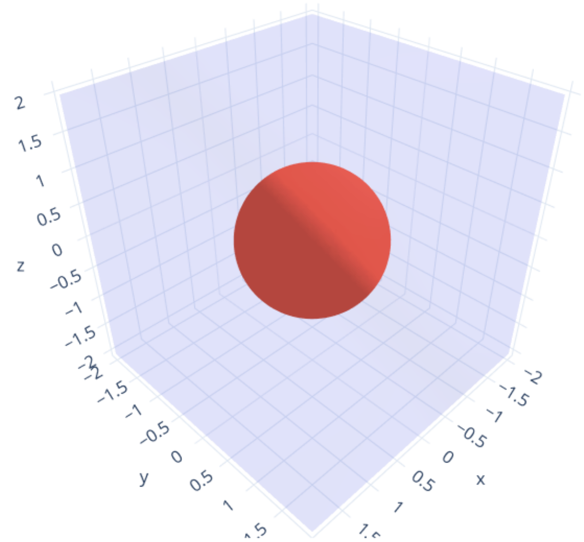
Computational Efficiency

- Solving Biased Optimization problem as main computational effort
- #iterations to solve a problem as a proxy for computational effort
- Average Iterations per Sampler (AIPS)

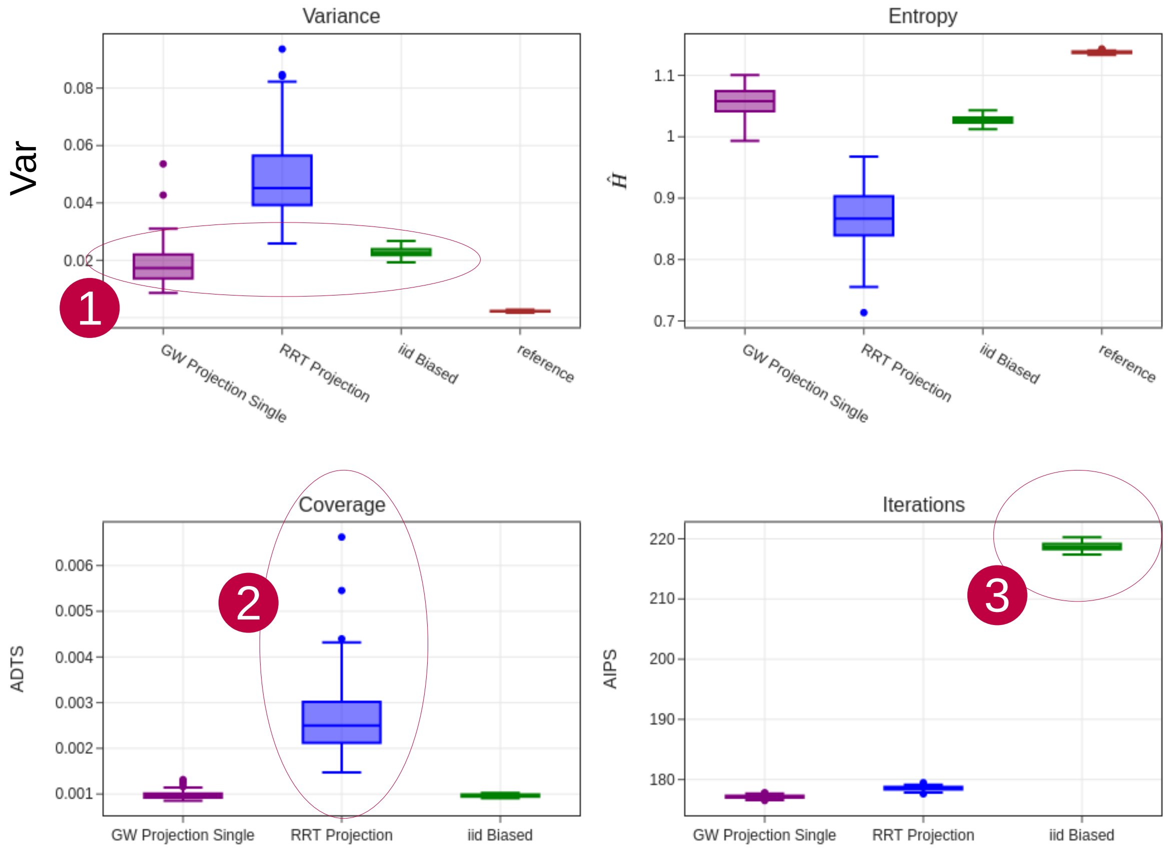
$$\mathrm{AIPS} = \frac{1}{n_{\mathrm{samples}}} \sum_{i=1}^{n_{\mathrm{iter},i}} n_{\mathrm{iter},i}$$

Center-Connected

- Algorithms
  - i.i.d-Biased-Sampler
  - RRT with multiple seeds (composite sampler)
  - Grid-Walk with a single seed
- *100x5000 samples per algorithm*



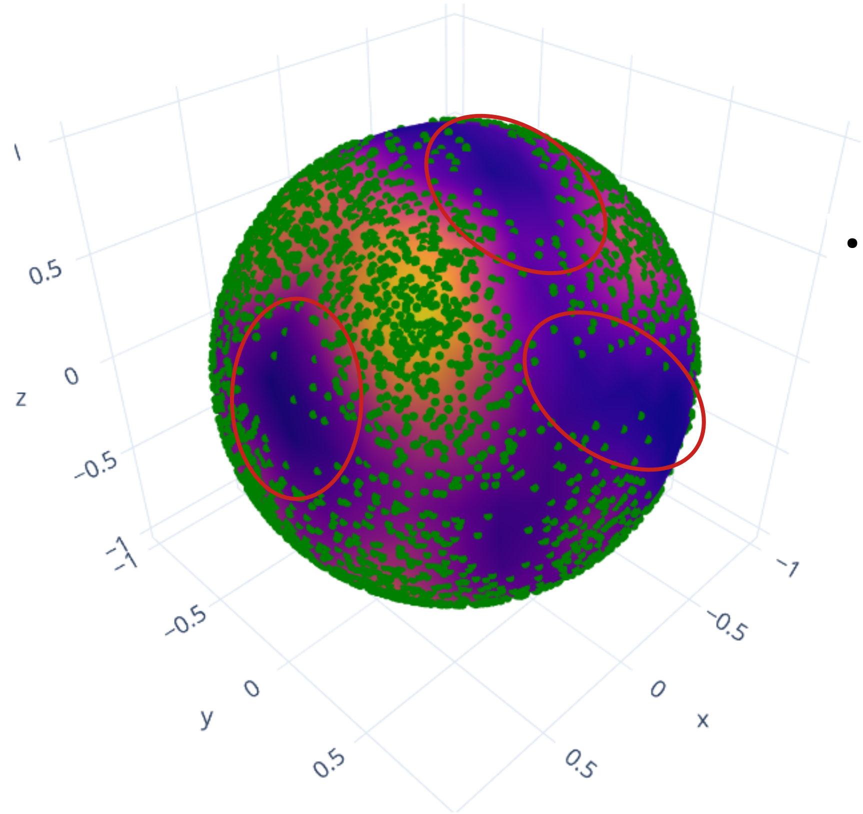
Center-Connected



1. Grid-Walk and i.i.d-Biased-Sampler best uniformity
2. RRT does not cover feasible set well
3. i.i.d-Biased-Sampler with most computational effort

Center-Connected

RRT

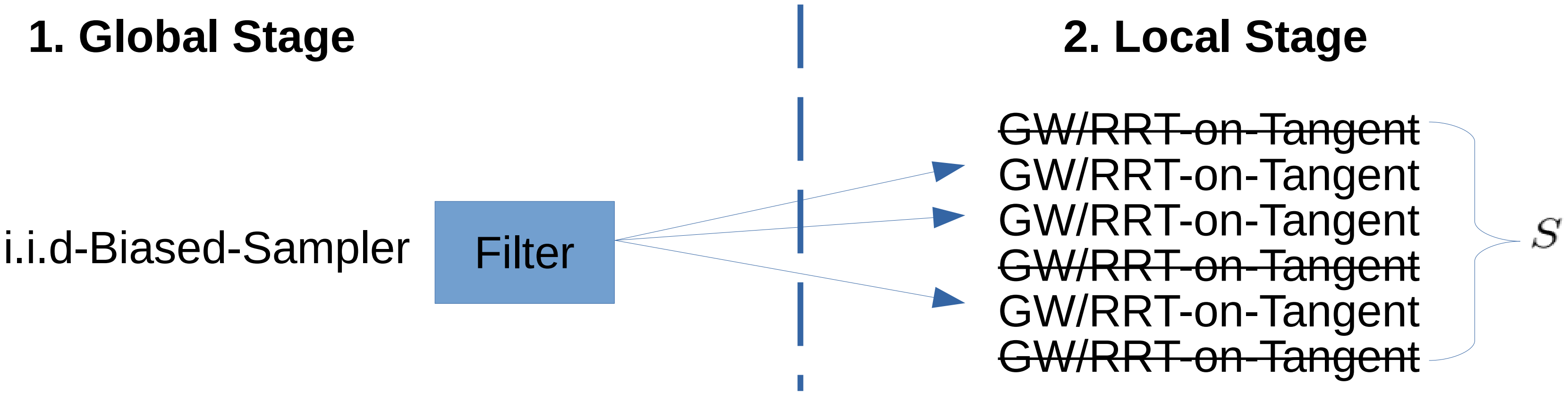


- RRT does not Cover Feasible Set well
  - Not sampled spots
    - Too narrow bounds on RRT expansion
  - Dense sampled spots
    - Many roots of RRTs near each other

Filter

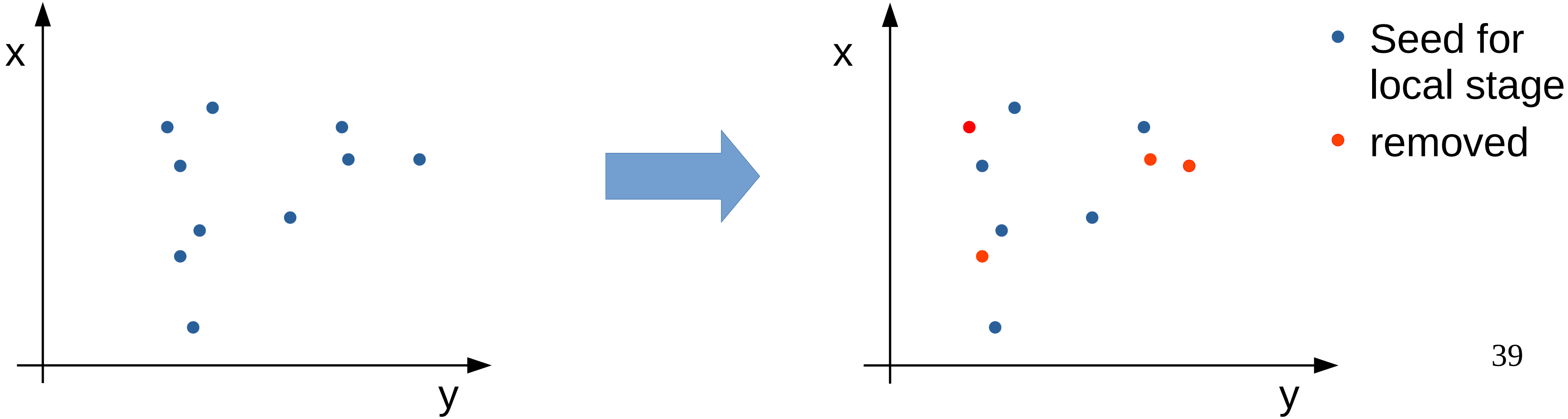
- Solve overlapping RRTs problem

Filter out starting points that are in close proximity to each other



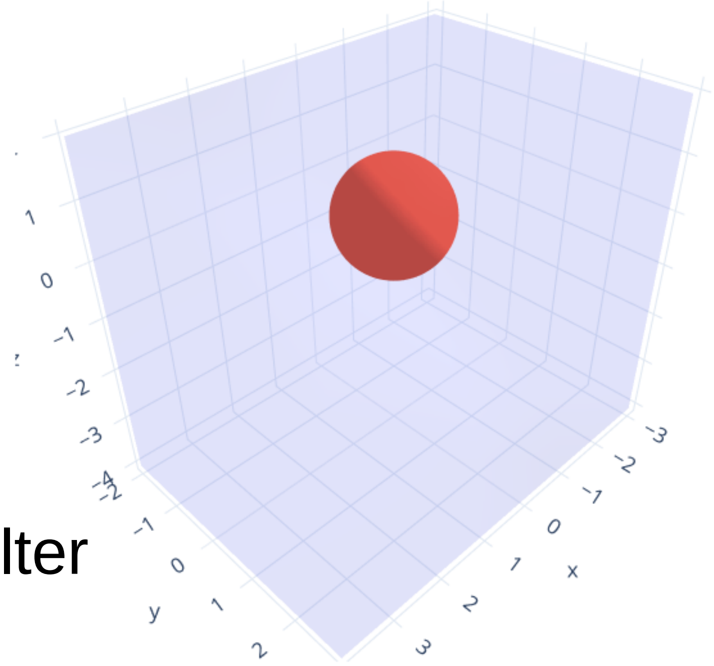
Removal Criteria

- If distance between between two points below a threshold
  - Then choose one point at random and remove

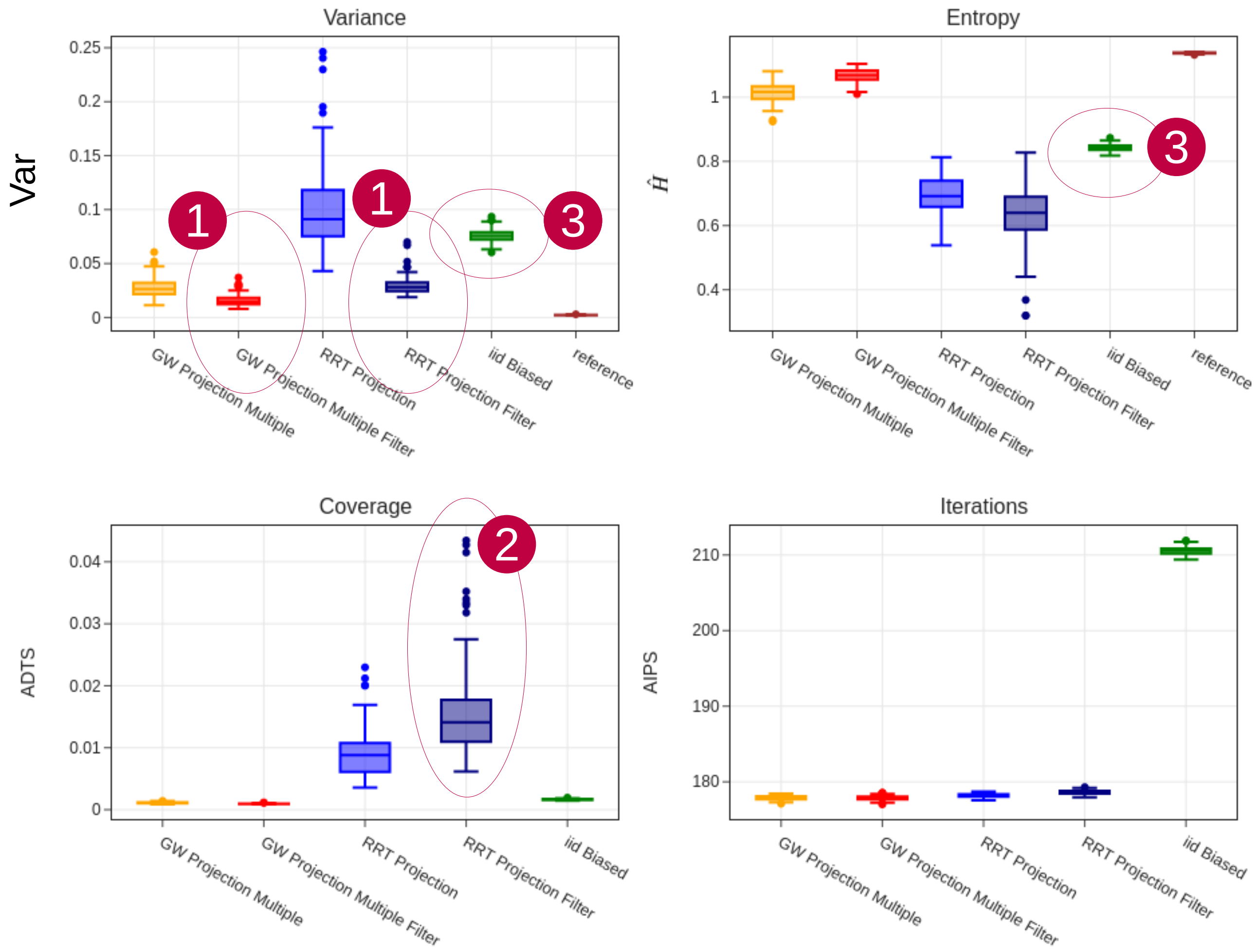


Off-Center-Connected

- Algorithms
  - i.i.d-Biased-Sampler
  - RRT-Projection
  - RRT-Projection-Filter
  - Grid-Walk-Projection-Multiple
  - Grid-Walk-Projection-Multiple-Filter



Off-Center-Connected



1 Filter decreases Variance

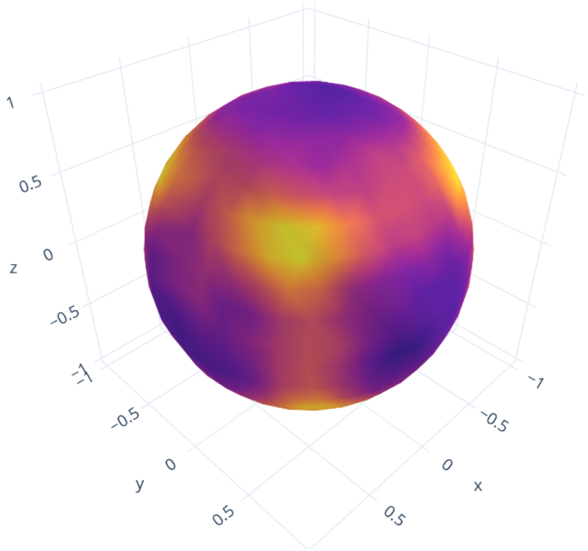
2 RRT-Filter with bad coverage
- Too many roots filtered out

3 i.i.d-Biased-Sampler dropped compared to Center-Connected

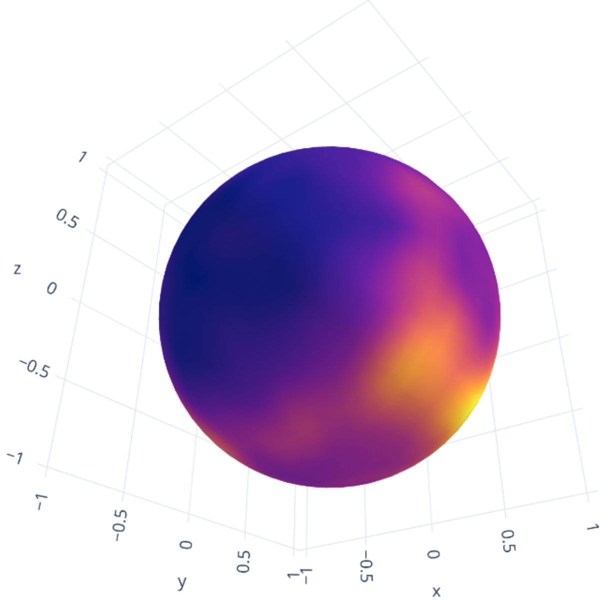
Off-Center-Connected

i.i.d.-Biased-Sampler

Center-Connected



Off-Center-Connected

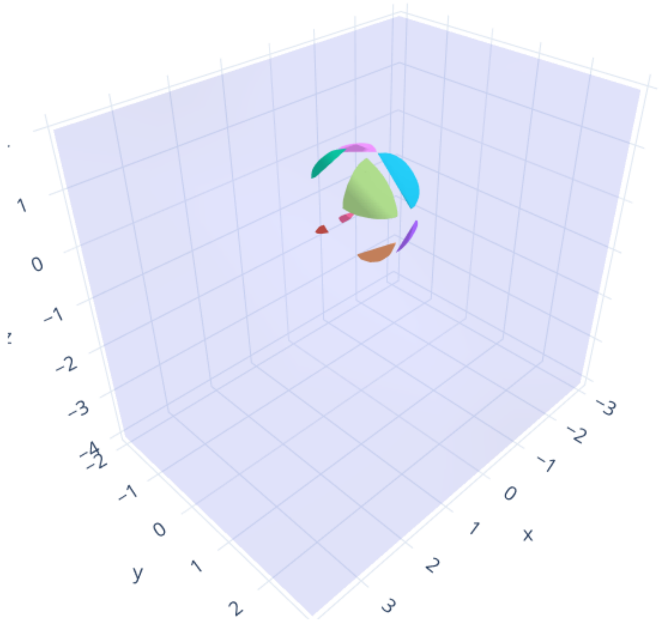


- i.i.d-Biased-Sampler dropped compared to Center-Connected
  - Configuration space not evenly distributed around manifold
  - Subspace of manifold that is exposed to large free configuration space sampled more often

Location of Manifold In Configuration Space matters

Off-Center-Disconnected

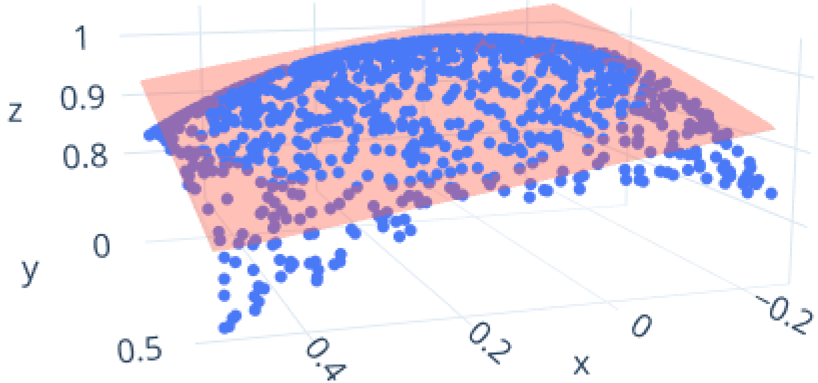
- What if manifold is disconnected?
- Should we use projection or rejection?
- What if a Grid-Walk chain is stuck on a disconnected manifold
  - Small manifolds parts will be sampled more dense
  - Manifolds with many Chains will also be sampled more dense



→ Sample chain proportional to size of the manifold it samples

Approximate Manifold Size *A*

- Use Eigendecomposition of **Cov** to find *k* largest eigenvalues (*k* – dimension of manifold)
- Eigenvalues λ relate to width *w* of the interval of support (uniform distribution)



$$\lambda = \mathrm{Var} = \frac{1}{12} w^2 \Rightarrow w = 12\lambda^{\frac{1}{2}}$$

$$A = \prod_{i=1}^{i=k} w_i = 12 \prod_{i=1}^{i=k} \lambda_i^{\frac{1}{2}}$$

Choose next Chain

$$P(\text{Chain}_i) = \frac{A_i}{\sum_{i=1}^{i=n_{\text{chains}}} A_i}$$

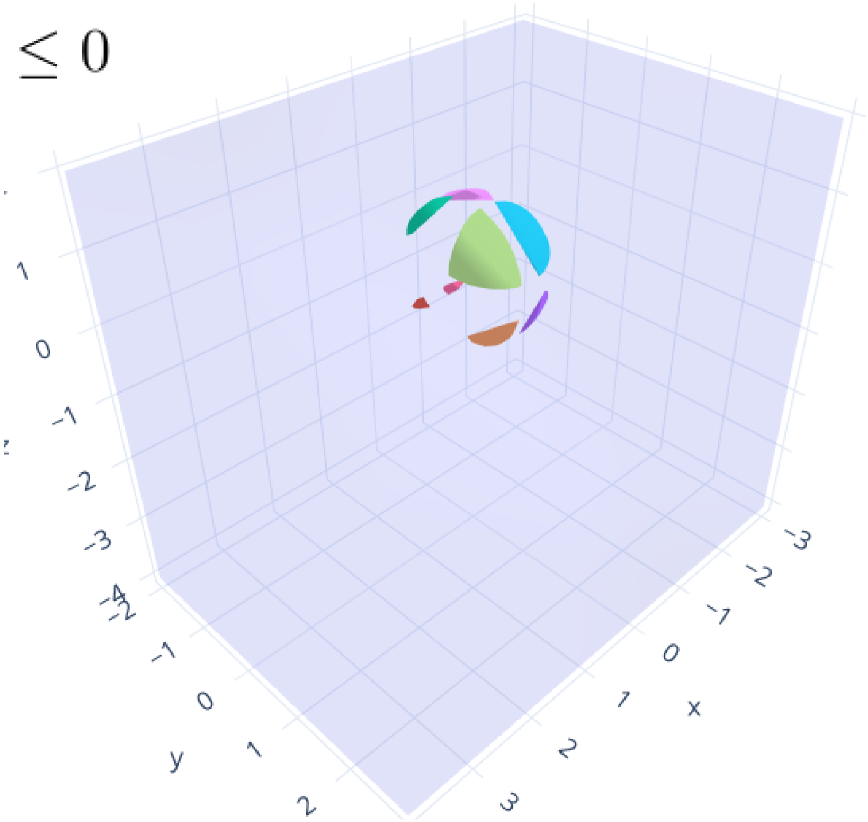
Off-Center-Disconnected

- Add inequality constraints $g_l(x) \leq 0$
- Sphere is separated into 8 parts
- Triangular manifolds of different size

$$g_1 = -5x_2^2 - x_3 + 1.2$$

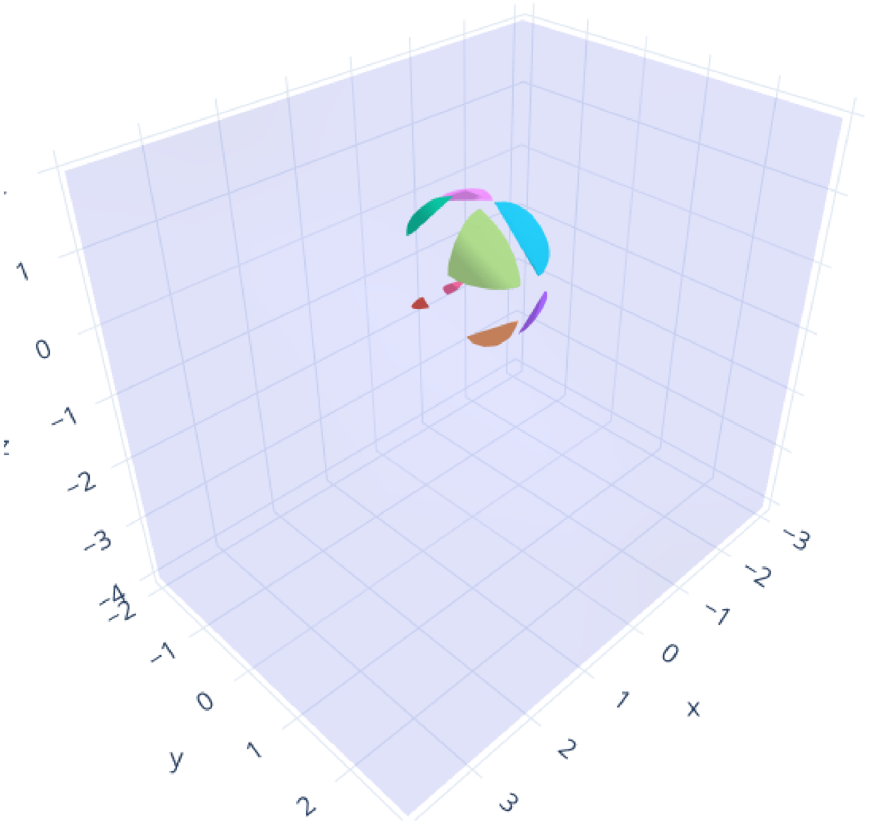
$$g_2 = -5x_3^2 - x_2 + 1.2$$

$$g_3 = -100x_1^2 - x_3 + 2$$

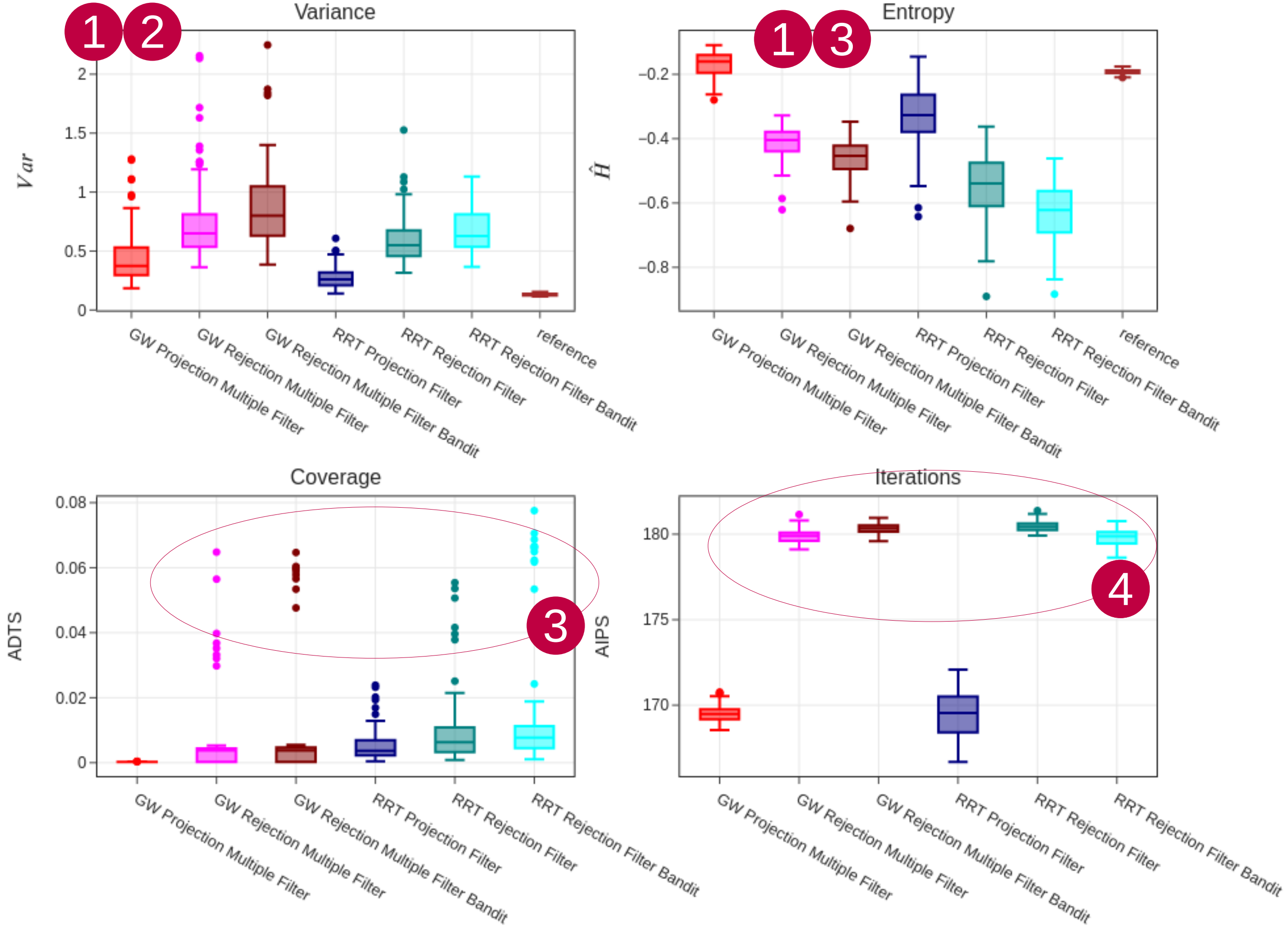


Off-Center-Disconnected

- Algorithms
  - Grid-Walk Projection Filter
  - Grid-Walk Rejection Filter
  - Grid-Walk Rejection Filter Bandit
  - RRT Projection Filter
  - RRT Rejection Filter
  - RRT Rejection Filter Bandit



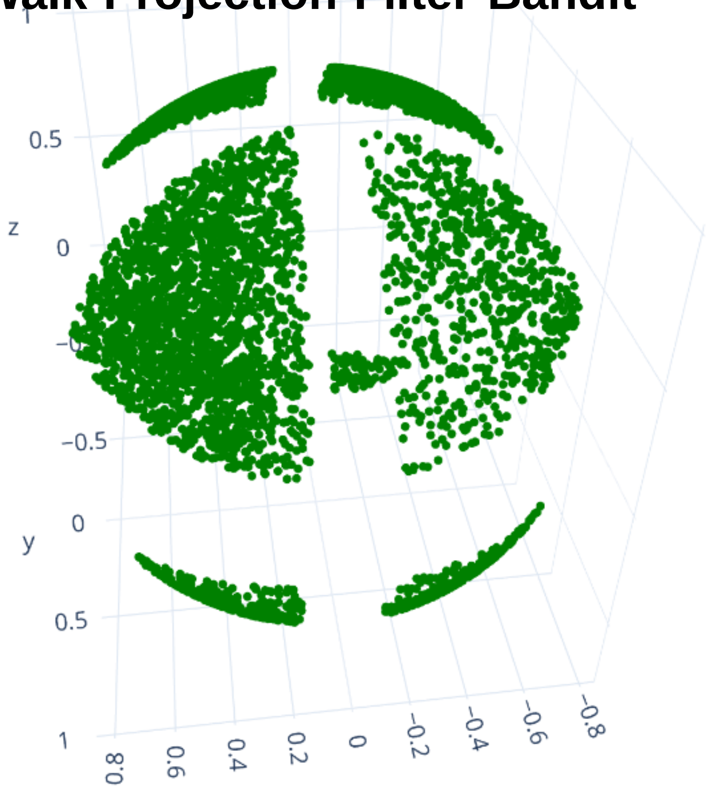
Off-Center-Disconnected



1. Bandit-Sampler not working very well
2. Also Rejection sampling does not bring improvement
3. **Outliers appear in metrics**
   - *Small manifold sometimes not covered at all*
4. **Rejection sampling is less computational efficient**
   - *Many rejected samples increase the number of optimization problems to solve*

Off-Center-Disconnected

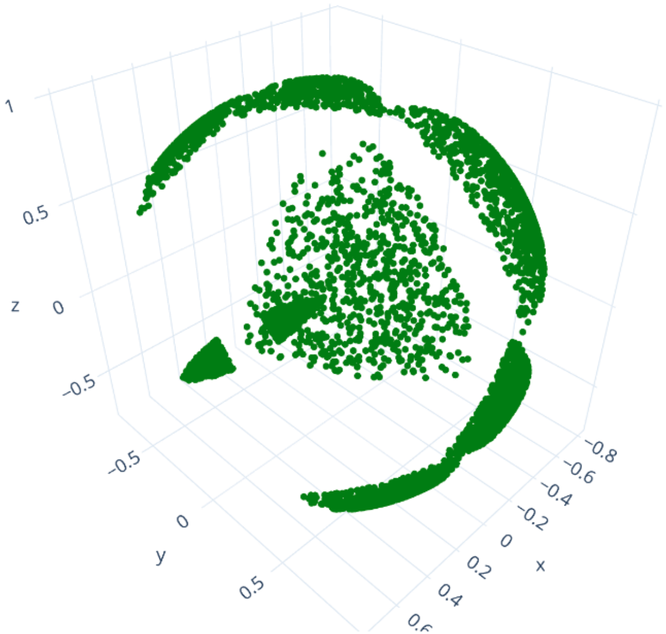
Grid-Walk-Projection-Filter-Bandit



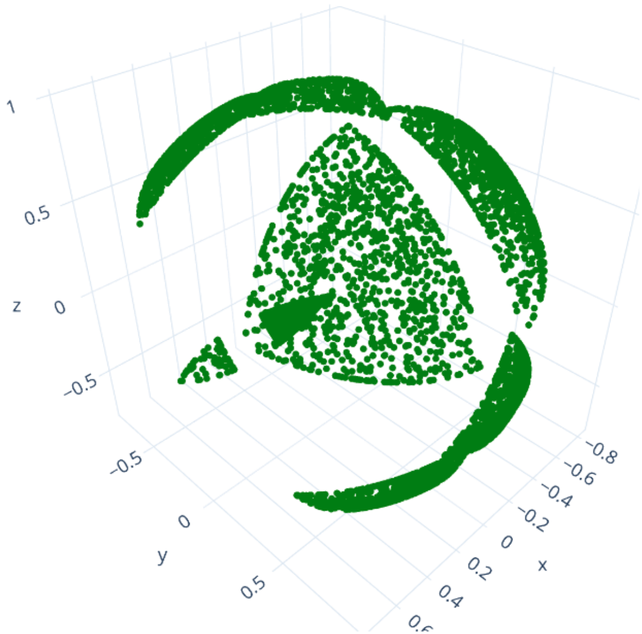
- Bandit-Sampler not working very well
  - Left pane much more sampled than right
  - Multiple chains sample the left pane
  - Possible Solution:
    - Cluster during sampling chains

Off-Center-Disconnected

Grid-Walk-Rejection-Filter



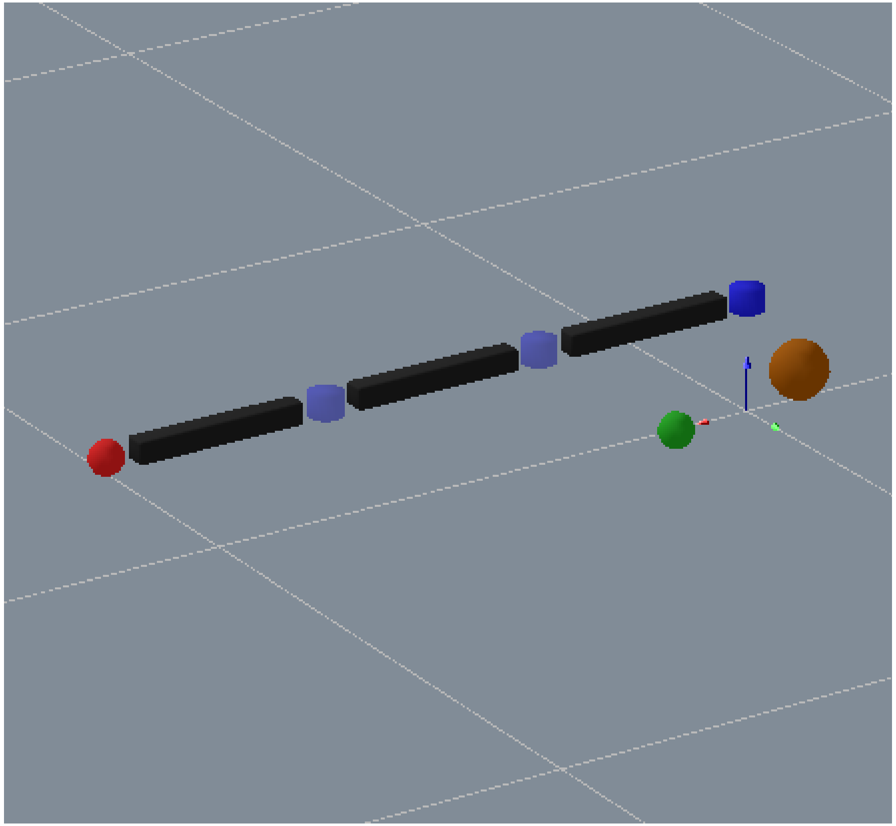
Grid-Walk-Projection-Filter



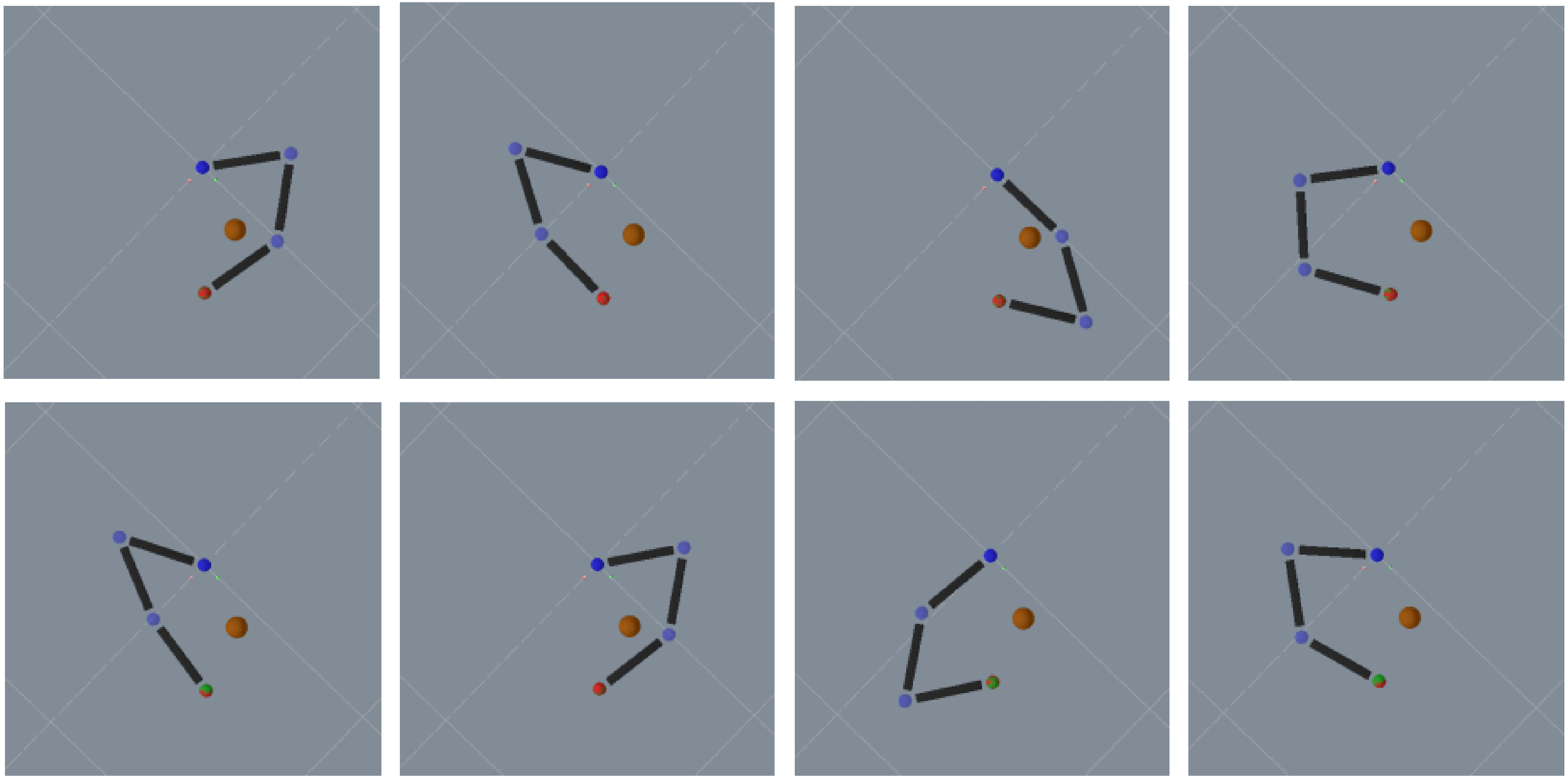
- Also Rejection sampling does not bring improvement
  - Corners and edges not well sampled in rejection
  - Distortion through accumulation of samples on edges not as big as expected

3-DOF Robotics Examples

- Algorithms
  - Grid-Walk Projection Filter
  - Grid-Walk Projection
  - RRT Projection Filter
  - RRT Projection
  - i.i.d-Biased-Sampler



3-DOF Robotics Examples



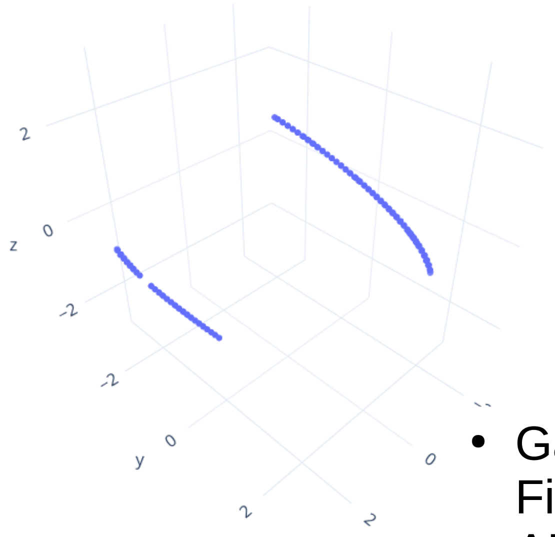
3-DOF Robotics Examples

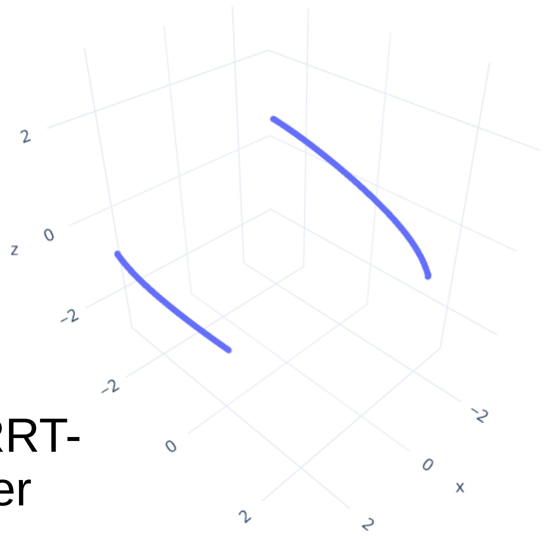
Algorithm	Variance Var	Entropy $\hat{H}$	ADTS	AIPS
i.i.d.-Biased-Sampler	0.182	-0.1	7.1e-5	341
RRT-filter	0.09	-0.13	1.4e-3	26
RRT	0.473	-0.11	1.9e-3	15
Grid-walk-filter	0.31	-0.19	4.1e-4	8
Grid-walk	0.2	-0.14	0.2e-4	5

- i.i.d-Biased-Sampler less computationally efficient
- i.i.d-Biased-Sampler and RRT-Filter overall best coverage and among best uniformity

3-DOF Robotics Examples

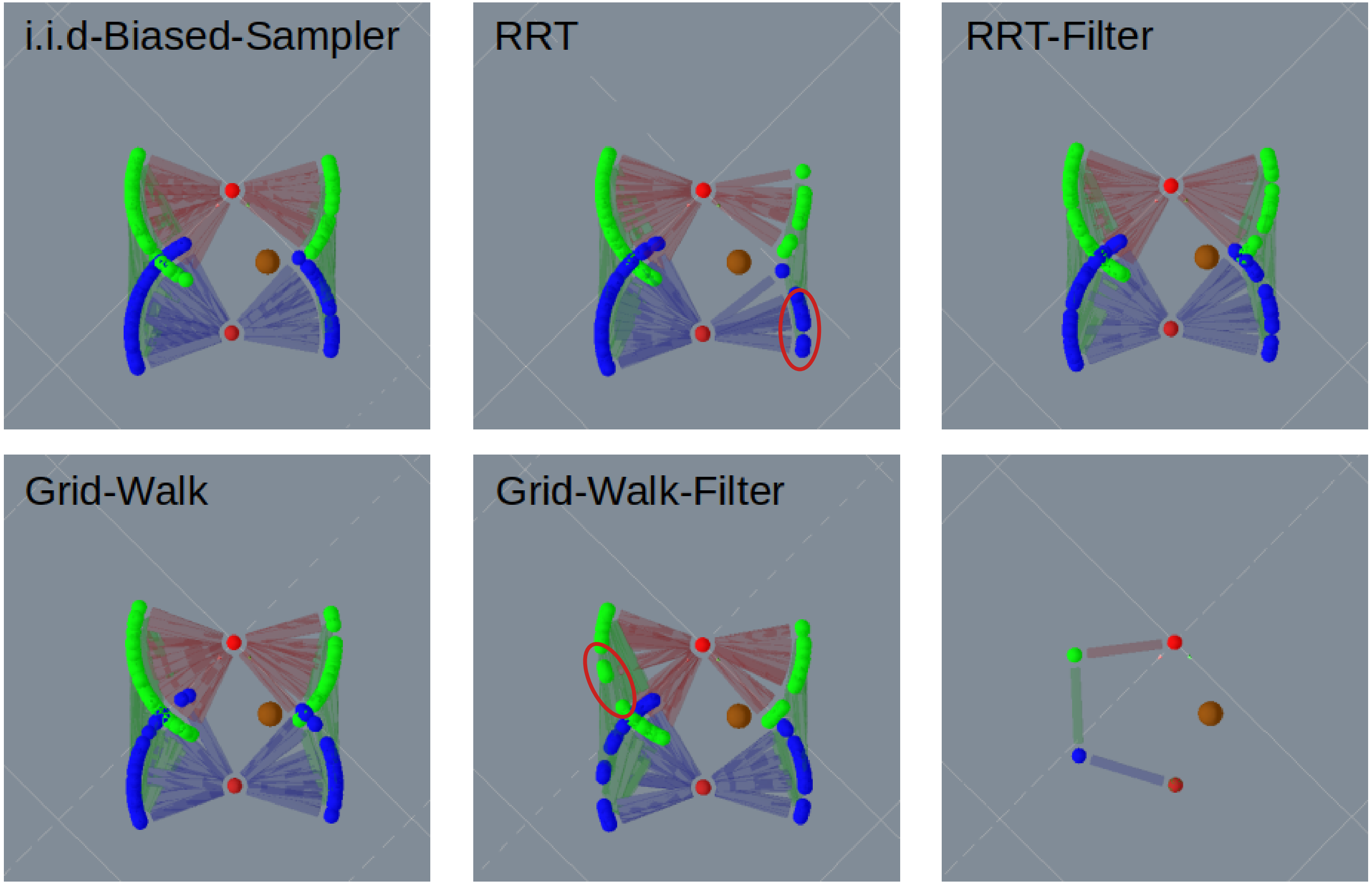
RRT-Filter | **i.i.d-Biased-Sampler**





- Gaps in manifold (RRT-Filter) result in higher ADTS

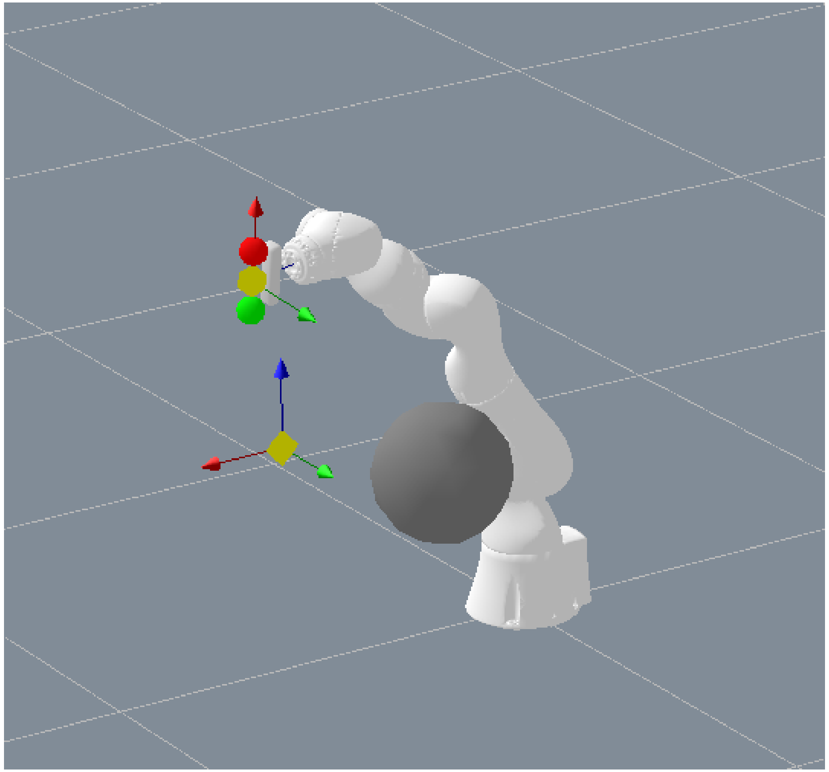
3-DOF Robotics Examples



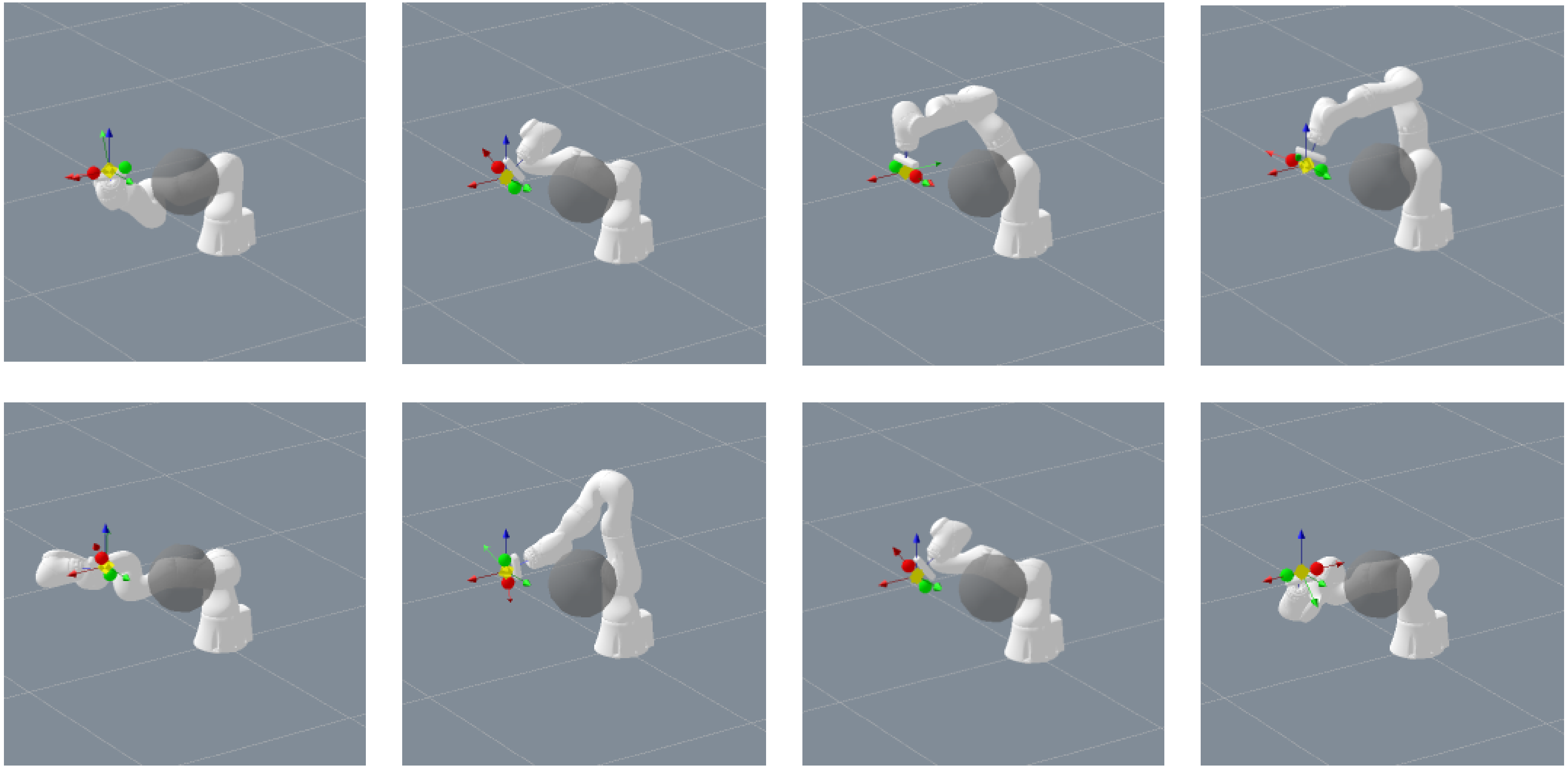
- Gaps in feasible set
- Parameter Choices?
  - Step size and neighborhood width too small?

7-DOF Robotics Examples

- Algorithms
  - Grid-Walk Projection Filter
  - Grid-Walk Projection
  - RRT Projection Filter
  - RRT Projection
  - i.i.d-Biased-Sampler



7-DOF Robotics Examples



7-DOF Robotics Examples

Algorithm	Variance Var	Entropy $\hat{H}$	ADTS	AIPS
i.i.d.-Biased-Sampler	7.77e-5	3.67	0.039	187
RRT-filter	0.018	1.41	0.621	21
RRT	0.03	1.23	0.687	20
Grid-walk-filter	0.069	-0.42	1.19	5
Grid-walk	0.064	-0.41	1.18	5

- i.i.d-Biased-Sampler less computationally efficient
- i.i.d-Biased-Sampler overall best coverage and among best uniformity

- Parameter Choices?
  - Step size and neighborhood width too small?
- i.i.d-Biased-Sampler
  - No parameters chosen
  - Better Ease-of-Use

Major Findings

- i.i.d-Biased-Sampler not computationally efficient
- Location of manifold in configuration space influence sampling distribution
  - i.i.d-Biased-Sampler works better when manifold is in the center
- Distribution of seeds for local stage influence the overall sampling distribution
  - If seeds are very near chains will sample the same subspace
  - If seeds are very far not reachable subspaces of the feasible set can be created
  - Filtering seeds can help (the situation that too many seeds are filtered out reverses the effects)
- Parameters
  - RRT is very sensitive to parameter choices
  - i.i.d-Biased-Sampler easier to use and more robust

Thank you!

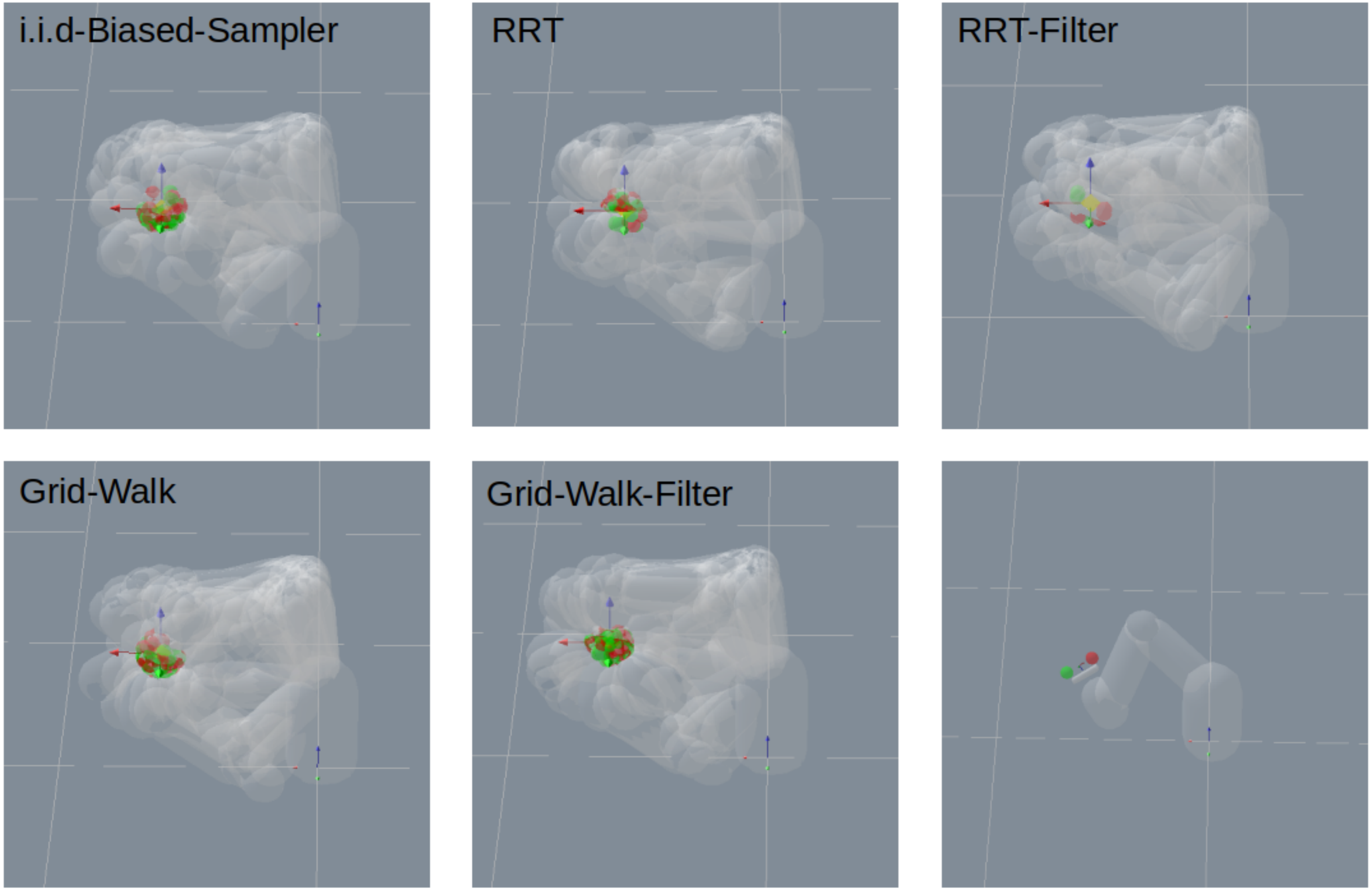
About

Philip Oedi
340896
Computational Engineering Science
M.Sc.
philip.oedi@gmail.com

Examiner:
Prof. Marc Toussaint (LIS)
Prof. Jörg Krüger (IAT)

Supervisor:
Ph.D. Jung-Su Ha (LIS)
M.Sc. Joaquim Ortiz-Haro (LIS)

7-DOF Robotics Examples

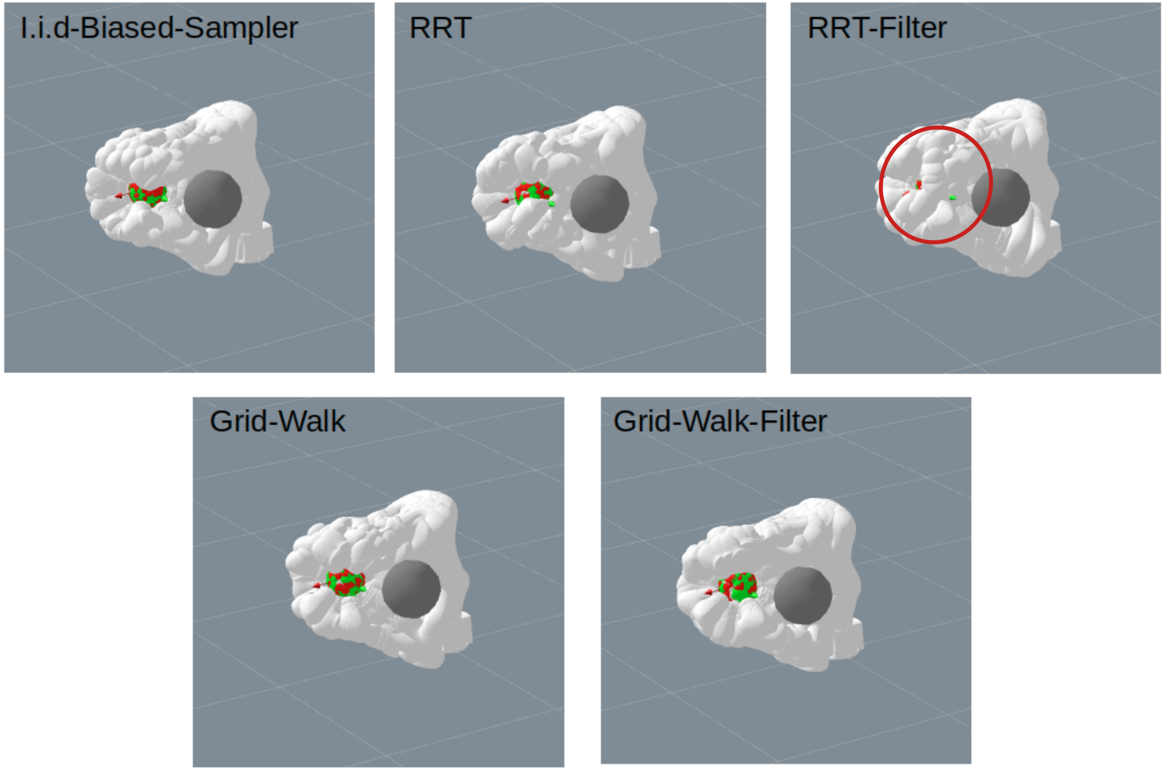


- Parameter Choices?
  - Step size and neighborhood width too small?
- i.i.d-Biased-Sampler
  - Without preceeding parameter choices
  - Better Ease-of-Use

Outlook

- Develop heuristics to choose parameters for Grid-Walk and RRT
- Clustering for Bandit-Sampler
- Apply other methods on composite sampler (for example AtlasRRT)

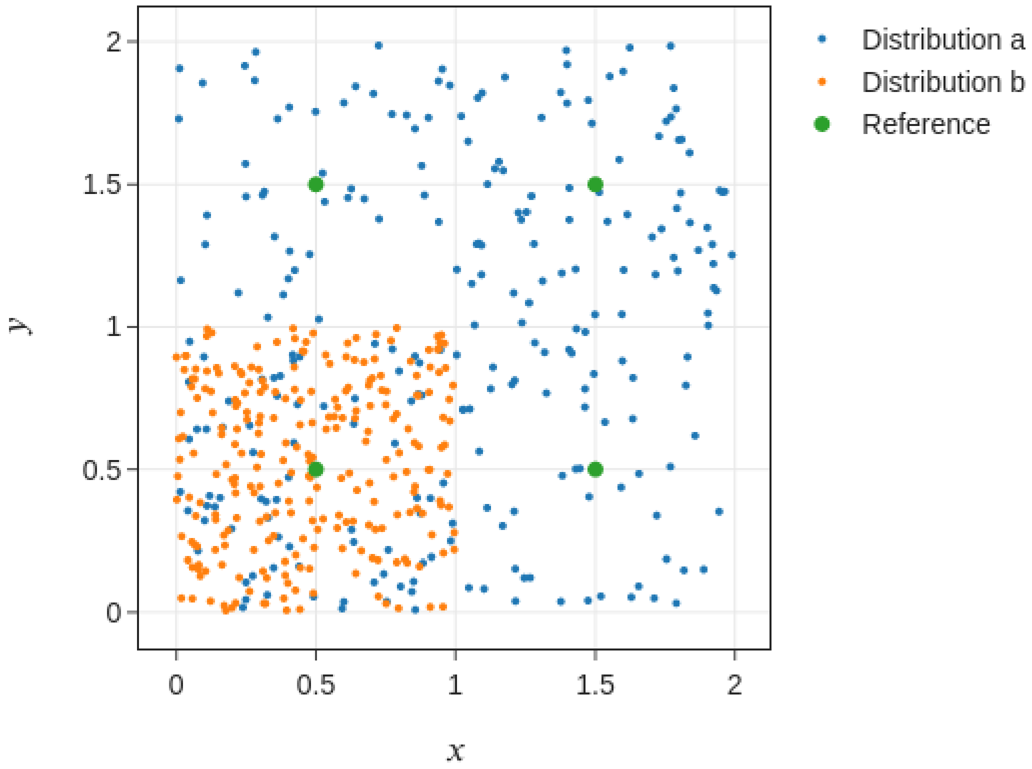
7-DOF Robotics Examples



- RRT
  - More poses sampled where arm between obstacle and goal
  - Maybe a manifold in a corner and difficult to reach for i.i.d-Biased-Sampler
  - RRT just walks there

Coverage

Two uniform sampling distributions with different ADTS.



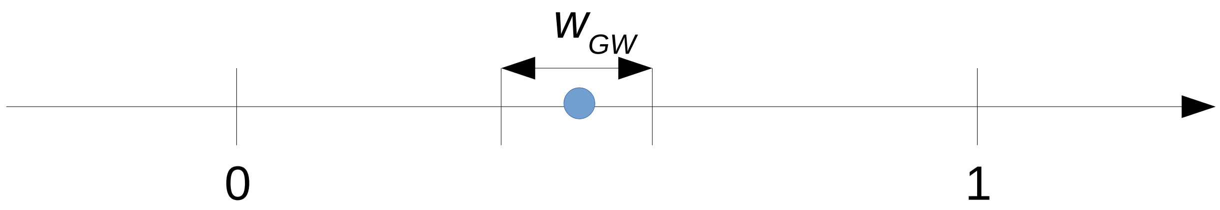
Unit-Line Experiment

- Study most simple scenario
- Projection vs. Rejection?
  - How does the sampling distribution look at the borders?
- How will parameters influence stationary distriubtion?

1. Generate samples using Grid-Walk and RRT over the unit-line
   - For varying w_{GW} and b parameters
2. Train KDE on samples
3. Visualize estimated PDF over unit-line

Unit-Line Experiment

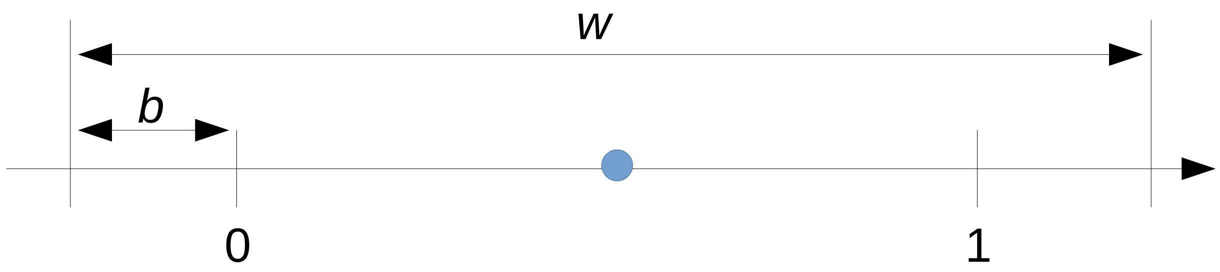
Grid-Walk



w_{GW} - neighborhood width

Unit-Line Experiment

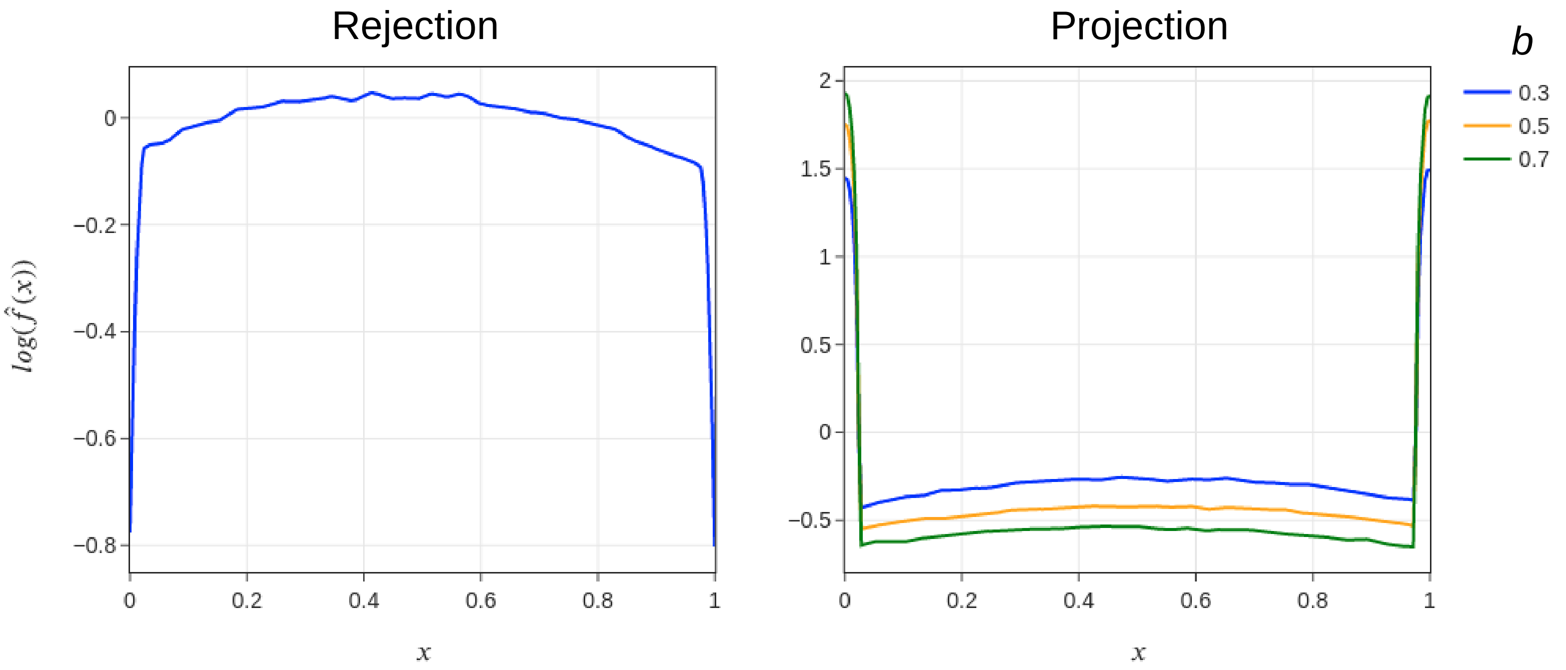
RRT



w - length of bounding window of RRT
b - length of extension over the feasible set

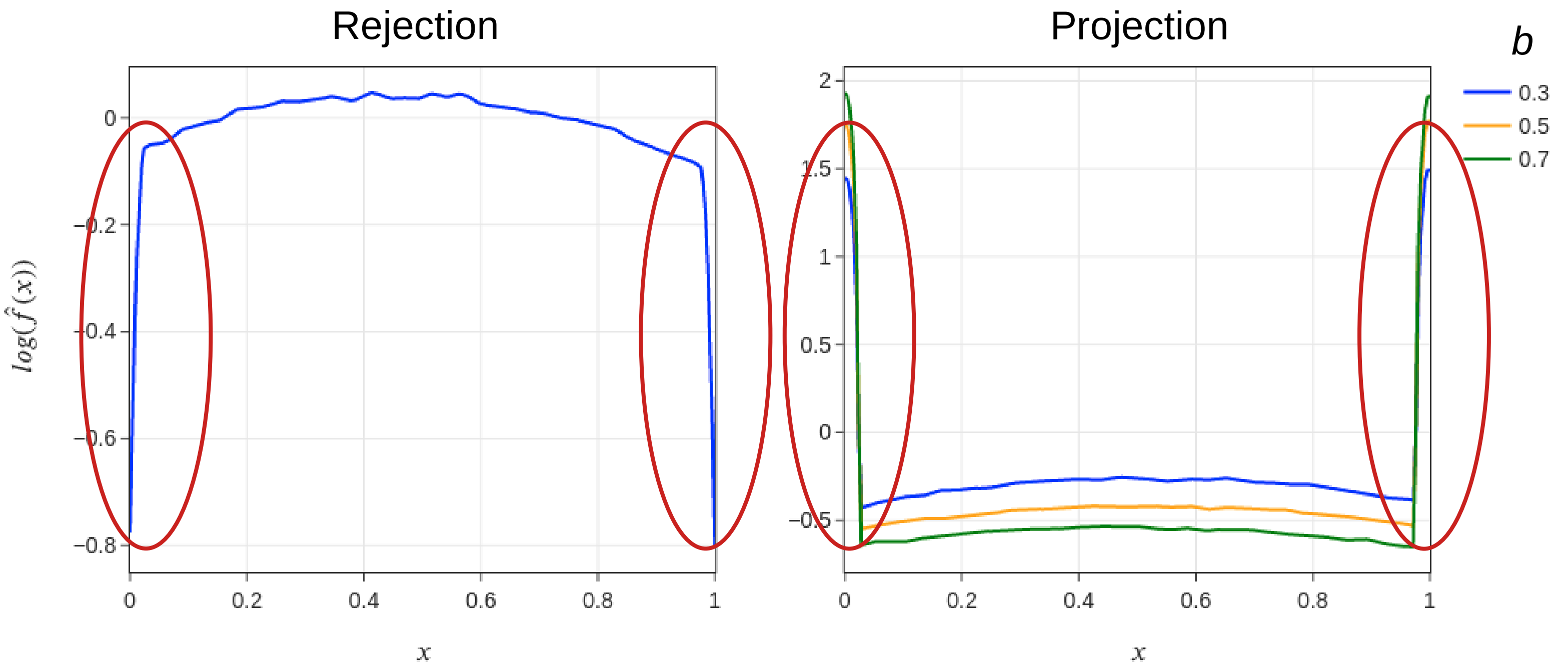
Unit-Line-Experiment

RRT



Unit-Line-Experiment

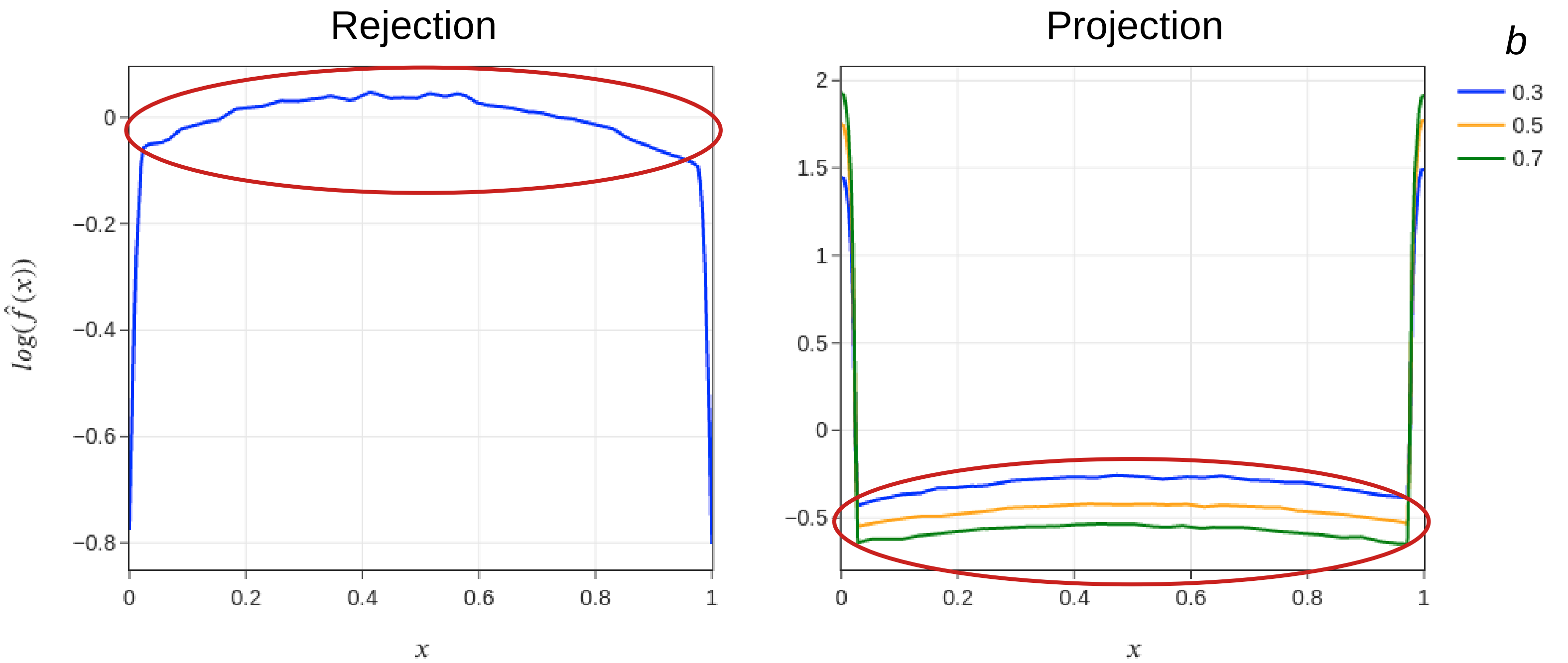
RRT



- Projection:
  - Borders accumulate samples
  - Large *b* result in high density on border
- Rejection:
  - Borders are not sampled in Rejection

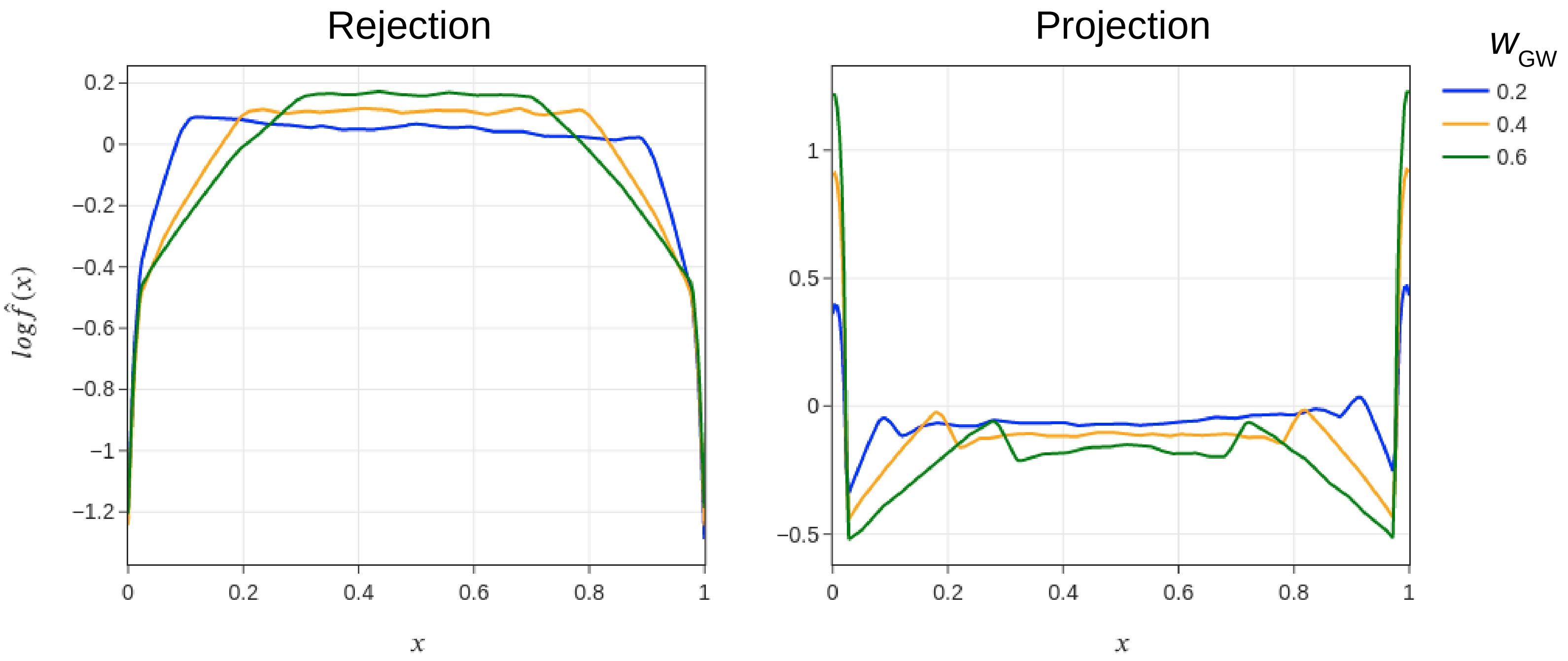
Unit-Line-Experiment

RRT

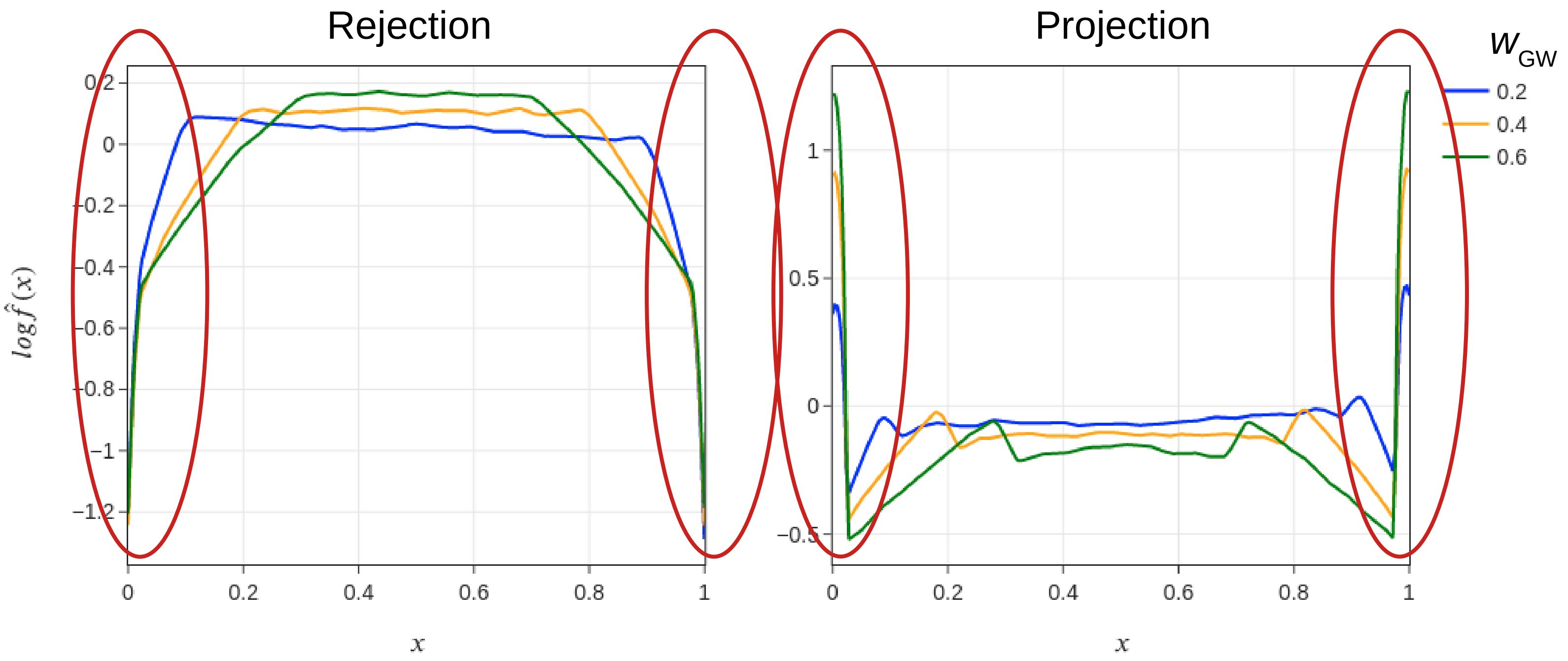


- Almost flat segment in the center
- Projection: Small *b* results in a more densely sampled center segment

Grid-Walk

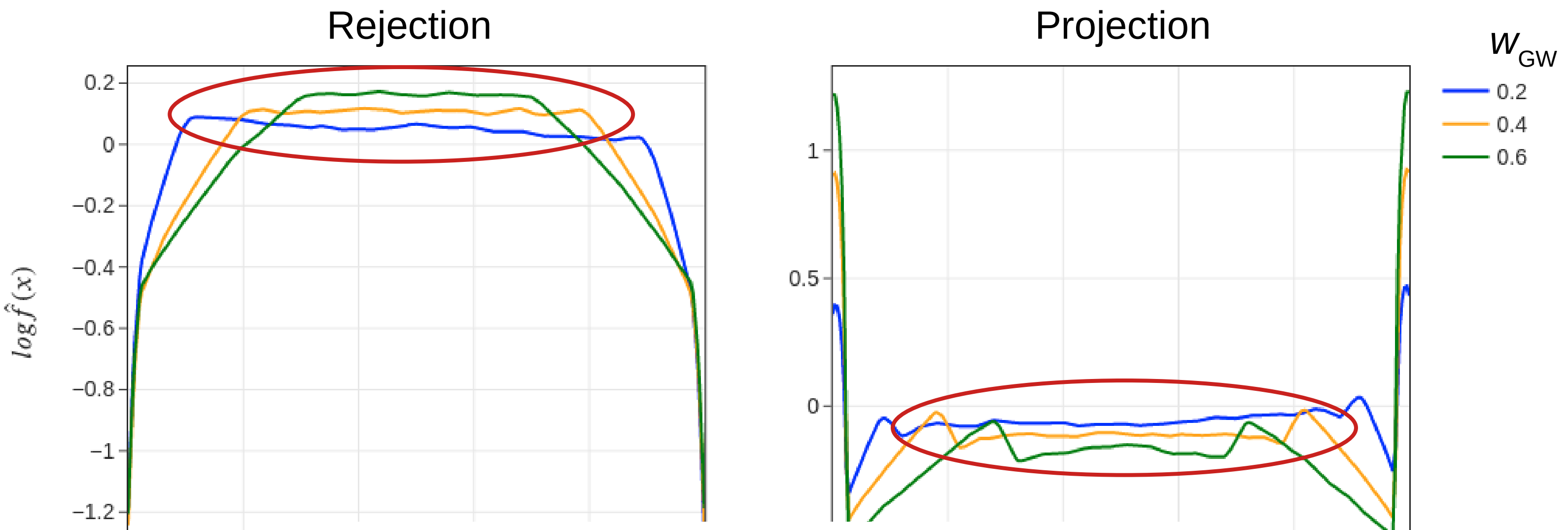


Grid-Walk



- Projection:
  - Borders accumulate samples
  - Large w_{GW} result in high density on border
- Rejection:
  - Borders are not sampled in Rejection

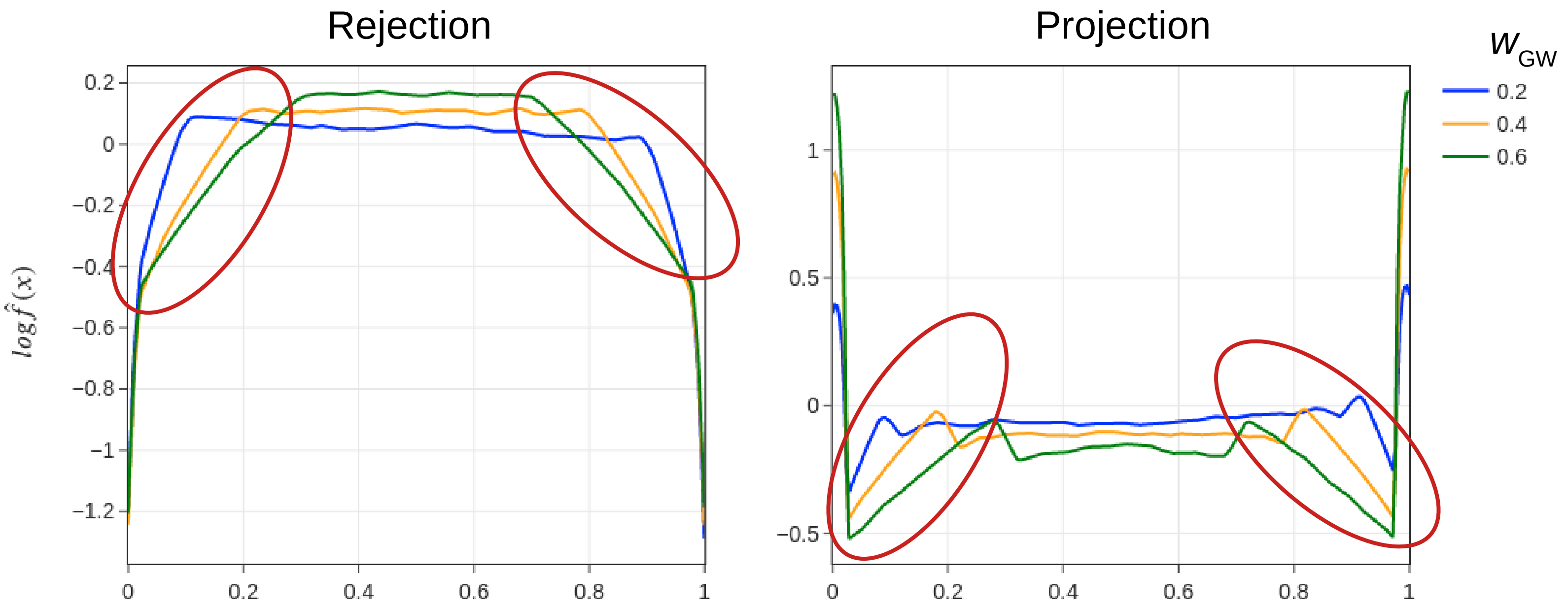
Grid-Walk



- Projection:
  - Large w_{GW} makes flat segment shorter
  - Large w_{GW} results in less dense sampled flat segment
- Rejection:
  - Borders are not sampled in Rejection
  - Large w_{GW} results in more dense sampled flat segment

Unit-Line Experiment

Grid-Walk



- Projection:
  - Linear slope
  - Many dips (local minima/maxima)
- Rejection:
  - Linear slope
  - Start and end of flat segment depend on w_{GW}

Take away

- Choices of w_{GW} and bounds of the RRT influence sampling distribution
- Samples accumulate at special points
- Accumulation of points reinforces local non-uniformity
- With far enough distance from points of accumulated samples, a more uniform distribution can be achieved

1-d Manifold embedded in 2-d

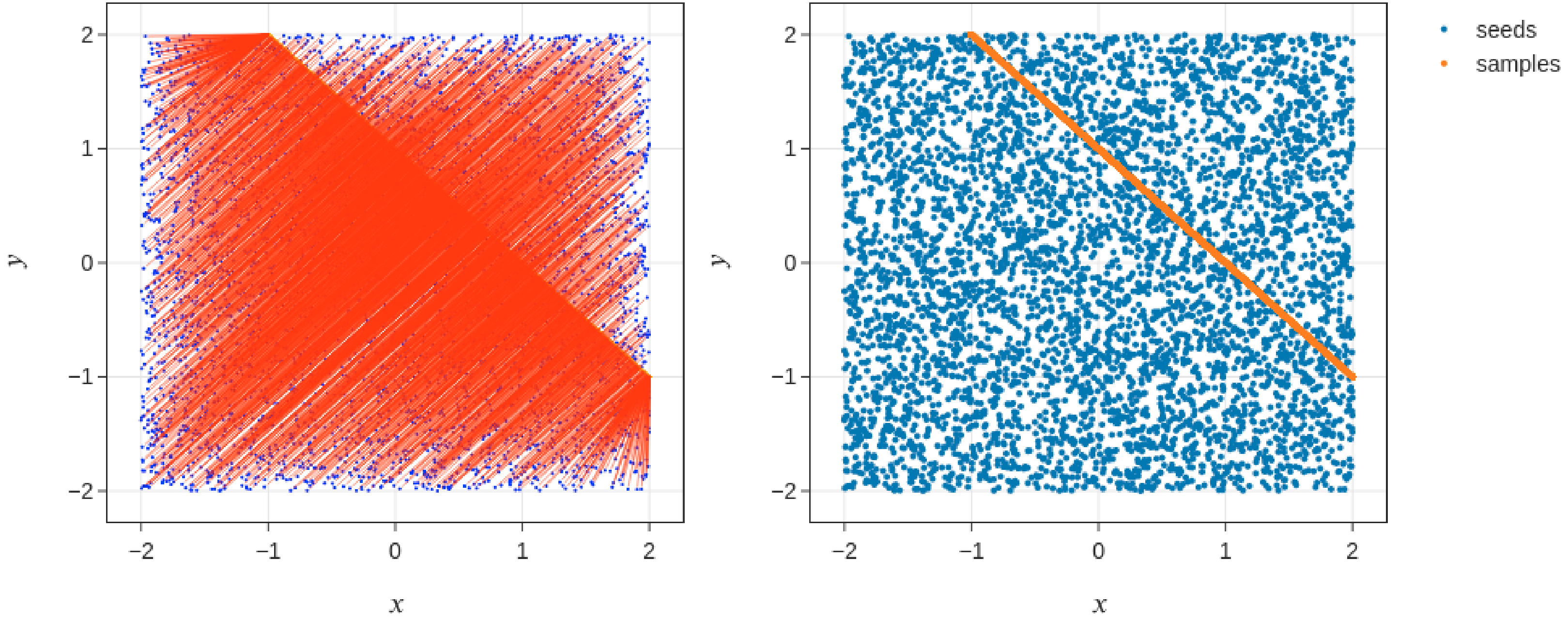
- Apply i.i.d-Biased-Sampler to sample from

$$\mathcal{X} = \{x \in \mathbb{R}^2 : -2 \leq x_i \leq 2; -\frac{2}{3}x_1 + \frac{4}{3} = x_2\}$$

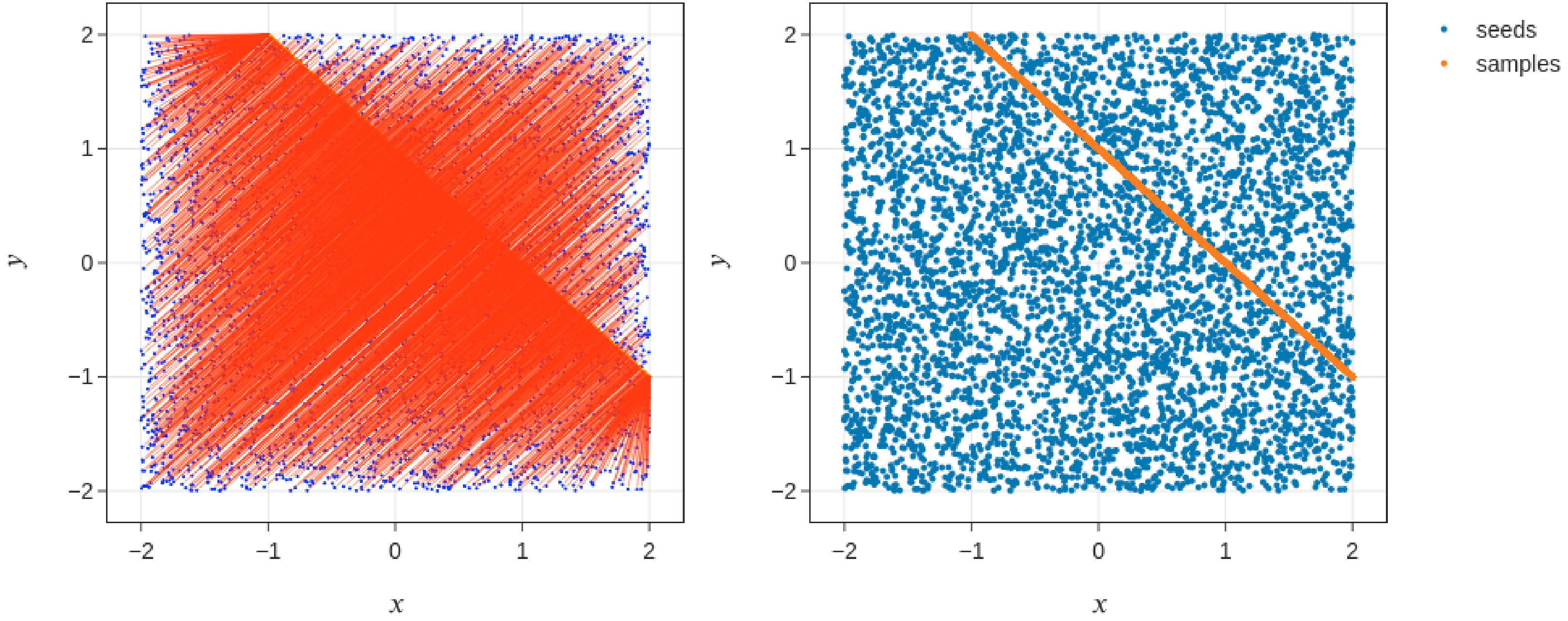
- How does Biased Optimization project onto the manifold?
- What is the resulting PDF over the manifold?

→ 1. Plot vertice between bias and feasible sample
2. Evaluate KDE over the manifold

1-d Manifold embedded in 2-d

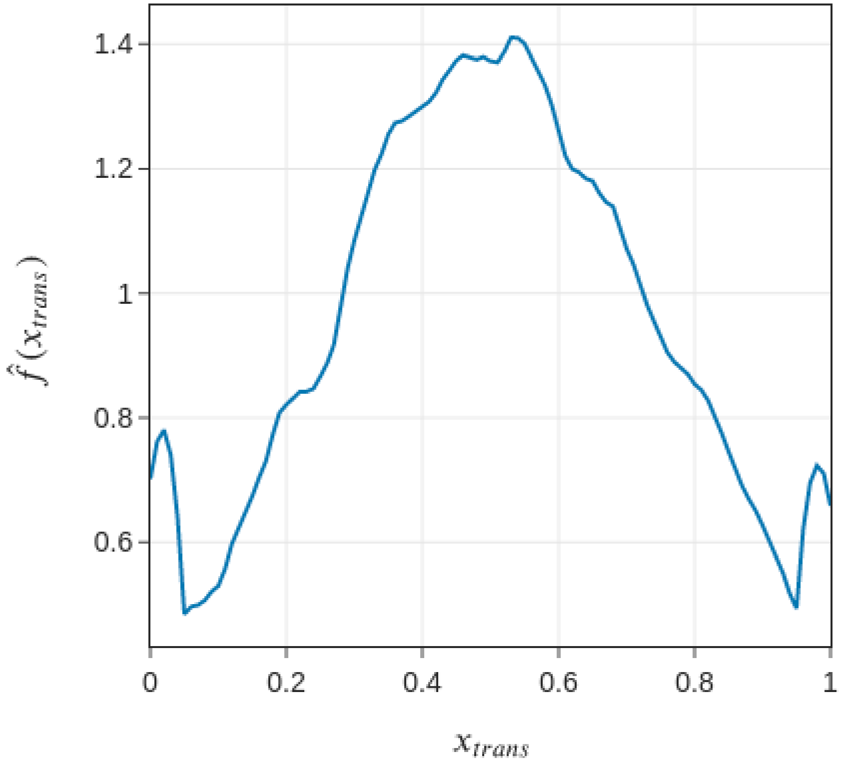


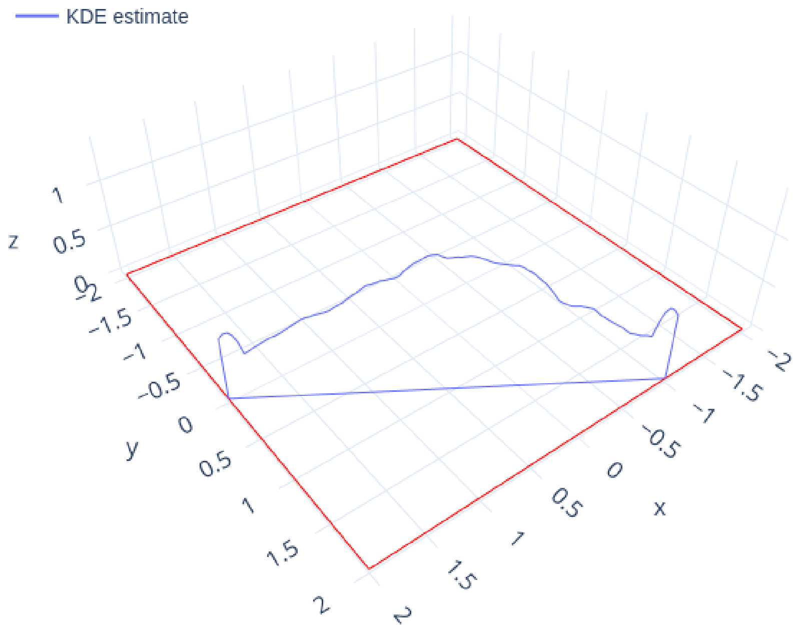
1-d Manifold embedded in 2-d



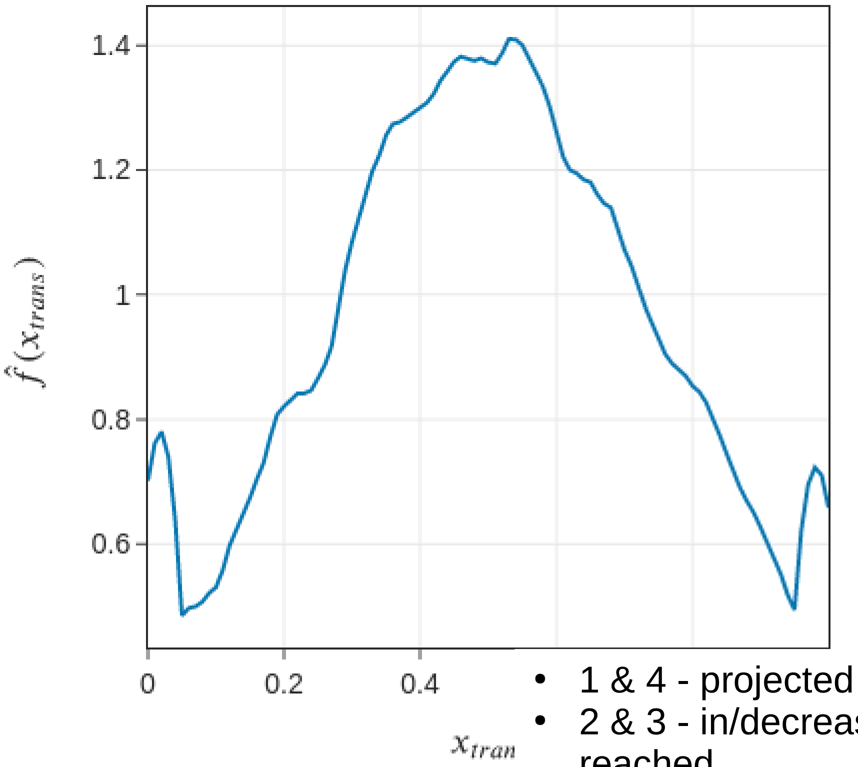
- Projects normal to manifold
  - Except for top left corner and bottom right
- Dense lines in the center

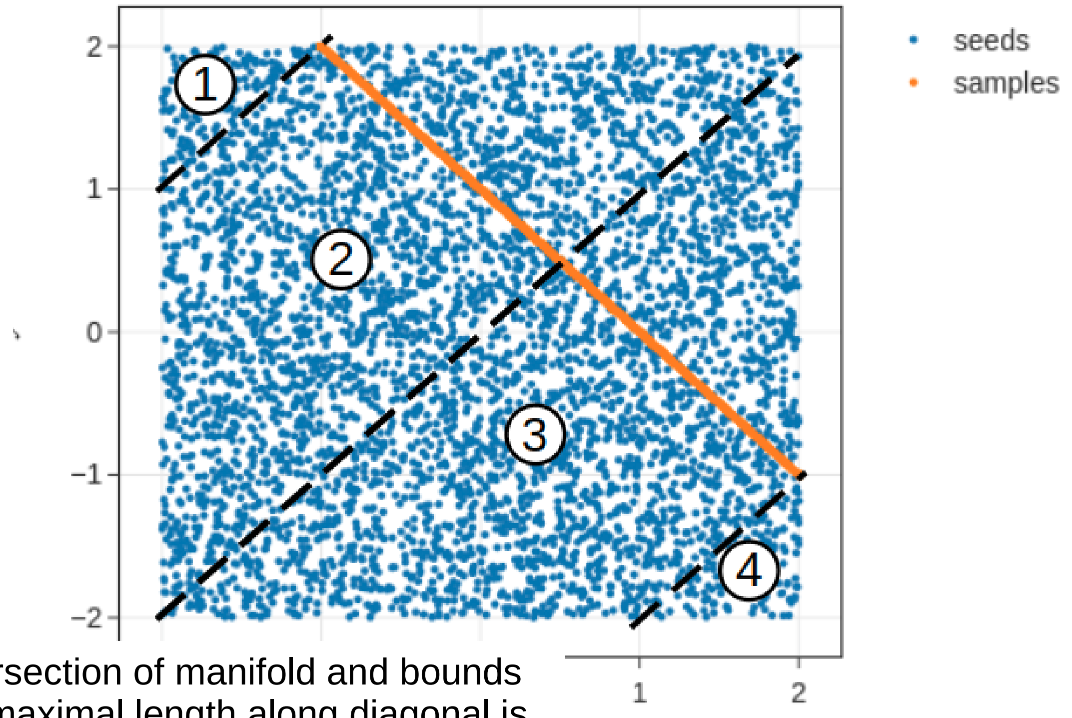
1-d Manifold embedded in 2-d





1-d Manifold embedded in 2-d





- 1 & 4 - projected to intersection of manifold and bounds
- 2 & 3 - in/decrease till maximal length along diagonal is reached

Take-Aways

- The sampling distribution depends on the location of the manifold
- For linear manifold Biased Optimization is projection to nearest feasible point
- Intersection of Manifold and Bounds accumulate samples
- Probability density depends on space normal to a linear manifold

Find-Tangent

Finds $\mathbf{\Theta} \in \mathbb{R}^{n,m}$ an orthonormal basis of the tangent space by solving:

$$\begin{bmatrix} \boldsymbol{J} \\ \mathbf{\Theta}^T \end{bmatrix} \mathbf{\Theta} = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I} \end{bmatrix}$$

with

$\boldsymbol{J}$ - Jacobian of $h(x)$

Sample Tangent

Sample a $u \in \mathbb{R}^m$ in the coordinates of the tangent from the distribution

$$u \sim U(-\frac{w_{\mathrm{GW}}}{2}, \frac{w_{\mathrm{GW}}}{2})$$

and map back into the coordinates of the configuration space to achieve a random sample on the tangent space

$$x_{\mathrm{rand}} = x_{i-1} + \boldsymbol{\Theta} u$$

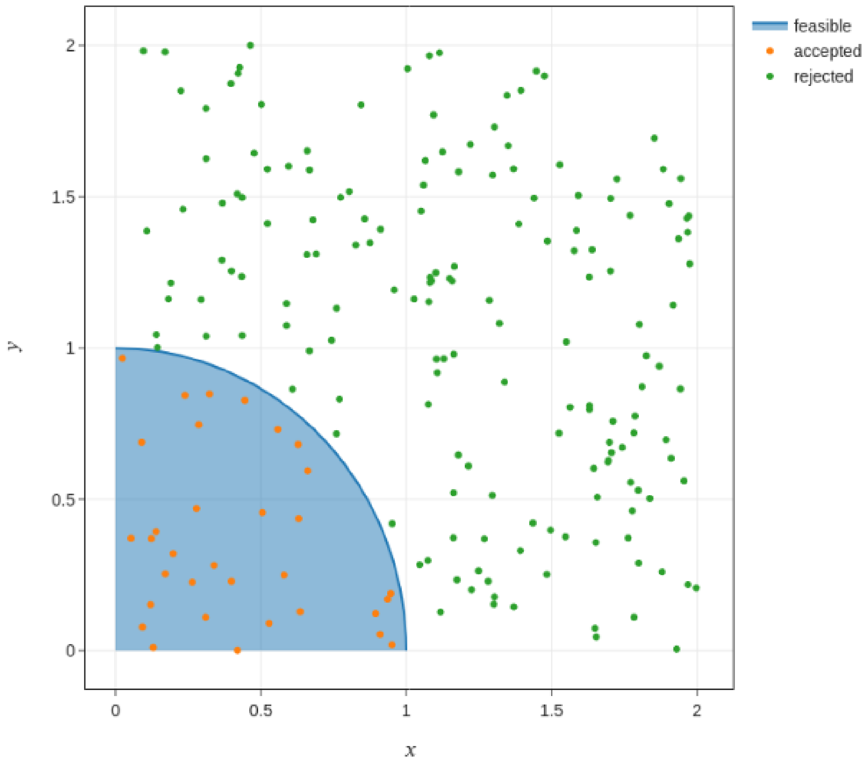
Rejection Sampling

Algorithm 2 Inequality Constrained Rejection Sampling

Input: $n \in \mathbf{N},\ b_{\text{low}},\ b_{\text{up}},\ g(x)$

$i \leftarrow 1$
while $i \leq n$ **do**
    $x_i \sim U(b_{\text{low}},\ b_{\text{up}})$
    if $g(x_i) \leq 0$ **then**
        Accept x_i
        $i \leftarrow i + 1$
    else
        Reject x_i
    end if
end while

Rejection Sampling



Uniform Samples over a Sphere

Deserno describes how this reference set can be sampled defining the spherical coordinates of a sphere with unit radius [Des04]:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \sin\theta \ \cos\phi \\ \sin\theta \ \sin\phi \\ \cos\theta \end{bmatrix} \tag{4.14}$$

Here $\theta \in [0; \pi]$ is the polar angle and $\phi \in [0; 2\pi]$ is the azimuthal angle. Then we sample a

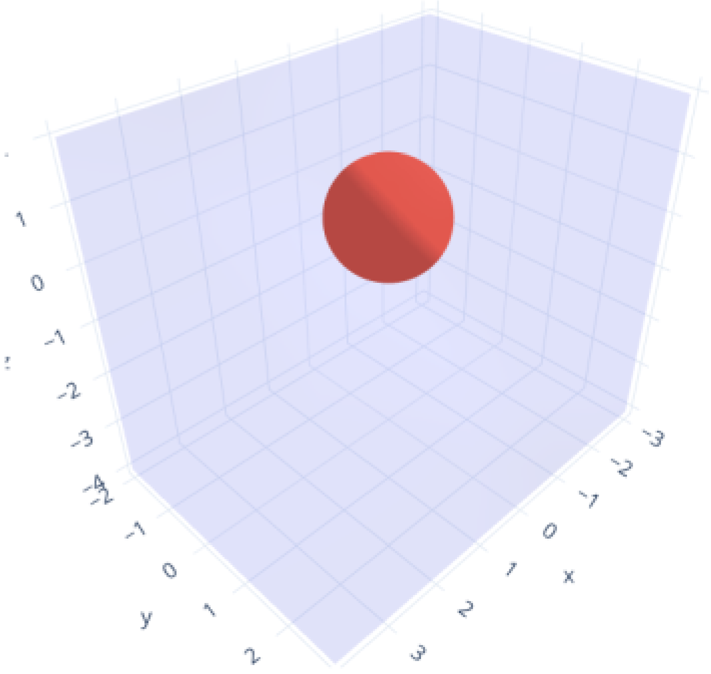
$x_3 \sim U(-1, 1)$ and a $\phi \sim U(0, 2\pi)$ and find the cartesian coordinates of our random point as:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} (1 - x_3^2)^{\frac{1}{2}} \ \cos\phi \\ (1 - x_3^2)^{\frac{1}{2}} \ \sin\phi \\ x_3 \end{bmatrix} \tag{4.15}$$

Off-Center-Connected

$$\mathcal{X} = \{x \in \mathbb{R}^3 : b_{\text{low}} \leq x \leq b_{\text{up}}; x_1^2 + x_2^2 + x_3^2 = 1\}$$

$$b_{\text{low}} = \begin{bmatrix} -3 \\ -2 \\ -4 \end{bmatrix}, \; b_{\text{up}} = \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix}$$



Kernel Density Estimation

The kernel density estimator $\hat{f}(x)$ for $x \in \mathbb{R}^q$ can also be found for multivariate sets of samples and is defined as follows.

$$\hat{f}(x) = \frac{1}{n|H|} \sum_{i=1}^{n} K\left(H - 1(x_i - x)\right) \tag{2.42}$$

H is a vector of dimension q which elements represent the bandwidth along the specified axis.

$$|H| = h_1 h_2 \ldots h_q \tag{2.43}$$

$K(u)$ is a multivariate kernel.

$$K(u) = k(u_1)k(u_2)\ldots k(u_3) \tag{2.44}$$

Kernel Density Estimator

$$k_{\text{epanechnikov}}(u) = \frac{3}{4}(1 - u^2)\mathbf{1}(|u| \leq 1)$$

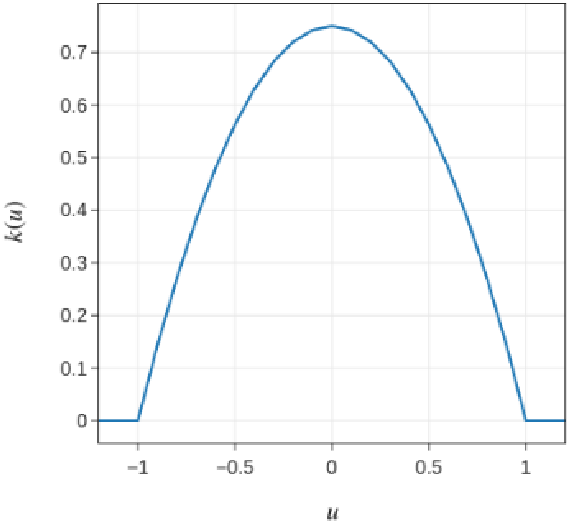


Figure 2.9: $k(u)$ of the Epanechnikov-Kernel

Updating Mean and Covariance

$$\bar{x}_{n+1} = \frac{1}{n-1}(n\bar{x}_n + x_{n+1})$$

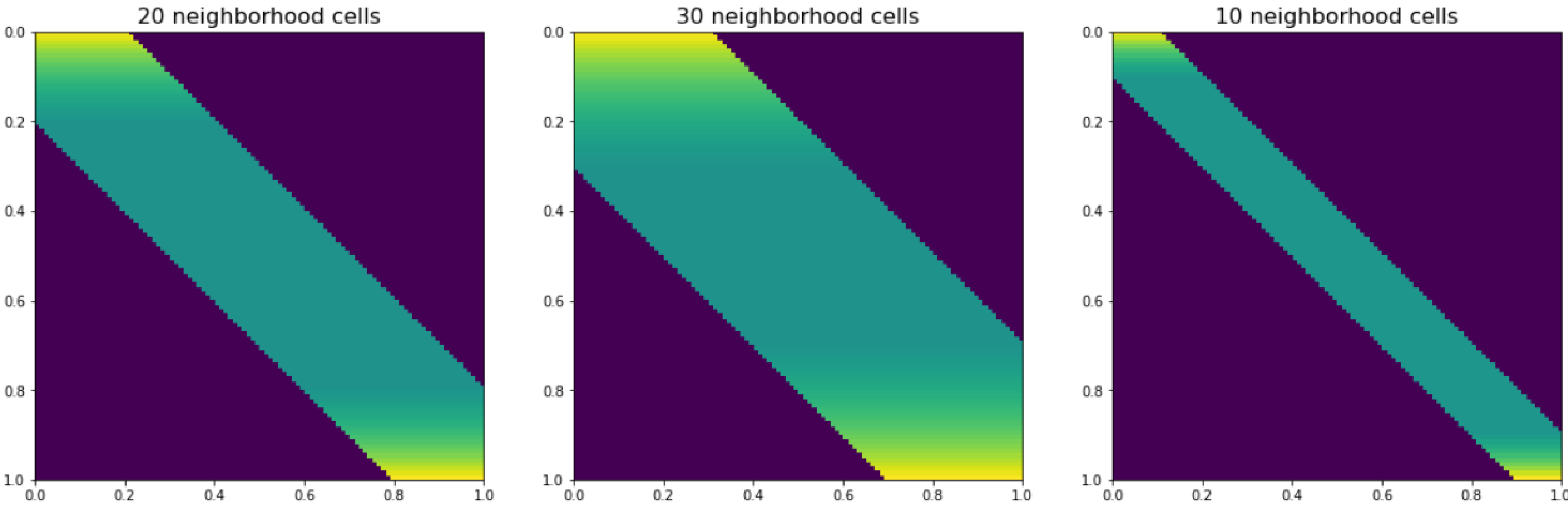
Given the updated mean we can then find update the covariance as:

$$\mathbf{Cov}_{n+1} = \frac{n}{1+n}\mathbf{Cov}_n + \frac{n}{(1+n)^2}(x_{n+1} - \bar{x}_{n+1})^T(x_{n+1} - \bar{x}_{n+1})$$

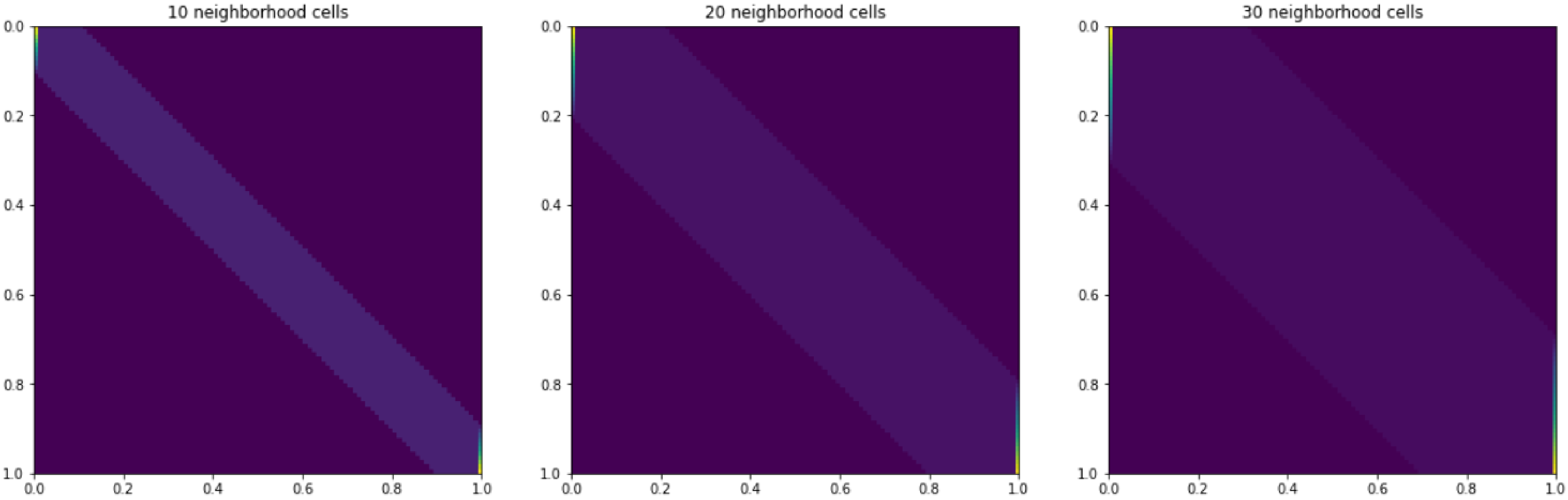
Experiment Parameters

Algorithm	Scenario	n_runs	n_global	n_local	d_thresh	w_rrt	alpha	w_gw
RRT-Filter	3-DOF	1	100	50	0.5	1.2	0.01	
GW-Filter	3-DOF	1	100	50	0.5			0.1
RRT	3-DOF	1	50	100		1	0.01	
GW	3-DOF	1	50	100	0.5			0.1
i.i.d.-Biased-Sampler	3-DOF	1	5,000					
RRT-Filter	7-DOF	1	1,000	100	0.5	1.2	0.01	
GW-Filter	7-DOF	1	1,000	100	0.5			0.05
RRT	7-DOF	1	1,000	100		1.2	0.01	
GW	7-DOF	1	1,000	100				0.05
i.i.d.-Biased-Sampler	7-DOF	1	100,000					
RRT-Projection	CC	100	100	50		0.5	0.01	
Gw-Projection	CC	100	1	5,000				0.5
i.i.d.-Biased-Sampler	CC	100	5,000					
Gw-Projection	OCC	100	100	50				0.5
Gw-Projection-filter	OCC	100	100	50	0.4			0.5
RRT-Projection	OCC	100	100	50		0.5	0.01	
RRT-Projection-Filter	OCC	100	100	50	0.4	0.5	0.01	
i.i.d.-Biased-Sampler	OCC	100	5,000					
Gw-Rejection-Filter	OCDC	100	100	50	0.4			0.5
Gw-Projection-filter	OCDC	100	100	50	0.4			0.5
RRT-Rejection-Filter	OCDC	100	100	50	0.4	0.5	0.01	
RRT-Projection-Filter	OCDC	100	100	50	0.4	0.5	0.01	
i.i.d.-Biased-Sampler	OCDC	100	5,000					
RRT-Rejection-Filter-Bandit	OCDC	100	100	50	0.4	0.5	0.01	
Gw-Rejection-Filter-Bandit	OCDC	100	100	50	0.4			0.5

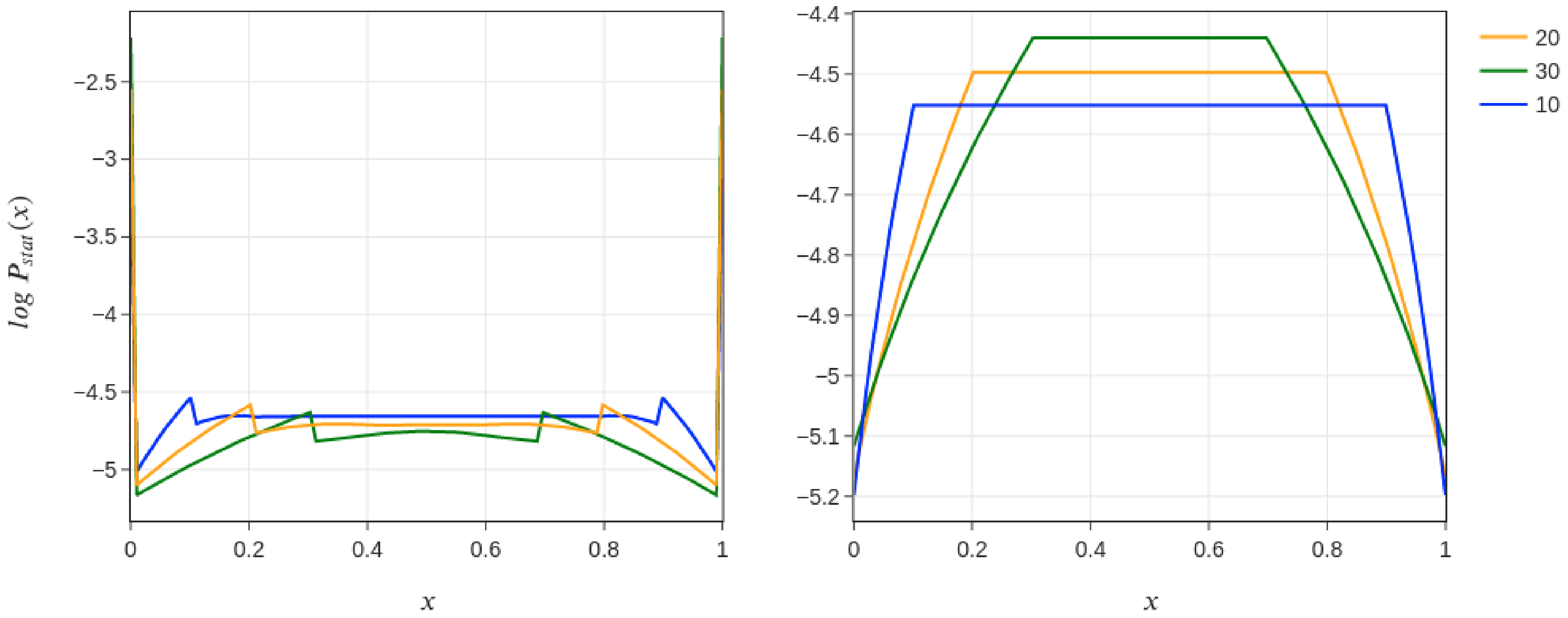
Cellular Sampler Grid-Walk-Rejection



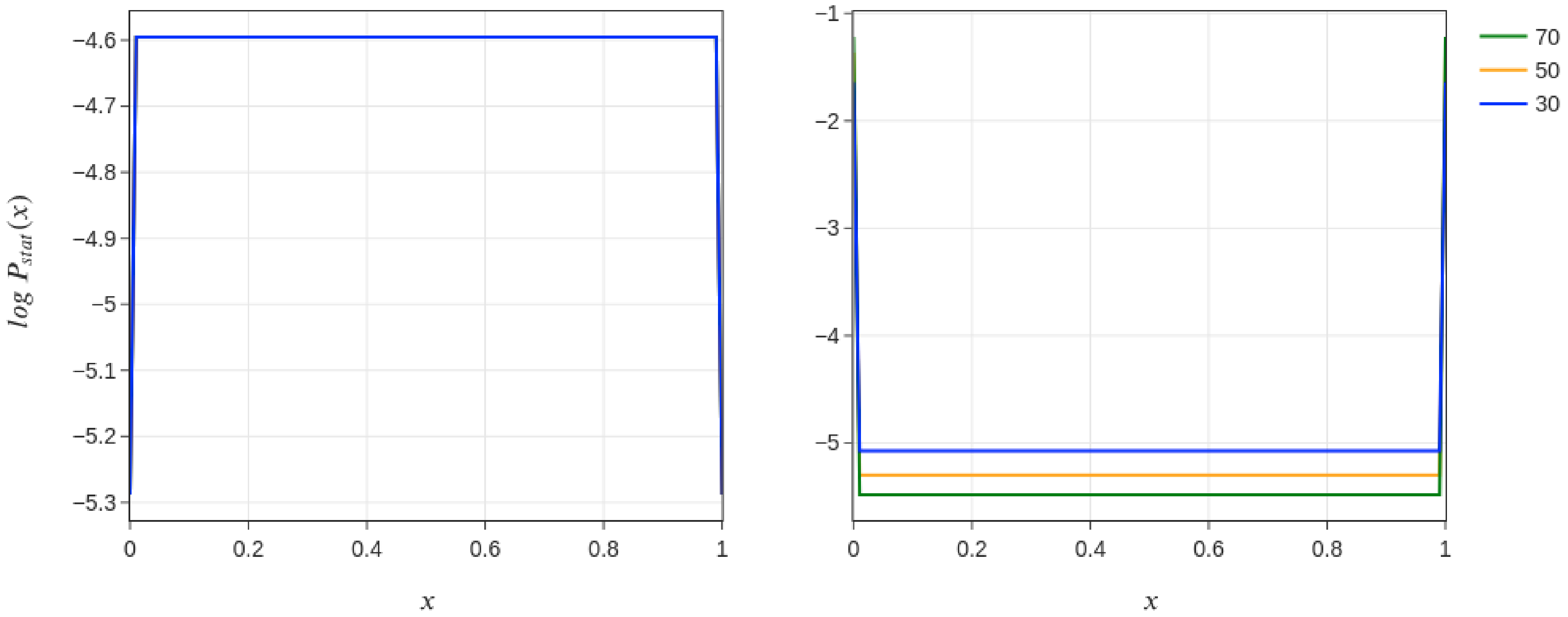
Cellular Sampler Grid-Walk-Projection



Cellular Sampler Grid-Walk



Cellular Sampler RRT



Off-Center-Disconnected

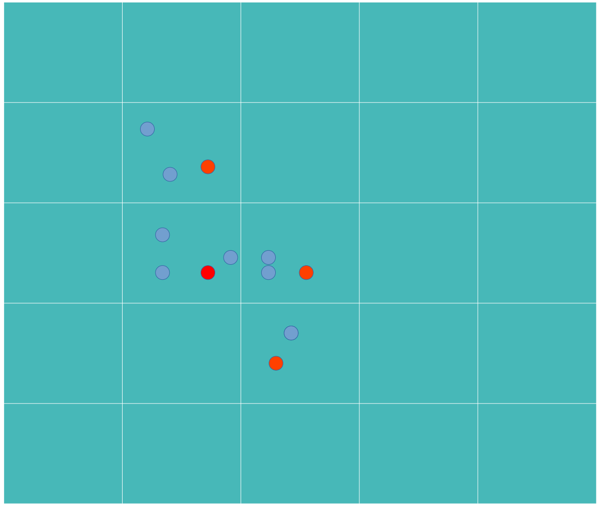
$$\mathcal{X} = \{x \in \mathbb{R}^3 : b_{\text{low}} \leq x \leq b_{\text{up}};\ x_1^2 + x_2^2 + x_3^2 = 1;\ g_l(x) \leq 0\ \forall\ l \in \{1,2,3\}\}$$

$$b_{\text{low}} = \begin{bmatrix} -3 \\ -2 \\ -4 \end{bmatrix},\ b_{\text{up}} = \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix}$$

$$\begin{aligned} g_1 &= -5x_2^2 - x_3 + 1.2 \\ g_2 &= -5x_3^2 - x_2 + 1.2 \\ g_3 &= -100x_1^2 - x_3 + 2 \end{aligned}$$

Reference Data Set Robotics

- Union of all Sampled points
- Bin Points
- Sample a point from each bin



About

Philip Oedi
340896
Computational Engineering Science
M.Sc.
philip.oedi@gmail.com

Examiner:
Prof. Marc Toussaint (LIS)
Prof. Jörg Krüger (IAT)

Supervisor:
Ph.D. Jung-Su Ha (LIS)
M.Sc. Joaquim Ortiz-Haro (LIS)