# Nonlinear approximation in bounded orthonormal product bases

Fabian Taubert

joint work with Lutz Kämmerer and Daniel Potts

Workshop on Mathematical Signal and Image Analysis

21.03.2023

UNIVERSITY OF TECHNOLOGY
IN THE EUROPEAN CAPITAL OF CULTURE
CHEMNITZ

## General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

### by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

### Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^d \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^d \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^d \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

### smooth function $f$

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### Truncation and approximation

$$S_I^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in I} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in I$
- suitable sparse index set $I \subset \mathbb{Z}^d$

## General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

### by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

### Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^{d} \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^{d} \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{d} \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

### smooth function $f$

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### Truncation and approximation

$$S_I^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in I} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in I$
- suitable sparse index set $I \subset \mathbb{Z}^d$

## General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

### by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

### Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^{d} \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^{d} \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{d} \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

### smooth function $f$

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### Truncation and approximation

$$S_I^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in I} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in I$
- suitable sparse index set $I \subset \mathbb{Z}^d$

## General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

### by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

### Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^d \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^d \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^d \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

### smooth function $f$

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### Truncation and approximation

$$S_I^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in I} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in I$
- suitable sparse index set $I \subset \mathbb{Z}^d$

## General aim in the Fourier setting

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathbb{T}^d, \mu)$

### by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

### Hilbert space $L_2(\mathbb{T}^d, \mu)$

domain $\mathbb{T}^d = \times_{j=1}^d \mathbb{T} \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^d \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^d \mathrm{e}^{2\pi \mathrm{i} k_j x_j}$

### smooth function $f$

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{Z}^d} c_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### Truncation and approximation

$$S_I^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in I} \hat{f}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}$$

### To compute

- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in I$
- suitable sparse index set $I \subset \mathbb{Z}^d$

## General aim for an arbitrary bounded orthonormal product basis (BOPB)

Approximation (by using samples) of a (smooth) high-dimensional function $f \in L_2(\mathcal{D}, \mu)$

### by using samples

black-box sampling, so we choose the sampling nodes $\boldsymbol{x}$ adaptively

### Hilbert space $L_2(\mathcal{D}, \mu)$

domain $\mathcal{D} = \times_{j=1}^{d} \mathcal{D}_j \subset \mathbb{R}^d$

measure $\mu = \times_{j=1}^{d} \mu_j$

basis $\Phi_{\boldsymbol{k}}(\boldsymbol{x}) = \prod_{j=1}^{d} \phi_{j,k_j}(x_j)$

### smooth function $f$

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in \mathbb{N}^d} c_{\boldsymbol{k}} \Phi_{\boldsymbol{k}}(\boldsymbol{x})$$

### Truncation and approximation

$$S_I^{\mathcal{A}} f(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in I} \hat{f}_{\boldsymbol{k}} \Phi_{\boldsymbol{k}}(\boldsymbol{x})$$

### To compute

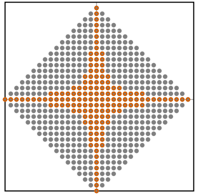
- approximated coefficients $\hat{f}_{\boldsymbol{k}} \approx c_{\boldsymbol{k}} \; \forall \boldsymbol{k} \in I$
- suitable sparse index set $I \subset \mathbb{N}^d$

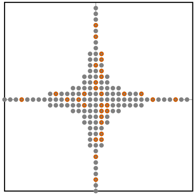
Problem: How to find a good, $s$-sparse index set $I$?

- first idea:
  - choose a large search space $\Gamma \supset I, |\Gamma| \gg |I|$
  - compute all coefficients
  - choose indices corresponding to the $s$ largest coefficients (absolute value)
- but: unfeasible in practice for large $\Gamma$ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

Problem: How to find a good, $s$-sparse index set $I$?

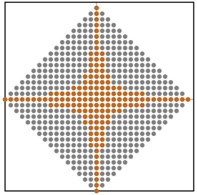
- first idea:
  - choose a large search space $\Gamma \supset I, |\Gamma| \gg |I|$
  - compute all coefficients
  - choose indices corresponding to the $s$ largest coefficients (absolute value)
- but: unfeasible in practice for large $\Gamma$ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

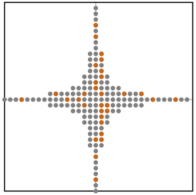




Problem: How to find a good, $s$-sparse index set $I$?

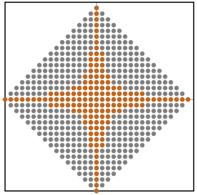
- first idea:
  - choose a large search space $\Gamma \supset I, |\Gamma| \gg |I|$
  - compute all coefficients
  - choose indices corresponding to the $s$ largest coefficients (absolute value)
- but: unfeasible in practice for large $\Gamma$ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

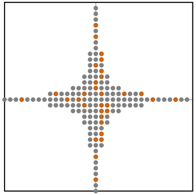




Problem: How to find a good, $s$-sparse index set $I$?

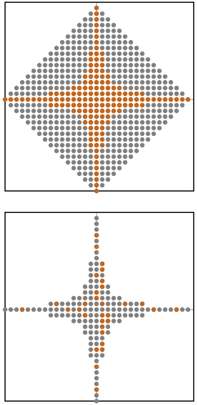
- first idea:
  - choose a large search space $\Gamma \supset I, |\Gamma| \gg |I|$
  - compute all coefficients
  - choose indices corresponding to the $s$ largest coefficients (absolute value)
- but: unfeasible in practice for large $\Gamma$ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach





Problem: How to find a good, $s$-sparse index set $I$?

- first idea:
  - choose a large search space $\Gamma \supset I, |\Gamma| \gg |I|$
  - compute all coefficients
  - choose indices corresponding to the $s$ largest coefficients (absolute value)
- but: unfeasible in practice for large $\Gamma$ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

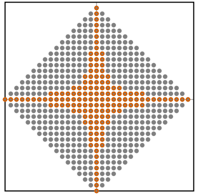


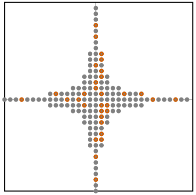
Question: Can we ensure to detect all important indices?

Problem: How to find a good, $s$-sparse index set $I$?

- first idea:
  - choose a large search space $\Gamma \supset I, |\Gamma| \gg |I|$
  - compute all coefficients
  - choose indices corresponding to the $s$ largest coefficients (absolute value)
- but: unfeasible in practice for large $\Gamma$ ($\rightarrow$ "curse of dimensionality")
- better idea: use a dimension-incremental approach

Question: Can we ensure to detect all important indices? $\rightarrow$ YES!





Previous works on sparse, high-dimensional approximation:

- our dimension-incremental approach in the Fourier setting:
  - sparse FFT using rank-1 lattices
    [Potts, Volkmer '16], [Kämmerer, Potts, Volkmer '21], [Kämmerer, Krahmer, Volkmer '22]
  - application to differential equations with high-dimensional random coefficients
    [Bochmann, Kämmerer, Potts '20], [Kämmerer, Potts, T. '22]
- other approximation methods in the Fourier setting:
  e.g. [Iwen '13], [Indyk, Kapralov '14], [Choi, Christlieb, Wang '19], ...
- sparse high-dimensional approximation in more general bases:
  - sparse polynomial chaos expansions
    literature survey: [Lüthen, Marelli, Sudret '21], basis-adaptive: [Lüthen, Marelli, Sudret '21]
  - compressive sensing approach for BOPB
    [Choi, Iwen, Krahmer '20], [Choi, Iwen, Volkmer '21]
  - sparse polynomial approximation via least squares and compressed sensing
    [Adcock, Brugiapaglia, Webster '22]
- ...

Previous works on sparse, high-dimensional approximation:

- our dimension-incremental approach in the Fourier setting:
  - sparse FFT using rank-1 lattices
    [Potts, Volkmer '16], [Kämmerer, Potts, Volkmer '21], [Kämmerer, Krahmer, Volkmer '22]
  - application to differential equations with high-dimensional random coefficients
    [Bochmann, Kämmerer, Potts '20], [Kämmerer, Potts, T. '22]
- other approximation methods in the Fourier setting:
  e.g. [Iwen '13], [Indyk, Kapralov '14], [Choi, Christlieb, Wang '19], ...
- sparse high-dimensional approximation in more general bases:
  - sparse polynomial chaos expansions
    literature survey: [Lüthen, Marelli, Sudret '21], basis-adaptive: [Lüthen, Marelli, Sudret '21]
  - compressive sensing approach for BOPB
    [Choi, Iwen, Krahmer '20], [Choi, Iwen, Volkmer '21]
  - sparse polynomial approximation via least squares and compressed sensing
    [Adcock, Brugiapaglia, Webster '22]
- ...

Previous works on sparse, high-dimensional approximation:

- our dimension-incremental approach in the Fourier setting:
  - sparse FFT using rank-1 lattices
    [Potts, Volkmer '16], [Kämmerer, Potts, Volkmer '21], [Kämmerer, Krahmer, Volkmer '22]
  - application to differential equations with high-dimensional random coefficients
    [Bochmann, Kämmerer, Potts '20], [Kämmerer, Potts, T. '22]
- other approximation methods in the Fourier setting:
  e.g. [Iwen '13], [Indyk, Kapralov '14], [Choi, Christlieb, Wang '19], ...
- sparse high-dimensional approximation in more general bases:
  - sparse polynomial chaos expansions
    literature survey: [Lüthen, Marelli, Sudret '21], basis-adaptive: [Lüthen, Marelli, Sudret '21]
  - compressive sensing approach for BOPB
    [Choi, Iwen, Krahmer '20], [Choi, Iwen, Volkmer '21]
  - sparse polynomial approximation via least squares and compressed sensing
    [Adcock, Brugiapaglia, Webster '22]
- ...

## Projected coefficients for the dimensions $\{1,\dots,t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{k}\in\mathbb{N}^t,\ \forall\tilde{\boldsymbol{x}}\in\mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k},\boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \ldots = \sum_{\boldsymbol{h}\in\mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})}\Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

## Example: Fourier setting with $d=3$ and $t=2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) := \int_{\mathbb{T}^2} f(\boldsymbol{\xi},\tilde{x})\,\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\xi}}\,\mathrm{d}\boldsymbol{\xi} = \sum_{h\in\mathbb{Z}} c_{(\boldsymbol{k},h)}\mathrm{e}^{2\pi\mathrm{i}h\tilde{x}} \qquad \forall\boldsymbol{k}\in\mathbb{Z}^2,\ \forall\tilde{x}\in\mathbb{T}$$

→ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u}\subset\{1,\dots,d\}$.

## Projected coefficients for the dimensions $\{1,\dots,t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{k}\in\mathbb{N}^t,\ \forall\tilde{\boldsymbol{x}}\in\mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k},\boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \ldots = \sum_{\boldsymbol{h}\in\mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})}\,\Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

## Example: Fourier setting with $d=3$ and $t=2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) := \int_{\mathbb{T}^2} f(\boldsymbol{\xi},\tilde{x})\,\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\xi}}\,\mathrm{d}\boldsymbol{\xi} = \sum_{h\in\mathbb{Z}} c_{(\boldsymbol{k},h)}\mathrm{e}^{2\pi\mathrm{i}h\tilde{x}} \qquad \forall\boldsymbol{k}\in\mathbb{Z}^2,\ \forall\tilde{x}\in\mathbb{T}$$

$\rightarrow$ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u}\subset\{1,\dots,d\}$.

## Projected coefficients for the dimensions $\{1,\dots,t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{k}\in\mathbb{N}^t,\ \forall\tilde{\boldsymbol{x}}\in\mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k},\boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \dots = \sum_{\boldsymbol{h}\in\mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})}\,\Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

## Example: Fourier setting with $d=3$ and $t=2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) := \int_{\mathbb{T}^2} f(\boldsymbol{\xi},\tilde{x})\,\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\xi}}\,\mathrm{d}\boldsymbol{\xi} = \sum_{h\in\mathbb{Z}} c_{(\boldsymbol{k},h)}\mathrm{e}^{2\pi\mathrm{i}h\tilde{x}} \qquad \forall\boldsymbol{k}\in\mathbb{Z}^2,\ \forall\tilde{x}\in\mathbb{T}$$

$\rightarrow$ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u}\subset\{1,\dots,d\}$.

## Projected coefficients for the dimensions $\{1,\dots,t\}$

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi}) \qquad \forall \boldsymbol{k}\in\mathbb{N}^t,\ \forall\tilde{\boldsymbol{x}}\in\mathcal{D}_{\{t+1,\dots,d\}}$$

Projected coefficients are an indicator for the importance of the indices $(\boldsymbol{k},\boldsymbol{h})$ with arbitrary $\boldsymbol{h}$, since

$$c_{\mathfrak{u},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) = \ldots = \sum_{\boldsymbol{h}\in\mathbb{N}^{d-t}} c_{(\boldsymbol{k},\boldsymbol{h})}\,\Phi_{\{t+1,\dots,d\},(\boldsymbol{k},\boldsymbol{h})}(\tilde{\boldsymbol{x}}).$$

## Example: Fourier setting with $d=3$ and $t=2$

$$c_{\{1,2\},\boldsymbol{k}}(\tilde{x}) := \int_{\mathbb{T}^2} f(\boldsymbol{\xi},\tilde{x})\,\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{\xi}}\,\mathrm{d}\boldsymbol{\xi} = \sum_{h\in\mathbb{Z}} c_{(\boldsymbol{k},h)}\mathrm{e}^{2\pi\mathrm{i}h\tilde{x}} \qquad \forall\boldsymbol{k}\in\mathbb{Z}^2,\ \forall\tilde{x}\in\mathbb{T}$$

$\rightarrow$ Works analogously for single dimensions $\{t\}$ and arbitrary $\mathfrak{u}\subset\{1,\dots,d\}$.

## Key idea of the dimension-incremental algorithm

- Construct a candidate set with possible indices in the current dimension(s).
- Compute approximations of the projected coefficients (using samples).
- Build the index set: $\begin{cases} \text{projected coefficient large}: & \text{keep index} \\ \text{projected coefficient small}: & \text{neglect index} \end{cases}$

$\rightarrow$ Increase the dimension by combining different detected index sets of smaller dimension.

Approximation of the projected coefficients via, e.g., cubature formulas (nodes $\boldsymbol{\xi}_j$, weights $w_j$)

$$c_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\dots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\dots,t\}}(\boldsymbol{\xi})$$
$$\approx \sum_{j=1}^{M} w_j f(\boldsymbol{\xi}_j,\tilde{\boldsymbol{x}})\overline{\Phi_{\{1,\dots,t\},\boldsymbol{k}}(\boldsymbol{\xi}_j)} =: \hat{f}_{\{1,\dots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}})$$

## Key idea of the dimension-incremental algorithm

- Construct a candidate set with possible indices in the current dimension(s).
- Compute approximations of the projected coefficients (using samples).
- Build the index set: $\begin{cases} \text{projected coefficient large :} & \text{keep index} \\ \text{projected coefficient small :} & \text{neglect index} \end{cases}$

$\rightarrow$ Increase the dimension by combining different detected index sets of smaller dimension.

Approximation of the projected coefficients via, e.g., cubature formulas (nodes $\boldsymbol{\xi}_j$, weights $w_j$)

$$c_{\{1,\ldots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}}) := \int_{\mathcal{D}_{\{1,\ldots,t\}}} f(\boldsymbol{\xi},\tilde{\boldsymbol{x}})\,\overline{\Phi_{\{1,\ldots,t\},\boldsymbol{k}}(\boldsymbol{\xi})}\,\mathrm{d}\mu_{\{1,\ldots,t\}}(\boldsymbol{\xi})$$

$$\approx \sum_{j=1}^{M} w_j f(\boldsymbol{\xi}_j,\tilde{\boldsymbol{x}})\overline{\Phi_{\{1,\ldots,t\},\boldsymbol{k}}(\boldsymbol{\xi}_j)} =: \hat{f}_{\{1,\ldots,t\},\boldsymbol{k}}(\tilde{\boldsymbol{x}})$$

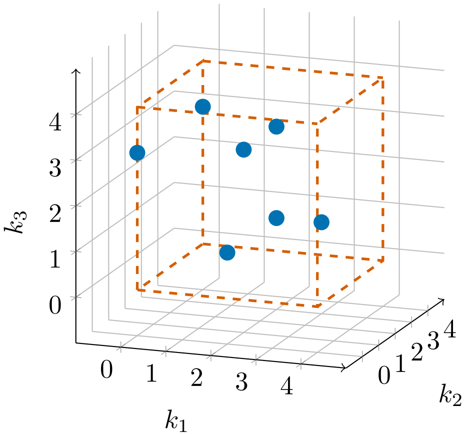



Figure: The desired but unknown index set $I$ and the search space $\Gamma = \{0, 1, 2, 3, 4\}^3$.

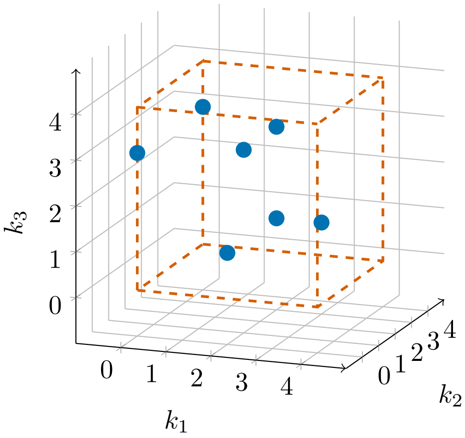



Figure: The desired but unknown index set $I$ and the search space $\Gamma = \{0, 1, 2, 3, 4\}^3$.

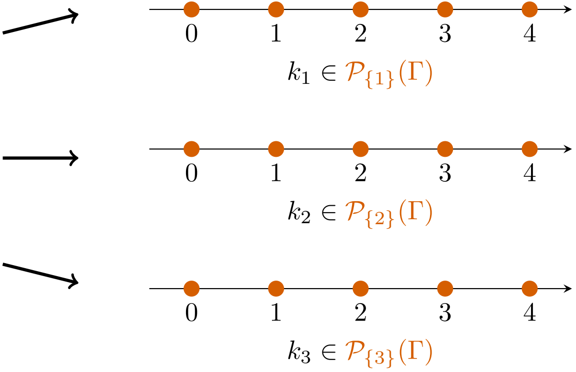



Figure: The one-dimensional candidate sets $\mathcal{P}_{\{j\}}(\Gamma)$.

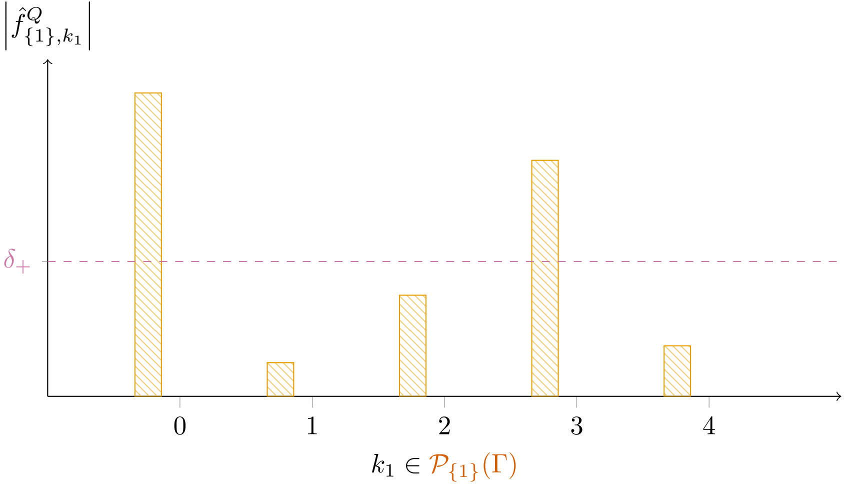



Figure: The one-dimensional detection step in the first dimension with $r = 3$ detection iterations.

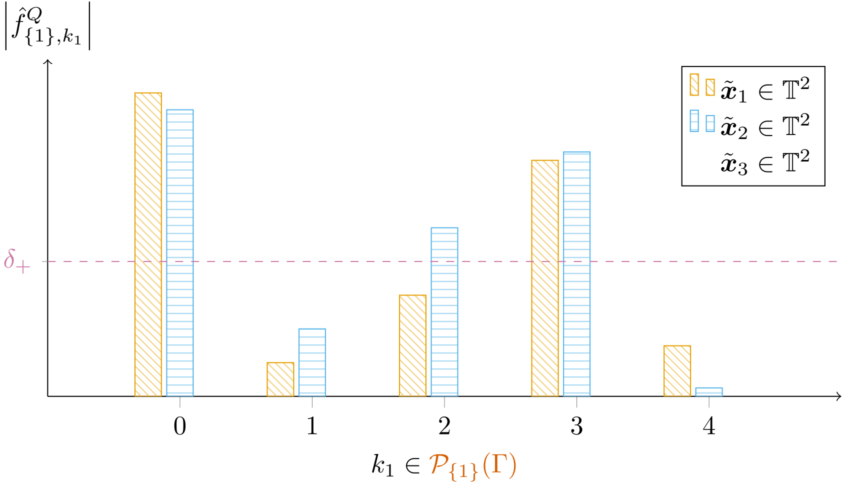



Figure: The one-dimensional detection step in the first dimension with $r = 3$ detection iterations.

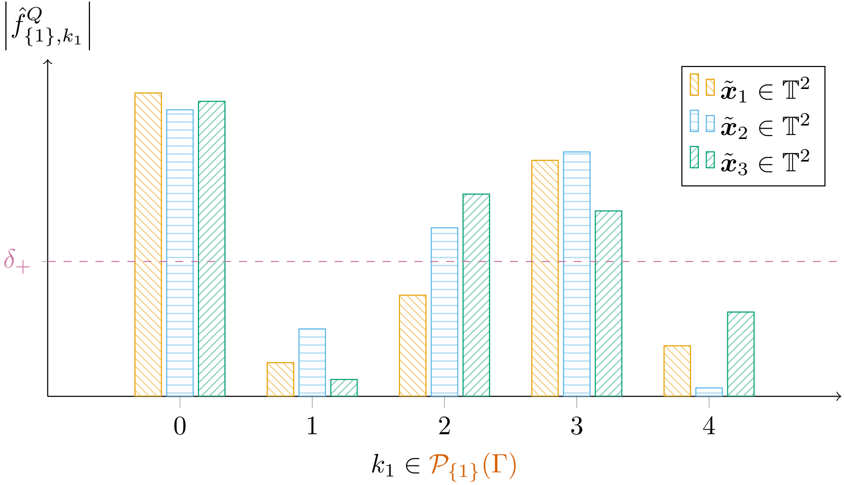



Figure: The one-dimensional detection step in the first dimension with $r = 3$ detection iterations.

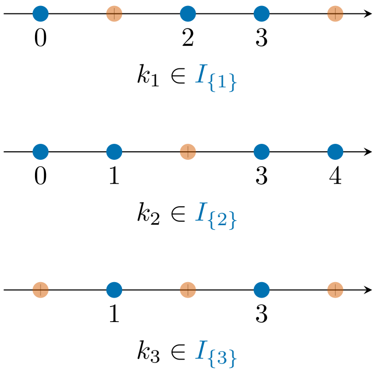



Figure: The one-dimensional index sets $I_{\{j\}}$.

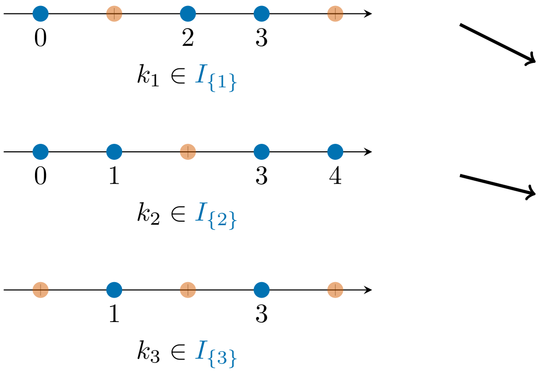



Figure: The one-dimensional index sets $I_{\{j\}}$.

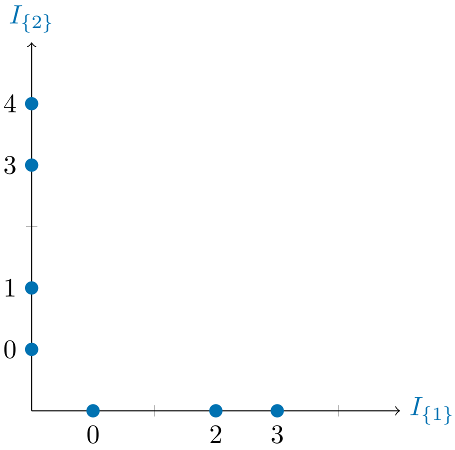



Figure: The two-dimensional candidate set $I_{\{1\}} \times I_{\{2\}}$.

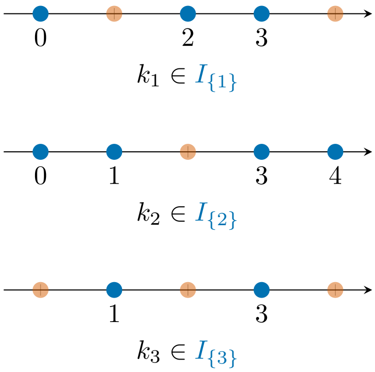



Figure: The one-dimensional index sets $I_{\{j\}}$.

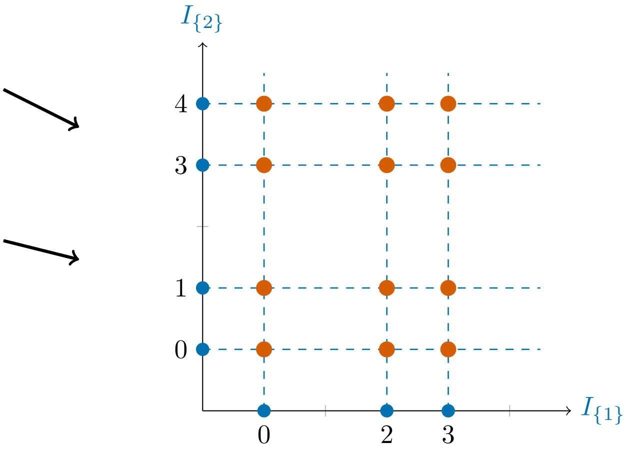



Figure: The two-dimensional candidate set $I_{\{1\}} \times I_{\{2\}}$.

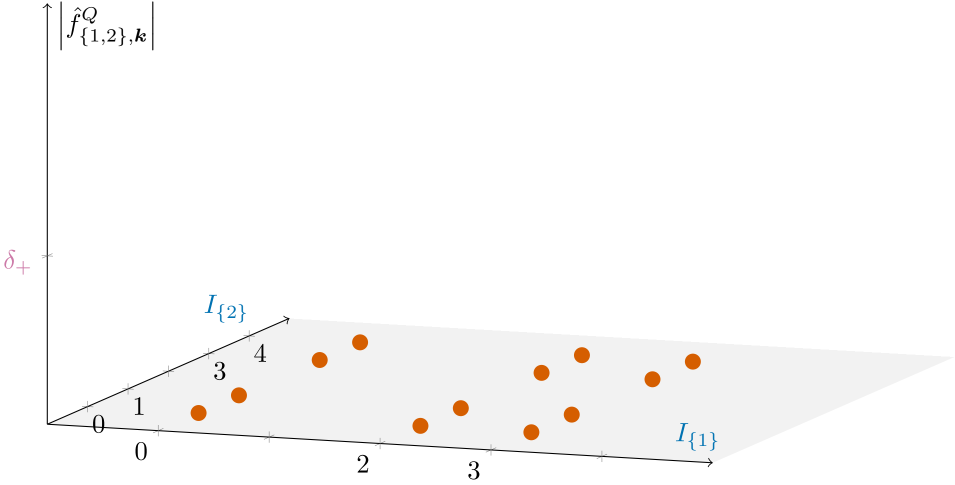



Figure: The two-dimensional detection step in the dimensions $\{1, 2\}$. (Only $r = 1$ detection iteration shown.)

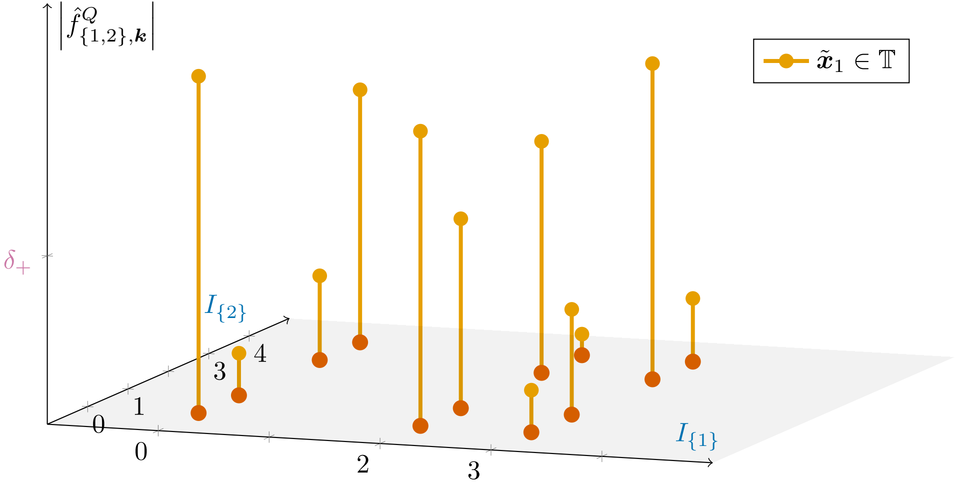



Figure: The two-dimensional detection step in the dimensions $\{1, 2\}$. (Only $r = 1$ detection iteration shown.)

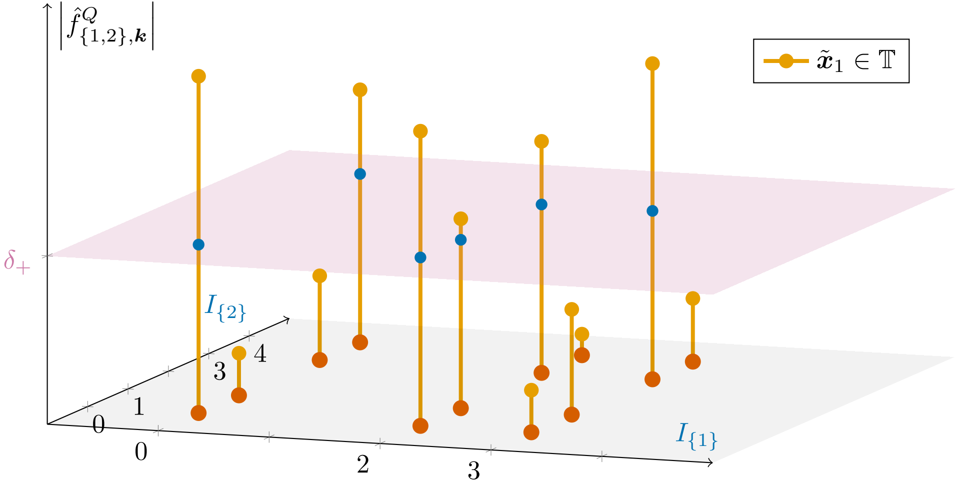



Figure: The two-dimensional detection step in the dimensions $\{1,2\}$. (Only $r=1$ detection iteration shown.)

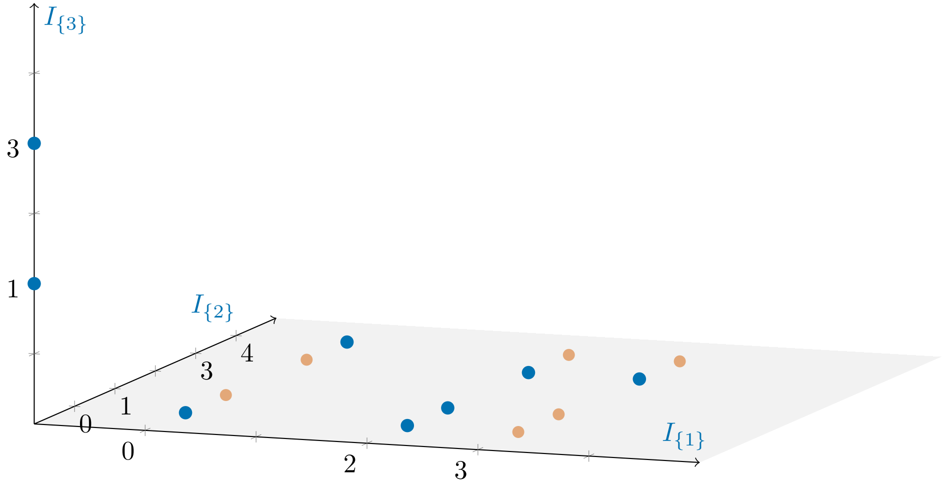



Figure: The two-dimensional index set $I_{\{1,2\}}$ and the three-dimensional candidate set $I_{\{1,2\}} \times I_{\{3\}}$.

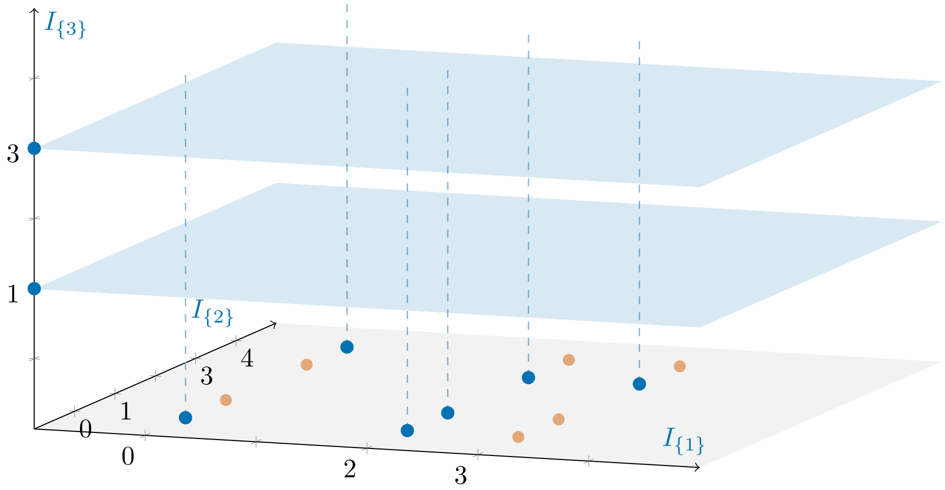



Figure: The two-dimensional index set $I_{\{1,2\}}$ and the three-dimensional candidate set $I_{\{1,2\}} \times I_{\{3\}}$.

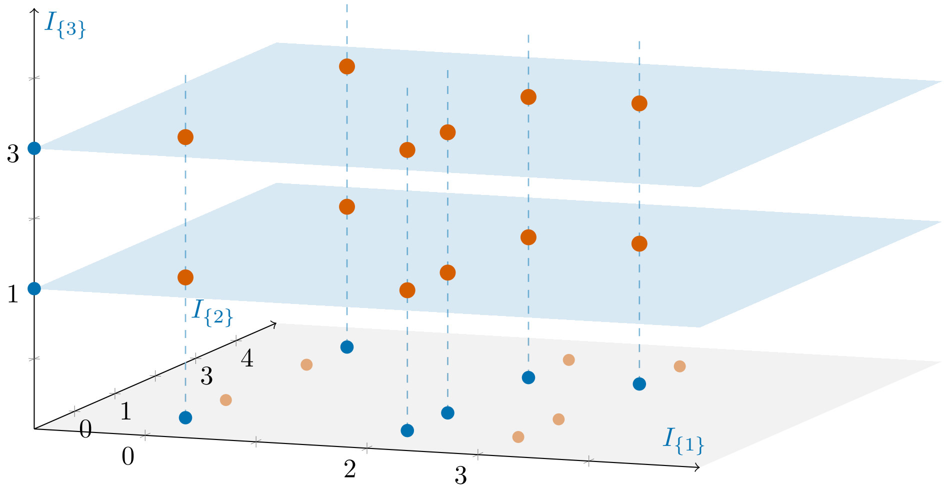



Figure: The two-dimensional index set $I_{\{1,2\}}$ and the three-dimensional candidate set $I_{\{1,2\}} \times I_{\{3\}}$.

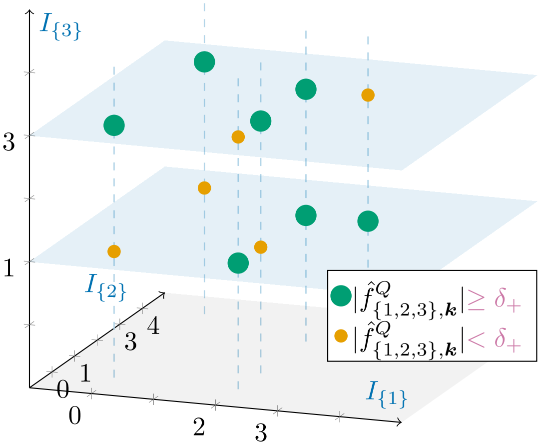



Figure: The three-dimensional detection step with the detected index set $I_{\{1,2,3\}}$.

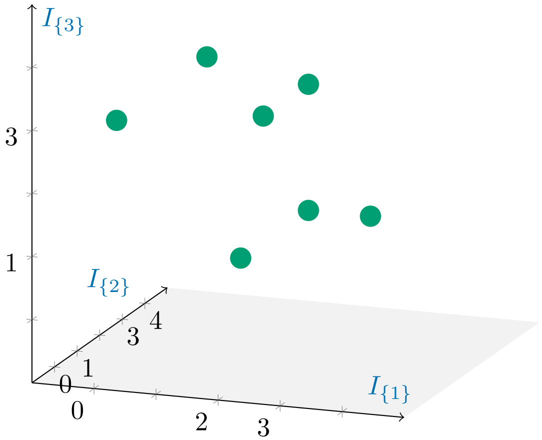



Figure: The three-dimensional detection step with the detected index set $I_{\{1,2,3\}}$.

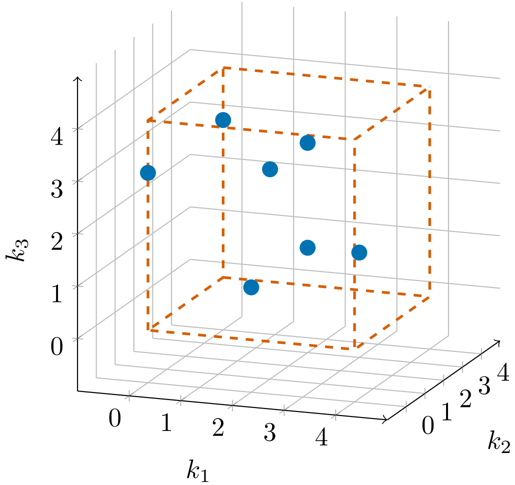



Figure: The correctly detected index set $I = I_{\{1,2,3\}}$.

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

**Theorem [Kämmerer, Potts, T. '22]**

- given: function $f \in L_2(\mathcal{D},\mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space $\Gamma \supset I_{3\delta}$
  - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k}\in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
  - number of detection iterations: $r \geq \left(1 + \frac{3}{2}B^2\,|I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h}\notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d\,|I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1-\varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!
**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

## Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space $\Gamma \supset I_{3\delta}$
  - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k} \in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
  - number of detection iterations: $r \geq \left(1 + \frac{3}{2} B^2 |I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h} \notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d\,|I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1-\varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!
**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

## Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space $\Gamma \supset I_{3\delta}$
  - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k} \in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
  - number of detection iterations: $r \geq \left(1 + \frac{3}{2} B^2 |I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h} \notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d\,|I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1 - \varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!
**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

## Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D},\mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space $\Gamma \supset I_{3\delta}$
  - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k}\in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
  - number of detection iterations: $r \geq \left(1 + \frac{3}{2}B^2 |I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h}\notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d\,|I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1-\varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!
**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

## Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space $\Gamma \supset I_{3\delta}$
  - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k} \in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
  - number of detection iterations: $r \geq \left(1 + \frac{3}{2} B^2 |I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h} \notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d\,|I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1-\varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!
**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

## Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0, 1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space $\Gamma \supset I_{3\delta}$
  - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k} \in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
  - number of detection iterations: $r \geq \left(1 + \frac{3}{2} B^2 |I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h} \notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d\,|I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1 - \varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!
**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

## Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
    - search space $\Gamma \supset I_{3\delta}$
    - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k} \in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
    - number of detection iterations: $r \geq \left(1 + \frac{3}{2} B^2 |I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h} \notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d\,|I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1-\varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!

**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

Previously: Only recovery results for sparse trigonometric polynomials.
Now:

## Theorem [Kämmerer, Potts, T. '22]

- given: function $f \in L_2(\mathcal{D}, \mu)$, BOPB constant $B$, threshold $\delta > 0$, failure prob. $\varepsilon \in (0,1)$
- index set: $I_{3\delta} := \{\boldsymbol{k} \in \mathbb{N}^d : |c_{\boldsymbol{k}}| \geq 3\delta\}$
- cubature method: bound on proj. error $\delta_\Psi < \delta$, weight constant $C > 0$, failure prob. $\leq \varepsilon/(3d)$
- parameters:
  - search space $\Gamma \supset I_{3\delta}$
  - detection threshold $\delta_+ < 3B^{-1} \min_{\boldsymbol{k} \in I_{3\delta}} |c_{\boldsymbol{k}}| - \delta_\Psi$
  - number of detection iterations: $r \geq \left(1 + \frac{3}{2} B^2 \, |I_{3\delta}| + \frac{B^3 C}{2\delta} \sum_{\boldsymbol{h} \notin I_{3\delta}} |c_{\boldsymbol{h}}|\right) \log \frac{3d \, |I_{3\delta}|}{\varepsilon}$
- **Then, with probability $1 - \varepsilon$, the output index set $I$ of the Algorithm contains $I_{3\delta}$.**

**Improvements**: works for **function approximation** in **any** BOPB!
**Open questions**: How to include a cut-off parameter $s$? How to show a better bound on $r$?

## Test function (from [Potts, Volkmer '16], [Kämmerer, Krahmer, Volkmer '21])

$$f(\boldsymbol{x}) := \prod_{j\in\{1,3,8\}} N_2(x_j) + \prod_{j\in\{2,5,6,10\}} N_4(x_j) + \prod_{j\in\{4,7,9\}} N_6(x_j)$$

$f \in L_2(\mathbb{T}^{10}, \mu)$, with the B-Spline of order $m \in \mathbb{N}$

$$N_m(x) := C_m \sum_{k\in\mathbb{Z}} \operatorname{sinc}\left(\frac{\pi}{m}k\right)^m (-1)^k \mathrm{e}^{2\pi \mathrm{i} k x}$$

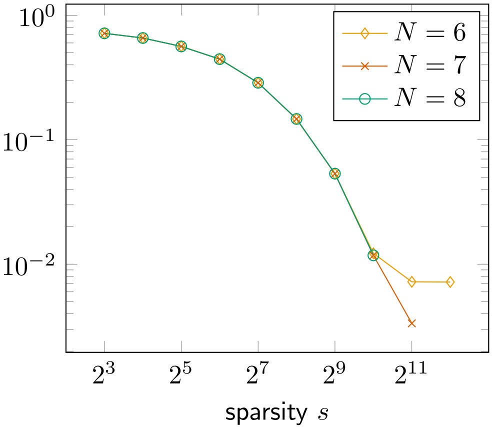
## Parameters

- hyperbolic cross search space:
  $\Gamma = \left\{\boldsymbol{k} \in \mathbb{Z}^d : \prod_{j=1}^d \max(1, 2\,|k_j|) \leq 2^N\right\}$
- detection iterations $r = 5$
- detection threshold $\delta_+ = 10^{-12}$

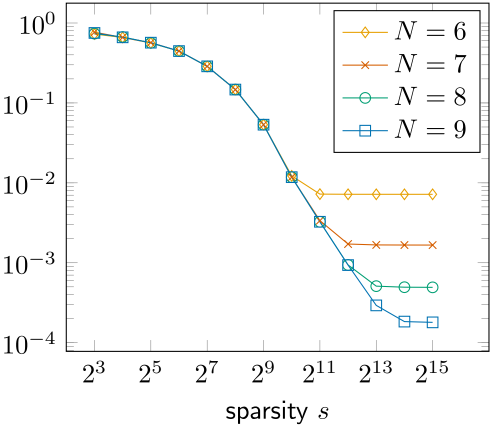
## Approximation error ("How good is $I$?")

- rel. $L_2(\mathbb{T}^{10}, \mu)$ error:

$$\frac{\|f - S_I f\|_{L_2(\mathbb{T}^{10})}}{\|f\|_{L_2(\mathbb{T}^{10})}}$$

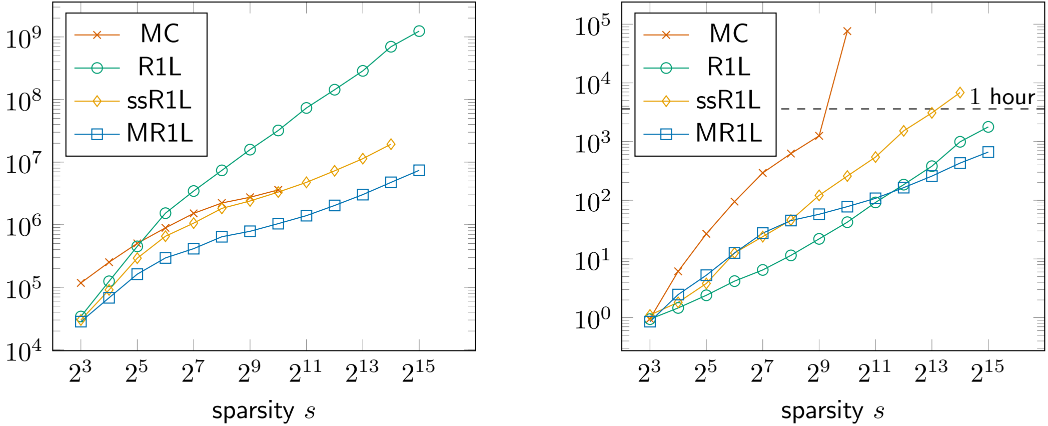



(a) approx. error for MC




(b) approx. error for R1L, ssR1L & MR1L

Figure: Approximation results for the 10-dimensional periodic test function




(a) amount of samples for $N = 8$

(b) computation time (in seconds) for $N = 8$

Figure: Approximation results for the 10-dimensional periodic test function

## Test function (from [Volkmer '17], [Potts, Volkmer '17])

$$f(\boldsymbol{x}) := \prod_{j\in\{1,3,4,7\}} B_2(x_j) + \prod_{j\in\{2,5,6,8,9\}} B_4(x_j)$$

$f \in L_2([-1,1]^9, \mu_{\mathrm{Cheb}})$, with $B_2$ and $B_4$ shifted, scaled and dilated B-Splines of order $2$ and $4$

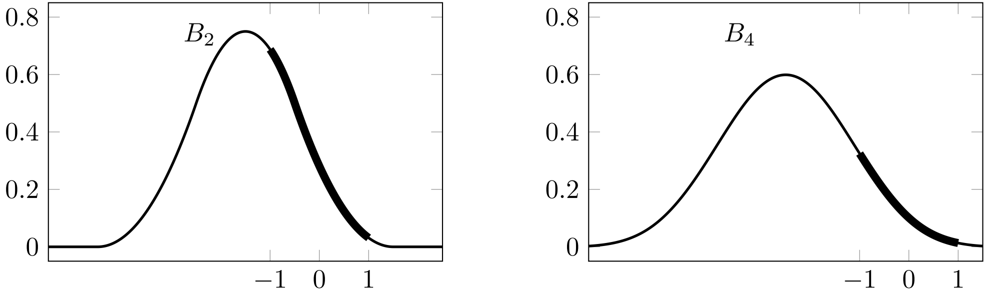
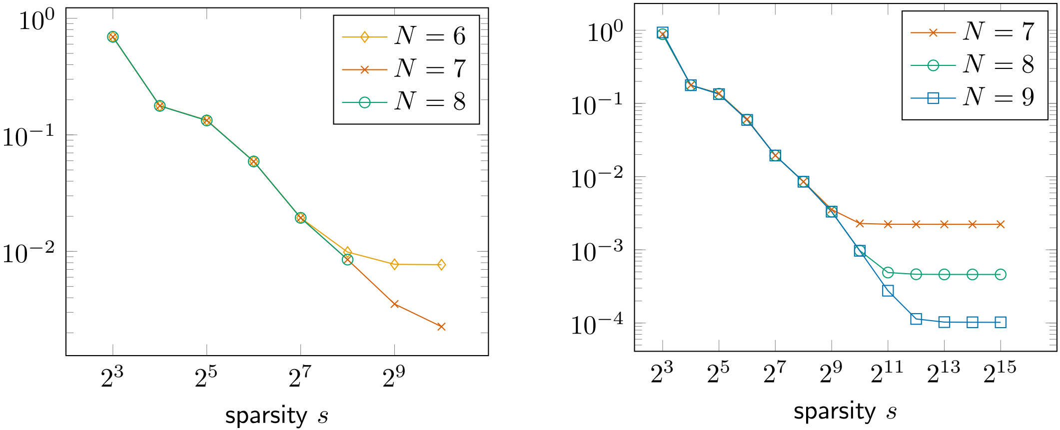



Figure: B-Splines $B_2$ and $B_4$ and the considered domain $[-1,1]$

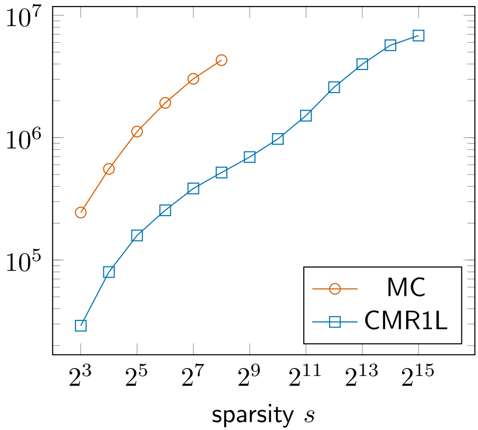



(a) approx. error for MC

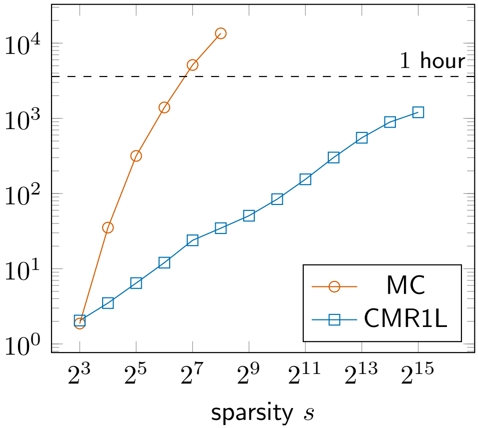


(b) approx. error for CMR1L

Figure: Approximation results for the 9-dimensional non-periodic test function



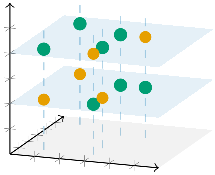

(a) amount of samples for $N = 8$




(b) computation time (in seconds) for $N = 8$

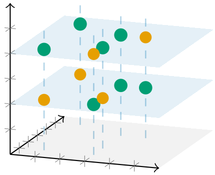
Figure: Approximation results for the $9$-dimensional non-periodic test function

- What did we do?
  - generalization of the dimension-incremental method to arbitrary BOPB
  - generalized algorithm also works
    - for various search spaces $\Gamma$
    - with other dimension-incremental strategies (dyadic, data-driven, ...)
    - ...



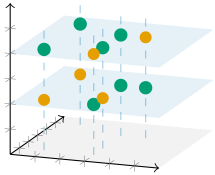
- What did we show?
  - first theoretical detection guarantee of the dimension-incremental method for function approximation
  - proof technique can be generalized for other reconstruction methods
- What did we see?
  - promising numerical tests with good approximations
  - efficiency is highly dependent on the reconstruction method
- Open problems?
  - Improved theoretical bounds on number of detection iterations $r$?
  - Theoretical results for the cut-off (sparsity $s$)?
  - Efficient reconstruction methods for various BOPB?

- **What did we do?**
  - generalization of the dimension-incremental method to arbitrary BOPB
  - generalized algorithm also works
    - for various search spaces $\Gamma$
    - with other dimension-incremental strategies (dyadic, data-driven, ...)
    - ...

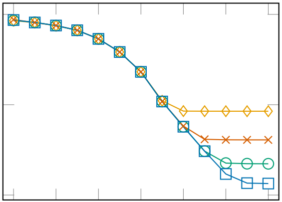


- **What did we show?**
  - first theoretical detection guarantee of the dimension-incremental method for function approximation
  - proof technique can be generalized for other reconstruction methods
- **What did we see?**
  - promising numerical tests with good approximations
  - efficiency is highly dependent on the reconstruction method
- **Open problems?**
  - Improved theoretical bounds on number of detection iterations $r$?
  - Theoretical results for the cut-off (sparsity $s$)?
  - Efficient reconstruction methods for various BOPB?

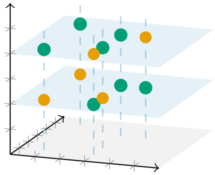
- What did we do?
  - generalization of the dimension-incremental method to arbitrary BOPB
  - generalized algorithm also works
    - for various search spaces $\Gamma$
    - with other dimension-incremental strategies (dyadic, data-driven, ...)
    - ...



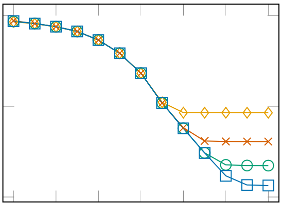
- What did we show?
  - first theoretical detection guarantee of the dimension-incremental method for function approximation
  - proof technique can be generalized for other reconstruction methods
- What did we see?
  - promising numerical tests with good approximations
  - efficiency is highly dependent on the reconstruction method
- Open problems?
  - Improved theoretical bounds on number of detection iterations $r$?
  - Theoretical results for the cut-off (sparsity $s$)?
  - Efficient reconstruction methods for various BOPB?



- **What did we do?**
  - generalization of the dimension-incremental method to arbitrary BOPB
  - generalized algorithm also works
    - for various search spaces $\Gamma$
    - with other dimension-incremental strategies (dyadic, data-driven, ...)
    - ...



- **What did we show?**
  - first theoretical detection guarantee of the dimension-incremental method for function approximation
  - proof technique can be generalized for other reconstruction methods
- **What did we see?**
  - promising numerical tests with good approximations
  - efficiency is highly dependent on the reconstruction method



- **Open problems?**
  - Improved theoretical bounds on number of detection iterations $r$?
  - Theoretical results for the cut-off (sparsity $s$)?
  - Efficient reconstruction methods for various BOPB?

- Lutz Kämmerer, Daniel Potts, Fabian Taubert
  *Nonlinear approximation in bounded orthonormal product bases*
  ArXiv e-prints, 2022. arXiv:2211.06071 [math.NA]
- Felix Bartel, Fabian Taubert
  *Nonlinear approximation with Subsampled Rank-1 Lattices*
  In preparation, 2023

- Lutz Kämmerer, Daniel Potts, Fabian Taubert
  *Nonlinear approximation in bounded orthonormal product bases*
  ArXiv e-prints, 2022. arXiv:2211.06071 [math.NA]
- Felix Bartel, Fabian Taubert
  *Nonlinear approximation with Subsampled Rank-1 Lattices*
  In preparation, 2023


**Thank you for your attention!**

**Questions? Ideas? Suggestions?**