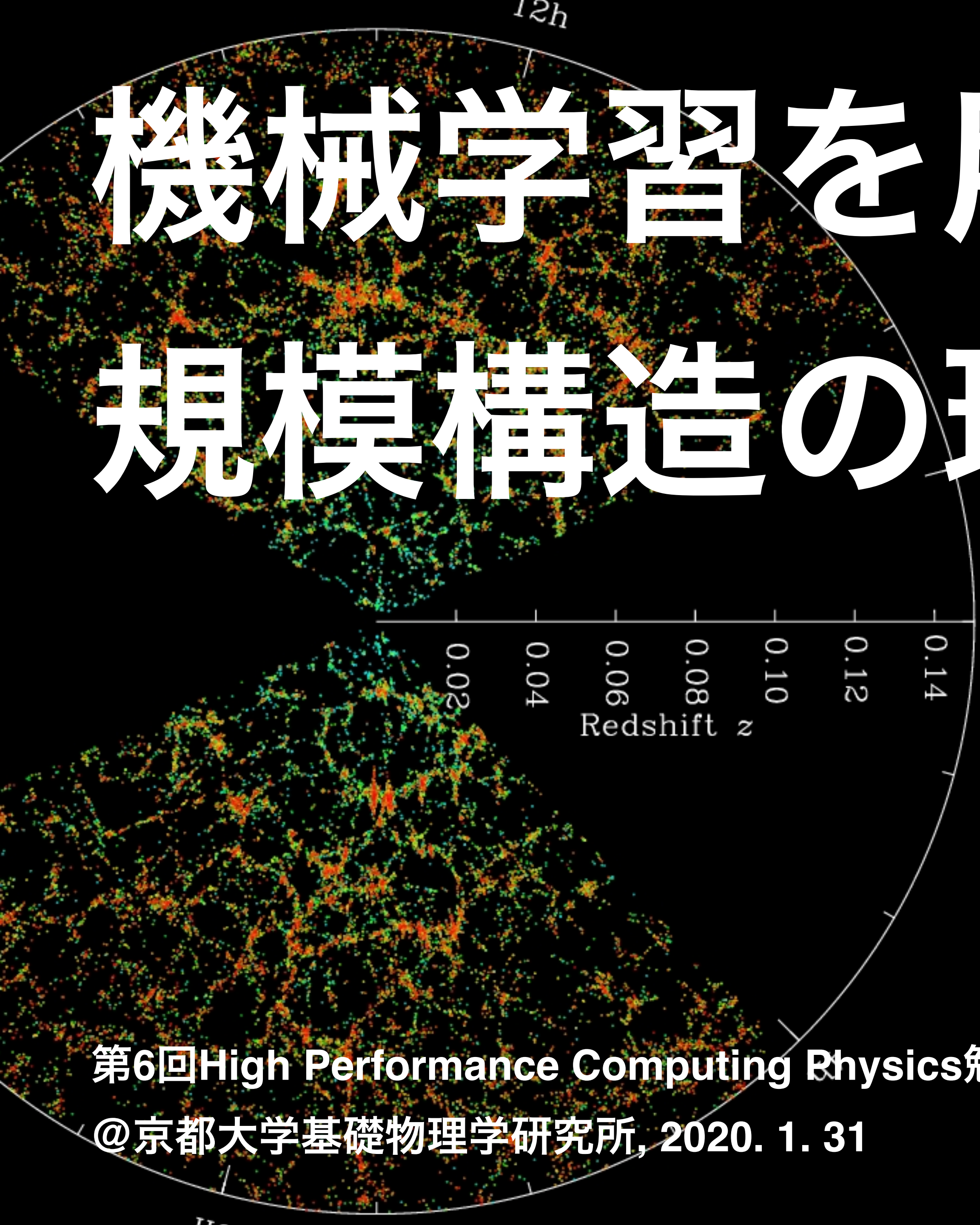


A visualization of the cosmic web, showing a dense network of galaxies and dark matter filaments. The background is black, with galaxies represented as small, multi-colored dots (red, green, blue, yellow) forming a complex, interconnected structure. A circular arc with tick marks is visible on the left side of the image.

機械学習を用いた宇宙大規模構造の理論予言構築

小林洋祐 (東大Kavli IPMU D2)

第6回High Performance Computing Physics勉強会

@京都大学基礎物理学研究所, 2020. 1. 31

今回のトーク

このトークは京大基研の西道さん（2018年までIPMU所属）主導のプロジェクト”Dark Quest”の成果に基づいています。

[1] Nishimichi et al. 2019, “Dark Quest. I. Fast and Accurate Emulation of Halo Clustering Statistics and Its Application to Galaxy Clustering” (arXiv:1811.09504)

使われている計算・機械学習手法の主要References

ラテン超方格法

[2] Ba et al. 2015, “Optimal Sliced Latin Hypercube Design”, <https://doi.org/10.1080/00401706.2014.957867>

EMPCA

[3] Bailey et al. 2012, “Principal Component Analysis with Noisy and/or Missing Data” (arXiv:1208.41122)

ガウス過程回帰

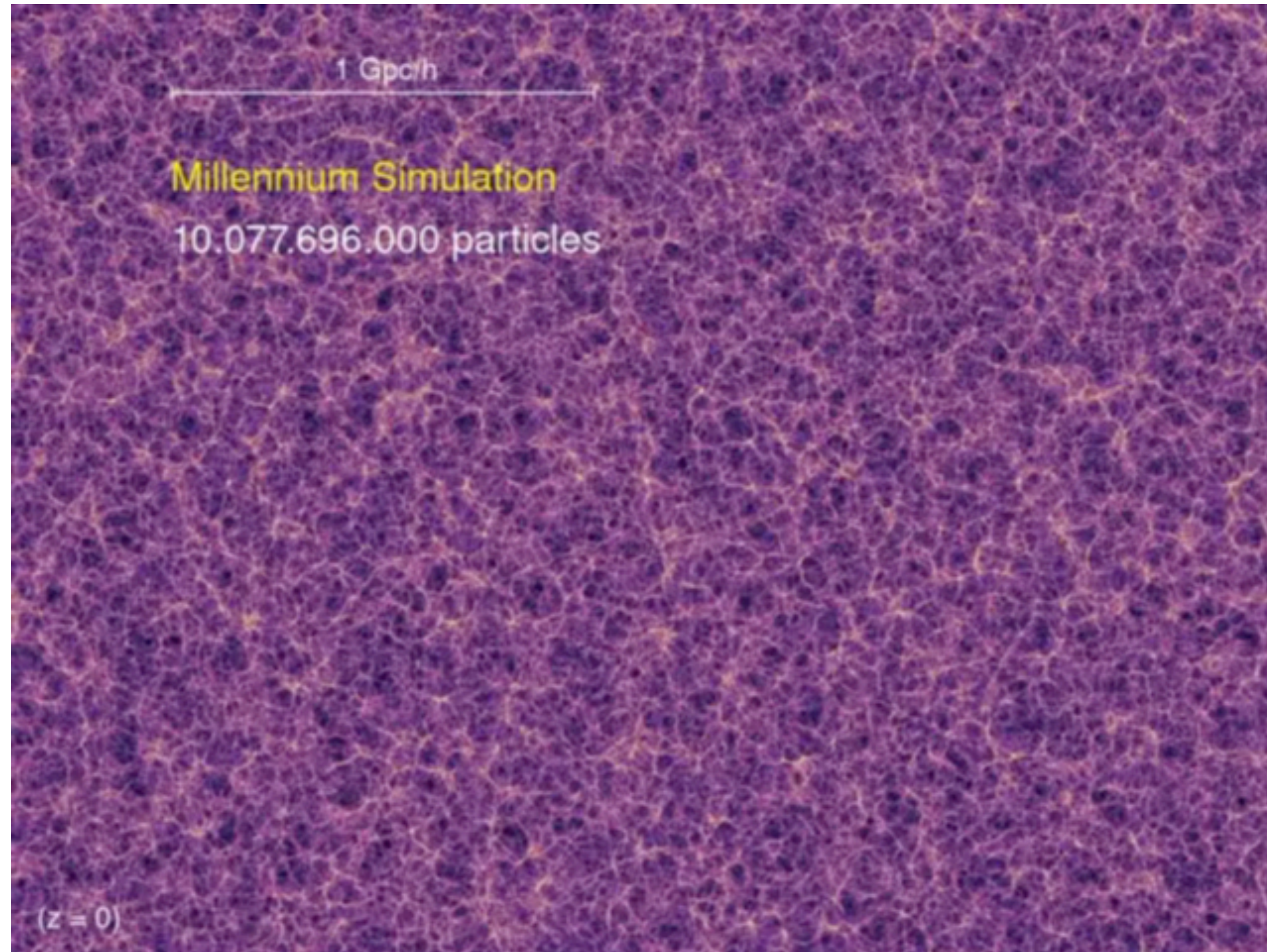
[4] Rasmussen & Williams, “Gaussian Process and Machine Learning”, the MIT press, 2006



1. 宇宙大規模構造を用いた宇宙論
2. エミュレータの概念とその目的
3. エミュレータ構築の手法
4. まとめ

宇宙の大規模構造

宇宙の ~ 1 Mpc (~ 100 万光年) 以上のスケールに広がる物質や銀河の分布



宇宙初期に生成された量子ゆらぎが成長してできたと考えられている

大規模構造の形成は宇宙の大スケールの枠組みを決める**宇宙論パラメータ**に依存しており、大規模構造の情報から宇宙論を制限できる

Credit: Springel et al. (Virgo Consortium)
Simulation code: Gadget-2

大規模構造の情報①: 重力レンズ効果

2次元天球面上で見える銀河の像は、その前景にある
重力源（主にダークマター）により歪められている

galaxy

galaxy cluster

lensed galaxy images

distorted light-rays

Earth

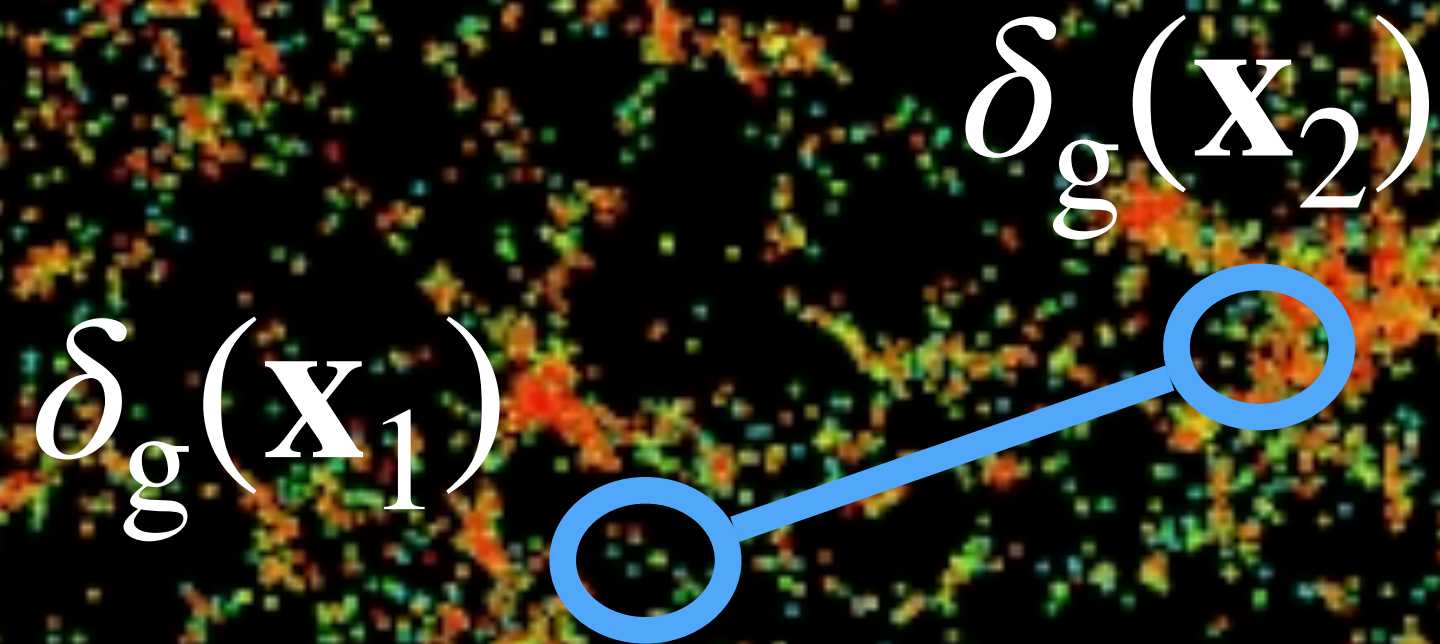
この歪み具合から、前景のダークマターハロー周
りの物質の集まり具合が統計的にわかる

$$\delta_g(\mathbf{x}) \equiv \frac{n_g(\mathbf{x})}{\bar{n}_g} - 1 \quad \delta_m(\mathbf{x}) \equiv \frac{\rho_m(\mathbf{x})}{\bar{\rho}_m} - 1$$
$$\left\langle \delta_g(\mathbf{x}_1) \delta_m(\mathbf{x}_2) \right\rangle \equiv \xi_{gm}(\mathbf{x}_1 - \mathbf{x}_2) \quad \text{にセンシティブ}$$

大規模構造の情報②: 銀河クラスタリング

クラスタリング = 「集まり具合」
大規模構造上に広がる銀河の数密度場の相関

$$\delta_g(\mathbf{x}) \equiv \frac{n_g(\mathbf{x})}{\bar{n}_g} - 1$$


$$\delta_g(\mathbf{x}_1)$$
$$\delta_g(\mathbf{x}_2)$$

$$\langle \delta_g(\mathbf{x}_1) \delta_g(\mathbf{x}_2) \rangle \equiv \xi_g(\mathbf{x}_1 - \mathbf{x}_2)$$

宇宙論ではとくに2点相関が重要で、これを
銀河の大域的分布から測定して解析する

Redshift z

Redshift z

ダークマターハローと銀河

ダークマター分布



galaxy-halo
connection



銀河分布



ダークマターハローと銀河

ダークマター分布

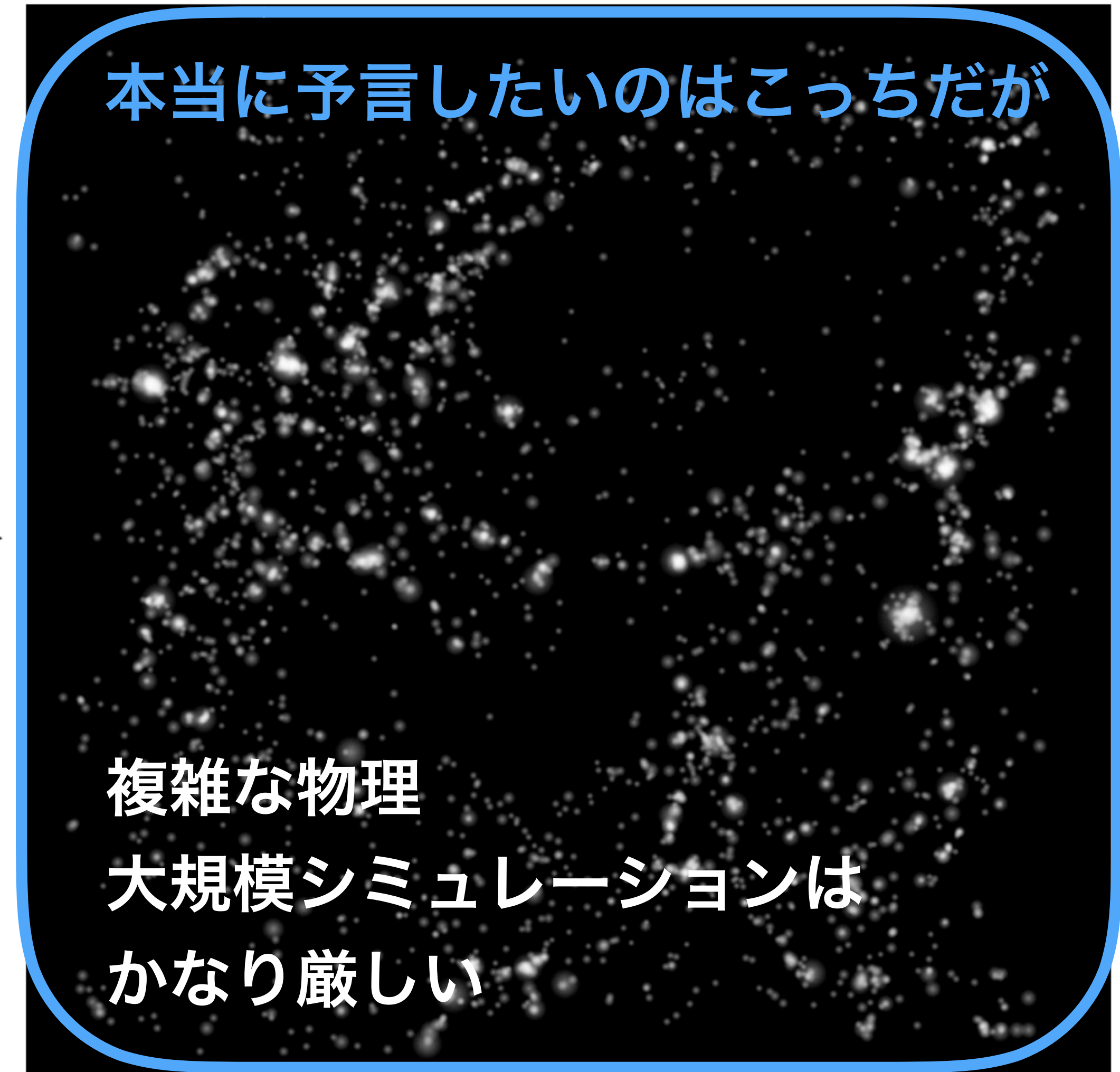


比較的単純な物理（重力）
大規模シミュレーションも
できる

galaxy-halo
connection



銀河分布

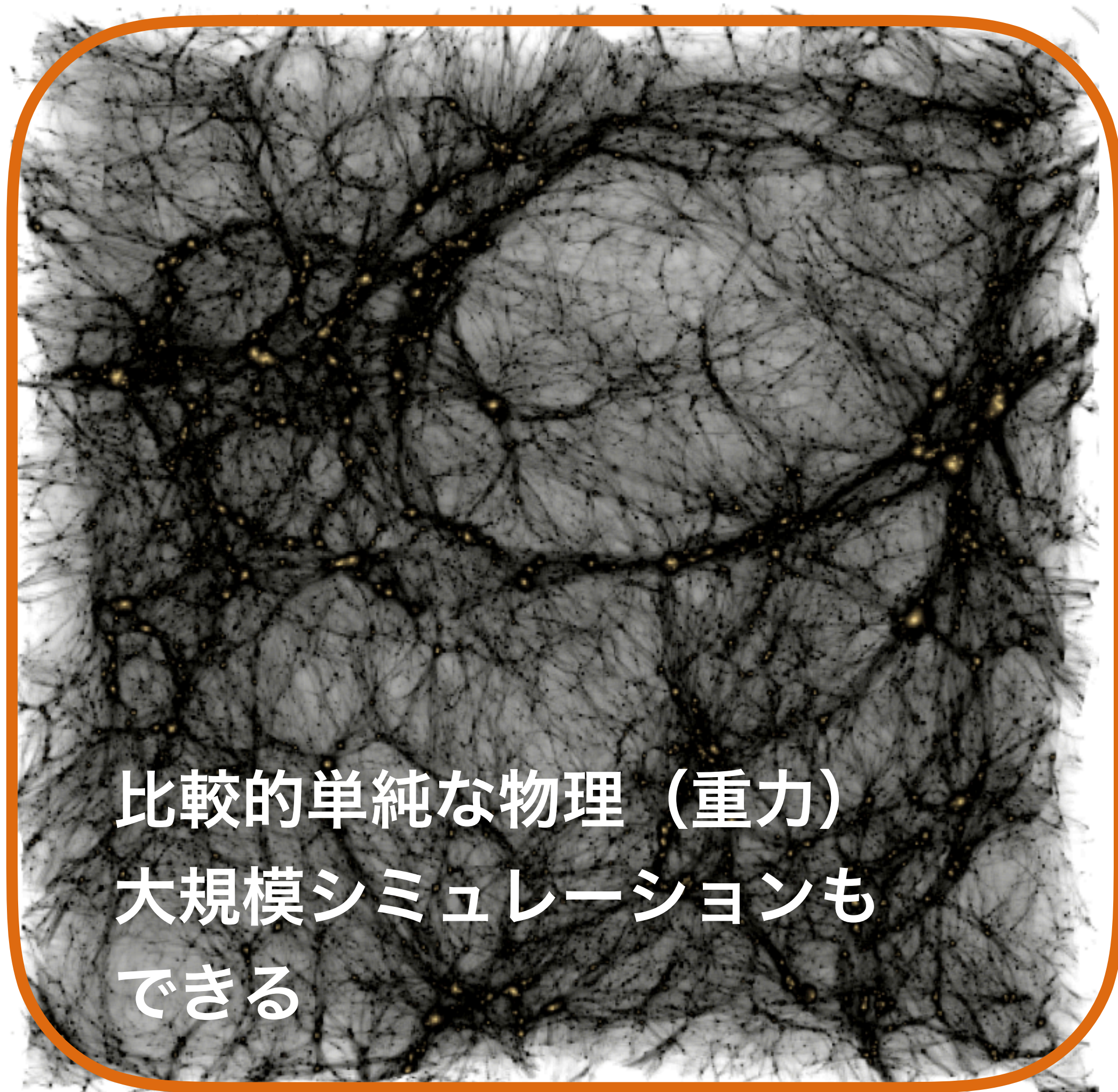


本当に予言したいのはこっちだが

複雑な物理
大規模シミュレーションは
かなり厳しい

ダークマターハローと銀河

ダークマター分布



比較的単純な物理（重力）
大規模シミュレーションも
できる

ここではこちらにフォーカス

銀河分布



複雑な物理
大規模シミュレーションは
かなり厳しい

Wechsler & Tinker 2018

宇宙大規模構造を用いた観測的宇宙論

宇宙大規模構造の観測（重力レンズ&銀河クラスタリング）からどうやって宇宙論につなげるか？

——→ 基本的には、**理論予言**と**観測データ**からベイズ推定をする

$$P(\text{model} \mid \text{data}) \propto P(\text{data} \mid \text{model})P(\text{model})$$

事後分布

尤度関数

事前分布

ここでは専らパラメータ推定を想定

$$P(\theta \mid \text{data}) \propto P(\text{data} \mid \theta)P(\theta)$$

これが目標

→ MCMCサンプリングでパラメータ事後分布を推定する問題



1. 宇宙大規模構造を用いた宇宙論
2. エミュレータの概念とその目的
3. エミュレータ構築の手法
4. まとめ

宇宙大規模構造の理論予言

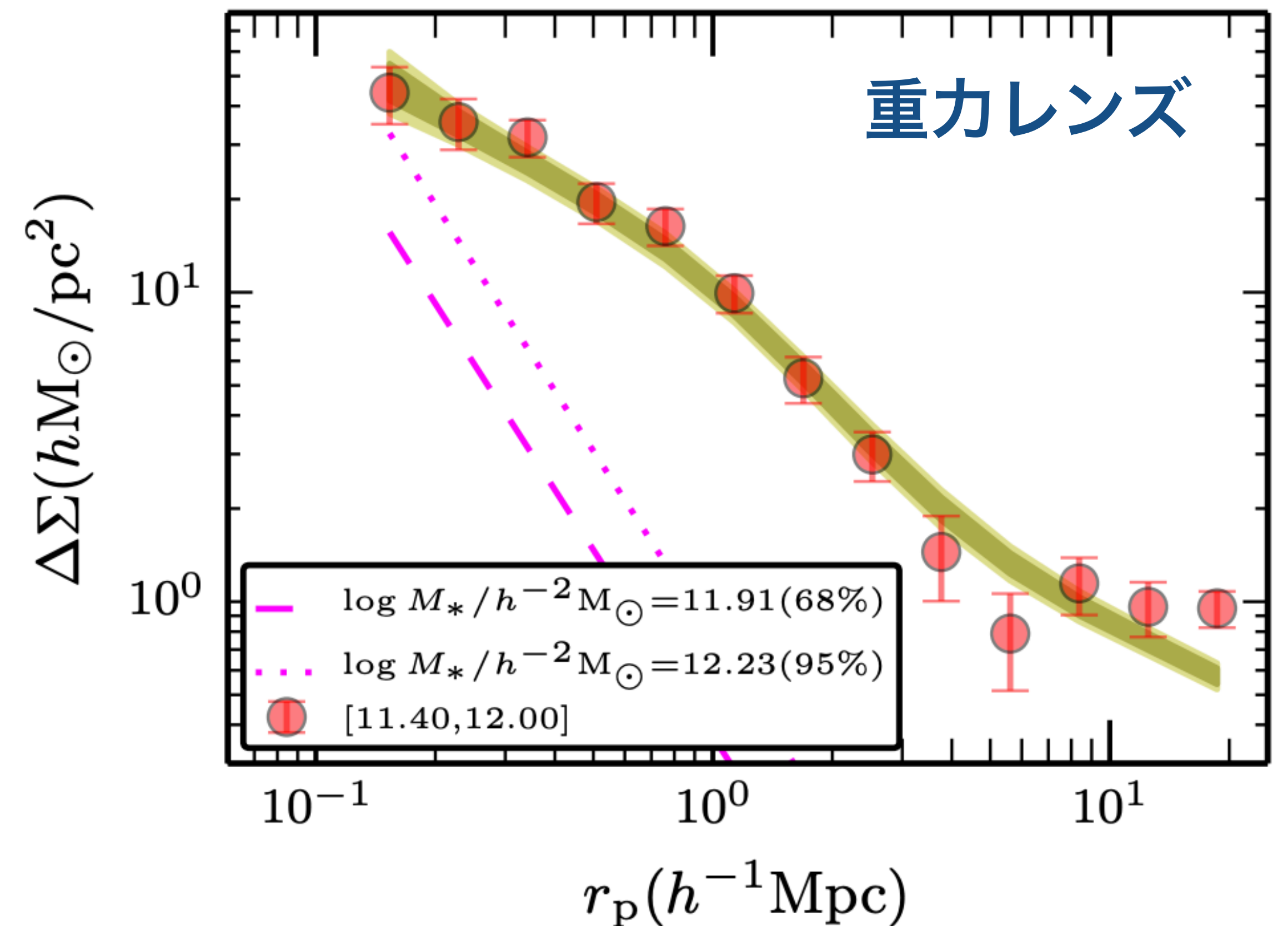
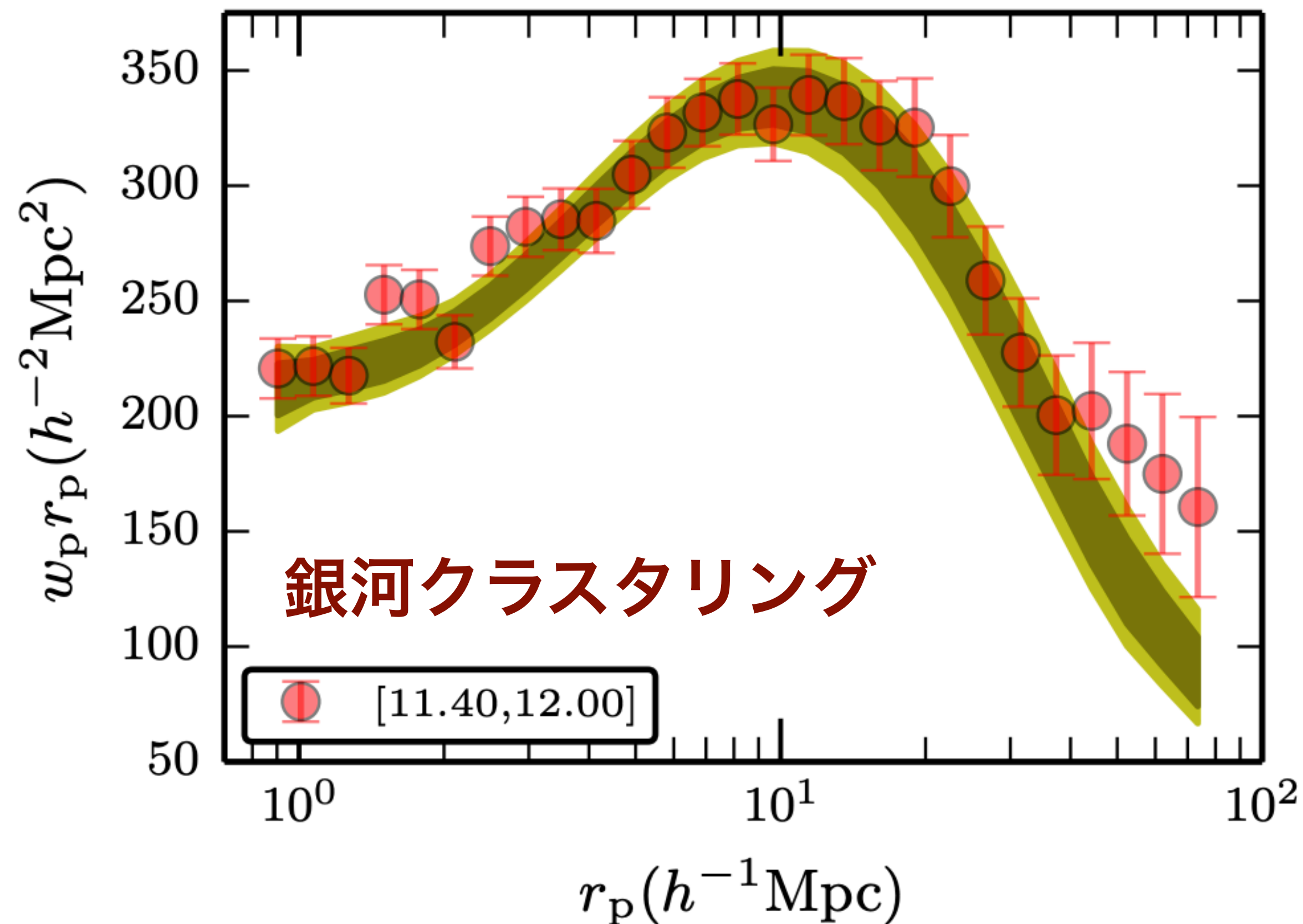
重力レンズ

$$\sim \xi_{\text{hm}}(r) = \langle \delta_{\text{h}}(\mathbf{x}) \delta_{\text{m}}(\mathbf{x} + \mathbf{r}) \rangle$$

銀河クラスタリング

$$\sim \xi_{\text{hh}}(r) = \langle \delta_{\text{h}}(\mathbf{x}) \delta_{\text{h}}(\mathbf{x} + \mathbf{r}) \rangle$$

More+ 2015



宇宙大規模構造の理論予言

解析的なアプローチは難しい -> ゆらぎの摂動展開が小さいスケールで破綻するから

2点相関などの宇宙論統計量は空間スケール r [Mpc] の関数なので、摂動論が有効な大スケールだけでなく、小スケールの情報も解析に取り入れたい

このとき、

1. ゆらぎの非線形効果が効く広範なスケールにわたって**精密に**

2. MCMCサンプリングによる宇宙論パラメータ推定ができるくらい**高速に**

重力レンズやクラスターリングの統計量を予言したい！

→ 宇宙論的N体シミュレーションなら、ダークマター分布を数値的に生成可能で、解析的な予言の難しい非線形効果まで正確に取り入れられる

N体シミュレーションに基づく“エミュレータ”構築

“エミュレータ”とは？

工学分野では「代理モデル (Surrogate model)」とも言う

→計算量のかかるシミュレーションをやらずに、好きなパラメータを入力してシミュレーションと同じように物理量の値を出力できるようにしたもの

地球統計学では「クリギング (Kriging)」という名前

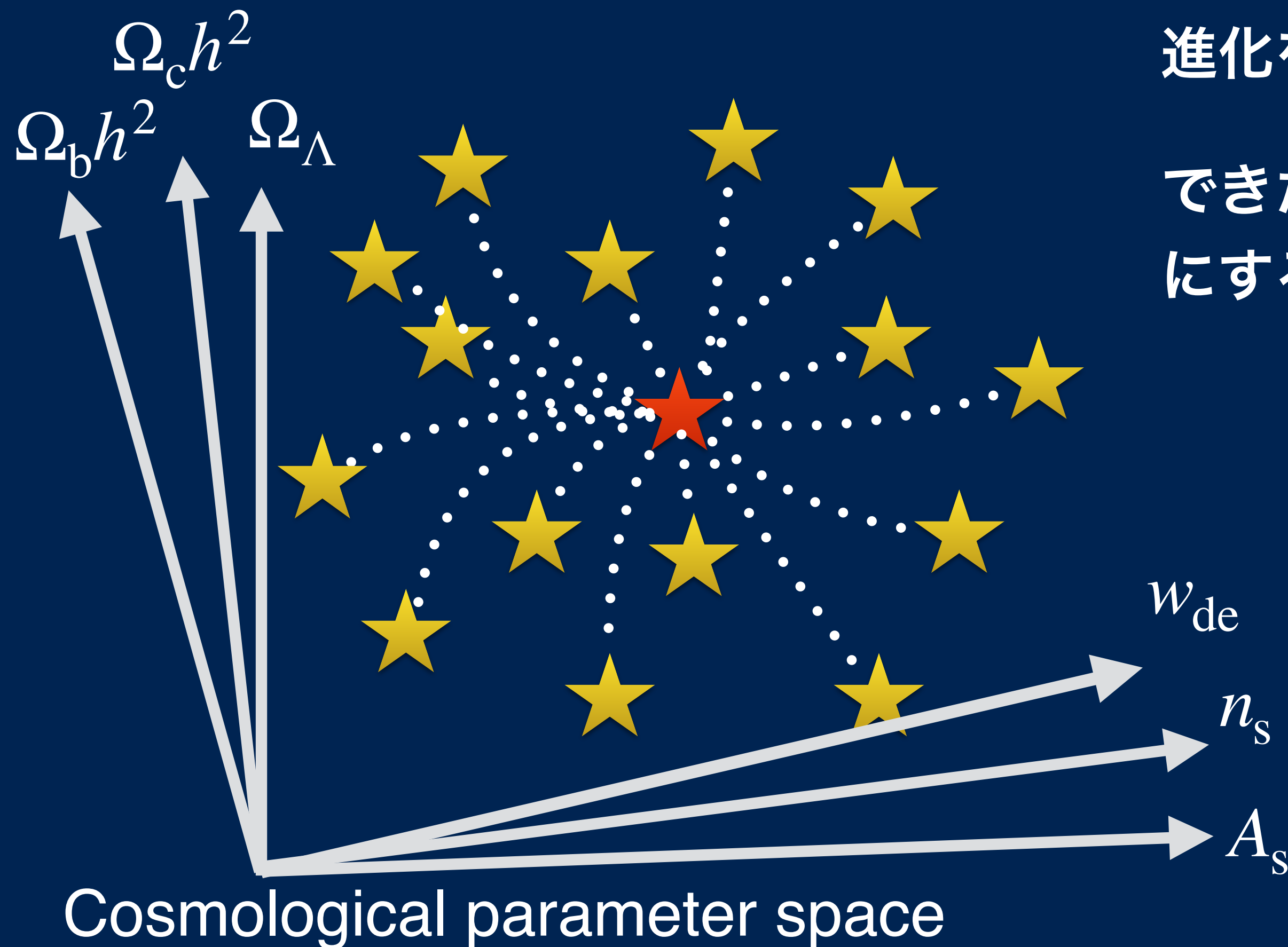
→特にエミュレータ構築の手法であるガウス過程回帰を指してクリギングと言うらしい

これを宇宙論の問題に適用した例がこのトークの内容

“いろいろな宇宙論パラメータで大規模構造形成シミュレーションを行い、宇宙論に依存するダークマターハローの統計量を測ってそのエミュレータを作る”

Dark Quest (Nishimichi+ 2019)

エミュレータ（呼称：“Dark Emulator”）構築のためのN体シミュレーション群



101個の宇宙論パラメータ点でダークマター分布の重力進化を解き、宇宙大規模構造を数値的に生成する

できたダークマター分布からハローを見つけ、カタログにする

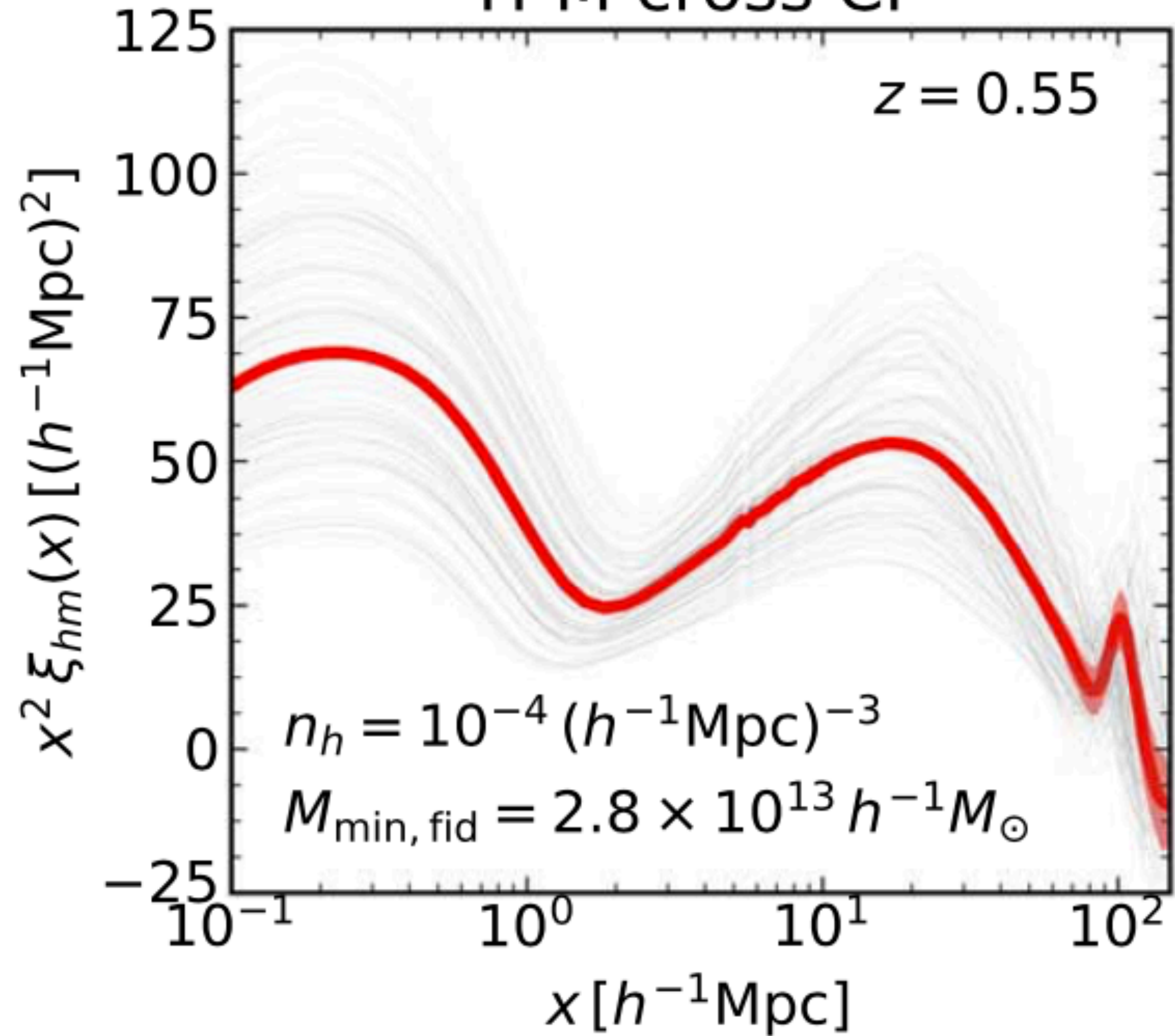
ハローカタログとダークマター分布のデータから興味ある統計量（2点相関関数など）を測定

得られたデータを宇宙論パラメータ空間上で補間、シミュレーションをやっていない宇宙論でも欲しい値を出力できるようにする

Dark Emulator

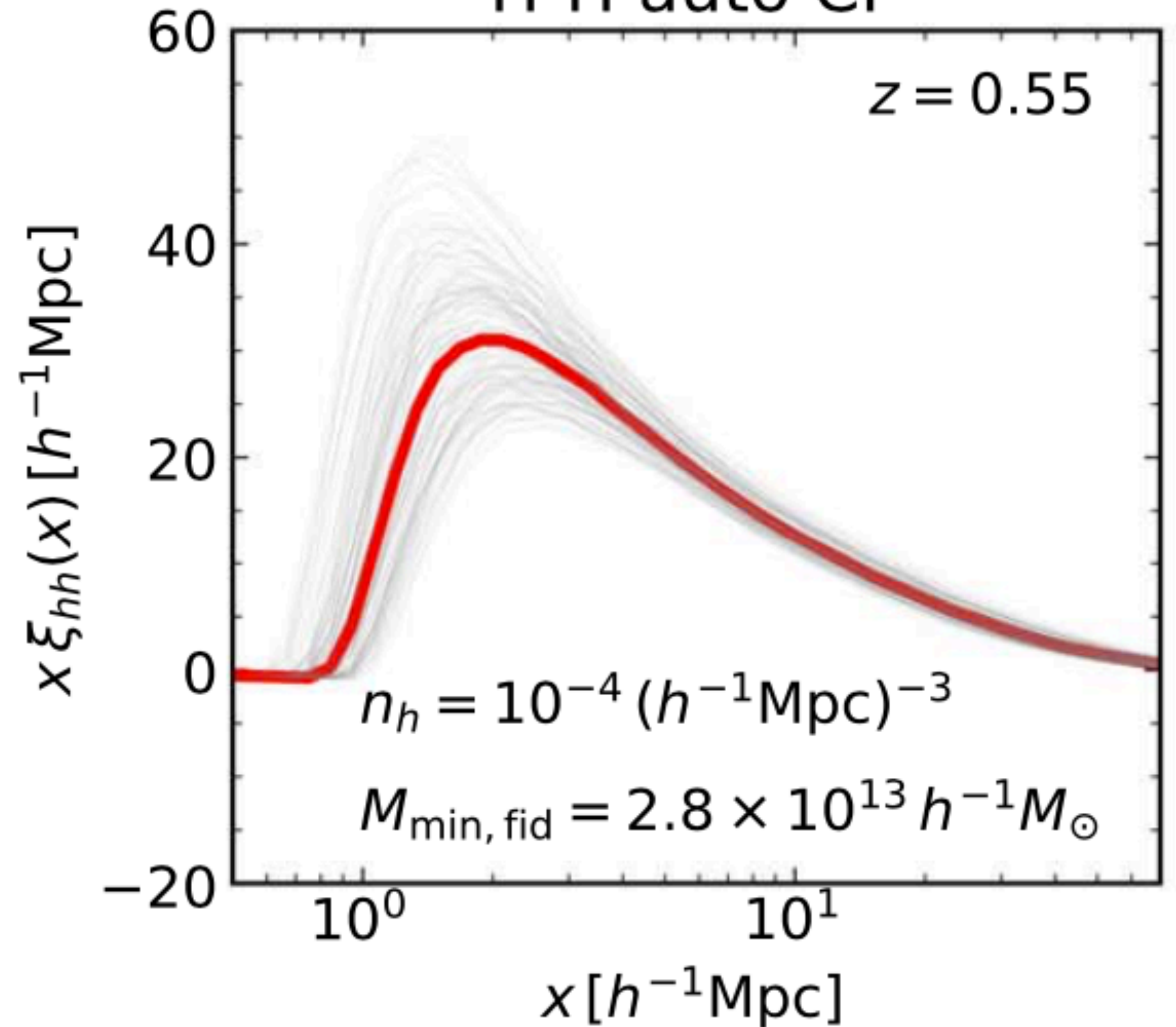
$$\xi_{hm}(r) = \langle \delta_h(\mathbf{x}) \delta_m(\mathbf{x} + \mathbf{r}) \rangle$$

H-M cross CF



$$\xi_{hh}(r) = \langle \delta_h(\mathbf{x}) \delta_h(\mathbf{x} + \mathbf{r}) \rangle$$

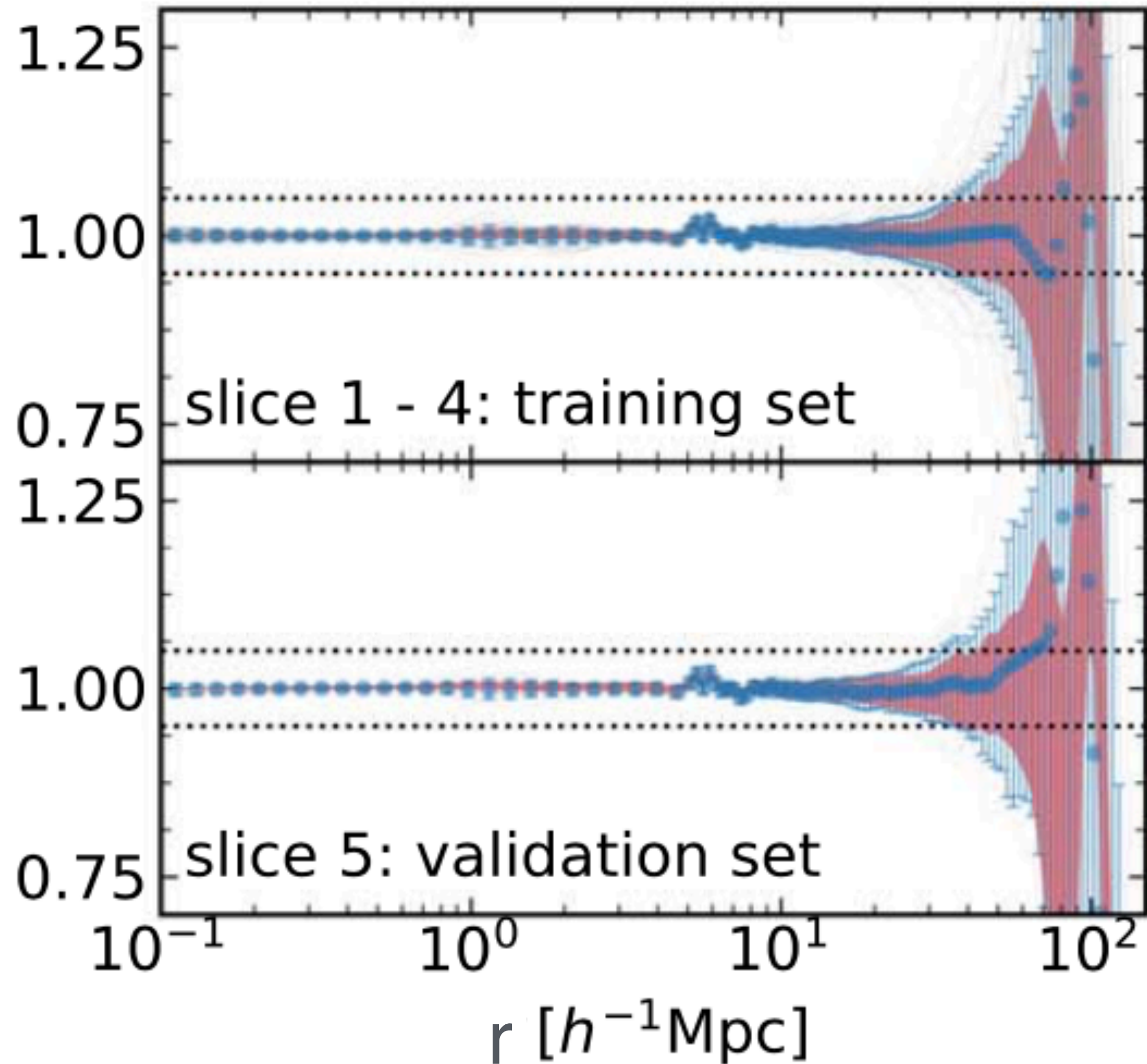
H-H auto CF



Dark Emulatorの学習精度

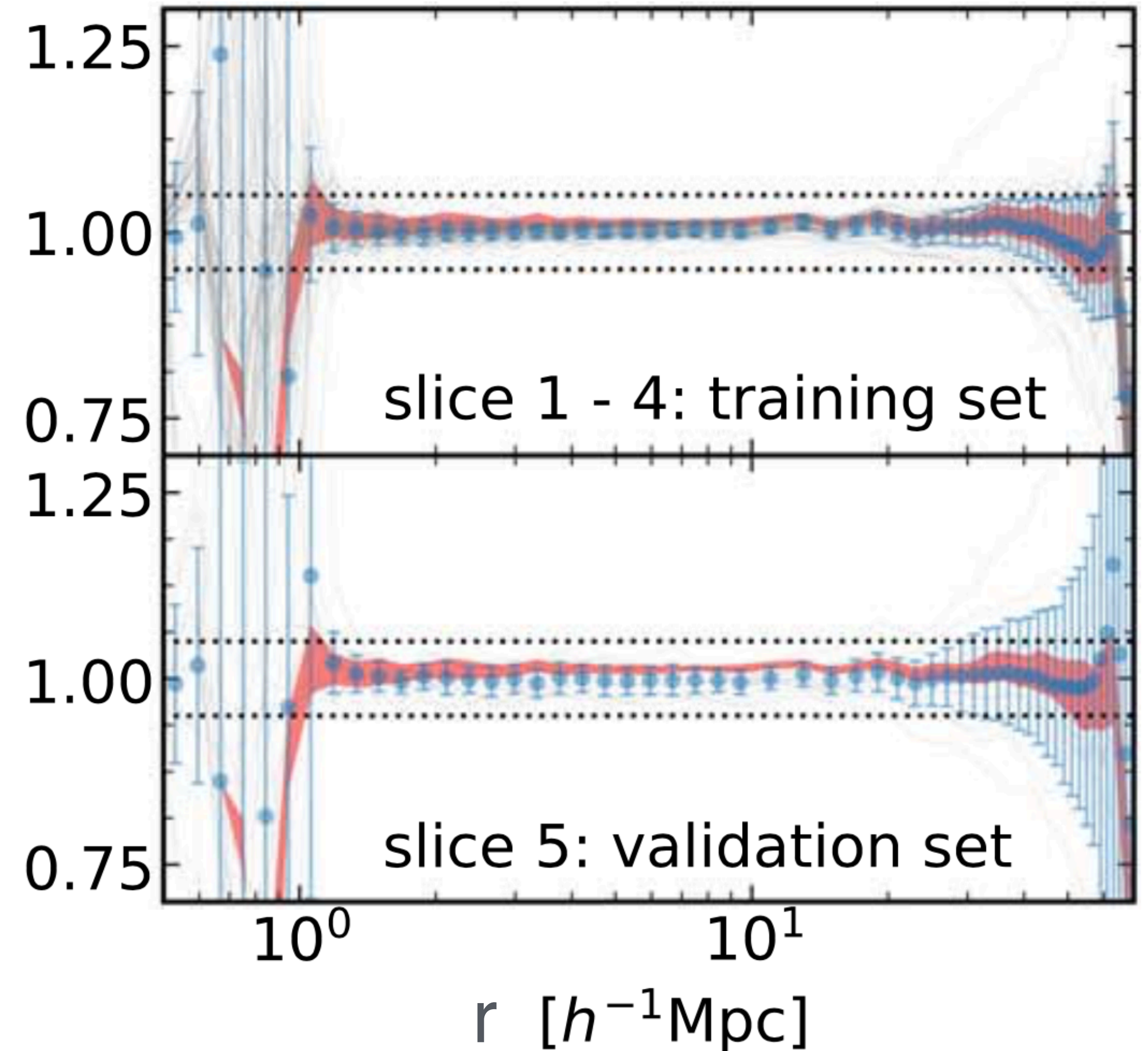
$$\xi_{\text{hm}}(r) = \langle \delta_{\text{h}}(\mathbf{x}) \delta_{\text{m}}(\mathbf{x} + \mathbf{r}) \rangle$$

sim / GP



$$\xi_{\text{hh}}(r) = \langle \delta_{\text{h}}(\mathbf{x}) \delta_{\text{h}}(\mathbf{x} + \mathbf{r}) \rangle$$

sim / GP





1. 宇宙大規模構造を用いた宇宙論
2. エミュレータの概念とその目的
3. エミュレータ構築の手法
4. まとめ

シミュレーション実験計画法

機械学習にかけるシミュレーションデータを生成するとき、宇宙論パラメータの値をどうとるか？

- 単純に「パラメータ1つにつきm個」とすると、6パラメータでは m^6 個の組み合わせ
- あまりに膨大な数のシミュレーションはできないが、考えているパラメータ領域は満遍なくサンプルしたい

→ **ラテン超方格法 (Latin hypercube design)**
「パラメータ1つにつきm個」かつ「6パラメータでもm個」

特色

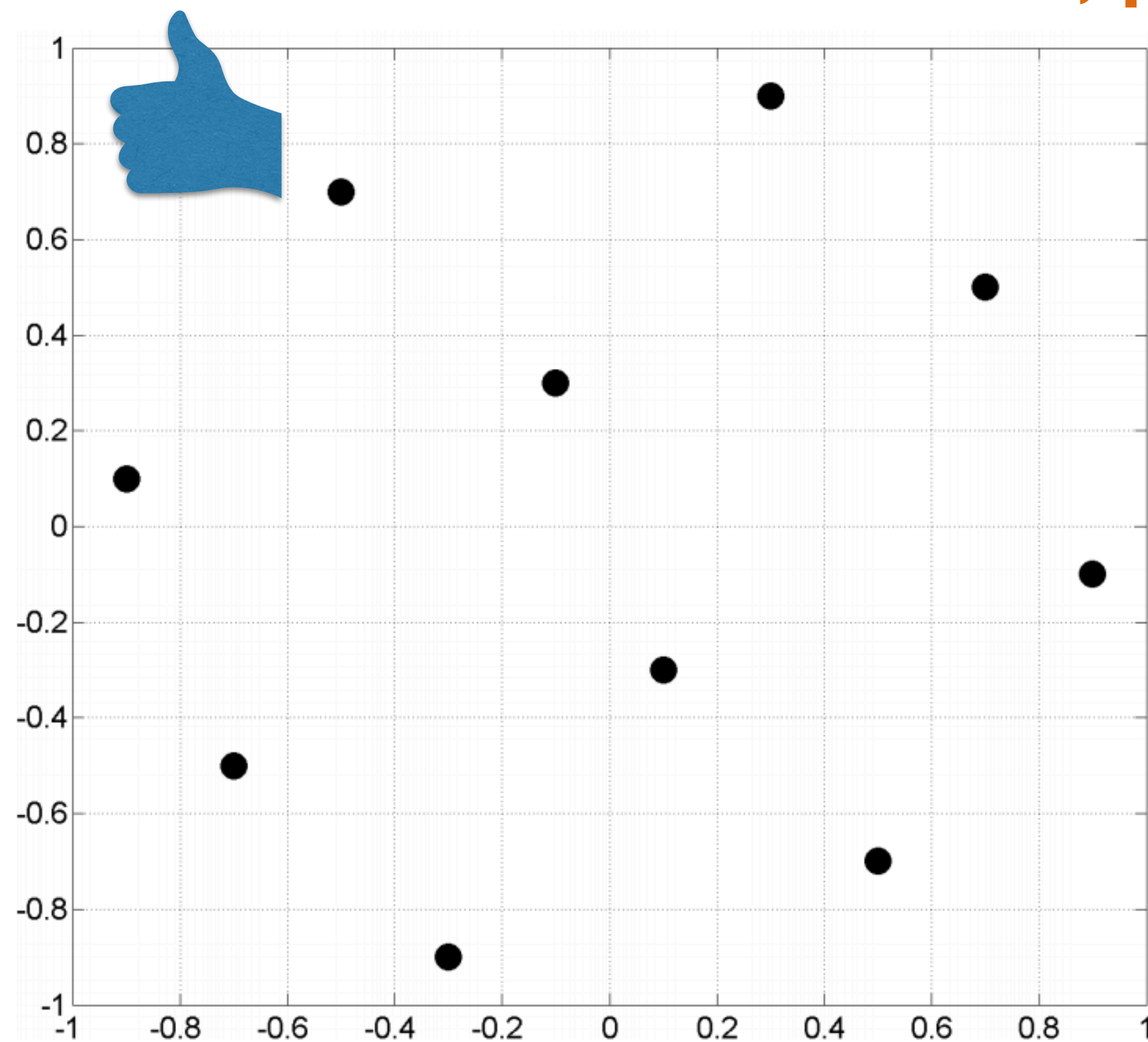
Space-filling property : パラメータ空間から一様にサンプルできる

Good projection property : 低次元空間へ射影したときにムダが少ない

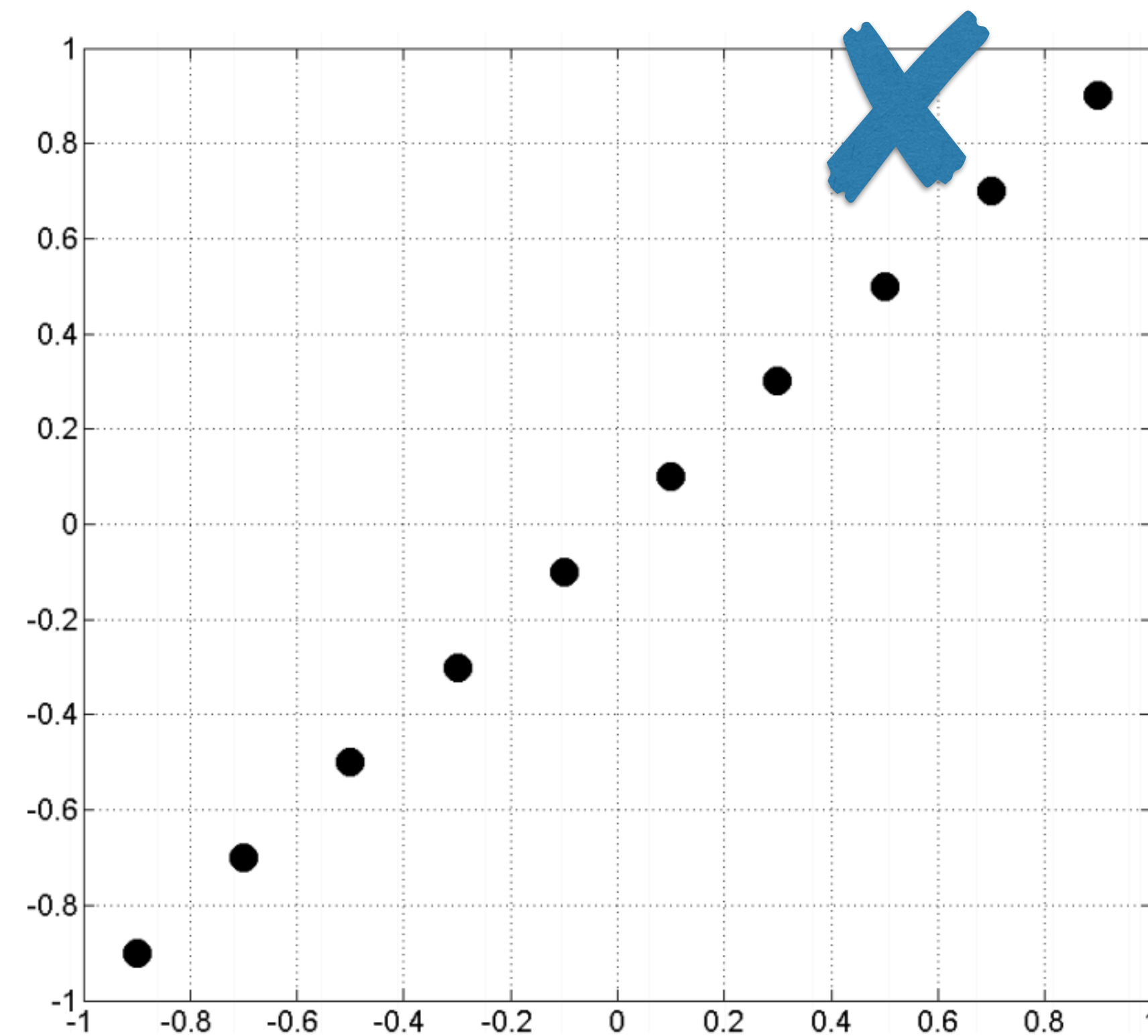
ラテン超方格法

p次元パラメータ空間でn回のシミュレーションをするとして、
まず、各々のパラメータで、n個全て異なる値にとる → good projection property

$n = 10, p = 2$ の場合



(a) Good Latin hypercube



(b) Bad Latin hypercube

学習・検証データを分けるラテン超方格法

機械学習では、学習データと検証データを分ける必要がある。このとき

- (1) 学習データも検証データも、**それぞれ**考えているパラメータ領域からの一様なサンプリングになっていて欲しい
- (2) 加えて、学習データと検証データを**合わせたデータ集合**がまたパラメータ領域からの一様なサンプリングになっていて欲しい

→ “**スライス**”という概念

1つのラテン超方格が複数の互いに素なラテン超方格を含むようなものを構成できる

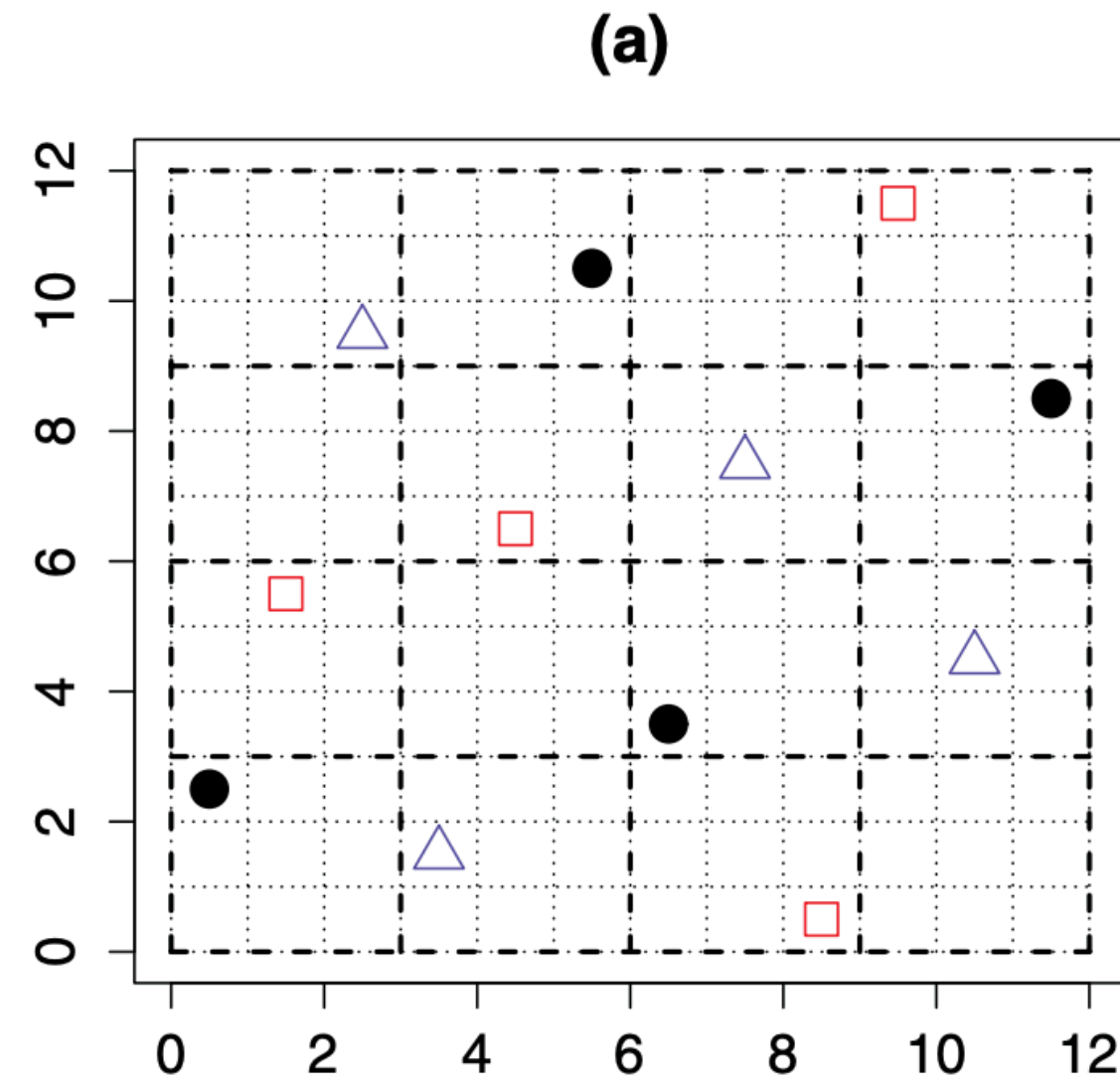
“Sliced” Latin Hypercube Design

ラテン超方格は $n \times p$ 行列で表せる。下図はパラメータが2つ ($p = 2$) の場合
3つの小さいラテン超方格（スライス）から1つの大きなラテン超方格を作る操作

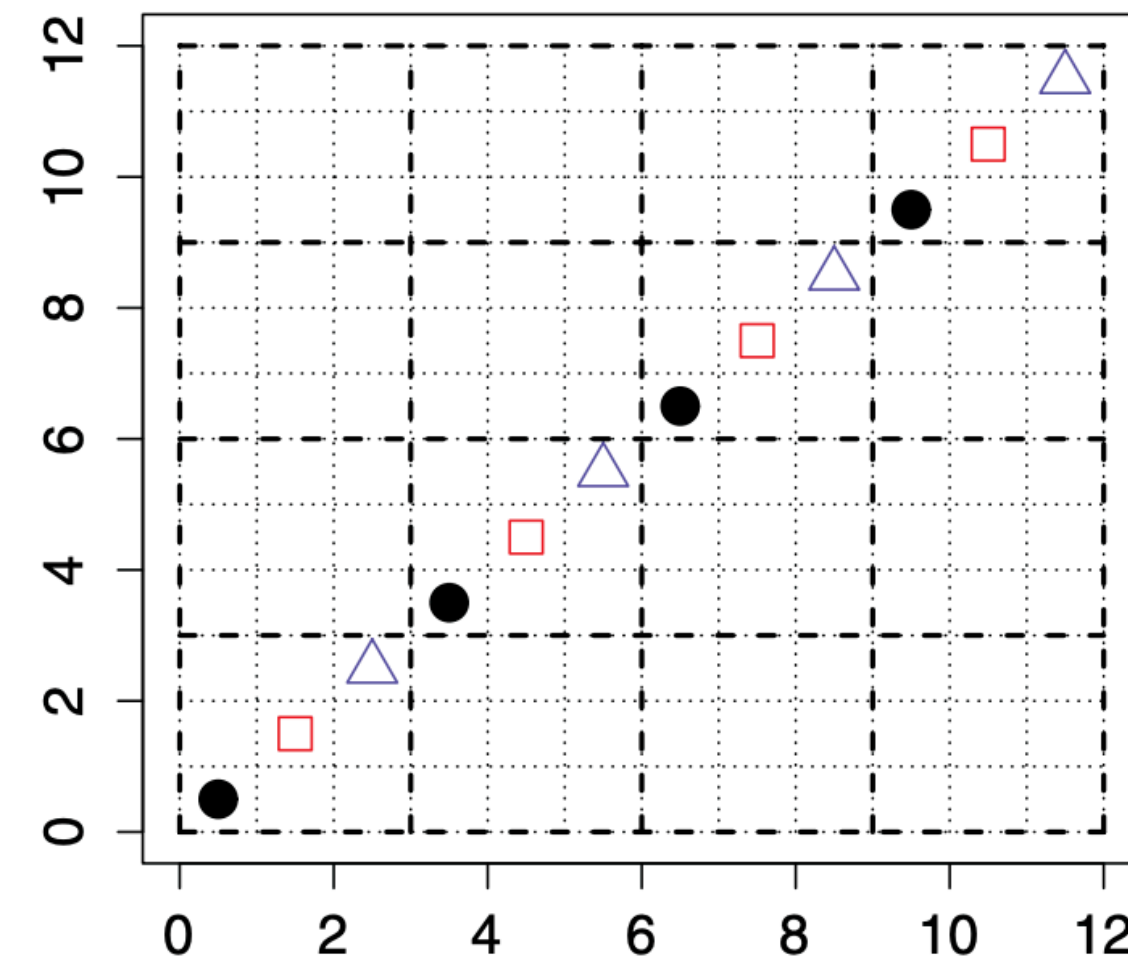


一番良いSliced Latin Hypercube Designはどれか？

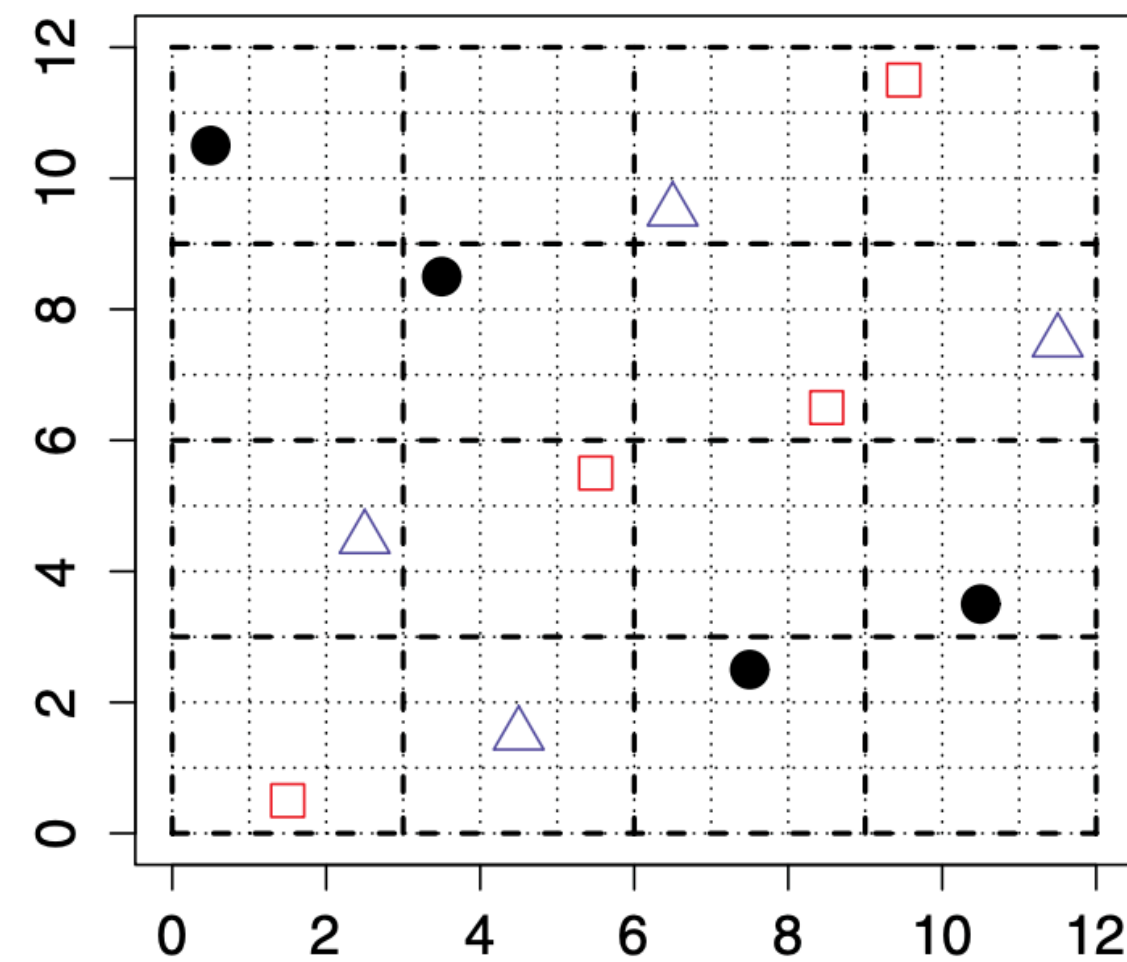
12シミュレーション、2パラメータ、3スライスの場合、下のどれもありうるが。。



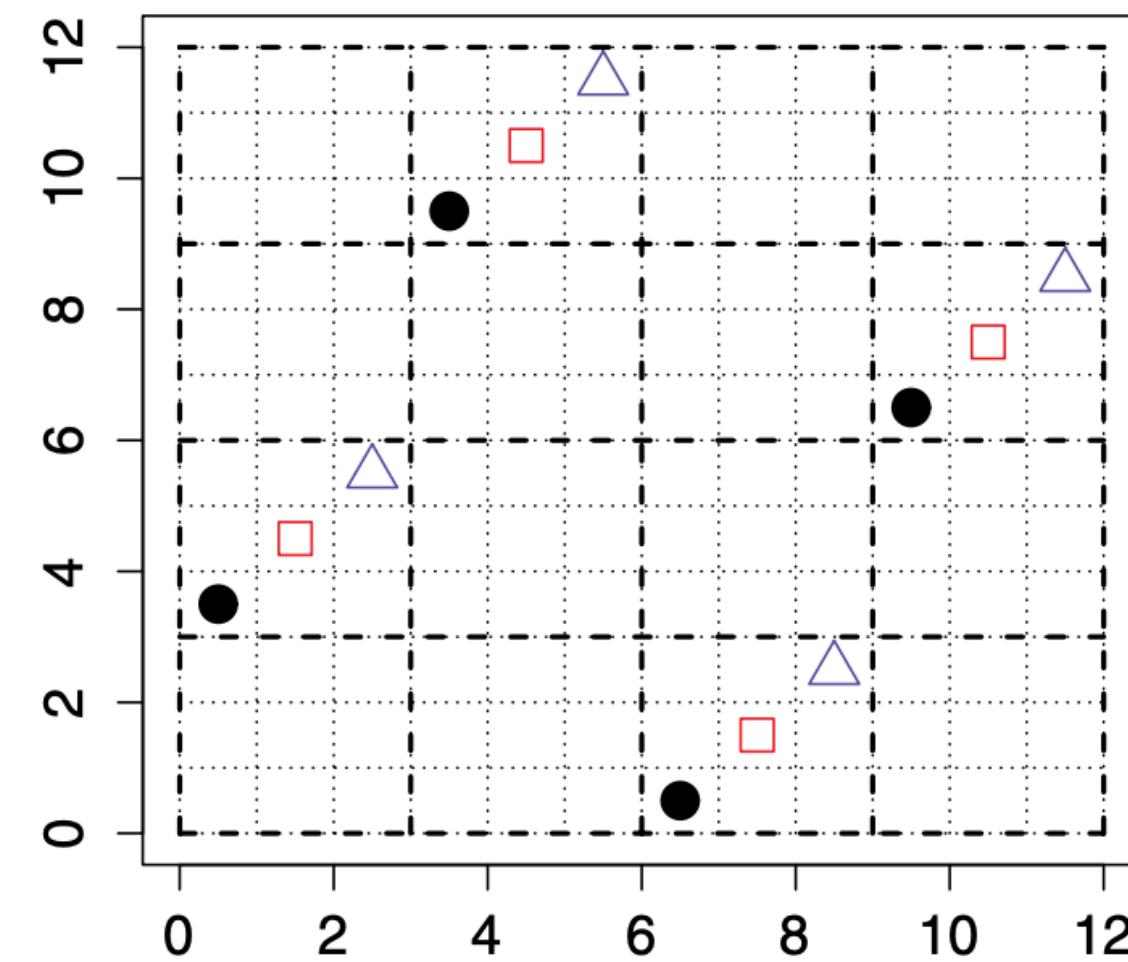
(b)



(c)



(d)



“Maximin” criteria

Sliced Latin hypercube構成のoptimalityをどんな尺度で決めるか？

→ ナイーブには、データ点の最小距離（のようなもの）が最大（=Maximin）になるように、つまり

$$\phi_r(X) = \left(\frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \frac{1}{d(x_i, x_j)^r} \right)^{\frac{1}{r}}$$

n : データ数 d : データ i, j 間の距離

を最小化するようにパラメータ点を入れ替えていく

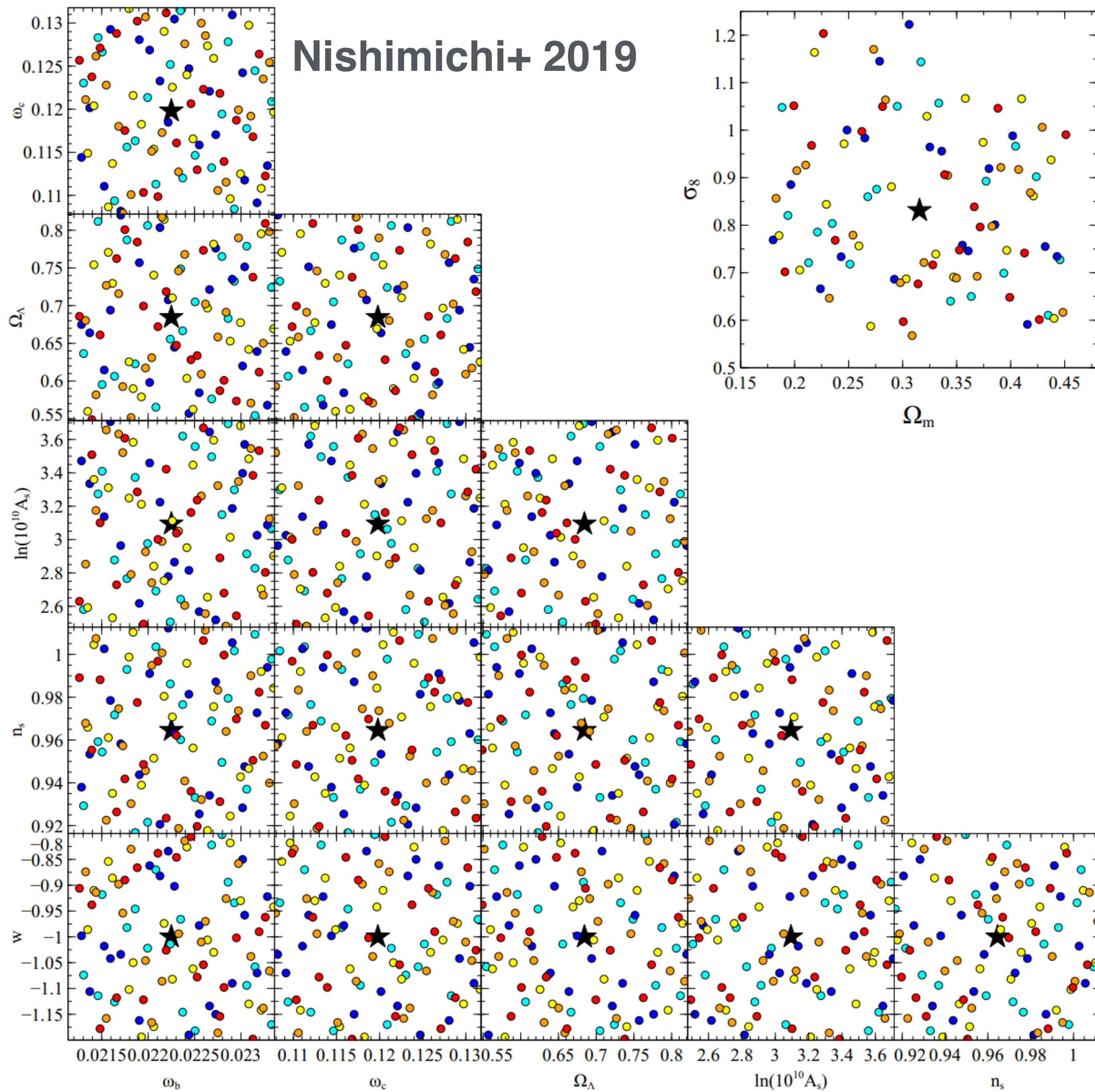
加えて、各スライスの中でも一様なサンプリングになっていてほしいので、

$$\phi_{Mm}(X) = \frac{1}{2} \left(\phi_r(X) + \frac{1}{t} \sum_{i=1}^t \phi_r(X_i) \right)$$

t : スライス数

を最小化すべき目的関数にする

Nishimichi+ 2019



Dark Questでは、宇宙論パラメータ20点ずつが1つのスライス
全部で5つのスライスがある

ガウス過程回帰を用いたエミュレータ構築

ダークマターハローの各種の統計量に対して、

Input: 宇宙論パラメータ

Output: その統計量のデータベクトル

という対応を学習させる（教師あり学習）

問題点

ガウス過程回帰では、普通は1つのガウス過程で1つのスカラー値関数を回帰する

シミュレーションから測ったデータベクトルはかなり多くの成分を持っているので、その成分一つ一つに対してガウス過程を構成していくのは現実的でない

→データベクトルの成分数（次元）を落として、その低次元データを回帰する

前処理：主成分分析(PCA)

高次元なデータベクトルを線形変換によって低次元ベクトルで要約する

このとき、低次元ベクトルから元のデータベクトルを最も忠実に復元するような変換を選ぶことができる

$\mathbf{X}$: (データの次元 $\times$ データの個数)

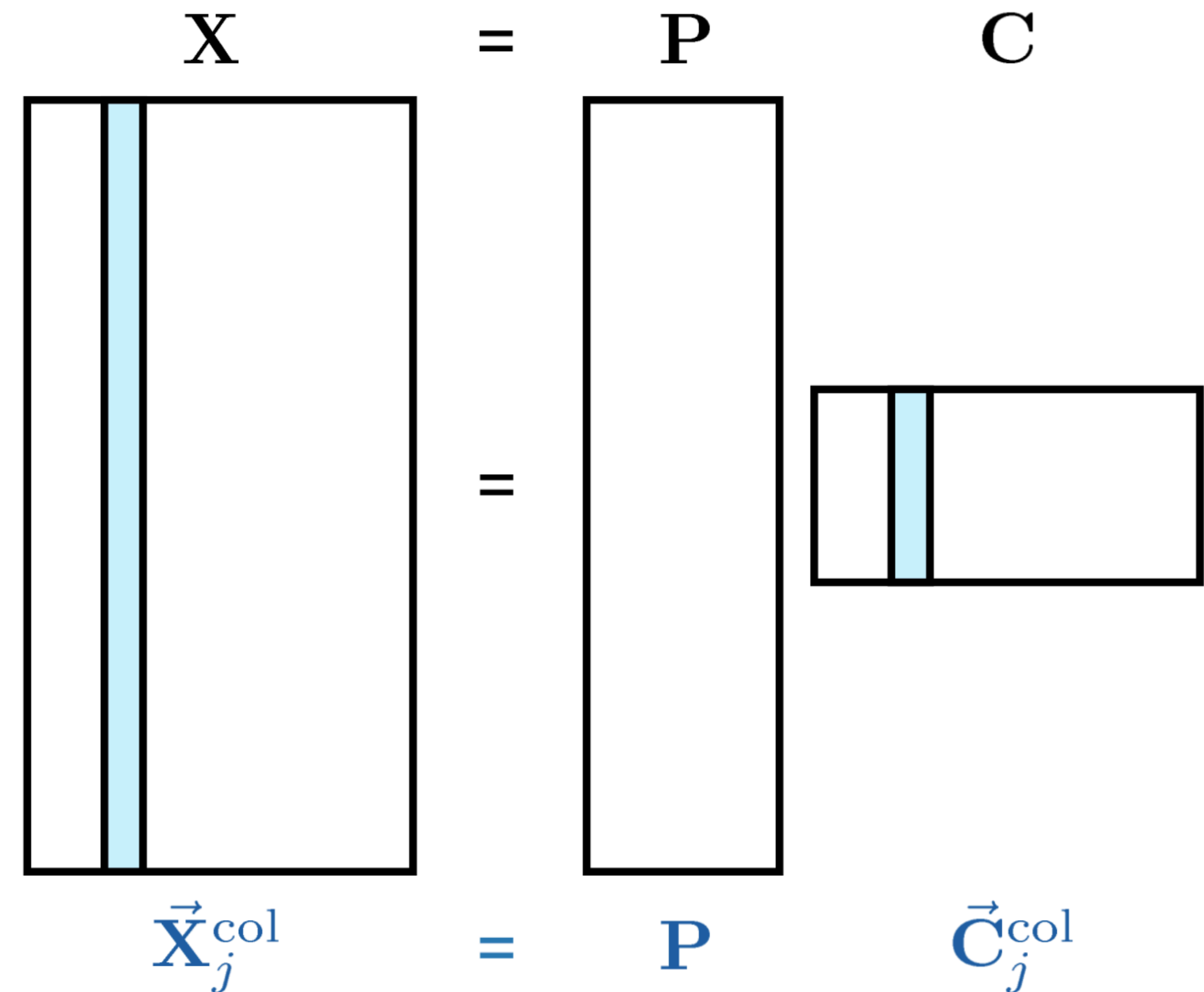
$\mathbf{P}$: (データの次元 $\times$ 主成分の個数)

$\mathbf{C}$: (主成分の個数 $\times$ データの個数)

→元のベクトルと復元されたベクトルとの間で

$$\chi^2 = \sum_{i,j} [\mathbf{X} - \mathbf{P}\mathbf{C}]_{ij}^2$$

を最小にする $\mathbf{P}$ と $\mathbf{C}$ をとる



EMPCA

データにノイズや欠損がある場合、それを考慮する重みをつけて主成分分析をしたい

PとCを交互に最適化していくことで、データのノイズ共分散を含んだ χ^2

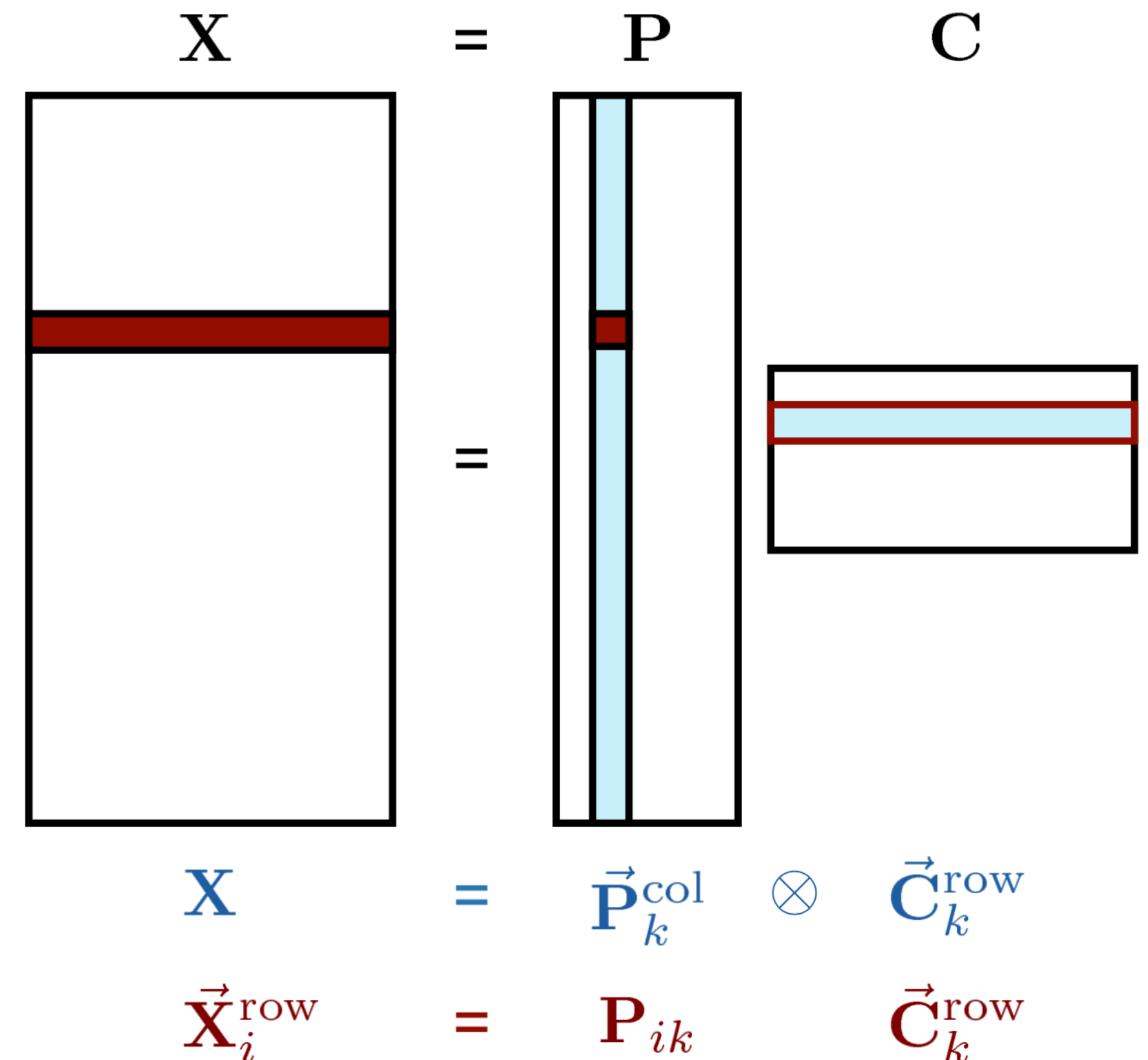
$$\chi^2 = \left(\vec{X} - [P] \vec{C} \right)^T V^{-1} \left(\vec{X} - [P] \vec{C} \right)$$

を最小化できる

参考：EM (Expectation Minimization) アルゴリズムの一例

Cの最適化: E-step

Pの最適化: M-step



ガウス過程回帰①

多次元パラメータ空間上の関数の非線形回帰

$$\mathbf{x}_n \mapsto f(\mathbf{x}_n)$$

入力ベクトル

出力値

ガウス過程とは・・・

- 確率変数の集合であって、その任意の有限個が多変数ガウス分布に従うというもの

入力空間の各点 $\mathbf{x}$ に対応する出力値 $f(\mathbf{x})$ を1つ1つ確率変数であると思って、
その統計的性質が

平均: $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$, $\rightarrow$ 簡単のため、通常は線形変換によってゼロにとる

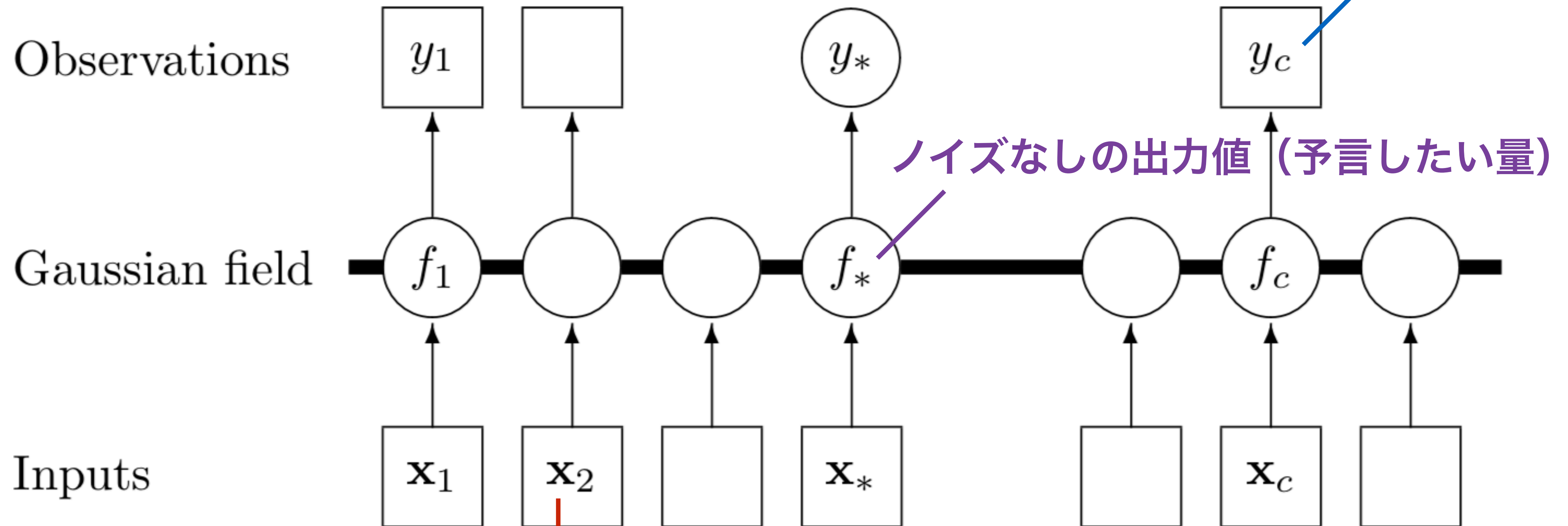
共分散: $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$,

であるようなガウス過程になっていると「見なす」

ガウス過程

グラフィカルモデルによるガウス過程 (Rasmussen & Williams 2006)

ノイズが乗った出力 (e.g. シミュレーションデータ)



入力ベクトル (e.g. 宇宙論パラメータ)

ガウス過程回帰② 事前分布

実際にデータでガウス過程回帰をするには：

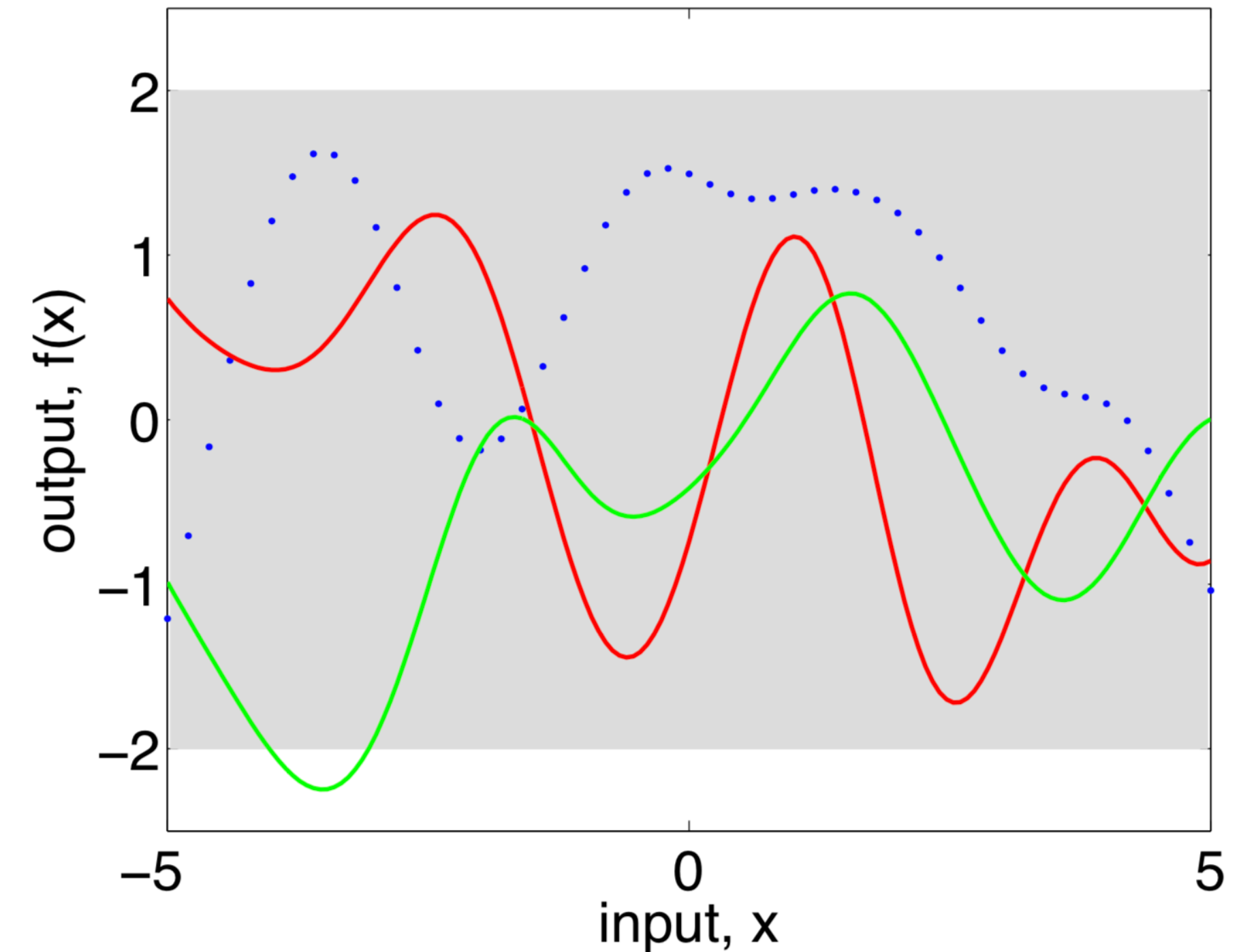
まず入力 $\mathbf{x}$ に対して出力値 $f(\mathbf{x})$ が対応するデータが n 点あるとして、この出力の集合 $\mathbf{f} := \{f(\mathbf{x}_i) \mid i = 1, \dots, n\}$ がある共分散関数 $k(\mathbf{x}, \mathbf{x}')$ をもつガウス過程であるとみなす。

このとき、 $n \times n$ 共分散行列が得られる

$$K = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

この時点ではまだきちんとした予言はできない

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*)) \rightarrow \text{事前分布}$$

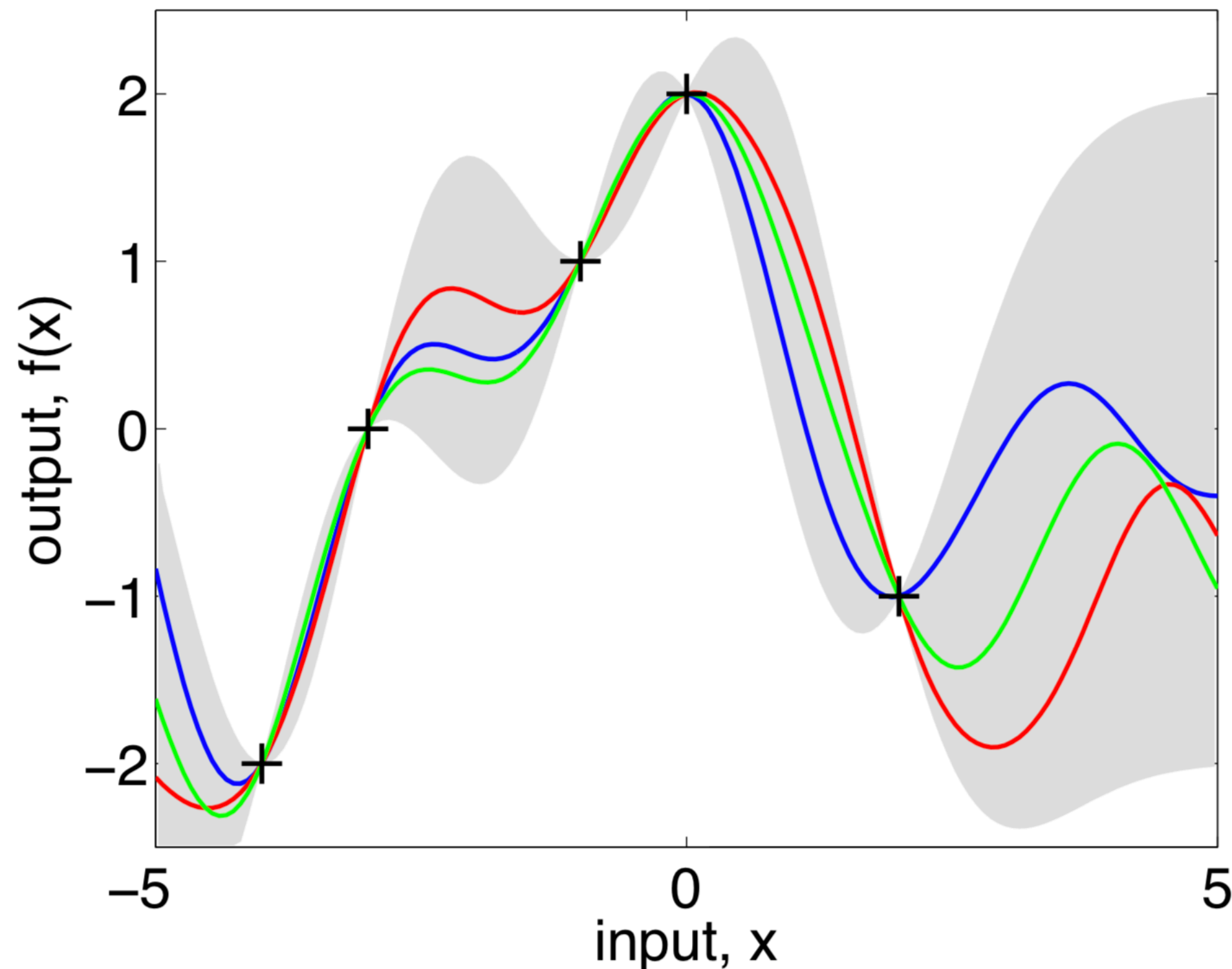


事前分布からサンプルされた1変数関数

ガウス過程回帰③ 事後分布

ガウス過程が回帰に有用なのは、学習データをもとに事後分布を作れるから

$$\text{事前分布} \begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right) \quad \begin{array}{l} \text{学習データ} \\ \text{未知のデータ} \end{array}$$



事後分布からサンプルされた1変数関数

事後分布: 学習データからの条件付き確率として得られる

$$\mathbf{f}_* | X_*, X, \mathbf{f} \sim \mathcal{N} \left(K(X_*, X) K(X, X)^{-1} \mathbf{f}, K(X_*, X_*) - K(X_*, X) K(X, X)^{-1} K(X, X_*) \right)$$

ガウス過程回帰④ ノイズありの場合

データのもつノイズを考慮すると、データの出力値は、実際には

$$y = f(\mathbf{x}) + \varepsilon \quad \varepsilon \sim \mathcal{N}(0, \sigma_n) \quad \text{だと思える}$$

このとき、 $n \times n$ 共分散行列は

$$\text{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \quad \text{or} \quad \text{cov}(\mathbf{y}) = K(X, X) + \sigma_n^2 I.$$

事前分布 $\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$ 学習データ
新しい予言値

事後分布

$$\mathbf{f}_* | X, \mathbf{y}, X_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)), \quad \text{where}$$

$$\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_* | X, \mathbf{y}, X_*] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y},$$

$$\text{cov}(\mathbf{f}_*) = \underline{K(X_*, X_*)} - \underline{K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*)}.$$

事前分布の共分散

この分だけ共分散が狭まる

ガウス過程回帰⑤ 学習

ガウス過程回帰のとき、共分散関数（あるいは、カーネルとも）

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

には具体的な関数形を仮定する

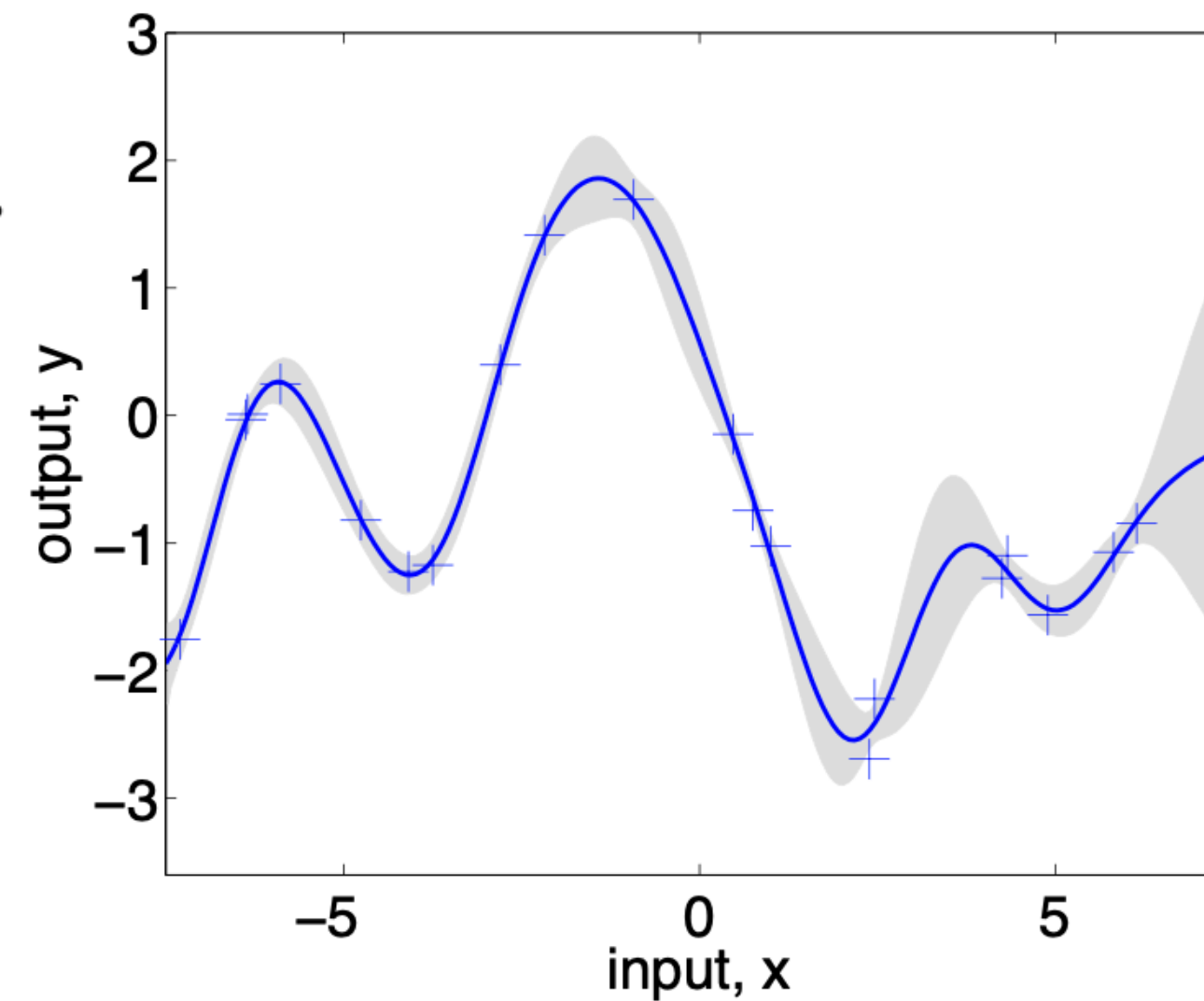
e.g. ガウス関数: $k(x, x') = \theta_0 \exp[-(x - x')^T \Theta (x - x')]$ ハイパーパラメータ

ここで導入したパラメータ θ_0 や Θ を通じて
共分散関数の形に自由度を持たせられる

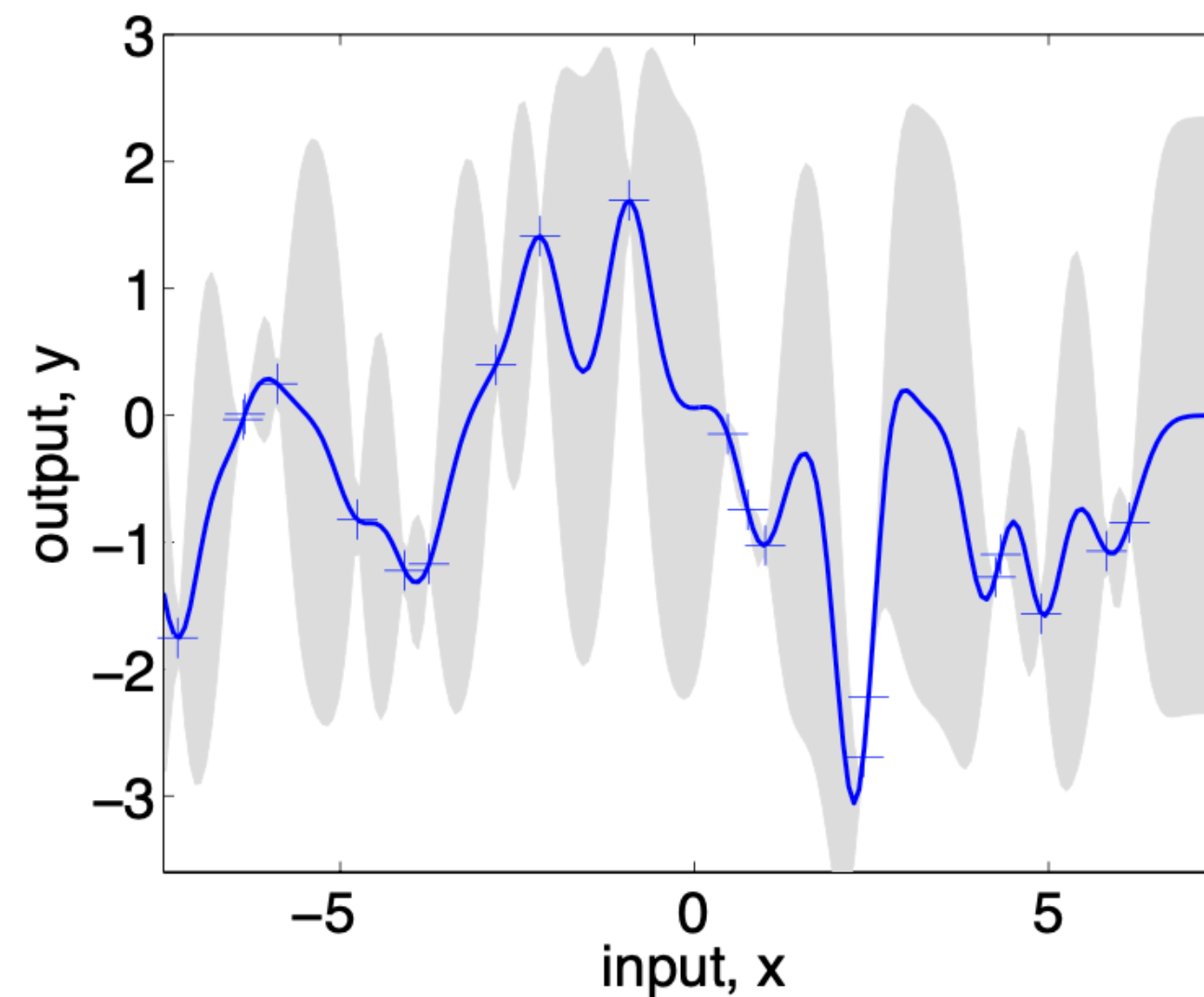
ガウス過程回帰⑤ 学習

$$k_y(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}(x_p - x_q)^2\right) + \sigma_n^2 \delta_{pq}.$$

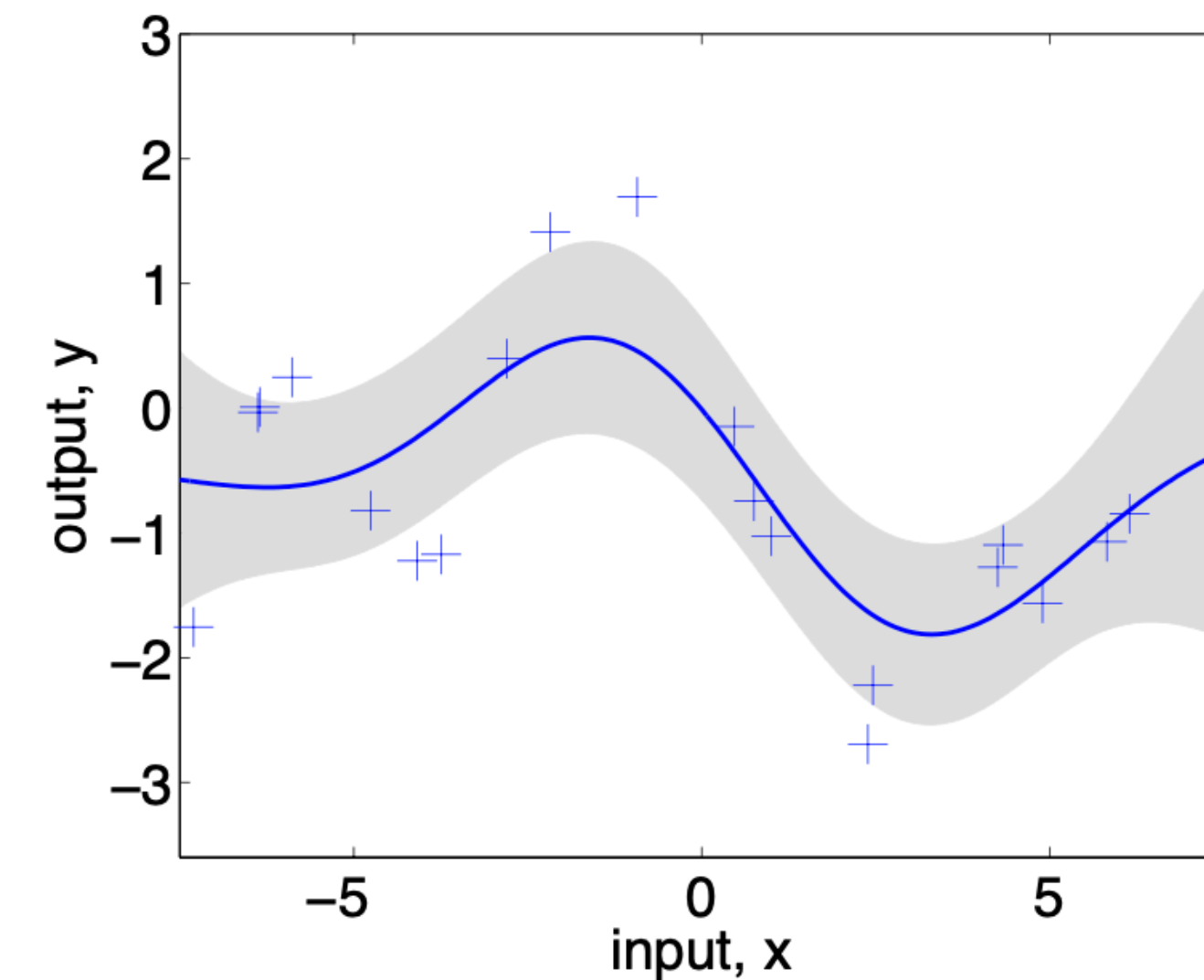
上のガウス型の共分散関数で、
ハイパーパラメータ ℓ を動かす



(a), $\ell = 1$



(b), $\ell = 0.3$



(c), $\ell = 3$

ガウス過程回帰⑤ 学習

ガウス過程回帰のとき、**共分散関数**（あるいは、**カーネル**とも）

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

には具体的な関数形を仮定する

e.g. ガウス関数: $k(x, x') = \theta_0 \exp[-(x - x')^T \Theta (x - x')]$ **ハイパーパラメータ**

ここで導入したパラメータ θ_0 や Θ を通じて
共分散関数の形に自由度を持たせられる

ハイパーパラメータを最適化すること=**ガウス過程による「学習」**

具体的には、marginal likelihoodを最大化するハイパーパラメータを見つける

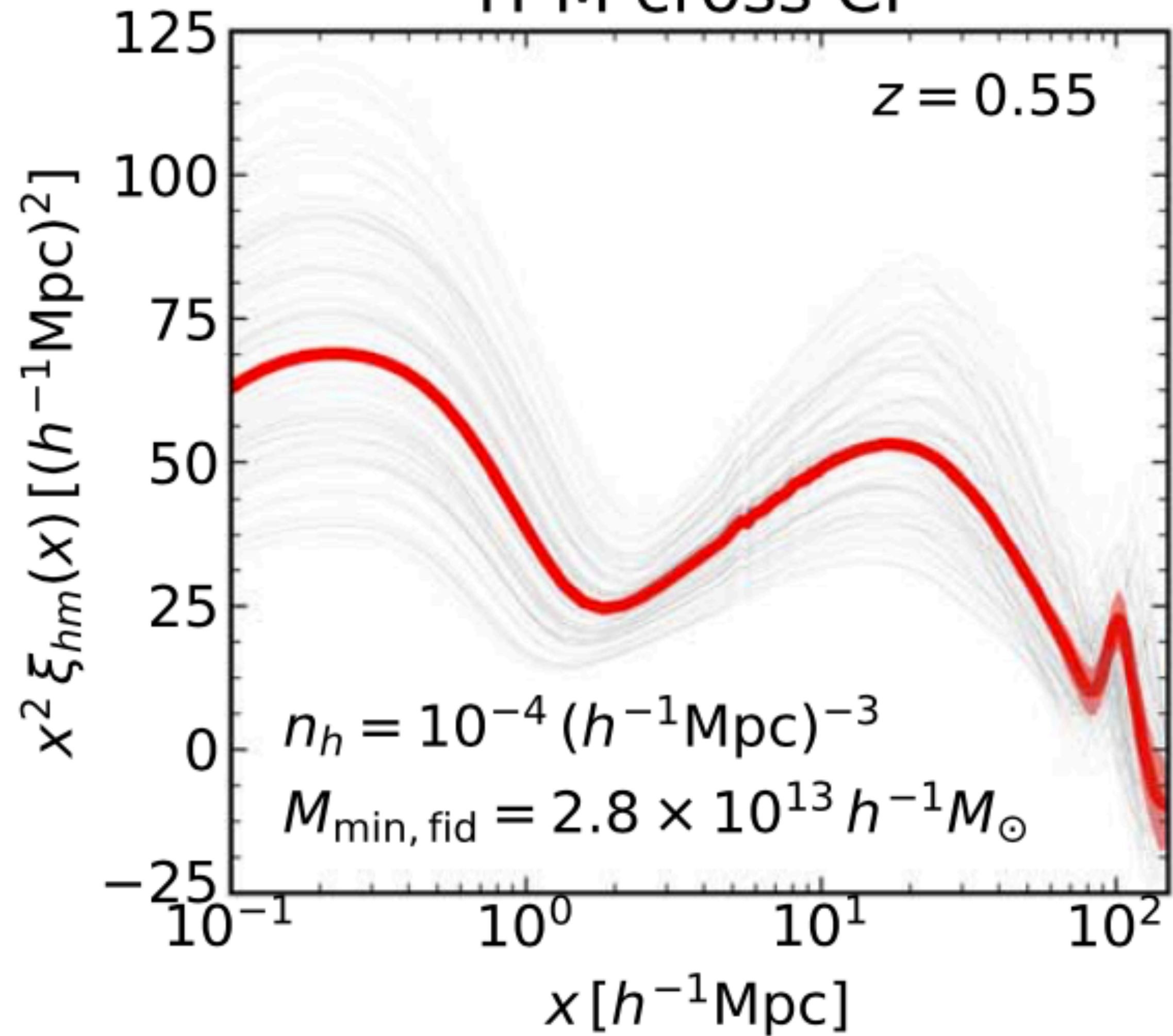
$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f}, X) p(\mathbf{f}|X) d\mathbf{f}.$$

$$\log p(\mathbf{y}|X) = -\frac{1}{2} \mathbf{y}^\top (K + \sigma_n^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |K + \sigma_n^2 I| - \frac{n}{2} \log 2\pi.$$

Dark Emulator

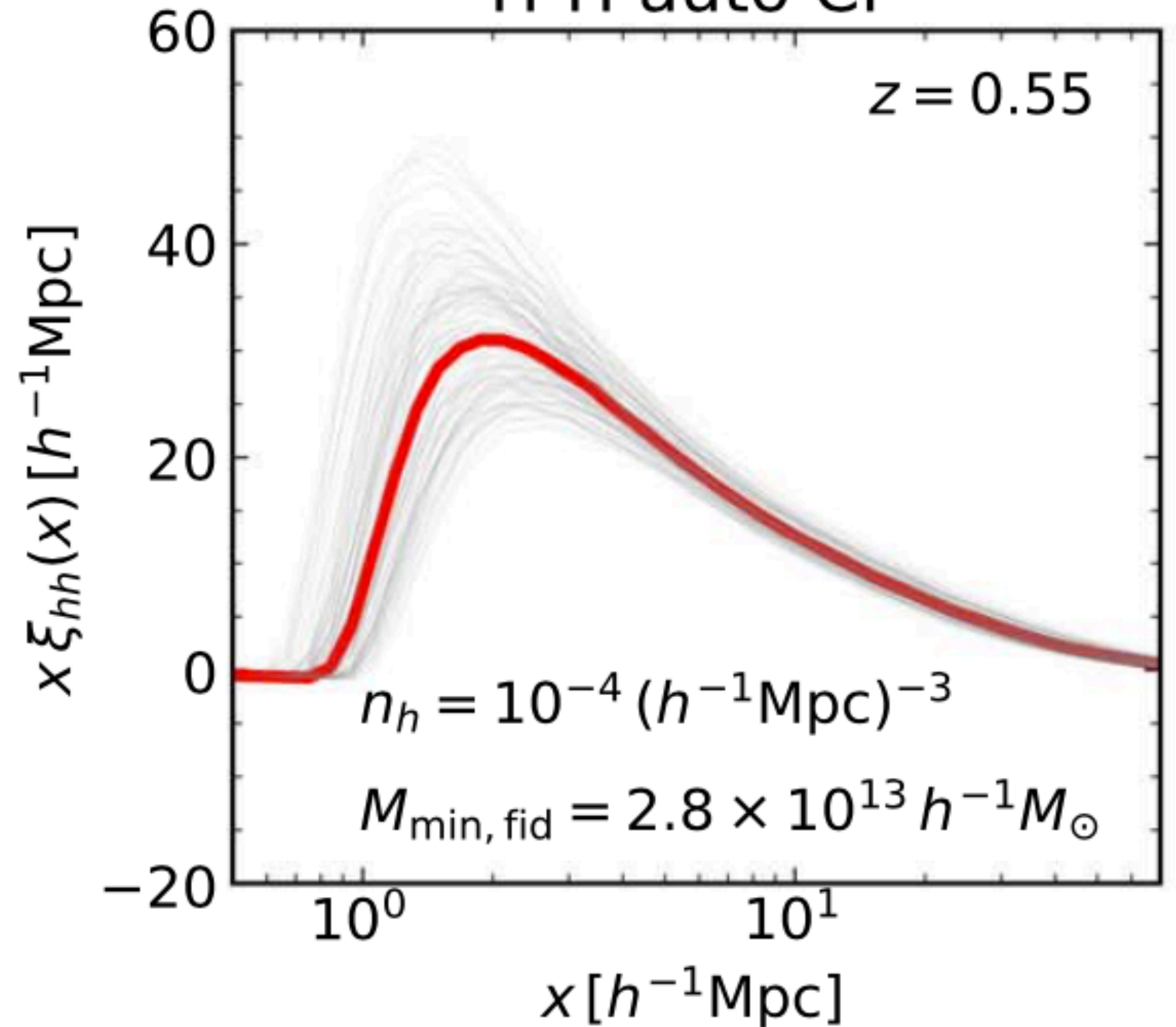
$$\xi_{hm}(r) = \langle \delta_h(\mathbf{x}) \delta_m(\mathbf{x} + \mathbf{r}) \rangle$$

H-M cross CF



$$\xi_{hh}(r) = \langle \delta_h(\mathbf{x}) \delta_h(\mathbf{x} + \mathbf{r}) \rangle$$

H-H auto CF



まとめ

宇宙大規模構造の統計量を精密かつ高速に予言するコード（Dark Emulator）を、N体シミュレーション群と機械学習手法を用いて構築

ハロー数密度場と物質密度場との相関、ハロー数密度場の2点相関などを、（解析的モデルを使ったモデリングが難しい）小さいスケールまで精密に予言可能

具体的なテクニック：

1. ラテン超方格を使って効率的なシミュレーション計画
2. 主成分分析を使ってシミュレーションデータの次元削減
3. ガウス過程回帰を使って宇宙論パラメータ空間上で学習データを補間