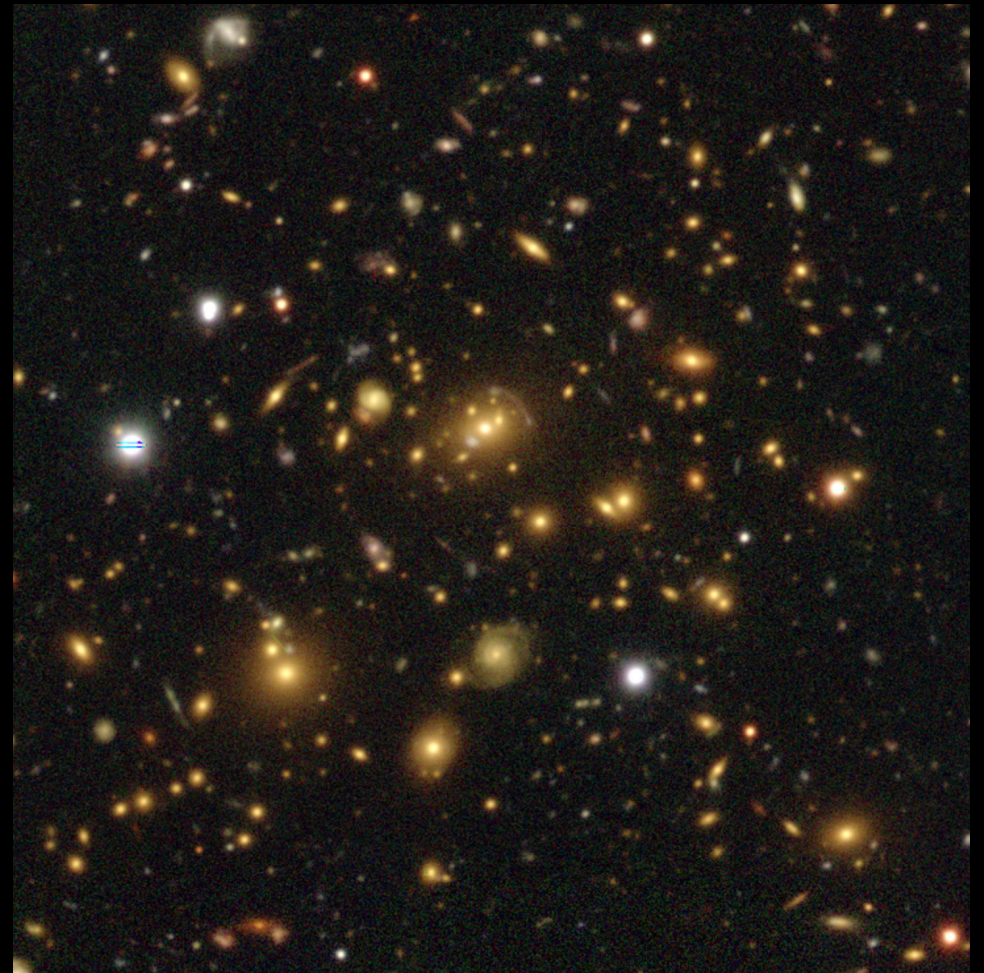


# 機械学習と宇宙論

吉田直紀  
東大理 物理  
カブリIPMU

# 深宇宙画像 before / after 2014



すばるHSCは1晩300ギガバイトのペースでデータを取得



# Large Sky Surveys 2020-

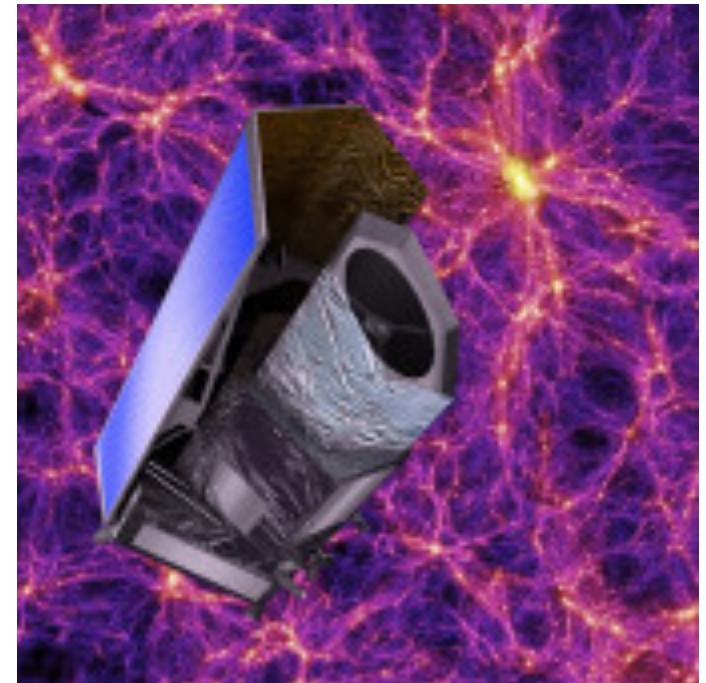
Large Synoptic Survey Telescope  
1晩で20テラバイト Google が支援  
10年で100ペタバイトとか



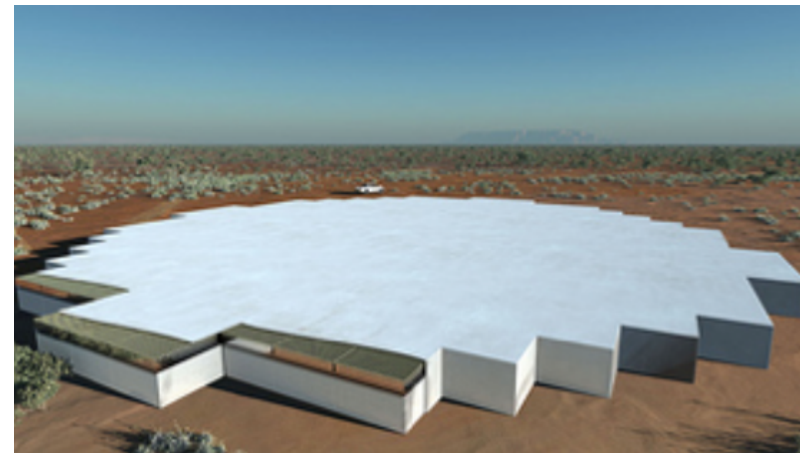
ESA's Euclid  
撮像 + 分光



NASA's Wide-Field Infrared Survey Telescope



# The Square Kilometre Array



- 11か国協同で2024年完成を目指す
- 南アフリカとオーストラリアに～1,000,000個のアンテナ
- 毎日生データは 2-3 Exabyte、保存するのは1Pbyte  
(iTunes 1 exabyte 分を聴くのに200万年かかる)
- データ処理に「京」コンピューター100台が必要

電力調達、データ転送ケーブルなどなど多くの課題  
どのデータを残しておくべきか誰にもわからない



*“The SKA is as much a supercomputing system as it is a telescope. It will require supercomputers faster than any in existence in 2016, and network technology that will generate data traffic at a comparable level to the entire current Internet.”*

*- Rob Adam in “SKA and Big Data”*

一方、世の中では



人工知能による画像認識はここまですんでいる。

Wow I can't believe that worked



a group of people standing  
around a room with  
remotes  
logprob: -9.17



a young boy is holding a  
baseball bat  
logprob: -7.61

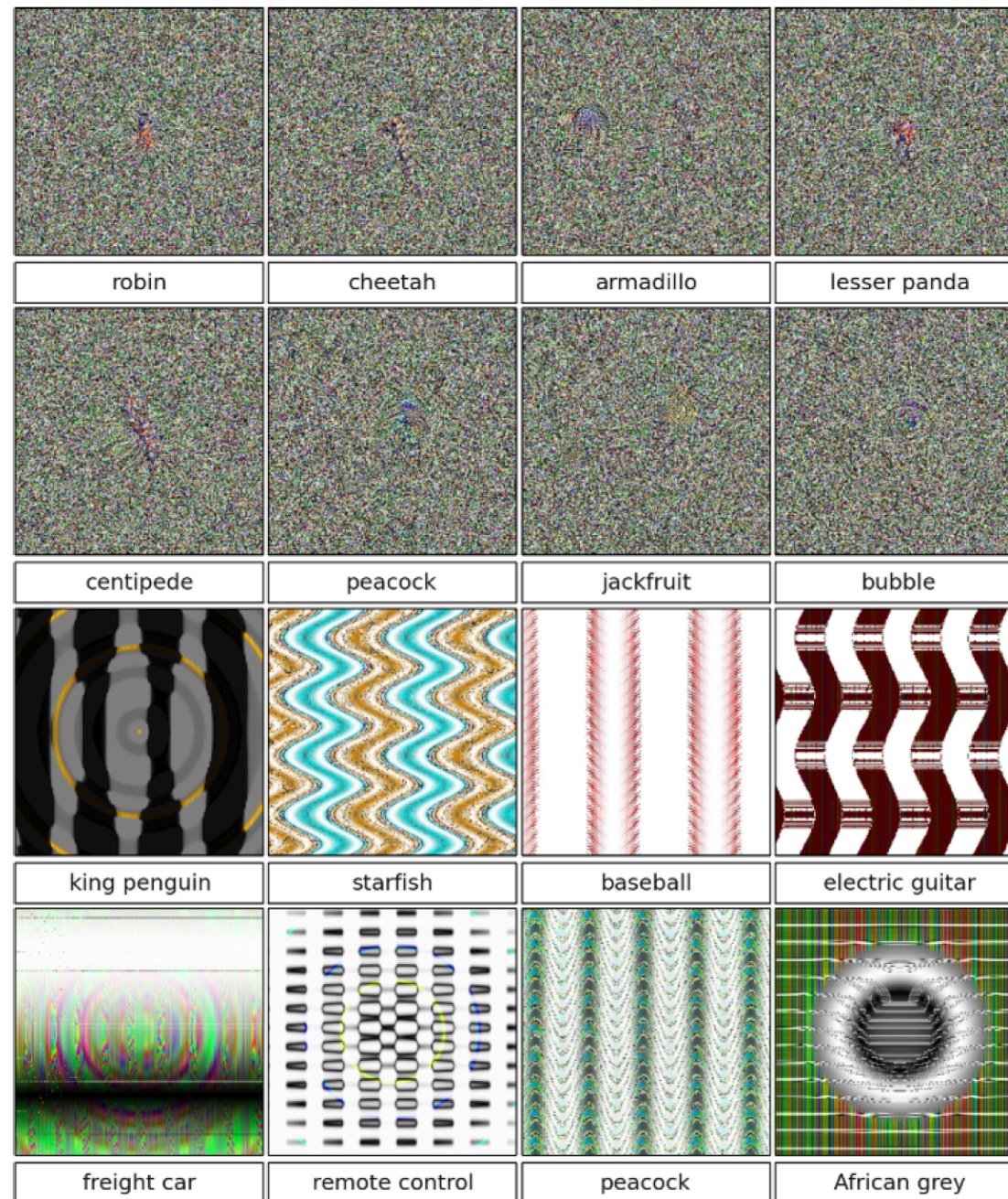


a cow is standing in the middle of a street  
logprob: -8.84

スタンフォード大学院生 Karpathyさん講演スライドより



# Convolutional neural network は物体をこんな風に認識している



A. Nguyen et al. "Deep Neural Networks are Easily Fooled, CVPR15, IEEE, 2015

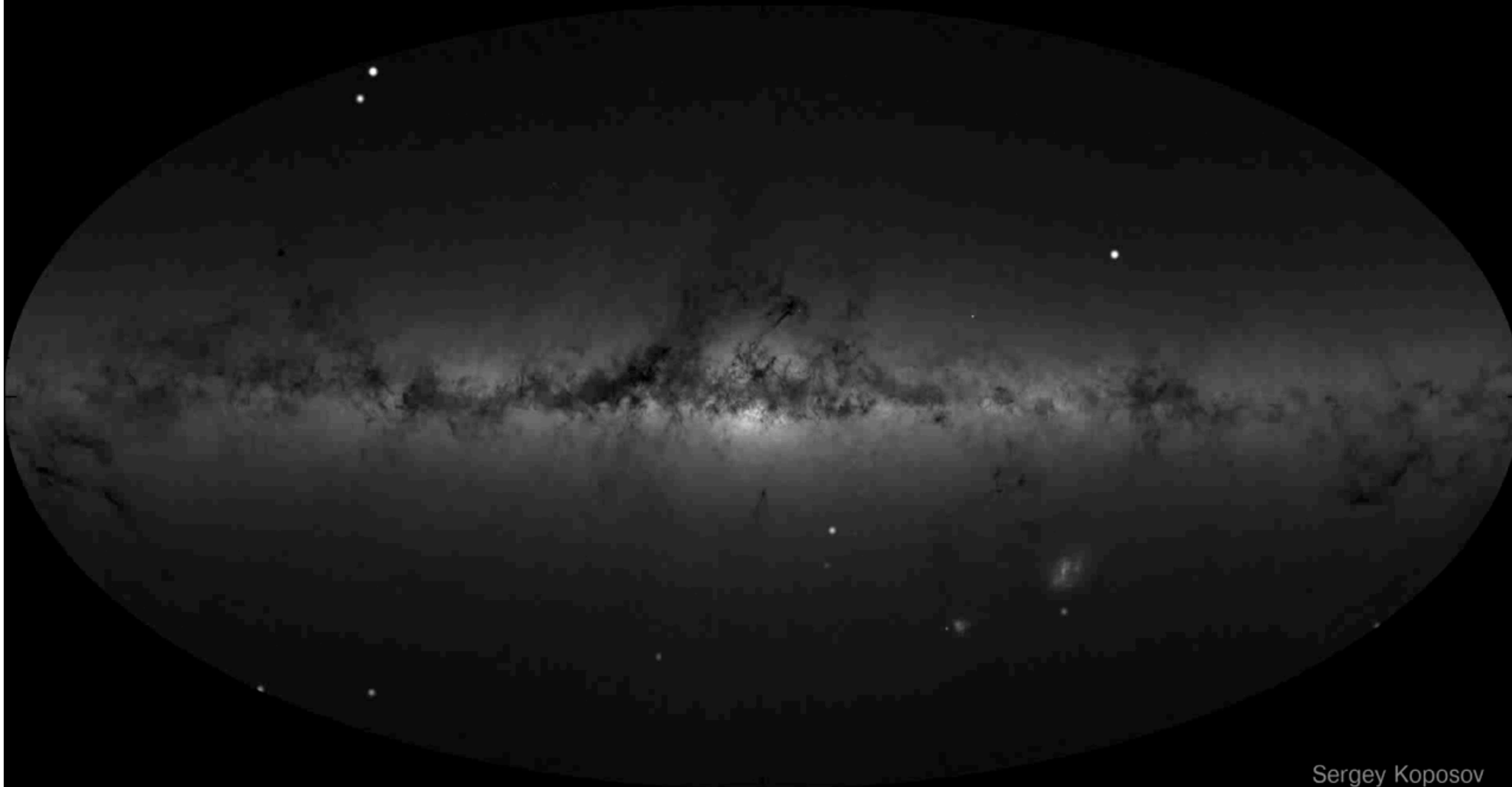


# 今日のお話

- 機械学習を用いた超新星の検出・分類
- 深層学習を用いた重力レンズ解析と  
Intensity Mapping の情報抽出
- 将来の話: サイエンスするAI

# Time Domain Astronomy

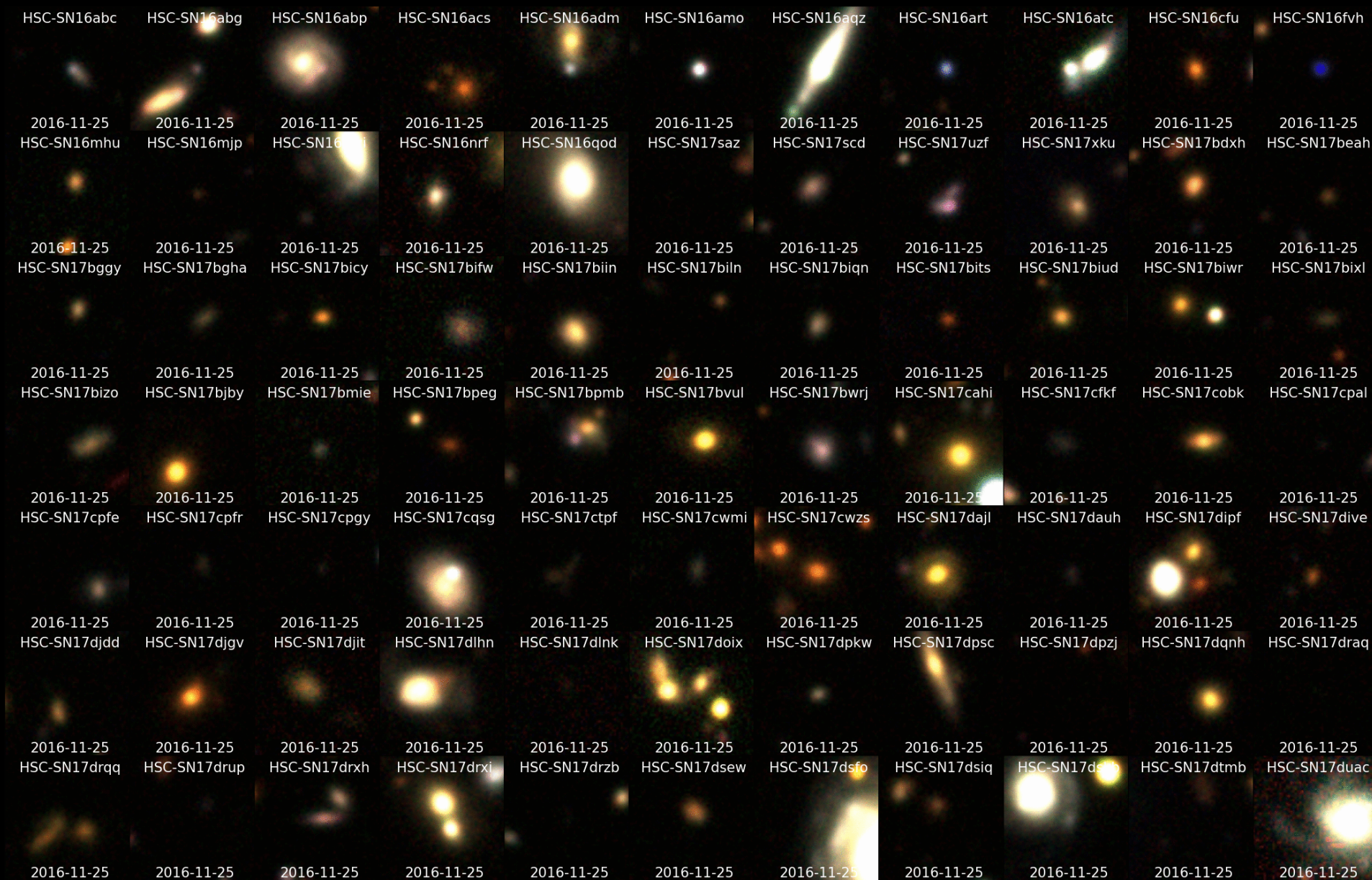
Year 1985.000



Sergey Koposov



# 超新星ギャラリー: 1晩で100個ほど見つかる





2014Apr

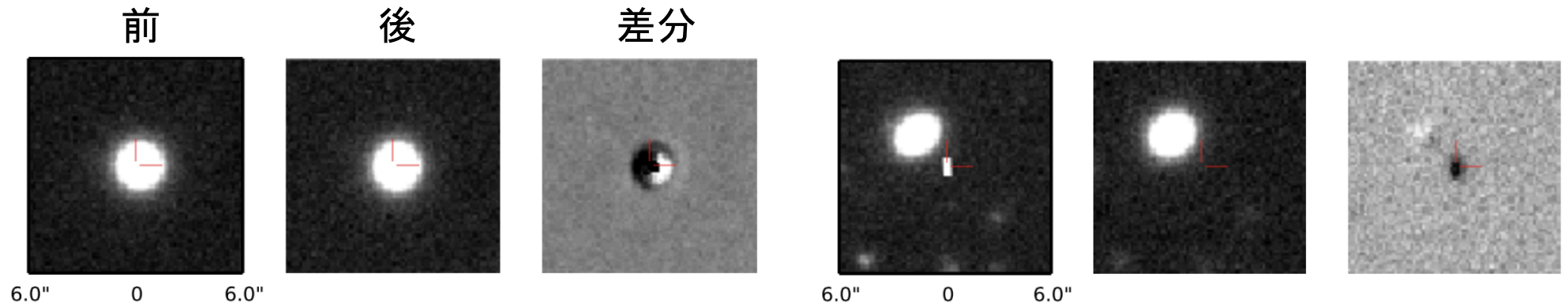
# レベル99の世界

1ギガピクセルの画像  
差分処理

2015May

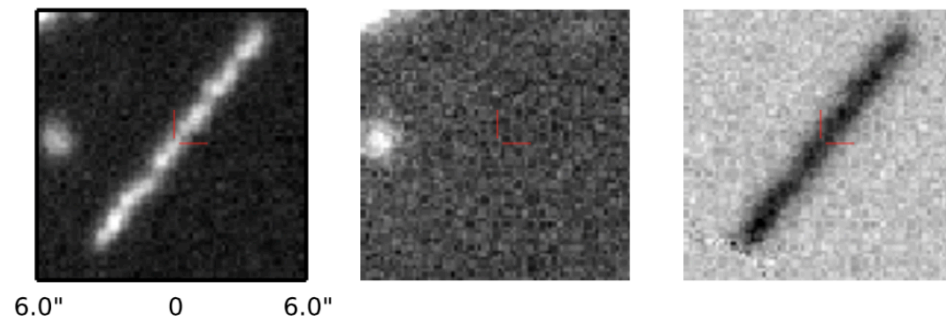


# 差分解析後

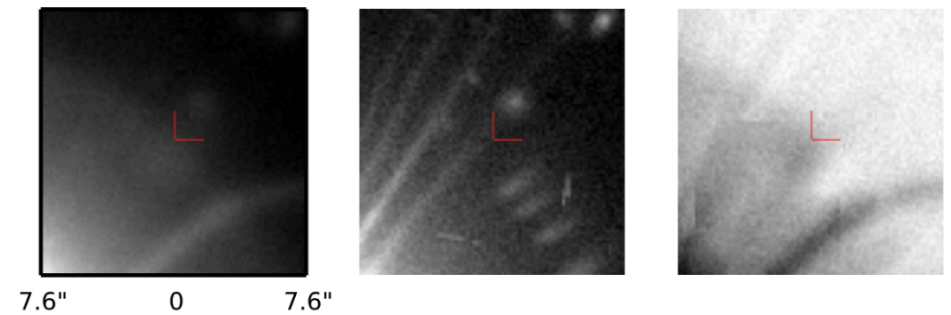


(a) bipolar (subtraction problem)

(b) cosmic ray



(c) moving object



(d) diffraction spike

このような“ごみ”と実天体の数の比は 1000:1 (= 10000:100)

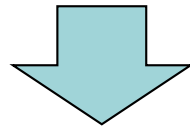
# 宇宙観測に特有の問題

## Data imbalance:

膨大な負例 (CCDエラー, 宇宙線, 太陽系小天体)  
に対してあまりに少ない正例

## Evaluation criterion:

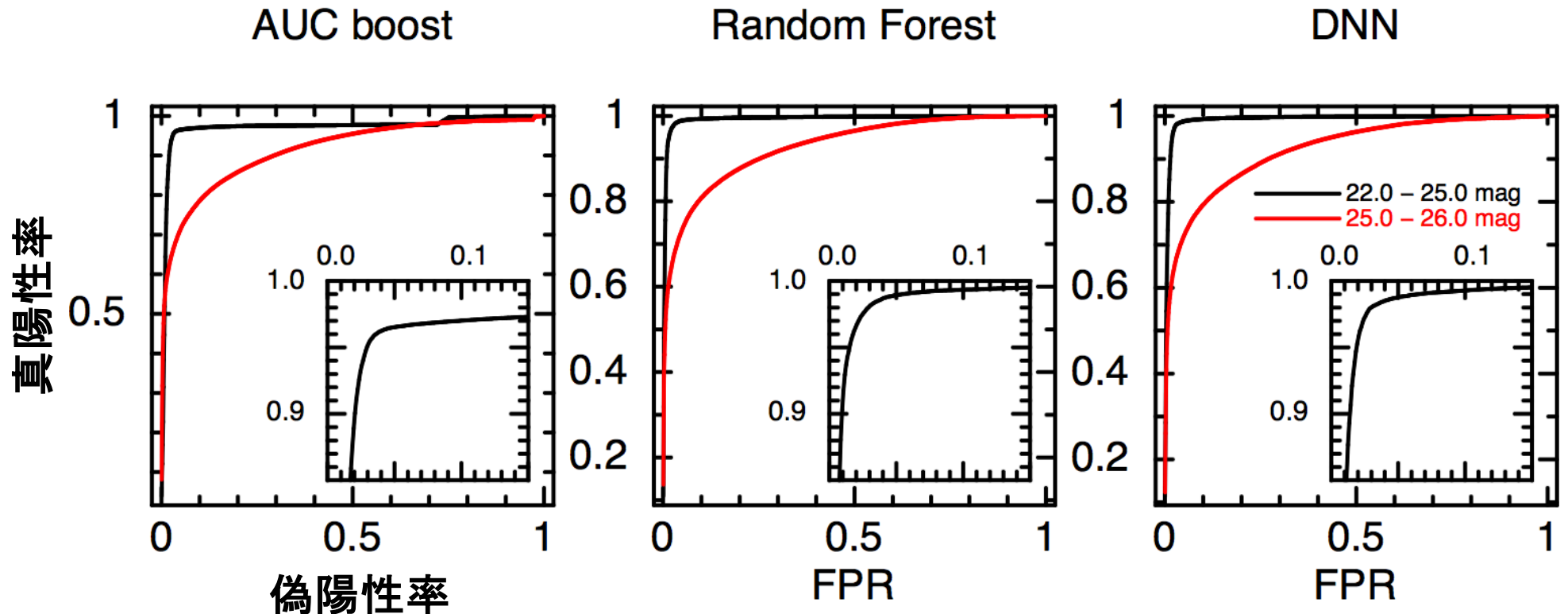
偽陽性率はできるだけ小さく (追観測の苦勞)  
真陽性率は大きく (興味深い天体はのがさない)



新たな機械学習検出・分類法を考案

# 機械学習による超新星検出

- ・トレーニングデータ: HSCサーベイから 24000 オブジェクト  
人工天体や data augmentation も利用
- ・分類器: Random Forest, 深層ネットワーク (DNN)  
Boosting by AUC optimization
- ・23 次元特徴量 and/or 2 次元画像



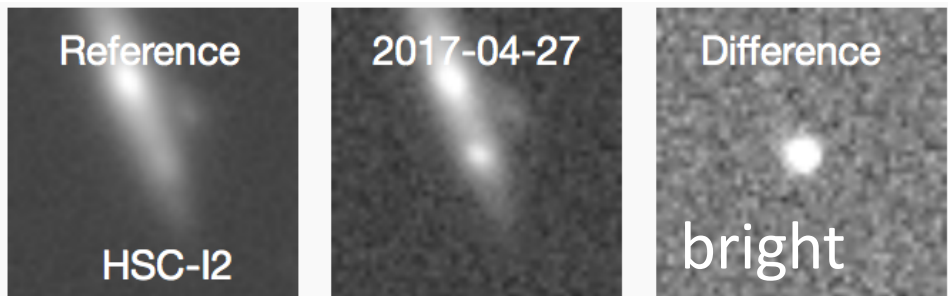
# 暗い(遠方)超新星



magnitude 21.7

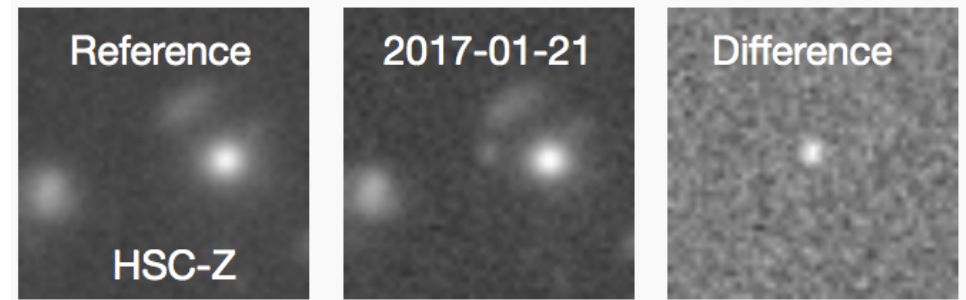


magnitude 22.7



magnitude 23.1

The power of Subaru  
magnitude 25.1



magnitude 25.3

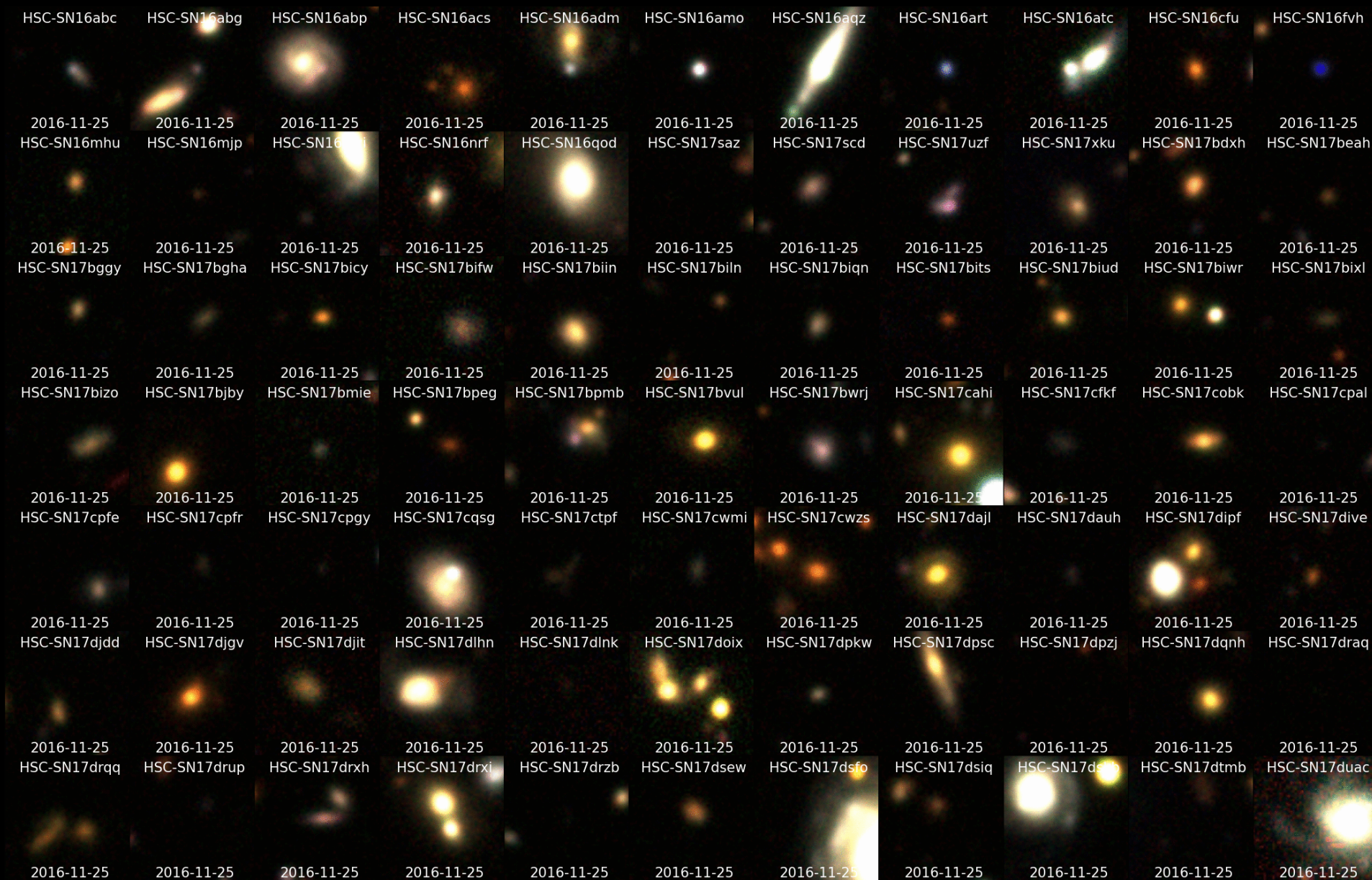


magnitude 25.7

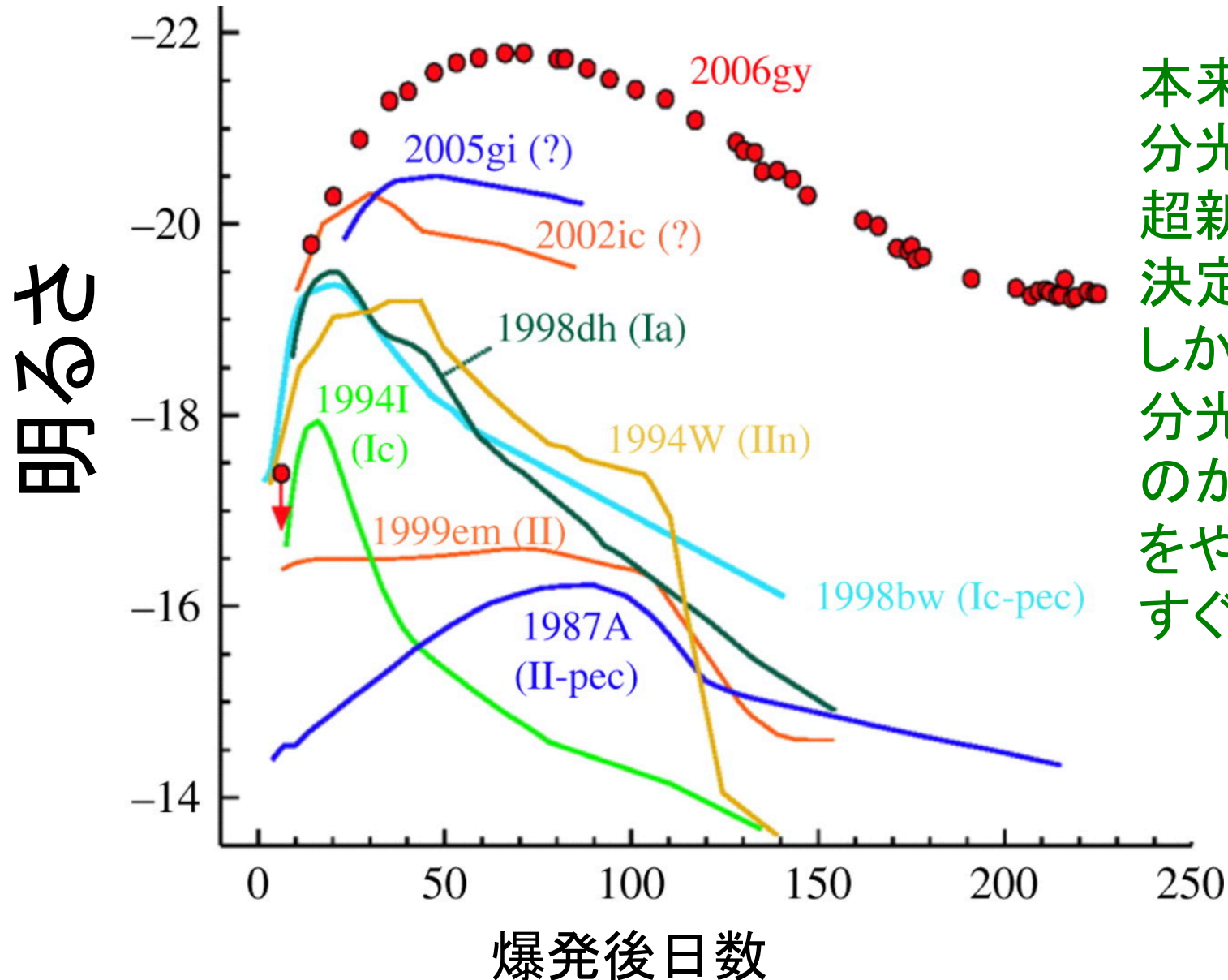




# 超新星ギャラリー



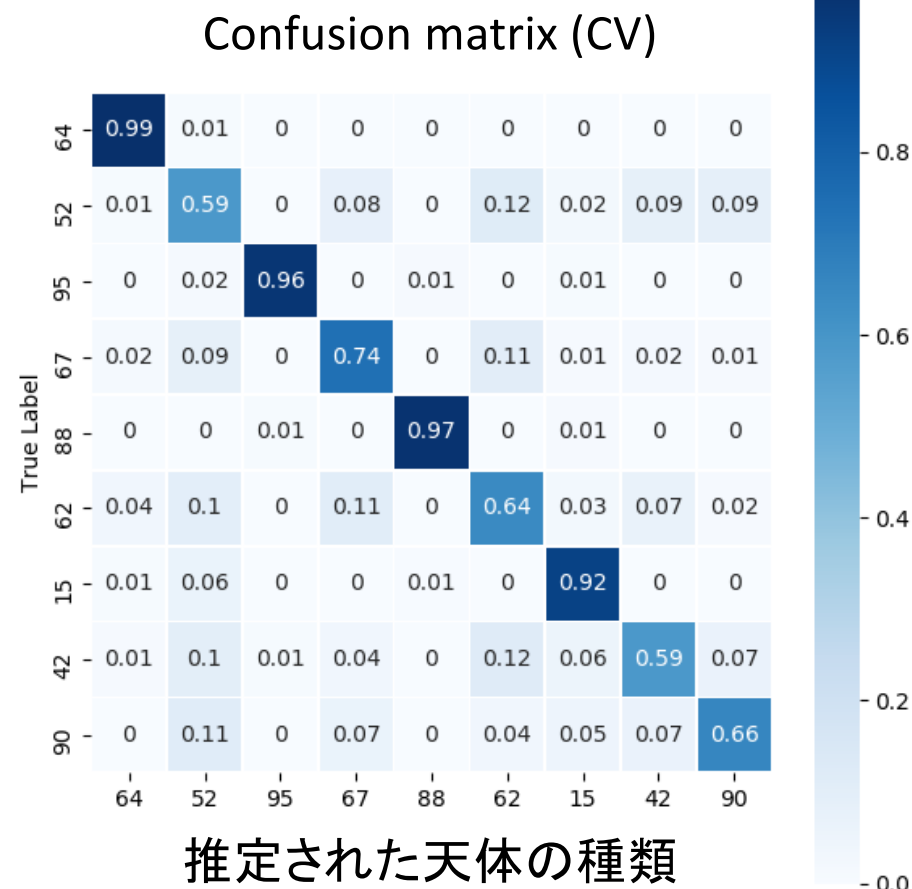
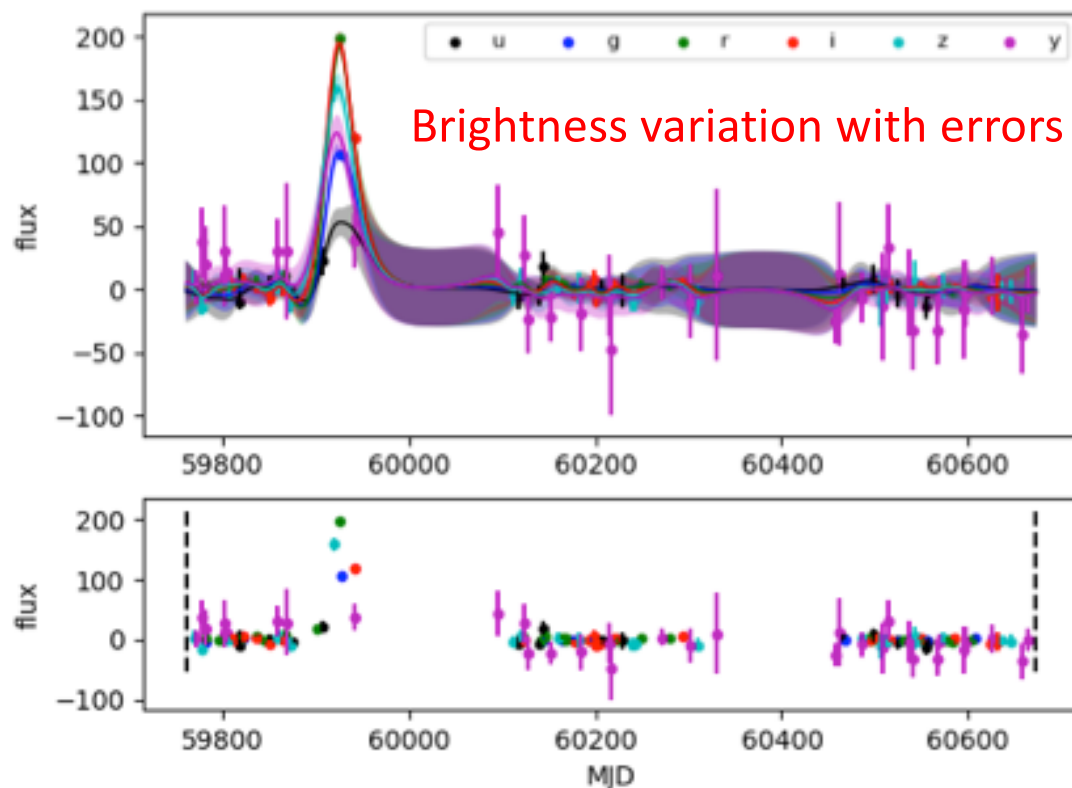
# 検出から分類へ



本来は  
分光観測によって  
超新星タイプを  
決定する。  
しかし、  
分光(という時間  
のかかる作業)  
をやるかどうかを  
すぐに決めたい

# 時間変動天体の即時分類

明るさを変える天体は銀河系内/系外に多種存在する

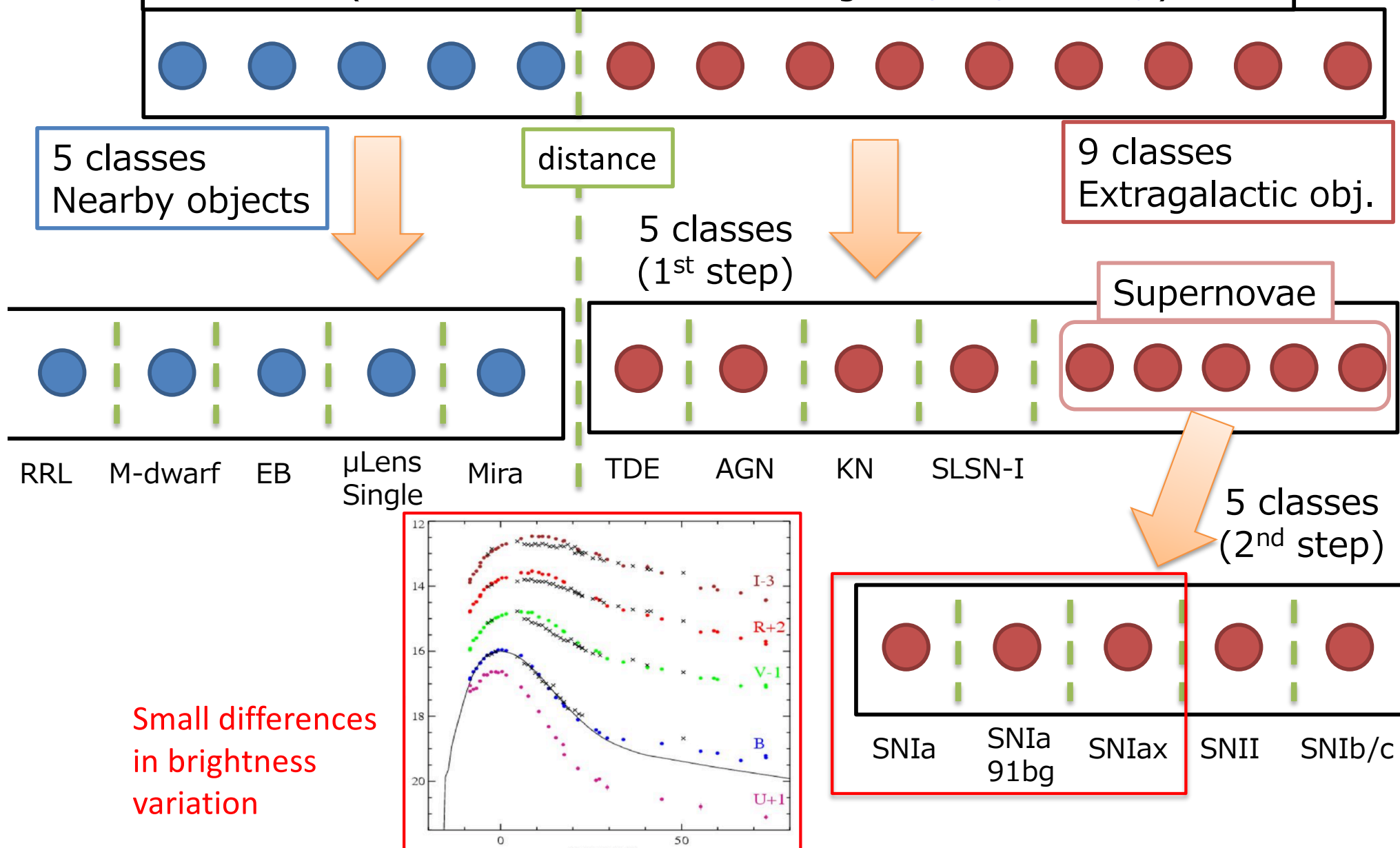


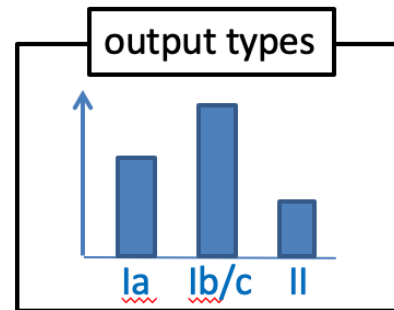
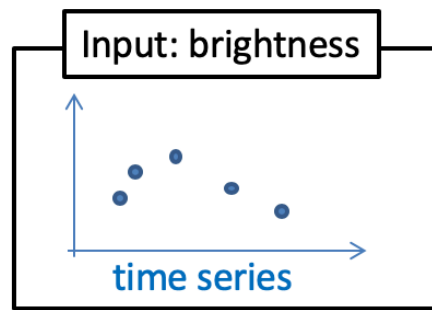
I. Takahashi (U-Tokyo), Y. Imoto (NTT), in preparation



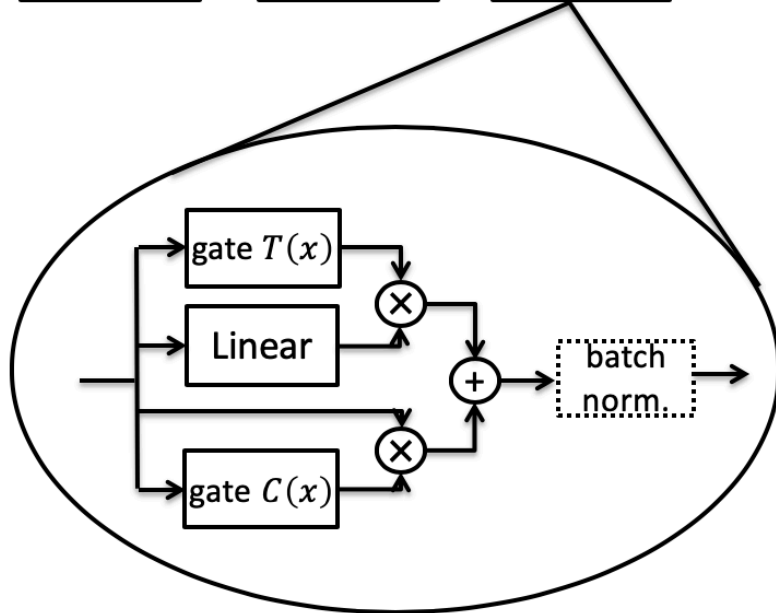
# 階層的分類器

14 classes (data from PLAsTiCC challenge <https://plasticc.org>)





370,000 データセット  
で訓練



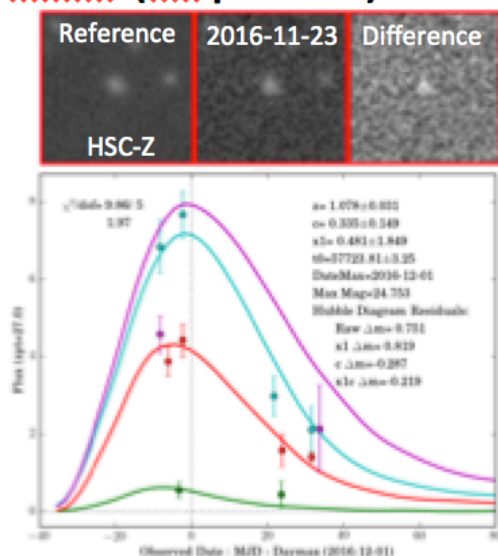
Classifier performance

True label	Ia	Ibc	II
	0.96 (964)	0.01 (8)	0.03 (28)
	0.06 (9)	0.74 (119)	0.20 (32)
	0.02 (21)	0.01 (9)	0.97 (1107)
Predicted label			

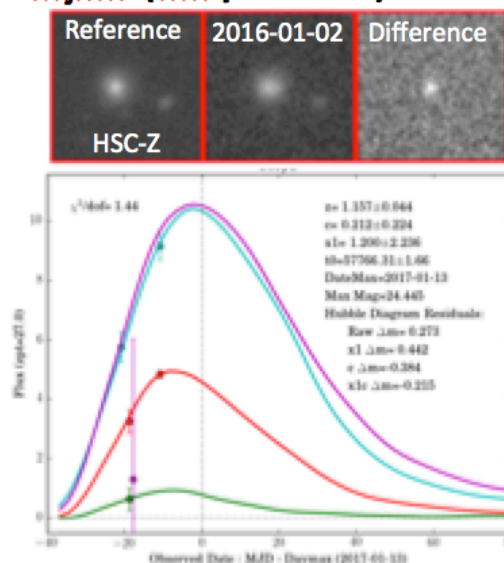
Cohen's kappa:  $(0.953 - 0.444) / (1 - 0.444) = 0.9162$

# 実際に見つけた遠方Ia型超新星

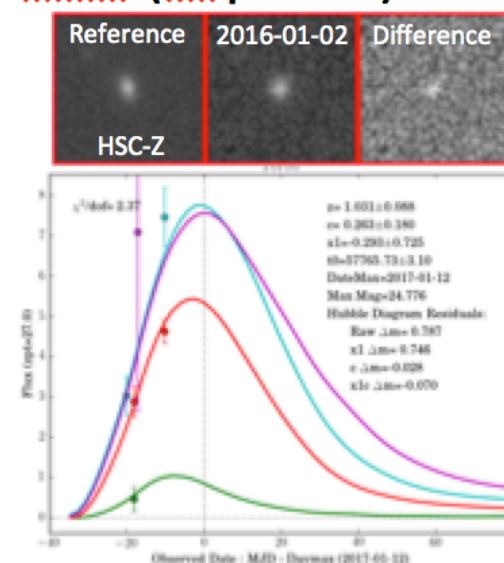
caa (Ia pred.)



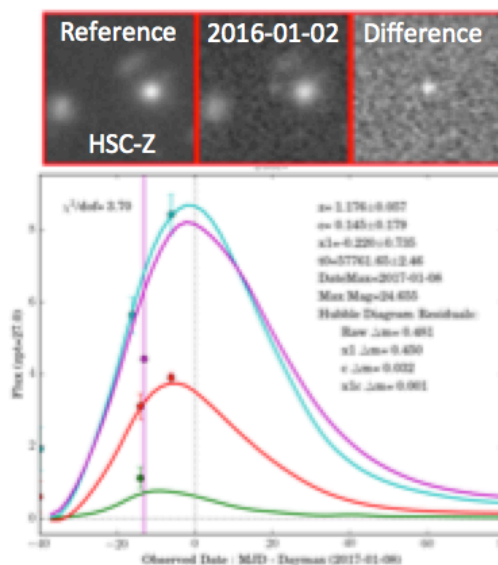
ryz (Ia pred.)



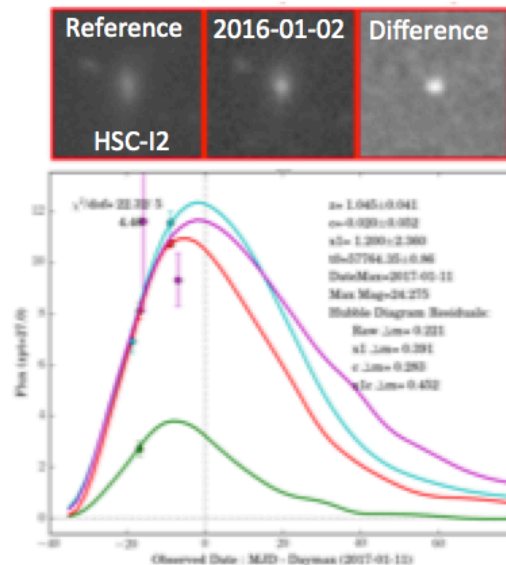
roo (Ia pred.)



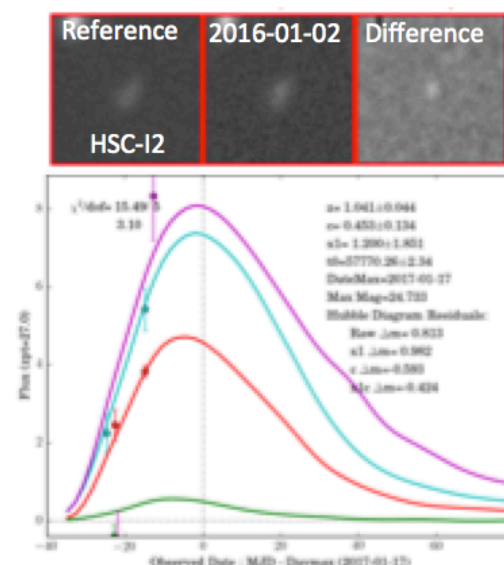
siv (Ia pred.)



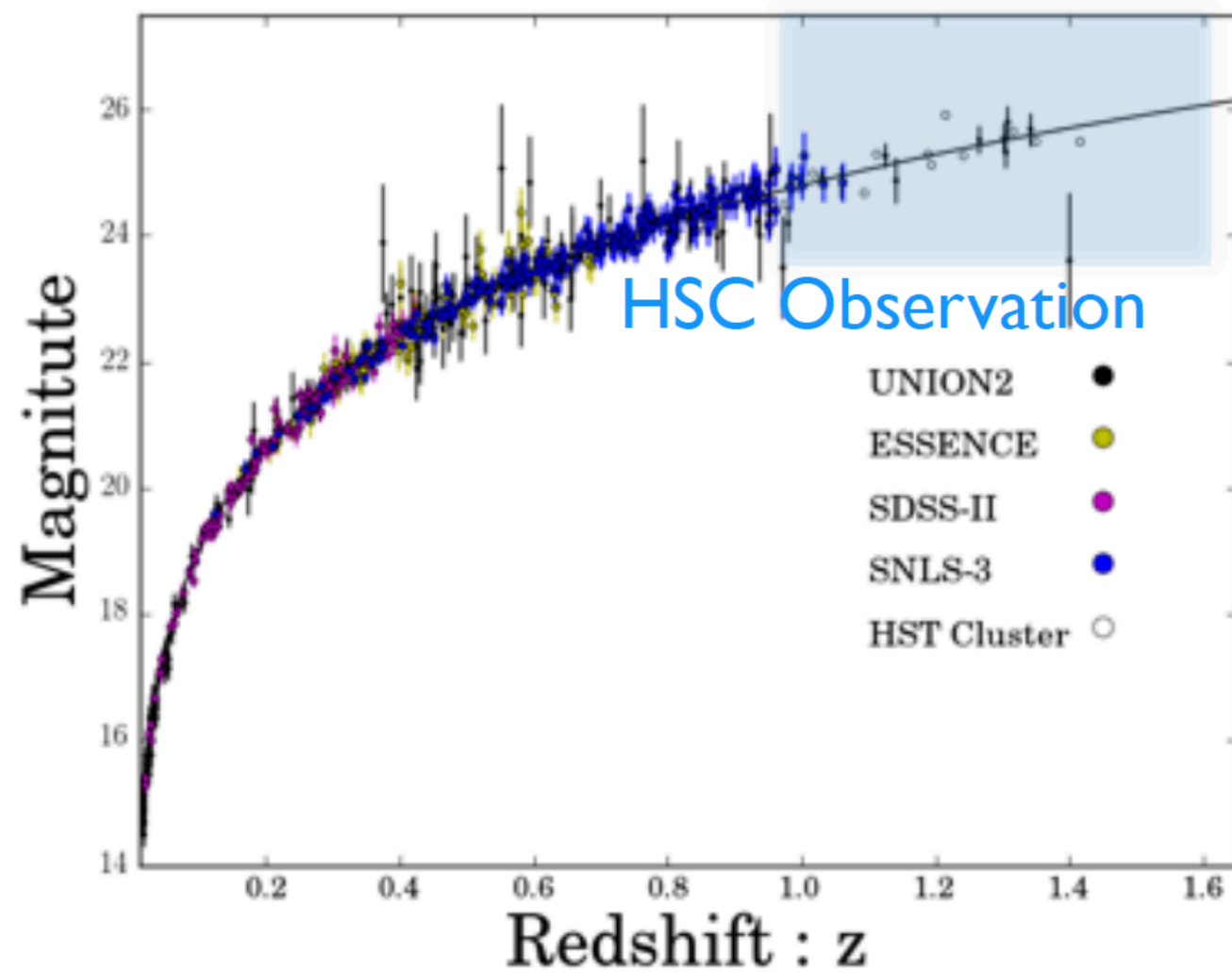
agj (Ia pred.)



vlr (Ia pred.)

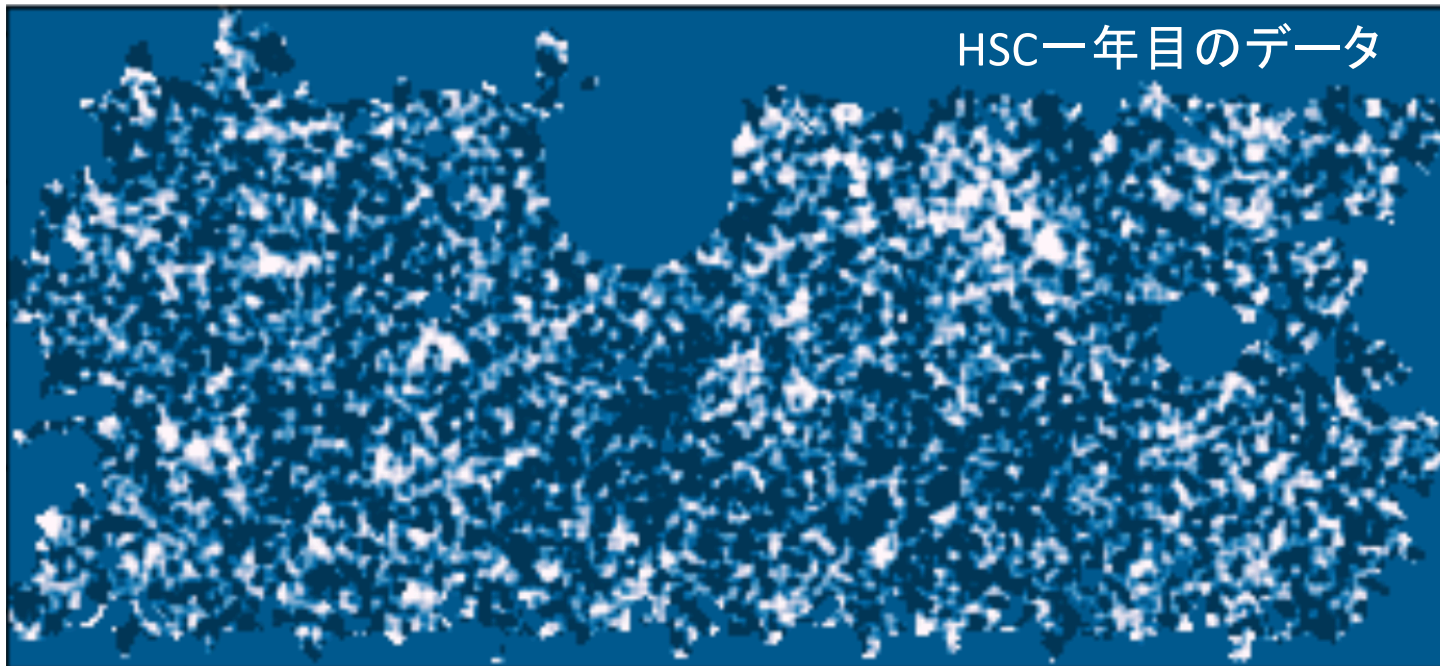




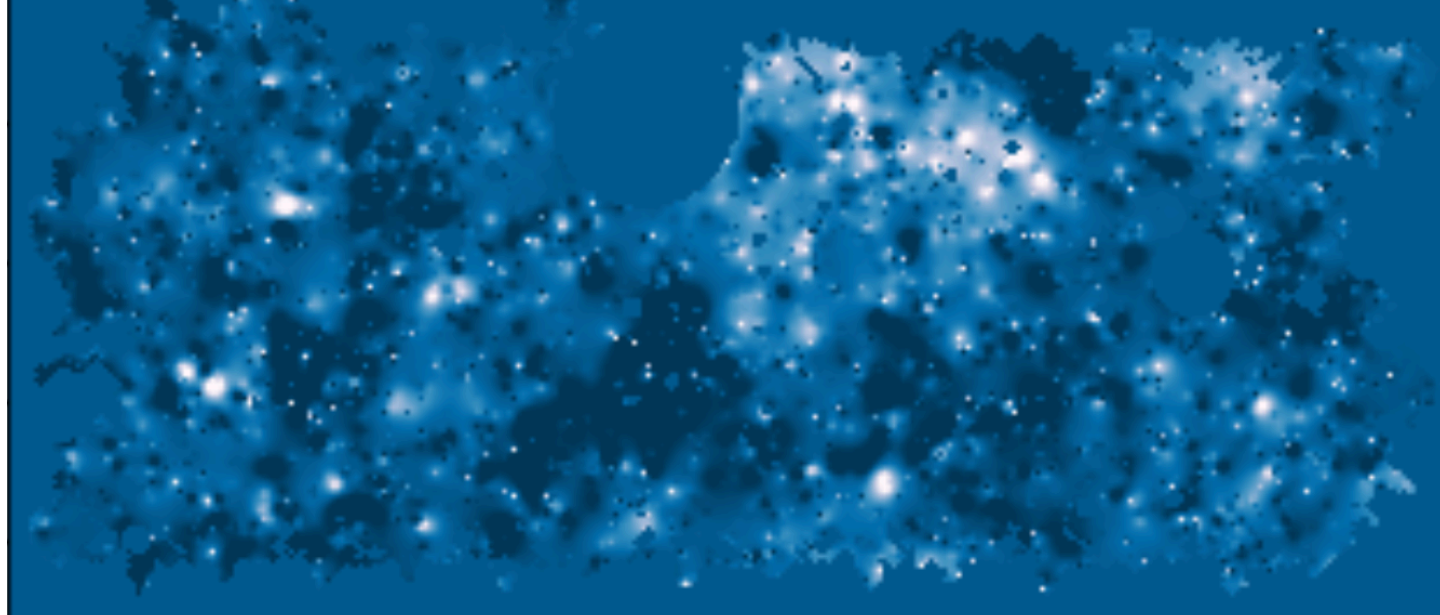


# 深層学習を用いた 宇宙大規模構造の解析

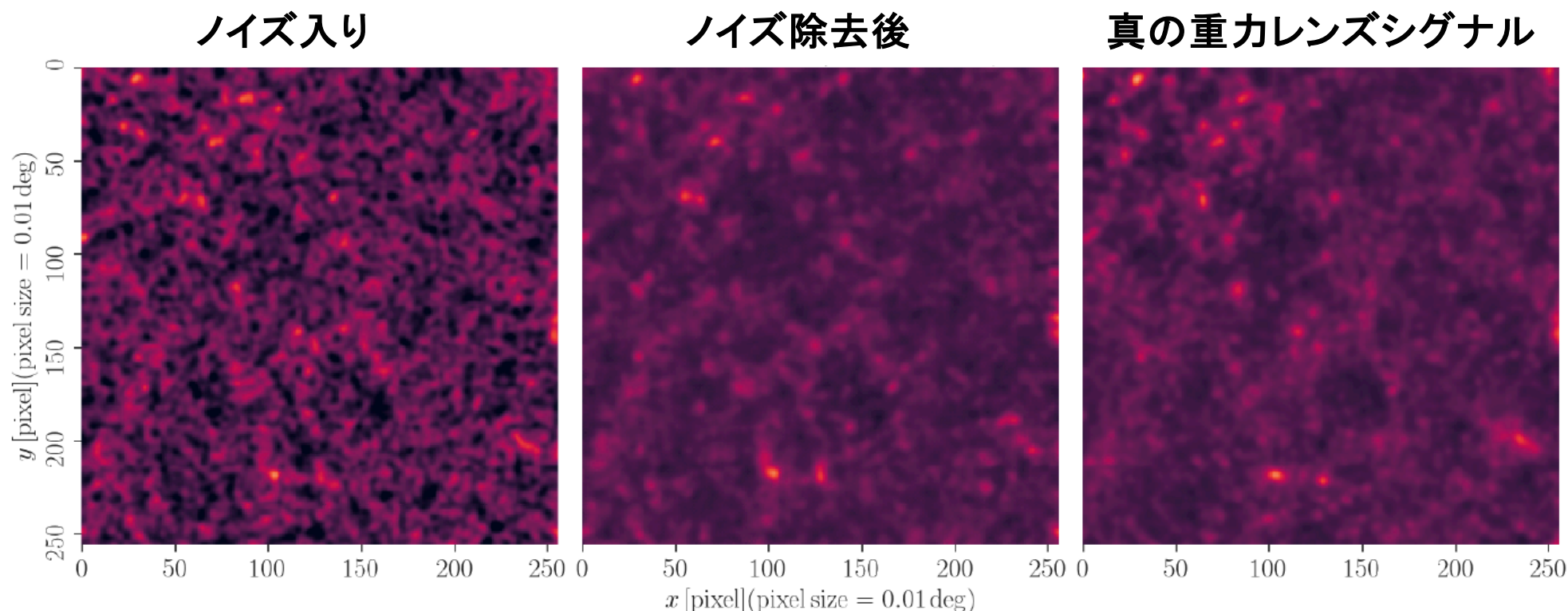
# HSC重力レンズマップ



大学院生 Xiangchong Li さんのスパースモデル再構築



# 深層学習によるノイズ除去: 機械にシグナルとノイズを学ばせる

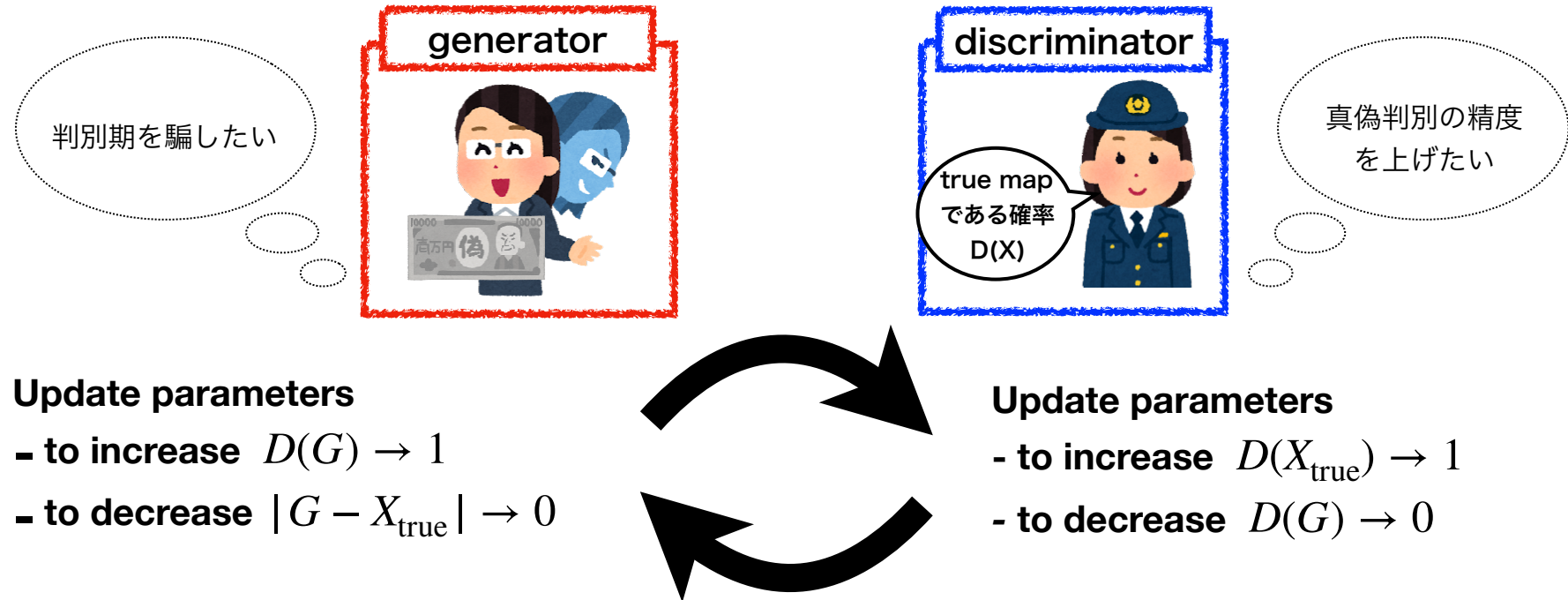


重カレンズの数値シミュレーション + HSC のノイズ (銀河分布)  
を用いたテスト計算

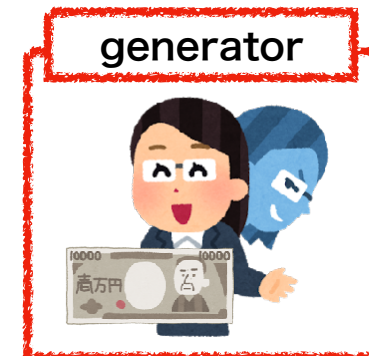


# 敵対的生成アルゴリズム

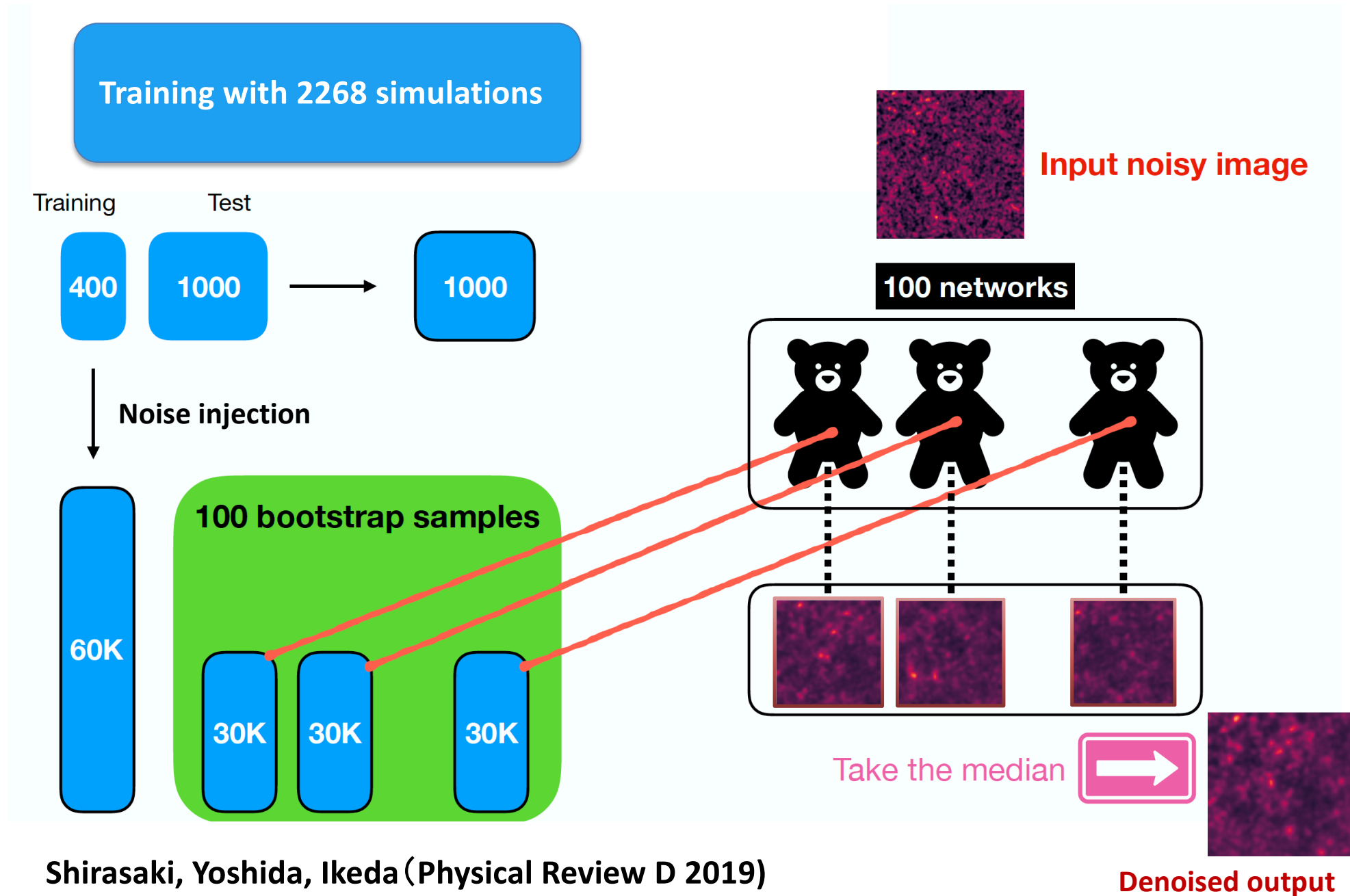
## TRAINING PHASE



“協働”してトレーニングすることにより、  
最終的には真偽判別のつかない画像を生成  
できるようになる

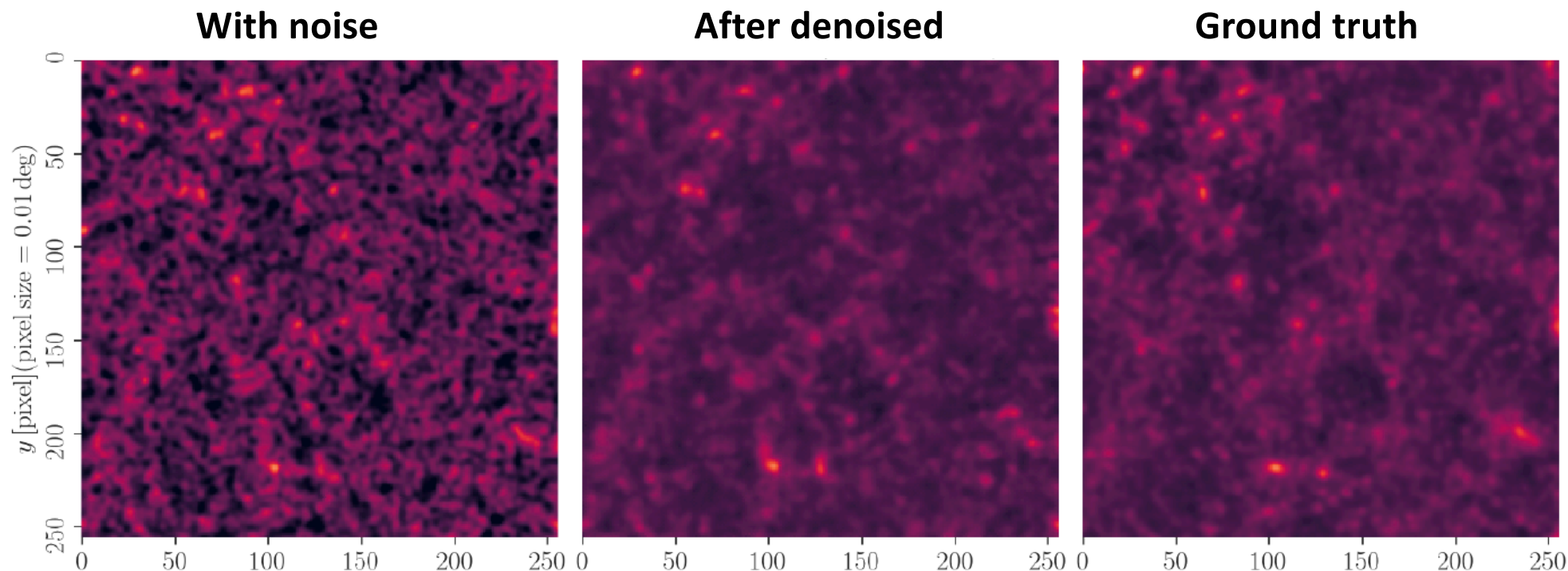


# 敵対的生成アルゴリズム(GAN)によるノイズ除去

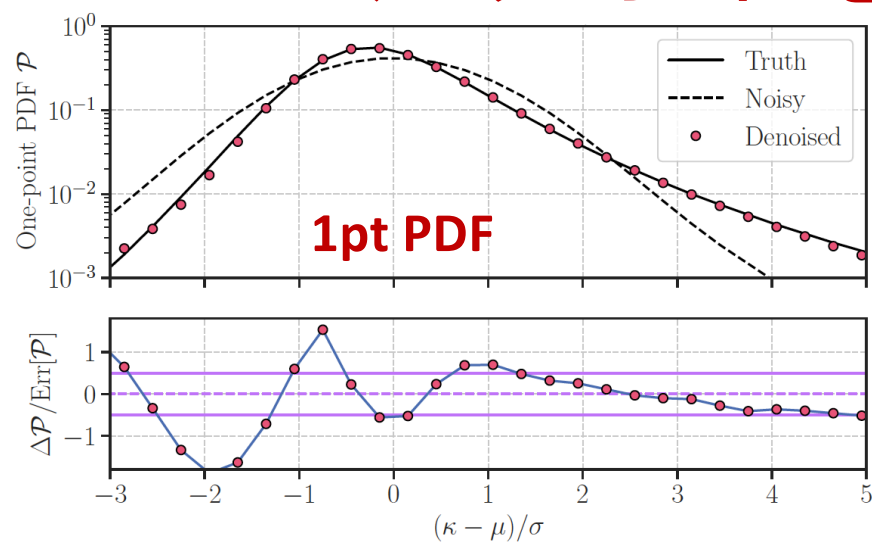




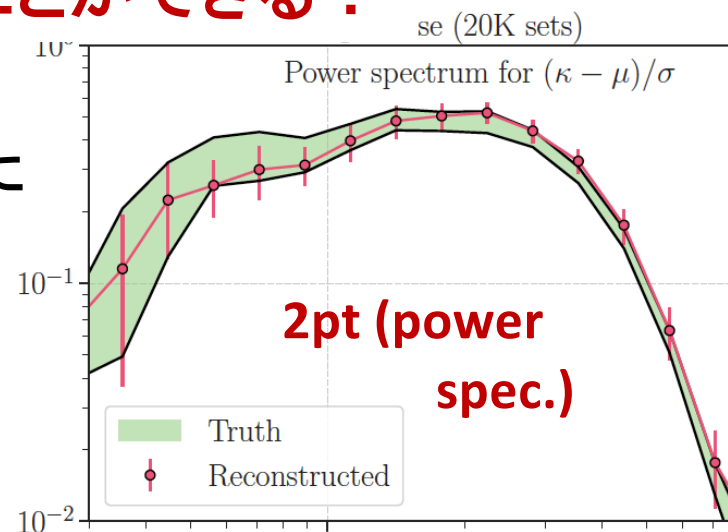
# 敵対的生成アルゴリズム(GAN)によるノイズ除去



データからノイズを低減することができる！



ノイズ除去により統計量も正確に再現

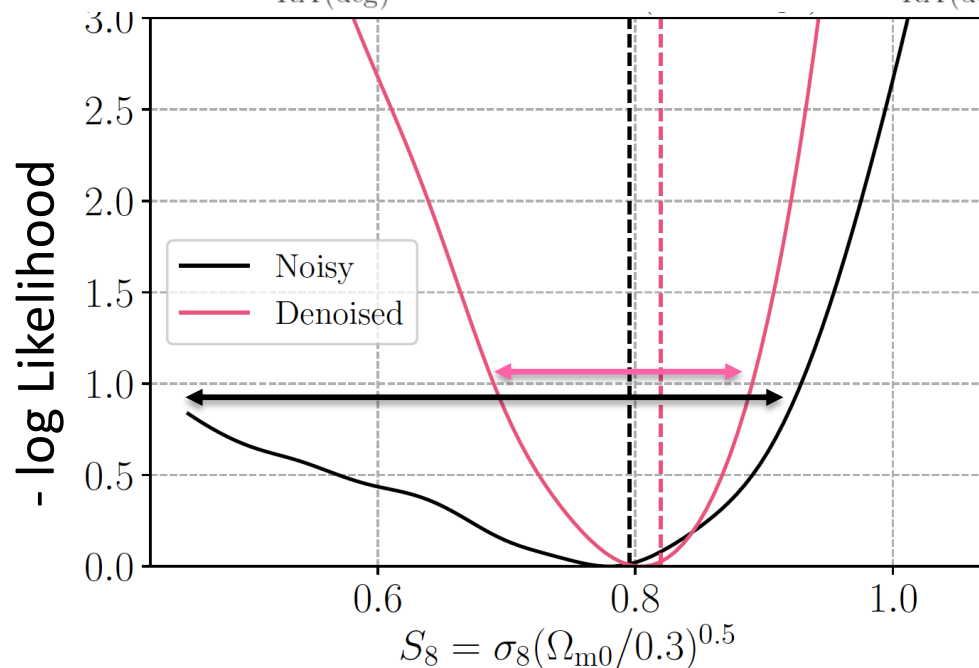
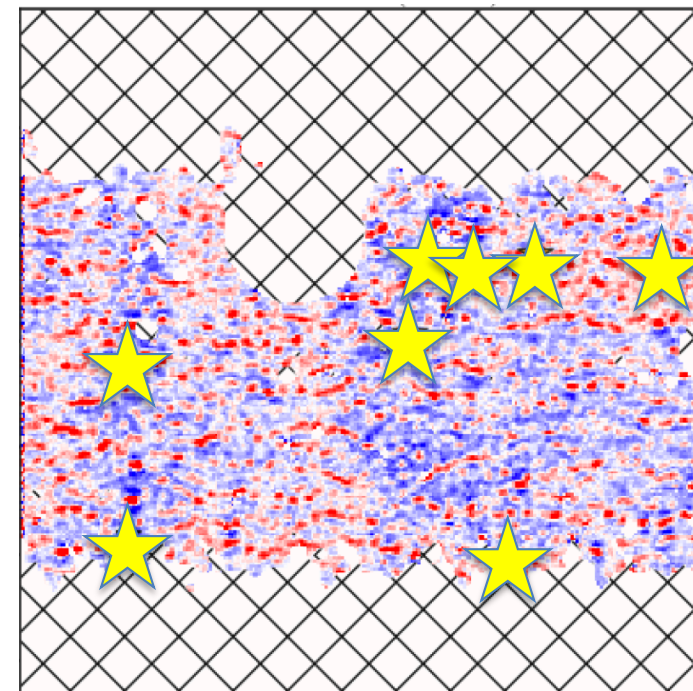
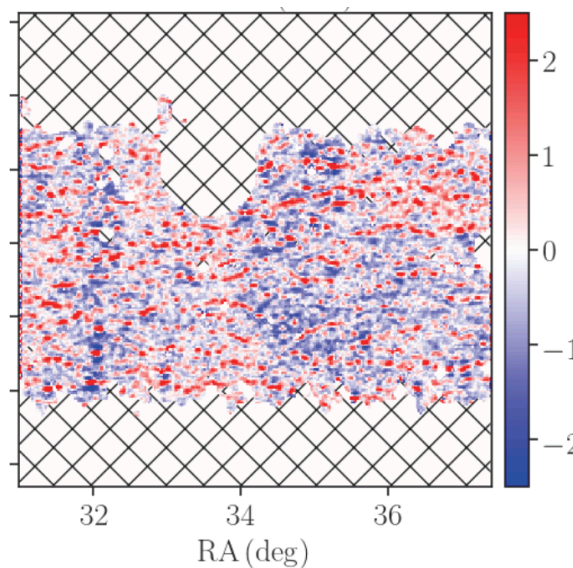
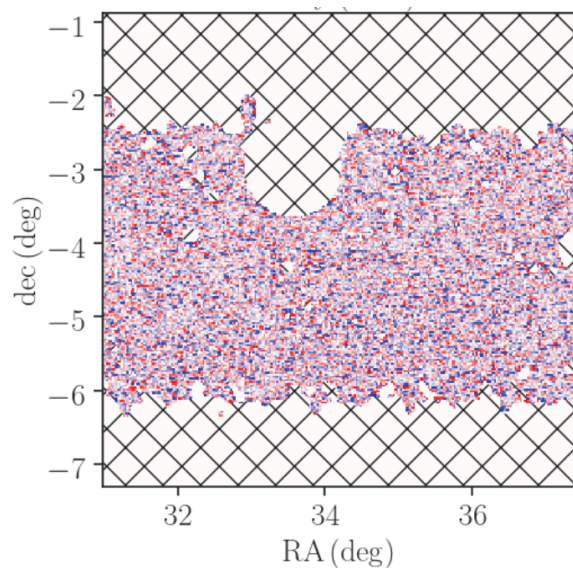


# HSC実データへの適用

With noise

After denoised

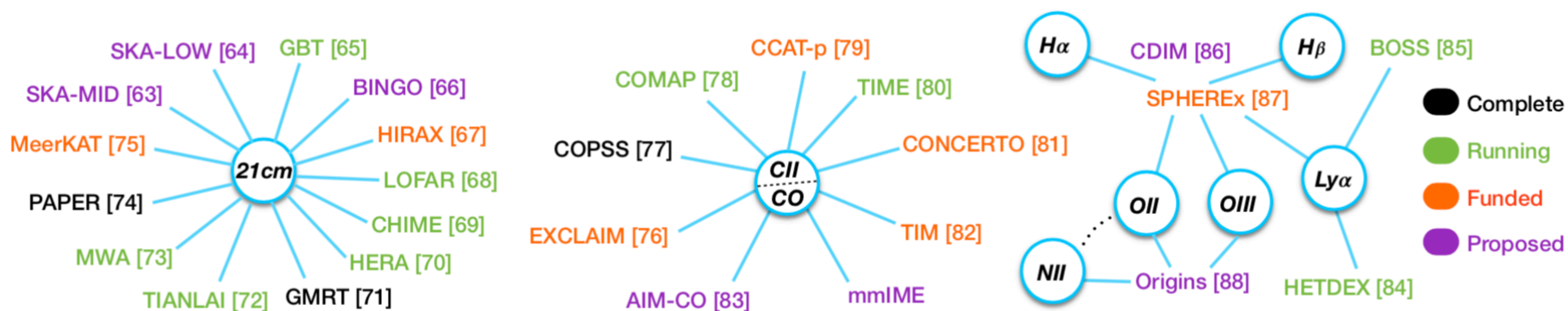
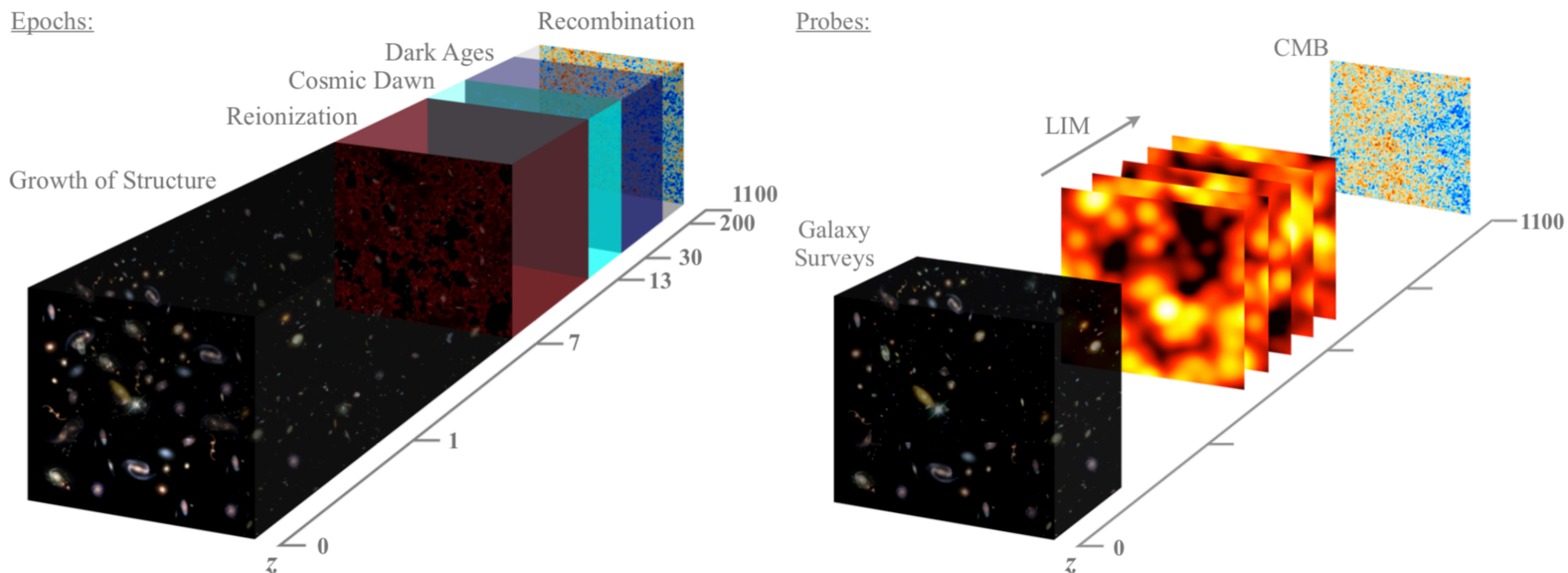
Object matching



With denoising, the parameter inference gets 50 % better. A factor of 2 improvement does not sound impressive, perhaps. But a factor of 2 improvement can be gained only by 4 times larger data (survey area).

To survey a 4 times larger area,  
we need a 20-year(!) HSC observation.

# 輝線強度マッピング





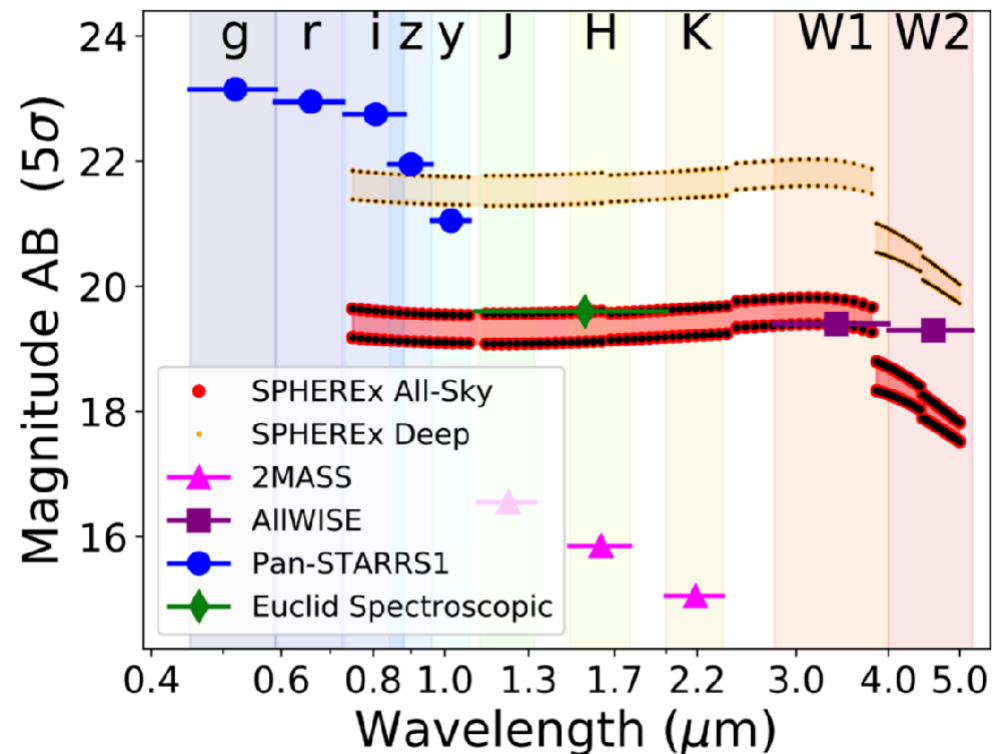
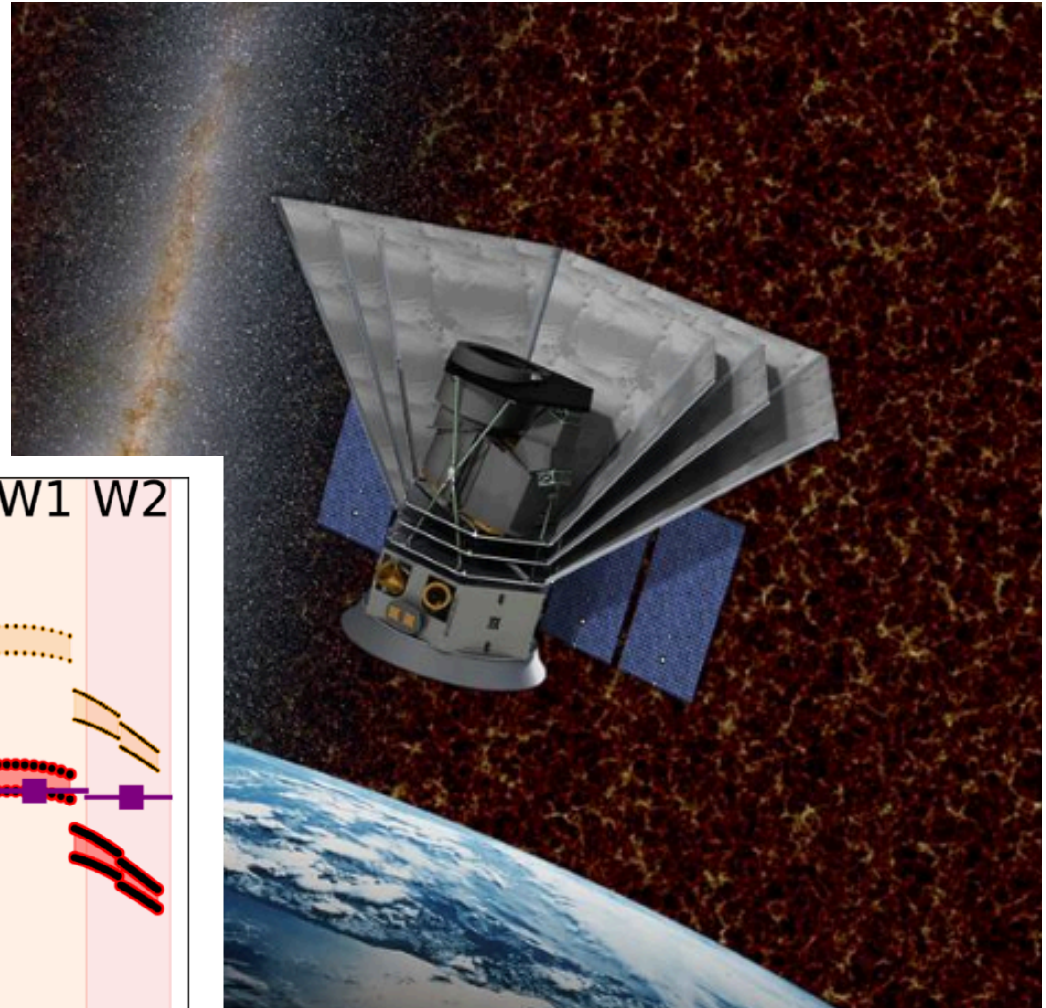
# NASAのSPHEREx 計画

( SPHEREx (2023 打ち上げ)

- angular resolution:  $6'' \times 6''$
- spectral resolution:  $R \sim 100$
- survey area (deep):  $\sim 200 \text{ deg}^2$
- $0.75\mu\text{m} - 5\mu\text{m}$

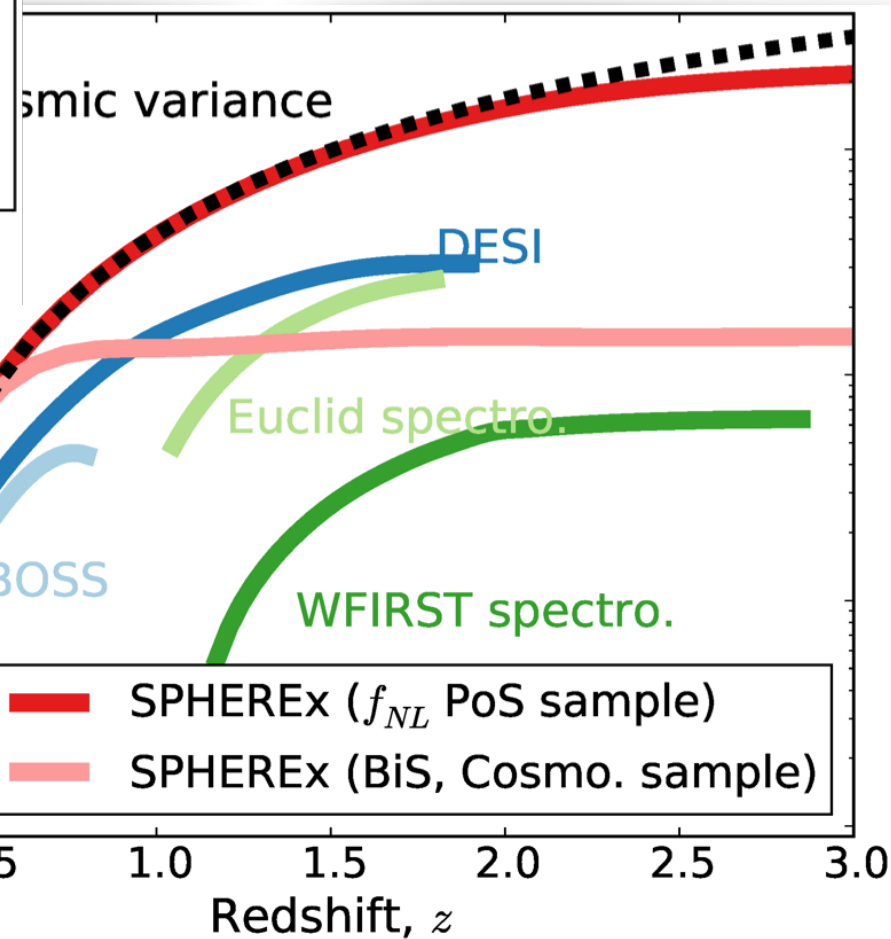
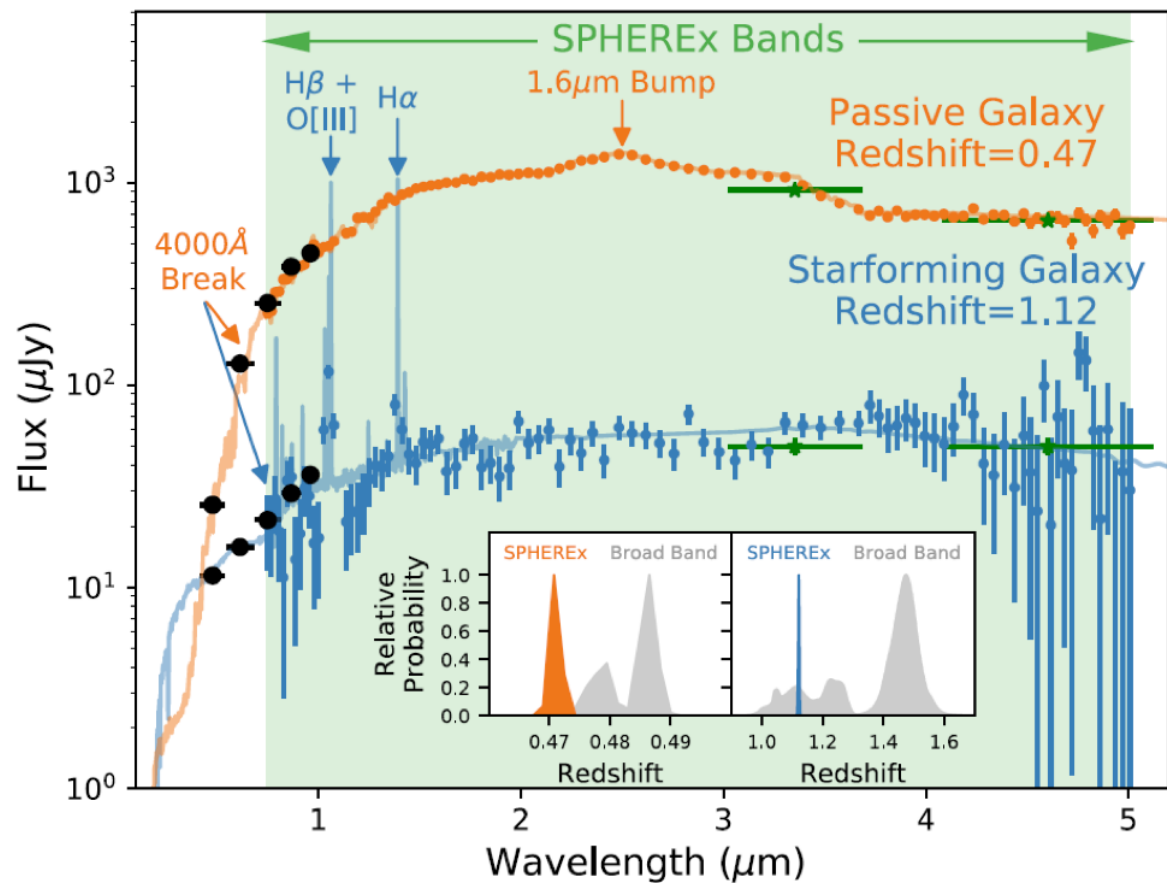
$z_{\text{H}\alpha} = 0.14 - 6.6$

$z_{\text{OIII}} = 0.5 - 9.0$



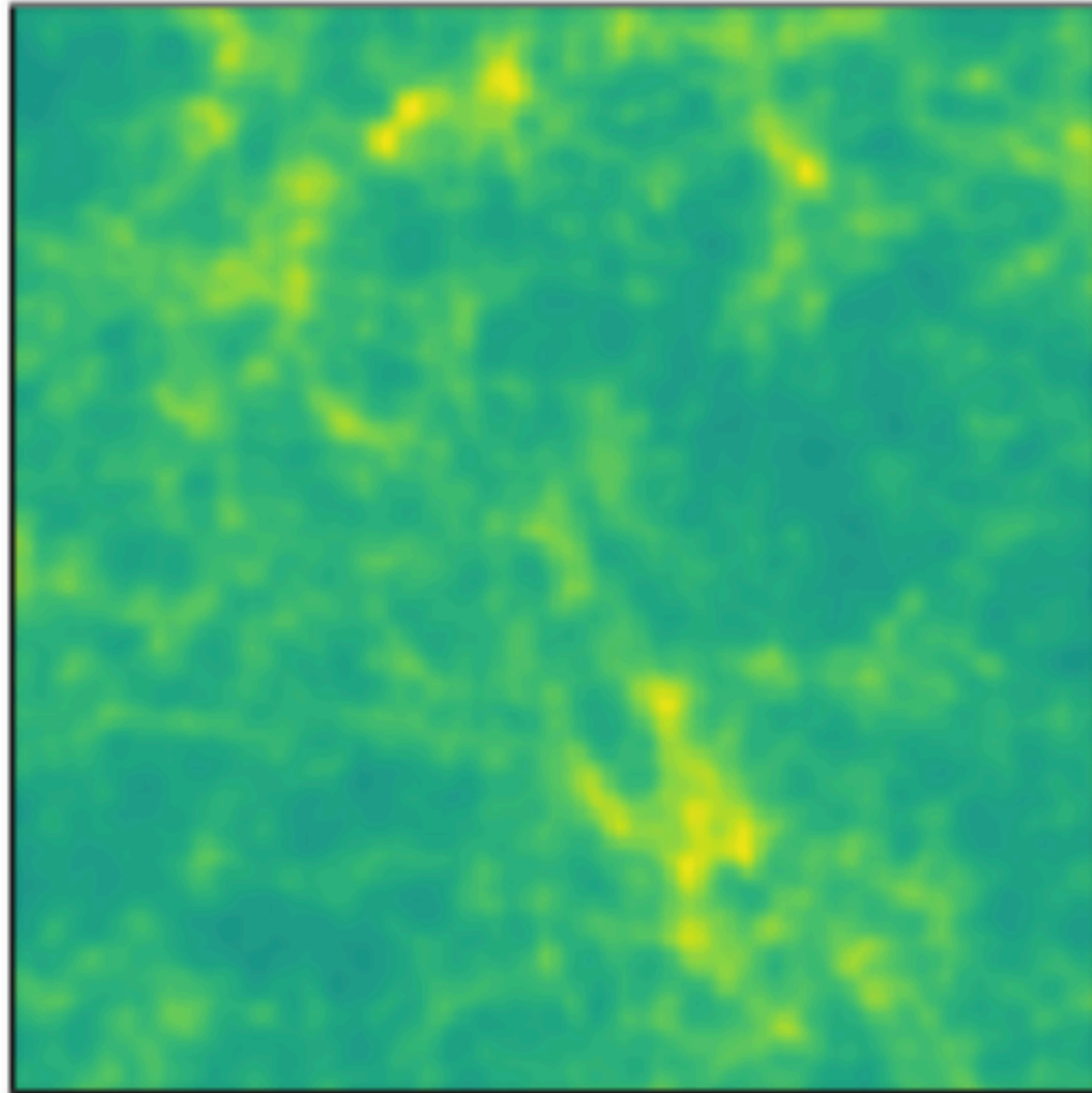


# 波長帯域

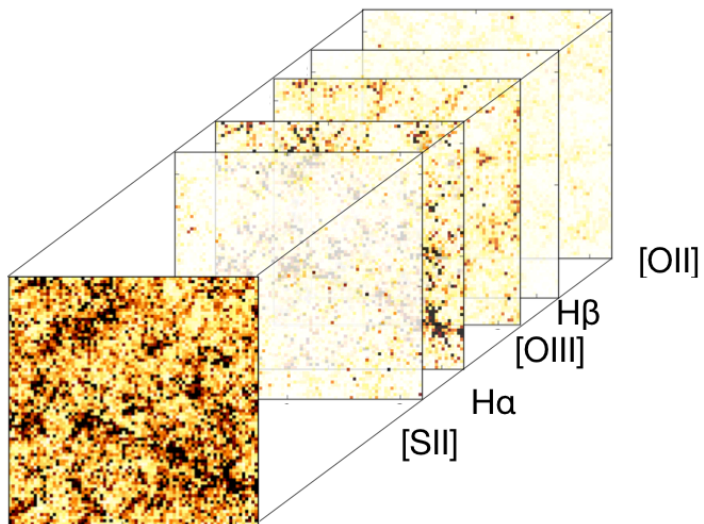


## 観測体積

# 例：2 $\mu\text{m}$ で見た宇宙



遠方近傍関係なく、2ミクロンで光るものの全てが混じった画像を得る(得てしまう)



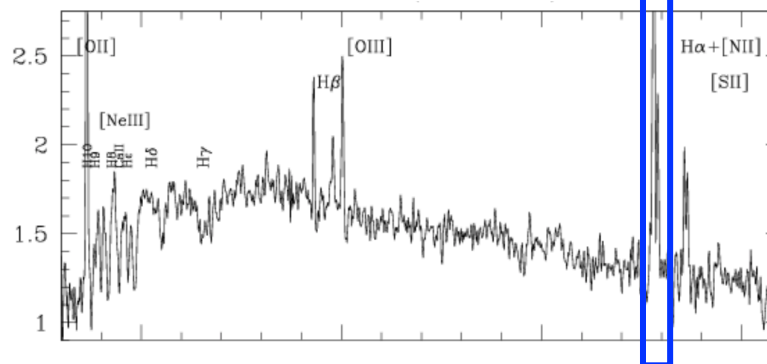
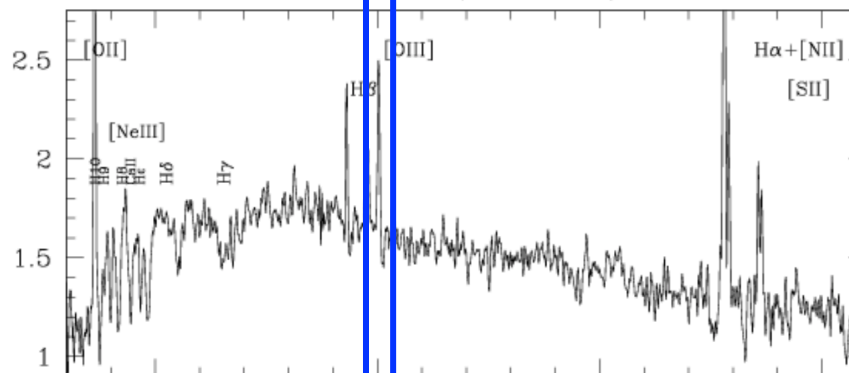
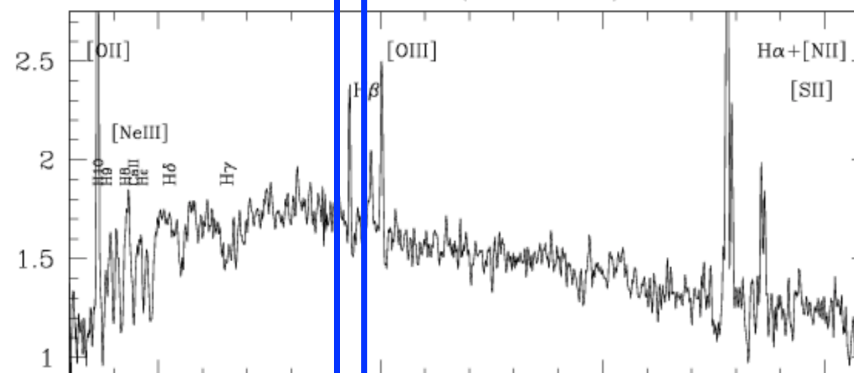
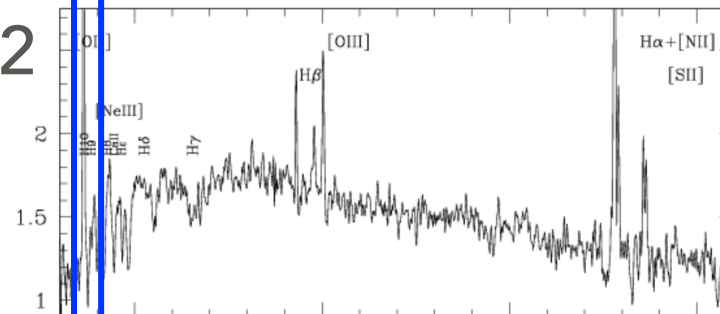
total

$\text{H}\beta @ z=3.1$

$\text{OIII} @ z=2.9$

$\text{H}\alpha @ z=2$

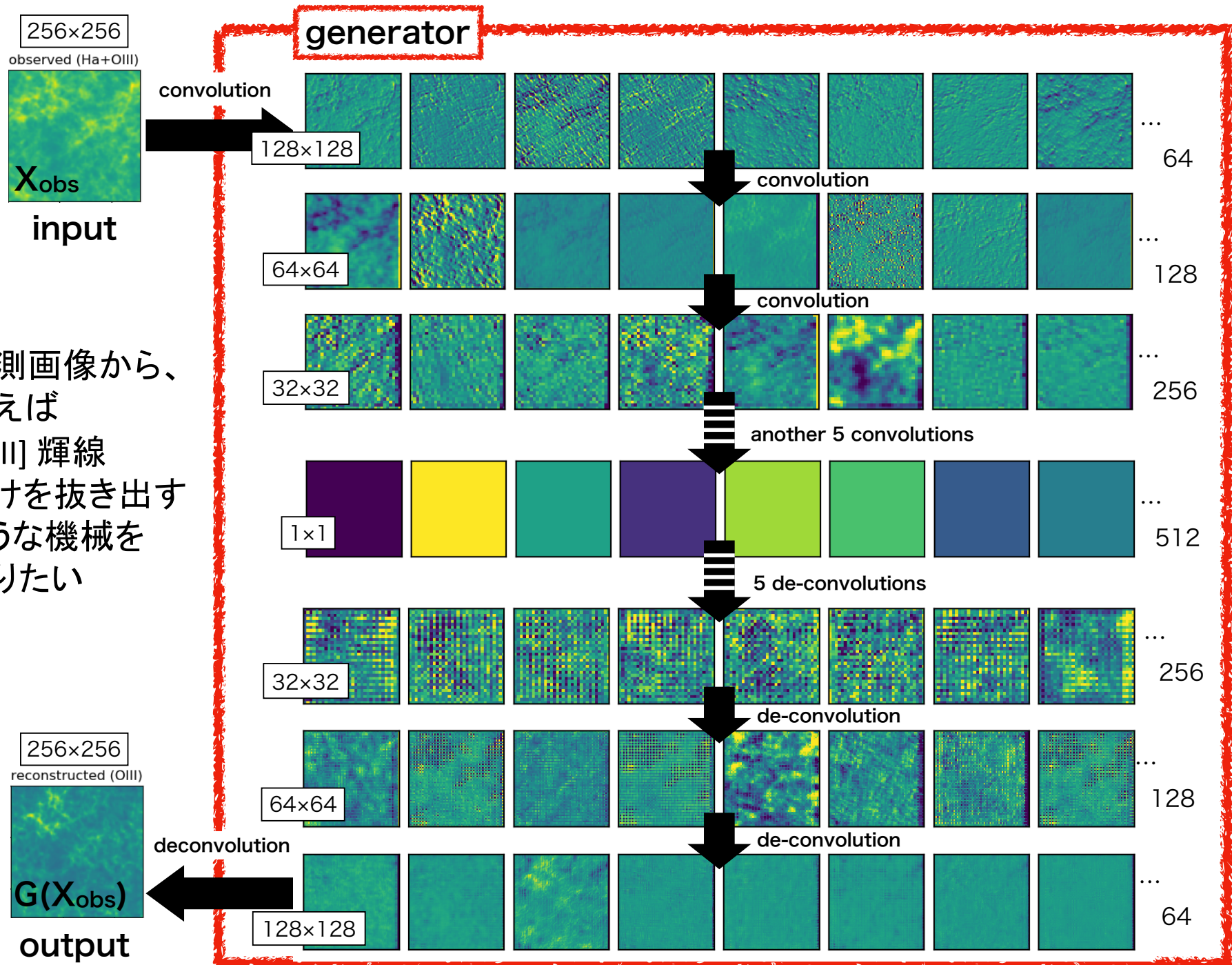
$\text{OII} @ z=4.2$



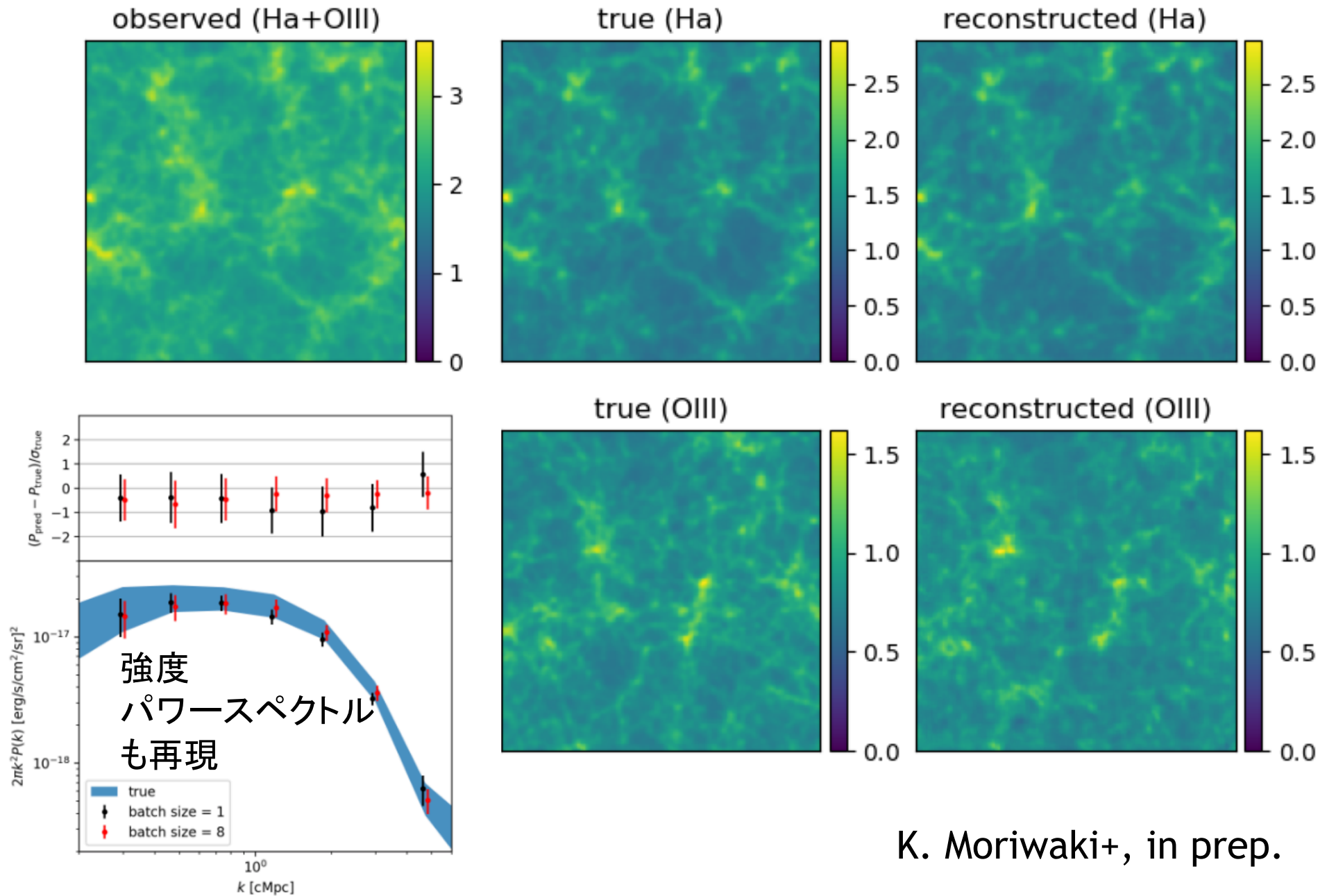
2 $\mu\text{m}$  band



観測画像から、  
例えば  
[OIII] 輝線  
だけを抜き出す  
ような機械を  
作りたい



# 情報分離の結果



おまけ



## サイエンスの自動化



1. AIはこの現象を見て支配方程式を導けるか？
2. AIはこの後の時間進化を正確に予測できるか？
3. AIはこの現象から基本法則を導くことができるか？

# AI COPERNICUS 'DISCOVERS' THAT EARTH ORBITS THE SUN

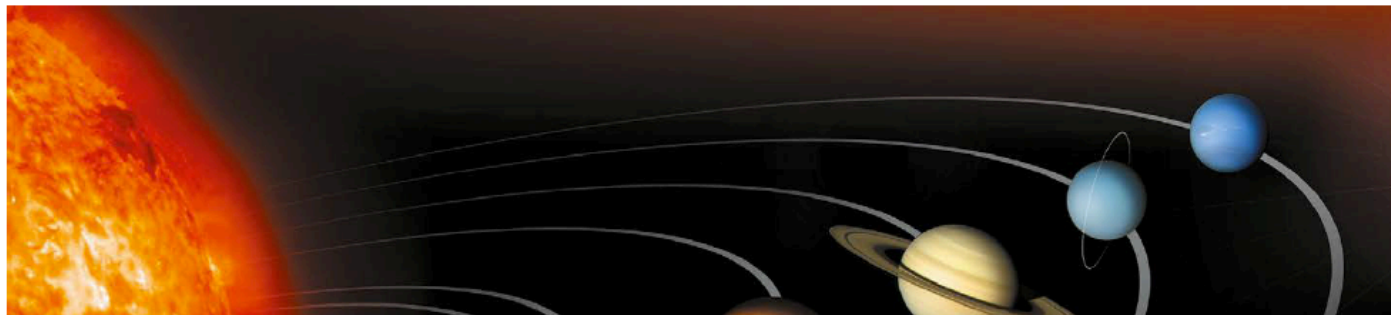
Neural network that teaches itself the laws of physics could help to solve quantum-mechanics mysteries.

By Davide Castelvecchi

**A**stronomers took centuries to figure it out. But now, a machine-learning algorithm inspired by the brain has worked out that it should place the Sun at the centre of the Solar System, on the basis of how movements of the Sun and Mars appear from Earth. The feat is one the first tests of a technique that researchers hope

they can use to discover new laws of physics, and perhaps to reformulate quantum mechanics, by finding patterns in large data sets. The results are due to appear in *Physical Review Letters* (R. Iten *et al. Phys. Rev. Lett.*; in the press).

Physicist Renato Renner at the Swiss Federal Institute of Technology (ETH) in Zurich and his collaborators wanted to design an algorithm that could distil large data sets down into a few basic formulae, mimicking how physicists come

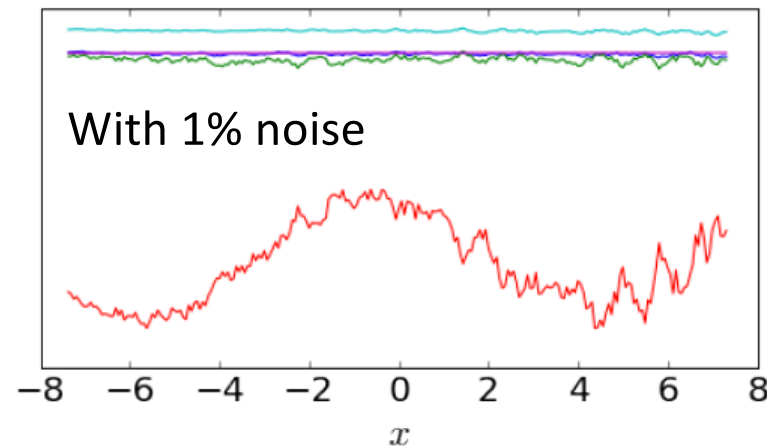
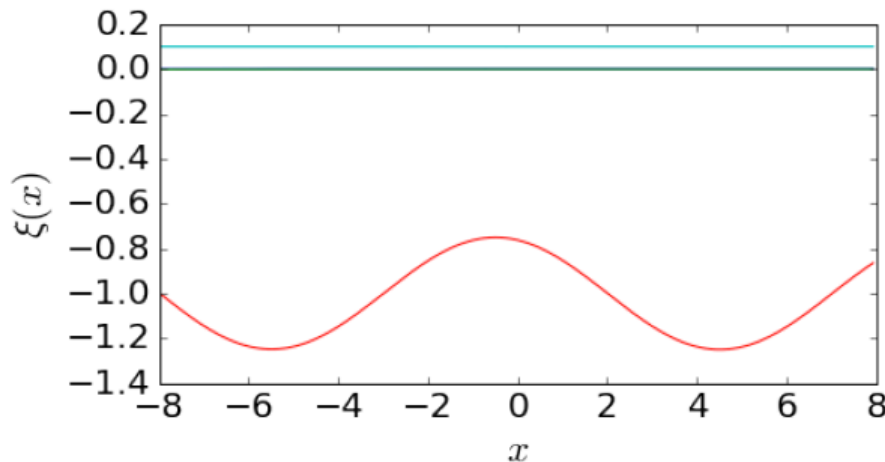
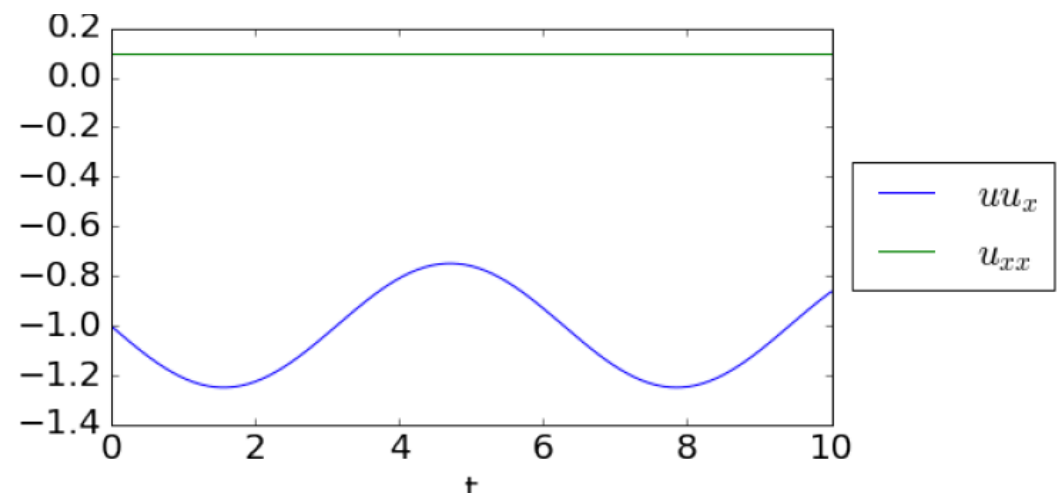
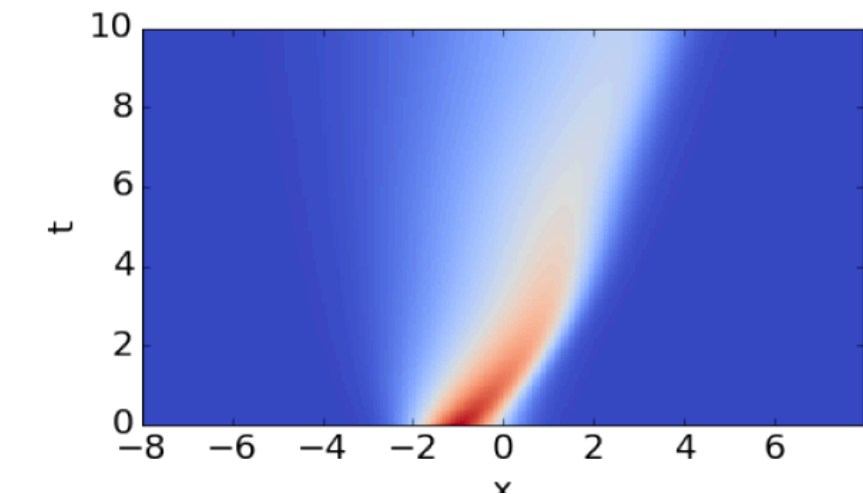


# 機械はバーガス方程式の解をみてバーガス方程式を導けるか ----- できる！

しかも、係数が時間変化する場合でも  
それを検知できる。

$$u_t = a(t)uu_x + 0.1u_{xx}$$

$$a(t) = -\left(1 + \frac{\sin(t)}{4}\right)$$





## バーガース方程式の解からバーガース方程式を導く

$$u_t = N(u, u_x, u_{xx}, \dots) = \sum_{j=1}^d N_j(u, u_x, u_{xx}, \dots) \xi_j$$

各時空点での値を使って行列形式で表される。

$$\underbrace{\begin{pmatrix} u_t(x_1, t_1) \\ u_t(x_2, t_1) \\ \vdots \\ u_t(x_n, t_m) \end{pmatrix}}_{\mathbf{U}_t} = \underbrace{\begin{pmatrix} 1 & u(x_1, t_1) & u_x(x_1, t_1) & \dots & u^3 u_{xxx}(x_1, t_1) \\ 1 & u(x_2, t_1) & u_x(x_2, t_1) & \dots & u^3 u_{xxx}(x_2, t_1) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & u(x_n, t_m) & u_x(x_n, t_m) & \dots & u^3 u_{xxx}(x_n, t_m) \end{pmatrix}}_{\Theta(\mathbf{U})} \xi$$

スパースプライヤー(“ $\xi$  には 0が多い”)を使うと関係のない項が判明する。  
各係数が時間依存する場合でも巨大な(無駄な)行列を処理すればよい。

$$\begin{pmatrix} u_t^{(1)} \\ u_t^{(2)} \\ \vdots \\ u_t^{(m)} \end{pmatrix} = \underbrace{\begin{pmatrix} \Theta(u^{(1)}) & & & \\ & \Theta(u^{(2)}) & & \\ & & \ddots & \\ & & & \Theta(u^{(m)}) \end{pmatrix}}_{\tilde{\Theta}} \begin{pmatrix} \xi^{(1)} \\ \xi^{(2)} \\ \vdots \\ \xi^{(m)} \end{pmatrix}$$

# Human vs AI

我々(人間)の理解:  $\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$  の解が  $u = f(x - t)$

であることを知り(学び)、なんとなく物理的描像(波)を得るようになる

現在のAIのやり方: 解群から次のように係数を決定し、

$$\frac{\Delta u}{\Delta t} = -1 \times \frac{\Delta u}{\Delta x} + 0 \times \frac{\Delta^2 u}{\Delta x^2} + 0 \times \frac{\Delta^3 u}{\Delta x^3} + 0.00000001 \times u \frac{\Delta u}{\Delta x} + \dots$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0 \quad \text{に絞り込む.}$$

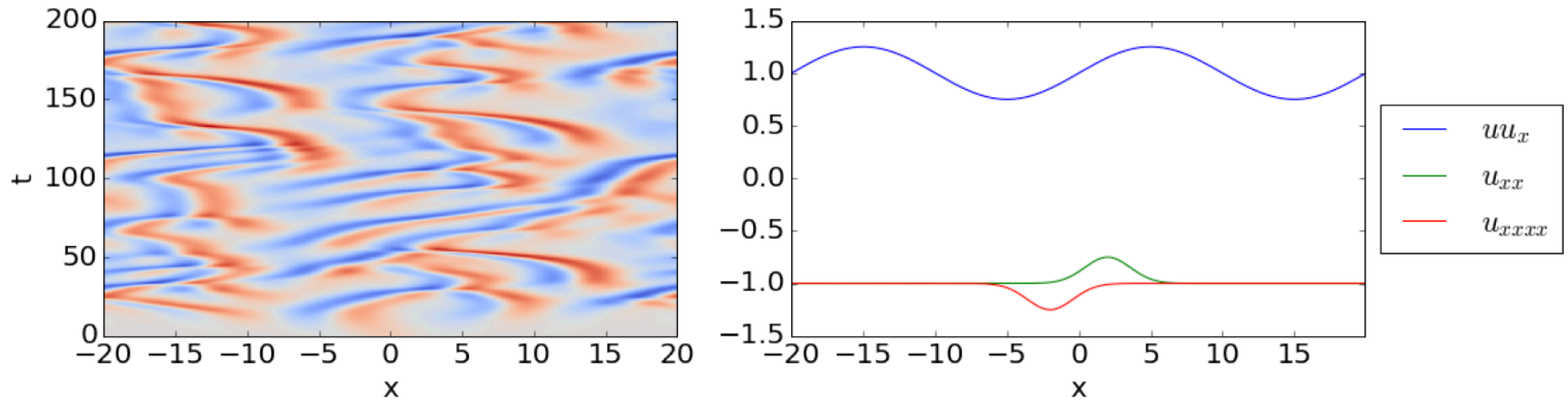
しかしこれは保存形に気づくことと同じだろうか:  $\frac{\partial}{\partial t} u + \frac{\partial}{\partial x} \left( \frac{u^2}{2} \right) = 0$

## 蔵本-Sivashinsky 方程式でも高階微分を含めて判定できる

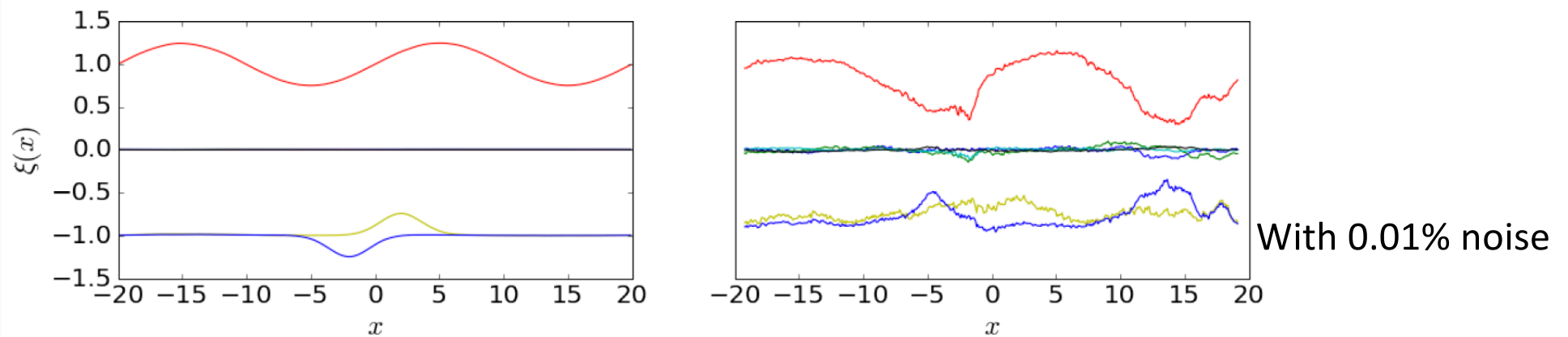
$$u_t = a(x)uu_x + b(x)u_{xx} + c(x)u_{xxxx}.$$

$$a(x) = 1 + \sin(2\pi x/L)/4$$

$$b(x) = -1 + e^{-(x-2)^2/5}/4 \text{ and } c(x) = -1 - e^{-(x+2)^2/5}/4$$



**Figure 9.** Left: dataset for identification of the spatially dependent Kuramoto Sivashinsky equation. Right: parametric dependency of the governing equations.





機械は偏微分方程式の最適な差分法を見出すことはできるか ----- できる！

微分は次の形で近似することができる

$$\frac{\partial^n u}{\partial x^n} \sim \sum_i \alpha_i u_i$$

ほとんどの数値スキームでは、差分化された式は次の形:

$$\frac{\partial u}{\partial t} = F(u_1, u_2, \dots, u_N) = c_1 u_1 + c_2 u_2 + \dots + c_N u_N$$

ここで係数  $c_i$  は時間的にも空間的にも定数である\_必要はない\_

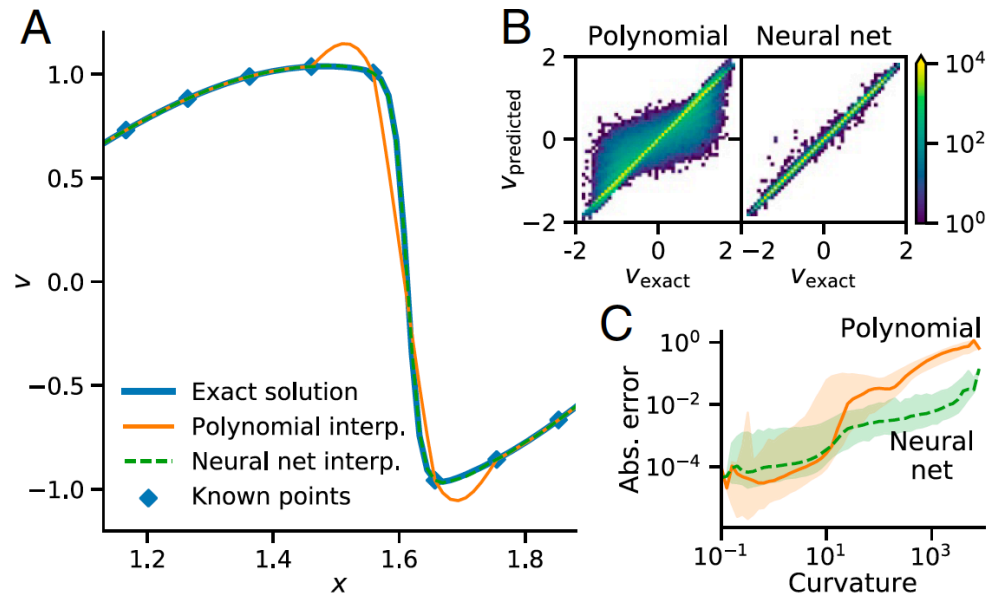
もっとも簡単な例として移流方程式を考えると、

$$\frac{\partial u}{\partial t} \Big|_{x_2} \sim \frac{\Delta u}{\Delta t} \Big|_{x_2} = \frac{1}{\Delta x} (u_3 - u_2)$$

さて、AI(機械)が最適な  $c_i$  を決めることはできるだろうか？

## 効率よくシミュレーションを行う機械

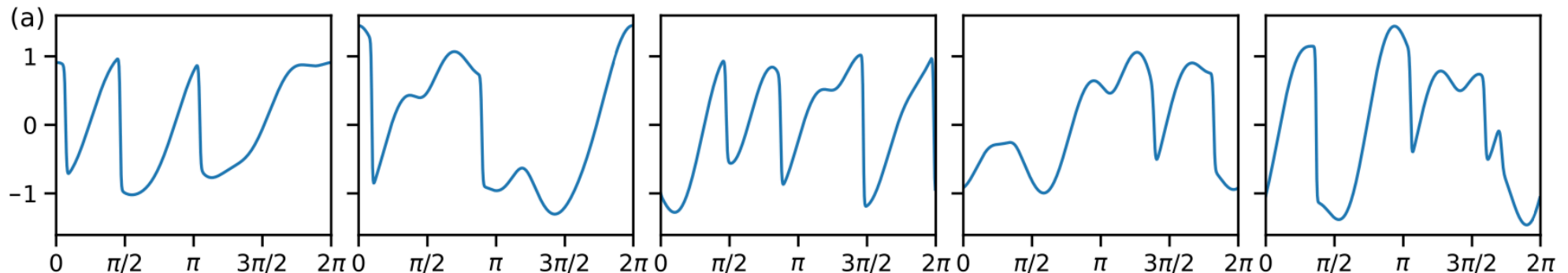
粘性項入りのバーガース方程式 
$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x} J \left( v, \frac{\partial v}{\partial x} \right) = f(x, t), \quad J \equiv \frac{v^2}{2} - \eta \frac{\partial v}{\partial x},$$



有限差分法は物理量  $v$  を局所的に多項式近似して補完を行う。  
(左の図のオレンジ線)  
しかしバーガース方程式の解は多項式では近似できない!

**Fig. 1.** Polynomial vs. neural net-based interpolation. (A) Interpolation between known points (blue diamonds) on a segment of a typical solution of Burgers' equation. Polynomial interpolation exhibits spurious "overshoots" in the vicinity of shock fronts. These errors compound when integrated in time, such that a naive finite-difference method at this resolution quickly diverges. In contrast the neural network interpolation is

ニューラルネットワークに  
下図のような40000例の解を学ばせる



## 効率よくシミュレーションを行う機械

### 差分のステンシルが時空点に依存する

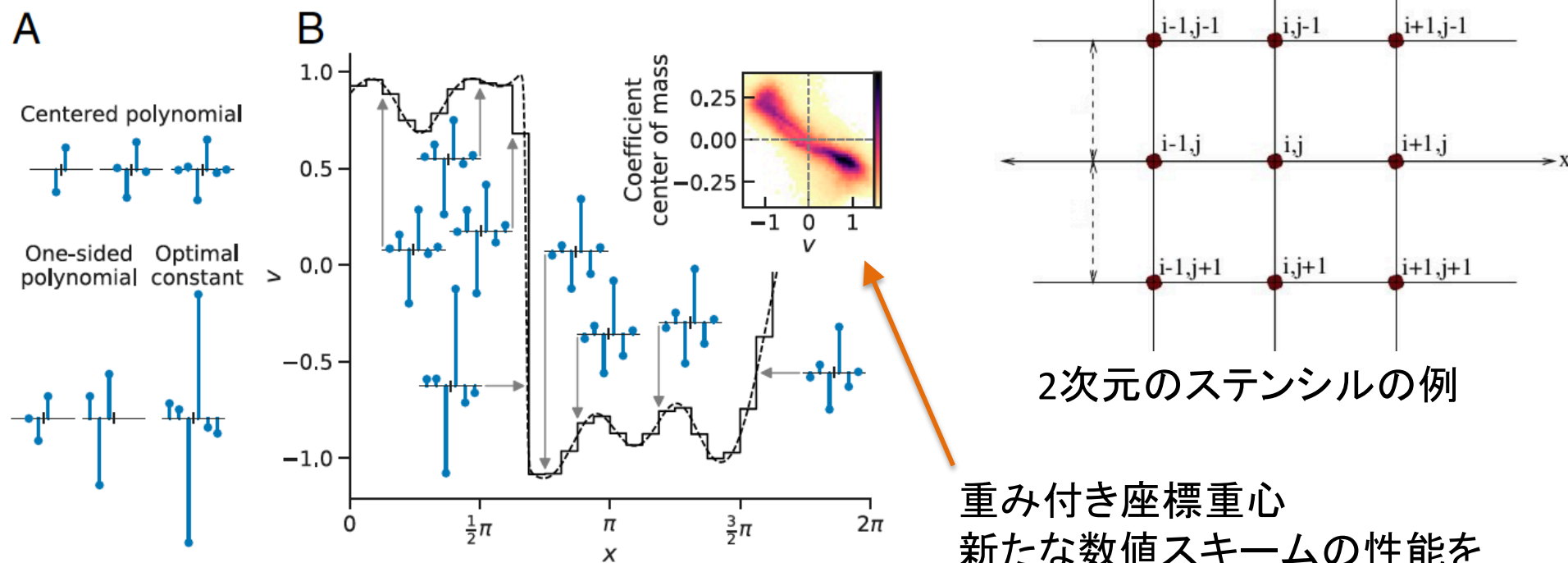


Fig. 2. Learned finite-volume coefficients for Burgers' equation. Shown are fixed and spatiotemporally varying finite-volume coefficients  $\alpha_1^{(1)}, \dots, \alpha_6^{(1)}$  (Eq. 2) for  $\partial v / \partial x$ . (A) Various centered and 1-sided polynomial finite-volume coefficients, along with optimized constant coefficients trained on this dataset ( $16\times$  resample factor in Fig. 3). The vertical scale, which is

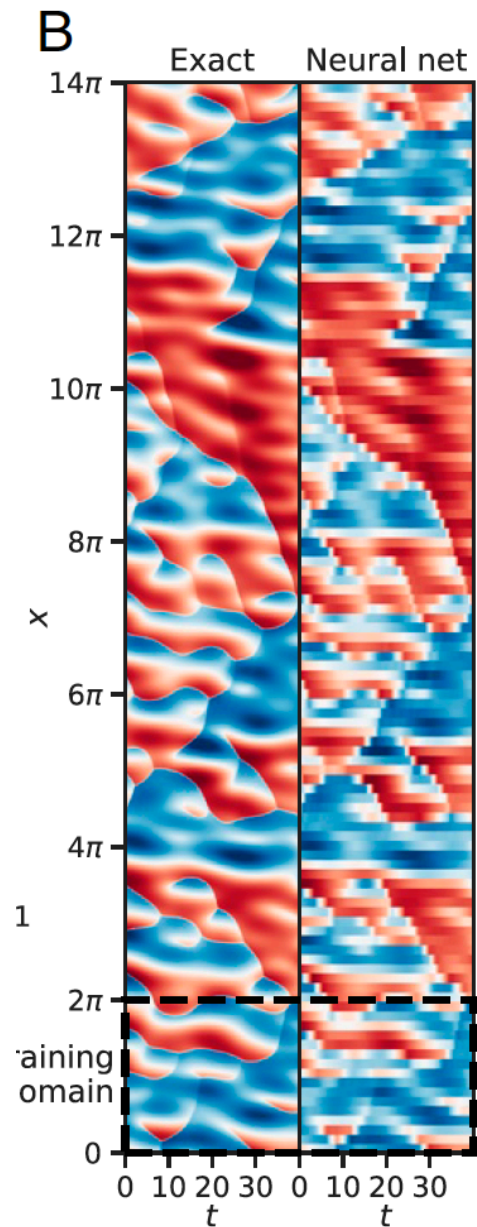
重み付き座標重心  
新たな数値スキームの性能を  
あらかじめ推測できる。  
(Neural network の解が最適と  
仮定する)

↑ 上記の1次元の例では  $N=6$

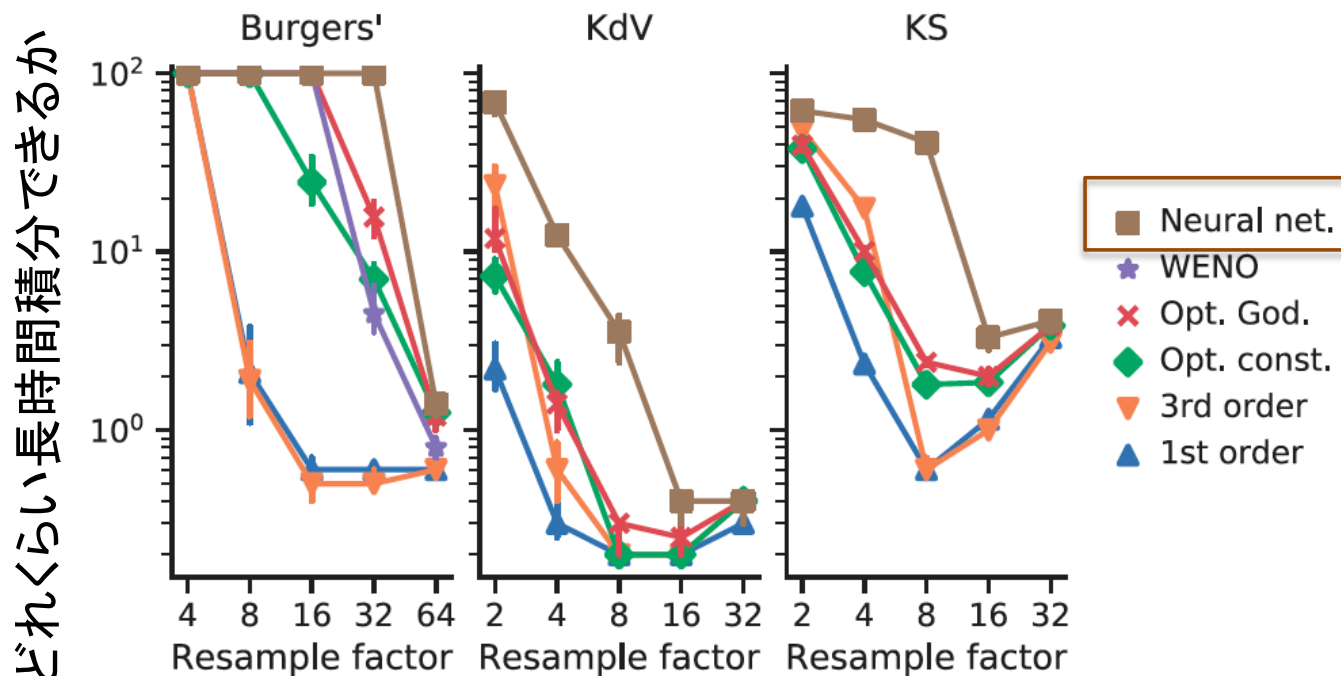
$(u_i, u_{i+1}, u_{i+2}, u_{i+3}, u_{i+4}, u_{i+5})$  が最適とあらかじめ調べている。

## 精度よくシミュレーションを行う機械

実効的に粗い格子上で  
高精度の解を得ることができる



## 他の非線形方程式



Resample factor: 解像度の悪さ

ニューラルネットによる解は、伝統的な手法(WENO等)  
に比べて解像度を下げても精度を保つことができる。



## AI COPERNICUS (ETHのAIプログラム [NN])

**目的:** 地球から見た火星と太陽の位置  $\vartheta_M(t)$ ,  $\vartheta_S(t)$  を予測

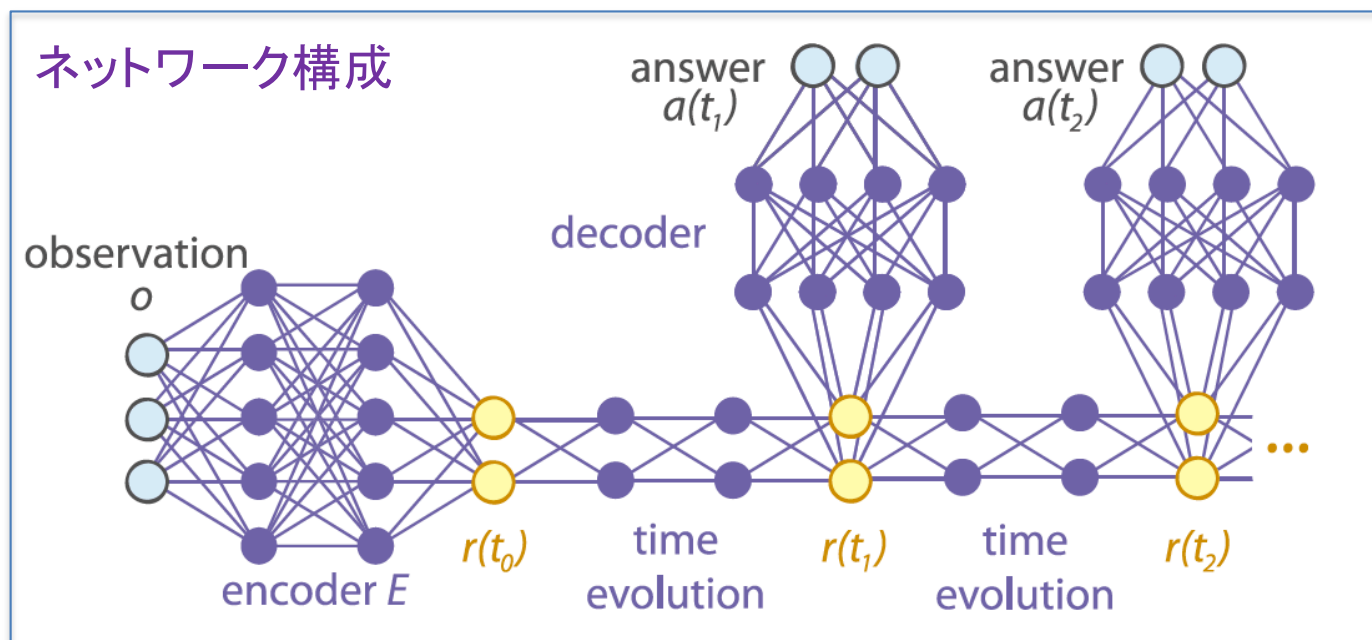
初期条件は  $\vartheta_M(0)$  and  $\vartheta_S(0)$

**正しい物理モデル:** 火星と地球が太陽の周りを巡っている

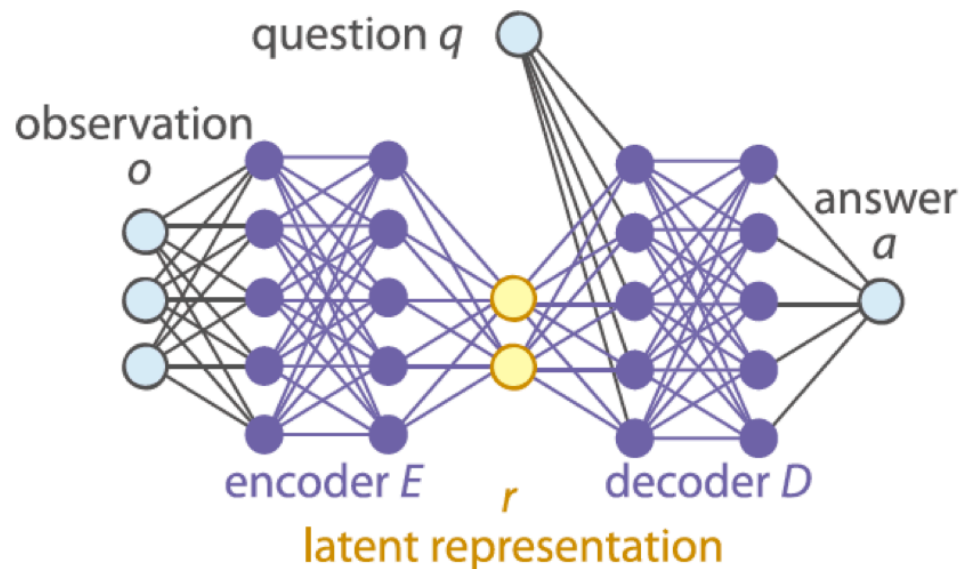
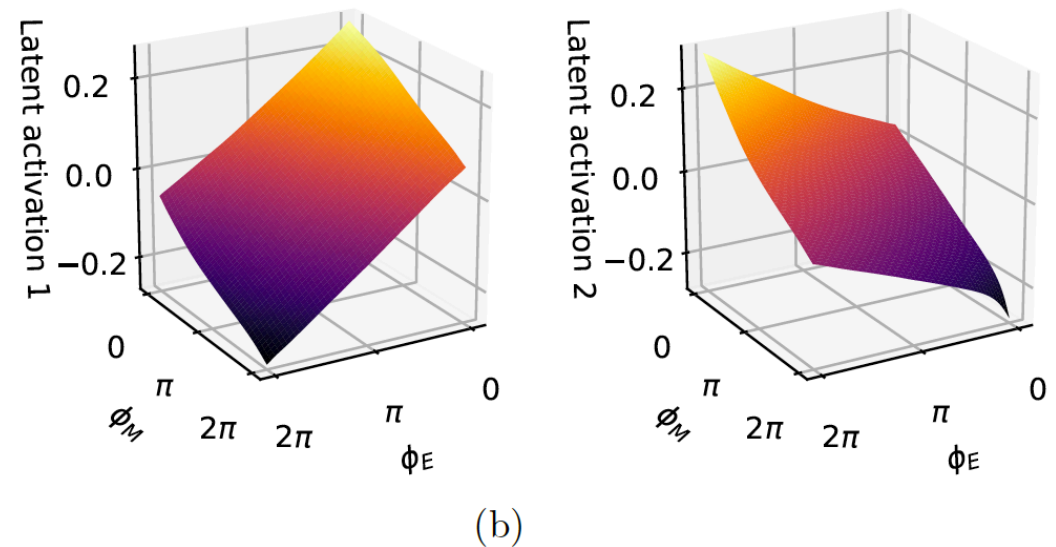
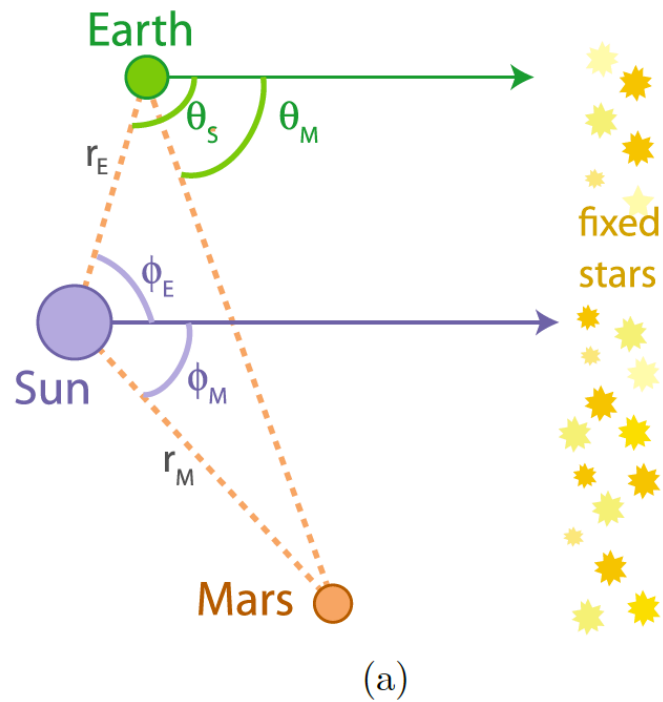
**トレーニング:** 時系列データ  $[\vartheta_M(t_1), \vartheta_S(t_1), \dots, \vartheta_M(t_n), \vartheta_S(t_n); n=20]$  をたくさん

**入力:**  $[\vartheta_M(t_0), \vartheta_S(t_0)]$  コペルニクスの生涯中の毎週の観測値からランダムに抽出

**結果:** ニューラルネットは  $[\vartheta_M(t_{\text{next}}), \vartheta_S(t_{\text{next}})]$  を誤差 0.4% で予測



# AI COPERNICUS



結果:  
中間層(潜在変数)を調べると、  
 $\phi_E$  と  $\phi_M$  の線形和を使っていることがわかった。

# まとめ

- 天文観測データ、特にサーベイ観測のデータは解析そのものが困難になっている (ペタバイト級)
- リアルタイム処理 (時間軸) は大変、無理
- 観測データを科学的発見に結びつける手法の構築とそのための人材育成が必要。  
似たようなことは、どの科学技術分野でも起こっている。
- この後2030年までこのトレンドは加速していく
- サイエンス自動化へ向けて (2030-):  
Forward modeling (理論モデル、物理法則) の強み