



13

スーパーカミオカンデによる

超新星ニュートリノ観測データの解析手法の開発

Harada et al., ApJ, **954**, 52 (2023)

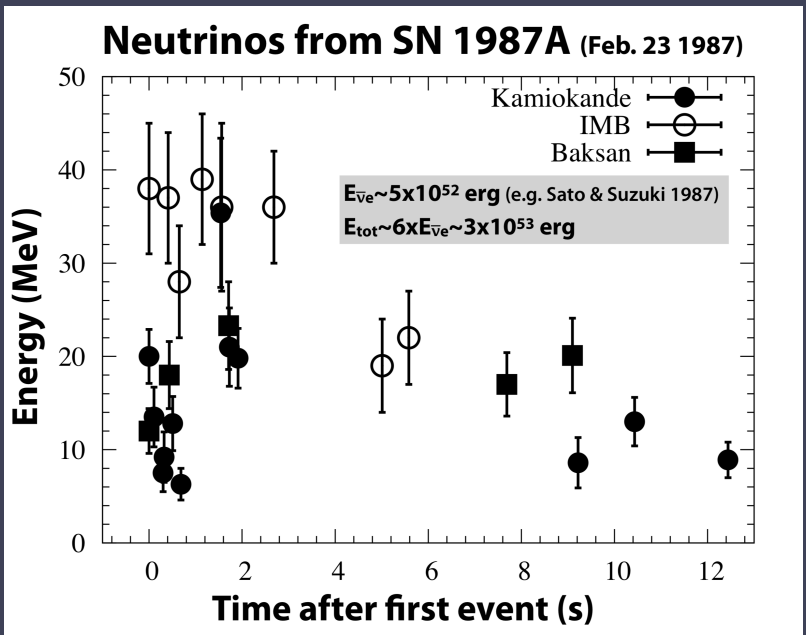
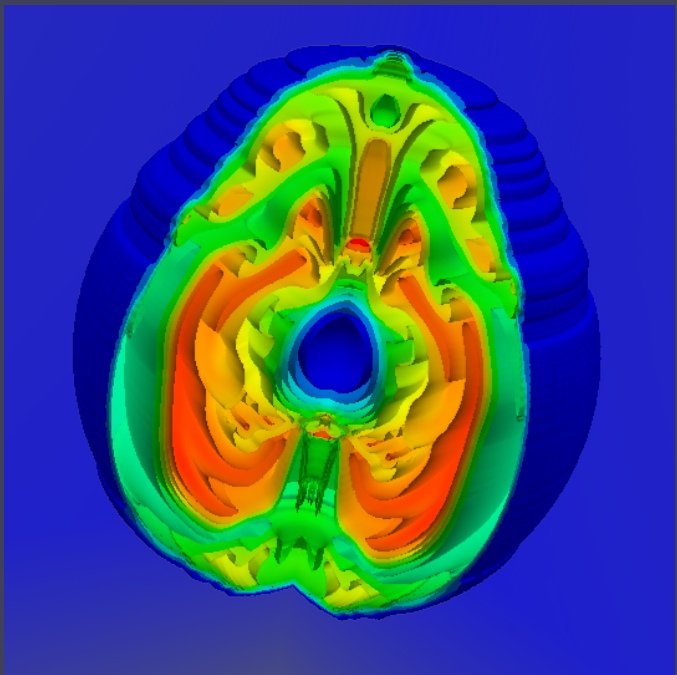
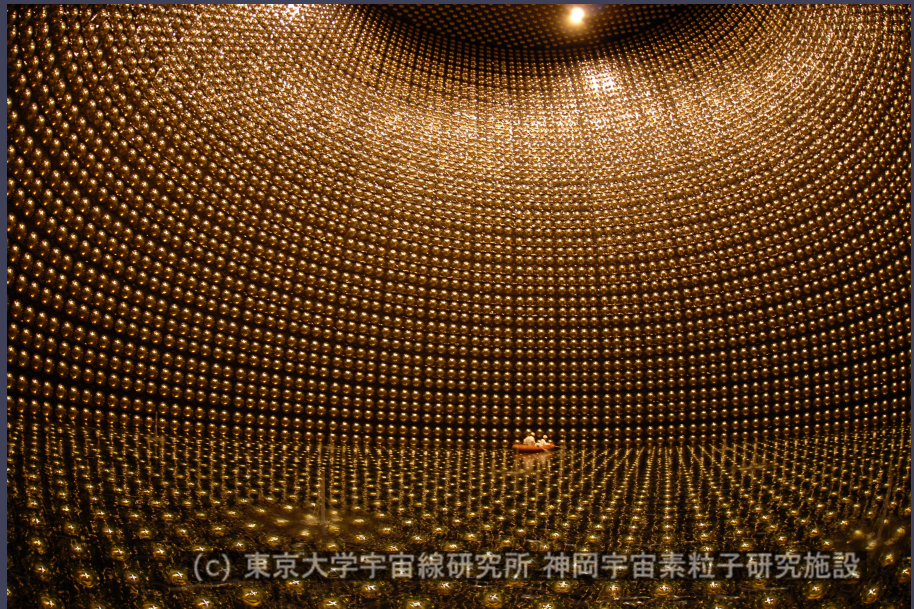
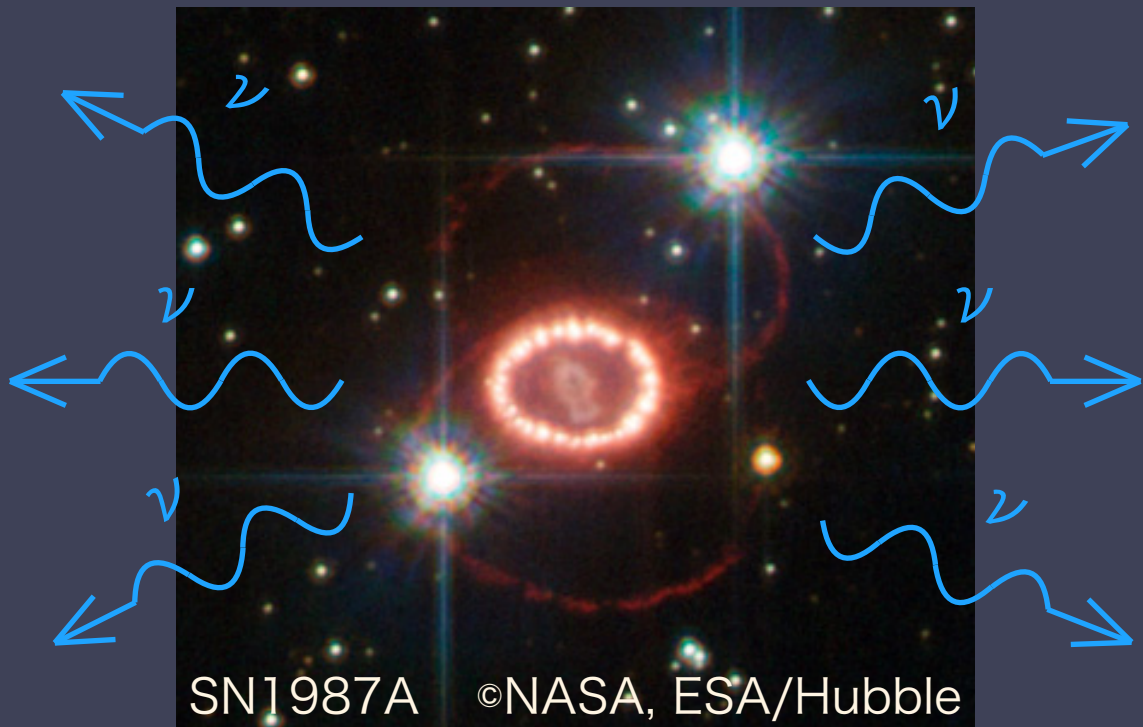
原田了 (理化学研究所)

共同研究者: nuLC collaboration (中西史美、原田将之、小汐祐介 (岡山大)、中里健一郎 (九州大)、ロジャー・ウェンデル (京都大)、森正光(国立天文台)、諏訪雄大 (東京大)、住吉光介 (沼津高専)、赤穂龍一郎 (早稲田大))

査定額：オンライン打ち合わせのためのslack利用料として20万円

超新星ニュートリノ観測と理論

超新星からのニュートリノ放射



超新星を
シミュレート

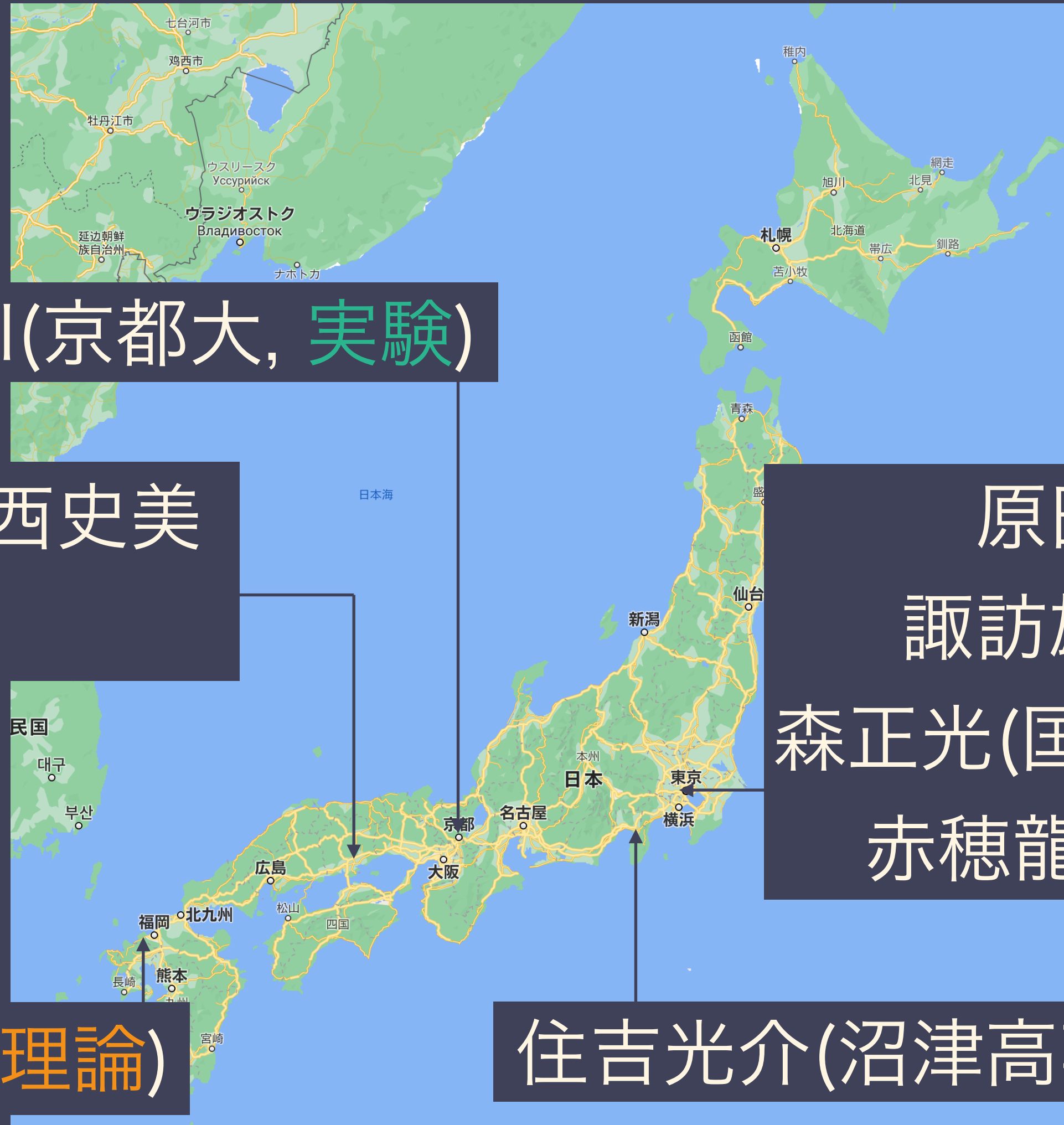
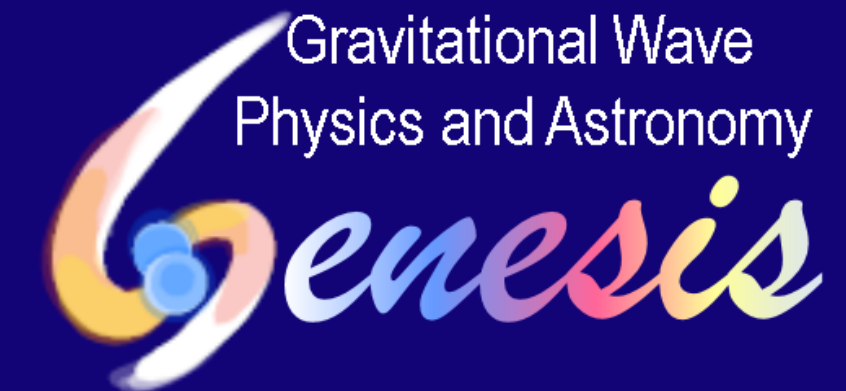
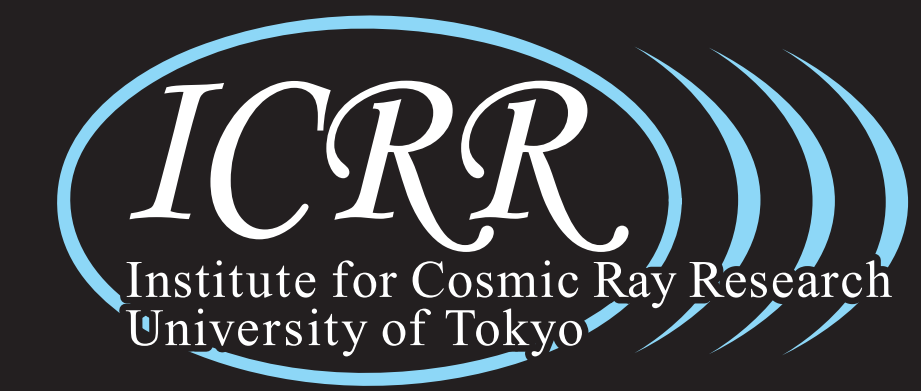
ニュートリノを観測

理論モデルと観測データの対応づけ

理論家と実験家で協力して超新星ニュートリノ検出に備える

nuLC コラボレーション

neutrino Light Curve



Roger Wendell(京都大, 実験)

小汐由介、原田将之、中西史美
(岡山大, 実験)

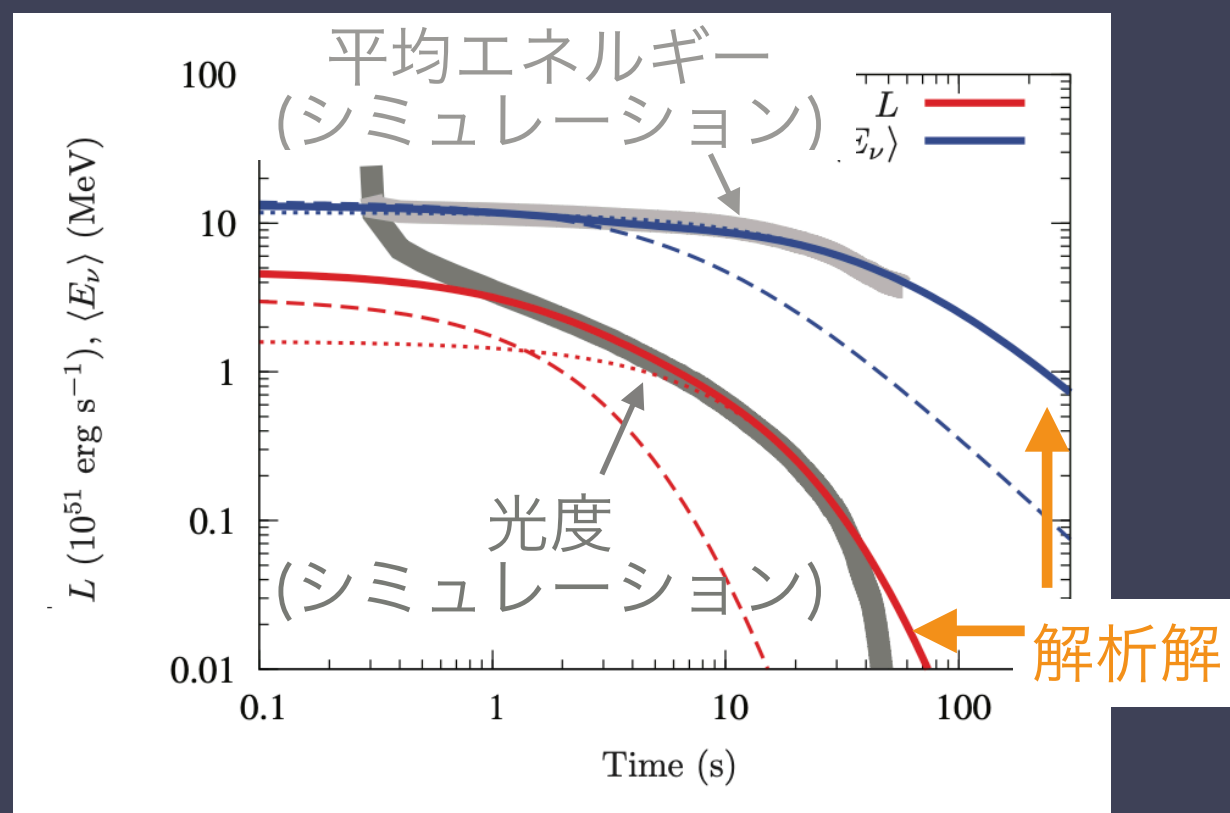
原田了(理研, 理論)、
諏訪雄大(東京大, 理論)、
森正光(国立天文台, 理論/実験)、
赤穂龍一郎(早稲田大, 理論)

中里健一郎(九州大, 理論)

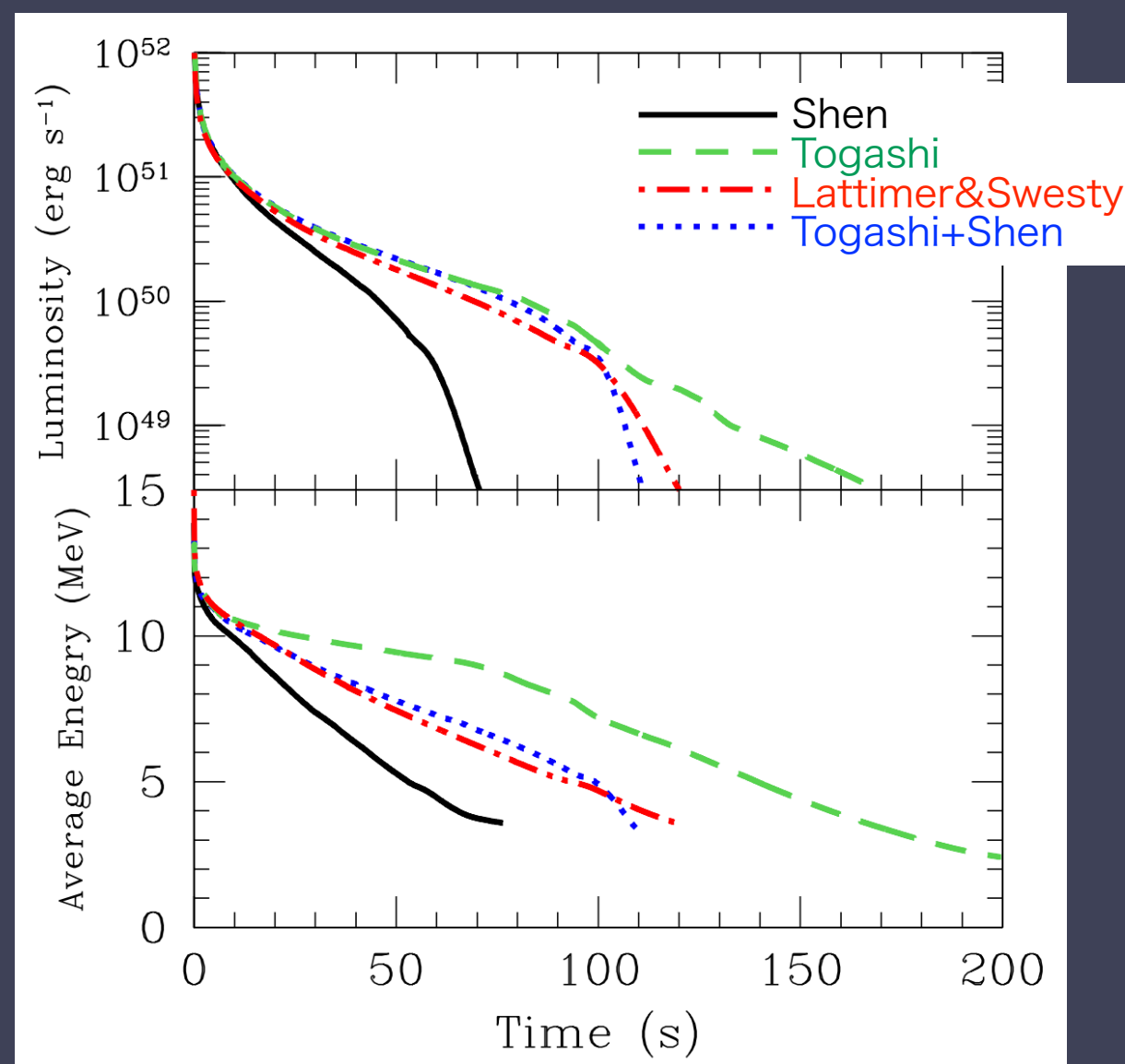
住吉光介(沼津高専, 理論)

nuLC コラボレーションの成果

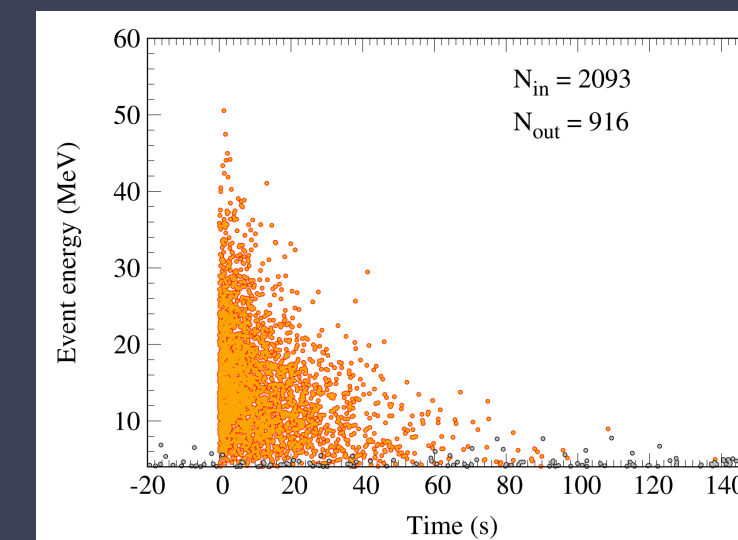
neutrino Light Curve



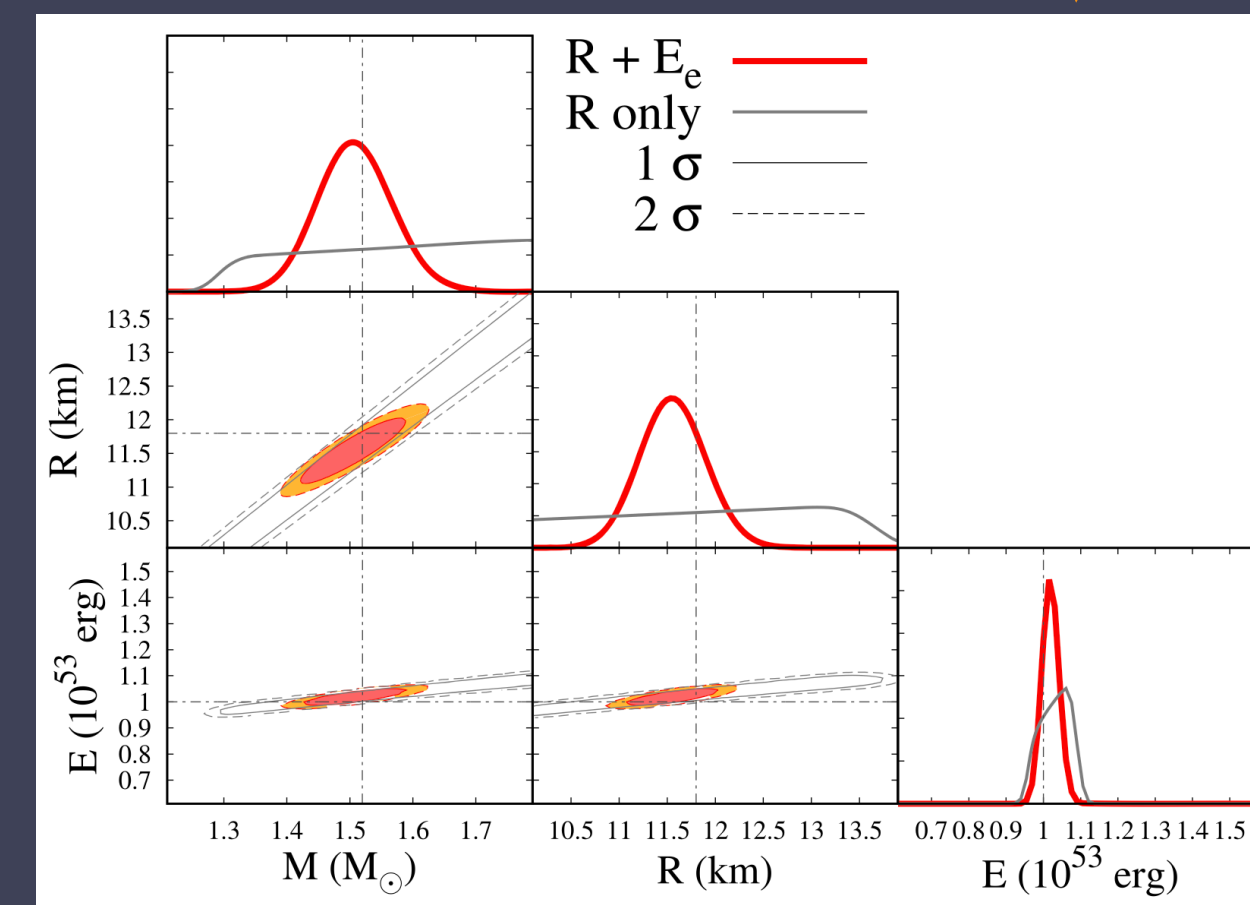
Suwa et al. (2021)



Nakazato et al. (2022)



模擬観測データ
(イベントエネルギー vs 時間)



Suwa et al. (2022)

2020年度(課題一年目)

- ・ニュートリノ光度曲線の解析解の開発→
詳細な数値シミュレーションに頼らず、簡単に光度曲線をモデル化可能

2021年度(課題二年目)

- ・様々な核物質状態方程式・親星について、超新星ニュートリノ信号の理論モデルを開発
- ・ニュートリノエネルギーの誤差推定法を提案

2022年度(課題三年目)

- ・解析解に基づく超新星ニュートリノ観測データ解析パイプラインを開発
- ・観測後の即座解析でパラメータに当たりをつける

nuLC コラボレーション2023

neutrino Light Curve

2023年度(課題四年目)

- ・超新星ニュートリノ観測データ解析パイプライン **Supernova Parameter Estimation Code based on Insight on Analytic Late-time Burst Light curve at Earth Neutrino Detector (SPECIAL BLEND; AH+, ApJ, 954, 52 (2023))**を公開コードとして開発
- ・フィッティング手法を改善したことで、比較的遠方の信号まで解析可能

```
SPECIAL_BLEND_pyinterface.ipynb
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ すべての変更を保存しました
+ コード + テキスト
[1] !git clone https://[user_name]:[access_token]@github.com/akira-harada/SPECIAL_BLEND.git
Cloning into 'SPECIAL_BLEND'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 54 (delta 24), reused 35 (delta 14), pack-reused 0
Unpacking objects: 100% (54/54), done.

import os, sys

!{sys.executable} -m numpy.f2py --quiet -c /content/SPECIAL_BLEND/SPECIAL_BLEND.F90 -m SPECIAL_BLEND

%run /content/SPECIAL_BLEND/event_generator.py

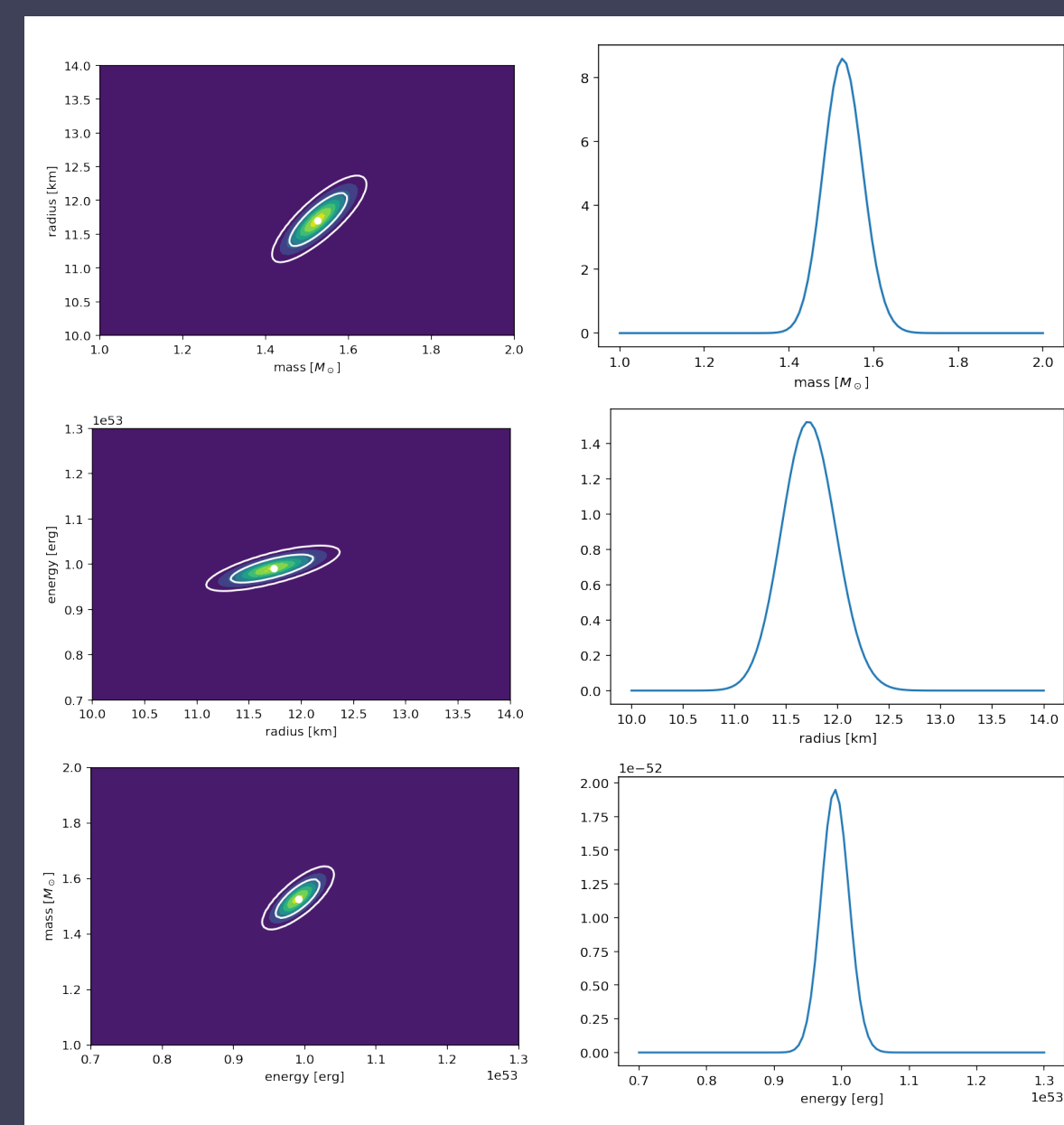
# define functions
%config InlineBackend.figure_format = 'retina'
import numpy as np
import csv
import matplotlib.pyplot as plt
import SPECIAL_BLEND as SB

def main():
    params = np.loadtxt('/content/SPECIAL_BLEND/parameters.dat') # 'parameters.dat' file has the following contents: assumed gbeta, distance to the SN [kpc], detector mass [kton], parameter
    origdata = np.loadtxt('/content/time_energy.dat') # 'time_energy.dat' file has the time and energy of each event: first column is time, second column is energy
    analysis_mode = int(params[10]) # 1:unbinned, 2:full-binned (mode 3 and 4 work only in fortran version, not implemented in this Google Colaborator version)
    tmin = params[13]
    tmax = params[14]
    data = loaddata(tmin, tmax, origdata)
    if analysis_mode == 1:
        print("unbinned analysis")
        mlogLH, mass, rad, et = unbinned_likelihood(data, params)
        print("likelihood calculation completed")
    elif analysis_mode == 2:
        print("binned analysis")
        mlogLH, mass, rad, et = binned_likelihood(data, params)
        print("likelihood calculation completed")
```

```
SPECIAL_BLEND_pyinterface.ipynb
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ すべての変更を保存しました
+ コード + テキスト
[1] 1D marginalized result
mass = 1.526519e+00 +4.92e-02/-4.78e-02 (68%) +9.14e-02/-8.68e-02 (95%)
radius = 1.171508e+01 +2.65e-01/-2.60e-01 (68%) +5.25e-01/-5.04e-01 (95%)
energy = 9.901252e+52 +1.90e+51/-1.88e+51 (68%) +4.30e+51/-4.16e+51 (95%)

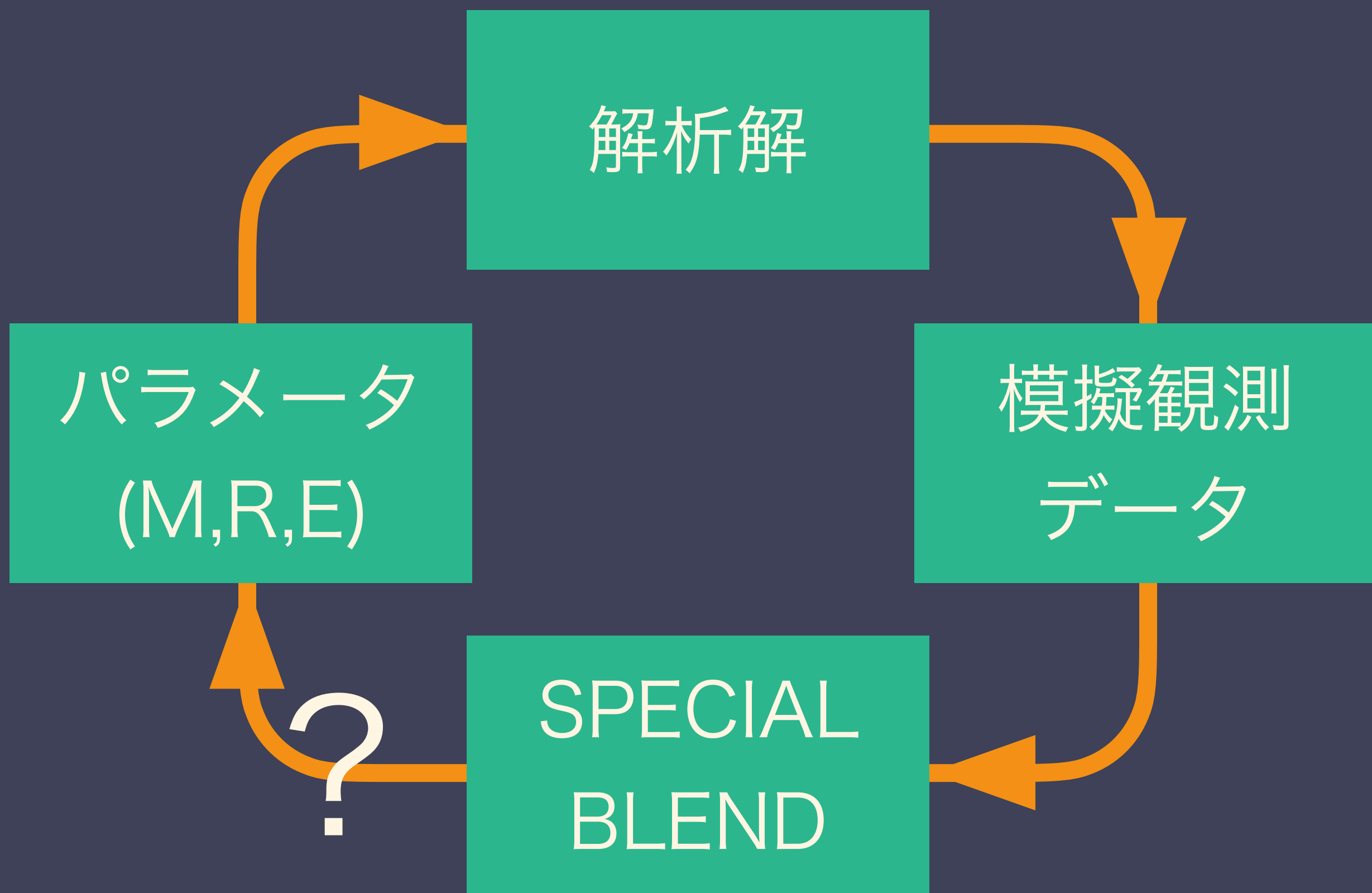
2D marginalized result
M-R: the best fit is (M,R)=(1.53e+00,1.17e+01), and the levels of CI is 7.36e+00 (68%) and 1.12e+00 (95%)
R-E: the best fit is (R,E)=(1.17e+01,9.91e+52), and the levels of CI is 1.39e-52 (68%) and 2.21e-53 (95%)
E-M: the best fit is (E,M)=(9.91e+52,1.53e+00), and the levels of CI is 8.32e-52 (68%) and 1.29e-52 (95%)

[11] # visualize likelihood
visualize(mass, rad, et, mlogLH, LH_MR, MRvals, LH_RE, REvals, LH_EM, EMvals, LHM, LHR, LHE)
```



github https://github.com/akira-harada/SPECIAL_BLEND にて公開中、Google colaboratoryにて利用可能

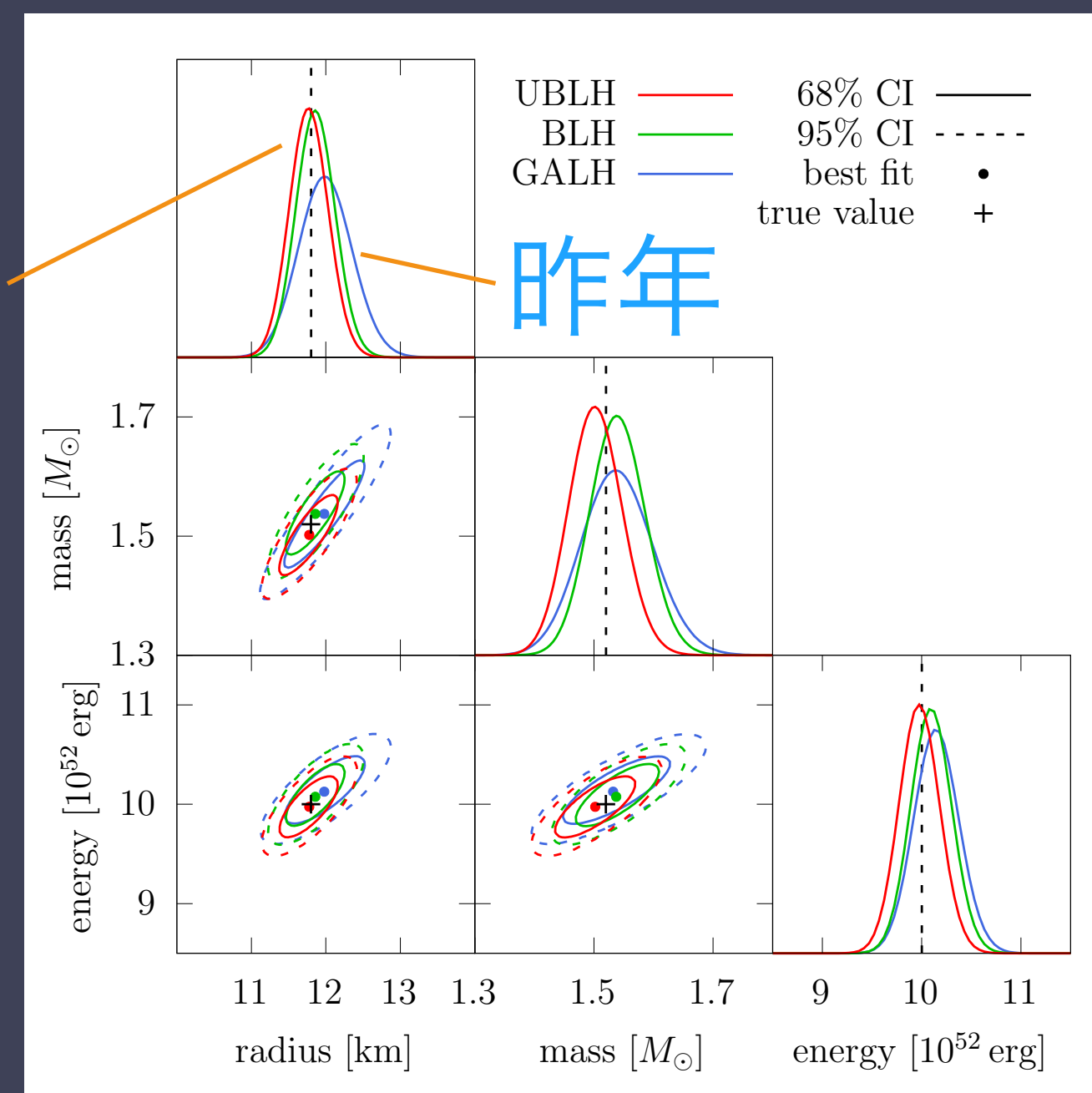
パラメータ推定テスト



- ・ 昨年はデータがガウス分布に従うことを仮定したが、SPECIAL BLENDはポアソン分布に基づくように改良→少数統計でも解析可
- ・ パラメータを決め、解析解をつくり、模擬観測データをつくり、それを解析してパラメータを再現できるか？

本年

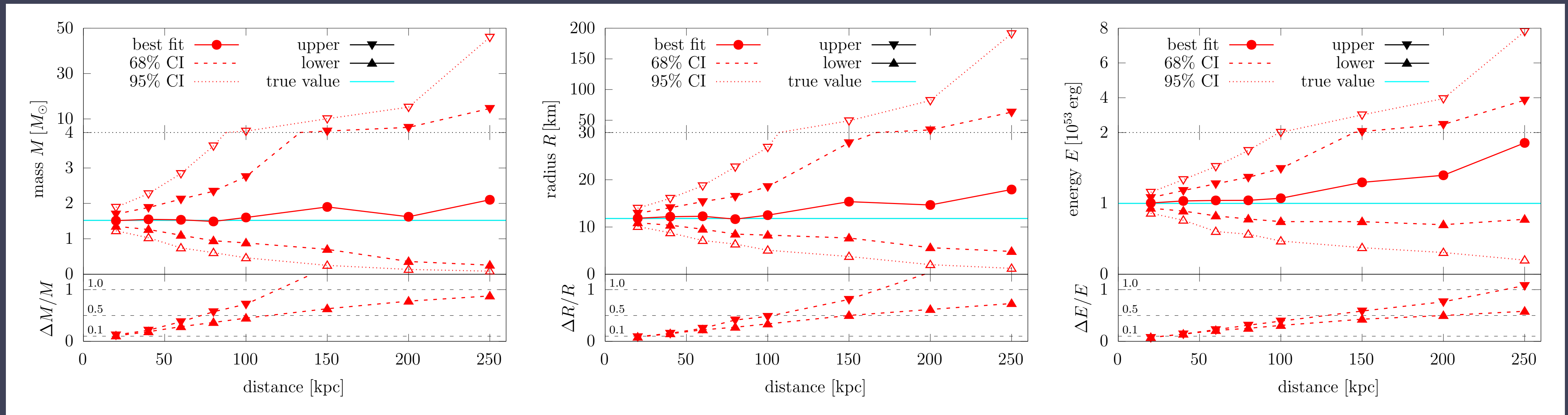
昨年



昨年のものより
不確かさ小

SPECIAL BLEND性能の距離依存性

- ・解析解モデルに基づくニュートリノ放射を様々な距離において、SKで受けた場合の模擬観測データを作り、SPECIAL BLENDでもともとのモデルパラメータを再現できるかを確認する
- ・ベストフィットは100 kpcくらいまでいい線行く
- ・10 kpcくらいまでエラー10%以内に抑えられる
- ・大マゼラン雲(50 kpc)なら20-30%程度



まとめ

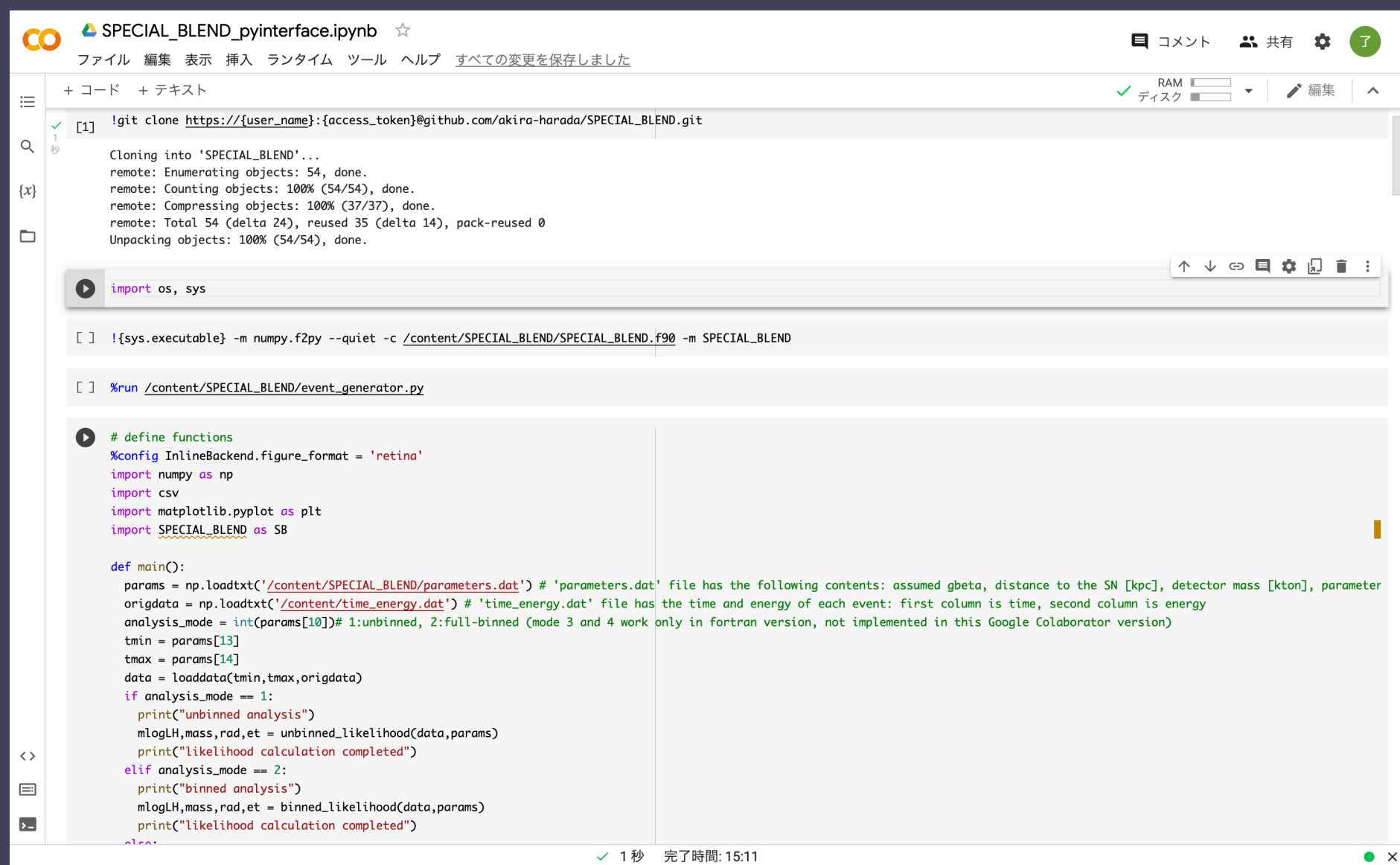
2020年度(課題一年目) Suwa et al., PTEP, 013E01 (2021): ν 光度曲線解析解の開発

2021年度(課題二年目) Nakazato et al., ApJ, **925**, 98 (2022): 超新星 ν 信号データベースの開発

2022年度(課題三年目) Suwa et al., ApJ, **934**, 15 (2022): 超新星 ν 信号解析手法の開発(in-house簡易版)

2023年度(課題四年目) Harada et al., ApJ, **954**, 52 (2023):

- ・超新星ニュートリノ観測データ解析パイプライン **Supernova Parameter Estimation Code based on Insight on Analytic Late-time Burst Light curve at Earth Neutrino Detector (SPECIAL BLEND)**を公開コードとして開発
- ・フィッティング手法を改善したことで、比較的遠方の信号まで解析可能



```
git clone https://[user_name]:[access_token]@github.com/akira-harada/SPECIAL_BLEND.git
Cloning into 'SPECIAL_BLEND'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 54 (delta 24), reused 35 (delta 14), pack-reused 0
Unpacking objects: 100% (54/54), done.

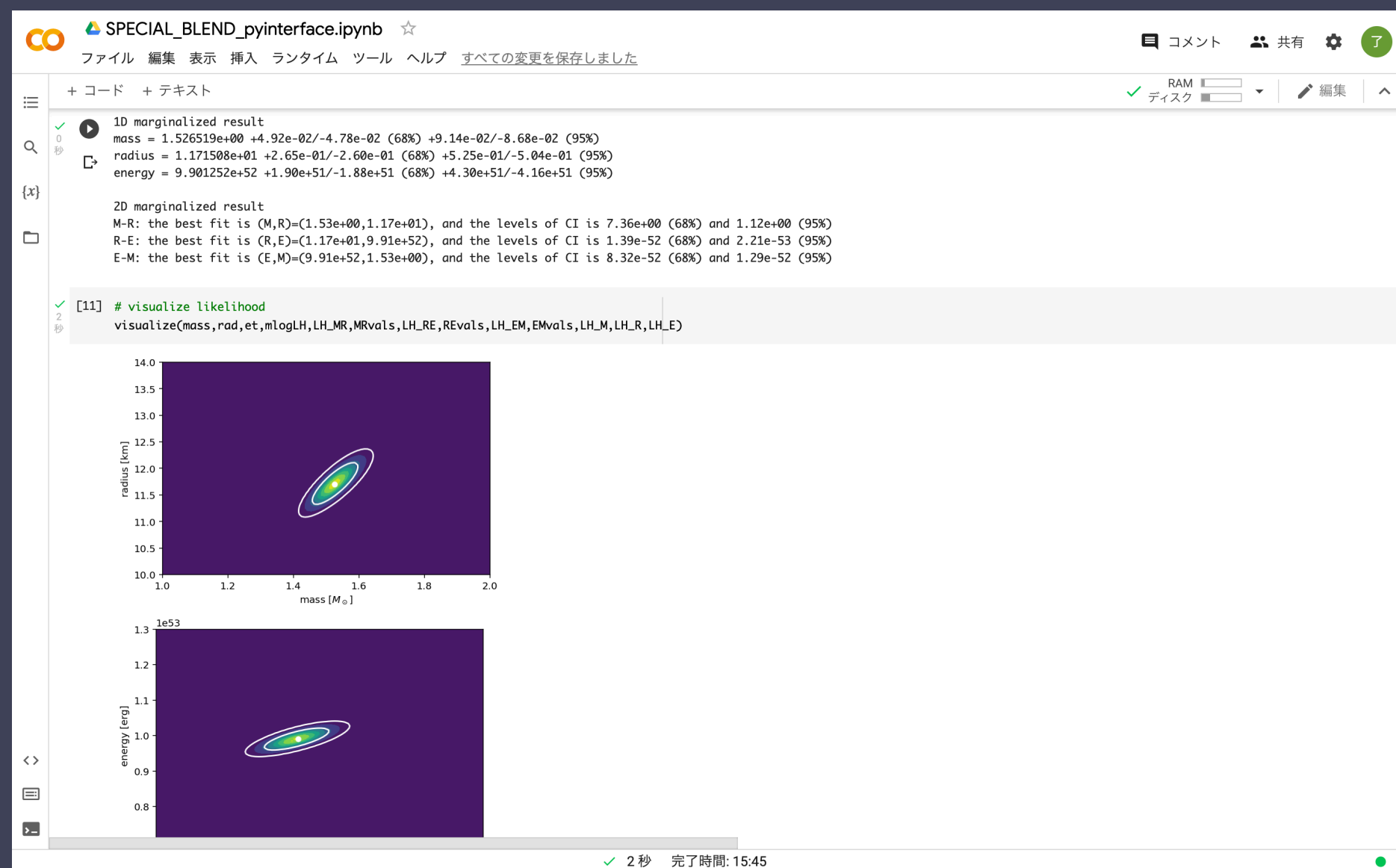
import os, sys

!{sys.executable} -m numpy.f2py --quiet -c /content/SPECIAL_BLEND/SPECIAL_BLEND.F90 -m SPECIAL_BLEND

%run /content/SPECIAL_BLEND/event_generator.py

# define functions
%config InlineBackend.figure_format = 'retina'
import numpy as np
import csv
import matplotlib.pyplot as plt
import SPECIAL_BLEND as SB

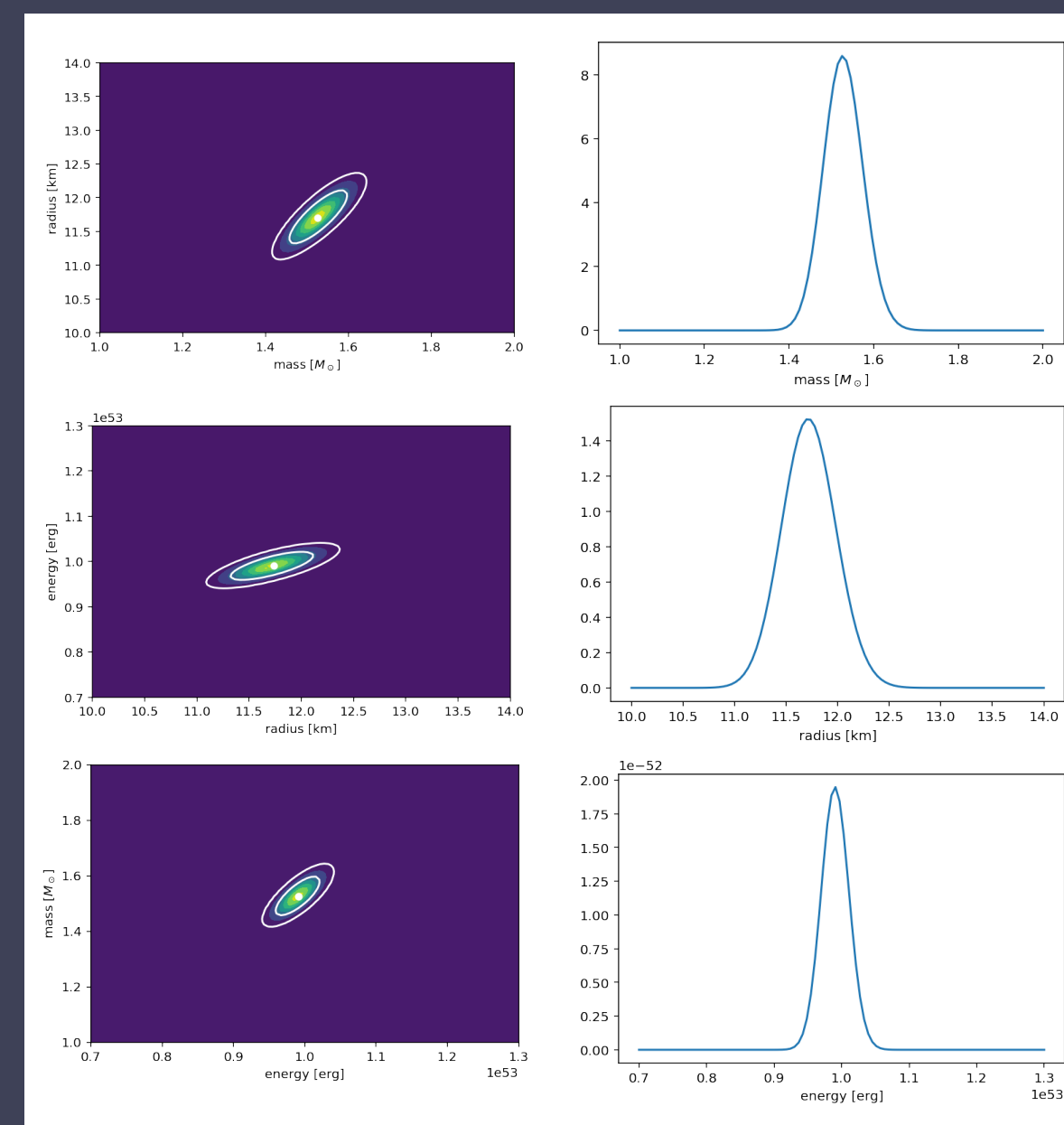
def main():
    params = np.loadtxt('/content/SPECIAL_BLEND/parameters.dat') # 'parameters.dat' file has the following contents: assumed gbeta, distance to the SN [kpc], detector mass [kton], parameter
    origdata = np.loadtxt('/content/time_energy.dat') # 'time_energy.dat' file has the time and energy of each event: first column is time, second column is energy
    analysis_mode = int(params[10]) # 1:unbinned, 2:full-binned (mode 3 and 4 work only in fortran version, not implemented in this Google Colaboratory version)
    tmin = params[13]
    tmax = params[14]
    data = loaddata(tmin, tmax, origdata)
    if analysis_mode == 1:
        print("unbinned analysis")
        mlogLH, mass, rad, et = unbinned_likelihood(data, params)
        print("likelihood calculation completed")
    elif analysis_mode == 2:
        print("binned analysis")
        mlogLH, mass, rad, et = binned_likelihood(data, params)
        print("likelihood calculation completed")
```



```
1D marginalized result
mass = 1.526519e+00 +4.92e-02/-4.78e-02 (68%) +9.14e-02/-8.68e-02 (95%)
radius = 1.171508e+01 +2.65e-01/-2.60e-01 (68%) +5.25e-01/-5.04e-01 (95%)
energy = 9.901252e+52 +1.90e+51/-1.88e+51 (68%) +4.30e+51/-4.16e+51 (95%)

2D marginalized result
M-R: the best fit is (M,R)=(1.53e+00,1.17e+01), and the levels of CI is 7.36e+00 (68%) and 1.12e+00 (95%)
R-E: the best fit is (R,E)=(1.17e+01,9.91e+52), and the levels of CI is 1.39e-52 (68%) and 2.21e-53 (95%)
E-M: the best fit is (E,M)=(9.91e+52,1.53e+00), and the levels of CI is 8.32e-52 (68%) and 1.29e-52 (95%)

# visualize likelihood
visualize(mass, rad, et, mlogLH, LH_MR, MRvals, LH_RE, REvals, LH_EM, EMvals, LHM, LHR, LHE)
```



github https://github.com/akira-harada/SPECIAL_BLEND にて公開中、Google colaboratoryにて利用可能