

機械学習による 輻射輸送クロージャー関係式

原田了 (宇宙線研究所)

共同研究者: 西河笙太, 山田章一(早稲田)

アウトライン

- ・ 超新星のレビュー (物理的背景)
- ・ 機械学習のレビュー (技術的背景)
- ・ クロージャー関係式の学習

超新星爆発

- ・ 超新星爆発: 星が死ぬ時の爆発
- ・ 分類:

重力崩壊型

- ・ 大質量星コアの重力崩壊に伴う爆発
- ・ 中性子星が中心に残される
- ・ 重力波・ニュートリノ観測のターゲット

熱核暴走型

- ・ 暴走的な核燃焼による爆発
- ・ 中心天体を残さない
- ・ 標準光源になる

超新星爆発

- ・ 超新星爆発: 星が死ぬ時の爆発
- ・ 分類:

重力崩壊型

- ・ 大質量星コアの重力崩壊に伴う爆発
- ・ 中性子星が中心に残される
- ・ 重力波・ニュートリノ観測のターゲット

熱核暴走型

- ・ 暴走的な核燃焼による爆発
- ・ 中心天体を残さない
- ・ 標準光源になる

重力崩壊型超新星爆発

- ・ 重力崩壊型超新星爆発:
 - 大質量星の最期の爆発現象(エネルギー 10^{51} erg)
- ・ 星のコアが中性子星に崩壊する時に解放される重力エネルギー(10^{53} erg)がエネルギー源
- ・ 残りはニュートリノとして放射



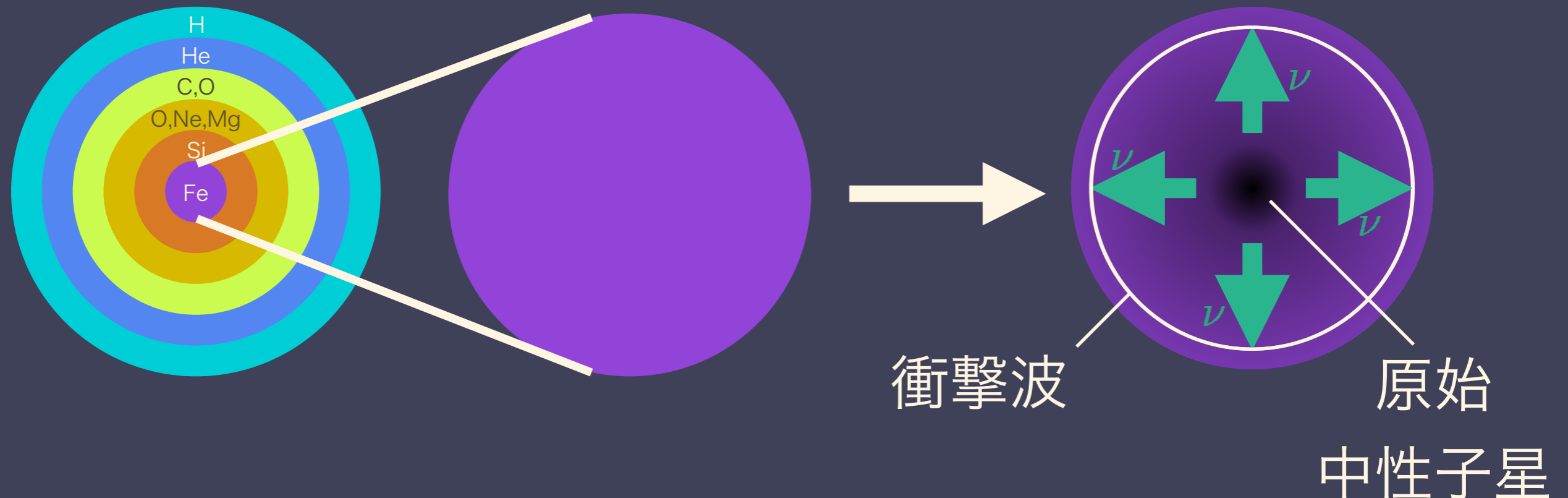
SN1987A

©NASA, ESA/Hubble

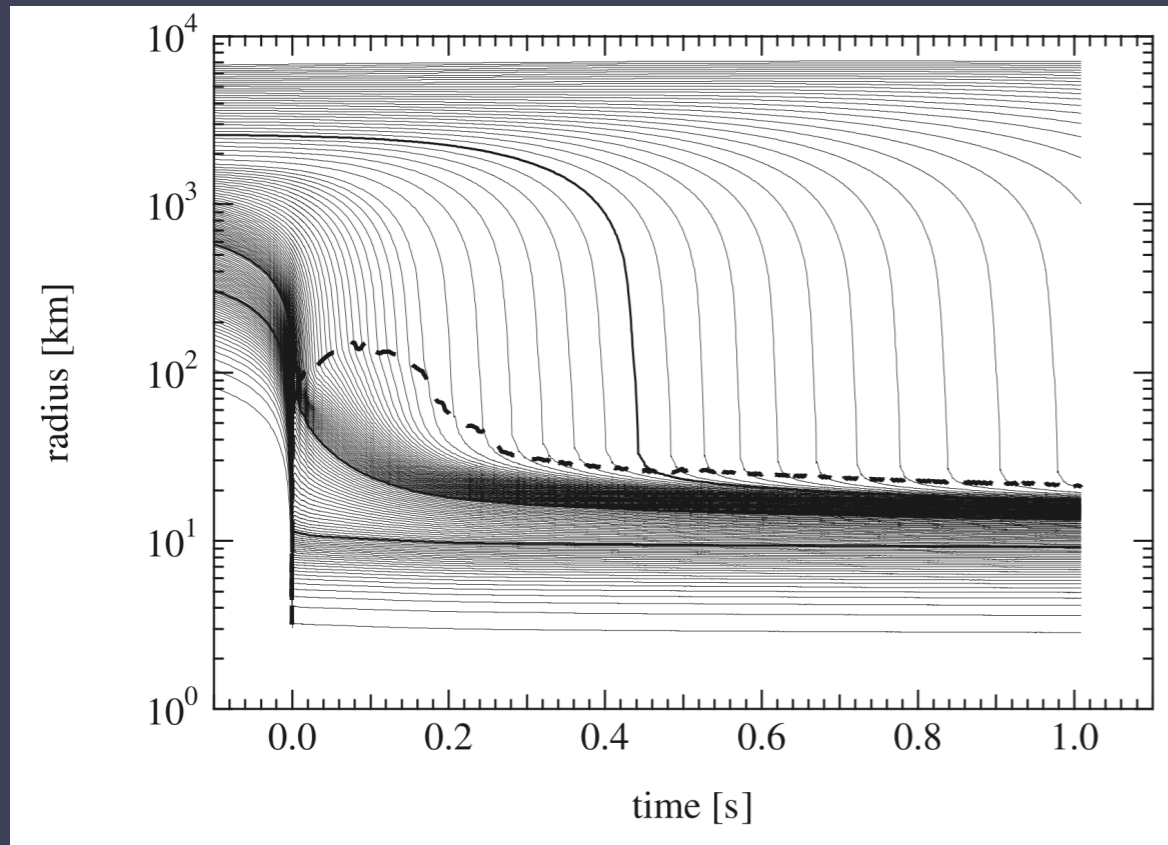
超新星爆発メカニズム

- ・ 重力崩壊→コアバウンス→衝撃波停滞
- ・ 中心の原始中性子星からのニュートリノ放射
- ・ 衝撃波がニュートリノ加熱によって復活

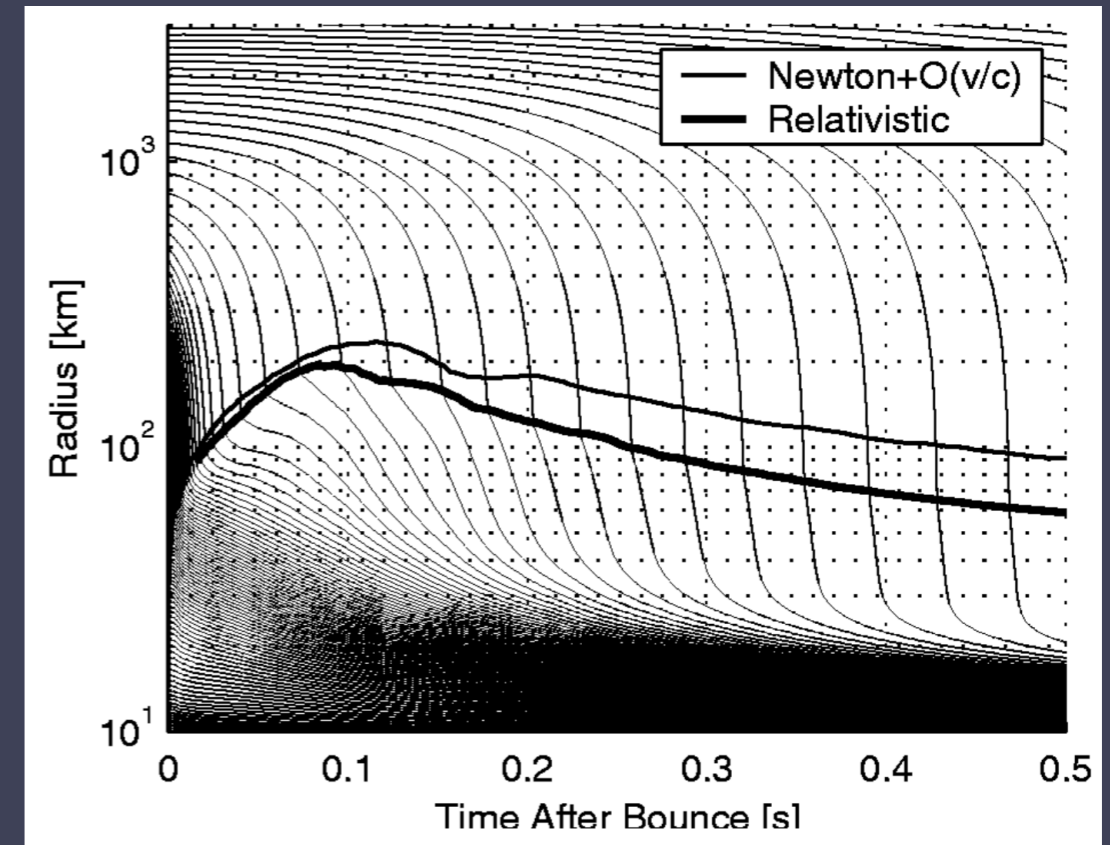
→ニュートリノ加熱メカニズム



超新星シミュレーション



Sumiyoshi+ (2005)

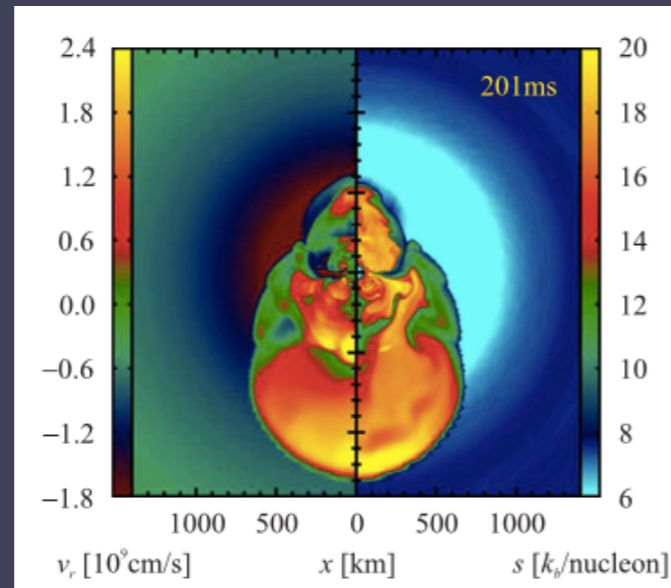


Liebendoerfer+ (2001)

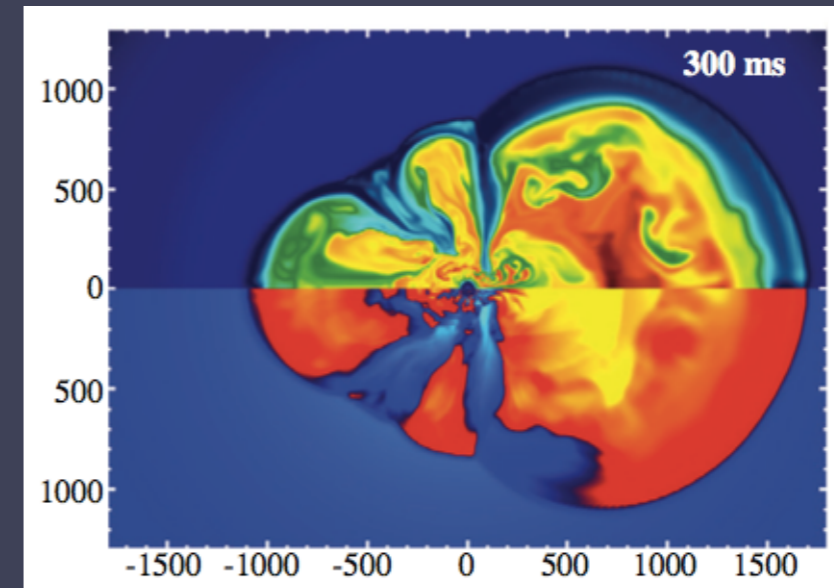
- ・ 1次元球対称シミュレーションではどんなに頑張っても爆発しない

超新星シミュレーション

2D

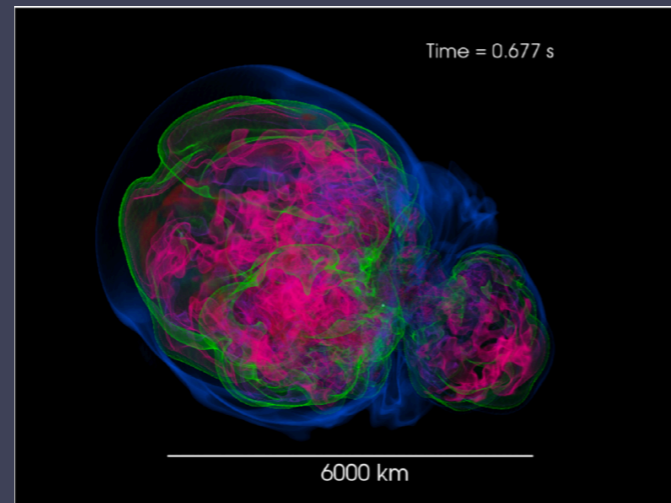


MPA (Mueller+ 2012)

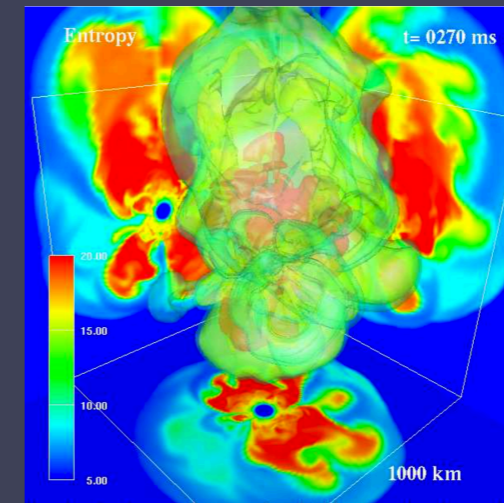


Oak Ridge (Bruenn+ 2013)

3D



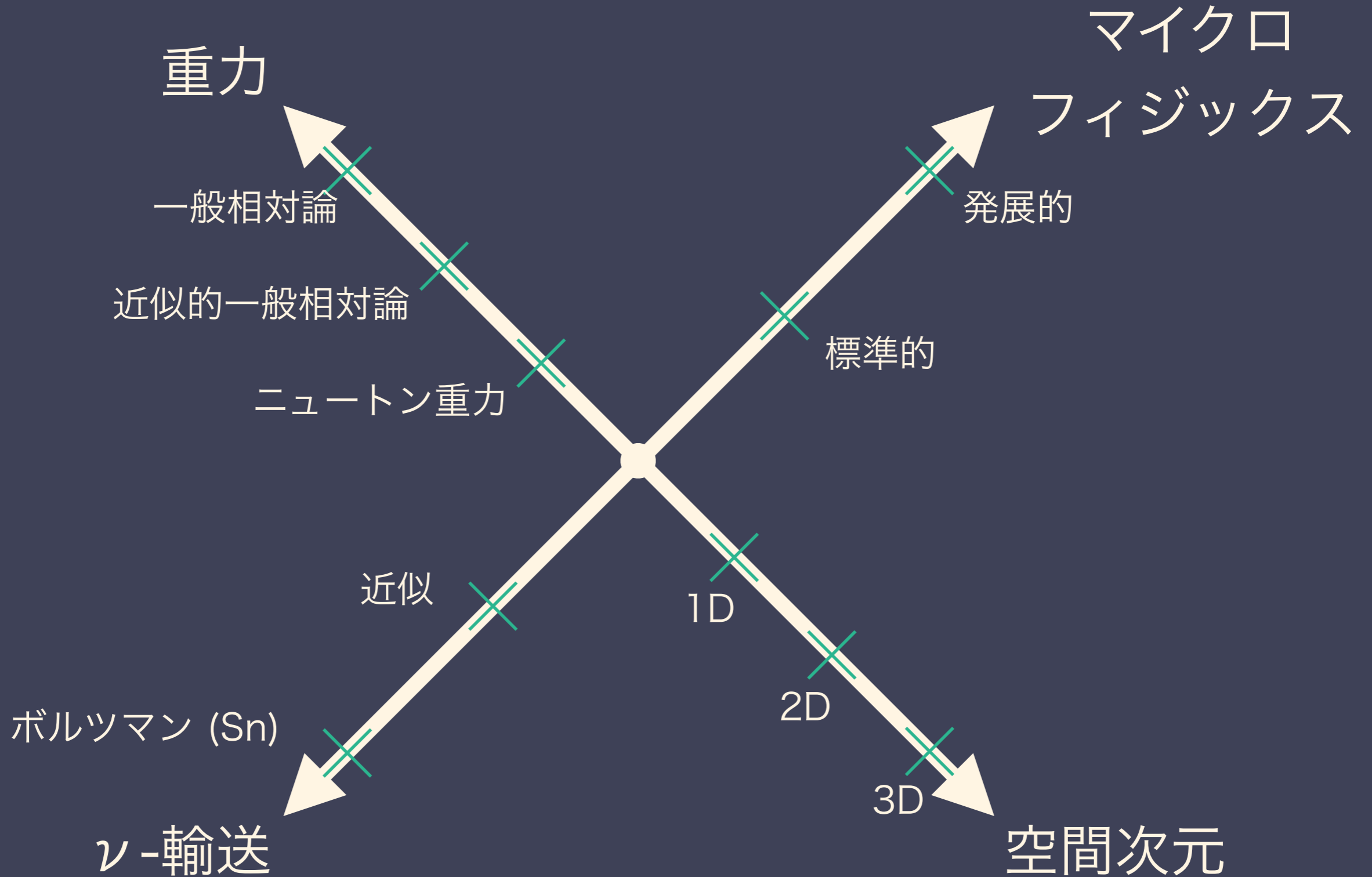
Princeton (Vartanyan+ 2019)



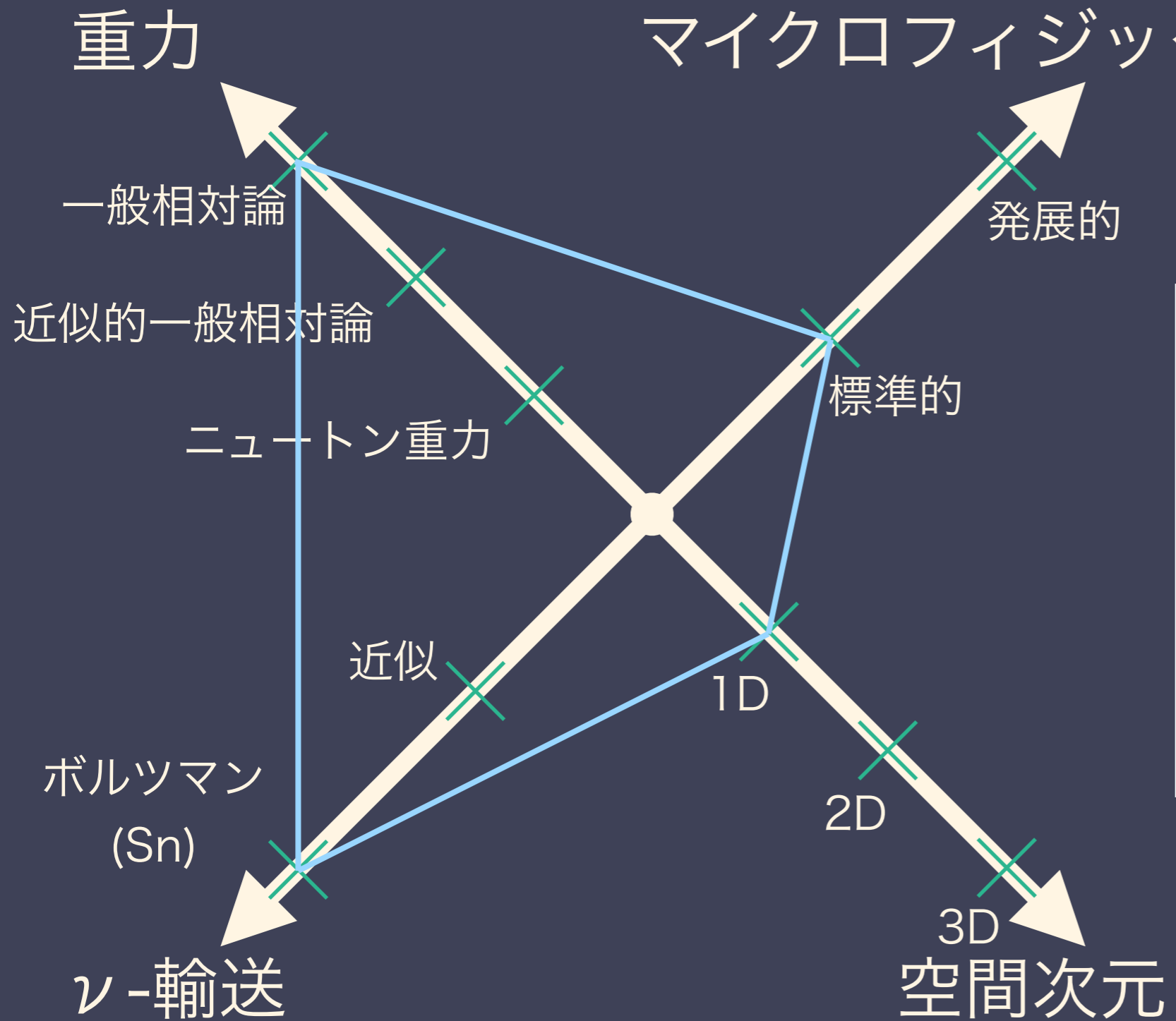
Fukuoka (Takiwaki+ 2014)

- ・ 1次元球対称シミュレーションではどんなに頑張っても爆発しない
- ・ 多次元シミュレーションだと爆発する

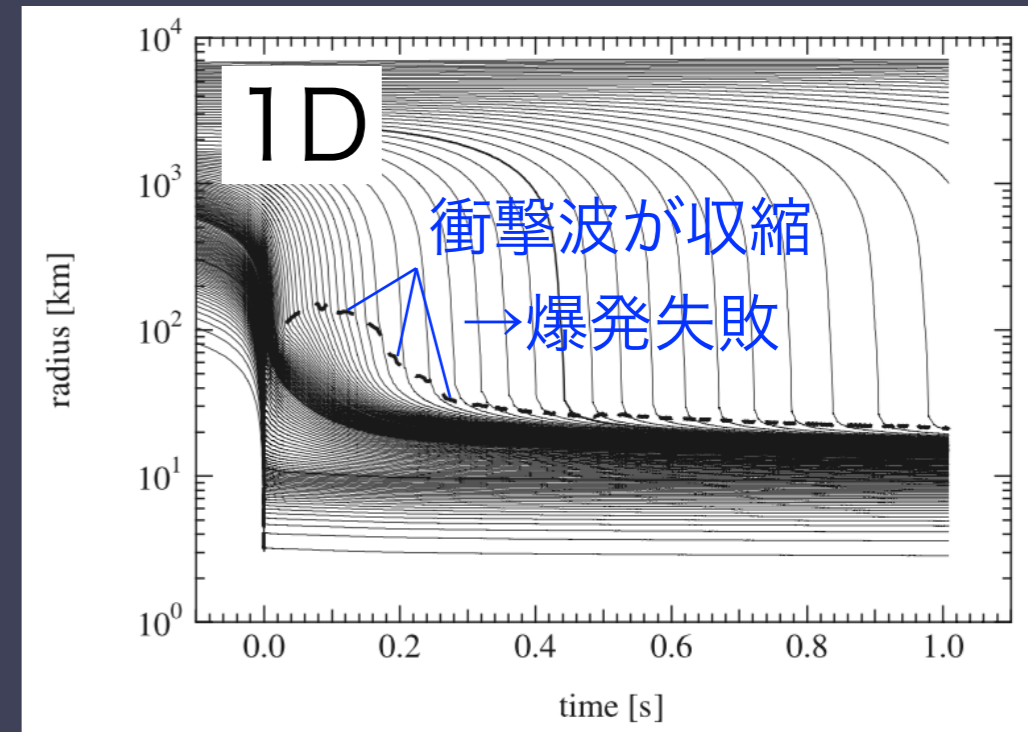
超新星シミュレーションの進展



超新星シミュレーションの進展

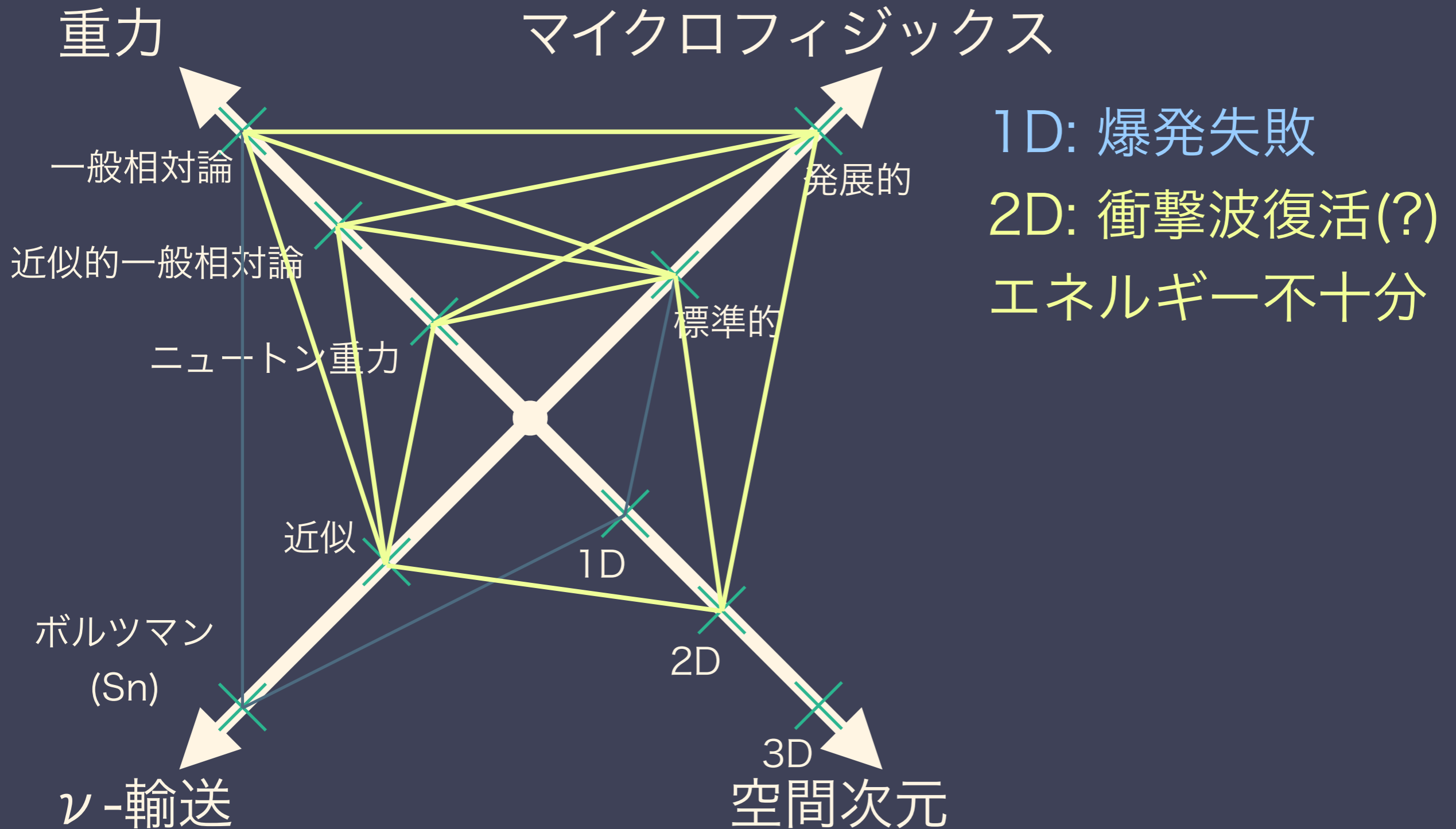


1D: 爆発失敗

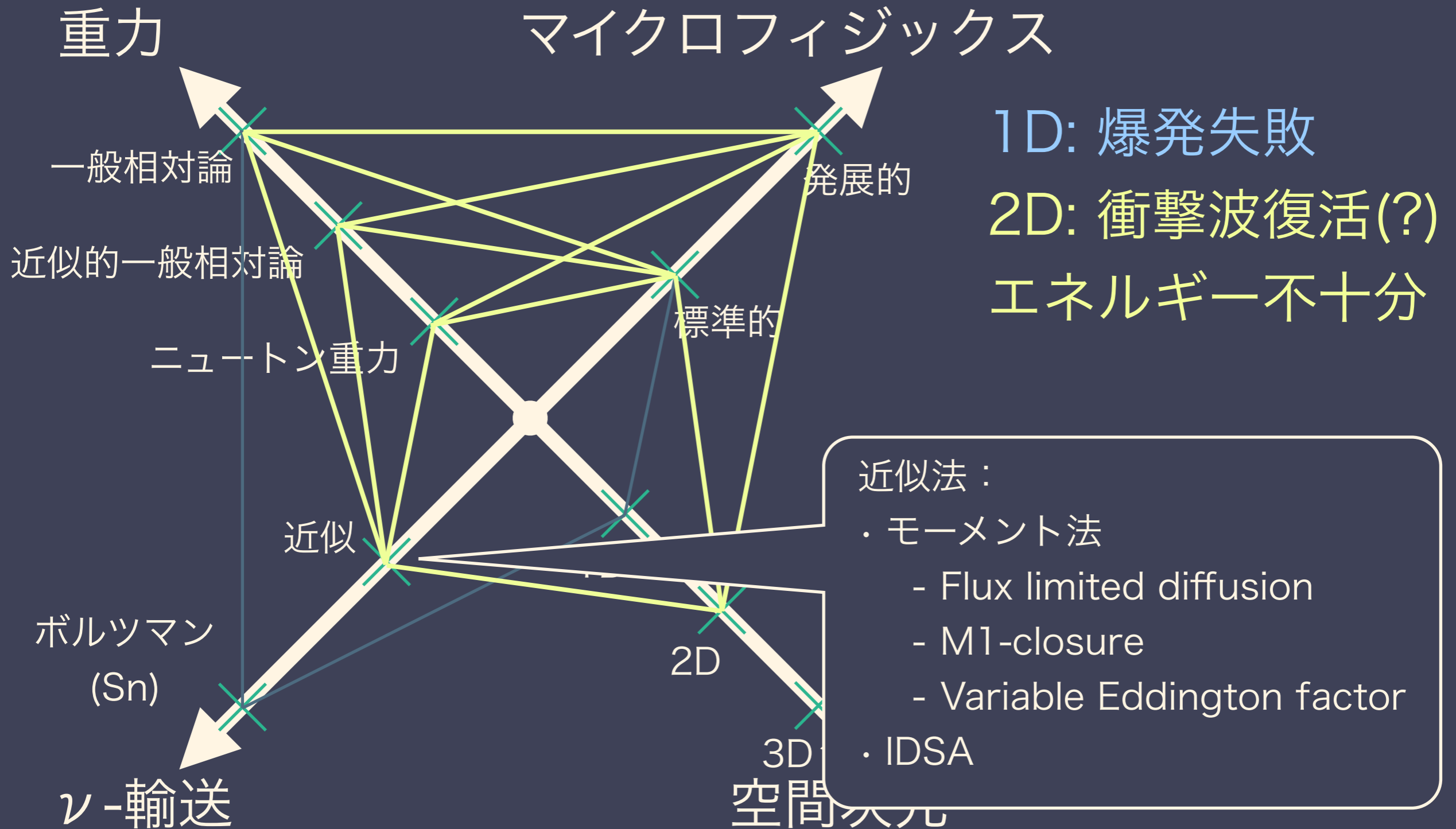


Sumiyoshi+(2005)

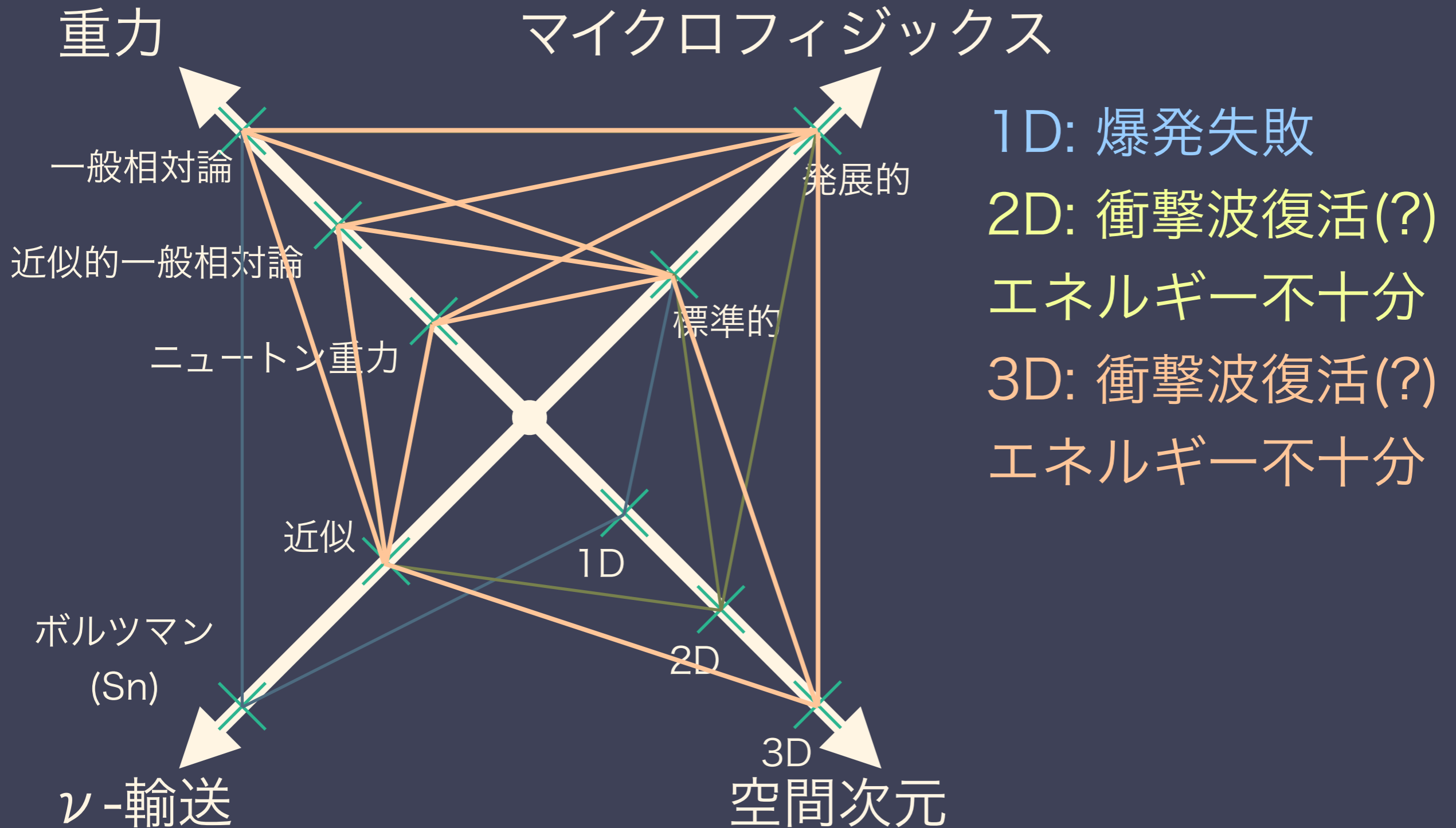
超新星シミュレーションの進展



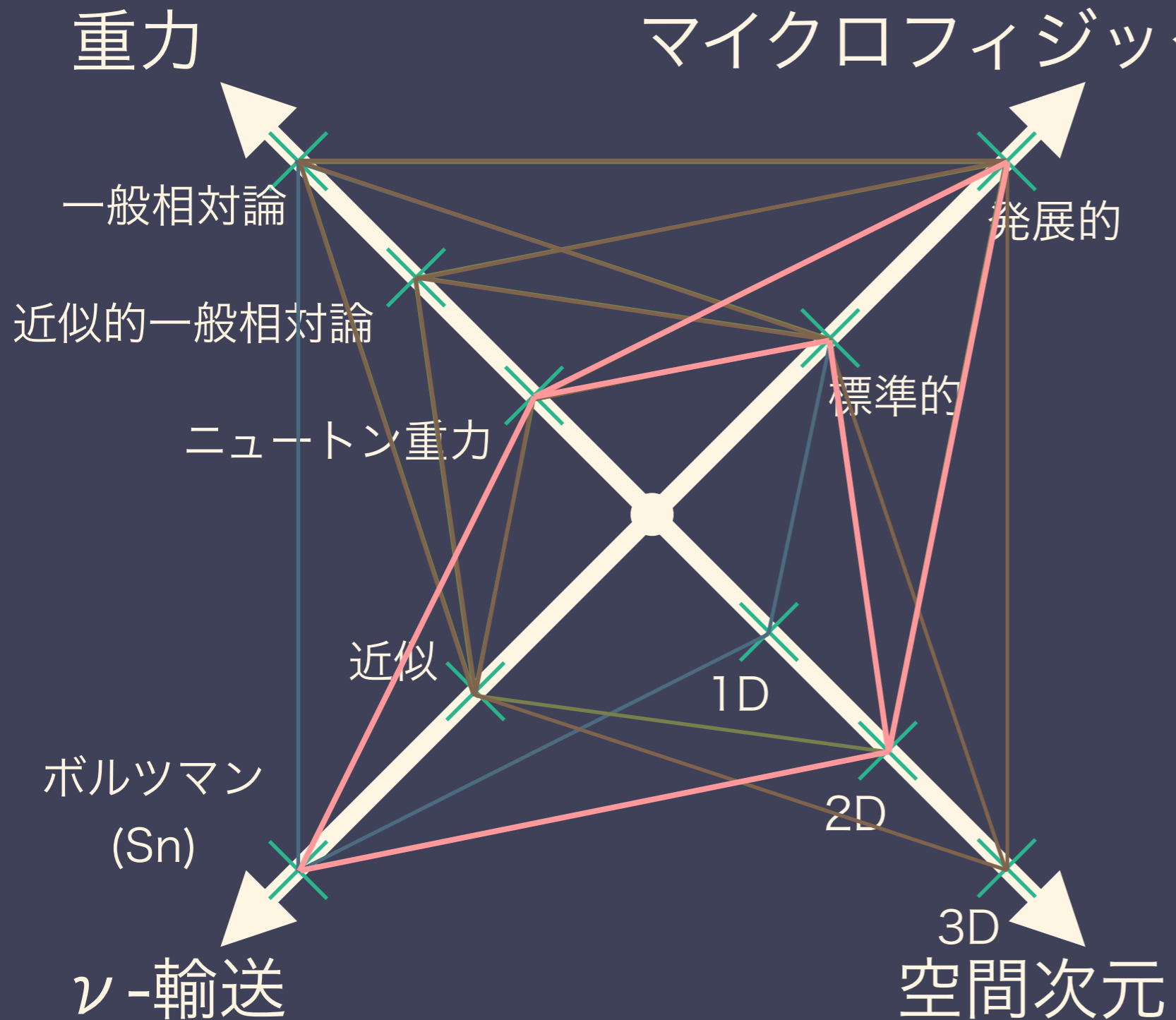
超新星シミュレーションの進展



超新星シミュレーションの進展



超新星シミュレーションの進展



- 1D: 爆発失敗
- 2D: 衝撃波復活(?)
エネルギー不十分
- 3D: 衝撃波復活(?)
エネルギー不十分
原田らの計算

モーメント法輻射輸送

・ボルツマン方程式

$$p^\alpha \frac{\partial f}{\partial x^\alpha} \Big|_{p^i} - \Gamma^i_{\alpha\beta} p^\alpha p^\beta \frac{\partial f}{\partial p^i} \Big|_{x^\alpha} = (-p^\alpha \hat{u}_\alpha) S_{\text{rad}}$$

0次モーメント

$$\int d\Omega$$

$$\frac{\partial E}{\partial t} + \frac{\partial F^i}{\partial x^i} = S_0$$

$$\frac{\partial F^i}{\partial t} + \frac{\partial P^{ij}}{\partial x^j} = S_1^i$$

1次モーメント

$$\int d\Omega p^i$$

$$E \sim \int d\Omega p^0 f$$

$$F^i \sim \int d\Omega p^i f$$

$$P^{ij} \sim \int d\Omega p^i p^j f$$

エディントンテンソル

- ・ 分布の2次モーメント=分布の0次・1次モーメントの関数
- ・ 1次モーメント式までで閉じる→M1-closure法

$$E^{ij} = \begin{cases} \frac{P^{ij}}{E} & \text{エディントンテンソルの定義} \\ \frac{3\chi - 1}{2} \delta^{ij} + \frac{1 - \chi}{2} \frac{F^i F^j}{F^2} & \text{M1-closure法によるもの} \end{cases}$$

$$w/\chi = \frac{3 + 4\tilde{F}^2}{5 + 2\sqrt{4 - 3\tilde{F}^2}}, \quad \tilde{F} = \frac{|F|}{E}$$

エディントン因子 フラックス因子

エディントンテンソル

- ・ 分布の2次モーメント=分布の0次・1次モーメントの関数
- ・ 1次モーメント式までで閉じる→M1-closure法

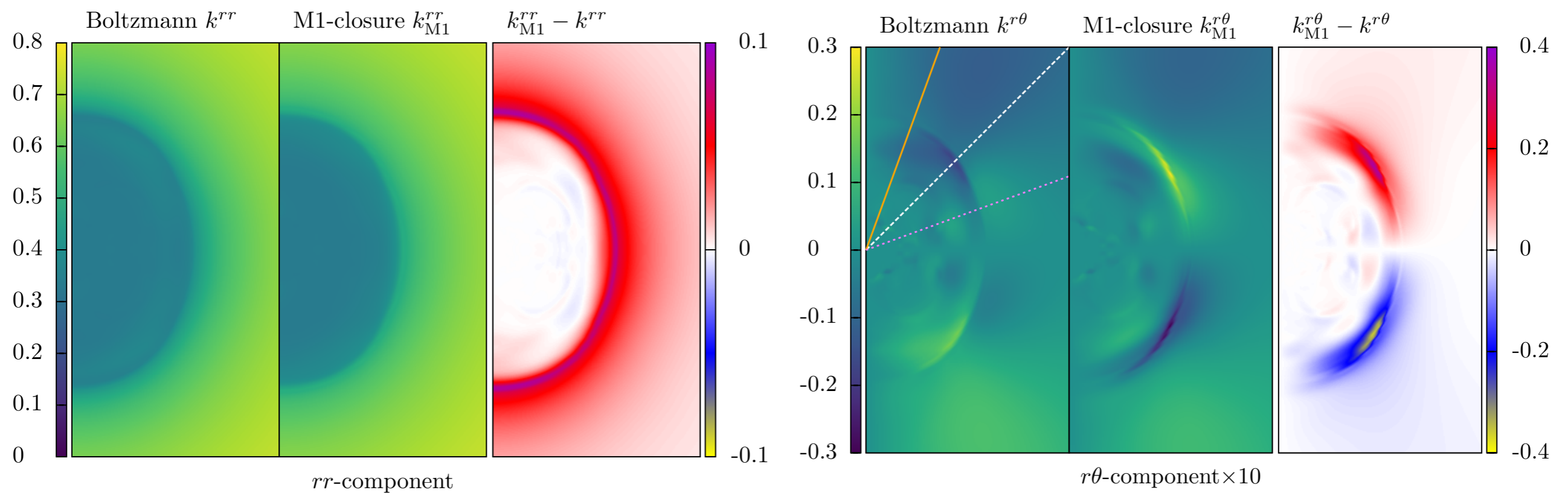
$$E^{ij} = \begin{cases} \frac{P^{ij}}{E} & \text{ボルツマン-} \\ & \text{エディントンテンソル} \\ \frac{3\chi - 1}{2} \delta^{ij} + \frac{1 - \chi}{2} \frac{F^i F^j}{F^2} & \\ & \text{M1-エディントンテンソル} \end{cases}$$

$$w/\chi = \frac{3 + 4\tilde{F}^2}{5 + 2\sqrt{4 - 3\tilde{F}^2}}, \quad \tilde{F} = \frac{|\mathbf{F}|}{E}$$

エディントン因子 フラックス因子

エディントン因子のズレ

- ・ バウンス後 ~10 msでのエディントンテンソル
- ・ エディントン因子そのものの評価
- ・ 速度に依存した効果の評価



超新星シミュレーションの進展

重力

マイクロフィジックス

1D: 爆発失敗

ボルツマン計算は大変

→近似的輸送法に何らかのfeedbackを与えて、
計算量を抑えつつボルツマン計算の結果を再現できないか？

→機械学習の関数近似能力に着目

(Sn)
ν-輸送

2D

3D
空間次元

機械学習

- ・ 近年急速に発達し、様々な場面で使われている
 - 画像認識
 - 自動翻訳
- ・ 天文でも
 - 銀河の形態認識 (Kojima et al. 2020など)
 - 超新星の分類 (Takahashi et al. 2020など)
 - その他いろいろ
- ・ 特にDeep Neural Network (DNN)に着目

Deep Neural Network

- ・ DNNによる機械学習
 - 入力値の関数として、入力がとあるものに対応する確率を返す装置
 - 例：入力された画像が猫である確率を返す関数(猫、犬、人である確率をまとめてベクトルとして返すことも)
- ・ 関数fit装置としてのDNN→万能近似定理

DNNの基礎

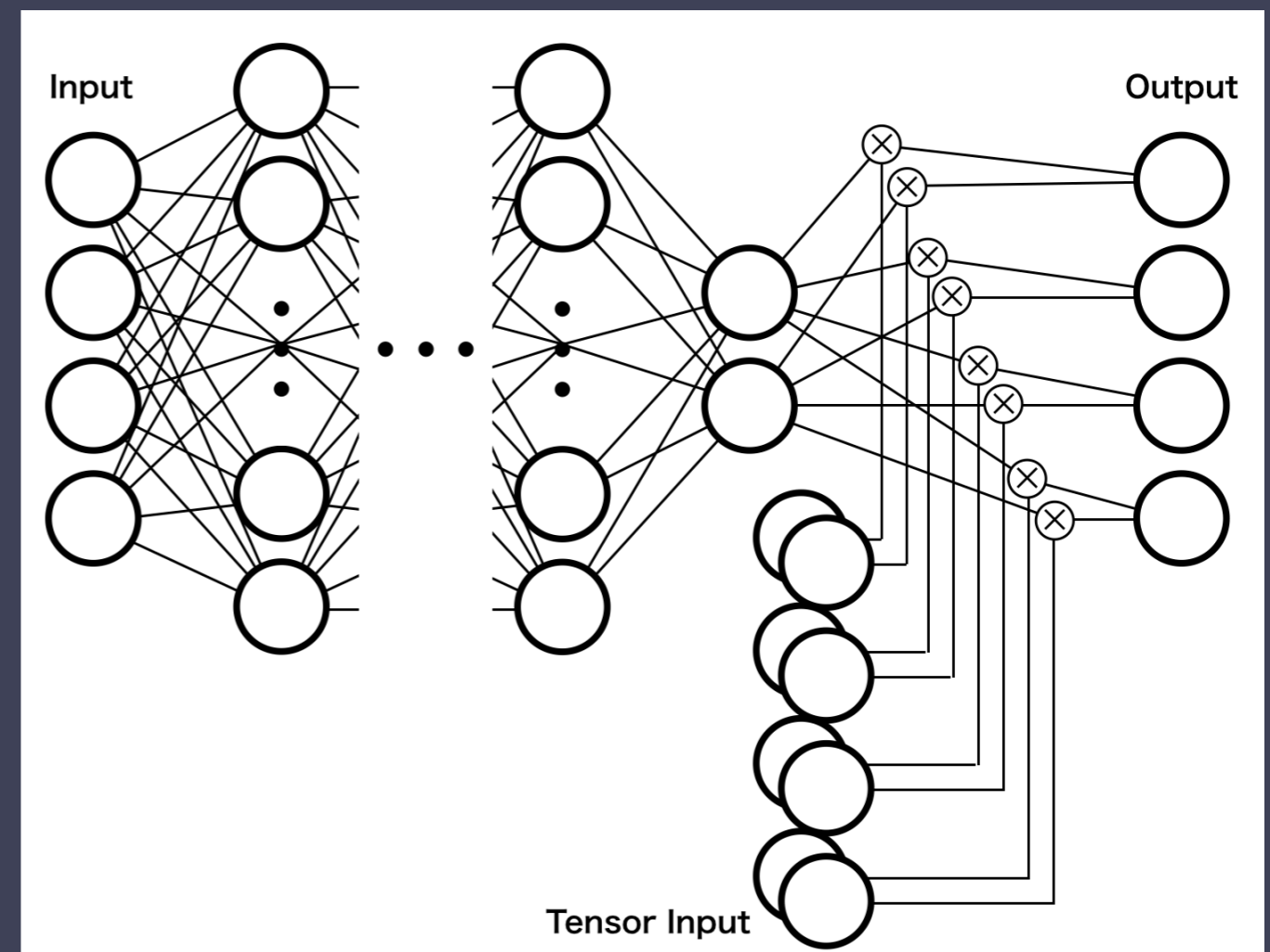
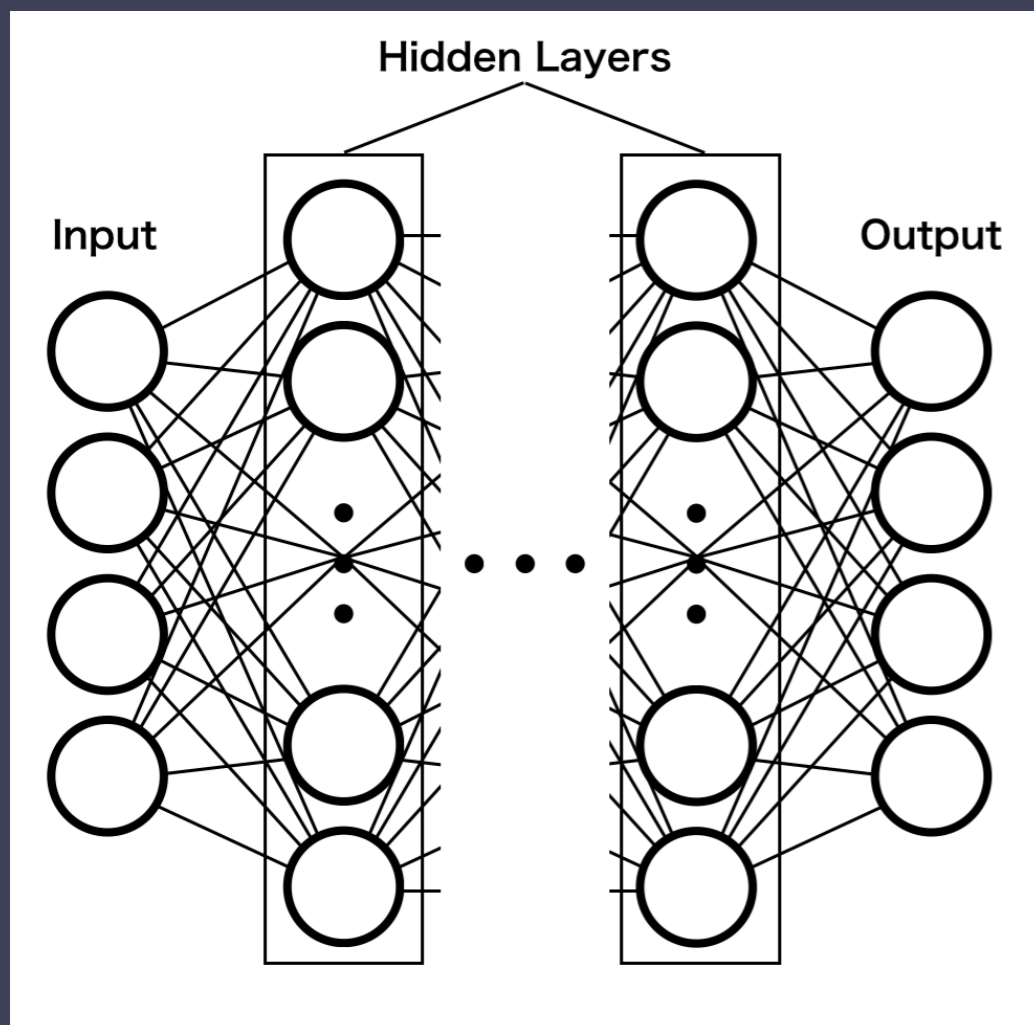
- ・ 板書
- ・ ポイントのまとめ
 - ▶ 人工ニューロンを組み合わせたDNNは連続関数を任意の精度で近似する能力を持つ
 - ▶ DNNの振る舞いを決める重みパラメータは最小二乗損失関数を最小化するようSGDで更新していく
 - ▶ 損失関数の勾配は誤差逆伝搬法により、代数計算で求める

用意したデータ

- ・ ボルツマンコードによる超新星計算(爆発はせず)
 - Woosley et al. (2002)の15 M_{\odot} 親星モデル
 - 富樫-古澤の核物質状態方程式モデル(Furusawa et al. 2017)
 - メッシュは384 (r)×128 (θ)×20 (ε)×10 (θ_{ν})×6 (ϕ_{ν})
- ・ 100 msのスナップショット
- ・ 半径100 km以内の各空間グリッド、各エネルギーbinについて実験室系の
 - エネルギー密度
 - エネルギーフラックス
 - 流体の速度
 - エディントンテンソル
- ・ →合計約60万データ
- ・ 8割のデータを学習に使用、残り2割で汎化性能評価

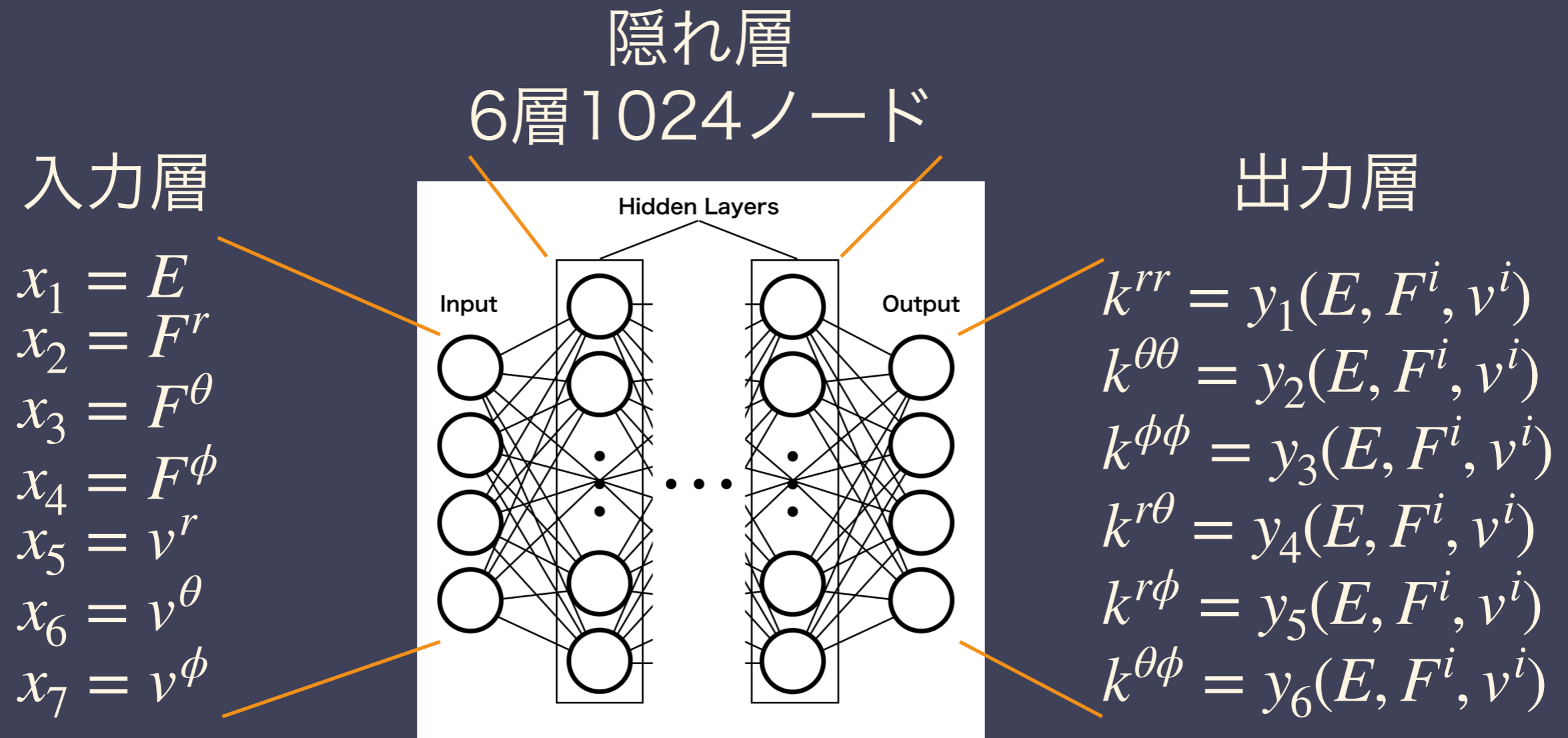
ネットワークの構成

- ・ 機械学習ライブラリKeras/Tensorflowを利用
- ・ Component-wise Neural Network (CWNN) と Tensor Basis Neural Network (TBNN)



ネットワークの構成

- Component-wise Neural Network (CWNN)



ネットワークの構成

- ・ Tensor Basis Neural Network (TBNN; Ling et al. 2016)
- ・ DNNの出力を係数とし、関係ありそうなテンソルの線形結合を取る

$$k^{ij} = y_1 F^i F^j + y_2 \delta^{ij} + y_3 (F^i v^j + v^i F^j) + y_4 v^i v^j$$

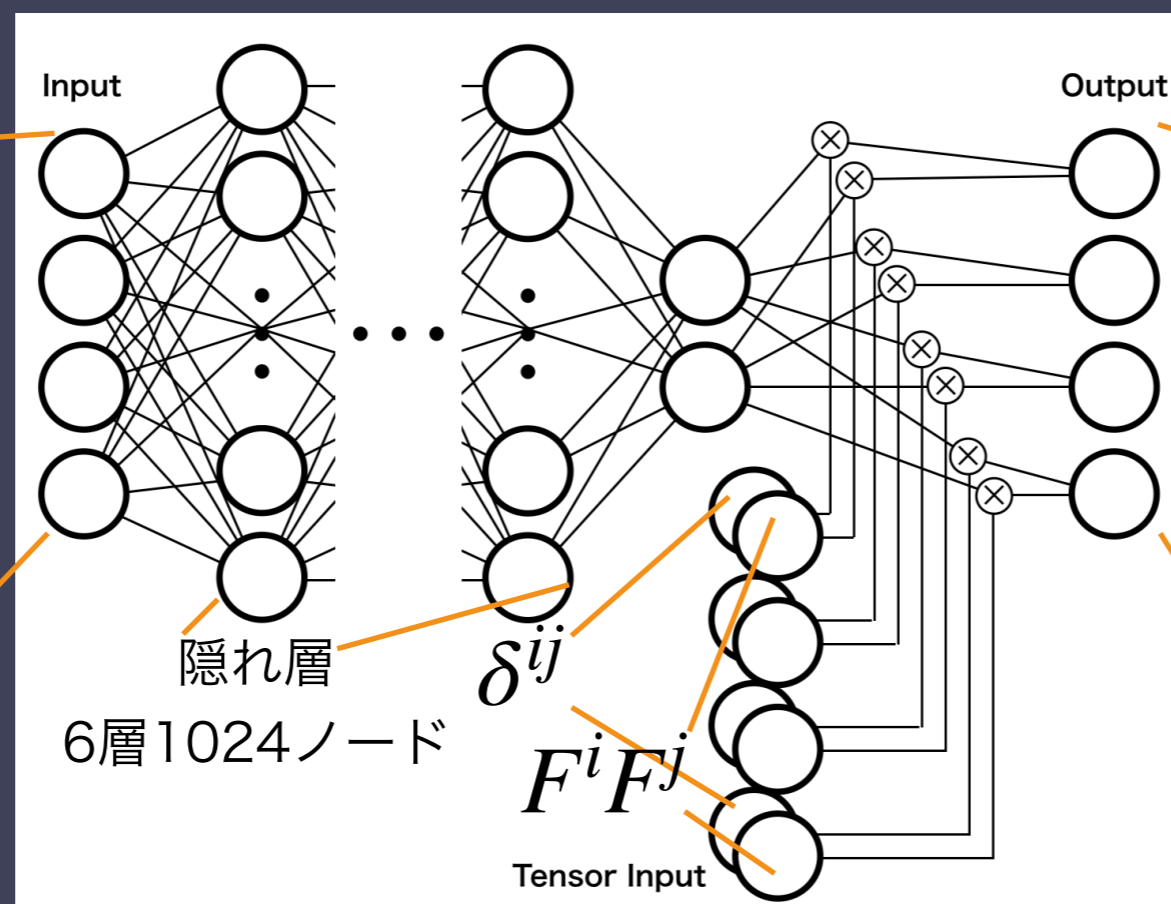
ネットワークの構成

- Tensor Basis Neural Network (TBNN; Ling et al. 2016)
- DNNの出力を係数とし、関係ありそうなテンソルの線形結合を取る

$$k^{ij} = y_1 F^i F^j + y_2 \delta^{ij} + y_3 (F^i v^j + v^i F^j) + y_4 v^i v^j$$

入力層

$$\begin{aligned} x_1 &= E \\ x_2 &= F^r \\ x_3 &= F^\theta \\ x_4 &= F^\phi \\ x_5 &= v^r \\ x_6 &= v^\theta \\ x_7 &= v^\phi \end{aligned}$$

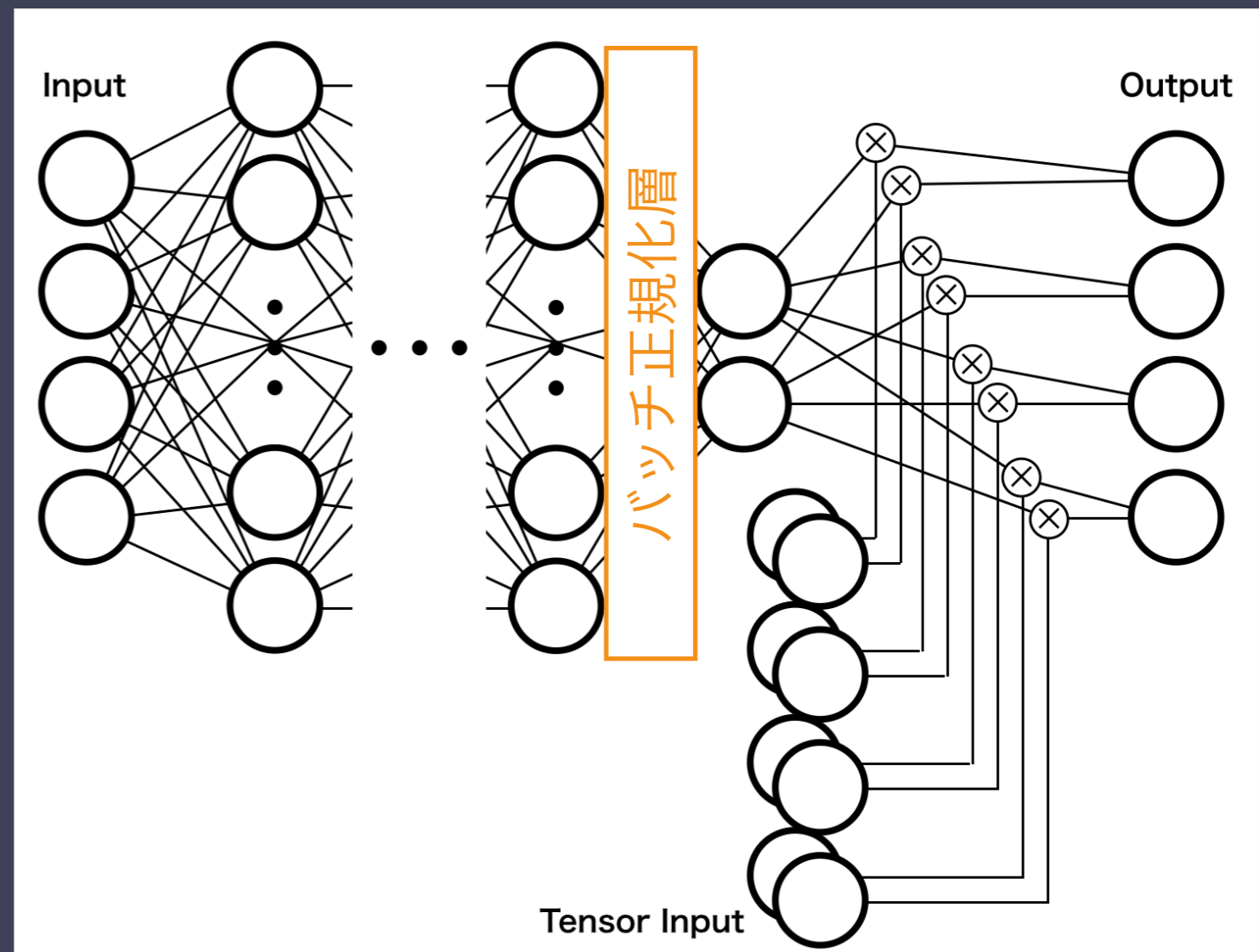
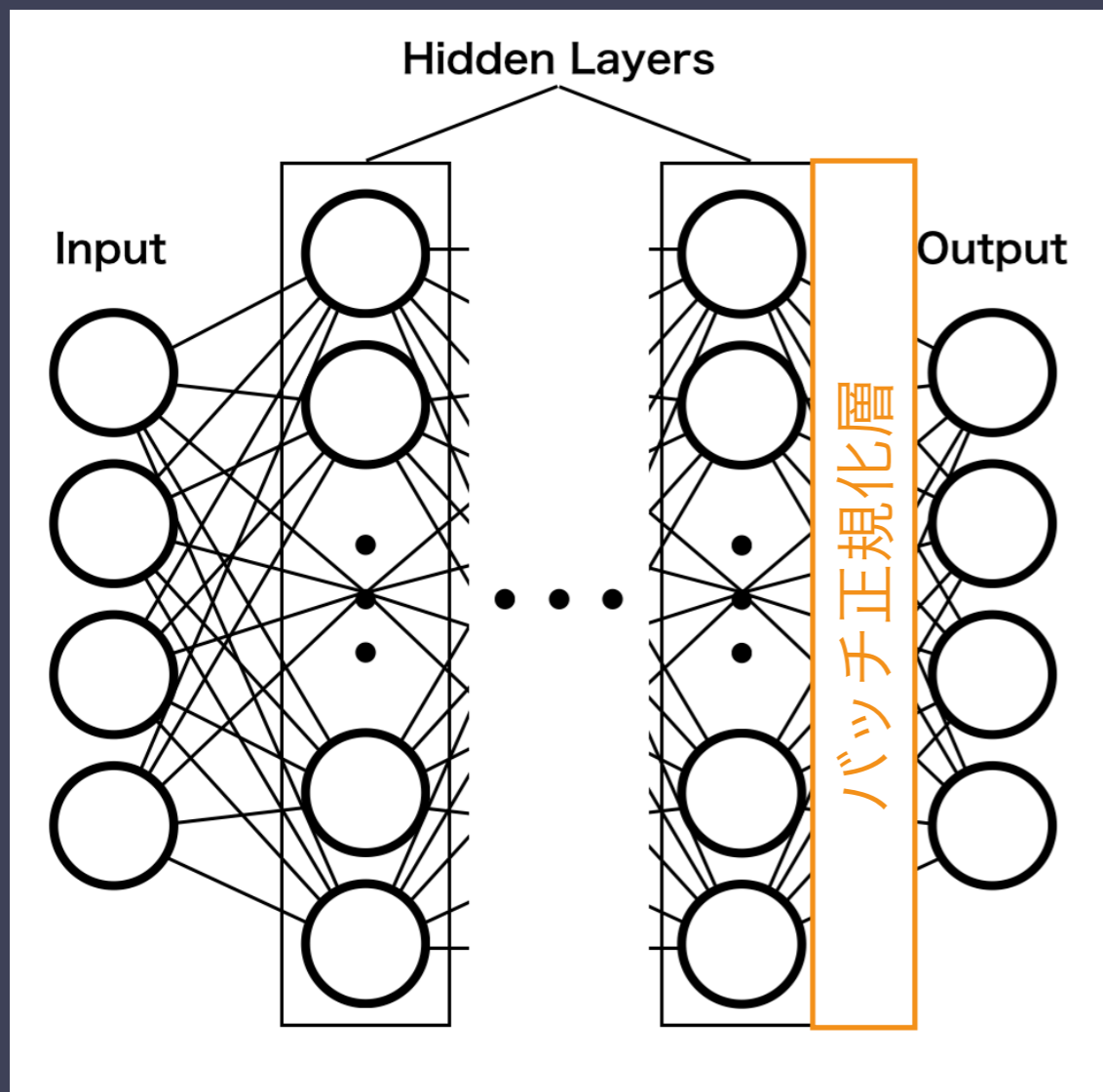


出力層

$$\begin{aligned} k^{rr} &= y_1 (E, F^i, v^i) \\ k^{\theta\theta} &= y_2 (E, F^i, v^i) \\ k^{\phi\phi} &= y_3 (E, F^i, v^i) \\ k^{r\theta} &= y_4 (E, F^i, v^i) \\ k^{r\phi} &= y_5 (E, F^i, v^i) \\ k^{\theta\phi} &= y_6 (E, F^i, v^i) \end{aligned}$$

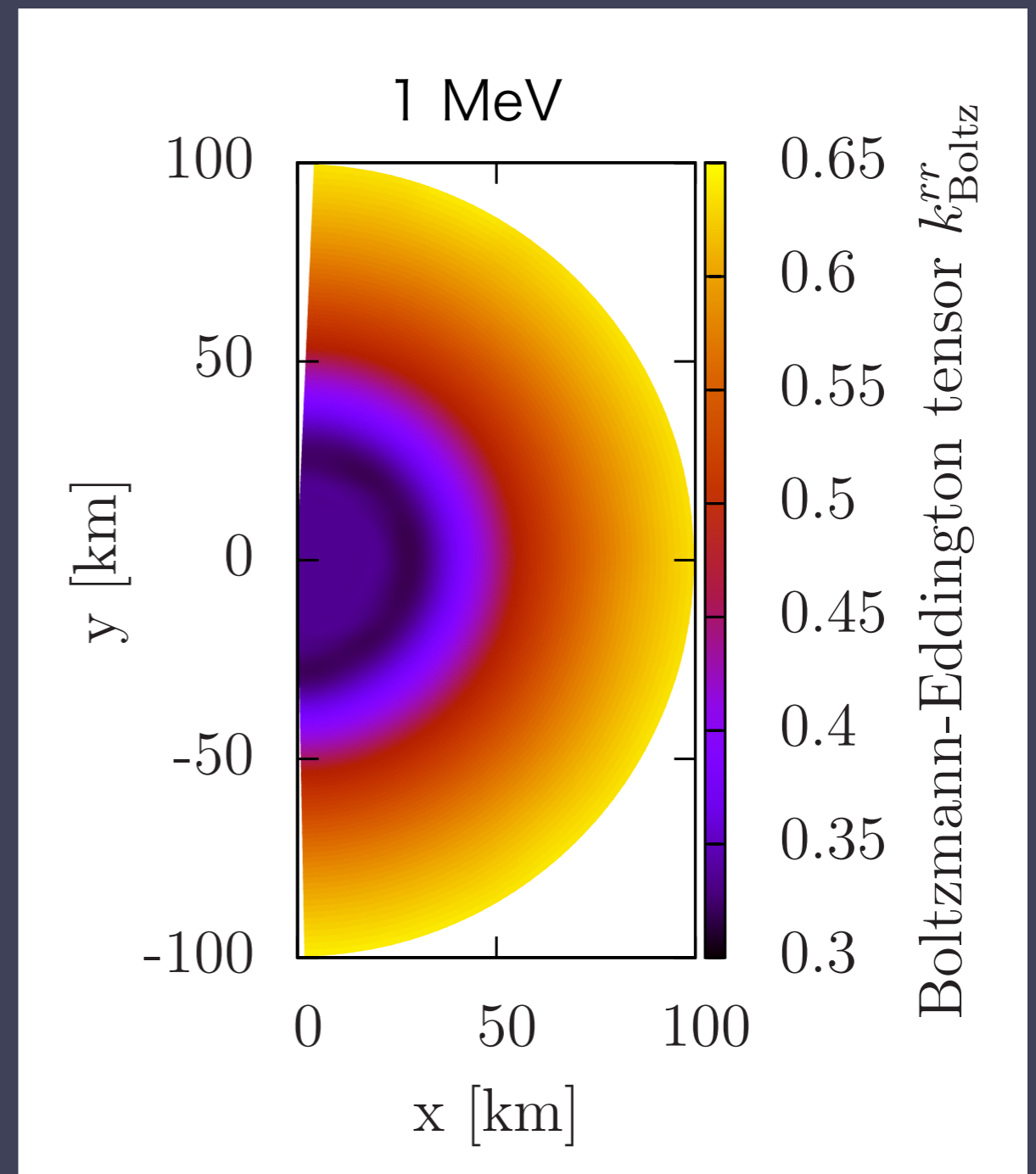
バッチ正規化

- 学習をうまく行かせるために、バッチ正規化の層を一つ挟む



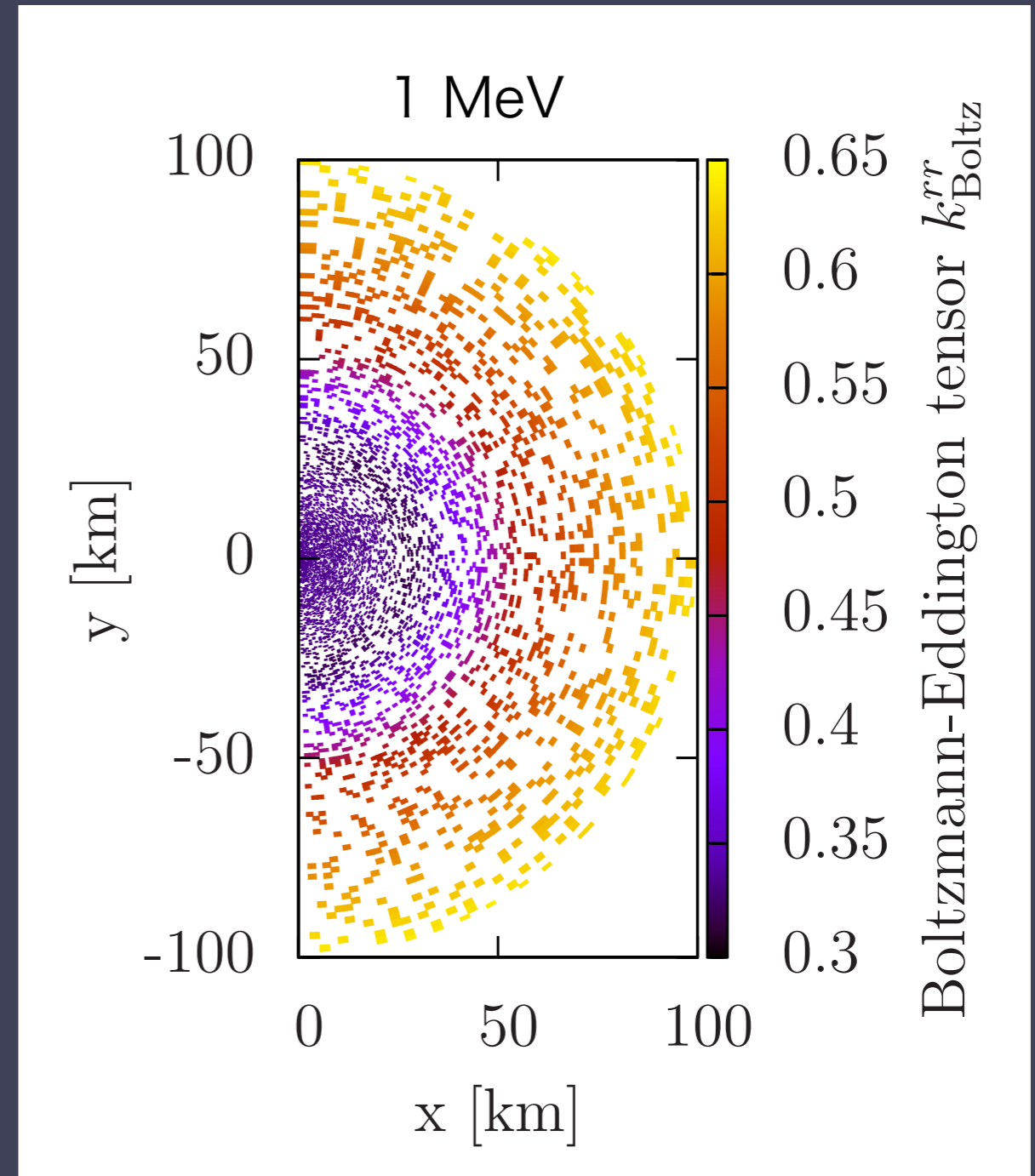
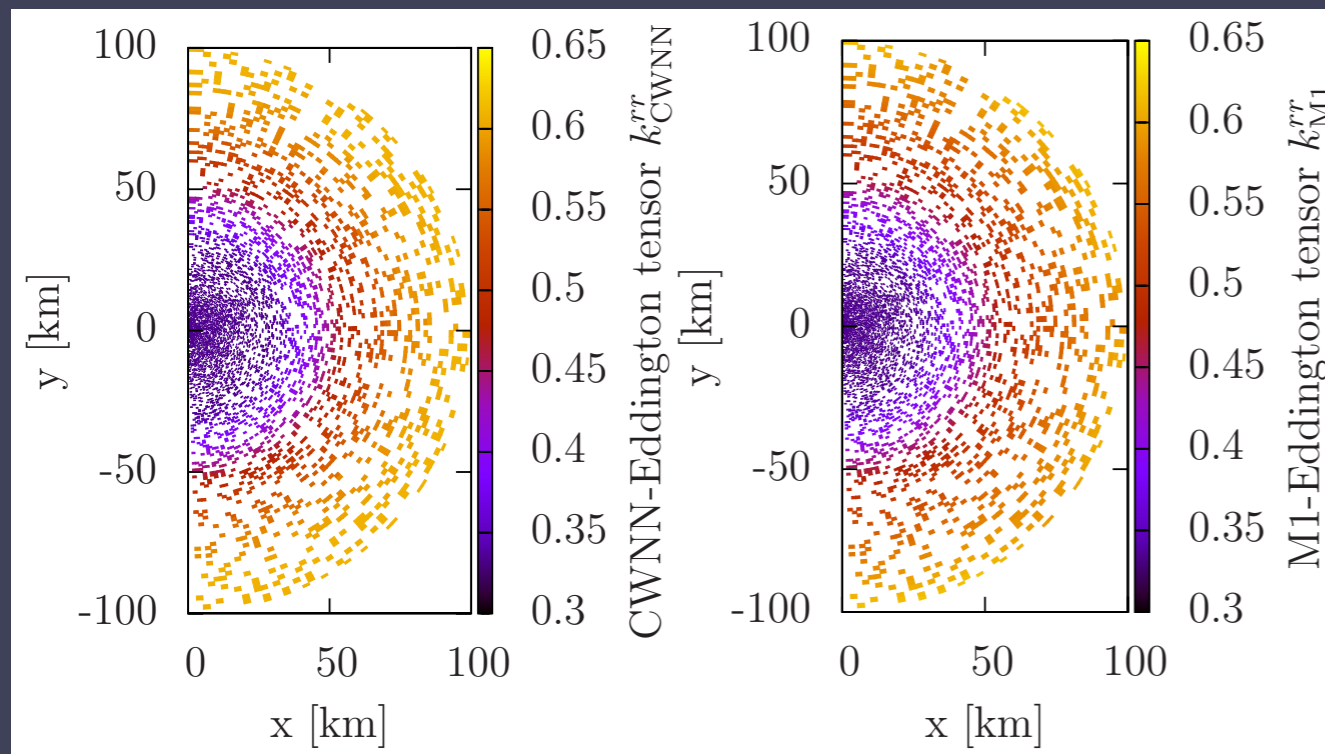
CWNNの結果(対角成分)

- もともとのデータ

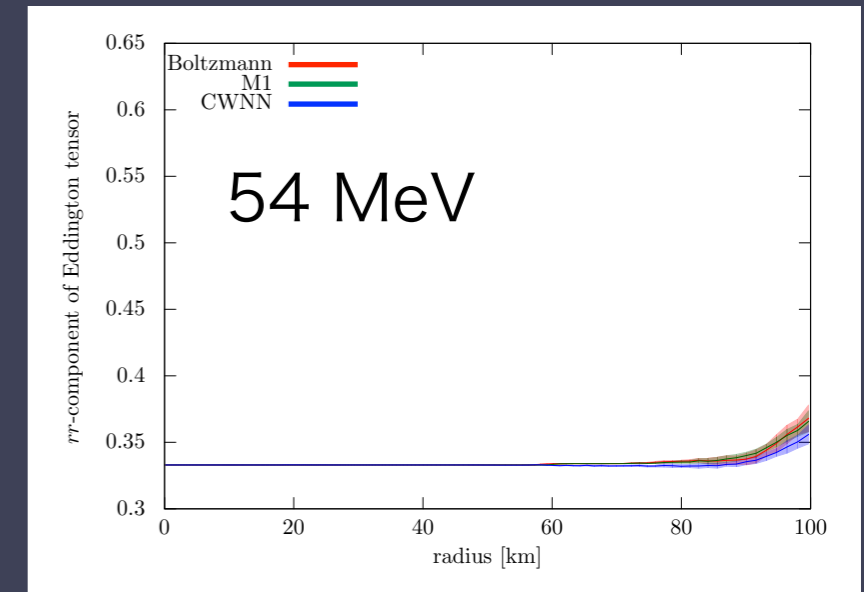
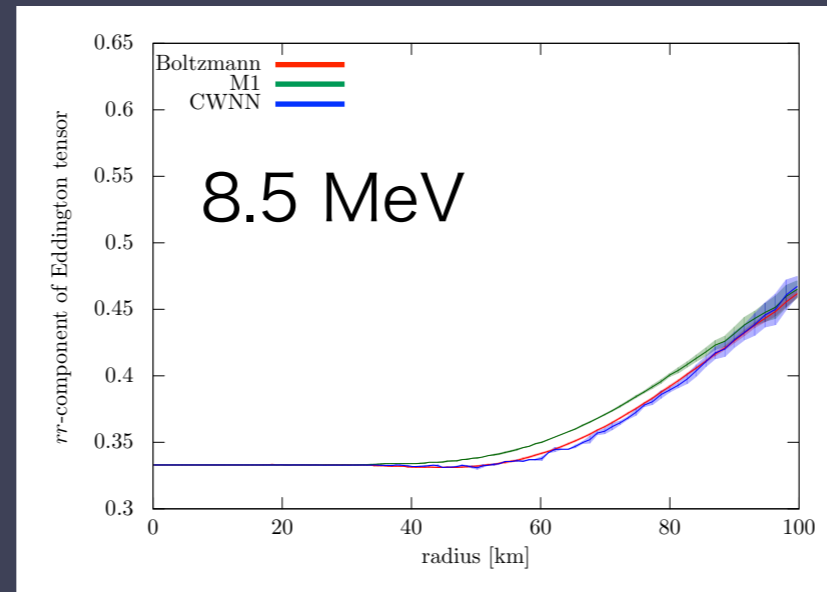
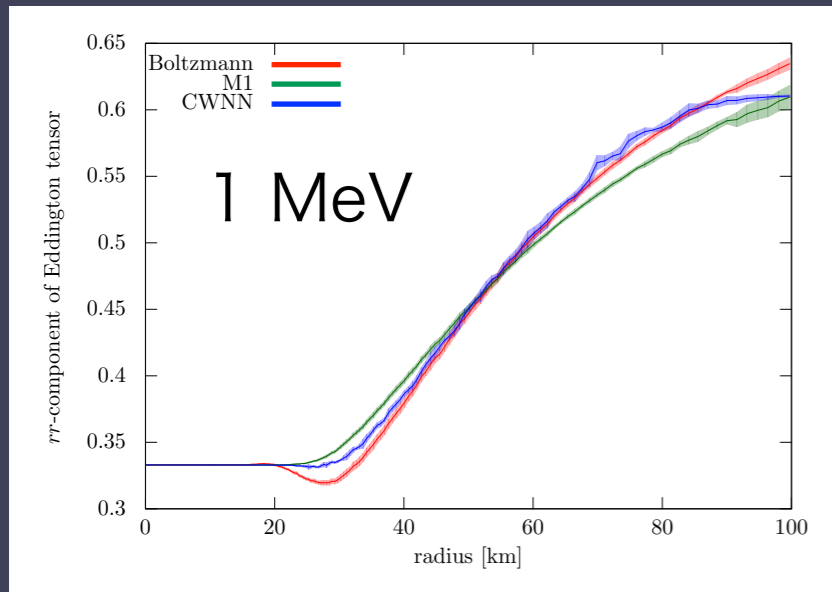


CWNNの結果(対角成分)

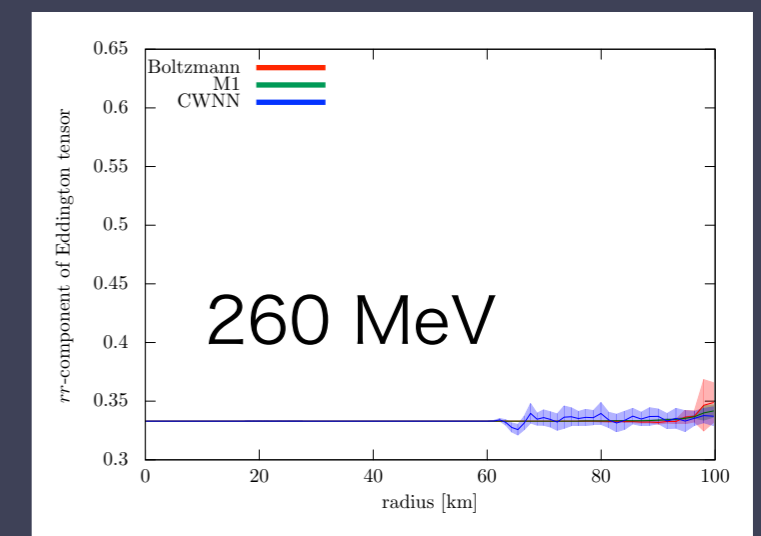
- ・ 学習に8割使った、残りの2割
- ・ 動径分布をプロット



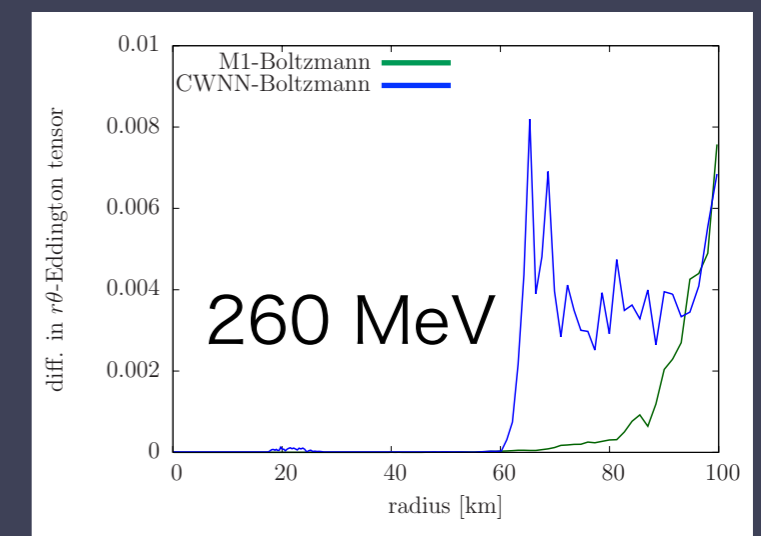
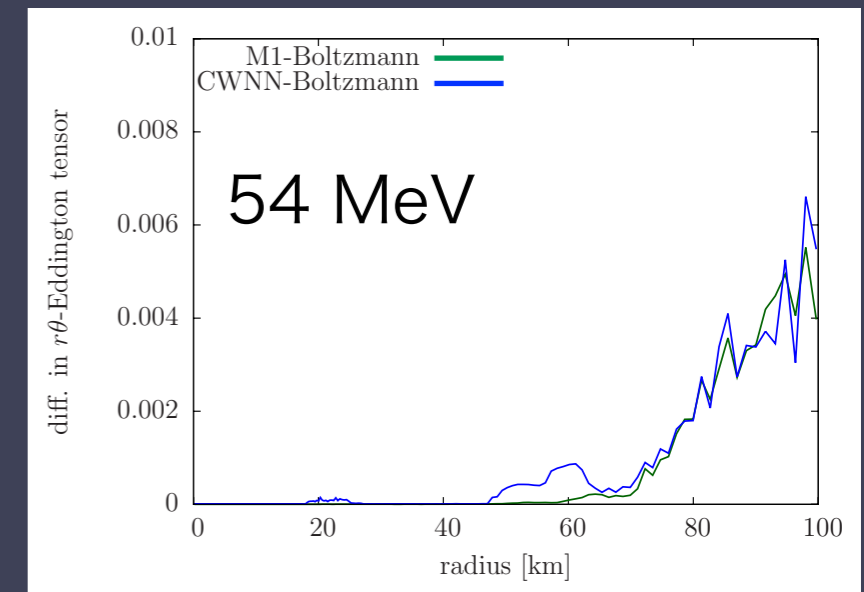
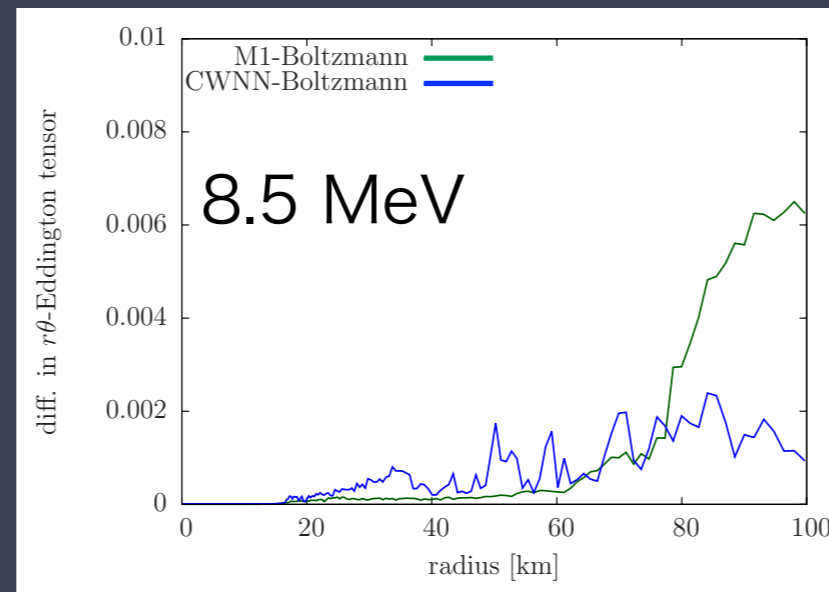
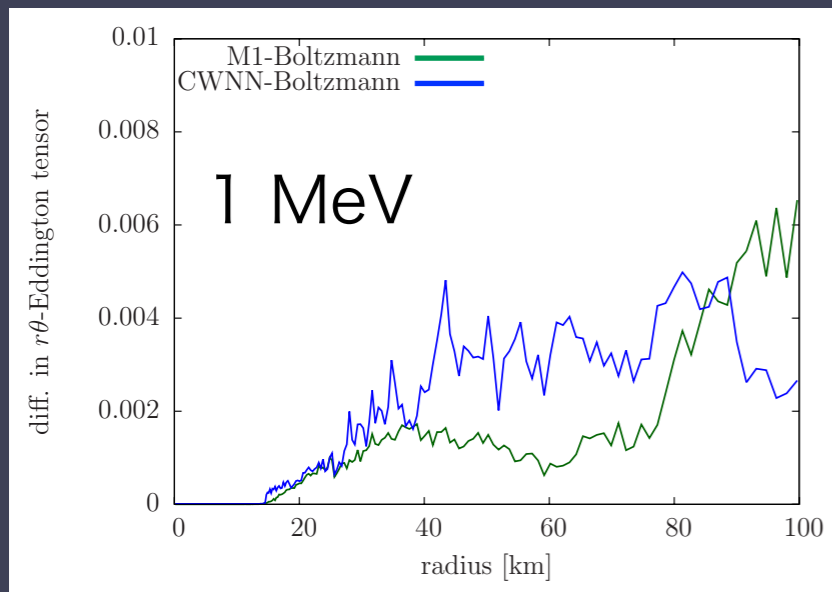
CWNNの結果(対角成分)



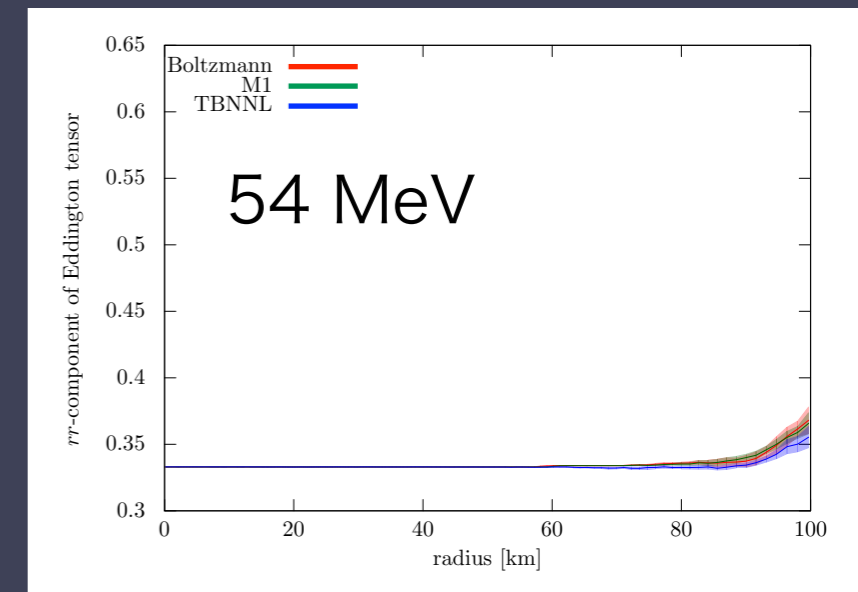
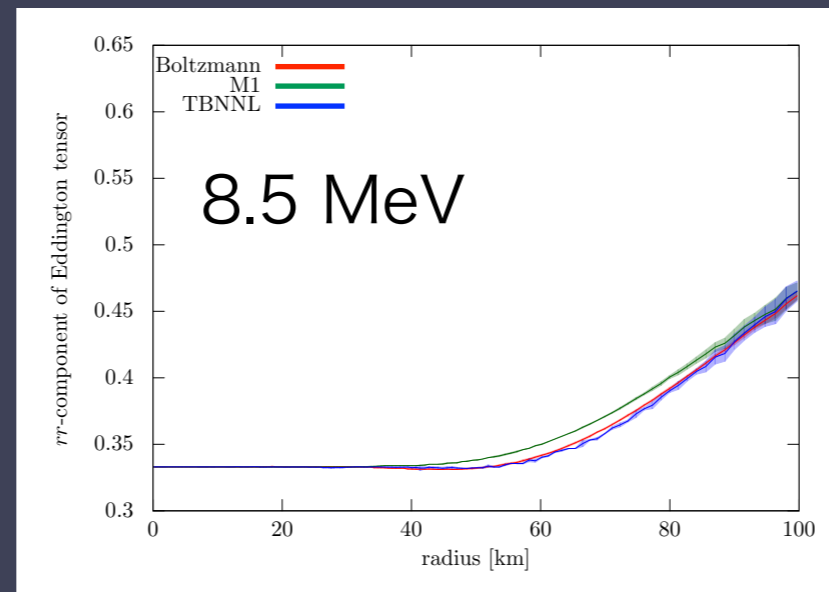
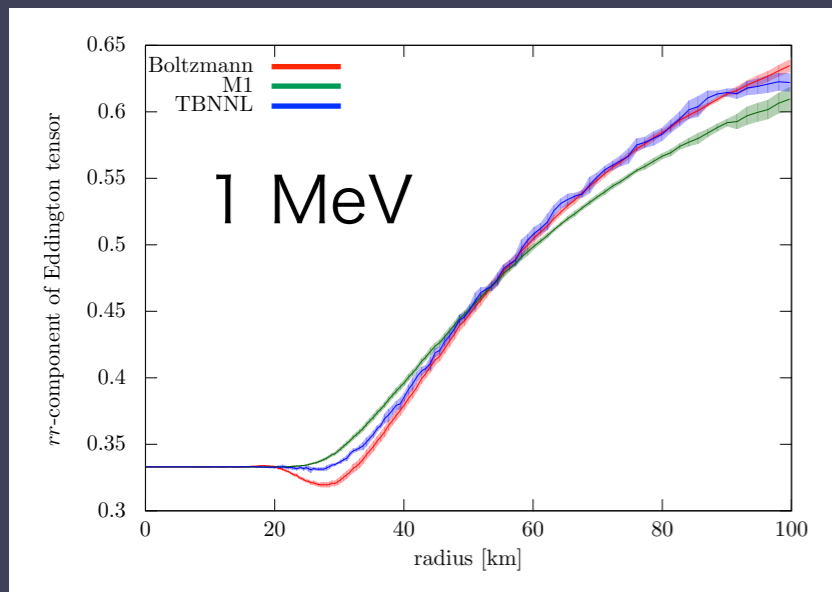
- ほぼEddington tensorを
どれだけきちんと再現で
きるか、みたいなもの



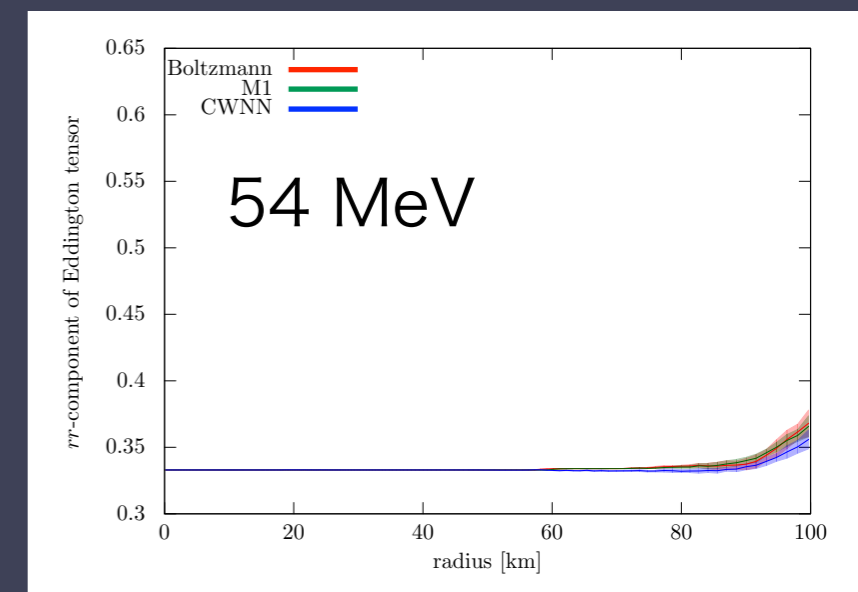
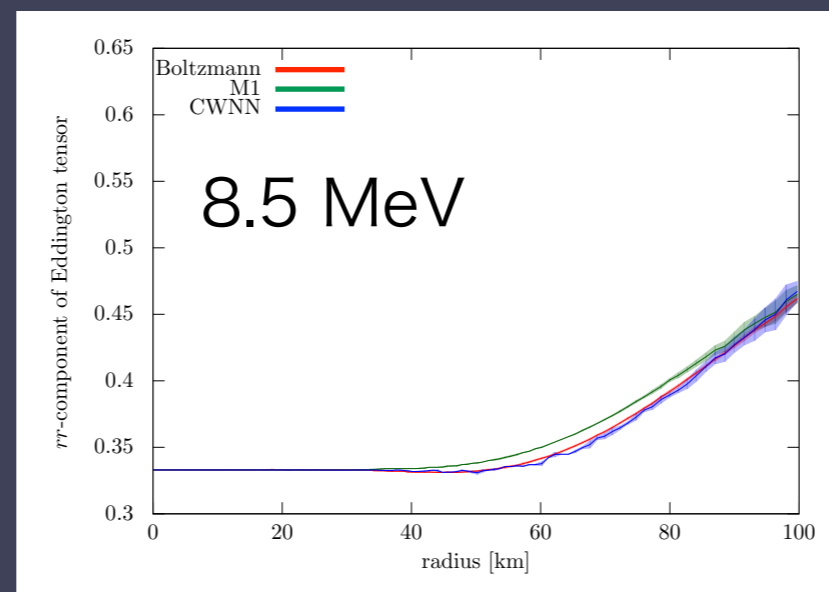
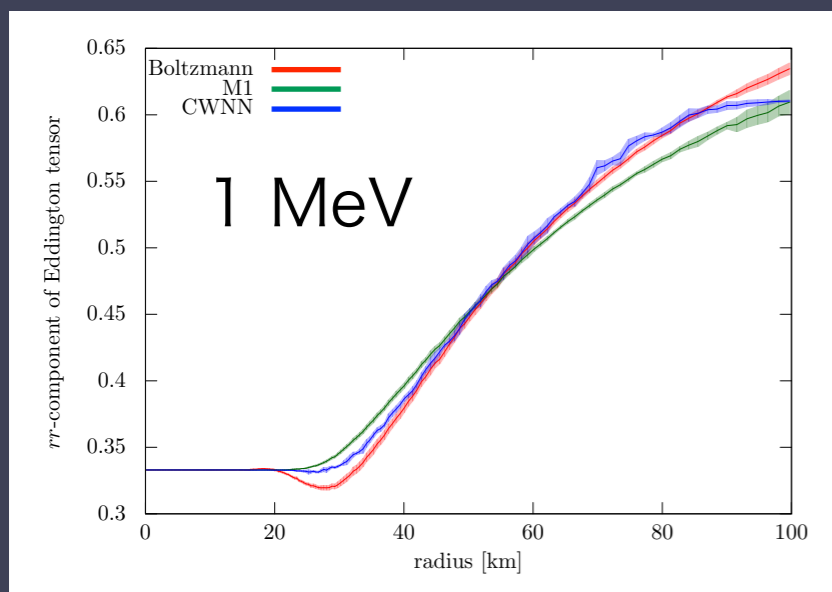
CWNNの結果(非対角成分)



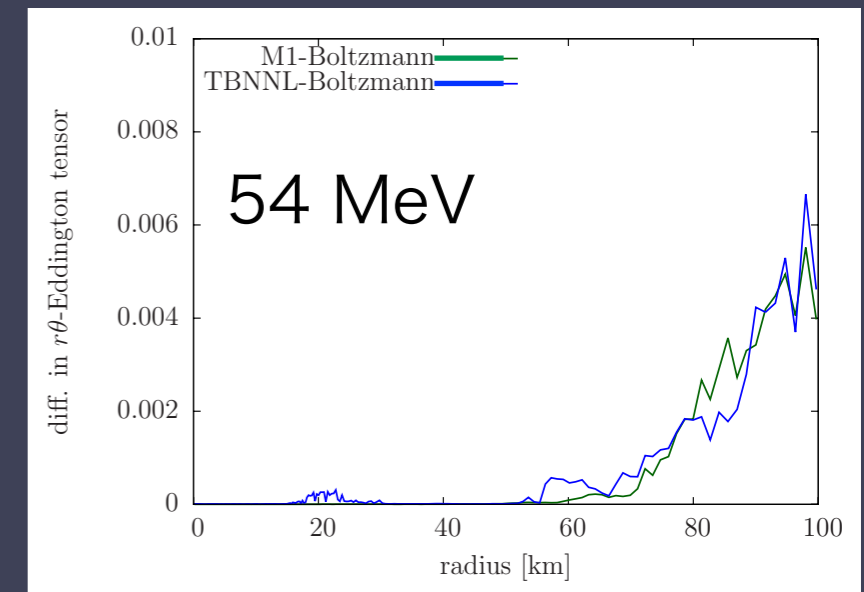
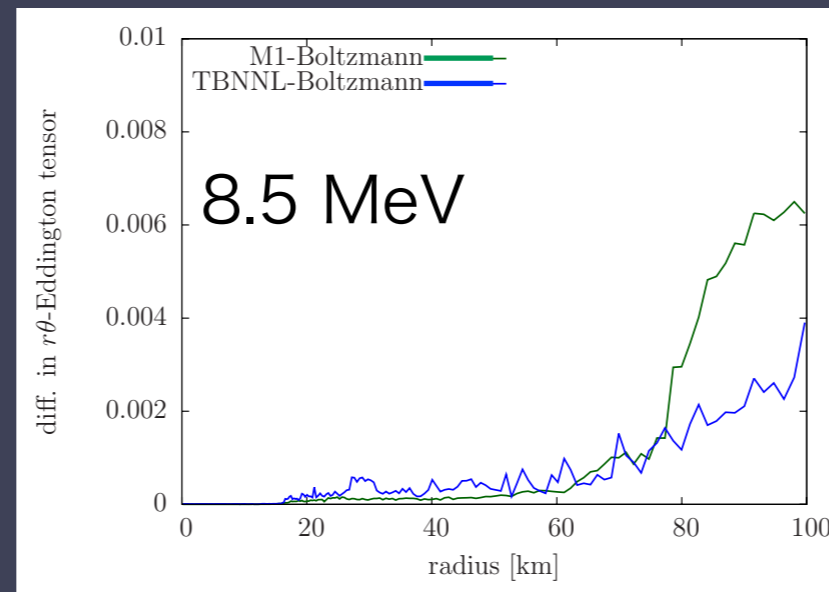
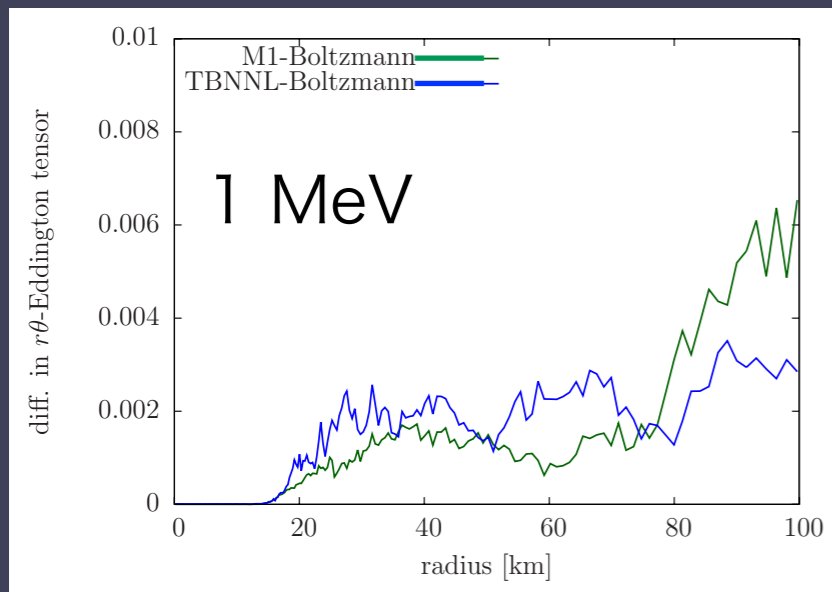
TBNNの結果(対角成分)



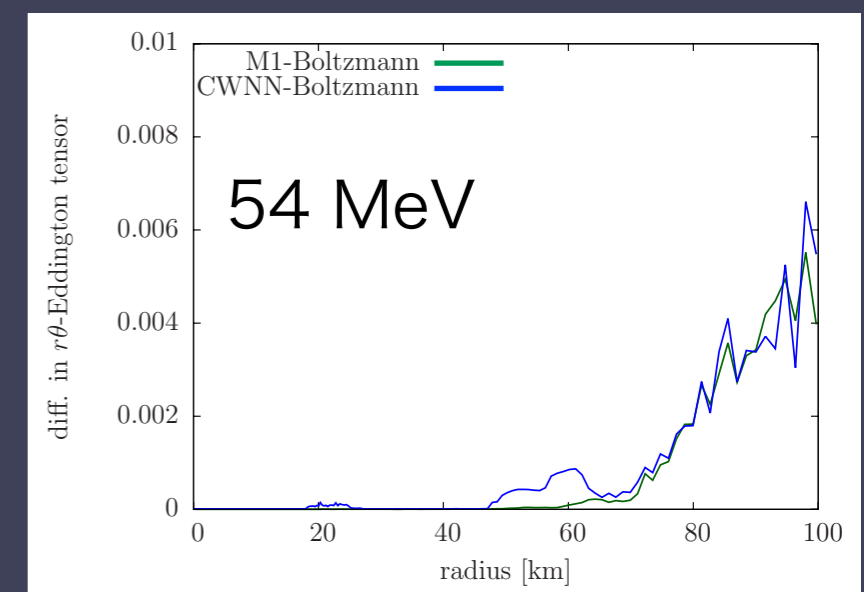
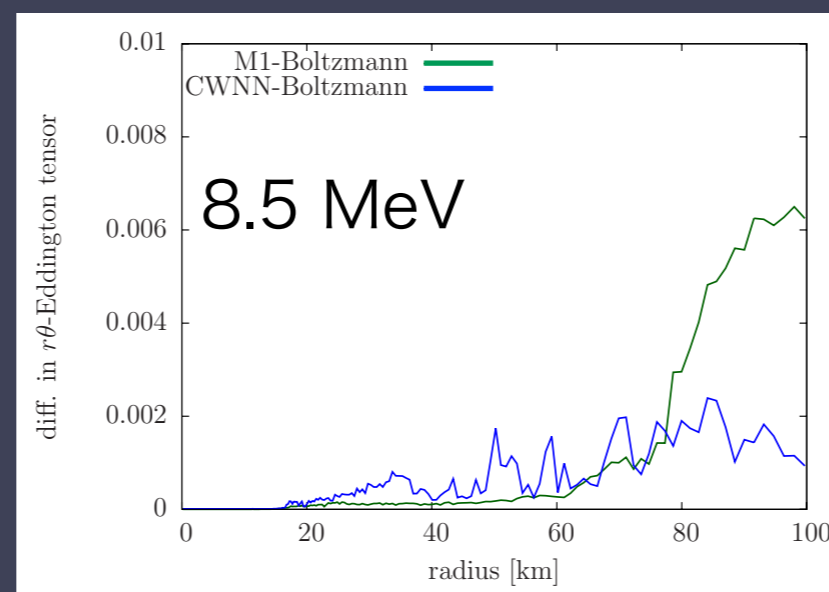
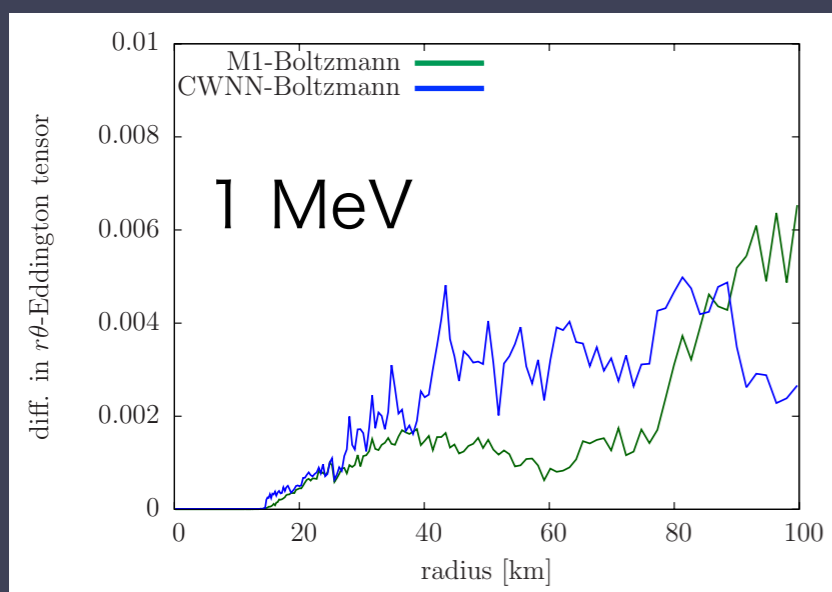
cf. CWNNの結果



TBNNの結果(非対角成分)



cf. CWNNの結果



今後

- ・ input/network構造を改善する
- ・ 学習データのバラエティを増やす
- ・ 学習したclosure関係式を輻射輸送へ応用する
- ・ **on-goingなので、何かアイデアがあればそれもほしい**

まとめ

- ・ 超新星爆発のボルツマン輻射流体シミュレーションの結果をもとに、輻射輸送近似法の精度を上げたい
- ・ どういう関数形にすればいいかわからないので、DNNの関数近似能力を利用する
- ・ 今後はもっと実用的なモデルになるよう改良していきたい

