2024-04-01, *workshop on Dense Matter EoS and Frontiers in NS Physics* (TDLI)

Reverse engineering the TOV equation:

analytical results and applications in deep learning

Shuzhe Shi(施舒哲), Tsinghua University

work in progress, w/ Sophia Han, Kai Zhou, Ronghao Li, Zidu Lin, Lingxiao Wang, + …

see also:

Zhou, Wang, Pang, ***SS*** Prog.Part.Nucl.Phys.104084(2023).

Soma, Wang, ***SS***, Stoecker, Zhou, JCAP08 (2022) 071; Phys. Rev. D **107**, 083028

Neutron Star: EoS <=> mass-radius relation

Tolman-Oppenheimer-Volkov equations:

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r}\,,$$

$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$

$$\varepsilon = \varepsilon(P)\,,$$

Neutron Star: EoS <=> mass-radius relation

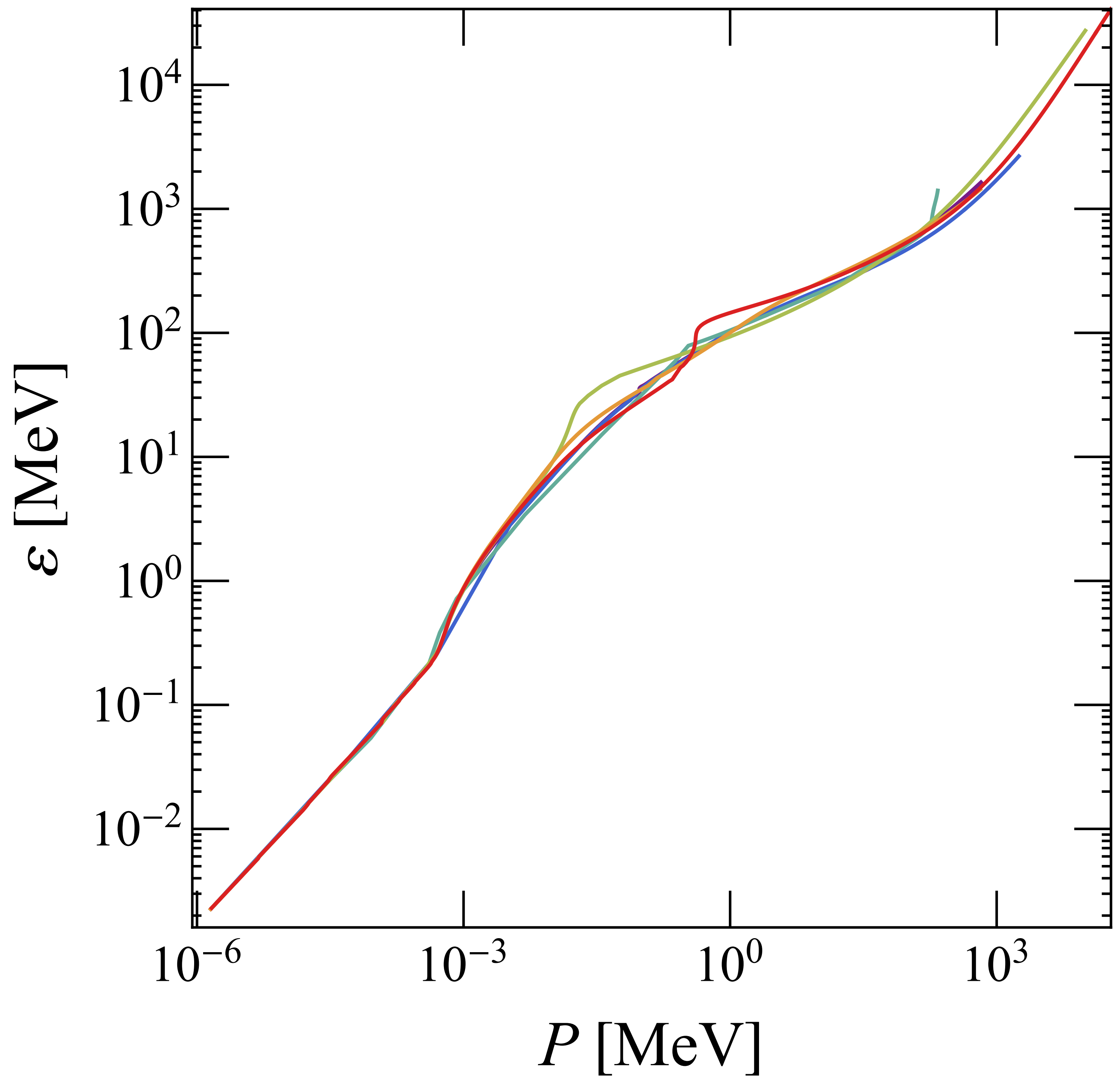
Tolman-Oppenheimer-Volkov equations:

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r}\,,$$

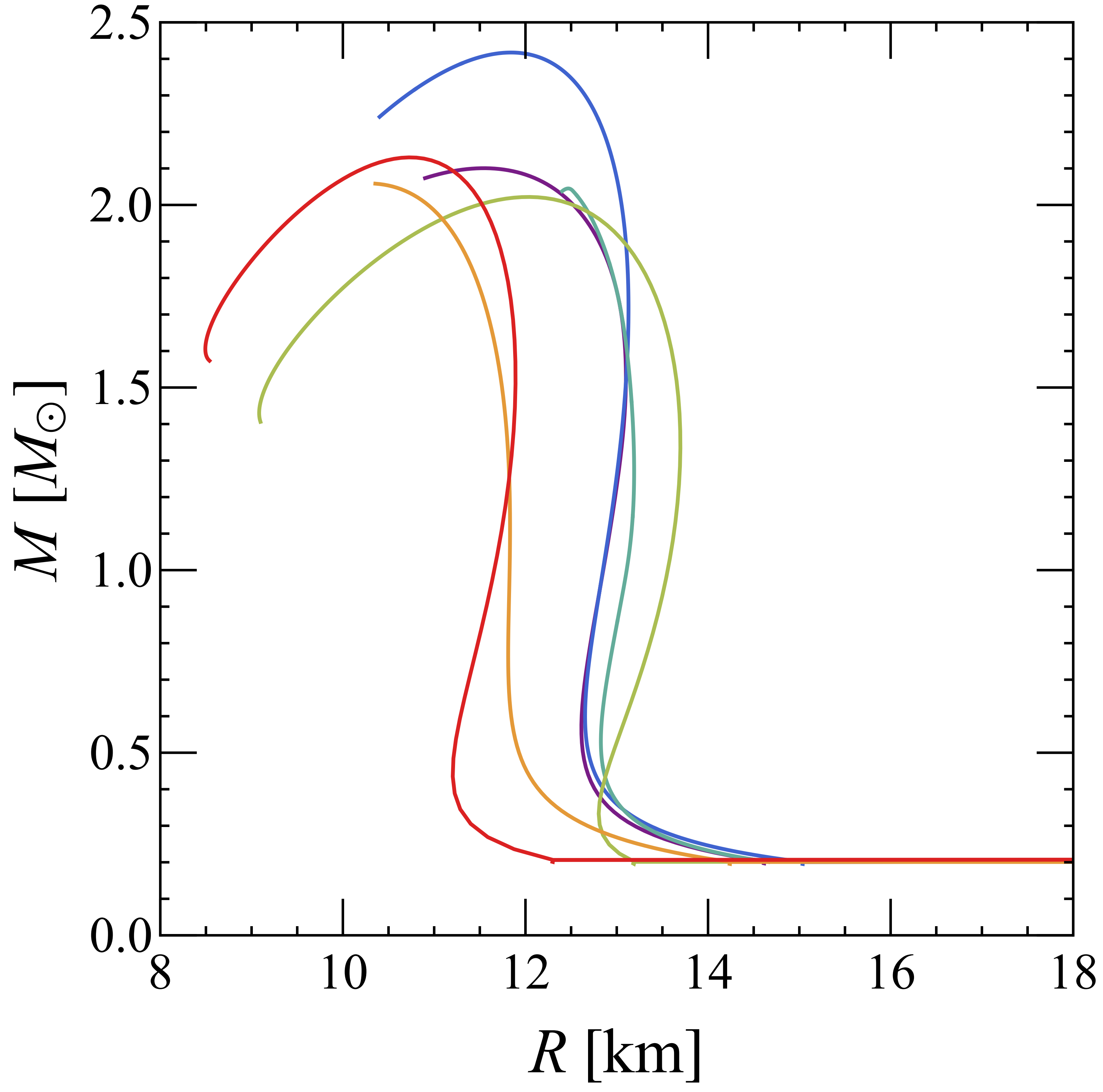
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$

$$\varepsilon = \varepsilon(P)\,,$$

$\varepsilon(P)$



$M(R)$



Neutron Star: EoS <=> mass-radius relation

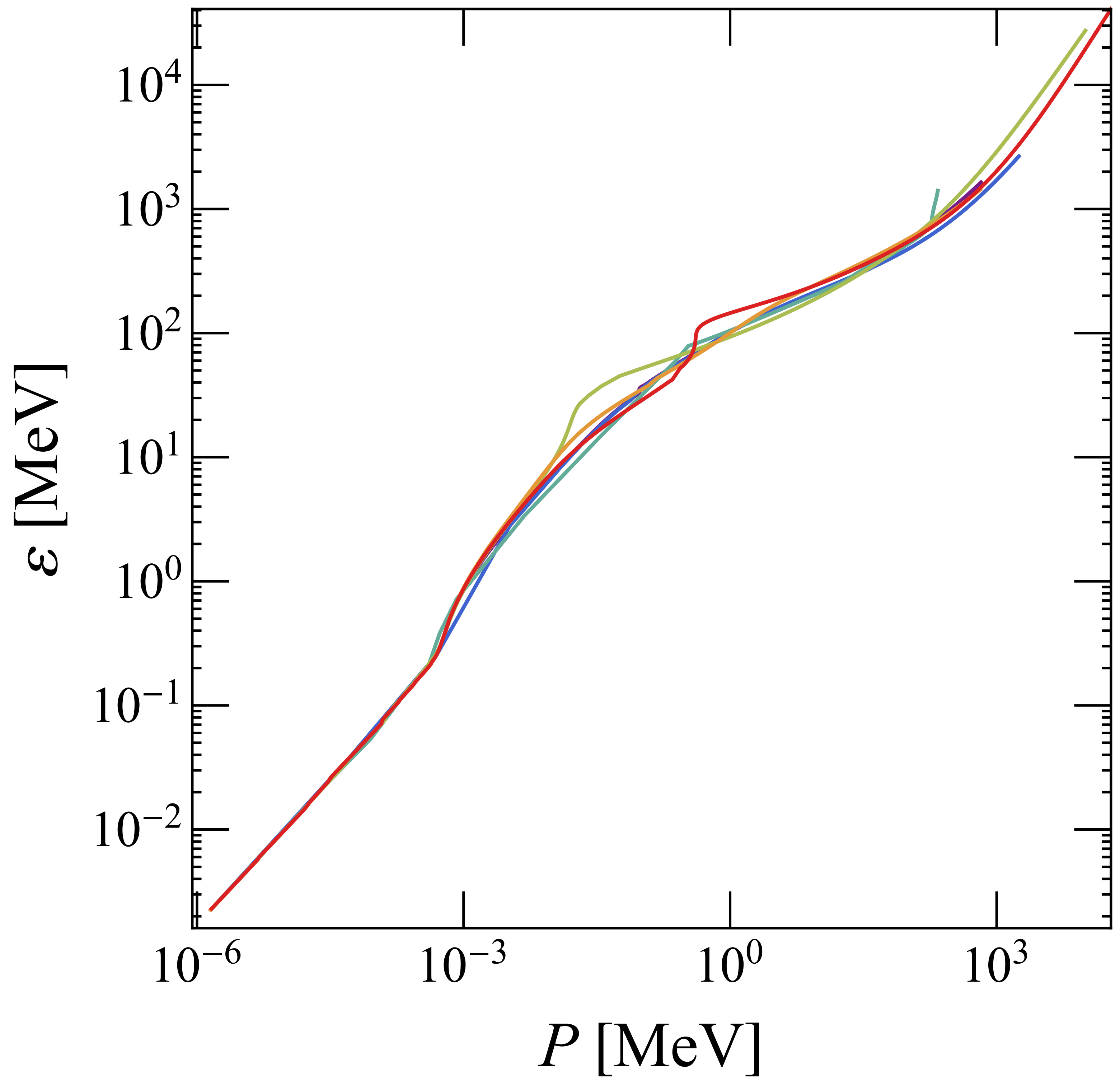
Tolman-Oppenheimer-Volkov equations:

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r}\,,$$

$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$

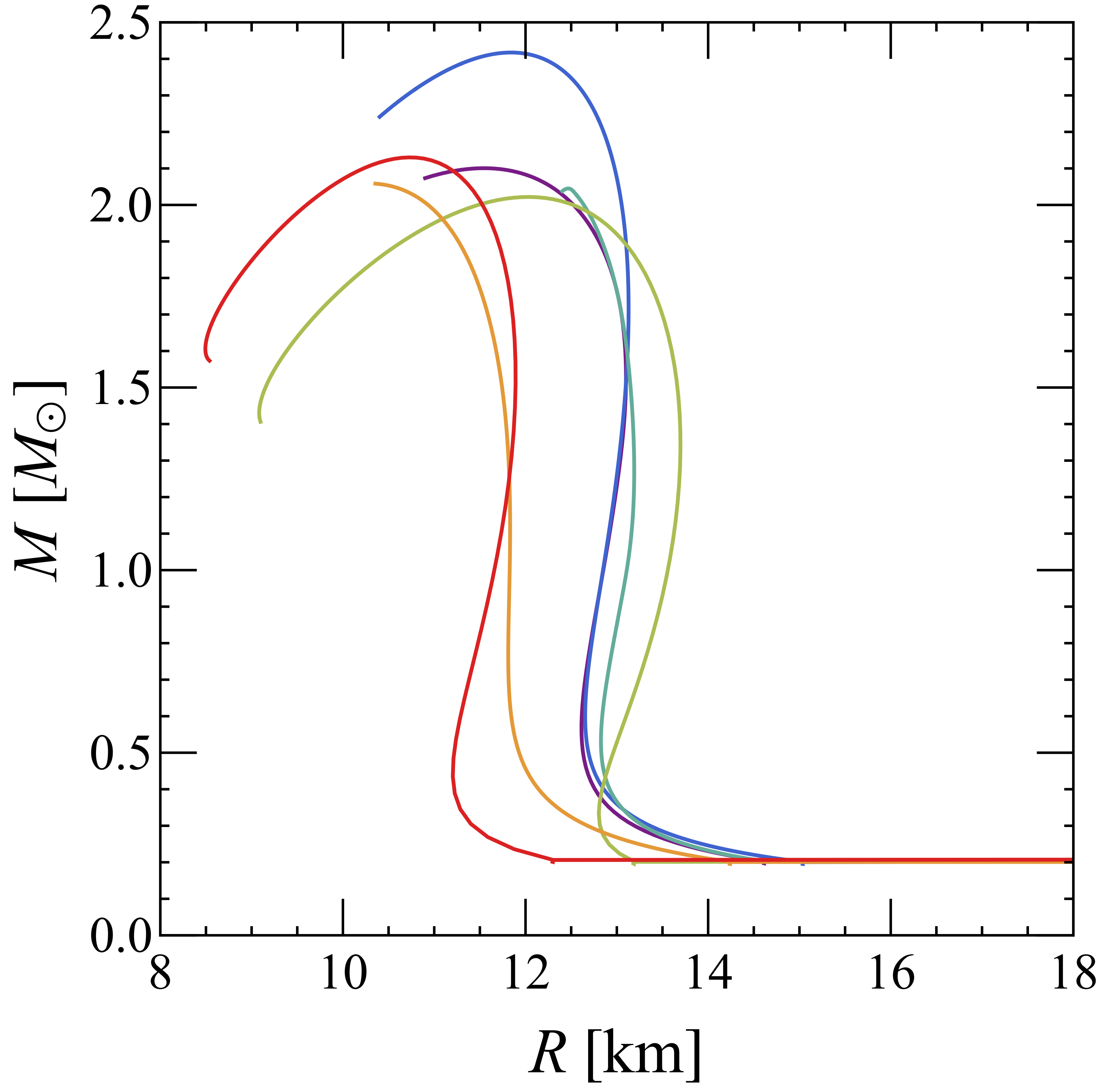
$$\varepsilon = \varepsilon(P)\,,$$

$\varepsilon(P)$



?

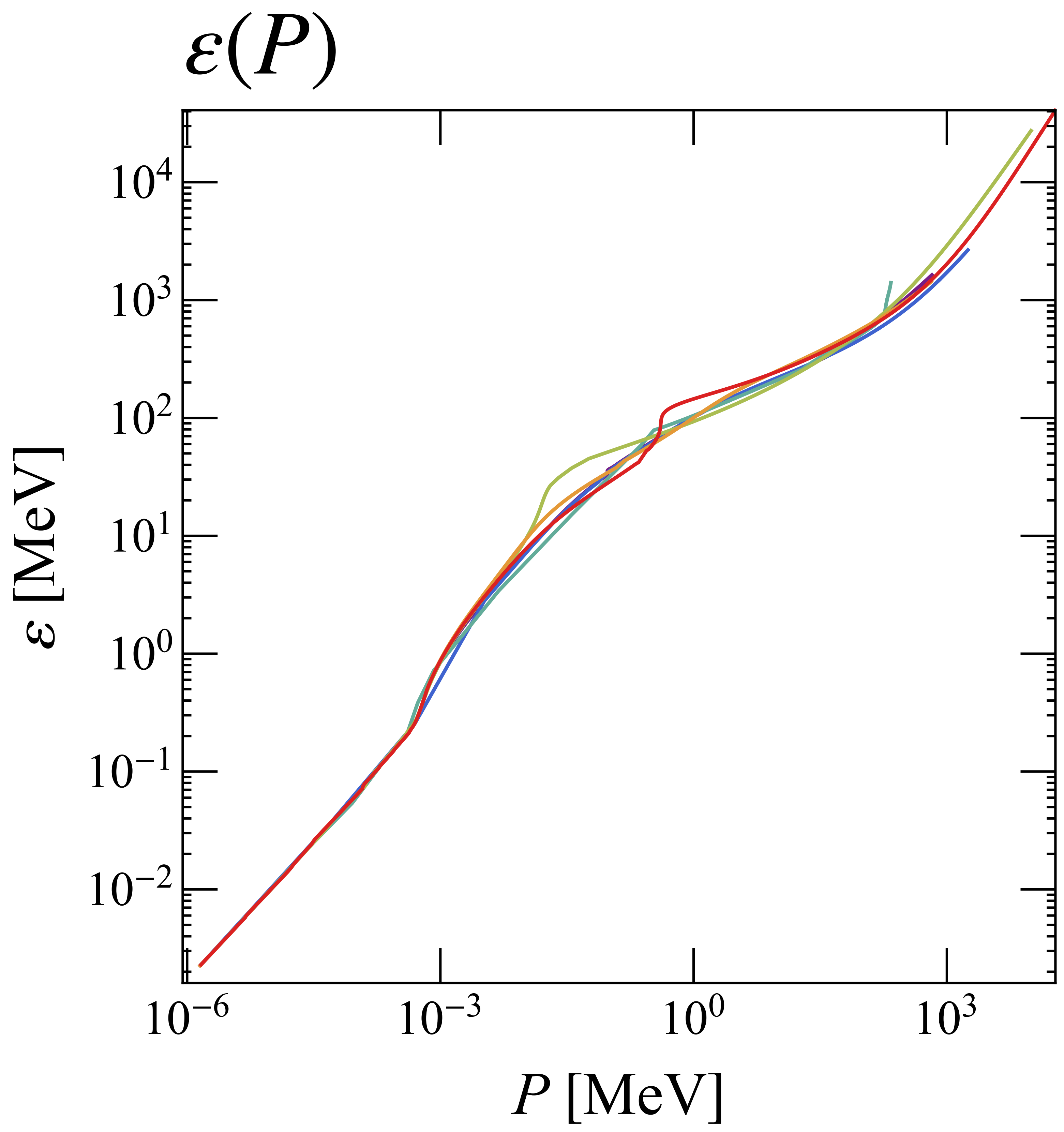
$M(R)$



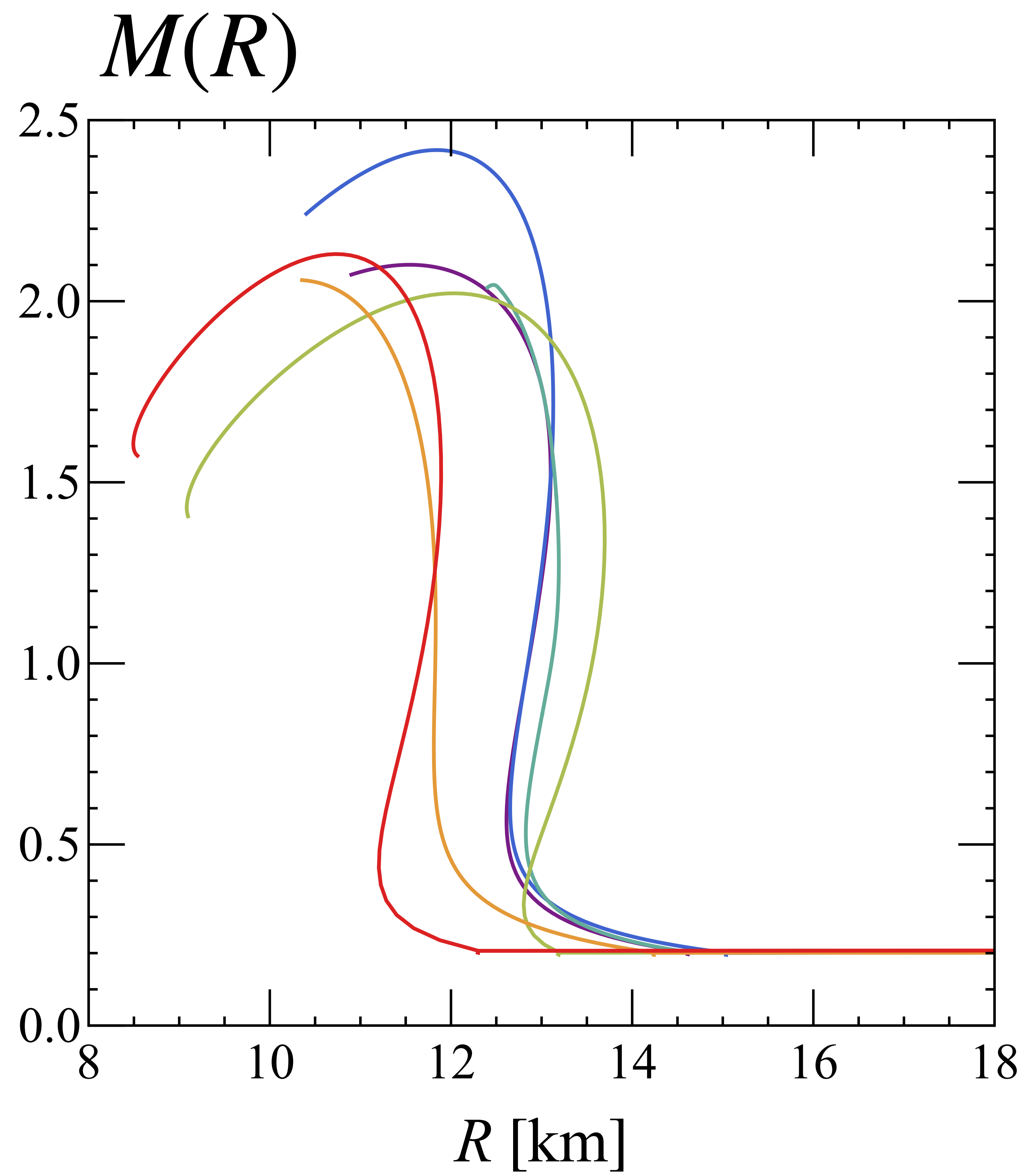
Neutron Star: EoS <=> mass-radius relation

Tolman-Oppenheimer-Volkov equations:

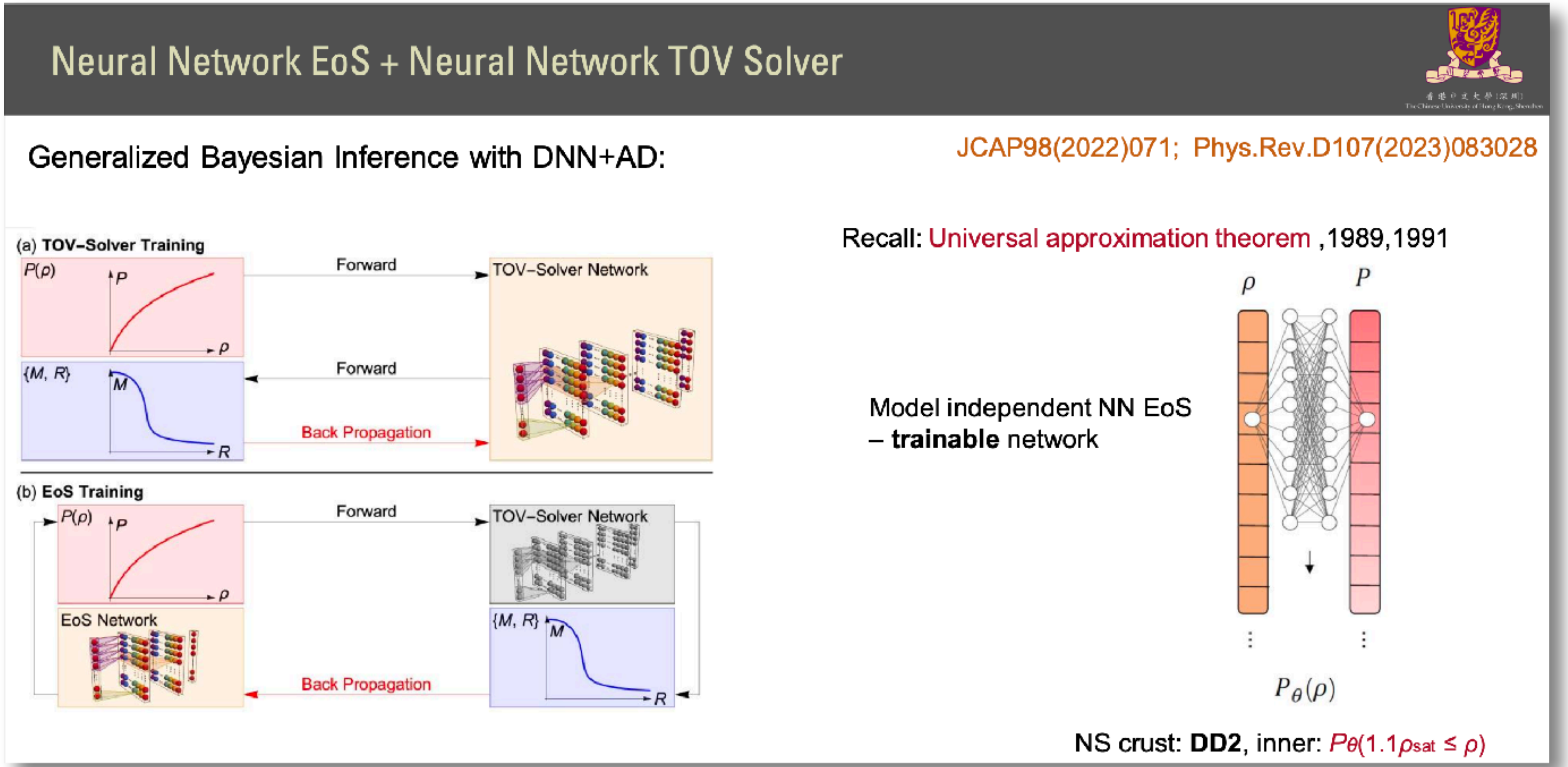
- Bayesian analysis;
- supervised deep learning;
- unsupervised DL;



?



unsupervised learning

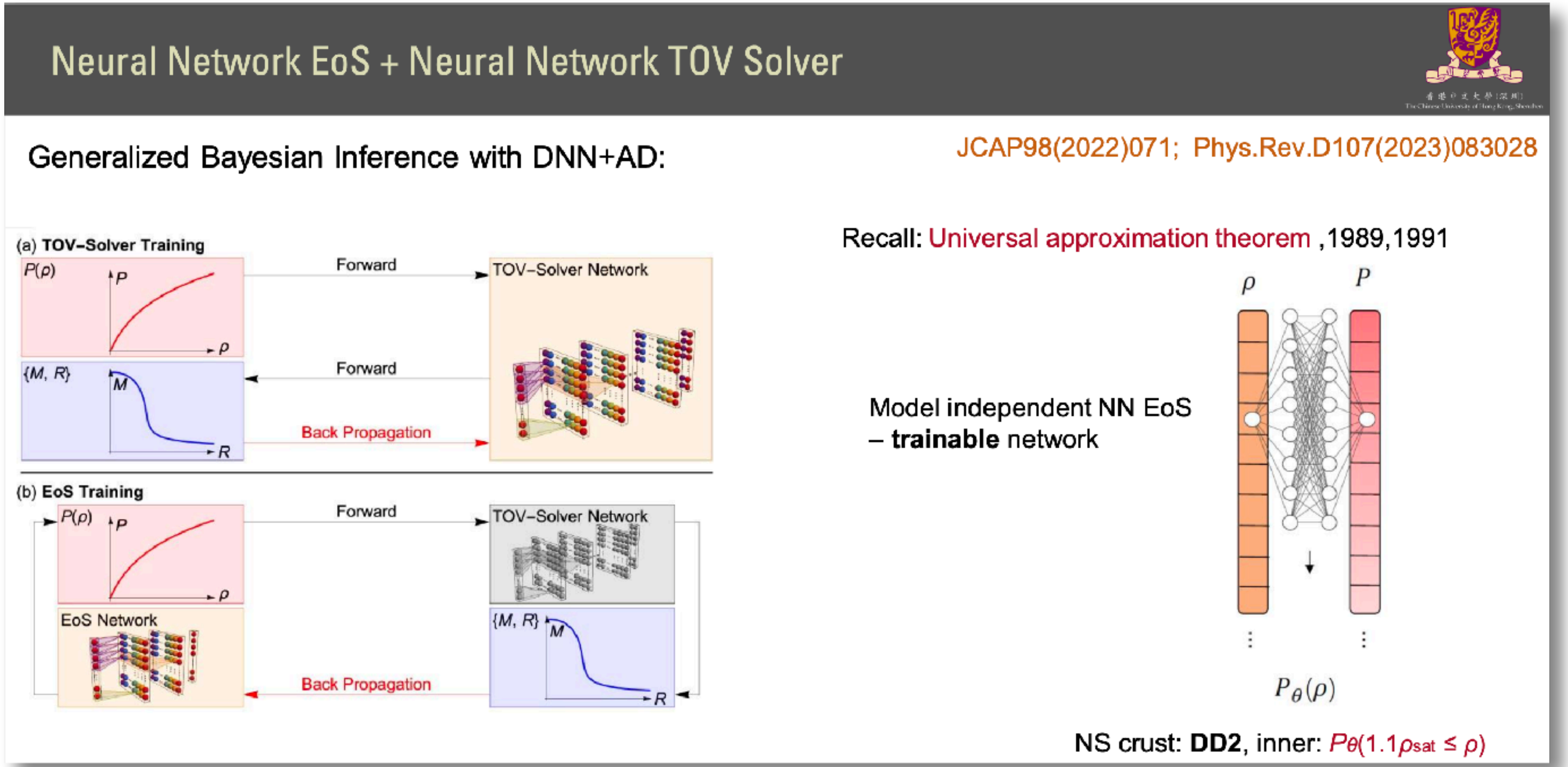


Kai Zhou's talk on Mar. 28

two neural networks:

1. represents EoS
2. approximates TOV solver

unsupervised learning

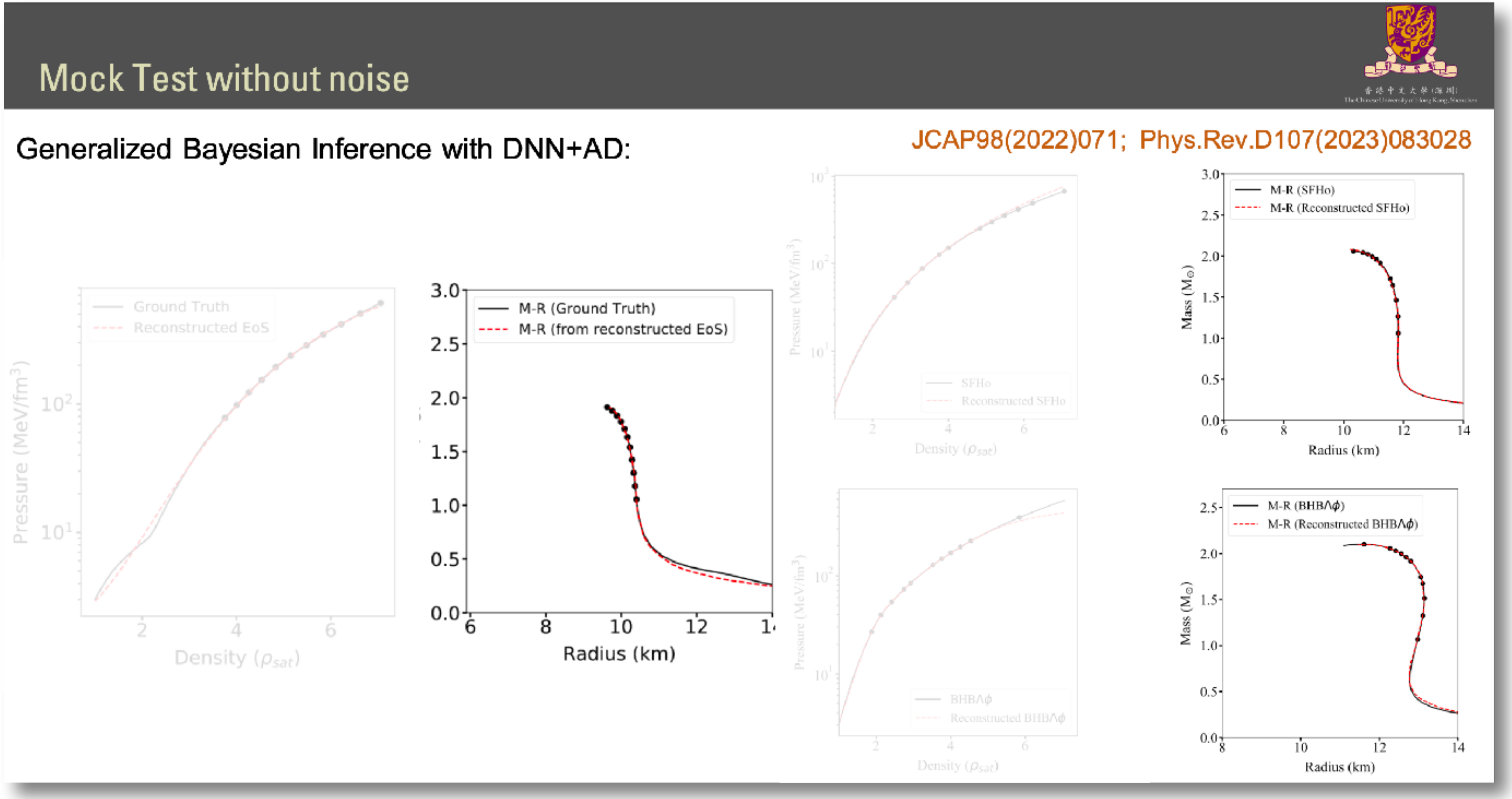


Kai Zhou's talk on Mar. 28

two neural networks: piecewise (nonlinear) interpolation → general, unbiased

1. represents EoS
2. approximates TOV solver ← reverse engineering w/ auto differentiation

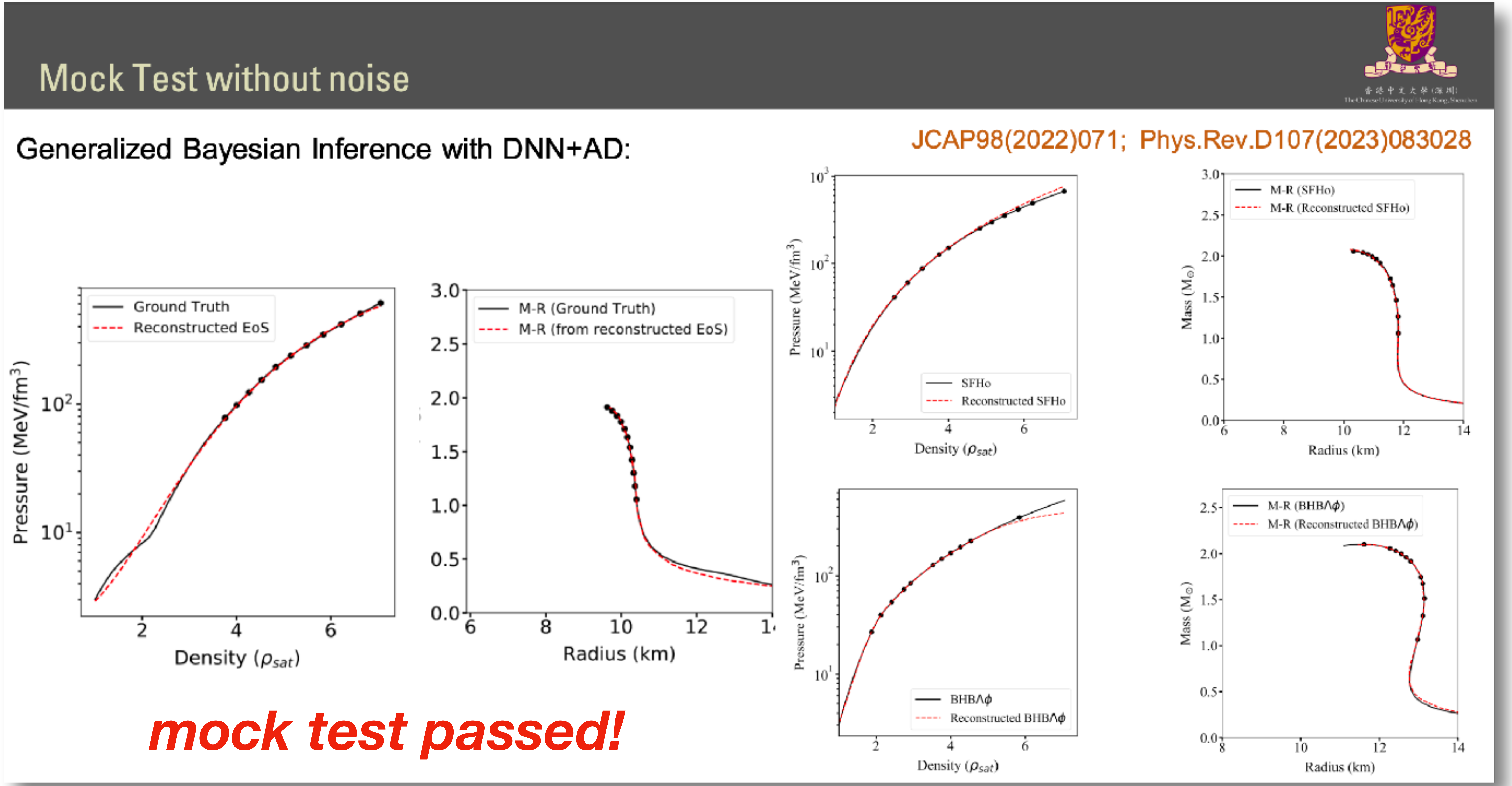
unsupervised learning



Kai Zhou's talk on Mar. 28

reverse engineering w/ auto differentiation

unsupervised learning



Kai Zhou's
talk on Mar. 28

reverse engineering w/ auto differentiation

What is behind the “magic”?

What is behind the "magic"?

input:
EoS

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r}\,,$$

$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$

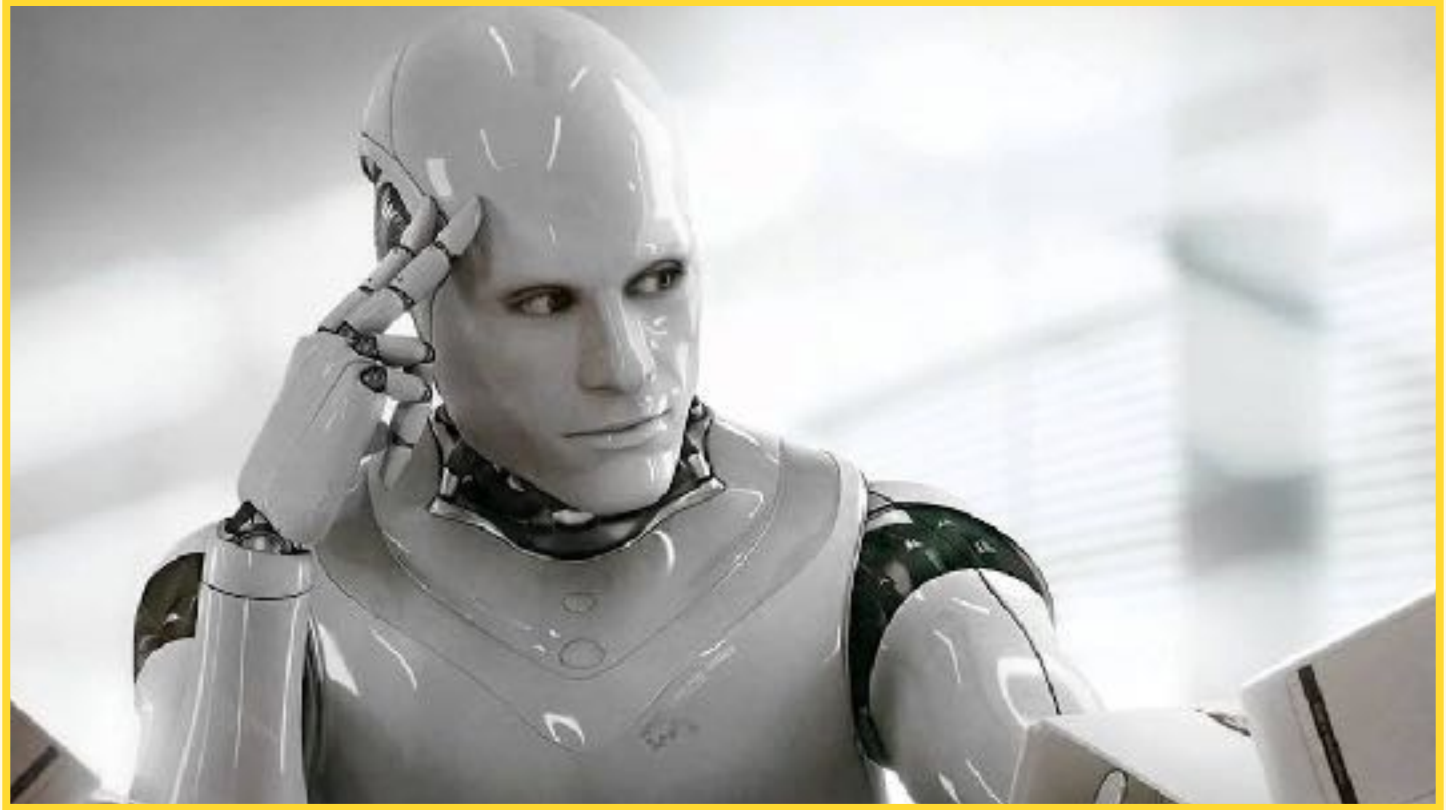
$$\varepsilon = \varepsilon(P)\,,$$

output:
M-*R* curve

What is behind the “magic”?

TOV solver network

input:
EoS

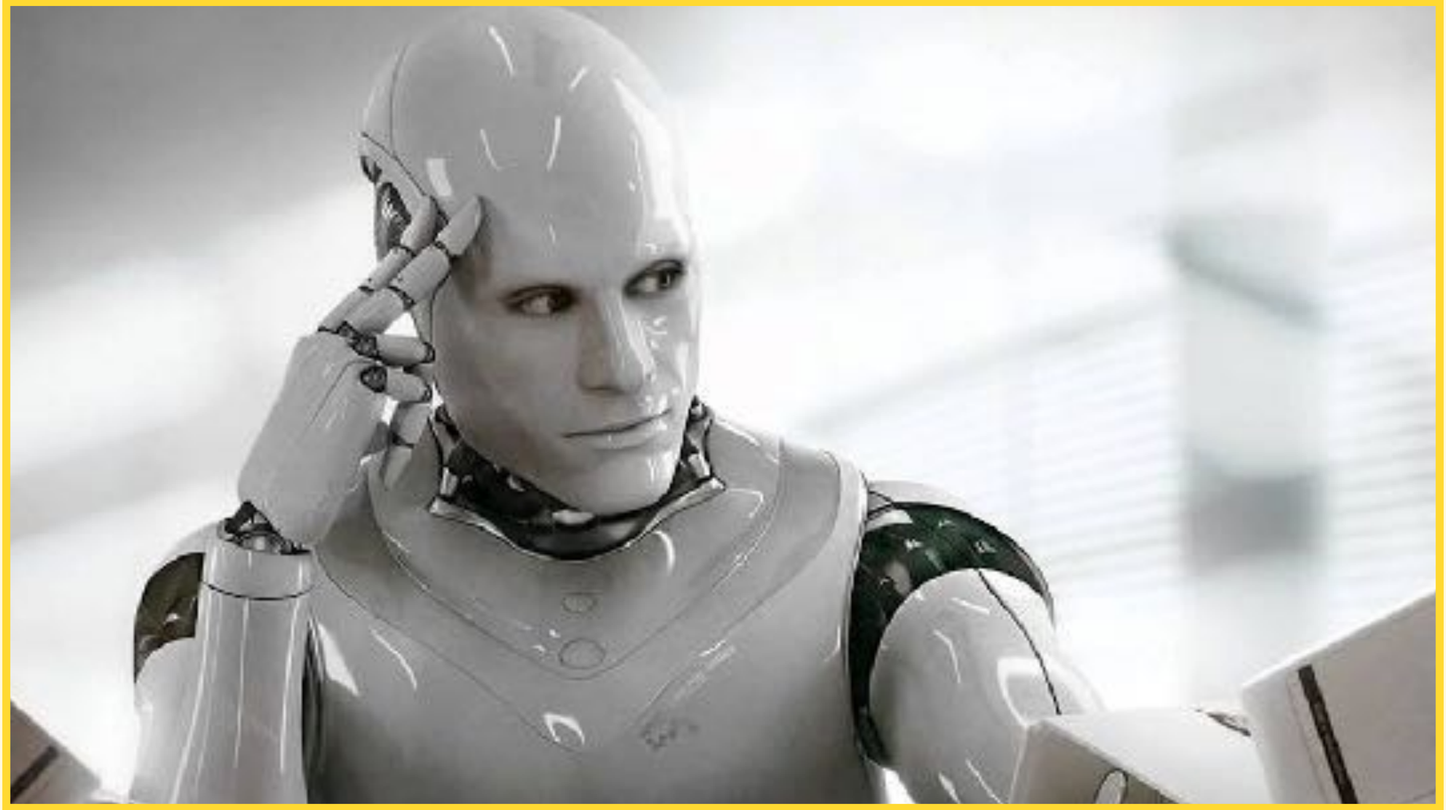


output:
M-*R* curve

What is behind the "magic"?

TOV solver network

input:
EoS



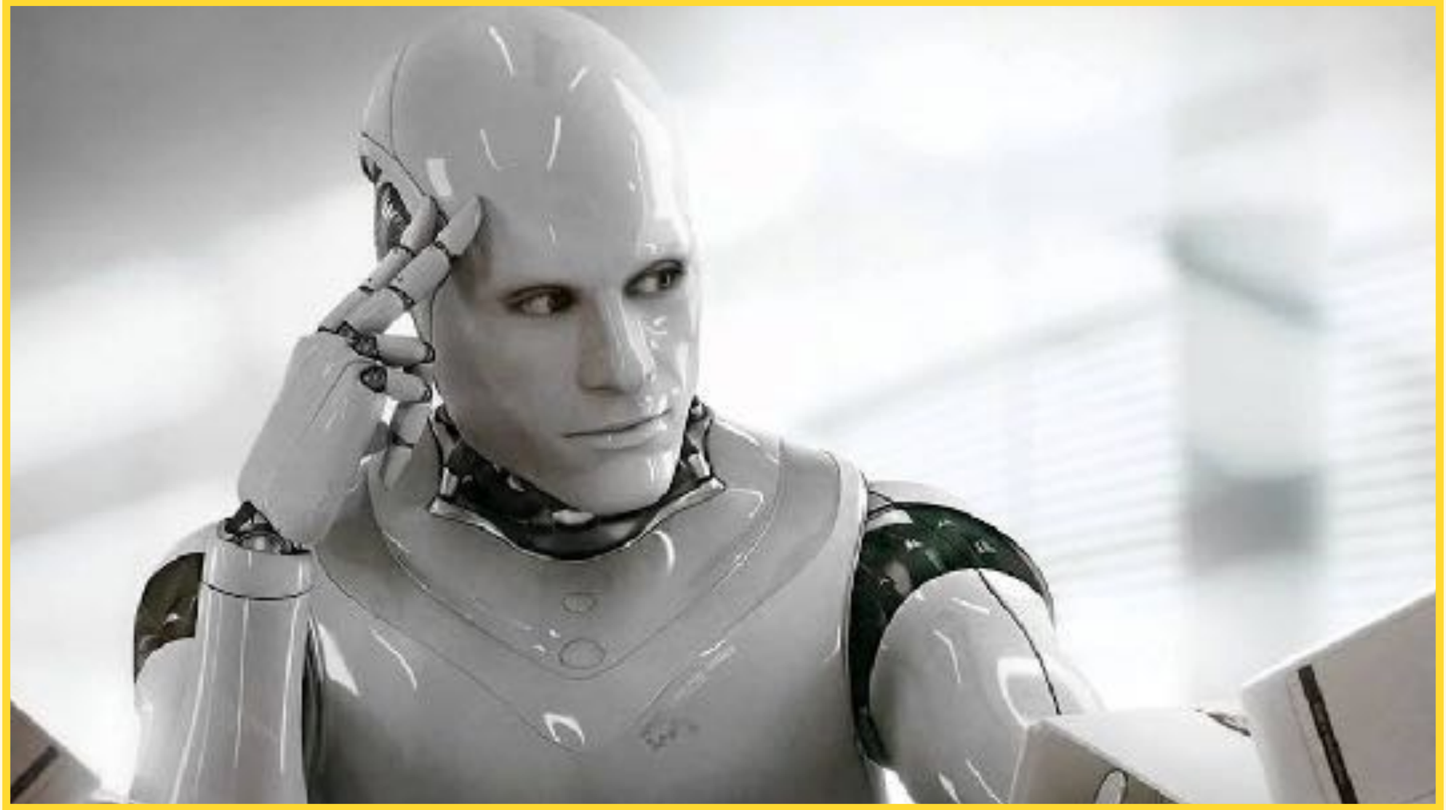
output:
M-*R* curve

auto differentiation: $\dfrac{\delta\,(M\text{-}R)}{\delta\,(\mathrm{EoS})}$

What is behind the "magic"?

TOV solver network

input:
EoS



output:
M-*R* curve

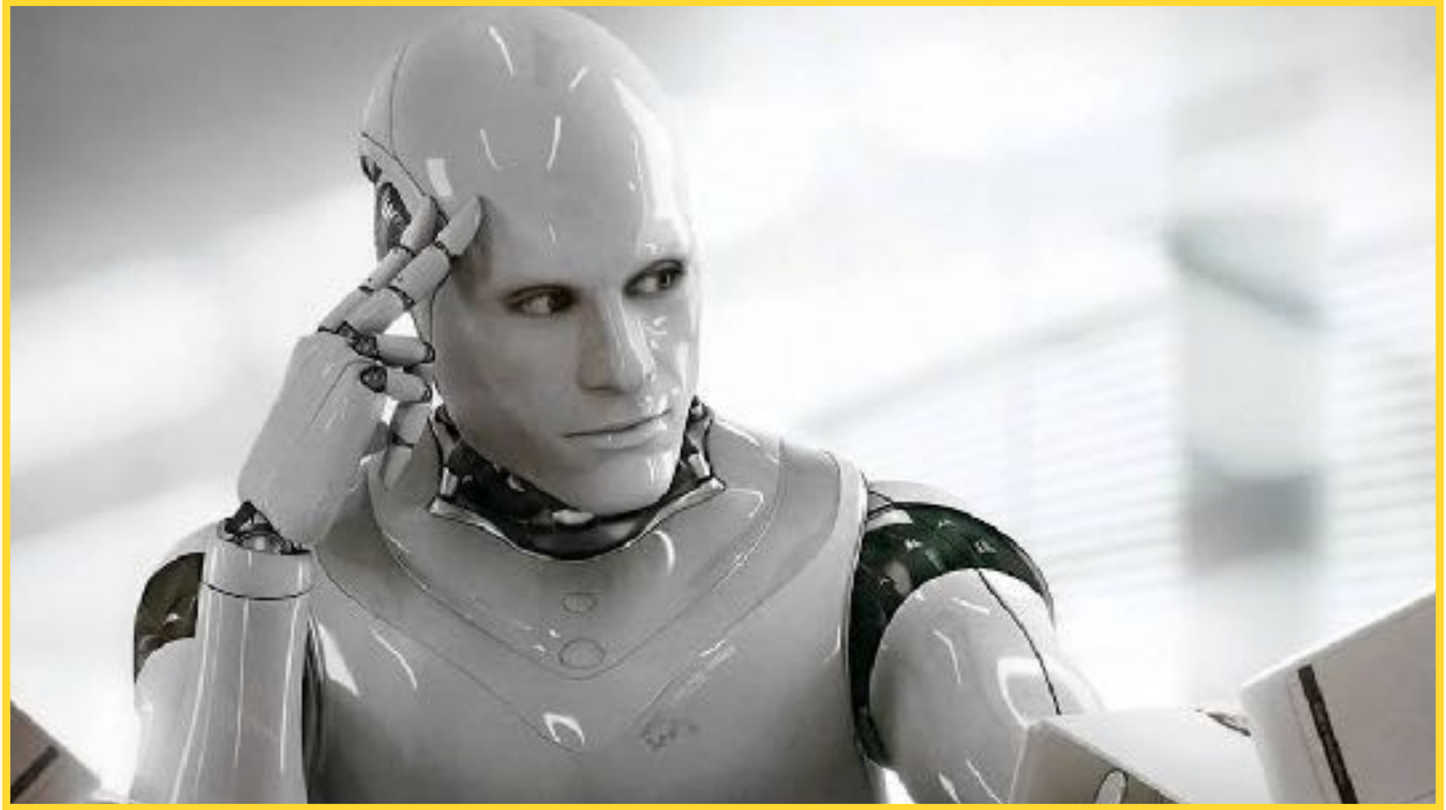
auto differentiation: $\dfrac{\delta\,(M\text{-}R)}{\delta\,(\mathrm{EoS})}$

$(M\text{-}R)_{\text{desired}} - (M\text{-}R)_{\text{curr.}}$

What is behind the "magic"?

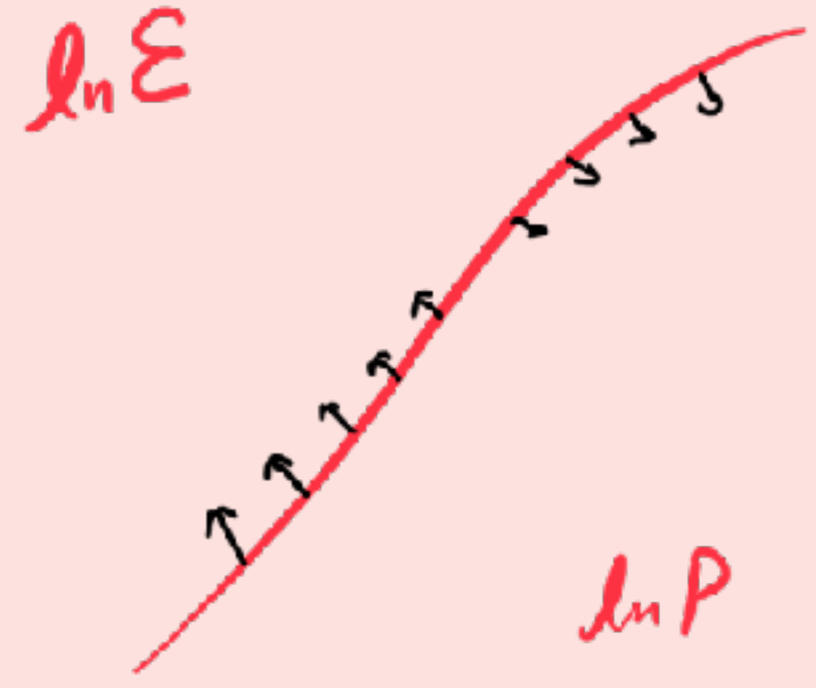
TOV solver network

input:
EoS



output:
M-R curve

auto differentiation: $\dfrac{\delta\,(M\text{-}R)}{\delta\,(\mathrm{EoS})}$



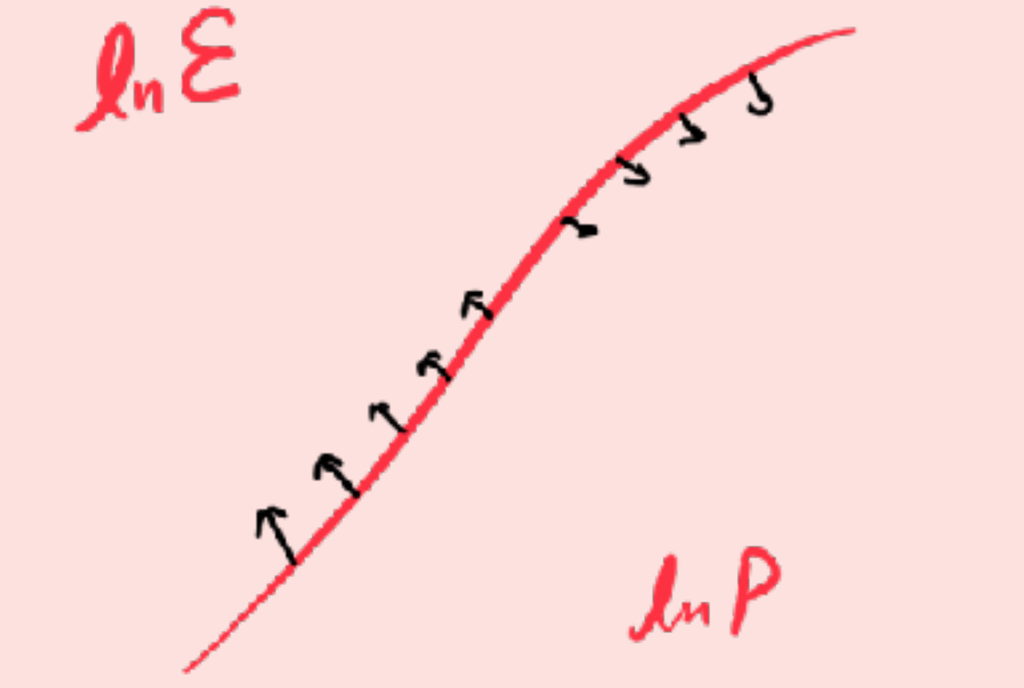
change in EoS

$(M\text{-}R)_{\mathrm{desired}} - (M\text{-}R)_{\mathrm{curr.}}$

mathematics behind the "magic"

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{(m + 4\pi r^3 P)(P+\varepsilon)}{r^2 - 2\,m\,r}\,,$$
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$
$$\varepsilon = \varepsilon(P)\,,$$

auto differentiation: $\dfrac{\delta\,(M\text{-}R)}{\delta\,(\mathrm{EoS})}$



change in EoS

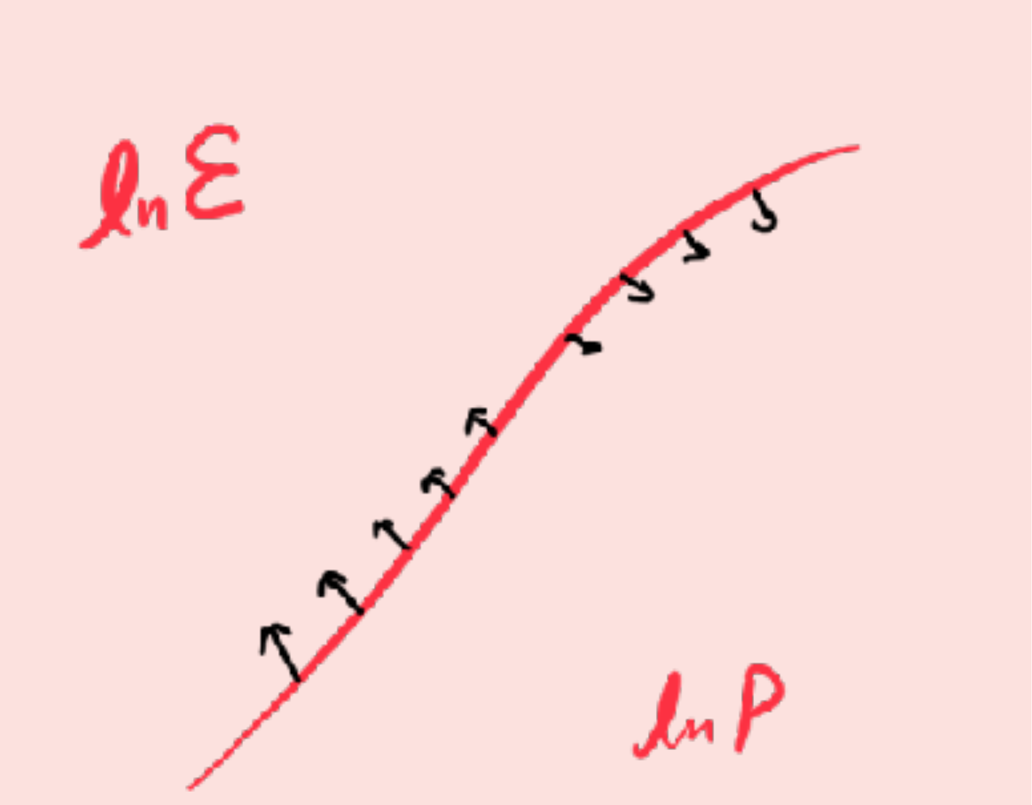
$(M\text{-}R)_{\text{desired}} - (M\text{-}R)_{\text{curr.}}$

mathematics behind the "magic"

"manual" differentiation:
linear response analysis
of the TOV equation

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r}\,,$$
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$
$$\varepsilon = \varepsilon(P)\,,$$

auto differentiation: $\dfrac{\delta\,(M\text{-}R)}{\delta\,(\mathrm{EoS})}$



change in EoS

$(M\text{-}R)_{\text{desired}} - (M\text{-}R)_{\text{curr.}}$

mathematics behind the "magic"

"manual" differentiation: linear response analysis of the TOV equation

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r}\,,$$
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$
$$\varepsilon = \varepsilon(P)\,,$$

central pressure

radius

$$P_c = \int_0^R \frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r}\mathrm{d}r\,,$$

$$M = 4\pi \int_0^R r^2\,\varepsilon\,\mathrm{d}r\,,$$

mass

mathematics behind the "magic"

"manual" differentiation:
linear response analysis
of the TOV equation

central pressure

radius

$$P_c = \int_0^R \frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r} \mathrm{d}r\,,$$

$$M = 4\pi \int_0^R r^2\,\varepsilon\,\mathrm{d}r\,,$$

mass

$$\varepsilon(P) \to \varepsilon(P) + \delta\varepsilon\,\delta(P - P')$$

$$R \to R + \delta R,\ \ M \to M + \delta M$$

mathematics behind the "magic"

"manual" differentiation:
linear response analysis
of the TOV equation

$\frac{\delta R(P_c)}{\delta \varepsilon(P')}$ *and* $\frac{\delta M(P_c)}{\delta \varepsilon(P')}$ *obtained by solving differential equations together with TOV.*

see Eq. (5.18) of Prog.Part.Nucl.Phys.104084(2023).

central pressure

radius

$$P_c = \int_0^R \frac{(m + 4\pi r^3 P)(P + \varepsilon)}{r^2 - 2\,m\,r} \mathrm{d}r\,,$$

$$M = 4\pi \int_0^R r^2\,\varepsilon\,\mathrm{d}r\,,$$

mass

$$\varepsilon(P) \to \varepsilon(P) + \delta\varepsilon\,\delta(P - P')$$

$$R \to R + \delta R,\ \ M \to M + \delta M$$

mathematics behind the “magic”

“manual” differentiation:
linear response analysis
of the TOV equation

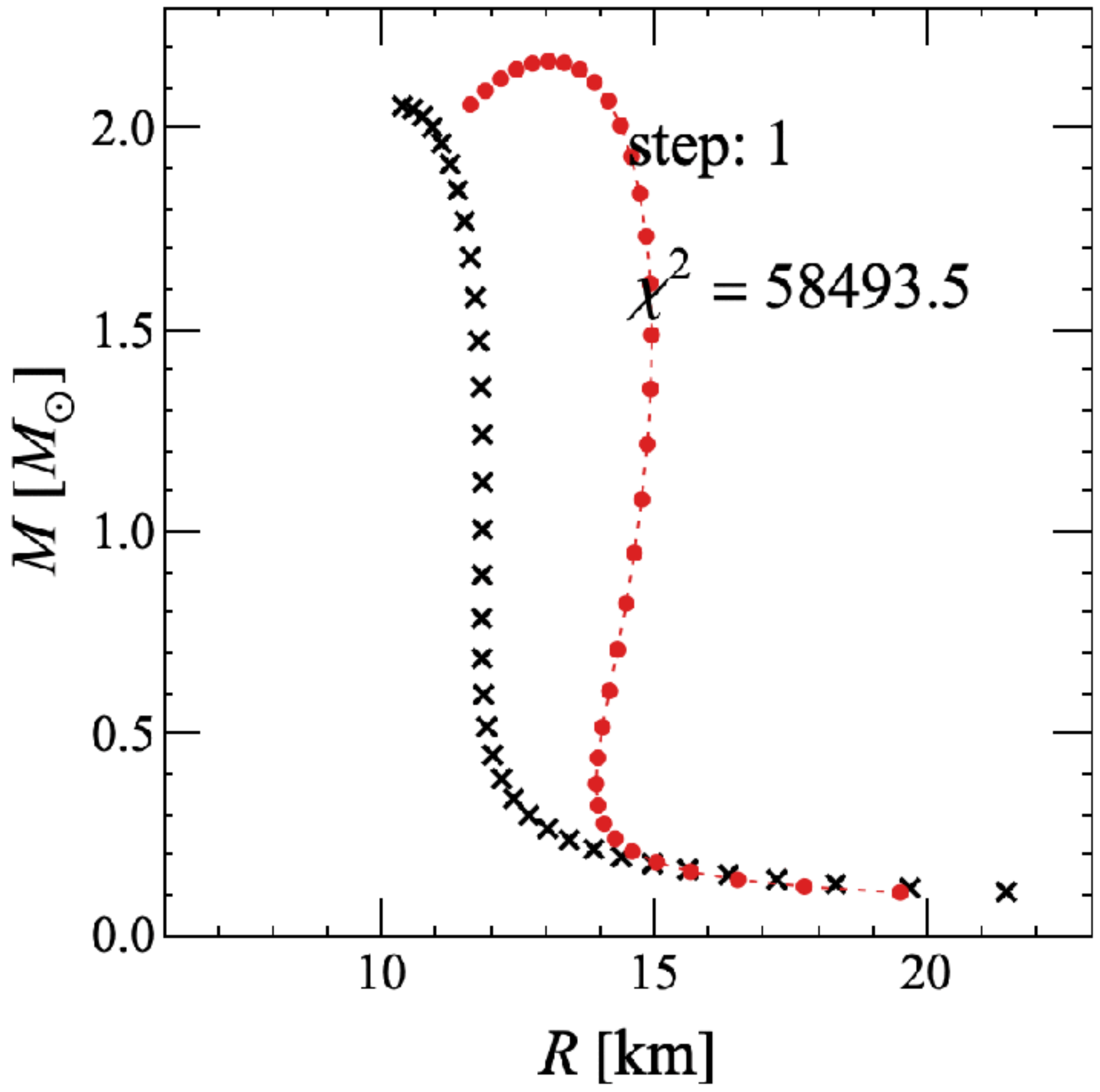
$\frac{\delta R(P_c)}{\delta \varepsilon(P')}$ and $\frac{\delta M(P_c)}{\delta \varepsilon(P')}$ obtained by solving differential equations together with TOV.

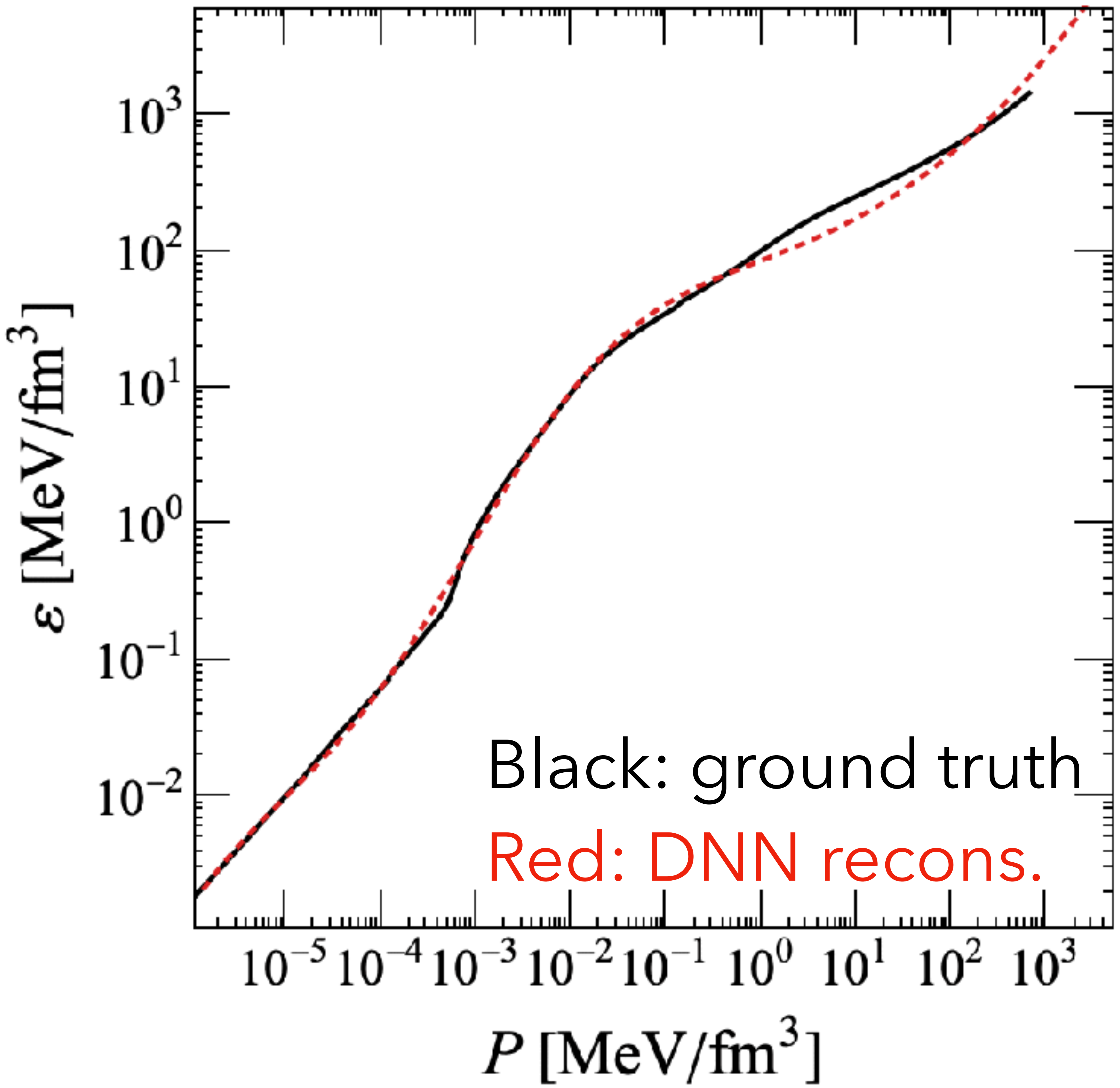
see Eq. (5.18) of Prog.Part.Nucl.Phys.104084(2023).

- Applicable to any parameterization of EoS;

- Used DNN in our work.

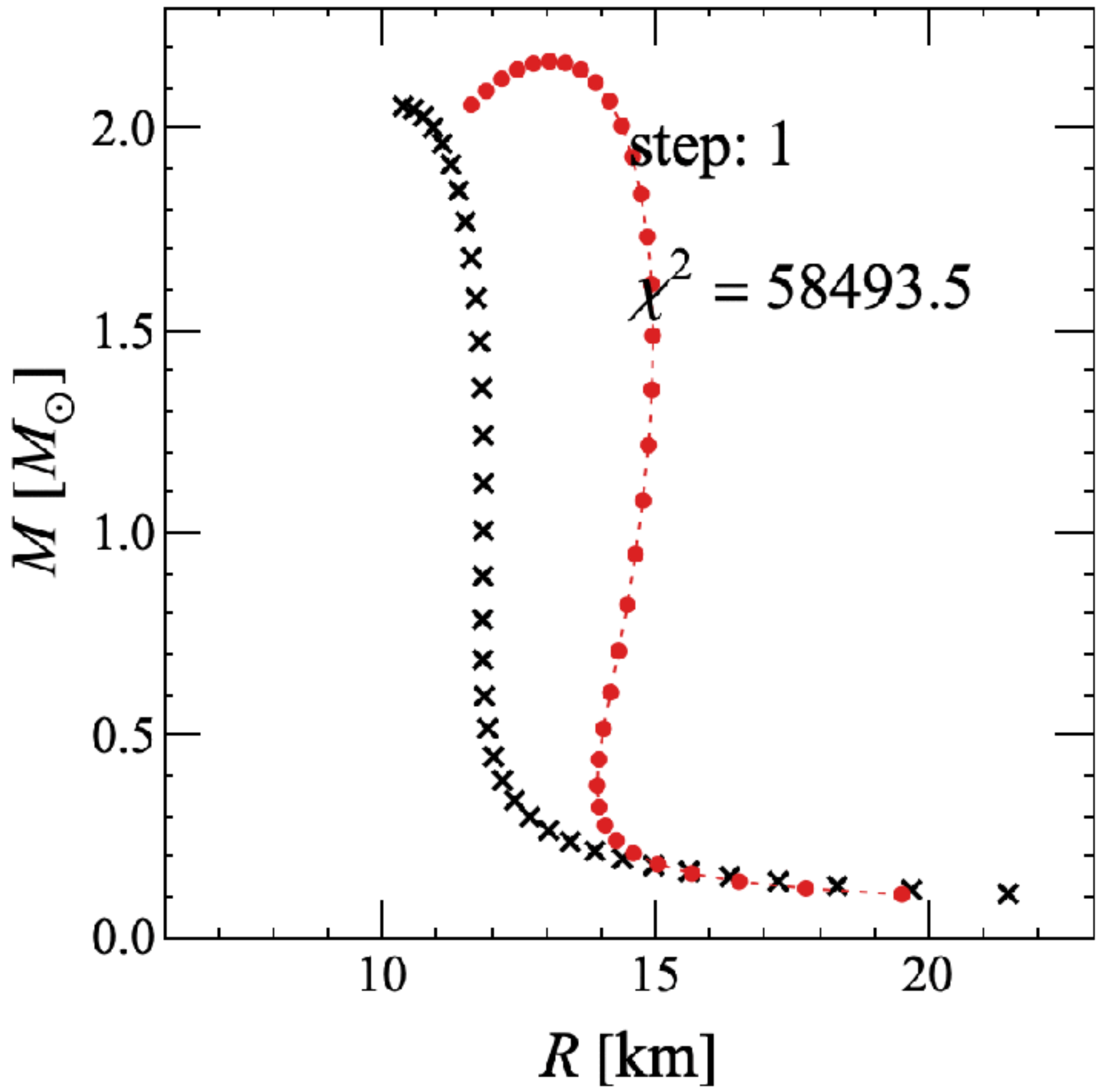
closure test

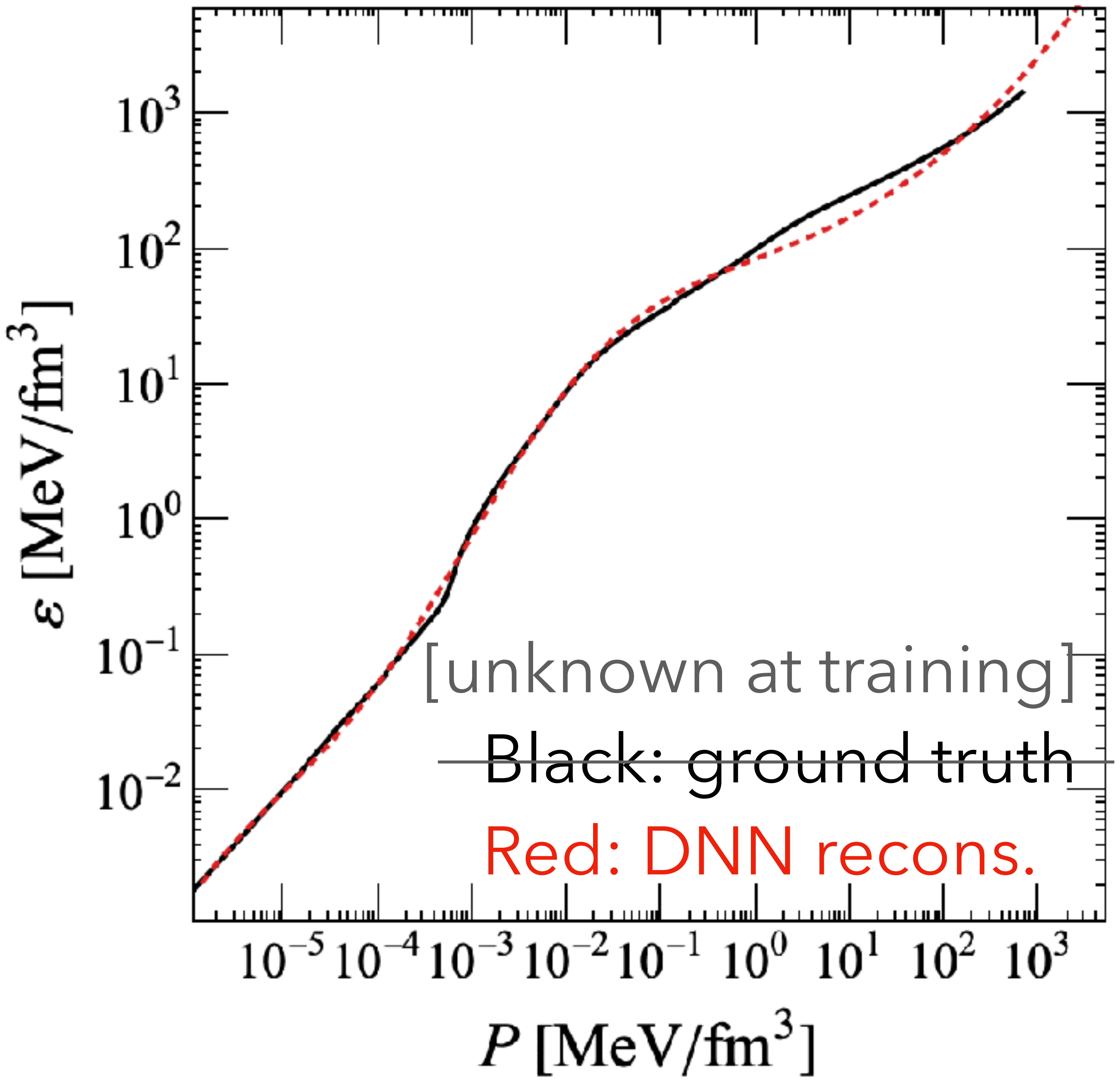




$$\chi^2 = \sum_i \left(\frac{m_i - m_i^{\mathrm{obs}}}{\Delta m_i}\right)^2 + \left(\frac{R_i - R_i^{\mathrm{obs}}}{\Delta R_i}\right)^2$$

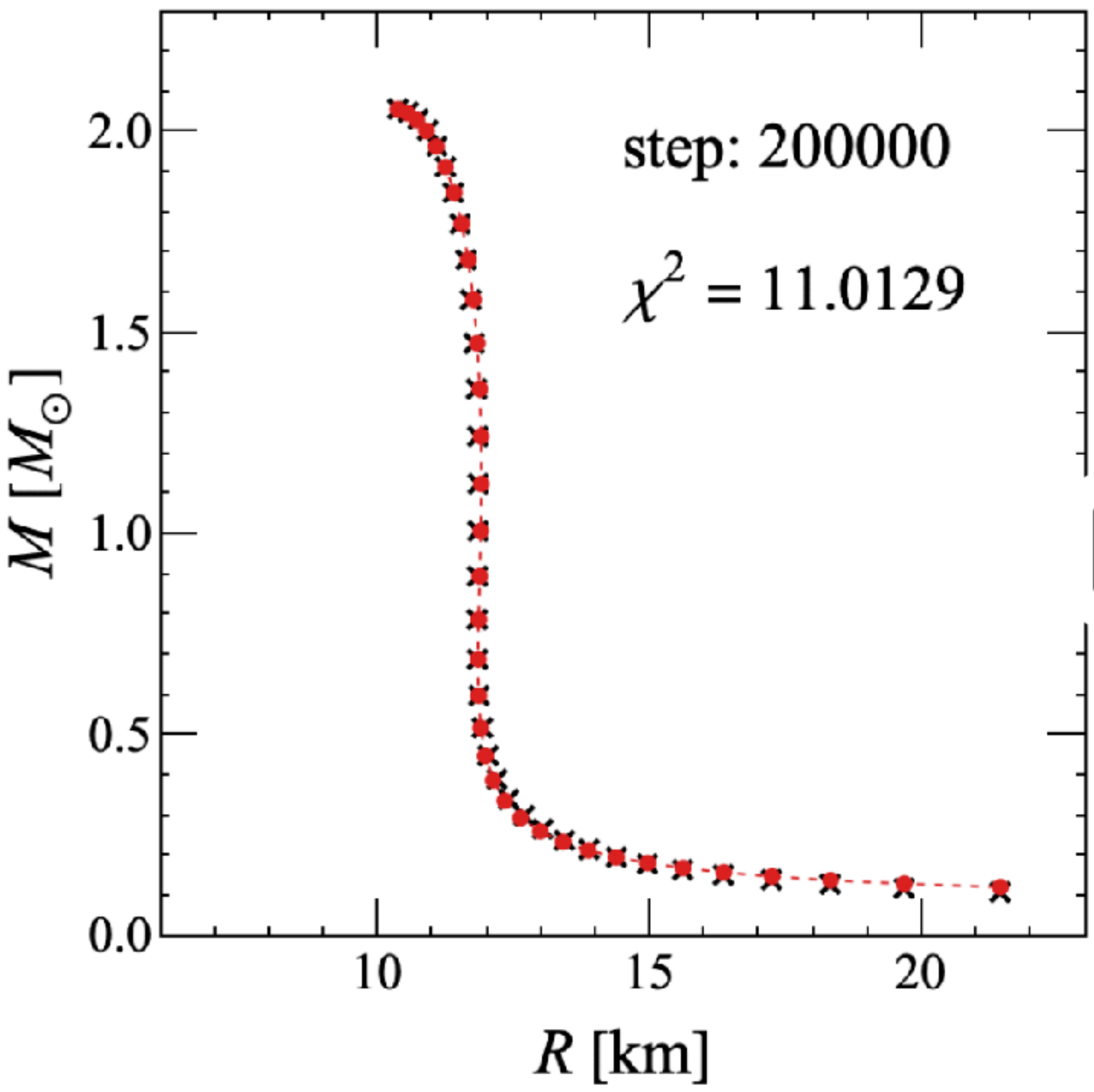
closure test

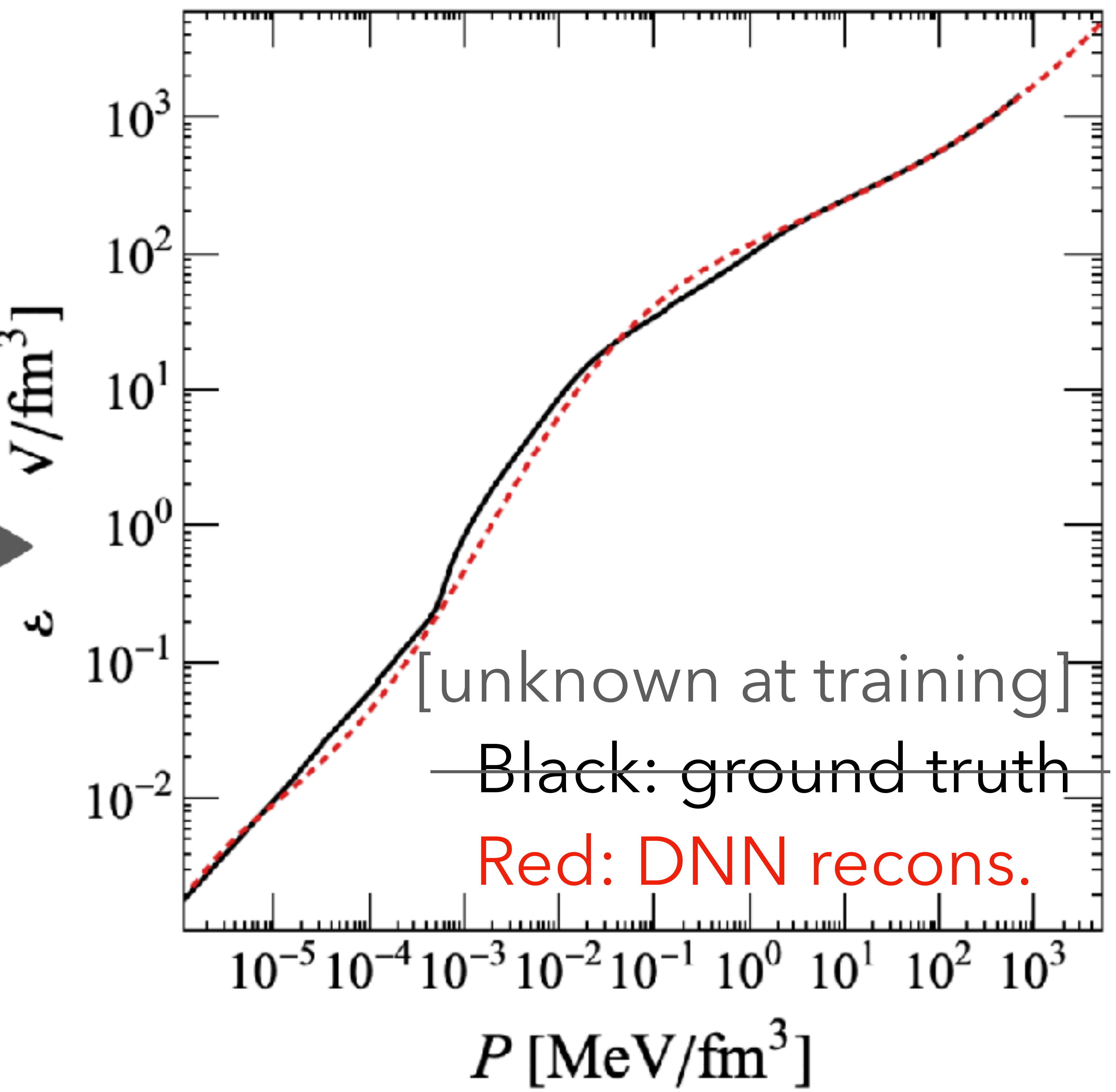




$$\chi^2 = \sum_i \left(\frac{m_i - m_i^{\rm obs}}{\Delta m_i}\right)^2 + \left(\frac{R_i - R_i^{\rm obs}}{\Delta R_i}\right)^2$$

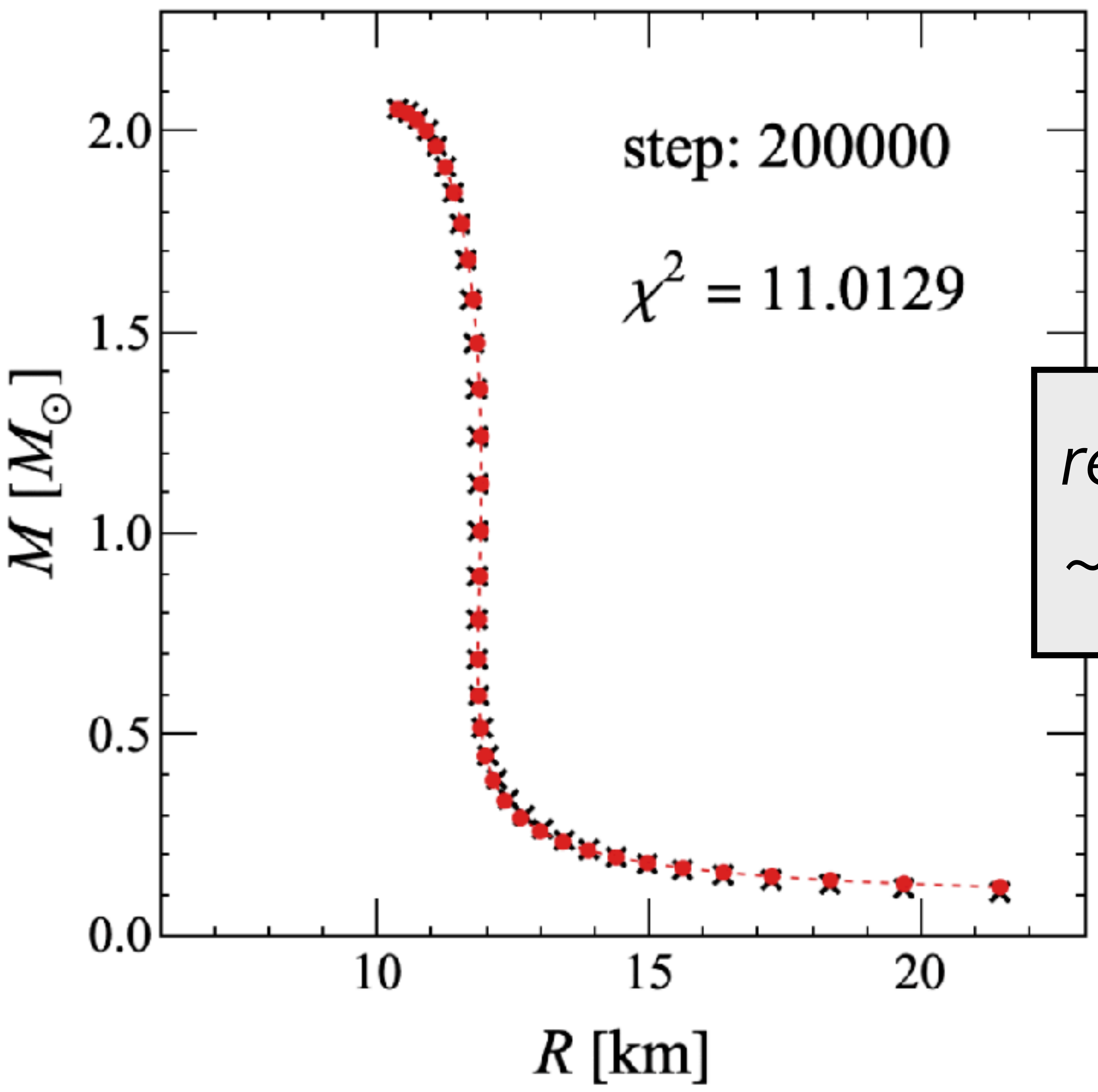
closure test

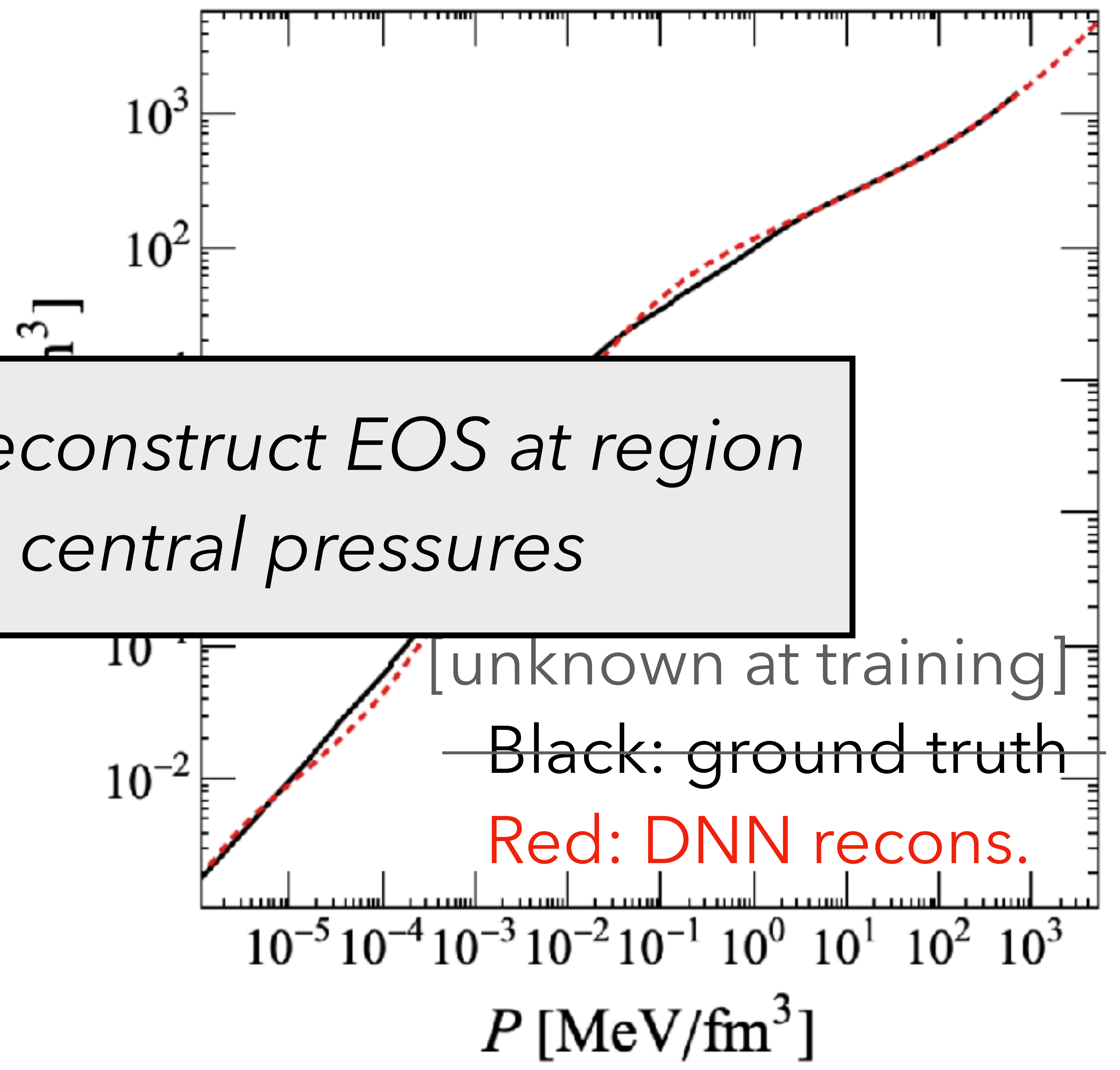




$$\chi^2 = \sum_i \left(\frac{m_i - m_i^{\rm obs}}{\Delta m_i}\right)^2 + \left(\frac{R_i - R_i^{\rm obs}}{\Delta R_i}\right)^2$$

closure test

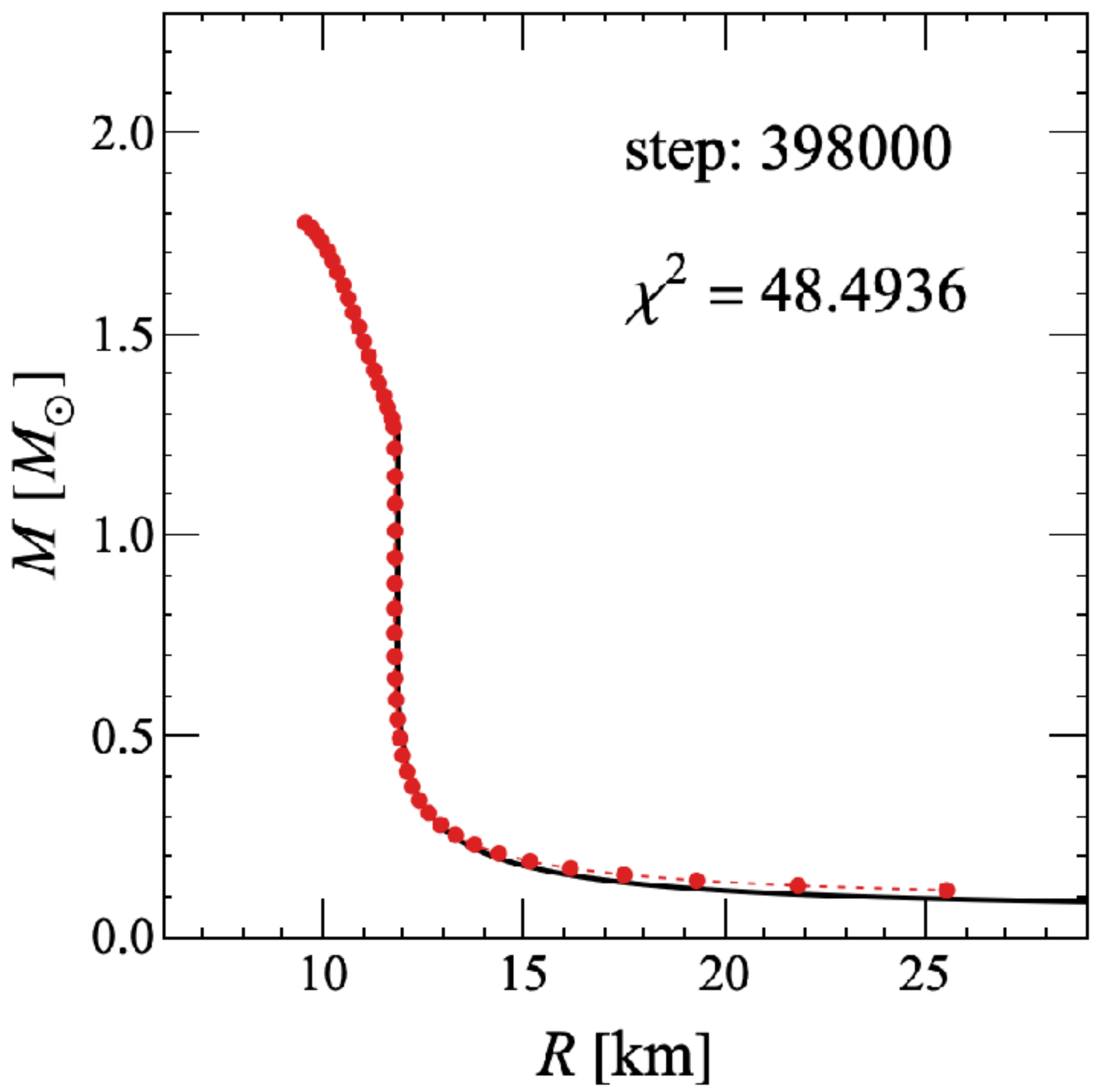


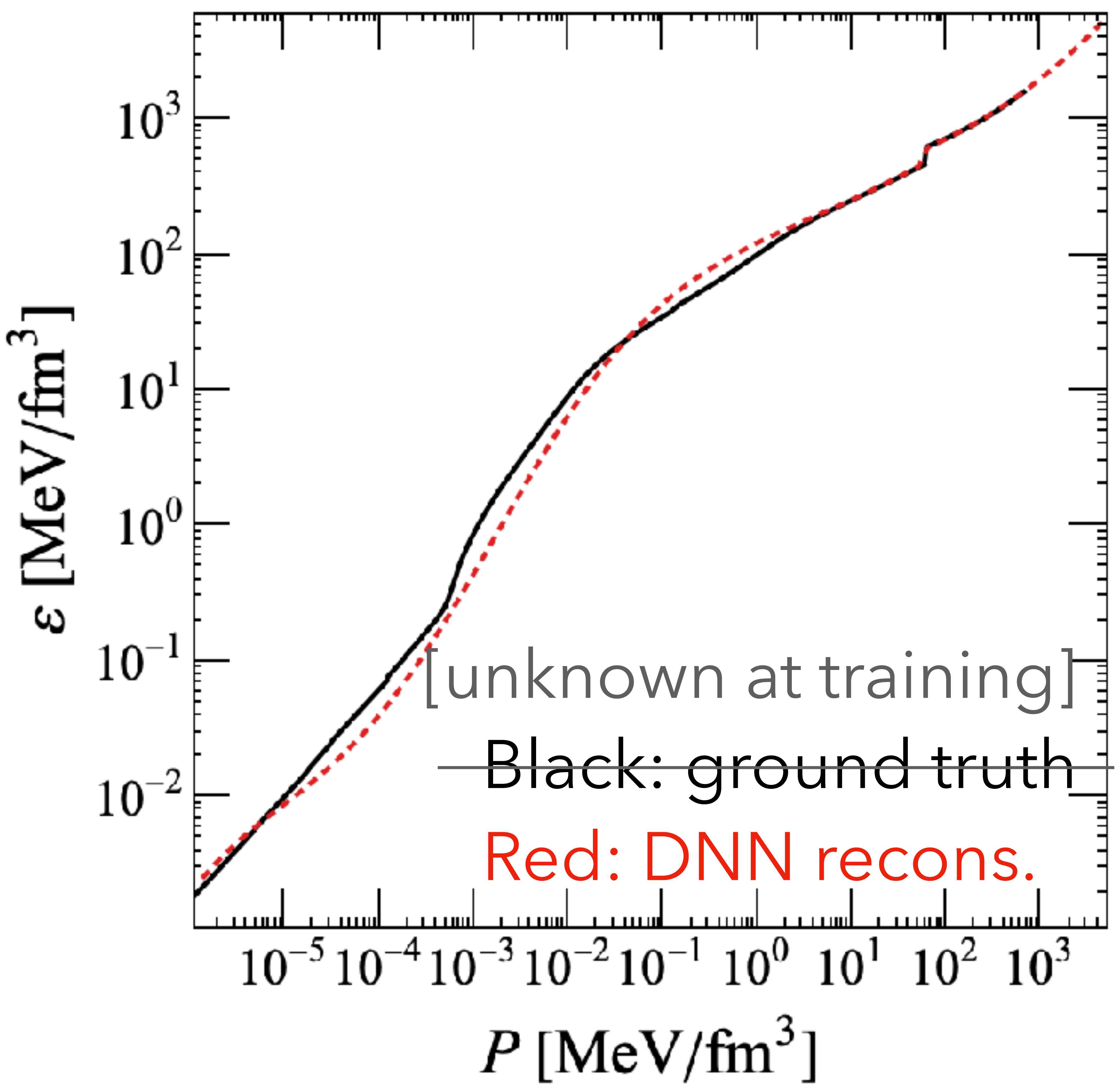


reconstruct EOS at region
~ central pressures

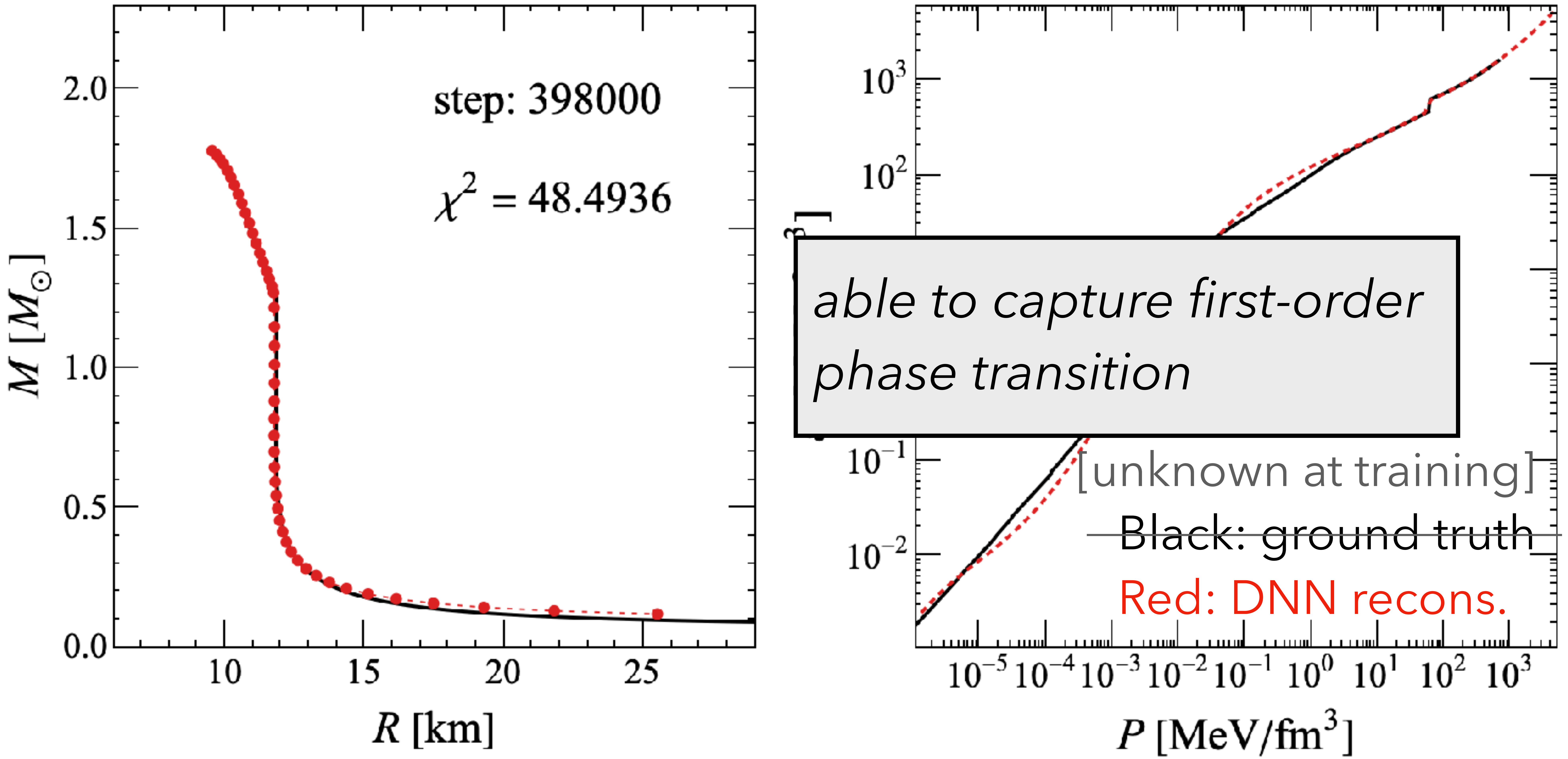
$$\chi^2 = \sum_i \left(\frac{m_i - m_i^{\rm obs}}{\Delta m_i}\right)^2 + \left(\frac{R_i - R_i^{\rm obs}}{\Delta R_i}\right)^2$$

closure test: with phase transition



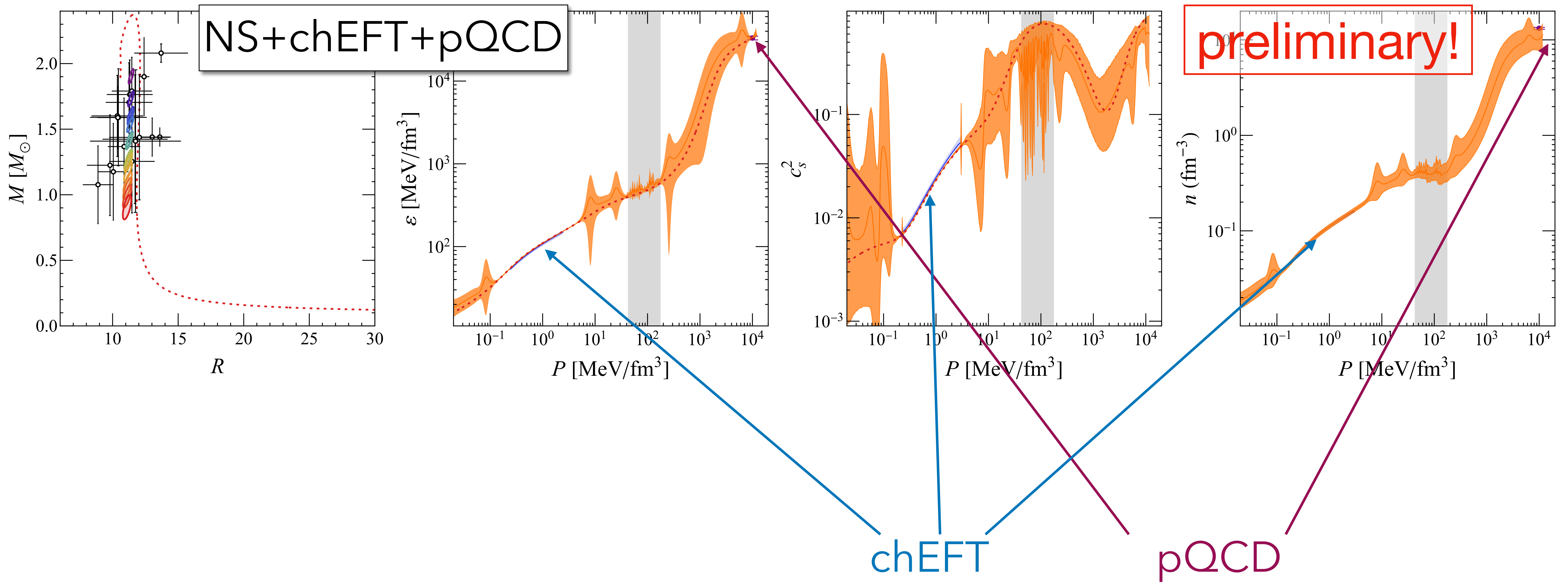


closure test: with phase transition

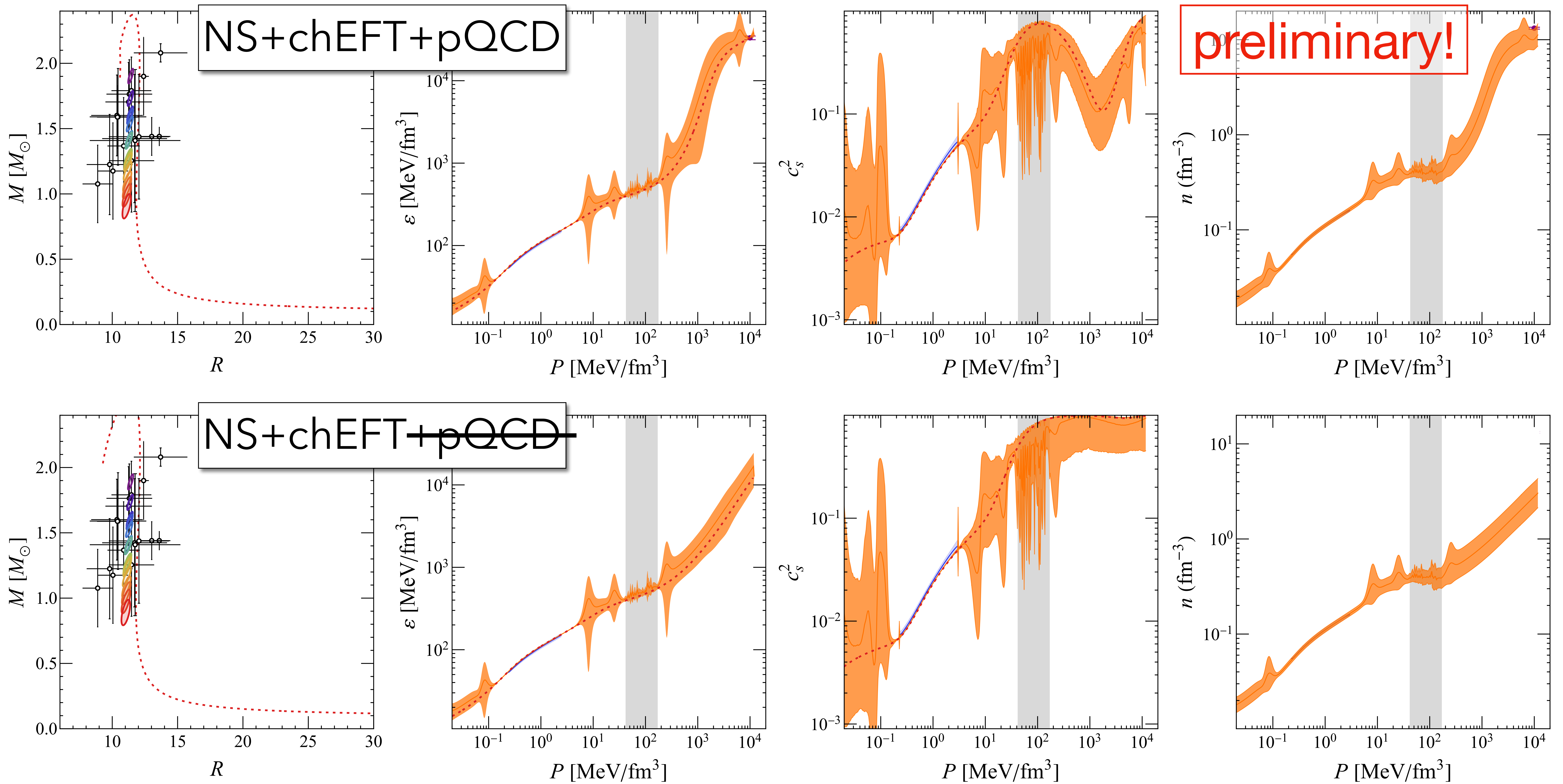


able to capture first-order phase transition

EOS reconstruction from NS + physics constraints



EOS reconstruction from NS + physics constraints



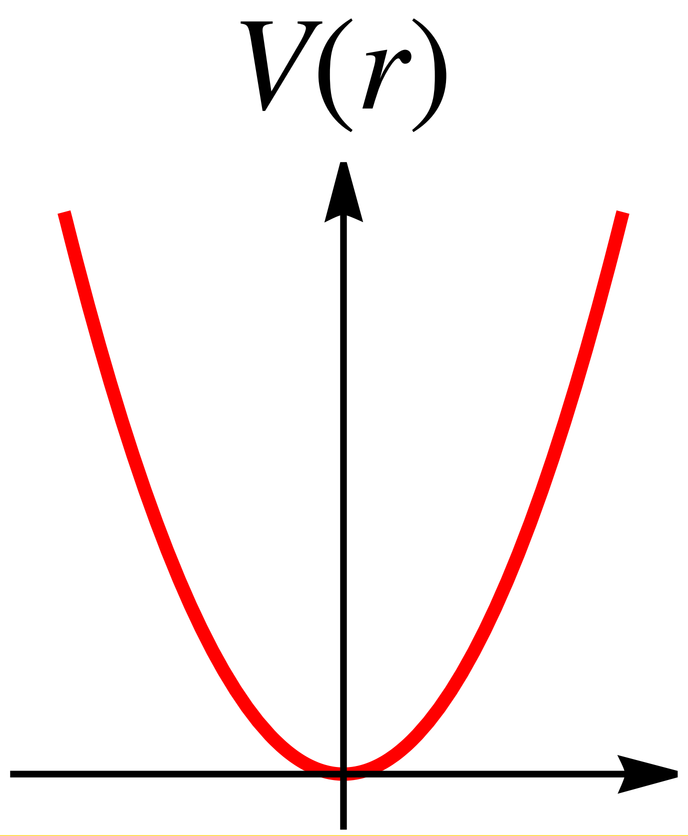
summary and outlook

- Derived *analytical* differential equations for linear responses of the TOV equation
    - tested using NN-EoS;
    - applicable to *tidal deformability* observables;
    - suitable for any parameterization of the EoS;
    - similar idea applicable to other physics topics

extension: Schroedinger equation

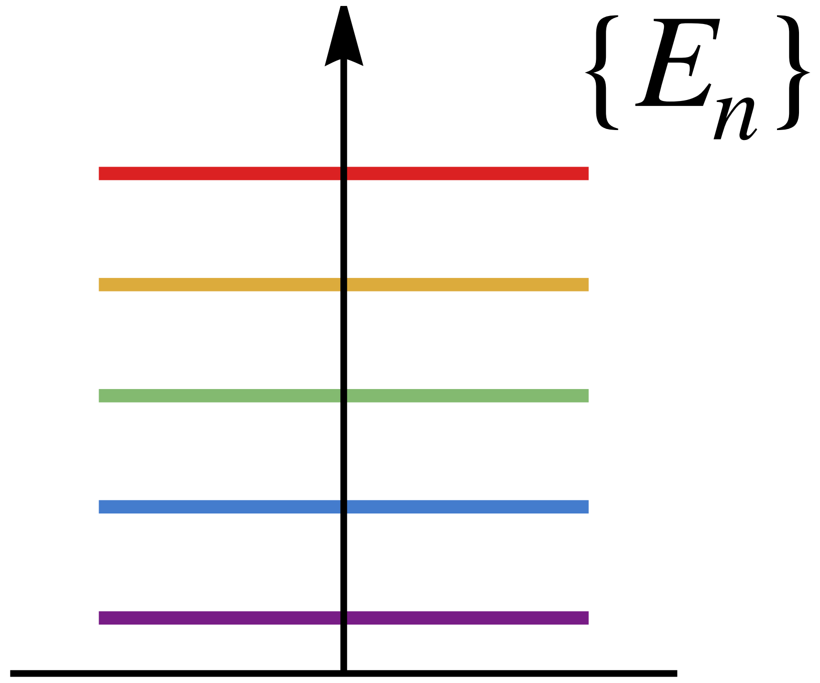
$$\hat{H}\psi_n = -\frac{\nabla^2}{2m}\psi_n + V(r)\psi_n = E_n\psi_n$$

$V(r)$ known $\Longrightarrow \{E_n, \psi_n(r)\}$:
numerical methods established.



information of interest

$\{E_n\}$ known $\Longrightarrow V(r)$

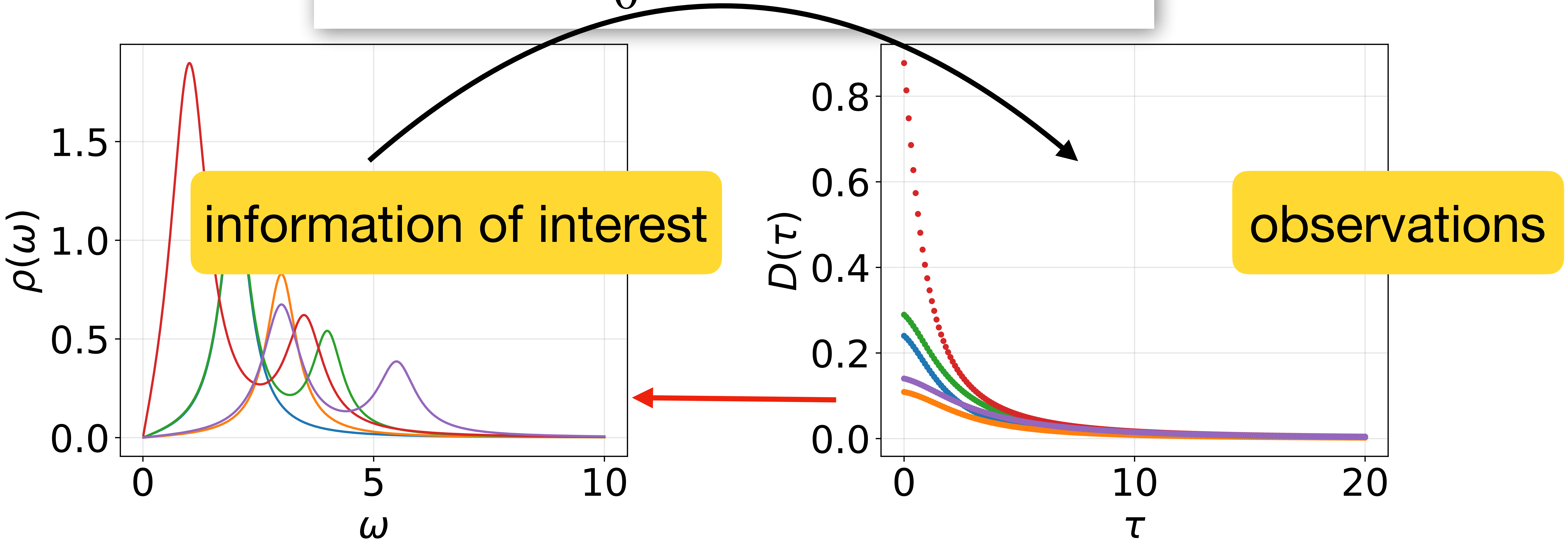


observations

reference: ***SS***, Zhou, Zhao, Mukherjee, Zhuang, PhysRevD.105.014017

extension: spectral function <=> correlation

$$D(p) = \int_0^\infty K(p,\omega)\,\rho(\omega)\,d\omega \qquad K(p,\omega) \equiv \frac{\pi^{-1}\,\omega}{\omega^2 + p^2}$$

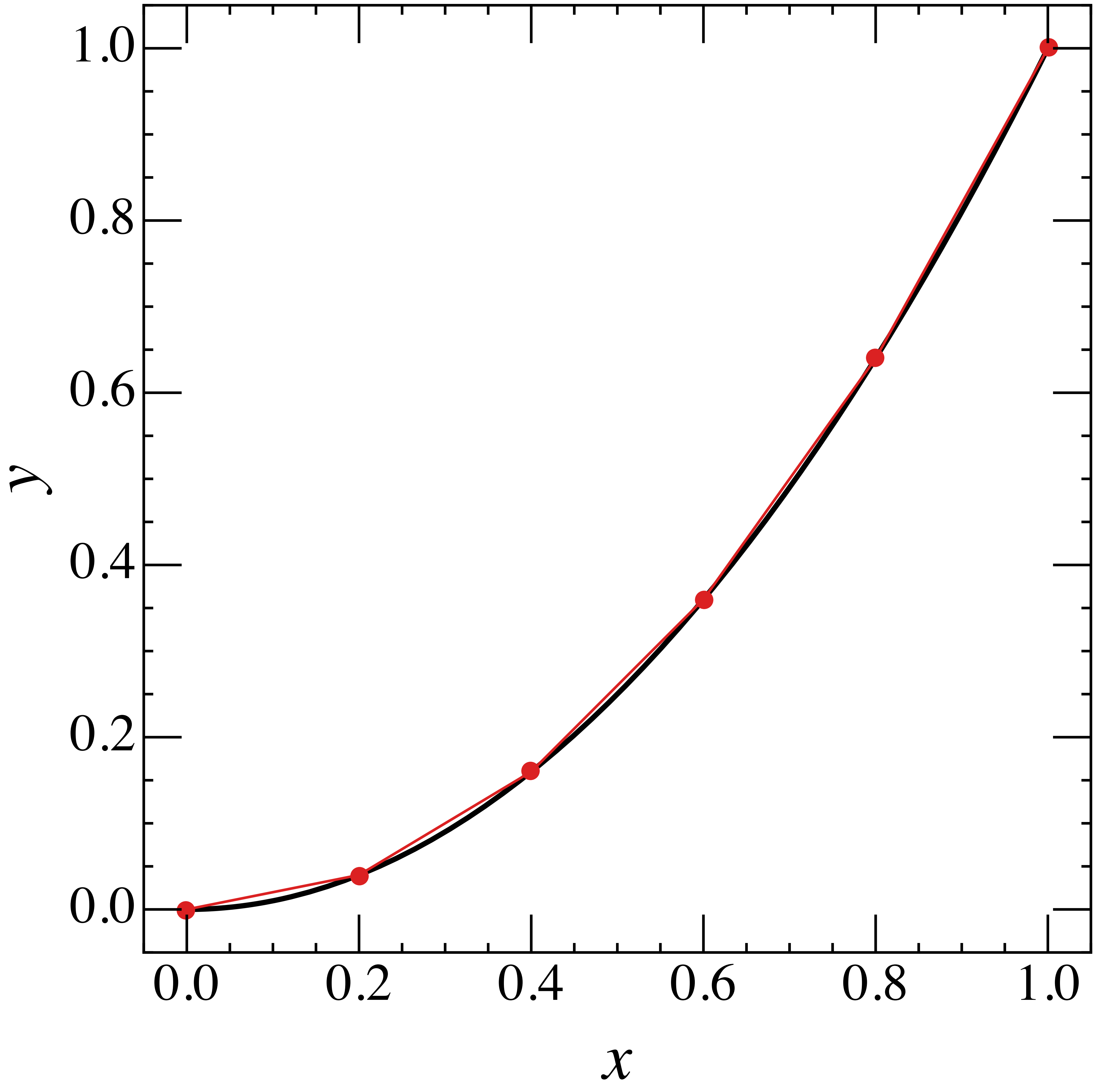


SS, Wang, Zhou, Comput.Phys.Commun. 282 (2023) 108547;
Wang, ***SS***, Zhou, Phys. Rev. D **106**, L051502;

What are Deep Neural Networks?

--- a general parameterization scheme to approximate continuous functions.

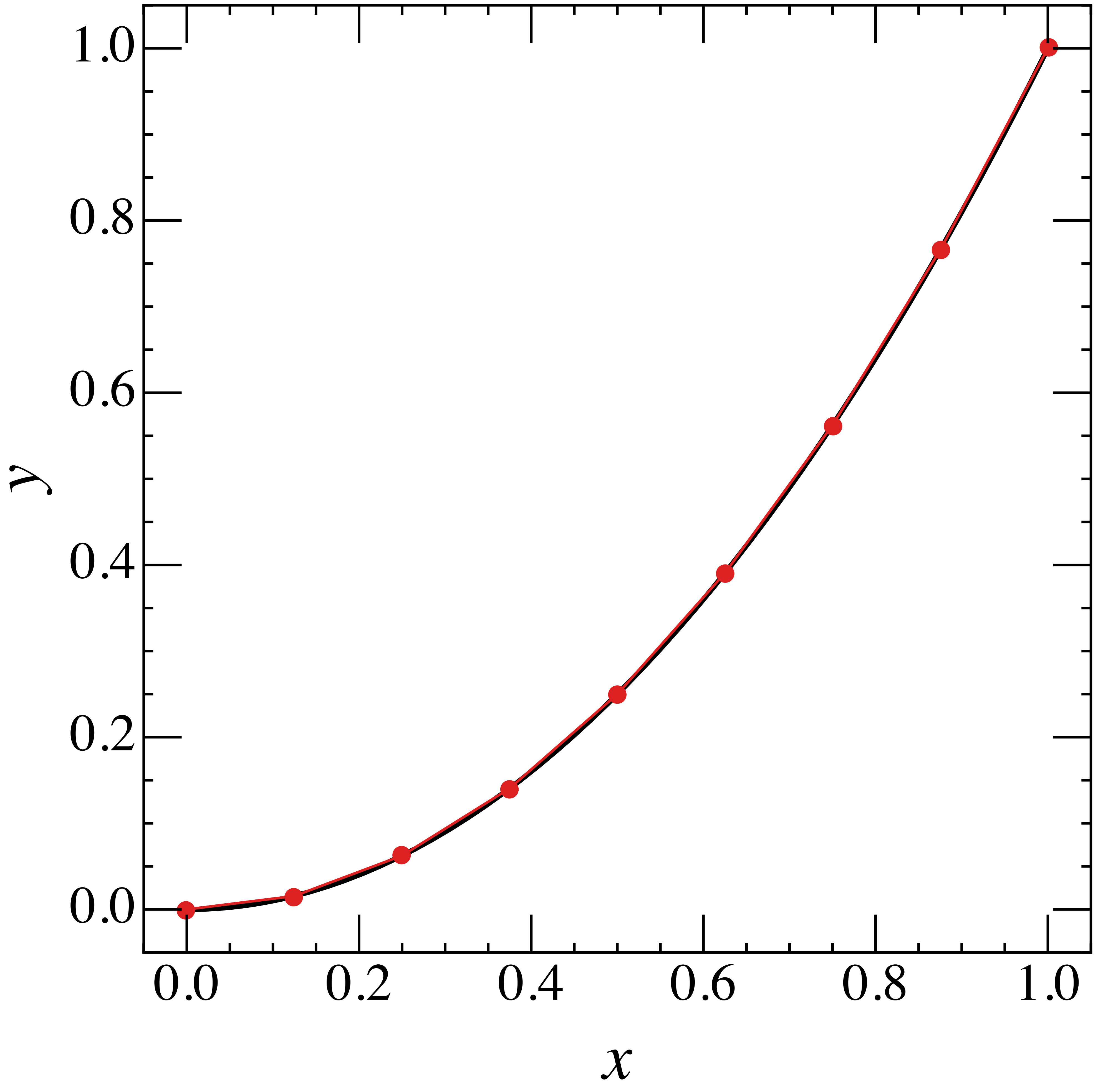
example: approximate $y = x^2$ for $x \in [0,1]$



What are Deep Neural Networks?

--- a general parameterization scheme to approximate continuous functions.

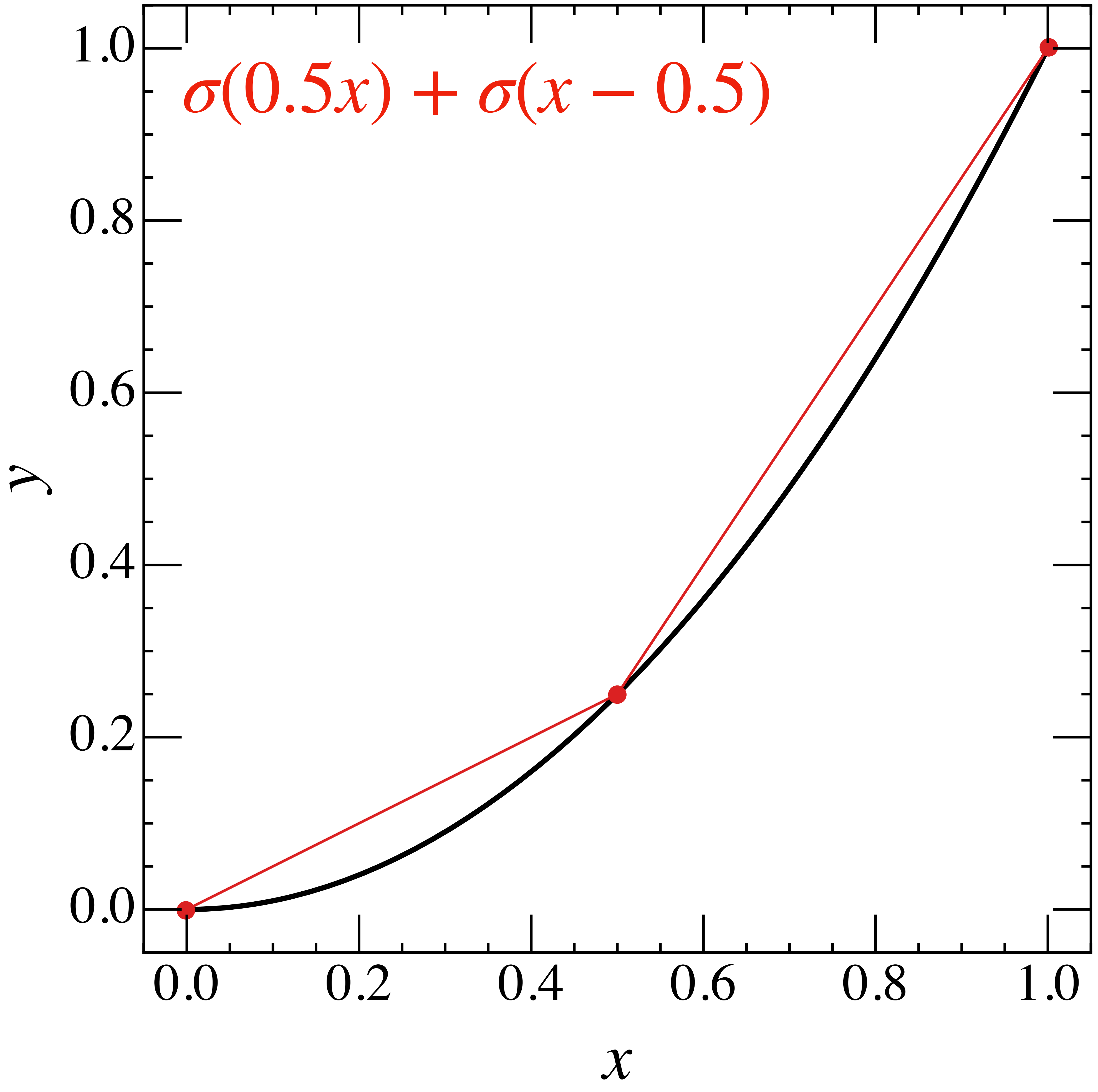
example: approximate $y = x^2$ for $x \in [0,1]$



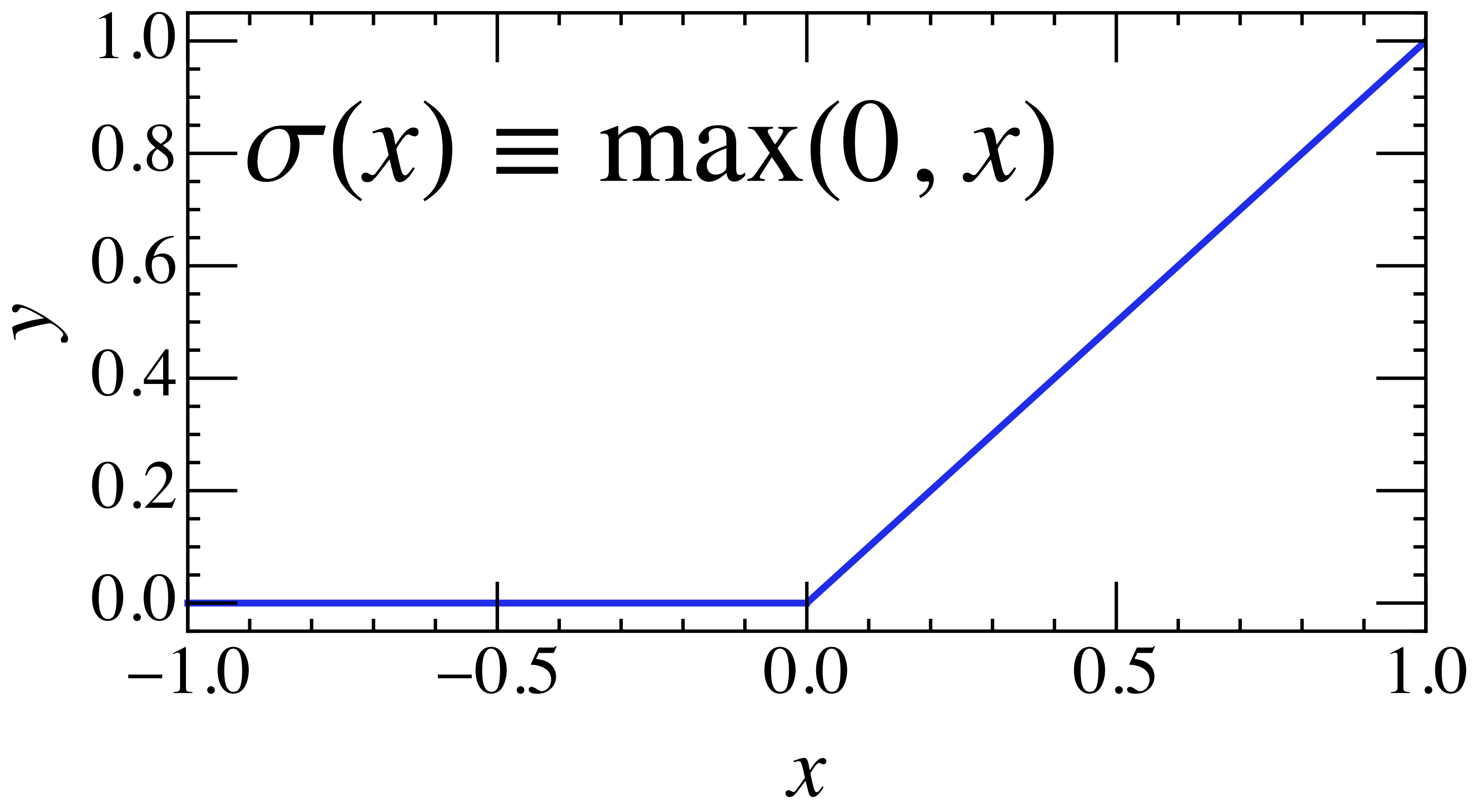
What are Deep Neural Networks?

--- a general parameterization scheme to approximate continuous functions.

example: approximate $y = x^2$ for $x \in [0,1]$



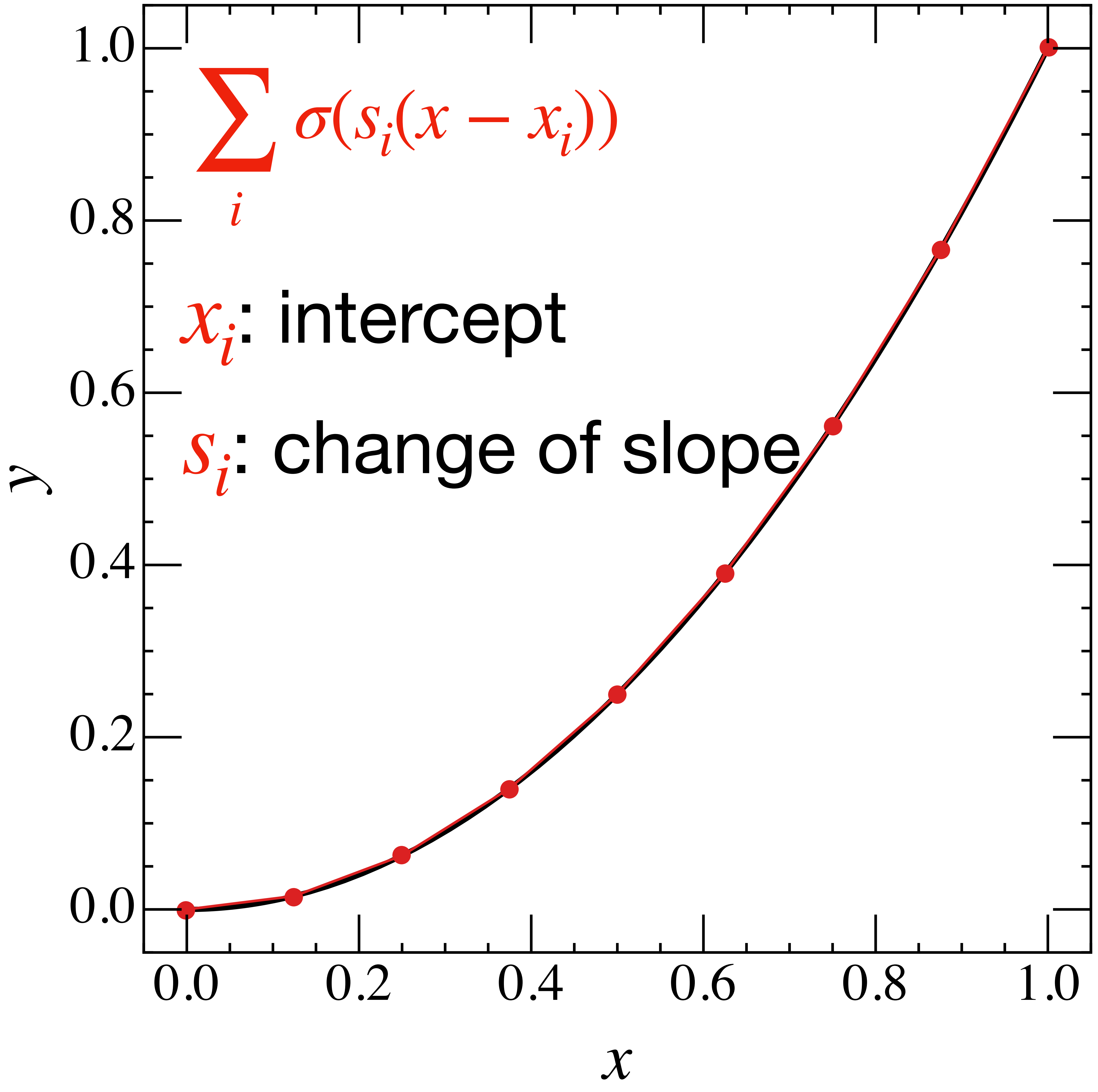
define

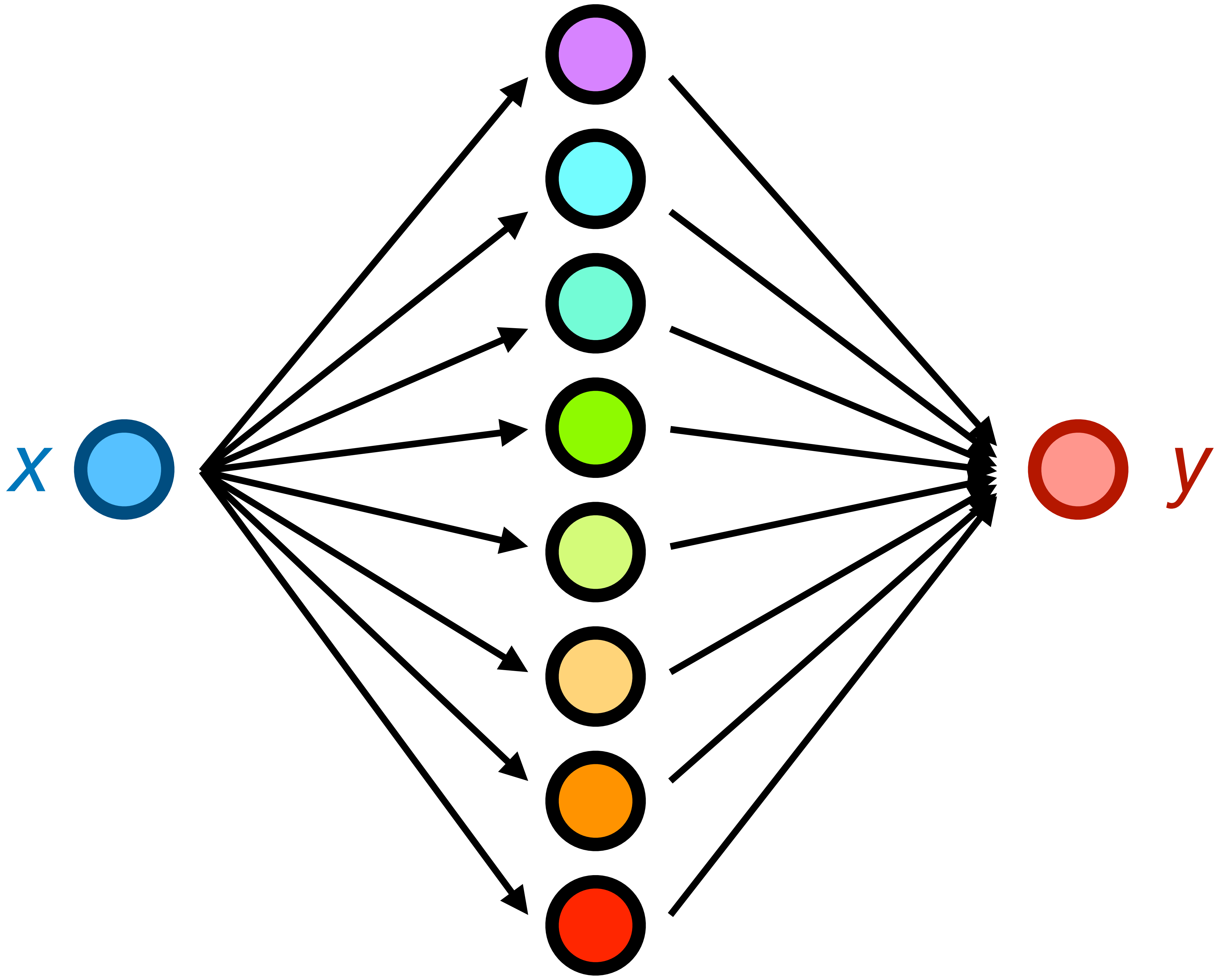


What are Deep Neural Networks?

--- a general parameterization scheme to approximate continuous functions.

example: approximate $y = x^2$ for $x \in [0,1]$





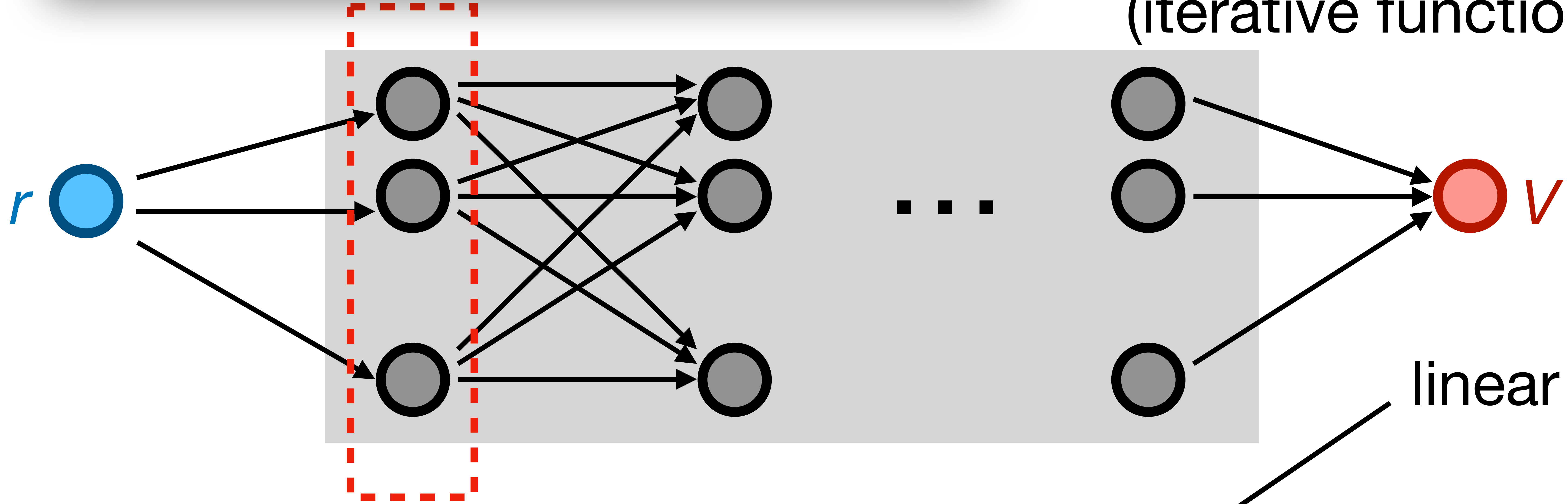
each represents one of $\sigma(s_i(x - x_i))$

What are Deep Neural Networks?

--- a general parameterization scheme to approximate continuous functions.

$$V(r) \approx V_{\mathrm{DNN}}(r \,|\, \text{parameters})$$

(iterative function substitution)

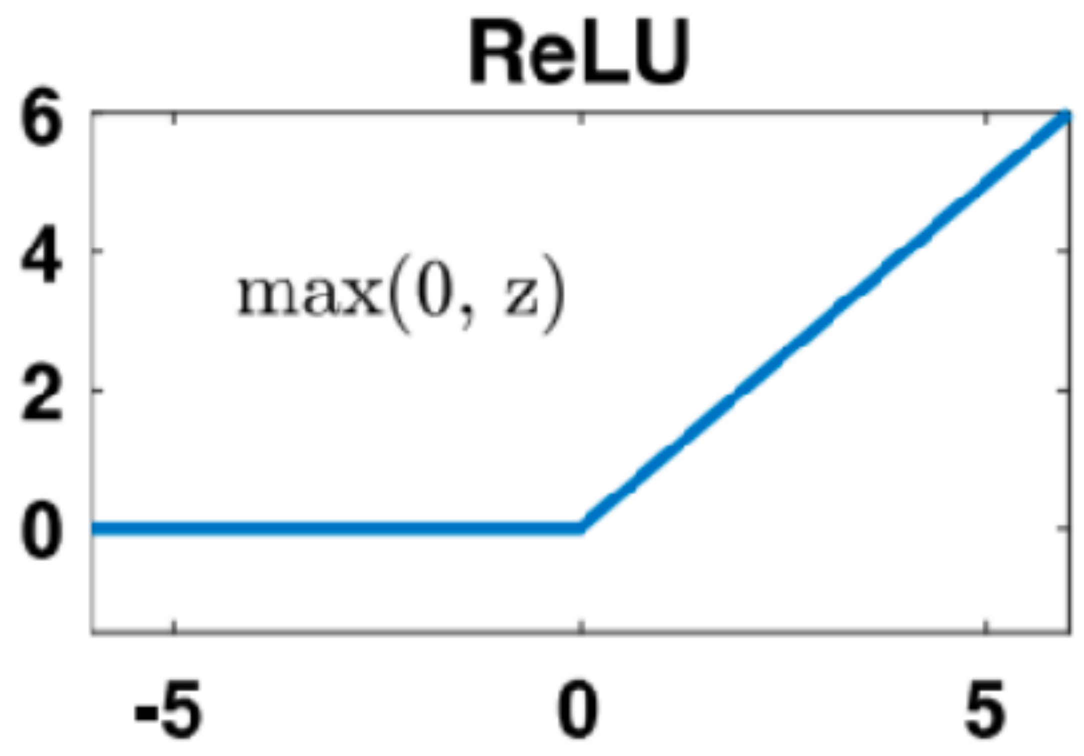


linear

nonlinear

Each ● is an intermediate function $(a_i^{(l)})$:

- At the first layer: $z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)}\, r, \qquad a_i^{(1)} = \sigma(z_i^{(1)})$

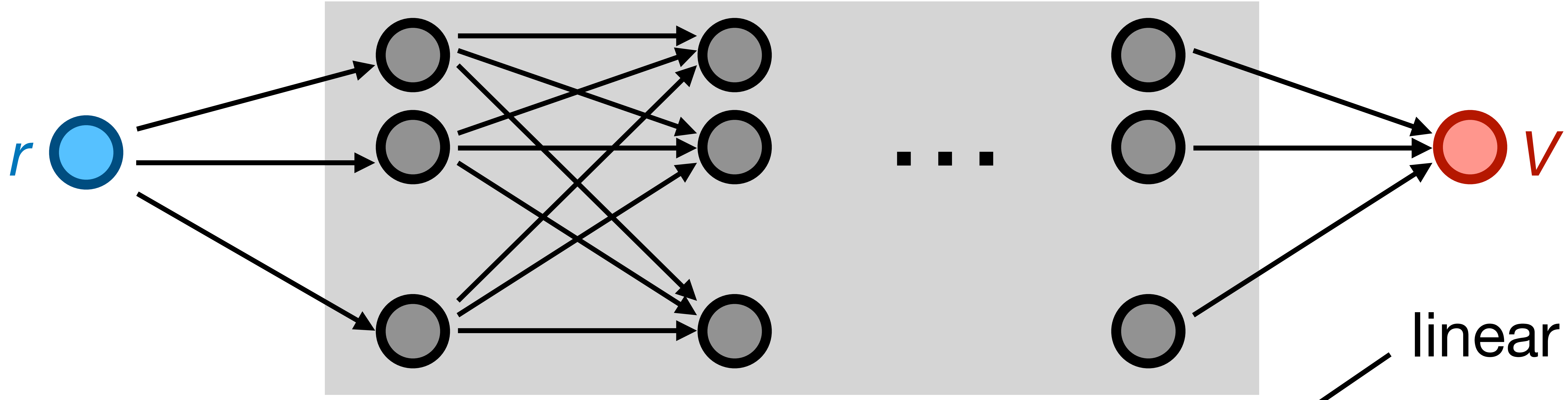


What are Deep Neural Networks?

--- a general parameterization scheme to approximate continuous functions.

$$\boxed{V(r) \approx V_{\mathrm{DNN}}(r \,|\, \text{parameters})}$$

(iterative function substitution)

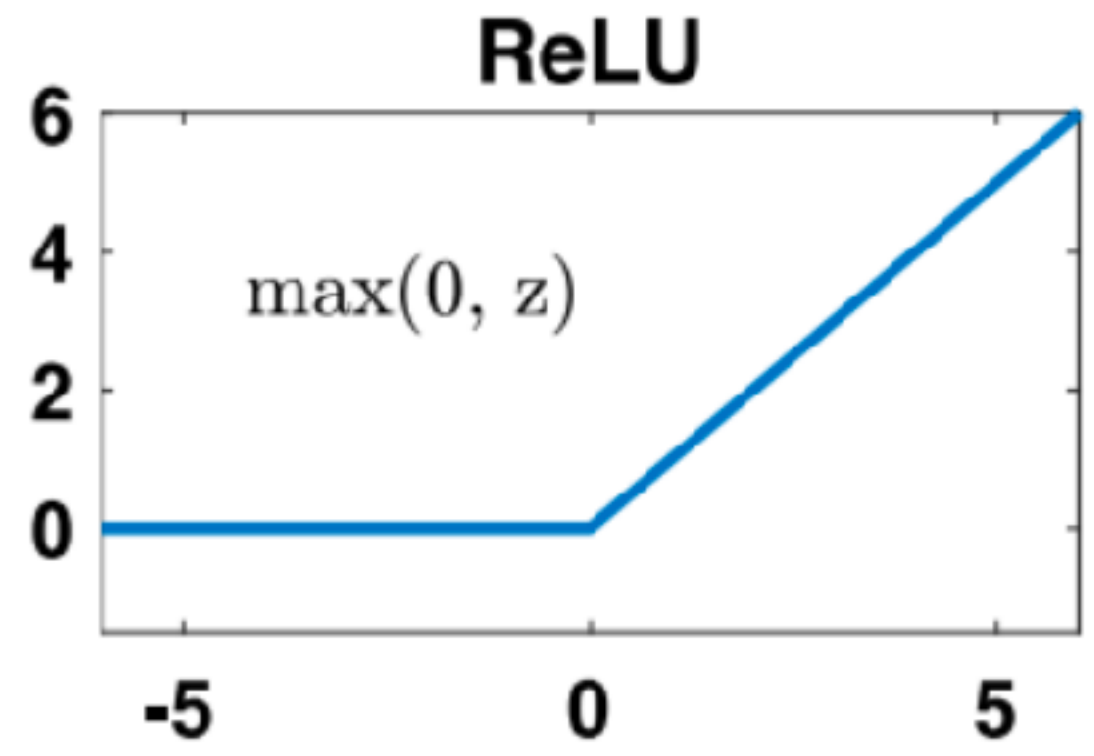


linear

nonlinear

Each ● is an intermediate function ($a_i^{(l)}$):

- At the first layer: $z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)}\, r, \qquad a_i^{(1)} = \sigma(z_i^{(1)})$
- At later layers: $z_i^{(l)} = b_i^{(l)} + \sum_j W_{i,j}^{(l)}\, a_j^{(l-1)}, \qquad a_i^{(l)} = \sigma(z_i^{(l)})$

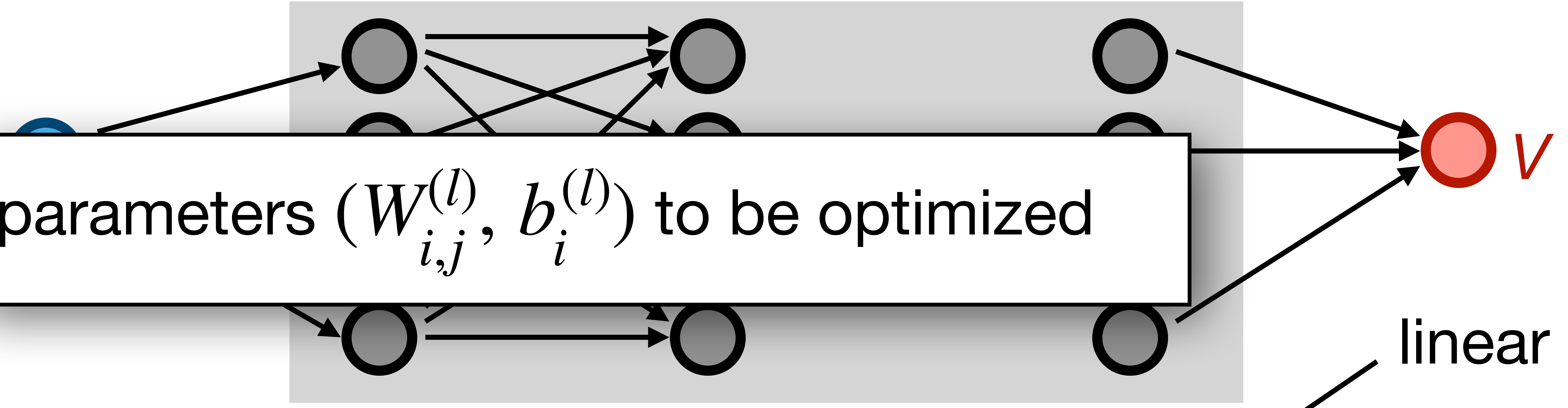


What are Deep Neural Networks?

--- a general parameterization scheme to approximate continuous functions.

$$V(r) \approx V_{\text{DNN}}(r\,|\,\text{parameters})$$

(iterative function substitution)



parameters $(W_{i,j}^{(l)}, b_i^{(l)})$ to be optimized

Each ⬤ is an intermediate function ($a_i^{(l)}$):

linear

nonlinear

- At the first layer: $z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)}\, r$, $\quad a_i^{(1)} = \sigma(z_i^{(1)})$
- At later layers: $z_i^{(l)} = b_i^{(l)} + \sum_j W_{i,j}^{(l)}\, a_j^{(l-1)}$, $\quad a_i^{(l)} = \sigma(z_i^{(l)})$