How Machine-Learning helps in the subtleties of the spectral function reconstruction

— *what we learned from solving inverse problems via Deep Learning*

施舒哲 清华大学

refs:

SS, Zhou, Zhao, Mukherjee, Zhuang, Phys. Rev. D **105**, 014017;
Wang, ***SS***, Zhou, Phys. Rev. D **106**, L051502;
SS, Wang, Zhou, Comput.Phys.Commun. 282 (2023) 108547.

a recent review:

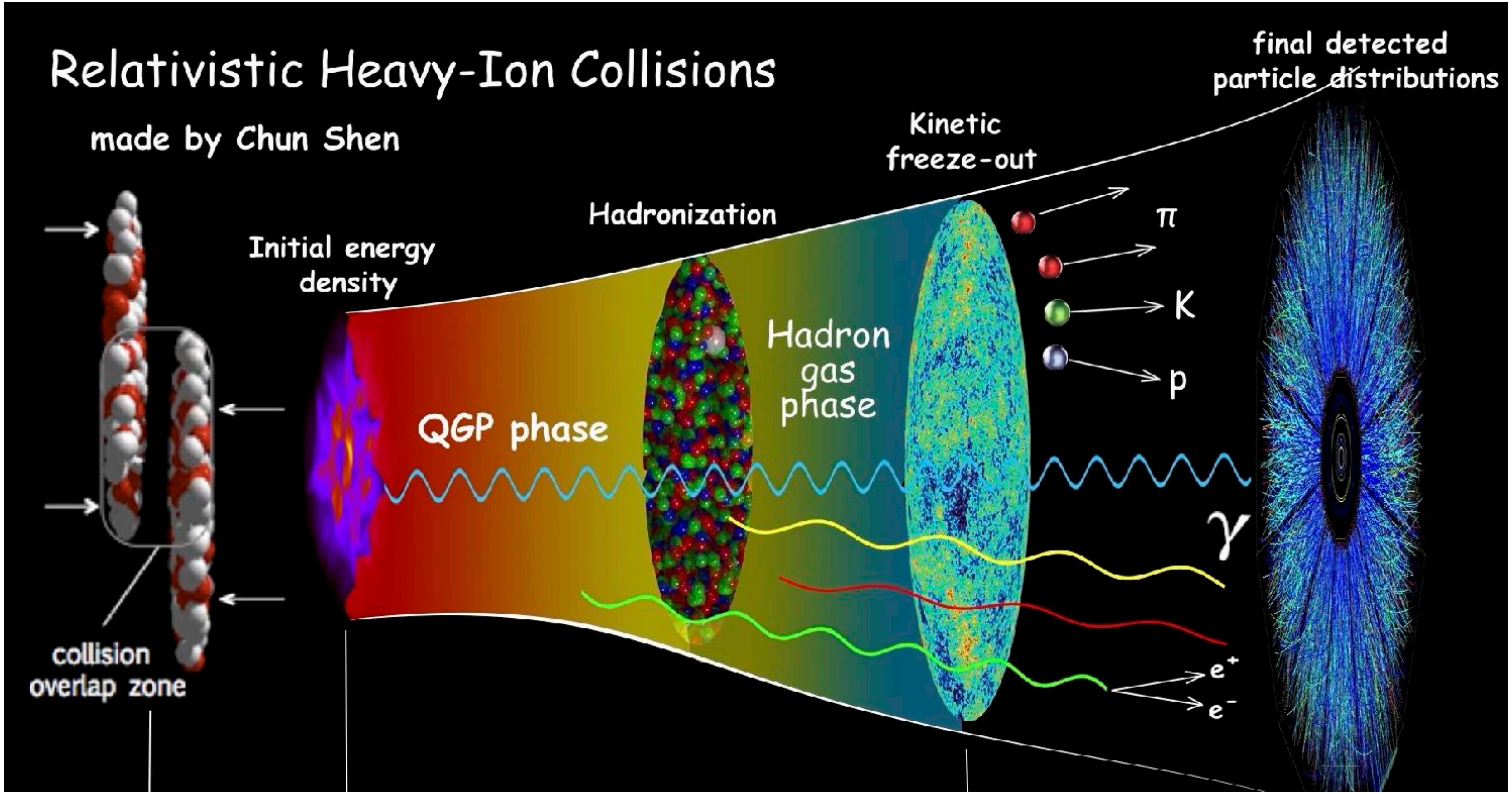
Zhou, Wang, Pang, ***SS*** Prog.Part.Nucl.Phys. 104084(2023)[2303.15136].

Outline

- Quarkonium potential

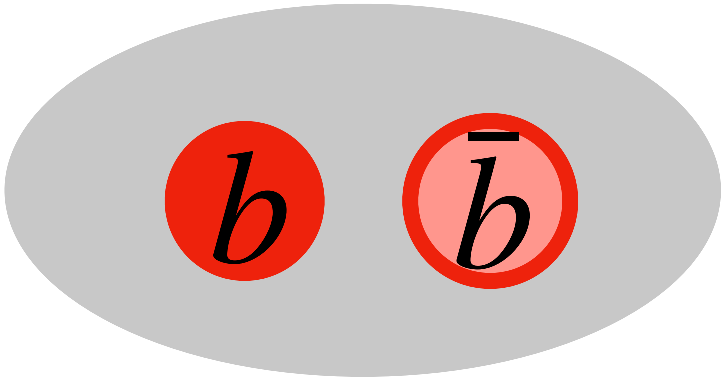
- Difficulties in spectral function reconstruction

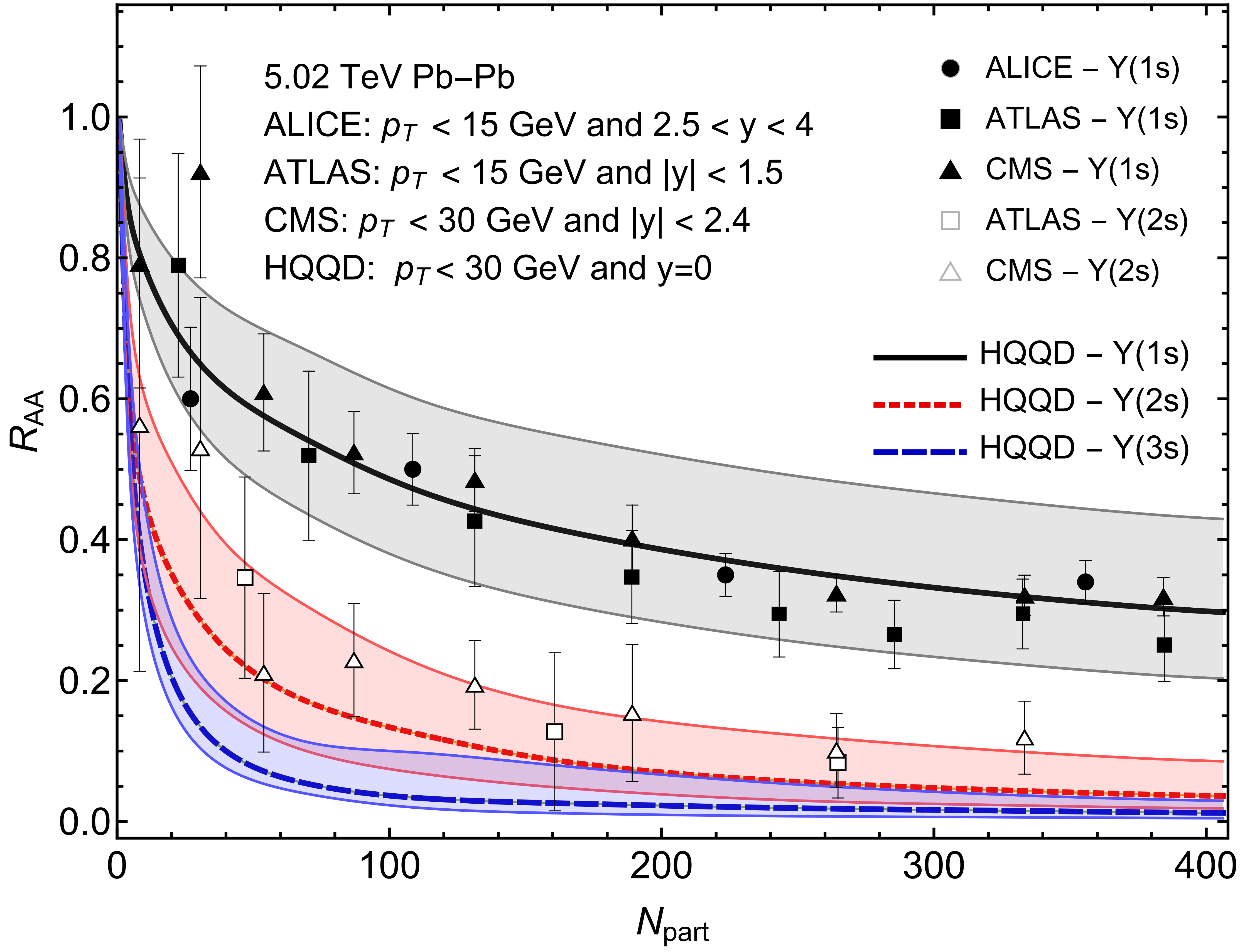
Relativistic Heavy-Ion Collisions



Quarkonium in the QGP

- In heavy-ion collisions, quarkonium production serves as a probe of the QGP.





Quarkonium in the QGP

- In heavy-ion collisions, quarkonium production serves as a probe of the QGP.
- Accurate understanding of the in-medium heavy-quark interaction?
  - Real potential modified by color-screening
  - Imaginary potential arises due to $(Q\bar{Q})_1 \to (Q\bar{Q})_8$, Landau damping, ...

Quarkonium in the QGP

- In heavy-ion collisions, quarkonium production serves as a probe of the QGP.
- Accurate understanding of the in-medium heavy-quark interaction?
  - Real potential modified by color-screening
  - Imaginary potential arises due to $(Q\bar{Q})_1 \to (Q\bar{Q})_8$, Landau damping, ...

Hard Thermal Loop potentials

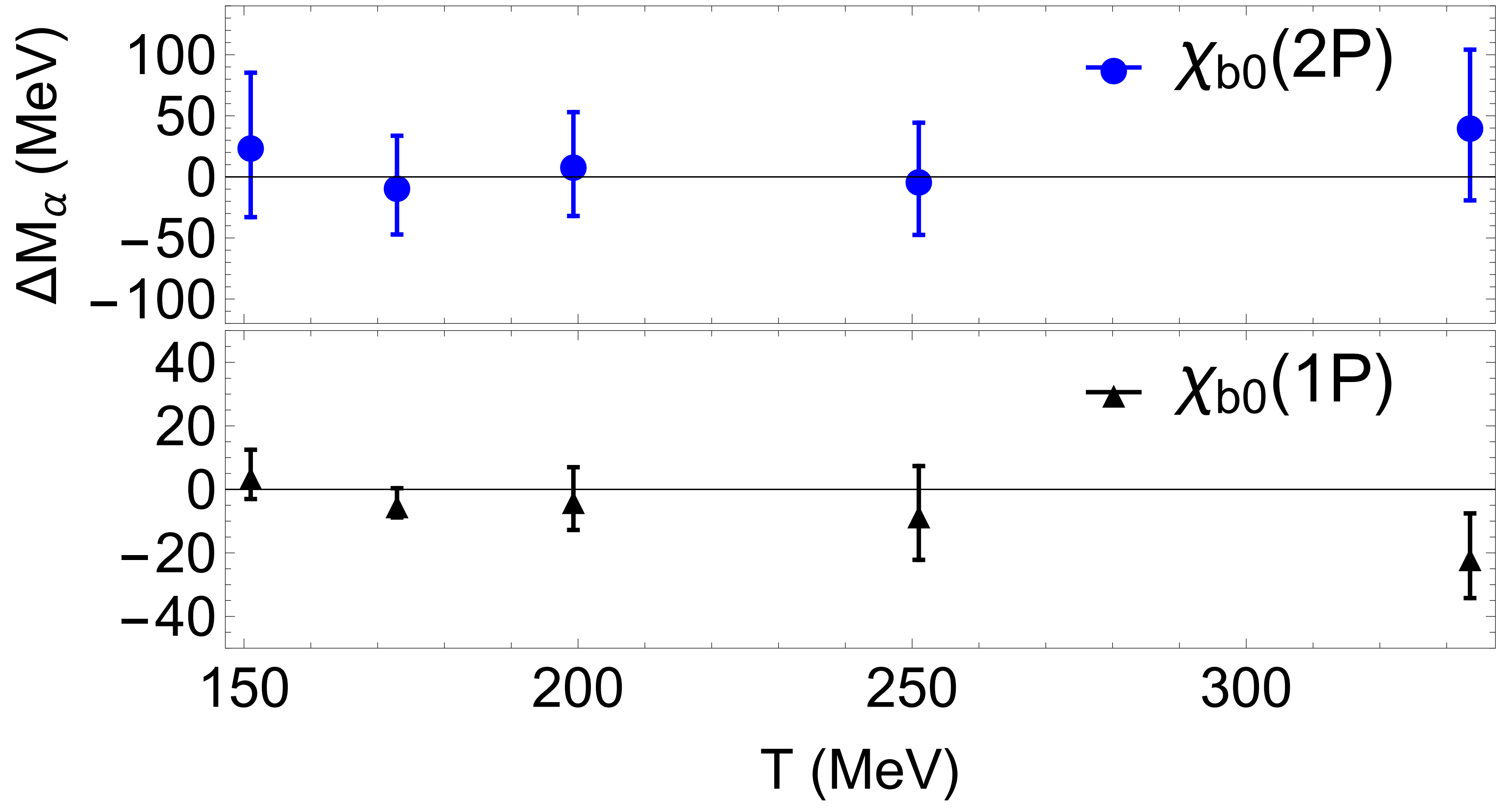
$$V_R(T,r) = \frac{\sigma}{\mu_D}\left(2 - (2+\mu_D r)e^{-\mu_D r}\right) - \alpha\left(\mu_D + \frac{e^{-\mu_D r}}{r}\right) + B\,,$$

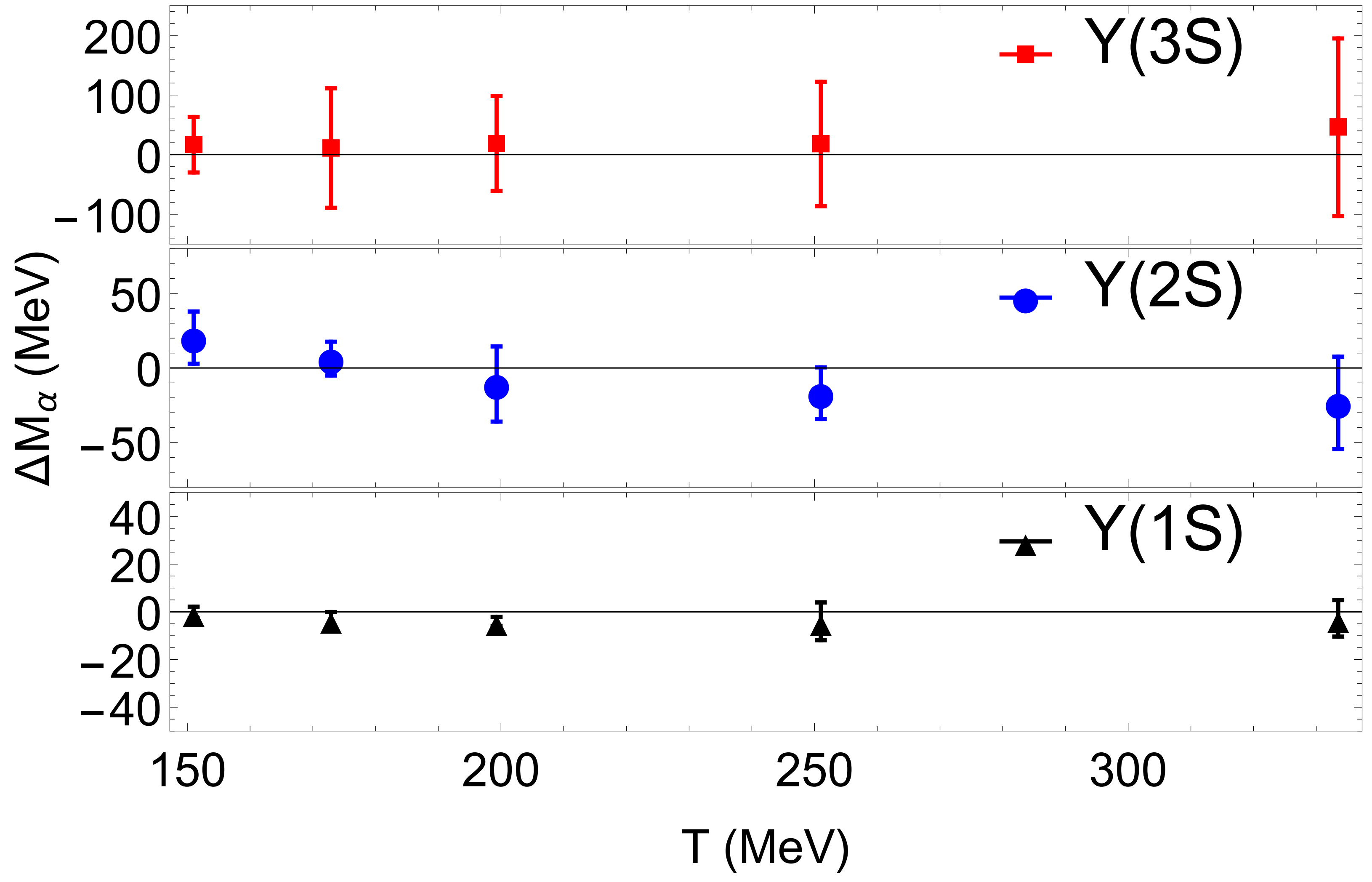
$$V_I(T,r) = -\frac{\sqrt{\pi}}{4}\mu_D T\sigma r^3 G^{2,2}_{2,4}\left(\begin{matrix}-\frac{1}{2},-\frac{1}{2}\\ \frac{1}{2},\frac{1}{2},-\frac{3}{2},-1\end{matrix}\middle|\,\frac{\mu_D^2 r^2}{4}\right) - \alpha\,T\,\phi(\mu_D r)\,.$$

see e.g., Laine, Philipsen, Romatschke, and Tassler, JHEP 03, 054 (2007)

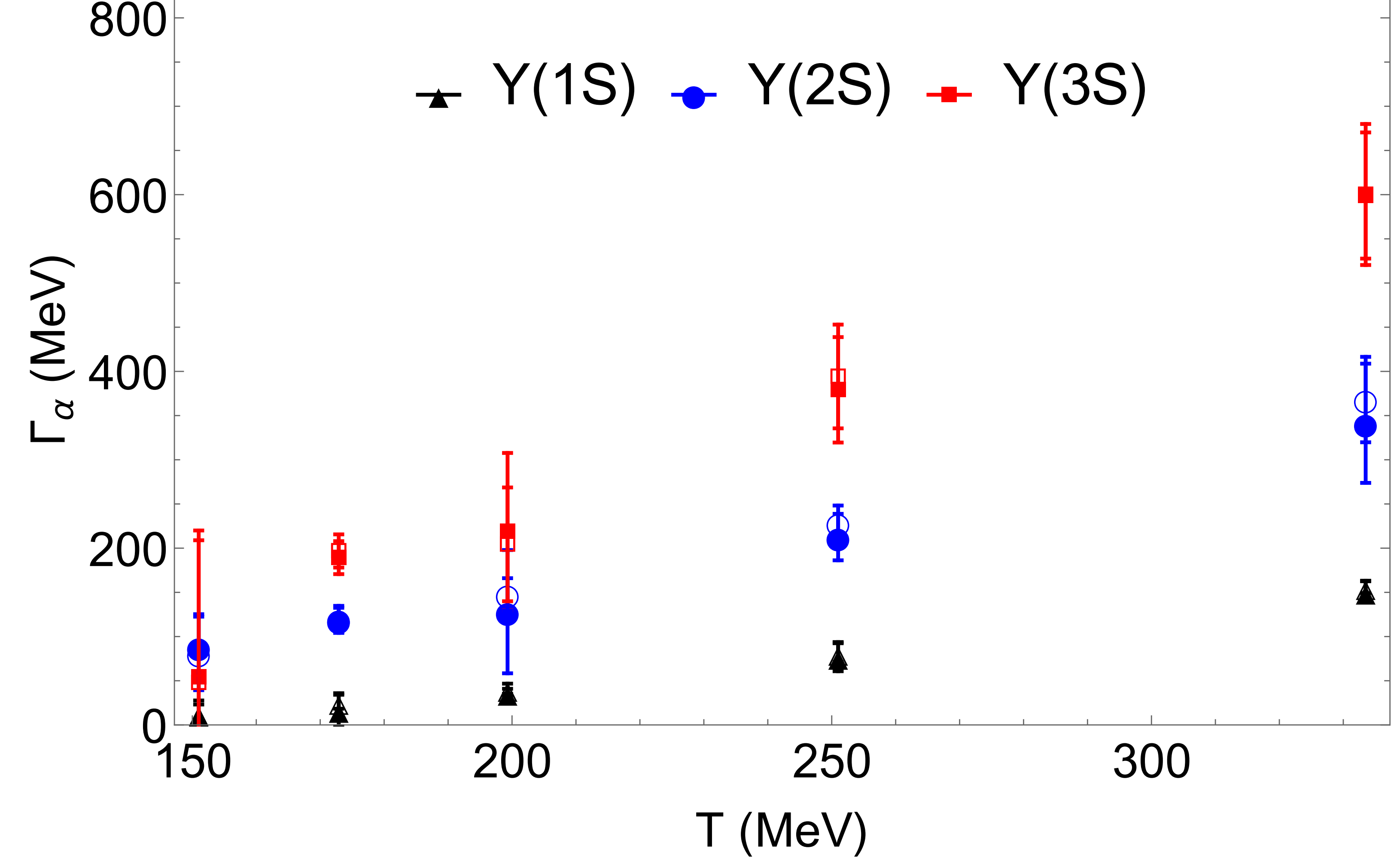
Bottomonium mass and thermal width, lattice QCD with finite m_Q

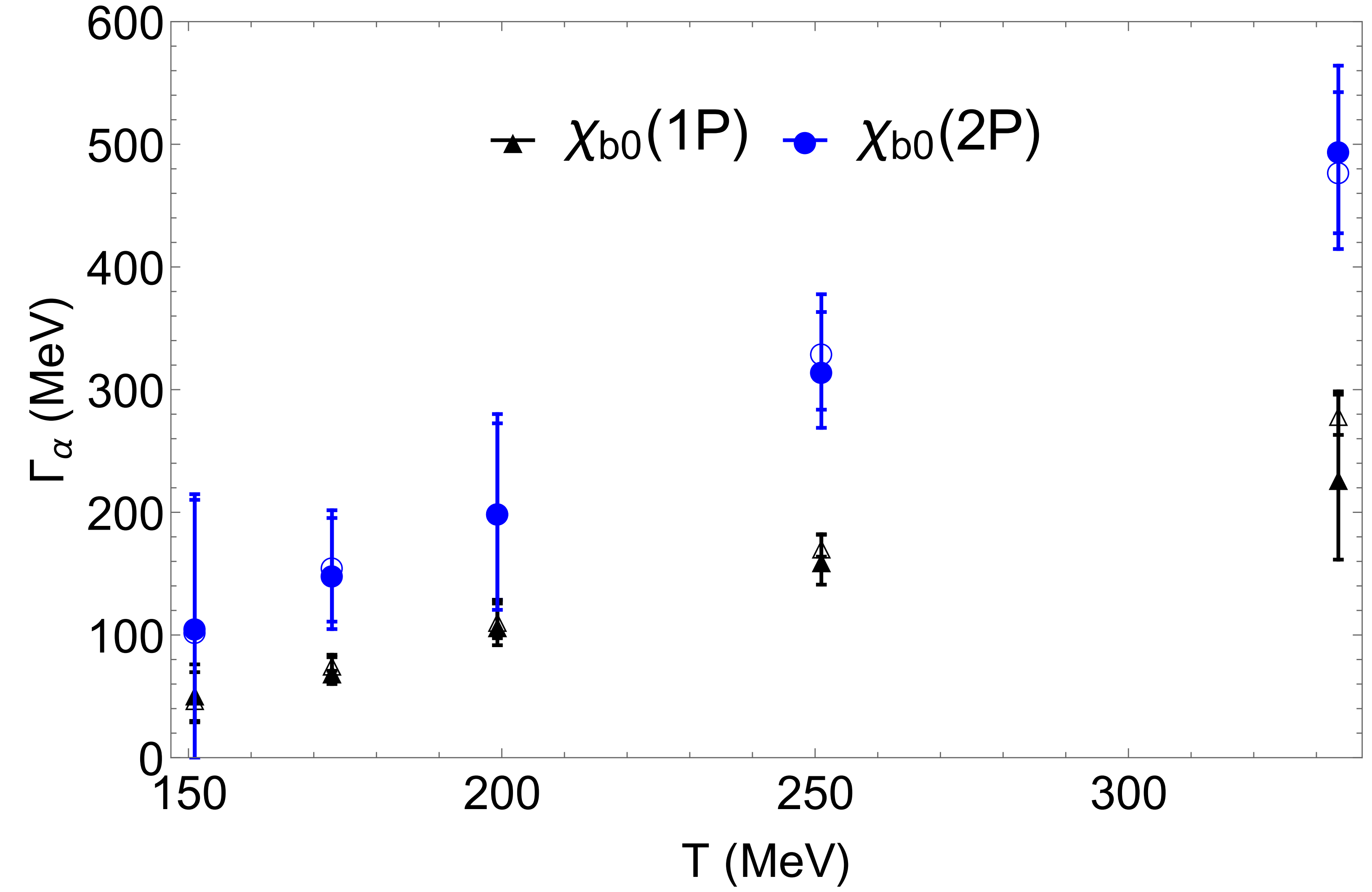
Mass





Thermal Width

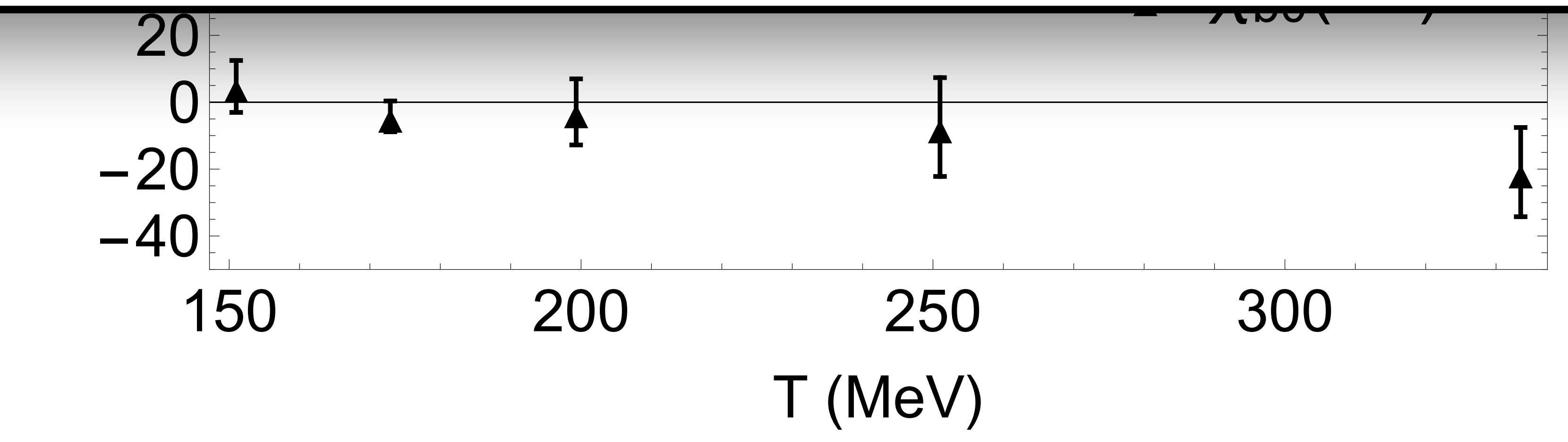


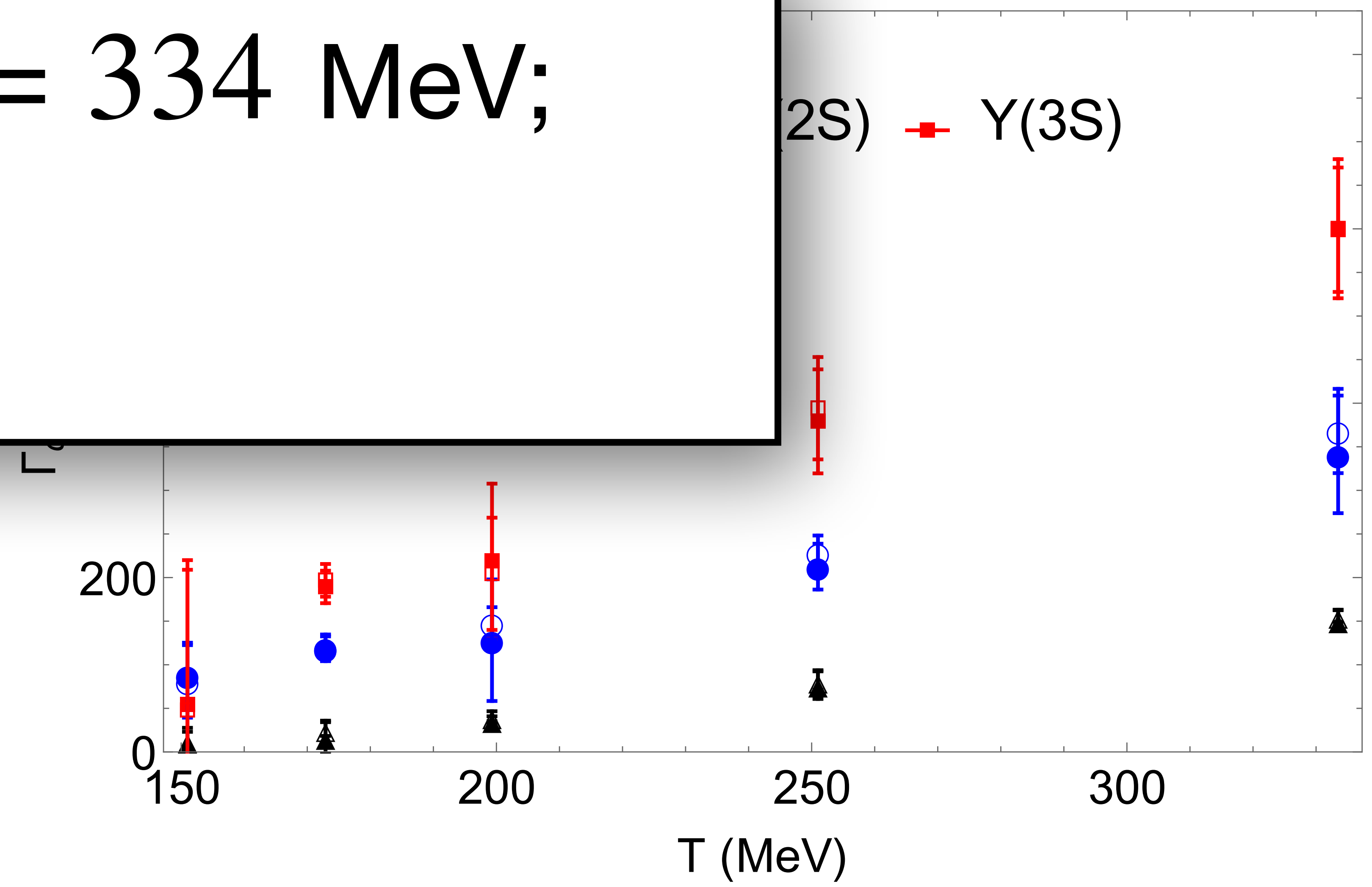


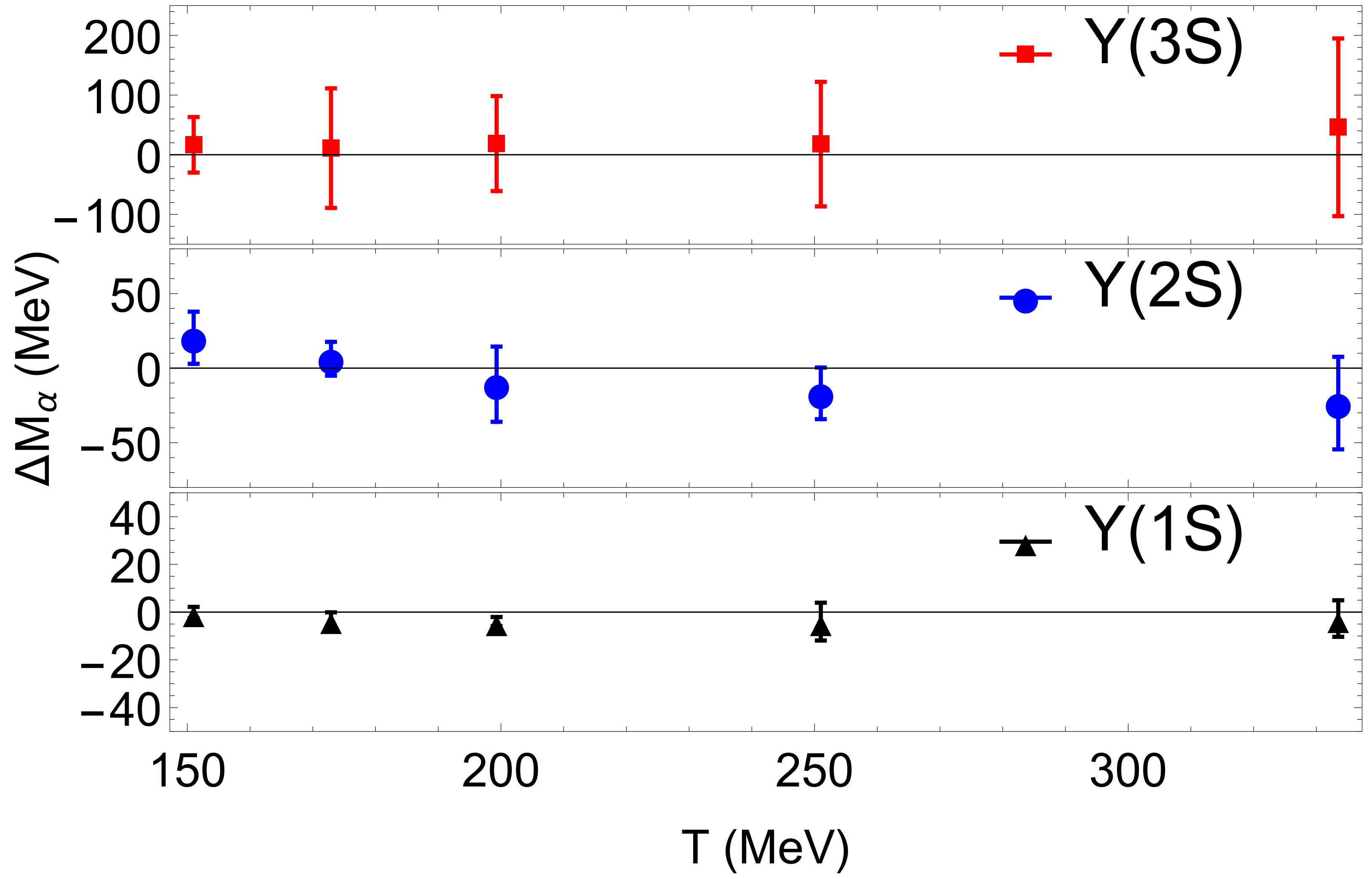
R. Larsen, S. Meinel, S. Mukherjee, and P. Petreczky:
Phys.Rev.D100,074506(2019), Phys.Lett.B800,135119(2020), Phys.Rev.D102,114508(2020)

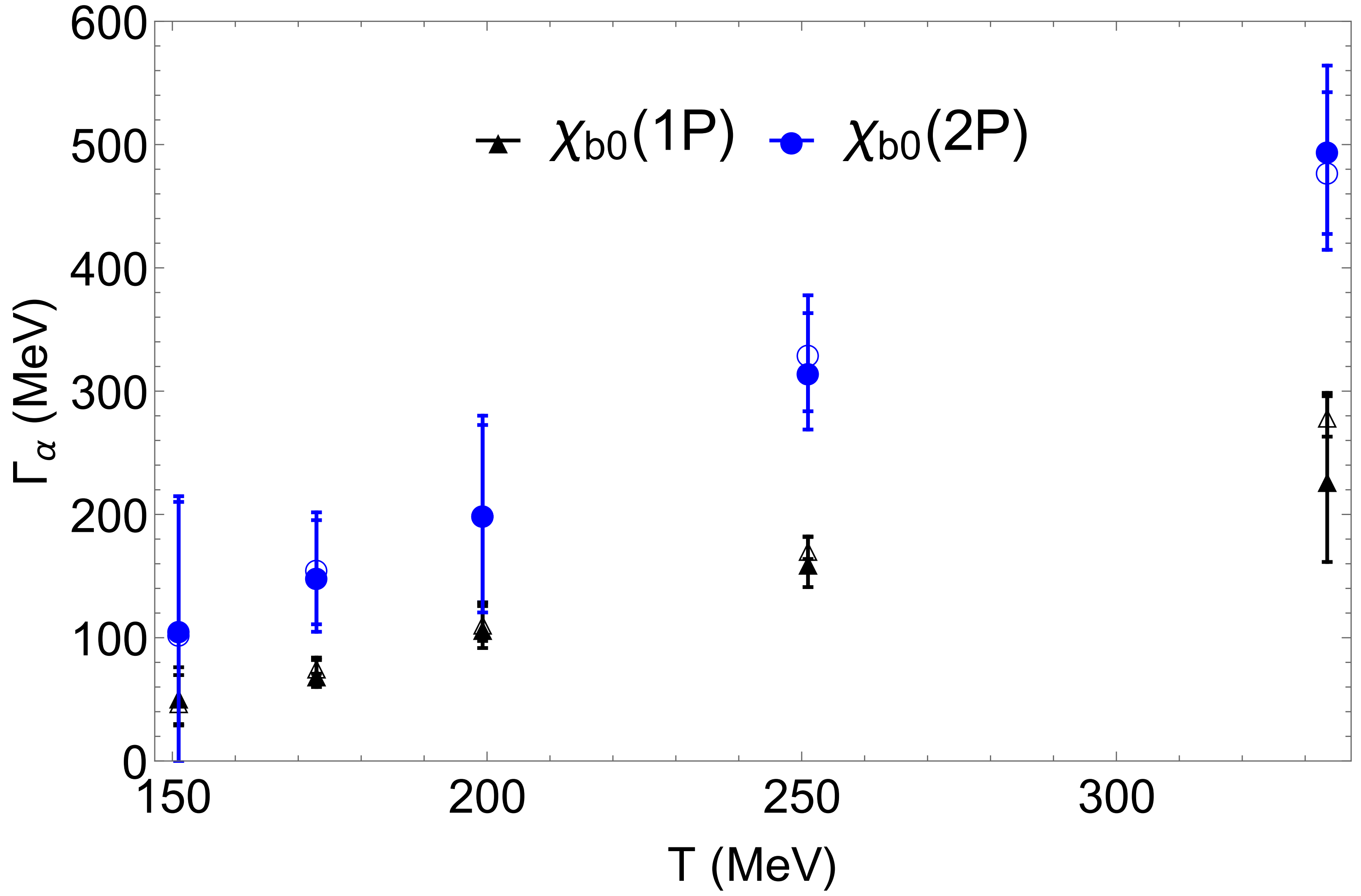
Bottomonium mass and thermal width, lattice QCD with finite m_Q

High excitations (2P, 3S) can survive at $T = 334$ MeV;
Mass - mild temperature dependence;
Thermal width - quantitatively larger.





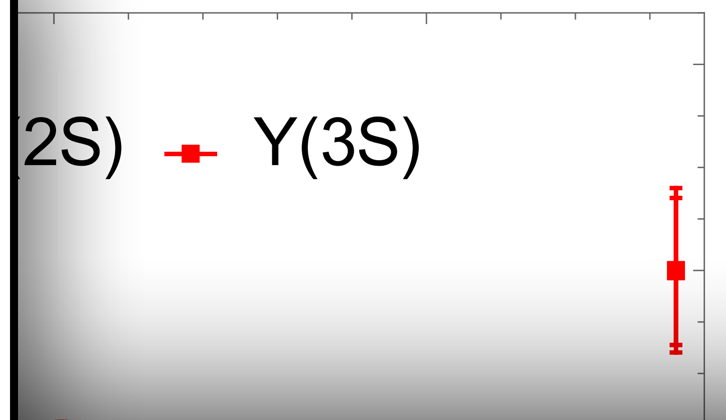


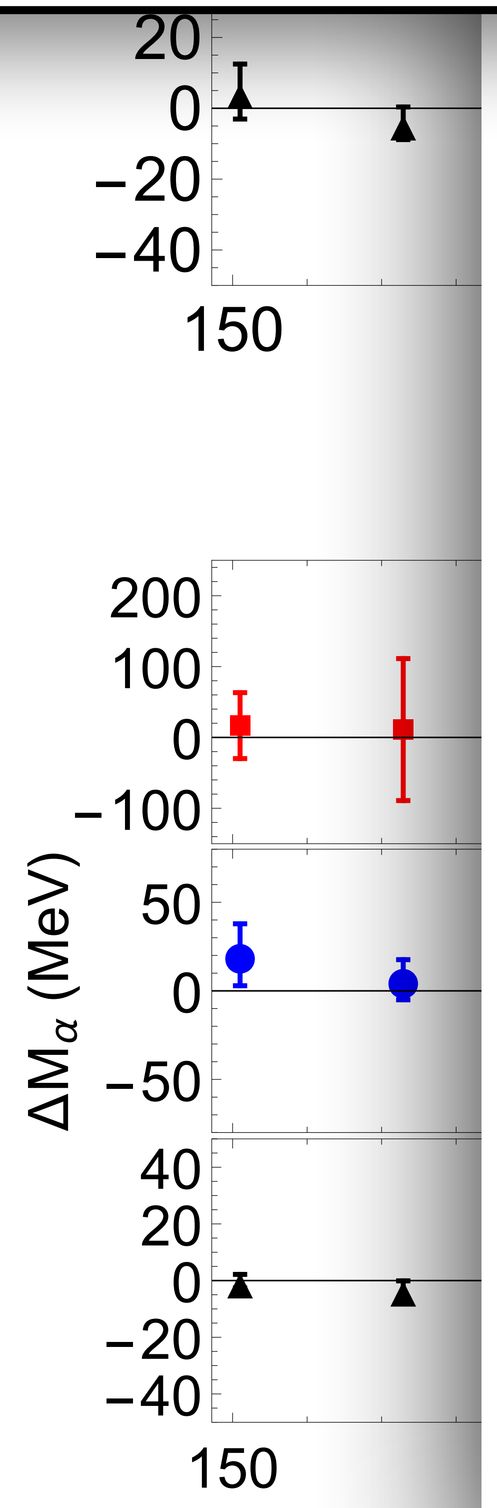


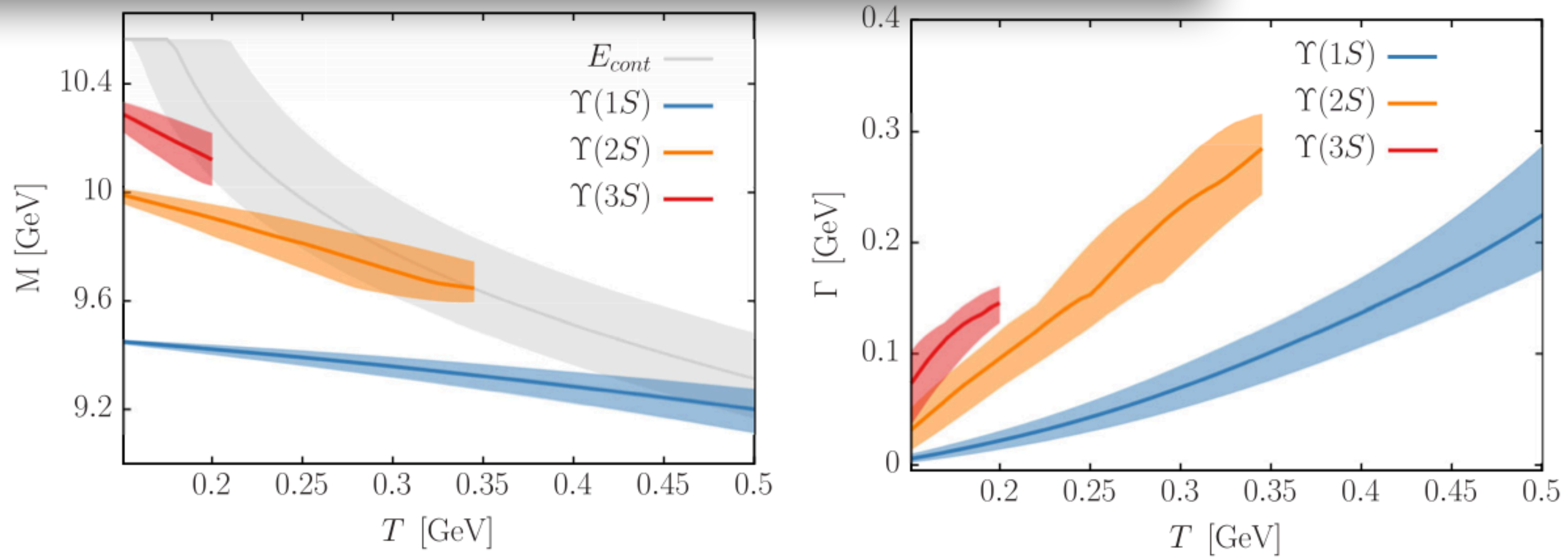
R. Larsen, S. Meinel, S. Mukherjee, and P. Petreczky:
Phys.Rev.D100,074506(2019), Phys.Lett.B800,135119(2020), Phys.Rev.D102,114508(2020)

Bottomonium mass and thermal width, lattice QCD with finite m_Q

High excitations (2P, 3S) can survive at $T = 334$ MeV;
Mass - mild temperature dependence;
Thermal width - quantitatively larger.





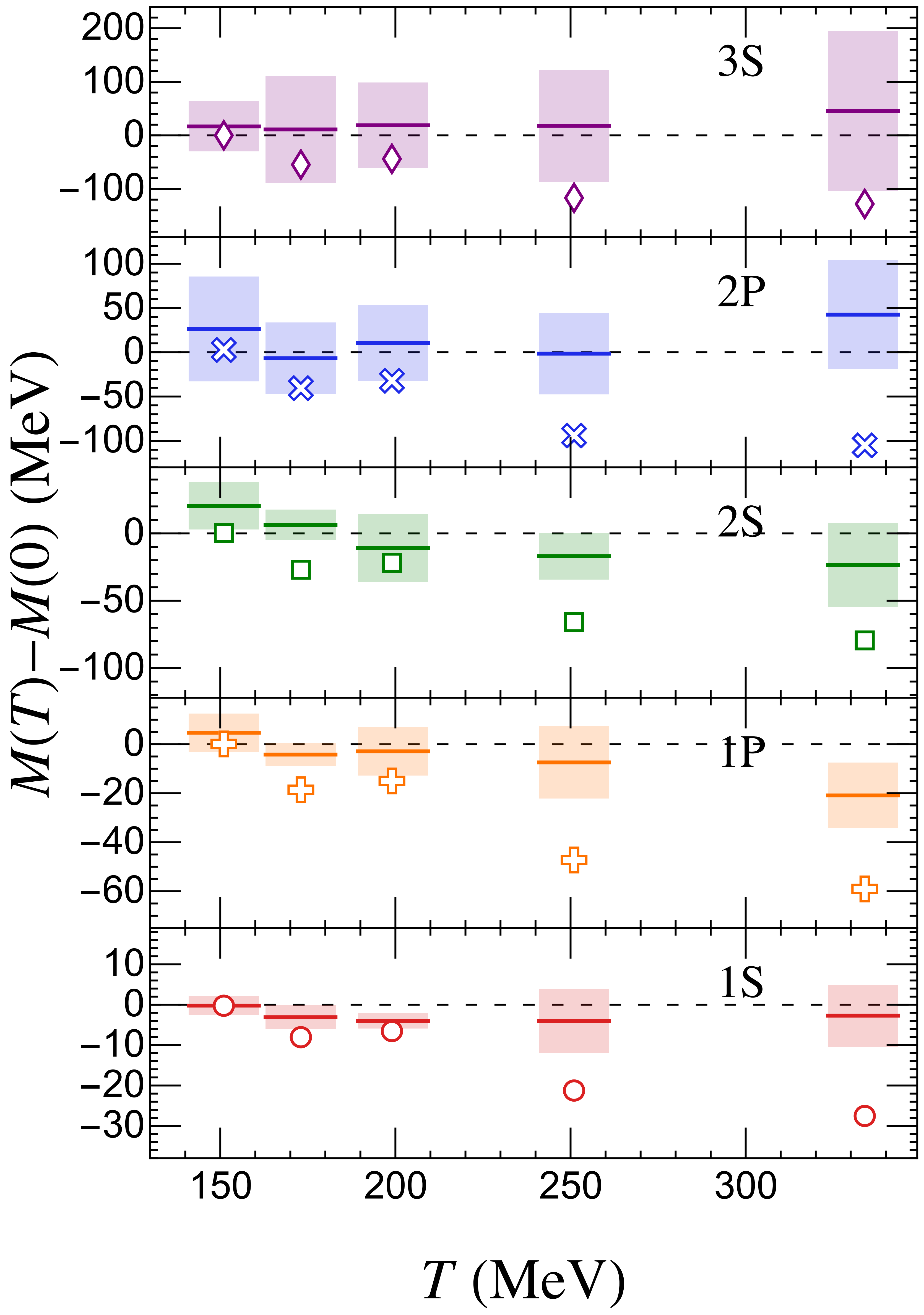


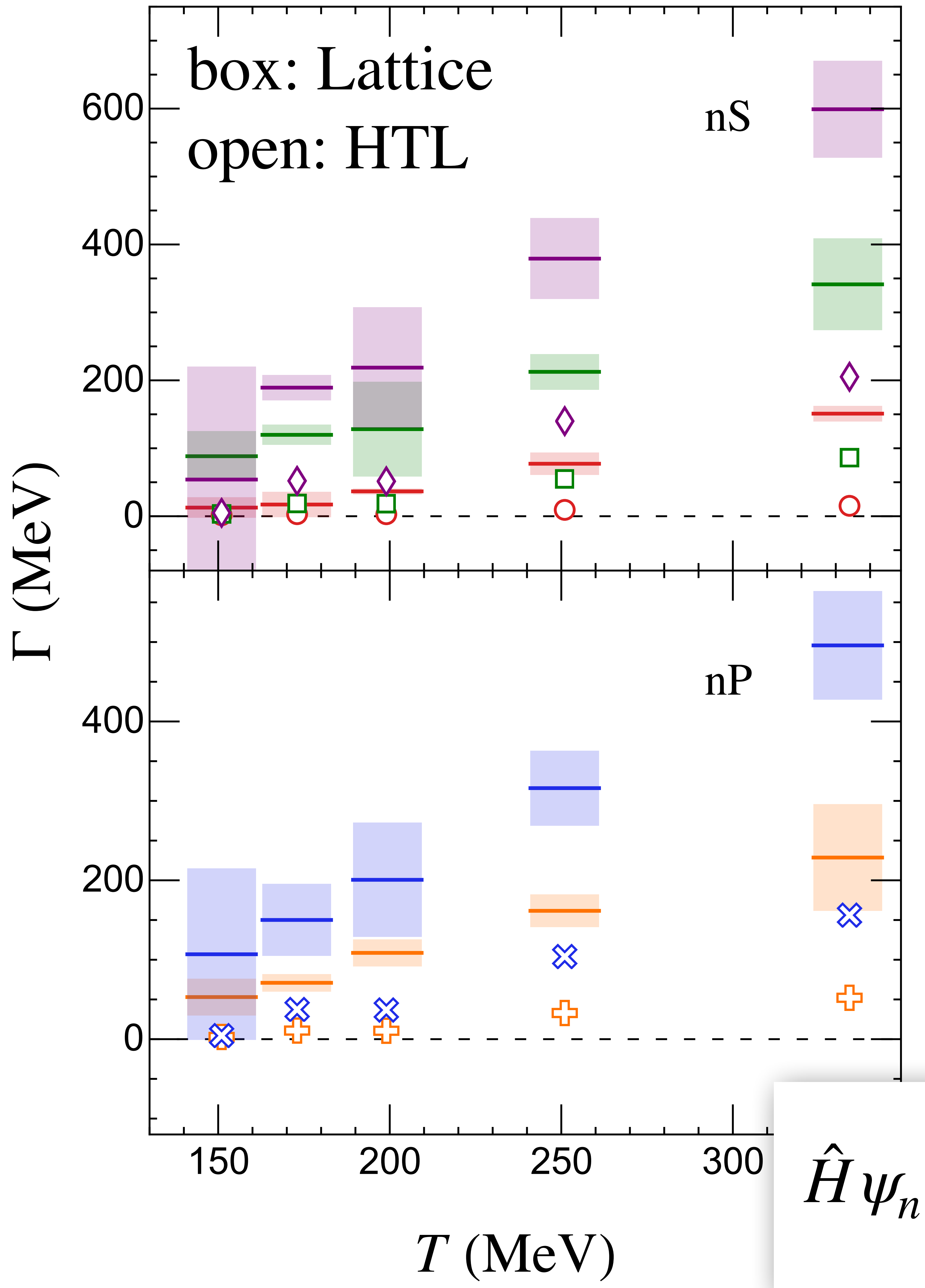
D. Lafferty and A. Rothkopf, PhysRevD.101.056010(2020)

T (MeV)

R. Larsen, S. Meinel, S. Mukherjee, and P. Petreczky:
Phys.Rev.D100,074506(2019), Phys.Lett.B800,135119(2020), Phys.Rev.D102,114508(2020)

Can we understand the new lattice result using Hard Thermal Loop potential?



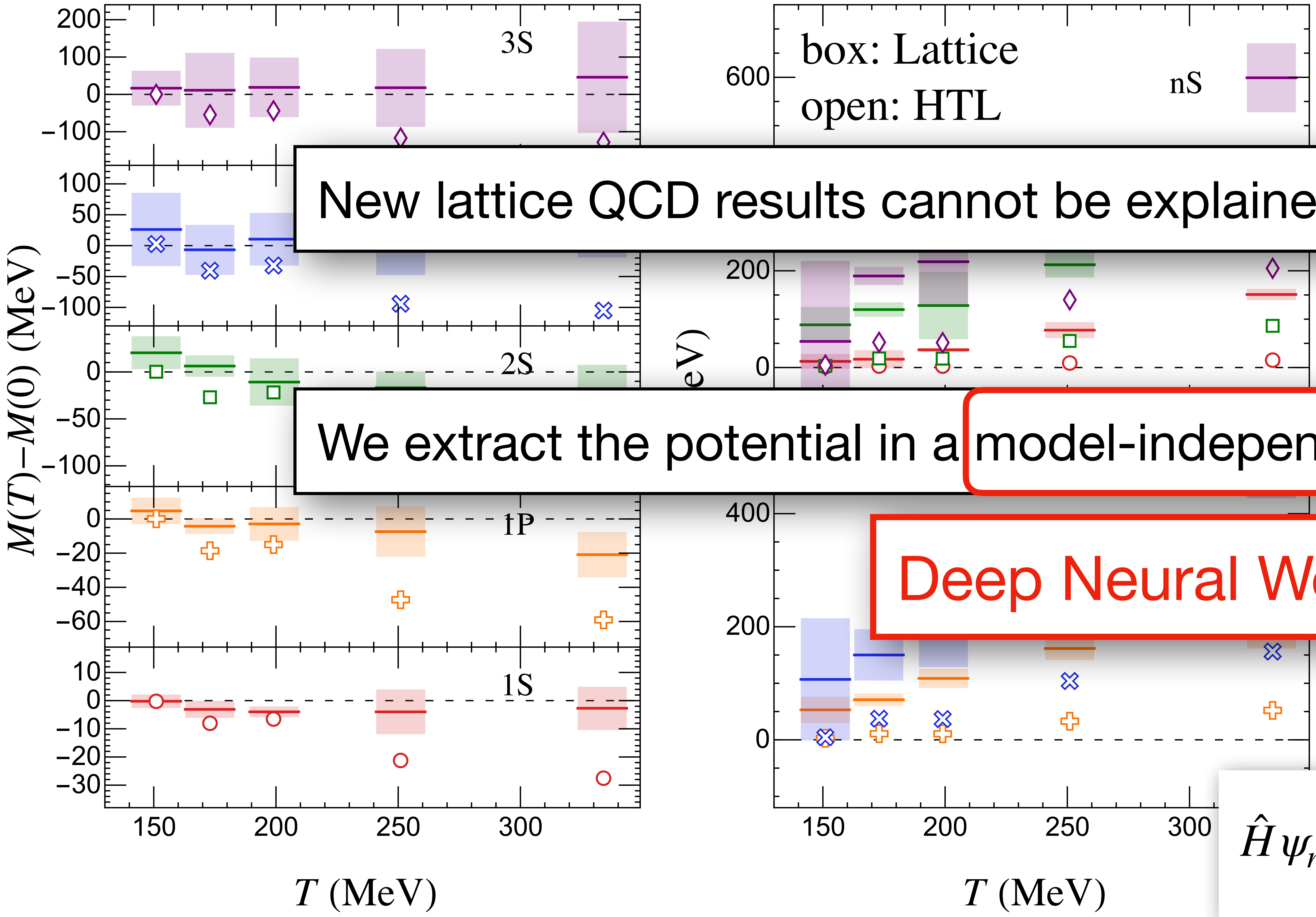


$m_D(T)$ fitted by LQCD

$$V(T,r) = V_R(T,r) + i\, V_I(T,r)$$

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

Can we understand the new lattice result using Hard Thermal Loop potential?



New lattice QCD results cannot be explained by the HTL potential

We extract the potential in a model-independent way

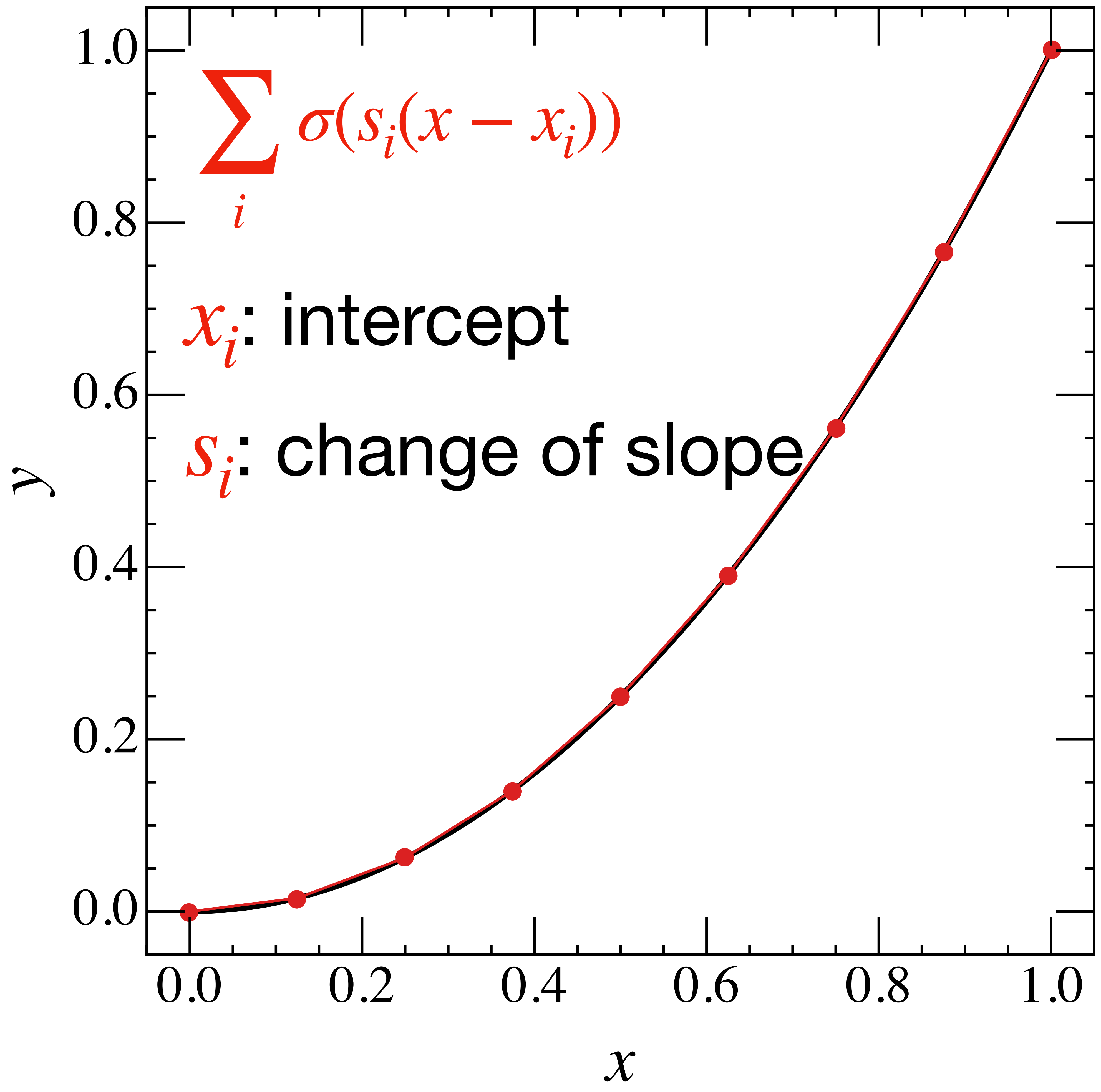
Deep Neural Works (DNN)

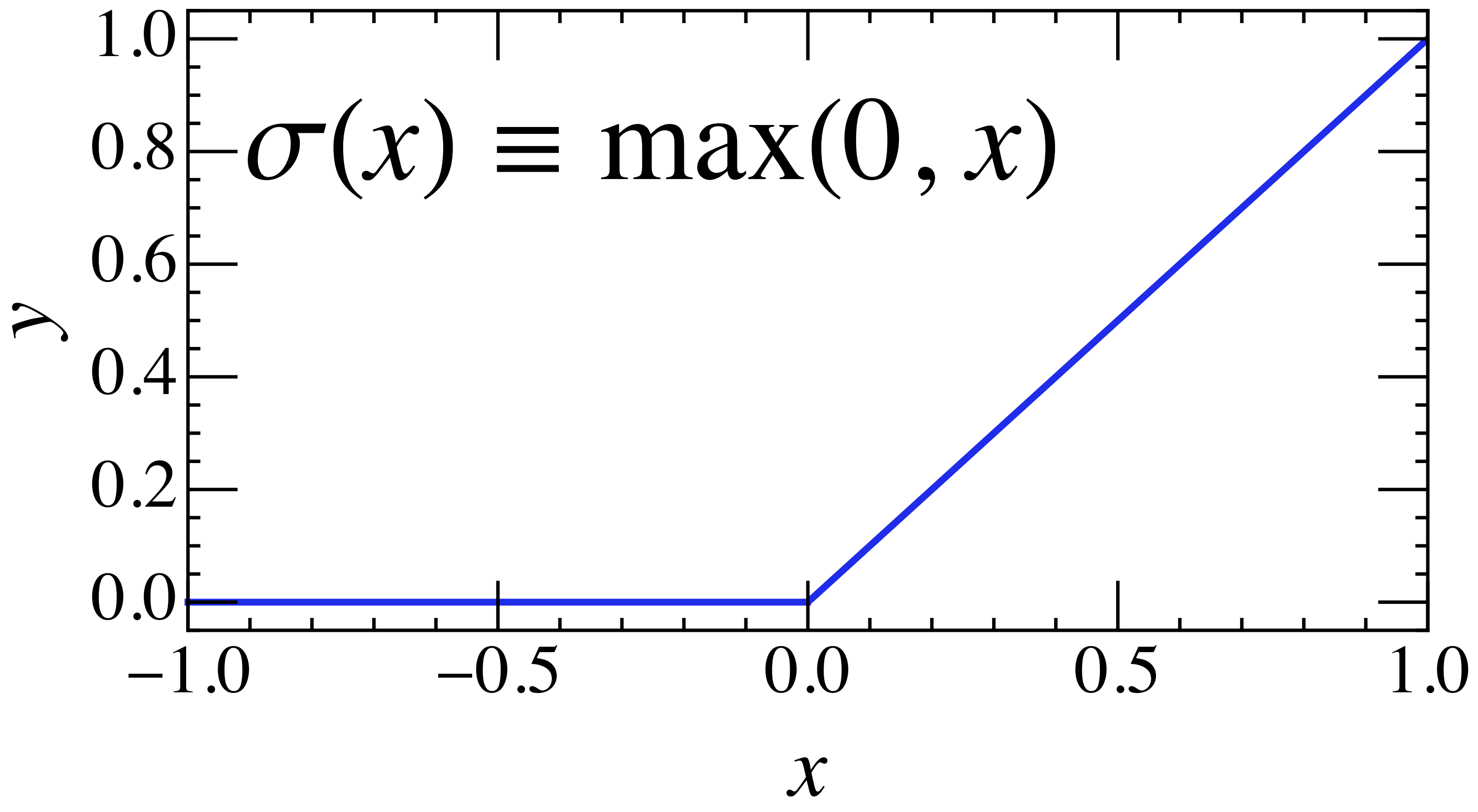
$$V(T,r) = V_R(T,r) + i\, V_I(T,r)$$

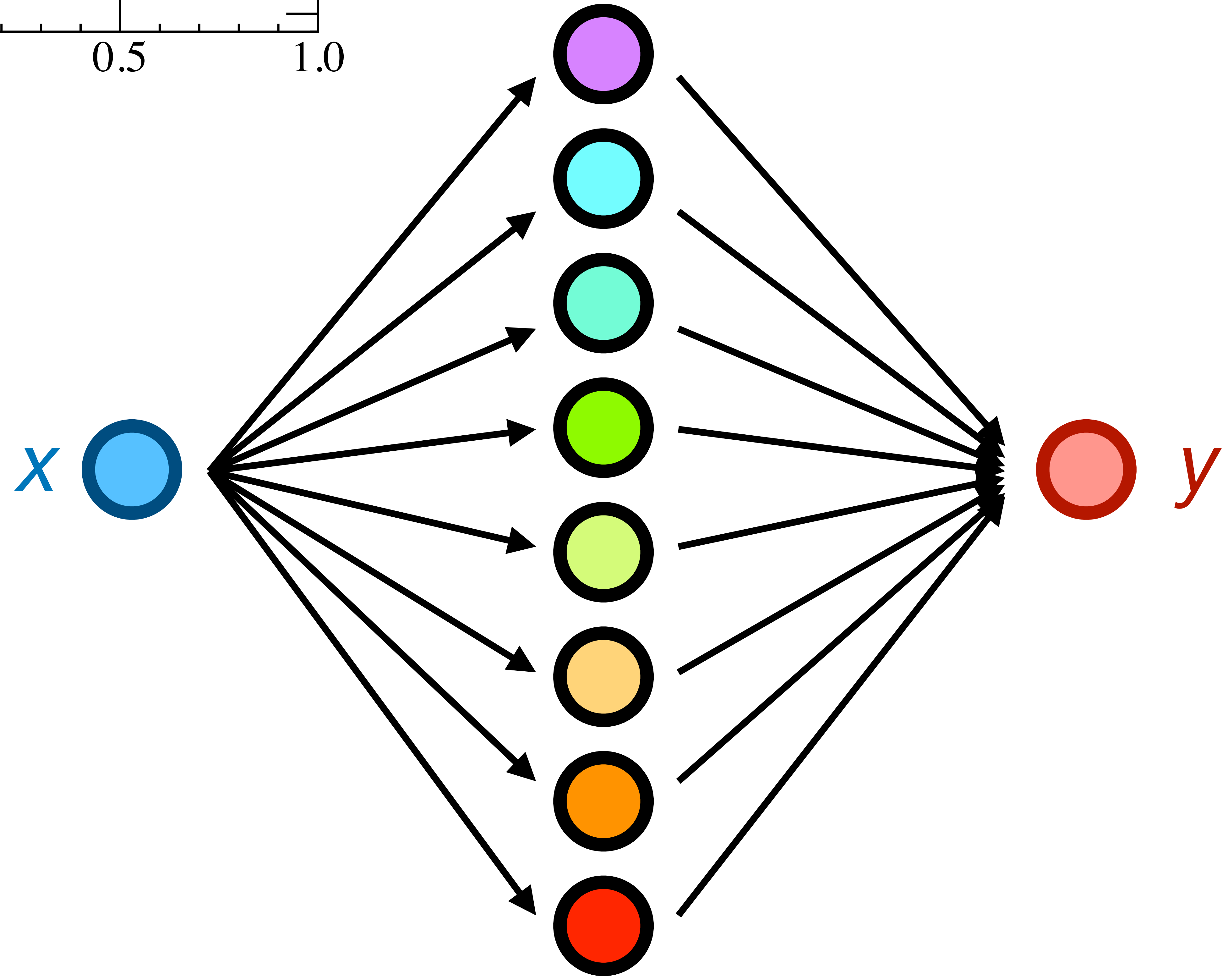
$$\hat{H}\psi_n = -\frac{\nabla^2}{2m_\mu}\psi_n + V(r)\psi_n = E_n\psi_n$$

What are Deep Neural Networks?

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



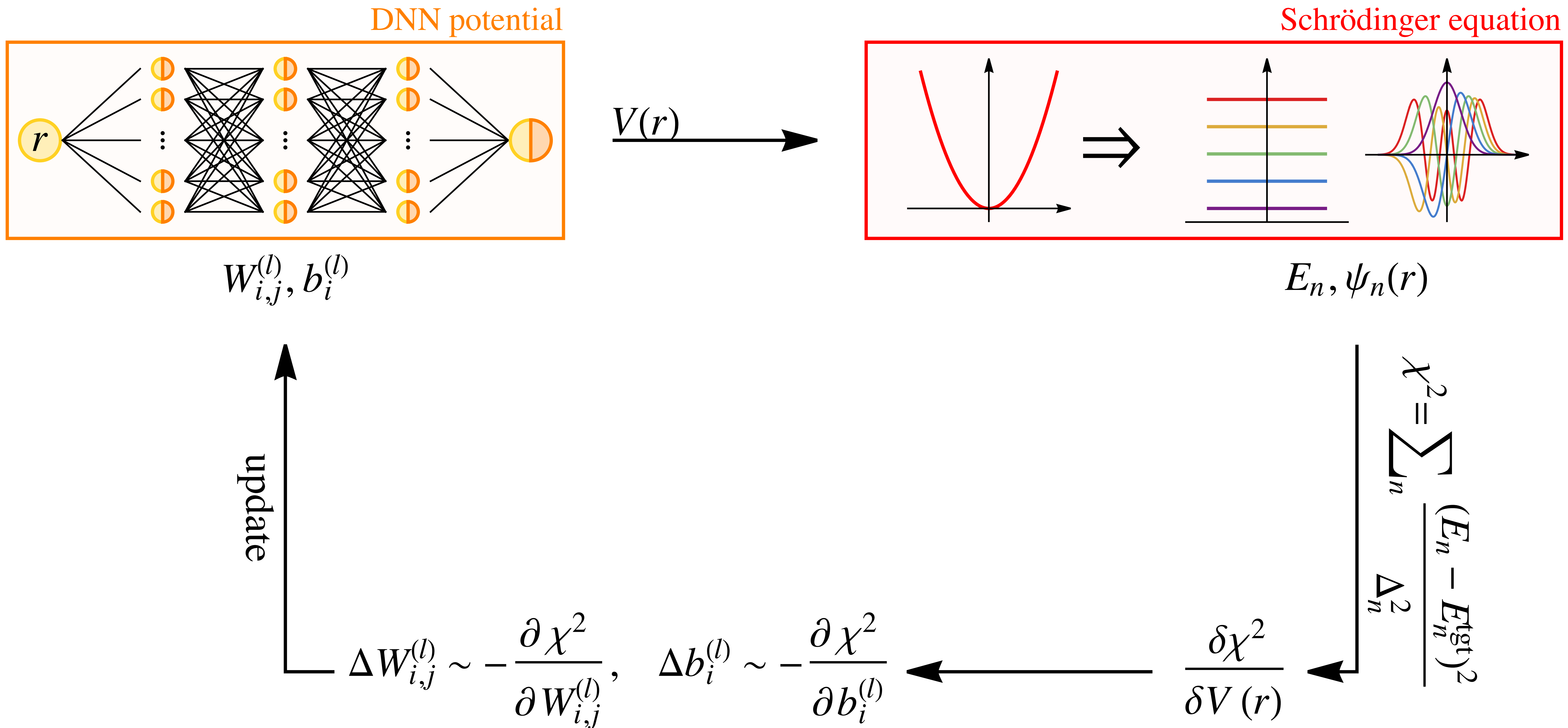




each ● represents one of $\sigma(s_i(x - x_i))$

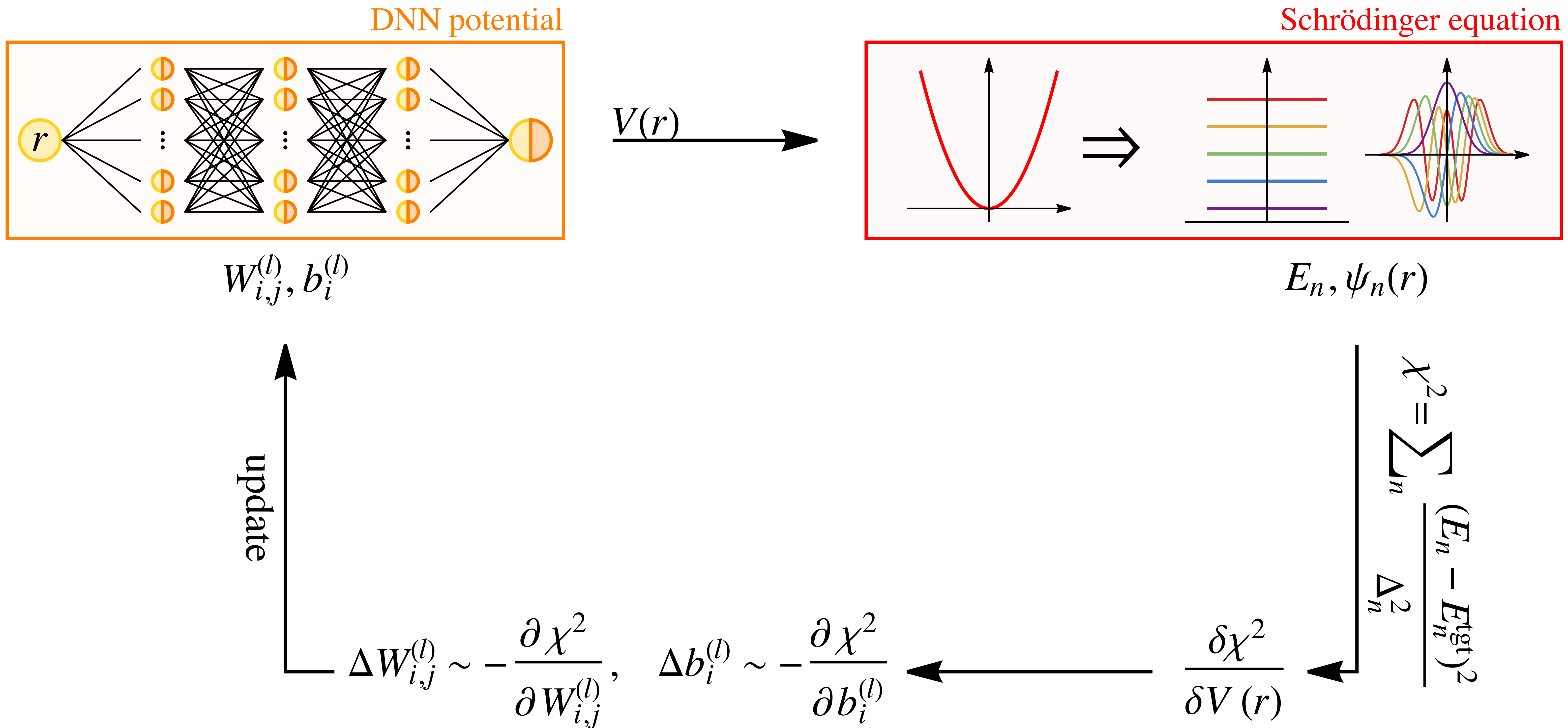
How to learn $V(r)$ from $\{E_n\}$?

- parameterize the potential $V(r\,|\,\boldsymbol{\theta})$, minimize $\chi^2 \equiv \sum_i \left(\frac{E_{\boldsymbol{\theta},i} - E_i}{\delta E_i}\right)^2$

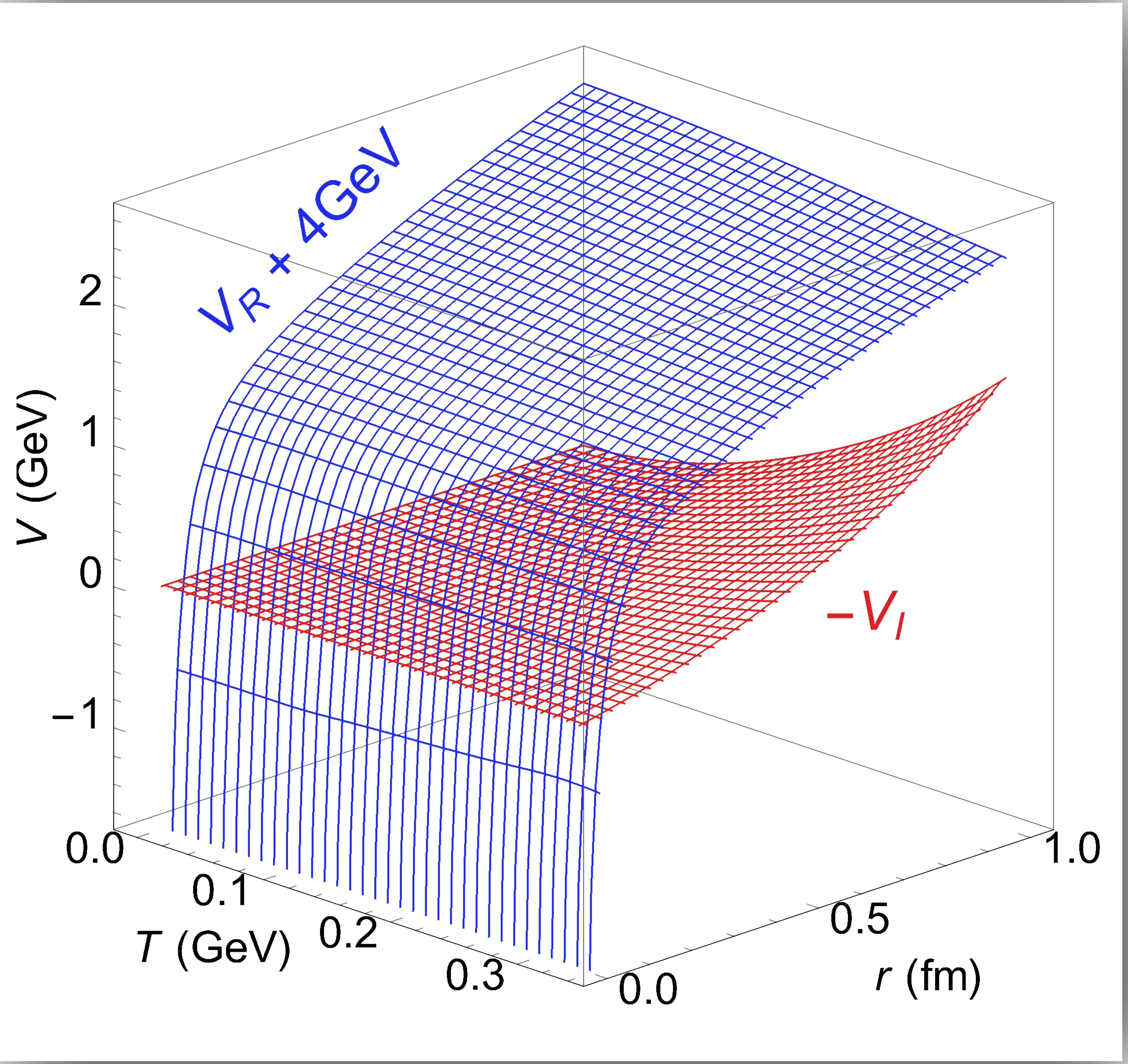


How to learn $V(r)$ from $\{E_n\}$?

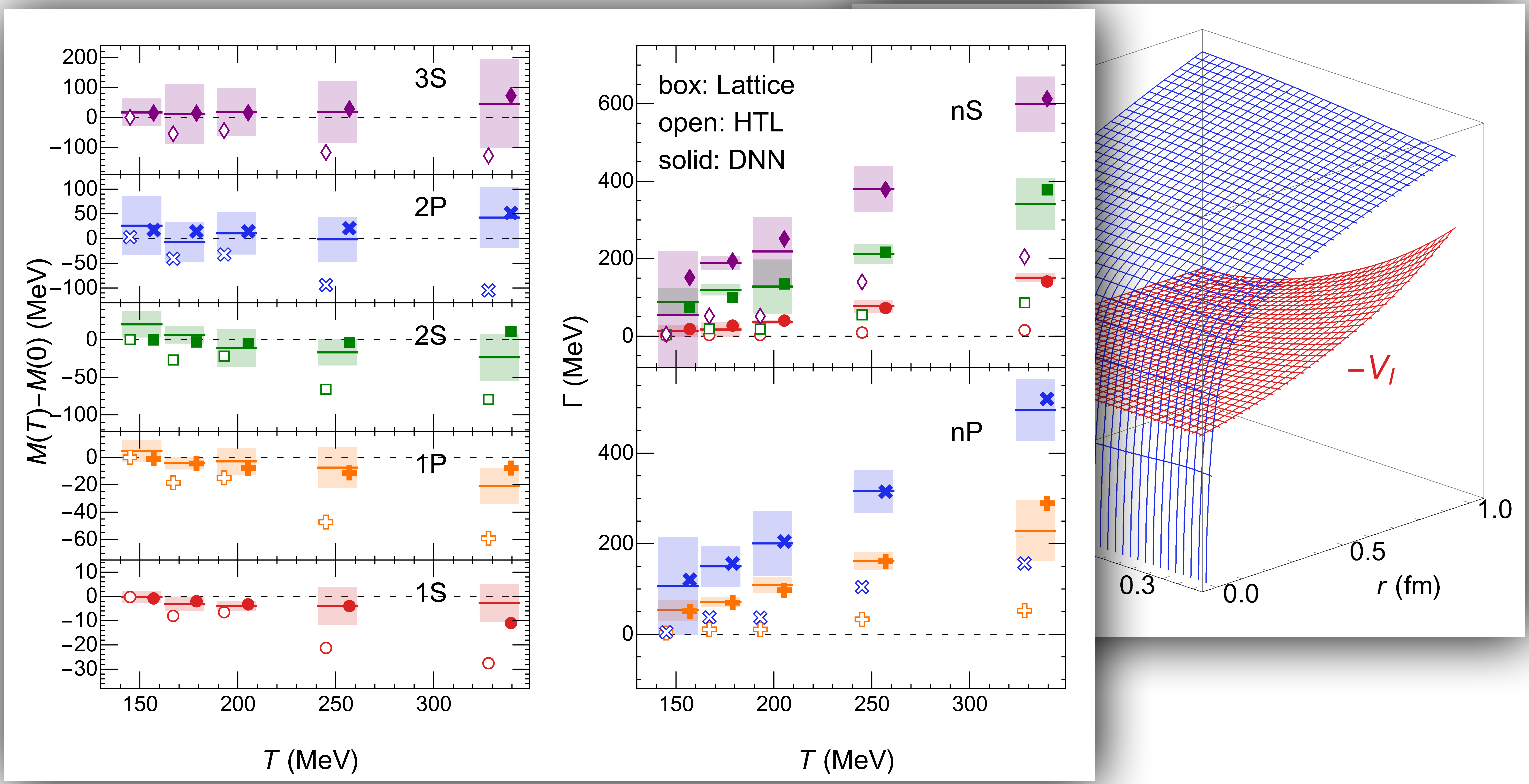
- parameterize the potential $V(r|\boldsymbol{\theta})$, minimize $\chi^2 \equiv \sum_i \left(\frac{E_{\boldsymbol{\theta},i} - E_i}{\delta E_i}\right)^2$
- a ***gradient-descent*** based method:
    - goal -- find the $\boldsymbol{\theta}$-point that $\nabla_{\boldsymbol{\theta}}\chi^2 = 0$
    - update $\boldsymbol{\theta}$ iteratively according to $\Delta\boldsymbol{\theta} \propto \nabla_{\boldsymbol{\theta}}\chi^2$



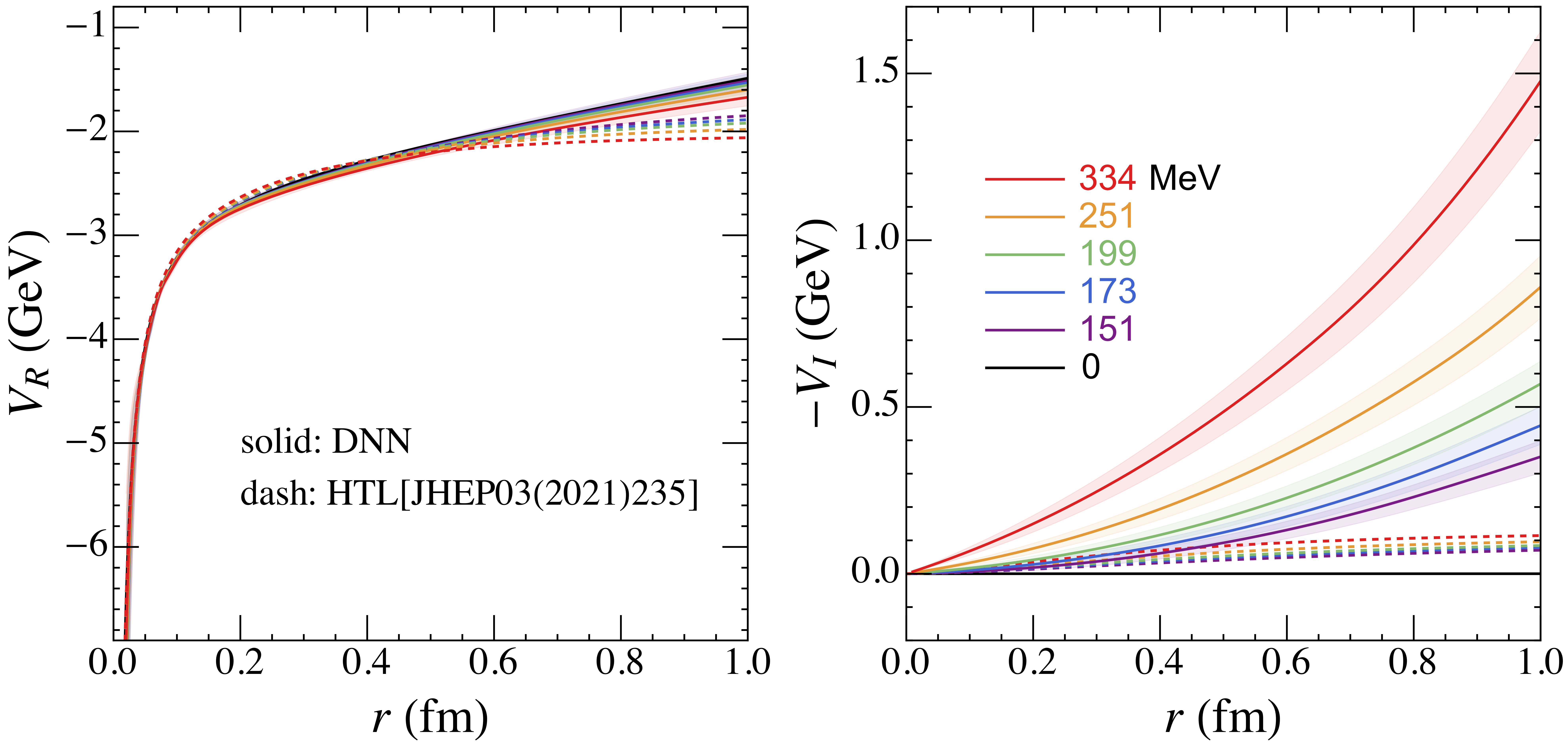
Finite Temperature Heavy-Quark Potential



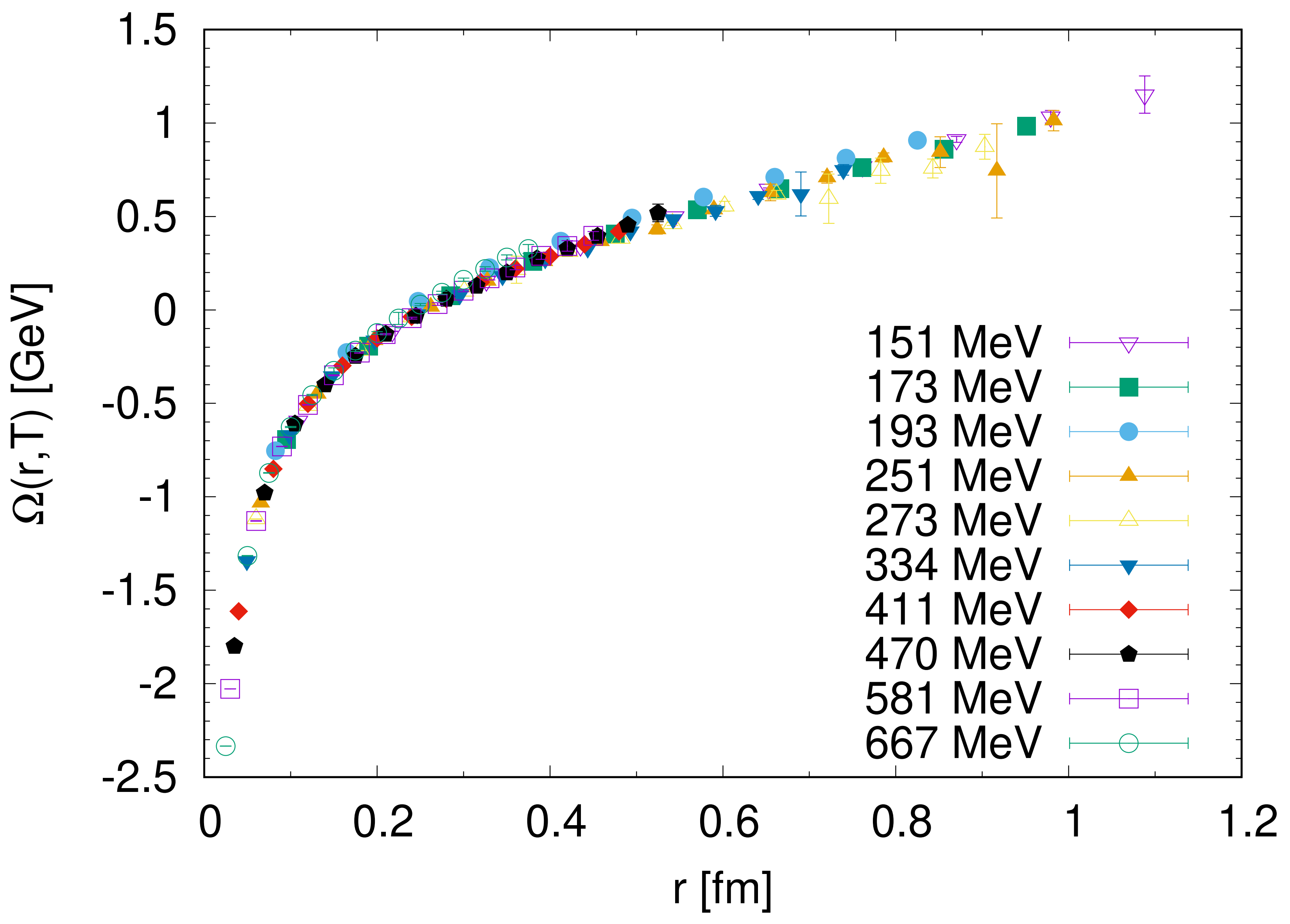
Finite Temperature Heavy-Quark Potential

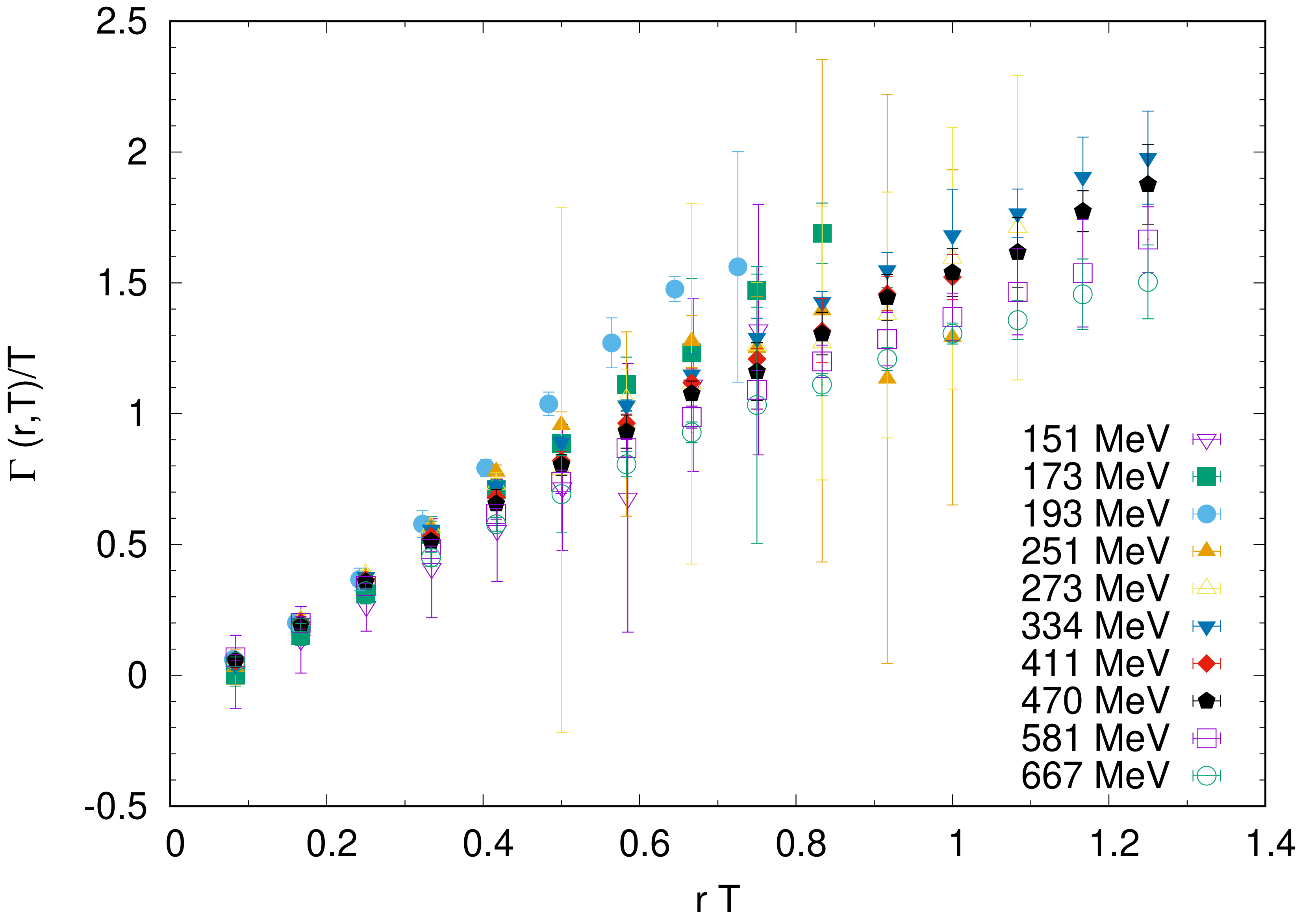


What physics we have learned from $V_{\text{DNN}}(T, r)$?



Reason of the difference?





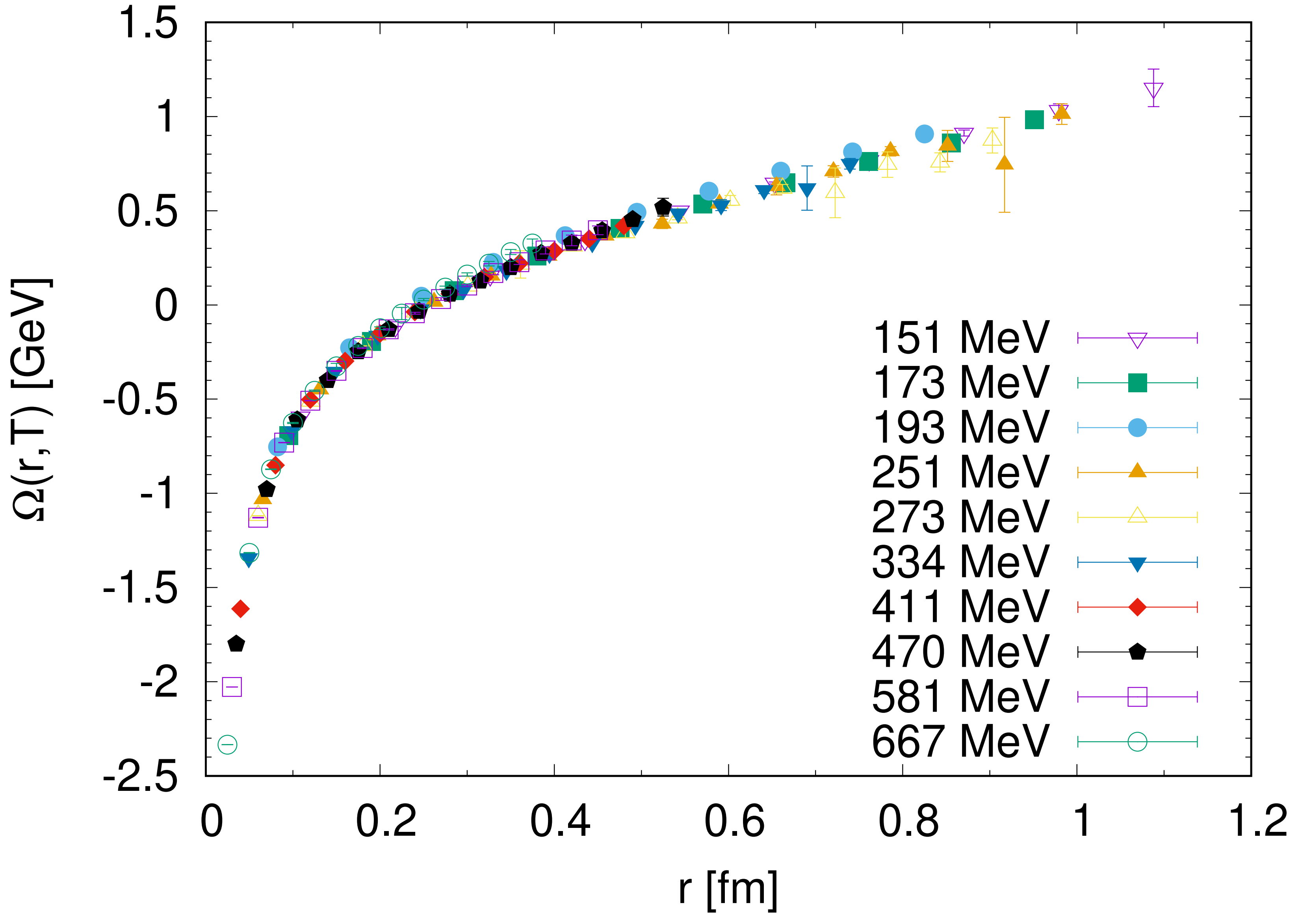
PHYSICAL REVIEW D **105,** 054513 (2022)

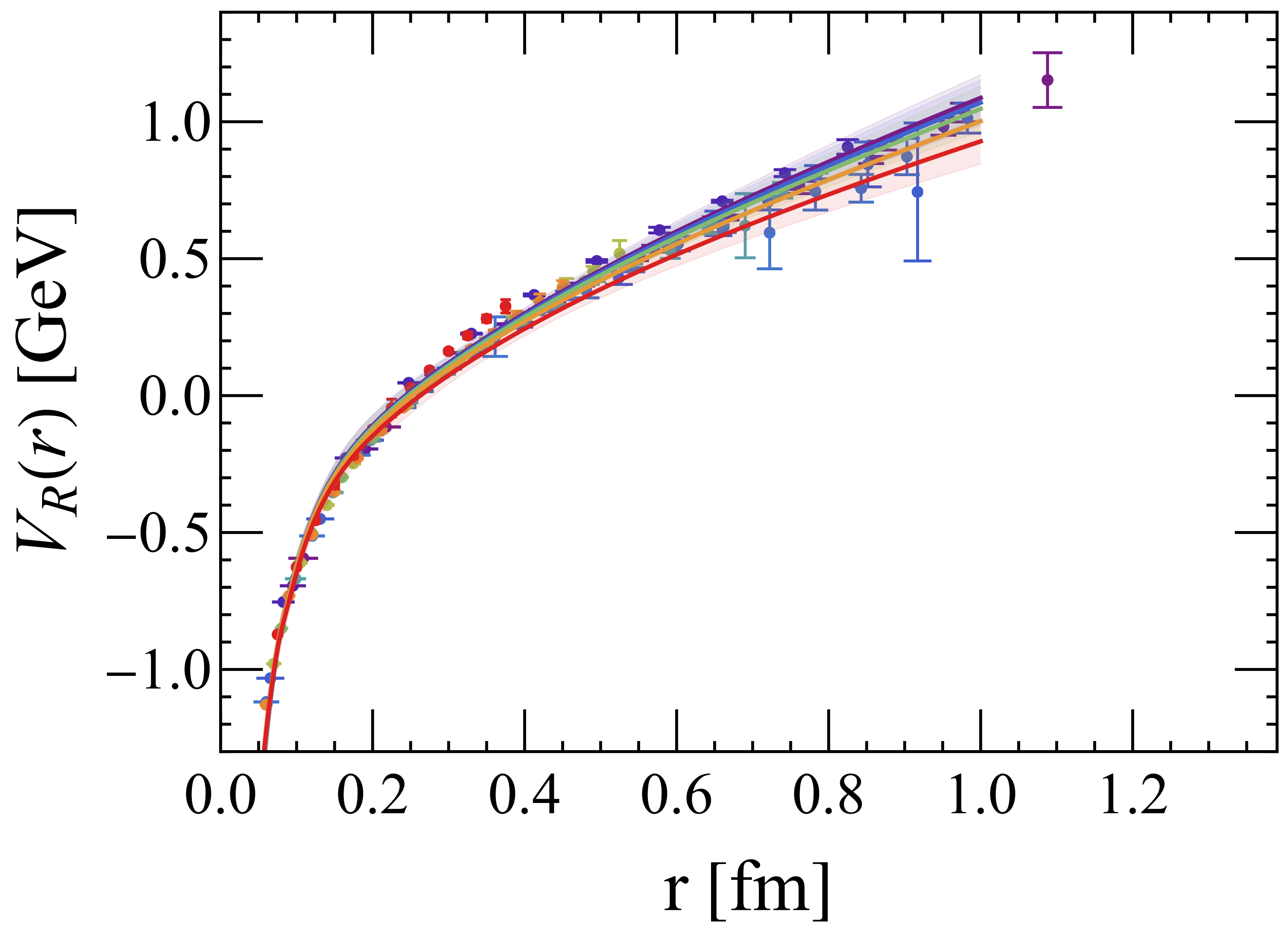
Static quark-antiquark interactions at nonzero temperature from lattice QCD

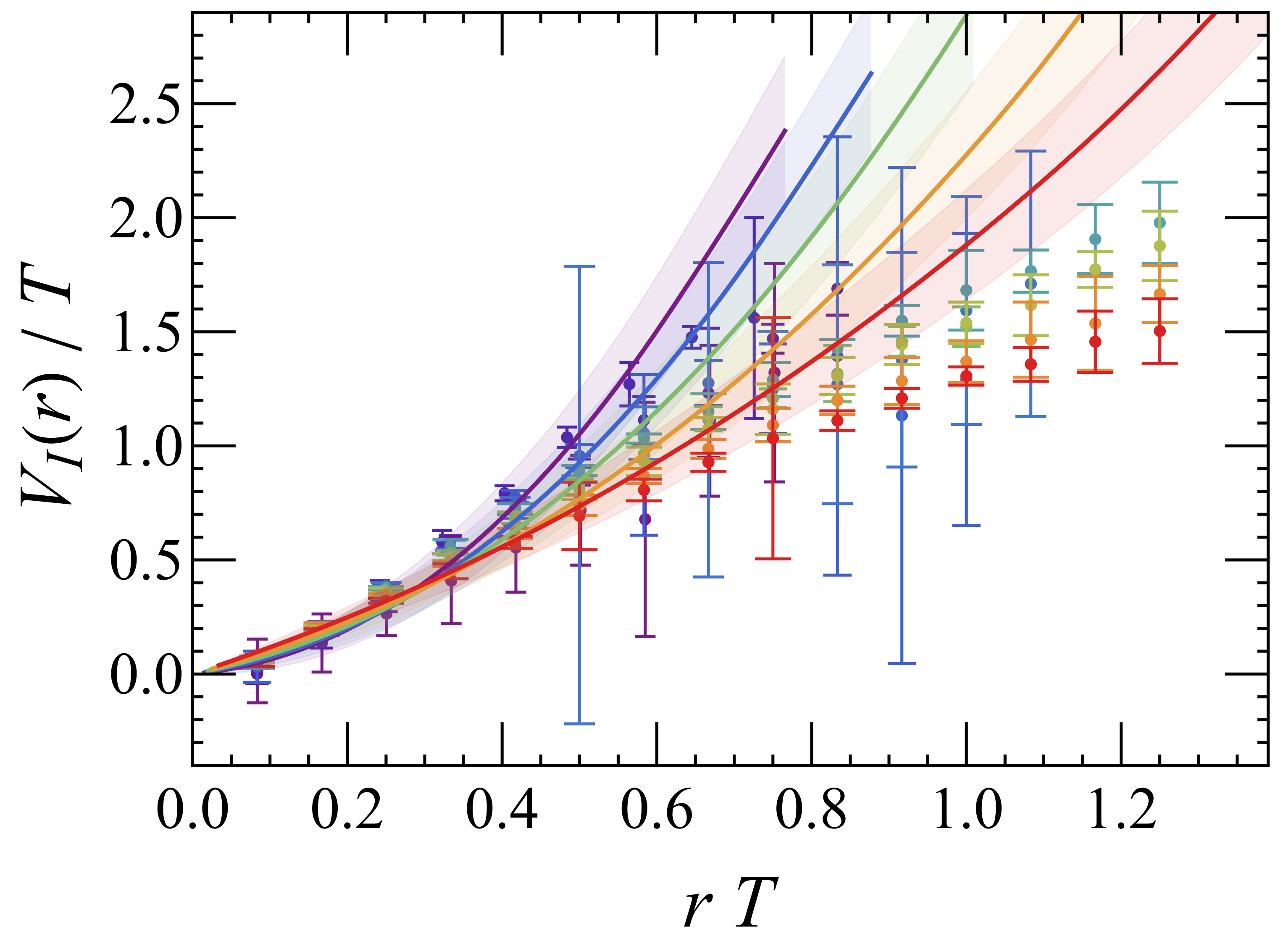
Dibyendu Bala,[1] Olaf Kaczmarek,[1] Rasmus Larsen,[2] Swagato Mukherjee,[3] Gaurang Parkar,[2] Peter Petreczky,[3] Alexander Rothkopf,[2] and Johannes Heinrich Weber[4]

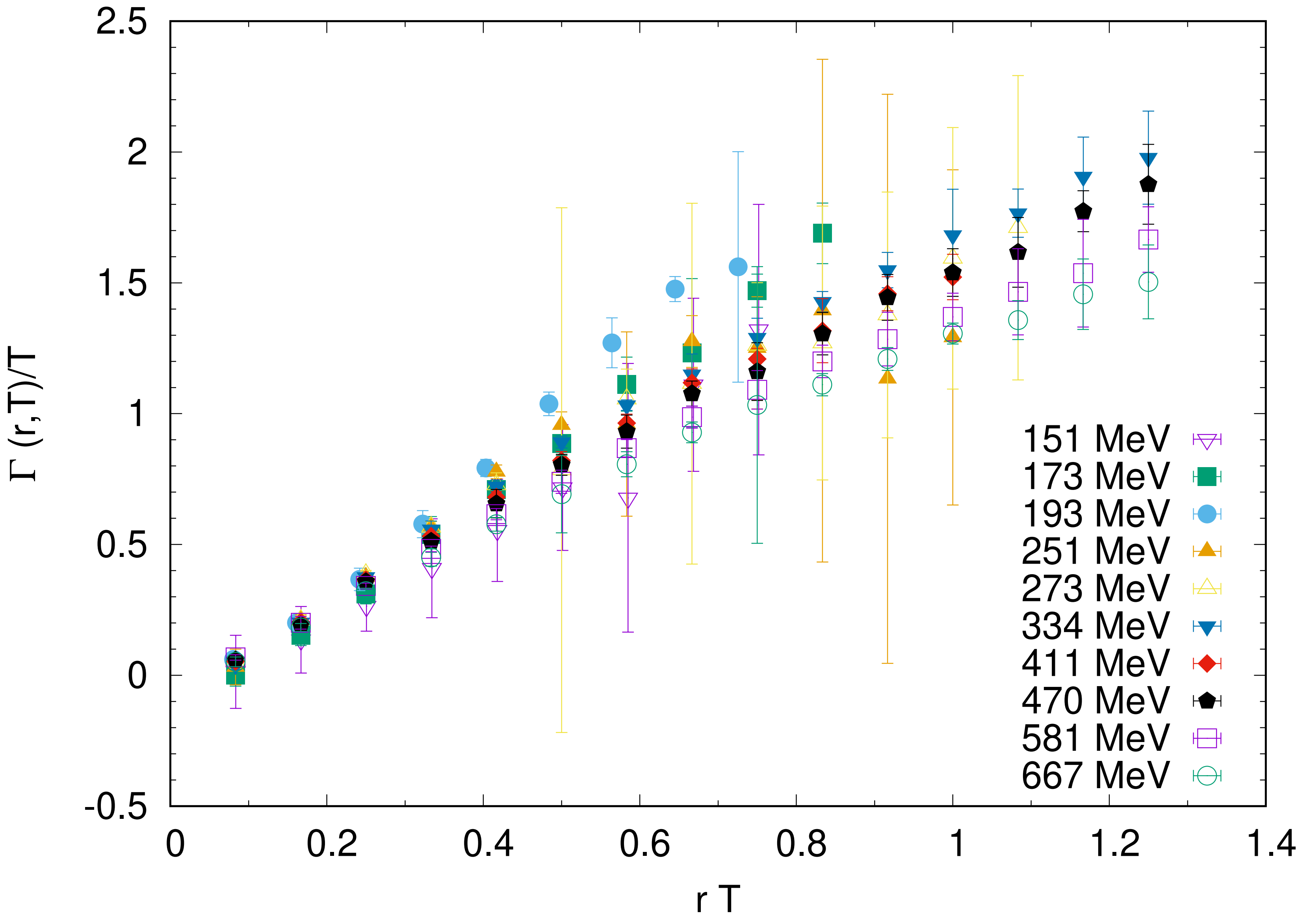
(HotQCD Collaboration)

Reason of the difference?









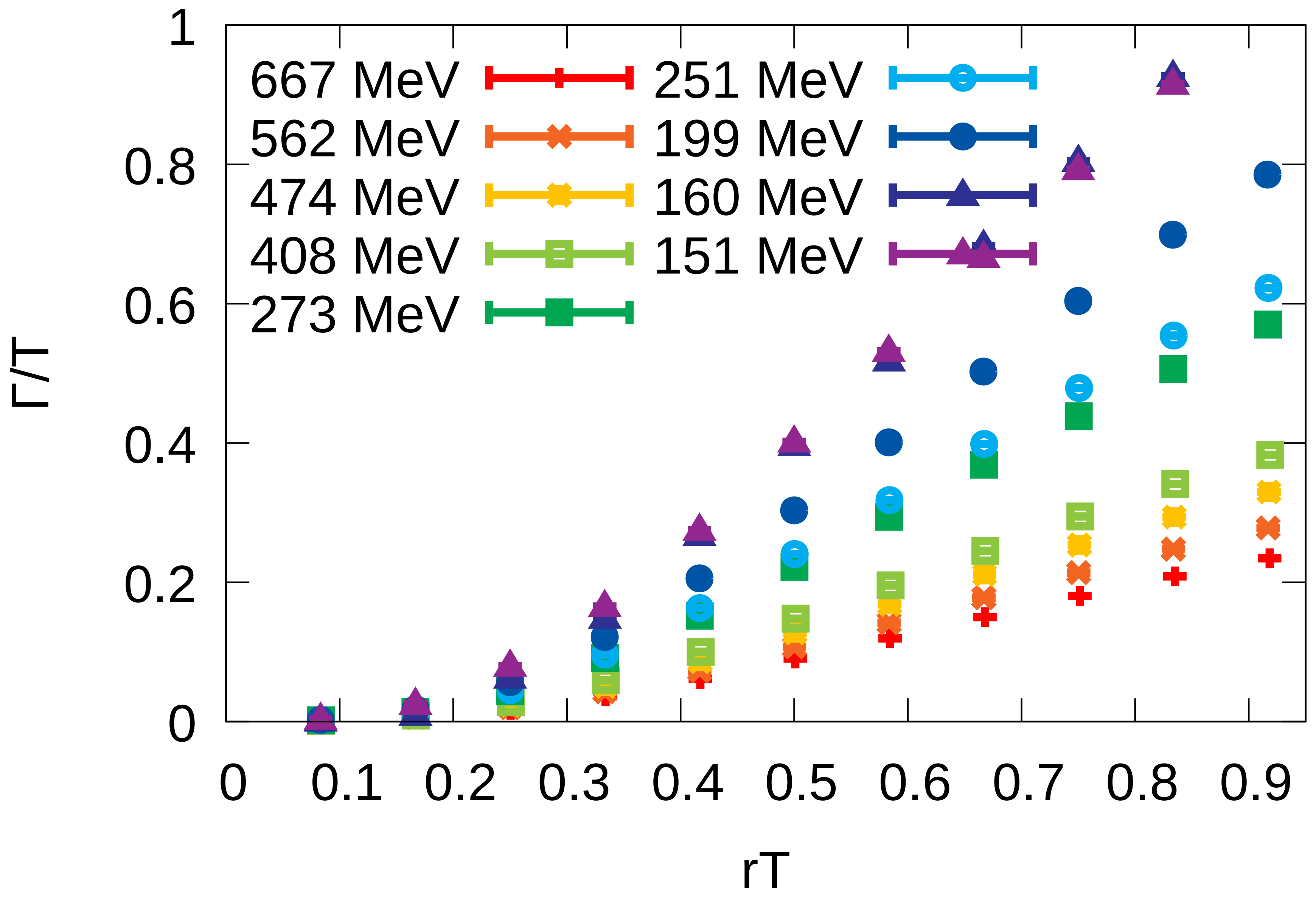
PHYSICAL REVIEW D **105,** 054513 (2022)

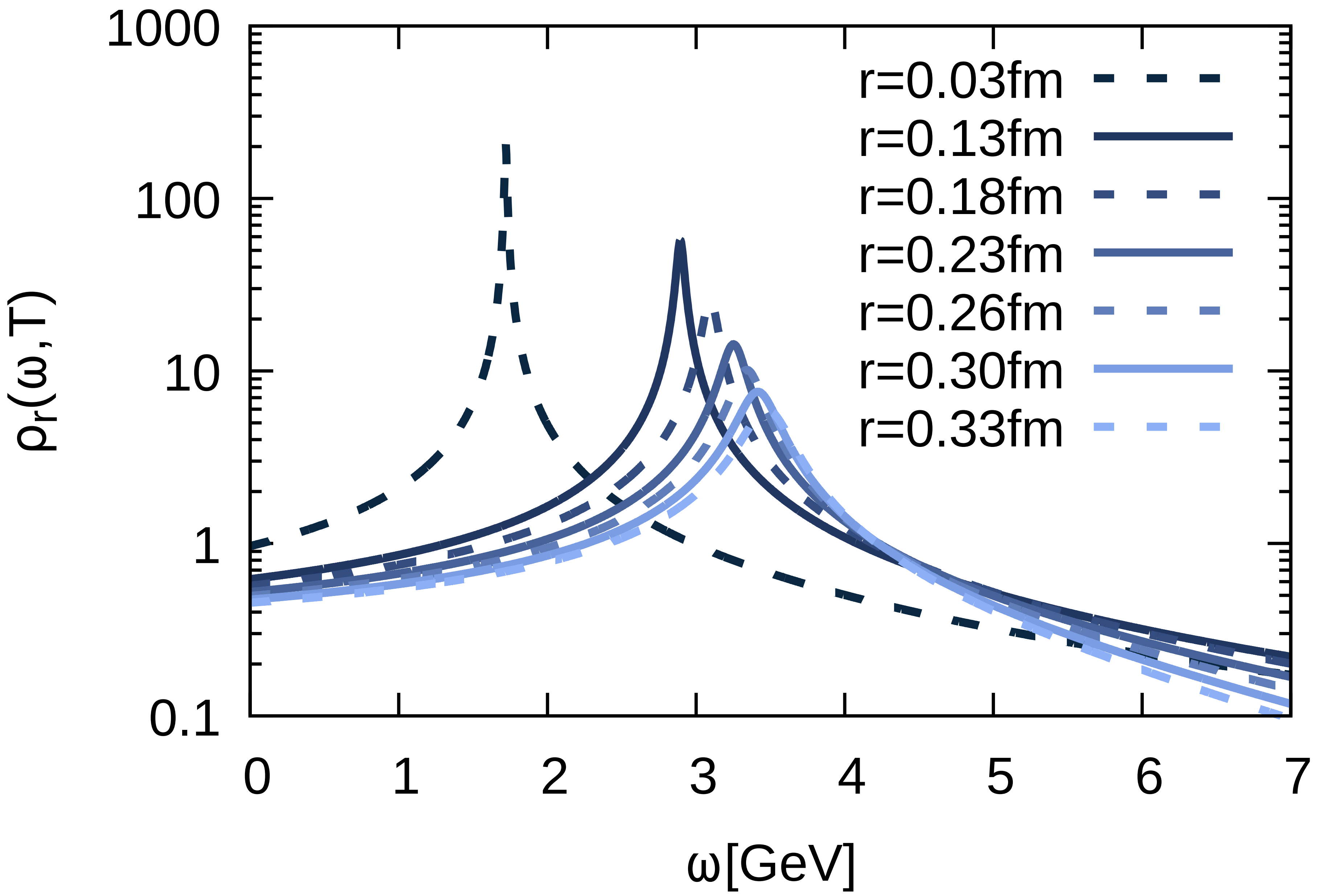
Static quark-antiquark interactions at nonzero temperature from lattice QCD

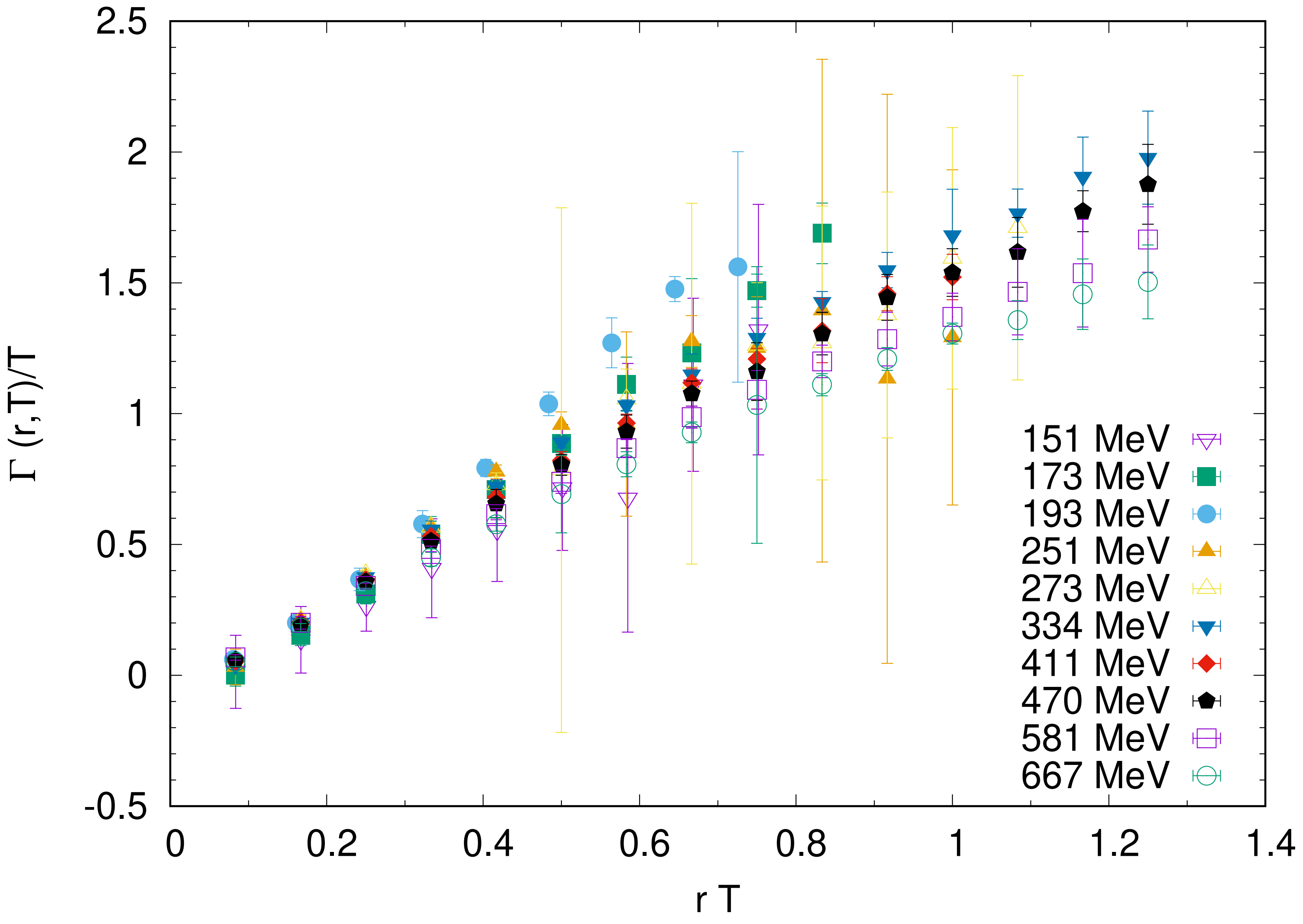
Dibyendu Bala,[1] Olaf Kaczmarek,[1] Rasmus Larsen,[2] Swagato Mukherjee,[3] Gaurang Parkar,[2] Peter Petreczky,[3] Alexander Rothkopf,[2] and Johannes Heinrich Weber[4]

(HotQCD Collaboration)

Reason of the difference?







PHYSICAL REVIEW D **105,** 054513 (2022)

Static quark-antiquark interactions at nonzero temperature from lattice QCD

Dibyendu Bala,[1] Olaf Kaczmarek,[1] Rasmus Larsen,[2] Swagato Mukherjee,[3] Gaurang Parkar,[2] Peter Petreczky,[3] Alexander Rothkopf,[2] and Johannes Heinrich Weber[4]

(HotQCD Collaboration)

Problem # II:

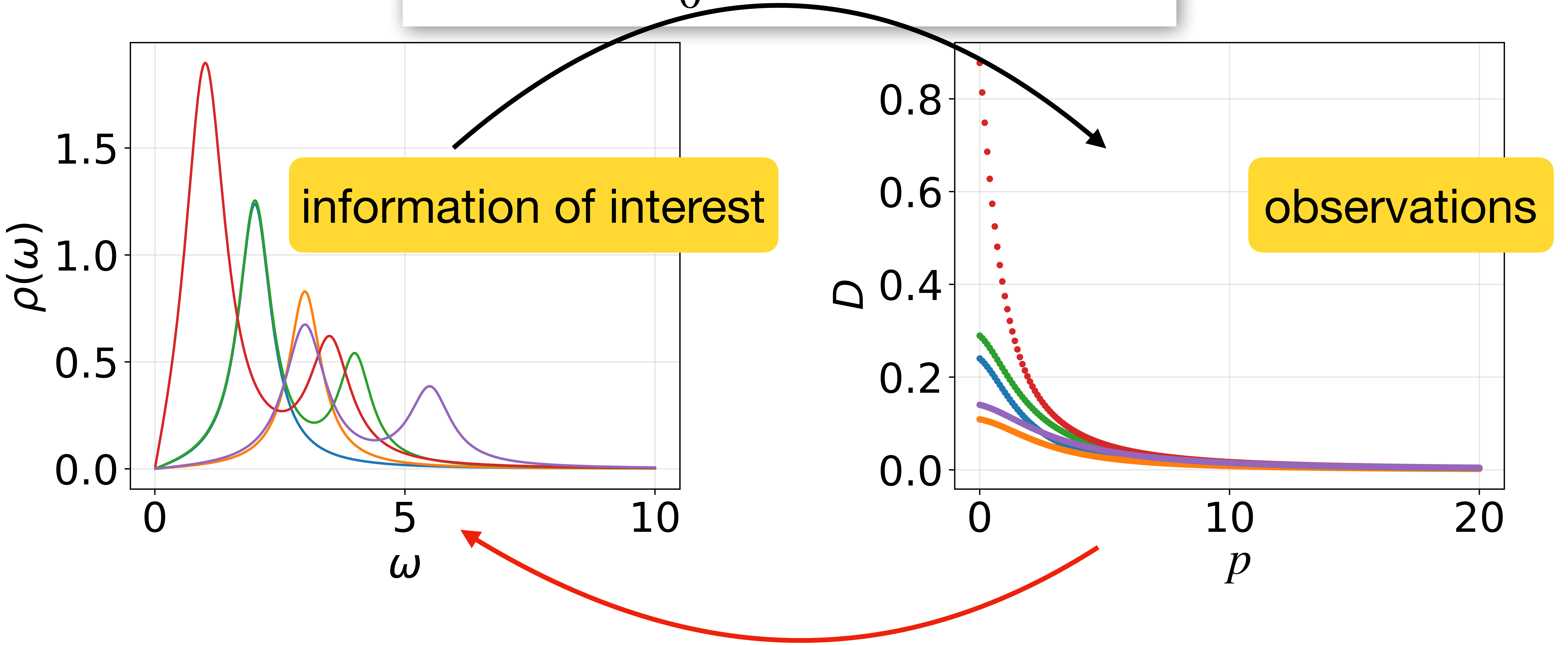
spectral function reconstruction

refs:

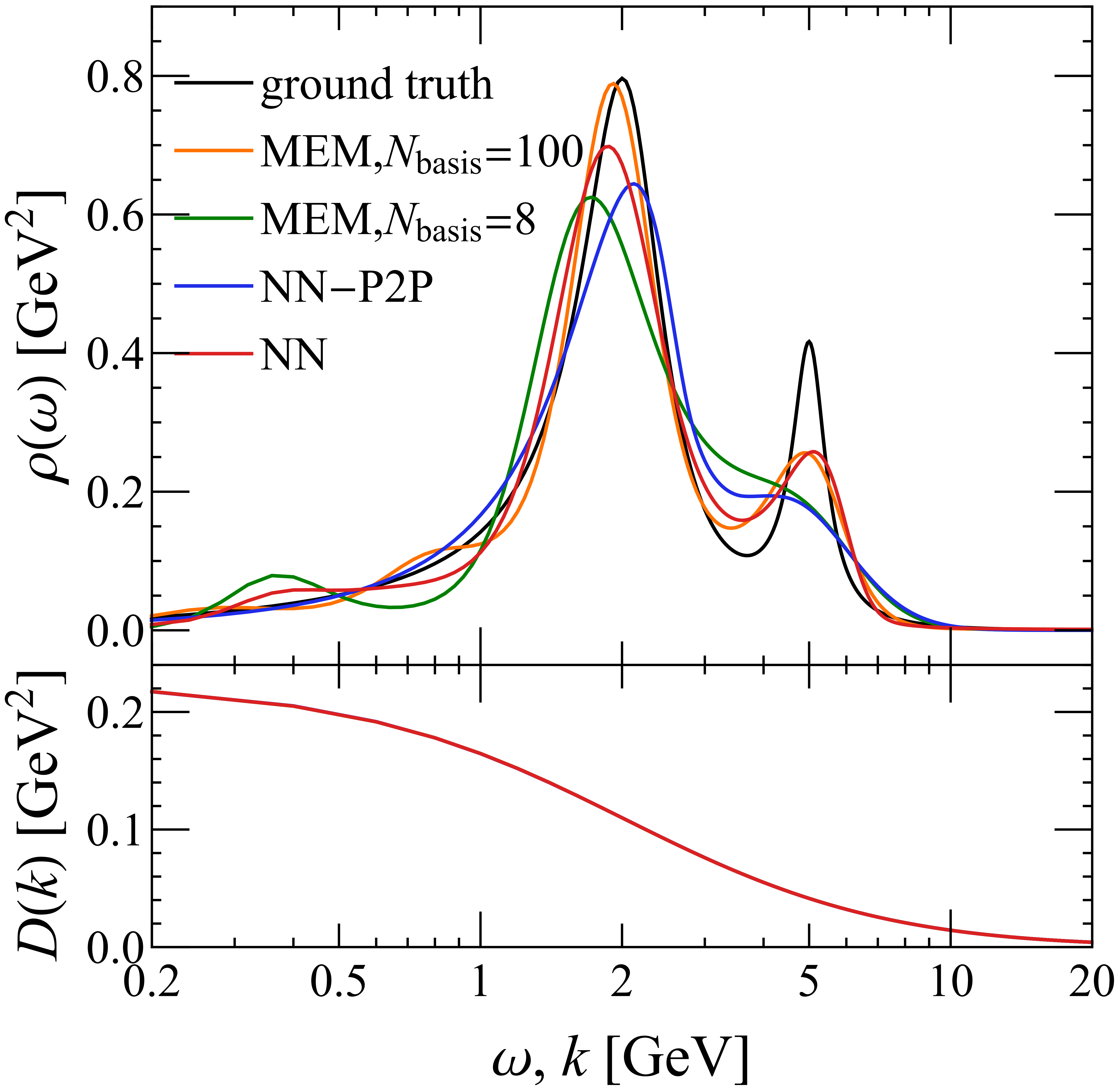
SS, Wang, Zhou, Comput.Phys.Commun. 282 (2023) 108547;
Wang, ***SS***, Zhou, Phys. Rev. D **106**, L051502;

3. spectral function <=> correlation

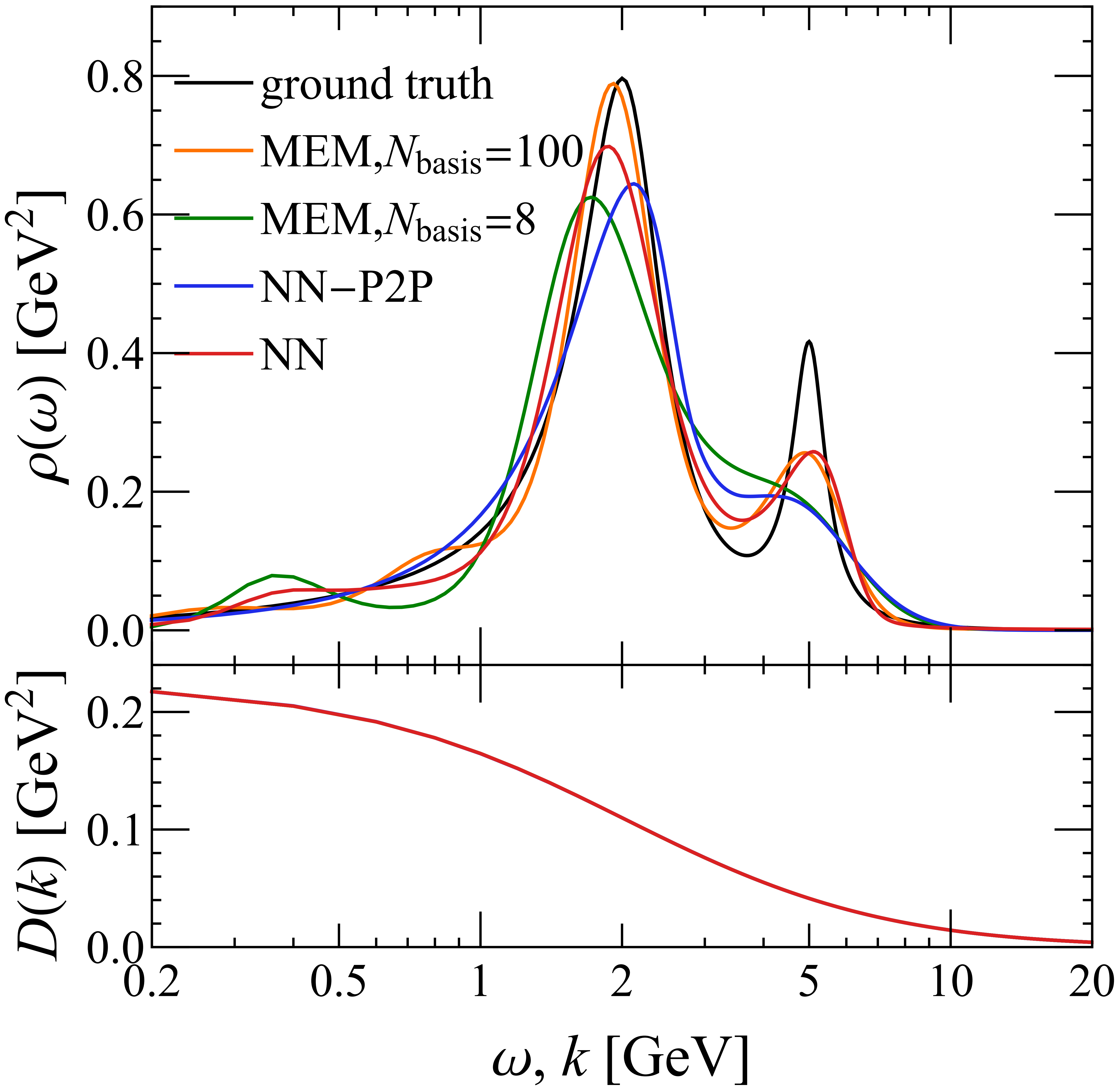
$$D(p) = \int_0^\infty K(p,\omega)\,\rho(\omega)\,d\omega \qquad K(p,\omega) \equiv \frac{\pi^{-1}\omega}{\omega^2 + p^2}$$



Can one learn the spectral function from correlation?

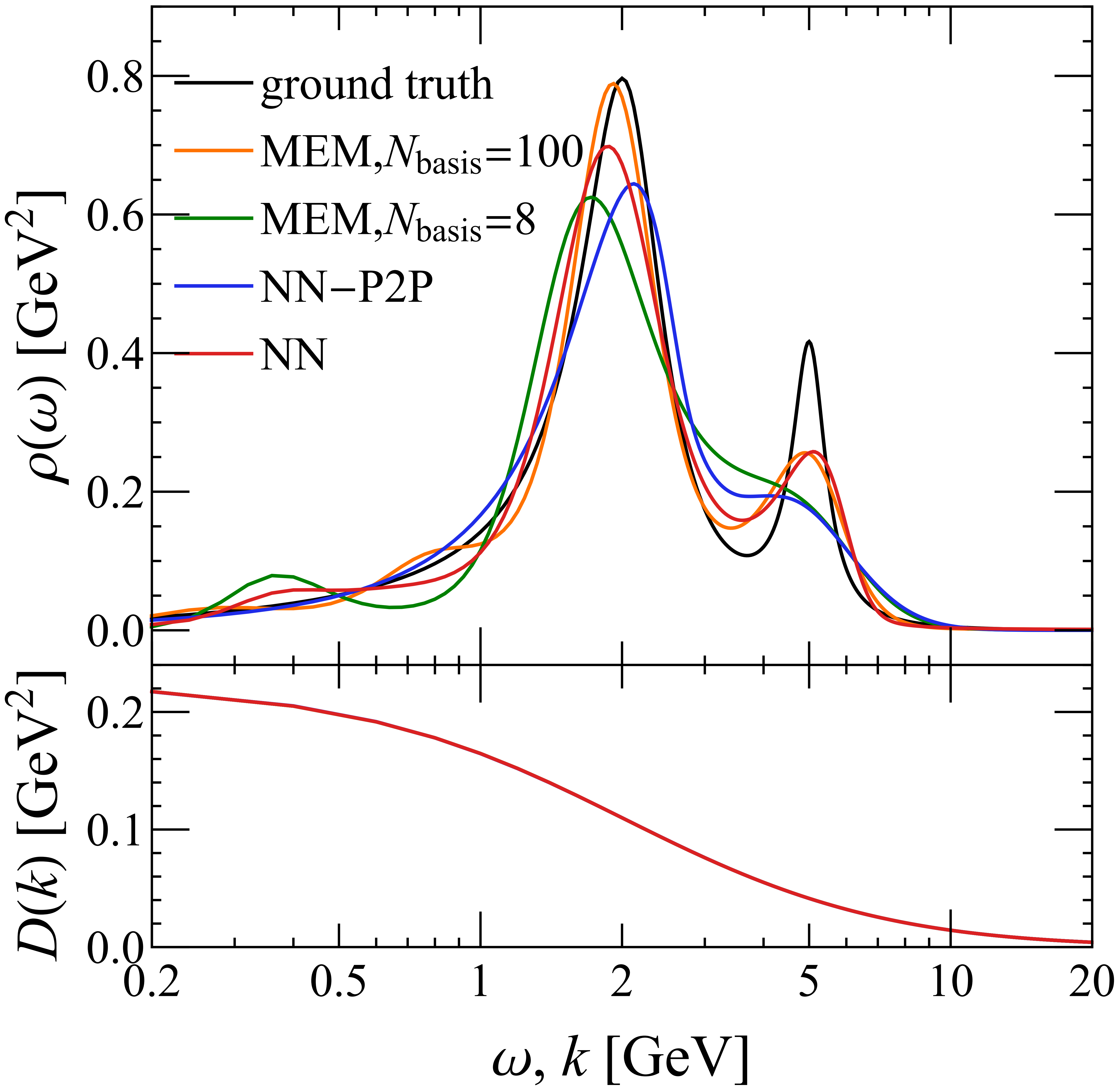


Can one learn the spectral function from correlation?



- ill-conditioned?
  - one needs more points in ω than that were given in k

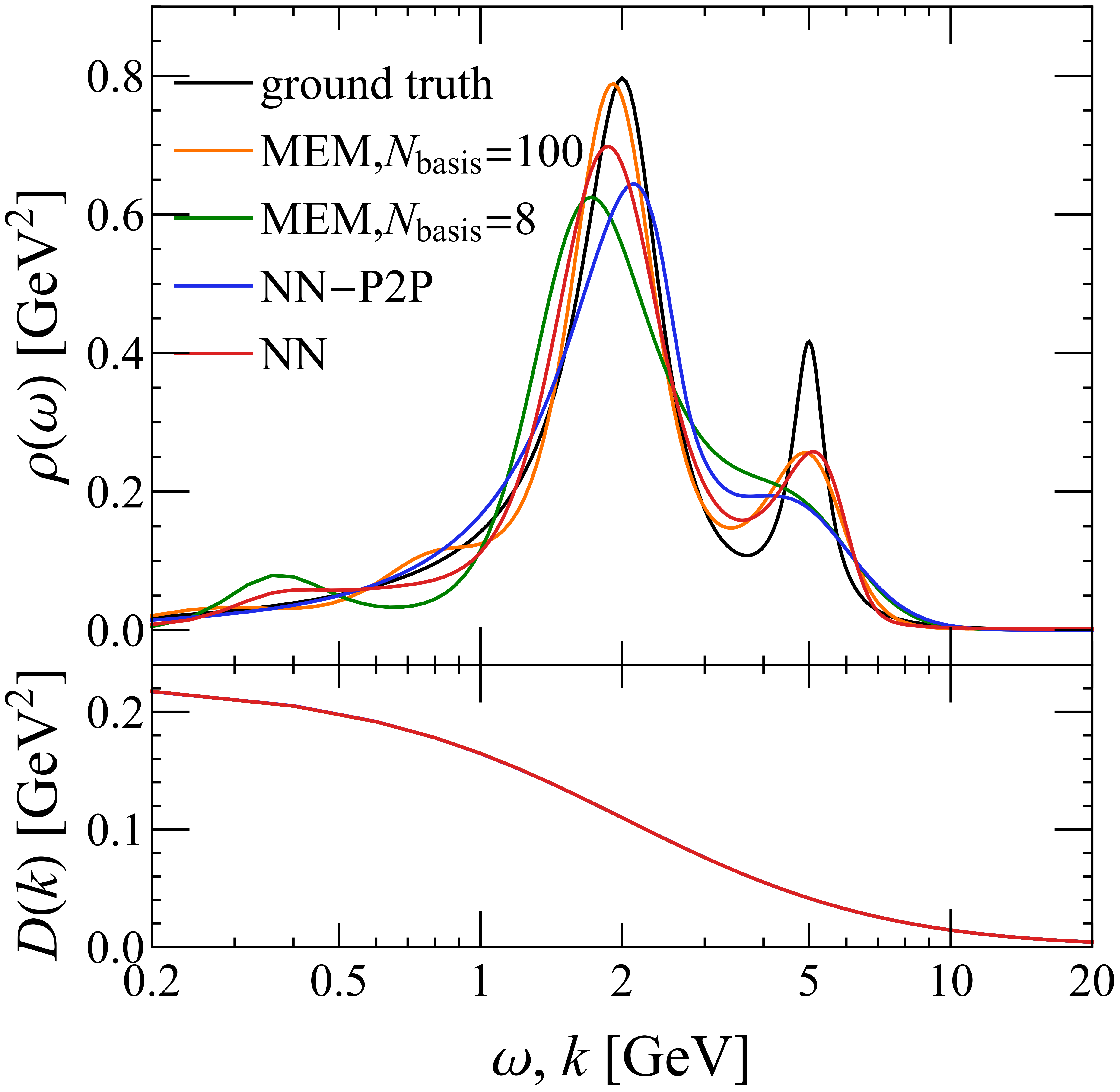
Can one learn the spectral function from correlation?



- ill-conditioned?
  - one needs more points in ω than that were given in k

resolved by increasing # of k-points?

Can one learn the spectral function from correlation?



- ill-conditioned?
  - one needs more points in ω than that were given in k

resolved by increasing # of k-points?

- No!!! The problem is ill-posed!!!

ill-posedness of inverse KL convolution

$$D(k) = \frac{1}{\pi}\int_0^\infty \frac{\omega\, \mathrm{d}\omega}{\omega^2 + k^2}\rho(\omega)$$

Linear operator in continuous space,

maps $\mathbb{R}_{[0,+\infty)}$ to $\mathbb{R}_{[0,+\infty)}$.

$\mathbb{R}_{[0,+\infty)}$: Real function defined in the domain $[0, +\infty)$.

$$D(k) = \frac{1}{\pi}\int_0^{\infty} \frac{\omega\, \mathrm{d}\omega}{\omega^2 + k^2} \rho(\omega)$$

Linear operator in continuous space,

maps $\mathbb{R}_{[0,+\infty)}$ to $\mathbb{R}_{[0,+\infty)}$.

One can define its eigenfunctions and eigenvalues:

$$\frac{1}{\pi}\int_0^{\infty} \frac{\omega\, \mathrm{d}\omega}{\omega^2 + k^2} \psi(\omega) = \lambda\, \psi(k)\,,$$

$\mathbb{R}_{[0,+\infty)}$: Real function defined in the domain $[0, +\infty)$.

eigenfunctions and eigenvalues of KL convolution:

$$\frac{1}{\pi}\int_0^\infty \frac{\omega\,\mathrm{d}\omega}{\omega^2+k^2}\psi(\omega) = \lambda\,\psi(k)\,,$$

infinite amount of solutions, labeled by (continuous) s:

$$\psi_{s,+}(x) = \frac{\cos\big(s\ln(x/a)\big)}{\sqrt{\pi\,x/a}}\,,$$

$$\psi_{s,-}(x) = \frac{\sin\big(s\ln(x/a)\big)}{\sqrt{\pi\,x/a}}\,,$$

$$\lambda_{s,\pm} = \frac{1}{2\cosh(\pi s/2)}\,.$$

eigenfunctions and eigenvalues of KL convolution:

$$D(k) = \frac{1}{\pi}\int_0^\infty \frac{\omega \, \mathrm{d}\omega}{\omega^2 + k^2}\rho(\omega)$$

$$\widetilde{\rho}_\pm(s) = \widetilde{D}_\pm(s) \, / \, \lambda_s$$

$$\frac{1}{\pi}\int_0^\infty \frac{\omega \, \mathrm{d}\omega}{\omega^2 + k^2}\psi_{\pm,s}(\omega) = \lambda_s \, \psi_{\pm,s}(k) \, ,$$

eigenfunctions and eigenvalues of KL convolution:

$$D(k) = \frac{1}{\pi}\int_0^\infty \frac{\omega\, \mathrm{d}\omega}{\omega^2 + k^2}\rho(\omega)$$

$$\widetilde{D}_\pm(s) = \int_0^{+\infty} \frac{k\, \mathrm{d}k}{a^2} D(k)\, \psi_{\pm,s}(k)$$

$$\widetilde{\rho}_\pm(s) = \widetilde{D}_\pm(s) \,/\, \lambda_s$$

$$\frac{1}{\pi}\int_0^\infty \frac{\omega\, \mathrm{d}\omega}{\omega^2 + k^2}\psi_{\pm,s}(\omega) = \lambda_s\, \psi_{\pm,s}(k)\,,$$

eigenfunctions and eigenvalues of KL convolution:

$$D(k) = \frac{1}{\pi}\int_0^\infty \frac{\omega\,\mathrm{d}\omega}{\omega^2 + k^2}\rho(\omega)$$

$$\widetilde{D}_\pm(s) = \int_0^{+\infty} \frac{k\,\mathrm{d}k}{a^2} D(k)\,\psi_{\pm,s}(k)$$

$$\rho(\omega) = \sum_{i=\pm}\int_{-\infty}^{+\infty} \frac{\mathrm{d}s}{2}\widetilde{\rho}_\pm(s)\,\psi_{i,s}(\omega)$$

$$\widetilde{\rho}_\pm(s) = \widetilde{D}_\pm(s)\,/\,\lambda_s$$

$$\frac{1}{\pi}\int_0^\infty \frac{\omega\,\mathrm{d}\omega}{\omega^2 + k^2}\psi_{\pm,s}(\omega) = \lambda_s\,\psi_{\pm,s}(k)\,,$$

eigenfunctions and eigenvalues of KL convolution:

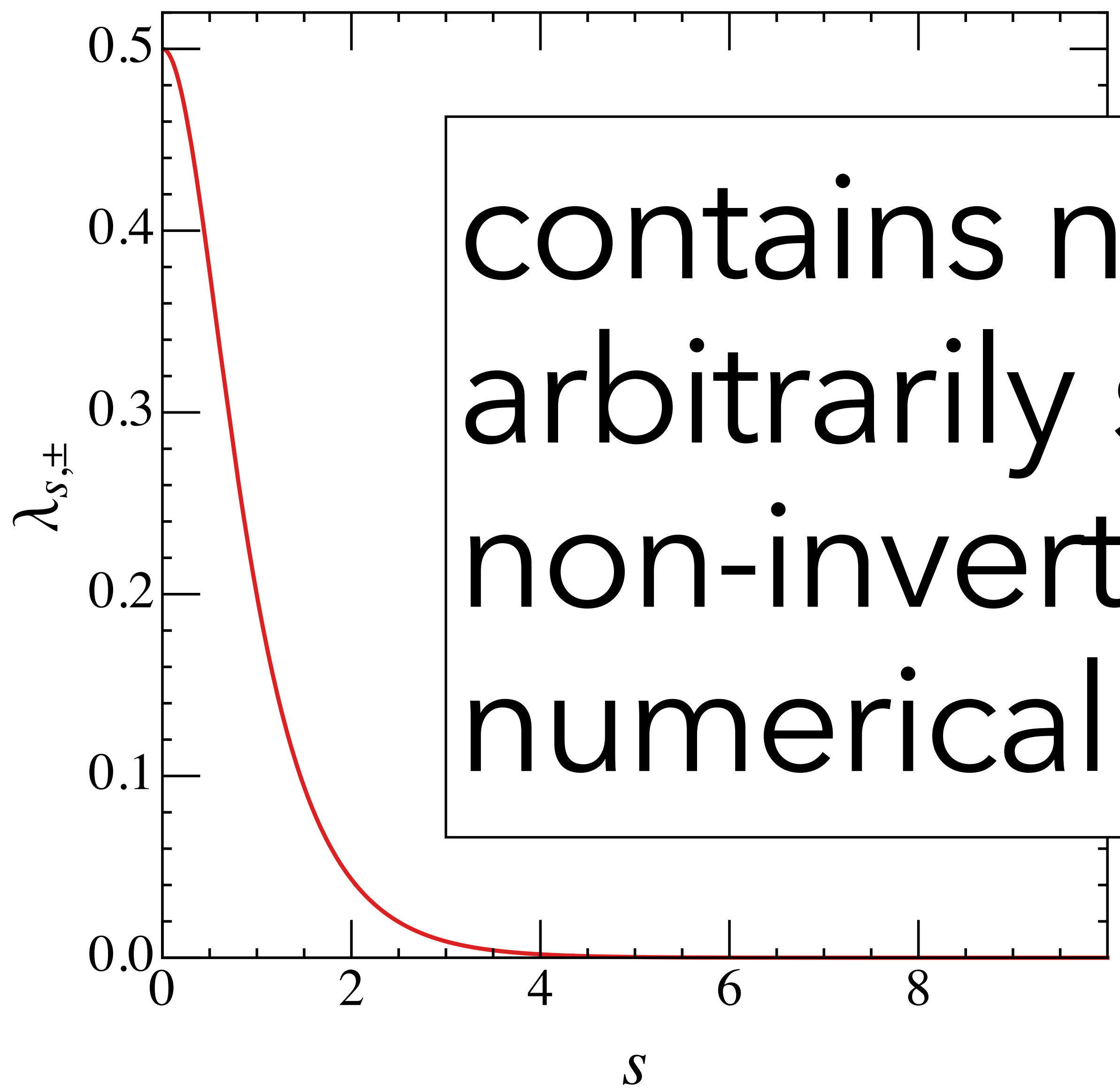
$$D(k) = \frac{1}{\pi}\int_0^\infty \frac{\omega\,\mathrm{d}\omega}{\omega^2 + k^2}\,\rho(\omega)$$

$$\widetilde{D}_\pm(s) = \int_0^{+\infty} \frac{k\,\mathrm{d}k}{a^2}\,D(k)\,\psi_{\pm,s}(k)$$

$$\rho(\omega) = \sum_{i=\pm}\int_{-\infty}^{+\infty} \frac{\mathrm{d}s}{2}\,\widetilde{\rho}_\pm(s)\,\psi_{i,s}(\omega)$$

$$\widetilde{\rho}_\pm(s) = \widetilde{D}_\pm(s)\,/\,\lambda_s$$

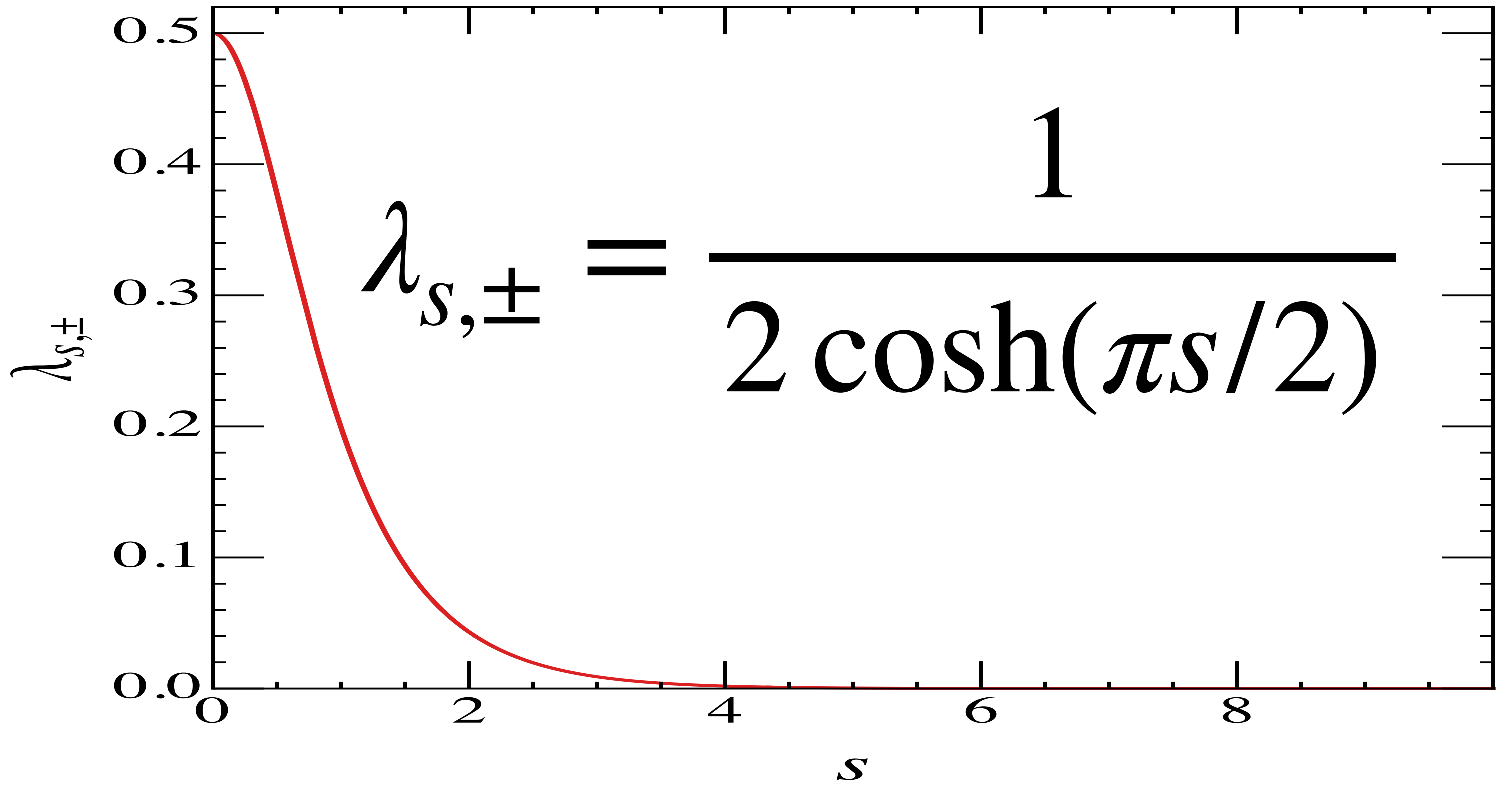
$$\lambda_{s,\pm} = \frac{1}{2\cosh(\pi s/2)}\,.$$



contains non-zero but arbitrarily small eigenvalues, non-invertible given finite numerical precision.

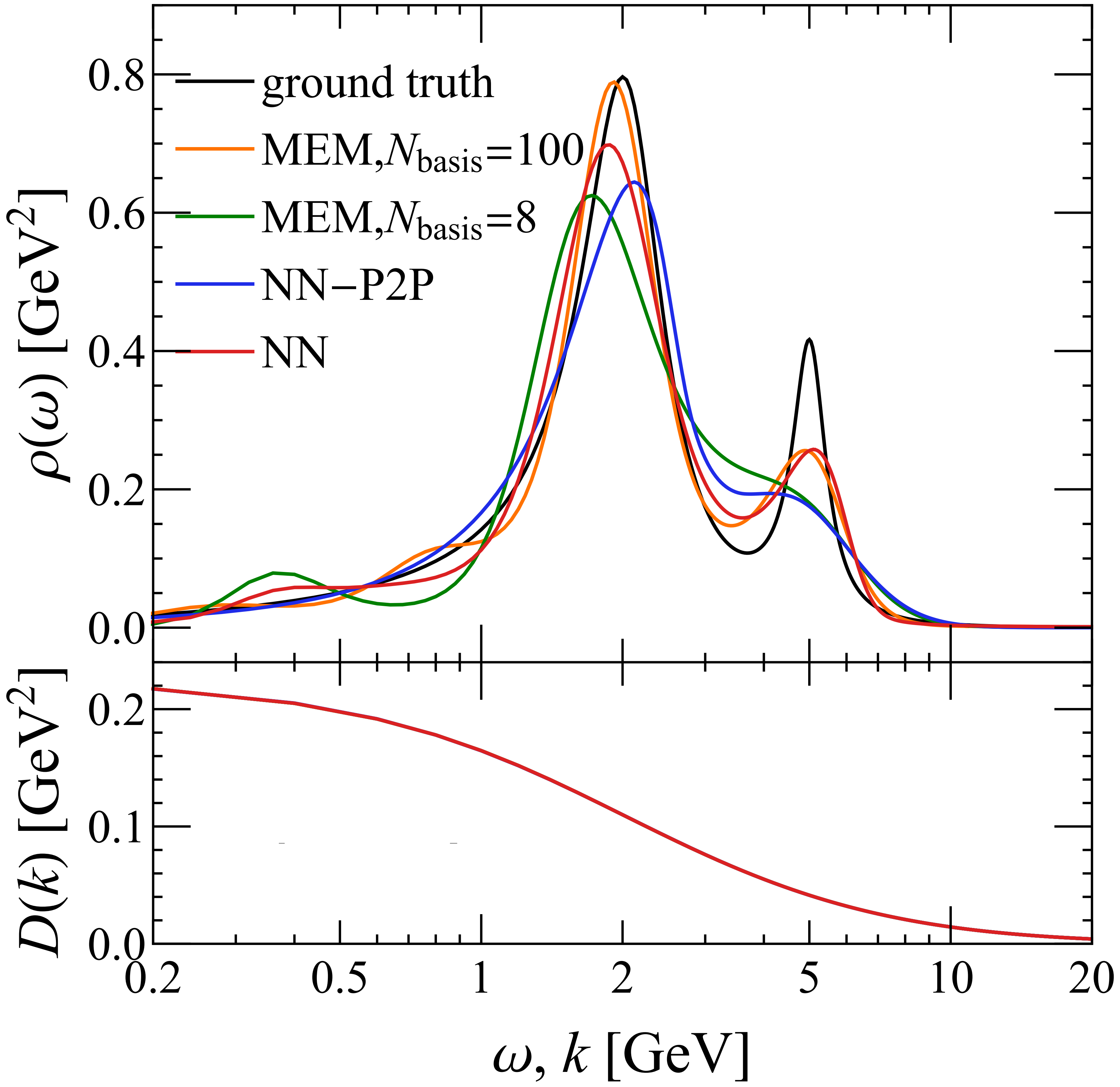
relook at the discrepancy

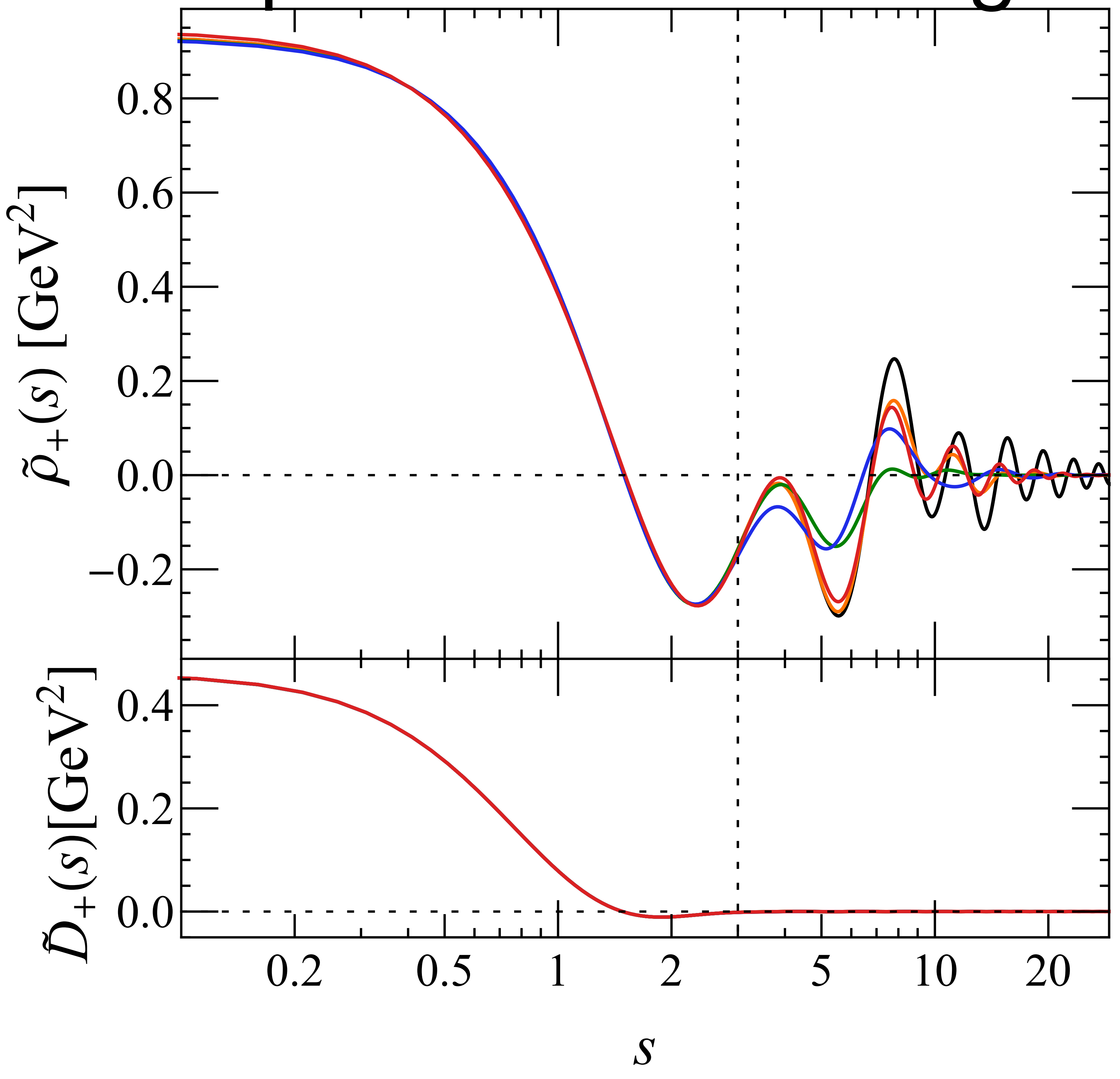
$$\psi_{s,+}(x) = \frac{\cos\big(s \ln(x/a)\big)}{\sqrt{\pi}\, x/a}$$



$$\lambda_{s,\pm} = \frac{1}{2\cosh(\pi s/2)}$$

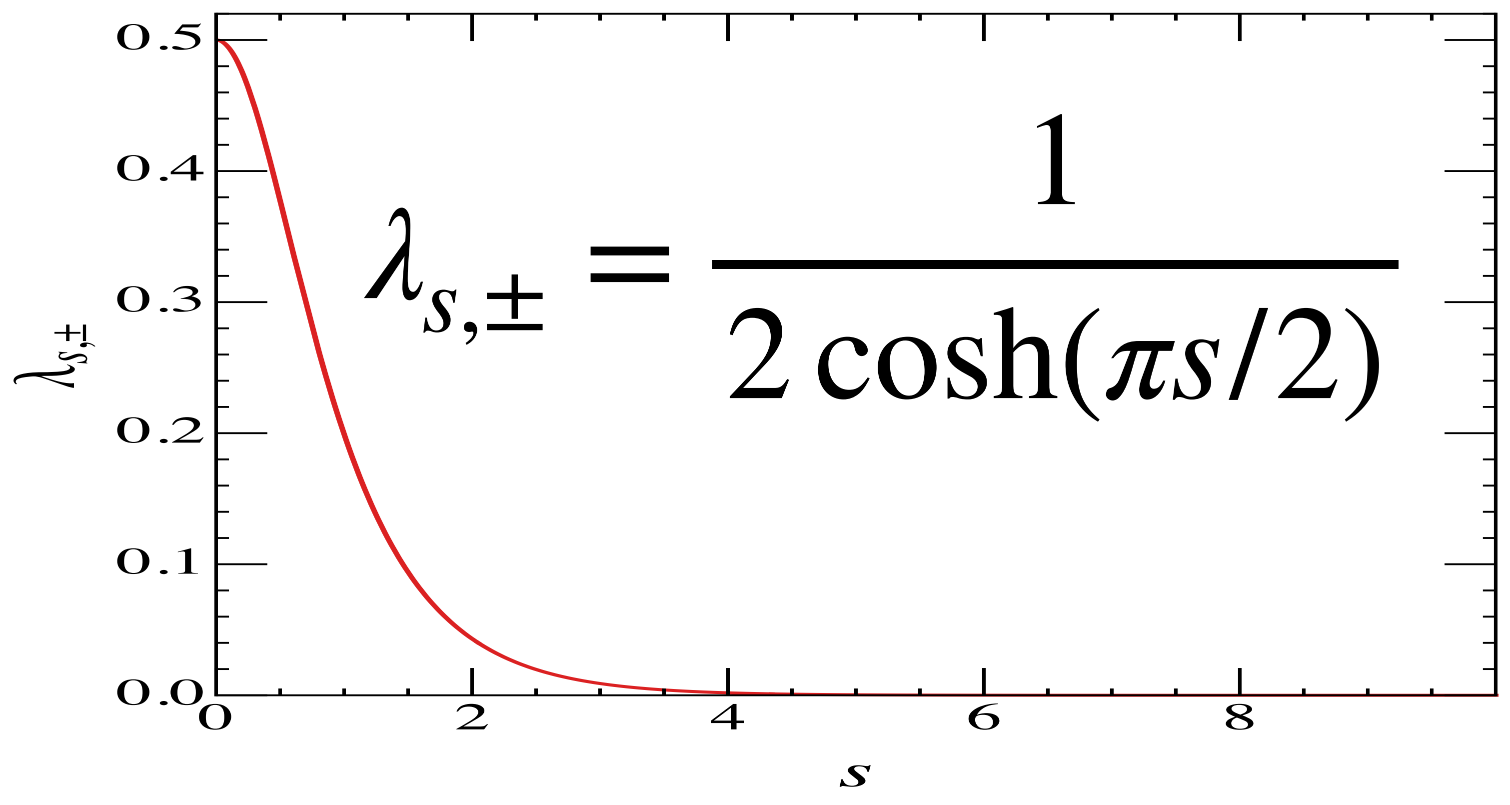
decomposition into the eigen-space

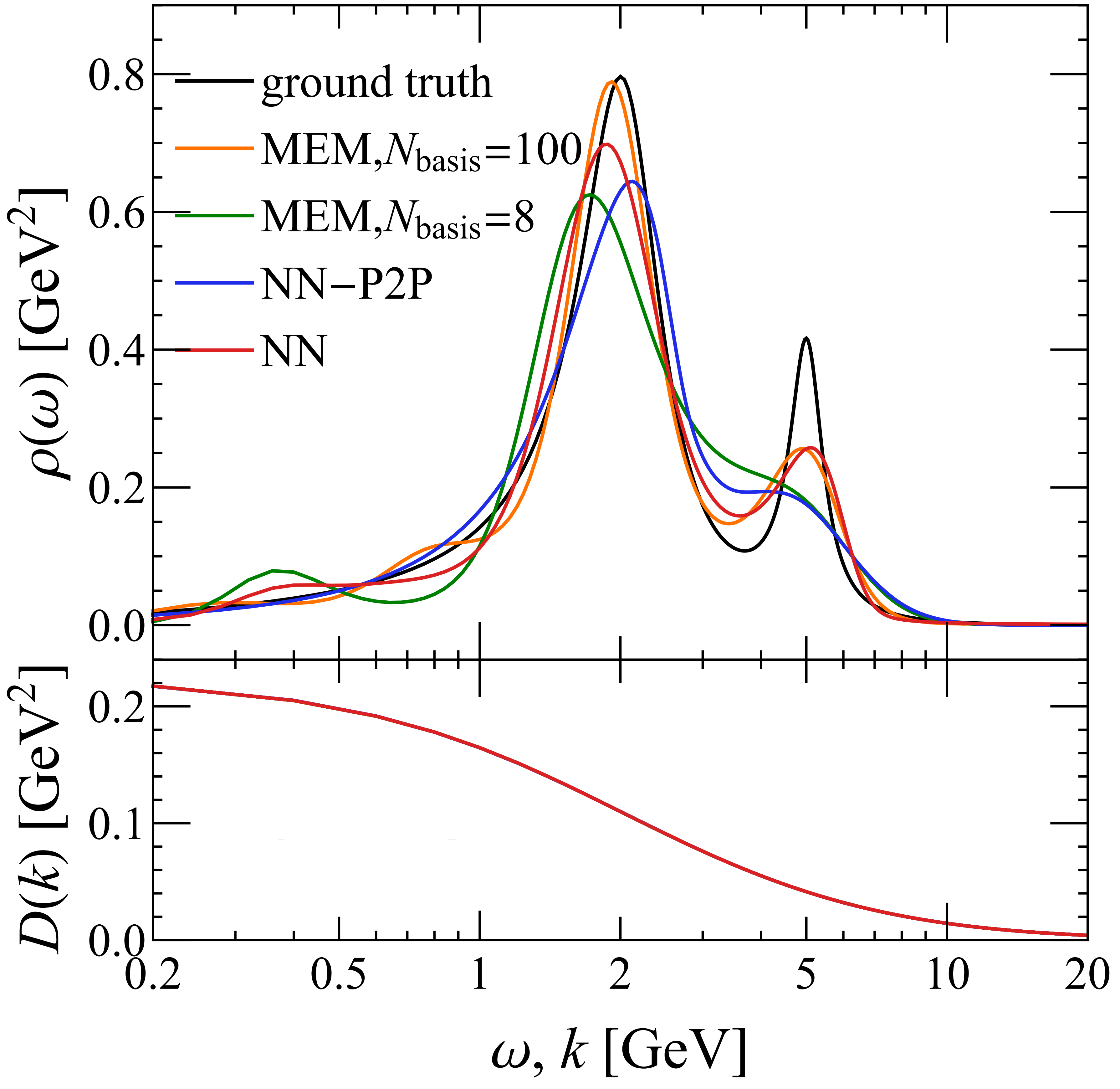




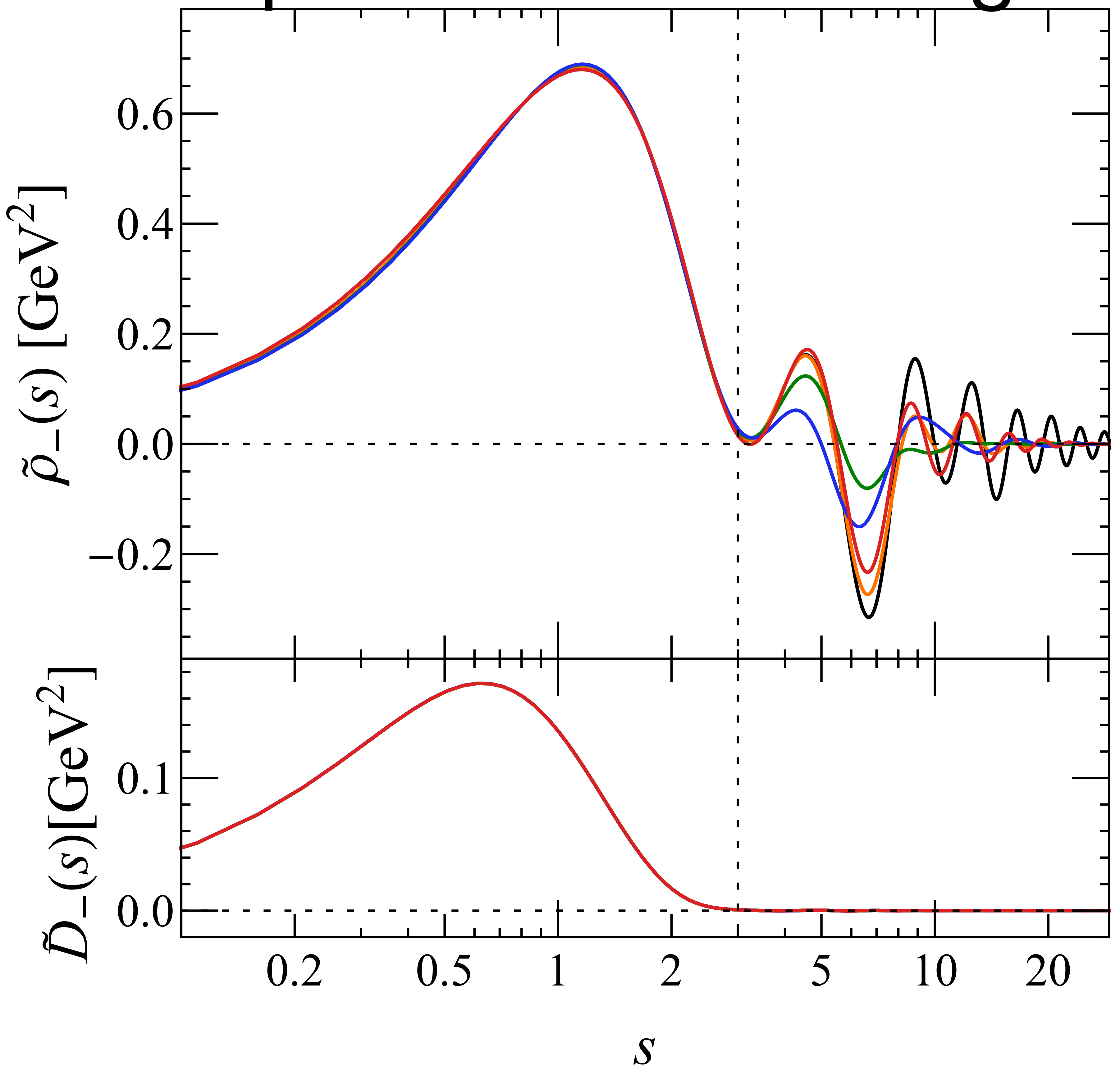
relook at the discrepancy

$$\psi_{s,-}(x) = \frac{\sin\left(s \ln(x/a)\right)}{\sqrt{\pi}\, x/a}$$





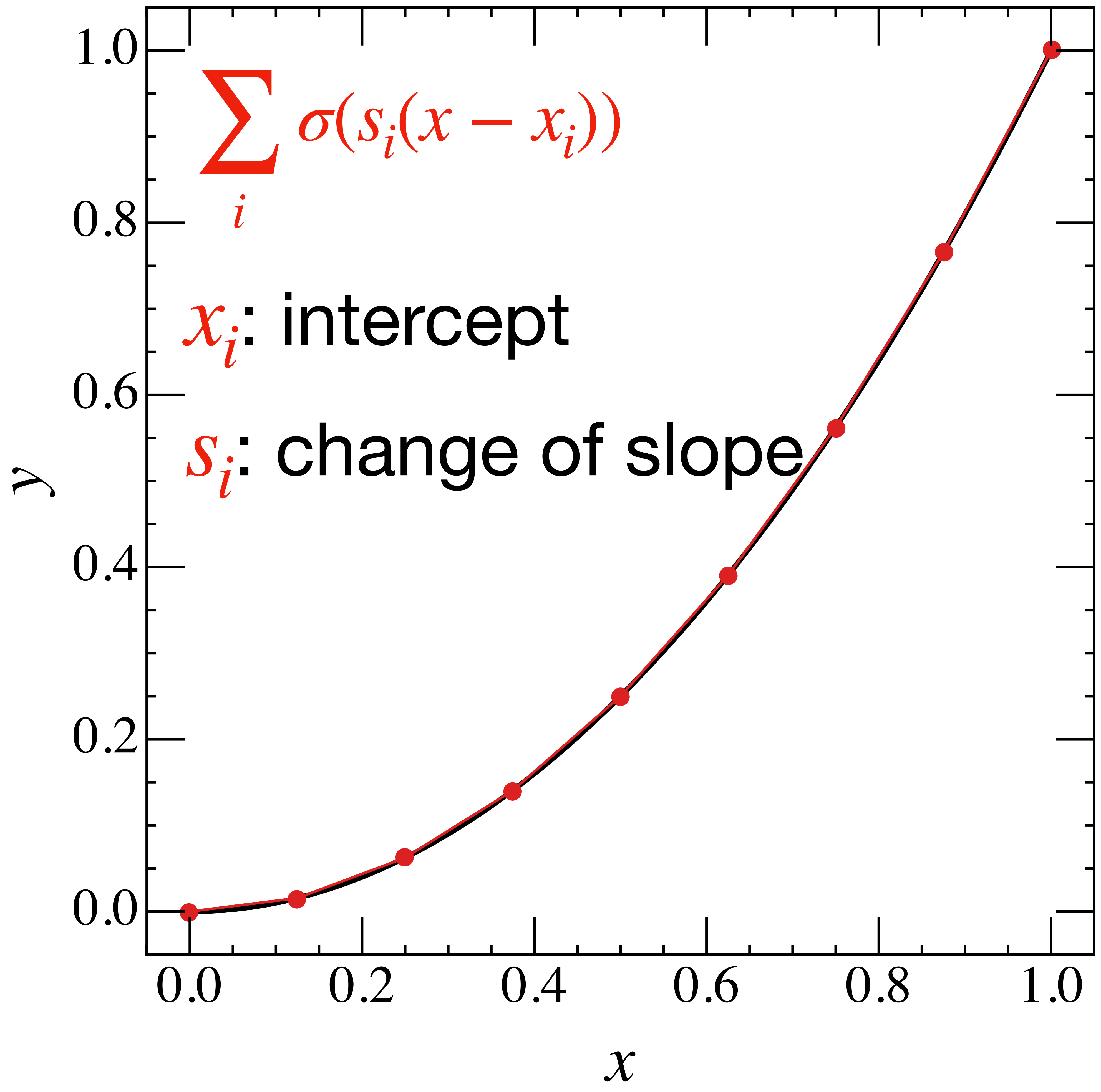
decomposition into the eigen-space



Then, why Deep Learning?

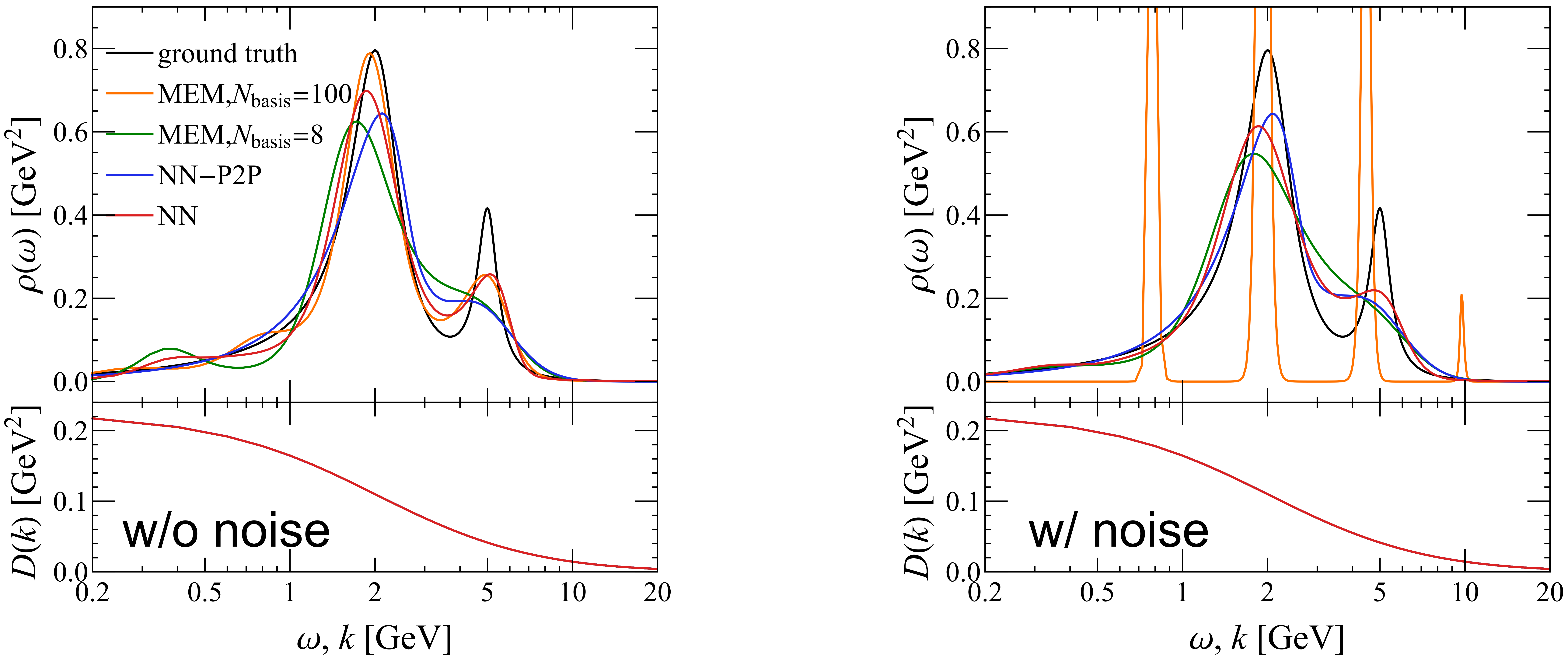
DNN is a natural implementation of smoothness regularization!

DNN is a natural implementation of smoothness regularization!

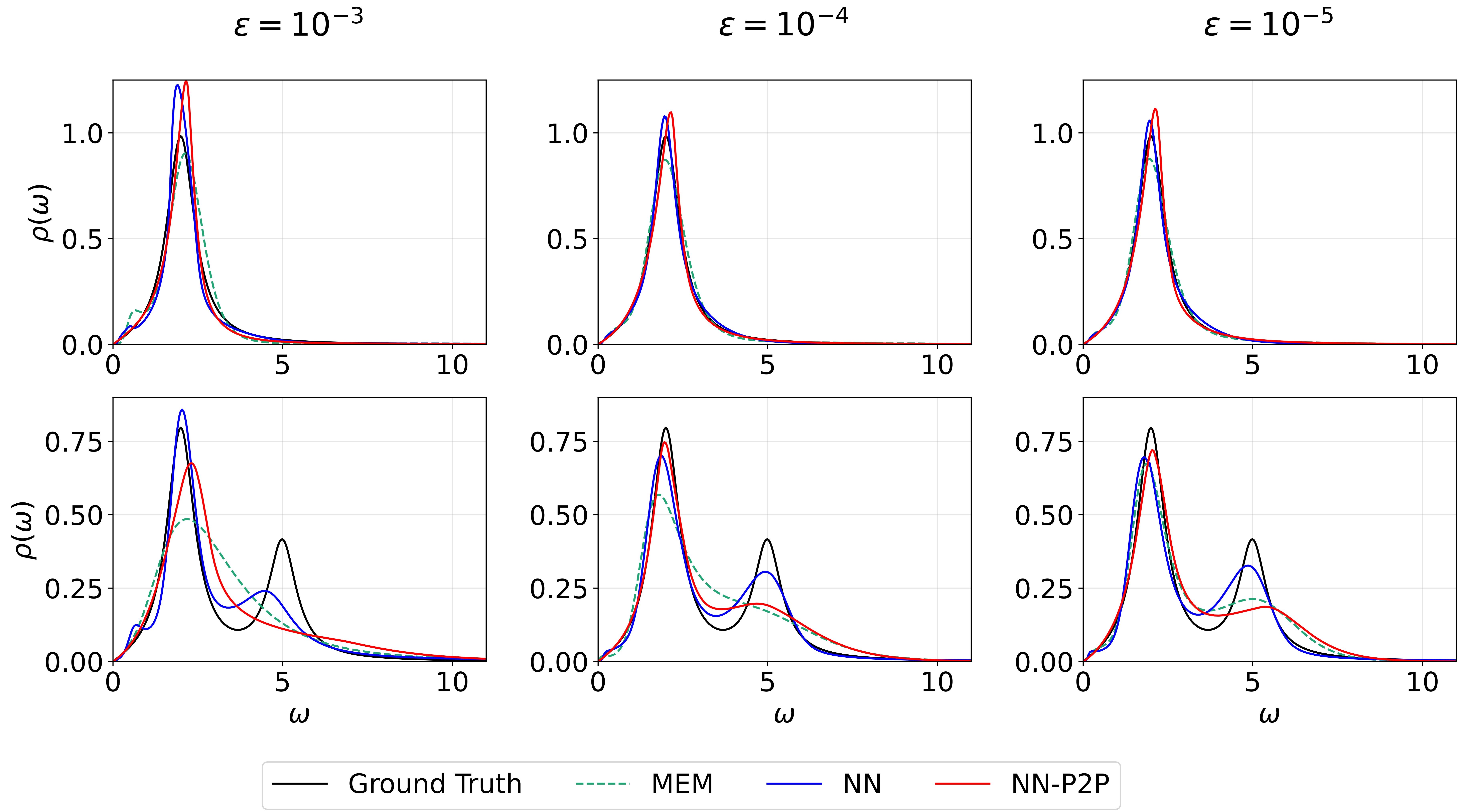


minimizing $\sum_i s_i^2 \leftrightarrow$ imposing smoothness

Robustness against larger noise



Robustness against larger noise



Summary

- New algorithm employing DNN to solve inverse problems;
- Extracted HF complex $V(T, r)$;
- Reconstructing spectral function …

Summary

- New algorithm employing DNN to solve inverse problems;
- Extracted HF complex $V(T, r)$;
- Reconstructing spectral function … is ill-posed!!!
  - physics-based parameterization
  - real time evolution?
  - supervised learning?

Machine learning spectral functions in lattice QCD

S.-Y. Chen (Hua-Zhong Normal U., LQLP and CCNU, Wuhan, Inst. Part. Phys.), H.-T. Ding (Hua-Zhong Normal U., LQLP and CCNU, Wuhan, Inst. Part. Phys.), F.-Y. Liu (Hua-Zhong Normal U., LQLP and CCNU, Wuhan, Inst. Part. Phys. and Eotvos U.), G. Papp (Eotvos U.), C.-B. Yang (Hua-Zhong Normal U., LQLP and CCNU, Wuhan, Inst. Part. Phys.)

Oct 26, 2021

25 pages

e-Print: 2110.13521 [hep-lat]

View in: ADS Abstract Service

eigenfunctions and eigenvalues of KL convolution:

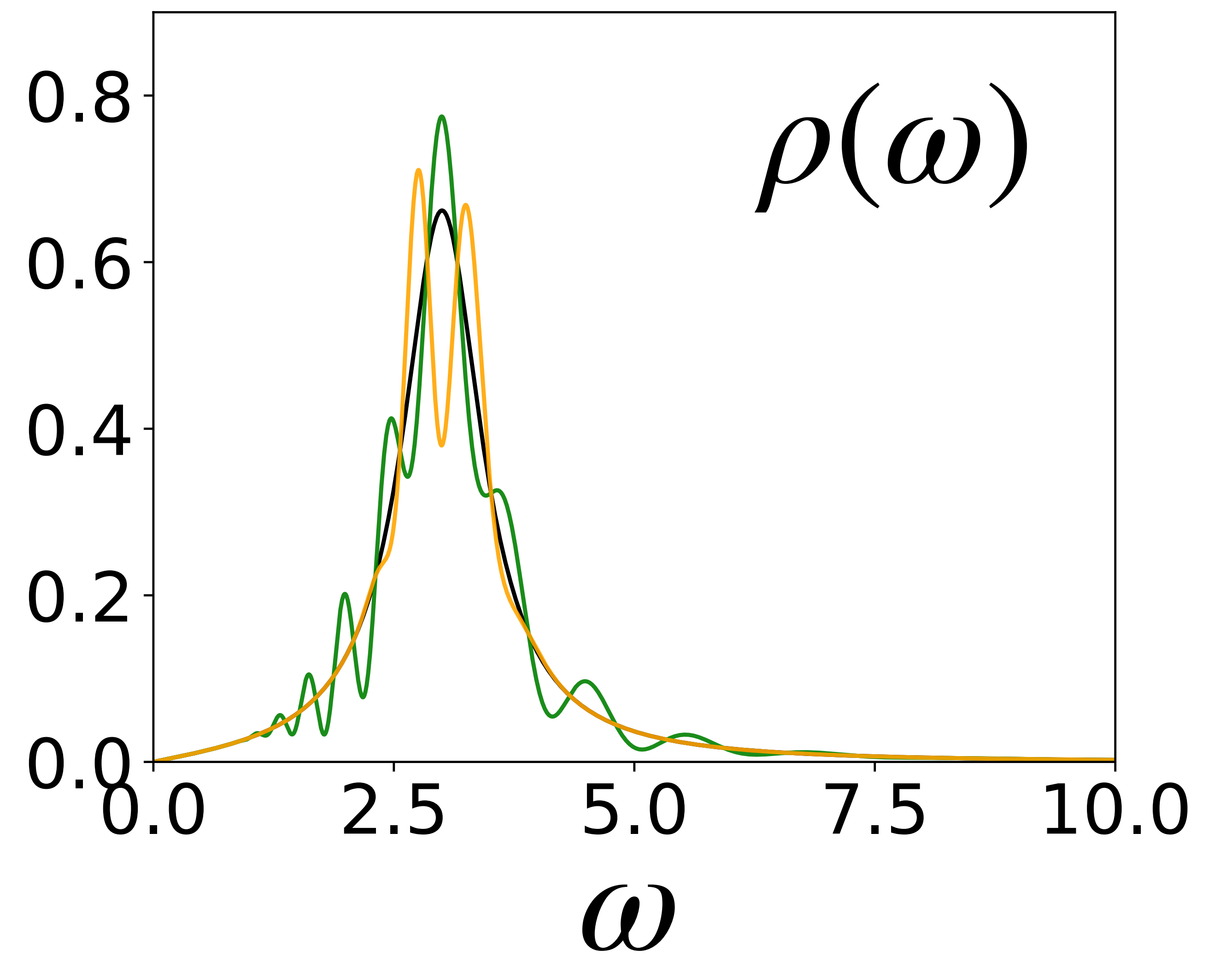
$$\frac{1}{\pi}\int_0^\infty \frac{\omega\,\mathrm{d}\omega}{\omega^2+k^2}\psi(\omega) = \lambda\,\psi(k)\,,$$

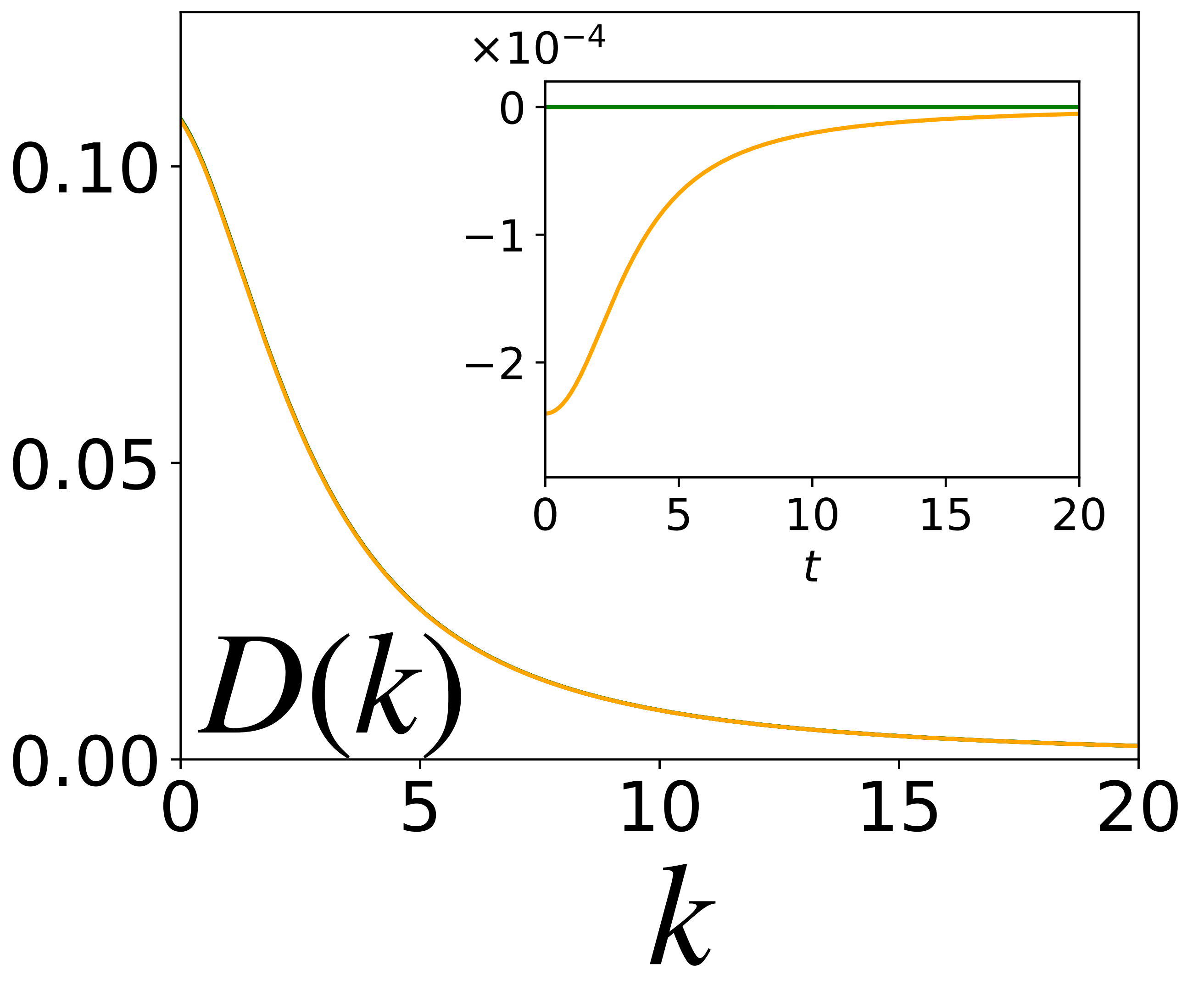
infinite amount of solutions, labeled by (continuous) s:

$$\psi_{s,+}(x) = \frac{\cos\big(s\ln(x/a)\big)}{\sqrt{\pi}\,x/a}\,,$$

$$\psi_{s,-}(x) = \frac{\sin\big(s\ln(x/a)\big)}{\sqrt{\pi}\,x/a}\,,$$

$$\lambda_{s,\pm} = \frac{1}{2\cosh(\pi s/2)}\,.$$





Can we really learn $V(r)$ from $\{E_n\}$?

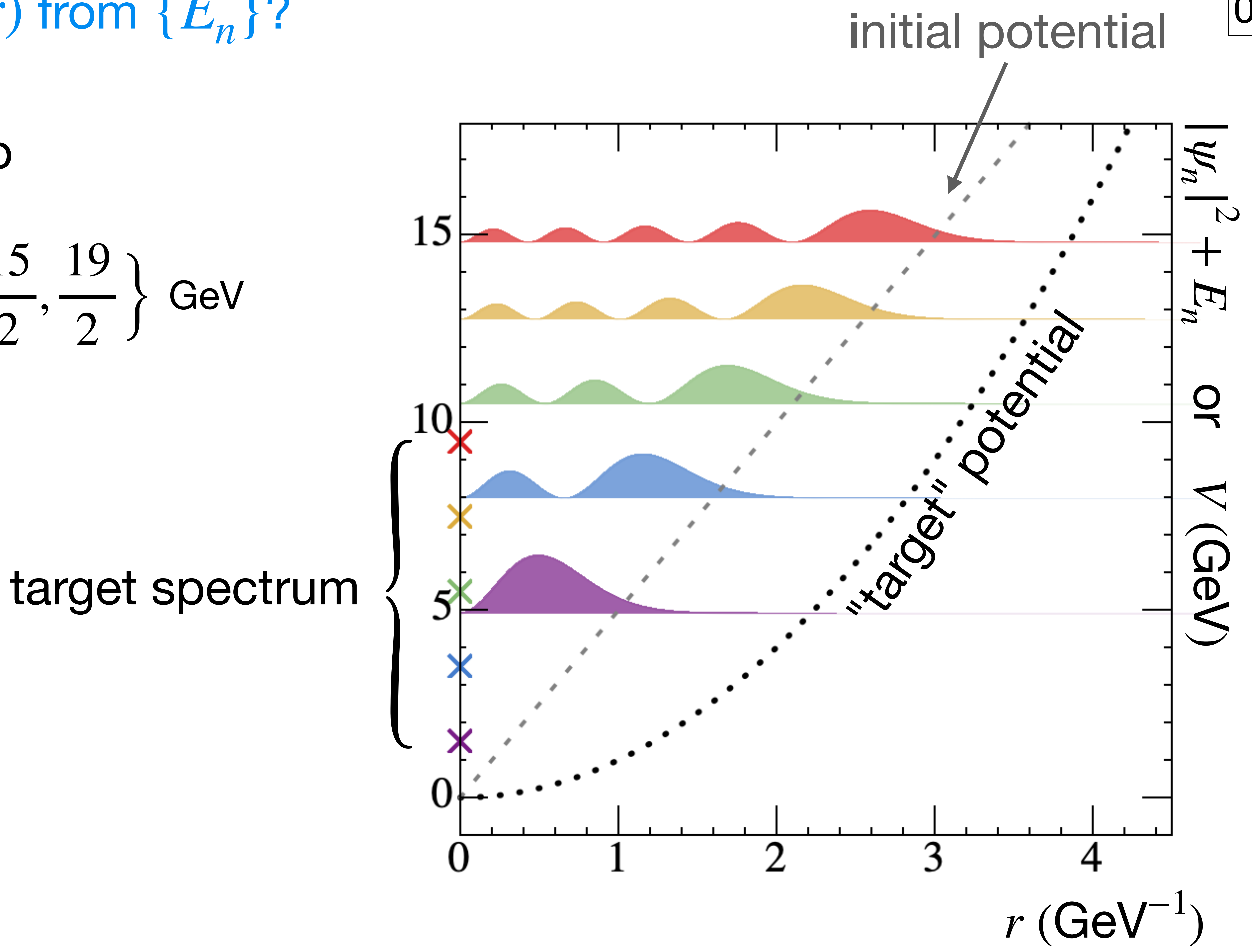
continuous → $V(r)$

discrete, finite → $\{E_n\}$

Can we really learn $V(r)$ from $\{E_n\}$?

learn $V(r)$ according to

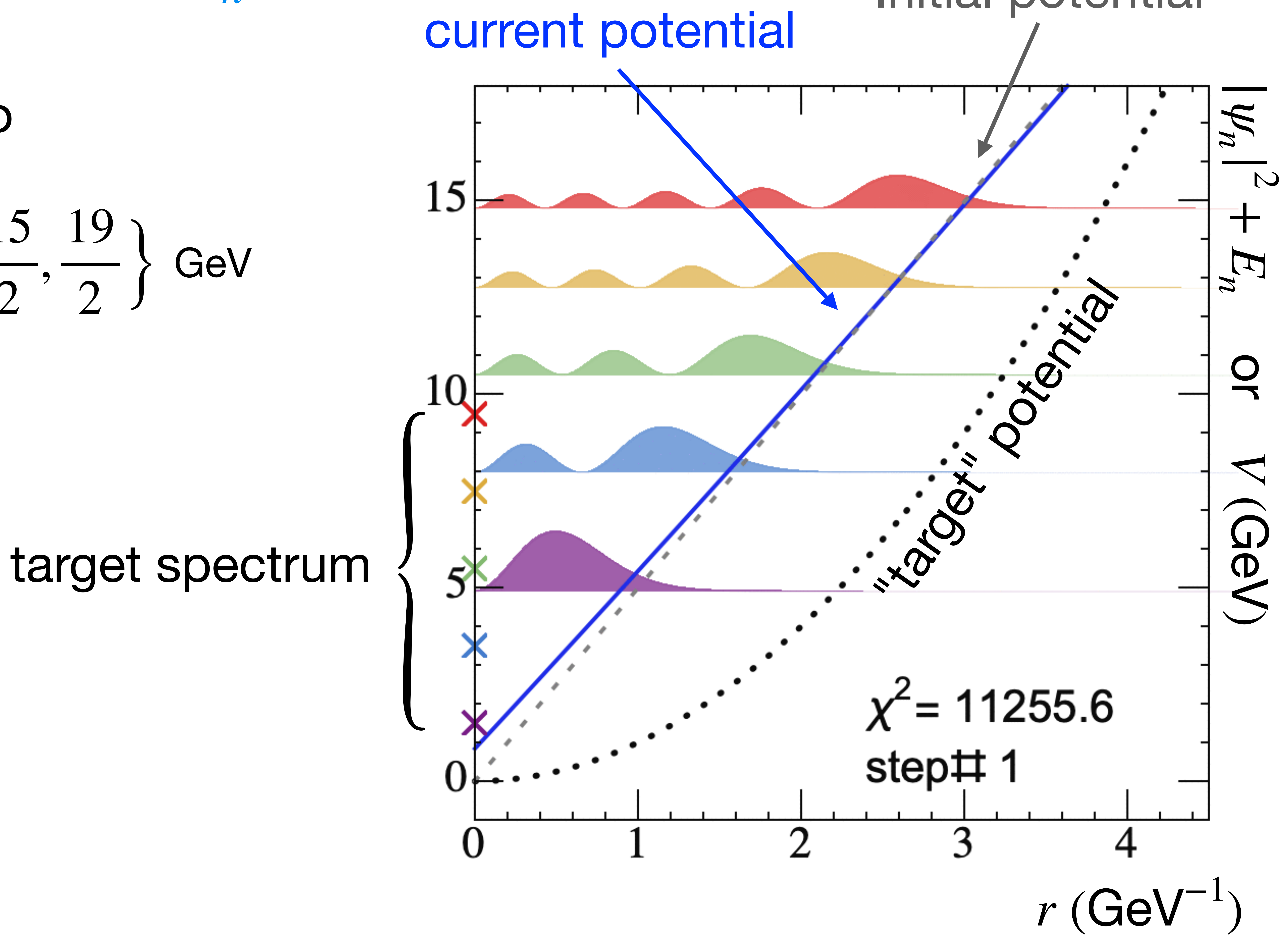
$$\{E_n\} = \left\{\frac{3}{2}, \frac{7}{2}, \frac{11}{2}, \frac{15}{2}, \frac{19}{2}\right\} \text{ GeV}$$



Can we really learn $V(r)$ from $\{E_n\}$?

learn $V(r)$ according to

$$\{E_n\} = \left\{\frac{3}{2}, \frac{7}{2}, \frac{11}{2}, \frac{15}{2}, \frac{19}{2}\right\} \text{ GeV}$$



Test - Can we recover a known complex $V(T, r)$?

- Start with a known potential

$$V_R(T,r) = \frac{\sigma}{\mu_D}\Big(2-(2+\mu_D r)e^{-\mu_D r}\Big) - \alpha\Big(\mu_D + \frac{e^{-\mu_D r}}{r}\Big) + B\,,$$

$$V_I(T,r) = -\frac{\sqrt{\pi}}{4}\mu_D T\sigma r^3 G^{2,2}_{2,4}\left(\begin{smallmatrix} -\frac{1}{2},-\frac{1}{2} \\ \frac{1}{2},\frac{1}{2},-\frac{3}{2},-1 \end{smallmatrix}\,\middle|\,\frac{\mu_D^2 r^2}{4}\right) - \alpha\, T\,\phi(\mu_D r)\,.$$

- Compute $\{m_n\,,\,\Gamma_n\}$ at

$T = \{0, 151, 173\,, 199, 251, 334\}$ MeV

- Learn the potential using DNN

Test - Can we recover a known complex $V(T, r)$?   -- Yes!

- Start with a known potential (solid line)

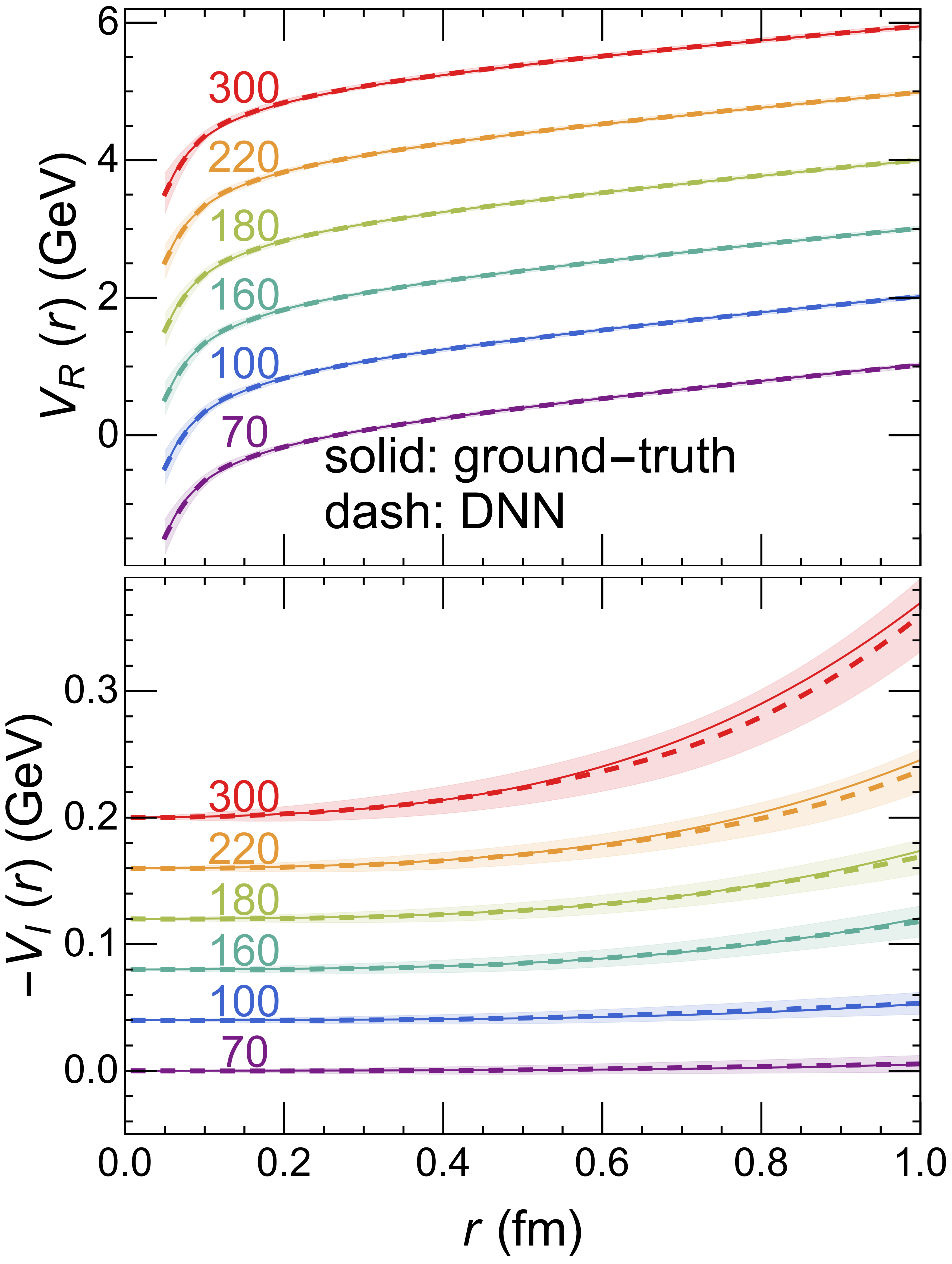
$$V_R(T,r) = \frac{\sigma}{\mu_D}\left(2-(2+\mu_D r)e^{-\mu_D r}\right) - \alpha\left(\mu_D + \frac{e^{-\mu_D r}}{r}\right) + B\,,$$

$$V_I(T,r) = -\frac{\sqrt{\pi}}{4}\mu_D T\sigma r^3 G^{2,2}_{2,4}\left(\begin{smallmatrix}-\frac{1}{2},-\frac{1}{2}\\ \frac{1}{2},\frac{1}{2},-\frac{3}{2},-1\end{smallmatrix}\,\middle|\,\frac{\mu_D^2 r^2}{4}\right) - \alpha\, T\,\phi(\mu_D r)\,.$$

- Compute $\{m_n\,,\,\Gamma_n\}$ at

$T = \{0, 151, 173\,, 199, 251, 334\}$ MeV

- Learn the potential using DNN (dash + band)



Test - Can we recover a known complex $V(T,r)$? extrapolation is risky!

- Start with a known potential (solid line)

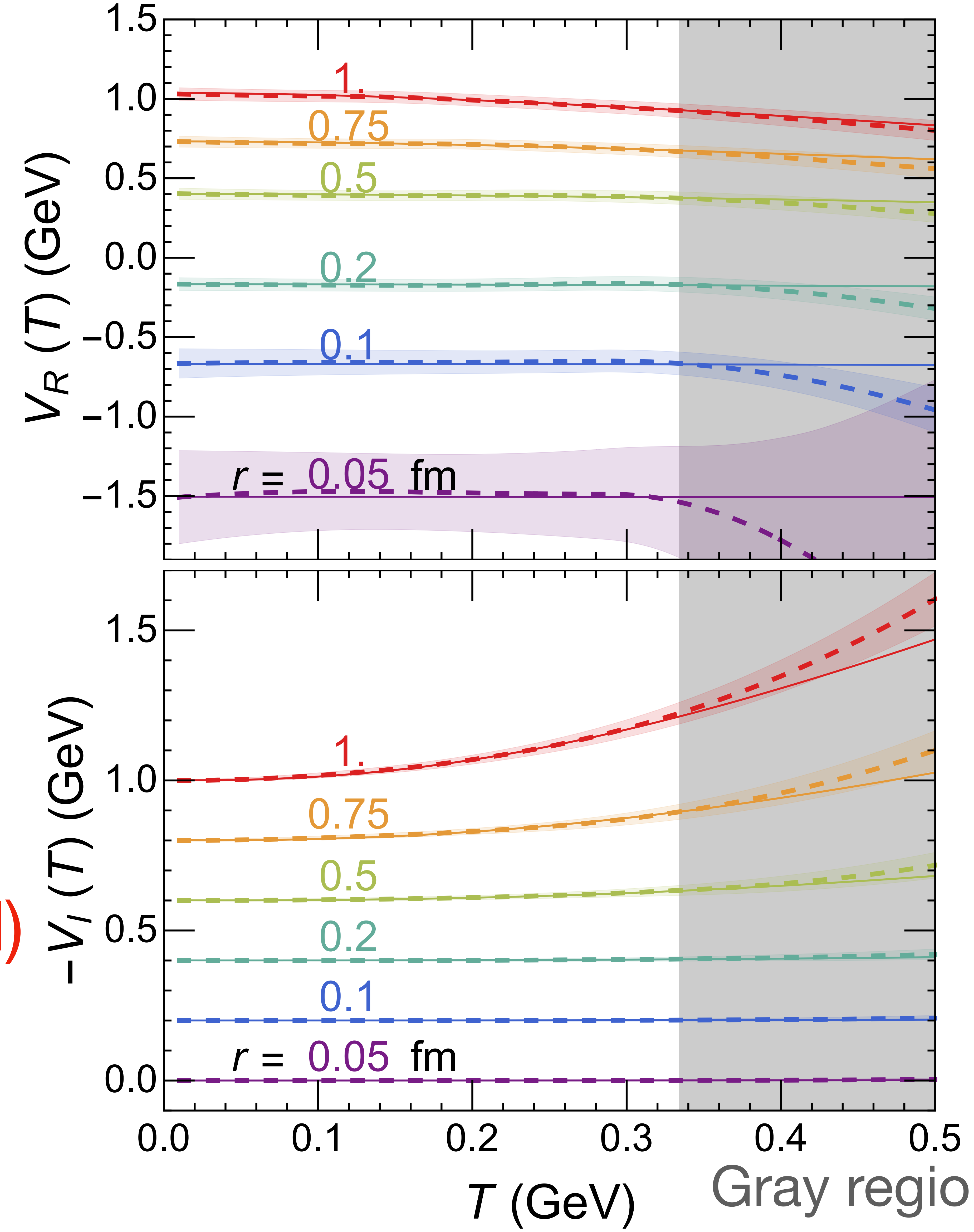
$$V_R(T,r) = \frac{\sigma}{\mu_D}\left(2-(2+\mu_D r)e^{-\mu_D r}\right) - \alpha\left(\mu_D + \frac{e^{-\mu_D r}}{r}\right) + B\,,$$

$$V_I(T,r) = -\frac{\sqrt{\pi}}{4}\mu_D T\sigma r^3 G^{2,2}_{2,4}\left(\begin{smallmatrix}-\frac{1}{2},-\frac{1}{2}\\ \frac{1}{2},\frac{1}{2},-\frac{3}{2},-1\end{smallmatrix}\,\middle|\,\frac{\mu_D^2 r^2}{4}\right) - \alpha T\phi(\mu_D r)\,.$$

- Compute $\{m_n\,,\,\Gamma_n\}$ at

$$T = \{0, 151, 173, 199, 251, 334\}\ \text{MeV}$$

- Learn the potential using DNN (dash + band)



Gray region: extrapolations

How to learn $V(r)$ from $\{E_n\}$?

- parameterize the potential $V(r\,|\,\boldsymbol{\theta})$, ***scan*** the whole $\boldsymbol{\theta}$-space,

  minimize $\chi^2 \equiv \sum_i \left(\frac{E_{\boldsymbol{\theta},i} - E_i}{\delta E_i}\right)^2$

- a ***gradient-descent*** based method:
  - goal -- find the $\boldsymbol{\theta}$-point that $\nabla_{\boldsymbol{\theta}}\chi^2 = 0$
  - update $\boldsymbol{\theta}$ iteratively according to $\boldsymbol{\Delta\theta} \propto \nabla_{\boldsymbol{\theta}}\chi^2$

$$\nabla_{\boldsymbol{\theta}}\chi^2 = 2\sum_i \frac{E_{\boldsymbol{\theta},i} - E_i}{(\delta E_i)^2}\nabla_{\boldsymbol{\theta}}E_{\boldsymbol{\theta},i}$$

$$\nabla_{\boldsymbol{\theta}}E_{\boldsymbol{\theta},i} = \langle\psi_i\,|\,\nabla_{\boldsymbol{\theta}}V(\boldsymbol{\theta})\,|\,\psi_i\rangle$$

- general unbiased parameterization scheme? Deep Neural Network!

How to compute the likelihood (density) distribution of $V_{\boldsymbol{\theta}}$

$$P(V_{\boldsymbol{\theta}})\mathrm{d}V = \text{Posterior}(\boldsymbol{\theta}\,|\,\text{data})\mathrm{d}^N\boldsymbol{\theta}$$

- Sample $\{\boldsymbol{\theta}_i\}$ according to a reference distribution: $P(\boldsymbol{\theta}) = \tilde{P}(\boldsymbol{\theta})$;
- Each data point corresponds to the element volume $\mathrm{d}^N\boldsymbol{\theta}_i = 1/\tilde{P}(\boldsymbol{\theta}_i)$;
- Compute $V_{\boldsymbol{\theta}_i}(r)$, $\chi^2_{\boldsymbol{\theta}_i}$, and Posterior$(\boldsymbol{\theta}_i\,|\,\text{data})$;
- For given *r*, histogram $V_{\boldsymbol{\theta}_i}(r)$ with weights

$$w_i = P(V_{\boldsymbol{\theta}_i})\mathrm{d}V_i = \text{Posterior}(\boldsymbol{\theta}_i)/\tilde{P}(\boldsymbol{\theta}_i)$$

- In practice:

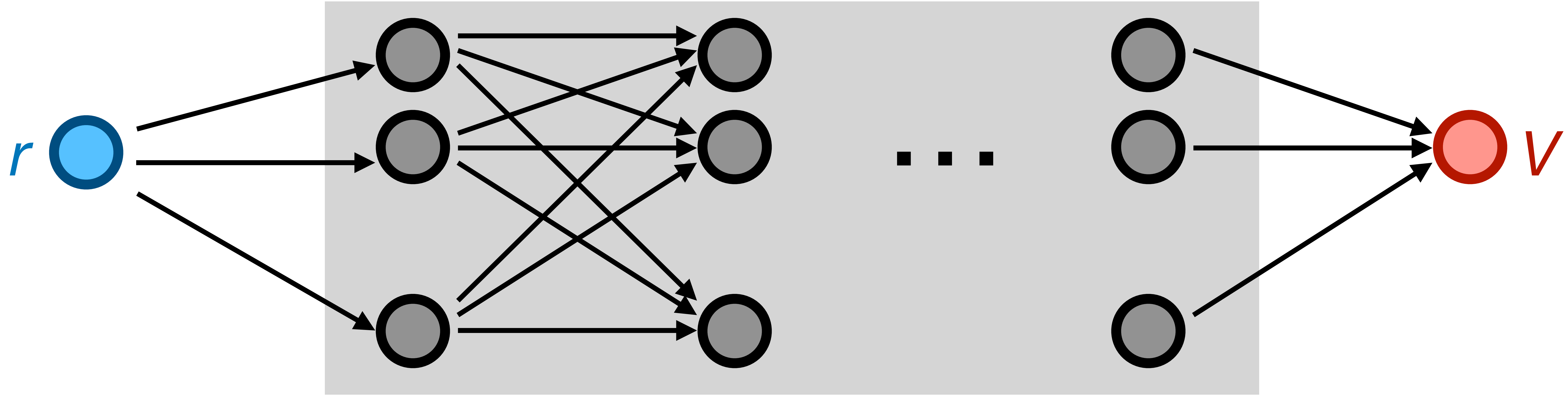
$$\tilde{P}(\boldsymbol{\theta}) = (2\pi)^{-N_\theta/2}\sqrt{\det[\Sigma^{-1}]} \times \exp\Big[-\frac{\Sigma^{-1}_{ab}}{2}(\theta_a - \theta_a^{\text{opt}})(\theta_b - \theta_b^{\text{opt}})\Big] \qquad \Sigma^{-1}_{ab} = \lambda\delta_{ab} + \frac{1}{2}\frac{\partial^2\chi^2(\boldsymbol{\theta})}{\partial\theta_a\partial\theta_b}$$

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)

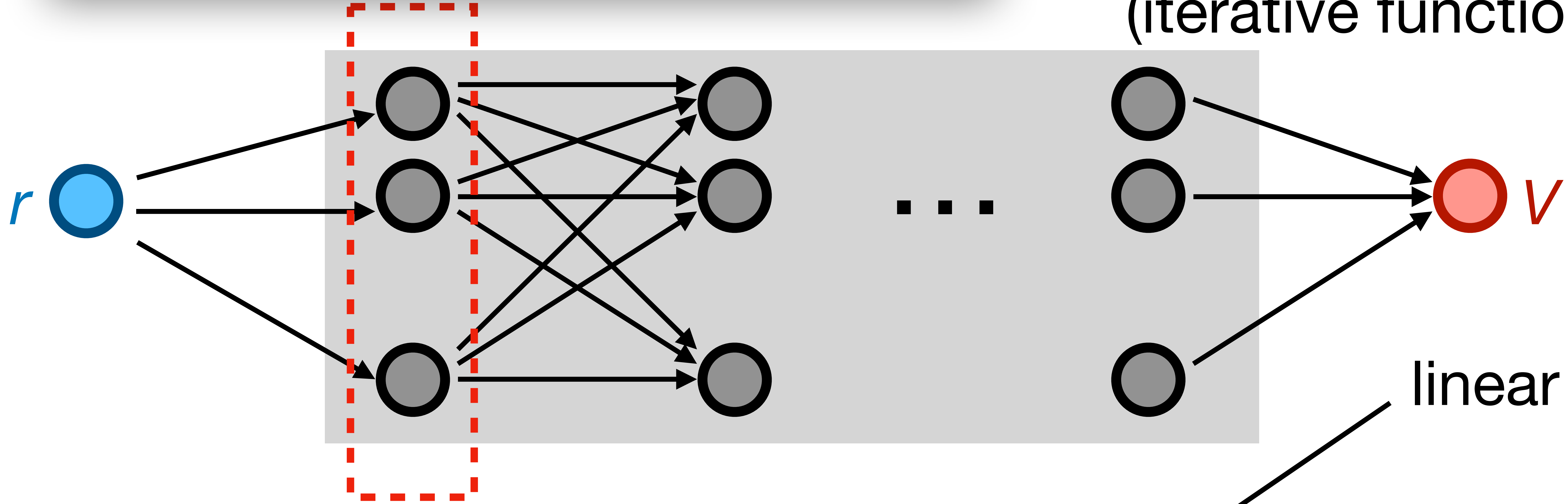


What are Deep Neural Networks?

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

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

(iterative function substitution)

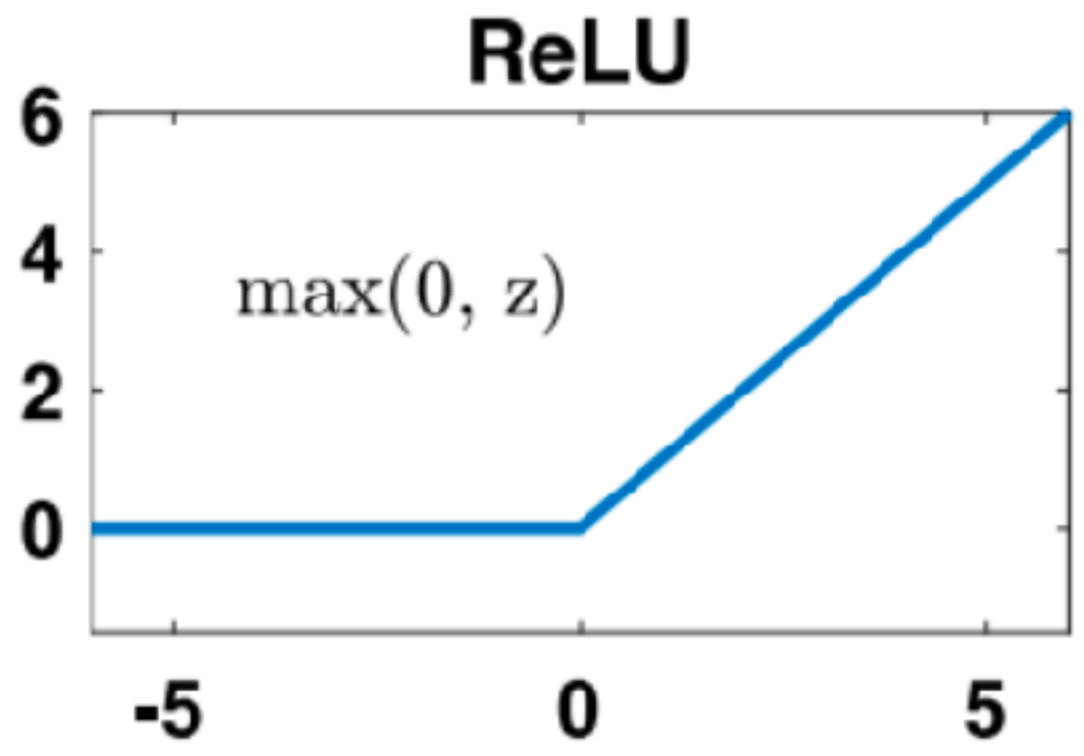


Each ● is an intermediate function ($a_i^{(l)}$):

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

linear

nonlinear

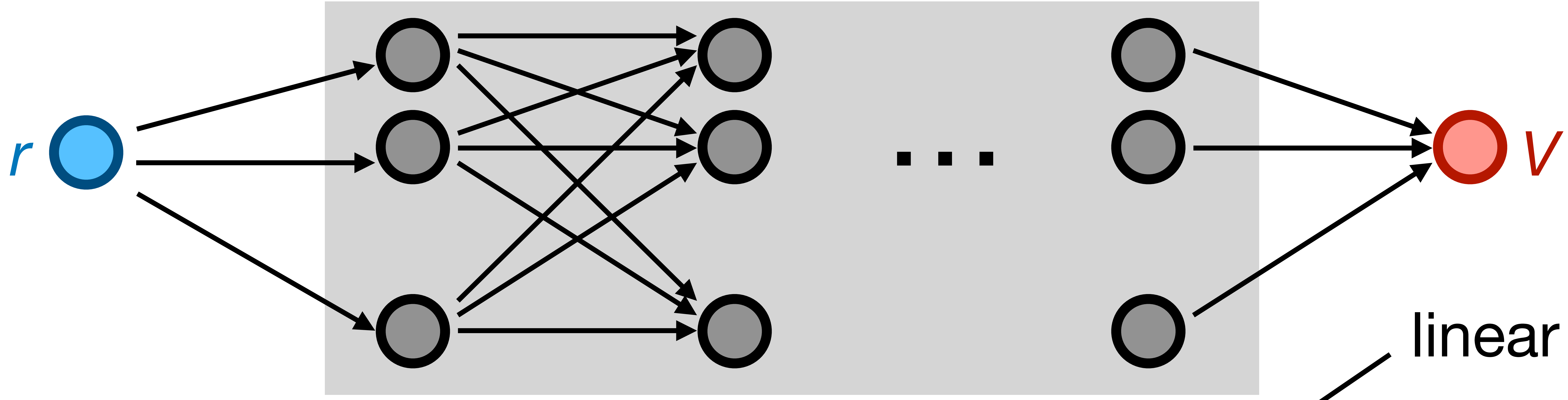


What are Deep Neural Networks?

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

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

(iterative function substitution)

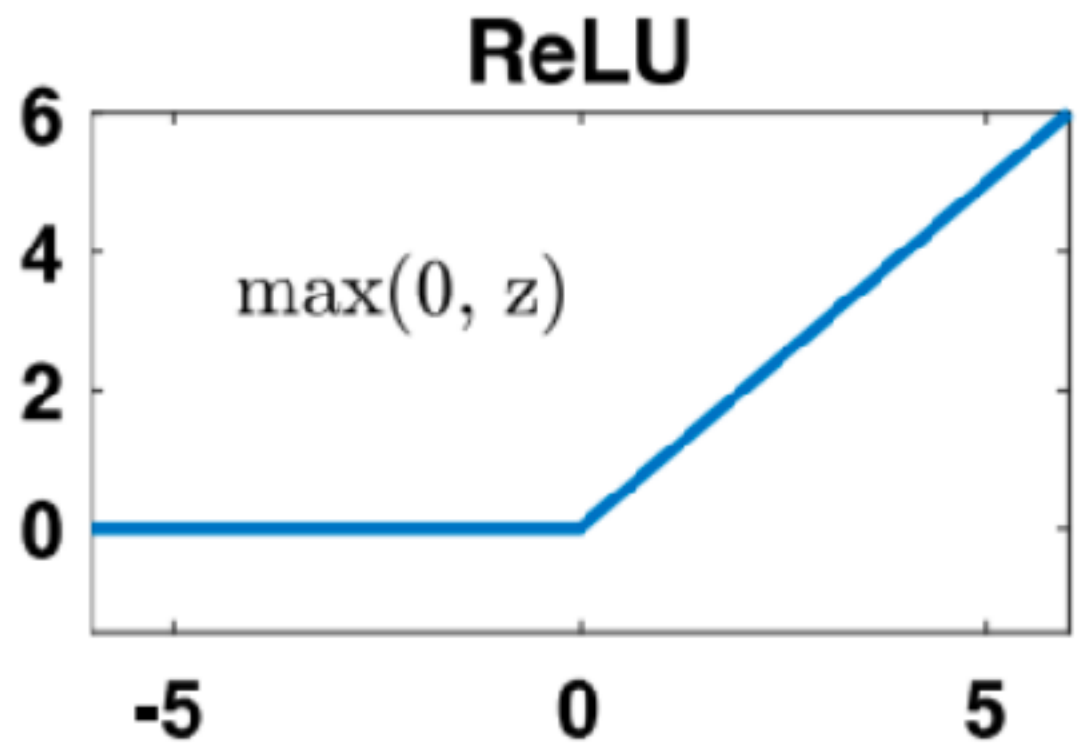


linear

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: $z_i^{(l)} = b_i^{(l)} + \sum_j W_{i,j}^{(l)}\, a_j^{(l-1)}, \qquad a_i^{(l)} = \sigma(z_i^{(l)})$

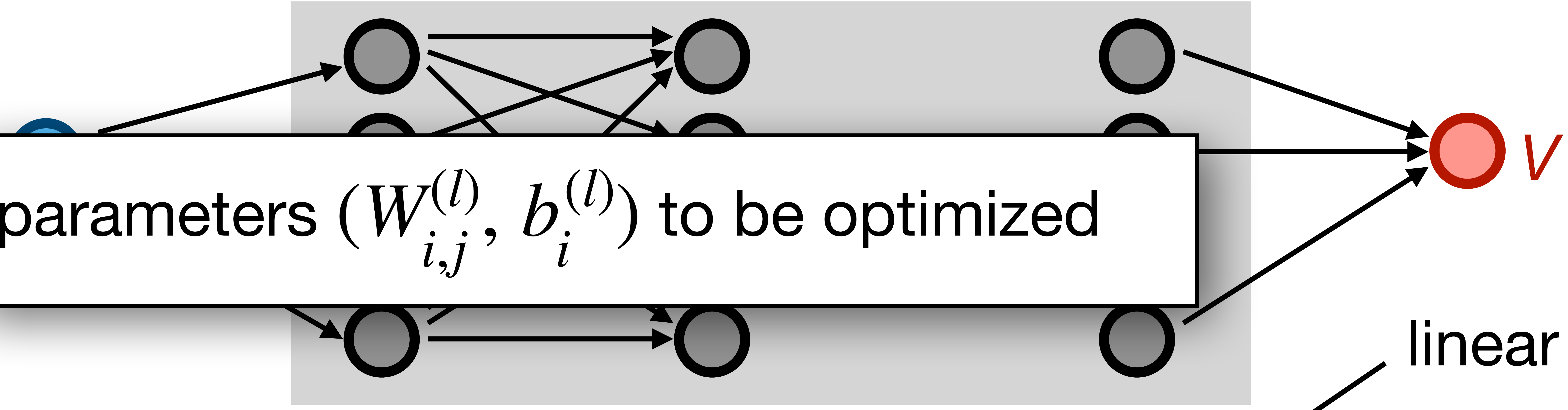


What are Deep Neural Networks?

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

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

(iterative function substitution)



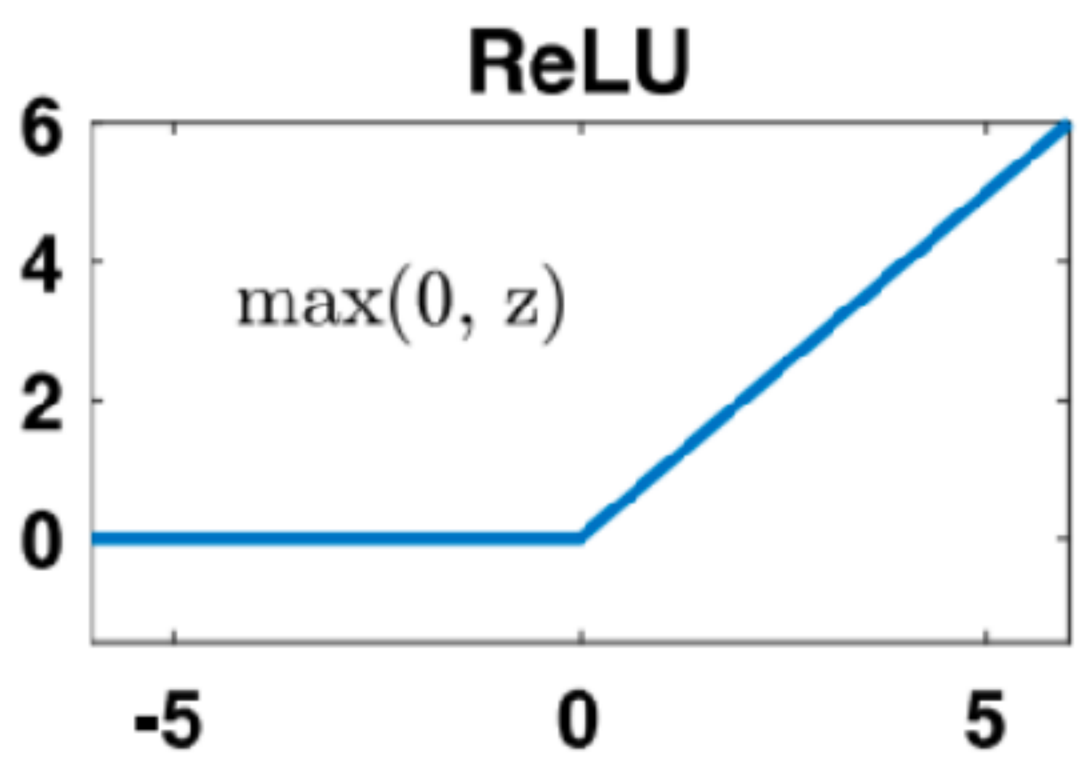
parameters ($W_{i,j}^{(l)}$, $b_i^{(l)}$) to be optimized

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: $z_i^{(l)} = b_i^{(l)} + \sum_j W_{i,j}^{(l)}\, a_j^{(l-1)}$, $\quad a_i^{(l)} = \sigma(z_i^{(l)})$



Backup

Using Deep-Learning to solve Inverse Problems

information of interest	observations
potential $V(x)$	energy level $\{E_n\}$
E. o. S. $\varepsilon(P)$	mass-radius $M(R)$

Using Deep-Learning to solve Inverse Problems

information of interest	observations
potential $V(x)$	energy level $\{E_n\} = \{E_n\}[V(r)]$
E. o. S. $\varepsilon(P)$	mass-radius $M(R) = M[\varepsilon(P)](R)$

Using Deep-Learning to solve Inverse Problems

$I(x)$

information of interest

explicit, 1-to-1 mapping

exist, but implicit

$\vec{O}(y) = \vec{O}[I(x)](y)$

observations

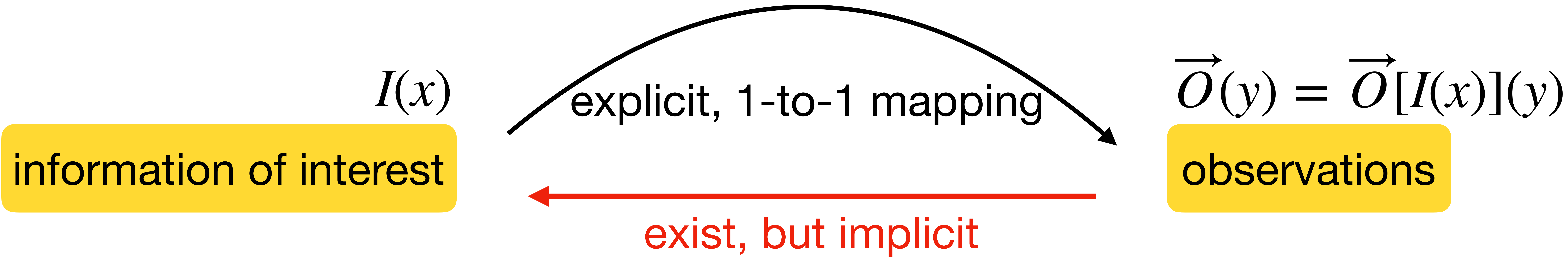
potential $V(x)$

energy level $\{E_n\} = \{E_n\}[V(r)]$

E. o. S. $\varepsilon(P)$

mass-radius $M(R) = M[\varepsilon(P)](R)$

Using Deep-Learning to solve Inverse Problems (Type A)

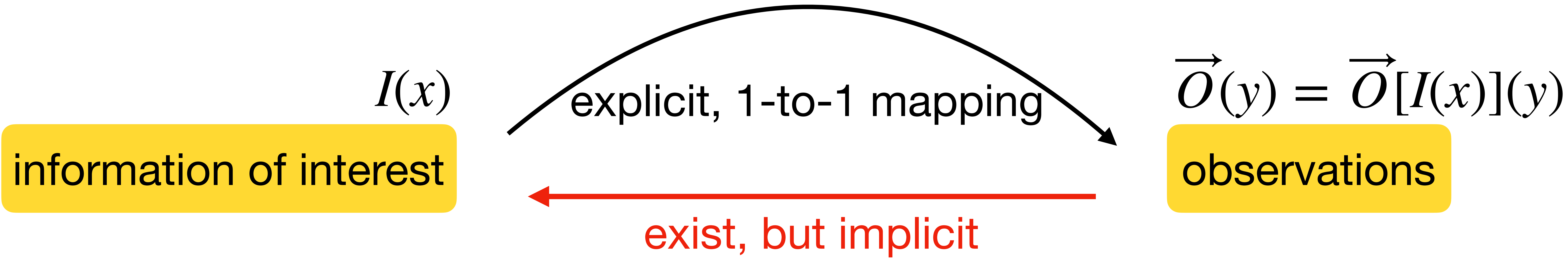


prepare data set $\{I_i \to \vec{O}_i\}$***, use NN to learn the inverse function(al)*** $\{\vec{O} \to I\}$

- dimension of input?
- completeness?
- bias/prior?
- uncertainty?

ref: Yuki's seminar this Monday

Using Deep-Learning to solve Inverse Problems (Type B)



$I(x) \equiv I_{\mathrm{DNN}}(x \,|\, \boldsymbol{\theta})$

$\vec{O}_{\boldsymbol{\theta}}(y) \equiv \vec{O}[I_{\mathrm{DNN}}(x \,|\, \boldsymbol{\theta})](y)$

Using Deep-Learning to solve Inverse Problems (Type B)

$I(x)$

information of interest

explicit, 1-to-1 mapping

exist, but implicit

$\vec{O}(y) = \vec{O}[I(x)](y)$

observations

$$I(x) \equiv I_{\text{DNN}}(x \,|\, \boldsymbol{\theta})$$

$$\vec{O}_{\boldsymbol{\theta}}(y_i) \equiv \vec{O}[I_{\text{DNN}}(x \,|\, \boldsymbol{\theta})](y_i)$$

loss: dist. bt/ ***reconstructed*** and ***true*** obsv.

$$J_{\boldsymbol{\theta}} \equiv \sum_i \left| \vec{O}_{\boldsymbol{\theta}}(y_i) - \vec{O}(y_i) \right|^2$$

Using Deep-Learning to solve Inverse Problems (Type B)

$I(x)$

information of interest

explicit, 1-to-1 mapping

$\vec{O}(y) = \vec{O}[I(x)](y)$

observations

exist, but implicit

$$I(x) \equiv I_{\text{DNN}}(x\,|\,\boldsymbol{\theta})$$

$$\vec{O}_{\boldsymbol{\theta}}(y_i) \equiv \vec{O}[I_{\text{DNN}}(x\,|\,\boldsymbol{\theta})](y_i)$$

$$J_{\boldsymbol{\theta}} \equiv \sum_i \left| \vec{O}_{\boldsymbol{\theta}}(y_i) - \vec{O}(y_i) \right|^2$$

Tune $\boldsymbol{\theta}$ to minimize $J_{\boldsymbol{\theta}}$. Prerequisite: $\dfrac{\delta O(y)}{\delta I(x)}$ can be computed

$$\nabla_{\boldsymbol{\theta}} J = 2 \sum_i \left[O_{\boldsymbol{\theta}}(y_i) - O(y_i) \right] \int dx \, \frac{\delta O(y)}{\delta I(x)} \bigg|_{O(y)=O_{\boldsymbol{\theta}}(y_i)} \nabla_{\boldsymbol{\theta}} I_{\text{DNN}}(x\,|\,\boldsymbol{\theta})$$

Problem # III:

Neutron Star: EoS <=> {M,R}

reference: Soma, Wang, ***SS***, Stoecker, Zhou,
JCAP08 (2022) 071; 2209.08883; in progress.

2. Neutron Star: EoS <=> mass-radius relation

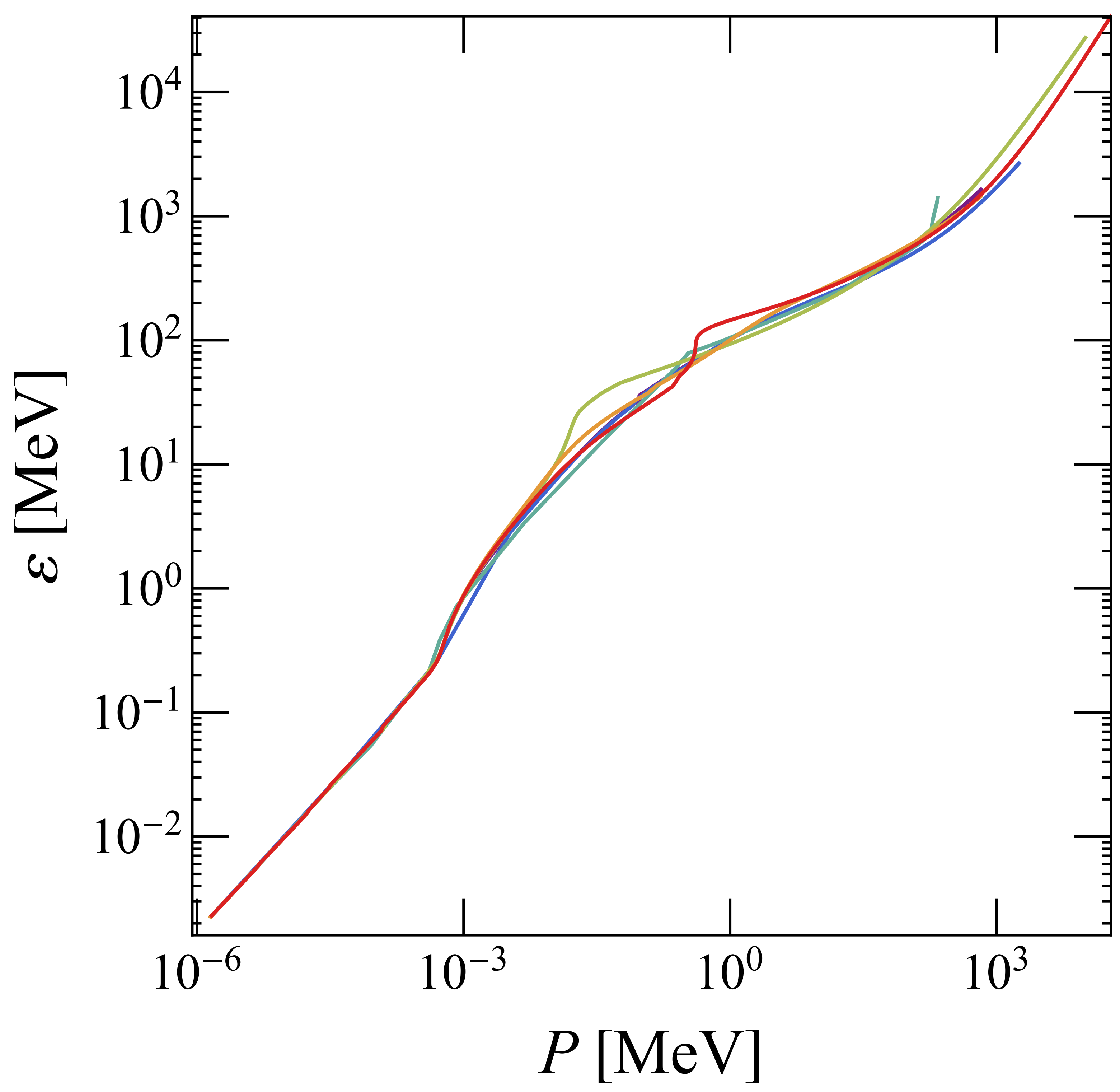
TOV equations:

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

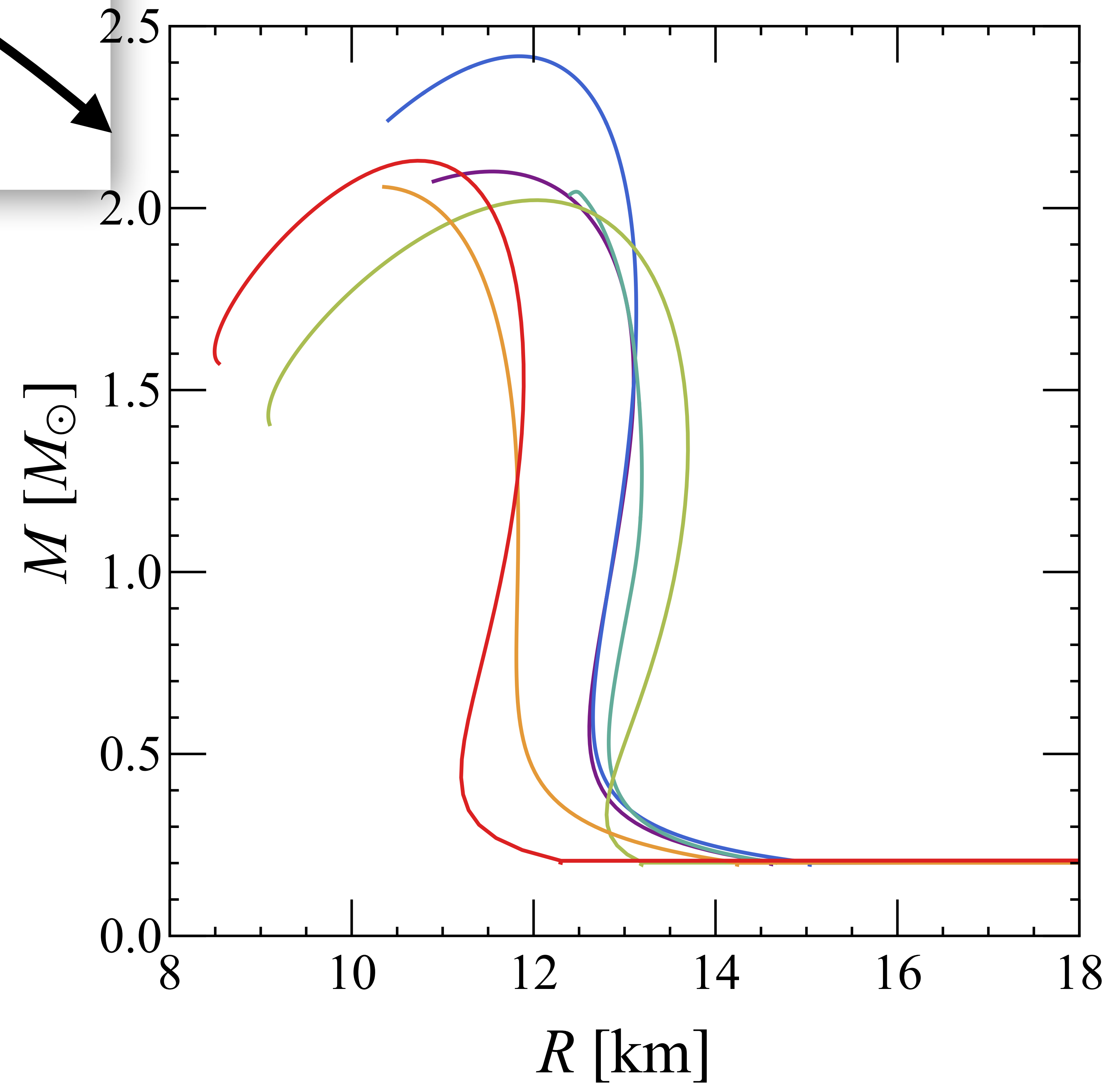
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$

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

$\varepsilon(P)$



$M(R)$



2. Neutron Star: EoS <=> mass-radius relation

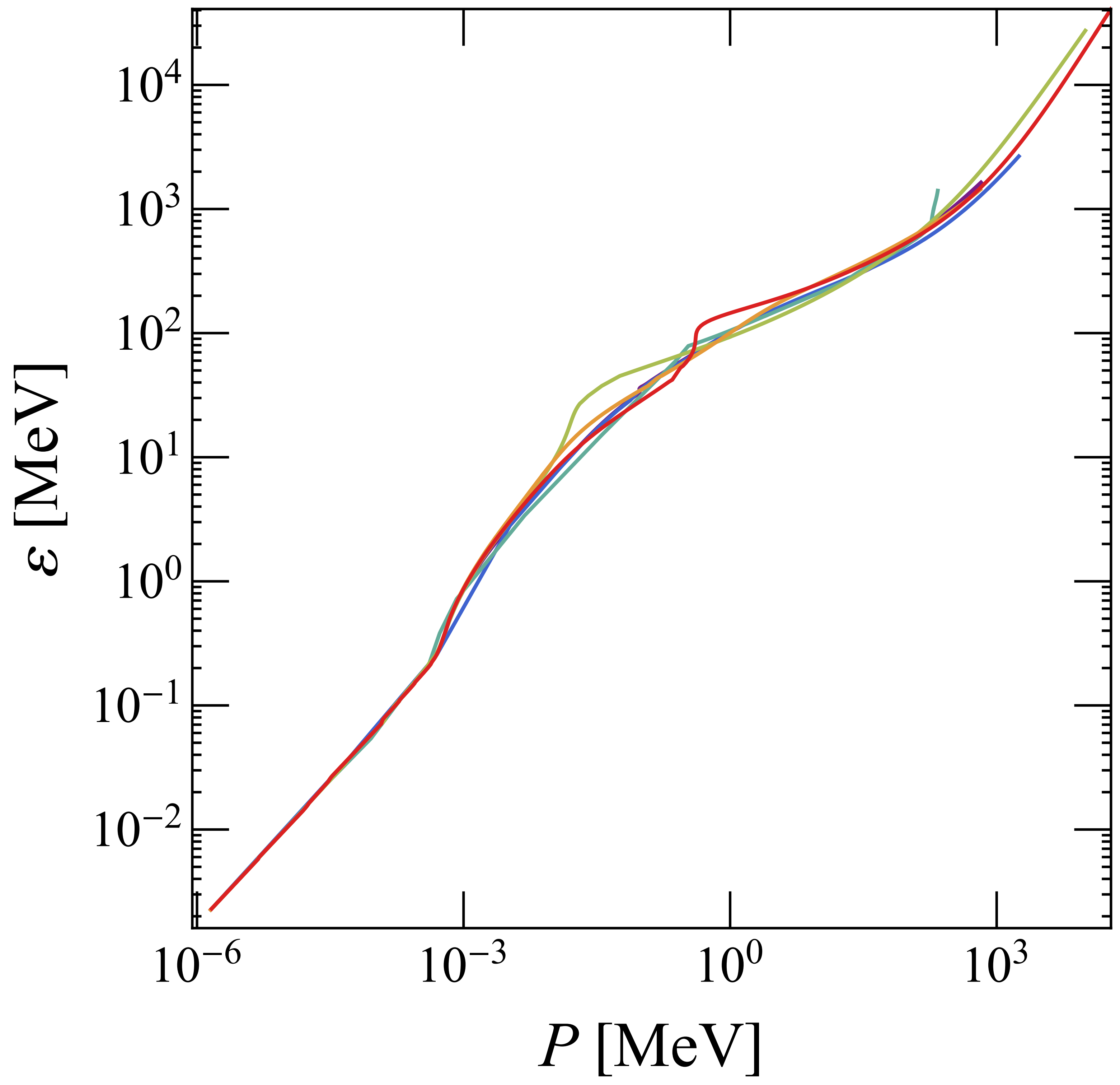
TOV equations:

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

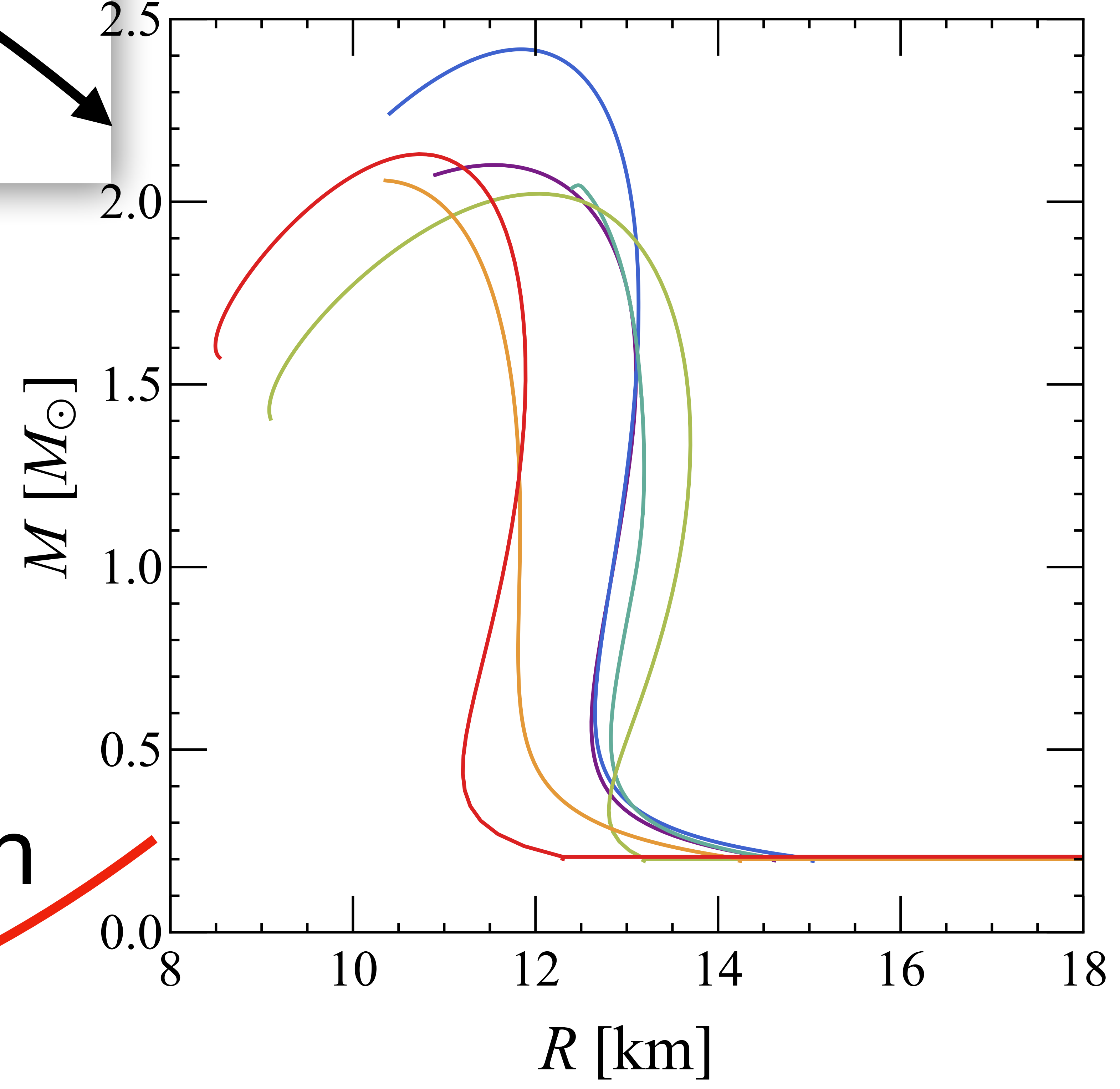
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$

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

$\varepsilon(P)$



$M(R)$



Lindblom's algorithm
if the whole M(R) is known

2. Neutron Star: EoS <=> mass-radius relation

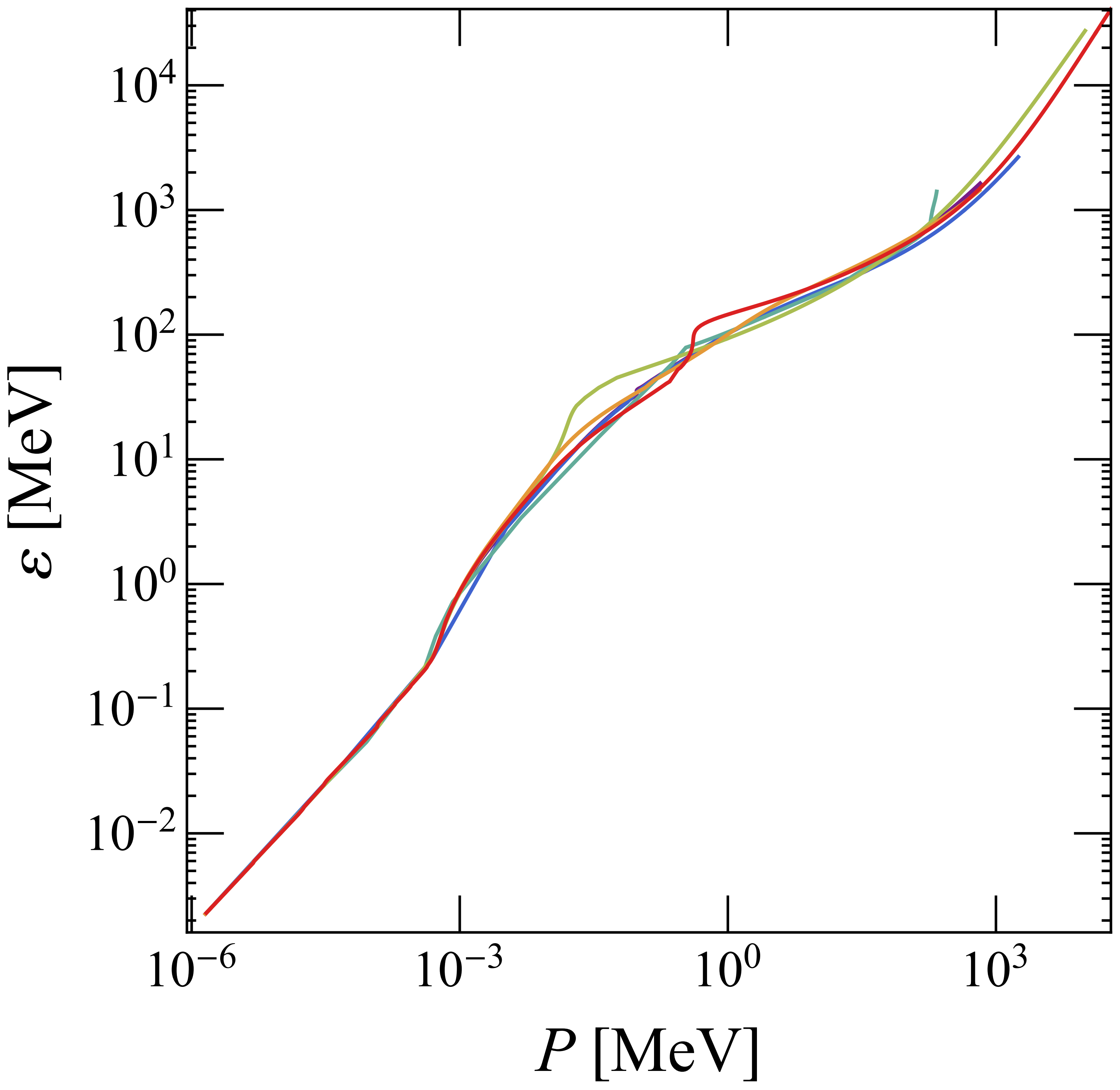
TOV equations:

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

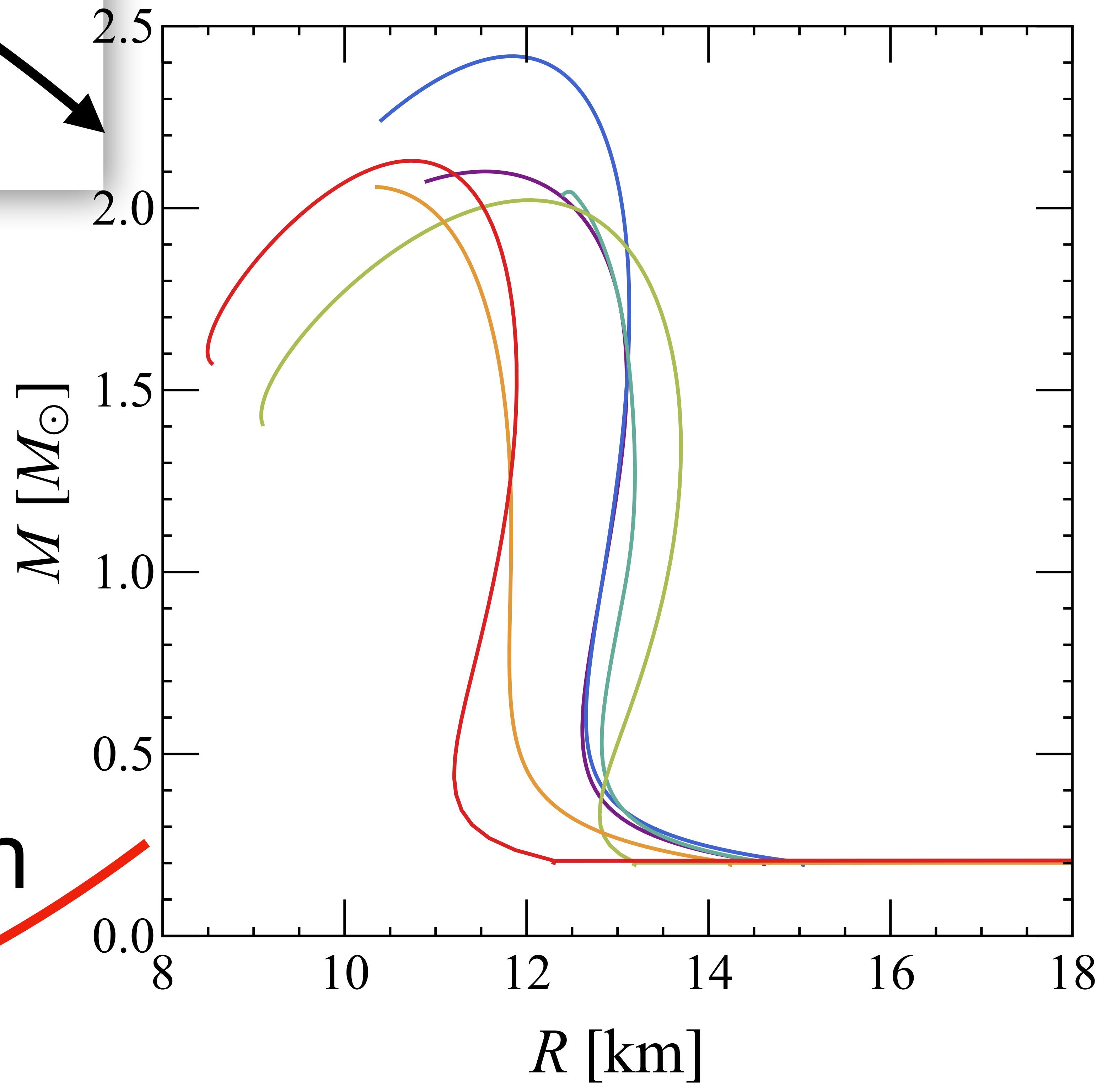
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2 \varepsilon\,,$$

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

$\varepsilon(P)$



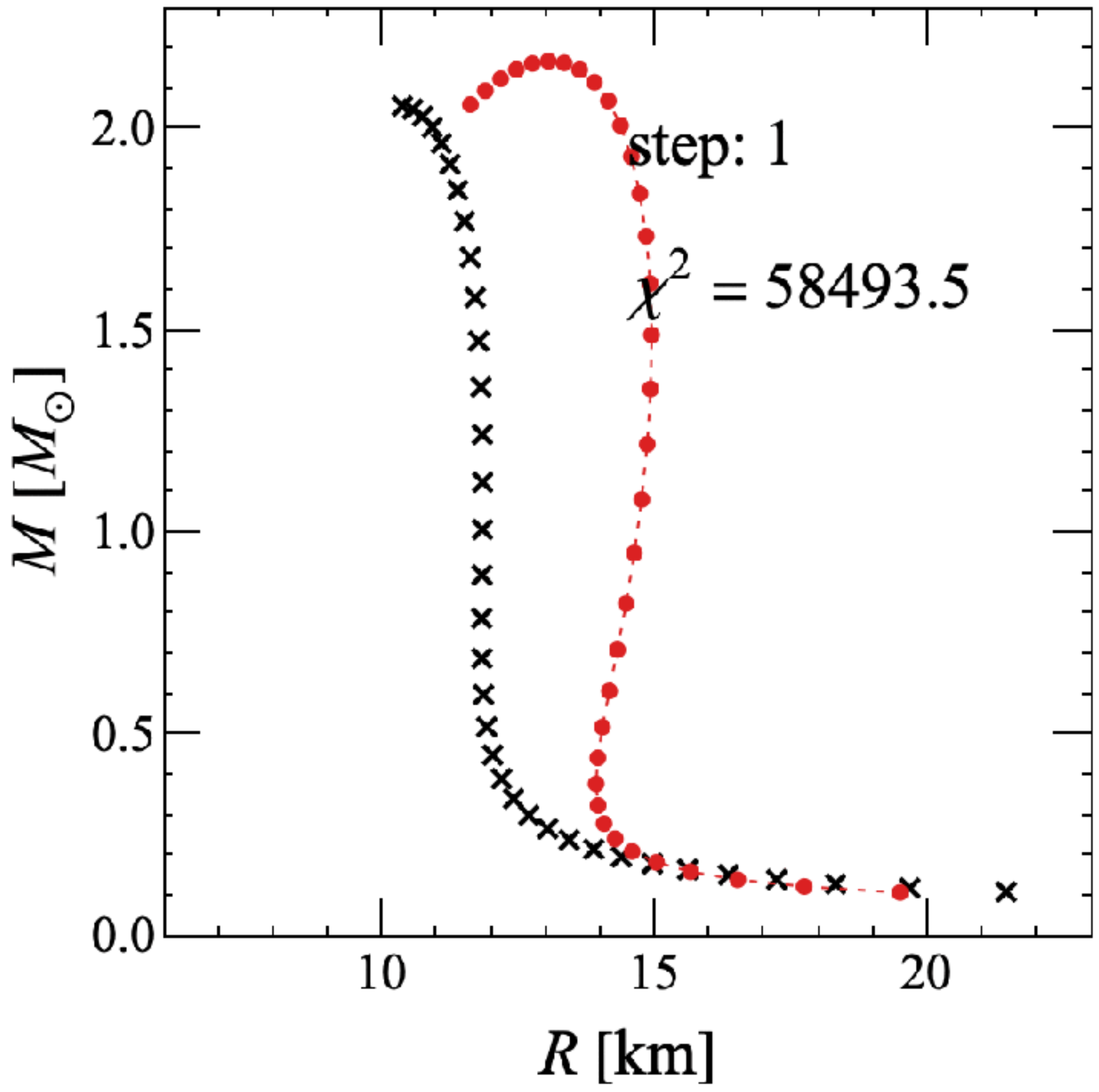
$M(R)$

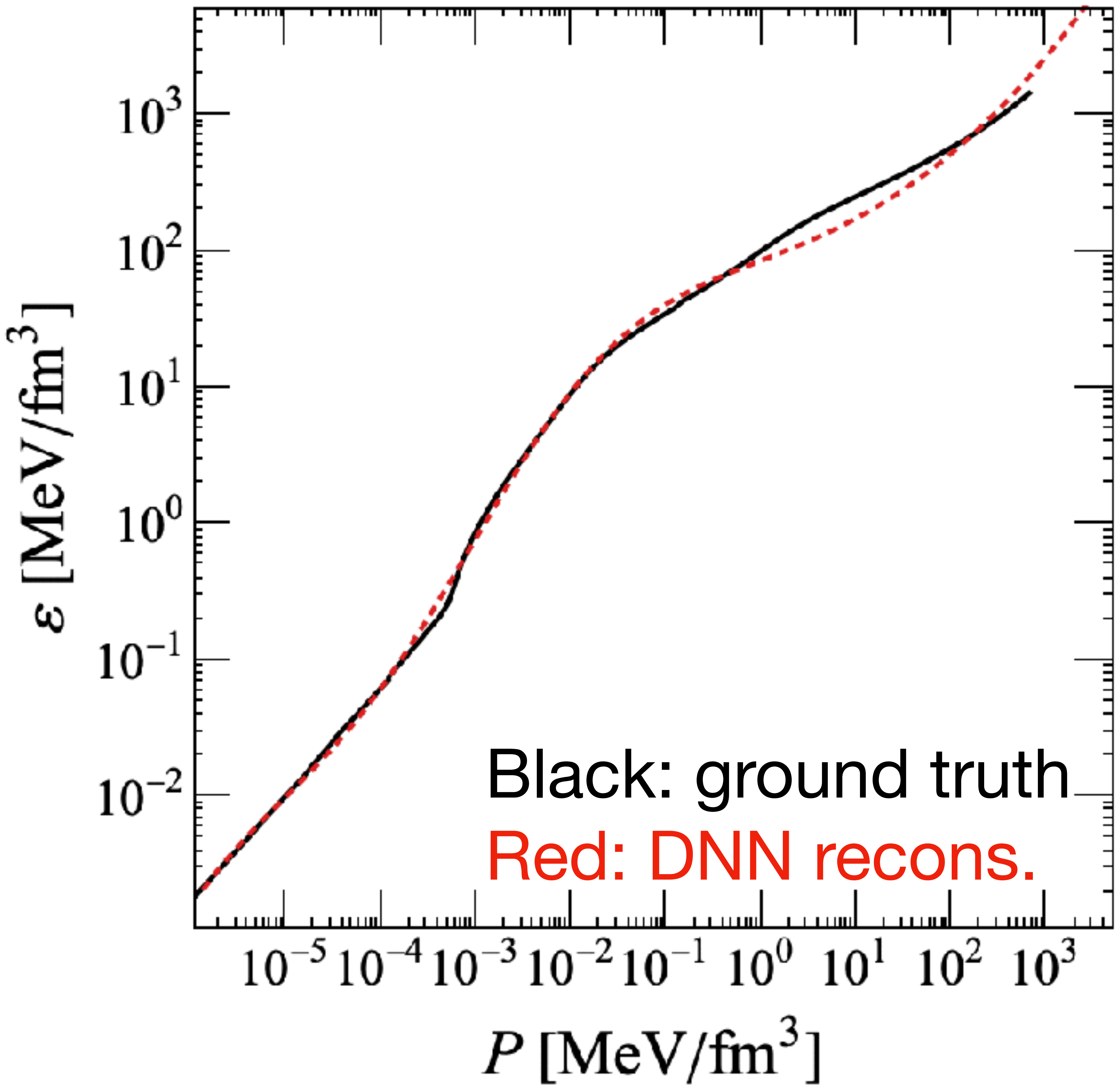


Discrete {M,R} observations?

Lindblom's algorithm if the whole M(R) is known

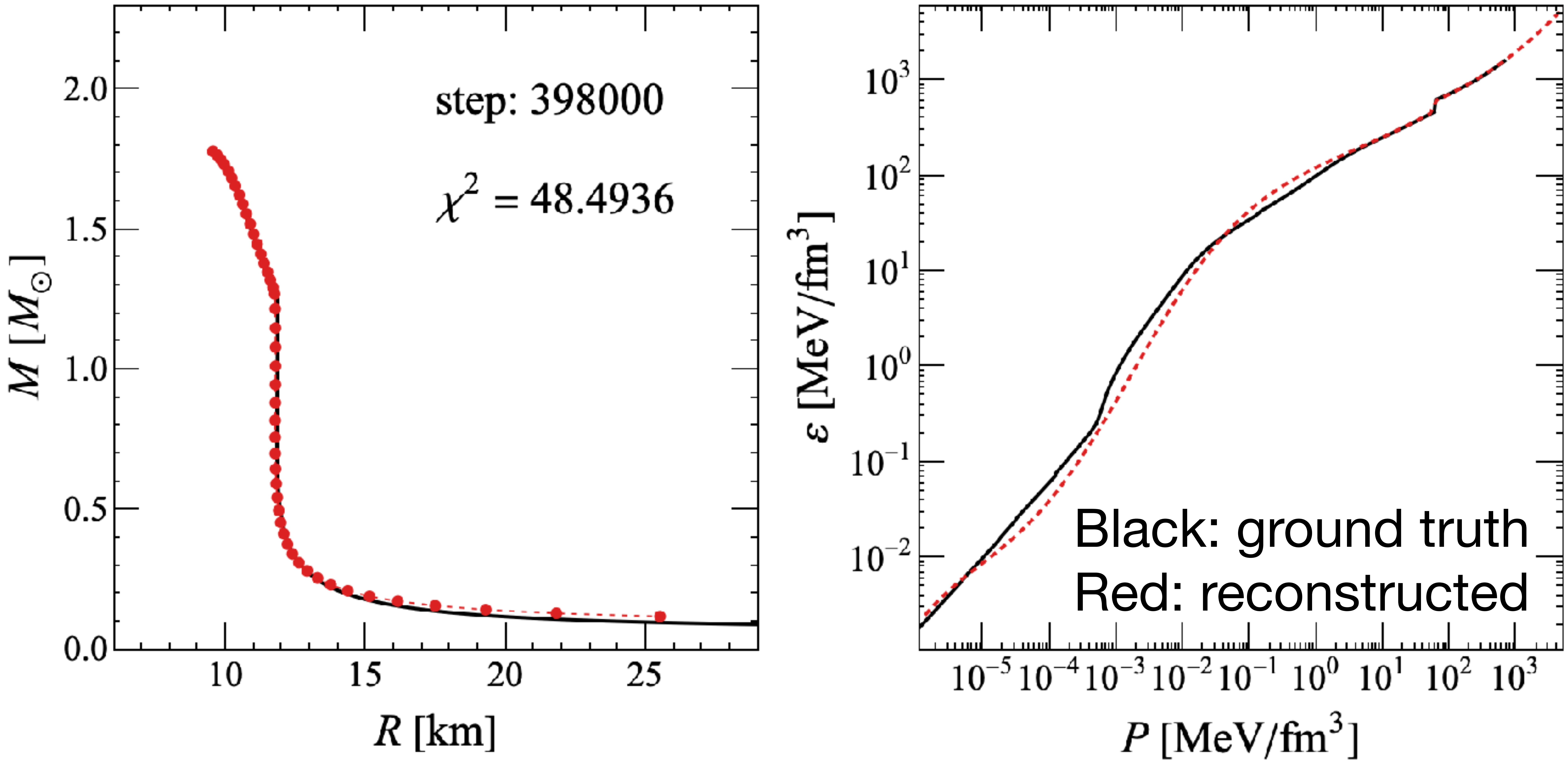
Closure test: can we learn EoS from $\{m_i, R_i\}$?





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

can we learn EoS w/ phase transition?



EoS reconstruction from real observations