Fonds de recherche sur la nature et les technologies Québec

McGill UNIVERSITY

LATTICE 21
JULY 26-30 2021, ZOOM/GATHER@MIT

Date: 2021 - July - 27

From lattice QCD (mass & thermal width) to in-medium heavy-quark interactions via deep learning

Passenger: Shuzhe Shi (McGill University)

Travel with:
Kai Zhou, Jiaxing Zhao, Swagato Mukherjee, and Pengfei Zhuang

Ticket #: (arXiv)2105.07862

Quarkonium in the Quark-Gluon Plasma

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

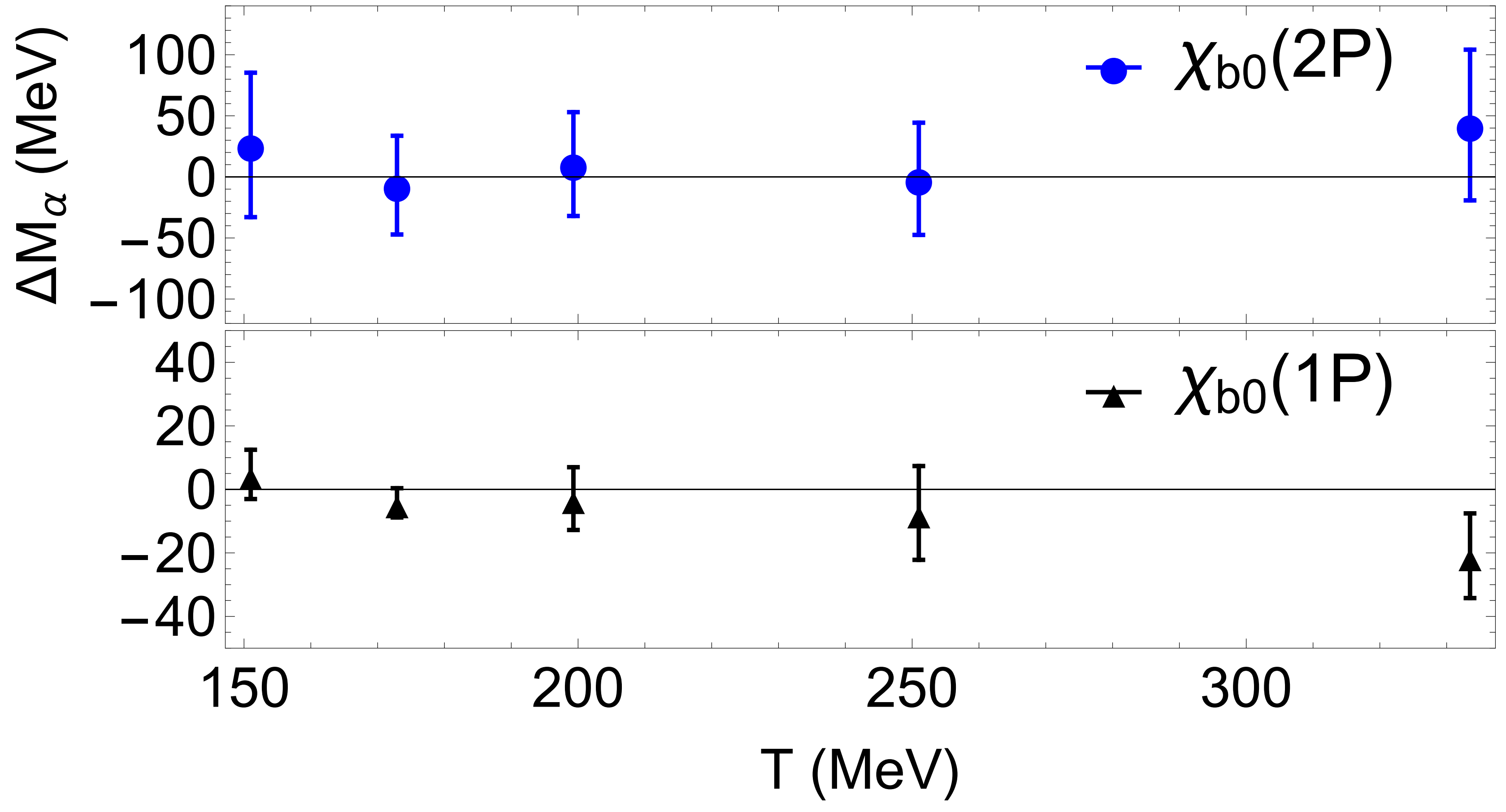
- Accurate understanding of the in-medium heavy-quark interaction?
  - Real potential modified by color-screening
  - Imaginary potential arises due to $(Q\bar{Q})_1 \to (Q\bar{Q})_8$, Landau damping, ...

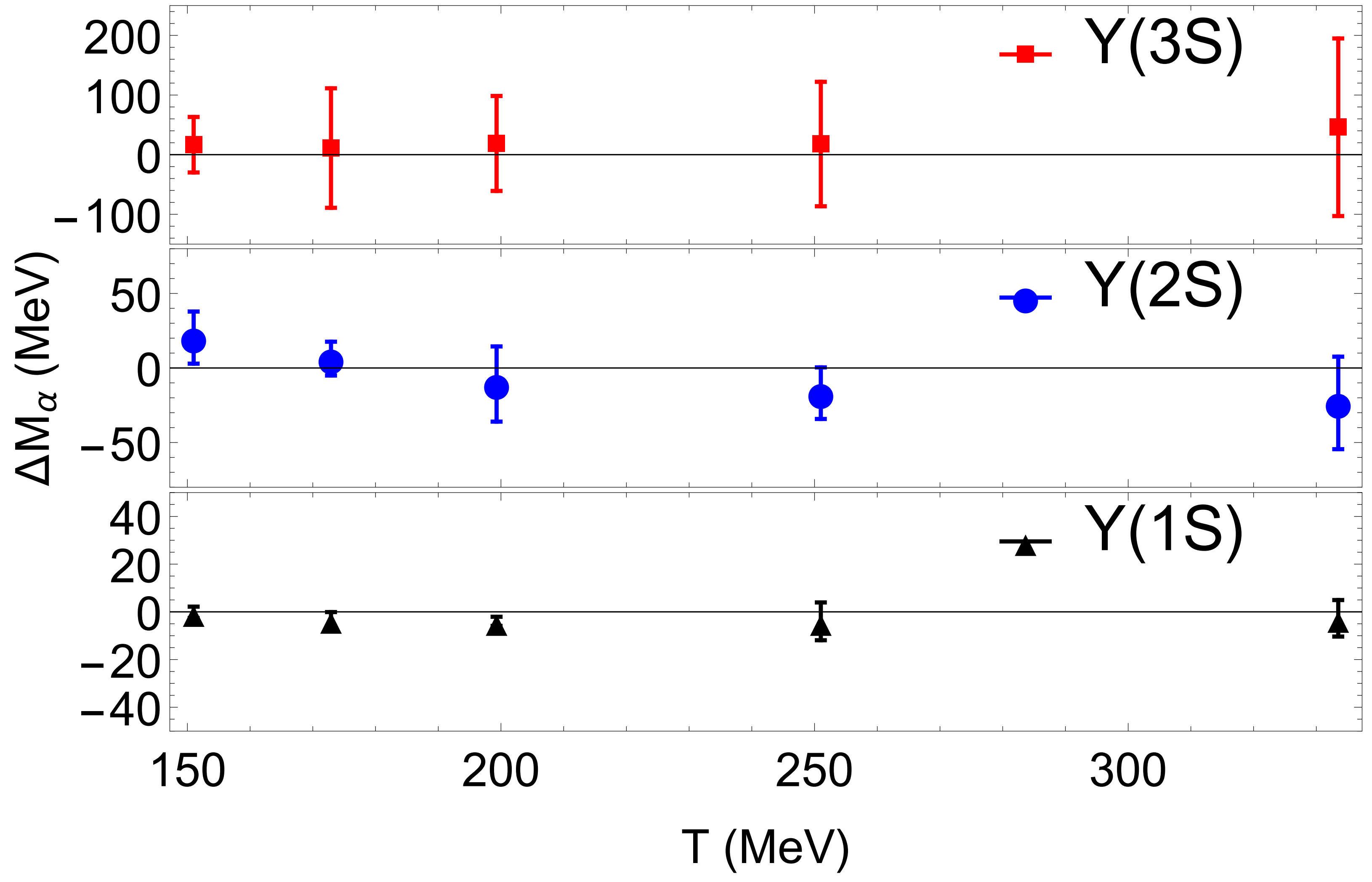
- Many efforts from the lattice-QCD community (potential for static quarks)

=> in this work: a new way to extract the potential

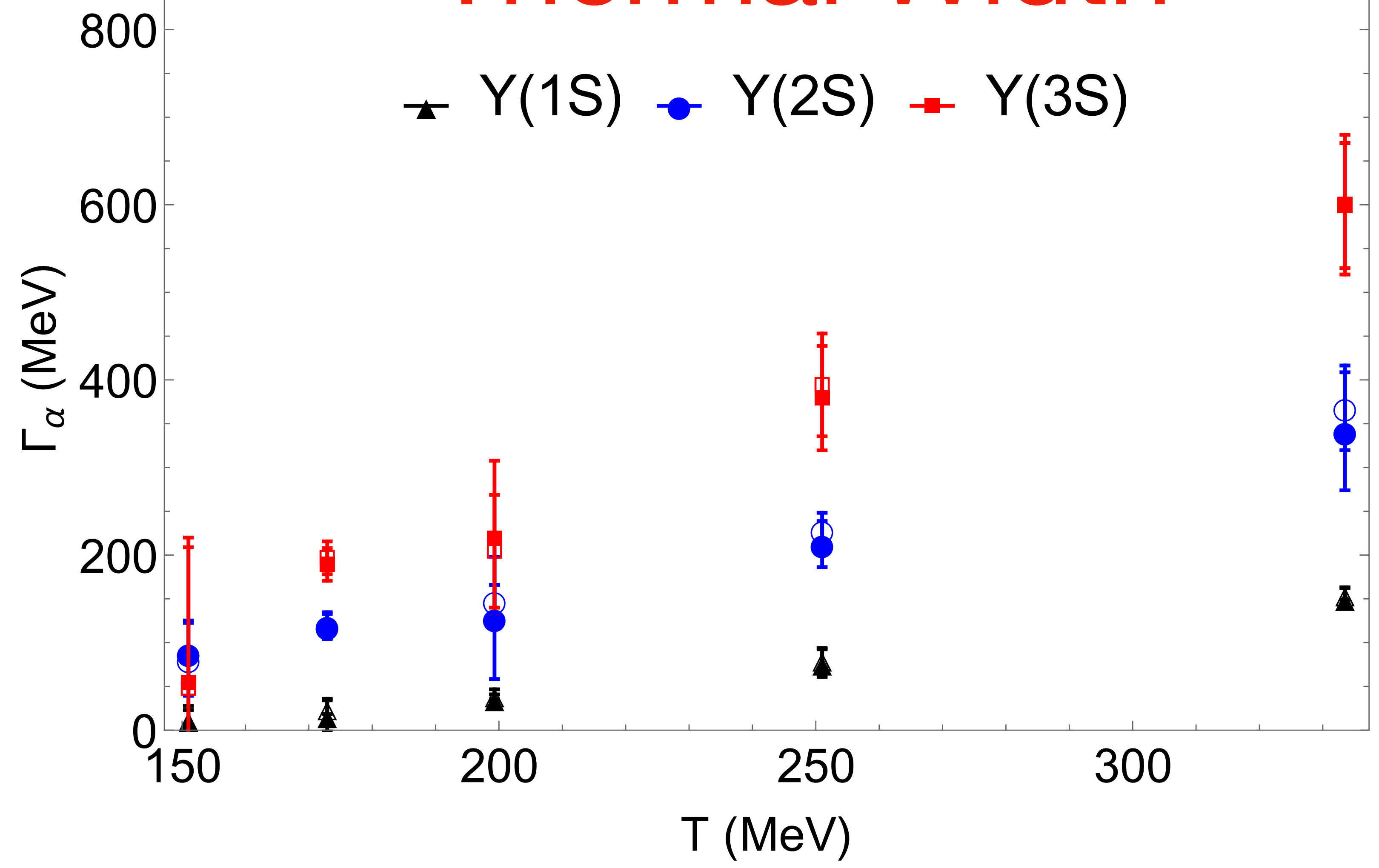
Bottomonium mass and thermal width, lattice QCD with finite m_Q

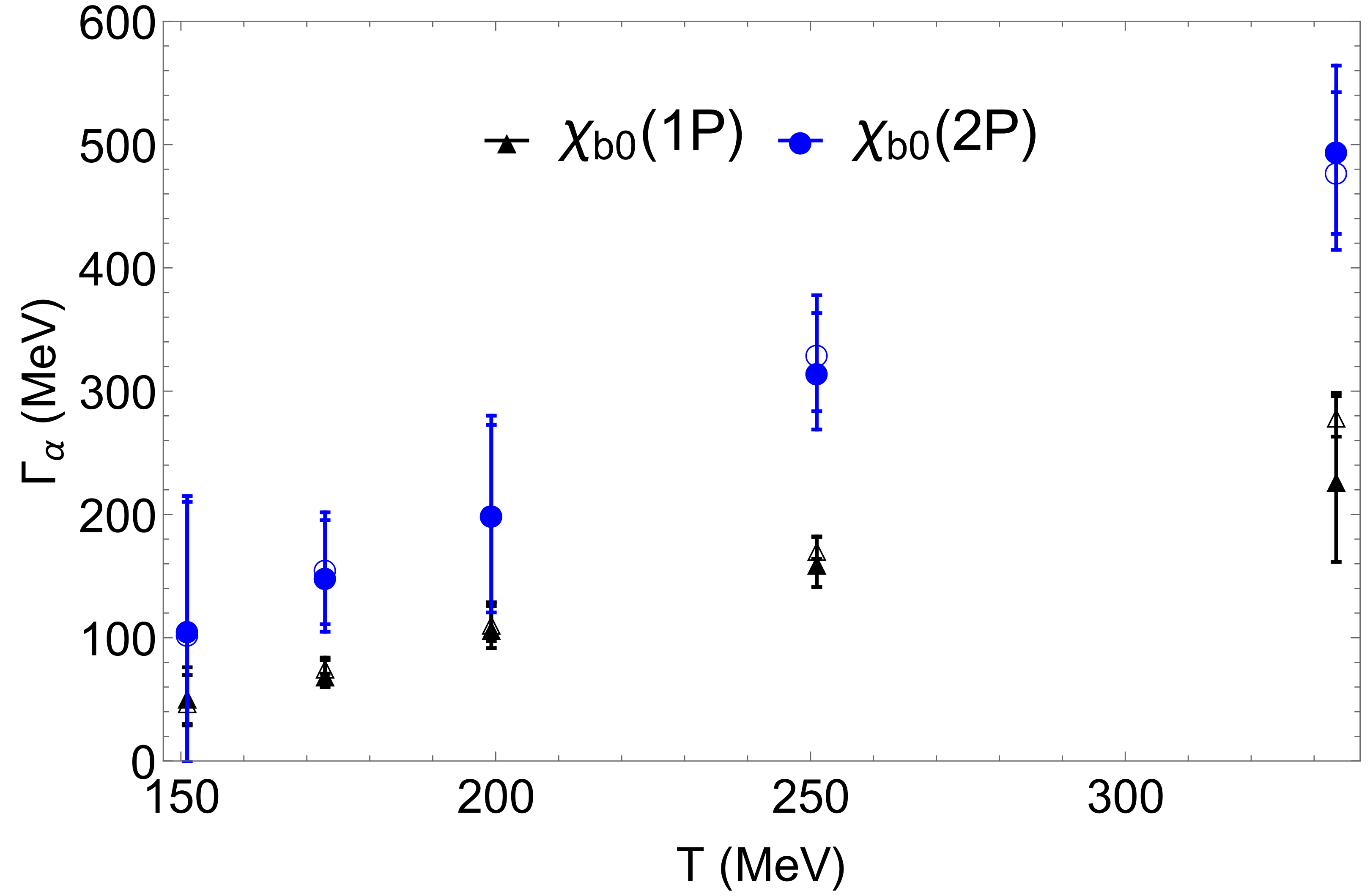
Mass





Thermal Width

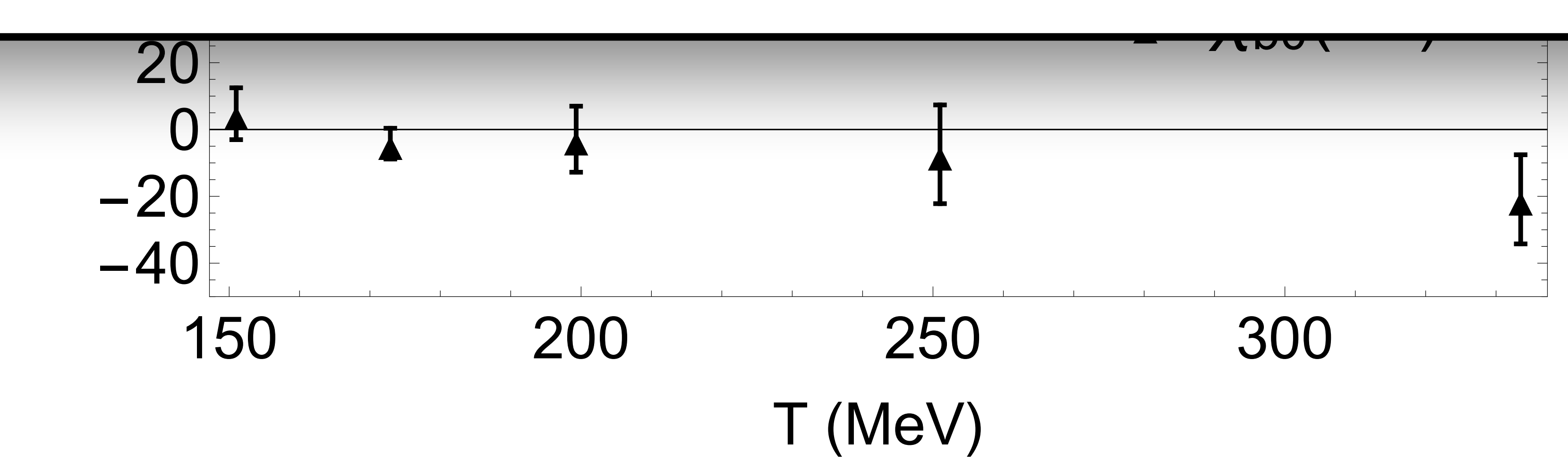


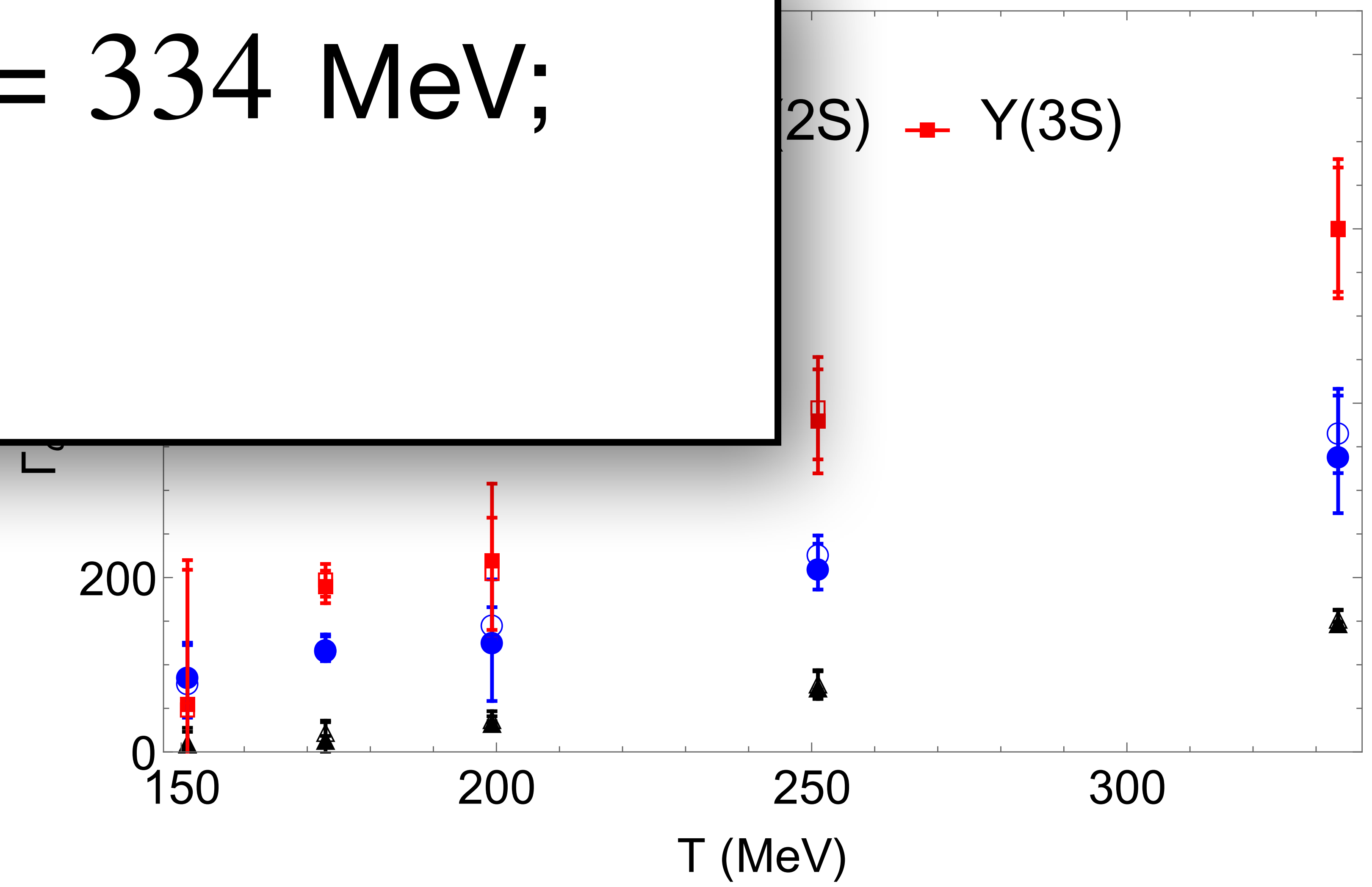


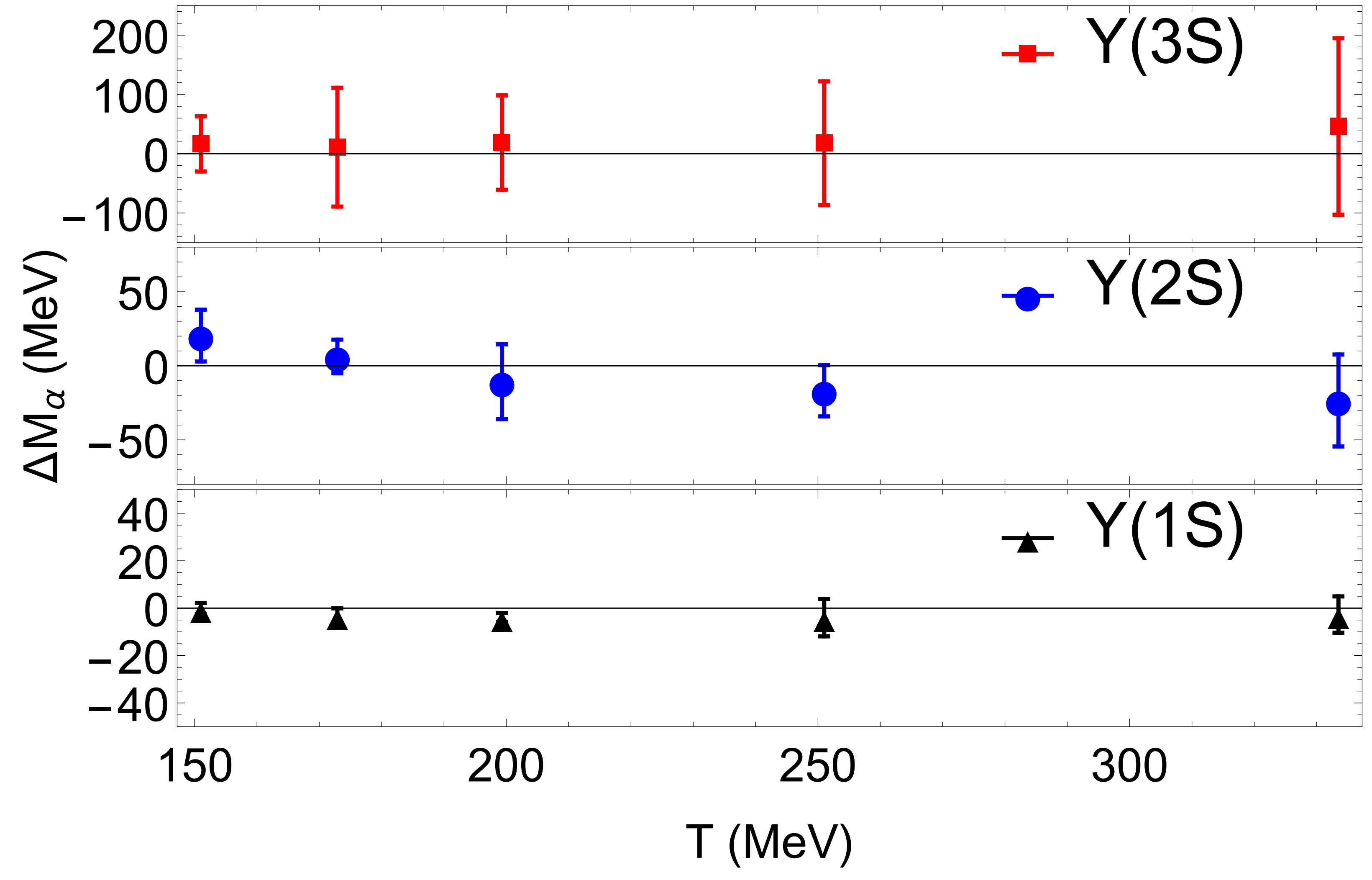
R. Larsen, S. Meinel, S. Mukherjee, and P. Petreczky:
Phys.Rev.D100,074506(2019), Phys.Lett.B800,135119(2020), Phys.Rev.D102,114508(2020)

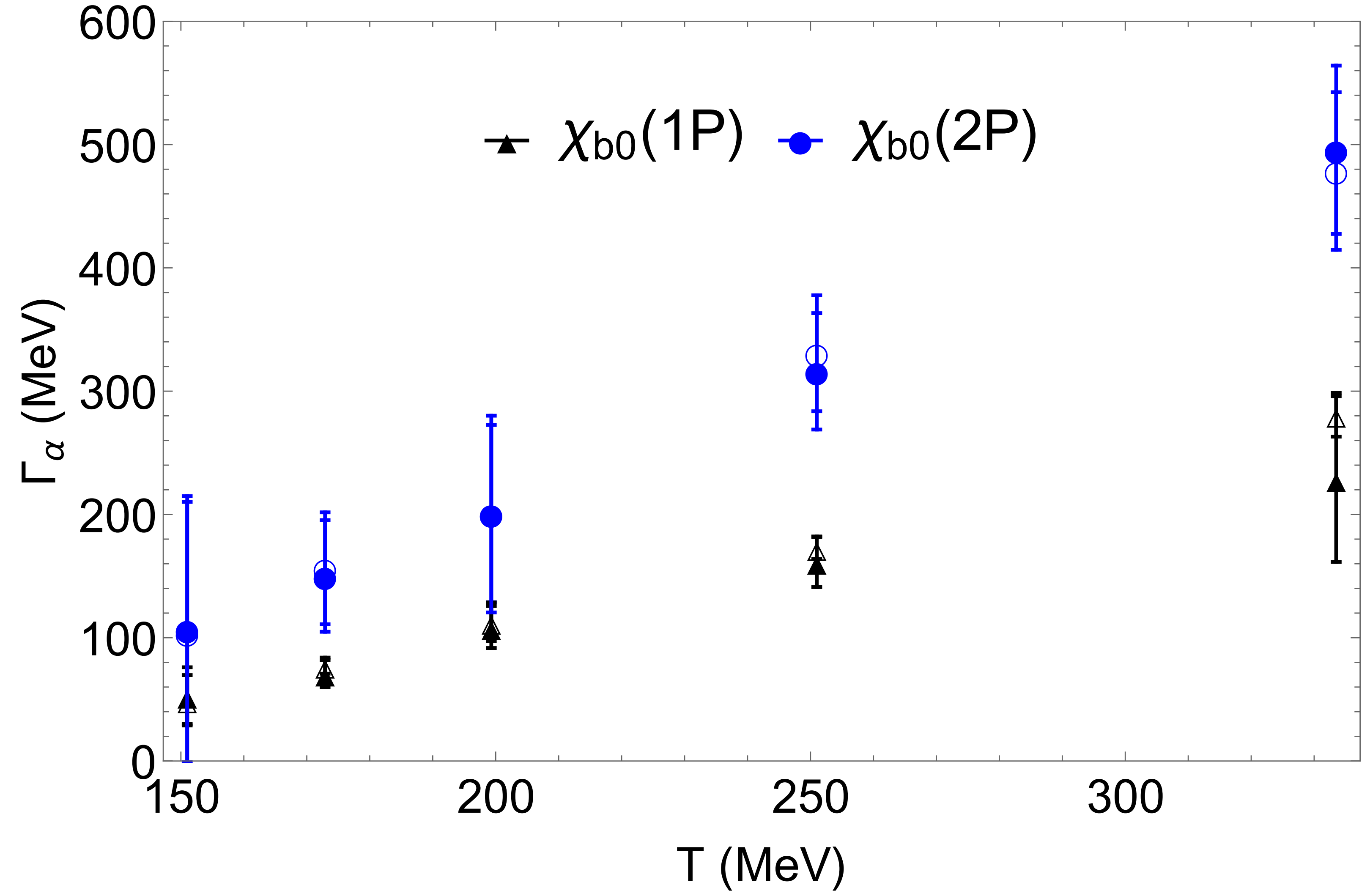
Bottomonium mass and thermal width, lattice QCD with finite m_Q

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





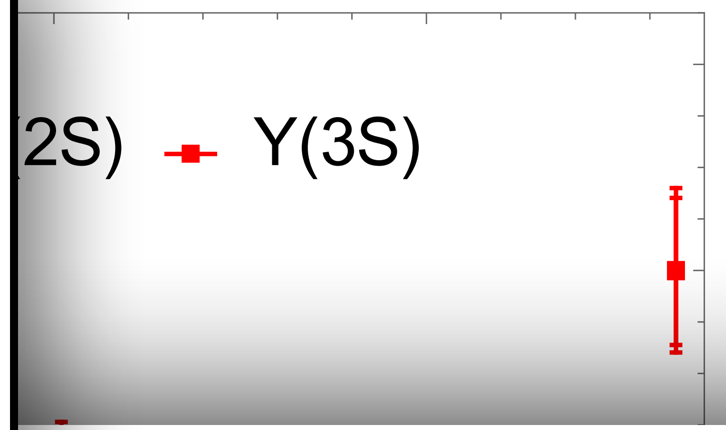


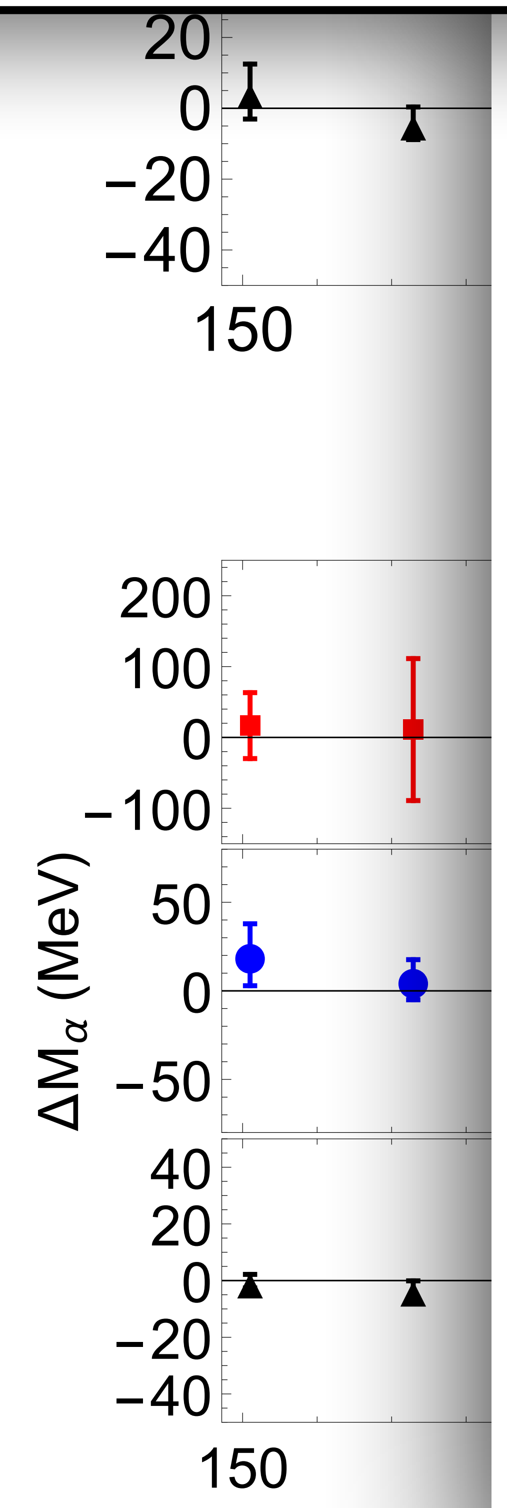


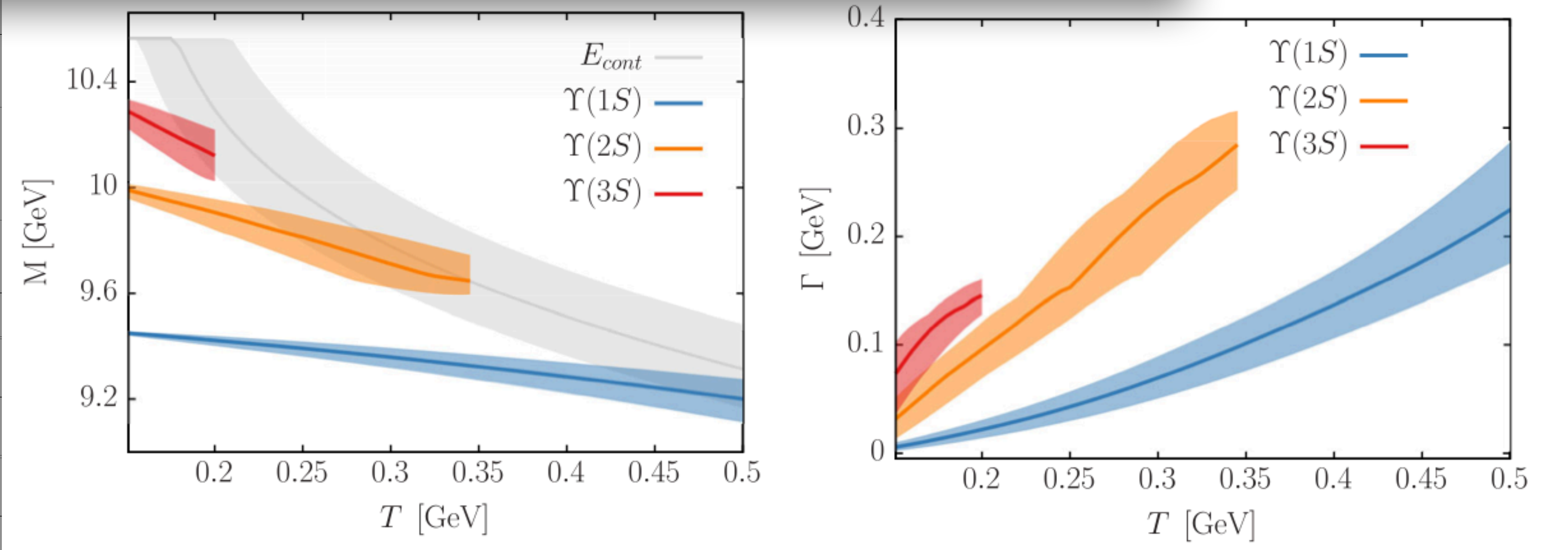
R. Larsen, S. Meinel, S. Mukherjee, and P. Petreczky:
Phys.Rev.D100,074506(2019), Phys.Lett.B800,135119(2020), Phys.Rev.D102,114508(2020)

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.





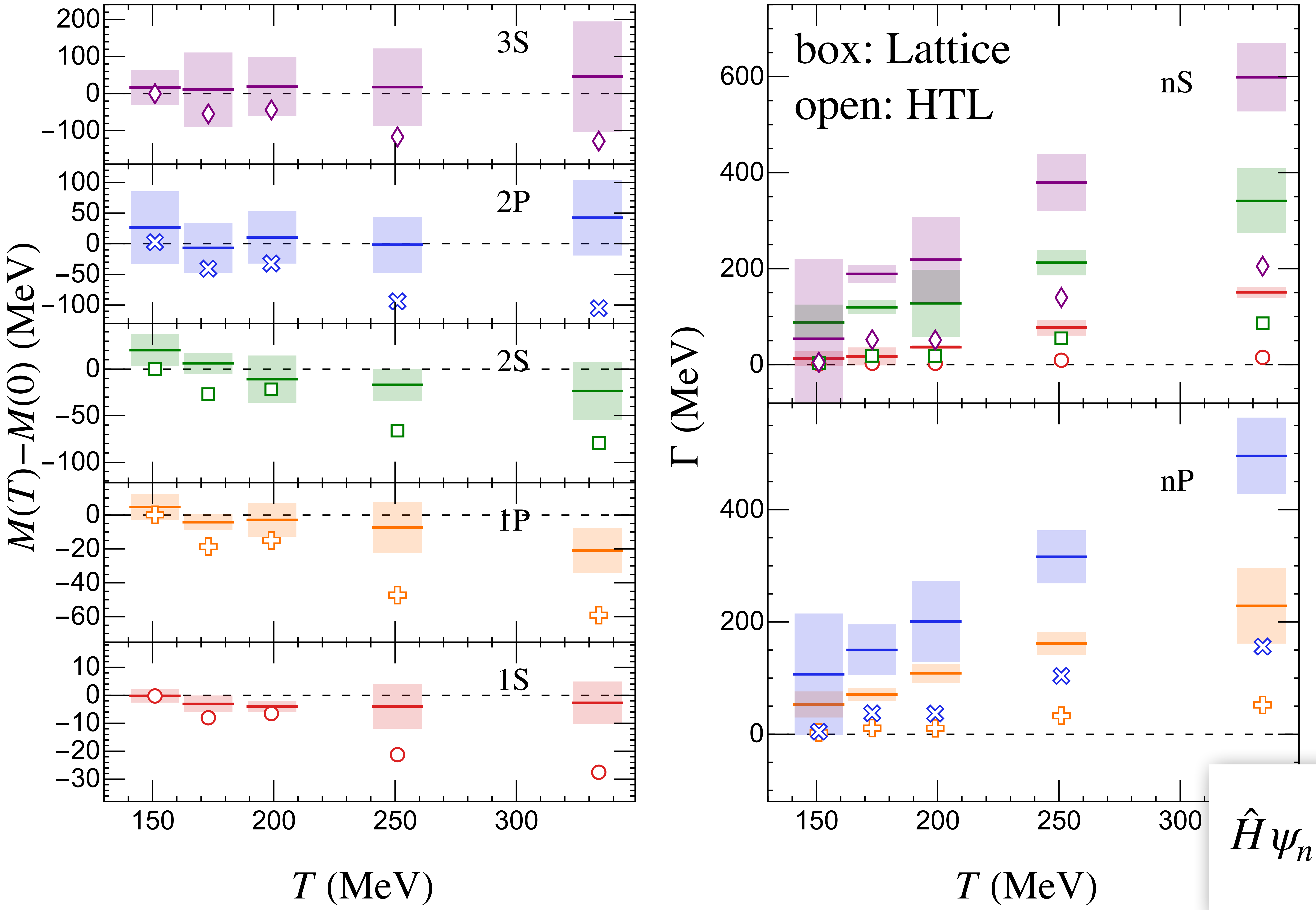


D. Lafferty and A. Rothkopf, PhysRevD.101.056010(2020)

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?

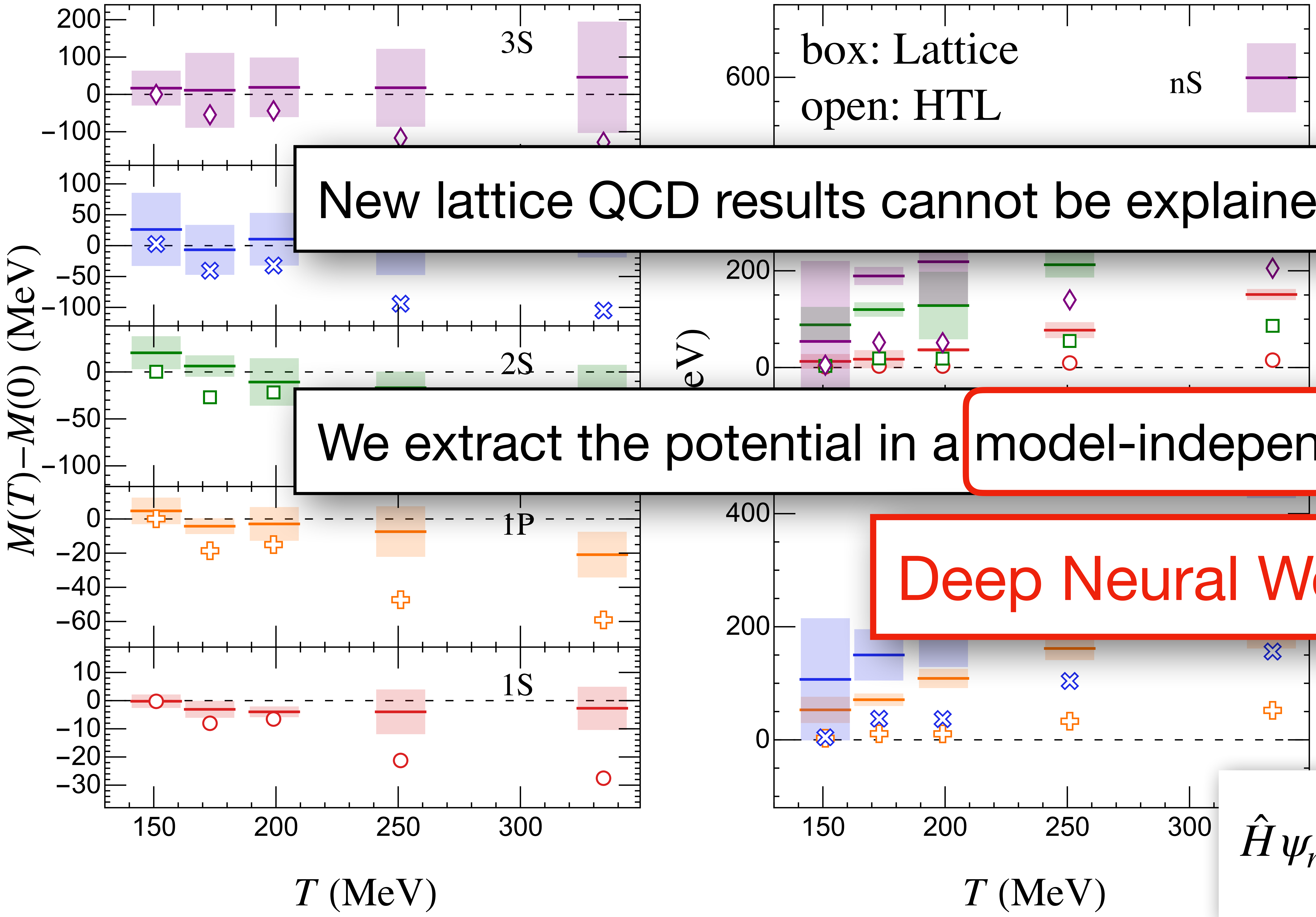


$m_D(T)$ fitted by LQCD

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

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

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)

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

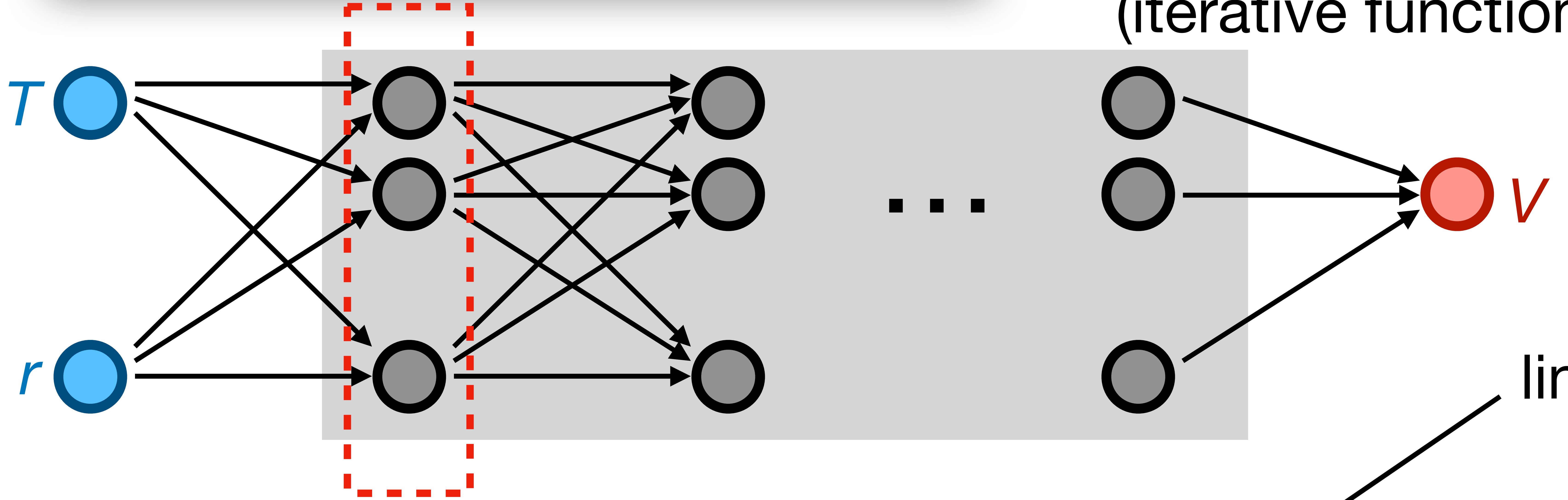
$$\hat{H}\psi_n = -\frac{\nabla^2}{2m_\mu}\psi_n + V(r)\psi_n = E_n\psi_n$$

What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

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

(iterative function substitution)



linear

nonlinear

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

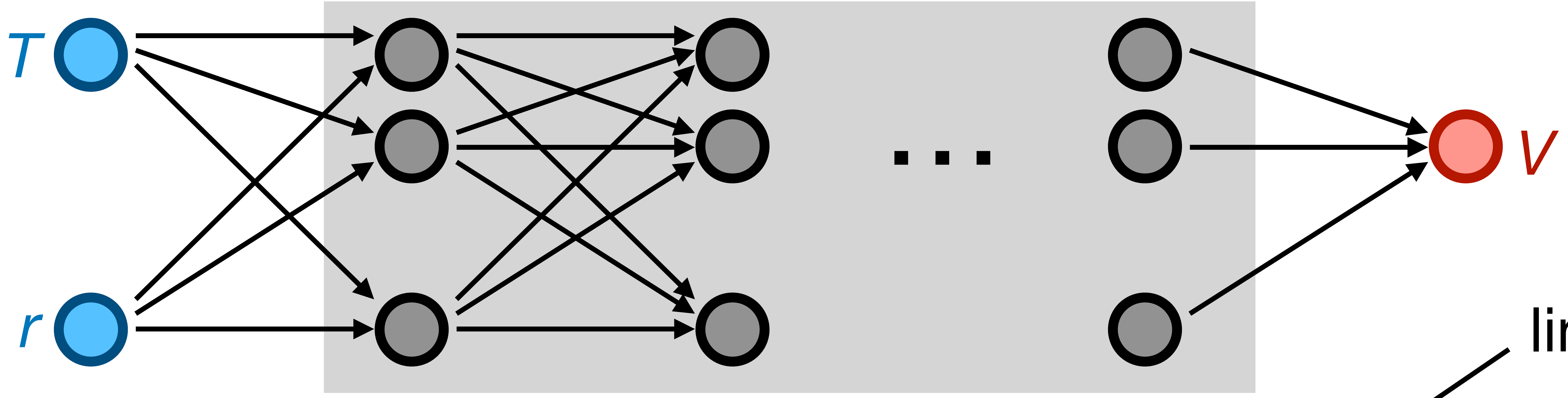
- At the first layer: $z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)}\,T + W_{i,2}^{(1)}\,r,\qquad a_i^{(1)} = \sigma(z_i^{(1)})$

What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

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

(iterative function substitution)



linear

nonlinear

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

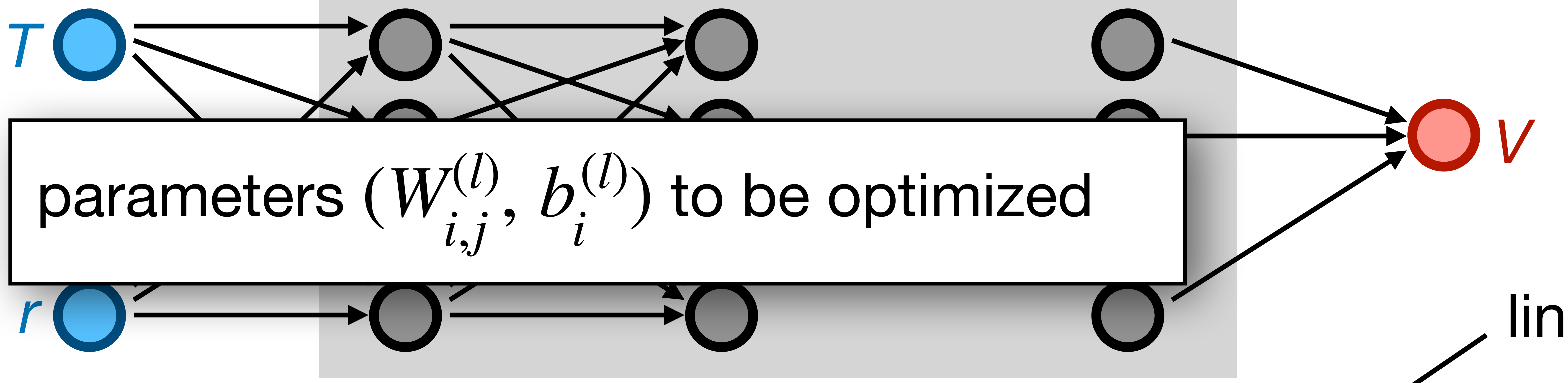
- At the first layer: $z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)}\, T + W_{i,2}^{(1)}\, r, \qquad a_i^{(1)} = \sigma(z_i^{(1)})$
- At later layers: $z_i^{(l)} = b_i^{(l)} + \sum_j W_{i,j}^{(l)}\, a_j^{(l-1)}, \qquad a_i^{(l)} = \sigma(z_i^{(l)})$

What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

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

(iterative function substitution)



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

linear

nonlinear

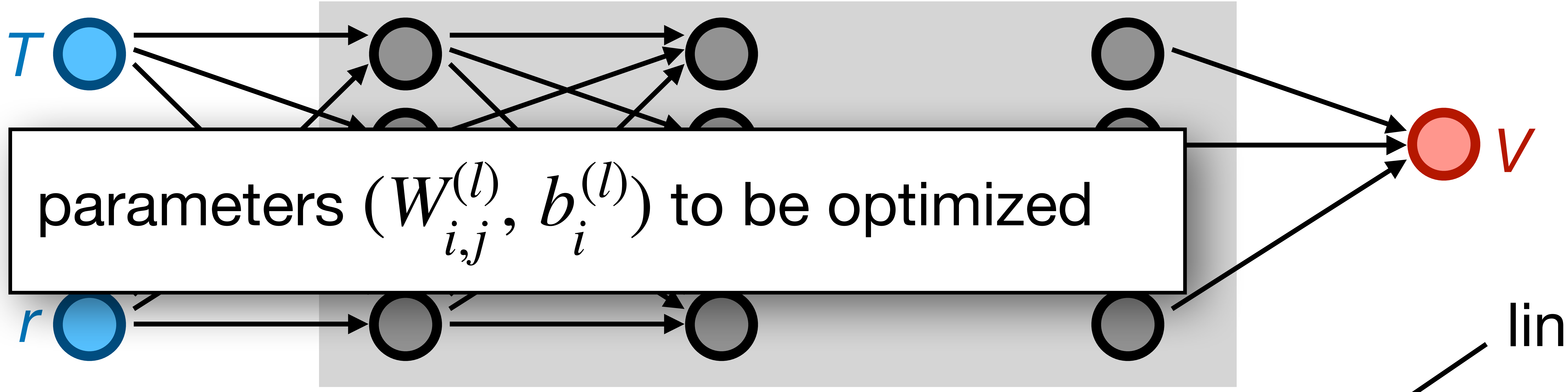
- At the first layer: $z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)}\,T + W_{i,2}^{(1)}\,r,\qquad a_i^{(1)} = \sigma(z_i^{(1)})$
- At later layers: $z_i^{(l)} = b_i^{(l)} + \sum_j W_{i,j}^{(l)}\,a_j^{(l-1)},\qquad a_i^{(l)} = \sigma(z_i^{(l)})$

What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

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

(iterative function substitution)



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

linear

nonlinear

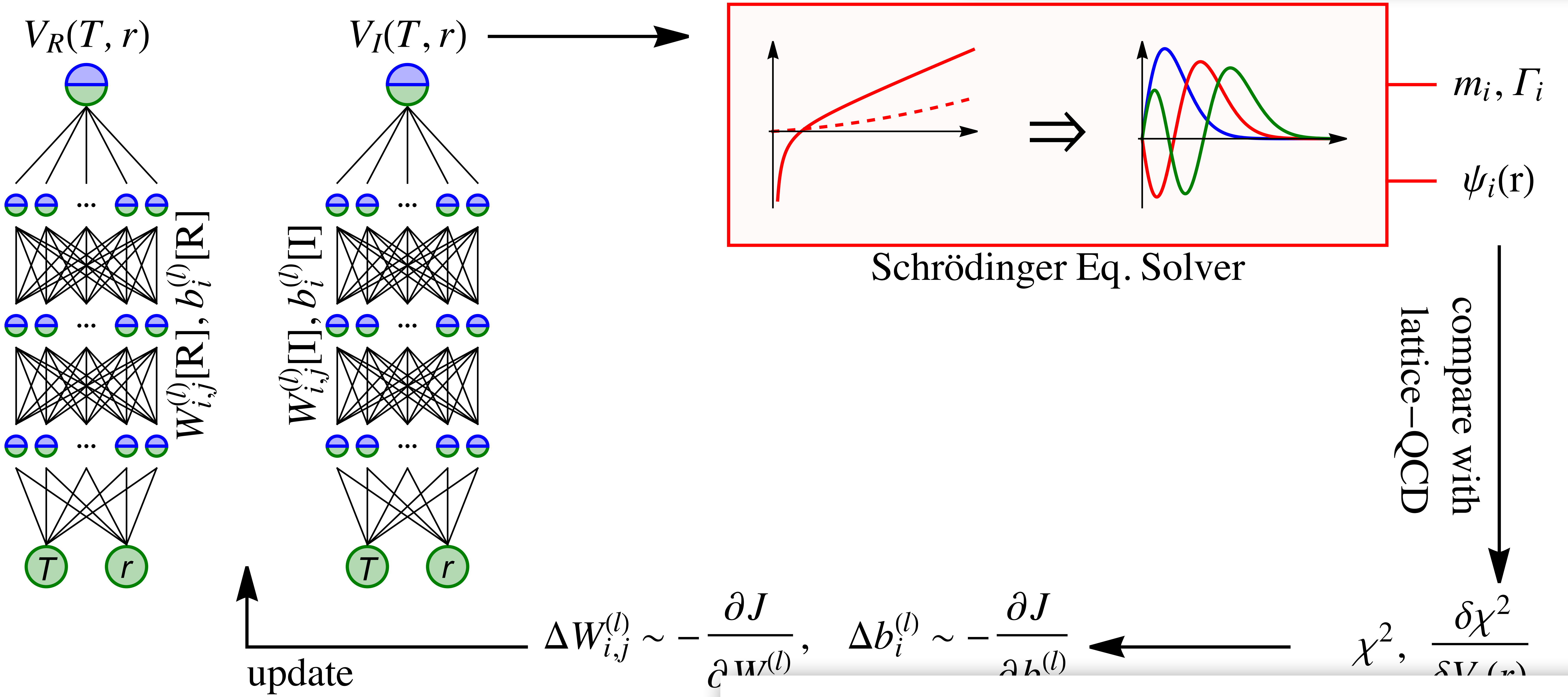
piecewise (linear) description, see Back-up

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

How to learn potential using DNN?

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

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



$$\chi^2 = \sum_{T,i}\left(\frac{m_{T,i} - m_{T,i}^{\text{lattice}}}{\delta m_{T,i}^{\text{lattice}}}\right)^2 + \left(\frac{\Gamma_{T,i} - \Gamma_{T,i}^{\text{lattice}}}{\delta \Gamma_{T,i}^{\text{lattice}}}\right)^2,$$

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

- Start with a known potential

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

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

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

- Learn the potential using DNN

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

- Start with a known potential (solid line)

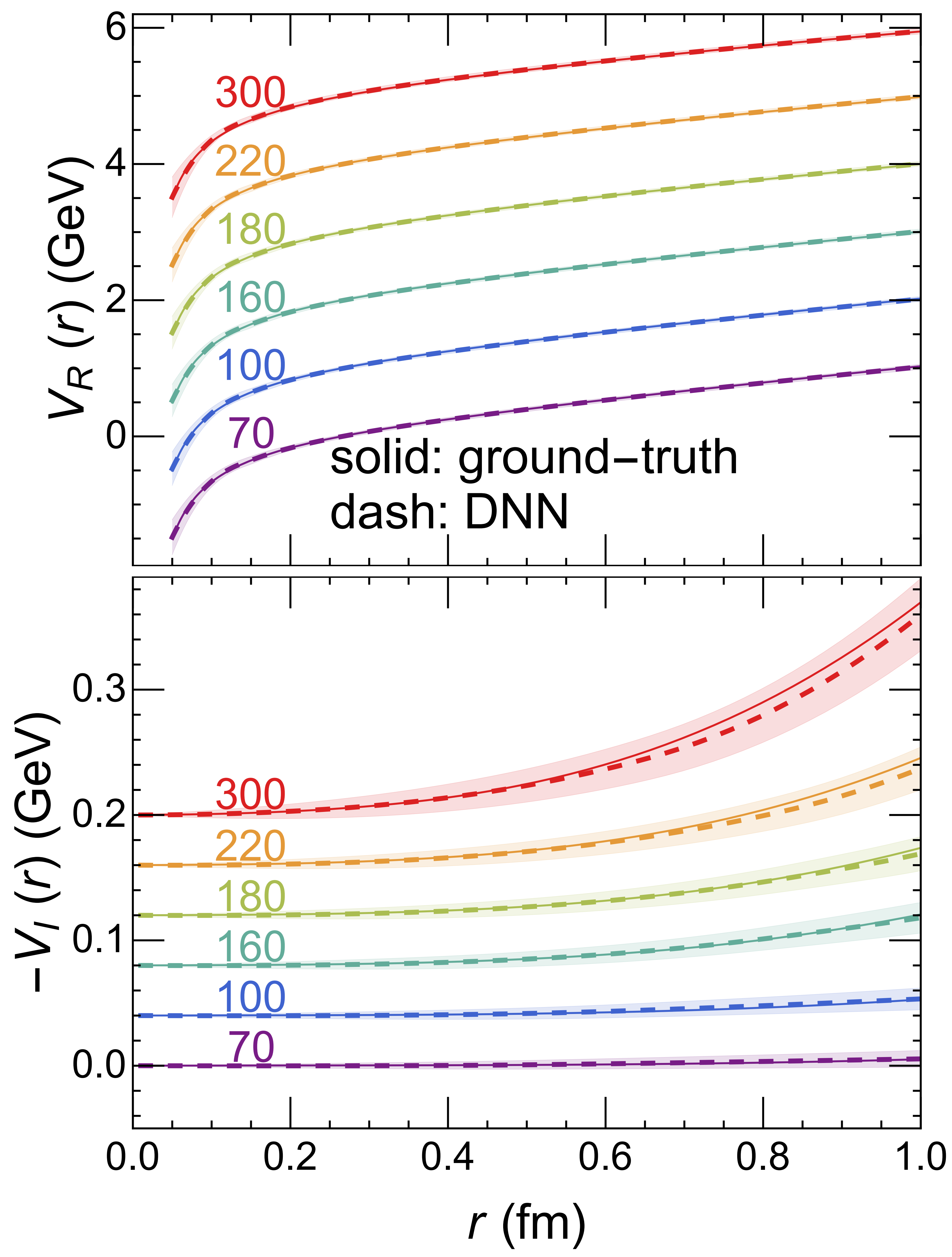
$$V_R(T,r) = \frac{\sigma}{\mu_D}\left(2-(2+\mu_D r)e^{-\mu_D r}\right) - \alpha\left(\mu_D + \frac{e^{-\mu_D r}}{r}\right) + B\,,$$

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

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

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

- Learn the potential using DNN (dash + band)



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

Gray region: extrapolations

- Start with a known potential (solid line)

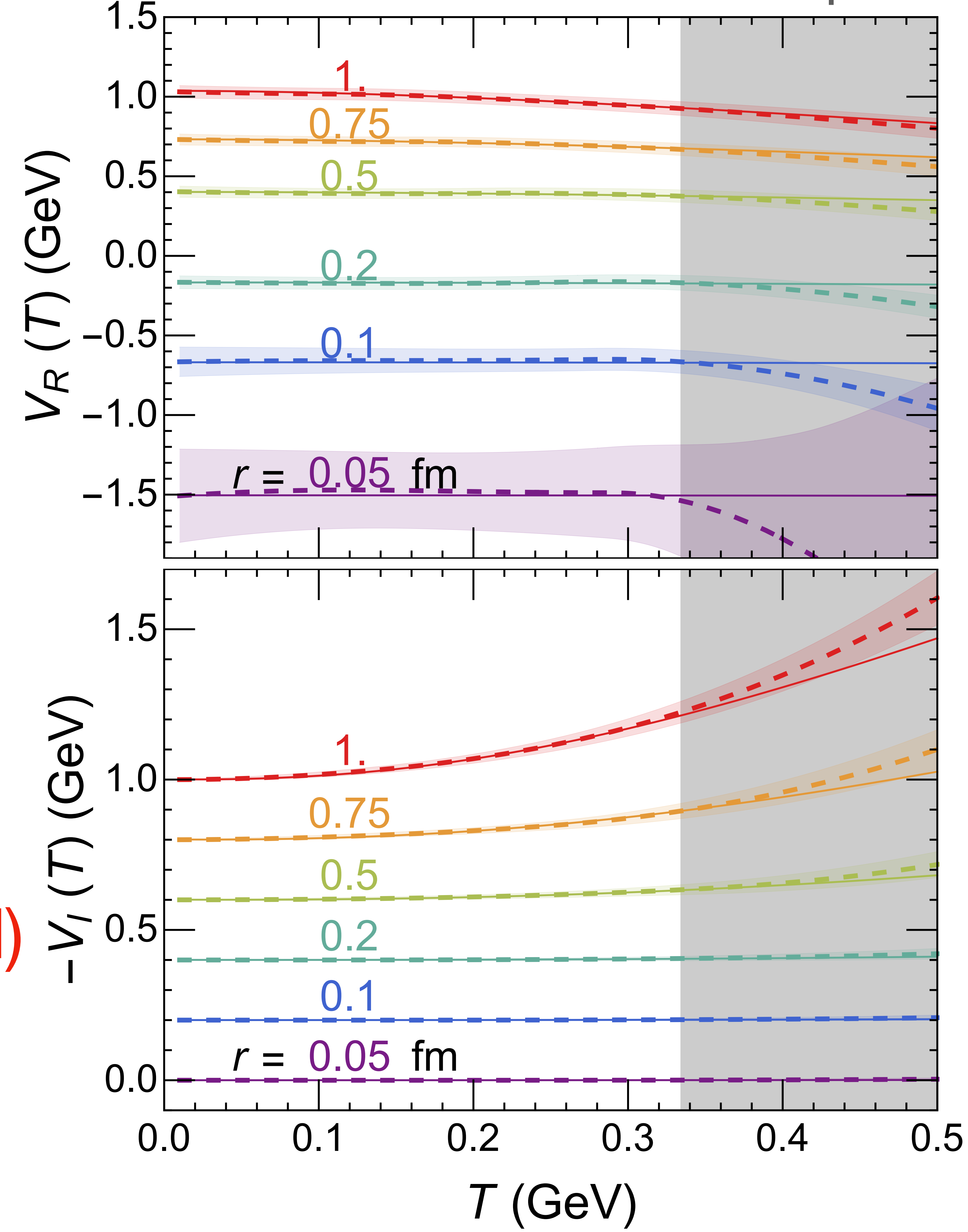
$$V_R(T,r) = \frac{\sigma}{\mu_D}\left(2-(2+\mu_D r)e^{-\mu_D r}\right) - \alpha\left(\mu_D + \frac{e^{-\mu_D r}}{r}\right) + B\,,$$

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

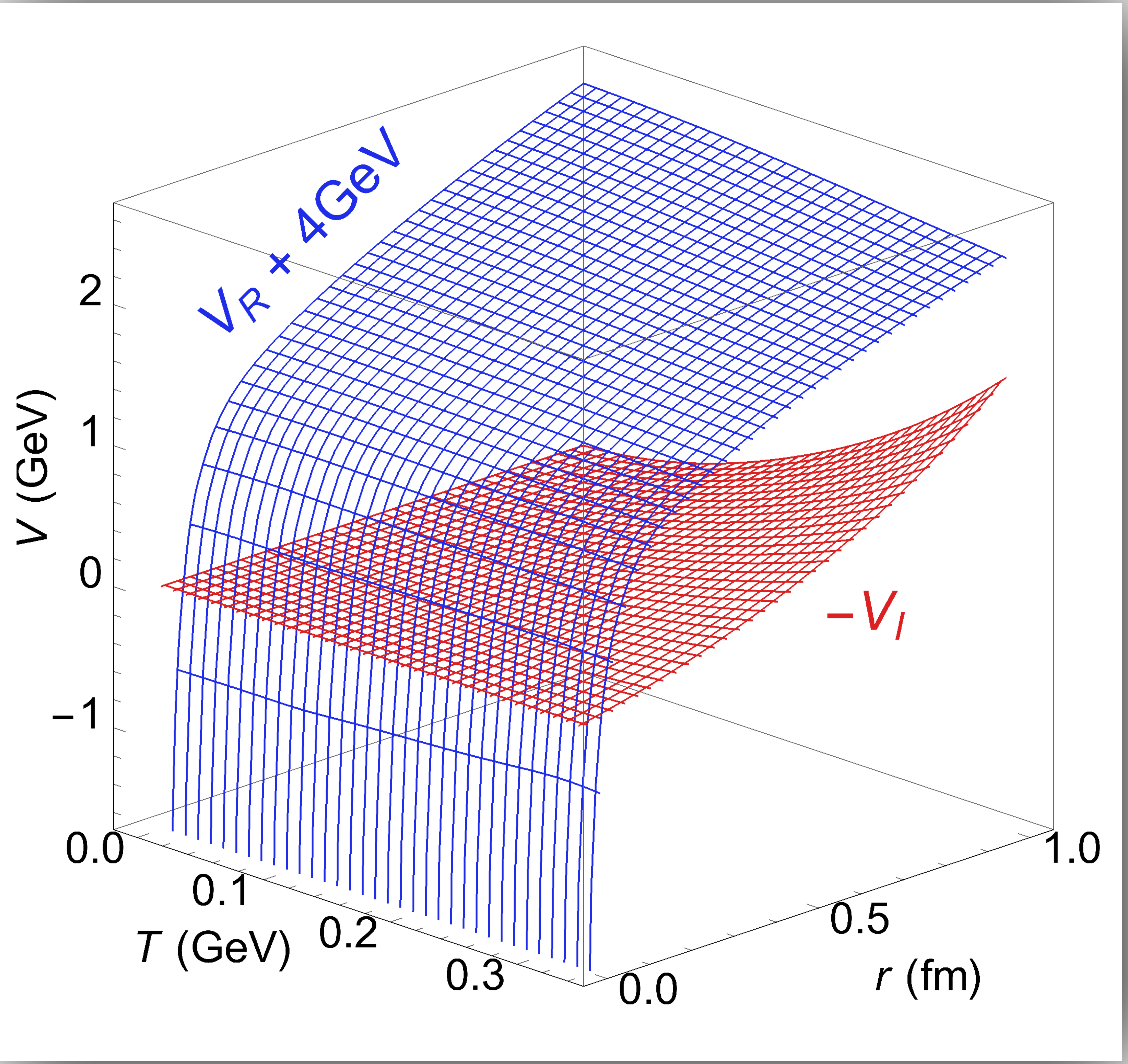
- Compute $\{m_n\,,\,\Gamma_n\}$ at

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

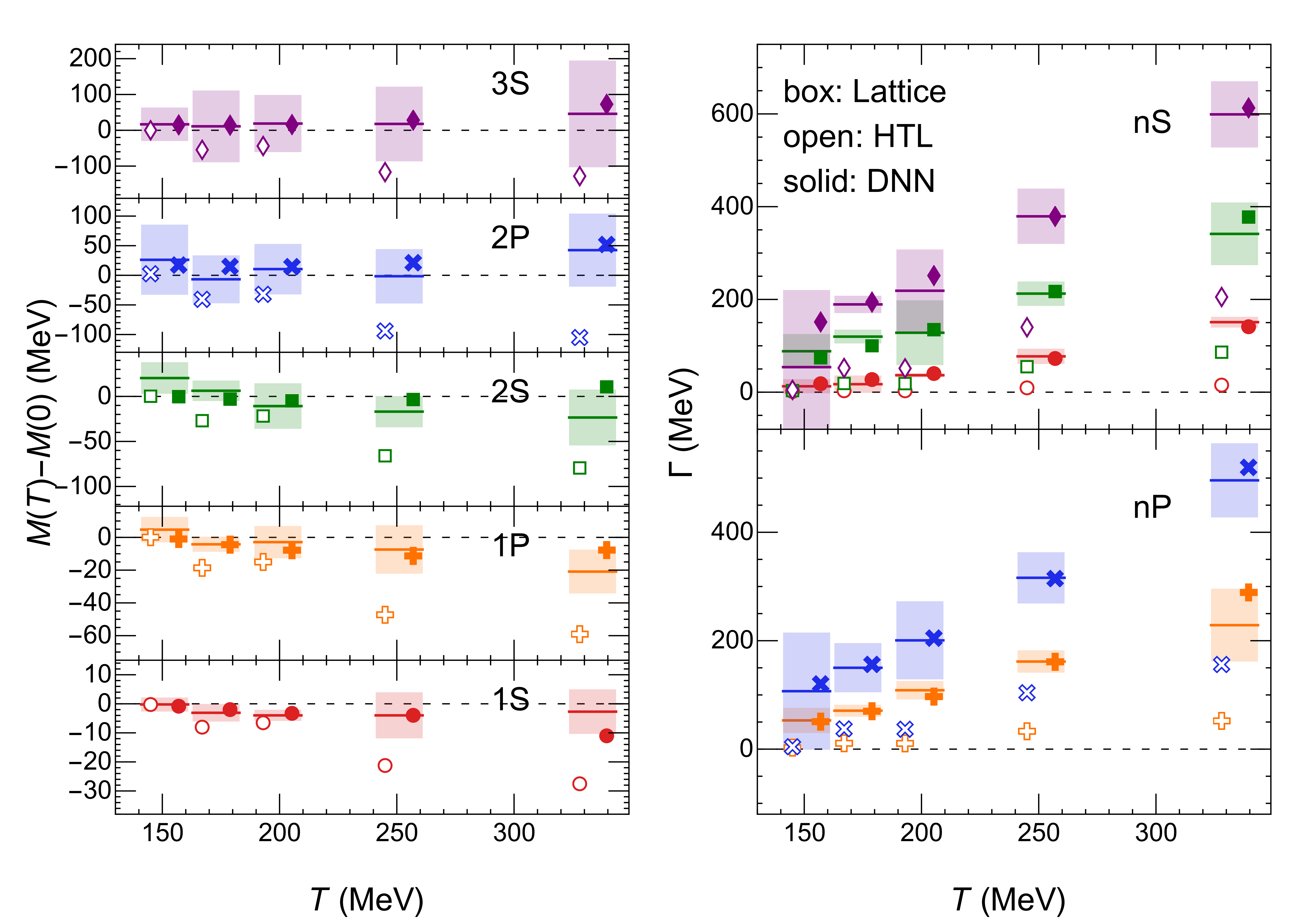
- Learn the potential using DNN (dash + band)



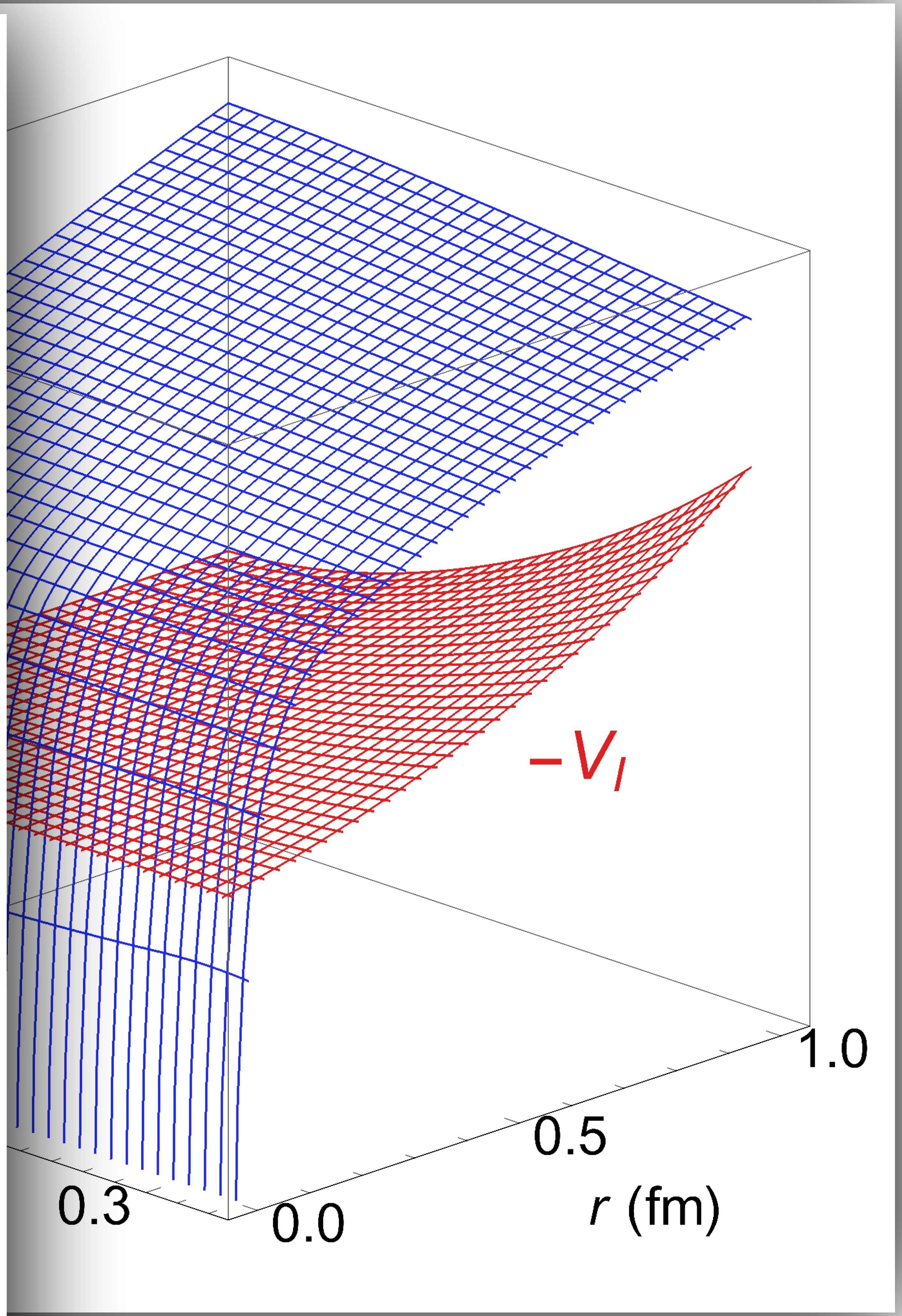
Results



Results

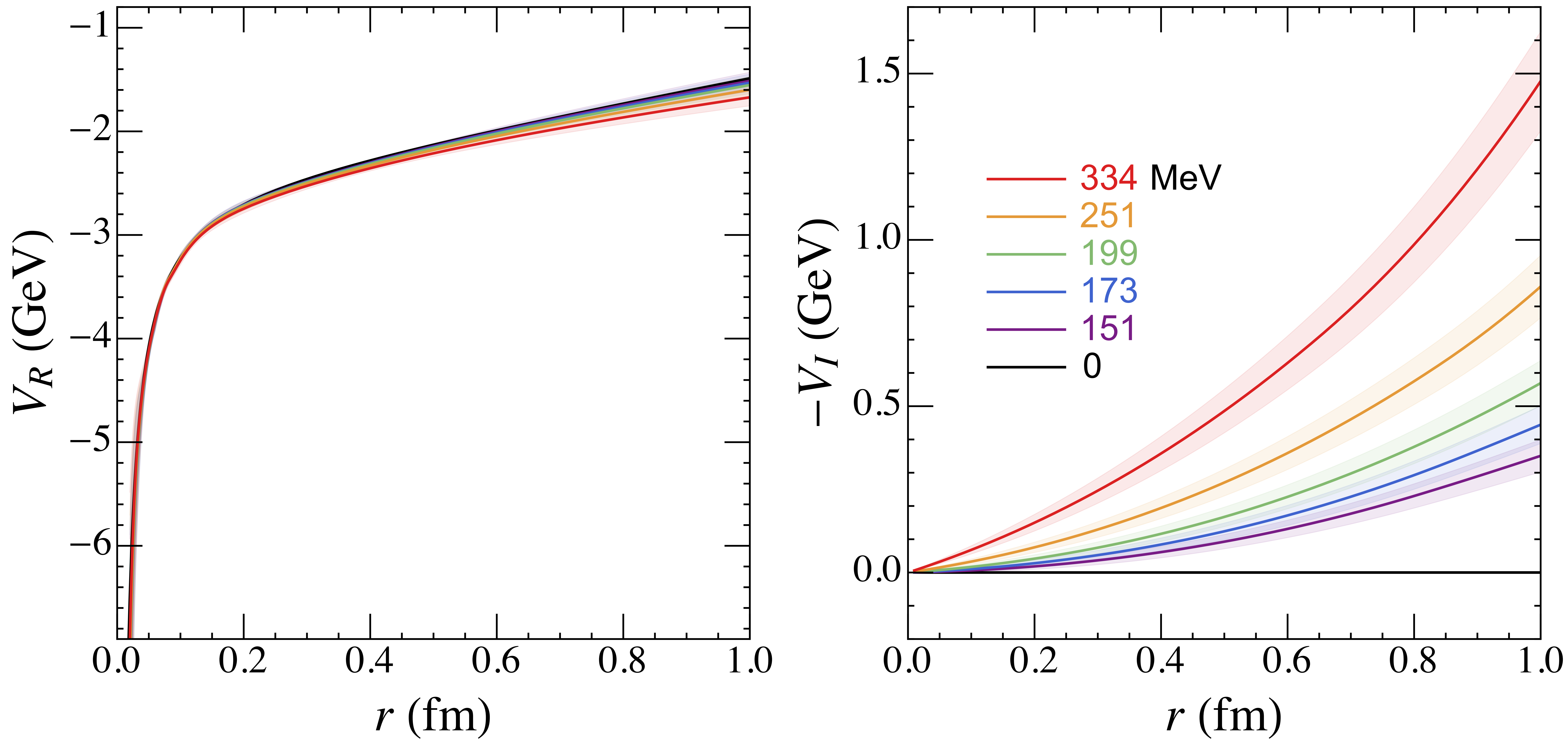


3S
2P
2S
1P
1S
M(T)−M(0) (MeV)
T (MeV)
box: Lattice
open: HTL
solid: DNN
nS
nP
Γ (MeV)
T (MeV)



−V_I
r (fm)

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



[quantitatively consistent with LQCD static potential: see Parkar, 07/30]

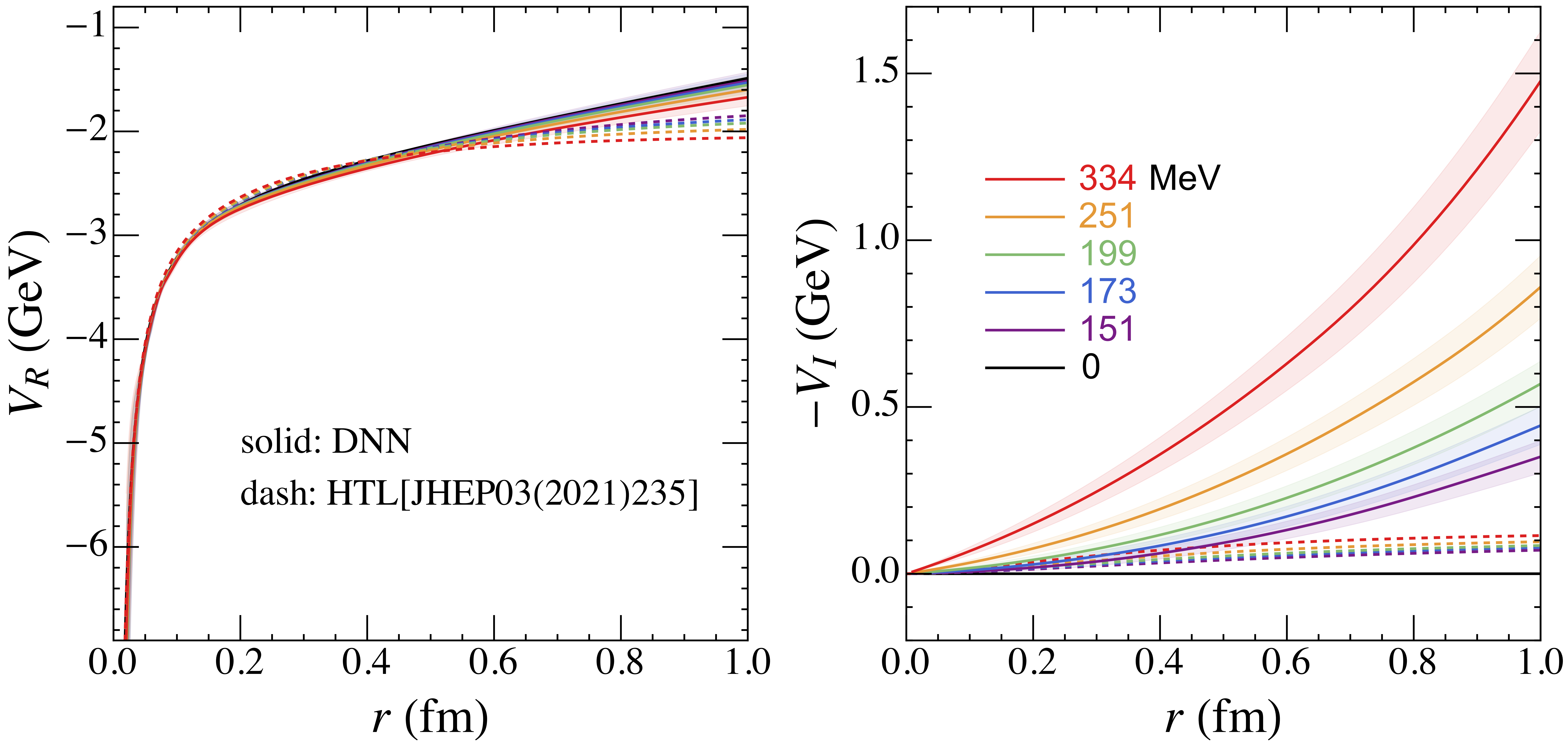
Summary and Outlook

- Develop new algorithm employing DNN to learn $V(r)$ from $\{E_n\}$.

- Extract HF complex $V(T, r)$ from LQCD results of bottomonium m and Γ.

- Phenomenological consequences in heavy-ion collisions?

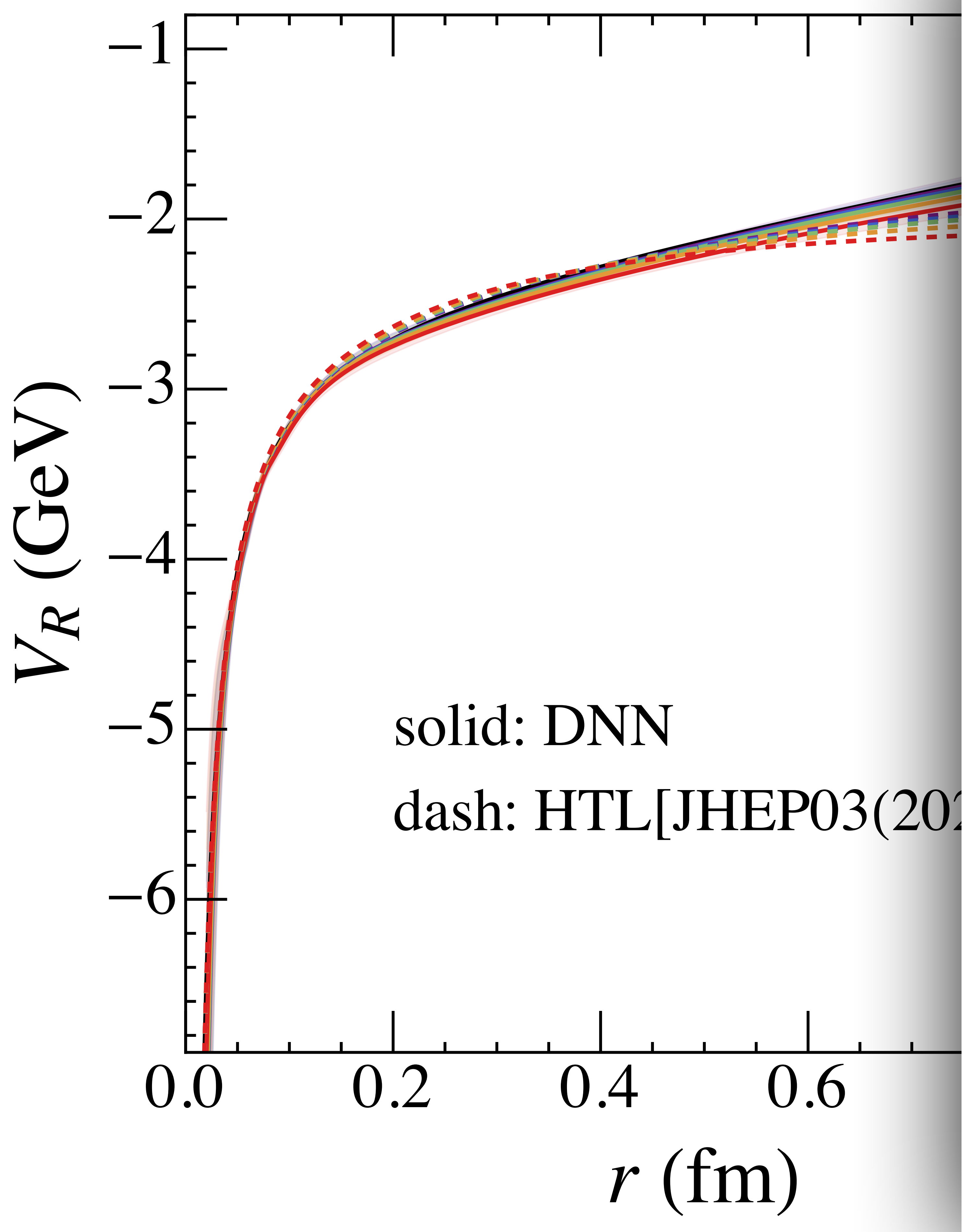
What physics we have learned from $V_{\mathrm{DNN}}(T, r)$?
--- compare with HTL potential used in [1]

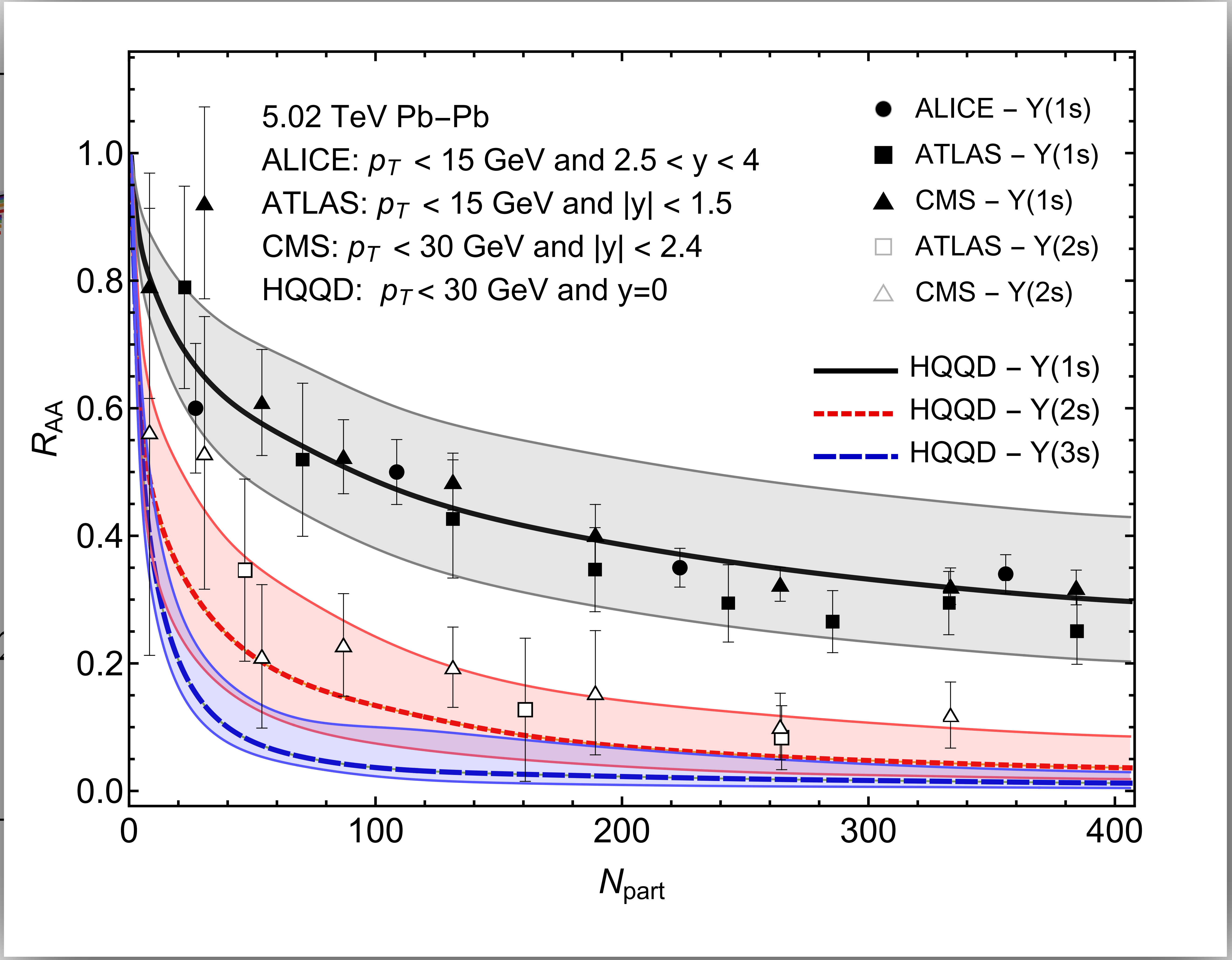


[1] A. Islam and M. Strickland, JHEP03(2021)235

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

--- compare with HTL potential used in [1]

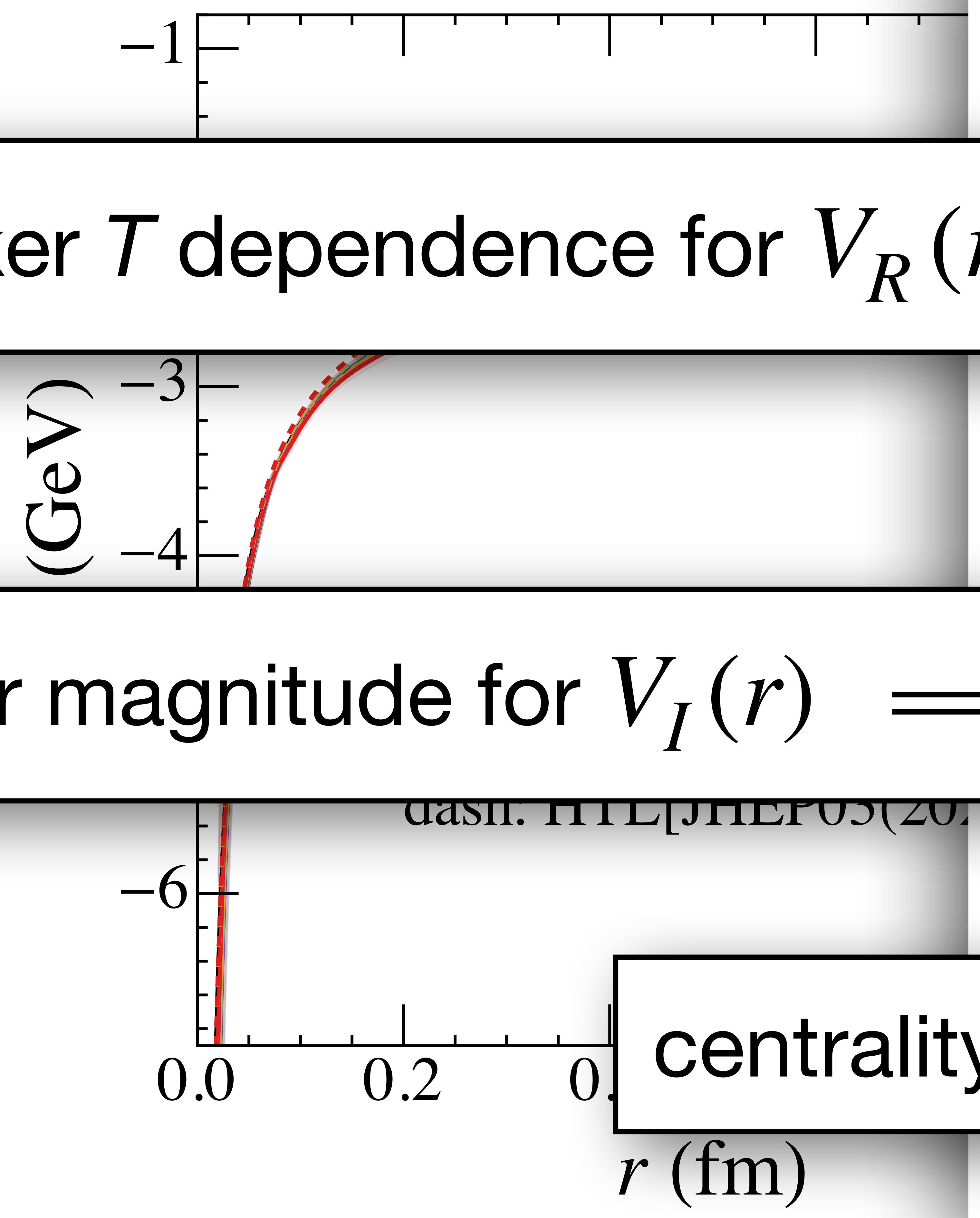


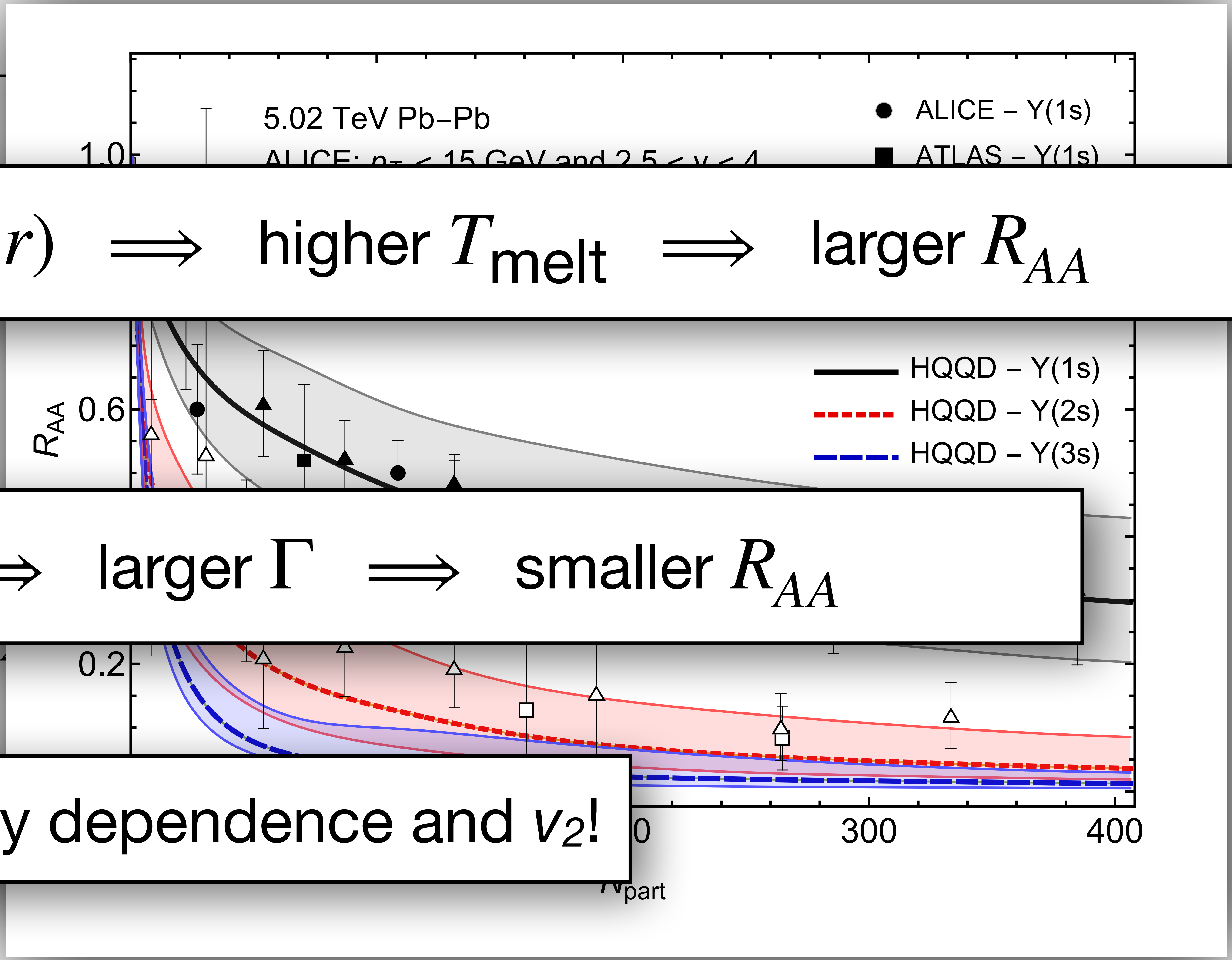


[1] A. Islam and M. Strickland, JHEP03(2021)235

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

--- compare with HTL potential used in [1]





weaker T dependence for $V_R(r)$ $\Longrightarrow$ higher $T_{\rm melt}$ $\Longrightarrow$ larger R_{AA}

larger magnitude for $V_I(r)$ $\Longrightarrow$ larger Γ $\Longrightarrow$ smaller R_{AA}

centrality dependence and v_2!

[1] A. Islam and M. Strickland, JHEP03(2021)235

Back-up Slides

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

continuous

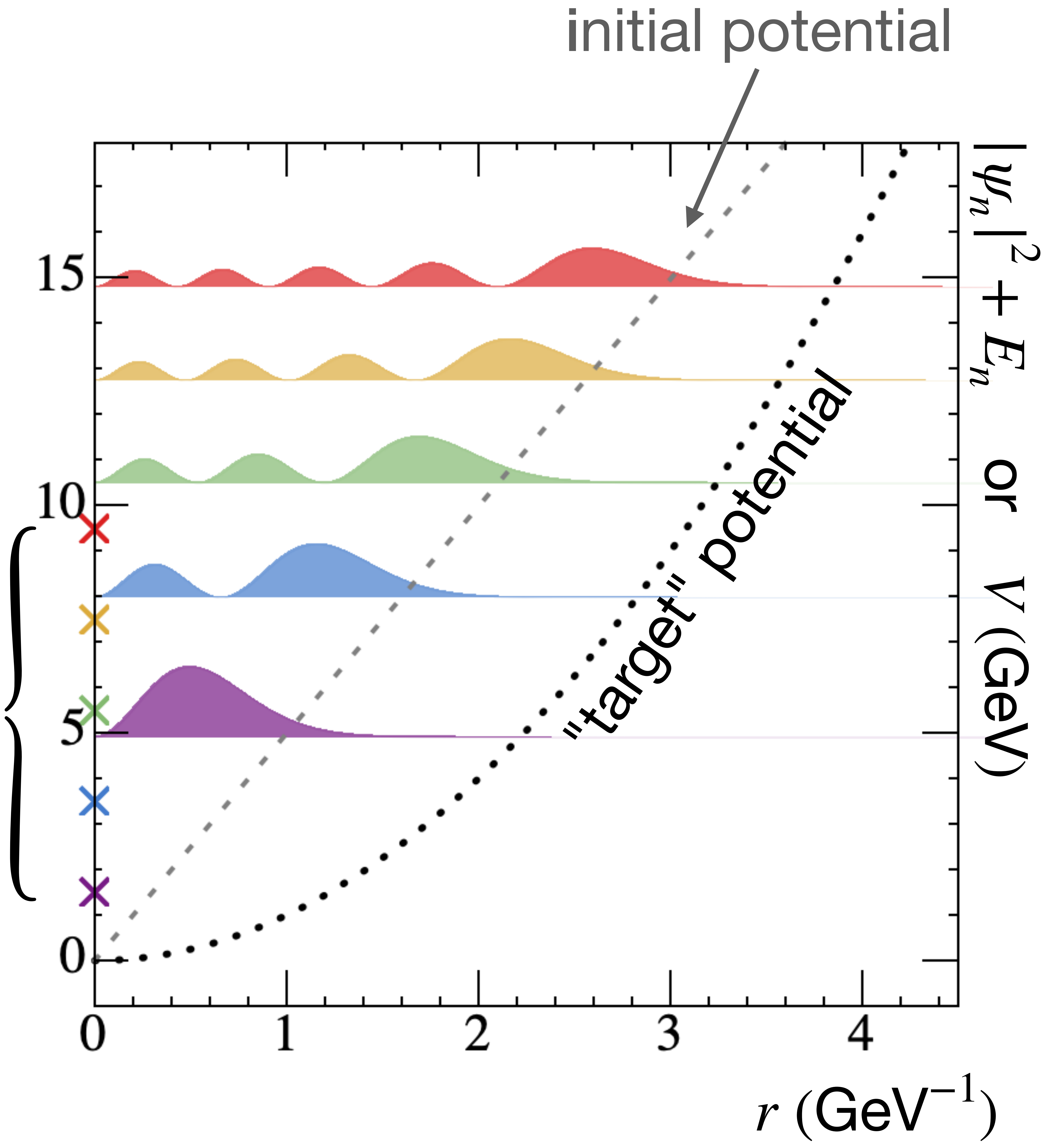
discrete, finite

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

learn $V(r)$ according to

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

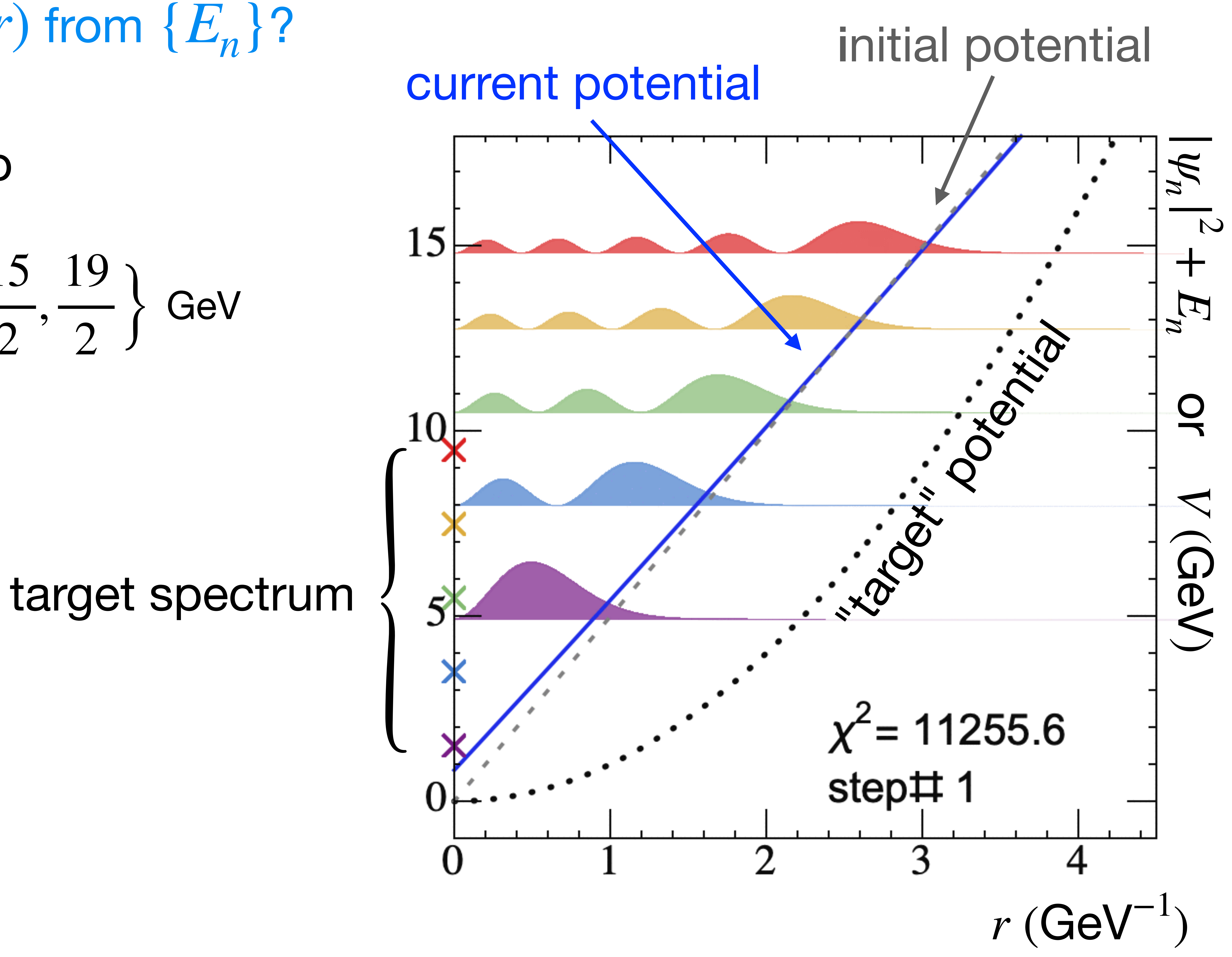
target spectrum



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

learn $V(r)$ according to

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



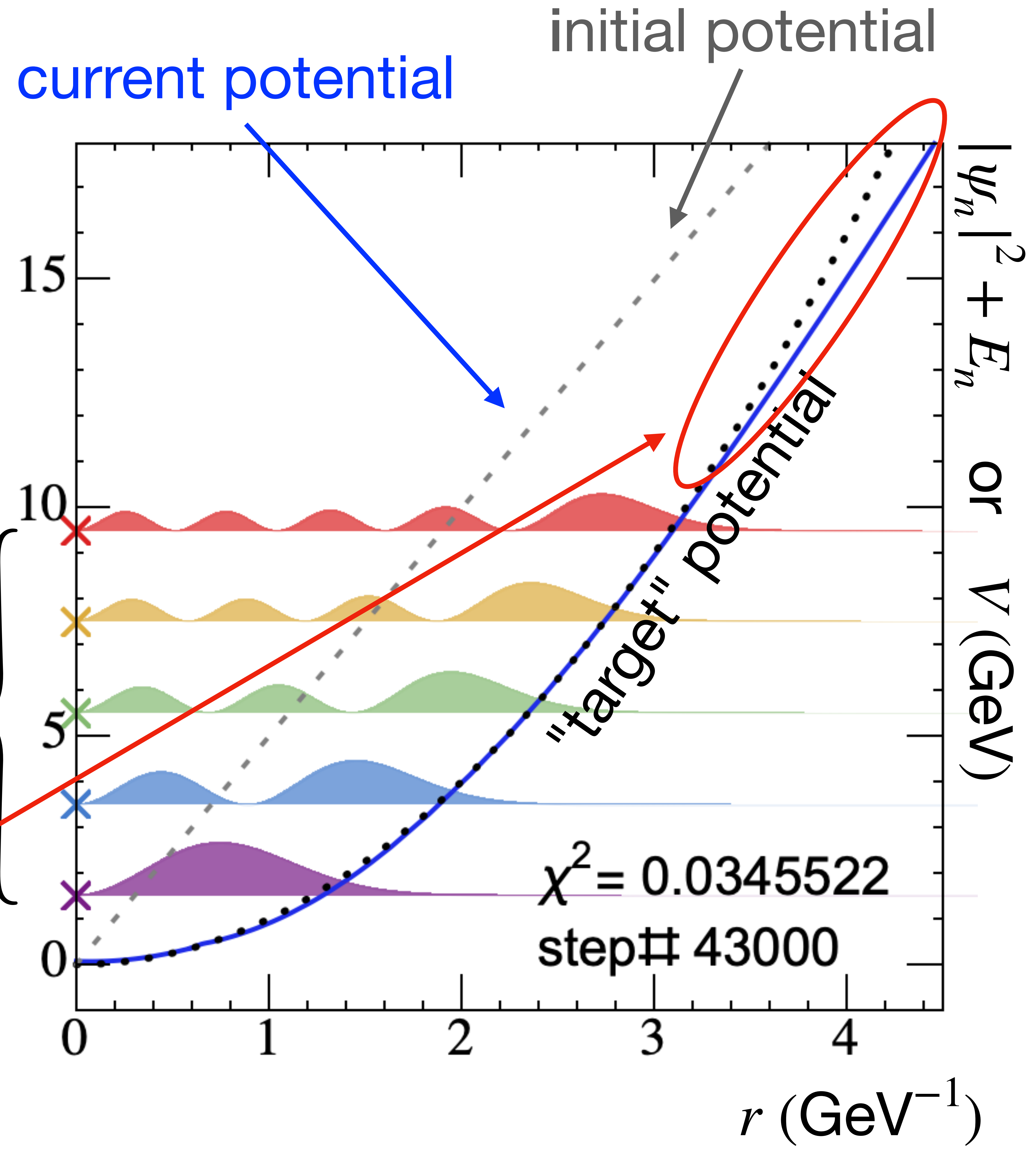
Test - Can we learn $V(r)$ from $\{E_n\}$?

-- Yes! (for a certain r range)

learn $V(r)$ according to

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

target spectrum



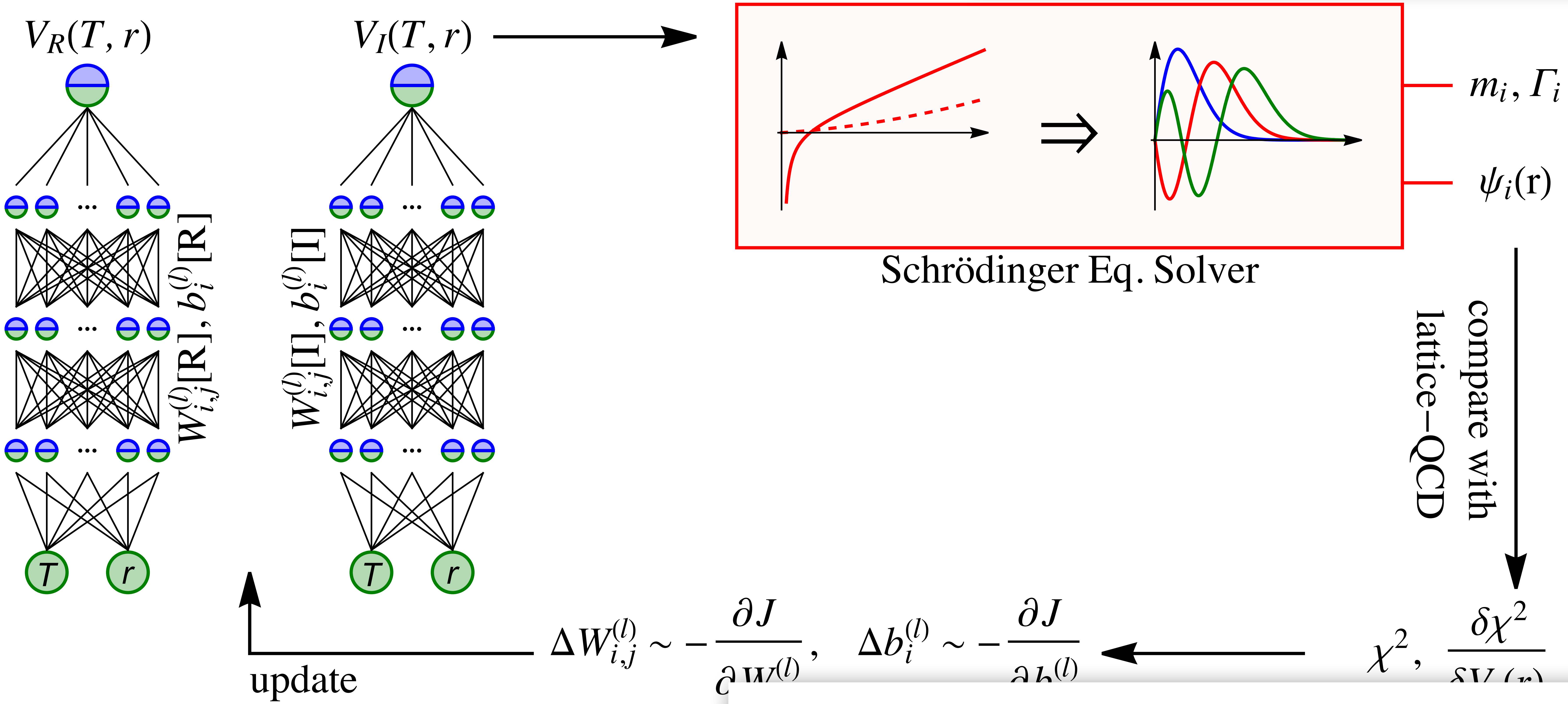
Deviate from the exact potential where all $\psi_n \to 0$,

$$\delta E_n = \langle \psi_n | \delta V(r) | \psi_n \rangle$$

How to learn potential using DNN?

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

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



$$\chi^2 = \sum_{T,i}\left(\frac{m_{T,i} - m_{T,i}^{\text{lattice}}}{\delta m_{T,i}^{\text{lattice}}}\right)^2 + \left(\frac{\Gamma_{T,i} - \Gamma_{T,i}^{\text{lattice}}}{\delta\Gamma_{T,i}^{\text{lattice}}}\right)^2,$$

How to learn potential using DNN?

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

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

In a ***typical*** Deep Learning problem, true function V(T,r) is ***known***,

$J = ||V_{\text{DNN}}(T,r) - V_{\text{true}}(T,r)||^2 +$ regularizer (distance)

In ***this work***, true value for V(T,r) is ***unknown***,

$J = \chi^2 +$ regularizer (***implicit*** function of W and b)

$\partial_W J,\ \partial_b J$ can be computed exactly, thanks to the perturbation theory.

update

$$\delta m_i = \langle \psi_i | \delta V_R(r) | \psi_i \rangle,$$
$$\delta\Gamma_i = -\langle \psi_i | \delta V_I(r) | \psi_i \rangle\,.$$

$$\chi^2 = \sum_{T,i}\left(\frac{m_{T,i} - m^{\text{lattice}}_{T,i}}{\delta m^{\text{lattice}}_{T,i}}\right)^2 + \left(\frac{\Gamma_{T,i} - \Gamma^{\text{lattice}}_{T,i}}{\delta\Gamma^{\text{lattice}}_{T,i}}\right)^2,$$

Uncertainty Quantification according to Bayesian inference

- Posterior distribution for DNN parameters $(\boldsymbol{\theta})$

$$\text{Posterior}(\boldsymbol{\theta}\,|\,\text{data}) \propto L(\boldsymbol{\theta}\,|\,\text{data}) \cdot \text{Prior}(\boldsymbol{\theta}) = N_0 \exp\left[-\frac{\chi^2(\boldsymbol{\theta})}{2} - \frac{\lambda}{2}\boldsymbol{\theta}\cdot\boldsymbol{\theta}\right].$$

$\chi^2(\boldsymbol{\theta})$ is an implicit function of $V_{\text{DNN}}(\boldsymbol{\theta}; r)$

- Task #1: find the most optimal parameter set by maximizing Posterior.

- Task #2: at any distance *r*, compute the likelihood (density) distribution of $V_{\boldsymbol{\theta}}$,

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

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

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

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

$$w_i = P(V_{\boldsymbol{\theta}_i})\mathrm{d}V_i = \text{Posterior}(\boldsymbol{\theta}_i\,|\,\text{data})$$

$\boxed{\text{Computationally Expensive!}}$

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

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

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

$$w_i = P(V_{\boldsymbol{\theta}_i})\mathrm{d}V_i = 1$$

$\boxed{\text{Hard to sample according to Posterior!}}$

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

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

- Method 3)
  - Sample $\{\boldsymbol{\theta}_i\}$ according to a reference distribution: $P(\boldsymbol{\theta}) = \tilde{P}(\boldsymbol{\theta})$;
  - Each data point corresponds to the element volume $\mathrm{d}^N\boldsymbol{\theta}_{\boldsymbol{i}} = 1/\tilde{P}(\boldsymbol{\theta}_{\boldsymbol{i}})$;
  - 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

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

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

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

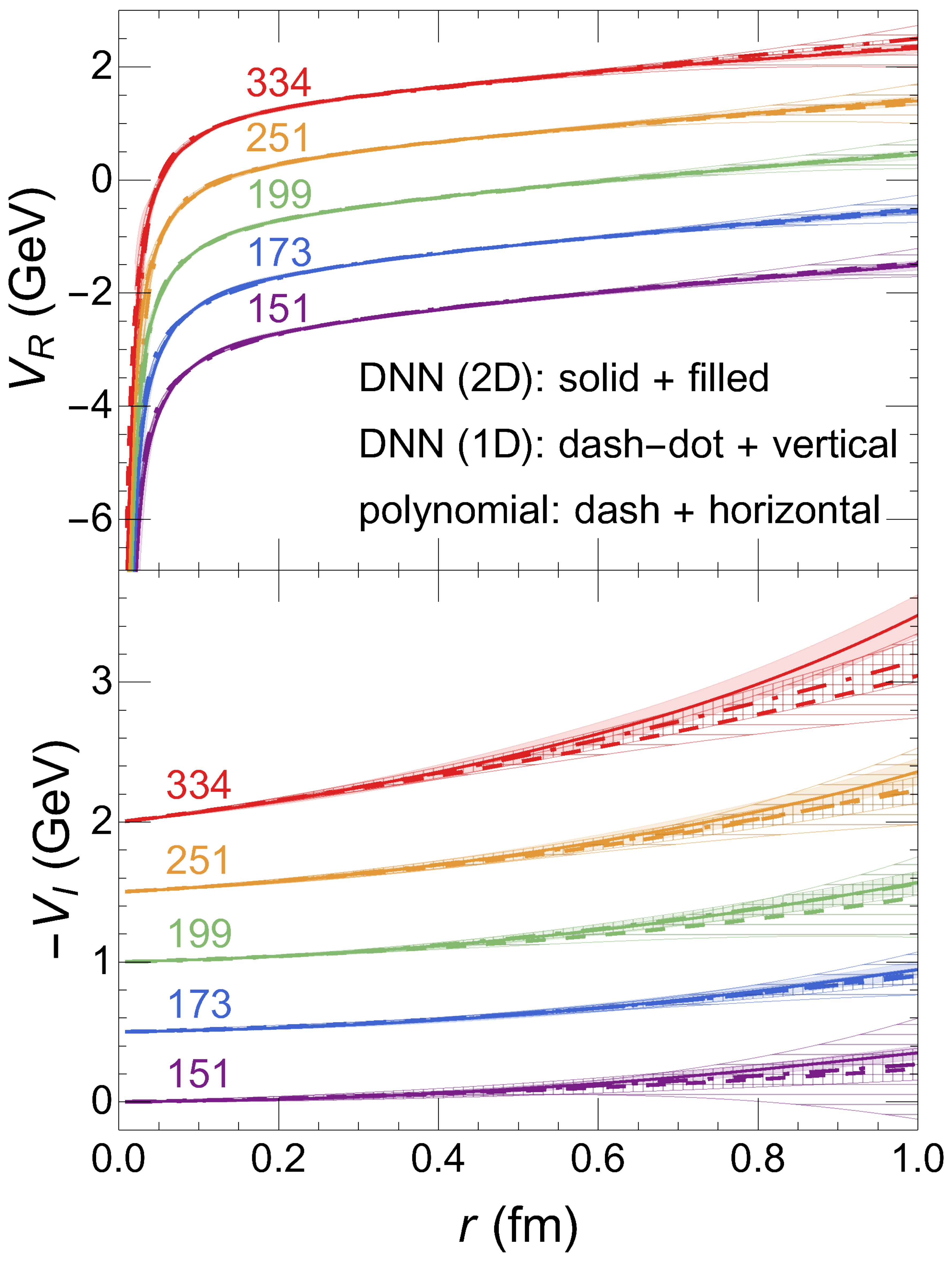
- Method 3)
  - Sample $\{\boldsymbol{\theta}_i\}$ according to a reference distribution: $P(\boldsymbol{\theta}) = \tilde{P}(\boldsymbol{\theta})$;
  - Each data point corresponds to the element volume $\mathrm{d}^N\boldsymbol{\theta_i} = 1/\tilde{P}(\boldsymbol{\theta_i})$;
  - Compute $V_{\boldsymbol{\theta}_i}(r)$, $\chi^2_{\boldsymbol{\theta}_i}$, and Posterior$(\boldsymbol{\theta}_i\,|\,\text{data})$;
  - For given *r*, histogram $V_{\boldsymbol{\theta}_i}(r)$ with weights

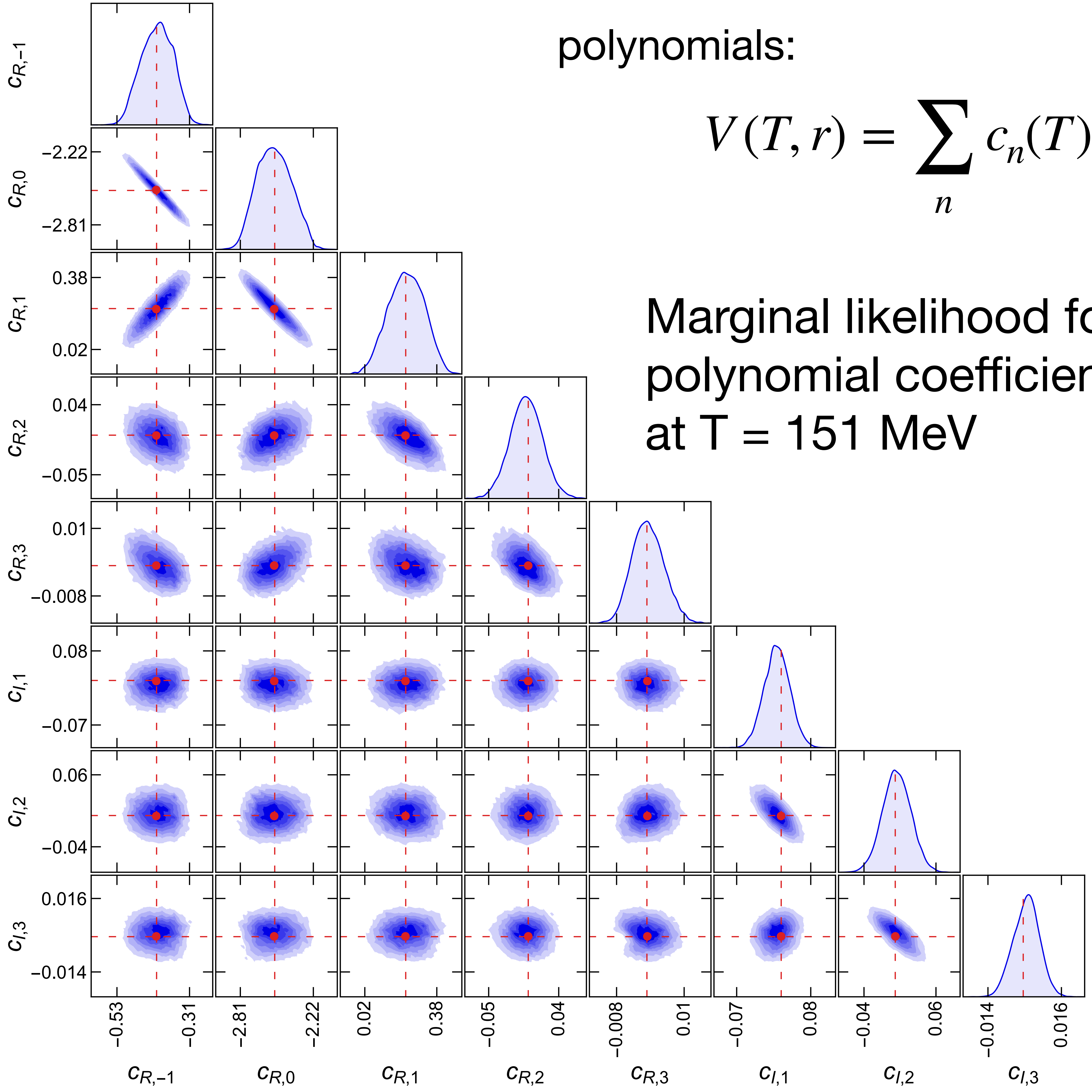
$$w_i = P(V_{\boldsymbol{\theta}_i})\mathrm{d}V_i = \text{Posterior}(\boldsymbol{\theta_i})/\tilde{P}(\boldsymbol{\theta_i})$$

  - In practice:

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

Consistency test with T-indept. DNN and polynomial

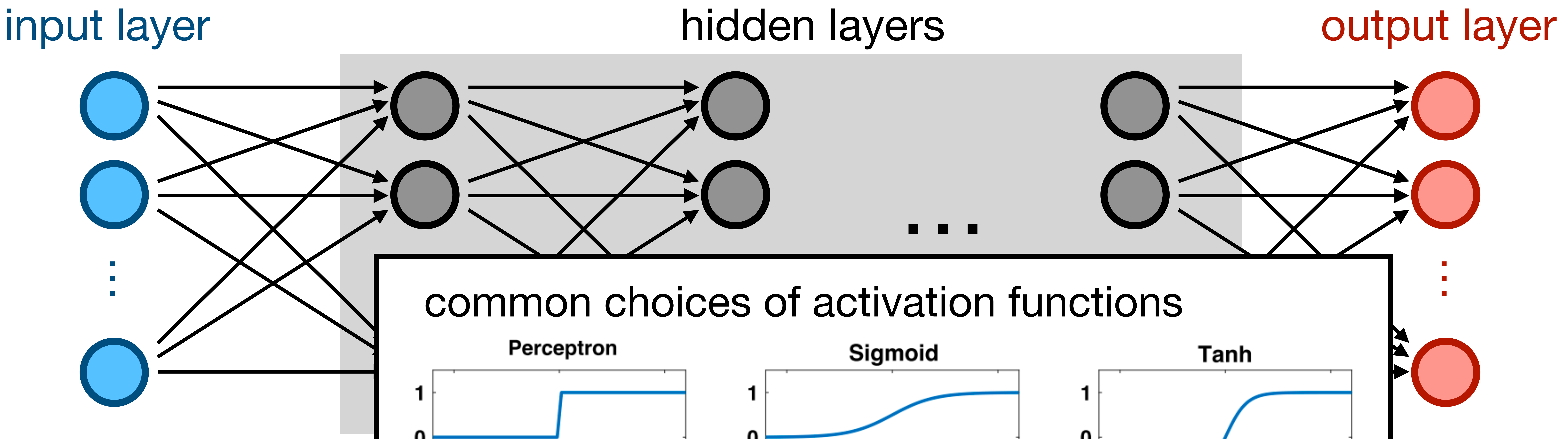


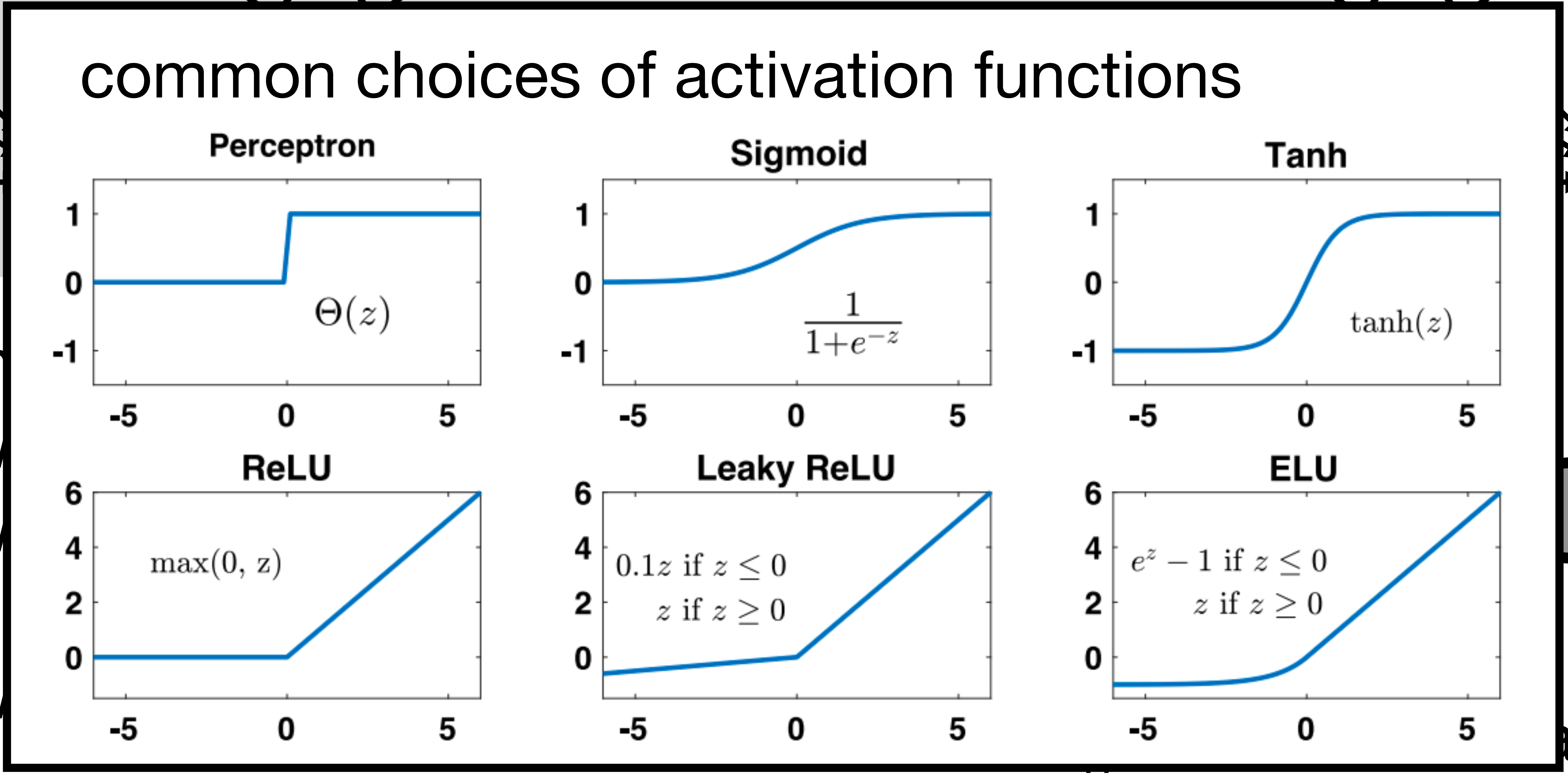


polynomials:

$$V(T,r) = \sum_n c_n(T)\, r^n$$

Marginal likelihood for polynomial coefficients at T = 151 MeV





- Multiple "hidden

$v^{(1)}{}_j = \sigma^{(1)}(\Sigma$ v

$v^{(2)}{}_j = \sigma^{(2)}(\Sigma$ v

...

$y_j = \sigma^{(l+1)}(\Sigma$ v

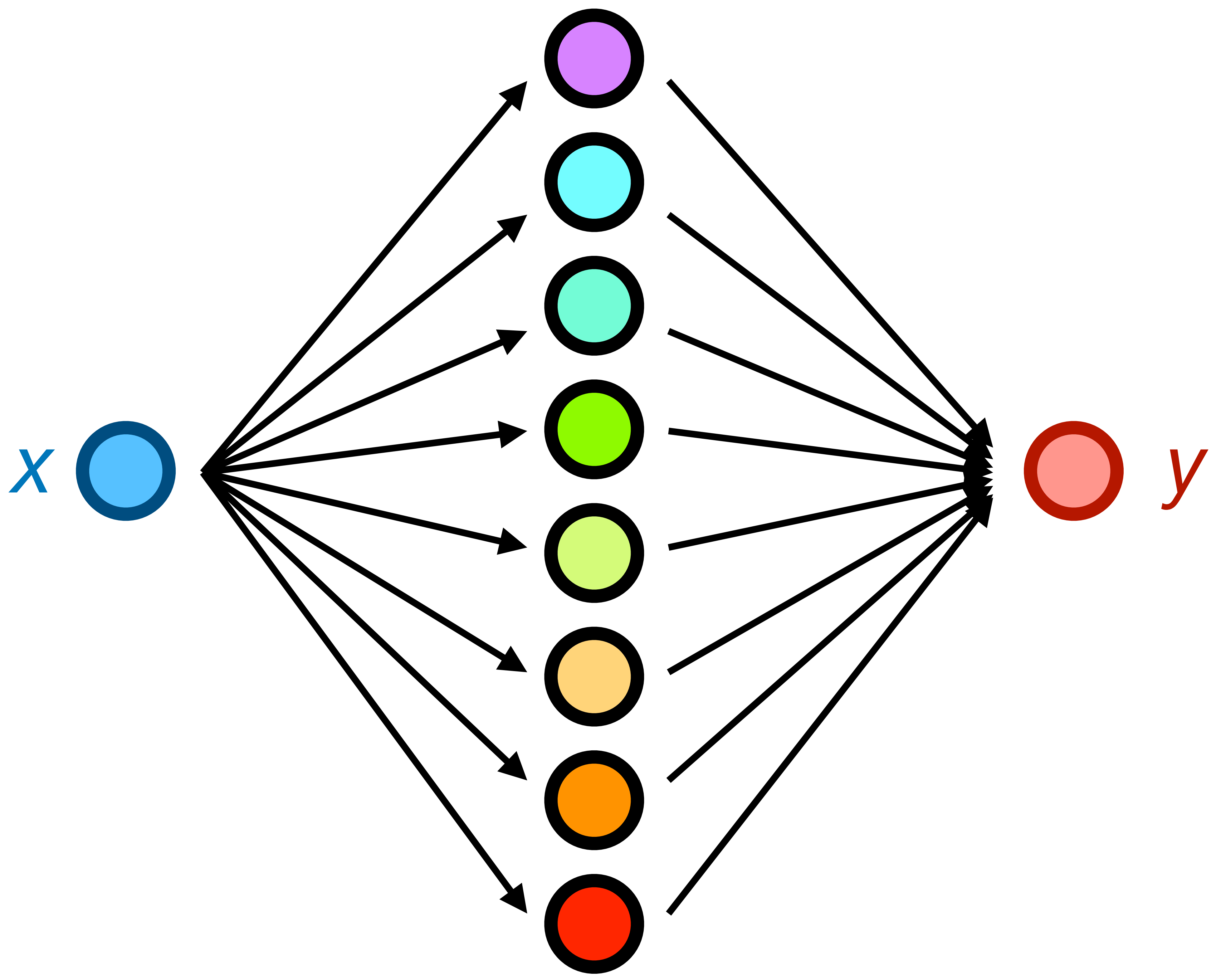
ayers/depth of the network

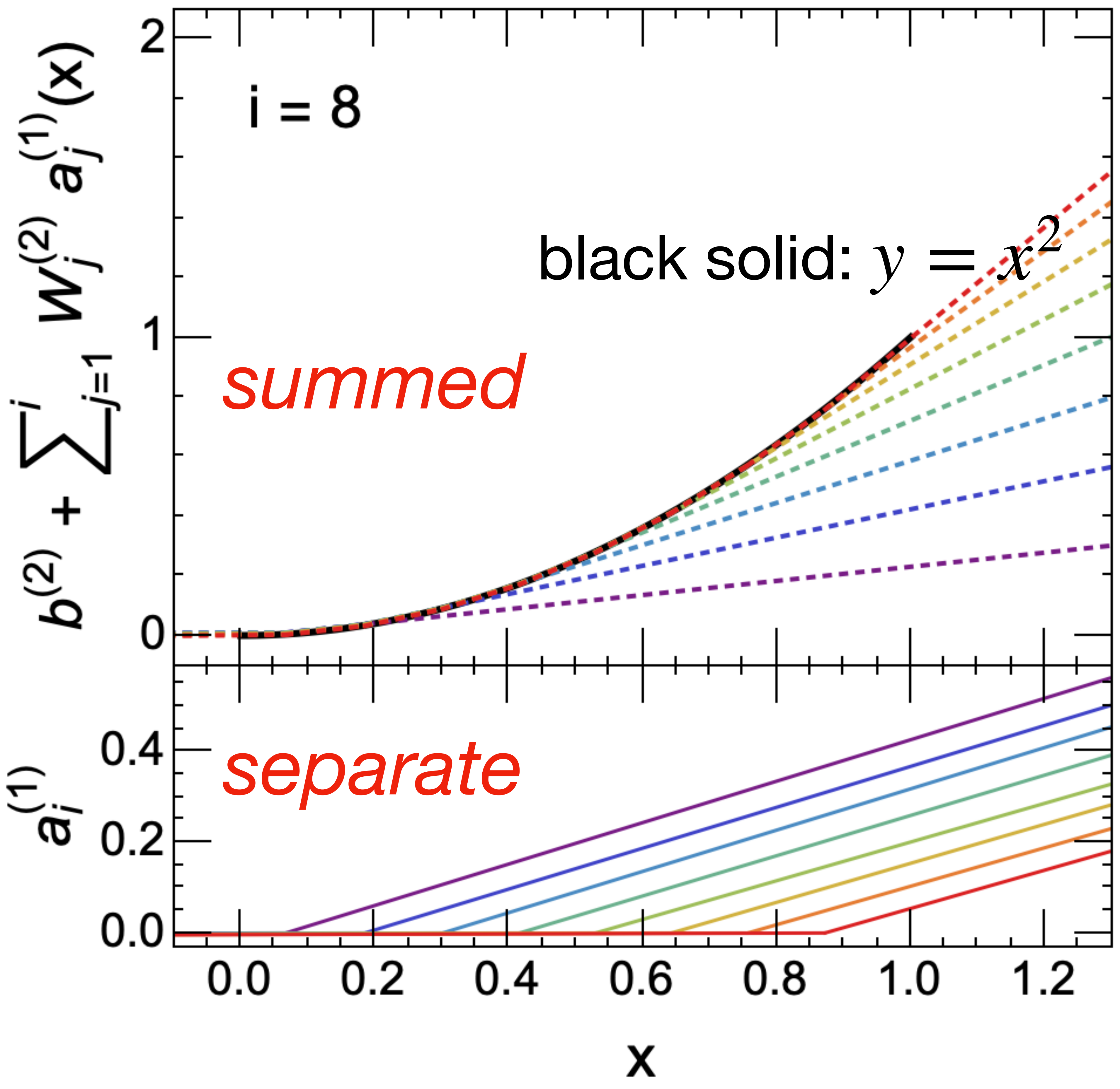
$M^{(l)}$ -- number of neurons at layer (l)

Note2: $\sigma^{(1)}$ ... $\sigma^{(l+1)}$ are activation functions.
They can be the same or different.

How do Deep Neural Networks work?

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





piecewise linear interpolation!!!

[Using activation function $\sigma(z)$ = max(0,z)]