

Fonds de recherche  
sur la nature  
et les technologies

Québec 



McGill  
UNIVERSITY

First MODE Workshop on Differentiable Programming

Date: 2021 - Sept - 7

6-8 September 2021  
CP3, Louvain-la-Neuve  
Europe/Zurich timezone

From (bottomonium) mass spectrum  
to (in-medium heavy-quark) potential  
via deep learning

Passenger: Shuzhe Shi (McGill University)

Travel with:

Kai Zhou, Jiaxing Zhao, Swagato Mukherjee, and Pengfei Zhuang

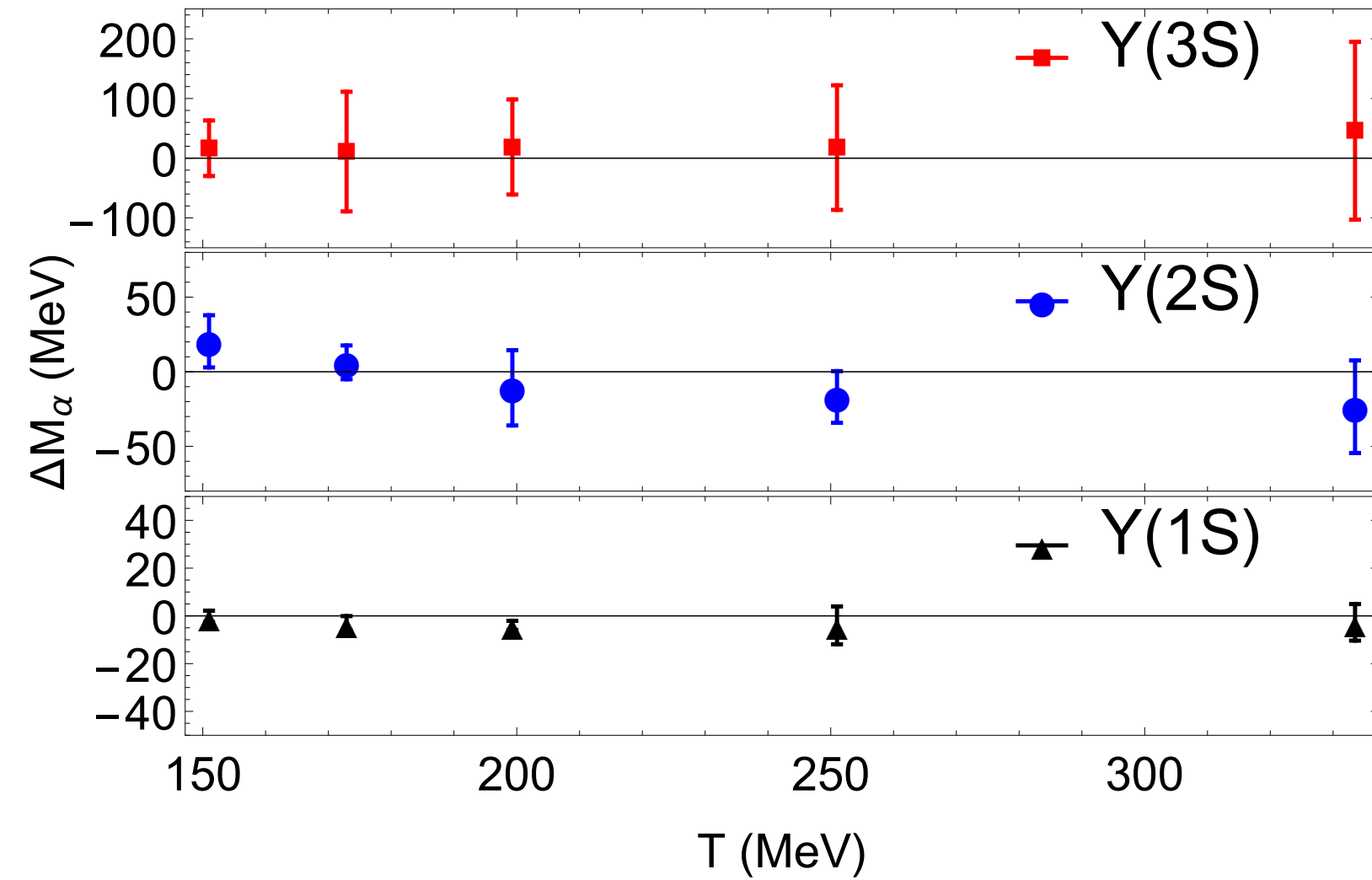
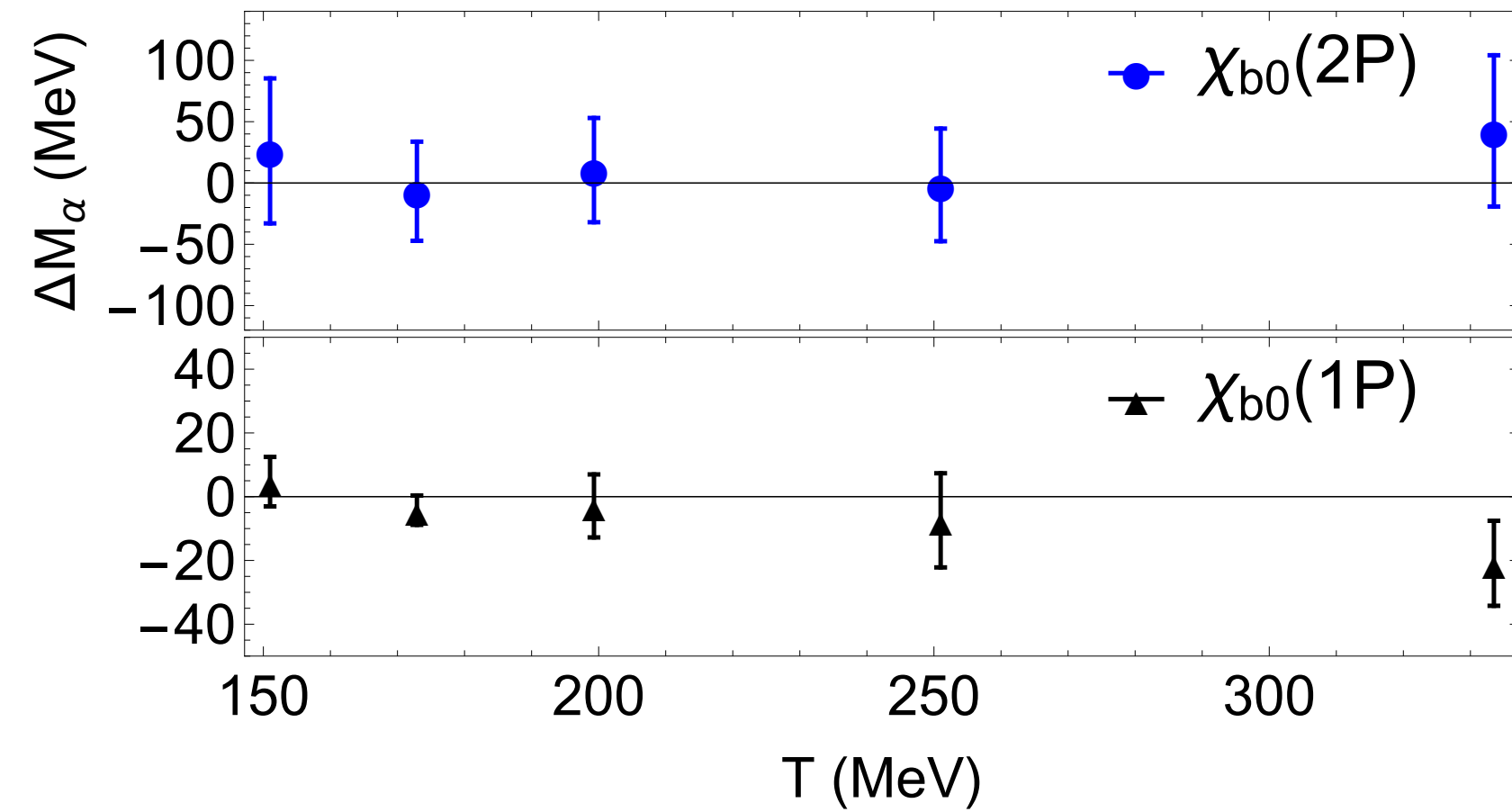
Ticket #: (arXiv)2105.07862

## Outline

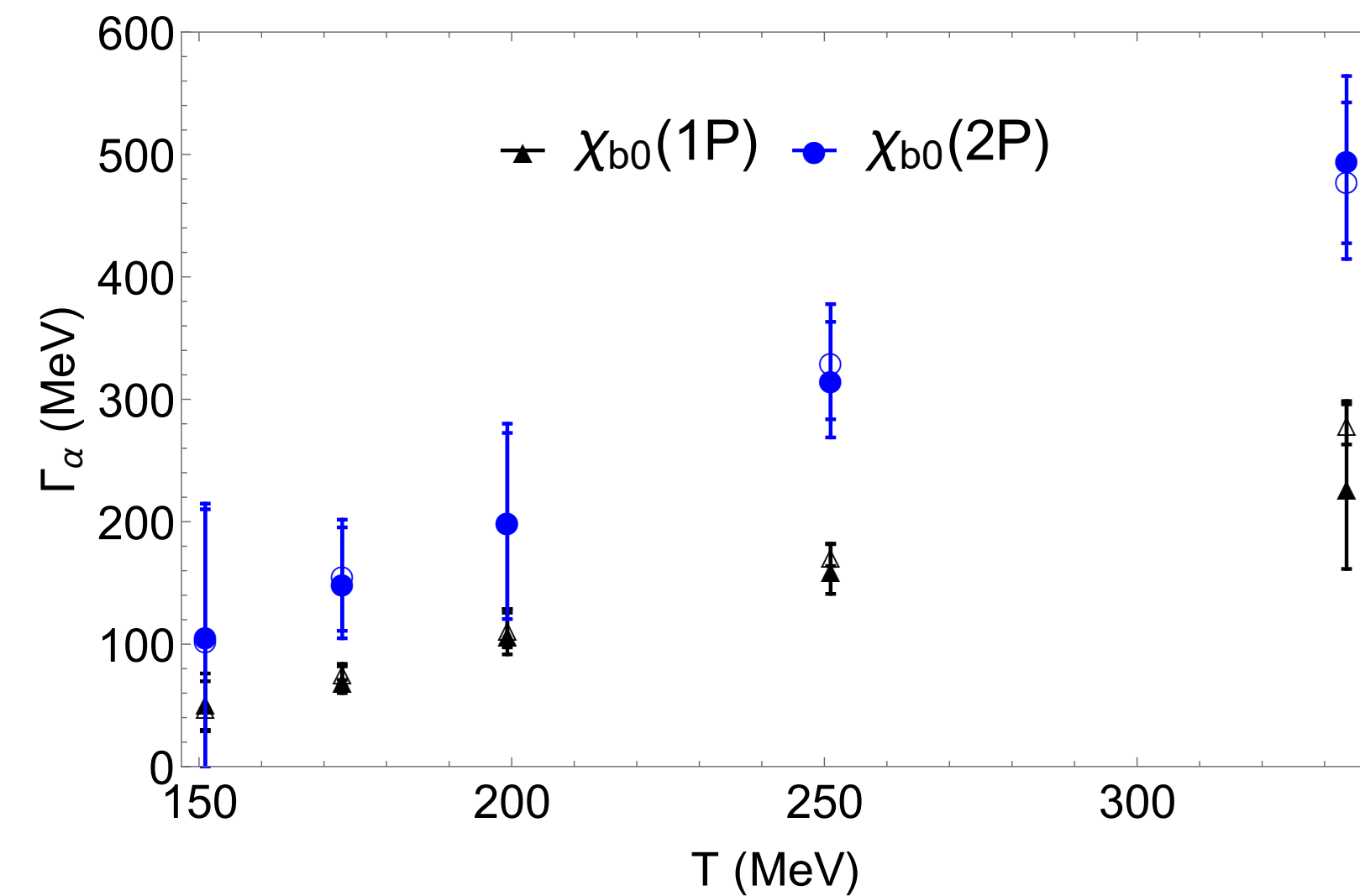
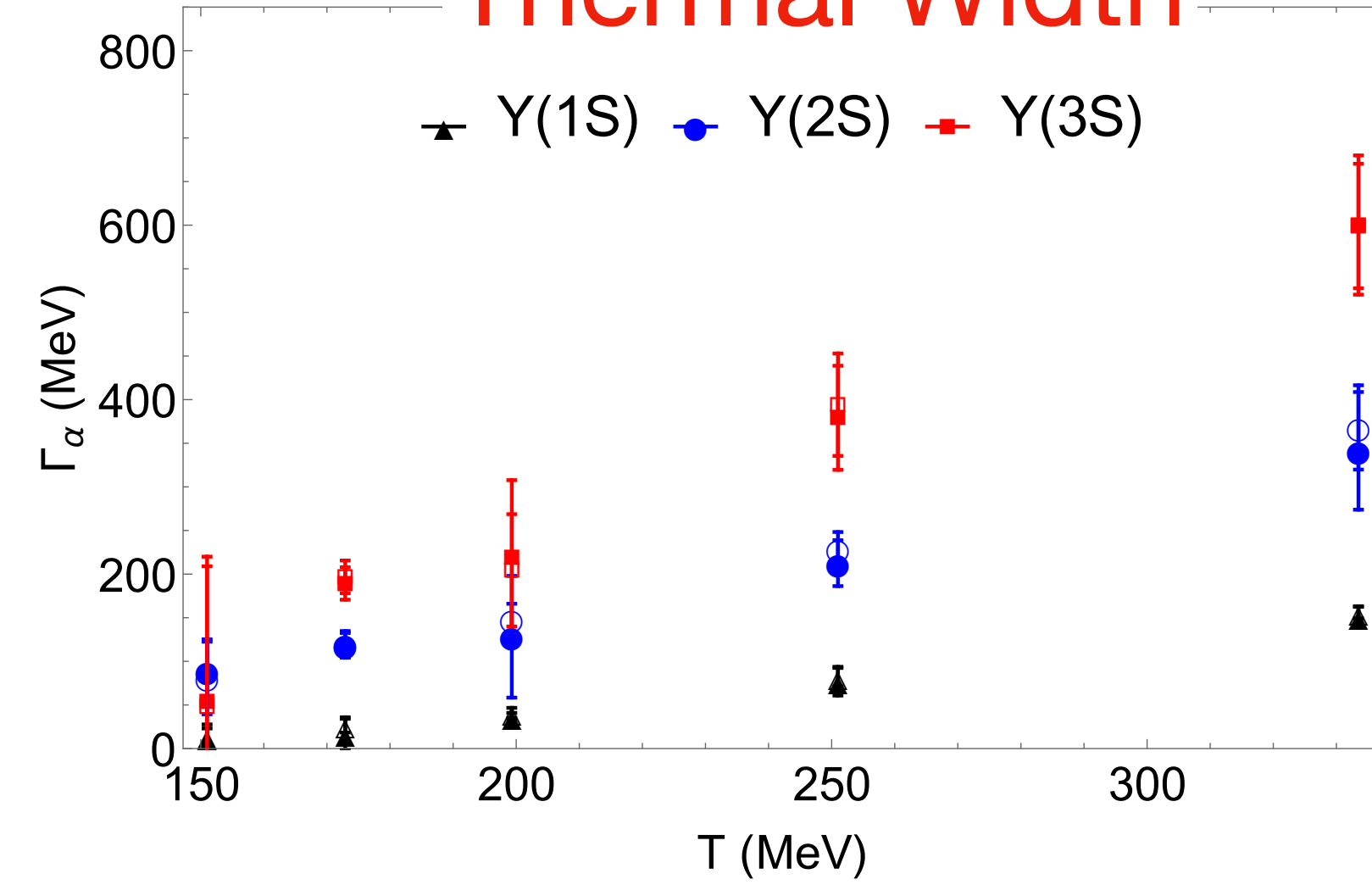
- Task: obtain  $V_{Q\bar{Q}}(T, r)$  from LQCD bottomonium masses & thermal widths
- Methodology
  - new algorithm using DNN to extract  $V_{Q\bar{Q}}(T, r)$
  - closure tests
- Results.

- 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  $(Q\bar{Q})_1 \rightarrow (Q\bar{Q})_8$ , Landau damping, ...

## Mass

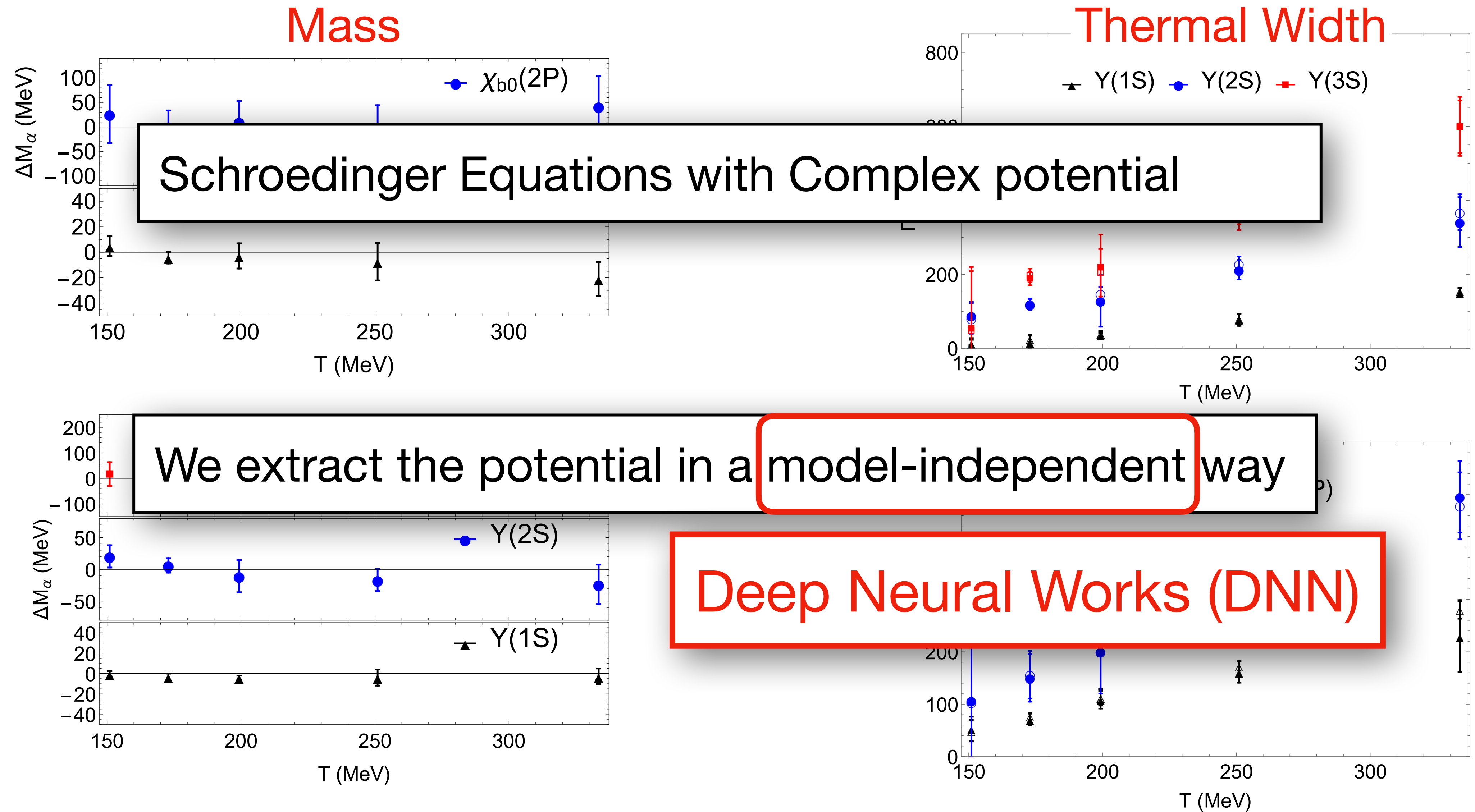


## Thermal Width



R. Larsen, S. Meinel, S. Mukherjee, and P. Petreczky:

Phys.Rev.D100,074506(2019), Phys.Lett.B800,135119(2020), Phys.Rev.D102,114508(2020)



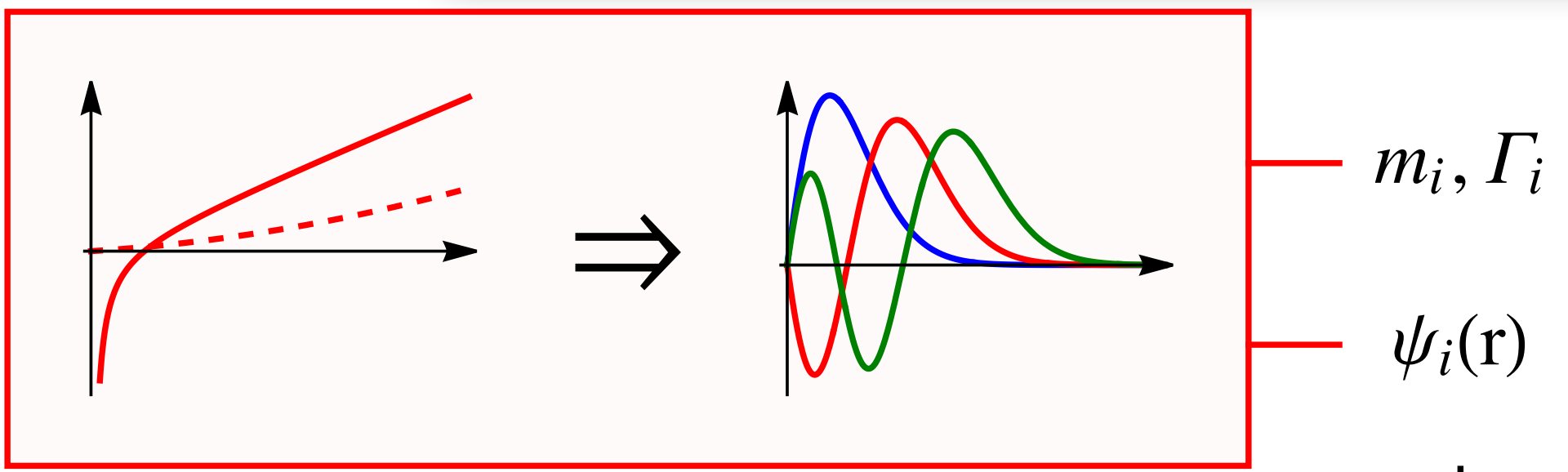
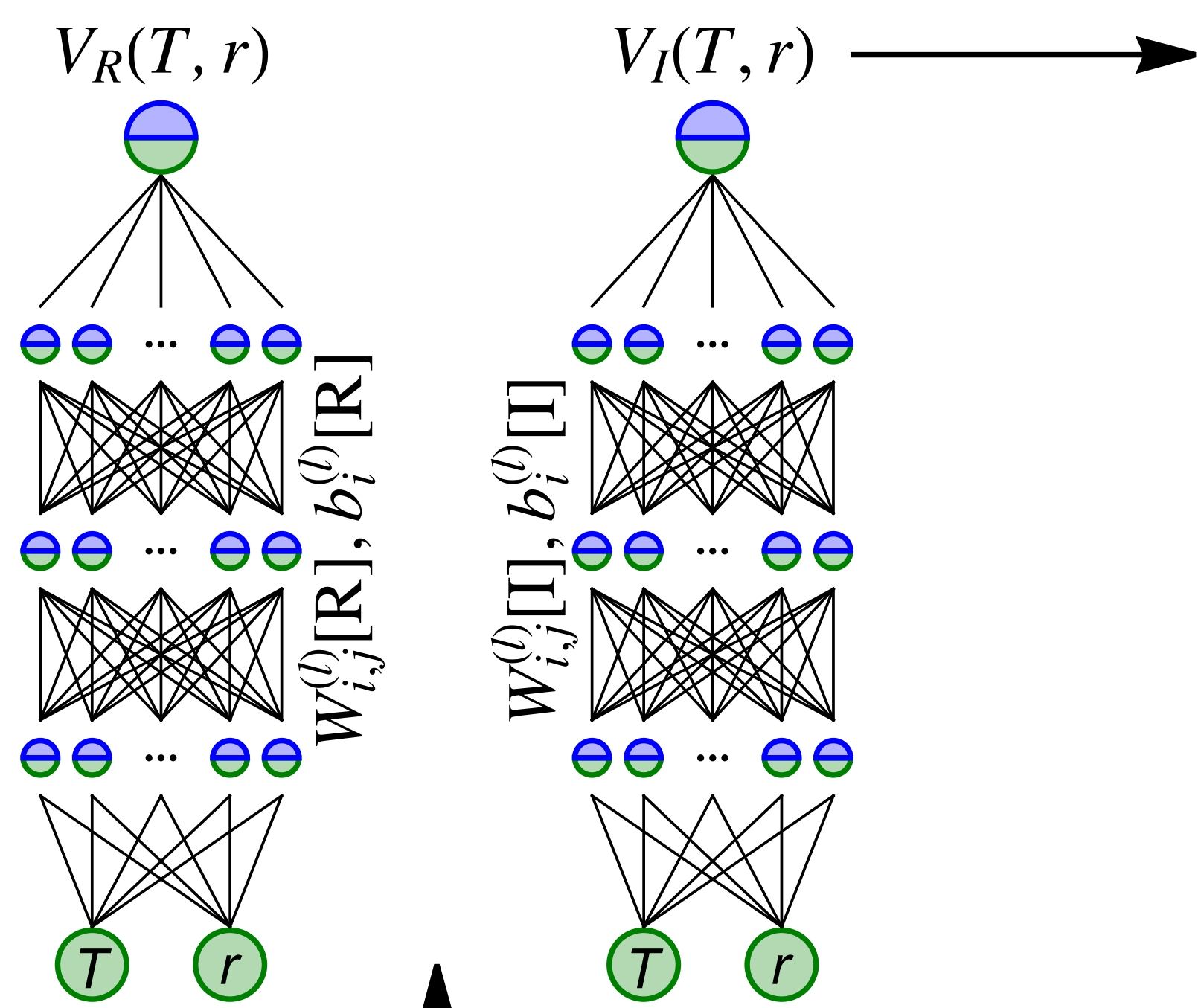
R. Larsen, S. Meinel, S. Mukherjee, and P. Petreczky:

Phys.Rev.D100,074506(2019), Phys.Lett.B800,135119(2020), Phys.Rev.D102,114508(2020)

# How to learn potential using DNN?

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

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



Schrödinger Eq. Solver

$$\text{update } \Delta W_{i,j}^{(l)} \sim -\frac{\partial J}{\partial W^{(l)}}, \quad \Delta b_i^{(l)} \sim -\frac{\partial J}{\partial b^{(l)}} \quad \leftarrow \quad \chi^2, \frac{\delta \chi^2}{\delta V(r)}$$

compare with lattice-QCD

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

# 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**,

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

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

update  $\Delta W_{i,j}^{(l)} \sim -\frac{\partial \chi^2}{\partial W^{(l)}}, \Delta b_i^{(l)} \sim -\frac{\partial \chi^2}{\partial b^{(l)}} \longleftarrow \chi^2, \frac{\partial \chi}{\partial V(r)}$

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

# 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**,

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

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

$$J = \chi^2 + \text{regularizer} \quad (\text{implicit function of } W \text{ and } b)$$

update  $\Delta W_{i,j}^{(l)} \sim -\frac{\partial \chi^2}{\partial W^{(l)}}$ ,  $\Delta b_i^{(l)} \sim -\frac{\partial \chi^2}{\partial b^{(l)}}$  ←  $\chi^2, \frac{\partial \chi^2}{\partial V(r)}$

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



# 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**,

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

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

$$J = \chi^2 + \text{regularizer} \quad (\text{implicit function of } W \text{ and } b)$$

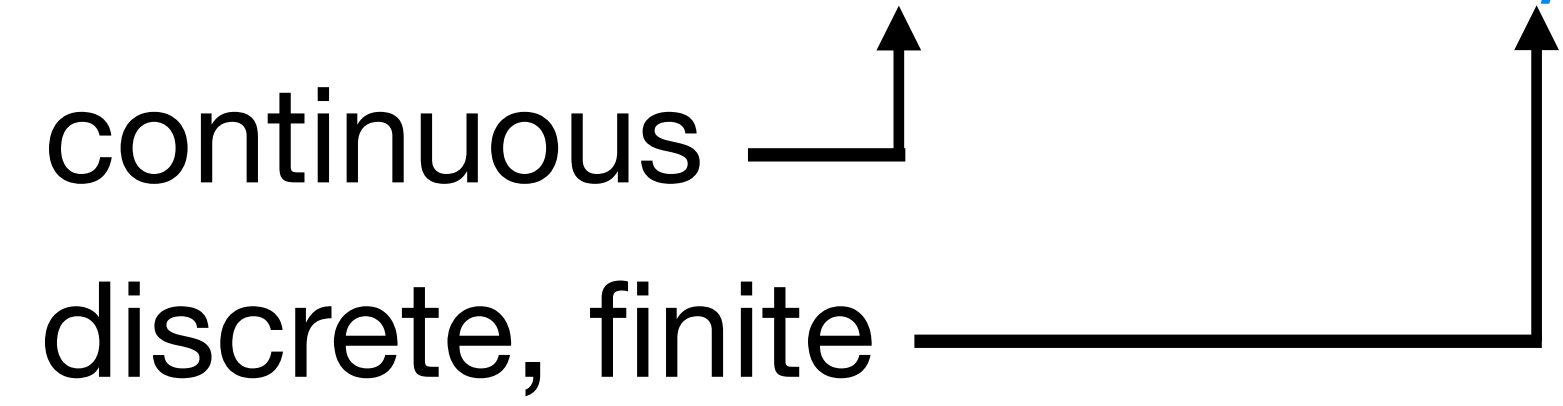
$\partial_W J, \partial_b J$  can be computed exactly, thanks to the perturbation theory.

$$\text{update} \quad \Delta W_{i,j}^{(l)} \sim -\frac{\partial J}{\partial W^{(l)}}, \quad \Delta b_i^{(l)} \sim -\frac{\partial J}{\partial b^{(l)}} \longleftarrow \chi^2, \frac{\partial \chi}{\partial V(r)}$$

$$\begin{aligned} \delta m_i &= \langle \psi_i | \delta V_R(r) | \psi_i \rangle, \\ \delta \Gamma_i &= -\langle \psi_i | \delta V_I(r) | \psi_i \rangle. \end{aligned}$$

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

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

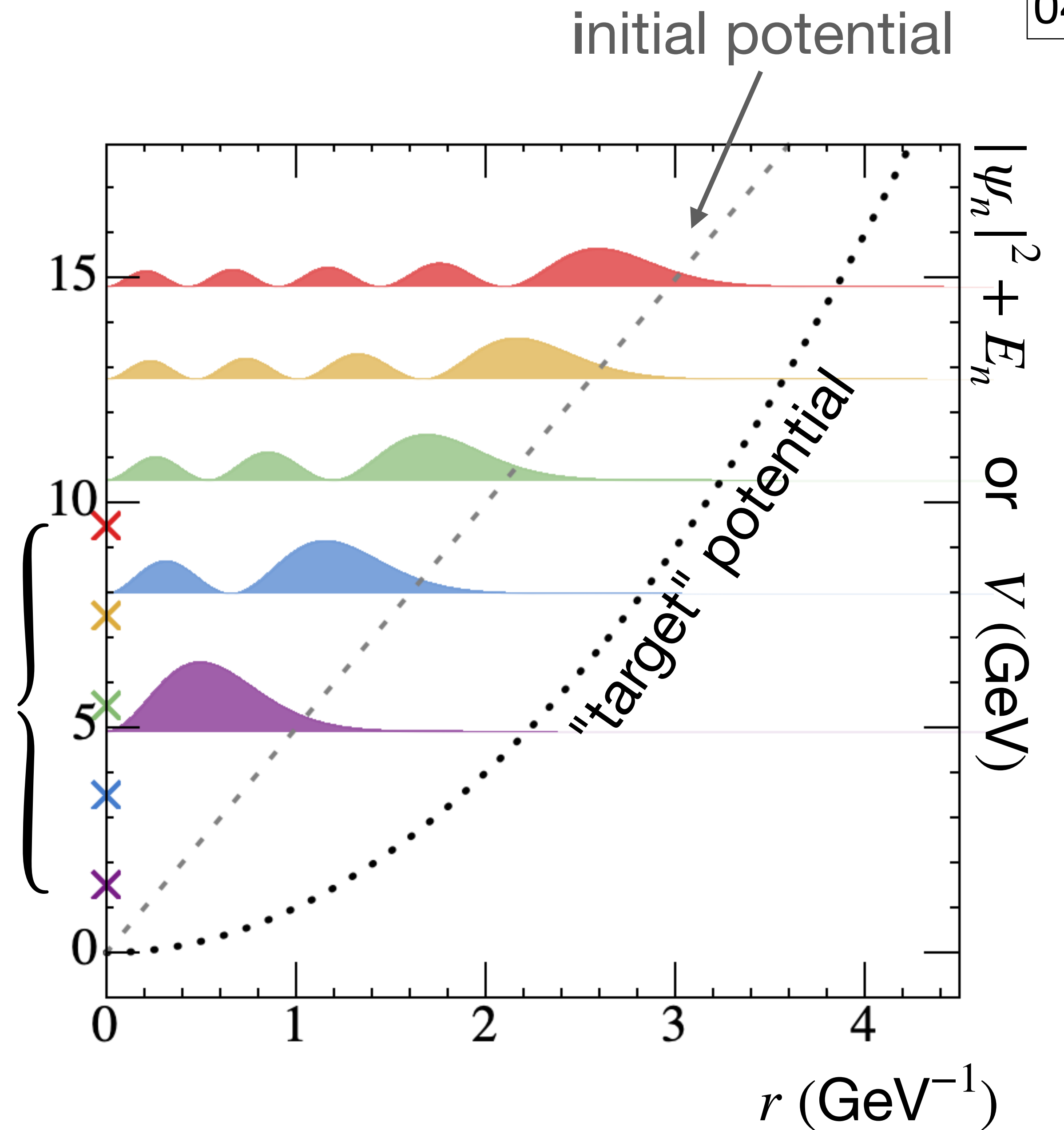


# Test I - Can we learn $V(r)$ from $\{E_n\}$ ?

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

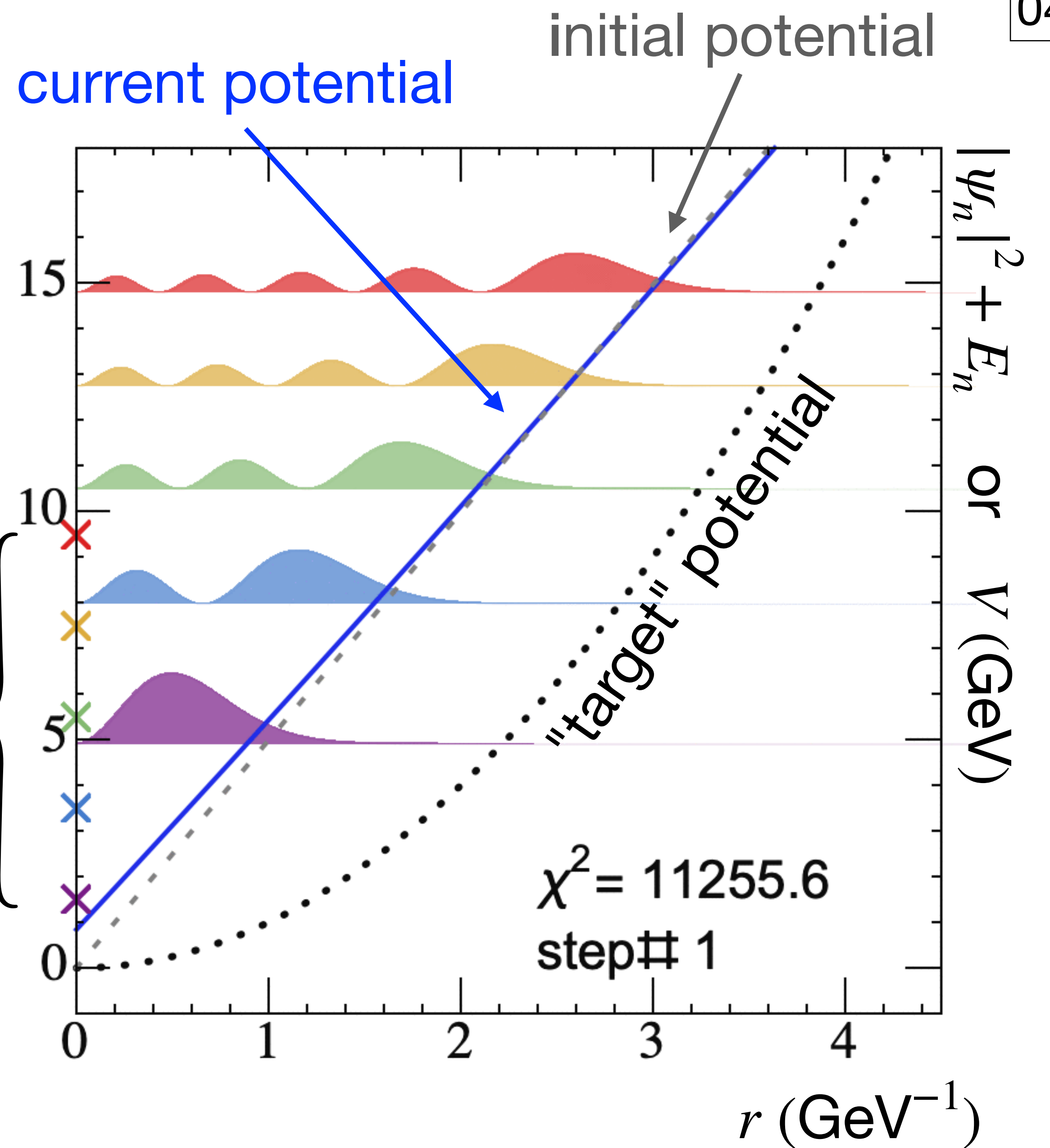


# Test I - Can we learn $V(r)$ from $\{E_n\}$ ?

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 I - Can we learn  $V(r)$  from  $\{E_n\}$ ?  
-- Yes! (for a certain  $r$  range)

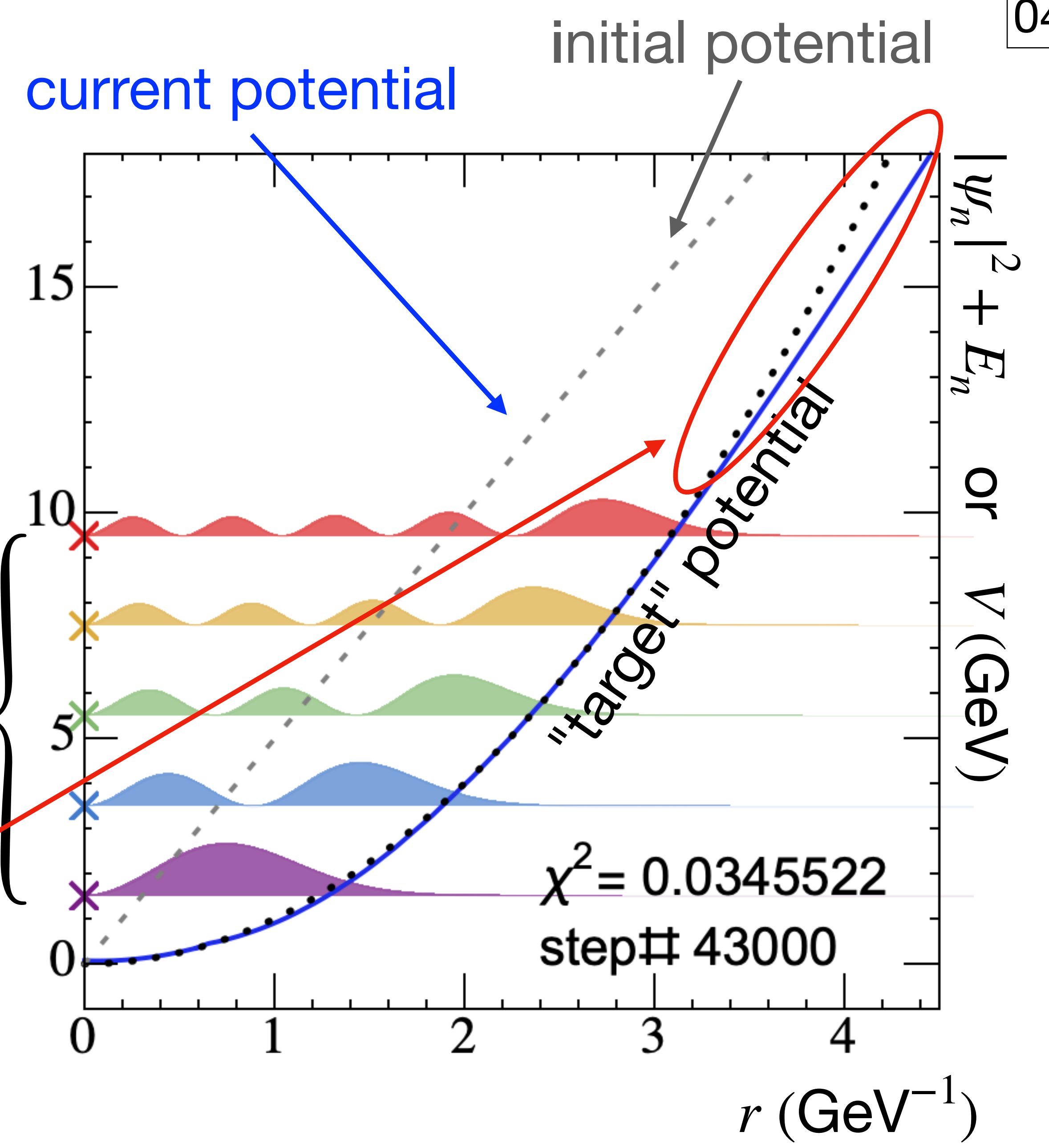
learn  $V(r)$  according to

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

target spectrum

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

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



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

- Start with a known potential

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

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

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

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

- Learn the potential using DNN

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

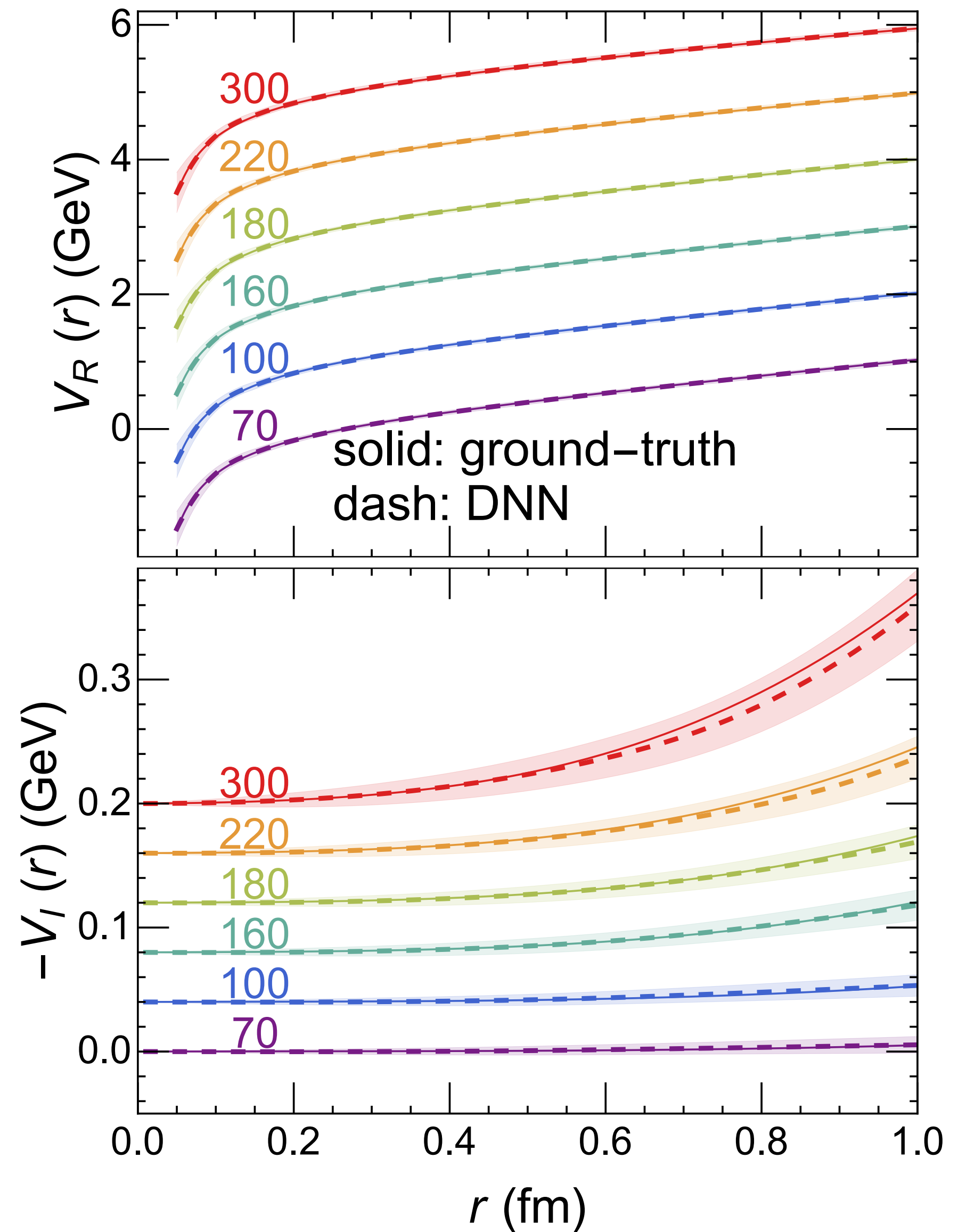
- Start with a known potential (solid line)

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

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

- Compute  $\{m_n, \Gamma_n\}$  at  $T = \{0, 151, 173, 199, 251, 334\}$  MeV

- Learn the potential using DNN (dash + band)



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

Gray region:  
extrapolations

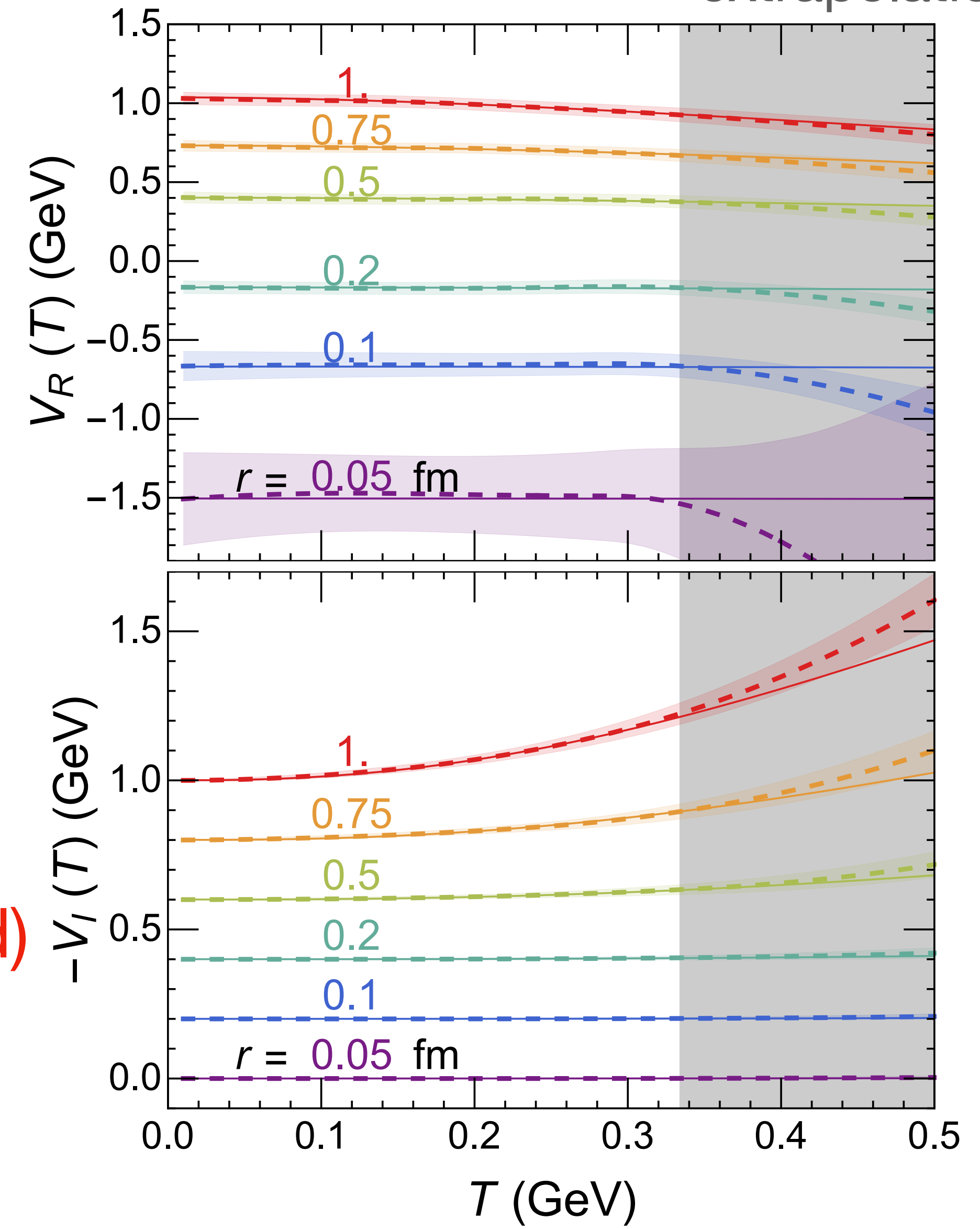
- Start with a known potential (solid line)

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

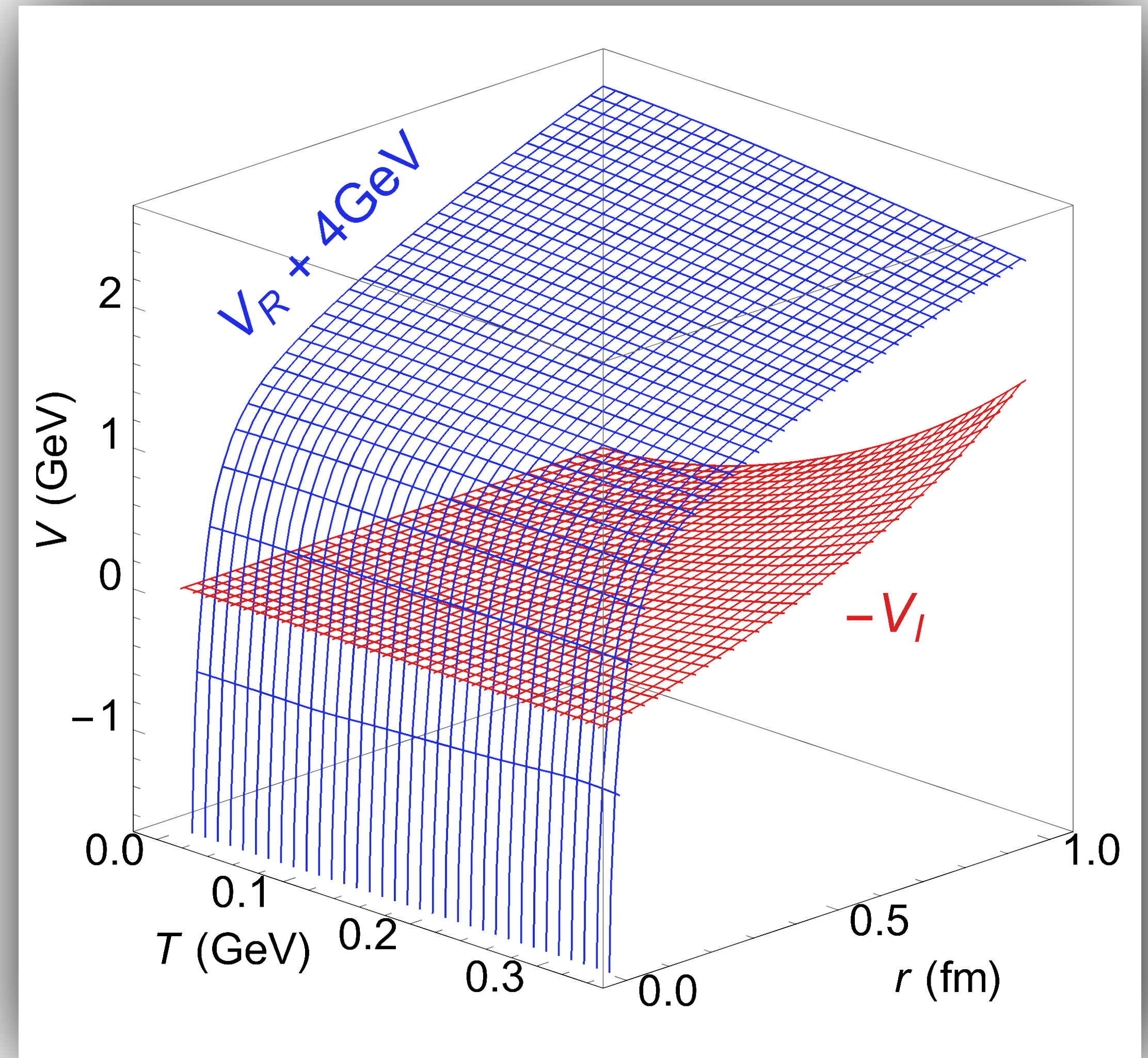
$$V_I(T, r) = -\frac{\sqrt{\pi}}{4} \mu_D T \sigma r^3 G_{2,4}^{2,2} \left( \begin{matrix} -\frac{1}{2}, -\frac{1}{2} \\ \frac{1}{2}, \frac{1}{2}, -\frac{3}{2}, -1 \end{matrix} \middle| \frac{\mu_D^2 r^2}{4} \right) - \alpha T \phi(\mu_D r).$$

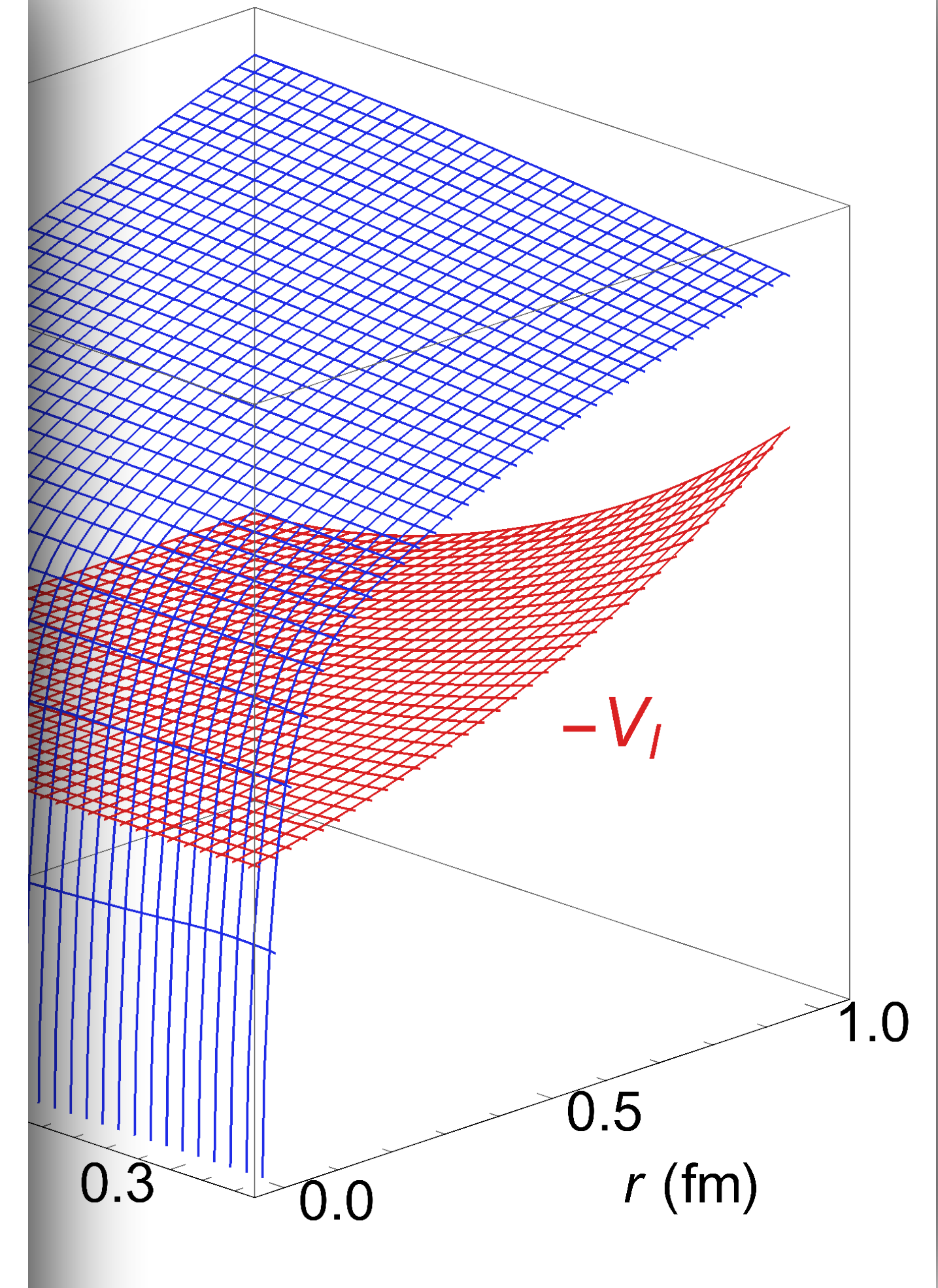
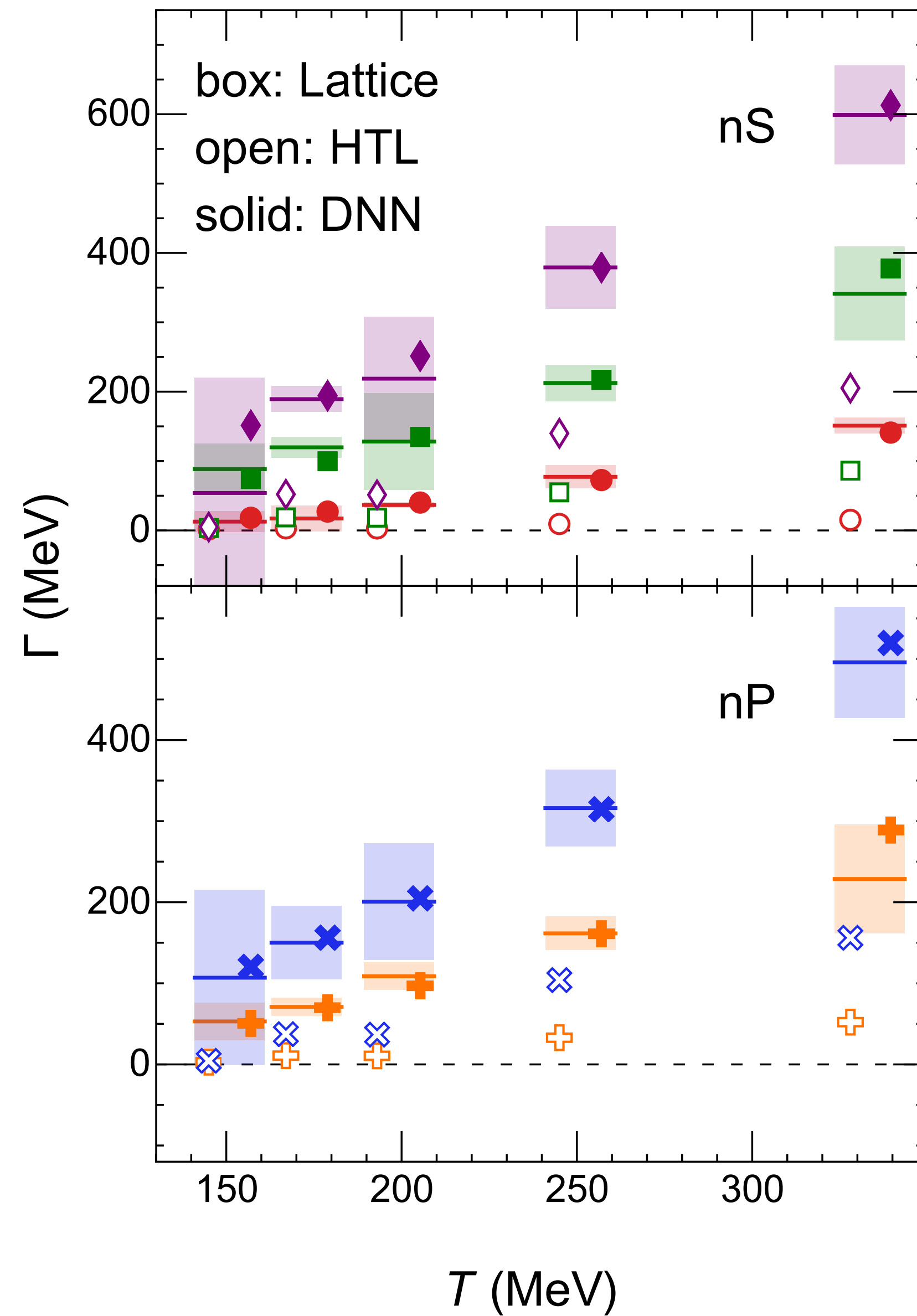
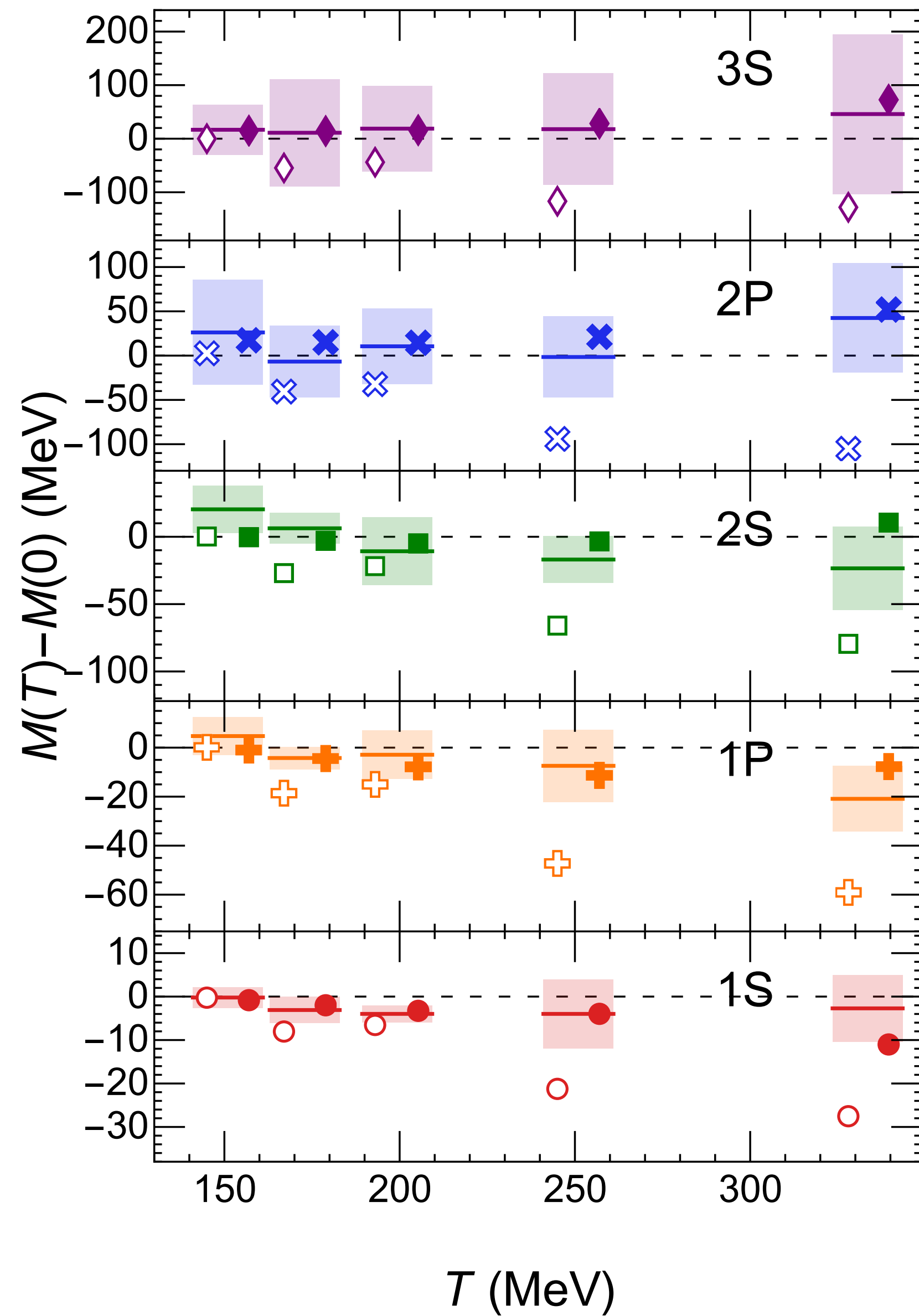
- Compute  $\{m_n, \Gamma_n\}$  at  
 $T = \{0, 151, 173, 199, 251, 334\}$  MeV

- Learn the potential using DNN (dash + band)









## 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  $\Gamma$ .
- Phenomenological consequences in heavy-ion collisions?

# **Back-up Slides**

## Uncertainty Quantification according to Bayesian inference

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

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

$\chi^2(\boldsymbol{\theta})$  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  $V_{\boldsymbol{\theta}}$ ,

$$P(V_{\boldsymbol{\theta}})dV = \text{Posterior}(\boldsymbol{\theta} \mid \text{data})d^N \boldsymbol{\theta}$$

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

$$P(V_{\theta})dV = \text{Posterior}(\boldsymbol{\theta} \mid \text{data})d^N\boldsymbol{\theta}$$

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

$$w_i = P(V_{\boldsymbol{\theta}_i})dV_i = \text{Posterior}(\boldsymbol{\theta}_i \mid \text{data})$$

Computationally Expensive!

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

$$P(V_{\theta})dV = \text{Posterior}(\boldsymbol{\theta} \mid \text{data})d^N\boldsymbol{\theta}$$

- Method 2)

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

$$w_i = P(V_{\boldsymbol{\theta}_i})dV_i = 1$$

Hard to sample according to Posterior!

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

$$P(V_{\theta})dV = \text{Posterior}(\boldsymbol{\theta} \mid \text{data})d^N\boldsymbol{\theta}$$

- 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  $d^N\boldsymbol{\theta}_i = 1/\tilde{P}(\boldsymbol{\theta}_i)$ ;
  - Compute  $V_{\boldsymbol{\theta}_i}(r)$ ,  $\chi_{\boldsymbol{\theta}_i}^2$ , and  $\text{Posterior}(\boldsymbol{\theta}_i \mid \text{data})$ ;
  - For given  $r$ , histogram  $V_{\boldsymbol{\theta}_i}(r)$  with weights

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



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

$$P(V_{\theta})dV = \text{Posterior}(\boldsymbol{\theta} \mid \text{data})d^N\boldsymbol{\theta}$$

- Method 3)

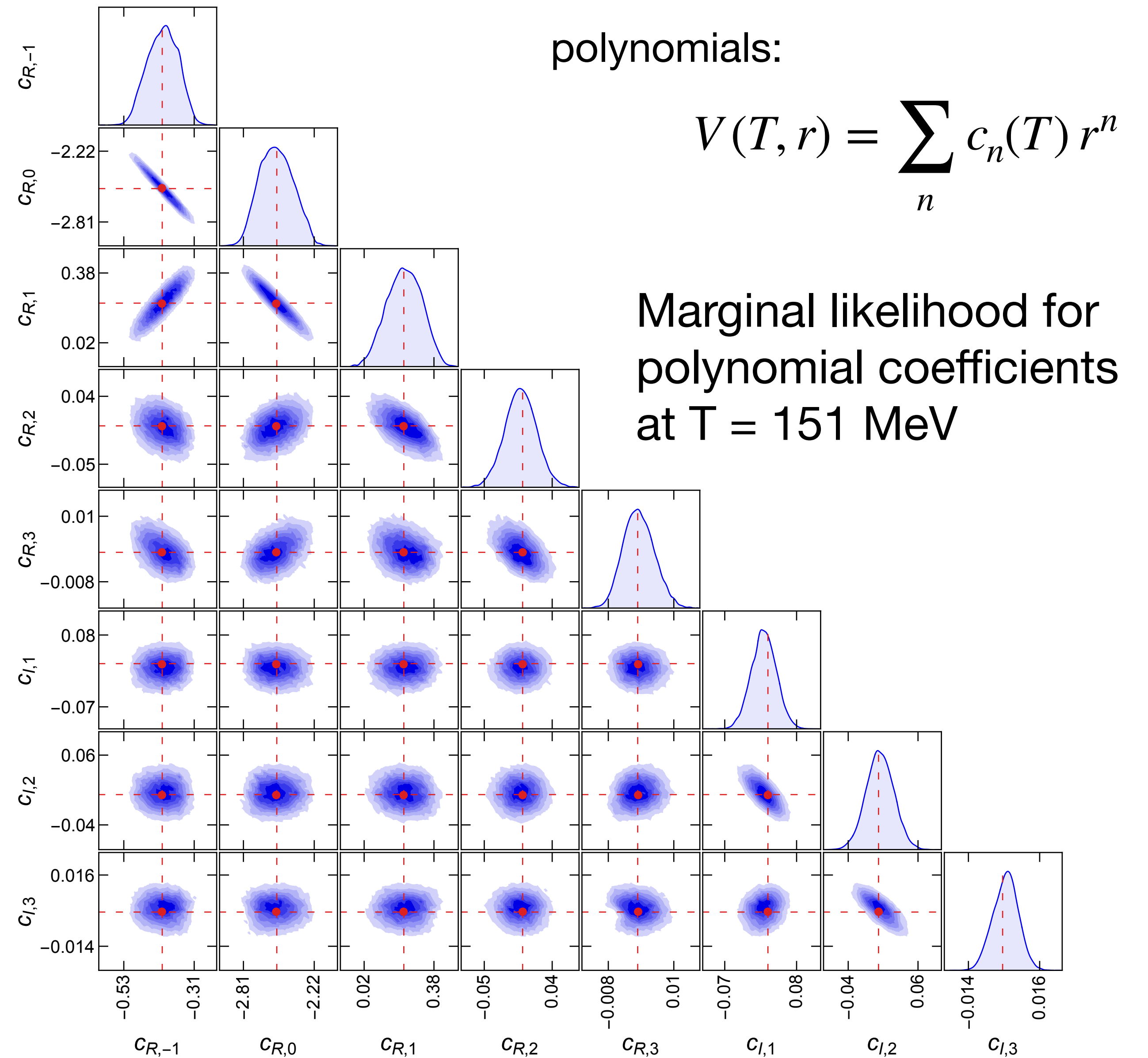
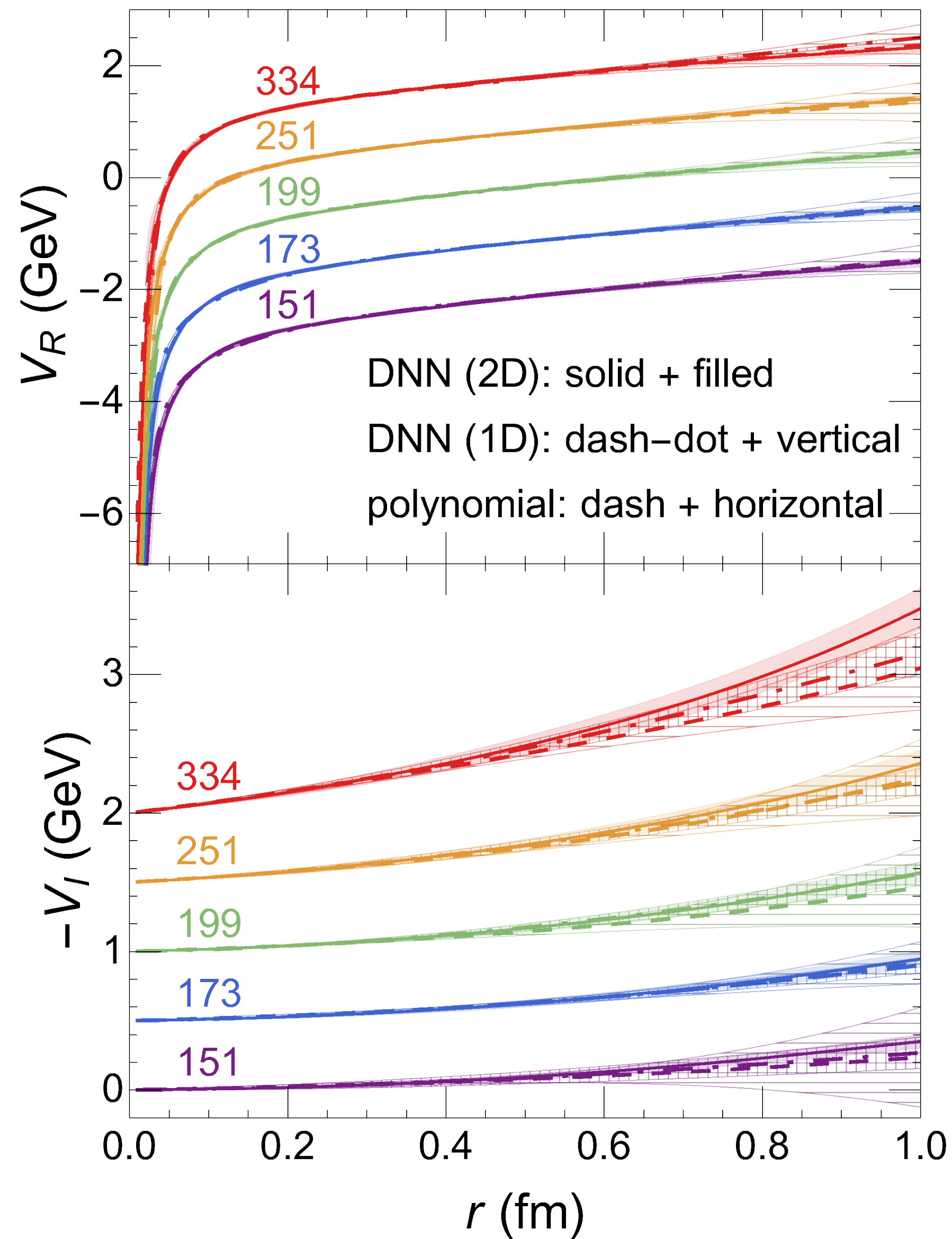
- Sample  $\{\boldsymbol{\theta}_i\}$  according to a reference distribution:  $P(\boldsymbol{\theta}) = \tilde{P}(\boldsymbol{\theta})$ ;
- Each data point corresponds to the element volume  $d^N\boldsymbol{\theta}_i = 1/\tilde{P}(\boldsymbol{\theta}_i)$ ;
- Compute  $V_{\boldsymbol{\theta}_i}(r)$ ,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

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

- In practice:

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

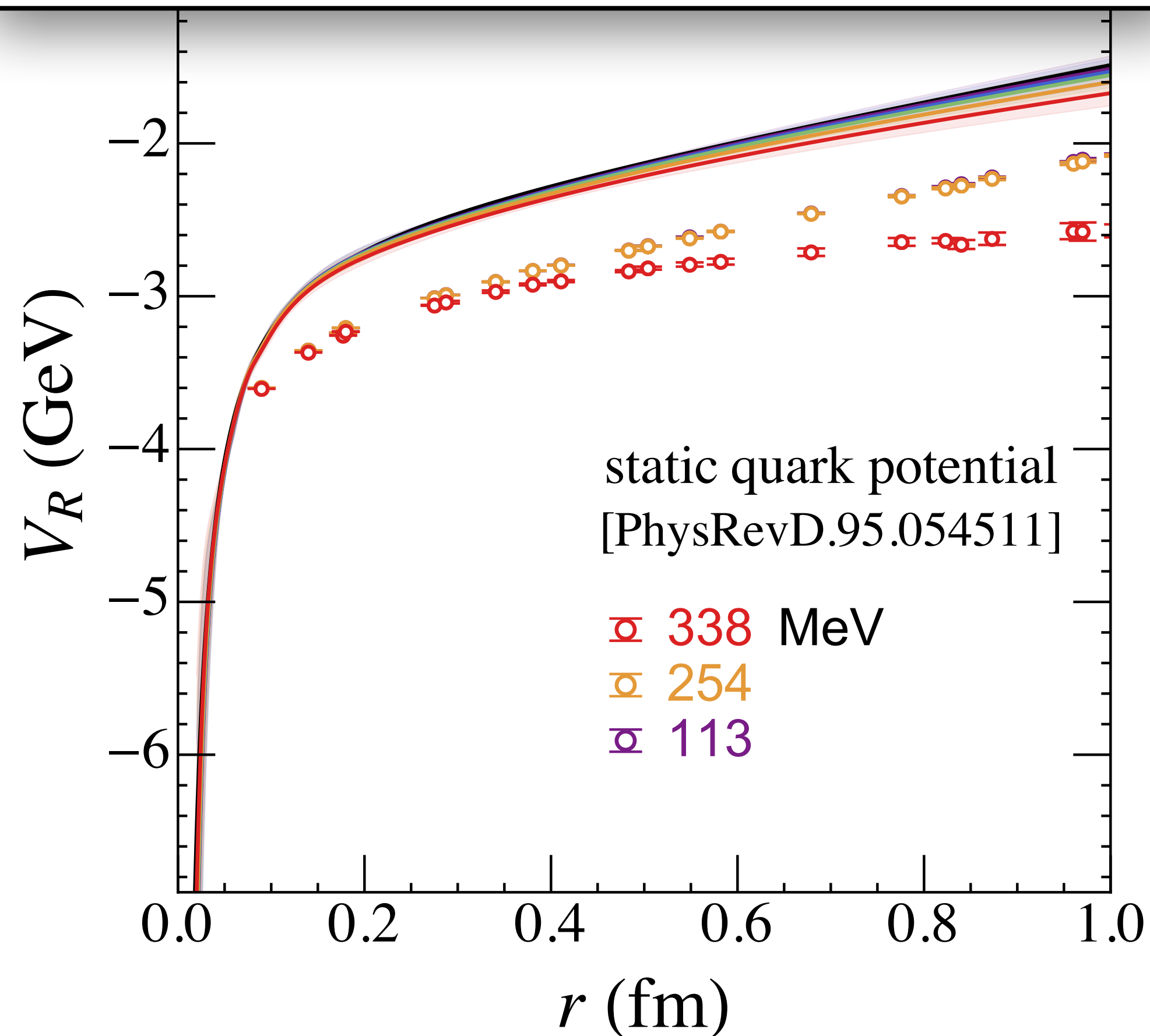
# Consistency test with $T$ -indept. DNN and polynomial



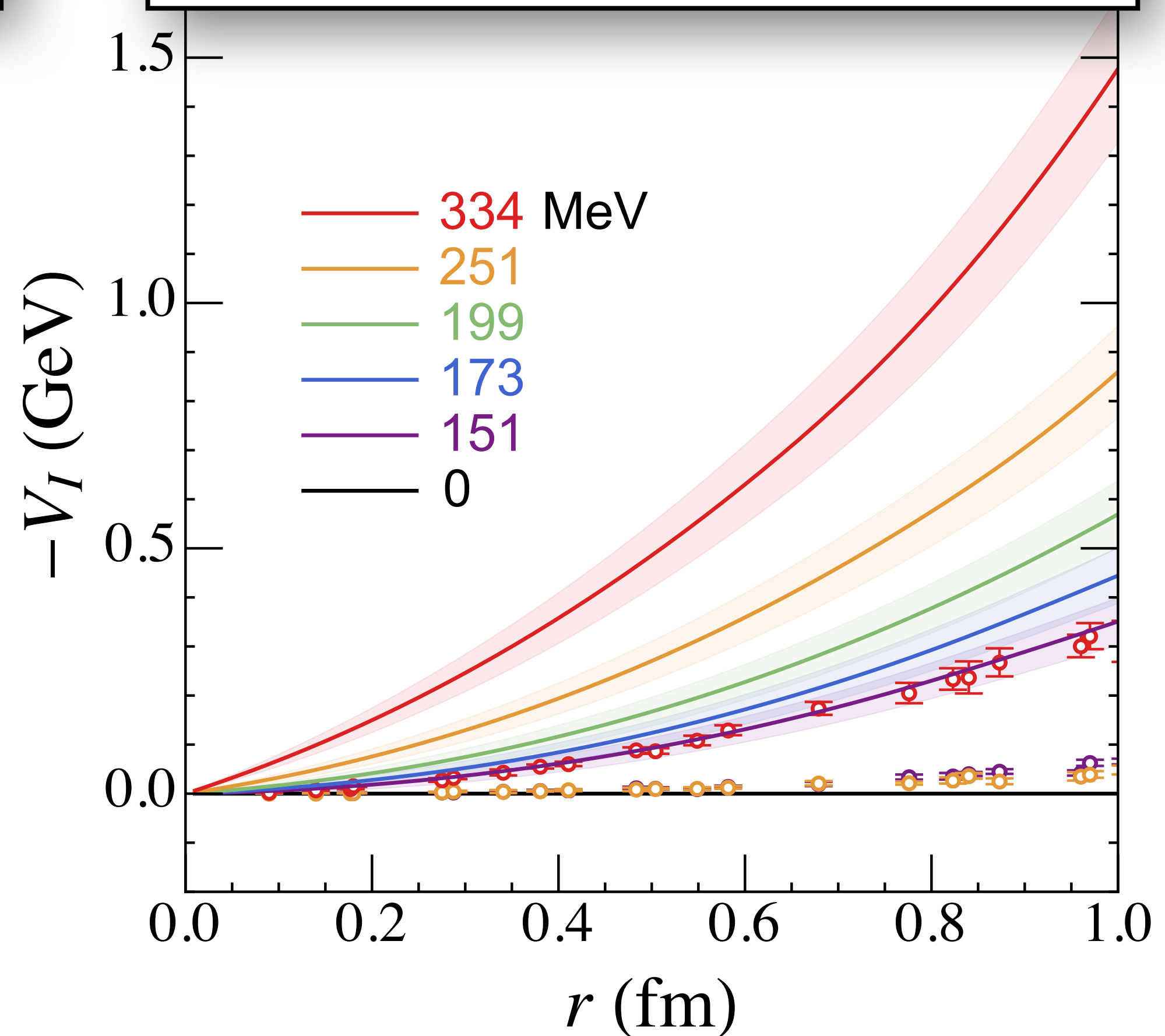
# What physics we have learned from $V_{\text{DNN}}(T, r)$ ?

--- compare with lattice result (static quark potential)

weaker  $T$  dependence for  $V_R(r)$



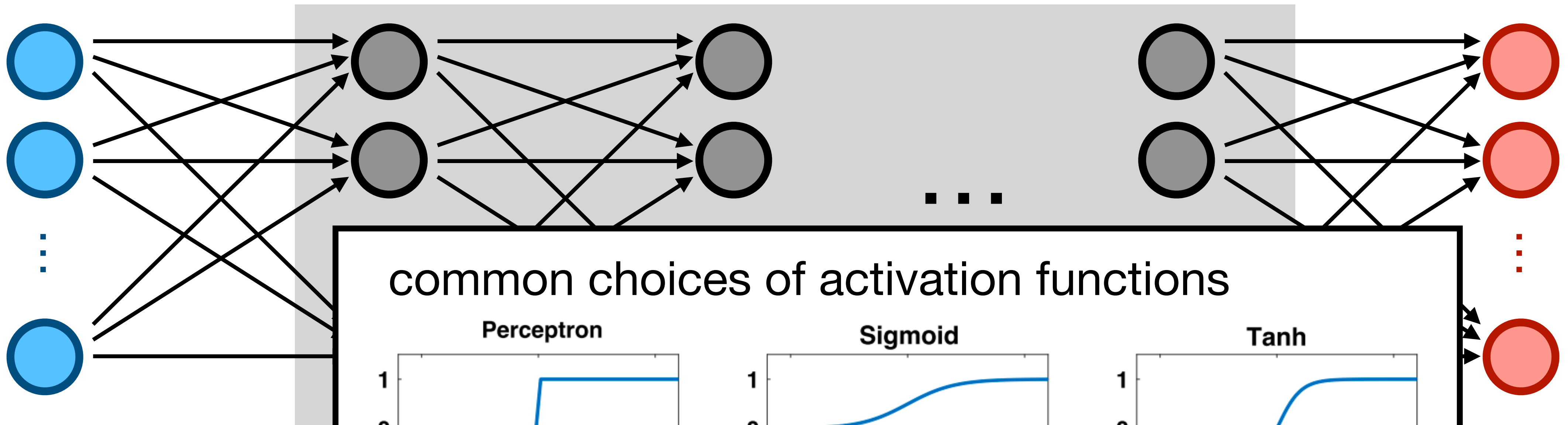
larger magnitude for  $V_I(r)$



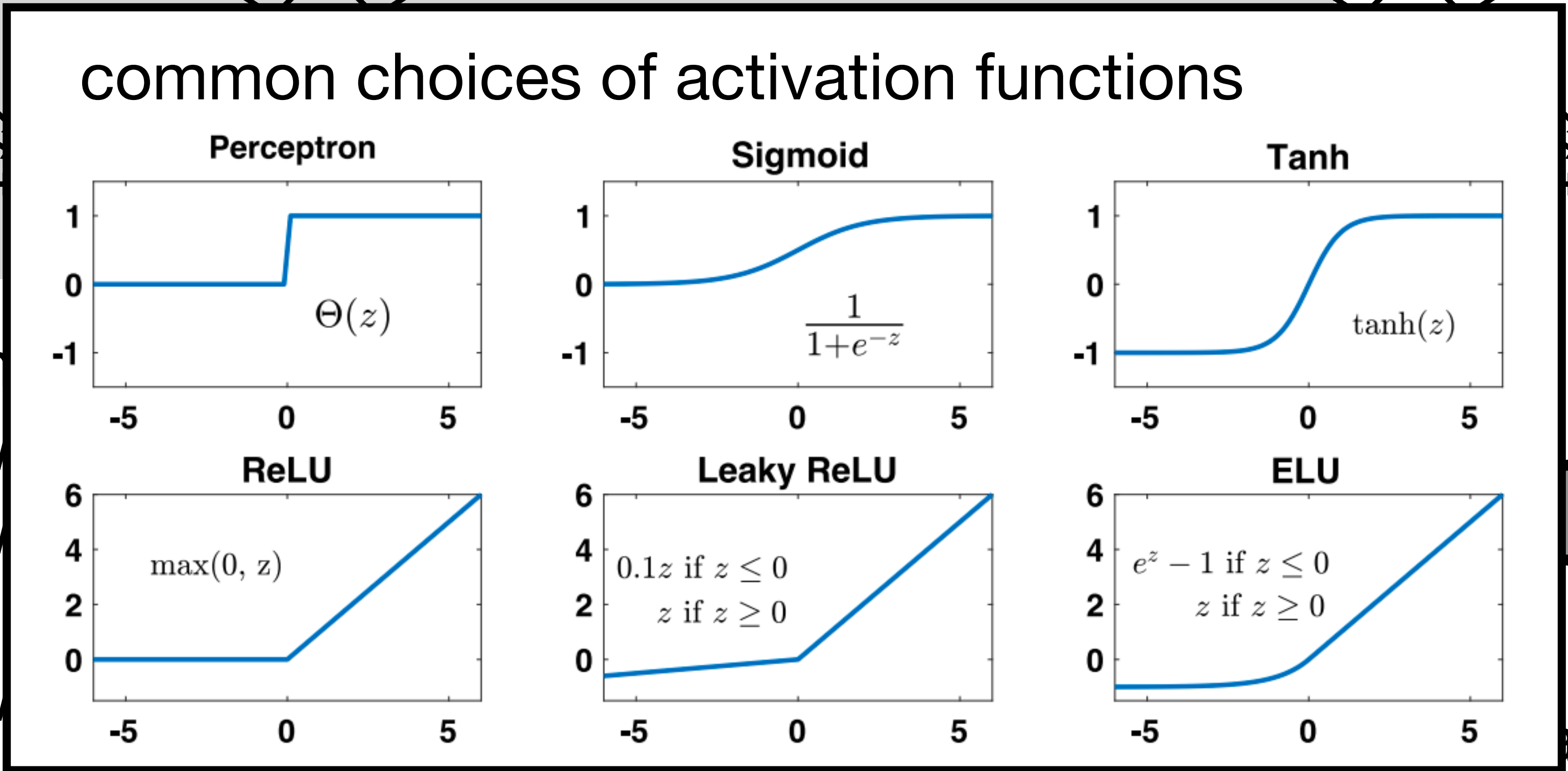
input layer

hidden layers

output layer



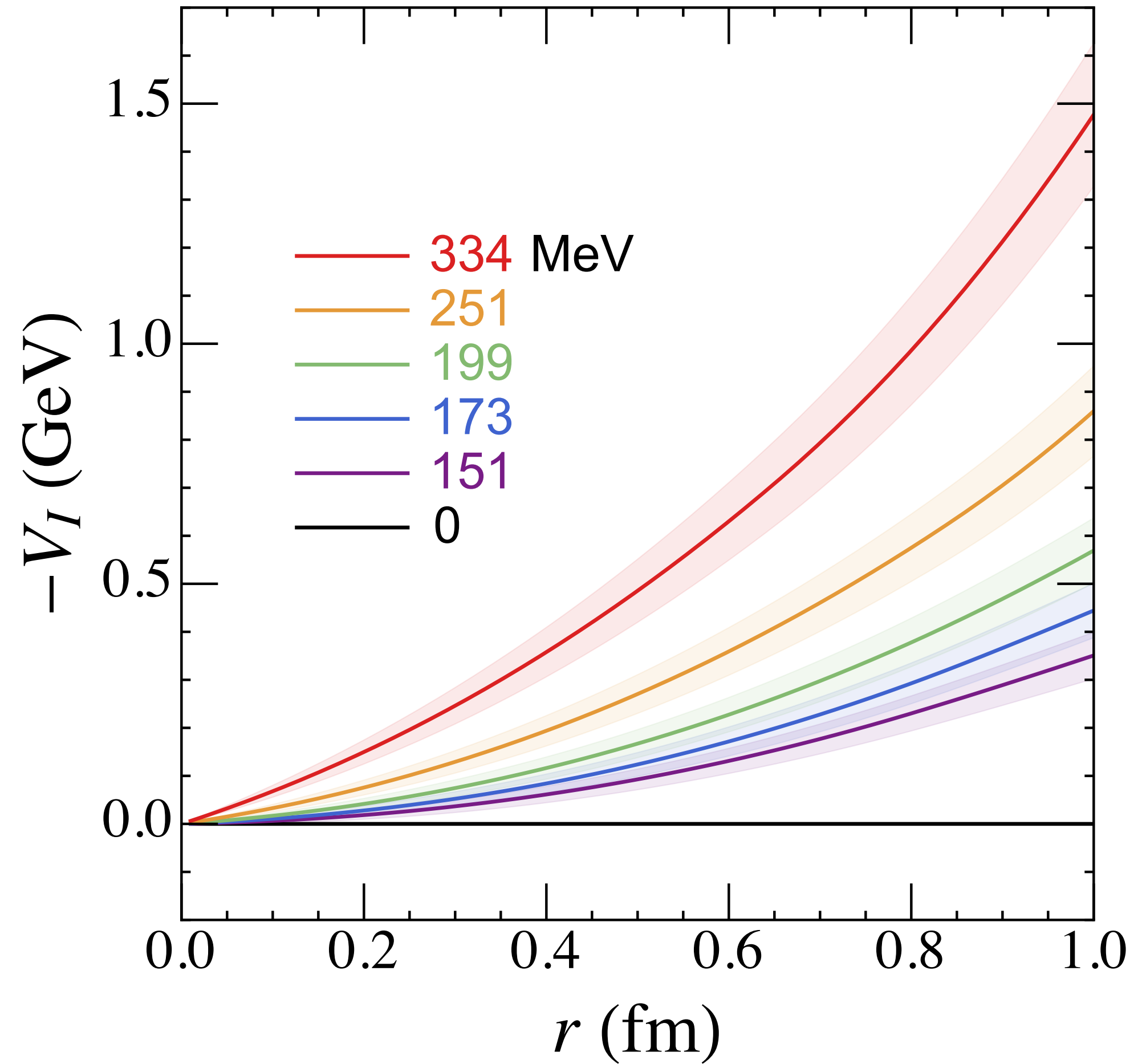
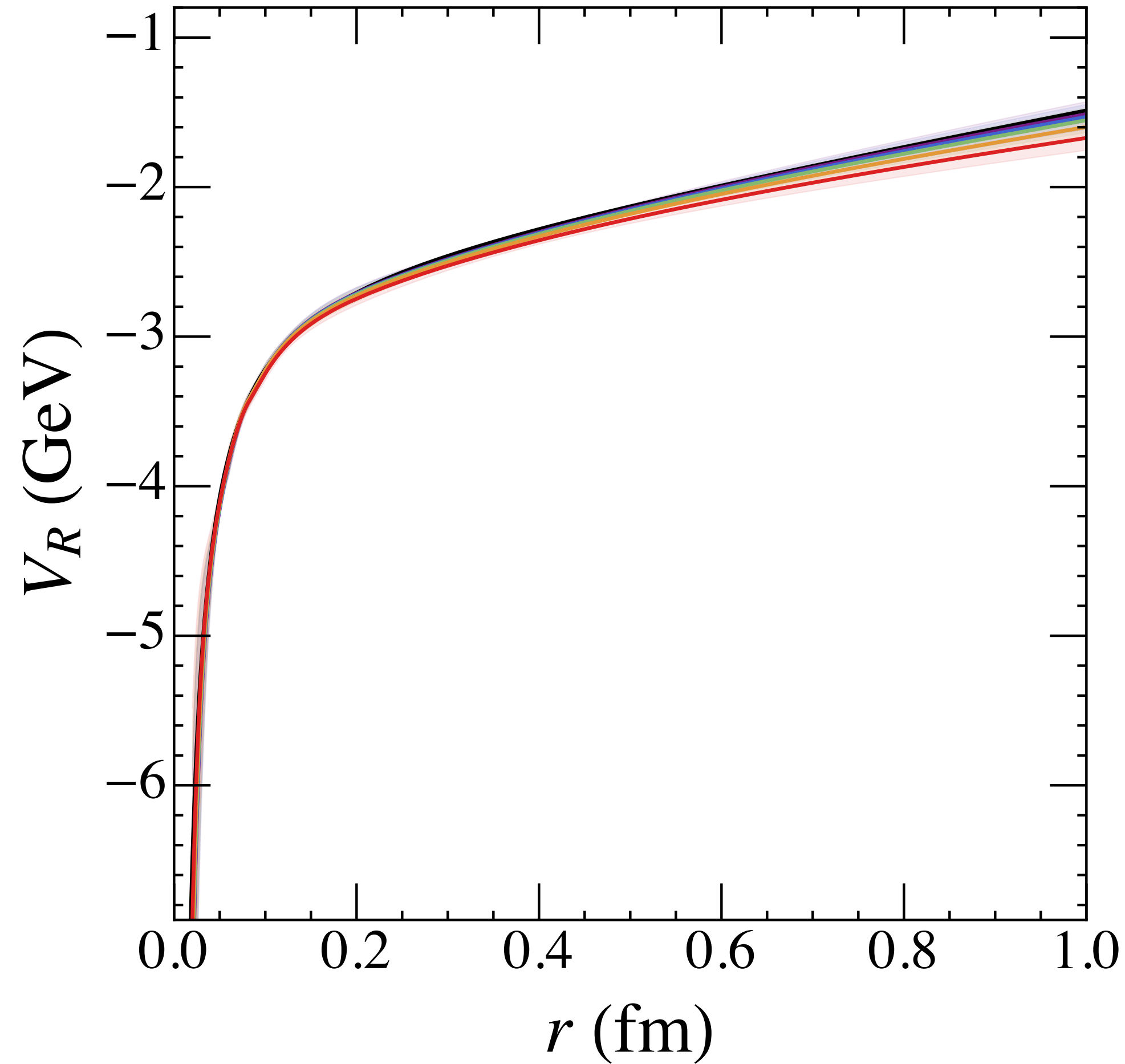
- Multiple "hidden layers"
  - $v^{(1)}_j = \sigma^{(1)}(\sum v^{(0)}_i w_{ij})$
  - $v^{(2)}_j = \sigma^{(2)}(\sum v^{(1)}_i w_{ij})$
  - ...
  - $y_j = \sigma^{(l+1)}(\sum v^{(l)}_i w_{ij})$



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

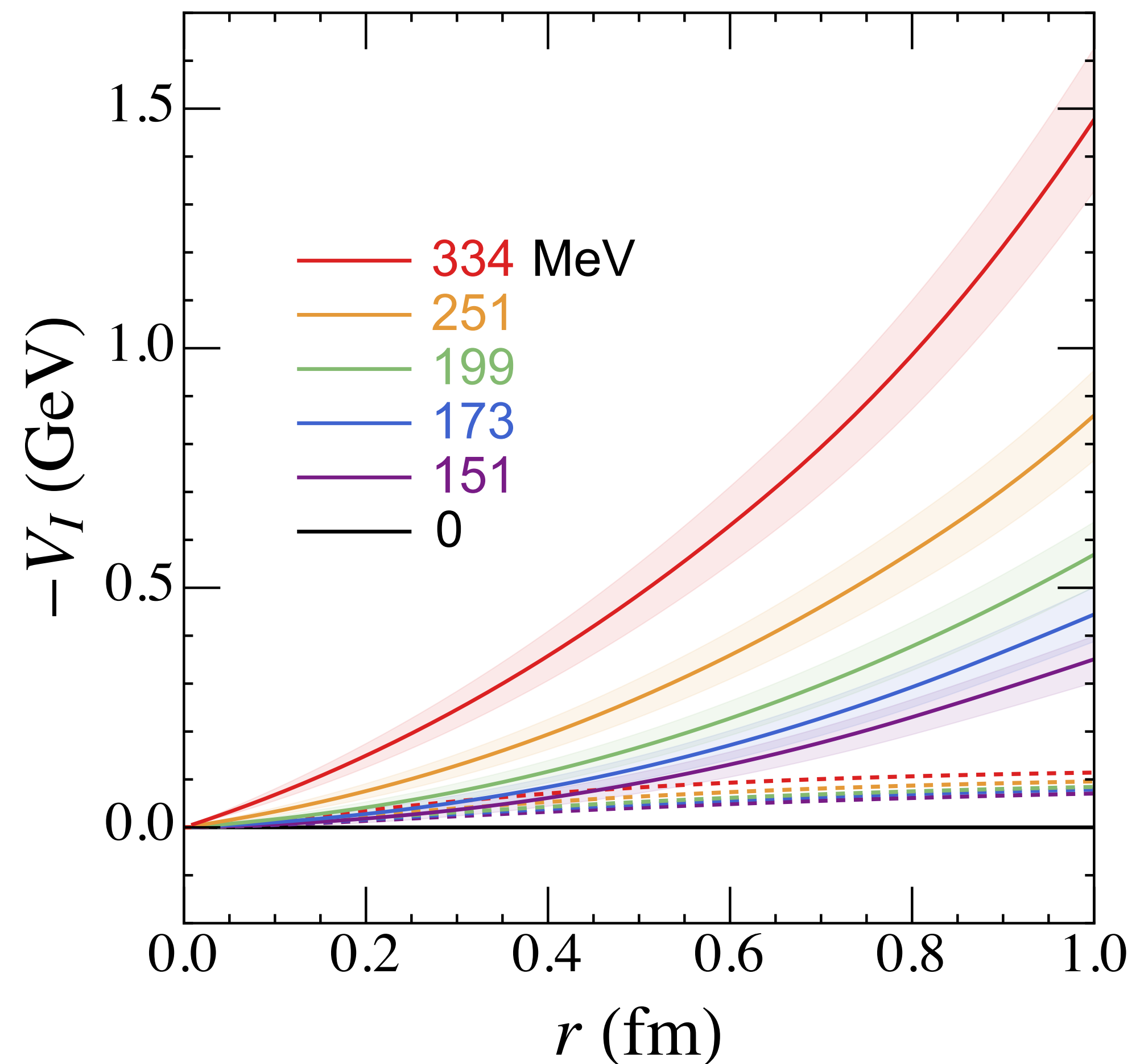
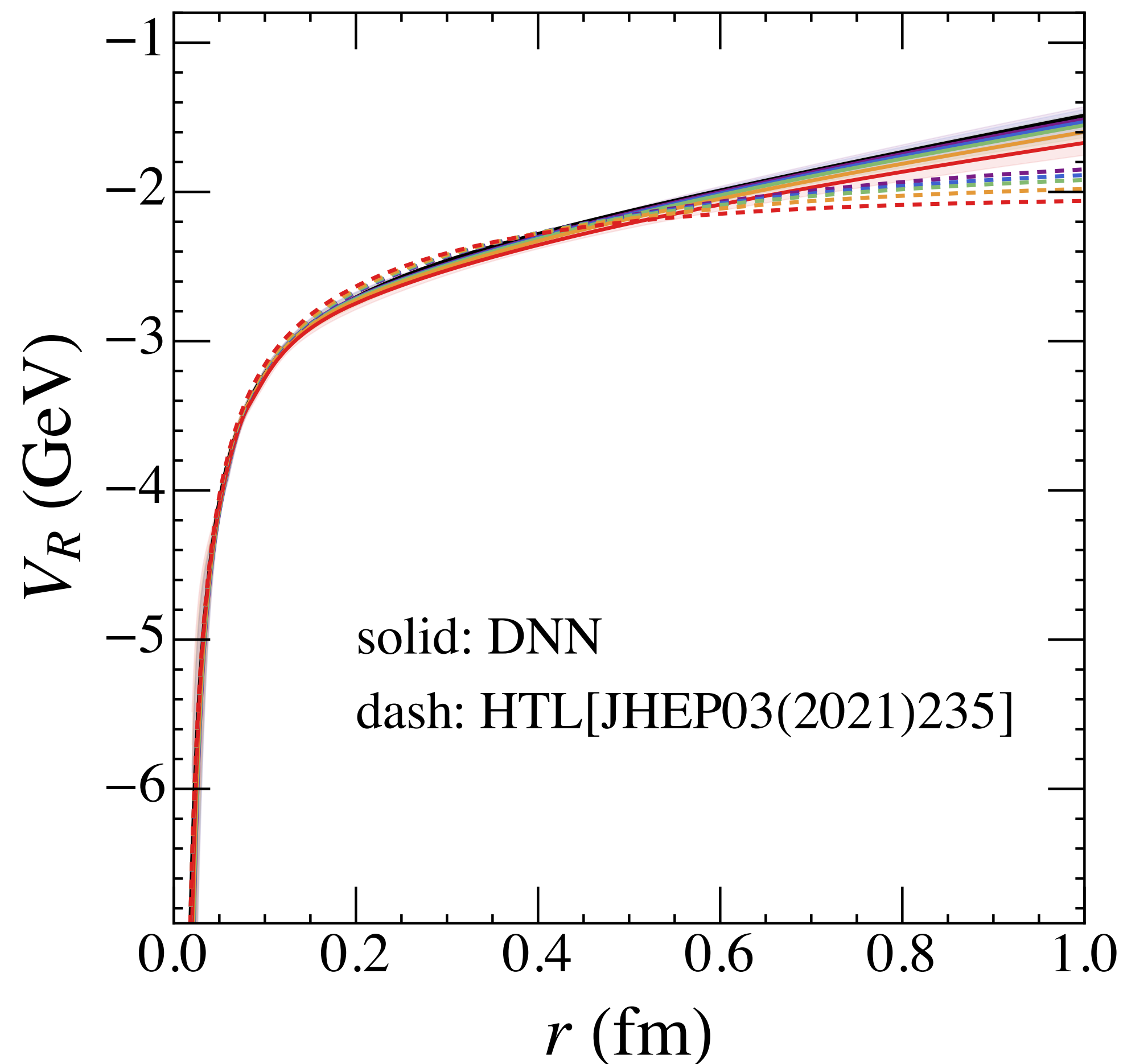
layers/depth of the network  
 $M^{(l)}$  -- number of neurons at layer ( $l$ )

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



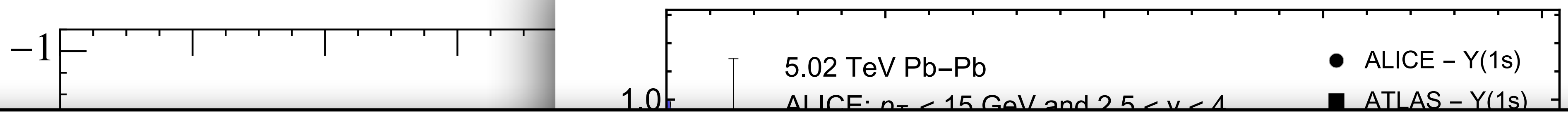
# What physics we have learned from $V_{\text{DNN}}(T, r)$ ?

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

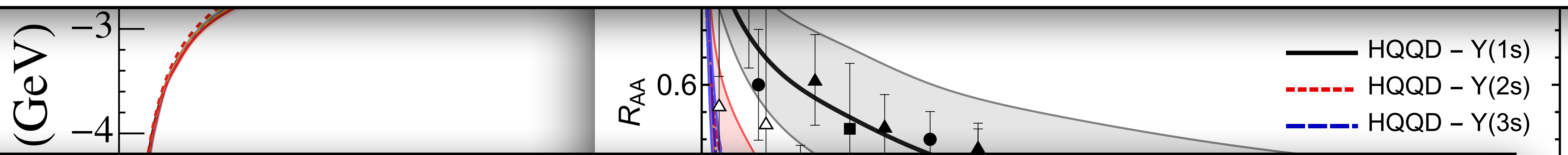


# What physics we have learned from $V_{DNN}(T, r)$ ?

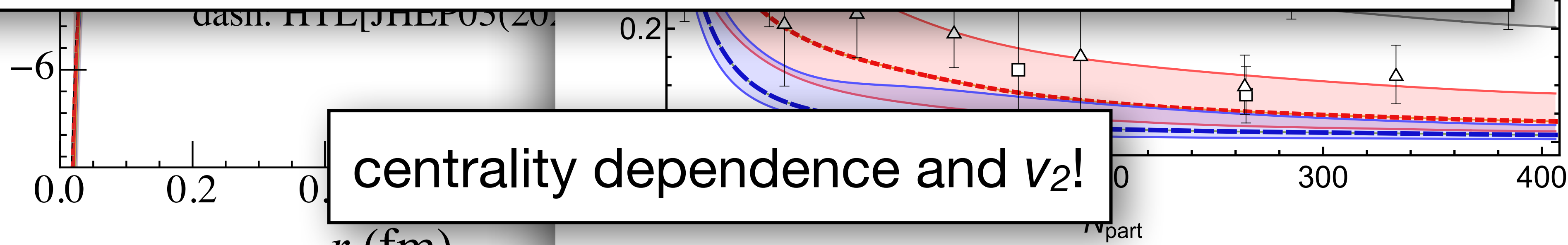
--- compare with HTL potential used in [1]



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



larger magnitude for  $V_I(r) \implies$  larger  $\Gamma \implies$  smaller  $R_{AA}$

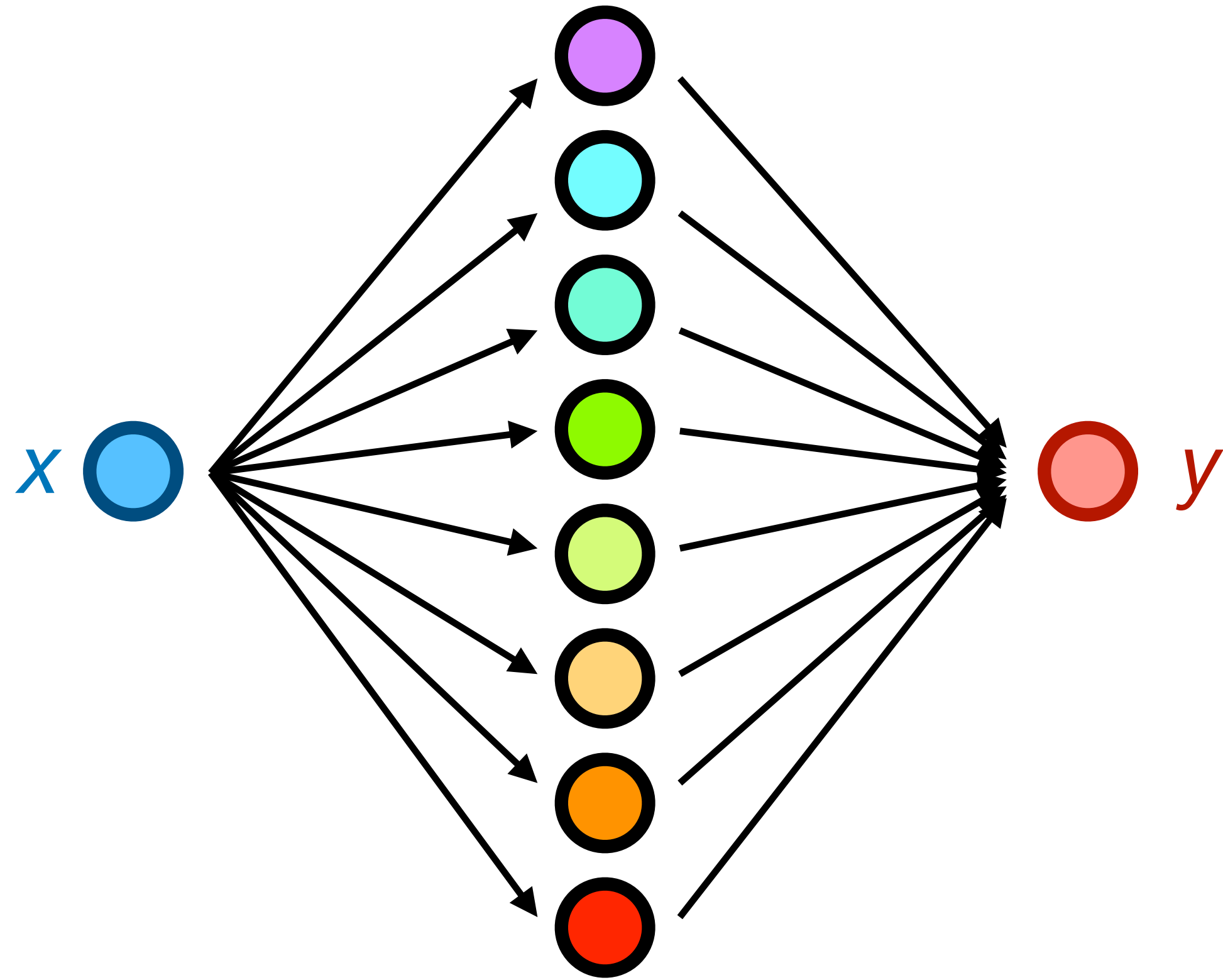


centrality dependence and  $v_2$ !

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

# How do Deep Neural Networks work?

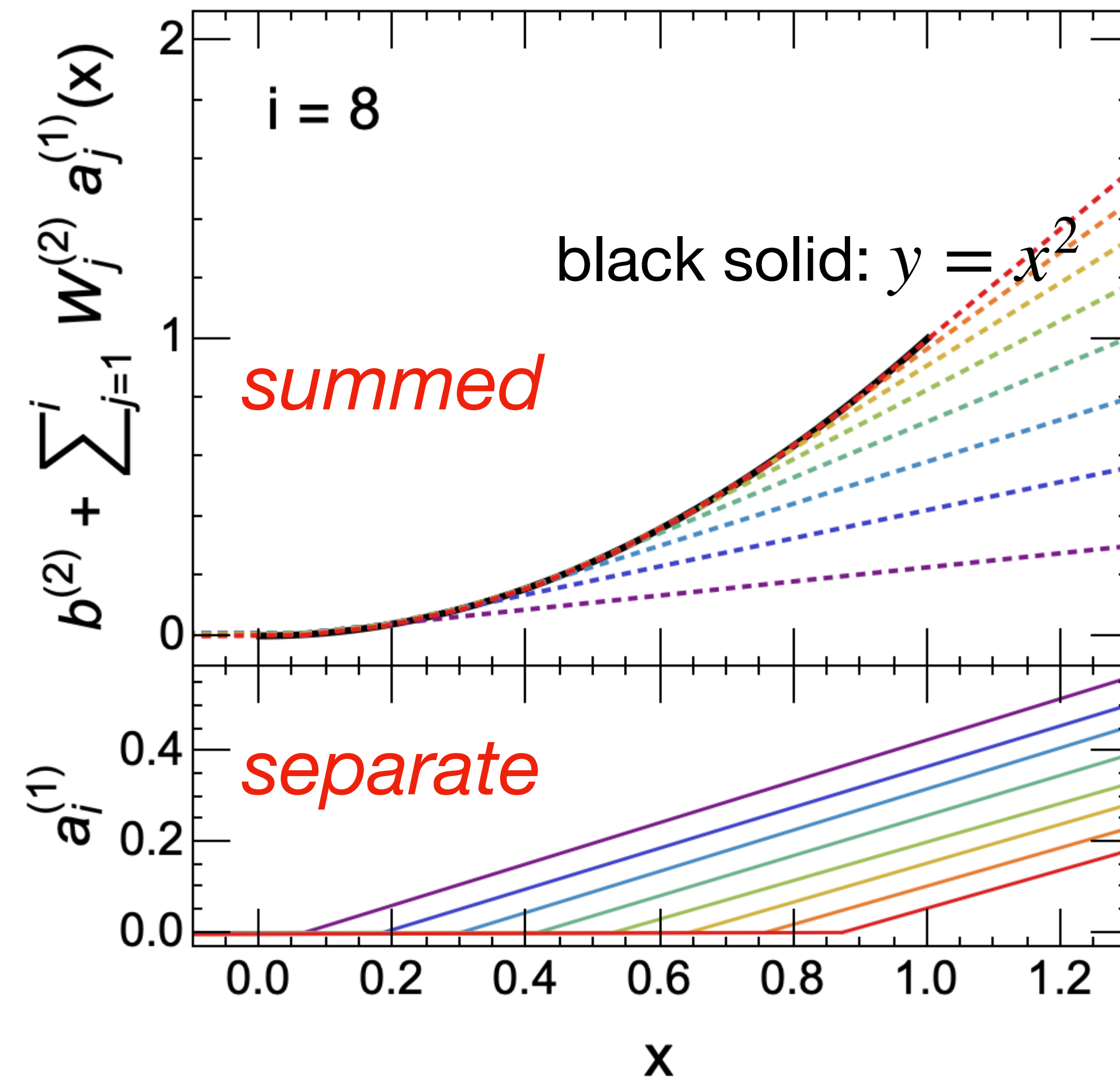
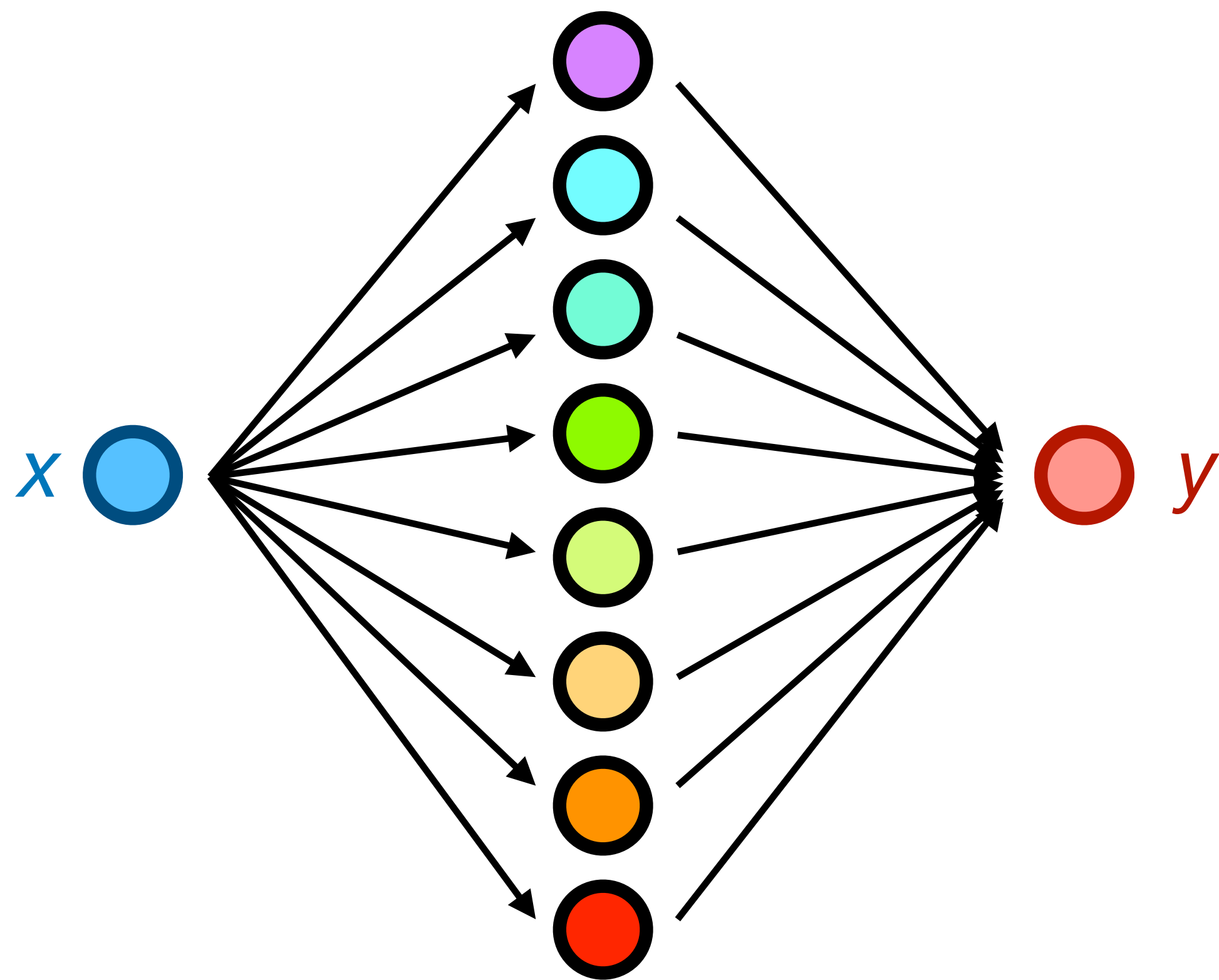
example: learn  $y = x^2$  for  $x \in [0,1]$





# How do Deep Neural Networks work?

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



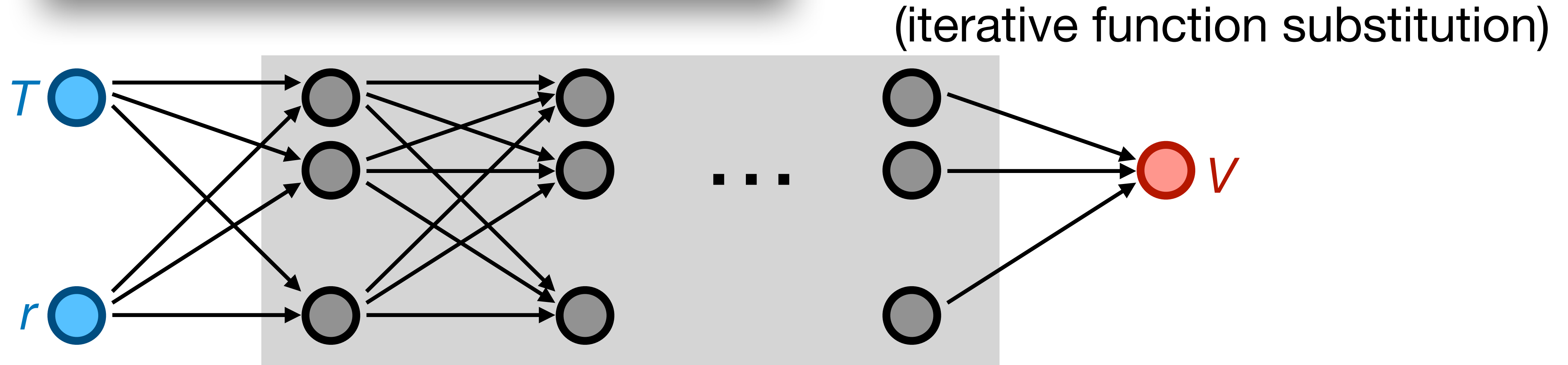
piecewise linear interpolation!!!

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

# What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

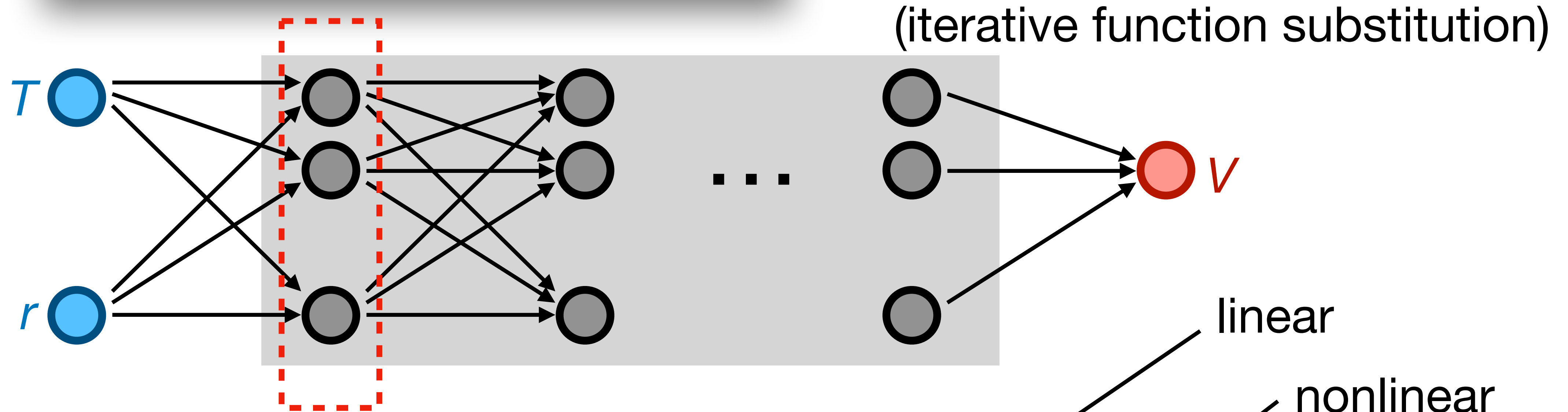
$$V(T, r) \approx V_{\text{DNN}}(T, r \mid \text{parameters})$$



# What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

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



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

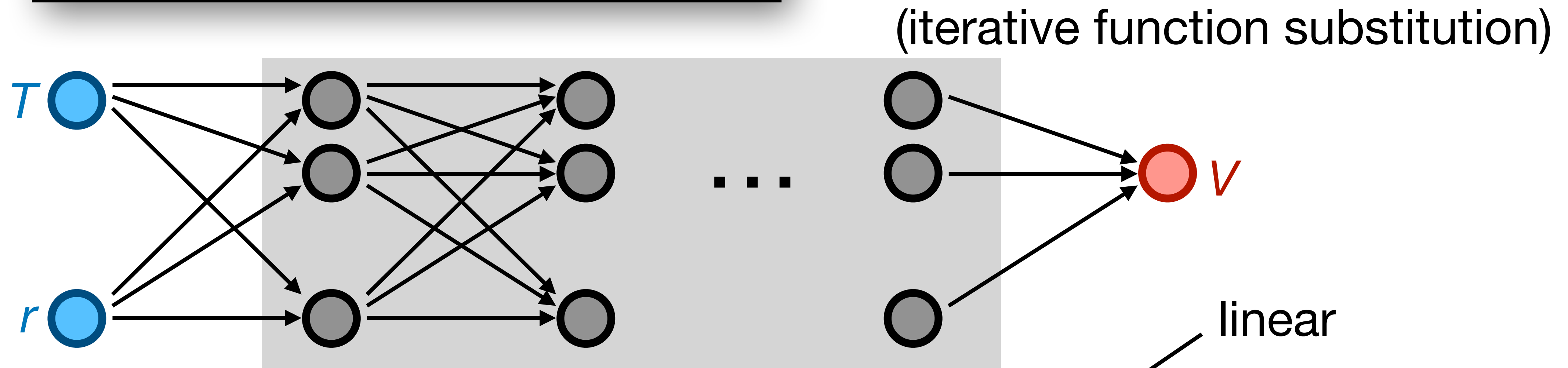
- At the first layer:

$$z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)} T + W_{i,2}^{(1)} r, \quad a_i^{(1)} = \sigma(z_i^{(1)})$$

# What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

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



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

- At the first layer:

$$z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)} T + W_{i,2}^{(1)} r,$$

$$a_i^{(1)} = \sigma(z_i^{(1)})$$

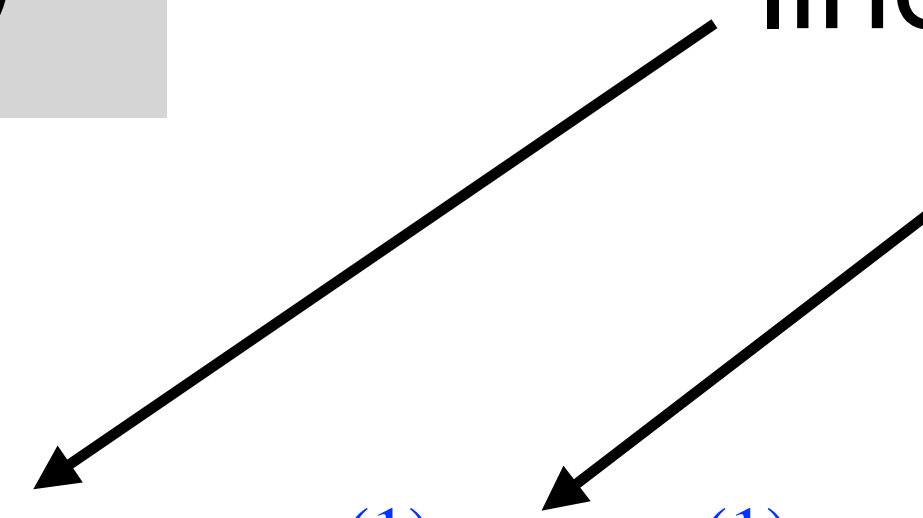
- At later layers:

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

$$a_i^{(l)} = \sigma(z_i^{(l)})$$

linear

nonlinear

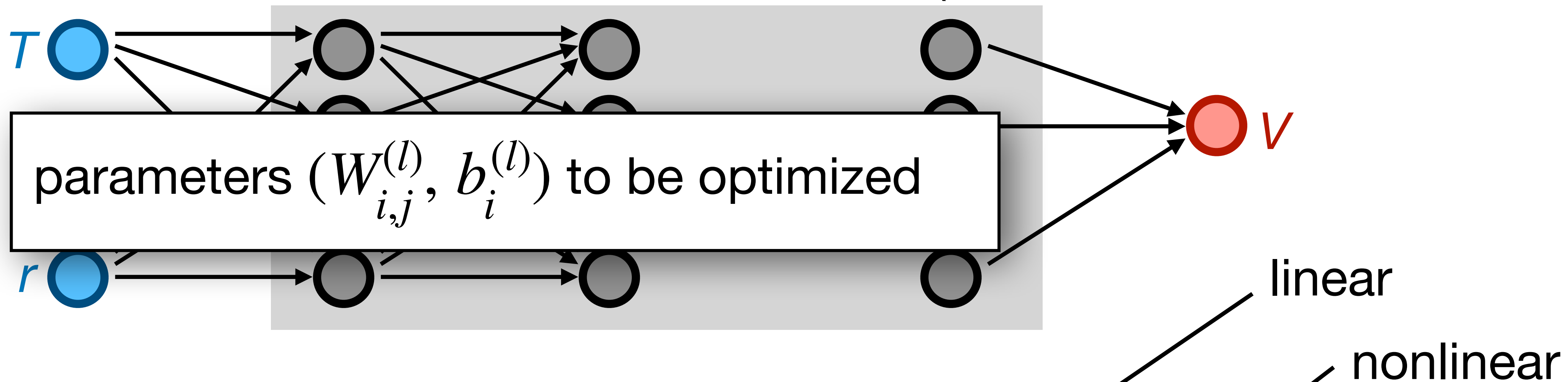


# What are Deep Neural Networks?

- DNN is a parameterization scheme to approximate continuous functions.

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

(iterative function substitution)



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

- At the first layer:
- At later layers:

$$z_i^{(1)} = b_i^{(1)} + W_{i,1}^{(1)} T + W_{i,2}^{(1)} r, \quad a_i^{(1)} = \sigma(z_i^{(1)})$$

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