Meson properties and symmetry emergence within deep neural network

Qiang LI

(liruo@nwpu.edu.cn)

Northwestern Polytechnical University

New Physics and Interdisciplinary Sciences

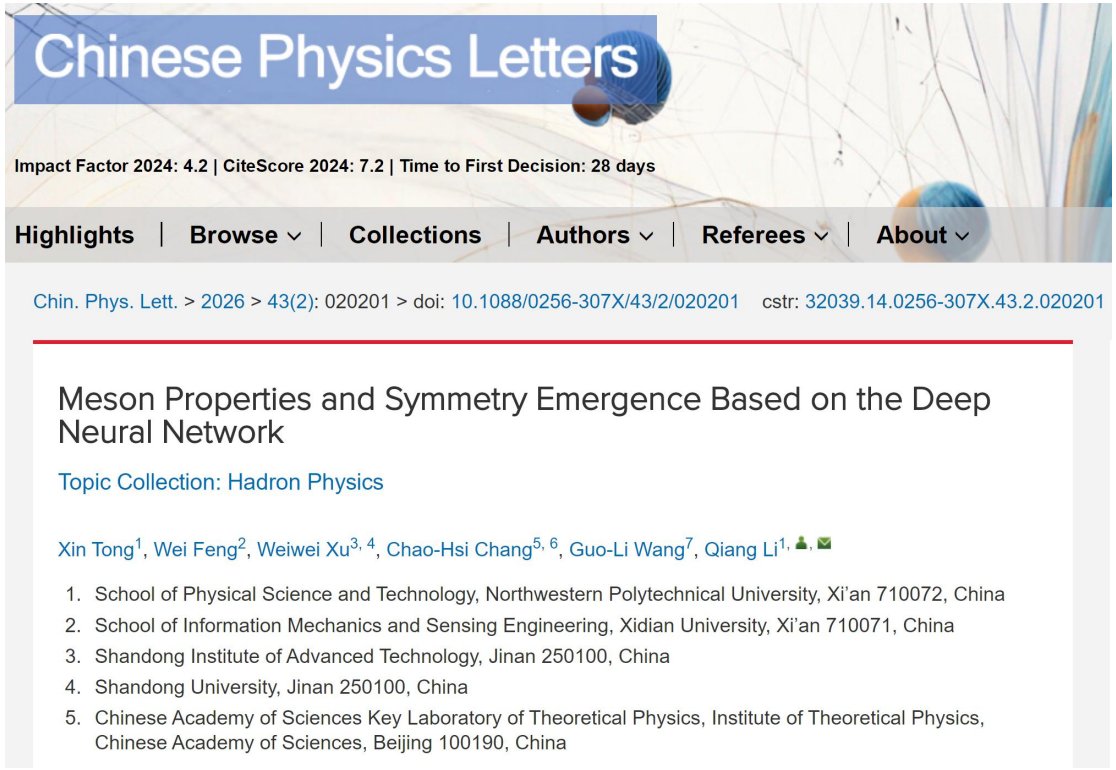
May. 16 2026, Nanjing

Outline

1. Motivation
2. Introduction to neural network
3. Hadron property predictions
4. Summary and outlook

0. About this talk

- Based on CPL 43(2026)020201, arXiv:2509.17093 **[hep-ph]**
- Coauthors: Xing Tong(NPU), Wei Feng(XDU), Weiwei Xu(SDU), Chao-Hsi Chang(ITP), Guo-Li Wang(HBU)
- A deep neural network model with the Transformer architecture is built to predict meson widths in the range of 10^(-14)-625 MeV based on meson quantum numbers and masses with the relative errors of about 1%.



Outline

1. Motivation
2. Introduction to neural network
3. Hadron property predictions
4. Summary and outlook

1. Motivation

- Quarks and gluons are bound by strong interactions to form hadrons. Due to quark confinement, hadron properties play key roles in understanding the strong interactions.
- Two main goals in Hadron Spectroscopy

  1. Obtain Hadron mass and decay behaviors based on Quantum number.
  2. Obtain Quantum number from mass and width.

1. Motivation

- Combining the potential model with bound states equation, we can obtain the hadron mass spectra, eg., the famous GI meson model.

$$H_0 \rightarrow \sum_{i=1}^{2} \left[m_i + \frac{p^2}{2m_i} \right]$$

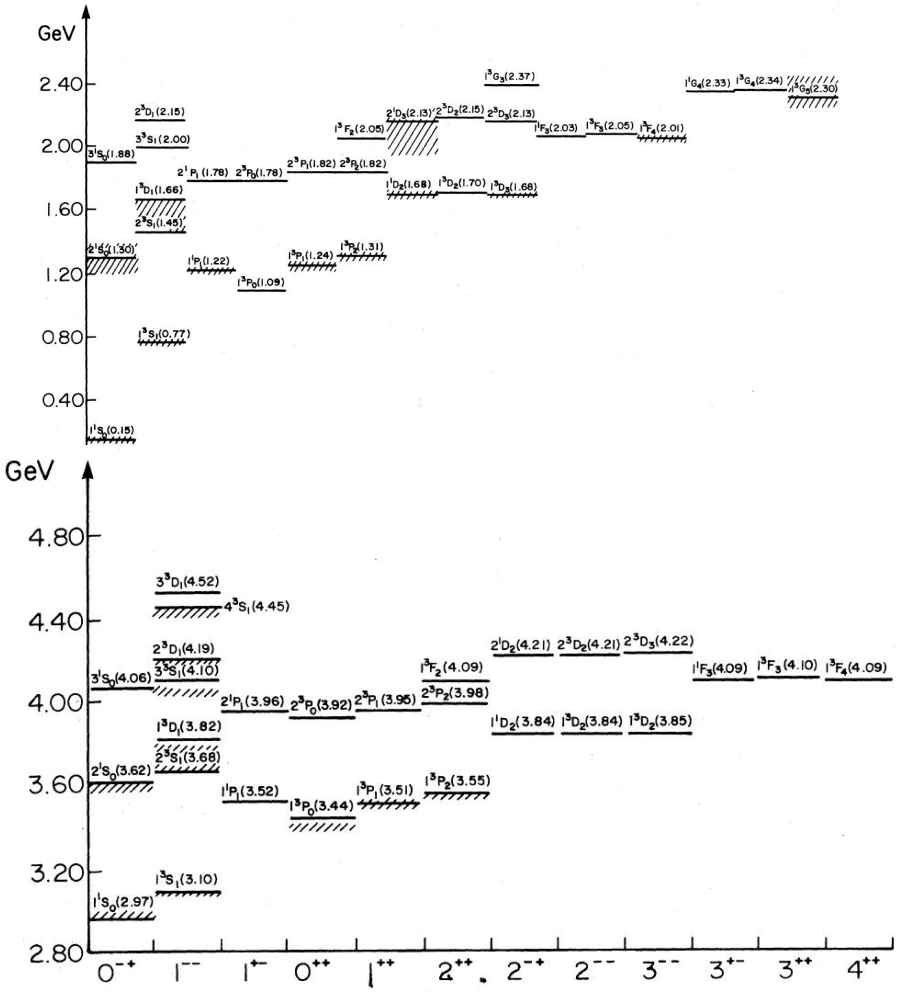
and

$$V_{ij}(\mathbf{p},\mathbf{r}) \rightarrow H_{ij}^{\text{conf}} + H_{ij}^{\text{hyp}} + H_{ij}^{\text{so}} + H_A$$

where

$$H_{ij}^{\text{conf}} = -\left[\frac{3}{4}c + \frac{3}{4}br - \frac{\alpha_s(r)}{r} \right] \mathbf{F}_i \cdot \mathbf{F}_j$$

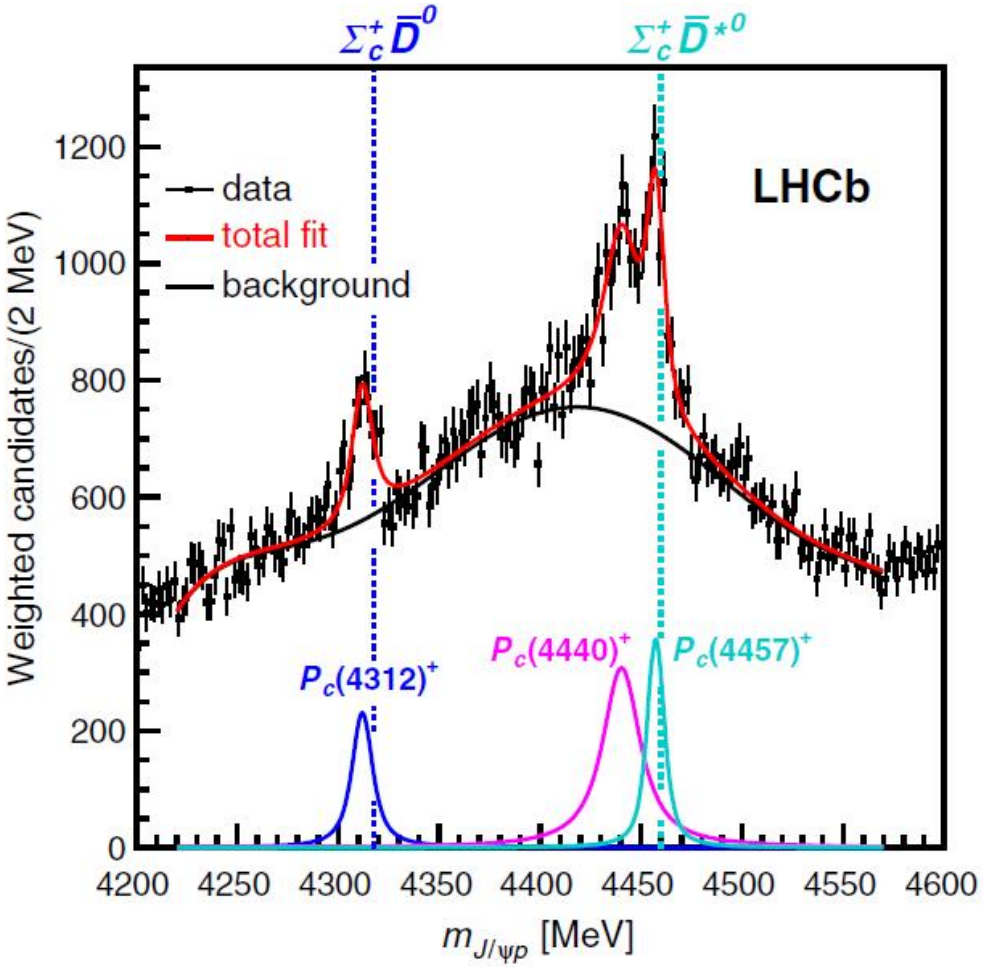
$$H_{ij}^{\text{hyp}} = -\frac{\alpha_s(r)}{m_i m_j} \left[\frac{8\pi}{3} \mathbf{S}_i \cdot \mathbf{S}_j \delta^3(\mathbf{r}) + \frac{1}{r^3} \left[\frac{3\mathbf{S}_i \cdot \mathbf{r}\mathbf{S}_j \cdot \mathbf{r}}{r^2} - \mathbf{S}_i \cdot \mathbf{S}_j \right] \right] \mathbf{F}_i \cdot \mathbf{F}_j$$



1. Motivation

- For the complication of QFT, quite difficult to obtain the decay width and lifetime within the traditional calculations based on the Standard Model theoretical calculation.

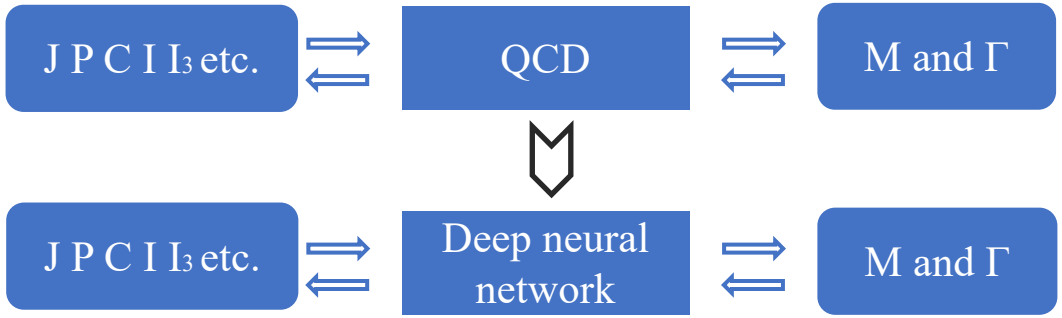
- The experiments usually detect a Hadron with its mass and Breit-Wigner width together, however, the quantum numbers are quite difficult to determine.

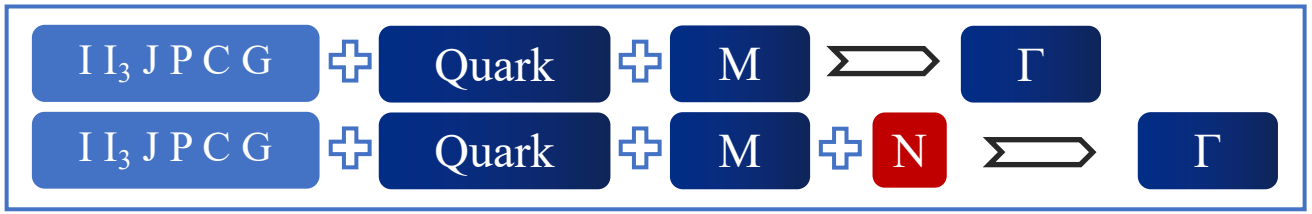


Goals of current work

1. Explore the generalization ability of deep neural network in hadron physics.

2. Predict the width Γ , or quantum numbers for undetermined hadrons.





Why neural network?

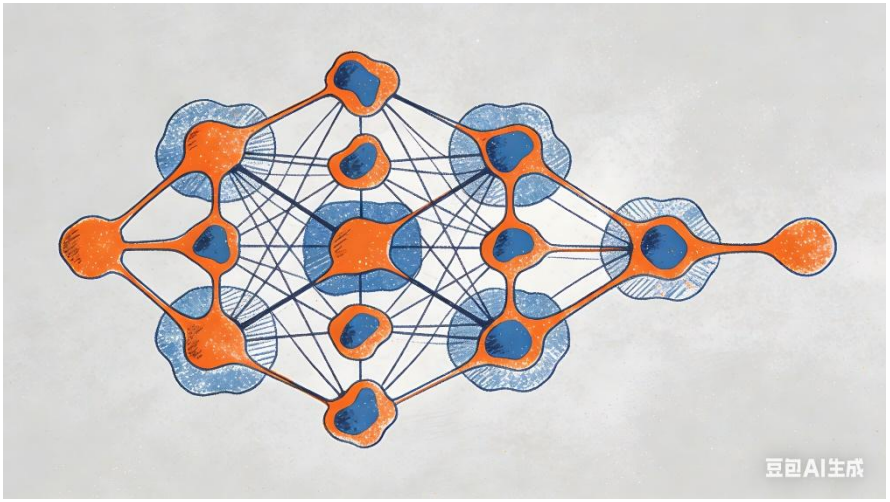
- The deep neural network provides new insights and powerful tools to deal with many traditional problems, AlphaGo, CV area, end-to-end autonomous driving.

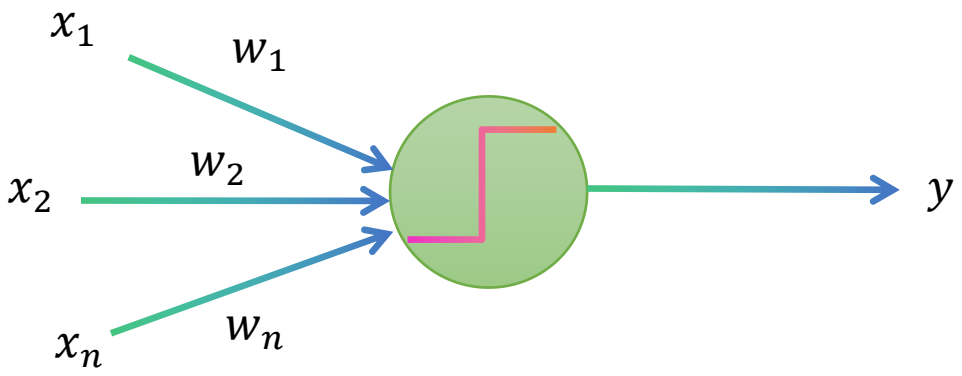
Outline

1. Motivation
2. Introduction to neural network
3. Hadron property predictions
4. Summary and outlook

Neurons of neural network

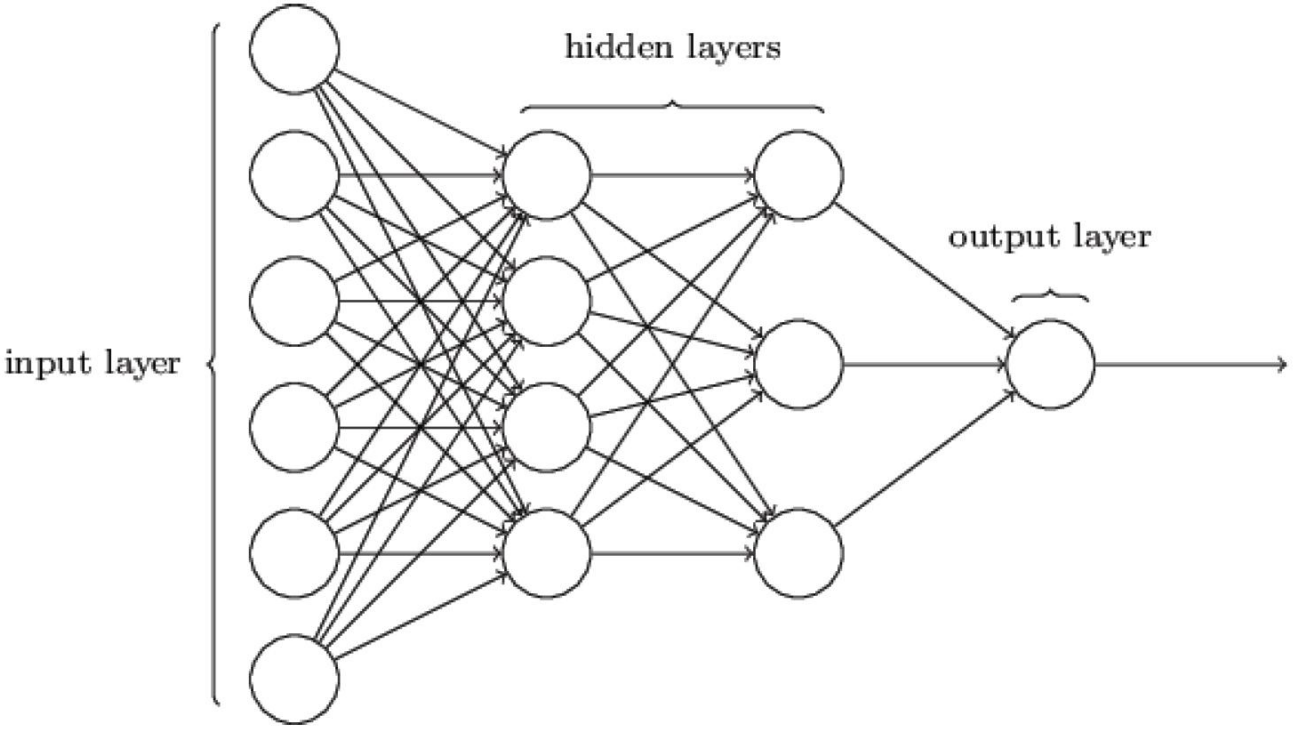
- Neural network consists of neurons.
- A neuron consists of several input x_1, x_2.....,x_n, every x_i corresponds to a weight w_i , an overal bias **b**, an activation function σ , and one output **y**.
- Activation function σ produces **nonlinear effects**. ReLU.
  $\sigma(z)=\mathrm{Max}(0,z)$
- Final output $y = \sigma(w \cdot x+b)$





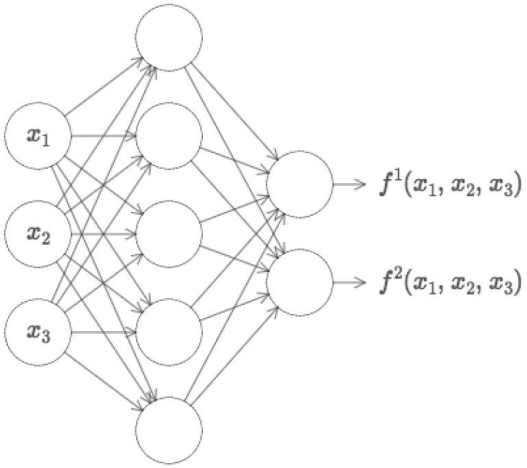
Architecture of neural network

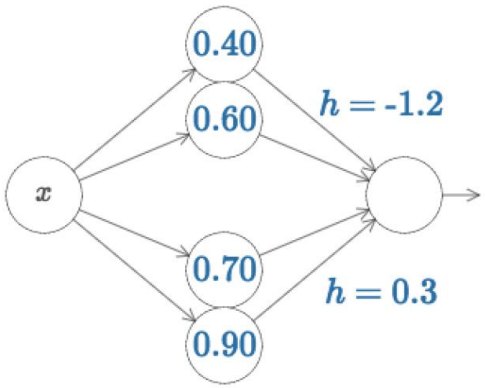
- A usual neural network consists of input layer, hidden layers, and output layer.
- eg., one handwritten image  (64×64) is 5 or not. Input layer x=(64×64)=4096.
- Output y in [0,1] represents the "probability" to be 5.

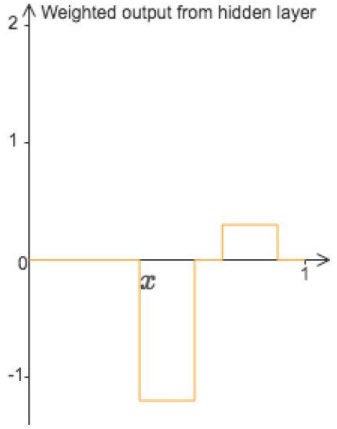


Universal Approximation Theorems

- One of the most striking facts: **a deep neural network with sufficient complexity can approximate any function to arbitrary accuracy**.
- Neural networks have a kind of universality which holds even if only using one hidden layer, while by increasing the number of hidden neurons we can improve the approximation.
- The class of functions which can be approximated in the way described are the continuous functions.







Loss function

- What we' d like is an algorithm which lets us find weights and biases, so that the output from the network approximates for all training inputs. To quantify how well we are achieving this goal we define a cost function.

$$C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$$

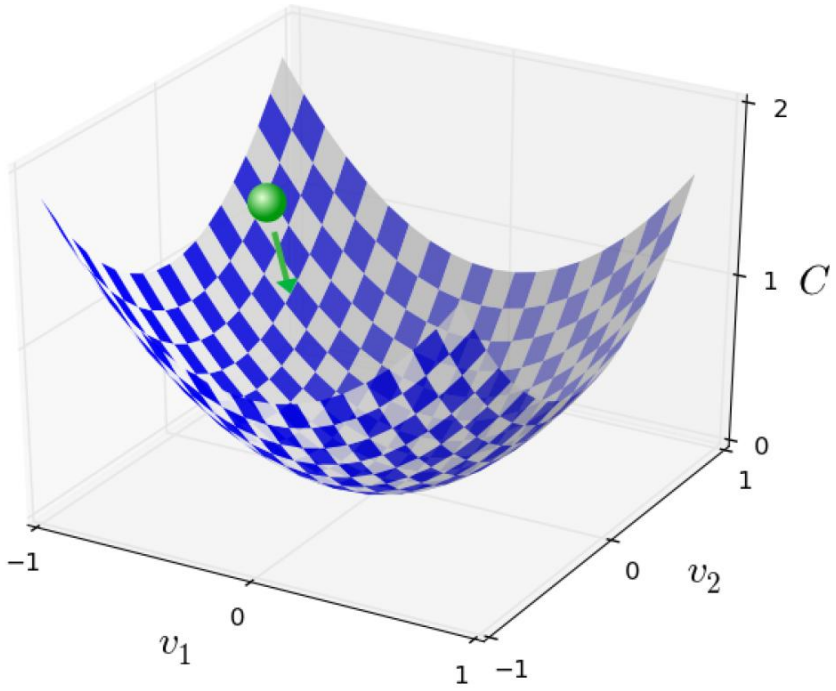
- n, total number of training inputs
- a, the vector of outputs from the network when x is input
- the sum is over all training inputs.
- C, quadratic cost function, or mean squared error(MSE)
- Goal: find appropriate w and b to minimize C(w,b).

Gradient descent

- By changing v(w,b) to minimize loss function C, $\Delta C \approx \nabla C \cdot \Delta v$
- Δv can be chosen to be in the direction of gradient $\Delta v = -\eta \nabla C$
- Then $\Delta C \approx -\eta \nabla C \cdot \nabla C = -\eta \|\nabla C\|^2 < 0$
- Strategy to update v: $v \to v' = v - \eta \nabla C.$
- Small value of η, learning rate.

$$w_k \to w_k' = w_k - \eta \frac{\partial C}{\partial w_k}$$

$$b_l \to b_l' = b_l - \eta \frac{\partial C}{\partial b_l}.$$



Backpropagation

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2 \qquad a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

- The complexity to calculate one derivative is O(N) for N parameters.
- The backpropagation algorithm was originally introduced in the 1970s.
- Its importance was not fully appreciated until a famous 1986 paper by David Rumelhart, Geoffrey Hinton, and Ronald Williams.
- Hinton etc. find: backpropagation works far faster than earlier approaches to learning, making it possible to use neural nets to solve problems which had previously been insoluble.
- Today, the backpropagation algorithm is the workhorse of learning in neural networks.

Backpropagation

- The goal of backpropagation is to compute $\partial C/\partial w$ and $\partial C/\partial b$ with respect to any weight w or bias b in the network.

$$\frac{\partial C}{\partial w_{ij}^l} = \left[\frac{\partial C}{\partial z_i^l}\right] \frac{\partial z_i^l}{\partial w_{ij}^l} = \delta_i^l a_j^{l-1}, \qquad \delta_i^l \equiv \frac{\partial C}{\partial z_i^l} = \left[\frac{\partial C}{\partial a_i^l} \frac{\partial a_i^l}{\partial z_i^l}\right] = \frac{\partial C}{\partial a_i^l} \sigma'(z_i^l),$$

$$\delta_i^l = \frac{\partial C}{\partial z_i^l} = \sum_j \left[\frac{\partial C}{\partial z_j^{l+1}}\right] \frac{\partial z_j^{l+1}}{\partial z_i^l} = \sum_j \delta_j^{l+1} \left[\frac{\partial z_j^{l+1}}{\partial a_i^l} \frac{\partial a_i^l}{\partial z_i^l}\right] = \sum_j \delta_j^{l+1} w_{ji}^{l+1} \sigma'(z_i^l).$$

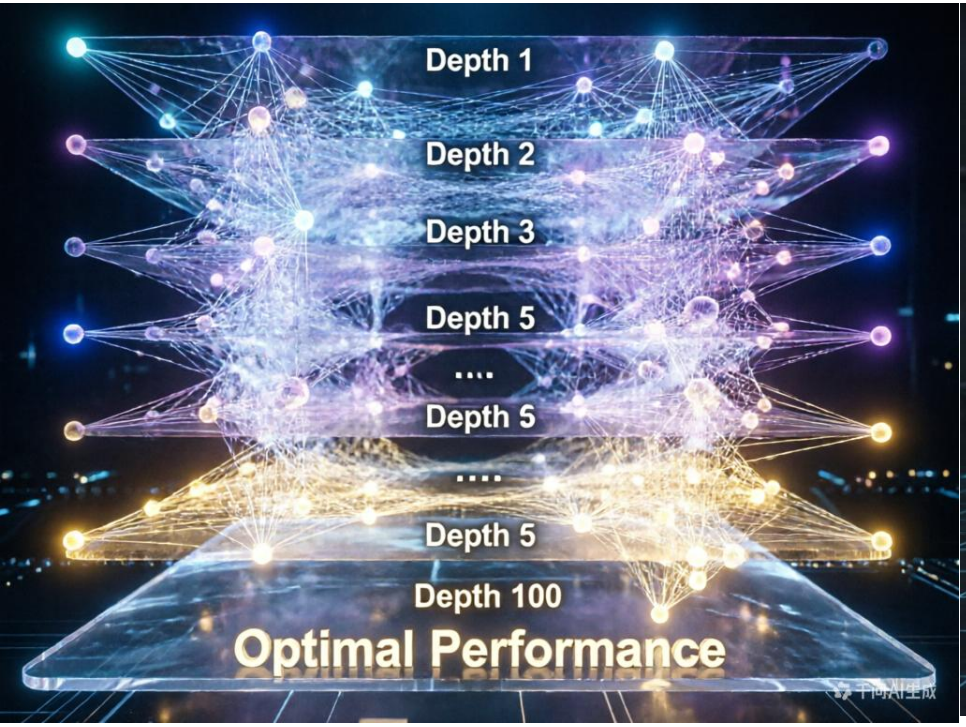
- We obtain the following recursive formular to calculate the error
- Derivatives to the Lth layer (the last one) is easy to calculate

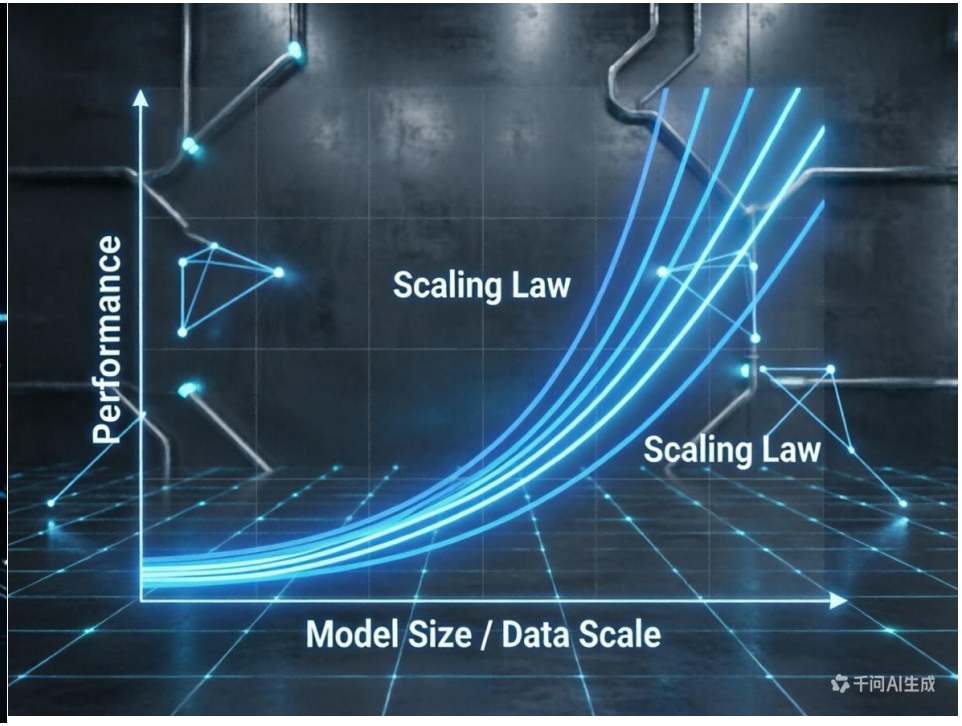
$$\delta_i^L = \frac{\partial C}{\partial a_i^L} \sigma'(z_i^L).$$

- Then the derivatives can be obtained as L→L-1→L-2⋯⋯1.

Deep learning

- Many other important topics in deep neural network are not involved, eg., ResNet, Convolutional Neural Network(CNN),⋯⋯
- Experiments show that Deeper neural networks generally yield better performance. Scaling law.





Transformer

- Self-attention mechanism is the core of Transformer.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

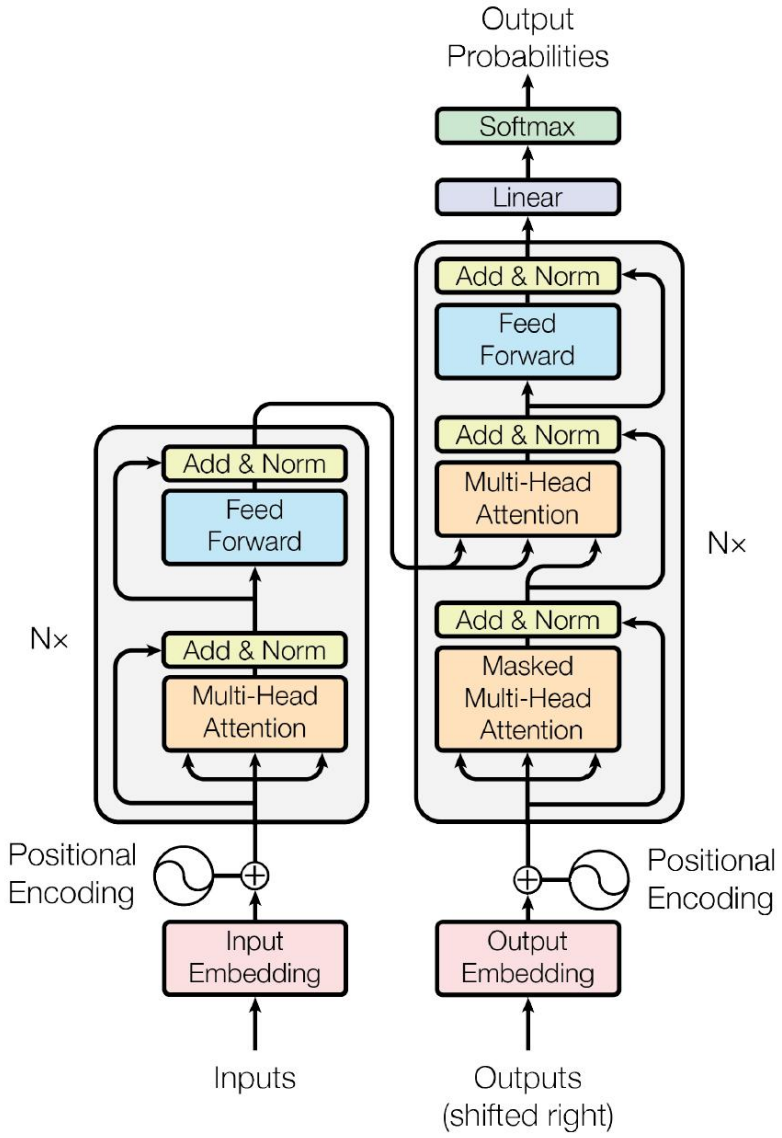
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.



Multi-headed self-attention of Transformer

1) This is our input sentence*

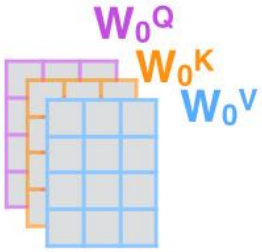
2) We embed each word*

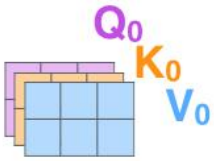
3) Split into 8 heads. We multiply X or R with weight matrices

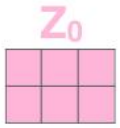
4) Calculate attention using the resulting Q/K/V matrices

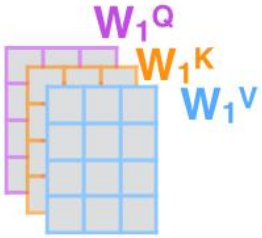
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

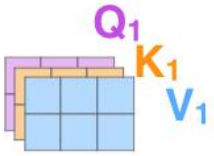
Thinking Machines

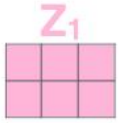


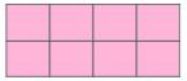




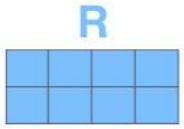


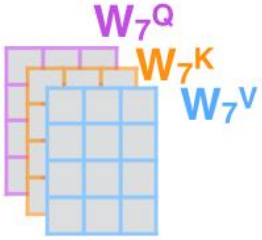


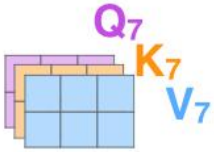




* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one







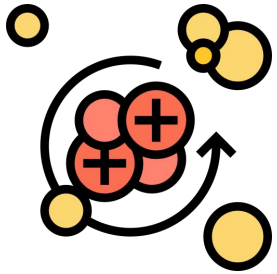
Outline

1. Motivation
2. Introduction to neural network
3. Hadron width predictions
4. Summary and outlook

Data

- Analysis framework: Anaconda->Python-> PyTorch.
- Hardware, NVIDIA GPU(RTX 3090). CUDA acceleration in torch.
- We use the hadron data from PDG, including 400+ mesons.

- PDG provides a Python API, which can be installed by "pip install pdg"
- Usage as



```
import pdg
api = pdg.connect()
pi_minus = api.get_particle_by_name('pi-')
print('MC ID  = ', pi_minus.mcid)
print('mass   = ', pi_minus.mass, 'GeV')
print('spin J = ', pi_minus.quantum_J)
```

Gaussian Monte Carlo data enhancement

- The most important challenge for this task is the lack of data.
- Implement the Gaussian Monte-Carlo data enhancement based on the uncertainties of experimental data.
- Produce the MC mass data ($\times 500$) for every hadron based on the Gaussian distribution with σ defined by the experimental error Δm

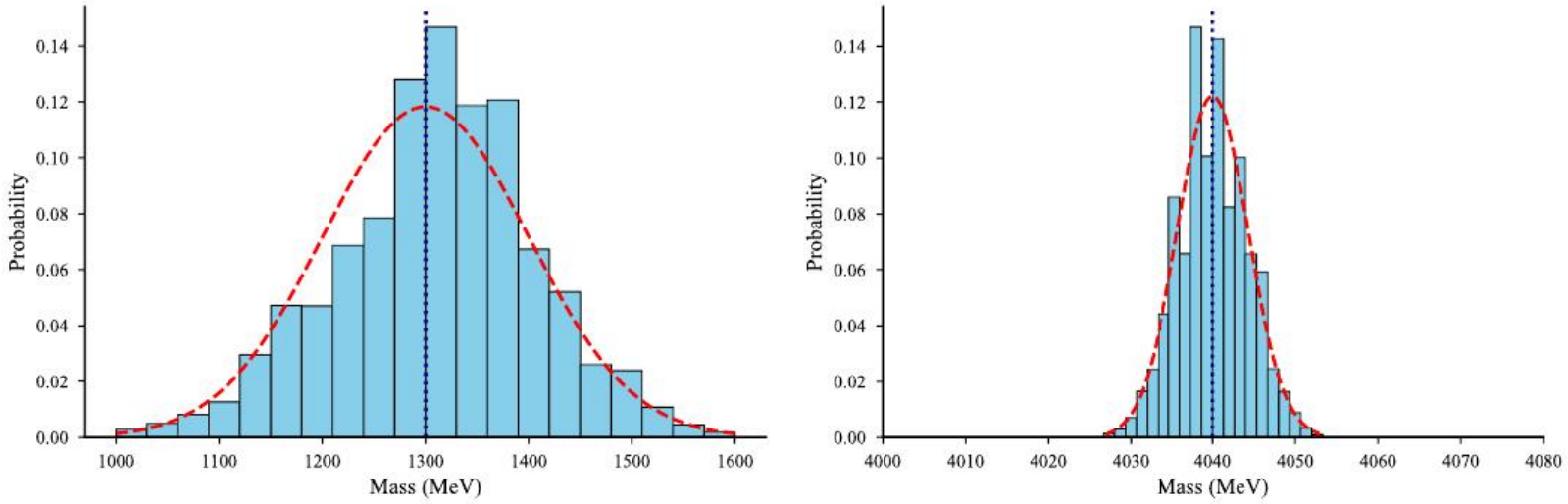


Fig. 1: The mass distribution of $\pi(1300)$ and $\psi(4040)$ after the Gaussian enhancement.

Input vectors and model flow

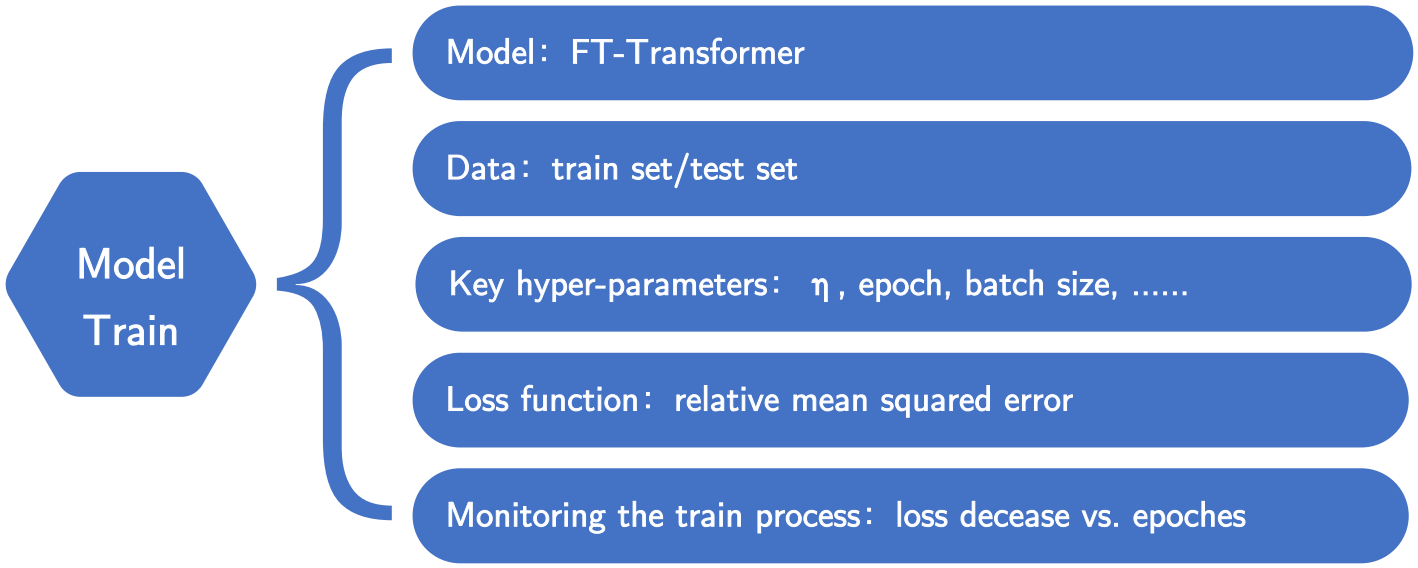
- Every hadron is finally encoded as an input vector.

$$\boldsymbol{v} = (J, P, C, G, I, I_3, u, \bar{u}, d, \bar{d}, s, \bar{s}, c, \bar{c}, b, \bar{b}, m, N).$$

$$\vec{v}_{\pi^+} = (1,0,0,1,0,0,0,0,0,0,1,1,-1,0,-1,\mathrm{N/A},0.14,1)$$

- An auxiliary feature N to represent the mass ordering, defined as

$$N = \frac{m}{m_\pi}$$



FT-Transformer data flow

- The Transformer model is able to process the input data in parallel and can also capture the long distance dependencies of data for its self-attention mechanism.

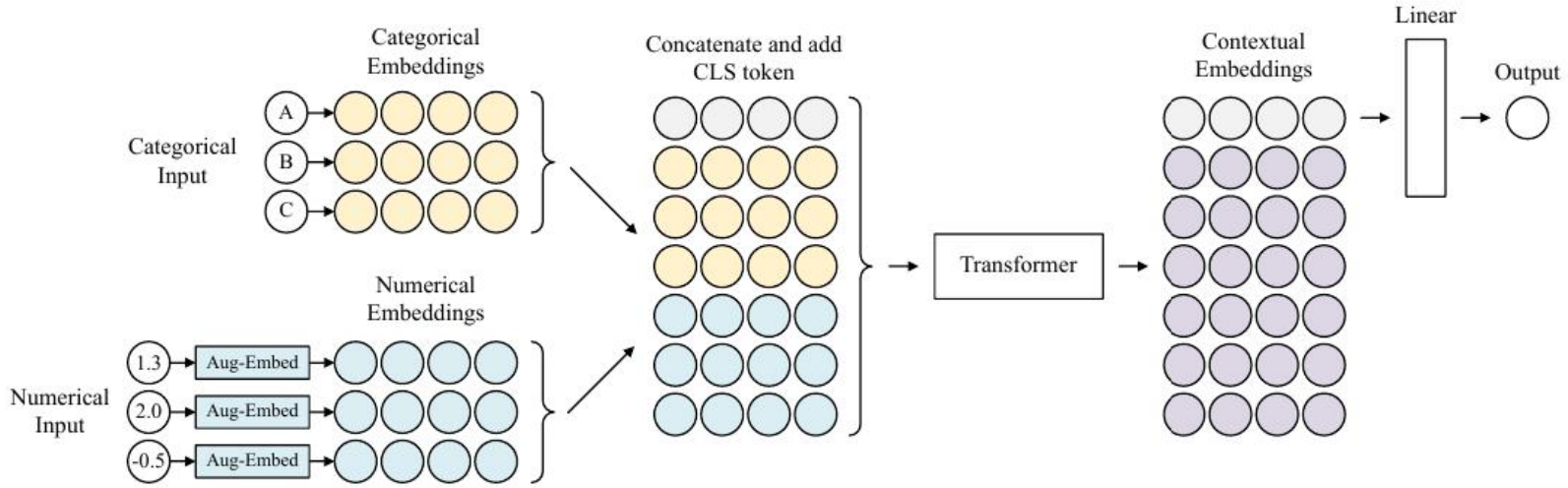


Fig. 2: The main parts of the FT-Transformer architecture. Both the numerical and categorical features are first embedded to d-dimensional vectors, which are then together with the [CLS] token fed into the typical Transformer encoder layer.

FT-Transformer backbone

- Perform a linear transformation on each continuous feature and then map it into the same d-dimensional space.

$$\boldsymbol{e}_{\text{num}} = x_{\text{num}}\boldsymbol{w} + \boldsymbol{b}, \qquad \boldsymbol{e}_{\text{fused}} = \text{Linear}[(\boldsymbol{e}_{\text{raw}}; \boldsymbol{e}_{\text{mask}})],$$

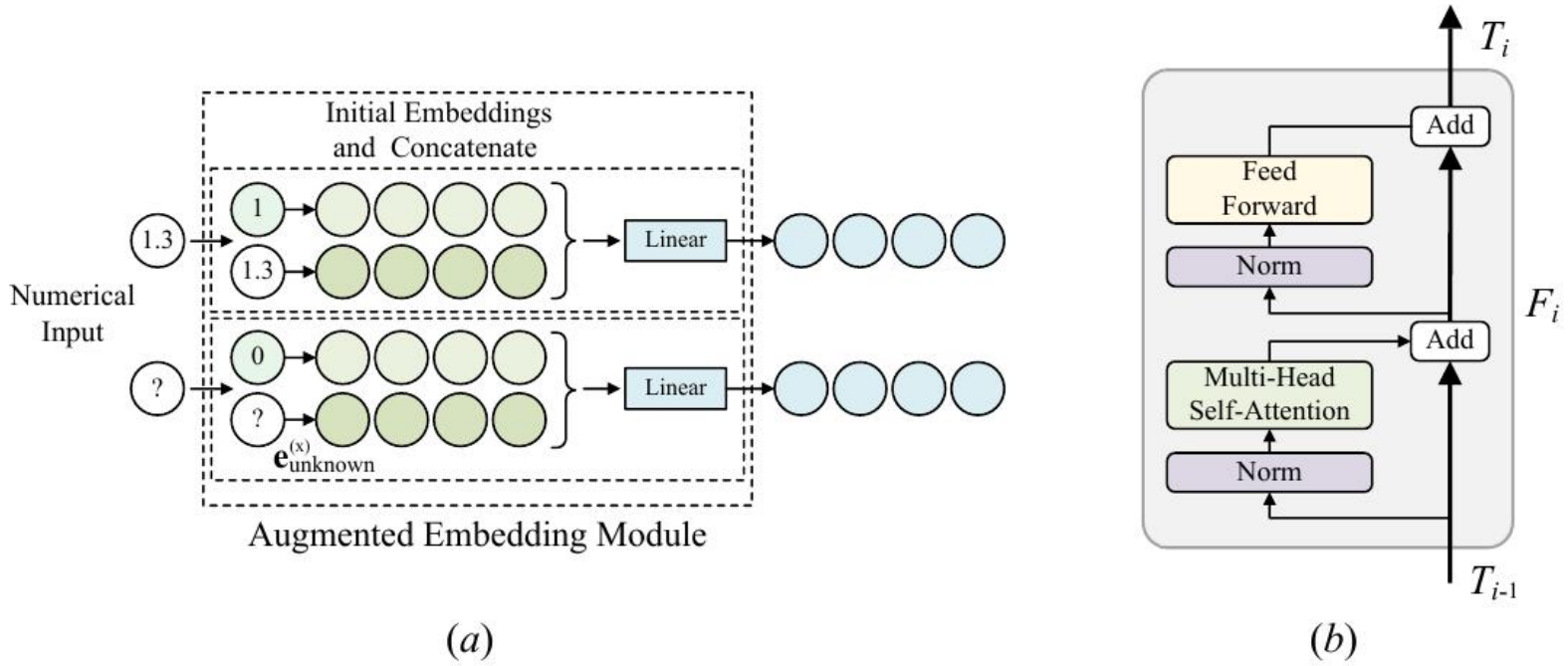


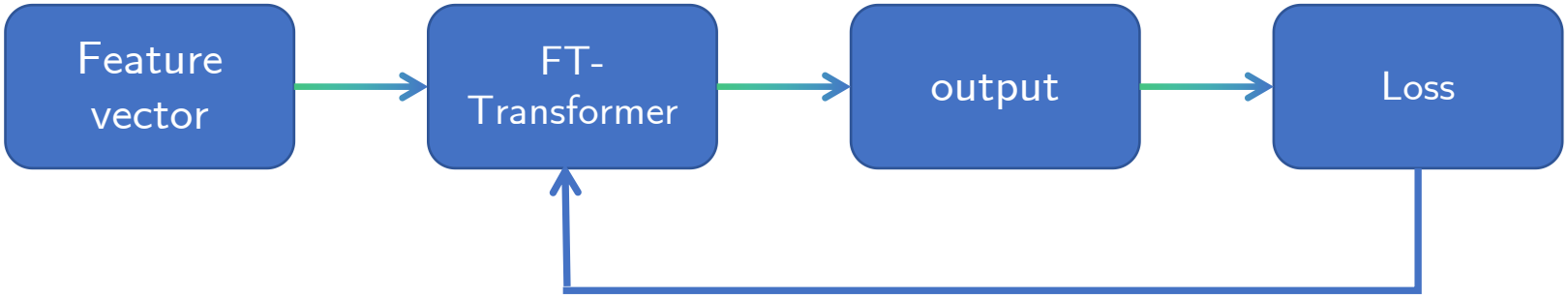
Fig. 3: (a) The Augmented Embedding module to deal with the numerical features; (b) one Transformer layer.

Loss function

- The only and ultimate aim of the model is to minimize the loss function, which is the least action principle of neural network.
- Here we use the relative mean square error. Every sample occupies the same weight in the model.

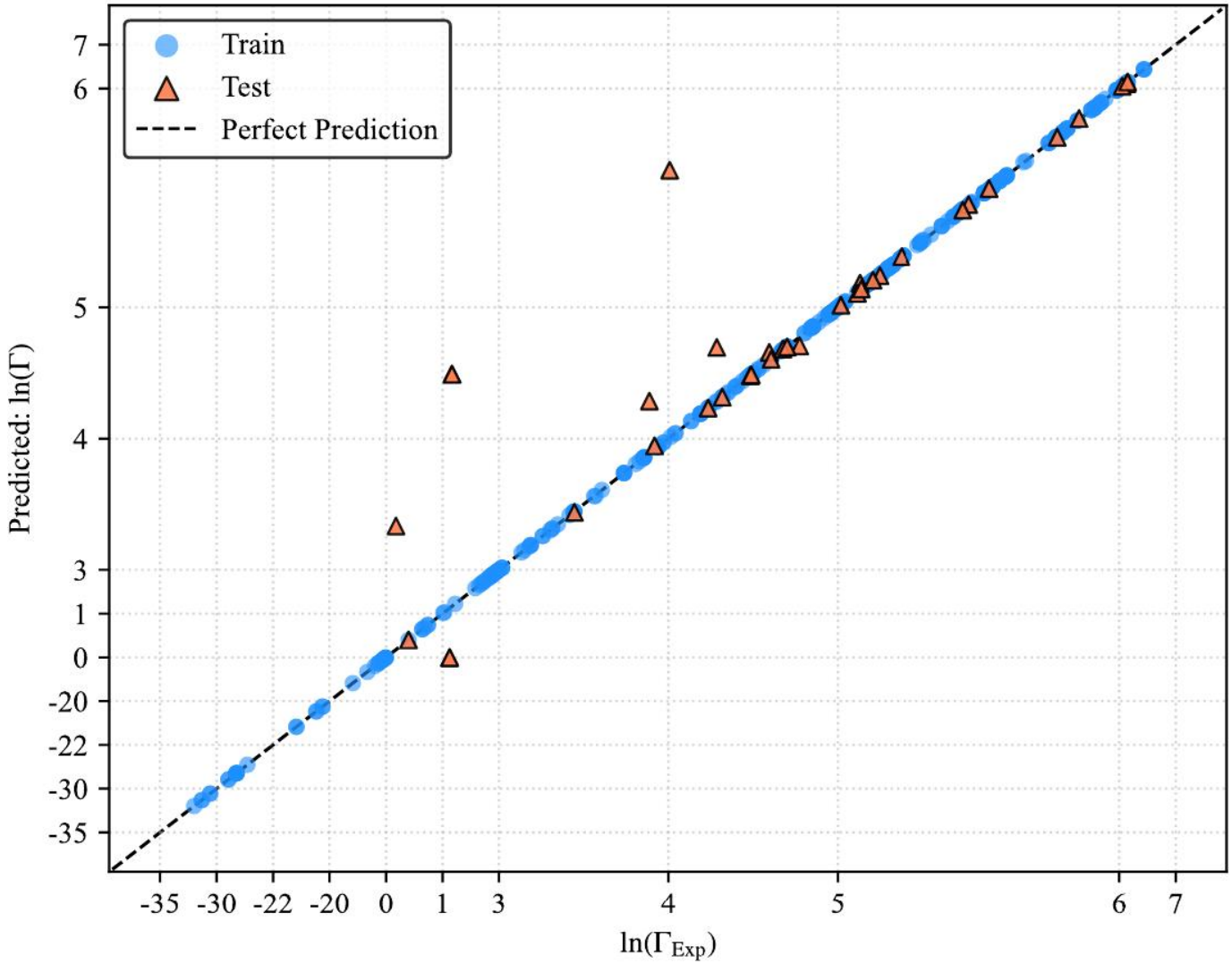
$$\text{Loss} = \epsilon^2 = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2,$$

- First ~100 epochs employs the LogCosh loss function.
- B_c is heavier then π but not more important.



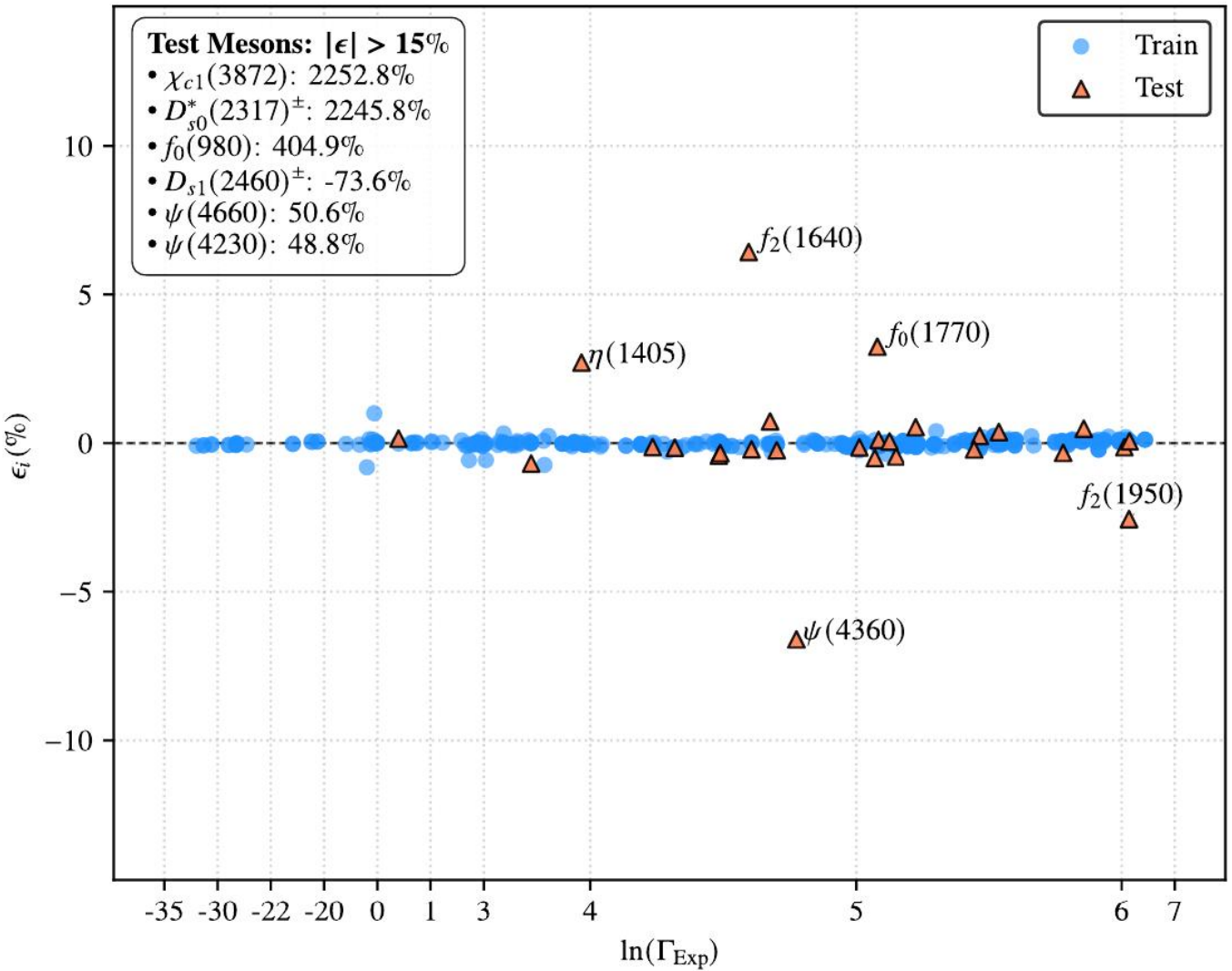
Overall performance

- Overall performance of the meson width predictions for all the data ranging from $10^{-14}\,\mathrm{MeV}$ to $625\,\mathrm{MeV}$

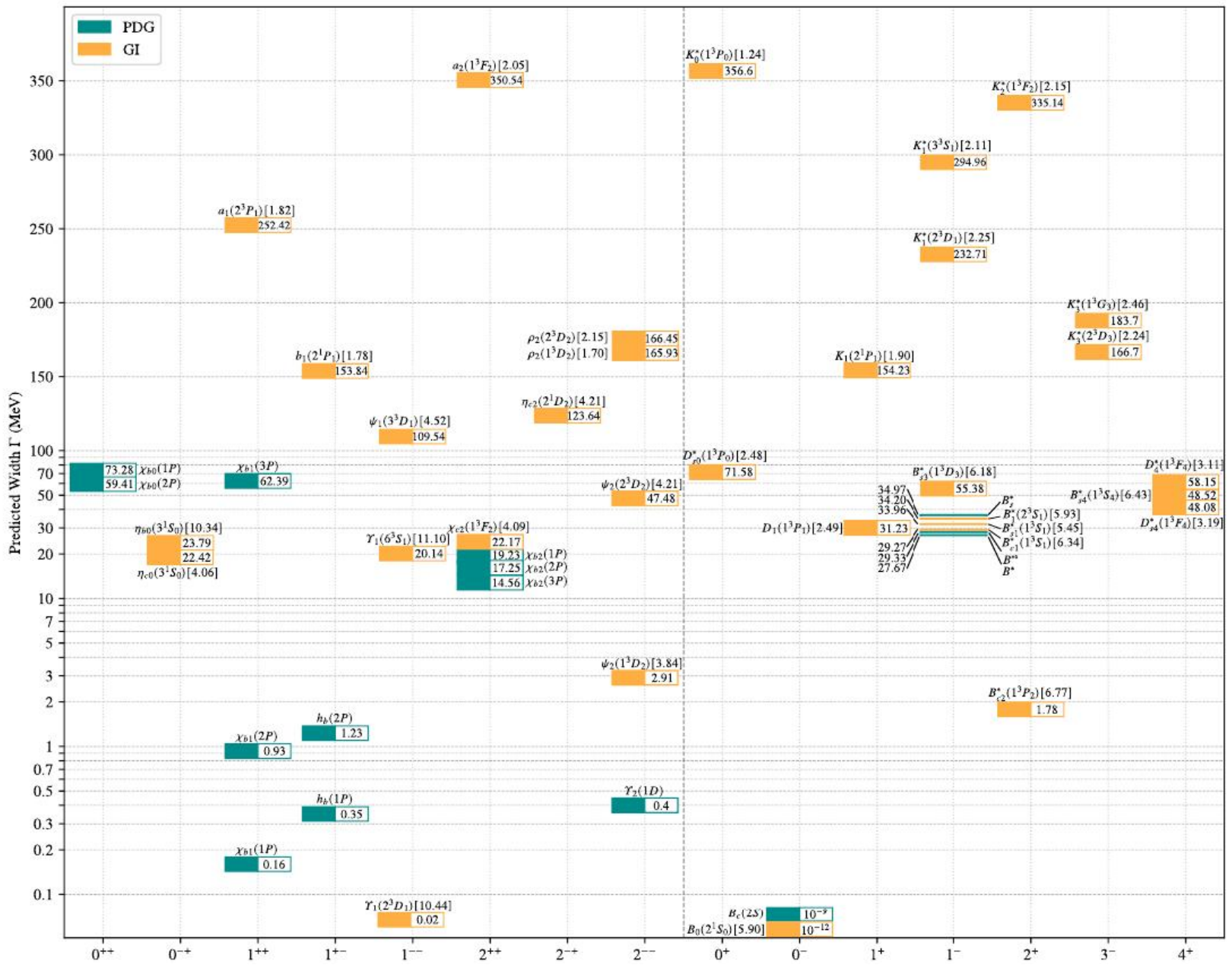


Relative error

- After excluding several states with $\epsilon_i > 48\%$, the obtained relative errors are $\epsilon = 0.12\%, 2.0\%,$ and 0.54% for training, test and all data.



Predictions

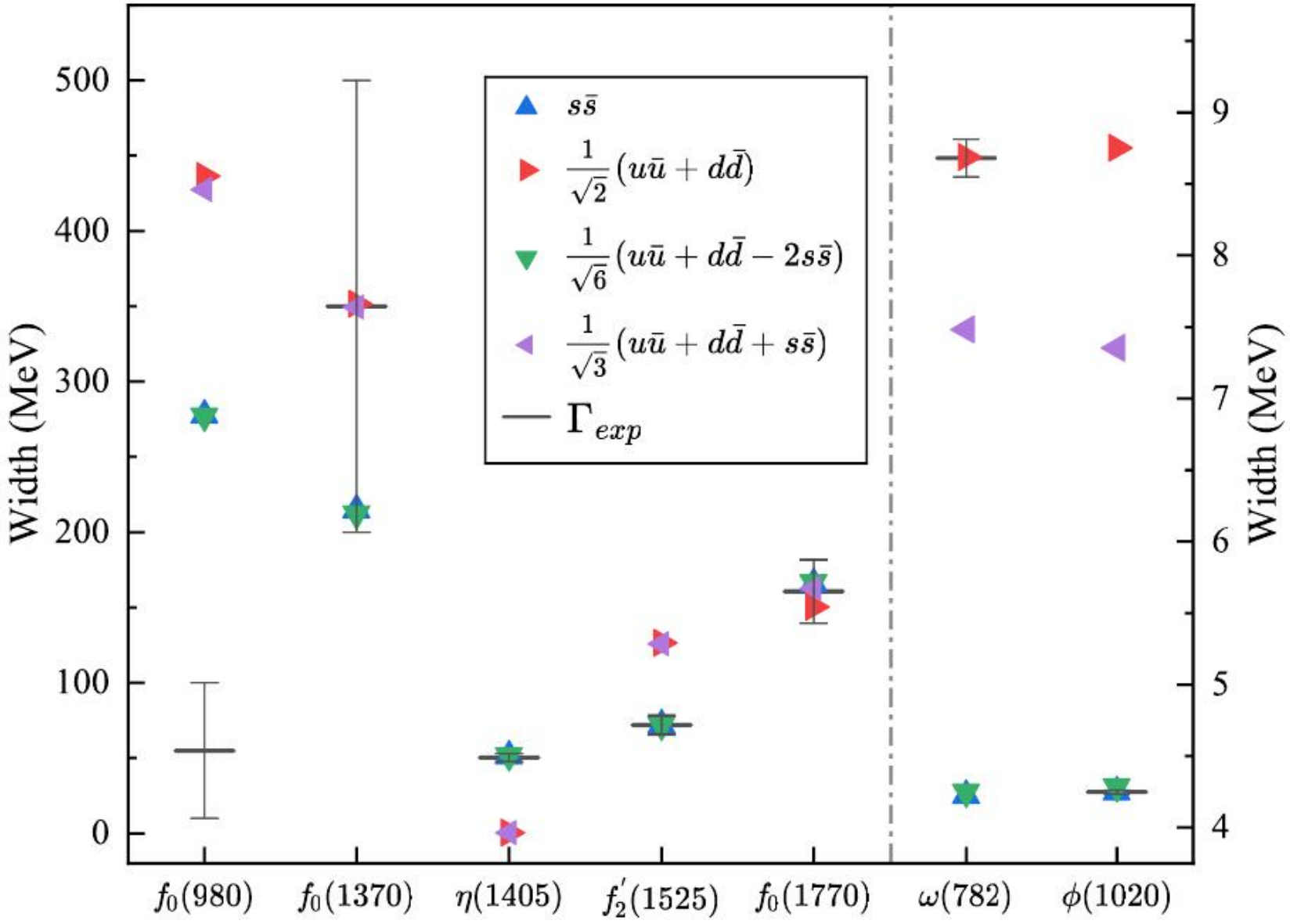


Predictions

Tab. III: The model's predicted widths and favored quantum numbers for the undetermined mesons.

States	$I^{(G)}(J^{P(C)})$	$\Gamma_{\rm Exp}$	$I^{(G)}(J^{P(C)})$	Quark	$\Gamma_{\rm Pre1}$	$I^{(G)}(J^{P(C)})$	Quark	$\Gamma_{\rm Pre2}$
$X(1750)$	$?^-(1^{--})$	120 ± 10	$0^-(1^{--})$	$s\bar{s}$	149.1	$0^-(1^{--})$	$u\bar{u}s\bar{s}$	149.1
$K(1630)$	$\frac{1}{2}(?^?)$	16^{+19}_{-16}	$\frac{1}{2}(?^-)$	$d\bar{s}$	16.4	-	-	-
$K(3100)$	$?^?(?^?)$	42 ± 16	$\frac{1}{2}(?^+)$	$u\bar{s}$	41.5	$\frac{3}{2}(2^+)$	$sd\bar{u}\bar{u}$	38.5
$D^*(2640)$	$\frac{1}{2}(?^?)$	< 15	$\frac{1}{2}(?^-)$	$c\bar{d}$	16.9	-	-	-
$D(3000)$	$\frac{1}{2}(?^?)$	190 ± 80	$\frac{1}{2}(0^+)$	$c\bar{u}$	186.1	-	-	-
$D_{sJ}(3040)$	$0(?^?)$	240 ± 60	$0(?^-)$	$c\bar{s}$	177.3	-	-	-
$X(3940)$	$?^?(?^{??})$	43^{+28}_{-18}	$0^-(3^{+-})$	$c\bar{c}$	36.1	$0^+(2^{++})$	$c\bar{c}$	35.4
$X(4160)$	$?^?(?^{??})$	136^{+60}_{-35}	$0^+(2^{-+})$	$u\bar{u}c\bar{c}$	138.5	$0^+(2^{-+})$	$d\bar{d}c\bar{c}$	160.9
$X(4630)$	$0^+(?^{?+})$	170^{+140}_{-80}	$0^+(2^{-+})$	$s\bar{s}c\bar{c}$	167.3	$0^+(1^{-+})$	$s\bar{s}c\bar{c}$	152.9
$B_J^*(5732)$	$?(?^?)$	128 ± 18	$\frac{1}{2}(?^-)$	$d\bar{b}$	126.5	-	-	-
$B_J(5840)^+$	$\frac{1}{2}(?^?)$	220 ± 80	$\frac{1}{2}(?^-)$	$u\bar{b}$	190.7	-	-	-
$B_J(5840)^0$	$\frac{1}{2}(?^?)$	130 ± 40	$\frac{1}{2}(?^-)$	$d\bar{b}$	123.9	-	-	-
$B_J(5970)^+$	$\frac{1}{2}(?^?)$	62 ± 20	$\frac{1}{2}(?^-)$	$u\bar{b}$	74.5	-	-	-
$B_J(5970)^0$	$\frac{1}{2}(?^?)$	81 ± 12	$\frac{1}{2}(?^-)$	$d\bar{b}$	90.5	-	-	-
$B_{sJ}^*(5850)$	$?(?^?)$	47 ± 22	$0(1^-)$	$s\bar{b}$	36.9	-	-	-
$B_{sJ}(6063)$	$0(?^?)$	26 ± 6	$0(2^-)$	$s\bar{b}$	22.9	-	-	-
$B_{sJ}(6114)$	$0(?^?)$	66 ± 28	$0(4^-)$	$s\bar{b}$	82.7	-	-	-
$T_{c\bar{c}}(4050)$	$1^-(?^{?+})$	82^{+50}_{-28}	$1^-(3^{-+})$	$ud\bar{c}\bar{c}$	82.8	$1^-(2^{-+})$	$ud\bar{c}\bar{c}$	105.0
$T_{c\bar{c}}(4055)$	$1^+(?^{?-})$	45 ± 13	$1^+(3^{+-})$	$ud\bar{c}\bar{c}$	49.1	$1^+(2^{+-})$	$ud\bar{c}\bar{c}$	35.9
$T_{c\bar{c}}(4100)$	$1^-(?^{?+})$	150^{+80}_{-70}	$1^-(1^{-+})$	$ud\bar{c}\bar{c}$	164.3	$1^-(0^{++})$	$ud\bar{c}\bar{c}$	230.4
$T_{c\bar{c}}(4250)$	$1^-(?^{?+})$	180^{+320}_{-70}	$1^-(1^{-+})$	$ud\bar{c}\bar{c}$	175.2	$1^-(0^{-+})$	$ud\bar{c}\bar{c}$	172.0
$T_{cc\bar{c}\bar{c}}(6600)$	$0^+(?^{??})$	440^{+230}_{-200}[64]	$0^+(2^{++})$	$cc\bar{c}\bar{c}$	96.2	$0^+(1^{+-})$	$cc\bar{c}\bar{c}$	120.3
$T_{cc\bar{c}\bar{c}}(6900)$	$0^+(?^{??})$	137 ± 21	$0^+(2^{++})$	$cc\bar{c}\bar{c}$	99.3	$0^+(1^{+-})$	$cc\bar{c}\bar{c}$	122.4
$T_{cc\bar{c}\bar{c}}(7100)$	$0^+(?^{??})$	97^{+40}_{-29}[64]	$0^+(2^{++})$	$cc\bar{c}\bar{c}$	100.6	$0^+(0^{++})$	$cc\bar{c}\bar{c}$	80.4
$T_{b\bar{s}}(5568)$	$1(?^?)$	19^{+9}_{-7}	$1(3^+)$	$ud\bar{s}\bar{b}$	23.2	$1(1^+/2^+)$	$ud\bar{s}\bar{b}$	25.7

Dependence on the quark contents



Charge conjugation symmetry

- The CPT theorem is one of the deepest results of QFT, which states that **the combined operation of time reversal, charge conjugation, and parity in any order is an exact symmetry of any interaction**.
- An important experimental implication of the CPT theorem claims that every particle must have precisely the same mass and lifetime(width) as its antiparticle.
- Tested by the K^0-$\bar{K}^0$ mass difference fraction being $< 6 \times 10^{-19}$

$$\boldsymbol{q}_{B_c^+} = (0,0,0,0,0,0,1,0,0,1) \rightleftharpoons (0,0,0,0,0,0,0,1,1,0) = \boldsymbol{q}_{B_c^-}.$$

Charge conjugation symmetry

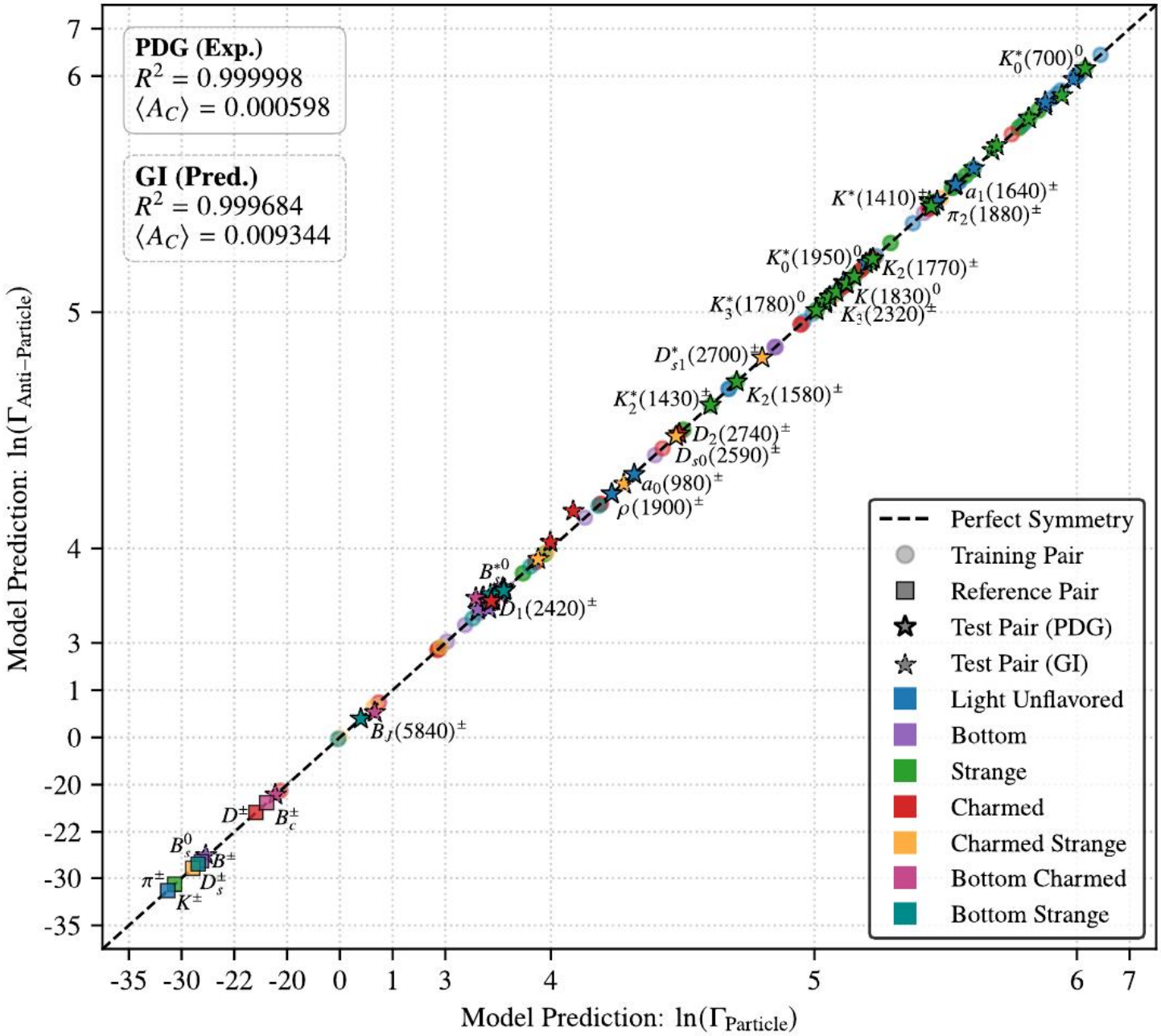
- In order to reflect the degree of this C symmetry violation for the neural network model, we introduce the violation coefficient as

$$A_{Ci} = \frac{\Gamma_i - \Gamma_{\bar{i}}}{\Gamma_i + \Gamma_{\bar{i}}},$$

- The root mean square violation coefficient to denote the overall violation of charge conjugation symmetry of the model for all mesons,

$$A_C = \sqrt{\langle A_{Ci}^2 \rangle} = \sqrt{\frac{1}{n} \sum A_{Ci}^2} = 0.06\%.$$

Charge conjugation symmetry



Isospin symmetry

- Within the dynamics of the Standard Model,

  strong >> electromagnetic >> weak interaction.

- Meson widths are then determined by the corresponding strong decay behaviors except for those can only electroweakly decay.

- The strong interaction is approximately invariant under the isospin transformation, and then the widths of mesons that can undergo strong decay also exhibit approximate isospin symmetry.

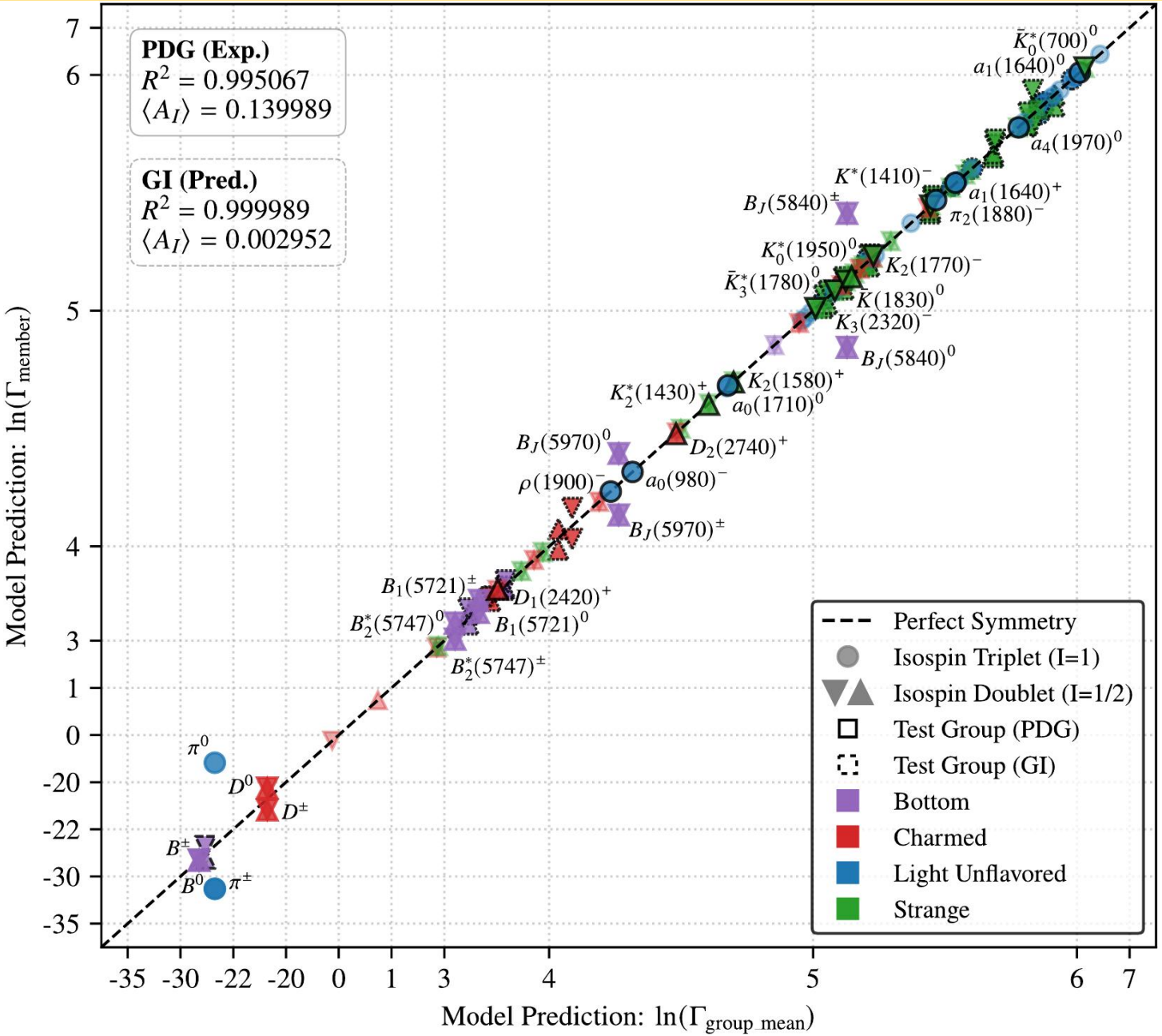
- The violation coefficient for isospin symmetry can be introduced similarly

$$A_{Ii} = \frac{\Gamma_i - \langle \Gamma_i \rangle_I}{\Gamma_i + \langle \Gamma_i \rangle_I},$$

- The overall violation of isospin symmetry of the model

$$A_I = \sqrt{\langle A_{Ii}^2 \rangle} = \sqrt{\frac{1}{n} \sum A_{Ii}^2} = 14.0\%,$$

Isospin symmetry



Outline

1. Motivation
2. Introduction to neural network
3. Hadron width predictions
4. Summary and outlook

Summary and outlook

- Introduce one hadron encoding and training scheme to describe and predict the hadron width and quantum numbers by using the neural network.
- Introduce a data enhancement technology to prepare enough data for NN.
- As a powerful tool and Universal Approximaton function, neural network can provide feasible solutions for many actual problems in many aspects.

I. Cross validation to introduce systematic errors.

II. Introduce the confidence interval.

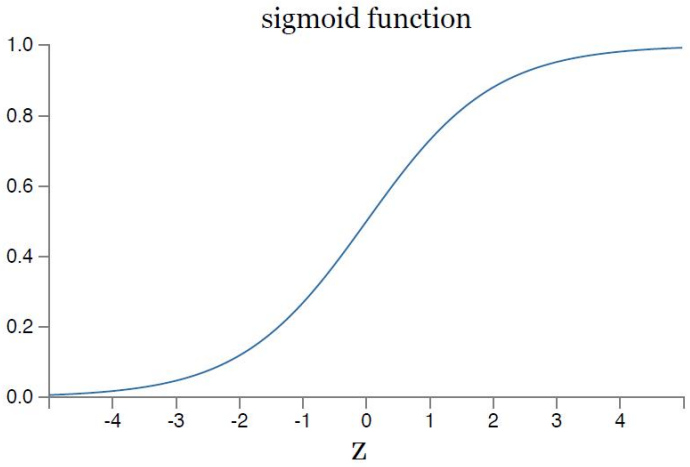
III. Using only the BCSUD quantum numbers.

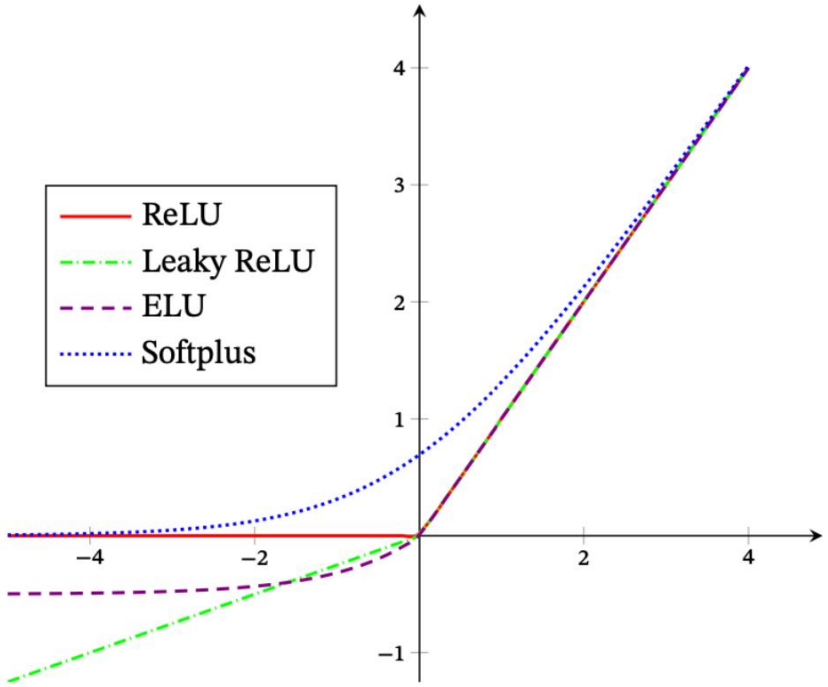
IV. .....

谢　谢

Activation function

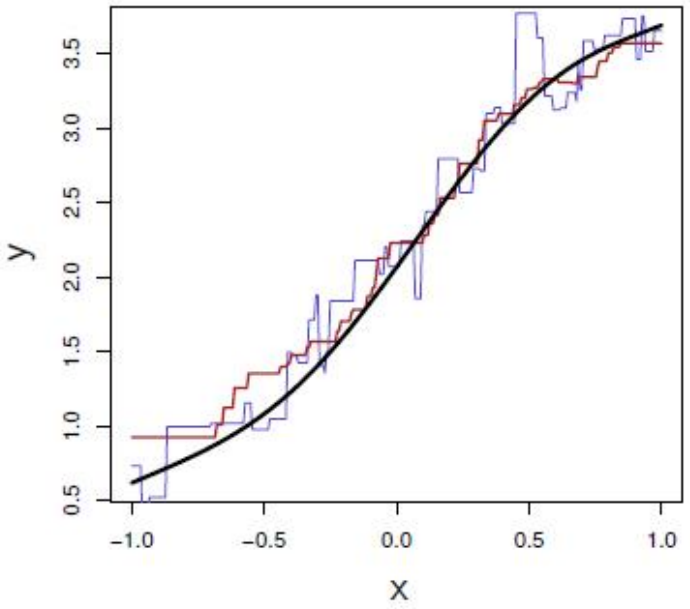
- The main task of activation function σ is to produce **nonlinear effects**.
- Sigmoid activation function $\sigma(z) \equiv \frac{1}{1+e^{-z}}$.
- More widely used activation ReLU. **σ (z)=Max(0,z)**.





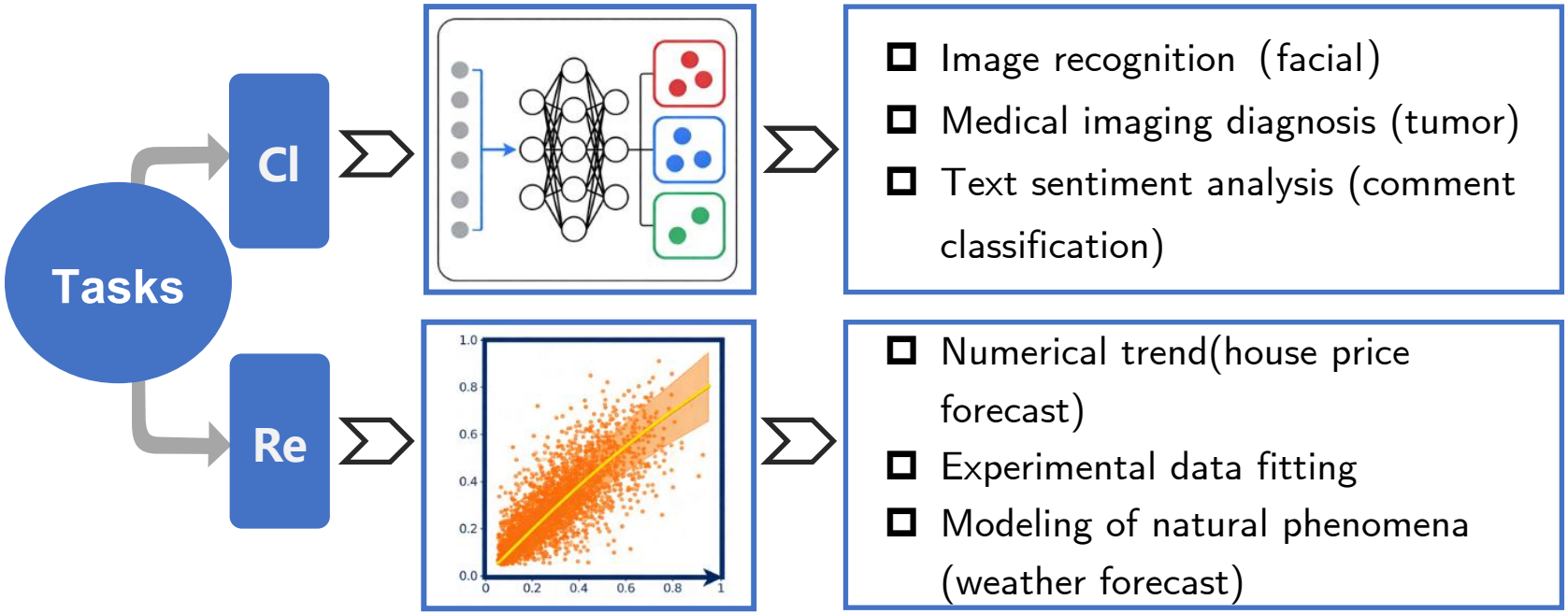
Core tasks of neural network

- Variables can be characterized as either qualitative(**categorical**) or **quantitative**. Neural network can solve two core tasks: **Classification** and **Regression**.
- Quantitative variables take on numerical values, problems with a quantitative response as regression problems.
- Qualitative variables take on values in one of K different categories. While those involving a qualitative response are often referred to as classification problems.



Core tasks of neural network

- Classification and Regression.



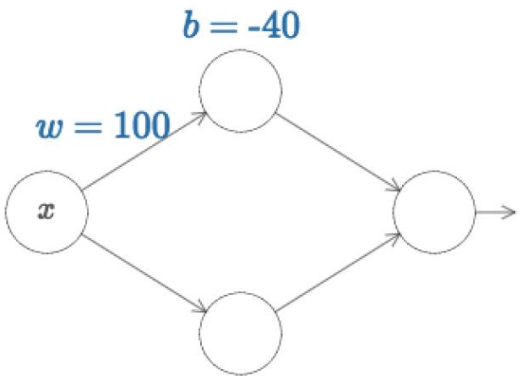
- Image recognition (facial)
- Medical imaging diagnosis (tumor)
- Text sentiment analysis (comment classification)
- Numerical trend(house price forecast)
- Experimental data fitting
- Modeling of natural phenomena (weather forecast)

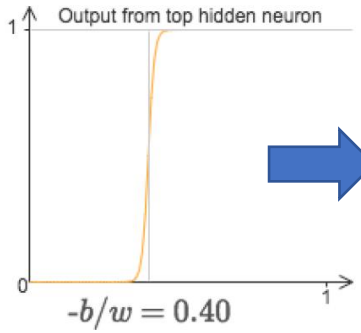
Stochastic gradient descent

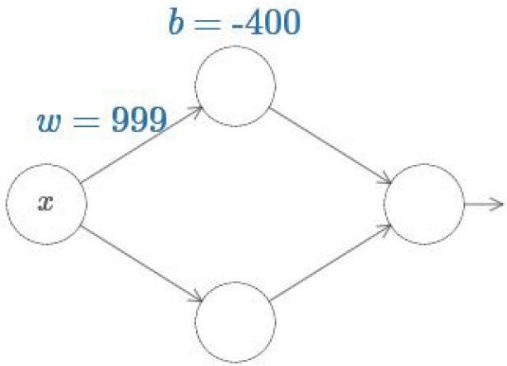
- In fact, number of challenges make the cost of gradient calculation too large to afford.
- I. $C = \frac{1}{n}\sum_x C_x \qquad C_x \equiv \frac{\|y(x)-a\|^2}{2}$
- When inputs are large, the learning is quite slow.
- Stochastic Gradient Descent(SGD) estimates the gradient by computing for a small sample of randomly chosen training inputs to speed up.
- II. The amount of calculating the derivatives is very large by the traditional methods, which extremely limits the application and development of neural network.

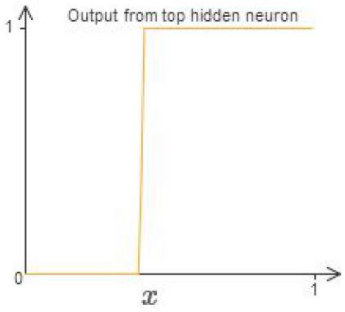
Universal Approximation Theorems

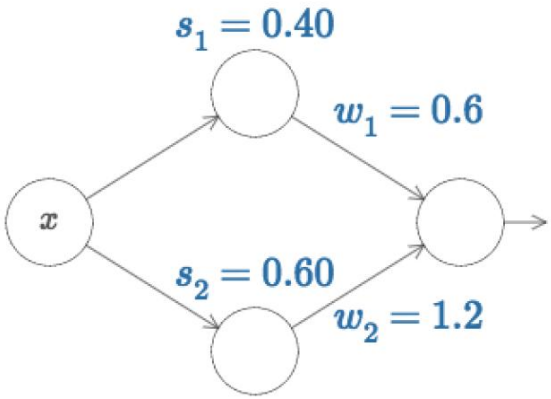
- Activation. $\sigma(z) \equiv \dfrac{1}{1+e^{-z}}$. z=wx+b=w(x-s), s=-b/w.

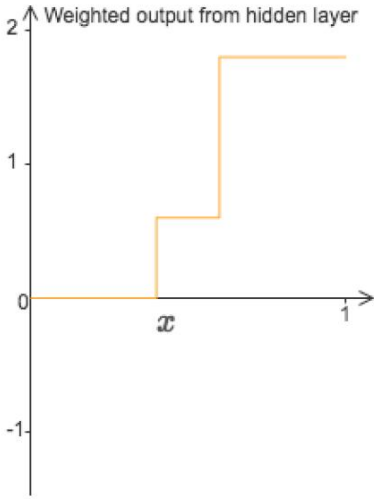


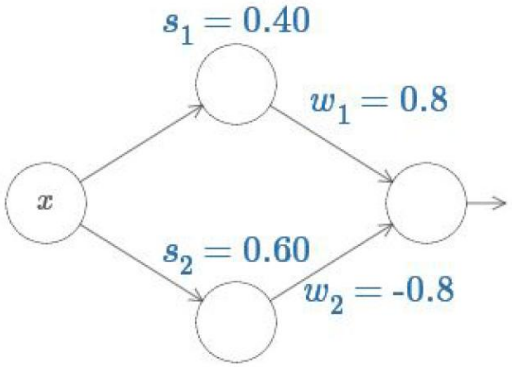


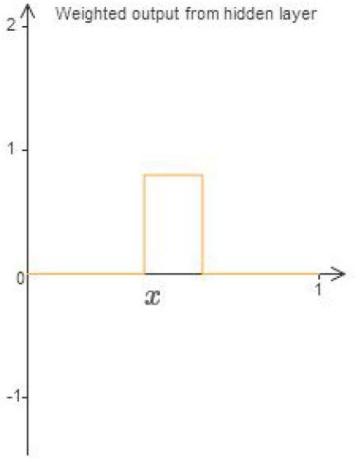






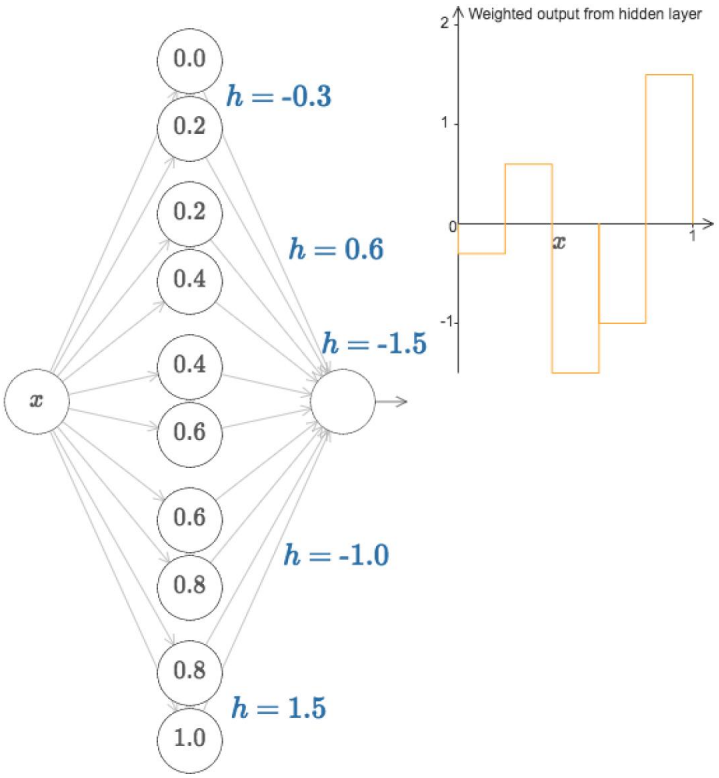


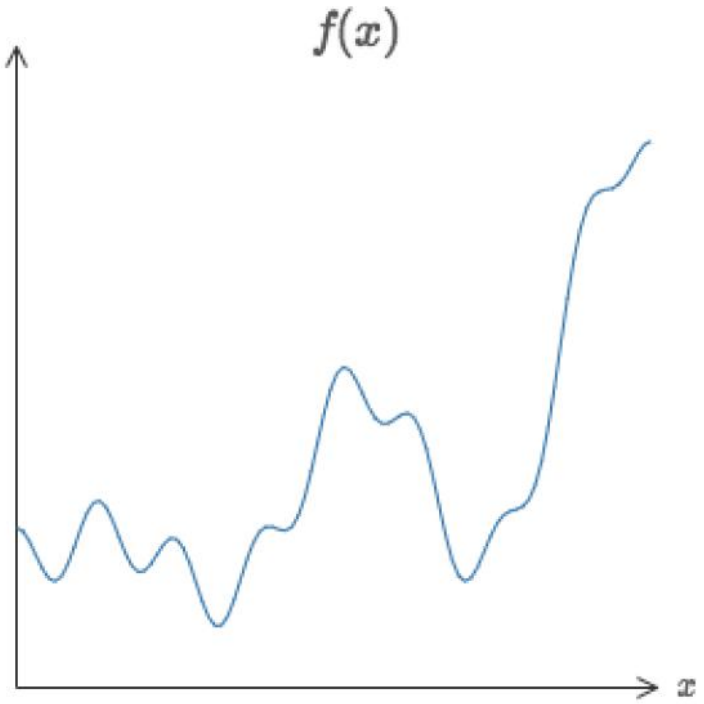




Universal Approximation Theorems

- Activation. $\sigma(z) \equiv \dfrac{1}{1+e^{-z}}$ z=wx+b=w(x-s), s=-b/w.

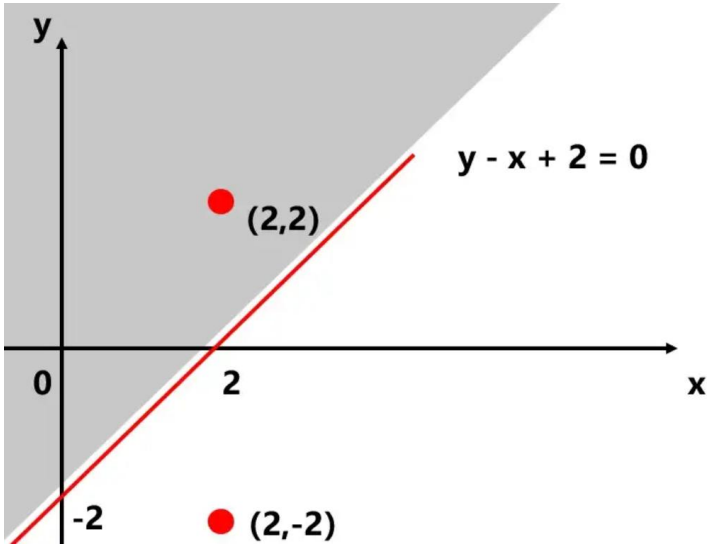


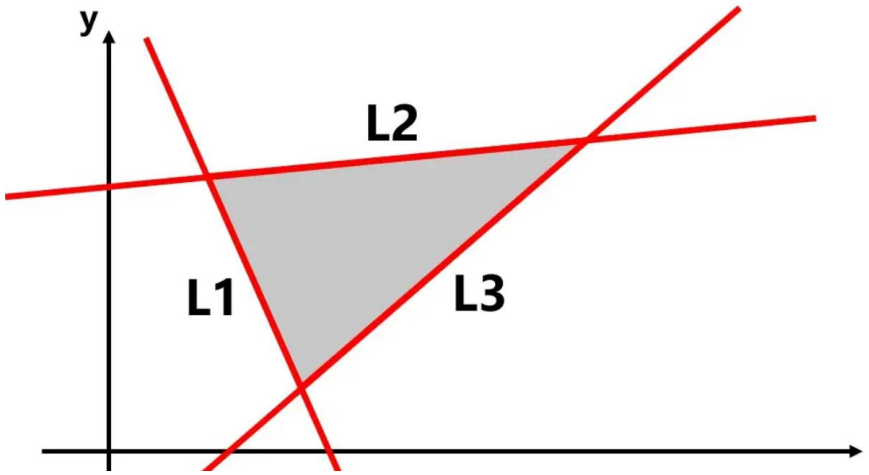


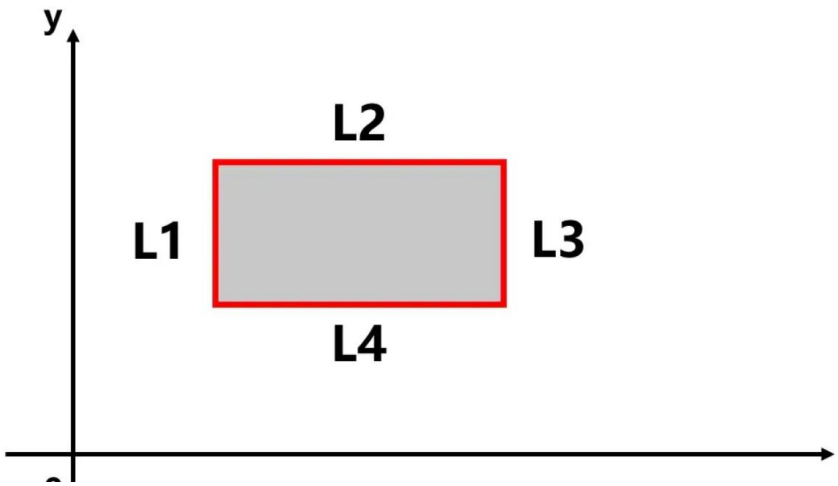
$$f(x) = 0.2 + 0.4x^2 + 0.3x\sin(15x) + 0.05\cos(50x)$$

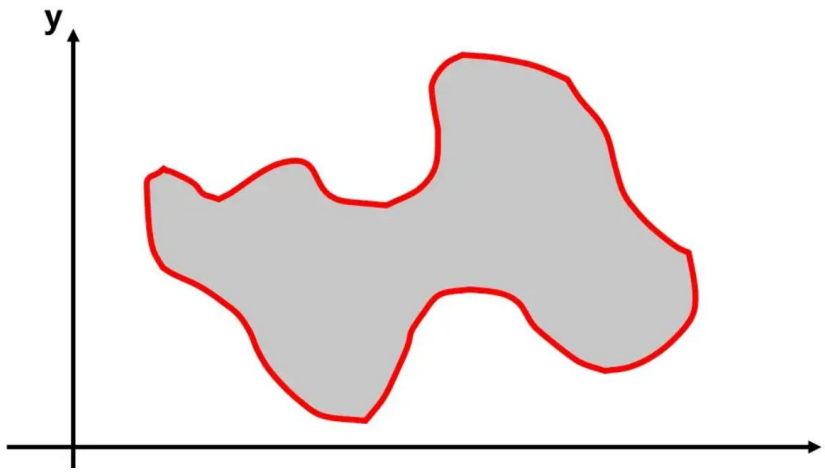
Universal Approximation Theorems

- Classification problem.

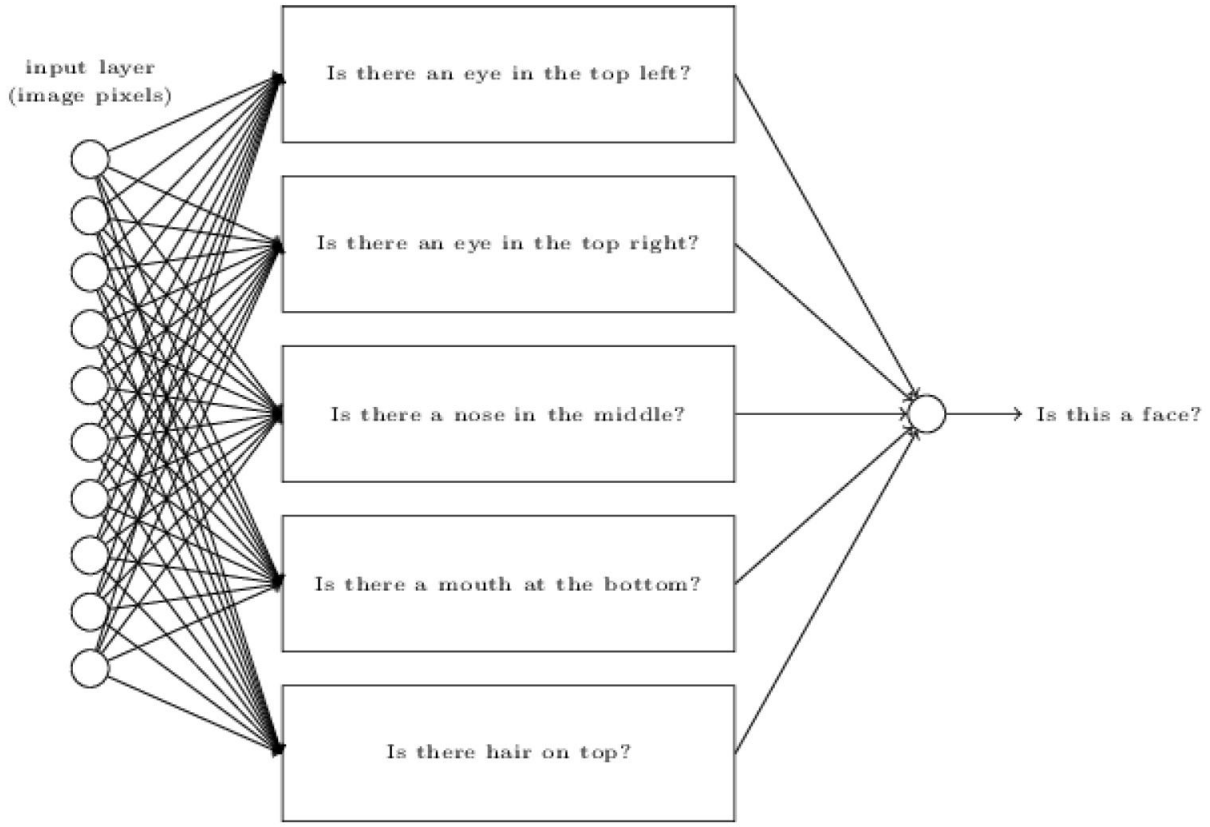


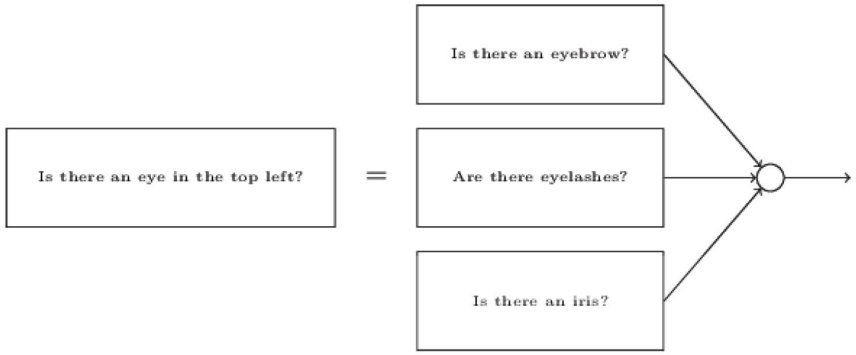






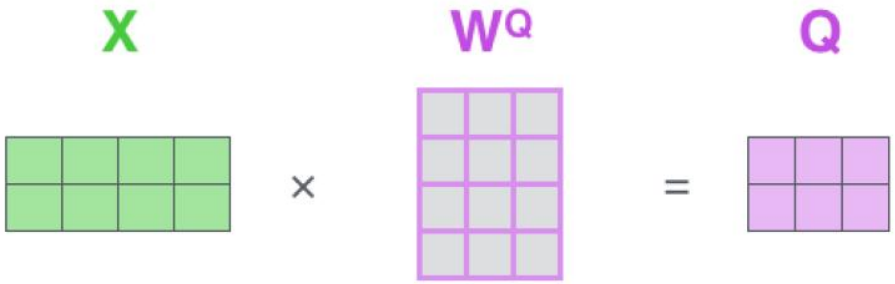
Deep learning



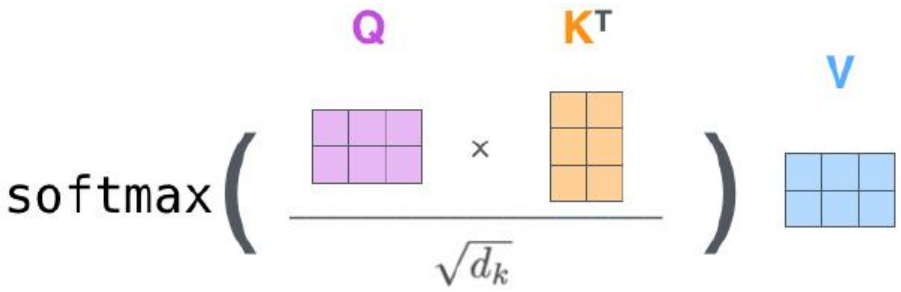


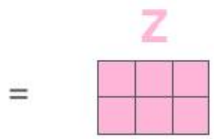
Self-attention mechanism of Transformer

- Self-attention mechanism is the core of Transformer.



$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_K}}\right)V,$$





Hadron encode

- Physical quantum number of hadron should first be encoded as features which can be recognized by neural network model.
- Encode scheme by quantum number, quark contents, and continuous values(mass, width).
- Quantum number includes I, I_3, J, G, P, C.

Quantum number	Values			Encode
I	0, 0.5, 1			Continuous
I_3	-1, -0.5, 0, 0.5, 1			Continuous
J	0, 1, 2, … , 6			Continuous
G	-1	N/A	1	Discrete
P	-1	N/A	1	Discrete
C	-1	N/A	1	Discrete

Name	*I*	I_3	*G*	*J*	*P*	*C*
pi+	1	1	-1	0	-1	N/A
pi-	1	-1	-1	0	-1	N/A
pi0	1	0	-1	0	-1	1

Hadron encode

- Quark contents are encoded as continuous values

$$\boldsymbol{q} = (u, \bar{u}, d, \bar{d}, s, \bar{s}, c, \bar{c}, b, \bar{b}),$$

- Mixing quark states bring continuous values in quark contents.

$$\boldsymbol{q}_{\pi^0} = (\tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}, 0, 0, 0, 0, 0, 0),$$

$$\boldsymbol{q}_{T_{c\bar{c}1}(3900)^+} = (1, 0, 0, 1, 0, 0, 1, 1, 0, 0),$$

- We don't include the sign in the quark contents, which are contained in I_3.
- Quark coefficients are not good quantum numbers in experiments. The PDG only gives a universal flavor representation $[c_1(u\bar{u} + d\bar{d}) + c_2 s\bar{s}]$ for the light unflavored mesons with I=0.

Challenges in encoding hadrons

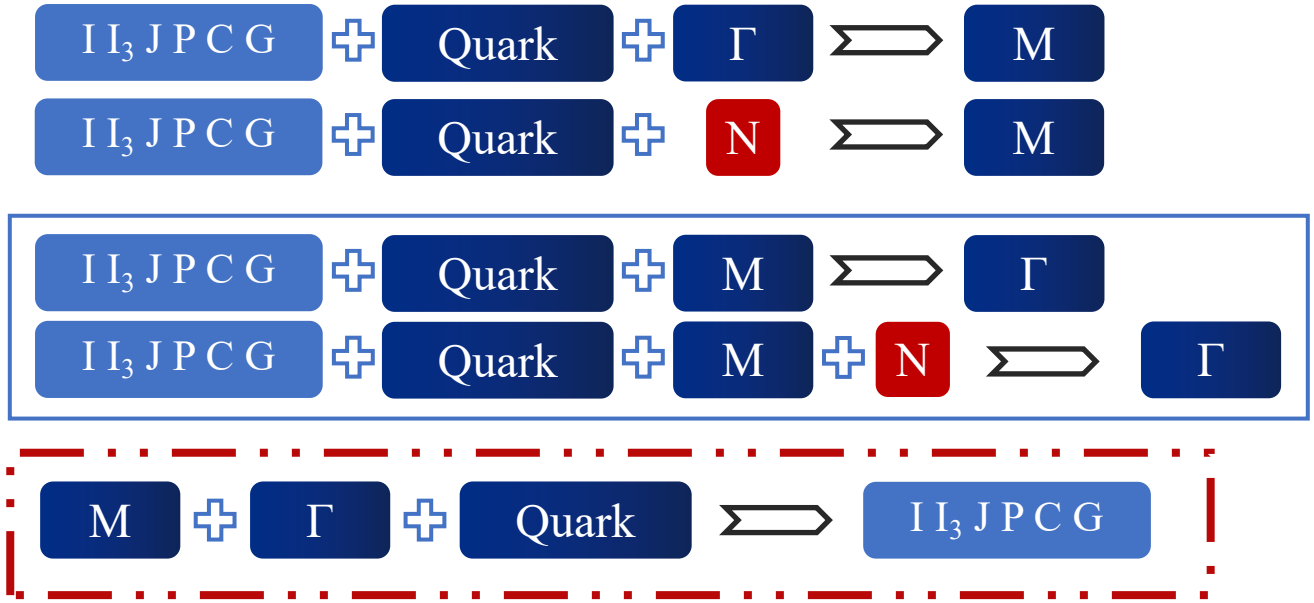
- Radial number is not a good quantum number in both experiments and theory. eg., A: some are missing in experiments; B: the so-called mixing, 2S-1D, 3S-2D……

Name	I	I_3	G	J	P	C	N	Mass	Width
ρ(770)0	1	0	1	1	-1	-1	1	775.261156	147.4
ρ(1450)0	1	0	1	1	-1	-1	2	1465	400
ρ(1570)0	1	0	1	1	-1	-1	3	1570	144. 0
ρ(1700)0	1	0	1	1	-1	-1	4	1720	250
ρ(1900)0	1	0	1	1	-1	-1	5	1880	69.0
ρ(2150)0	1	0	1	1	-1	-1	6	2044	163

- Non-natural parity mesons, J^P=1^+, 2^-, 3^+,4^-,……, are degenerate within the current quantum numbers, $D_1(2420)$, $D_1(2430)$, $D_{s1}(2536)$, $D_{s1}(2460)$, ……

Possible scenarios

- One tentative scheme is to introduce a quite phenomenological feature N labelling the mass from low to high($\times$).
- Or include other features as input, such as, mass or width, and then predicting another one.



- This also explains our goals aforementioned.

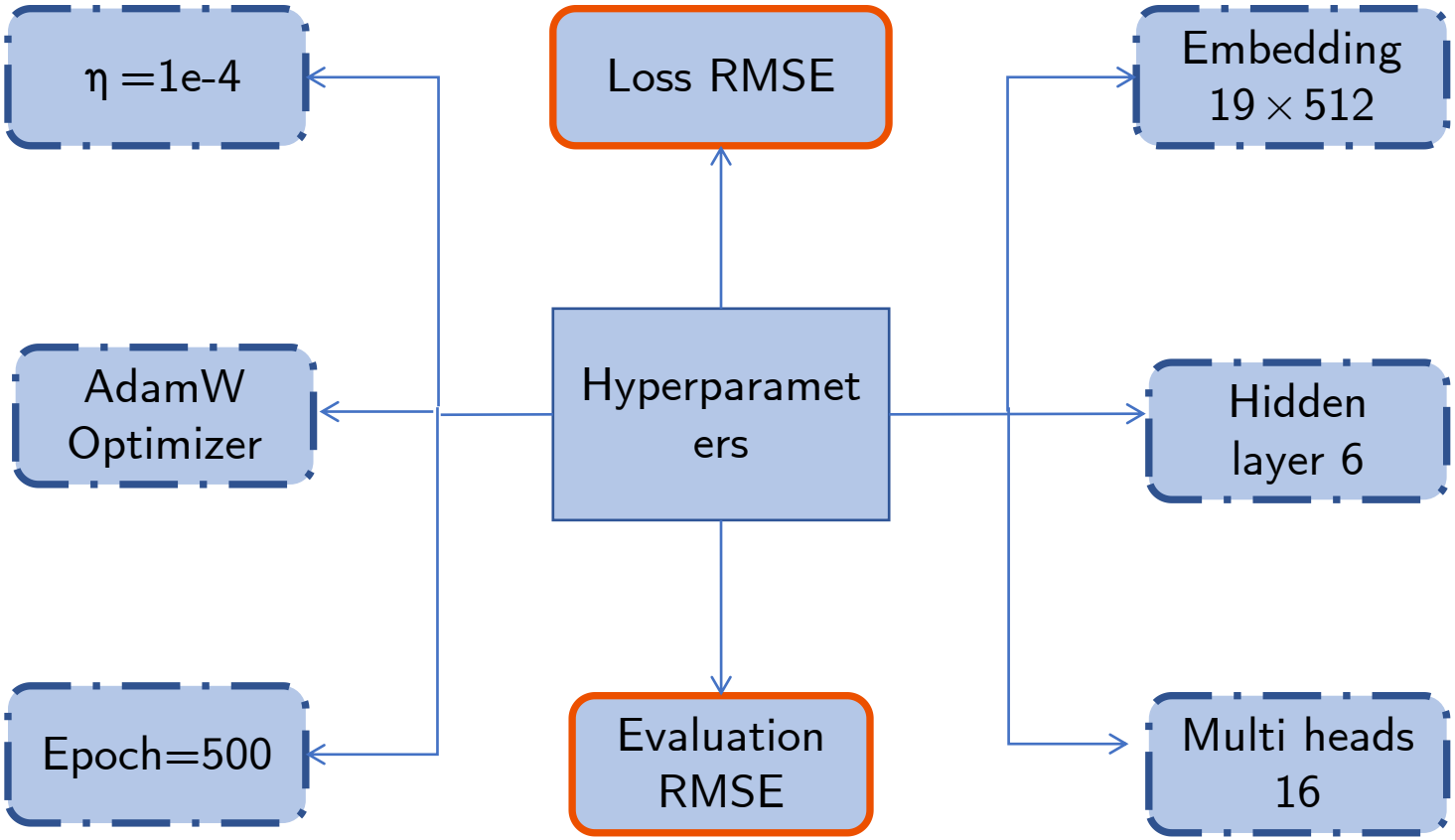
FT-Transformer

- Transformer, currently, the most famous and most influential model.
- FT-Transformer(Feature Tokenizer Transformer), designed for tableau data.
- Encode all the features to tokens as the Transformer input.

- Sophisticated algorithm $<$ simple learning algorithm $+$ good training data.
- Train data 90%, learn the inner relationships.
- Test(validation) data 10%, test the performance.

Hyper-parameters

- Set the hyperparameters for the model.



Comparison

Tab. I: Comparison of meson widths (in units of MeV) with theoretical predictions for test sample mesons. $\Gamma_{\rm Exp}$ denotes experimental width from the PDG [11]; $\Gamma_{\rm Th.1(2)}$ denotes theoretical predictions from various approaches; $\Gamma_{\rm This}$ denotes the predictions of this work. Here the relative errors are calculated by Error=$|\Gamma_{\rm pred} - \Gamma_{\rm Exp}|/\Gamma_{\rm Exp} \times 100\%$.

Particle	$\Gamma_{\rm PDG}$	$\Gamma_{\rm Th.1}$	Ref.	Error (%)	$\Gamma_{\rm Th.2}$	Ref.	Error (%)	$\Gamma_{\rm This}$	Error (%)
$B^*_{s2}(5840)$	1.49	1.0	[56]	32.9	10.3	[57]	560	1.49	0.2
$K^*_2(1430)$	100.0	80.1	[58]	20.0	-	-	-	99.8	0.2
$K^*(1410)$	231.8	214	[58]	7.7	-	-	-	231.3	0.2
$D_1(2420)$	31.3	25	[59]	20.1	-	-	-	31.1	0.7
$D_2(2740)$	88.5	56.3	[60]	36.3	111.6	[61]	26.1	88.1	0.4
$\psi(4360)$	118.4	37.5	[62]	216	38.6	[62]	207	110.6	6.6

Tab. II: Comparison of predicted widths (MeV) for several unmeasured ($b\bar{b}$) states with theoretical estimates from Godfrey-Moats (GM) in ref. [63].

State J^{PC}	$\chi_{b0}(1P)$ 0^{++}	$\chi_{b0}(2P)$ 0^{++}	$\chi_{b1}(1P)$ 1^{++}	$\chi_{b1}(2P)$ 1^{++}	$h_b(1P)$ 1^{+-}	$h_b(2P)$ 1^{+-}	$\chi_{b2}(1P)$ 2^{++}	$\chi_{b2}(2P)$ 2^{++}	$\chi_{b2}(3P)$ 2^{++}
This	73.3	59.4	0.16	0.93	0.35	1.23	19.2	17.3	14.6
GM	2.6	2.6	0.096	0.117	0.073	0.084	0.18	0.23	0.25

Predictions

For the $X(4160)$ state, the PDG favors the $J^{PC} = 2^{-+}$ assignment with a significance of over 4σ. Under the assumption of a 2^{-+} $[c\bar{c}u\bar{u}]$ tetraquark state, the predicted width 138.5 MeV is well consistent with the experimental value 136 MeV. For $X(4630)$, the PDG favors the 1^- assignment over the 2^-; the model shows that under the $[c\bar{c}s\bar{s}]$ hypothesis, both the 1^{-+} and 2^{-+} yield predictions consistent with the experimental width. For $T_{c\bar{c}}(4100)$, the PDG suggests the assignment 0^+ or 1^-. The model's prediction 164.3 MeV under the 1^{-+} hypothesis is quite close to the experimental value 150^{+80}_{-70} MeV and then favors the latter assignment. The model shows good consistency between its predictions and the preliminary conclusions drawn from experimental analyses by the PDG, which in turn provides independent, data-driven support for these findings.

For particles where the PDG has not provided a clear preference, the model offers independent assignments for their quantum numbers. For example, the $J^P = 0^+$ hypothesis for $D(3000)$ yields the predicted width with about a 2% relative error from the experimental value. For the tetraquark $T_{c\bar{c}}(4050)$, under the 3^{-+} assignment the model gives a predicted width that is quite close to the experimental value. Plausible quantum number assignments were also identified for several other particles, such as $X(1750)$ and $B^*_{sJ}(5850)$.