上海理工大學

Based on arxiv: 2407.08682

Code： https://github.com/USST-HEP/MIParT

Jet Tagging with More-Interaction Particle Transformer

Kun Wang

University of Shanghai for Science and Technology (USST)

16 November, 2024 @ CLHCP2024

Outline

- Introduction to Jet Tagging with Deep Learning
- Overview of Transformer Models
- Architecture of the More Interaction Particle Transformer (MIParT)
- Results and Discussion
- Conclusion

Jet Tagging

- Jets are collimated sprays of particles produced in high-energy collisions.

- Identifying the particle that initiated the jet is complex and challenging.

- Jet Tagging is critical for revealing fundamental physical processes and discovering new particles.

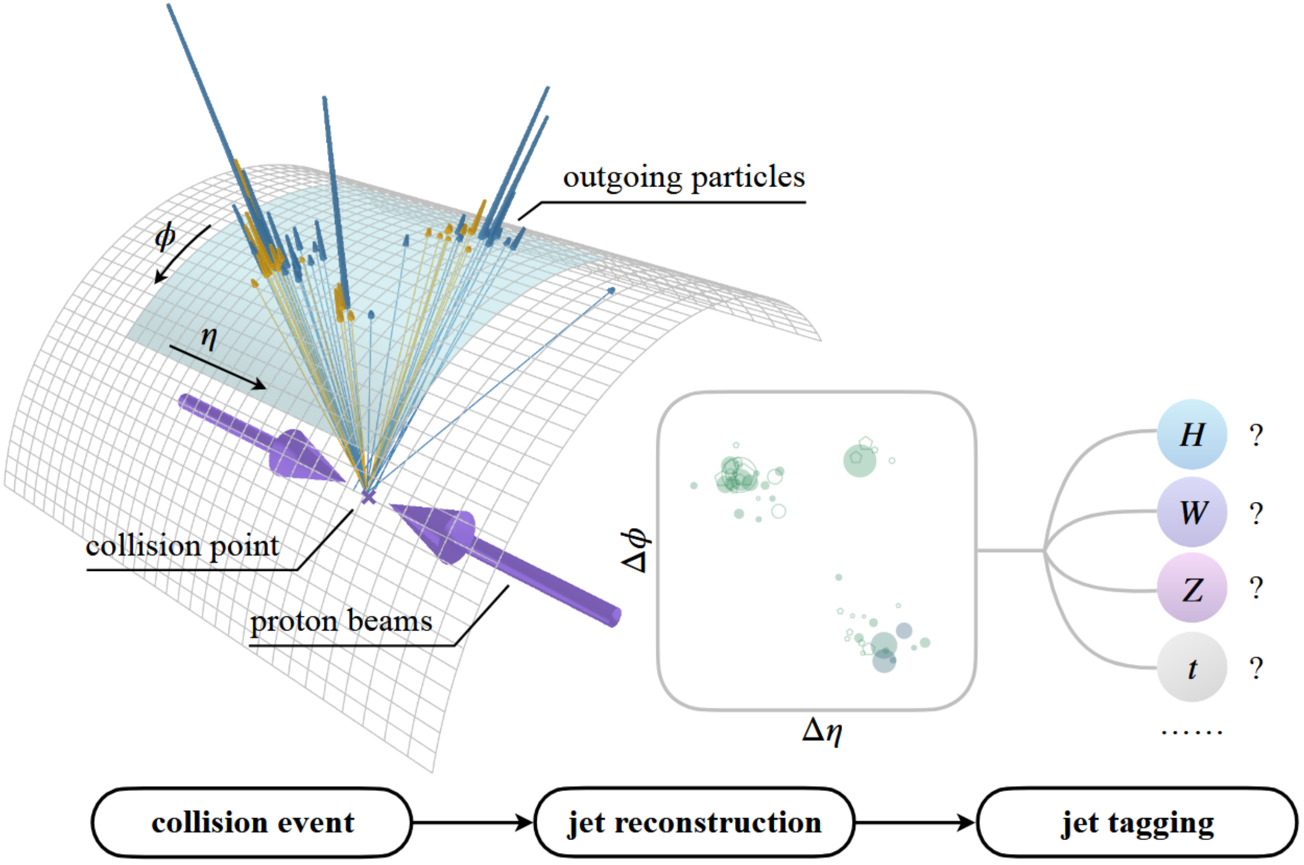
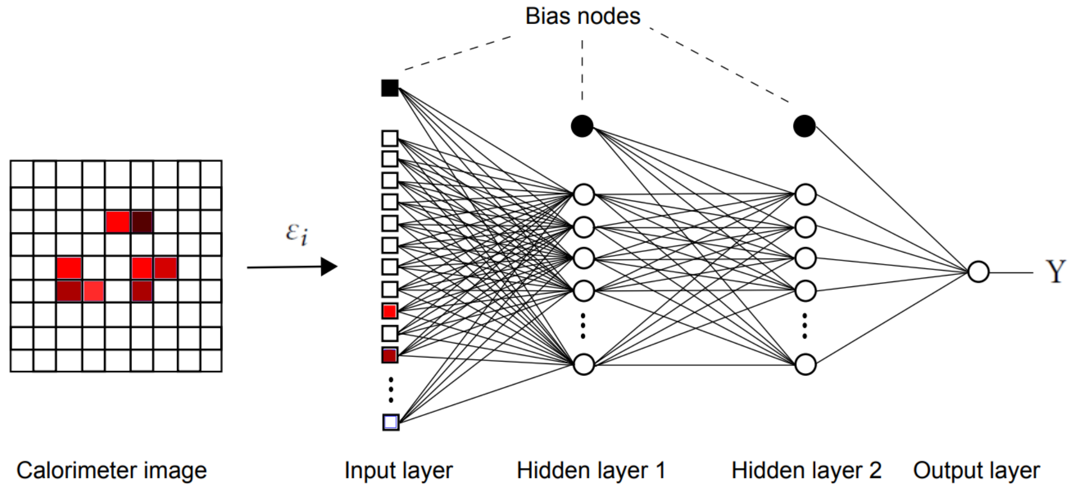


Fig from 2202.03772

History of Jet Tagging with Deep Learning

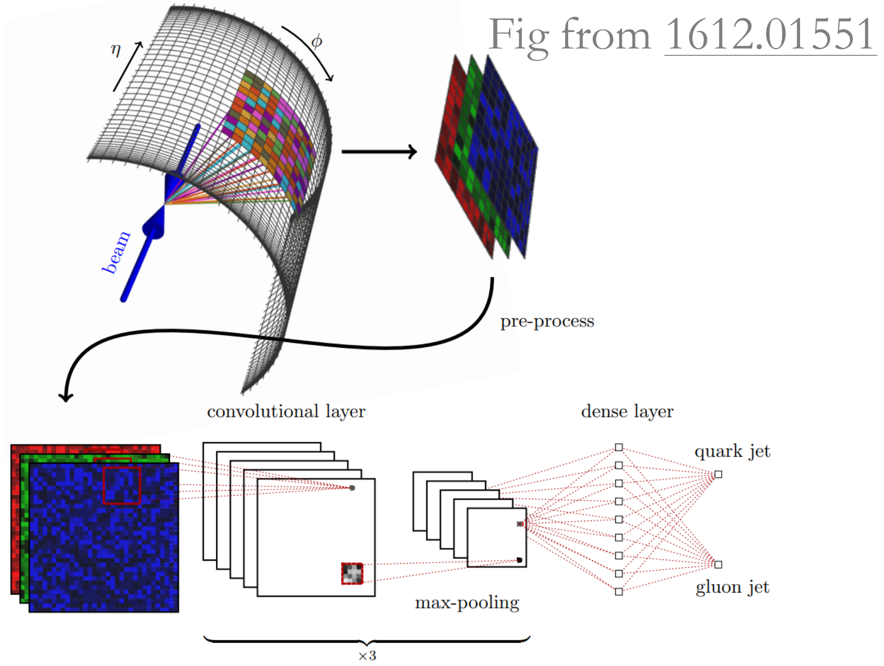
- DNN

Fig from 1501.05968



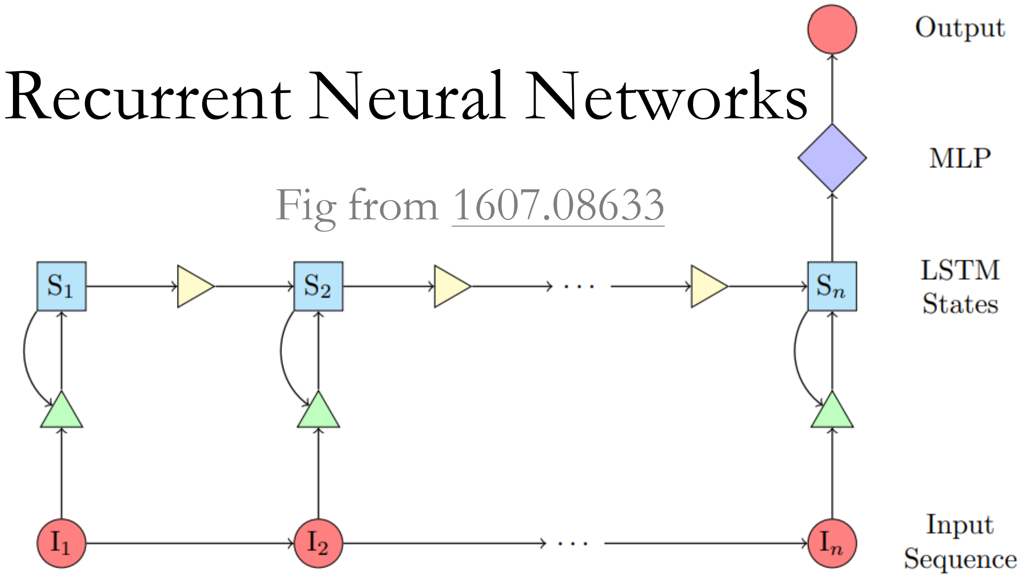
- CNN

Fig from 1612.01551



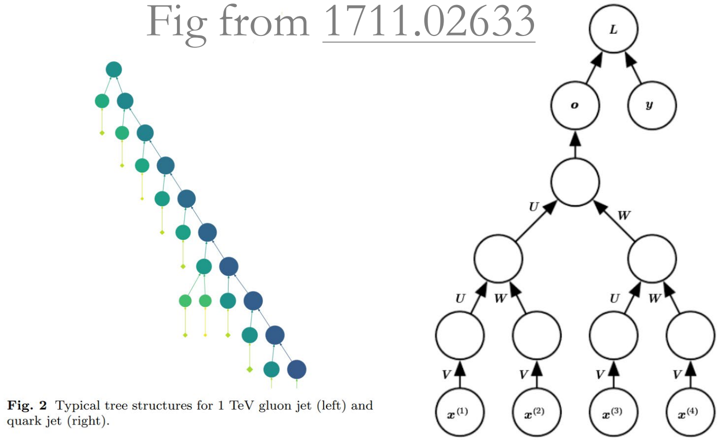
- RNN: Recurrent Neural Networks

Fig from 1607.08633



- RvNN: Recursive Neural Networks

Fig from 1711.02633



History of Jet Tagging with Deep Learning

- Energy Flow Networks: (Via DeepSet)
- ParticleNet: (Via Point Cloud)

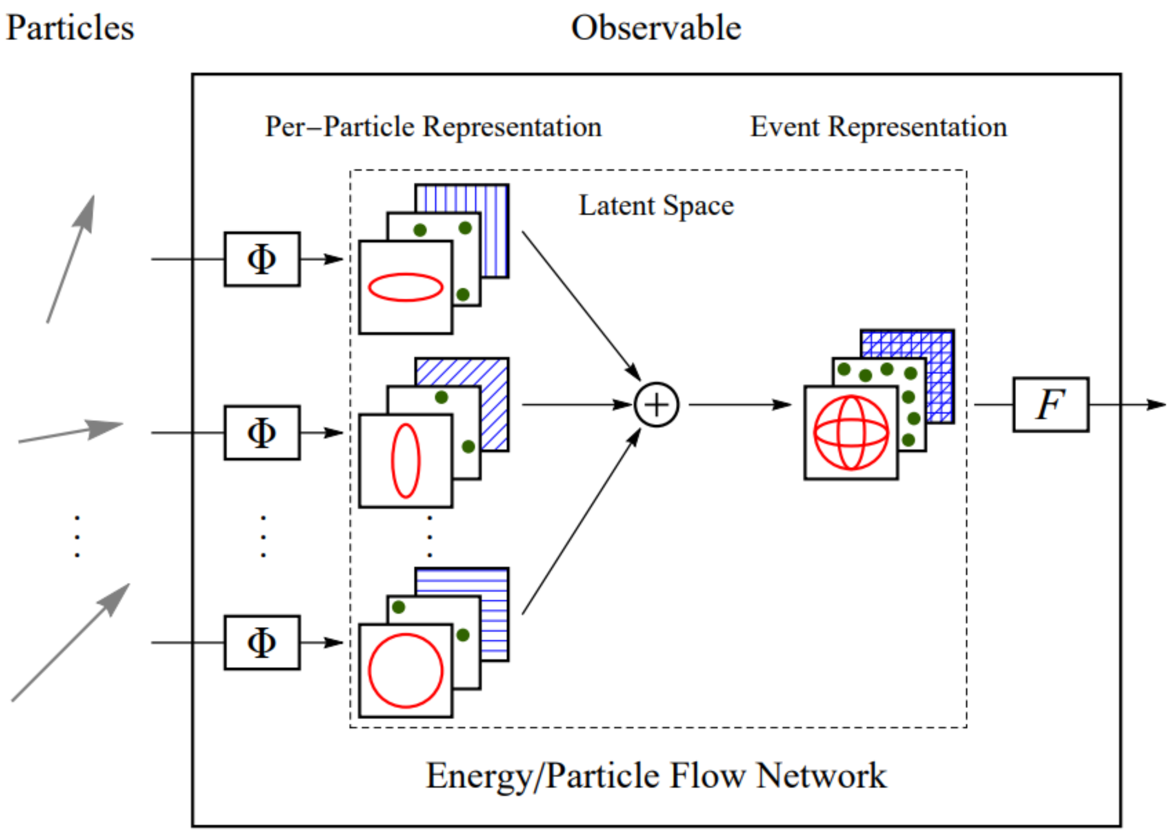


Fig from 1810.05165

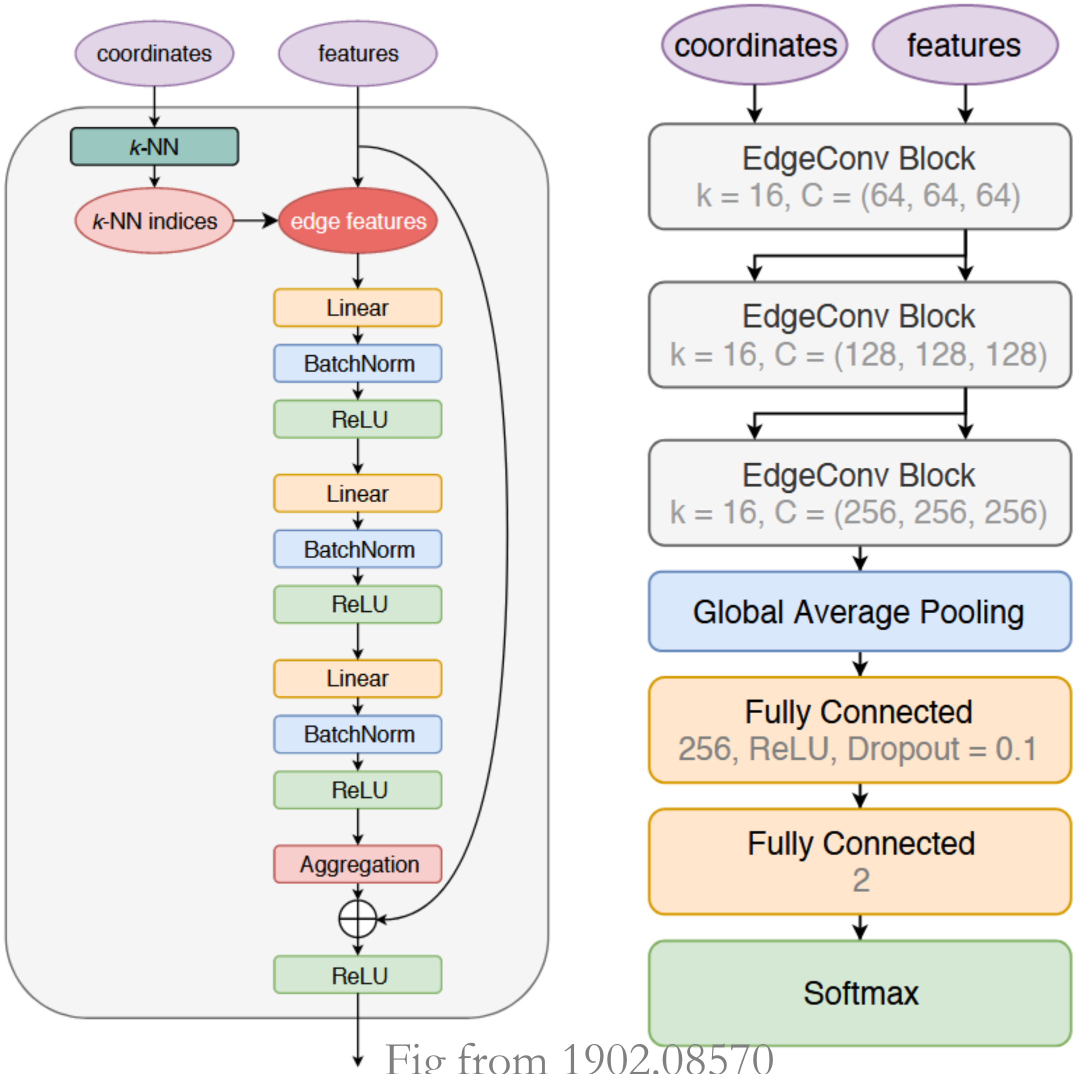


Fig from 1902.08570

- ABCNet: (Via Attention)
- Particle Transformer(Via Transformer)

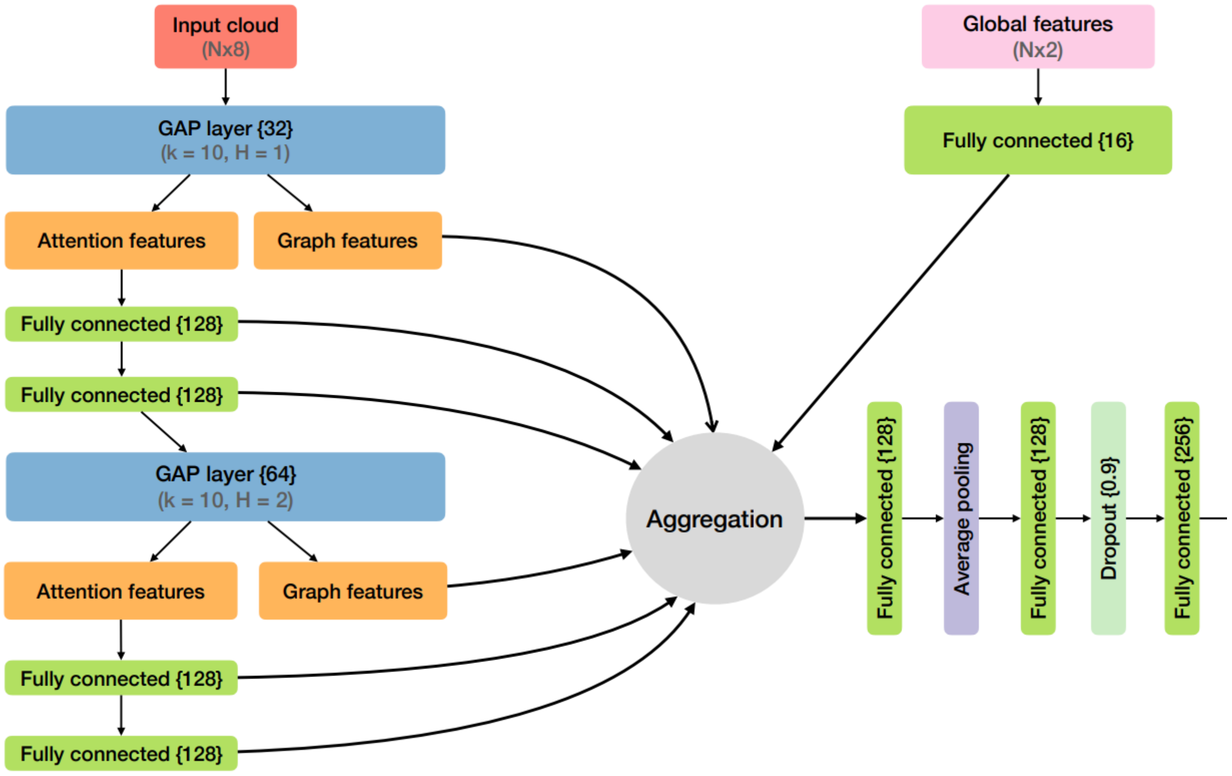


Fig from 1810.05165

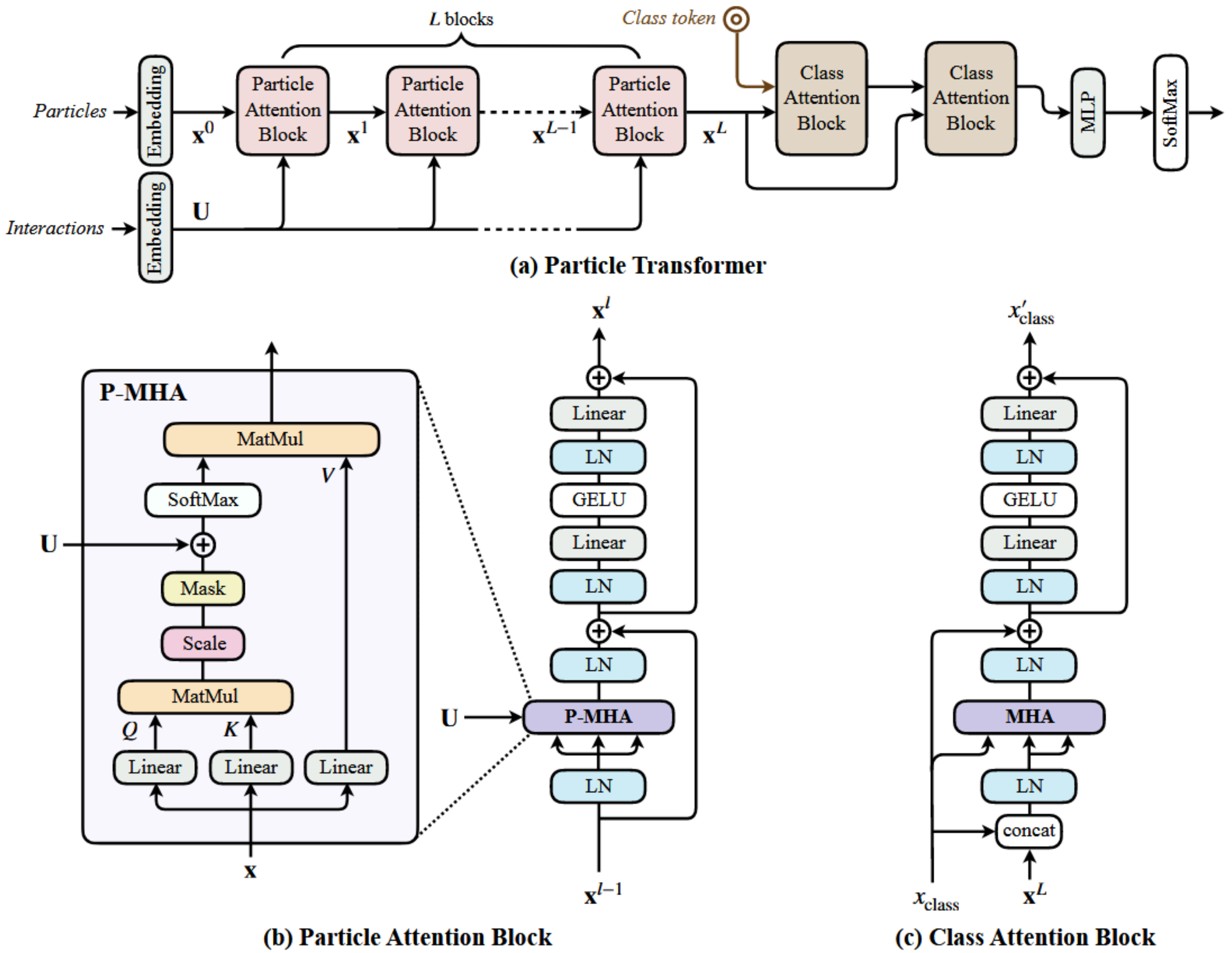
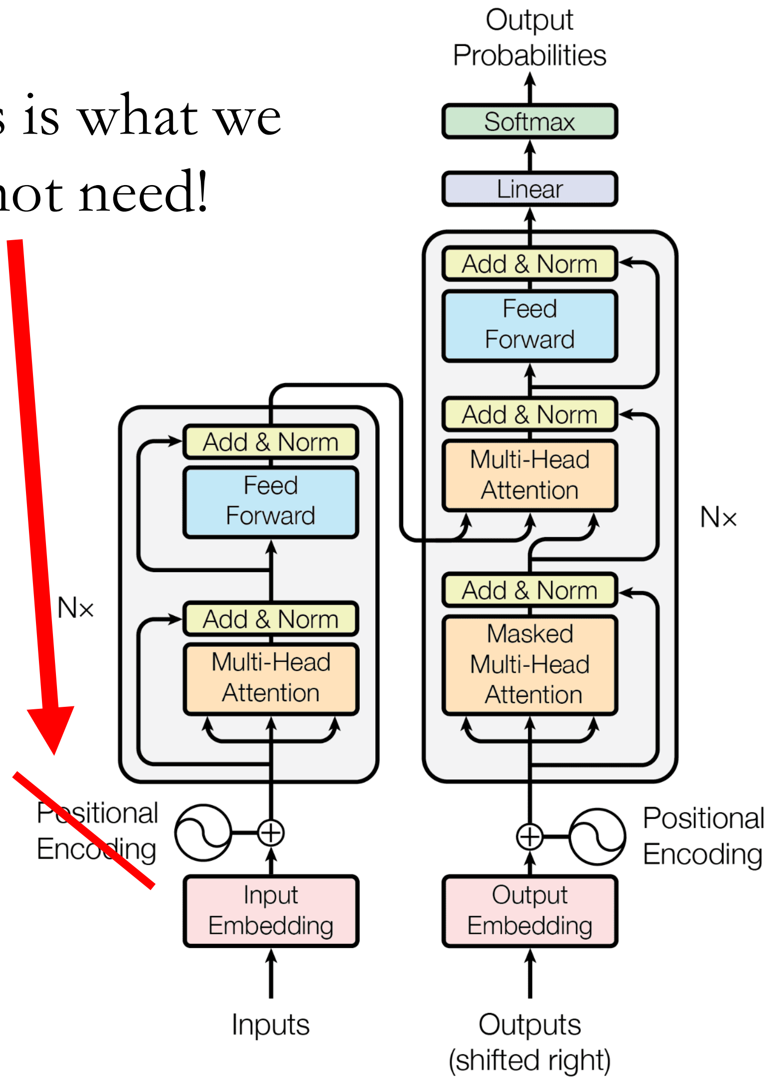


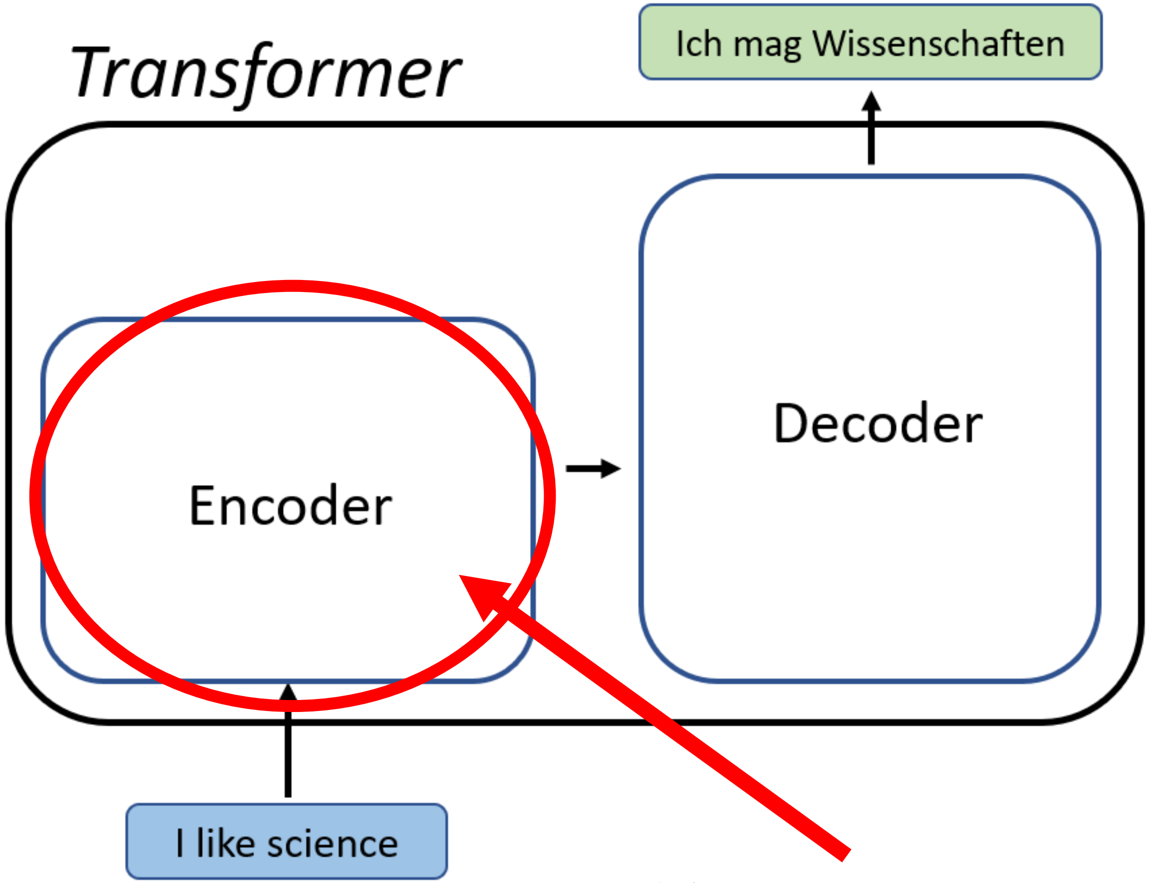
Fig from 2202.03772

Transformer Models

This is what we do not need!



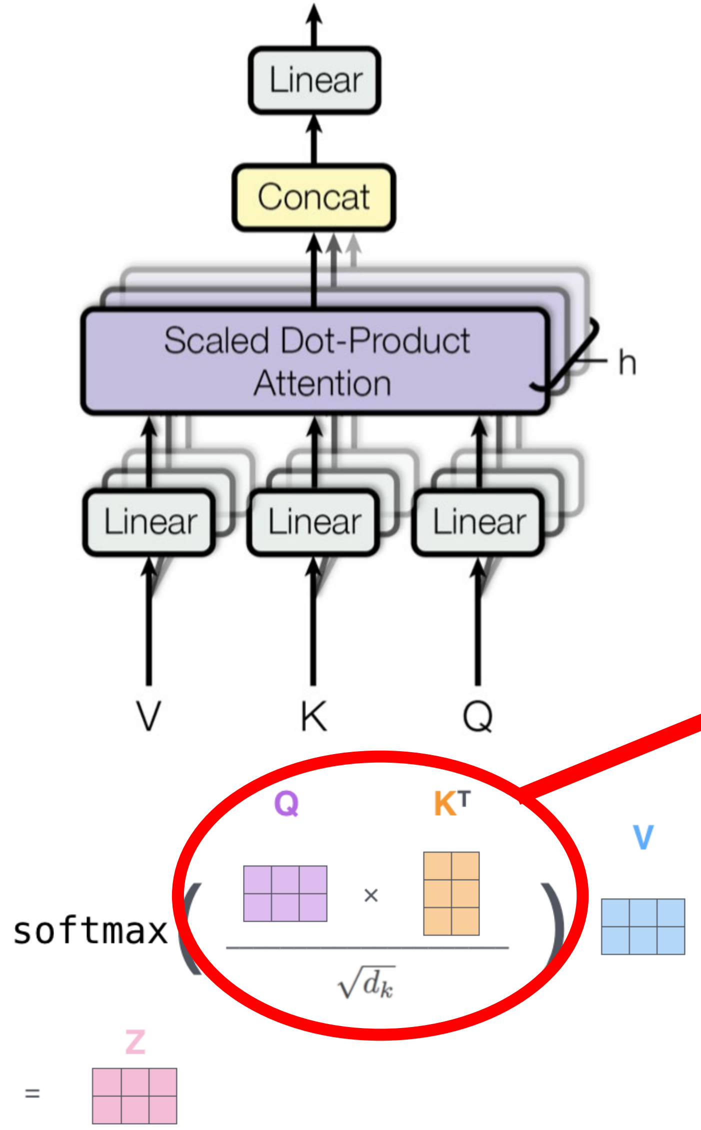
Transformer

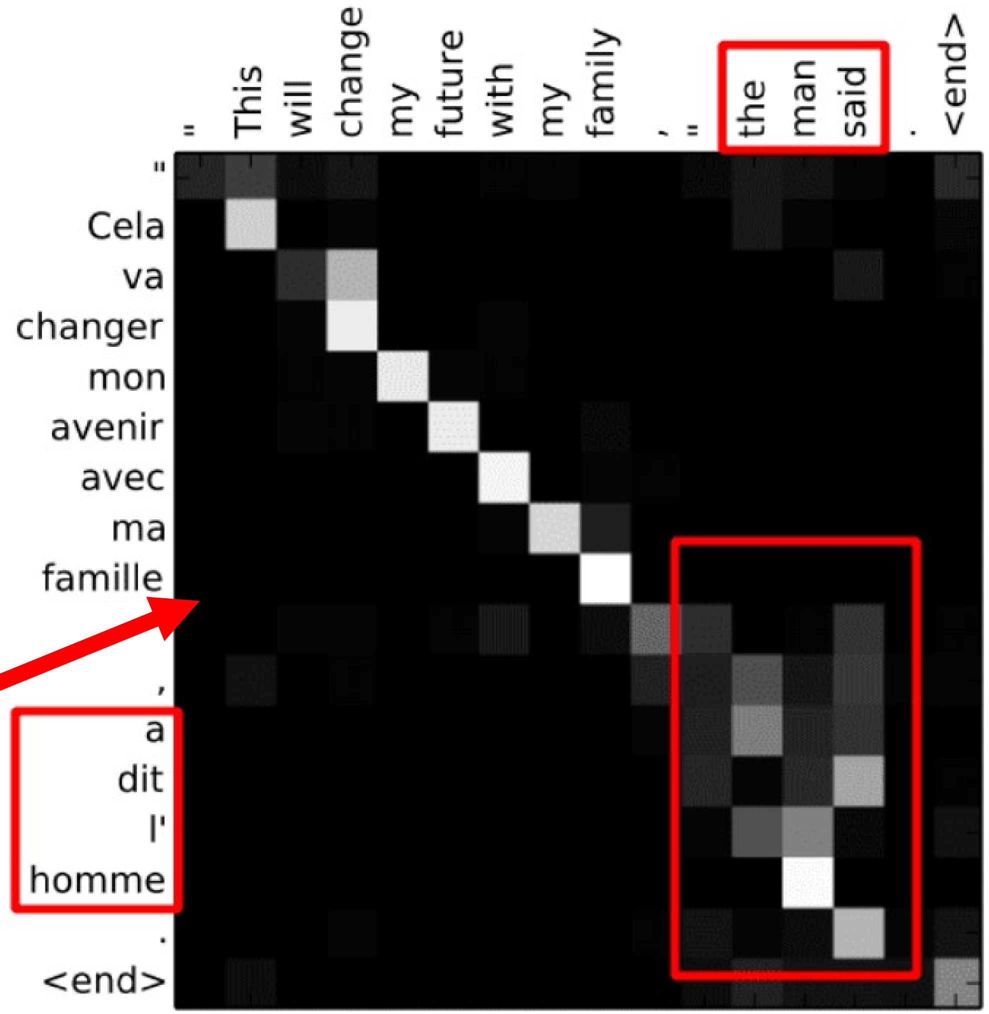


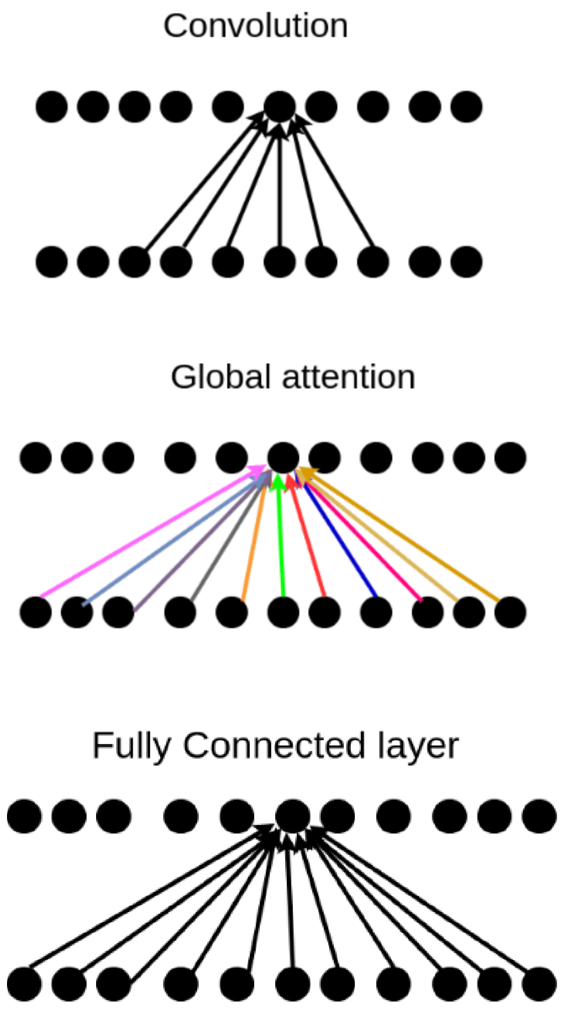
This is what we need!

Attention Mechanism

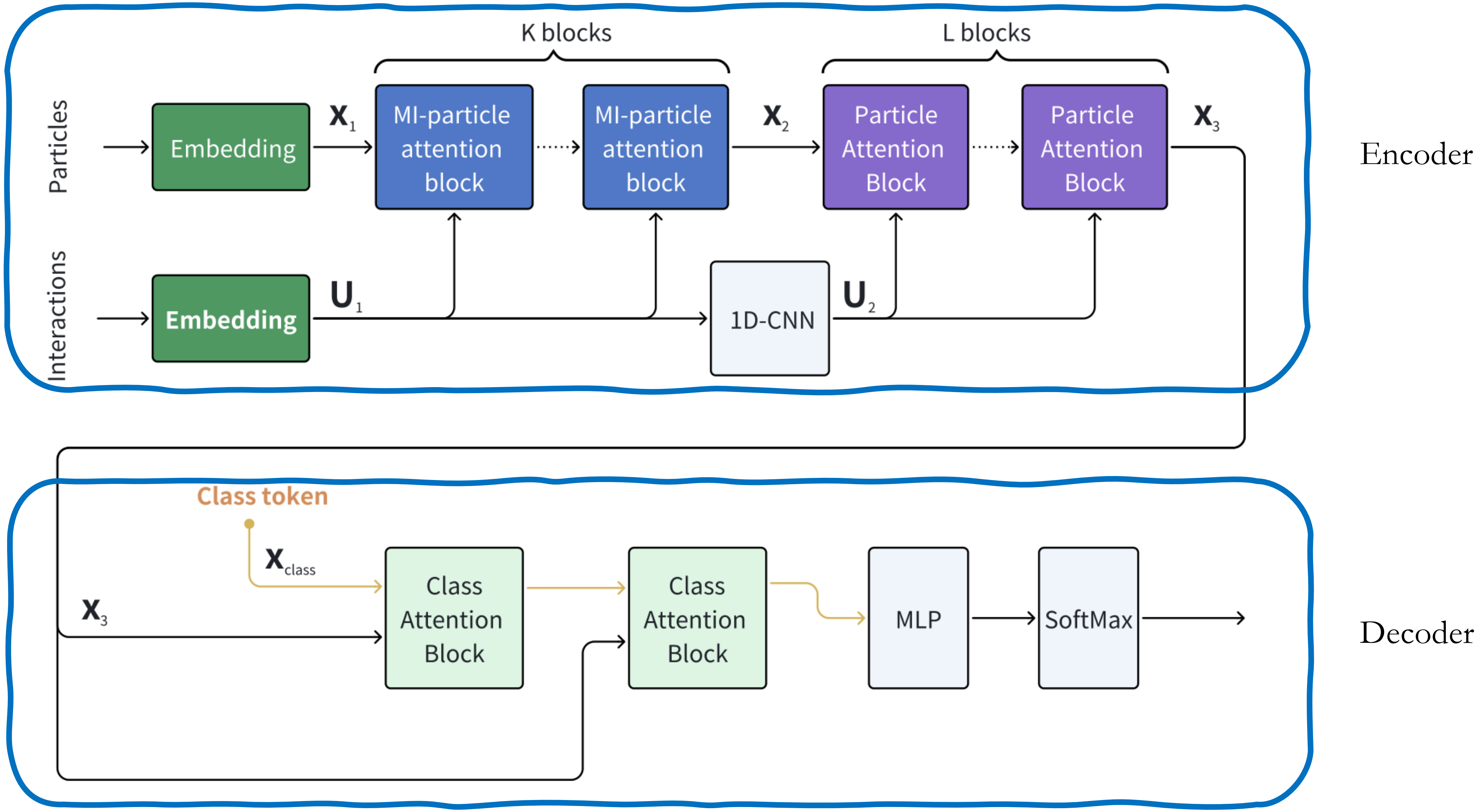
Multi-Head Attention







Architecture of MIParT

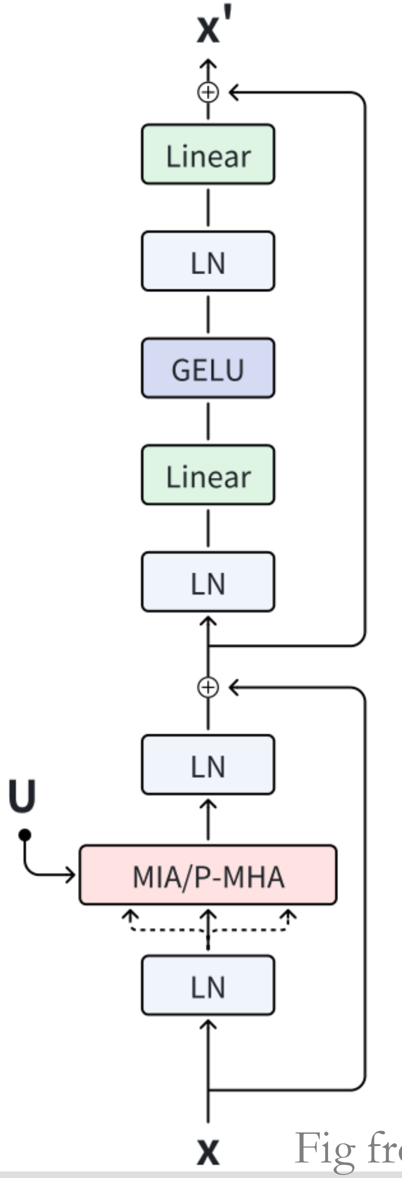


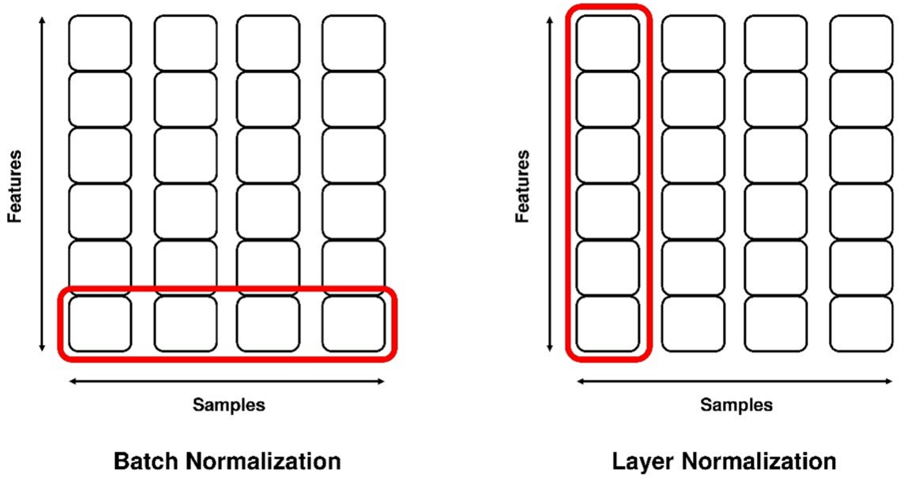
More-Interaction Particle Transformer (MIParT)

Fig from 2407.08682

Attention Block

Transformer →Normformer





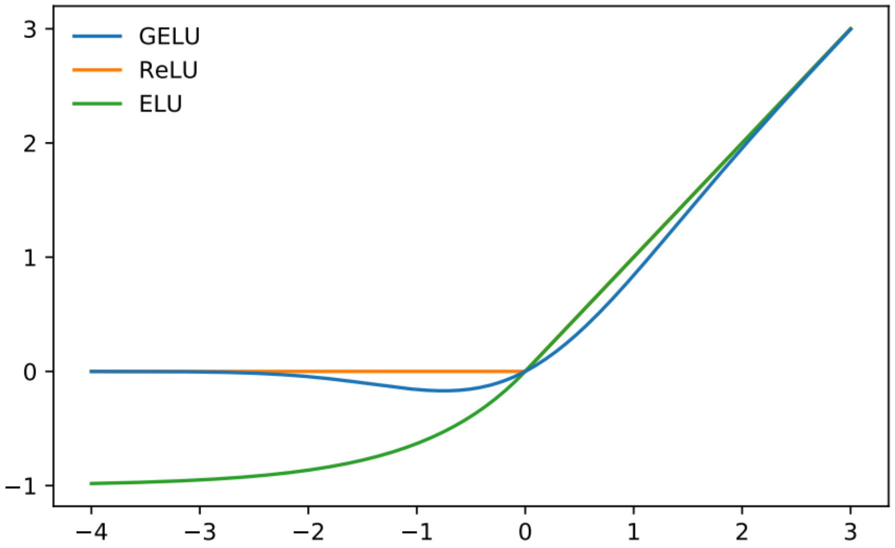


Fig from 2407.08682

BatchNorm: Better Suited for Computer Vision , Less effective in NLP due to unaligned word vectors and incomparable features at the same positions.

LayerNorm: Effective for NLP , Performs normalization at the layer level, so its effectiveness does not depend on the batch size.

GELU works better than ReLU and ELU because it provides a smoother way of activating neurons, which helps the model learn more complex patterns.

GELU handles both positive and negative values more effectively than ReLU, which ignores negative values, and ELU, which can be more complex to compute.

MI-Particle Attention Block

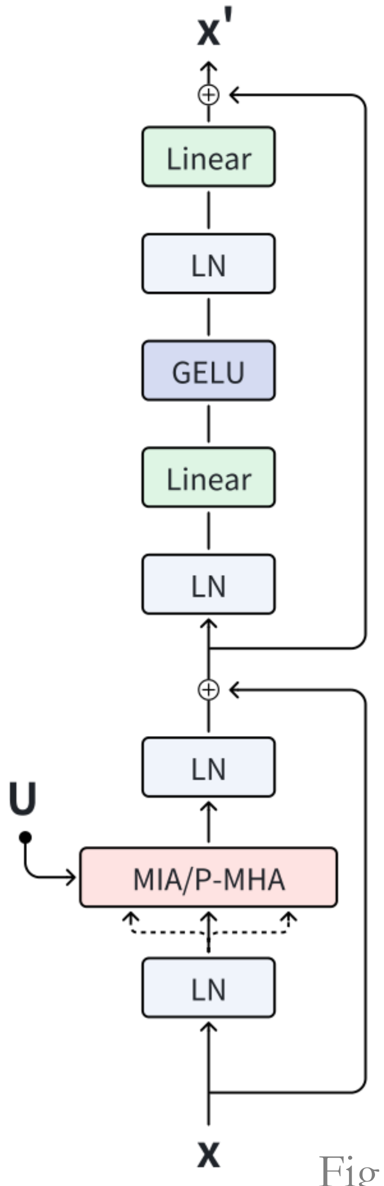
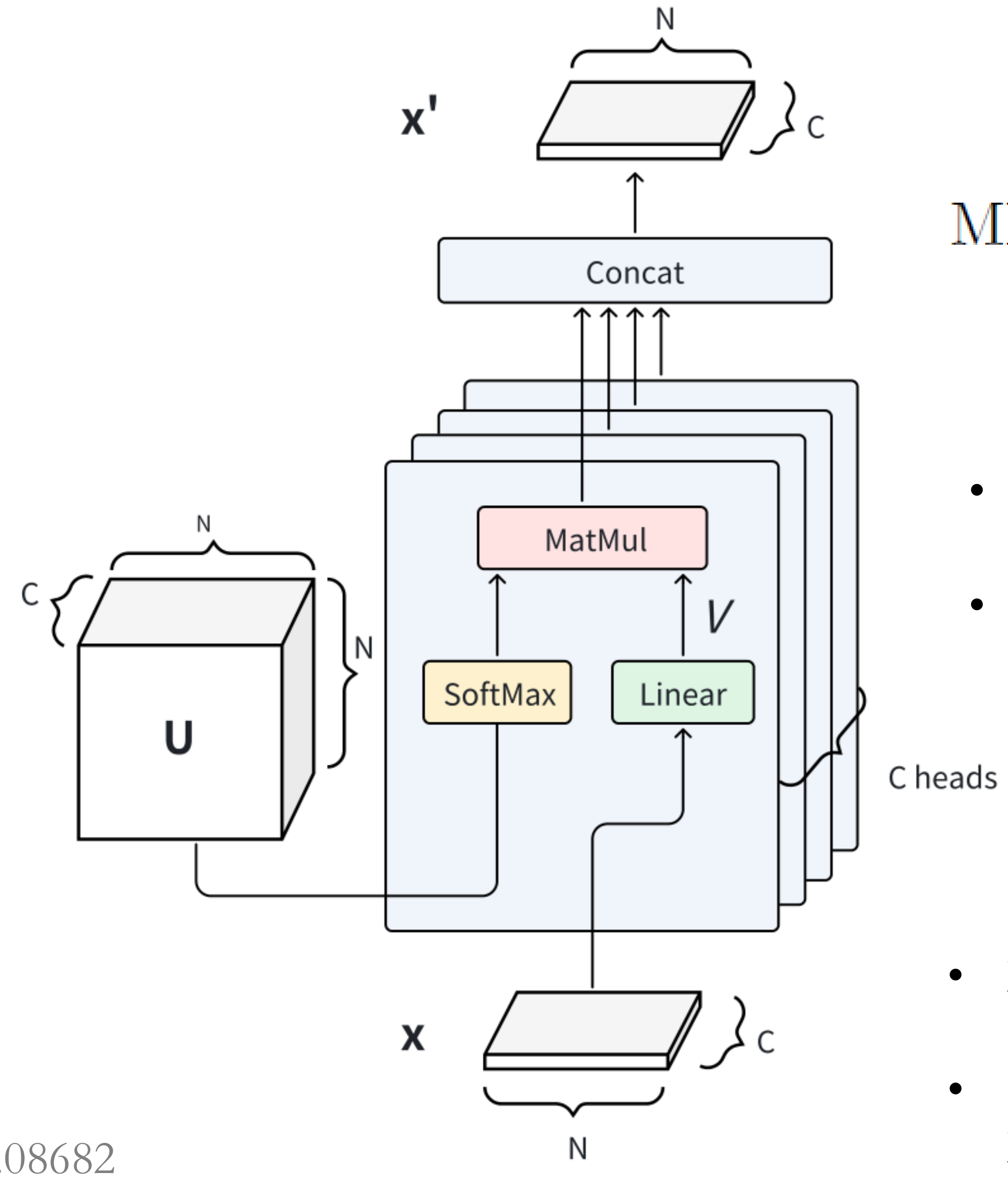


Fig from 2407.08682



$$\mathrm{MIA}(\mathbf{U}, V) = \mathrm{SoftMax}(\mathbf{U})V,$$

- The shape of U is (N,N,C), while both the
- Input x and the output x′ have the same shape (N,C)

- LN represents Layer Normalization
- GELU represents the Gaussian Error Linear Unit activation function

Particle Attention Block

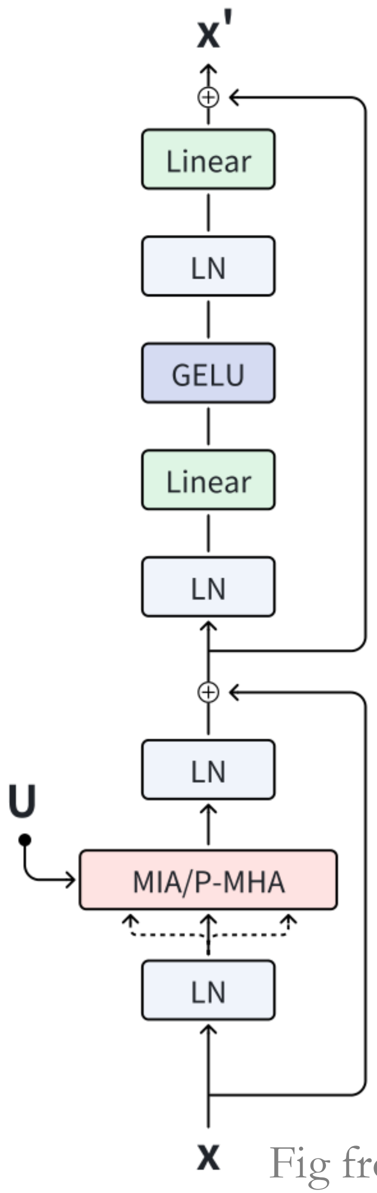


Fig from 2407.08682

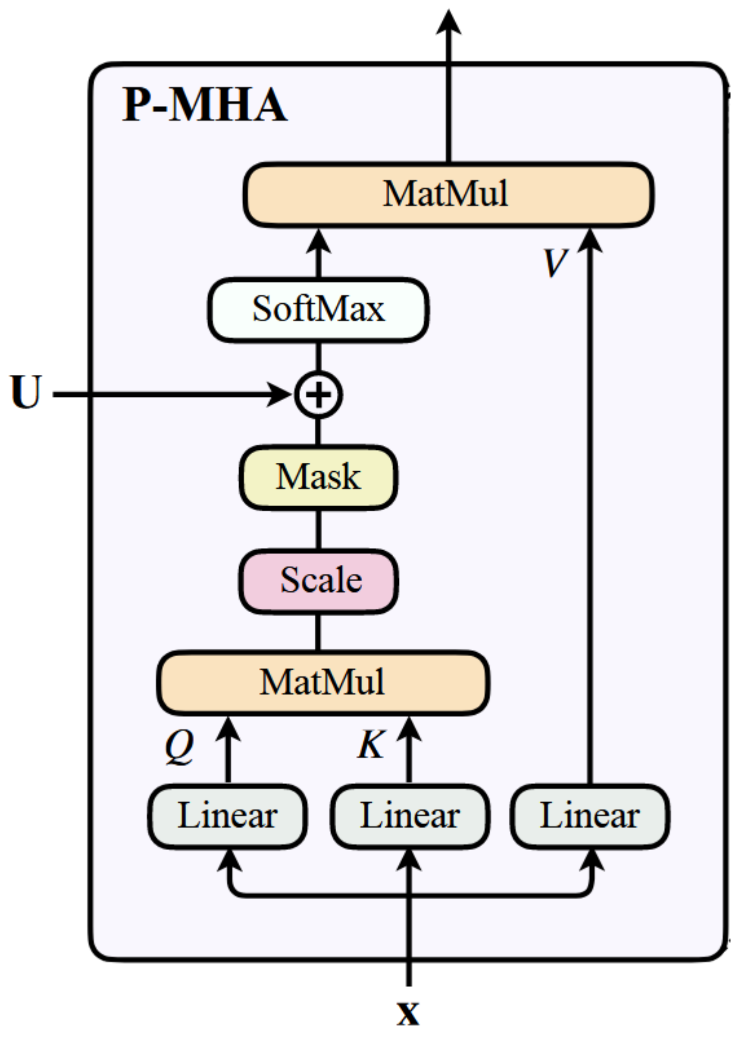


Fig from 2202.03772

$$\text{P-MHA}(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d_k}} + \mathbf{U}\right)V,$$

- The P-MHA is implemented using the PyTorch's *MultiheadAttention* by providing the interaction matrix U as the attn mask input.

- The shape of U is (N,N,C), while both the
- Input x and the output x′ have the same shape (N,C)

- LN represents Layer Normalization

- GELU represents the Gaussian Error Linear Unit activation function

Class Attention Block

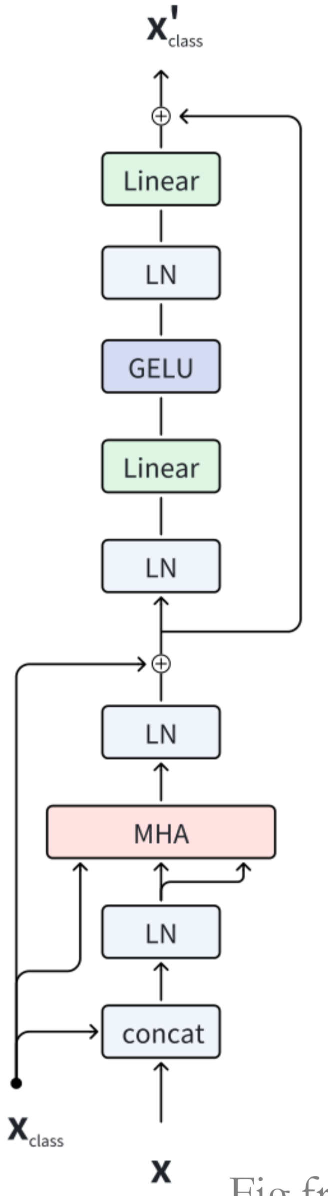


Fig from 2407.08682

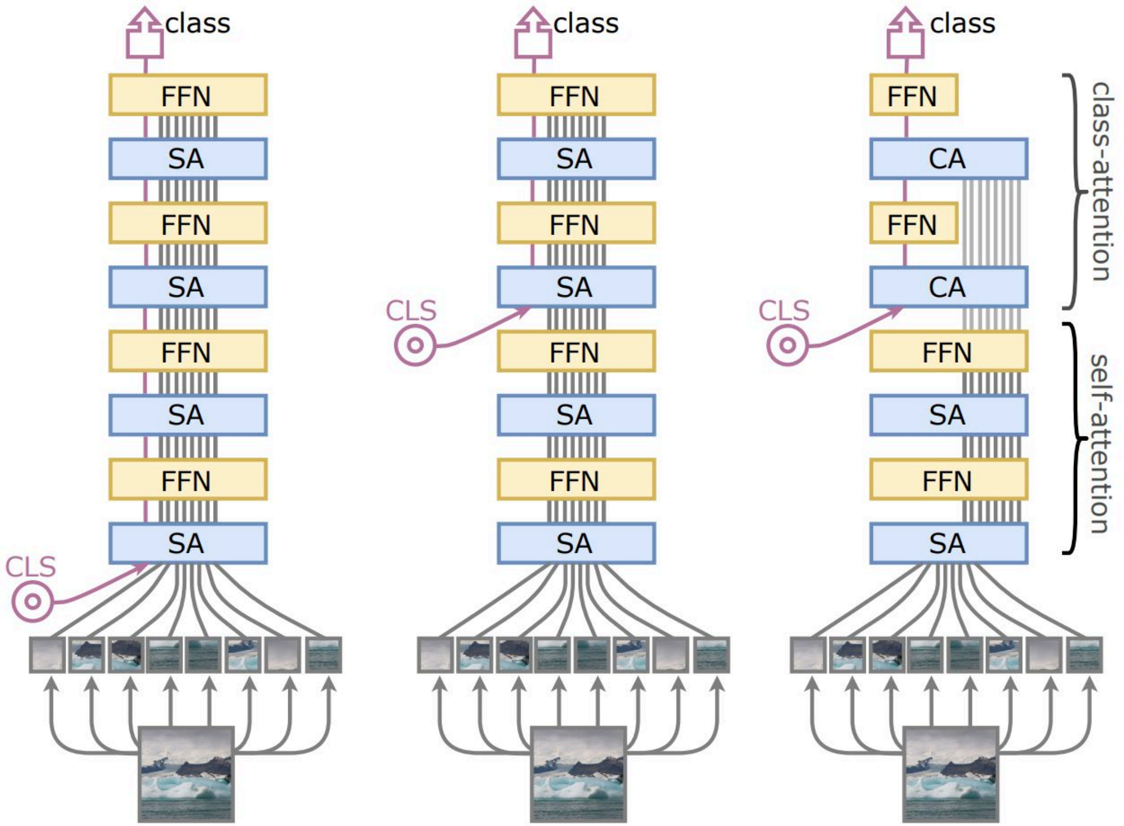


Fig from 2103.17239

ViT Transformer (left): The class embedding (CLS) is added with the patch embeddings from the beginning, which uses the same weights for attention and classification, leading to suboptimal performance.

Improved Approach (middle): Inserting the class embedding (CLS) later in the network after processing the patch embeddings shows better performance.

CaiT Architecture (right): The class embedding (CLS) is added later, with frozen patch embeddings to save computation, and the last layers are dedicated to summarizing information for classification, improving efficiency and performance.

Implementation Details

- Input features for $\mathbf{x}$:
  - 7 kinematic
  - 6 particle identification

- Input features for $\mathbf{U}$:

$$(\ln \Delta, \ln k_T, \ln z, \ln m^2)$$

(particle interaction features)

Category	Variable	TOP	QG	JC
Kinematics	$\Delta\eta$	⋆	⋆	⋆
	$\Delta\phi$	⋆	⋆	⋆
	$\log p_{\mathrm{T}}$	⋆	⋆	⋆
	$\log E$	⋆	⋆	⋆
	$\log p_{\mathrm{T}}/p_{\mathrm{T}}(\mathrm{jet})$	⋆	⋆	⋆
	$\log E/E(\mathrm{jet})$	⋆	⋆	⋆
	ΔR	⋆	⋆	⋆
Particle identification	charge		⋆	⋆
	Electron		⋆	⋆
	Muon		⋆	⋆
	Photon		⋆	⋆
	Charged Hadron		⋆	⋆
	Neutral Hadron		⋆	⋆
Trajectory displacement	$\tanh d_0$			⋆
	$\tanh d_z$			⋆
	σ_{d_0}			⋆
	σ_{d_z}			⋆

2202.03772

$$\Delta = \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2},$$
$$k_T = \min(p_{\mathrm{T},a}, p_{\mathrm{T},b})\Delta,$$
$$z = \min(p_{\mathrm{T},a}, p_{\mathrm{T},b})/(p_{\mathrm{T},a} + p_{\mathrm{T},b}),$$
$$m^2 = (E_a + E_b)^2 - |\mathbf{p}_a + \mathbf{p}_b|^2,$$

- K = 5 MI-particle attention blocks,
- L = 5 particle attention blocks,
- 2 class attention blocks
- D_1 = 64, D_2 = 8
- Trained on an NVIDIA RTX 4090 GPU, using a learning rate of 0.001 and a batch size of 256. Training was limited to 15 epochs to prevent overfitting.

Implementation Details

MIParT

- K = 5 MI-particle attention blocks,
- L = 5 particle attention blocks,
- 2 class attention blocks
- D_1 = 64, D_2 = 8

MIParT-Large

- K = 5 MI-particle attention blocks,
- L = 5 particle attention blocks,
- 2 class attention blocks
- D_1 = 128, D_2 = 8

- **For Top Tagging & Quark-gluon Dataset**

Trained on an NVIDIA RTX 4090 GPU, using a learning rate of 0.001 and a batch size of 256. Training was limited to 15 epochs to prevent overfitting.

- **Pre-trained MIParT-L on 100M JetClass Dataset**

Pre-trained on dual NVIDIA RTX 3090 GPUs using a learning rate of 0.0008 and a batch size of 384, with pre-training limited to 50 epochs to avoid overfitting.

- **Fine-tuned on Top Tagging & Quark-gluon Dataset**

Replaced the last MLP for classification with a newly initialized MLP having two output nodes. All weights were then fine-tuned across the datasets for 20 epochs. We used a learning rate of 0.00016 for the pre-trained weights and 0.008 for the new MLP.

Key Performance Metrics

Accuracy:

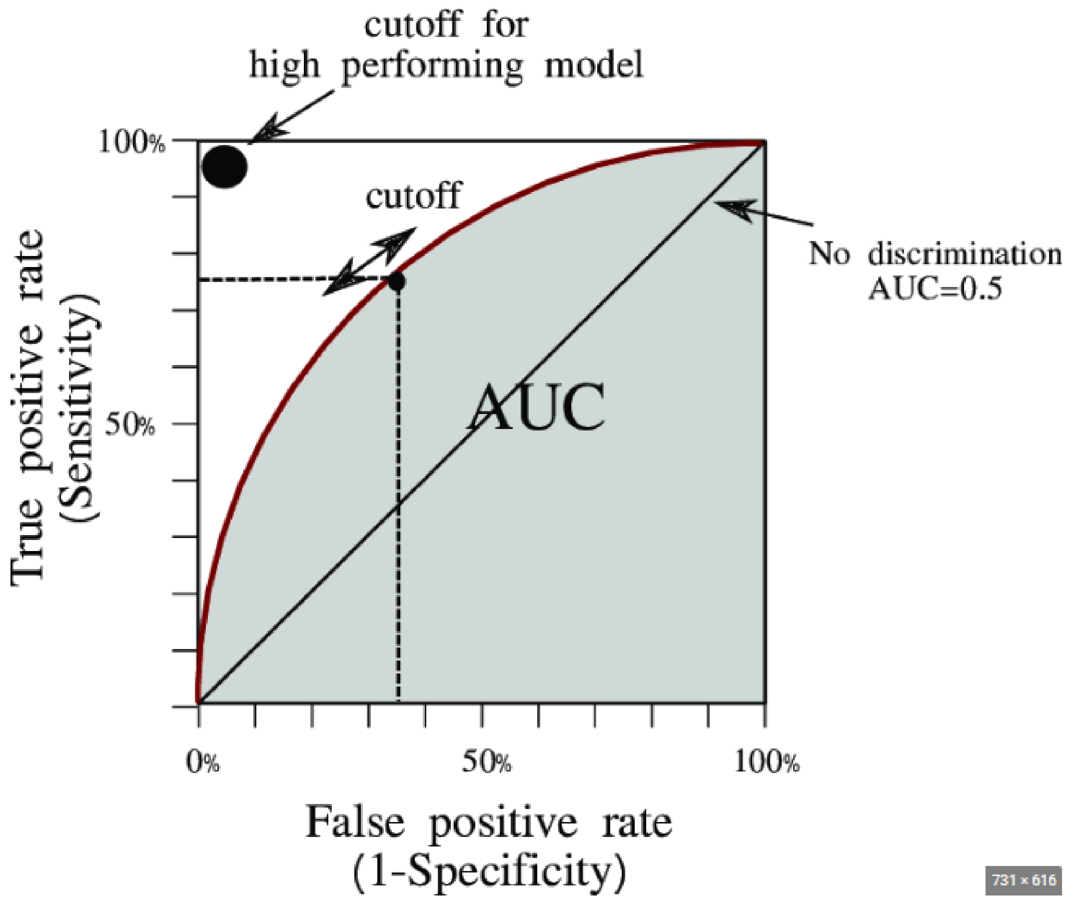
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}\,,$$

Background Rejection at a Certain Signal Efficiency:

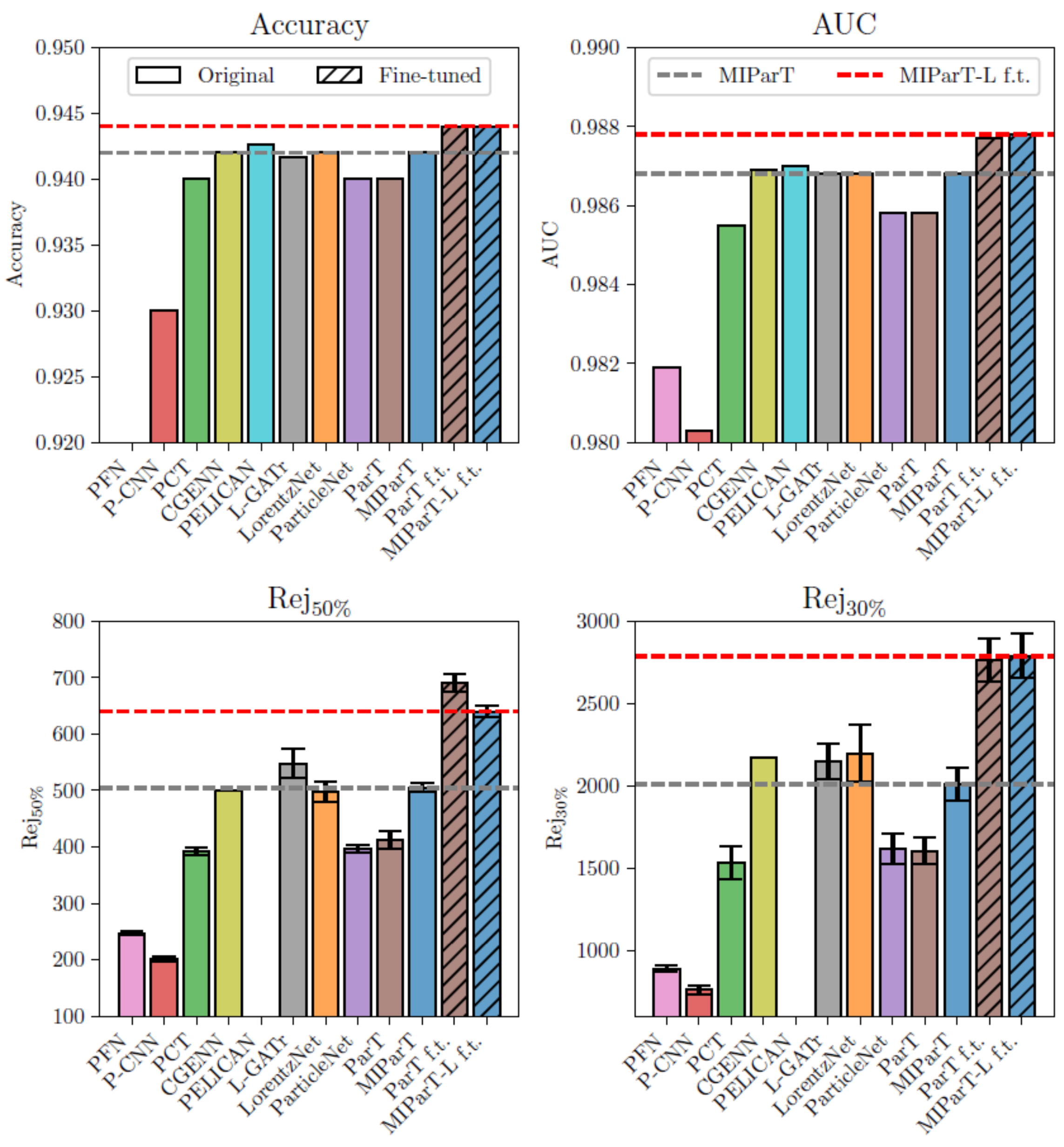
$$\text{Rej}_{X\%} = \frac{1}{\text{FPR}}\bigg|_{\text{TPR}=X\%}$$

For example, a $\text{Rej}_{30\%}$ value of 2500 indicates that at a TPR of 30%, the inverse of the FPR is 2500. This equates to only one false positive for every 2500 negative instances

AUC:



On the Top Tagging Dataset

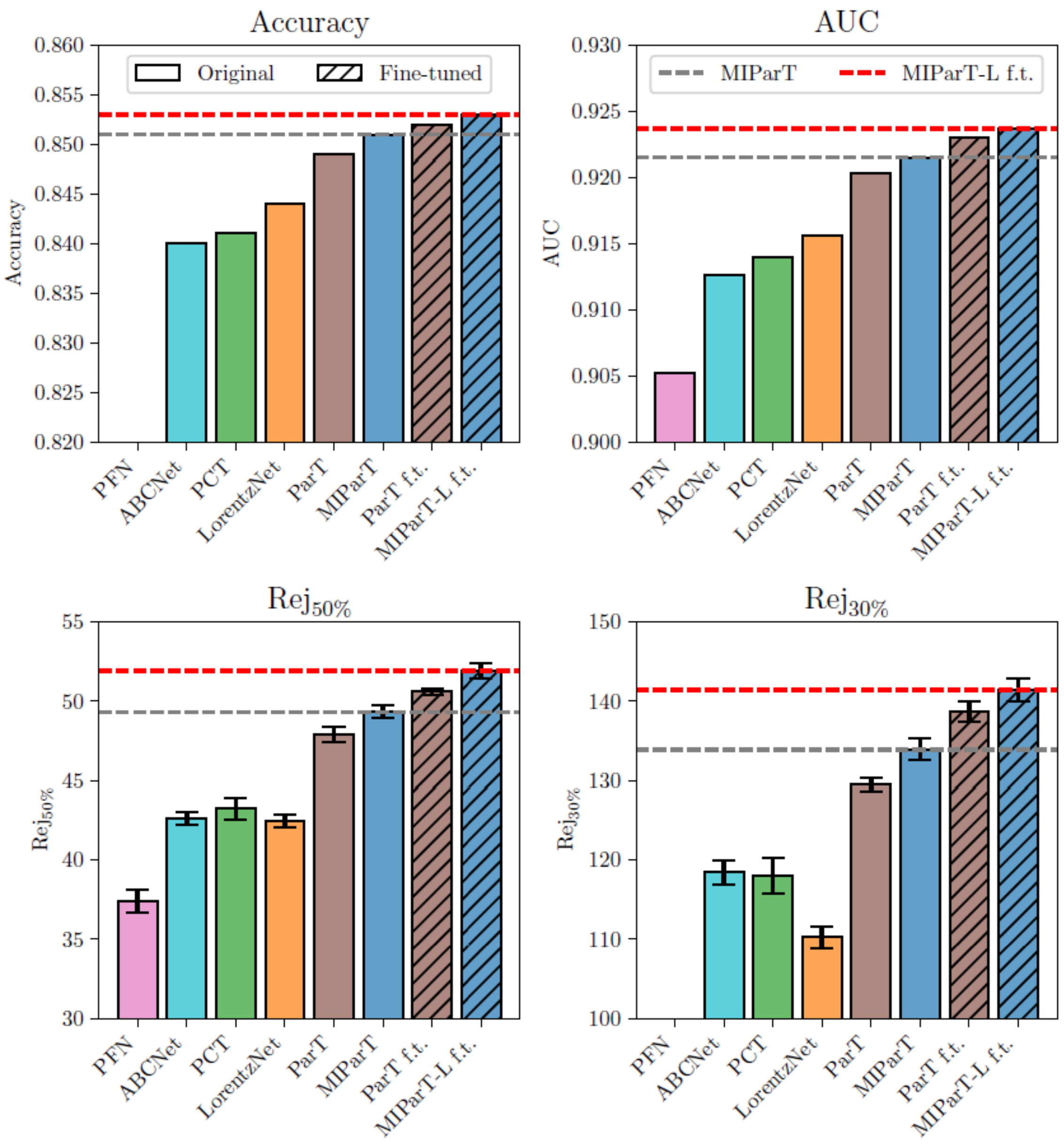


	Accuracy	AUC	Rej$_{50\%}$	Rej$_{30\%}$
PFN	—	0.9819	247±3	888±17
P-CNN	0.930	0.9803	201±4	759±24
PCT	0.940	0.9855	392±7	1533±101
CGENN	0.942	0.9869	500	2172
PELICAN	**0.9426**	**0.9870**	—	—
L-GATr	0.9417	0.9868	**548±26**	2148±106
LorentzNet	0.942	0.9868	498±18	**2195±173**
ParticleNet	0.940	0.9858	397±7	1615±93
ParT	0.940	0.9858	413±16	1602±81
MIParT (ours)	0.942	0.9868	505±8	2010±97
ParT f.t.	**0.944**	0.9877	**691±15**	2766±130
MIParT-L f.t. (ours)	**0.944**	**0.9878**	640±10	**2789±133**

- MIParT achieved accuracy and AUC metrics similar to LorentzNet (Lorentz-equivariant methods) , with comparable Rej50% and Rej30%.
- MIParT outperformed ParT, with 25% better background rejection at 30% signal efficiency.
- MIParT-L (pre-trained on 100M JetClass) showed a 39% improvement in background rejection, matching fine-tuned ParT.

Fig from 2407.08682

On the Quark-gluon Dataset



	Accuracy	AUC	$\mathrm{Rej}_{50\%}$	$\mathrm{Rej}_{30\%}$
PFN	—	0.9052	37.4±0.7	—
ABCNet	0.840	0.9126	42.6±0.4	118.4±1.5
PCT	0.841	0.9140	43.2±0.7	118.0±2.2
LorentzNet	0.844	0.9156	42.4±0.4	110.2±1.3
ParT	0.849	0.9203	47.9±0.5	129.5±0.9
MIParT (ours)	**0.851**	**0.9215**	**49.3±0.4**	**133.9±1.4**
ParT f.t.	0.852	0.9230	50.6±0.2	138.7±1.3
MIParT-L f.t. (ours)	**0.853**	**0.9237**	**51.9±0.5**	**141.4±1.5**

- The MIParT model significantly outperforms LorentzNet across all metrics
- MIParT achieves the best performance across all evaluation metrics, improving background rejection power by approximately 3% compared to ParT.
- MIParT-L (pre-trained on 100M JetClass) showed a 6% improvement in background rejection, surpassing fine-tuned ParT.

Fig from 2407.08682

Performance on different sizes of JetClass

	All classes		$H \to b\bar{b}$	$H \to c\bar{c}$	$H \to gg$	$H \to 4q$	$H \to \ell\nu qq'$	$t \to bqq'$	$t \to b\ell\nu$	$W \to qq'$	$Z \to qq'$
	Accuracy	AUC	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{99.5\%}$	$\text{Rej}_{50\%}$	$\text{Rej}_{50\%}$
ParticleNet (2 M)	0.828	0.9820	5540	1681	90	662	1654	4049	4673	260	215
ParticleNet (10 M)	0.837	0.9837	5848	2070	96	770	2350	5495	6803	307	253
ParticleNet (100 M)	0.844	0.9849	7634	2475	104	954	3339	10526	11173	347	283
ParT (2 M)	0.836	0.9834	5587	1982	93	761	1609	6061	4474	307	236
ParT (10 M)	0.850	0.9860	8734	3040	110	1274	3257	12579	8969	431	324
ParT (100 M)	0.861	0.9877	10638	4149	123	1864	5479	32787	15873	543	402
MIParT-L (2 M)	0.837	0.9836	5495	1940	95	819	1778	6192	4515	311	242
MIParT-L (10 M)	0.850	0.9861	8000	3003	112	1281	3650	16529	9852	440	336
MIParT-L (100 M)	0.861	0.9878	10753	4202	123	1927	5450	31250	16807	542	402

- As the dataset size increases, the performance of the models improves.
- MIParT-L and ParT exhibit nearly identical effectiveness on very large datasets, surpassing that of ParticleNet.

Conclusion

- **On the Top Tagging Dataset:** MIParT model significantly outperformed ParT in the top tagging benchmark, with approximately 25% better background rejection at a 30% signal efficiency.

- **On the Quark-gluon Dataset:** MIParT achieves the best performance across all evaluation metrics, improving background rejection power by approximately 3% compared to ParT.

- MIParT outperformed ParT on both tasks, while requires only 30% of the parameters and 53% of the complexity needed by ParT.
- Fine-tuned MIParT-L improved 39% on top tagging and 6% on quark-gluon, surpassing Fine-tuned ParT.

	TOP	QG	Params	FLOPs
PFN	—		86.1k	4.62M
P-CNN	0.930	—	354k	15.5M
ParticleNet	0.940	—	370k	540M
ParT	0.940	0.849	2.14M	340M
MIParT (ours)	**0.942**	**0.851**	720.9k	180M
MIParT-L f.t. (ours)	**0.944**	**0.853**	2.38M	368M

Thanks！