Representation of Decay Relations in Hyperbolic Space

Boyang Yu
LMU München

Belle II Germany Meeting, 20 Sep 2022

Boyang.Yu@physik.uni-muenchen.de

Goals:

- Prediction of decay channels from final state particles
  -> Tell the branching ratios of different decay modes in a dataset
- Full reconstructions of decay trees

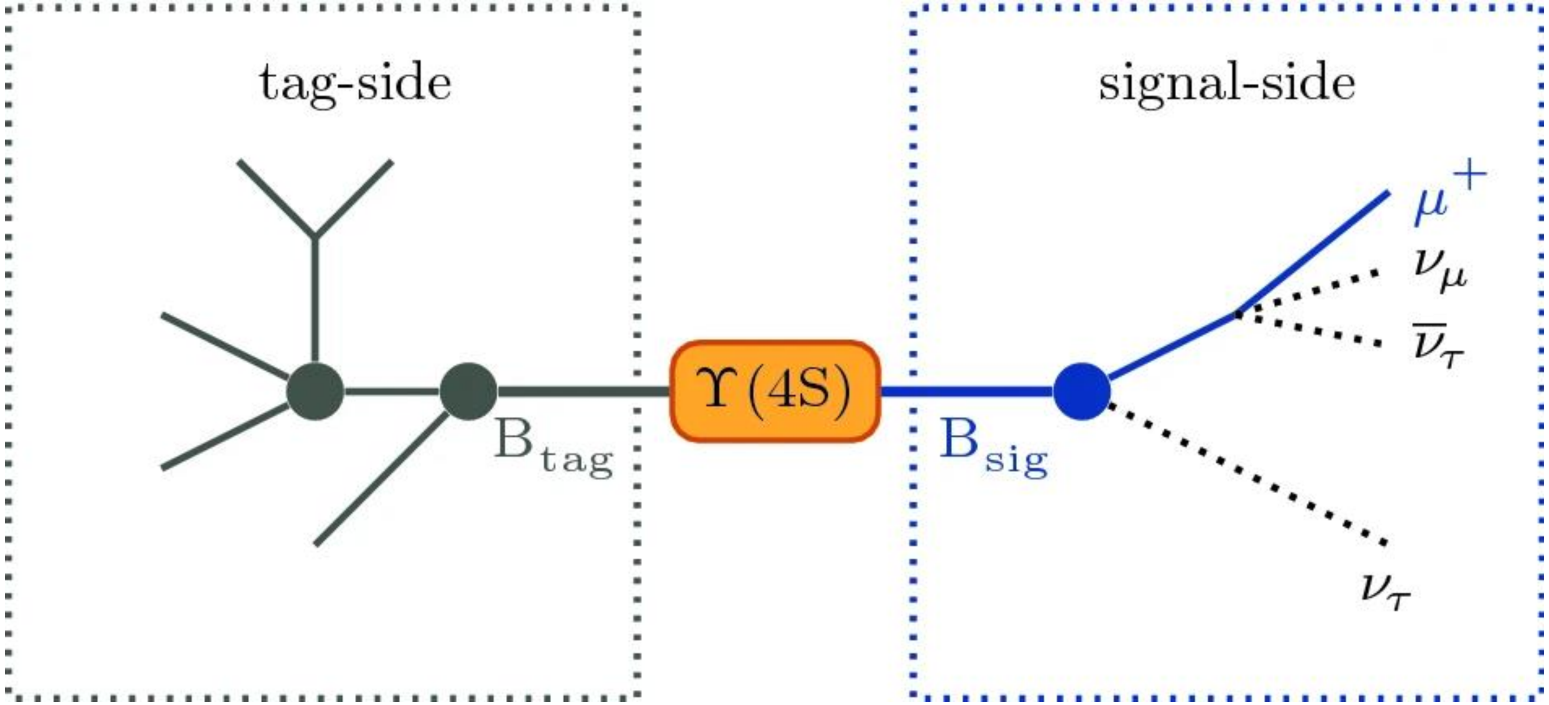
Related work:

- Full Event Interpretation

Limitation of FEI:

- Low tagging efficiency or tag-side efficiency
- Low covered branching fractions

Full Event Interpretation



- Explicitly reconstruct tag side
- Recover the kinematic and flavour information of signal side
- Kernel: Decision Tree to predict reconstructions
  -> Performance strongly restricted by training

Full Event Interpretation

- Low tag-side efficiency (the fraction of correctly tagged Υ(4S) events)

	$B^{\pm}$ (%)	B^{0} (%)
Hadronic	0.76	0.46
Semileptonic	1.80	2.04

- Low covered branching fractions

	Inclusive		Exclusive	
	$B^{\pm}$ (%)	B^{0} (%)	$B^{\pm}$ (%)	B^{0} (%)
Hadronic	9.0	9.8	1.7	1.1
Semileptonic	17.4	15.3	5.2	4.0

Design:

- Create a space to continuously represent all possible decays

  -> not restriced by the channels used in the training

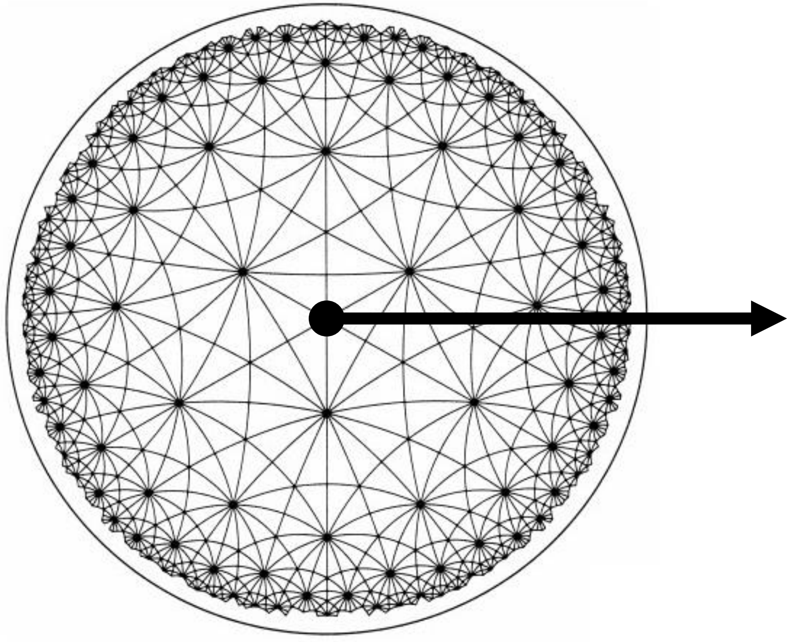
- Encode decay relations in the space
- Tolerant to missing particles

  -> ensure higher efficiency

  -> enable the reconstruction of both B mesons at the same time

- Build dynamics in the space to introduce reconstruction processes

Possible solution: Hyperbolic Space (2D example – Poincare disc)



Properties:

- Rotational symmetry
- Size of an object with distance *d* to the center is proportional to $1 - d^2$
  -> Points will never reach the boundary
  -> Effective space near the boundary is infinite
- Volume of the space scales exponentially with radius

Comparison:

- In Euclidean spaces: Volume grows **polynomially** with radius
- For trees: Number of nodes grows **exponentially** with level

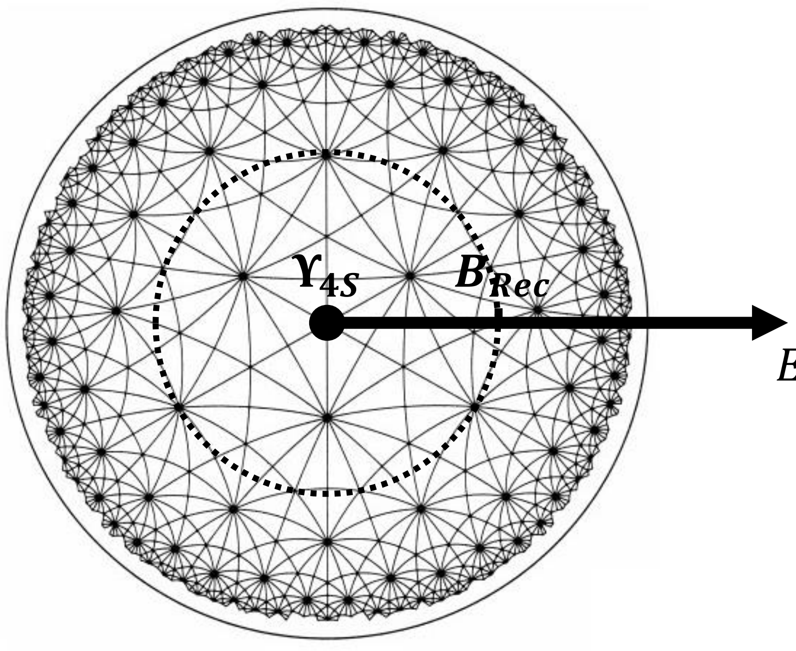
Curves = Straight lines in Poincare disc

$\mathbb{D}^n = \{x \in \mathbb{R}^n : c\|x\|^2 < 1, c \geq 0\}$

$g^{\mathbb{D}} = \lambda_c^2 g^E$

$\lambda_c = \frac{2}{1 - c\|x\|^2}$

Possible solution: Hyperbolic Space (2D example – Poincare disc)



Const. $\quad \sim 1 - Radius^2$

$$E_{ROE} = E_{\Upsilon(4S)} - E_{Reconstructed}$$

- Center: Singularity containing all full reconstructions of $\Upsilon(4S)$ -> Empty rest of event (ROE)
- Bulk points: Partially reconstructed decays
- Points near boundary: Starting points of reconstructions
  -> The less reconstructed, the smaller branching ratio (taking less place in embedded space)
  -> Enable all possible decays

Proof of concept: Toy Monte Carlo

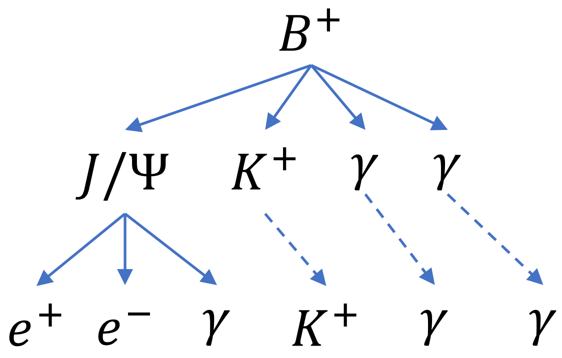
Dataset:

Four channels:

- $B^+ \to (J/\Psi \to e^+e^-)K^+$
- $B^- \to (D^0 \to K^-\pi^+)\,\pi^-$
- $B^+ \to \overline{D^0}\,\pi^+\,\pi^0$
- $B^- \to D^0\,\pi^+\,\pi^-\,\pi^-$

Each event (Y4S Decay) produces several samples according to the depth of particles to its root B meson, e.g.

- Depth 1 (Sample 1)
- Depth 2 (Sample 2)
- Depth 3 (Sample 3)



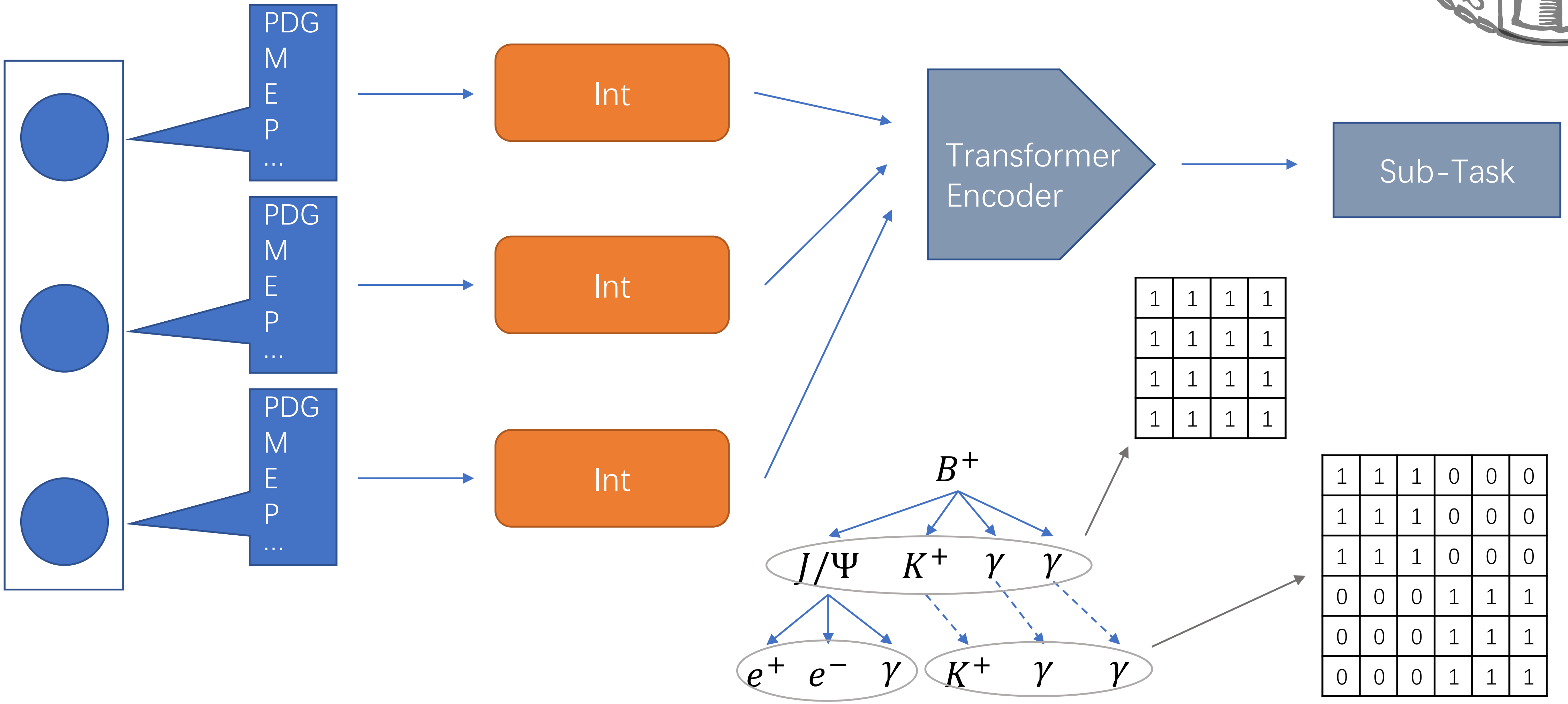
Each particle carries 12 features (**Bold** for reconstruction part)

PDG, mass, charge, energy, production time, x, y, z, **px**, **py**, **pz**, nDaughters

Workflow:

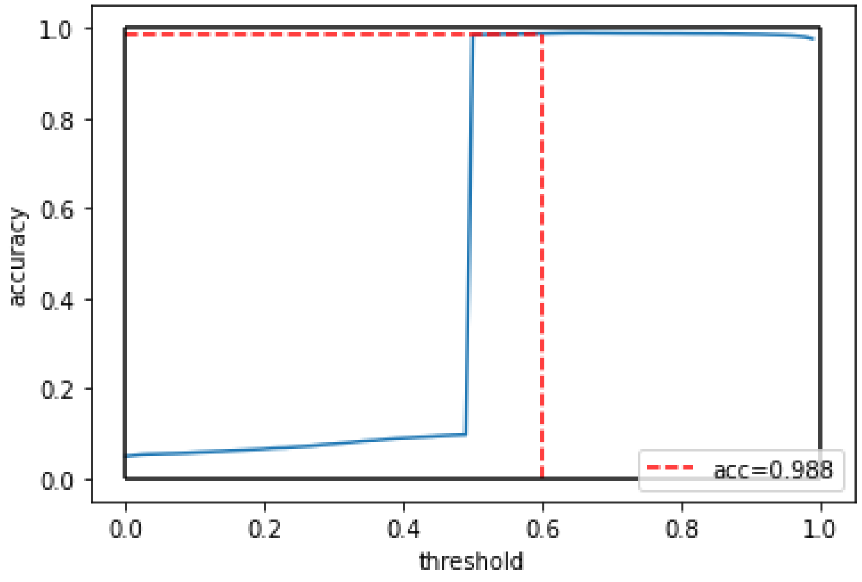
Stage	Neural Networks	Task	Technics	Status
Particle Level Embedding	Automatic Feature Interaction (AutoInt) + Transformer Encoder	Prediction of combinations of daughter particles	Supervised pre-training	Finished on toy MC
Sample Level Embedding	Transformer Encoder + Hyperbolic Embedding (HypTr)	Learning the representation of decays in hyperbolic space	Unsupervised training + Knowledge transfer	Finished on toy MC
Reconstruction	Hyperbolic Transformer Decoder + Generative Adversarial Set Transformer (GAST)	Generation of samples with mother particles	Unsupervised training + Knowledge transfer	On going

Particle Level Embedding:

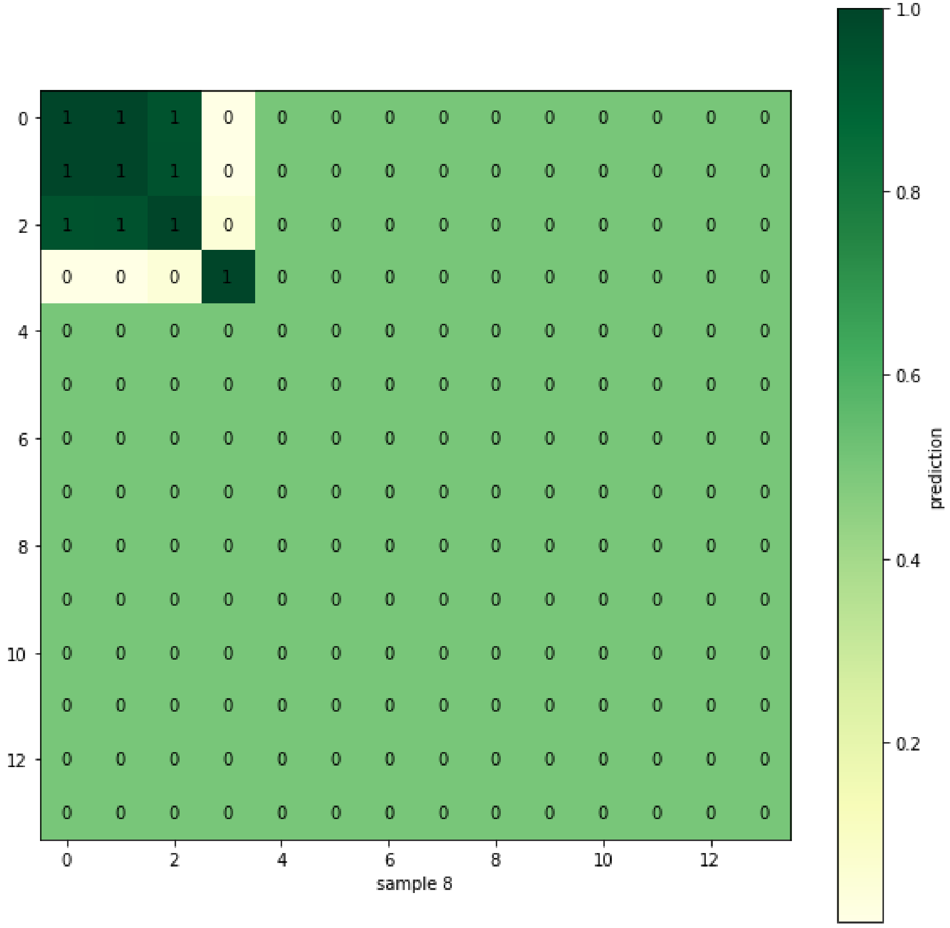


Particle Level Embedding:

Performance on toy MC

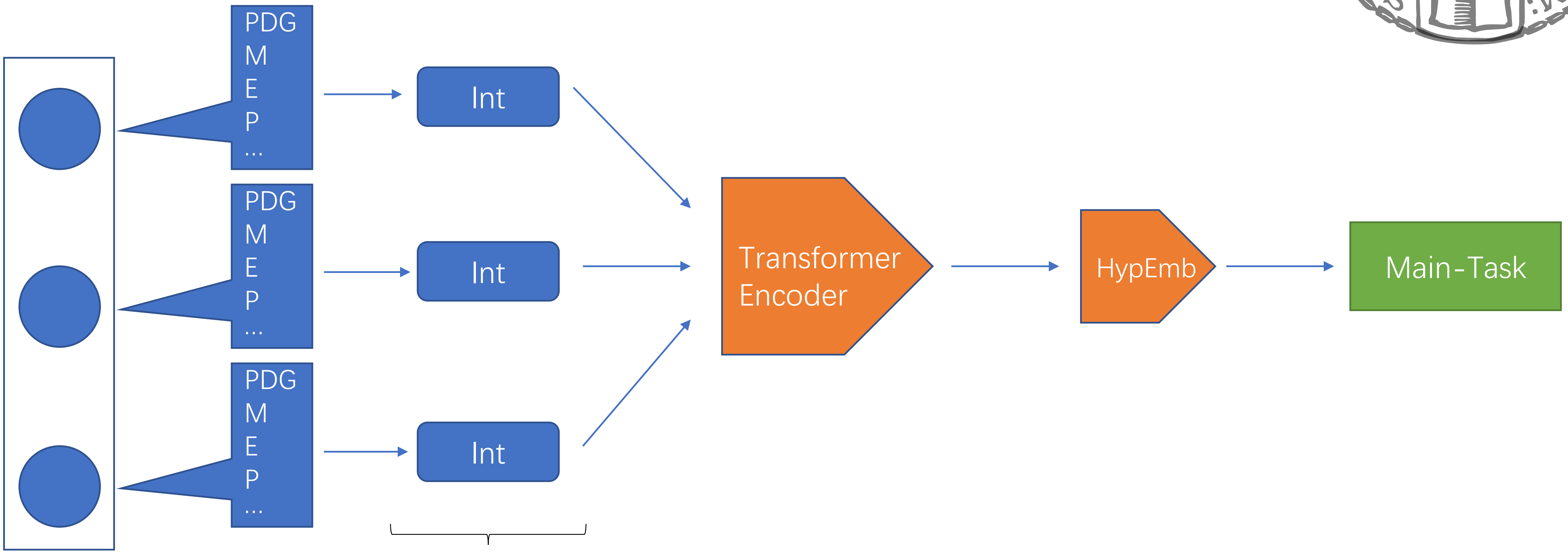


Accuracy



Prediction

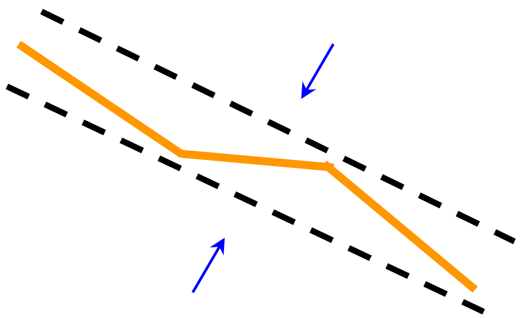
Sample Level Embedding:



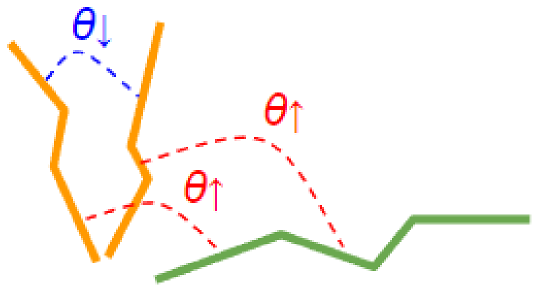
Pre-learned particle level embedding:
Frozen at the beginning of trainings

Sample Level Embedding – Losses:

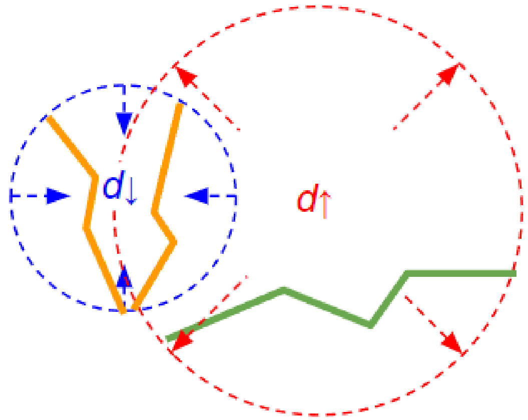
- Intra loss: align the samples from the same decay event, separate otherwise
- Inter loss:
  - Angle loss: minimize the angles between pairs from similar decays (same channel for toy MC), maximize otherwise
  - Distance loss: minimize the hyperbolic distance between pairs from similar decays, maximize otherwise
- Radius loss: encourage the radius of embedded samples to be certain values according to their depths will be replaced by fix radius calculated from E_{ROE} in the future



Intra loss



Inter loss - Angle

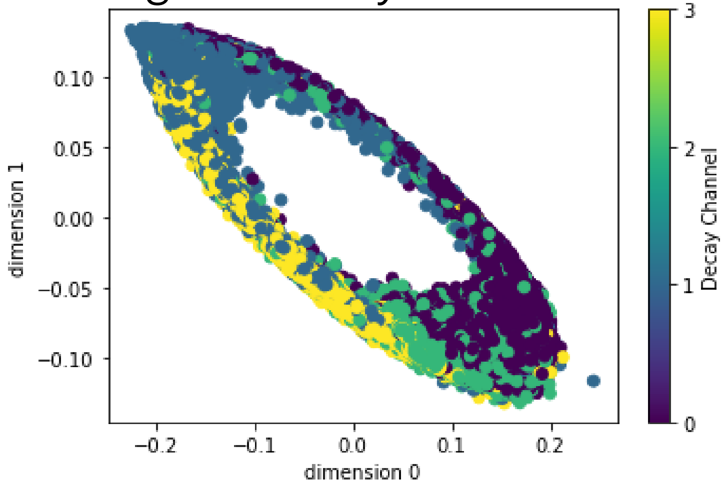


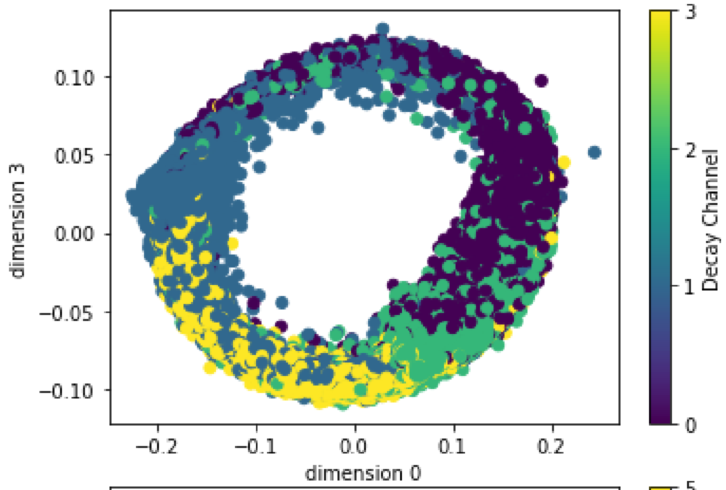
Inter loss - Distance

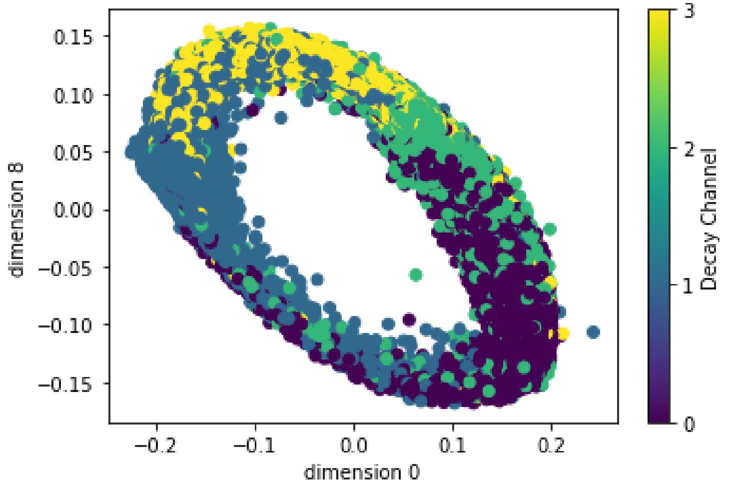
Sample Level Embedding:

Visualisation with 16 dimensional hyperbolic embedding

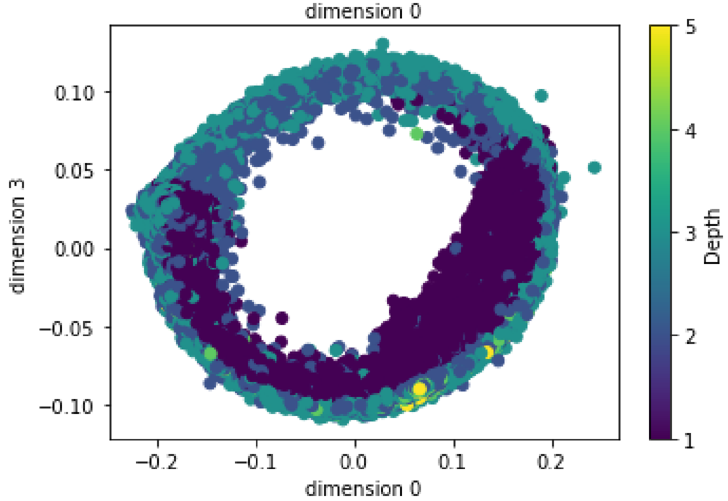
- Clustering vs. Decay channels







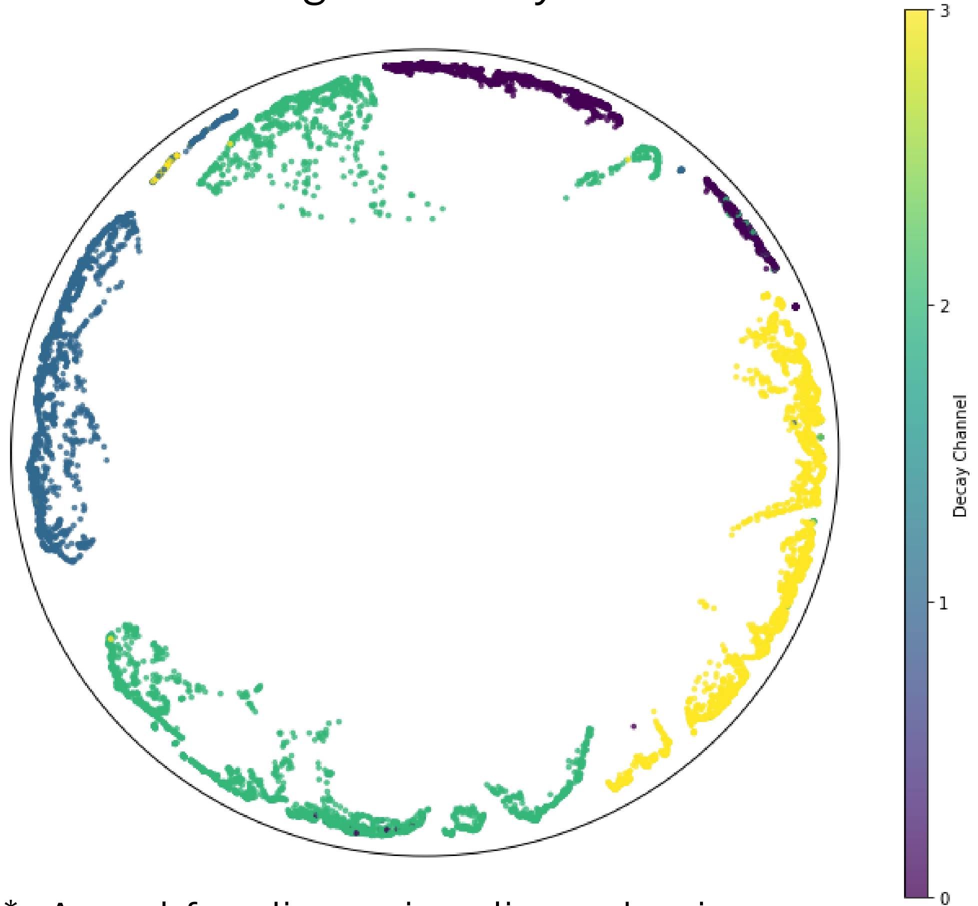
- Clustering vs. Depth



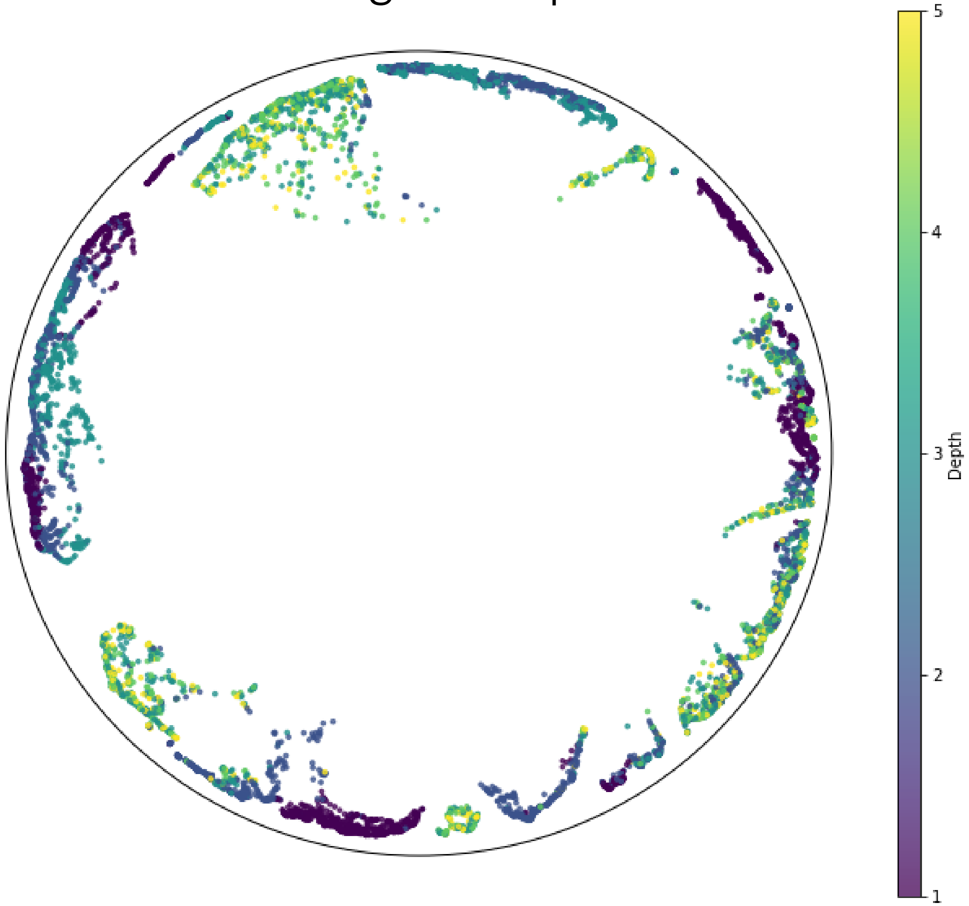
Sample Level Embedding:

Visualisation with UMAP* for 16 dimensional hyperbolic embedding

Clustering vs. Decay channels



Clustering vs. Depth



UMAP*: A tool for dimensionality reduction

Capacities:

Particle Level Embedding
- 12K Parameters

Sample Level Embedding
- 900K Parameters
- 16-D hyperbolic space

In Comparison – Famous Networks using Transformer
- Vision Transformer (small): 85M Parameters
- BERT (small): 110M Parameters
- GPT-3: 175B Parameters
- Hyperbolic Vision Transformer: 22M Parameters, 384-D hyperbolic space

-> Great potential for improvement

Summary:

- Finished the prediction of decay channels from final state particles for toy MC
- Hyperbolic embedding works for the representation of decays

To do:

- Finish the generation part
- Study the necessity of using hyperbolic embedding, i.e. improvement against Euclidean space
- Try with real dataset with general channels
- Test the performance on rare decays

Outlook:

- Once well trained with large dataset, can be used for the reconstruction of any decay channels
- The workflow / well trained networks can also be invested on other HEP projects

Thank You for your Attention

Boyang Yu
Boyang.Yu@physik.uni-muenchen.de
LMU München
Belle II Germany Meeting, 20 Sep 2022

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Reference:

1. T. Keck et al. “The Full Event Interpretation -- An exclusive tagging algorithm for the Belle II experiment”, *arXiv:1807.08680*
2. T. Keck, “The Full Event Interpretation for Belle II”, *IEKP-KA-2014-18*
3. W. Peng et al. “Hyperbolic Deep Neural Networks: A Survey”, *arXiv:2101.04562*
4. A. Vaswani et al. “Attention Is All You Need”, *arXiv:1706.03762*
5. W. Song et al. “AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks”, *arXiv:1810.11921*
6. K. Stelzner et al. “Generative Adversarial Set Transformers”, *Workshop on Object-Oriented Learning at ICML 2020*
7. L. McInnes et al. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”, *arXiv:1802.03426*
8. A. Ermolov et al. “Hyperbolic Vision Transformers: Combining Improvements in Metric Learning”, *arXiv:2203.10833*

Backup

Hyperbolic metrics

Addition:

$$\mathbf{x} \oplus_c \mathbf{y} = \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1 - c\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$$

Distance:

$$D_{hyp}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \operatorname{arctanh}(\sqrt{c}\| - \mathbf{x} \oplus_c \mathbf{y}\|)$$

Exponential:

$$\exp_{\mathbf{x}}^c(\mathbf{v}) = \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{c} \frac{\lambda_{\mathbf{x}}^c \|\mathbf{v}\|}{2} \right) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \right)$$

with x the base point, usually set to 0

Pairwise Cross-Entropy Loss

- Pairwisely calculate hyperbolic distance and euclidical cosine similarity

$$D_{hyp}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \operatorname{arctanh}(\sqrt{c} \| -\mathbf{x} \oplus_c \mathbf{y} \|)$$

$$D_{cos}(\mathbf{z}_i, \mathbf{z}_j) = \left\| \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2} - \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2} \right\|_2^2 = 2 - 2 \frac{\langle \mathbf{z}_i, \mathbf{z}_j \rangle}{\|\mathbf{z}_i\|_2 \cdot \|\mathbf{z}_j\|_2}$$

- Calculate the cross entropy losses w.r.t the two metrics for positive pairs (i, j)

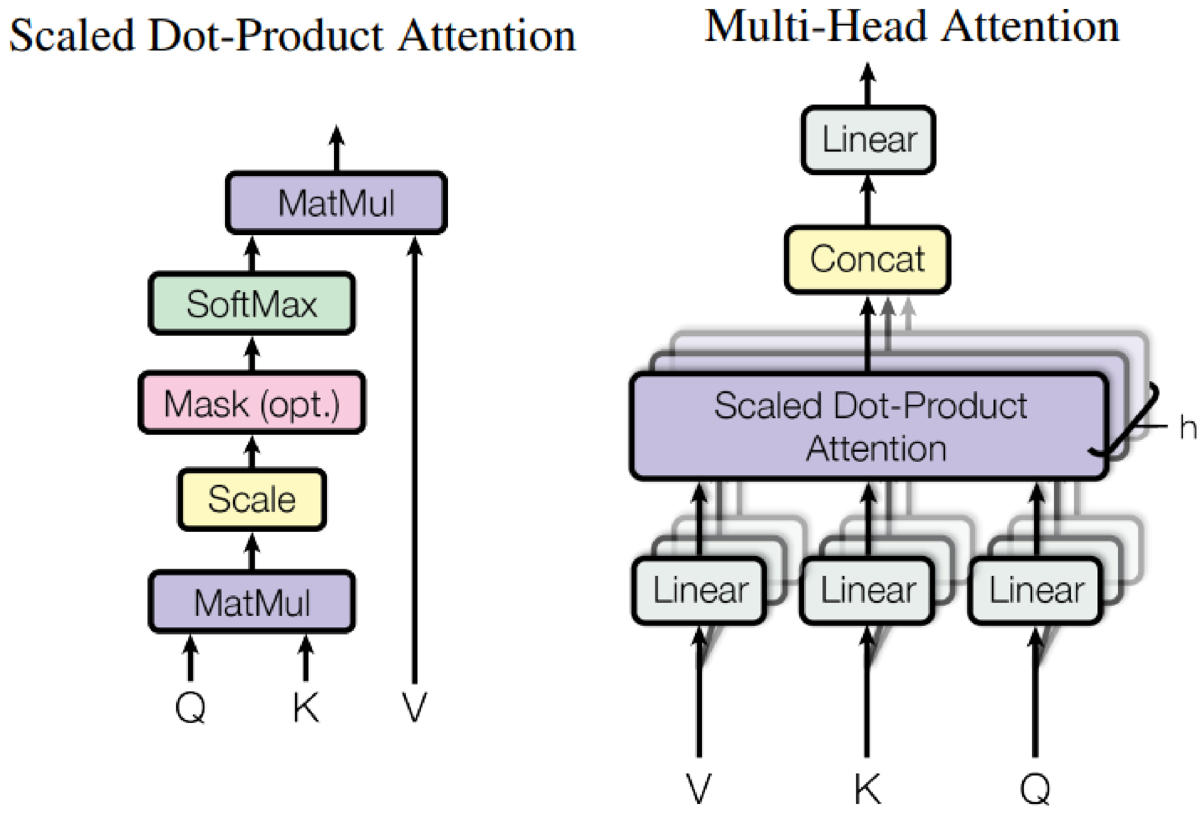
$$l_{i,j} = -\log \frac{\exp\left(-D(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}{\sum_{k=1, k \neq i}^{K} \exp\left(-D(\mathbf{z}_i, \mathbf{z}_k)/\tau\right)}$$

Building Block: Multihead Attention

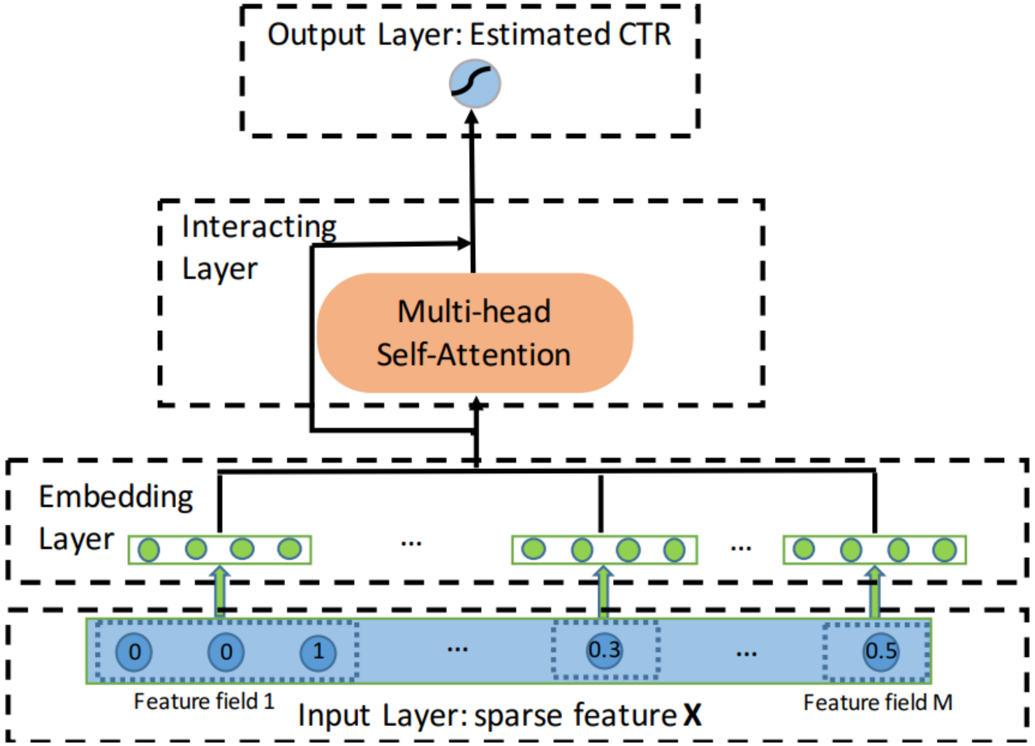
- Inputs and outputs are all vectors
  - Q: Query
  - K: Keys
  - V: Values
- Weights represent the similarity of Q and K
- Attention is reweighted V

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

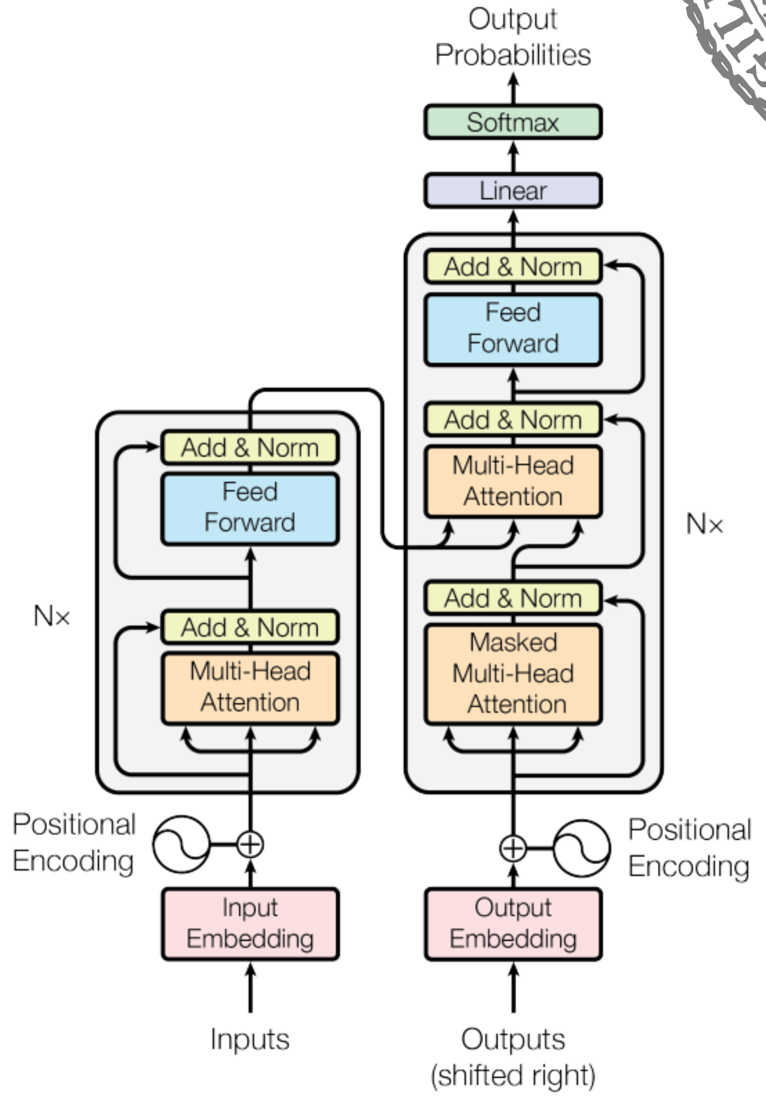
- Multi-Head enables different combinations of the subspaces of the inputs through linear projections



Building Block: Interactor and Transformer



AutoInt: Interactor



Transformer: Encoder Decoder

Reconstruction: Generative Adversarial Set Transformers + Knowledge Transfer