Transformer-based Deep Regression Model for Estimating Missing Transverse Momentum

Seungjin Yang

Kyung Hee University

November 28–30, 2024
KSHEP 2024 Fall

경희대학교
KYUNG HEE UNIVERSITY

Missing Transverse Momentum

- Missing transverse momentum (MET) $\vec{p}_T^{\,\text{miss}}$ is defined as the negative vector sum of the transverse momenta of all reconstructed particle candidates in an event

$$\vec{p}_T^{\,\text{miss}} = - \sum_{i \in \text{event}} \vec{p}_{T,i}$$

- MET serves as a proxy for invisible particles like neutrinos and dark matter candidates
- Good MET reconstruction is important for the Standard Model (SM) process studies involving neutrinos and dark matter searches
- Korea-CMS Machine Learning group aims to develp a deep learning (DL) model that predicts MET!
  - Kyung Hee U.: Junghwan Goh, Junwon Oh and Seungjin Yang
  - Kyungpook National U.: Chang-Seong Moon and Bongho Tae

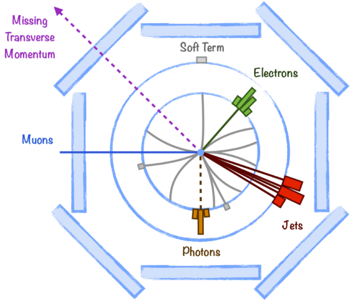
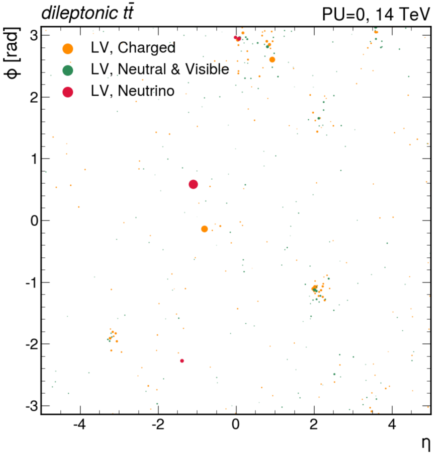


Figure: Source: Bo Liu, Missing Transverse Momentum Measurement using the ATLAS Detector

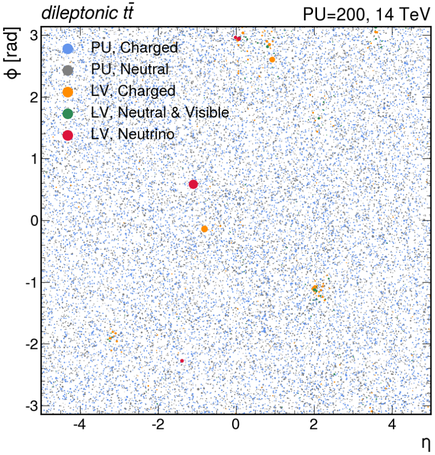
Dataset

- Simulation of *p*-*p* collisions at 14 TeV with an average of 200 pileup interactions
- Dileptonic $t\bar{t}$ with up to two jets at LO
- MadGraph5_aMC@NLO + PYTHIA8 + Delphes



Dataset

- Simulation of *p*-*p* collisions at 14 TeV with an average of 200 pileup interactions
- Dileptonic $t\bar{t}$ with up to two jets at LO
- MadGraph5_aMC@NLO + PYTHIA8 + Delphes



PUPPI

- Pile-Up Per Particle Identification (PUPPI) is a pileup mitigation method built on the CMS Particle Flow (PF) and Charge Hadron Subtraction (CHS) algorithms
- PUPPI gives weights to particles based on the probability that they are originated from a leading vertex (LV) or pileup (PU) vertices
  - LV particles are likely to have more activity around them than PU particles
- PUPPI MET is defined as a MET calculated from PUPPI candidates

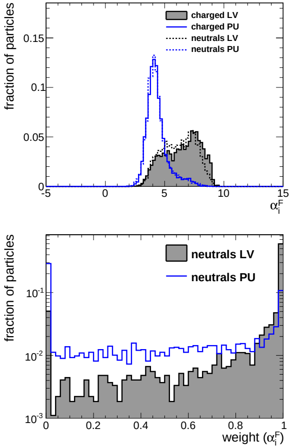


Figure: Source: D. Bertolini, et al. Pileup per particle identification

DeepMET

- CMS has introduced a position-wise feedforward-based MET regression network, called DeepMET
- DeepMET takes in reconstructed particles and then **predicts offsets and scales correcting particles' transverse momenta**
- While MET is an event-level observable, DeepMET solely consists of particle-wise operations, lacking the ability to capture dependencies between particles

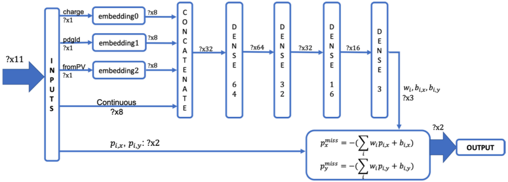
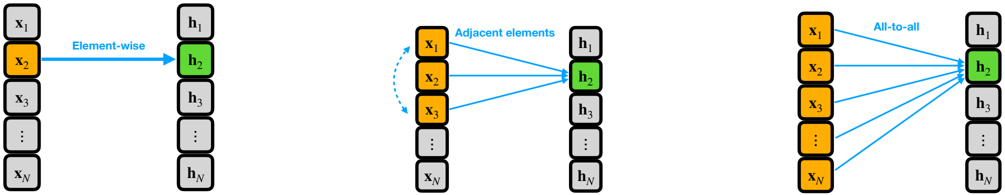


Figure: Source: Y. Feng, A New Deep-Neural-Network–Based Missing Transverse Momentum Estimator, and its Application to W Recoil.

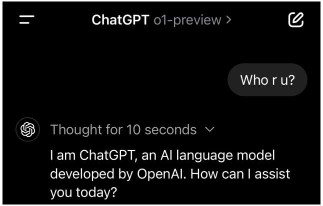
Perceptron, Convolution and Attention



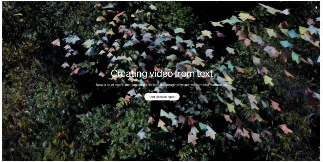
(a) Perceptron **(b)** Convolution **(c)** Attention

- In DL, an input data is represented as an array of vectors
- **Perceptron** acts on each element of the input, ignoring the arrangement of input elements → DeepMET
- **Convolution** computes the weighted sum of adjacent input elements with sliding filters, capturing local patterns in the input data → needs to consider detector resolution and require very deep networks
- **Attention** assigns input-driven weights to input elements, enabling it to **capture both local and global patterns** → Our approach!

Transformer



(a) A dialog with ChatGPT



(b) Sora: Creating video from text

- A transformer is a DL architecture that uses self-attention mechanisms to process and generate sequences, enabling efficient handling of long-range dependencies
- ChatGPT and Sora build on the transformer architecture

Transformer

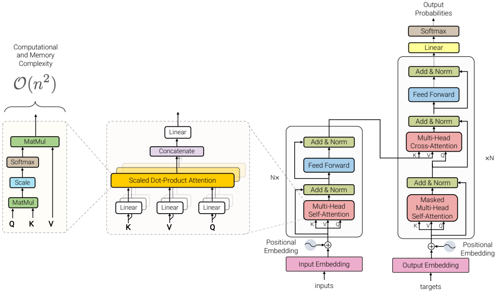
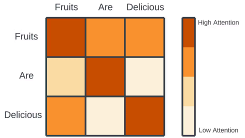


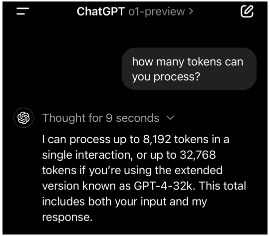
Figure: Credit: Tay, Yi, et al. "Efficient Transformers: A Survey."

- A transformer is a DL architecture that uses self-attention mechanisms to process and generate sequences, enabling efficient handling of long-range dependencies
- ChatGPT and Sora build on the transformer architecture

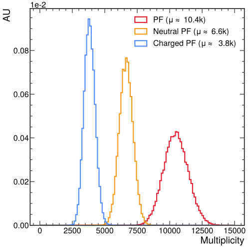
Quadratic Complexity of Attention



(a) Attention matrix



(b) A dialog with ChatGPT



(c) PF candidate multiplicity (14 TeV, PU200)

- An attention has $O(n^2)$ complexity
- To avoid out-of-memory, ChatGPT-4o has about 8k input token limit
- In the hash environment of 200 pileup, a DL model have to deal with an average of 10k particles!

Beyond Transformers

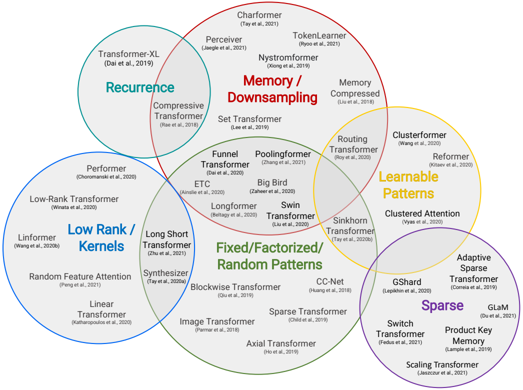


Figure: Source: Tay, Yi, et al. "Efficient Transformers: A Survey."

- Make it sparse, make it block-diagonal, make it small and so on...

Perceiver

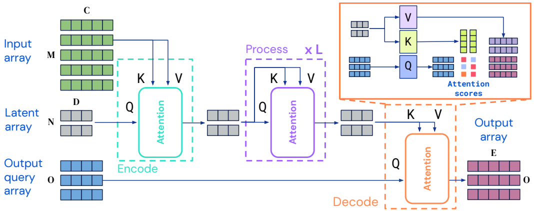
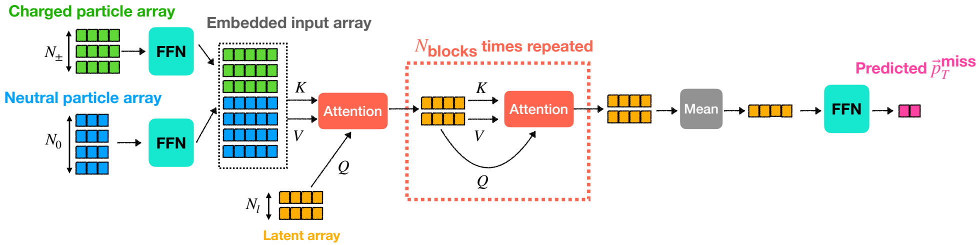


Figure: Source: DeepMind, Perceiver IO: A General Architecture for Structured Inputs & Outputs

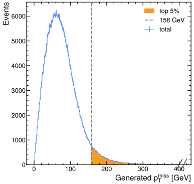
- A Perceiver contains a cross-attention between an input array and a trainable latent array
- The latent array with k latent vectors is assumed to be shorter than the input array with N vectors
- The Perceiver consists of a single $O(kN)$ attention and several $O(k^2)$ attentions
- The Perceiver achieves the state of the art resutls in many data domains

PerceiverMET

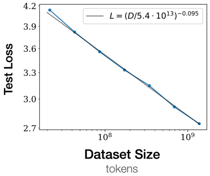


- Charged particle: $(p_x, p_y, \eta, \text{IsRecoPU})$
  - IsRecoPU is a boolean bit indicating whether a particle is associated with a pileup vertex or not
- Neutral particle: (p_x, p_y, η)
- A model with $N_l = 128$ and $N_{\text{blocks}} = 4$ is trained

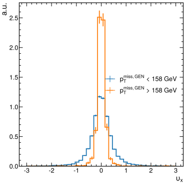
Revisinting Target Variables



(a) p_T^{miss} distribution



(b) Source: J. Kaplan, et al. Scaling Laws for Neural Language Models.

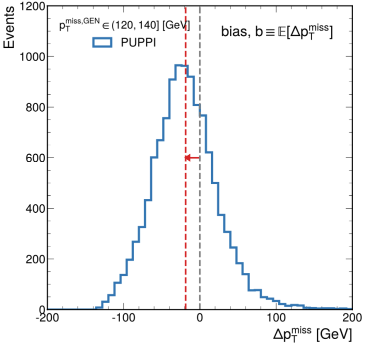


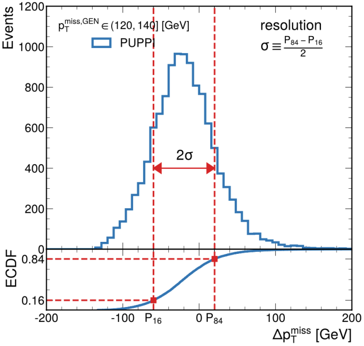
(c) New target variable distribution

- There is huge imbalance in p_T^{miss}
- Unfortunately, the performance of the Transformer follows a power law, where the parameters, dataset size and computations are considered as variables
- Change of target variables from $(p_x^{\text{miss}}, p_y^{\text{miss}})$ to

$$\vec{\upsilon}^* = \frac{\vec{p}_T^{\text{miss,GEN}} - \vec{p}_T^{\text{miss,REC}}}{p_T^{\text{miss,REC}}} = (\upsilon_x, \upsilon_y)$$

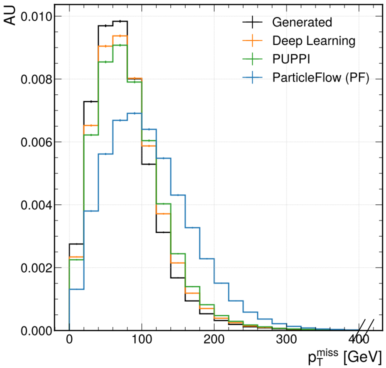
Evaluation Metrics: Bias and Resolution



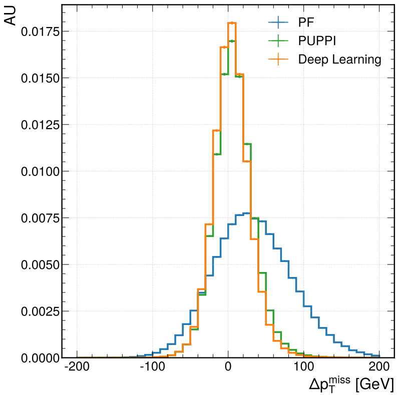


- Residual: $\Delta\mathcal{O} = \mathcal{O}^{\text{REC}} - \mathcal{O}^{\text{GEN}}$
  - $\mathcal{O}$ denotes a component of $\vec{p}_T^{\text{miss}}$
- Bias: $b[\mathcal{O}] = \mathbb{E}[\Delta\mathcal{O}]$
- Resolution: $\sigma[\mathcal{O}] := \frac{P_{84}[\Delta\mathcal{O}] - P_{16}[\Delta\mathcal{O}]}{2}$
  - P_k denotes k-th percentile

Results (1)

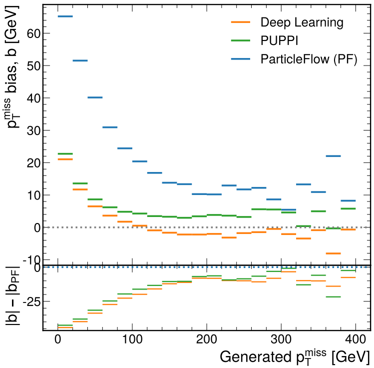


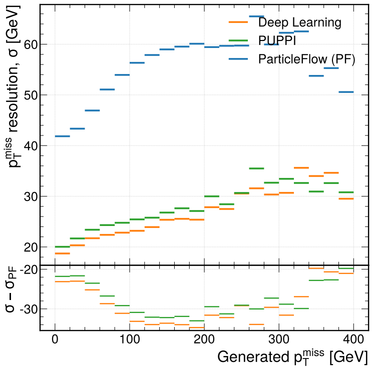
AU
0.010
0.008
0.006
0.004
0.002
0.000
0
100
200
300
400
Generated
Deep Learning
PUPPI
ParticleFlow (PF)
p_T^{miss} [GeV]



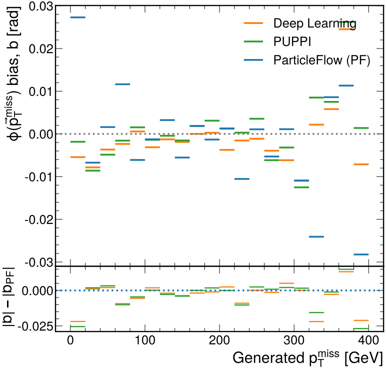
AU
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
-200
-100
0
100
200
PF
PUPPI
Deep Learning
Δp_T^{miss} [GeV]

Results (2)

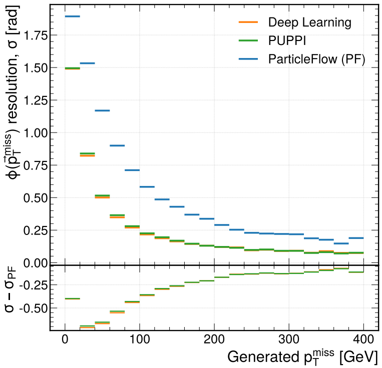

Deep Learning
PUPPI
ParticleFlow (PF)
p_T^{miss} bias, b [GeV]
$|b| - |b_{PF}|$
Generated p_T^{miss} [GeV]


Deep Learning
PUPPI
ParticleFlow (PF)
p_T^{miss} resolution, σ [GeV]
$\sigma - \sigma_{PF}$
Generated p_T^{miss} [GeV]

Results (3)



Deep Learning
PUPPI
ParticleFlow (PF)
0.03
0.02
0.01
0.00
-0.01
-0.02
-0.03
0.000
-0.025
0
100
200
300
400
Generated p_T^{miss} [GeV]
$\phi(\vec{p}_T^{miss})$ bias, b [rad]
$|b| - |b_{PF}|$



Deep Learning
PUPPI
ParticleFlow (PF)
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
0
100
200
300
400
Generated p_T^{miss} [GeV]
$\phi(\vec{p}_T^{miss})$ resolution, σ [rad]
$\sigma - \sigma_{PF}$

Summary

- We aim to develop a attention-mechanism-based MET reconstruction model
- We deploy the Perceiver architecture to incoporate about 10k particles into the attention mechanism
- Perceiver shows smaller bias and resolution than the PUPPI MET
- We plan to refine network architectures and test various physics processes including dark matter candidates

Thanks! :)

PUPPI

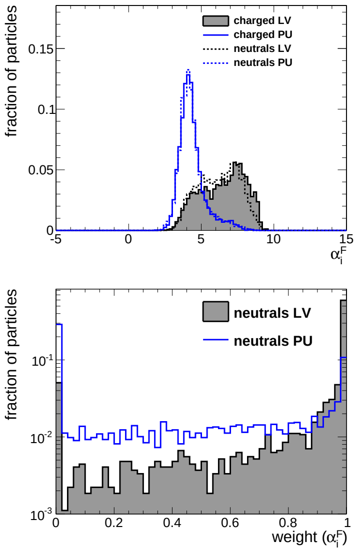
- Calculate a local shape variable α that quantifies how much of a particle i is likely to have originated from parton shower-like radiation (or leading vertex, LV) or pileup-like radiation (or pileup vertices, PU)

$$\alpha_i = \log \sum_{j \in \text{event}} \frac{p_T^j}{\Delta R_{ij}} \times \Theta\left(R_{\min} \leq R_{ij} \leq R_0\right)$$

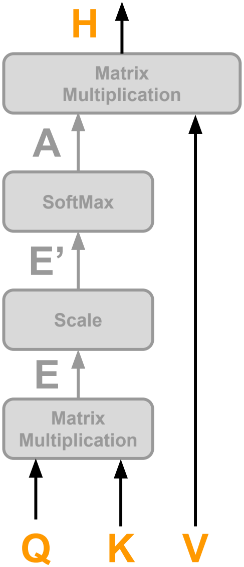
- Estimate a distribution of α_{PU} using charged particles associated with pileup vertices
- Calculate signed χ^2 for each neutral particle

$$\chi_i^2 = \frac{(\alpha_i - \bar{\alpha}_{PU})|\alpha_i - \bar{\alpha}_{PU}|}{\text{RMS}[\alpha_{PU}]}$$

- Calculate weights for neutral particles: $w_i = F_{\chi^2,\text{NDF}=1}(\chi_i^2)$
- Update neutral particles' momenta: $p_i \to w_i \times p_i$
- Remove neutral particles with w_i and $p_{T,i}$ less than thresholds



Scaled Dot-Product Attention (0)



- An array of vectors can be represented as a matrix

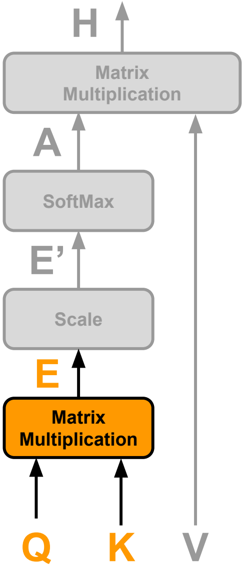
$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \end{bmatrix}$$

- An attention function takes three matrices K, V and Q as input and produces a single matrix. In general, K and V are originated from the same matrix X

$$\begin{aligned} H &= Attention(K, V, Q) \\ &= Attention(\phi_K(X), \phi_V(X), Q) \end{aligned}$$

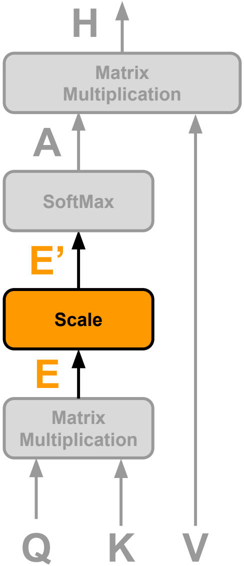
- A self-attention is a special case of attention, where Q is also came from X

Scaled Dot-Product Attention (1)



$$
\begin{aligned}
E &= QK^T \\
&= \begin{bmatrix} \vec{q}_1 \\ \vec{q}_2 \\ \vec{q}_3 \end{bmatrix} \begin{bmatrix} \vec{k}_1 & \vec{k}_2 & \vec{k}_3 \end{bmatrix} \\
&= \begin{bmatrix} \vec{q}_1 \cdot \vec{k}_1 & \vec{q}_1 \cdot \vec{k}_2 & \vec{q}_1 \cdot \vec{k}_3 \\ \vec{q}_2 \cdot \vec{k}_1 & \vec{q}_2 \cdot \vec{k}_2 & \vec{q}_2 \cdot \vec{k}_3 \\ \vec{q}_3 \cdot \vec{k}_1 & \vec{q}_3 \cdot \vec{k}_2 & \vec{q}_3 \cdot \vec{k}_3 \end{bmatrix} .
\end{aligned}
$$

Scaled Dot-Product Attention (2)



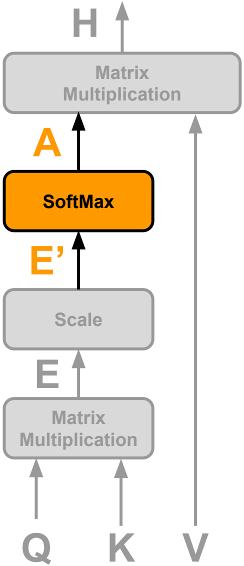
$$E = QK^T$$

$$= \begin{bmatrix} \vec{q}_1 \\ \vec{q}_2 \\ \vec{q}_3 \end{bmatrix} \begin{bmatrix} \vec{k}_1 & \vec{k}_2 & \vec{k}_3 \end{bmatrix}$$

$$= \begin{bmatrix} \vec{q}_1 \cdot \vec{k}_1 & \vec{q}_1 \cdot \vec{k}_2 & \vec{q}_1 \cdot \vec{k}_3 \\ \vec{q}_2 \cdot \vec{k}_1 & \vec{q}_2 \cdot \vec{k}_2 & \vec{q}_2 \cdot \vec{k}_3 \\ \vec{q}_3 \cdot \vec{k}_1 & \vec{q}_3 \cdot \vec{k}_2 & \vec{q}_3 \cdot \vec{k}_3 \end{bmatrix}.$$

$$E' = \frac{1}{\sqrt{dim(\vec{k})}} E.$$

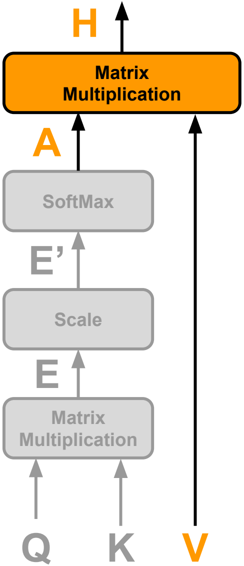
Scaled Dot-Product Attention (3)



$$A = \begin{bmatrix} \frac{e^{E'_{11}}}{Z_1} & \frac{e^{E'_{12}}}{Z_1} & \frac{e^{E'_{13}}}{Z_1} \\ \frac{e^{E'_{21}}}{Z_2} & \frac{e^{E'_{22}}}{Z_2} & \frac{e^{E'_{23}}}{Z_2} \\ \frac{e^{E'_{31}}}{Z_3} & \frac{e^{E'_{32}}}{Z_3} & \frac{e^{E'_{33}}}{Z_3} \end{bmatrix},$$

where $Z_j = \sum_j e^{E'_{ij}}$.

Scaled Dot-Product Attention (4)



$$H = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vec{v}_3 \end{bmatrix}$$

$$= \begin{bmatrix} \sum_i A_{1i}\vec{v}_i \\ \sum_i A_{2i}\vec{v}_i \\ \sum_i A_{3i}\vec{v}_i \end{bmatrix} .$$