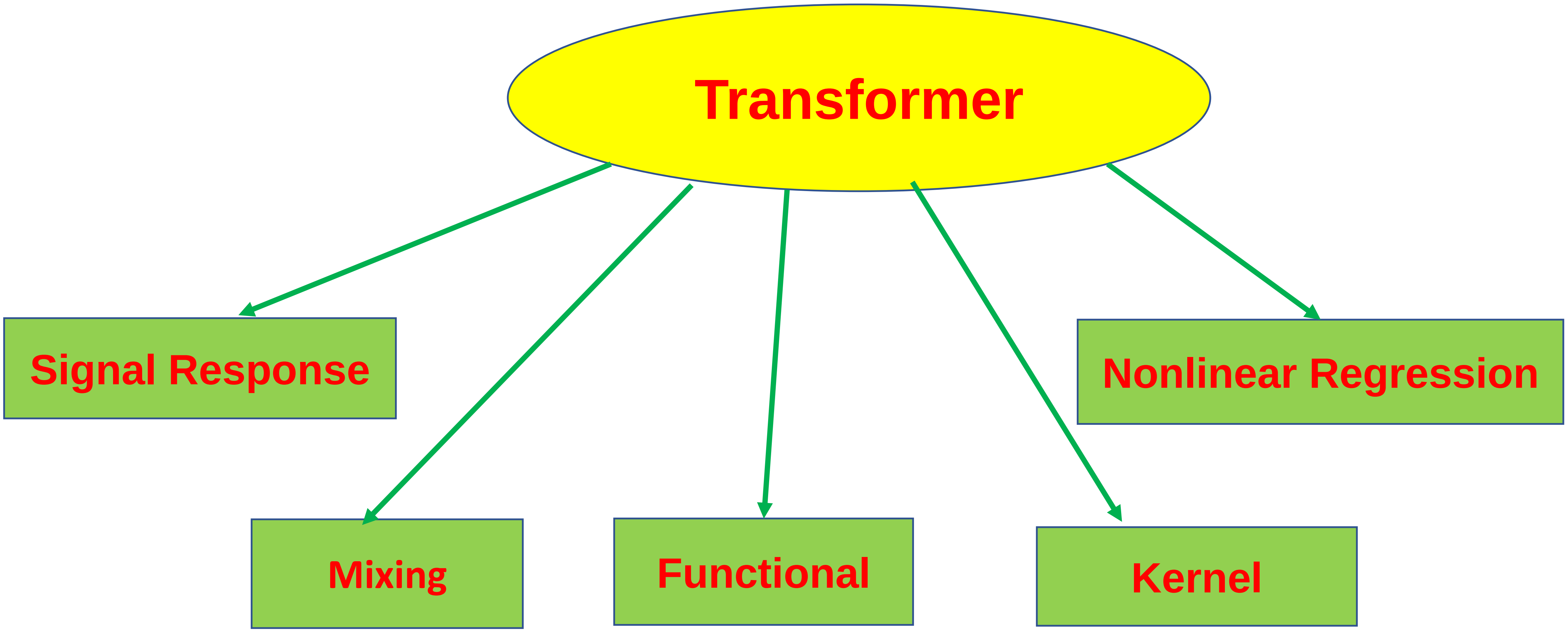
Manifold Learning and Artificial Intelligence
Lecture 13
Mixed Transformer
Spectral and Space Transformer

Momiao Xiong, University of Texas School of Public Health

- Time: 10:00 pm, US East Time, 04/22/2023
- 10:00 am, Beijing Time. 04/23/2023

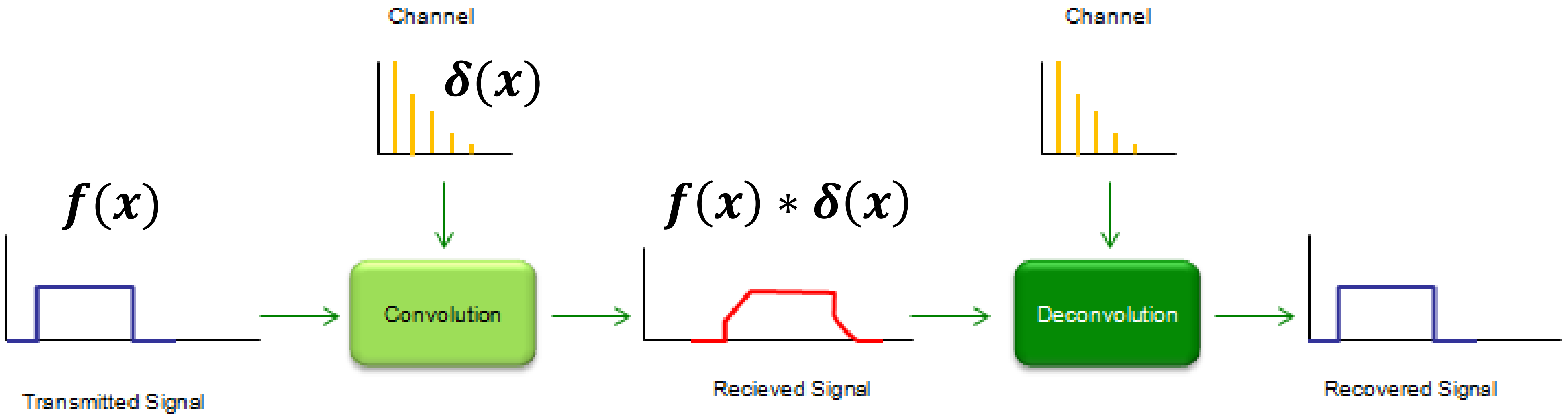
Github Address: https://ai2healthcare.github.io/

- Theoretic foundation of transformer
- View transformer as response of system
- View transformer as nonlinear regression
- Kernel transformer
- Generalized Fourier Integral Theorems and their applications to transformer
- Functional Model
- Mixing MLP



Transformer
Signal Response
Nonlinear Regression
Mixing
Functional
Kernel

13.1 Signal Response



$$h = f * \delta + \epsilon,\ f(x,y) * \delta(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)\delta(\text{x-m,y-n})$$

$$f(x) * \delta(x) = \int_{-\infty}^{\infty} f(u)\delta(x-u)du = \sum_{m=0}^{M-1} f(m)\delta(x-m)$$

$$f(x,y) * \delta(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)\delta(x-u,y-v)dudv$$

Causal Model

$$h(x) = f(x) * \delta(x) + \varepsilon$$

$$x \perp\!\!\!\perp \varepsilon$$

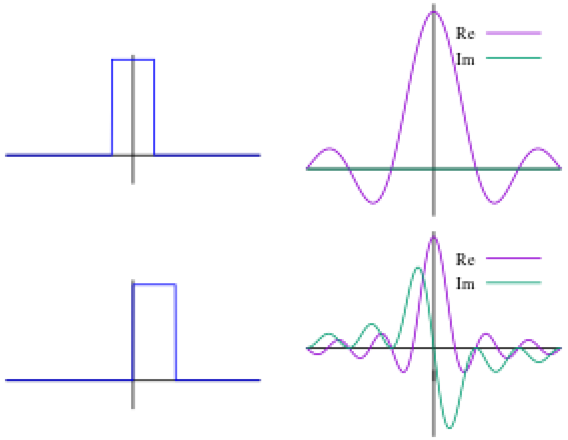
13.2. Fourier Transform

- **Fourier Transform**

Continuous Function

$$\hat{f}(s) = \int_{-R^D}^{R^D} f(x)e^{-i2\pi s^T x} dx, x \in R^D, s \in R^D \quad \text{(F1)}$$

$$\hat{f}(u,v) = \int_{-R^D}^{R^D} f(x,y)e^{-i2\pi(u^T x + v^T y)} dxdy \quad \text{(F2)}$$



Discrete Function

$$\hat{f}(s) = \sum_{n=-\infty}^{n=\infty} f(x_n)e^{-i2\pi s^T x_n}$$

$$\hat{f}(u,v) = \sum_{n=-\infty}^{n=\infty} \sum_{n=-\infty}^{n=\infty} f(x_n, y_m)e^{-i2\pi(u^T x_n + v^T y_m)}$$

13.2. Fourier Transform

- **Inverse Fourier Transform**

Continuous Function

$$f(x) = \int_{-R^D}^{R^D} \hat{f}(s) e^{i2\pi s^T x} ds \quad \text{(F3)}$$

$$f(x, y) = \int_{-R^D}^{R^D} \hat{f}(u, v) e^{i2\pi(u^T x + v^T y)} du dv \quad \text{(F4)}$$

Discrete Function

$$f(x) = \sum_{n=-\infty}^{n=\infty} \hat{f}(s_n) e^{-i2\pi x^T s_n}$$

$$f(x, y) = \sum_{n=-\infty}^{n=\infty} \sum_{n=-\infty}^{n=\infty} \hat{f}(u_n, v_m) e^{-i2\pi(x^T u_n + y^T v_m)}$$

13.3. Convolution and Fourier Transform

$$\hat{h}(s) = \hat{f}(s)\hat{\delta}(s)$$

$$h = f * \delta + \epsilon$$

$$\hat{h}(x) = F^{-1}(\hat{f}(s)\hat{\delta}(s))$$

F^{-1}: Inverse Foureier Transform

$$f(x, y) * \delta(x, y) \;<=>\; \hat{f}(u, v)\hat{\delta}(u, v)$$

$$f(x, y) * \delta(x, y) = F^{-1}\{\hat{f}(u, v)\hat{\delta}(u, v)\}$$

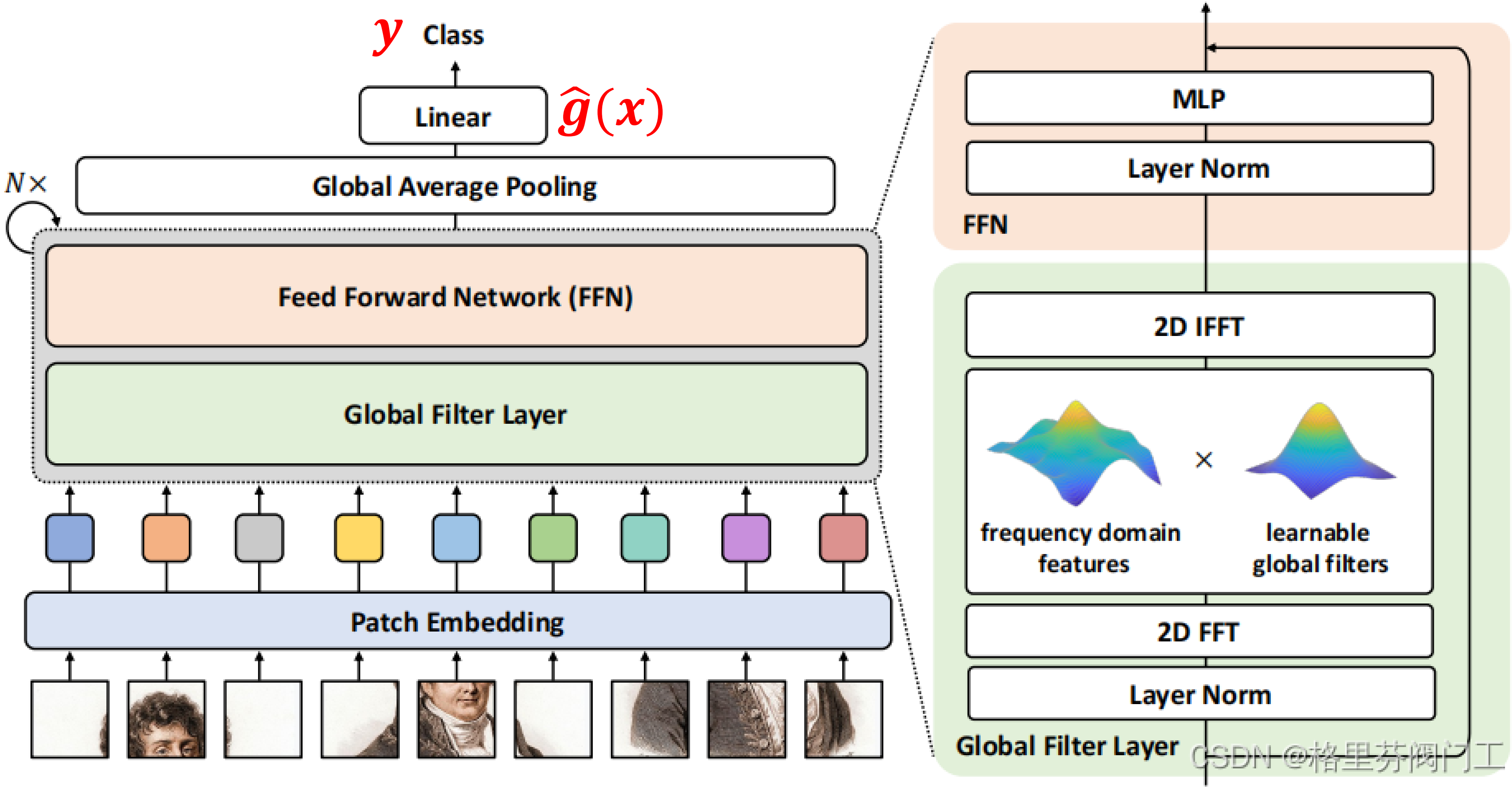
Causal Test

$$\varepsilon = h(x) - \hat{h}(x)$$

$$\varepsilon \perp\!\!\!\perp x$$

13.4. Global Filter Networks for Image Classification

Code is available at https://github.com/raoyongming/GFNet.



Causal Test

$$\varepsilon = y - \widehat{g}(x)$$
$$\varepsilon \perp\!\!\!\perp x$$

Yongming Rao et al. 2021

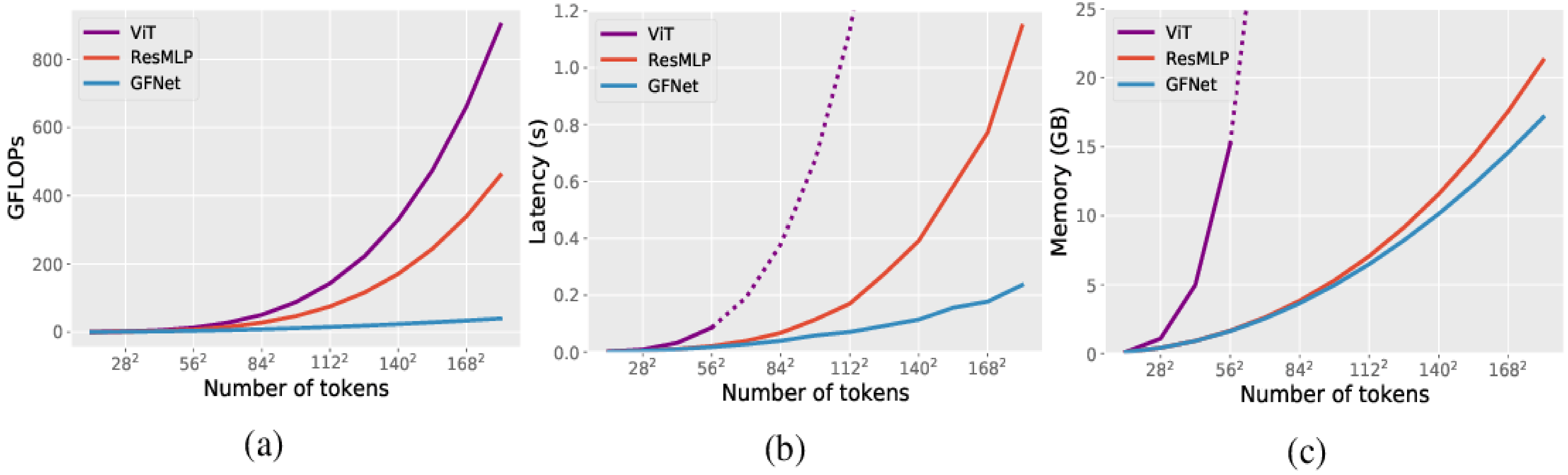
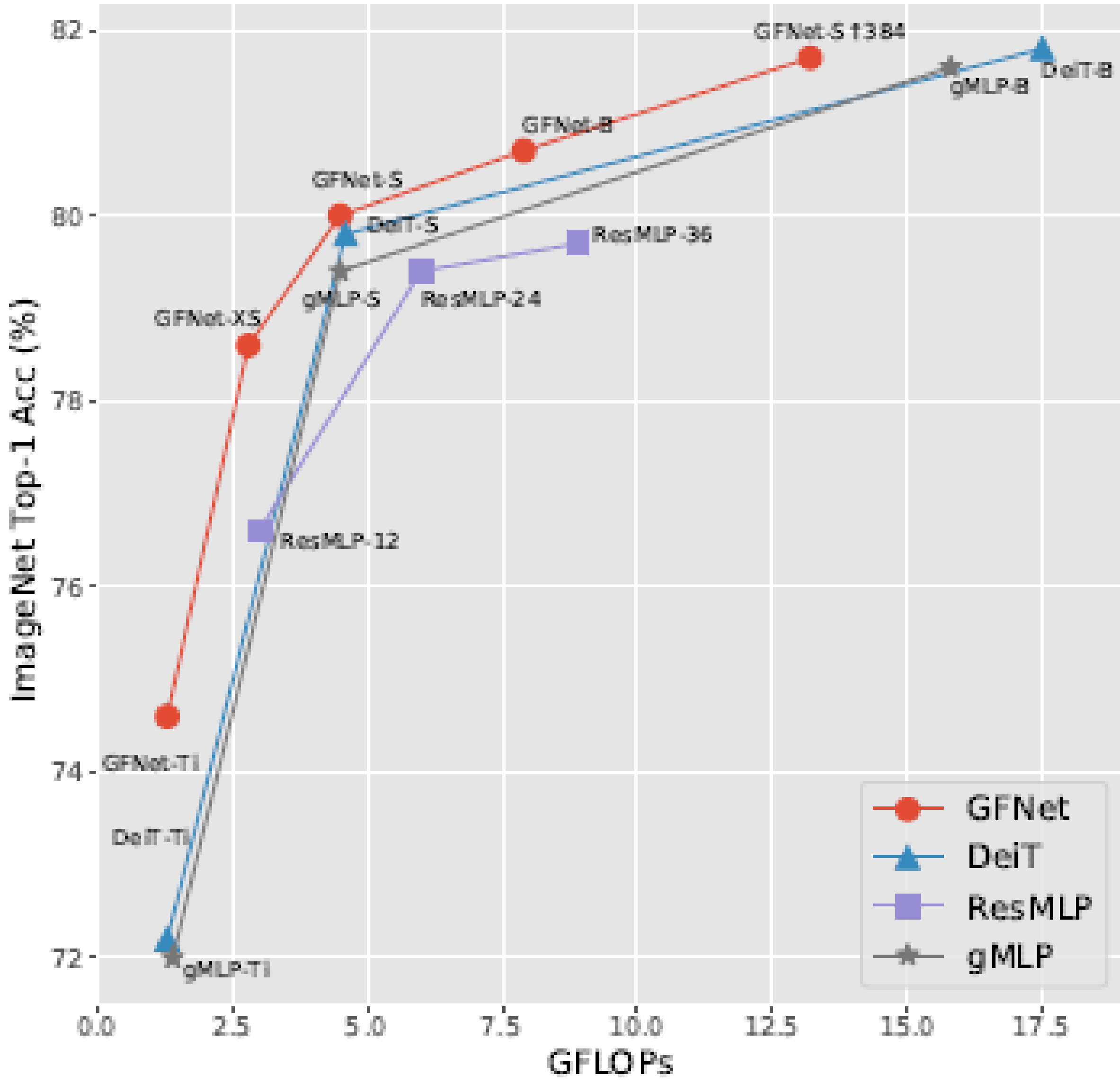


Figure 2: Comparisons among GFNet, ViT [10] and ResMLP in (a) FLOPs (b) latency and (c) GPU memory with respect to the number of tokens (feature resolution). The dotted lines indicate the estimated values when the GPU memory has run out. The latency and GPU memory is measured using a single NVIDIA RTX 3090 GPU with batch size 32 and feature dimension 384.

FLOPs: floating point operations per second



Top-N Accuracy takes the N model predictions with higher probability. If one of them is a true label, it classifies the prediction as correct.

Figure 3: ImageNet acc. vs model complexity

13. 5. FourierFormer: Transformer Meets Generalized Fourier Integral Theorem

- In response, we first interpret attention in transformers as a nonparametric kernel regression. We then propose the FourierFormer, a new class of transformers in which the dot-product kernels are replaced by the novel generalized Fourier integral kernels.
- the generalized Fourier integral kernels can automatically capture dependency and remove the need to tune the covariance matrix.
- Our PyTorch code with documentation can be found at **https://github.com/minhtannguyen/FourierFormer_NeurIPS**

Tan M. Nguyen⇤
Department of Mathematics
University of California, Los Angeles
tanmnguyen89@ucla.edu

Minh Pham⇤
Department of Mathematics
University of California, Los Angeles
minhrose@ucla.edu

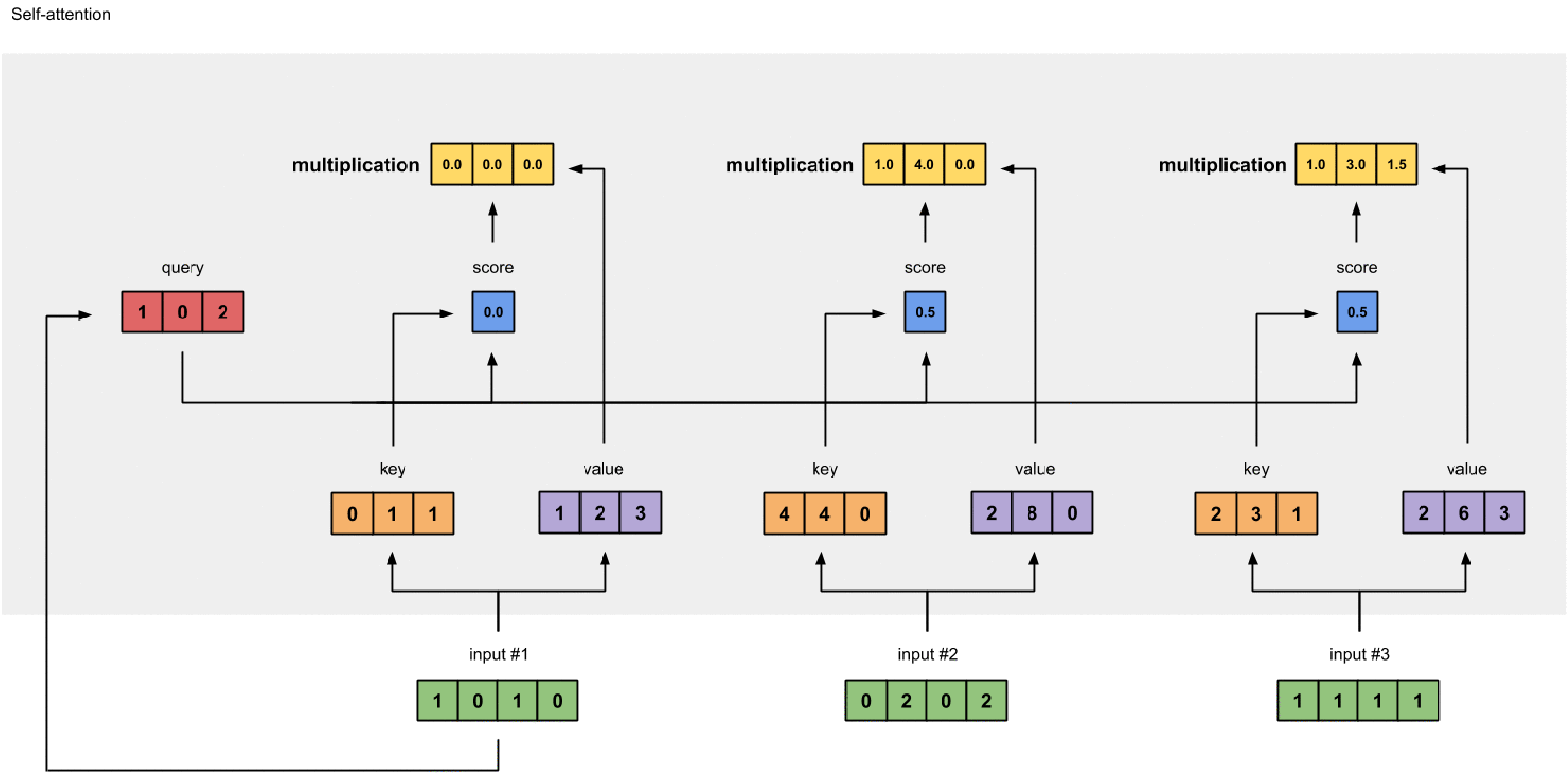
Tam Nguyen
Department of ECE
Rice University
nguyenminhtam9520@gmail.com

Khai Nguyen
Department of Statistics and Data Sciences
University of Texas at Austin
khainb@utexas.edu

Stanley J. Osher⇤⇤
Department of Mathematics
University of California, Los Angeles
sjo@math.ucla.edu

Nhat Ho⇤⇤
Department of Statistics and Data Sciences
University of Texas at Austin
minhnhat@utexas.edu

Scheme of Self Attention



13.5.1. A Nonparametric Regression Interpretation of Self-attention

- The key vectors k_j and value vectors v_j are training inputs and training targets
- The query vectors q_i and the output vectors h_i form a set of new inputs and their corresponding targets that need to be estimated.
- nonparametric regression model:

$$v_j = f\left(k_j\right) + \varepsilon_j, j = 1, \dots, N, \qquad k_j \sim p, (v_j, k_j) \sim p$$

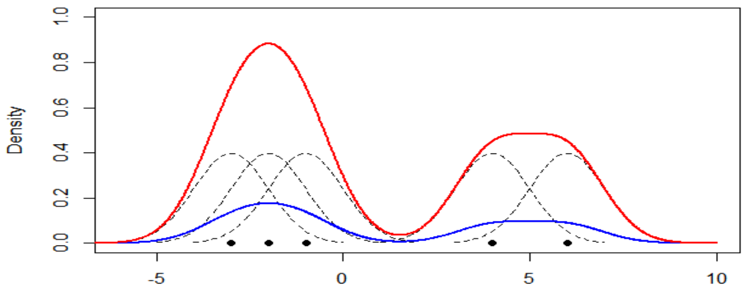
- Nadaraya–Watson's nonparametric kernel regression

$$f(k) = E[v|k]$$

$$E[v|k] = \int_{R^D} v p(v|k) dv = \int_{R^D} \frac{vp(v,k)}{p(k)} dv = \begin{bmatrix} \int_R \frac{v_1 p(v,k)}{p(k)} dv_1 \\ \vdots \\ \int_R \frac{v_N p(v,k)}{p(k)} dv_N \end{bmatrix}$$

- ## Kernel density estimator

Using the isotropic Gaussian kernel with bandwidth ", we have the following estimators:



$$\hat{p}_\sigma(v,k) = \frac{1}{N}\sum_{j=1}^{N} \varphi_\sigma(v - v_j)\varphi_\sigma(k - k_j)$$

$$\hat{p}_\sigma(k) = \frac{1}{N}\sum_{j=1}^{N} \varphi_\sigma(k - k_j)$$

the isotropic multivariate Gaussian density function with diagonal covariance matrix

- Given the kernel density estimators, we obtain the following estimation of the function f:

$$\hat{f}_\sigma(k) = \int_{R^D} \frac{v\hat{p}_\sigma(v,k)}{\hat{p}_\sigma(k)} dv = \int_{R^D} \frac{v\sum_{j=1}^{N}\varphi_\sigma(v - v_j)\varphi_\sigma(k - k_j)}{\sum_{j'=1}^{N}\varphi_\sigma(k - k_{j'})} dv$$

$$= \sum_{j=1}^{N} \frac{\varphi_\sigma\left(k - k_j\right)}{\sum_{j'=1}^{N}\varphi_\sigma\left(k - k_{j'}\right)} \int_{R^D} v\,\varphi_\sigma\left(v - v_j\right)dv = \sum_{j=1}^{N} \frac{\varphi_\sigma\left(k - k_j\right)}{\sum_{j'=1}^{N}\varphi_\sigma\left(k - k_{j'}\right)} v_j \quad \textbf{(1)}$$

13.5.2. Connection between Self-Attention and nonparametric regression

The query vectors q_i and the output vectors h_i form a set of new inputs and their corresponding targets.

By plugging the query vectors q_i into the function $\hat{f}_\sigma$ in equation (1), we obtain that

$$\boldsymbol{h_i} = \hat{\boldsymbol{f}}_{\boldsymbol{\sigma}}(\boldsymbol{q_i}) = \sum_{j=1}^{N} \frac{\boldsymbol{\varphi_\sigma}\left(\boldsymbol{q_i} - \boldsymbol{k_j}\right)}{\sum_{j'=1}^{N} \boldsymbol{\varphi_\sigma}\left(\boldsymbol{q_i} - \boldsymbol{k_{j'}}\right)} \boldsymbol{v_j} \qquad (2)$$

Note that

$$\frac{\boldsymbol{\varphi_\sigma}\left(\boldsymbol{q_i} - \boldsymbol{k_j}\right)}{\sum_{j'=1}^{N} \boldsymbol{\varphi_\sigma}\left(\boldsymbol{q_i} - \boldsymbol{k_{j'}}\right)} = \frac{\boldsymbol{exp}\{-\frac{\left\|\boldsymbol{q_i} - \boldsymbol{k_j}\right\|^2}{2\boldsymbol{\sigma}^2}\}}{\sum_{j'=1}^{N} \boldsymbol{exp}\{-\frac{\left\|\boldsymbol{q_i} - \boldsymbol{k_{j'}}\right\|^2}{2\boldsymbol{\sigma}^2}\}} = \frac{\boldsymbol{exp}\left\{-\frac{\|\boldsymbol{q_i}\|^2 + \left\|\boldsymbol{k_j}\right\|^2}{2\boldsymbol{\sigma}^2}\right\} \boldsymbol{exp}\left\{\frac{\boldsymbol{q_i^T k_j}}{\boldsymbol{\sigma}^2}\right\}}{\sum_{j'=1}^{N} \boldsymbol{exp}\left\{-\frac{\|\boldsymbol{q_i}\|^2 + \left\|\boldsymbol{k_j}'\right\|^2}{2\boldsymbol{\sigma}^2}\right\} \boldsymbol{exp}\left\{\frac{\boldsymbol{q_i^T k_{j'}}}{\boldsymbol{\sigma}^2}\right\}} \qquad (3)$$

Assume that k_j *is normalized,* then equation (3) is reduced to

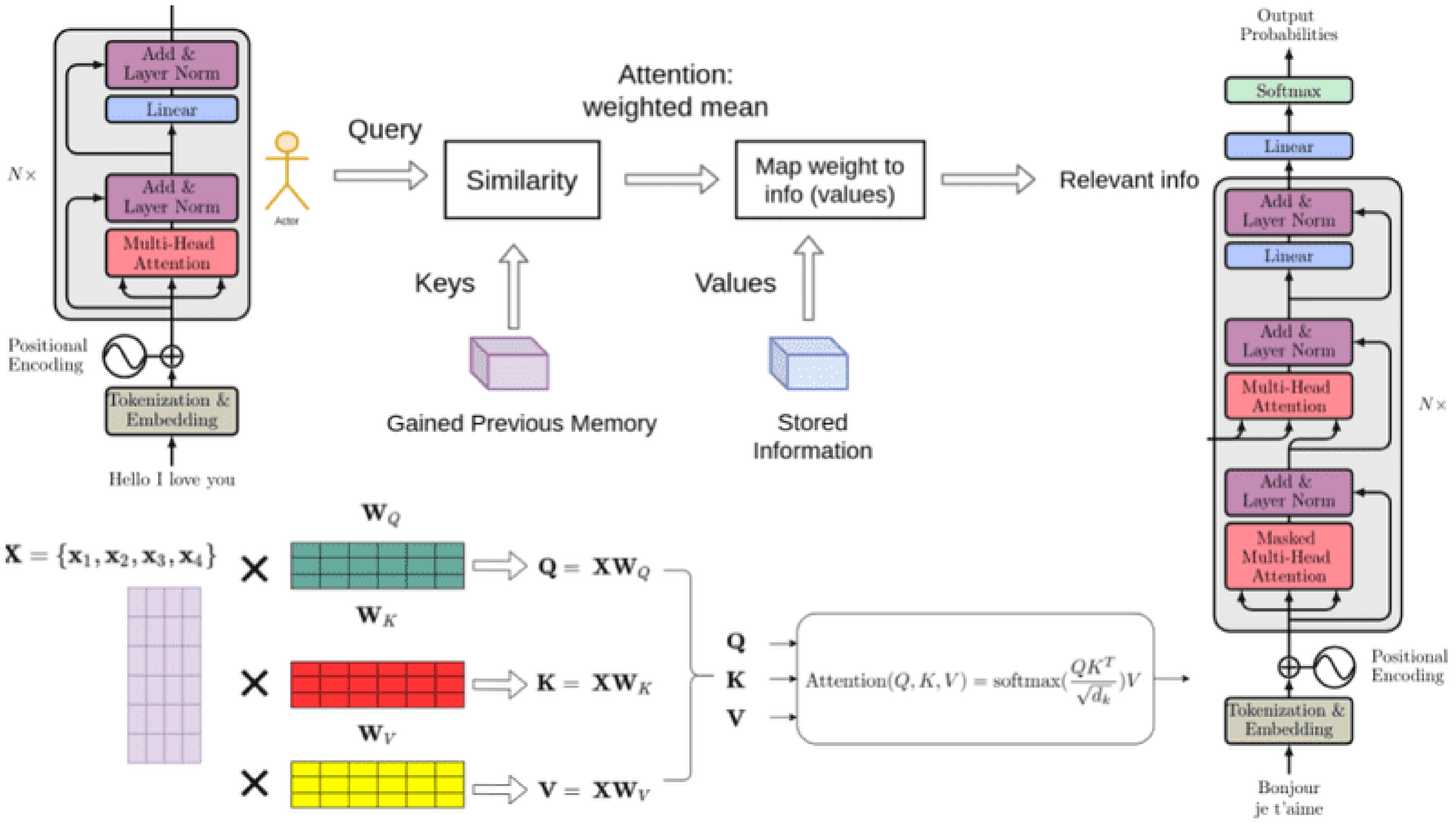
$$\frac{\varphi_\sigma\left(q_i - k_j\right)}{\sum_{j'=1}^{N} \varphi_\sigma\left(q_i - k_{j'}\right)} = \frac{exp\left\{\frac{q_i^T k_j}{\sigma^2}\right\}}{\sum_{j'=1}^{N} exp\left\{\frac{q_i^T k_{j'}}{\sigma^2}\right\}} \tag{4}$$

Substituting equation (4) into equation (2), we obtain

$$h_i = \hat{f}_\sigma(q_i) = \sum_{j=1}^{N} \frac{exp\left\{\frac{q_i^T k_j}{\sigma^2}\right\}}{\sum_{j'=1}^{N} exp\left\{\frac{q_i^T k_{j'}}{\sigma^2}\right\}} v_j$$

$$= \sum_{j=1}^{N} softmax\left(\frac{q_i^T k_j}{\sigma^2}\right) v_j \tag{5}$$

Choose $\sigma^2 = \sqrt{D}$where D is the dimension of q_i and k_j , equation (5) matches equation of self-attention.



Add & Layer Norm
Linear
Multi-Head Attention
N×
Positional Encoding
Tokenization & Embedding
Hello I love you
Actor
Query
Similarity
Attention: weighted mean
Map weight to info (values)
Relevant info
Keys
Values
Gained Previous Memory
Stored Information
Output Probabilities
Softmax
Masked Multi-Head Attention
Bonjour je t'aime
$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$
$\mathbf{W}_Q$
$\mathbf{W}_K$
$\mathbf{W}_V$
$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$
$\mathbf{K} = \mathbf{X}\mathbf{W}_K$
$\mathbf{V} = \mathbf{X}\mathbf{W}_V$
Q
K
V
$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$

- ## Limitation

Limitation of Self-Attention from our nonparametric regression interpretation, self-attention is derived from the use of isotropic Gaussian kernels for kernel density estimation and nonparametric regression estimation,

- which may fail to capture the complex correlations between D features in q_i and k_j .
- Using multivariate Gaussian kernels with dense covariance matrices can help capture such correlations; however, choosing good covariance matrices is challenging and inefficient.
- In the following section, we discuss the Fourier integral estimator and its use as a kernel for computing self-attention in order to overcome these limitations.

13.5.3. FourierFormer: Transformer via Generalized Fourier Integral Theorem

13.5.3.1. Generalized Fourier Integral Theorems and Their Applications

Fourier integral theorem is a combination of Fourier transform and Fourier inverse transform. Let $\boldsymbol{p} \in \boldsymbol{L_1}(\boldsymbol{R^D})$, **using equations (F1) and (F3), we obtain** the Fourier integral theorem:

$$\boldsymbol{p(k)} = \int_{-R^D}^{R^D} \widehat{\boldsymbol{p}}(\boldsymbol{s}) \boldsymbol{e}^{i2\pi s^T k} \boldsymbol{ds} = \int_{-R^D}^{R^D} \int_{-R^D}^{R^D} \boldsymbol{p(y)} \boldsymbol{e}^{-i2\pi s^T y} \boldsymbol{dy}\, \boldsymbol{e}^{i2\pi s^T k} \boldsymbol{ds}$$

$$= \int_{-R^D}^{R^D} \int_{-R^D}^{R^D} \boldsymbol{p(y)} \boldsymbol{e}^{i2\pi s^T (k-y)} \boldsymbol{dy}\, \boldsymbol{ds} = \frac{1}{(2\pi)^D} \int_{-R^D}^{R^D} \int_{-R^D}^{R^D} \boldsymbol{p(y)} \boldsymbol{e}^{is^T (k-y)} \boldsymbol{dy}\, \boldsymbol{ds} \qquad (6)$$

Note that

$$\int_{-R^D}^{R^D} e^{is^T(k-y)} ds = \int_{-R^D}^{R^D} e^{i\left[\sum_{j=1}^{D} s_j (k_j - y_j)\right]} ds$$

$$= \prod_{j=1}^{D} \int_{-R}^{R} e^{is_j (k_j - y_j)} ds_j$$

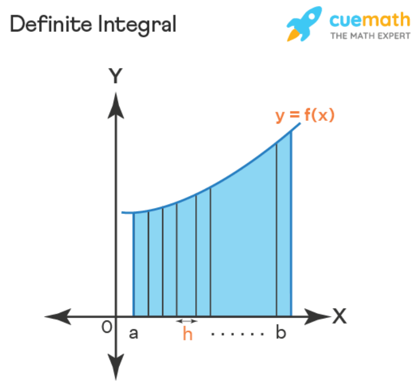
$$= \prod_{j=1}^{D} \int_{-R}^{R} \frac{1}{i\,(k_j - y_j)} d\left[e^{is_j (k_j - y_j)}\right]$$

$$= 2^D \prod_{j=1}^{D} \frac{Sin\left[R\left(k_j - y_j\right)\right]}{(k_j - y_j)} \qquad (8)$$

Substitituting equation (8) into equation (6) yields

$$p(k) = \frac{1}{\pi^D} \int_{R^D} \prod_{j=1}^{D} \frac{Sin\left[R\left(k_j - y_j\right)\right]}{\left(k_j - y_j\right)} p(y) dy \qquad (9)$$

13.5.3.2. Generalized Fourier integral estimator



- **Generalized Fourier integral theorem:**

$$p(k) = \lim_{n\to\infty} p_R^{\emptyset} = \lim_{n\to\infty} \frac{R^D}{A^D} \int_{R^D} \prod_{j=1}^{D} \emptyset\left(\frac{Sin\left[R\left(k_j - y_j\right)\right]}{R\left(k_j - y_j\right)}\right) p(y)dy \qquad (10)$$

$A = \int_R \emptyset\left(\frac{sin(z)}{z}\right)$**dz,** $\emptyset: R \to R$ **is a given function.**

- **Generalized Fourier density estimator**

$$\int_R g\left(y_j\right)p\left(y_j\right)dy_j = \sum_{i=1}^{N} g\left(y_{ij}\right)\frac{1}{N}$$

Let $p(y) = \frac{1}{N}$, then we have

$$\int_{R^D} \prod_{j=1}^{D} \emptyset\left(\frac{Sin\left[R\left(k_j - y_j\right)\right]}{R\left(k_j - y_j\right)}\right) p(y)dy = \frac{1}{N}\sum_{i=1}^{N} \prod_{j=1}^{D} \emptyset\left(\frac{Sin\left[R\left(k_j - y_{ij}\right)\right]}{R\left(k_j - y_{ij}\right)}\right) \qquad (11)$$

Assume that $\boldsymbol{k_1, \dots, k_i, \dots, k_D} \in \boldsymbol{R^D}$ **and be i.i.d, where** $\boldsymbol{k_i} = (\boldsymbol{k_{i1}, \dots, k_{ij}, \dots, k_{iD}})$

Substituting equation (11) into equation (10) yields the Generalized Fourier density estimator:

$$\boldsymbol{p_{N,R}^{\emptyset}(k) = \frac{R^D}{NA^D} \sum_{i=1}^{N} \prod_{j=1}^{D} \emptyset\left(\frac{Sin\left[R\left(k_j - k_{ij}\right)\right]}{R\left(k_j - k_{ij}\right)}\right)} \qquad (12)$$

where $\boldsymbol{k} = (k_1, \dots, k_j, \dots, k_D)$

13.5.3.3. FourierFormer: Transformers with Fourier Attentions

Recall the nonparametric regression model

$$\boldsymbol{v} = \boldsymbol{f}(\boldsymbol{k}) + \boldsymbol{\varepsilon}$$

- **The Nadaraya–Watson estimator of the function $\boldsymbol{f}$**

$$\boldsymbol{f}_{N,R}(\boldsymbol{k}) = \frac{\sum_{i=1}^{N} \boldsymbol{v}_i \prod_{j=1}^{D} \emptyset\left(\frac{\boldsymbol{Sin}\left[\boldsymbol{R}\left(\boldsymbol{k}_j - \boldsymbol{k}_{ij}\right)\right]}{\boldsymbol{R}\left(\boldsymbol{k}_j - \boldsymbol{k}_{ij}\right)}\right)}{\sum_{i=1}^{N} \prod_{j=1}^{D} \emptyset\left(\frac{\boldsymbol{Sin}\left[\boldsymbol{R}\left(\boldsymbol{k}_j - \boldsymbol{k}_{ij}\right)\right]}{\boldsymbol{R}\left(\boldsymbol{k}_j - \boldsymbol{k}_{ij}\right)}\right)} \qquad (13)$$

$$\boldsymbol{k} = (\boldsymbol{k}_1, \dots, \boldsymbol{k}_j, \dots, \boldsymbol{k}_D)$$

- **FourierFormer**

Given the generalized Fourier nonparametric regression estimator $f_{N,R}$ in equation (13), by plugging he query values $q_1, \dots, q_N$ into that function, we obtain the following definition of the Fourier attention:

$$\boldsymbol{h_i = f_{N,R}(q_i) = \frac{\sum_{i=1}^{N} v_i \prod_{j=1}^{D} \emptyset\left(\frac{Sin\left[R\left(q_{ij} - k_{ij}\right)\right]}{R\left(q_{ij} - k_{ij}\right)}\right)}{\sum_{i=1}^{N} \prod_{j=1}^{D} \emptyset\left(\frac{Sin\left[R\left(q_{ij} - k_{ij}\right)\right]}{R\left(q_{ij} - k_{ij}\right)}\right)}} \qquad (14)$$

Definition 2 (FourierFormer)

Define a FourierFormer as a transformer that uses Fourier attention to capture dependency between tokens in the input sequence and the correlation between features in each token.

13.5.3.4. Results

Table 1. Perplexity (PPL) on WikiText-103 of FourierFormers compared to the baselines. FourierFormers achieve much better PPL than the baselines.

Method	Valid (PPL)	Test (PPL)
Baseline (Dot-product) (Small)	33.15	34.29
Fourier Former (small)	31.86	32.85
Baseline (Medium)	27.90	29.60
Fourier Former (medium)	26.51	28.01

Image Classification on ImageNet

Table 2. Top-1 and top-5 accuracy (%) of FourierFormer Deit vs. the baseline Deit with dot-product attention. FourierFormer Deit outperforms the baseline in both top-1 and top-5 accuracy.

Method	Top-1 Acc	Top-5 Acc
Baseline Deit	**72.23**	**91.13**
Fourier Former	**73.25**	**91.66**