Practical Transformer Architectures

Qinhang (William) Wu
Time: 3:00pm – 4:00pm, June 12, 2025

Slide Credits

1. *Introduction to Deep Learning*, 11-785, by Liangze Li and Kateryna Shapovalenko, Carnegie Mellon University: https://deeplearning.cs.cmu.edu/S24/document/slides/lec19.transformers.pdf

Index

- Part I: Real-World Applications of Transformer
- Part II: Transformer Architecture for Classification
- Part III: Adapting Transformers for Vision Task

Part I: Real-World Applications of Transformer

Application Domain

- Language
- Speech (or audio)
- Vision
- Time-series Data
- … and more!

Application Domain

- Language
  - Text Classification (e.g., sentiment analysis, spam detection)
  - Machine Translation
  - Question Answering
  - Text Summarization
  - Text Generation
- Speech (or audio)
  - Speech Recognition
  - Speaker Identification
  - Audio Synthesis
- Vision
  - Image Classification
  - Object Detection
  - Image Captioning
- Time-series Data
- … and more!

Transformer: One Architecture, Many Modalities

- Universal input format
  - All modalities (text, image, audio, code, …) are first embedded into a common shape: Input $\rightarrow (L \times d)$ matrix (L: sequence length, d: hidden size)
- Flexible "Front-End"
  - Text → Token Embedding
  - Image → Patch Embedding
  - Audio → Frame Embedding
  - … and more!
- Attention is the core mechanism:
  - It captures dependencies across input elements
- Shared backbone, task-specific heads
  - Classification
  - Generation
  - Segmentation
  - … and more!

Text Input: From Tokens to Embeddings

- Input: raw text

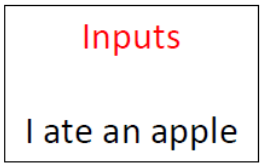


Figure from: Li and Shapovalenko (2024)

Text Input: From Tokens to Embeddings

- Input: raw text
- Step 1: Tokenization
  - Raw text is split into subword tokens

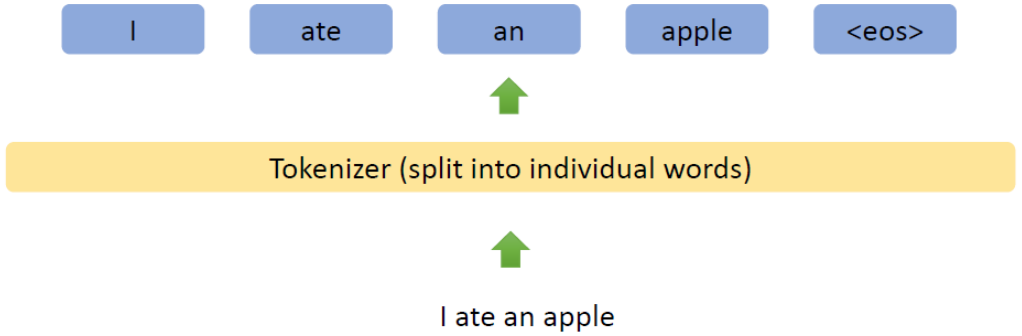


Figure from: Li and Shapovalenko (2024)

Text Input: From Tokens to Embeddings

- Input: raw text
- Step 1: Tokenization
  - Raw text is split into subword tokens
- Step 2: Token Embedding
  - Each token is mapped to a learned vector
  - After that, a sequence of shape $(L \times d)$ is produced

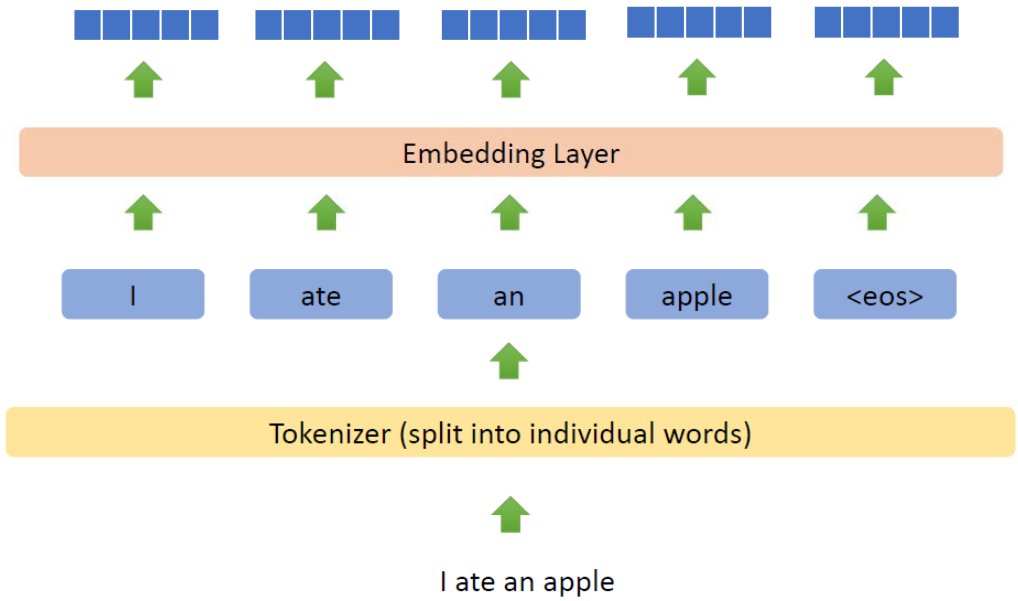


Figure from: Li and Shapovalenko (2024)

Text Input: From Tokens to Embeddings

- Input: raw text
- Step 1: Tokenization
  - Raw text is split into subword tokens
- Step 2: Token Embedding
  - Each token is mapped to a learned vector
  - After that, a sequence of shape $(L \times d)$ is produced
- Step 3: Positional Encoding
  - Since attention is permutation-invariant, positional information is added to token embedding

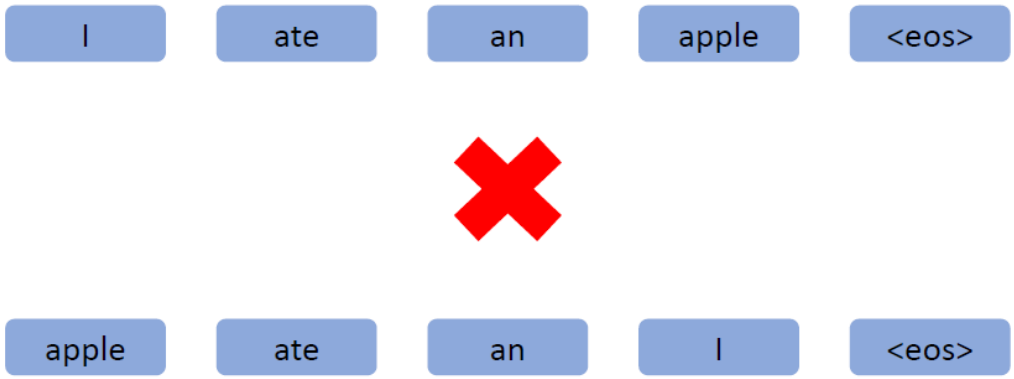
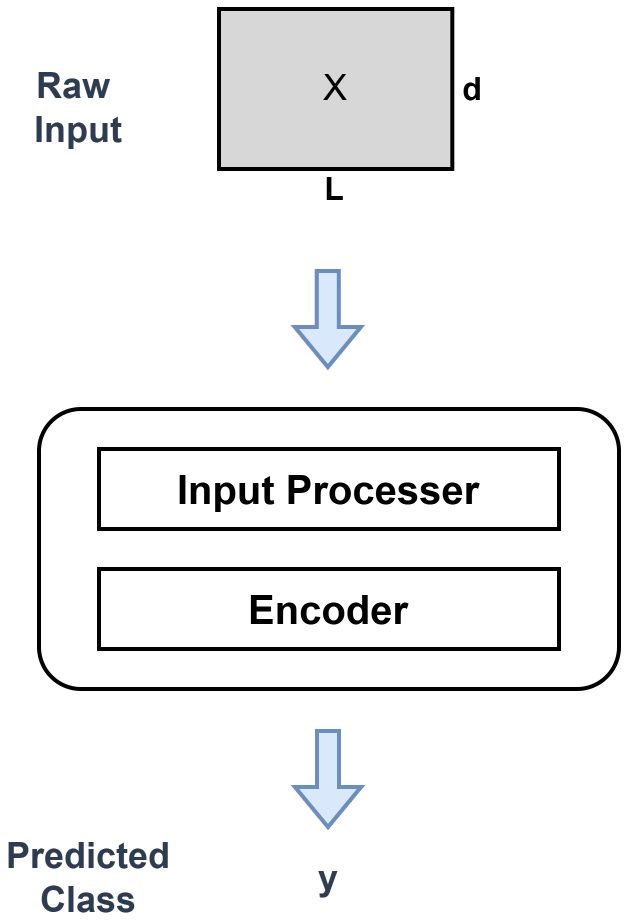


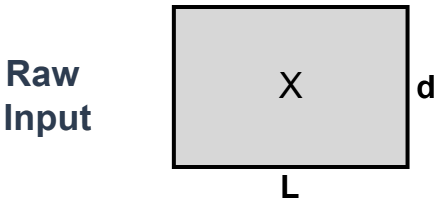
Figure from: Li and Shapovalenko (2024)

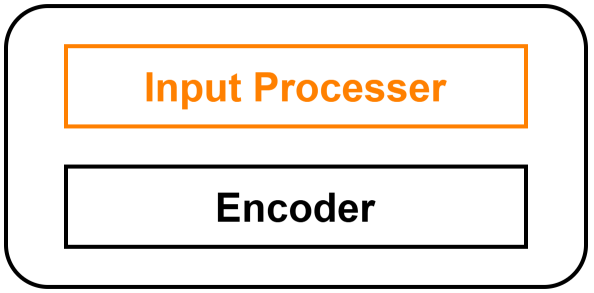
Part II: Transformer Architecture for Classification

High-level Architecture

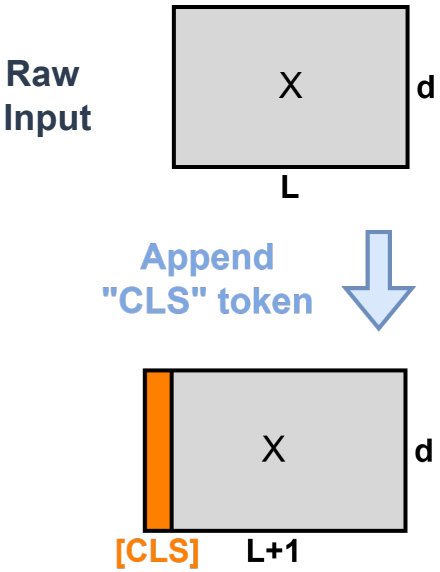


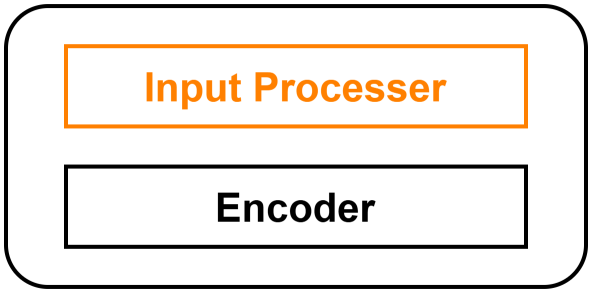
Input Processing



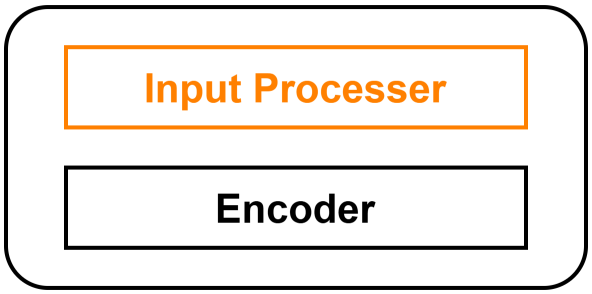


Input Processing





Input Processing



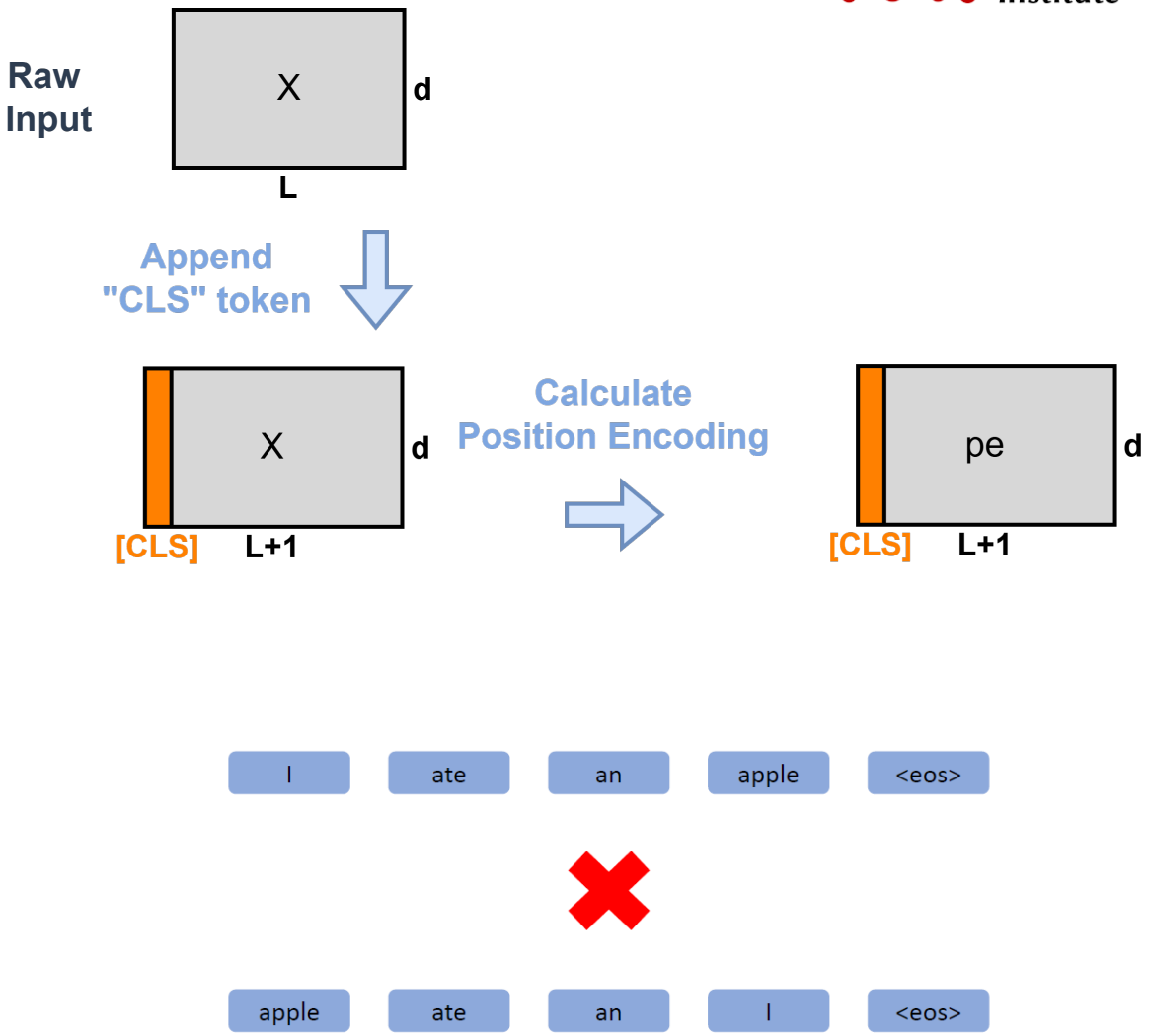
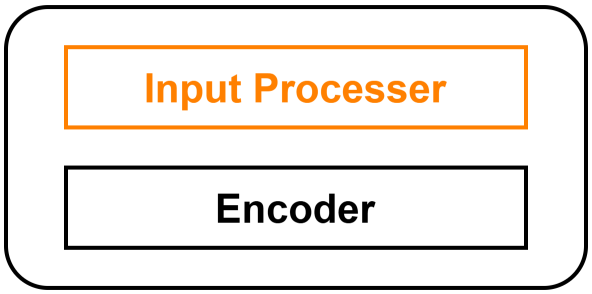
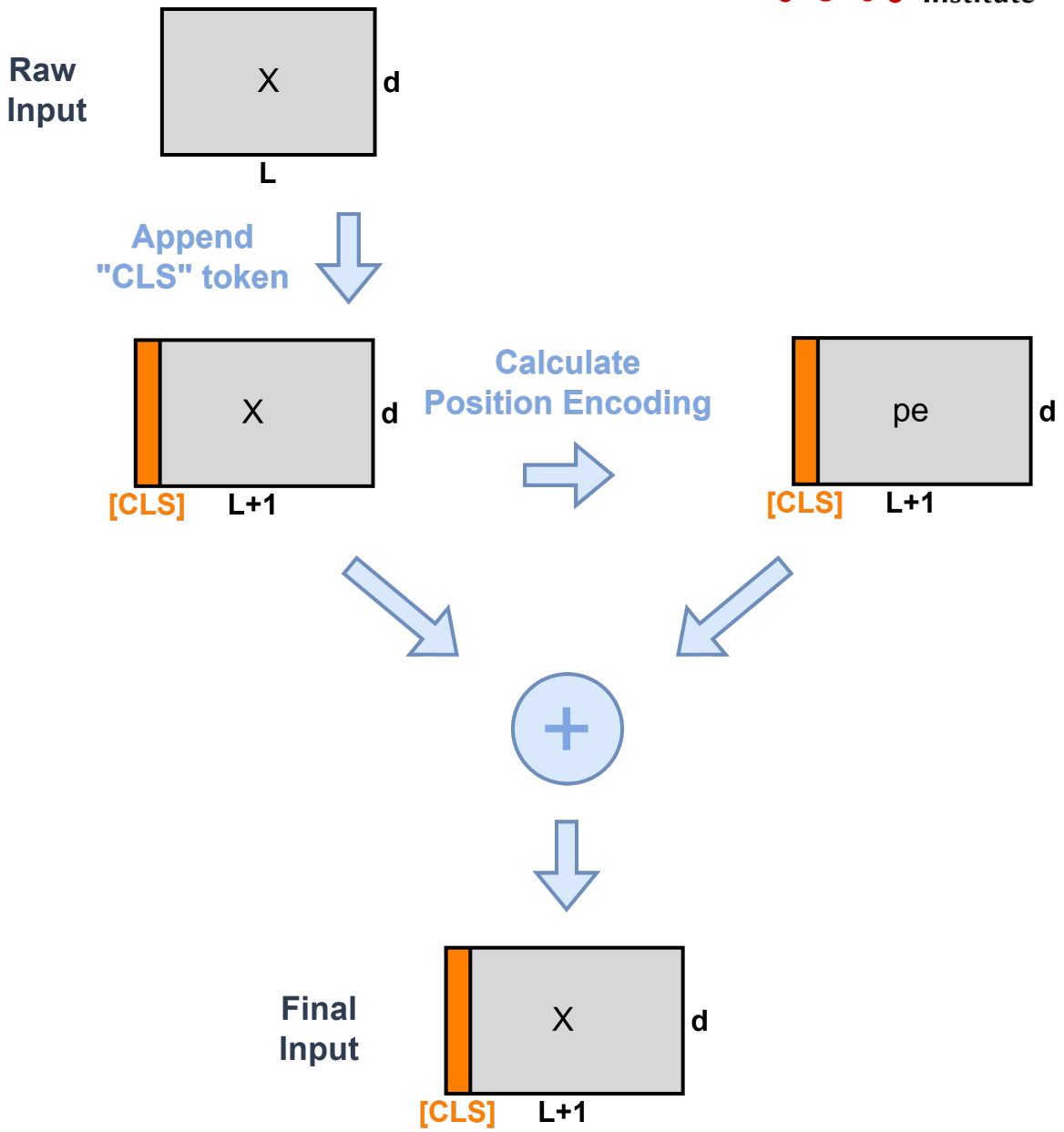


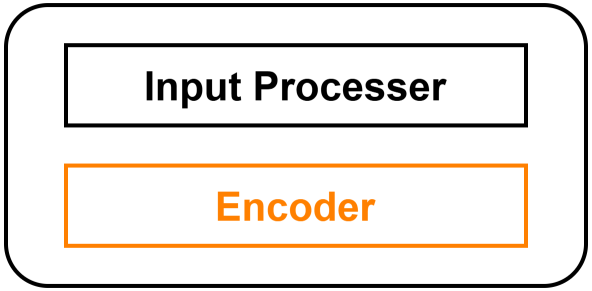
Figure from: Li and Shapovalenko (2024)

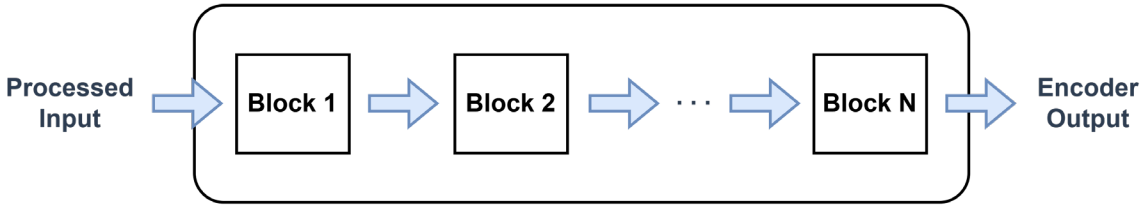
Input Processing



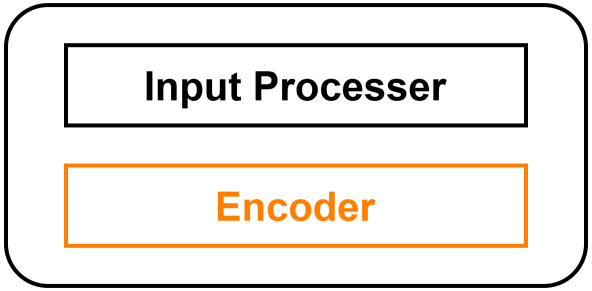


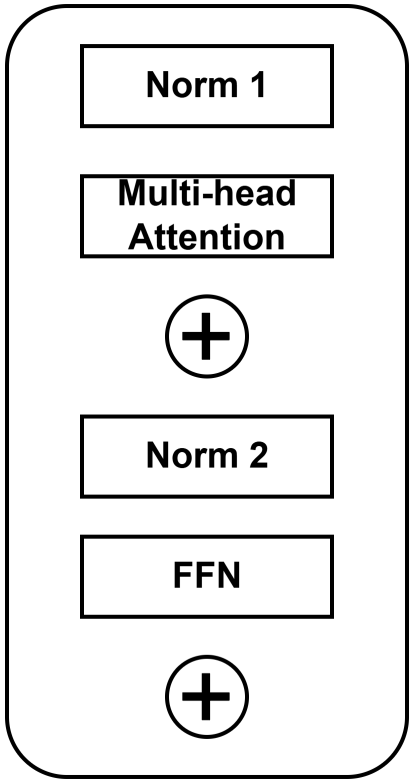
Encoder: High-level Structure



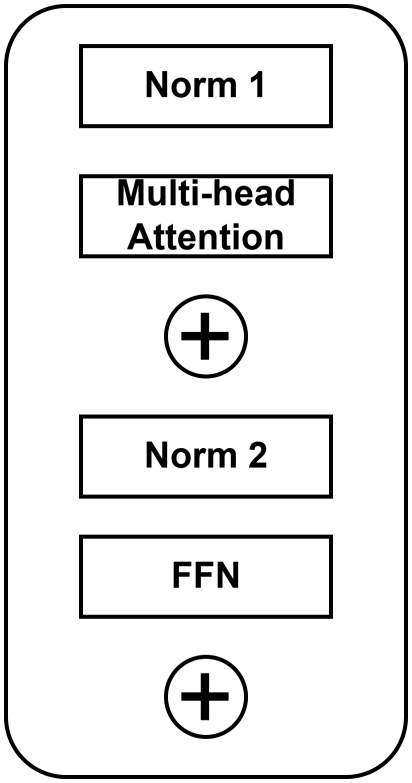


Encoder: Single Block

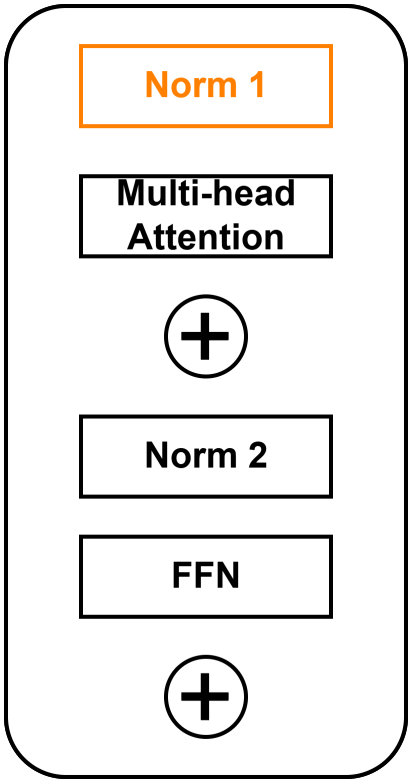


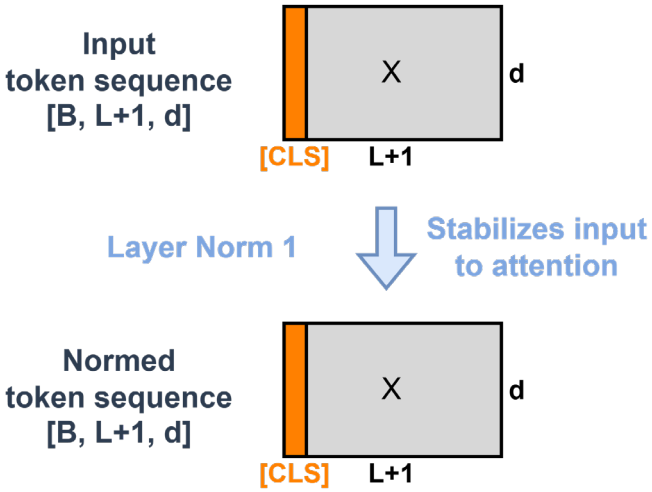


Encoder: Single Block

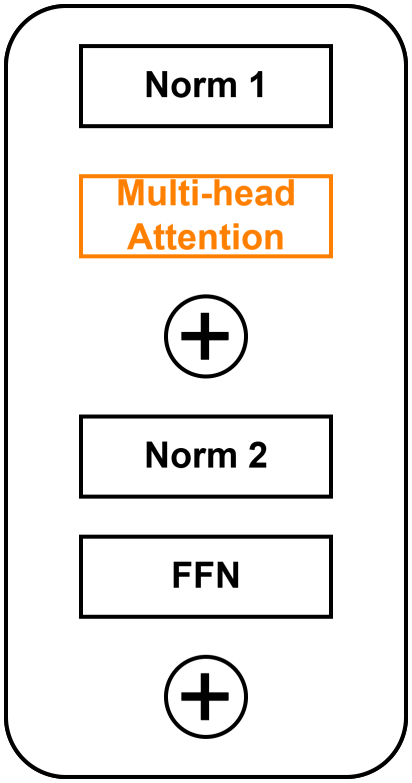


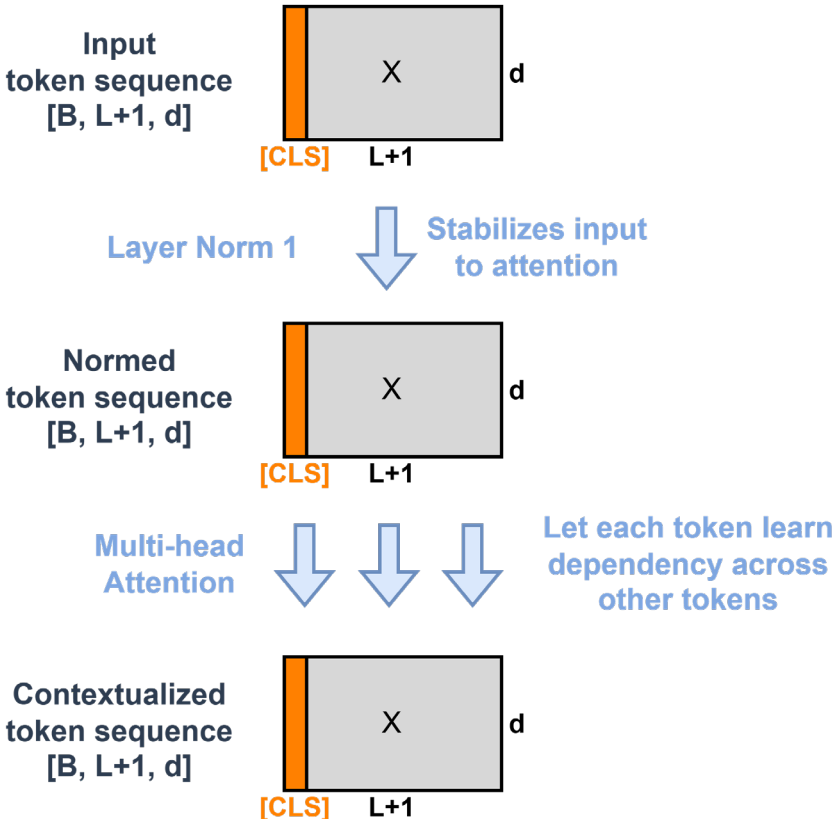
Encoder: Single Block



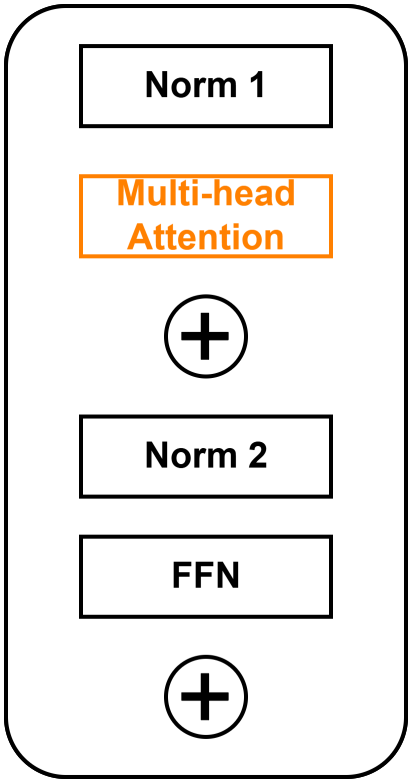


Encoder: Single Block

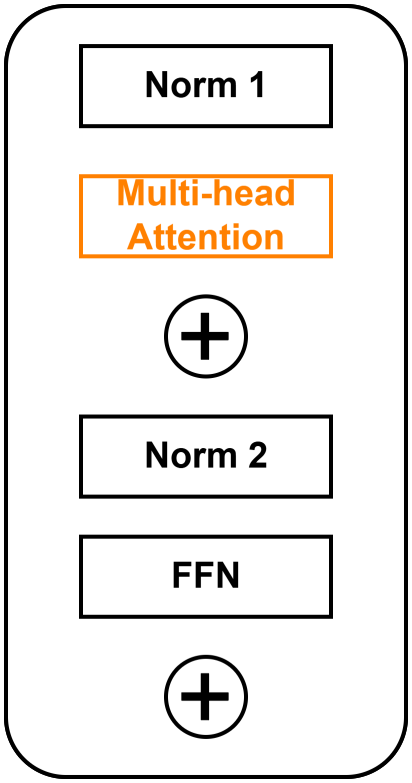


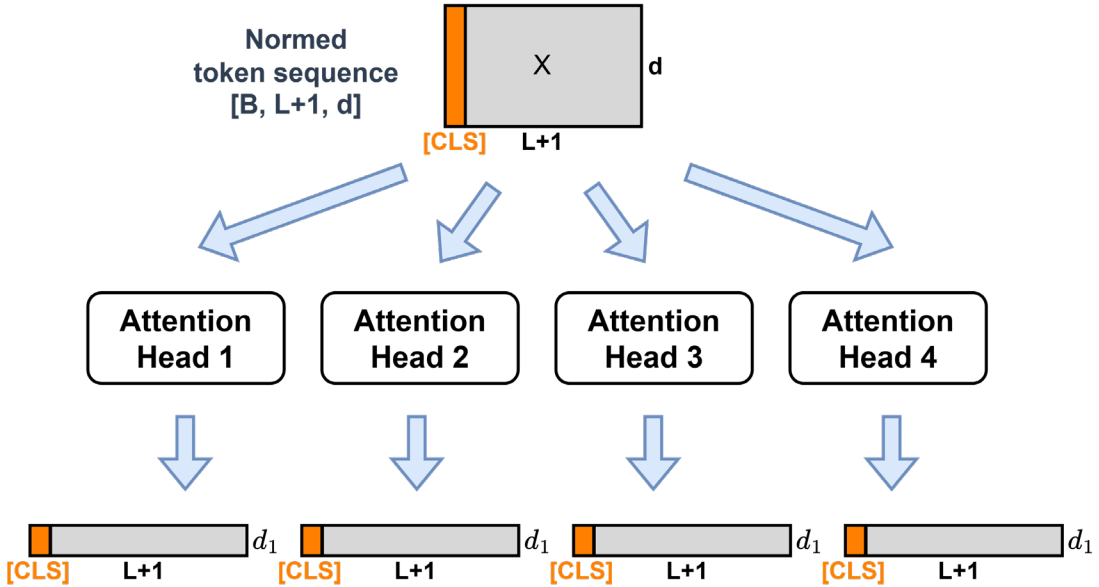


Encoder: Multi-head Attention

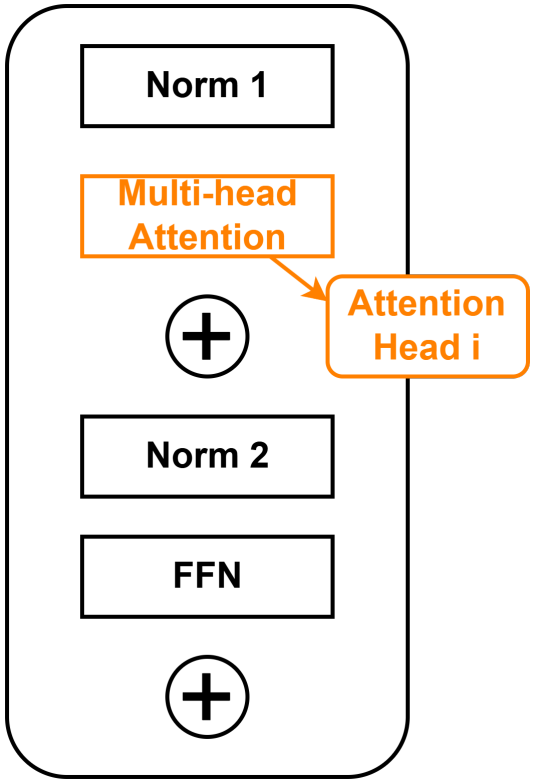


Encoder: Multi-head Attention

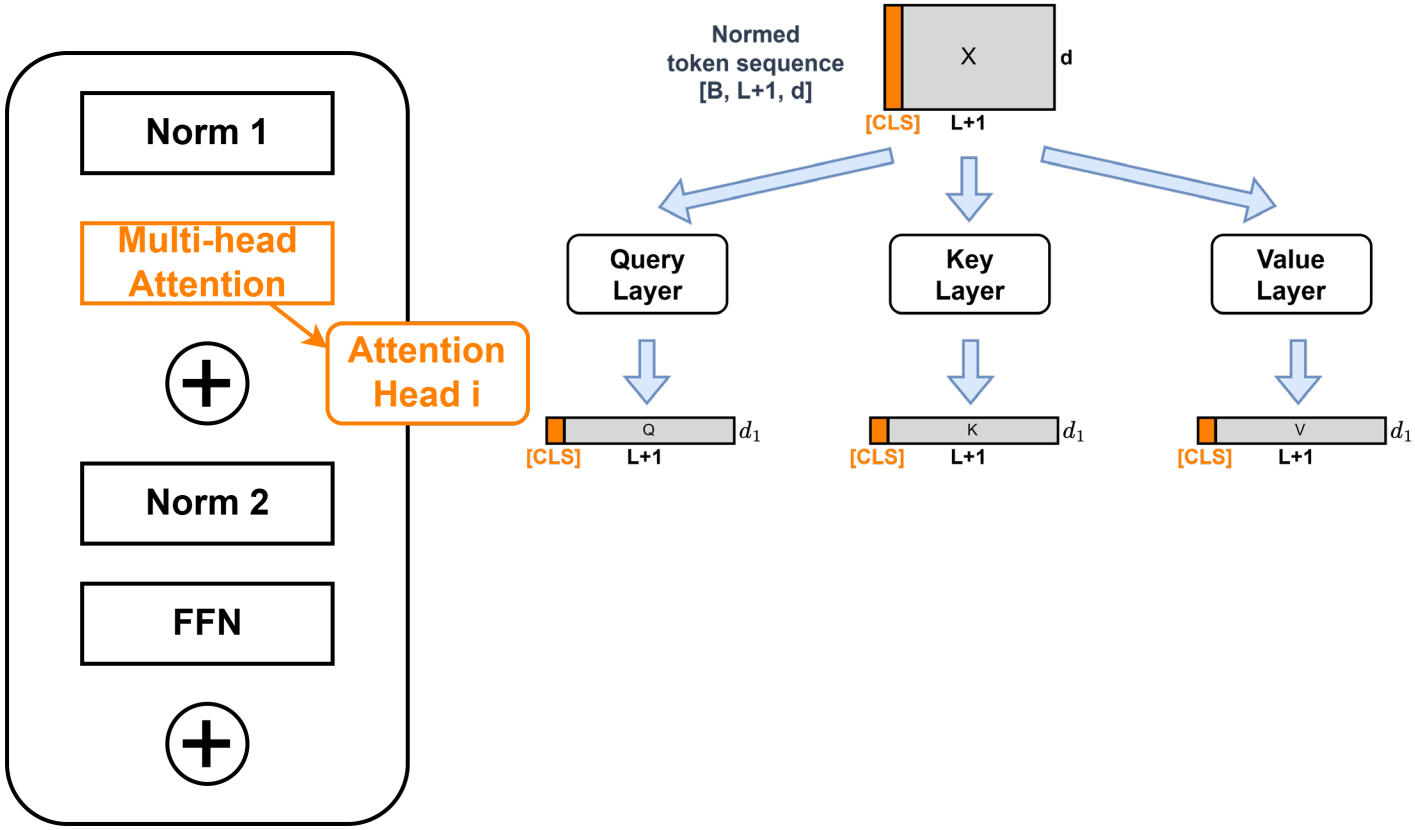




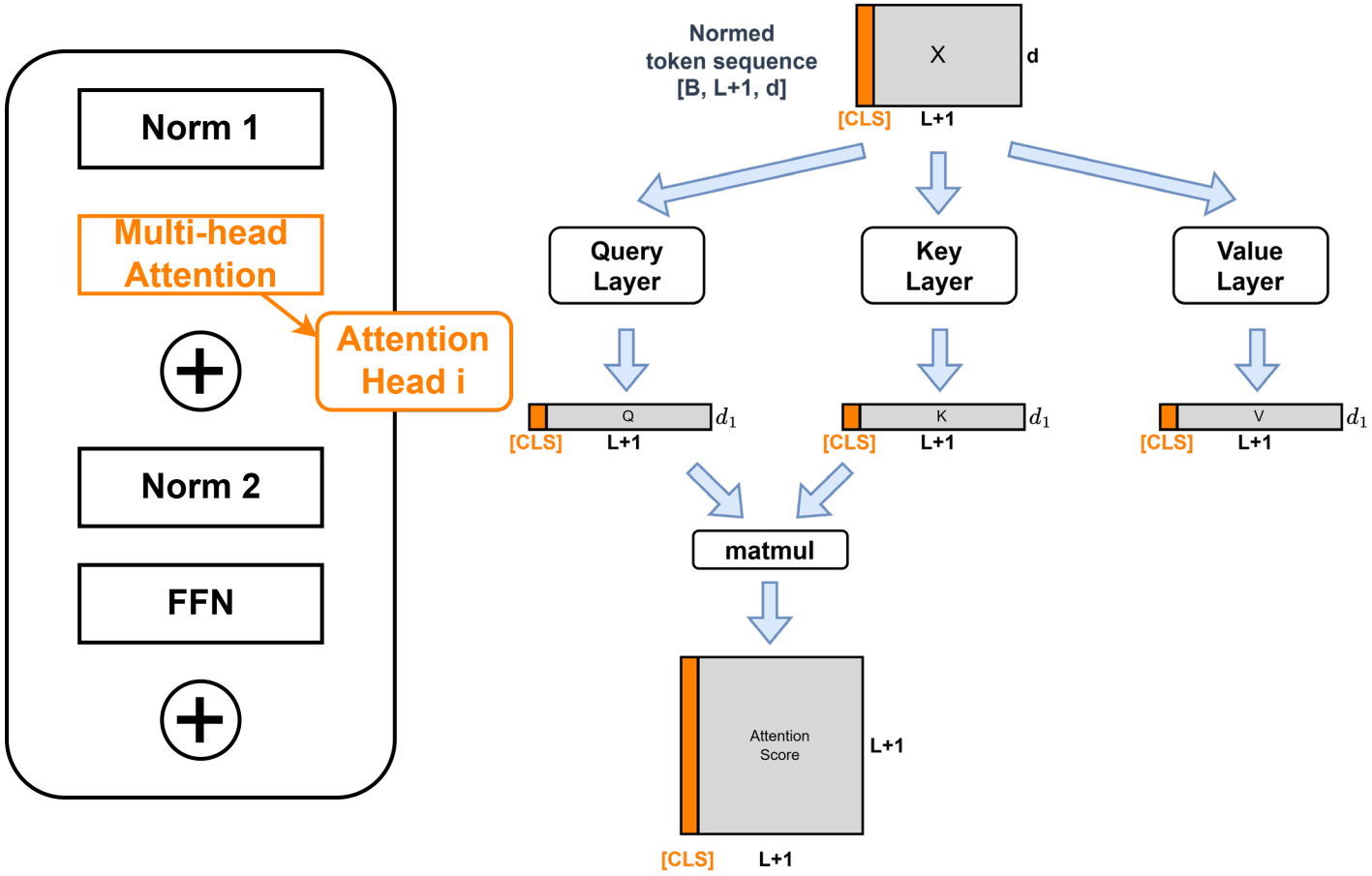
Encoder: Single Attention Head



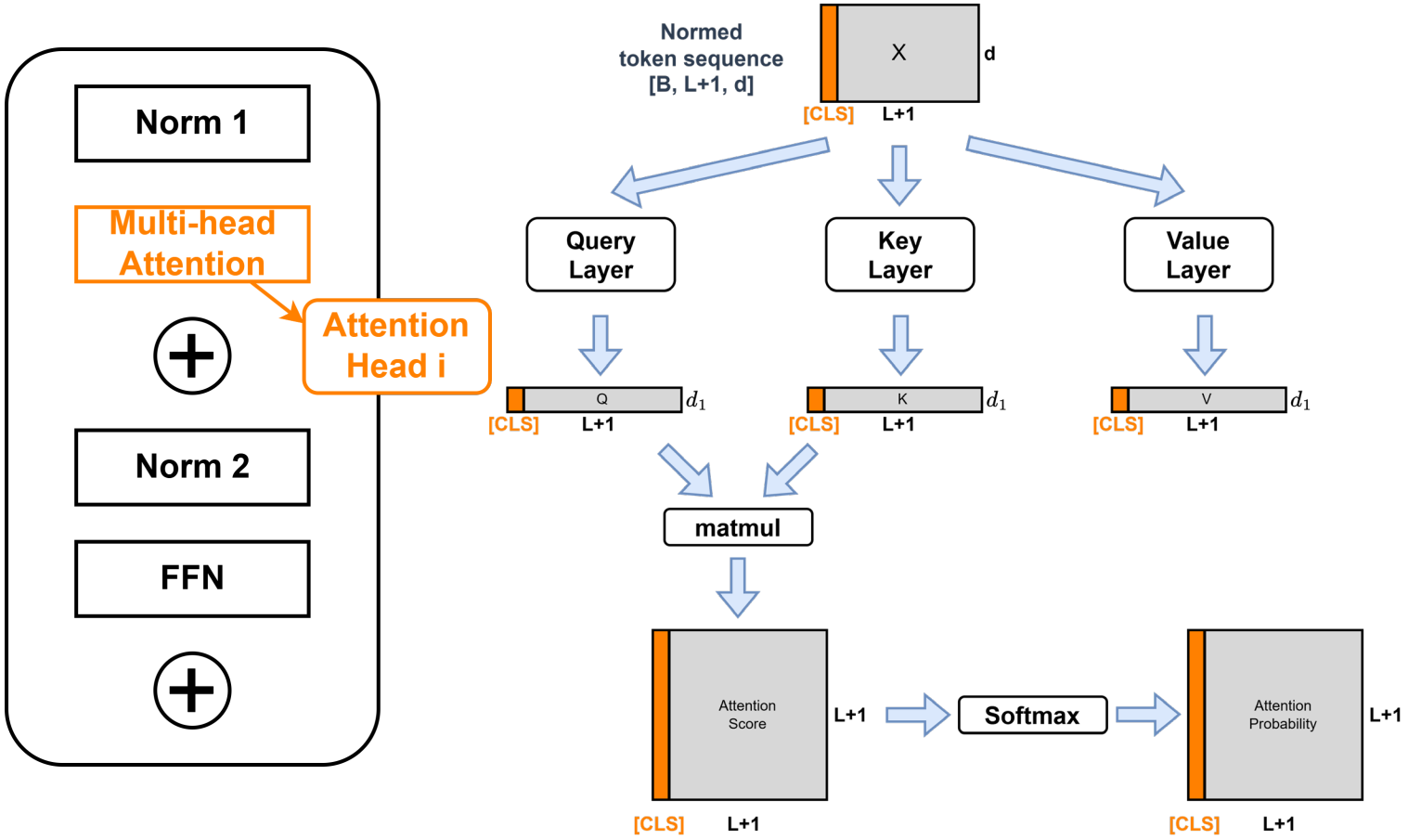
Encoder: Single Attention Head



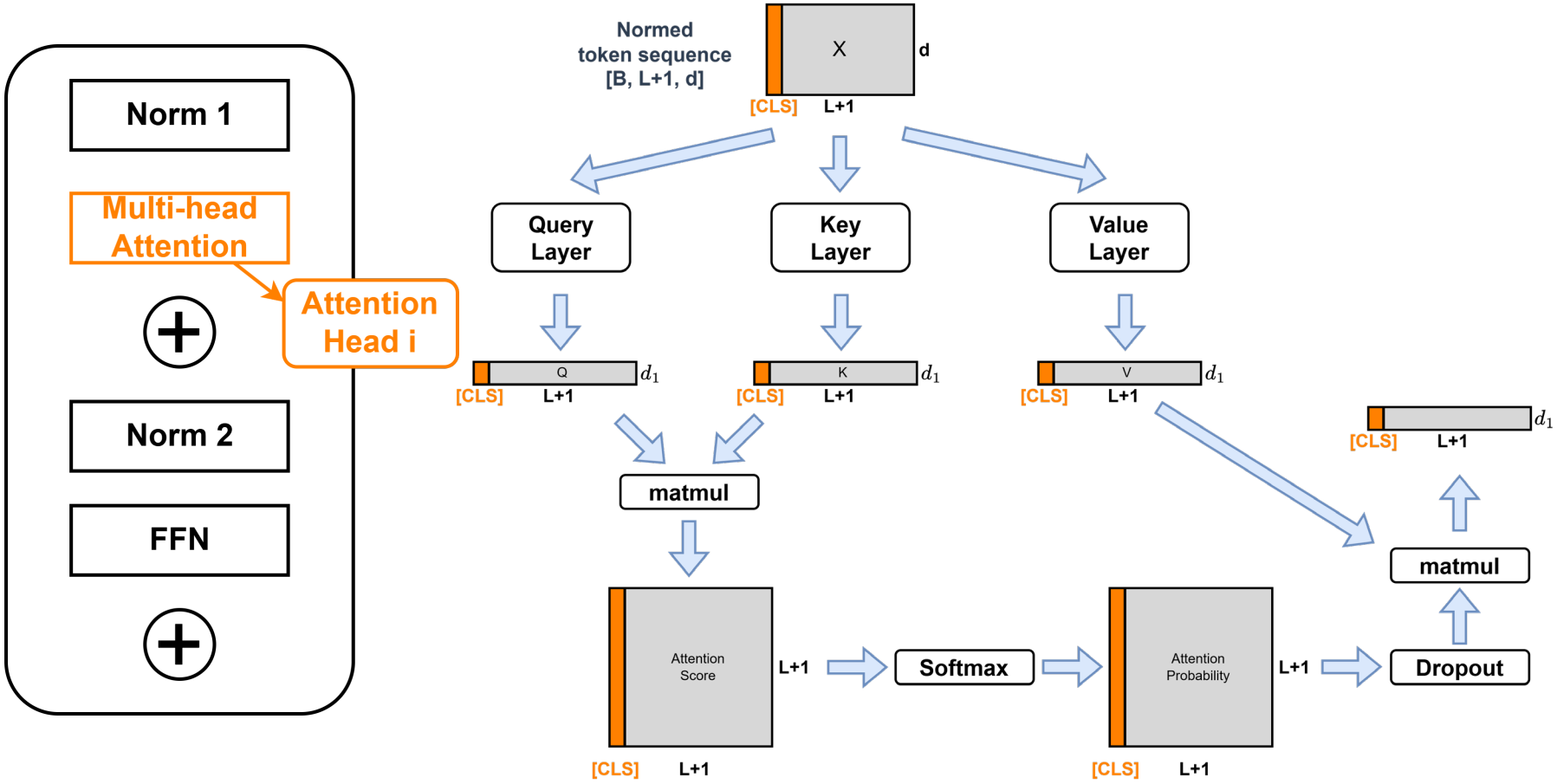
Encoder: Single Attention Head



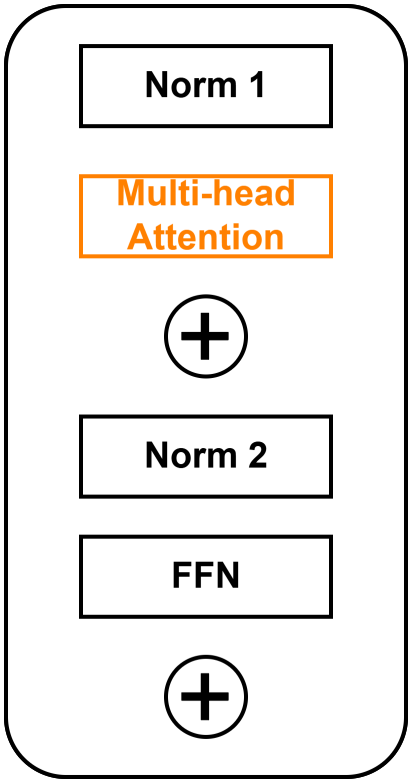
Encoder: Single Attention Head

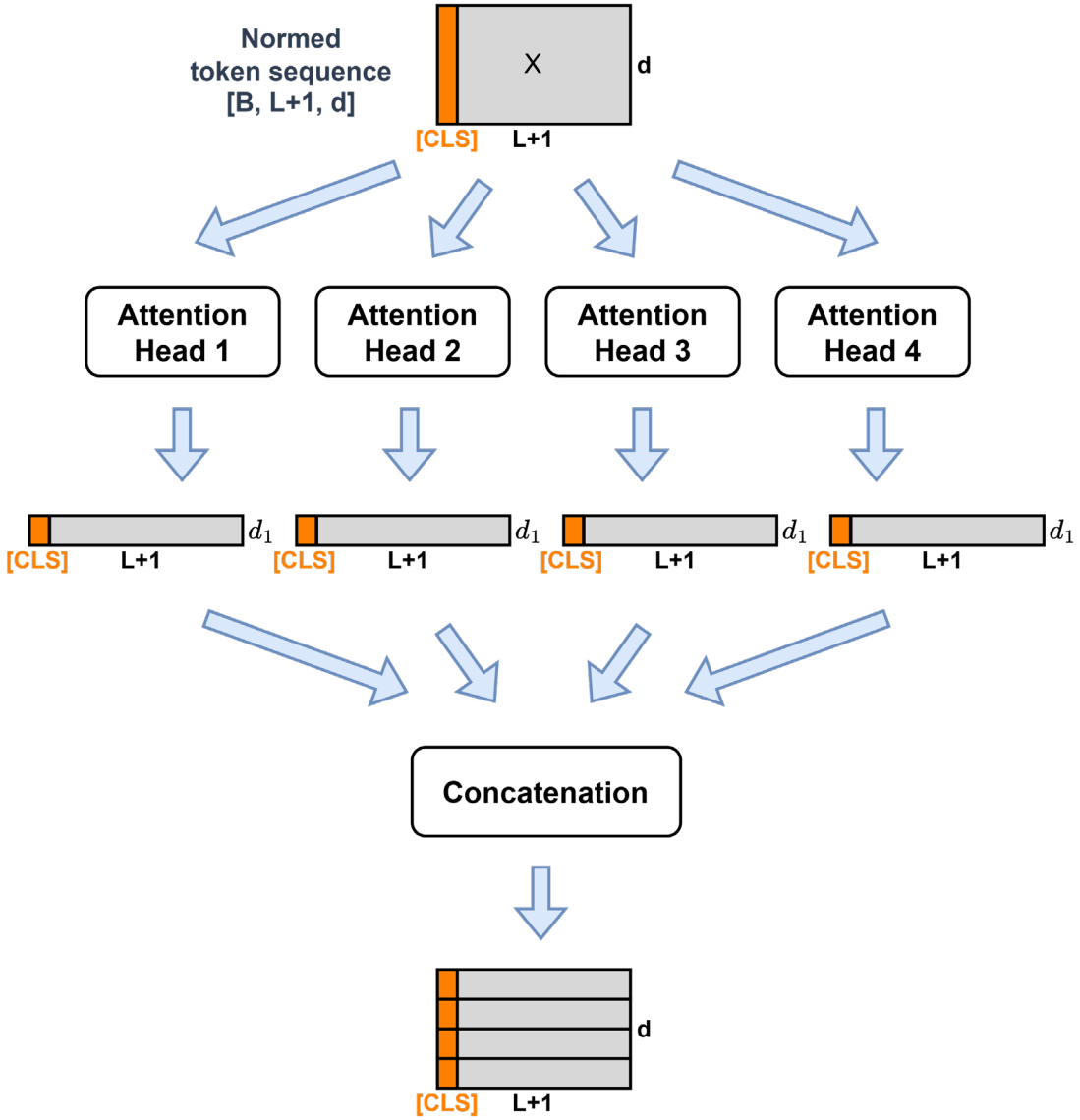


Encoder: Single Attention Head

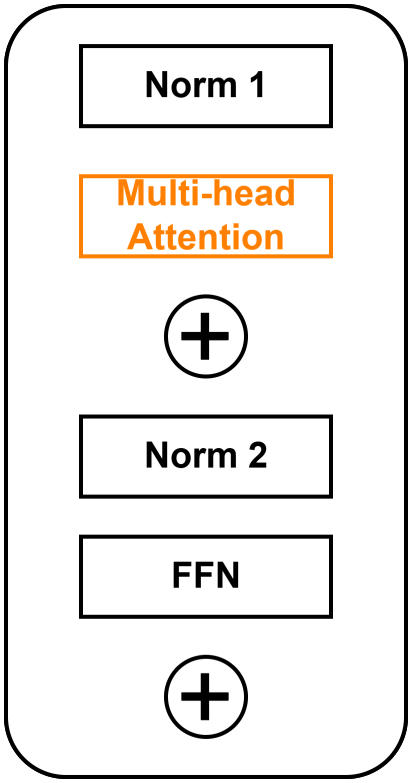


Encoder: Multi-head Attention (cont.)

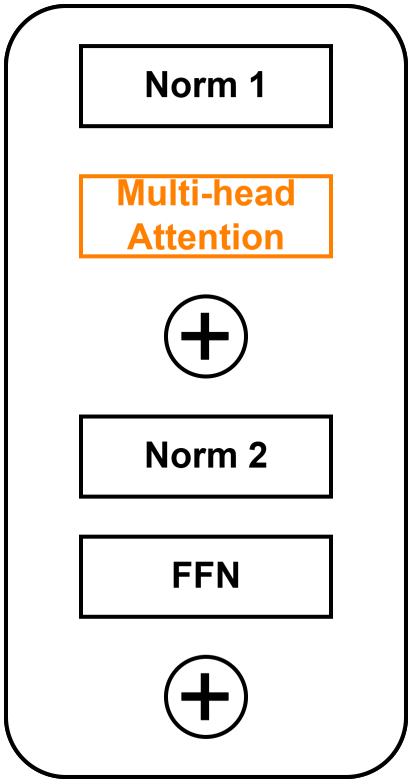


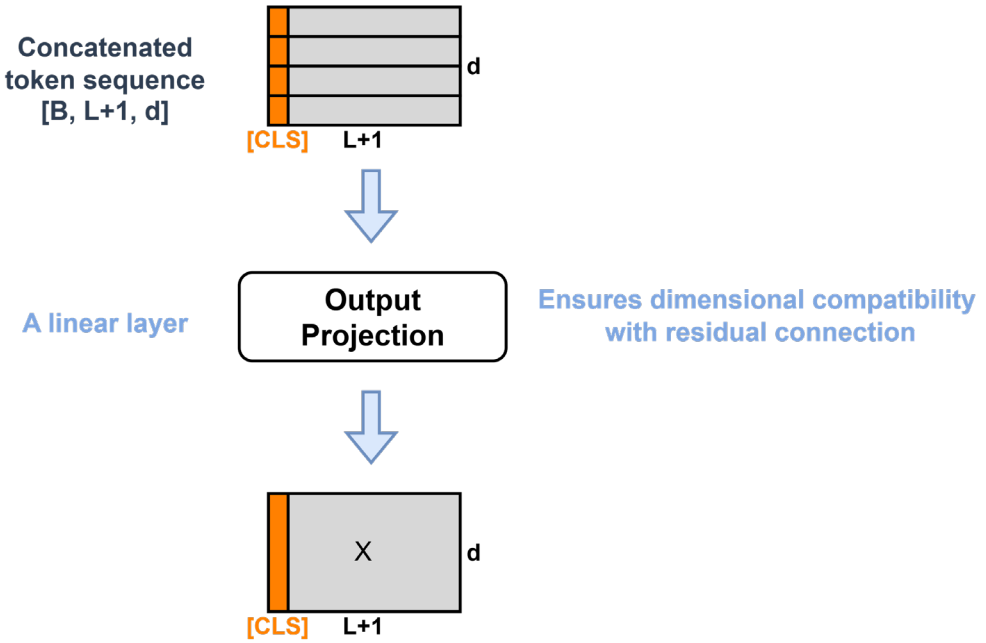


Encoder: Multi-head Attention

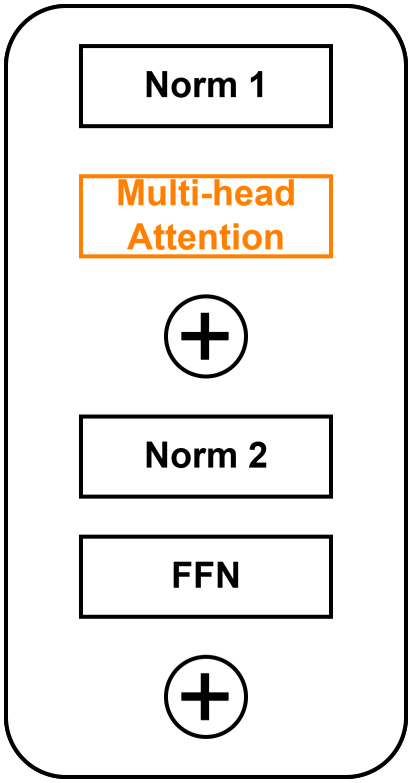


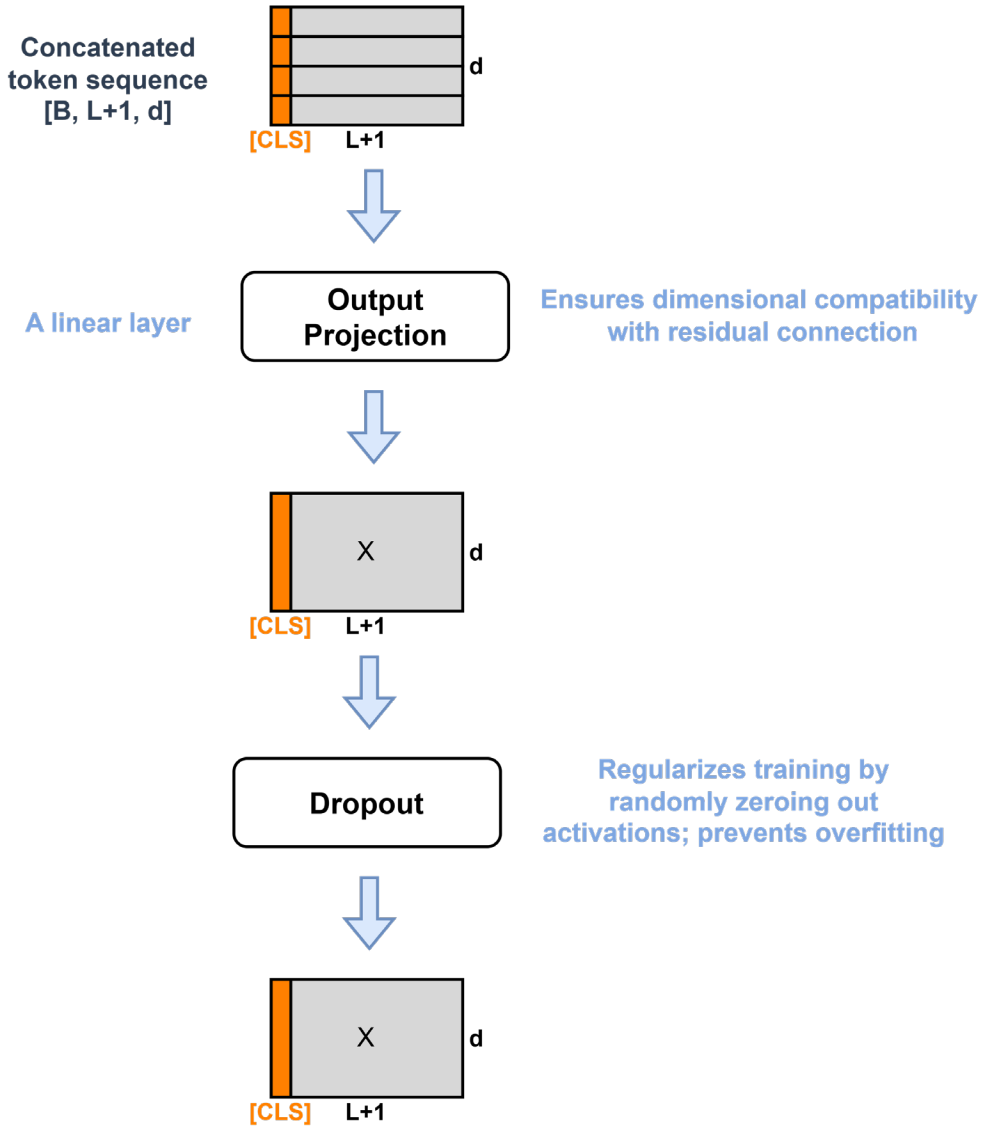
Encoder: Multi-head Attention



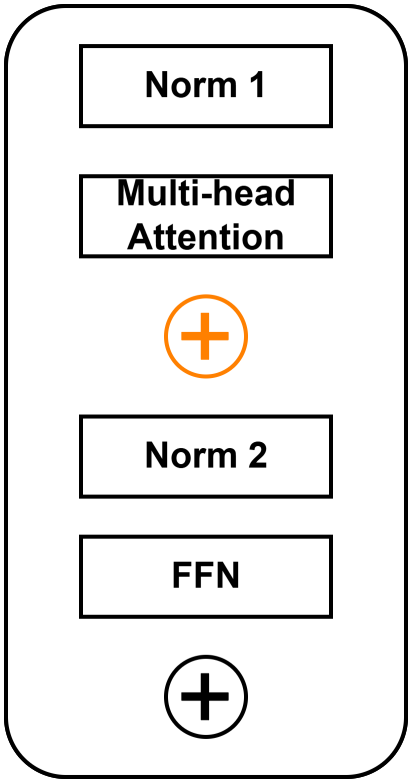


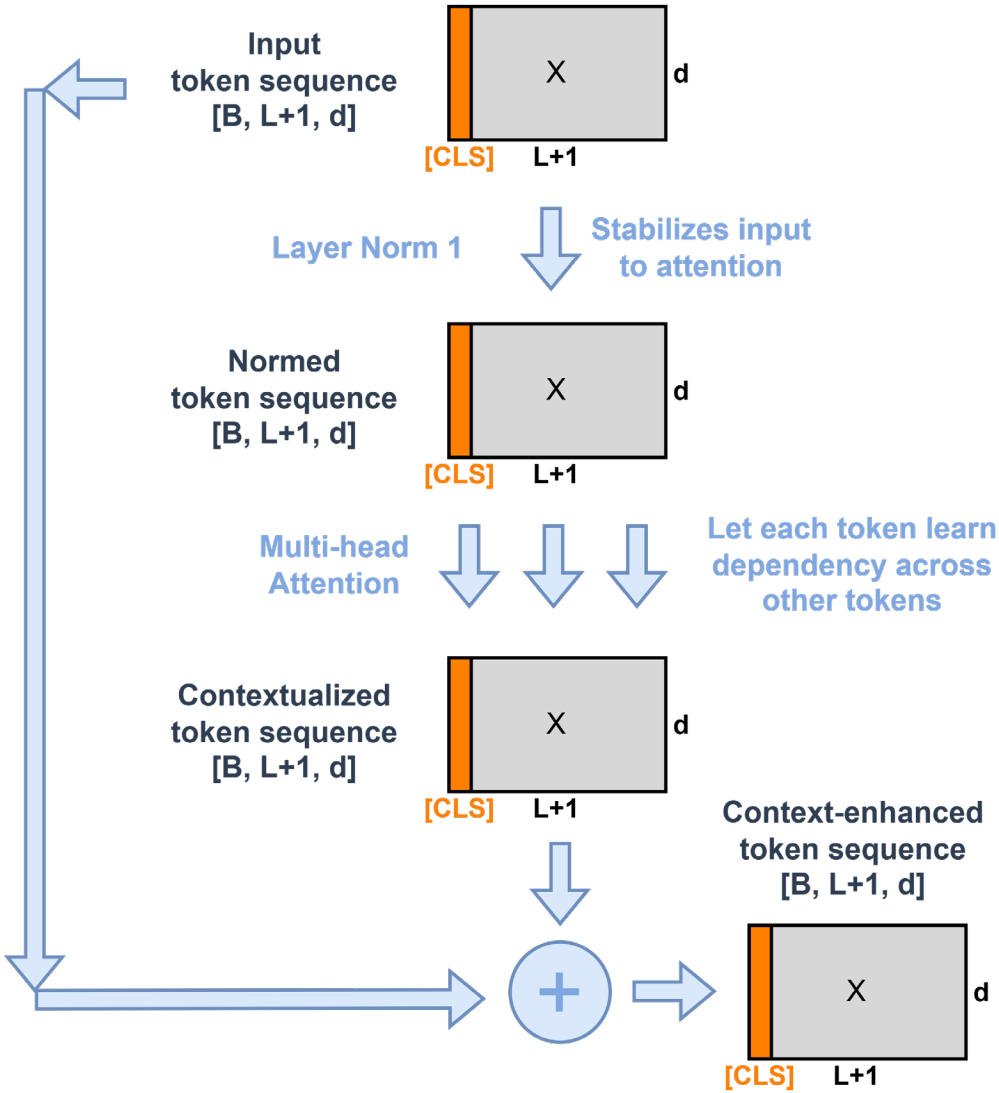
Encoder: Multi-head Attention



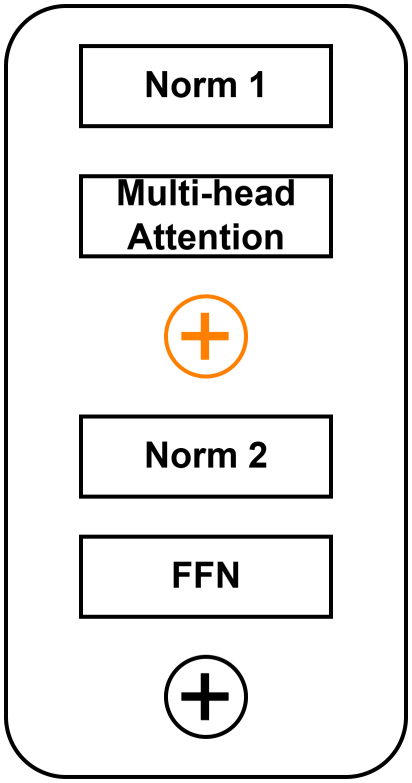


Encoder: Single Block (cont.)

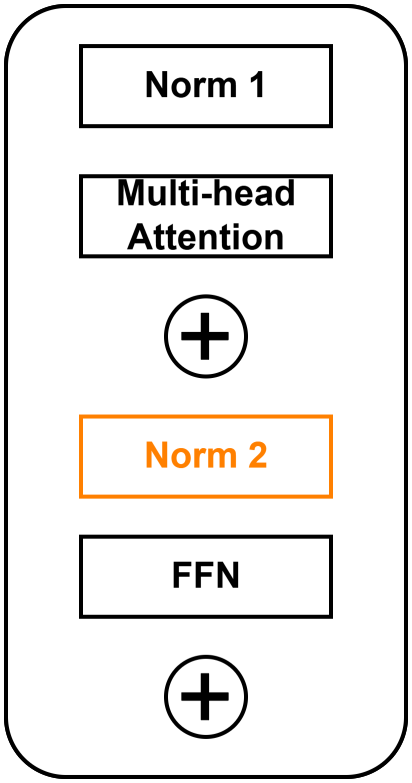


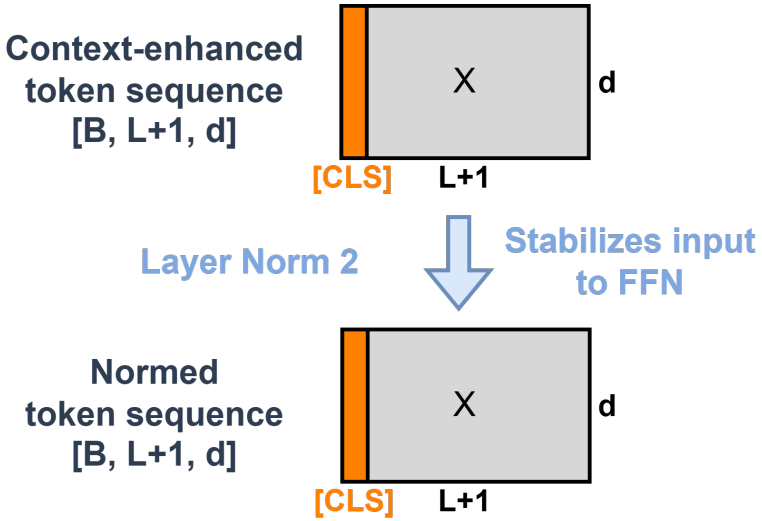


Encoder: Single Block

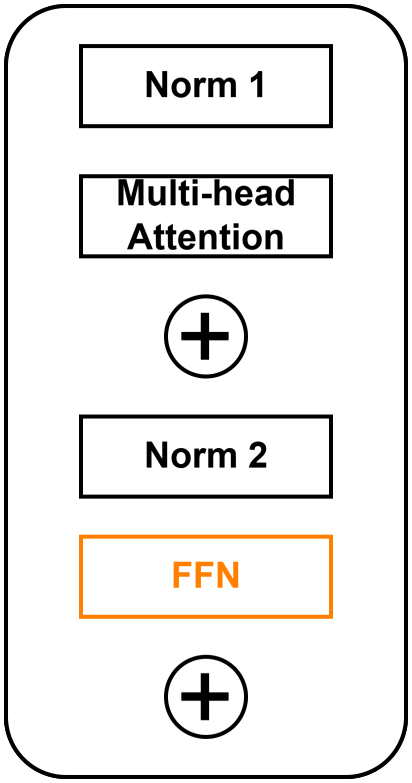


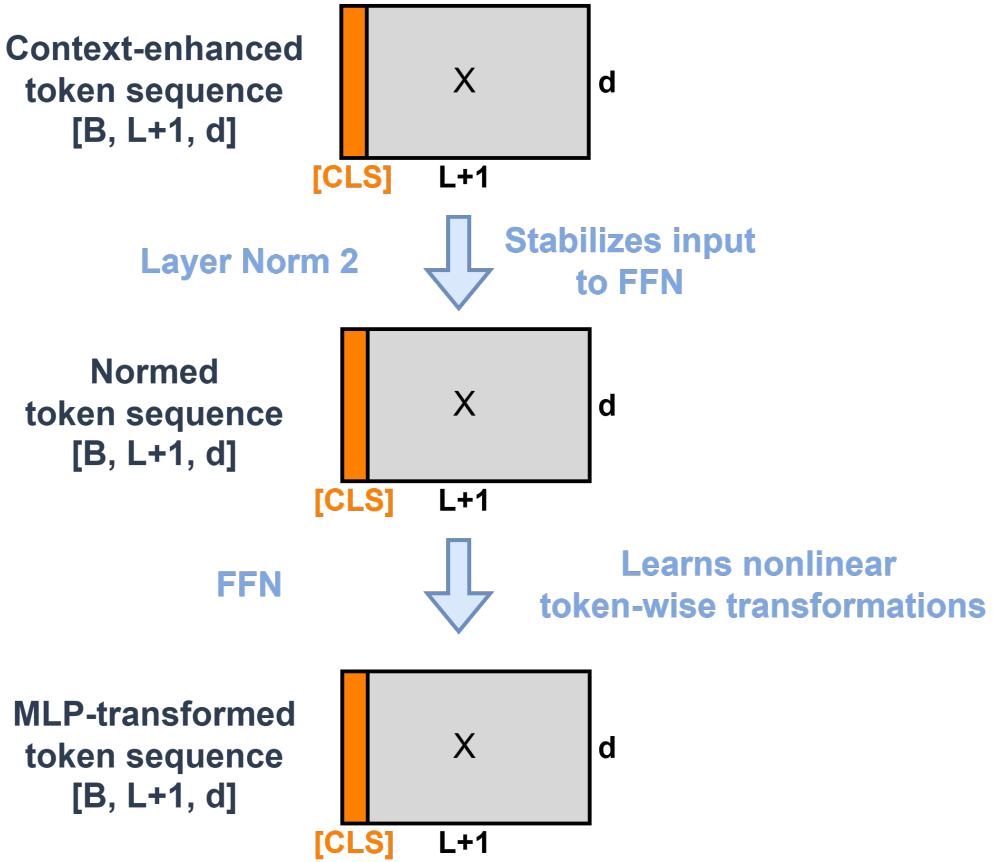
Encoder: Single Block



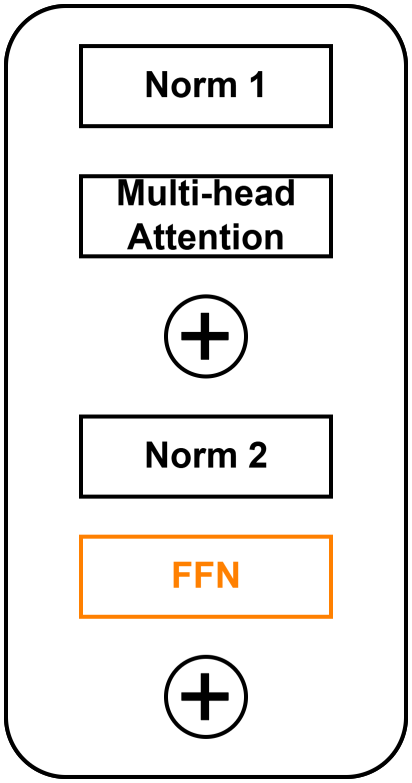


Encoder: Single Block



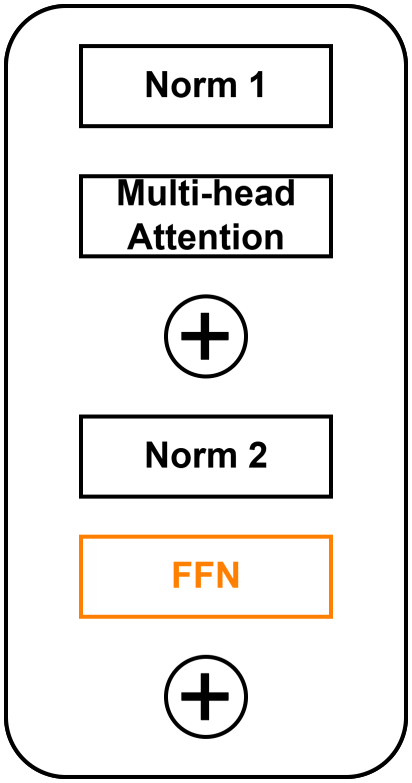


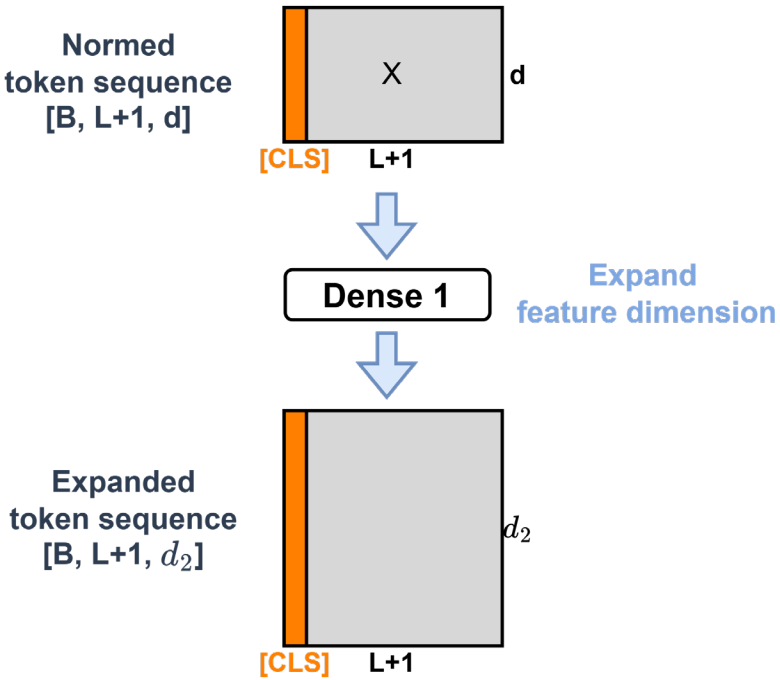
Encoder: FFN



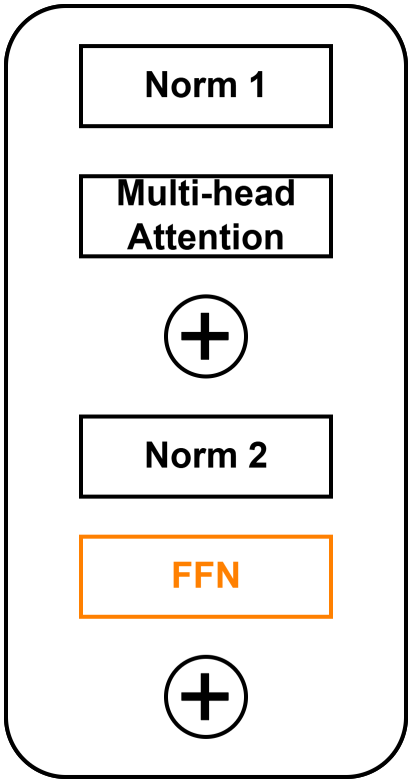
Normed token sequence [B, L+1, d]

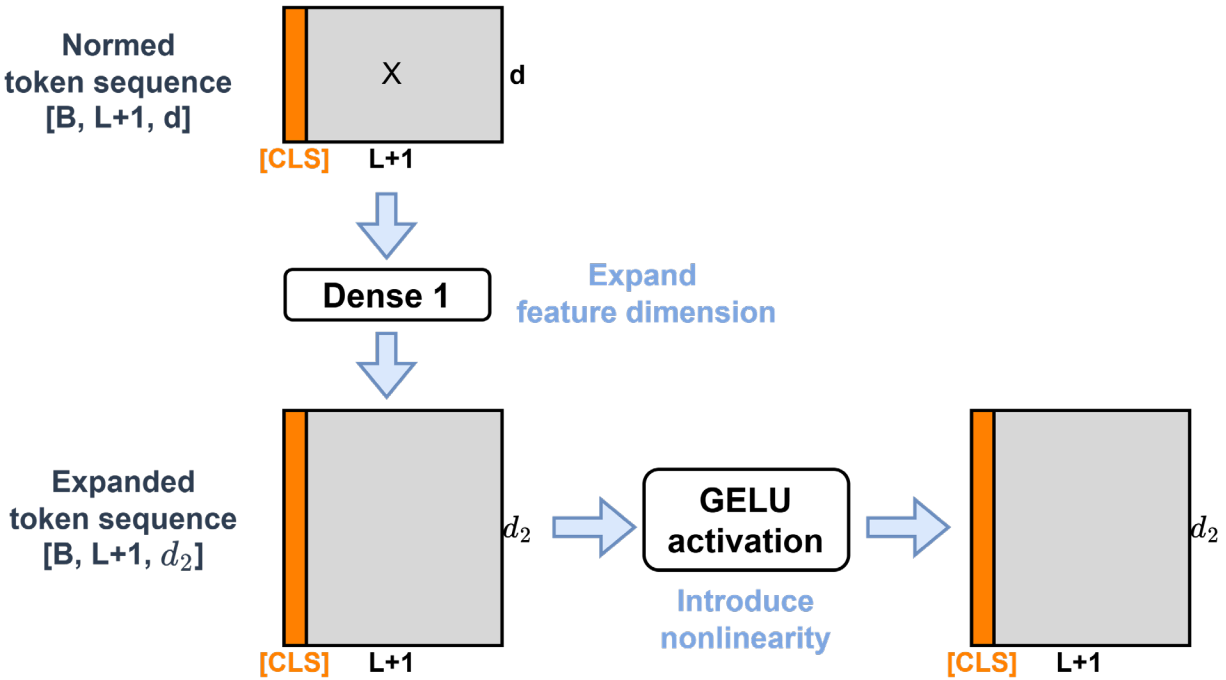
Encoder: FFN


Norm 1
Multi-head Attention
Norm 2
FFN

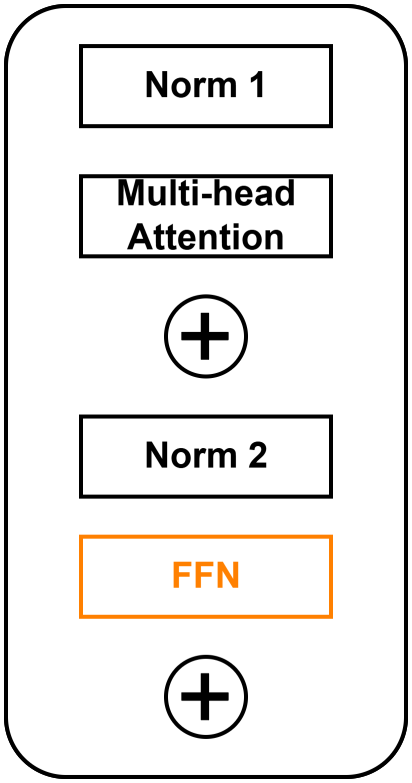

Normed token sequence [B, L+1, d]
X
d
[CLS]
L+1
Dense 1
Expand feature dimension
Expanded token sequence [B, L+1, d_2]
d_2
[CLS]
L+1

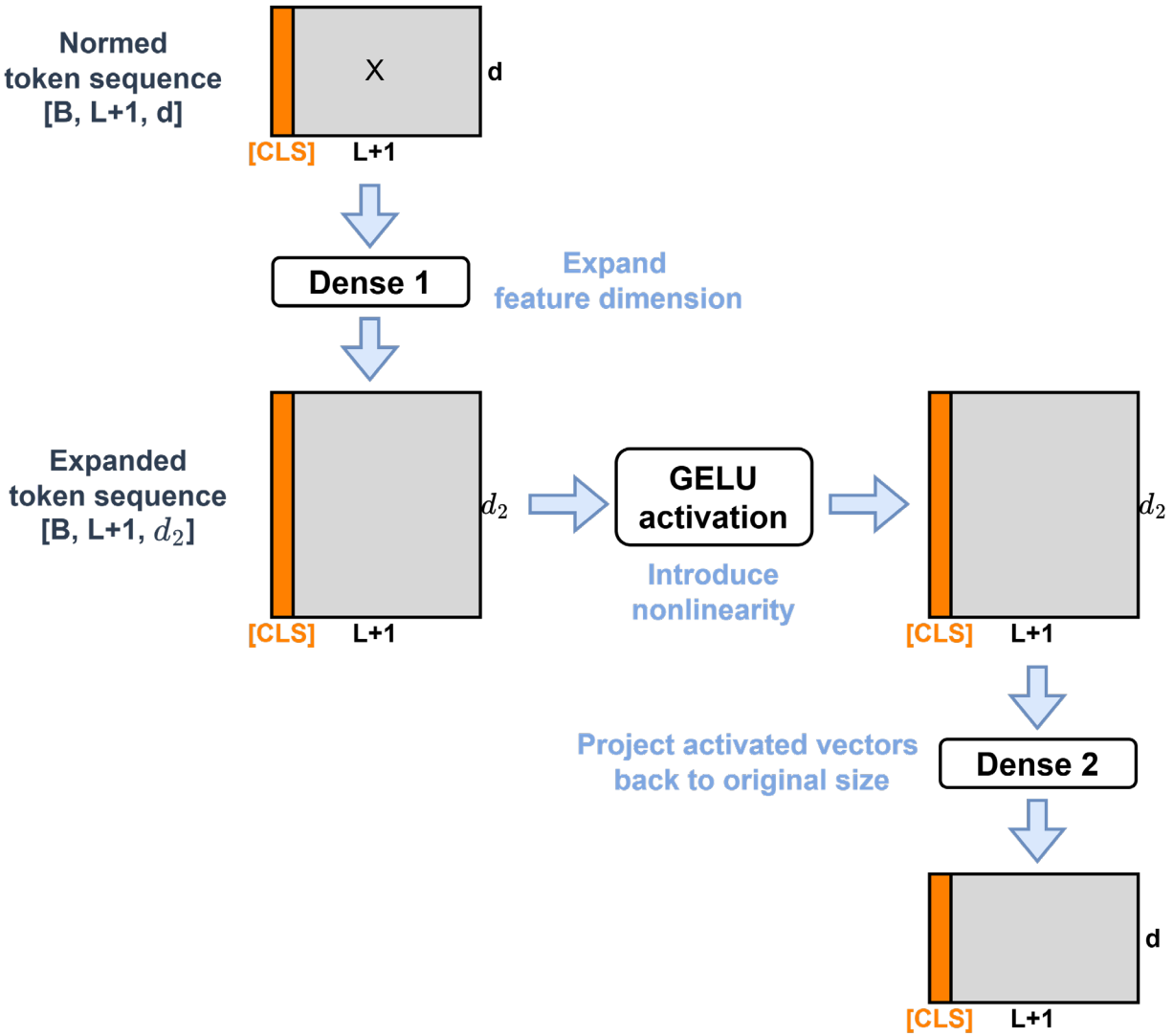
Encoder: FFN


Norm 1
Multi-head Attention
Norm 2
FFN

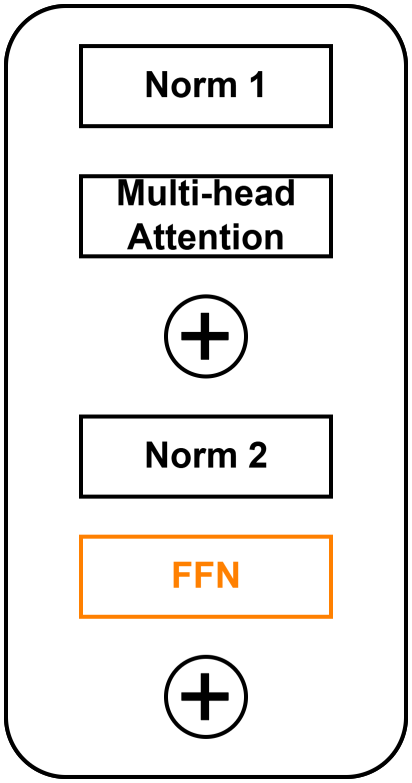

Normed token sequence [B, L+1, d]
X
d
[CLS]
L+1
Dense 1
Expand feature dimension
Expanded token sequence [B, L+1, d_2]
d_2
GELU activation
Introduce nonlinearity
d_2
[CLS]
L+1

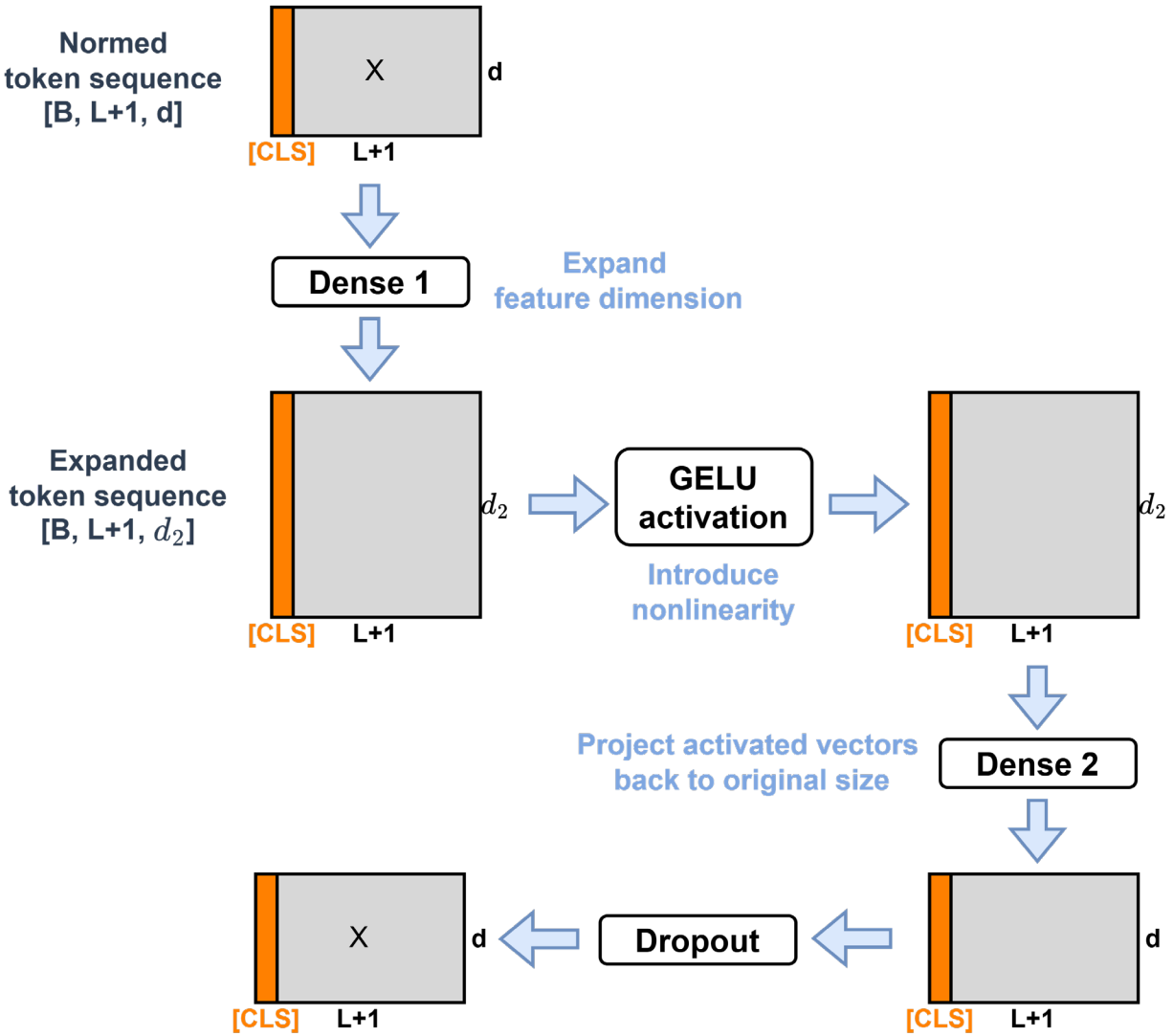
Encoder: FFN


Norm 1
Multi-head Attention
Norm 2
FFN

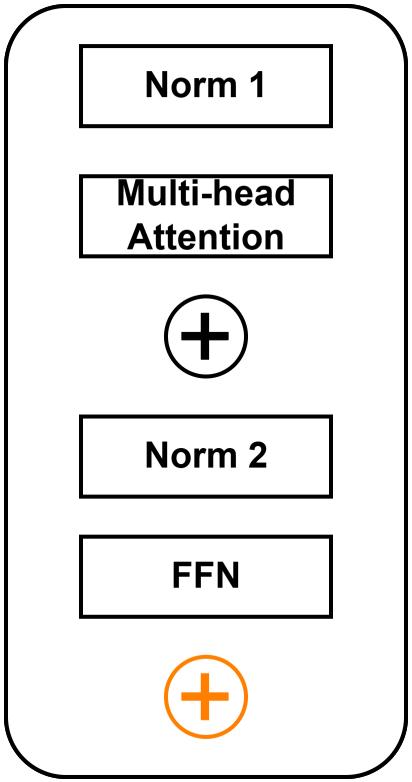

Normed token sequence [B, L+1, d]
X
d
[CLS]
L+1
Dense 1
Expand feature dimension
Expanded token sequence [B, L+1, d_2]
d_2
GELU activation
Introduce nonlinearity
d_2
[CLS]
L+1
Project activated vectors back to original size
Dense 2
d
[CLS]
L+1

Encoder: FFN


Norm 1
Multi-head Attention
Norm 2
FFN


Normed token sequence [B, L+1, d]
X
d
[CLS]
L+1
Dense 1
Expand feature dimension
Expanded token sequence [B, L+1, d_2]
d_2
GELU activation
Introduce nonlinearity
d_2
[CLS]
L+1
[CLS]
L+1
Project activated vectors back to original size
Dense 2
X
d
Dropout
d
[CLS]
L+1
[CLS]
L+1

Encoder: Single Block (cont.)



Context-enhanced token sequence [B, L+1, d]

X

d

[CLS] L+1

Layer Norm 2

Stabilizes input to FFN

Normed token sequence [B, L+1, d]

X

d

[CLS] L+1

FFN

Learns nonlinear token-wise transformations

MLP-transformed token sequence [B, L+1, d]

X

d

[CLS] L+1

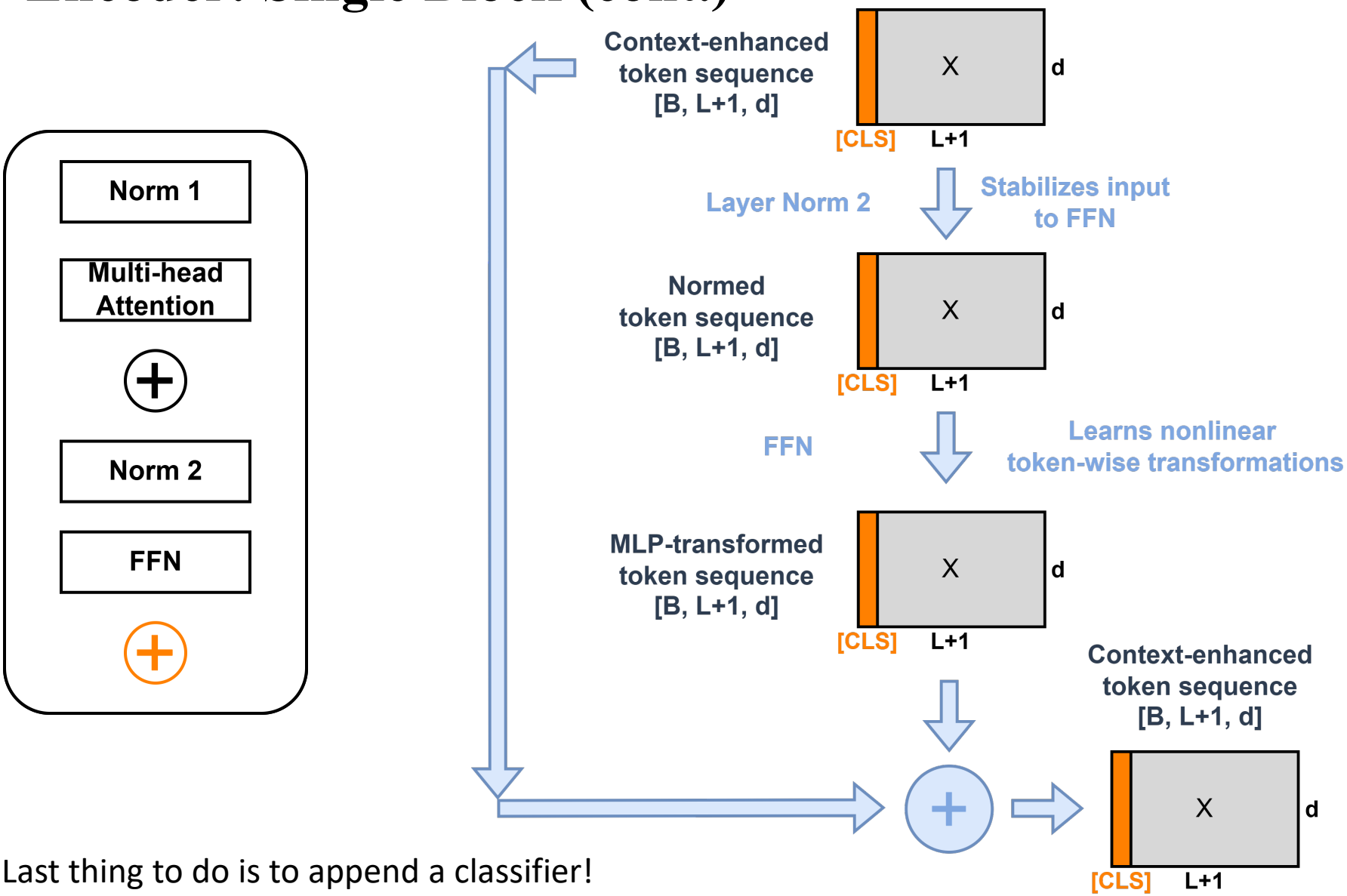
Context-enhanced token sequence [B, L+1, d]

X

d

[CLS] L+1

Encoder: Single Block (cont.)



Last thing to do is to append a classifier!

Lab Demo: Sequence classification using a Transformer encoder

https://github.com/williamqwu/ml-tutorials-suite

Part III: Adapting Transformers for Vision Task

Patch Embedding for Images

Figure from: Li and Shapovalenko (2024)

Vision Transformer (ViT) Model Overview

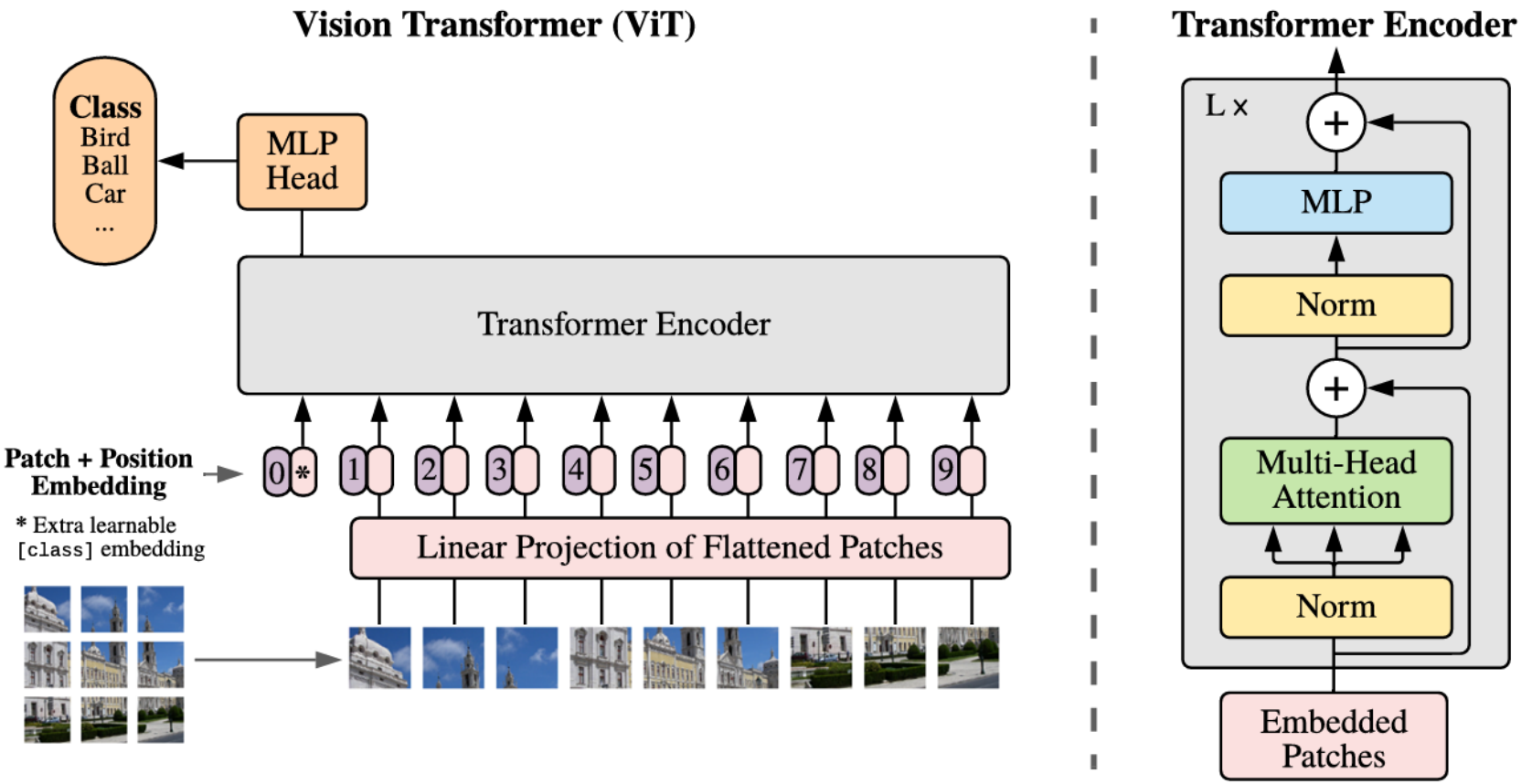


Figure from: https://arxiv.org/pdf/2010.11929v2

Lab Demo: Vision Transformer (ViT) on image patches

https://github.com/williamqwu/ml-tutorials-suite

рахмат
danke
謝謝
ngiyabonga
teşekkür ederim
Баярлалаа
spas
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
gracias
хвала
kiitos
dankie
nanni
nandri
dhanyavad
thank you
asante
manana
obrigada
tenki
bedankt
enkosi
bayarlalaa
gràcie
hvala
mauruuru
köszönöm
ačiū
dankon
akun
djiere dieuf
tau
mochchakkeram
murakoze
дякую
mamnun
chokrane
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
děkuji
obrigado
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
sagolun
najis tuke
ありがとう
tanemirt
grazie
terima kasih
kam sah hamnida
rahmat
rahmet
diolch
dhanyavadagalu
shukriya
mercê
мерси
धन्यवाद
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci