Cross attention

Queries, keys, and values

$$\{\boldsymbol{q}_i\}_{i=1}^{t} \rightsquigarrow \boldsymbol{Q} \in \mathbb{R}^{d' \times t}$$

$$\boldsymbol{q} = \boldsymbol{W_q x}, \quad \boldsymbol{k} = \boldsymbol{W_k} \overset{\boldsymbol{\xi}}{\cancel{\boldsymbol{x}}}, \quad \boldsymbol{v} = \boldsymbol{W_v} \overset{\boldsymbol{\xi}}{\cancel{\boldsymbol{x}}} \qquad \beta = \frac{1}{\sqrt{d'}}$$

$$\boldsymbol{q}, \boldsymbol{k} \in \mathbb{R}^{d'}, \quad \boldsymbol{v} \in \mathbb{R}^{d''}$$

$$\{\boldsymbol{\xi}_j\}_{j=1}^{\tau} \rightsquigarrow \{\boldsymbol{k}_j\}_{j=1}^{\tau}, \{\boldsymbol{v}_j\}_{j=1}^{\tau} \rightsquigarrow \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{\{d', d''\} \times \tau}$$

$$\boldsymbol{a} = \operatorname{softargmax}_{\beta}(\boldsymbol{K}^{\top} \boldsymbol{q}) \in \mathbb{R}^{\tau} \qquad \boldsymbol{h} = \boldsymbol{V a} \in \mathbb{R}^{d''}$$

$$\{\boldsymbol{q}_i\}_{i=1}^{t} \rightsquigarrow \{\boldsymbol{a}_i\}_{i=1}^{t} \rightsquigarrow \boldsymbol{A} \in \mathbb{R}^{\tau \times t} \qquad \boldsymbol{H} = \boldsymbol{V A} \in \mathbb{R}^{d'' \times t}$$

Self attention

$$d' = d'' \overset{\downarrow}{\doteq} d$$

Implementation

from the *RNN* lecture

$$\boldsymbol{h}[t] = g\left(\boldsymbol{W_h} \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b_h}\right)$$

$$\boldsymbol{h}[0] \doteq \boldsymbol{0}, \boldsymbol{W_h} \doteq \begin{bmatrix} \boldsymbol{W_{hx}} & \boldsymbol{W_{hh}} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{k} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W_q} \\ \boldsymbol{W_k} \\ \boldsymbol{W_v} \end{bmatrix} \boldsymbol{x} \in \mathbb{R}^{3d}$$

considering h heads we get a vector in $\mathbb{R}^{3hd}$

using a $\boldsymbol{W_h} \in \mathbb{R}^{d\times hd}$ to go back to $\mathbb{R}^{d}$

$$\begin{bmatrix} \boldsymbol{q}^1 \\ \boldsymbol{q}^2 \\ \vdots \\ \boldsymbol{q}^h \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{\boldsymbol{q}}^1 \\ \boldsymbol{W}_{\boldsymbol{q}}^2 \\ \vdots \\ \boldsymbol{W}_{\boldsymbol{q}}^h \end{bmatrix} \boldsymbol{x} \quad \begin{bmatrix} \boldsymbol{k}^1 \\ \boldsymbol{k}^2 \\ \vdots \\ \boldsymbol{k}^h \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{\boldsymbol{k}}^1 \\ \boldsymbol{W}_{\boldsymbol{k}}^2 \\ \vdots \\ \boldsymbol{W}_{\boldsymbol{k}}^h \end{bmatrix} \boldsymbol{x} \quad \begin{bmatrix} \boldsymbol{v}^1 \\ \boldsymbol{v}^2 \\ \vdots \\ \boldsymbol{v}^h \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{\boldsymbol{v}}^1 \\ \boldsymbol{W}_{\boldsymbol{v}}^2 \\ \vdots \\ \boldsymbol{W}_{\boldsymbol{v}}^h \end{bmatrix} \boldsymbol{x}$$

Transformer

Encoders-predictor-decoder architecture

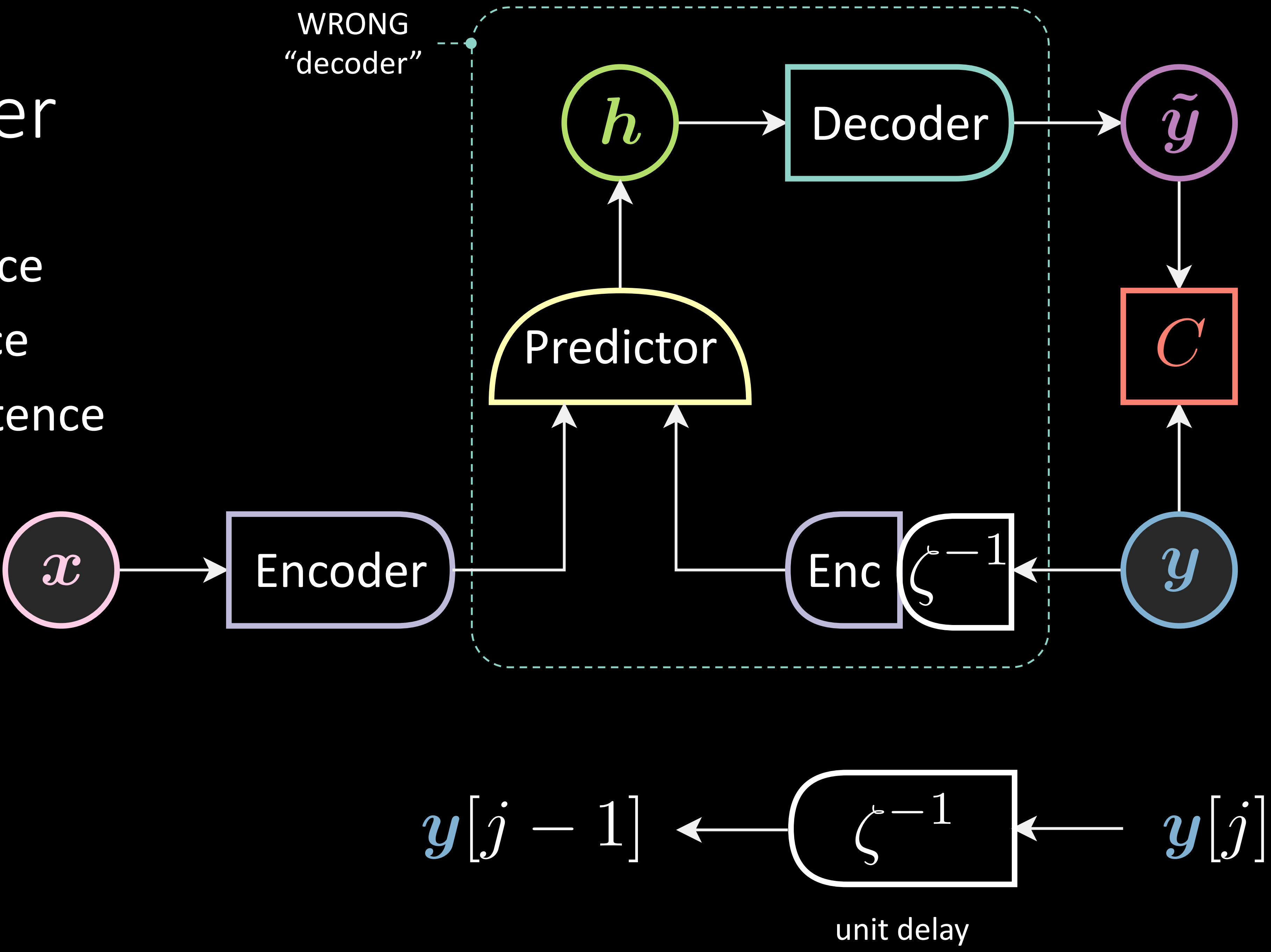
(for Neural Machine Translation)

Transformer

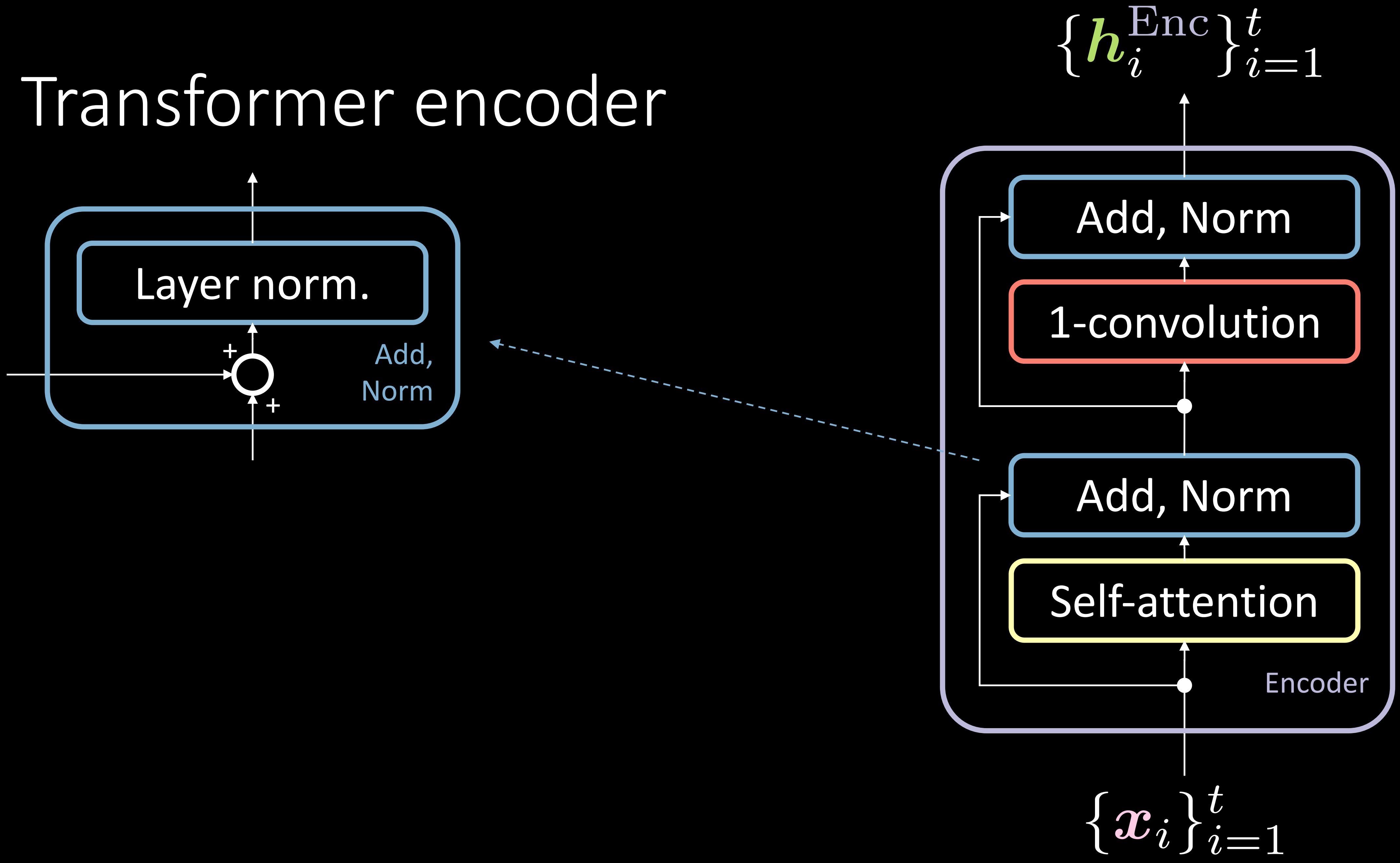
x source sentence

y target sentence

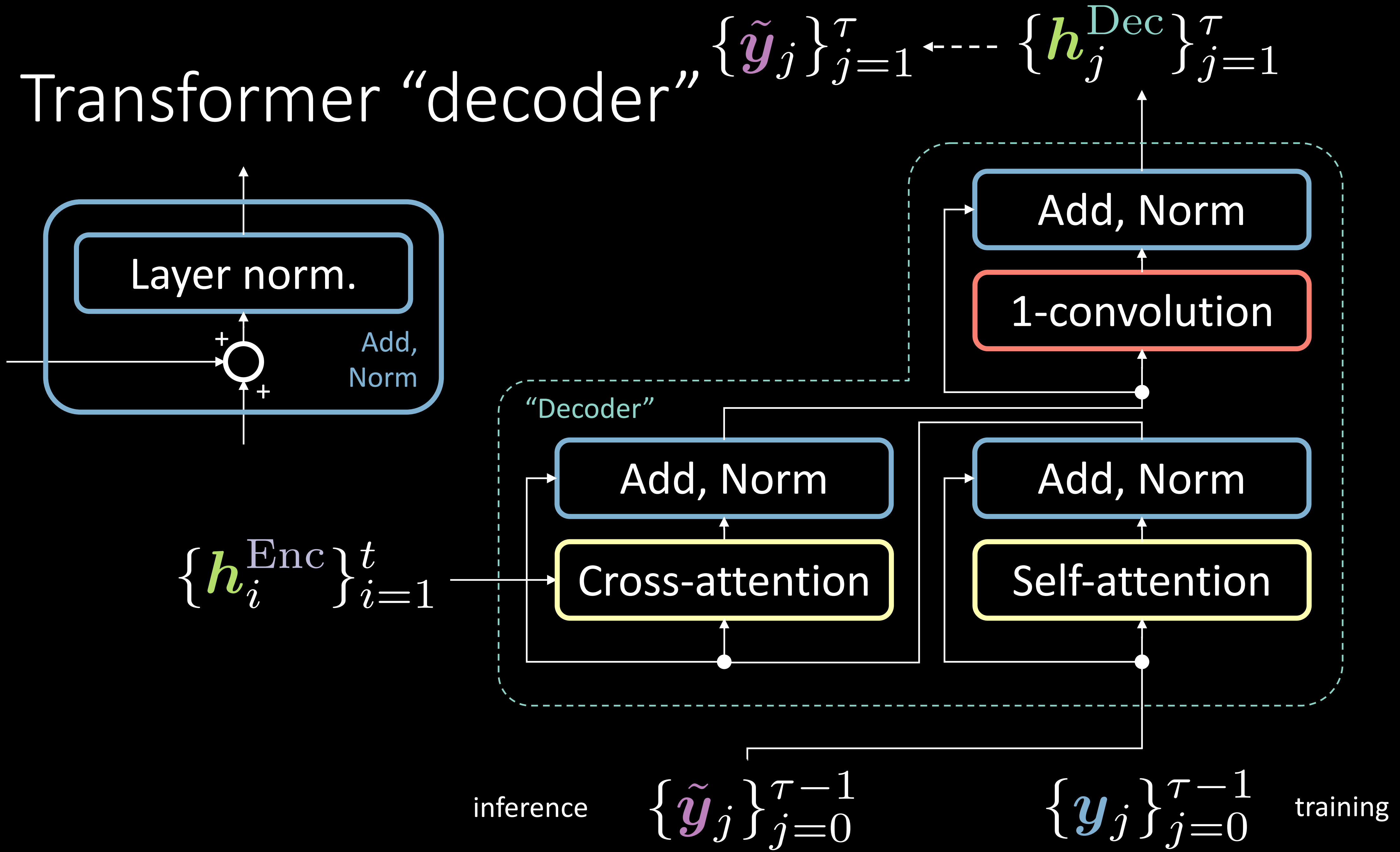
$\tilde{y}$ predicted sentence



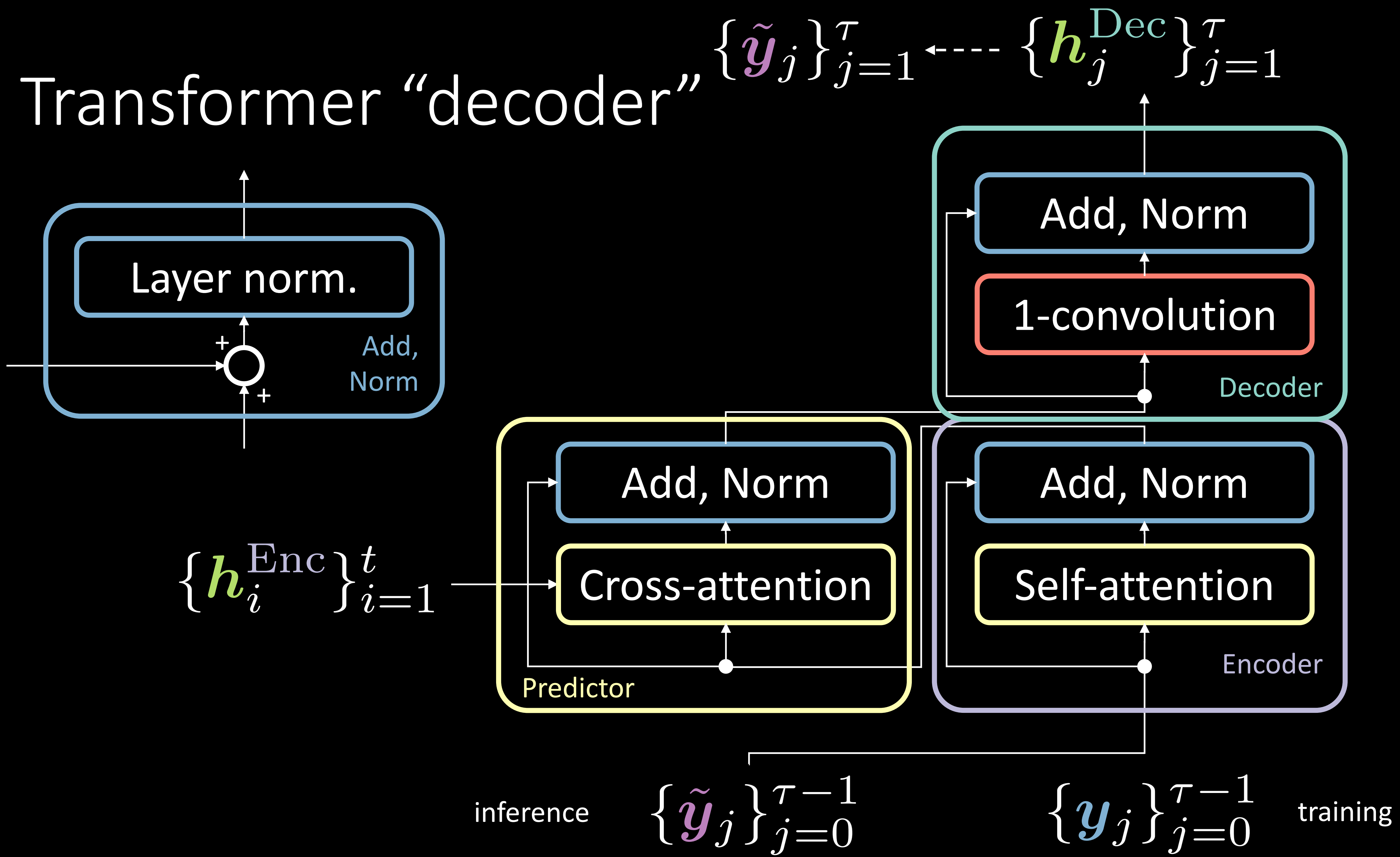
Transformer encoder


Layer norm.
+
+
Add,
Norm
$\{\boldsymbol{h}_i^{\text{Enc}}\}_{i=1}^{t}$
Add, Norm
1-convolution
Add, Norm
Self-attention
Encoder
$\{\boldsymbol{x}_i\}_{i=1}^{t}$

Transformer “decoder”



Transformer “decoder”


$\{\tilde{y}_j\}_{j=1}^{\tau}$
$\{h_j^{\text{Dec}}\}_{j=1}^{\tau}$
Layer norm.
+
+
Add,
Norm
Add, Norm
1-convolution
Decoder
Add, Norm
Cross-attention
Predictor
Add, Norm
Self-attention
Encoder
$\{h_i^{\text{Enc}}\}_{i=1}^{t}$
inference
$\{\tilde{y}_j\}_{j=0}^{\tau-1}$
$\{y_j\}_{j=0}^{\tau-1}$
training