TIME SERIES ANALYSIS & RECURRENT NEURAL NETWORKS #13

- Bayesian RNN
- Generative Adversarial Networks (GAN)
- Attention & Transformers

Main lecture: Daniel Durstewitz

Exercises: Leonard Bereska, Manuel Brenner, Daniel Kramer, Georgia Koppe

Heidelberg University

Bayesian RNN

$X = \{x_1 \dots x_T\}$

$p_\theta(x, z)$, $p_\theta(z|x)$

likelihood

$$p(\theta, z|x) = \frac{p(x|\theta, z)\, p(\theta, z)}{p(x)}$$

$p(\theta, z|x)$: posterior — $p(x|\theta, z)$: likelihood — $p(\theta, z)$: prior

$$p(x) = \int_\theta \int_z p(x, z, \theta)\, dz\, d\theta$$

→ Variational Inference

$$\arg\min_\zeta \{ KL[q_\zeta(\theta, z|x) \,\|\, p(\theta, z|x)]$$

$$= \int_{\theta, z} q_\zeta(\theta, z|x) \log \frac{q_\zeta(\theta, z|x)}{p(\theta, z|x)}\, d\theta\, dz$$

$$= -H_q[q_\zeta(\theta, z|x)] - E_q\left[\log \frac{p(\theta, z, x)}{p(x)}\right]$$

$$= -H_q[q] - E_q[\log p(\theta, z, x)] + E_q[\log p(x)]$$

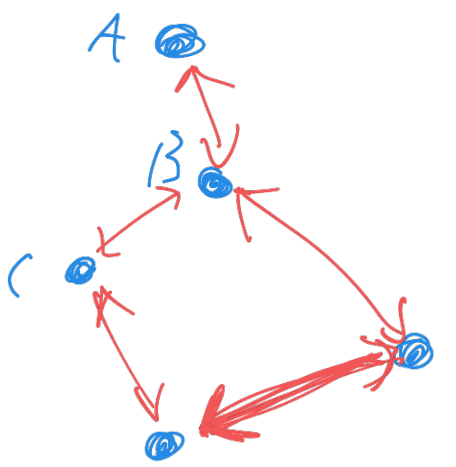
$$= -ELBO[\zeta] + \log p(x) \}$$

$q_\zeta(\theta, z|x) = q_\phi(\theta|x)\, q_\psi(z|x)$ assumpt.!

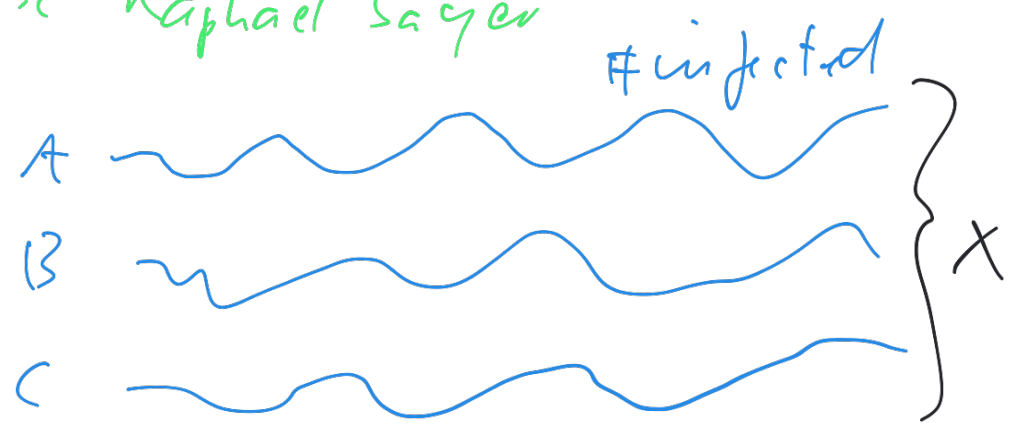
$\zeta = \{\phi, \psi\}$

→ $\arg\max_{\phi, \psi} E_{q_\psi}[ELBO_2(\phi, z) + \log p(\theta) - \log q_\psi(\theta|x)]$

- appendix Kingma & Welling (2014)
- MSc Raphael Sager



struct. prior



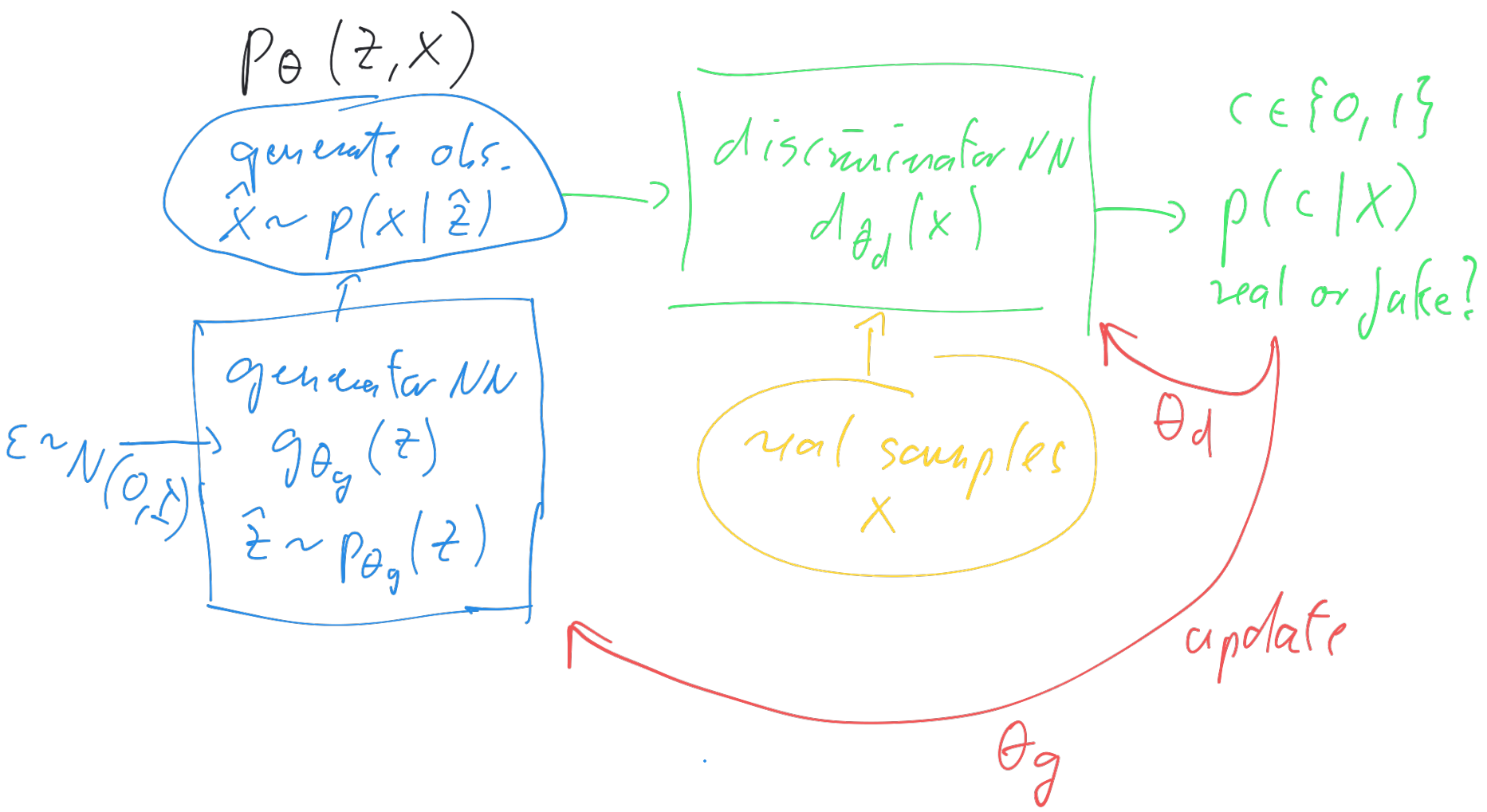
$z_t = A z_{t-1} + W\phi(z_{t-1}) + b$

$p(W)$ ⟶ $p(W, z|x)$

- previous experiments → $p(A, W, b)$

Generative Adversarial Networks (GAN)

Goodfellow et al. (2014)



→ g and d play a '0-sum' game

$v(\theta_d, \theta_g)$: payoff for discriminator

$-v(\theta_d, \theta_g)$: payoff for generator

e.g. $v(\theta_d, \theta_g) = \mathbb{E}_{x \sim p_{data}}[\log d_{\theta_d}(x)] + \mathbb{E}_{\hat{x} \sim p_{gen}}[\log(1 - d_{\theta_d}(x))]$

$p(c=1|x)$

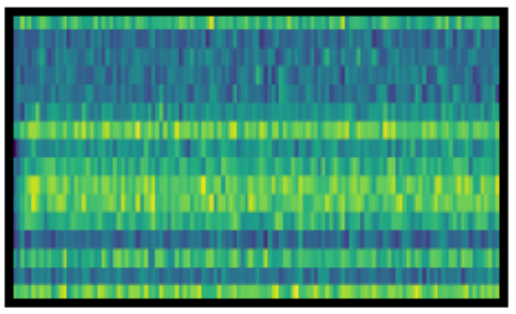
$c=1$: real

$c=0$: fake

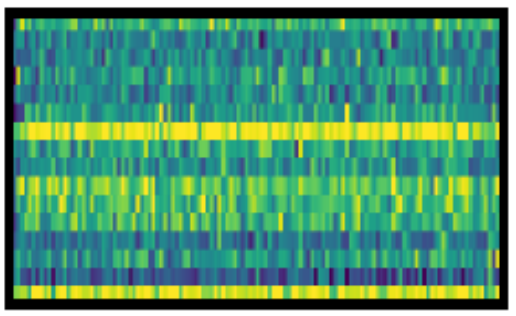
$$\theta_g^* = \arg\min_{\theta_g} \left[\max_{\theta_d} [v(\theta_d, \theta_g)] \right]$$

→ data augmentation

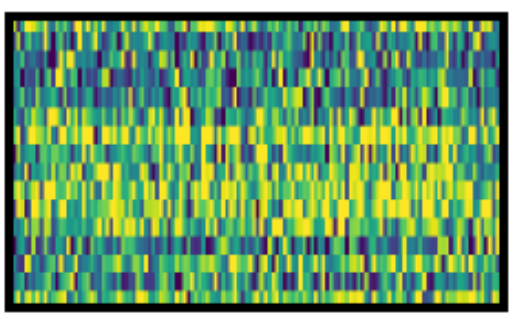
real data



Spike-GAN

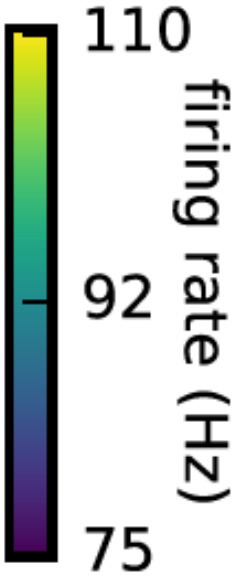


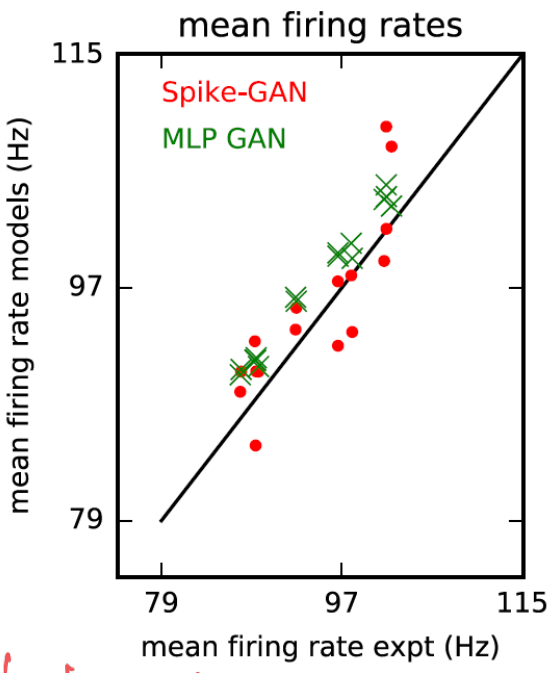
MLP GAN

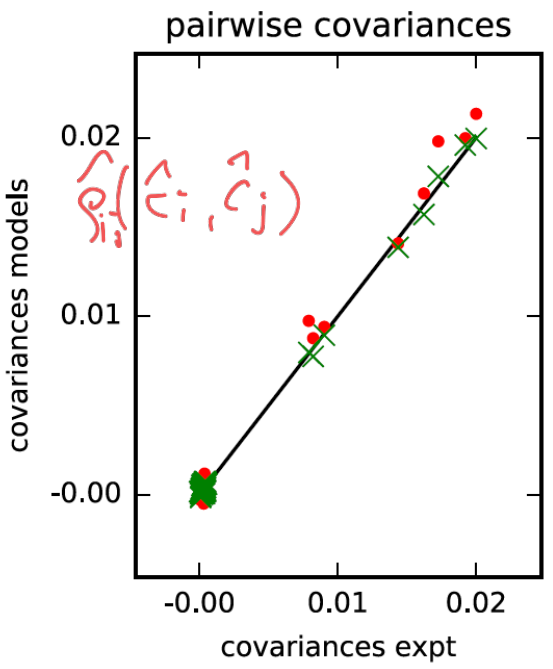


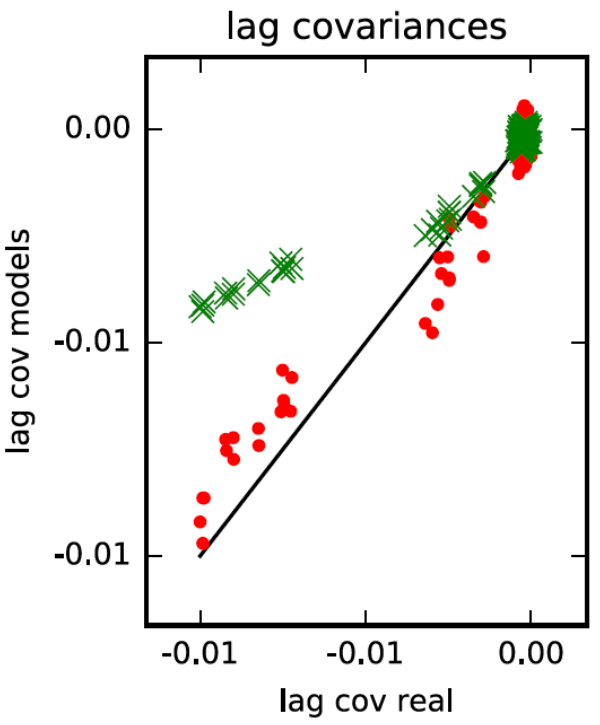
neuron

time









Molano-Mazon ... Panzeri (2018) ICLR

Attention mechanisms

language models
- machine translation
- Query

- Representation of language
  - categ. distr., "1-hot-encoding"
    vocab. V of K words

$$c_t = (0\,0\,0\,\underset{\uparrow}{1}\,0\,0\,0\,\ldots\,0) \in \{0,1\}^K$$

$\uparrow$ i^{th} word in V

$$\sum c_{it} = 1, \quad \{c_1 \ldots c_T\}$$

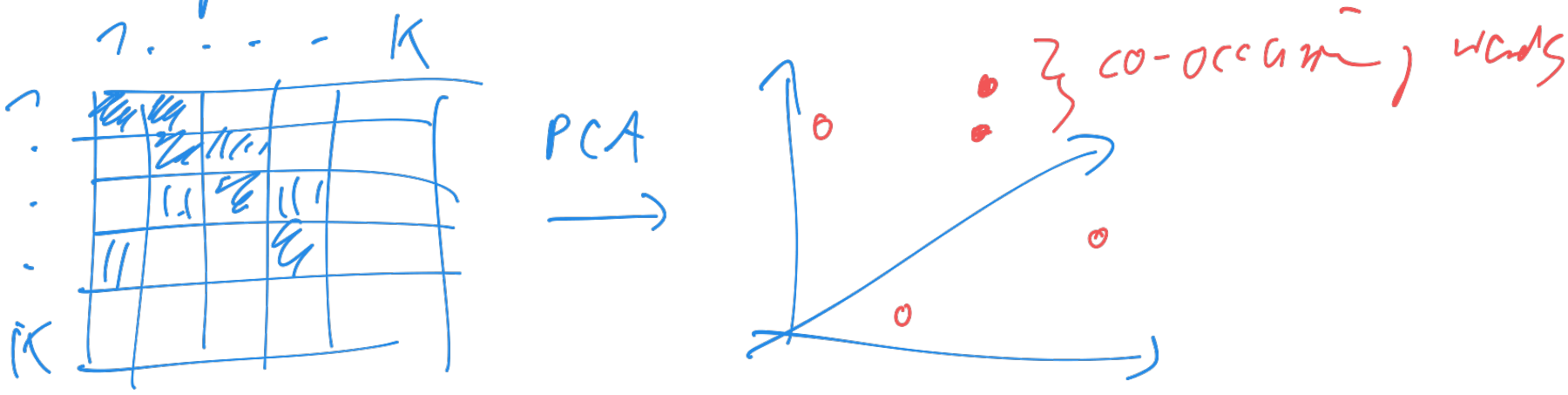
$$\rightarrow p(c_t) = \prod_{i=1}^{K} \pi_i^{c_{it}}$$

output from NN

$$\pi_i = \text{Softmax}[x_i] = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

- word embedding: - contin. repr.
  - lower-dim.
  - acknowledge context

e.g. PCA on word cov. matrix

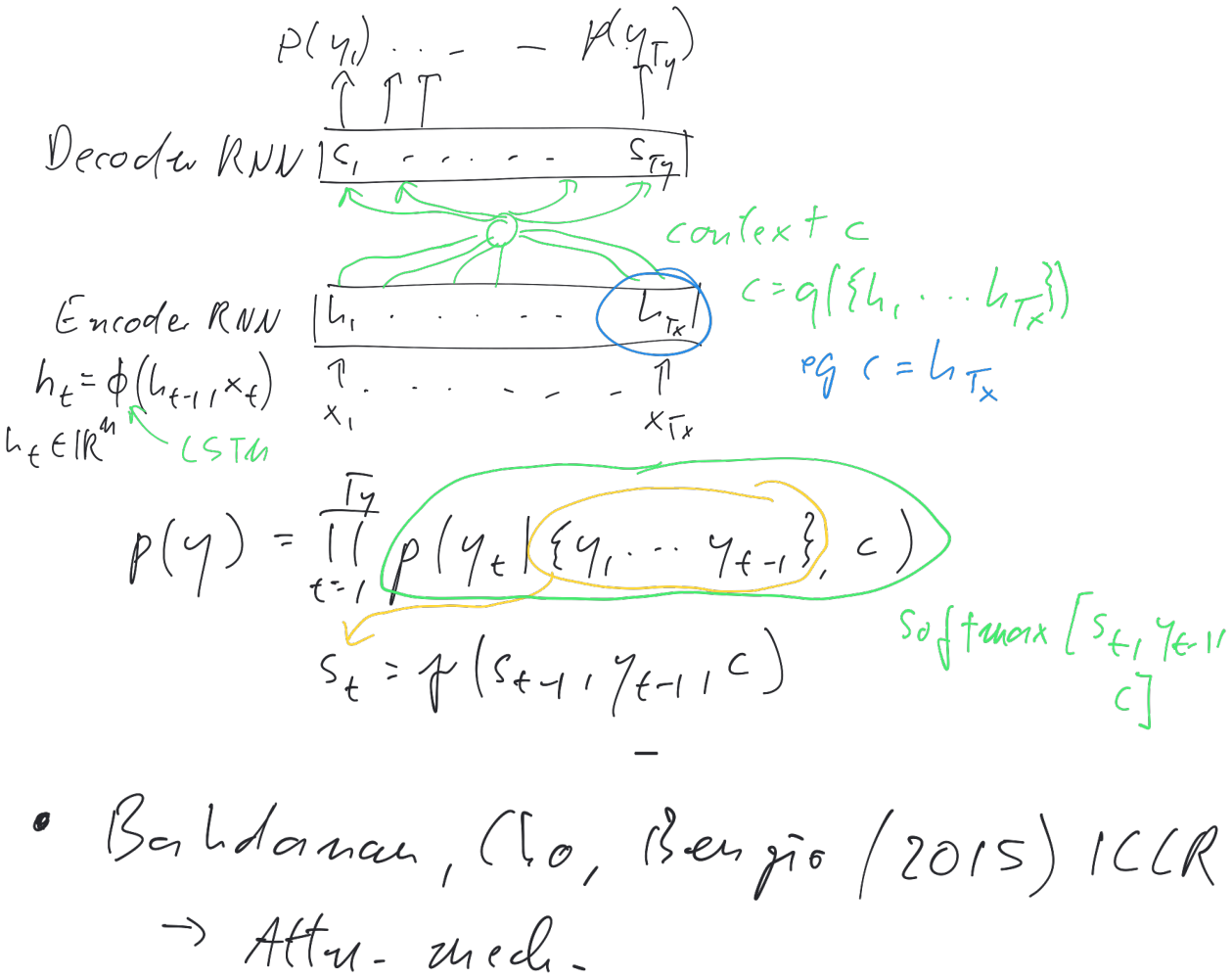


$$y_i = \underbrace{A}_{d \times K,\ d \ll K} x_i$$

"classical approach" (Sutskever, Graves, Bengio)

$(x_1 \dots x_{T_x}) \rightarrow (y_1 \dots y_{T_y})$

source language query — target language answer



$$p(y) = \prod_{t=1}^{T_y} p(y_t | \{y_1 \dots y_{t-1}\}, c)$$

$$s_t = f(s_{t-1}, y_{t-1}, c)$$

Softmax $[s_{t-1}, y_{t-1}, c]$

- Bahdanau, Cho, Bengio (2015) ICLR
  → Attn. mech.

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

$$c_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j$$

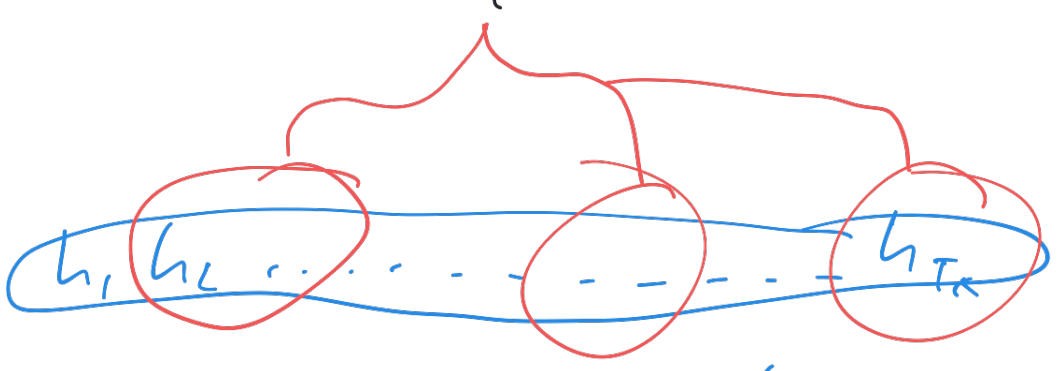
"alignment, conn. attn. weights"

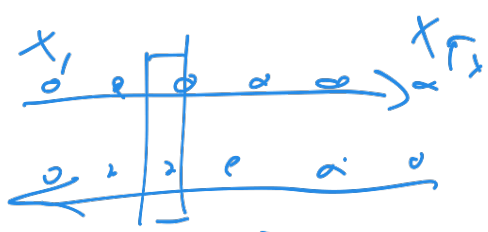
$$\alpha_{tj} = \text{Softmax}(e_{tj}) = \frac{\exp(e_{tj})}{\sum_{k=1}^{T_x} \exp(e_{tk})}$$

"relevance / attn. func." $e_{tj} = g(h_j, s_{t-1})$

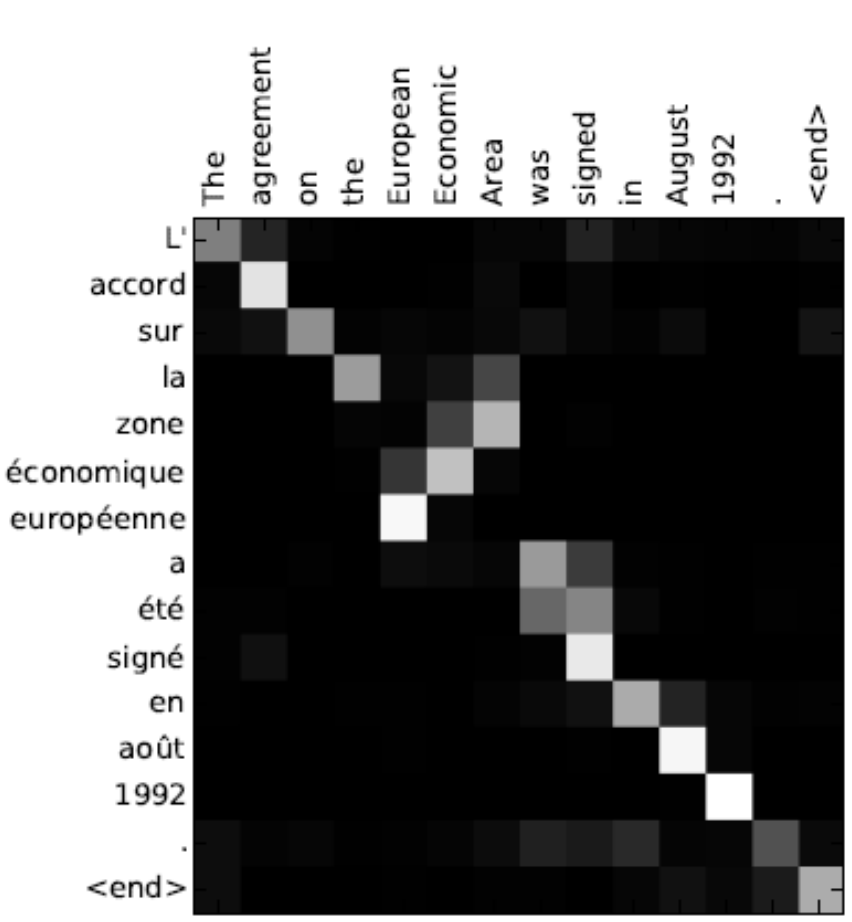
$$= \underbrace{v_a^T \tanh(W_a s_{t-1} + U_a h_j)}_{\text{1-hidd-layer FNN}}$$

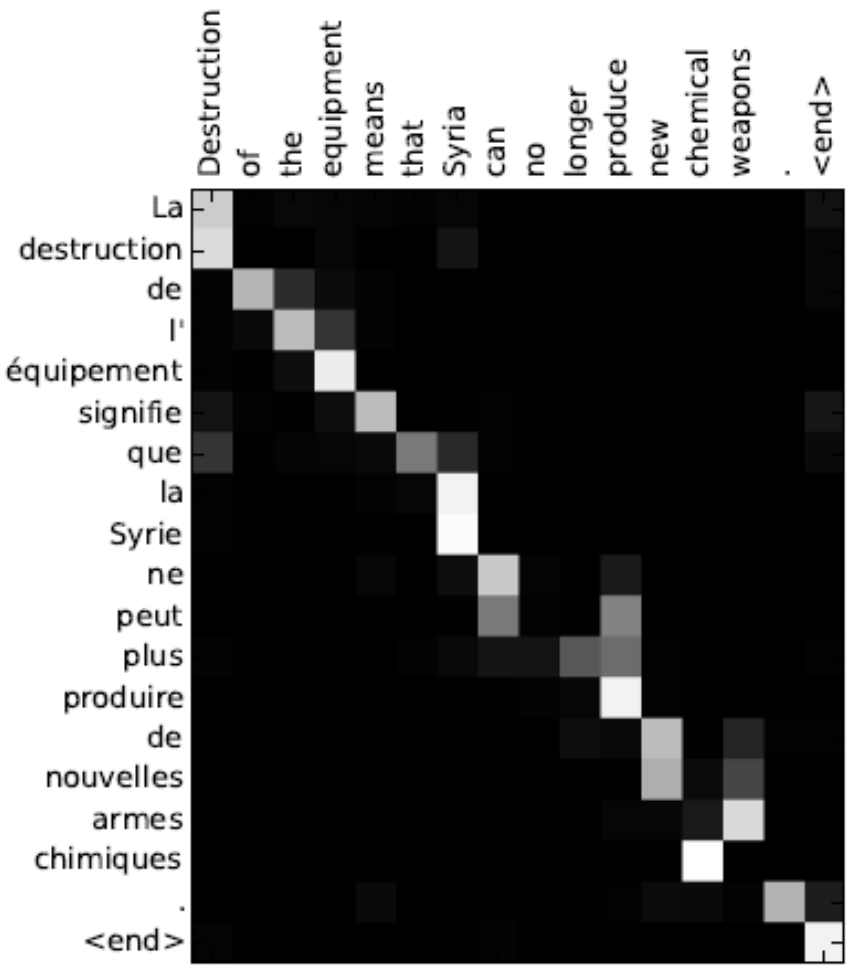
$p(y_t)$ ↑ s_t decoder



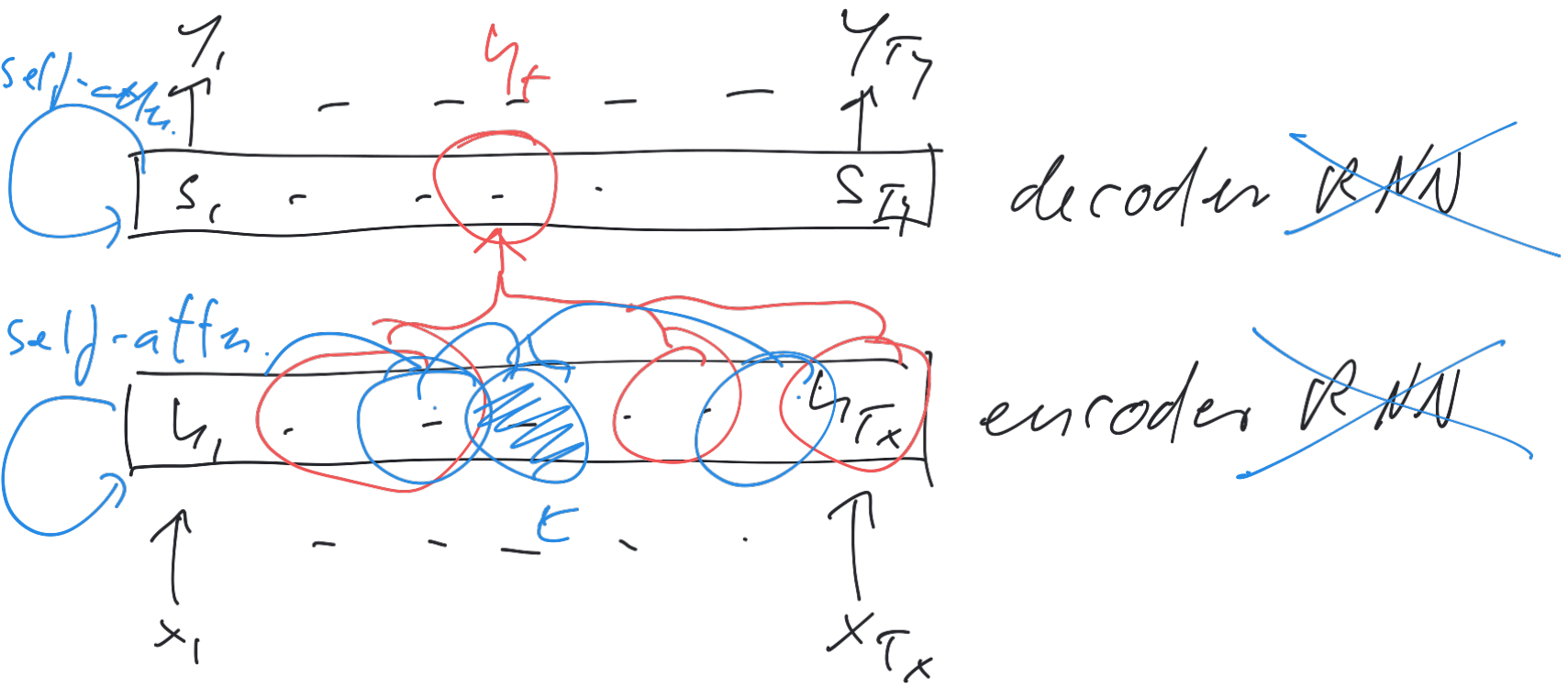


$$\begin{bmatrix} h_t^{for} \\ h_t^{backw.} \end{bmatrix}$$





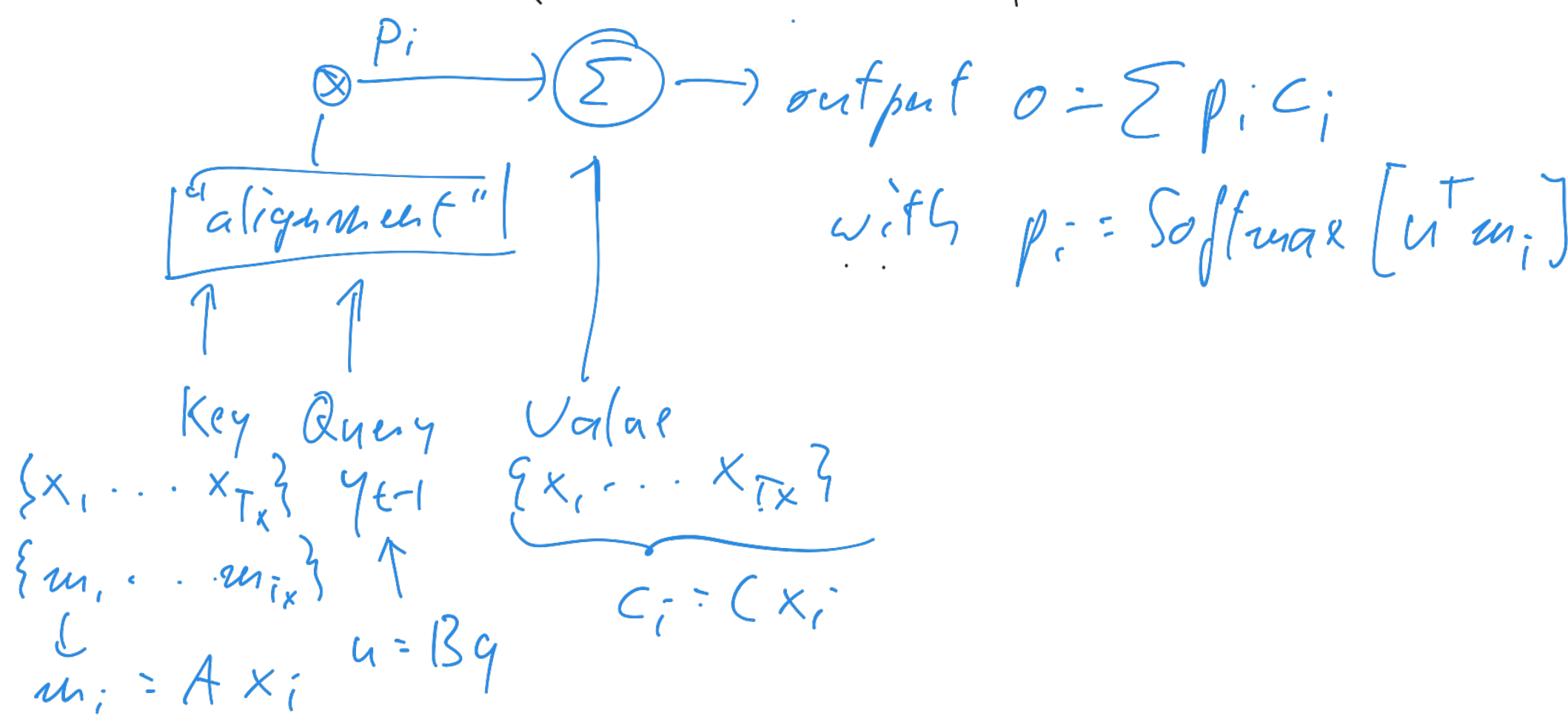
Bahdanau ... Bengio (2015) ICLR

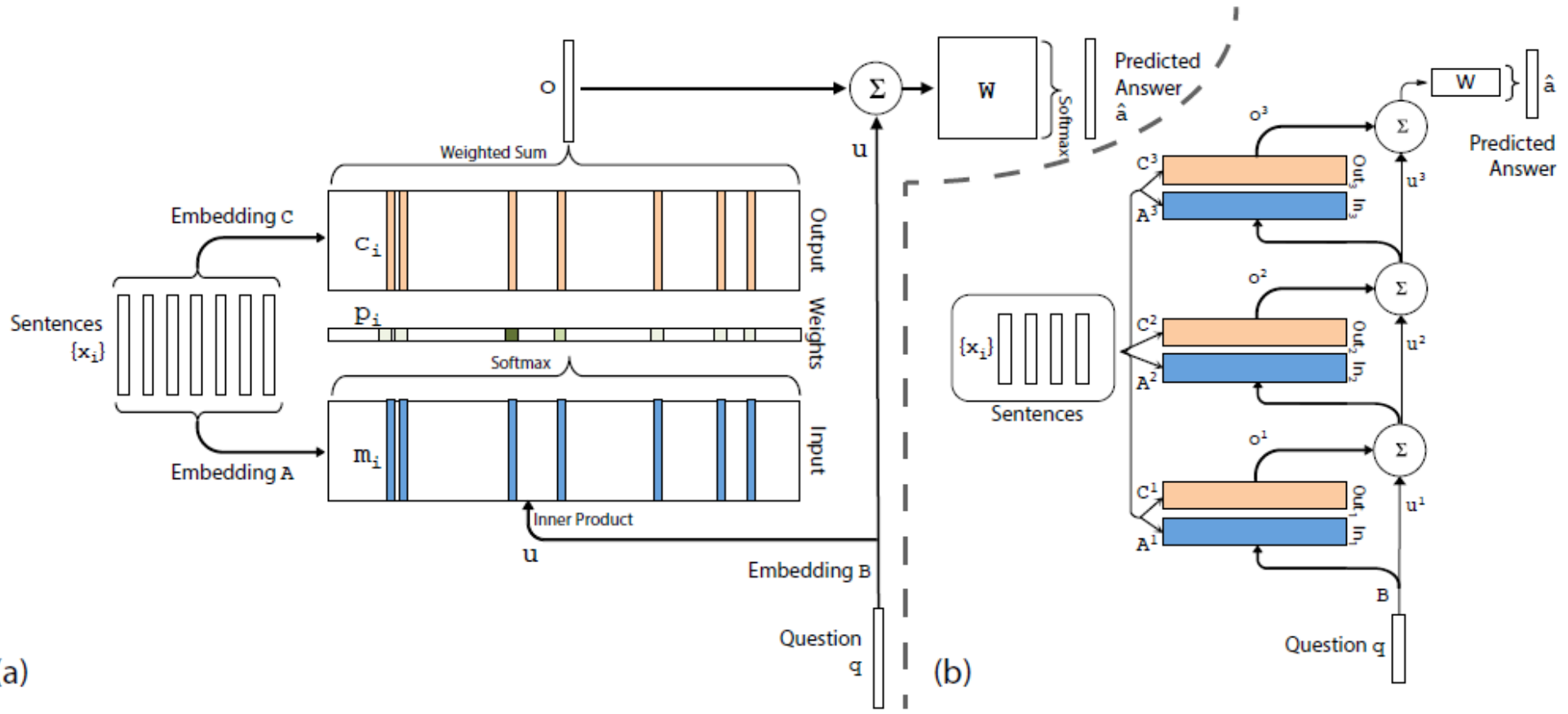


- strongly parallelizable

→ Transformer, Vaswani et al. (2017) NIPS

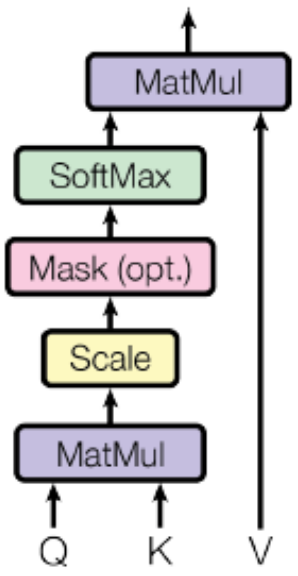
Attn. mech. (Sukhbaatar et al. 2015,





Sukhbataar ... Fergus (2015) NIPS

Scaled Dot-Product Attention



Vaswani et al. (2017) NIPS

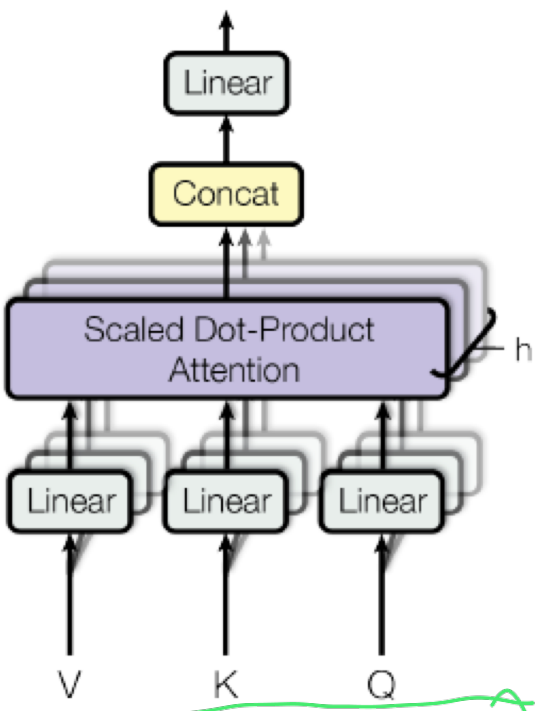
$$Q = \begin{bmatrix} q_1 \\ q_i \\ q_{T_y} \end{bmatrix}, \quad K = \begin{bmatrix} k_1 \\ \vdots \\ k_{T_x} \end{bmatrix}, \quad V = \begin{bmatrix} v_1 \\ \vdots \\ v_{T_x} \end{bmatrix}$$

$$Attn(Q, K, V) = softmax\left[\frac{Q K^T}{\sqrt{d_k}}\right] V$$

$x_1 \ldots x_T$

$y_1 \ldots \quad y_t$ XXXX y_{T_y}

Multi-Head Attention



$$MH(Q, K, V) = \text{attn. head}$$

$$Concat\left[H_1 \ldots H_n\right] W_0$$

with

$$H_i = Attn\left(Q W_i^Q, K W_i^K, V W_i^V\right)$$

$y_1 \ldots \quad y_t$ XXXX

Self-Attn.

Decoder

Cross-Attn.

Encoder

Self-Attn

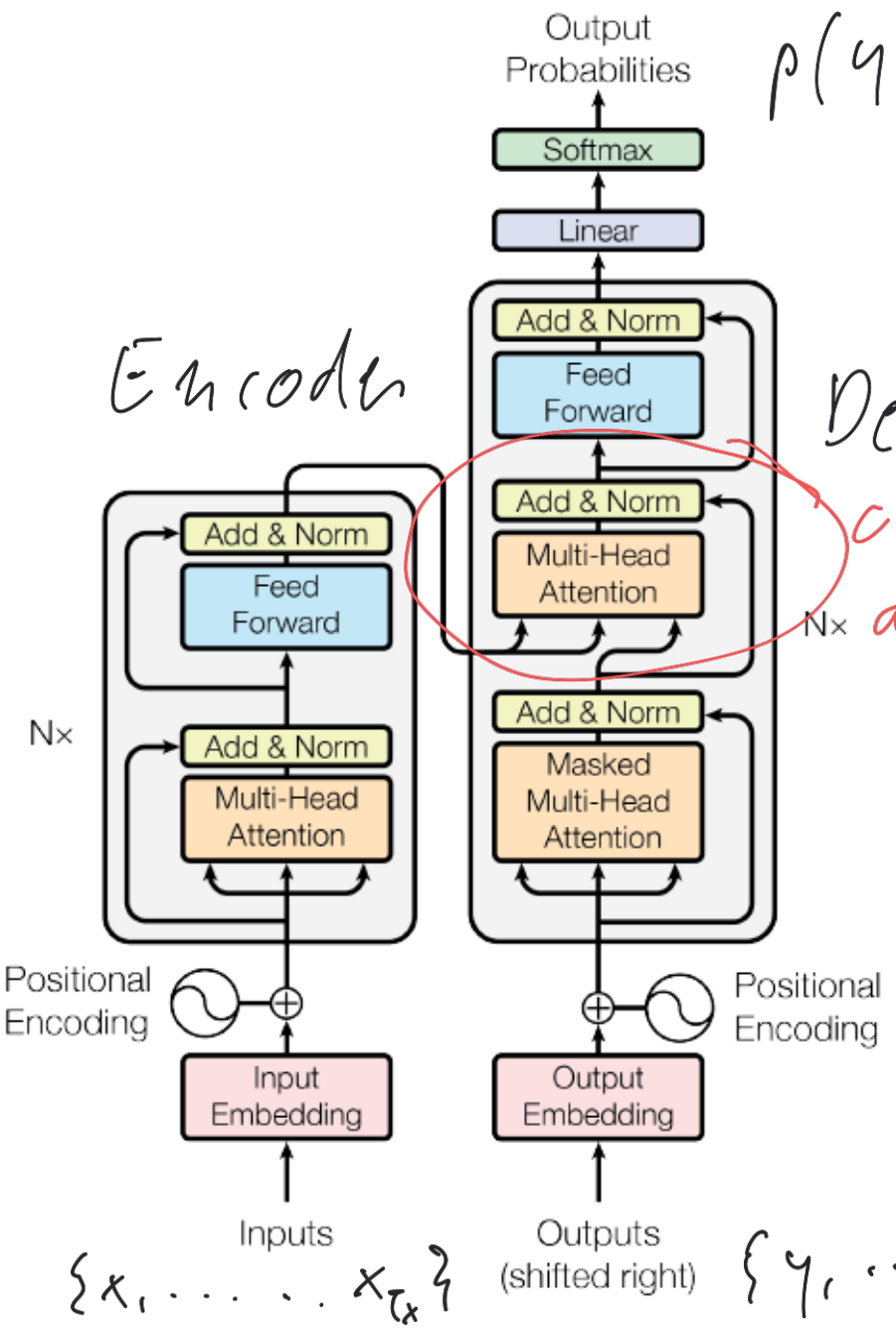
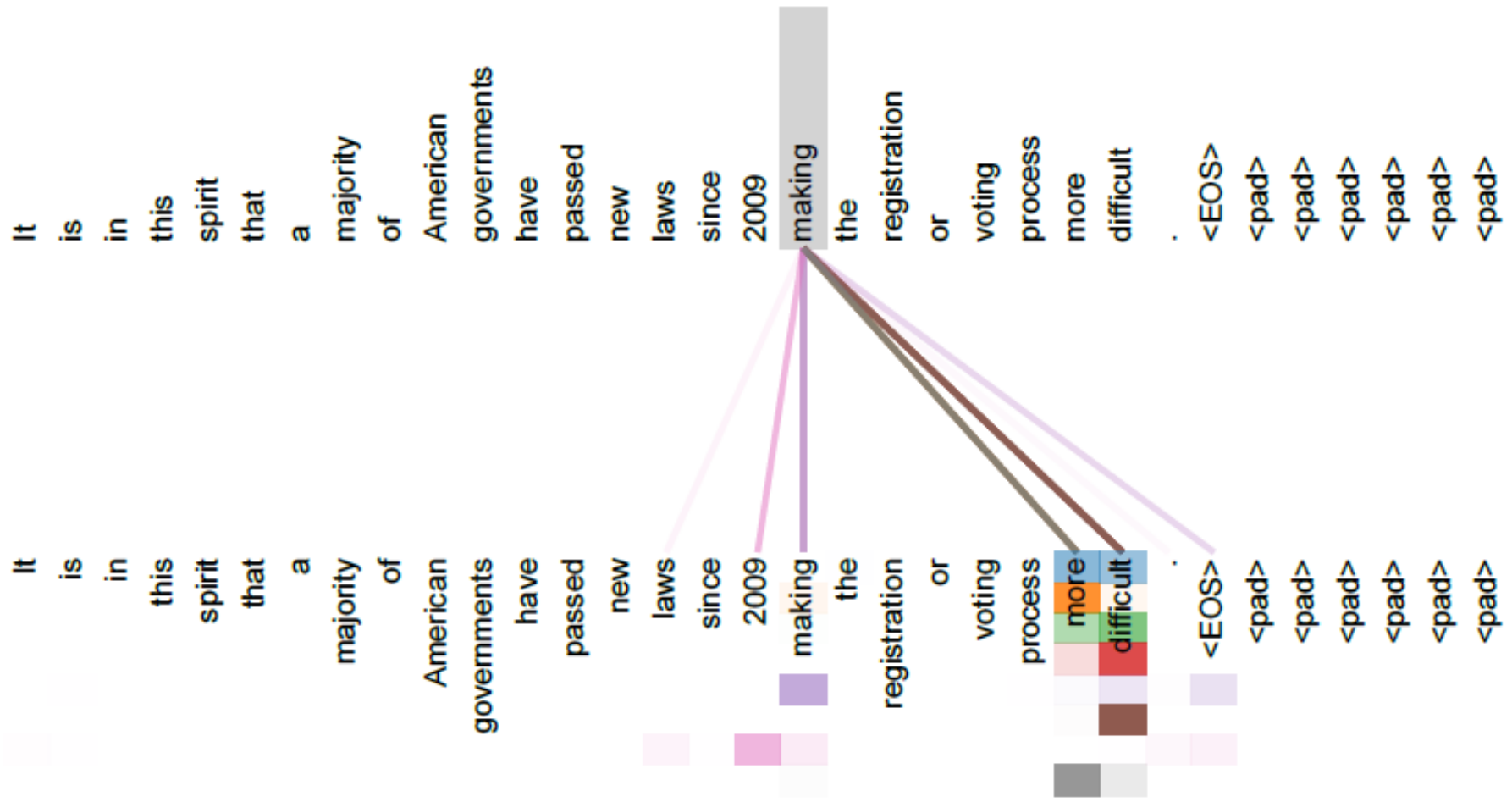


Figure 1: The Transformer - model architecture.

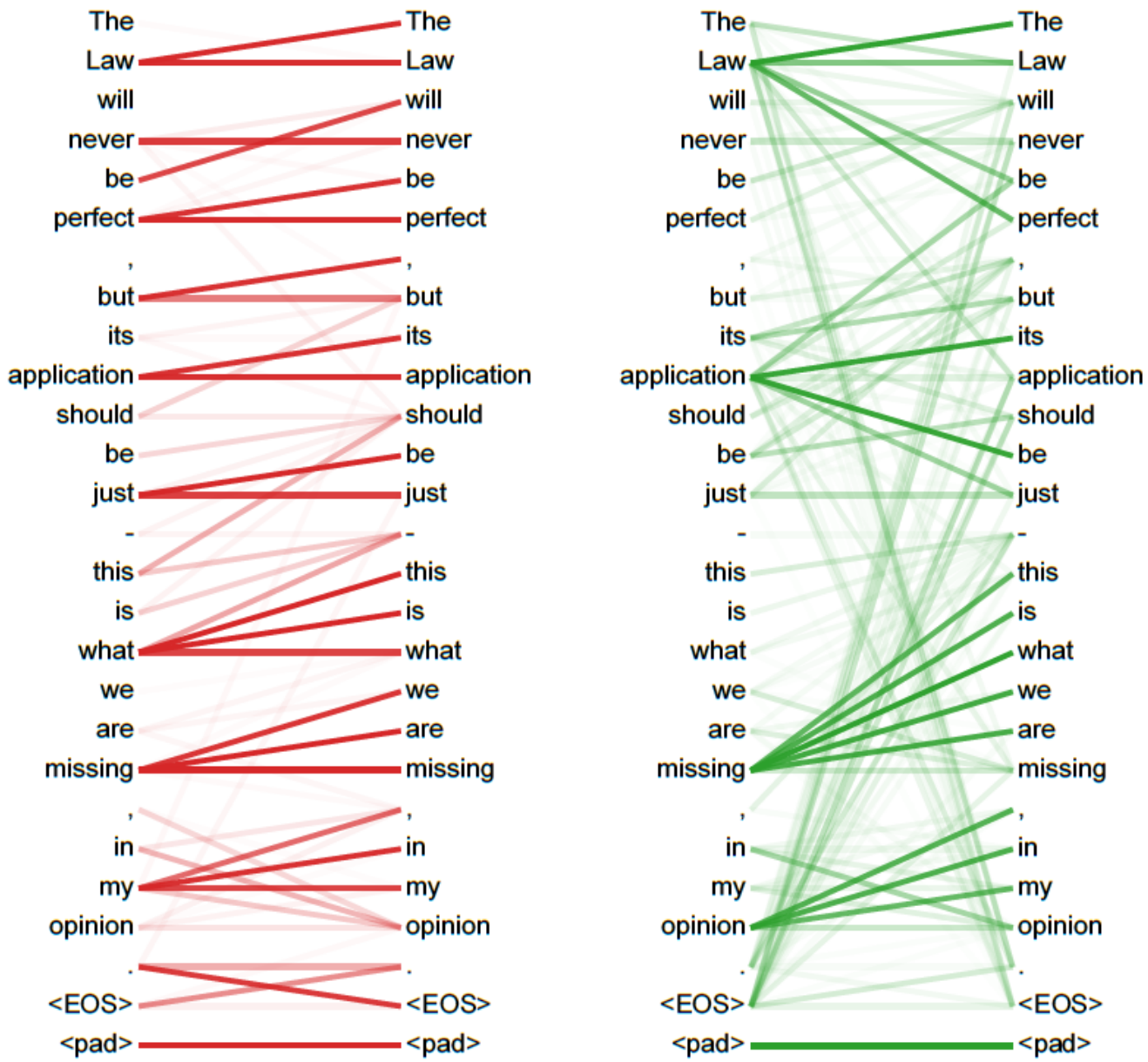
Vaswani et al. (2017) NIPS

Attention Visualizations



Vaswani et al. (2017) NIPS

Vaswani et al. (2017) NIPS



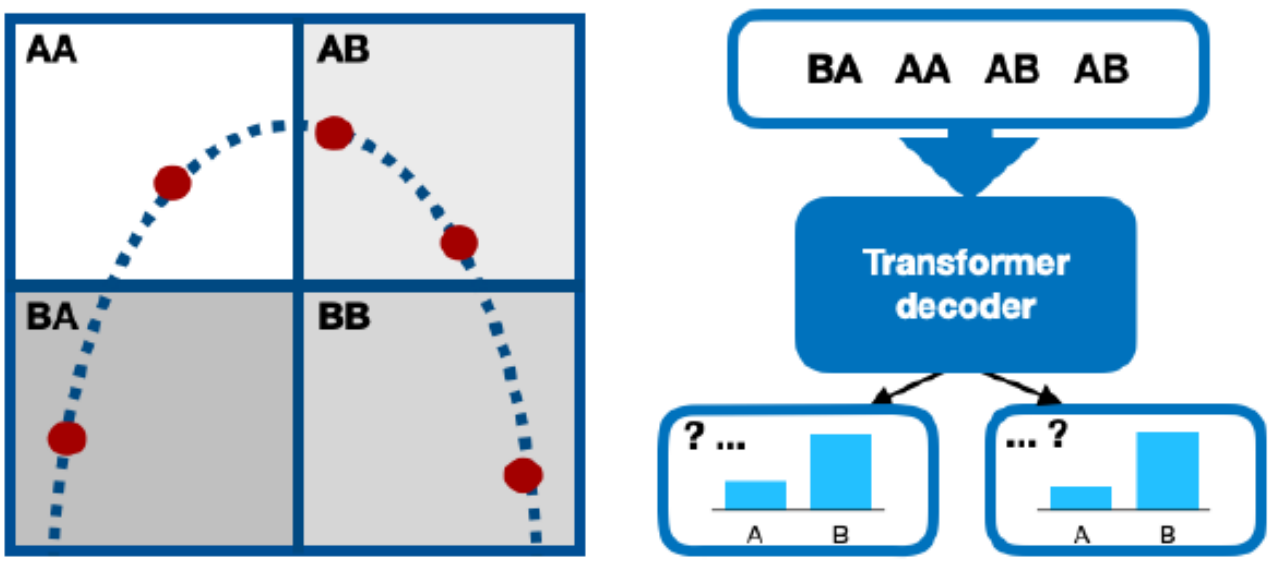


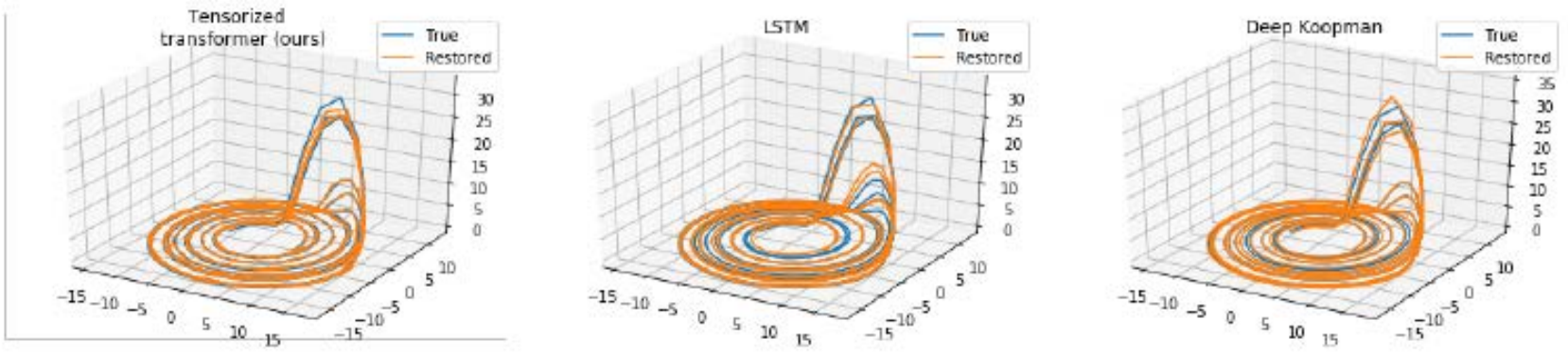
Figure 1: An example of the proposed method for the 2-dimensional trajectory. The initial states are discretized along each dimension; then transformer decoder is used to predict conditional probabilities along each dimension.

Shalova & Oseledets (2020) arXiv

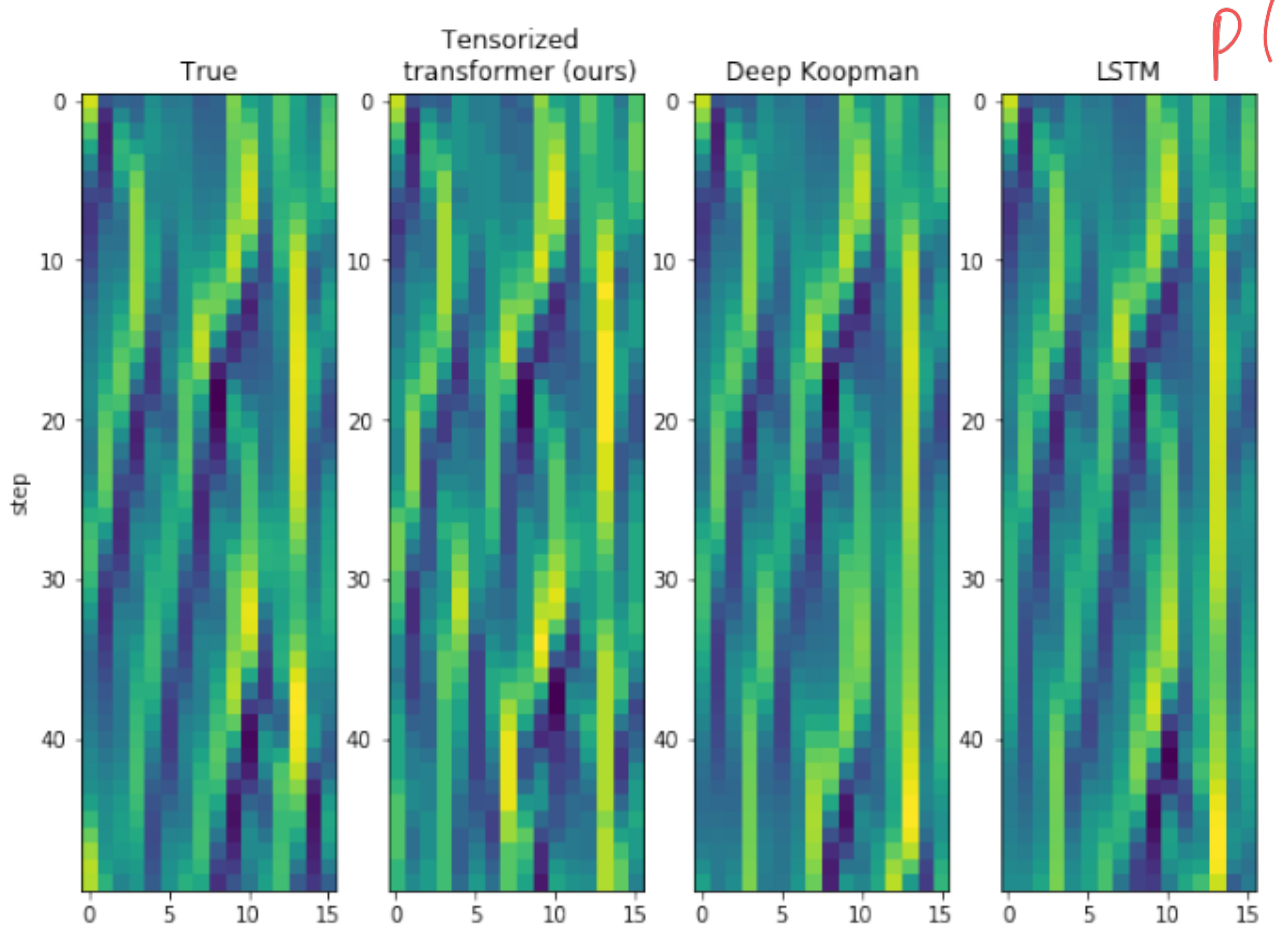
$$p(x_{k+1} \mid x_1 \dots x_k)$$

$$\mathcal{L} = \text{cross-entropy} = -\frac{1}{N} \sum_t \log p(x_t \mid x_1 \dots x_{t-1})$$

Softmax [Transform] evaluated at x_t



$p(x_t | x_1 \dots x_{t-1})$



Shalova & Oseledets (2020) arXiv

- basic def. cov, corr } (X)
- ARMA
- Poisson-AR
- latent var., lat. DS (Z) → (X)
- EM, KFS $p(z|x)$
- DS
- RNN, long-term dep.
- (RNN) → (X), EM: EKF, PF, Laplace approx.
- seq-VAE / UI
- Bayesian RNN, GAN
- Attn., Transformers