Neural Computation

Monday Session 27 November

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

- Sequence-to-Sequence Architecture

INPUT | Je suis étudiant | OUTPUT | I am a student |

- Hugely influential
- Basis for ChatGPT (**G**enerative **P**re-trained **T**ransformer)
- Also for vision

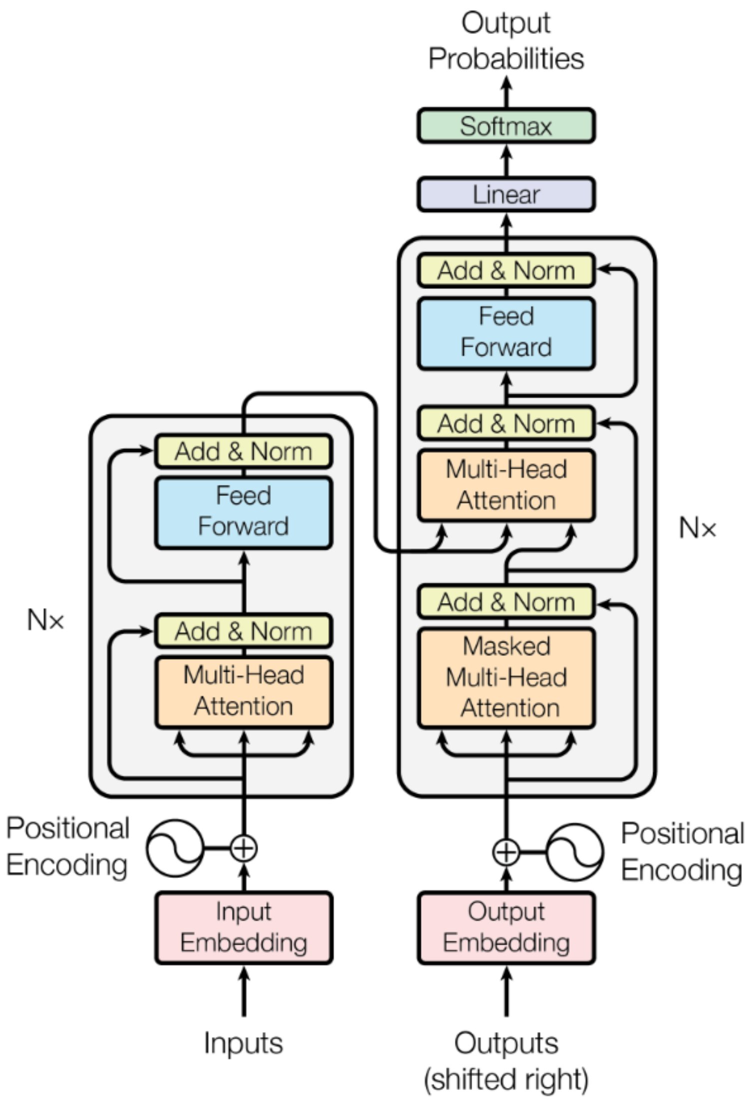
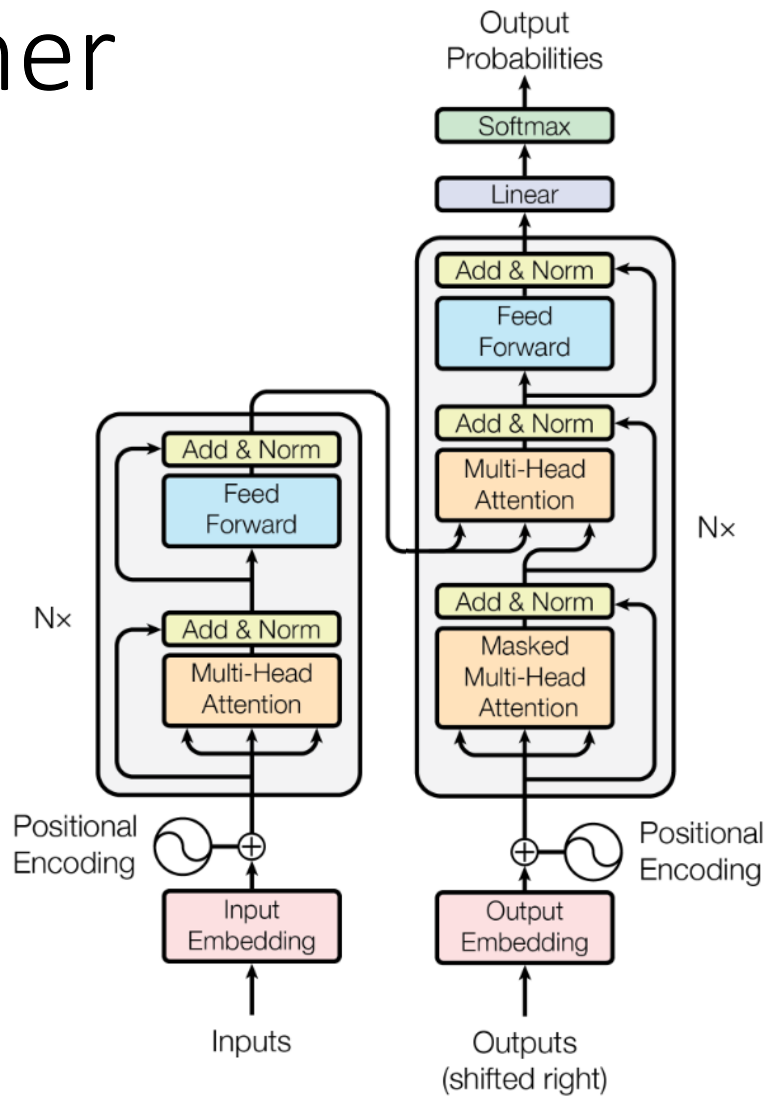
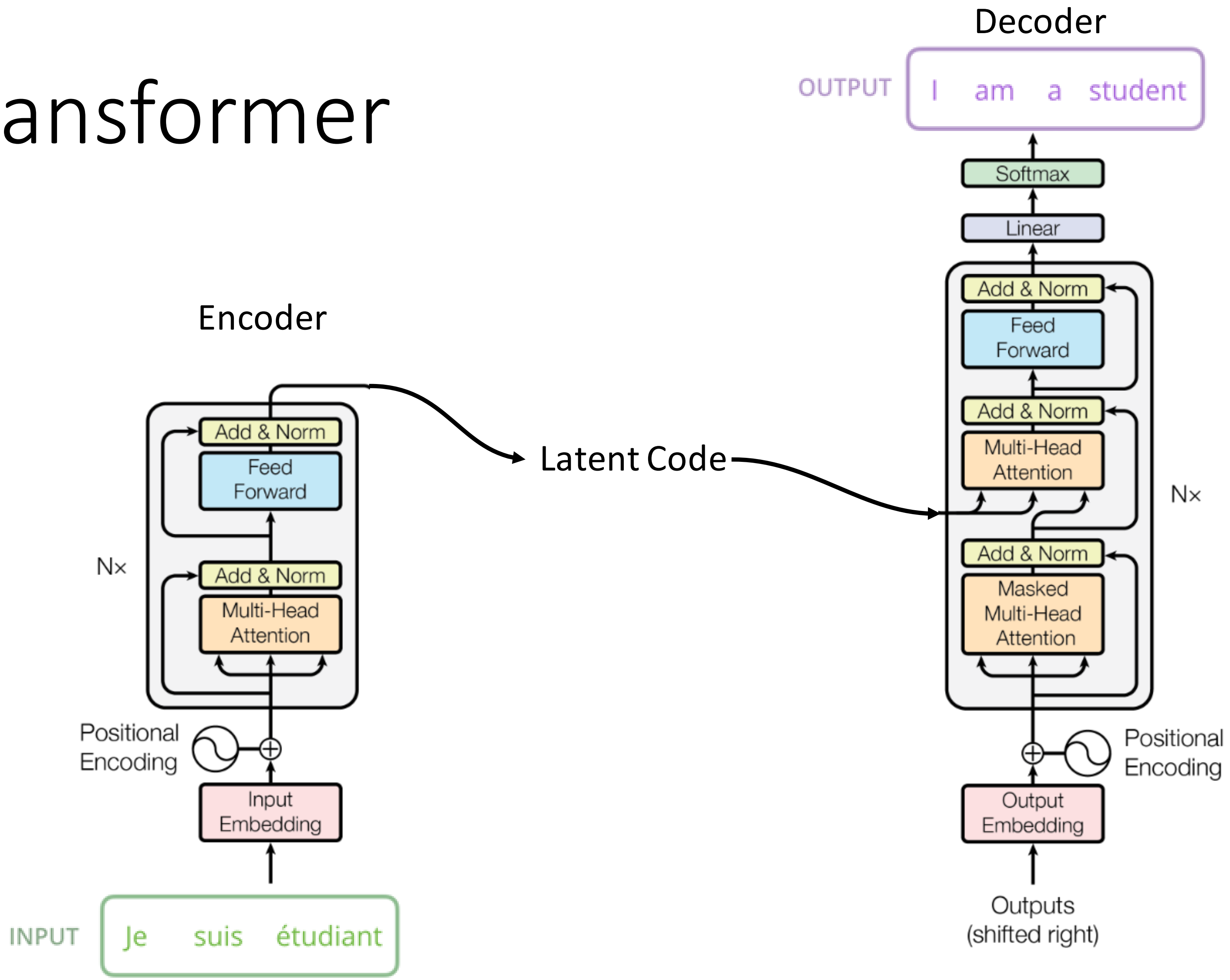


Figure 1: The Transformer - model architecture.

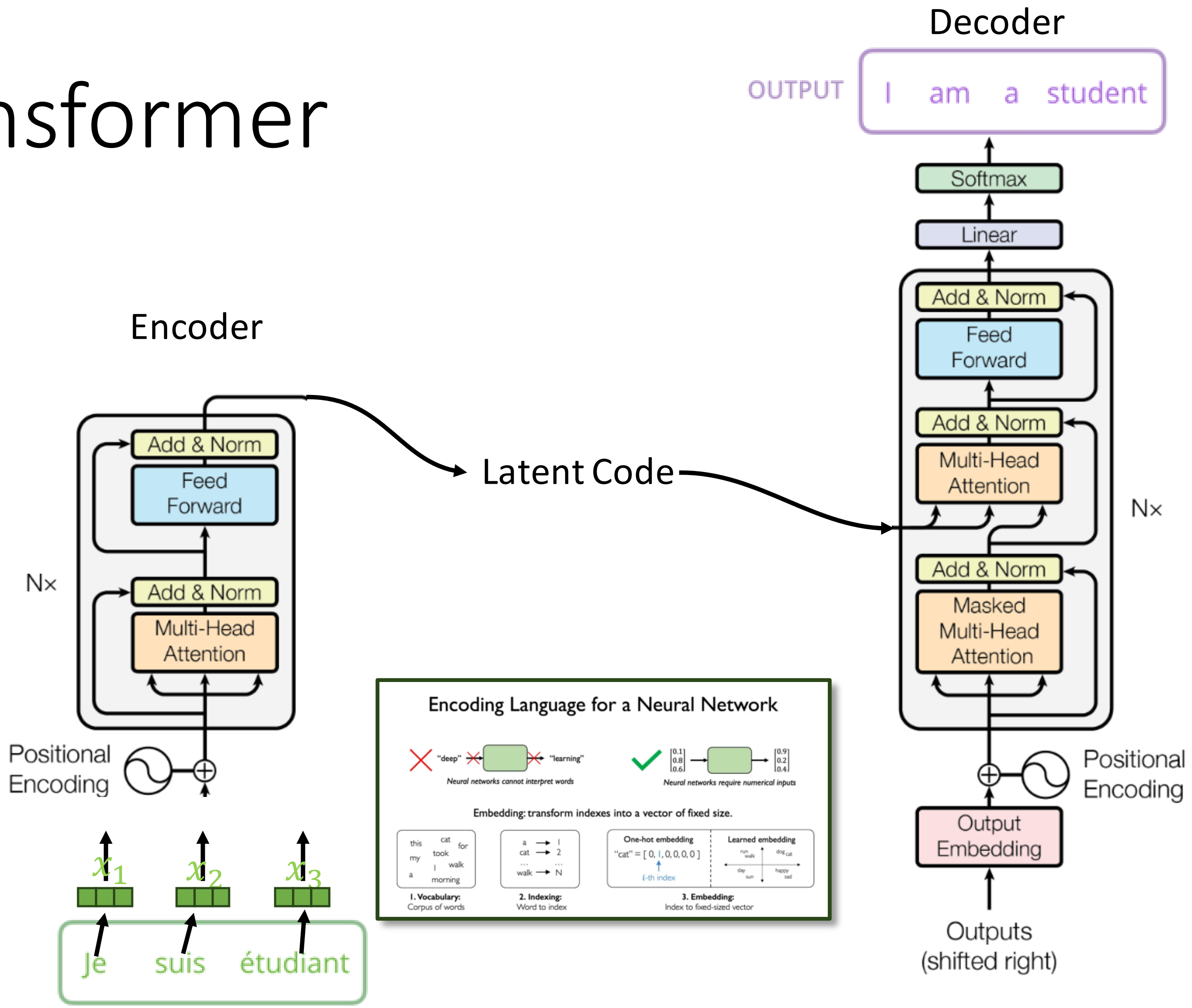
The Transformer



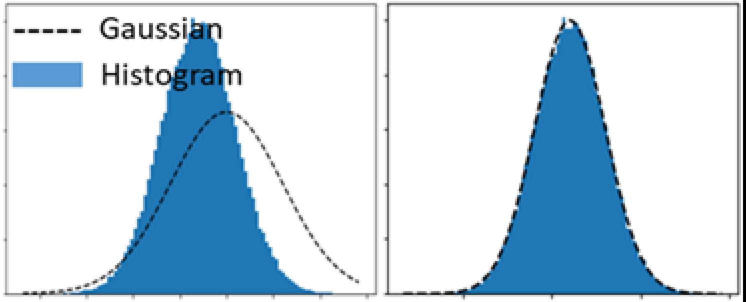
The Transformer



The Transformer

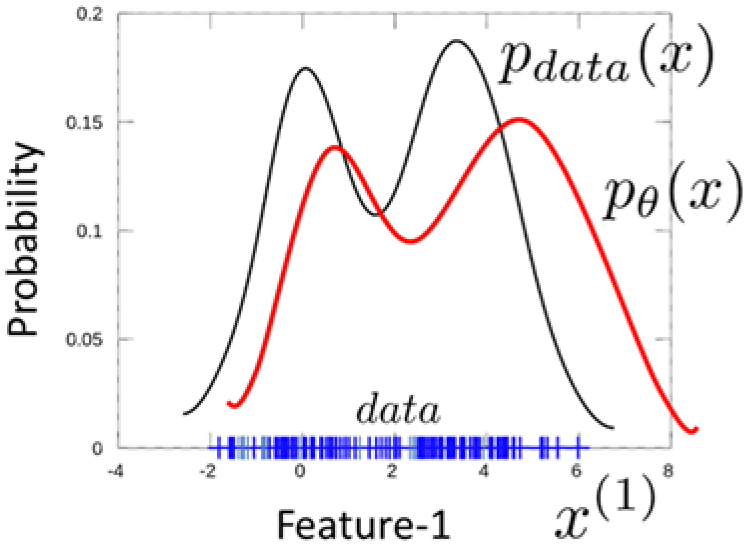


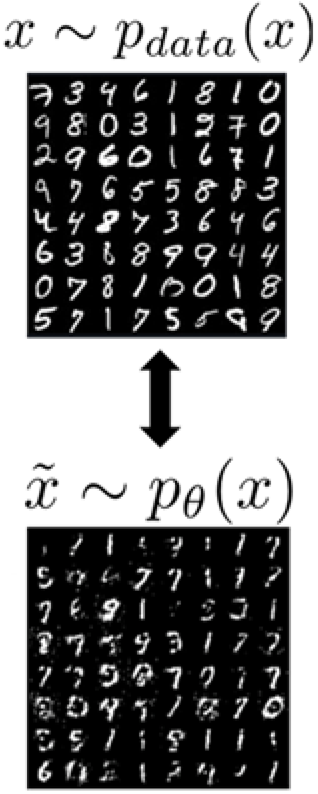
Probability Density Estimation



One of the main aims of unsupervised approaches and Generative Modelling.

Goal of Density Estimation:
We could try to fit a probabilistic model $p_\theta(x)$ to the data,
to learn their underlying distribution $p_{data}(x)$.
How? By learning its parameters θ so that: $p_\theta(x) \approx p_{data}(x)$

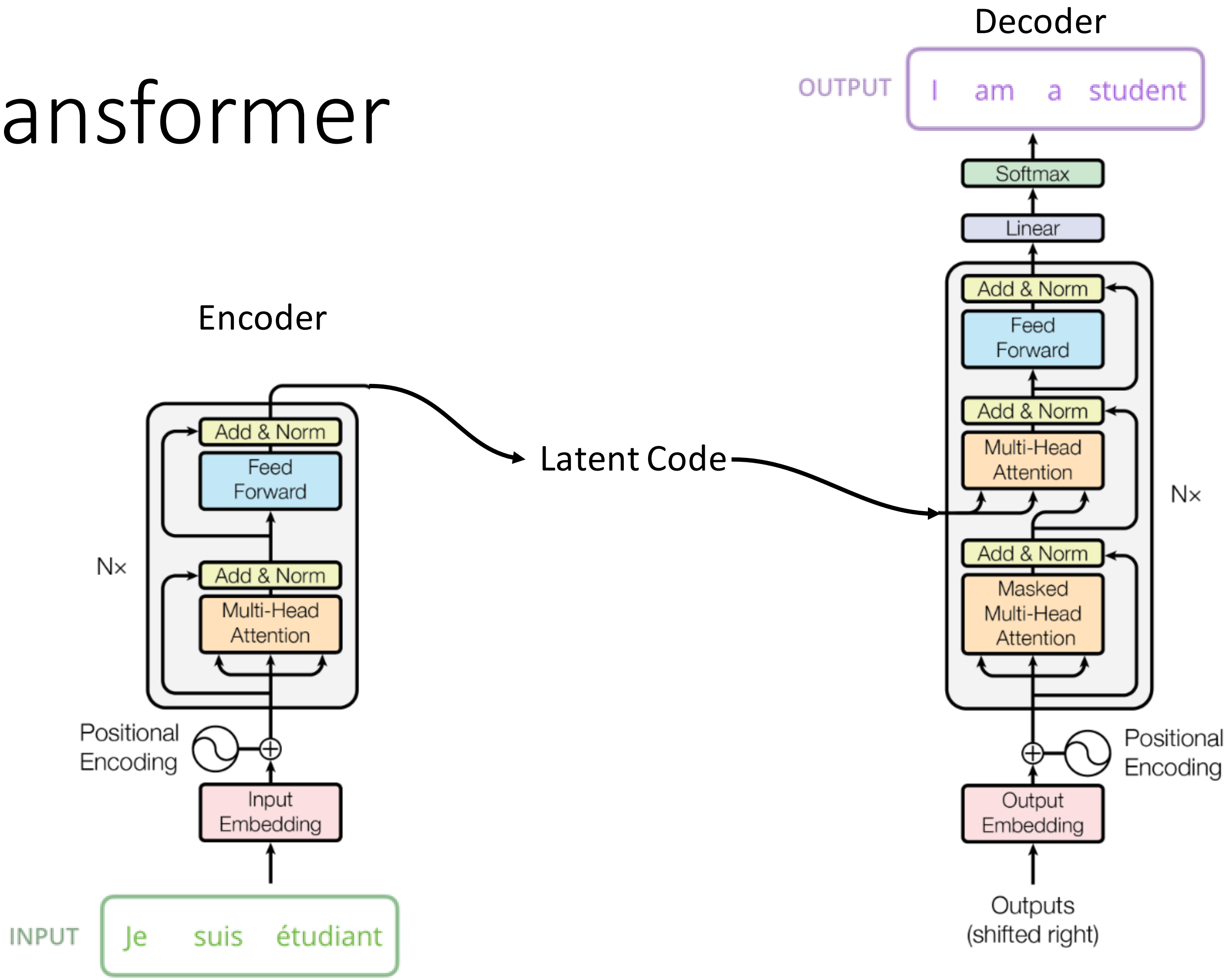


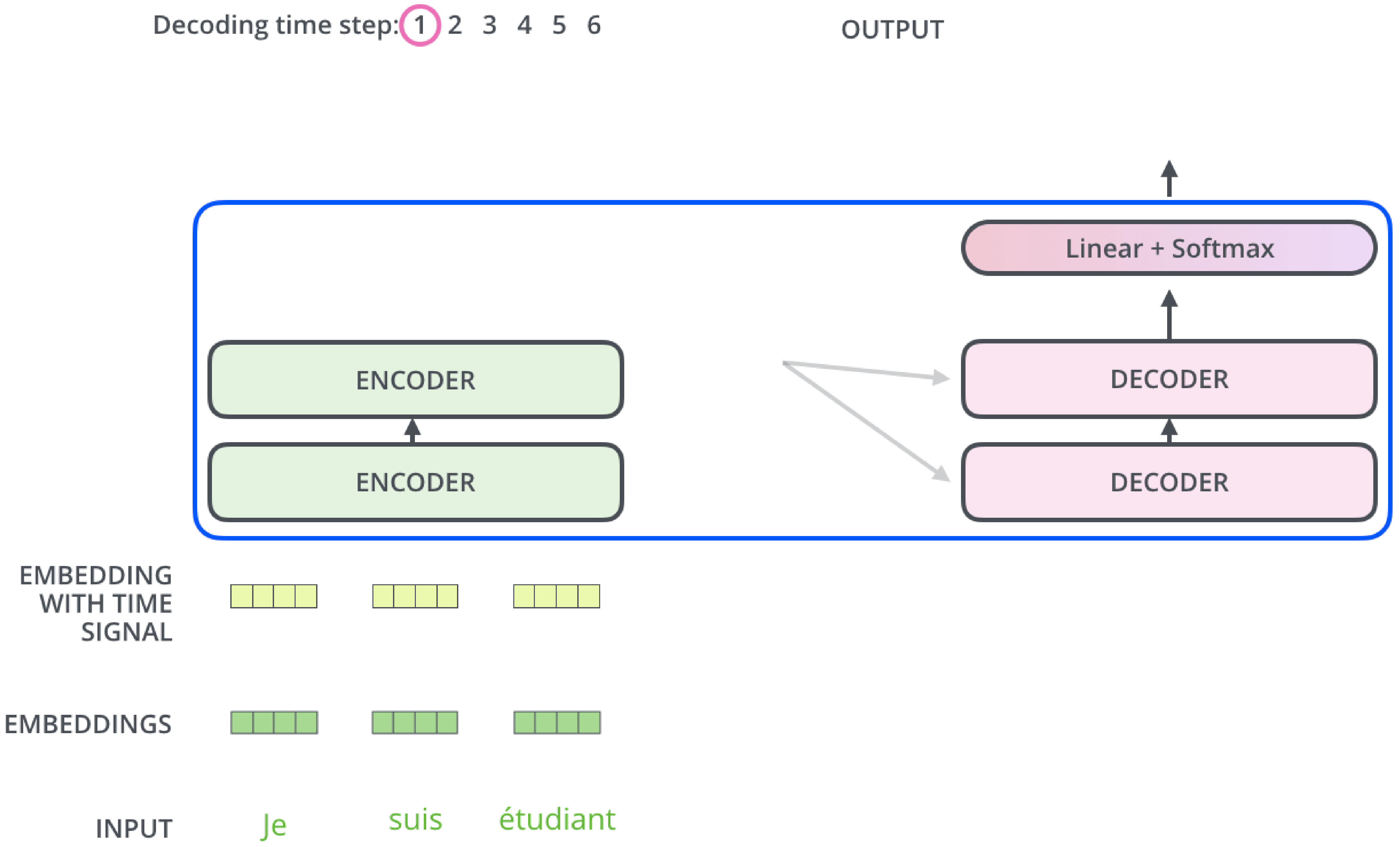


But we cannot always do that directly.
Perhaps we cannot compute $p_{data}(x)$ or $p_\theta(x)$.
Instead, we could do PDE *indirectly:*
Enforce samples from model to be similar to real data instead:

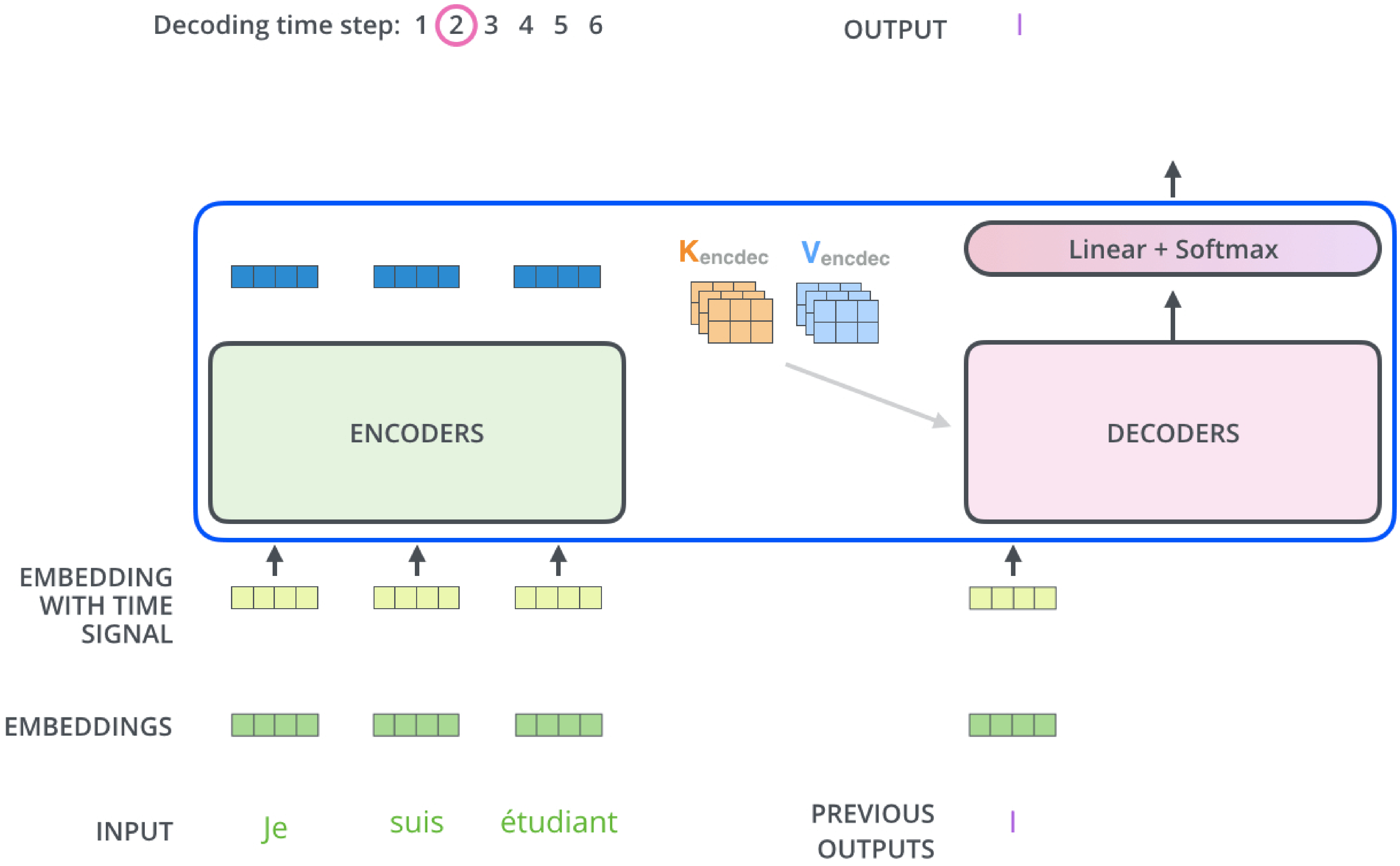
Both VAEs and GANs can be seen as following this approach.

The Transformer





Jay Alammar: https://jalammar.github.io/illustrated-transformer/



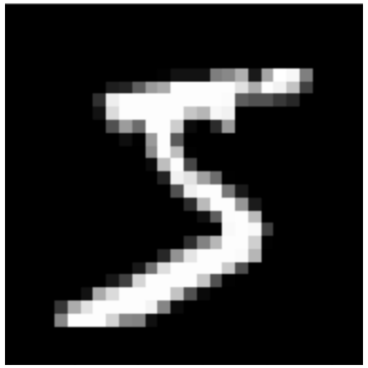
Jay Alammar: https://jalammar.github.io/illustrated-transformer/

Autoregressive Models

Problem:

- Model high dimensional difficult distribution

$$p_\theta(\boldsymbol{x}) = p_{\text{data}}(\boldsymbol{x}), \text{ with } \boldsymbol{x} = (x_1, \ldots, x_n)$$



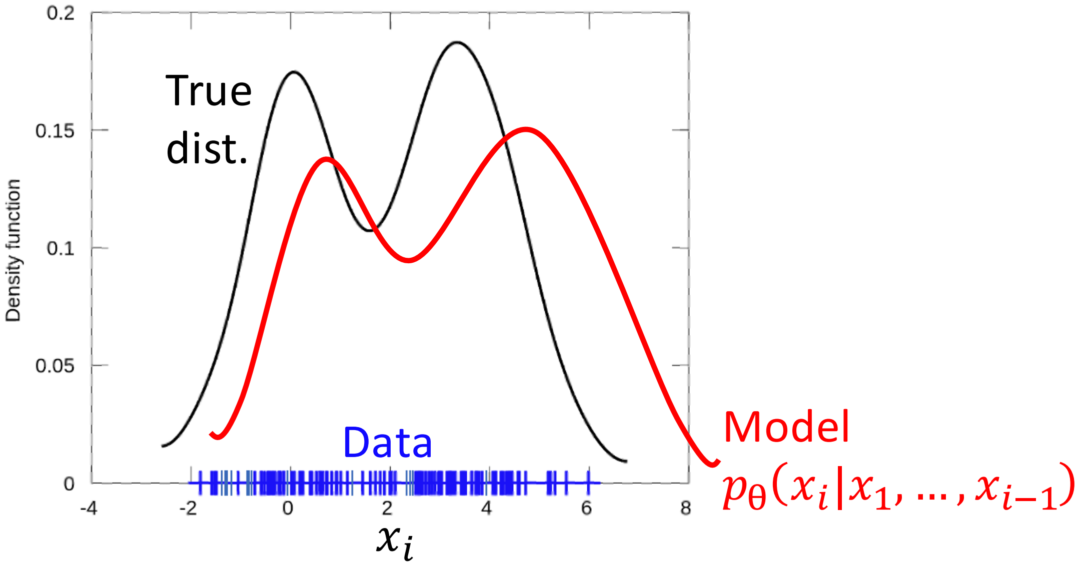
Idea:

- Factorise distribution

$$p_\theta(\boldsymbol{x}) = \prod_{i=1}^{n} p_\theta(x_i | x_1, \ldots, x_{i-1})$$

Neural network:

- Parameters θ
- Input $x_1, \ldots, x_{i-1}$
- Output dist. over x_i



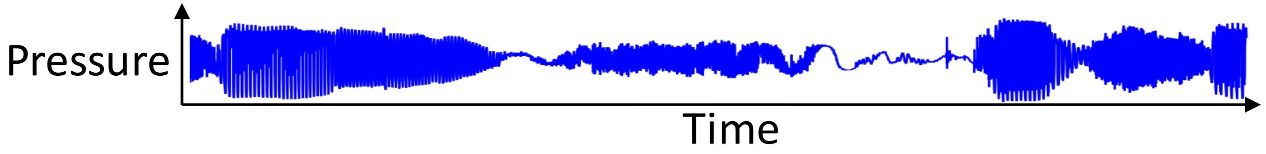
WaveNet: A Generative Model for Raw Audio

Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

Karen Simonyan **Oriol Vinyals** **Alex Graves**

Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK



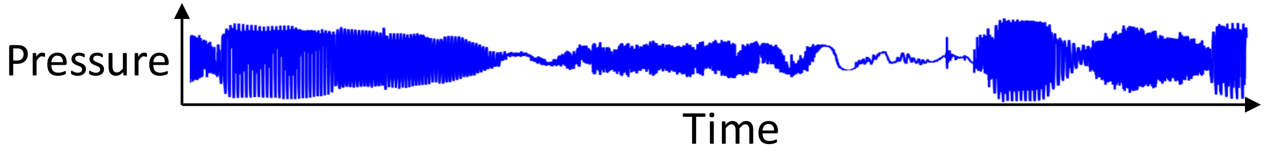
WaveNet: A Generative Model for Raw Audio

Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

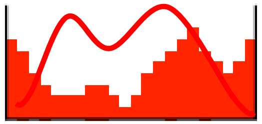
Karen Simonyan **Oriol Vinyals** **Alex Graves**

Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

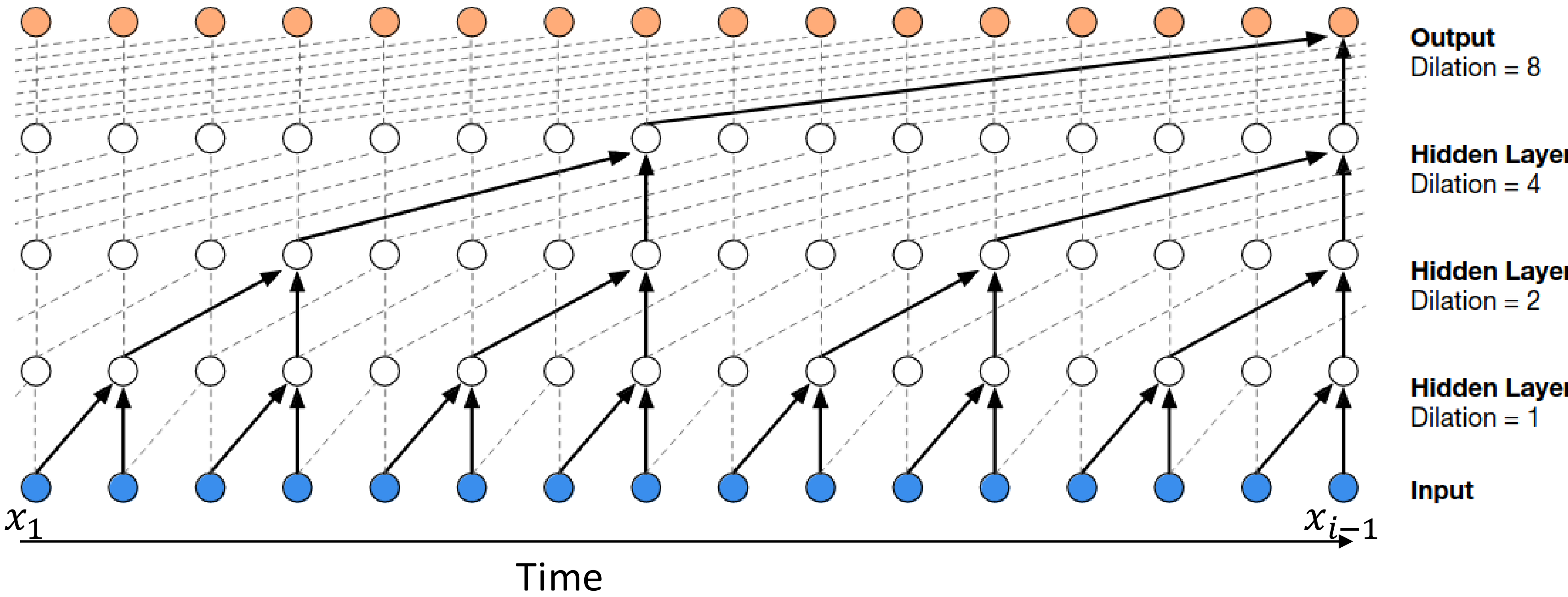
{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK



- Predict dist. for next audio sample



$p_\theta(x_i | x_1, \ldots, x_{i-1})$



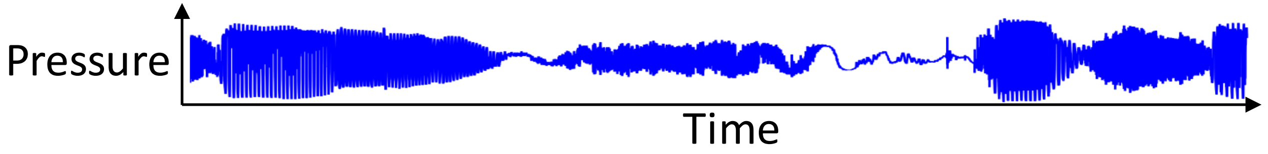
WaveNet: A Generative Model for Raw Audio

Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

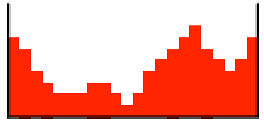
Karen Simonyan **Oriol Vinyals** **Alex Graves**

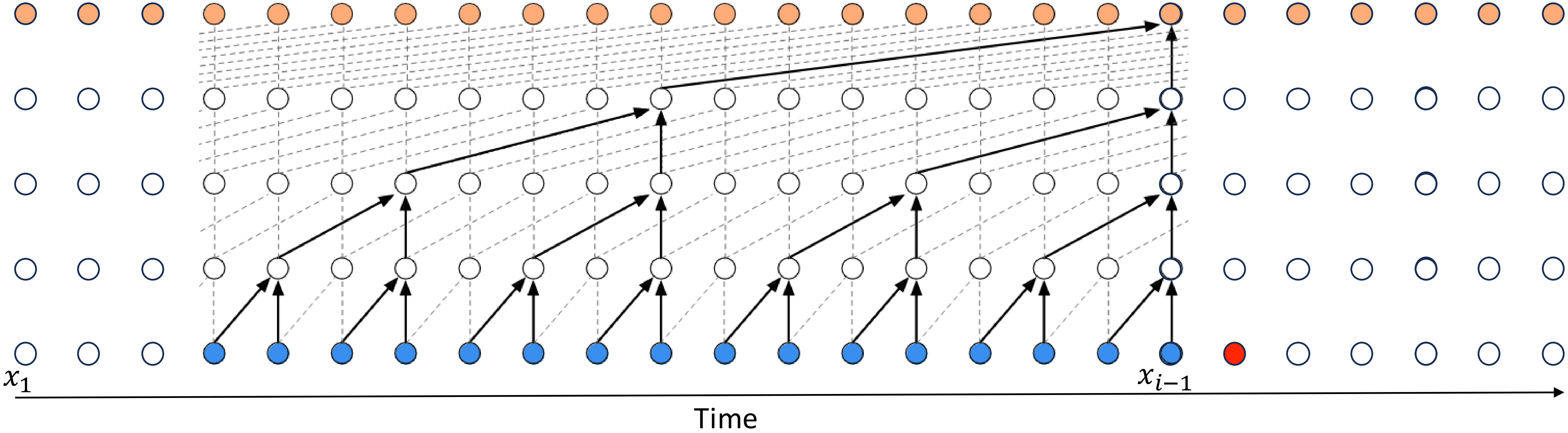
Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK



- Predict dist. for next audio sample
- Fully conv architecture:
  - Simultaneous pred. for all timepoints





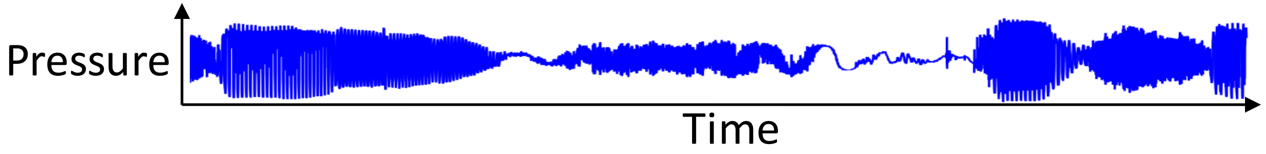
WaveNet: A Generative Model for Raw Audio

Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

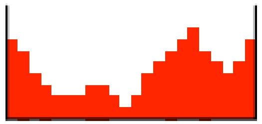
Karen Simonyan **Oriol Vinyals** **Alex Graves**

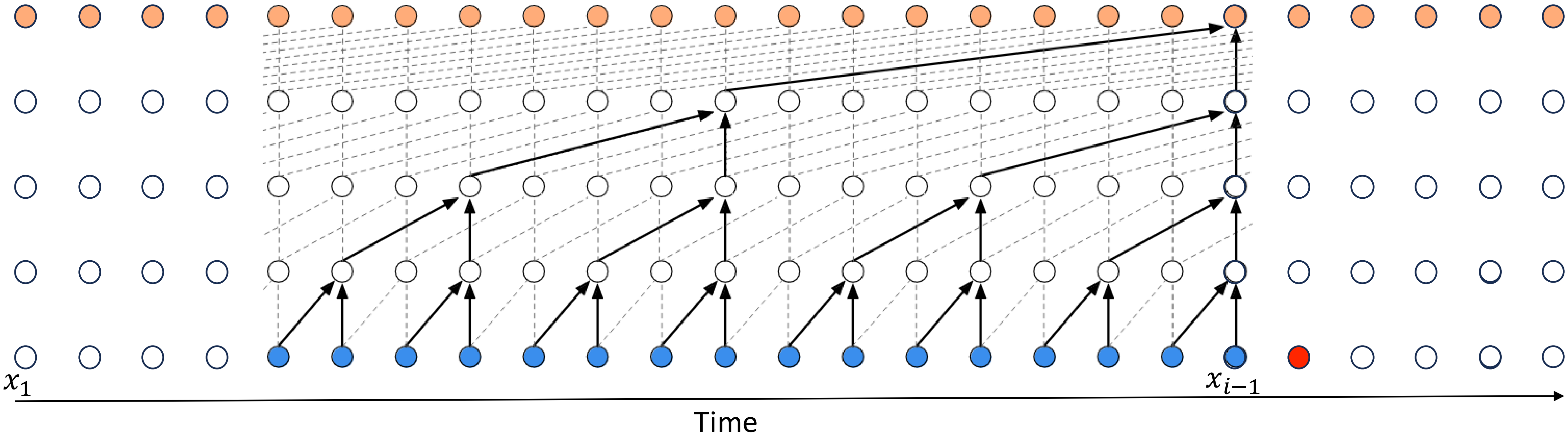
Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK



- Predict dist. for next audio sample
- Fully conv architecture:
  - Simultaneous pred. for all timepoints





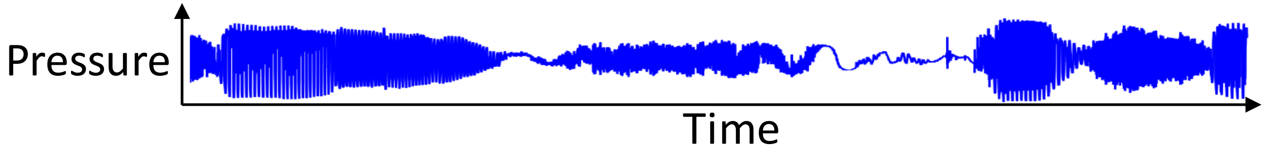
WaveNet: A Generative Model for Raw Audio

Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

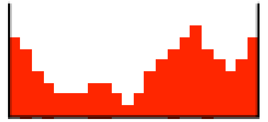
Karen Simonyan **Oriol Vinyals** **Alex Graves**

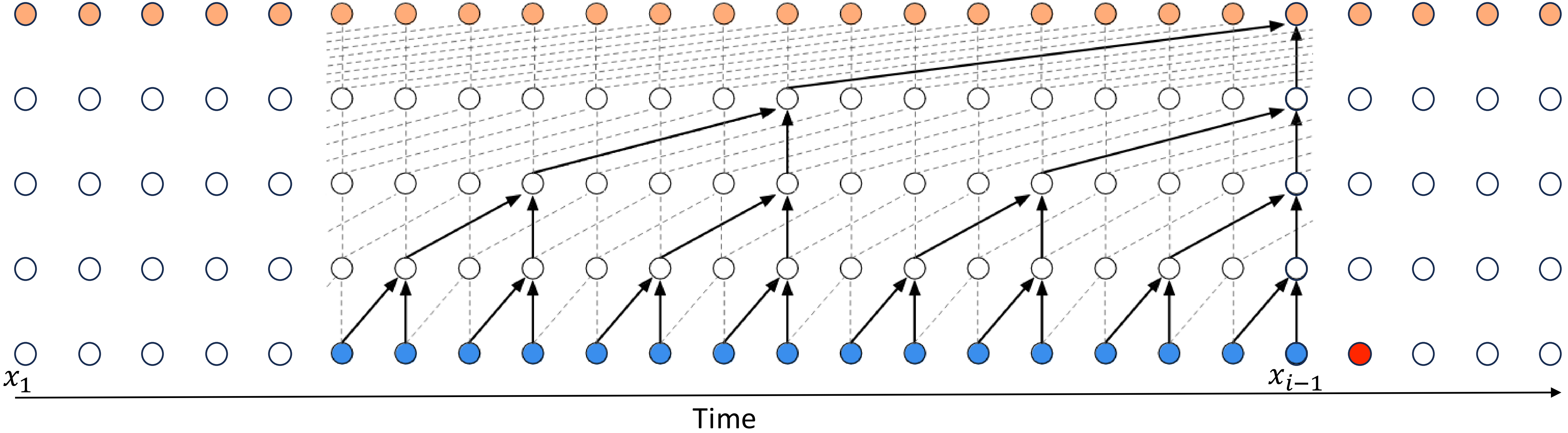
Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK



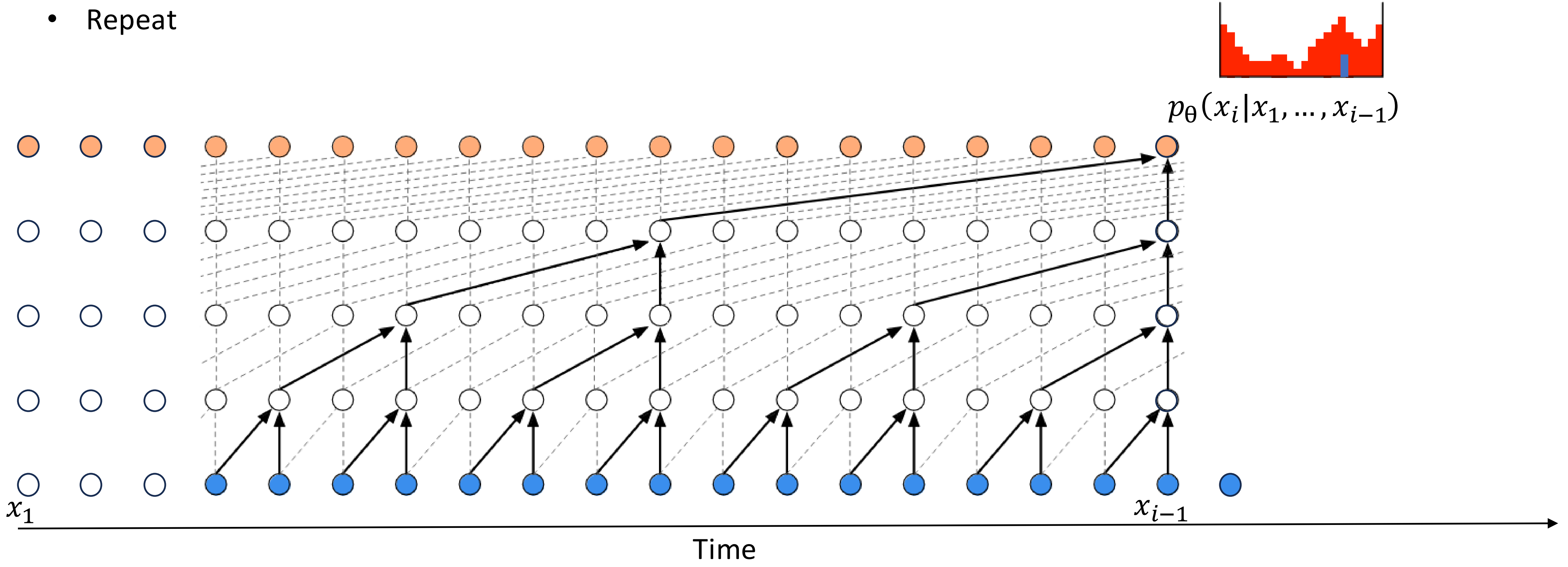
- Predict dist. for next audio sample
- Fully conv architecture:
  - Simultaneous pred. for all timepoints





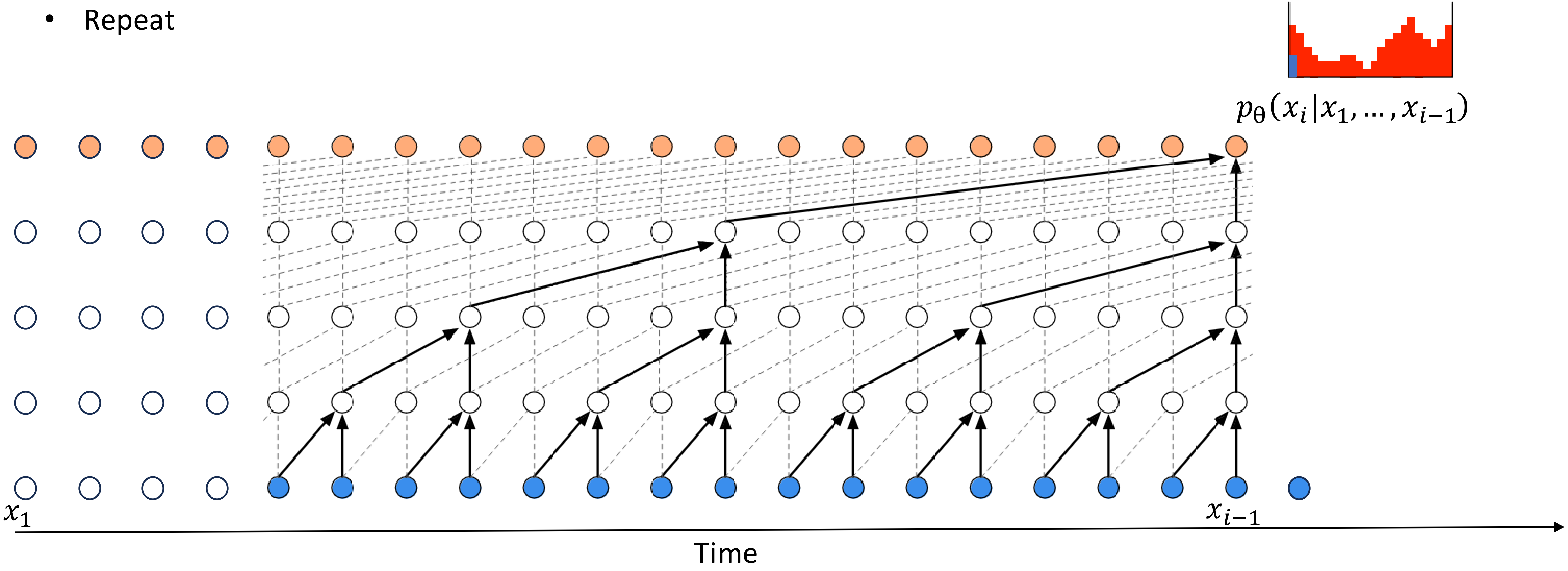
Sampling from the Model

- Predict dist. for next audio sample
- Sample from distribution
- Append new sample
- Repeat



Sampling from the Model

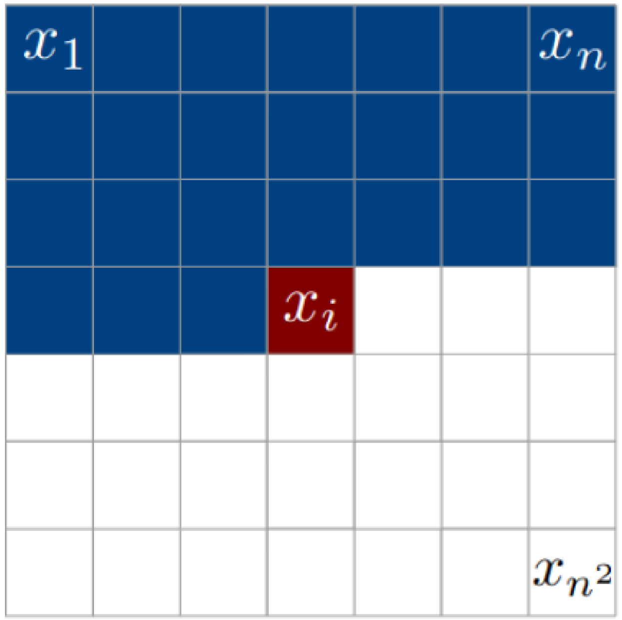
- Predict dist. for next audio sample
- Sample from distribution
- Append new sample
- Repeat

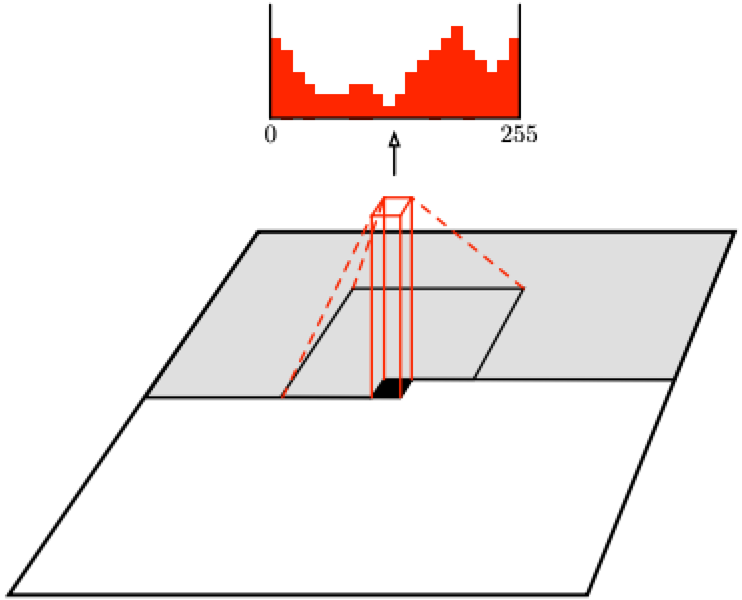


Pixel Recurrent Neural Networks

Aäron van den Oord AVDNOORD@GOOGLE.COM
Nal Kalchbrenner NALK@GOOGLE.COM
Koray Kavukcuoglu KORAYK@GOOGLE.COM

Google DeepMind



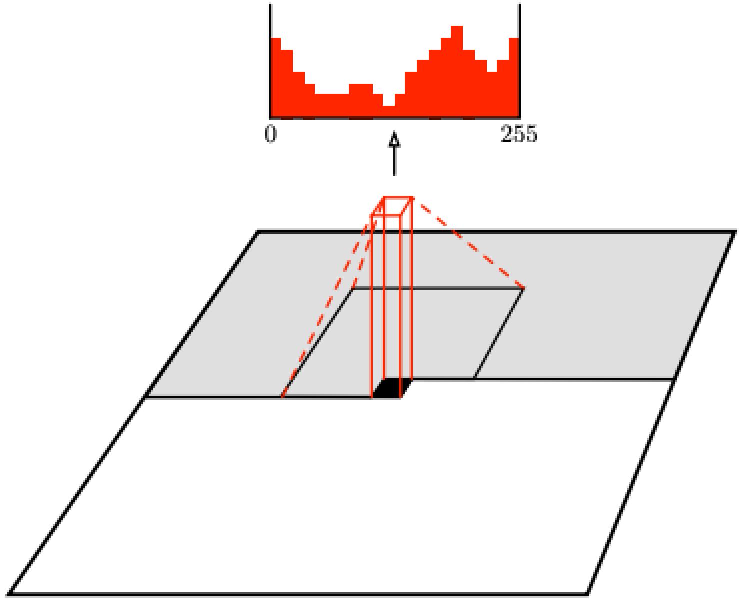


Pixel Recurrent Neural Networks

Aäron van den Oord AVDNOORD@GOOGLE.COM
Nal Kalchbrenner NALK@GOOGLE.COM
Koray Kavukcuoglu KORAYK@GOOGLE.COM

Google DeepMind

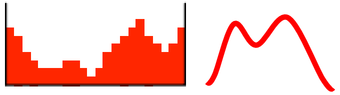
Image Generation:
- Sample one pixel
- Apply network
- Repeat



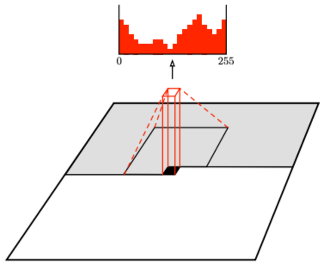
Summary

$$p_\theta(\boldsymbol{x}) = \prod_{i=1}^{n} p_\theta(x_i | x_1, \dots, x_{i-1})$$

- Interpret data as sequence
- Train neural network
  - Input: previous values $(x_1, \dots, x_{i-1})$
  - Distribution of possible next values $p_\theta(x_i | x_1, \dots, x_{i-1})$
    - E.g. as histogram
    - Or Parametric dist.

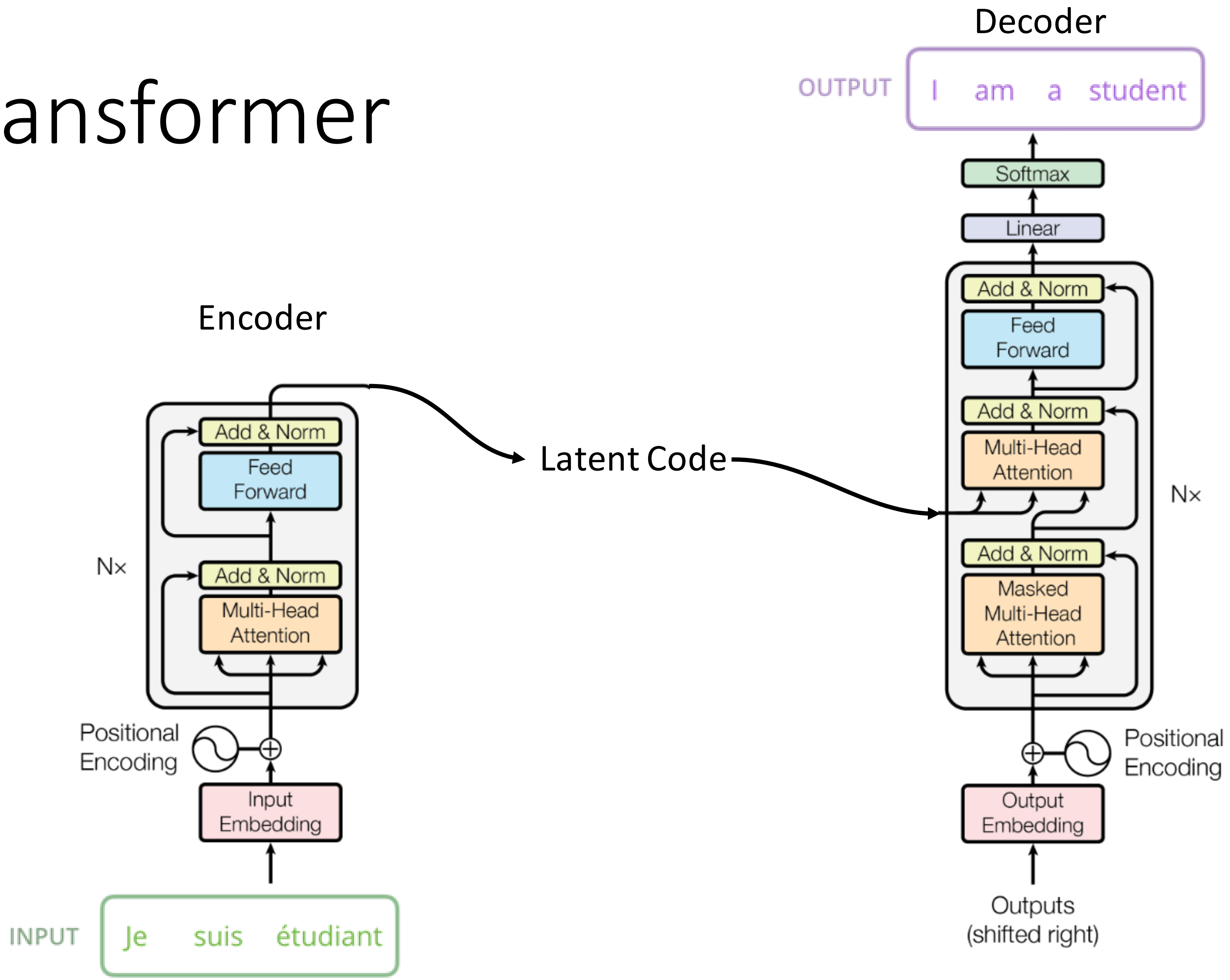


  - Ensure correct receptive field, e.g. special convolutions



- Sampling:
  - One sample at a time
  - Slow, involves repeated application of model

The Transformer



The Encoder

- Process set of tokens
- Tokens remain separate
  - (except for attention layer)
- Tokens don't have order
  - (except for positional encoding)