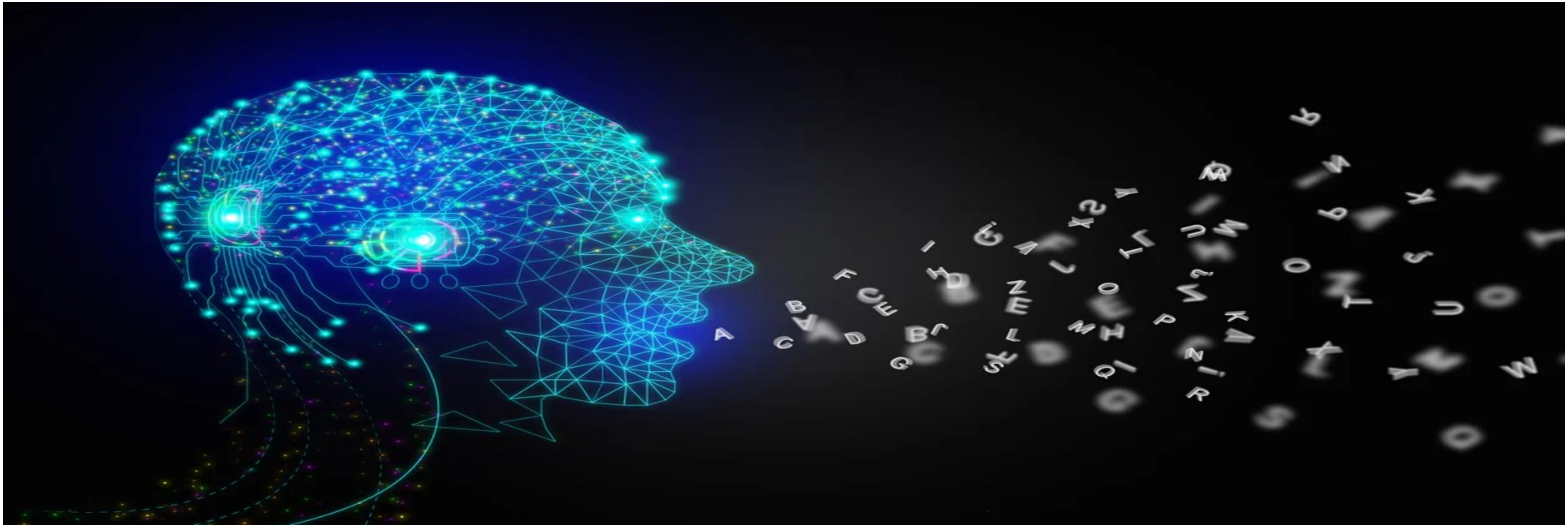
GENERATIVE AI

ON THE CONCEPT AND HISTORICAL PERSPECTIVE OF GENERATIVE AI



PRESENTED BY VAHID MOHAMMADZADEH EIVAGHI

CO-FOUNDER AT VIRA AI GROUP - SPECIALIZED IN THE APPLICATION OF CV IN INDUSTRY

Part 1:

INTRODUCTION

Generative modeling vs discriminative modeling, pros and cons

PRESENTED BY VAHID MOHAMMADZADEH EIVAGHI

CO-FOUNDER AT VIRA AI GROUP - SPECIALIZED IN THE APPLICATION OF CV IN INDUSTRY

MACHINE LEARNING SYSTEMS

Supervised learning

- There is supervision data forcing model to produce the same supervision given input variables.

Unsupervised learning

- There is no supervision data, and the model force to discover existing patterns.

Reinforcement learning

- Machines learn based on a set of possible actions and policies

SUPERVISED LEARNING

- In supervised setting, we have a dataset $S = \{x_k, y_k\}_{k=1}^{N}$, and we are seeking to find a mathematical function to map from input space spanned by $x_k \in R^d$ to output space spanned by $y_k \in R^p$.
  - Discriminative modeling – approximate the conditional distribution $P(y|x)$ indirectly, without requiring the distribution of data.
    - Linear regression, logistic regression, decision tree, MLP, CNN, RNN, transformers, …
  - Generative modeling – approximate the conditional distribution $P(y|x)$ directly, relying on the distribution of data.
    - Naïve Bayes, Linear/quadratic discriminant function.

UNSUPERVISED LEARNING

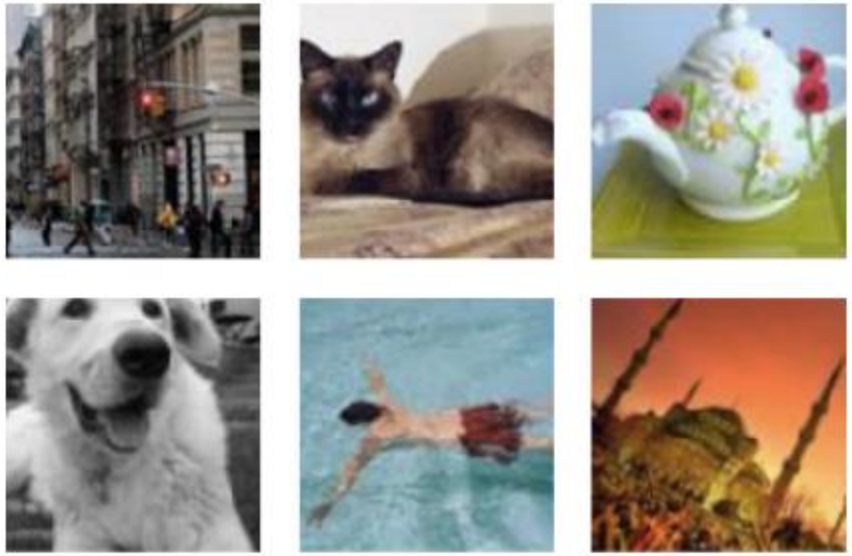
- In unsupervised setting, we have a dataset $S = \{x_k\}_{k=1}^{N}$, there is no target to which we find a mapping from input, thus nothing to predict nor to discriminate.
  - Pattern discovery – create a homogenous group of objects.
  - Structure learning – detect structure and infer the relationship between variables.
  - PDF estimation (generative modeling) – model the joint distribution over observation through either latent variable models or without it.

WHY GENERATIVE MODELING

1. Improving the discriminative models
   - How discriminative models create a mapping? -> they uses some sort of distance measuring to perform the task -> similar samples belong to the similar categories -> discriminative features say the last words!
   - What about the objects from the same class with different characteristics?
2. Sampling itself – content generation
   - Content generation -> the main goal of generative models in todays' world (Artificial Intelligent Generated Content (AIGC))
3. Inter-correlated structure detection

GENERATIVE MODELING – DEFINITION

Train from $x \sim P_{data}(x)$



Generate from $x \sim P_{model}(x)$

- We want to learn a model $P_{model}(x)$ similar to $P_{data}(x)$

GENERATED SAMPLES

Karras, Tero et al. (2018). \Progressive Growing of GANs for Improved Quality,Stability, and Variation". In: International Conference on Learning Representations

Part 2:

HISTORICAL PERSPECTIVE

From GMM to ChatGPT, the most important tools blooming generative AI

PRESENTED BY VAHID MOHAMMADZADEH EIVAGHI

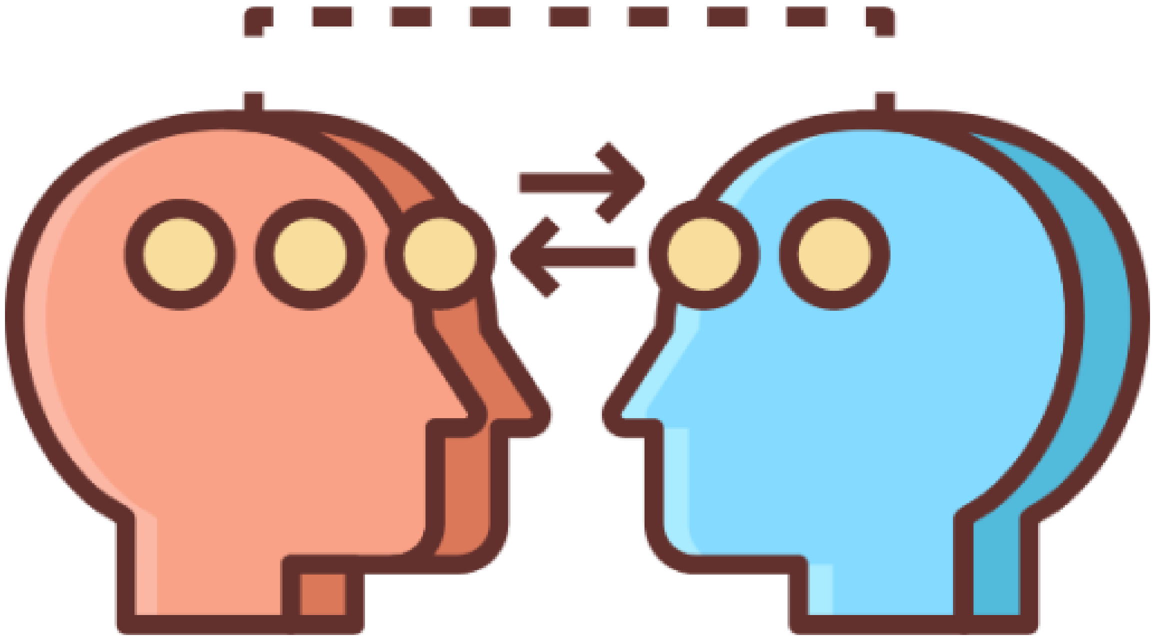
CO-FOUNDER AT VIRA AI GROUP - SPECIALIZED IN THE APPLICATION OF CV IN INDUSTRY

HISTORICAL PERSPECTIVE

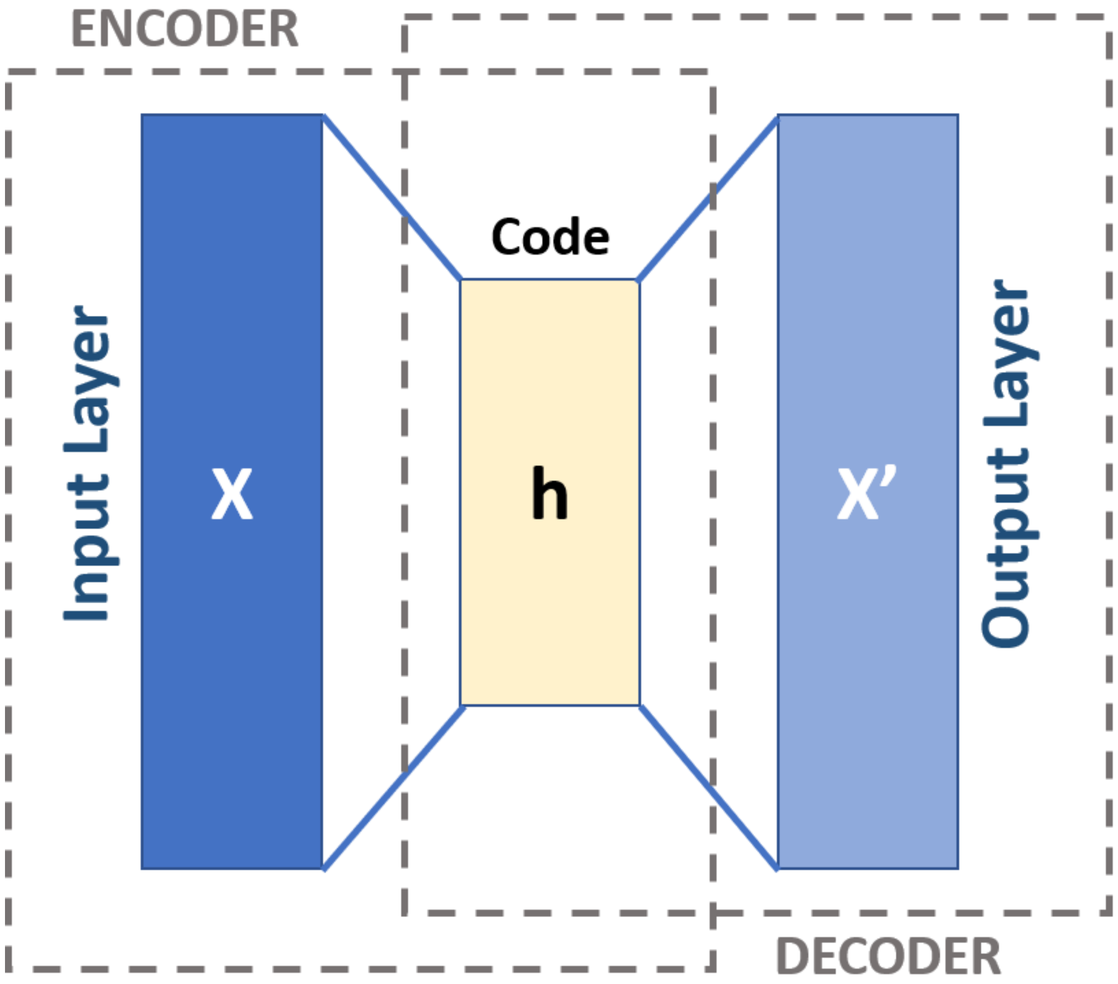
- Attempts for making generative models dating back to 1950, started from introducing GMM and HMM for sequential data.
  - Limited performance and major restriction on utilizing for high dimensional space.
- Image generation based on manipulated samples texture synthesize, and text generation based on word distribution estimation using N-gram.
- Deep learning emergence
  - Structure and technologies advancement – Energy based models, GAN, VAE, autoregressive models, BERT, BART, GPT, DALLE-2, CLIP, Bloom, …

PRE-TRAINING STRATEGIES

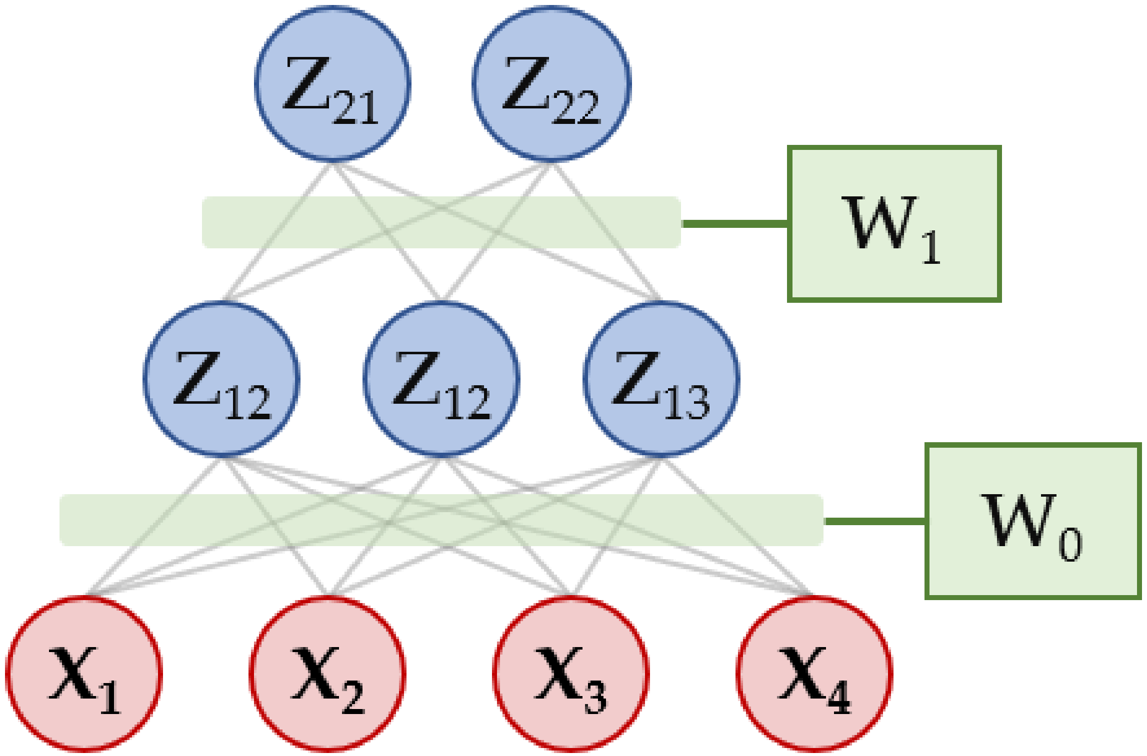
- The model is trained to perform well on unspecific task to expect perform a good performance in all related down-stream tasks -> transfer learning
  - Data understanding



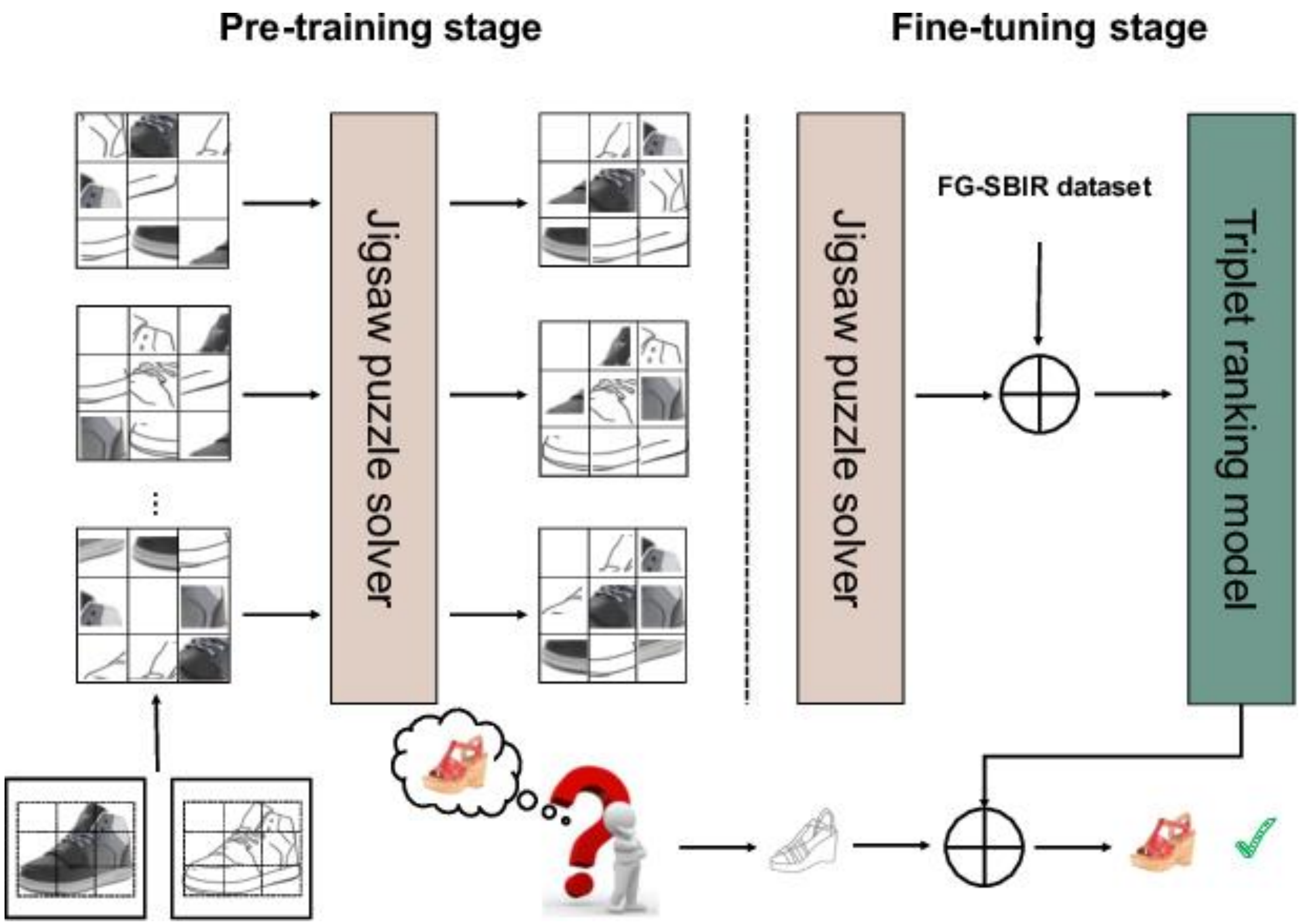
DIRECT MAPPING



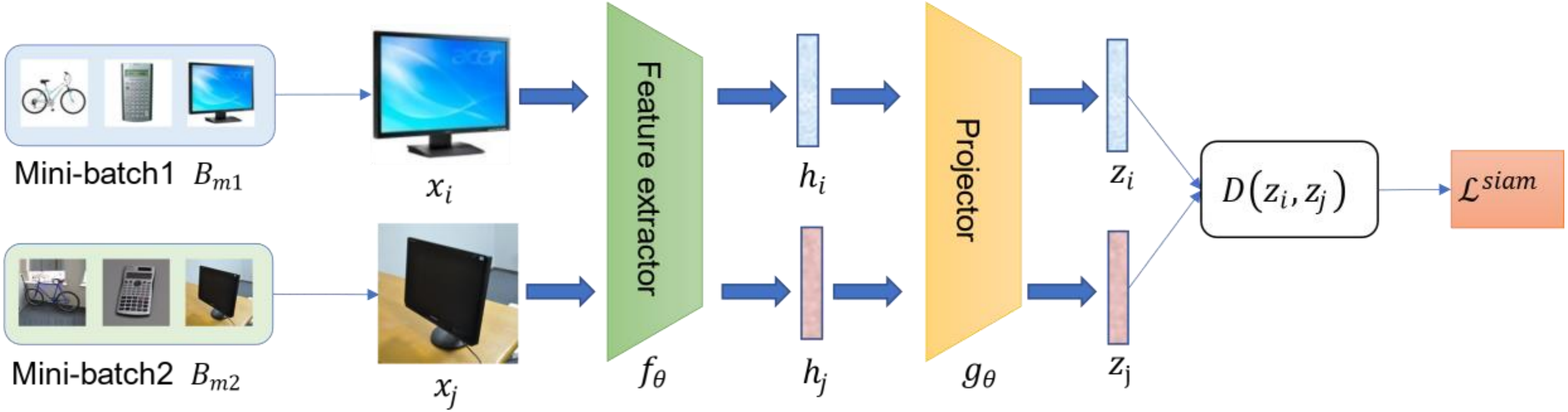
LATENT VARIABLE MODELS



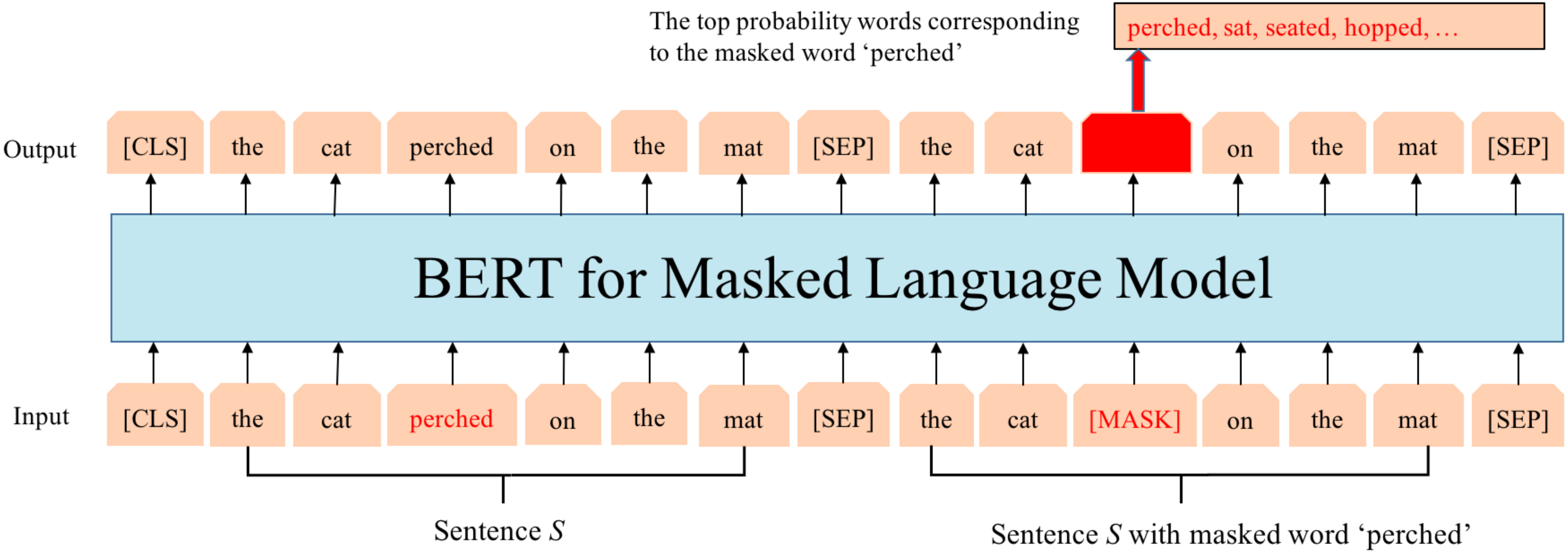
SOLVING JIGSAW PUZZLE



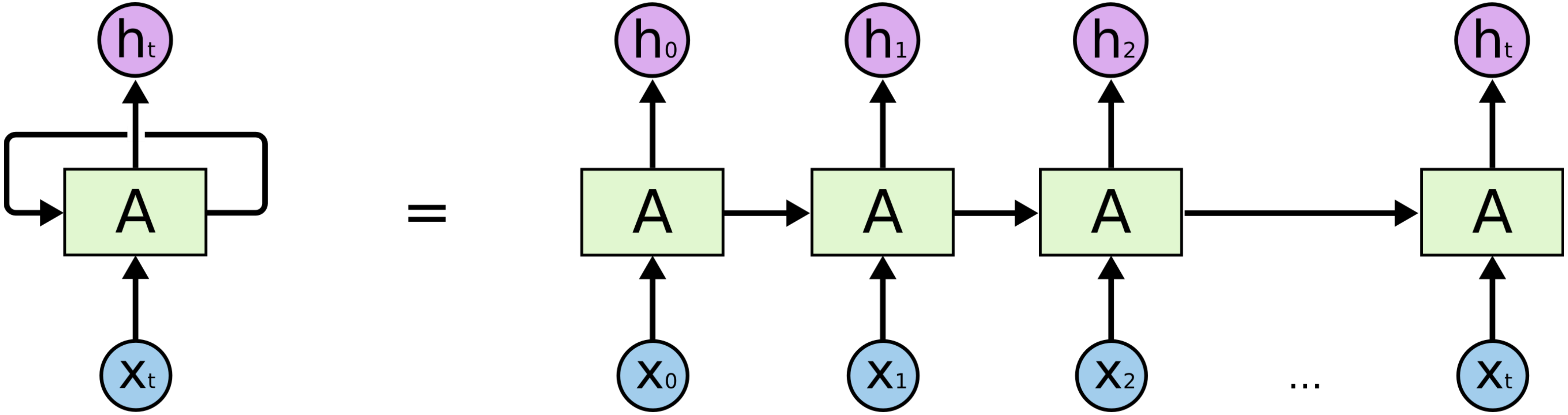
CONTRASTIVE LEARNING



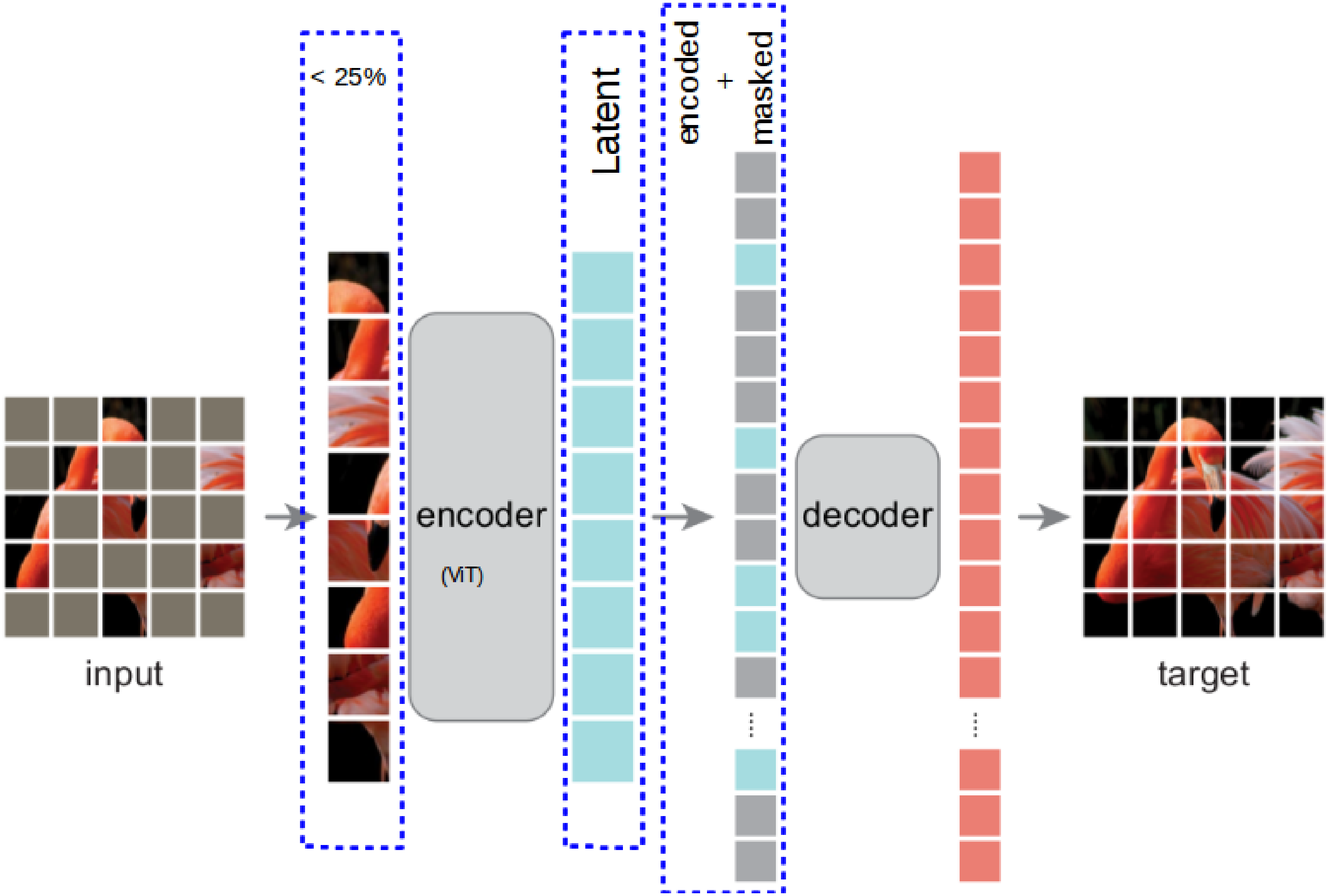
MASKED LANGUAGE MODELS



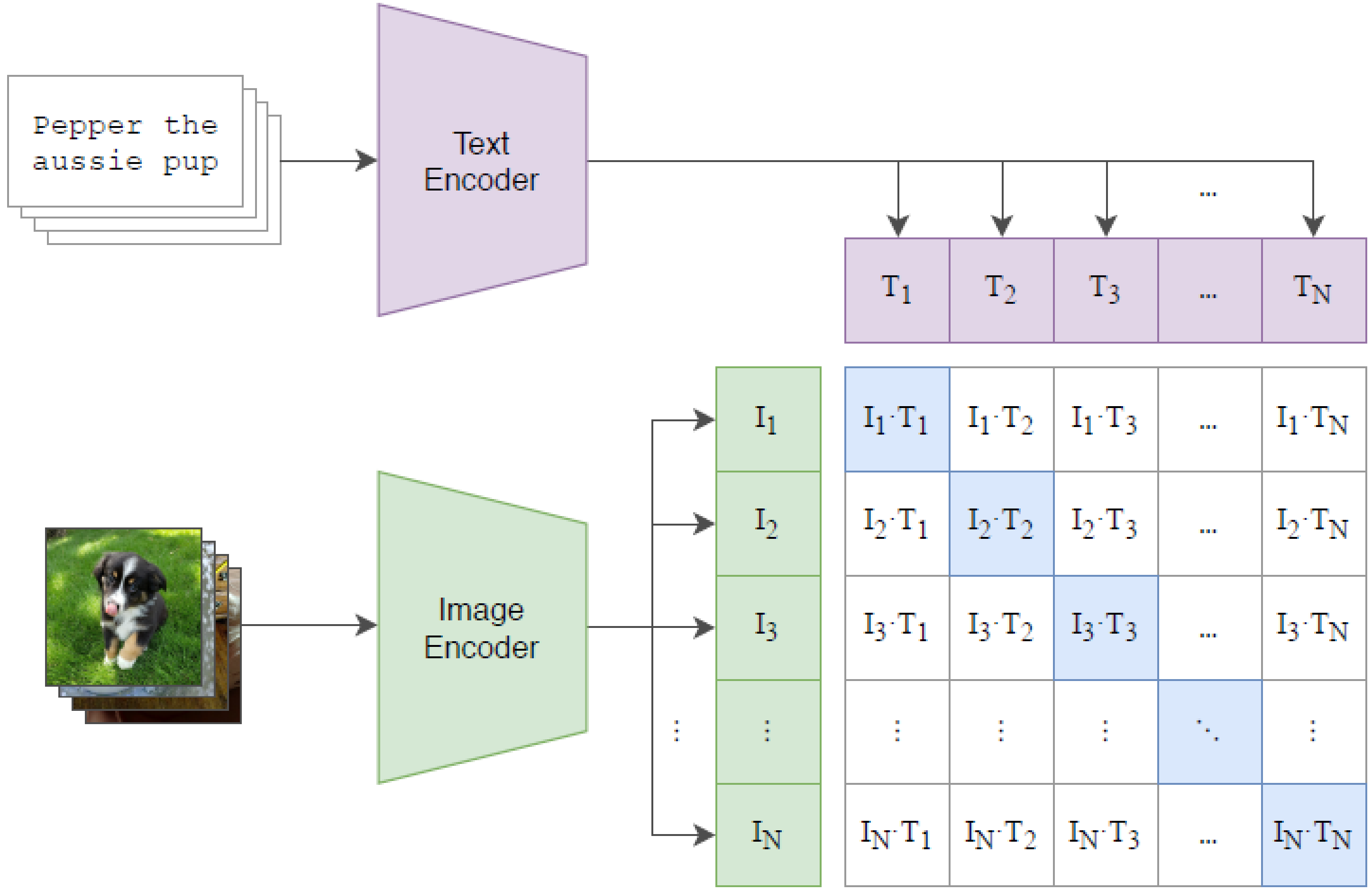
AUTOREGRESSIVE LANGUAGE MODELS



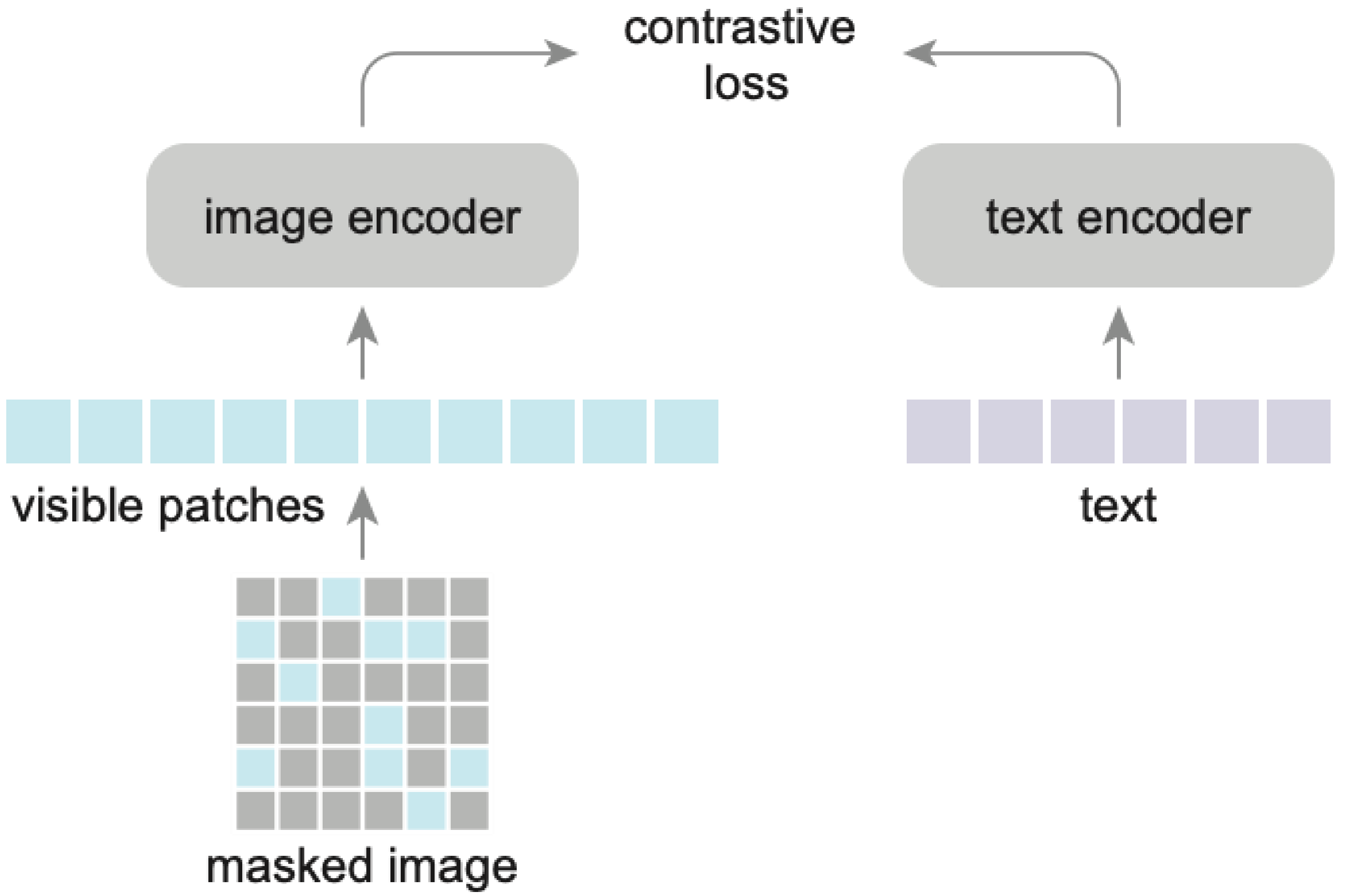
MASKED AUTO-ENCODER



MULTI-MODAL PRE-TRAINING



MASKED CLIP

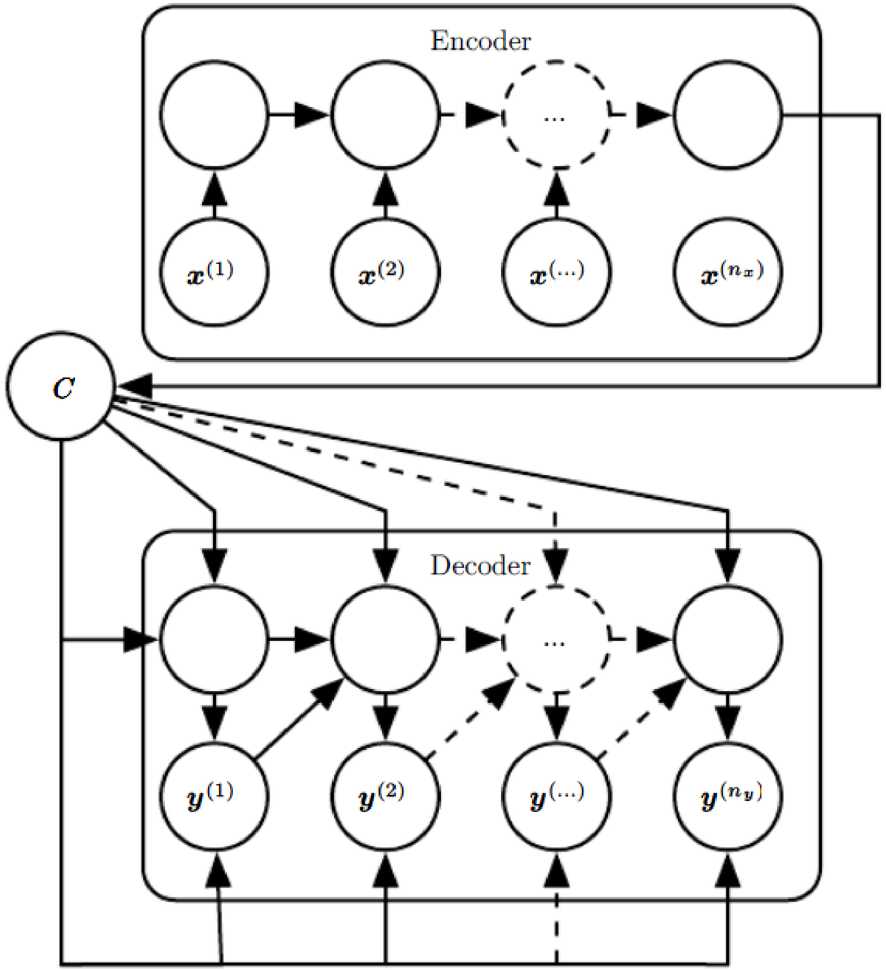


contrastive loss
image encoder
text encoder
visible patches
text
masked image

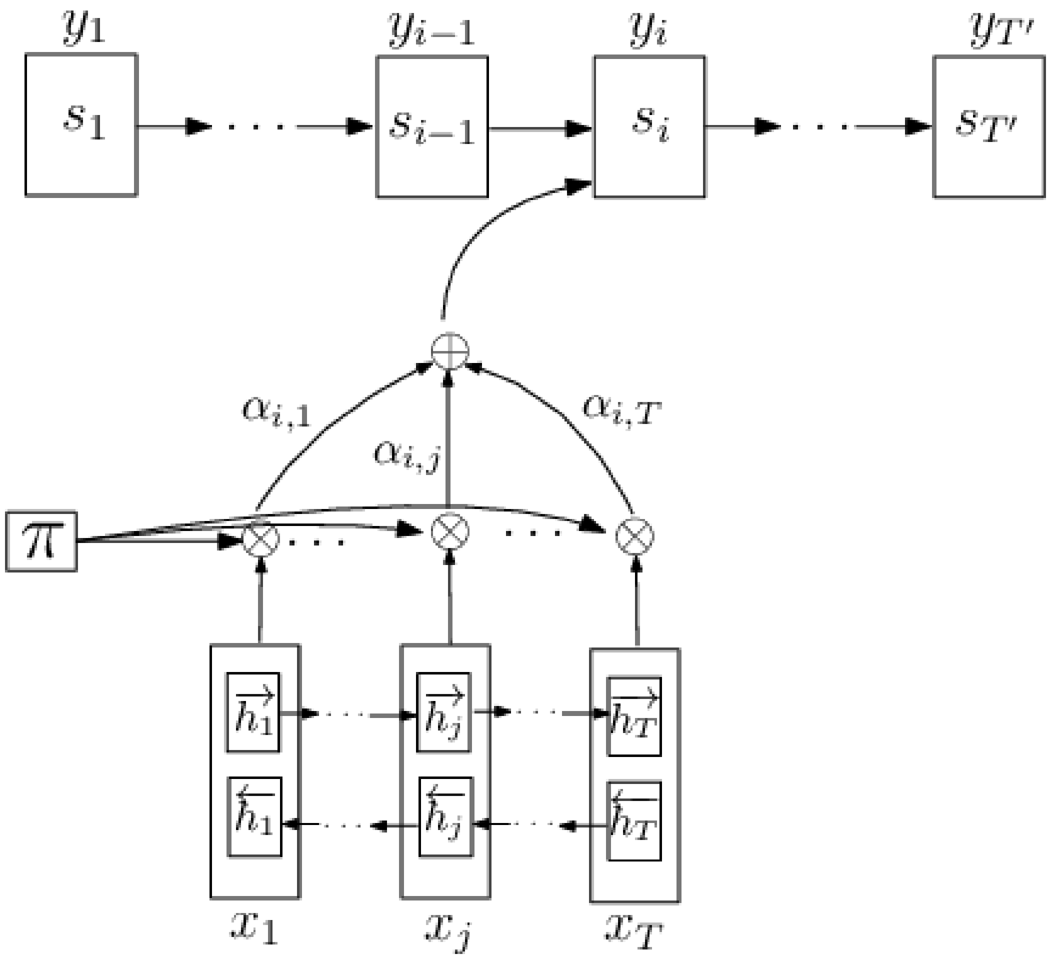
ATTENTION IS ALL YOU NEED

- Transformers are a specific type deep neural network originally developed for neural machine translation.
  - Transformers make it possible to process a sequence of tokens in parallel in exchange for the high number of parameters.
  - Self attention is core module of transformers.
- Preliminary
  - We need to know what are attention and self-attention mechanisms

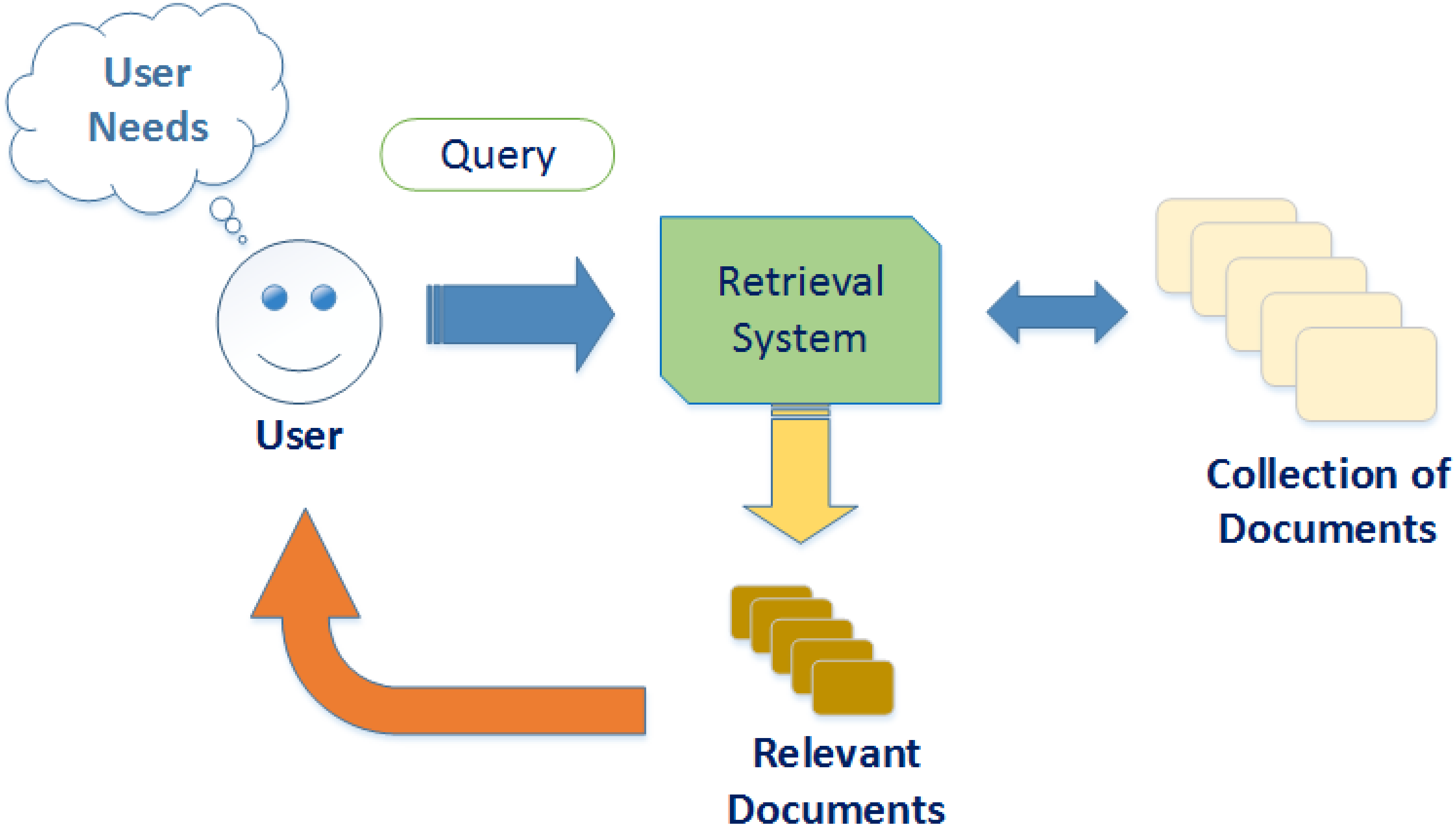
ENCODER-DECODER ARCHITECTURE



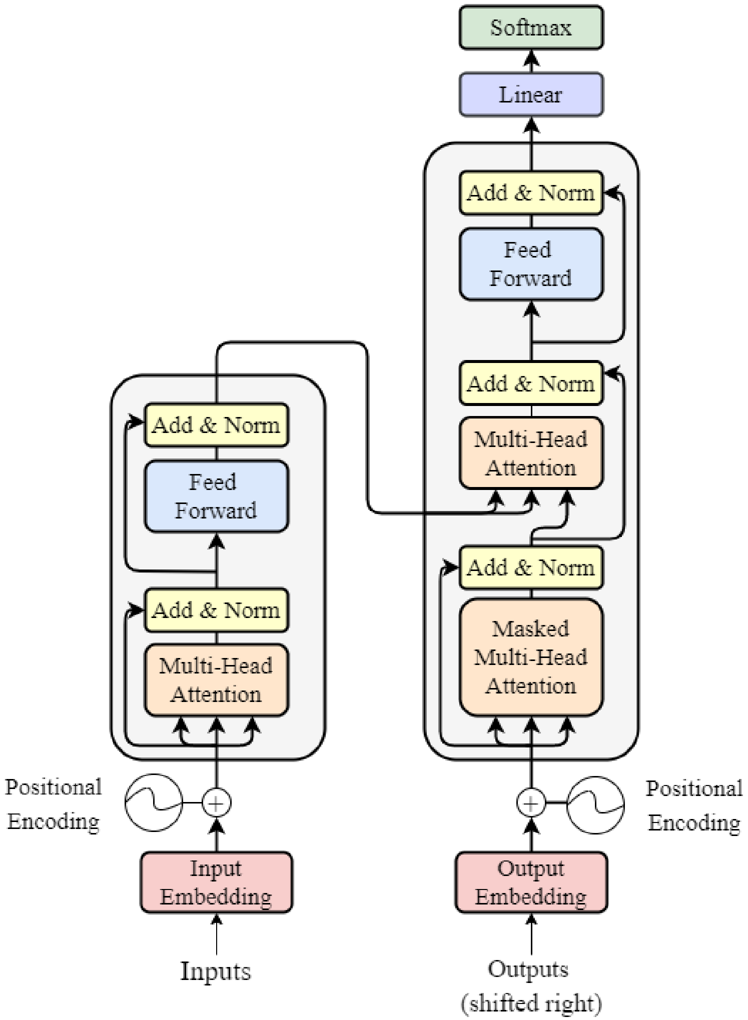
ATTENTION MECHANISM



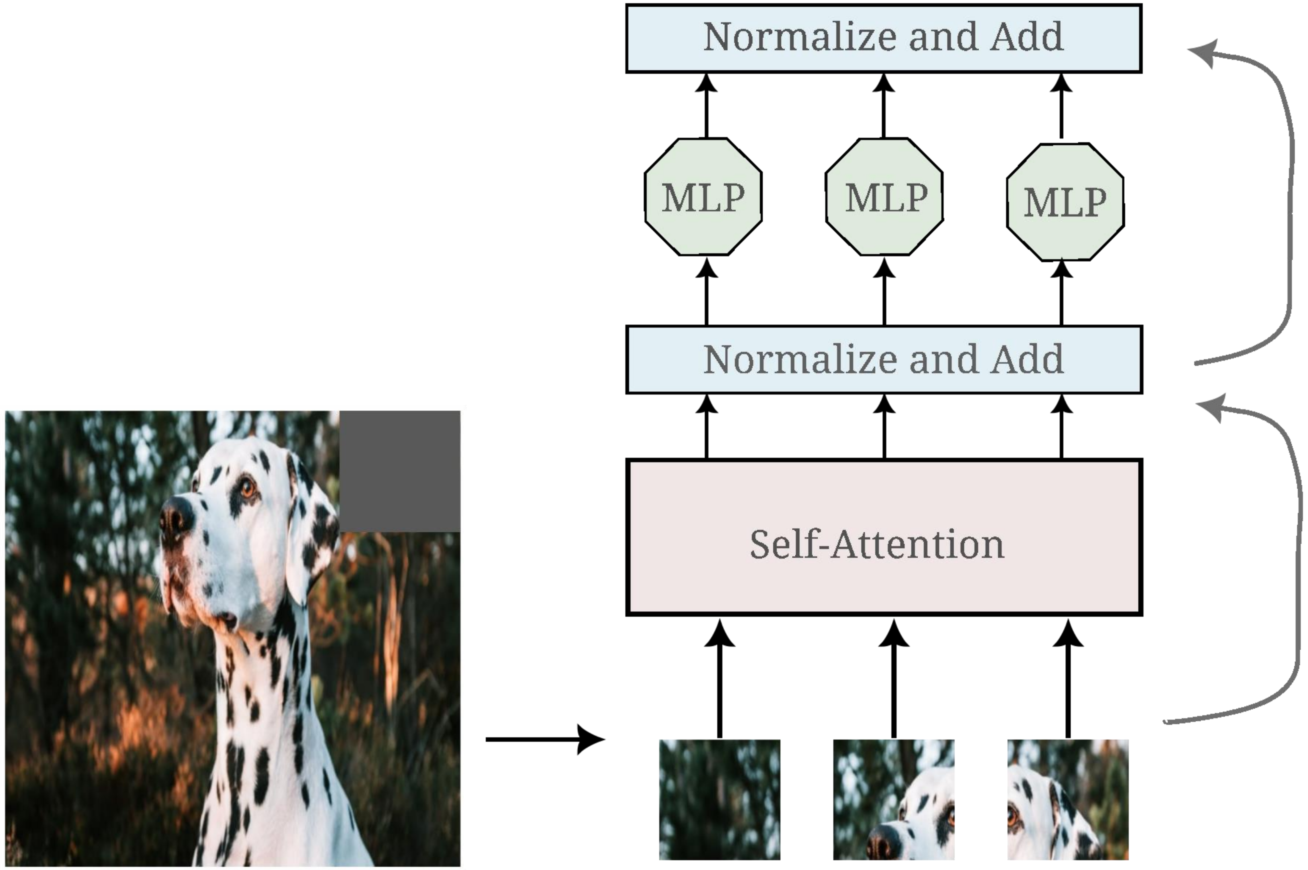
INFORMATION RETRIEVAL SYSTEM



SELF-ATTENTION MECHANISM



VISION TRANSFORMERS (VIT)



Part 3:

GENERATIVE MODELS

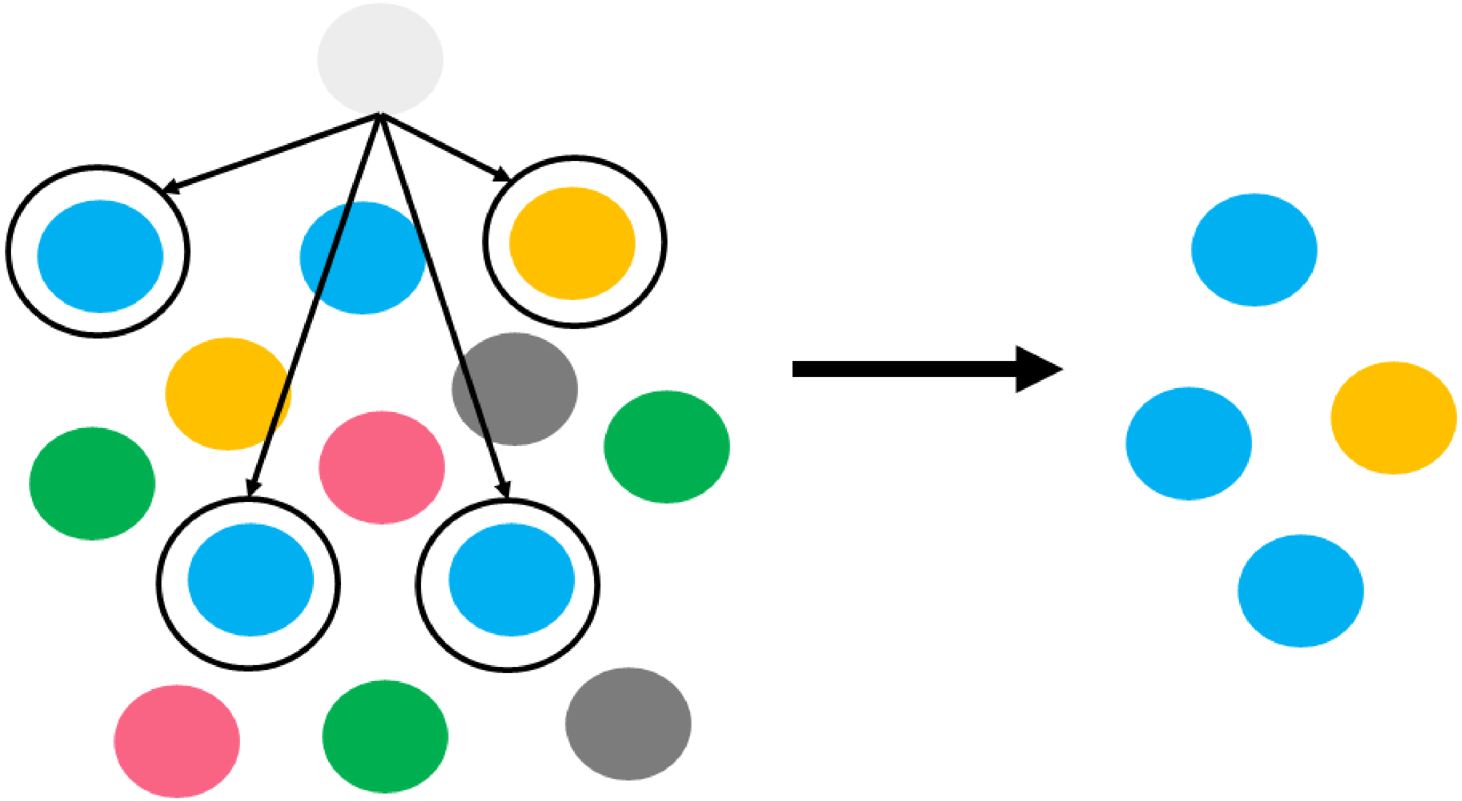
Types, tools, architectures, algorithms, and sampling principles

PRESENTED BY VAHID MOHAMMADZADEH EIVAGHI

CO-FOUNDER AT VIRA AI GROUP - SPECIALIZED IN THE APPLICATION OF CV IN INDUSTRY

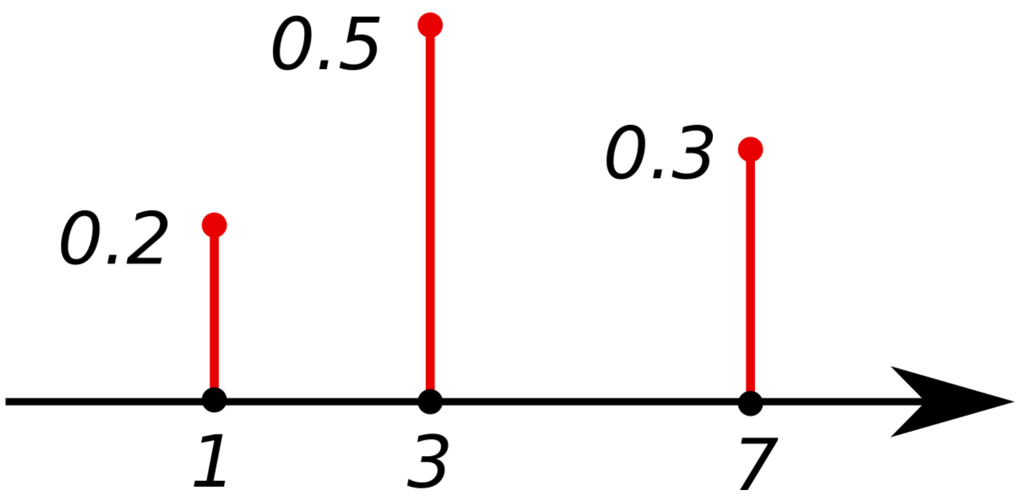
SAMPLING

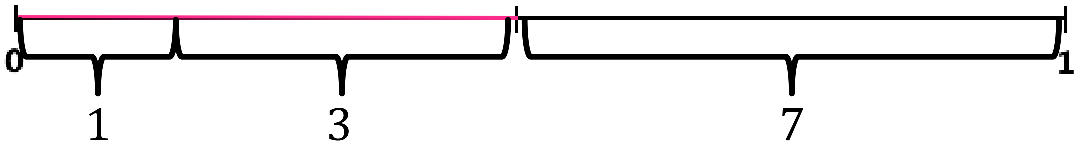
- Given a probability distribution $p(x)$, how one can draw samples from it?



DISCRETE RANDOM VARIABLES

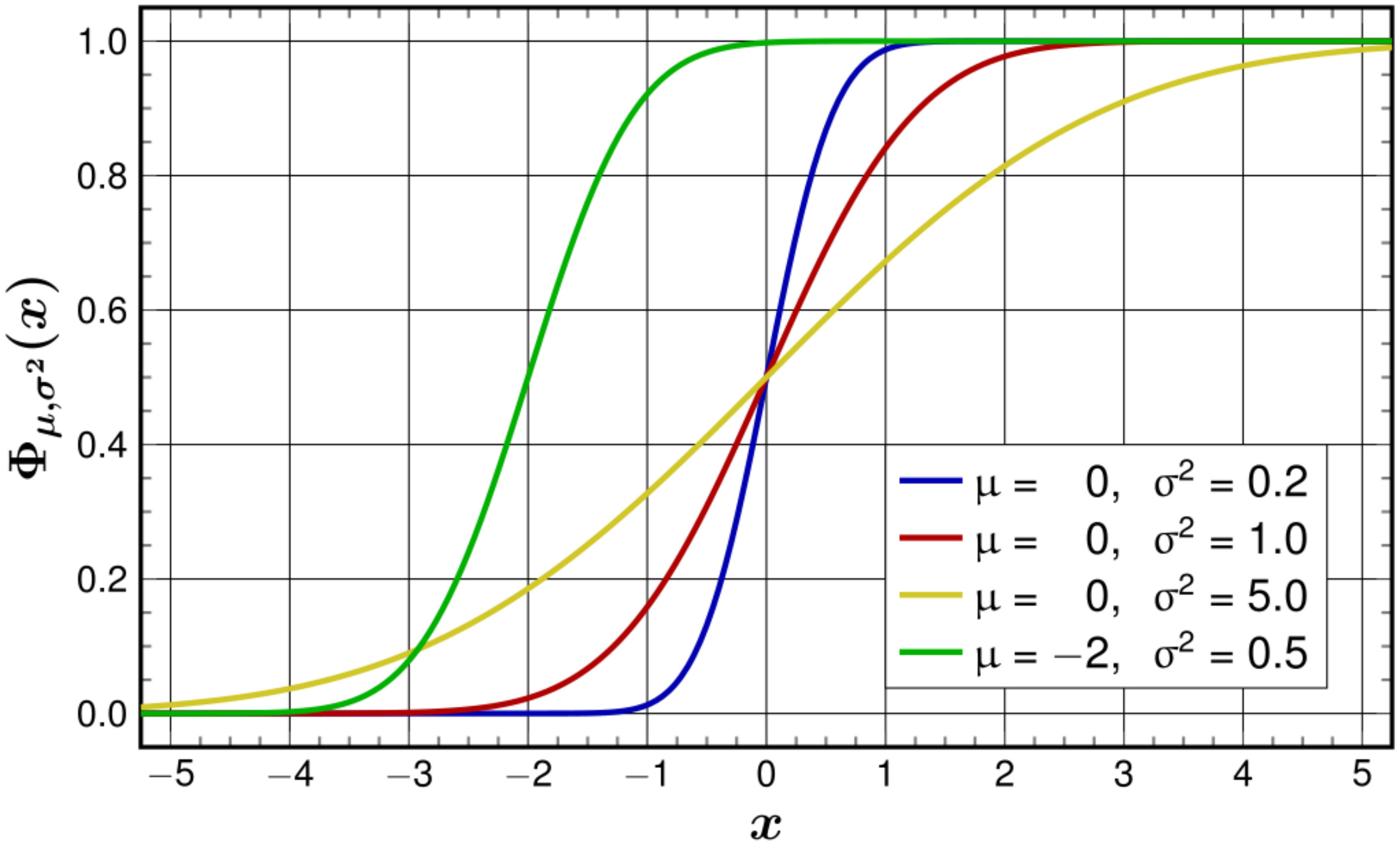
- If it is assumed that there is a PDF in hand, from which we can draw samples by sampling from an uniform distribution.





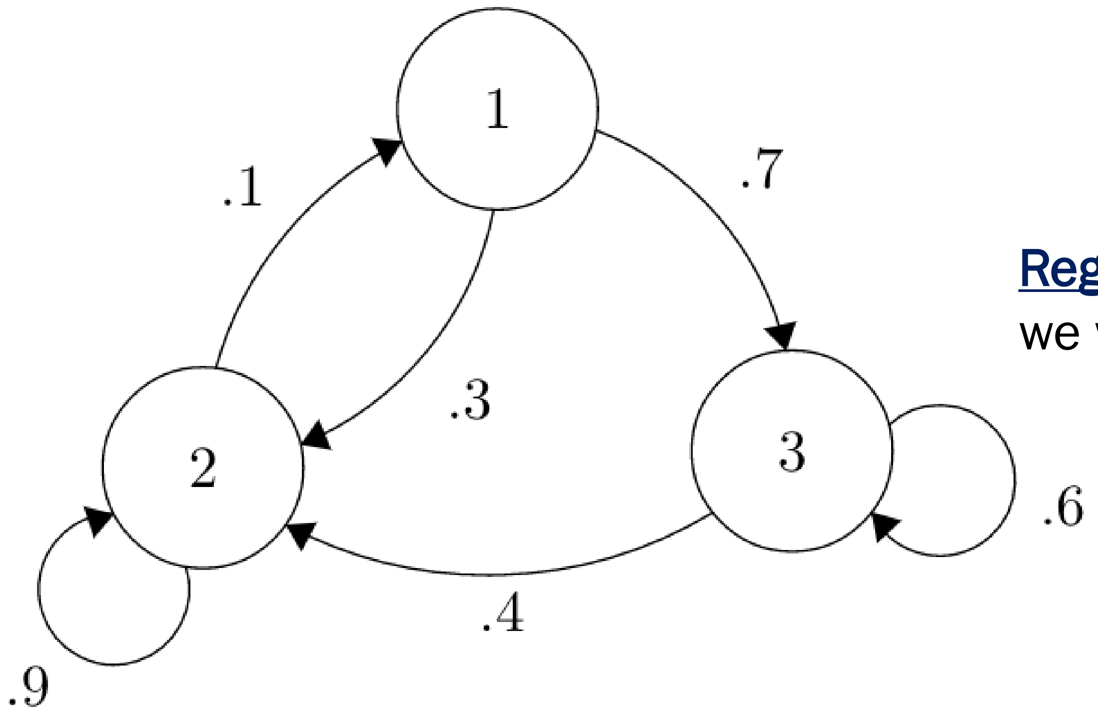
INVERSE CDF TRANSFORM

- What about continues distributions?



MONTE CARLO MARKOV CHAIN

- Define a Markov chain with stationary distribution of the one from which we are going to sample.



$$x_{t+1} = P x_t = \begin{pmatrix} 0 & 0.1 & 0 \\ 0.3 & 0.9 & 0.4 \\ 0.7 & 0 & 0.6 \end{pmatrix} x_t$$

Regular Markov chain: From any arbitrary initialization we will reach the same distribution

$$\pi = P\pi$$

GIBBS SAMPLING

- Define a Markov chain with stationary distribution of the one from which we are going to sample.

Gibbs sampling uses the following procedure

- Repeat until convergence for $t = 1, 2, \ldots,$
  - Set $\mathbf{x} \leftarrow \mathbf{x}^{t-1}$.
  - For each variable x_i in the order we fixed:
    1) Sample $x_i' \sim p(x_i \mid \mathbf{x}_{-i})$.
    2) Update $\mathbf{x} \leftarrow (x_1, \ldots, x_i', \ldots, x_d)$.
  - Set $\mathbf{x}^t \leftarrow \mathbf{x}$.

We use $\mathbf{x}_{-i}$ to denote all variables in $\mathbf{x}$ except x_i.

TYPE OF GENERATIVE MODELS

- Generative models are grouped based on either the way they are trained or the final model they will provide.
  - Generative models are either trained based on maximum likelihood criterion or adversarial training
  - Generative models give us either a probability density function or just sampling mechanism.

PARAMETRIC DENSITY ESTIMATION

- A specific form of distribution is assumed, whose parameters are estimated using data
  - Given an iid set of samples $\{x_1, \dots x_N\},\ x_i \in R^d$, a distribution with known form $P_\theta(x)$ is defined as the following:

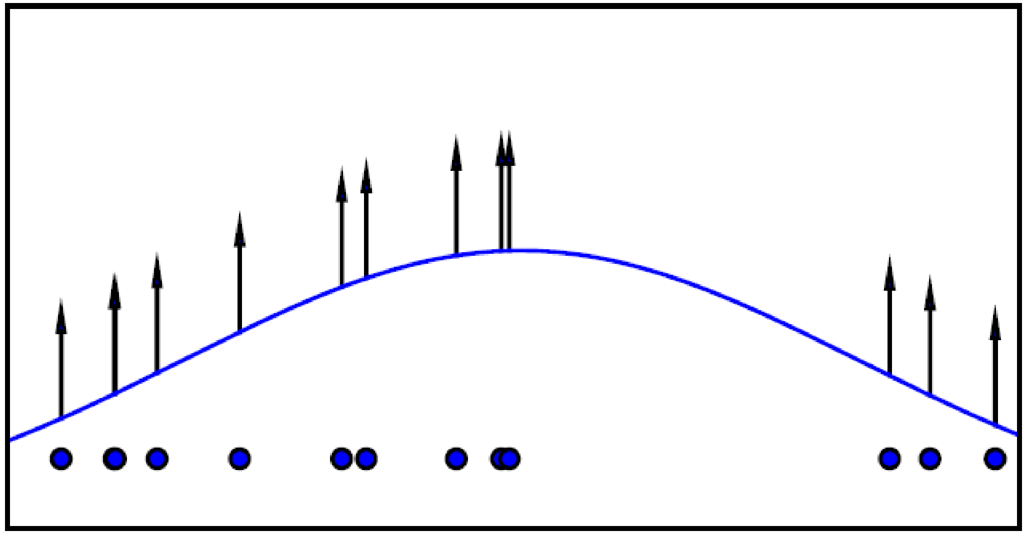
$$P_\theta(x) = \prod_{k=1}^{N} P_\theta(x_k)$$

  - The parameters θ is estimated through maximizing the log-likelihood. **Why log-likelihood**?

MLE SOLUTION

- Take the derivative with respect to the parameters:

$$LL(\theta) = \sum_{k=1}^{N} \ln P_\theta(x_k) \rightarrow \theta^* = \arg\max_\theta LL(\theta)$$

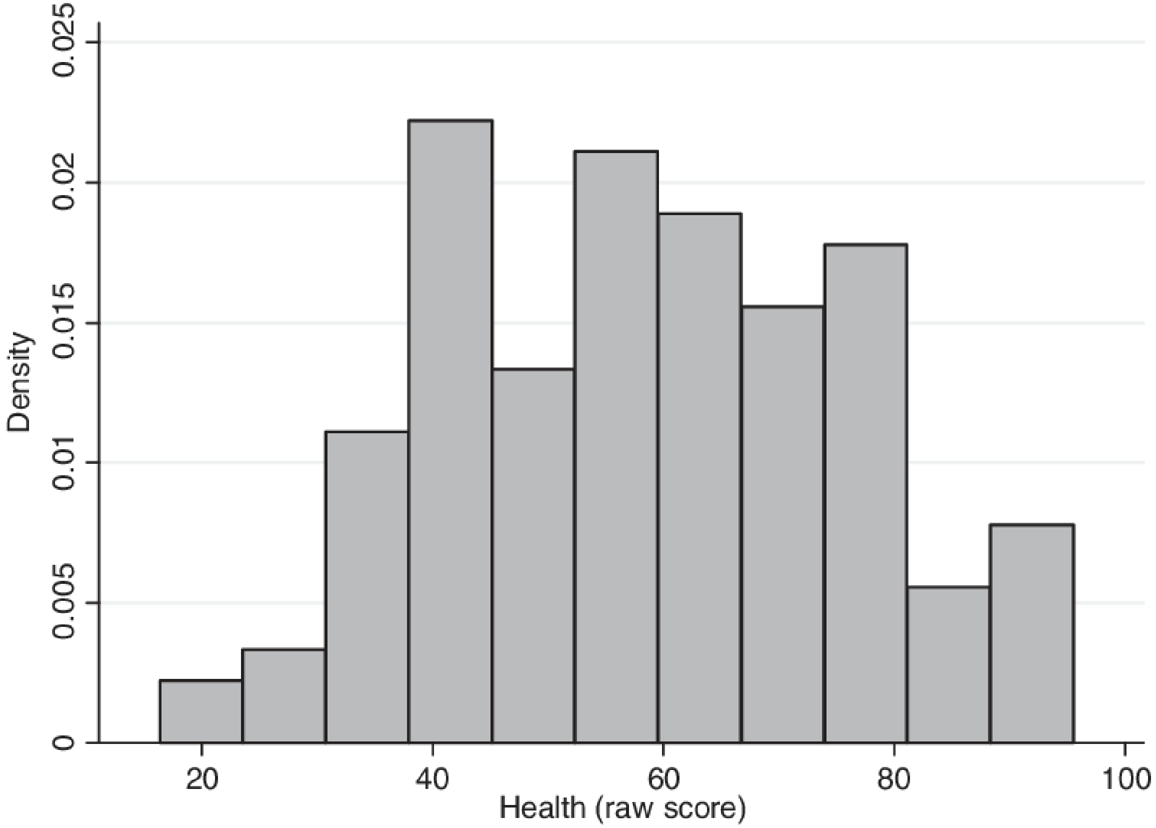


NON-PARAMETRIC MODELS

- Which form we should select to be matched to given data?
  - Often, one about which we think is far from the reality.
  - The parametric models are often unimodal while the real world is multimodal.
  - High-dimensional parameter space
- Non-parametric models
  - Parzen
  - K-nearest neighbors

HISTOGRAM

- How histograms are formed? For one dimensional data
  - Sort data in descending order and divide it into some intervals



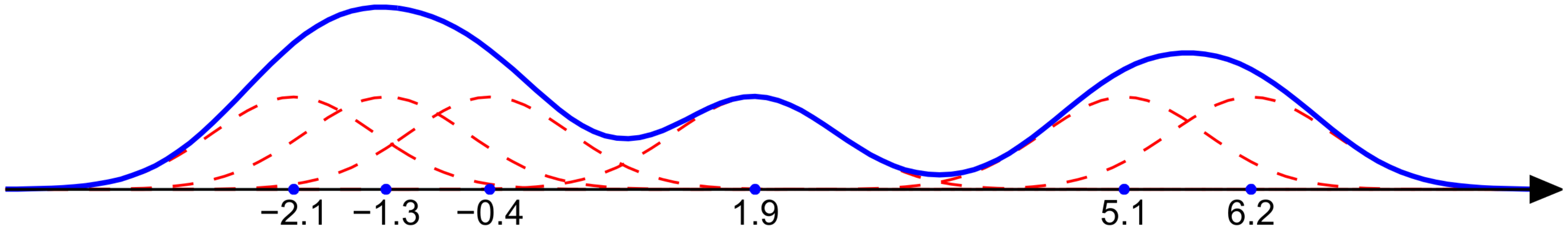
- Intervals are arranged with an assumption where the density is defined as the proportion of samples falling into each intervals.
- The volume should be small enough to be ensured over which the density is constant

$$\int p(x)dx = \frac{K}{N} \rightarrow p(x)V = \frac{K}{N} \rightarrow p(x) = \frac{K}{NV}$$

PARZEN WINDOW

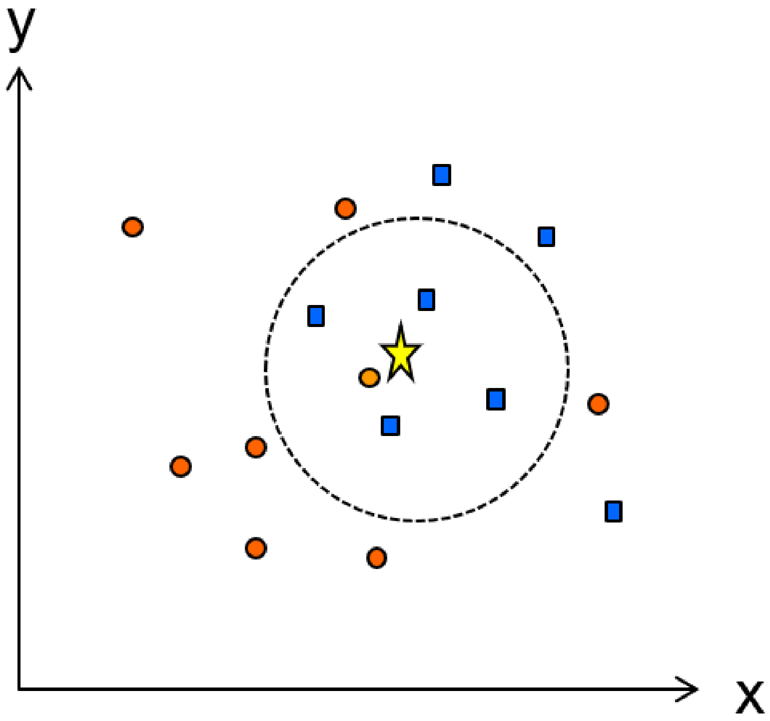
- An extension over histogram methods for high dimensional space
  - The basic utilities of kernel function

$$p(x) = \frac{1}{N}\sum_{n}\frac{1}{h^D}k\left(\frac{x - x_n}{h}\right)$$



KNN

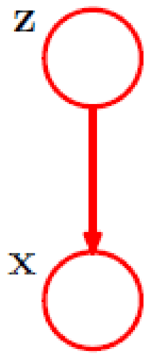
- It performs like Parzen window with an exception where the volume is changed.



- Sort training samples based on their distances to a selected test sample.
- KNN will not give us the likelihood distribution since its integration over the space will be diverged. **How?**
- Euclidean kernel is usually used, while using complex kernels is also possible. **What we mean of complex kernels?**

GAUSSIAN MIXTURE MODELS (GMM)

- GMM is a simple class of latent variable models, where the latent space is formed by K-dimensional discrete variable.

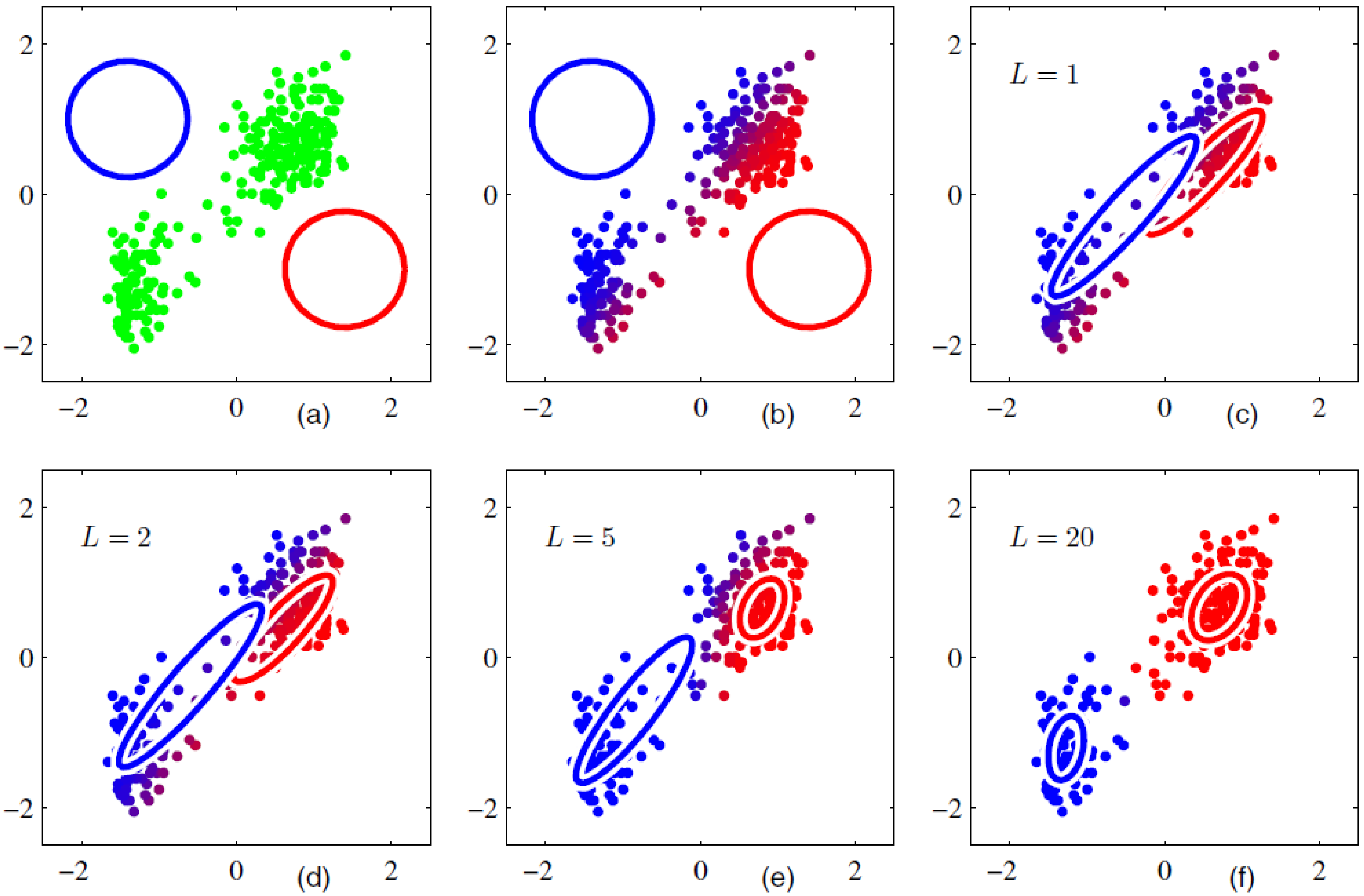


$$p(x) = \sum_{z} p(z)p(x|z), \qquad p(z) = \prod_{k=1}^{K} \pi_k^{z_k}, \qquad p(x|z_k = 1) \sim N(x|\mu_k, \Sigma_k)$$

$$p(x) = \sum_{k} p(z_k = 1)p(x|z_k = 1) = \sum_{k} \pi_k N(x|\mu_k, \Sigma_k)$$

- Similar to parametric models, the structure of the model is fixed and only remained step is parameter estimation

EM ALGORITHM



GMM FOR SEQUENTIAL DATA

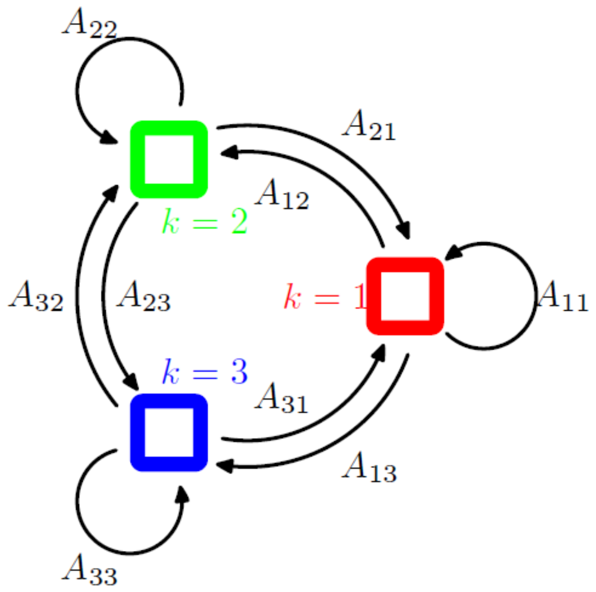
- Sequential data
  - A simple vector with an additional dimension that has physical meaning (time or order)
- How GMM can be extended to deal with sequential data?

$$z = \{z_0, z_1, z_2, \dots, z_T\} \Longrightarrow X = \begin{pmatrix} z_0 & z_1 & \cdots & z_{d-1} \\ \vdots & & \ddots & \vdots \\ z_{n-d} & z_{n-d+1} & \cdots & z_{n-1} \end{pmatrix}$$

HIDDEN MARKOV MODELS (HMM)

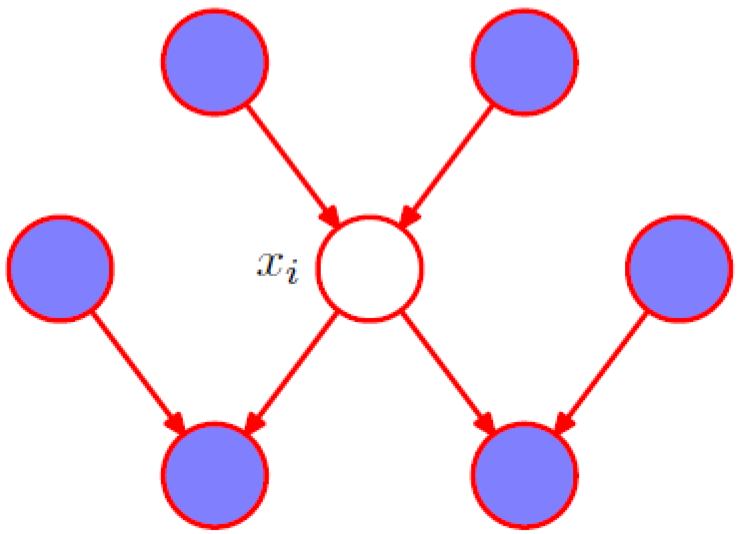
- Hidden (latent) markov model is a mathematical system whose states are limited to be countable -> an instance of state space models
  - The observations $y_{1:T}$ are generated by a set of unobservable variables $z_{1:T}$

$$P(y_{1:T}, z_{1:T}) = P(z_1) \prod_{t=1}^{T} P(z_t | z_{t-1}) P(y_t | z_t)$$

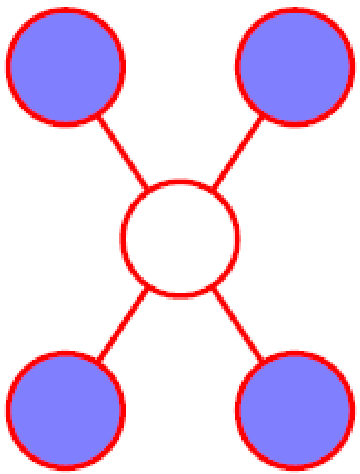


GRAPHICAL MODELS

Bayesian network

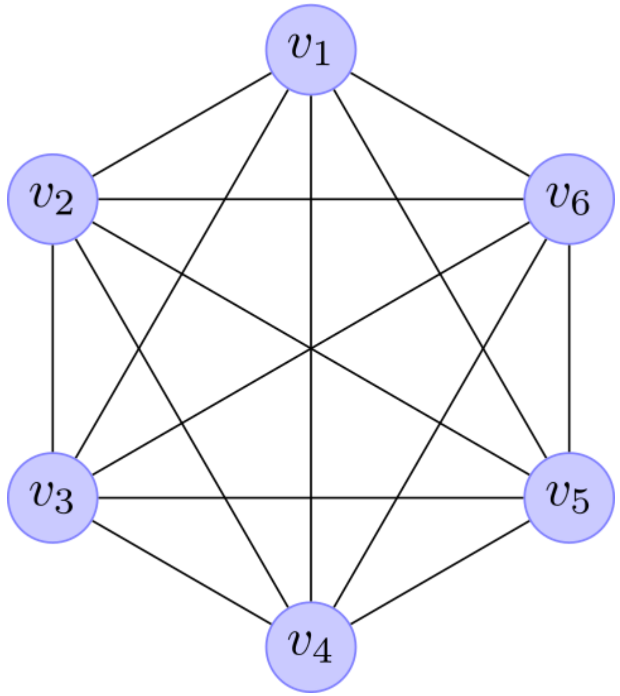


Markov random field



BOLTZMANN MACHINES (BM)

- BMs are fully connected Markov Random Field (MRF) -> **what are MRFs**?



- MRFs are a specific type of probabilistic graphical models factorizing the joint distribution over some variables as the product of some positive terms, so-called potential functions.
- In BMs, potential functions are defined using energy concept, introduced from statistical mechanics.

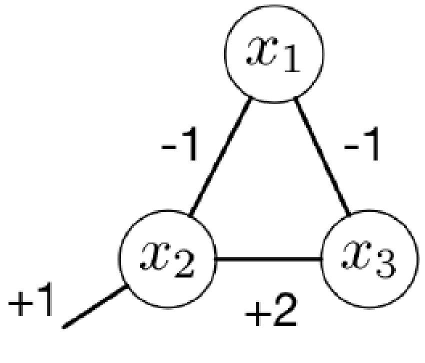
FULLY VISIBLE BM

- In a fully visible network, the energy function is defined as the following:

$$E(x) = -x^T W x - b^T x \rightarrow \quad P(x) = \frac{1}{Z}\exp\big(-E(x)\big), \qquad Z = \sum_x \exp\big(-E(x)\big)$$

x_1	x_2	x_3	$w_{12}x_1x_2$	$w_{13}x_1x_3$	$w_{23}x_2x_3$	b_2x_2	$H(\mathbf{x})$	$\exp(H(\mathbf{x}))$	$p(\mathbf{x})$
-1	-1	-1	-1	-1	2	-1	-1	0.368	0.0021
-1	-1	1	-1	1	-2	-1	-3	0.050	0.0003
-1	1	-1	1	-1	-2	1	-3	0.368	0.0021
-1	1	1	1	1	2	1	5	148.413	0.8608
1	-1	-1	1	1	2	-1	3	20.086	0.1165
1	-1	1	1	-1	-2	-1	-3	0.050	0.0003
1	1	-1	-1	1	-2	1	-1	0.368	0.0021
1	1	1	-1	-1	2	1	1	2.718	0.0158

$\mathcal{Z} = 172.420$



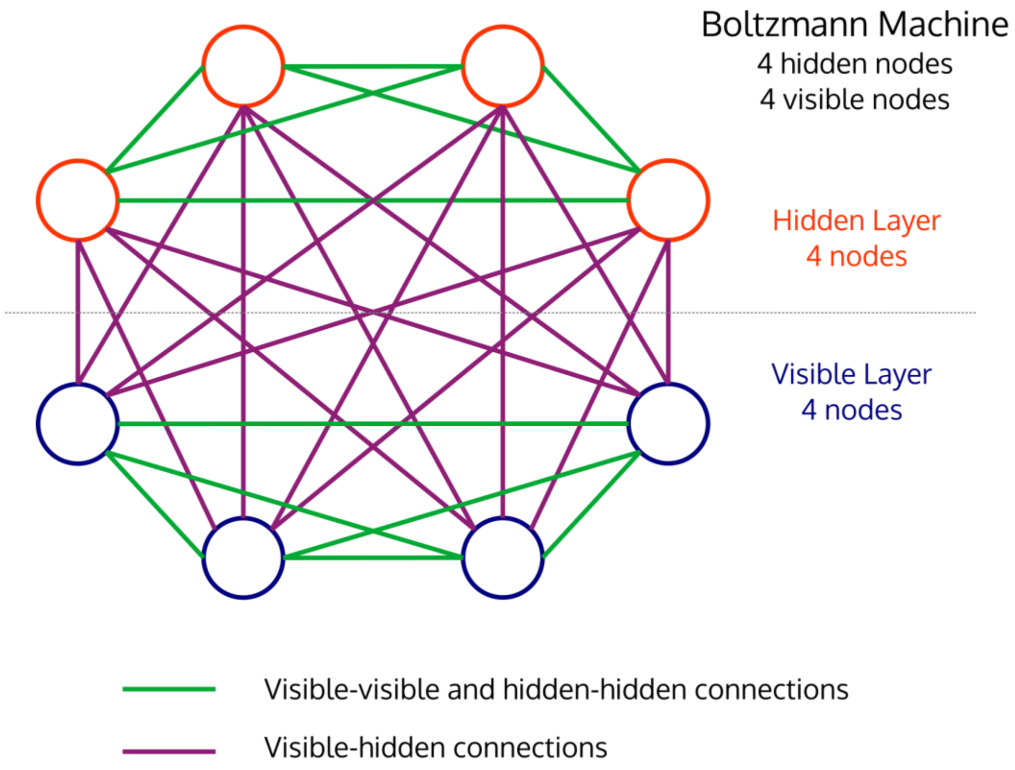
BOLTZMANN MACHINE WITH HIDDEN UNITS

- The power of BM will be shined if we have some hidden variables.

$$E(x,h) = -x^T W x - -h^T V h - x^T F h - a^T h - b^T x$$

$$P(x,h) = \frac{1}{Z} \exp\left(-E(x,h)\right),$$

$$Z = \sum_{x,h} \exp\left(-E(x,h)\right)$$



LEARNING IN MRF

- The learning is based on maximizing likelihood function using GD
  - All Boltzmann machines have intractable partition function

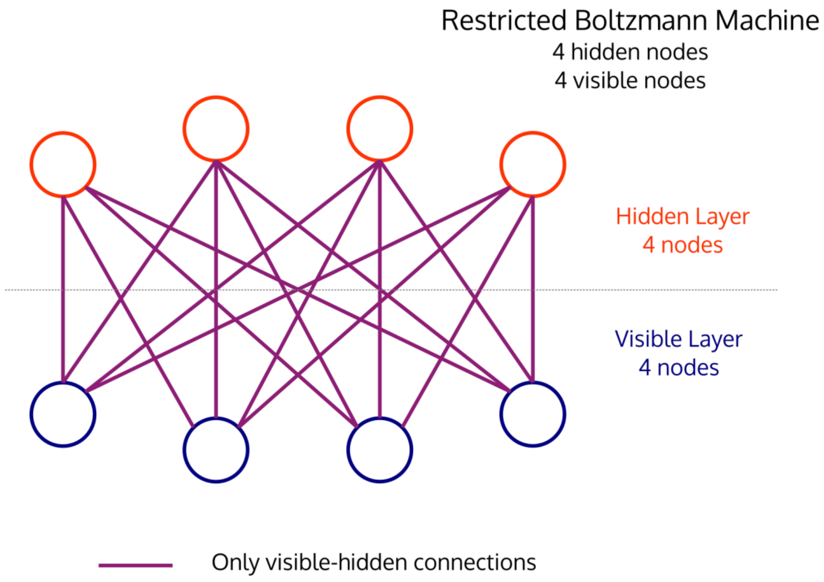
$$p(x;\theta) = \frac{1}{Z_\theta}\tilde{p}(x;\theta)$$

$$\nabla_\theta(\log p(x;\theta)) = -\nabla_\theta \log Z_\theta + \nabla_\theta(\log \tilde{p}(x;\theta)) = \nabla_\theta(\log \tilde{p}(x;\theta)) - \sum_x \frac{\tilde{p}(x;\theta)\nabla_\theta(\log \tilde{p}(x;\theta))}{Z_\theta}$$

$$\boxed{\nabla_\theta(\log p(x;\theta)) = \nabla_\theta(\log \tilde{p}(x;\theta)) - \mathrm{E}_{x\sim\tilde{p}(x;\theta)}[\nabla_\theta(\log \tilde{p}(x;\theta))]}$$

RESTRICTED BOLTZMANN MACHINE (RBM)

- The tractability of joint distribution defined by BMs limits their application in practice.
- RBM is an instance of Boltzmann machine formed using a bipartite graph.



- Make benefits from conditional independency

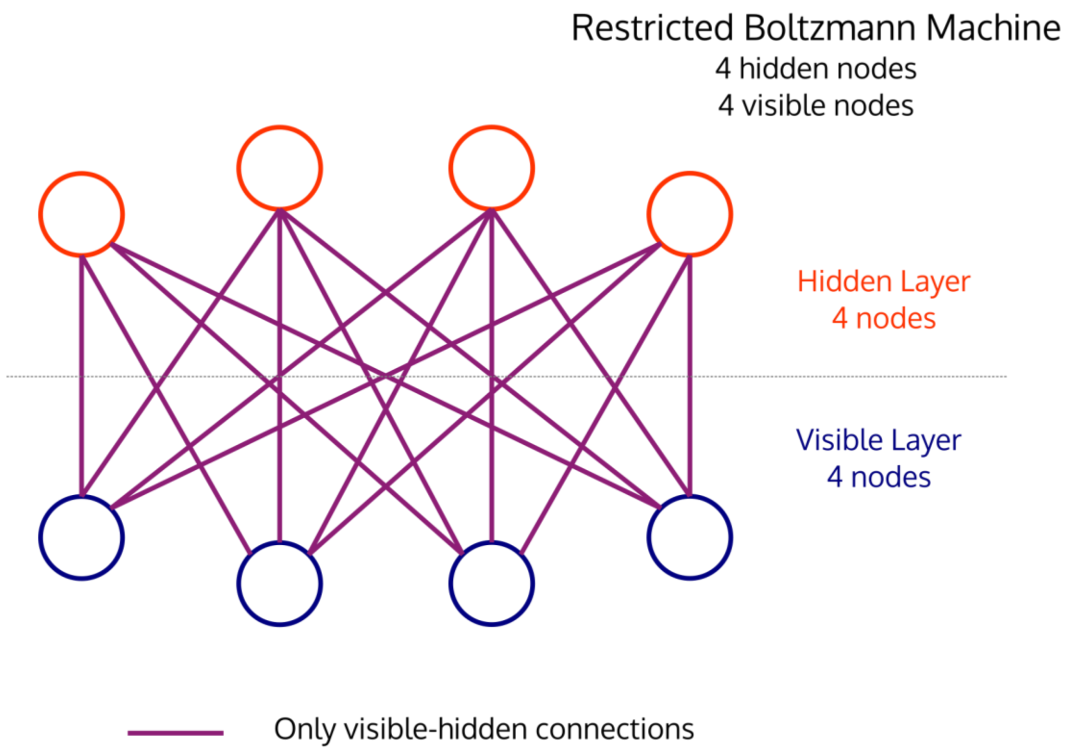
$$p(h|v) = \prod_{i=1}^{M} p(h_i|v),\, p(v|h) = \prod_{i=1}^{N} p(v_i|h)$$

LEARNING IN RBM

- Energy function of RBM

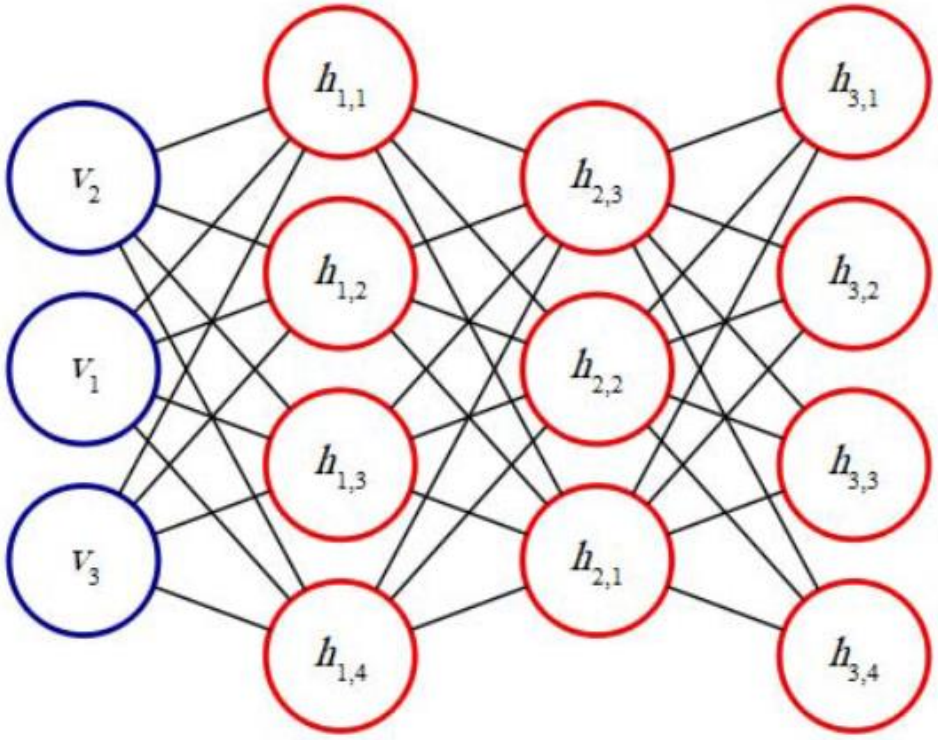
$$E(v, h) = -v^T Wh - a^T h - b^T v$$

$$p(v; \theta) = \frac{1}{Z_\theta} \exp\big(-E(v, h)\big)$$

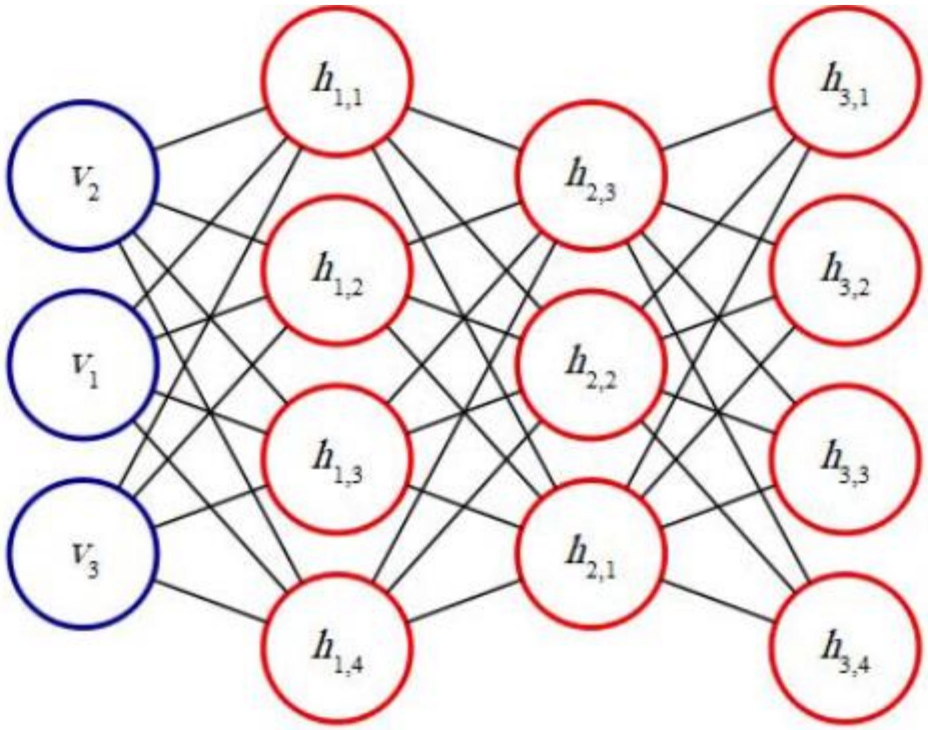


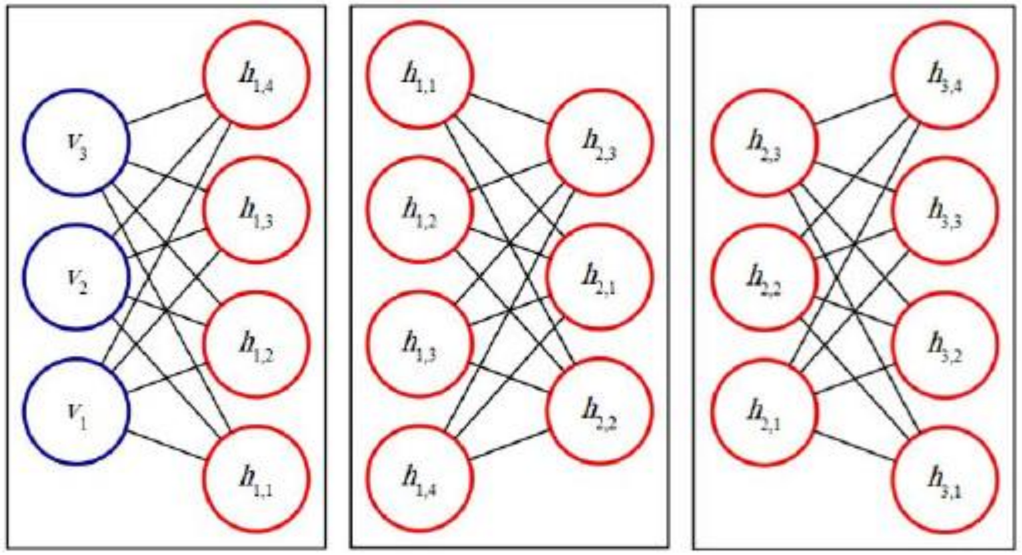
DEEP BOLTZMANN MACHINE (DBM)

- A multi-layered configuration of RBMs



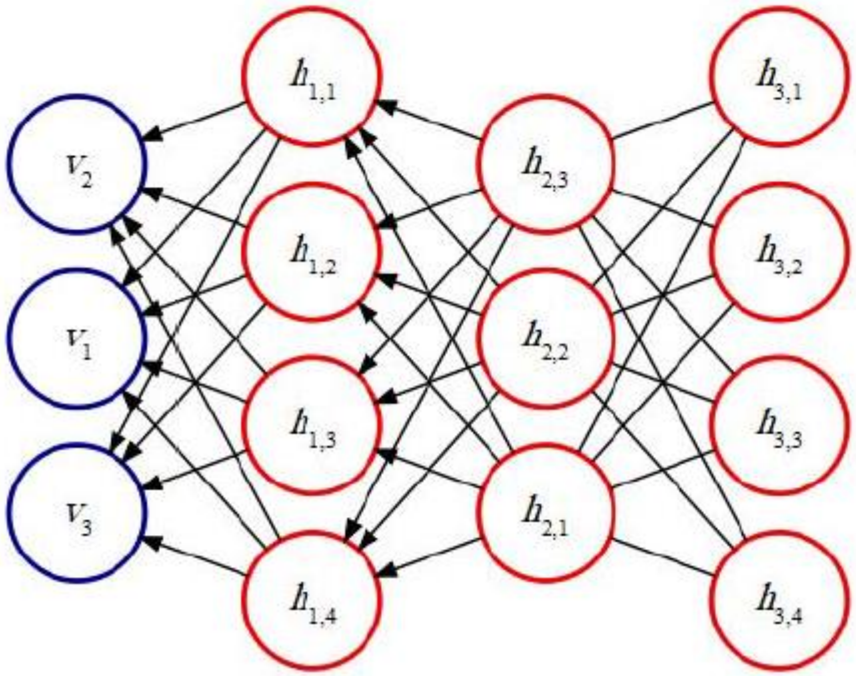
LEARNING IN DBM





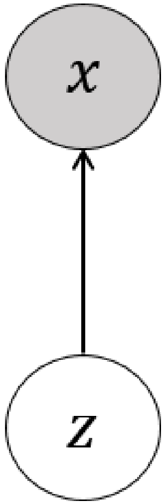
DEEP BELIEF NETWORK (DBN)

- Hybrid probabilistic graphical models



CONTINUES LATENT VARIABLE MODEL

- There would be a latent mechanism that is responsible for variations behind the data

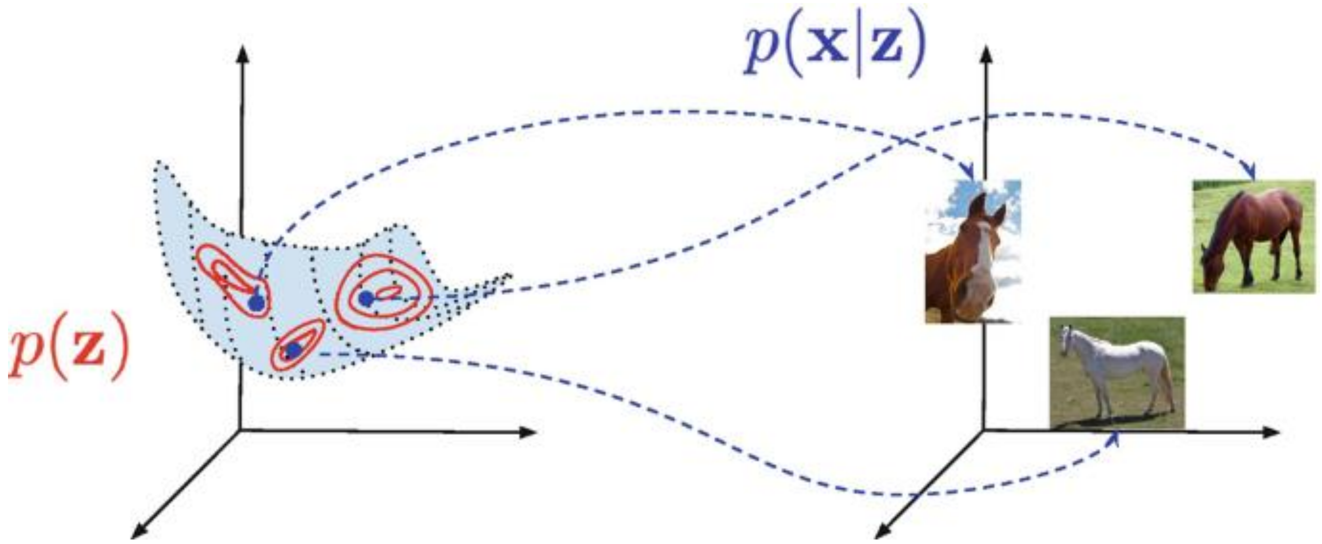


$$p_\theta(x,z) = p_\theta(z)p_\theta(x|z) \rightarrow p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$$

What forms the prior distribution and conditional distribution can take?

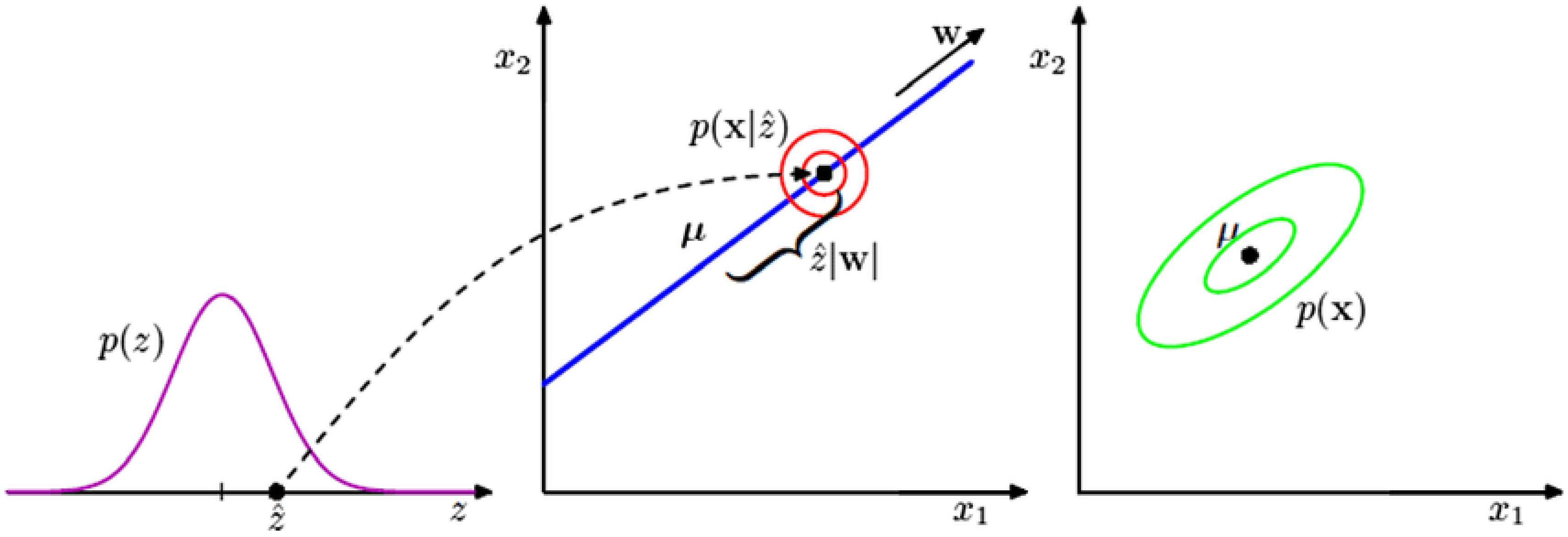
GAUSSIAN PRIOR

- Flexible mapping applied to standard Gaussian can model <u>any</u> complex distribution.



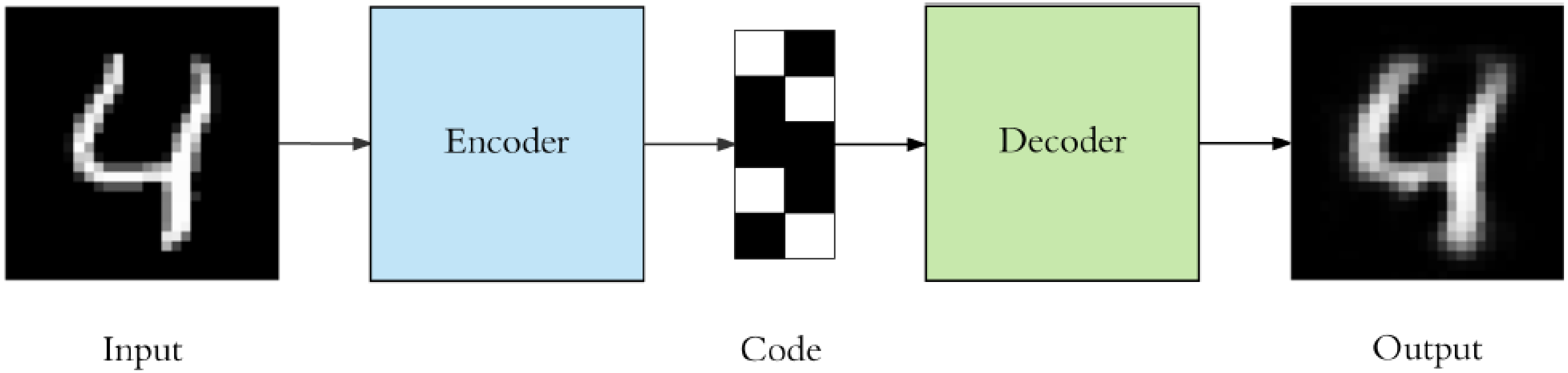
PROBABILISTIC PCA

- Linear Gaussian latent variable models
  - It has been shown that PCA is the MLE solution to probabilistic PCA



AUTO-ENCODER

- A simple neural networks with two layers, encoder and decoder

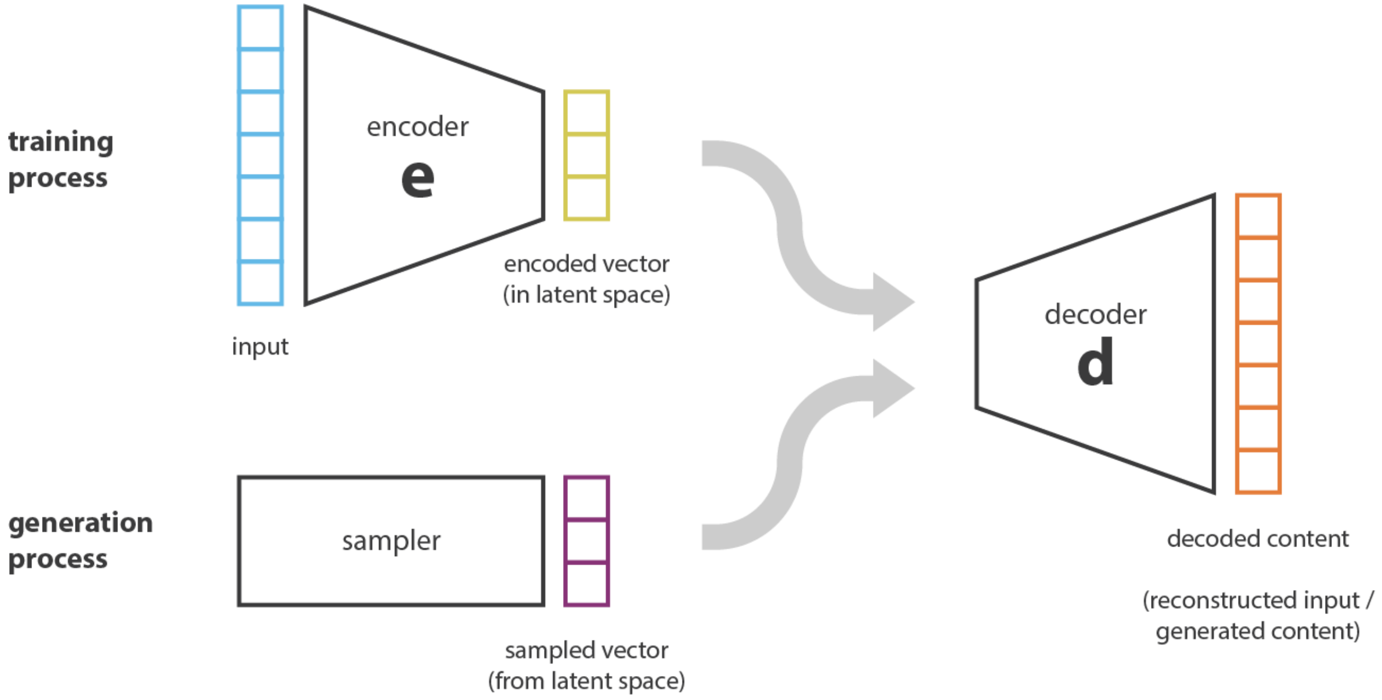


VARIATIONAL AUTO-ENCODERS

- The marginal distribution over a latent variable models can be approximated using Monte-Carlo simulation
  - However, it is not practical, since the samples generated from a standard Gaussian has been shown posses a low probability under conditional distribution $p(x|z)$, meaning we should generate infinite number of samples for generating one sample of x.

$$p_\theta(x) = \frac{1}{N} \sum_{z_k \sim p_\theta(z)} p_\theta(x|z_k)$$

RECOGNITION NETWORK



$$q_\phi(z|x) \sim N(\mu_{en}, \sigma_{en}^2) \rightarrow \begin{cases} \mu_{en} = NN_{en}^{\mu}(x) \\ \log \sigma_{en}^2 = NN_{en}^{\sigma}(x) \end{cases}$$

$$p_\theta(x|z) \sim N\left(\mu_{de}, \sigma_{de}^2\right) \rightarrow \begin{cases} \mu_{de} = NN_{de}^{\mu}(z) \\ \log \sigma_{de}^2 = NN_{de}^{\sigma}(z) \end{cases}$$

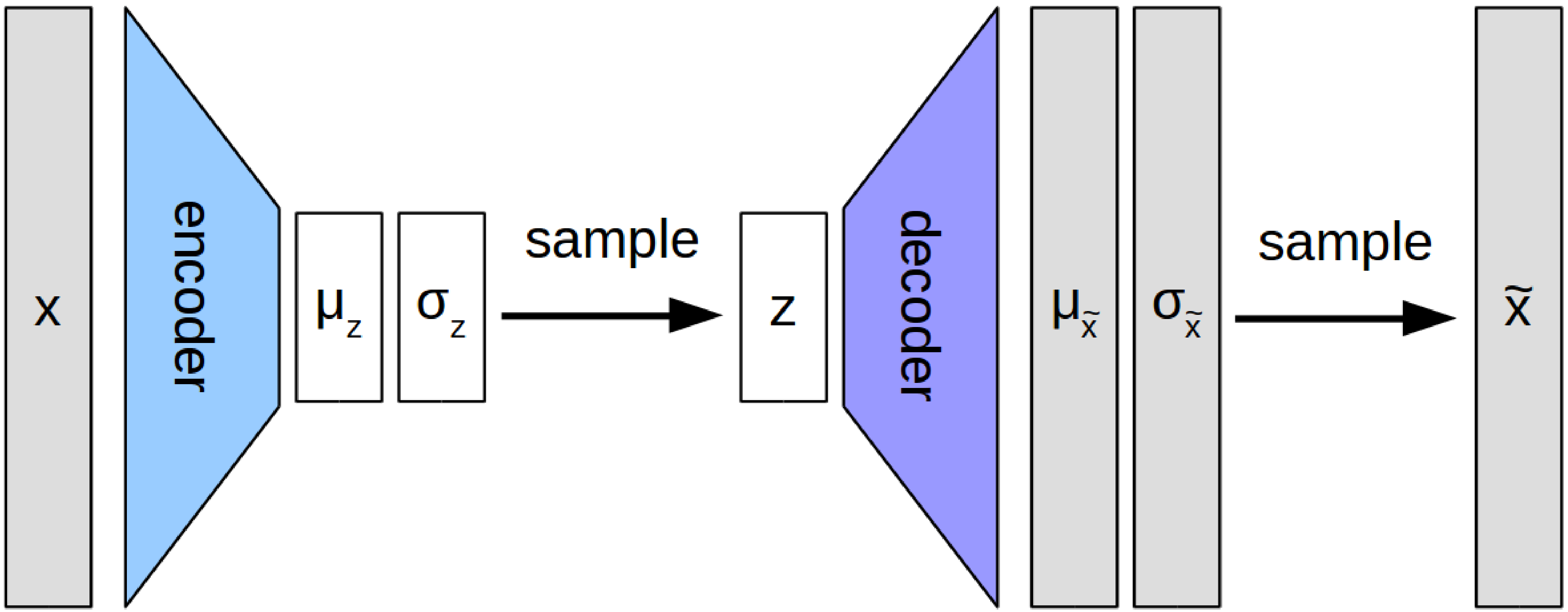
VARIATIONAL EM

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

$$\log p_\theta(x) = \log \int p_\theta(x,z)\,dz \to \log \int \frac{p_\theta(x,z)}{q_\phi(z|x)} q_\phi(z|x)dz \geq \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz = F(\theta,\phi)$$

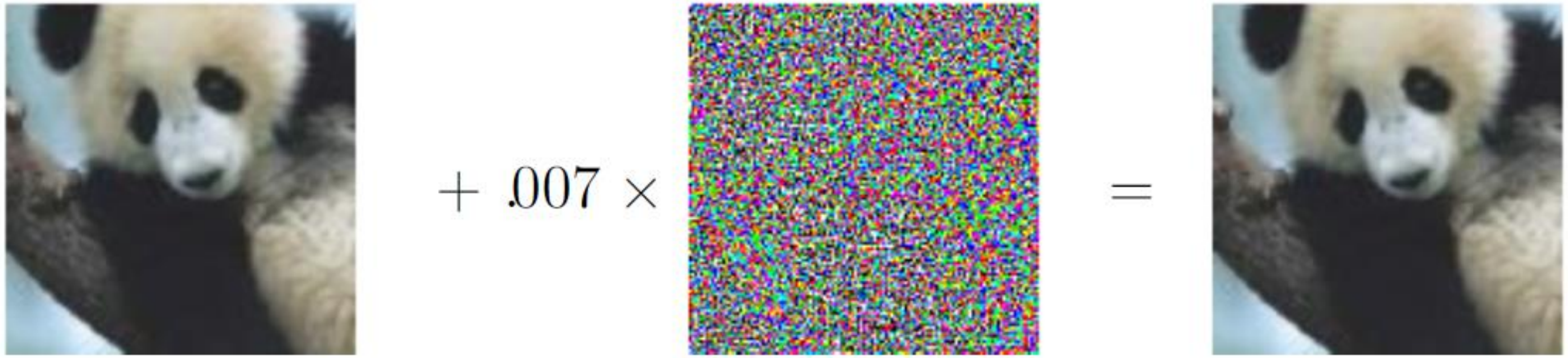
$$\theta^*,\phi^* = \arg\max_{\theta,\phi} F(\theta,\phi) \to \begin{cases} \phi_{k+1} = \arg\max_{\phi} F(\theta_k,\phi), & \mathrm{E-Step} \\ \theta_{k+1} = \arg\max_{\theta} F(\theta,\phi_{k+1}), & \mathrm{M-Step} \end{cases}$$

VAE STRUCTURE



ADVERSARIAL MACHINE LEARNING

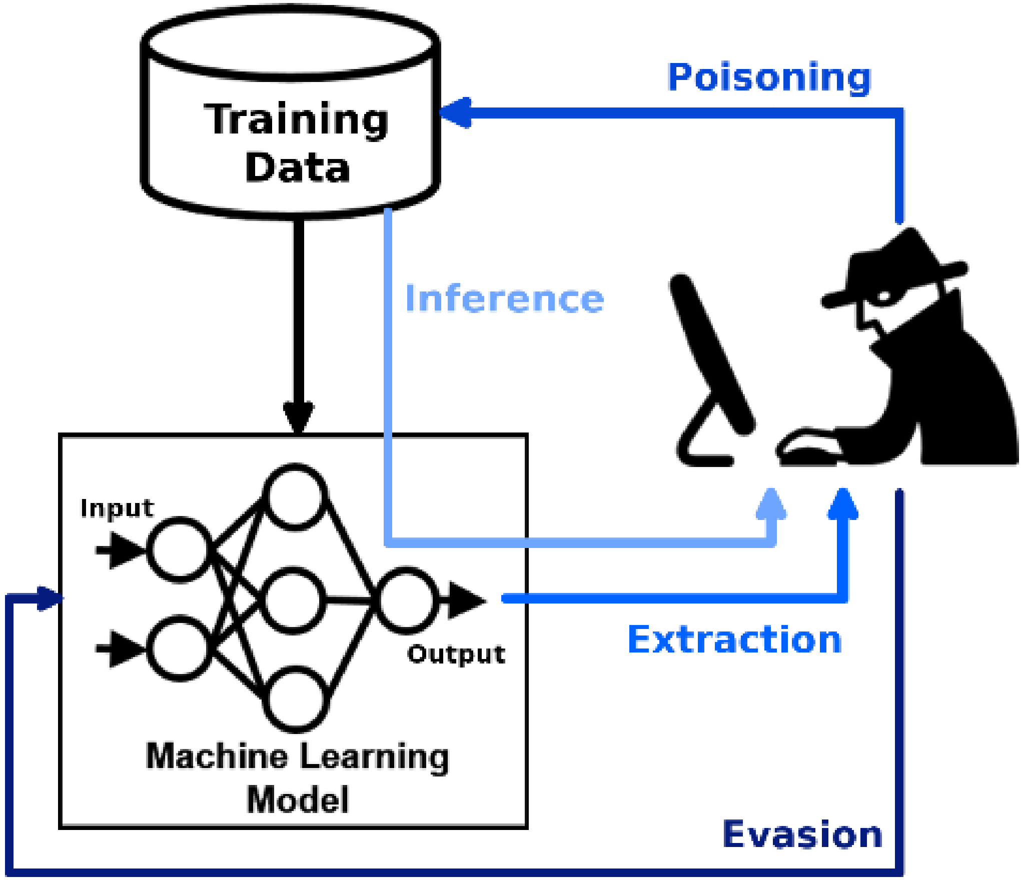
- Do really deep learning models perform tasks as performant as human?
  - Search for examples which cannot be misclassified by humans but can be misclassified by model -> adversarial examples



Panda

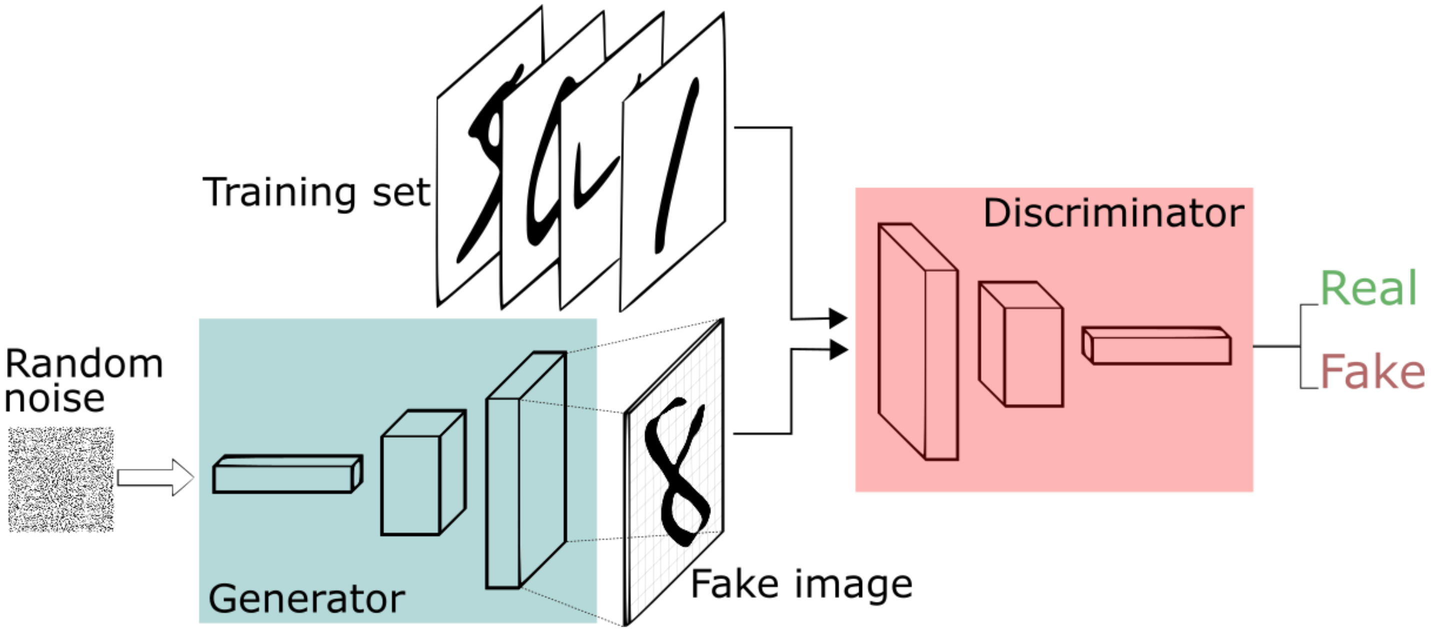
Gibbon

MACHINE LEARNING SECURITY



GENERATIVE ADVERSARIAL NETS (GAN)

- Generative adversarial net is the first model which is trained in an opposite direction of the dominant paradigm.



$$\theta^* = \min_G \max_D V(D,G) = E_{x\sim p_D}[\log D(x)] + E_{z\sim p_z}\left[1 - \log D\big(G(z)\big)\right]$$

GAN – IMPLEMENTATION

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

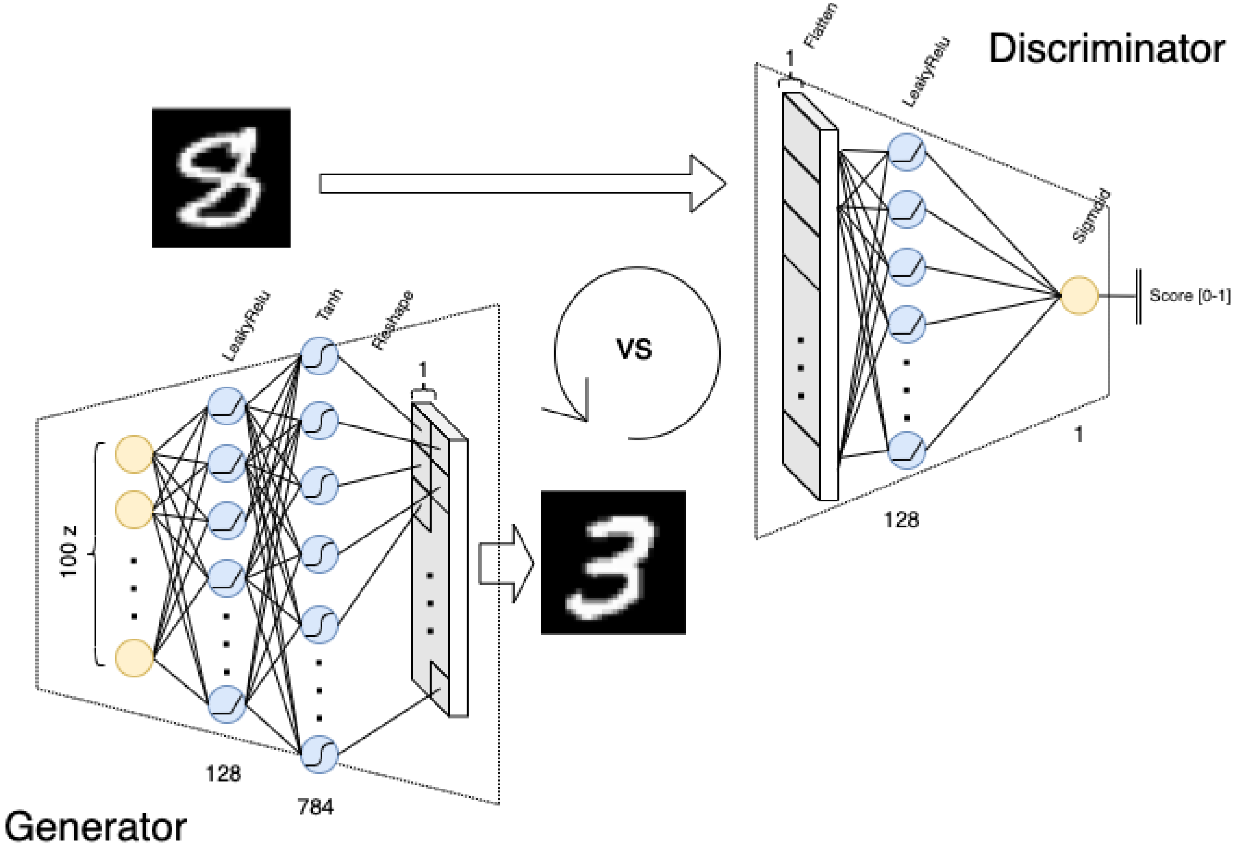
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

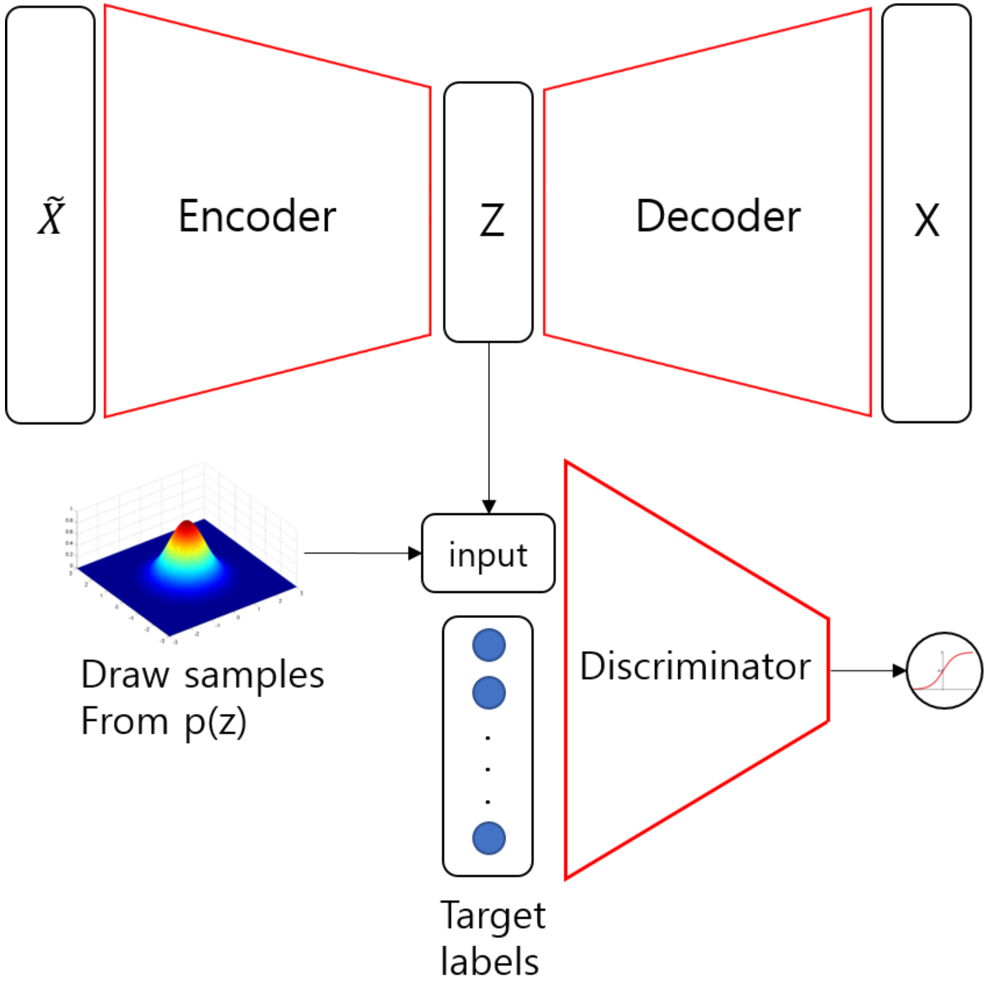
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

MODE COLLAPSE

- It is likely that generator produce samples belonging to specific mode rather than the entire distribution.



ADVERSARIAL AE



CONDITIONAL GAN

- One way for mitigating the mode collapse problem is to use class information

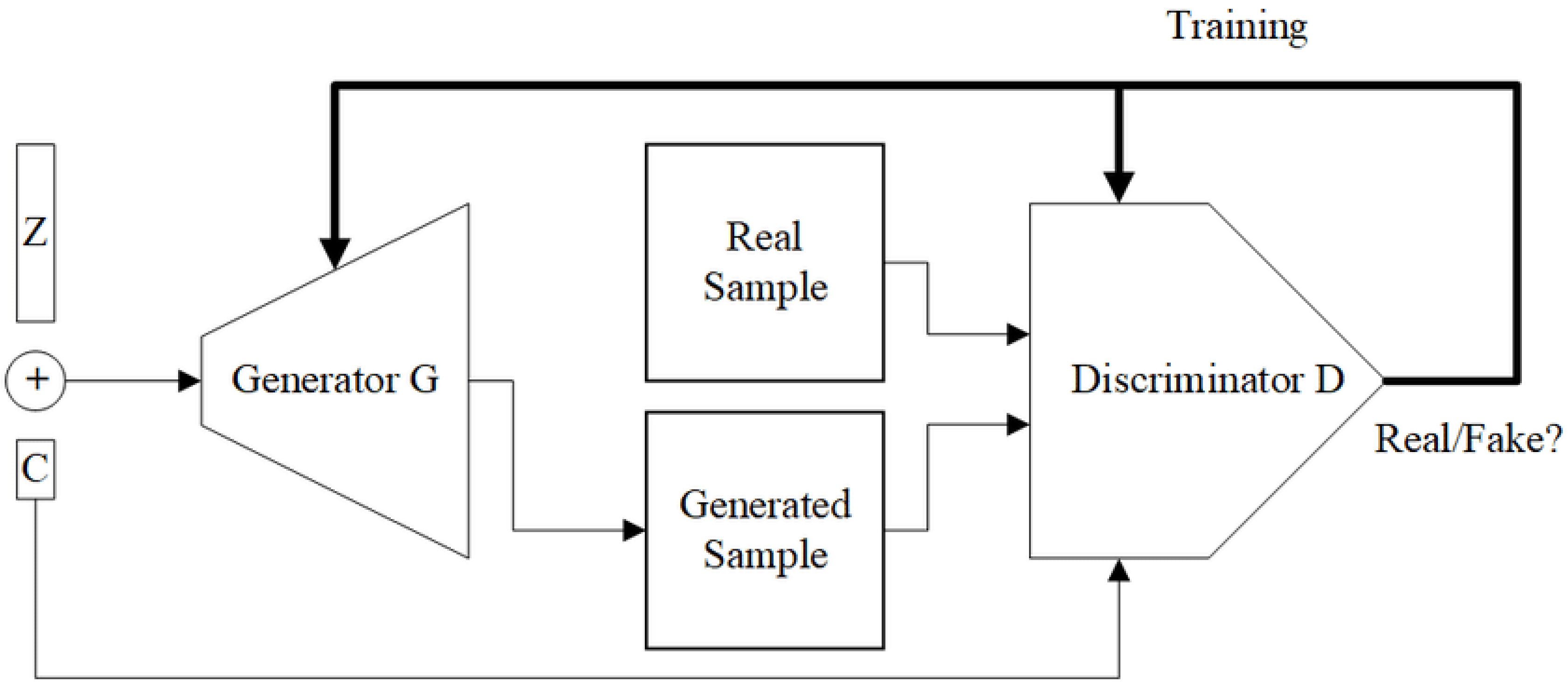
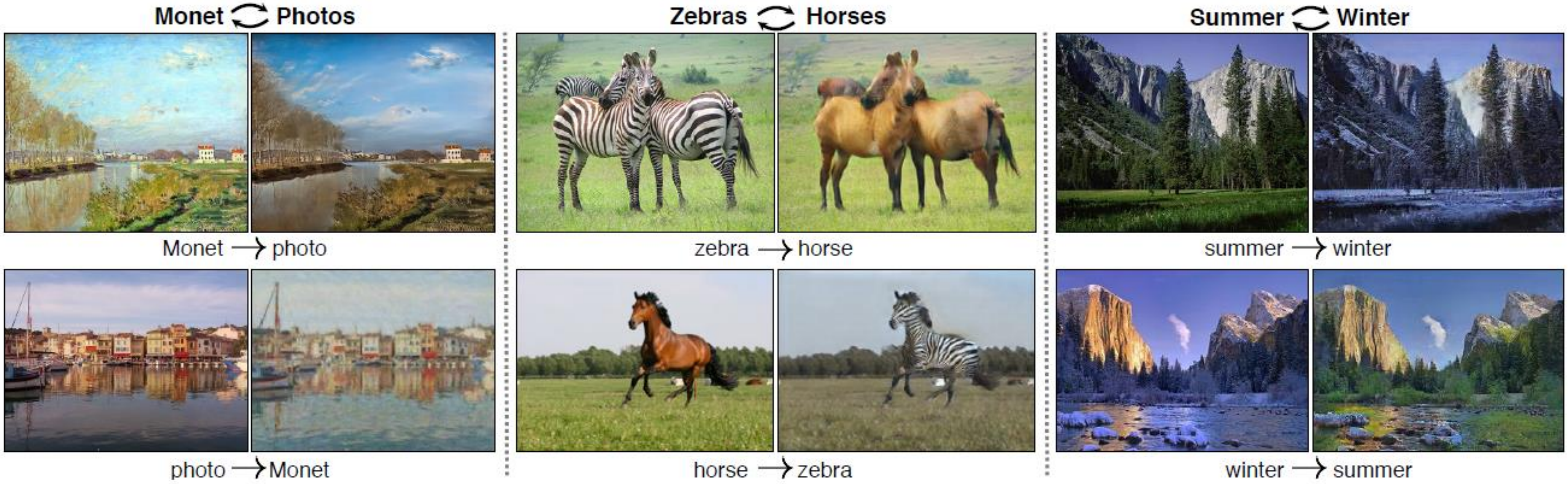
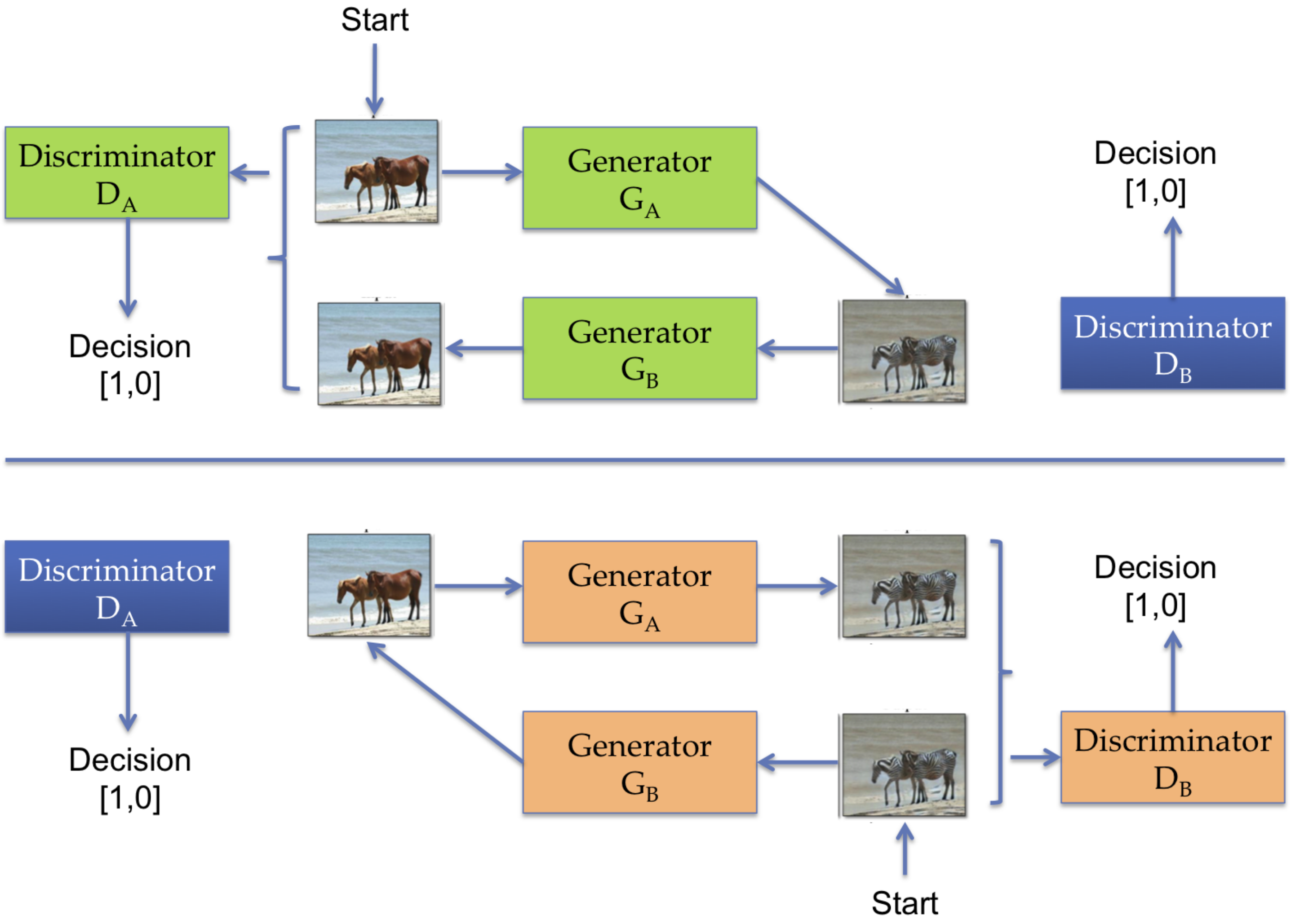


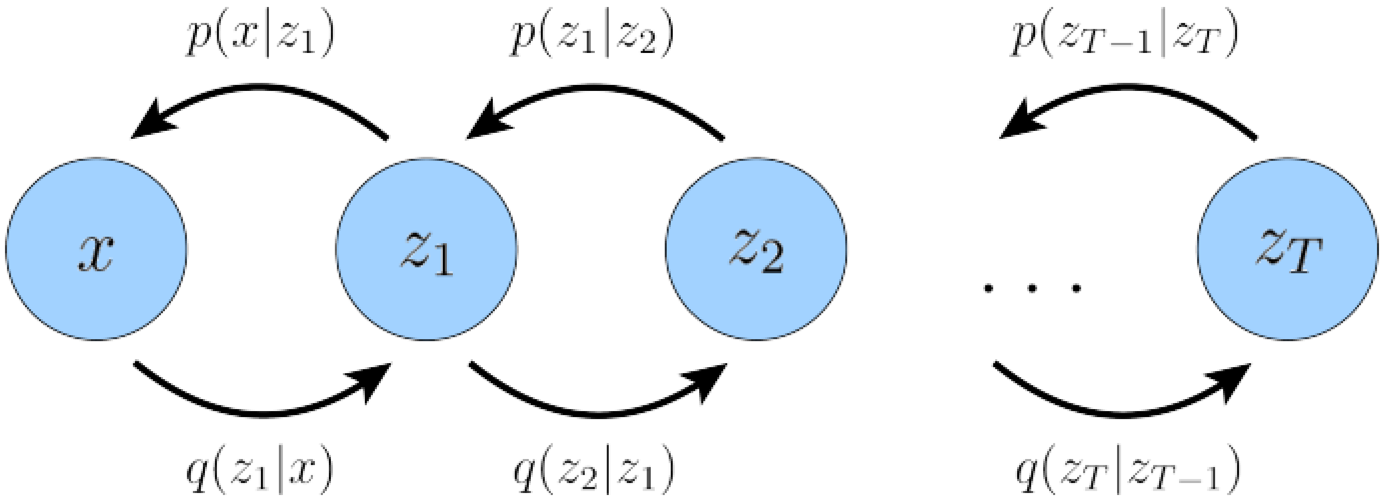
IMAGE-TO-IMAGE TRANSLATION



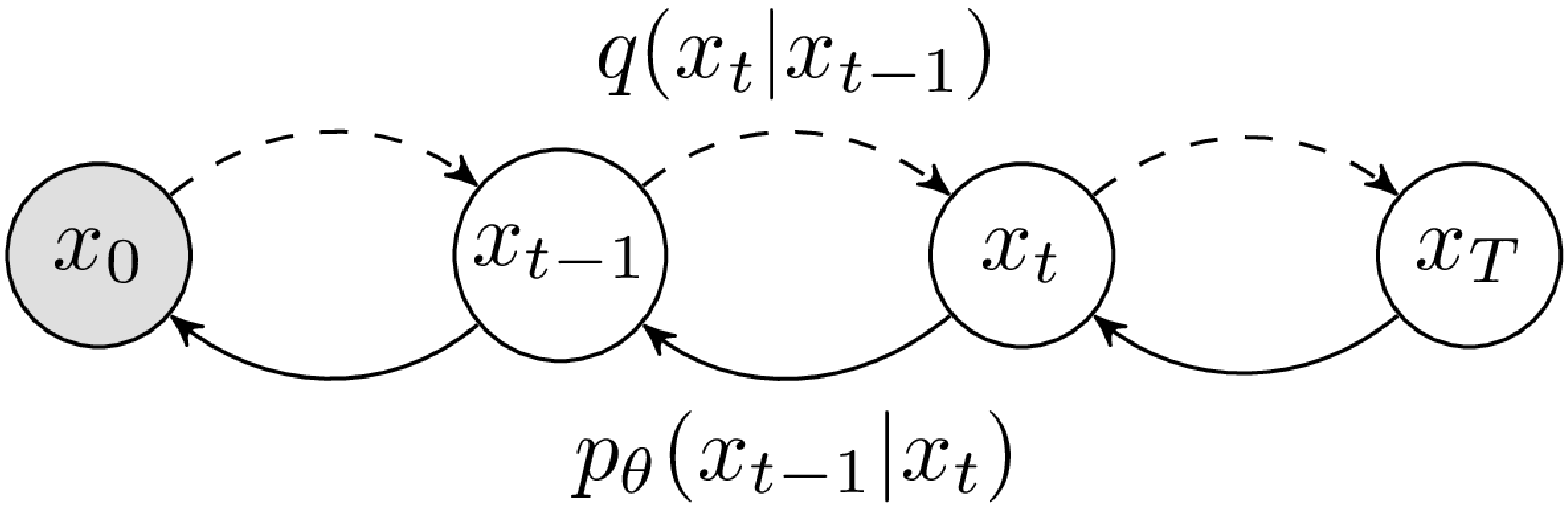
CYCLE GAN


Start
Discriminator
D_A
Generator
G_A
Decision
[1,0]
Decision
[1,0]
Generator
G_B
Discriminator
D_B
Discriminator
D_A
Generator
G_A
Decision
[1,0]
Decision
[1,0]
Generator
G_B
Discriminator
D_B
Start

HIERARCHICAL VAE

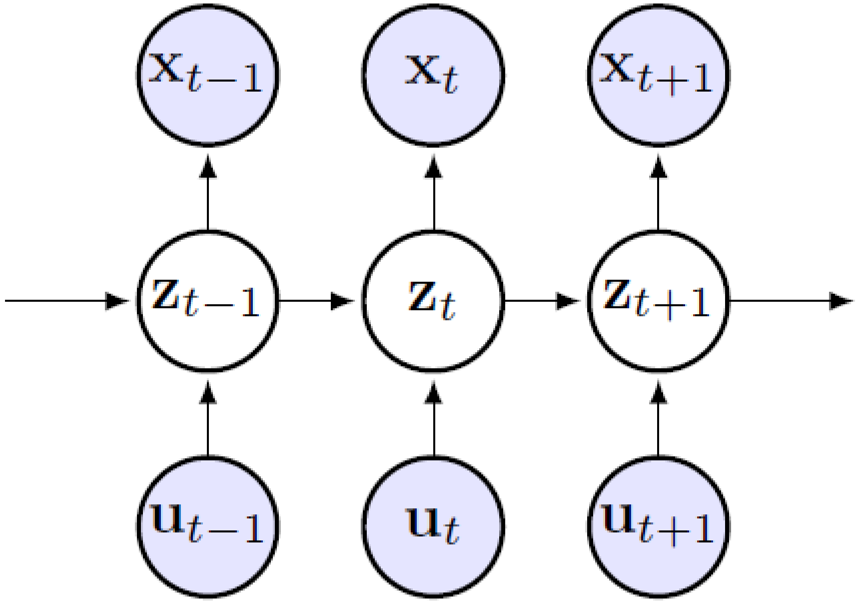


DIFFUSION MODELS



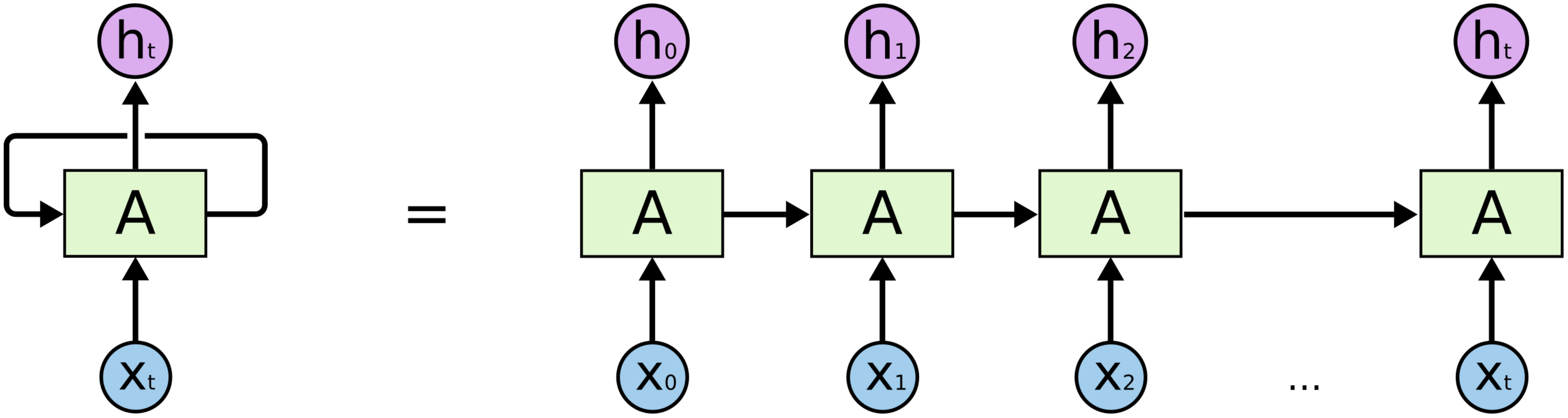
AUTOREGRESSIVE MODELS

- What we means of sequential data modeling?
  - Given a sequence of data $y_{1:T}$, we are wiling to model $P(y_{1:T})$.

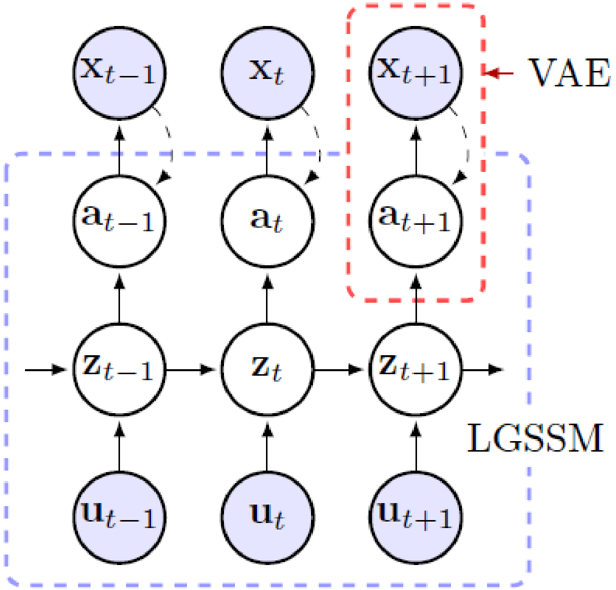


RECURRENT NEURAL NETWORKS

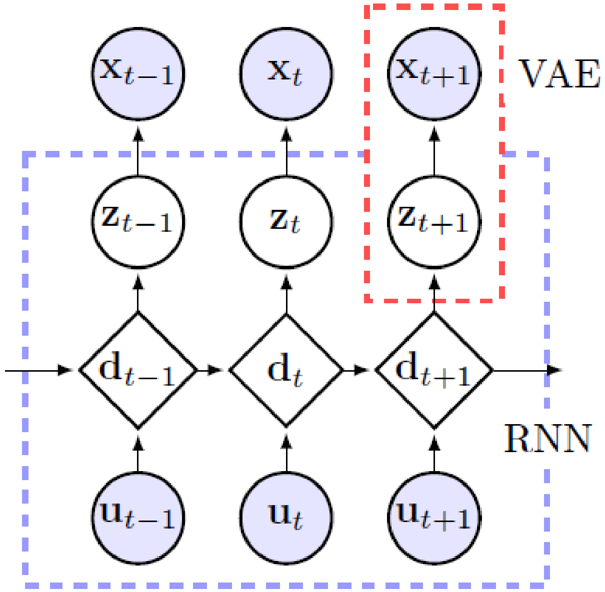
- The transition model is a deterministic mapping while the output model follows a Gaussian distribution -> incapable of capturing the variation behind data



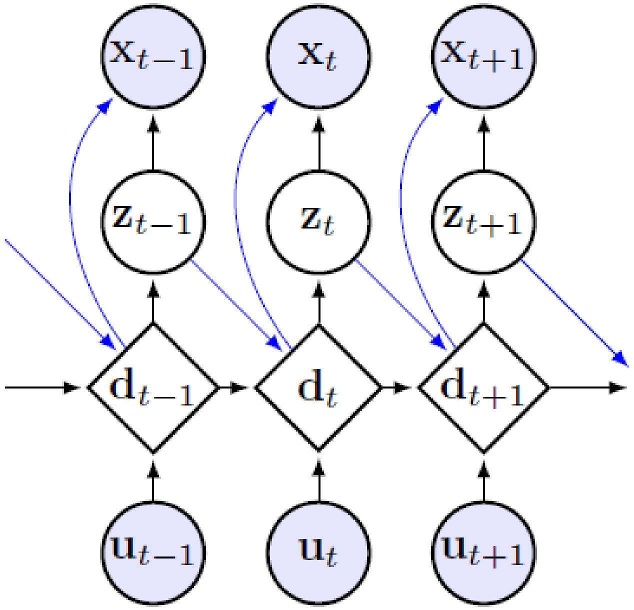
DYNAMICAL VAE



<u>Kalman VAE</u>

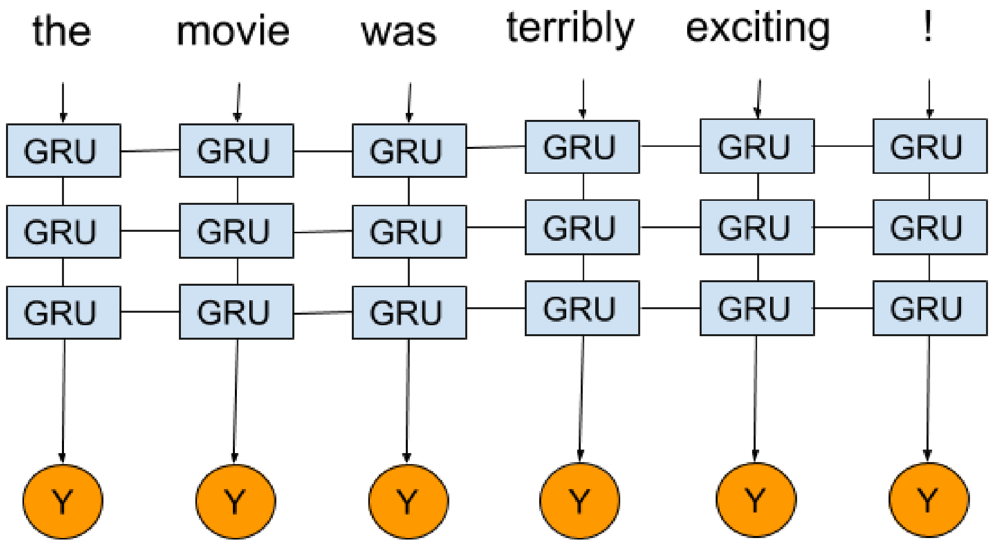


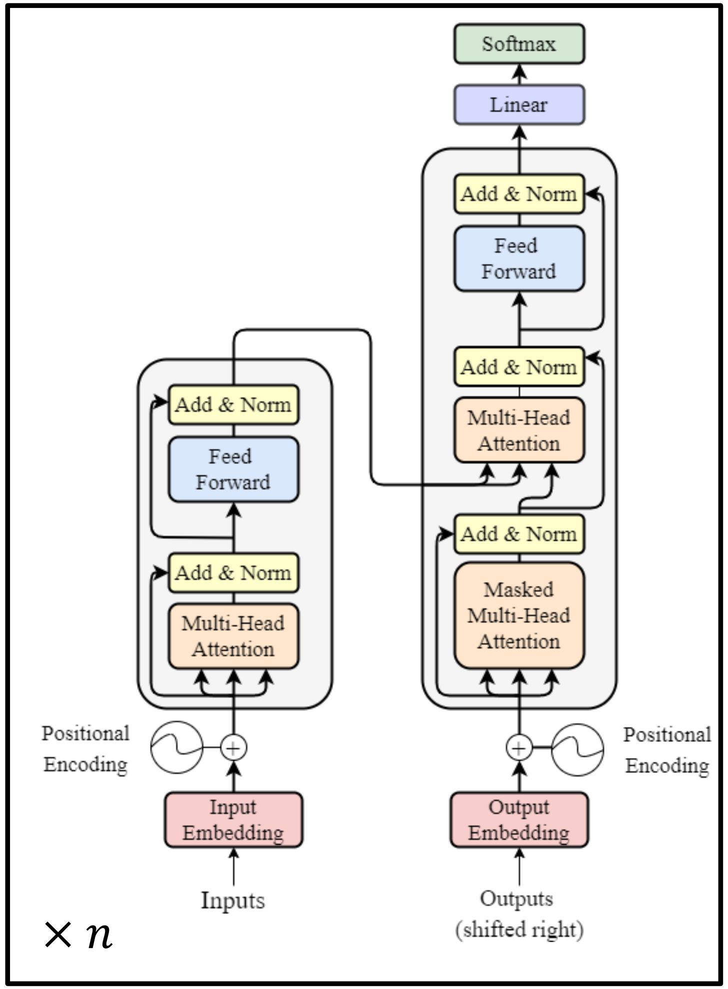
<u>VAE-RNN</u>



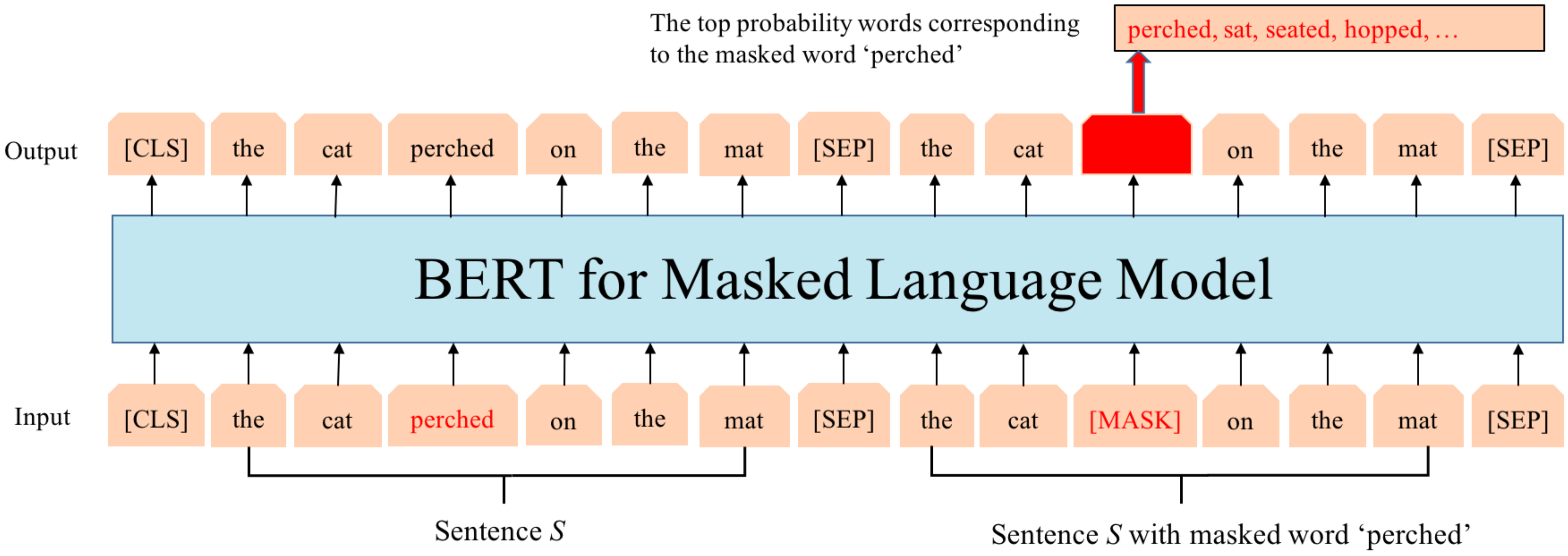
<u>VRNN</u>

DEEP AUTOREGRESSIVE MODELS

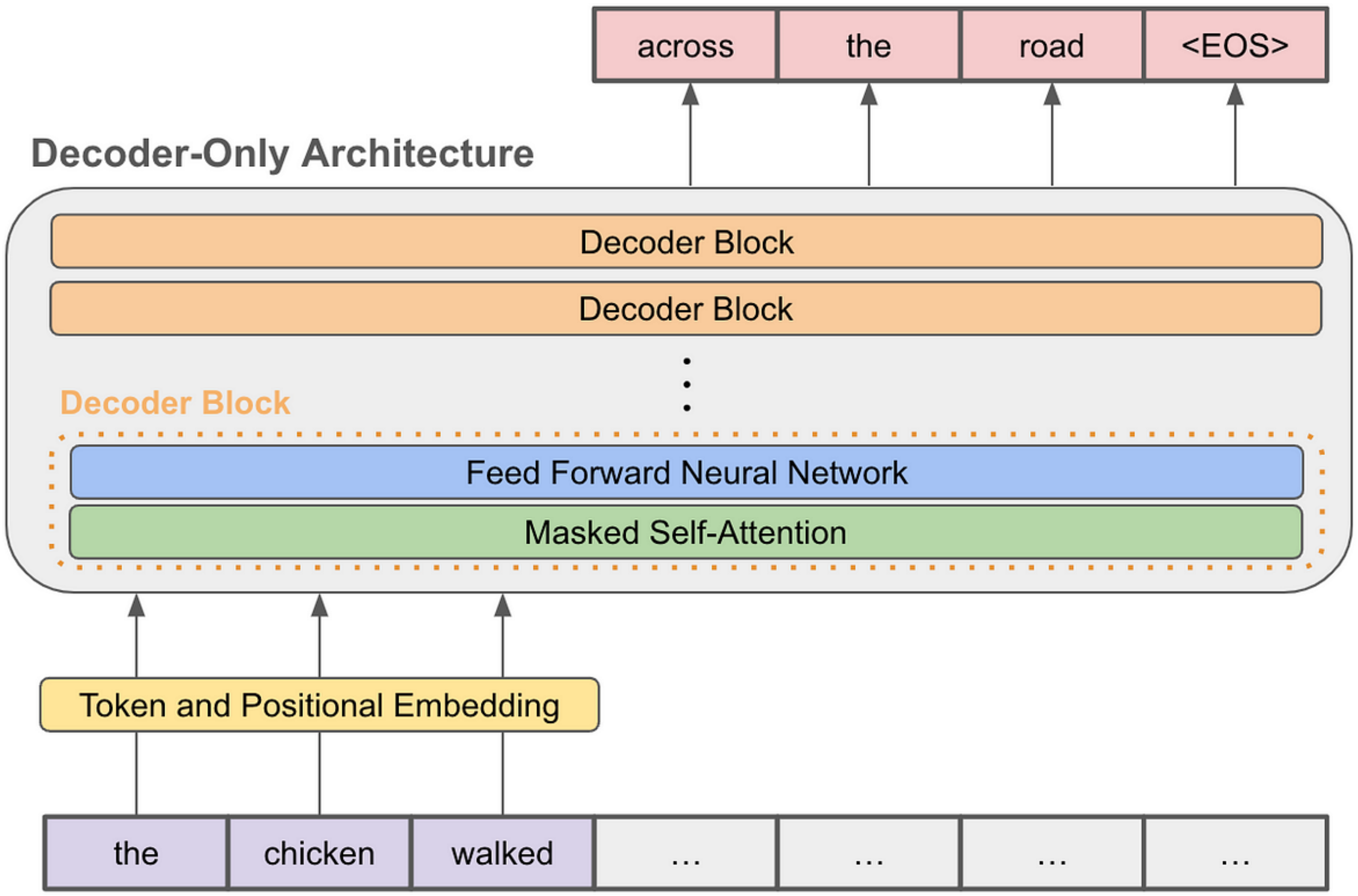




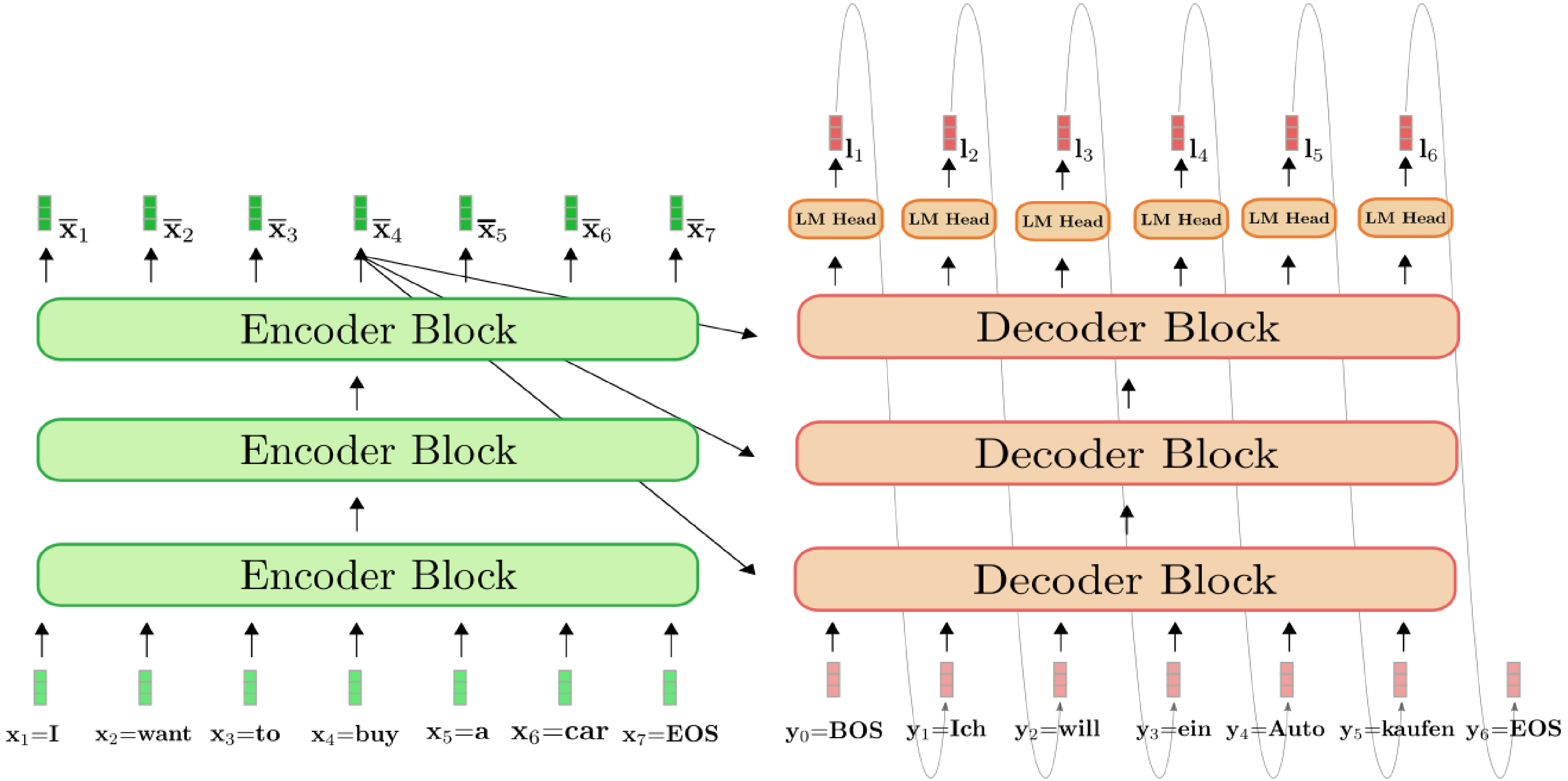
ENCODER-ONLY ARCHITECTURE



DECODER-ONLY ARCHITECTURE



ENCODER-DECODER ARCHITECTURE



NORMALIZING FLOW NETWORK