DSC291: Machine Learning with Few Labels

Enhancing Large Language Models: Overview

Zhiting Hu

Lecture 11, April 24, 2024

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

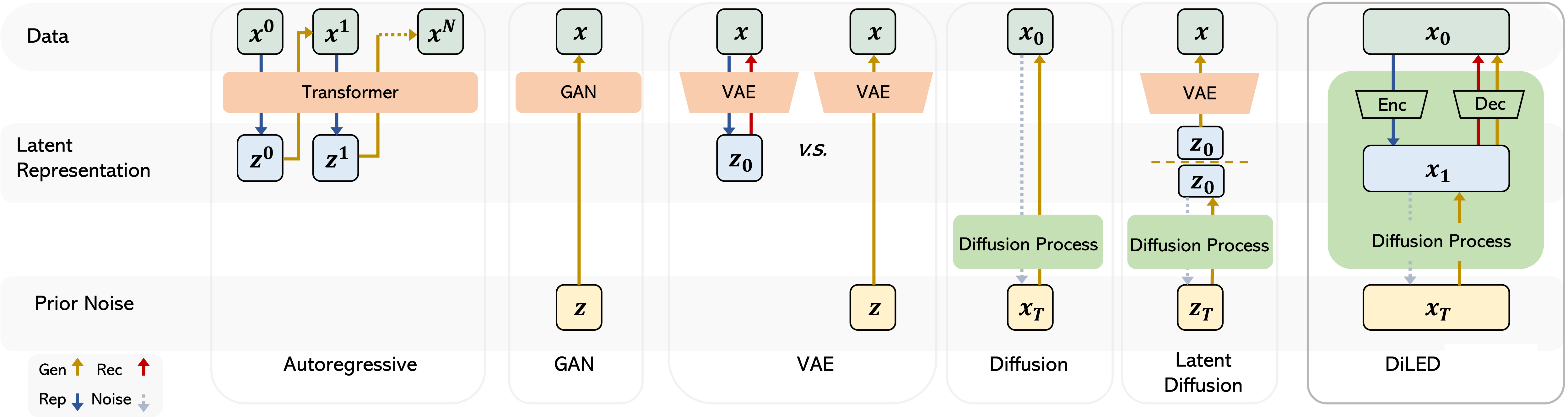
Discussion

- **No Free Lunch (NFL) theorem** (suggested reading of Lecture#10):
    - No single machine learning algorithm is universally the best-performing algorithm for all problems

- Do generalist models (LLMs) violate this theorem?
- Does "the Bitter Lesson" contradict with this theorem?
    - (suggested reading of Lecture#6)

Latent-space Reasoning (Recap)

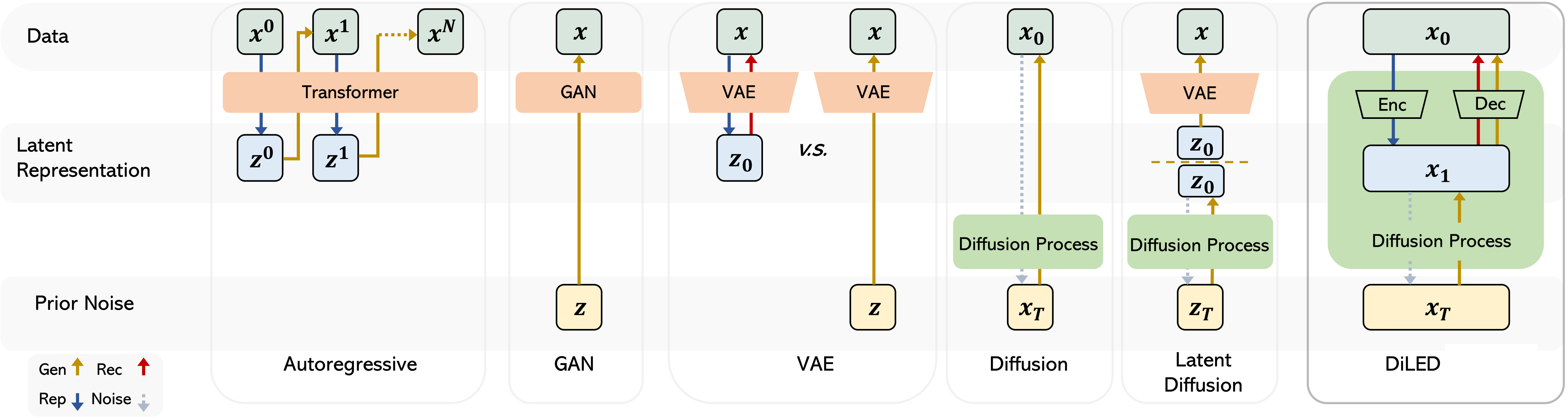
- But how to learn a good latent space in the first place?



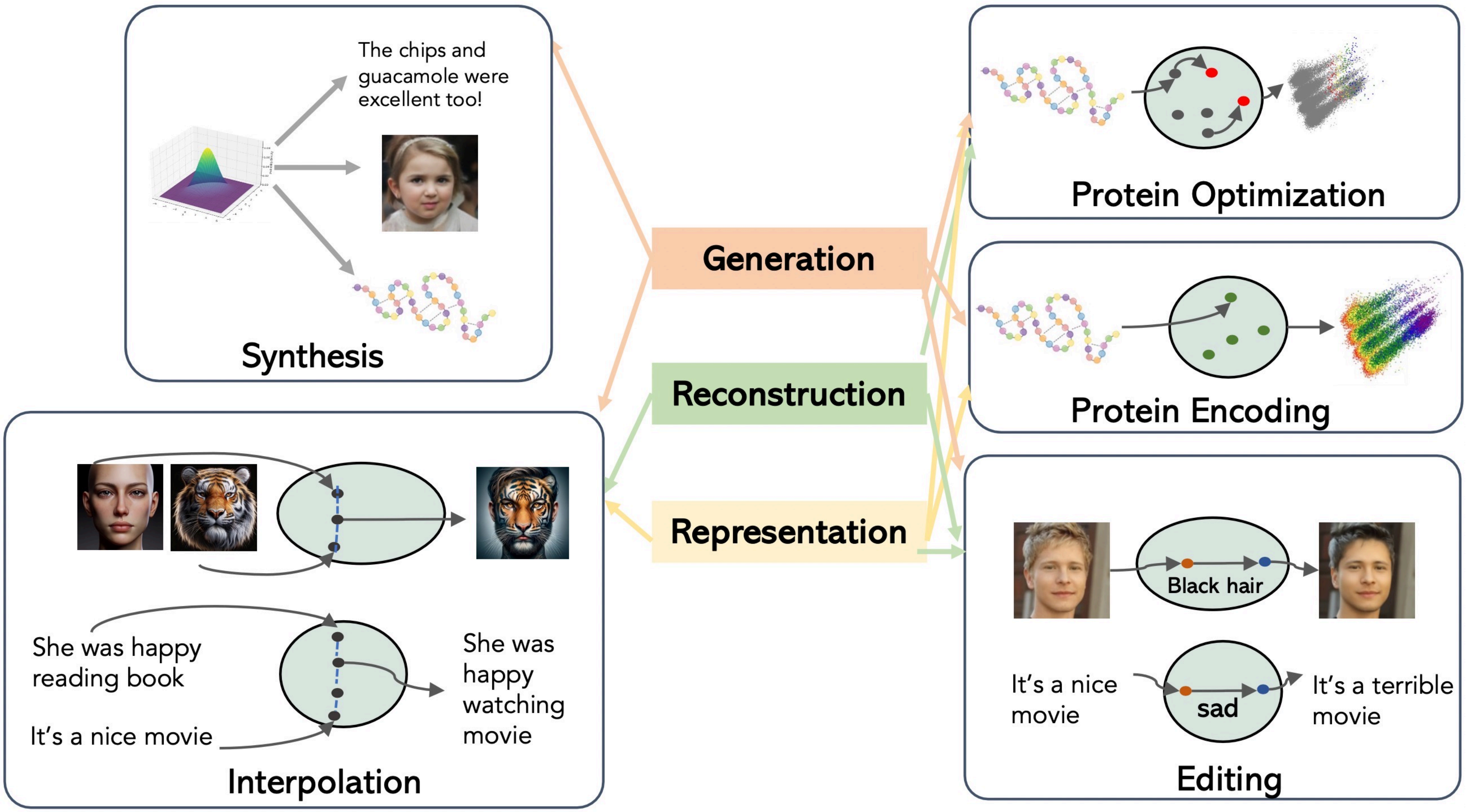
[Liu et al., 2024] Generating, Reconstructing, and Representing Discrete and Continuous Data: Generalized Diffusion with Learnable Encoding-Decoding

Latent-space Reasoning

- But how to learn a good latent space in the first place?
  - Compact and well-structured **representation** of the world, enabling realistic **generation** and consistent **reconstruction**



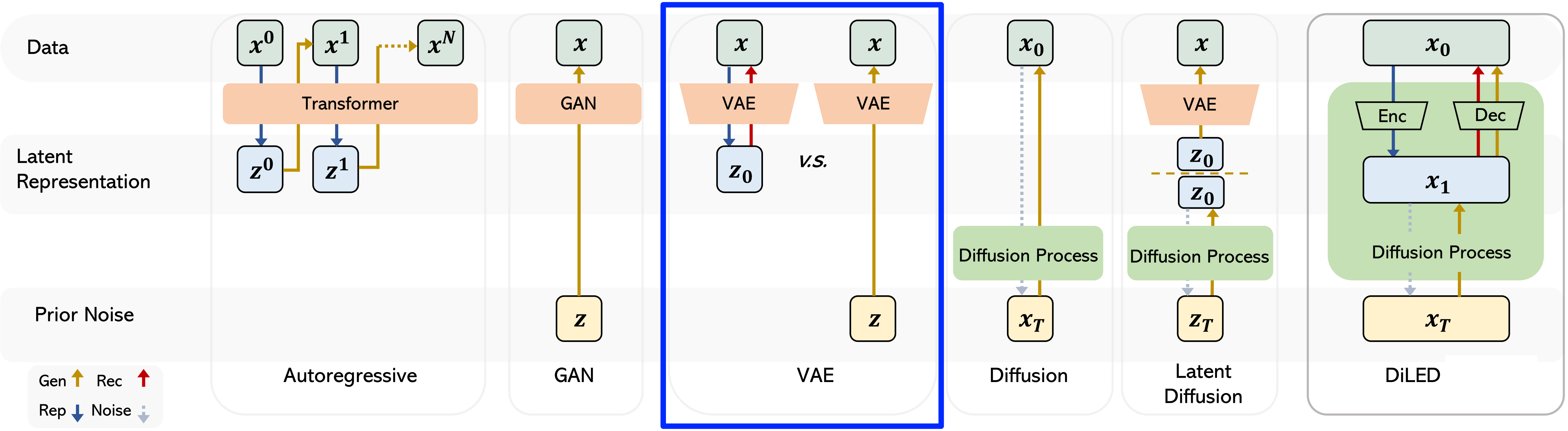
[Liu et al., 2024] Generating, Reconstructing, and Representing Discrete and Continuous Data: Generalized Diffusion with Learnable Encoding-Decoding



[Liu et al., 2024] Generating, Reconstructing, and Representing Discrete and Continuous Data: Generalized Diffusion with Learnable Encoding-Decoding

Latent-space Reasoning

- But how to learn a good latent space in the first place?
  - Compact and well-structured **representation** of the world, enabling realistic **generation** and consistent **reconstruction**



Variational Autoencoders (VAEs)

[Liu et al., 2024] Generating, Reconstructing, and Representing Discrete and Continuous Data: Generalized Diffusion with Learnable Encoding-Decoding

Supervised Learning, Unsupervised Learning

Probabilistic Models: Why Probability?

- The world is a very uncertain place
  - “What will the weather be like today?”
  - “Will I like this movie?”
- We often can’t prove something is true, but we can still ask how likely different outcomes are or ask for the most likely explanations
- Predictions need to have associated confidence
  - Confidence -> probability
- Not all machine learning models are probabilistic
  - … but most of them have probabilistic interpretations

[CS60020, Bhattacharya; CSC2515, Wang]

Notations

- A random variable $\boldsymbol{x}$ represents outcomes or states of the world.
  - We write $p(\boldsymbol{x}_0)$ to mean Probability($\boldsymbol{x} = \boldsymbol{x}_0$)
- Sample space: the space of all possible outcomes (may be discrete, continuous, or mixed)
- $p(\boldsymbol{x})$ is the probability mass (density) function
  - Assigns a number to each point in sample space
  - Non-negative, sums (integrates) to 1
  - Intuitively: how often does $\boldsymbol{x}$ occur, how much do we believe in $\boldsymbol{x}$.

Notations

- Joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$
- Conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$
  - $p(\boldsymbol{y}|\boldsymbol{x}) = \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})}$
- Expectation:

$$\mathbb{E}[f(\boldsymbol{x})] = \sum_{\boldsymbol{x}} f(\boldsymbol{x})\, p(\boldsymbol{x})$$

or

$$\mathbb{E}[f(\boldsymbol{x})] = \int_{\boldsymbol{x}} f(\boldsymbol{x}) p(\boldsymbol{x}) dx$$

Rules of Probability

- Sum rule

$$p(x) = \sum_y p(x, y)$$ (Marginalize out y)

$$p(x_1) = \sum_{x_2} \sum_{x_3} \cdots \sum_{x_N} p(x_1, x_2, \ldots, x_N)$$

- Product/chain rule

$$p(x, y) = p(y \mid x) p(x)$$

$$p(x_1, \ldots, x_N) = p(x_1) p(x_2 \mid x_1) \ldots p(x_N \mid x_1, \ldots, x_{N-1})$$

Bayes' Rule

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})}{p(\boldsymbol{x})}$$

- This gives us a way of "reversing" conditional probabilities
- We call $p(\boldsymbol{y})$ the "prior", and $p(\boldsymbol{y}|\boldsymbol{x})$ the "posterior"
- Ex: Bayes' Rule in machine learning:
  - $\mathcal{D}$: data (evidence)
  - $\boldsymbol{\theta}$: unknown quantities, such as model parameters, predictions

Likelihood: How likely is the observed data under the particular unknown quantities $\boldsymbol{\theta}$

Posterior belief on the unknown quantities you see data $\mathcal{D}$

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}$$

Prior belief on the unknown quantities **before** you see data $\mathcal{D}$

Independence

- Two random variables are said to be **independent** iff their joint distribution factors

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$$

- Two random variables are **conditionally independent** given a third if they are independent after conditioning on the third

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{z})$$

Entropy

- Shannon entropy $H(p) = -\sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(\boldsymbol{x})$
  - The average level of "information", "surprise", or "uncertainty" inherent to the variable $\boldsymbol{x}$'s possible outcomes

KL Divergence

- Kullback-Leibler (KL) divergence: measures the closeness of two distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$

$$\mathrm{KL}(q(\boldsymbol{x}) \,||\, p(\boldsymbol{x})) = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}$$

  - a.k.a. Relative entropy
  - KL >= 0 (Jensen's inequality)
  - Intuitively:
    - If q is high and p is high, then we are happy (i.e. low KL divergence)
    - If q is high and p is low then we pay a price (i.e. high KL divergence).
    - If q is low then we don't care (i.e. also low KL divergence, regardless of p)
  - not a true "distance":
    - not commutative (symmetric) $\mathrm{KL(p||q)} \,!= \mathrm{KL(q||p)}$
    - doesn't satisfy triangle inequality

Supervised Learning

- Model to be learned $p_\theta(\boldsymbol{x})$
- Observe **full** data $\mathcal{D} = \{\, \boldsymbol{x}_i \,\}_{i=1}^{N}$
  - e.g., $\boldsymbol{x}_i$ includes both input (e.g., image) and output (e.g., image label)
  - $\mathcal{D}$ defines an empirical data distribution $\tilde{p}(\boldsymbol{x})$
    - $\boldsymbol{x} \sim \mathcal{D} \quad \Leftrightarrow \quad \boldsymbol{x} \sim \tilde{p}(\boldsymbol{x})$

- Maximum Likelihood Estimation (MLE)
  - The most classical learning algorithm

$$\min_\theta - \mathbb{E}_{x\sim\tilde{p}(x)}\left[\quad \log p_\theta(\boldsymbol{x}) \quad \right]$$

- MLE is minimizing the KL divergence between the empirical data distribution and the model distribution

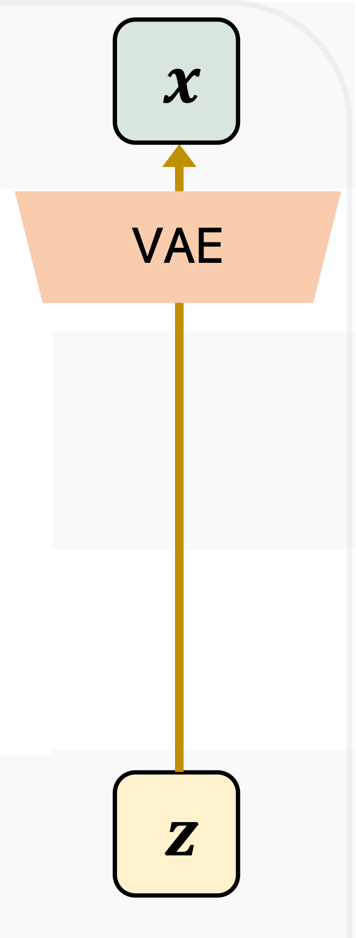
$$\mathrm{KL}(\tilde{p}(\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{x})) = -\mathbb{E}_{\tilde{p}(\boldsymbol{x})}[\, \log p_\theta(\boldsymbol{x}) \,] + H(\tilde{p}(\boldsymbol{x}))$$

$$\downarrow$$

Cross entropy

Unsupervised Learning

- Each data instance is partitioned into two parts:
  - observed variables $\boldsymbol{x}$
  - latent (unobserved) variables $\boldsymbol{z}$
- Want to learn a model $p_\theta(\boldsymbol{x}, \boldsymbol{z})$



[Content adapted from CMU 10-708]

Latent (unobserved) variables

- A variable can be unobserved (latent) because:
  - imaginary quantity: meant to provide some simplified and abstractive view of the data generation process
    - e.g., speech recognition models, mixture models, …

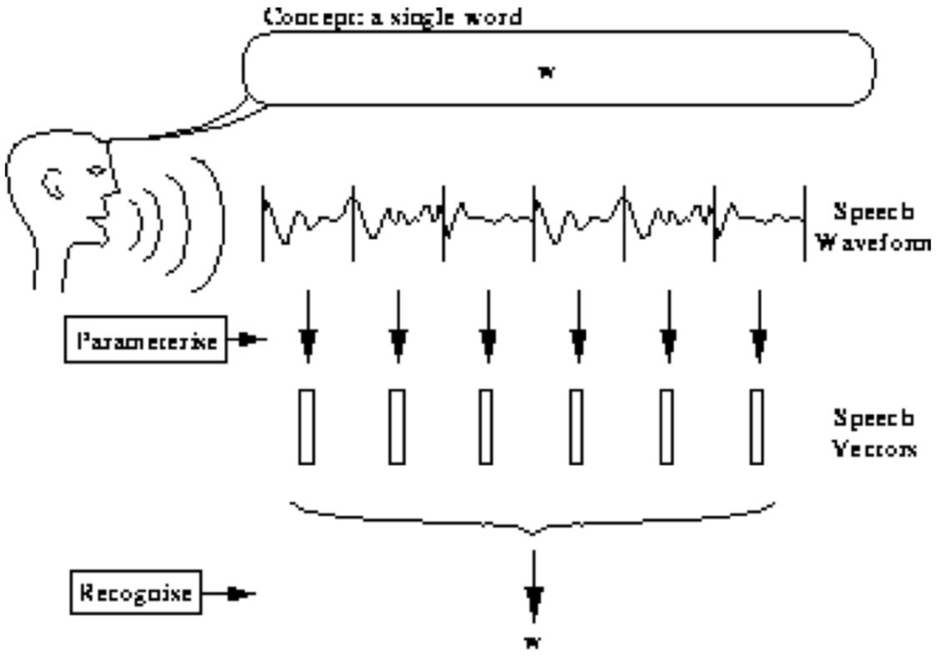
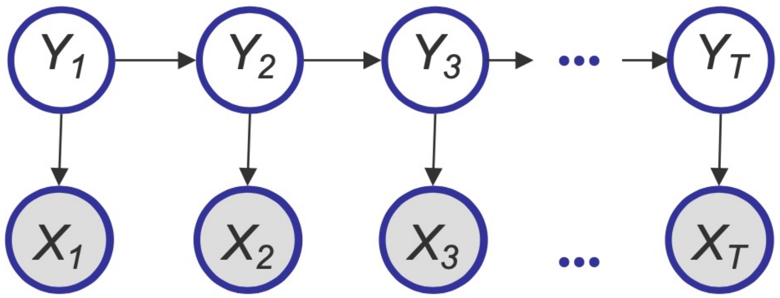
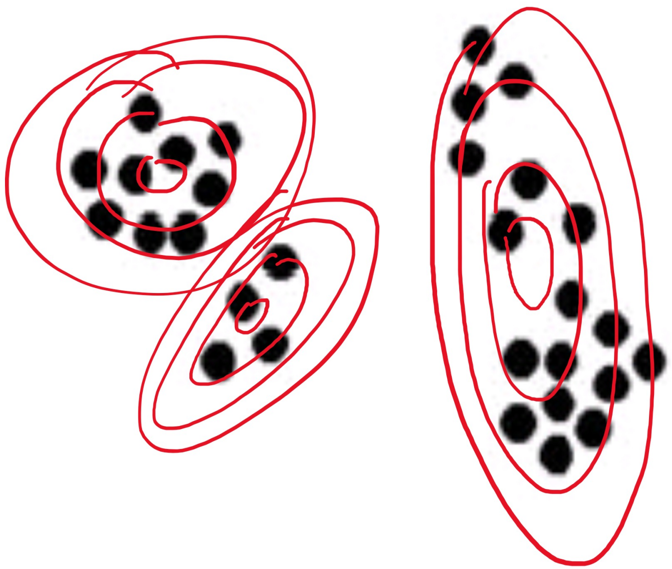


Fig. 1.2 Isolated Word Problem



Latent (unobserved) variables

- A variable can be unobserved (latent) because:
  - imaginary quantity: meant to provide some simplified and abstractive view of the data generation process
    - e.g., speech recognition models, mixture models, ...



Latent (unobserved) variables

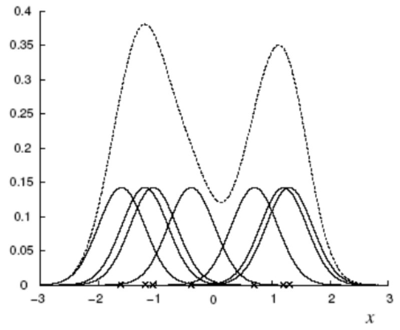
- A variable can be unobserved (latent) because:
  - imaginary quantity: meant to provide some simplified and abstractive view of the data generation process
    - e.g., speech recognition models, mixture models, ...
  - a real-world object (and/or phenomena), but difficult or impossible to measure
    - e.g., the temperature of a star, causes of a disease, evolutionary ancestors ...
  - a real-world object (and/or phenomena), but sometimes wasn't measured, because of faulty sensors, etc.
- Discrete latent variables can be used to partition/cluster data into sub-groups
- Continuous latent variables (factors) can be used for dimensionality reduction (e.g., factor analysis, etc.)

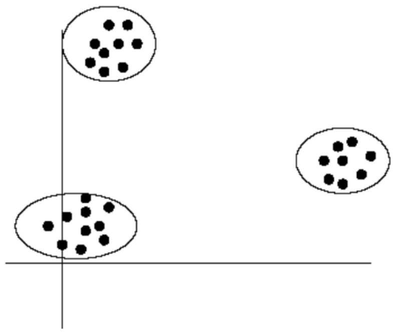
Example: Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components:

$$p(x_n|\mu,\Sigma) = \sum_k \pi_k N(x,|\,\mu_k,\Sigma_k)$$

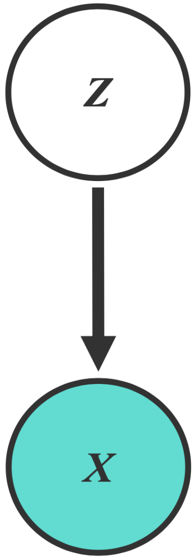
mixture proportion   mixture component





- This model can be used for unsupervised clustering.
  - This model (fit by AutoClass) has been used to discover new kinds of stars in astronomical data, etc.

Example: Gaussian Mixture Models (GMMs)



- Consider a mixture of K Gaussian components:
  - Z is a latent class indicator vector:

$$p(z_n) = \mathrm{multi}(z_n : \pi) = \prod_k (\pi_k)^{z_n^k}$$

  - X is a conditional Gaussian variable with a class-specific mean/covariance

$$p(x_n \mid z_n^k = 1, \mu, \Sigma) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left\{ -\tfrac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) \right\}$$

  - The likelihood of a sample:

Parameters to be learned:

mixture component

mixture proportion

$$\begin{aligned} p(x_n | \mu, \Sigma) &= \sum_k p(z^k = 1 \mid \pi) p(x, \mid z^k = 1, \mu, \Sigma) \\ &= \sum_{z_n} \prod_k \left((\pi_k)^{z_n^k} N(x_n : \mu_k, \Sigma_k)^{z_n^k} \right) = \sum_k \pi_k N(x, \mid \mu_k, \Sigma_k) \end{aligned}$$

Example: Gaussian Mixture Models (GMMs)

- Consider a mixture of K Gaussian components: $p(x_n|\mu,\Sigma)=\sum_k \pi_k N(x,|\mu_k,\Sigma_k)$
- Recall MLE for completely observed data
  - Data log-likelihood: $\ell(\boldsymbol{\theta};D)=\log\prod_n p(z_n,x_n)=\log\prod_n p(z_n\,|\,\pi)p(x_n\,|\,z_n,\mu,\sigma)$

$$=\sum_n \log\prod_k \pi_k^{z_n^k}+\sum_n \log\prod_k N(x_n;\mu_k,\sigma)^{z_n^k}$$

$$=\sum_n\sum_k z_n^k \log\pi_k-\sum_n\sum_k z_n^k \tfrac{1}{2\sigma^2}(x_n-\mu_k)^2+C$$

  - MLE:

$$\hat{\pi}_{k,MLE}=\arg\max_\pi \ell(\boldsymbol{\theta};D),$$

$$\hat{\mu}_{k,MLE}=\arg\max_\mu \ell(\boldsymbol{\theta};D) \qquad \Rightarrow \quad \hat{\mu}_{k,MLE}=\frac{\sum_n z_n^k x_n}{\sum_n z_n^k}$$

$$\hat{\sigma}_{k,MLE}=\arg\max_\sigma \ell(\boldsymbol{\theta};D)$$

- What if we do not know z_n?

Why is Learning Harder?

- Complete log likelihood: if both $\boldsymbol{x}$ and $\boldsymbol{z}$ can be observed, then

$$\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z}) = \log p(\boldsymbol{x}, \boldsymbol{z}|\theta) = \log p(\boldsymbol{z}|\theta_z) + \log p(\boldsymbol{x}|\boldsymbol{z}, \theta_x)$$

  - Decomposes into a sum of factors, the parameter for each factor can be estimated separately
- But given that $\boldsymbol{z}$ is not observed, $\ell_c(\theta; \boldsymbol{x}, \boldsymbol{z})$ is a random quantity, cannot be maximized directly
- Incomplete (or marginal) log likelihood: with $\boldsymbol{z}$ unobserved, our objective becomes the log of a marginal probability:

$$\ell(\theta; \boldsymbol{x}) = \log p(\boldsymbol{x}|\theta) = \log \sum_z p(\boldsymbol{x}, \boldsymbol{z}|\theta)$$

  - All parameters become coupled together
  - In other models when $\boldsymbol{z}$ is complex (continuous) variables (as we'll see later), marginalization over $\boldsymbol{z}$ is intractable.

Questions?