Deep Generative Models

Jimeng Sun

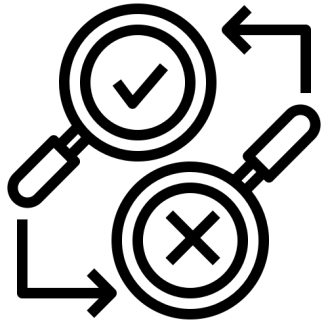
ILLINOIS

Outline

- Generative Adversarial Network (GAN)
  - Method
  - Application: Synthetic patient record generation
- Variational Autoencoder (VAE)
  - Method
  - Application: Molecule generation

Generative Adversarial Network (GAN)

- **Goal**: to create new samples that resemble training data
- **Idea**: train two neural networks:



Generator for creating realistic but synthetic samples

Discriminator for differentiating synthetic and real samples

Goodfellow et al. *Generative Adversarial Networks*. NIPS 2014

ILLINOIS

Healthcare applications of GAN

Generate synthetic electronic health records

Choi et al. 2017. “Generating Multi-Label Discrete Patient Records Using Generative Adversarial Networks.” In *Proceedings of the 2nd Machine Learning for Healthcare Conference*

Zhang et al. 2020. “Ensuring Electronic Medical Record Simulation through Better Training, Modeling, and Evaluation.” *Journal of the American Medical Informatics Association: JAMIA* 27 (1): 99–108.

De Novo design: molecule generation

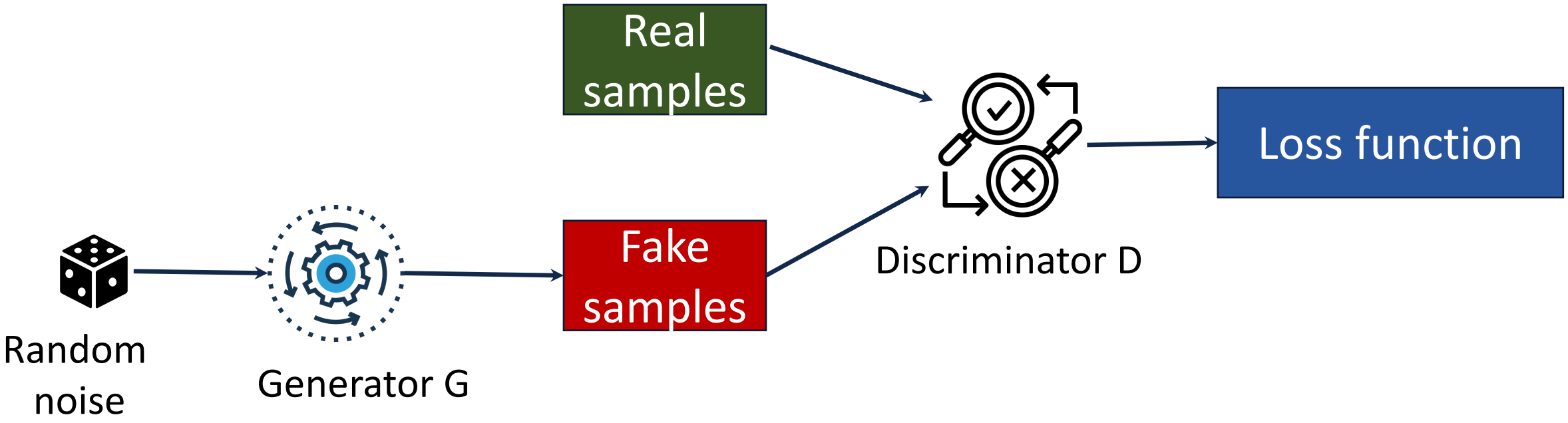
De Cao, Nicola, and Thomas Kipf. 2018. “MolGAN: An Implicit Generative Model for Small Molecular Graphs.” *arXiv [stat.ML]*. arXiv. http://arxiv.org/abs/1805.11973.

Jin, Wengong, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. 2018. “Learning Multimodal Graph-to-Graph Translation for Molecular Optimization.” *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1812.01070.

Maziarka, Łukasz, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoł. 2020. “Mol-CycleGAN: A Generative Model for Molecular Optimization.” *Journal of Cheminformatics* 12 (1): 2.

ILLINOIS

GAN Structure



- Generator learns to create realistic data (fake samples)
- Discriminator learns to separate fake samples with real samples

ILLINOIS

GAN Loss function

Log-likelihood that real samples are real

Log-likelihood that fake samples are fake

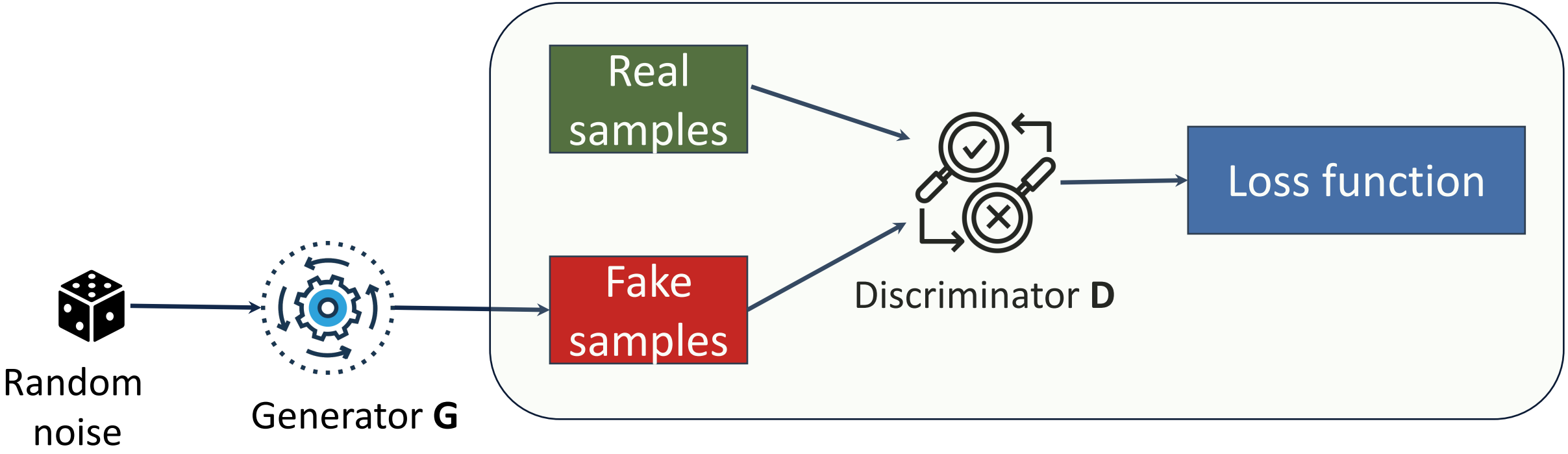
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Sample **real data**

Sample **random noise**

ILLINOIS

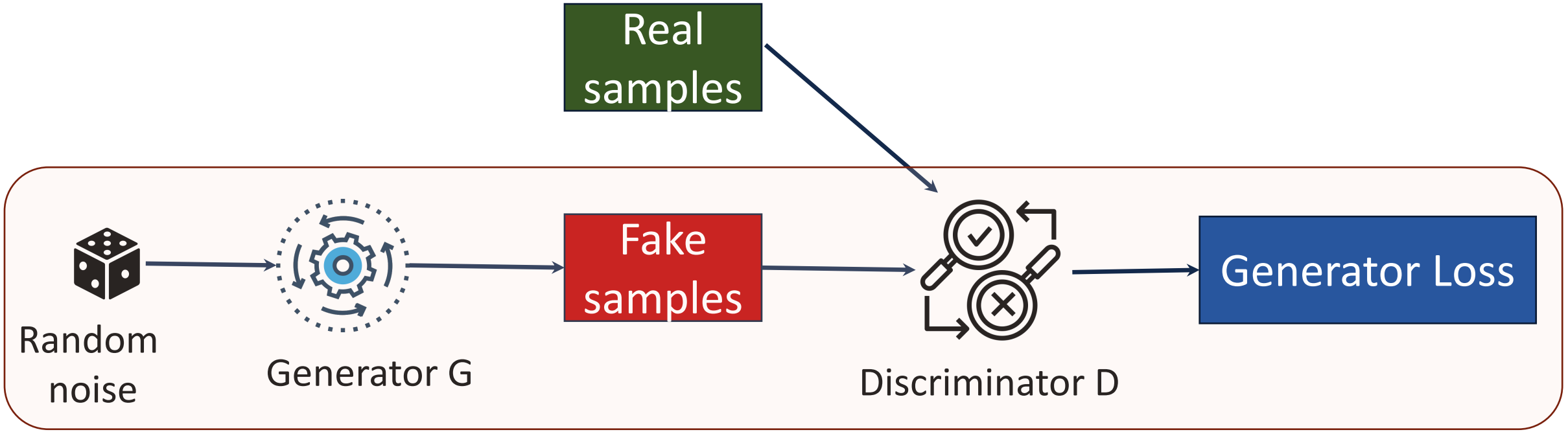
Discriminator D



- D classifies real and fake samples
- Discriminator loss penalizes any mistake for misclassifying fake from real
- Update parameters via backprop from Discriminator loss

ILLINOIS

Generator G



- Random noise will be converted to realistic samples (fake samples)
- D classifies the fake samples
- Generator loss penalizes for failing to fool D
- Backprop through D and G

Generator Loss

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

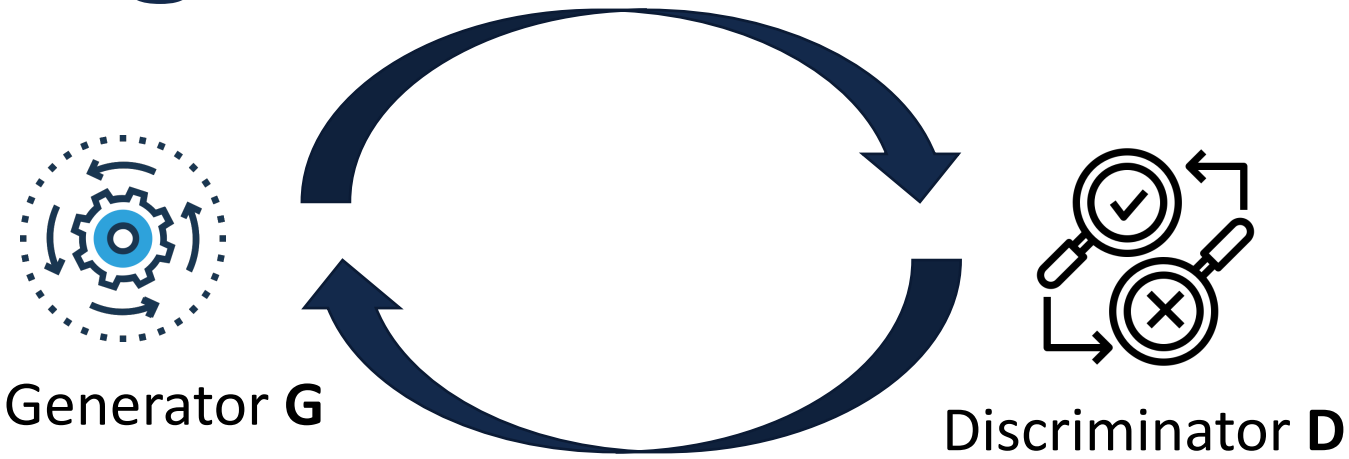
Generator Loss

- The generator loss is equivalent to

$$L_G = -\mathbb{E}_{\boldsymbol{z}} \log D(G(\boldsymbol{z}))$$

which is easier to train

Training GAN



- Iteratively, train D for one or few epochs, then train G for one or few epochs
- Convergence of GAN training is still an open problem
  - No reliable metrics to indicate the convergence
  - Overtraining can lead to performance degradation

Generating Multi-Label Discrete Patient Records using Generative Adversarial Networks

How to generate realistic EHR data

Choi, Edward, et al. 2017. “Generating Multi-Label Discrete Electronic Health Records Using Generative Adversarial Networks.” *Machine Learning for Health Care 2017*.

ILLINOIS

Healthcare Research and Data

- Massive data are electronically collected by healthcare organizations
- Privacy concerns often hinder data science research in healthcare

ILLINOIS

Approaches to handle privacy concerns

- De-identification by perturbation methods
  - Derived from the real data; not impregnable to attacks

- Generating synthetic datasets
  - Has not been realistic enough for machine learning tasks

ILLINOIS

Objective of MedGAN

Generate realistic EHR data using Generative Adversarial Network (GAN)

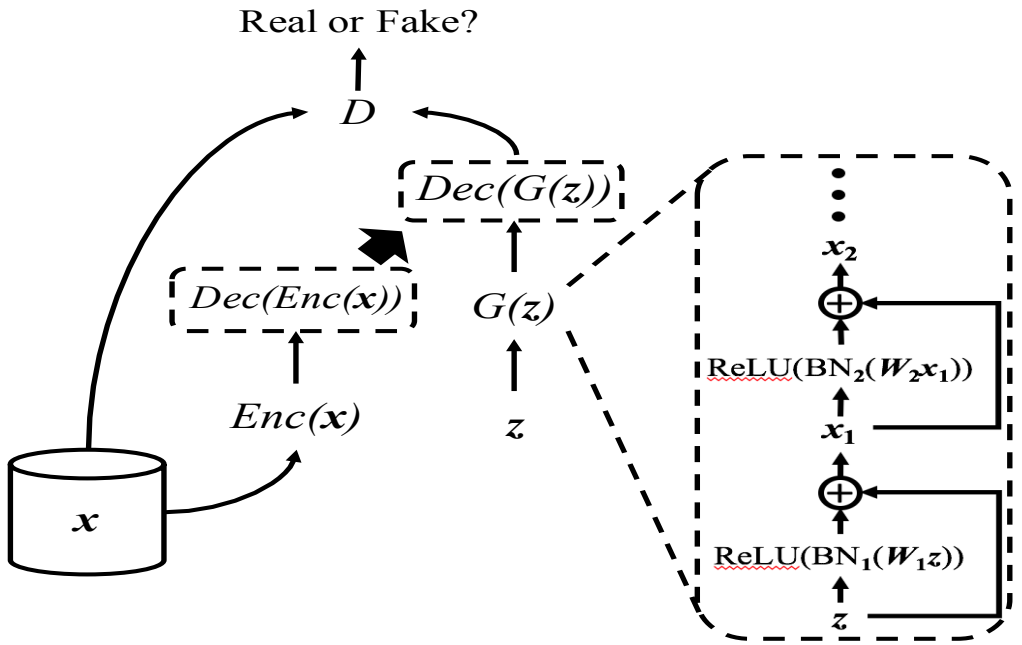
- High-quality: Synthetic datasets can train ML models that perform similarly to the ones trained on the real datasets

- Privacy-preserving: Practically impossible to gain knowledge on real patients from the synthetic dataset

ILLINOIS

MedGAN Architecture

- x: real-data
- z: random noise
- *Enc*: encoder of autoencoder
- *Dec*: decoder of autoencoder
- *G*: Generator
- *D*: Discriminator
- *BN*: Batch-norm



Generating completely synthetic data using GAN

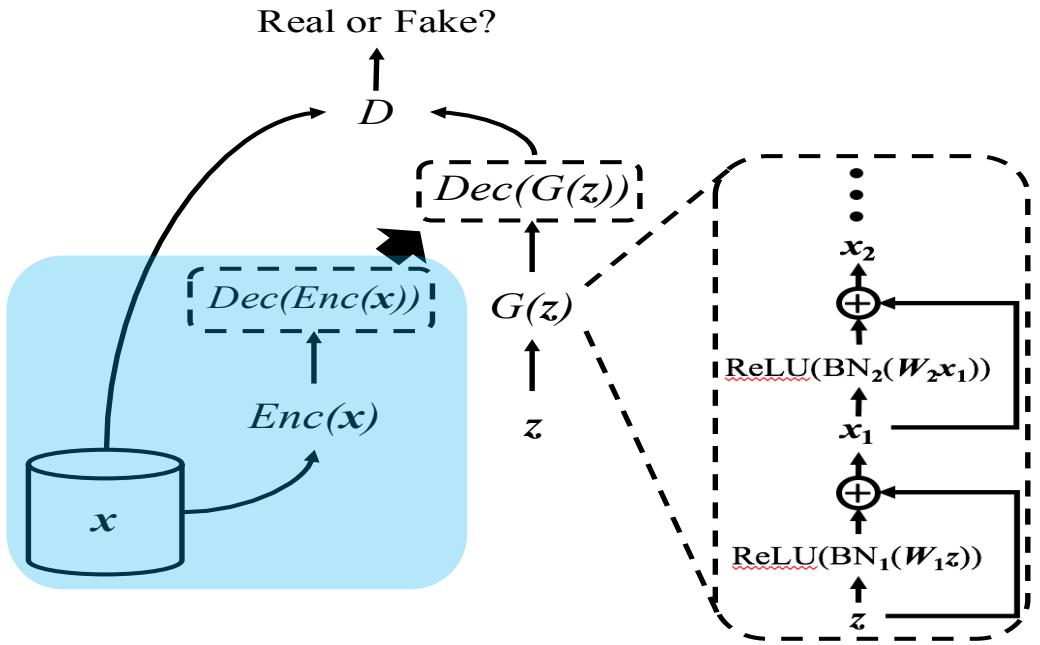
ILLINOIS

MedGAN Architecture

Pre-train Autoencoder

$$\frac{1}{m}\sum_{i=0}^{m} \mathbf{x}_i \log \mathbf{x}'_i + (1-\mathbf{x}_i)\log(1-\mathbf{x}'_i)$$

where $\mathbf{x}'_i = Dec(Enc(\mathbf{x}_i))$



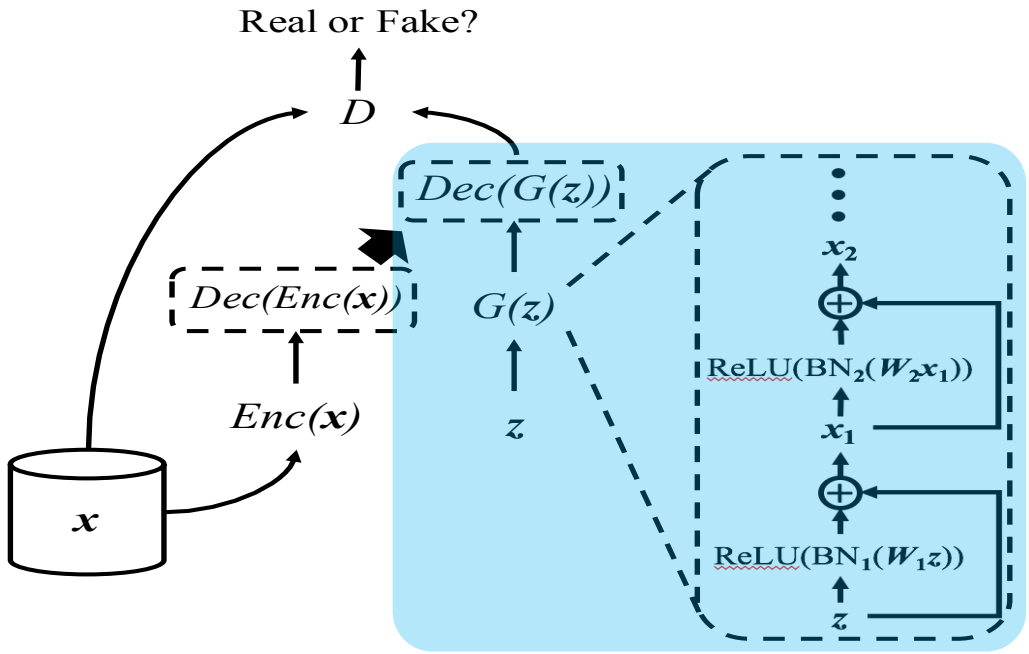
ILLINOIS

MedGAN Architecture

Update Generator parameters

$$\theta_{g,dec} \leftarrow \theta_{g,dec} + \alpha \nabla_{\theta_{g,dec}} \frac{1}{m} \sum_{i=1}^{m} \log D(\mathbf{x}_{\mathbf{z}_i})$$

where $\mathbf{x}_{\mathbf{z}_i} = Dec(G(\mathbf{z}_i))$



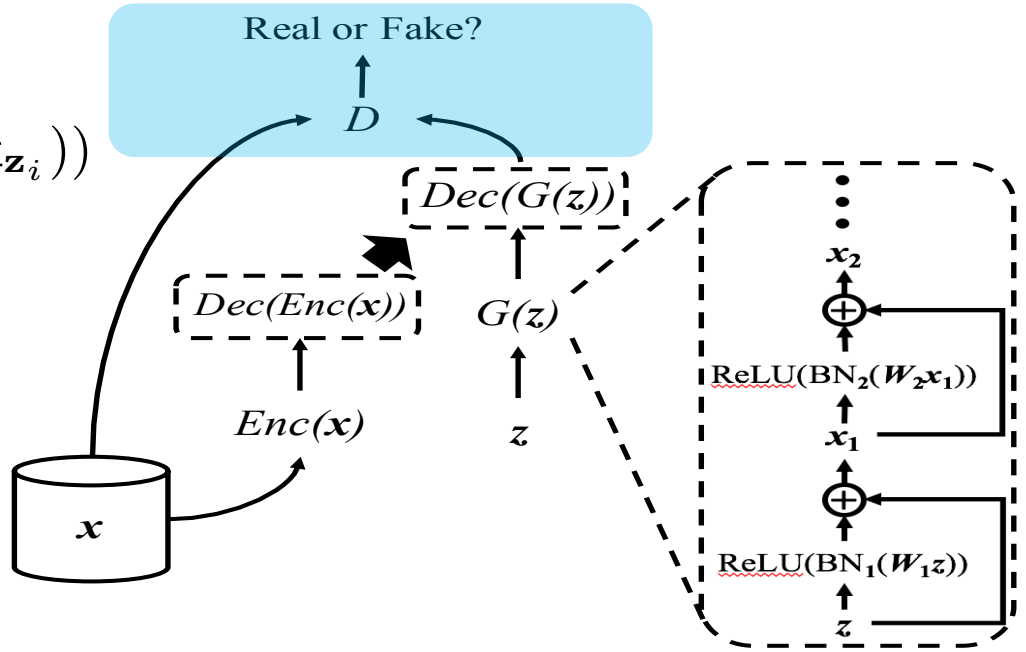
ILLINOIS

MedGAN Architecture

Update Discriminator parameters

$$\theta_d \leftarrow \theta_d + \alpha \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \log D(\mathbf{x}_i) + \log(1 - D(\mathbf{x}_{\mathbf{z}_i}))$$

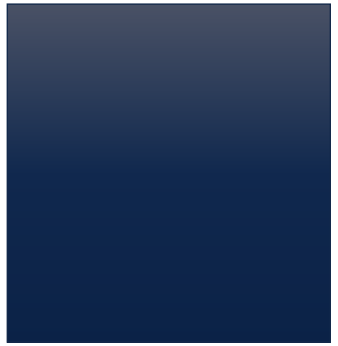
where $\mathbf{x}_{\mathbf{z}_i} = Dec(G(\mathbf{z}_i))$

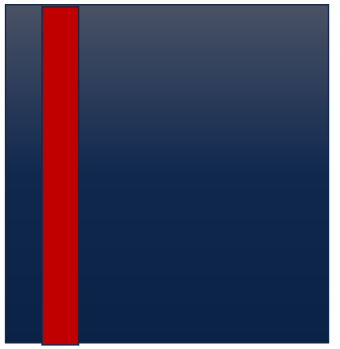


ILLINOIS

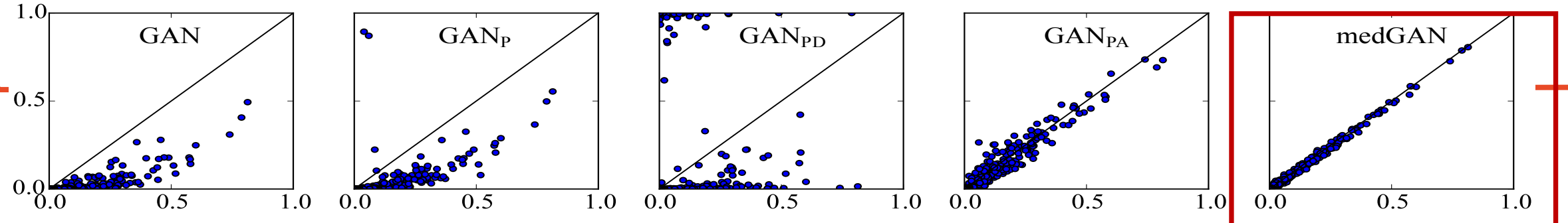
Quantitative Evaluation

- Dimension-wise probability
  - Compare the dimension-wise probability of the training data and the synthetic data
- Dimension-wise prediction
  - Assess how well medGAN captures the inter-dimensional relationships
  - Select one dimension k as the label, use the remaining dimensions as features for a logistic regression (LR) classifier.
- Experiments conducted on Sutter PAMF

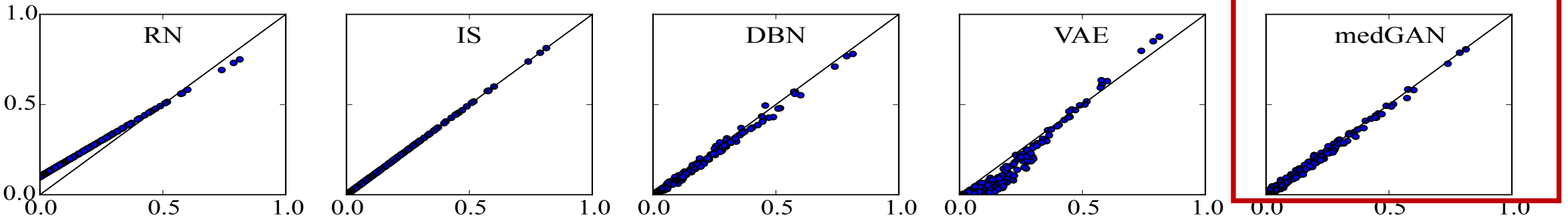




Dimension-wise Probability



(a) Dimension-wise probability performance of various versions of medGAN.

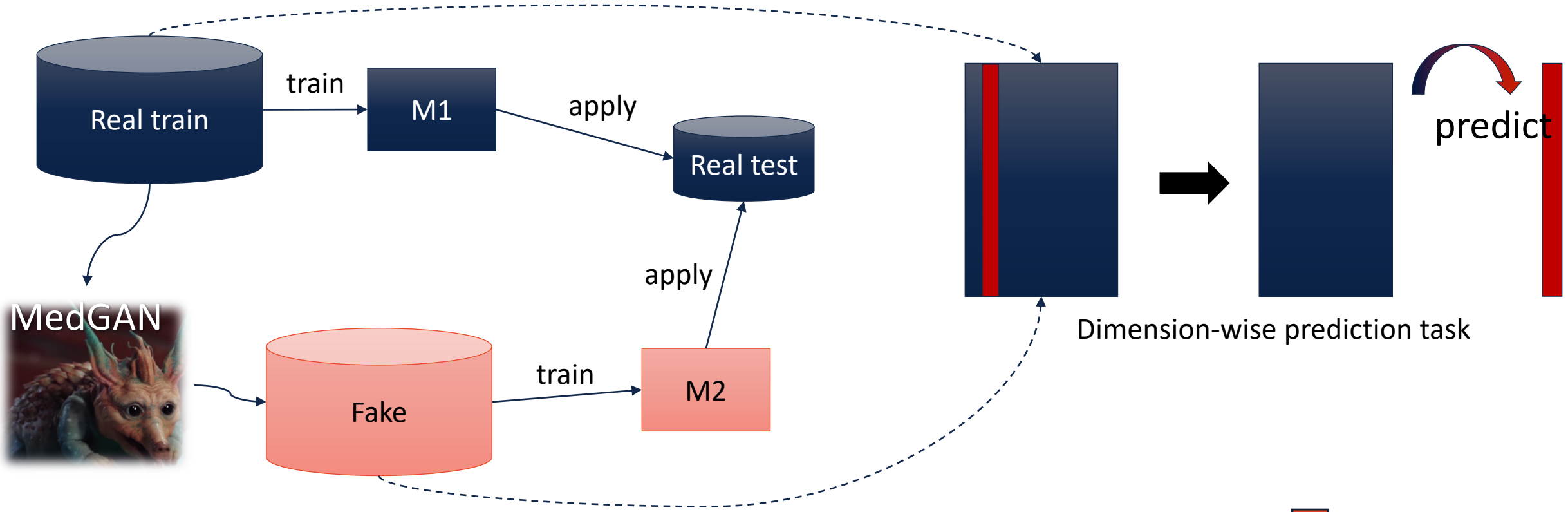


(b) Dimension-wise probability performance of baseline models and medGAN.

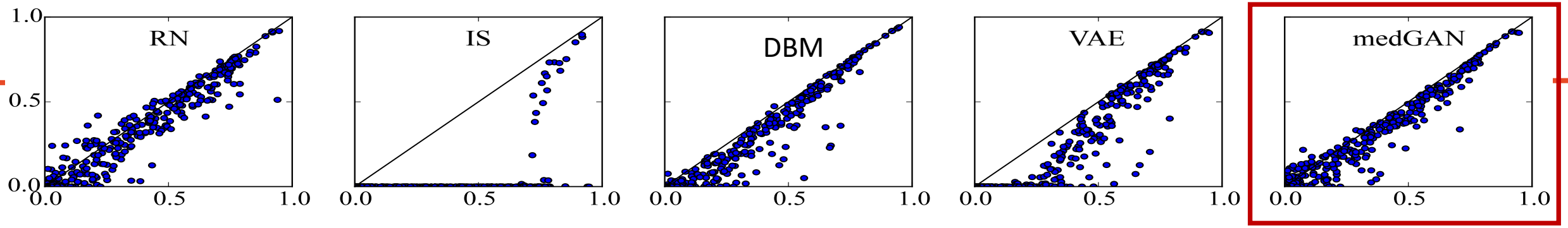
- X-axis: dimension-wise probability of real data
- Y-axis: dimension-wise probability of synthetic data
- A dot represents a single dimension (i.e. medical code)
- Diagonal line indicates the ideal performance

Quantitative Evaluation: Dimension-wise prediction

Select one dimension as the target, use the remaining dimensions as features for a logistic regression (LR) classifier.



Dimension-wise Prediction



Real data + random noise

Independent dimensional wise sampling

Deep Boltzmann machine

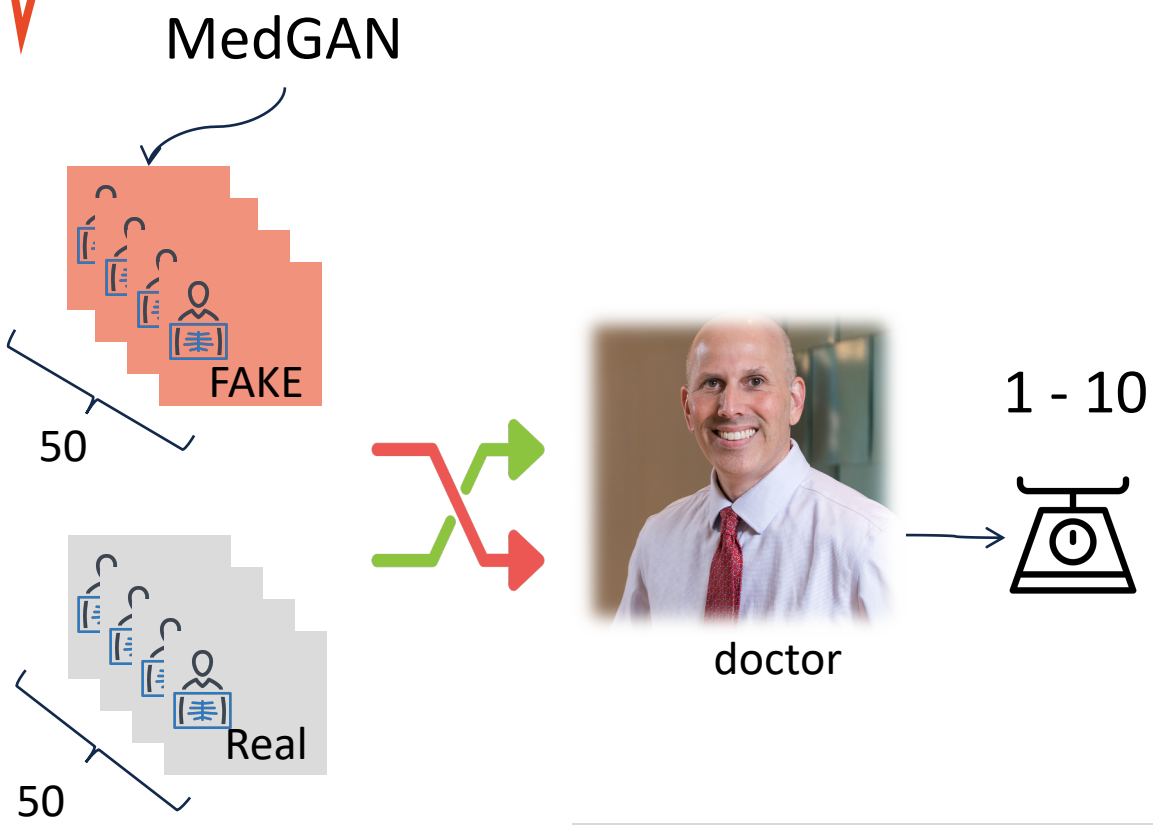
Variational autoencoder

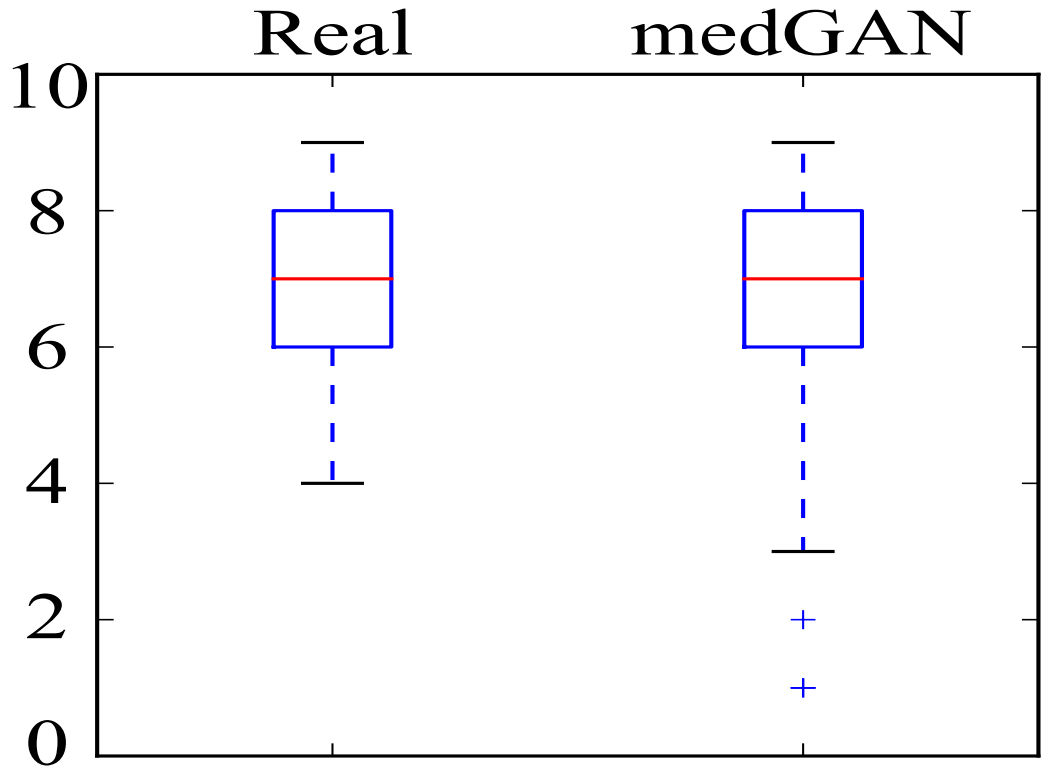
- X-axis: F1-score of LR trained by real data
- Y-axis: F1-score of LR trained by synthetic data
- A dot represents a single dimension selected as the label
- Diagonal line indicates the equal performance with real data

Data generated by medGAN performs great for ML tasks

ILLINOIS

Qualitative Evaluation





Indistinguishable to a human doctor except a few outliers

ILLINOIS

Conclusion

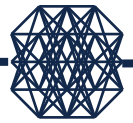
- MedGAN generates high-quality synthetic data for ML tasks
  - Dimension-wise prediction
  - Qualitative evaluation by a human doctor

- MedGAN preserves privacy
  - Attacker can confirm the presence of very small number of patients
  - Attacker cannot gain useful knowledge on unknown attributes

Choi, Edward, et al. 2017. “Generating Multi-Label Discrete Electronic Health Records Using Generative Adversarial Networks.” *Machine Learning for Health Care 2017*.

ILLINOIS

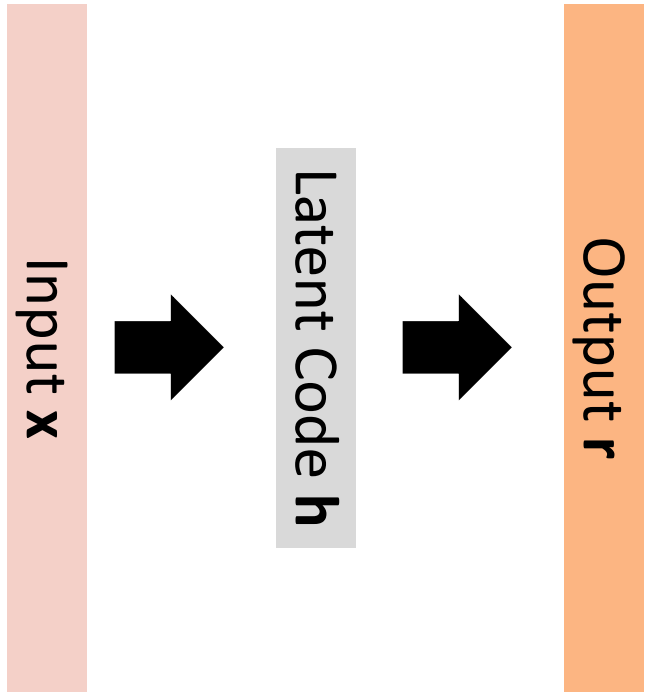
Variational Autoencoder (VAE)



- VAE is a generative model for creating realistic data samples
- VAE is in the intersection of **deep learning** and **probabilistic graphical model**
- VAE has many healthcare applications in
  - Molecule generation
  - Medical imaging analysis

ILLINOIS

Review: Autoencoders



Minimize reconstruction error

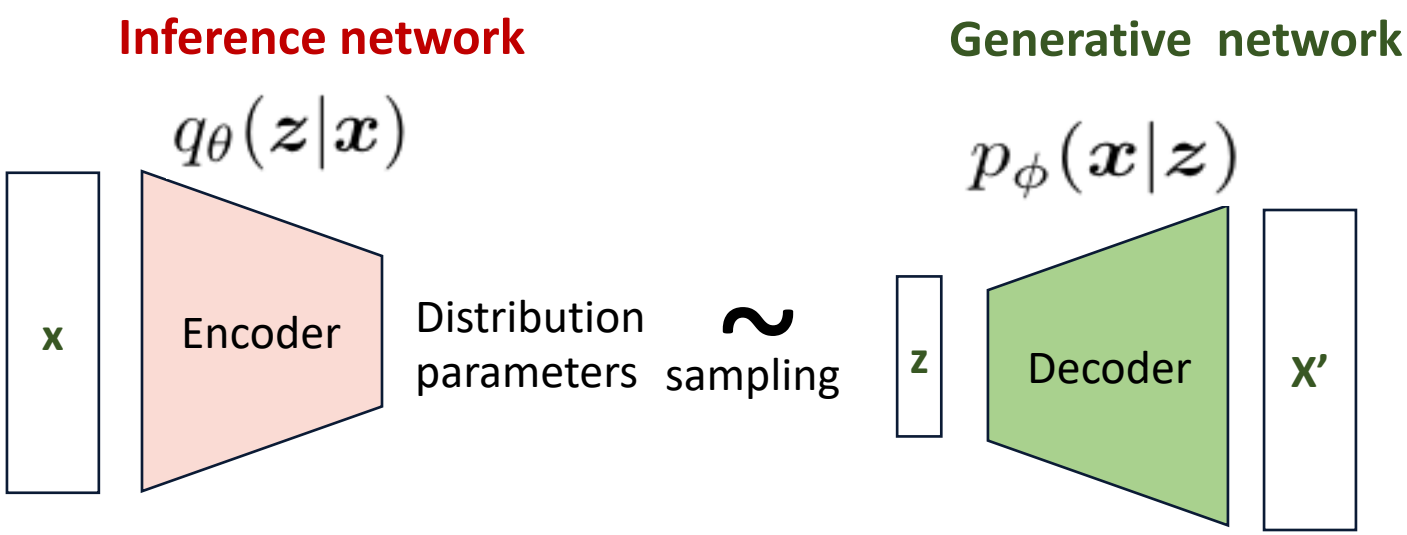
$$L(\boldsymbol{x}, \boldsymbol{r}) = ||\boldsymbol{x} - \boldsymbol{r}||^2$$ for Gaussian input

$$L(\boldsymbol{x}, \boldsymbol{r}) = -\sum_i [\boldsymbol{x}_i \log \boldsymbol{r}_i + (1 - \boldsymbol{x}_i) \log(1 - \boldsymbol{r}_i)]$$

for binary input

ILLINOIS

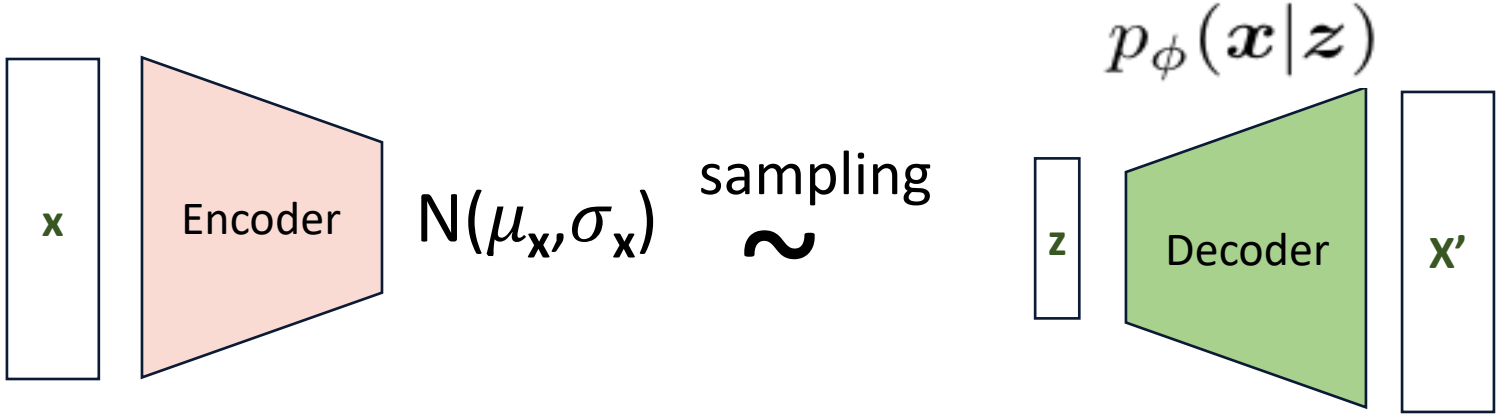
VAE: Deep learning view (high-level)



- Encode input as a distribution
- Sample a point from the encoding distribution
- Decode from the sample to generate a realistic data point

ILLINOIS

VAE: Deep learning view



- Encoder

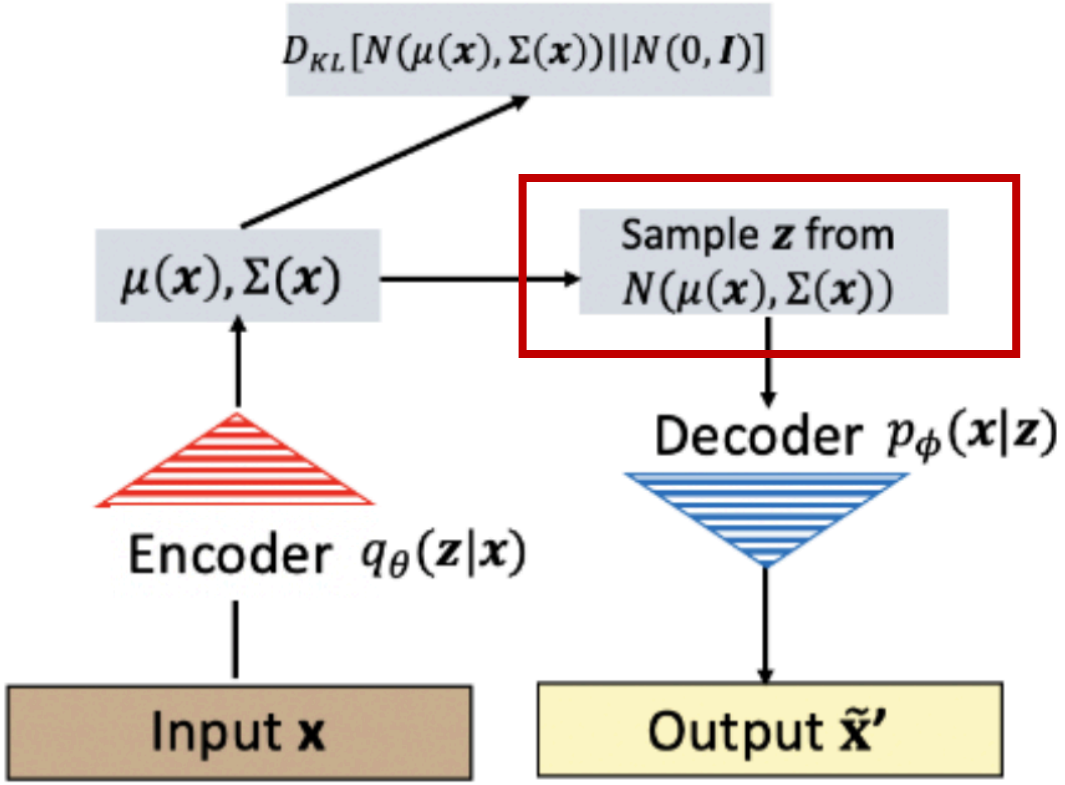
$$q_\theta(\boldsymbol{z}|\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}), \Sigma(\boldsymbol{x})) \qquad p_\phi(\boldsymbol{z}) \sim \mathcal{N}(0, \boldsymbol{I})$$

- Loss function for data point x

$$l_{\boldsymbol{x}} = -\mathbb{E}_{\boldsymbol{x}\sim D}\left[\log p_\phi(\boldsymbol{x}|\boldsymbol{z}) + \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\|p_{\boldsymbol{\phi}}(\boldsymbol{z}))\right]$$

ILLINOIS

VAE: Reparameterization trick

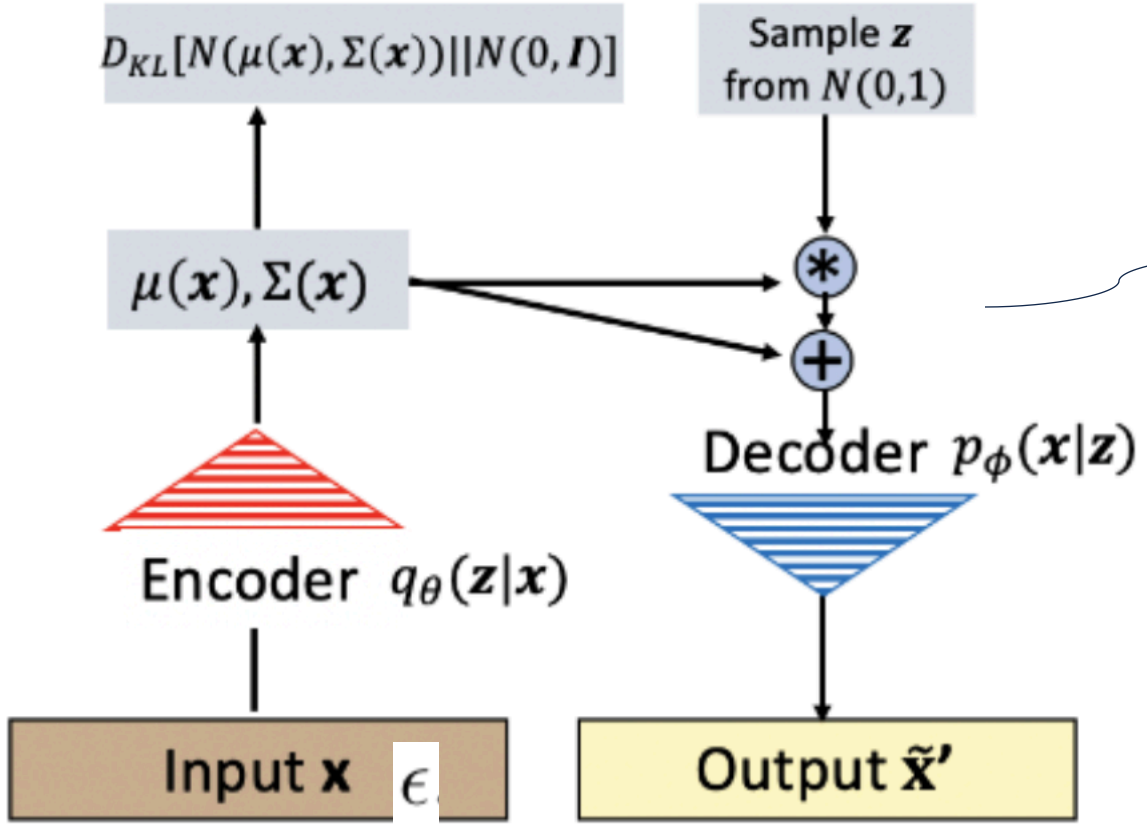


Without Reparameterization Trick

- Sampling step makes backprop difficulty

ILLINOIS

VAE: Reparameterization trick (cont.)



$$z = \mu(x) + \Sigma^{1/2}(x) * \epsilon$$

With Reparameterization Trick

ILLINOIS

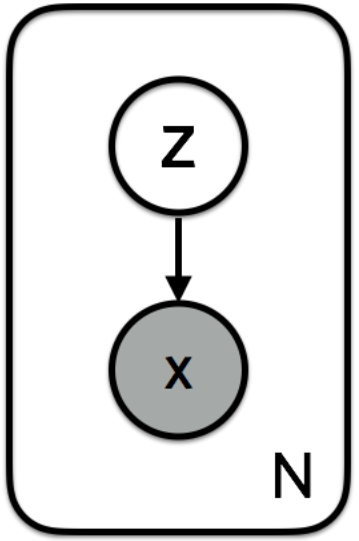
VAE: Probability view

- Posterior probability (encoder)

$$p_{\phi}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_{\phi}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p_{\phi}(\boldsymbol{x})}$$

where $p_{\phi}(\boldsymbol{x}) = \int p_{\phi}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$

Difficult to compute



$p_{\phi}(\boldsymbol{z}|\boldsymbol{x})$ → $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ Such that $\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \| p_{\phi}(\boldsymbol{z}|\boldsymbol{x}))$: is minimized

Approximated with

ILLINOIS

VAE: Variational approximation

$$\log p_{\boldsymbol{\phi}}(\boldsymbol{x}) = \underbrace{\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\right]}_{\text{ELBO}} + \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \| p_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})).$$

- Maximize ELBO == Minimize DL

ILLINOIS

VAE: Variational approximation (cont.)

$$\mathrm{ELBO}(\boldsymbol{x};\boldsymbol{\phi},\boldsymbol{\theta}) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})] - \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\|p_{\boldsymbol{\phi}}(\boldsymbol{z}))$$

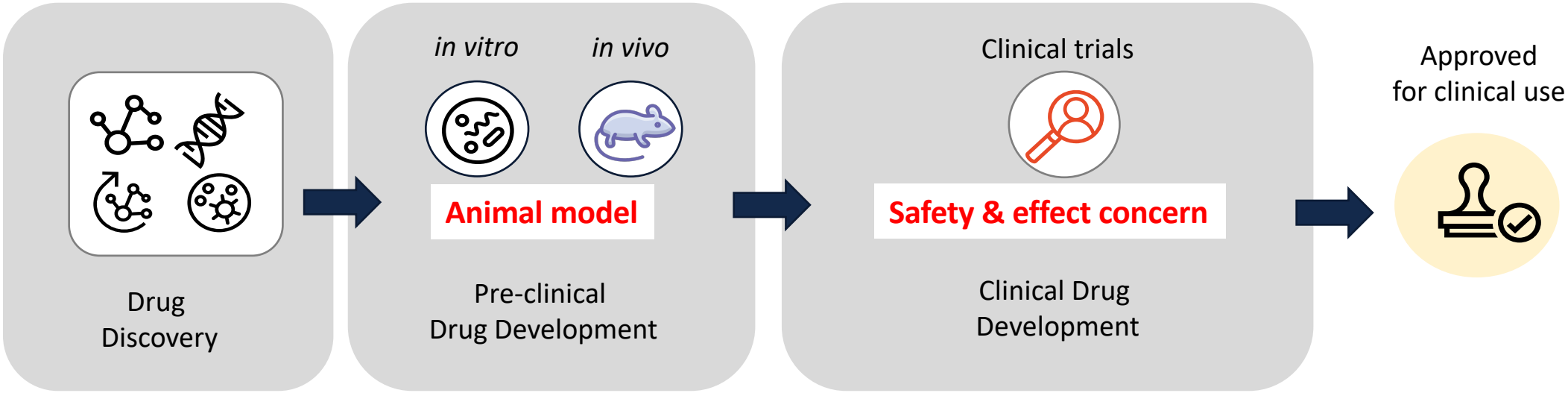
$$l_{\boldsymbol{x}} = \quad -\mathbb{E}_{\boldsymbol{x}\sim D}\left[\log p(\boldsymbol{x}|\boldsymbol{z}) + \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\|p_{\boldsymbol{\phi}}(\boldsymbol{z}))\right]$$

ILLINOIS

Deep learning for Drug discovery

ILLINOIS

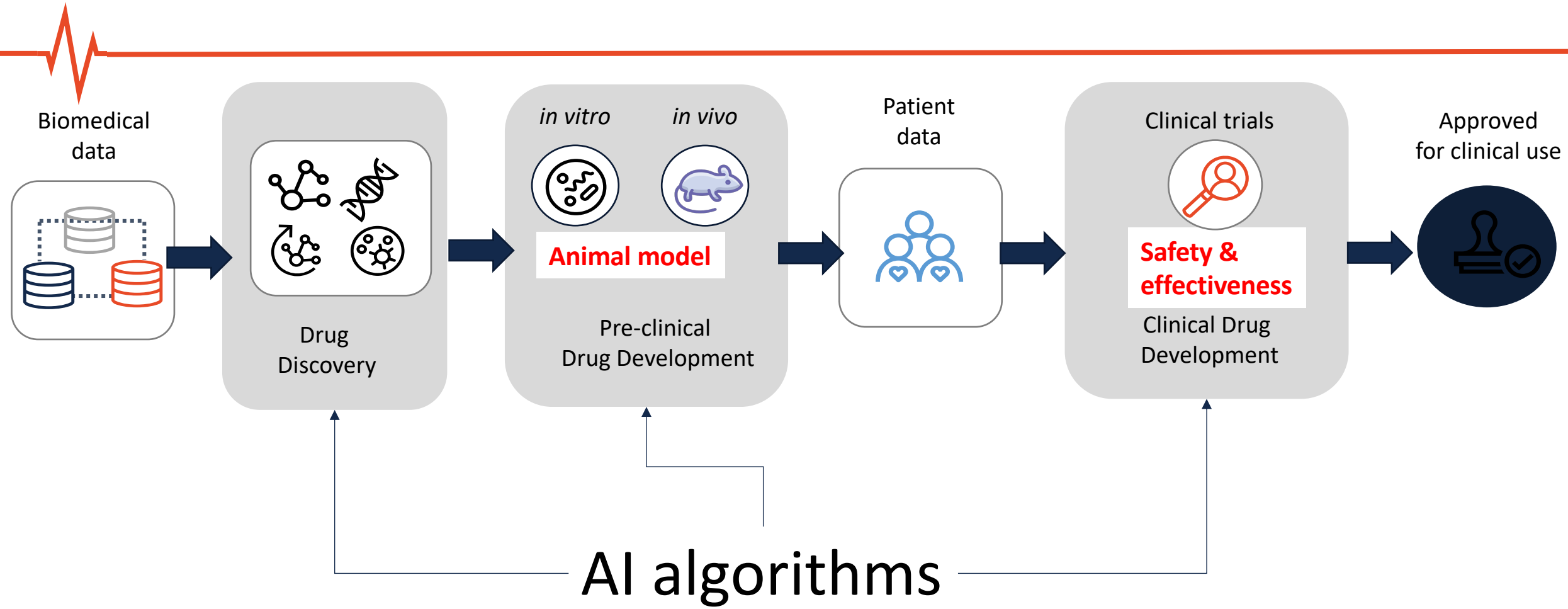
Traditional Drug Discovery & Development Process



	Drug discovery	Pre-clinical	Phase 1	Phase 2	Phase 3
Time spent	4-5 years	1-2 years	1-2 years	1-2 years	2-3 years
$ spent	$550M	$125M	$225M	$250M	$250M
Output	5,000 - 10,000 compounds	10-20 candidates	5-10 candidates	2-5 candidates	1-2 candidates

ILLINOIS

AI/ML to the Rescue: Why and How?



ILLINOIS

Molecule generation with VAE

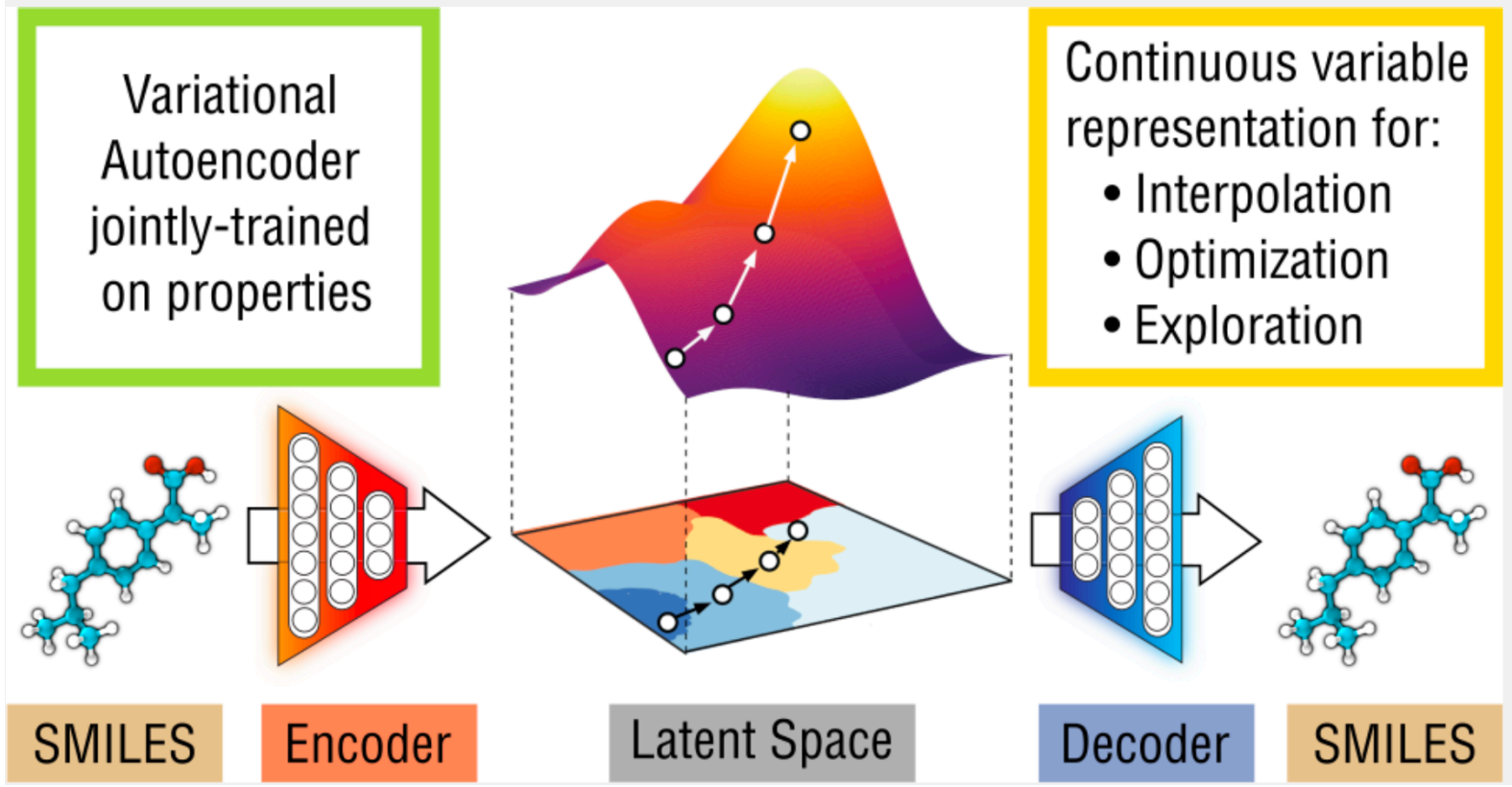
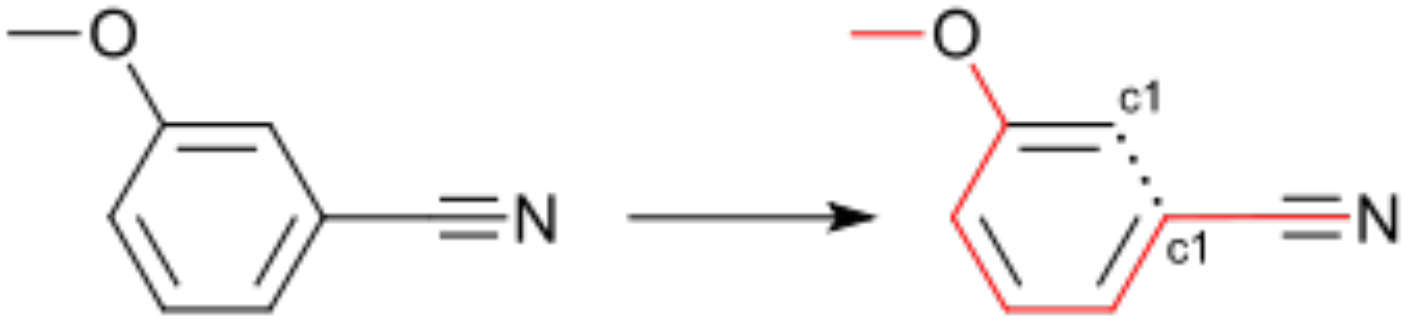


Image credit and citation: Gómez-Bombarelli et al. 2018. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules." *ACS Central Science* 4 (2): 268–76.

ILLINOIS

Simplified Molecular-Input Line-Entry System (SMILES)

3-cyanoanisole



() are used to branches

COc(c1)cccc1C#N

SMILES

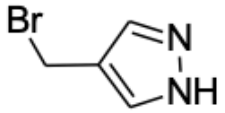
Numbers are used to represent rings

- SMILES string: Traverse the molecular graph in a depth-first manner following the atom with the smallest label at each branch point.

ILLINOIS

Molecular Representation Learning

Graph



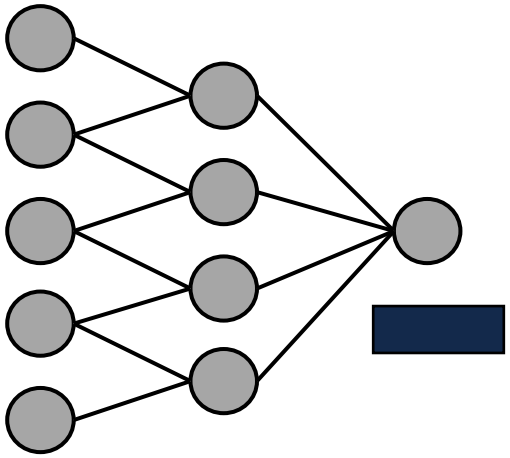
SMILES BrCc1c[nH]nc1

One-hot encoding

	Br	C	c	1	c	[	n	H	]	n	c	1
Br	1	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	1	0	0	0	0
c	0	0	1	0	1	0	0	0	0	0	1	0
n	0	0	0	0	0	0	1	0	0	1	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
[	0	0	0	0	0	1	0	0	0	0	0	0
]	0	0	0	0	0	0	0	0	1	0	0	0

vectorize

f



y
molecular property
e.g.,
solubility,
toxicity

ILLINOIS

VAE for molecule generation

- SMILES strings as input
- Seq2Seq model with RNN is used to encode SMILES
- VAE is used to generate new SMILES
- Another property prediction network is also added

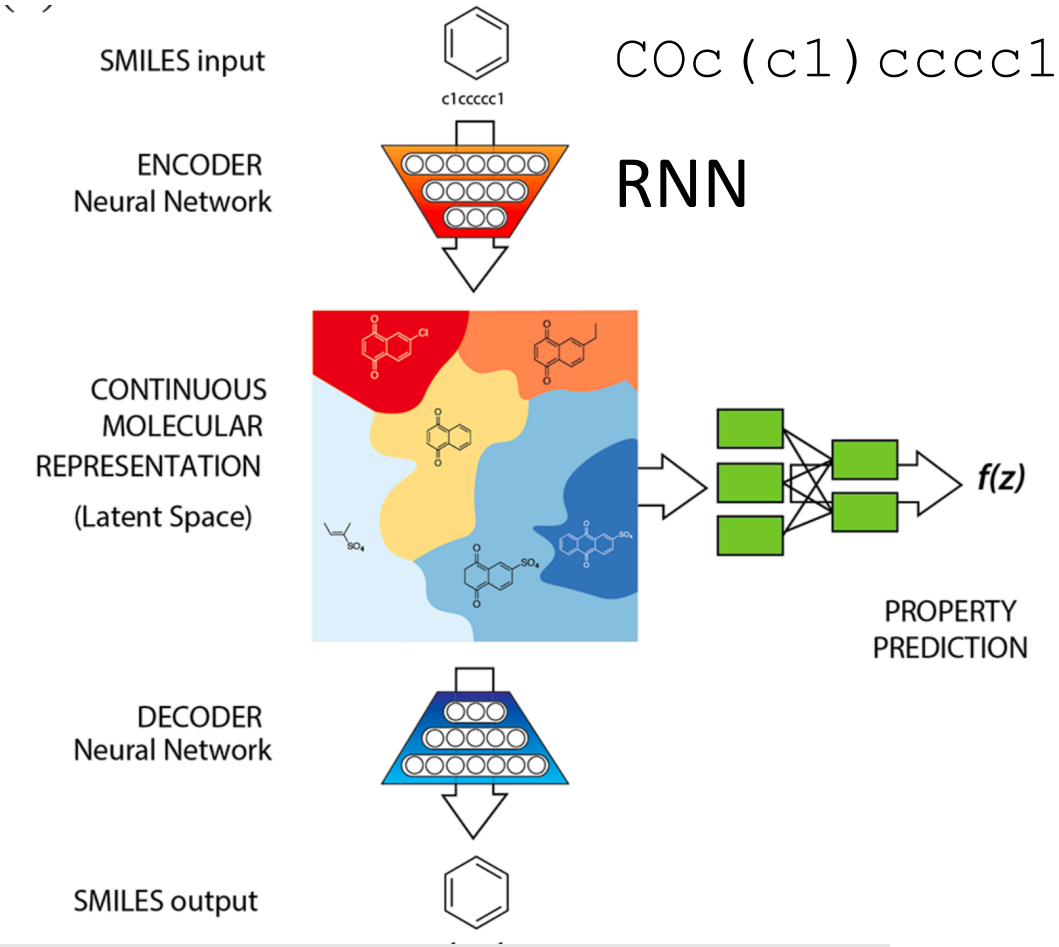


Image credit: Gómez-Bombarelli et al. 2018. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules." *ACS Central Science* 4 (2): 268–76.

Datasets

- 108K molecules from the QM9 data set of molecules
- 250K drug-like molecules in ZINC database

ILLINOIS

Results

water−octanol partition coefficient (logP)

synthetic accessibility score

Drug-likeness

source[a]	data set[b]	samples[c]	logP[d]	SAS[e]	QED[f]
Data	ZINC	249k	2.46 (1.43)	3.05 (0.83)	0.73 (0.14)
GA	ZINC	5303	2.84 (1.86)	3.80 (1.01)	0.57 (0.20)
VAE	ZINC	8728	2.67 (1.46)	3.18 (0.86)	0.70 (0.14)
Data	QM9	134k	0.30 (1.00)	4.25 (0.94)	0.48 (0.07)
GA	QM9	5470	0.96 (1.53)	4.47 (1.01)	0.53 (0.13)
VAE	QM9	2839	0.30 (0.97)	4.34 (0.98)	0.47 (0.08)

- GA is another method using genetic programming
- Goal is to generate realistic molecules that are similar to the input data

ILLINOIS