

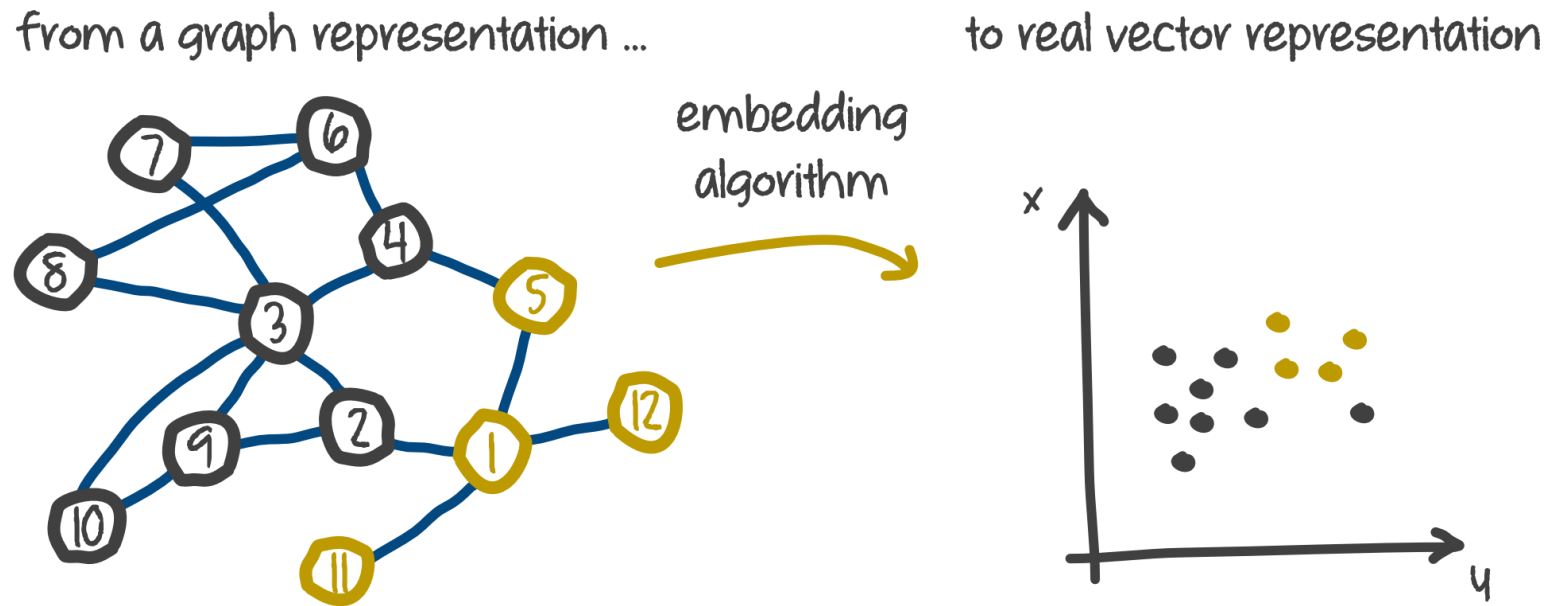
Adversarially Regularized Graph Autoencoder for Graph Embedding

Hyundo Lee

Seoul National University

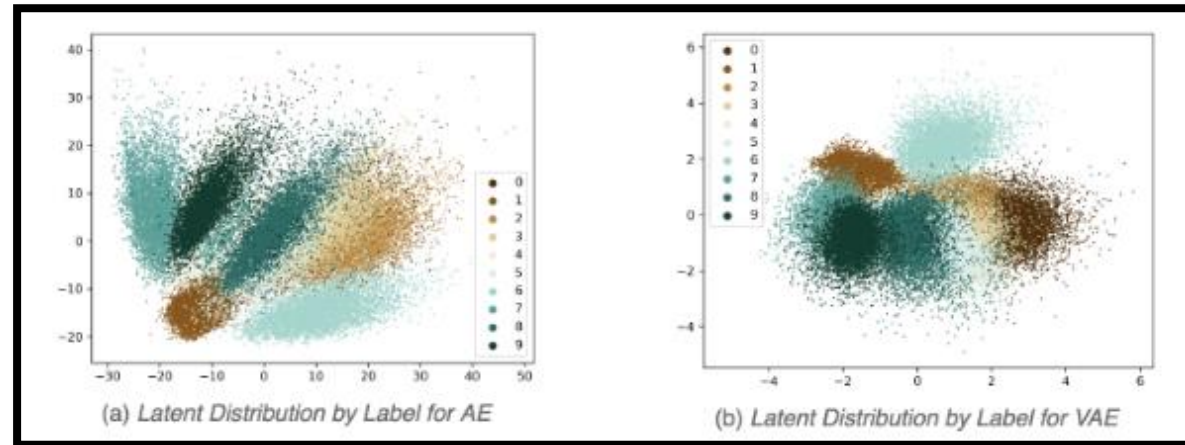
Graph Embedding Method with Prior Data Distribution

- Graph Embedding
 - Converts graph data into a low dimensional feature space,
 - Preserving **topological structure**, **vertex content**



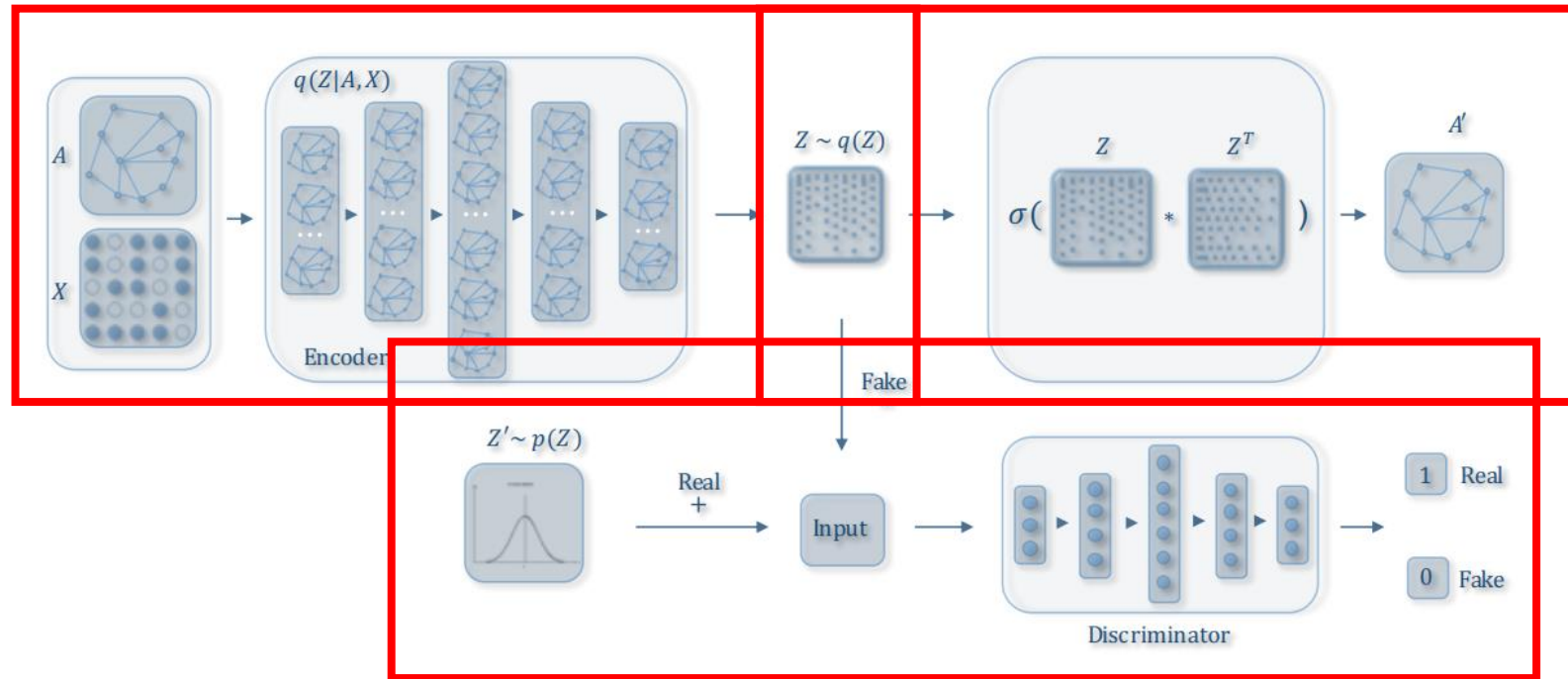
Graph Embedding Method with Prior Data Distribution

- Limitations of previous methods
 - Ignores the **data distribution** of the latent codes
 - Poor representation in real-world data



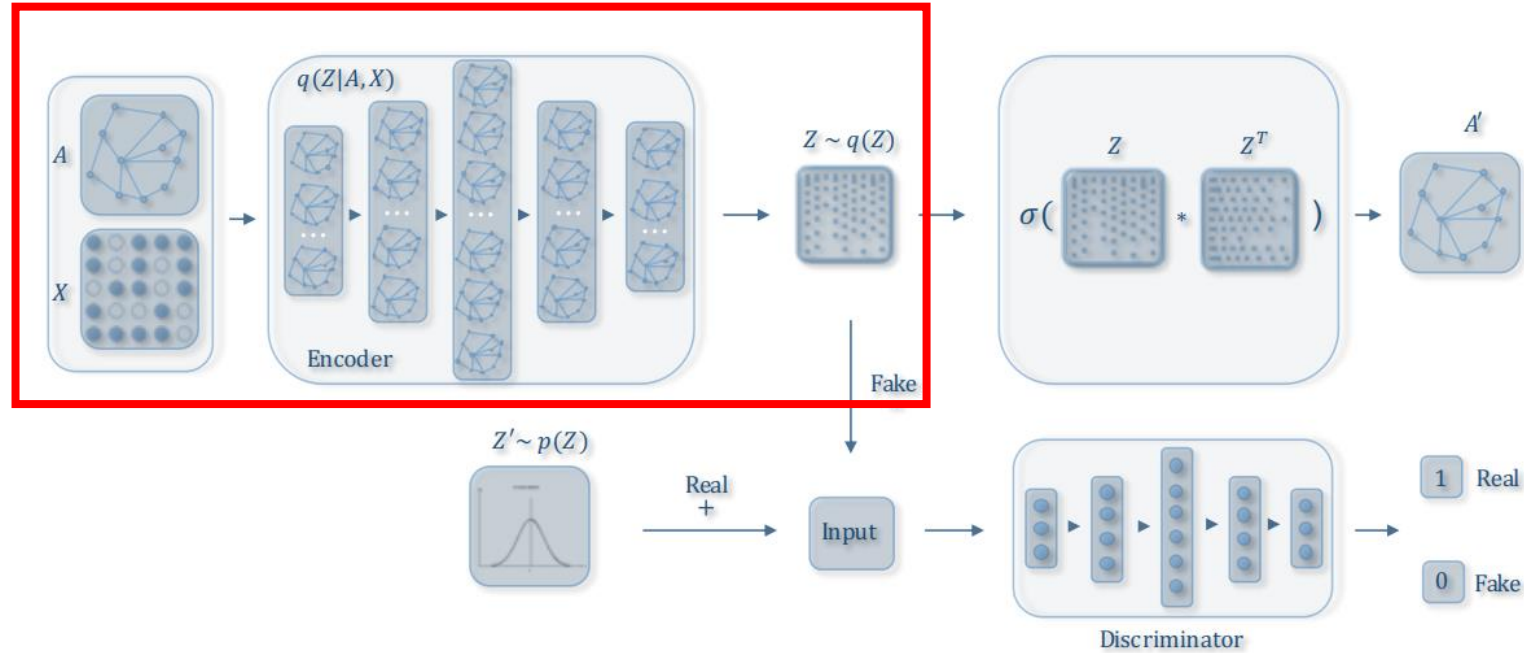
- ARGAs: **Adversarial Training** scheme
- ARVGA: + using **Variational** Graph Encoder

Overall Framework



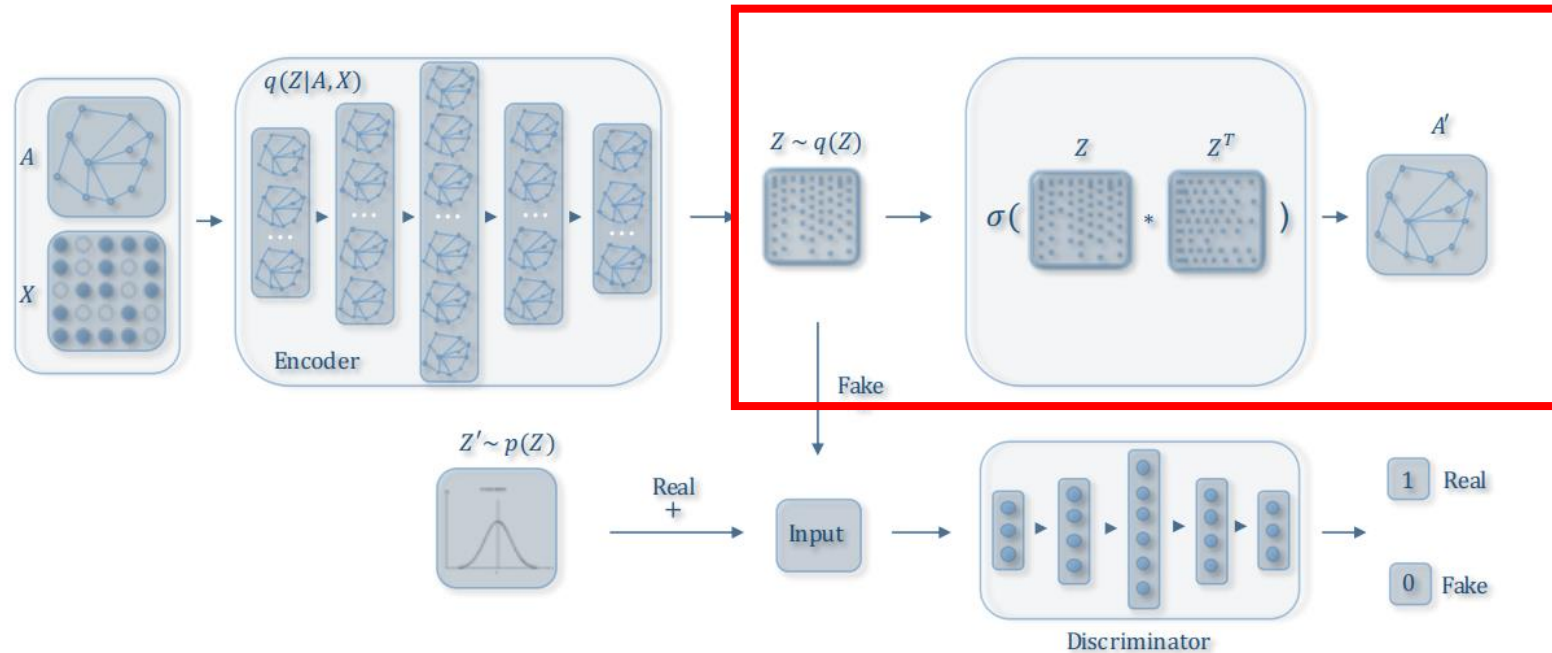
- **Encoder** $q: A \times X \rightarrow Z$ (or $\mu \times \sigma$)
- **Decoder** $p: Z \rightarrow \hat{A}$
- **Discriminator** $D: Z \rightarrow (0, 1)$

Framework – Encoder



- Encoder: GCN
 - $Z^{(l+1)} = f(Z^{(l)}, A | W^{(l)}) = \phi(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} Z^{(l)} W^{(l)})$
 - $Z^{(0)} = X \in \mathbb{R}^{n \times m}$
- Variational Graph Encoder: $Z^{(2)} = \mu, \log \sigma$

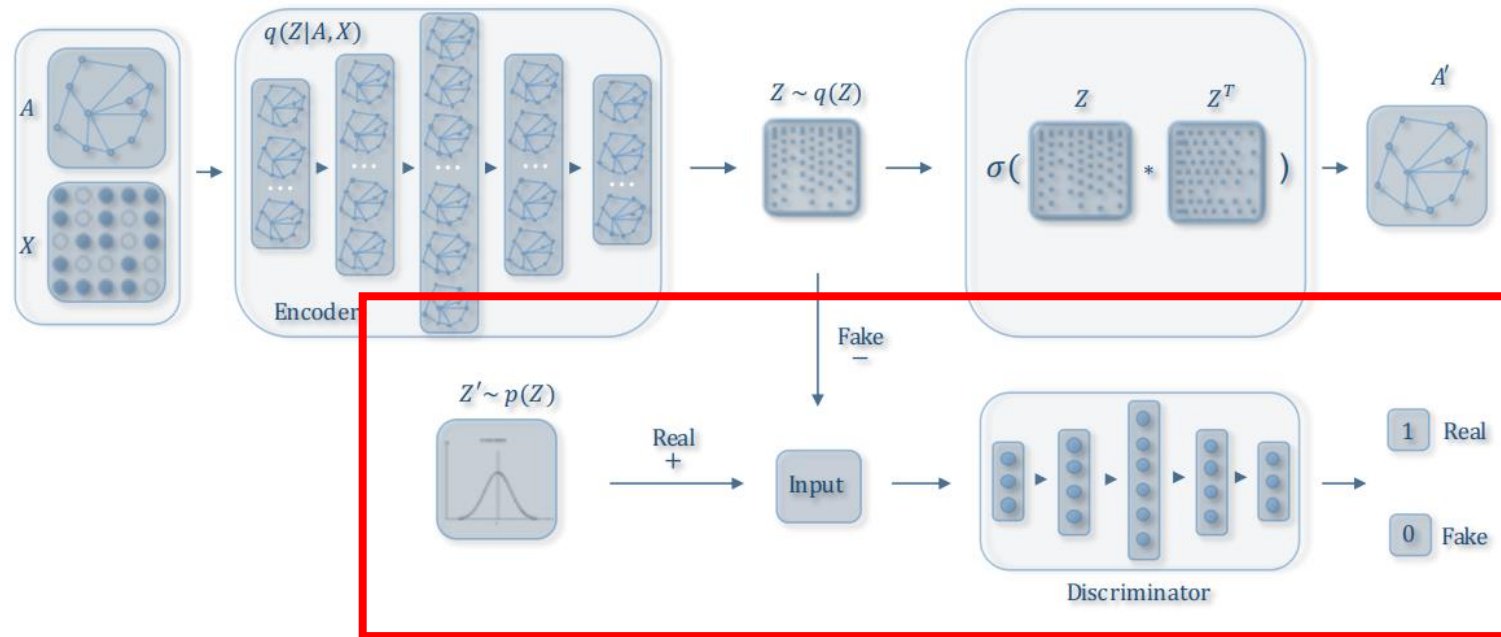
Framework – Decoder



- Decoder
 - $\hat{A} = \text{sigmoid}(ZZ^T)$
 - Tries to reconstruct the **topological structure from latent Z**

$$\mathcal{L}_1 = \mathbb{E}_{q(\mathbf{Z}|\mathbf{X}, \mathbf{A})} [\log p(\hat{\mathbf{A}}|\mathbf{Z})] - \text{KL}[q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \parallel p(\mathbf{Z})]$$

Framework – Adversarial Model



- Discriminator: tries to **discriminate**
 - $Z' \sim p(Z) = N(0, I)$
 - $Z \sim G(X, A)$
- Encoder: tries to **fool** the discriminator
 - $G(X, A) \rightarrow N(0, I)$

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{\mathbf{Z} \sim p_z} [\log \mathcal{D}(\mathbf{Z})] + \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\log (1 - \mathcal{D}(\mathcal{G}(\mathbf{X}, \mathbf{A})))]$$