Hot Topics Of Generative AI: LLMs and Diffusion Models

Yang et al. - “Directional diffusion models for graph representation learning“

TU Dortmund – Department of Computer Science
Chair 9 – Prof. Dr. Müller

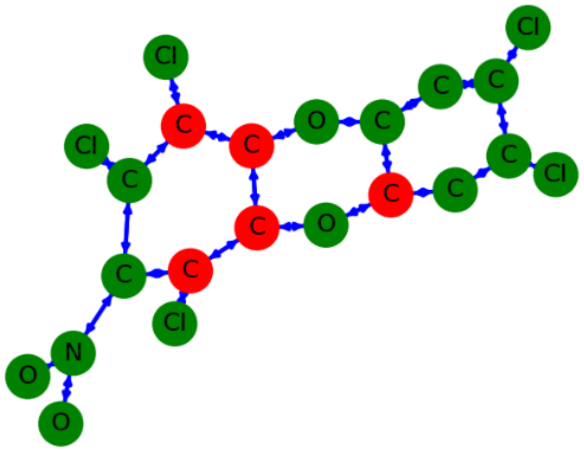
presented by Marlon Kapala
supervised by Tim Katzke

Agenda

1 What is Graph Representation Learning?

What is a graph?

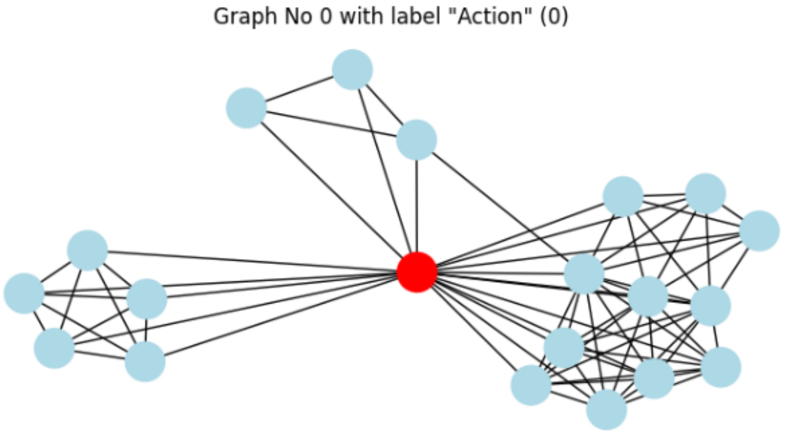
- G = (V, E), vertices V and edges E
- Graphs can represent anything from molecules to road networks or social networks
- Nodes can also have node features X



Zhu et al., 2020

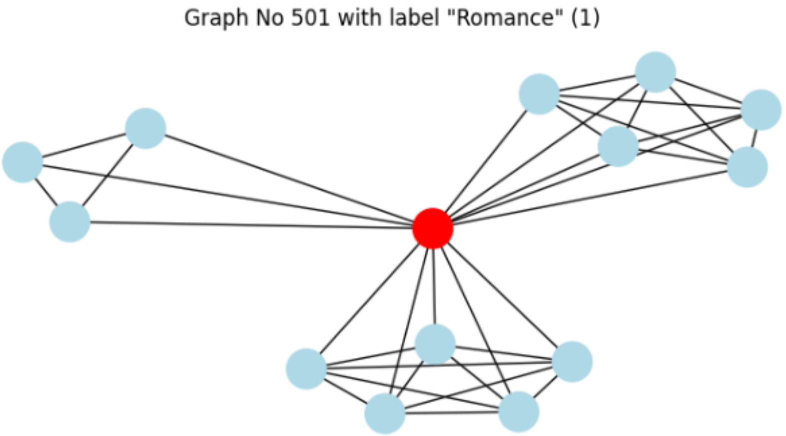
IMDB-Binary – A Graph Dataset

- Graph Learning on Graph Datasets enables the use of AI on those data structures



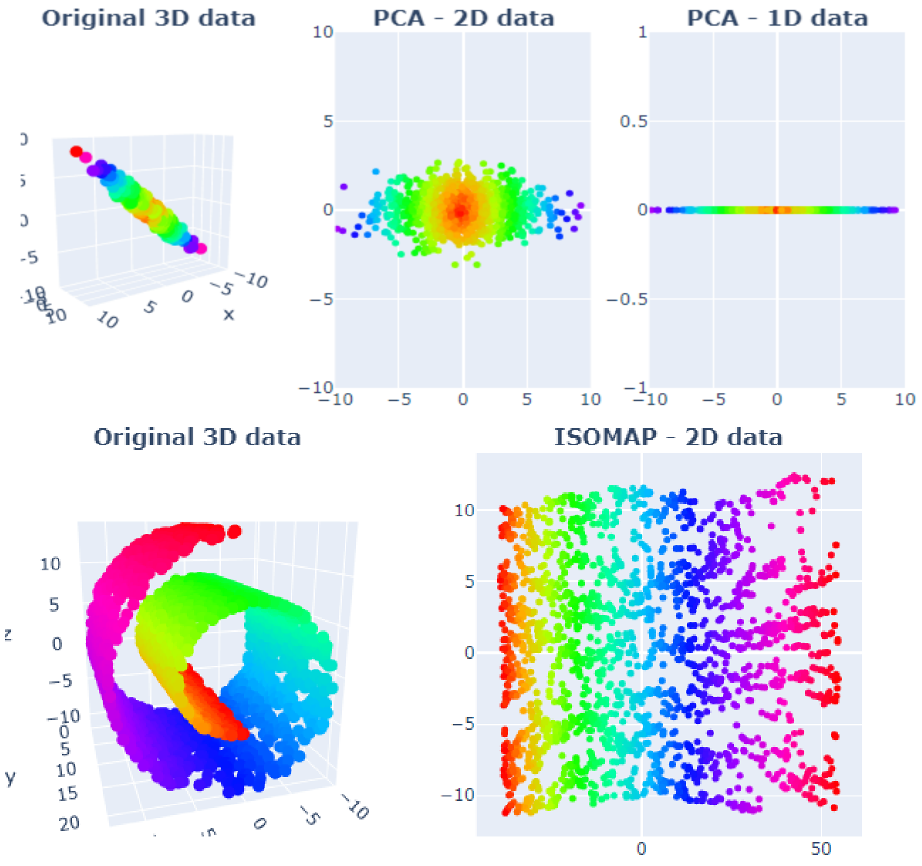
- IMDB-B contains ego-networks of actors from Action or Romance movies (Yanardag et al., 2015)

- From only the knowledge how actors have co-starred, models can determine the genre with an accuracy of up to 95% (Nguyen et al., 2019)



Representation Learning

- RL is an important part of Machine Learning that converts data to a form that can be worked with more easily

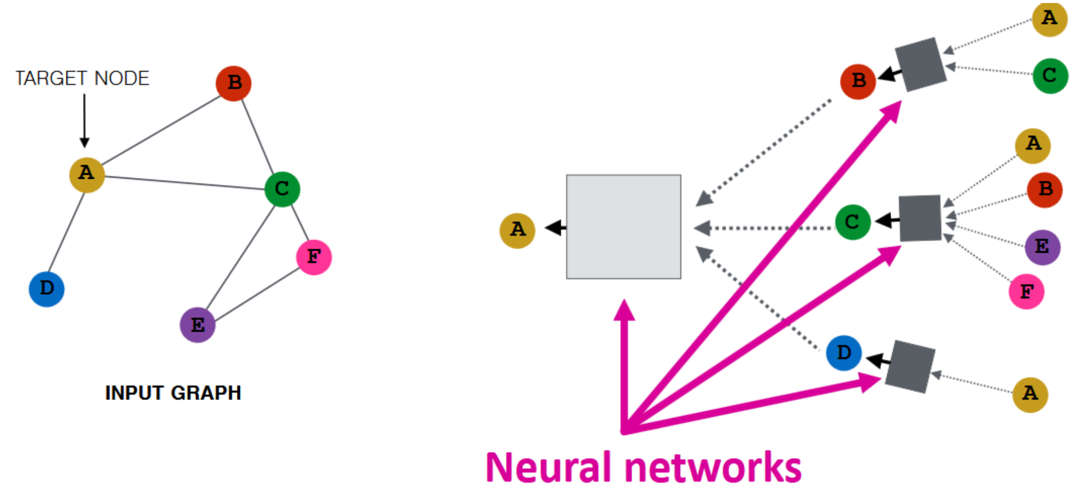


- Here, there is also a difference between unsupervised (without labels) and supervised learning (with labels)

- Dimensionality Reduction is a prominent subfield

Graph Representation Learning - GNNs

- For RL on Graphs, Graph Neural Networks are often used
- They work similar to Convolutional Neural Networks (CNNs)
- Instead of using neighboring pixels, the adjacency matrix, which encodes the edges, is used

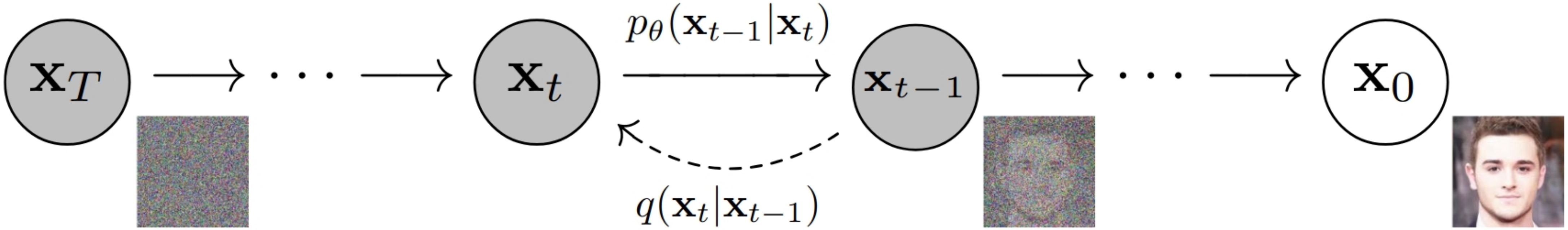


Jure Leskovec, Stanford CS224W: Machine Learning with Graphs

2 The Successful Diffusion Model

Denoising Probabilistic Diffusion Models

- Introduced to Machine Learning only recently (Ho et al., 2020), but has become the standard for image generation beating former SOTA technology (Dhariwal et al., 2021)



Ho et al., 2020

The DM's Algorithm

- To add noise to the images, Diffusion Models add Gaussian noise in each training step

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

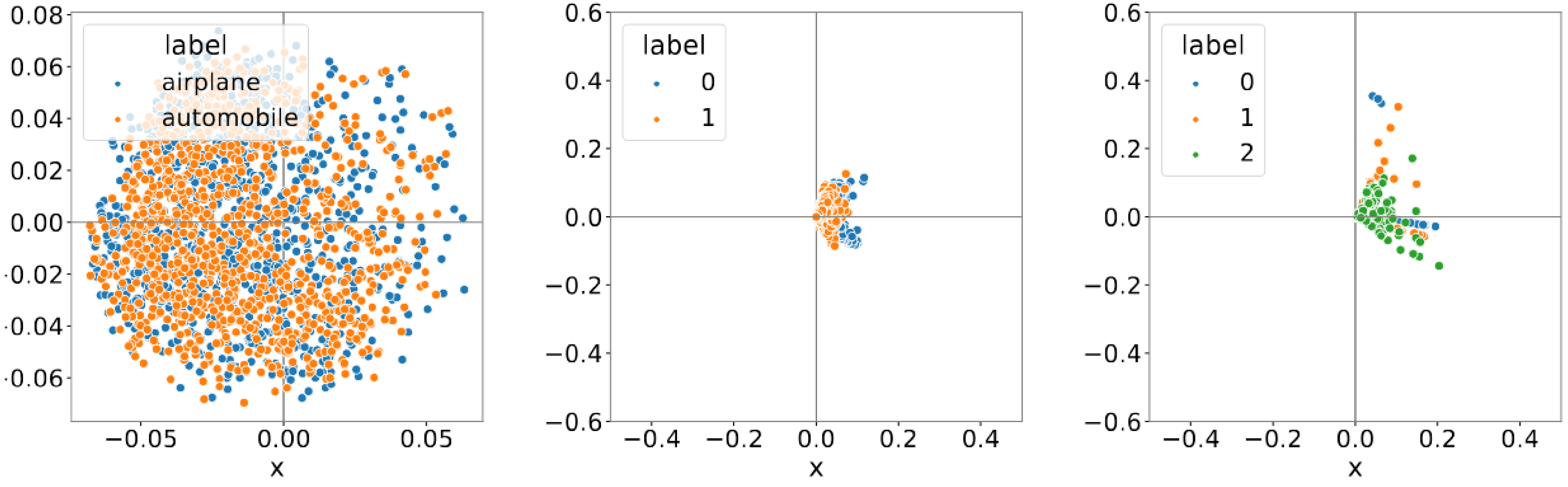
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Ho et al., 2020

3 Graphs vs. Images

The Anisotropy of Graphs

- While images are naturally isotropic and euclidean, Graphs are anisotropic (i. e., properties vary with direction)



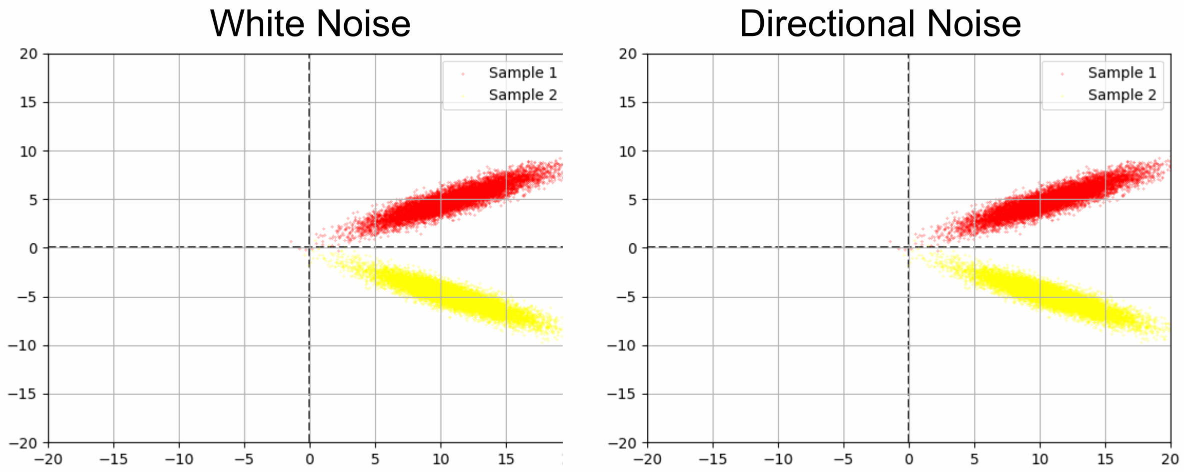
(a) CIFAR-10 (b) Amazon-Photo (c) IMDB-M

(Yang et al., 2023)

White Noise vs. Directional Noise

- Hence, Yang et al. introduce "Directional Noise":

$$x_{t,i} = \sqrt{\bar{\alpha}_t} x_{0,i} + \sqrt{1 - \bar{\alpha}_t}\epsilon',$$
$$\epsilon' = \operatorname{sgn}(x_{0,i}) \odot |\bar{\epsilon}|,$$
$$\bar{\epsilon} = \mu + \sigma \odot \epsilon \quad \text{where } \epsilon \sim \mathcal{N}(0, \mathbf{I})$$



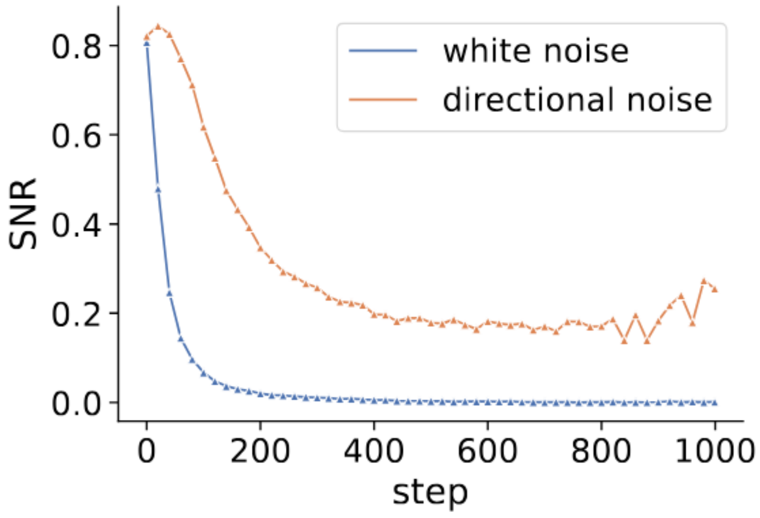
- α and β are the noise schedule analogous to Ho's paper

$$\bar{\alpha}_t := \prod_{i=0}^{t}(1 - \beta_i) \in (0, 1)$$

The effect on the Signal-To-Noise-Ratio

- The Signal-To-Noise-Ratio is fundamental for the learning process of Diffusion Models

- The application of directional noise has a vital effect on the SNR

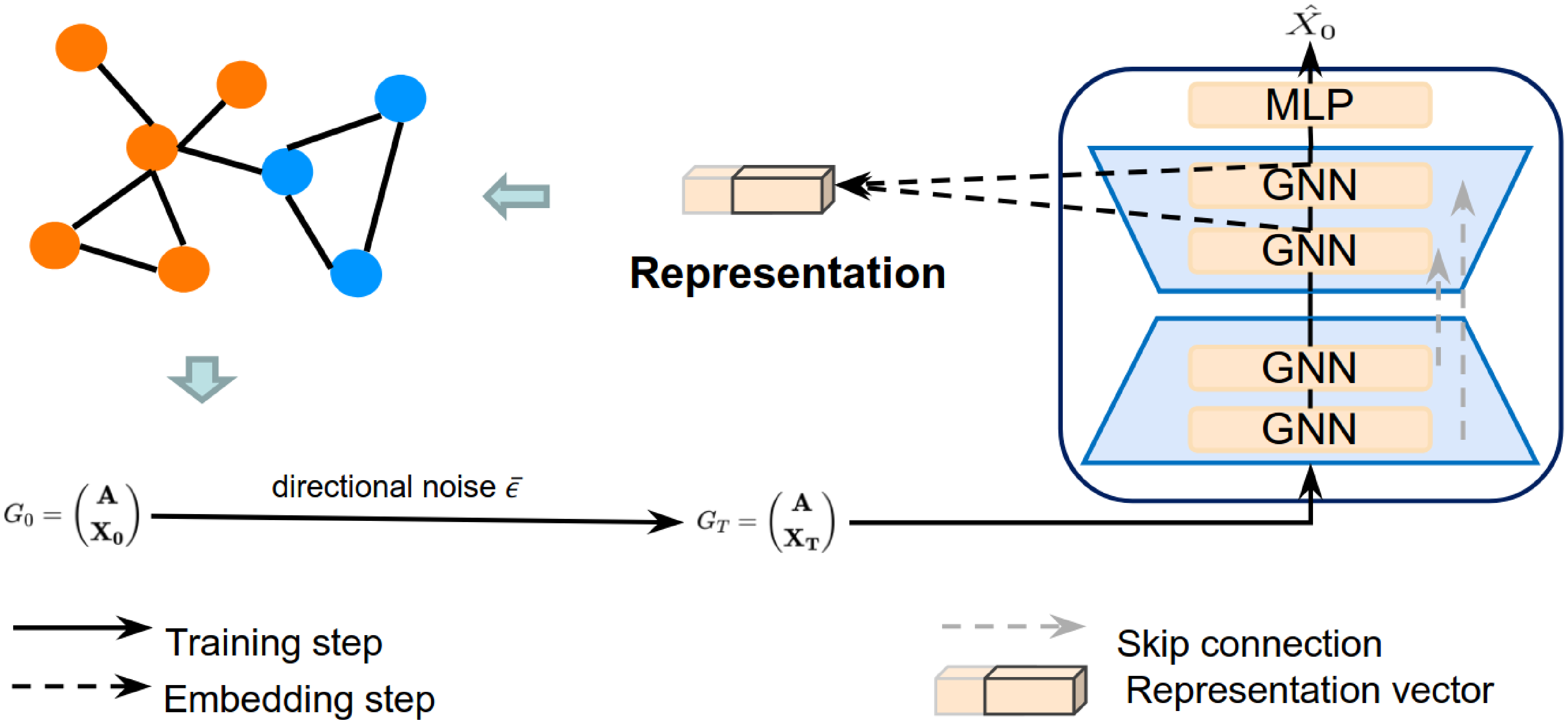


(a) Amazon-Photo

Yang et al., 2023

4 Directional Diffusion Models - Architecture

Components of the Model



Yang et al., 2023

The Training Algorithm

- Similar to the DPDM algorithm

Algorithm 1 The training algorithm.

Input: A batch of graphs $\mathcal{G} = \{G_1, \cdots G_B\}$
Output: The denoising network f_θ

Initialize: the denoising network f_θ
Compute μ, the mean of node features across batch $\mathcal{G}$
Compute σ, the standard deviation of node features across batch $\mathcal{G}$
while not convergence **do**
    for G_i in $\mathcal{G}$ **do**
        for $t = 1, \ldots, T$ **do**
            Sample directional noise ϵ' using equation (2)
            Take gradient descent step on
            $\nabla_\theta \left\| \mathbf{X}_0 - f_\theta(\sqrt{\bar{\alpha}_t}\mathbf{X_i} + \sqrt{1-\bar{\alpha}_t}\epsilon', \mathbf{A}, t) \right\|$
        end for
    end for
end while

Yang et al., 2023

The Extraction Algorithm

- Instead of generating an image, a representation is generated

---

Algorithm 2 Extracting representations.

Input: $G = (\mathbf{A}, \mathbf{X})$, forward step set $\{T_0, T_1, \ldots, T_K\}$, pre-trained denoising network f_θ
Output: $\mathbf{H}$, the representation of G

Compute μ the mean of node features
Compute σ the standard deviation of node features
for k in $\{T_0, T_1, \ldots, T_K\}$ **do**
    Sample directional noise ϵ' using equation (2)
    $\mathbf{X_k} \leftarrow \sqrt{\bar{\alpha}_k}\mathbf{X_0} + \sqrt{1 - \bar{\alpha}_k}\epsilon'$
    $\mathbf{H}_k \leftarrow f_\theta(\mathbf{X_k}, \mathbf{A}, k)$
end for
Concatenate $\mathbf{H} = [\mathbf{H}_{T_0}, \mathbf{H}_{T_1}, \ldots, \mathbf{H}_{T_K}]$
return $\mathbf{H}$

Yang et al., 2023

5 Resulting Benchmarks

Graph Classification

- The paper compares multiple State-Of-The-Art models with DDMs
- SVMs are used on the learned representations
- While here only graph classification results are presented, the results from node classification are similarly promising

Results

Dataset	IMDB-B	IMDB-M	COLLAB	REDDIT-B	PROTEINS	MUTAG	
GIN	75.1±5.1	52.3±2.8	80.2±1.9	92.4±2.5	76.2±2.8	89.4±5.6	supervised
DiffPool	72.6±3.9	-	78.9±2.3	92.1±2.6	75.1±2.3	85.0±10.3	supervised
Infograph	73.03±0.87	49.69±0.53	70.65±1.13	82.50±1.42	74.44±0.31	89.01±1.13	unsupervised
GraphCL	71.14±0.44	48.58±0.67	71.36±1.15	89.53±0.84	74.39±0.45	86.80±1.34	unsupervised
JOAO	70.21±3.08	49.20±0.77	69.50±0.36	85.29±1.35	74.55±0.41	87.35±1.02	unsupervised
GCC	72	49.4	78.9	89.8	-	-	unsupervised
MVGRL	74.20±0.70	51.20±0.50	-	84.50±0.60	-	89.70±1.10	unsupervised
GraphMAE	75.52±0.66	51.63±0.52	80.32±0.46	88.01±0.19	75.30±0.39	88.19±1.26	unsupervised
DDM	**76.40±0.22**	**52.53±0.31**	**81.72±0.31**	89.15 ±1.3	**75.47 ±0.50**	**91.51 ±1.45**	unsupervised

Yang et al., 2023

6 Conclusion

Research Outlook

- Yang et al. only introduce the idea, they admit that their hyperparameters are not optimal yet
- One open question is how the optimal set of diffusion steps can be determined
- Variants of DDMs could bring value to areas such as computer vision and natural language processing
- To the best of my knowledge, nobody has yet continued their work

What makes this paper special?

- As mentioned, the technology introduced holds great potential for future research
- The benchmarks are remarkable, particularly compared to the supervised models
- The researchers consider themselves „among the pioneers in the literature“ regarding the „exploration of anisotropic structure in graph data“

References & Weblinks

- Yang et al. (2023). Directional diffusion models for graph representation learning
- Zhu et al. (2020). GSSNN: Graph Smoothing Splines Neural Networks
- Yanardag et al. (2015). Deep Graph Kernels
- Nguyen et al. (2019). Universal Graph Transformer Self-Attention Networks
- Ho et al. (2020). Denoising Diffusion Probabilistic Model
- Dhariwal et al. (2021). Diffusion Models Beat GANs on Image Synthesis

- Presentation Code: https://github.com/JavaLangMarlon/ddm-proseminar-tu-dortmund