Score-based Generative Modeling of Graphs via the System of SDEs

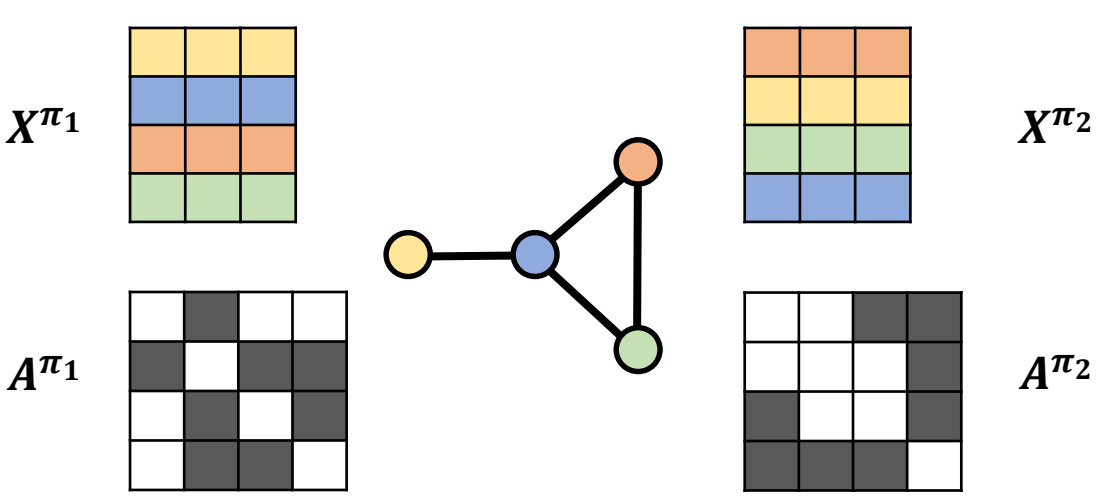
Jaehyeong Jo[1*], Seul Lee[1*], Sung Ju Hwang[1,2]

(*: equal contribution)

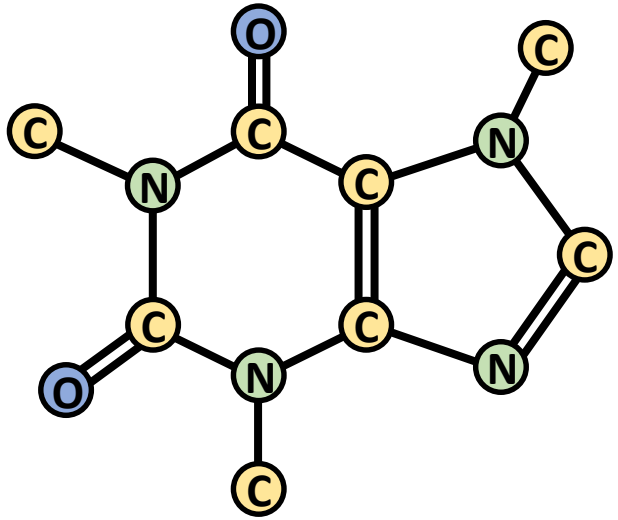
KAIST[1], AITRICS[2], South Korea

Challenge of Graph Generation

Graphs have **non-unique representations** as the order of the nodes are not fixed, and **complex dependency between nodes and edges.**



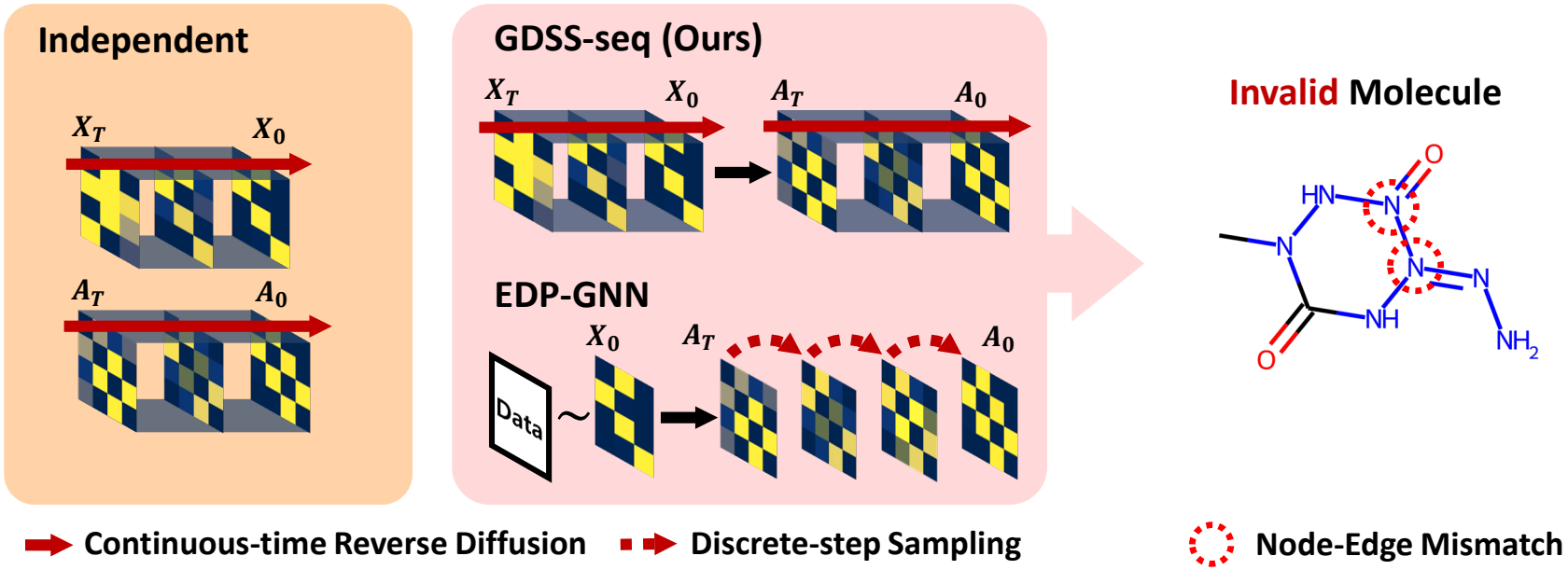
(a) Permutation Invariance



(b) Node-edge dependency

Naïve Extension of Score-Based Models to Graphs

Unlike images, **graphs consist of nodes and edges** which are interdependent.

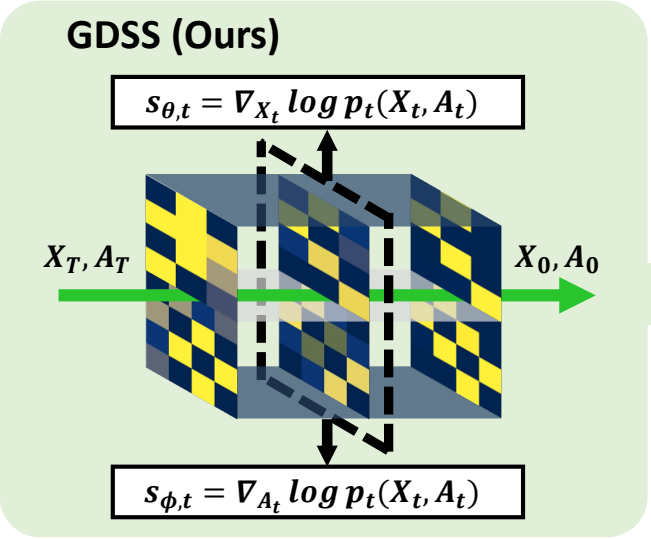


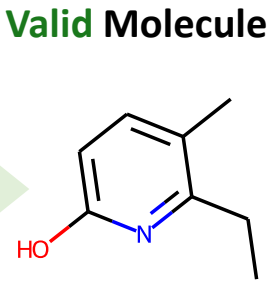
Straight forward extension of [Song et al. 21] **fails to capture the complex relations** between the nodes and edges.

[Song et al. 21] Score-Based Generative Modeling Through Stochastic Differential Equations, ICLR 2021.

Graph Diffusion Via the System of SDEs (GDSS)

We propose a **novel diffusion process for graphs** via the system of SDEs, that can model the complex dependencies between nodes and edges.





Reverse-time Diffusion Process

$$\begin{cases} \mathrm{d}\boldsymbol{X}_t = \left[\mathbf{f}_{1,t}(\boldsymbol{X}_t) - g_{1,t}^2 \nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)\right] \mathrm{d}\bar{t} + g_{1,t} \mathrm{d}\bar{\mathbf{w}}_1 \\ \mathrm{d}\boldsymbol{A}_t = \left[\mathbf{f}_{2,t}(\boldsymbol{A}_t) - g_{2,t}^2 \nabla_{\boldsymbol{A}_t} \log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t)\right] \mathrm{d}\bar{t} + g_{2,t} \mathrm{d}\bar{\mathbf{w}}_2 \end{cases}$$

Since GDSS is able to model the dependency, in contrast to EDP-GNN, we can **generate both the node features and adjacency matrices**.

Estimating the Partial Scores

Estimating the partial score functions are not equivalent to estimating the score function, since former **requires capturing the dependency** between $\boldsymbol{X}$ and $\boldsymbol{A}$.

$$\nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{X}_t, \boldsymbol{A}_t) \quad \textbf{vs} \quad \nabla_{\boldsymbol{X}_t} \log p_t(\boldsymbol{X}_t)$$

Partial Score **Original Score**

Therefore, we derive **new training objectives** for estimating the partial scores:

$$\min_{\theta} \mathbb{E}_t \left\{ \lambda_1(t)\, \mathbb{E}_{\boldsymbol{G}_0} \mathbb{E}_{\boldsymbol{G}_t|\boldsymbol{G}_0} \| \boldsymbol{s}_{\theta,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{X}_t} \log p_{0t}(\boldsymbol{X}_t|\boldsymbol{X}_0) \|_2^2 \right\}$$

$$\min_{\phi} \mathbb{E}_t \left\{ \lambda_2(t)\, \mathbb{E}_{\boldsymbol{G}_0} \mathbb{E}_{\boldsymbol{G}_t|\boldsymbol{G}_0} \| \boldsymbol{s}_{\phi,t}(\boldsymbol{G}_t) - \nabla_{\boldsymbol{A}_t} \log p_{0t}(\boldsymbol{A}_t|\boldsymbol{A}_0) \|_2^2 \right\}$$

Generic Graph Generation

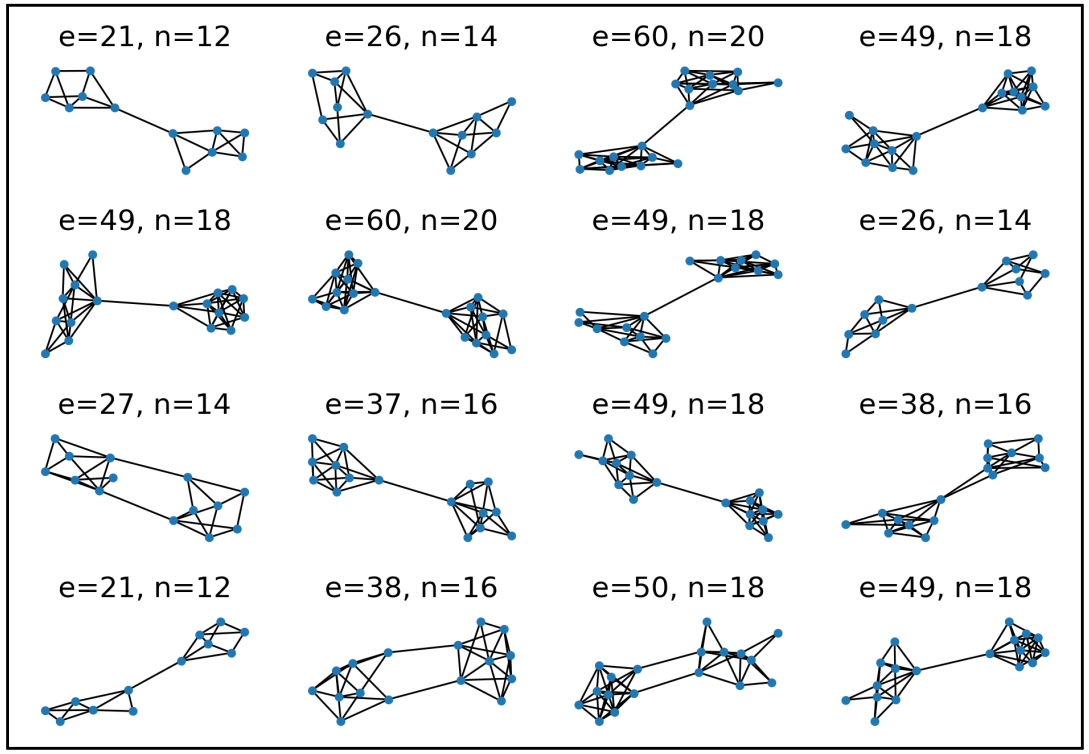
GDSS significantly outperforms the one-shot generative models including EDP-GNN, and also outperforms the autoregressive generative models.

	Ego-small				Community-small				Enzymes			
	Deg.	Clus.	Orbit	Avg.	Deg.	Clus.	Orbit	Avg.	Deg.	Clus.	Orbit	Avg.
DeepGMG	0.040	0.100	0.020	0.053	0.220	0.950	0.400	0.523	-	-	-	-
GraphRNN	0.090	0.220	0.003	0.104	0.080	0.120	0.040	0.080	**0.017**	0.062	0.046	0.042
GraphAF	0.03	0.11	**0.001**	0.047	0.18	0.20	0.02	0.133	1.669	1.283	0.266	1.073
GraphDF	0.04	0.13	0.01	0.060	0.06	0.12	0.03	0.070	1.503	1.061	0.202	0.922
GraphVAE	0.130	0.170	0.050	0.117	0.350	0.980	0.540	0.623	1.369	0.629	0.191	0.730
GNF	0.030	0.100	**0.001**	0.044	0.200	0.200	0.110	0.170	-	-	-	-
EDP-GNN	0.052	0.093	0.007	0.051	0.053	0.144	0.026	0.074	0.023	0.268	0.082	0.124
GDSS-seq (Ours)	0.032	0.027	0.011	0.023	0.090	0.123	**0.007**	0.073	0.099	0.225	0.010	0.111
GDSS (Ours)	**0.021**	**0.024**	0.007	**0.017**	**0.045**	**0.086**	**0.007**	**0.046**	0.026	**0.061**	**0.009**	**0.032**

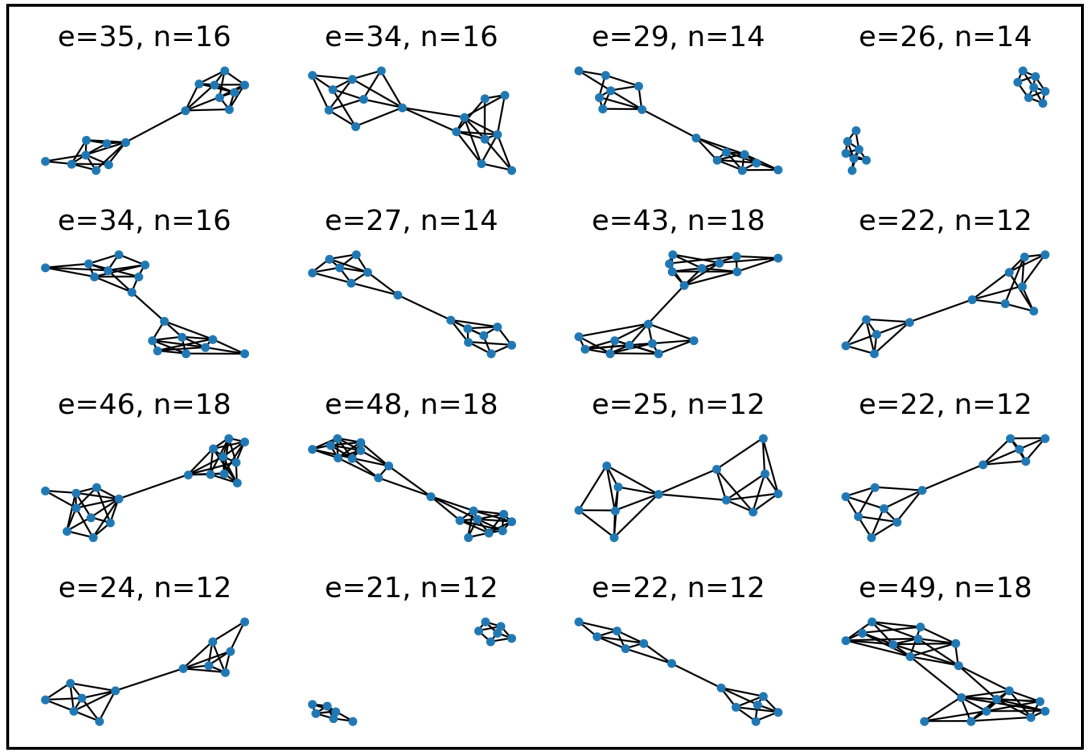
Table: Generation results on the generic graph datasets.

Generic Graph Generation

GDSS is able to generate the **community structures** represented in graphs.



(a) Training Data



(b) GDSS (Ours)

Figure: Visualization of the graphs from the training set and the generated graphs.

Molecule Generation

GDSS achieves incomparably high validity without valency correction, and further significantly outperforms the baselines in NSPDK MMD and FCD.

		QM9			ZINC250k		
	Method	Val. w/o corr.↑	NSPDK↓	FCD↓	Val. w/o corr.↑	NSPDK↓	FCD↓
Autoreg.	GraphAF	67	0.020	5.268	68	0.044	16.289
	GraphAF+FC	74.43	0.021	5.625	68.47	0.044	16.023
	GraphDF	82.67	0.063	10.816	89.03	0.176	34.202
	GraphDF+FC	93.88	0.064	10.928	90.61	0.177	33.546
One-shot	MoFlow	91.36	0.017	4.467	63.11	0.046	20.931
	EDP-GNN	47.52	0.005	**2.680**	82.97	0.049	16.737
	GraphEBM	8.22	0.030	6.143	5.29	0.212	35.471
	GDSS-seq (Ours)	94.47	0.010	4.004	92.39	0.030	16.847
	GDSS (Ours)	**95.72**	**0.003**	2.900	**97.01**	**0.019**	**14.656**

Table: Generation results on the **QM9** and **ZINC250k** datasets.

Molecule Generation

GDSS is able to **generate valid molecules by capturing the node-edge relationship**, modeled through time via the system of SDEs.

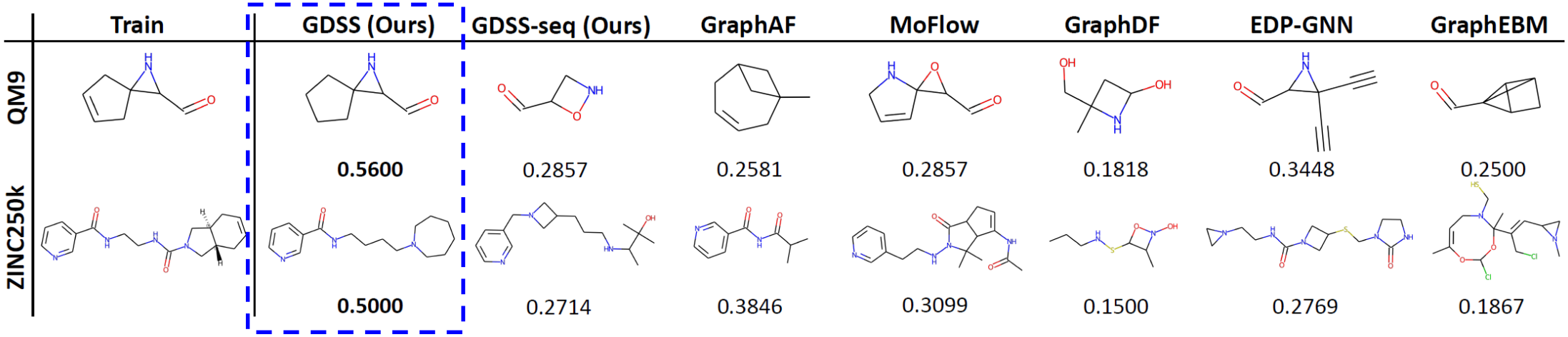


Figure: Visualization of the generated molecules and the similarity score.

Generated molecules of GDSS share a large substructure with the molecules in the training set, whereas the baselines fail to do so.

Conclusion

- We propose a novel **score-based generative model for graphs** that overcomes the limitation of previous generative methods, by introducing a **diffusion process for graphs** that can generate node features and adjacency simultaneously via the system of SDEs.

- We derive **novel training objectives** to estimate the gradient of the joint log-density for the proposed diffusion process and further introduce an **efficient integrator** to solve the proposed system of SDEs.

- We validate our method on both **synthetic and real-world graph generation tasks**, on which ours outperforms existing graph generative models.

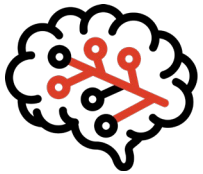
Thank you

Contact information:
Jaehyeong Jo

 harryjo97@kaist.ac.kr

 Jaehyeong_Jo

KAIST



MLAI
Machine Learning & Artificial Intelligence