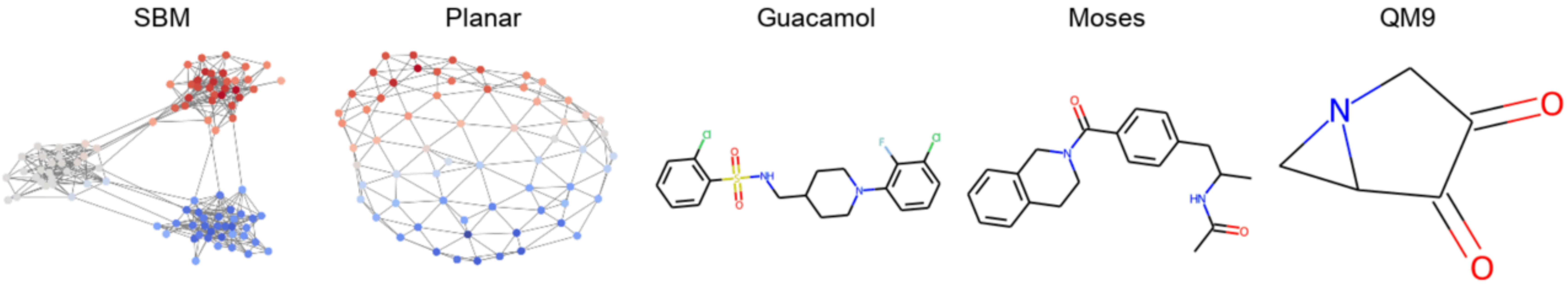
DiGress: Discrete denoising diffusion for graph generation

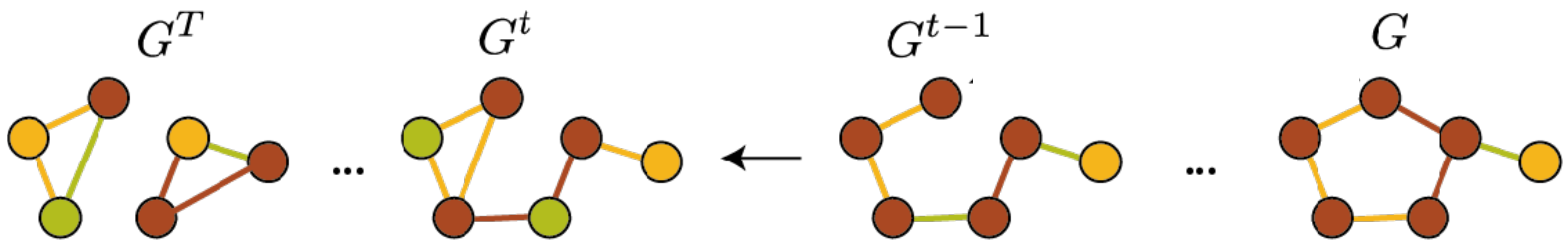
Clément Vignac

Joint work with Igor Krawczuk (co-1st author), Antoine Siraudin, Bohan Wang, Volkan Cevher & Pascal Frossard (EPFL)



ICLR 2023

DIGRESS: DIFFUSION ON A DISCRETE SPACE



- Motivation for discrete diffusion: no need to predict continuous values that do not exist in the data + do not break sparsity

- Adding noise = sampling node or edge types from a categorical distribution

- No edge = one particular edge type

The noise is sampled independently on each node and edge

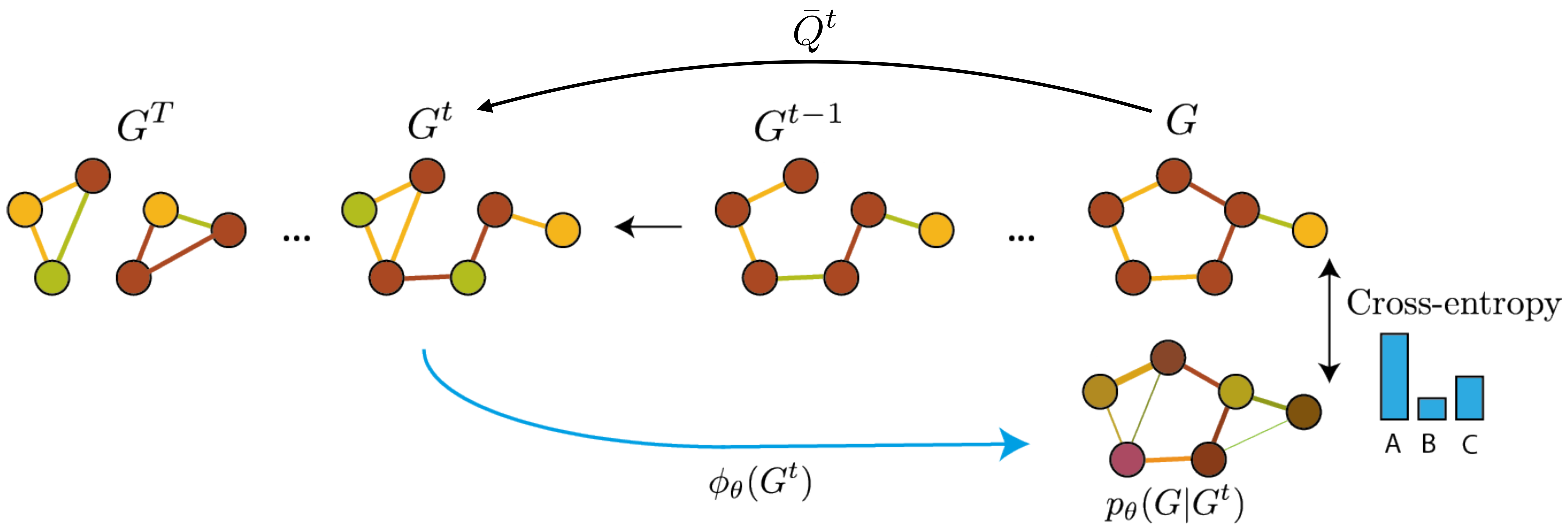
$\mathcal{X}$: space of node types (cardinality a)

$\mathcal{E}$: space of edge types (cardinality b)

$$Q_t^X : a \times a, \quad Q_t^E : b \times b$$

$$XQ^t = \begin{bmatrix} x_1^T Q^t \\ \cdots \\ x_n^T Q^t \end{bmatrix}$$ batch operation

$n \times a$ matrix of probabilities

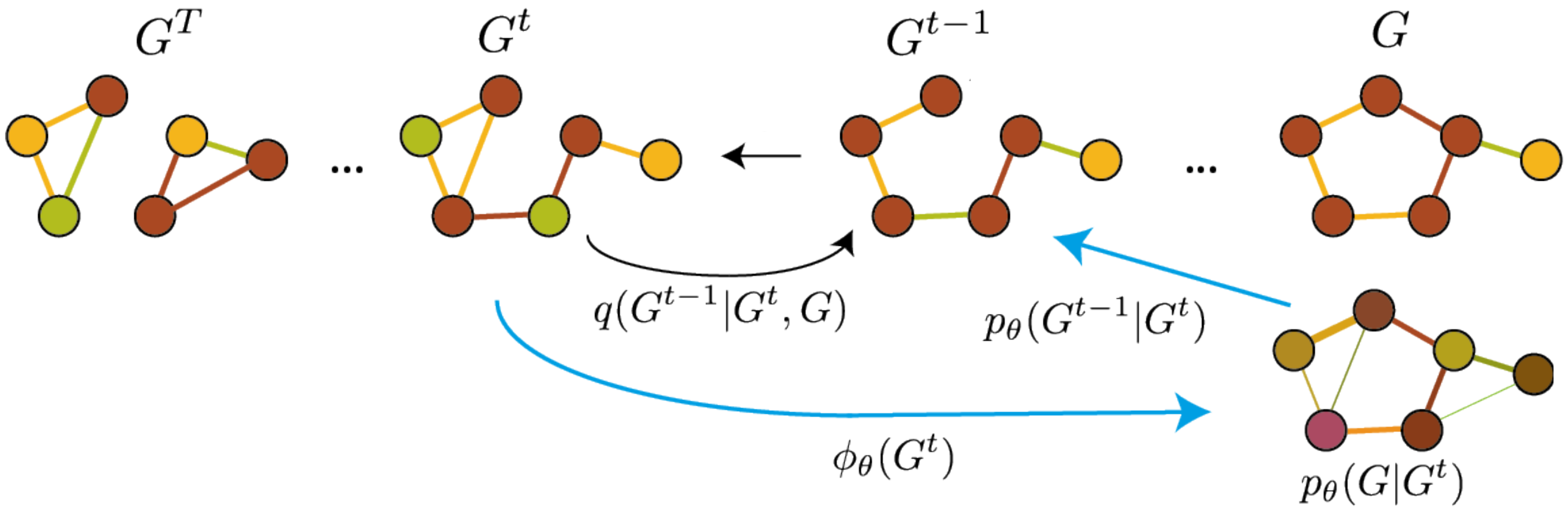


Use a graph transformer network

Algorithm : Training DiGress

Input: A graph $G = (\boldsymbol{X}, \boldsymbol{E})$
Sample $t \sim \mathcal{U}(1, ..., T)$
Sample $G^t \sim \boldsymbol{X}\bar{\boldsymbol{Q}}^t_X \times \mathbf{E}\bar{\boldsymbol{Q}}^t_E$
$\hat{p}^X, \hat{p}^E \leftarrow \phi_\theta(G_t, t)$
$loss \leftarrow l_{CE}(\hat{p}^X, \boldsymbol{X}) + \lambda\, l_{CE}(\hat{p}^E, \mathbf{E})$
optimizer. step($loss$)

Graph generation = sequence of node and edge classification tasks

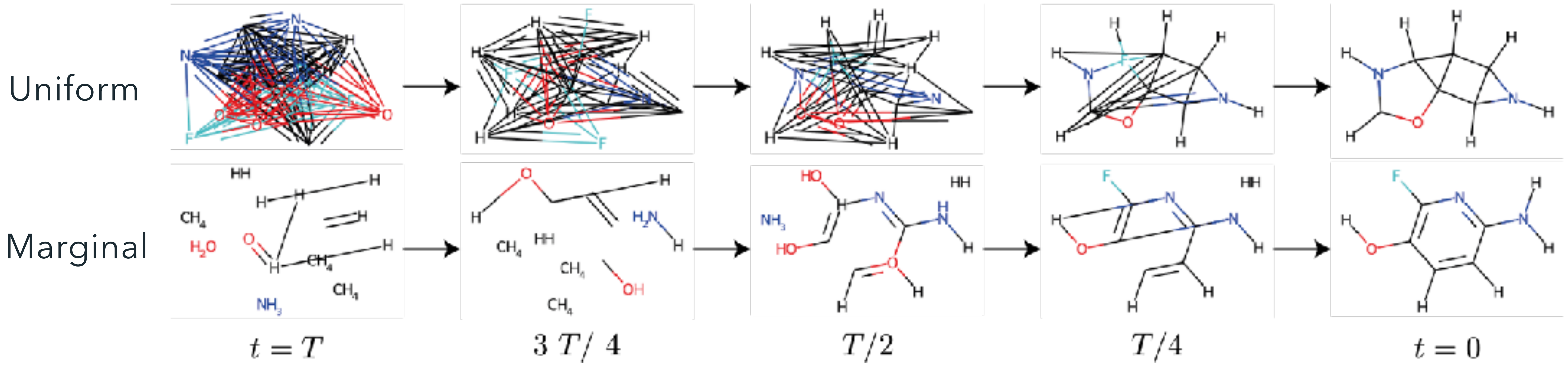


Algorithm : Sampling from DiGress

Sample n from the training data distribution
Sample $G^T \sim q_X(n) \times q_E(n)$
for t = T **to** *1* **do**
  $\hat{p}^X, \hat{p}^E \leftarrow \phi_\theta(G^t, t)$
  Sample $G^{t-1} \sim \prod_i p_\theta(x_i^{t-1}|G^t) \times \prod_{ij} p_\theta(e_{ij}^{t-1}|G^t)$
end
return G^0

$$p_\theta(x_i^{t-1} \,|\, G^t) = \int_{x_i} p_\theta(x_i^{t-1} \,|\, x_i, G^t)\ dp_\theta(x_i \,|\, G^t)$$
$$= \sum_{x \in \mathcal{X}} q(x_i^{t-1} \,|\, x_i = x, x_i^t)\ \hat{p}_i^X(x)$$

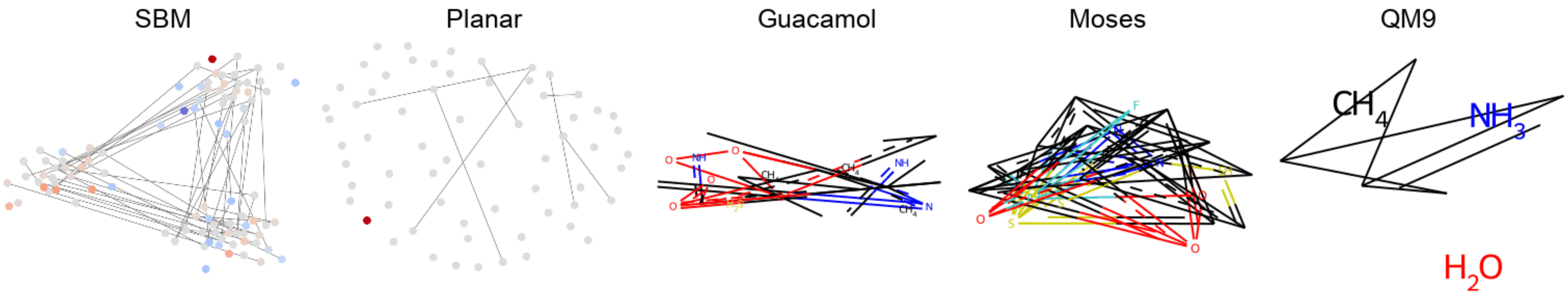
- Better noise model



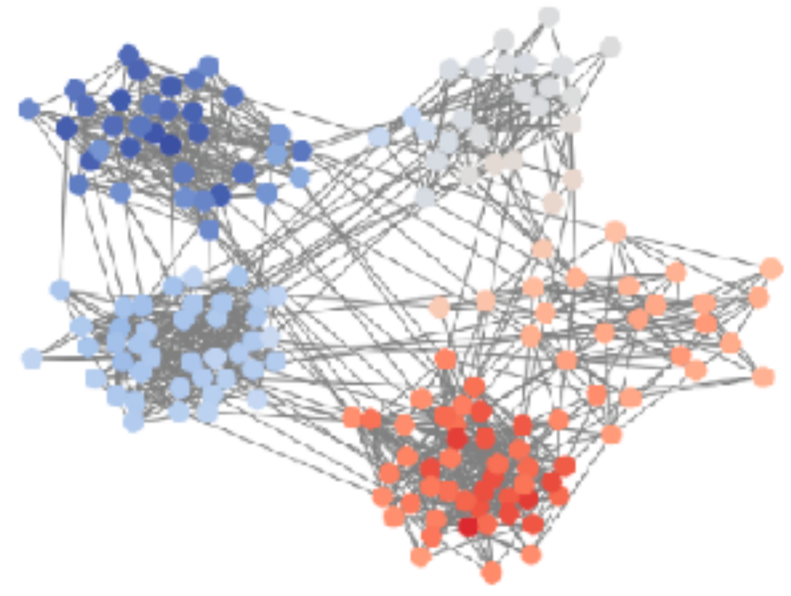
- Additional features for improving the graph Transformer expressivity

- Regression guidance for conditioning on graph-level properties

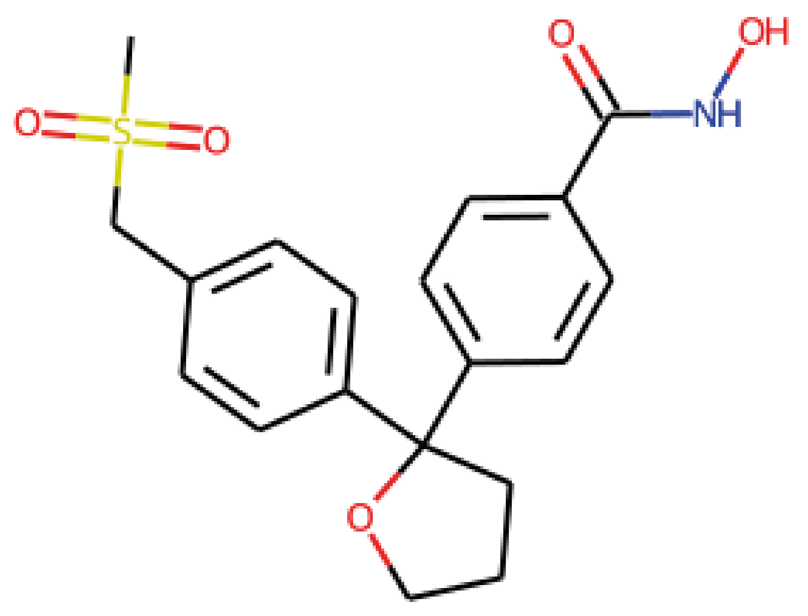
Results



RESULTS

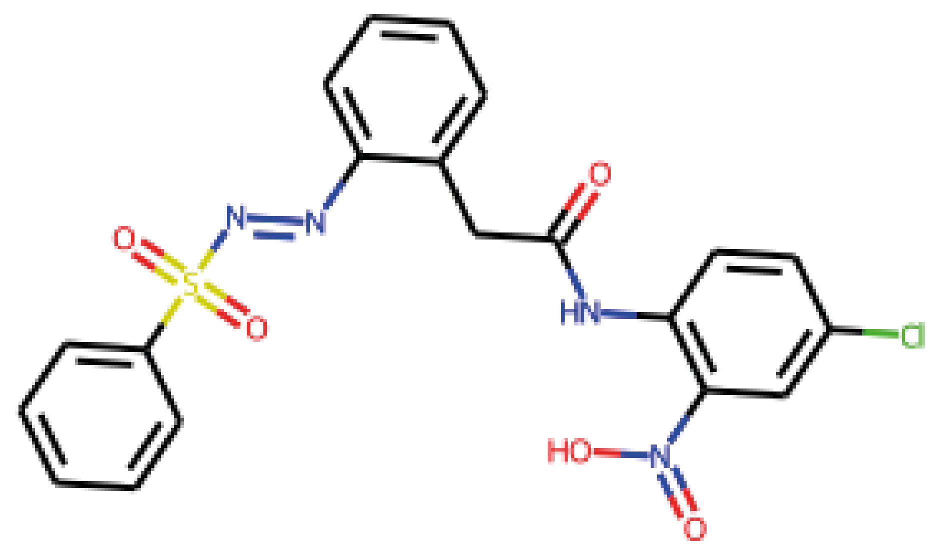


- We achieve state-of-the-art performance on Planar graphs, SBM graphs, QM9, MOSES, GuacaMol across graph based method that operate at the node level.



- On small graphs, Gaussian and discrete diffusion models achieve similar performance, but DiGress is much faster to train (1 hour vs 7 hour on QM9)

- On larger graphs, DiGress clearly outperform our Gaussian based diffusion model



- First non autoregressive model to scale to the Guacamol dataset

Summary

- DiGress solves graph generation as a sequence of node and edge classification task
- Diffusion models for graphs significantly outperform previous methods – opens the way to many applications
- Discrete diffusion helps scaling to large graphs
- Limitation: $O(n^2)$ complexity