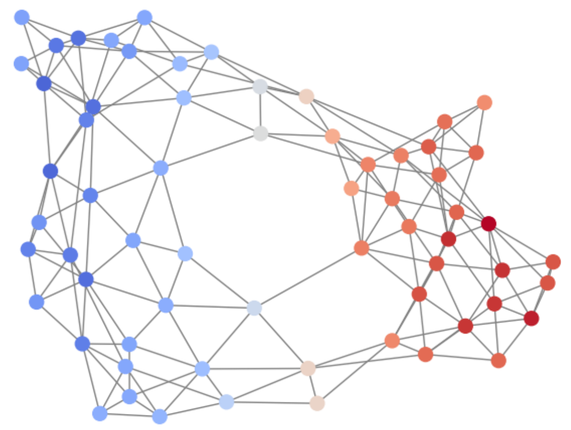
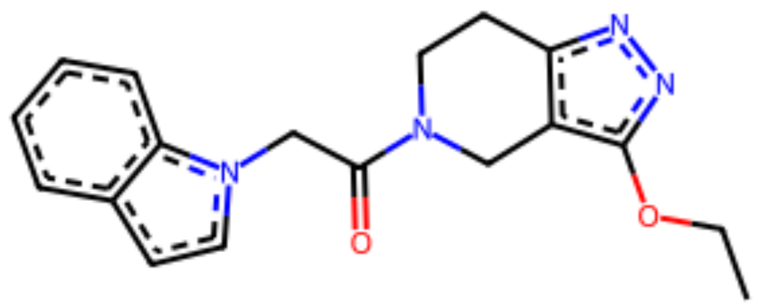
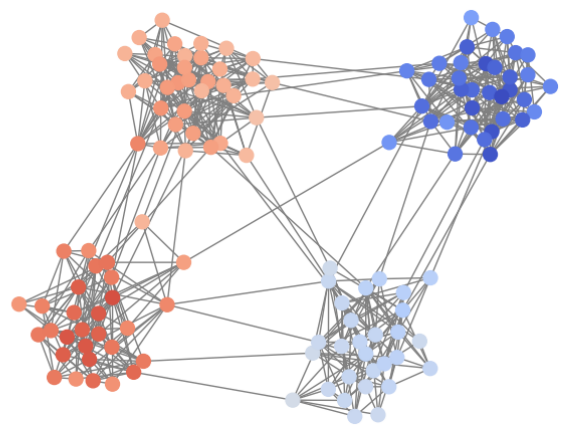
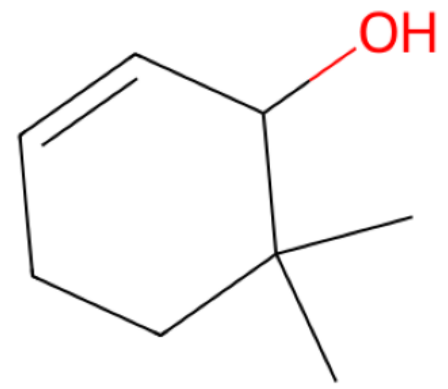




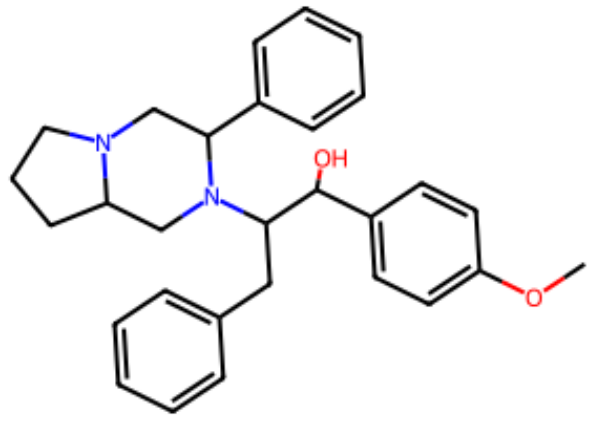


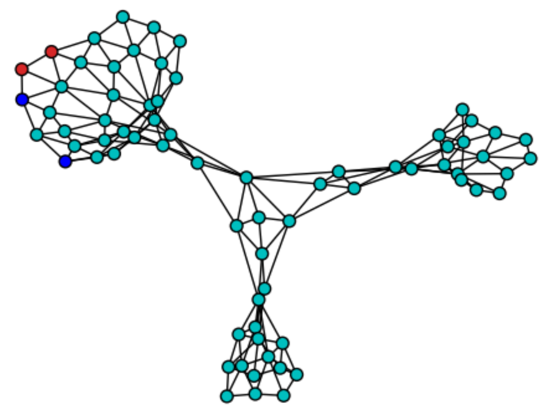


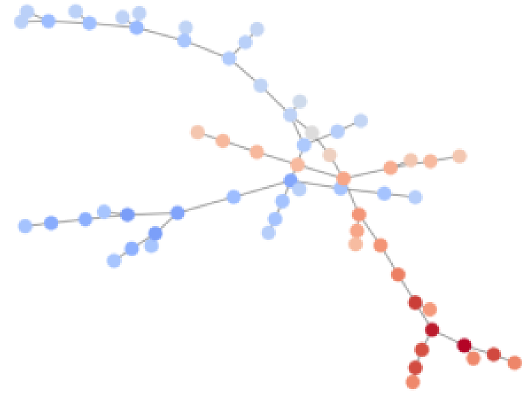
DeFoG: Discrete Flow Matching for Graph Generation

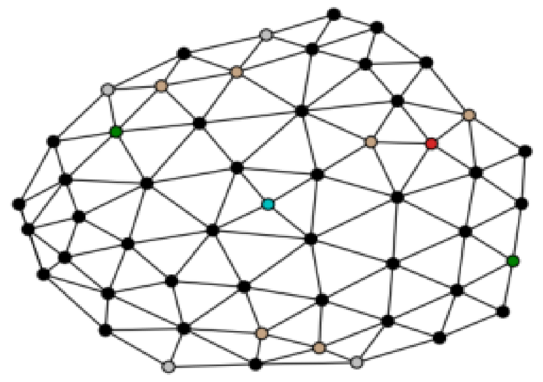
Yiming Qin*, Manuel Madeira*, Dorina Thanou, Pascal Frossard

ICML 2025



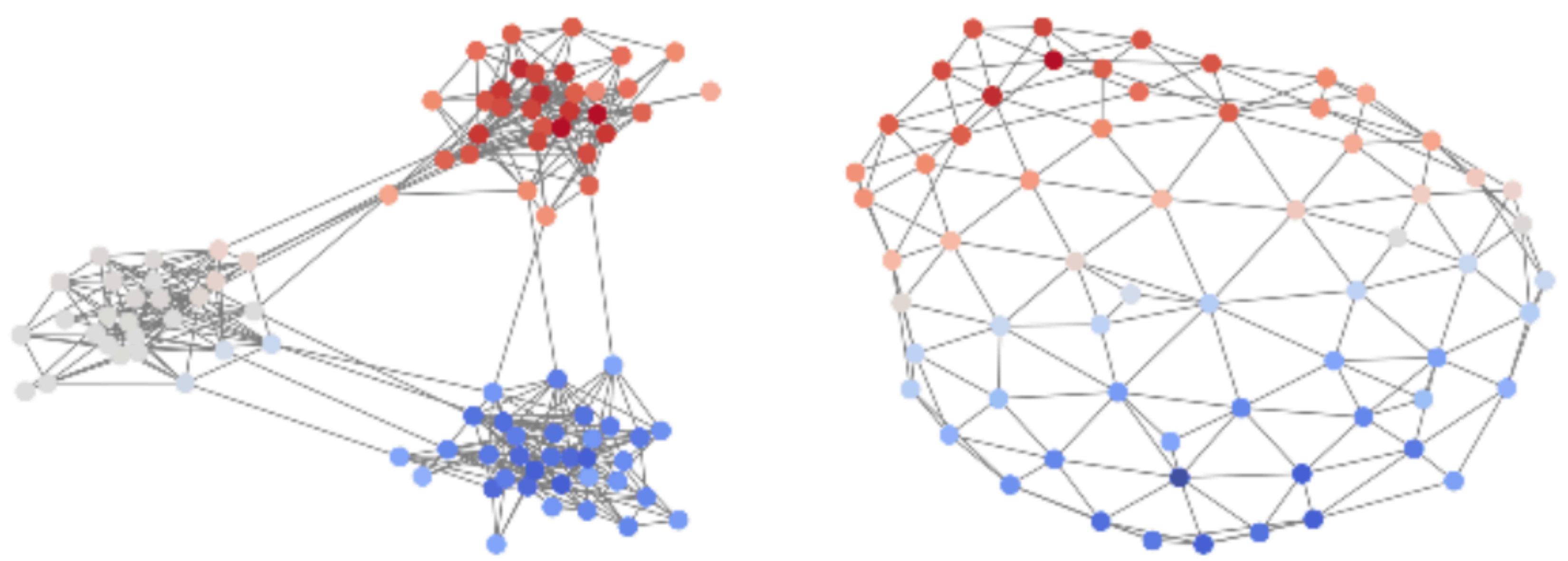




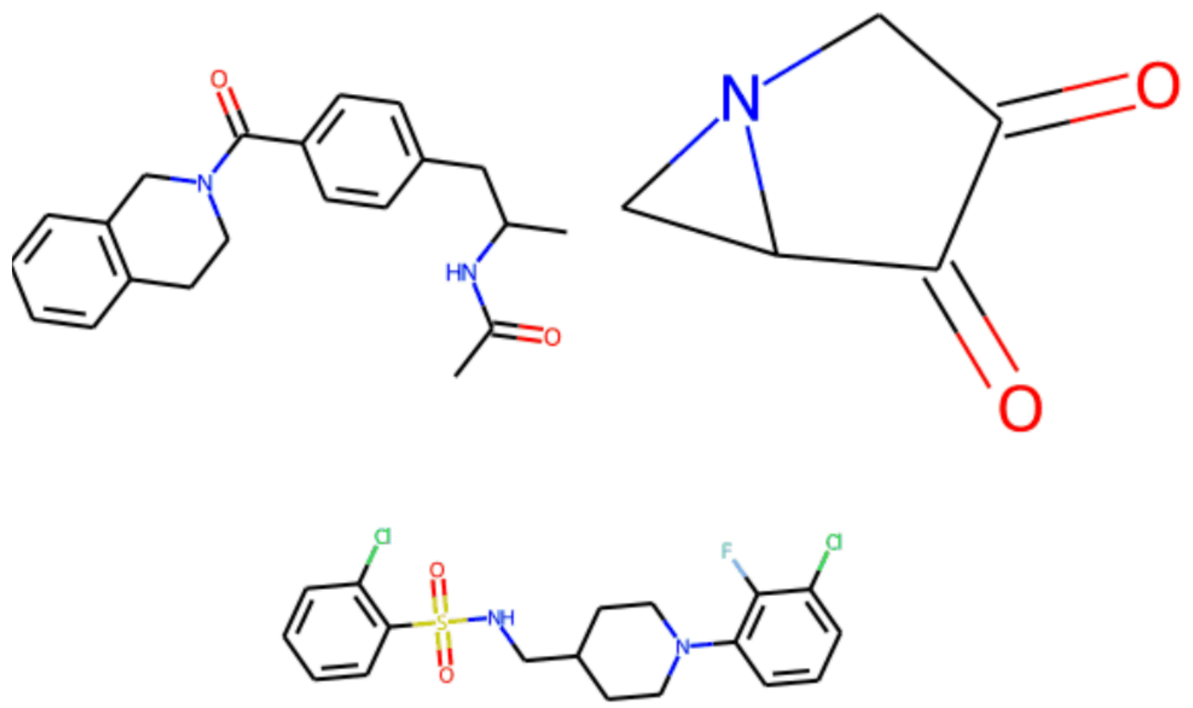


Graph generation

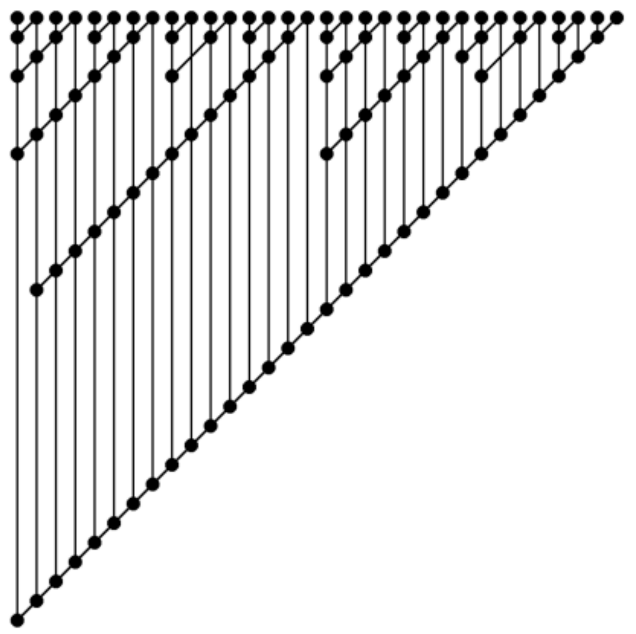
Objective: sample graphs that *resemble* those in the dataset



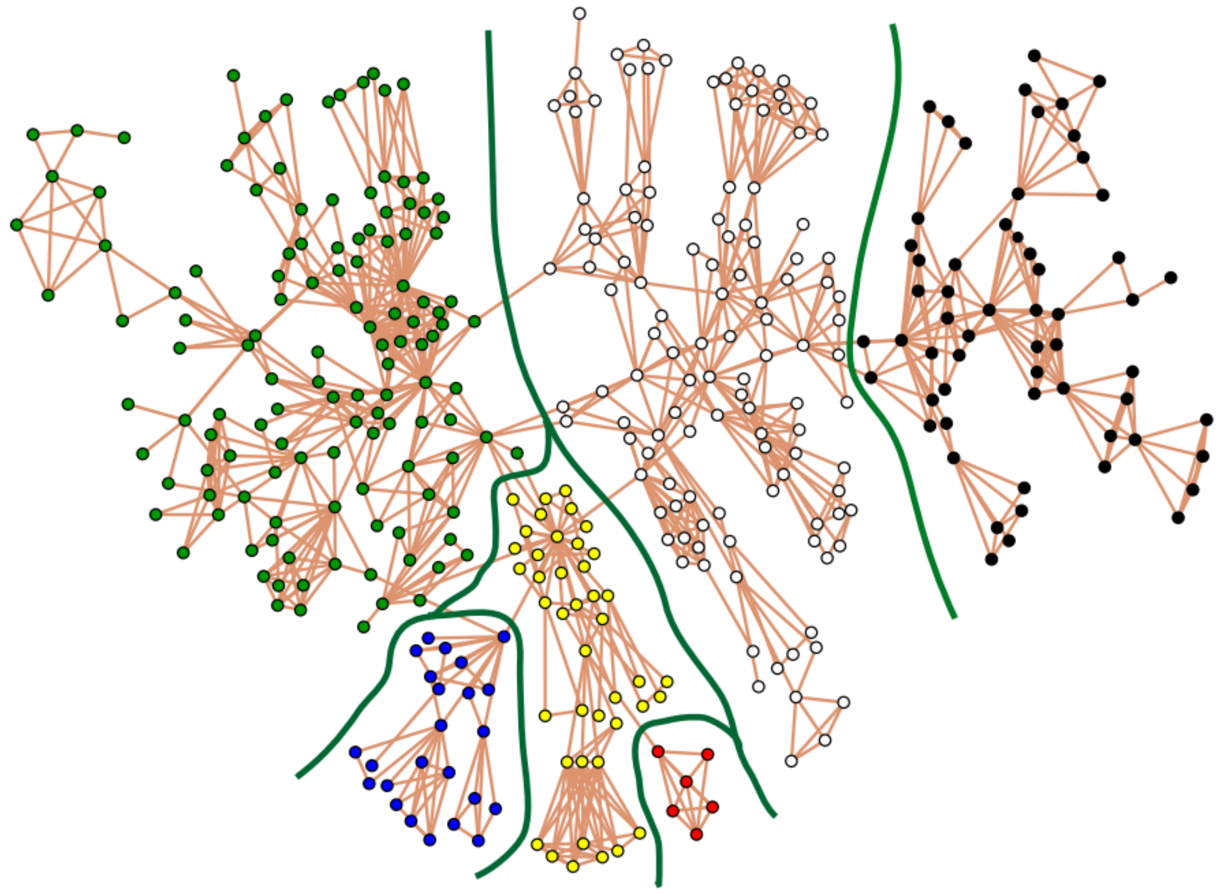
Molecular Generation



Circuit Design



Network Modeling

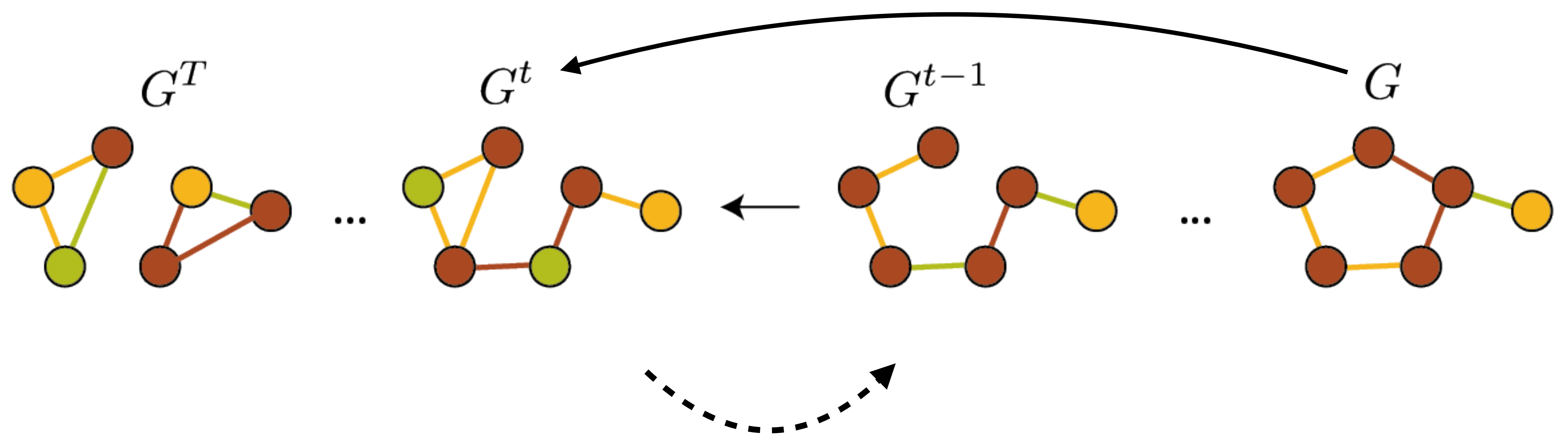


CircuitVAE: Efficient and scalable latent circuit optimization, Song et al., IEEE Design Automation Conference 2024
Statistical Properties of Community Structure in Large Social and Information Networks, Leskovec et al., WWW 2008

Graph discrete diffusion models

Noising:

Sampling node and edge classes from categorical distributions



Denoising: $p_\theta(G_{t-1}|G_t)$

DiGress: Discrete Denoising diffusion for graph generation, Vignac et al., ICLR 2023

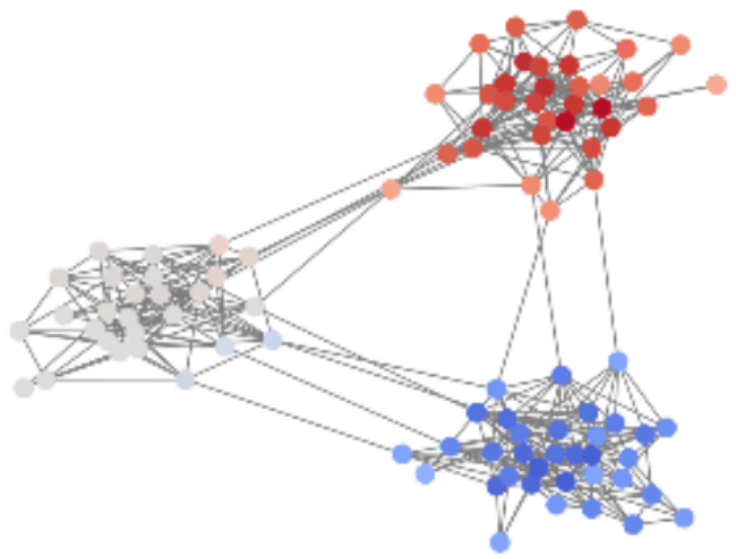
However…

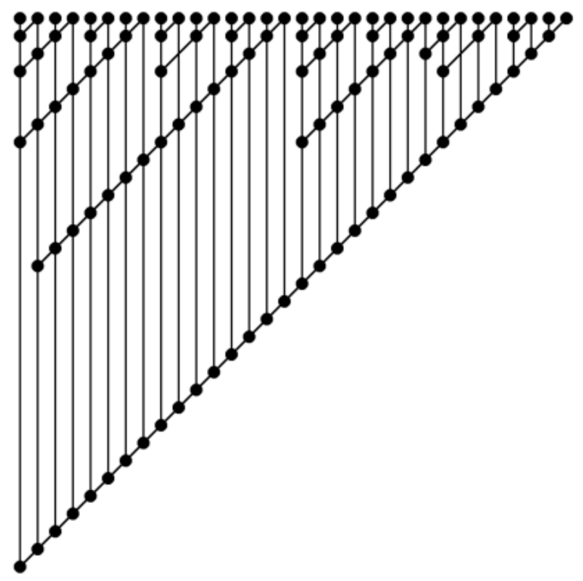
Graph diffusion models are costly to fine-tune:

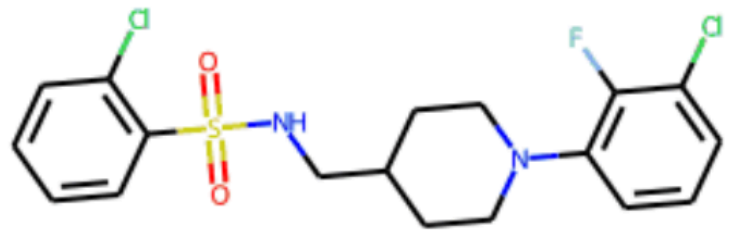
- Choices during training constrain sampling
- Hyperparameter (e.g., noise schedule) tuning requires retraining

One recipe for all graph datasets

Graph datasets are of very diverse nature







Discrete flow matching

Discrete flow matching (DFM) generalizes discrete diffusion models:

- Higher flexibility
- Improved performance on text and images

Our contribution:

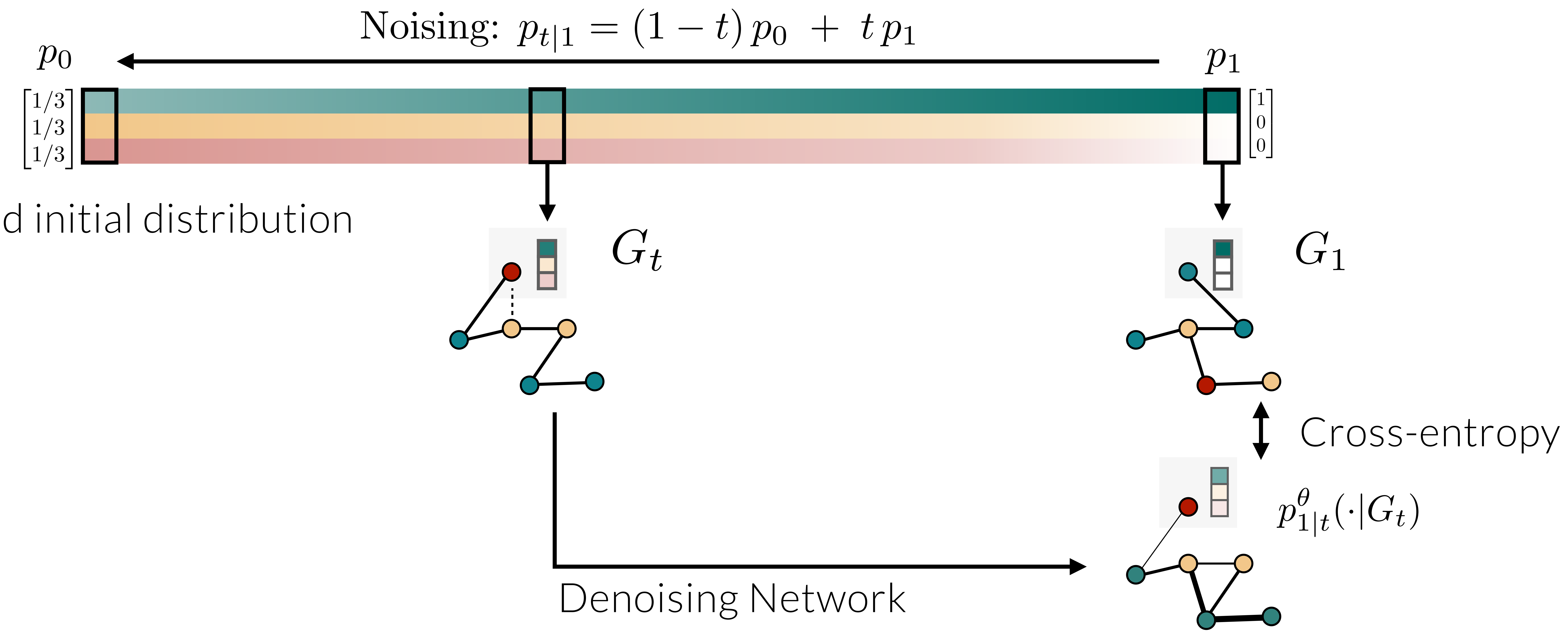
We extend DFM to graph generation and achieve SOTA results

Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design, Campbell et al., ICML 2024
Discrete Flow Matching, Gat et al., NeurIPS 2024

DeFoG: Applying DFM to graphs

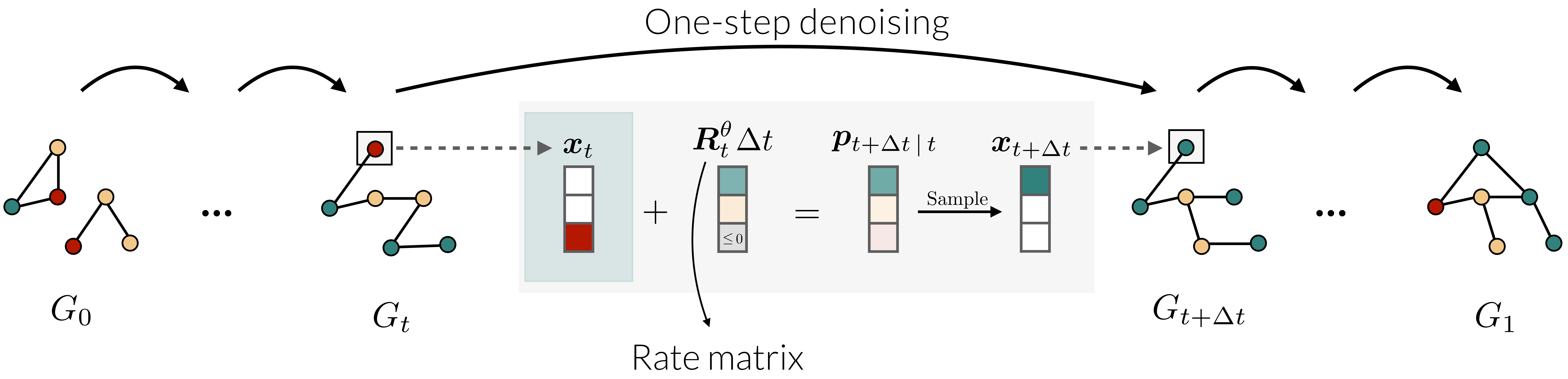
Noising process:

Linear interpolation between data distribution p_1 and initial distribution p_0



DeFoG: Applying DFM to graphs

Denoising process:
Iteratively jump from G_t to $G_{t+\Delta t}$



Making DFM effective for graph generation

Vanilla DFM works on par with existing graph diffusion models

We leverage the increased flexibility of DFM:

$$\boldsymbol{p}_{t+\Delta t|t} = \boldsymbol{x}_t + \underbrace{\boldsymbol{R}_t^\theta}_{} \underbrace{\Delta t}_{}$$

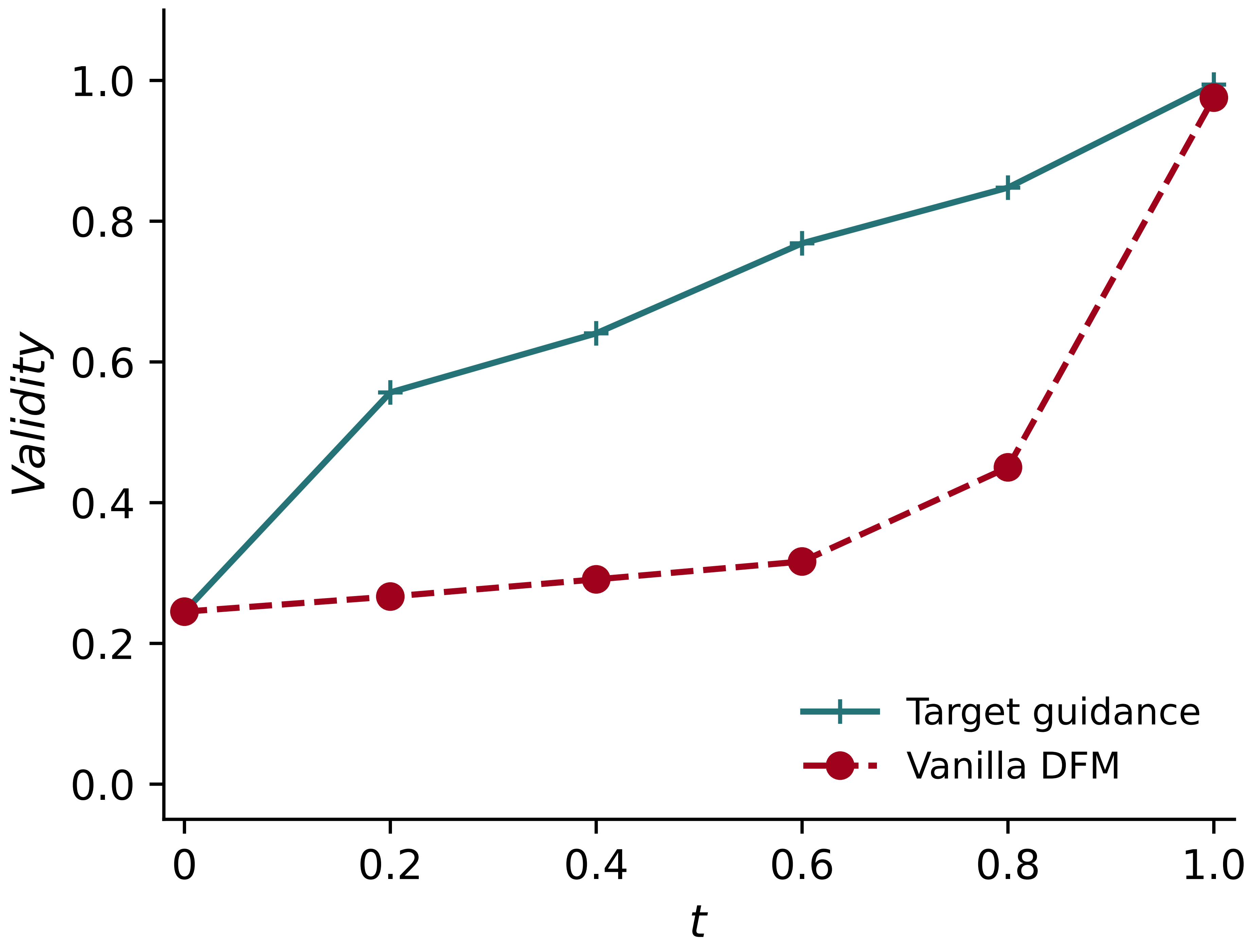
Rate matrix modifiers

- *Target guidance*: amplifies predicted distribution
- *Stochasticity*: control trajectory randomness

Time distortion

- *Sampling distortion*: adjusts denoising time steps

Target guidance

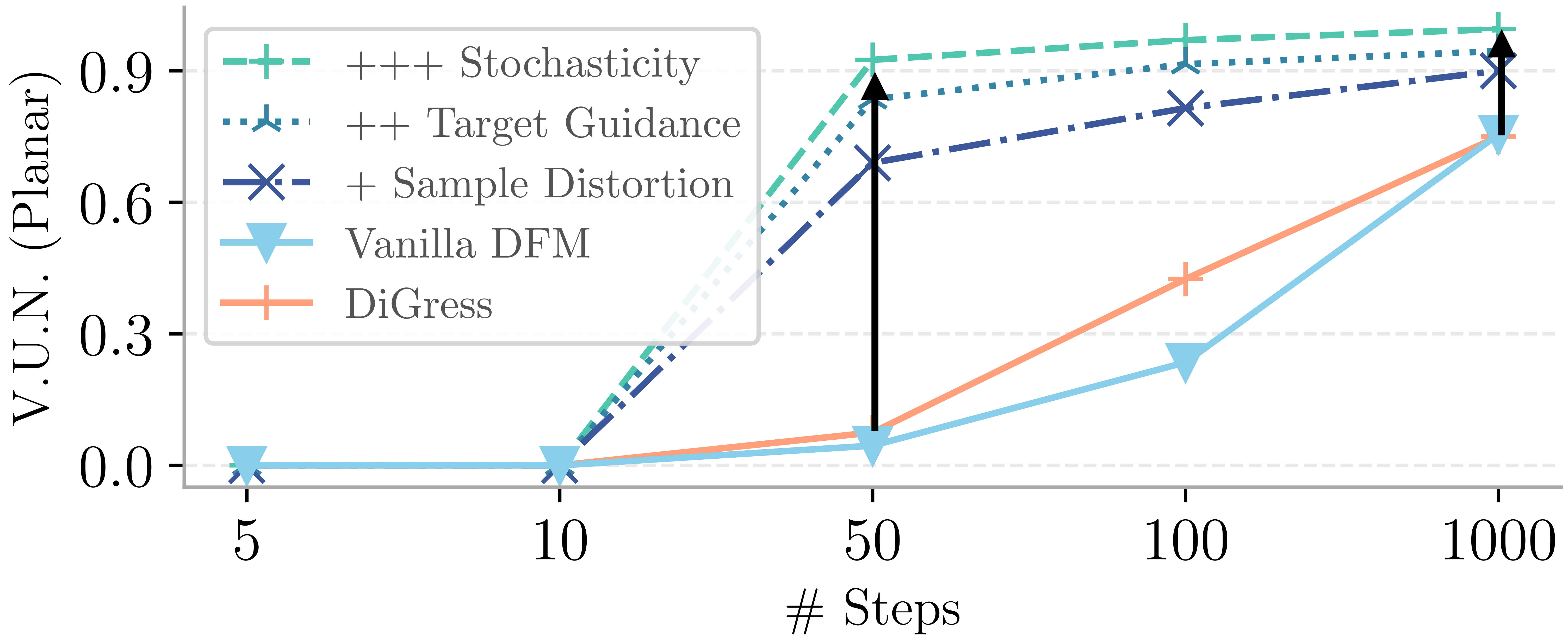


Target guidance: $R_t^* \boxed{+\,\omega\, R_t^{\omega}}$

- Bias toward predicted clean graph with $\omega\, R_t^{\omega}$
- Validity rises sooner

Inference stage improvement

The introduced techniques lead to **cumulative improvement of graph generative performance**



Similar techniques are explored to achieve **faster training**

Leveraging and ensuring graph-specific properties

GNNs have **limited representation power**

↓

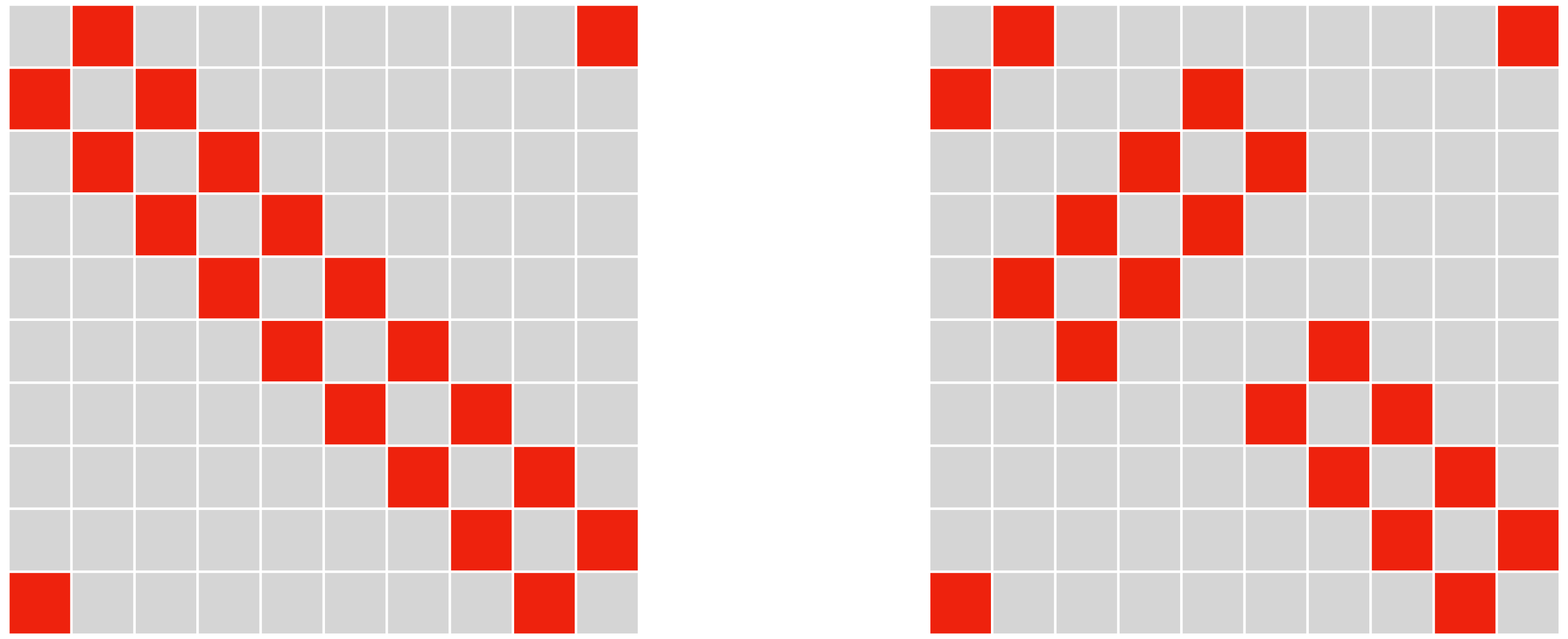
DeFoG employs **RRWP features** to enhance graph generation performance

↓

↑ Expressivity
↑ Efficiency

DeFoG respects graph symmetries

Sampling probability is permutation invariant

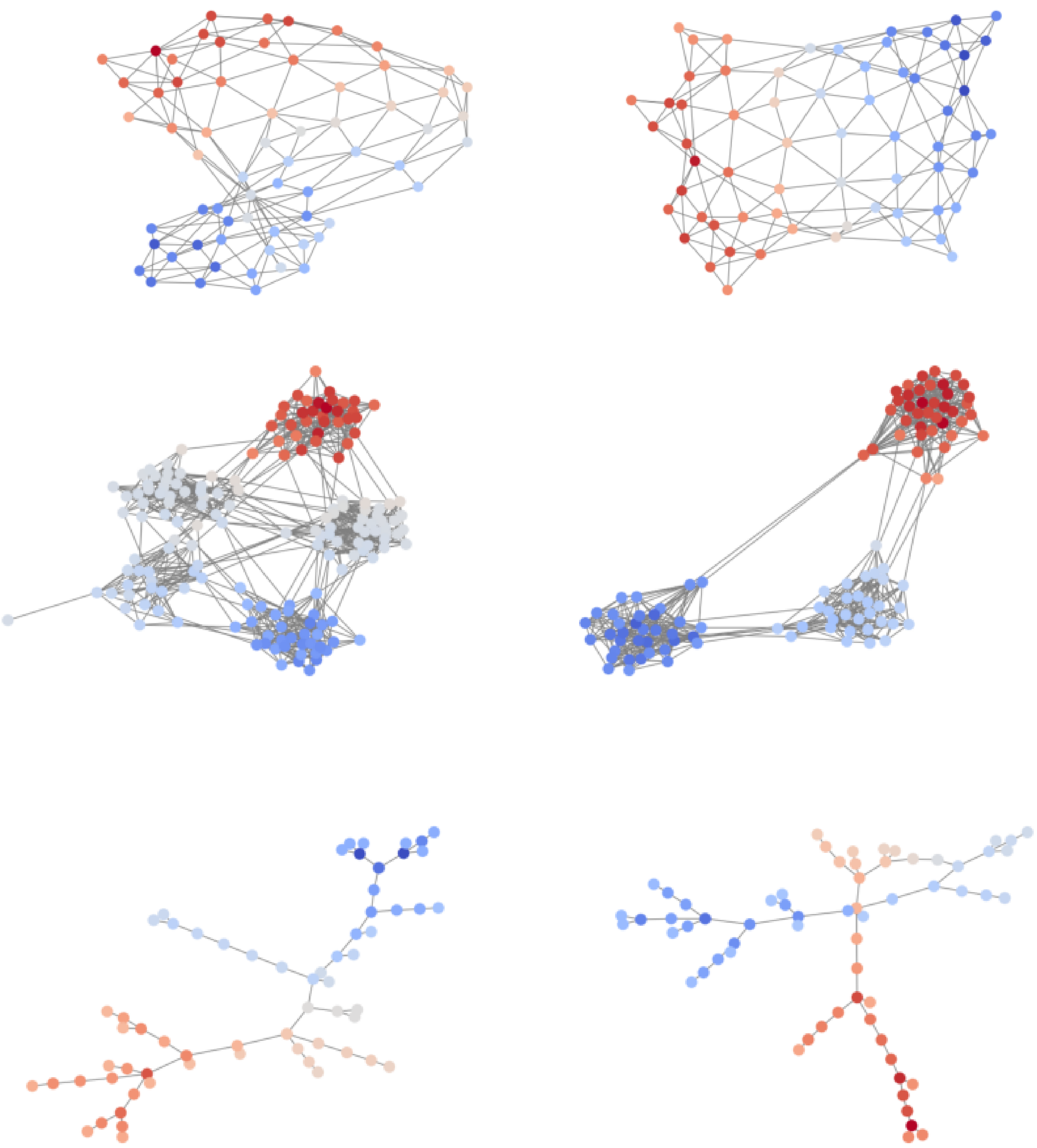


Graph Inductive Biases in Transformers without Message Passing, Ma et al., ICML 2023
Cometh: A continuous-time discrete-state graph diffusion model, Siraudin et al., ArXiv 2024

Evaluating graph generation

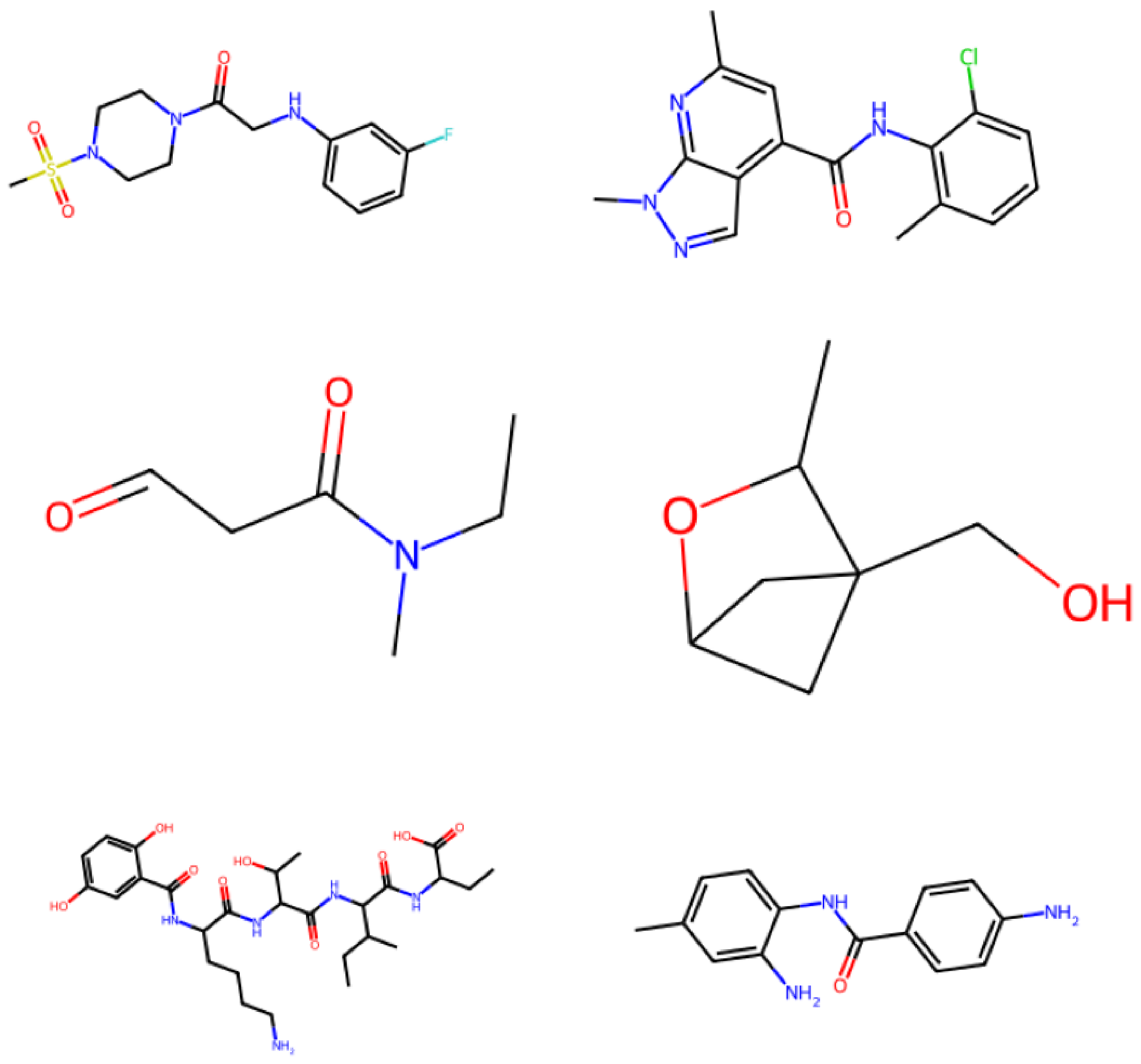
Synthetic graph generation

With varying topologies



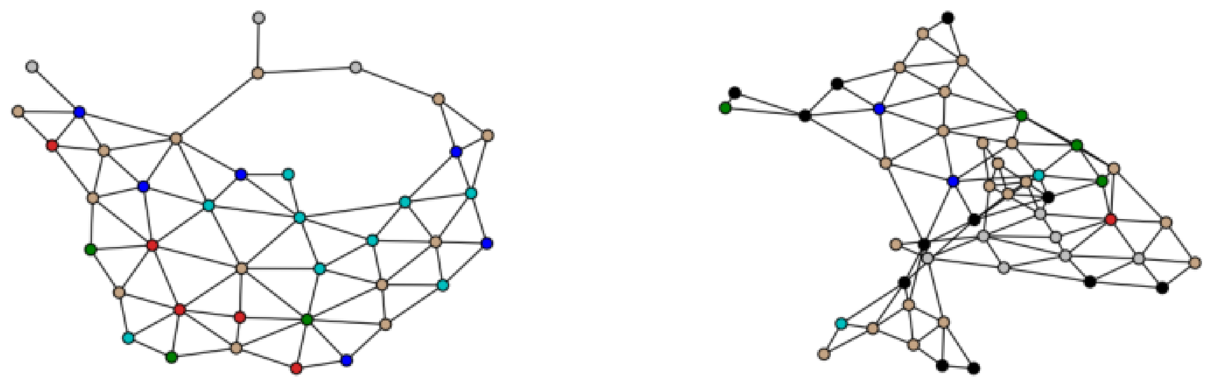
Molecular graph generation

With rich node and edge classes

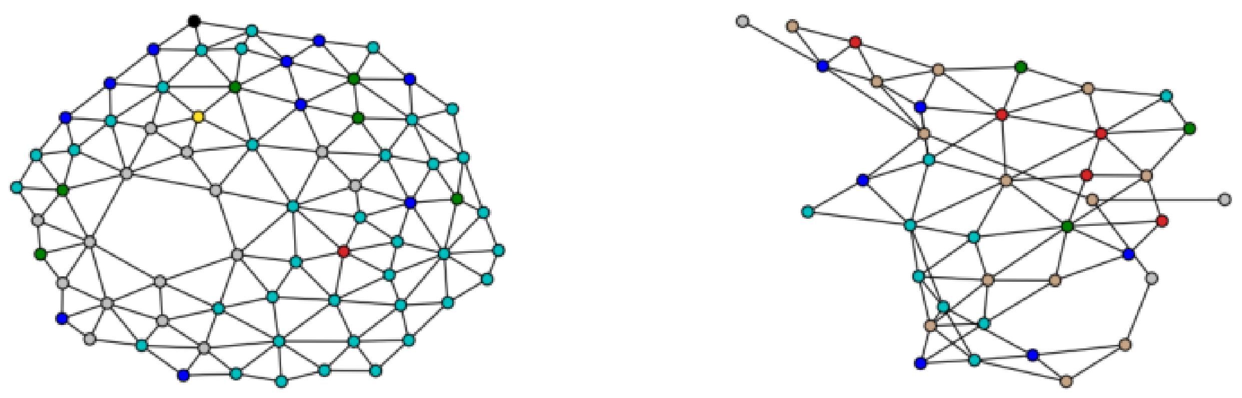


Conditional graph generation

With biological relevance



Low TLS

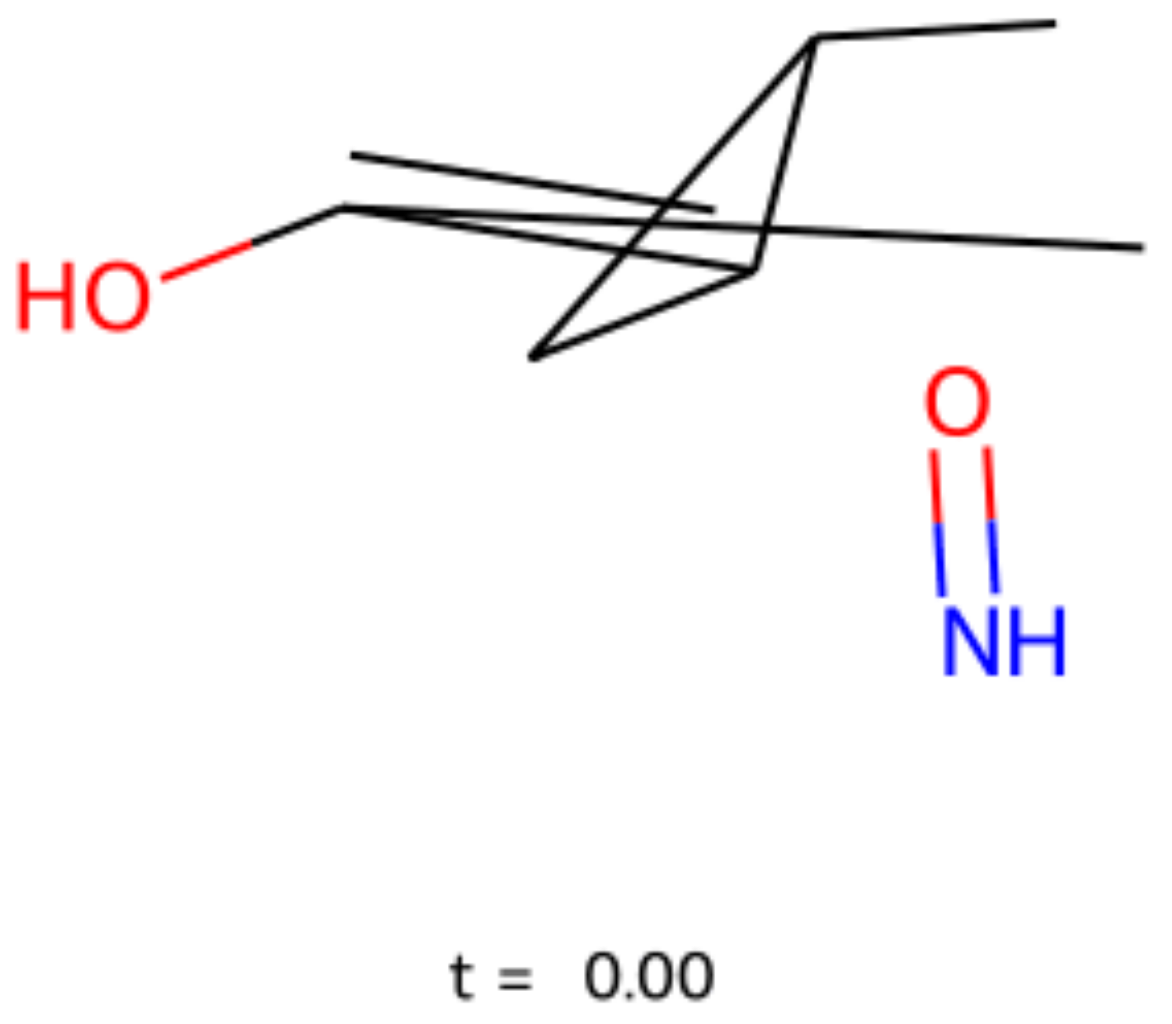


High TLS

TLS: Tertiary Lymphoid Structure

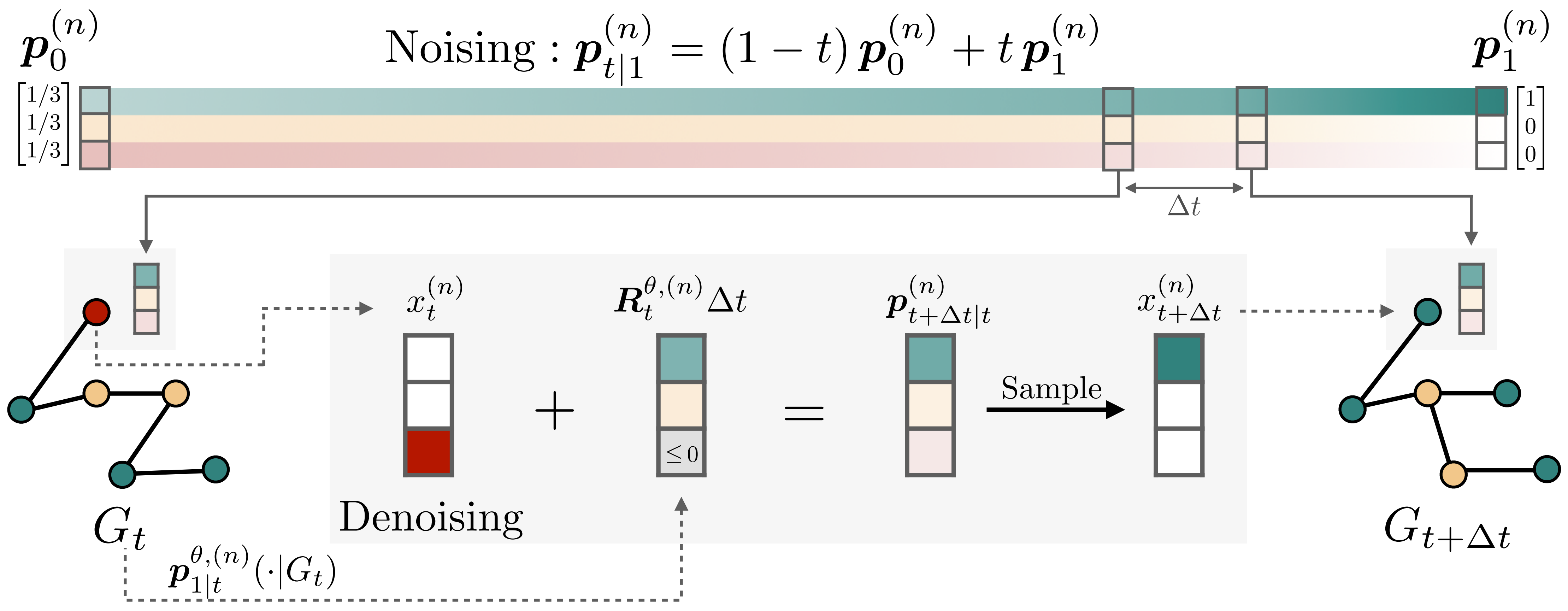
Generating molecules with DeFoG

Molecule generated for QM9



	Guacamol				
Model	Val. ↑	V.U. ↑	V.U.N.↑	KL div ↑	FCD ↑
Training set	100.0	100.0	0.0	99.9	92.8
DiGress (Vignac et al., 2022)	85.2	85.2	85.1	92.9	68.0
DisCo (Xu et al., 2024)	86.6	86.6	86.5	92.6	59.7
Cometh (Siraudin et al., 2024)	<u>98.9</u>	<u>98.9</u>	<u>97.6</u>	<u>96.7</u>	<u>72.7</u>
DeFoG (10% steps)	91.7	91.7	91.2	92.3	57.9
DeFoG	**99.0**	**99.0**	**97.9**	**97.7**	**73.8**

Takeaways



- **DeFoG** leverages the **flexibility** of the discrete flow matching formulation
- Exploiting its design space enables **improved and more efficient graph generation**
- **State-of-the-art performance** across diverse graph benchmarks

See you at Poster #E-3004

Today at 4:30pm