Bures-Wasserstein Flow Matching for Graph Generation

Keyue Jiang[1,2], Jiahao Cui[1], Xiaowen Dong[2], Laura Toni[1]

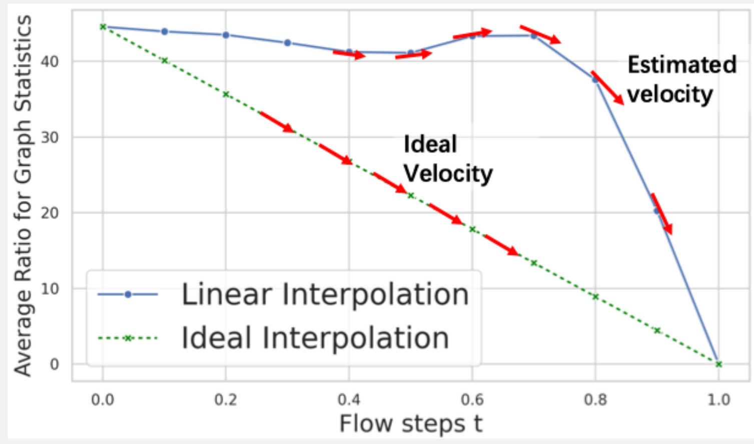
1. University College London, London (UK) 2. University of Oxford, Oxford (UK)

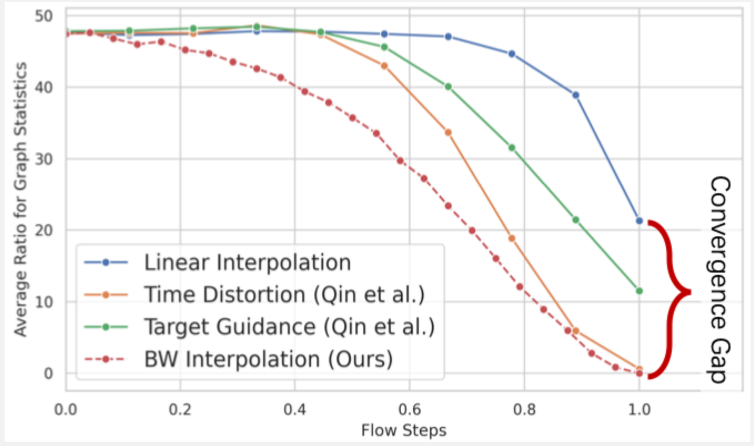
Overview

Graph generation is critical for tasks like drug discovery, circuit design, etc.
Diffusion/flow-based graph generation has achieved solid performance

The problem:

- Flow methods rely on building a probability path between reference and data distributions via independent linear Interpolations in nodes and edges.
- Graphs are relational, not independent collections of nodes and edges.
- Irregular paths lead to poor velocity estimation and weak convergence.

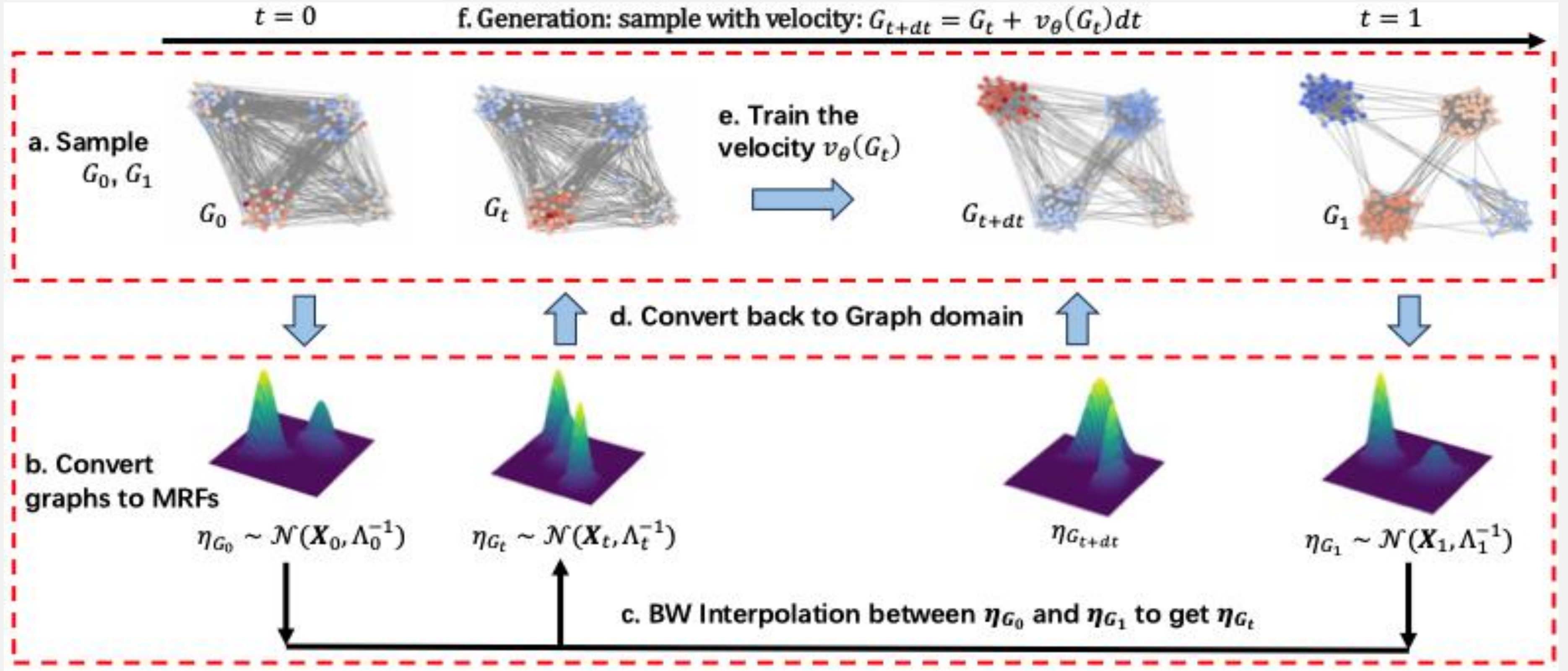




Solution

Our solution:

- Introduced GraphMRF based graph representation for coupled node-edge evolution.
- Derived Bures-Wasserstein graph interpolation and closed-form velocity fields.
- Proposed BWFlow for smooth, graph-aware flow matching.
- Extended the framework to discrete graph generation.



Bures-Wasserstein Flow Matching for Graph generation

Flow Matching

- Drawing Samples from a data distribution p_1 and a reference distribution p_0
- Construct a probability path between p_0 and p_1 to get p_t
- Train a model to reconstruct the probability path via approximating $p_{1|t}$, or velocity v_t.
- Sampling ng from p_0 and transforming it through the learned velocity to get samples follow p_1

Graph Markov Random Fields

- Let $G = (V, E)$ denote a graph with node features $x_v \in \mathbb{R}^K$,weighted adjacency matrix W, degree matrix $D = \mathrm{diag}(W\mathbf{1})$,and Laplacian $L = D - W$.
- The GraphMRF formulation is

$$p(G; \mathbf{G}) = p(X, E; \mathbf{X}, W) = p(X; \mathbf{X}, W)\, p(E; W),$$
$$E \sim \delta(W), \mathrm{vec}(X) \sim \mathcal{N}(\mathbf{X}, \Lambda^{\dagger}),\ \mathbf{X} = \mathrm{vec}(V^{\dagger}\mu),\ \ \Lambda = (\nu I + L) \otimes V^{\top}V.$$

- Here, $\otimes$denotes the Kronecker product, $\mathrm{vec}(\cdot)$denotes vectorization, and I is the identity matrix.

Optimal Transport between graphs: Bures-Wasserstein distance

- We measure the optimal transport discrepancy between two graphs $G_0$and $G_1$with a **Bures-Wasserstein distance** in the GraphMRF space.
- The distance combines a node-feature term and a graph-structure term:

$$d_{\mathrm{BW}}(G_0, G_1) = \| \mathbf{X}_0 - \mathbf{X}_1 \|_F^2 + \beta\, \mathrm{tr}\left(L_0^{\dagger} + L_1^{\dagger} - 2\left(L_0^{\dagger/2} L_1^{\dagger} L_0^{\dagger/2}\right)^{1/2}\right)$$

BW interpolation

- The G_t by interpolating between G_0 and G_1 in the BW geometry.
- The graph evolves as

$$\mathbf{X}_t = (1-t)\mathbf{X}_0 + t\mathbf{X}_1, \qquad L_t^{\dagger} = L_0^{1/2}\left((1-t)L_0^{\dagger} + t\left(L_0^{\dagger/2} L_1^{\dagger} L_0^{\dagger/2}\right)^{1/2}\right)^2 L_0^{1/2}$$

- This produces a smoother probability path than standard linear interpolation.

Bures-Wasserstein Velocity

- The **velocity field** that drives the graph from the intermediate state $G_t$toward the target distribution. For node features, the velocity is

$$v_t(X_t \mid G_0, G_1) = \frac{1}{1-t}(X_1 - X_t).$$

- The velocity field for graph structure

$$v_t(E_t \mid G_0, G_1) = \dot{W}_t = \mathrm{diag}(\dot{L}_t) - \dot{L}_t.$$

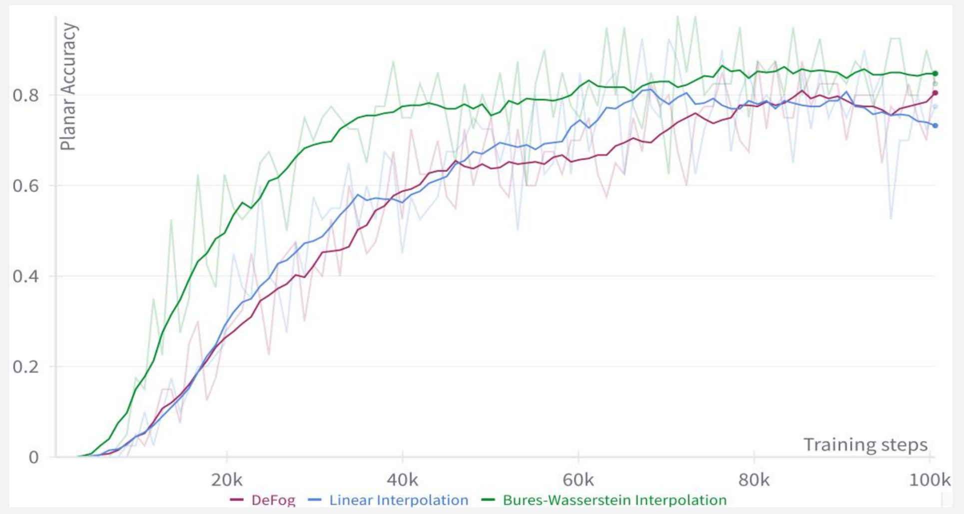
Experiment Results

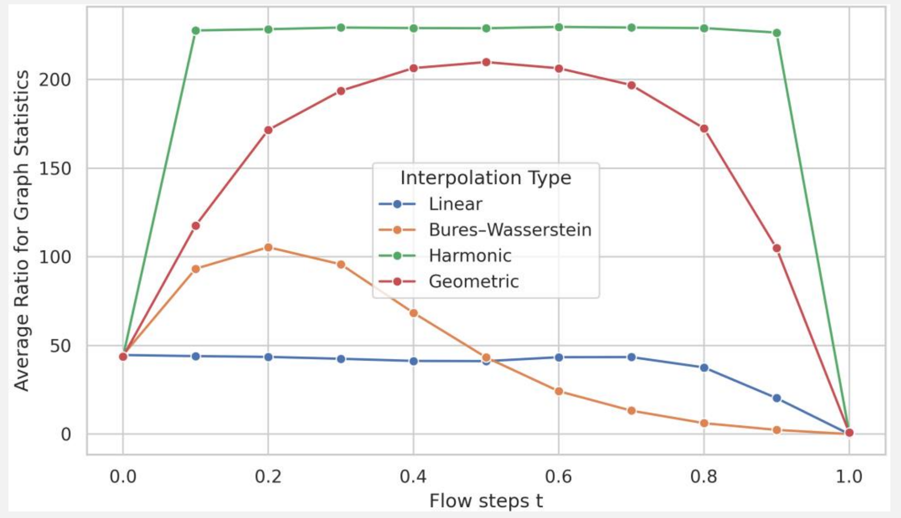
- BWFlow on plain graph generation and molecule generation

Model	Class	Planar V.U.N. ↑	Planar A.Ratio ↓	Tree V.U.N. ↑	Tree A.Ratio ↓	SBM V.U.N. ↑	SBM A.Ratio ↓
Train set	—	100	1.0	100	1.0	85.9	1.0
DiGress (CAVG) (Vignac et al., 2023a)	Diffusion	61.5±10.1	9.9 ±3.3	56.0 ±11.0	8.9±3.2	56.0±8.5	3.5±0.5
DisCo (CAVG) (Xu et al., 2024)	Diffusion	57.5± 2.5	9.0± 1.4	/	/	55.0± 5.9	11.6± 2.9
HSpectre (Bergmeister et al., 2024)	Diffusion	67.5	3.0	82.5	2.1	75.0	10.5
GruM (CAVG) (Jo et al., 2024)	Diffusion	74.4±5.15	3.2±0.4	52.5±3.2	2.4±0.7	73.5±6.7	2.6±0.6
Cometh (CAVG) (Siraudin et al., 2024)	Diffusion	80.5± 5.79	3.0± 0.6	**84.5**± 7.8	2.0± 0.4	77.5± 5.7	4.7±0.6
DeFoG (CAVG) (Qin et al., 2024)	Flow	77.5±8.37	3.5±1.7	83.5±10.8	1.9±0.4	**85.0**±7.1	3.4±0.4
BWFlow (CAVG)	Flow	**84.8**±6.44	**2.4**±0.9	81.5±4.9	**1.3**±0.2	84.5±8.0	**2.3**±0.5

Dataset	Interpolation	μ	V.U.N(%)	Mol.Stab.	Atom.Stab.	Connected(%)	Charge(10^{-2})	Atom(10^{-2})	Angles(°)
QM9 (with h)	MiDi	1.01	93.13	93.98	99.60	99.21	0.2	3.7	2.21
	FlowMol	1.01	87.53	88.45	99.13	99.09	0.4	4.2	2.72
	BWFlow	1.01	**96.45**	**97.84**	**99.84**	**99.24**	**0.1**	**2.3**	**1.96**
GEOM (with h)	Midi	1.34	78.23	32.42	89.61	**79.15**	0.6	11.2	9.6
	FlowMol	1.34	82.20	36.90	94.60	59.98	0.4	8.8	6.5
	BWFlow	**1.20**	**87.75**	**46.80**	**95.08**	73.53	**0.1**	**6.5**	**3.96**

- BWFlow improves **training and sampling dynamics**. It first explores more out-of-distribution regions, then moves smoothly toward the data distribution.





Future Work

- Extending the framework to **multiple edge or relation types .**
- The GraphMRF prior is most beneficial for graphs with narrower spectral spread, which may explain weaker gains on tree graphs.
- Scalable interpolation methods and more general graph priors.

Contact: keyue.jiang.18@ucl.ac.uk