Feature-aware Hypergraph Generation via Next-Scale Prediction

Dorian Gaillhard   Enzo Tartaglione   Lirida Naviner   Jhony H. Giraldo

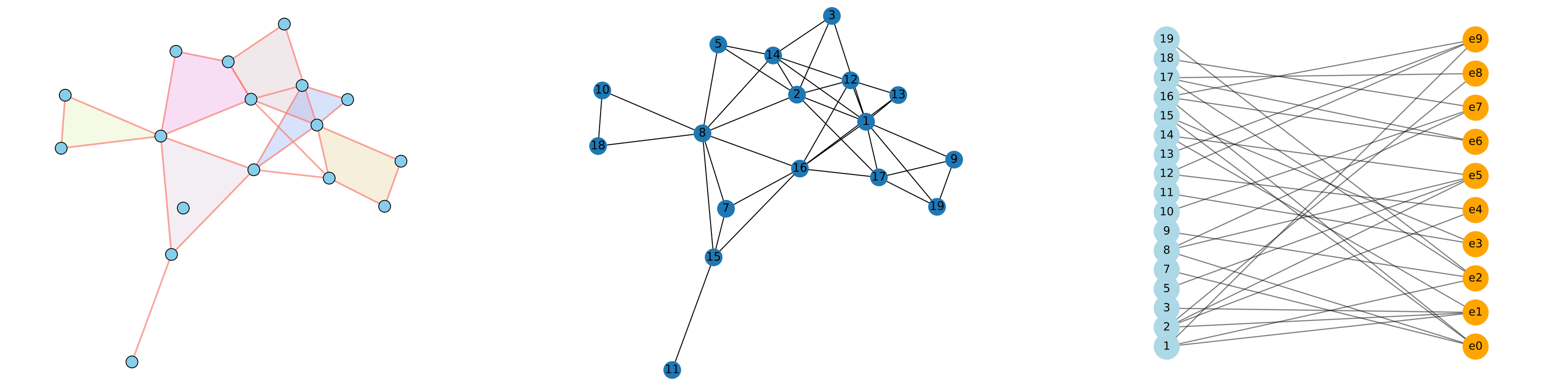
LTCI, Télécom Paris, Institut Polytechnique de Paris, France

{name.surname}@telecom-paris.fr

We introduce the first hierarchical method for large and complex featured (hyper)graph generation

Motivation

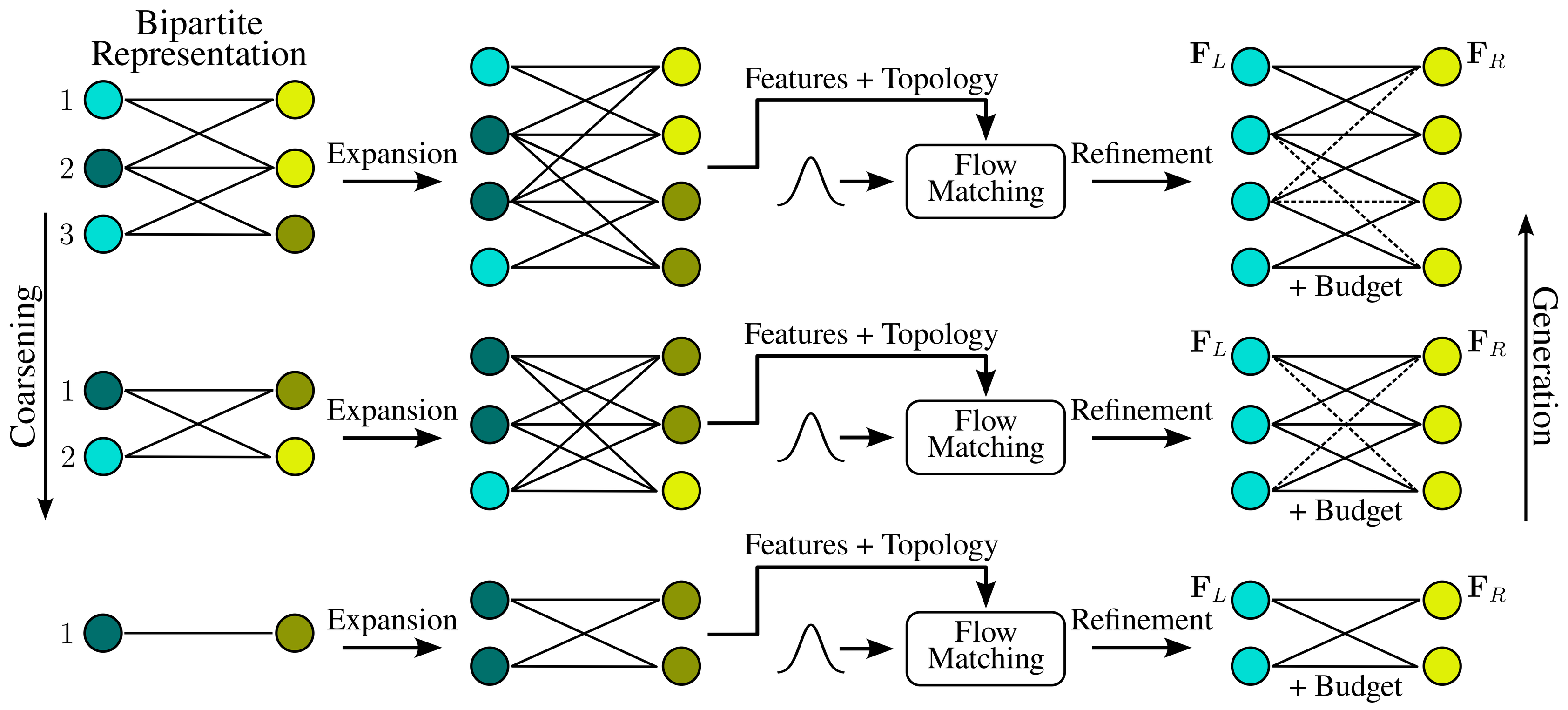
- Traditional one-shot approaches to graph generation struggle to scale effectively, which restricts the size and complexity of the graphs they can produce. This highlights the need for hierarchical methods.
- However, existing hierarchical approaches lack the ability to generate node and edge features, significantly limiting their applicability. Sequentially generating the topology followed by features is insufficient; a joint generation strategy is required.
- Hypergraphs—an extension of graphs in which *hyper*edges can connect more than two nodes—offer greater expressive power than standard graphs. For instance, they can naturally represent complex structures such as 3D meshes.



(a) Original hypergraph   (b) Clique expansion   (c) Bipartite representation

Figure. A hypergraph, its clique expansion and its bipartite representation

Pipeline



Like graphs [1] or images [2], unzoom (lose details) and train a model to zoom (reconstruct details). During training, hypergraphs are coarsened by merging nodes and hyperedges, averaging features and summing budgets. The model learns to predict merged nodes at each scale. In the expansion phase, those nodes are expanded with the same connectivity, feature and budget as their parents, and the model learns to: (*a*) identify excess edges, (*b*) split budgets, and (*c*) refine features.

Ideas

1. **Node budgets:** Rather than specifying a fixed number of nodes as input, the model learns to allocate *node budgets*—the number of nodes each coarse cluster should expand into—across different regions of the hypergraph during generation.
2. **Coarsening:** A coarsening algorithm [3] is applied to simplify the hypergraph by merging nodes while preserving overall connectivity. When nodes are merged, their features are averaged, and we record the number of original nodes within each cluster, *i.e. node budgets*.
3. **Generation:** The model is trained to reverse the coarsening process by expanding clusters back into individual nodes, inheriting their parent's connectivity, features, and budget. It then refines the structure by selectively pruning edges, updating features, and distributing each parent's budget among its children.

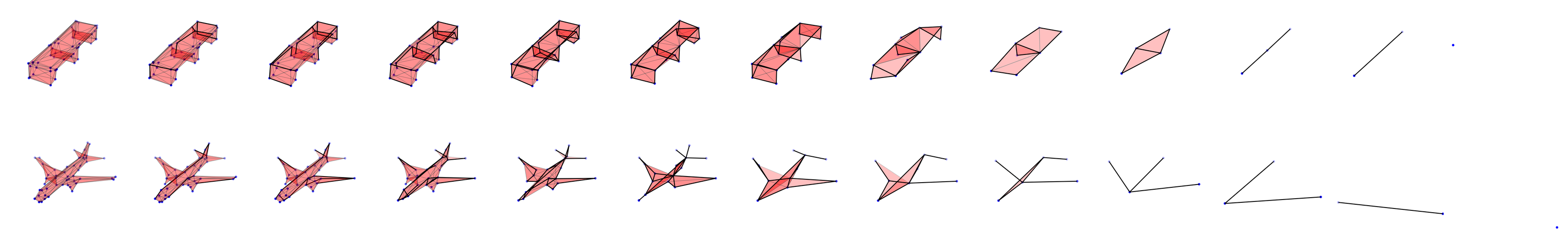


Figure. Examples of coarsening sequences for two 3D meshes.

Generated examples

Train samples   Generated   Train samples   Generated   Train samples   Generated   Train samples   Sequential   Joint   Train samples   Sequential   Joint

(*i*) Stochastic Block Model hypergraphs   (*ii*) Ego hypergraphs   (*iii*) Tree hypergraphs   (*vi*) Generated QM9 molecules   (*iv*) Bench 3D meshes   (*v*) Airplane 3D meshes

Numerical Results

Model	SBM Hypergraphs					Ego Hypergraphs					Tree Hypergraphs				
	Valid SBM ↑	Node Num ↓	Node Deg ↓	Edge Size ↓	Spectral ↓	Valid Ego ↑	Node Num ↓	Node Deg ↓	Edge Size ↓	Spectral ↓	Valid Tree ↑	Node Num ↓	Node Deg ↓	Edge Size ↓	Spectral ↓
HyperPA	2.5%	0.075	4.062	0.407	0.273	0%	35.83	2.590	0.423	0.237	0%	2.350	0.315	0.284	0.159
VAE	0%	0.375	1.280	1.059	0.024	0%	47.58	0.803	1.458	0.133	0%	9.700	0.072	0.480	0.124
GAN	0%	1.200	2.106	1.203	0.059	0%	60.35	0.917	1.665	0.230	0%	6.000	0.151	0.459	0.089
Diffusion	0%	0.150	1.717	1.390	0.031	0%	4.475	3.984	2.985	0.190	0%	2.225	1.718	1.922	0.127
HYGENE	65%	0.525	**0.321**	0.002	0.010	90%	12.55	**0.063**	0.220	**0.004**	77.5%	**0.000**	0.059	0.108	0.012
Ours	**81.4%**	**0.010**	0.603	0.005	**0.005**	**100%**	**0.162**	0.171	**0.129**	0.007	**82.8%**	**0.000**	**0.043**	**0.046**	**0.002**

Model	QM9		
	Valid Mol ↑	Unique Mol ↑	FCD ↓
DiGress	99.0%	96.2	–
DisCo	99.3%	–	–
Cometh	**99.6%**	**96.8**	0.25
DeFoG	99.3%	96.3	**0.12**
Ours	77.8%	94.3	3.86

Metrics	Manifold40 Bench		Manifold40 Airplane	
	Sequential Generation	Joint Generation	Sequential Generation	Joint Generation
Node Num ↓	0.367	**0.067**	0.078	**0**
Node Deg ↓	0.801	**0.581**	0.332	**0.304**
Edge Size ↓	**0.004**	0.008	**0.011**	0.033
Spectral ↓	**0.007**	0.014	**0.015**	**0.015**
Cham Dist ↓	0.143	**0.073**	0.117	**0.049**

[1] Bergmeister, A., Martinkus, K., Perraudin, N., Wattenhofer, R. (2023). *Efficient and scalable graph generation through iterative local expansion*

[2] S. Ren, Q. Yu, J. He, X. Shen, A. Yuille, L.-C. Chen. (2024). *FlowAR: Scale-wise autoregressive image generation meets flow matching*

[3] Loukas, A. (2019). *Graph reduction with spectral and cut guarantees*

Conclusions

- We propose a novel approach for hierarchical generation of featured (hyper)graphs.
- Our method draws inspiration from local expansion techniques [1] and the progressive upsampling strategy used in image generation [2].
- The introduction of the node budget mechanism significantly enhances the quality of the generated topologies.
- However, our method currently underperforms in molecule generation tasks compared to traditional one-shot approaches.

Next: Improve the robustness of the generation process.

Acknowledgments: The authors acknowledge the ANR – France (French National Research Agency) for its financial support of the System On Chip Design leveraging Artificial Intelligence (SODA) project under grant ANR-23-IAS3-0004 and the JCJC project DeSNAP ANR-24-CE23-1895-01. This project has also been partially funded by the Hi!PARIS Center on Data Analytics and Artificial Intelligence.