QUVA Deep Vision Lab

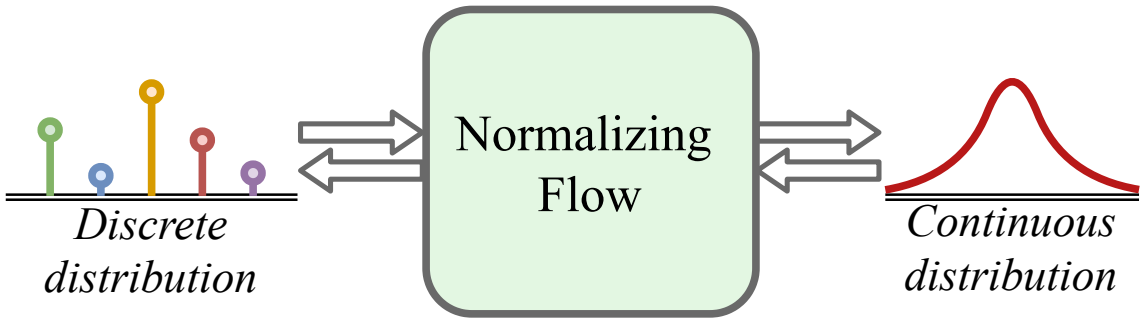
University of Amsterdam
Faculty of Science

Categorical Normalizing Flows via Continuous Transformations

Phillip Lippe, Efstratios Gavves

Introduction
Motivation

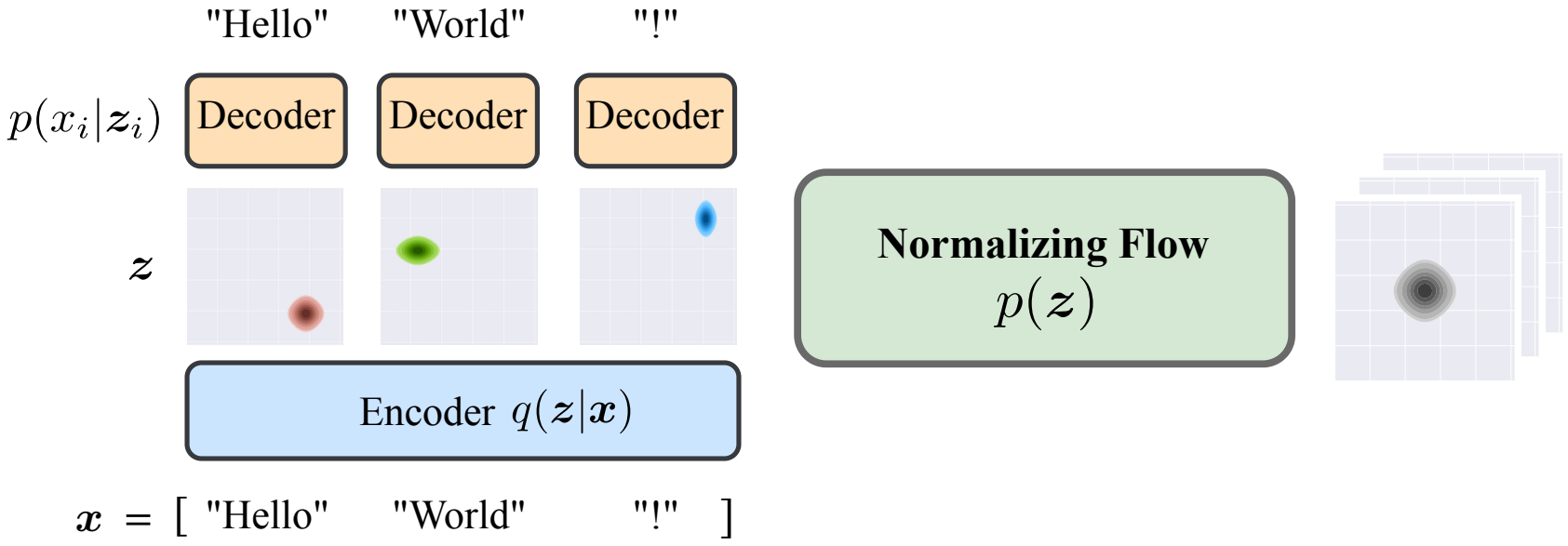
- Generative model for categorical data via Normalizing Flows
- Current practice
  - Dequantization ➜ categories not quantized values
  - Discrete NF ➜ hard to optimize
- Categorical NF:



Categorical Normalizing Flows
Encoding

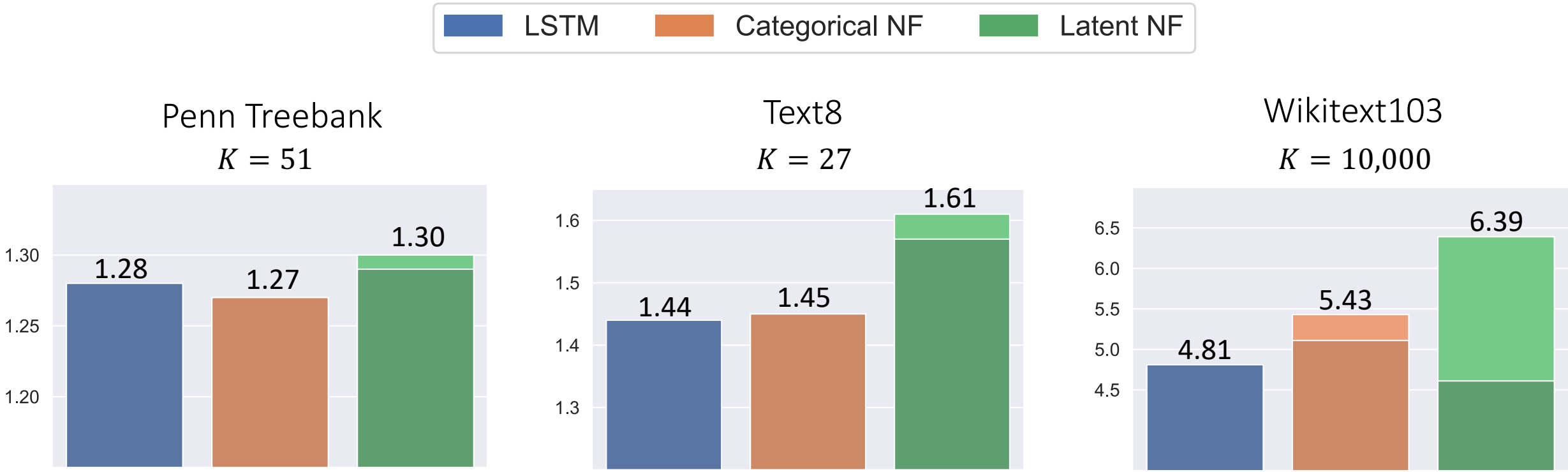
- Representing categorical variables in continuous space
- Encoding function learnable and flexible, yet no loss of information

⇒ Variational inference with factorized decoder: $p(\boldsymbol{x}) \geq \mathbb{E}_{\boldsymbol{z} \sim q(\cdot|\boldsymbol{x})} \left[\frac{\prod_i p(x_i|\boldsymbol{z}_i)}{q(\boldsymbol{z}|\boldsymbol{x})} p(\boldsymbol{z}) \right]$



Categorical Normalizing Flows

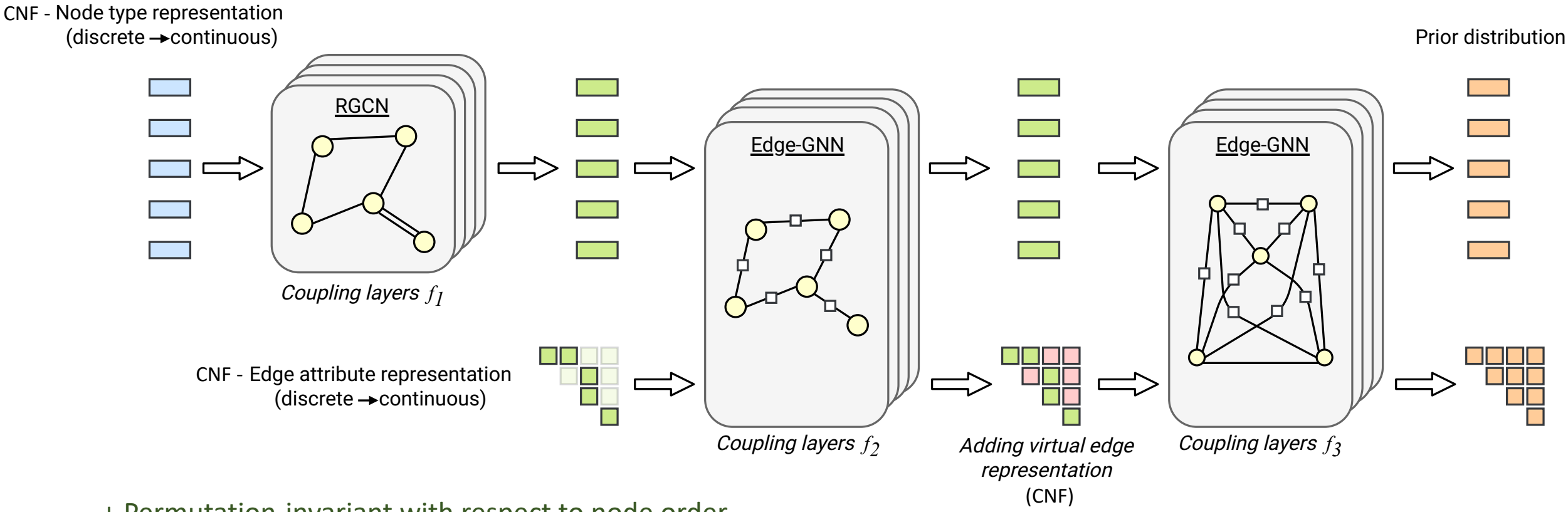
Experiments – Language Modeling



Metric: bits per character/word

Graph Generation with CNF

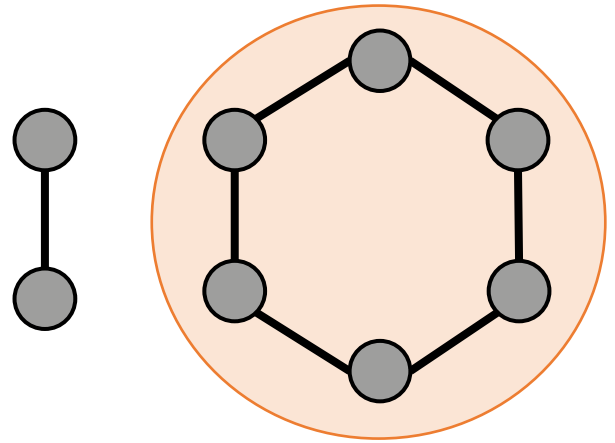
GraphCNF



+ Permutation-invariant with respect to node order
+ Efficient three-step approach

Graph Generation with CNF

Experiments – Molecule Generation



Results on the Zinc250k dataset

(224k examples)

Method	Validity	Uniqueness	Novelty	Reconstruction	Parallel	General
JT-VAE	100%	100%	100%	71%	✗	✗
GraphAF	68%	99.10%	100%	100%	✗	✓
R-VAE	34.9%	100%	–	54.7%	✓	✓
GraphNVP	42.60%	94.80%	100%	100%	✓	✓
GraphCNF	83.41% (±2.88)	99.99% (±0.01)	100% (±0.00)	100% (±0.00)	✓	✓
+ Sub-graphs						

Conclusion

- Categorical Normalizing Flow is a powerful, parallel generative model for categorical data
- GraphCNF enables permutation-invariant graph modeling with considerable gains on molecule generation

For more details, see our paper and visit the poster session!

Code available at: github.com/phlippe/CategoricalNF