Adversarial Learned Molecular Graph Inference and Generation

Sebastian Pölsterl and Christian Wachinger

Artificial Intelligence in Medical Imaging, Ludwig-Maximilians-Universität, Munich

European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases

September 14–18th 2020

Goal

Find a molecule with certain properties, e.g., an antiviral drug to inhibit SARS-CoV-2 replication.

Goal

Find a molecule with certain properties, e.g., an antiviral drug to inhibit SARS-CoV-2 replication.

Problem

1. The space of molecules is extremely large – in the order of 10^{33} drug-like molecules.[1]
2. Molecules are discrete in nature, which prevents the use of gradient-based optimization.

[1] P. G. Polishchuk et al. (2013). "Estimation of the size of drug-like chemical space based on GDB-17 data". In: *Journal of Computer-Aided Molecular Design* 27.8, pp. 675–679

De Novo Chemical Design

Goal

Find a molecule with certain properties, e.g., an antiviral drug to inhibit SARS-CoV-2 replication.

Problem

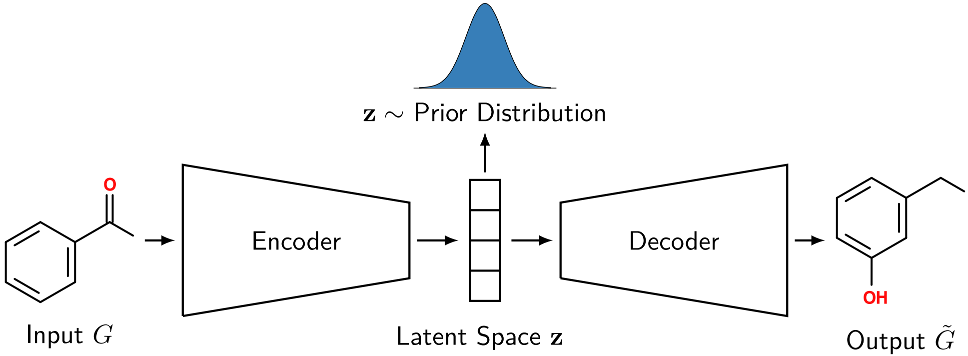
1. The space of molecules is extremely large – in the order of 10^{33} drug-like molecules.[1]
2. Molecules are discrete in nature, which prevents the use of gradient-based optimization.

Solution

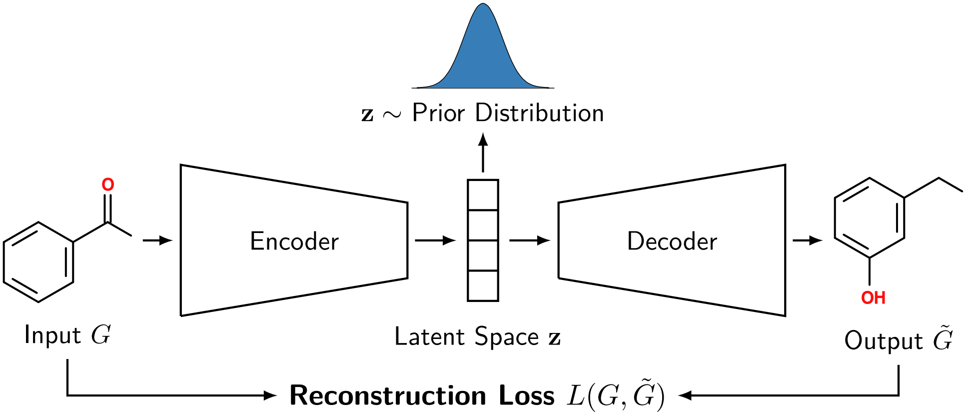
Use a **deep generative model** to project molecules into a continuous latent space and perform gradient-based optimization there.

[1] P. G. Polishchuk et al. (2013). "Estimation of the size of drug-like chemical space based on GDB-17 data". In: *Journal of Computer-Aided Molecular Design* 27.8, pp. 675–679

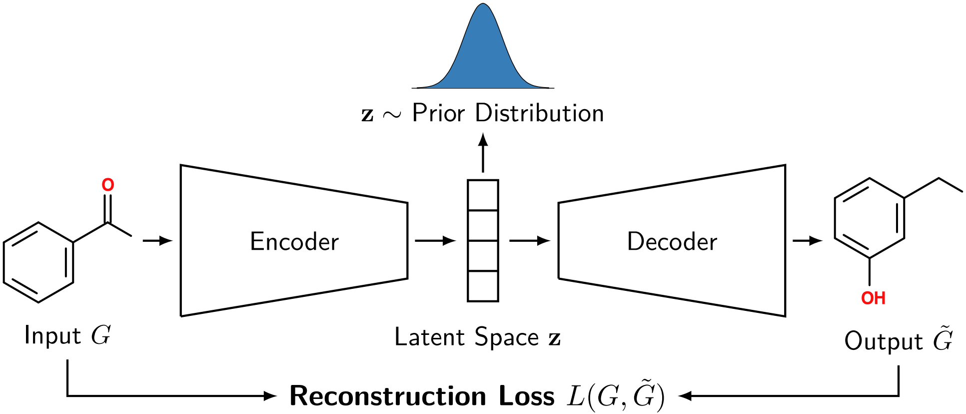
Graph Variational Autoencoder



Graph Variational Autoencoder



Graph Variational Autoencoder



Requires solving expensive graph isomorphism problem!

Prior Work I

Inference (Encoder): Various Graph Convolutional Neural Networks.

Generation (Decoder):

- In a single step using MLP (De Cao and Kipf, 2018; Ma et al., 2018; Simonovsky and Komodakis, 2018).
- Sequentially using RNN (Bradshaw et al., 2019; Jin et al., 2018; Li, Zhang, et al., 2018; Li, Vinyals, et al., 2018; Liu et al., 2018; Podda et al., 2020; Samanta et al., 2019; You et al., 2018).

Generative Models for Molecular Graphs:

- Likelihood-based (VAEs): compute reconstruction loss by (i) traversing nodes in a fixed order, (ii) Monte-Carlo sampling, or (iii) graph matching.
- Adversarial: MolGAN is the only such model, but cannot do inference (De Cao and Kipf, 2018).

Prior Work II

Generative Models for Molecular Graphs:

- Likelihood-based (VAEs): compute reconstruction loss by (i) traversing nodes in a fixed order, (ii) Monte-Carlo sampling, or (iii) graph matching.
- Adversarial: MolGAN is the only such model, but cannot do inference (De Cao and Kipf, 2018).

Generative Models for Continuous Data:

- Adversarial Learned Inference (ALI) and its extension ALICE learn an encoder/decoder **without optimizing an explicit reconstruction loss** (Dumoulin et al., 2017; Li, Liu, et al., 2017).
- ALI & ALICE are only applicable to continuous-valued data, such as images.

Our Contributions

- We propose **Adversarial Learned Molecular Graph Inference and Generation** (ALMGIG) that
  1. does not require solving an expensive graph isomorphism problem,

Our Contributions

- We propose **Adversarial Learned Molecular Graph Inference and Generation** (ALMGIG) that
  1. does not require solving an expensive graph isomorphism problem,
  2. performs inference over graphs by extending the Graph Isomorphism Network to multi-graphs (Xu et al., 2019),

Our Contributions

- We propose **Adversarial Learned Molecular Graph Inference and Generation** (ALMGIG) that
  1. does not require solving an expensive graph isomorphism problem,
  2. performs inference over graphs by extending the Graph Isomorphism Network to multi-graphs (Xu et al., 2019),
  3. generates discrete data (atoms and bonds) via the Gumbel-softmax trick (Jang et al., 2017; Maddison et al., 2017),

Our Contributions

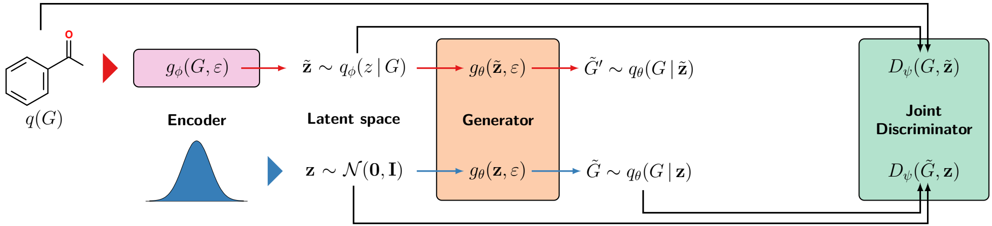
- We propose **Adversarial Learned Molecular Graph Inference and Generation** (ALMGIG) that
  1. does not require solving an expensive graph isomorphism problem,
  2. performs inference over graphs by extending the Graph Isomorphism Network to multi-graphs (Xu et al., 2019),
  3. generates discrete data (atoms and bonds) via the Gumbel-softmax trick (Jang et al., 2017; Maddison et al., 2017),
  4. generates chemically valid molecules by enforcing connectivity constraints via penalty terms (Ma et al., 2018).

Our Contributions

- We propose **Adversarial Learned Molecular Graph Inference and Generation** (ALMGIG) that
  1. does not require solving an expensive graph isomorphism problem,
  2. performs inference over graphs by extending the Graph Isomorphism Network to multi-graphs (Xu et al., 2019),
  3. generates discrete data (atoms and bonds) via the Gumbel-softmax trick (Jang et al., 2017; Maddison et al., 2017),
  4. generates chemically valid molecules by enforcing connectivity constraints via penalty terms (Ma et al., 2018).
- We show that current evaluation metrics are flawed, and propose a **better evaluation metric** to assess the distribution learning capabilities of methods.

Adversarial Learned Inference

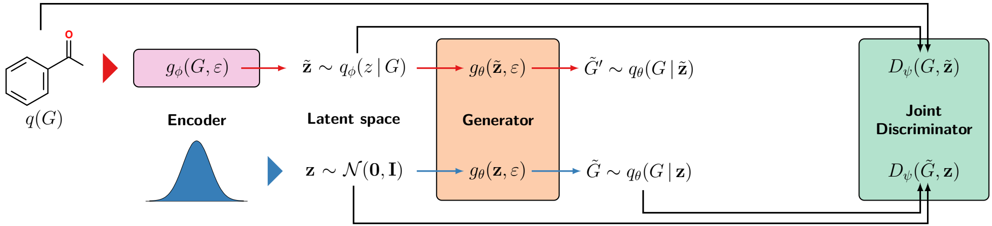
Dumoulin et al. (2017)



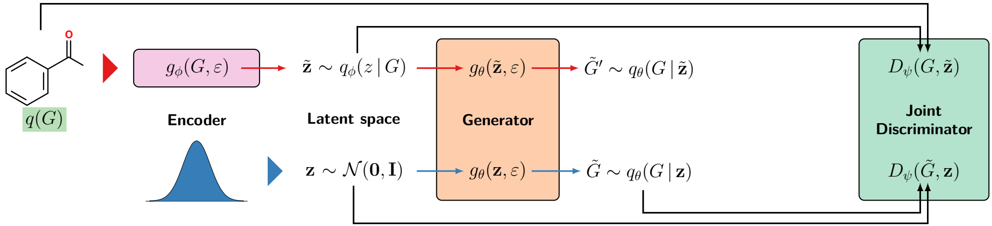
- **Training**: match joint distributions over graphs and latent variables

Adversarial Learned Inference

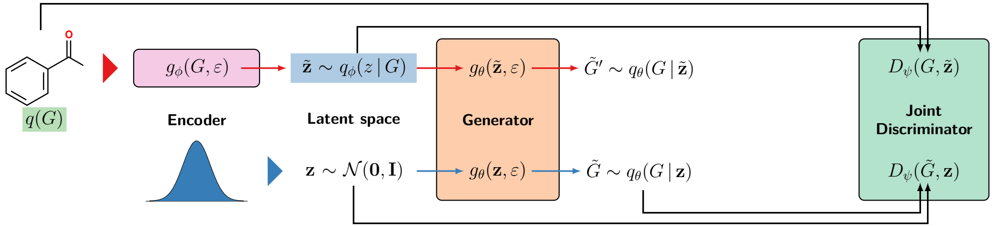
Dumoulin et al. (2017)



- **Training**: match joint distributions over graphs and latent variables
  1. encoder joint distribution: $q_\phi(G, z) = q(G)\, q_\phi(z \mid G)$

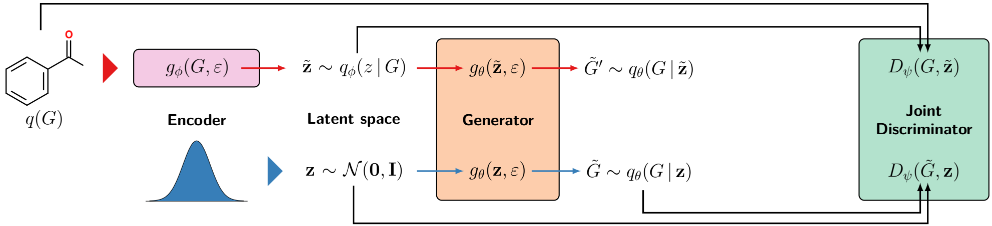


- **Training**: match joint distributions over graphs and latent variables
  1. encoder joint distribution: $q_\phi(G, z) = q(G)\, q_\phi(z \mid G)$

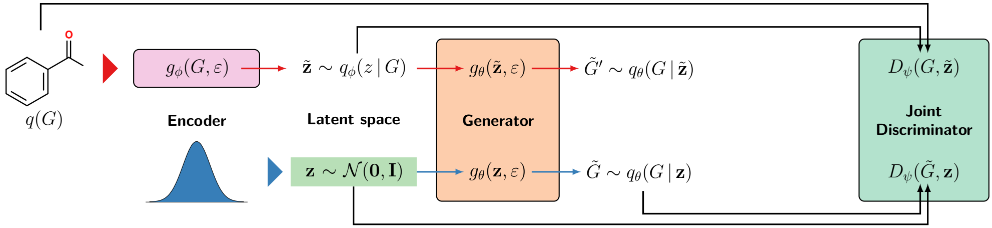


- **Training**: match joint distributions over graphs and latent variables
  1. encoder joint distribution: $q_\phi(G, z) = q(G)\, q_\phi(z \mid G)$

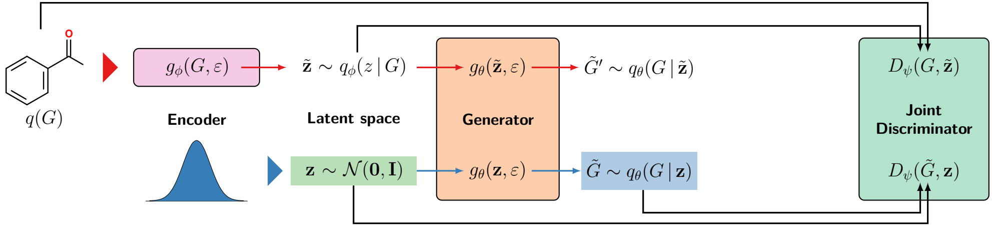
Dumoulin et al. (2017)



- **Training**: match joint distributions over graphs and latent variables
  1. encoder joint distribution: $q_\phi(G, z) = q(G)\, q_\phi(z \mid G)$
  2. decoder joint distribution: $p_\theta(G, z) = p_z(z)\, q_\theta(G \mid z)$



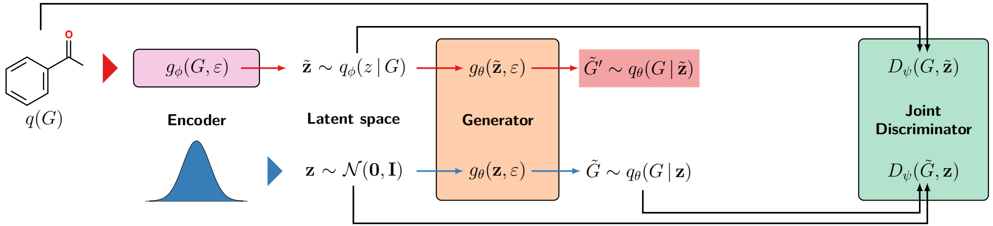
- **Training**: match joint distributions over graphs and latent variables
  1. encoder joint distribution: $q_\phi(G, z) = q(G)\, q_\phi(z \mid G)$
  2. decoder joint distribution: $p_\theta(G, z) = p_z(z)\, q_\theta(G \mid z)$



- **Training**: match joint distributions over graphs and latent variables
  1. encoder joint distribution: $q_\phi(G, z) = q(G)\, q_\phi(z \mid G)$
  2. decoder joint distribution: $p_\theta(G, z) = p_z(z)\, q_\theta(G \mid z)$

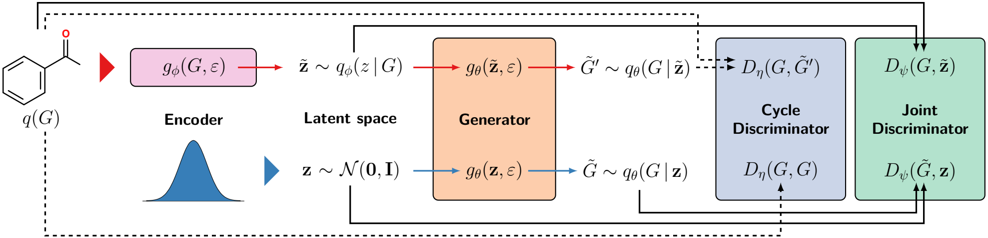
Adversarial Learned Inference

Dumoulin et al. (2017)

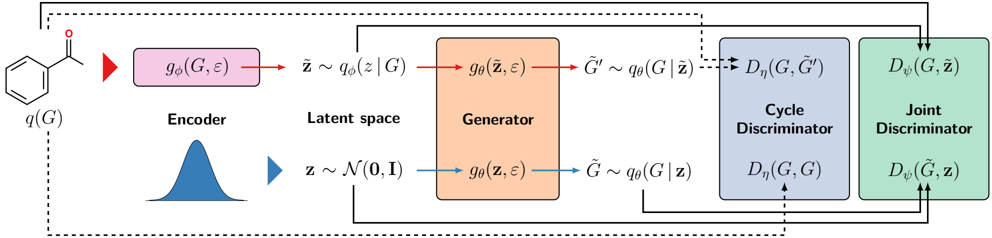


- **Training**: match joint distributions over graphs and latent variables
  1. encoder joint distribution: $q_\phi(G, z) = q(G)\, q_\phi(z \mid G)$
  2. decoder joint distribution: $p_\theta(G, z) = p_z(z)\, q_\theta(G \mid z)$
- However, reconstruction $\tilde{G}'$ remains unconstrained.

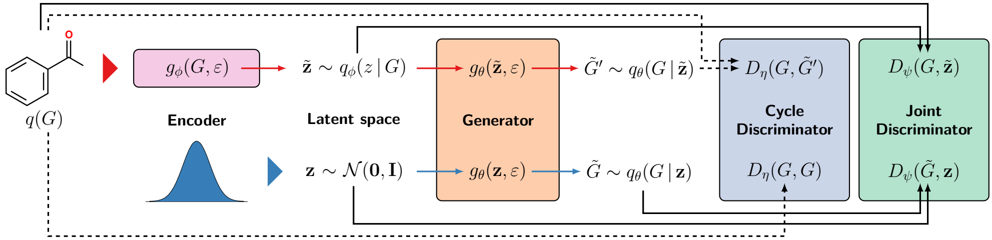
Adversarial Learned Inference
ALICE (Li, Liu, et al., 2017)



- ALICE adds cycle discriminator on pairs of graphs to enforce consistent reconstruction.



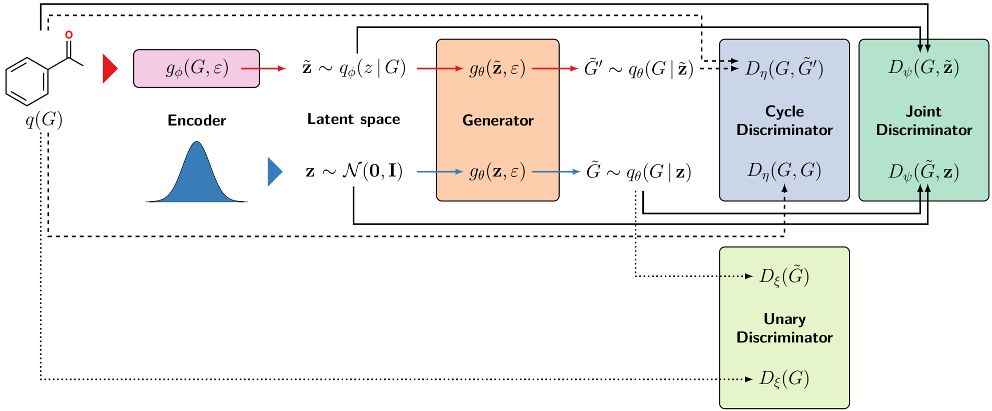
- ALICE adds cycle discriminator on pairs of graphs to enforce consistent reconstruction.
- At the optimum, encoder and decoder joint distribution will match, and $\tilde{G}' = G$.



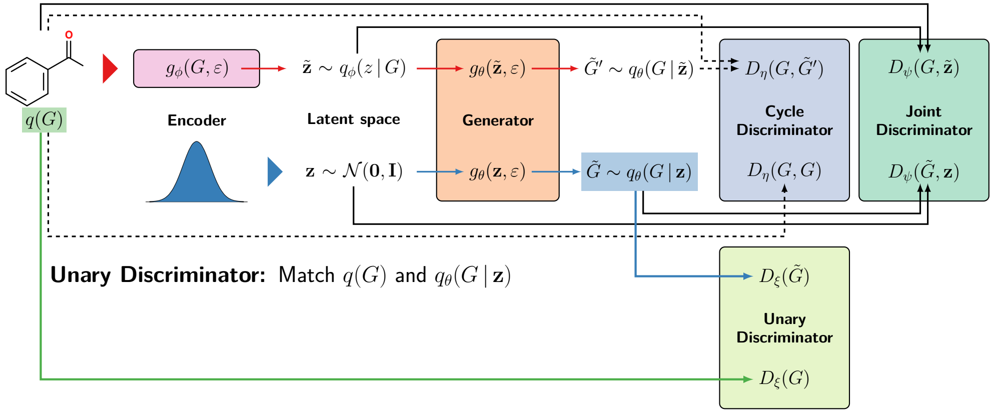
- ALICE adds cycle discriminator on pairs of graphs to enforce consistent reconstruction.
- At the optimum, encoder and decoder joint distribution will match, and $\tilde{G}' = G$.
- **However, in practice reaching the optimum is extremely hard.**

Adversarial Learned Inference

Unary Discriminator

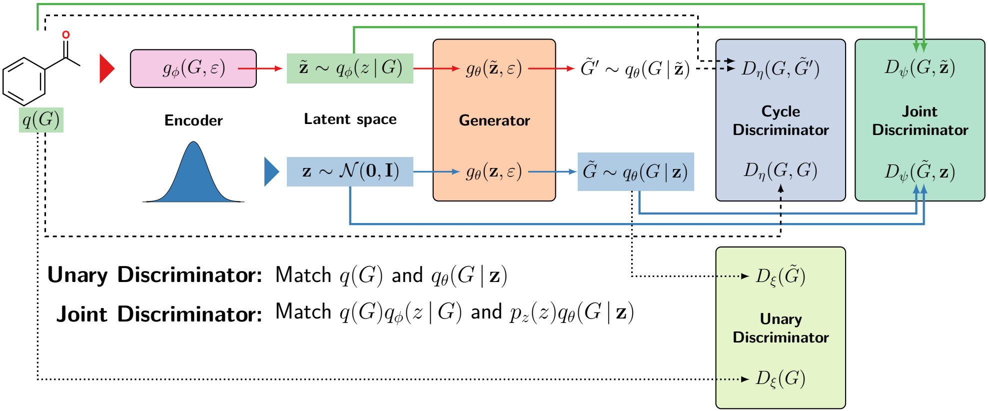


- Unary discriminator facilitates training when the joint distribution is difficult to learn.



- Unary discriminator facilitates training when the joint distribution is difficult to learn.

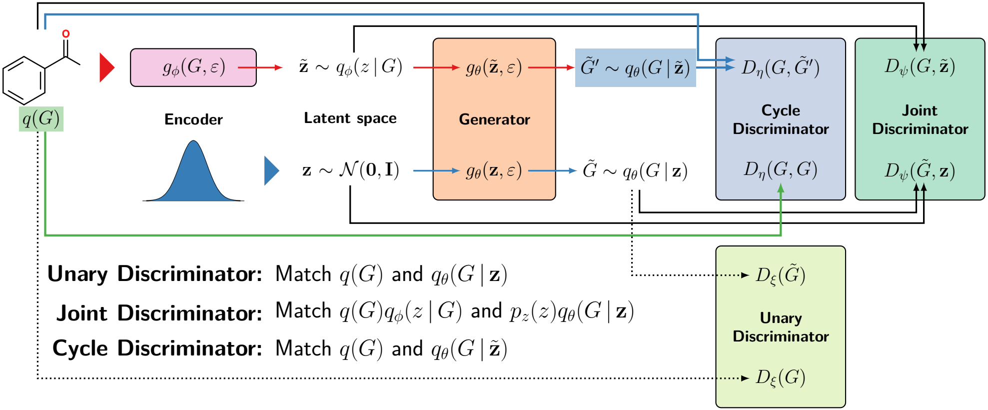
Adversarial Learned Inference
Unary Discriminator



- Unary discriminator facilitates training when the joint distribution is difficult to learn.

Adversarial Learned Inference

Unary Discriminator



- Unary discriminator facilitates training when the joint distribution is difficult to learn.

Experiments

Data: Molecules from the QM9 dataset (≤9 heavy atoms, 4 atom types, 3 bond types).

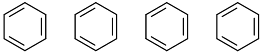
Competing Methods

- **CGVAE** (Liu et al., 2018), **NeVAE** (Samanta et al., 2019): Graph-based VAE with RNN-decoder and valence constraints.
- **GrammarVAE** (Kusner et al., 2017): SMILES-based VAE.
- **MolGAN** (De Cao and Kipf, 2018): Graph-based WGAN without encoder.
- **Random**: chooses atom and bonds randomly, but honors valence constraints.

Simple Metrics are Flawed

- **Validity:** Percentage of valid molecules.
- **Uniqueness:** Percentage of unique molecules.
- **Novelty:** Percentage of unique molecules not in the data.

- **Validity:** Percentage of valid molecules.

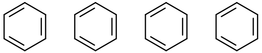
 $\cdots$ $\Rightarrow$ **100% Validity**

- **Uniqueness:** Percentage of unique molecules.

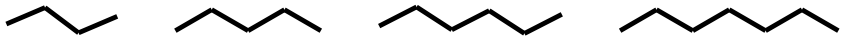
- **Novelty:** Percentage of unique molecules not in the data.

Simple Metrics are Flawed

- **Validity:** Percentage of valid molecules.

 · · · ⇒ **100% Validity**

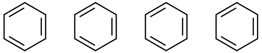
- **Uniqueness:** Percentage of unique molecules.

 · · · ⇒ **100% Uniqueness**

- **Novelty:** Percentage of unique molecules not in the data.

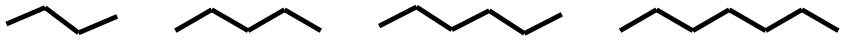
Simple Metrics are Flawed

- **Validity:** Percentage of valid molecules.



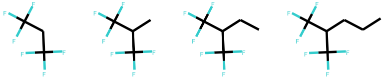
$\cdots$ $\Rightarrow$ **100% Validity**

- **Uniqueness:** Percentage of unique molecules.



$\cdots$ $\Rightarrow$ **100% Uniqueness**

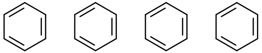
- **Novelty:** Percentage of unique molecules not in the data.



$\cdots$ $\Rightarrow$ **100% Novelty**

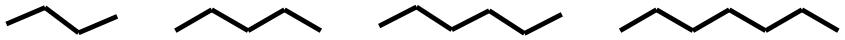
Simple Metrics are Flawed

- **Validity:** Percentage of valid molecules.



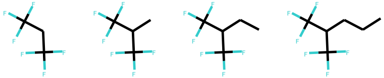
· · · ⇒ **100% Validity**

- **Uniqueness:** Percentage of unique molecules.



· · · ⇒ **100% Uniqueness**

- **Novelty:** Percentage of unique molecules not in the data.



· · · ⇒ **100% Novelty**

- **Metrics do not capture what models learned from the training data.**

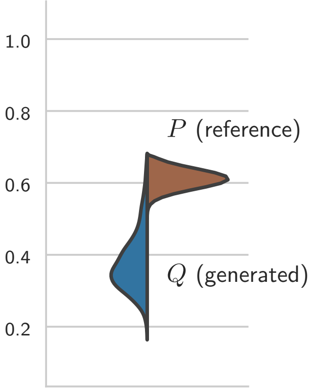
Advanced Metrics

What we are actually interested in:

Can we generate chemically meaningful molecules with similar properties as in the training data?

What we are actually interested in:

Can we generate chemically meaningful molecules with similar properties as in the training data?

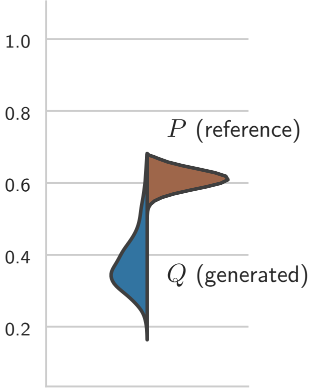


- Brown et al. (2019) compared the distribution of 10 chemical descriptors in terms of KL divergence $D_{\mathsf{KL}}(P \parallel Q)$.

Advanced Metrics

What we are actually interested in:

Can we generate chemically meaningful molecules with similar properties as in the training data?



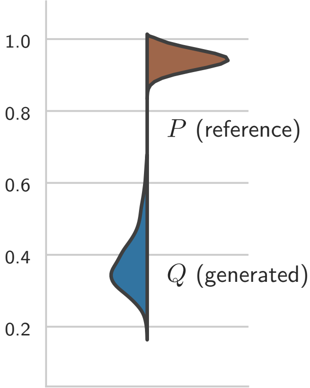
- Brown et al. (2019) compared the distribution of 10 chemical descriptors in terms of KL divergence $D_{\mathsf{KL}}(P \parallel Q)$.
- We propose using Earth Mover's Distance (EMD):

	EMD	KL div
Indiscernibility of identicals	✓	✓
Symmetry	✓	✗
Triangle inequality	✓	✗
Quantify spatial shift	✓	✗
Non-overlapping supports	✓	✗

Advanced Metrics

What we are actually interested in:

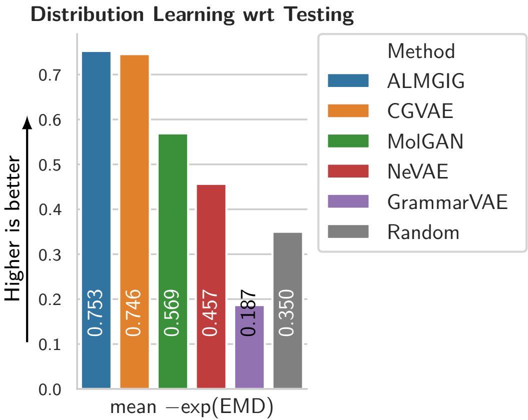
Can we generate chemically meaningful molecules with similar properties as in the training data?



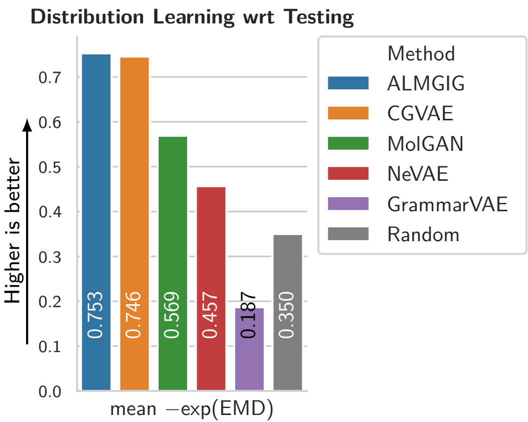
- Brown et al. (2019) compared the distribution of 10 chemical descriptors in terms of KL divergence $D_{\mathsf{KL}}(P \parallel Q)$.
- We propose using Earth Mover's Distance (EMD):

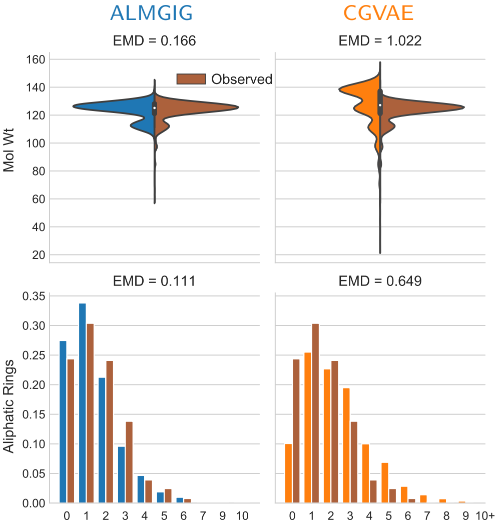
	EMD	KL div
Indiscernibility of identicals	✓	✓
Symmetry	✓	✗
Triangle inequality	✓	✗
Quantify spatial shift	✓	✗
Non-overlapping supports	✓	✗

Comparison – Advanced Metrics

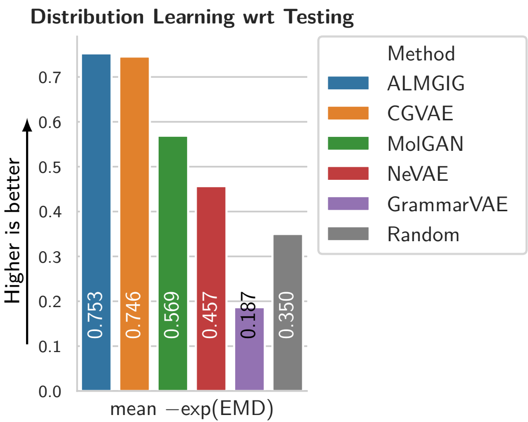


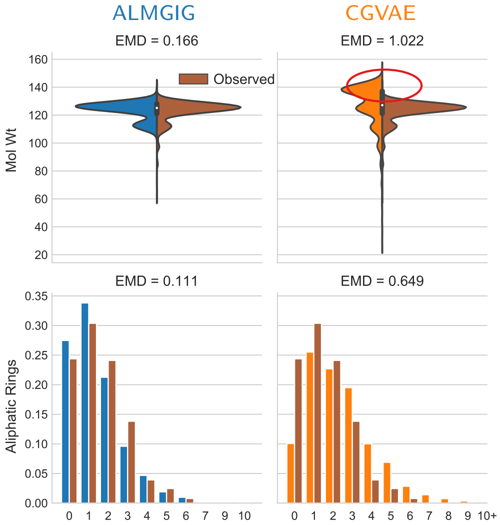
Comparison – Advanced Metrics



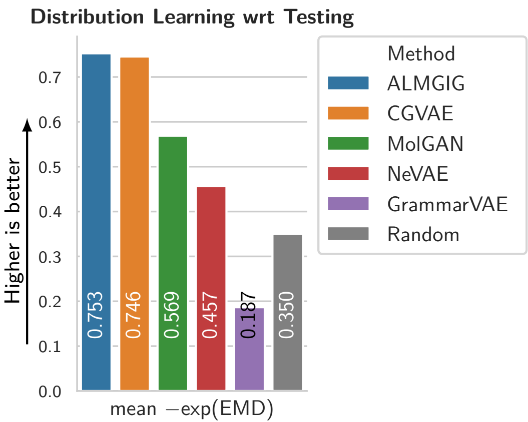


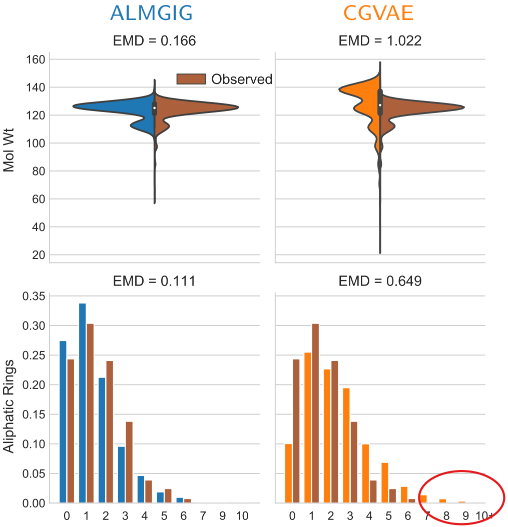
Comparison – Advanced Metrics



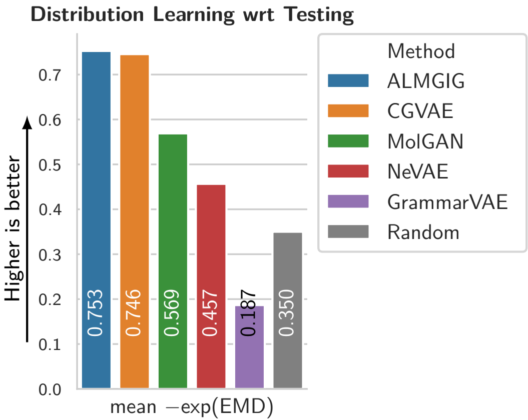


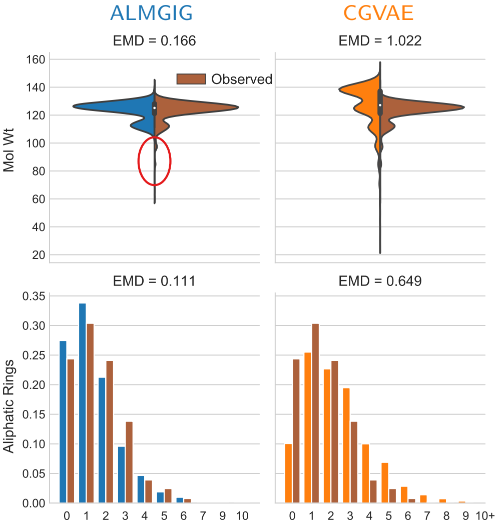
Comparison – Advanced Metrics



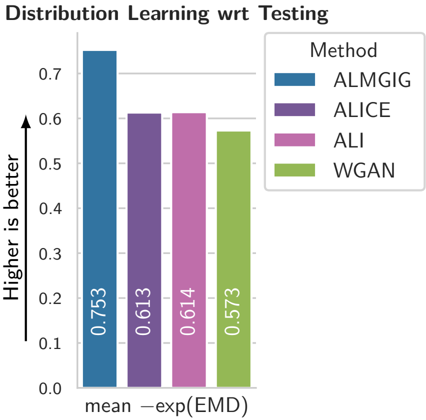


Comparison – Advanced Metrics

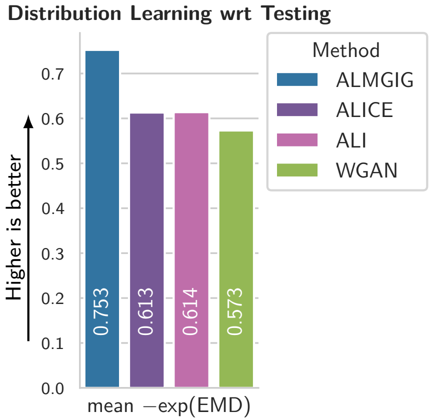


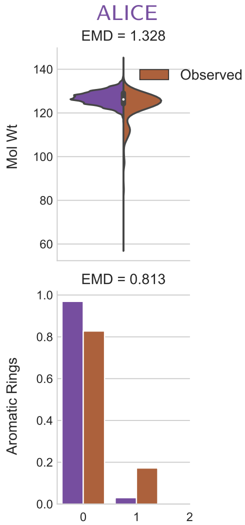


Comparison – Adversarial Learning Scheme



Comparison – Adversarial Learning Scheme





Conclusion

1. ALMGIG allows **training without computing a reconstructing loss**, which would require solving an expensive graph isomorphism problem.

Conclusion

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

1. ALMGIG allows **training without computing a reconstructing loss**, which would require solving an expensive graph isomorphism problem.
2. ALMGIG more **accurately represents the distribution over the space of molecules** than previous methods.

Conclusion

1. ALMGIG allows **training without computing a reconstructing loss**, which would require solving an expensive graph isomorphism problem.
2. ALMGIG more **accurately represents the distribution over the space of molecules** than previous methods.
3. Common validation metrics **validity, novelty, and uniqueness are insufficient** to properly assess the performance of methods.

Conclusion

1. ALMGIG allows **training without computing a reconstructing loss**, which would require solving an expensive graph isomorphism problem.
2. ALMGIG more **accurately represents the distribution over the space of molecules** than previous methods.
3. Common validation metrics **validity, novelty, and uniqueness are insufficient** to properly assess the performance of methods.
4. Distributions of **chemical descriptors provide detailed insight** into what type of molecules a model can generate.

Conclusion

1. ALMGIG allows **training without computing a reconstructing loss**, which would require solving an expensive graph isomorphism problem.
2. ALMGIG more **accurately represents the distribution over the space of molecules** than previous methods.
3. Common validation metrics **validity, novelty, and uniqueness are insufficient** to properly assess the performance of methods.
4. Distributions of **chemical descriptors provide detailed insight** into what type of molecules a model can generate.
5. Code available at `https://github.com/ai-med/almgig`

References I

Bradshaw, J., B. Paige, M. J. Kusner, M. Segler, and J. M. Hernández-Lobato (2019). "A Model to Search for Synthesizable Molecules". In: *Advances in Neural Information Processing Systems 32*, pp. 7937–7949.

Brown, N., M. Fiscato, M. H. Segler, and A. C. Vaucher (2019). "GuacaMol: Benchmarking Models for de Novo Molecular Design". In: *Journal of Chemical Information and Modeling* 59.3, pp. 1096–1108.

De Cao, N. and T. Kipf (2018). "MolGAN: An implicit generative model for small molecular graphs". In:

Dumoulin, V., I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville (2017). "Adversarially learned inference". In: *5th International Conference on Learning Representations*.

Jang, E., S. Gu, and B. Poole (2017). "Categorical Reparameterization with Gumbel-Softmax". In: *5th International Conference on Learning Representations*.

Jin, W., R. Barzilay, and T. Jaakkola (2018). "Junction Tree Variational Autoencoder for Molecular Graph Generation". In: *35th International Conference on Machine Learning*, pp. 2323–2332.

Kusner, M. J., B. Paige, and J. M. Hernández-Lobato (2017). "Grammar Variational Autoencoder". In: *34th International Conference on Machine Learning*, pp. 1945–1954.

Li, C., H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin (2017). “ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching”. In: *Advances in Neural Information Processing Systems 30*, pp. 5495–5503.

Li, Y., L. Zhang, and Z. Liu (2018). “Multi-objective de novo drug design with conditional graph generative model”. In: *Journal of Cheminformatics* 10, p. 33.

Li, Y., O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia (2018). “Learning Deep Generative Models of Graphs”. In:

Liu, Q., M. Allamanis, M. Brockschmidt, and A. Gaunt (2018). “Constrained Graph Variational Autoencoders for Molecule Design”. In: *Advances in Neural Information Processing Systems 31*, pp. 7806–7815.

Ma, T., J. Chen, and C. Xiao (2018). “Constrained Generation of Semantically Valid Graphs via Regularizing Variational Autoencoders”. In: *Advances in Neural Information Processing Systems 31*, pp. 7113–7124.

Maddison, C. J., A. Mnih, and Y. W. Teh (2017). “The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables”. In: *5th International Conference on Learning Representations*.

References III

Podda, M., D. Bacciu, and A. Micheli (2020). "A Deep Generative Model for Fragment-Based Molecule Generation". In: *Proc. of AISTATS.*

Polishchuk, P. G., T. I. Madzhidov, and A. Varnek (2013). "Estimation of the size of drug-like chemical space based on GDB-17 data". In: *Journal of Computer-Aided Molecular Design* 27.8, pp. 675–679.

Samanta, B., A. De, G. Jana, N. Ganguly, and M. Gomez-Rodriguez (2019). "NeVAE: A Deep Generative Model for Molecular Graphs". In: *33rd AAAI Conference on Artificial Intelligence*, pp. 1110–1117.

Simonovsky, M. and N. Komodakis (2018). "GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders". In: *ICANN*, pp. 412–422.

Xu, K., W. Hu, J. Leskovec, and S. Jegelka (2019). "How Powerful are Graph Neural Networks?" In: *7th International Conference on Learning Representations.*

You, J., B. Liu, R. Ying, V. Pande, and J. Leskovec (2018). "Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation". In: *Advances in Neural Information Processing Systems 31*, pp. 6412–6422.