Advances in Machine Learning for Molecules

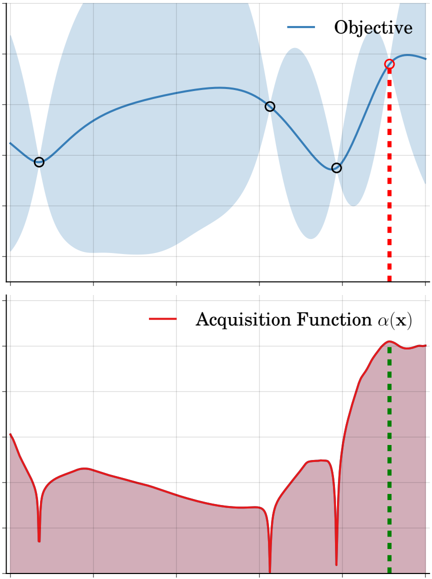
José Miguel Hernández–Lobato
Department of Engineering
University of Cambridge,
Microsoft Research Cambridge,
Alan Turing Institute

http://jmhl.org, jmh233@cam.ac.uk

Machine Learning Summer School, Madrid, 2018.

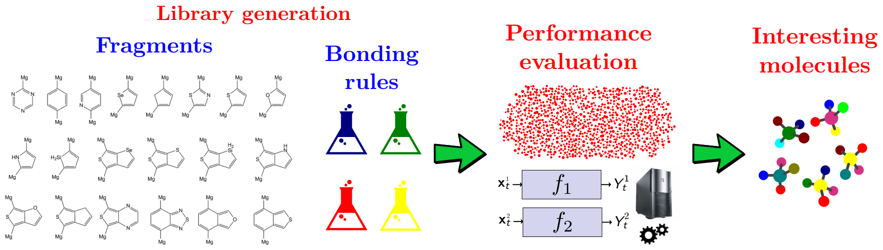
Part III: Bayesian Optimization of Molecules

Bayesian optimization



1. Get initial sample.
2. Fit a model to the data:
$$p(y|\mathbf{x}, \mathcal{D}_n)\,.$$
3. Select data collection strategy:
$$\alpha(\mathbf{x}) = \mathbf{E}_{p(y|\mathbf{x},\mathcal{D}_n)}[U(y|\mathbf{x}, \mathcal{D}_n)]\,.$$
4. Optimize acquisition function $\alpha(\mathbf{x})$.
5. Collect data and update model.
6. Repeat!

Discovering new optimal molecules



Library generation

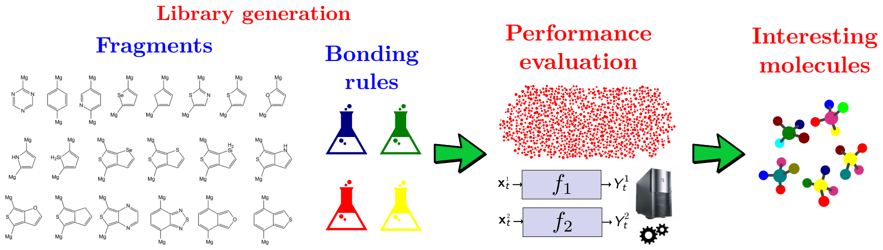
Fragments

Bonding rules

Performance evaluation

Interesting molecules

Discovering new optimal molecules



Library generation

Fragments

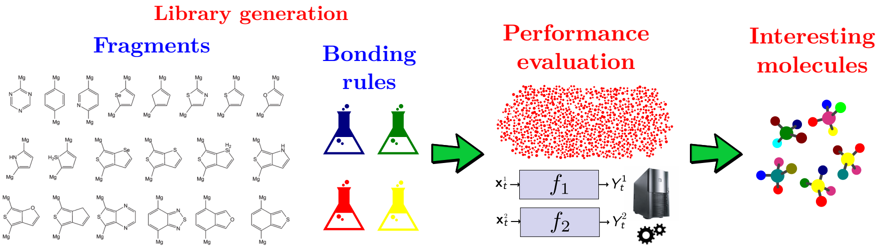
Bonding rules

Performance evaluation

Interesting molecules

Bayesian optimization (BO) can **accelerate** the search!

Discovering new optimal molecules



Library generation

Fragments

Bonding rules

Performance evaluation

Interesting molecules

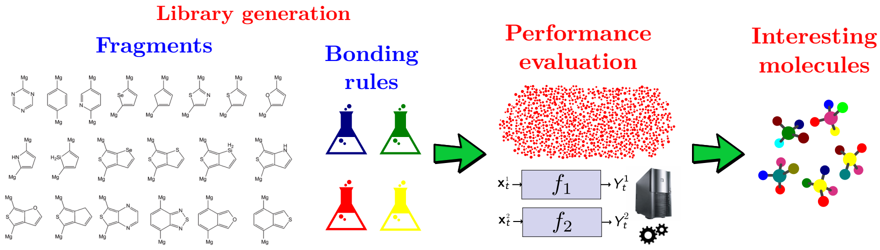
$\mathbf{x}_t^1 \rightarrow$ f_1 $\rightarrow Y_t^1$
$\mathbf{x}_t^2 \rightarrow$ f_2 $\rightarrow Y_t^2$

Bayesian optimization (BO) can **accelerate** the search!

Challenges:

- The search space is discrete and structured.
- BO output should exhibit regularities found in real-world molecules.

Discovering new optimal molecules



Library generation

Fragments

Bonding rules

Performance evaluation

Interesting molecules

Bayesian optimization (BO) can **accelerate** the search!

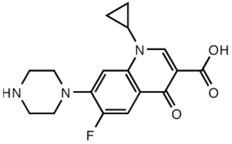
Challenges:

- The search space is discrete and structured.
- BO output should exhibit regularities found in real-world molecules.

Solution: combine BO methods with **generative models** of molecules.

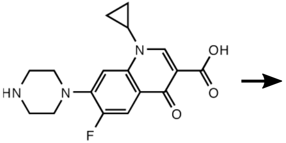
Gómez-Bombarelli*, Wei*, Duvenaud*, Hernández-Lobato*, Sánchez-Lengeling, Sheberla, Aguilera-Iparraguirre, Hirzel, Adams and Aspuru-Guzik, 2018. (* equal contributors).

Molecules encoded as **character strings** using SMILES language.



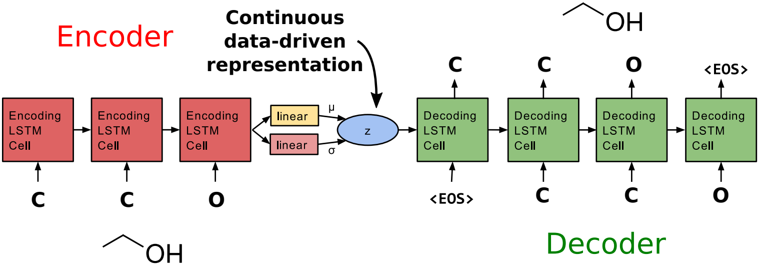
→ N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

Molecules encoded as **character strings** using SMILES language.



N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

A **generative model** on SMILES is obtained using a **seq2seq** **VAE**.



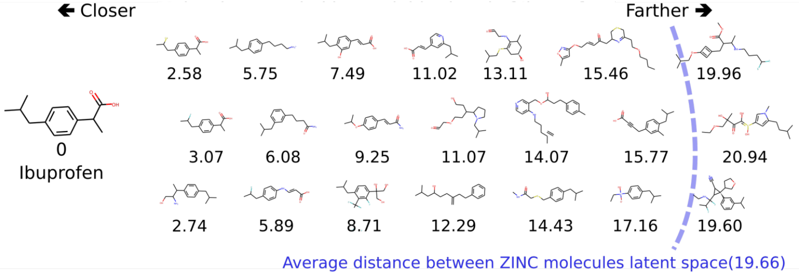
Sampled molecules have **statistics similar** to those of real molecules:

source	data set	logP	SAS	QED
Data	ZINC	2.46 (1.43)	3.05 (0.83)	0.73 (0.14)
GA	ZINC	2.84 (1.86)	3.80 (1.01)	0.57 (0.20)
VAE	ZINC	2.67 (1.46)	3.18 (0.86)	0.70 (0.14)
Data	QM9	0.30 (1.00)	4.25 (0.94)	0.48 (0.07)
GA	QM9	0.96 (1.53)	4.47 (1.01)	0.53 (0.13)
VAE	QM9	0.30 (0.97)	4.34 (0.98)	0.47 (0.08)

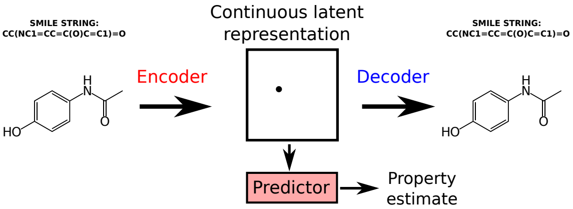
Sampled molecules have statistics similar to those of real molecules:

source	data set	logP	SAS	QED
Data	ZINC	2.46 (1.43)	3.05 (0.83)	0.73 (0.14)
GA	ZINC	2.84 (1.86)	3.80 (1.01)	0.57 (0.20)
VAE	ZINC	2.67 (1.46)	3.18 (0.86)	0.70 (0.14)
Data	QM9	0.30 (1.00)	4.25 (0.94)	0.48 (0.07)
GA	QM9	0.96 (1.53)	4.47 (1.01)	0.53 (0.13)
VAE	QM9	0.30 (0.97)	4.34 (0.98)	0.47 (0.08)

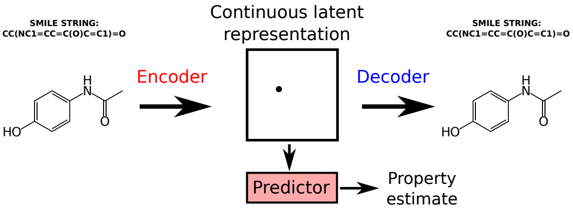
Nearby latent representations decode into similar molecules:



Easy to add a **surrogate model** from latent space to property.



Easy to add a **surrogate model** from latent space to property.



SMILE STRING:
CC(NC1=CC=C(O)C=C1)=O

Encoder

Continuous latent representation

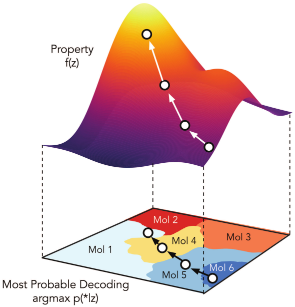
Decoder

SMILE STRING:
CC(NC1=CC=C(O)C=C1)=O

Predictor

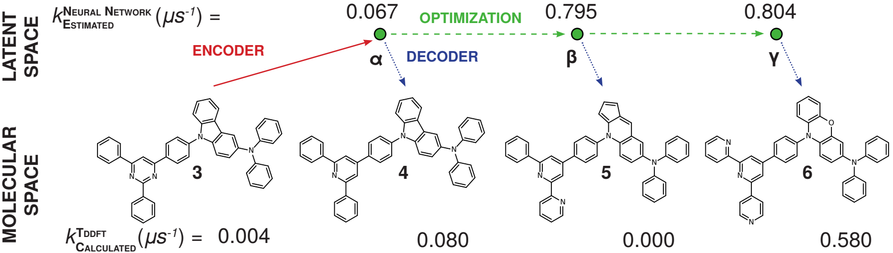
Property estimate

Gradient-based optimization can be used in latent space.



Local optimization of OLED molecules

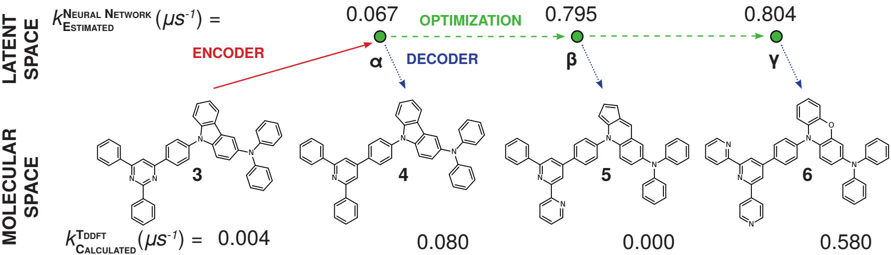
We optimize the delayed fluorescence decay rate κ_{TADF}, as estimated from TDDFT computations on 150,000 molecules.



LATENT SPACE
$k^{\text{Neural Network}}_{\text{Estimated}}$ (μs^{-1}) =
0.067
OPTIMIZATION
0.795
0.804
ENCODER
α
DECODER
β
γ
MOLECULAR SPACE
3
4
5
6
N
O
$k^{\text{TDDFT}}_{\text{Calculated}}$($\mu s^{-1}$) = 0.004
0.080
0.000
0.580

Local optimization of OLED molecules

We optimize the delayed fluorescence decay rate κ_{TADF}, as estimated from TDDFT computations on 150,000 molecules.



LATENT SPACE

$k^{\text{Neural Network}}_{\text{Estimated}}$ (μs^{-1}) = 0.067 0.795 0.804

OPTIMIZATION

ENCODER

DECODER

α β γ

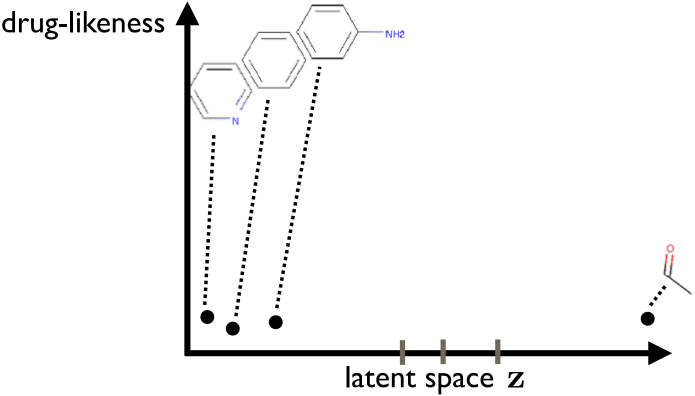
MOLECULAR SPACE

3 4 5 6

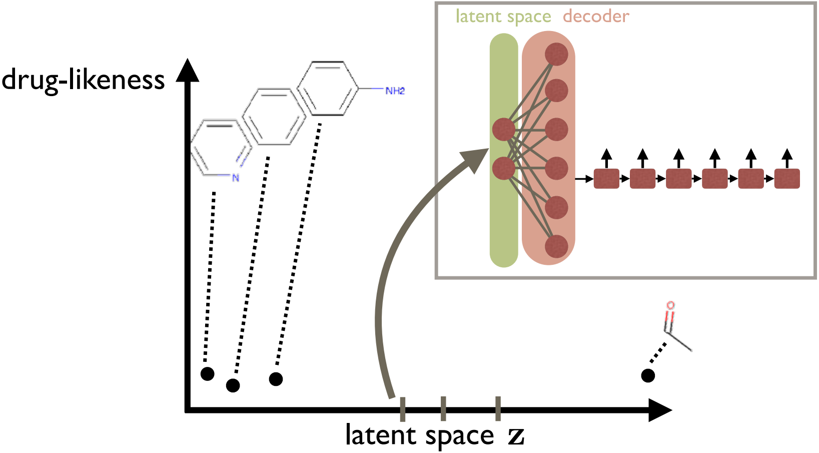
$k^{\text{TDDFT}}_{\text{Calculated}}$ (μs^{-1}) = 0.004 0.080 0.000 0.580

However, many of the sampled SMILES strings are not valid molecules.

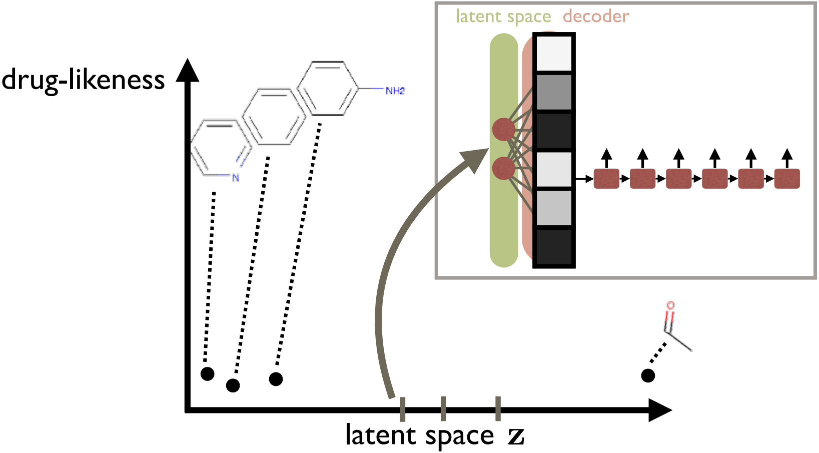
THE PROBLEM



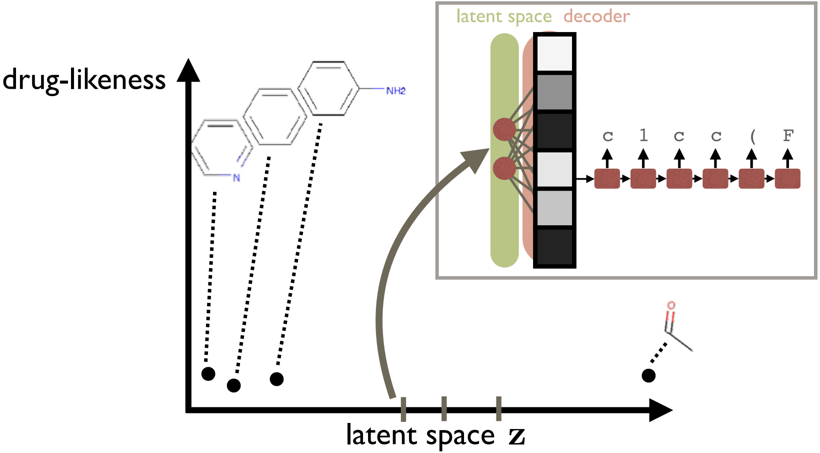
THE PROBLEM


latent space
decoder
drug-likeness
NH2
N
O
latent space $\mathbf{z}$

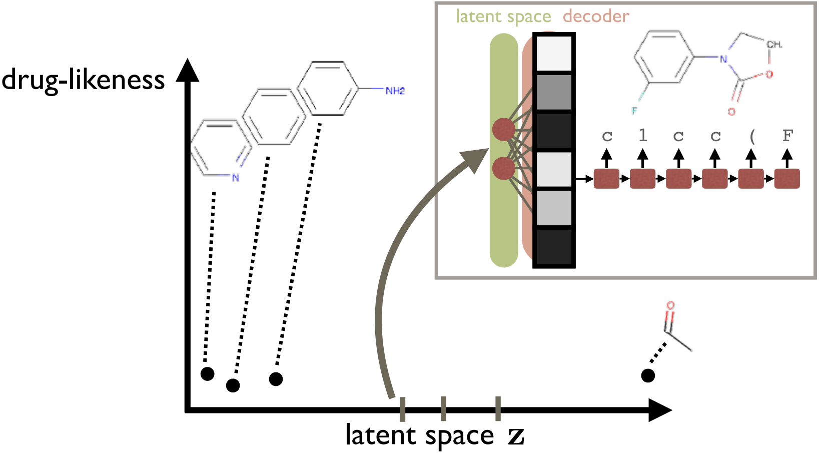
THE PROBLEM



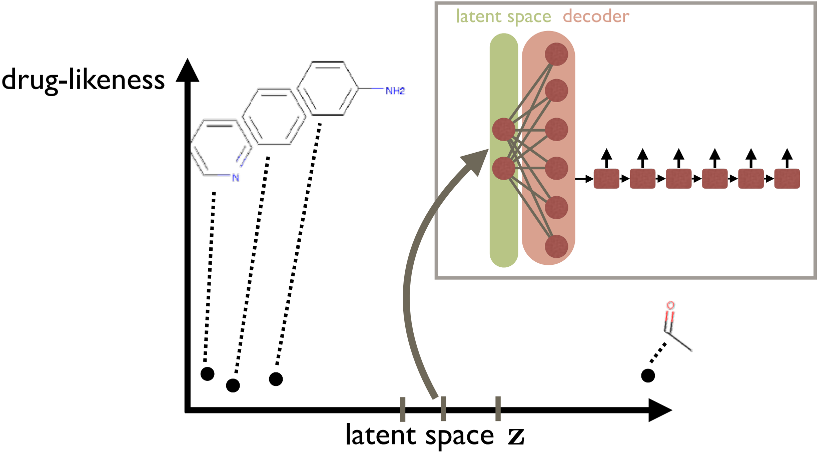
THE PROBLEM


drug-likeness
N
NH2
latent space
decoder
c
1
c
c
(
F
O
latent space $\mathbf{z}$

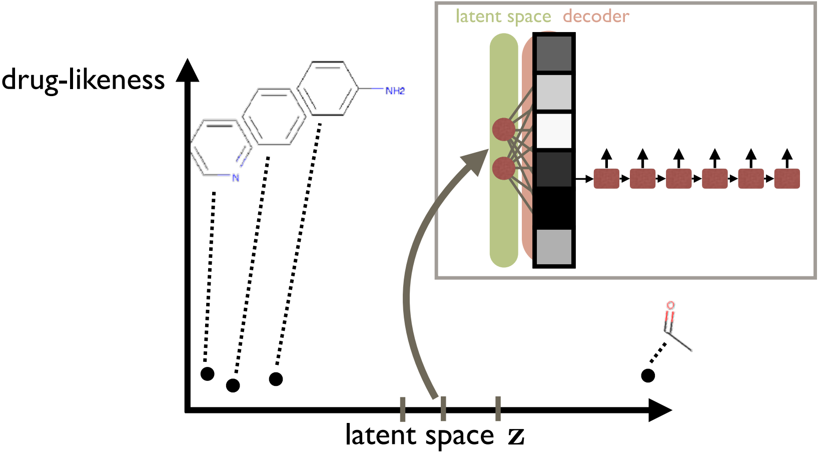
THE PROBLEM


drug-likeness
N
NH2
latent space
decoder
N
CH
O
O
F
c
1
c
c
(
F
O
latent space z

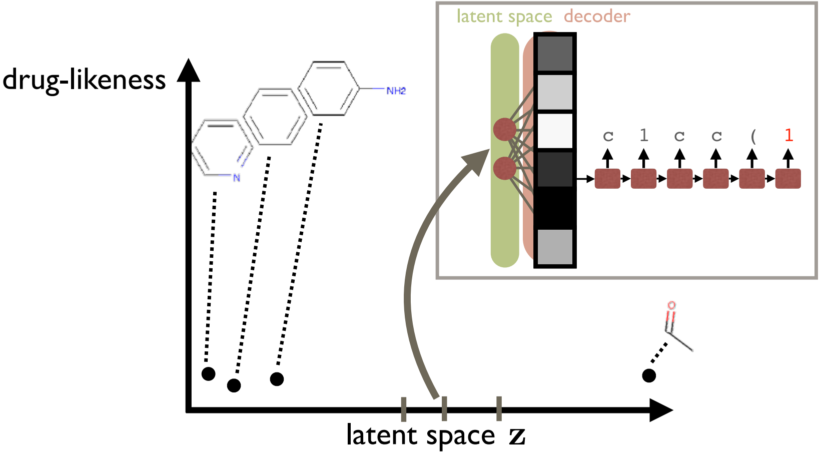
SMALL CHANGES RUIN EVERYTHING



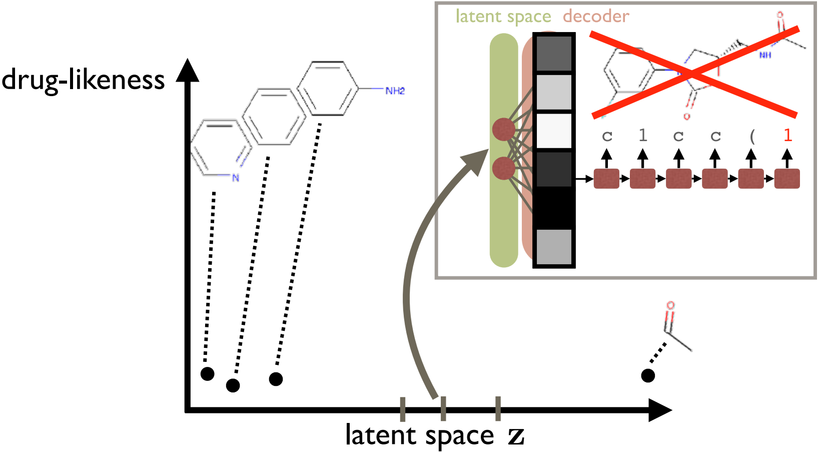
SMALL CHANGES RUIN EVERYTHING



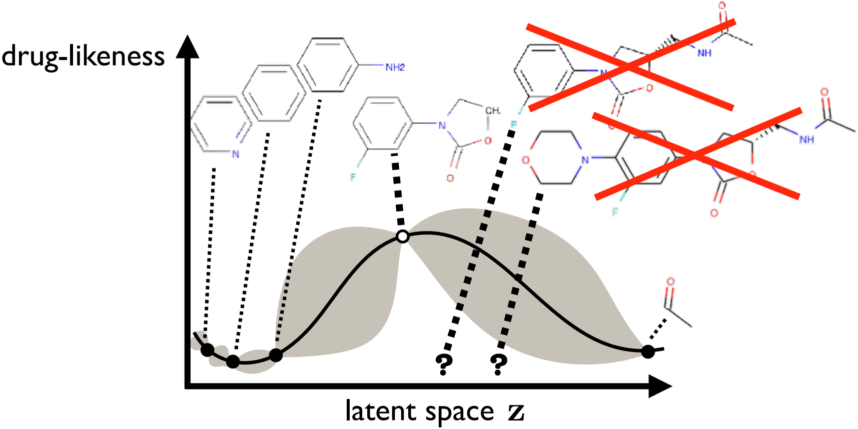
SMALL CHANGES RUIN EVERYTHING



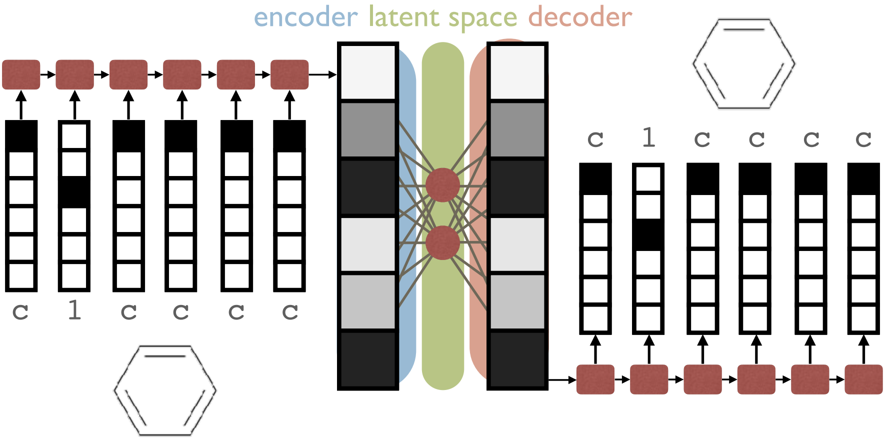
SMALL CHANGES RUIN EVERYTHING



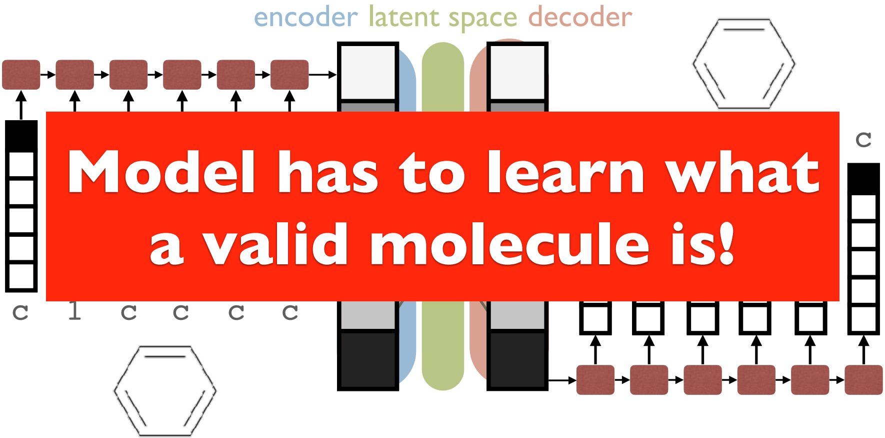
SMALL CHANGES RUIN EVERYTHING



DISCRETE GENERATIVE MODELS: CAN WE DO BETTER?



DISCRETE GENERATIVE MODELS: CAN WE DO BETTER?



Model has to learn what a valid molecule is!

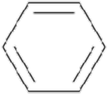
A NEW REPRESENTATION

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'



c1ccccc1

A NEW REPRESENTATION

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

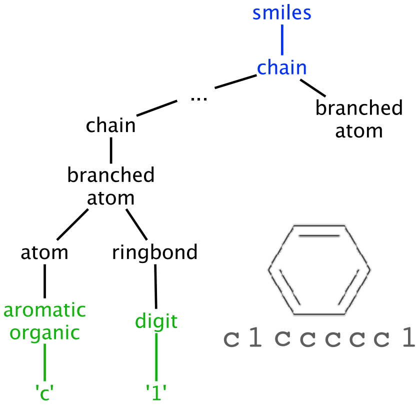
aromatic organic → 'c'

aliphatic organic → 'C'

aliphatic organic → 'N'

digit → '1'

digit → '2'



A NEW SEQUENCE!

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

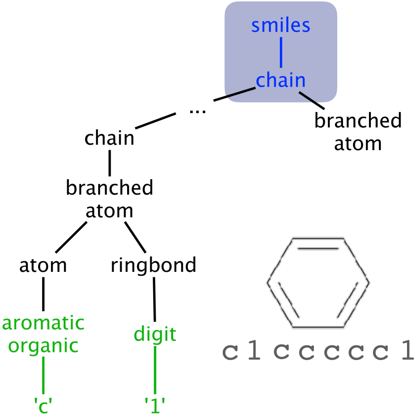
aromatic organic → 'c'

aliphatic organic → 'C'

aliphatic organic → 'N'

digit → '1'

digit → '2'



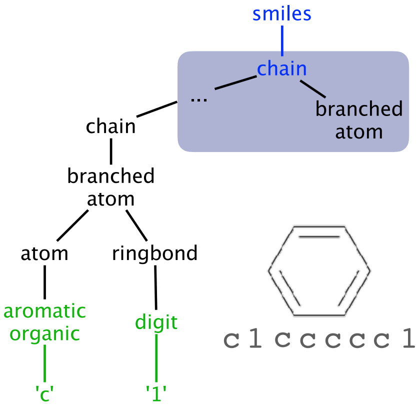
A NEW SEQUENCE!

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'



c1ccccc1

A NEW SEQUENCE!

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

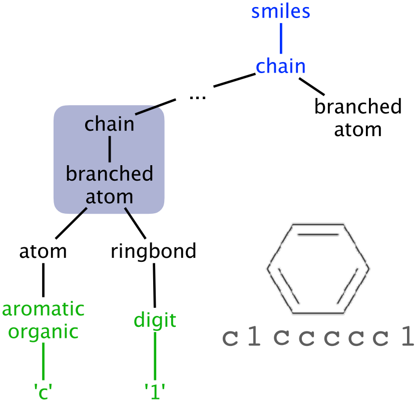
aromatic organic → 'c'

aliphatic organic → 'C'

aliphatic organic → 'N'

digit → '1'

digit → '2'



c 1 c c c c c 1

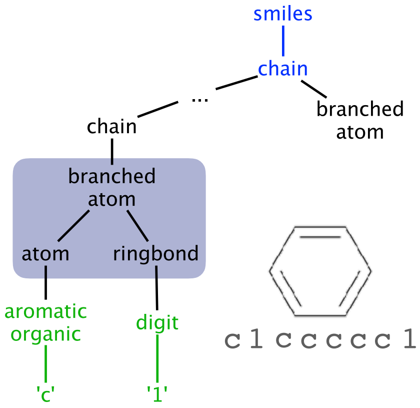
A NEW SEQUENCE!

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'



c1ccccc1

A NEW SEQUENCE!

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

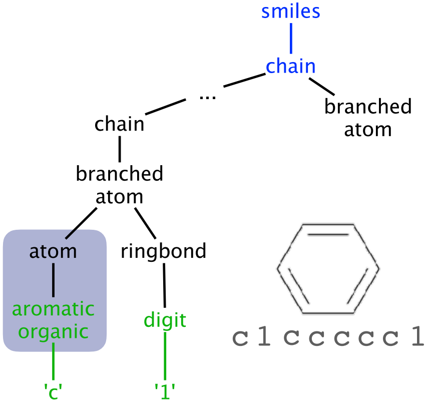
aromatic organic → 'c'

aliphatic organic → 'C'

aliphatic organic → 'N'

digit → '1'

digit → '2'



A NEW SEQUENCE!

[Kusner, Paige, Hernández-Lobato, 2017]

Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

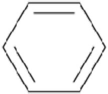
aromatic organic → 'c'

aliphatic organic → 'C'

aliphatic organic → 'N'

digit → '1'

digit → '2'



smiles → chain

chain → chain, branched atom

...

chain → branched atom

branched atom → atom, ringbond

atom → aromatic organic

aromatic organic → 'c'

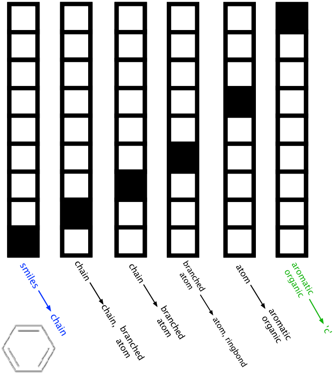
ringbond → digit

digit → '1'

...

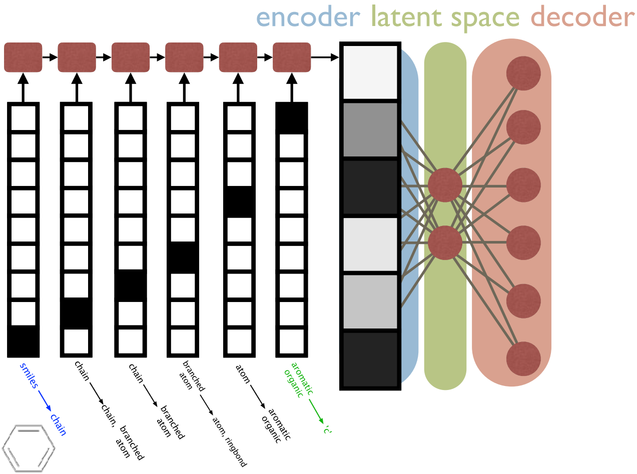
ENCODER

[Kusner, Paige, Hernández-Lobato, 2017]



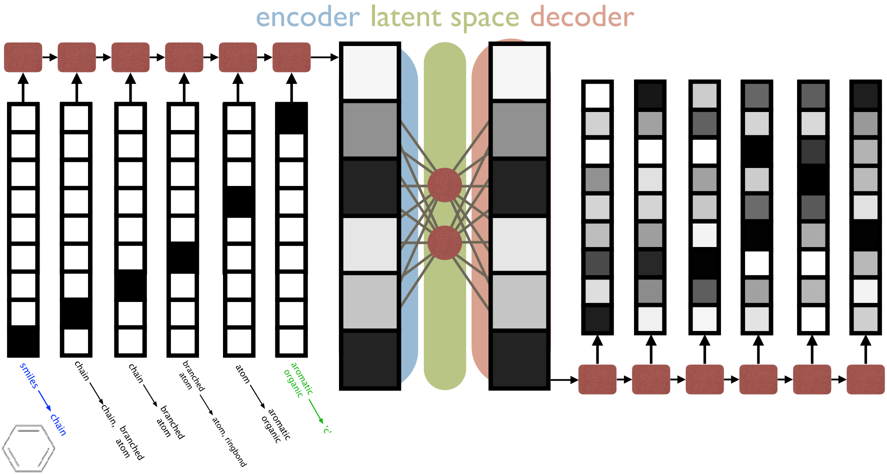
ENCODER

[Kusner, Paige, Hernández-Lobato, 2017]



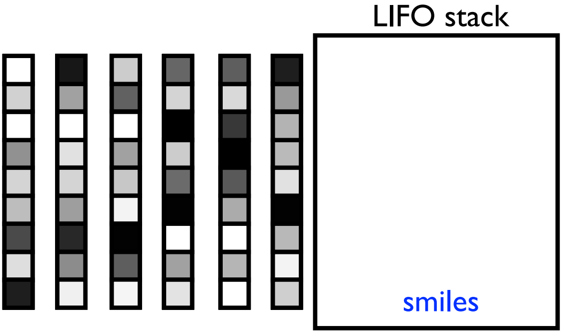
DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



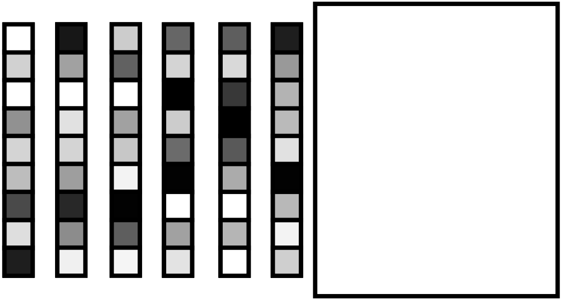
DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



DECODER

[Kusner, Paige, Hernández-Lobato, 2017]

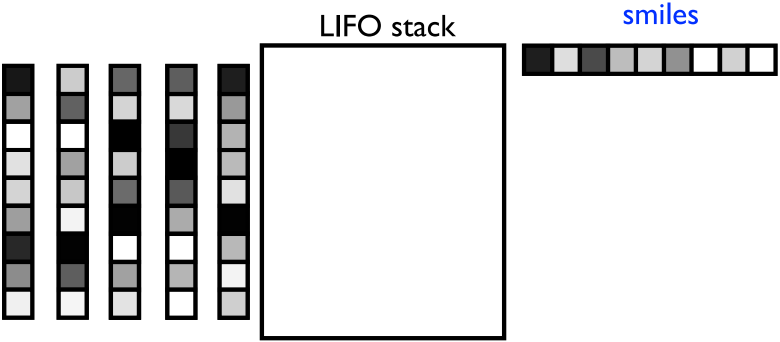


LIFO stack

smiles

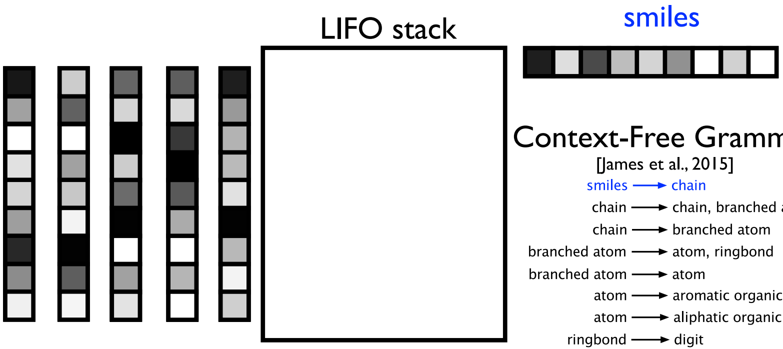
DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

aromatic organic → 'c'

aliphatic organic → 'C'

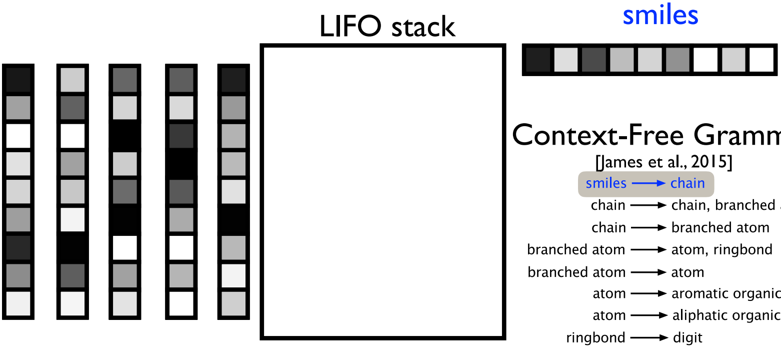
aliphatic organic → 'N'

digit → '1'

digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



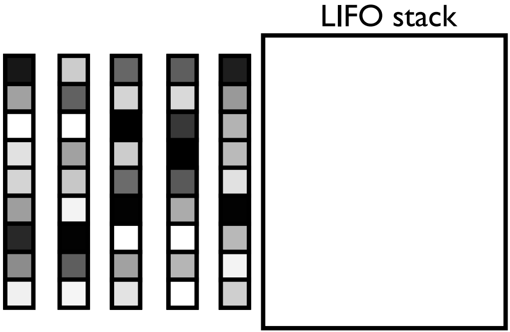
Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



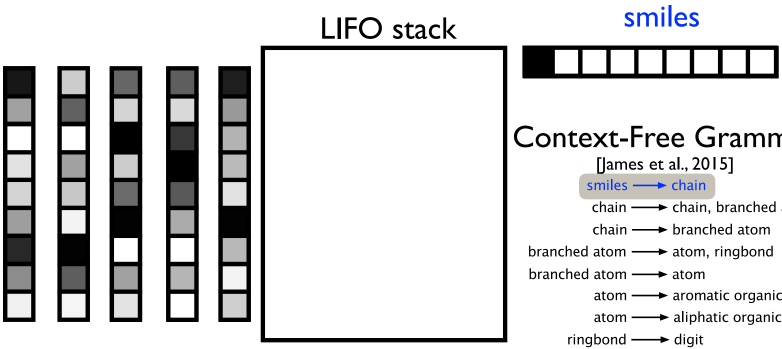
Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



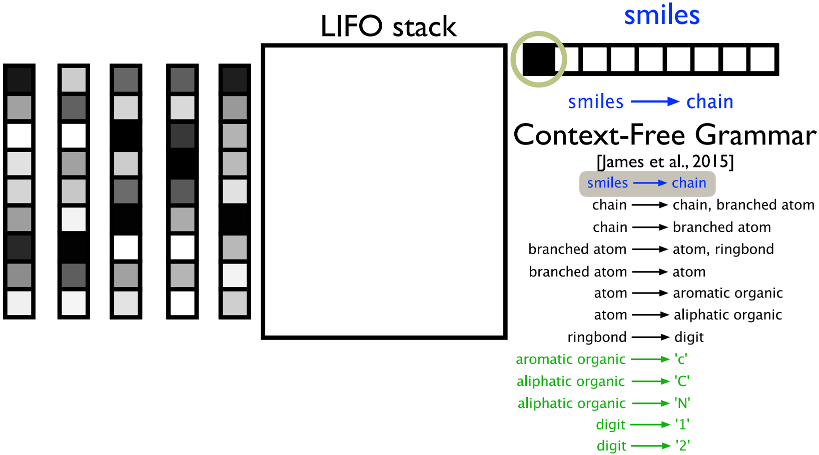
Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



LIFO stack

smiles

smiles → chain

Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

aromatic organic → 'c'

aliphatic organic → 'C'

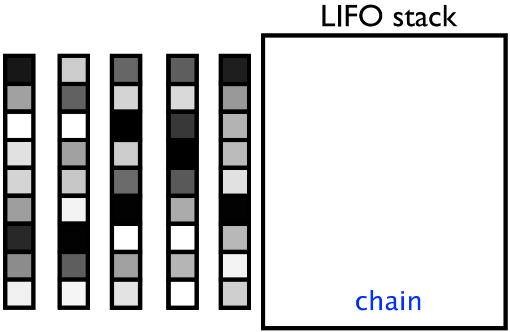
aliphatic organic → 'N'

digit → '1'

digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



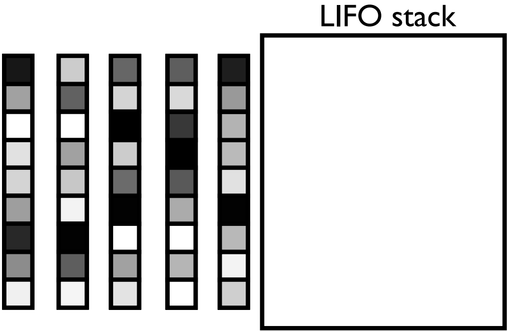
Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



chain

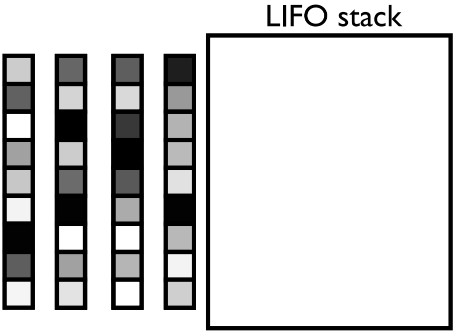
Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



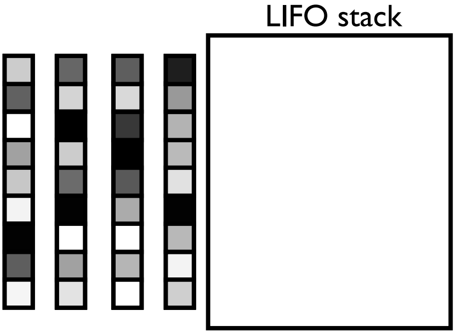
Context-Free Grammar

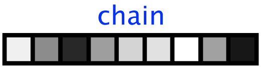
[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]





Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

aromatic organic → 'c'

aliphatic organic → 'C'

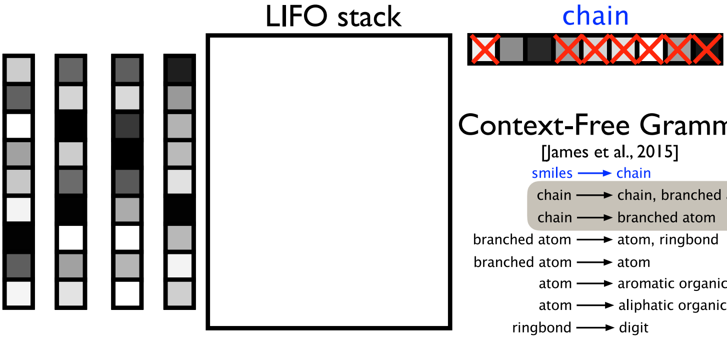
aliphatic organic → 'N'

digit → '1'

digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



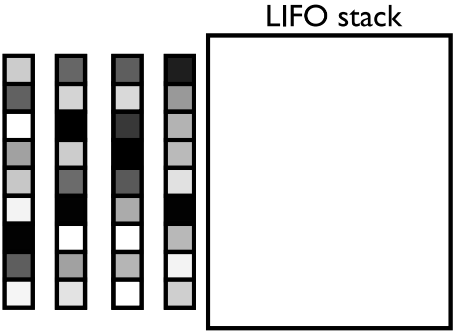
Context-Free Grammar

[James et al., 2015]

smiles → chain
chain → chain, branched atom
chain → branched atom
branched atom → atom, ringbond
branched atom → atom
atom → aromatic organic
atom → aliphatic organic
ringbond → digit
aromatic organic → 'c'
aliphatic organic → 'C'
aliphatic organic → 'N'
digit → '1'
digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

aromatic organic → 'c'

aliphatic organic → 'C'

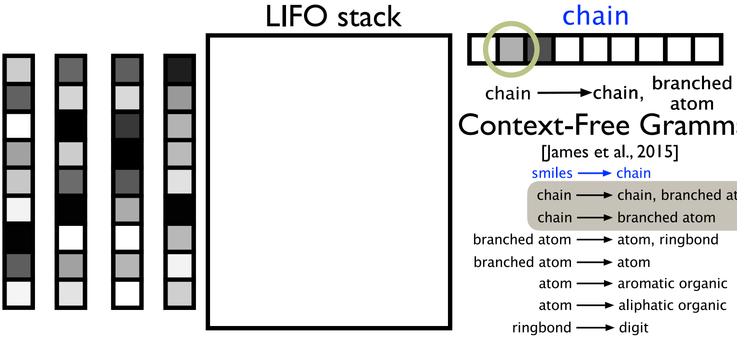
aliphatic organic → 'N'

digit → '1'

digit → '2'

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



Context-Free Grammar

[James et al., 2015]

smiles → chain

chain → chain, branched atom

chain → branched atom

branched atom → atom, ringbond

branched atom → atom

atom → aromatic organic

atom → aliphatic organic

ringbond → digit

aromatic organic → 'c'

aliphatic organic → 'C'

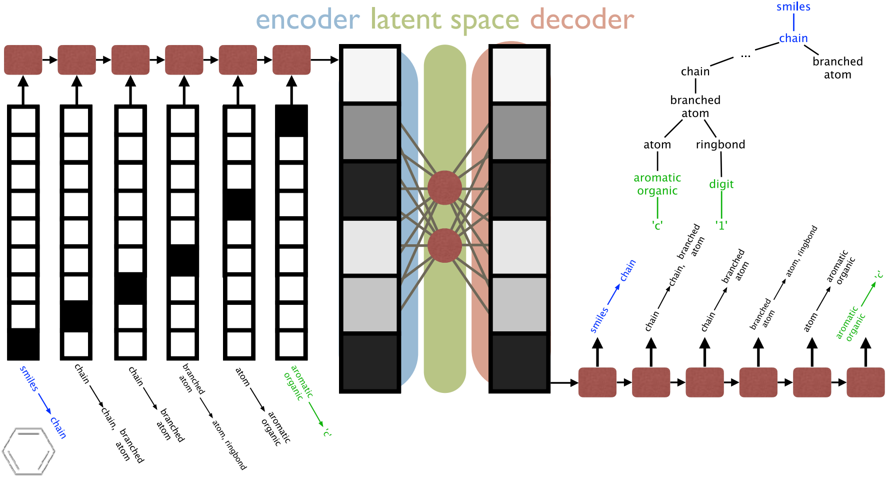
aliphatic organic → 'N'

digit → '1'

digit → '2'

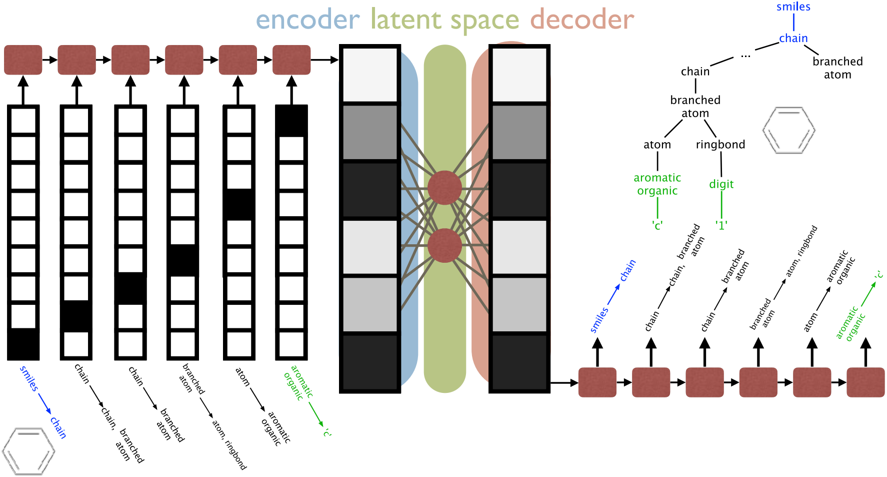
DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



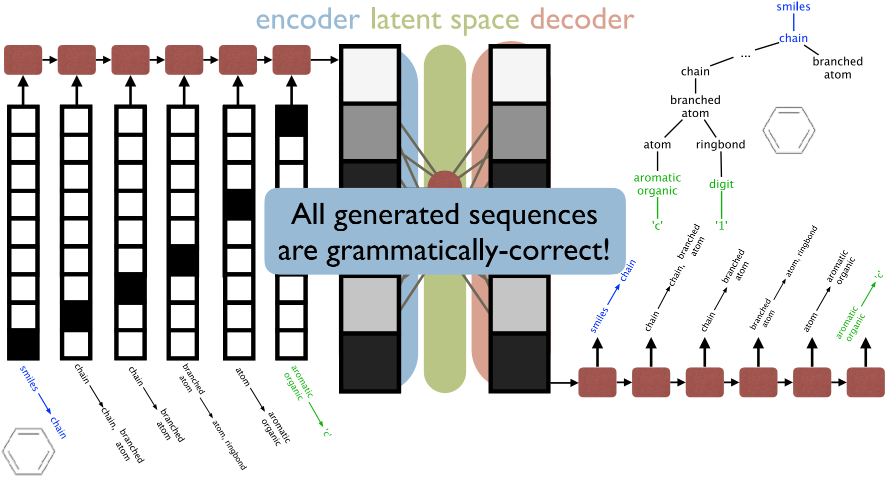
DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



DECODER

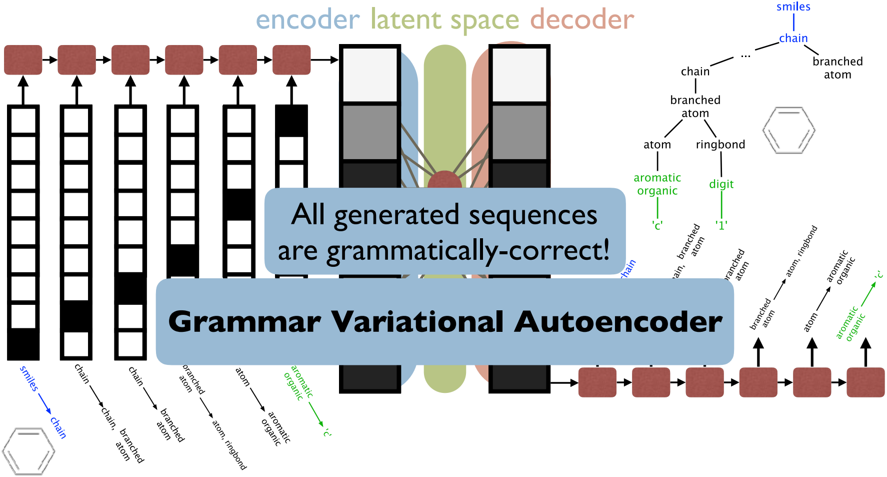
[Kusner, Paige, Hernández-Lobato, 2017]



All generated sequences are grammatically-correct!

DECODER

[Kusner, Paige, Hernández-Lobato, 2017]



All generated sequences are grammatically-correct!

Grammar Variational Autoencoder

HOW WELL DOES THIS WORK?

Reconstruction accuracy and sample validity results.

Method	% Reconstruct	% Prior Valid
GVAE	**53.7**	**7.2**
CVAE	44.6	0.70

Results finding best molecule

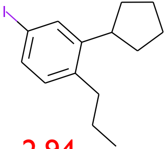
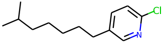
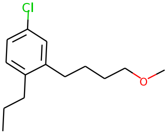
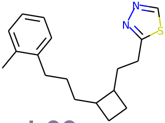
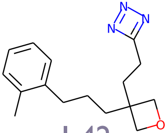
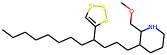
Method	Frac. valid	Avg. score
GVAE	**0.31±0.07**	**-9.57 ±1.77**
CVAE	0.17±0.05	-54.66±2.66

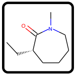
Test Log-likelihood (LL) and RMSE for predictions from latent space

Objective	Method	Molecules
LL	GVAE	**-1.739 ±0.004**
	CVAE	-1.812±0.004
RMSE	GVAE	**1.404 ±0.006**
	CVAE	1.504±0.006

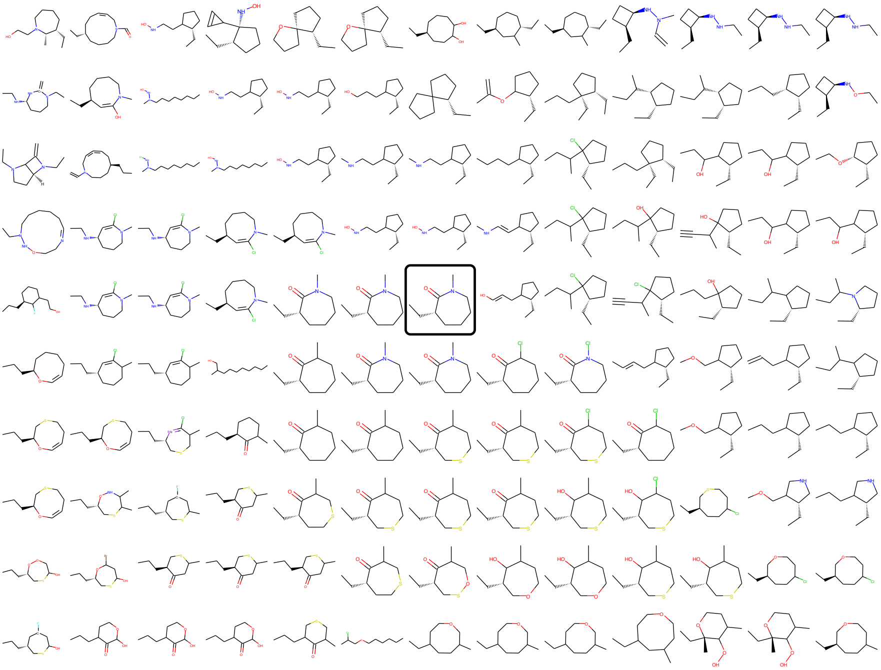
DRUG SEARCH

Trained on **ZINC**
250K molecules
maximum 120 chars

	1st	2nd	3rd
GVAE [Kusner, Paige, Hernández-Lobato, 2017]			
drug-likeness:	2.94	2.89	2.80
CVAE [Gómez-Bombarelli et al., 2016]			
drug-likeness:	1.98	1.42	1.19



O
N



Take home messages

Generative models of molecues...

1. **Data-driven approach** to molecule generation, no expertise needed.
2. Sampled molecules are **realistic**, unlike those generated with rules.
3. Create a **continuous latent space** which is useful for optimization.

Grammar variational autoencoder...

1. Produces a **larger fraction of valid molecules** when decoding.
2. Produces **better predictions of molecule properties** from latent space.
3. **Molecule optimization** results are **improved**.

A lot of work going on this area...

Adding semantic constraints into GVAE:

- Dai, Tian, Dai, Skiena, and Song, 2018.

Generative models of graphs:

- Li, Vinyals, Dyer, Pascanu and Battaglia, 2018.
- Liu, Allamanis, Brockschmidt and Gaunt, 2018.
- Jin, Barzilay and Jaakkola, 2018.

Semi-supervised generative models:

- Kang and Cho, 2018.

Using GANs insead of VAEs:

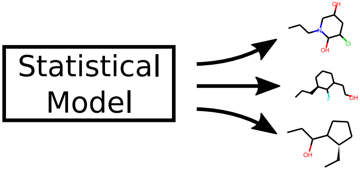
- Guimaraes, Sanchez-Lengeling, Outeiral, Farias and Aspuru-Guzik, 2017.
- De Cao and Kipf, 2018.

and many more!

Part IV: Predicting Electron Paths

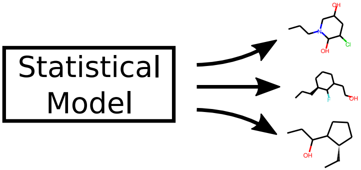
Motivation

Current generative models for molecules have **limitations**: will not tell you how to **synthesize** the generated molecules.

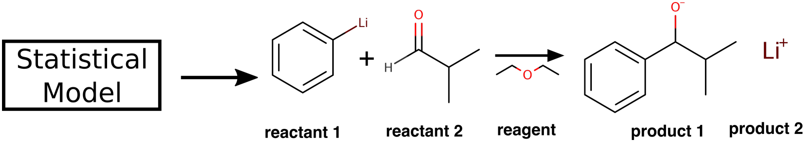


Motivation

Current generative models for molecules have **limitations**: will not tell you how to **synthesize** the generated molecules.

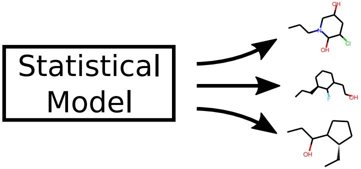


We want models that generate **synthetic routes** instead of just molecules!

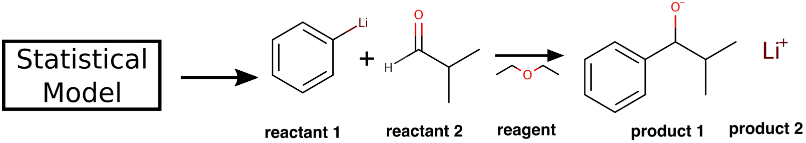


Motivation

Current generative models for molecules have **limitations**: will not tell you how to **synthesize** the generated molecules.



We want models that generate **synthetic routes** instead of just molecules!

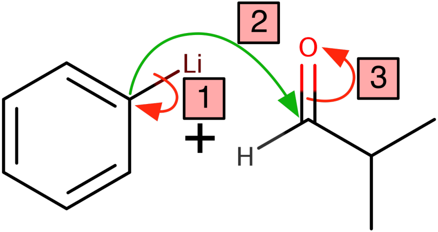


Perhaps too challenging. We just focus for now on reaction prediction.

Electron path

A chemical reaction is characterized by a **sequential movement of electrons** alternating between breaking and forming bonds.

Represented by an **arrow pushing diagram** showing the **path followed by electrons** (detailed steps of the reaction):

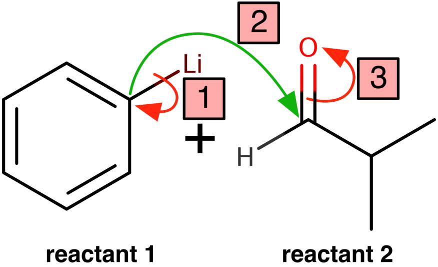


reactant 1 **reactant 2**

Electron path

A chemical reaction is characterized by a **sequential movement of electrons** alternating between breaking and forming bonds.

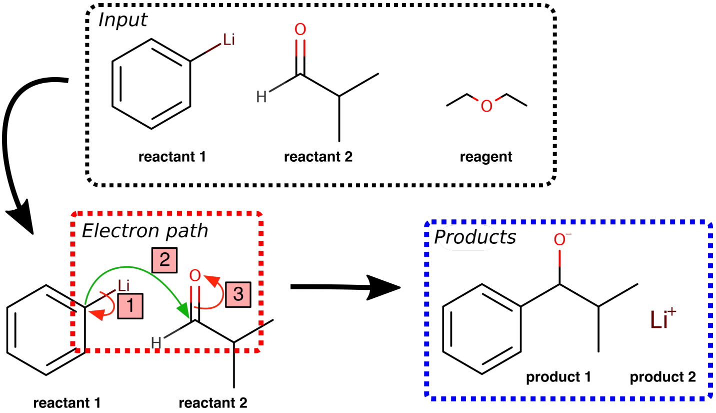
Represented by an **arrow pushing diagram** showing the **path followed by electrons** (detailed steps of the reaction):



Goal: predict the **electron path** from input molecules.

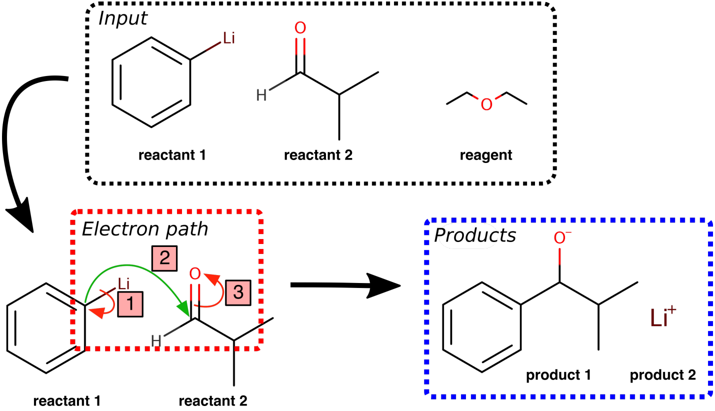
Prediction problem

Given the **electron path**, we directly obain the final **products** of the reaction.



Prediction problem

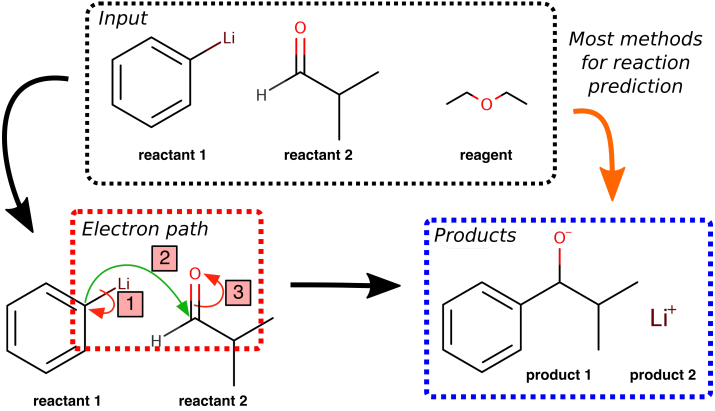
Given the **electron path**, we directly obain the final **products** of the reaction.



Most reaction prediction approaches directly **predict products** from **input**.

Prediction problem

Given the **electron path**, we directly obain the final **products** of the reaction.



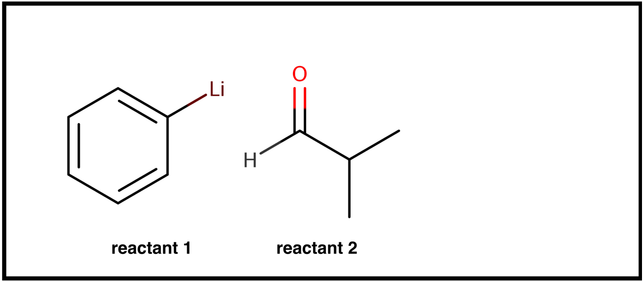
Most reaction prediction approaches directly **predict products** from **input**.

Why predict electron paths?

- **Easy to interpret**: if the model makes a mistake, its easy to see where, and maybe even why.
- **Sparse**: typically reactions only affect 3-7 atoms out of 10-50. Modeling an electron path enforces this sparsity.
- **Chemically consistent**: easy to incorporate chemical constraints directly into a model for predicting electron paths.
- **Generalizable**: as electron paths exhibit regularities across different reactions, we naturally generalize to new inputs.

ELECTRO: an algorithm for electron paths

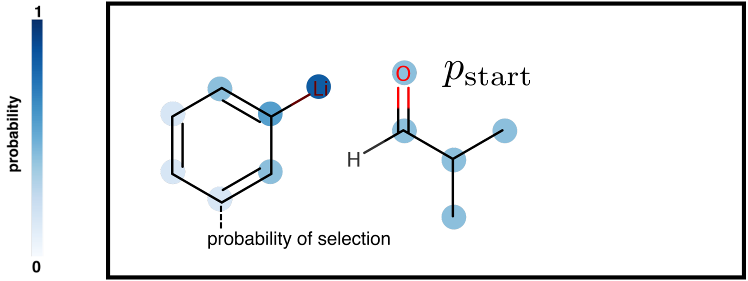
```
→ Select starting atom A.
  Repeat:
     Select atom B bonded to A and remove 2 electrons from bond A-B.
     Should we stop?
     Select new atom A and add bond B-A.
     Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

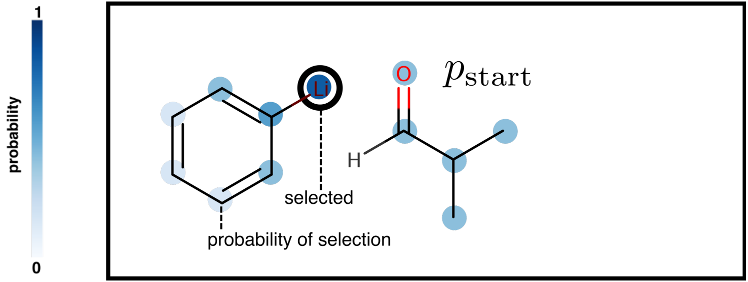
```
→ Select starting atom A.
  Repeat:
     Select atom B bonded to A and remove 2 electrons from bond A-B.
     Should we stop?
     Select new atom A and add bond B-A.
     Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

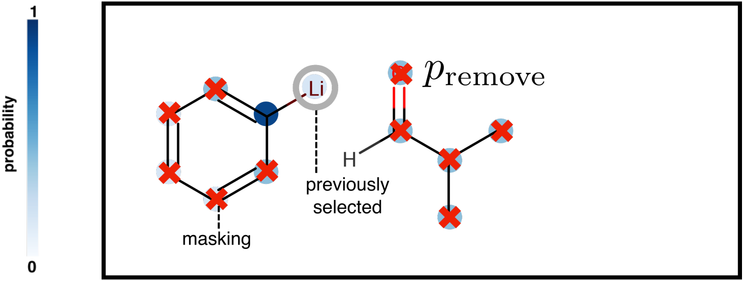
```
→ Select starting atom A.
  Repeat:
     Select atom B bonded to A and remove 2 electrons from bond A-B.
     Should we stop?
     Select new atom A and add bond B-A.
     Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

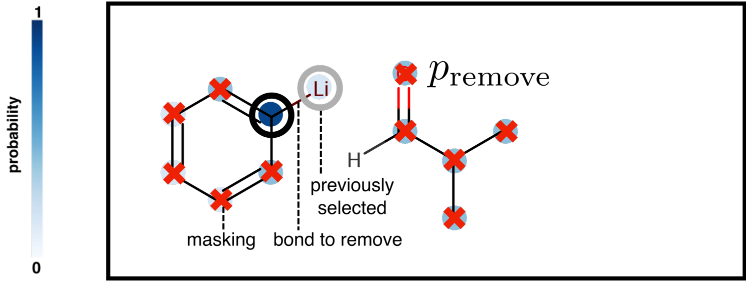
```
  Select starting atom A.
  Repeat:
→     Select atom B bonded to A and remove 2 electrons from bond A-B.
      Should we stop?
      Select new atom A and add bond B-A.
      Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

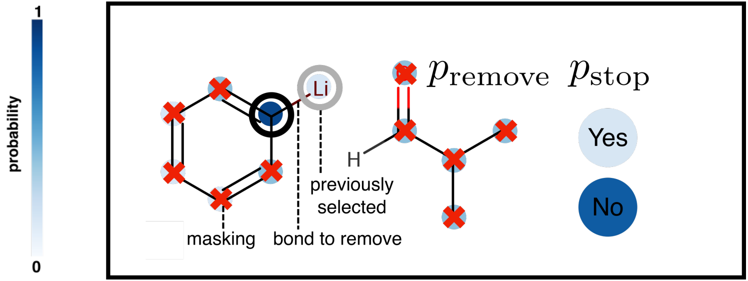
```
  Select **starting** atom A.
  Repeat:
→     Select atom B bonded to A and **remove** 2 electrons from bond A-B.
      Should we **stop**?
      Select new atom A and **add** bond B-A.
      Should we **stop**?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

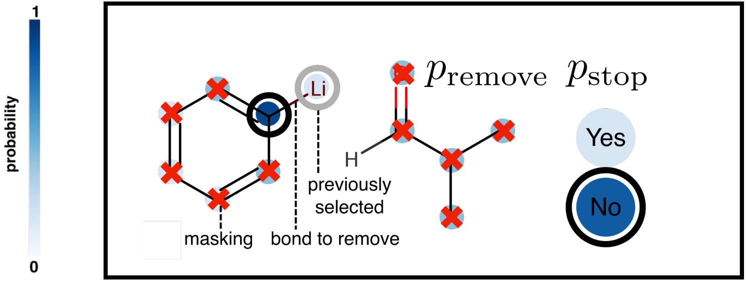
```
  Select starting atom A.
  Repeat:
      Select atom B bonded to A and remove 2 electrons from bond A-B.
→     Should we stop?
      Select new atom A and add bond B-A.
      Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

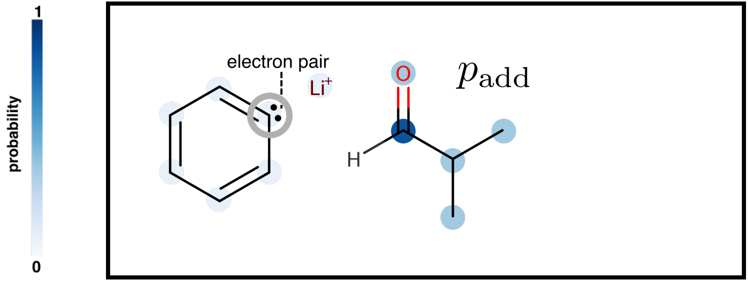
```
  Select starting atom A.
  Repeat:
    Select atom B bonded to A and remove 2 electrons from bond A-B.
→   Should we stop?
    Select new atom A and add bond B-A.
    Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

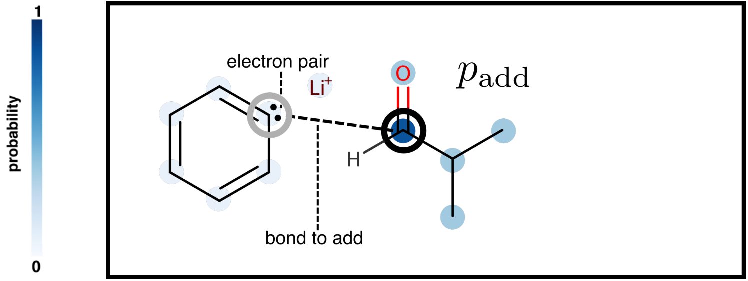
```
    Select starting atom A.
    Repeat:
        Select atom B bonded to A and remove 2 electrons from bond A-B.
        Should we stop?
→       Select new atom A and add bond B-A.
        Should we stop?
    Compute resulting electron path.
    Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

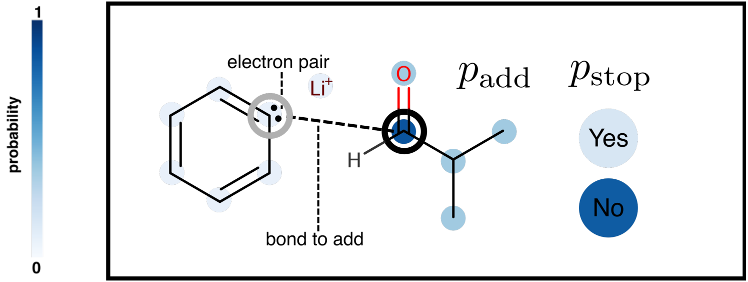
```
Select starting atom A.
Repeat:
    Select atom B bonded to A and remove 2 electrons from bond A-B.
    Should we stop?
→   Select new atom A and add bond B-A.
    Should we stop?
Compute resulting electron path.
Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

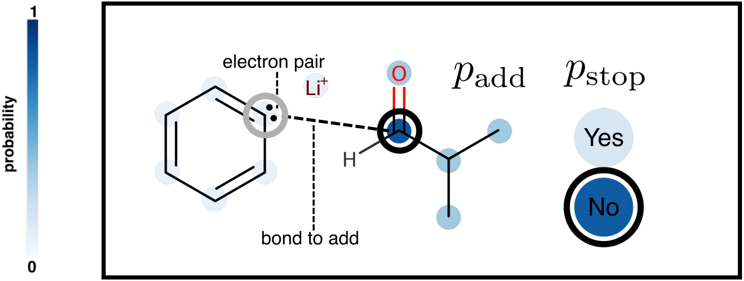
```
  Select starting atom A.
  Repeat:
     Select atom B bonded to A and remove 2 electrons from bond A-B.
     Should we stop?
     Select new atom A and add bond B-A.
→    Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

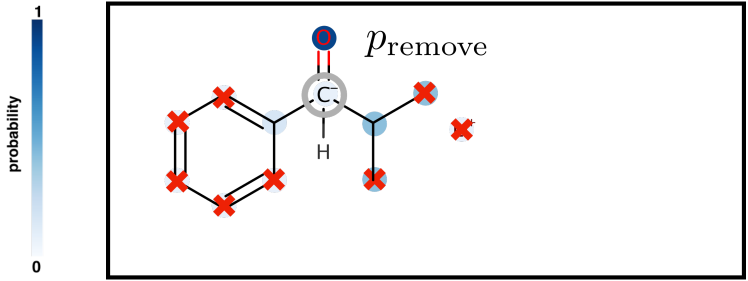
```
Select starting atom A.
Repeat:
    Select atom B bonded to A and remove 2 electrons from bond A-B.
    Should we stop?
    Select new atom A and add bond B-A.
→   Should we stop?
Compute resulting electron path.
Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

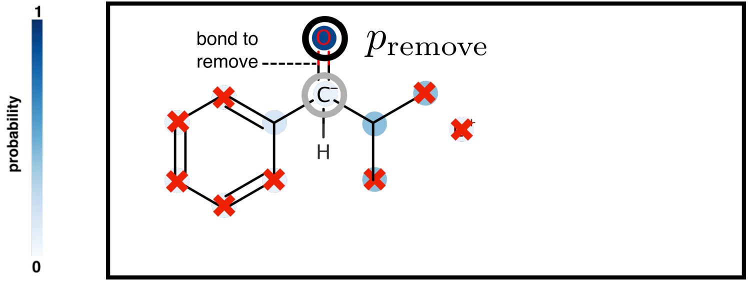
```
  Select starting atom A.
  Repeat:
→     Select atom B bonded to A and remove 2 electrons from bond A-B.
      Should we stop?
      Select new atom A and add bond B-A.
      Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

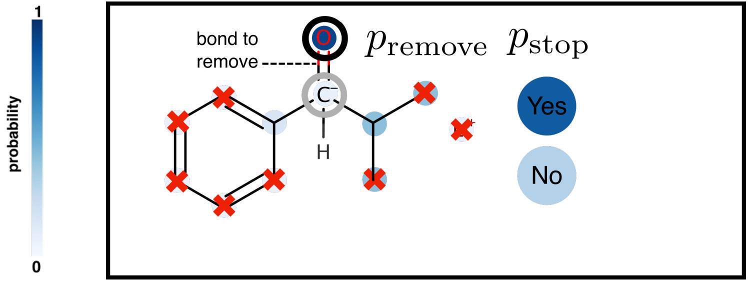
```
   Select **starting** atom A.
   Repeat:
→      Select atom B bonded to A and **remove** 2 electrons from bond A-B.
       Should we **stop**?
       Select new atom A and **add** bond B-A.
       Should we **stop**?
   Compute resulting electron path.
   Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

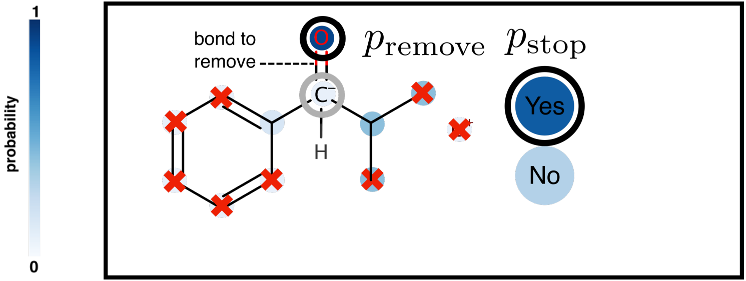
```
  Select starting atom A.
  Repeat:
      Select atom B bonded to A and remove 2 electrons from bond A-B.
→     Should we stop?
      Select new atom A and add bond B-A.
      Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

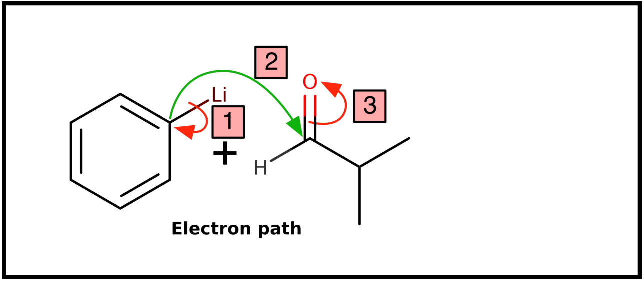
```
  Select starting atom A.
  Repeat:
     Select atom B bonded to A and remove 2 electrons from bond A-B.
→    Should we stop?
     Select new atom A and add bond B-A.
     Should we stop?
  Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

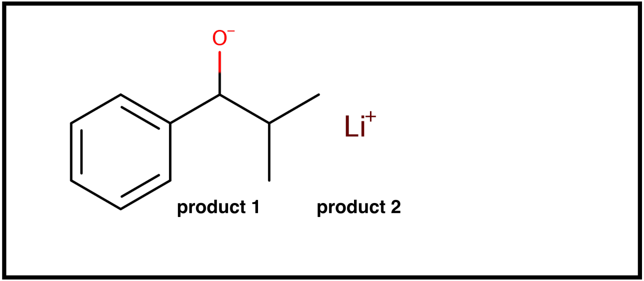
```
  Select **starting** atom A.
  Repeat:
     Select atom B bonded to A and **remove** 2 electrons from bond A-B.
     Should we **stop**?
     Select new atom A and **add** bond B-A.
     Should we **stop**?
→ Compute resulting electron path.
  Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

ELECTRO: an algorithm for electron paths

```
  Select starting atom A.
  Repeat:
     Select atom B bonded to A and remove 2 electrons from bond A-B.
     Should we stop?
     Select new atom A and add bond B-A.
     Should we stop?
  Compute resulting electron path.
→ Compute resulting products.
```



Bradshaw, Kusner, Paige, Segler and Hernández-Lobato, Arxiv, 2018.

A probabilistic model

Electron path $\mathbf{a} = (a_1, a_2, \ldots, a_T)$ defined by **sequence of selected atoms**.

Electrons **removed and added alternatively** from pairs of consecutive atoms.

A probabilistic model

Electron path $\mathbf{a} = (a_1, a_2, \ldots, a_T)$ defined by **sequence of selected atoms**.

Electrons **removed and added alternatively** from pairs of consecutive atoms.

We assume $\mathbf{a}$ is sampled from

$$p_{\boldsymbol{\theta}}(\mathbf{a}|\mathcal{M}_1) = \left[\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(a_t|a_{t-1}, \mathcal{M}_t) p_{\boldsymbol{\theta}}(\mathrm{CONT}|\mathcal{M}_t)\right] p_{\boldsymbol{\theta}}(\mathrm{STOP}|\mathcal{M}_{T+1}),$$

where

- $\boldsymbol{\theta}$ are the model parameters.
- $\mathcal{M}_t$ represents the current state of molecules.
- $p_{\boldsymbol{\theta}}(\mathrm{STOP}|\mathcal{M}_t) = 1 - p_{\boldsymbol{\theta}}(\mathrm{CONT}|\mathcal{M}_t)$ is the probability of stopping.

A probabilistic model

Electron path $\mathbf{a} = (a_1, a_2, \ldots, a_T)$ defined by **sequence of selected atoms**.

Electrons **removed and added alternatively** from pairs of consecutive atoms.

We assume $\mathbf{a}$ is sampled from

$$p_{\boldsymbol{\theta}}(\mathbf{a}|\mathcal{M}_1) = \left[\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(a_t|a_{t-1},\mathcal{M}_t)p_{\boldsymbol{\theta}}(\text{CONT}|\mathcal{M}_t)\right] p_{\boldsymbol{\theta}}(\text{STOP}|\mathcal{M}_{T+1})\,,$$

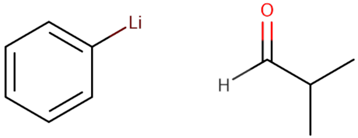
where

- $\boldsymbol{\theta}$ are the model parameters.
- $\mathcal{M}_t$ represents the current state of molecules.
- $p_{\boldsymbol{\theta}}(\text{STOP}|\mathcal{M}_t) = 1 - p_{\boldsymbol{\theta}}(\text{CONT}|\mathcal{M}_t)$ is the probability of stopping.

How to specify $p_{\boldsymbol{\theta}}(a_t|a_{t-1},\mathcal{M}_t)$ and $p_{\boldsymbol{\theta}}(\textbf{CONT}|\mathcal{M}_t)$?

Forming node and graph embeddings

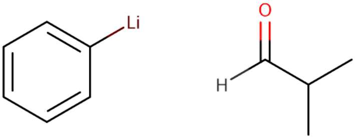
```
→ Initialize atom features $\mathbf{h}_{0,i}$.
  For $t = 1$ to $T$:
      Compute atom messages $\mathbf{m}_{t,i} = \sum_{j \sim i} \mathbf{A}_{\text{type}(j \sim i)} \mathbf{h}_{t-1,j}$.
      Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
  Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```



Forming node and graph embeddings

```
→ Initialize atom features $\mathbf{h}_{0,i}$.
  For $t = 1$ to $T$:
      Compute atom messages $\mathbf{m}_{t,i} = \sum_{j\sim i} \mathbf{A}_{\text{type}(j\sim i)}\mathbf{h}_{t-1,j}$.
      Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
  Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```

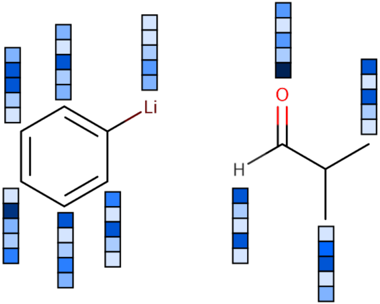
Feature
Atom type
Degree
Explicit Valence
Hybridization
H count
Electronegativity
Atomic number
Part of an aromatic ring



Forming node and graph embeddings

```
→ Initialize atom features $\mathbf{h}_{0,i}$.
  For $t = 1$ to $T$:
      Compute atom messages $\mathbf{m}_{t,i} = \sum_{j \sim i} \mathbf{A}_{\text{type}(j \sim i)} \mathbf{h}_{t-1,j}$.
      Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
  Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```

Feature
Atom type
Degree
Explicit Valence
Hybridization
H count
Electronegativity
Atomic number
Part of an aromatic ring



Forming node and graph embeddings

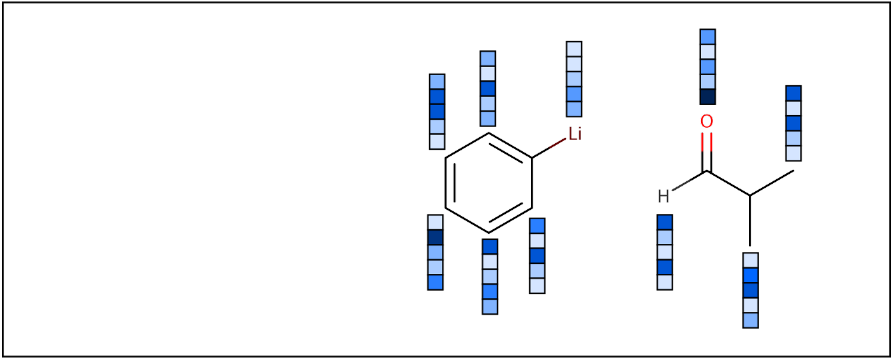
Initialize atom features $\mathbf{h}_{0,i}$.

For $t = 1$ to T:

→ Compute atom messages $\mathbf{m}_{t,i} = \sum_{j \sim i} \mathbf{A}_{\text{type}(j \sim i)} \mathbf{h}_{t-1,j}$.

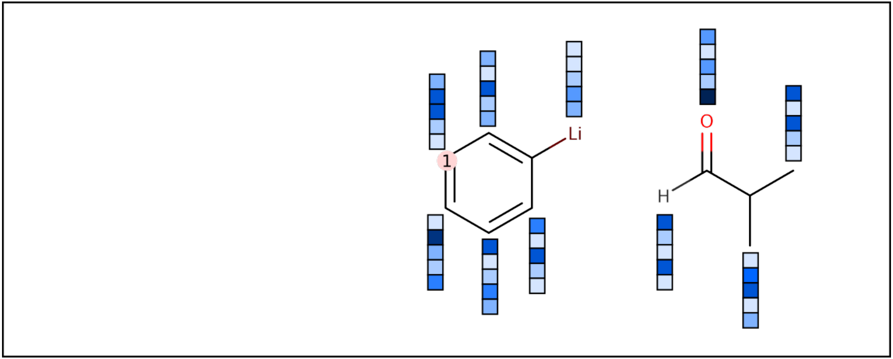
Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.

Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.



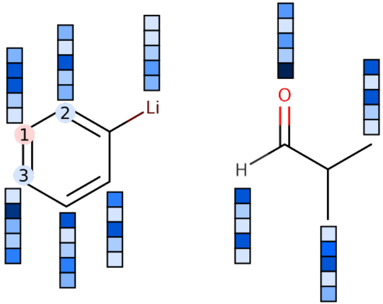
Forming node and graph embeddings

```
Initialize atom features $\mathbf{h}_{0,i}$.
For $t = 1$ to $T$:
→   Compute atom messages $\mathbf{m}_{t,i} = \sum_{j\sim i} \mathbf{A}_{\text{type}(j\sim i)} \mathbf{h}_{t-1,j}$.
    Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```



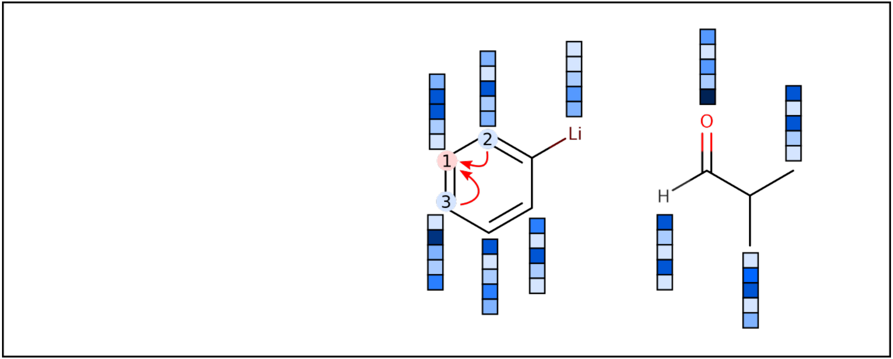
Forming node and graph embeddings

```
Initialize atom features $\mathbf{h}_{0,i}$.
For $t = 1$ to $T$:
→   Compute atom messages $\mathbf{m}_{t,i} = \sum_{j\sim i} \mathbf{A}_{\text{type}(j\sim i)}\mathbf{h}_{t-1,j}$.
    Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```



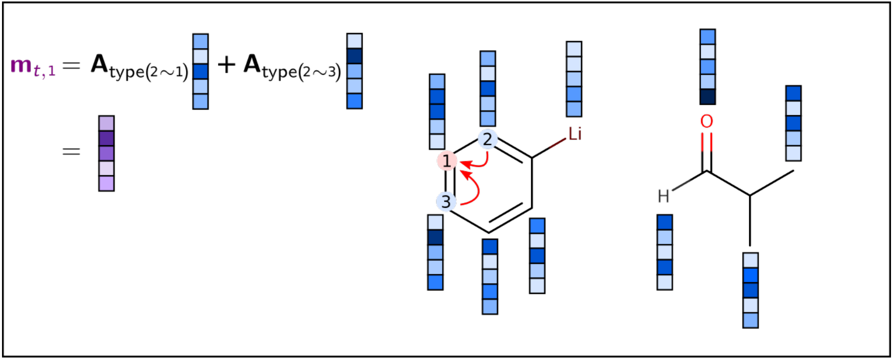
Forming node and graph embeddings

```
   Initialize atom features $\mathbf{h}_{0,i}$.
   For $t = 1$ to $T$:
→      Compute atom messages $\mathbf{m}_{t,i} = \sum_{j\sim i} \mathbf{A}_{\text{type}(j\sim i)} \mathbf{h}_{t-1,j}$.
       Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
   Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```



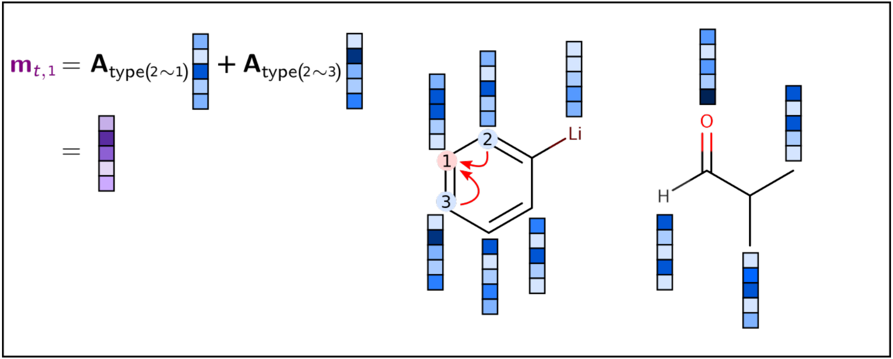
Forming node and graph embeddings

```
Initialize atom features $\mathbf{h}_{0,i}$.
For $t = 1$ to $T$:
→   Compute atom messages $\mathbf{m}_{t,i} = \sum_{j \sim i} \mathbf{A}_{\text{type}(j \sim i)} \mathbf{h}_{t-1,j}$.
    Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```


$\mathbf{m}_{t,1} = \mathbf{A}_{\text{type}(2\sim1)}$
$+ \mathbf{A}_{\text{type}(2\sim3)}$
$=$
1
2
3
Li
O
H

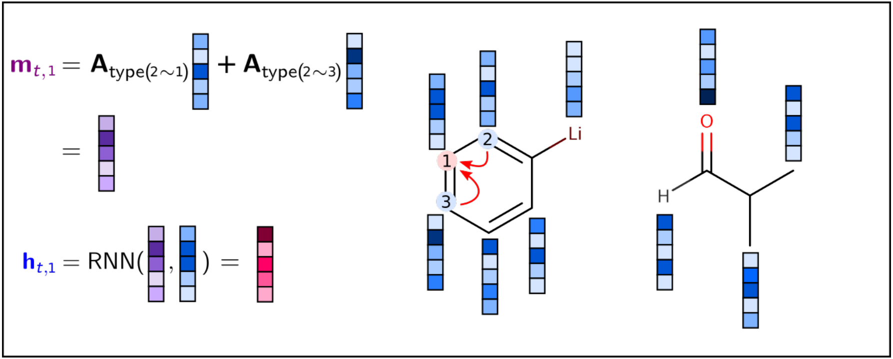
Forming node and graph embeddings

```
  Initialize atom features $\mathbf{h}_{0,i}$.
  For $t = 1$ to $T$:
      Compute atom messages $\mathbf{m}_{t,i} = \sum_{j \sim i} \mathbf{A}_{\text{type}(j \sim i)} \mathbf{h}_{t-1,j}$.
→     Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
  Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```


$\mathbf{m}_{t,1} = \mathbf{A}_{\text{type}(2\sim1)}$ $+$ $\mathbf{A}_{\text{type}(2\sim3)}$
$=$
1
2
3
Li
O
H

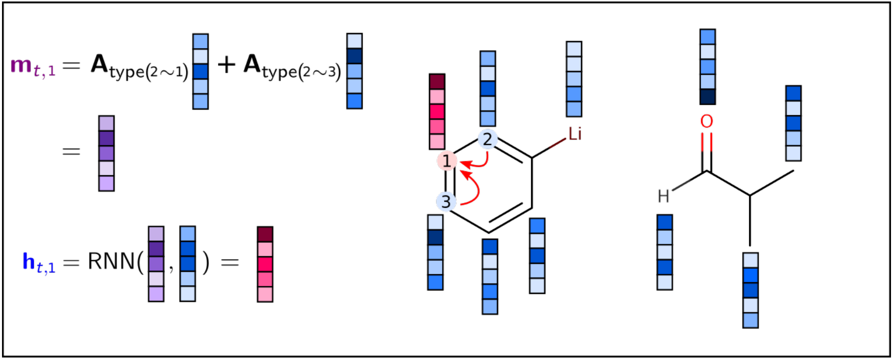
Forming node and graph embeddings

```
  Initialize atom features $\mathbf{h}_{0,i}$.
  For $t = 1$ to $T$:
      Compute atom messages $\mathbf{m}_{t,i} = \sum_{j \sim i} \mathbf{A}_{\text{type}(j \sim i)} \mathbf{h}_{t-1,j}$.
→     Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
  Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```


$\mathbf{m}_{t,1} = \mathbf{A}_{\text{type}(2\sim1)}$
$+ \mathbf{A}_{\text{type}(2\sim3)}$
$=$
$\mathbf{h}_{t,1} = \text{RNN}($
$,$
$) =$
1
2
3
Li
O
H

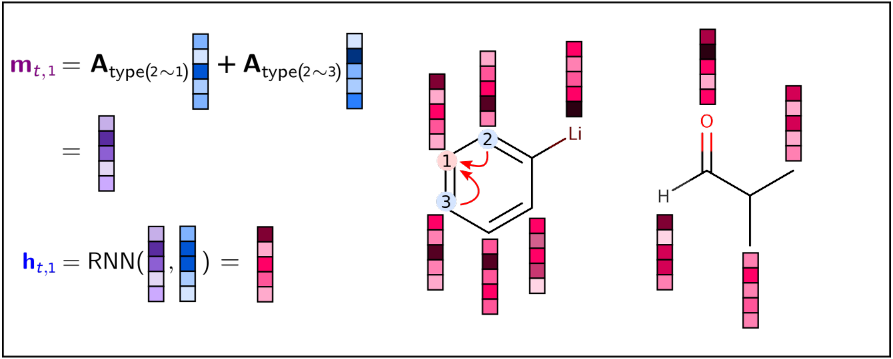
Forming node and graph embeddings

```
   Initialize atom features $\mathbf{h}_{0,i}$.
   For $t = 1$ to $T$:
      Compute atom messages $\mathbf{m}_{t,i} = \sum_{j\sim i} \mathbf{A}_{\text{type}(j\sim i)} \mathbf{h}_{t-1,j}$.
→     Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
   Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```


$\mathbf{m}_{t,1} = \mathbf{A}_{\text{type}(2\sim1)}$
$+ \mathbf{A}_{\text{type}(2\sim3)}$
$=$
$\mathbf{h}_{t,1} = \text{RNN}($
$,$
$) =$
1
2
3
Li
O
H

Forming node and graph embeddings

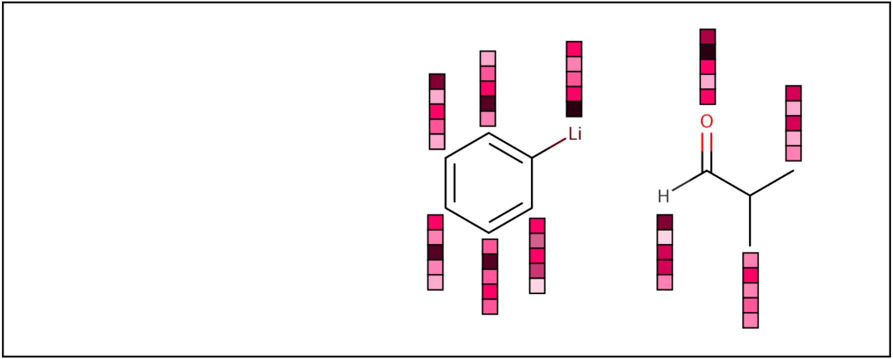
```
Initialize atom features $\mathbf{h}_{0,i}$.
For $t = 1$ to $T$:
    Compute atom messages $\mathbf{m}_{t,i} = \sum_{j\sim i} \mathbf{A}_{\text{type}(j\sim i)} \mathbf{h}_{t-1,j}$.
→   Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```


$\mathbf{m}_{t,1} = \mathbf{A}_{\text{type}(2\sim1)}$ + $\mathbf{A}_{\text{type}(2\sim3)}$
=
$\mathbf{h}_{t,1} = \text{RNN}($, $) =$
1
2
3
Li
O
H

Forming node and graph embeddings

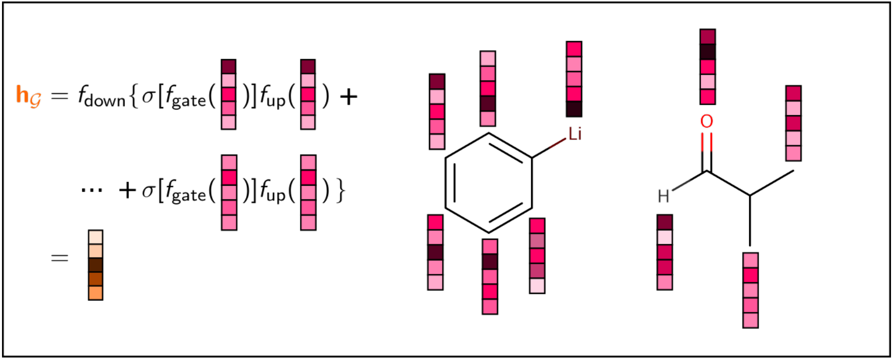
```
Initialize atom features $\mathbf{h}_{0,i}$.
For $t = 1$ to $T$:
    Compute atom messages $\mathbf{m}_{t,i} = \sum_{j \sim i} \mathbf{A}_{\text{type}(j \sim i)} \mathbf{h}_{t-1,j}$.
    Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
```

→ Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})] f_{\text{up}}(\mathbf{h}_{T,i})\}$.



Forming node and graph embeddings

```
Initialize atom features $\mathbf{h}_{0,i}$.
For $t = 1$ to $T$:
    Compute atom messages $\mathbf{m}_{t,i} = \sum_{j\sim i} \mathbf{A}_{\text{type}(j\sim i)}\mathbf{h}_{t-1,j}$.
    Update atom features $\mathbf{h}_{t,i} = \text{RNN}(\mathbf{m}_{t,i}, \mathbf{h}_{t-1,i})$.
→ Compute graph embedding $\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sum_i \sigma[f_{\text{gate}}(\mathbf{h}_{T,i})]f_{\text{up}}(\mathbf{h}_{T,i})\}$.
```



$\mathbf{h}_{\mathcal{G}} = f_{\text{down}}\{\sigma[f_{\text{gate}}(\;)]f_{\text{up}}(\;) +$
$\cdots + \sigma[f_{\text{gate}}(\;)]f_{\text{up}}(\;)\}$
$=$
Li
O
H

Stopping probability and next atom in path

The stopping probability is defined as $p_{\boldsymbol{\theta}}(\mathrm{STOP}|\mathcal{M}_t) = \sigma(f_{\text{stop}}(\mathbf{h}_{\mathcal{G}}))$.

Stopping probability and next atom in path

The stopping probability is defined as $p_\theta(\text{STOP}|\mathcal{M}_t) = \sigma(f_{\text{stop}}(\mathbf{h}_\mathcal{G}))$.

For the next atom, we use a **softmax layer** with inputs $s_i = \text{MLP}(\mathbf{h}_{T,i}, \text{CONTEXT})$ and corresponding binary masks $m_i \in \{0, 1\}$ to remove invalid choices:

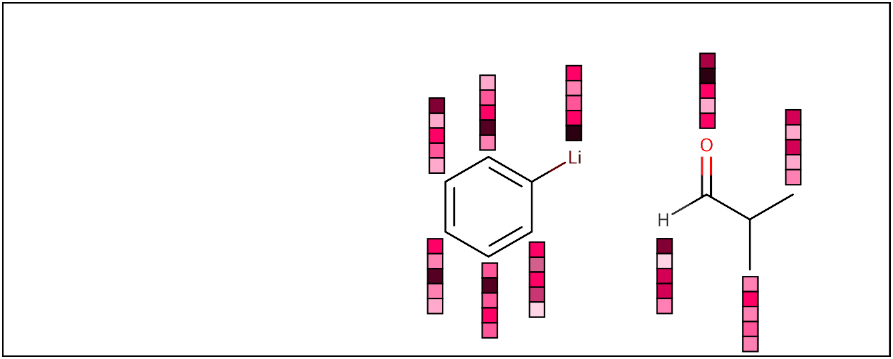
$$p_\theta(a_t = i|a_{t-1}, \mathcal{M}_t) = \frac{m_i \exp(s_i)}{\sum_j m_j \exp(s_j)} \,.$$

Stopping probability and next atom in path

The stopping probability is defined as $p_{\theta}(\text{STOP}|\mathcal{M}_t) = \sigma(f_{\text{stop}}(\mathbf{h}_{\mathcal{G}}))$.

For the next atom, we use a **softmax layer** with inputs $s_i = \text{MLP}(\mathbf{h}_{T,i}, \text{CONTEXT})$ and corresponding binary masks $m_i \in \{0,1\}$ to remove invalid choices:

$$p_{\theta}(a_t = i | a_{t-1}, \mathcal{M}_t) = \frac{m_i \exp(s_i)}{\sum_j m_j \exp(s_j)} .$$

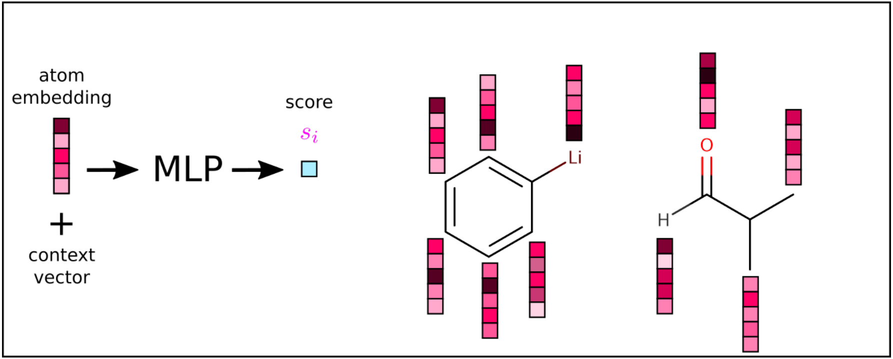


Stopping probability and next atom in path

The stopping probability is defined as $p_\theta(\mathrm{STOP}|\mathcal{M}_t) = \sigma(f_{\mathrm{stop}}(\mathbf{h}_\mathcal{G}))$.

For the next atom, we use a **softmax layer** with inputs $s_i = \mathrm{MLP}(\mathbf{h}_{T,i}, \mathrm{CONTEXT})$ and corresponding binary masks $m_i \in \{0,1\}$ to remove invalid choices:

$$p_\theta(a_t = i | a_{t-1}, \mathcal{M}_t) = \frac{m_i \exp(s_i)}{\sum_j m_j \exp(s_j)}\,.$$

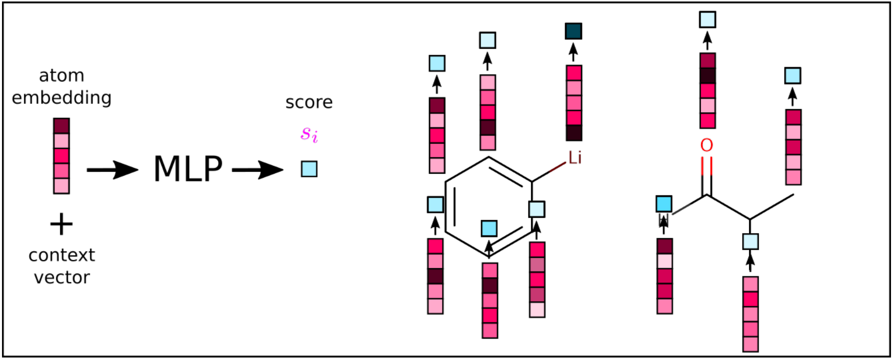


Stopping probability and next atom in path

The stopping probability is defined as $p_\theta(\text{STOP}|\mathcal{M}_t) = \sigma(f_{\text{stop}}(\mathbf{h}_\mathcal{G}))$.

For the next atom, we use a **softmax layer** with inputs $s_i = \text{MLP}(\mathbf{h}_{T,i}, \text{CONTEXT})$ and corresponding binary masks $m_i \in \{0,1\}$ to remove invalid choices:

$$p_\theta(a_t = i|a_{t-1}, \mathcal{M}_t) = \frac{m_i \exp(s_i)}{\sum_j m_j \exp(s_j)}\,.$$



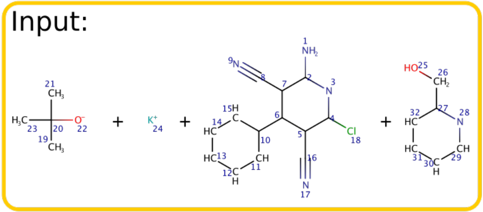
Training data

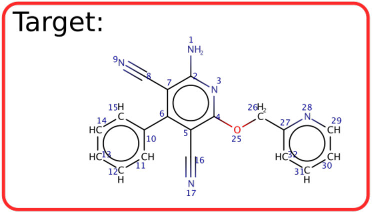
USPTO: 480K chemical reactions extracted from the US patent database.

Atoms have unique ID to easy match them before and after the reaction.

Approximate electron paths obtained for **73%** of USPTO by using simple rules.

```
[CH3:19][C:20]([CH3:21])([O-:22])[CH3:23].[K+:24].[NH2:1][c:2]1[n:3]
[c:4]([Cl:18])[c:5]([C:16]#[N:17])[c:6](-[c:10]2[cH:11][cH:12][cH:13]
[cH:14][cH:15]2)[c:7]1[C:8]#[N:9].[OH:25][CH2:26][c:27]1[n:28][cH:29]
[cH:30][cH:31][cH:32]1>>[NH2:1][c:2]1[n:3][c:4]([O:25][CH2:26][c:
27]2[n:28][cH:29][cH:30][cH:31][cH:32]2)[c:5]([C:16]#[N:17])[c:6](-[c:
10]2[cH:11][cH:12][cH:13][cH:14][cH:15]2)[c:7]1[C:8]#[N:9]
```





Results

Beam search to find top-K predictions for the electron path.

Multiple electron paths can result in same product (harder prediction task).

LITE variant ignores reagent information.

Electron path prediction

	Accuracies (%)			
Model Name	Top-1	Top-2	Top-3	Top-5
Electro-Lite	70.3	82.8	87.7	92.2
Electro	77.8	89.2	92.4	94.7

Product prediction

	Accuracies (%)			
Model Name	Top-1	Top-2	Top-3	Top-5
Electro-Lite	78.2	87.7	91.5	94.4
Electro	**87.0**	**92.6**	**94.5**	**95.9**
WLDN	84.0	89.2	91.1	92.3
Seq2Seq	80.3*	84.7*	86.2*	87.5*

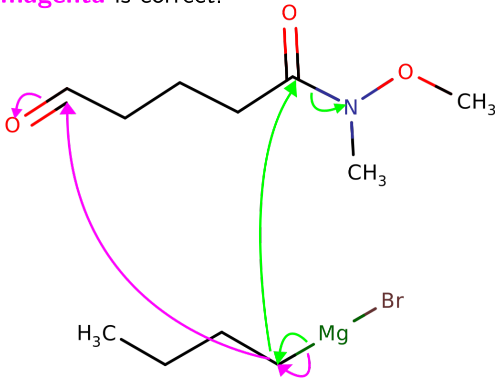
Bonus: **interpretable explanation** for product prediction task.

Qualitative analysis

Easy to identify what went wrong!

First choice in **green** is incorrect, but “chemically reasonable”.

Second choice in **magenta** is correct.



Take home messages

ELECTRO...

1. predicts electron paths with **high accuracy**.
2. produces an output that is very easy to **interpret** by chemists.
3. exploits that only a **small number of atoms** interact during the reaction.
4. also predicts **final products** with high accuracy.

Collaborators

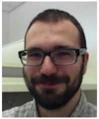
R. P. Adams

J. Aguilera-Ipaguirre

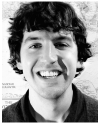
A. Aspuru-Guzik

John Bradshaw

D. Duvenaud


R. Gomez-Bonbardelli

T. Hirzel


M. Kusner

B. Paige

B. Sánchez-Lengeling

Marwin H. S. Segler

D. Sheberla

J. Wey

Thanks!

Take home messages

Generative models of molecues...

1. **Data-driven approach** to molecule generation, no expertise needed.
2. Sampled molecules are **realistic**, unlike those generated with rules.
3. Create a **continuous latent space** which is useful for optimization.

Grammar variational autoencoder...

1. Produces a **larger fraction of valid molecules** when decoding.
2. Produces **better predictions of molecule properties** from latent space.
3. **Molecule optimization** results are **improved**.

ELECTRO...

1. predicts electron paths with **high accuracy**.
2. produces an output that is very easy to **interpret** by chemists.
3. exploits that only a **small number of atoms** interact during the reaction.
4. also predicts **final products** with high accuracy.