Predicting and optimizing the properties of small molecules

Utilizing in silico models in both directions

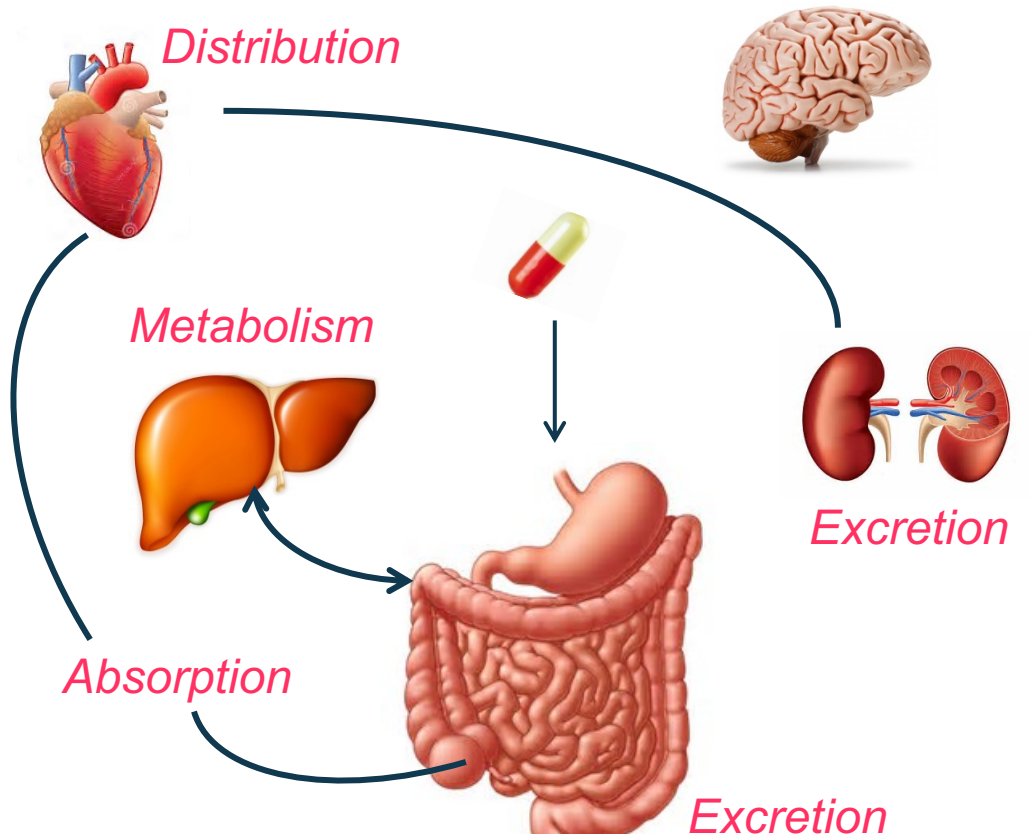
Floriane Montanari
&
Robin Winter

RDKit User Group Meeting, Hamburg

25.09.2019

Multitask learning for ADMET prediction

Absorption – Distribution – Metabolism – Excretion – Toxicity



Where will it go?

How much will get there?

How long will it stay?

Will it be transformed?

How will it be removed?

Will it reach unwanted sites?

How can we improve the predictions of such properties?

Absorption – Distribution: physico-chemical properties

Solubility

- *Nephelometry*:	88 000 measured cpds
- PBS pH 6.5 from DMSO	38 800 measured cpds
- PBS pH 6.5 from Powder	2 300 measured cpds
- *PBS pH 6.5 from DMSO* *not fully dissolved*	7 300 measured cpds
- *PBS pH 6.5 unknown starting point*	50 000 measured cpds

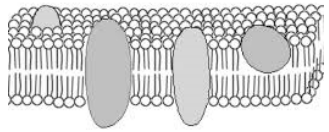
LogD

- pH 7.5	88 000 measured cpds
- pH 2.3	236 000 measured cpds

Melting point

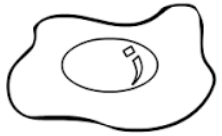
92 000 compounds

Membrane affinity



66 800 compounds

Serum albumin binding

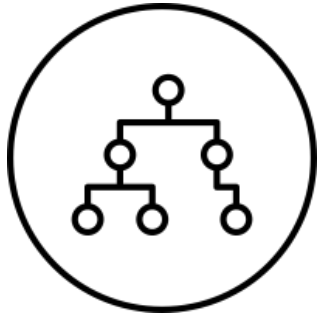


64 000 compounds

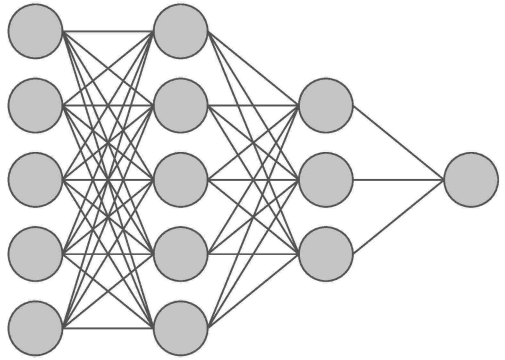
Montanari *et al.*, ChemRxiv, 2019

Different methods compared

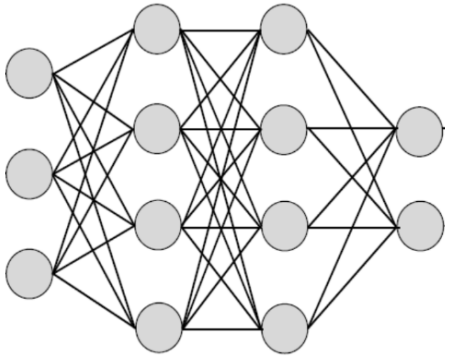
Baseline



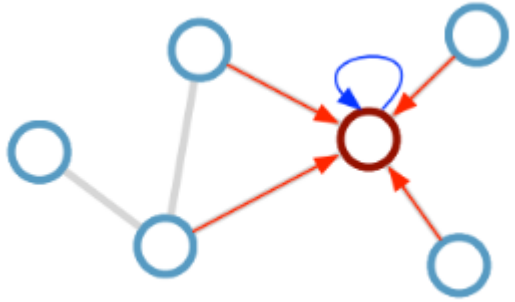
Single task neural networks



Multitask neural networks

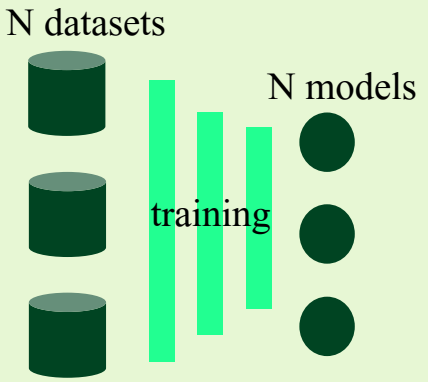


Graph convolutional neural networks



Multitask learning

Multitask



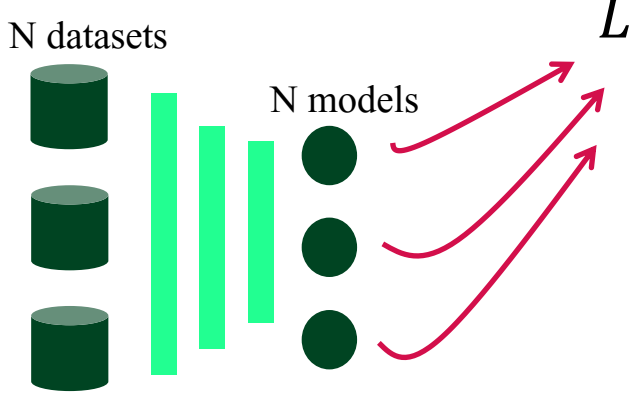
Multiple related learning tasks are learned simultaneously using a shared representation. Ex: text translation in multiple languages.

- Choose N datasets that can be learned together:
LogD (X_1, y_1)
Solubility (X_2, y_2)
Membrane affinity (X_3, y_3)

- Combine them into a multitask training set (X_4, Y)
X_4 contains *U* unique compounds from X_1, X_2 and X_3
Y is of shape (U, N) with missing values when a given *u* doesn‘t have a measurement

There is no requirement of overlap between the different datasets. Some overlap helps!

Multitask learning in practice



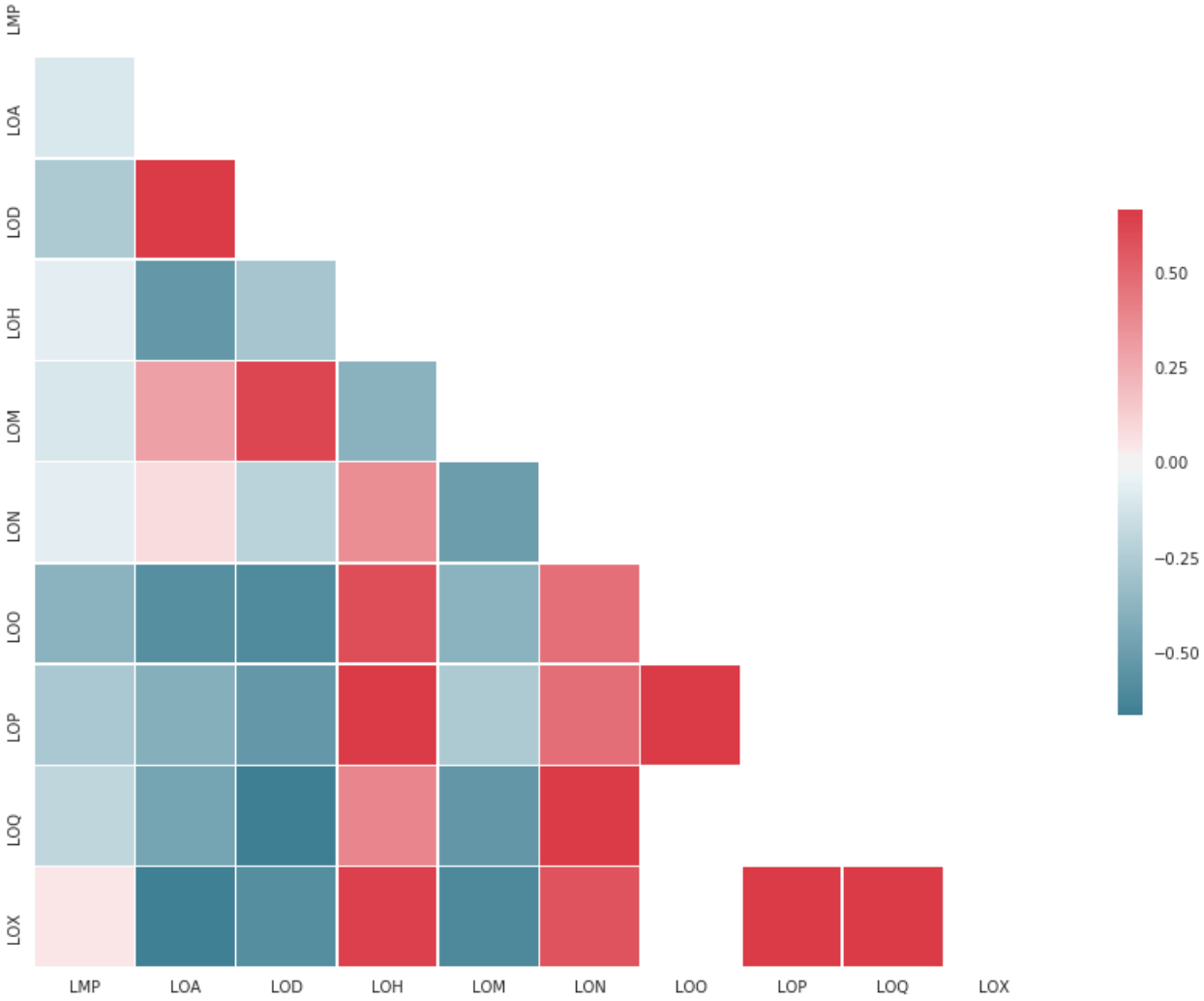
Loss calculation:

$$L_t = \frac{1}{|B|}\sum_{i \in B}(yi - \hat{y}_i)^2 \Rightarrow L_{global} = \frac{\sum_t w_t L_t}{\sum w_t}$$

Within a minibatch *B*
Independent for each task *t*
Ignore missing values

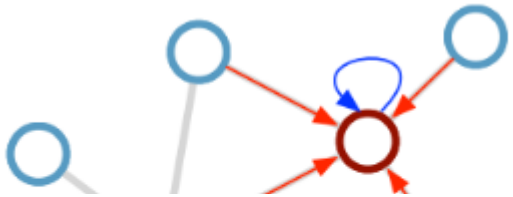
Consequence: the N tasks must be giving outputs in the same range! Necessary to scale all y_t (z-scaling works well in practice)

Motivation for a multitask approach and expected benefits



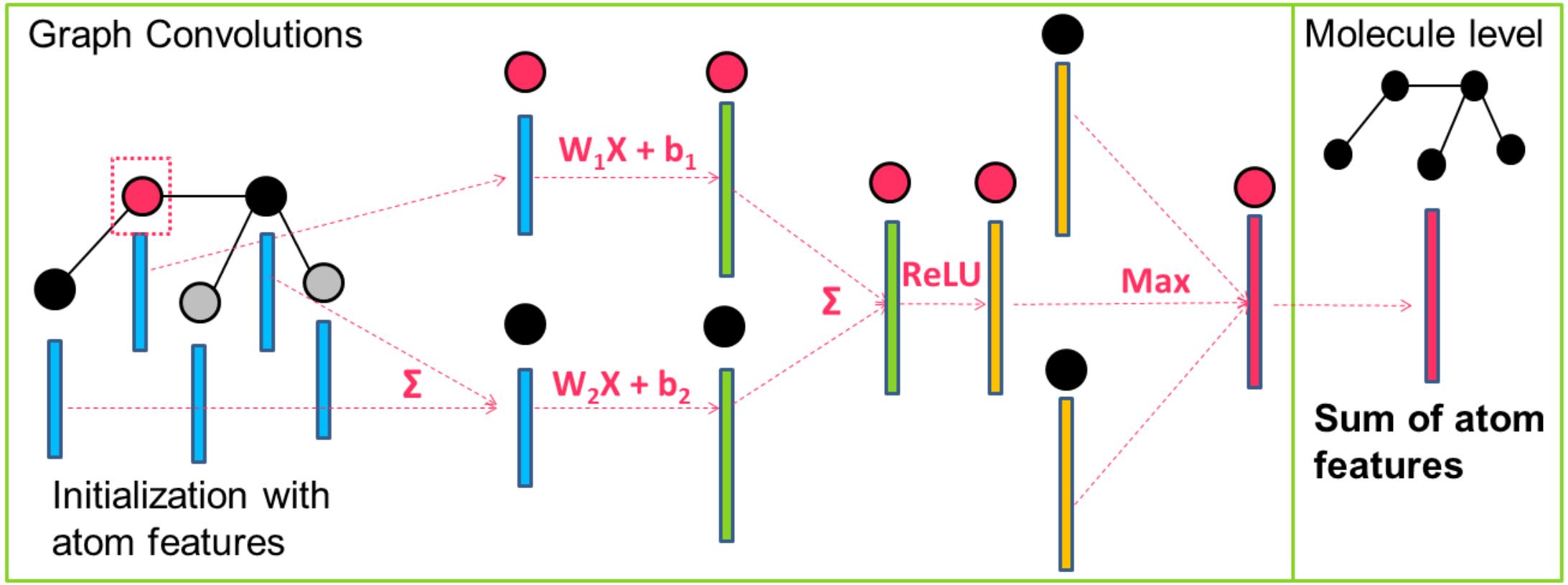
- **Larger training set**: endpoints with less compounds benefit from the chemical space of endpoints with more compounds.
- **Exploits correlations between endpoints**
- **Regularization** method

Graph convolutional networks for chemical data



Concept: represent the molecules as graphs (nodes = atoms, edges = bonds)

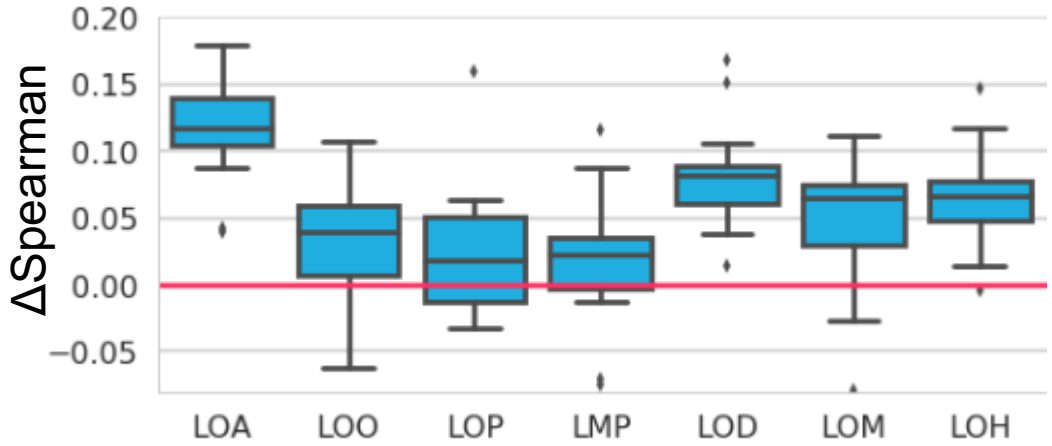
Learn node (atom) representations that will help with the task at hand



Results

Random Forest versus single task neural networks

Average over 4 leave-cluster-out CV experiments, networks hyperparamters were only optimized on task *LMP, Spearman*



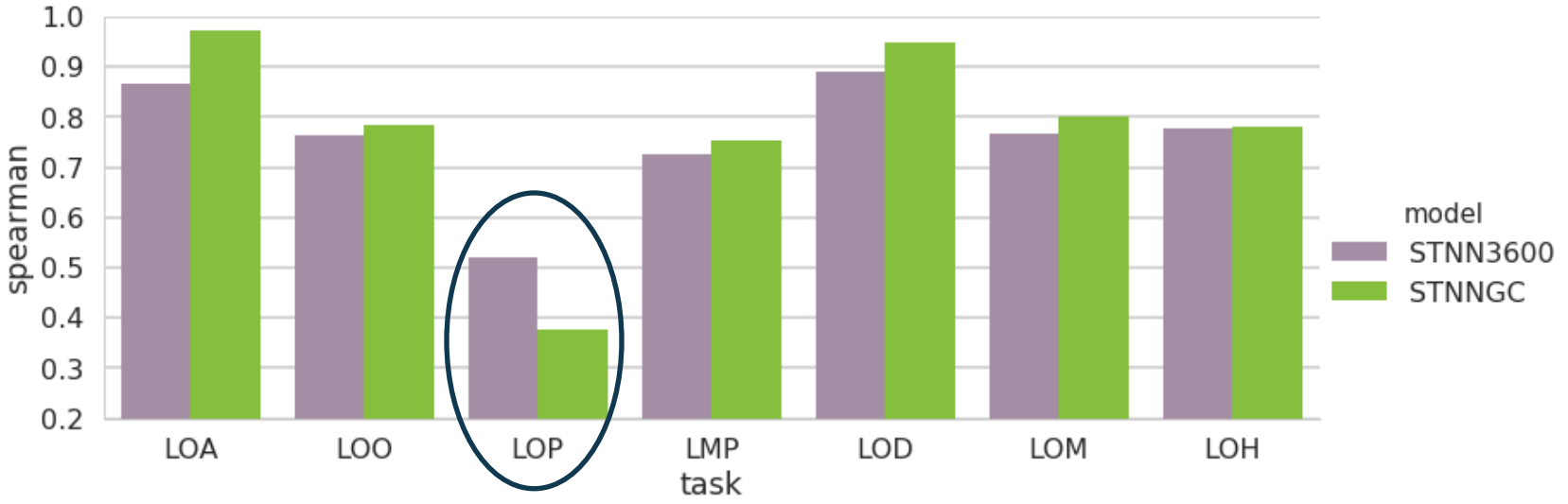
Neural networks (fully connected, same features as RF) outperform RF for the physchem properties

LogD acid / Solubility DMSO / Solubility powder / melting point / LogD / Membrane affinity / HSA binding

Montanari *et al.*, ChemRxiv, 2019

Fully connected networks versus graph convolutional networks

Single task



Graph convolution brings better performance on average, especially true for the larger tasks. LOP is very small (≈2000 cpds) so probably graph conv is overfitting.

Average over 2 leave-cluster-out CV experiments

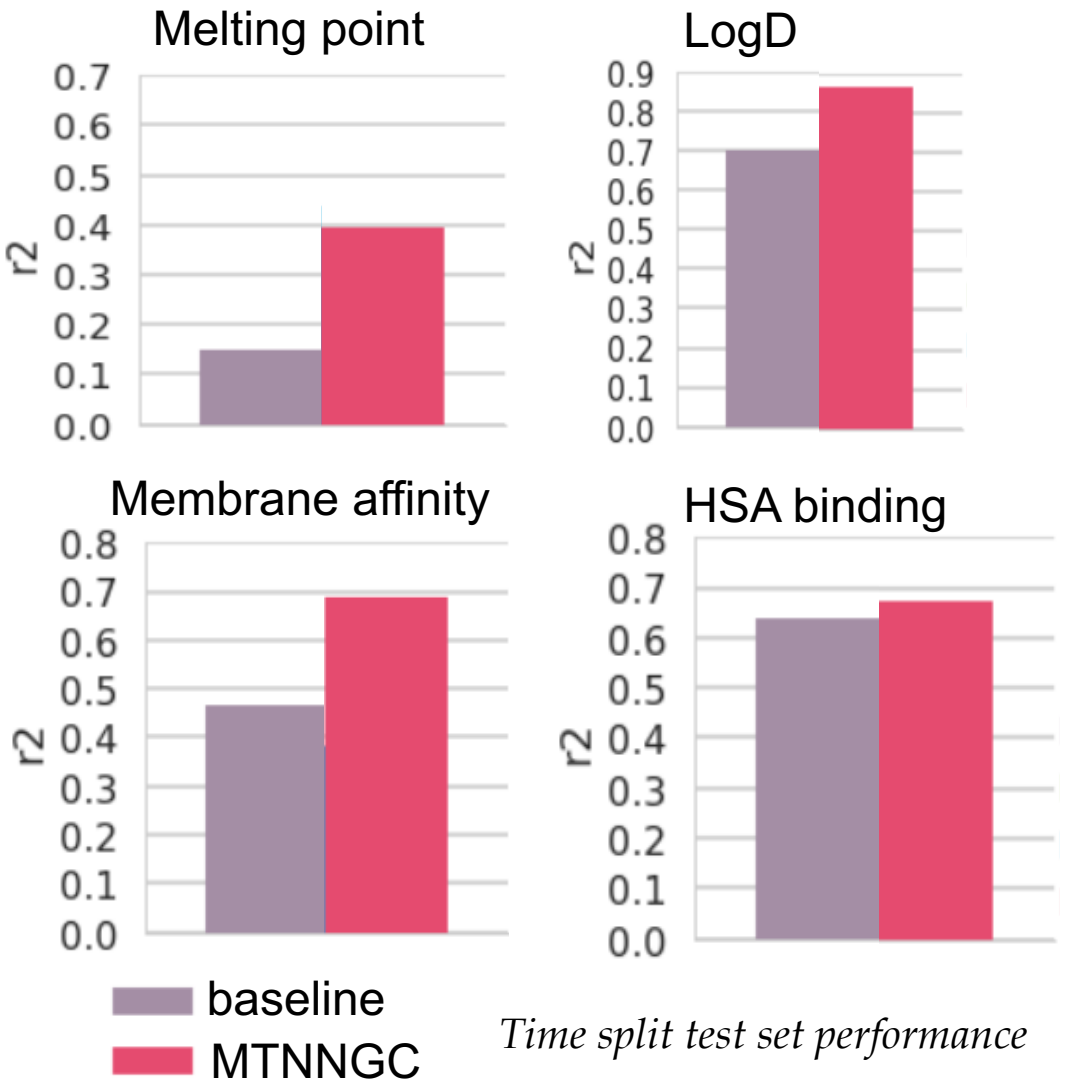
Absorption – Distribution: physico-chemical properties

Best model: multitask graph convolutional network

Average over 2 leave-cluster-out CV experiments

	R^2	Spearman	RMSE
LogD pH 7.5	0.88	0.94	0.34
LogD pH 2.3	0.91	0.96	0.36
Membrane affinity	0.71	0.84	0.51
hSA binding	0.63	0.82	0.50
Melting point	0.53	0.74	39
Solubility DMSO	0.58	0.77	0.83
Solubility Powder	0.55	0.75	0.79

Excellent performance for all modeled endpoints and significant improvement over models previously in production.



Time split test set performance

Montanari *et al.*, ChemRxiv, 2019

Wrap-up

With the current amount of data for physico-chemical properties, Deep Learning boosts performance with respect to classical ML models.

Graph convolutional networks are very poweful for those assays once the training set size is large enough.

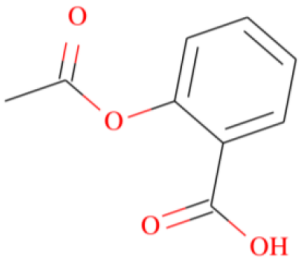
Multitask learning improves the performance on all but the largest task, and adding more related tasks also can help.

Current model is in production and used by the medicinal chemists at Bayer.

Other endpoints have been modeled, graph convolutional networks are not always the best! and it is hard to know how to group tasks, but overall one can get better performance with DL compared to Random Forest.

Molecule swarm optimization

Motivation



Featurization

Descriptor

Classifier

Score: 0.65

Optimize

???

Decode

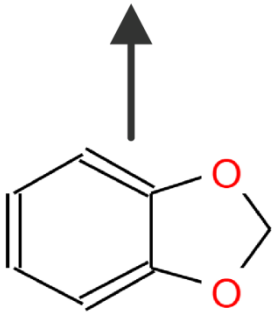
Descriptor *

Classifier

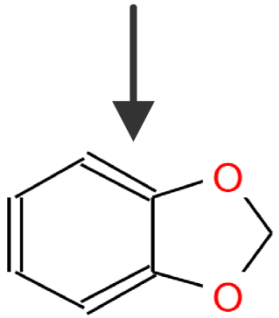
Score: 0.98

Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations*

InChI=1S/C7H6O2/
c1-2-4-7-6(3-1)8-5-9
-7/h1-4H,5H2

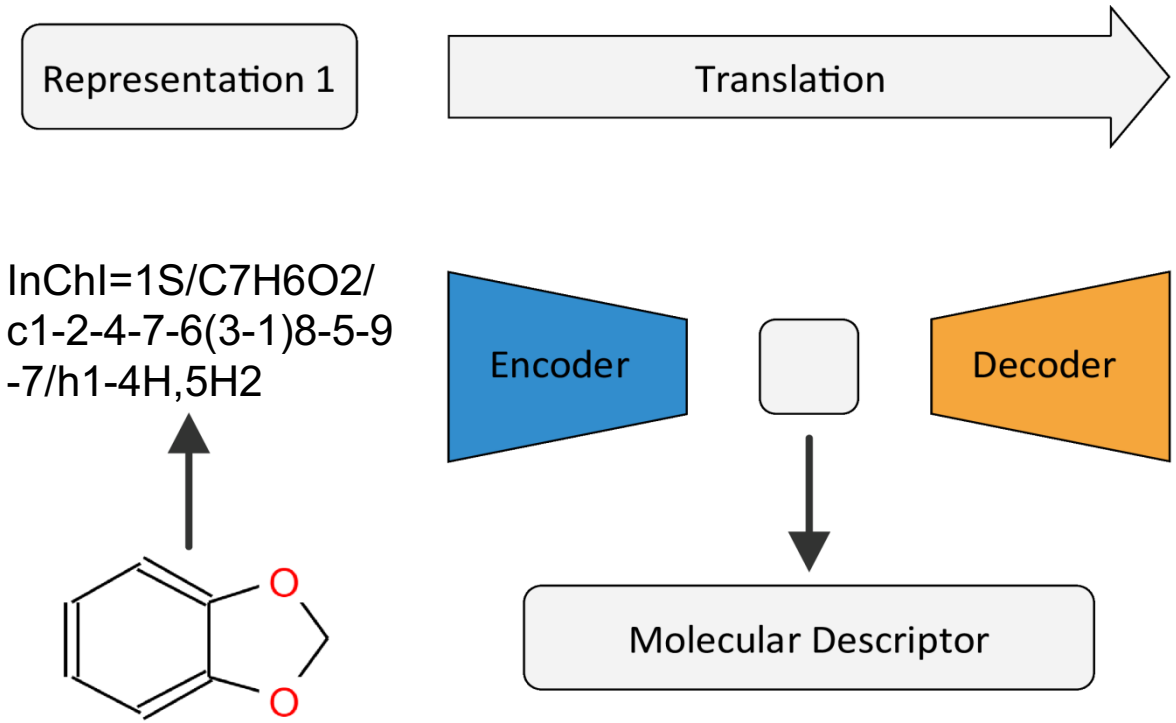


c2ccc1OCOc1c2



* Winter *et al. Chemical Science* (2019)
Source code: https://github.com/jrwnter/cddd

Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations*



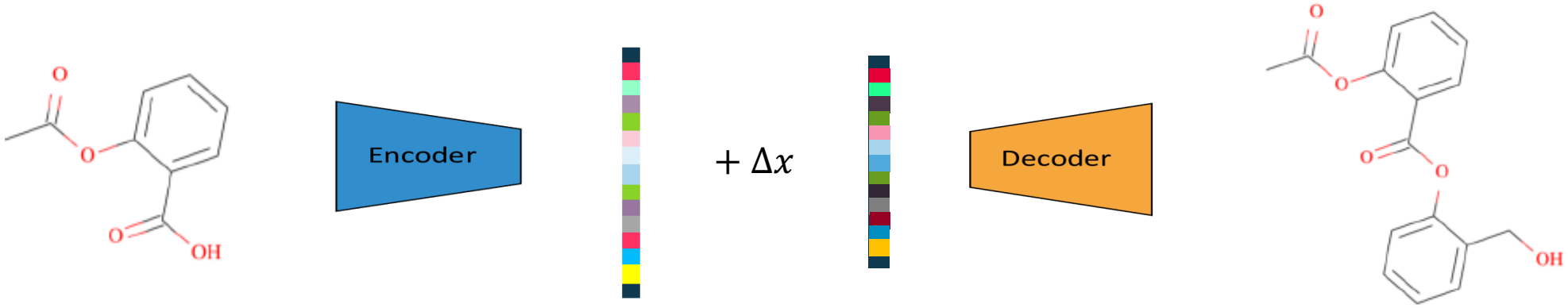
- Train on ~70 million compounds
- Model learns to extract the essence both representations have in common

* Winter *et al. Chemical Science* (2019)
Source code: https://github.com/jrwnter/cddd

Learning **C**ontinuous and **D**ata-**D**riven molecular **D**escriptors (CDDD) by translating equivalent chemical representations

Resulting molecular descriptors show competitive performance to state-of-the-art descriptors

and are continuous and reversible (can be decoded back to interpretable chemical descriptions)



Particle Swarm Optimization (PSO)

Particles x_i are initialized with random velocities v_i

Position update of particle *i* at step *k:*

$$x_i^{k+1} = x_i^k + v_i^k$$

Velocity update of particle *i :*

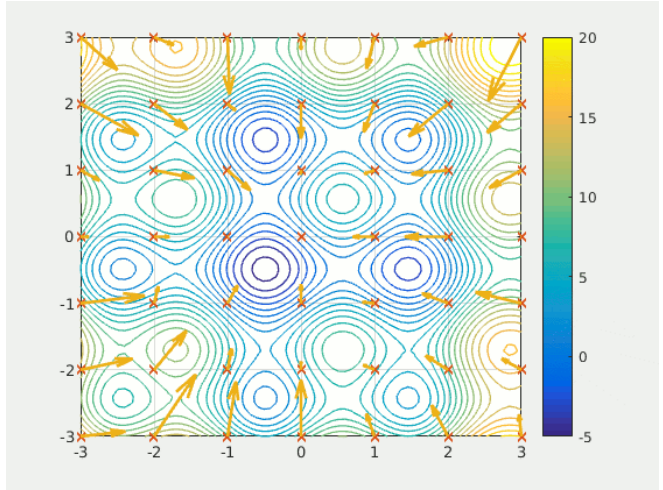
$$v_i^{k+1} = wv_i^k + c_1 r_1 \left(x_i^{\text{best}} - x_i^k\right) + c_2 r_2 \left(x^{\text{best}} - x_i^k\right)$$

$$x_i^{\text{best}} = \text{argmax} f(x_i^k)$$

$$x^{\text{best}} = \text{argmax} f(x_i^{\text{best}})$$

f : objective function

$r_i \sim U(0,1)$



Molecular Swarm Optimization*

Multi-objective optimization in the CDDD space

- Objective:
  - Multi-target activity profile (protein)
    - Epidermal Growth Factor (EGFR)
    - β-site APP cleaving enzyme-1 (BACE1)
  - Pharmacokinetic features
    - Solubility
    - Metabolic stability
    - Permeability

  Trained models

  - Structure filter, QED, SA etc. — hard-coded
- Experiments:
  1. maximize BACE1 minimize EGFR
  2. maximize EGFR minimize BACE1
  3. maximize EGFR maximize BACE1

  selectivity

  + maximize ADME and Structure features

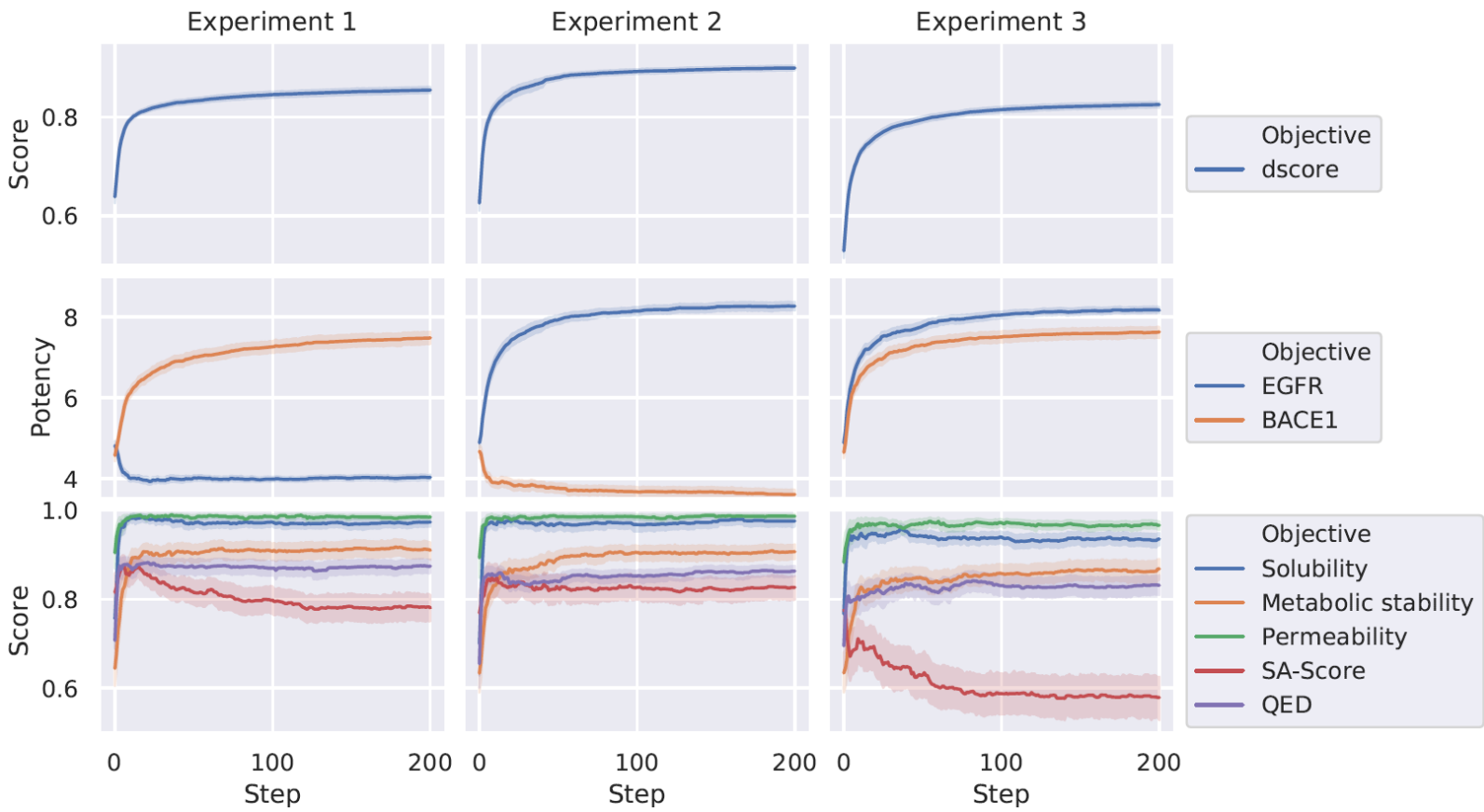
* Winter *et al. Chemical Science* (2019)
Source code: https://github.com/jrwnter/mso

Molecular Swarm Optimization*

Multi-objective optimization in the CDDD space

- Objective:
  - Multi-target activity profile (protein)
    - Epidermal Growth Factor (EGFR)
    - β-site APP cleaving enzyme-1 (BACE1)
  - Pharmacokinetic features
    - Solubility
    - Metabolic stability
    - Permeability
  - Structure filter, QED, SA etc.
- Experiments:
  1. maximize BACE1 minimize EGFR
  2. maximize EGFR minimize BACE1
  3. maximize EGFR maximize BACE1

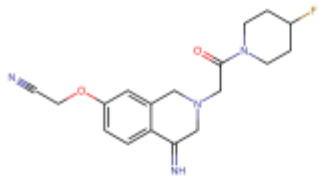
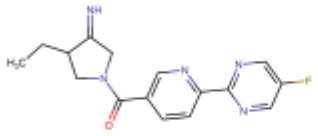
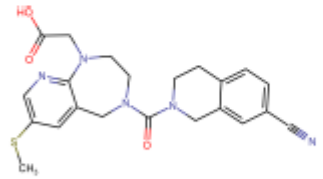
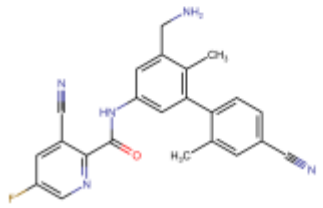
  + maximize ADME and Structure features



* Winter *et al. Chemical Science* (2019)
Source code: https://github.com/jrwnter/mso

Multi-objective optimization in the CDDD space

Experiment 1

Compound	dscore	EGFR [nM]	BACE1 [nM]	metstab	sol [mg/L]	perm [nm/s]	QED	SA
	0.903	190000	9.9	0.86	390	72	0.90	3.0
	0.899	37000	3.0	0.86	500	130	0.94	3.4
	0.894	72000	5.5	0.78	570	90	0.73	3.0
	0.877	49000	5.4	0.86	25	69	0.69	2.8

Outlook

Limitations:

- Most interesting molecular properties (e.g. biological activity) are only accessible by trained models (limited domain of applicability)
- Common sense of medicinal chemist is hard to model → unwanted chemistry in solutions

Next steps:

- Synthesize proposed compound to evaluate *in-vitro*
- Utilize active-learning to evolve the underlying model during an optimization process

Conclusions

Thank you!

Machine learning research
Djork-Arné Clevert
Paula Marin Zapata
Joren Retel
Santiago Villalba
Paul Kim
Tuan Le

Computational molecular design
Lara Kuhnke
Antonius ter Laak

Biomedical data sciences
Andreas Steffen