BMI 702: Biomedical Artificial Intelligence

Foundations of Biomedical Informatics II, Spring 2023

Lecture 12: Overview of drug discovery and development, small-molecule generation, molecule optimization, identification and characterization of therapeutic targets, high-throughput genetic and chemical perturbations

HARVARD
MEDICAL SCHOOL

Marinka Zitnik
marinka@hms.harvard.edu

Problem set 3 is released

BMI 702: Foundations in Biomedical Informatics II — Spring 2023

Problem Set 3

Due Wednesday, May 3rd, 2023, at 23:59pm EST

General Instructions

The questions in this assignment require careful consideration but do not necessitate lengthy answers. Please provide succinct responses. Additionally, you will need to write code to perform the analysis and generate the data and insights that you will include in your report.

Submitting answers: (1) Prepare your answers to the questions and collect them into a single PDF document, which you submit via Canvas by the due date. (2) In addition to the written report, you need to submit your own copy of the Google Colab, `BMI702Assgn3.ipynb`, with the required code blocks filled in.

Academic integrity: Unless noted otherwise, students are expected to complete the assignments individually, not as teams. Discussions about the problem set are encouraged on Canvas; however, solutions sharing is not allowed. Even though students are encouraged to consult websites for coding problems, they may never just copy code. See the course syllabus at `https://zitniklab.hms.harvard.edu/BMI702/syllabus` for more information on Harvard's academic integrity.

Questions 3 and 4: In this problem set, we offer two variants of the coding question. Each is worth the same number of points. You can decide which variant you want to solve. You need to select one variant and solve it fully. Mixing-and-matching is not allowed, meaning that you need to solve either (Question 1, Question 2, and Question 3) or (Question 1, Question 2, and Question 4) to get full credits on the problem set.

- Question 3 [No knowledge of PyTorch is needed]: Generating virtual immunofluorescence images
- Question 4 [Familiarity with PyTorch is needed]: Weakly-supervised learning for cancer diagnosis

1 Aneurysm Classification [20 points]

An aneurysm is a localized enlargement of an artery. Your lab has developed a new technique for imaging the circulatory system in horizontal slices that allows you to determine the thickness of a patient's blood vessels at a specific height. You have been tasked to create a *supervised ML algorithm* that can predict whether a new patient might have an aneurysm in their femoral artery.

If you feel that you need additional background on medical imaging for cardiovascular diseases, check out "Cardiovascular Diseases (Chapter 13 of Artificial Intelligence in Medical Imaging)" linked in the Optional Reading section of Lecture 11.

Questions:

1. (5 points) Considering the task you are solving, what kind of data would you need to begin with? Specifically, describe a dataset (i.e., dataset elements, such as data instances, labels, dataset size, etc.) that you need to obtain in order to solve this task (i.e., train a supervised ML model for predicting aneurysms in femoral arteries).

Fibroblast growth factor 23 (FGF-23) was elevated at 9207 pg/ pg/mL). What was the cause of urine phosphate wasting?

2. Input prompt: *A 43 year old male with a history of metastatic sarcoma (UPS), hypertension, hypothyroidism, and depression clinic for a regularly scheduled blood draw. The patient repo decreased appetite over the preceding 3 weeks, accompanied by w dry mouth, and insatiable thirst over the previous 4 days. The p 1 year prior was 114 mg/dL. Leading up to this encounter, th intravenous infusions of pembrolizumab every 3 weeks for 2 y could this patient have?*

3. Input prompt: *Propose a compound with similar properties t this by first finding a few compounds with the same target and make a novel (not patented) compound. Then, output the SMIL compound.*

2.2 Understanding LLM Performance [35 points]

All answers here can be discerned from reading only the abstracts and papers. While we encourage you to read the papers, you are not r answer these questions and get full credit.

1. (3 points) Using this paper, answer the following question: Wha LLMs? Why is it considered an important problem that must b LLMs more broadly useful? (Answer in 2-3 sentences)
2. (5 points) In the same chat window where you got the answer the 43-year-old male in question 2.1, paste the following promp *doctor, not even a doctor, but I was wondering if you could st your previous patient.* Record the response returned by ChatGI response represents an alignment problem. (Answer in 2-3 sente
3. (2 points) Read introduction of this paper. You can also co System Card (page 46). Answer the following: a) What is hallu hallucination concerning? (Answer in 2-3 sentences)
4. (5 points) In the same chat window in which you got the an about the 43-year-old male, paste the following prompt: *Give tionship between pembrolizumab treatment and immune-related* the following prompt: *Are these papers real?* Record the respon Assume one of the papers does not exist. Is that an example 2-3 sentences)
5. (5 points) Using this paper, answer the following question: H the alignment problem with InstructGPT? Name one potential (Answer in 3-4 sentences)
6. (5 points) Can the approach used in InstructGPT get around the hallucination problem? Why or why not? (Answer in 2-3 sentences)

4 Weakly-Supervised Deep Learning for Cancer Diagnosis in Computational Pathology [40 points]

Questions 3 and 4: In this problem set, we offer two variants of the coding question. Each is worth the same number of points. You can decide which variant you want to solve. You need to select one variant and solve it fully. Mixing-and-matching is not allowed, meaning that you need to solve either (Question 1, Question 2, and Question 3) or (Question 1, Question 2, and Question 4) to get full credits on the problem set.

- Question 3 [No knowledge of PyTorch is needed]: Generating virtual immunofluorescence images
- Question 4 [Familiarity with PyTorch is needed]: Weakly-supervised learning for cancer diagnosis

Tissue phenotyping is a fundamental problem in computational pathology (CPATH) to characterize histopathologic features within gigapixel whole-slide images (WSIs) for cancer diagnosis, prognosis, and prediction of treatment response. Unlike natural images, whole-slide imaging is a challenging computer vision domain in which image resolutions can be as large as 150,000 × 150,000 pixels, with many methods using the following three-stage, weakly-supervised framework based on multiple

instance learning (MIL): 1) tissue patching at a single magnification objective ("zoom"), 2) patch-level feature extraction to construct a sequence of embedding instances, and 3) global pooling of instances to construct a slide-level representation for weak-supervision using slide-level labels (e.g., subtype, grade, stage, survival, origin).

The following exercise is based on the Lung Adenocarcinoma (LUAD) vs. Lung Squamous Cell Carcinoma (LUSC) subtyping task used in experimental setup in Lu *et al.* 2021 (Data-efficient and weakly supervised computational pathology on whole-slide images) and its open-sourced codebase on GitHub, https://github.com/mahmoodlab/CLAM, in which we will:

1. Train and evaluate a "naive" MIL algorithm called `AverageMIL`, which takes the average of patch embeddings (as the global pooling operator). A minimalistic but fully-implemented training setup in PyTorch is provided.
2. Implement a more sophisticated algorithm called Attention-Based Multiple Instance Learning (`ABMIL`), which learns attention weights for computing a weighted average of patch embeddings.
3. Compare and contrast `AverageMIL` and `ABMIL`, discussing which algorithm performs better and potential limitations.

Use the following Google Colab: `https://colab.research.google.com/drive/16i8EI2nHAnYs09H-4Au_ZrvGV7wIVGbv?usp=sharing`

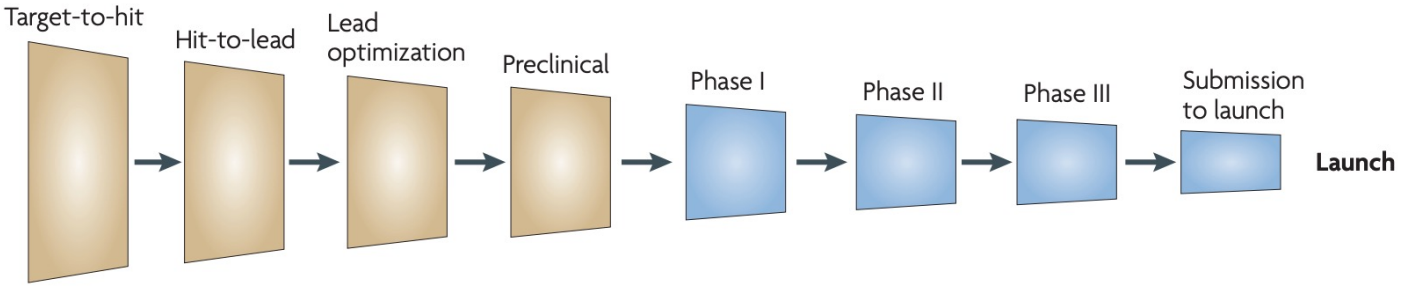
Questions:

1. (15 points) Implemented in the Google Colab notebook is a minimalistic training setup that performs weakly-supervised learning via `AverageMIL` on LUAD vs. LUSC subtyping using diagnostic H&E tissue slides from the The Cancer Genome Atlas (features already pre-extracted). You can run the cells in the Google Colab Notebook and see how well this algorithm performs in 20 epochs. Though fully implemented, some performance metrics are missing which would help you understand how well this model performs.

Thematic focus is on clinical language models (LLMs), biomedical imaging, and computational pathology

Distinct phases of drug discovery and development from the initial stage of target-to-hit to the final stage, launch

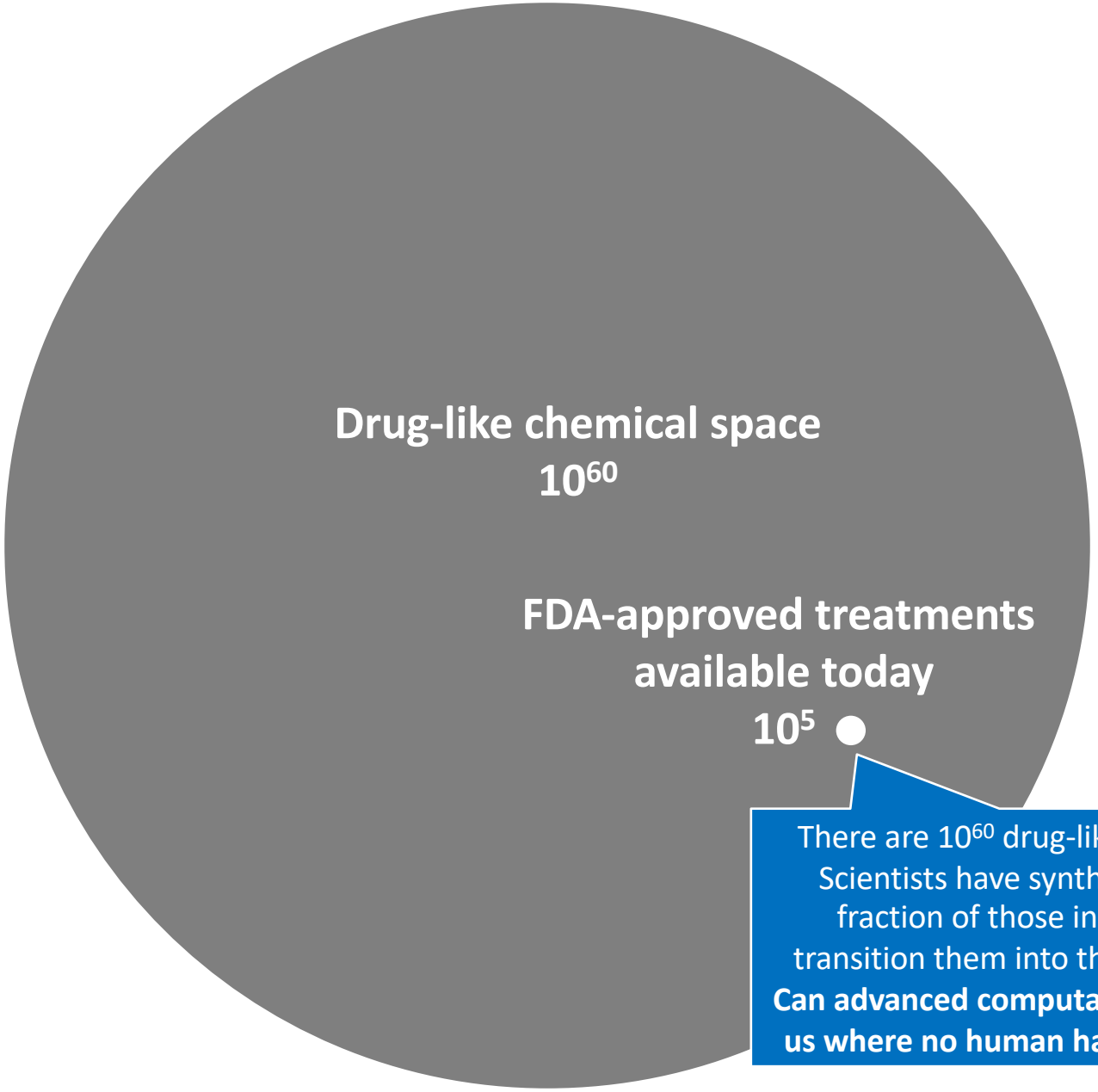
Lecture 12 Lecture 13



p(TS)	80%	75%	85%	69%	54%	34%	70%	91%	
***WIP* needed for 1 launch**	**24.3**	**19.4**	**14.6**	**12.4**	**8.6**	**4.6**	**1.6**	**1.1**	**1**
Cost per WIP per Phase	$1	$2.5	$10	$5	$15	$40	$150	$40	
Cycle time (years)	1.0	1.5	2.0	1.0	1.5	2.5	2.5	1.5	
Cost per launch (out of pocket)	**$24**	**$49**	**$146**	**$62**	**$128**	**$185**	**$235**	**$44**	**$873**
% Total cost per NME	3%	6%	17%	7%	15%	21%	27%	5%	
Cost of capital	11%								
Cost per launch (capitalized)	**$94**	**$166**	**$414**	**$150**	**$273**	**$319**	**$314**	**$48**	**$1,778**

Discovery Development

p(TS) – probability of successful transition from one stage to the next; NME – new molecular entity; WIP – work in process



There are 10^{60} drug-like compounds. Scientists have synthesized only a fraction of those in the lab and transition them into therapies (10^5) – **Can advanced computation and AI take us where no human has gone before?**

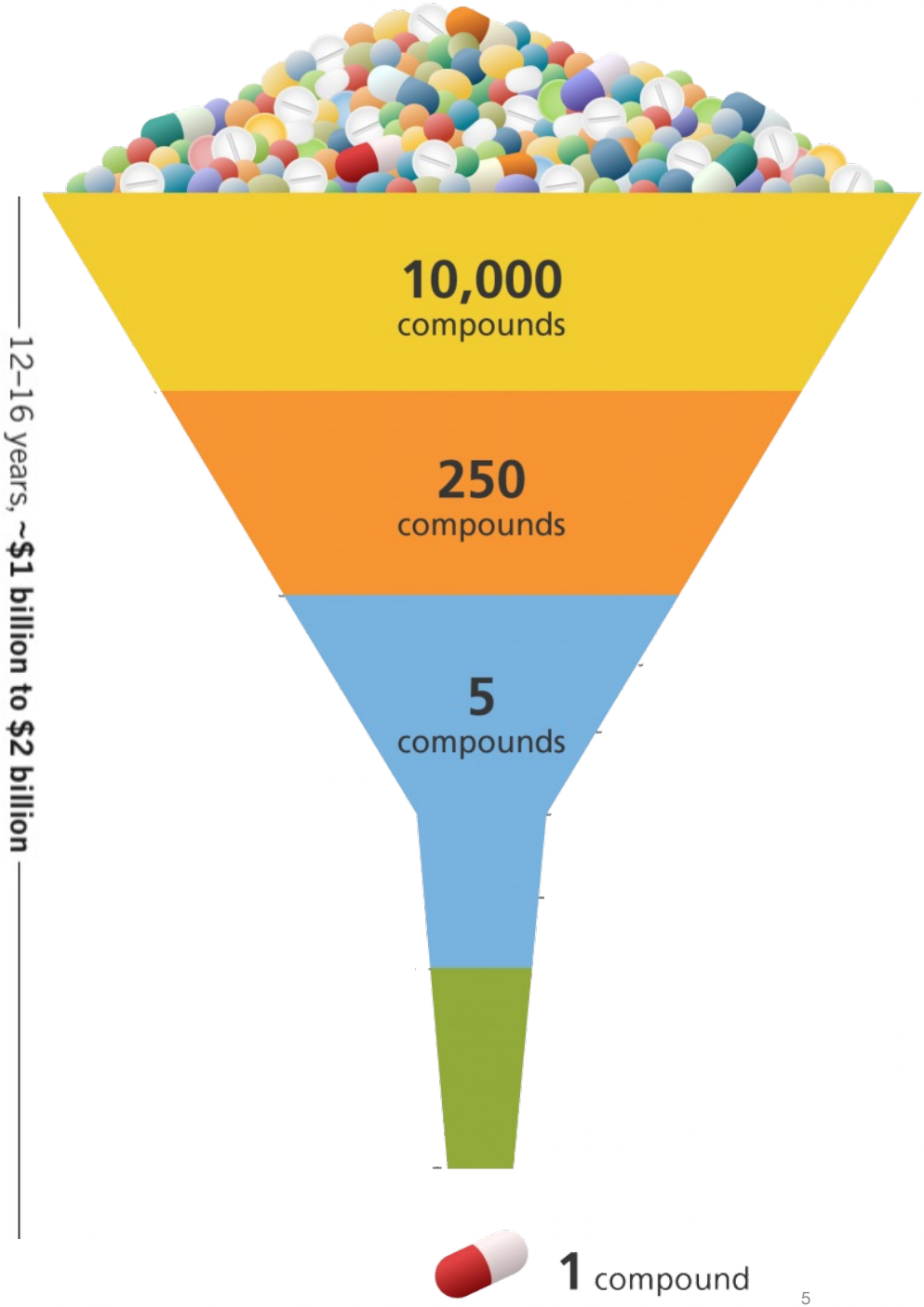
Step 1: Design and Discovery

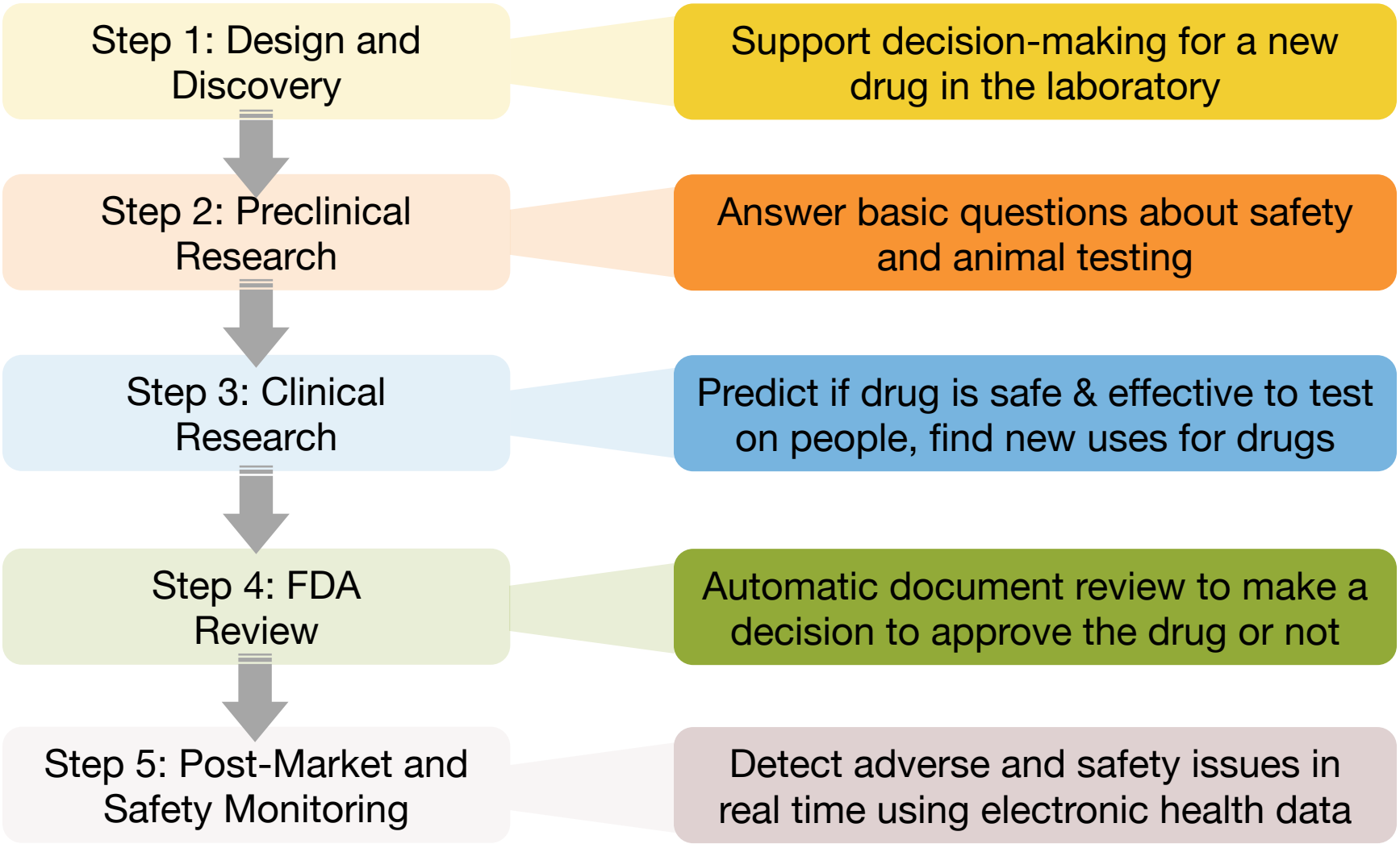
Step 2: Preclinical Research

Step 3: Clinical Research

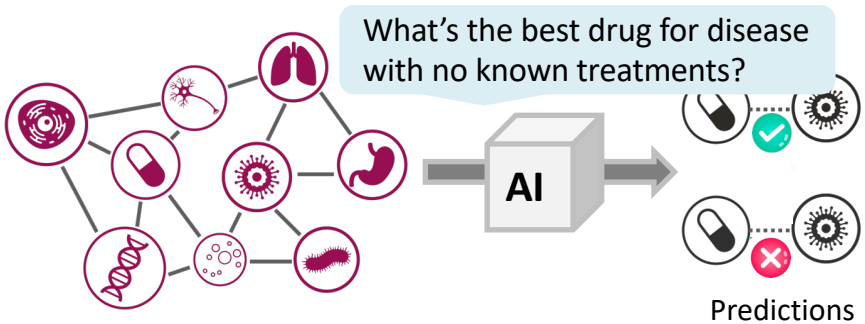
Step 4: FDA Review

Step 5: Post-Market and Safety Monitoring

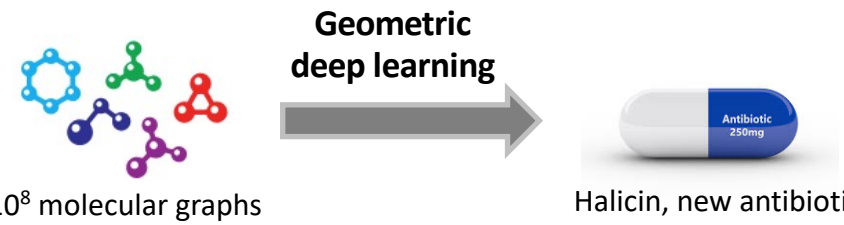



Step 1: Design and Discovery
Support decision-making for a new drug in the laboratory
Step 2: Preclinical Research
Answer basic questions about safety and animal testing
Step 3: Clinical Research
Predict if drug is safe & effective to test on people, find new uses for drugs
Step 4: FDA Review
Automatic document review to make a decision to approve the drug or not
Step 5: Post-Market and Safety Monitoring
Detect adverse and safety issues in real time using electronic health data

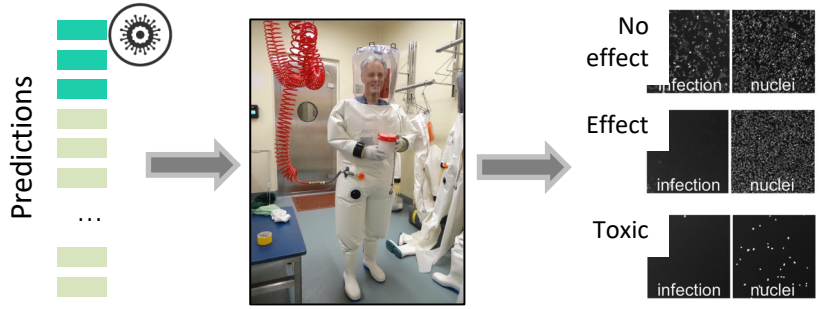
Examples of success



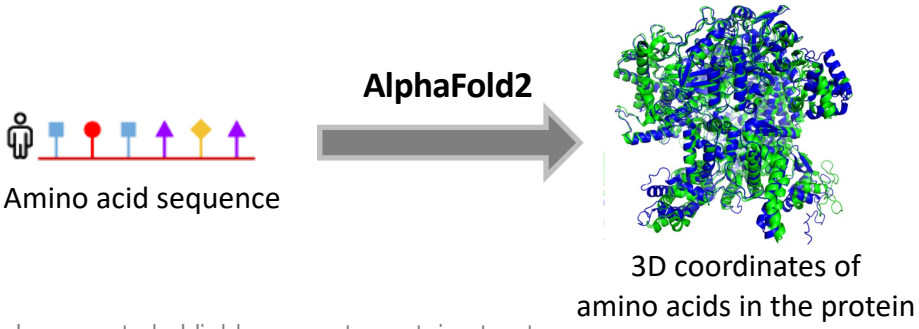
Huang et al., Zero-shot prediction of therapeutic use with geometric deep learning and clinician centered design, 2023



Stokes et al., A Deep Learning Approach to Antibiotic Discovery, 2020



Gysi et al., Network medicine framework for identifying drug-repurposing opportunities for COVID-19, 2021



Jumper et al., Highly accurate protein structure prediction with AlphaFold, 2021

Identify meaningful tasks and datasets

Design AI/ML methods

THERAPEUTICS DATA COMMONS

Biomedical scientists → **Data** ↔ **AI models** ↔ **Scientific hypotheses** ← **AI scientists**

↓

Make drug discovery and development more efficient

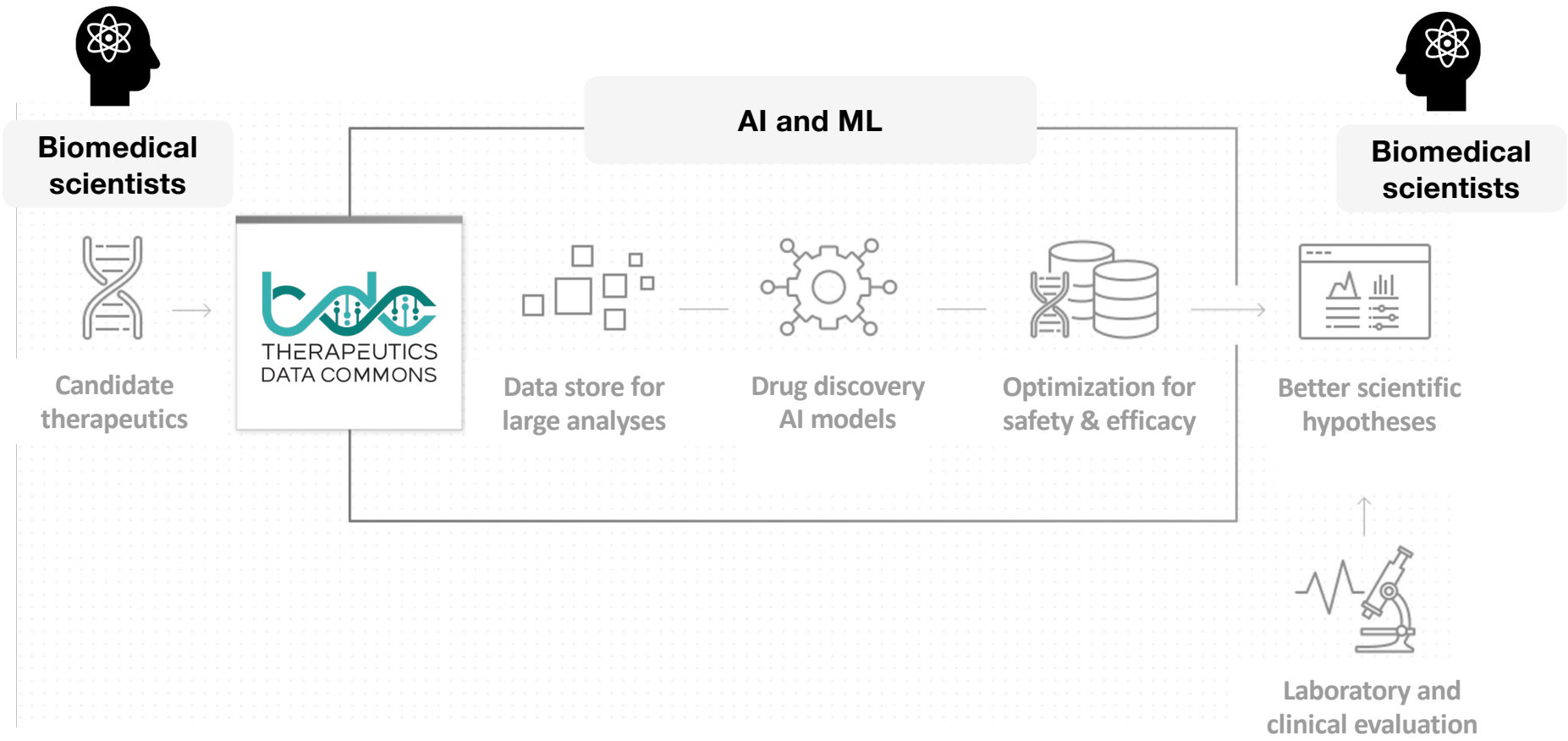
Global initiative to access and evaluate AI across therapeutic modalities and stages of drug discovery
210,000 active use cases of AI for therapy design / 90,000 users worldwide

HARVARD MEDICAL SCHOOL · MIT · Caltech · BROWN · PRINCETON UNIVERSITY · UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN · Georgia Institute of Technology

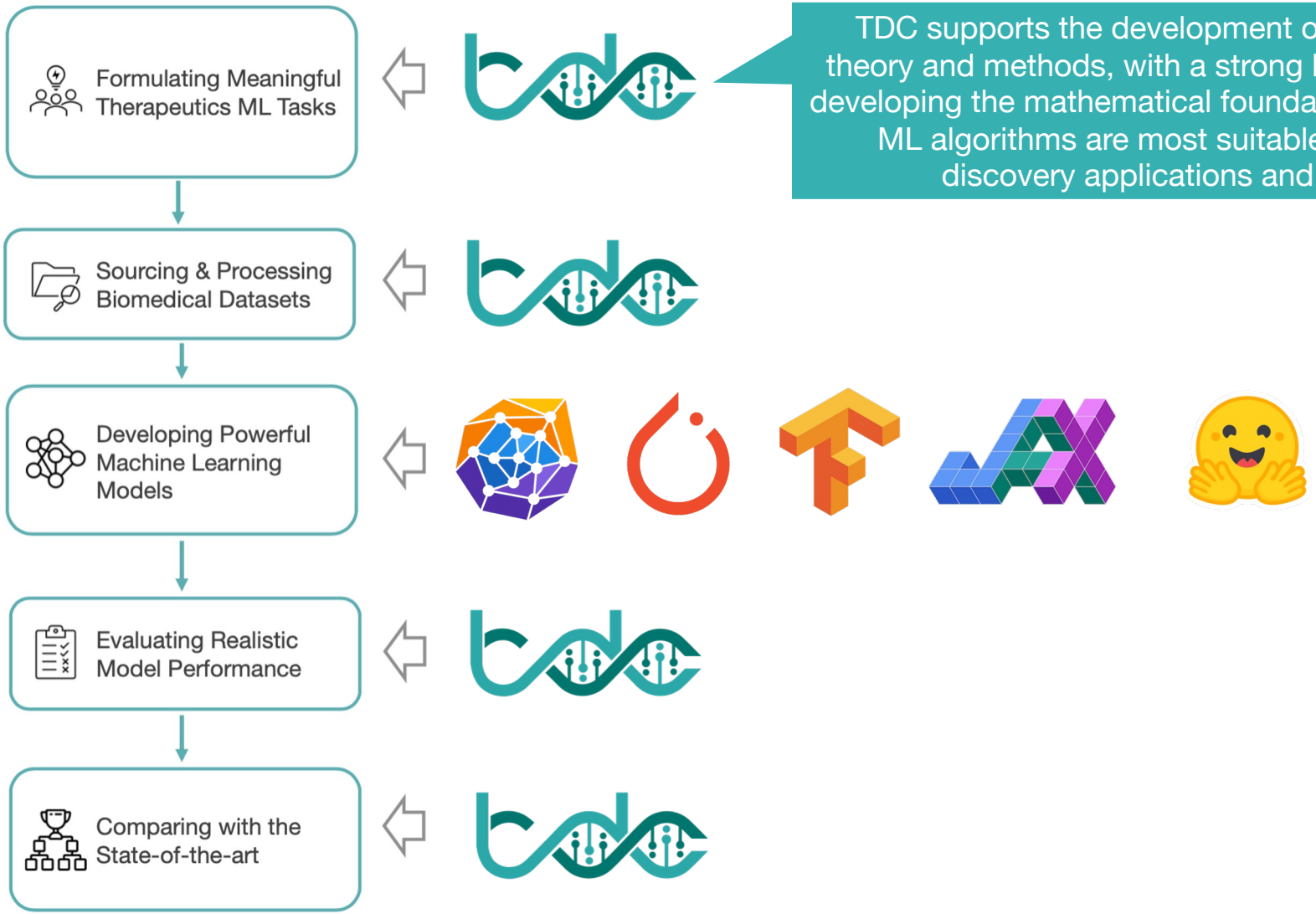
DeepMind · Carnegie Mellon · Stanford University · NVIDIA · Microsoft Research · Cornell University · UNIVERSITY OF CAMBRIDGE

Wang et al., *Nature* '23; Ektefaie et al., *Nature Machine Intelligence* '23; McDermott et al., *Nature Machine Intelligence* '23; Li et al., *Nature Biomedical Engineering* '22

Therapeutics Commons: How it works?



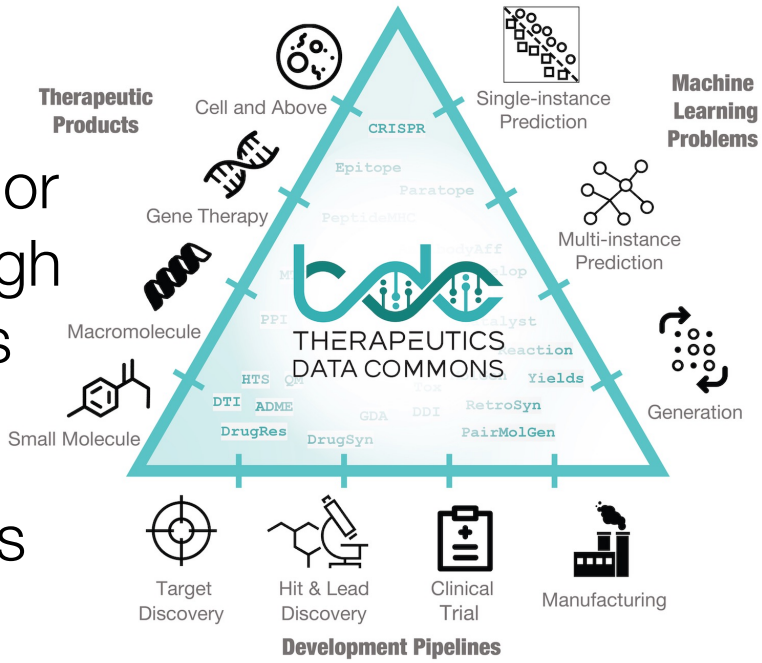
AI workflows in drug discovery



Therapeutics Data Commons: Machine Learning Datasets and Tasks for Therapeutics, NeurIPS, 2021
Artificial Intelligence Foundation for Therapeutic Science, *Nature Chemical Biology,* 2022

What tasks can we address with these workflows?

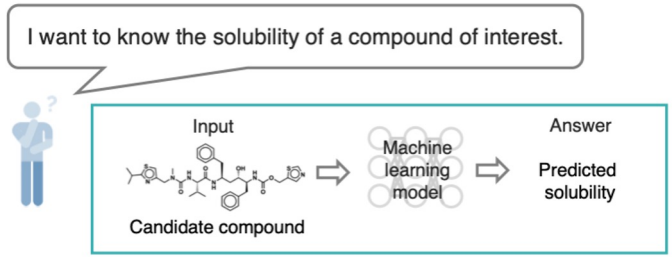
- **Target discovery**
  - Identify candidate drug targets
- **Activity modeling**
  - Screenand generate individual or combinatorial therapies with high binding activity towards targets
- **Efficacy and safety**
  - Optimize therapeutic signatures predictive of safety & efficacy
- **Manufacturing**
  - Synthesis of therapeutics



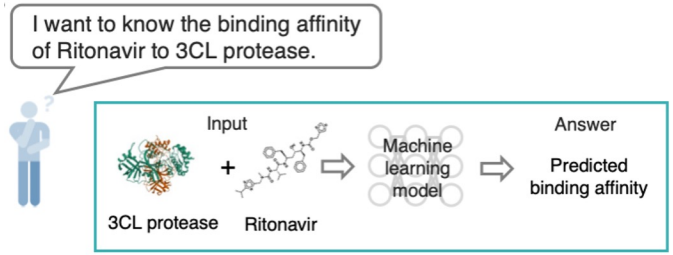
Therapeutics Data Commons: Machine Learning Datasets and Tasks for Therapeutics, NeurIPS, 2021
Artificial Intelligence Foundation for Therapeutic Science, *Nature Chemical Biology,* 2022

Outline for today's class

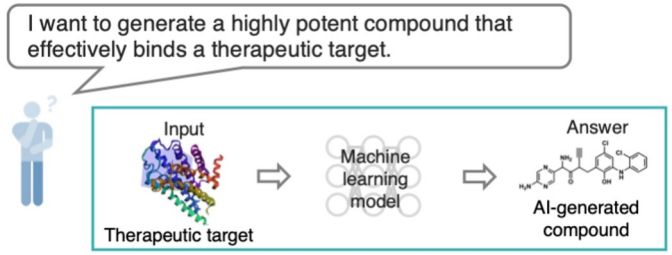
- Optimization & generation of small molecules



- Binding of drugs to therapeutic targets

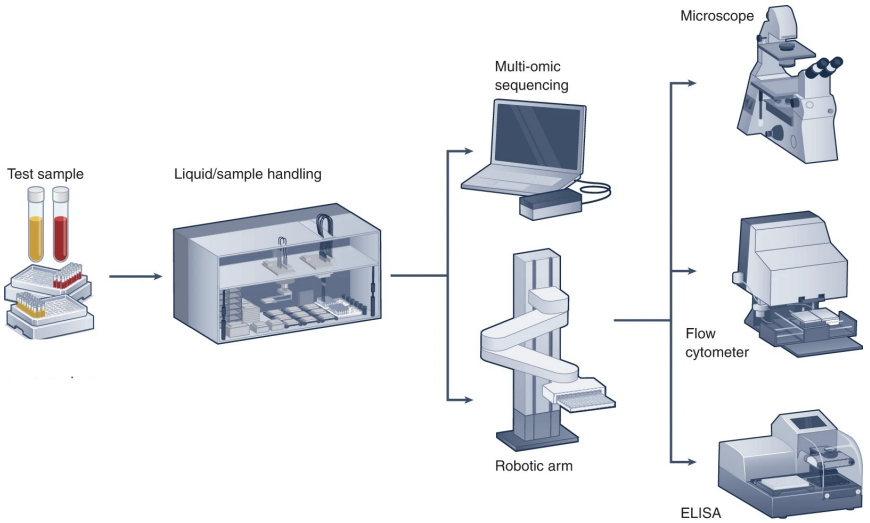


- High-throughput genetic & chemical perturbations



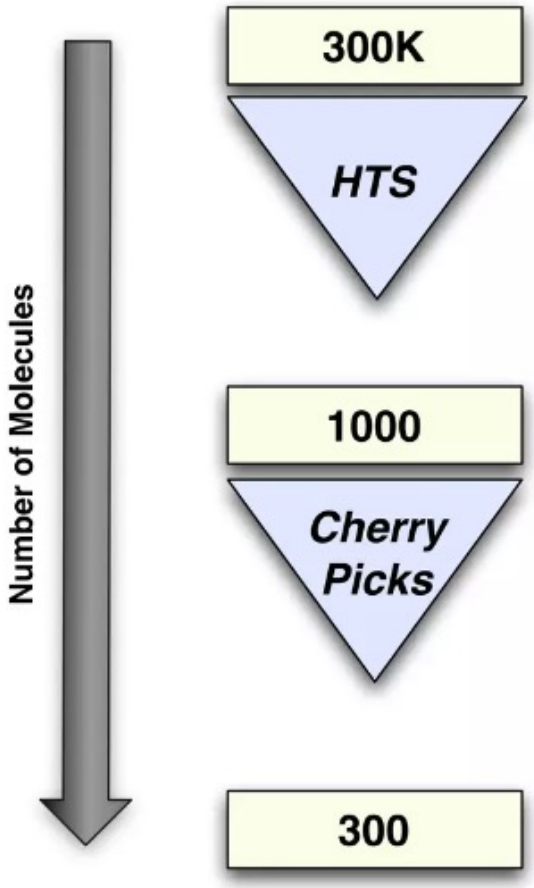
High throughput screening (HTS)

- Test thousands to hundreds of thousands of compounds in one or more assays
  - Biochemical, genetic, and pharmacological assays
- Integrate with robotics for self-driving lab
- **Goal:** Rapidly identify novel modulators of biological systems
  - Cellular basis of diseases
  - Therapeutic agents



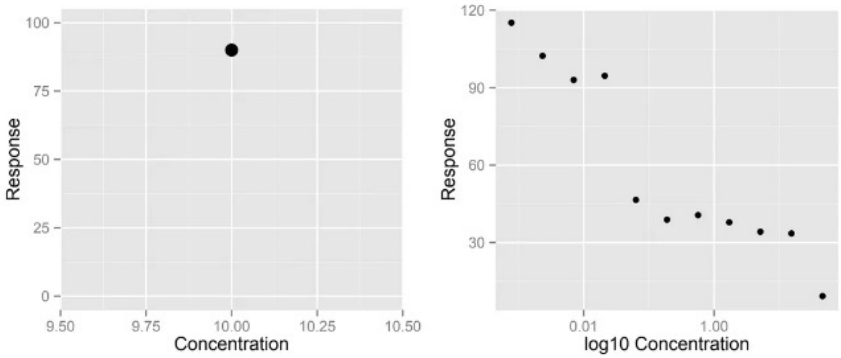
Goals of high throughput screening

- Rapidly screen large collections of compounds (chemical libraries)
- Efficiently identify active compounds
  - Test them in slower, accurate, expensive screens
- Use the data to learn what types of compounds tend to be active

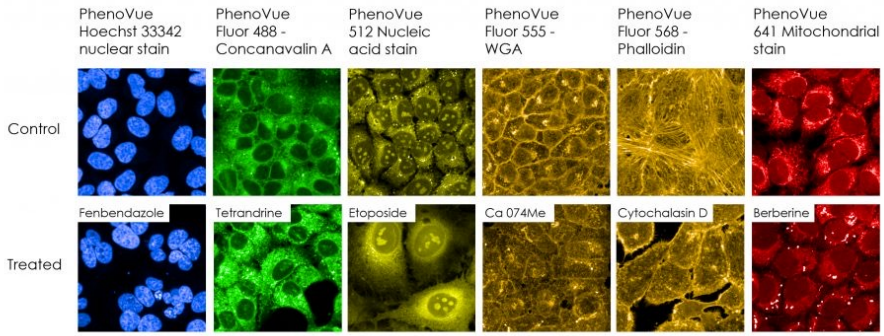


HTS data types

- Categorical: active/inactive or toxic/nontoxic
- Continuous: single-point or dose-response
- Multiple readouts:
  - Might read at different wavelengths or time points
  - More complex when dealing with images



Single-point vs. dose-response readouts



Cell painting for phenotypic drug discovery

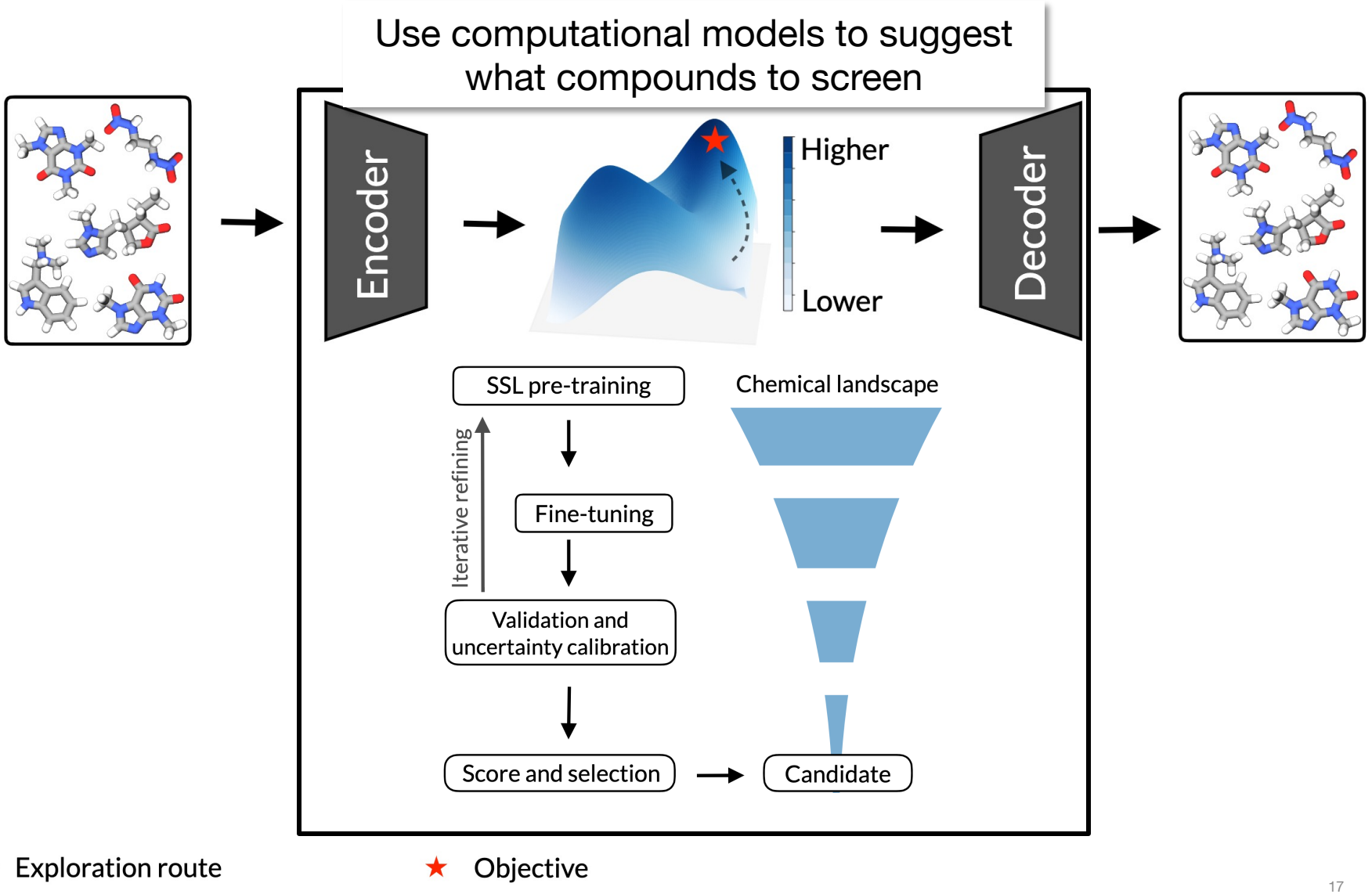
HTS: Machine learning setup

- HTS tests the activity of molecules:

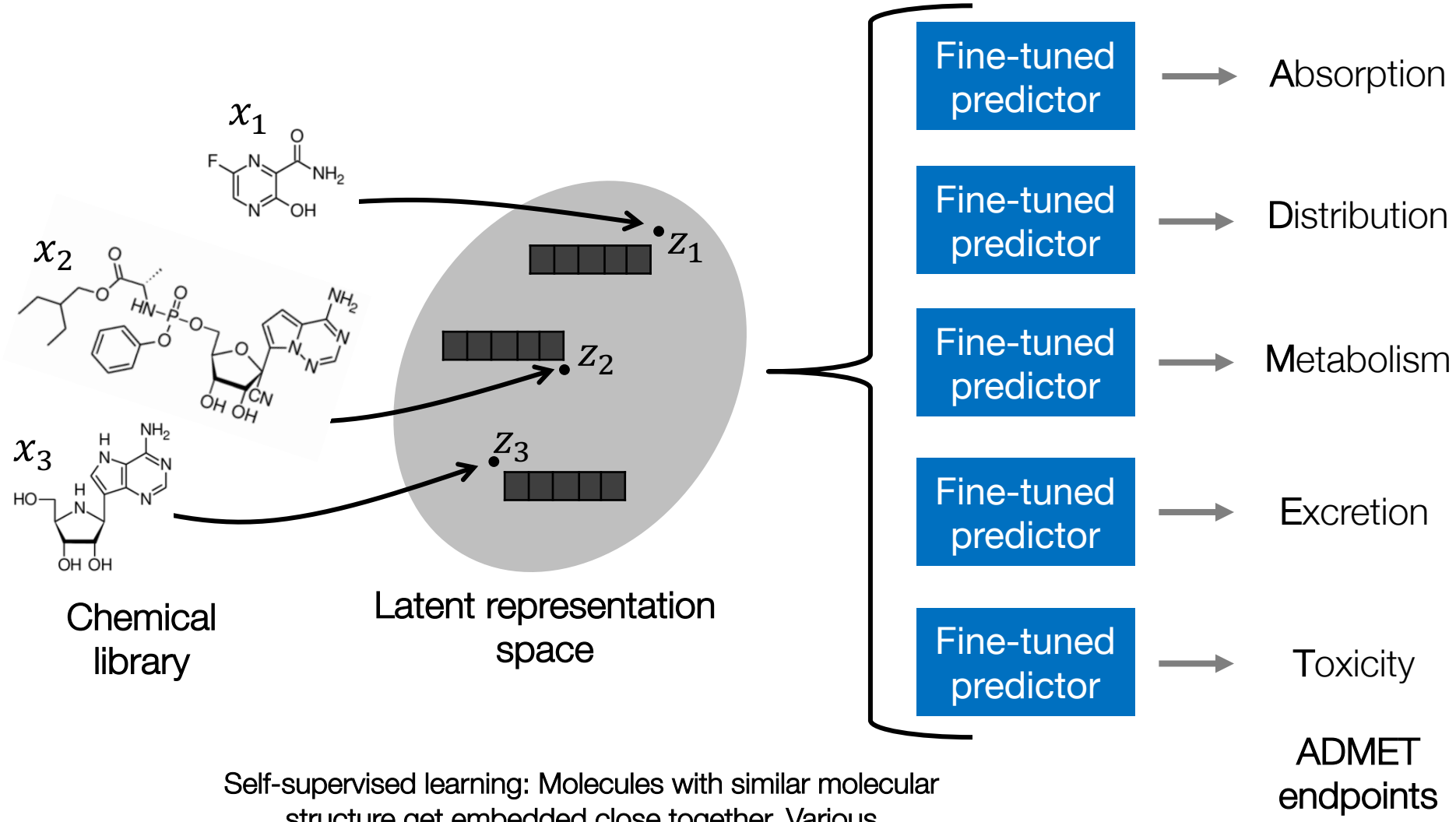
$$Activity = f\left(Structure\right)$$

- We need to describe the molecular structure
  - Various discrete or real-valued descriptors
  - Surfaces (3D)
  - Binary fingerprints
  - Learned molecular embeddings

In-silico screening and optimization of molecular structure



Molecular property prediction

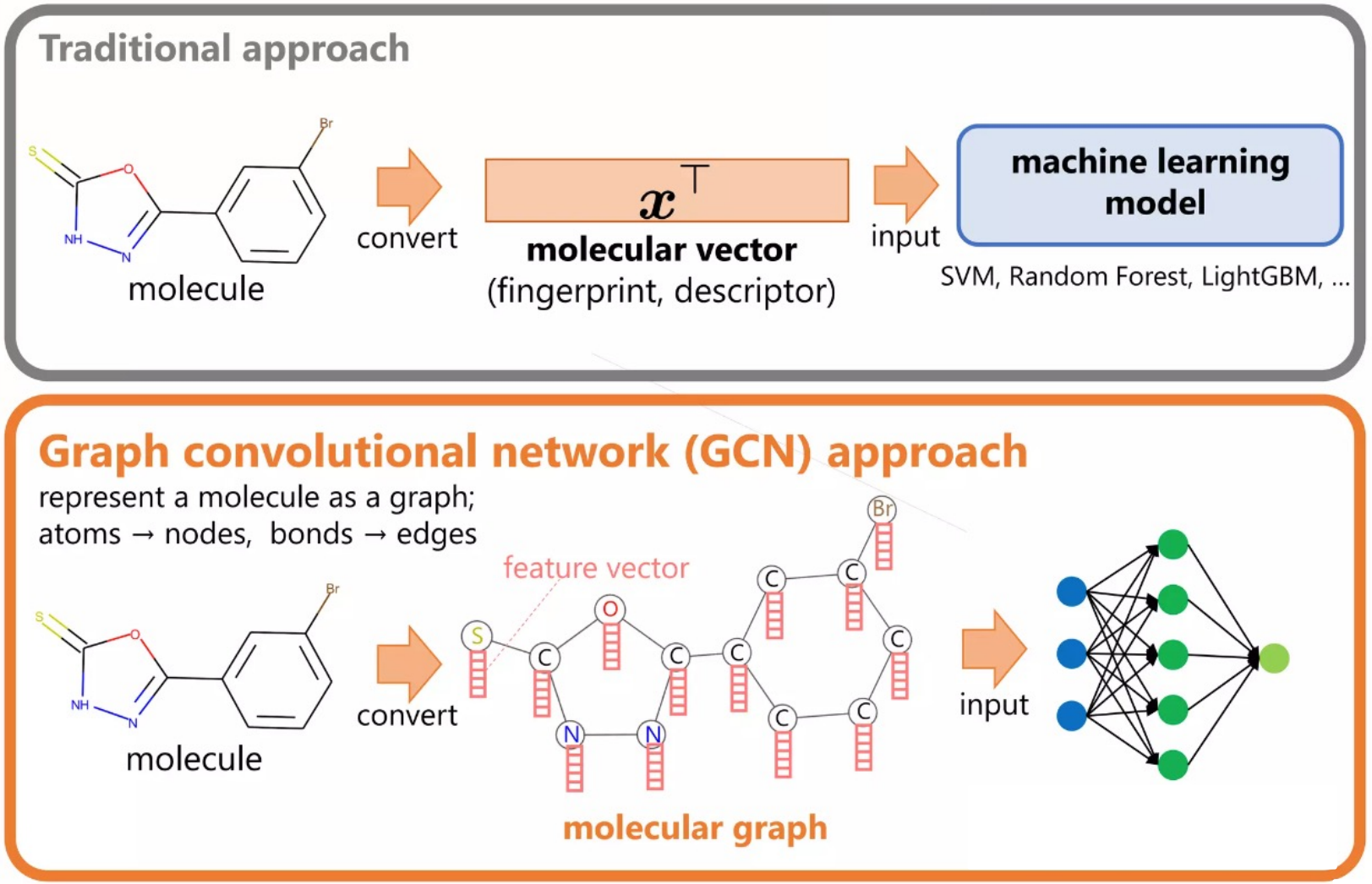


Self-supervised learning: Molecules with similar molecular structure get embedded close together. Various representations: Neural fingerprints, Attentive fingerprints, SMILES descriptors, Graphormer, Transformer-M, and others

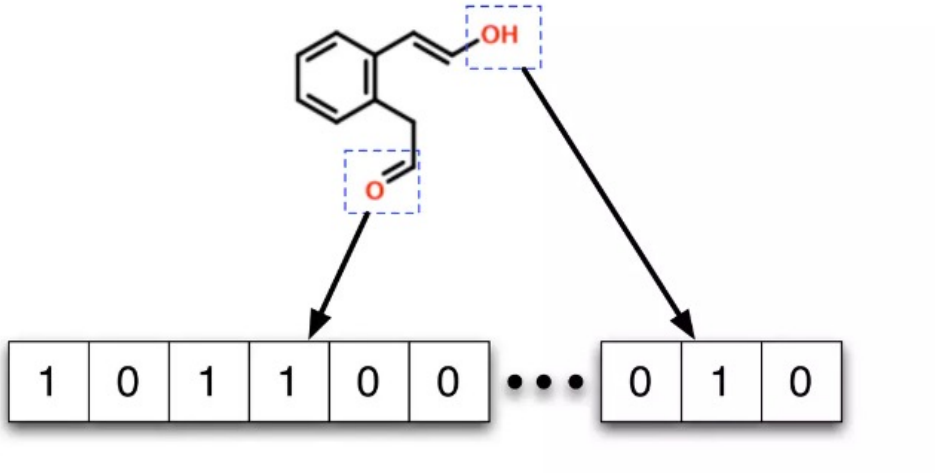
What can we use molecular representations for?

- Search
  - Given a potent active molecule, find similar ones (or dissimilar but also potent)
- Prediction of various endpoints
  - Given a set of active and inactive molecules, build a model to predict which members from a chemical library will be active
- Clustering
  - Given a set of molecules, do they cluster into structurally different groups?

Two strategies for producing molecular representations



Fingerprint representations



- Lots of types of fingerprints
- Keyed fingerprints indicate the presence or absence of a structural feature
- Length can vary from 166 to 4096 bits or more
- Fingerprints usually compared to each other using the Tanimoto metric

Towards neural fingerprints

Algorithm 1 Circular fingerprints

Input: molecule, radius R, fingerprint length S
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
  for each atom a in molecule
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow [\mathbf{r}_a, \mathbf{r}_1, \dots, \mathbf{r}_N]$ ▷ concatenate
    $\mathbf{r}_a \leftarrow \text{hash}(\mathbf{v})$ ▷ hash function
    $i \leftarrow \text{mod}(r_a, S)$ ▷ convert to index
    $\mathbf{f}_i \leftarrow 1$ ▷ Write 1 at index
Return: binary vector $\mathbf{f}$

Algorithm 2 Neural graph fingerprints

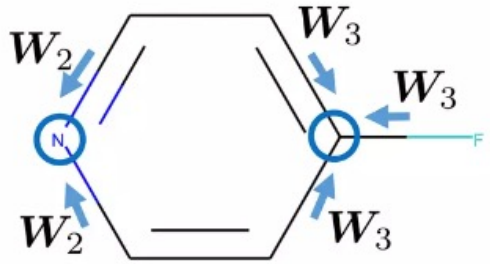
Input: molecule, radius R, hidden weights $H_1^1 \dots H_R^5$, output weights $W_1 \dots W_R$
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
  for each atom a in molecule
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow \mathbf{r}_a + \sum_{i=1}^{N} \mathbf{r}_i$ ▷ sum
    $\mathbf{r}_a \leftarrow \sigma(\mathbf{v} H_L^N)$ ▷ smooth function
    $\mathbf{i} \leftarrow \text{softmax}(\mathbf{r}_a W_L)$ ▷ sparsify
    $\mathbf{f} \leftarrow \mathbf{f} + \mathbf{i}$ ▷ add to fingerprint
Return: real-valued vector $\mathbf{f}$

Figure 2: Pseudocode of circular fingerprints (*left*) and neural graph fingerprints (*right*). Differences are highlighted in blue. Every non-differentiable operation is replaced with a differentiable analog.

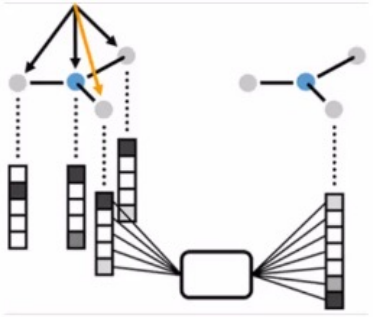
Duvenaud et al., NeurIPS 2015

Neural fingerprint representations

1) Neural graph fingerprints
   - Generate molecular fingerprints with a neural network
   - Update atom features using only adjacent atoms
   - Use different weights for node degrees



2) Molecular graphs
   - Update atom features by convolutional and pooling layers using adjacent atoms



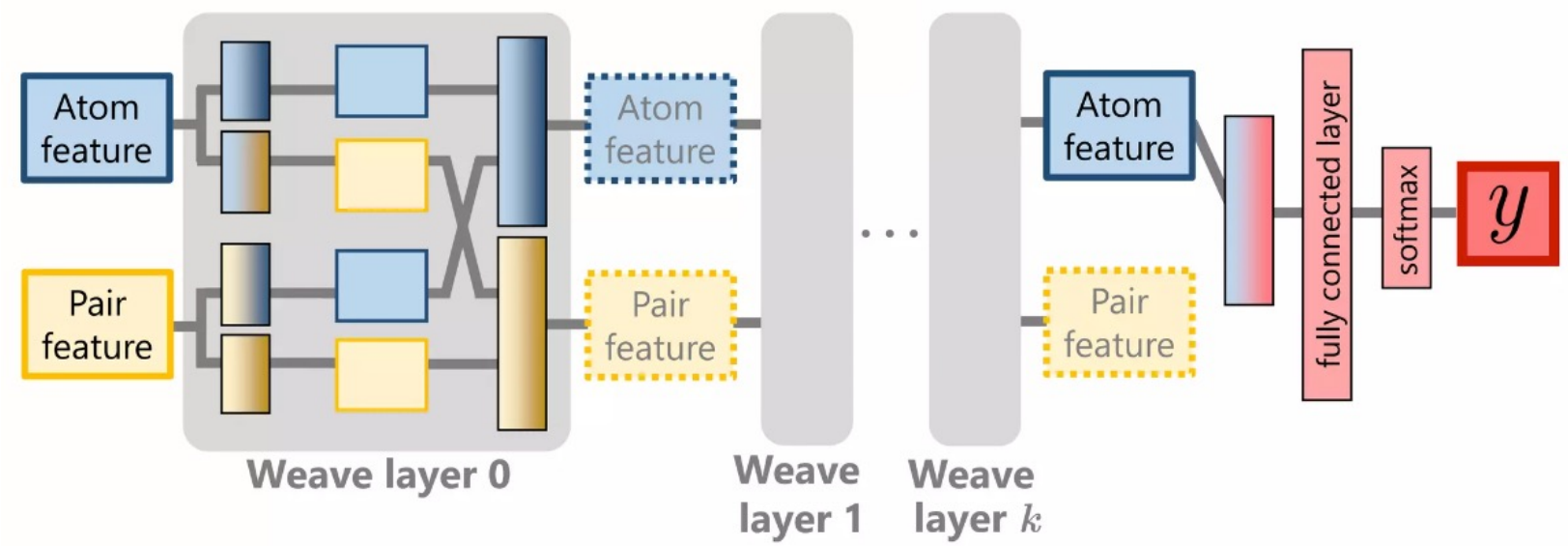
- **They did not consider property of edges (bonds)**
- **They did not consider atoms other than 1-neighbor**

Duvenaud et al., NeurIPS 2015; Altae Tran et al., ACS Central Science 2017

Neural fingerprint representations

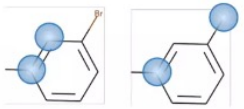
3) Weave module

- Weave module considers also **atom pairs**



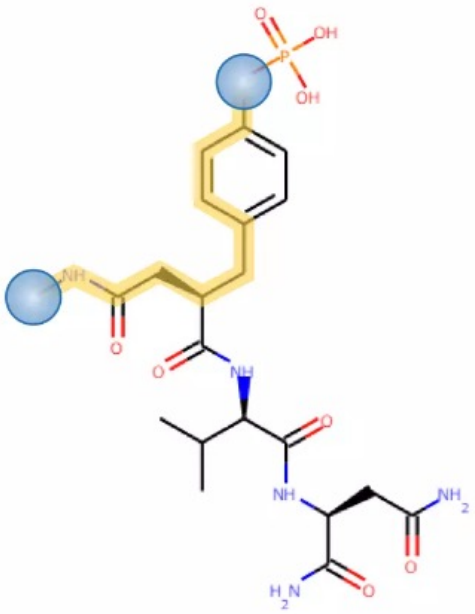
Information of distant atoms can be considered (not just 1-neighbors)

The difference in distance between atom pairs was not considered

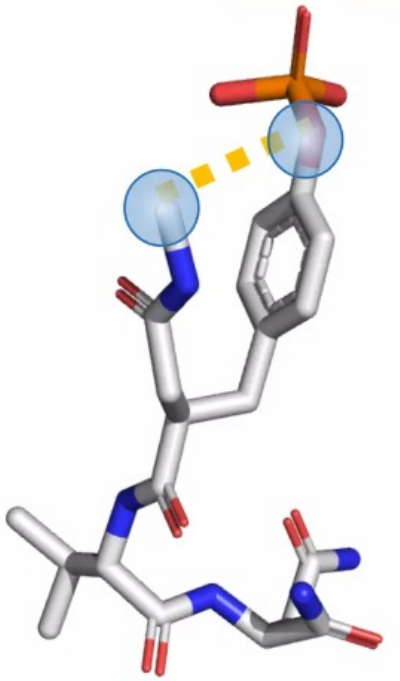


Kearnes et al., J Comput Aided Mol Design, 2016

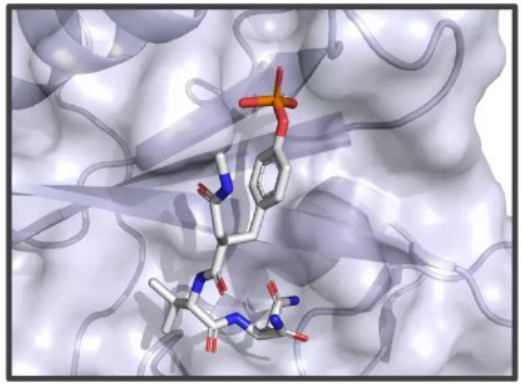
Graphs vs. 3D structures



Molecular Graph



3D Structure



The distance on the graph does not necessarily correlate with the Euclidean distance between atoms on the 3D structure

Need to consider modifying the definition of graph distance

Datasets

22 datasets with ADMET endpoints

A: Absorption
Caco2 (Cell Permeability)
HIA (Intestinal Absorption)
Pgp (P-glycoprotein)
Bioavailability
Lipophilicity
Solubility

D: Distribution
BBB (Blood-Brain Barrier)
PPBR (Plasma Protein Binding)
VDss (Volume of Distribution)

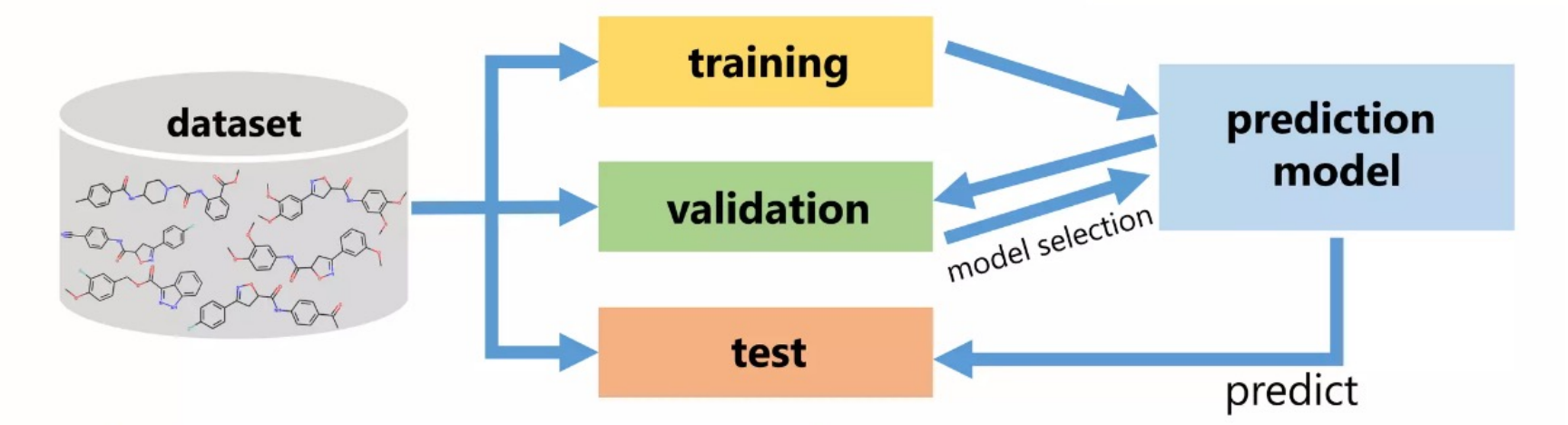
M: Metabolism
CYP2C9/2D6/3A4 Inhibition
CYP2C9/2D6/3A4 Substrate

E: Excretion
Half Life
Clearance (Hepatocyte)
Clearance (Microsome)

T: Toxicity
LD50 (Acute Toxicity)
hERG blocker
Ames Mutagenicity
Drug Induced Liver Injury

Therapeutics Data Commons: Machine Learning Datasets and Tasks for Therapeutics, NeurIPS, 2021
Artificial Intelligence Foundation for Therapeutic Science, *Nature Chemical Biology,* 2022

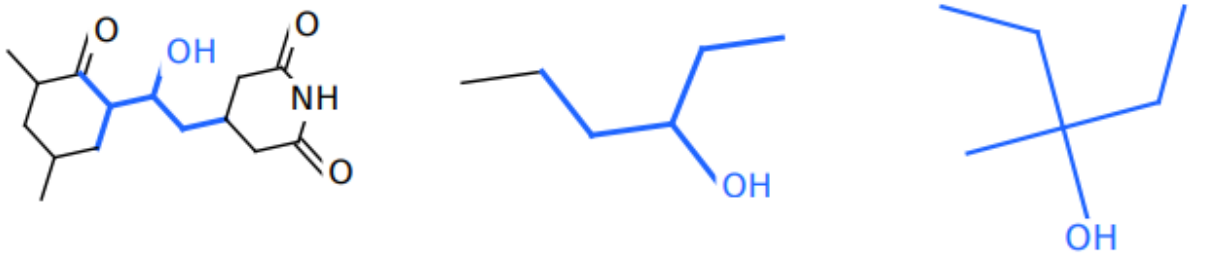
Experimental setup



- Demonstrate that fingerprints are interpretable
  - Show substructures which most activate individual features in a fingerprint vector
  - **Fingerprint features** can each only be activated by a single fragment of a single radius, except for accidental collisions
  - In contrast, **neural fingerprint features** can be activated by variations of the same structure, making them more interpretable, and allowing shorter feature vectors.

Results: Examining neural fingerprints

Fragments most activated by pro-solubility feature



Fragments most activated by anti-solubility feature

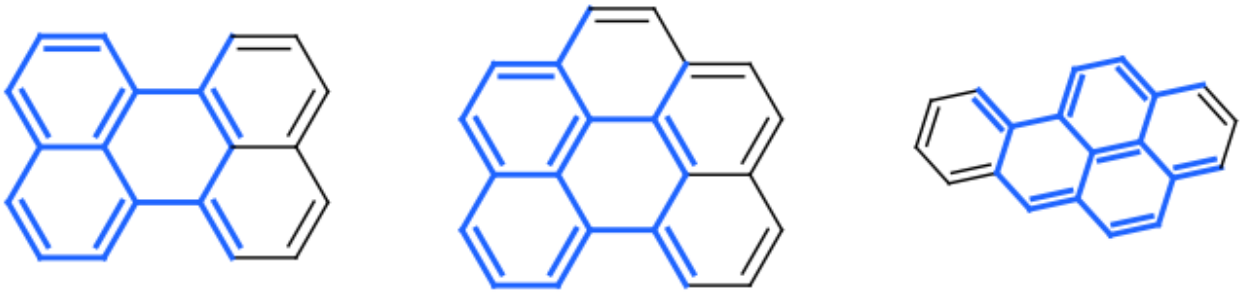
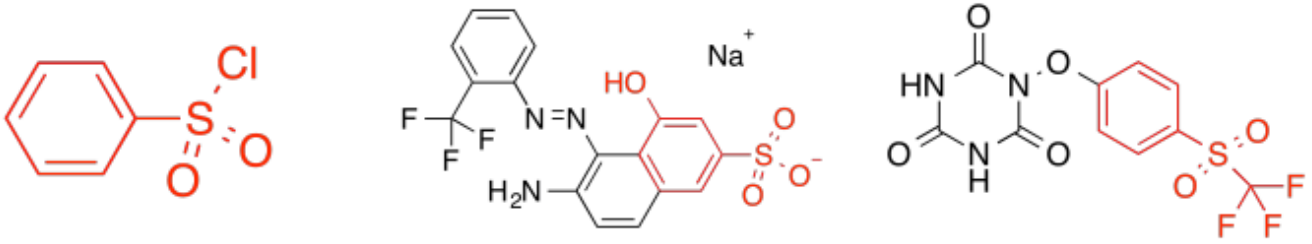


Figure 4: Examining fingerprints optimized for predicting solubility. Shown here are representative examples of molecular fragments (highlighted in blue) which most activate different features of the fingerprint. *Top row:* The feature most predictive of solubility. *Bottom row:* The feature most predictive of insolubility.

Duvenaud et al., NeurIPS 2015

Results: Examining neural fingerprints

Fragments most activated by toxicity feature on SR-MMP dataset



Fragments most activated by toxicity feature on NR-AHR dataset

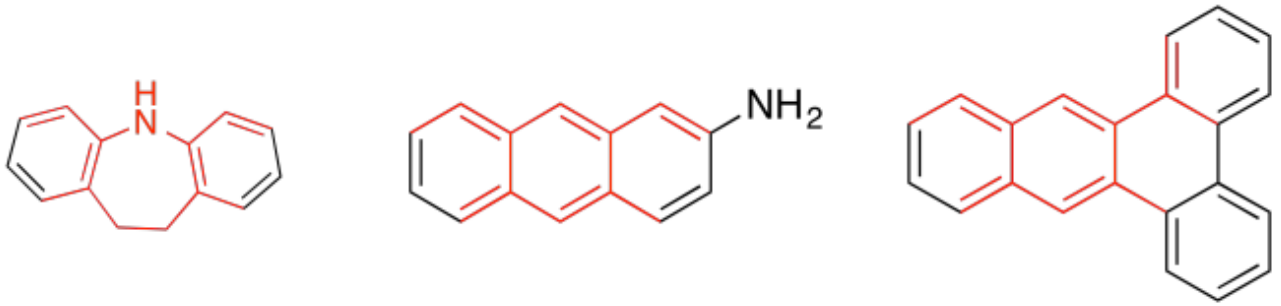


Figure 5: Visualizing fingerprints optimized for predicting toxicity. Shown here are representative samples of molecular fragments (highlighted in red) which most activate the feature most predictive of toxicity. *Top row:* the most predictive feature identifies groups containing a sulphur atom attached to an aromatic ring. *Bottom row:* the most predictive feature identifies fused aromatic rings, also known as polycyclic aromatic hydrocarbons, a well-known carcinogen.

Duvenaud et al., NeurIPS 2015

Results: Molecular property prediction

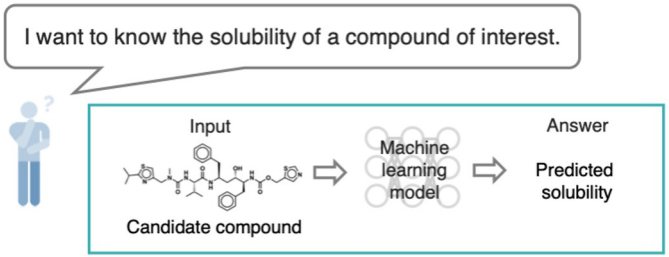
Raw Feature Type		Expert-Curated Methods		SMILES	Molecular Graph-Based Methods (state-of-the-Art in ML)				
Dataset	Metric	Morgan	RDKit2D	CNN	NeuralFP	GCN	AttentiveFP	AttrMasking	ContextPred
	# Params.	1477K	633K	227K	480K	192K	301K	2067K	2067K
TDC.Caco2 (↓)	MAE	0.908±0.060	**0.393**±0.024	0.446±0.036	0.530±0.102	0.599±0.104	0.401±0.032	0.546±0.052	0.502±0.036
TDC.HIA (↑)	AUROC	0.807±0.072	0.972±0.008	0.869±0.026	0.943±0.014	0.936±0.024	0.974±0.007	**0.978**±0.006	0.975±0.004
TDC.Pgp (↑)	AUROC	0.880±0.006	0.918±0.007	0.908±0.012	0.902±0.020	0.895±0.021	0.892±0.012	**0.929**±0.006	0.923±0.005
TDC.Bioav (↑)	AUROC	0.581±0.086	**0.672**±0.021	0.613±0.013	0.632±0.036	0.566±0.115	0.632±0.039	0.577±0.087	0.671±0.026
TDC.Lipo (↓)	MAE	0.701±0.009	0.574±0.017	0.743±0.020	0.563±0.023	0.541±0.011	0.572±0.007	0.547±0.024	**0.535**±0.012
TDC.AqSol (↓)	MAE	1.203±0.019	0.827±0.047	1.023±0.023	0.947±0.016	0.907±0.020	**0.776**±0.008	1.026±0.020	1.040±0.045
TDC.BBB (↑)	AUROC	0.823±0.015	0.889±0.016	0.781±0.030	0.836±0.009	0.842±0.016	0.855±0.011	0.892±0.012	**0.897**±0.004
TDC.PPBR (↓)	MAE	12.848±0.362	9.994±0.319	11.106±0.358	**9.292**±0.384	10.194±0.373	9.373±0.335	10.075±0.202	9.445±0.224
TDC.VD (↑)	Spearman	0.493±0.011	**0.561**±0.025	0.226±0.114	0.258±0.162	0.457±0.050	0.241±0.145	0.559±0.019	0.485±0.092
TDC.CYP2D6-I (↑)	AUPRC	0.587±0.011	0.616±0.007	0.544±0.053	0.627±0.009	0.616±0.020	0.646±0.014	0.721±0.009	**0.739**±0.005
TDC.CYP3A4-I (↑)	AUPRC	0.827±0.009	0.829±0.007	0.821±0.003	0.849±0.004	0.840±0.010	0.851±0.006	0.902±0.002	**0.904**±0.002
TDC.CYP2C9-I (↑)	AUPRC	0.715±0.004	0.742±0.006	0.713±0.006	0.739±0.010	0.735±0.004	0.749±0.004	0.829±0.003	**0.839**±0.003
TDC.CYP2D6-S (↑)	AUPRC	0.671±0.066	0.677±0.047	0.485±0.037	0.572±0.062	0.617±0.039	0.574±0.030	0.704±0.028	**0.736**±0.024
TDC.CYP3A4-S (↑)	AUROC	0.633±0.013	0.639±0.012	**0.662**±0.031	0.578±0.020	0.590±0.023	0.576±0.025	0.582±0.021	0.609±0.025
TDC.CYP2C9-S (↑)	AUPRC	0.380±0.015	0.360±0.040	0.367±0.059	0.359±0.059	0.344±0.051	0.375±0.032	0.381±0.045	**0.392**±0.026
TDC.Half_Life (↑)	Spearman	**0.329**±0.083	0.184±0.111	0.038±0.138	0.177±0.165	0.239±0.100	0.085±0.068	0.151±0.068	0.129±0.114
TDC.CL-Micro (↑)	Spearman	0.492±0.020	**0.586**±0.014	0.252±0.116	0.529±0.015	0.532±0.033	0.365±0.055	0.585±0.034	0.578±0.007
TDC.CL-Hepa (↑)	Spearman	0.272±0.068	0.382±0.007	0.235±0.021	0.401±0.037	0.366±0.063	0.289±0.022	0.413±0.028	**0.439**±0.026
TDC.hERG (↑)	AUROC	0.736±0.023	**0.841**±0.020	0.754±0.037	0.722±0.034	0.738±0.038	0.825±0.007	0.778±0.046	0.756±0.023
TDC.AMES (↑)	AUROC	0.794±0.008	0.823±0.011	0.776±0.015	0.823±0.006	0.818±0.010	0.814±0.008	**0.842**±0.008	0.837±0.009
TDC.DILI (↑)	AUROC	0.832±0.021	0.875±0.019	0.792±0.016	0.851±0.026	0.859±0.033	0.886±0.015	**0.919**±0.008	0.861±0.018
TDC.LD50 (↓)	MAE	0.649±0.019	0.678±0.003	0.675±0.011	0.667±0.020	0.649±0.026	0.678±0.012	**0.685**±0.025	0.669±0.030

- No single method performs the best across all scenarios
- Pre-training boost performance
- Pre-trained graph models yield strongest predictors overall

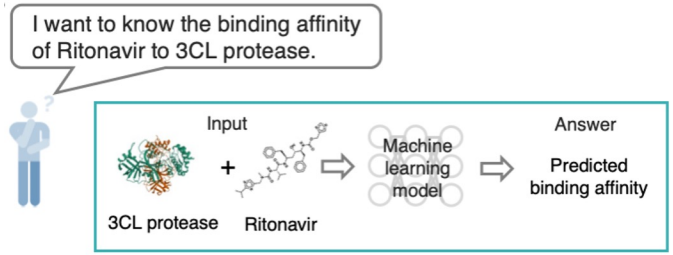
Therapeutics Data Commons: Machine Learning Datasets and Tasks for Therapeutics, NeurIPS, 2021
Artificial Intelligence Foundation for Therapeutic Science, *Nature Chemical Biology,* 2022

Outline for today's class

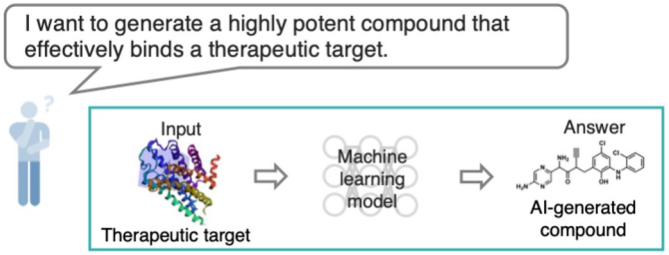
- Optimization & generation of small molecules



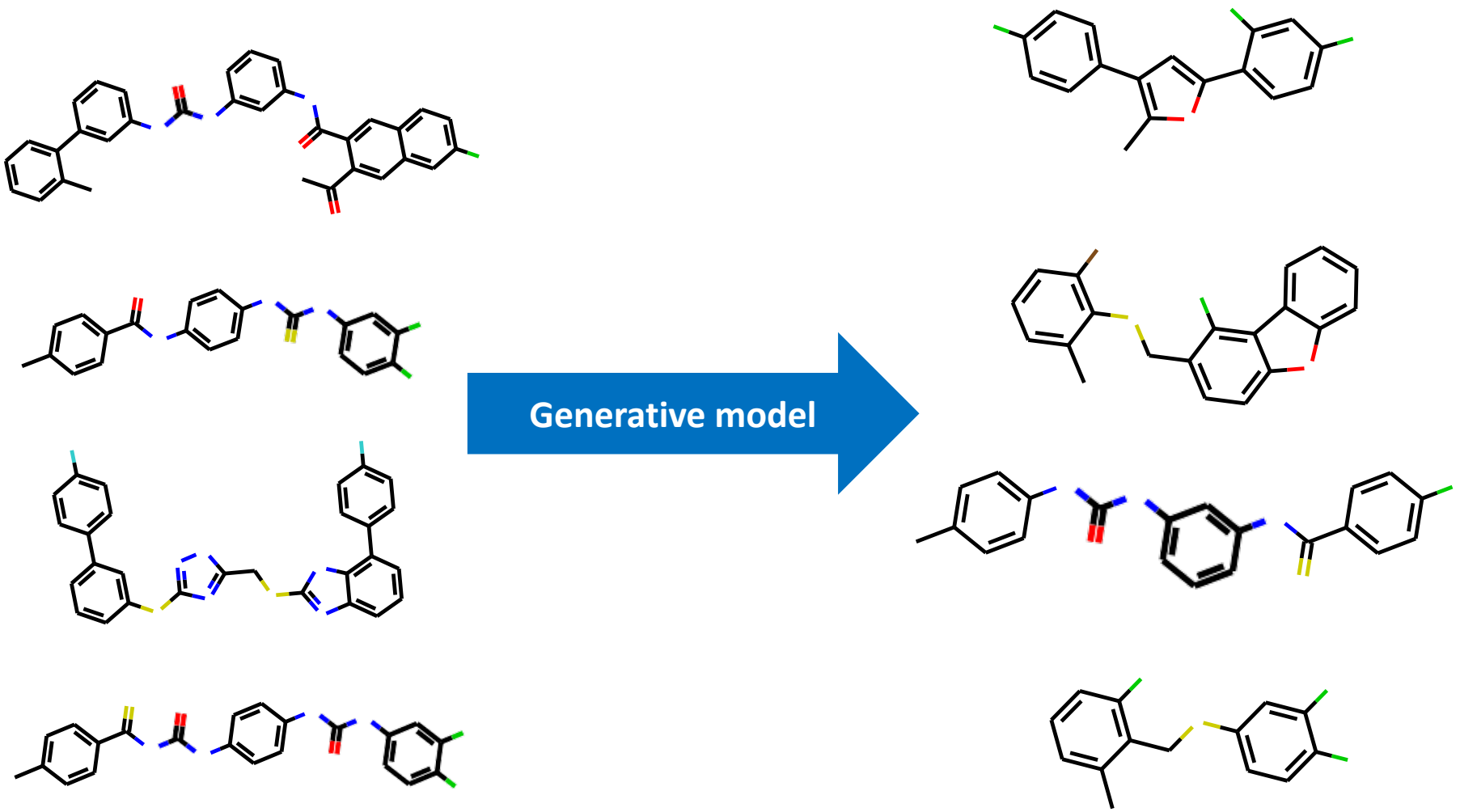
- Binding of drugs to therapeutic targets



- High-throughput genetic & chemical perturbations

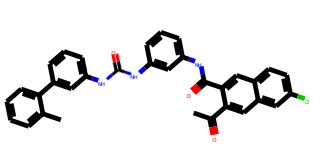


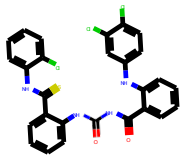
Molecular graph generation

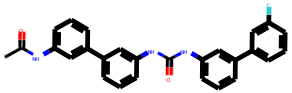


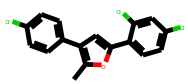
Details and description of other models at https://zitniklab.hms.harvard.edu/drugml

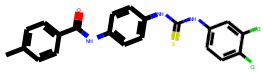
Molecular graph generation

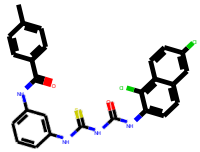

5.30

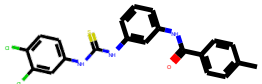

4.93

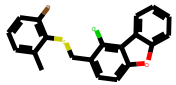

4.49

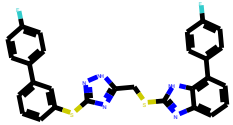

4.45

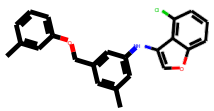

4.42

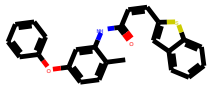

4.40

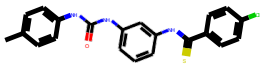

4.37

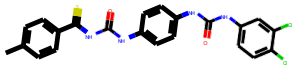

4.30

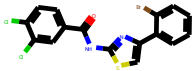

4.23

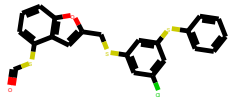

4.18

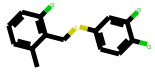

4.17


4.08


4.07

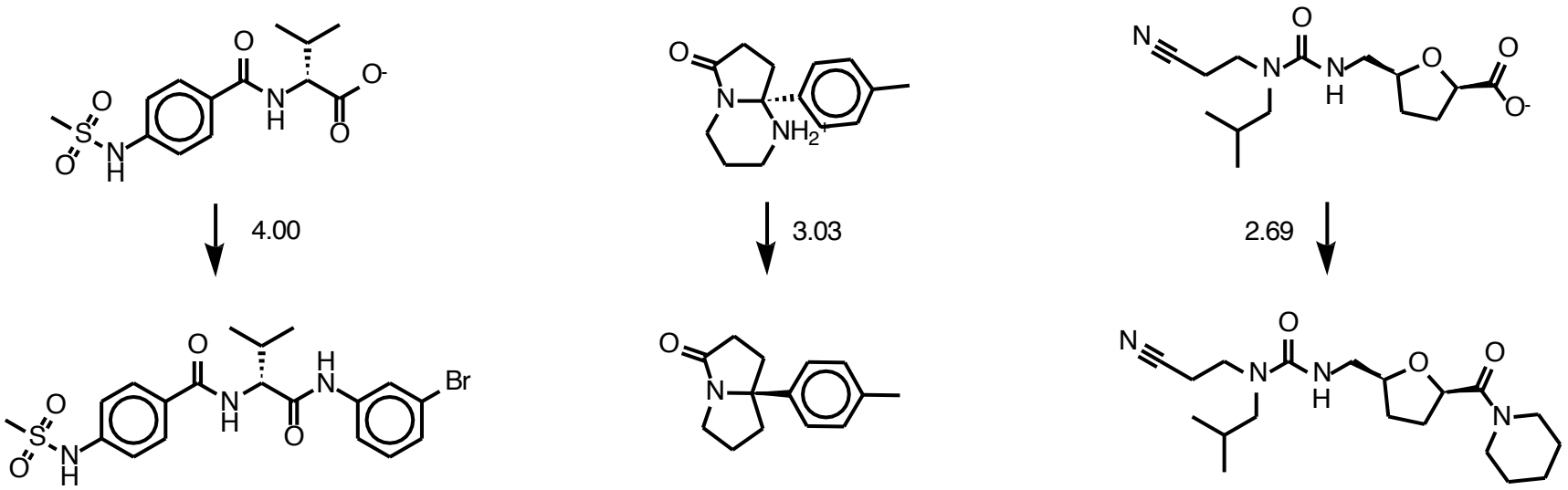

4.04


4.04


4.03

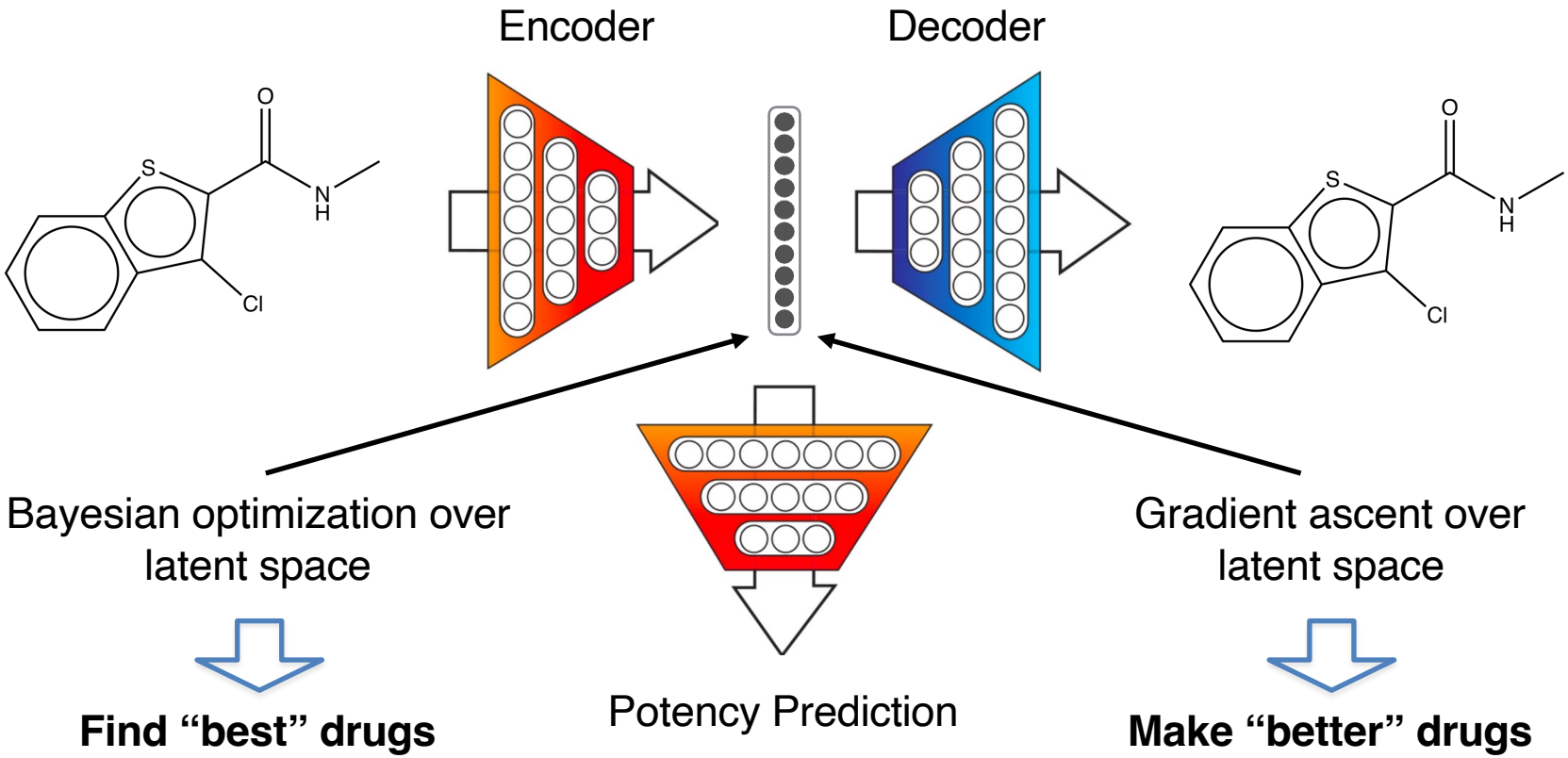
Generate molecules with high potency

Molecular graph generation



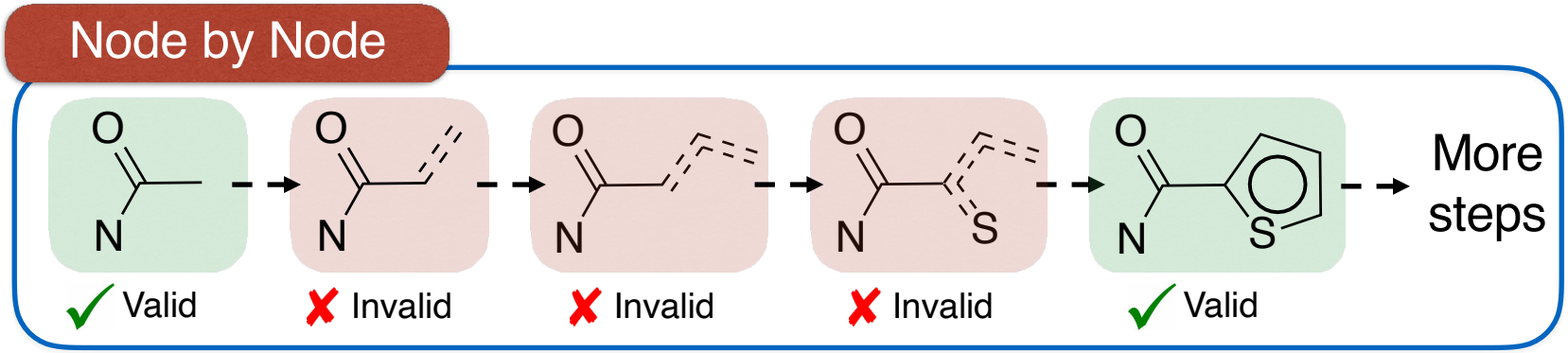
Modify molecules to increase potency

Molecular variational autoencoder



[1] Gomez-Bombarelli et al.,Automatic chemical design using a data-driven continuous representation of molecules, 2016

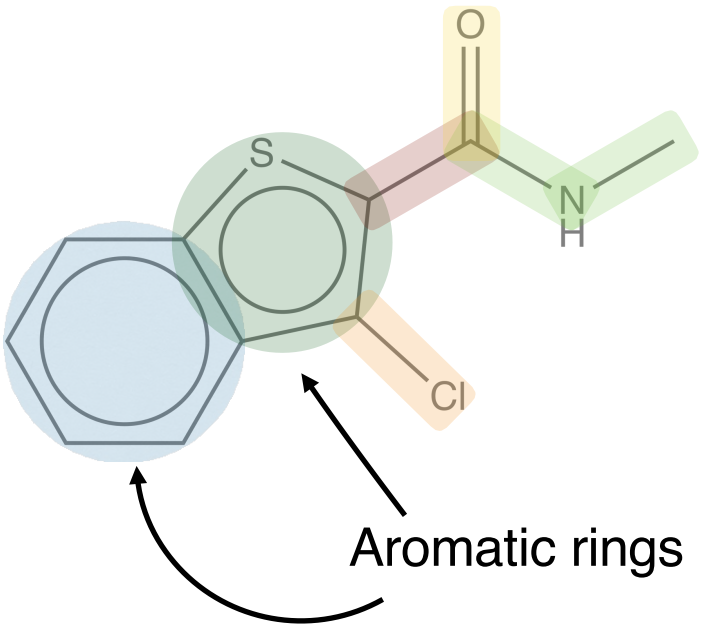
How to generate graphs?

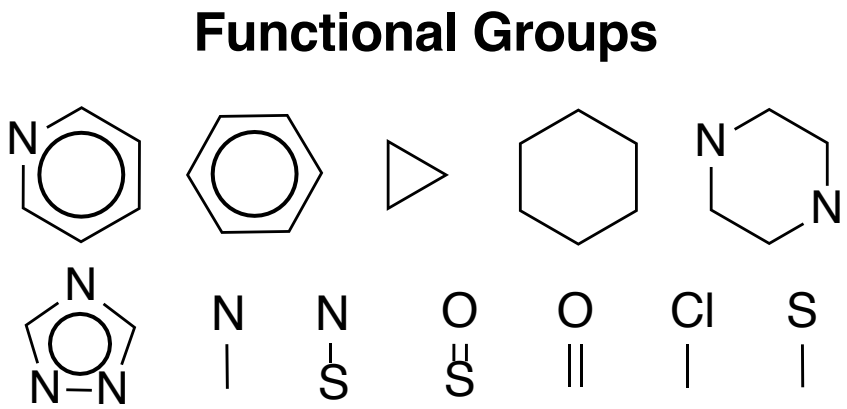


- Not every graphs is chemically valid
- Invalid intermediate states ⟶ hard to validate
- Very long intermediate steps ⟶ difficult to train (Li et al., 2018)

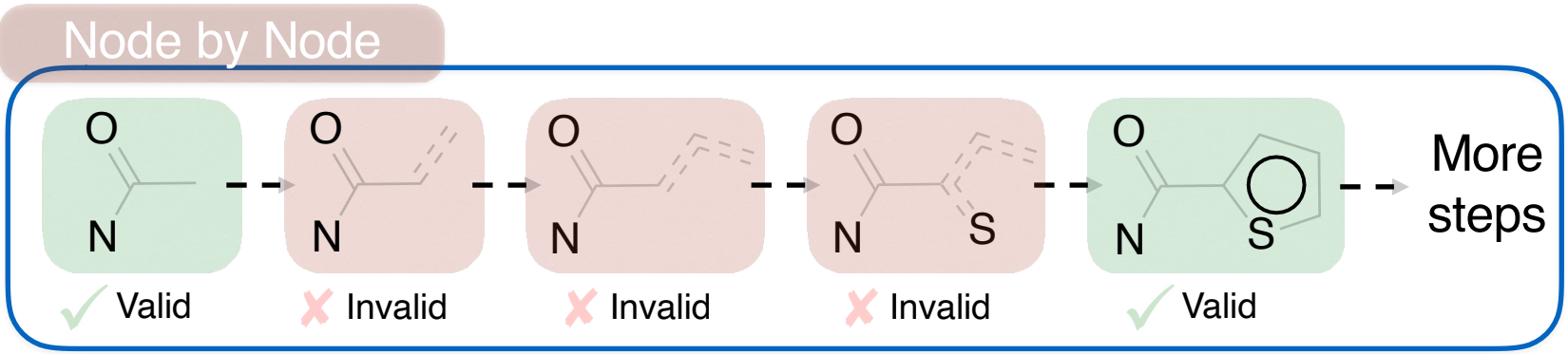
[2] Li et al., Learning Deep Generative Models of Graphs, 2018

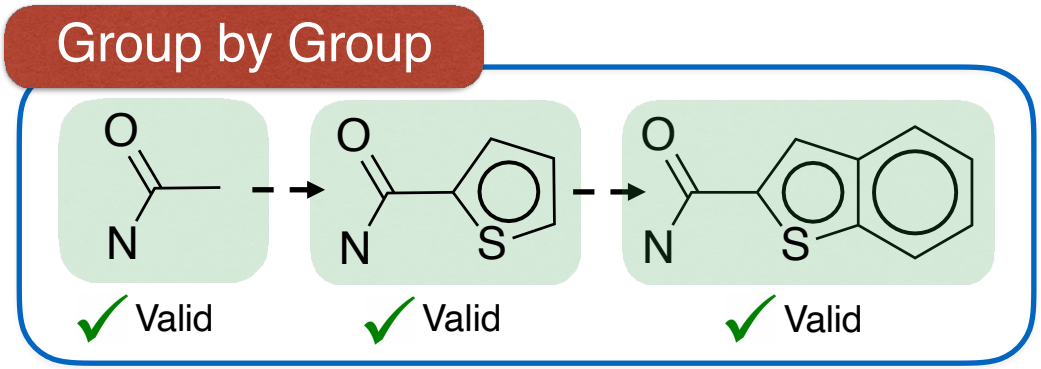
Functional groups





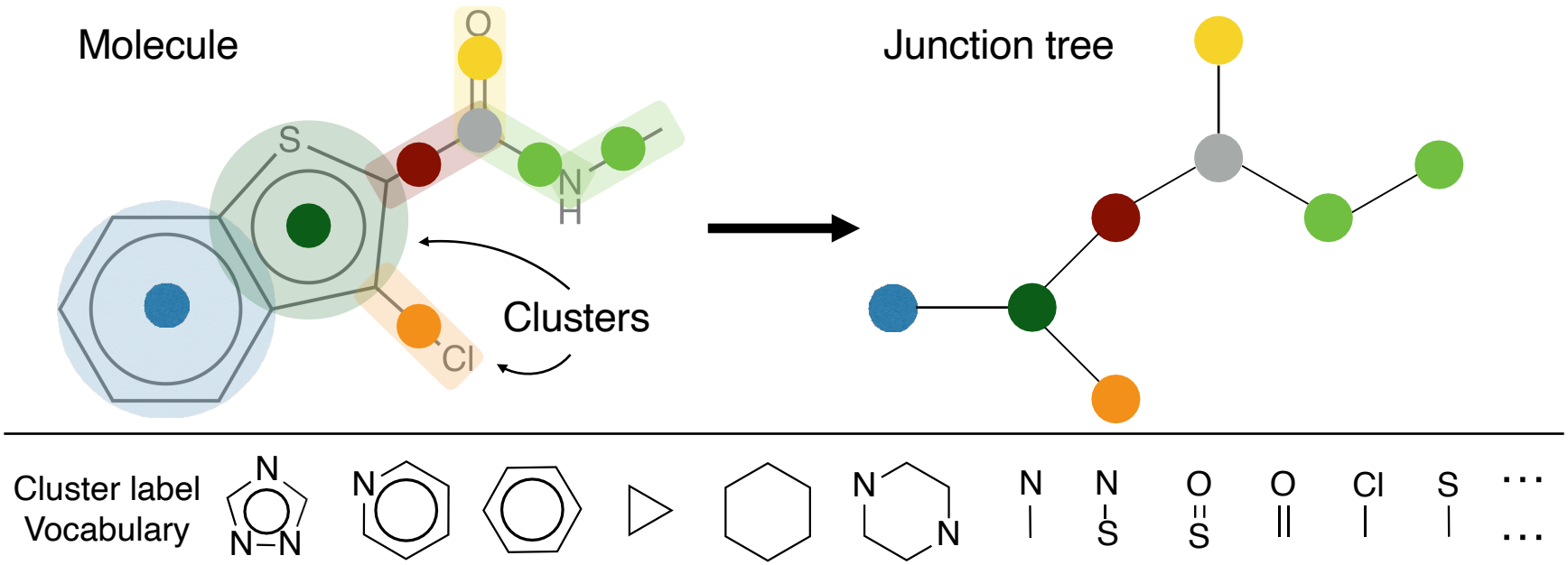
How to generate graphs?





- Shorter action sequence
- Easy to check validity

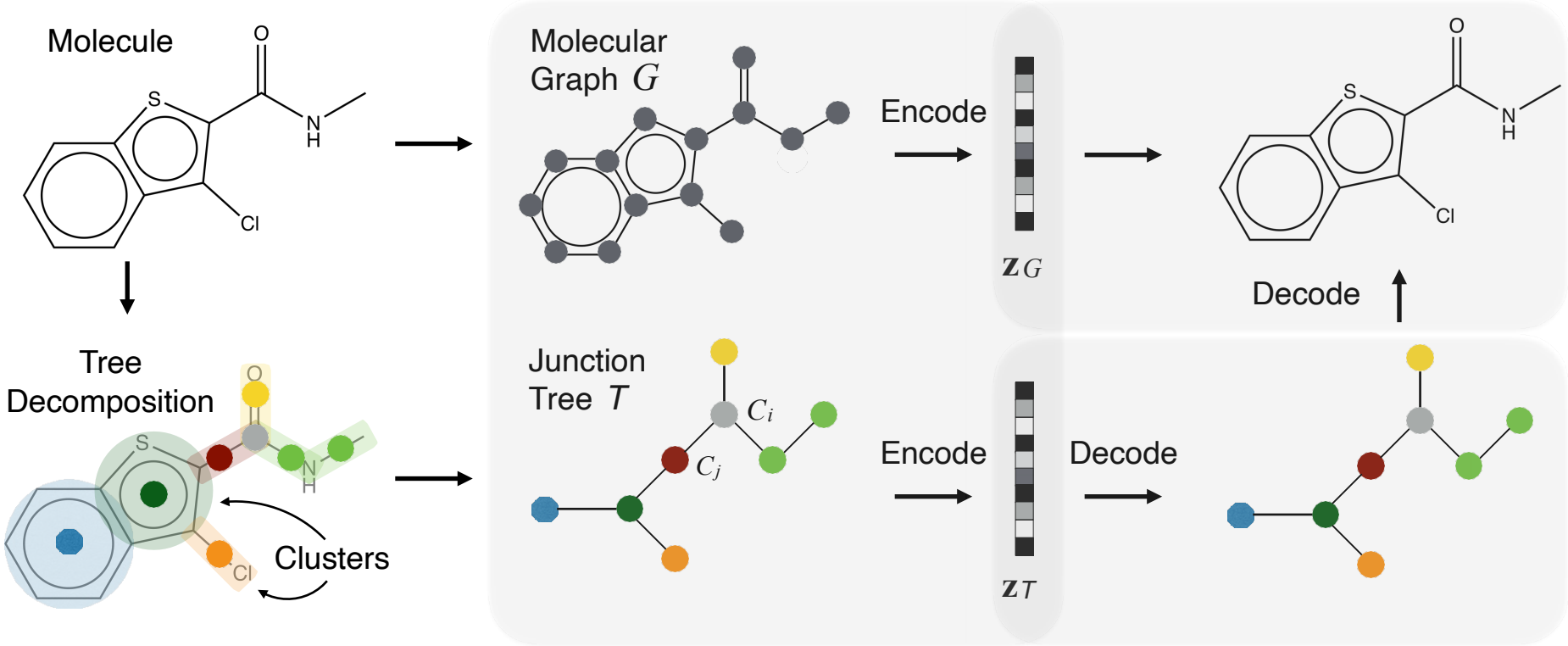
Tree decomposition



- Generate junction tree → Generate graph group by group
- Vocabulary size: less than 800 given 250K molecules

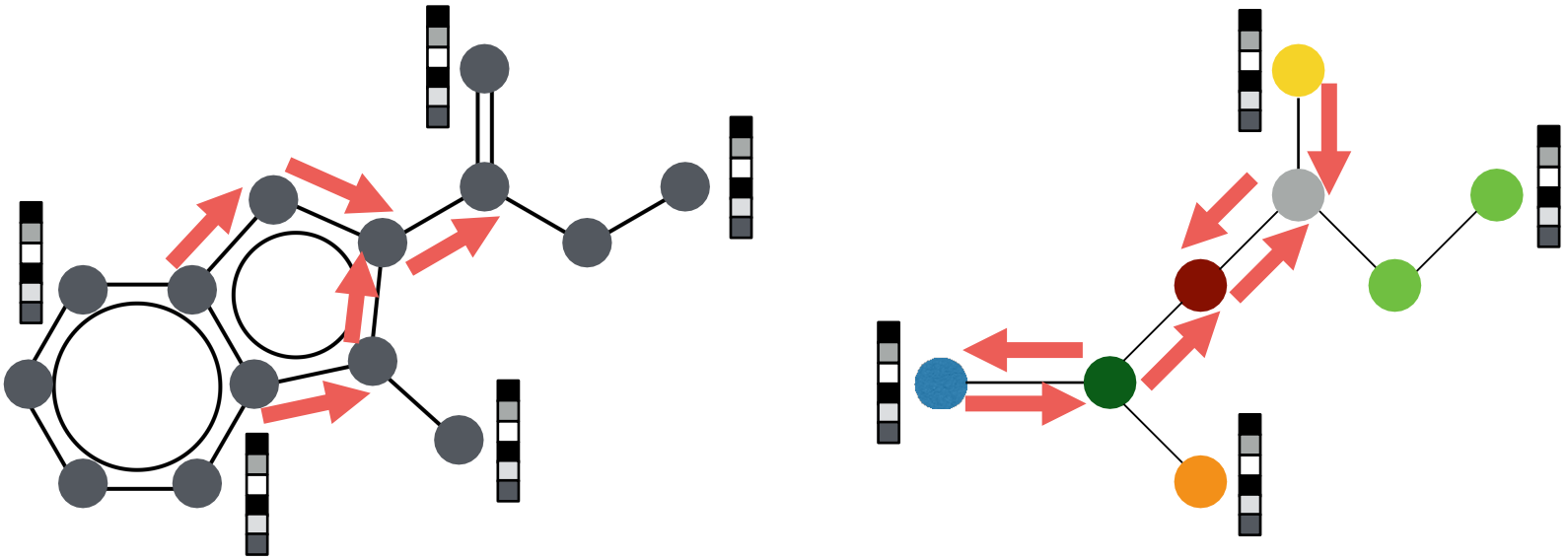
Jin et al., ICML 2018

Approach: Junction-tree variational autoencoder



Jin et al., ICML 2018

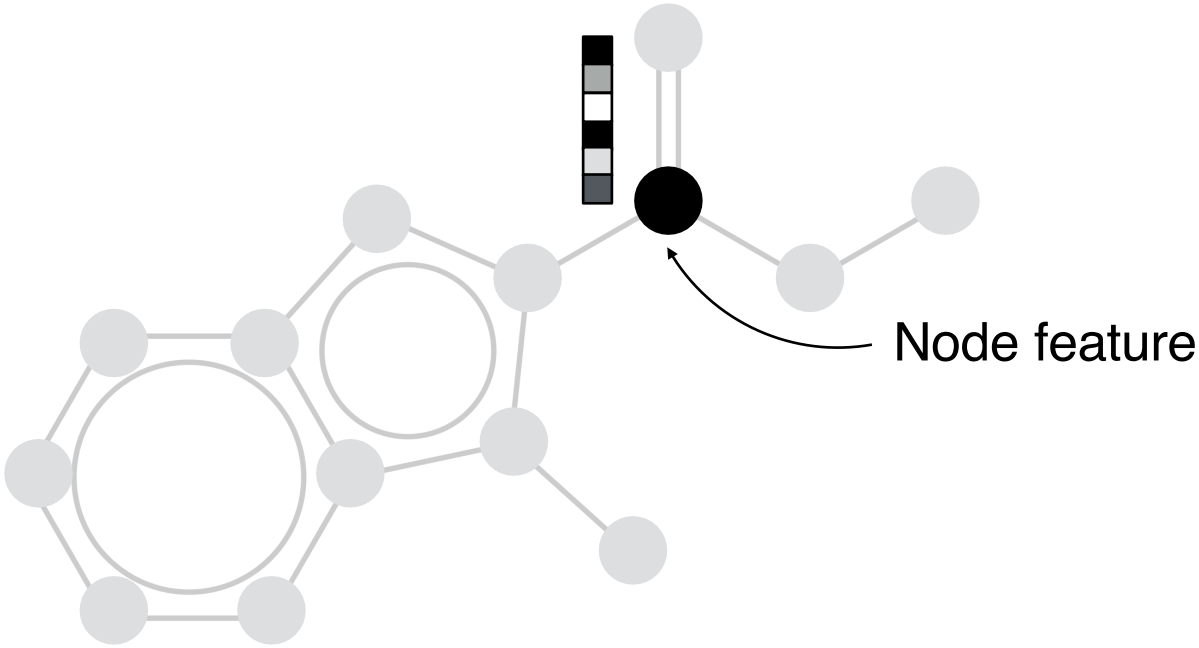
Graph and tree encoders



Neural Message Passing Network (MPN)

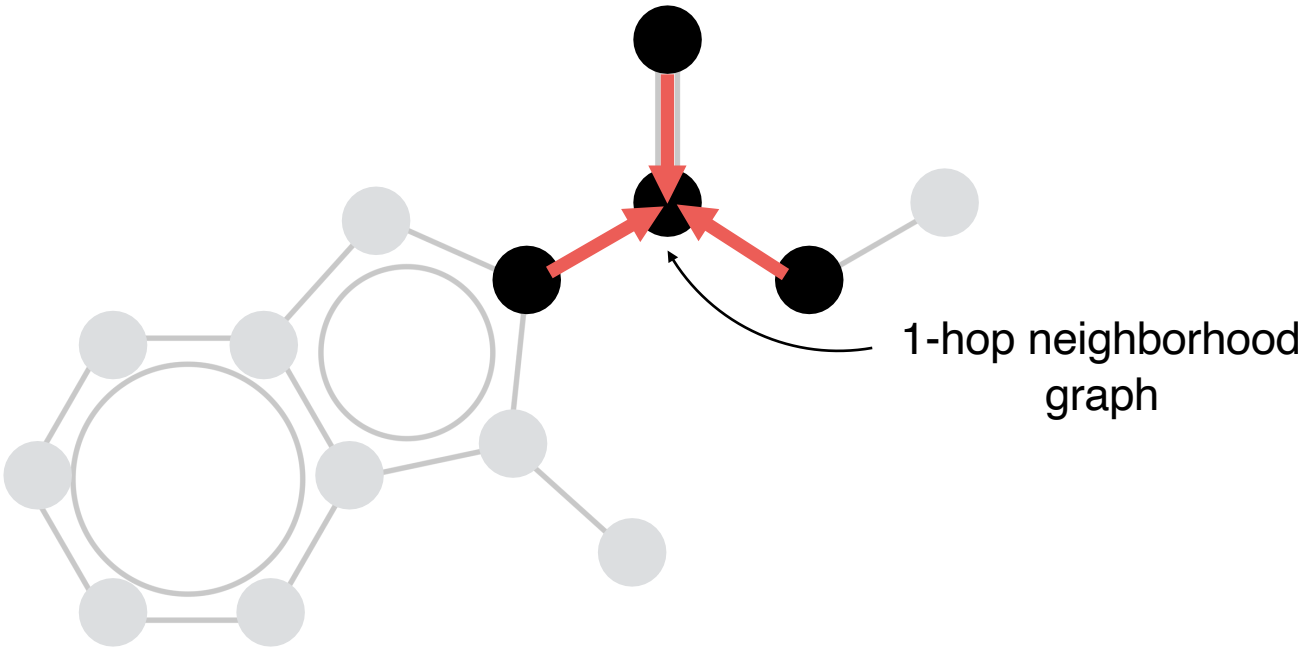
Jin et al., ICML 2018

Graph encoding



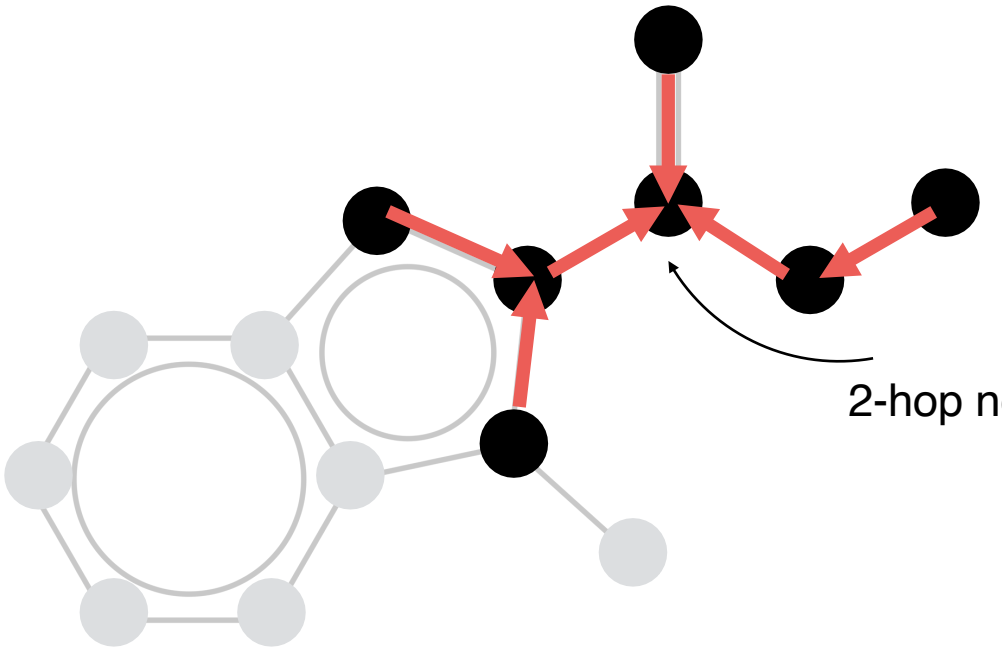
[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

Graph encoding



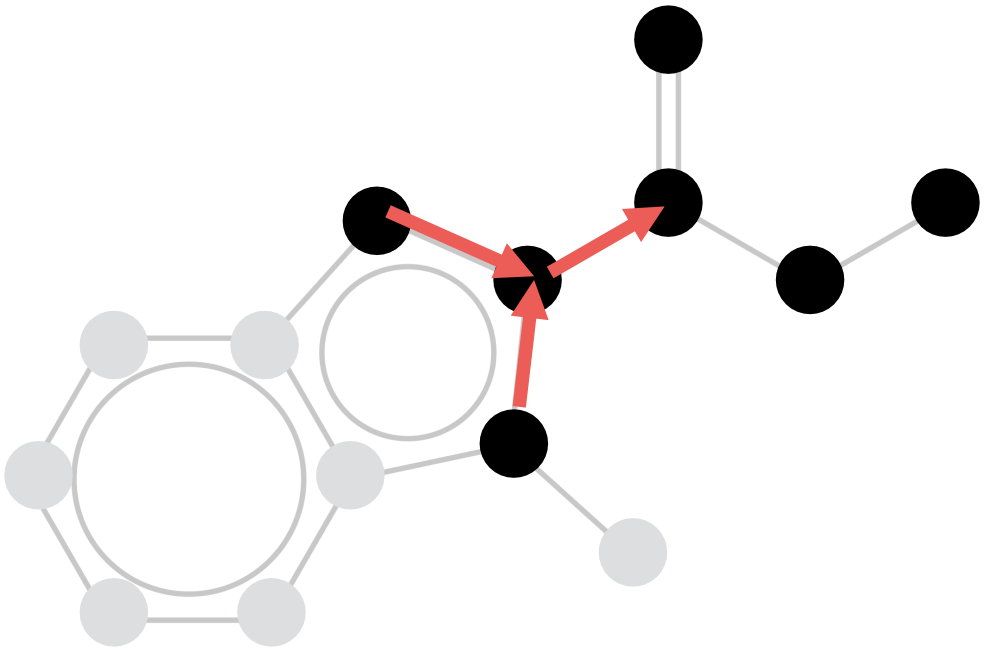
[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

Graph encoding



[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

Graph encoding

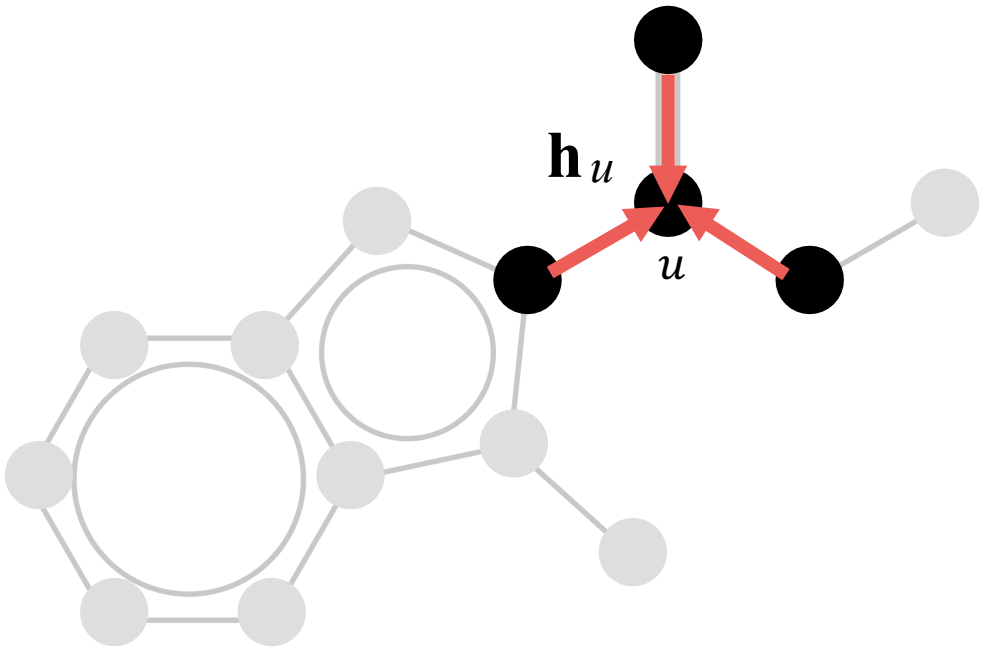


$$\boldsymbol{\nu}_{uv}^{(t)} = \tau(\mathbf{W}_1^g \mathbf{x}_u + \mathbf{W}_2^g \mathbf{x}_{uv} + \mathbf{W}_3^g \sum_{w \in N(u) \setminus v} \boldsymbol{\nu}_{wu}^{(t-1)})$$

Messages | Node feature | Edge feature

[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

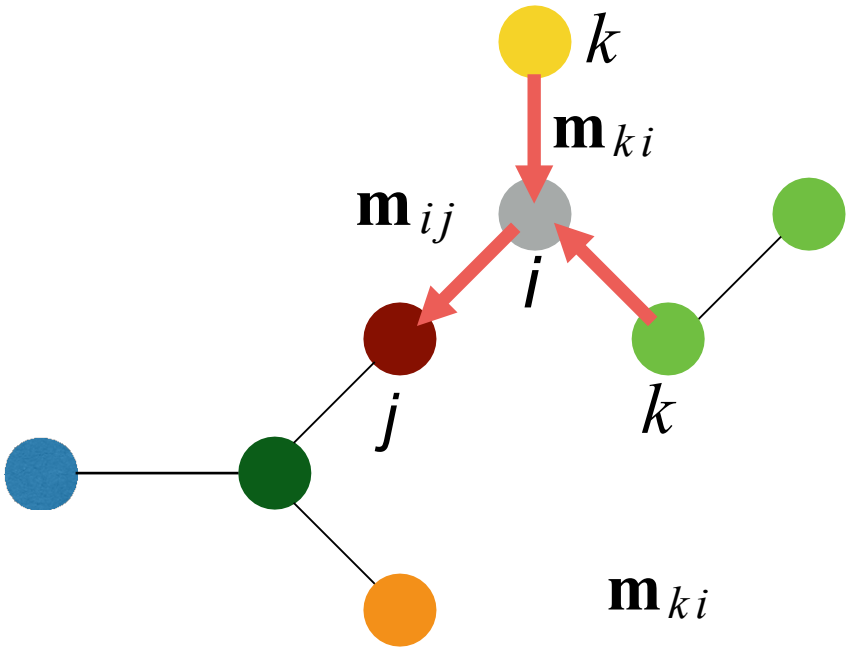
Graph encoding



$$\mathbf{h}_u = \tau(\mathbf{U}_1^g \mathbf{x}_u + \sum\nolimits_{v \in N(u)} \mathbf{U}_2^g \boldsymbol{\nu}_{vu}^{(T)})$$

[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

Tree encoding

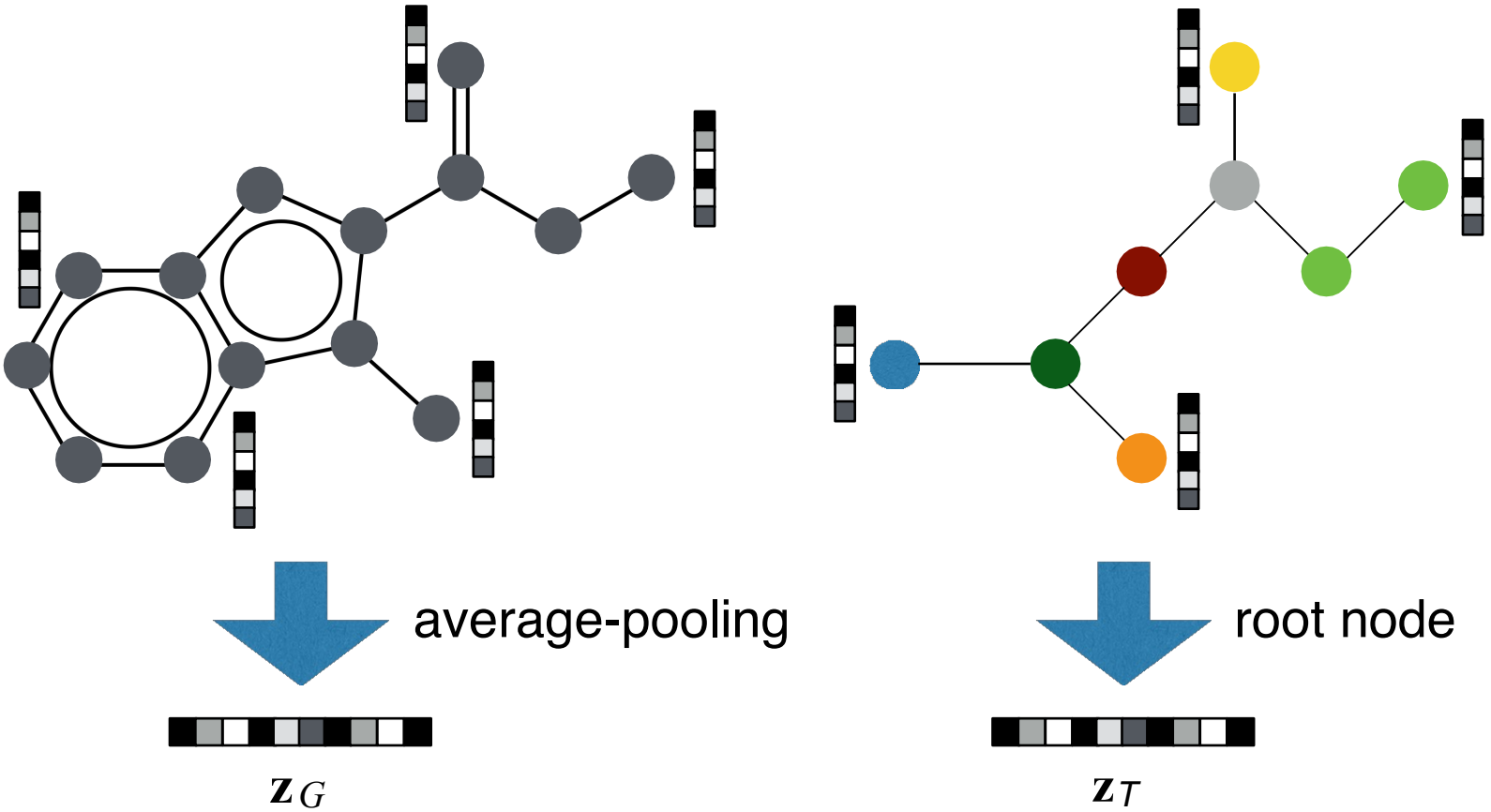


$$\mathbf{m}_{ij} = \text{GRU}(\mathbf{x}_i, \{\mathbf{m}_{ki}\}_{k \in N(i) \setminus j})$$

To capture long range interactions

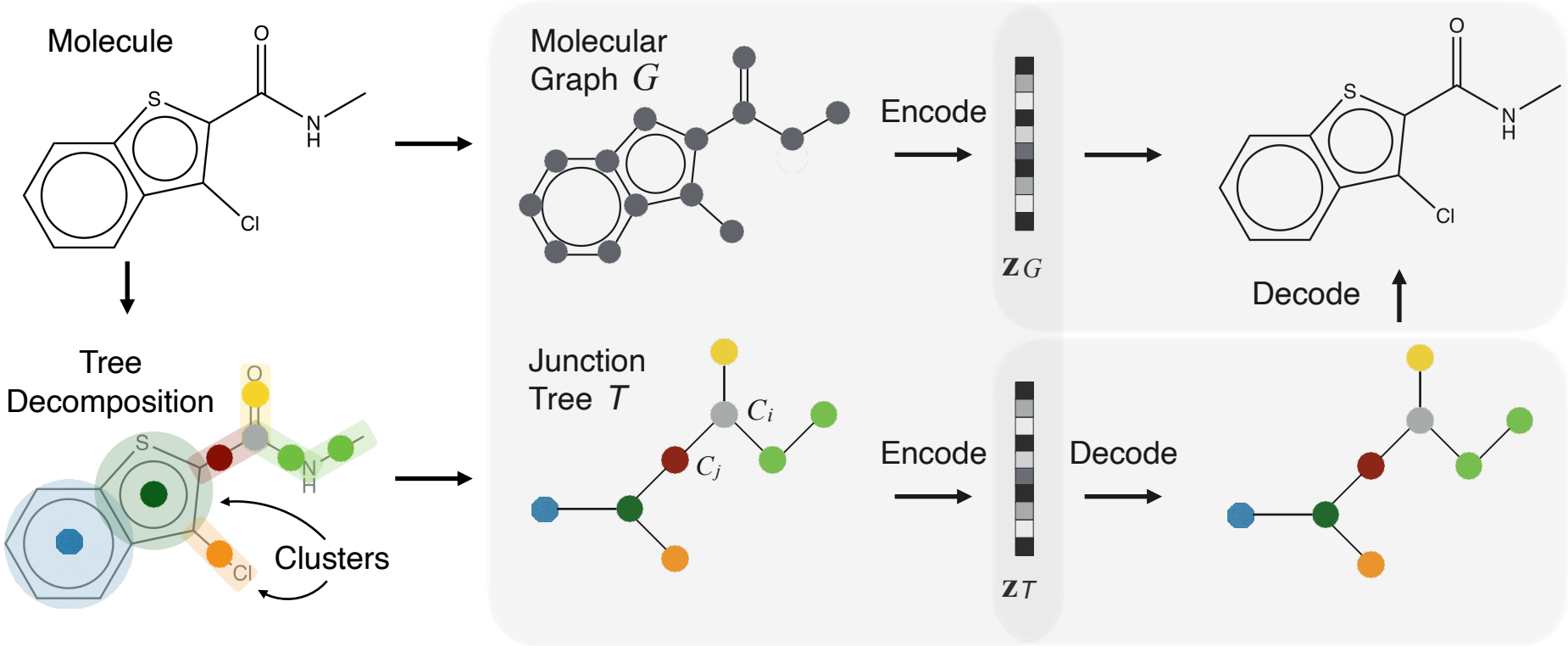
Jin et al., ICML 2018

Graph and tree encoders



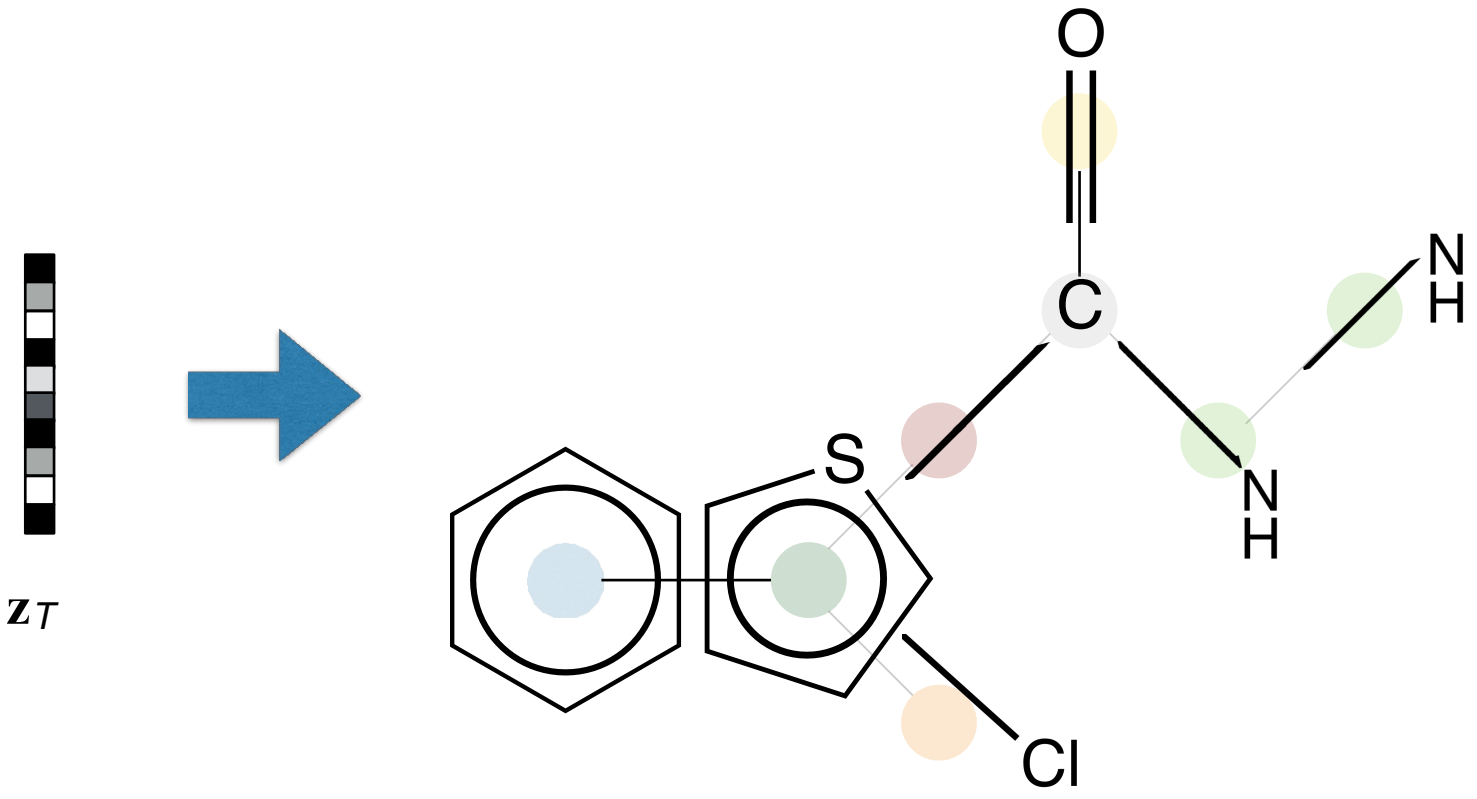
Jin et al., ICML 2018

Approach: Junction-tree variational autoencoder



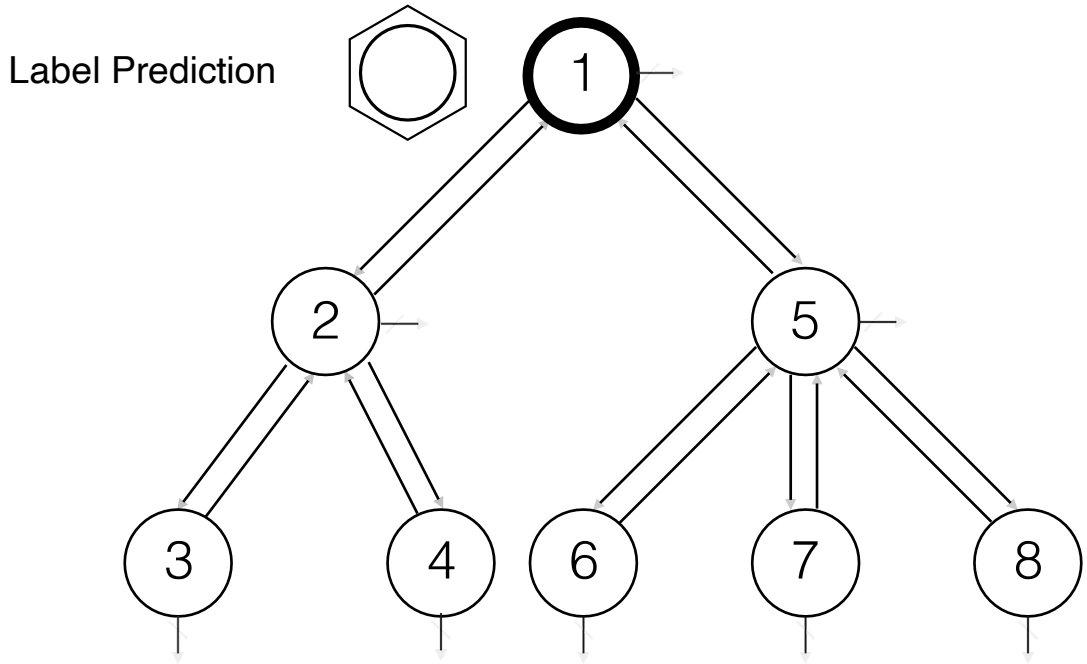
Jin et al., ICML 2018

Tree decoder



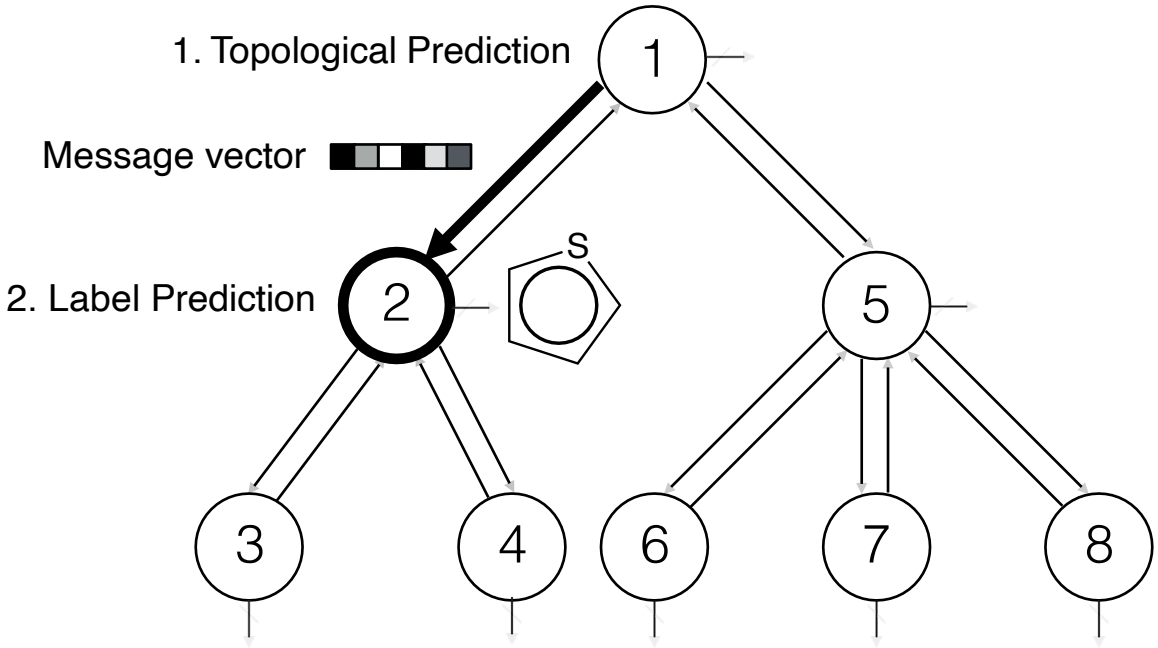
Jin et al., ICML 2018

Tree decoder



[4] Alvarez-Melis & Jaakkola, Tree-structured decoding with doubly-recurrent neural networks

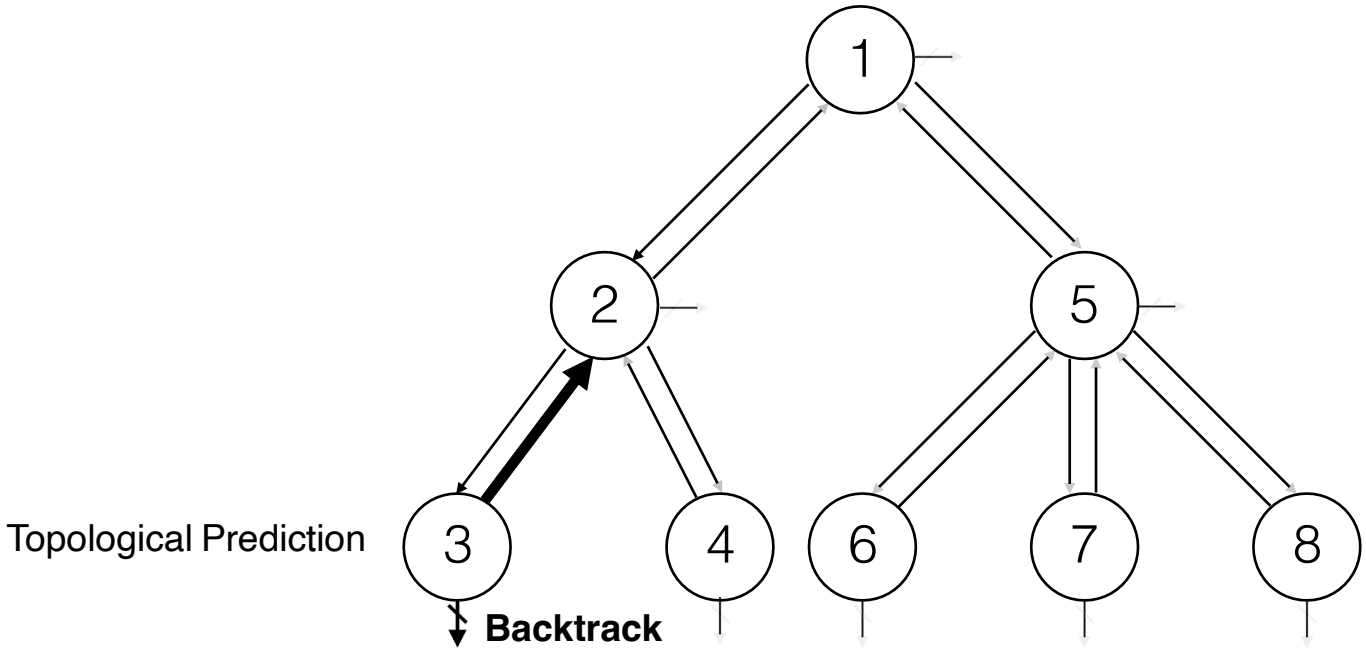
Tree decoder



Topological Prediction: Whether to expand a child or backtrack?

Label Prediction: What is the label of a node?

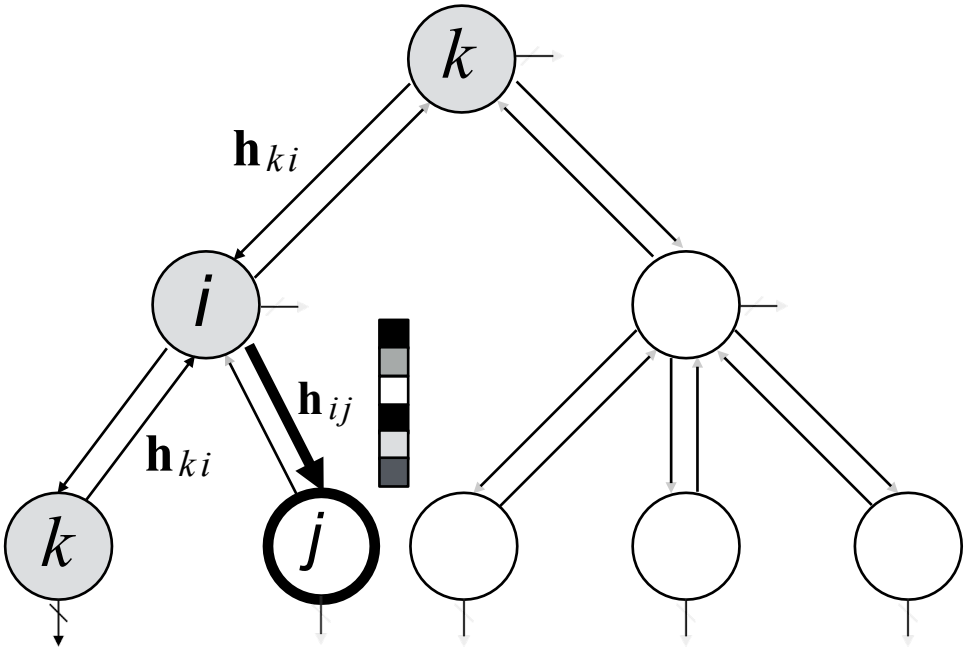
Tree decoder



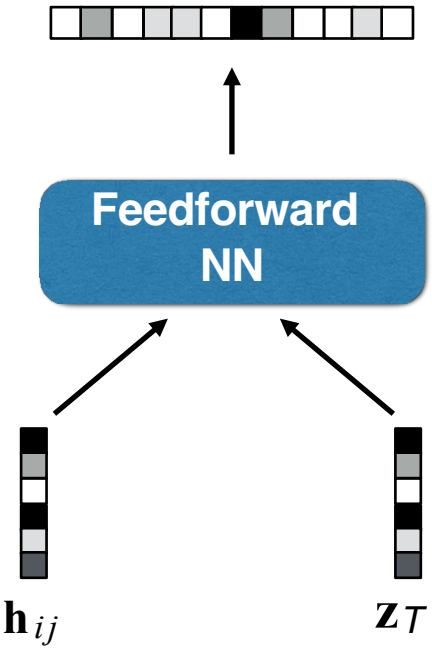
Topological Prediction: Whether to expand a node or backtrack?

Label Prediction: What is the label of a node?

Tree decoder



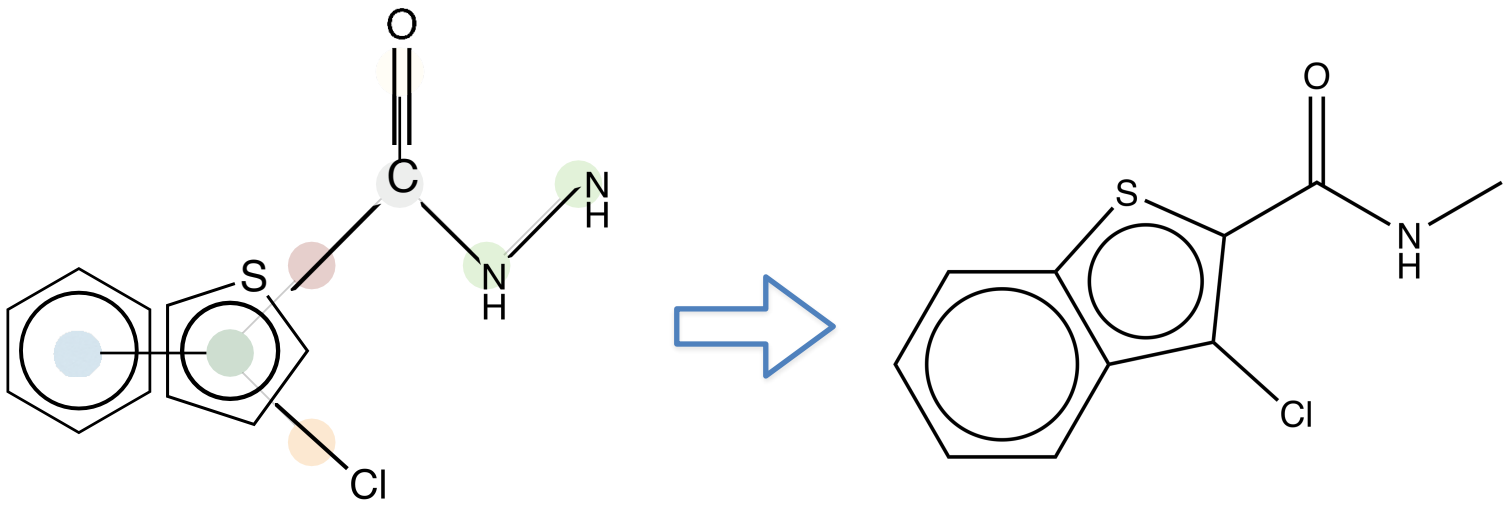
Label Prediction



$$\mathbf{h}_{ij} = \mathrm{GRU}(\mathbf{x}_i, \{\mathbf{h}_{ki}\}_{k \in N_t(i) \setminus j})$$

Encodes the entire subtree of current state

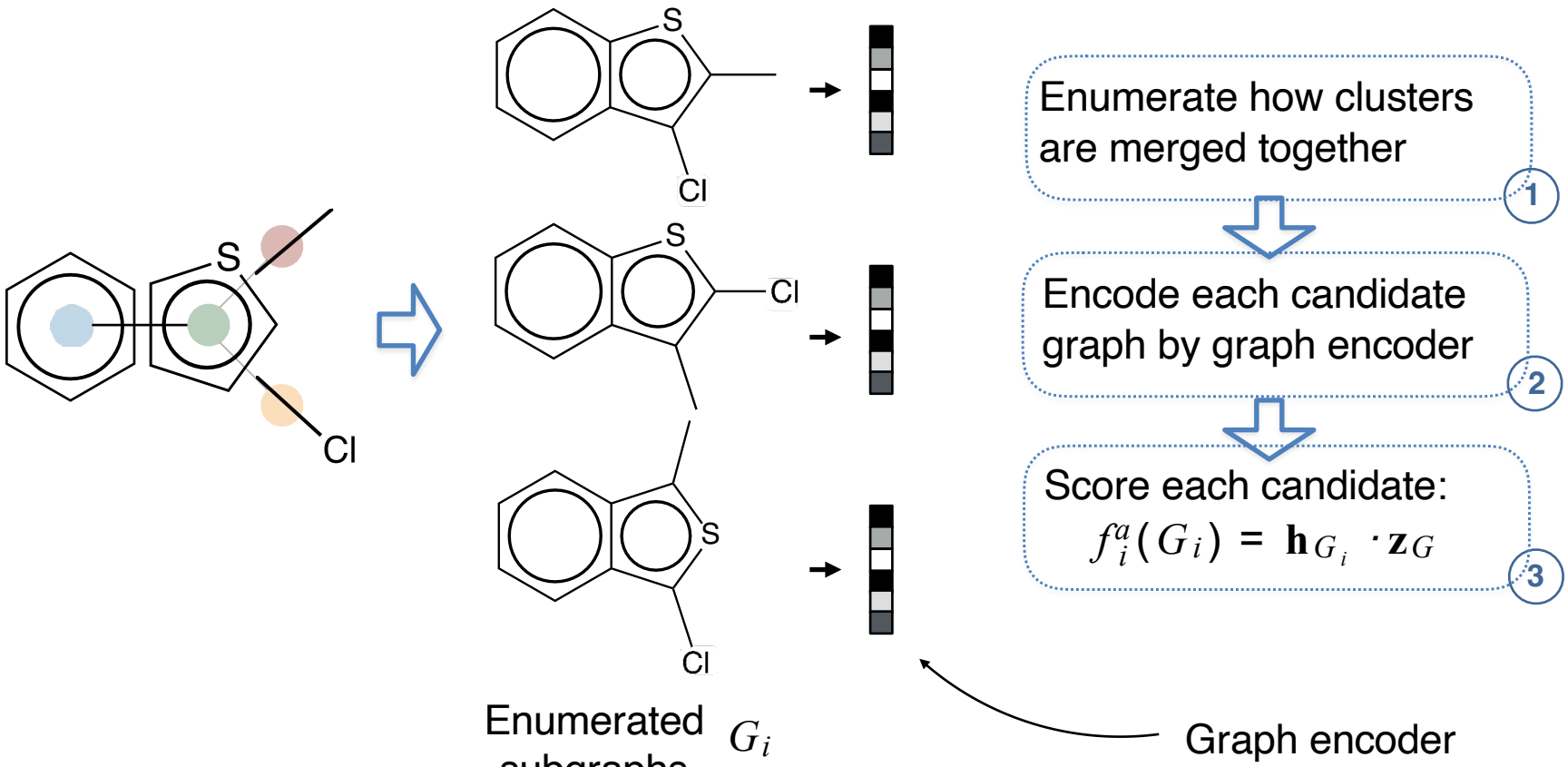
Graph decoder



Predicted Junction Tree

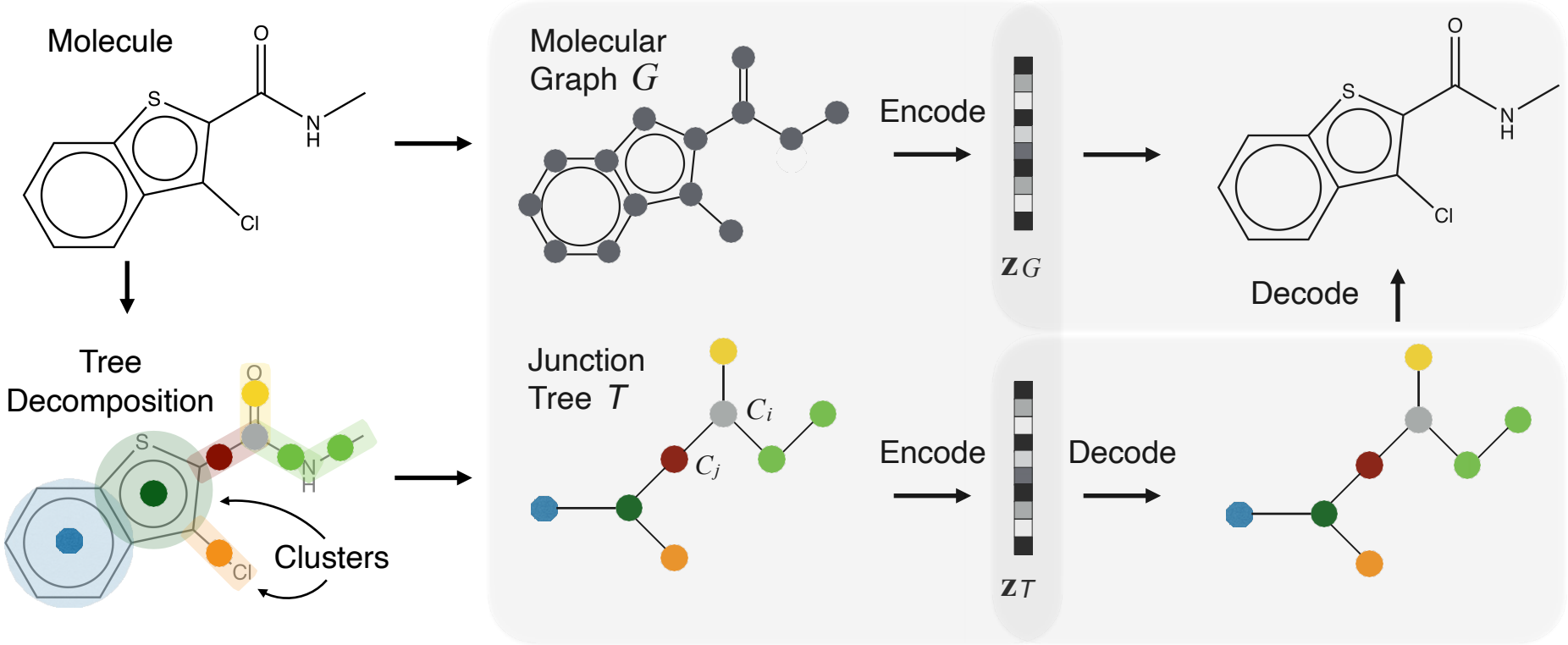
Molecular Graph

Graph decoder



$$\mathcal{L}_g(G) = \sum_i \left[f^a(G_i) - \log \sum_{G_i' \in \mathcal{G}_i} \exp(f^a(G_i')) \right] \quad (16)$$

Recap: Junction-tree variational autoencoder



Jin et al., ICML 2018

Experiments

- **Data**: 250K compounds from ZINC dataset
- **Molecule Generation**: How many molecules are valid when sampled from Gaussian prior?
- **Molecule Optimization**
  - **Global**: Find the best molecule in the entire latent space.
  - **Local**: Modify a molecule to increase its potency

Jin et al., ICML 2018

Baselines

SMILES string based:

1. Grammar VAE (GVAE) (Kusner et al., 2017);

2. Syntax-directed VAE (SD-VAE) (Dai et al., 2018)

Graph based:

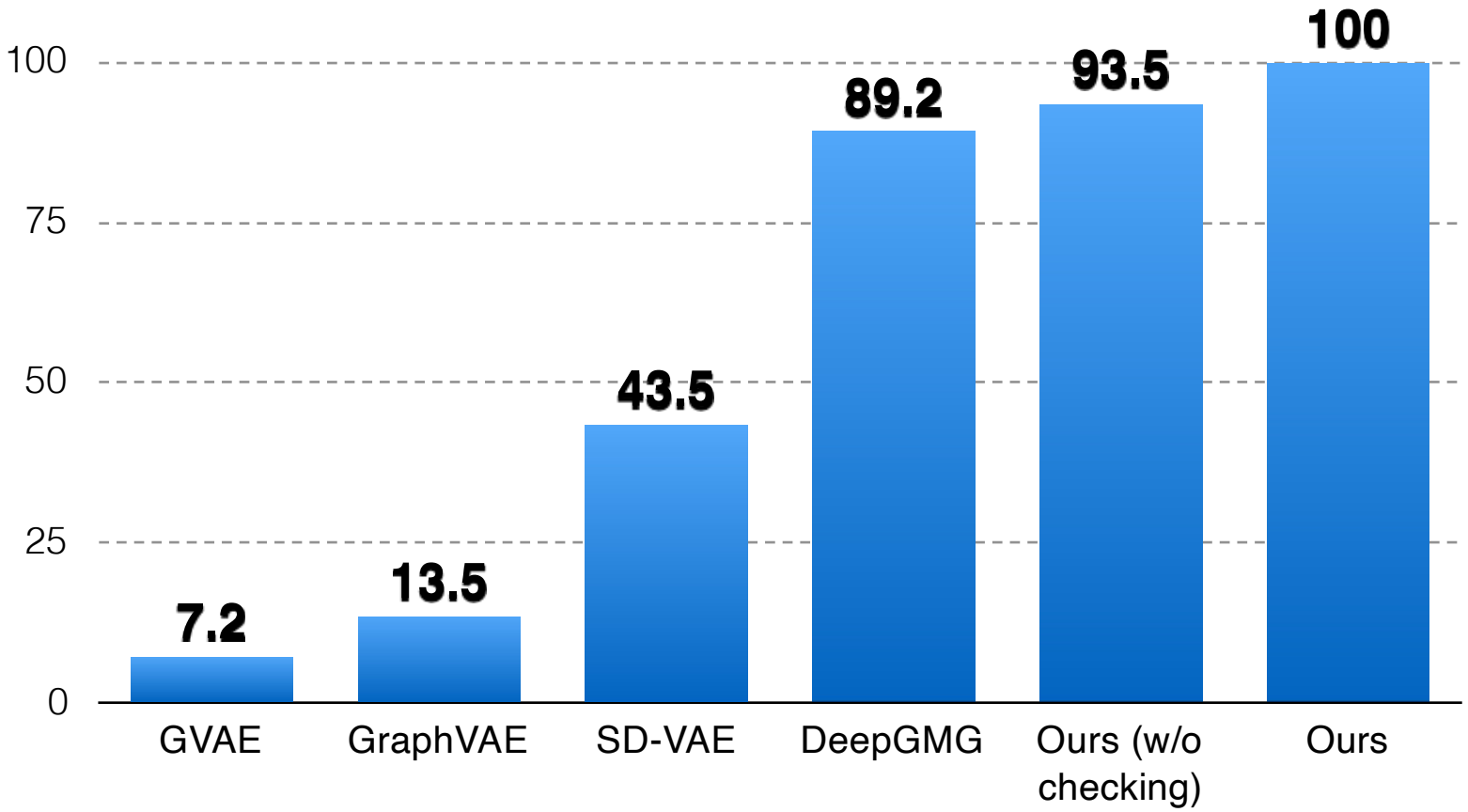
1. Graph VAE (Simonovsky & Komodakis, 2018)

2. DeepGMG (Li et al., 2018)

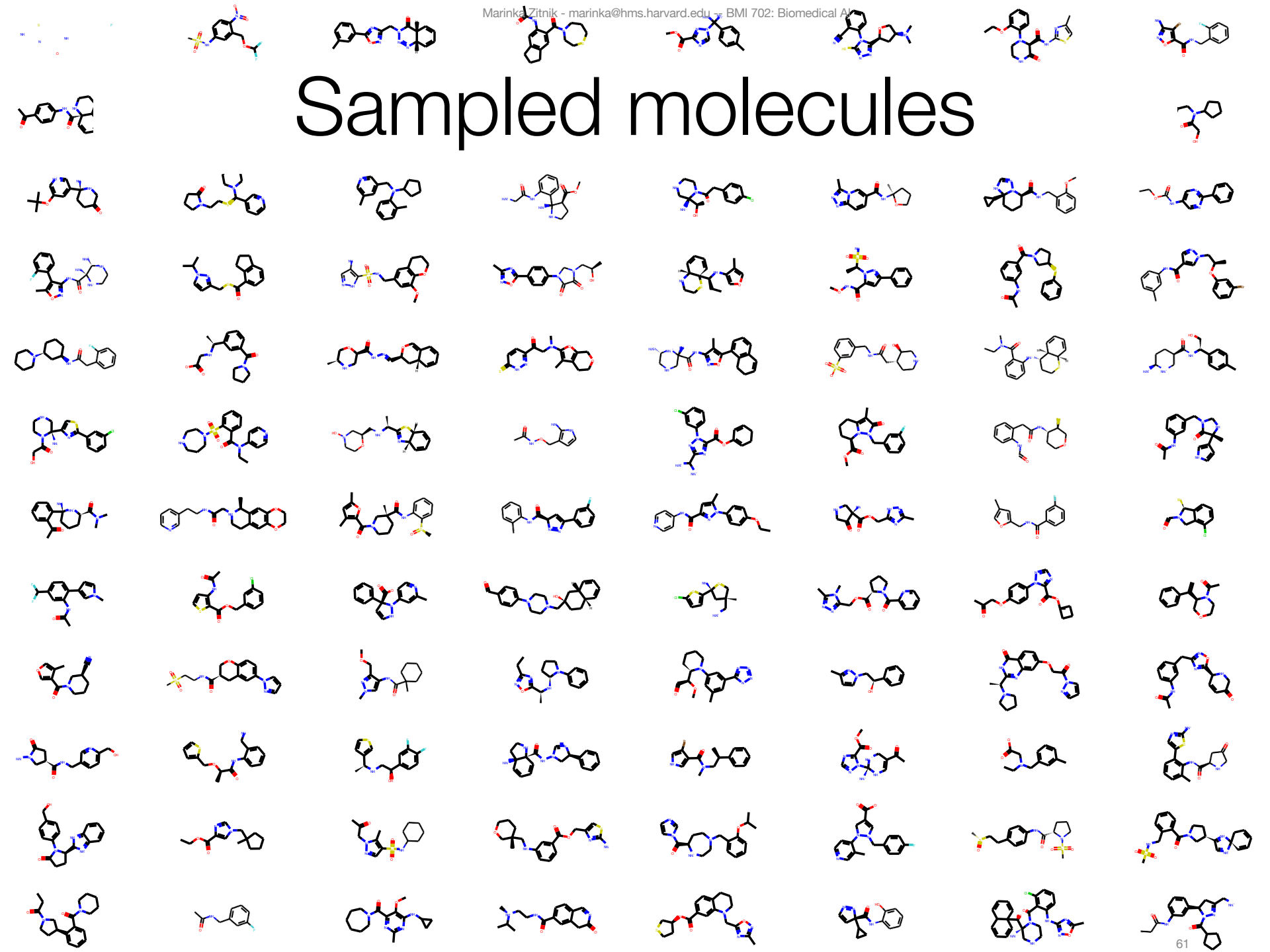
---

[2] Li et al., Learning Deep Generative Models of Graphs, 2018
5 Kusner et al., Grammar Variational Autoencoder, 2017
6 Dai et al., Syntax-directed Variational Autoencoder for structured data, 2018
7 Simonovsky & Komodakis, GraphVAE: Towards generation of small graphs using variational autoencoders

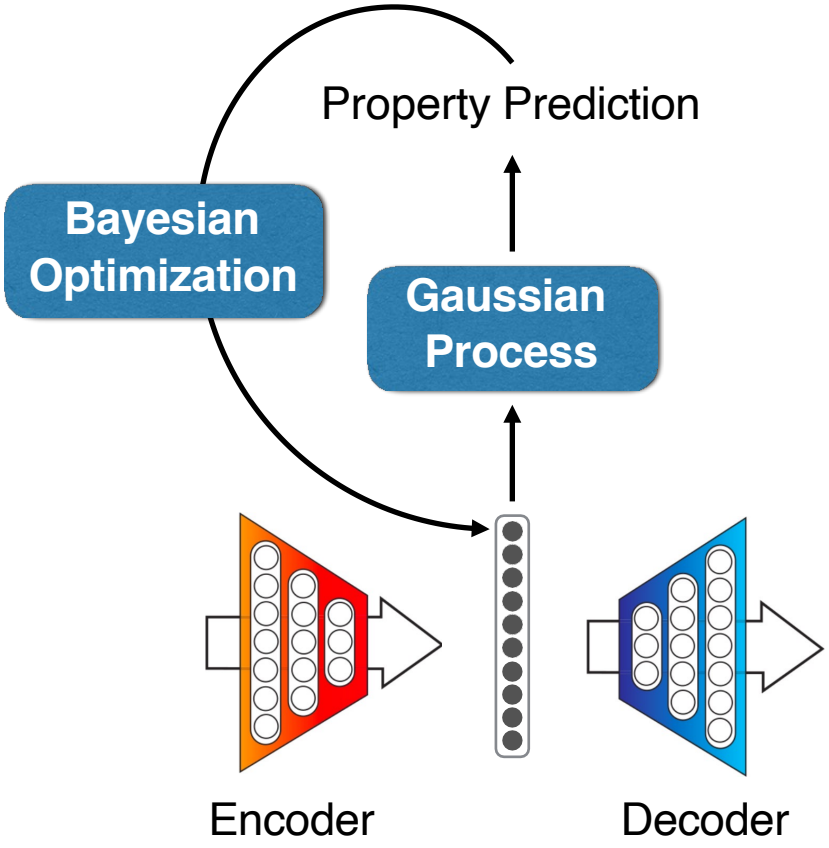
Molecule generation (Validity)

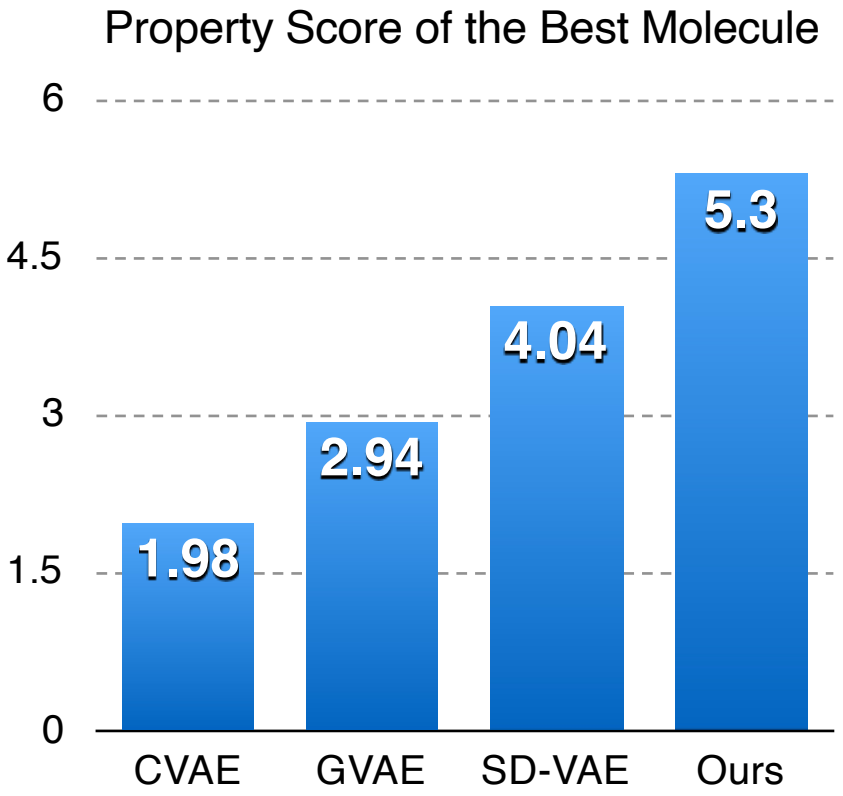


Sampled molecules



Molecule optimization (Global)

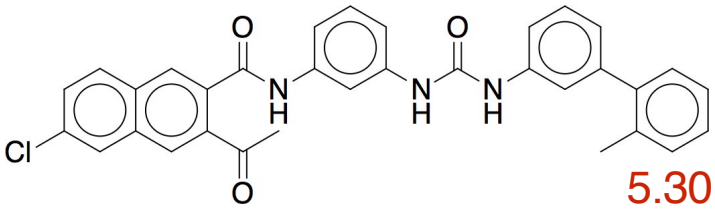


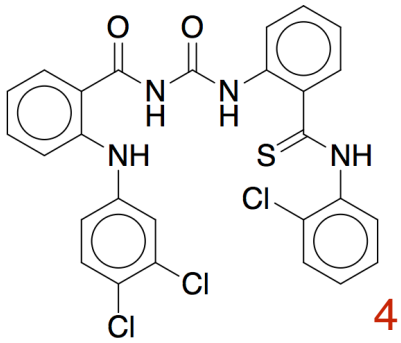


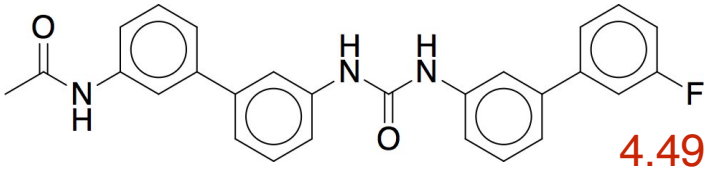
Property: Solubility + Ease of Synthesis

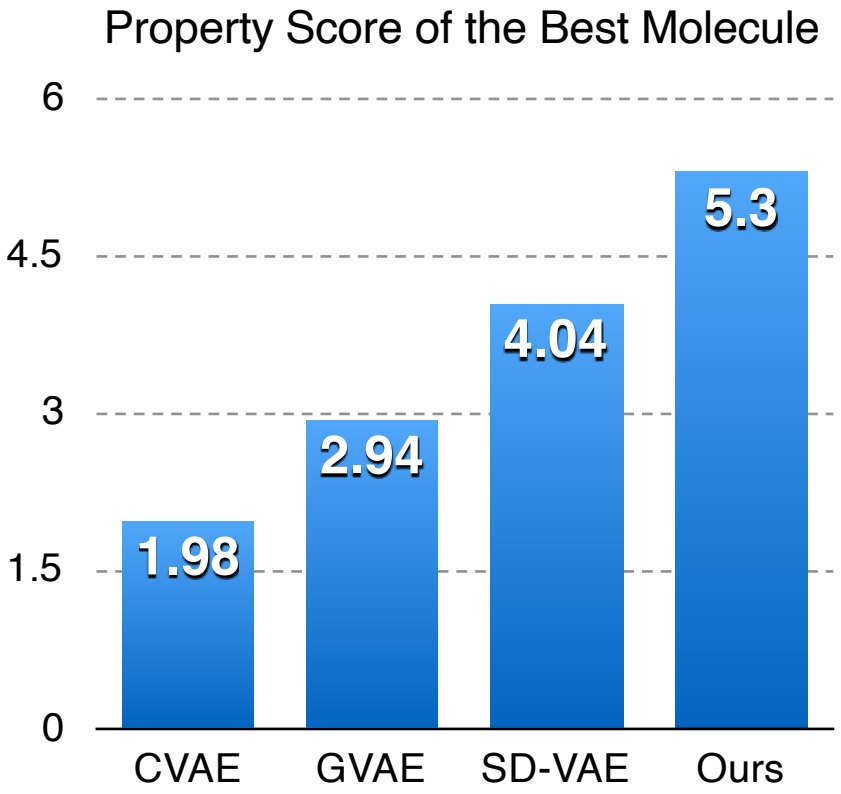
Jin et al., ICML 2018

Molecule optimization (Global)





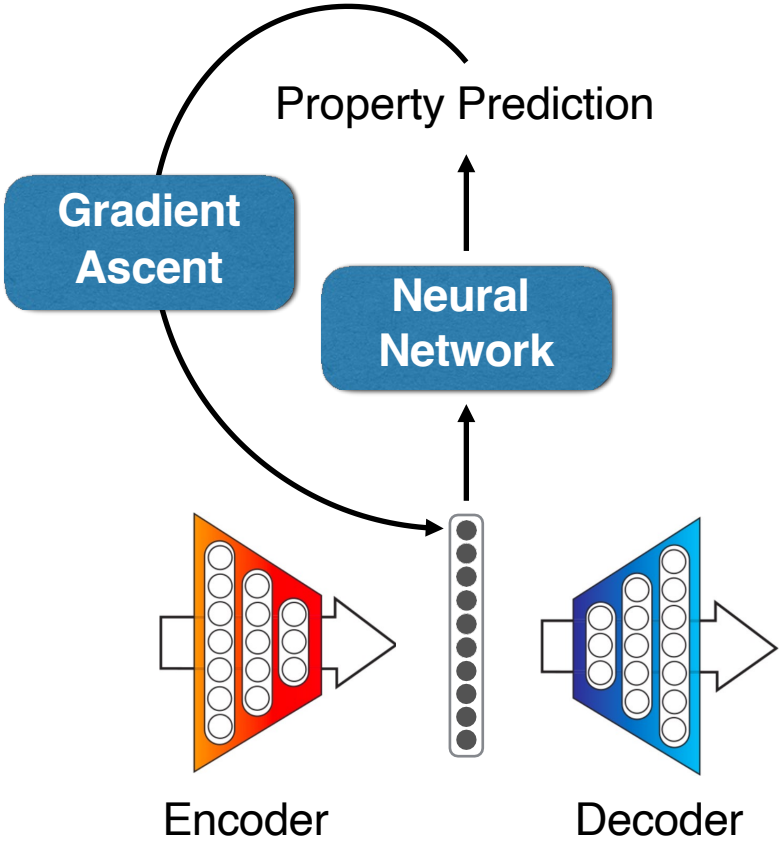


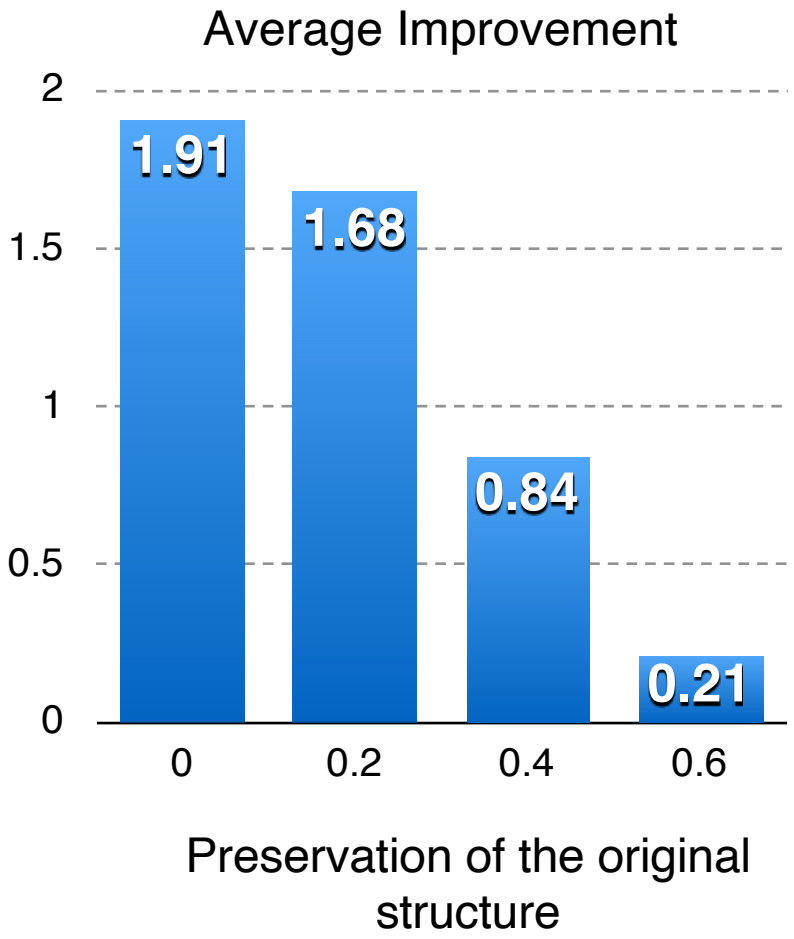


Property: Solubility + Ease of Synthesis

Jin et al., ICML 2018

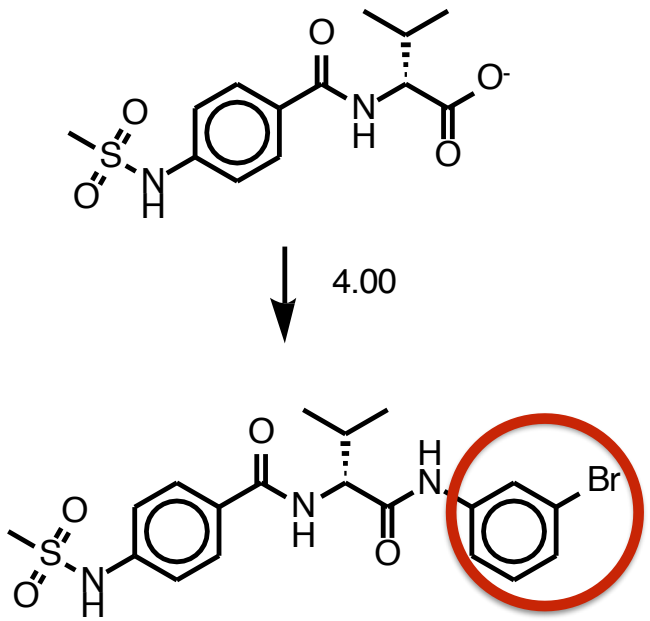
Molecule optimization (Local)



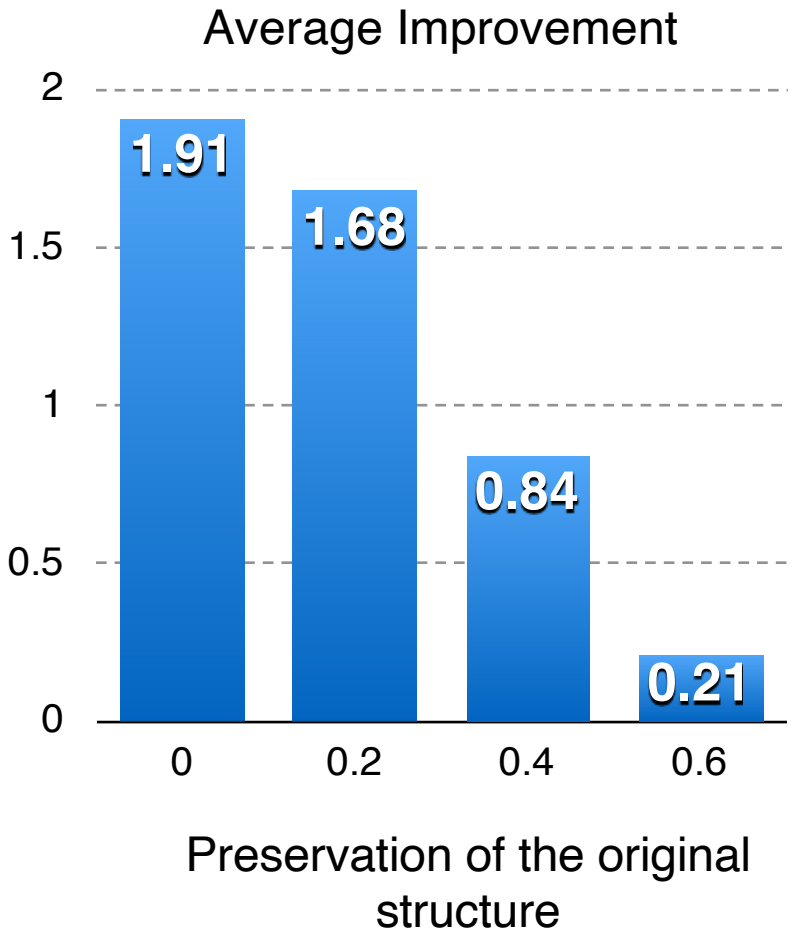


Jin et al., ICML 2018

Molecule optimization (Local)

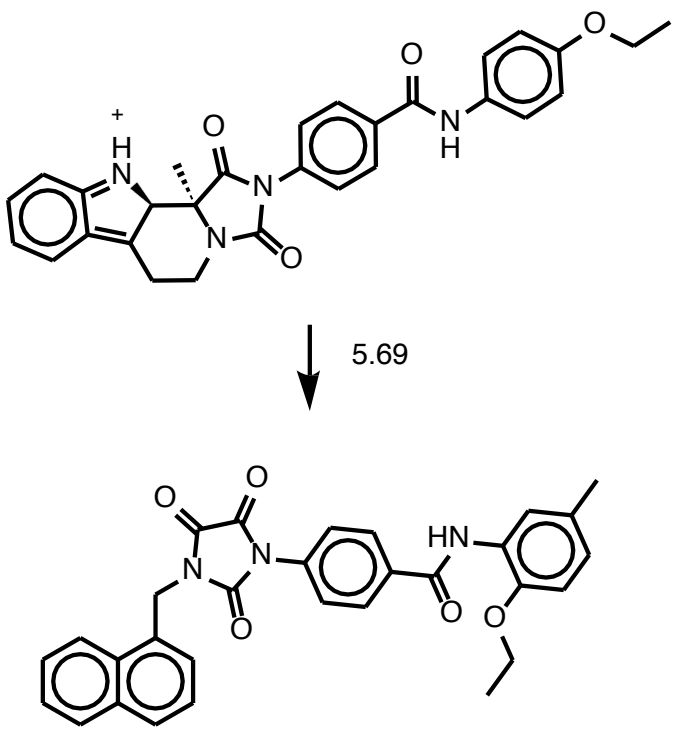


Preservation ≈ 0.6

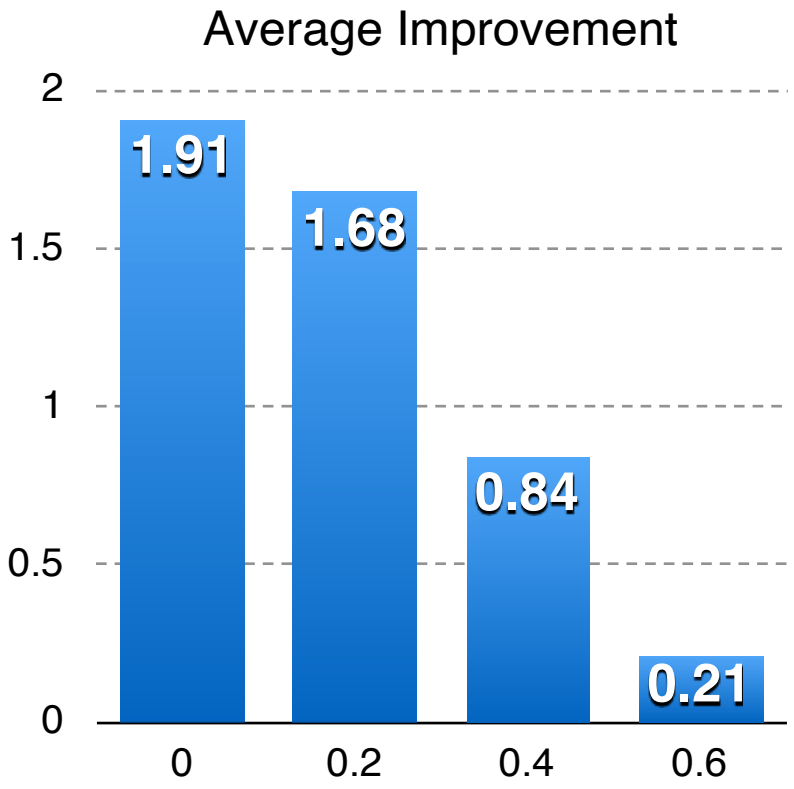


Jin et al., ICML 2018

Molecule optimization (Local)



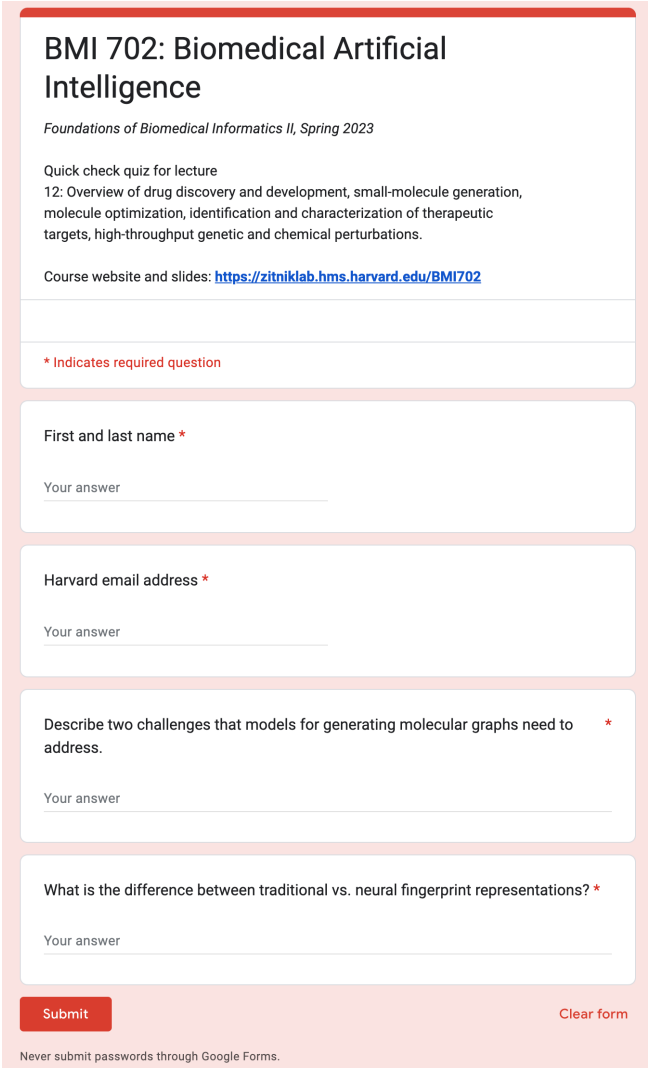
Preservation ≈ 0.4



Jin et al., ICML 2018

Quick Check

https://forms.gle/4DJYidWsL4KkDau57



BMI 702: Biomedical Artificial Intelligence

Foundations of Biomedical Informatics II, Spring 2023

Quick check quiz for lecture
12: Overview of drug discovery and development, small-molecule generation, molecule optimization, identification and characterization of therapeutic targets, high-throughput genetic and chemical perturbations.

Course website and slides: https://zitniklab.hms.harvard.edu/BMI702

* Indicates required question

First and last name *

Your answer

Harvard email address *

Your answer

Describe two challenges that models for generating molecular graphs need to address. *

Your answer

What is the difference between traditional vs. neural fingerprint representations? *

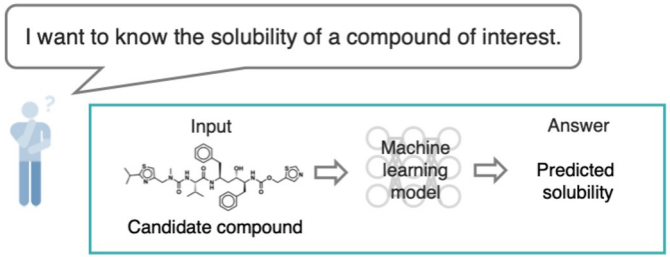
Your answer

Submit Clear form

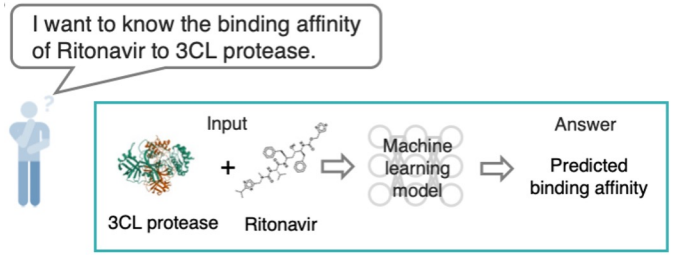
Never submit passwords through Google Forms.

Outline for today's class

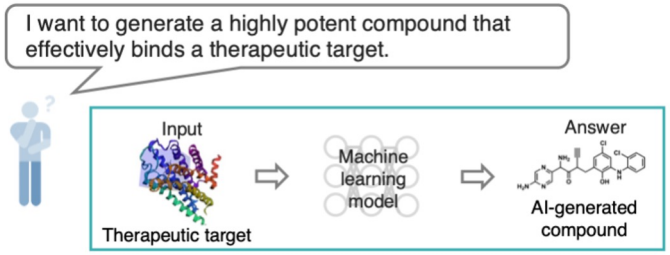
- Optimization & generation of small molecules



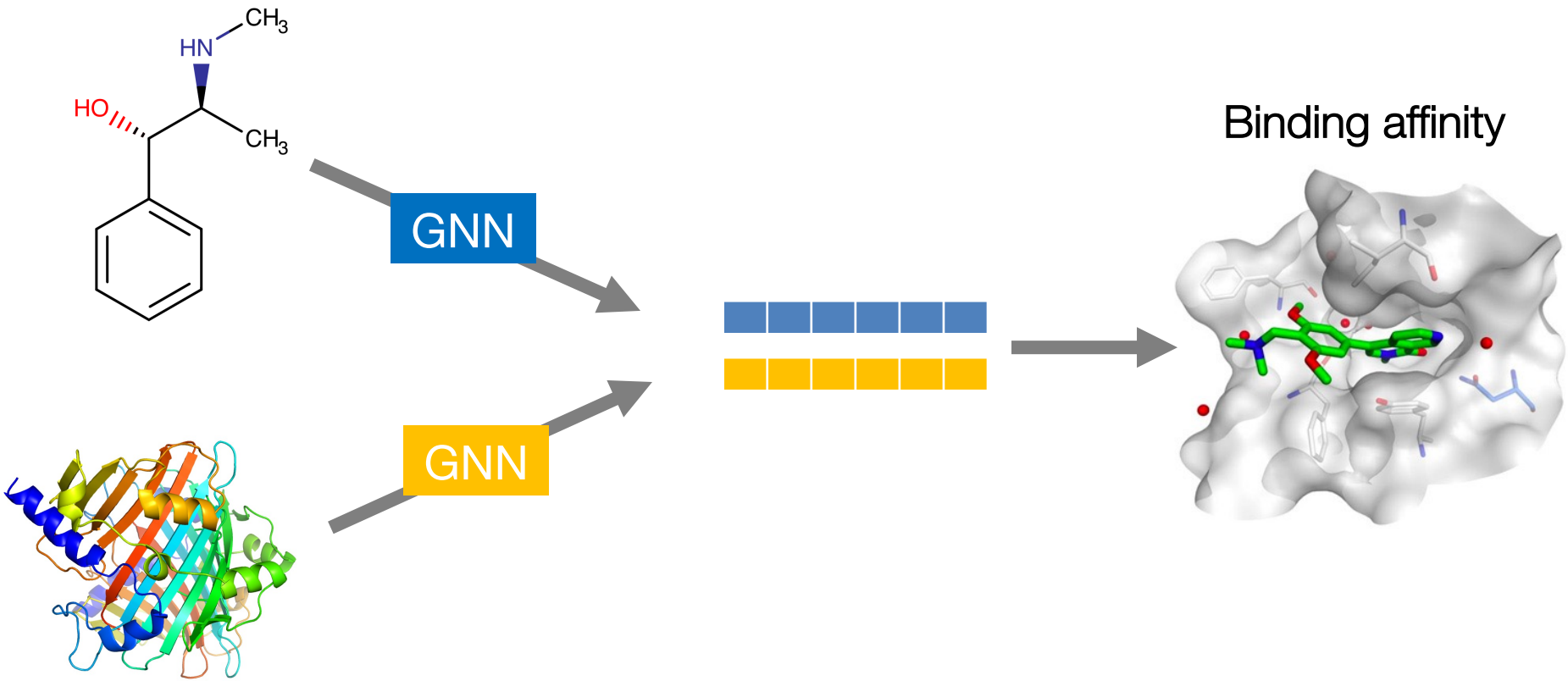
- Binding of drugs to therapeutic targets



- High-throughput genetic & chemical perturbations

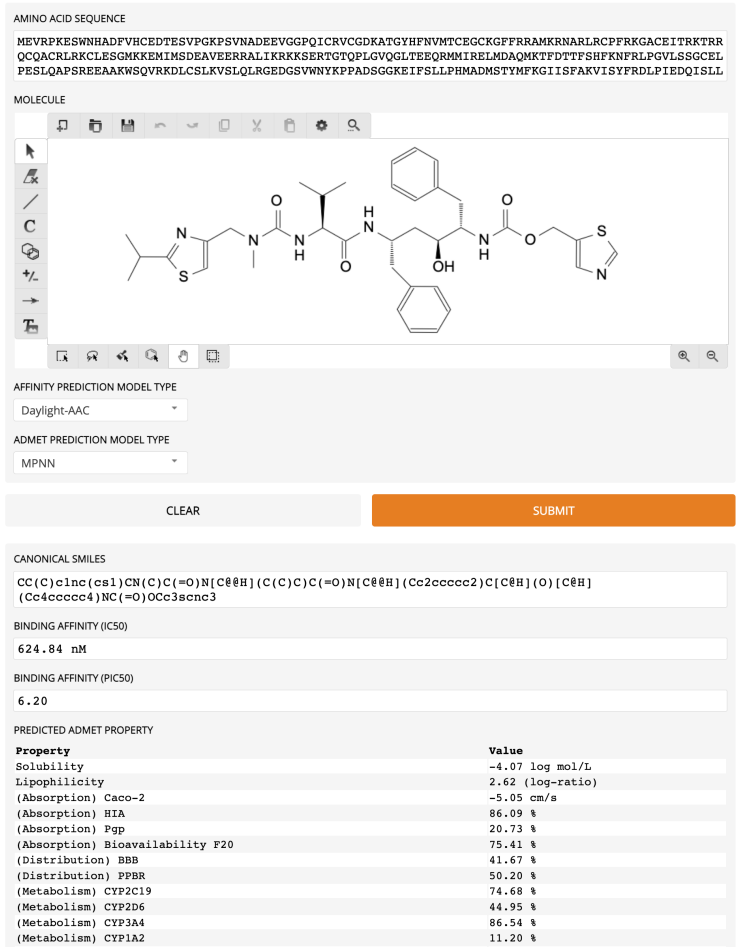


Geometric modeling of binding



Therapeutics Data Commons: Machine Learning Datasets and Tasks for Therapeutics, NeurIPS, 2021
Artificial Intelligence Foundation for Therapeutic Science, *Nature Chemical Biology,* 2022

Results: Binding affinity prediction



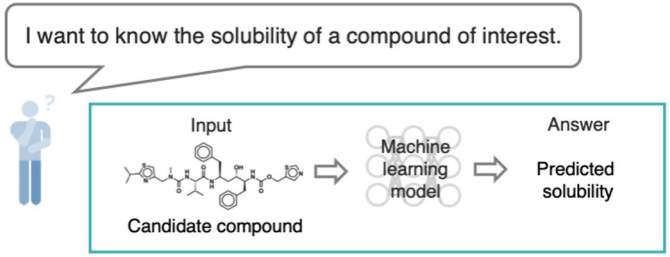
- **ERM** is a standard strategy to minimize errors across all domains
- **MMD** minimizes maximum mean discrepancy across domains
- **CORAL** matches mean and covariance of features across domains
- **IRM** optimizes features using a cross-domain optimized linear classifier
- **GroupDRO** optimizes ERM and adjusts weights of domains with larger errors
- **Marginal transfer learning** augments features with marginal distributions
- **ANDMask** masks gradients that have inconsistent signs in the corresponding weights across domains

Modern data management
Human-AI collaboration

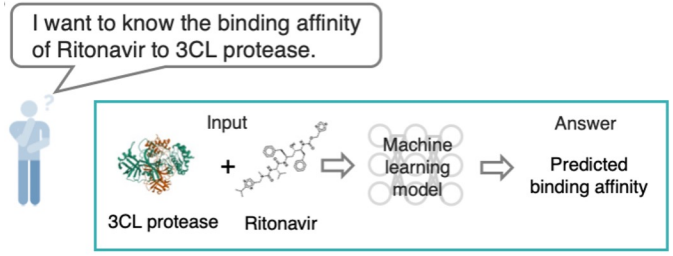
Therapeutics Data Commons: Machine Learning Datasets and Tasks for Therapeutics, NeurIPS, 2021
Artificial Intelligence Foundation for Therapeutic Science, *Nature Chemical Biology,* 2022

Outline for today's class

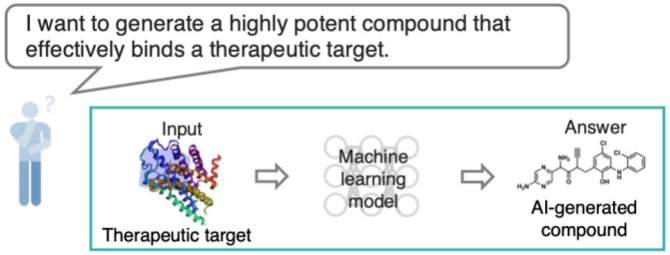
- Optimization & generation of small molecules



- Binding of drugs to therapeutic targets



- High-throughput genetic & chemical perturbations



Words and genes share a correspondence:
their **meanings** arise from their **context**.

Gene perturbation measurements across diverse cell contexts
induce **semantics for genes**

(under the right approach)

“apple” is a **polysemic** word…

… whose **particular meaning** is resolved via **sentence context**.

Google

grow an apple

grow an apple **tree**

grow an apple **tree from seed**

grow an apple **tree in a pot**

grow an apple **tree indoors**

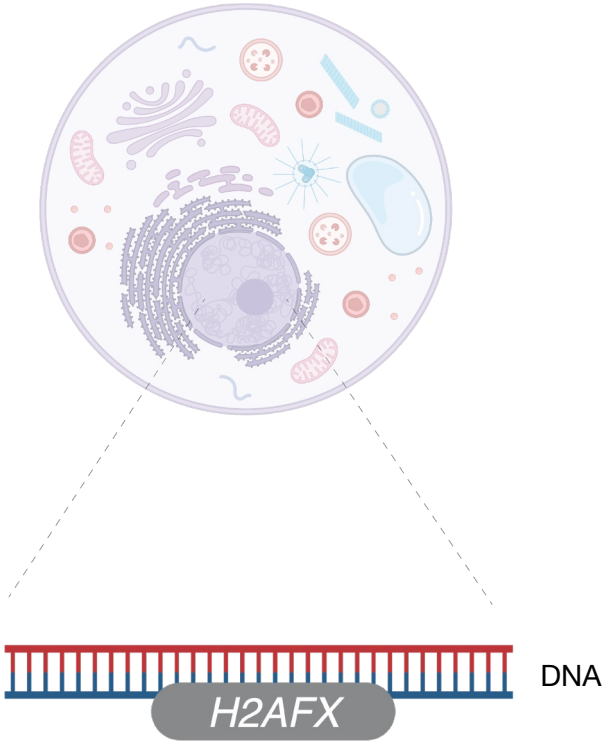
buy an apple

buy an apple **watch**

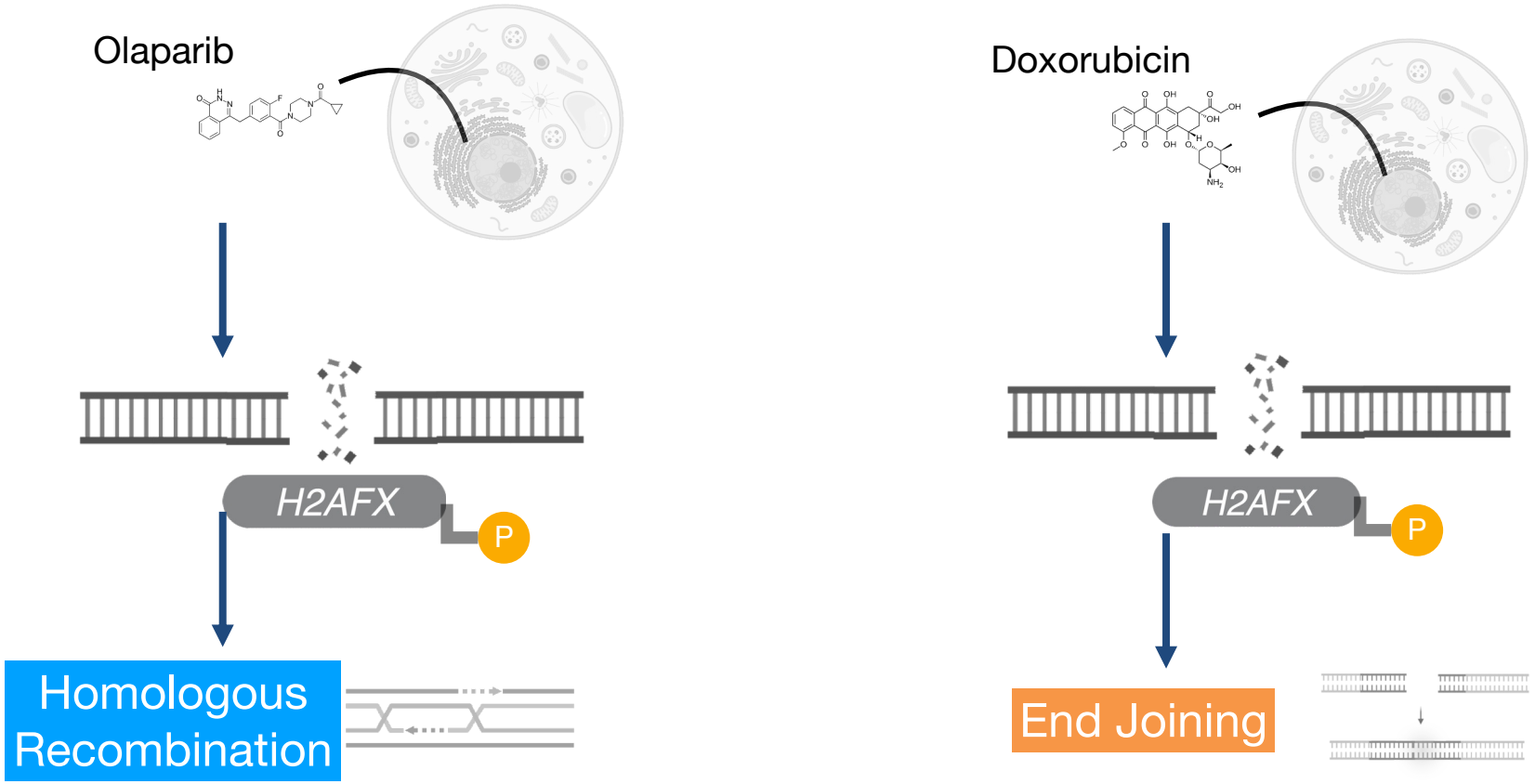
buy an apple **gift card**

buy an apple **tv**

H2AFX is a **pleiotropic** gene…



… whose **particular function** is resolved via **cell context.**



While unsupervised learning of word polysemy is **common**…

Data: corpus of sentence contexts

Approach: word embeddings w/ linear semantics

king - man + woman ≈ queen

unsupervised learning of gene pleiotropy is **unsolved**

Data: **?**

Approach: **?**

geneA - func1 + func2 ≈ geneB

Our goal for today

Unsupervised learning of gene pleiotropy with applications to therapeutic science

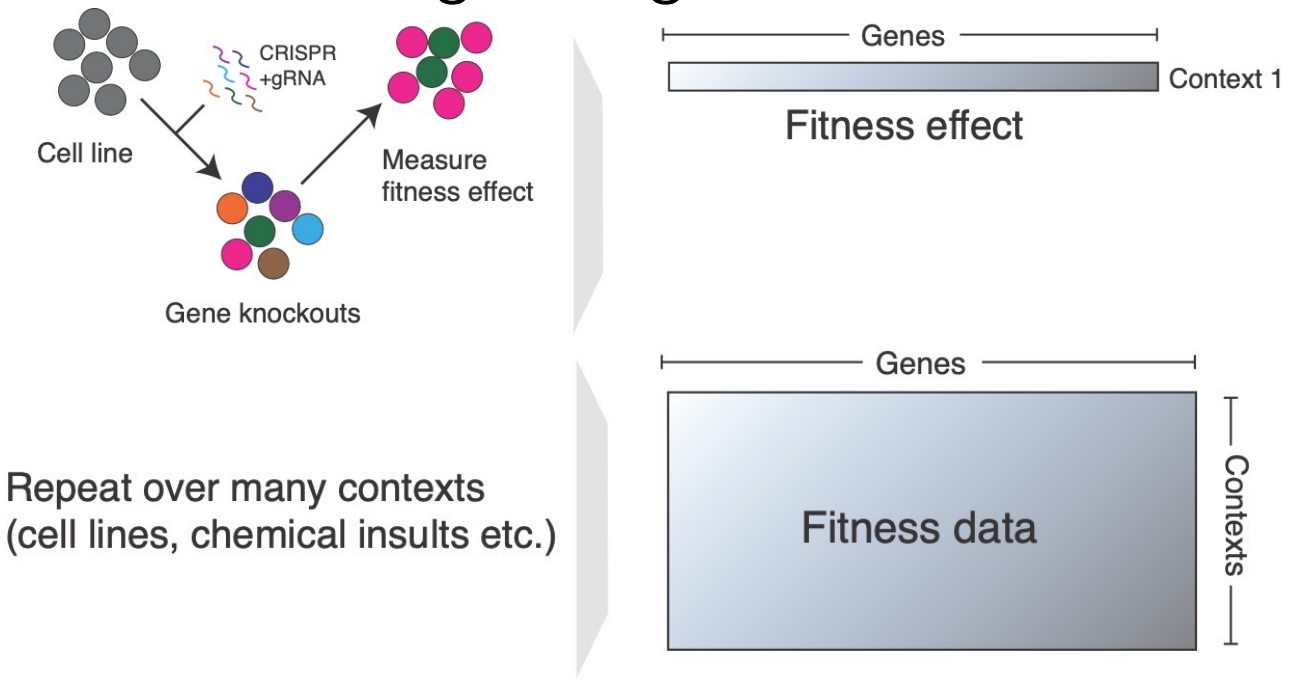
Data: ?

Approach: ?

geneA - func1 + func2 ≈ geneB

Data

Use **gene perturbation effect measurements** for inferring biological functions

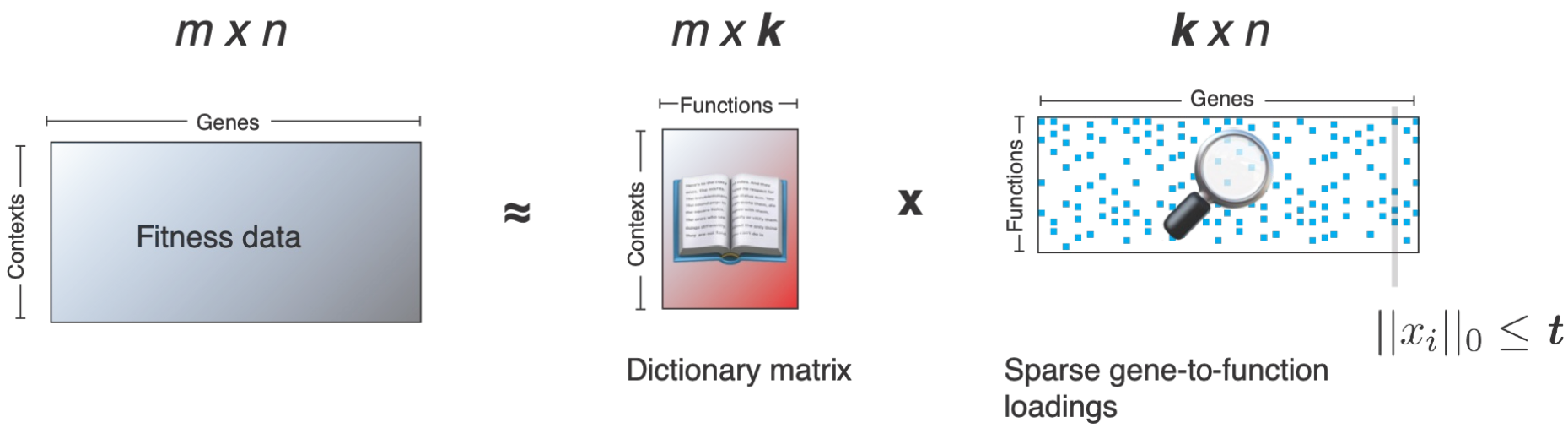


Why perturbation datasets? Alternative data types:

- **Transcriptomics:** gene co-expression is necessary but not sufficient for co-function
- **Protein-protein interactions:** direct interactions are not necessary for co-function

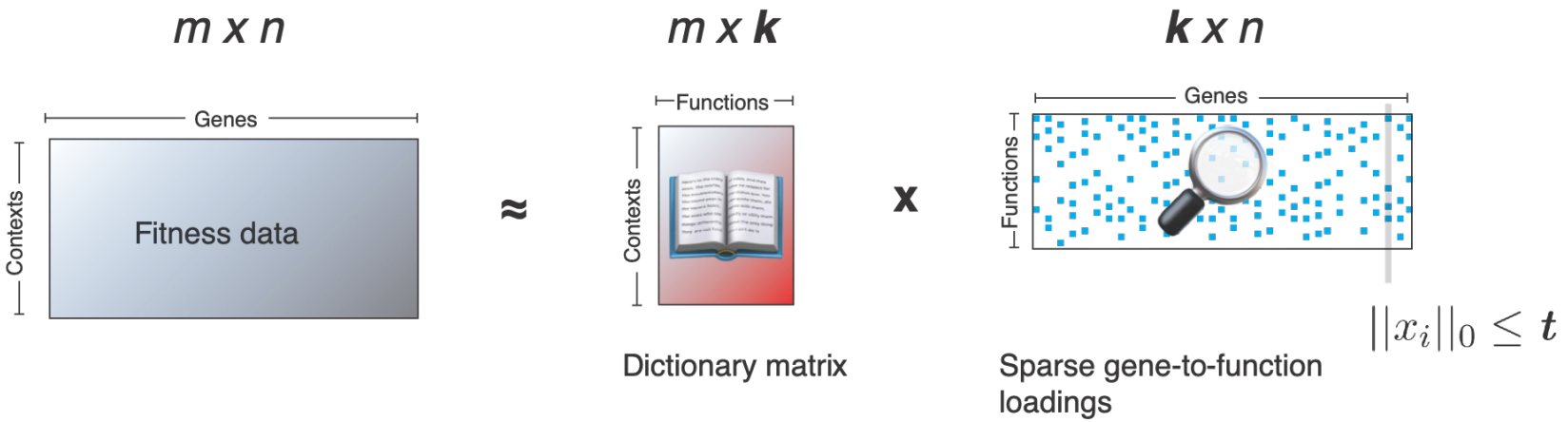
Approach: Webster

- **Low-dimensional vector embeddings** that satisfy three criteria:
  - Sparse
  - Latents are biologically meaningful
  - Account for redundancy between cell contexts

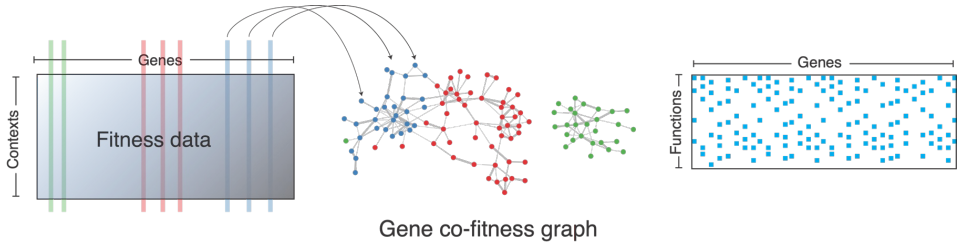


Approach: Webster

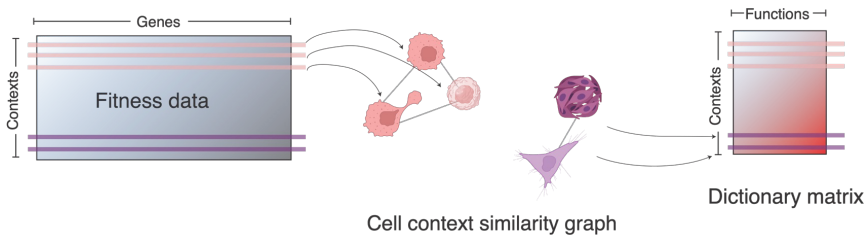
Webster learns a dictionary matrix that **sparsely** approximates gene effects…



1 … while preserving interpretable relationships between genes



2 … and accounting for redundancies between cell contexts

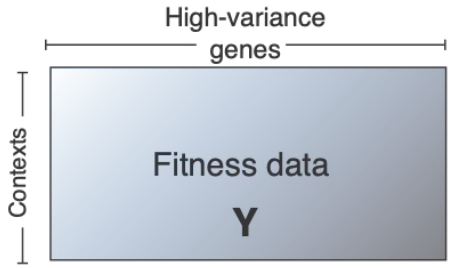


Overview of Webster

Preprocessing

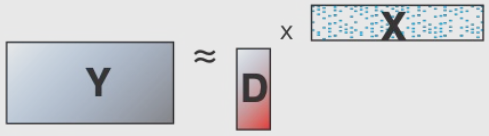
Raw fitness data

Standardize cell lines
Center gene effects
Filter genes by variance



Graph-regularized dictionary learning
Objectives

Reduce dimensionality

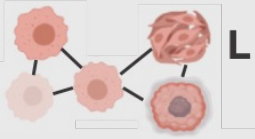


$$\|Y - DX\|_F^2$$

Preserve gene similarity

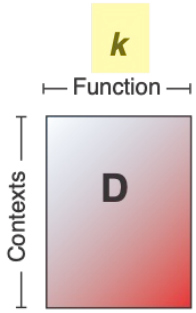
$$Tr(XL_cX^T)$$

Preserve cell context similarity

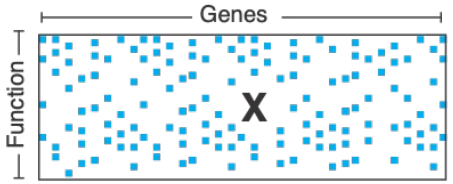


$$Tr(D^T LD)$$

Output



Dictionary matrix

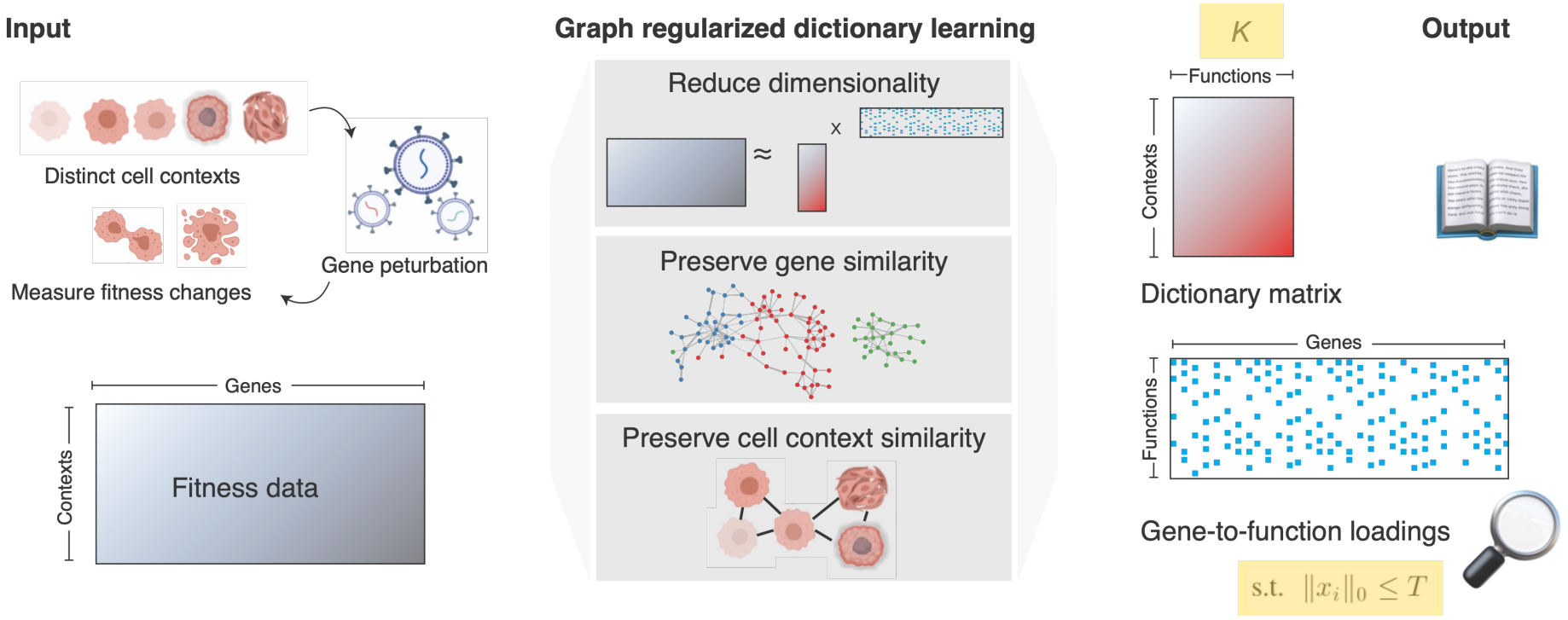


Gene-to-function loadings

$$||x_i||_0 \leq t$$

= key hyperparameters

Its key parameters are dictionary size (K) and sparsity on loadings (T)

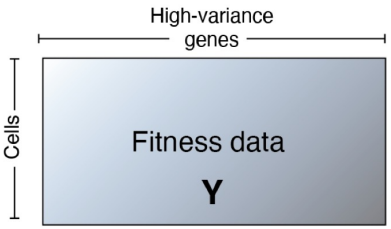


Model optimization

Preprocessing

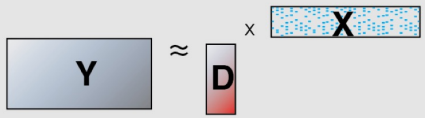
Raw fitness data

Standardize cell lines
Filter genes by variance



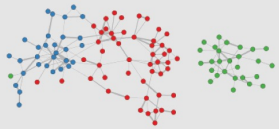
Graph-regularized dictionary learning
Objectives

Reduce dimensionality



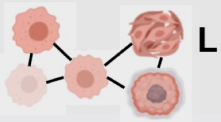
$$\|Y - DX\|_F^2$$

Preserve gene dependency manifold



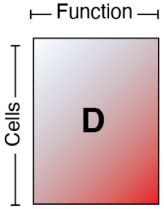
$$Tr(XL_cX^T)$$

Preserve cell state manifold

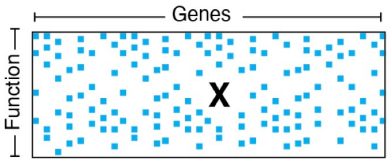


$$Tr(D^T LD)$$

Output



Dictionary matrix



Gene-to-funciton loadings

$$\text{s.t.} \quad \|x_i\|_0 \leq T$$

Parameters

$$\arg\min_{D,X} \|Y - DX\|_F^2 + \alpha Tr(D^T LD)$$
$$+ \beta Tr(XL_cX^T) \quad \text{s.t.} \quad \|x_i\|_0 \leq T \quad \forall i.$$

k	=latent dimension size	α	=weight of cell Laplacian
L	=cell Laplacian (num neighbors, metric)	β	=weight of gene Laplacian
Lc	=gene Laplacian (num neighbors, metric)	T	=sparsity

Applications to three screens of gene perturbation effects

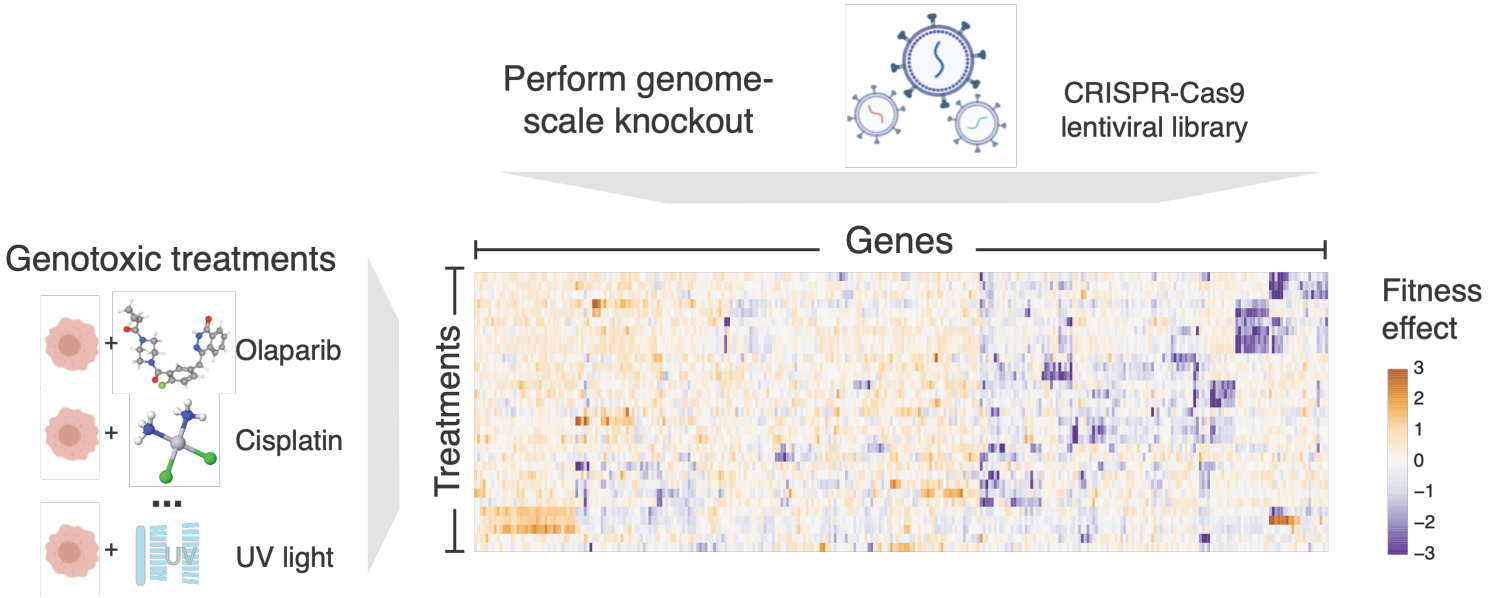
1) Genotoxic screens

2) Cancer fitness screens

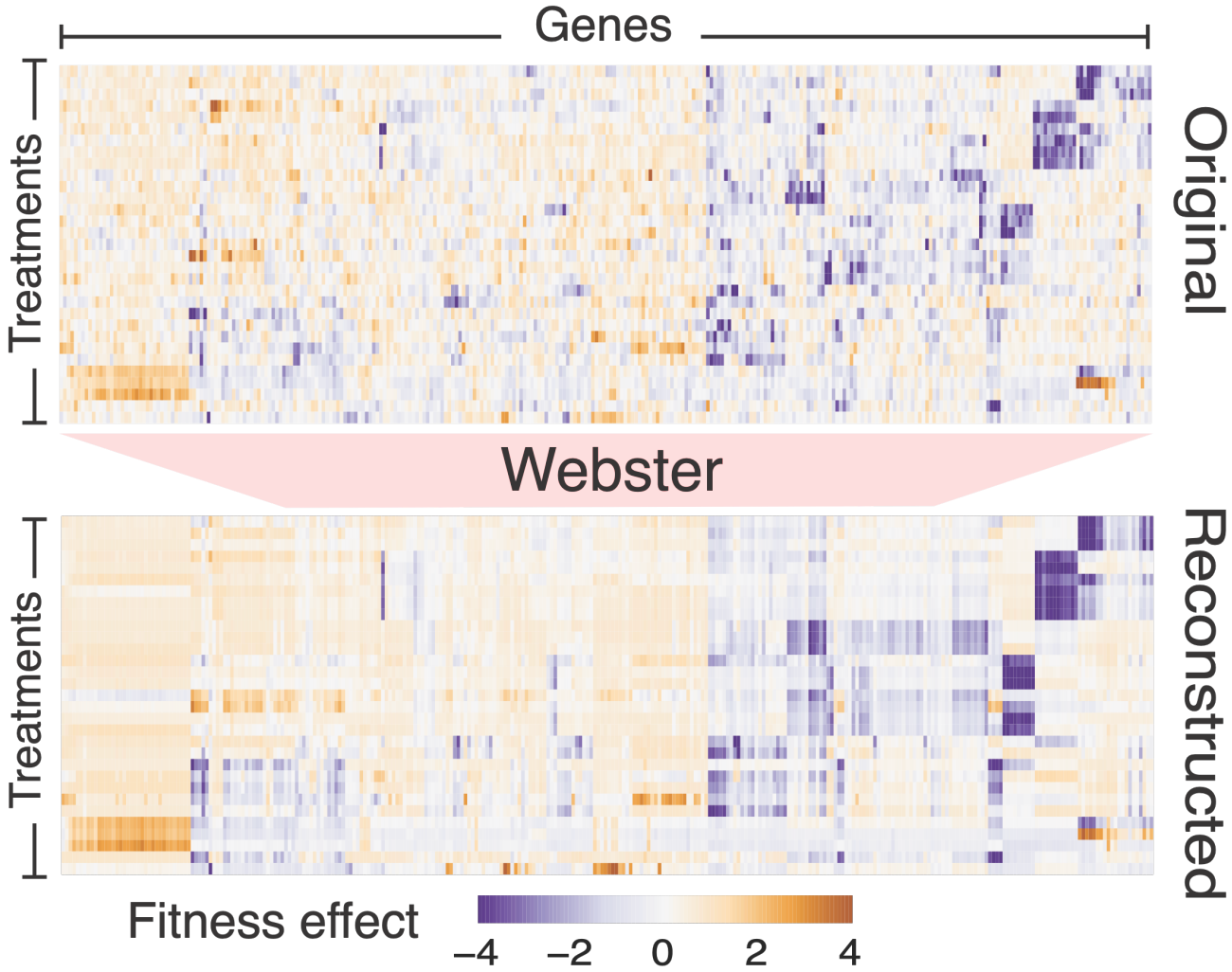
3) Compound sensitivity screens

Part 1: Genotoxic screens

Olivieri et al. 2020: fitness effect of gene knockout in presence of genotoxins

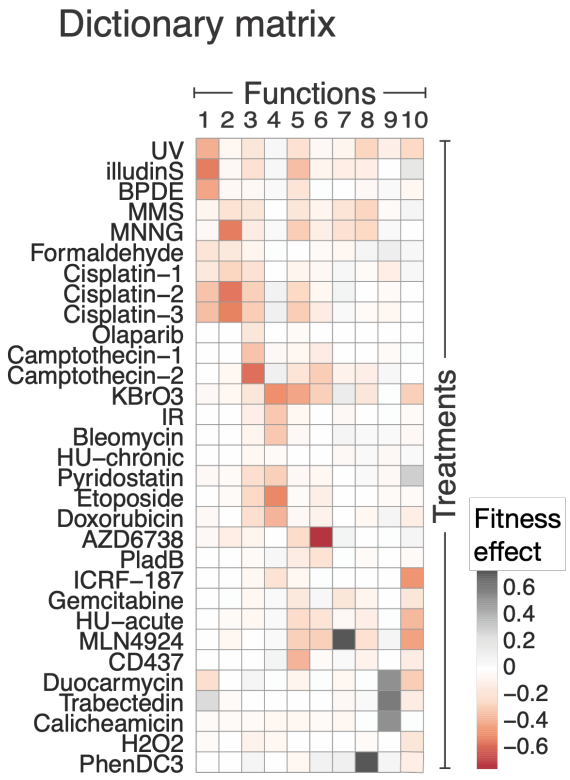


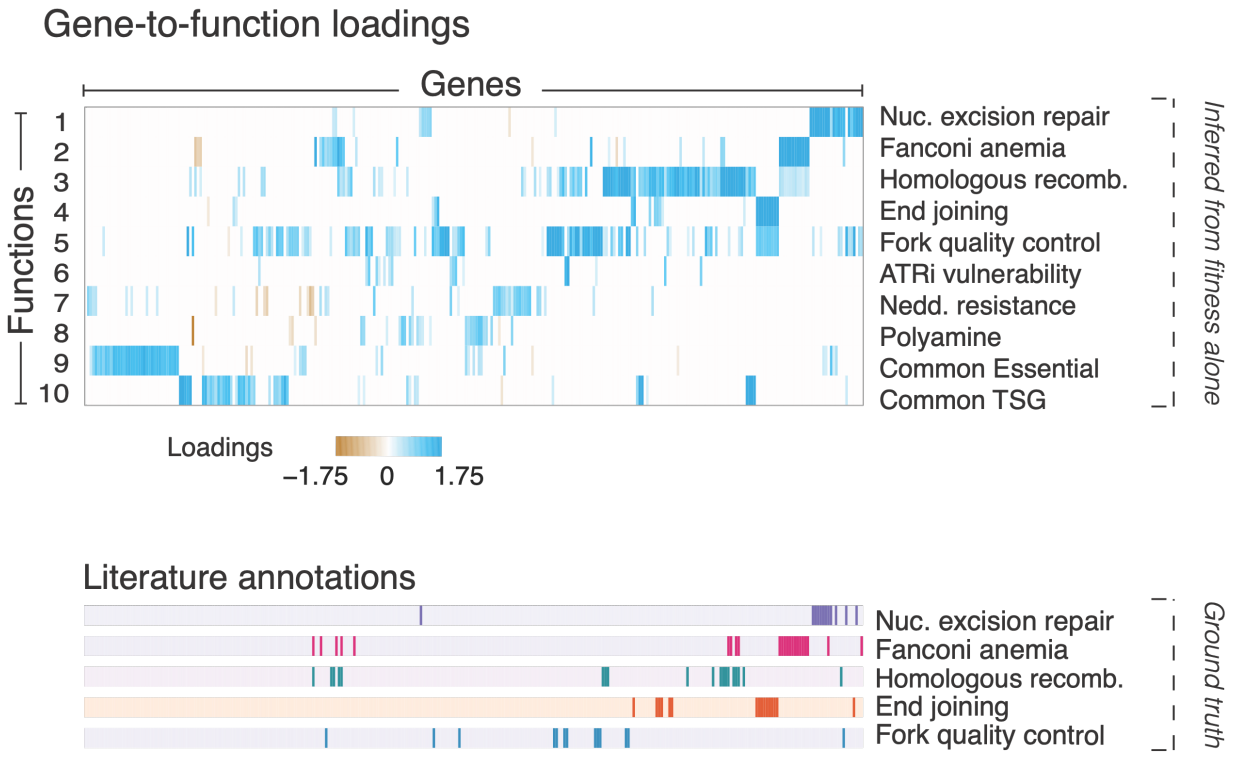
Webster approximates the input data matrix…



k=10
t=2

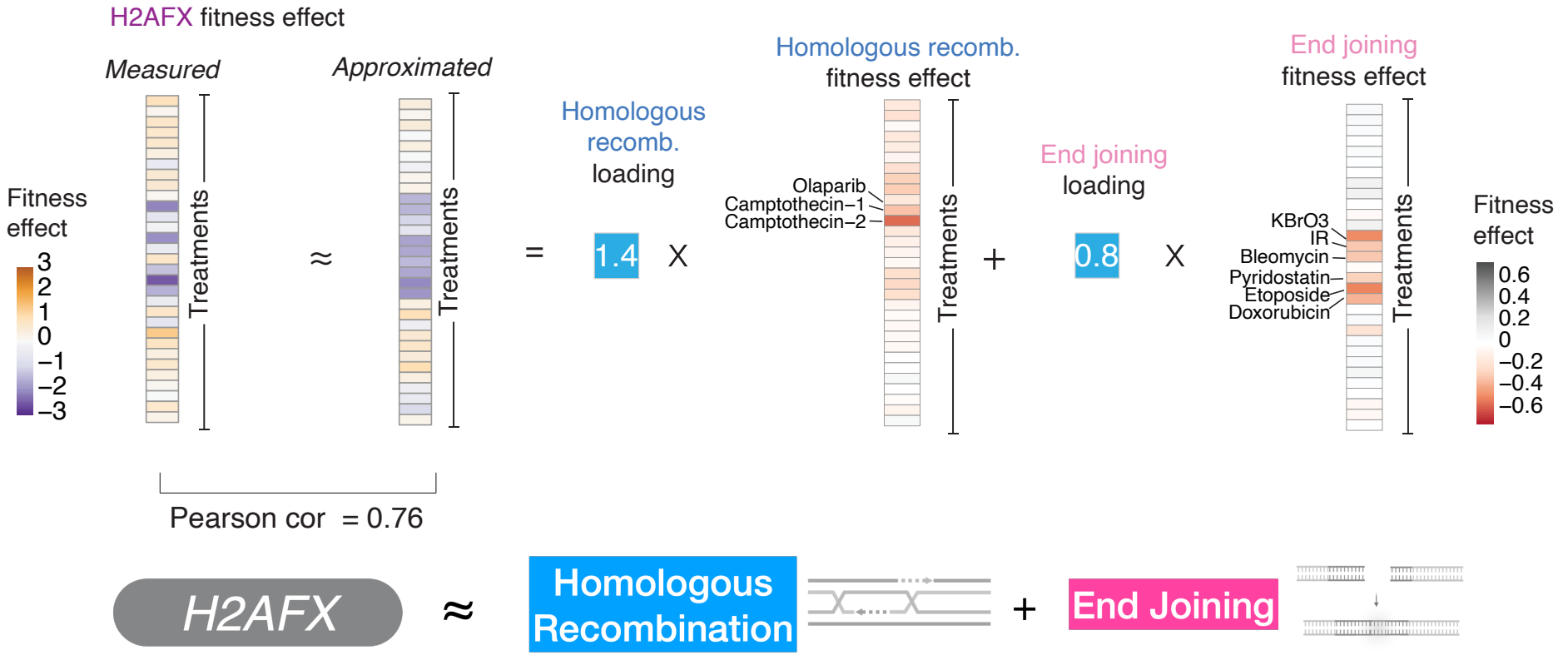
… as a product between a dictionary matrix and a loadings matrix





Learned gene-to-function loadings recover biological genesets hidden during model training

Latents inferred by the model recapitulate pleiotropy *without prior knowledge*

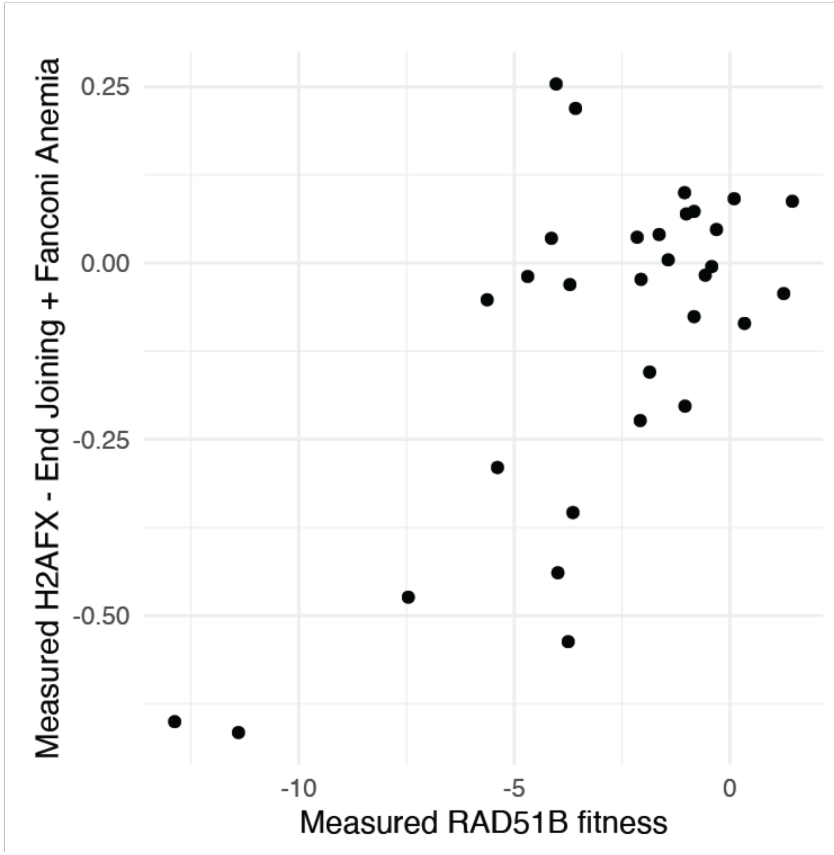


(hidden during model training!)

Latents are biologically meaningful

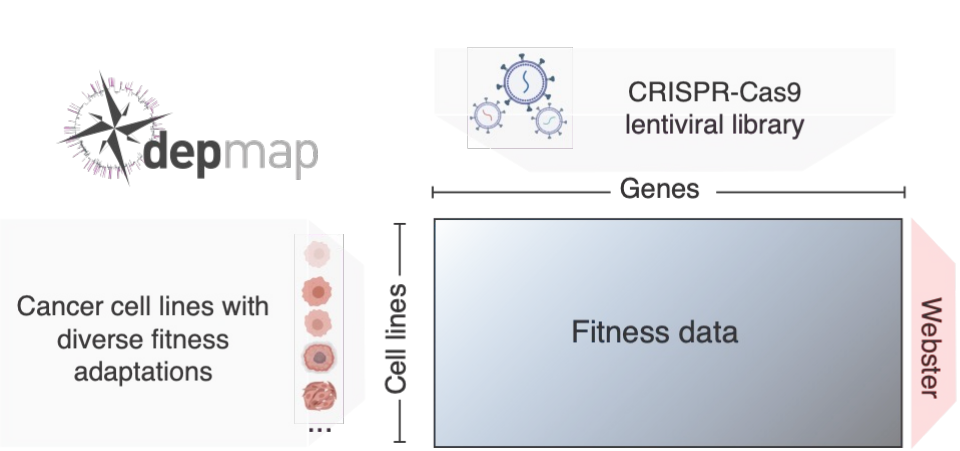
geneA - func1 + func2 ≈ geneB

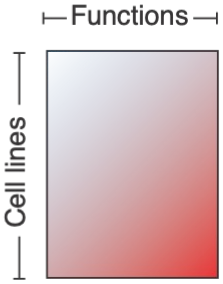
H2AFX - End Joining + Fanconi Anemia ≈ RAD51B



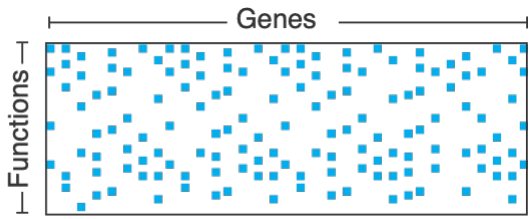
= cell context (treatment)

Part 2: Cancer fitness screens





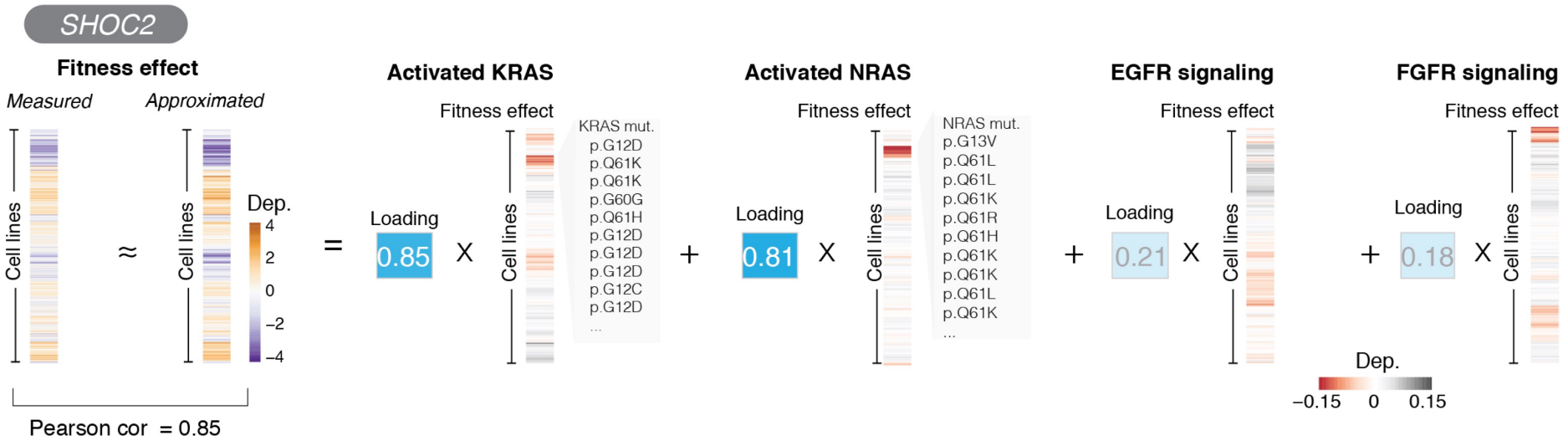
Dictionary matrix
675 cell lines x 220 functions



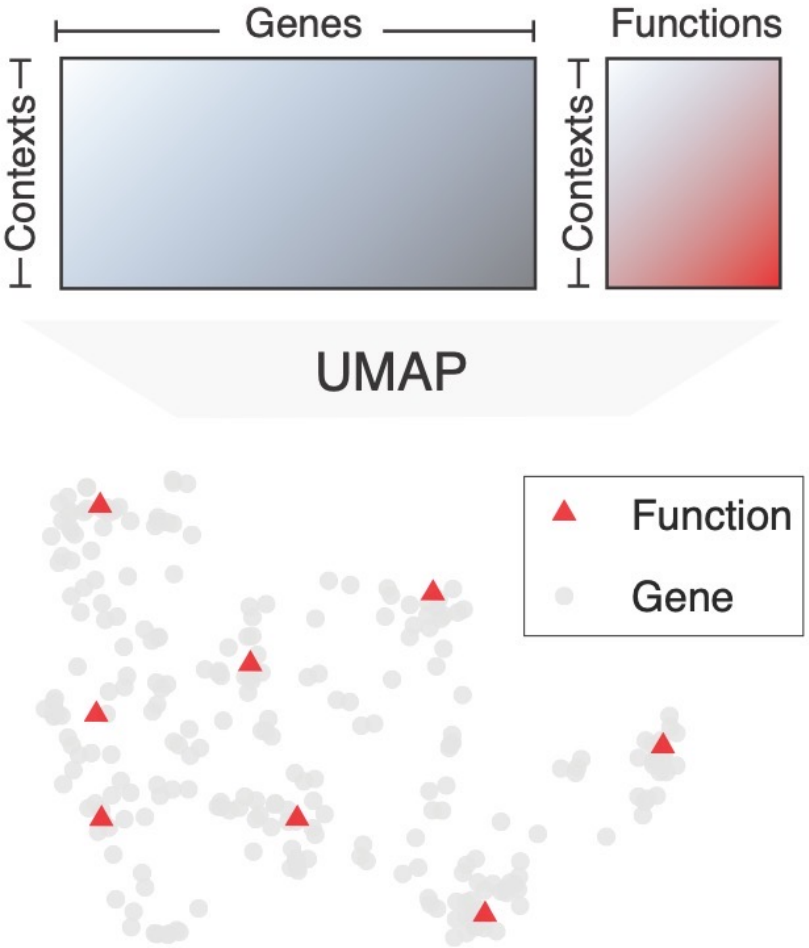
Gene to function loadings
220 functions x 2,921 genes, T = 4

Pleiotropic genes obey linear semantics in the latent space

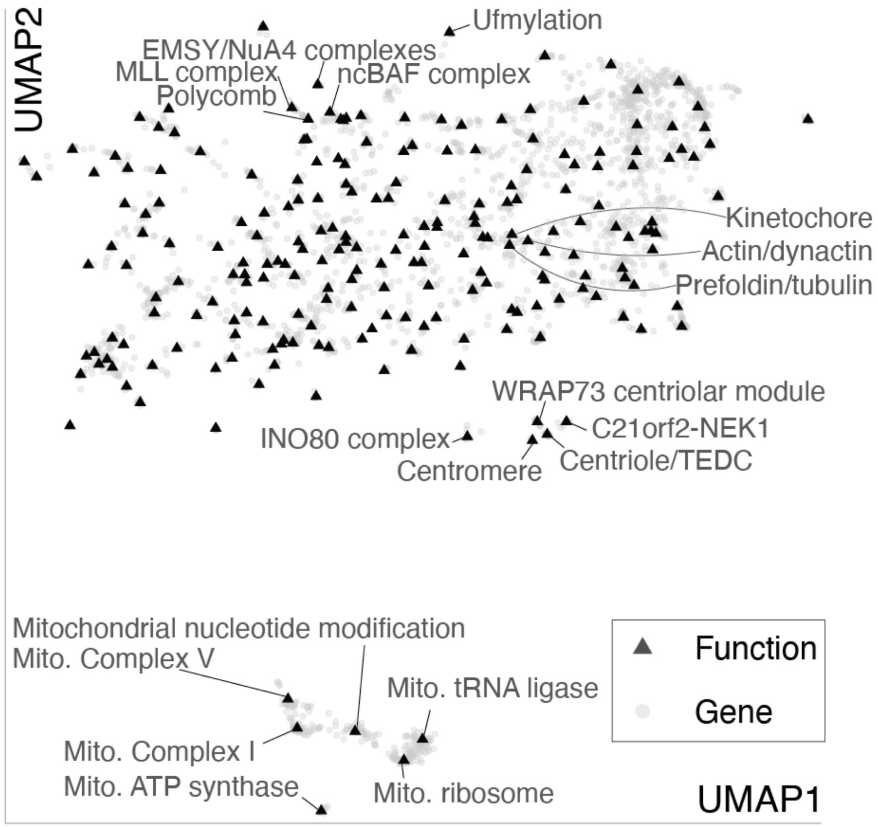
SHOC2 ≈ Activated KRAS + Activated NRAS + EGFR Signaling + FGFR Signaling



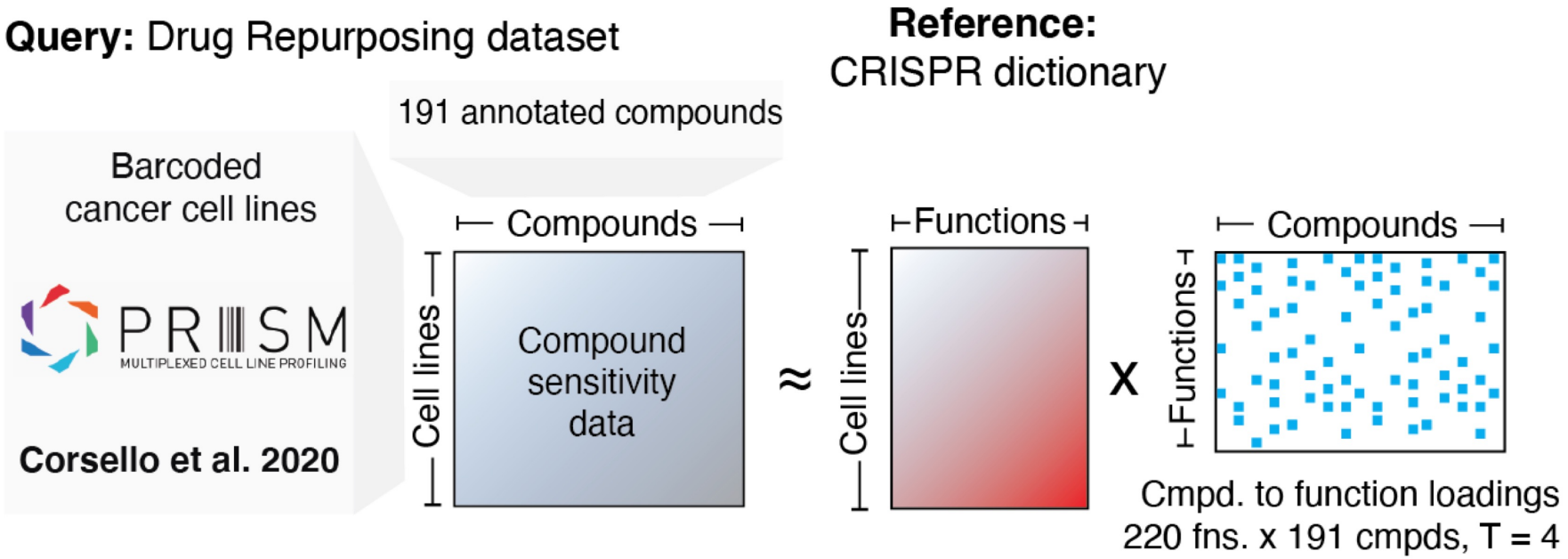
Joint embedding space of genes and functions



It captures interpretable processes in cancer



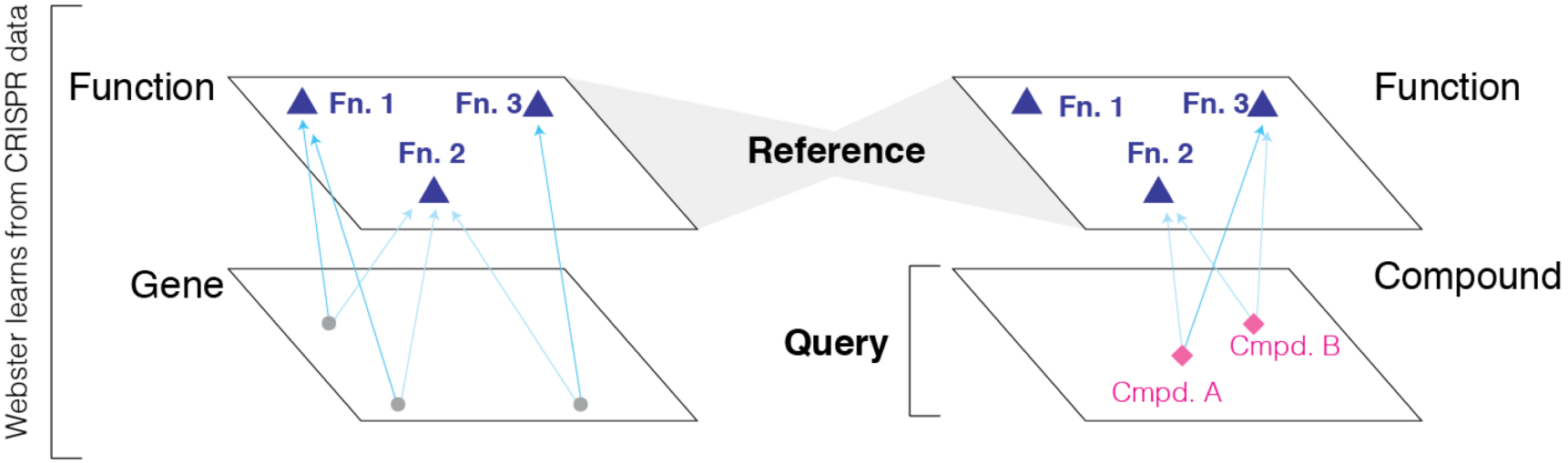
Part 3: Compound sensitivity screens



Modeling compound sensitivity profiles as mixtures of functions learned from CRISPR

Modeling compounds as mixtures of latent functions

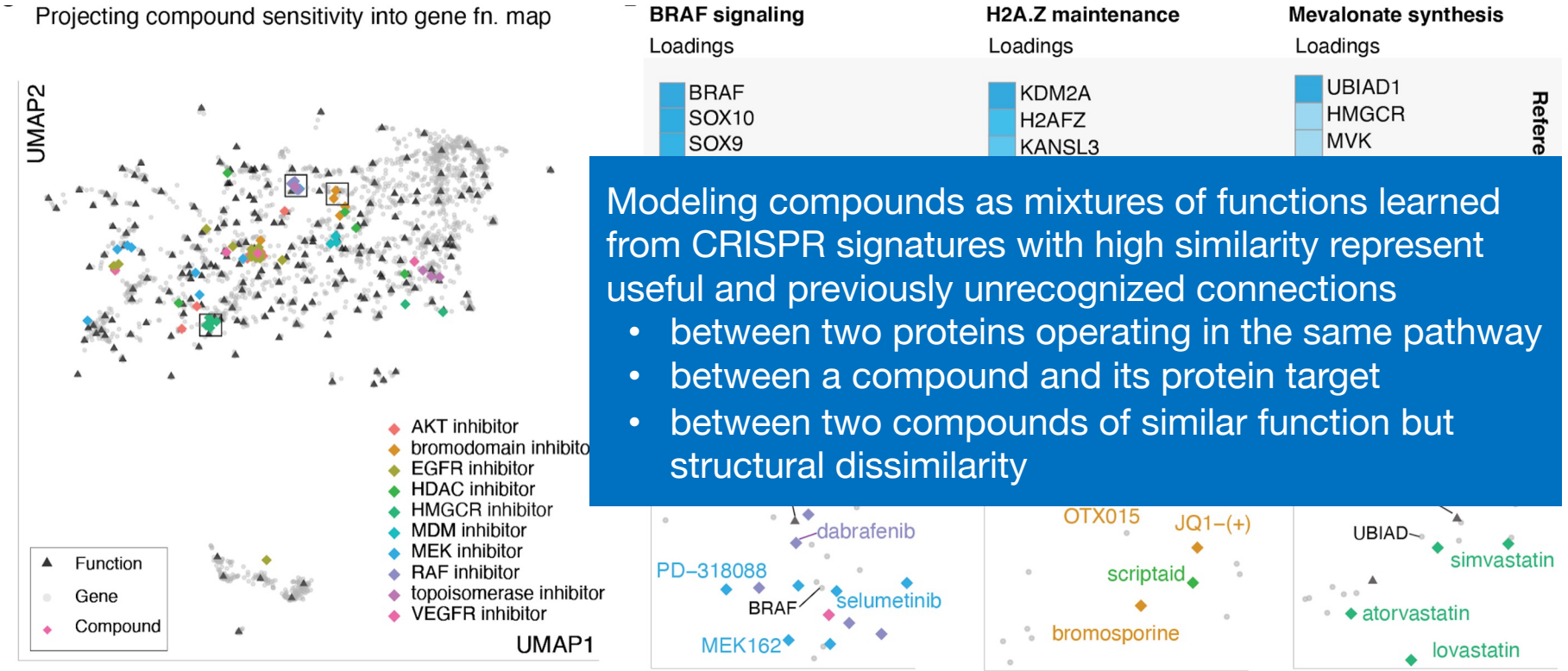
Reference-query projection



- Modeling compounds as mixtures of functions learned from CRISPR signatures with high similarity represent useful and previously unrecognized connections
  - between two proteins operating in the same pathway
  - between a small-molecule and its protein target
  - between two small-molecules of similar function but structural dissimilarity
- Such a catalog of connections can serve as a functional look-up table of compounds to predict sensitivity and genotoxic profiles and to inform therapeutic use

Compounds' mechanisms of action

Compounds are embedded nearby gene functions, reflecting their mechanism of action



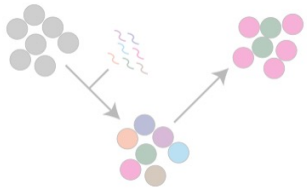
Modeling compounds as mixtures of functions learned from CRISPR signatures with high similarity represent useful and previously unrecognized connections

- between two proteins operating in the same pathway
- between a compound and its protein target
- between two compounds of similar function but structural dissimilarity

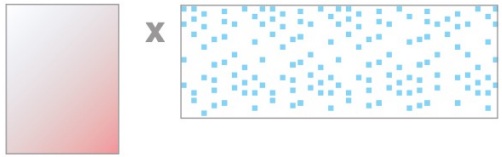
Key takeaways

- Analogously to word semantics, genes can be modeled as **distributions over latent bio functions**
  - **Sparse learning** is an effective strategy for learning bio functions from high-dimensional chemical and genetic perturbations
  - New perturbations can be **projected** into learned space

Data: high-dimensional gene perturbation measurements

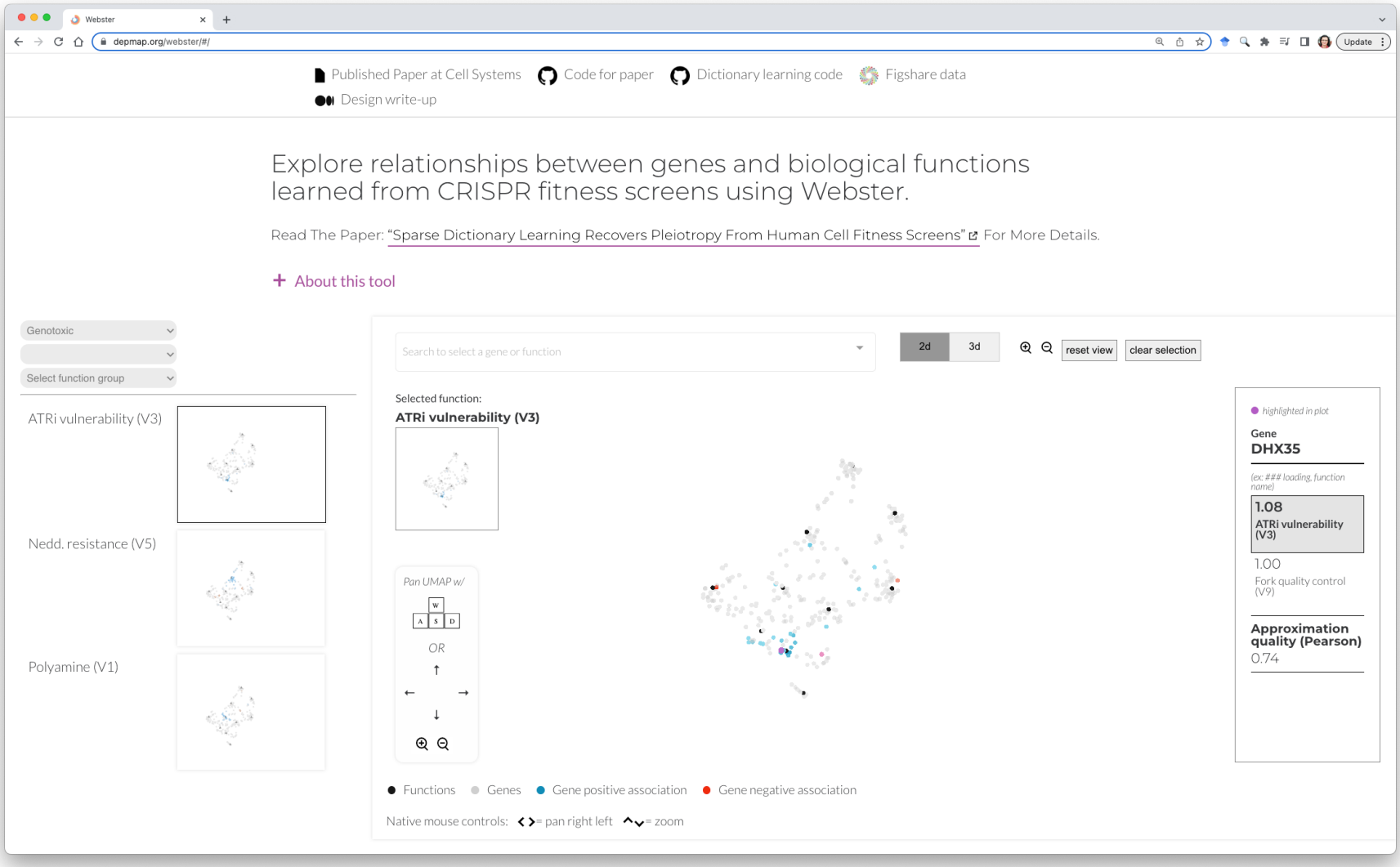


Approach: sparse approximation embeddings



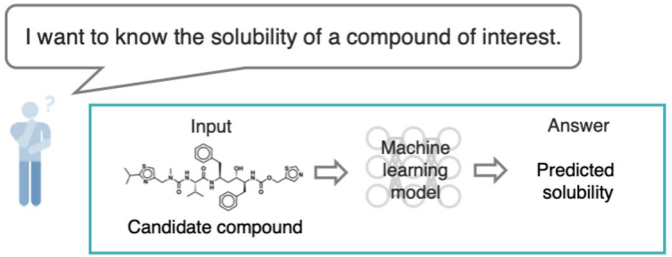
geneA - func1 + func2 ≈ geneB

https://depmap.org/webster

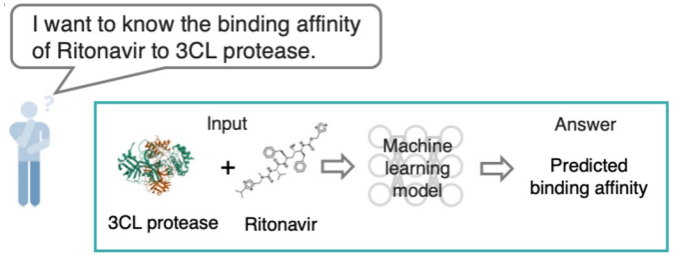


Outline for today's class

- Optimization & generation of small molecules



- Binding of drugs to therapeutic targets



- High-throughput genetic & chemical perturbations