JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

Machine Learning Based Prediction of Half-Lives of Environmental Pollutants

Bachelor Thesis

Steven Lang

September 13, 2017

Introduction

The problem:

- Industries release chemicals into the environment

Introduction

The problem:

- Industries release chemicals into the environment
- Chemicals can cause harm (e.g. pesticides)

Introduction

The problem:

- Industries release chemicals into the environment
- Chemicals can cause harm (e.g. pesticides)
- Persistence of harmful chemicals is strongly undesired

Introduction

The problem:

- Industries release chemicals into the environment
- Chemicals can cause harm (e.g. pesticides)
- Persistence of harmful chemicals is strongly undesired
- It is necessary to know the half-life of a molecule beforehand

Introduction

The problem:

- Industries release chemicals into the environment
- Chemicals can cause harm (e.g. pesticides)
- Persistence of harmful chemicals is strongly undesired
- It is necessary to know the half-life of a molecule beforehand

$\Rightarrow$ Solution: Machine learning based prediction of half-lives

The Data

Eawag-Soil package:

- Published by Latino et al. [1]
- Available on *enviPath* (environmental contaminant biotransformation pathway resource system)
- Microbial biotransformation pathways
- Meta-data of organic contaminants in different environments
- Half-life values for compounds under certain environmental conditions

[1] Latino *et al.*, "Eawag-Soil in enviPath: a new resource for exploring regulatory pesticide soil biodegradation pathways and half-life data"

The Data

- Contains:
  - 744 unique **compounds**
  - 3108 unique **scenarios** (set of environmental conditions)
  - 4890 **half-lives**
  - $\frac{4890}{744 \times 3108} = 0.21\%$ occupation rate

The Data: Compounds

Given as SMILES string, e.g. 7-OH-metosulam:

CC1=CC=C(C(=C1Cl)NS(=O)(=O)C2=NN3C(=CC(=NC3=N2)OC)O)Cl

The Data: Compounds

Given as SMILES string, e.g. 7-OH-metosulam:

CC1=CC=C(C(=C1Cl)NS(=O)(=O)C2=NN3C(=CC(=NC3=N2)OC)O)Cl

Fingerprinter:

- Translate molecule structure into bit-vector

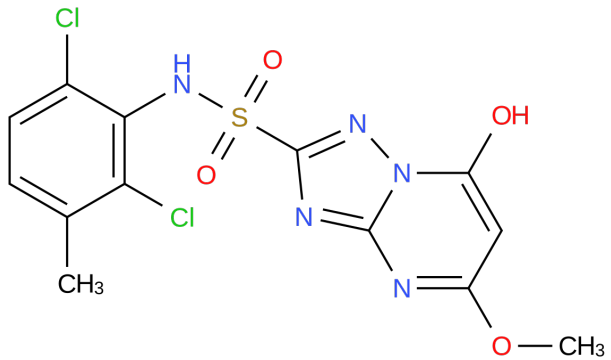


Figure: Molecule structure of 7-OH-metosulam

The Data: Compounds

Given as SMILES string, e.g. 7-OH-metosulam:

CC1=CC=C(C(=C1Cl)NS(=O)(=O)C2=NN3C(=CC(=NC3=N2)OC)O)Cl

Fingerprinter:

- Translate molecule structure into bit-vector
- Bits correspond to query results against the graph

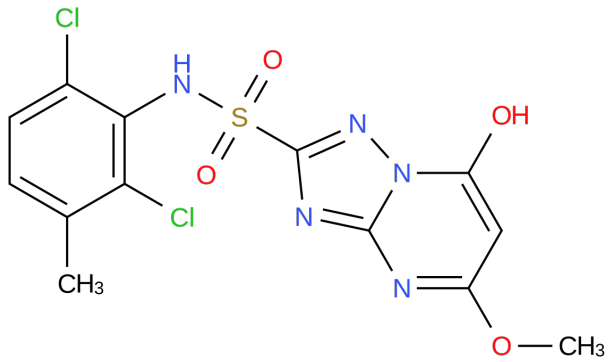


Figure: Molecule structure of 7-OH-metosulam

The Data: Compounds

Given as SMILES string, e.g. 7-OH-metosulam:

CC1=CC=C(C(=C1Cl)NS(=O)(=O)C2=NN3C(=CC(=NC3=N2)OC)O)Cl

Fingerprinter:

- Translate molecule structure into bit-vector
- Bits correspond to query results against the graph
- *"Are there fewer than 3 oxygens?"*
- *"Is there a ring of size 4?"*

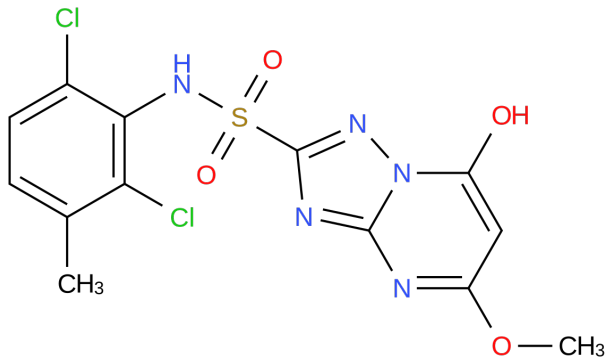


Figure: Molecule structure of 7-OH-metosulam

The Data: Scenarios

Environmental conditions under which the degradation process took place

- 13 numeric features:
  - acidity
  - biomass
  - temperature
  - water storage capacity
  - ...

The Data: Scenarios

Environmental conditions under which the degradation process took place

- 13 numeric features:
  - acidity
  - biomass
  - temperature
  - water storage capacity
  - ...
- 2 categorical features:
  - soil classification system
  - soiltexture (result)

The Data: Preprocessing

- Standard scaling: $z^{(k)} = \frac{x^{(k)} - \mu^{(k)}}{\sigma^{(k)}}$, for the k-th feature

The Data: Preprocessing

- Standard scaling: $z^{(k)} = \frac{x^{(k)} - \mu^{(k)}}{\sigma^{(k)}}$, for the k-th feature
- Target variable transformation: $DT_{50} \rightsquigarrow ln(DT_{50})$

The Data: Preprocessing

- Standard scaling: $z^{(k)} = \frac{x^{(k)} - \mu^{(k)}}{\sigma^{(k)}}$, for the k-th feature
- Target variable transformation: $DT_{50} \rightsquigarrow ln(DT_{50})$
- Cleaning implausible scenarios

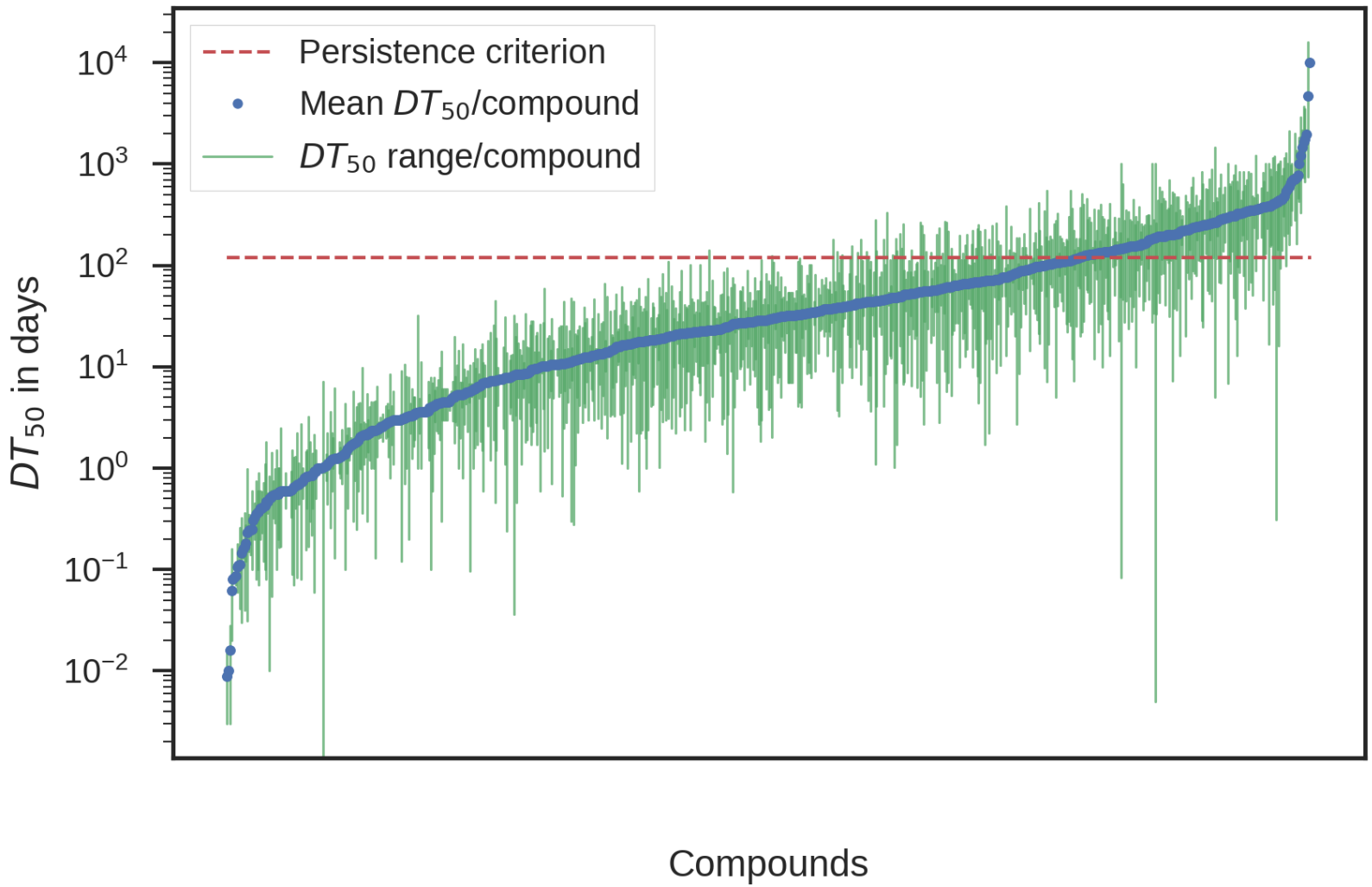
The Data: Preprocessing

- Standard scaling: $z^{(k)} = \frac{x^{(k)} - \mu^{(k)}}{\sigma^{(k)}}$, for the k-th feature
- Target variable transformation: $DT_{50} \rightsquigarrow ln(DT_{50})$
- Cleaning implausible scenarios
- Imputation of missing scenario values:
  - Mean Imputation
  - KNN
  - Matrix Factorization
  - Spectral Regularization
  - Multiple Imputation by Chained Equations

Parameter	Miss.
Acidity, pH	0.02
Biomass end	0.43
Biomass start	0.32
Bulk density	0.62
CEC	0.19
OC	0.04
Spike conc.	0.15
Temperature	0.02
WSC	0.16
% humidity	0.19
% clay	0.11
% sand	0.12
% silt	0.12

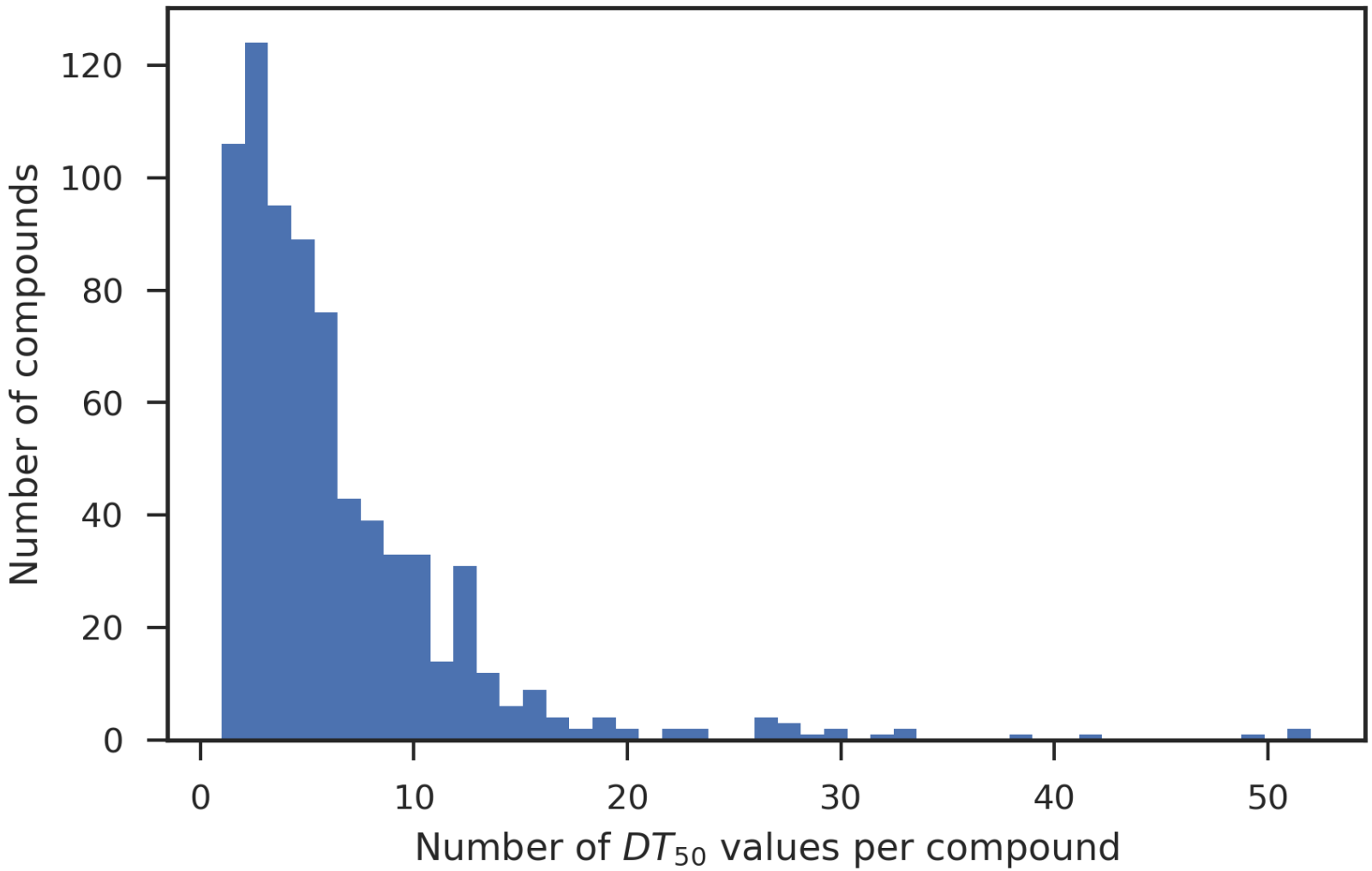
The Data: Half-Lives

DT_{50} Per Compound



The Data: Half-Lives

Distribution



Methods: Models

Baseline approaches:

- Random Forest
- Support Vector Regression
- Bagged Support Vector Regression

Methods: Models

Baseline approaches:

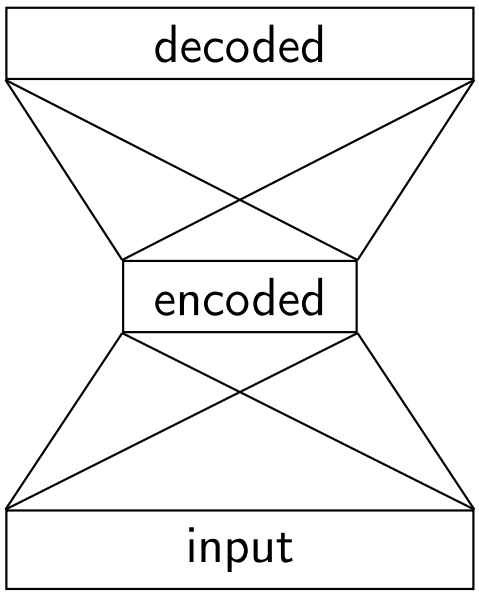
- Random Forest
- Support Vector Regression
- Bagged Support Vector Regression

Advanced approaches:

- Neural Network Regression
- Denoising Autoencoder as feature encoding

Methods: Denoising Autoencoder Regression Network

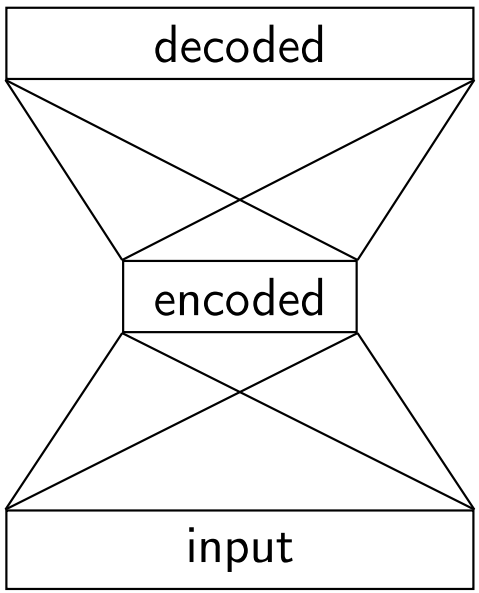
Architecture



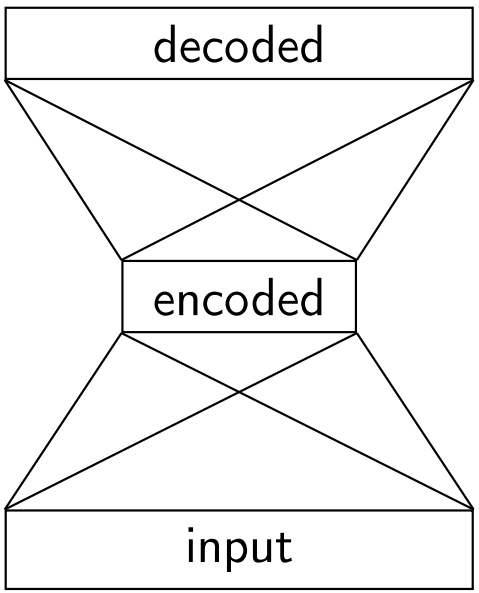
Compound DAE

Methods: Denoising Autoencoder Regression Network

Architecture



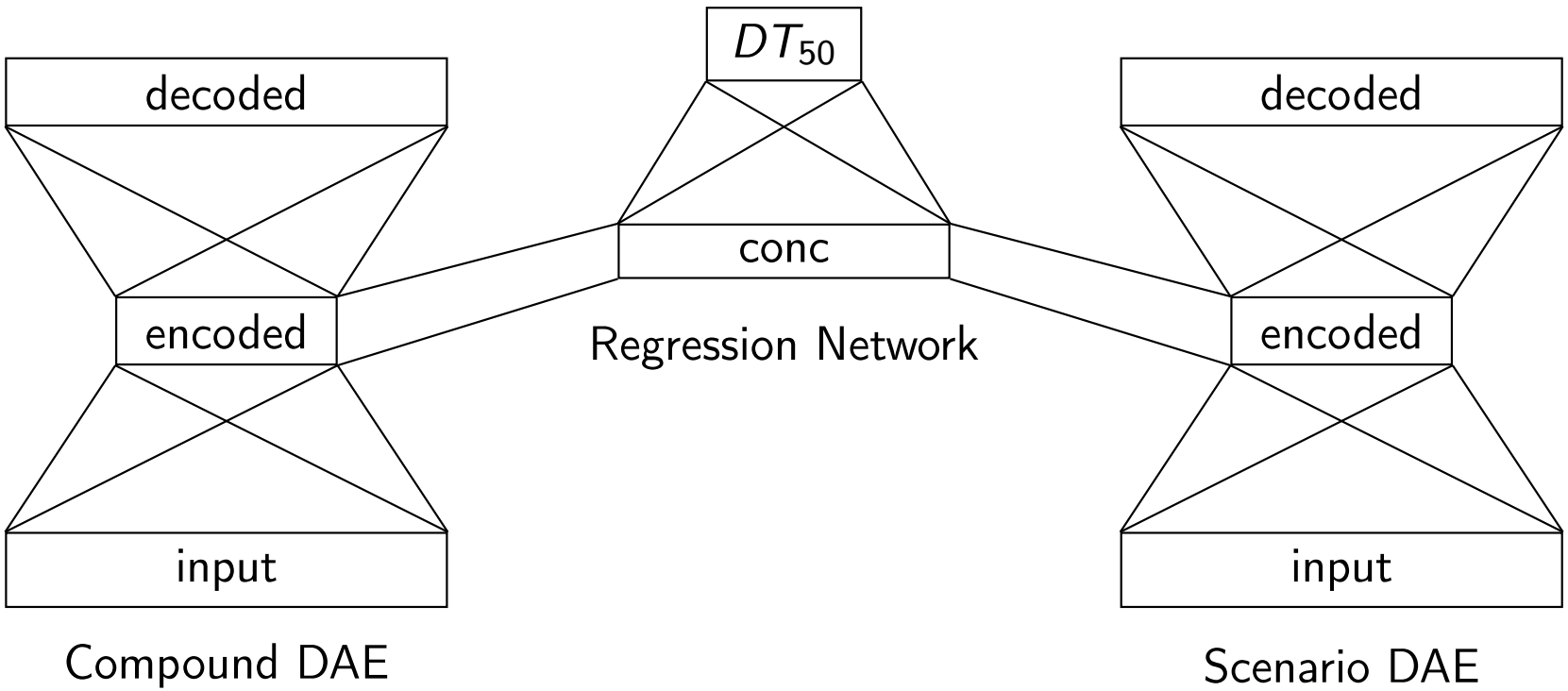
Compound DAE



Scenario DAE

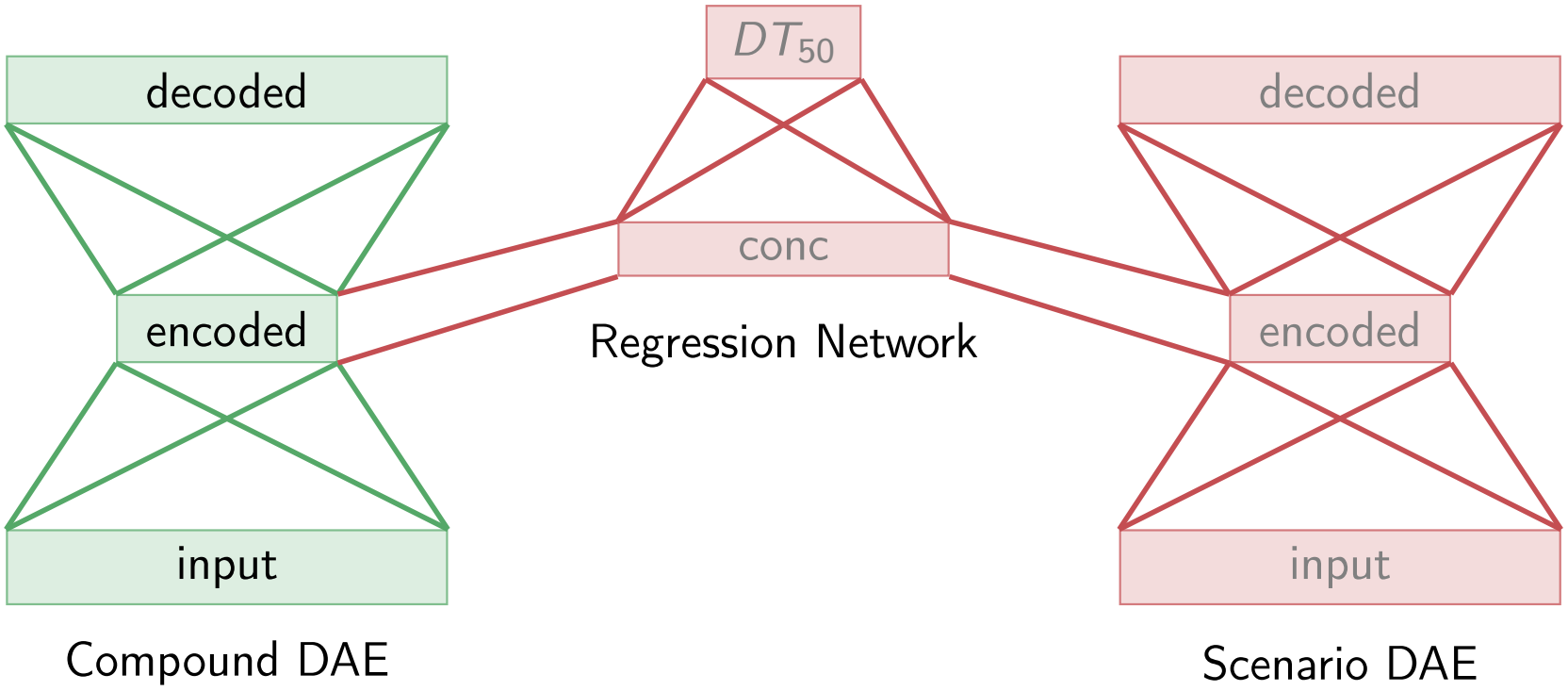
Methods: Denoising Autoencoder Regression Network

Architecture



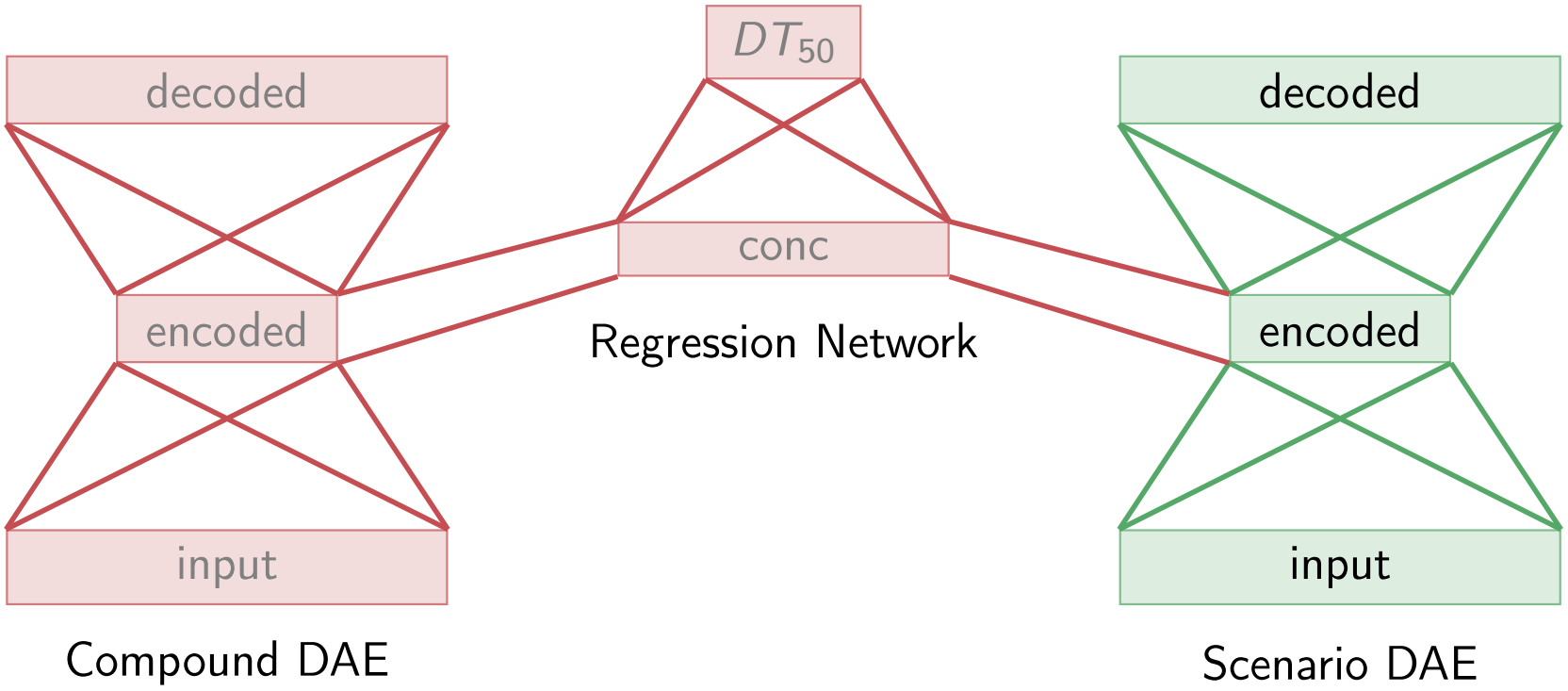
Methods: Denoising Autoencoder Regression Network

Training Phase 1)



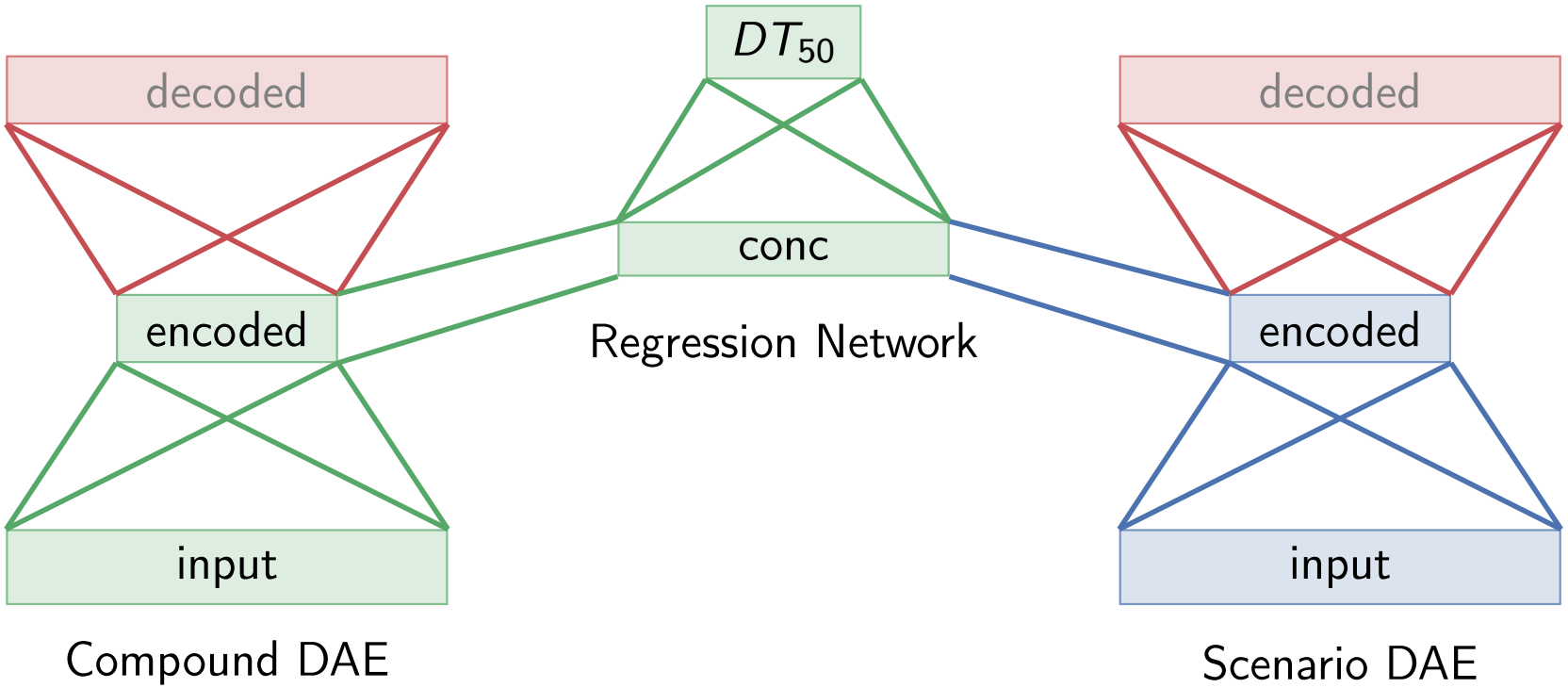
Methods: Denoising Autoencoder Regression Network

Training Phase 2)



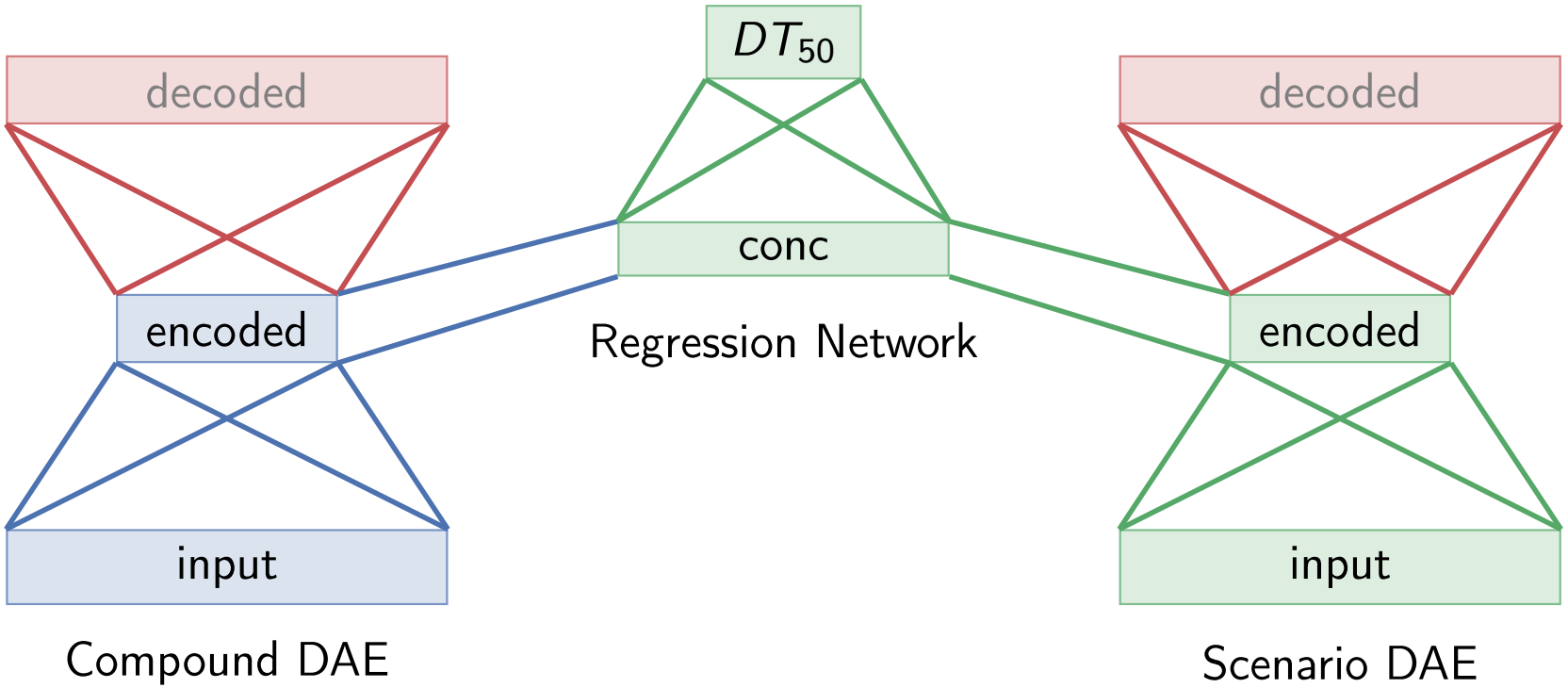
Methods: Denoising Autoencoder Regression Network

Training Phase 3)



Methods: Denoising Autoencoder Regression Network

Training Phase 4)



Methods: Denoising Autoencoder Regression Network

Weight Optimization

$$L = MSE(\mathbf{y}, \hat{\mathbf{y}}) + \lambda_1 R_{weights} + \lambda_2 R_{acts}$$

$$R_{weights} = \sum_i ||\mathbf{W}^{(i)}||_2^2$$

$$R_{acts} = \sum_i ||\mathbf{z}^{(i)}||_2^2$$

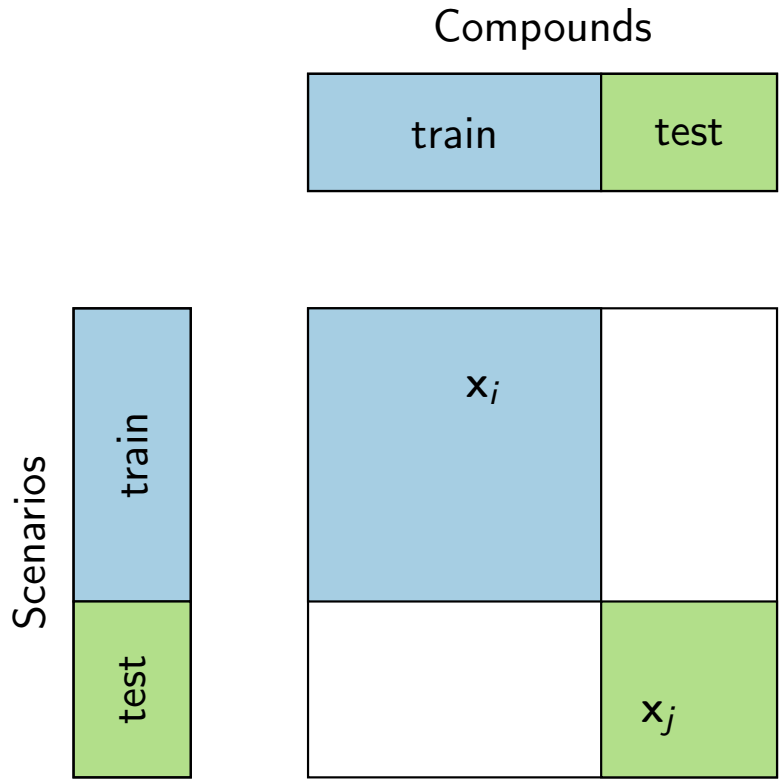
Optimizer: Adam[2], learning rate = 0.01

[2] Kingma and Ba, "Adam: A Method for Stochastic Optimization".

Experiments: Model Validation

10-fold cross validation with standard bi-relational splitting approach

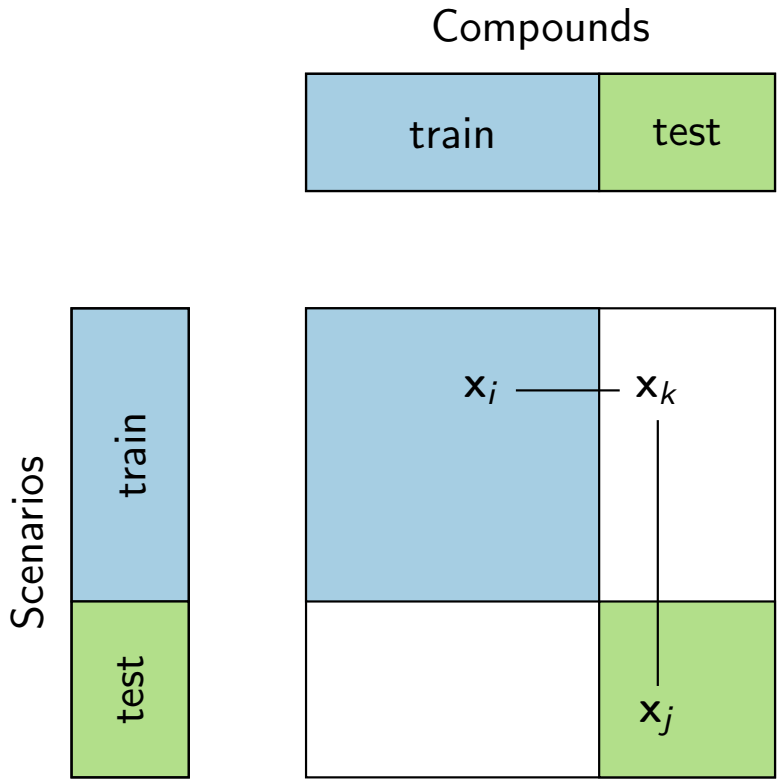
- Train datapoint: $\mathbf{x}_i = (\mathbf{c}_i, \mathbf{s}_i)$
- Test datapoint: $\mathbf{x}_j = (\mathbf{c}_j, \mathbf{s}_j)$ with $\mathbf{c}_i \neq \mathbf{c}_j \wedge \mathbf{s}_i \neq \mathbf{s}_j$



Experiments: Model Validation

10-fold cross validation with standard bi-relational splitting approach

- Train datapoint: $\mathbf{x}_i = (\mathbf{c}_i, \mathbf{s}_i)$
- Test datapoint: $\mathbf{x}_j = (\mathbf{c}_j, \mathbf{s}_j)$ with $\mathbf{c}_i \neq \mathbf{c}_j \wedge \mathbf{s}_i \neq \mathbf{s}_j$
- Where to put $\mathbf{x}_k = (\mathbf{c}_k, \mathbf{s}_k)$ with $\mathbf{c}_k = \mathbf{c}_j \wedge \mathbf{s}_k = \mathbf{s}_i$?

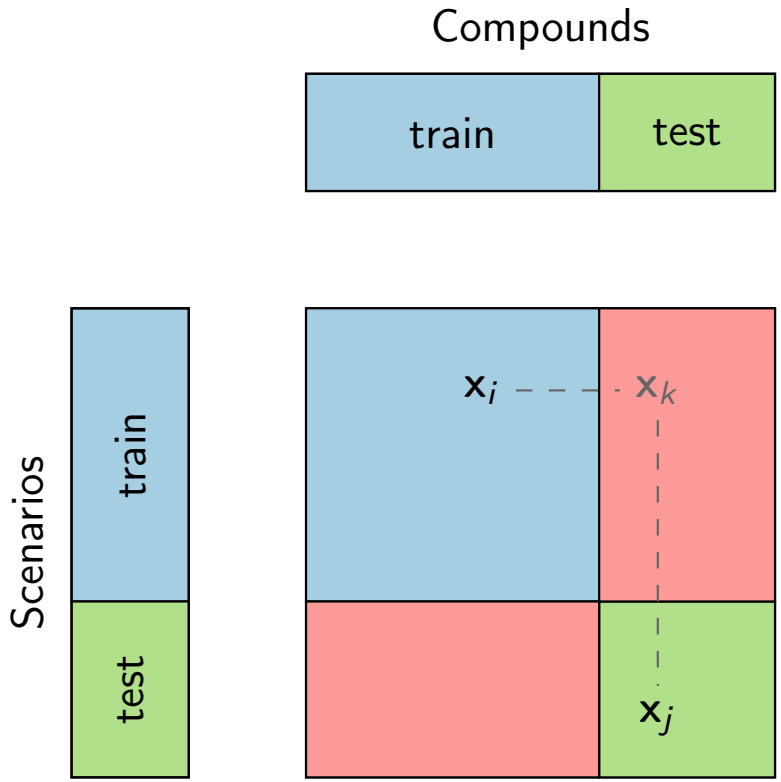


Experiments: Model Validation

10-fold cross validation with standard bi-relational splitting approach

- Train datapoint: $\mathbf{x}_i = (\mathbf{c}_i, \mathbf{s}_i)$
- Test datapoint: $\mathbf{x}_j = (\mathbf{c}_j, \mathbf{s}_j)$ with $\mathbf{c}_i \neq \mathbf{c}_j \wedge \mathbf{s}_i \neq \mathbf{s}_j$
- Where to put $\mathbf{x}_k = (\mathbf{c}_k, \mathbf{s}_k)$ with $\mathbf{c}_k = \mathbf{c}_j \wedge \mathbf{s}_k = \mathbf{s}_i$?

$\Rightarrow$ Remove $\mathbf{x}_k$ for validation split



Experiments: Model Validation

10-fold cross validation with advanced bi-relational splitting approach

- DT_{50} matrix as graph G

Experiments: Model Validation

10-fold cross validation with advanced bi-relational splitting approach

- DT_{50} matrix as graph G
- Each datapoint **x** is a node

Experiments: Model Validation

10-fold cross validation with advanced bi-relational splitting approach

- DT_{50} matrix as graph G
- Each datapoint **x** is a node
- Two nodes are connected if they share the same compound, or the same scenario

Experiments: Model Validation

10-fold cross validation with advanced bi-relational splitting approach

- DT_{50} matrix as graph G
- Each datapoint **x** is a node
- Two nodes are connected if they share the same compound, or the same scenario
- Goal: remove nodes, such that G is disconnected into two subgraphs (train, test set)

Experiments: Model Validation

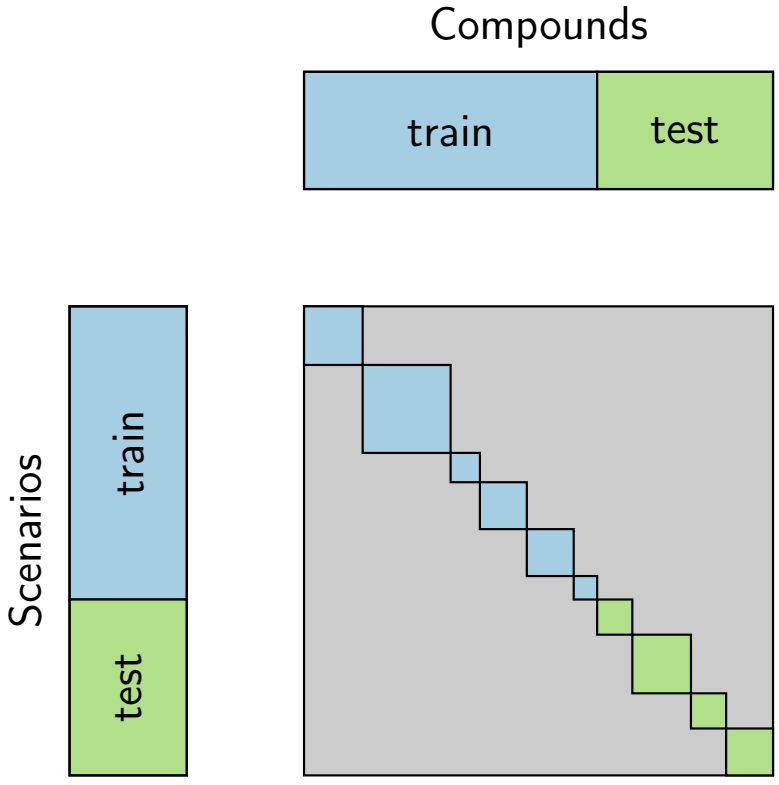
10-fold cross validation with advanced bi-relational splitting approach

- DT_{50} matrix as graph G
- Each datapoint **x** is a node
- Two nodes are connected if they share the same compound, or the same scenario
- Goal: remove nodes, such that G is disconnected into two subgraphs (train, test set)
- Sparseness (0.21%) of the data already builds disconnected subgraphs (408)

Experiments: Model Validation

10-fold cross validation with advanced bi-relational splitting approach

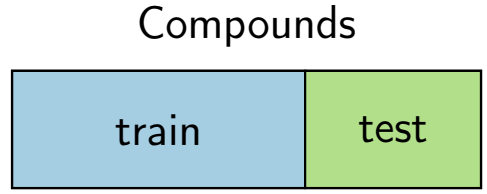
- DT_{50} matrix as graph G
- Each datapoint **x** is a node
- Two nodes are connected if they share the same compound, or the same scenario
- Goal: remove nodes, such that G is disconnected into two subgraphs (train, test set)
- Sparseness (0.21%) of the data already builds disconnected subgraphs (408)
- Select subgraphs for train set and use all others for test set

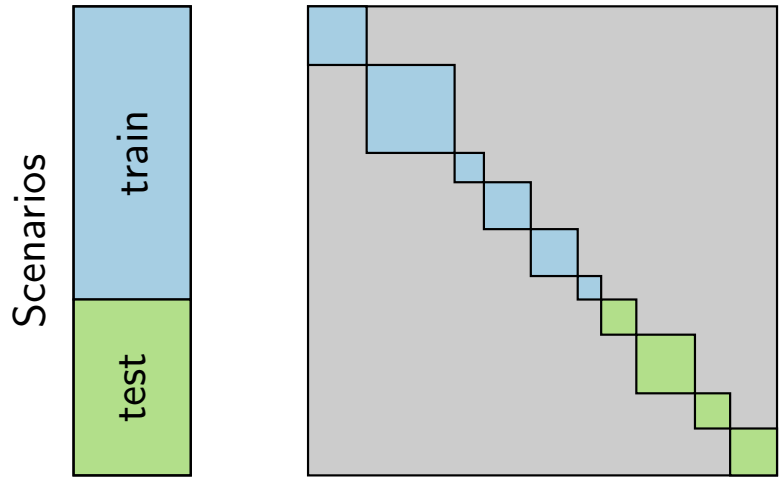


Experiments: Model Validation

10-fold cross validation with advanced bi-relational splitting approach

- DT_{50} matrix as graph G
- Each datapoint **x** is a node
- Two nodes are connected if they share the same compound, or the same scenario
- Goal: remove nodes, such that G is disconnected into two subgraphs (train, test set)
- Sparseness (0.21%) of the data already builds disconnected subgraphs (408)
- Select subgraphs for train set and use all others for test set





⇒ **Splitting problem solved while using *all* datapoints available**

Experiments: Model Validation

Validation Metric

Coefficient of determination:

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2}$$

Experiments: Model Validation

Validation Metric

Coefficient of determination:

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

Interesting values:

- $R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 \iff \hat{\mathbf{y}} = \mathbf{y}$

Experiments: Model Validation

Validation Metric

Coefficient of determination:

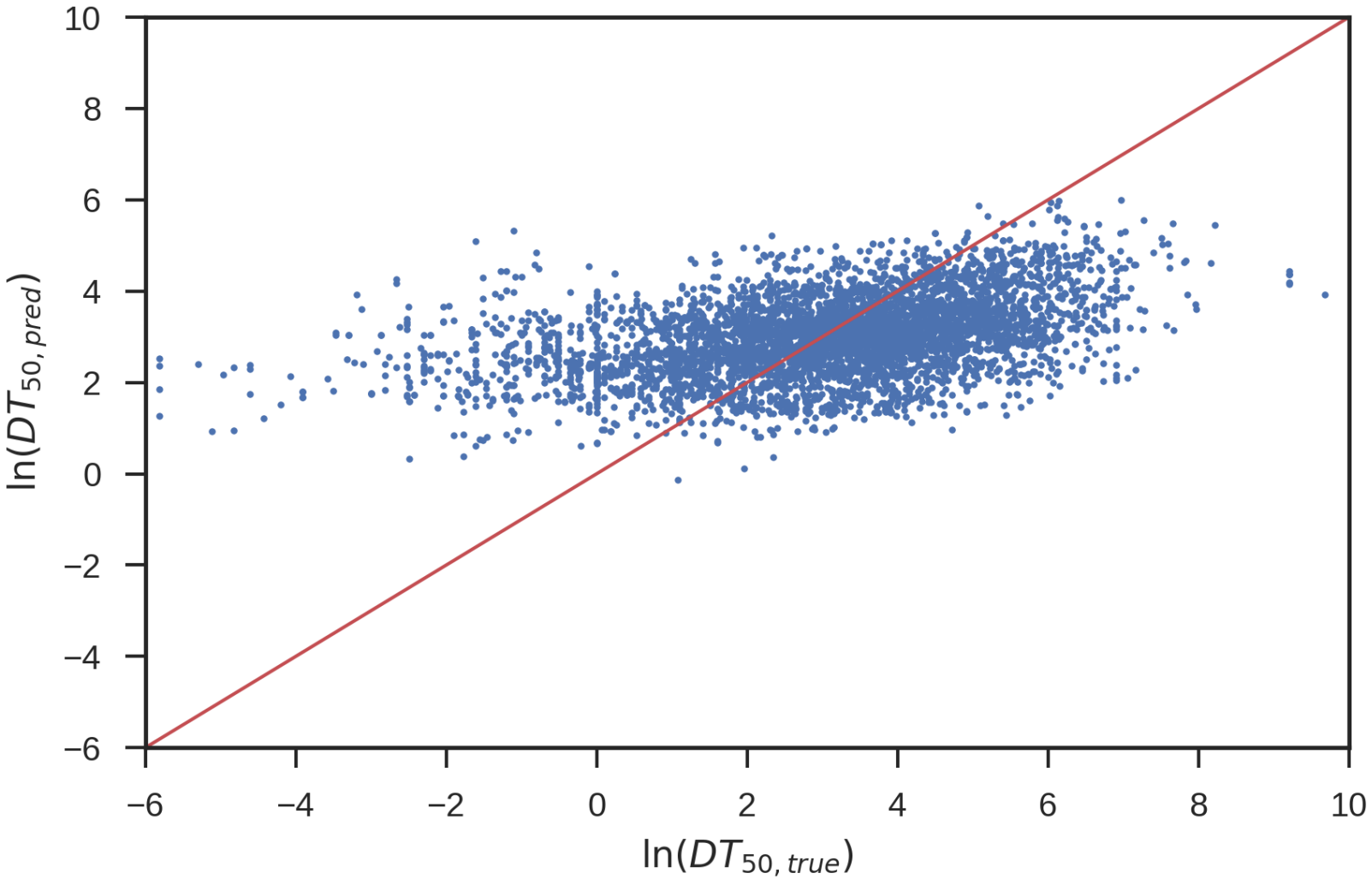
$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2}$$

Interesting values:

- $R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 \iff \hat{\mathbf{y}} = \mathbf{y}$
- $R^2(\mathbf{y}, \hat{\mathbf{y}}) = 0 \Leftarrow \hat{y}_i = \overline{y}, \quad 1 \leq i \leq n$

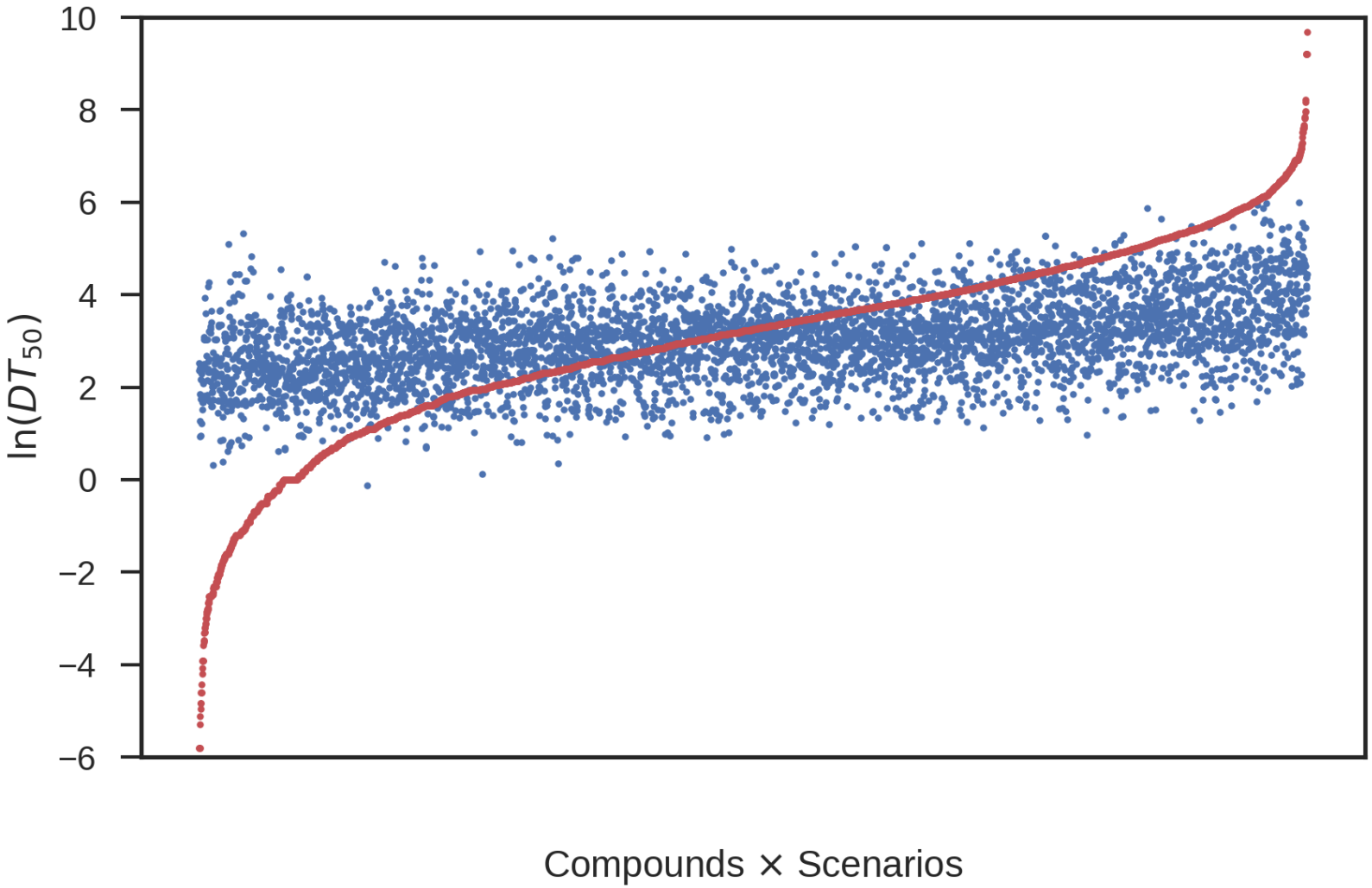
Results: Baseline Results

Random Forest: Predicted vs. True



Results: Baseline Results

Random Forest: Predictions



Results: Preprocessing Configurations

Fingerprinter

Testing different Fingerprinter from *Open Babel*[3]

- *MACCS*: 166 queries, "Are there fewer than 3 oxygens"
- *FP2*: 1022 queries, locates molecule fragments in linear segments of up to 7 atoms
- *FP3*: 56 SMARTS patterns
- *FP4*: 308 SMARTS patterns

[3] O'Boyle *et al.*, "Open Babel: An open chemical toolbox".

Results: Preprocessing Configurations

Fingerprinter

Testing different Fingerprinter from *Open Babel*[3]

- *MACCS*: 166 queries, "Are there fewer than 3 oxygens"
- *FP2*: 1022 queries, locates molecule fragments in linear segments of up to 7 atoms
- *FP3*: 56 SMARTS patterns
- *FP4*: 308 SMARTS patterns

Model	MACCS	FP2	FP3	FP4
Random Forest	0.182	0.181	0.001	0.143
SVR rbf	0.176	0.052	-0.177	0.096
Bagged SVR rbf	**0.185**	0.070	-0.086	0.126

[3] O'Boyle *et al.*, "Open Babel: An open chemical toolbox".

Results: Preprocessing Configurations

Fingerprinter

Testing different Fingerprinter from *Open Babel*[3]

- *MACCS*: 166 queries, "Are there fewer than 3 oxygens"
- *FP2*: 1022 queries, locates molecule fragments in linear segments of up to 7 atoms
- *FP3*: 56 SMARTS patterns
- *FP4*: 308 SMARTS patterns

Model	MACCS	FP2	FP3	FP4
Random Forest	0.182	0.181	0.001	0.143
SVR rbf	0.176	0.052	-0.177	0.096
Bagged SVR rbf	**0.185**	0.070	-0.086	0.126

- Advanced Fingerprinter do not incorporate more structural information that correlate with the DT_{50}

[3] O'Boyle *et al.*, "Open Babel: An open chemical toolbox".

Results: Preprocessing Configurations

Imputation Methods

Model	Mean	KNN	MICE	SoftImpute	Matrix Fact.
Random Forest	0.182	0.174	0.168	0.168	0.160
SVR rbf	0.176	0.174	0.178	0.166	0.166
Bagged SVR rbf	**0.185**	0.183	**0.185**	0.173	0.173

- Advanced methods do not provide better imputation, regarding the DT_{50} values

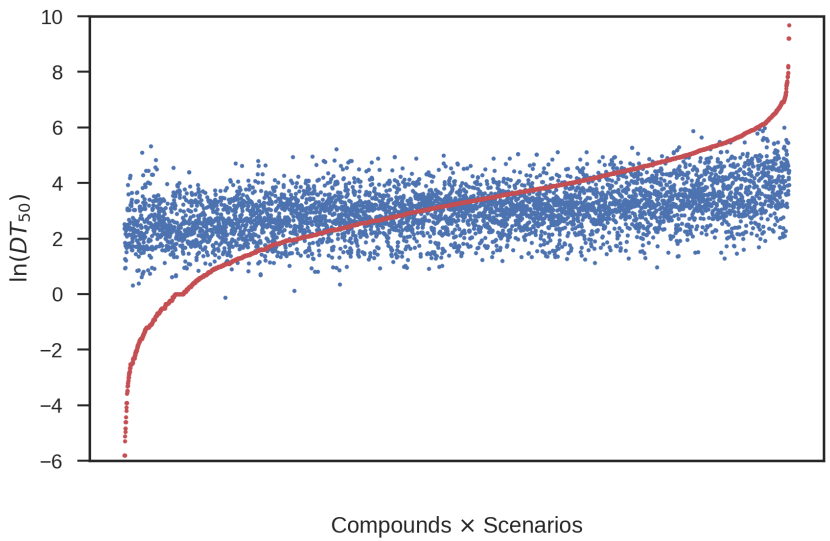
Results: Outlier Removal

Drop datapoints in two different schemes:

Results: Outlier Removal

Drop datapoints in two different schemes:

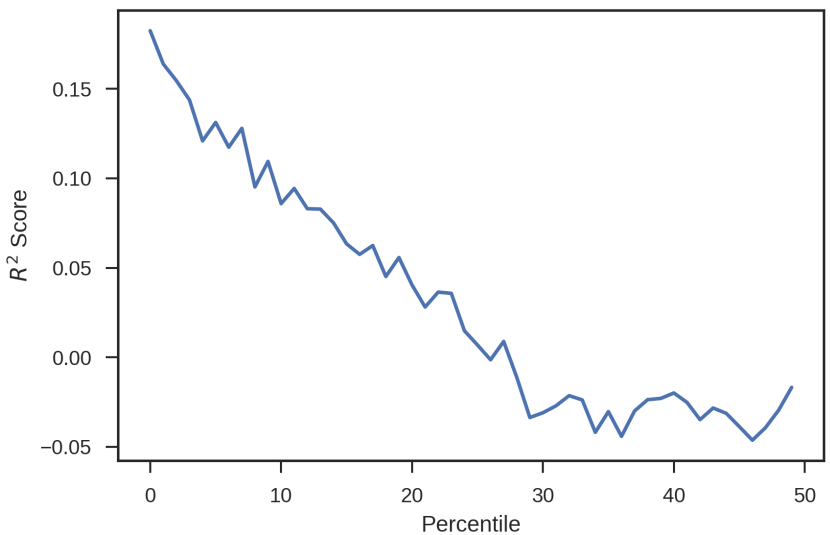
1. Half-lives of extraordinary length (short/long)



Results: Outlier Removal

Drop datapoints in two different schemes:

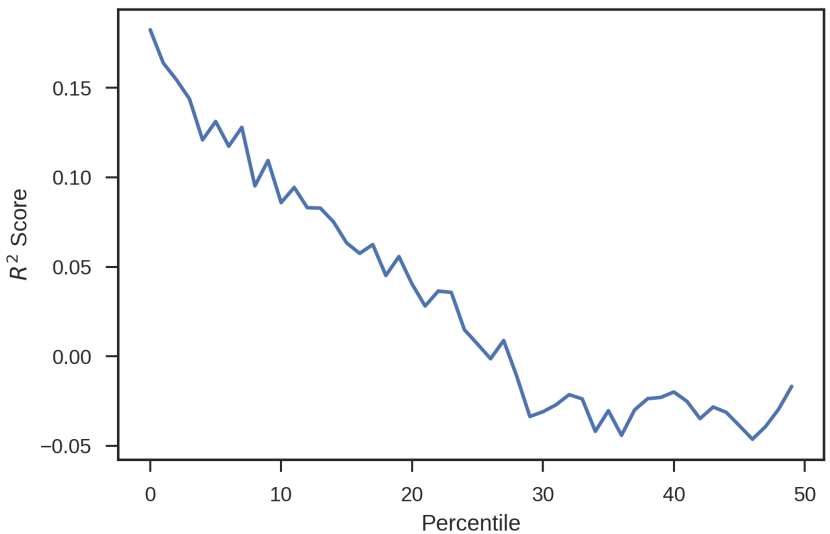
1. Half-lives of extraordinary length (short/long)

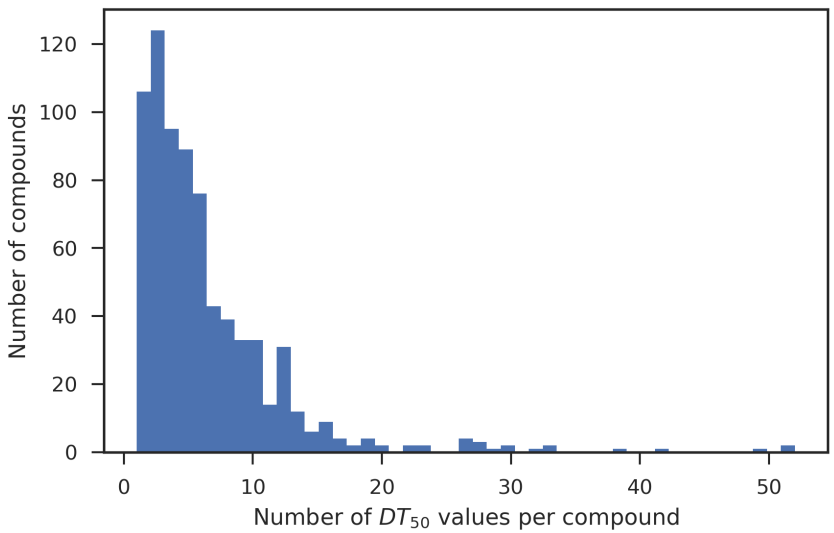


Results: Outlier Removal

Drop datapoints in two different schemes:

1. Half-lives of extraordinary length (short/long)
2. Compounds with too few annotated scenarios

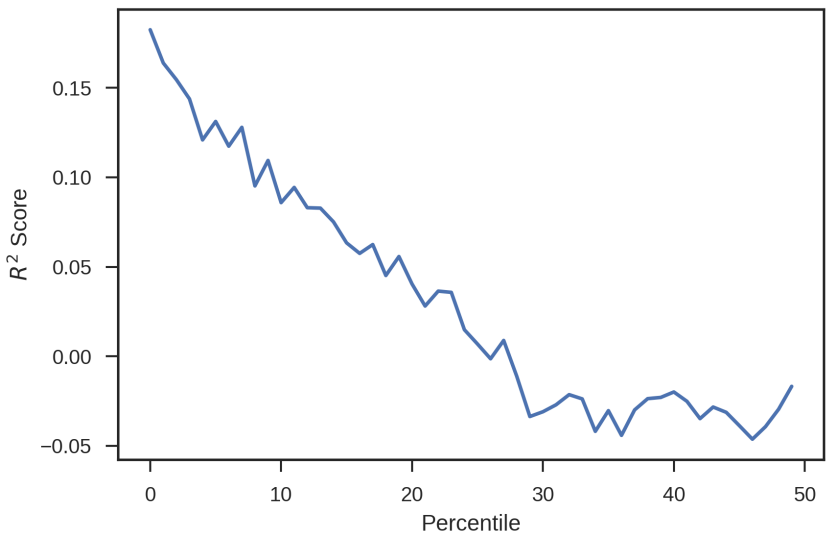


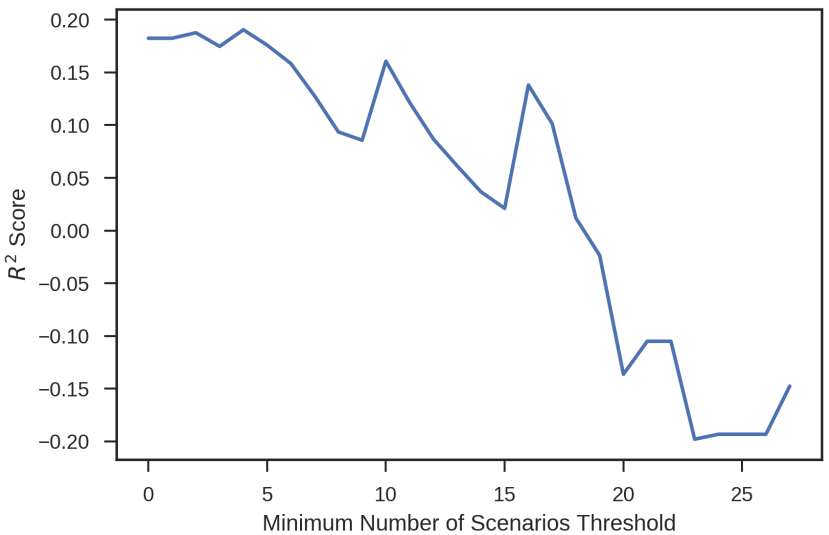


Results: Outlier Removal

Drop datapoints in two different schemes:

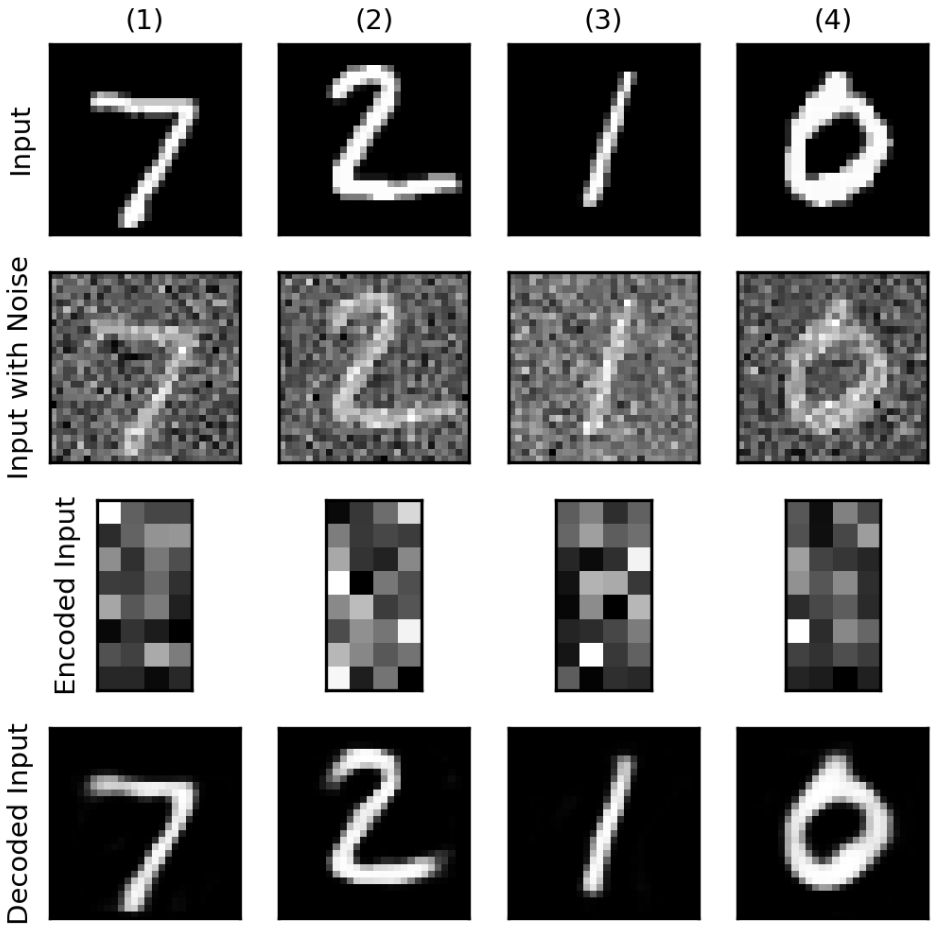
1. Half-lives of extraordinary length (short/long)
2. Compounds with too few annotated scenarios





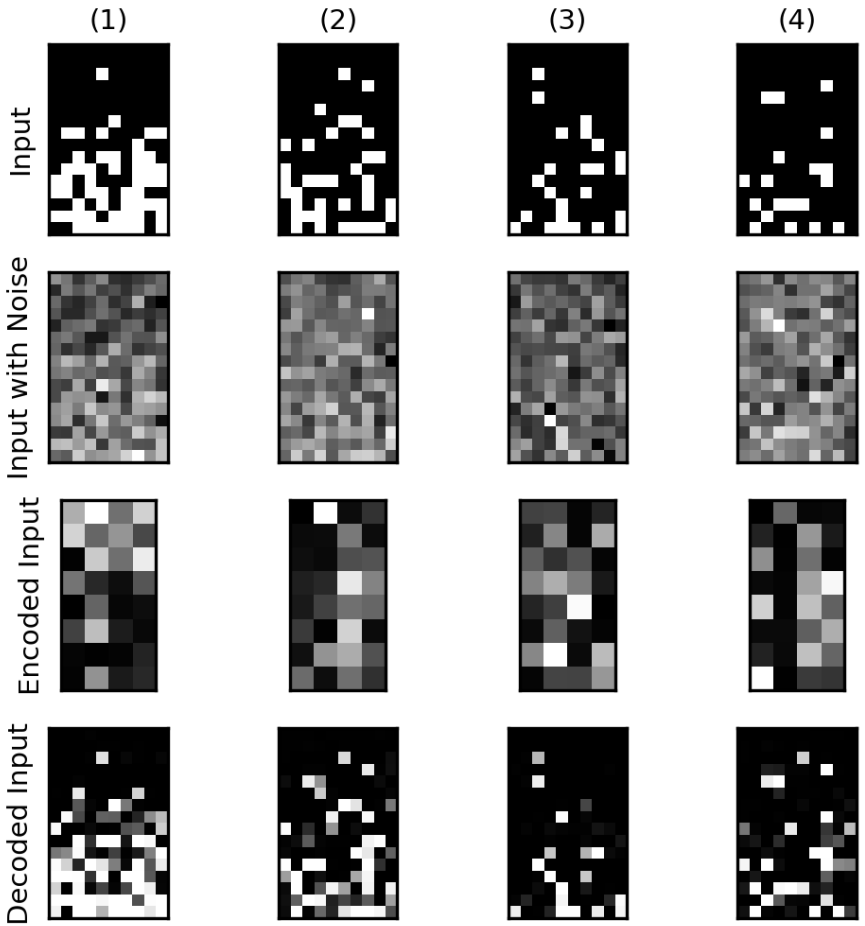
Results: Autoencoder Quality

Autoencoder quality: MNIST



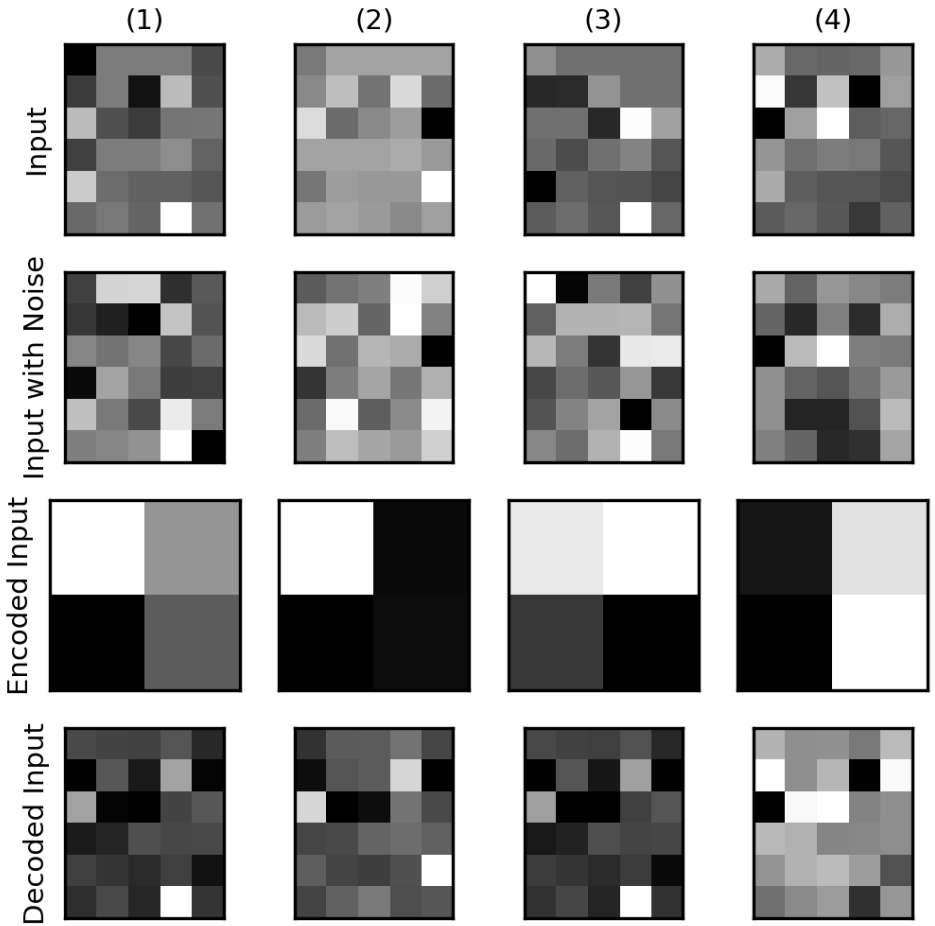
Results: Autoencoder Quality

Autoencoder quality: Compounds



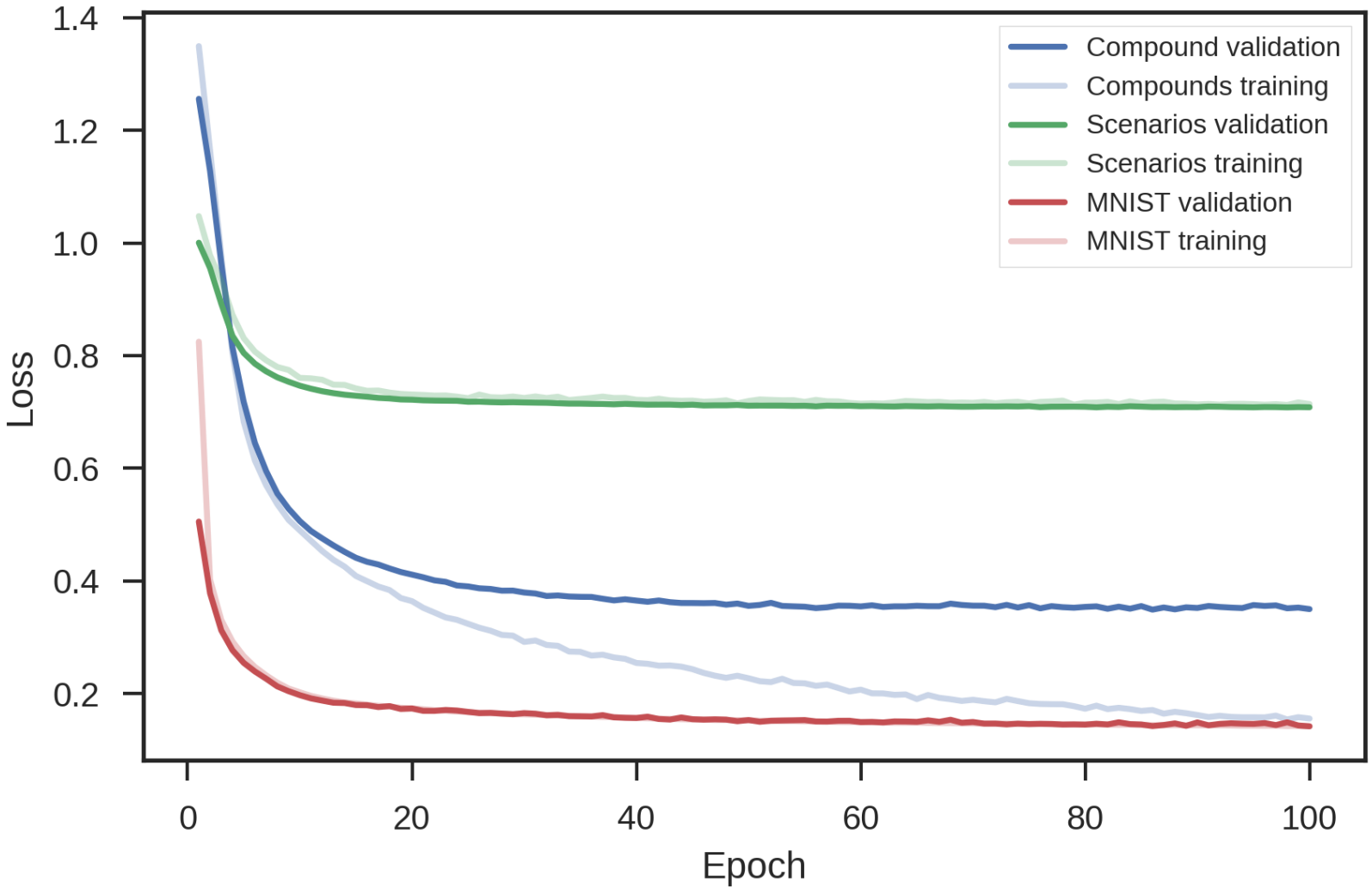
Results: Autoencoder Quality

Autoencoder quality: Scenarios



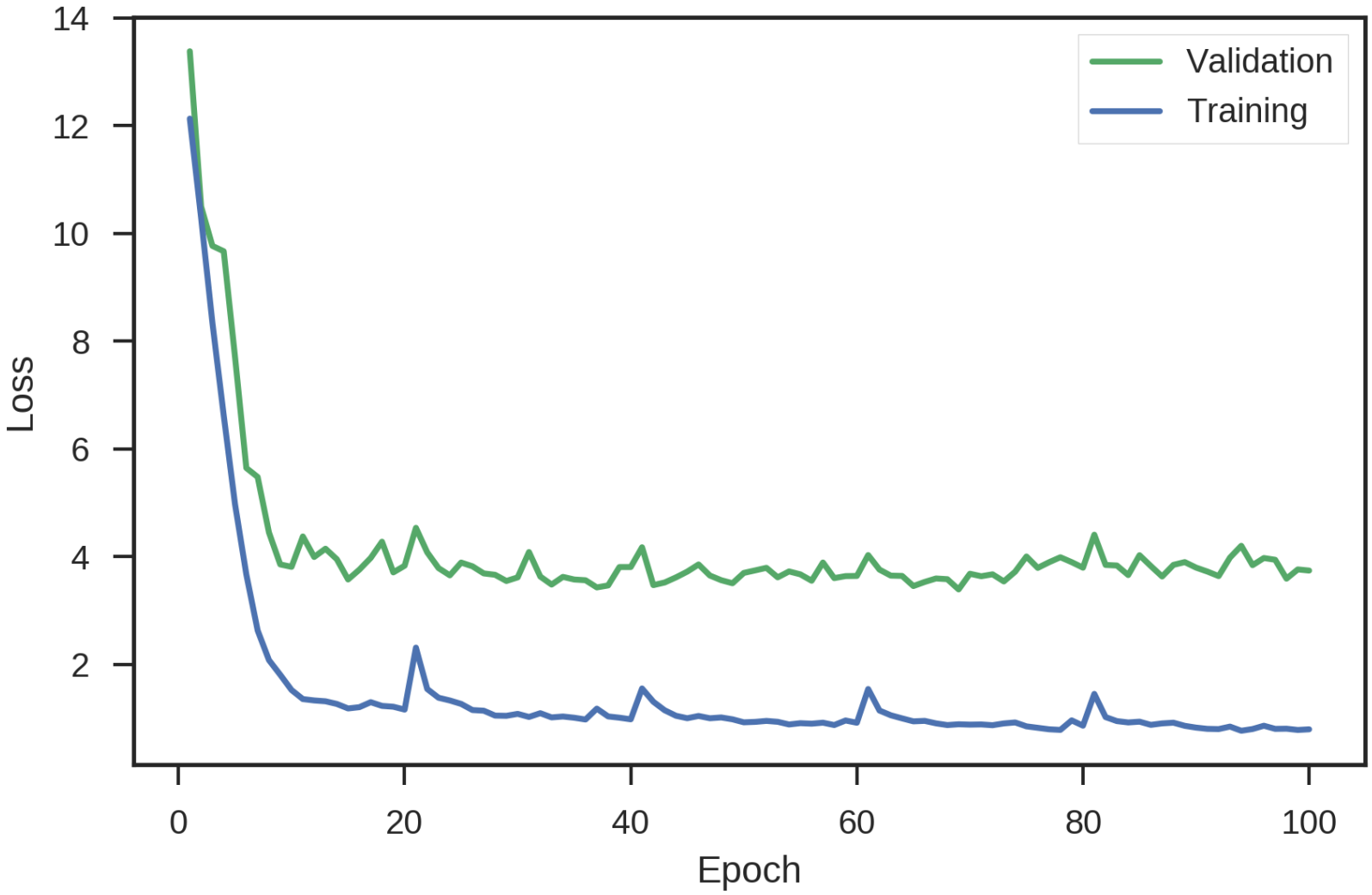
Results: Autoencoder Quality

Autoencoder quality: Loss



Results: DT_{50} Regression

Full DAE regression network run with default parameters



Results: DAE Regression Network Parameters

Encoding Dimension

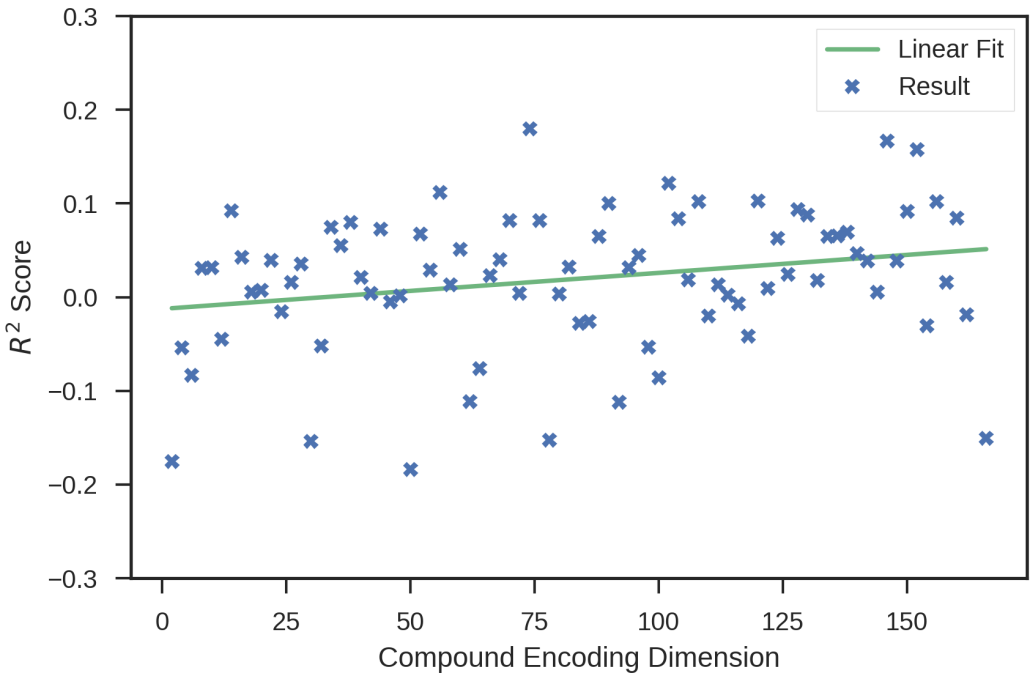
- Lower dimension $\rightsquigarrow$ more dense representation, deeper network

Results: DAE Regression Network Parameters

Encoding Dimension

- Lower dimension $\rightsquigarrow$ more dense representation, deeper network

- Compounds: Slight slope towards higher encoding dimension $\Rightarrow$ Encoding does either not, or negatively impact DT_{50} predictions



Results: DAE Regression Network Parameters

Noise Factor

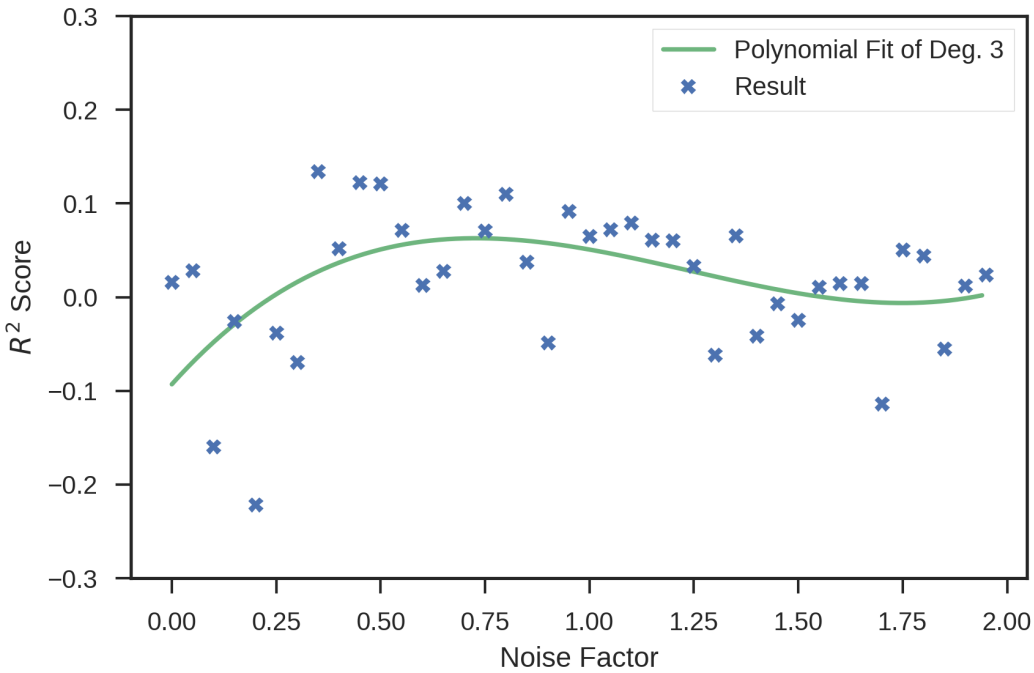
- Scale of random noise added to the input

Results: DAE Regression Network Parameters

Noise Factor

- Scale of random noise added to the input

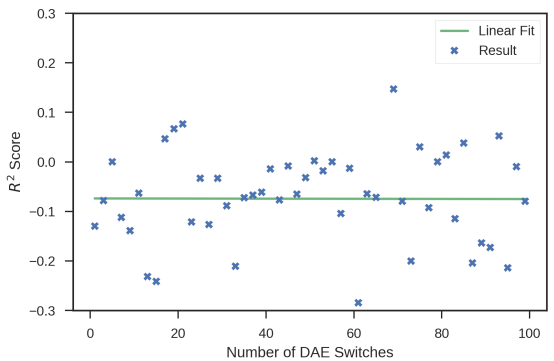
- Peak at ~ 0.5
- Below: No effect of DAE's robust feature generation
- Above: Features get lost in noise



Results: DAE Regression Network Parameters

No clear influence for:

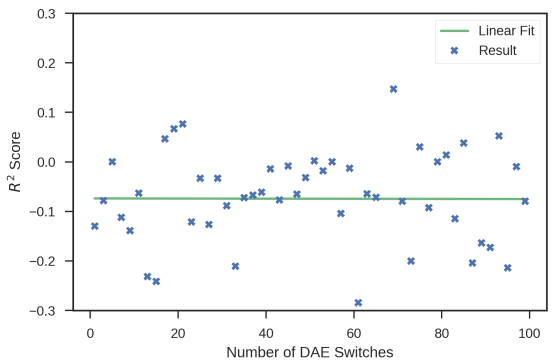
- Switch Rate

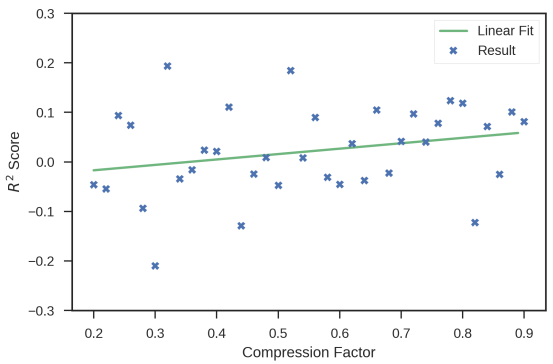


Results: DAE Regression Network Parameters

No clear influence for:

- Switch Rate
- Compression Factor

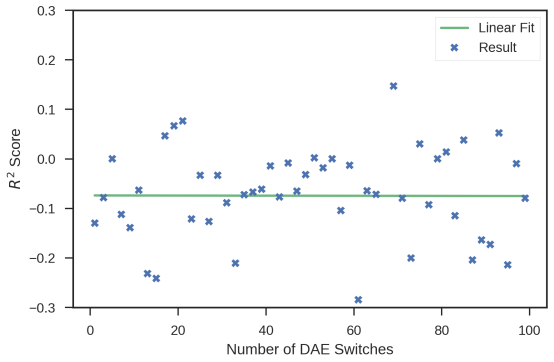


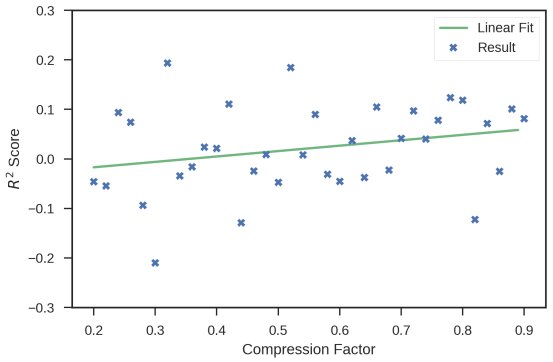


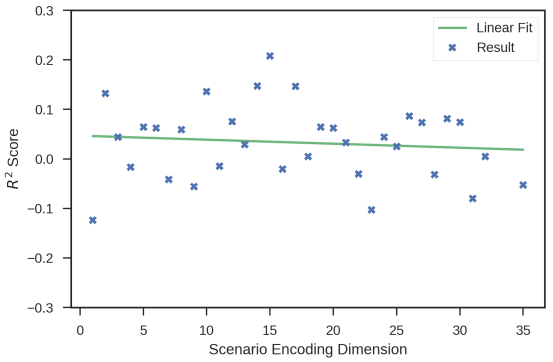
Results: DAE Regression Network Parameters

No clear influence for:

- Switch Rate
- Compression Factor
- Scenario Encoding Dimension







Results: Permutation based P-Value Tests

Why is nothing *really* working?

[4] Ojala and Garriga, "Permutation Tests for Studying Classifier Performance".

Results: Permutation based P-Value Tests

Why is nothing *really* working?

Permutation based *p*-value tests[4]

- **Test 1:** Permute labels **y**
  - Did the model find a connection between **X** and **y**?
- **Test 2:** Permute data columns per class
  - Does there exist a dependency between features of **X** that may reduce the error score?

[4] Ojala and Garriga, "Permutation Tests for Studying Classifier Performance".

Results: Permutation based P-Value Tests

Why is nothing *really* working?

Permutation based *p*-value tests[4]

- **Test 1:** Permute labels **y**
  - Did the model find a connection between **X** and **y**?
- **Test 2:** Permute data columns per class
  - Does there exist a dependency between features of **X** that may reduce the error score?

$$p = \frac{\left|\left\{D' \in \hat{D} : e\left(f, D'\right) \leq e\left(f, D\right)\right\}\right| + 1}{\left|\hat{D}\right| + 1}$$

Significance threshold $\alpha = 0.05$

(likelihood of the model achieving the error $e(f, D)$ by chance)

[4] Ojala and Garriga, "Permutation Tests for Studying Classifier Performance".

Results: Permutation based P-Value Tests

Setup: Random Forest, 100 permutations, $e(f, D) = 0.182$

Results: Permutation based P-Value Tests

Setup: Random Forest, 100 permutations, $e(f, D) = 0.182$

Test	Features	p-value	R^2 mean
1	all	0.010	-0.043

- Test 1: No permutation achieved a better results $\Rightarrow$ Significant dependency explored

Results: Permutation based P-Value Tests

Setup: Random Forest, 100 permutations, $e(f, D) = 0.182$

Test	Features	p-value	R^2 mean
1	all	0.010	-0.043
2	all	0.089	0.157

- Test 1: No permutation achieved a better results $\Rightarrow$ Significant dependency explored
- Test 2 all: Only slightly worse than $e(f, D) \Rightarrow$ No significant dependency between features of the same class detected by the model

Results: Permutation based P-Value Tests

Setup: Random Forest, 100 permutations, $e(f, D) = 0.182$

Test	Features	p-value	R^2 mean
1	all	0.010	-0.043
2	all	0.089	0.157
2	compound	0.020	0.149

- Test 1: No permutation achieved a better results $\Rightarrow$ Significant dependency explored
- Test 2 all: Only slightly worse than $e(f, D) \Rightarrow$ No significant dependency between features of the same class detected by the model
- Test 2 compound: $p = 0.020$ significant, $R^2 = 0.149$ only slight margin to original dataset

Results: Permutation based P-Value Tests

Setup: Random Forest, 100 permutations, $e(f, D) = 0.182$

Test	Features	p-value	R^2 mean
1	all	0.010	-0.043
2	all	0.089	0.157
2	compound	0.020	0.149
2	scenario	0.525	0.171

- Test 1: No permutation achieved a better results $\Rightarrow$ Significant dependency explored
- Test 2 all: Only slightly worse than $e(f, D)$ $\Rightarrow$ No significant dependency between features of the same class detected by the model
- Test 2 compound: $p = 0.020$ significant, $R^2 = 0.149$ only slight margin to original dataset
- Test 2 scenario: Randomly performs better than the original dataset $\Rightarrow$ Scenarios are of no help to detect any dependency at all

Conclusion

- Baseline models revealed modest dependencies, merely capable of a better regression than mean-value predictor

Conclusion

- Baseline models revealed modest dependencies, merely capable of a better regression than mean-value predictor
- Autoencoder worked well for compounds and bad for scenarios

Conclusion

- Baseline models revealed modest dependencies, merely capable of a better regression than mean-value predictor
- Autoencoder worked well for compounds and bad for scenarios
- DAE regression network parameter evaluation is hard to interpret, due to high variance of the results

Conclusion

- Baseline models revealed modest dependencies, merely capable of a better regression than mean-value predictor
- Autoencoder worked well for compounds and bad for scenarios
- DAE regression network parameter evaluation is hard to interpret, due to high variance of the results
- *P*-value tests revealed that scenario data was of no help in dependency exploration

Conclusion

Possible sources for the bad evaluation results:

1. Dataset contains an insufficient number of instances, represents only a small fraction of the patterns
   - See performance loss while removing data

[5] Zhang *et al.*, "Understanding deep learning requires rethinking generalization".

Conclusion

Possible sources for the bad evaluation results:

1. Dataset contains an insufficient number of instances, represents only a small fraction of the patterns
   - See performance loss while removing data
2. Dataset lacks in informative features
   - See *p*-value test results

[5] Zhang *et al.*, "Understanding deep learning requires rethinking generalization".

Conclusion

Possible sources for the bad evaluation results:

1. Dataset contains an insufficient number of instances, represents only a small fraction of the patterns
   - See performance loss while removing data
2. Dataset lacks in informative features
   - See *p*-value test results
3. Network weights underdetermination
   - See Zhang et al.[5]

[5] Zhang *et al.*, "Understanding deep learning requires rethinking generalization".

Outlook

What next?

- Capturing more features for existing studies is impossible

Outlook

What next?

- Capturing more features for existing studies is impossible
- Measuring more datapoints

Outlook

What next?

- Capturing more features for existing studies is impossible
- Measuring more datapoints
- Improve autoencoder performance by training on further datasets that are not connected to half-lives

- Thank you for your attention!
- Any questions?