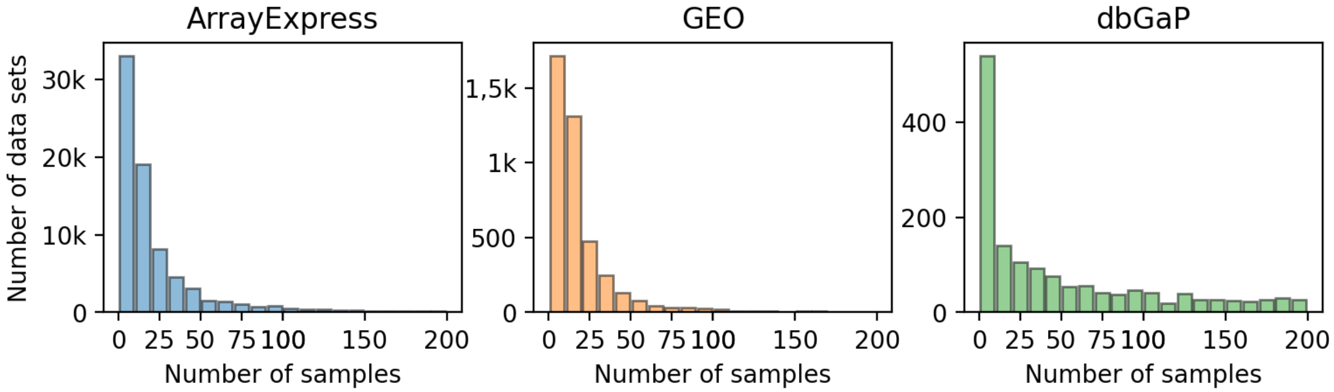
Transfer Learning for Phenotype Prediction from Small Gene Expression Data Sets

DOMEN MOHORČIČ

MENTOR: PROF. DR. BLAŽ ZUPAN

Problem definition

- Biomedical research produces huge amounts of data, analysed using machine learning, and is the foundation for personalised medicine.
- Many data sets have small sample sizes and numerous features, which makes learning difficult (Progeria, 144 patients). This is called the small data set problem.
- Solved by repeating samples, interpolating between samples, synthetic generation of new samples…

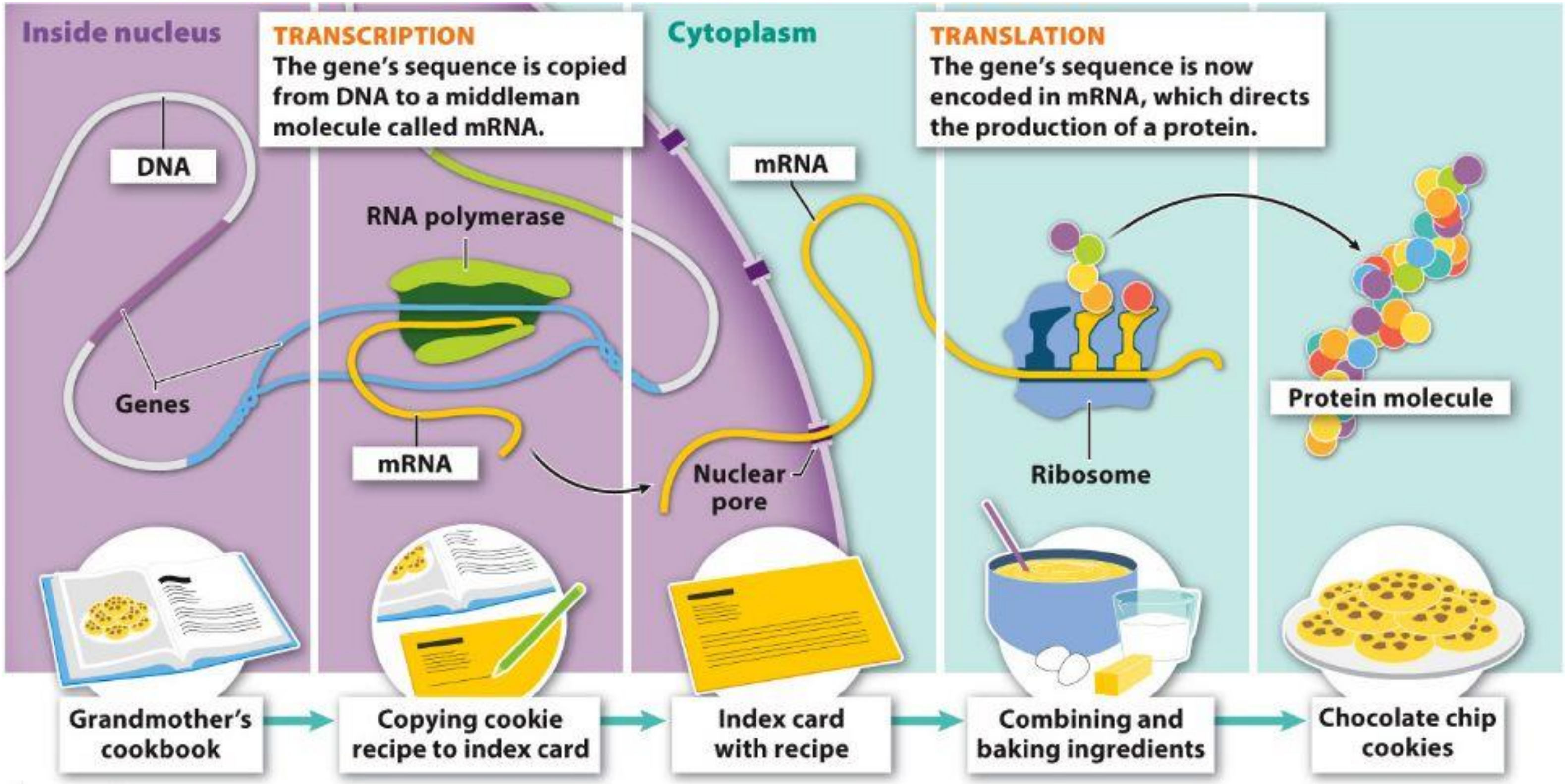


Proposed solution

1. Combine many small gene expression data sets into a larger one,
2. train a model, capable of producing informative gene expression encodings,
3. test the model by encoding unseen data and train new models for phenotype prediction.

From genotype to phenotype

We focused on phenotype prediction from small gene expression data sets.



Transfer learning using gene expressions

- Neural networks: gene expression embeddings for cancer data clustering (Choi et al., Frontiers in Genetics 2019).
- gene2vec model for gene embeddings, trained on gene co-expression pairs, tested on inference of interaction maps from gene names (Du et al., BMC Genomics 2019).
- predicted corn nitrogen use efficiency from model organism with boosting and tree-based methods (Cheng et al., Nature Communications 2021),
- scDEAL predicted cancer drug responses to DNA sequences with gene expressions as intermediate step (Chen et al., Nature Communications 2022),
- predicted cancer types from gene expressions and DNA sequences with focus on the performance of models based on hyperparameters (Henczar et al., BMC Bioinformatics 2022).

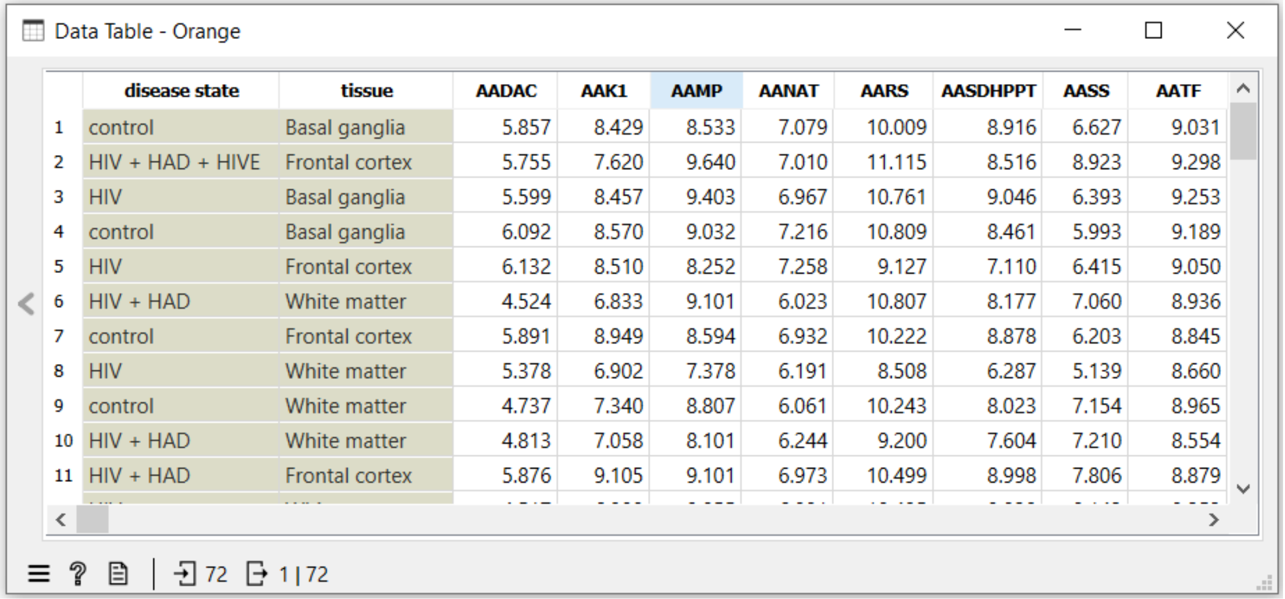
Data selection

We used the Gene Expression Omnibus database (GEO) and selected data sets with human data that have already been processed.

We used genes from L1000 method for feature selection and manually inspected all data sets for potential target variables from sample annotations.

Summary:

- 70 data sets (35 / 35),
- 6,713 samples (3,334 / 3,379),
- 884 gene expressions,
- 185 target variables (92 / 93).



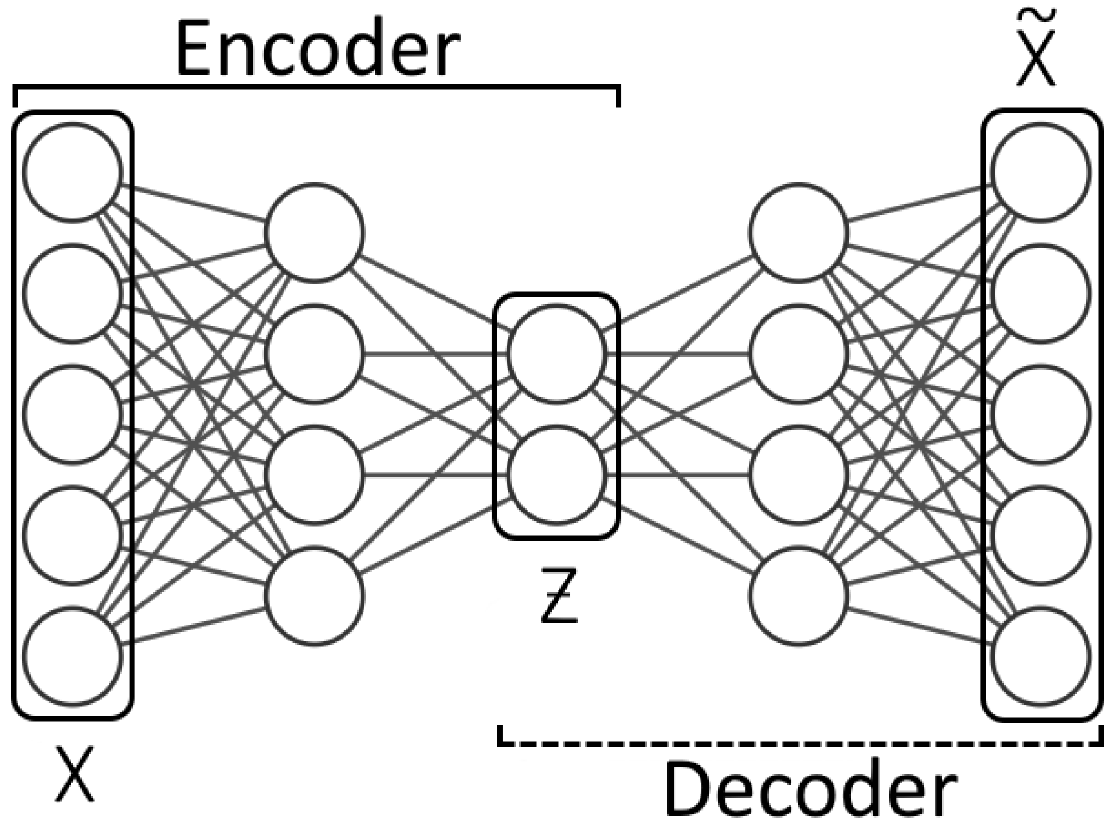
	disease state	tissue	AADAC	AAK1	AAMP	AANAT	AARS	AASDHPPT	AASS	AATF
1	control	Basal ganglia	5.857	8.429	8.533	7.079	10.009	8.916	6.627	9.031
2	HIV + HAD + HIVE	Frontal cortex	5.755	7.620	9.640	7.010	11.115	8.516	8.923	9.298
3	HIV	Basal ganglia	5.599	8.457	9.403	6.967	10.761	9.046	6.393	9.253
4	control	Basal ganglia	6.092	8.570	9.032	7.216	10.809	8.461	5.993	9.189
5	HIV	Frontal cortex	6.132	8.510	8.252	7.258	9.127	7.110	6.415	9.050
6	HIV + HAD	White matter	4.524	6.833	9.101	6.023	10.807	8.177	7.060	8.936
7	control	Frontal cortex	5.891	8.949	8.594	6.932	10.222	8.878	6.203	8.845
8	HIV	White matter	5.378	6.902	7.378	6.191	8.508	6.287	5.139	8.660
9	control	White matter	4.737	7.340	8.807	6.061	10.243	8.023	7.154	8.965
10	HIV + HAD	White matter	4.813	7.058	8.101	6.244	9.200	7.604	7.210	8.554
11	HIV + HAD	Frontal cortex	5.876	9.105	9.101	6.973	10.499	8.998	7.806	8.879

Autoencoders

Designed to compress and reconstruct data, they are similar to PCA, LDA, and SVD, but are non-linear.

Problems:

- the latent space Z is not constraint,
- encoder is not aware of any downstream tasks.

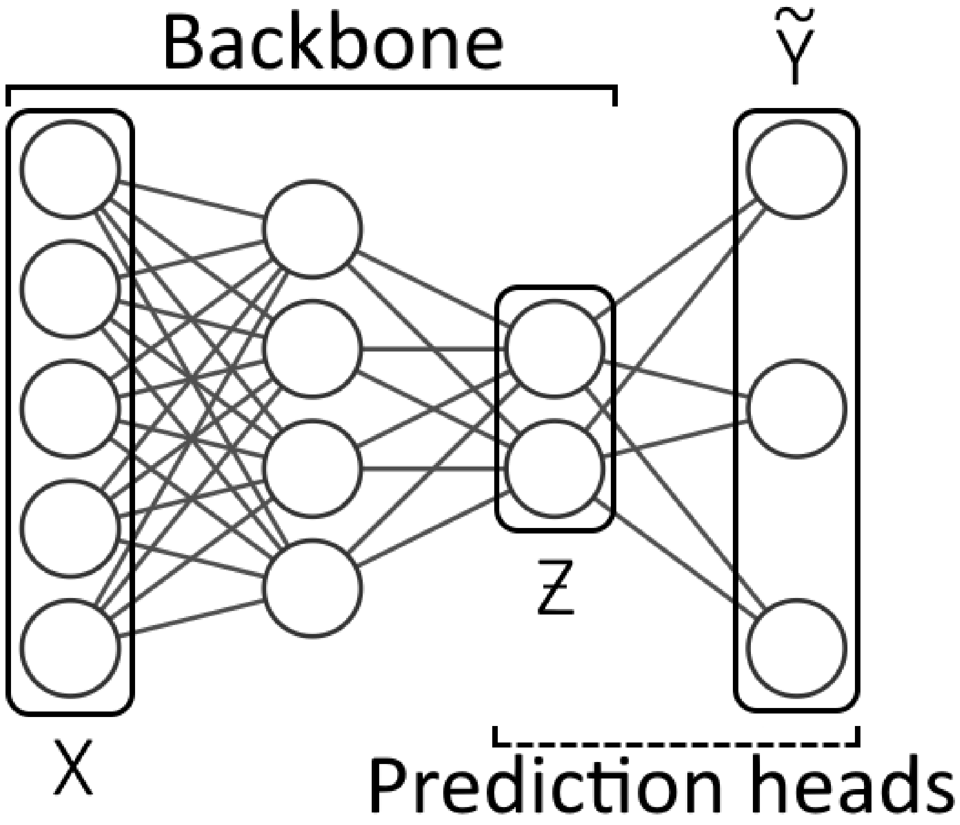


Multi-task models

Designed as universal feature extractors by learning several tasks at the same time.

Problems:

- each prediction head has few samples, which makes learning slow.



Training

- Adam optimizer for parameter updates with early stopping,
- loss functions: MSE and log loss,
- alternate metrics: R^2 and AUC,
- fixed hidden layer size, the size of encoding layer varied from 4 to 64,
- trained 10 models for variance estimation.

Testing

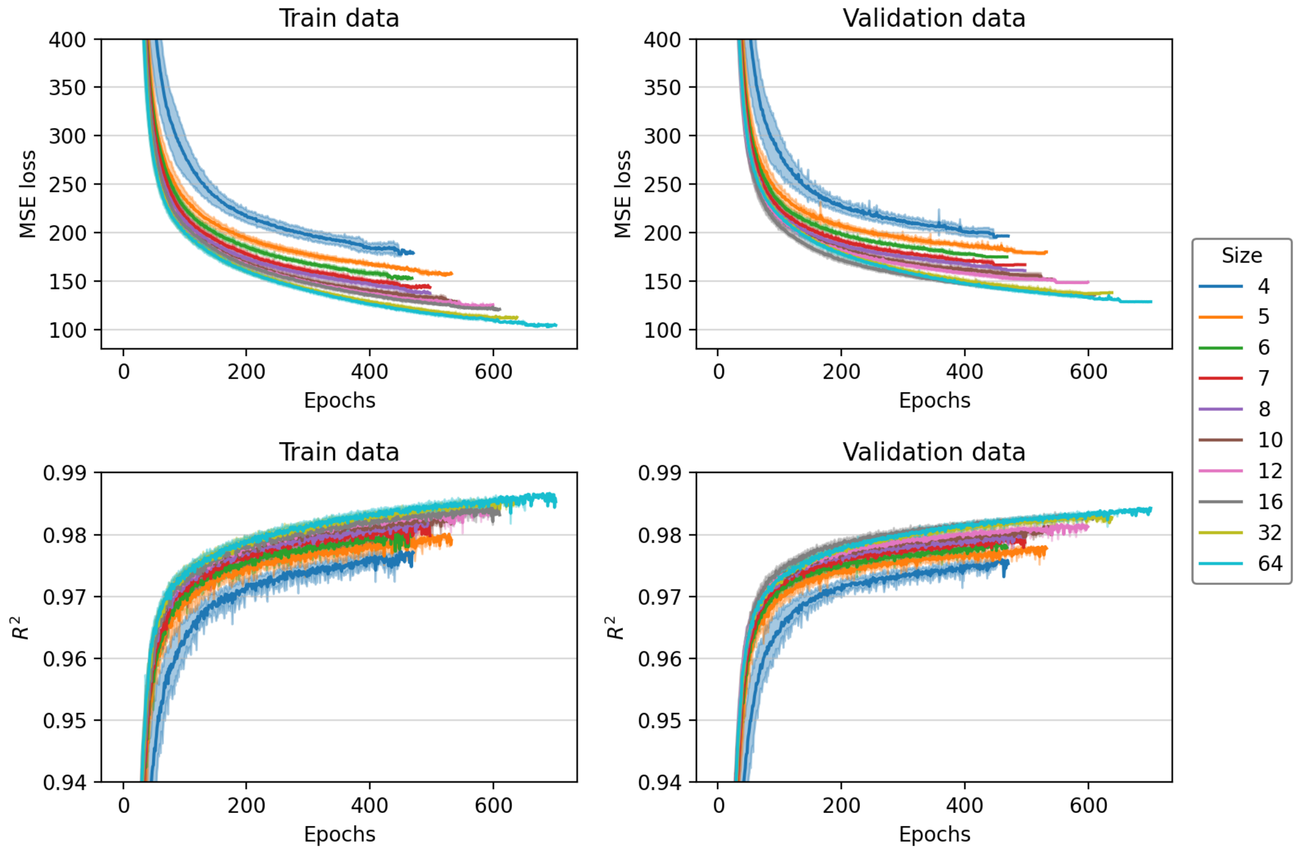
- Used LOOCV with logistic regression for predictions on testing data,
- plotted AUC of testing data against baseline,
- analysed confusion matrix with TPR and TNR,
- inspected PCA projections of encoded data.

Training results: autoencoders

Autoencoders exhibit stable training behaviour.

Best MSE: 64-autoencoder (135.1 ± 3.3)

Best R^2: 64-autoencoder (0.983 ± 0.001)

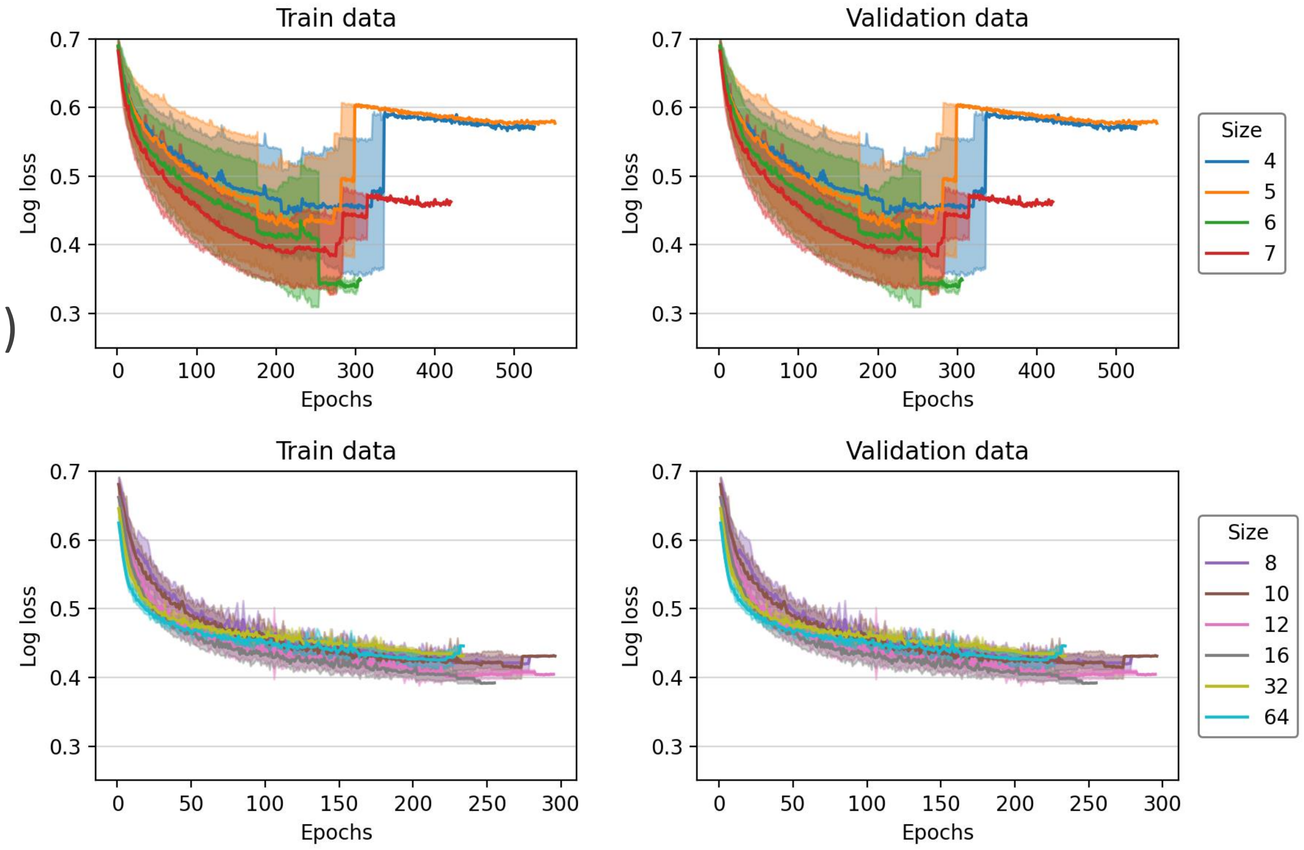


Training results: multi-task models

Multi-task models with smaller latent layer sizes are unstable and difficult to train.

Best logloss: 16-multitask (0.406 ± 0.008)

Best AUC: 64-multitask (0.816 ± 0.010)

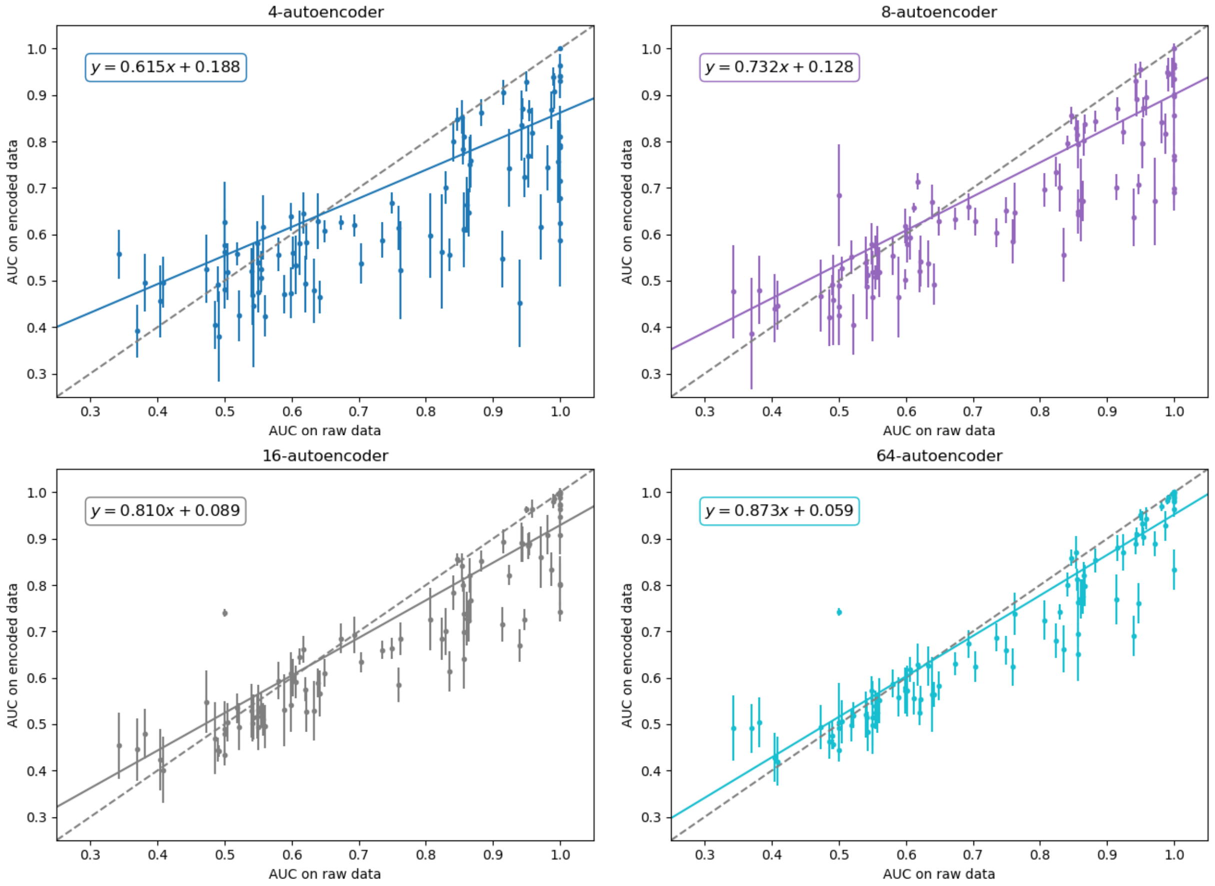


Testing results: AUC-AUC plots

Models with larger latent layers approach baseline, but neither architecture surpasses it.

The performance is improved for lower-performing, while degraded for higher-performing data sets.

Model with closest results to baseline: 64-multitask (y = 0.888x + 0.050)

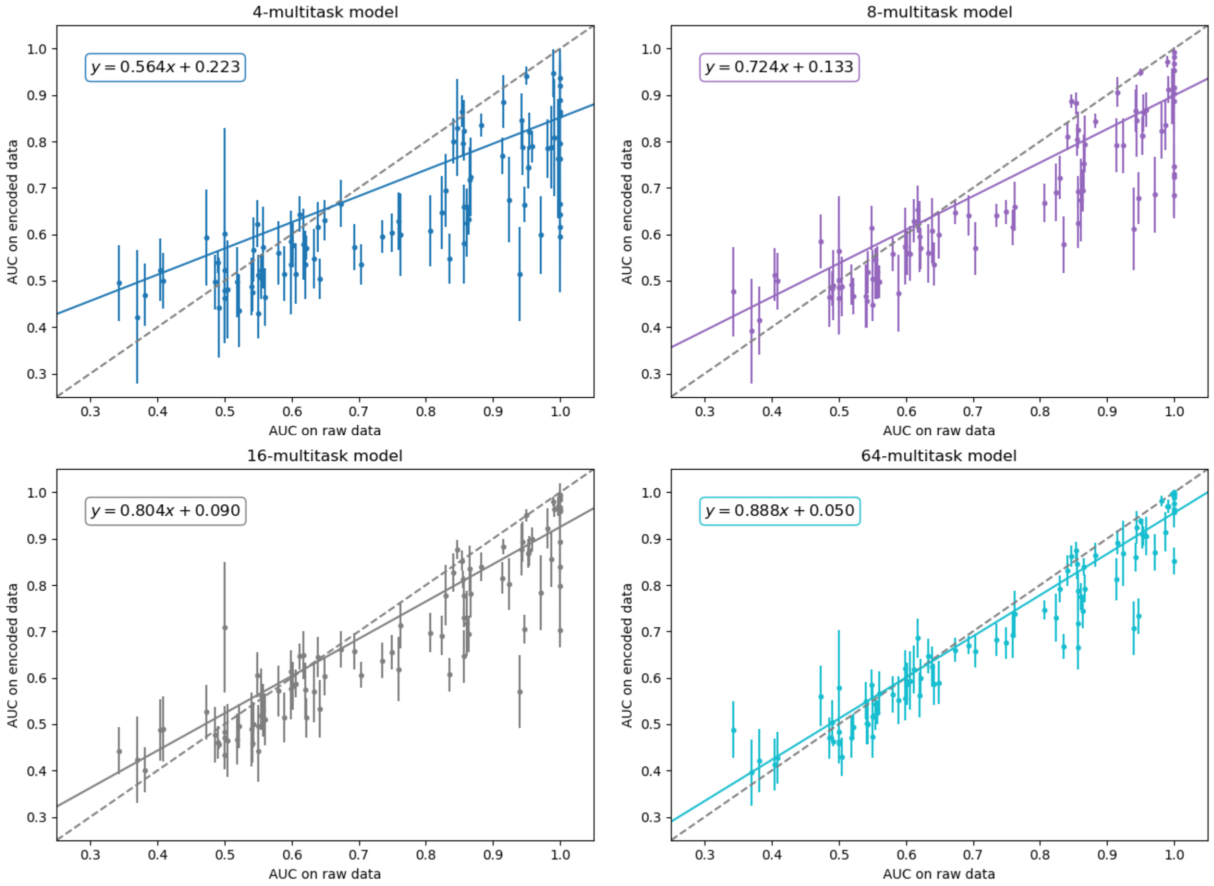


Testing results: AUC-AUC plots

Models with larger latent layers approach baseline, but neither architecture surpasses it.

The performance is improved for lower-performing, while degraded for higher-performing data sets.

Model with closest results to baseline: 64-multitask (y = 0.888x + 0.050)

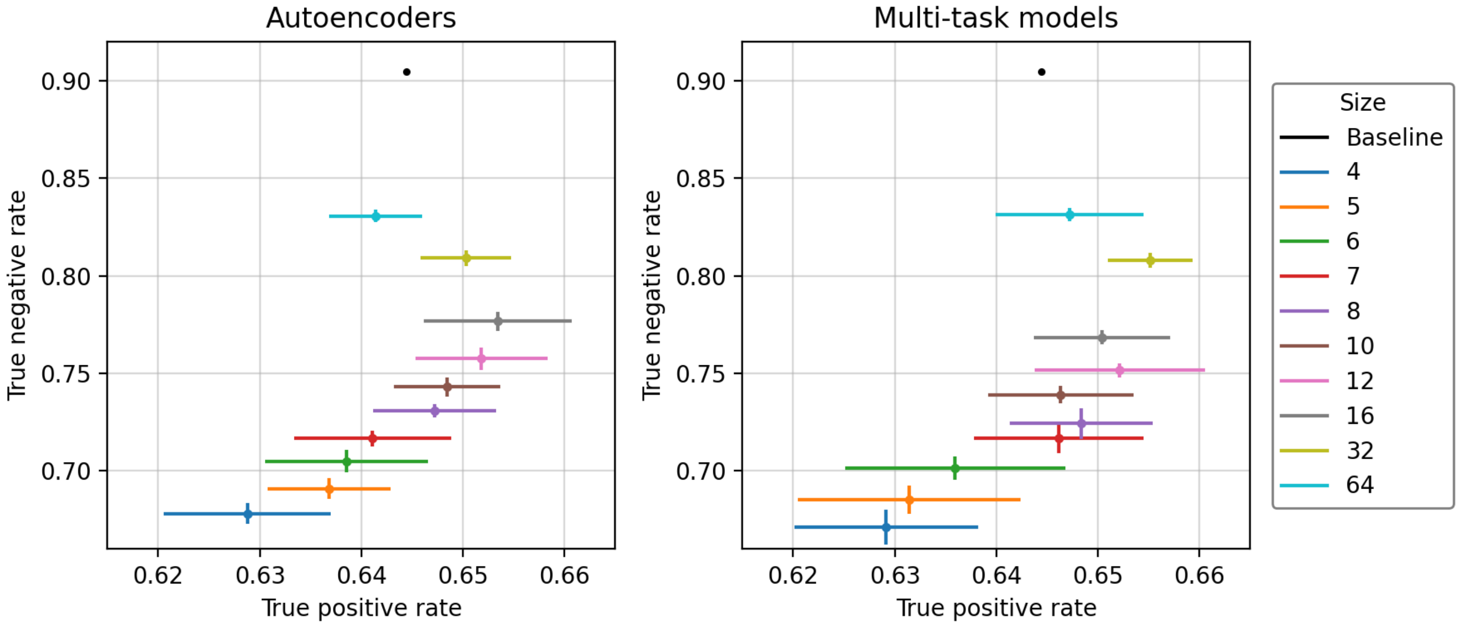


Testing results: TPR-TNR plots

Models with larger latent layers achieve higher TNR, but TPR increases only up to a certain value.

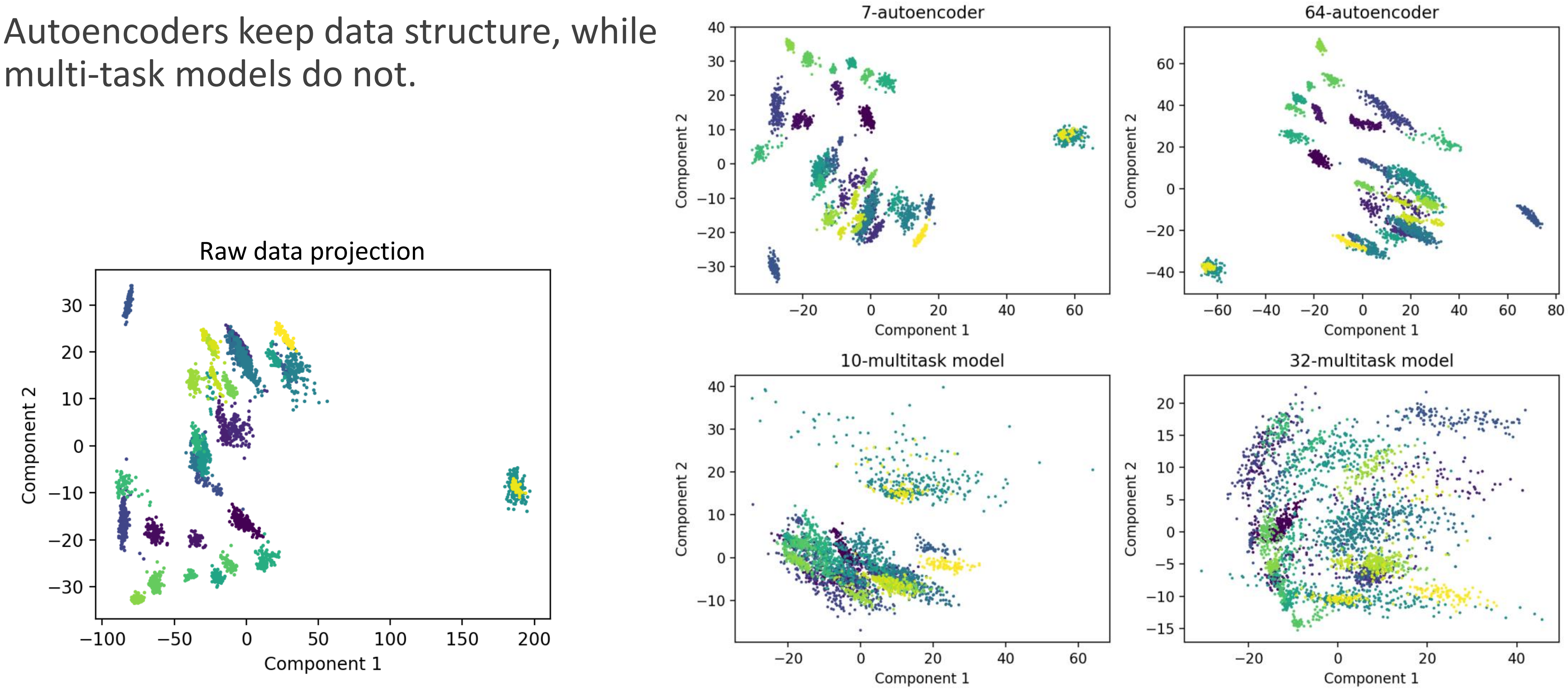
Best TNR: 64-autoencoder and 64-multitask (0.831 ± 0.003)

Best TPR: 32-multitask (0.655 ± 0.004)



Encodings: PCA

Autoencoders keep data structure, while multi-task models do not.



Encodings: explained variance

	Autoencoder		Multi-task model	
Size	Component 1	Component 2	Component 1	Component 2
4	0.522 ± 0.046	0.284 ± 0.050	0.639 ± 0.146	0.258 ± 0.105
5	0.488 ± 0.050	0.310 ± 0.043	0.648 ± 0.178	0.208 ± 0.106
6	0.480 ± 0.020	0.307 ± 0.031	0.595 ± 0.208	0.238 ± 0.132
7	0.486 ± 0.029	0.273 ± 0.018	0.549 ± 0.106	0.235 ± 0.077
8	0.475 ± 0.031	0.282 ± 0.022	0.459 ± 0.041	0.251 ± 0.040
10	0.462 ± 0.032	0.272 ± 0.035	0.540 ± 0.094	0.240 ± 0.076
12	0.432 ± 0.022	0.286 ± 0.019	0.443 ± 0.070	0.257 ± 0.053
16	0.433 ± 0.017	0.282 ± 0.030	0.440 ± 0.091	0.221 ± 0.057
32	0.388 ± 0.009	0.298 ± 0.021	0.518 ± 0.072	0.230 ± 0.029
64	0.359 ± 0.017	0.299 ± 0.016	0.510 ± 0.047	0.256 ± 0.027
raw	0.808	0.051	0.808	0.051

Conclusion

A novel approach to small data set problem by combining many small data sets, training an encoder, and using transfer learning on unseen data sets.

Two architectures: autoencoders and multi-task models.

Autoencoders were easier to train, but multi-task models yielded better results on unseen data. Autoencoders kept the data structure while multi-task models did not.

Future improvements:

- better and more informed data set selection,
- improved analysis of embedding space.

Appendix: data sets table

Name	N	A
GDS4404	50	2
GDS3829	50	3
GDS5074	52	1
GDS4167	52	1
GDS4299	52	1
GDS4513	53	1
GDS4906	54	3
GDS4896	54	3
GDS4412	56	3
GDS2643	56	5
GDS3459	56	4
GDS5093	56	3
GDS4266	58	3
GDS3627	58	1
GDS4607	60	2
GDS4056	61	5
GDS4379	62	5
GDS4176	62	3

Name	N	A
GDS3057	64	3
GDS5083	64	1
GDS4381	64	1
GDS4587	66	1
GDS4198	70	3
GDS5205	70	1
GDS4358	72	6
GDS4471	76	5
GDS4282	76	5
GDS4103	78	1
GDS4758	79	3
GDS3329	79	1
GDS4181	80	1
GDS4975	81	1
GDS3539	82	4
GDS5277	86	3
GDS4088	86	1
GDS4837	88	2

Name	N	A
GDS4336	90	2
GDS4761	91	7
GDS3885	92	3
GDS4456	93	1
GDS4182	96	1
GDS4562	96	2
GDS4968	99	1
GDS4273	103	2
GDS4057	103	5
GDS4130	104	2
GDS4516	104	5
GDS3257	107	4
GDS4318	108	3
GDS2767	108	4
GDS5037	108	4
GDS4549	116	3
GDS4129	120	1
GDS3837	120	1

Name	N	A
GDS5393	120	3
GDS4222	130	5
GDS4274	130	1
GDS5000	131	2
GDS5363	139	2
GDS5499	140	3
GDS4267	154	3
GDS4278	154	1
GDS5027	156	3
GDS3952	162	3
GDS3312	163	1
GDS4600	170	1
GDS4296	174	9
GDS4602	180	1
GDS2771	192	1
GDS4206	197	3

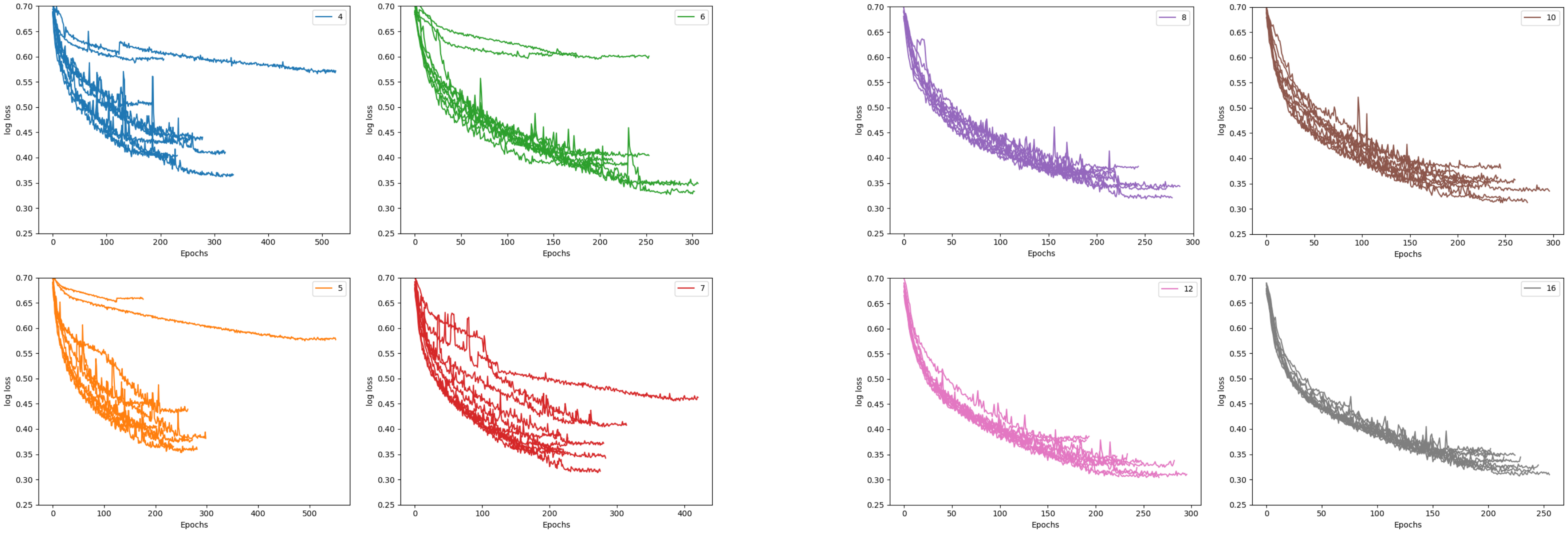
Appendix: autoencoder training results

	MSE		R^2	
Size	Train	Validation	Train	Validation
4	186.7 ± 6.7	202.1 ± 5.2	0.9757 ± 0.0012	0.9747 ± 0.0009
5	164.6 ± 5.4	183.9 ± 3.9	0.9785 ± 0.0008	0.9769 ± 0.0007
6	159.2 ± 4.3	178.1 ± 2.1	0.9790 ± 0.0016	0.9777 ± 0.0003
7	151.1 ± 5.5	171.2 ± 3.4	0.9802 ± 0.0006	0.9784 ± 0.0006
8	145.4 ± 4.7	165.6 ± 3.3	0.9809 ± 0.0008	0.9792 ± 0.0005
10	136.6 ± 4.7	158.5 ± 3.5	0.9820 ± 0.0010	0.9800 ± 0.0003
12	130.4 ± 4.4	151.9 ± 3.0	0.9824 ± 0.0010	0.9808 ± 0.0006
16	125.2 ± 3.5	138.5 ± 2.6	0.9832 ± 0.0007	0.9826 ± 0.0005
32	112.8 ± 0.9	137.2 ± 1.0	0.9849 ± 0.0007	0.9828 ± 0.0003
64	*111.6 ± 4.0*	*135.1 ± 3.3*	*0.9850 ± 0.0006*	*0.9831 ± 0.0006*

Appendix: multi-task models training results

	log loss		AUC	
Size	Train	Validation	Train	Validation
4	0.458 ± 0.072	0.507 ± 0.064	0.720 ± 0.068	0.670 ± 0.060
5	0.442 ± 0.093	0.493 ± 0.084	0.757 ± 0.085	0.694 ± 0.080
6	0.421 ± 0.093	0.479 ± 0.082	0.768 ± 0.089	0.712 ± 0.076
7	0.380 ± 0.039	0.443 ± 0.032	0.806 ± 0.040	0.735 ± 0.039
8	0.357 ± 0.018	0.424 ± 0.010	0.832 ± 0.017	0.758 ± 0.016
10	0.349 ± 0.020	0.422 ± 0.012	0.845 ± 0.018	0.767 ± 0.017
12	0.340 ± 0.025	0.408 ± 0.012	0.855 ± 0.022	0.783 ± 0.022
16	0.339 ± 0.016	*0.406 ± 0.008*	0.857 ± 0.013	0.783 ± 0.019
32	0.340 ± 0.016	0.444 ± 0.007	0.859 ± 0.015	0.800 ± 0.016
64	*0.322 ± 0.014*	0.430 ± 0.009	*0.873 ± 0.013*	*0.816 ± 0.010*

Appendix: multi-task models have unstable training



Appendix: training times

	Autoencoder		Multi-task model	
Size	Train time [s]	Val time [s]	Train time [s]	Val time [s]
4	0.286 ± 0.025	0.012 ± 0.001	0.257 ± 0.023	0.013 ± 0.002
5	0.283 ± 0.023	0.012 ± 0.001	0.250 ± 0.017	0.013 ± 0.001
6	0.283 ± 0.021	0.012 ± 0.001	0.255 ± 0.019	0.013 ± 0.001
7	0.286 ± 0.025	0.012 ± 0.001	0.248 ± 0.019	0.013 ± 0.001
8	0.282 ± 0.023	0.012 ± 0.002	0.252 ± 0.023	0.013 ± 0.002
10	0.285 ± 0.025	0.012 ± 0.001	0.247 ± 0.017	0.013 ± 0.001
12	0.283 ± 0.025	0.012 ± 0.001	0.248 ± 0.016	0.013 ± 0.001
16	0.281 ± 0.025	0.012 ± 0.002	0.249 ± 0.020	0.013 ± 0.002
32	1.310 ± 0.408	0.016 ± 0.004	0.254 ± 0.021	0.013 ± 0.002
64	0.442 ± 0.152	0.015 ± 0.004	0.253 ± 0.051	0.013 ± 0.001

Appendix: training epochs

Size	Autoencoder Epochs	Multi-task model Epochs
4	413.0 ± 40.4	278.9 ± 94.5
5	447.7 ± 62.0	265.8 ± 104.4
6	413.9 ± 31.4	243.4 ± 42.0
7	422.9 ± 44.9	273.0 ± 58.7
8	445.2 ± 35.7	239.4 ± 29.6
10	466.5 ± 55.8	241.7 ± 28.8
12	514.9 ± 53.3	241.1 ± 32.8
16	552.9 ± 45.9	225.2 ± 17.7
32	605.1 ± 18.9	201.2 ± 20.5
64	593.6 ± 49.4	211.0 ± 17.7

Appendix: AUC-AUC trend lines

	Autoencoder		Multi-task model	
Size	slope	intercept	slope	intercept
4	0.615 ± 0.051	0.188 ± 0.039	0.564 ± 0.043	0.223 ± 0.034
5	0.650 ± 0.046	0.168 ± 0.035	0.611 ± 0.045	0.195 ± 0.035
6	0.691 ± 0.044	0.145 ± 0.034	0.673 ± 0.042	0.155 ± 0.032
7	0.753 ± 0.042	0.104 ± 0.033	0.692 ± 0.041	0.153 ± 0.032
8	0.732 ± 0.041	0.128 ± 0.032	0.724 ± 0.038	0.133 ± 0.030
10	0.771 ± 0.041	0.105 ± 0.032	0.755 ± 0.037	0.115 ± 0.029
12	0.785 ± 0.039	0.101 ± 0.030	0.773 ± 0.037	0.108 ± 0.028
16	0.810 ± 0.036	0.089 ± 0.028	0.804 ± 0.037	0.090 ± 0.029
32	0.866 ± 0.035	0.059 ± 0.027	0.842 ± 0.031	0.080 ± 0.024
64	*0.873 ± 0.033*	*0.059 ± 0.025*	*0.888 ± 0.027*	*0.050 ± 0.021*

Appendix: TPR-TNR values

	Autoencoder		Multi-task model	
Size	TPR	TNR	TPR	TNR
4	0.629 ± 0.008	0.678 ± 0.005	0.629 ± 0.009	0.671 ± 0.009
5	0.637 ± 0.006	0.691 ± 0.005	0.631 ± 0.011	0.685 ± 0.007
6	0.639 ± 0.008	0.705 ± 0.006	0.636 ± 0.011	0.702 ± 0.006
7	0.641 ± 0.008	0.717 ± 0.004	0.646 ± 0.008	0.717 ± 0.008
8	0.647 ± 0.006	0.731 ± 0.003	0.648 ± 0.007	0.724 ± 0.008
10	0.648 ± 0.005	0.743 ± 0.005	0.646 ± 0.007	0.739 ± 0.004
12	0.652 ± 0.007	0.758 ± 0.006	0.652 ± 0.008	0.752 ± 0.004
16	*0.653 ± 0.007*	0.777 ± 0.005	0.650 ± 0.007	0.768 ± 0.004
32	0.650 ± 0.004	0.809 ± 0.004	*0.655 ± 0.004*	0.808 ± 0.004
64	0.641 ± 0.005	*0.831 ± 0.003*	0.647 ± 0.007	*0.831 ± 0.003*

Appendix: PCA explained variance

	Autoencoder			Multi-task model		
Size	50%	90%	95%	50%	90%	95%
4	1.3 ± 0.5	3.0 ± 0.0	3.6 ± 0.5	1.1 ± 0.3	2.4 ± 0.7	3.0 ± 0.6
5	1.8 ± 0.4	3.1 ± 0.3	4.1 ± 0.3	1.2 ± 0.4	2.5 ± 0.8	3.3 ± 1.0
6	1.9 ± 0.3	3.3 ± 0.5	4.0 ± 0.0	1.6 ± 0.5	2.8 ± 1.1	3.2 ± 1.2
7	1.7 ± 0.5	3.7 ± 0.5	4.8 ± 0.4	1.5 ± 0.5	3.4 ± 0.7	4.2 ± 0.9
8	1.8 ± 0.4	3.9 ± 0.3	5.0 ± 0.0	1.8 ± 0.4	3.7 ± 0.5	4.7 ± 0.5
10	1.9 ± 0.3	4.2 ± 0.4	5.3 ± 0.5	1.3 ± 0.5	3.7 ± 0.5	4.9 ± 0.5
12	2.0 ± 0.0	4.3 ± 0.5	5.8 ± 0.4	1.9 ± 0.3	4.6 ± 0.7	5.9 ± 0.8
16	2.0 ± 0.0	4.5 ± 0.5	6.1 ± 0.3	1.7 ± 0.5	4.9 ± 0.5	6.1 ± 0.5
32	2.0 ± 0.0	5.0 ± 0.0	6.9 ± 0.3	1.4 ± 0.5	4.2 ± 0.6	5.7 ± 0.5
64	2.0 ± 0.0	5.3 ± 0.5	7.2 ± 0.4	1.5 ± 0.5	3.7 ± 0.5	4.8 ± 0.4
raw	1	4	13	1	4	13