Predicting Aphasia Type and Severity Using Machine Learning

Matvey Kairov
Research Assistant, Laboratory of Artificial Intelligence for Cognitive Sciences
HSE University, Moscow, Russia

supervised by Soroosh Shalileh
Laboratory Head, Laboratory of Artificial Intelligence for Cognitive Sciences
HSE University, Moscow, Russia

19 June 2024

Table of contents

Introduction

- Aphasia: a language processing disorder which results from brain damage
  - Affecting around 30% of the 15 million annual stroke patients
  - Lesions of specific brain regions cause specific aphasic symptoms [1]
- **Machine learning** can be used for predicting aphasic symptoms from brain imaging or its derivatives

[1] Broca, 1861. Remarks on the seat of the faculty of articulated language, following an observation of aphemia (loss of speech)

Motivation

- Most researches use **small or insufficient** datasets
- Prediction quality for complex aphasia classifications is **low**
- Data gathering is **limited** by high expenses and varying disease classifications

Challenges

- Small dataset size (406 patients)
- Missing target values for some patients
- **High class imbalance** for aphasia types

Aphasia type	Number of patients
Efferent motor + Afferent motor	128
Sensory	76
Efferent motor	49
Dynamic	46
Acoustic-mnestic	43
Dysarthria	26
Afferent motor	21

Class counts for aphasia type

Severity class	Number of patients
Mild	49
Mild-moderate	78
Moderate	111
Moderate-severe	50
Severe	44
Very severe	69

Class counts for aphasia severity

Problem statement

- **(Q1)**: Is it possible to predict aphasia type from MRI data?
- **(Q2)**: Is it possible to predict aphasia severity from MRI data?
- **(Q3)**: How can extremely small dataset size be combatted?
- **(Q4)**: What is the optimal combination of brain MRI features for aphasia type and severity prediction?
- **(Q5)**: What is the optimal representation of target values for the classification of aphasia severity?

Problem statement: data

- Input data:
    - derivatives of brain MRIs in the form of grey and white brain matter tabular features
    - demographic features
- Aphasia type (class labels) or severity (empirical diagnoses and test scores) as the target values
- Use classical ML methods to model the given datasets
- Combat small dataset size and class imbalance using generative data augmentation methods

Methodology: applied methods

- **K-Nearest Neighbors (KNN)**: a machine learning method that assigns classes or regression values based on the distances between objects.

- **Multi-Layer Perceptron (MLP)**: is a neural network consisting of multiple layers of neurons, and weights and biases between them which are used to finding a mapping between the input data target values.

- **Random Forest (RF)**: an ensemble learning method that builds multiple decision trees and makes predictions based on the aggregation the most popular outputs.

- **Gradient Boosting (GB)**: an ensemble learning method that builds a series of weak learners, each correcting the errors made by the previous ones.

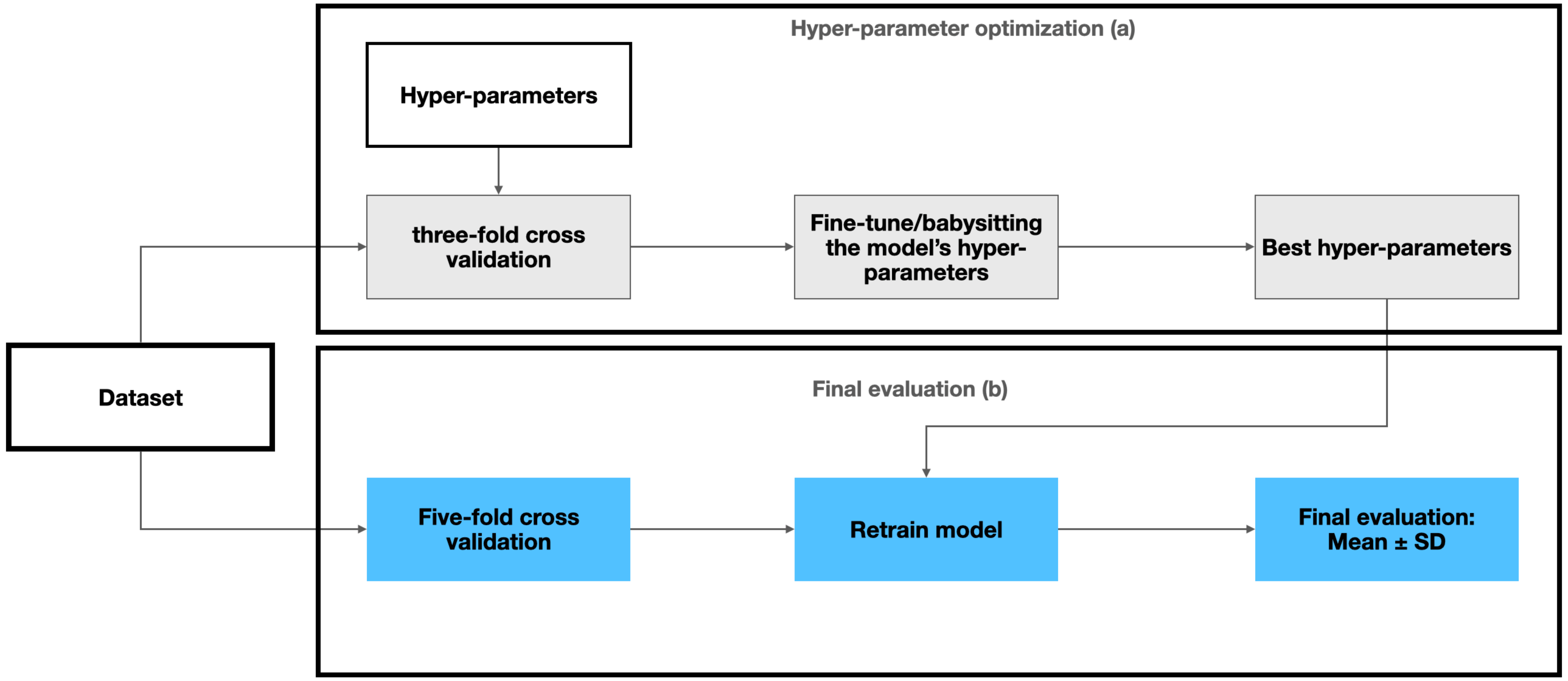
- **Support Vector Machines (SVM)**: a supervised learning algorithm that works by finding the optimal hyperplane that best separates different classes in the input data.

Methodology: computational setting

We applied the following computational setting:

A. Tuning hyper-parameters, using Bayesian Optimization [2]

B. Assessing the fine-tuned models



[2] Mockus J, Tiesis V, Zilinskas A. The application of Bayesian methods for seeking the extremum. Towards global optimization. 1978;2(117-129):2.

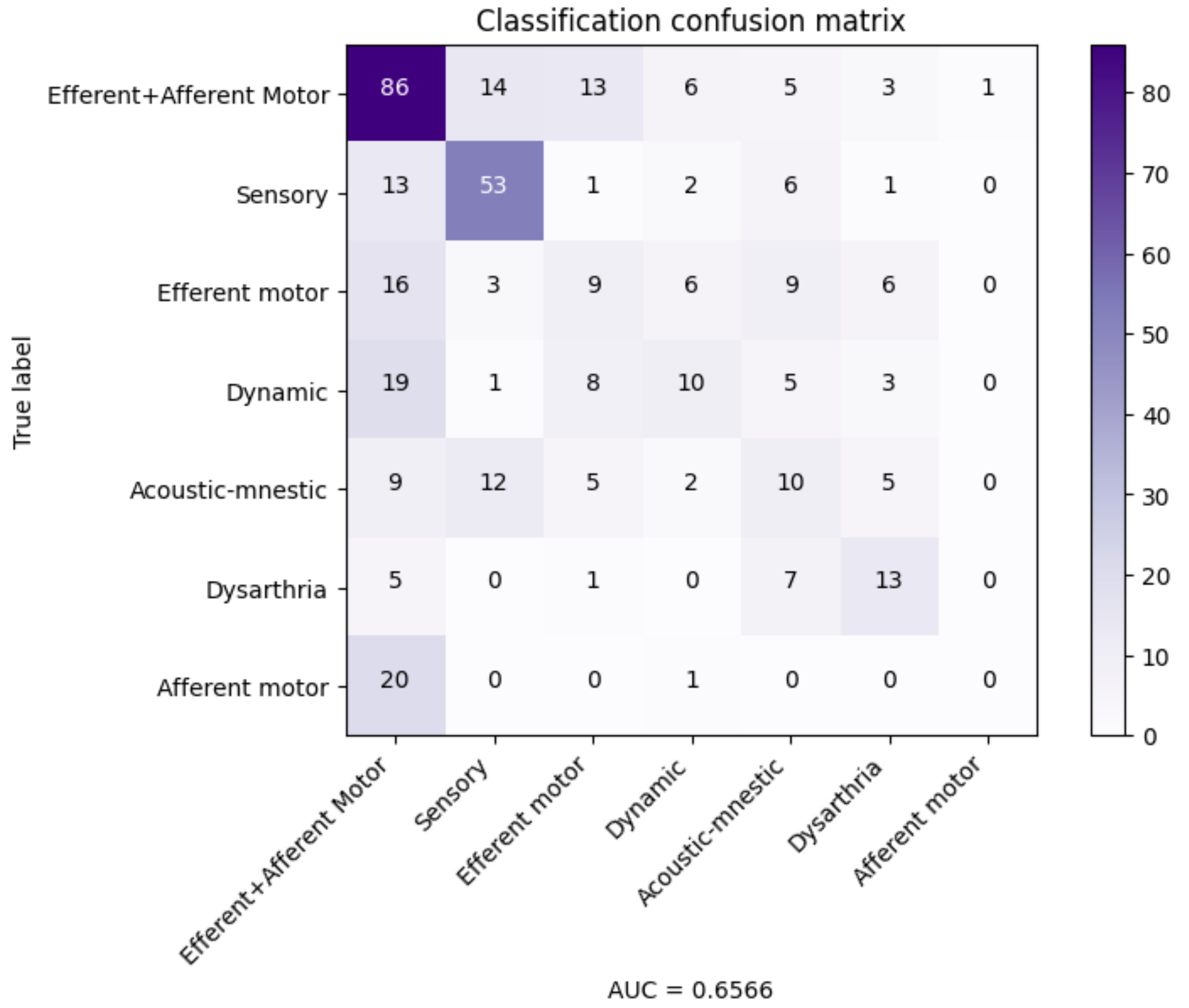
Methodology: vanilla classification

- Classification views target values as independent classes
- Aphasia type and severity are originally given as classes
- We applied classification algorithms to those target values

Methodology: vanilla classification

Methods	Metrics			
	Precision	Recall	$F1$-score	ROC-AUC
Random prediction	0.12058 ± 0.06204	0.08149 ± 0.05481	0.08579 ± 0.05383	0.25854 ± 0.24020
grey-knn	0.38203 ± 0.00678	0.43953 ± 0.01529	0.39224 ± 0.00910	0.64048 ± 0.00677
grey-d-knn	0.39158 ± 0.06417	0.42414 ± 0.03104	0.38433 ± 0.03923	0.63341 ± 0.02203
white-knn	0.36496 ± 0.08156	0.44496 ± 0.05072	0.38485 ± 0.05764	0.63823 ± 0.03556
white-d-knn	0.37205 ± 0.03457	0.42940 ± 0.03234	0.38137 ± 0.03315	0.62991 ± 0.02689
both-knn	0.37442 ± 0.03846	0.44226 ± 0.03798	0.39697 ± 0.03973	0.64101 ± 0.02695
both-d-knn	0.35421 ± 0.02821	0.42674 ± 0.02967	0.37799 ± 0.02715	0.62881 ± 0.02001
grey-mlp	0.40971 ± 0.05770	0.41662 ± 0.05311	0.40044 ± 0.05070	0.63458 ± 0.03620
grey-d-mlp	0.39952 ± 0.04271	0.43447 ± 0.03766	0.40471 ± 0.03886	0.63600 ± 0.02230
white-mlp	0.40553 ± 0.03571	0.43949 ± 0.03731	0.41420 ± 0.03062	0.64564 ± 0.02401
white-d-mlp	0.41807 ± 0.04720	0.41905 ± 0.07015	0.40546 ± 0.05049	0.63992 ± 0.04005
both-mlp	**0.44132 ± 0.04929**	**0.46533 ± 0.04857**	**0.43620 ± 0.04263**	**0.65689 ± 0.02451**
both-d-mlp	0.38600 ± 0.04584	0.40882 ± 0.03667	0.37920 ± 0.03740	0.62028 ± 0.02165
grey-rf	0.38072 ± 0.07551	0.44715 ± 0.04411	0.39325 ± 0.04539	0.63331 ± 0.02605
grey-d-rf	0.41724 ± 0.04609	0.48338 ± 0.04862	0.42294 ± 0.04068	0.65595 ± 0.02971
white-rf	0.41349 ± 0.05611	0.47056 ± 0.03485	0.42287 ± 0.03557	0.65227 ± 0.02118
white-d-rf	0.43054 ± 0.07073	0.47040 ± 0.04131	0.41840 ± 0.04442	0.64838 ± 0.02876
both-rf	0.40333 ± 0.06451	0.45751 ± 0.02874	0.39802 ± 0.02043	0.63860 ± 0.01904
both-d-rf	0.40616 ± 0.03694	0.46021 ± 0.02156	0.41615 ± 0.03474	0.64538 ± 0.02239
grey-gb	0.39591 ± 0.05227	0.43183 ± 0.04135	0.39922 ± 0.03855	0.63064 ± 0.02317
grey-d-gb	0.42352 ± 0.08054	0.43157 ± 0.07762	0.41449 ± 0.07881	0.63372 ± 0.05107
white-gb	0.38247 ± 0.01891	0.41395 ± 0.01676	0.38874 ± 0.01702	0.62484 ± 0.01380
white-d-gb	0.39983 ± 0.04528	0.44732 ± 0.02875	0.40201 ± 0.02474	0.63784 ± 0.01658
both-gb	0.41276 ± 0.04135	0.43720 ± 0.04772	0.40625 ± 0.04463	0.63530 ± 0.03149
both-d-gb	0.38316 ± 0.04982	0.42930 ± 0.01720	0.39445 ± 0.02964	0.63135 ± 0.01290
grey-svm	0.43234 ± 0.02517	0.45764 ± 0.02481	0.41005 ± 0.01106	0.64823 ± 0.01070
grey-d-svm	0.40139 ± 0.05557	0.43713 ± 0.05033	0.40480 ± 0.04892	0.63886 ± 0.03552
white-svm	0.43996 ± 0.11386	0.45238 ± 0.06614	0.41099 ± 0.07703	0.64749 ± 0.04482
white-d-svm	0.41931 ± 0.03292	0.44732 ± 0.03766	0.41634 ± 0.03537	0.64541 ± 0.02699
both-svm	0.40914 ± 0.03426	0.44732 ± 0.03301	0.40701 ± 0.02868	0.64565 ± 0.02140
both-d-svm	0.38051 ± 0.03254	0.40606 ± 0.03657	0.38689 ± 0.03222	0.62386 ± 0.02297

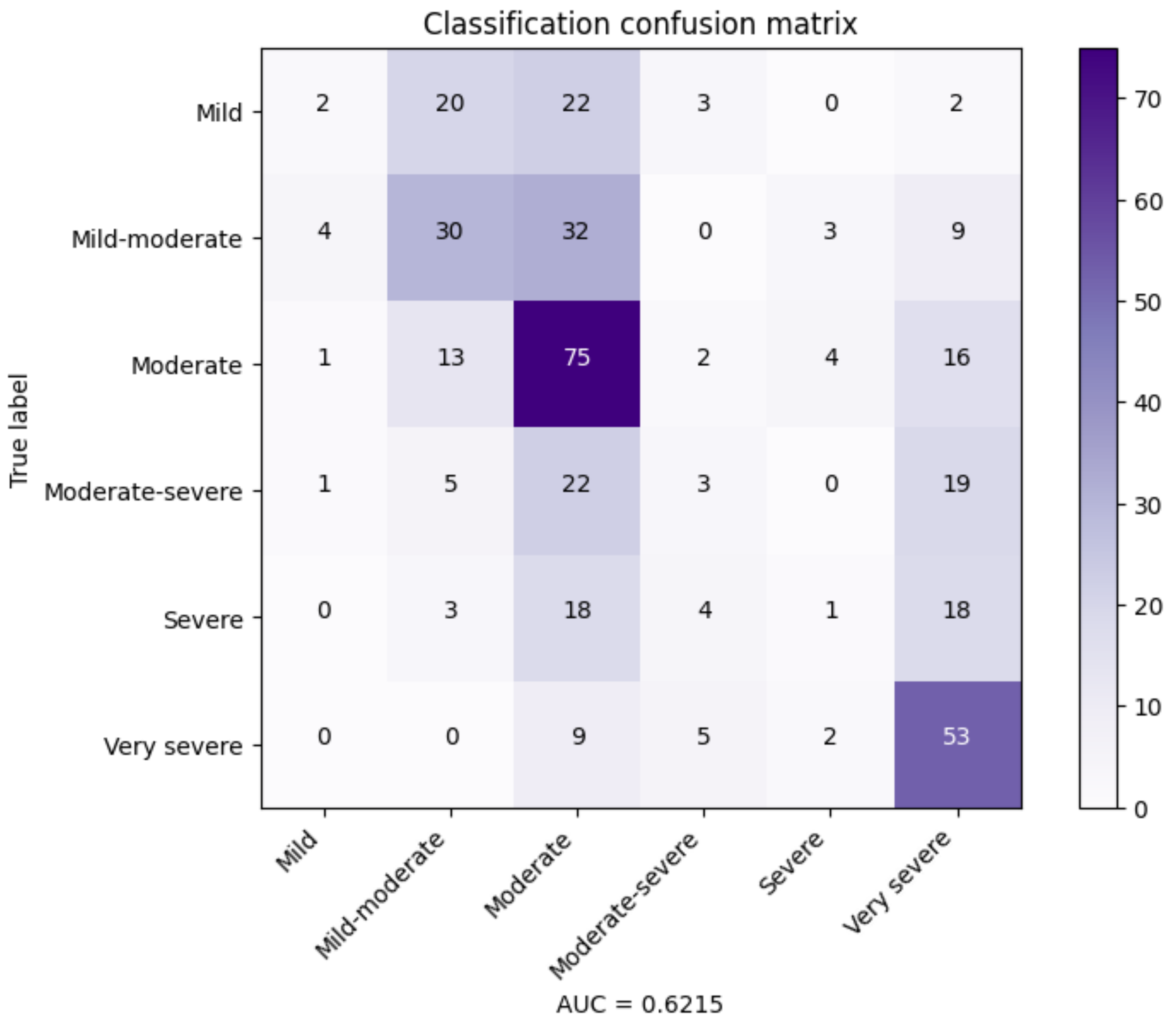
The best results obtained for type classification by MLP using grey and white matter features (AUC 0.66).



Methodology: vanilla classification

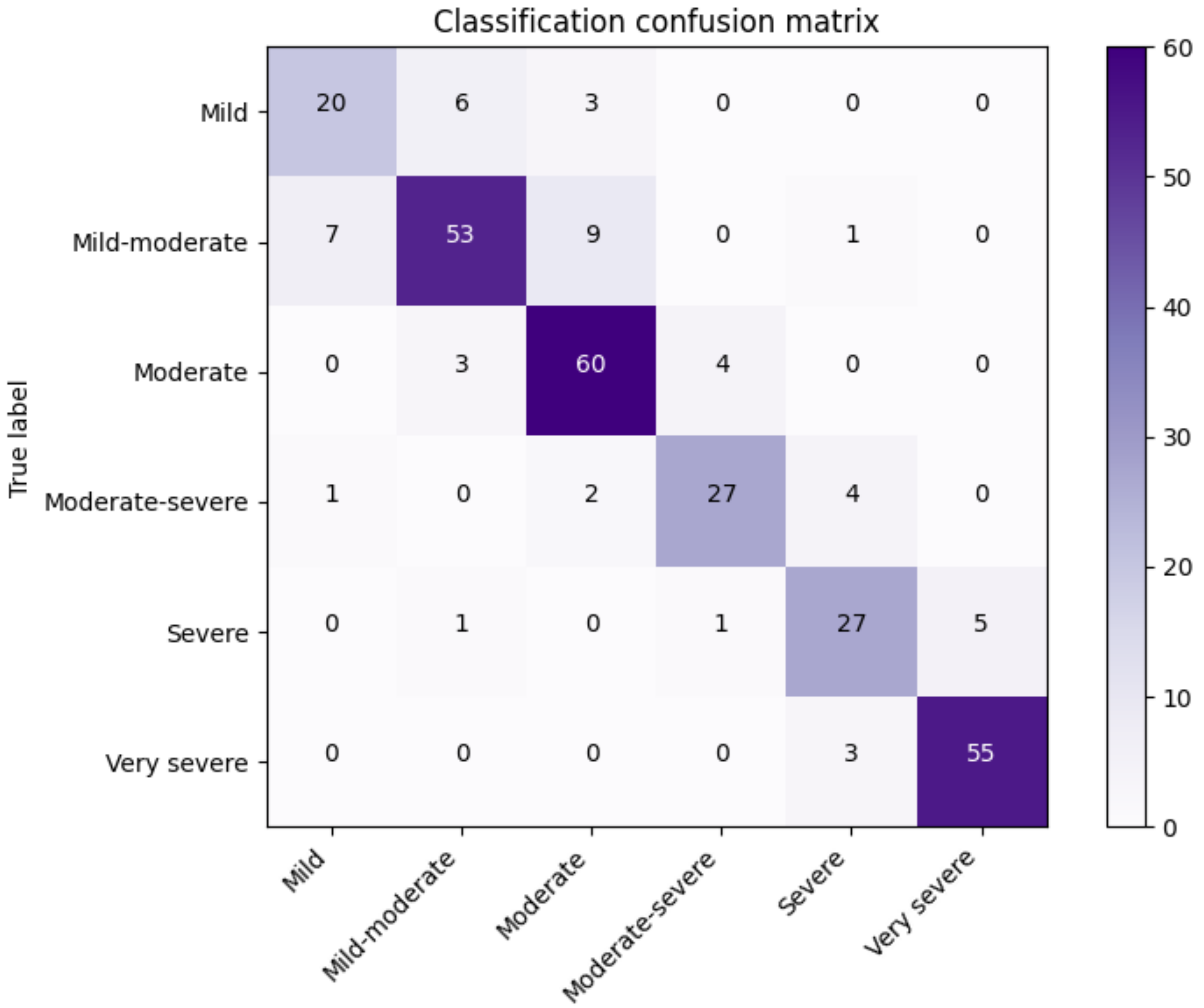
Methods	Metrics			
	Precision	Recall	$F1$-score	ROC-AUC
Random prediction	0.12252 ± 0.07177	0.10564 ± 0.07133	0.10967 ± 0.06923	0.26435 ± 0.24190
grey-knn	0.37729 ± 0.03855	0.38164 ± 0.02172	0.35714 ± 0.02351	0.61246 ± 0.01929
grey-d-knn	0.32710 ± 0.03581	0.34414 ± 0.04367	0.32070 ± 0.03655	0.58464 ± 0.02767
white-knn	0.35689 ± 0.04535	0.39157 ± 0.01798	0.34471 ± 0.01641	0.61664 ± 0.01128
white-d-knn	0.29231 ± 0.03490	0.38407 ± 0.03138	0.32431 ± 0.02883	0.60742 ± 0.01933
both-knn	0.30977 ± 0.07287	0.37904 ± 0.05886	0.32363 ± 0.05749	0.60327 ± 0.03488
both-d-knn	0.37320 ± 0.05257	0.37148 ± 0.04884	0.33546 ± 0.04865	0.59889 ± 0.03453
grey-mlp	0.28750 ± 0.03553	0.40151 ± 0.02800	0.32711 ± 0.02994	0.61031 ± 0.02155
grey-d-mlp	0.33940 ± 0.04474	0.35404 ± 0.04306	0.34071 ± 0.04082	0.60027 ± 0.02665
white-mlp	0.33164 ± 0.03851	0.36648 ± 0.02872	0.33594 ± 0.02985	0.59983 ± 0.01714
white-d-mlp	0.32240 ± 0.03507	0.36907 ± 0.02017	0.33864 ± 0.02424	0.60432 ± 0.01267
both-mlp	0.35185 ± 0.07242	0.38417 ± 0.04236	0.33202 ± 0.03522	0.60224 ± 0.02398
both-d-mlp	0.30977 ± 0.04973	0.31654 ± 0.05032	0.30415 ± 0.04626	0.58061 ± 0.03133
grey-rf	0.29291 ± 0.03808	0.39904 ± 0.01467	0.31203 ± 0.01057	0.60751 ± 0.01166
grey-d-rf	0.36262 ± 0.04818	0.40151 ± 0.03916	0.34781 ± 0.04216	0.61534 ± 0.02614
white-rf	**0.32770** ± **0.04231**	**0.40898** ± **0.01221**	**0.34315** ± **0.01641**	**0.62161** ± **0.00889**
white-d-rf	0.31555 ± 0.03488	0.39654 ± 0.01050	0.32550 ± 0.01890	0.61082 ± 0.00559
both-rf	0.26606 ± 0.02537	0.40139 ± 0.04315	0.30945 ± 0.03608	0.61035 ± 0.02946
both-d-rf	0.30234 ± 0.02956	0.41139 ± 0.04011	0.32707 ± 0.02775	0.61688 ± 0.02586
grey-gb	0.34438 ± 0.05510	0.37401 ± 0.06035	0.34457 ± 0.05765	0.60363 ± 0.03999
grey-d-gb	0.37199 ± 0.07608	0.40907 ± 0.04220	0.36132 ± 0.04584	0.62092 ± 0.02648
white-gb	0.31443 ± 0.04765	0.33404 ± 0.04097	0.31388 ± 0.03774	0.58397 ± 0.02272
white-d-gb	0.31104 ± 0.02803	0.34664 ± 0.02147	0.31616 ± 0.01930	0.58915 ± 0.01305
both-gb	0.32130 ± 0.06473	0.35151 ± 0.06372	0.32840 ± 0.06200	0.59621 ± 0.03831
both-d-gb	0.33447 ± 0.01856	0.35898 ± 0.02593	0.33566 ± 0.02413	0.59583 ± 0.01937
grey-svm	0.29692 ± 0.02635	0.37417 ± 0.03264	0.31929 ± 0.02973	0.59771 ± 0.02199
grey-d-svm	0.36693 ± 0.04986	0.39160 ± 0.02682	0.36007 ± 0.02455	0.61477 ± 0.01701
white-svm	0.30282 ± 0.02484	0.38407 ± 0.02201	0.32215 ± 0.02110	0.60250 ± 0.01517
white-d-svm	0.35529 ± 0.11287	0.37676 ± 0.07837	0.34160 ± 0.07863	0.60275 ± 0.04823
both-svm	0.27613 ± 0.03553	0.38651 ± 0.03323	0.31180 ± 0.03140	0.60210 ± 0.02128
both-d-svm	0.36668 ± 0.02604	0.40148 ± 0.02866	0.37110 ± 0.03019	0.61978 ± 0.02203

The best results obtained for severity classification by RF using features for white matter only (AUC 0.62).



Methodology: severity estimation

- Severity estimations are given in one or two forms – classes (mild, moderate etc.) or behavioral scores (ASA)
- Severity is easier for doctors to understand, but has lower precision
- ASA is more accurate but is out of use due to a time-consuming testing process
- We can see that in approximately **15% of cases** doctors made mistakes (compared to the case where ASA was applied)



Methodology: severity estimation

- Severity classes are ordered → use ordinal regression:
  - Naive: map severity classes to numerical values ("Mild" to 0, "Mild-moderate to 1 etc.), estimate with regression, round and evaluate like classification
  - Distance-based: use more precise ASA scores as regression targets
  - Ordinal regression method by Frank and Hall: use #*classes - 1* estimators, the i-th estimator predicting whether an object falls above the i-th class, compute probabilities with:

$$P(V_1) = 1 - P(Target > V_1)$$

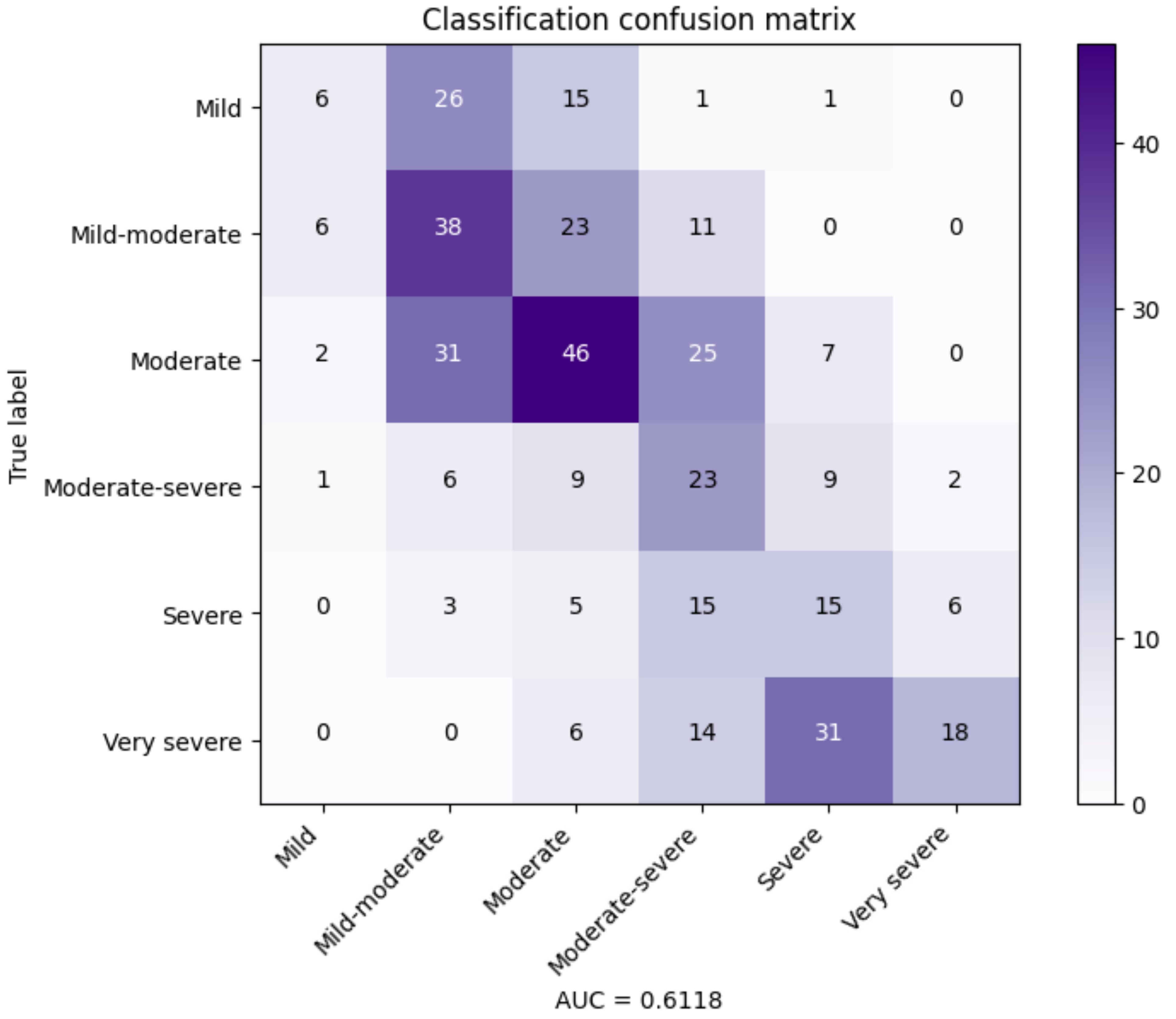
$$P(V_i) = P(Target > V_{i-1}) - P(Target > V_i), 1 < i < n$$

$$P(V_n) = P(Target > V_{n-1})$$

Methodology: severity estimation

Methods	Metrics					
	MAE	R^2	Precision	Recall	$F1$-score	ROC-AUC
Random prediction	1.02990 ± 1.01454	-0.70598 ± 0.73997	0.03216 ± 0.01909	0.09209 ± 0.05742	0.04683 ± 0.02841	0.25860 ± 0.23709
grey-knn	1.02851 ± 0.06218	0.36475 ± 0.07862	0.32565 ± 0.10390	0.30694 ± 0.06128	0.26821 ± 0.05543	0.57289 ± 0.03789
grey-d-knn	1.02310 ± 0.08271	0.35573 ± 0.09958	0.28531 ± 0.13078	0.31401 ± 0.06363	0.26110 ± 0.06874	0.57494 ± 0.04089
white-knn	1.04864 ± 0.06852	0.36294 ± 0.07110	0.24701 ± 0.07733	0.28435 ± 0.03699	0.24646 ± 0.03369	0.56039 ± 0.02258
white-d-knn	1.00975 ± 0.09225	0.38597 ± 0.08974	0.27619 ± 0.08712	0.30420 ± 0.03719	0.26280 ± 0.03390	0.57304 ± 0.02331
both-knn	1.02794 ± 0.12624	0.37610 ± 0.12617	0.27561 ± 0.10212	0.28941 ± 0.06773	0.25117 ± 0.06067	0.56202 ± 0.04474
both-d-knn	1.02743 ± 0.10335	0.36712 ± 0.13195	0.29832 ± 0.07974	0.29664 ± 0.04183	0.25494 ± 0.03974	0.56508 ± 0.02871
grey-mlp	1.02068 ± 0.07252	0.37508 ± 0.07713	0.32692 ± 0.06539	0.31938 ± 0.04744	0.29971 ± 0.04124	0.57734 ± 0.03452
grey-d-mlp	0.91044 ± 0.05861	0.50657 ± 0.05896	0.39121 ± 0.05195	0.31670 ± 0.05211	0.31132 ± 0.05067	0.58504 ± 0.03215
white-mlp	0.99305 ± 0.02477	0.40578 ± 0.03386	0.30371 ± 0.03782	0.31682 ± 0.04150	0.28657 ± 0.04296	0.57596 ± 0.02614
white-d-mlp	0.92786 ± 0.04147	0.47237 ± 0.06981	0.36654 ± 0.04136	0.34910 ± 0.02045	0.33360 ± 0.02340	0.59957 ± 0.01508
both-mlp	0.98520 ± 0.07614	0.41487 ± 0.10547	0.32220 ± 0.05444	0.30660 ± 0.03959	0.28187 ± 0.03556	0.57096 ± 0.02385
both-d-mlp	**0.90448 ± 0.04456**	**0.50048 ± 0.03463**	**0.42254 ± 0.03240**	**0.36407 ± 0.04120**	**0.35851 ± 0.03060**	**0.61159 ± 0.02812**
grey-rf	1.00513 ± 0.07841	0.40530 ± 0.07770	0.25612 ± 0.07450	0.29417 ± 0.02796	0.25247 ± 0.02814	0.56351 ± 0.01783
grey-d-rf	0.94567 ± 0.02757	0.47183 ± 0.04120	0.29065 ± 0.08994	0.31182 ± 0.02942	0.26927 ± 0.03349	0.57508 ± 0.02064
white-rf	0.99859 ± 0.02555	0.41402 ± 0.02780	0.22029 ± 0.03151	0.30176 ± 0.03665	0.25180 ± 0.03466	0.56907 ± 0.02315
white-d-rf	0.94534 ± 0.09588	0.47518 ± 0.09861	0.33450 ± 0.08953	0.31651 ± 0.05537	0.28301 ± 0.05076	0.58020 ± 0.03416
both-rf	0.99011 ± 0.07216	0.42030 ± 0.04795	0.23108 ± 0.04353	0.30654 ± 0.05602	0.25972 ± 0.04703	0.57467 ± 0.03518
both-d-rf	0.96216 ± 0.04276	0.45338 ± 0.06143	0.25955 ± 0.05444	0.29676 ± 0.05384	0.25560 ± 0.03948	0.56798 ± 0.03126
grey-gb	1.02043 ± 0.04933	0.39270 ± 0.03963	0.22194 ± 0.02663	0.29688 ± 0.02890	0.24990 ± 0.02562	0.56518 ± 0.01950
grey-d-gb	0.87543 ± 0.05283	0.53429 ± 0.03452	0.44453 ± 0.06937	0.35920 ± 0.04700	0.35142 ± 0.04989	0.60607 ± 0.02457
white-gb	1.00860 ± 0.02109	0.39668 ± 0.03147	0.30017 ± 0.08510	0.30429 ± 0.02376	0.26881 ± 0.02011	0.57075 ± 0.01422
white-d-gb	0.86614 ± 0.08332	0.54479 ± 0.06606	0.42724 ± 0.09361	0.34157 ± 0.06504	0.32918 ± 0.06165	0.59812 ± 0.03762
both-gb	0.99935 ± 0.06887	0.41819 ± 0.05283	0.29807 ± 0.08183	0.28410 ± 0.04625	0.25158 ± 0.03504	0.55948 ± 0.02397
both-d-gb	0.91162 ± 0.07176	0.49820 ± 0.06816	0.44003 ± 0.04855	0.33898 ± 0.06078	0.32828 ± 0.05932	0.59366 ± 0.03295
grey-svm	0.96675 ± 0.05893	0.42311 ± 0.05770	0.32946 ± 0.03947	0.33664 ± 0.05166	0.31280 ± 0.04807	0.59006 ± 0.03278
grey-d-svm	0.92222 ± 0.09235	0.48852 ± 0.09357	0.42558 ± 0.07078	0.36157 ± 0.04234	0.36118 ± 0.04268	0.61080 ± 0.02714
white-svm	0.99000 ± 0.05800	0.40897 ± 0.04288	0.31940 ± 0.03184	0.32173 ± 0.04160	0.29918 ± 0.03677	0.57834 ± 0.02606
white-d-svm	0.91077 ± 0.05668	0.50326 ± 0.05019	0.38515 ± 0.06010	0.33407 ± 0.03814	0.32831 ± 0.03778	0.59132 ± 0.02289
both-svm	0.99969 ± 0.03220	0.39952 ± 0.03896	0.29287 ± 0.08051	0.30951 ± 0.07252	0.28689 ± 0.07089	0.57250 ± 0.04391
both-d-svm	0.93237 ± 0.03752	0.48112 ± 0.04567	0.36812 ± 0.02925	0.33176 ± 0.02670	0.32577 ± 0.02432	0.58909 ± 0.01478

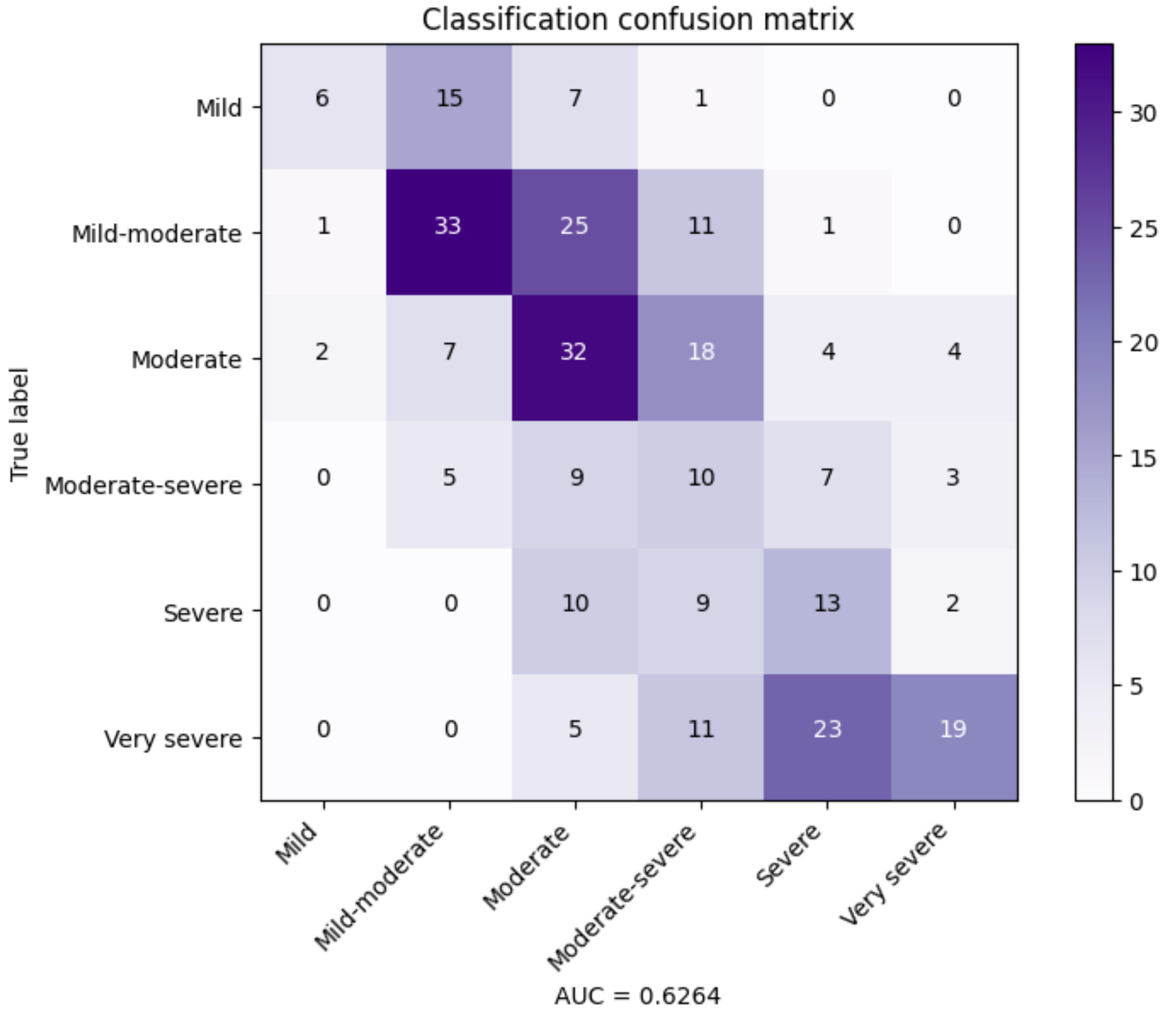
MLP on grey/white matter + demographic features is the best for naive ordinal regression (AUC 0.61).



Methodology: severity estimation

Methods	Metrics					
	MAE	R^2	Precision	Recall	$F1$-score	ROC-AUC
Random predictions	87.62083 ± 84.60780	-2.29912 ± 2.72637	0.02093 ± 0.01828	0.10231 ± 0.09558	0.03475 ± 0.03069	0.25000 ± 0.25000
grey-knn	50.74104 ± 5.00295	0.33915 ± 0.11430	0.32415 ± 0.08769	0.29684 ± 0.03449	0.27234 ± 0.03457	0.56908 ± 0.02032
grey-d-knn	50.86403 ± 4.88619	0.35867 ± 0.08395	0.29605 ± 0.07021	0.28334 ± 0.02631	0.25800 ± 0.03562	0.55571 ± 0.01756
white-knn	50.42725 ± 2.85286	0.36327 ± 0.04269	0.30012 ± 0.05236	0.27329 ± 0.05501	0.25071 ± 0.04771	0.55597 ± 0.03660
white-d-knn	52.71450 ± 6.04260	0.30530 ± 0.12795	0.28309 ± 0.09327	0.24880 ± 0.04481	0.24250 ± 0.04841	0.53928 ± 0.03118
both-knn	51.38165 ± 1.84967	0.34138 ± 0.04943	0.32653 ± 0.05217	0.28001 ± 0.02474	0.25463 ± 0.02169	0.55649 ± 0.01367
both-d-knn	51.77606 ± 2.72122	0.33696 ± 0.10686	0.29128 ± 0.05033	0.26283 ± 0.03541	0.25034 ± 0.03477	0.54664 ± 0.02472
grey-mlp	48.02655 ± 2.31533	0.42566 ± 0.04838	0.32692 ± 0.01923	0.30707 ± 0.02936	0.29598 ± 0.02996	0.57334 ± 0.01658
grey-d-mlp	43.00122 ± 3.71335	0.53258 ± 0.05933	0.41247 ± 0.06005	0.35155 ± 0.06122	0.35777 ± 0.06158	0.60968 ± 0.03632
white-mlp	49.89421 ± 2.08738	0.39962 ± 0.05614	0.28326 ± 0.07232	0.28346 ± 0.04091	0.26296 ± 0.05534	0.56311 ± 0.02577
white-d-mlp	43.84262 ± 2.01511	0.51124 ± 0.05031	0.37817 ± 0.05011	0.33425 ± 0.04216	0.33675 ± 0.03649	0.59574 ± 0.02256
both-mlp	49.49715 ± 3.82272	0.37605 ± 0.10271	0.32098 ± 0.06422	0.29328 ± 0.05175	0.28858 ± 0.05223	0.56879 ± 0.03304
both-d-mlp	46.37948 ± 4.30500	0.45162 ± 0.11082	0.34773 ± 0.05042	0.31373 ± 0.08412	0.31428 ± 0.06957	0.58169 ± 0.04804
grey-rf	48.84381 ± 3.64611	0.40170 ± 0.05397	0.27438 ± 0.07509	0.27598 ± 0.06038	0.25619 ± 0.06178	0.56055 ± 0.03679
grey-d-rf	47.02193 ± 2.80248	0.44922 ± 0.03854	0.36290 ± 0.07492	0.32075 ± 0.05952	0.30830 ± 0.06008	0.58639 ± 0.03454
white-rf	49.51421 ± 2.74688	0.39274 ± 0.06530	0.25831 ± 0.03151	0.26978 ± 0.02659	0.24242 ± 0.02049	0.55820 ± 0.01556
white-d-rf	45.62030 ± 4.14178	0.48076 ± 0.08108	0.41516 ± 0.10602	0.34471 ± 0.05348	0.33214 ± 0.04484	0.60077 ± 0.03157
both-rf	49.23136 ± 2.12639	0.39726 ± 0.03728	0.29320 ± 0.09141	0.27960 ± 0.06027	0.26447 ± 0.05808	0.56325 ± 0.03482
both-d-rf	45.89406 ± 1.16482	0.46642 ± 0.02647	0.37113 ± 0.05122	0.34798 ± 0.04020	0.32842 ± 0.03078	0.60309 ± 0.02378
grey-gb	48.54087 ± 2.51181	0.40118 ± 0.04462	0.33978 ± 0.02343	0.29673 ± 0.02761	0.28755 ± 0.03173	0.57002 ± 0.01771
grey-d-gb	42.42614 ± 2.59993	0.52094 ± 0.05153	0.46780 ± 0.07545	0.36540 ± 0.05725	0.37866 ± 0.06751	0.61554 ± 0.03834
white-gb	49.12067 ± 3.25705	0.39442 ± 0.07598	0.30097 ± 0.03792	0.29036 ± 0.04356	0.26670 ± 0.03930	0.56755 ± 0.02738
white-d-gb	**42.66543 ± 3.59227**	**0.51818 ± 0.07584**	**0.47705 ± 0.04647**	**0.38556 ± 0.04585**	**0.39089 ± 0.05329**	**0.62605 ± 0.02831**
both-gb	47.87280 ± 2.98297	0.39349 ± 0.08678	0.34033 ± 0.06934	0.30695 ± 0.02824	0.29293 ± 0.03894	0.57926 ± 0.01667
both-d-gb	42.42853 ± 3.67623	0.52519 ± 0.08492	0.44616 ± 0.06730	0.37563 ± 0.03273	0.37629 ± 0.03241	0.62027 ± 0.02117
grey-svm	48.32567 ± 4.66214	0.41549 ± 0.06682	0.31777 ± 0.03432	0.31432 ± 0.04095	0.30107 ± 0.03549	0.57609 ± 0.02627
grey-d-svm	44.44686 ± 4.29660	0.48210 ± 0.09812	0.40966 ± 0.04443	0.35874 ± 0.04748	0.36123 ± 0.03384	0.60721 ± 0.02830
white-svm	49.08008 ± 3.07969	0.37289 ± 0.08265	0.30547 ± 0.03620	0.31058 ± 0.04607	0.29287 ± 0.04170	0.57453 ± 0.02807
white-d-svm	44.97302 ± 2.56402	0.47735 ± 0.04290	0.38512 ± 0.05353	0.33092 ± 0.03544	0.32807 ± 0.03411	0.58954 ± 0.02179
both-svm	49.33867 ± 1.92634	0.37927 ± 0.05965	0.33569 ± 0.03750	0.33466 ± 0.03840	0.32304 ± 0.03418	0.58844 ± 0.02571
both-d-svm	45.42318 ± 6.49437	0.47336 ± 0.15019	0.37827 ± 0.10671	0.33098 ± 0.06961	0.33202 ± 0.07608	0.58989 ± 0.04747

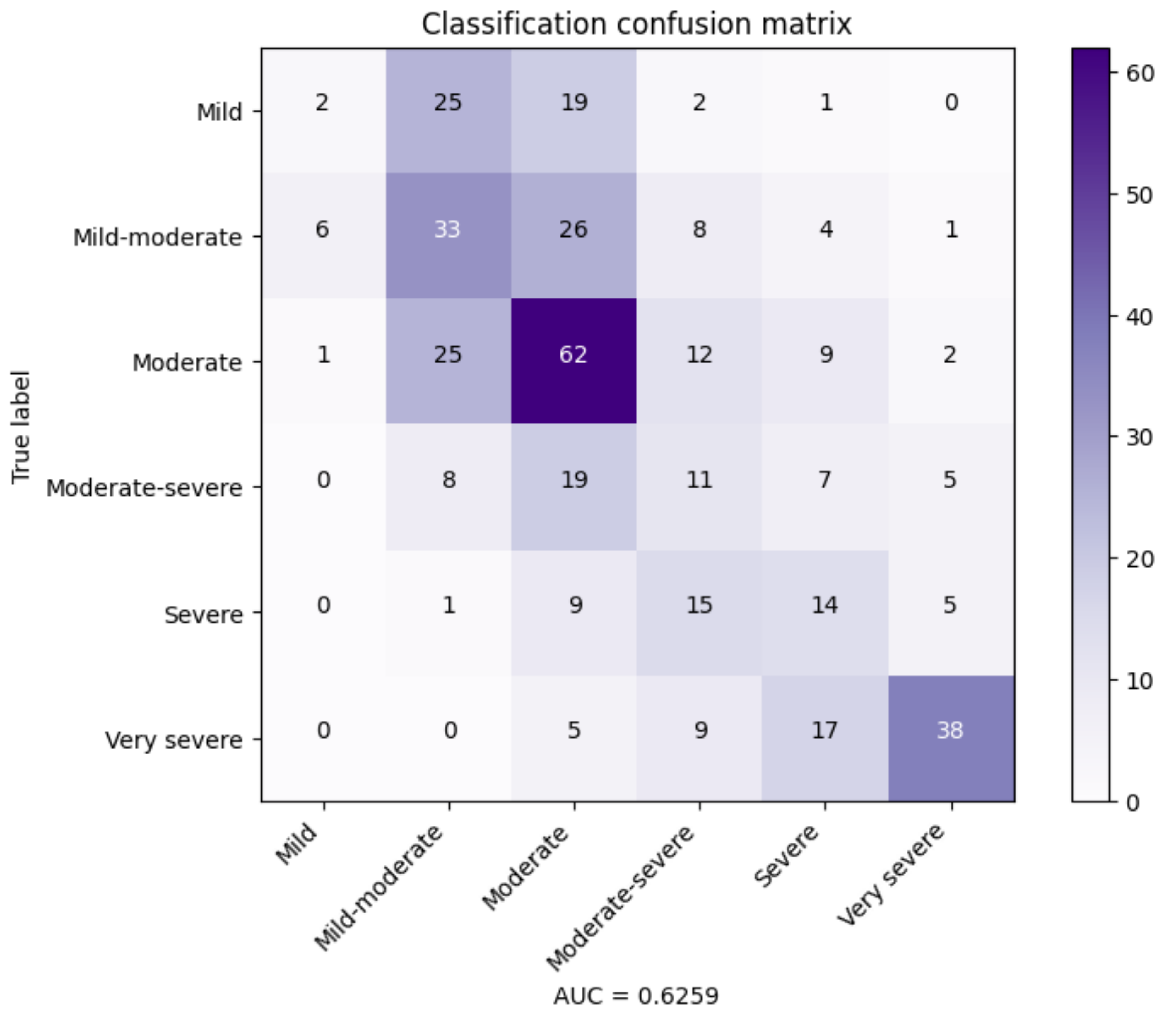
Gradient boosting on white matter + demographic features is the best for distance-based ordinal regression (AUC 0.63).



Methodology: severity estimation

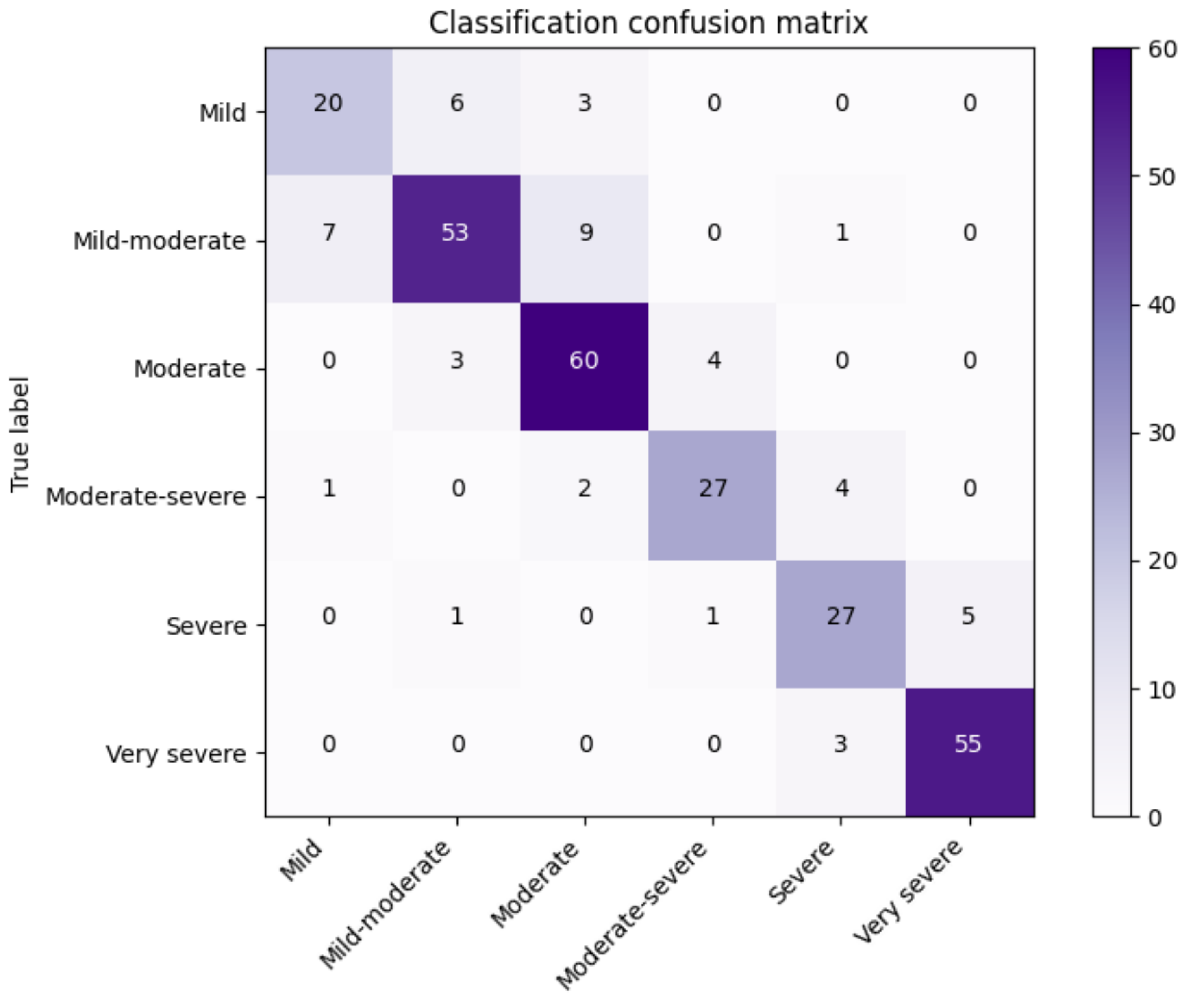
Methods	Metrics			
	Precision	Recall	$F1$-score	ROC-AUC
Random predictions	0.02093 ± 0.01828	0.10231 ± 0.09558	0.03475 ± 0.03069	0.25000 ± 0.25000
grey-knn	0.32353 ± 0.05007	0.35407 ± 0.03541	0.32841 ± 0.04549	0.59338 ± 0.02706
grey-d-knn	0.31500 ± 0.04292	0.33676 ± 0.03252	0.31192 ± 0.03260	0.58050 ± 0.02154
white-knn	0.30005 ± 0.03527	0.35417 ± 0.04265	0.31734 ± 0.03823	0.58925 ± 0.02775
white-d-knn	0.34756 ± 0.04431	0.35660 ± 0.03212	0.32905 ± 0.02930	0.58890 ± 0.01569
both-knn	0.31077 ± 0.04946	0.35929 ± 0.07162	0.31865 ± 0.05820	0.59246 ± 0.04514
both-d-knn	0.32244 ± 0.03354	0.35157 ± 0.02936	0.32661 ± 0.02940	0.59036 ± 0.01545
grey-mlp	0.33695 ± 0.04058	0.39404 ± 0.04670	0.35474 ± 0.04166	0.61621 ± 0.02665
grey-d-mlp	0.34327 ± 0.02012	0.39898 ± 0.02188	0.36195 ± 0.02021	0.61910 ± 0.01635
white-mlp	0.30725 ± 0.05096	0.36654 ± 0.04035	0.32817 ± 0.04619	0.59889 ± 0.02558
white-d-mlp	0.35518 ± 0.06720	0.35679 ± 0.06336	0.34844 ± 0.06724	0.60174 ± 0.03981
both-mlp	0.34526 ± 0.04121	0.37901 ± 0.02240	0.34805 ± 0.02812	0.60741 ± 0.01715
both-d-mlp	0.36013 ± 0.05461	0.36660 ± 0.06413	0.35743 ± 0.05717	0.61064 ± 0.03918
grey-rf	0.33398 ± 0.04646	0.38667 ± 0.04961	0.34110 ± 0.04040	0.60574 ± 0.02917
grey-d-rf	0.32675 ± 0.05915	0.40160 ± 0.05454	0.34426 ± 0.05603	0.61229 ± 0.03670
white-rf	0.35694 ± 0.07250	0.38670 ± 0.05626	0.35657 ± 0.06160	0.61022 ± 0.03366
white-d-rf	0.30375 ± 0.01600	0.36917 ± 0.02682	0.32421 ± 0.01854	0.59719 ± 0.01614
both-rf	0.31600 ± 0.05406	0.38664 ± 0.03880	0.33236 ± 0.04114	0.60557 ± 0.02500
both-d-rf	0.31990 ± 0.02743	0.38160 ± 0.03281	0.33750 ± 0.03133	0.60301 ± 0.02282
grey-gb	0.34924 ± 0.07496	0.34185 ± 0.06036	0.33124 ± 0.06300	0.58974 ± 0.04006
grey-d-gb	0.37345 ± 0.05631	0.37154 ± 0.04330	0.35683 ± 0.04813	0.60718 ± 0.02478
white-gb	0.32934 ± 0.05005	0.32926 ± 0.05083	0.31746 ± 0.05109	0.58129 ± 0.03337
white-d-gb	**0.39245 ± 0.03826**	**0.39920 ± 0.06071**	**0.38018 ± 0.05161**	**0.62596 ± 0.03903**
both-gb	0.34503 ± 0.06464	0.36173 ± 0.05549	0.34277 ± 0.05598	0.59806 ± 0.03738
both-d-gb	0.36075 ± 0.03946	0.37151 ± 0.05338	0.35153 ± 0.04625	0.60568 ± 0.03444
grey-svm	0.33047 ± 0.04040	0.36176 ± 0.04970	0.32832 ± 0.04414	0.59477 ± 0.02889
grey-d-svm	0.31737 ± 0.05233	0.36188 ± 0.06085	0.33067 ± 0.05392	0.59655 ± 0.03530
white-svm	0.35947 ± 0.05962	0.37167 ± 0.04211	0.34390 ± 0.04385	0.60299 ± 0.02578
white-d-svm	0.35075 ± 0.02175	0.41139 ± 0.02592	0.36780 ± 0.01867	0.62443 ± 0.01490
both-svm	0.33360 ± 0.03365	0.35176 ± 0.03417	0.32242 ± 0.03014	0.58775 ± 0.02012
both-d-svm	0.34099 ± 0.01878	0.39157 ± 0.01964	0.35549 ± 0.01767	0.61627 ± 0.01464

Gradient Boosting on white matter+demographic features is the best for Frank-Hall ordinal regression (AUC 0.63).



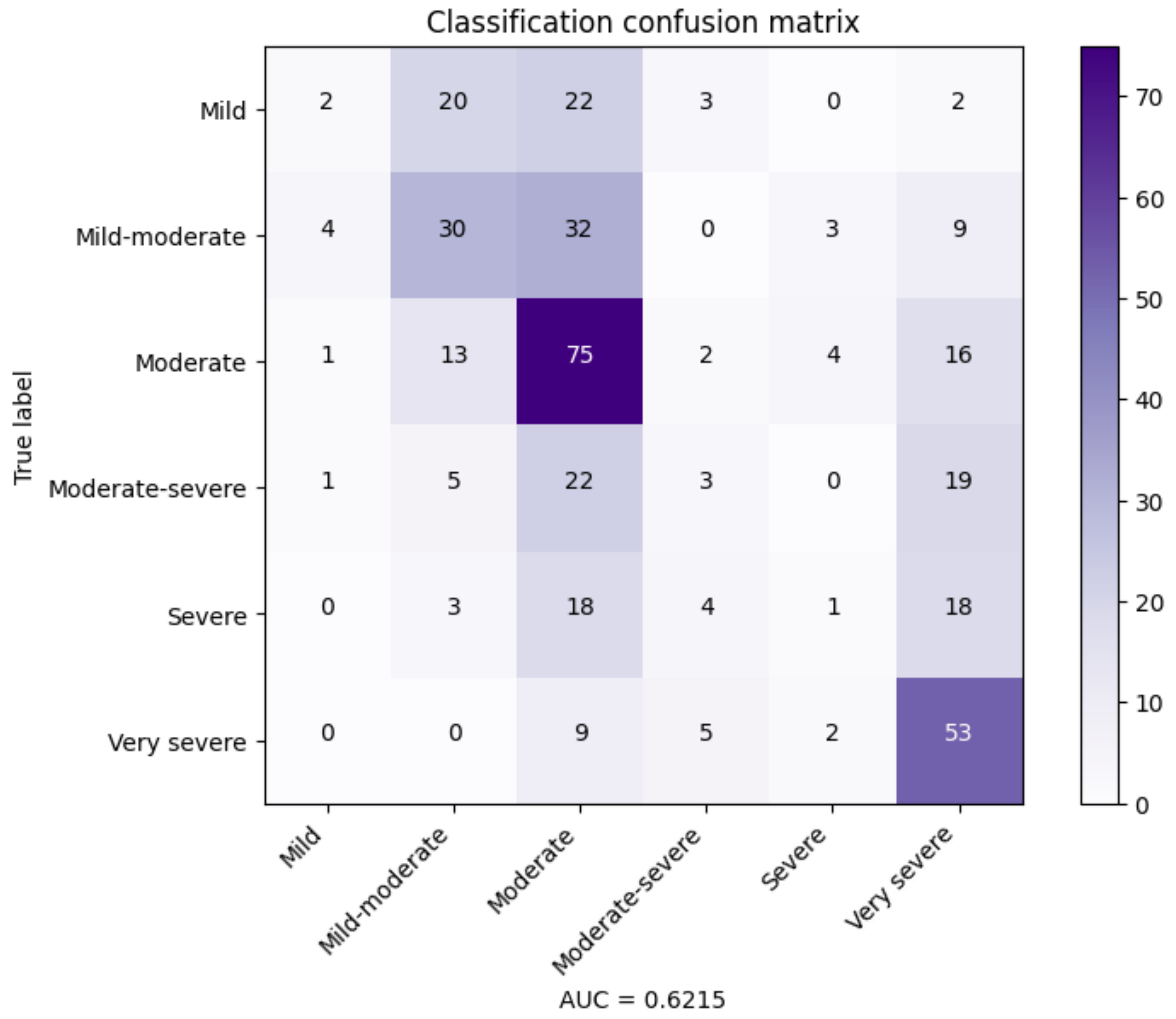
Methodology: severity estimation

Comparison with real-life physician performance:



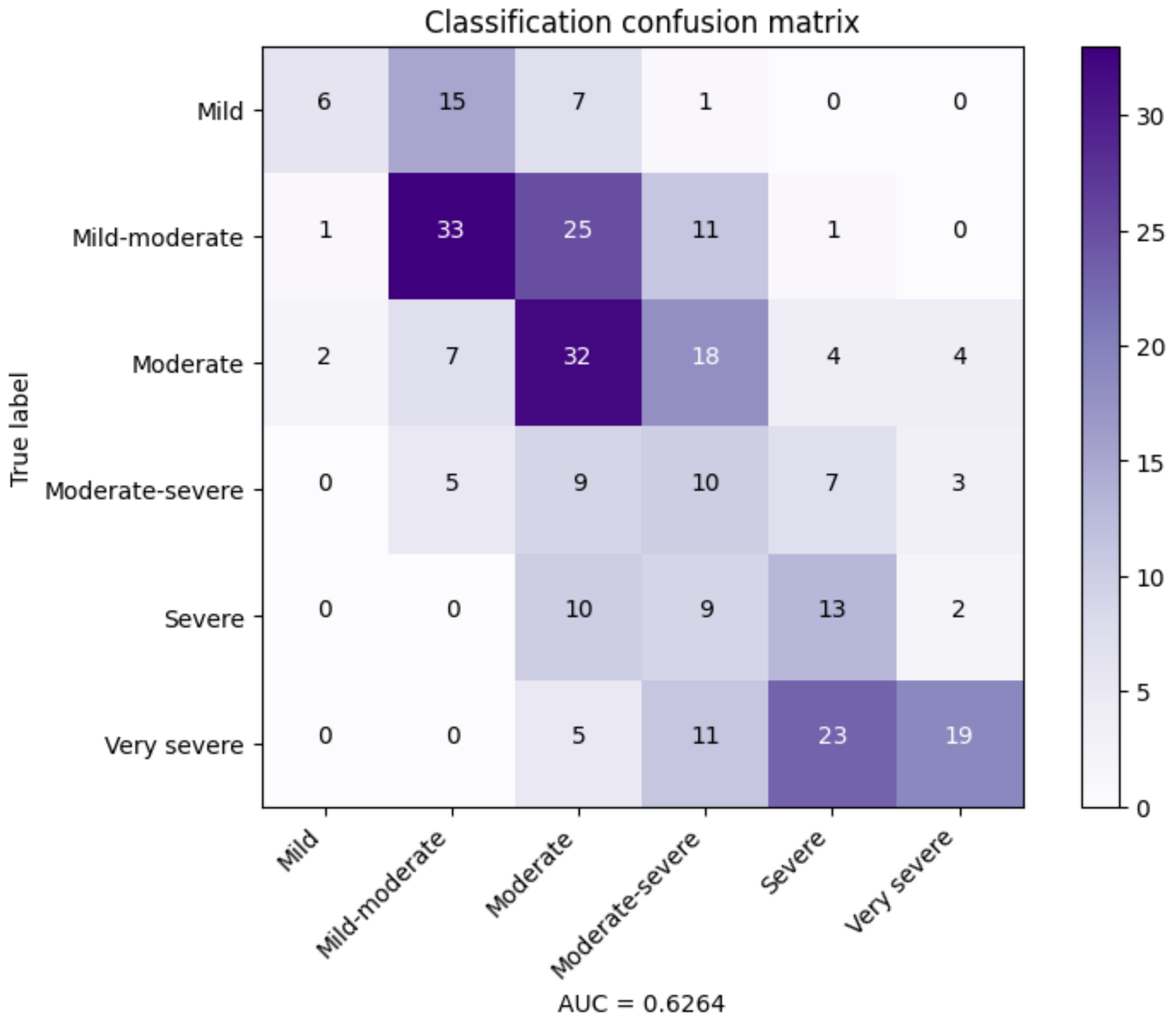
Empirical prediction vs accurate testing

vs.



Vanilla classification

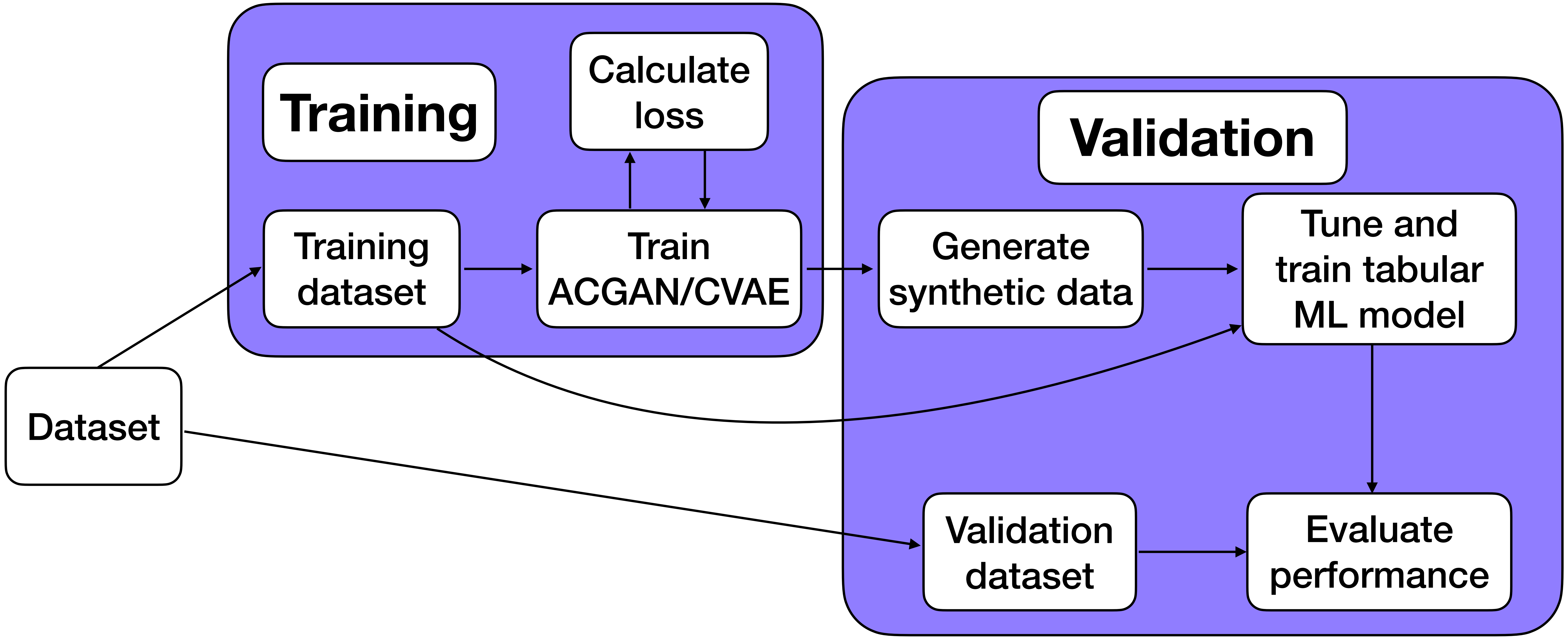
vs.



Distance-based ordinal regression

Methodology: data augmentation

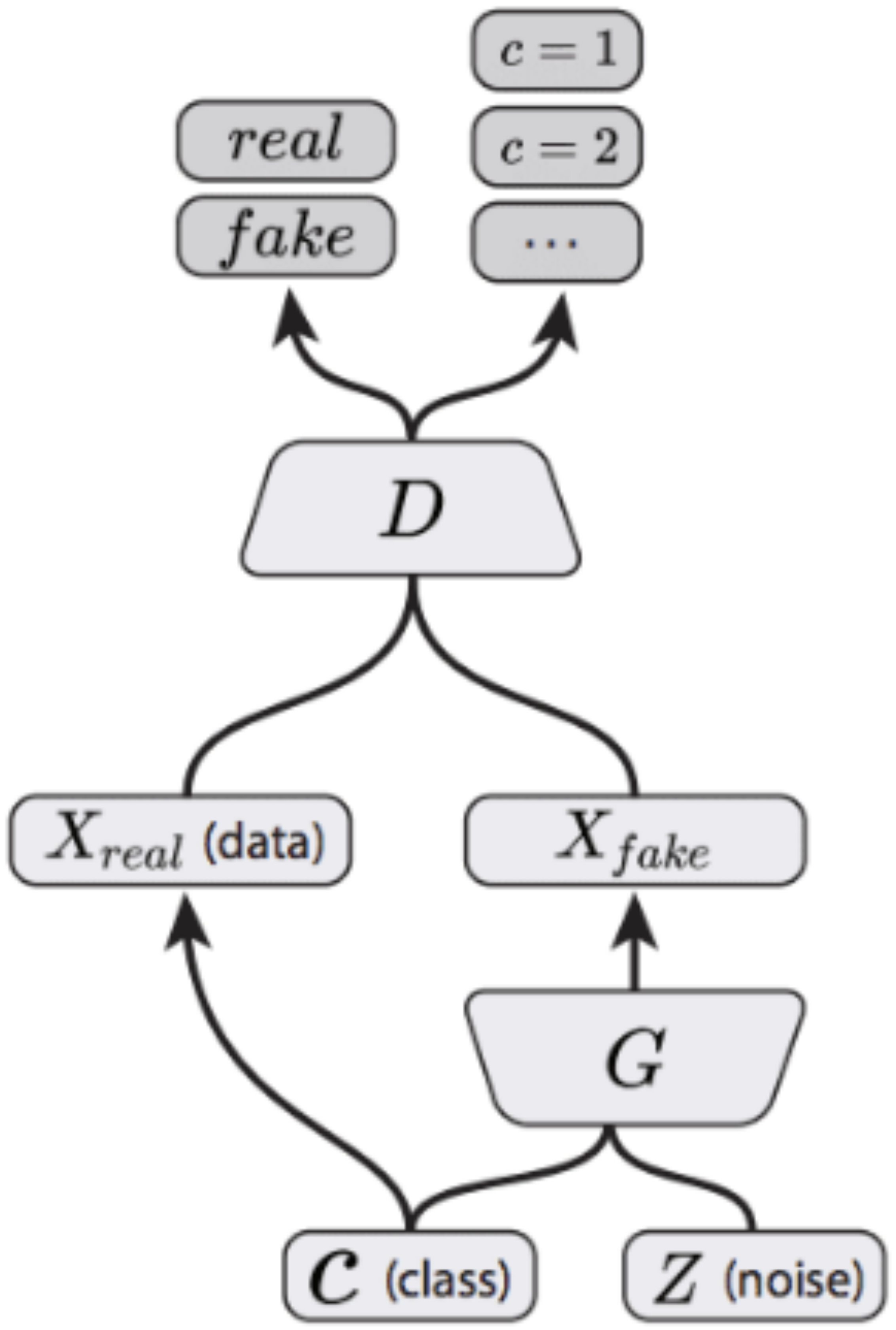
- Dataset is obviously too small for classification
- We conducted experiments on augmenting our dataset conditioned on aphasia type



Methodology: data augmentation

- **Conditional Variational Autoencoder (CVAE)** – a type of generative models that learns a latent representation of input data while taking into account additional conditioning information.

- **Auxiliary Classifier Generational Adversarial Network (ACGAN)** – an extension of the traditional GAN architecture, a combination of two models – a generator and a discriminator, with the first one learns to generate an object on a condition and the other one trying to discriminate both the condition and whether the object was real or generated.

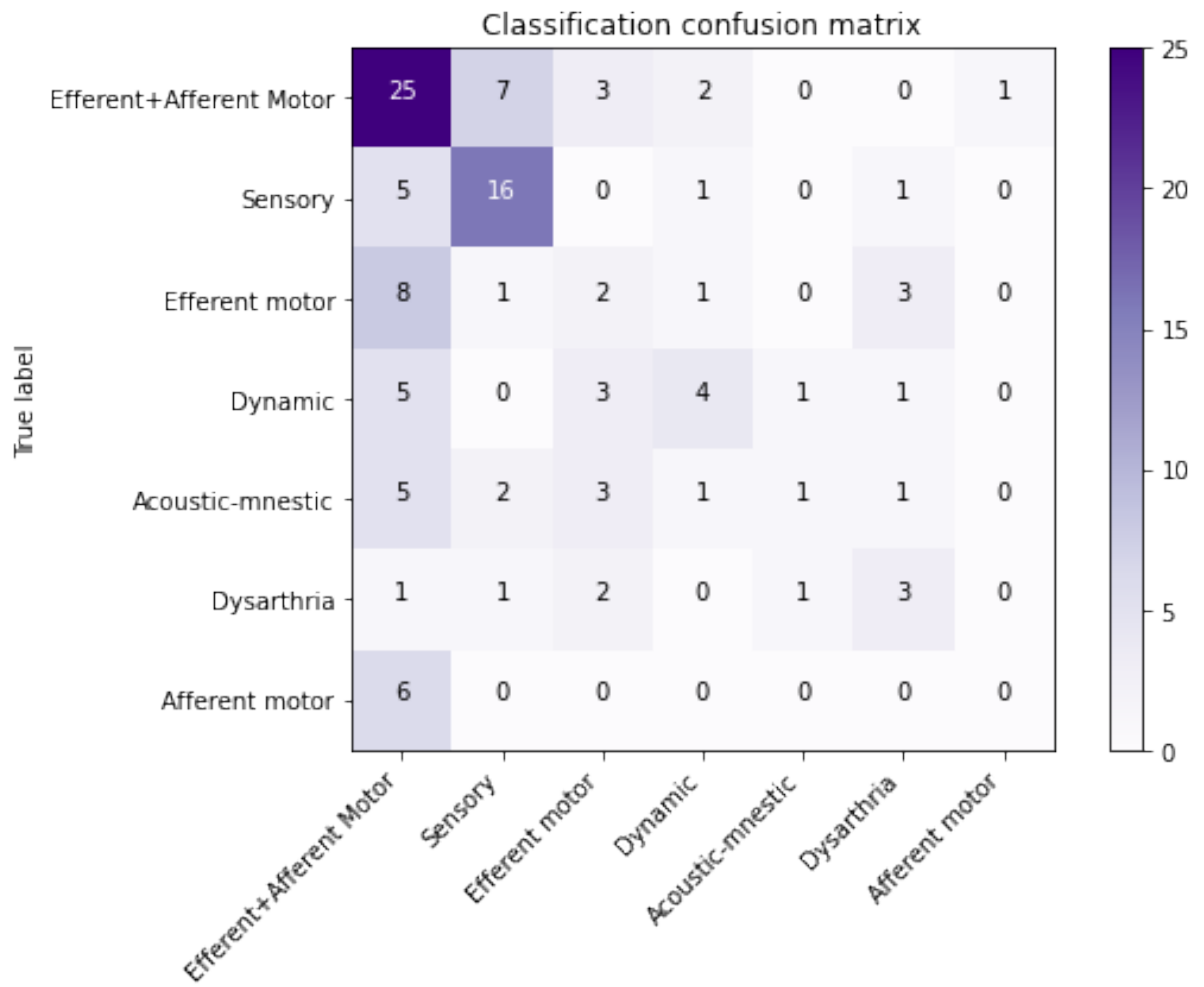


Schematic representation of ACGAN training.

Methodology: data augmentation

Model	Metrics			
	Precision	Recall	F_1-score	ROC-AUC
CVAE on grey (e175)	0.4759	0.4786	0.4723	0.6762
CVAE on grey+demo (e2100)	0.4544	0.4615	0.4530	0.6674
CVAE on white (e1625)	0.5034	0.4957	0.4898	0.6887
CVAE on white+demo (e600)	0.4718	0.4701	0.4668	0.6716
CVAE on both (e1325)	0.5056	0.4786	0.4702	0.6776
CVAE on both+demo (e3350)	0.4684	0.4701	0.4593	0.6709
ACGAN on grey (e4275)	**0.4732**	**0.4872**	**0.4613**	**0.6908**
ACGAN on grey+demo (e1525)	0.4921	0.5043	0.4842	0.6784
ACGAN on white (e4200)	0.4231	0.4786	0.4387	0.6720
ACGAN on white+demo (e3175)	0.4524	0.5043	0.4664	0.6862
ACGAN on both (e800)	0.4881	0.4957	0.4655	0.6688
ACGAN on both+demo (e500)	0.4743	0.5128	0.4702	0.6792

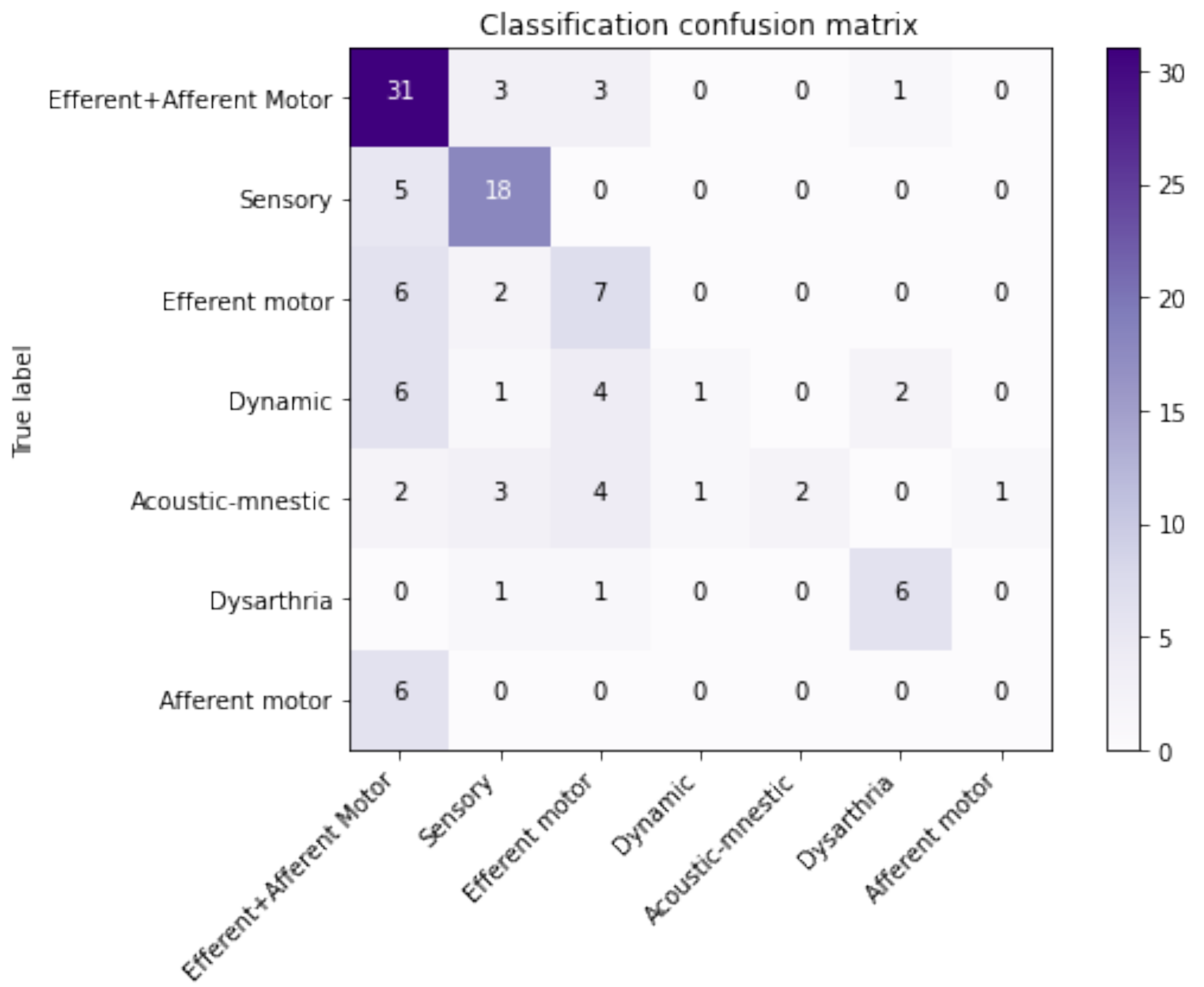
Best performance by MLP validation model on ACGAN trained to generate grey matter features (AUC 0.69).



Methodology: data augmentation

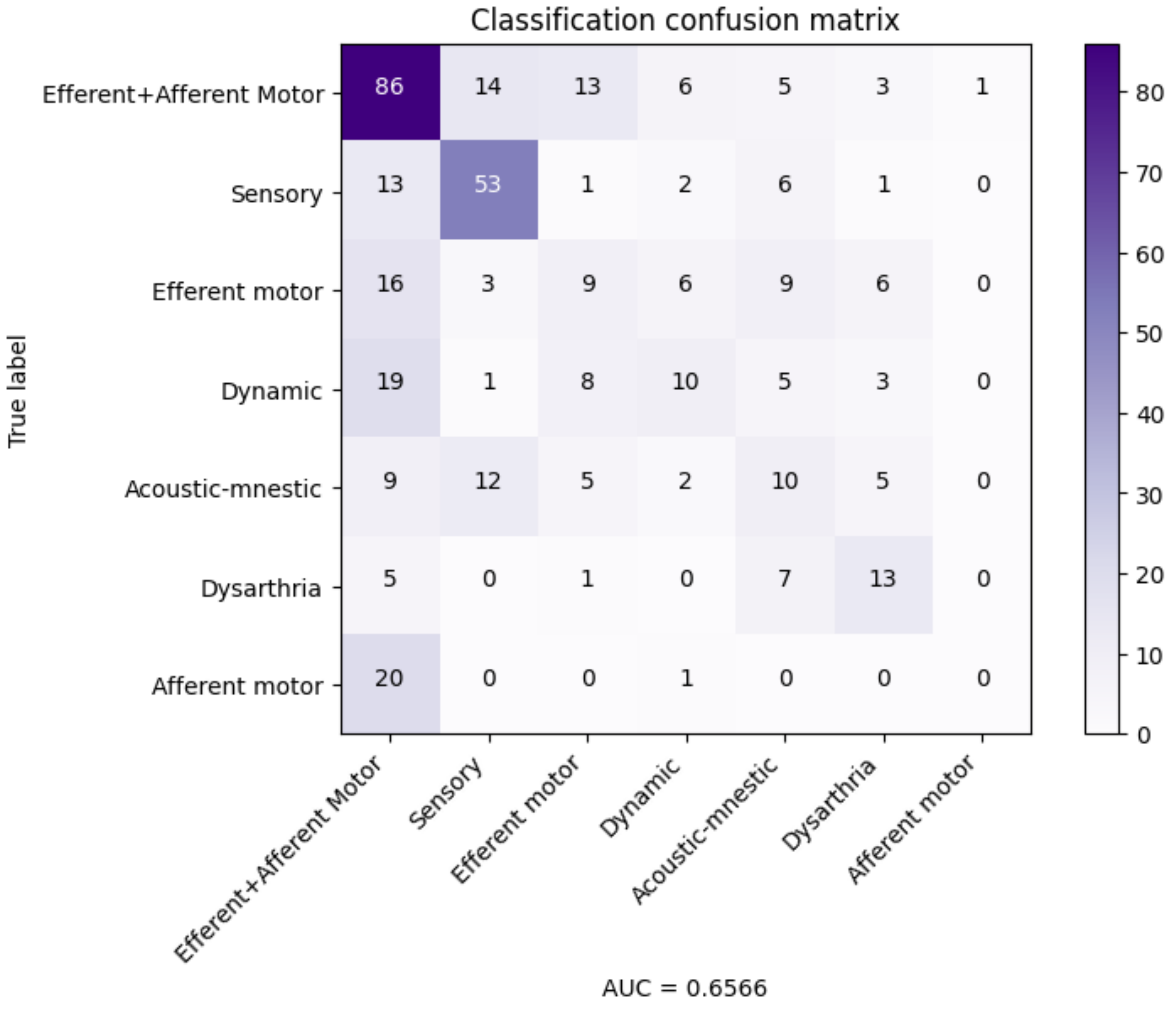
Model	Metrics			
	Precision	Recall	F_1-score	ROC-AUC
CVAE on grey (e2525)	0.5199	0.5385	0.5008	0.6901
CVAE on grey+demo (e1675)	0.4517	0.5299	0.4752	0.6869
CVAE on white (e1625)	0.5266	0.5385	0.5083	0.6956
CVAE on white+demo (e50)	0.5049	0.5299	0.4959	0.6958
CVAE on both (e3300)	0.5216	0.5385	0.5033	0.6974
CVAE on both+demo (e3875)	0.5304	0.5385	0.4982	0.6912
ACGAN on grey (e400)	0.5059	0.5214	0.4620	0.6753
ACGAN on grey+demo (e2225)	0.5144	0.5385	0.5030	0.6921
ACGAN on white (e1050)	0.5196	0.5299	0.4989	0.6880
ACGAN on white+demo (e4225)	0.4992	0.5385	0.4924	0.6901
ACGAN on both (e4550)	0.5501	0.5556	0.5070	0.7017
ACGAN on both+demo (e1900)	**0.5699**	**0.5556**	**0.4986**	**0.7066**

Best performance by RF validation model on ACGAN trained on grey-, white-matter and demographic features (AUC 0.71).



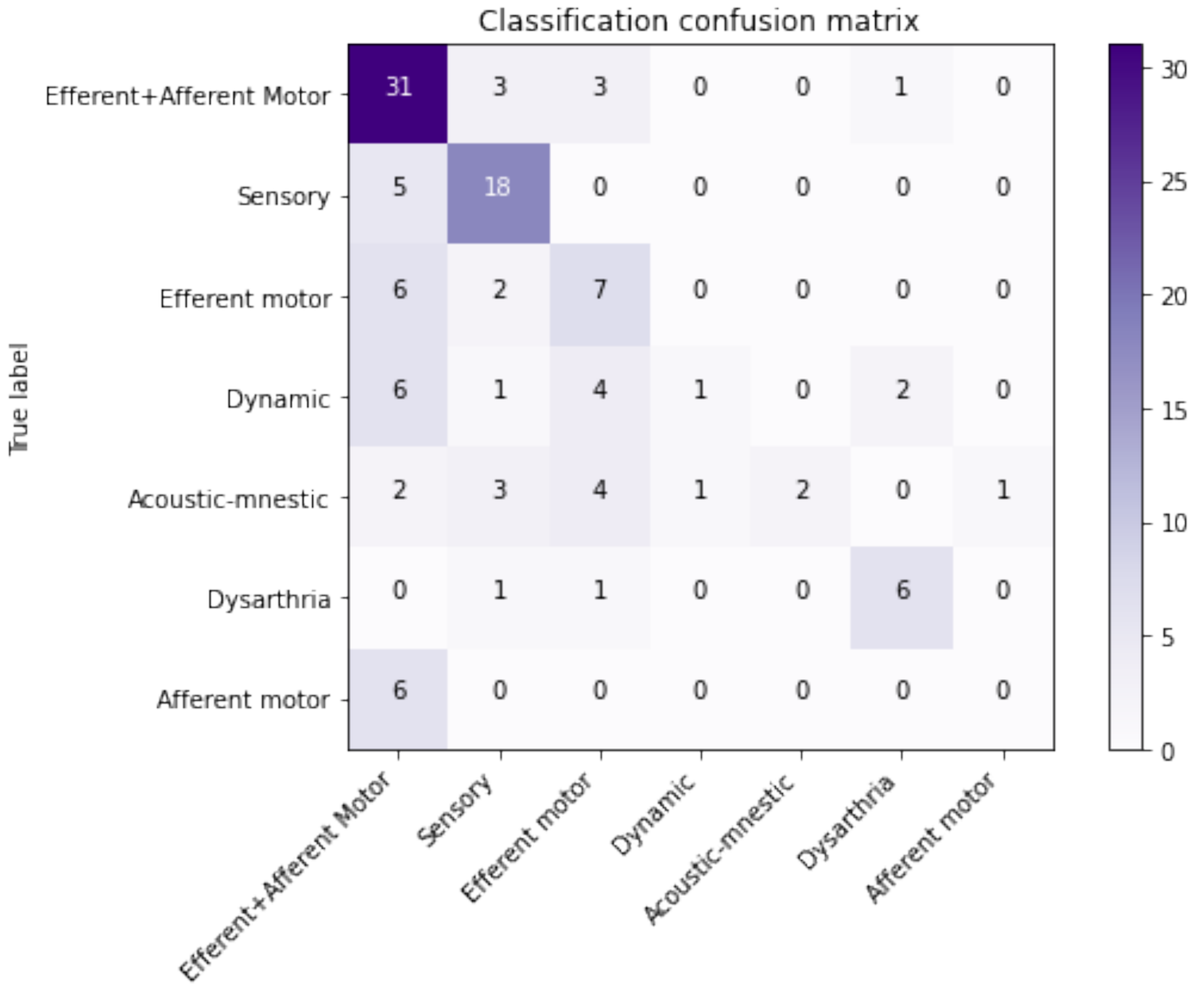
Methodology: data augmentation

Comparison with vanilla classification.



Vanilla classification

vs.



Augmented with ACGAN

Problem statement

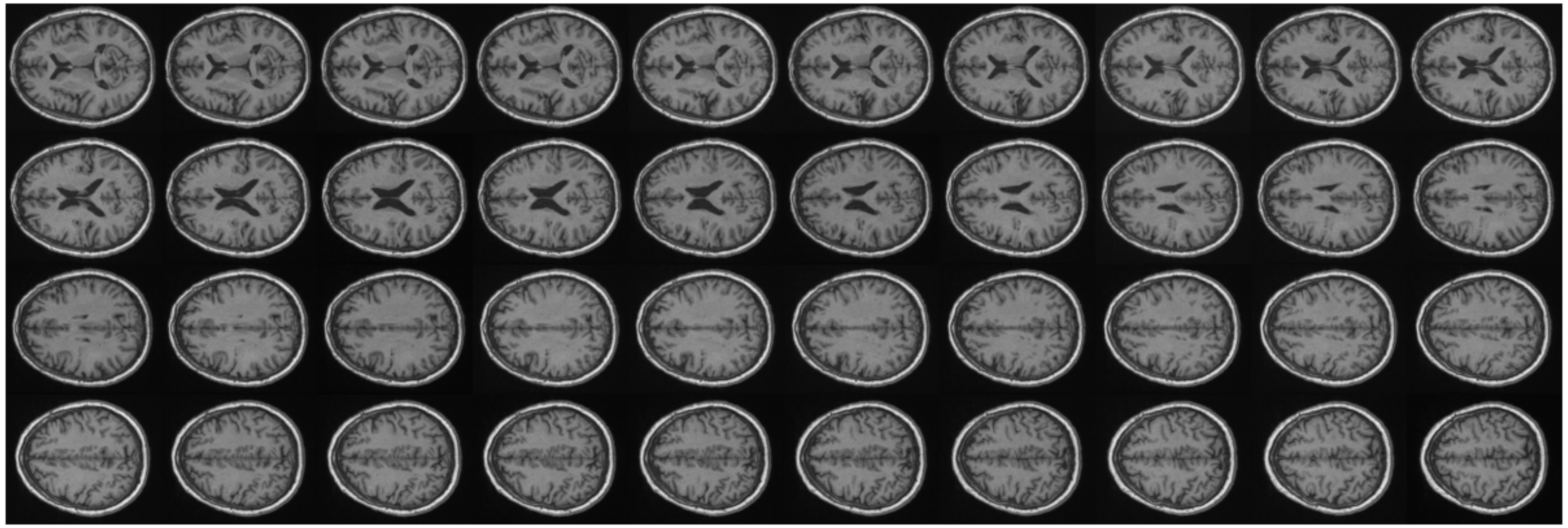
- **(Q1)**: Is it possible to predict aphasia type from MRI data?
- **(Q2)**: Is it possible to predict aphasia severity from MRI data?
- **(Q3)**: How can extremely small dataset size be combatted?
- **(Q4)**: What is the optimal combination of brain MRI features for aphasia type and severity prediction?
- **(Q5)**: What is the optimal representation of target values for the classification of aphasia severity?

Conclusion

- Q1 remains largely unanswered, while results for Q2 show a lot of promise.
- Q3 is partially answered, with generatively augmented datasets leading to slightly better performance.
- Q4 remains unanswered, since no combination of brain MRI features seems to be superior to the others.
- Q5 is clearly answered by the fact that ordinal regression models outperform classification models (at least, in an empirical sense).

Future work: 3D scan classification

- Predictions from full 3-dimensional brain MRI scans.
- Problems: high dimensionality
- Solution: Convolutional Neural Networks (CNNs)



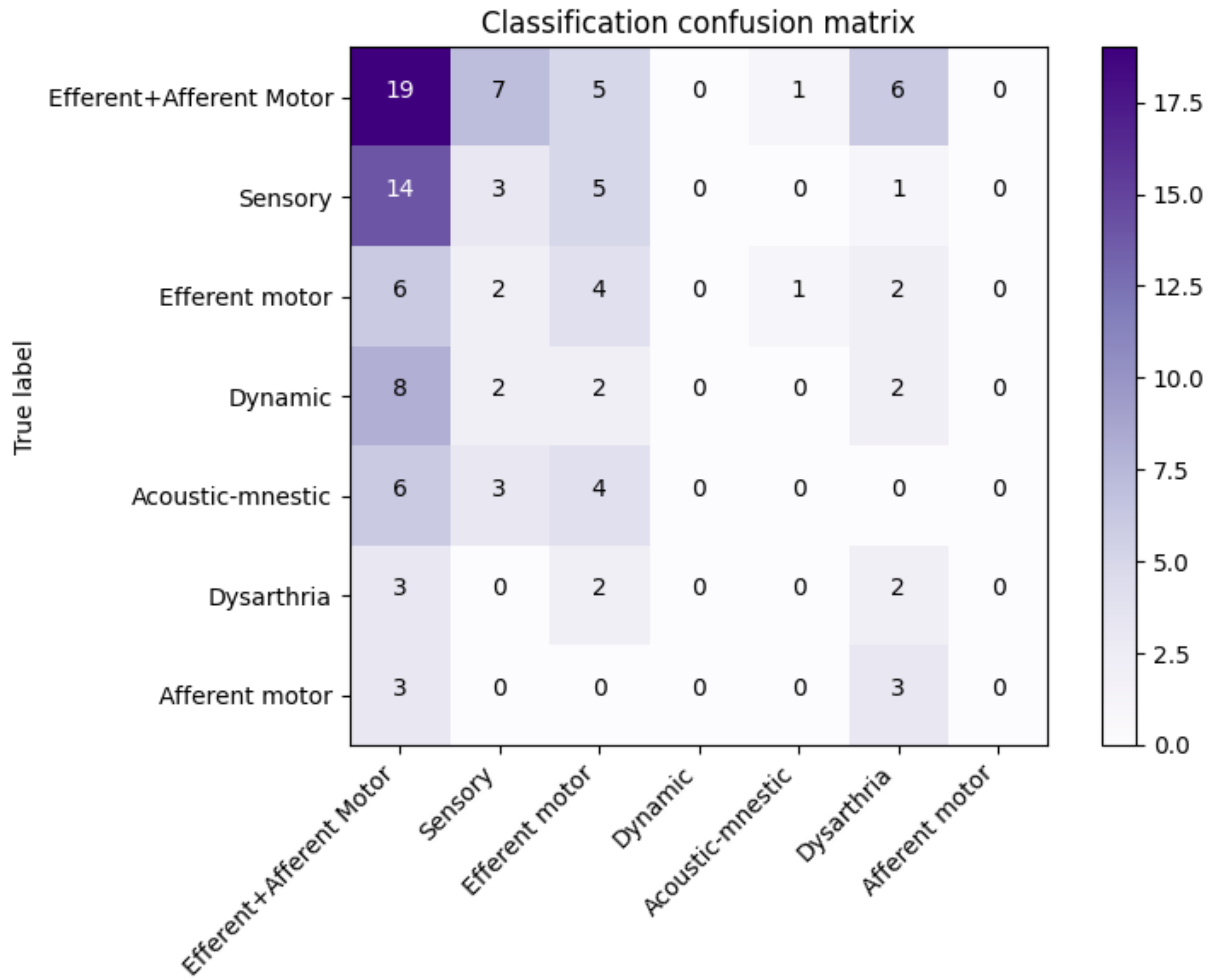
Future work: 3D scan classification

- Convolutional Neural Networks (CNNs) are a type of Deep Learning algorithms that use a series of filters and pooling layers to extract information about an image.
- We constructed 3 datasets:
  - Raw MRI inputs
  - Skull-stripped scans (with use of FSL BET)
  - "Channel" scans: another channel with lesion masks was added to skull stripped scans.
- We use a 4-conv layer "shallow" CNN and a 6-conv layer "deep" CNN.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 182, 218, 182, 1)	0
conv3d (Conv3D)	(None, 180, 216, 180, 64)	1,792
max_pooling3d (MaxPooling3D)	(None, 90, 108, 90, 64)	0
batch_normalization (BatchNormalization)	(None, 90, 108, 90, 64)	256
conv3d_1 (Conv3D)	(None, 88, 106, 88, 64)	110,656
max_pooling3d_1 (MaxPooling3D)	(None, 44, 53, 44, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 44, 53, 44, 64)	256
conv3d_2 (Conv3D)	(None, 42, 51, 42, 128)	221,312
max_pooling3d_2 (MaxPooling3D)	(None, 21, 25, 21, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 21, 25, 21, 128)	512
conv3d_3 (Conv3D)	(None, 19, 23, 19, 256)	884,992
max_pooling3d_3 (MaxPooling3D)	(None, 9, 11, 9, 256)	0
batch_normalization_3 (BatchNormalization)	(None, 9, 11, 9, 256)	1,024
global_average_pooling3d (GlobalAveragePooling3D)	(None, 256)	0
dense (Dense)	(None, 512)	131,584
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Future work: intermediate results

Methods	Metrics			
	Precision	Recall	$F1$-score	ROC-AUC
Random prediction	0.12252 ± 0.07177	0.10564 ± 0.07133	0.10967 ± 0.06923	0.26435 ± 0.24190
raw-cnn3d-shallow	0.28448	0.31034	0.28514	0.55672
raw-cnn3d-deep	0.335	0.33621	0.26918	0.55275
strip-cnn3d-shallow	0.30072	0.31034	0.29528	0.56894
strip-cnn3d-deep	0.29755	0.30172	0.29411	0.56668
channel-cnn3d-shallow	**0.36817**	**0.36207**	**0.3529**	**0.60502**
channel-cnn3d-deep	0.29772	0.31897	0.30471	0.58144



Validation ROC-AUC score of 0.61 achieved.

Appendix: best model hyperparameters

KNN models.

Run name	Hyperparameters				
	$N_{neighbors}$	weights	algorithm	leaf size	p
Type classification					
grey-knn	16	distance	kd_tree	2	1.00000
grey-d-knn	16	distance	ball_tree	32	1.00000
white-knn	16	distance	ball_tree	32	3.87216
white-d-knn	13	distance	ball_tree	5	1.00000
both-knn	13	distance	ball_tree	7	2.86851
both-d-knn	16	distance	kd_tree	2	1.00000
Severity classificaion					
grey-knn	8	distance	ball_tree	31	4.83665
grey-d-knn	7	distance	ball_tree	2	1.00000
white-knn	16	distance	ball_tree	32	1.00000
white-d-knn	14	uniform	ball_tree	8	1.23732
both-knn	16	distance	kd_tree	32	5.00000
both-d-knn	16	distance	ball_tree	3	1.00000
Severity ordinal regression					
grey-knn	13	distance	kd_tree	32	4.39064
grey-d-knn	15	distance	ball_tree	32	1.00000
white-knn	16	distance	ball_tree	2	1.00000
white-d-knn	13	distance	ball_tree	2	1.00000
both-knn	12	distance	ball_tree	29	1.02754
both-d-knn	16	distance	ball_tree	32	1.00000
ASA ordinal regression					
grey-knn	14	distance	ball_tree	23	1.75118
grey-d-knn	16	distance	ball_tree	3	1.40529
white-knn	16	distance	kd_tree	26	1.01317
white-d-knn	10	distance	kd_tree	2	1.00000
both-knn	16	distance	ball_tree	13	2.04484
both-d-knn	16	uniform	ball_tree	32	1.00000
Frank-Hall ordinal regression					
grey-knn	10	distance	ball_tree	2	3.89023
grey-d-knn	10	distance	ball_tree	14	4.24529
white-knn	14	distance	ball_tree	32	4.78469
white-d-knn	12	distance	kd_tree	32	1.36219
both-knn	15	uniform	kd_tree	12	1.43292
both-d-knn	7	uniform	kd_tree	24	1.27656

MLP models.

Run name	Hyperparameters				
	$N_{neurons}$	N_{epochs}	learning rate	activation	optimizer
Type classification					
grey-mlp	610	410	0.00040	identity	adam
grey-d-mlp	1011	175	0.00018	tanh	adam
white-mlp	16	1577	0.00100	tanh	adam
white-d-mlp	1024	3494	0.00072	identity	adam
both-mlp	1024	5000	0.00100	identity	sgd
both-d-mlp	931	881	0.00003	relu	adam
Severity classificaion					
grey-mlp	470	3462	0.00069	identity	sgd
grey-d-mlp	1024	5000	0.00017	identity	adam
white-mlp	16	5000	0.00100	identity	adam
white-d-mlp	1024	5000	0.00100	logistic	adam
both-mlp	864	2959	0.00067	tanh	sgd
both-d-mlp	1024	3591	0.00100	logistic	adam
Severity ordinal regression					
grey-mlp	932	4112	0.00100	logistic	adam
grey-d-mlp	16	755	0.00100	tanh	adam
white-mlp	418	4878	0.00015	identity	adam
white-d-mlp	938	3308	0.00060	logistic	adam
both-mlp	732	3693	0.00100	tanh	sgd
both-d-mlp	477	1161	0.00100	logistic	adam
ASA ordinal regression					
grey-mlp	1024	2168	0.00001	relu	sgd
grey-d-mlp	349	664	0.00055	logistic	sgd
white-mlp	634	3216	0.00002	logistic	sgd
white-d-mlp	897	719	0.00009	tanh	sgd
both-mlp	994	3649	0.00004	tanh	sgd
both-d-mlp	364	4483	0.00001	relu	sgd
Frank-Hall ordinal regression					
grey-mlp	612	4787	0.00023	logistic	adam
grey-d-mlp	457	3941	0.00005	identity	adam
white-mlp	1001	4815	0.00035	identity	adam
white-d-mlp	873	1343	0.00012	relu	adam
both-mlp	699	977	0.00005	identity	adam
both-d-mlp	938	1196	0.00057	identity	adam

Appendix: best model hyperparameters

RF models.

Run name	Hyperparameters				
	N_{trees}	min_{split}	min_{leaf}	$max_{features}$	max_{depth}
Type classification					
grey-rf	1685	10	3	sqrt	64
grey-d-rf	7911	13	5	sqrt	12
white-rf	1525	10	3	sqrt	43
white-d-rf	7140	16	1	sqrt	6
both-rf	4506	3	12	sqrt	61
both-d-rf	7397	5	1	log2	10
Severity classificaion					
grey-rf	819	4	10	sqrt	60
grey-d-rf	2934	7	2	sqrt	32
white-rf	5979	13	7	log2	7
white-d-rf	7640	2	10	log2	59
both-rf	2784	16	16	sqrt	58
both-d-rf	7803	6	16	sqrt	15
Severity ordinal regression					
grey-rf	7098	3	7	sqrt	5
grey-d-rf	1809	5	1	sqrt	30
white-rf	7037	4	1	sqrt	3
white-d-rf	1751	2	1	sqrt	36
both-rf	5048	15	14	sqrt	57
both-d-rf	4723	10	7	sqrt	60
ASA ordinal regression					
grey-rf	7542	8	6	sqrt	46
grey-d-rf	68	4	1	sqrt	53
white-rf	4077	3	11	log2	32
white-d-rf	8192	4	1	sqrt	20
both-rf	3327	5	6	sqrt	54
both-d-rf	5816	2	2	sqrt	10
Frank-Hall ordinal regression					
grey-rf	2872	16	9	sqrt	6
grey-d-rf	4259	15	11	sqrt	25
white-rf	6632	4	3	log2	44
white-d-rf	3574	3	12	log2	31
both-rf	5956	8	15	log2	45
both-d-rf	4713	5	15	sqrt	18

GB models.

Run name	Hyperparameters					
	learning rate	N_{trees}	min_{split}	min_{leaf}	$max_{features}$	max_{depth}
Type classification						
grey-gb	0.18957	2408	6	3	sqrt	7
grey-d-gb	0.65899	2252	12	10	log2	2
white-gb	0.45862	1490	12	1	log2	56
white-d-gb	0.00100	3728	3	1	log2	2
both-gb	0.40659	8192	16	1	log2	64
both-d-gb	0.27485	1329	6	11	sqrt	12
Severity classificaion						
grey-gb	0.00100	4809	16	14	sqrt	64
grey-d-gb	0.00100	2338	10	16	sqrt	62
white-gb	0.22214	7819	11	13	sqrt	38
white-d-gb	0.21468	1071	14	3	sqrt	17
both-gb	0.77654	2599	11	7	log2	14
both-d-gb	0.26006	4472	5	15	sqrt	25
Severity ordinal regression						
grey-gb	0.00100	2928	12	16	log2	2
grey-d-gb	0.00100	7302	16	16	sqrt	26
white-gb	0.00100	3425	5	1	sqrt	48
white-d-gb	0.00100	8192	2	3	sqrt	2
both-gb	0.00100	6320	2	16	sqrt	2
both-d-gb	0.00100	8192	7	16	log2	64
ASA ordinal regression						
grey-gb	0.00100	5063	16	1	sqrt	2
grey-d-gb	0.00100	8192	16	16	log2	45
white-gb	0.00100	4639	2	16	log2	2
white-d-gb	0.00100	7182	16	16	sqrt	64
both-gb	0.00100	8192	16	6	log2	2
both-d-gb	0.00100	7168	16	16	sqrt	64
Frank-Hall ordinal regression						
grey-gb	0.69264	1183	6	3	log2	7
grey-d-gb	0.00133	5723	15	13	sqrt	8
white-gb	0.02155	398	11	5	log2	45
white-d-gb	0.04465	7566	7	6	log2	9
both-gb	0.01634	292	2	14	log2	38
both-d-gb	0.00197	4614	10	14	log2	51

Appendix: best model hyperparameters

SVM models.

Run name	C	γ
Type classification		
grey-svm	32768.0	0.00003
grey-d-svm	13572.20860	0.00016
white-svm	25258.26463	0.00005
white-d-svm	32768.0	0.00003
both-svm	576.84575	0.00133
both-d-svm	32768.0	0.00026
Severity classificaion		
grey-svm	656.60479	0.00134
grey-d-svm	111.68176	0.01253
white-svm	33.26394	0.02161
white-d-svm	5497.37007	0.00033
both-svm	4.59077	0.03841
both-d-svm	22044.74881	0.00004
Severity ordinal regression		
grey-svm	14.01685	0.03127
grey-d-svm	6.13521	0.12058
white-svm	344.02779	0.00032
white-d-svm	27.73901	0.02298
both-svm	43.00607	0.01155
both-d-svm	26675.51170	0.00003
ASA ordinal regression		
grey-svm	4006.43253	0.00135
grey-d-svm	1212.30772	0.00946
white-svm	104.67072	0.07563
white-d-svm	6859.77902	0.00171
both-svm	4857.71000	0.00090
both-d-svm	32768.00000	0.00052
Frank-Hall ordinal regression		
grey-svm	295.09284	0.00534
grey-d-svm	290.97968	0.00097
white-svm	4871.67216	0.00562
white-d-svm	200.24807	0.00783
both-svm	296.91327	0.01304
both-d-svm	110.86846	0.00636

Appendix: generative model architectures

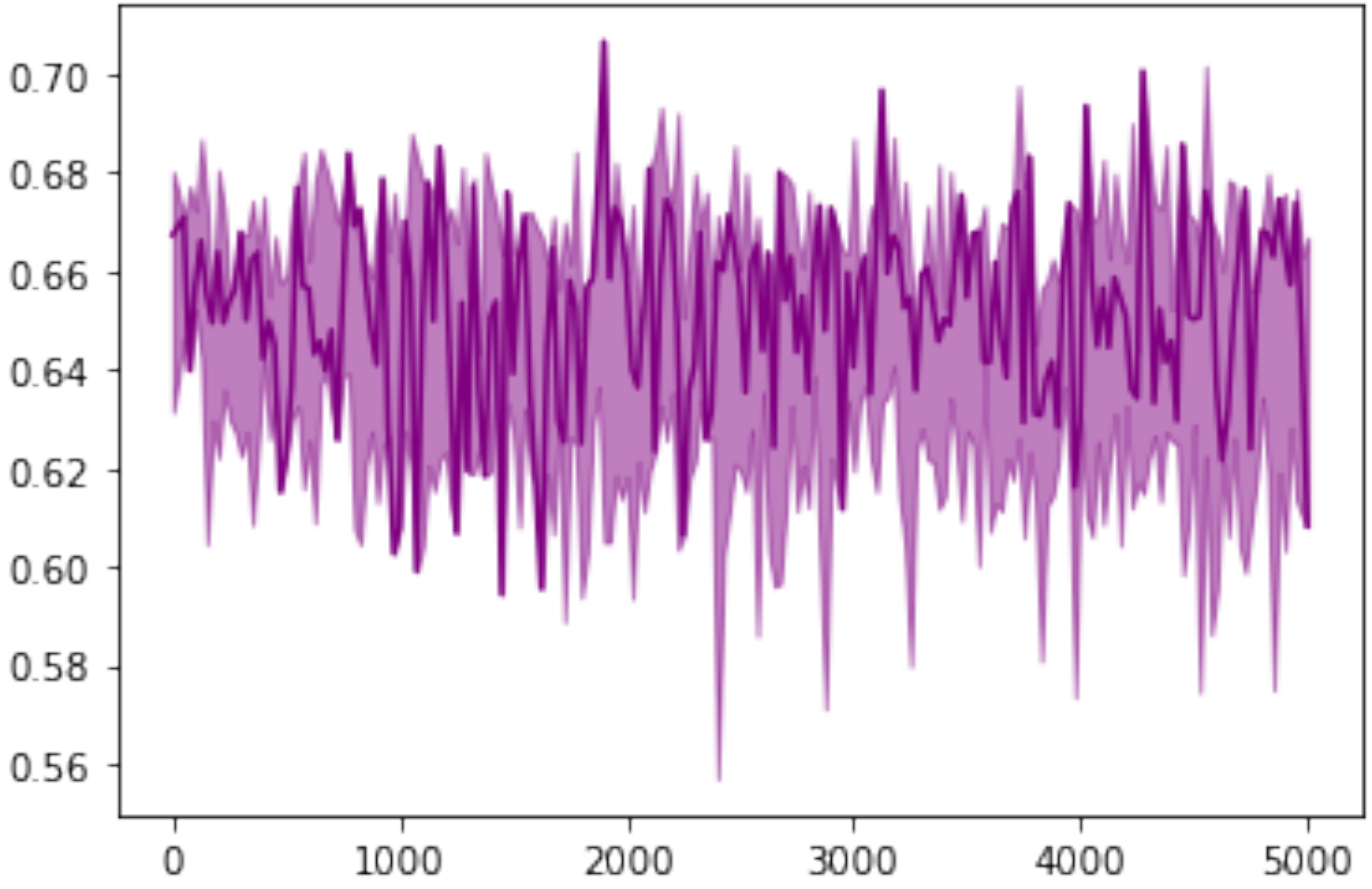
Layer (type)	Output Shape	Param #
ac_generator (ACGenerator)	?	2,317
└ concatenate (Concatenate)	?	0
└ sequential (Sequential)	?	2,317
└ dense (Dense)	(4, 16)	288
└ dense_1 (Dense)	(4, 32)	544
└ dense_2 (Dense)	(4, 45)	1,485
ac_discriminator (ACDiscriminator)	?	2,136
└ sequential_1 (Sequential)	?	2,000
└ dense_3 (Dense)	(4, 32)	1,472
└ dense_4 (Dense)	(4, 16)	528
└ dense_5 (Dense)	?	17
└ dense_6 (Dense)	?	119

ACGAN summary.

Layer (type)	Output Shape	Param #
encoder (Encoder)	?	2,564
└ concatenate (Concatenate)	?	0
└ sequential (Sequential)	?	2,224
└ dense (Dense)	(4, 32)	1,696
└ dense_1 (Dense)	(4, 16)	528
└ dense_2 (Dense)	?	170
└ dense_3 (Dense)	?	170
decoder (Decoder)	?	2,317
└ concatenate_1 (Concatenate)	?	0
└ sequential_1 (Sequential)	?	2,317
└ dense_4 (Dense)	(4, 16)	288
└ dense_5 (Dense)	(4, 32)	544
└ dense_6 (Dense)	(4, 45)	1,485

CVAE summary.

Appendix: training graphs



RF loss for ACGAN validation.