SeBA: Semi-supervised few-shot learning via Separated-at-Birth Alignment for tabular data

Kacper Jurek[1,2,†] **Wojciech Batko**[1,†] **Marek Śmieja**[1] **Marcin Przewięźlikowski**[1,3 *]
[1] Jagiellonian University, Faculty of Mathematics and Computer Science
[2] AGH University of Krakow
[3] Jagiellonian University, Doctoral School of Exact and Natural Sciences

Abstract

Learning from scarce labeled data with a larger pool of unlabeled samples, known as semi-supervised few-shot learning (SS-FSL), remains critical for applications involving tabular data in domains like medicine, finance, and science. The existing SS-FSL methods often rely on self-supervised learning (SSL) frameworks developed for vision or language, which assume the availability of a natural form of data augmentations. For tabular data, defining meaningful augmentations is non-trivial and can easily distort semantics, limiting the effectiveness of conventional SSL. In this work, we rethink SSL for tabular data and propose Separated-at-Birth Alignment (SeBA), a joint-embedding framework for SS-FSL that eliminates the dependence on augmentations. Our core idea is to separate the data into two independent, but complementary views and align the representations of one view to mirror the nearest-neighbor correspondence of the data in the second view. A type-aware separation scheme ensures robust handling of mixed categorical and numerical attributes, while a lightweight architecture with ensemble aggregation improves generalization and reduces sensitivity to misselection of model parameters. An experimental study conducted in various benchmark datasets demonstrates that SeBA often achieves state-of-the-art performance in tabular SS-FSL, opening a new avenue for SSL paradigm in the domain of tabular data.

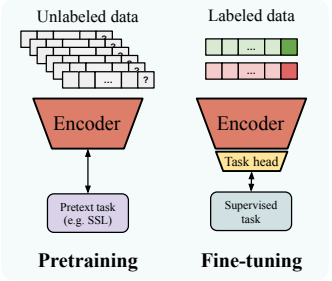
1 Introduction

Learning with a limited amount of labeled data remains a fundamental challenge in machine learning and data analysis. Although collecting additional annotations is costly, access to raw unlabeled data is often inexpensive. This imbalance motivates the practical setting of semi-supervised few-shot learning (SS-FSL), where classification must be performed with scarce labeled data and a large pool of unlabeled samples.

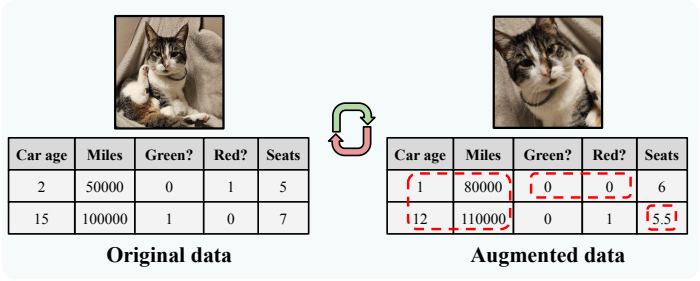
The tabular modality poses unique challenges for typical SS-FSL methods, which rely on pretraining with unlabeled data followed by fine-tuning on a few labeled examples (see Figure 1a). State-of-the-art pretraining approaches often use self-supervised learning (SSL), which encourages models to produce similar representations for semantically related *positive pairs* while avoiding collapse to trivial solutions [Wang and Isola, 2020]. Such pairs are usually created by sampling multiple augmentations of the same data point. In CV, augmentations are straightforward: image transformations such as cropping, rotation, or color jittering yield valid semantically consistent samples. However, for tabular data, there is no natural way to define proper augmentations. Poorly chosen transformations, such as zero masking, Gaussian noise, or sampling features from marginal distribution, can distort semantics

*Corresponding author: marcin.przewiezlikowski@doctoral.uj.edu.pl
† – K.J. and W.B. contributed equally.

AI for Tabular Data workshop at EurIPS 2025.



(a) Semi-supervised Few-shot Learning (SS-FSL) setup. The model is pretrained on a large pool of unlabeled data and fine-tuned on a few labeled examples.



(b) While semantic-preserving augmentations are straightforward to define for modalities such as images, they must be designed much more carefully for tabular data. Improperly designed augmentations can generate samples from outside the data manifold (decreasing car age, but increasing mileage), obfuscate the categorical values (neither option marked as true), or assign incorrect values (number of car seats must be an integer).

Figure 1: Typical Semi-supervised Few-shot Learning (SS-FSL) approaches **(a)** pretrain their representations on large pools of unlabeled data, usually via Self-supervised Learning (SSL). In the case of tabular data, the state-of-the-art augmentation-based SSL approaches cannot be directly applied, due to challenges with defining proper augmentations **(b)**.

or even generate out-of-distribution samples (see Figure 1b), ultimately undermining the effectiveness of SSL.

In this paper, we rethink SSL for tabular data and show that, with carefully designed positive pairs, it yields significantly stronger tabular representations than previously assumed. Instead of aligning the representations of positive pairs created via augmentations, we introduce Separated-at-Birth Alignment (SeBA), illustrated in Figure 2. SeBA projects data into two complementary subspaces: feature and target views. The model is then pretrained by identifying nearest-neighbor correspondences in the target view, using only the information encoded in the feature view. This replaces the problematic reliance on augmentations with a nearest-neighbor graph. To properly handle mixed data types, we employ a type-aware separation scheme that accounts for both categorical and numerical features, ensuring that the resulting projections remain semantically meaningful.

SeBA requires far less dataset-specific knowledge than hand-crafting augmentations, making it practical and easy to apply. Moreover, the model pretrained by SeBA is lightweight and thus less prone to overfitting for small datasets. Finally, the applied ensemble strategy minimizes the need for the selection of model's parameters, which is crucial in the FSL scenario. The experimental results clearly show that SeBA generalizes effectively in a wide variety of tabular datasets, achieving impressive results in few-shot classification.

2 Method

Overview. The design of SeBA follows Self-supervised Joint-Embedding Architectures (JEAs), which learn through aligning semantically-related positive pairs of data in the representation space [Chen et al., 2020, He et al., 2020] and pushing away the unrelated ones. Unlike conventional JEAs, SeBA does not rely on sampling multiple data augmentations for the construction of positive pairs, which is problematic for tabular data. Instead, in every minibatch, SeBA separates the tabular records "at birth" into two random complementary views, which we denote as feature and target views. The model is trained to align the data representations of feature views according to the similarity graph induced by the target views, therefore, learning semantically meaningful correspondences without relying on augmentations. We outline the schema of SeBA in Figure 2 and describe its components in detail below.

Separating the data at-birth. Let $\mathcal{B}$ be a minibatch of (unlabeled) data points, and let $m \in \{0,1\}^D$ be a binary mask vector sampled once per batch that defines features included in each view. The proportion of 1-s in m is controlled by the hyperparameter named *separation ratio*. For every $x \in \mathcal{B}$, we create a feature view:

$$x_f = x \odot (1 - m) \tag{1}$$

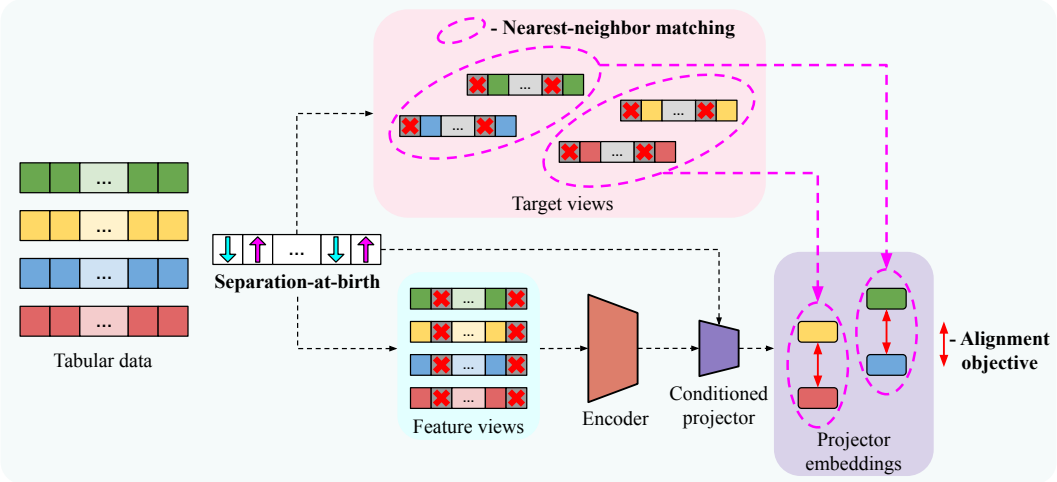


Figure 2: Representation learning via Separated-at-Birth Alignment (SeBA). In each minibatch, we separate the columns of tabular data "at-birth" into two complementary and independent subsets, which define target an feature views. Instead of augmentation, semantically-related positive pairs for a pretraining contrastive task are defined using the nearest neighbor relation in their target view (upper side). The encoder is trained to create the representation of the feature view, which aligns the positive pairs defined in the target view (bottom side). To allow the encoder to create general data representation, SeBA uses a conditioned projector to build a task-specific representation for every separation mask.

and a target view:

$$x_t = x \odot m, \tag{2}$$

where $\odot$ is an element-wise multiplication, and $x_t, x_f \in \mathbb{R}^D$. As such, the target and feature views are complementary and independent.

Target similarity graph. We use target views to define the positive data pairs with respect to the sampled mask m. For each sample x in the batch, we identify its nearest neighbor in terms of the target views:

$$x' = \arg \min_{a \in \mathcal{B} \setminus \{x\}} d(x_t, a_t). \tag{3}$$

In other words, m defines positive pairs (x, x') based on the nearest-neighbor graph defined in the target view.

Alignment objective. Finally, we train the encoder to align the feature-view representations to match the nearest-neighbor relation defined in the target view. For this purpose, we first construct the encoder representations of the feature views:

$$h = f(x_f); h' = f(x'_f) \tag{4}$$

Observe that the feature views x_f do not contain unequivocal information about the data separation scheme m and, as such, it may not be able to solve the alignment task on their own. To address this problem, SeBA incorporates a conditioned projector $\pi : \mathbb{R}^E \times \mathbb{R}^D \to \mathbb{R}^P$, where P is the embedding shape of π [Przewięźlikowski et al., 2024, Bordes et al., 2023]. The projector combines the general feature view representation of the encoder and information about how the data was separated (i.e. the mask vector m). The projector transforms the encoder representation into the task-specific latent space:

$$z = \pi(h, m); z' = \pi(h', m), \tag{5}$$

in which we optimize the alignment objective. The objective takes form of the InfoNCE loss, which pulls together the positive representation pairs, and pushes away the unrelated ones [Oord et al., 2018]:

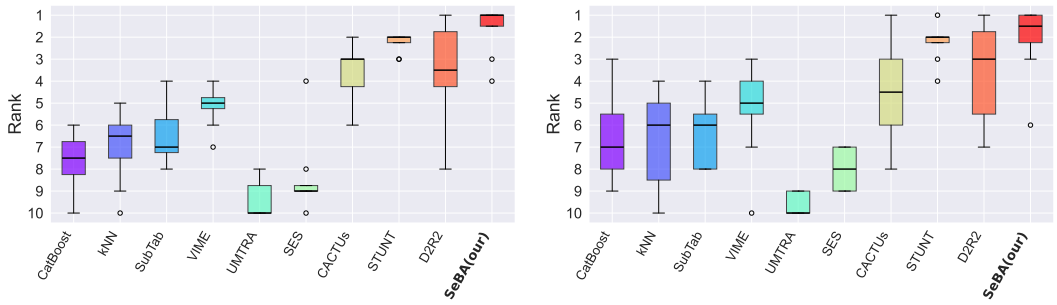
$$\mathcal{L}(x) = -\log \frac{\exp(d(z, z'))}{\Sigma_{a \in \mathcal{B}, a \neq x} \exp\Big(d\Big(z, \pi\big(f(a_f), m\big)\Big)\Big)} \tag{6}$$

Because SeBA trains on numerous separation schemes (multiple mask vectors), the encoder adapts repeatedly to different target matching objectives. This exposure yields robust representations that generalize well to downstream tasks.

3 Experiments

Few-shot learning efficacy. We evaluate SeBA in terms of its performance in downstream few-shot learning tasks. We compare our method with the state-of-the-art SS-FSL methods, STUNT [Nam et al., 2023], and D2R2 [Liu et al., 2024], which we run on exactly the same splits as SeBA. Moreover, we also report the results of 9 other baselines from [Nam et al., 2023]. They represent the best supervised, self-supervised and meta-learning approaches (see Appendix A.3 for more details).

We rank the performance of models in terms of 1- and 5-shot classification on 8 common datasets used previously by [Nam et al., 2023, Liu et al., 2024], and report the rank distributions in Figure 3. It is evident that SeBA is significantly better ranked than competitive methods. We report the detailed classification results in Appendix B.1



(a) 1-shot classification accuracy

(b) 5-shot classification accuracy

Figure 3: Box-plots of 1-shot (left) and 5-shot (right) classification ranks of benchmarked approaches. SeBA is the most consistently high-ranking method.

Detailed analysis of SeBA. We refer to Appendix B.2 for a detailed ablation study of the hyperparameters of SeBA. Moreover, in Appendix B.3 we demonstrate how the SeBA pretraining objective corresponds to the downstream few-shot classification tasks, further justifying its design.

4 Conclusion

In this paper, we introduce Separated-at-Birth Alignment (SeBA), a novel Semi-supervised Few-Shot Learning framework designed for tabular data. SeBA uses the powerful Joint-Embedding Architecture (JEA) paradigm to pretrain its representations, while avoiding the problematic need to for manual data augmentation design – the issue that has prevented the use of JEAs for tabular data in the past. Instead, our core idea is to separate the data "at birth" into two independent, complementary subspaces and align the representations of one subspace to mirror the nearest-neighbor correspondence of the data in the second subspace. We demonstrate that this pretraining task indeed captures the semantic correspondence in a wide variety of tabular datasets.

SeBA achieves impressive performance in few-shot learning on various tabular datasets, confirming its effectiveness. Our findings open new avenues for further investigations into tabular representation learning and are a useful foundation for data-constrained applications.

Acknowledgment

The work of K. Jurek and M. Śmieja was supported by the National Science Centre (Poland), grant no. 2023/50/E/ST6/00169. The research of M. Przewięźlikowski was supported by the National Science Centre (Poland), grant no. 2023/49/N/ST6/03268. The work of W. Batko was funded by the program Excellence Initiative – Research University at the Jagiellonian University in Kraków. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018312

References

Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael G Mantovani, Jan N van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. *arXiv preprint arXiv:1708.03731*, 2017.

Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ZgXfXSz51n.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9640–9649, October 2021.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. *arXiv preprint arXiv:1810.02334*, 2018.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Ruoxue Liu, Linjiajie Fang, Wenjia Wang, and Bingyi Jing. D2r2: Diffusion-based representation with random distance matching for tabular few-shot learning. *Advances in Neural Information Processing Systems*, 37:36890–36913, 2024.

Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. Stunt: Few-shot tabular learning with self-generated tasks from unlabeled tables. In *11th International Conference on Learning Representations, ICLR 2023*. International Conference on Learning Representations, ICLR, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Marcin Przewięźlikowski, Mateusz Pyla, Bartosz Zieliński, Bartłomiej Twardowski, Jacek Tabor, and Marek Śmieja. Augmentation-aware self-supervised learning with conditioned projector. *Knowledge-Based Systems*, 305:112572, December 2024. ISSN 0950-7051. doi: 10.1016/j.knosys.2024.112572. URL http://dx.doi.org/10.1016/j.knosys.2024.112572.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning, 2019.

Talip Ucar, Ehsan Hajiramezanali, and Lindsay Edwards. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:18853–18865, 2021.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR, 2020.

Han-Jia Ye, Lu Han, and De-Chuan Zhan. Revisiting unsupervised meta-learning via the characteristics of few-shot tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3721–3737, 2022.

Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33, 2020.

A Experiment details

A.1 Experimental setup

We follow the benchmark proposed by [Nam et al., 2023] and then developed by [Liu et al., 2024] verifying the performance of the models in a few-shot learning scenario.

Datasets preparation. In addition to the 8 datasets used in previous SS-FSL benchmarks [Nam et al., 2023], we select 4 more datasets from the OpenML-CC18 benchmark [Asuncion et al., 2007, Bischl et al., 2017], see Appendix A.2 for details.

All datasets are randomly divided into train and test sets in a ratio of 5:1. The training data are treated unlabeled and are used for model pretraining. In addition, 10% of the training data is used for validation. Once the model is pretrained, it is fine-tuned on the support set and evaluated on the query set. The support and query sets are randomly sampled from the test set. In the N-shot K-way setting, the support set contains N examples of each of K classes. We consider 1-, 5-, and 10-shot settings.

Setup of SeBA. We pretrain the encoder and projector of SeBA for 10 000 epochs, using the early stopping. We stop training when the value of the objective function, measured on the validation set, stops decreasing for 100 epochs.

Following [Nam et al., 2023], the encoder is a 2-layer MLP with a hidden dimension of 1024, and the projector is also a 2-layer network with the same hidden dimension and an embedding dimension of 256, a common choice in contrastive learning [Chen et al., 2021]. In the fine-tuning stage, we freeze the encoder and train a classification head at the top of the encoder representation using the support set. For 5- and 10-shot, we use linear probing, while for 1-shot setting, we assign query samples to the closest class prototypes based on the support set.

Experiments on 1- and 5-shot classification are repeated with 100 different random seeds, while in the case of 10-shot learning, we use 20 seeds. A higher number of seeds reduces the randomness related to model initialization and dataset splits. For each train-test split, we sample the support and query sets 100 times and average the accuracy metrics over all splits and all selections of the support/query sets.

A.2 Datasets

The details of the datasets are presented in Table 1.

Table 1: Overview of the datasets used in the experiments, including the number of instances, proportion of numerical and categorical features, and the number of classes.

Dataset code	Dataset	# Instances	# Features (num., cate.)	# Classes
CMC	cmc	1473	9 (2,7)	3
DIA	diabetes	768	8 (8,0)	2
DNA	dna	3186	180 (0,180)	3
INC	income	48842	14 (6,8)	2
KAR	karhunen	2000	64 (64,0)	10
OPT	optdigits	5620	64 (64,0)	10
PIX	pixel	2000	240 (0,240)	2
SEM	semeion	1593	256 (256,0)	10
GES	GesturePhaseSegmentationProcessed	9873	32 (32,0)	5
MAR	bank-marketing	45211	16 (5,11)	2
PHO	phoneme	5404	5 (5,0)	2
SAT	satimage	6430	36 (36,0)	6
TEX	texture	5500	40 (40,0)	11

A.3 Baselines

Along with SeBA, we report the performance of nine methods taken from [Liu et al., 2024], which represent three types of baselines:

1. **Supervised.** CatBoost [Prokhorenkova et al., 2018] is considered a shallow SOTA approach to tabular data; k-NN [Peterson, 2009] works well for the few-shot case; TabPFN [Hollmann et al., 2023] represents a transformer-based zero-shot technique, which can be applied to small datasets.
2. **Self-supervised.** VIME [Yoon et al., 2020], SubTab [Ucar et al., 2021] and SCARF [Bahri et al., 2021] represents typical SSL approaches for tabular data. The representations acquired from those models are used to conduct Center Prototype Classification.
3. **Few-shot meta-learning.** Although UMTRA [Sun et al., 2019], SES [Ye et al., 2022] and CACTUs [Hsu et al., 2018] are designed for image data, their structures were modified for tabular data modality.

STUNT [Nam et al., 2023] and D2R2 [Liu et al., 2024] were evaluated using authors' repositories with hyperparameters selection procedures implemented there. For D2R2, we run the variant denoted by D2R2-c, which uses mean support embeddings as the classifier. The default D2R2 uses an instance-wise iterative prototype scheme, additionally using query data for class prototype estimation. This is not consistent with the inductive setting, where queries are unseen during classifier training.

A.4 Hyperparameters

We report the values of the hyperparameters used by SeBA in Table 2.

A.5 Implementation details

We implement SeBA in PyTorch Paszke et al. [2019]. We include the codebase as supplementary material and will publish it along with the paper. All of the experiments described in the paper were run on a single NVidia-V100 GPU.

B Additional experimental results

B.1 Detailed Few-shot learning performance results

We present the results of the evaluation in 1-, 5-, and 10-shot classification in Tables 3 to 5, respectively. The efficacy of SeBA increases consistently with the number of support examples, as opposed to

Table 2: SeBA hyperparameters

Hyperparameter	Value
Pretraining	
Epochs	10.000
Learning rate	0.001
Optimizer	Adam [Kingma and Ba, 2014]
Batch size	1024
Early stopping patience	100
Encoder depth	2
Encoder hidden size	1024
Encoder output size	256
Projector depth	2
Projector hidden size	1024
Projector output size	256
Few-shot classification	
Epochs	10.000
Learning rate	0.001
Optimizer	Adam [Kingma and Ba, 2014]

Table 3: Evaluation in terms of 1-shot classification accuracy.

Method	CMC	DIA	DNA	INC	KAR	OPT	PIX	SEM	GES	MAR	SAT	TEX
CatBoost	36.03	56.74	39.15	57.55	53.24	58.30	54.74	43.21	–	–	–	–
kNN	35.39	58.50	42.20	51.45	54.61	65.60	60.79	44.35	–	–	–	–
TabPFN	35.37	53.35	–	–	46.02	55.74	–	–	–	–	–	–
SubTab	36.23	58.22	46.98	62.45	50.22	62.01	60.34	39.99	–	–	–	–
VIME	35.90	58.99	51.23	61.82	59.81	69.26	63.28	46.99	–	–	–	–
Scarf	35.39	55.64	57.86	57.94	60.96	63.31	63.93	–	–	–	–	–
UMTRA	35.46	57.64	25.13	57.23	49.05	49.87	34.26	26.33	–	–	–	–
SES	34.59	59.97	39.56	56.39	49.19	56.30	49.19	33.73	–	–	–	–
CACTUs	36.10	58.92	65.93	64.02	65.59	71.98	67.61	48.96	–	–	–	–
STUNT	37.10	61.08	66.20	63.52	71.20	76.94	79.05	55.91	27.04	53.88	63.12	58.69
D2R2-c	**40.81**	60.10	61.29	**72.85**	61.45	77.41	61.45	34.26	26.58	51.70	60.92	61.78
SeBA (our)	36.76	**61.14**	**66.79**	62.89	**76.40**	**78.94**	**83.06**	**61.11**	**27.07**	**58.43**	**65.70**	**70.94**

approaches like D2R2-c, which exhibit significant variance in quality on datasets like SEM. SeBA achieves the best accuracy in 29 out of 36 instances and the second-best in 3 out of the remaining 7, which confirms its practicality and applicability to a wide range of datasets. We summarize the average performance of each method in Figure 3, from which it is evident that SeBA is the generally best-performing approach.

B.2 Detailed ablation study results

In this section, we ablate the design choices of SeBA: data preprocessing, separation ratio, and the choice of the multi-shot classifier. The model variants are evaluated in terms of 5-shot classification accuracy with 5 random seeds. We detail the model variants and report the results in Tables 6 to 8, and summarize them in Figure 4.

Data preprocessing (Figure 4a / Table 6). We ablate the usefulness of data normalization and two variants of missing data imputation: zero filling and sampling column values from marginal distribution. In most cases, the combination of data normalization and zero imputation yields representations of the highest quality.

Table 4: Evaluation in terms of 5-shot classification accuracy.

Method	CMC	DIA	DNA	INC	KAR	OPT	PIX	SEM	GES	MAR	SAT	TEX
CatBoost	39.89	64.51	60.20	67.99	77.94	83.07	83.38	68.69	–	–	–	–
kNN	37.65	65.61	61.16	62.19	80.08	84.16	84.75	68.33	–	–	–	–
TabPFN	38.31	64.06	–	–	76.59	81.68	–	–	–	–	–	–
SubTab	39.81	68.26	62.49	72.14	70.88	83.27	80.41	59.87	–	–	–	–
VIME	39.83	67.64	71.29	72.19	19.42	83.21	85.24	68.45	–	–	–	–
Scarf	37.75	68.66	62.75	66.09	69.96	85.67	81.32	–	–	–	–	–
UMTRA	38.05	64.41	25.08	65.78	67.28	73.29	51.32	35.90	–	–	–	–
SES	39.04	66.61	52.25	68.27	74.80	78.46	74.80	52.74	–	–	–	–
CACTUs	38.81	66.79	81.52	72.03	82.20	85.92	85.25	65.00	–	–	–	–
STUNT	40.40	69.88	79.18	72.69	85.45	88.42	89.08	71.54	**32.19**	58.62	74.25	68.57
D2R2-c	**43.39**	68.69	**81.39**	**73.34**	79.49	87.12	82.22	60.16	30.26	56.24	70.66	71.82
SeBA (our)	42.85	**69.54**	79.86	71.28	**87.59**	**90.11**	**91.88**	**79.41**	32.07	**65.22**	**78.66**	**87.51**

Table 5: Evaluation in terms of 10-shot classification accuracy.

Method	CMC	DIA	DNA	INC	KAR	OPT	PIX	SEM	GES	MAR	SAT	TEX
STUNT	42.01	72.82	80.96	74.08	86.95	89.91	89.98	74.74	34.30	61.08	75.58	71.10
D2R2-c	37.86	72.02	81.72	**75.34**	84.81	89.27	73.70	36.30	31.22	59.80	71.96	73.43
SeBA (our)	**46.30**	**73.61**	**83.59**	72.68	**90.88**	**92.62**	**93.88**	**84.11**	**34.60**	**69.96**	**81.17**	**90.18**

Separation ratio and model ensembling (Figure 4b / Table 7). We ablate the choice of target / feature separation ratio and compare it with an ensemble of encoders trained with all ratios. Although certain ratios yield the best results for several datasets, we find that ensembles of encoders perform most reliably.

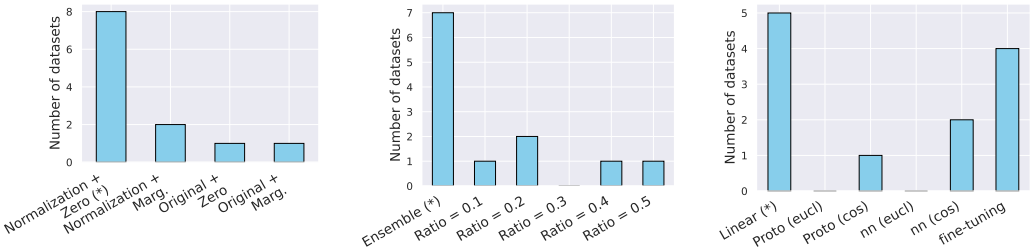
Learning the few-shot classifier (Figure 4c / Table 8). We compare several choices of learning the classifier on top of the pretrained representation from the support data in the multi-shot setting. Our analysis shows that linear probing is the simplest and most effective approach.

Table 6: Ablation of normalizing the numerical columns in tabular data (**Norm.**) and of the type of data imputation in separated columns (**Imput.**), where we compare zero-imputation (Zero), and sampling from the column's marginal distribution (Marg.).

Norm.	**Imput.**	CMC	DIA	DNA	INC	KAR	OPT	PIX	SEM	GES	MAR	SAT	TEX
True	Zero (*)	**42.85**	69.54	**79.86**	**71.28**	87.59	**90.11**	**91.88**	**79.41**	32.07	**65.22**	78.66	**87.51**
	Marg.	41.71	**69.78**	68.69	67.89	86.36	89.18	89.67	77.32	**32.70**	53.19	78.64	84.83
False	Zero	40.50	53.49	70.82	47.01	**91.00**	87.91	90.54	77.28	31.06	60.16	78.67	68.43
	Marg.	37.95	54.97	69.71	46.44	89.09	89.41	89.29	77.85	30.87	59.40	**78.69**	79.04

Table 7: Ablation of the separation ratios between the target and feature data views, compared with the ensemble of encoders trained with different ratios.

Mode	CMC	DIA	DNA	INC	KAR	OPT	PIX	SEM	GES	MAR	SAT	TEX
Ensemble (*)	**42.85**	69.54	**79.86**	**71.28**	87.59	90.11	**91.88**	**79.41**	32.07	**65.22**	**78.66**	87.51
Ratio = 0.1	41.18	68.20	69.07	69.73	84.21	85.84	89.37	72.02	**32.21**	64.98	78.53	85.08
Ratio = 0.2	41.56	**69.73**	73.64	68.14	86.36	88.83	91.13	76.04	31.88	62.33	**78.66**	**88.16**
Ratio = 0.3	42.71	68.19	77.01	68.40	87.53	90.29	91.24	77.33	31.80	63.58	78.21	87.98
Ratio = 0.4	41.86	68.41	73.33	70.11	87.43	**90.69**	91.79	78.69	31.40	64.28	77.82	87.08
Ratio = 0.5	41.16	68.26	72.69	70.74	**88.11**	90.51	89.83	78.94	31.35	59.66	77.87	85.37



(a) Ablation of data preprocessing techniques.

(b) Ablation of separation ratio and model ensembling.

(c) Ablation of few-shot classification approaches.

Figure 4: Ablation of the design aspects of SeBA ((*) denotes the default setting of SeBA). In the barplots, we report the number of datasets in which a given variant of SeBA performs best.

Table 8: Ablation of different ways of forming the many-shot classifier. We compare linear probing (Linear), using support data to form prototypes and assigning queries based on Euclidean od cosine distance (Proto eucl/cos), matching individual support representations as nearest neighbors based on Euclidean od cosine distance (nn eucl /cos), and fine-tuning the whole encoder along with the classifier (fine-tuning).

Mode	CMC	DIA	DNA	INC	KAR	OPT	PIX	SEM	GES	MAR	SAT	TEX
Linear **(*)**	**42.85**	69.54	**79.86**	71.28	87.59	90.11	91.88	79.41	**32.07**	65.22	**78.66**	**87.51**
Proto (eucl)	39.78	67.27	75.85	70.45	86.06	86.76	92.33	75.16	28.70	66.68	73.46	76.66
Proto (cos)	39.35	67.66	78.49	70.30	87.36	89.61	**92.71**	75.50	31.10	65.45	77.29	79.97
nn (eucl)	40.31	66.57	72.68	69.22	86.26	89.74	92.37	76.41	28.14	64.59	74.97	84.76
nn (cos)	40.83	67.34	74.53	70.14	**87.74**	**91.09**	92.55	77.74	30.41	63.57	77.78	86.28
fine-tuning	42.52	**70.31**	76.30	**71.63**	85.27	89.09	91.71	**79.48**	31.27	**66.91**	78.15	86.17

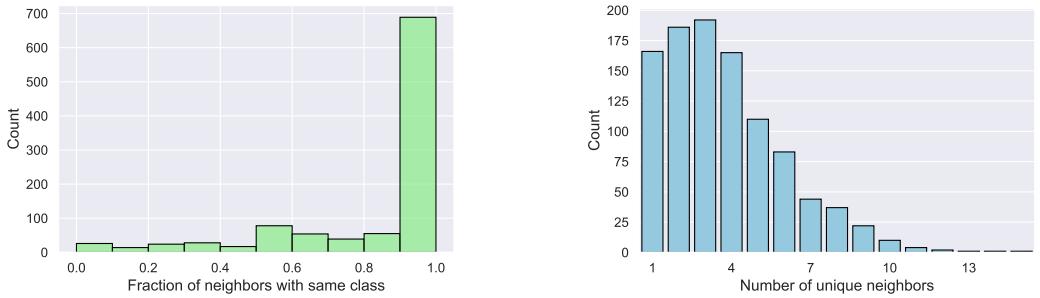
B.3 Alignment of the SeBA pretraining objective with recognition task

In this section, we evaluate the validity of the proposed Separated-at-Birth Alignment as an unsupervised pretraining objective. For this purpose, we analyze its stability and the semantic relationship of the positive pairs created by SeBA. For each dataset, we generate 100 random separations into feature and target views with a separation ratio of 0.2. Next, we identify the nearest neighbors of the samples in terms of target views (see eq. (3)).

To measure how the SeBA objective aligns with the downstream classification task, we measure the proportion of pairs in which the nearest neighbors share the same class as the original instance, see Figure 5a for the SEM dataset. A detailed inspection of the remaining datasets shows that the vast majority of nearest-neighbor pairs share the same class, which indicates that the pretraining objective learns features useful for a downstream task, see Figure 6.

We also verify the stability of SeBA. To this end, we count the number of unique samples that are matched to a given instance as nearest neighbors. It is evident from Figure 5b that the average number of unique neighbors for the SEM dataset reaches up to 5% of the entire data, showing low noise in the pretraining objective, see Figure 7.

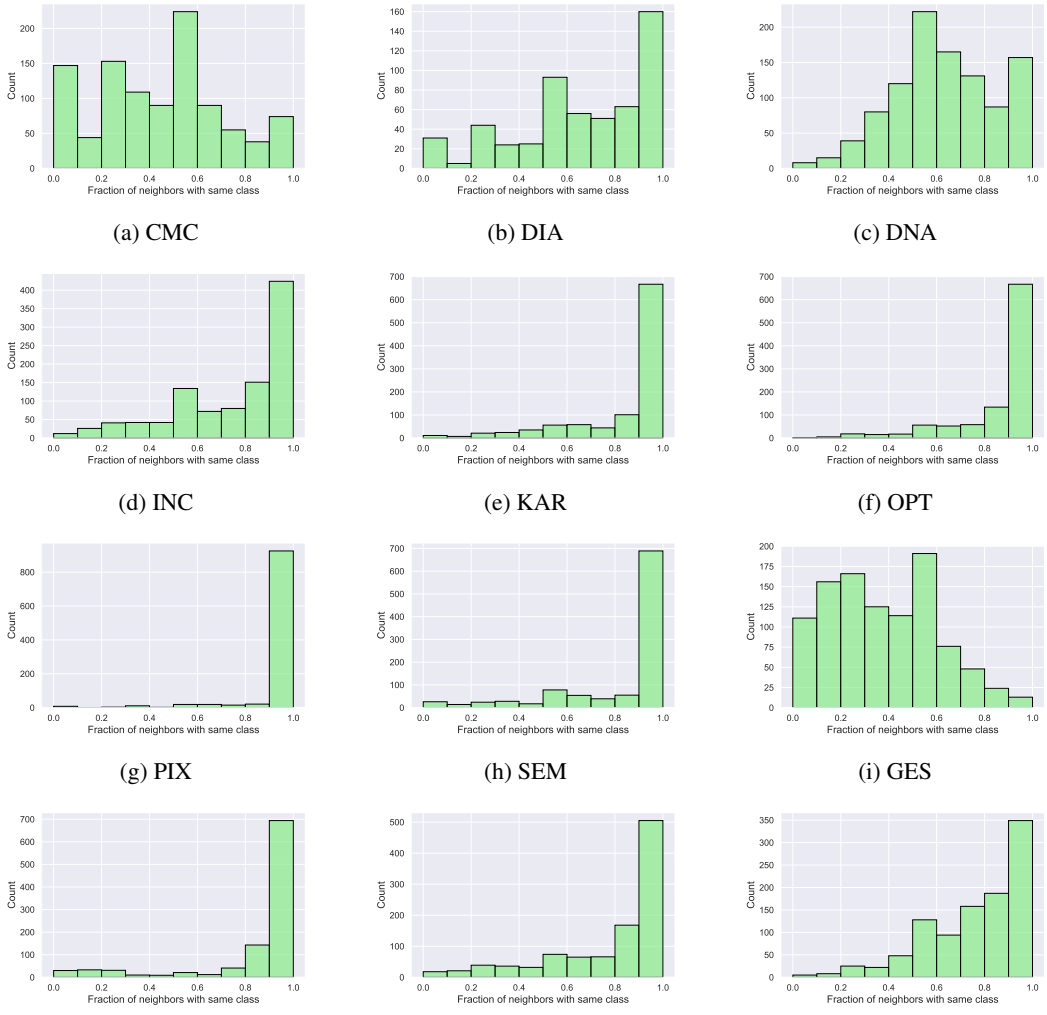
Both metrics follow similar-shaped distributions for the majority of datasets, indicating high stability of the SeBA pretraining objective. The exceptions include the CMC and GES datasets, in which the performance of SeBA is relatively lower, especially in 1- and 5-shot classification tasks. This indicates that while SeBA is generally a good pretext task for learning tabular representation, there is still room for improvement in future work.



(a) Same-class neighbor ratios.

(b) Unique neighbor number distribution.

Figure 5: Analysis of neighbor stability under masked perturbations for Semeion dataset: (a) high fraction of neighbors sharing the same class label as the original instance confirms high consistency between pretext and downstream tasks, (b) low number of unique neighbors for each sample indicates high stability of SeBA.



(a) CMC

(b) DIA

(c) DNA

(d) INC

(e) KAR

(f) OPT

(g) PIX

(h) SEM

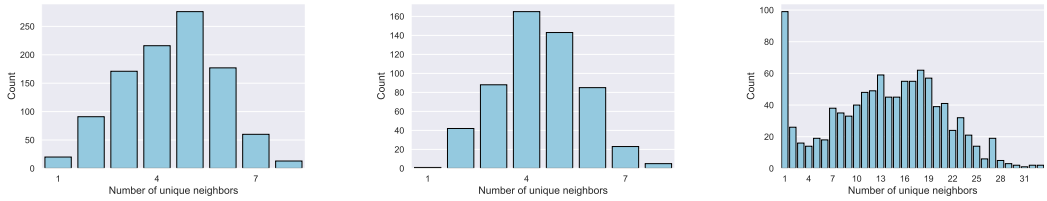
(i) GES

(j) MAR

(k) SAT

(l) TEX

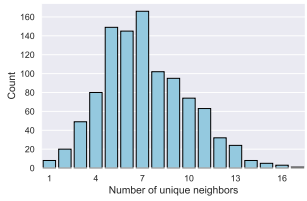
Figure 6: The fraction of neighbors sharing the same class label as the original instance. High values indicate high alignment between pretext and downstream tasks.



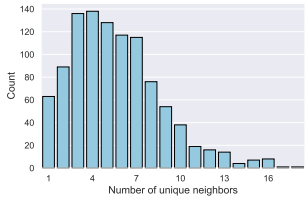
(a) CMC

(b) DIA

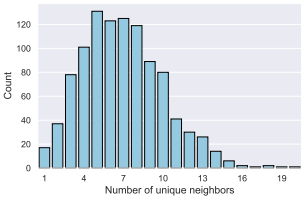
(c) DNA



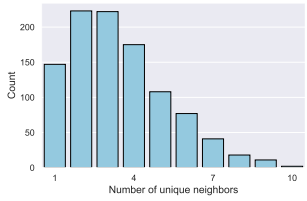
(d) INC



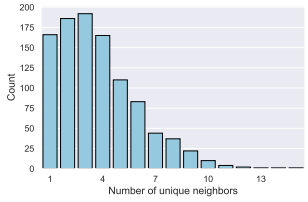
(e) KAR



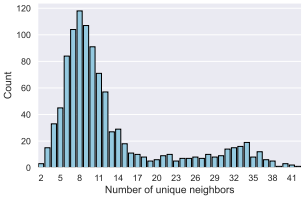
(f) OPT



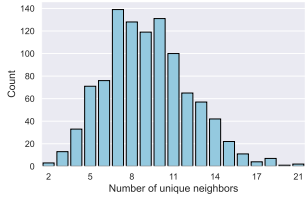
(g) PIX



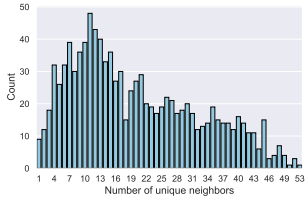
(h) SEM



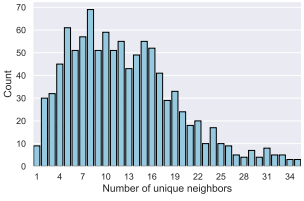
(i) GES



(j) MAR



(k) SAT



(l) TEX

Figure 7: A distribution of the number of unique neighbors matched to the original instance. Low values indicate high stability of SeBA.