ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

# AdaCap: An Adaptive Contrastive Approach for Small-Data Neural Networks

Bruno Belucci[1], Karim Lounici[2] and Katia Meziani[1] *

1- Université Paris Dauphine-PSL - CEREMADE
Pl. du Maréchal de Lattre de Tassigny, 75016 Paris - France

2- Ecole Polytechnique - CMAP
Route de Saclay, 91128 PALAISEAU - France

**Abstract**. Neural networks struggle on small tabular datasets, where tree-based models remain dominant. We introduce **Ada**ptive **C**ontrastive **Ap**proach (AdaCap), a training scheme that combines a permutation-based contrastive loss with a Tikhonov-based closed-form output mapping. Across 85 real-world regression datasets and multiple architectures, AdaCap yields consistent and statistically significant improvements in the small-sample regime, particularly for residual models. A meta-predictor trained on dataset characteristics (size, skewness, noise) accurately anticipates when AdaCap is beneficial. These results show that AdaCap acts as a targeted regularization mechanism, strengthening neural networks precisely where they are most fragile. All results and code are publicly available at https://github.com/BrunoBelucci/adacap.

## 1 Introduction

Deep neural networks (NNs) achieve state-of-the-art results in vision, language, and large-scale tabular modeling, yet their performance remains unreliable in the **small-sample regime**, characteristic of many real-world tabular tasks. In contrast, gradient boosting decision trees (GBDTs) remain remarkably robust under limited data, and consistently outperform NNs on curated tabular benchmarks [1, 2] and the references cited therein. Understanding why NNs collapse in this setting, and how to mitigate this collapse, remains an open problem.

Several approaches have attempted to improve NN robustness on tabular data: architectural modifications inspired by tree ensembles [3], hyperparameter tuning [4], regularization strategies [5], Bayesian inference [6], and data augmentation [7]. In parallel, contrastive objectives [8] have emerged as powerful tools for learning structure from limited supervision. Yet, none of these techniques directly address a key challenge of small-data tabular regression: **NNs must learn a meaningful representation while simultaneously avoiding overfitting**, a trade-off for which standard training pipelines offer little guidance.

We introduce **AdaCap**, a simple training scheme designed to strengthen neural networks for the regression task *precisely where they are most fragile*: in low-data, high-variance regimes. AdaCap combines two components. ($i$) A **Tikhonov layer** that replaces the standard output layer with a closed-form

*This project was provided with HPC resources by GENCI at IDRIS/TGCC thanks to the grant 2024-A0181015860.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Tikhonov-regularized solution, providing a stable and data-efficient mapping between the learned representation and the target. $(ii)$ A **permutation-based contrastive loss** that increases the fit to the true labels while decreasing the fit obtained when the labels are randomly shuffled, thereby enforcing that the model captures genuine input–target dependencies rather than patterns independent of the inputs.

Across **85 real-world regression datasets** and diverse architectures , AdaCap produces consistent improvements in small-sample settings. The gains are particularly strong and frequent for **residual architectures**. Finally, we show that the effectiveness of AdaCap is **predictable**: an XGBoost meta-classifier trained on dataset characteristics (e.g., number of instances, target skewness, noise levels) reaches $\approx 70\%$ accuracy in predicting whether AdaCap will improve a given model. This indicates that AdaCap does not act as a universal booster, but rather as a **targeted and learnable regularizer** whose impact depends on identifiable structural properties of the dataset.

## 2 AdaCap Training Scheme

AdaCap combines two complementary components: $(i)$ a **Tikhonov-based closed-form** output mapping that replaces the standard learned final layer, and $(ii)$ a **permutation-based contrastive loss** that contrasts the fit obtained with true versus shuffled labels. Together, these components provide a strong regularization signal in low-data regimes while preserving compatibility with standard neural architectures.

**Tikhonov output mapping.** Let $H \in \mathbb{R}^{n\times d}$ be the final hidden representation produced by the network and let $Y \in \mathbb{R}^n$ be the target vector. In a standard network, the output layer computes $\widehat{Y} = HW$ for trainable weights $W \in \mathbb{R}^d$. AdaCap replaces these weights by: $W(\lambda) = (H^\top H + \lambda I_d)^{-1} H^\top Y$, $\widehat{Y} = HW(\lambda)$, where $\lambda \in \mathbb{R}^+$ is trainable. Using the SVD of $H^\top H$, this computation requires only a single matrix factorization and thus remains computationally efficient.

**Permutation-based contrastive loss.** To add a contrastive signal, we generate $P$ independent permutations $\pi_1(Y), \ldots, \pi_P(Y)$ by randomly shuffling the target vector. Each permutation preserves the marginal distribution of $Y$ while destroying any input-output dependency. For each permuted vector, we reuse the same Tikhonov mapping: $W^{(p)}(\lambda) = (H^\top H + \lambda I_d)^{-1} H^\top \pi_p(Y)$, $\widehat{Y}^{(p)} = HW^{(p)}(\lambda)$. The Tikhonov solutions all reuse the same SVD of $H^\top H$, avoiding repeated inversions. The AdaCap loss contrasts the fit under true labels with the average fit under permuted labels: $\mathcal{L}_{\text{AdaCap}} = \|Y - \widehat{Y}\| - \frac{1}{P}\sum_{p=1}^{P} \|\pi_p(Y) - \widehat{Y}^{(p)}\|$.

This loss is contrastive: it increases when the model fits the true labels better than their shuffled counterparts. Because both fits are computed using the same Tikhonov estimator, the parameter $\lambda$ implicitly scales the contrast by controlling how strongly the estimator responds to structure in the representation $H$.

**Initialization of the Tikhonov parameter.** The performance of AdaCap is sensitive to the initial value of $\lambda$. We initialize it by performing a single forward pass over the training set (without updating the network) and evalu-

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

ating the loss for values of $\lambda$ on a logarithmic grid ranging from $10^{-3}$ to $10^3$. We pick the value that yields the largest variation in the loss. This procedure provides a stable starting point and consistently improves subsequent training.

## 3 Experimental Results

**Datasets.** We evaluate AdaCap on 85 real-world regression datasets from the OpenML-CTR23 benchmark[9, 2]. The datasets span a broad range of sizes and dimensionalities, with heterogeneous proportions of numerical and categorical features. A summary of dataset characteristics is provided in Table 1. For each dataset, we follow the official train-test splits and use 10-fold cross-validation to report RMSE, MAE, MAPE, and $R^2$.

Table 1: Dataset characteristics summary (85 datasets).

| Characteristic | Min | Q1 | Median | Q3 | Max | Mean |
|---|---|---|---|---|---|---|
| Instances ($N$) | 103 | 768 | 8,192 | 21,263 | 5,465,575 | 106,355 |
| Features ($k$) | 3 | 8 | 11 | 21 | 525 | 41 |
| Categorical ($C$) | 0 | 0 | 0 | 3 | 359 | 7 |

**Effect of the number of permutations.** We first study how the number of permutations $P$ influences performance. A simple MLP trained with AdaCap was evaluated with $P \in \{1, 2, 5, 10, 20\}$, and for each configuration, we counted the number of wins across all datasets, folds, and metrics. As reported in Table 2, the setting $P = 10$ achieves the highest overall number of wins. We therefore fix $P = 10$ for all subsequent experiments.

Table 2: Win counts and percentages across different permutation levels.

| $P$ | RMSE | MAE | MAPE | $R^2$ | Total Wins | % Wins |
|---|---|---|---|---|---|---|
| 1 | 147 | 141 | 164 | 147 | 599 | 17.41 |
| 2 | 138 | 126 | 134 | 138 | 536 | 15.58 |
| 5 | 151 | 151 | 170 | 151 | 623 | 18.11 |
| 10 | **228** | 215 | **197** | **228** | **868** | **25.23** |
| 20 | 196 | **227** | 195 | 196 | 814 | 23.66 |

**Models.** We consider MLP and ResNet architectures [10, 4] with four layers and a 256-dimensional hidden size, using 256-dimensional embeddings for categorical variables and ReLU activations. Variants include: (i) Deeper (8 layers), (ii) GLU (Gated Linear Units [11]), and (iii) GReLUOneCycleLR (Generalized ReLU with One-Cycle learning rate scheduling [12]). We also evaluate a vanilla Transformer [13]. All architectures are trained both with and without AdaCap .

**Impact Across Architectures and Dataset Sizes.** We next evaluate the effect of AdaCap across architectures and dataset sizes. Figure 1 summarizes the Wilcoxon signed-rank outcomes comparing each model with and without AdaCap across RMSE, MAE, MAPE, and $R^2$.

Across dataset sizes, the effect of AdaCap is strong but not uniform. In the **small-data regime** ($N < 500$), AdaCap consistently improves performance for several architectures: MLP GReLUOneCycleLR, ResNet, ResNet Deeper, and ResNet GReLUOneCycleLR are all significantly better across every metric. The plain MLP and ResNet GLU also benefit in this regime, although their gains

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

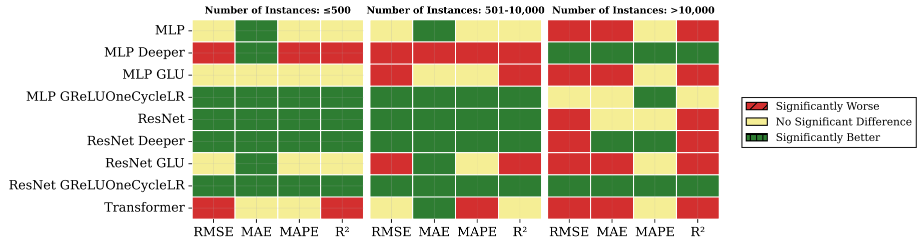


Fig. 1: Wilcoxon outcomes across dataset-size regimes.

are mostly limited to MAE. This confirms that AdaCap is particularly effective when data are scarce and the representation needs stronger regularization.

In the **intermediate regime** ($500 < N \leq 10{,}000$), the effect becomes more heterogeneous. Residual models still improve in many cases, but some variants (e.g., ResNet GLU) begin to show mixed outcomes across metrics. Certain MLP variants shift from improvements to degradations, and Transformers exhibit a metric-dependent behavior, ranging from neutral to mildly positive or negative.

For **large datasets** ($N > 10{,}000$), AdaCap is no longer uniformly beneficial. MLP Deeper becomes consistently better across all metrics, while the plain MLP is often neutral or worse. Some ResNet variants lose their improvements, except ResNet GReLUOneCycleLR, which remains strongly positive. Transformers are mostly unaffected or degraded in this regime.

Overall, these results indicate that AdaCap is not a universal booster: **its strongest gains appear in small-data settings and in specific architecture–size combinations**. This structured behavior motivates the following analysis on predicting when AdaCap is most effective.

## 4 Predicting When AdaCap Helps

A key question is to predict *a priori* when AdaCap will improve a given architecture on a given dataset. To answer this question, for each (dataset, architecture) pair, we build a feature vector of dataset characteristics: number of samples and features; categorical-variables counts and cardinalities; summary statistics of numerical variables (skewness, kurtosis, correlations); numerical-features outlier ratios; intrinsic dimensionality; PCA variance ratios; target skewness and kurtosis; and a noise estimate from a random forest regressor.

To assess predictability using these data, we train an XGBoost classifier on all 85 datasets for each architecture. Each paired run of models trained with and without AdaCap is treated as a single sample, with the associated label indicating whether AdaCap yields an improvement in performance. The classifiers achieve a mean accuracy of 70% (std. 11%) on held-out runs with a 20% test split, showing that improvements from AdaCap are indeed predictable from dataset structure. Restricting the classifier to only the top three features, according to the sum of their rank using the mean absolute SHAP [14] values (Table 3), still

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Table 3: Top 10 features ranked by SHAP importance.

| Feature | Sum of Ranks |
|---|---|
| Number of Instances | 224 |
| Target Skewness | 538 |
| Noise Estimation (RF) | 588 |
| Min Skewness of Numerical Features | 613 |
| Max Absolute Correlation of Numerical Features | 630 |
| Target Kurtosis | 634 |
| Min Absolute Correlation of Numerical Features | 637 |
| Min Kurtosis of Numerical Features | 644 |
| Outlier Ratio of Numerical Features | 668 |
| Mean Skewness of Numerical Features | 674 |

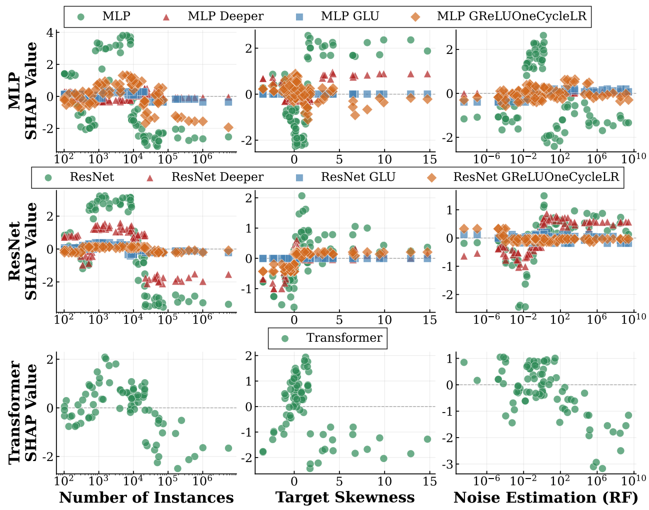


Fig. 2: SHAP values of the top dataset characteristics for several architectures.

yields an accuracy of 67% (std. 12%), indicating that a small subset of dataset characteristics is sufficient for reliable predictions.

Figure 2 presents SHAP scatter plots for several architectures. The dominant predictor across all models is **dataset size**: smaller datasets make improvements from AdaCap substantially more likely. Transformers benefit more from datasets with symmetric target distributions, whereas MLPs and ResNets benefit more strongly from positively skewed targets. Noise level also interacts with architecture: AdaCap tends to help ResNets in noisy datasets, MLPs in low-noise regimes, and Transformers in low to medium noise settings.

## 5 Conclusion

We presented AdaCap, a simple combination of a Tikhonov closed-form output and a permutation-based contrastive loss that strengthens neural networks

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

in small-data tabular regression. Experiments on 85 datasets show substantial gains for residual architectures in low-sample regimes, with more heterogeneous effects for larger datasets. A meta-classifier accurately predicts when AdaCap helps, revealing dataset size, target skewness, and noise level as dominant factors. AdaCap thus provides a targeted and interpretable improvement where neural networks are most brittle. An avenue for future research involves extending AdaCap to supervised classification settings, as well as conducting a rigorous theoretical analysis of the obtained empirical results.

## References

[1] Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Benjamin Feuer, Chinmay Hegde, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When Do Neural Nets Outperform Boosted Trees on Tabular Data?, October 2023.

[2] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *NeurIPS 2022 Datasets and Benchmarks Track*, New Orleans, United States, November 2022.

[3] Witold Wydmański, Oleksii Bulenok, and Marek Śmieja. HyperTab: Hypernetwork Approach for Deep Learning on Small Tabular Datasets. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–9, October 2023.

[4] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned Simple Nets Excel on Tabular Datasets, November 2021.

[5] Ira Shavitt and Eran Segal. Regularization learning networks: Deep learning for tabular datasets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 1386–1396, Red Hook, NY, USA, December 2018. Curran Associates Inc.

[6] 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*, September 2022.

[7] Lorenzo Brigato and Luca Iocchi. A Close Look at Deep Learning with Small Data. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2490–2497, January 2021.

[8] Haigen Hu, Xiaoyuan Wang, Yan Zhang, Qi Chen, and Qiu Guan. A comprehensive survey on contrastive learning. *Neurocomputing*, 610:128645, December 2024.

[9] Sebastian Felix Fischer, Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – A curated tabular regression benchmarking suite. In *AutoML Conference 2023 (Workshop)*, August 2023.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385, 2015.

[11] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 933–941, Sydney, NSW, Australia, August 2017. JMLR.org.

[12] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates, May 2018.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017.

[14] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.