# AGATa: Attention-Guided Augmentation for Tabular Data in Contrastive Learning

**Moonjung Eo, Kyungeun Lee, Min-Kook Suh, Hye-Seung Cho, Ye Seul Sim, Woohyung Lim**
LG AI Research
Seoul, Republic of Korea
{moonj,kyungeun.lee,minkook.suh,hs.cho,ysl.sim,w.lim}@lgresearch.ai

## Abstract

Contrastive learning has demonstrated significant potential across various domains, including recent applications to tabular data. However, adapting this approach to tabular structures presents distinct challenges, particularly in developing effective augmentation techniques. While existing methods have shown promise, there remains room for improvement in preserving critical feature relationships during the augmentation process. In this paper, we explore an alternative approach that utilizes attention scores to guide augmentation, aiming to introduce meaningful variations while maintaining important feature interactions. This method builds upon existing work in the field, offering a complementary perspective on tabular data augmentation for contrastive learning. Our approach explores two main aspects: 1) Attention-guided Feature Selection, which focuses augmentations on features with lower attention scores, and 2) Dynamic Augmentation Strategy, which alternates between different augmentation techniques during training. This combination aims to maintain key data characteristics while introducing diverse variations. Experimental results suggest that our method performs competitively with existing augmentation techniques in preserving tabular data structure and enhancing downstream task performance.

## 1 Introduction

In recent years, contrastive learning has emerged as a powerful self-supervised learning framework, showing success across various domains [1, 2]. Despite its potential, the application of contrastive learning to tabular data remains relatively underexplored and presents distinct challenges [3, 4]. A crucial component of contrastive learning is the creation of positive samples through data augmentation, which aims to introduce variation while preserving the essential properties of the original data [5]. However, popular augmentation methods like random masking, feature shuffling, and CutMix [6, 7, 8, 9], though effective for other types of data, can pose unique difficulties when applied to tabular datasets due to their lack of inherent spatial or positional structure.

In image data, for example, the spatial relationships between pixels ensure that random masking, shuffling, or partial removal often preserves much of the image's meaning due to the surrounding context [10]. Similarly, in textual data, the positional relationships between words or tokens allow for the retention of overall meaning even when some elements are modified or rearranged [11]. Tabular data, however, is fundamentally different. It lacks the spatial or positional structure found in images and text. Each column in a table represents a distinct attribute with little or no inherent connection to neighboring columns. This absence of context means that applying random augmentations to tabular data can distort important inter-feature relationships, leading to samples that are no longer representative of the original data [4, 3]. The challenge emphasizes the importance of preserving core information during augmentation, as noted in various studies [12, 13]. Despite this, random

Table Representation Learning Workshop at NeurIPS 2024.

augmentation methods are still commonly applied to tabular data, although their effectiveness remains uncertain [14]. Research on augmentation techniques tailored specifically to tabular data has been limited, underscoring the need for more carefully crafted strategies that account for its unique structure and characteristics.

Our work addresses the gap in data augmentation techniques for tabular data by proposing a novel method that leverages self-attention scores from a transformer model. We call our approach **AGATa** (**A**ttention-**G**uided **A**ugmentation for **Ta**bular data in contrastive learning). We base our method on the hypothesis that features with high self-attention scores are crucial for capturing underlying data patterns. Specifically, we use a transformer as the backbone model to extract self-attention scores from each feature [15, 3]. We then calculate the average inter-attention scores to quantify feature importance, retaining a certain percentage of top-scoring features and performing augmentations on the remaining ones. The specific augmentation—whether masking, shuffling, or CutMix—is randomly chosen each epoch, adding variability while ensuring the essential structure of the data remains intact.

A key advantage of our method is its ability to perform sample-specific augmentation. By using attention scores unique to each data sample, we can tailor the augmentation to reflect the significance of individual features, ensuring that augmentations are applied to less impactful features while preserving important ones.

In the context of contrastive learning, our approach generates positive samples that are distinct enough to be informative but still similar to the original data, preserving meaningful relationships. This balance helps the model learn more robust and generalizable representations [1]. Our main contributions can be summarized as follows:

- Development of a selective, sample-wise augmentation method for tabular data based on transformer self-attention scores.
- Introduction of a dynamic augmentation strategy that varies techniques across epochs while preserving high-impact features.
- Empirical demonstration of our method's effectiveness across various datasets and tasks, showcasing its adaptability and performance improvements over existing approaches.

The detailed implementation of our method, including the code, is provided in the Appendix for reproducibility. Our approach offers a new direction for data augmentation in tabular domains, potentially benefiting a wide range of machine-learning applications.

## 2 Related Work

Self-supervised learning (SSL) has been applied to tabular data primarily through two prominent techniques: auto-encoding and contrastive learning. Auto-encoding approaches [6, 16, 17, 8] focus on reconstructing samples from corrupted versions, aiming to learn robust features adaptable to the heterogeneity of tabular data. These methods employ objectives such as corruption detection or prediction of binning information as pre-text tasks. On the other hand, contrastive learning [14, 9, 6, 15, 18, 3] maximizes the similarity between augmented views of the same sample while minimizing similarity to others, using strategies like masking or feature cropping to define positive and negative pairs. VIME [6] utilizes this augmentation to develop pre-training tasks, such as reconstructing the original features and predicting the mask vector from the masked features. In contrast, SCARF [14] employs contrastive learning by comparing the original data with the corrupted version. SubTab [9] and Transtab [15] propose a different data augmentation approach, which involves partitioning the input tabular data into several subsets. Contrary to approaches that augment views in input space, our approach augments in latent space to address the heterogeneity and lack of clear structure in tabular datasets. Contrastive Mix-up [18] takes a different approach, employing manifold mix-up to generate two distinct data views for contrastive learning purposes. Similarly, SAINT [3] utilizes cut mix and manifold mix-up to create diverse data views, which are then used for contrastive learning. In this study, we define positive pairs using top singular vectors of the representation space.

## 3 Backgrounds

### 3.1 Augmentation techniques

Data augmentation plays a critical role in improving task performance across a variety of applications, as it can provide benefits as regularization in supervised learning, consistency regularization in semi-supervised learning, and generating positive views in contrastive learning. In contrast to domains such as image, text, or time-series data, tabular data lacks clear inductive biases like spatial or temporal correlations, local dependencies, or contextual relationships. Additionally, tabular datasets often exhibit high irregularity and heterogeneous structures, making it challenging to design augmentation methods that both preserve semantic information and introduce meaningful perturbations. As a result, most augmentation methods in tabular data rely on feature-level or cell-level modifications, as outlined below: There are five augmentation techniques for tabular data, each designed to introduce perturbations while preserving the underlying structure of the data. Below are the methods and their corresponding mathematical formulations:

Let $x_i \in \mathbb{R}^d$ represent a sample from a dataset $\mathcal{D}$, where $i$ indexes the sample, and $d$ denotes the number of features. For each sample $x_i$, $x_{i,k}$ refers to the value of feature $k$ for sample $i$. A binary mask $m_{i,k} \in \{0, 1\}$ is generated, with each element independently sampled from a Bernoulli distribution, controlling whether or not the feature is modified. The batch size is $N$, and $j$ represents the index of another randomly selected sample from the batch. $\mu_k$ refers to the mean of feature $k$ across the training dataset, $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$ denotes Gaussian noise added to a feature, and $M \in \{0, 1\}^d$ is a binary mask for CutMix, where $\odot$ indicates element-wise multiplication.

Below, we outline four augmentation techniques used for tabular data:

- **Masking** [6]: Randomly mask selected feature values by replacing them with a constant, typically the mean of the feature across the dataset:
$$\tilde{x}_{i,k} = m_{i,k} \cdot x_{i,k} + (1 - m_{i,k}) \cdot \mu_k$$
- **Subset** [9]: Divide the input features into multiple subsets and select only one subset for the augmentation process:
$$\tilde{x}_i = x_i[s]$$
- **Shuffling** [8]: Shuffle selected feature values by replacing them with the corresponding values from another randomly selected sample in the batch:
$$\tilde{x}_{i,k} = m_{i,k} \cdot x_{i,k} + (1 - m_{i,k}) \cdot x_{j,k}, \quad j \in \{1, 2, \ldots, N\} \setminus \{i\}$$
- **CutMix** [7]: Mix regions from two input samples based on a binary mask $M$, generating a new sample:
$$\hat{x} = M \odot x_A + (1 - M) \odot x_B$$

Masking and subset can be categorized as similar techniques, as both involve selectively modifying portions of the input features. Therefore, we focused our experiments on three augmentation methods: Masking, Shuffling, and CutMix. In our augmentation process, one of these three techniques was randomly selected for each iteration, ensuring a diverse application of augmentations while maintaining consistency with the structure of the data.

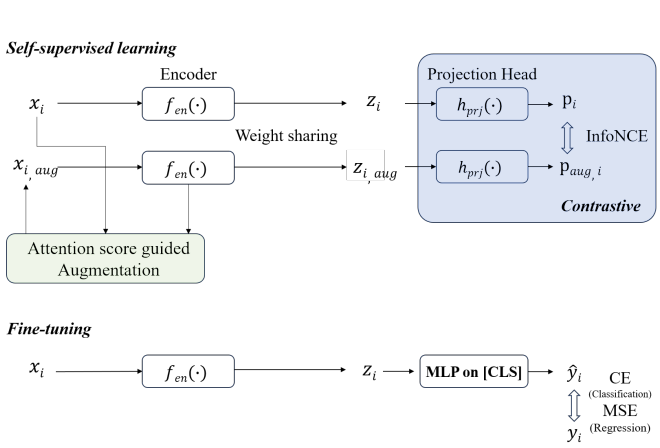
### 3.2 Transformer for tabular domain

In this work, we leverage a transformer architecture specifically designed for tabular data incorporating both categorical and numerical features. The core component of the transformer is its self-attention mechanism, which allows the model to weigh the importance of different features by computing attention scores. For each feature, the self-attention mechanism calculates a weighted sum of all other features, where the weights are learned dynamically based on the input data. The input processing begins with feature-type-specific embeddings. For categorical features, learnable embedding tables are used.

$$e_c = \text{Embed}(x_c) \tag{1}$$

where $x_c$ is the categorical input and Embed is the embedding function. For numerical features, a simple linear transformation is applied.

$$e_n = W_n x_n + b_n \tag{2}$$




(a) Self-supervised pre-training process integrated with AGATa

| Feature | F1 | F2 | F3 |
|---|---|---|---|
| $x_1$ | 0.1 | 0.9 | 0.7 |
| $x_2$ | 0.2 | 0.3 | 0.1 |

*Attention score guided Augmentation*

| Feature | F1 | F2 | F3 |
|---|---|---|---|
| Attention Score of $x_1$ | **0.8** | 0.2 | **0.7** |
| Attention Score of $x_2$ | 0.1 | **0.8** | 0.1 |

$A \sim \{Masking, Shuffling, CutMix\}$ $Aug(x) = a(x)$, where $a \sim Uniform(A)$

| Feature | F1 | F2 | F3 |
|---|---|---|---|
| $x_1$ | 0.1 | Aug. | 0.7 |
| $x_2$ | Aug. | 0.3 | Aug. |

(b) Augmentation module

Figure 1: This figure illustrates the integration of AGATa into the self-supervised pre-training process, including (a) an overview of the process and (b) a detailed depiction of the AGATa augmentation module.

where $x_n$ is the numerical input, and $W_n$ and $b_n$ are learnable parameters.

These embeddings are then concatenated to form the input to the self-attention module:

$$E = [e_c; e_n] \tag{3}$$

The self-attention mechanism operates on this combined embedding:

$$Z = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

where $Q = EW_Q$, $K = EW_K$, and $V = EW_V$ are query, key, and value matrices derived from the input embeddings, $d_k$ is the dimension of the keys, and $Z$ represents the output of the attention mechanism.

This self-attention mechanism enables the model to learn complex relationships between features, making it effective for various types of data, including tabular data. By combining categorical and numerical embeddings and applying the attention formula, the model captures feature interactions. The attention mechanism helps in understanding the relative importance of different features, while the feed-forward layers further process these attended features. In our application, attention scores derived from this mechanism guide the augmentation process, selectively manipulating features with lower attention scores. This contrasts with conventional augmentation methods, which are applied uniformly without regard to feature importance. The use of self-attention allows for a more nuanced, data-driven approach to both modeling and augmentation, focusing on the most relevant features.

# 4 Methodology

## 4.1 Overview

This work introduces an attention-guided augmentation strategy for tabular data in the context of contrastive learning. Conventional techniques, such as masking, shuffling, and CutMix, may disrupt the feature dependencies in tabular data. To mitigate this, our method applies augmentations selectively to features with low attention scores, preserving the critical structure of the data while enhancing contrastive learning performance.

### 4.2 AGATa: Attention-Guided Augmentation for Tabular Data

**Attention-Guided Feature Selection** To guide augmentation, we leverage attention scores from the transformer's final layer, quantifying feature interaction importance. This approach, inspired by NLP practices, captures refined feature relationships [19, 20, 21]. Let $A \in \mathbb{R}^{B \times H \times F \times F}$ represent the attention scores, where $B$ is the batch size, $H$ is the number of heads, and $F$ is the number of features. The attention scores are averaged across all heads to compute a feature importance matrix (Equation (5)):

$$A_{\text{mean}} = \frac{1}{H} \sum_{h=1}^{H} A_h \tag{5}$$

The feature selection ratio for augmentation, denoted as $k$, defines the proportion of features selected for augmentation based on their attention scores. Specifically, $k$ represents the percentage of features with lower attention scores that will be augmented, ensuring that less critical features are targeted while preserving the core structure of the data.

From the attention scores, the average score for each feature is computed by averaging across all feature interactions (Equation (6)):

$$A_{\text{feature}} = \frac{1}{F} \sum_{i=1}^{F} A_{\text{mean}}[:, i, :] \tag{6}$$

This produces a vector $A_{\text{feature}}$, which represents the overall importance of each feature. Features with the lowest attention scores are considered the least informative, and a subset of these features is selected for augmentation based on the feature selection ratio $k$.

We used a pre-defined selection ratio $k = 0.4$, meaning 40% of the features are chosen for augmentation. This selection ratio of 0.4 was adopted because many augmentation methods commonly use 0.4 as the default value.

After conducting experiments, we found that the model's performance remained consistent using this ratio. Therefore, for simplicity, all subsequent experiments were conducted using the pre-defined ratio of $k = 0.4$.

**Dynamic Augmentation Strategy** The selected low-attention features are augmented using one of three techniques: masking, shuffling, or CutMix. Each augmentation method is tailored to modify the less informative features identified by the attention mechanism while preserving critical structures in the high-attention features. Let $\mathcal{A} = \{\text{Masking}, \text{Shuffling}, \text{CutMix}\}$ represent the set of augmentation functions. The augmentation function $\text{Aug}(x)$ can be defined as:

$$\text{Aug}(x) = a(x) \quad \text{where} \quad a \sim \text{Uniform}(\mathcal{A}) \tag{7}$$

Each augmentation process operates as follows:

- **Masking:** For the selected low-attention features, their values are replaced with a mask token, effectively nullifying their contribution to the model during that iteration, while preserving the high-attention features that carry significant information.
- **Shuffling:** The values of the low-attention features are shuffled across the batch, breaking their correlations while maintaining the consistency of high-attention features within each sample.
- **CutMix:** The low-attention features from different samples in the batch are mixed, creating new combinations while leaving the high-attention features unchanged. This encourages the model to generalize over low-impact features without disrupting the core feature relationships.

We conduct the random selection of the augmentation technique for each batch. Instead of applying a fixed augmentation method across all batches, we randomly choose between masking, shuffling, and CutMix for each iteration. This randomization enhances the model's robustness by exposing it to various augmented data while preserving the tabular data's critical structure. By focusing on the least informative features and employing a random selection of augmentation techniques, our approach strikes a balance between introducing variability and maintaining the underlying statistical properties of the data. This process ensures that the augmentations are both meaningful and structure-preserving, avoiding disruption to critical feature relationships that are essential in tabular data.

### 4.3 Overall Training Procedure

The overall training process of AGATa involves several stages, starting with the generation of augmented data and continuing through transformer-based encoding, projection layers, and contrastive learning. The main steps are outlined as follows.

**Generation of Augmented Samples** Based on the attention-guided feature selection, the augmentation module alters the low-attention features using one of three augmentation techniques: masking, shuffling, or CutMix. These augmented samples, along with their original versions, form positive pairs for the contrastive learning task, allowing the model to learn meaningful feature interactions while preserving the essential structure of the data.

**Transformer and Projection Heads** Once the augmented samples are created, they are input into a transformer model, which processes the data and generates contextual embeddings. The final hidden state of the [CLS] token is passed through a projection head, a multi-layer perceptron (MLP) with one hidden layer, which maps the embeddings to a lower-dimensional space suitable for contrastive learning. The MLP also introduces additional non-linearity to help the model better distinguish between positive and negative samples.

**Loss Function** Following augmentation, the data is used in a contrastive learning framework, where it is trained using the InfoNCE loss. Given an anchor sample $x$, its corresponding positive pair $x^+$, and all negative samples $x^-$, the InfoNCE loss is calculated as:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\text{sim}(x, x^+)/\tau)}{\sum_{x^-} \exp(\text{sim}(x, x^-)/\tau)} \tag{3}$$

where $\text{sim}(\cdot, \cdot)$ represents a similarity function (such as cosine similarity) and $\tau$ is the temperature parameter (set to $0.7$). The InfoNCE loss encourages the model to bring positive pairs (augmented and original samples) closer together in the embedding space while pushing negative samples apart, resulting in a representation space where similar data points are more tightly clustered.

**Fine-tuning** After pre-training, the model undergoes fine-tuning on downstream tasks, such as classification or regression. During this phase, the transformer-based model is further trained using labeled data, adapting the learned embeddings to the specific task. For classification, cross-entropy loss is used, while for regression, mean squared error is applied. The [CLS] token embedding is passed through a simple MLP with ReLU activation to produce the final predictions. This fine-tuning step optimizes the model for task-specific performance in real-world applications.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets** We utilize twelve open-source tabular datasets in our experiments: German Credit (CS), Phoneme (PO), FIFA (FI), Gesture Phase Prediction (GE, [22]), Churn Modeling (CH, Kaggle dataset), Eye Movements (EY, [23]), California Housing (CA, [24]), Adult (AD, [25]), Jannis (JA, [26]), Otto Group Product Classification (OT, Kaggle dataset), Higgs Small (HI, [27]), and Year (YE, [28]). For all datasets, preprocessing steps and train-validation-test splits are standardized following the protocols outlined by [3, 29].

**Implementation Details** Our method is implemented using PyTorch in Python 3.7, with all experiments conducted on an NVIDIA RTX 3090 GPU. We employ the AdamW optimizer with default settings. All results are averaged over five independent runs. Performance results for other benchmarks were taken from their recent state-of-the-art papers [3, 29].

**Comparison Methods** We compare AGATa against a variety of baselines, including the non-deep method XGBoost [30] and several deep neural networks (DNNs), such as NODE [31], TabNet [32], DCNv2 [33], FT-Transformer [4], SCARF [14], SAINT [3], and T2G-FORMER [29]. In addition, we also evaluate other commonly used DNNs like MLP and SNN (a multi-layer perceptron with SELU activation [34].

Table 1: Performance comparison of augmentation techniques. The evaluation metrics for classification and regression are accuracy (↑) and mean squared error (↓), respectively. **Bold** values indicate the best-performances and underlined values represent the second-best performances.

| Augmentation method | Binary | | | | | Multiclass | | | | Regression | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CH ↑ | AD ↑ | HI ↑ | CS ↑ | PO ↑ | GE ↑ | EY ↑ | OT ↑ | JA ↑ | CA ↓ | YE ↓ | FI ↓ |
| Masking | 0.870 | 0.860 | 0.728 | 0.743 | 0.879 | 0.570 | 0.659 | 0.804 | 0.727 | 0.342 | 9.587 | 15125.823 |
| Shuffling | 0.873 | 0.860 | 0.729 | 0.743 | 0.868 | 0.586 | 0.663 | 0.803 | 0.726 | 0.353 | 9.164 | 15266.500 |
| CutMix | 0.871 | 0.860 | 0.729 | 0.754 | 0.875 | 0.596 | 0.663 | 0.801 | 0.725 | 0.352 | 8.741 | 14976.226 |
| AGATa | **0.879** | **0.864** | **0.734** | **0.780** | **0.909** | **0.694** | **0.739** | **0.820** | **0.779** | **0.256** | **8.705** | **10108.383** |

Table 2: Comparison with XGBoost and DNNs. The evaluation metrics for classification and regression are accuracy (↑) and mean squared error (↓), respectively. The best results are highlighted in **bold** and underlined values represent the second-best performances.

| Augmentation method | Binary | | | | | Multiclass | | | | Regression | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CH ↑ | AD ↑ | HI ↑ | CS ↑ | PO ↑ | GE ↑ | EY ↑ | OT ↑ | JA ↑ | CA ↓ | YE ↓ | FI ↓ |
| ***GBDT model*** | | | | | | | | | | | | |
| XGBoost | 0.859 | 0.873 | 0.724 | 0.730 | 0.876 | 0.684 | 0.725 | 0.808 | 0.719 | 0.436 | 8.850 | 11131.716 |
| ***DNN models*** | | | | | | | | | | | | |
| MLP | 0.858 | 0.849 | 0.720 | 0.680 | 0.906 | 0.586 | 0.611 | 0.804 | 0.720 | 0.499 | 8.849 | 10754.350 |
| SNN | 0.857 | 0.854 | 0.722 | - | - | 0.647 | 0.616 | 0.811 | 0.719 | 0.498 | 8.901 | - |
| FT-Transformer | 0.867 | 0.859 | 0.731 | 0.730 | **0.909** | 0.613 | 0.708 | 0.803 | 0.732 | 0.460 | 8.852 | **9567.579** |
| NODE | 0.859 | 0.858 | 0.725 | - | - | 0.539 | 0.655 | 0.803 | 0.728 | 0.463 | 8.778 | - |
| TabNet | 0.850 | 0.850 | 0.720 | - | - | 0.600 | 0.621 | 0.791 | 0.723 | 0.513 | 8.916 | - |
| DCNv2 | 0.857 | 0.853 | 0.722 | - | - | 0.557 | 0.614 | 0.802 | 0.716 | 0.489 | 8.882 | - |
| T2G-FORMER | 0.863 | 0.862 | **0.734** | 0.730 | 0.887 | 0.656 | **0.782** | 0.819 | 0.737 | 0.455 | 8.851 | 10121.356 |
| SAINT | 0.875 | 0.863 | 0.728 | 0.737 | 0.866 | 0.565 | 0.647 | 0.800 | 0.719 | 0.355 | 8.820 | 19366.582 |
| SCARF | 0.858 | 0.842 | 0.653 | 0.720 | 0.830 | 0.485 | 0.672 | 0.604 | 0.725 | 0.458 | 8.920 | 11208.313 |
| AGATa | **0.879** | **0.864** | **0.734** | **0.780** | **0.909** | **0.694** | 0.739 | **0.820** | **0.779** | **0.256** | **8.705** | 10108.383 |

### 5.2 Experimental Results

**Comparison with Other Augmentation Techniques** We conducted a comprehensive evaluation of AGATa, comparing its performance against previously established augmentation techniques across a diverse range of tasks. The primary objective of this assessment was to determine whether AGATa's attention-guided approach could provide more consistent and significant improvements in model performance compared to randomly applied augmentation methods across various tasks. The results of our experiments demonstrate that AGATa consistently outperforms previous augmentation methods across various tasks. These findings highlight AGATa's effectiveness in enhancing model performance across different types of predictive tasks. The superior performance of AGATa can be attributed to its novel attention-guided approach to data augmentation. By selectively augmenting less important features while preserving the core structure and crucial feature relationships, AGATa creates more meaningful and effective data transformations. This targeted strategy ensures that the augmented data maintains its essential characteristics while introducing beneficial variations.

**Comparison with SoTA Models: GBDTs and DNNs** To further evaluate AGATa's performance, we conducted a comprehensive comparison against SoTA GBDT and DNN models across various tasks. Table 2 presents the results for binary classification, multiclass classification, and regression tasks. The empirical results demonstrate that AGATa not only competes with but often outperforms these established models across a wide range of tasks. Out of the 12 diverse datasets spanning binary classification, multiclass classification, and regression tasks, AGATa exhibited superior performance in 10 cases. This consistent outperformance was observed across various data complexities and task types. Notably, AGATa demonstrated strong performance against XGBoost, which is traditionally considered an effective baseline for tabular data. This is particularly interesting as it suggests that AGATa's attention-guided augmentation strategy can capture complex feature interactions that are typically well-handled by tree-based models. These findings provide strong evidence for the efficacy of AGATa as a viable alternative to existing augmentation techniques and state-of-the-art models. The consistent performance advantages observed across multiple benchmarks suggest that AGATa's novel approach to data augmentation offers tangible benefits in model performance and generalization capabilities.

## 6 Ablation Study and Analysis

**Ablation Study: Reverse Attention-Gated Augmentation** We conducted an ablation study by inverting AGATa's core mechanism and applying augmentations to features with high attention scores. This resulted in decreased performance, confirming the importance of preserving high-attention features. The performance decline was moderated, likely due to InfoNCE's contrastive nature. This study highlights the critical role of attention-guided feature selection in AGATa's effectiveness.

Table 3: Comparison of AGATa and AGATa-Reverse across different datasets. AGATa-Reverse augments features with high attention scores.

| Method | GE (↑) | CH (↑) | AD (↑) | HI (↑) | CS (↑) |
|---|---|---|---|---|---|
| AGATa | **0.694** | **0.879** | **0.864** | **0.734** | **0.780** |
| AGATa-Reverse | 0.559 | 0.850 | 0.824 | 0.727 | 0.759 |

**Ablation Study: Augmentation Technique Diversity** This study compares the performance of individual augmentation techniques (masking, shuffling, and CutMix) against AGATa-Random, which randomly selects one method per epoch. Results show that AGATa-Random, benefiting from increased augmentation diversity, outperforms fixed strategies on most datasets. This demonstrates the advantage of varied augmentation techniques in enhancing model performance.

Table 4: Comparison of AGATa variants across different datasets.

| Method | GE (↑) | CH (↑) | AD (↑) | HI (↑) | CS (↑) |
|---|---|---|---|---|---|
| AGATa | **0.694** | **0.879** | **0.864** | **0.734** | **0.780** |
| AGATa-Masking | 0.673 | 0.876 | 0.863 | **0.734** | **0.780** |
| AGATa-Shuffling | 0.657 | 0.875 | 0.862 | 0.731 | 0.764 |
| AGATa-CutMix | 0.655 | 0.875 | 0.862 | 0.732 | 0.764 |

**Sensitivity Analysis: AGATa's Performance Across Augmentation Ratios ($k$)** We evaluated AGATa's performance sensitivity to the augmentation ratio $k$ ($0.2 \leq k \leq 0.8$) across various datasets. Results revealed minimal fluctuations, with peak performance frequently observed at $k = 0.4$. This stability across $k$ values demonstrates AGATa's robustness and adaptability to diverse datasets, while suggesting that augmenting approximately 40% of features generally yields favorable performance.

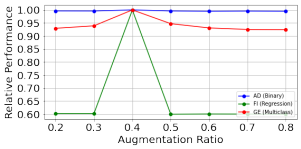



Figure 2: Relative performance across different augmentation ratio.

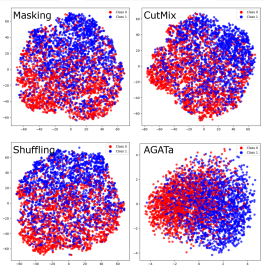



Figure 3: t-SNE visualizations

**Embedding Analysis** AGATa aims to preserve crucial data structures while enhancing diversity through random selection of augmentation functions. This approach seeks to achieve distinct separation between classes and diverse representations within classes. To validate these claims, we analyzed the embeddings produced by the trained encoder using t-SNE visualization technique. We visualized the encoder embeddings of input augmentation (Shuffling, Masking, CutMix), and AGATa using t-SNE on HI data. The results in Figure 3 show that AGATa learns embeddings that most distinctly differentiate between classes compared to other methods.

## 7 Conclusion

In this paper, we introduced AGATa, a novel approach to data augmentation for tabular data in contrastive learning. Our method addresses the unique challenges inherent to tabular data, such as its non-sequential nature, and lack of intrinsic inter-column relationships. AGATa leverages transformer-based self-attention scores to guide the augmentation process, focusing on less important features while preserving critical relationships. AGATa combines attention-guided feature selection with a dynamic augmentation strategy. By utilizing attention scores, it identifies and preserves important features during augmentation, while randomly selecting from a set of augmentation techniques for each epoch to introduce diverse variations. This design specifically caters to the unique characteristics of tabular datasets, enhancing its adaptability to various tabular data structures. Our experimental results across various datasets demonstrate improvements in downstream task performance compared to existing techniques. This work contributes to the ongoing efforts to adapt contrastive learning techniques to tabular data, which is prevalent in many real-world applications.

## References


[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

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

[3] Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

[4] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[5] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

[6] 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*, 33:11033–11043, 2020.

[7] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019.

[8] Kyungeun Lee, Ye Seul Sim, Hye-Seung Cho, Moonjung Eo, Suhee Yoon, Sanghyu Yoon, and Woohyung Lim. Binning as a pretext task: Improving self-supervised learning in tabular domains. *arXiv preprint arXiv:2405.07414*, 2024.

[9] 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.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[12] Soma Onishi and Shoya Meguro. Rethinking data augmentation for tabular data in deep learning. *arXiv preprint arXiv:2305.10308*, 2023.

[13] Pedro Machado, Bruno Fernandes, and Paulo Novais. Benchmarking data augmentation techniques for tabular data. pages 146–156, 2022.

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

[15] Zifeng Wang and Jimeng Sun. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915, 2022.

[16] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[17] Kushal Majmundar, Sachin Goyal, Praneeth Netrapalli, and Prateek Jain. Met: Masked encoding for tabular data. *arXiv preprint arXiv:2206.08564*, 2022.

[18] Sajad Darabi, Shayan Fazeli, Ali Pazoki, Sriram Sankararaman, and Majid Sarrafzadeh. Contrastive mixup: Self-and semi-supervised learning for tabular domain. *arXiv preprint arXiv:2108.12296*, 2021.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, 2019.

[22] Renan C Madeo, Fernando Lima, and Samuel M Peres. Gesture phase segmentation using support vector machines. *arXiv preprint arXiv:1301.7722*, 2013.

[23] J Salojarvi, K Puolamäki, et al. Inferring relevance from eye movements: Feature extraction. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 79–79. IEEE, 2005.

[24] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[25] Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

[26] Isabelle Guyon and Li Sun-Hosoya. Design of the 2019 chalearn autodl challenge: Automl with a human touch. In *NeurIPS 2019 Competition and Demonstration Track*, pages 135–157, 2019.

[27] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):4308, 2014.

[28] Thierry Bertin-Mahieux, Daniel PW Ellis, et al. The million song dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[29] Jiahuan Yan, Jintai Chen, Yixuan Wu, Danny Z Chen, and Jian Wu. T2g-former: Organizing tabular features into relation graphs promotes heterogeneous feature interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10720–10728, 2023.

[30] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[31] Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[32] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6679–6687, 2021.

[33] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, pages 1785–1797, 2021.

[34] Günter Klambauer, Thomas Unterthiner, et al. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 971–980, 2017.