MultiTab: A Scalable Foundation for Multitask Learning on Tabular Data

Dimitrios Sinodinos[1,2], **Jack Yi Wei**[1,2], **Narges Armanfard**[1,2]

[1]McGill University
[2]Mila - Quebec AI Institute
dimitrios.sinodinos@mail.mcgill.ca, yi.wei4@mail.mcgill.ca, narges.armanfard@mcgill.ca

Abstract

Tabular data is the most abundant data type in the world, powering systems in finance, healthcare, e-commerce, and beyond. As tabular datasets grow and span multiple related targets, there is an increasing need to exploit shared task information for improved multitask generalization. Multitask learning (MTL) has emerged as a powerful way to improve generalization and efficiency, yet most existing work focuses narrowly on large-scale recommendation systems, leaving its potential in broader tabular domains largely underexplored. Also, existing MTL approaches for tabular data predominantly rely on multi-layer perceptron-based backbones, which struggle to capture complex feature interactions and often fail to scale when data is abundant, a limitation that transformer architectures have overcome in other domains. Motivated by this, we introduce MultiTab-Net, the first multitask transformer architecture specifically designed for large tabular data. MultiTab-Net employs a novel multitask masked-attention mechanism that dynamically models feature–feature dependencies while mitigating task competition. Through extensive experiments, we show that MultiTab-Net consistently achieves higher multitask gain than existing MTL architectures and single-task transformers across diverse domains including large-scale recommendation data, census-like socioeconomic data, and physics datasets, spanning a wide range of task counts, task types, and feature modalities. In addition, we contribute MultiTab-Bench, a generalized multitask synthetic dataset generator that enables systematic evaluation of multitask dynamics by tuning task count, task correlations, and relative task complexity.

Code — https://github.com/Armanfard-Lab/MultiTab

Introduction

Tabular data is a fundamental data format that supports a wide range of industries, including finance, healthcare, and e-commerce. Its structured layout of rows and columns enables efficient storage, querying, and manipulation, supporting critical applications from large-scale recommendation systems (Li et al. 2020) to population studies based on survey data and medical diagnosis (Centers for Disease Control and Prevention 2017). As tabular data becomes increasingly ubiquitous, advancements in data collection and digital infrastructure have led to massive, high-dimensional datasets with multiple related prediction targets. In healthcare, for instance, patient records containing age, BMI, and lab results can be used to predict both the presence of diabetes and the risk of high blood pressure. Similarly, on an online retail platform, it is desirable to not only predict if a user will click on a product, but also if they will add it to their cart and make a purchase (Su et al. 2024).

In such settings, multitask learning (MTL) (Caruana 1997) offers a way to leverage task correlations and improve generalization by training models on multiple tasks simultaneously. This shared learning enhances feature representations, mitigates overfitting, and acts as a form of regularization (Vandenhende et al. 2021). Beyond improving performance, MTL improves computational efficiency by enabling a single model to handle multiple tasks in parallel, reducing training and inference time (Sinodinos and Armanfard 2022). In large-scale production environments, particularly recommendation systems, MTL has already proven effective in modeling related metrics such as click-through rate (CTR) and conversion rate (CVR) (Ma et al. 2018; Tang et al. 2020; Su et al. 2024). However, MTL for tabular data remains largely underexplored outside of recommendation systems and existing approaches typically rely on simple multi-layer perceptron (MLP) backbones (Ma et al. 2018; Tang et al. 2020; Su et al. 2024). These MLP-based backbones lack explicit architectural mechanisms for modeling complex inter-feature interactions and task-specific dependencies (Cheng et al. 2016), which can limit their scalability as datasets and task counts grow (McElfresh et al. 2023).

Transformer architectures offer a promising alternative. While MLPs implicitly learn feature interactions through dense layers, transformers can leverage self-attention to dynamically model the relationships between features and tasks (Vaswani 2017). In domains like NLP and vision, transformers have demonstrated their strength in capturing long-range dependencies and complex interactions, particularly when trained on large datasets (Devlin 2018; Dosovitskiy 2020). Specifically for tabular data, transformers can also capture sample-level dependencies through self-attention, modeling relationships across rows and achieving substantial performance gains (Somepalli et al. 2021). This ability to model both inter-feature and inter-sample depen-

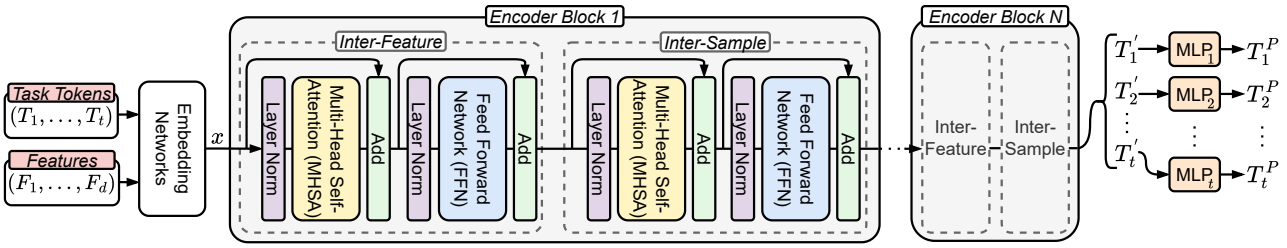

Figure 1: Overview of MultiTab-Net, our proposed multitask transformer for tabular data. A sample of d features (categorical and/or numerical) and t task tokens are passed through an embedding network to generate embeddings of size e for each token. Task tokens are appended to the feature tokens, forming a combined input sequence, which is then processed by stacked encoder blocks. Each encoder block consists of inter-feature and inter-sample attention modules, along with feed-forward networks and residual connections. After N encoder blocks, the processed task tokens $T' = \{T'_1, \ldots, T'_t\}$ are passed through task-specific multilayer perceptrons (MLP) to generate the final task predictions $T^P = \{T^P_1, \ldots, T^P_t\}$.

dencies is particularly valuable for tabular datasets, where the relative importance of columns (features) and rows (samples) often varies across tasks. These advantages, combined with the increasing availability of large-scale, multi-target tabular datasets motivates the exploration of transformer-based architectures for MTL in tabular data.

We introduce MultiTab-Net, a novel multitask transformer designed specifically for tabular data. It combines the generalization and efficiency benefits of multitask learning with transformers' ability to model complex feature interactions via attention. A key innovation is our multitask masked attention framework, the first such mechanism applied to multitask transformers. While demonstrated on tabular data, this framework has broader potential in domains like computer vision and NLP, where managing task competition while capturing intricate feature interactions is equally critical. This mechanism prevents task interference in inter-feature learning (Kendall, Gal, and Cipolla 2018), leading to superior multitask gain, Δ_m (Maninis, Radosavovic, and Kokkinos 2019), on large-scale datasets, outperforming traditional multitask MLPs and single-task transformers. Notably, the magnitude of these gains exceeds those typically reported in prior multitask tabular works (Ma et al. 2018; Tang et al. 2020; Su et al. 2024).

In addition to public tabular datasets, synthetic data has become a standard approach for evaluating multitask learning performance and robustness to variations in task correlation (Ma et al. 2018; Tang et al. 2020). The existing method by (Ma et al. 2018) allows for tunable task correlations but is limited to two tasks and lacks a notion of relative task difficulty—both key factors in multitask learning dynamics (Vandenhende et al. 2021). These constraints limit its effectiveness in analyzing multitask behaviors. To address this, we introduce MultiTab-Bench, a novel synthetic multitask dataset generator that enables full tunability of pairwise task correlations and relative task difficulty for any number of tasks, providing a more comprehensive synthetic baseline for multitask evaluations in tabular settings.

Overall, our contributions are as follows:

- We introduce **MultiTab-Net**, the first multitask transformer architecture for tabular data, which outperforms existing MLP-based multitask models and single-task transformer models on widely used tabular benchmarks.
- We propose a **novel multitask masked attention** method that mitigates the effects of task competition during inter-feature relationship learning.
- We develop **MultiTab-Bench**, a new synthetic multitask tabular dataset generator that enables fine-grained control over pairwise task correlations and relative task difficulty for any number of tasks, allowing for evaluation on virtually any multitask learning setting.

Related Works

Deep Learning has transformed domains like vision and language (Krizhevsky, Sutskever, and Hinton 2012; Devlin 2018), but has historically struggled to handle the heterogeneous feature types and irregular distributions common in tabular data. Recent studies now show that deep learning can also achieve competitive or superior performance on tabular benchmarks (McElfresh et al. 2023; Erickson et al. 2025). Multi-Layer Perceptrons (MLPs) remain strong baselines when paired with careful regularization, tuning, and embedding strategies (Kadra et al. 2021; Gorishniy, Rubachev, and Babenko 2022). Building on this, transformer-based architectures such as TabTransformer (TabT) (Huang et al. 2020), FT-Transformer (FT-T) (Gorishniy et al. 2021), and SAINT (Somepalli et al. 2021) adapt attention mechanisms to tabular data, yielding consistent gains. More recently, foundation models like TabPFN (Hollmann et al. 2022, 2025) and TabICL (Qu et al. 2025) introduced in-context learning for tabular settings, but their scalability is limited to datasets with fewer than 100k samples, making them unsuitable for large-scale applications.

Multitask learning in tabular data has received limited attention outside of recommendation systems. MMoE (Ma et al. 2018) introduced a multitask mixture-of-experts (MoE) architecture using MLPs. PLE (Tang et al. 2020) extended this design by partitioning experts into shared and task-specific groups, mitigating task interference through a more structured gating scheme. STEM (Su et al. 2024) then incorporated task-specific and shared embeddings and introduced stop-gradient constraints to prevent cross-task updates during backpropagation. This idea of controlling cross-task

interactions directly influenced the multitask masked attention mechanism in MultiTab-Net. To the best of our knowledge, MultiTab-Net is the first transformer-based MTL model for tabular data, offering a scalable architecture and novel attention design tailored for multitask learning.

MultiTab-Net: A Multitask Tabular Transformer

We propose MultiTab-Net, a novel multitask transformer architecture designed for tabular data. It introduces two major innovations: (1) a multi-token mechanism that enables parallel task processing while capturing both shared and task-specific information, and (2) a multitask masked attention mechanism that encourages task-specific focus and inhibits task interference for inter-feature attention.

MultiTab-Net Architecture

The architecture of our multitask transformer, illustrated in Figure 1, builds upon existing transformer-based tabular models, such as FT-Transformer (Gorishniy et al. 2021) and SAINT (Somepalli et al. 2021), that adopt the BERT-style approach (Devlin 2018) by adding an input token embedding corresponding to the downstream task. We extend this idea to a multitask setting by introducing a distinct task token T_i for each of the t tasks. This multi-token design enables parallel task processing, allowing each task token to interact with shared feature representations while retaining task-specific information. In Table 2, we demonstrate empirically that this multi-token approach, combined with appropriate masking, outperforms the naïve multitask variant of using one shared task token as input to t decoder heads.

The model first processes the t task tokens and d input features (both categorical and numerical) through separate embedding networks for each token. Following (Gorishniy et al. 2021; Somepalli et al. 2021; Su et al. 2024), the embedding networks ensure that all tokens are transformed into appropriate vector representations with embedding size e for the transformer layers. The resulting tokens are concatenated to form a sample $x \in \mathbb{R}^{(d+t)\times e}$, which is fed into the encoder blocks. Following (Somepalli et al. 2021), each encoder block contains two types of attention layers: Inter-Feature and Inter-Sample. The Inter-Feature layer models relationships across features (columns) within a sample. This mechanism is central to all transformers. In our model, this layer also allows tasks to attend to features and vice versa. The Inter-Sample attention layer captures patterns between samples (rows) in a batch, which has been shown to improve generalization for tabular data (Somepalli et al. 2021). Each attention layer is followed by a feed-forward network (FFN), and layer norms are used to improve training stability. After N encoder blocks, the task tokens are sent to task-specific MLPs to generate each task prediction, T_i^P.

Multitask Masked Attention

In extending transformers to a multitask setting for tabular learning, the introduction of task tokens fundamentally changes the structure of the inter-feature (column-wise) attention. Instead of attention operating solely between feature tokens, the attention matrix now contains additional quadrants; where task tokens can attend to other task tokens, task tokens can attend to feature tokens, and feature tokens can attend to task tokens. This richer interaction space enables task tokens to explicitly influence other task and feature tokens. However, it also introduces new risks of task competition and the "seesaw phenomenon" (Tang et al. 2020), where certain tasks dominate the shared capacity and harm overall performance (Vandenhende et al. 2021). Since attention scores directly govern how tokens influence each other, they present an intuitive point of intervention. If task tokens are allowed to freely influence each other or the feature tokens, dominant tasks may distort attention distributions, biasing feature–feature interactions and harming generalization. To address this, we investigate explicit masking strategies that selectively remove certain connections in the attention map, thereby limiting the pathways through which interference can arise. Concretely, we propose and evaluate three candidate masking schemes, each grounded in logical intuition:

- **F$\not\rightarrow$T**: Attention scores for the task tokens are masked from the feature tokens. The idea is to have only feature tokens influence other feature tokens, without influence from task tokens that may compete for representation.
- **T$\not\rightarrow$T**: Attention scores between different task tokens are masked. The idea is to limit their direct influence over each other and to further mitigate task competition.
- **F$\not\rightarrow$T & T$\not\rightarrow$T**: Attention scores for the task tokens are masked from the feature tokens and different task tokens. The idea is to compound the benefits from both schemes.

We now present the mathematical formulation of our masking method in the context of Inter-Feature attention. Given a sample $x \in \mathbb{R}^{(d+t)\times e}$ obtained by appending t task tokens to d feature tokens (each with embedding size e), we compute the multi-head attention scores for the i^{th} head and the output x_{out} as follows:

$$Q_i = xW_{Q_i}, \quad K_i = xW_{K_i}, \quad V_i = xW_{V_i}, \tag{1}$$

$$A_i = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_k}}\right), \quad x_{\text{att},i} = A_i V_i \tag{2}$$

$$x_{\text{out}} = \text{Concat}(x_{\text{att},1}, x_{\text{att},2}, ..., x_{\text{att},h})W_O \tag{3}$$

Here, $W_{Q_i}, W_{K_i}, W_{V_i} \in \mathbb{R}^{e\times d_k}$ are learnable projection matrices, d_k is the attention head dimension split across h heads such that $d_k = \frac{e}{h}$ for each head, and $A_i \in \mathbb{R}^{(d+t)\times(d+t)}$ captures the attention scores among all tokens. The outputs from all heads are concatenated and multiplied with the output projection layer $W_O \in \mathbb{R}^{e\times e}$ to form x_{out}. To prevent any task from disproportionately influencing the attention process, we apply a mask M_A to selectively constrain the attention maps A_i according to Equation 4.

$$A_i = \text{softmax}\left(\frac{Q_i K_i^\top}{\sqrt{d_k}} + M_A\right), \tag{4}$$

In practice, the mask M_A modifies the pre-activation attention scores by adding $-\infty$ to masked positions and 0 to unmasked ones. After applying the softmax activation, the masked entries become 0, and the unmasked scores

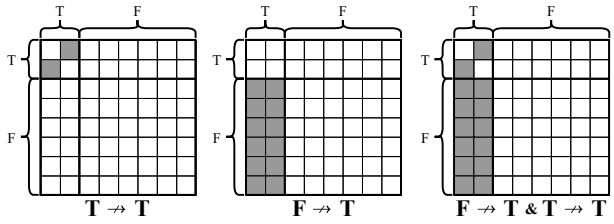

Figure 2: Illustration of the attention mask M_A under different masking schemes. Masked cells are shaded in grey and unmasked in white. In this example, there are two task tokens and six feature tokens. In $\mathbf{T} \not\rightarrow \mathbf{T}$, task tokens do not attend to other task tokens. Similarly, in $\mathbf{F} \not\rightarrow \mathbf{T}$, feature tokens do not attend to task tokens. Finally, $\mathbf{F} \not\rightarrow \mathbf{T}$ **&** $\mathbf{T} \not\rightarrow \mathbf{T}$ combines both schemes.

are unchanged. The candidate masking schemes are illustrated in Figure 2. It is worth noting that we do not evaluate masking the feature tokens from the task tokens (i.e., $\mathbf{T} \not\rightarrow \mathbf{F}$), as task tokens must attend to feature tokens for task-specific information extraction, as shown across numerous domains (Dosovitskiy 2020; Somepalli et al. 2021; Devlin 2018). In contrast to inter-feature attention, which relies on explicit token-to-token interactions that can be masked, inter-sample attention pools information across entire samples, making masking inapplicable. The formulation of our multitask inter-sample attention is detailed in the Appendix.

MultiTab-Bench: A Generalized Synthetic Multitask Dataset Generator

In tabular learning, synthetic data generation has become a valuable tool for evaluating supervised learning performance (Ma et al. 2018) and even for achieving impressive zero-shot performance on real data when generated with sufficient diversity in underlying distributions and feature interactions (Hollmann et al. 2022, 2025). However, unlike domains such as computer vision (Zamir et al. 2018; Mottaghi et al. 2014), there is a notable lack of evaluations focused on multitask tabular learning involving more than two tasks. This represents an important gap in the literature, as it has been shown that effective multitask learning models are capable of managing complex training dynamics associated with an increasing number of tasks (Standley et al. 2020; Sinodinos and Armanfard 2025). To address this, we introduce MultiTab-Bench, a novel synthetic dataset generator designed to evaluate multitask performance across an arbitrary number of tasks with tunable pairwise task correlations and relative task complexity, offering ample flexibility for simulating diverse multitask settings.

Formulation of Weight Vectors

Ma et al. (2018) introduced a synthetic multitask dataset generator designed for exactly two tasks. They would generate a weight vector for each task, w_1 and w_2, such that their cosine similarity matched a predefined correlation constant p. These weight vectors were applied to samples drawn from a standard normal distribution through a series of non-linear transformations, which empirically demonstrated a predictable influence on the Pearson correlation between the output labels. While this setup allowed for flexible control over task correlations, it was limited to two tasks and lacked the complexity needed to assess the robustness of modern multitask models. Building on their approach, we propose a generalized multitask synthetic dataset generator that supports full tunability of task correlations for an arbitrary number of tasks, while introducing additional complexities such as relative task difficulty and task-specific uncertainty (Kendall, Gal, and Cipolla 2018). Specifically, we formulate our task-specific weight vectors as follows:

1. **Define the Correlation Matrix $\mathbf{P}$**: We define $\mathbf{P} \in \mathbb{R}^{t \times t}$ as a symmetric correlation matrix where $P_{ij} = 1$ if $i = j$ to represent self-correlation, and $P_{ij} = p$ if $i \neq j$ to represent the desired pairwise correlation between tasks.
2. **Perform eigendecomposition of $\mathbf{P}$**: Since $\mathbf{P}$ is real symmetric, we have $\mathbf{P} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, where $\mathbf{Q} \in \mathbb{R}^{t \times t}$ is the orthogonal matrix of eigenvectors (not to be confused with the query matrix used in attention), and $\mathbf{\Lambda} \in \mathbb{R}^{t \times t}$ is the diagonal matrix of eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_t$. By definition of eigendecomposition, the matrix entries P_{ij} can be expressed as $P_{ij} = \sum_{k=1}^{t} Q_{ik} Q_{jk} \lambda_k$, where Q_{ik} and Q_{jk} denote the entries of matrix $\mathbf{Q}$ corresponding to the i^{th} and j^{th} rows, respectively, and the k^{th} column. Since $0 \leq p \leq 1$ and $t \geq 2$, $\mathbf{P}$ is guaranteed to be positive semidefinite (see Horn & Johnson, 2012) and thus the entries of $\mathbf{\Lambda}$ are non-negative.
3. **Construct the task-specific weight matrix**: Expressed as $\mathbf{W} = \mathbf{Q}\mathbf{\Lambda}^{1/2}\mathbf{U}^T$, where $\mathbf{\Lambda}^{1/2}$ is the diagonal matrix containing the square roots of the eigenvalues, $\mathbf{U} \in \mathbb{R}^{d \times t}$ is a matrix with columns of orthogonal unit vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t \in \mathbb{R}^d$.

We can now verify cosine similarity between weight vectors $\mathbf{w}_i$ and $\mathbf{w}_j$: $\text{cosine\_similarity}(\mathbf{w}_i, \mathbf{w}_j) = \frac{\mathbf{w}_i^T \mathbf{w}_j}{\|\mathbf{w}_i\| \|\mathbf{w}_j\|}$. Let $\mathbf{w}_i$ be the i^{th} row of $\mathbf{W}$, from $\mathbf{w}_i = \sum_{k=1}^{t} Q_{ik} \sqrt{\lambda_k} \mathbf{u}_k$ and the orthogonality of $\mathbf{U}$, we derive: $\mathbf{w}_i^T \mathbf{w}_j = \sum_{k=1}^{t} Q_{ik} Q_{jk} \lambda_k = P_{ij}$. Since $\|\mathbf{w}_i\| = 1$, the cosine similarity is exactly P_{ij}. Further details about the intermediate steps in the proof can be found in the appendix.

Data Generation

To generate task labels with predictable pairwise Pearson correlations, we first produce task-specific weight vectors with tunable pairwise cosine similarity. The data generation process proceeds as follows:

1. **Generate Input Features:** Similar to the approach in (Ma et al. 2018), the input features $\mathbf{x} \in \mathbb{R}^d$ are sampled from a standard multivariate normal distribution: $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$
2. **Generate Labels:** Task-specific labels for the i^{th} task are then generated using a polynomial transformation combined with task-specific noise such that $y_i = \sum_{k=1}^{d_i} (\mathbf{w}_i^T \mathbf{x})^k + \epsilon_i$, and $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, where d_i is the polynomial degree and σ_i^2 controls the task-specific noise variance, introducing uncertainty into the labels.

Prior work (Kendall, Gal, and Cipolla 2018) highlights the link between task uncertainty, difficulty, and training balance. By adding task-specific noise, our method enables finer control over relative task difficulty.

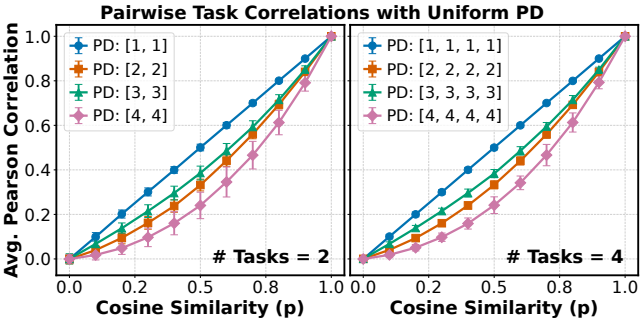

Figure 3: Average pairwise Pearson correlation for two and four tasks using the same polynomial degree (PD) for each task label. PD = [1, 1] indicates that two tasks were generated using a polynomial of degree 1, and similarly for other PD values and task counts.

For each plot in Figures 3 and 4, we use MultiTab-Bench to generate 100 datasets with 10k samples each and plot the mean and 2σ pairwise Pearson correlations between task labels. In Figure 3, all tasks share the same polynomial degree (PD), resulting in lower label correlations as PD increases, implying a more difficult multitask learning setting. Also, the mean label correlations are the same in both plots, demonstrating that tuning task correlations is predictable regardless of the number of tasks. To achieve a more complex multitask scenario, we can simply use different PD for each task, as seen in Figure 4 for a simple three-task scenario.

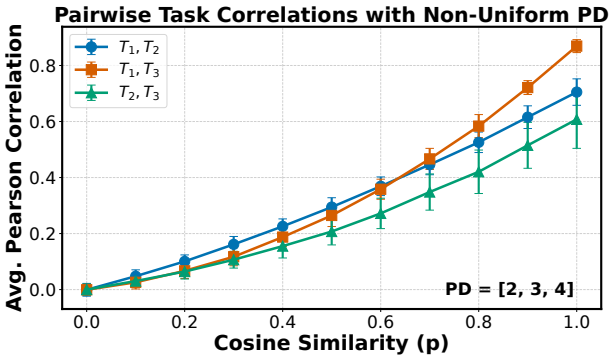

Figure 4: Average pairwise Pearson correlation for three tasks using different polynomial degrees (PD) for each task label. PD = [2, 3, 4] indicates that tasks T_1, T_2, and T_3 were generated using polynomial degrees 2, 3, and 4, respectively.

Experimental Setup

Public Datasets

AliExpress (Li et al. 2020) is a large-scale recommendation dataset containing user behavior and transaction data. We use the preprocessed version from (Xi et al. 2021), which includes two binary classification tasks: predicting product clicks (**Click**) and whether clicks convert to purchases (**Conv.**). **ACS Income** (Ding et al. 2021) is a modern replacement for the Adult Census dataset (Becker and Kohavi 1996), featuring survey data with demographic and employment attributes. We follow prior MTL setups (Ma et al. 2018) with two tasks: prediction of income level being above 50k (**Income**) and marital status classification (**Marital**). **Higgs** (Baldi, Sadowski, and Whiteson 2014) contains simulated particle collision data and is a popular single-task benchmark. We extend it to a multitask setting by predicting the main classification target (**Target**) alongside seven high-level properties (**HLP1–HLP7**) handcrafted by physicists, allowing evaluation under a higher task count. A summary of all dataset statistics can be found in Table 1, which demonstrates a variety of feature counts, feature types, task counts, task types, and class balance for categorical tasks.

Dataset	# Samples (M) (train/val/test)	# Features Num.	# Features Cat.	# Tasks Num.	# Tasks Cat.	Categorical Class Ratio (%)
Higgs	9.90/0.55/0.55	17	4	7	1	(47.01, 52.99)
AliExpress	10.94/1.22/5.56	60	15	0	2	(97.99, 2.01) (99.03, 0.07)
ACS Income	1.17/0.25/0.25	0	10	0	2	(60.56, 39.44) (54.59, 2.08, 10.74, 1.71, 30.88)

Table 1: Summary of datasets used in our experiments.

Evaluation Metrics

For classification and regression tasks, we report the area under the ROC curve (AUC) and explained variance (EV), respectively. To evaluate overall multitask performance, we use the multitask gain (Δ_m) (Maninis, Radosavovic, and Kokkinos 2019), which quantifies the average improvement achieved by a multitask learning method m compared to a single task learning baseline b across all tasks $i \in T$, as seen in equation 5.

$$\Delta_m = \frac{1}{T}\sum_{i}^{T}(-1)^{l_i}\left(\frac{M_{m,i} - M_{b,i}}{M_{b,i}}\right) \tag{5}$$

Here, $M_{m,i}$ and $M_{b,i}$ denote the performance metric for method m and baseline b respectively for task i. The value l_i is set to 1 if a lower metric value is desirable for task i, and 0 otherwise. The resulting Δ_m is interpreted as a percentage of the average improvement over the baseline b. Additional metrics and variance analysis can be found in the appendix.

Baselines

For the single-task baseline (**STL**), we follow the standard practice in MTL by using independent multi-layer perceptrons (MLPs) (Goodfellow et al. 2016) for each task. The multitask baseline (**MTL**) follows the "shared-bottom" architecture (Caruana 1997), where a portion of neurons is shared across tasks. We also compare against state-of-the-art multitask models used in large-scale recommendation systems: **MMoE** (Ma et al. 2018), **PLE** (Tang et al. 2020), and **STEM** (Su et al. 2024), which are all MLP-based. As the

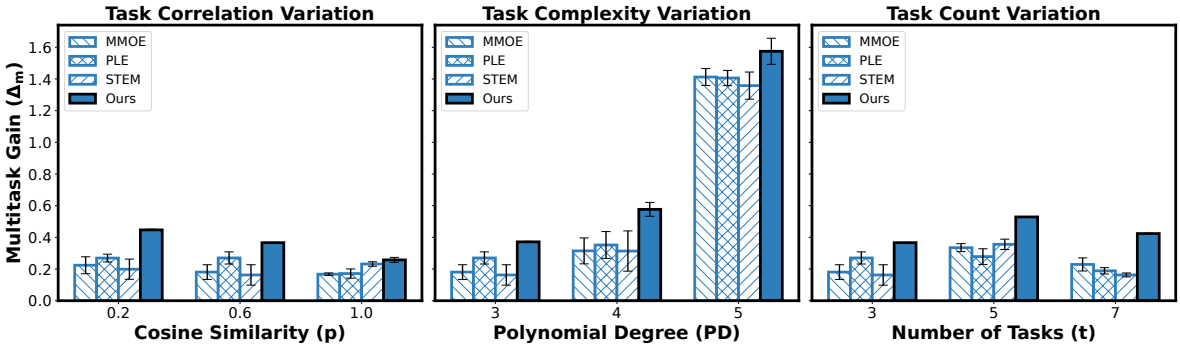

Figure 5: Average multitask gain (Δ_m) comparison under controlled variations of task properties on the synthetic benchmark. Error bars represent the standard error. We vary pairwise task correlation, task complexity, and task count.

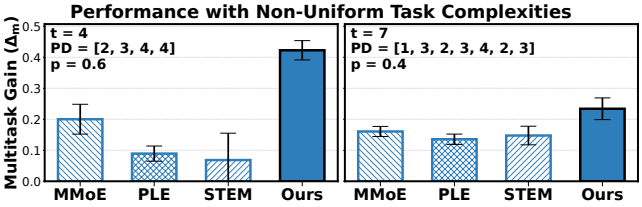

Figure 6: Average multitask gain (Δ_m) comparison using non-uniform task complexities. Error bars represent the standard error. Here, t denotes the number of tasks, *PD* specifies the polynomial degrees used to generate task labels, and p represents the cosine similarity controlling task correlation.

first multitask transformer for tabular data, there is no direct multitask baseline, so we will also evaluate single-task tabular transformers, including **TabT** (Huang et al. 2020), **FT-T** (Gorishniy et al. 2021), and **SAINT** (Somepalli et al. 2021). Lastly, we include **XGBoost** (Chen and Guestrin 2016) due to its strong performance on tabular data.

Implementation Details

We implement all baseline models from scratch, following open-source implementations closely when available to ensure a fair and unified experimental setting. It has been shown that neural networks typically require extensive tuning for each tabular dataset (McElfresh et al. 2023), so we perform a thorough hyperparameter sweep for every model using grid search with `WANDB` (Biewald 2020). Details of the sweep configurations and the selected parameters are provided in the appendix. To make the evaluation fair, we keep model capacities comparable across methods, although we note that simply increasing capacity does not always lead to better performance. For consistency, we transform the inputs to embeddings for all baselines, as prior work has shown this to improve tabular performance (Somepalli et al. 2021; Su et al. 2024). All models are trained with the Adam optimizer (Kingma 2014) and weight decay. Results are aggregated over five random seeds, with the same five seeds used for every model and dataset. All experiments are run on a single RTX A5500 GPU. Our full `PyTorch Lightning` implementation is included in the supplementary material.

MultiTab-Net		*AliExpress*	*ACS Income*	*Higgs*
# Task Tokens	Attn. Mask	$\Delta_m \uparrow$	$\Delta_m \uparrow$	$\Delta_m \uparrow$
Single	No Masking	0.2669	0.0893	-6.3491
	F$\not\to$T	0.1301	0.0837	-12.5587
Multiple	No Masking	0.2579	0.0783	1.1182
	F$\not\to$T & T$\not\to$T	0.2975	0.1007	1.0197
	F$\not\to$T	0.3698	0.0951	0.9626
	T$\not\to$T	**0.5512**	**0.1064**	**1.2337**

Table 2: Ablation of multitask attention (Attn.) masking methods for MultiTab-Net using a single shared task token and multiple task tokens (i.e., one for each task). When using a single task token, only F$\not\to$T is possible. Boldface indicates the best value per column.

Results

Ablation Study

To assess the effectiveness of our architecture and candidate masking strategies, we conduct an ablation across all three public datasets, varying the number of task tokens (single shared vs. multiple) and the candidate attention masks. As shown in Table 2, two key trends emerge. First, the advantage of using multiple task tokens increases with the number of tasks. While the difference is minimal on AliExpress and ACS Income (each with two tasks), the multi-token setup yields clear gains on the eight-task Higgs dataset. This suggests that a single shared token cannot adequately capture task-specific information in more complex multitask settings. Second, the choice of mask strongly affects performance. Notably, F$\not\to$T offers inconsistent benefits, implying that features should retain access to task context. In contrast, T$\not\to$T consistently performs the best, showing that preventing task tokens from attending to each other helps reduce task interference. Accordingly, all subsequent experiments use the multi-token configuration with T$\not\to$T masking.

Experiments with MultiTab-Bench

Our synthetic tabular data generator enables controlled variation of key multitask factors, allowing us to evaluate the

Models	*AliExpress*			*ACS Income*			*Higgs*								
	Click AUC ↑	**Conv.** AUC ↑	Δ_m ↑	**Income** AUC ↑	**Marital** AUC ↑	Δ_m ↑	**Target** AUC ↑	**HLP1** EV ↑	**HLP2** EV ↑	**HLP3** EV ↑	**HLP4** EV ↑	**HLP5** EV ↑	**HLP6** EV ↑	**HLP7** EV ↑	Δ_m ↑
STL	72.07	85.67	0.0000	90.19	88.54	0.0000	84.90	94.39	32.42	38.79	60.43	99.19	68.16	62.83	0.0000
MTL	72.30	85.59	0.1129	90.28	88.56	0.0612	84.13	93.71	32.66	38.60	59.34	96.67	68.39	62.93	-0.6531
MMoE	72.28	85.57	0.0873	90.29	88.60	0.0893	85.03	93.98	32.34	38.48	59.95	97.91	68.39	62.99	-0.3525
PLE	72.42	85.73	0.2778	90.30	88.59	0.0892	85.31	94.31	32.37	38.84	60.27	98.20	68.40	63.01	-0.0314
TabT	72.15	85.91	0.1956	90.16	88.51	-0.0336	72.61	78.90	26.10	28.66	37.54	57.23	54.90	48.88	-24.7917
FT-T	72.28	85.87	0.2624	90.13	88.46	-0.0784	78.38	94.42	31.59	36.52	54.05	**99.87**	66.85	63.05	-3.4380
SAINT	72.21	85.70	0.1146	**90.31**	88.59	0.0948	85.54	**94.44**	31.14	37.14	57.96	98.14	67.66	62.87	-1.6514
XGB	72.15	**86.02**	0.2598	89.91	88.47	-0.1948	73.45	79.36	26.51	30.79	43.80	63.50	62.73	57.37	-18.5526
STEM	72.24	85.77	0.1763	90.28	88.58	0.0725	85.32	94.29	32.59	38.78	60.36	98.35	68.36	62.98	0.0571
MultiTab-Net	**72.57**	**86.02**	**0.5512**	90.28	**88.64**	**0.1064**	**85.99**	93.99	**33.63**	**39.82**	**61.36**	97.96	**68.93**	**63.58**	**1.2337**

Table 3: Test set results for all models across all public datasets. Boldface indicates the best value per column.

effects of task correlation (p), task complexity (measured by polynomial degree, PD), and task count (t). Since it is not feasible to test every possible configuration, we isolate the effect of each factor by varying one at a time while keeping the others fixed. As shown in Figure 5, MultiTab-Net (ours) consistently achieves higher multitask gain than existing baselines (MMoE, PLE, STEM) across all variations, demonstrating robustness to changes in task relatedness, scalability with increasing task count, and improved handling of task complexity. We further evaluate two settings where tasks have markedly different levels of task complexity and observe in Figure 6 that MultiTab-Net (ours) again outperforms all baselines by a large margin. These results highlight that MultiTab-Net not only handles balanced multitask scenarios well but also excels when tasks vary widely in difficulty, which is common in real-world applications. Full dataset configurations and experimental details are provided in the appendix.

Comparison to State-of-the-Art

The results in Table 3 show that MultiTab-Net consistently achieves the highest multitask gain, Δ_m, across all datasets. Notably, MultiTab-Net consistently outperforms STEM, the most recent multitask model for tabular data, highlighting the strength of a transformer architecture for multitask learning on large-scale tabular data. The comparison with SAINT, which MultiTab-Net extends into a multitask framework, further shows the value of our design choices. Despite their architectural similarities, SAINT shows limited or even negative multitask gain, whereas MultiTab-Net achieves clear improvements. On Higgs, MultiTab-Net again delivers the strongest results, achieving a substantial Δ_m. XGBoost's poor performance aligns with findings in the original Higgs paper, where tree-based models struggled to capture the complex targets. TabT also performs poorly on Higgs because the dataset contains only four categorical features, leaving most of the architecture reduced to a shallow MLP—a limitation documented in its original work. Overall, these results demonstrate that MultiTab-Net's combination of multi-token design and T$\not\to$T masking leads to consistently better generalization and scalability across diverse tabular benchmarks.

Computational Efficiency Analysis

In Table 4, we evaluate the computational requirements for MultiTab-Net against the other multitask tabular models (MMoE, PLE, STEM), and MultiTab-Net's closest single-task counterpart (SAINT). When comparing to other MTL methods, it is clear that we have a substantial advantage in computational efficiency in ACS Income and Higgs, however, MultiTab-Net is slightly more expensive in AliExpress due to a high number of features. When comparing to SAINT, which shares many architectural similarities to MultiTab-Net, we achieve an efficiency multiplier roughly equal to the number of tasks (i.e., roughly 2x fewer cost on AliExpress and ACS Income, and 8x on Higgs), highlighting the benefits of multitask learning.

Model	*AliExpress*	*ACS Income*	*Higgs*
	Params/FLOPs (M)	Params/FLOPs (M)	Params/FLOPs (M)
SAINT	3.62 / 9.70	0.49 / 1.35	5.50 / 15.02
MMOE	**1.53 / 3.07**	0.64 / 1.25	1.05 / 2.11
PLE	1.55 / 3.11	0.65 / 1.28	1.22 / 2.46
STEM	1.55 / 3.11	0.69 / 1.29	1.25 / 2.51
MultiTab-Net	1.80 / 4.85	**0.28 / 0.77**	**0.70 / 1.90**

Table 4: Number of parameters (Params) and floating point operations (FLOPs) in millions (M). Lower values are desirable. Boldface indicates the best result in each column.

Conclusion

We introduced MultiTab-Net, the first transformer-based architecture for multitask learning on tabular data. Its innovative multi-token approach with the multitask masked-attention effectively limits task interference while capturing complex feature interactions. MultiTab-Net consistently outperformed existing MTL baselines across diverse datasets and scaled well with increasing task count and complexity. We also proposed MultiTab-Bench, a synthetic multitask dataset generator for systematic evaluation of key MTL factors, including task correlation, complexity, and count. Together, these contributions advance the state of multitask learning in tabular domains and establish new benchmarks for future research.

References

Baldi, P.; Sadowski, P.; and Whiteson, D. 2014. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1): 4308.

Becker, B.; and Kohavi, R. 1996. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Biewald, L. 2020. Experiment Tracking with Weights and Biases. Software available from wandb.com.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.

Centers for Disease Control and Prevention. 2017. CDC Diabetes Health Indicators. UCI Machine Learning Repository.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; Anil, R.; Haque, Z.; Hong, L.; Jain, V.; Liu, X.; and Shah, H. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, 7–10. New York, NY, USA: Association for Computing Machinery.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, F.; Hardt, M.; Miller, J.; and Schmidt, L. 2021. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34: 6478–6490.

Dosovitskiy, A. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Erickson, N.; Purucker, L.; Tschalzev, A.; Holzmüller, D.; Desai, P. M.; Salinas, D.; and Hutter, F. 2025. TabArena: A Living Benchmark for Machine Learning on Tabular Data. arXiv:2506.16791.

Goodfellow, I.; Bengio, Y.; Courville, A.; and Bengio, Y. 2016. *Deep learning*, volume 1. MIT Press.

Gorishniy, Y.; Rubachev, I.; and Babenko, A. 2022. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35: 24991–25004.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943.

Hollmann, N.; Müller, S.; Eggensperger, K.; and Hutter, F. 2022. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*.

Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S. B.; Schirrmeister, R. T.; and Hutter, F. 2025. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045): 319–326.

Huang, X.; Khetan, A.; Cvitkovic, M.; and Karnin, Z. 2020. TabTransformer: Tabular Data Modeling Using Contextual Embeddings. arXiv:2012.06678.

Kadra, A.; Lindauer, M.; Hutter, F.; and Grabocka, J. 2021. Well-tuned simple nets excel on tabular datasets. *Advances in neural information processing systems*, 34: 23928–23941.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Kingma, D. P. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Li, P.; Li, R.; Da, Q.; Zeng, A.-X.; and Zhang, L. 2020. Improving Multi-Scenario Learning to Rank in E-commerce by Exploiting Task Relationships in the Label Space. In *proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19- 23,2019*. New York,NY,USA: ACM.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.

Maninis, K.-K.; Radosavovic, I.; and Kokkinos, I. 2019. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1851–1860.

McElfresh, D.; Khandagale, S.; Valverde, J.; Prasad C, V.; Ramakrishnan, G.; Goldblum, M.; and White, C. 2023. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36: 76336–76369.

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qu, J.; Holzmüller, D.; Varoquaux, G.; and Morvan, M. L. 2025. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. *arXiv preprint arXiv:2502.05564*.

Sinodinos, D.; and Armanfard, N. 2022. Attentive task interaction network for multi-task learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, 2885–2891. IEEE.

Sinodinos, D.; and Armanfard, N. 2025. Cross-Task Affinity Learning for Multitask Dense Scene Predictions. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1546–1555.

Somepalli, G.; Goldblum, M.; Schwarzschild, A.; Bruss, C. B.; and Goldstein, T. 2021. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. *arXiv preprint arXiv:2106.01342*.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *International conference on machine learning*, 9120–9132. PMLR.

Su, L.; Pan, J.; Wang, X.; Xiao, X.; Quan, S.; Chen, X.; and Jiang, J. 2024. STEM: Unleashing the Power of Embeddings for Multi-task Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9002–9010.

Tang, H.; Liu, J.; Zhao, M.; and Gong, X. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 269–278.

Vandenhende, S.; Georgoulis, S.; Van Gansbeke, W.; Proesmans, M.; Dai, D.; and Van Gool, L. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3614–3633.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Xi, D.; Chen, Z.; Yan, P.; Zhang, Y.; Zhu, Y.; Zhuang, F.; and Chen, Y. 2021. Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3745–3755.

Zamir, A. R.; Sax, A.; ; Shen, W. B.; Guibas, L.; Malik, J.; and Savarese, S. 2018. Taskonomy: Disentangling Task Transfer Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.