Regularized Pairwise Relationship based Analytics for Structured Data

ZHAOJING LUO, National University of Singapore, Singapore
SHAOFENG CAI, National University of Singapore, Singapore
YATONG WANG, University of Electronic Science and Technology of China, P. R. China
BENG CHIN OOI, National University of Singapore, Singapore

In line with the increasing machine learning model inference accuracy, deep learning (DL) models have been increasingly applied to structured data for a wide spectrum of real-world applications, including product recommendations, online advertisement, healthcare analytics and risk analysis. However, unlike unstructured data, structured data is high-dimensional and sparse and therefore engenders a large number of parameters in DL, making DL models more prone to overfitting. To alleviate the overfitting problem, various regularization methods have been designed to constrain the model parameters as a means to control the model complexity. Unfortunately, these methods are often restricted to regularizing the parameter values directly without considering the intrinsic correlations and dependencies between attribute fields of structured data which is however key to effective structured data modeling.

In this paper, we re-examine DL for structured data from a new perspective of attribute interactions. In particular, we seek to explicitly model and regularize the pairwise relationships between attribute fields of structured data, in a field-adaptive manner, via a proposed attentive and interpretable framework called ATT-Reg. Specifically, in this framework, a set of *attentive weight matrices* are introduced to each attribute field for modeling obviously different relationships with its neighboring attribute fields. Further, we derive from the Bayesian viewpoint a novel *Attentive Regularization* method for imposing adaptive regularization strengths on different pairs of attribute fields, based on the *informativeness* of their relationship, which is calculated using both data-driven information and functional dependency (FD) knowledge. Such adaptive regularization facilitates each attribute field to learn discriminative and diversified representations for more effective predictive analytics. We also develop a feature attribution method for supporting more interpretable predictions. We validate the effectiveness of our ATT-Reg on six real-world datasets. Extensive experimental results show that ATT-Reg achieves significant improvement over state-of-the-art graph models, attentive models as well as regularization methods and supports an excellent degree of interpretation.

CCS Concepts: • **Applied computing**; • **Computing methodologies**; • **Information systems → Data mining**;

Additional Key Words and Phrases: Structured Data, Pairwise Relationship, Attentive Regularization, Interpretability, Feature Importance

ACM Reference Format:
Zhaojing Luo, Shaofeng Cai, Yatong Wang, and Beng Chin Ooi. 2023. Regularized Pairwise Relationship based Analytics for Structured Data. *Proc. ACM Manag. Data* 1, 1, Article 82 (May 2023), 27 pages. https://doi.org/10.1145/3588936

Authors' addresses: Zhaojing Luo, zhaojing@comp.nus.edu.sg, National University of Singapore, Singapore, Singapore; Shaofeng Cai, shaofeng@comp.nus.edu.sg, National University of Singapore, Singapore, Singapore; Yatong Wang, wangyatong@std.uestc.edu.cn, University of Electronic Science and Technology of China, Chengdu, Sichuan, P. R. China; Beng Chin Ooi, ooibc@comp.nus.edu.sg, National University of Singapore, Singapore, Singapore.

2836-6573/2023/5-ART82
https://doi.org/10.1145/3588936

1 INTRODUCTION

Structured data analytics plays a pivotal role in various future-ready industries, such as e-commerce, healthcare, fintech and logistics [6, 30, 32, 35, 46]. Mining patterns from structured data is known to be challenging due to high-dimensionality and sparsity [6, 30, 59]. In particular, each attribute field of structured data, e.g., the item ID in e-commerce or the diagnosis code in health, can comprise up to millions of unique features, presenting great challenges for learning representations from such high-dimensional and sparse data. Recently, deep learning (DL) models have achieved record-breaking performance on a wide spectrum of areas, e.g., computer vision and natural language processing. Although there is a growing interest in developing DL models for structured data, their performances have not been satisfactory because of the overfitting problem caused by the high-dimensionality and sparsity challenges [6, 49, 50, 59].

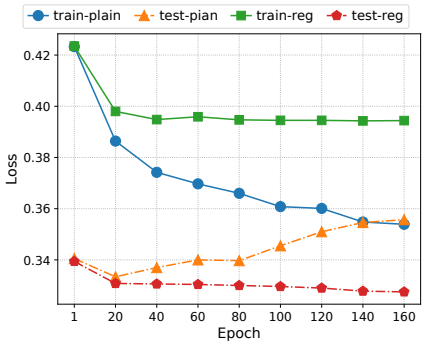


Fig. 1. Learning Curves of a DL Model Trained with and without Regularization.

Take a real-world click-through rate prediction dataset such as Avazu as an example. This dataset consists of 40,428,967 samples, 22 attribute fields and 1,544,250 different features. For this dataset, suppose we train a DL model, the Graph Neural Network (GNN) [63], where each node represents an attribute field, to predict whether a mobile ad will be clicked. The training curves of the plain and regularized GNN models are illustrated in Figure 1. For the plain model, we observe that the loss on training data (*train-plain*) keeps decreasing, while the loss on test data (*test-plain*) reaches a low point and then quickly starts to increase after only a few epochs. This suggests that as the training process progresses, the plain model trained without regularization quickly overfits to the training data and cannot generalize well to unseen test data. For real-world high-dimensional structured data, training DL models become increasingly more complex with large numbers of model parameters. Such complex and over-parameterized models can easily memorize the noise in the training data by sheer brute force and thus, tend to detract from generality over the unseen test data [6, 12, 59].

Regularization is an effective and the most commonly adopted mechanism for mitigating the overfitting problem [33, 34, 56, 60]. In general, regularization can be considered as introducing our prior knowledge about the model or training to the learning process as a means to control the model complexity, e.g., adding a regularization term to the loss function to constrain the values of model parameters. For instance, as illustrated in Figure 1, when the GNN model is regularized with L2-norm regularization [26], a widely used regularization technique in practice, the loss on the test data (*test-reg*) decreases steadily as training progresses, indicating that regularization can indeed help the model to converge and generalize better.

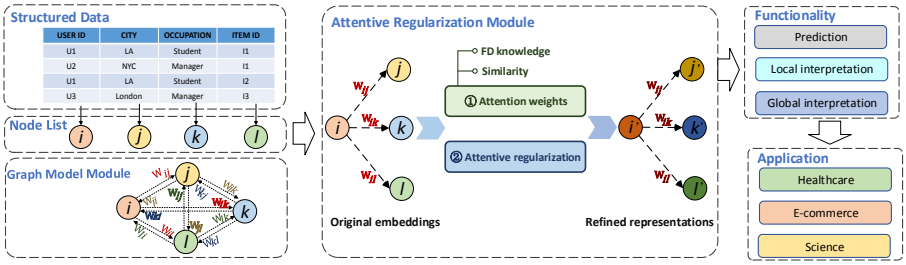


Fig. 2. Overview of ATT-Reg Framework.

Unfortunately, existing regularization methods [26, 51, 57] are often restricted to regularizing the parameter values directly by imposing the same regularization strength, without considering the informativeness of different model parameters [24, 36, 56]. Additionally and more importantly, inductive biases of structured data are not taken into consideration [28, 51, 52]. Specifically, unlike text data where the ordering of words impacts the semantic meaning, and image data where nearby pixels constitute local patterns, structured data follows set semantics, i.e., there is no natural ordering of the attribute fields. Therefore, one major challenge for modeling structured data is actually how to capture intrinsic correlations and dependencies between each attribute field and its neighboring attribute fields, which is essential for predictive analytics of structured data [6, 10, 14, 30].

In this paper, with the consideration of the above-mentioned inductive biases of structured data, we propose an attentive framework called ATT-Reg which calculates adaptive regularization to solve the overfitting problem in structured data modeling. The overview of ATT-Reg is shown in Figure 2. In our framework, the structured data is represented by a graph model, where each node is an embedding vector that represents one attribute field (nodes i, j, k, l in the figure) and each node is connected with all other nodes, i.e., attributed fields, through *attentive weight matrices* $\boldsymbol{W}$, in order to capture the no-order relationships between attribute fields [54]. The framework aims to learn a *refined representation* (nodes i', j', k', l' in Figure 2) for each attribute field according to different applications by considering every other attribute field.

Our observation is that each connecting *attentive weight matrix* $\boldsymbol{W}$ can capture the relationship between attribute field pairs, which is the key to learning each attribute field's refined representation. That is, the importance of an attribute field with respect to a given neighboring attribute field should be unique among the two fields, and therefore, discriminating the strength of regularization helps to differentiate the contribution of neighboring attribute fields for learning better refined representations. To this end, we design a novel attentive regularization module in the ATT-Reg framework to capture the intrinsic pairwise relationships between attribute fields to facilitate adaptive and customizable regularization on the corresponding attentive weight matrix.

In ATT-Reg, the pairwise relationships are captured based on a designed attention mechanism that takes into consideration both the similarity between attribute fields and the FD knowledge. Weaker regularization is imposed on attribute field pairs with a larger attention weight and vice versa. As a result, the adaptive regularization strengths help each attribute field to focus on more informative neighboring attribute fields and disregard uninformative ones, leading to diversified learned representations that help alleviate overfitting [11, 53, 64].

A key question here is how to integrate pairwise relationship into regularization? To be specific, we derive our regularization method from the Bayesian viewpoint which suggests regularization corresponds to the prior distribution of parameters. Consequently, we propose to integrate

the attentive pairwise relationship between attribute field pairs to the prior distribution of the corresponding attentive weight matrices.

For structured data applications, especially high-stake applications such as fintech and healthcare [5, 6, 16, 17, 62], interpretability is as important as predictive performance given that the users have to understand how the inference has been made in order to accept the results for decision making. As such, ATT-Reg is designed to support both global and local interpretations via a novel feature attribution method that takes advantage of the attention weights learned during the regularization process.

We summarize our contributions as follows:

- We re-examine DL for structured data from a new perspective of *attribute interactions*. In particular, we explicitly model and regularize the pairwise relationships between attribute fields of structured data, in a *feld-adaptive* manner.
- We introduce a set of *attentive weight matrices* to each attribute field for modeling the obviously different relationships with its neighboring attribute fields.
- We design a novel attention mechanism that considers both similarity information and FD knowledge to capture the pairwise relationships between attribute fields.
- We derive from the Bayesian viewpoint a novel attentive regularization method for imposing adaptive regularization strengths on different pairs of attribute fields. Such adaptive regularization facilitates each attribute field to learn discriminative and diversified representations for more effective predictive analytics.
- We design an efficient *feature attribution* method based on attentive regularization to provide easy-to-understand global and local interpretations for supporting more interpretable predictions of various structured data applications.
- We conduct extensive experiments on six real-world datasets. Results from all the datasets demonstrate that ATT-Reg achieves better performance than state-of-the-art regularization methods, i.e., L1-norm regularization [57], L2-norm regularization [26], GradClip [9] and Max-norm [28, 51, 52].

The remainder of the paper is structured as follows. Section 2 introduces the preliminaries. Section 3 presents the ATT-Reg framework. Section 4 reports the experimental results. Related works are reviewed in Section 5, and Section 6 concludes the paper.

2 PRELIMINARIES

2.1 Structured Data

Structured data (or relational data, tabular data) refers to the data that are stored as tables where each row is a *tuple*/*sample* and each column corresponds to an *attribute field* [27, 43, 44, 59]. We note that features and attribute fields refer to similar, albeit slightly different concepts. The value of an attribute field is either numerical or categorical. A domain describes the set of possible values for each attribute field. The categorical domain has a fixed number of values where each value corresponds to one feature, while each numerical domain corresponds to one feature with scalar values [6, 43, 59].

In this paper, we represent the structured data as a logical table T consisting of a set of rows and columns, where each row presents a specific sample and the columns indicate numerical/categorical attribute fields of that sample. Let y denote the dependent attribute field, i.e., the prediction target, and $\boldsymbol{x} = [x_1, \cdots, x_j, \cdots, x_J]$ denote the attribute field vector with x_j indicating the j-th attribute field.

To conduct the predictive analytics on structured data, the most common approach is to select some columns as predictor variables and then apply an analytics model to learn a function

for a specific column. While there has been a growing interest in adopting deep learning techniques for addressing database problems [13, 22, 29, 61] and integrating predictive analytics into RDBMS [4, 8, 21, 30, 40, 55], our work mainly focuses on regularizing structured data to learn better representations for effective predictive analytics.

2.2 Bayesian Interpretation of Regularization

From the Bayesian perspective, regularization corresponds to a prior distribution over the model parameters $\boldsymbol{w}$. Let $\mathcal{D}$ denote the observed data and $\boldsymbol{w}$ denote the model parameters.

According to Bayes' Theorem [1], the posterior probability of model parameters $\boldsymbol{w}$ is given by

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})}{p(\mathcal{D})}, \tag{1}$$

where $p(\mathcal{D}|\boldsymbol{w})$ is the likelihood function and $p(\mathcal{D})$ is a constant. Maximum a posteriori (MAP) methods are commonly used to estimate model parameters $\boldsymbol{w}$ as the mode of the posterior distribution. In this way, the MAP problem can be formulated as:

$$\begin{aligned} \boldsymbol{w}_{MAP} &= \underset{\boldsymbol{w}}{\operatorname{argmax}}\ p(\boldsymbol{w}|\mathcal{D}) \\ &= \underset{\boldsymbol{w}}{\operatorname{argmax}}\ p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w}) \\ &= \underset{\boldsymbol{w}}{\operatorname{argmax}}\ \log p(\mathcal{D}|\boldsymbol{w}) + \log p(\boldsymbol{w}) \end{aligned} \tag{2}$$

The term $\log p(\boldsymbol{w})$ is the log of the model parameter prior distribution, which corresponds to the regularization term. A classic example of the parameter prior distribution $p(\boldsymbol{w})$ is the Laplacian distribution and the Gaussian distribution, which corresponds to L1-norm and L2-norm regularization respectively.

2.3 Functional Dependency

Functional dependencies (FDs) [15, 41, 44] model relationships between attribute fields for relational data. An FD $X \rightarrow Y$ over a relation schema R states that the values of attribute field set X uniquely determine the values of attribute field Y, where X is a set of attribute fields in R and Y is an attribute field in R.

As a kind of integrity constraints, functional dependencies are critical for various core data management tasks, such as data repairing [45], schema normalization [42] and query optimization [7], etc. In comparison, ATT-Reg learns the relationships between attribute fields adaptively via a designed attention mechanism, which considers both data-driven information and FD knowledge, for more discriminative representations and thus better predictive analytics.

3 ATT-REG FRAMEWORK FOR STRUCTURED DATA

In this section, we first present an overview of ATT-Reg which is designed to capture pairwise relationships between attribute fields in an attentive manner with adaptive regularization. We then elaborate on each component of ATT-Reg and introduce the feature attribution method for local and global interpretation.

3.1 Overview

The main intuition of ATT-Reg is that pairwise relationships between attribute fields are significant for predictive analytics of the structured data. We seek to explicitly model and regularize the pairwise relationships between attribute fields of structured data, in a field-adaptive manner. Specifically, we first design a set of attentive weight matrices for each attribute field to model the obviously different relationships with its neighboring attribute fields. We further derive a novel attentive

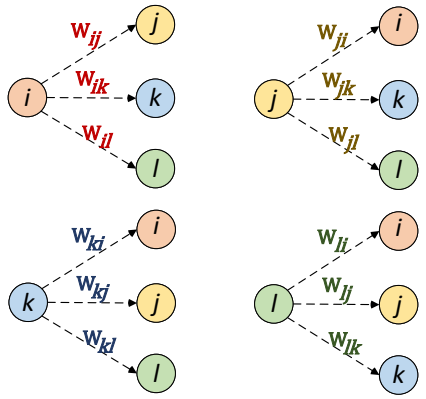


Fig. 3. Example Graph Model with Four Nodes for Structured Data.

regularization method from the Bayesian viewpoint to impose adaptive regularization strengths on different pairs of attribute fields, based on the informativeness of their relationships that are learned via a well-designed attention mechanism. Such adaptive regularization facilitates each attribute field to learn discriminative and diversified representations for more effective predictive analytics.

3.2 Graph Model Module

3.2.1 Graph Model with Attentive Weight Matrices. The graph model is designed to capture the no natural ordering property of structured data. The rationale behind the design is that we assume no prior knowledge on the attribute fields so that we construct a fully-connected graph to represent their relationships. In this graph, each node represents an attribute field and all nodes are connected in a fully-connected manner, with customized attentive weight matrices between nodes. Both categorical and numerical attribute fields are projected into the embedding space so that each node is an embedding vector. An example graph model with four nodes (attribute fields) is shown in Figure 3.

For a centre node i of layer $(m+1)$, the node representation is obtained by:

$$h_i^{(m+1)} = \sigma(\Sigma_{j \in \mathcal{N}(i)} \boldsymbol{W}_{i,j}^{(m+1)} h_j^{(m)}). \tag{3}$$

This process is also called learning *refined representations.*

3.2.2 Prediction Function. The graph model defined in Section 3.2.1 learns the refined representation for each node by considering all its neighboring nodes as Equation 3 shows. After this process, there are I embeddings of size n_e, where I denotes the number of all attribute fields. These embeddings are then concatenated to form a vector y with dimensions $I \times n_e$. Subsequently, this vector is fed to a multilayer perceptron (MLP) to derive higher-level feature representations via non-linear transformations:

$$h = \boldsymbol{MLP}(y), h \in n_h \tag{4}$$

the learned h is then fed to the final prediction layer:

$$\hat{y} = Wh + b, \tag{5}$$

where $W \in R^{n_p \times n_h}$ and $b \in n_p$ are the weight and bias respectively. n_p corresponds to the number of the prediction targets of the learning task.

3.2.3 Loss Function. The final prediction output $\hat{y}$ calculated in Equation 5 can be adopted in various learning tasks, e.g., classification, regression, as long as the corresponding loss functions are used, e.g., Cross Entropy (CE), MSE, etc. Take the binary classification task as an example, the corresponding loss function is the Binary Cross Entropy (BCE) :

$$BCELoss(\tilde{y}, \hat{y}) = -\frac{1}{N}\Sigma_{i=1}^{N}\tilde{y}_i \log \sigma(\hat{y}_i) + (1 - \tilde{y}_i)\log(1 - \sigma(\hat{y}_i)), \tag{6}$$

where $\tilde{y}$ and $\hat{y}$ are the ground truth labels and the predictions respectively. N is the number of training samples.

3.3 Attentive Regularization Module

Attentive regularization is designed to capture the intrinsic pairwise relationships between attribute fields in structured data and to impose weaker regularization on more informative attribute field pairs and vice versa.

First, we need to rethink the assumption of common L2-norm regularization, i.e., gaussian prior for model parameters. To be specific, instead of each model parameter having the same Gaussian prior, in reality, the model parameters may have priors that of different shapes of Gaussian distribution.

According to the Bayesian theorem, Gaussian with a larger variance corresponds to larger model parameters and weaker regularization, which leads to more informative features. Inspired by this fact, we argue that the prior distributions of attentive weight matrices for different pairs of fields should be different and customized in order to model the different relative importance among different pairs of attribute fields.

3.3.1 Regularization Term. The prior distribution is designed to capture the relationship between attribute fields. Specifically, we define the generation probability of each model parameter $w_{i,j}$ in the attentive weight matrix $\boldsymbol{W}_{i,j}$ that connects attribute field i with attribute field j as follows:

$$p(w_{i,j}) = C\{\exp(-\frac{1}{2}(\frac{w_{i,j}}{\frac{1}{\sqrt{\lambda\alpha_{i,j}}}})^2)\}, \tag{7}$$

where C is the normalization coefficient, and λ is the hyperparameter that controls the regularization strength of model parameter $w_{i,j}$.

This equation shows the prior distribution for model parameter $w_{i,j}$ is a Gaussian distribution with variance $\frac{1}{\lambda\alpha_{i,j}}$. Note that the variance of the Gaussian distribution is the same for the model parameters from the same attentive weight matrix, and are different for different attentive weight matrices $\boldsymbol{W}_{i,j}$, depending on the customized variance $\frac{1}{\lambda\alpha_{i,j}}$, leading to the *adaptive Gaussian distribution*.

To be specific, weaker regularization should be imposed on attentive weight matrices that connect more informative attribute field pairs and vice versa. We thus propose to learn the informativeness in an adaptive and attentive way. The key idea is that informativeness is indicated by how correlated an attribute field pair is so that each attribute field attends to more informative neighboring attribute fields. As a result, different attribute fields are able to learn diversified and discriminative refined representations.

In Equation 3, we denote the centre node i is attached with a node embedding $h_i^{(m)} \in R^{n_e}$ and the neighboring node j is attached with $h_j^{(m)} \in R^{n_e}$. We propose to calculate the similarity $z_{i,j}^{(m+1)}$between $h_i^{(m)}$ and $h_j^{(m)}$ in the layer m as follows:

$$z_{i,j}^{(m+1)} = \frac{\langle h_i^{(m)}, h_j^{(m)} \rangle}{|h_i^{(m)}||h_j^{(m)}|}. \tag{8}$$

This design of cosine similarity takes into consideration the magnitude of the node embeddings. Specifically, small node embedding values lead to nearly indistinguishable attention values, which is meaningless. This issue is addressed by the cosine similarity via normalizing each node embedding.

We then use the softmax to normalize the similarity as the *attention weights*:

$$z_{i,j}^{(m+1)} = softmax(z_{i,j}^{(m+1)}). \tag{9}$$

This attention weight determines how much a neighboring node j contributes to the centre node i. The Softmax function in Equation 9 exhibits a zero-sum game behavior in the regularization: a relaxation in one neighbor will strengthen the regularization in other neighbors, leading to diversified regularization strengths.

Lastly, we derive $\boldsymbol{\alpha}_i^{(m+1)}$ from $\boldsymbol{z}_i^{(m+1)}$ as follows:

$$\boldsymbol{\alpha}_i^{(m+1)} = \frac{1}{\boldsymbol{z}_i^{(m+1)}}. \tag{10}$$

The rationale here is that the attentive regularization method tends to impose weaker regularization on the more informative neighbors, which are measured by the attention weights calculated in Equation 9.

According to Bayesian theorem introduced in Section 2.2, after we take the negative logarithm of $p(w_{i,j}) = C\{\exp(-\frac{1}{2}(\frac{w_{i,j}}{\frac{1}{\sqrt{\lambda\alpha_{i,j}}}})^2)\}$, the regularization for attentive weight matrix between attribute field i and attribute field j of layer m can be rewritten as the adaptive L2-norm regularization as follows:

$$\begin{aligned} Reg &= \frac{1}{2}\lambda\alpha_{i,j}^{(m+1)} w_{i,j}^{(m+1)^2} = \frac{\lambda}{2}\frac{1}{z_{i,j}^{(m+1)}} w_{i,j}^{(m+1)^2} \\ &= \frac{\lambda}{2}\{[\Sigma_{k\in\mathcal{N}(i)}\exp(\frac{\langle h_i^{(m)}, h_k^{(m)}\rangle}{|h_i^{(m)}||h_k^{(m)}|})]/[\exp(\frac{\langle h_i^{(m)}, h_j^{(m)}\rangle}{|h_i^{(m)}||h_j^{(m)}|})]\} w_{i,j}^{(m+1)^2}, \end{aligned} \tag{11}$$

where the first part is the regularization strength and the second part is the quadratic term of model parameters. Equation 11 indicates that attentive regularization takes all the neighbors into consideration. Moreover, it affects model parameters through both the regularization strength and the quadratic term because these two parts will have gradients flow back to model parameters through hidden node embeddings $h_i^{(m)}$ and $h_j^{(m)}$.

In summary, if the cosine similarity of $h_i^{(m)}$ and $h_j^{(m)}$ in the same layer has larger values, higher attention weight $z_{i,j}^{(m+1)}$ will be learned which leads to weaker regularization on their attentive weight matrix $\boldsymbol{W}_{ij}^{(m+1)}$.

3.3.2 Attentive Regularization using Two Layers. For the cosine similarity defined in Section 3.3.1, the regularization strength in Equation 11 has gradients flowing back to hidden nodes $h_i^{(m)}$ and $h_j^{(m)}$, which increases their cosine similarity if the attentive weight matrix $\boldsymbol{W}_{i,j}^{(m+1)}$ is large. This will over-smooth the node representations and harms the predictive performance.

Consequently, we further propose to calculate the similarity using the node representations of two layers as follows:

$$z_{i,j}^{(m+1)} = \frac{\langle h_i^{(m+1)}, \boldsymbol{W}_{i,j}^{(m+1)} h_j^{(m)} \rangle}{|h_i^{(m+1)}||\boldsymbol{W}_{i,j}^{(m+1)} h_j^{(m)}|}. \tag{12}$$

This calculates the cosine similarity between $h_i^{(m+1)}$ and $\boldsymbol{W}_{i,j}^{(m+1)} h_j^{(m)}$, which is an adding term of $h_i^{(m+1)}$ according to Equation 3. This design involves both layers (m) and $(m+1)$, so that the regularization will not over-smooth the hidden nodes of the same layer.

Further, if the cosine similarity between $h_i^{(m+1)}$ and $\boldsymbol{W}_{i,j}^{(m+1)} h_j^{(m)}$ from two layers has larger values, higher attention weight $z_{i,j}^{(m+1)}$ will be learned which leads to weaker regularization on their attentive weight matrix $\boldsymbol{W}_{ij}^{(m+1)}$.

For this ATT-Reg using the node representations of two layers, the regularization is thus changed to:

$$\begin{aligned} Reg &= \frac{1}{2}\lambda \alpha_{i,j}^{(m+1)} {w_{i,j}^{(m+1)}}^2 = \frac{\lambda}{2} \frac{1}{{z_{i,j}}^{(m+1)}} {w_{i,j}^{(m+1)}}^2 \\ &= \frac{\lambda}{2}\{[\Sigma_{k \in \mathcal{N}(i)} \exp(\frac{\langle h_i^{(m+1)}, W_{i,k}^{(m+1)} h_k^{(m)} \rangle}{|h_i^{(m+1)}||W_{i,k}^{(m+1)} h_k^{(m)}|})]/[\exp(\frac{\langle h_i^{(m+1)}, W_{i,j}^{(m+1)} h_j^{(m)} \rangle}{|h_i^{(m+1)}||W_{i,j}^{(m+1)} h_j^{(m)}|})]\} {w_{i,j}^{(m+1)}}^2. \end{aligned} \tag{13}$$

In this equation, the first part is the regularization strength and the second part is the quadratic term of model parameters. The intuition is that the regularization strength between attribute fields i and j is related to the attention weight $z_{i,j}^{(m+1)}$, which is adaptively calculated by considering all the neighbors of the attribute field i, namely $N(i)$. The attention weight is determined by the similarity value calculated in Equation 12 using the node representations of two layers. With Equation 13, larger similarity values between i and j indicate that the corresponding attribute field pair is more informative, and thus weaker regularization is imposed.

3.3.3 Exploitation of Functional Dependencies. As mentioned in Section 2.3, FDs explicitly model the relationships between attribute fields. An FD $X \rightarrow Y$, where X is a set of attribute fields and Y is an attribute field, indicates that the attribute field Y is highly related to the attribute fields in X. Likewise, the attention weight $z_{i,j}^{(m+1)}$ calculated in Equation 9 is designed to measure the importance of the attribute field j to i. We thus propose to inject the relationship knowledge modeled by FDs into the calculation of attention weight.

To this end, we represent the relationship knowledge from FDs using a matrix $\boldsymbol{F} \in \mathcal{R}^{I \times I}$, where I is the number of all attribute fields. $\boldsymbol{F}_{i,j}$ denotes the contribution of attribute field j to attribute field i. In the first step, we initialize every entry of knowledge matrix $\boldsymbol{F}$ to zero. After which, we decompose each FD with more than one attribute fields on the right-hand side, namely the dependent attribute fields, into multiple FDs with only one dependent attribute field, so that the contributions of other attribute fields to the dependent attribute field can be derived in the next

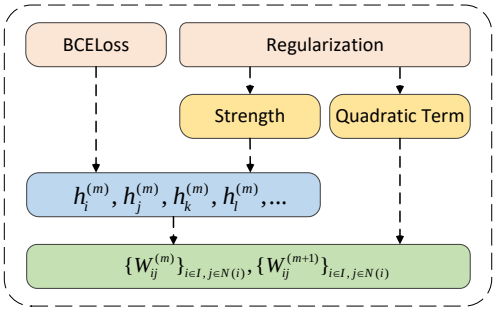


Fig. 4. Attentive Gradient Flow.

step. Lastly, we iterate all the FDs, and for a specific FD $X \rightarrow i$ with a single dependent attribute field i, where X represents a set of k attribute fields, we add $1/k$ to $\boldsymbol{F}_{i,j}$ for every $j \in X$.

Such a constructed knowledge matrix $\boldsymbol{F}$ contains the relationship information derived from FDs for every attribute field pair. This can be exploited as additional knowledge to the attention weight $z_{i,j}^{(m+1)}$ that measures the importance of the attribute field j to i. We can thus inject $\boldsymbol{F}$ into the calculation of $z_{i,j}^{(m+1)}$ and update Equation 9 as follows:

$$z_{i,j}^{(m+1)} = \frac{\exp\left(z_{i,j}^{(m+1)}\right) + \beta \boldsymbol{F}_{i,j}}{\sum_{k=1}^{N(i)} \{\exp\left(z_{i,k}^{(m+1)}\right) + \beta \boldsymbol{F}_{i,k}\}}. \tag{14}$$

The equation shows that the attention weight $z_{i,j}^{(m+1)}$ is dependent on two factors. The first is the similarity calculated using the node representations of two layers (Equation 12), which is the data-driven information. The second is the FD knowledge represented by $\boldsymbol{F}$, and β is a hyperparameter to balance the two factors.

3.3.4 Optimization. The model parameters in ATT-Reg are optimized using the gradient descent method. Figure 4 illustrates the gradient flow. The gradients of model parameters, e.g., $W_{ij}^{(m)}$, $W_{ij}^{(m+1)}$, etc, come from two sources. The first source is hidden features, e.g., $h_i^{(m)}$, $h_j^{(m)}$, etc, and the second is the quadratic term in the regularization defined in Equation 13.

Notably, two factors affect the hidden features $h_i^{(m)}$. The first factor is BCELoss defined in Equation 6 which measures the discrepancy between the ground truth and the prediction. The second factor corresponds to the first part of regularization defined in Equation 13, denoted as the regularization strength. Different from traditional regularization methods that only affect model parameter values explicitly, regularization in ATT-Reg also sends gradients to these hidden features for additional supervision. Such a regularization design produces a synergistic effect that the hidden features help to optimize model parameters with more supervision signals, and meanwhile, these model parameters support learning hidden features in an adaptive and data-driven manner for improving predictive performance.

3.4 Basis Decomposition

In the current design of the graph model module, there is one trainable attentive weight matrix per pair of nodes. Consequently, there may be drastic increase in the number of parameters in certain applications with a large number of attribute fields, which leads to the overfitting and slow training.

We thus introduce a scheme to improve the efficiency of the training by parameter sharing with basis matrices [48], where

$$\boldsymbol{W}_{i,j}^{(m+1)} = \Sigma_{p=1}^{P} \alpha_{p,i,j}^{(m+1)} \boldsymbol{B}_{p}^{(m+1)} \tag{15}$$

In this basis matrix approach, for the same layer, e.g., $(m+1)$, all the attentive weight matrices are defined as linear combinations of P basis matrices $\boldsymbol{B}_1^{(m+1)}$, $\boldsymbol{B}_2^{(m+1)}$, ..., $\boldsymbol{B}_P^{(m+1)}$, and the only node-pair-specific parameters are the P combination weights $\alpha_{1,i,j}^{(m+1)}$, $\alpha_{2,i,j}^{(m+1)}$, ..., $\alpha_{P,i,j}^{(m+1)}$ for each i, j pair.

Basis decomposition facilitates weight sharing among attentive weight matrices between different pairs of nodes, and it reduces the model size by a large margin. Specifically, denote I as the number of attribute fields, n_{in} and n_{out} as the input and output dimension of the attentive weight matrices, the model complexity can be reduced from $I \times I \times n_{in} \times n_{out}$ to $P \times n_{in} \times n_{out}$, where P is much smaller than $I \times I$ and it is related to number of attribute fields I.

In the framework of basis decomposition, now the effects of regularization are pushed from adjusting the attentive weight matrices to adjusting the basis matrices and combination weights.

3.5 Interpretation and Feature Attribution

Interpretability measures the extent to which the prediction results can be understood by human [3, 17, 37], which encourages users to accept the results for decision making. There exist general post-hoc interpretation methods which treat the trained model as a black box and provide model-agnostic interpretation results [31, 47]. However, these methods provide interpretations by approximating the model's behaviour, which is not reliable and can be misleading [17, 31]. On the contrary, we design ATT-Reg to provide interpretation results in a more *transparent* manner.

Specifically, the attention weight calculated in Equation 9 measures how much a neighboring node contributes to the centre node. Hence, for a specific node, we can thus aggregate their attention weights towards all the centre nodes as its feature attribution [17, 47], which measures the contribution of this attribute field to the final prediction. Formally, for node j of layer m, we sum $z_{i,j}^{(m+1)}$ over all centre nodes i as follows:

$$fea\_att(j^{(m)}) = \Sigma_i z_{i,j}^{(m+1)}. \tag{16}$$

The proposed feature attribution method facilitates *local interpretability*, i.e., feature attribution on a per-sample basis; Meanwhile, it supports *global interpretability* by aggregating the feature attribution of all the samples.

4 EXPERIMENTS

In this section, we evaluate the effectiveness, efficiency and interpretability of ATT-Reg using six real-world datasets.

4.1 Experimental Setup

4.1.1 Datasets. We conduct experiments on six datasets, with diversified feature types consisting of numerical features, categorical features and mixed features. The diversity of datasets and applications should allow us to show the robustness of ATT-Reg. Table 1 summarizes the characteristics of the datasets.

1) Adult Dataset[1]

[1]https://archive.ics.uci.edu/ml/datasets

Table 1. Dataset Characteristics

Dataset	# Samples	# Fields	# Features	Feature Type
Adult	48842	14	14	numerical
Bank	45211	19	70	mixed
Cardiovascular	70000	11	23	mixed
Creditcard	284807	29	29	numerical
Movielens	2006859	3	90445	categorical
Avazu	40428967	22	1544250	categorical

The Adult dataset [25] is from a real-world Census database. In this dataset, the prediction task is to determine whether a person makes over 50k a year. We use this dataset to show the usage of ATT-Reg in the social science application area. In this dataset, there are 48842 samples, 14 attribute fields including age, workclass, education, race, capital-gain, etc, with 14 different numerical embedding vectors.

2) Bank Dataset[2]

The Bank dataset [38, 39] is related to direct marketing campaigns (phone calls) of a banking institution. The classification goal is to predict if a client will subscribe a term deposit. In this dataset, there are 45211 samples and 19 attribute fields such as age, job, marital status, education, loan, etc, with 70 different numerical/categorical embedding vectors, including eight numerical attribute fields and eleven categorical attribute fields.

3) Cardiovascular Diseases Prediction Dataset[3]

For the healthcare dataset, the prediction task is to predict the presence or absence of cardiovascular disease for a patient. Cardiovascular diseases are the leading cause of death globally. By using this dataset, we want to demonstrate how can ATT-Reg contribute meaningfully to healthcare analytics. There are three types of input features, including factual information, results of medical examination and patients' demographics information. All the values were collected at the moment of medical examination. In total, there are 70000 records of patients data, 11 attribute fields, such as age, height, blood pressure, cholesterol, alcohol intake, etc, with 23 different numerical/categorical embedding vectors, including four numerical attribute fields and seven categorical attribute fields.

4) Creditcard Dataset[4]

For this finance dataset, the prediction task is the fraud detection in a bank. In modern banking, it is important to apply advanced analytics to analyze fraudulent behaviors to pre-empt or reduce fraud cases. This dataset, together with the Bank dataset, are used to evaluate the effectiveness of ATT-Reg in the finance application area. In total, there are 284807 samples, 29 attribute fields with 29 numerical embedding vectors.

5) MovieLens Dataset[5]

The MovieLens dataset [18] is from the MovieLens website, which provides movie recommendation services. The dataset describes 5-star rating and free-text tagging activity from MovieLens and the task is personalized tag recommendation. The dataset consists of 2006859 samples, three attribute fields, namely, user ID, movie ID and tag, with 90445 different categorical embedding vectors.

[2]https://archive.ics.uci.edu/ml/datasets

[3]https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset

[4]https://www.kaggle.com/jacklizhi/creditcard

[5]https://grouplens.org/datasets/movielens/

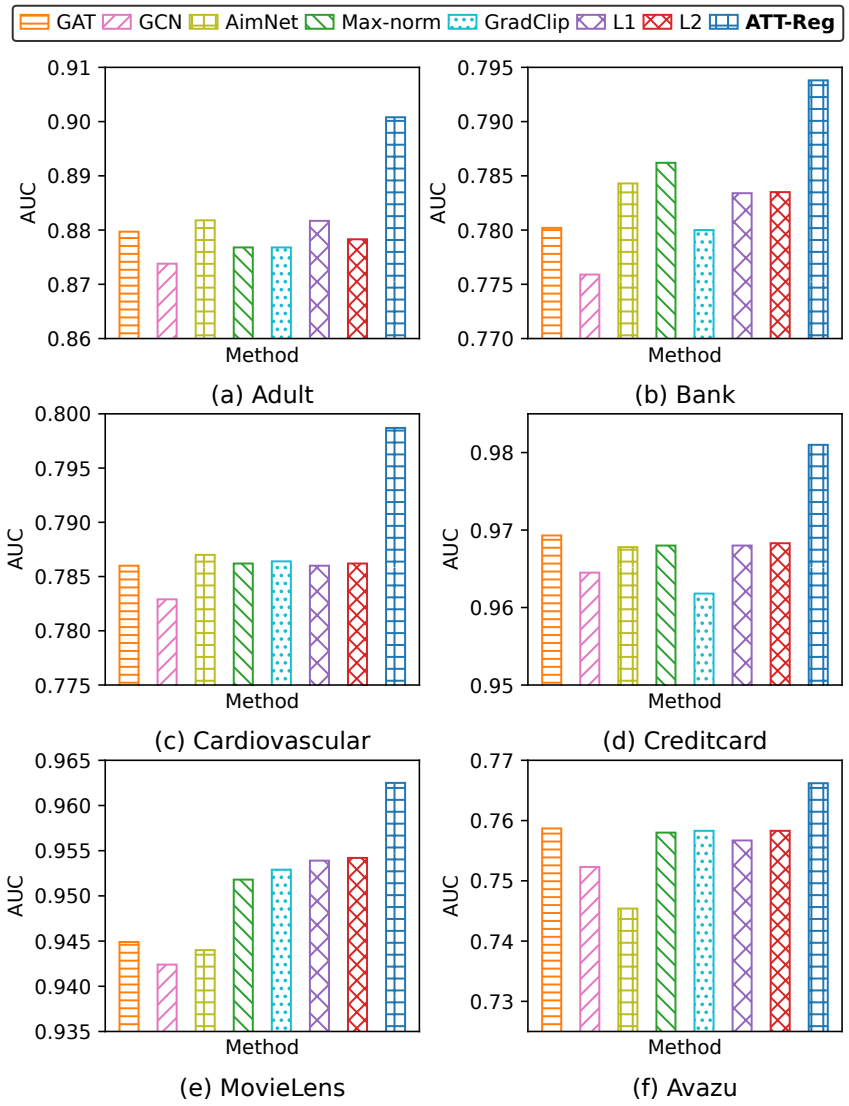


Fig. 5. Comparison between ATT-Reg and Baseline Methods.

6) Avazu Dataset[6]

Avazu is a publicly accessible online advertising click-through rate (CTR) prediction dataset. In online advertising, CTR is an important metric for evaluating the effectiveness of advertisements. Consequently, CTR predictions are essential and widely used for sponsored search and real-time bidding. This dataset consists of 11 days of Avazu data for buidling and testing prediction models. The prediction task is whether a mobile ad will be click. In total, the dataset has 40428967 samples, 22 attribute fields with 1544250 different numerical/categorical embedding vectors such as app and device information.

[6]https://www.kaggle.com/c/avazu-ctr-prediction

4.1.2 Preprocessing Methods. Each attribute field, either categorical or numerical, of the raw input is transformed into an embedding vector. To be specific, each categorical attribute field is mapped into an embedding vector via the standard embedding lookup. It should be noted that the number of embedding vectors for a categorical attribute field can be extremely large in real-world applications. Also, each numerical attribute field is transformed into an embedding vector of the same dimension by linearly scaling the corresponding embedding vector of this attribute field with the scalar attribute field value.

We divide the whole dataset into a random 6.8-1.2-2 training-validation-test split. Then we use the training dataset for training, validation dataset for determining the hyperparameters. The test dataset is used for reporting the final performance of the model.

4.1.3 Baseline Methods. We compare ATT-Reg with graph models, attentive models and regularization methods. A key property of ATT-Reg is to adaptively consider all the neighboring nodes in the graph through the attentive regularization. We thus first compare ATT-Reg with existing graph and attentive models which extract neighbor information in different ways. Secondly, we will compare ATT-Reg with existing regularization methods in deep learning models. Both ATT-Reg and the baseline regularization methods share the same model structured as introduced in Section 3.2.

The brief introduction of these baseline methods is as follows.

- GCN [23] represents each attribute field as a node in the graph and captures the interactions between neighboring nodes via graph convolution. When considering the information of neighboring nodes, GCN tends to downweight neighbors with very high degrees as it assumes information from very high-degree nodes may not be very useful.
- GAT [54] represents each attribute field as a node in the graph as GCN does. The interaction between neighboring nodes are modeled via the attention mechanism. For a specific node, it considers the neighbors via the calculated attention weights.
- AimNet [58] represents each attribute field as an embedding vector. This attentive model captures the interaction between the target attributed field and non-target attribute fields via a variation of the dot product attention mechanism.
- L1-norm regularization [57], which is also known as Lasso [36], is defined as adding the absolute values of the model parameters to the objective function. The L1-norm regularization forces insignificant model parameters to be zero, which is desirable in situations where a sparse solution is preferable. L1-norm regularization corresponds to a Laplacian prior on model parameters.
- L2-norm regularization, also known as weight decay [26], ridge regression or Tikhonov regularization [56], adds a quadratic term to the objective function. The addition of this weight decay term shrinks the values of model parameters. L2-norm regularization corresponds to a Gaussian prior on model parameters.
- Max-norm regularization [28, 51, 52] regularizes the values of the model parameters similar to L1-norm and L2-norm regularization by constraining the norm of model parameters to be bounded by a constant.
- Gradient clip (GradClip) [9] method imposes constraints on the gradients of the model parameters to ensure that the updates of model parameters at each step are not too large. This method is different from the above three regularization methods which constrain the values of model parameters directly.

4.1.4 Hyperparameter Settings. For a fair comparison, we use the same training settings for all the baselines and our ATT-Reg. Specifically, the fully-connected graph models for all the baseline regularization methods are the same as the graph model for ATT-Reg. We denote this graph model

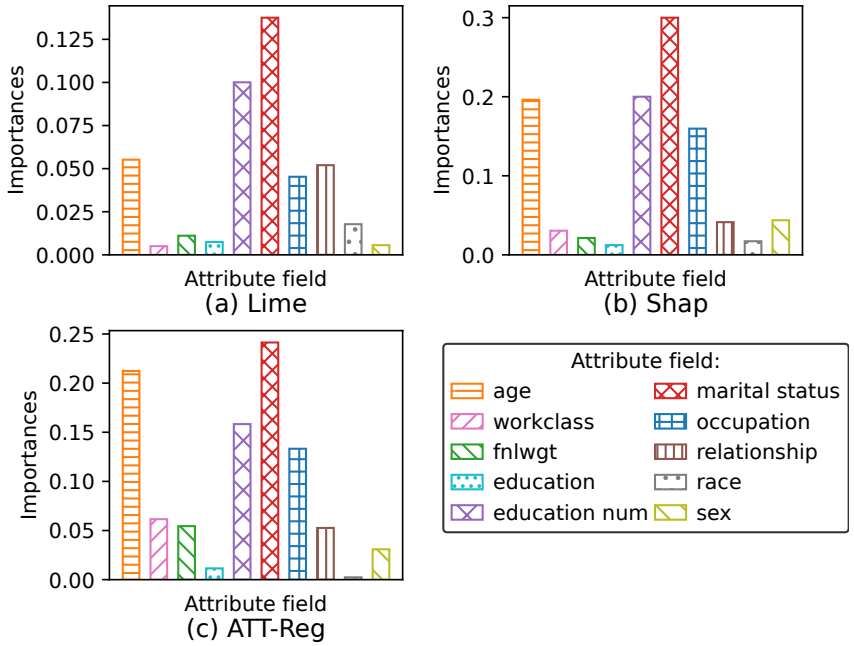


Fig. 6. Global Feature Attribution on Adult Dataset.

as the *base graph model*. For GCN, GAT, AimNet and the base graph model, the input and the hidden embedding sizes are set to 10 and 25 respectively, and the number of layers is set to two.

In all the experiments, the optimizer is Adam with a learning rate of 1e-3. We adopt a batchsize of 512 for Adult and Bank datasets, 1024 for the cardiovascular diseases prediction dataset, and 4096 for all the other datasets. The training epochs for all the datasets and methods are set to 100.

4.1.5 Evaluation Metric. As the prediction tasks of all the datasets are binary classification, and some of these datasets are unbalanced. To evaluate the proposed ATT-Reg, we use the metric, Area Under the receiver operating characteristic Curve (AUC) to measure the classification performance. A receiver operating characteristic curve is a graph showing the performance of a classification model at all classification thresholds. AUC is then calculated as the area under the curve and it measures a model's ability to distinguish between different classes. Larger AUC values indicate better performance of the model.

4.1.6 Environment. All the experiments are carried out on a server equipped with Xeon(R) Silver 4114 CPU @ 2.2GHz (10 cores), 256G memory and GeForce RTX 2080 Ti. We implement all the models with PyTorch 1.6.0 with CUDA 10.2.

4.2 Predictive Performance on Real-World Datasets

In this section, we compare our ATT-Reg with GAT, GCN, AimNet, and four state-of-the-art regularization methods on six real-world datasets.

The comparison results are shown in Figure 5. Comparing ATT-Reg with the three baseline models, namely, GAT, GCN and AimNet, GAT and GCN perform the worst on most of the datasets. This is due to the model structure. Comparing with other regularization methods which use the base graph model with attentive weight matrices between different attribute field pairs, GAT

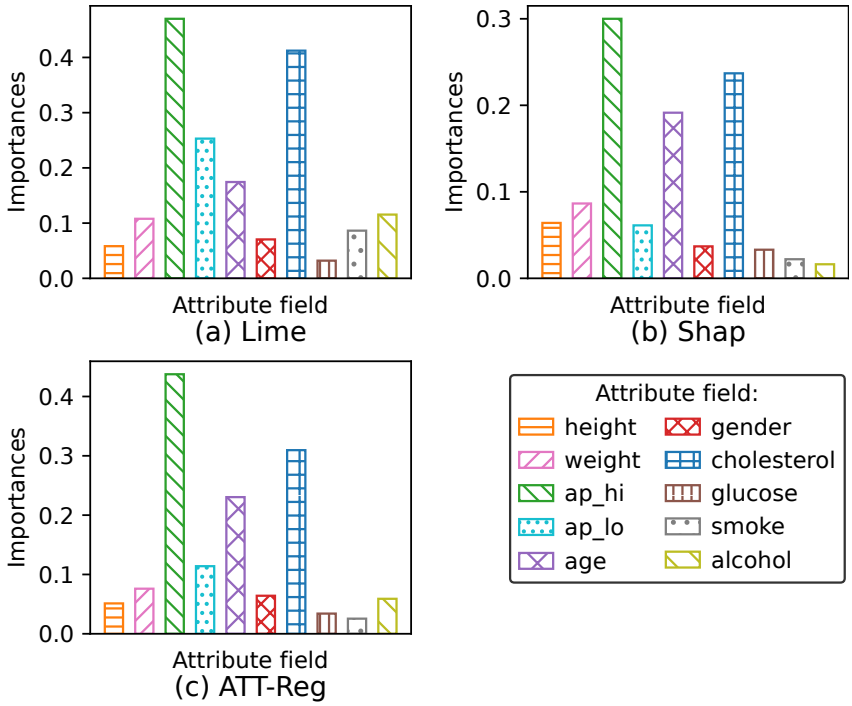
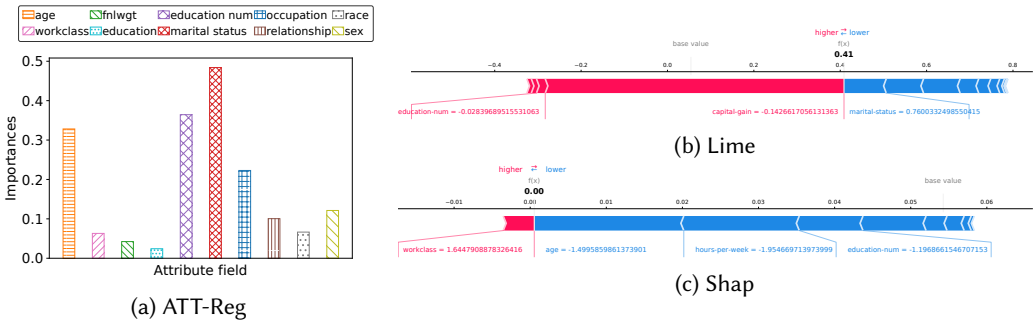


Fig. 7. Global Feature Attribution on Cardiovascular Dataset.

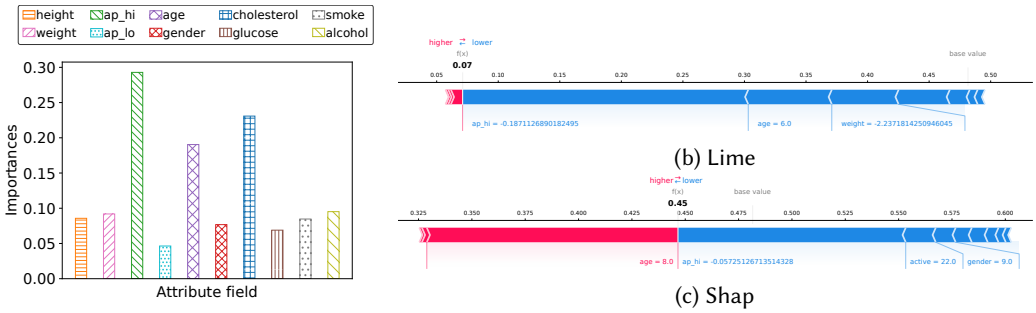
and GCN share the same weight matrix among all pair of attribute nodes, which lead to worse results. Although GAT adopts the attention mechanism to differentiate the importance of different neighboring nodes, the attention weight does not affect the shared weight matrix directly, which is less effective than ATT-Reg that affects the values of the attentive weight matrices between different attribute field pairs directly through the attentive regularization. AimNet achieves comparable or the best results among all the baseline methods on four smaller datasets. However, for the two largest datasets, namely MovieLens and Avazu, it obtains worse results than the baseline regularization methods. This could be due to the fact that AimNet has been designed for a different application. AimNet solves the problem of missing data imputation where the raw data may have missing values across different attribute fields. To predict the target attribute fields with missing values, AimNet captures the relationships between the target attribute field and non-target attribute fields via an attention mechanism. While the applications of ATT-Reg have a fixed target attribute field and complete raw data. The attention of ATT-Reg is calculated between all the non-target attribute field pairs for field interaction modeling.

Comparing ATT-Reg with the baseline regularization methods, ATT-Reg outperforms all of them. This is attributed to the adaptive and customized regularization ATT-Reg imposes on different pairs of attribute fields. For all other baseline regularization methods, although they try to constrain either the norm or the gradient of the model parameters, the regularization strength is the same for all pairs of attribute fields, without considering the informativeness of different attribute field pairs, which leads to inferior performance.



(a) ATT-Reg (b) Lime (c) Shap

Fig. 8. Local Feature Attribution on Adult Dataset.



(a) ATT-Reg (b) Lime (c) Shap

Fig. 9. Local Feature Attribution on Cardiovascular Dataset.

4.3 Interpretability

In this section, We show the interpretability results, namely, global and local feature attribution, of ATT-Reg using two representative application domains, namely, the income prediction on Adult dataset and the cardiovascular disease prediction on the healthcare dataset. For both datasets, we show 10 most important attribute fields for illustration. The explanations of the attribute fields for both datasets are publicly available from their dataset link given in Section 4.1.1.

4.3.1 Global Feature Attribution. The global feature attribution is obtained by aggregating the feature attribution of each sample as explained by Equation 16. We compare the global feature attribution results of ATT-Reg with two widely used interpretation methods, namely Lime [47] and Shap [31]. To be specific, Lime provides interpretation for a model by giving variations of the data into the tested model while Shap, whose key idea comes from the coalitional game theory, explains the prediction results by computing the contribution of each feature to the prediction. For both Lime and Shap, we feed the prediction results from ATT-Reg for test data to them and then aggregate the interpretation results of all these test samples as the global feature attribution for these two baseline interpretation methods.

Figures 6 and 7 show the global feature attribution results for two baseline interpretation methods and ATT-Reg respectively. From Figure 6(c), we can observe that the most significant attribute fields identified by ATT-Reg for the Adult dataset are age, marital status and education number. Comparing with the results of Lime and Shap in Figures 6(a) and 6(b) respectively, we found that the most significant attributes identified by ATT-Reg are consistent with those of these methods. This

Table 2. Efficiency of Interpretation

Dataset	Lime	Shap	ATT-Reg
Adult	11.43 hours	4.83 hours	5.64 seconds
Cardiovascular	4.75 hours	3.57 hours	5.62 seconds

observation indicates that the feature attribution calculated by informativeness between attribute fields in ATT-Reg is as reliable as the state-of-the-art interpretation methods. In the meantime, both the two baseline interpretation methods and ATT-Reg identify Systolic blood pressure (ap_hi) and Cholesterol as the two key attribute fields for the cardiovascular diseases prediction.

4.3.2 Local Feature Attribution. Local feature attribution is the interpretation result of a specific sample. In this section, we show a representative sample for each dataset. The interpretation results from these three methods are illustrated in Figures 8 and 9. From the figures, we observe that for both Lime and Shap, in the Adult dataset, they only capture partial significant attribute fields identified by the global feature attribution. In the meantime, they also include other attribute fields as significant ones, e.g., capital-gain captured by Lime and workclass captured by Shap. Similar phenomenon can be found in the other datasets. This indicates that these two interpretation methods may not be able to provide consistent local interpretations for single samples. A possible explanation is that Lime and Shap are surrogate models which approximate the model to provide interpretation, which may cause variations and less reliable results for specific samples.

On the contrary, the local feature attribution results of ATT-Reg are rather consistent with the global feature attribution results. For all the datasets, the top attribute fields identified by ATT-Reg for this representative sample are the same as those identified by glocal feature attribution in Figures 6 and 7. This is attributed to the mechanism of interpretation for ATT-Reg. Specifically, ATT-Reg obtains the interpretable attention weights in a more transparent manner, i.e., within the model forward pass process, which captures the intrinsic properties of the model.

It should be noted that in the cardiovascular disease prediction dataset, for this specific patient, apart from the significant attribute fields identified by the global feature attribution, namely, Systolic blood pressure (ap_hi) and Cholesterol, ATT-Reg additionally identifies age and diastolic blood pressure (ap_lo) as significant attribute fields. With this local interpretation ability, ATT-Reg is able to support personalized healthcare in daily life.

4.3.3 Comparison of Interpretation Time between ATT-Reg and Baseline Interpretation Methods. In addition to providing reliable interpretation results, the interpretation efficiency of ATT-Reg outperforms the two baseline interpretation methods by a large margin. Table 2 presents the time for generating interpretations for all the test data samples for the two datasets. From the Table, we found that while it takes Lime or Shap several hours to generate all the interpretation results, ATT-Reg generates interpretation results within several seconds. This is attributed to the differences of interpretation mechanisms between ATT-Reg and the two baseline interpretation methods. To be specific, ATT-Reg generates interpretation results via only one forward pass of the trained model on the test dataset. On the contrary, both Lime and Shap are surrogate models which need to tweak each sample slightly and do evaluation in order to generate interpretation for this specific sample. This process costs a lot more time by evaluating more extra samples.

4.4 Effectiveness of Design Choices

In this section, we discuss the effectiveness of four key design choices for ATT-Reg, namely, FD Knowledge, one layer and two layer ATT-Reg, attention designs, and the number of bases.

Table 3. Comparison between ATT-Reg and ATT-Reg-Knwl

Dataset	ATT-Reg	ATT-Reg-Knwl
Adult	0.9008	**0.9090**
Bank	0.7938	**0.8004**

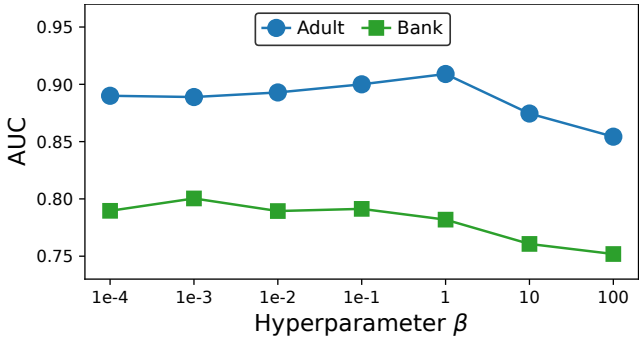


Fig. 10. Performance for Different β Values.

4.4.1 Effectiveness of FD Knowledge. In this section, we present the results of exploiting FDs in ATT-Reg on the UCI datasets which are widely used to benchmark FD discovery [2, 19, 41].

We consider the exact, minimal and non-trivial FDs, which are obtained using the TANE algorithm [20, 41] from the training dataset. For both the Adult and Bank datasets, six attribute fields are identified on the right-hand side of the FDs, i.e., the dependent attribute fields. We then construct the knowledge matrix $\boldsymbol{F}$ for each of the two datasets as introduced in Section 3.3.3, and inject such FD information into ATT-Reg.

Table 3 shows the results of ATT-Reg and ATT-Reg-Knwl that exploits FD knowledge. From Table 3, we can observe that ATT-Reg-Knwl further enhances the predictive performance of ATT-Reg, which confirms the effectiveness of taking advantage of the relationships modeled by FDs.

As defined in Equation 14, β is a key hyperparameter that trades off the data-driven information and FD knowledge. Here, we also investigate the impact of β on the prediction performance and illustrate the results in Figure 10. We find that smaller β values can generally improve the prediction performance, and a β value larger than one over-focuses on the FD knowledge, which leads to a noticeable decrease in the overall performance. This suggests that both the static FD knowledge and the dynamic data-driven information are beneficial to the model training, and a harmonious balance between the two can greatly improve the prediction performance.

4.4.2 Comparison between One Layer and Two Layer Attentive Regularization. In section 3.3.2, we analyze tha the cosine similarity defined in Section 3.3.1 has gradients flowing back to hidden nodes $h_i^{(m)}$ and $h_j^{(m)}$. This would increase their cosine similarity if the attentive weight matrix $\boldsymbol{W}_{i,j}^{(m+1)}$ is large and causes over-smoothing of the node representations. To demonstrate the adverse effects of over-smoothing, we show the comparison results of calculating cosine similarity using one layer node representations as in Equation 8 and using two layer node representations as in Equation 12. The results are shown in Table 4.

From this table, we observe that ATT-Reg dominates ATT-Reg-One-Layer in all the datasets. These results are expected because calculating the cosine similarity using the node representations of only

Table 4. Comparison Results between ATT-Reg-One-Layer and ATT-Reg

Dataset	ATT-Reg-One-Layer	ATT-Reg
Adult	0.8995	**0.9008**
Bank	0.7885	**0.7938**
Cardiovascular	0.798	**0.7987**
Creditcard	0.9804	**0.981**
Movielens	0.9614	**0.9625**
Avazu	0.765	**0.7662**

one layer is harmful for the predictive performance by over-smoothing all the node representations in the same layer.

In order to further investigate this issue, we show the global feature attribution (values without normalization) over all test samples for the top 10 attribute fields of two representative datasets, namely, Adult and Cardiovascular. The results are shown in Table 5.

As shown in the table, the global feature attributions of the top 10 attribute fields by ATT-Reg are more diversified that those by ATT-Reg-One-Layer. This is also due to the over-smoothing issue. To be specific, as Equation 16 indicates, the global feature attribution is calculated by summing the attention weights. Meanwhile, from Equations 8 and 9, we know that attention weights are obtained by calculating the cosine similarity between node representations. In ATT-Reg-One-Layer where over-smoothing happens, all the node representations tend to become the same, which leads to the same attention weights $z_{i,j}^{(m+1)}$ and further leads to the same global feature attributions.

Apart from showing the global feature attribution of the top 10 attribute fields, we also present the standard deviation of the global feature attributions of all the attribute fields for both datasets in Table 6. As shown in the table, the standard deviation values of ATT-Reg-One-Layer are much smaller than those of ATT-Reg. This confirms the fact that the global feature attribution values of different attribute fields calculated by ATT-Reg-One-Layer are very similar, which is consistent with the results presented in Table 5.

Table 5. Comparison of Global Feature Attribution

Adult Dataset										
Method	**age**	**workclass**	**fnlwgt**	**education**	**education num**	**marital status**	**occupation**	**relationship**	**race**	**sex**
ATT-Reg-One-Layer	16356.1	16334.5	16198.6	16497.8	16154.7	16267.3	16351.6	16096.5	16175.3	16551.9
ATT-Reg	18035.4	15870.0	15034.8	14218.3	17007.9	18585.9	16531.1	15000.0	14045.4	14087.2
Cardiovascular Dataset										
Method	**height**	**weight**	**ap_hi**	**ap_lo**	**age**	**gender**	**cholesterol**	**glucose**	**smoke**	**alcohol**
ATT-Reg-One-Layer	13440.1	13506.4	13939.2	14168.4	14854.2	13891.4	14943.9	13877.8	14045.2	13507.4
ATT-Reg	12819.1	13215.9	17500.0	14323.0	15685.8	13025.8	15952.5	12544.6	12407.7	12943.4

4.4.3 Comparison of Attention Designs. In Equation 10, we assume that larger attention weights indicate more informative attribute field pairs and thus weaker regularization is imposed.

Table 6. Standard Deviation of Global Feature Attribution

Dataset	ATT-Reg-One-Layer	ATT-Reg
Adult	128.553	1523.015
Cardiovascular	501.143	1870.913

Table 7. Comparison of Attention Designs

Dataset	Attn	1/Attn
Adult	0.8995	**0.9008**
Bank	0.7328	**0.7938**
Cardiovascular	0.7866	**0.7987**
Creditcard	0.9621	**0.981**
Movielens	0.9540	**0.9625**
Avazu	0.7548	**0.7662**

Table 8. Experimental Results on Basis Decomposition

Dataset	$P = 5$	ATT-Reg
Adult	**0.9017**	0.9008
Bank	**0.7944**	0.7938
Cardiovascular	0.7965	**0.7987**
Creditcard	0.9792	**0.981**
Movielens	0.9618	**0.9625**
Avazu	0.7557	**0.7662**

To verify the effectiveness of this design, we compare the performance of ATT-Reg and a variant of ATT-Reg where the $\boldsymbol{\alpha}_i^{(m+1)}$ in Equation 10 is calculated as $\boldsymbol{\alpha}_i^{(m+1)} = \boldsymbol{z}_i^{(m+1)}$. This means larger attention weights lead to stronger regularization. We denote the ATT-Reg and the variant of ATT-Reg as $\mathbf{1}/\boldsymbol{Attn}$ and $\boldsymbol{Attn}$ respectively. We tune these two methods using the same hyper-parameters and show the best results in Table 7.

From the table, we observe that $\mathbf{1}/\boldsymbol{Attn}$ outperforms $\boldsymbol{Attn}$ in all the datasets by a large margin. These results confirm that using attention weights to represent the informativeness of attribute field pairs are reasonable. In the meantime, the design of larger attention weights corresponding to weaker regularization strengths help to improve the performance of ATT-Reg.

4.4.4 Experimental Results on Basis Decomposition. In Section 3.4, we introduce how ATT-Reg can be integrated with the basis decomposition technique in order to further improve the efficiency of the whole model. For this technique, the number of basis matrices P is a key hyperparameter to trade-off between efficiency and effectiveness. To be specific, if P is set too small, the model capacity will be reduced largely which degrades the model performance greatly. On the contrary, if P is set too large, it still involves large numbers of model parameters and does not relieve the efficiency issue as well as causes overfitting.

In this experiment, we fix P to five and see the performance across different datasets. The results are shown in Table 8. From the table, we found that using basis decomposition degrades the overall performance slightly. This is expected as setting P to five reduces the number of model parameters by a large margin. In the meantime, for Adult and Bank datasets, basis decomposition achieves better results than ATT-Reg. This indicates that integrating basis decomposition with ATT-Reg has the potential to improve existing results further if we fine-tune P by taking more data characteristics, e.g., number of fields, number of features, etc, into consideration.

4.5 Sensitivity Analysis of Hyperparameters

4.5.1 Effectiveness of λ Values. As defined in Equation 7, λ is a hyperparameter that controls the strength of the regularization. Apart from the attentive regularization module which imposes

Table 9. Comparison on Per Epoch Training Time (Seconds)

Dataset	Max-Norm	Grad-Clip	Lasso	WD	ATT-Reg
Adult	0.855	0.831	0.902	0.907	1.250
Creditcard	1.947	1.967	1.923	1.948	2.163
Cardiovascular	0.917	0.811	0.838	0.831	1.088
Bank	0.864	0.872	0.910	0.915	1.259
Movielens	6.214	6.174	6.164	6.137	6.92
Avazu	6.389	6.317	6.301	6.268	8.423

regularization on the graph model module, for other components in the model, e.g., the embedding layer, the MLP before the final prediction layer, etc, weight decay (wd) is applied, which also affects the overall model performance. As a result, in this section, we investigate the effects of λ and wd on four representative datasets. The results are shown in Figure 11. From this figure, we can observe that ATT-Reg with smaller λ values performs better than the one with larger λ values. When λ exceeds 0.1, the overall performance of the model decreases dramatically. The reason is that a large λ imposes excessive regularization on the model, which greatly constrains the model capacity for learning refined representations. For wd values, they affect the model less significantly than λ since the components wd act on do not directly deal with relationships of attribute field pairs.

4.6 Efficiency Comparison

In this section, we compare the efficiency between ATT-Reg and different regularization baseline methods. We report the per epoch training time and per epoch inference time (validation and testing) in Tables 9 and 10 respectively.

In terms of training, ATT-Reg consumes a bit more time than baseline methods. The majority of the extra computational overhead comes from calculating the attention weights and the corresponding gradients for attentive regularization. Nevertheless, ATT-Reg provides both performance gain in predictive performance as well as reliable interpretation results.

During inference, ATT-Reg incurs no additional computational overhead as shown in Table 10. This is due to the fact that regularization is only imposed on model parameters during training. In validation and testing, there is no overhead from regularization.

5 RELATED WORK

5.1 Graph and Attentive Models

A key property of ATT-Reg is to adaptively consider all the neighboring nodes in the graph through the attentive regularization. For existing graph models, they have their own logic of considering neighbors, e.g., they take the average of neighbors or take the weighted average of neighbors based on degrees, e.g., GCN [23]. The rationale behind GCN is to downweight neighbors with very high degrees, since the "signal" coming from these high-degree neighbors may be less precise and informative. However, GCN is based on graph structure statistics without the adaptability/learnability during the training process. Compared with GAT [54], ATT-Reg differentiates different neighboring nodes by different attentive weight matrices and their corresponding adaptive regularization strengths. While for GAT, the weight matrices are shared by all pairs of nodes and the differences lie in the attention weights. Second, GAT's attention mechanism is on the model side and is related to labels. In comparison, ATT-Reg is not related to labels and it is regularization, i.e., prior of model

Table 10. Comparison on Per Epoch Inference Time (Seconds)

Dataset	Max-Norm	Grad-Clip	Lasso	WD	ATT-Reg
Adult	0.645	0.638	0.641	0.635	0.638
Creditcard	0.908	0.908	0.911	0.912	0.908
Cardiovascular	0.623	0.634	0.623	0.624	0.623
Bank	0.626	0.628	0.626	0.632	0.626
Movielens	2.423	2.407	2.447	2.437	2.429
Avazu	4.627	4.692	4.666	4.626	4.655

parameters, in order to prevent overfitting. Moreover, attentive regularization acts on the training phase to aid model learning, in the inference phase, since all the model parameters are learned, attention calculation is not required. In terms of attentive models, AimNet [58] is designed for the missing data imputation problem where the raw data has missing values in different attribute fields. To predict target attribute fields that have missing values, AimNet models the relationships between the target attribute field and non-target attribute fields via a variation of the dot product attention mechanism. While ATT-Reg is developed for predictive analytics with complete data and one fixed target attribute field, and the attention interaction is between non-target attribute field pairs.

5.2 Regularization

The most widely used regularization is L2-norm regularization [26], also called the weight decay [56]. It adds a quadratic term to the overall loss function in order to shrink the values of the model parameters. From the bayesian perspective, L2-norm regularization corresponds to a Gaussian prior on the model parameters.

L1-norm regularization [57], also called Lasso [36], has a similar form to L2-norm regularization, but it adds the absolute values of the model parameters to the overall loss function. The L1-norm regularization forces coefficients to be close to zero, and consequently, under the setting of the L1-norm regularization, some coefficients are exactly zero, leading to sparse models. L1-norm regularization corresponds to a Laplacian prior on the model parameters.

Similar to L1-norm and L2-norm regularization, Max-norm [28, 51, 52] also constrains the norm of the model parameters. It enforces an absolute upper bound on the L2-norm of the model parameters. In practice, it first performs the parameter updates as normal, and then enforces the constraint by clamping the model parameters. An appealing property of Max-norm is that the model will not explode even when the learning rates are set too high because the model parameters are always bounded.

Like Max-norm, gradient clip methods also impose constraints on the model parameters. However, different from Max-norm which imposes constraints on the model parameters directly, gradient clip method imposes constraints on the gradients of the model parameters so that the updates of model parameters at each step are not too large.

These methods impose the same regularization strengths on the model parameters instead of being adaptive or customized. Also, they are designed for general applications without considering the inductive biases of the structured data. For the ATT-Reg method we have proposed, we take advantage of the inductive biases of the structured data and decide the customized regularization strengths for each model parameter in an adaptive and attentive manner.

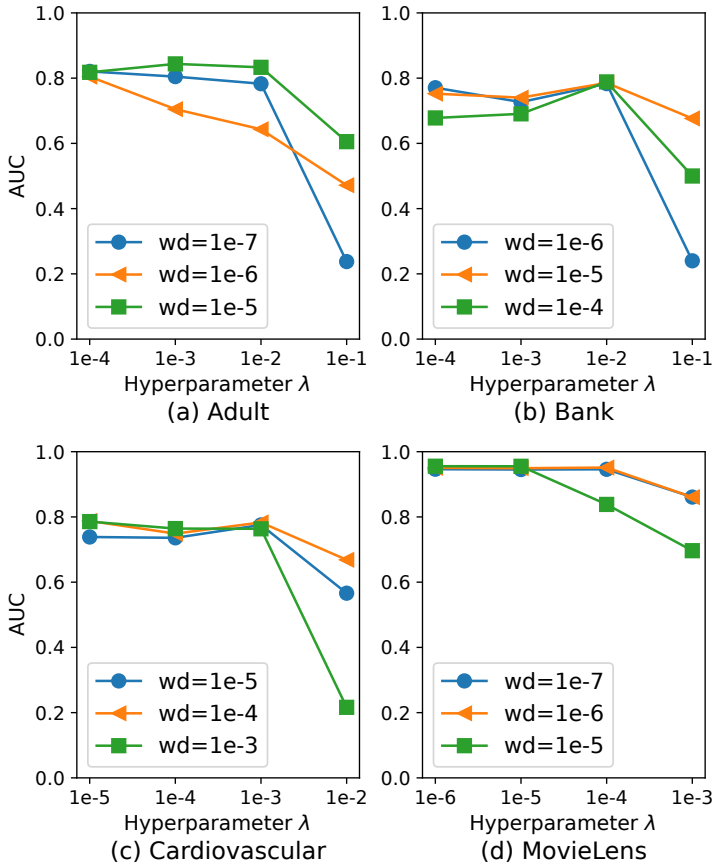


Fig. 11. Performance for Different λ and wd Values.

6 CONCLUSIONS

In this paper, we propose ATT-Reg which imposes adaptive regularization on different attribute field pairs in the structured data based on pairwise relationships. Specifically, a set of attentive weight matrices are introduced to each attribute field for modelling obviously different relationships with its neighboring attribute fields. We derive the attentive regularization method from the Bayesian viewpoint for imposing adaptive regularization on these attentive weight matrices, and the regularization strength is calculated based on the similarity information and FD knowledge. The whole framework learns the refined representation for each attribute field by adaptively considering all its neighboring attribute fields through attentive regularization. In addition, both global and local interpretations are provided for the prediction results. In order to reduce the model complexity and accelerate training, basis decomposition method is designed to impose weight sharing on the attentive weight matrices. Experiments show that ATT-Reg yields better performance in terms of AUC than existing methods and additionally provides easy-to-understand interpretations.

7 ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their constructive comments. This research is supported by Singapore Ministry of Education Academic Research Fund Tier 3 under MOE's official grant number MOE2017-T3-1-007.

REFERENCES

[1] Yuichiro Anzai. 2012. *Pattern recognition and machine learning*. Elsevier.

[2] Nabiha Asghar and Amira Ghenai. 2015. Automatic discovery of functional dependencies and conditional functional dependencies: a comparative study. *university of Waterloo* (2015).

[3] Or Biran and Courtenay Cotton. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI*, Vol. 8. 8–13.

[4] Rajesh Bordawekar and Oded Shmueli. 2017. Using word embedding to enable semantic queries in relational databases. In *Proceedings of the 1st Workshop on Data Management for End-to-End Machine Learning*. 1–4.

[5] Shaofeng Cai, Gang Chen, Beng Chin Ooi, and Jinyang Gao. 2019. Model Slicing for Supporting Complex Analytics with Elastic Inference Cost and Resource Constraints. *Proceedings of the VLDB Endowment* 13, 2 (2019), 86–99.

[6] Shaofeng Cai, Kaiping Zheng, Gang Chen, HV Jagadish, Beng Chin Ooi, and Meihui Zhang. 2021. ARM-Net: Adaptive relation modeling network for structured data. In *Proceedings of the 2021 International Conference on Management of Data*. 207–220.

[7] Upen S Chakravarthy, John Grant, and Jack Minker. 1990. Logic-based approach to semantic query optimization. *ACM Transactions on Database Systems* (1990), 162–207.

[8] Lingjiao Chen, Arun Kumar, Jeffrey Naughton, and Jignesh M. Patel. 2017. Towards Linear Algebra over Normalized Data. *Proceedings of the VLDB Endowment* 10, 11 (2017), 1214–1225.

[9] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. 2020. Understanding gradient clipping in private SGD: A geometric perspective. *Advances in Neural Information Processing Systems* 33 (2020), 13773–13782.

[10] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 3609–3616.

[11] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. 2015. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068* (2015).

[12] Tom Dietterich. 1995. Overfitting and undercomputing in machine learning. *ACM computing surveys* (1995), 326–327.

[13] Bailu Ding, Sudipto Das, Ryan Marcus, Wentao Wu, Surajit Chaudhuri, and Vivek R Narasayya. 2019. Ai meets ai: Leveraging query executions to improve index recommendations. In *Proceedings of the 2019 International Conference on Management of Data*. 1241–1258.

[14] Lun Du, Fei Gao, Xu Chen, Ran Jia, Junshan Wang, Jiang Zhang, Shi Han, and Dongmei Zhang. 2021. TabularNet: A neural network architecture for understanding semantic structures of tabular data. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 322–331.

[15] Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. 2008. Conditional functional dependencies for capturing data inconsistencies. *ACM Transactions on Database Systems* (2008), 1–48.

[16] Krishna Gade, Sahin Cem Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly. 2019. Explainable AI in industry. In *Proceedings of ACM SIGKDD international conference on knowledge discovery & data mining*. 3203–3204.

[17] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys* 51, 5 (2018), 1–42.

[18] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems* 5, 4 (2015), 1–19.

[19] Y. Huhtala, J. Karkkainen, P. Porkka, and H. Toivonen. 1998. Efficient discovery of functional and approximate dependencies using partitions. In *Proceedings 14th International Conference on Data Engineering*. 392–401.

[20] Yka Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. 1999. TANE: An efficient algorithm for discovering functional and approximate dependencies. *The computer journal* (1999), 100–111.

[21] Mahmoud Abo Khamis, Hung Q Ngo, XuanLong Nguyen, Dan Olteanu, and Maximilian Schleich. 2020. Learning models over relational data using sparse tensors and functional dependencies. *ACM Transactions on Database Systems* (2020), 1–66.

[22] Andreas Kipf, Dimitri Vorona, Jonas Müller, Thomas Kipf, Bernhard Radke, Viktor Leis, Peter Boncz, Thomas Neumann, and Alfons Kemper. 2019. Estimating cardinalities with deep sketches. In *Proceedings of the 2019 International Conference on Management of Data*. 1937–1940.

[23] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[24] Thomas Kneib, Susanne Konrath, and Ludwig Fahrmeir. 2011. High dimensional structured additive regression models: Bayesian regularization, smoothing and predictive performance. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 60, 1 (2011), 51–70.

[25] Ron Kohavi et al. 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 202–207.

[26] Anders Krogh and John A. Hertz. 1991. A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems*. 950–957.

[27] Arun Kumar, Jeffrey Naughton, Jignesh M Patel, and Xiaojin Zhu. 2016. To join or not to join? Thinking twice about joins before feature selection. In *Proceedings of the 2016 International Conference on Management of Data*. 19–34.

[28] Jason D Lee, Ben Recht, Nathan Srebro, Joel Tropp, and Russ R Salakhutdinov. 2010. Practical large-scale optimization for max-norm regularization. *Advances in neural information processing systems* 23 (2010).

[29] Guoliang Li, Xuanhe Zhou, Shifu Li, and Bo Gao. 2019. Qtune: A query-aware database tuning system with deep reinforcement learning. *Proceedings of the VLDB Endowment* 12, 12 (2019), 2118–2130.

[30] Side Li, Lingjiao Chen, and Arun Kumar. 2019. Enabling and optimizing non-linear feature interactions in factorized linear algebra. In *Proceedings of the 2019 International Conference on Management of Data*. 1571–1588.

[31] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[32] Zhaojing Luo, Shaofeng Cai, Gang Chen, Jinyang Gao, Wang-Chien Lee, Kee Yuan Ngiam, and Meihui Zhang. 2019. Improving data analytics with fast and adaptive regularization. *IEEE Transactions on Knowledge and Data Engineering* 33, 2 (2019), 551–568.

[33] Zhaojing Luo, Shaofeng Cai, Can Cui, Beng Chin Ooi, and Yang Yang. 2021. Adaptive knowledge driven regularization for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 8810–8818.

[34] Zhaojing Luo, Shaofeng Cai, Jinyang Gao, Meihui Zhang, Kee Yuan Ngiam, Gang Chen, and Wang-Chien Lee. 2018. Adaptive lightweight regularization tool for complex analytics. In *Proceedings of the 34th International Conference on Data Engineering*. 485–496.

[35] Zhaojing Luo, Sai Ho Yeung, Meihui Zhang, Kaiping Zheng, Lei Zhu, Gang Chen, Feiyi Fan, Qian Lin, Kee Yuan Ngiam, and Beng Chin Ooi. 2021. MLCask: Efficient management of component evolution in collaborative data analytics pipelines. In *Proceedings of the 37th International Conference on Data Engineering*. 1655–1666.

[36] Nicolai Meinshausen and Peter Bühlmann. 2006. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics* 34, 3 (2006), 1436–1462.

[37] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.

[38] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.

[39] Sergio Moro, Raul Laureano, and Paulo Cortez. 2011. Using data mining for bank direct marketing: An application of the crisp-dm methodology. (2011).

[40] Milos Nikolic, Haozhe Zhang, Ahmet Kara, and Dan Olteanu. 2020. F-IVM: learning over fast-evolving relational data. In *Proceedings of the 2020 International Conference on Management of Data*. 2773–2776.

[41] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schönberg, Jakob Zwiener, and Felix Naumann. 2015. Functional dependency discovery: An experimental evaluation of seven algorithms. *Proceedings of the VLDB Endowment* 8, 10 (2015), 1082–1093.

[42] J-M Petit, Farouk Toumani, J-F Boulicaut, and Jacques Kouloumdjian. 1996. Towards the reverse engineering of renormalized relational databases. In *Proceedings of the 12th International Conference on Data Engineering*. 218–227.

[43] Jiarui Qin, Weinan Zhang, Rong Su, Zhirong Liu, Weiwen Liu, Ruiming Tang, Xiuqiang He, and Yong Yu. 2021. Retrieval & Interaction Machine for Tabular Data Prediction. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1379–1389.

[44] Raghu Ramakrishnan, Johannes Gehrke, and Johannes Gehrke. 2003. *Database management systems*. Vol. 3. McGraw-Hill New York.

[45] Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. 2017. HoloClean: Holistic Data Repairs with Probabilistic Inference. *Proceedings of the VLDB Endowment* 10, 11 (2017), 1190–1201.

[46] Steffen Rendle. 2013. Scaling factorization machines to relational data. *Proceedings of the VLDB Endowment* 6, 5 (2013), 337–348.

[47] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of ACM SIGKDD international conference on knowledge discovery & data mining*. 1135–1144.

[48] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.

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

[50] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of ACM International Conference on Information and Knowledge Management*. 1161–1170.

[51] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. 2005. Maximum-margin matrix factorization. In *Advances in neural information processing systems*. 1329–1336.

[52] Nathan Srebro and Adi Shraibman. 2005. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*. Springer, 545–560.
[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
[54] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[55] Wei Wang, Meihui Zhang, Gang Chen, H. V. Jagadish, Beng Chin Ooi, and Kian-Lee Tan. 2016. Database Meets Deep Learning: Challenges and Opportunities. *SIGMOD Record* 45, 2 (2016), 17–22.
[56] Y. Wang, Xue Sun, and Le Liu. 2016. A variable step size LMS adaptive filtering algorithm based on L2 norm. In *IEEE International Conference on Signal Processing, Communications and Computing*. 1–6.
[57] Peter M Williams. 1995. Bayesian regularization and pruning using a Laplace prior. *Neural computation* 7, 1 (1995), 117–143.
[58] Richard Wu, Aoqian Zhang, Ihab Ilyas, and Theodoros Rekatsinas. 2020. Attention-based learning for missing data imputation in HoloClean. *Proceedings of Machine Learning and Systems* 2 (2020), 307–325.
[59] Yuexiang Xie, Zhen Wang, Yaliang Li, Bolin Ding, Nezihe Merve Gürel, Ce Zhang, Minlie Huang, Wei Lin, and Jingren Zhou. 2021. Fives: Feature interaction via edge search for large-scale tabular data. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3795–3805.
[60] Kai Yang, Zhaojing Luo, Jinyang Gao, Junfeng Zhao, Beng Chin Ooi, and Bing Xie. 2021. LDA-Reg: Knowledge Driven Regularization Using External Corpora. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2021), 5840–5853.
[61] Ji Zhang, Yu Liu, Ke Zhou, Guoliang Li, Zhili Xiao, Bin Cheng, Jiashu Xing, Yangtao Wang, Tianheng Cheng, Li Liu, et al. 2019. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In *Proceedings of the 2019 International Conference on Management of Data*. 415–432.
[62] Kaiping Zheng, Shaofeng Cai, Horng Ruey Chua, Wei Wang, Kee Yuan Ngiam, and Beng Chin Ooi. 2020. Tracer: A framework for facilitating accurate and interpretable analytics for high stakes applications. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1747–1763.
[63] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* (2020), 57–81.
[64] Xiaotian Zhu, Wengang Zhou, and Houqiang Li. 2018. Improving Deep Neural Network Sparsity through Decorrelation Regularization. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3264–3270.

Received July 2022; revised October 2022; accepted November 2022