August 2023

Three Essays on Artificial Intelligence in Business and Healthcare

Zongxi Liu
University of Wisconsin-Milwaukee

THREE ESSAYS ON ARTIFICIAL INTELLIGENCE IN BUSINESS AND HEALTHCARE

by

Zongxi Liu

A Dissertation Submitted in

Partial Fulfilment of the

Requirements for the Degree of

Doctor of Philosophy

in Management Science

at

University of Wisconsin-Milwaukee

August 2023

ABSTRACT

THREE ESSAYS ON ARTIFICIAL INTELLIGENCE IN BUSINESS AND HEALTHCARE
by

Zongxi Liu

The University of Wisconsin-Milwaukee, 2023
Under the Supervision of Professor Huimin Zhao

The big data era has provided researchers with challenges and opportunities for data-centric research. On the one hand, recent developments in AI technology have allowed advanced techniques to process text/image/audio/video and graph-structured data, providing new opportunities to employ big data for explanatory and predictive analytics in information systems research. On the other hand, the field requires a new level of artificial intelligence–transparent, robust, and ethical AI–to facilitate reliable business decision-making. My three dissertation essays apply, develop, and enhance state-of-the-art AI methods, leveraging various data sources as well as domain knowledge synthesis, to deal with issues in business and healthcare fields.

In Essay 1, I investigate the possibility of using deep learning models for Computed Tomography (CT) localizer image reconstruction. CT has become an important clinical imaging modality, as well as the leading source of radiation dose from medical imaging procedures. Modern CT exams are usually led by two quick orthogonal localization scans, which are used for patient positioning and diagnostic scan parameter definition. These two localization scans contribute to the patient dose but are not used for diagnosis purposes. I investigate the possibility of using deep learning models to reconstruct one localization scan image from the other, thus

reducing the patient dose and simplifying the clinical workflow. I propose a modified encoder-decoder network and a scaled mixture loss function specifically for the focal task. Experiment results indicate that although the reconstructed abdominal CT localization images may lack some details on the internal organ structures, they could be used effectively for tube current modulation calculation and patient positioning purposes, leading to a reduction of radiation dose and scan time in clinical CT exams.

In Essay 2, I propose a robust meta-graph learning method for multimodal time series prediction. Multimodal time series prediction is a difficult problem given the intricate feature interrelationships. I explore interrelationships of multilevel features in multimodal time series data and disentangle the intricate interrelationships with a robust meta-graph learning method named RMGL. The design of RMGL is rooted in theoretical foundations regarding graph convolutional networks and a novel graph attention mechanism. The core of RMGL is a meta-graph composed of three hierarchically interconnected graphs, representing feature-wise, modality-wise, and time-step-wise interrelationships, respectively. The interconnections across the graphs allow feature representations to propagate simultaneously, thereby quantifying multilevel feature interrelationships with graph structures synchronously and efficiently. Furthermore, RMGL introduces a novel weight regularization scheme to effectively learn the meta-graph for prediction based on the low-pass nature of graph convolutional filters. RMGL outperformed state-of-the-art alternatives in an empirical evaluation with a financial risk prediction task. Ablation experiments and further analyses indicated the effectiveness of RMGL.

In Essay 3, I propose a knowledge-enhanced, transformer-based text categorization model to detect employee trust indices from employee reviews. The indices of Employee Trust Model (ETM) are intangibles. Extant measurement options that require members of an organization to complete surveys make it difficult to collect data from large samples of firms across times. The use of small samples has led to conflicting results in managerial and finance research and made findings less appealing to practitioners. Furthermore, the absence of data in the time dimension has restricted analytical methods in use and limited the application of theoretical frameworks. I propose *DeepEmployee*, a novel design artifact based on automated text classification, to detect ETM indices from employee-generated reviews. *DeepEmployee* stems from design science research and includes three cohesive and complementary parts: (1) domain-specific knowledge construction based on theoretical frameworks in the management field, (2) a state-of-the-art deep learning design artifact that incorporates domain-specific knowledge to improve performance, and (3) a rigorous two-part evaluation of improvements in ETM detection and increased explanatory and predictive power in downstream tasks.

To

my parents,

my sister,

my children,

and my wife.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to express my heartful gratitude to my advisor, Dr. Huimin Zhao, for unwavering support throughout my Ph.D. study. I have always valued his invaluable advice and encouragement, which enabled me to pursue my research and achieve significant milestones. Additionally, I extend my sincere appreciation to Dr. Donglai Huo, Dr. Xiang Fang, Dr. Scott Schanke, Dr. Gang Chen, and Dr. Shuaiyong Xiao for their inspiring contributions during my dissertation research.

I am deeply grateful to my esteemed committee members: Dr. Huimin Zhao, Dr. Yang Wang, Dr. Scott Schanke, Dr. Xiang Fang, and Dr. Gang Chen. Their guidance and expertise have been instrumental in shaping my research.

I also would like to express my gratitude to the entire faculty and staff at the Lubar College of Business for their financial and academic support toward my scholarship. In particular, I am thankful to Dr. Atish Sinha for advising me on my first-year paper and guiding me into the information systems research field.

Finally, I want to convey my heartful appreciation to my family members for their unwavering support throughout my Ph.D. journey. I would not have been able to complete this degree without them. I would also like to thank my two wonderful children, Jinxi and Benjamin, for their patience and understanding during this demanding time. I dedicate my graduation success to my parents, my sister, and my wife, whose unwavering support has been the cornerstone of my academic journey.

1. Essay 1: Abdominal Computed Tomography Localizer Image Generation: A Deep Learning Approach

1.1 Introduction

Medical imaging has now become an essential part of modern medicine, and computed tomography (CT) has been one of the most important medical imaging modalities since it was invented in the 1970s. It has been estimated that more than 90 million CT scans were performed in the U.S. in 2019 (Division 2019). While CT provides invaluable diagnostic information, it alone contributes almost half of the radiation dose from medical use and one quarter of the average radiation dose in the U.S. (NCRP 2019). It has been estimated that about 0.4% of all cancers in the U.S. may be attributable to the radiation from CT studies (De Gonzalez et al. 2004). Reducing unnecessary radiation dose in CT studies could directly lead to lowered radiation-induced cancer risks, in addition to possible workflow and time savings.

"Normal" or diagnostic CT images are acquired by having the patient lie on a table that moves through the gantry while an x-ray tube rotates around the table and shoots x-rays through the patient body. However, before these images are acquired, one or more CT "localization" images are usually acquired first, where the x-ray tube is in stationary position and the table moves through the scan field. These localization images are not cross-sectional and are more similar to general x-ray images. These localization images have various names from manufacturers, including localizer, scout, topogram, scanogram, pilot, surview, and preview, and we will call them *localizer images* herein. Localizer images are acquired mostly for two purposes: (1) to confirm the location of the patient and anatomy in the field of view and determine the location and range for the following diagnostic CT scan; and (2) to determine the parameters for the following diagnostic CT scan at different locations using tube current modulation. Although these localizer images usually do not provide additional diagnostic information due to low image quality, research shows that they account for 0.4%-8.6% of the corresponding organ doses for a typical CT scan and 1.1%-20.8% of the organ doses for a low-dose lung cancer screening scan (Hoye et al. 2019).

In the current clinical practice, usually two orthogonal localizer images are acquired before the diagnostic CT scans can be started: an anterior–posterior (AP) view and a lateral view (Figure 1.1

shows an example). It is desirable to reduce the number of localizers if possible. If only one localizer image is acquired instead of two, about half of the dose introduced by localizer scans could be avoided, and the clinical workflow and efficiency will be improved significantly.

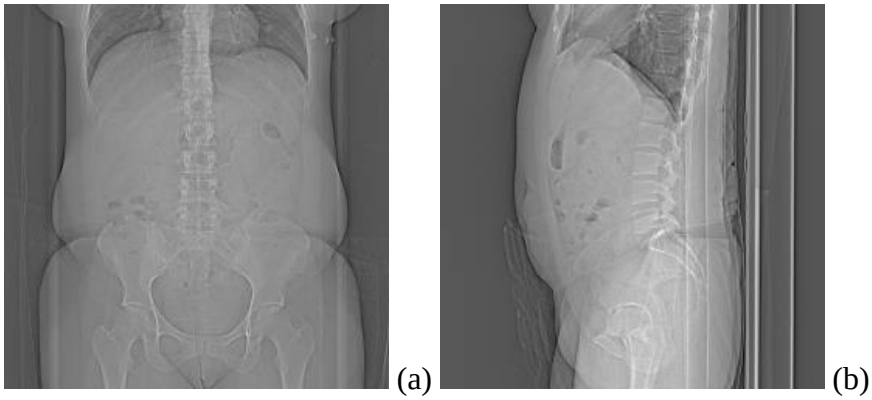


Figure 1.1. The AP localizer (a) and lateral localizer (b) of a typical abdominal CT exam.

Machine learning techniques, especially deep convolutional neural networks (CNNs), have been developed so fast recently and have almost revolutionized many fields of computer vision and image processing, including but not limited to, object detection (Liu et al. 2020), motion tracking (Kwok 2019), pose estimation (Mathis et al. 2018), and action recognition (Ji et al. 2013). Machine learning has also been successfully applied to the field of image transformation (Chen et al. 2020, Dong et al. 2014, Zhang et al. 2020). Researchers have demonstrated that it is possible to reconstruct 3D images from one or two 2D projection views (Henzler et al. 2018, Montoya et al. 2019, Shen et al. 2019).

In this study, we investigate the feasibility of generating one abdominal CT localizer from the other acquired orthogonal localizer (i.e., a 2D to 2D image transformation) using a deep learning approach. Although mathematically this transformation is challenging, our experiment results show that the additional information embedded in large datasets and extracted by the training process of our proposed encoder-decoder network could help and make it feasible.

1.2 Material and Methods

1.2.1 Clinical Images and Information

Institutional Review Boards approval was obtained for this HIPAA-compliant retrospective study and the requirement of written informed consent was waived.

To ensure data consistency, we included only abdominal CT exams from one CT scanner (Somatom Definition Flash, Siemens Medical Solutions USA, Inc., Malvern, PA, USA) in this study. An initial search in the electronic medical records system (Epic Systems Corporation, Verona, WI, USA) of a major hospital in a metropolitan area in west U.S. for all the adult (age $\geqslant$ 18) abdominal CT exams performed on this scanner between April 1, 2013 and June 4, 2020 returned a total of 29,567 exams.

We downloaded the images of the corresponding exams directly from the picture archiving and communication system (PACS). We then recorded DICOM (digital imaging and communications in medicine) header information, including table height and image resolution.

1.2.1.1 Data Screening

The data screening process is illustrated in Figure 1.2. First, we examined the images to make sure two orthogonal localizers exist for each exam. Exams that do not contain images or localizers, contain only one localizer, or contain more than two localizer images were excluded from this study. A total of 13,805 exams were excluded.

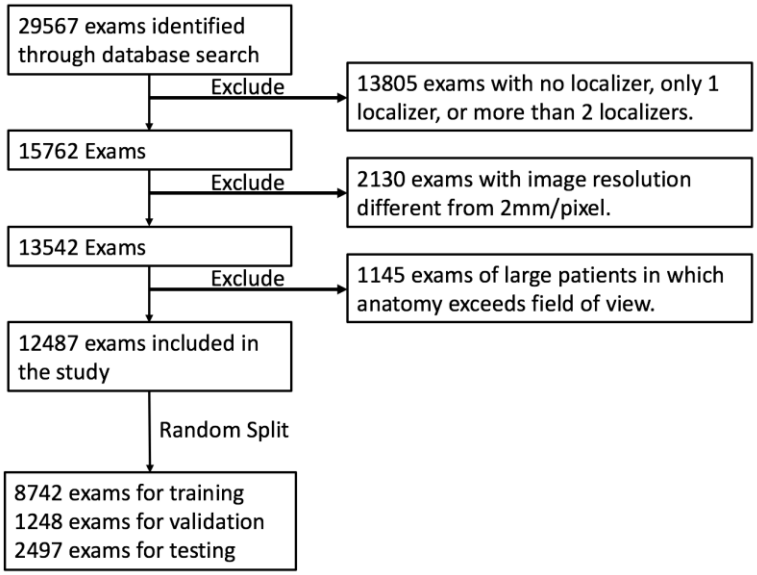


Figure 1.2 Data collection, screening, and splitting in the study.

Since most of the localizer images have the resolution of 2mm/pixel in both horizontal and vertical directions, we excluded the images that have a different resolution. 2,130 exams were excluded in this step.

For the purpose of this study, we also excluded large-patient exams in which localizer images do not cover the whole body. Since all of the images have the anatomy aligned in the same direction as shown in Figure 1, we applied a simple rule to exclude these exams: if the five leftmost or five rightmost columns of the images have patient body pixels, we deemed the patient size too big for this study and hence excluded the exam. 1,145 exams were excluded in this step.

The 12,487 remaining exams eventually included in this study are mainly scanned with two kVp settings (100kVp and 120kVp). The protocol is always the same for the lateral and AP scans. The detailed protocol and reported dose (CTDI) are listed in Table 1.1.

kVp	Number of Exams	Tube Current (mA)	Table Speed (mm/s)	Focal Spot (mm)	CTDI (reported, mGy)
100	224	35	100	0.7	0.085
120	12,263	35	100	0.7	0.140

Table 1.1. Protocols of Exams

1.2.1.2 Data Preprocessing

The screened dataset includes 12,487 remaining exams and each exam has two orthogonal localizer images of the same patient, i.e., AP and lateral images, in the format of DICOM. We pre-processed the exams in the following steps.

Within each exam, the localizer images were first aligned at the y direction of the image coordinate system based on image position from the DICOM header. The localizer image with higher z-coordinate value of image position was shifted up at the y direction to match its corresponding localizer counterpart. The aligned AP and lateral localizer image pair was then cropped to make sure the two images have the same effective scan length and matching location. The cropped image pair was then squared through zero-padding or cropping out the image length

to match the image width. After the cropping and squaring, all images are of 276×276 pixels (height×width).

Since the range of DICOM intensity varies, to facilitate model training, we rescaled the images to a standard range of 0-255 (to be stored in one byte) with calculated minimum and maximum intensity values. Specifically, the rescaling was taken using equation (1.1), where f_{min} and f_{max} are the means of the minimum intensity values and maximum intensity values across all images, respectively.

$$g(x,y) = 255 \times \frac{f(x,y)-f_{min}}{f_{max}-f_{min}} \tag{1.1}$$

The occasional out-of-range intensity values were set to their closest boundary values, i.e., 0 or 255. The scaled images were then down-sampled to 256×256 pixels and saved as gray-scaled PNG images. The PNG images were then used as inputs to the proposed deep learning model.

1.2.2 Proposed Deep Learning Model

We propose a deep learning model, a modified encoder-decoder network, to learn the mapping between the localizer images from one orientation to the other. Denote A as a set of image pairs $\{(X_i^T, Y_i^T), (X_j^V, Y_j^V)\}$, where $(X_i^T, Y_i^T), (X_j^V, Y_j^V) \in R^{m\times \mathrm{n}}$ are localizer pairs of i and j, such that $1 \le i \le K$, $1 \le j \le L$, K is the total number of training exams and L is the total number of validation exams, and m and n are the image height and width, respectively. Given an input set A, the goal is to train the model to find an optimal mapping $\mathcal{F}$, such that

$$\mathcal{F} = \underset{F}{\operatorname{argmin}}\{\mathcal{L}\left(F\left(X_j^V, (X_i^T, Y_i^T)\right), Y_j^V\right)\}, 1 \le i \le K, 1 \le j \le L, \tag{1.2}$$

where $\mathcal{L}$ is the loss function. We formulate the mapping as a composition of three sub functions to be learned by the model, i.e.,

$$F = E \circ T \circ D. \tag{1.3}$$

E is the encoder function, which maps the 2D image domain to the feature domain, T is the transformation function, which maps the feature representations across localizer orientations within the feature domain, and D is the decoder function mapping the feature domain back to the 2D image domain. Accordingly, the learning model is framed as a deep neural network composed

of three sub modules, i.e., encoder, transformation, and decoder, to jointly learn the mapping functions (Figure 1.3).

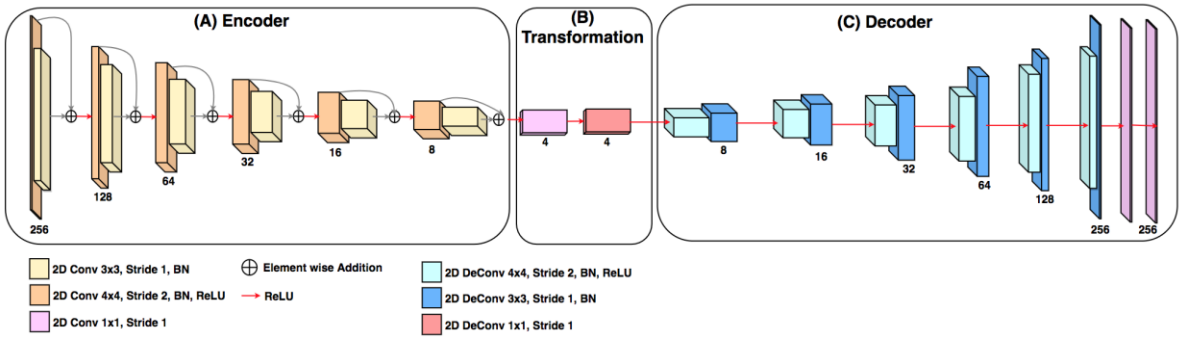


Figure 1.3. Architecture of the proposed deep learning network.

1.2.2.1 Encoder

The encoder module (Figure 1.3(A)) learns a high-level semantic representation of a 2D image object in a sequence of down-sampling encoder blocks. We apply the classical encoder network (Badrinarayanan et al. 2017) and make a few adjustments according to the characteristics of our focal task. A classical encoder uses a max pooling layer to reduce the feature map's resolution, but it may lose some pixel-level information which could be essential to a dense prediction problem. We remove the max pooling layers because the goal of the network is to predict at the pixel level (Gao et al. 2019). We then add residual connection to facilitate the training of the deep network (He et al. 2016).

The encoder module consists of six encoder blocks. Each encoder block first starts with a 2D convolutional layer and batch normalization using a 4×4 kernel with sliding stride 2×2. This operation down-samples the spatial size by a factor 2 and goes through a ReLU activation to generate the first-layer output. Next is a 2D convolutional layer and batch normalization with a kernel size of 3×3 and sliding stride 1×1, keeping the spatial size unchanged. The output of the second convolutional layer is linked to that of the first layer by an element-wise addition before passing through the ReLU activation. The channel size of the feature map increases as the encoder block goes deeper. Our input image is of size 1×256×256 (channel size × image height × image

width) and the filter size of the first encoder block is 128. The filter size doubles as the model goes one step deeper, thereby the feature map dimensionality follows the sequence of changes:

1×256×256→128×128×128→256×64×64→512×32×32→1024×16×16→2048×8×8→4096×4×4

1.2.2.2 Transformation

The transformation module (Figure 1.3(B)) converts the feature representation from one localizer orientation to the other. Because the converted representation stays in the same feature domain, we construct the module to include two blocks, i.e., a 1x1 convolutional layer followed by a ReLU activation and a 1x1 deconvolutional layer followed by a ReLU activation. The 1x1 convolutional and deconvolutional operations are coordinate-dependent linear transformations in the filter space. They are immediately followed by a non-linear ReLU activation. The transformation maintains the filter size of the feature map so that the output dimensionality does not change (i.e., 4096×4×4).

1.2.2.3 Decoder

The decoder module (Figure 1.3(C)) generates a new image from the transformed feature representation. To match the max pooling operation, a classical decoder block uses a max un-pooling layer to enlarge the output feature map. We remove the un-pooling layers from the decoder module due to the adjustment in the encoder module. The adjusted decoder module consists of six deconvolutional blocks, a 1x1 convolutional layer followed by a ReLU activation, and a final 1x1 convolutional layer. Each deconvolution block first starts with a 2D deconvolutional layer and batch normalization using a 4×4 kernel with sliding stride 2×2. This operation up-samples the spatial size by a factor 2 and goes through a ReLU activation. Next is a 2D deconvolutional layer and batch normalization with kernel size of 3×3 and sliding stride 1×1, maintaining the same spatial size. The output is then followed by a ReLU activation. The channel size of the feature map decreases as the deconvolution block goes deeper. After the deconvolution blocks, the output connects a 1×1 convolutional layer and a ReLU activation, reducing the dimensionality in the filter space to one. The last layer of the decoder module is a 1×1 convolutional layer, which learns a linear map from the up-sampled image to the final output image. The input matrix to the decoder module is of size 4096×4×4 (channel size × feature height × feature width) and the filter size of the first decoder block is 4096. The filter size cuts in half as the model goes one step deeper, thereby the dimensionality follows the sequence of changes:

4096×4×4→2048×8×8→1024×16×16→512×32×32→256×64×64→128×128×128→64×256×256→1×256×256

1.2.2.4 Loss Function

We propose a novel loss function for the focal task. The proposed loss function is named *scaled mixture loss* as it consists of three major components: an ℓ_2 loss, a structural similarity (SSIM) loss, and a scaling parameter. The ℓ_2 loss (or MSE, the mean squared error) captures an overall prediction error by averaging the squares of pixel-wise prediction errors. Specifically, the MSE loss is defined as follows.

$$MSE(f, \hat{f}) = \frac{1}{m * n} \sum_{i=1}^{m} \sum_{j=1}^{n} (f(i,j) - \hat{f}(i,j))^2 \quad , \tag{1.4}$$

where m and n are the image height and width, and f and $\hat{f}$ are the intensity matrices of the ground-truth image and predicted image, respectively.

The MSE loss does not consider image structures and tends to produce images that are overly smooth and blurry (Seif et al. 2018). To account for this issue, we introduce the second component, the SSIM loss, which is expected to help the network produce sharper and human-perceivable images. Specifically, the SSIM loss is defined as follows.

$$SSIM\_LOSS(f, \hat{f}) = 1 - \frac{(2\mu_f \mu_{\hat{f}} + c_1)(2\sigma_{f\hat{f}} + c_2)}{(\mu_f^2 + \mu_{\hat{f}}^2 + c_1)(\sigma_f^2 + \sigma_{\hat{f}}^2 + c_2)}, \tag{1.5}$$

where μ_f and $\mu_{\hat{f}}$ are the averages of f and $\hat{f}$, σ_f and $\sigma_{\hat{f}}$ are the variances of f and $\hat{f}$, and c_1 and c_2 are used to stabilize the weak denominator.

In addition, we include a third component, the scaling parameter, to our focal task. The image intensity values are commonly processed (scaled and normalized) to have the magnitude between 0 and 1 before being inputted to a deep neural network. Considering the depth of our proposed network (Figure 1.3), as well as the magnitude of the mean squared error loss, the notorious *gradient vanishing* issue is likely to occur and hinder the learning at a more granular level. Intuitively, the scaling parameter shifts the intensity value to a new space, increases the gradient of the MSE loss and allows it to stay at a certain magnitude when the error gets smaller, thus alleviating the gradient vanishing at an early stage and allowing the network to continue to improve

its accuracy. The output intensity values are then shifted back to its original scale at testing time. Specifically, the proposed scaled mixture loss is defined as follows.

$$\mathcal{L} = \alpha \cdot MSE\left(r \cdot f, r \cdot \hat{f}\right) + r^2(1-\alpha)SSIM\_LOSS(r \cdot f, r \cdot \hat{f}), \quad (1.6)$$

where α is the weight parameter and r is the scaling parameter.

1.2.2.5 Ablation Experiments

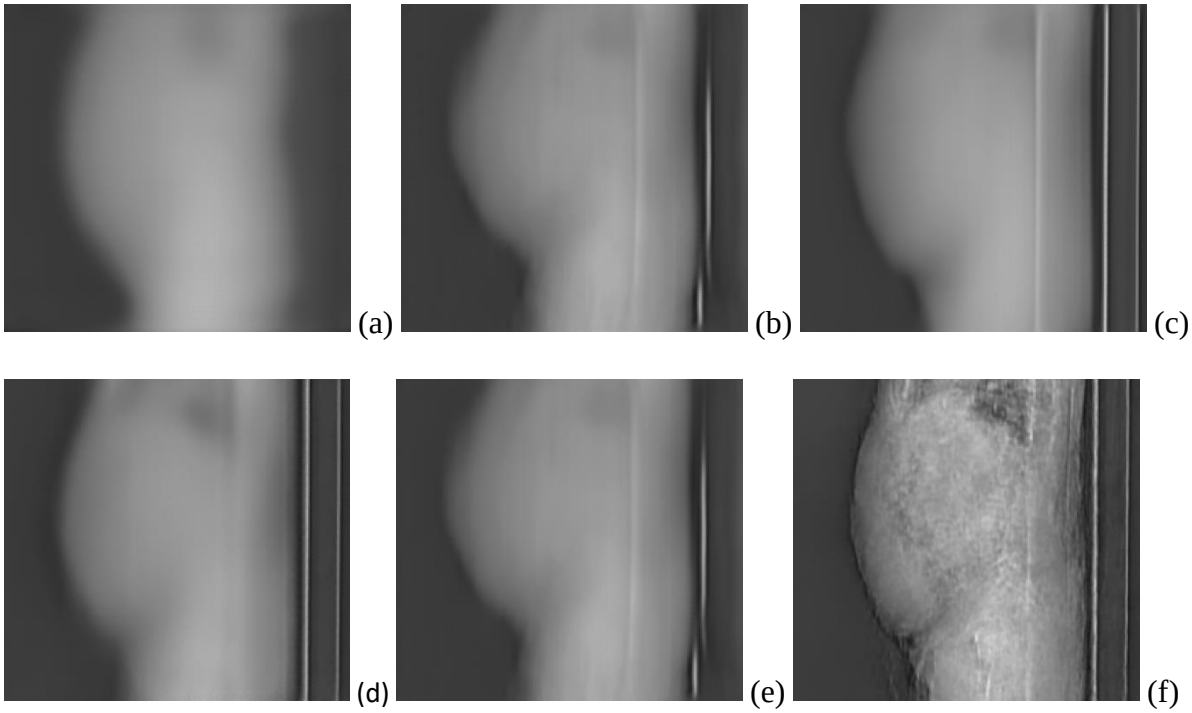


Figure 1.4. Predicted images at 200-epoch with subsampled training set of 615 localizer pairs ($\boldsymbol{r}$ = 1000). (a) MSE loss. (b) SSIM loss. (c) MSE+SSIM loss. (d) Scaled MSE loss. (e) Scaled SSIM loss. (f) Scaled mixture loss.

With a randomly subsampled dataset (with 615 localizer image pairs), we first conducted an ablation experiment on the selection of loss function using the proposed encoder-decoder network. Specifically, we evaluated six loss function options: MSE, SSIM, MSE+SSIM, Scaled MSE, Scaled SSIM, and Scaled Mixture (proposed). The results are summarized in Appendix B, and Figure 4 shows one example. The proposed scaled mixture loss function (Figure 1.4f) achieved the best overall performance and much better perceived image quality than the other loss functions.

Similarly, with the chosen loss function, we conducted another ablation experiment to compare the proposed encoder-decoder network (6-block encoder-decoder network) with five alternatives, i.e., (1) 4-block encoder-decoder network, (2) 5-block encoder-decoder network, (3) 4-block encoder-decoder network with skip-connection, (4) 5-block encoder-decoder network with skip-connection, and (5) 6-block encoder-decoder network with a max pooling layer. The results (summarized in Appendix C) show that the proposed network achieved the best overall performance.

1.2.3 Implementation and Evaluation

We implemented the proposed deep neural network with the Pytorch library. In our evaluation experiment, we used an Amazon EC2 instance, which contains 8 Intel Xeon CPUs (2.50GHz) and a NVIDIA Tesla T4 GPU with 12 GB memory, to train the model.

The initial learning rate was set to 0.00001. If the loss stopped decreasing for more than 50 epochs, the learning rate was reduced by a factor of 0.2. Adam was used as the optimizer and weight decay was set to 0.0001. We followed (Zhao et al. 2017) and empirically set $\alpha = 0.16$. The scaling parameter r was set to 1000. The training terminated if, after 200 epochs, the loss stopped decreasing for over 80 epochs.

We randomly split the original 12,487 exams (from 10,553 patients) into training, validation, and test datasets in the ratio of 7:1:2. The three datasets were disjoint at the patient level; no exams from the same patient were included in different datasets (training, validation, or test) simultaneously. Demographics of the patient population are listed in Table 1.2. We trained two separate models, one model to predict the AP localizer based on the lateral localizer and the other to predict the lateral localizer based on the AP localizer.

Dataset	Number of Patients	Male	Female	Mean Age	Median Age	Range of Age [Min, Max]
Training	7,404	3,120	4,282	48.83	48	[18,120*]
Validation	1,045	448	597	48.00	47	[18,120*]
Test	2,104	883	1,221	49.02	48	[18,119*]
All	10,553	4,451	6,100	48.79	48	[18,120*]

Note: Patient age is calculated based on the date of the first exam if multiple exams are included.
*Certain patients' age may not be accurate as the information is extracted from DICOM header, and demographic information of some emergency patients may be missing or incorrect.

Table 1.2. Demographics of the Patients

We evaluated the performance of the reconstruction using three metrics, i.e., location accuracy, profile accuracy, and attenuation accuracy. All metrics are based on comparisons between information extracted from the predicted images and that from the actual images in the test dataset, and these three metrics were chosen according to clinical needs. Location accuracy was chosen because this information is used clinically to determine if patients are correctly positioned or centered. Profile accuracy was chosen because this information is used to estimate the size of the patient (Boone et al. 2011). Attenuation accuracy was chosen because this information is used to calculate diagnostic scan parameters (more specifically, the "mAs" in tube current modulation). These three metrics collectively provide an insightful and comprehensive evaluation of the quality of the predicted localizer images.

1.2.3.1 Location Prediction Error

Location accuracy (or inversely, error) is important because we want the predicted images to accurately reflect the location of the patient, both in the AP direction and in the lateral direction. This information is essential for the quality control and the determination of the parameters for the following diagnostic CT scan.

1.2.3.2 Lateral Location Prediction Error

The lateral location can be represented by the table height in the DICOM header. The DICOM table height refers to the distance (in millimeters) from the top of the patient table to the center of rotation. Therefore, we extract the ground-truth table height from the lateral DICOM file.

The predicted table height is estimated by detecting the table position in the horizontal direction from the predicted lateral image (pseudo code for detecting the table position in a lateral localizer image is presented in the Appendix A). The actual table height in the ground-truth lateral image is estimated by the same detection technique.

To assess the accuracy of the detection algorithm, we compute the Pearson correlation between the estimated actual table height and the ground-truth table height and use a scatter plot to examine the fitted line. We then measure the lateral prediction error by taking the absolute difference between the predicted table height and the actual table height (scaled from number of pixels to millimeters).

1.2.3.3 AP Location Prediction Error

The AP location can be represented by the patient center position in the horizontal direction. Since there is no ground-truth information for the patient center position, we apply the following steps to estimate the predicted patient center position and actual patient center position from the predicted AP image and ground-truth AP image, respectively.

We first extract the patient profile (left boundary and right boundary) based on a heuristic patient boundary detection algorithm (pseudo code for extracting patient profile in an AP image is presented in the Appendix A). Denote the detected patient profile in the 256-dimensional vector space as follows.

$$BDLT = \{bdlt_i\}, 0 \leq i \leq 255, \tag{7}$$

$$BDRT = \{bdrt_i\}, 0 \leq i \leq 255, \tag{8}$$

where $bdlt_i$ is the pixel index of the patient's left boundary at image row i and $bdrt_i$ is the pixel index of the patient's right boundary at row i.

A middle point between the left boundary and the right boundary is calculated at each row, and the patient center point is estimated by taking the most frequent value among all middle points. It can be written as follows.

$$AP\_LOC = r \cdot \underset{v_i}{\operatorname{argmax}}\{freq(v_i = \frac{bdlt_i + bdrt_i}{2}) | 0 \leq i \leq 255\}, \tag{9}$$

where r is the scaling parameter (from number of pixels to millimeters).

Finally, we measure the AP location prediction error by taking the absolute difference between the predicted patient center position and actual patient center position.

1.2.3.4 Profile Prediction Error

For both the AP and lateral images, patient profile is calculated based on the patient boundary, and the profile value is defined as follows.

$$BDS = \{bds_i \mid bds_i = r(bdrt_i - bdlt_i)\}, 0 \leq i \leq 255. \tag{1.10}$$

bds_i is the number of pixels between the patient's left and right boundaries at row i, and r is the scaling parameter (millimeters per pixel).

Patient profile is calculated for both the AP and lateral directions. For each direction, the profile of the predicted image is compared with that of the ground-truth image at each row to acquire the absolute percentage difference. The mean absolute percentage difference (MAPD) is used to measure the profile prediction error.

1.2.3.5 Attenuation Prediction Error

Attenuation refers as the percentage of x-ray reduction after it penetrates the human body. Since in CT images, the attenuation is linearly correlated with the pixel intensity, the patient attenuation is defined as follows.

$$BDV = \{bdv_i \mid bdv_i = \sum_{j=bdlt_i}^{bdrt_i} f(i,j)\}, 0 \le i \le 255 \quad . \tag{1.11}$$

bdv_i is the sum of the intensity values of the pixels between the patient's left and right boundaries at row i.

Similar to patient profile, patient attenuation is computed at both directions. For each direction, the patient attenuation of the predicted image is compared with that of the ground-truth image at each row to acquire the absolute percentage difference. The mean absolute percentage difference (MAPD) is used to measure the attenuation prediction error.

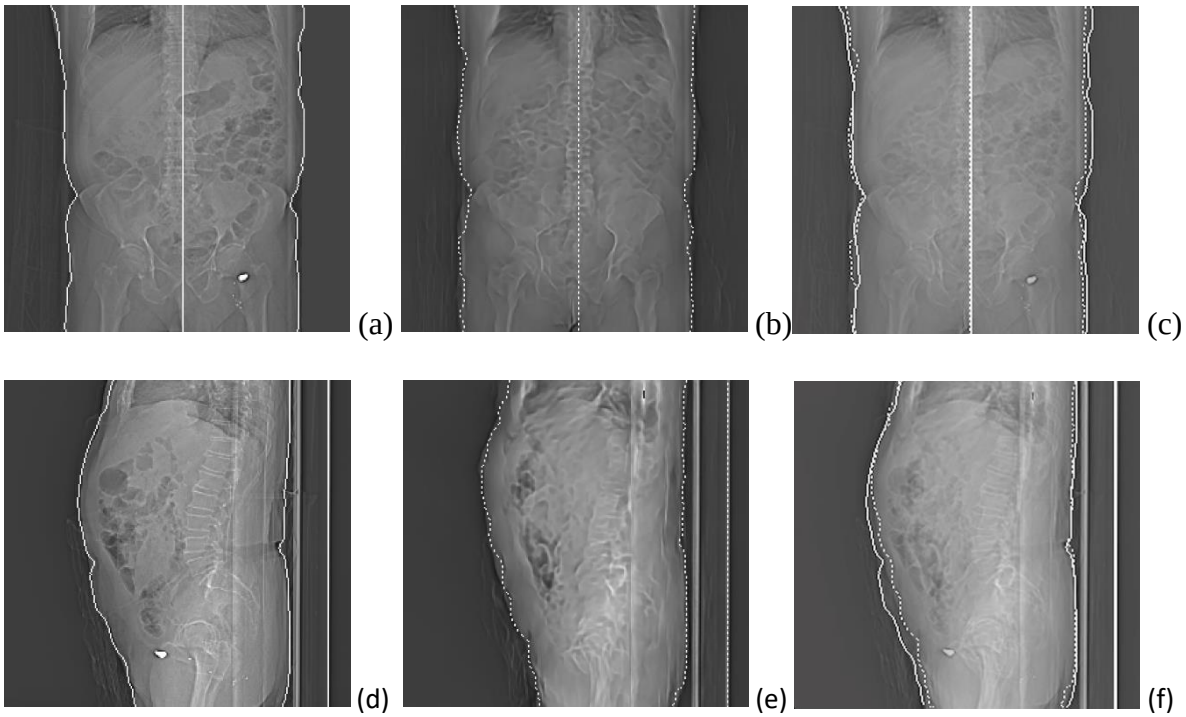
1.3 Results

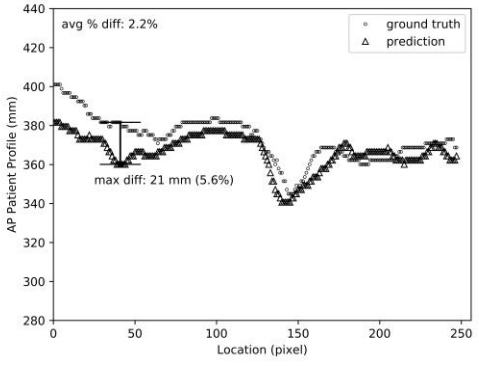
Figure 1.5 shows an example of model prediction, location calculation, patient profile, and attenuation. Figure 1.5(a) shows the actual AP localizer image, and Figure 1.5(b) shows the predicted AP localizer image. For both images, the calculated patient position (center line) and patient boundary are indicated. Figure 1.5(c) gives the overlap (between the actual and predicted images) display of center line, patient boundary, and patient body. Similarly, Figure 1.5(d) and (e) show the actual and predicted lateral localizers, with the table location and the patient boundary indicated, respectively, and Figure 1.5(f) gives the overlap display between the actual and predicted images.

Figure 1.5(g) and (h) contrast the ground truth and prediction in patient profile for the AP and lateral predictions, respectively. For this particular patient, the AP prediction has a profile

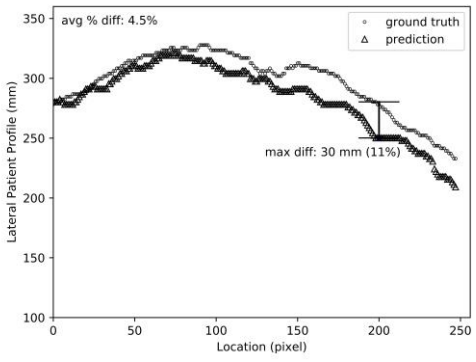
prediction error averaged at 4.5% with a maximum of 11%. The lateral prediction has an average error of 2.2% and maximum of 5.6%.

Figure 1.5(i) and (j) contrast the ground truth and prediction in attenuation for the AP and lateral predictions, respectively. For this particular patient, the AP prediction has an attenuation prediction error averaged at 4.9% with a maximum of 10.9%. The lateral prediction has an average error of 9.1% and maximum of 14.4%.

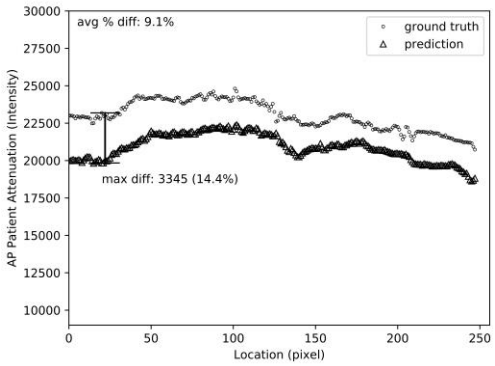




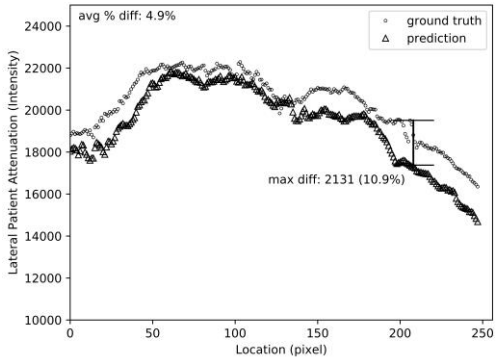
(g)



(h)



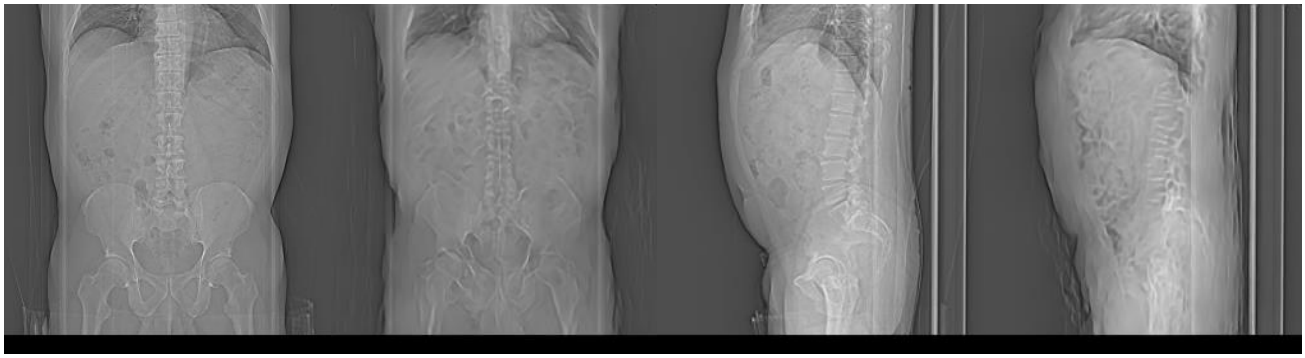
(i)



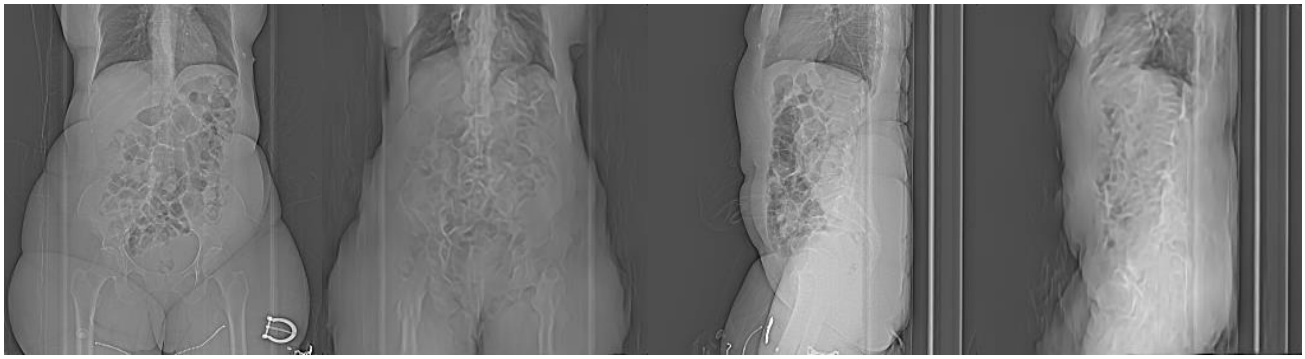
(j)

Figure 1.5. Example of model prediction, location calculation, patient profile, and attenuation. (a) The actual AP localizer image, with the estimated patient position (center line) and patient boundary indicated. (b) The predicted AP localizer image. (c) Overlap display of (a) and (b). (d) The actual lateral localizer image, with the estimated table location and patient boundary indicated. (e) The predicted lateral localizer image. (f) Overlap display of (d) and (e). (g) The ground truth vs. prediction in patient profile for the AP prediction. (h) The ground truth vs. prediction in patient profile for the lateral prediction. (i) The ground truth vs. prediction in attenuation for the AP prediction. (j) The ground truth vs. prediction in attenuation for the lateral prediction.

Two more examples of model prediction are presented in Figure 1.6.



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 1.1.6. Two additional examples of model prediction. (a) and (e), (b) and (f), (c) and (g), (d) and (h) are actual AP images, predicted AP images, actual lateral images, and predicted lateral images, respectively.

Table 1.3 summarizes the results of the location prediction error on the test dataset. The results look very promising, with an average prediction error of 1.02±3.37mm in the lateral direction and 6.46±6.43mm in the AP direction.

Orientation	Mean (mm)	Std Dev(mm)	Median (mm)	Max (mm)	<2mm (%)	<5mm (%)	<15 mm (%)	<20 mm (%)
Lateral	1.02	3.37	0.00	53.91	66.79	98.89	99.11	99.24
AP	6.46	6.43	4.31	62.53	12.78	54.63	89..51	96.44

Table 1.3 Location Prediction Error

Figure 1.7 visualizes the table position detection accuracy and location prediction error results. Figure 1.7 (a) contrasts the true table height (extracted from DICOM header) and those estimated based on the actual and predicted images, showing that: 1. The Pearson correlation between the ground-truth and actual table heights is 0.9986, and the fitted line is straight with no significant outliers, indicating that the detection algorithm is accurate. 2. Most of the predictions for lateral location were accurate. Figure 1.7(b) and (c) show the histogram distributions of the absolute location difference between the ground-truth images and predicted images for lateral and AP predictions, respectively. Our model achieved very high accuracy in lateral prediction with over 98% of the cases having < 2mm (1 pixel) errors. AP prediction was relatively less accurate but still reasonable with over 96% of the cases having < 2cm (10 pixels) errors. The lack of an apparent reference mark (the table) could be the reason that the AP prediction is less accurate than the lateral prediction.

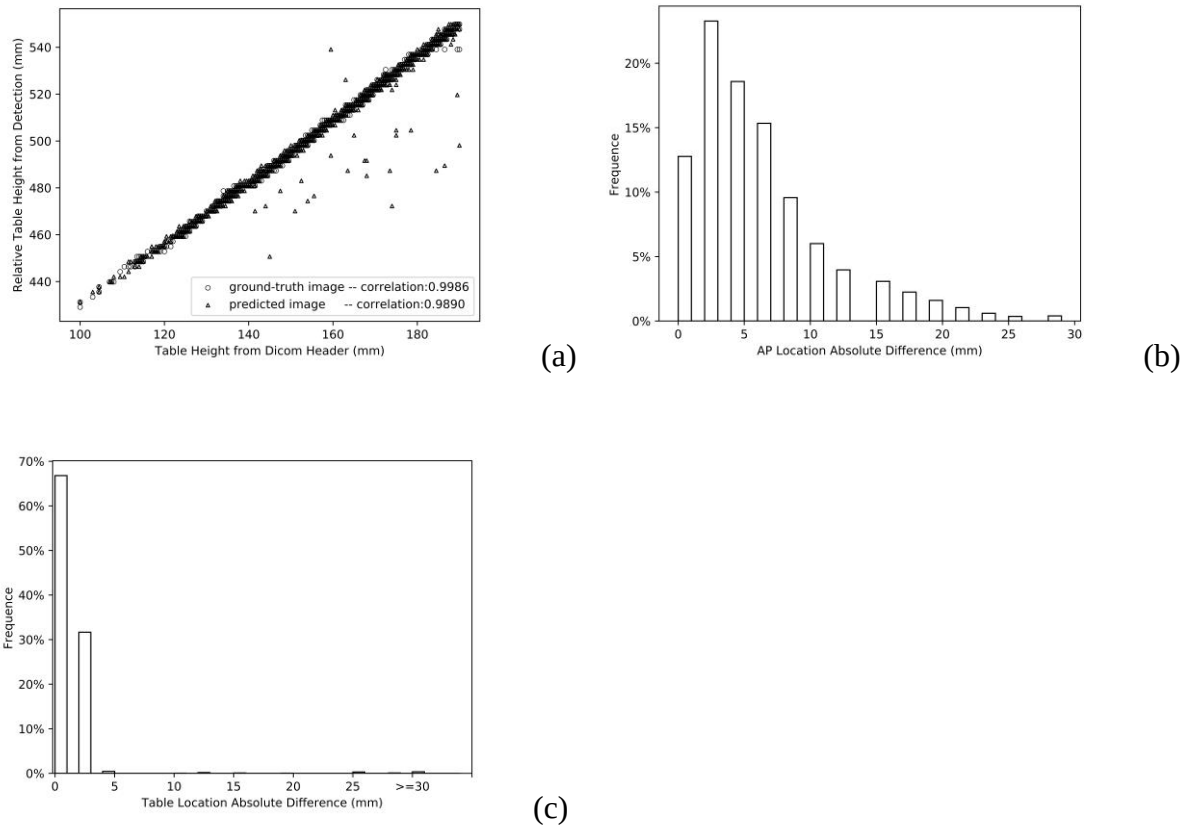


Figure 1.7. Location prediction results. (a) The true table height (extracted from DICOM Header) vs. the table heights estimated based on the actual and predicted images. (b) The histogram distribution of the location difference between the ground-truth images and predicted images for lateral prediction. (c) The histogram distribution of the location difference between the ground-truth images and predicted images for AP prediction.

Table 1.4 summarizes the results of the profile prediction error on the test dataset. The results also look very promising, with an MAPD of 4.43±2.02% in the lateral direction and 3.90±2.32% in the AP direction.

Orient	Mean (%)	StdDev (%)	Median (%)	Max (%)	<5% (%)	<10 % (%)
Lateral	4.43	2.02	3.98	20.78	69.88	98.20
AP	3.90	2.32	3.42	32.75	79.66	98.32

Table 1.4. Profile Prediction Error (MAPD)

Figure 1.8 visualizes the profile prediction error results. The histogram distributions of the average error for all test data are plotted in Figure 1.7(a) and (b) for the AP prediction and lateral prediction, respectively.

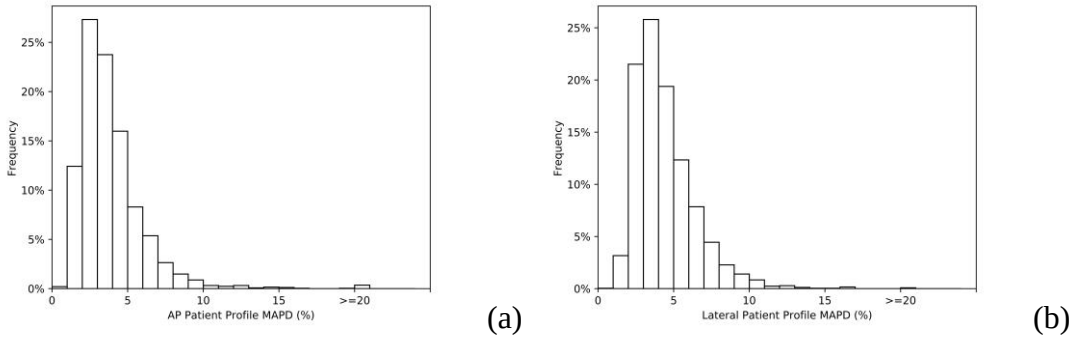


Figure 1.1.8. Histogram of patient profile mean absolute percentage difference (MAPD). (a) AP. (b) Lateral.

Table 1.5 summarizes the results of the attenuation prediction error on the test dataset. The results again look very promising, with an MAPD of 6.20±2.94% in the lateral direction and 7.12±3.54% in the AP direction.

Orient	Mean (%)	StdDev (%)	Median (%)	Max (%)	<5% (%)	<10% (%)
Lateral	6.20	2.94	5.51	27.42	41.05	89.55
AP	7.12	3.54	6.58	31.80	30.16	82.98

Table 1.5. Attenuation Prediction Error (MAPD)

Figure 1.9 visualizes the attenuation prediction error results. The histogram distributions of the average error for all test data are plotted in Figure 1.8(a) and (b) for the AP prediction and lateral prediction, respectively.

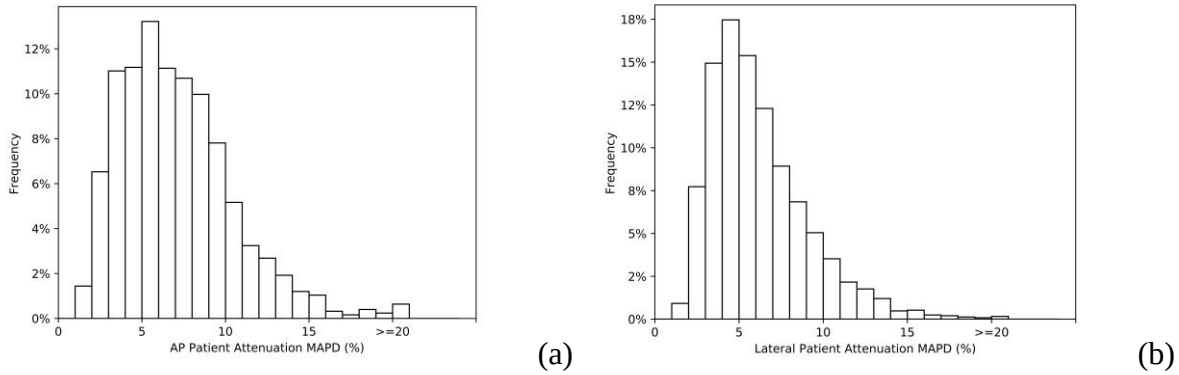


Figure 1.1.9. Histogram of patient attenuation mean absolute percentage difference (MAPD). (a) AP. (b) Lateral.

1.4 Discussion

CT is one of the most, if not the most, important imaging modalities in healthcare. Researchers and developers constantly strive to improve image quality, shorten scan time, and lower patient dose. Knowledge of the patient position, anatomy localization, and attenuation distribution is very important to carry out a successful diagnostic scan, and that is the reason why it is currently still a common clinical practice to perform two localizer scans in advance, even though these two scans only contribute to patient dose but do not provide diagnostic information. Reducing the dual localizer scans to a single localizer scan will lower patient dose, and at the same time, improve workflow and clinical efficiency.

Research on image transformation in medical imaging could be roughly divided into two directions: image transformation in the contrast domain and that in the geometry domain. In the contrast domain (also called image generation or image synthesis), researchers have demonstrated the feasibility of transforming MR images into CT images (Wolterink et al. 2017, Xiang et al. 2018, Han et al. 2017), CT images into MR images (Jin et al. 2019, Zhao et al. 2017), MR images into X-ray images (Stimpel et al. 2019), and so on, using deep CNN-based frameworks, such as GAN. The contrast or the appearance of the images is changed to meet various clinical needs, but the location and orientation of the images stay the same. In another direction of image transformation research (including this work), researchers transform images to different location and/or orientation but keep the contrast the same. Henzler et al. and Pradhan et al. demonstrated the feasibility of predicting 3D object based on single 2D x-ray image using a deep CNN-based network, but their research focused on high-intensity bone-only objects. Montoya et al. demonstrated that a volume 3D CT localizer could be predicted from two orthogonal 2D localizers, with a deep CNN-based network. Shen et al suggested that one could predict 3D)volume CT data from a single 2D view using an encoder-decoder architecture. Their work requires multiphase 3D CT of the same patient to train the network and is thus less applicable in the day-to-day diagnostic CT workflow, but its success made the encoder-decoder architecture very promising in solving image transformation problems. In this work, we have proposed a modified encoder-decoder network that can directly make 2D X-ray projection (localizer) prediction of CT images based on

an orthogonal projection in a clinical diagnostic CT setting, thereby providing the first solution to this particular clinical problem.

Mathematically, predicting one projection view of a 3D object (patient) from another orthogonal projection view is very challenging if not impossible. Spatial information needed in the desired view is overlapped on the given orthogonal view, and without additional information, it is impossible to separate them. Fortunately, for this particular problem, there are three unique characteristics that we can leverage. First, our objects (i.e., patients) have very similar internal structures. The similarity in human anatomy makes it possible for the information to be extracted effectively from a large set of training examples. Second, the fast development in deep learning techniques now offers the opportunity to learn the information from large amount of data and make reliable predictions. Finally, since the localizer images are not used for actual diagnosis, there is no need to have all the details precisely predicted. As long as the predicted image can roughly represent the anatomy and accurately reflect the patient location, size (profile), and attenuation, for many clinical applications, it is good enough and could be used to replace the actual localizer scan. It should also be mentioned that although the predicted image has adequate perceived image similarity with the actual acquired image, the possibility of missing anatomy details and actual disease evidence is still high. In certain applications (such as perfusion or cardiac) where the technologists need more anatomy details to determine the scan range, they always have the option to use an acquired, instead of predicted, localizer.

We have developed a modified encoder-decoder network for this CT image transformation task. We adopt the encoder module as the starting point for representation learning, as the convolutional encoder module has been widely used in medical imaging segmentation (Ronneberger et al. 2015, Wang et al. 2021) and image synthesis (Gao et al. 2019) tasks and achieved great success. We remove the max pooling layers to reduce the loss in pixel information and introduce the residual connection to overcome the gradient vanishing and exploding issues in training. We design a transformation module to learn the feature mapping across the localizer 2D domains. Finally, we use the decoder module to generate the predicted image in the target domain.

Our experiment results appeared to be very promising. For location accuracy, most (98%) of our lateral predictions had an error within 5mm or 2 pixels. Lateral positioning is very important clinically as the perceived size of the patient directly affects the patient dose and image quality.

Our location prediction in the AP direction was relatively less accurate but still reasonable. The profile error and attenuation error of our predictions were all in the range of 3-7%. Given that the predicted images and the actual images in general correlate well in terms of profile and attenuation estimations (as shown in Figures 8 and 9), we are confident that the tube current modulation, which is the algorithm for CT scanners to adjust the strength of x-ray based on the attenuation, could work well on our predicted images.

There are several limitations in this study. First, in order to get a quality prediction, our selected input/output image size is quite high (256×256) and the model used is quite deep. As a result, training requires a large memory and takes a long time. As a matter of fact, we needed to train on a low-resolution network to tune the hyper parameters first before we could start the actual 15-days-long training. Second, as can be seen in Figure 1.4, the details of the anatomy are not completely predicted. If the predicted image is used to determine the scan range, some organs, such as the diaphragm, are harder to distinguish, compared to other high-contrast organs, such as the lungs. Third, since the model is trained largely based on the images of "regular" patients, prediction may fail in "irregular" situations. Figure 1.10 illustrates two "failed" cases, including the existence of foreign objects (Figure 1.10 (a) and (c) for the actual and predicted images, respectively) and irregular patient positioning (hands down in this case, Figure 1.10 (b) and (d) for the actual and predicted images, respectively). Finally, since our evaluation was based on a single scanner and a single anatomy, the robustness of the model in other contexts with different scanners and anatomies needs to be further validated in future research. However, even with these limitations, our results show great potential in the clinical applicability of our proposed technique. The predicted CT localizer could be displayed right after the acquired localizer, and if the prediction does not meet the expectation, the technologist could immediately start the second localizer. Even if only some of the dual localizer scans are replaced by single localizer scans, the clinical benefits may still be significant given the volume of scans that need to be done.

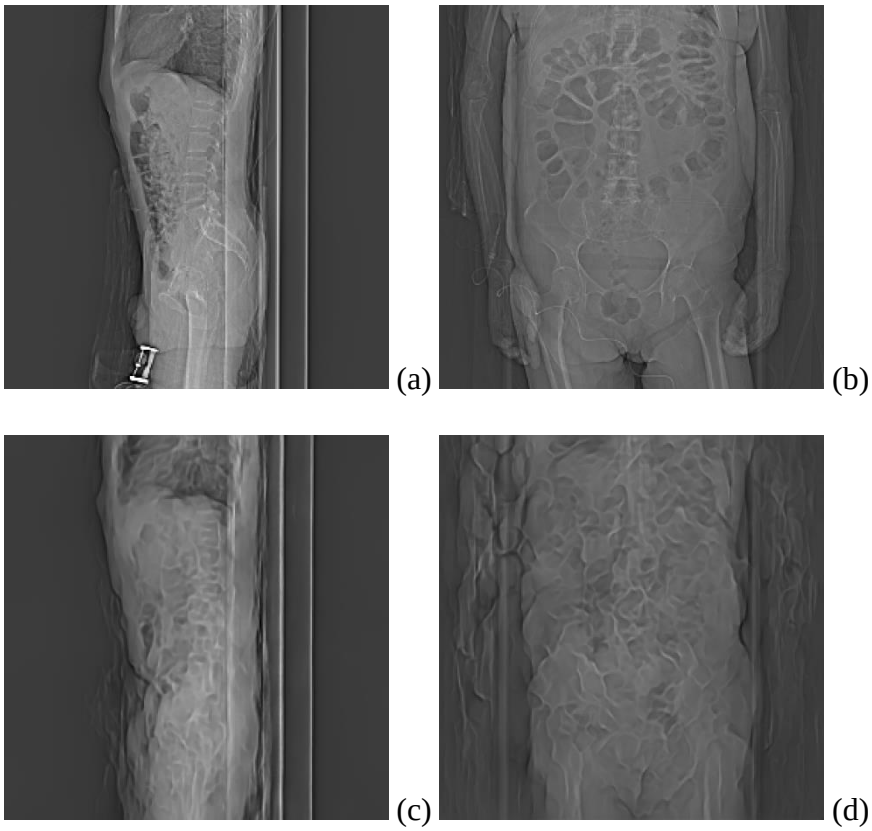


Figure 1.1.10. Examples of "irregular" situations. (a) The ground-truth image of a patient with foreign object. (b) The ground-truth image of a patient scanning with irregular position (hands down). (c) The model prediction for (a). (d) The model prediction for (b).

Our work could be extended or improved in several directions in future research. First, from the network structure perspective, other networks, e.g., UNet (Ronneberger et al. 2015) and the generative adversarial network (GAN) (Goodfellow et al. 2015), could be integrated with our current encoder-decoder network. The UNet architecture allows to connect the high-resolution features from contracting path to the up-sampled outputs, and the larger number of feature channels and the skip connection feature might help improve the prediction accuracy. GAN has achieved superior performance in many image generation tasks (Marzullo et al. 2020, Armanious et al. 2020) and the network structure could potentially greatly improve the current prediction accuracy (Liu et al. 2019). Second, from the model robustness perspective, future research may conduct more comprehensive experiments with a wide coverage of patient sizes, positions, implants, external objects (tubes, lines, patient covers, etc.), CT scanner types, and institutions. Data augmentation operations, such as patient (or organ) position shifting and rotation, could also be considered to

increase the sample size and account for abnormally positioned patients. Third, as this is a retrospective study based on historical data, we could not insert the predicted image back to the CT system and generate the tube current modulation curve for the following spiral scan to compare the tube current with actual and predicted attenuation maps. Researchers may collaborate with major CT manufactures to implement such evaluation metrics in future studies. Finally, the generalizability of our approach can be tested by training and testing separate models for other anatomies, such as the head/neck area and chest/cardiac area.

1.5 Conclusion

We have demonstrated the feasibility of using a deep learning model to predict one CT localizer image based on the other orthogonal acquired localizer image with reasonable accuracy. This model could be used to clinically reduce the patient dose and simplify the workflow.

1.6 Acknowledgements

This work was supported by the Radiology Pilot Grant from the Department of Radiology, School of Medicine, University of Colorado. We would also like to thank Amazon's EC2 Hardware Accelerated Instance Free Trial program for providing computational resources.

2. Essay 2: Robust Meta-graph Learning: Exploring Multilevel Feature Interrelationships to Enhance Multimodal Time Series Prediction

2.1 Introduction

The booming digital economy has triggered the proliferation of multimodal data, promoting the seamless integration and use of multiple (e.g., textual, acoustic, and visual) modalities in various applications (Lahat et al. 2015). Empowered by recent advances in artificial intelligence, especially deep learning, multimodal data analysis is now increasingly used in a variety of fields, e.g., multimodal-data-based decision-making in Fintech (Sawhney et al. 2020), inferring multimodal latent topics from electronic health records in healthcare (Y. Li et al. 2020), and intelligent planning with multimodal data in transportation (Farahani et al. 2021). The competence of a multimodal deep learning method, which incorporates multiple data modalities into a unified learning framework, lies in its capability to enhance learning by reducing conflicts while leveraging the complementarity across modalities.

When multimodal data are used in real-world settings, they are often accompanied by dynamic variation patterns of data modalities along the temporal dimension, e.g., a company's financial risk depicted by a series of quarterly accounting reports and earnings calls (Li et al. 2020) and a user's sequential preference modeled by changing demographical and networking characteristics over time (Wu et al. 2019). In such cases, multimodal data analysis is extended to multimodal time series (MTS) analysis, especially MTS prediction (MTSP), an emerging research trend that accommodates temporal and spatial variation patterns[1] and temporal-spatial covariation patterns in prediction (Chambon et al. 2018, Cheng et al. 2022, Tan et al. 2020).

[1] We use the temporal and spatial dimensions to describe the data patterns over time and across features, respectively.

Given the existence of intricate feature interrelationships[2], effective MTSP is non-trivial. While traditional time series studies have either used a single modality or ignored the spatial variation patterns (Brockwell and Davis 2009, Hochreiter and Schmidhuber 1997, Whittle 1951), recent studies have attempted to account for both temporal and spatial variation patterns. Examples are temporal-spatial neural networks (Eldele et al. 2021) and sequential multimodal deep learning methods (Akbari et al. 2021, Zhang et al. 2020). Both types of methods have adopted a two-stage aggregation strategy, i.e., aggregating features in temporal and spatial dimensions at separate stages sequentially. The aggregation process has commonly applied attention mechanisms (Ma et al. 2019, Song et al. 2019) and diverse multimodal deep learning methods (Kosaraju et al. 2019, Wang et al. 2019) to incorporate the intricate feature interrelationships. Importantly, since the temporal and spatial features are aggregated at separate stages, the *temporal-spatial covariations* are not captured. However, this may give rise to such issues as information loss, information redundancy, and information conflict. For instance, in a temporal-spatial neural network, where spatial features are aggregated at the first stage, fine-grained predictive clues may get annihilated due to the ignoration of information in the temporal dimension. Meanwhile, the aggregated spatial features may be redundant and get confounded by the second-stage aggregation. Furthermore, because temporal-spatial covariations are ignored, conflicting temporal and spatial features may not be processed properly, consequently impairing the prediction performance.

Intuitively, mitigating the issues of information loss, information redundancy, and information conflict in MTSP requires effectively disentangling multilevel feature interrelationships within multimodal time series. According to different data granularities and

Specifically, the spatial dimension can include both intra- and cross-modal features.

[2] We use the term "feature interrelationships" to refer to the combination of interrelationships across temporal and spatial dimensions.

dimensions, we categorize multilevel feature interrelationships in MTS into seven types: (1) feature-wise (*F2F*) interrelationships, (2) modality-wise (*M2M*) interrelationships, (3) time-step-wise (*T2T*) interrelationships, pairwise superposed (interaction), i.e., (4) *F2F-M2M* (*FM2FM*), (5) *F2F-T2T* (*FT2FT*), and (6) *M2M-T2T* (*MT2MT*), interrelationships, and finally, (7) ternary superposed (interaction), i.e., *F2F-M2M-T2T* (*FMT2FMT*), interrelationships[3]. We posit that effectively disentangling multilevel feature interrelationships would benefit MTSP with fine-grained temporal and spatial features and allow it to search for an effective combination of the fine-grained features for prediction tasks. Given the growing importance of MTSP, in this study, we strive to design an MTSP method that can effectively disentangle multilevel feature interrelationships for improved prediction performance.

We first probe into our first research question (RQ1): *how to effectively disentangle multilevel feature interrelationships in MTSP?* To this end, based on theoretical foundations regarding graph convolutional networks (GCNs) and a novel graph attention mechanism, we propose a novel *meta-graph learning* (*MGL*) method to disentangle multilevel feature interrelationships. The proposed method learns a *meta-graph*, which is composed of three hierarchically interconnected graphs, to represent multilevel feature interrelationships. Intuitively, each graph consists of an adjacency matrix with learnable weights, representing *F2F*, *M2M*, and *T2T* feature interrelationships, respectively. The interconnections across the graphs allow feature representations to propagate simultaneously through a novel *graph attention mechanism*, which captures superposed interrelationships (i.e., *FM2FM*, *FT2FT*, *MT2MT*, and *FMT2FMT*), thereby quantifying multilevel feature interrelationships with graph structures synchronously and efficiently. The quantification

[3] The ternary interaction, i.e., *FMT2FMT*, subsumes (implies) the pairwise binary interactions, i.e., *FM2FM*, *FT2FT*, and *MT2MT*.

with hierarchically interconnected graphs avoids the combinatorial explosion issue faced by *MGL* when searching for an optimal combination of fine-grained features.

The main challenge in applying the proposed meta-graph is that fine-grained temporal and spatial features and their multilevel interrelationships can lead to a growth in the number of learnable parameters, thus threatening the efficiency and reliability of the model optimization. To overcome this difficulty, we delve into our second research question (RQ2): *how to effectively learn the meta-graph and the multilevel feature interrelationships for MTSP*? In response to RQ2, we inject a novel *robust learning objective* to the meta-graph learning method based on the low-pass nature of graph convolutional filters. We propose *RMGL*, a *robust meta-graph learning* method, to learn the meta-graph and the multilevel feature interrelationships in low-rank parameter space for MTSP.

We have evaluated the effectiveness of *RMGL* in an MTSP problem in the finance area. The dataset consists of multimodal financial data of Standard & Poor's 1500 companies, spanning the period from 2009 to 2020. We constructed MTS features from time-varying contents in acoustic, textual, and numerical modalities to predict firm-level financial risks. The results show that *RMGL* consistently outperformed state-of-the-art alternatives (i.e., benchmarks) in the MTSP task.

Our work contributes to both research and practice. In the era of big data, MTSP is increasingly becoming an appealing way to boost performance in various applications. Rooted in theoretical foundations regarding GCNs, *RMGL* introduces a novel *meta-graph* to disentangle multilevel feature interrelationships for fine-grained representations. The representations are learned through interconnected graphs with a novel *graph attention* mechanism designed for the graph propagation phase. Drawing upon the low-pass nature of GCNs' convolutional filters, *RMGL* introduces a novel *robust learning objective* to ensure the effectiveness and stability of the

learning process. From a practical perspective, *RMGL* has learned fine-grained feature representations and consistently outperformed state-of-the-art alternatives in a real MTSP prediction problem, providing a promising solution for informed decision-making.

2.2. Literature Review

2.2.1 Multimodal Time Series Prediction

Traditional methods for time series analysis, e.g., autoregression, vector autoregression, and their variants, have been commonly applied to prediction tasks with low-dimensional features. While deep-learning-based methods capture non-linearity within high-dimensional features in the temporal dimension, they tend to ignore spatial variation patterns. Examples are recurrent neural networks and their variants (e.g., long short-term memory (LSTM) and gated recurrent units (GRUs)), which are specially designed for sequential inputs. Recently, multimodal deep learning methods (Cheng et al. 2019, Liu et al. 2018) have evolved to learn modality fusion with deep neural networks (DNNs). Learning modality fusion can be seen as modeling spatial variation patterns, but such multimodal deep learning methods do not account for temporal variation patterns.

Given the increasing prevalence of multimodal time series data, MTSP has become an appealing way to boost performance in various fields, including finance, online marketing, healthcare, and transportation (Farahani et al. 2021, Y. Li et al. 2020, Sawhney et al. 2020). For instance, Wang et al. (Wang et al. 2021) showed that applying multimodal financial data (e.g., financial indicators, annual reports, and news) enables earlier prediction of firm-level financial risks. Tao et al. (Tao et al. 2020) obtained a more accurate personalized recommendation using a graph attention network to capture a user's historical preferences over items with multimodal contents. In another case, Mohamed et al. (Mohamed et al. 2020) showed that pedestrian trajectory can be forecasted with video clips modeled through a spatial-temporal graph neural

network. Although MTSP fares better compared to traditional methods, there are several methodological challenges, among which the most salient one is in effectively modeling the intricate feature interrelationships.

2.2.2 Learning Feature Interrelationships in Deep Learning

As noted earlier, we categorize multilevel feature interrelationships in MTS into seven types: *F2F*, *M2M*, *T2T*, *FM2FM*, *FT2FT*, *MT2MT*, and *FMT2FMT* interrelationships. The literature on representation learning has accumulated studies using diverse methods to model different types of feature interrelationships. Among them, attention-based and DNN-based methods have been widely adopted in the deep learning literature. For instance, Song et al. (Song et al. 2019) integrated an attention mechanism with a residual network to model *F2F* interrelationships; the method first transforms features into an embedding space and then applies a multi-head self-attention mechanism to infer interrelationships between feature embeddings. Luo et al. (Luo et al. 2018) adopted GRU cells in a generative adversarial network to model *T2T* interrelationships for missing value imputation. In multimodal representation learning, Wang et al. (Wang et al. 2019) modeled *M2M* interrelationships using a cross-modal correlation metric in the loss function. In another case, Wang et al. (Wang et al. 2016) first encoded visual and linguistic modalities separately and then fine-tuned the feature representations using intra-cross-modal loss to simultaneously capture *FM2FM* interrelationships.

Intuitively, modeling feature interrelationships can be seen as learning a function to map raw features to an embedding space and to infer the interrelationships based on inherent feature associations embedded in the data. While attention mechanisms explicitly learn feature interrelationships through distance metrics or non-linear functions, DNN-based methods rely on network architecture (e.g., LSTM, GRU, convolutional neural network (CNN), convolutional

LSTM (ConvLSTM), Transformer, and graph neural network (GNN)) or curated objective functions to implicitly learn a mapping function. However, both methods fall short of capturing a holistic structure of multilevel feature interrelationships.

Extant methods have primarily adopted a two-stage aggregation strategy to account for parts of multilevel feature interrelationships. For instance, Gu et al. (Gu et al. 2018) proposed a multimodal hierarchical attention network to deal with affection analysis. In the first stage, audio and text contents are aligned in the temporal dimension and then fed into an attention module to capture *T2T* interrelationships within the same modality. In the second stage, another attention module is used to fuse the features across modalities (*M2M*) for making the final prediction. Similarly, temporal-spatial neural networks and sequential multimodal deep learning methods have commonly applied a two-stage aggregation strategy. Specifically, temporal-spatial neural networks aggregate first on the spatial dimension and then on the temporal dimension (Eldele et al., Qin et al.), whereas sequential multimodal deep learning methods aggregate first on the temporal dimension and then on the spatial dimension (Akbari et al. 2021, Zhang et al. 2020). As noted earlier, such a two-stage aggregation strategy can lead to such issues as information loss, information redundancy, and information conflicts due to the ignoration of temporal-spatial covariations.

2.2.3 Graph Convolutional Networks

In recent years, GCNs (Kipf and Welling 2016) have gained explosive development due to their unique capability in learning deep representations from graph-structured data (Fani et al. 2020, Gao et al. 2018, Mohamed et al. 2020, Yao et al. 2019). For a given graph, GCNs formulate it into an adjacency matrix, whose elements indicate the existence, polarity, and intensity of node-to-node interrelationships. Rooted in the graph theory, GCNs adopt *spatial*

graph convolutions, which approximate *spectral graph convolutions* through aggregations on neighboring nodes (S. Zhang et al. 2019). Specifically, a GCN layer first encodes inputs into messages and then aggregates the messages from neighboring nodes to obtain node representations. GCNs can be used to model undirected graphs, directed graphs, signed graphs, and heterogeneous graphs, and caters to a wide range of tasks, such as personalized recommendation (He et al. 2020), traffic prediction (Zhao et al. 2019), and molecular structure inference (Ryu et al. 2018).

In MTSP, GCNs have also shown advantages in learning feature interrelationships. For example, by taking each feature modality as a set of graph nodes, Mai et al. (Mai et al. 2020a) used a hierarchical graph neural network to learn *M2M* interrelationships. Yan et al. (Yan et al. 2018) represented human body skeletons as graphs and modeled the spatial and temporal dynamics (*FT2FT*) using a GCN for human action recognition. Zhao et al. (Zhao et al. 2019) used a GCN to model the topological structures of the road network (*F2F*) for traffic prediction. Song et al. (Song et al. 2020) captured the temporal and spatial dependencies (*FT2FT*) of network data using a GCN. However, in MTSP, extant GCN methods only capture one or a few aspects of multilevel feature interrelationships. Moreover, they tend to rely on the unique characteristics of particular settings to construct the graph (e.g., the traffic network, human body skeletons, and social networks), whereas in general MTSP, a prior graph structure of features is not available, thus calling for learnable graphs.

2.2.4 Research Gaps

Our review of existing studies pinpoints their incompetence in disentangling multilevel feature interrelationships in MTSP. First, extant methods tend to model feature interrelationships using a two-stage aggregation strategy, which falls short of utilizing temporal-spatial

covariations. Second, although there are a few studies that explore temporal and spatial dynamics simultaneously, they fail to investigate multilevel feature interrelationships systematically in a holistic way. Third, in MTSP, GCN methods often model the network with prior graph structures, whereas investigating multilevel feature interrelationships calls for learnable graphs. To overcome the challenges, we propose *RMGL*, a novel robust meta-graph learning method, for MTSP. To manifest the advantages of our proposed method in filling the abovementioned gaps in the literature, we compare it with major related studies in terms of the ability to model multilevel feature interrelationships (Table 2.1).

Study	**F** **2F**	**M** **2M**	**T** **2T**	**FM2** **FM**	**MT2** **MT**	**FT** **2FT**	**FMT2F** **MT**
(Song et al. 2019)	√						
(Wang et al. 2019)		√					
(Wang et al. 2016)				√			
(Duke et al. 2021, Song et al. 2020, Tonekaboni et al. 2021, Yan et al. 2018)						√	
(Cao et al. 2020, Shih et al. 2019, Zhao et al. 2019)	√		√				
(Akbari et al. 2021, Cheng et al. 2022, Eldele et al., Gu et al. 2018, Mai et al. 2020a, Qin et al., Tan et al. 2020, Zhang et al. 2020)		√	√				
(Luo et al. 2018, C. Zhang et al. 2019)	√		√				
(Deng and Hooi 2021)	√		√				
(Chambon et al. 2018, Mai et al. 2020b)	√	√					
(Wei et al. 2019)		√					
RMGL	√	√	√	√	√	√	√

Table 2.1. Comparison of *RMGL* with Existing Relevant Methods.

2.3. Proposed Method

2.3.1 Design Rationales

Our proposed method, *RMGL*, has been built on sound design rationales, which theoretically ensure its effectiveness in learning robust multilevel feature interrelationships. First, our design of meta-graph learning is based on the theoretical foundations of GCNs (Kipf and Welling 2016, Schlichtkrull et al. 2017) and a novel graph attention mechanism. Second, prior studies have analytically pinpointed that GCN is essentially a low-pass filter for its node features (NT and Maehara 2019). That is, when the features of the neighboring nodes are aggregated toward a central node, the ones corresponding to low-frequency graph signals tend to be more intensively scaled. The frequency of graph signals herein can be understood as the feature-wise variance between the neighboring nodes and the central node, with a larger variance corresponding to higher filtering (or smoothing) intensity.

2.3.1.1 Design Rationale of Meta-Graph Learning

Relational GCNs have been used to model large-scale relational data using directed and labeled multi-graphs to deal with various prediction tasks (Kipf and Welling 2016, Schlichtkrull et al. 2017). In MTSP, multilevel feature interrelationships include three unit-level interrelationships, i.e., *F2F*, *M2M*, and *T2T* interrelationships, each of which can be represented by a relational graph. Furthermore, *RMGL* considers fine-grained interrelationships of multilevel features as a superposed effect of *F2F*, *M2M*, and *T2T* interrelationships. Therefore, the design of *RMGL* essentially implements three learnable relational graphs and acquires the fine-grained interrelationships of multilevel features through a novel graph attention mechanism.

Traditional graph attention networks (GATs) (Veličković et al. 2017) learn the edge weight e_{ij} between nodes i and j by the self-attention between the nodes' hidden states. Specifically,

$$e_{ij} = attention(\mathbf{wf}_i, \mathbf{wf}_j), \tag{2.1}$$

$$\alpha_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}, \tag{2.2}$$

where $\mathbf{wf}_i$ and $\mathbf{wf}_j$ are the hidden states of nodes i and j, α_{ij} is the normalized edge weight (i.e., importance) between nodes i and j, and N_i is the set of the neighboring nodes of node i. In this way, GATs implement graph edge weighting at the feature level.

In contrast to traditional GATs, which learn edge weights (or attention weights) at the feature level through self-attention, *RMGL* learns edge weights at multiple levels (i.e., in multiple hierarchically interconnected graphs) and integrates the multilevel edge weights into fine-grained edge weights in a *meta-graph* through operations of *broadcasting* and *multiplication*. Therefore, through the novel *graph attention mechanism*, *RMGL* learns the edge weights (or attention weights) for graph convolutions.

Specifically, *RMGL* defines three graph adjacency matrices $\mathbf{A}^F \in \mathbb{R}^{p \times p}$, $\mathbf{A}^M \in \mathbb{R}^{q \times q}$, and $\mathbf{A}^T \in \mathbb{R}^{T \times T}$ to represent the *F2F*, *M2M*, and *T2T* graphs, respectively. The *F2F* graph contains $p = d \times M \times T$ feature nodes, the *M2M* graph contains $q = M \times T$ modality nodes, the *T2T* graph contains T time-step nodes, and d is the number of feature nodes within a modality. Therefore, *RMGL* has the attended edge weights at the feature, modality, and time-step levels, denoted, respectively, as

$$\alpha_{ij}^F = \frac{a_{ij}^F}{\sum_{i,k=1}^{p} a_{ik}^F}, \tag{2.3}$$

$$\alpha_{ij}^M = \frac{a_{ij}^M}{\sum_{i,k=1}^{q} a_{ik}^M}, \tag{2.4}$$

and

$$\alpha_{ij}^{T} = \frac{a_{ij}^{T}}{\sum_{i,k=1}^{T} a_{ik}^{T}}. \tag{2.5}$$

Finally, the edge weight in the *meta-graph* $\mathbf{A}$ is computed as

$$\mathbf{A} = \mathbf{A}^{T} \prec \mathbf{A}^{M} \prec \mathbf{A}^{F}, \tag{2.6}$$

which accommodates multilevel feature interrelationships (i.e., *FMT2FMT* interrelationships and hence *FM2FM*, *FT2FT*, and *MT2MT* interrelationships). $\prec$ denotes the operation of broadcasting and element-wise multiplication. Specifically, $\mathbf{A}^{T} \prec \mathbf{A}^{M}$ first broadcasts $\mathbf{A}^{T}$ to the dimension of q and then performs the Hadamard product with $\mathbf{A}^{M}$. The broadcasting operation expands each edge weight inside $\mathbf{A}^{T}$ into an $M \times M$ matrix and then concatenates the matrices to match the corresponding sub-matrices in $\mathbf{A}^{M}$. Similarly, $\mathbf{A}^{M} \prec \mathbf{A}^{F}$ first broadcasts $\mathbf{A}^{M}$ to the dimension of $p \times p$ and then performs the Hadamard product with $\mathbf{A}^{F}$.

3.1.2 Design Rationale of Robust Graph Learning

Consider a general undirected graph $G = (V, E)$, where $|V| = a$ and $|E| = a \times a$; V is the set of a nodes; E is the set of $a \times a$ edges. Let $\mathbf{A} \in \mathbb{R}^{a \times a}$ denote an adjacency matrix of G, where $a_{ij} \in \{0, 1\}$ indicates whether an edge e_{ij} exists. In our case, the nodes V are the elementary features within and across modalities and time steps. Let the degree matrix of $\mathbf{A}$ be the diagonal matrix $\mathbf{D} = diag(d_1, d_2, \dots, d_a) \in \mathbb{R}^{a \times a}$, where $d_i = \sum_{j \in V} a_{ij}$. Define the graph Laplacian as $\mathbf{L} = \mathbf{D} - \mathbf{A} \in \mathbb{R}^{a \times a}$ and the augmented normalized graph Laplacian $\tilde{\mathbf{L}} = \mathbf{I} - \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}}$, where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I} \in \mathbb{R}^{a \times a}$ is the identity matrix, and $\widetilde{\mathbf{D}}$ is the degree matrix of the adjacency matrix $\widetilde{\mathbf{A}}$ augmented with self-loops. Denote the matrix of node features as $\mathbf{F} \in \mathbb{R}^{d \times d}$. The GCN model can be expressed as

$$\mathbf{F}^{l+1} = \sigma(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{F}^{l}\mathbf{W}^{l}), \tag{2.7}$$

where σ is the nonlinear activation function, the superscript l denotes the l-th GCN layer, and $\mathbf{W}^{l}$ is the feature weighting matrix of the l-th GCN layer.

Let $\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$. It has been shown that multiplying $\mathbf{F}^{l}$ with $\widehat{\mathbf{A}}$ is equivalent to applying a low-pass filter, resulting in a decrease of high-frequency components (NT and Maehara 2019, F. Wu et al. 2019) of node features and accurate estimations of the true features (Heng et al. 2021, F. Wu et al. 2019). Specifically, denote $\widehat{\mathbf{A}}$ as a GCN filter,

$$\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\big(\widetilde{\mathbf{D}} - \mathbf{L}\big)\widetilde{\mathbf{D}}^{-\frac{1}{2}} = \mathbf{I} - \widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{L}\widetilde{\mathbf{D}}^{-\frac{1}{2}}. \tag{2.8}$$

Let $\hat{\mathbf{L}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{L}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$. $\hat{\mathbf{L}}$ can be decomposed as $\hat{\mathbf{L}} = \widehat{\mathbf{V}}\widehat{\mathbf{\Lambda}}\widehat{\mathbf{V}}^{\mathbf{T}}$, where $\widehat{\mathbf{\Lambda}} = diag(\lambda_1, \lambda_2, \dots, \lambda_a) \in \mathbb{R}^{a\times a}$ is the diagonal matrix with the diagonal elements filled with eigenvalues of $\hat{\mathbf{L}}$. It has been proven that the eigenvalues of $\hat{\mathbf{L}}$, i.e., $\lambda_i \in [0,2)$ (F. Wu et al. 2019). Furthermore, we have

$$\widehat{\mathbf{A}} = \widehat{\mathbf{V}}(\mathbf{1} - \widehat{\mathbf{\Lambda}})\widehat{\mathbf{V}}^{\mathbf{T}}. \tag{2.9}$$

Therefore, the eigenvalue of the GCN filter $\widehat{\mathbf{A}}$ is $(1 - \lambda_i) \in [-1,1)$. Thus, the GCN model (Equation 2.7) can be deemed as a low-pass filter of the node features. Such a result indicates that high-frequency signals (i.e., neighboring nodes with large variance) will be smoothed by the GCN filter.

With an in-depth analysis, researchers have revealed that not all low-pass filters are robust in that lower-frequency components ($\lambda_i \in$ [0, 1) or $1 - \lambda_i \in$ (0, 1]) of the GCN filter can be more robust than higher-frequency ones ($\lambda_i \in$ (1, 2] or $1 - \lambda_i \in$ [-1, 0)) (Heng et al. 2021). In light of this, we introduce the trace norm to regularize the singular values of the graph Laplacian matrix

within the low-rank space (i.e., to induce low-frequency components) to learn *robust* graph filters. Specifically, let $\mathbf{A}$ be the adjacency matrix of a relational graph in *RMGL*. We define its graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$. Define $\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}}$ as the augmented normalized graph Laplacian matrix, where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\widetilde{\mathbf{D}}$ is the degree matrix of $\widetilde{\mathbf{A}}$. Similar to equation (2.8), we have

$$\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}} = \widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{D}} - \mathbf{L}) = \mathbf{I} - \widetilde{\mathbf{D}}^{-1}\mathbf{L}. \tag{2.10}$$

Let $\hat{\mathbf{L}} = \widetilde{\mathbf{D}}^{-1}\mathbf{L}$ and $\left\|\hat{\mathbf{L}}\right\|_{tr}$ denote the trace norm of $\hat{\mathbf{L}}$, that is,

$$\left\|\hat{\mathbf{L}}\right\|_{tr} = \|\sigma\|_1 = \sum_{i=1}^{a} \sigma_i, \tag{2.11}$$

where σ_i is the i-th singular value of matrix $\hat{\mathbf{L}}$. $\left\|\hat{\mathbf{L}}\right\|_{tr}$ is essentially the L1 norm of the singular values of $\tilde{\mathbf{L}}$. Therefore, we impose $\left\|\hat{\mathbf{L}}\right\|_{tr}$ to impel the singular values to be smaller and have $\hat{\mathbf{L}}$ optimized in a *low-rank space* to induce the low-frequency components (i.e., graph filters), such that noise signals can be more effectively filtered.

In this study, we fill $\mathbf{A}$ with learnable weights and decompose $\hat{\mathbf{L}}$ using the *standard vector decomposition* $\hat{\mathbf{L}} = \widehat{\mathbf{X}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{Y}}^{\boldsymbol{T}}$. Denote $\widetilde{\mathbf{\Sigma}}$ as a diagonal matrix filled with a set of descending-ordered diagonal elements in $\widehat{\mathbf{\Sigma}}$, explaining k percent of the total sum of all diagonal elements in $\widehat{\mathbf{\Sigma}}$. We then decompose $\hat{\mathbf{L}} = \widehat{\mathbf{U}} + \widehat{\mathbf{V}}$, where $\widehat{\mathbf{U}} = \widehat{\mathbf{X}}\widetilde{\mathbf{\Sigma}}\widehat{\mathbf{Y}}^{\boldsymbol{T}}$, to consider *filter-wise and weight-wise robustness*, i.e.,

$$\min_{\widehat{\mathbf{U}},\widehat{\mathbf{V}}} \left\|\widehat{\mathbf{U}}\right\|_{tr} + \left\|\widehat{\mathbf{V}}\right\|_{21} \text{ s.t. } \hat{\mathbf{L}} = \widehat{\mathbf{U}} + \widehat{\mathbf{V}}, \tag{2.12}$$

where $\left\|\widehat{\mathbf{V}}\right\|_{21}$ is the L21 norm of $\widehat{\mathbf{V}}$ and is set to increase the *robustness* of node-wise (row-wise) feature weights.

2.3.1.3 Methodological Challenges

We summarize the challenges we have identified, along with our proposed solutions for addressing them, in Table 2.2 *RMGL* identifies multilevel feature interrelationships based on fine-grained temporal and spatial features and leverages them for MTSP through synchronous temporal-spatial modeling.

Challenge	Description	Impact	Solution
Two-stage aggregation	Conducting temporal and spatial aggregation separately	Information loss, redundancy, and conflict	Synchronous temporal-spatial modeling
Multilevel feature interrelationships	*F2F*	Feature-wise interrelationships affect the joint prediction effect of two features.	Meta-graph learning at the feature level
	M2M	Modality-wise interrelationships affect the joint prediction effect of two feature modalities.	Meta-graph learning at the modality level
	T2T	Time-step-wise interrelationships affect the joint prediction effect of features from two time steps.	Meta-graph learning at the time step level
	FT2FT, *FM2FM*, *MT2MT*, *FMT2FMT*	Fine-grained feature interrelationships with superposed feature interrelationships.	Interconnection of *F2F*, *M2M*, and *T2T* graphs
Synchronous temporal-spatial modeling	Delivering fine-grained temporal and spatial features synchronously to prediction	The large number of parameters increases overfitting risk and weakens the robustness of optimization.	Robust graph learning

Table 2.2. Challenges in Leveraging Multilevel Feature Interrelationships for MTSP.

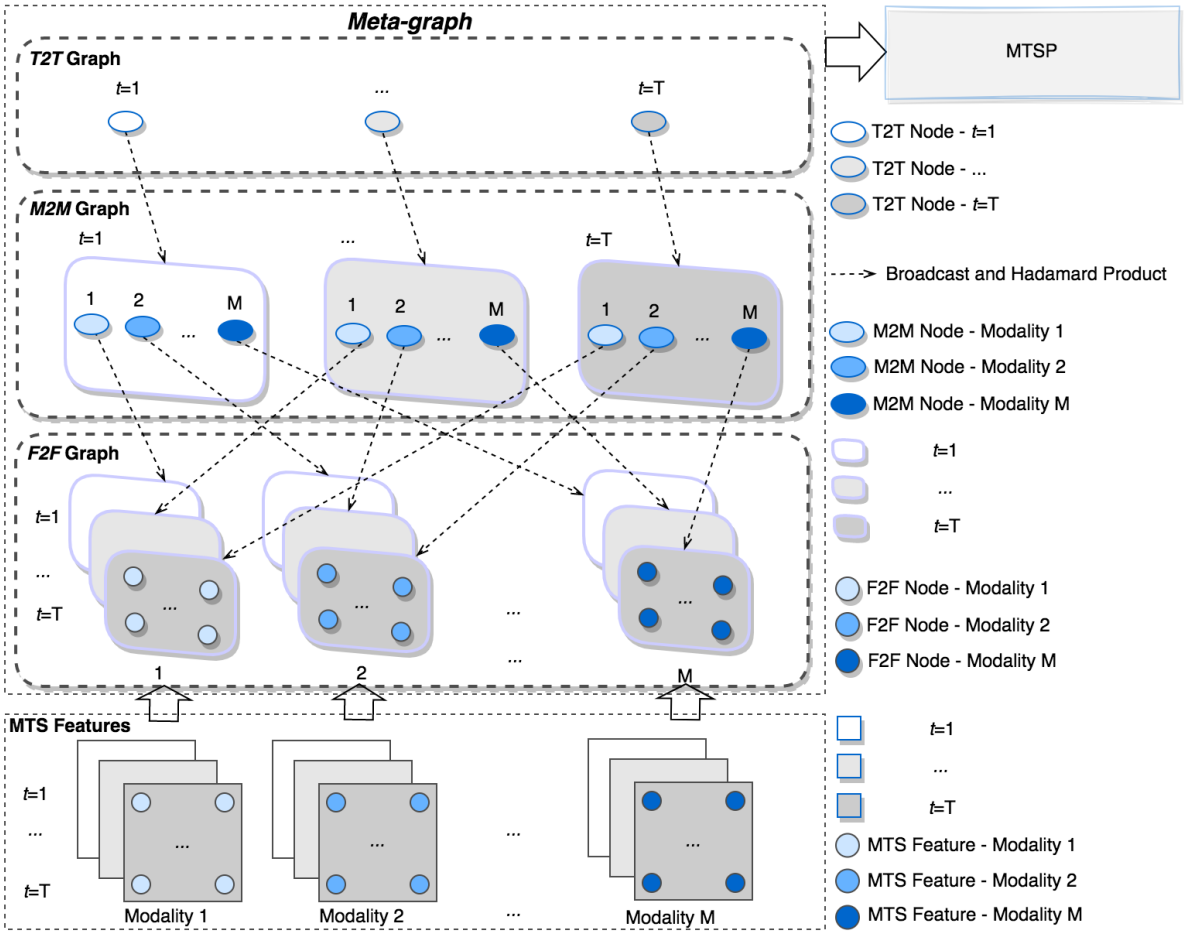


Figure 2.1 Framework of *RMGL*

2.3.2 Framework Overview and Problem Formulation

Overall, *RMGL* is an end-to-end MTS-based deep learning method that learns multilevel feature interrelationships with MTS data and leverages them for MTSP in a robust way. The overall framework of *RMGL* is outlined in Figure 2.1. It first aligns the MTS data in modalities and time steps for the given MTSP task. Next, for each data modality, it applies a one-layer deep neural network to ensure a uniform dimensionality of feature modalities before constructing the graphs. Then, taking each feature, modality, and time step as a graph node successively, it formulates *F2F*, *M2M*, and *T2T* graphs, respectively. The graphs comprise edges representing the intensity and polarity of the interrelationships. Zero edges herein induce graph structures such that a robust feature interaction can be learned for MTSP. Furthermore, the graphs are

interconnected, constituting the meta-graph through operations of broadcasting and Hadamard product to capture the ternary *FMT2FMT* interrelationships. Finally, a *robust meta-graph learning objective* leverages the fine-grained interrelationships to enhance the prediction.

Granularity	Concept	Definition
Interrelationship Level	*F2F* Interrelationship	Interrelationship between two features.
	M2M Interrelationship	Interrelationship between two feature modalities.
	T2T Interrelationship	Interrelationship between two time steps.
	FM2FM Interrelationship	*F2F-M2M* superposed interrelationship.
	FT2FT Interrelationship	*F2F-T2T* superposed interrelationship.
	MT2MT Interrelationship	*M2M-T2T* superposed interrelationship.
	FMT2FMT Interrelationship	Fine-grained *F2F-M2M-T2T* superposed interrelationship.
Feature Importance Style	Inner Importance	The importance of a feature itself.
	Outer Importance	The sum of the contribution degrees of a feature to other
Interrelationship Style	Positive Interrelationship	The positive interrelationship value learned by *RMGL*
	Zero Interrelationship	The zero interrelationship value learned by *RMGL* between
	Negative Interrelationship	The negative interrelationship value learned by *RMGL*

Table 2.3. Definitions of Concepts Related to Multilevel Feature Interrelationships

We formulate the MTSP problem as follows. Consider MTS data instances along with *M* modalities, i.e., $D = \{S_1, S_2, \dots, S_M\}$, where $S_j = \{S_j^{(1)}, S_j^{(2)}, \dots, S_j^{(T)}\}$ is the instance set containing instances of the j-th modality across T time steps. $S_j^{(t)} = \{S_{1j}^{(t)}, S_{2j}^{(t)}, \dots, S_{nj}^{(t)}\}$, where $S_{ij}^{(t)}$ is the i-th instance of the j-th modality at the t-th time step. Denote the variable to be predicted as $\mathbf{y} = [y_1, y_2, \dots, y_n] \in \mathbb{R}^n$. To ensure a uniform dimensionality for all feature modalities, we use a one-layer neural network for each modality instance set (i.e., S_j) to map the dimensionality of the modality to *d*. In line with *D*, we define a batch of MTS representation features of it as $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M] \in \mathbb{R}^{b \times M \times T \times d}$, where $\mathbf{X}_j = [\mathbf{X}_j^{(1)}, \mathbf{X}_j^{(2)}, \dots, \mathbf{X}_j^{(T)}] \in \mathbb{R}^{b \times T \times d}$ and $\mathbf{X}_j^{(t)} = [\mathbf{x}_{j1}^{(t)}, \mathbf{x}_{j2}^{(t)}, \dots, \mathbf{x}_{jd}^{(t)}] \in \mathbb{R}^{b \times d}$ is the output matrix of the j-th modality at the t-th time step over b (i.e., batch size) instances. The MTSP task can be expressed as $\mathbf{y} = f_{MTS}(\mathbf{X})$. For the clarity of

model description in subsequent sections, we define concepts related to multilevel feature interrelationships in Table 2.3.

2.3.3 Meta-graph Construction for Multilevel Feature Interaction

Given the MTS-based representation features, i.e., **X**, the first goal of *RMGL* is to learn fine-grained multilevel feature interrelationships, in response to RQ1. Specifically, corresponding to the three types of feature interrelationships, i.e., *F2F*, *M2M*, and *T2T* interrelationships, *RMGL* constructs three levels of graphs. At the feature level, *RMGL* takes $p = d \times M \times T$ representation features of **X** as p feature nodes and constructs a feature-wise (*F2F*) graph (square adjacency matrix) $\mathbf{A}^F \in \mathbb{R}^{p \times p}$, where $a_{ij}^F \in \mathbb{R}$ is a trainable weight for measuring the interrelationship intensity and polarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Similarly, for $q = M \times T$ feature modalities and T time steps at the modality and time-step levels, *RMGL* constructs the modality-wise (*M2M*) graph $\mathbf{A}^M \in \mathbb{R}^{q \times q}$ and the time-step-wise (*T2T*) graph $\mathbf{A}^T \in \mathbb{R}^{T \times T}$, respectively. $a_{ij}^M \in \mathbb{R}$ (or $a_{ij}^T \in \mathbb{R}$) is a trainable weight for evaluating the interrelationship intensity and polarity between the i-th and j-th modalities (or time steps).

Especially, using a novel *graph attention* mechanism, *RMGL* interconnects $\mathbf{A}^F$, $\mathbf{A}^M$, and $\mathbf{A}^T$ and thus creates a *meta-graph* that provides channels to accommodate the interflow of feature information across modalities and time steps. For a given elementary feature, while *RMGL* leans its interrelationship with other features, the feature interrelationship weights will be adjusted by the weights of modality-wise interrelationships, which will be further adjusted by the weights of time-step-wise interrelationships. In doing so, the learned feature interrelationships are indeed superposed multilevel interrelating effects. This conforms to the methodological requirements of delivering fine-grained temporal and spatial features synchronously to MTSP by considering their intricate interrelationships.

2.3.4 Robust Meta-graph Learning

In response to RQ2, *RMGL* leverages multilevel feature interrelationships to enhance MTSP in a *robust* way. To this end, it performs graph convolutions on the meta-graph to realize the meta-graph learning effect and, in the meanwhile, introduces a *robust learning objective* to realize the robust meta-graph learning effect. Figure 2.2 illustrates the meta-graph structure for learning multilevel feature interrelationships.

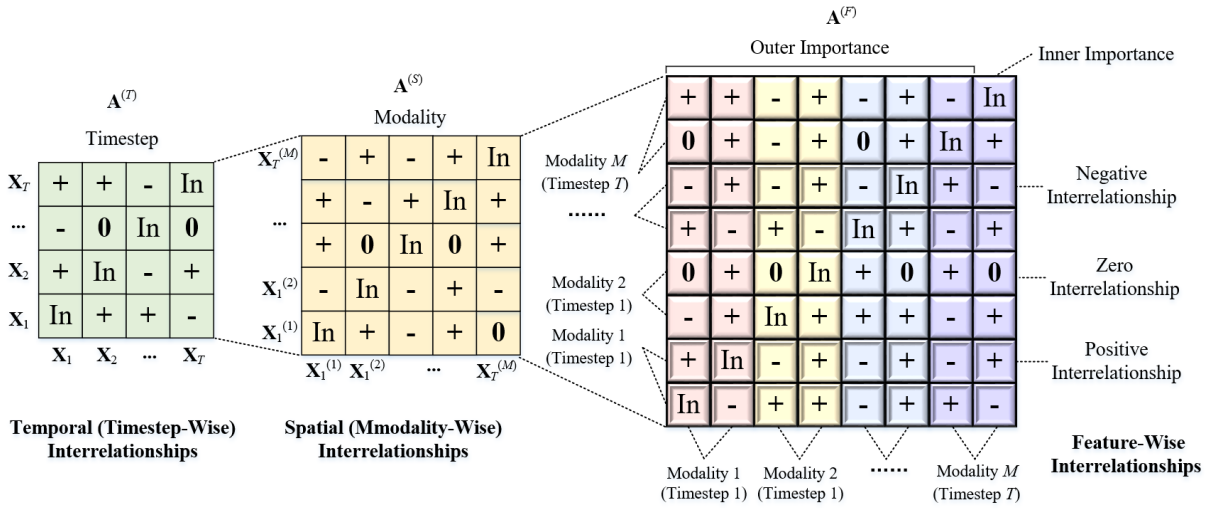


Figure 2.2. Illustration of Meta-graph for Multilevel Feature Interrelationships. From left to right: adjacency matrices of *T2T*, *M2M*, and *F2F* graphs, respectively. The particular values (+, -, 0) in the matrices are arbitrary examples for illustration only. Plus sign (+): positive weight; Negative sign (-): negative weight; Zero (0): zero weight.

Specifically, *RMGL* conducts graph convolutions over MTS features, i.e.,

$$\mathbf{z}^A = \sigma\left(\widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}}\mathbf{X}^T\mathbf{W}^G\right) \in \mathbb{R}^{d_A} \quad , \tag{2.13}$$

$$\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}, \mathbf{A} = \mathbf{A}^T \prec \mathbf{A}^M \prec \mathbf{A}^F. \tag{2.14}$$

$\mathbf{z}^A$ are the representation features output by *RMGL*'s meta-graph layer, which can be single/multi-layer. Adding an output layer on $\mathbf{z}^A$, which encodes multilevel feature

interrelationships, *RMGL* can make the final prediction. The prediction loss depends on the prediction task, e.g., mean square error if $\mathbf{y}$ is continuous:

$$\mathcal{L}_{mse} = \frac{1}{b}\sum_{i=1}^{b}(y_i - \widehat{y}_i)^2, \tag{2.15}$$

Besides, to realize a *robust* learning effect for multilevel feature interrelationships and better leverage them for MTSP, *RMGL* conducts meta-graph learning in the *robust low-rank parameter space*. Specifically, it adds the following penalty terms on the prediction loss ($\mathcal{L}_{mse}$):

$$\mathcal{L}_{reg} = \lambda_1\left(\left\|\mathbf{A}^{(T)}\right\|_1 + \left\|\mathbf{A}^{(M)}\right\|_1 + \left\|\mathbf{A}^{(F)}\right\|_1\right) + \lambda_2\left(\left\|\mathbf{A}^{(T)}\right\|_F + \left\|\mathbf{A}^{(M)}\right\|_F + \left\|\mathbf{A}^{(F)}\right\|_F\right) + \lambda_3\left\|\widehat{\mathbf{U}}\right\|_{tr} + \lambda_4\left\|\widehat{\mathbf{V}}\right\|_{21}, \tag{2.16}$$

where $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}$ are decomposed from the augmented normalized Laplacian matrix ($\widehat{\mathbf{L}} = \widetilde{\mathbf{D}}^{-1}\mathbf{L}$), representing the low-rank and residual matrices of $\widehat{\mathbf{L}}$, respectively. The decomposition is defined as

$$\widehat{\mathbf{L}} = \widehat{\mathbf{U}} + \widehat{\mathbf{V}}, \tag{2.17}$$

where $\widehat{\mathbf{U}} = \widehat{\mathbf{X}}\widetilde{\mathbf{\Sigma}}\widehat{\mathbf{Y}}^{\boldsymbol{T}}$, $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$ are the corresponding left and right rotation matrices from the standard value decomposition on $\widehat{\mathbf{L}}$, and $\widetilde{\mathbf{\Sigma}}$ is defined as *k* percent of the stretching matrix $\widehat{\mathbf{\Sigma}}$,

$$\widehat{\mathbf{L}} = \widehat{\mathbf{X}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{Y}}^{\boldsymbol{T}} \text{ and } \widetilde{\mathbf{\Sigma}} = k \text{ of } \widehat{\mathbf{\Sigma}}. \tag{2.18}$$

The operation of "k of $\widehat{\mathbf{\Sigma}}$" is defined as a diagonal matrix whose diagonal elements are filled with a set of descending-ordered diagonal elements in $\widehat{\mathbf{\Sigma}}$, explaining k percent of the total sum of all diagonal elements in $\widehat{\mathbf{\Sigma}}$.

$\|.\|_1$ denotes the elementwise L1-norm-based penalty term over a matrix to ensure elementwise matrix sparsity (i.e., zero edge weights in *F2F*, *M2M*, and *T2T* graphs). $\|.\|_F$ denotes the Frobenius-norm-based penalty term to regularize the weight elements of a matrix to overcome overfitting. $\|.\|_{tr}$ denotes the trace-norm-based penalty term over a matrix to increase the robustness of graph filters. $\|.\|_{21}$ denotes the L21-norm-based penalty terms over a matrix to induce structural sparsity.

By equation (2.16), *RMGL* optimizes the weights of multilevel feature interrelationships in the low-rank parameter space. That is, *RMGL* applies the regularizations on the decomposed graph Laplacian matrix, respectively with the trace-norm-based penalty ($\|.\|_{tr}$) and the L21-norm-based penalty ($\|.\|_{21}$). As we analyzed before, $\|.\|_{tr}$ helps to increase the robustness of graph filters in the low-rank space and $\|.\|_{21}$ helps to increase the robustness of the residual matrix, inducing patterns of node-wise (row-wise) weights such that row weight vectors with greater importance tend to be less shrunk.

In summary, the overall learning objective of *RMGL* can be expressed as

$$\mathcal{L}_{RMGL} = \mathcal{L}_{mse} + \mathcal{L}_{reg}. \tag{2.19}$$

2.3.5 The Algorithm for *RMGL*

The proposed algorithm for *RMGL* is outlined in Figure 2.3.

Input:	$D, \boldsymbol{y}$
Parameters:	$d, k, \lambda_1, \lambda_2, \lambda_3, \lambda_4, e, b$
Output:	$\hat{y} = f_{MTS}(\mathbf{X})$, along with learned multilevel feature interrelationships $\overline{\mathbf{A}}^F$, $\overline{\mathbf{A}}^M$, $\overline{\mathbf{A}}^T$

1:	Construct feature modalities for MTS.
2:	Map the dimensionality of each feature modality to d using a one-layer neural network.
3:	Encode D as MTS representation features $\mathbf{X}$ by feature modalities.
4:	Initialize $\mathbf{A}^F$, $\mathbf{A}^M$, $\mathbf{A}^T$, $\mathbf{W}^G$ with Xavier initializer (Glorot and Bengio 2010)
5:	**for** epoch←1 to e
6:	**for** batch ←1 to b
7:	Normalize $\mathbf{A}^F$, $\mathbf{A}^M$, $\mathbf{A}^T$ by Eq. (3-5).
8:	Derive the meta-graph $\mathbf{A}$ by Eq. (6).
9:	Derive $\mathbf{z}^A$ by Eq. (13-14).
10:	Pass $\mathbf{z}^A$ through the output layer to get prediction $\hat{y}_i$.
11:	Compute prediction loss by Eq. (15).
12:	Compute robustness loss by Eq. (16-18).
13:	Compute overall loss by Eq. (19).
14:	**end for**
15:	Run a backpropagation for optimization.
16:	**end for**
Return	The trained model $f_{MTS}(\mathbf{X})$, along with $\overline{\mathbf{A}}^F$, $\overline{\mathbf{A}}^M$, $\overline{\mathbf{A}}^T$.

Figure 2.3. The Algorithm for *RMGL*

2.4. Empirical Evaluation

2.4.1 Data

We have collected a financial dataset of Standard & Poor's 1500 companies (S&P1500), spanning the period from 2009 to 2020, to evaluate the proposed method. Specifically, the dataset includes 1) quarterly accounting information collected from the Compustat database, 2) audio recordings of firms' quarterly earnings call collected from the Factset Database, 3) trader- and investor-generated content at the firm level collected from a social media platform dedicated to stock markets, 4) employee-generated content at the firm level collected from a social media platform dedicated to employer brandings, and 5) firms' daily stock prices collected from the Center for Research in Securities Prices (CRSP). Based on the dataset, we extracted five modalities of features and formulated the forecasting of the n-day stock volatility following quarterly earnings call events (i.e., each time step corresponds to a quarter) as an MTSP problem.

Earnings calls are quarterly conference calls hosted by company executives to discuss the firm's overall performance with outside investors and analysts. They can result in significant stock price moves, which are also known as post-earnings announcement drift, a phenomenon widely studied in finance and accounting research (Bernard and Thomas 1989). Investors and analysts analyze the earnings announcement, interpret the vocal and verbal cues during the conference call, such as the tones and emotions of firm CEOs, and then react on the market. In the meanwhile, it has been shown that user-generated content on social media platforms can predict stock price moves. For instance, Deng et al. (Deng et al. 2018) showed that microblog sentiments extracted from Stocktwits.com can reflect firms' fast stock price moves at the hourly level. In another case, Green et al. (Green et al. 2019) showed that employee ratings on Glassdoor.com reveal firms' intangible assets and therefore can predict future stock returns.

Variable	**Value**
Dimensionality of FI	12
Dimensionality of EA	12
Dimensionality of ET	1,024
Dimensionality of TC	1,024
Dimensionality of EC	1,024
ET text length in words (Min / Max / Mean / Std)	215 / 27,221 / 7,879.84 / 2,479.75
EA audio length in seconds (Min / Max / Mean / Std)	486.03 / 10,148.26 / 3,331.39 / 893.80
TC text length in words (Min / Max / Mean / Std)	1 / 808,370 / 4,828.70 / 30,845.79
EC text length in words (Min / Max / Mean / Std)	9 / 126,360 / 2,052.94 / 5,283.33
Number of companies / Sample size (T=1)	1,369 / 26,496
Number of companies / Sample size (T=2)	1,199 / 18,815
Number of companies / Sample size (T=3)	1,053 / 14,870
Number of companies / Sample size (T=4)	951 / 12,390
Number of companies / Sample size (T=5)	826 / 10,622
Number of companies / Sample size (T=6)	733 / 9,263
Number of companies / Sample size (T=7)	637 / 8,205
Number of companies / Sample size (T=8)	571 / 7,351

Table 2.4. Descriptive Statistics of the S&P1500 MTS dataset

The n-day stock volatility captures the stock price moves in a window of n days and is an essential measure of risk levels in the financial market (Qin and Yi 2019). With the availability of MTS data, predicting the *n*-day stock volatility following earnings call events presents an opportunity for evaluating the proposed graph learning artifact (i.e., *RMGL*). The collected dataset consists of a large sample of firms with MTS data representing firm values in various dimensions, including 1) Financial and accounting Indicators (the FI modality), 2) vocal features of CEOs embedded in the Earnings call Audio (the EA modality), 3) verbal features of CEOs embedded in the Earnings call Transcripts (the ET modality), 4) text features embedded in the Trader- and investor-generated Content (the TC modality), and 5) text features embedded in the Employee-generated Content (the EC modality). *RMGL* can capture the feature interrelationships within and between modalities and across the time dimension, thereby improving the overall prediction performance. Table 2.4 summarizes descriptive statistics about the dataset.

2.4.2 Multimodal Time Series Processing and Feature Alignment

We preprocessed the dataset to obtain the variable to be predicted (i.e., n-day stock volatility) and five modalities of features and then aligned the multimodal features in the time dimension to carry out the experiments. Specifically, the n-day stock volatility is defined as

$$v_{[0,n]} = \ln\left(\sqrt{\frac{\sum_{i=1}^{n}(r_i-\overline{r})^2}{n}}\right), \tag{2.20}$$

where $r_i = (P_i - P_{i-1})/P_{i-1}$ is the stock return on day i, $\overline{r}$ is the average stock return in a window of *n* days following an earnings call, and P_i is the adjusted closing price of a stock on day *i*. Based on equation (2.20) and the dates of earnings call events, we constructed the *n*-day

stock volatility with varying number of days (n=7, 15, 30) to capture short- and long-term predictions.

The five feature modalities (i.e., M=5) we constructed essentially cover three data types: quantitative data (FI), text data (ET, TC, EC), and audio data (EA). To construct the quantitative FI features, following the finance and accounting literature (Edmans 2011, 2012, Green et al. 2019, Pointer and Khoi 2019), we first merged the CRSP and Compustat databases based on the fiscal year and quarter of an earnings call event and then extracted twelve financial indicators at the firm level, including total asset size, fixed asset size, research and development asset size, cash asset size, debt asset size, capital expenditure, operating income, book to market, return on assets, return on equity, Tobin's q, and the quarterly stock returns prior to the earnings call event. To obtain the EA features, we applied librosa, a python package to extract Mel-Frequency cepstral coefficients (MFCC), a common parametric representation of acoustic signals. We extracted, from each firm-quarter audio, a 1024-dimensional MFCC vector and used it to represent the EA modality. To obtain the ET features, we first applied a commercial speech-to-text software to convert each audio recording to a text document and then employed BERT (Bidirectional Encoder Representations from Transformers), a common pre-trained text representation model (Devlin et al. 2018), to obtain a 1024-dimensional vector representing the document. Because BERT limits the input length to 512 characters, we split the document into multiple sub-documents at the unit length of 512 characters, fed the sub-documents into BERT, and then obtained an aggregated 1024-dimensional vector by averaging the feature vectors of the sub-documents. Thus, we obtained a 1024-dimensional vector per firm-quarter transcript to represent the ET modality. To obtain the TC features, for each earnings call, we collected a document of the trader- and investor-generated messages posted under the firm's stock symbol in

a 30-day window prior to (including) the date of the earnings call. We then applied the pre-trained BERT model to obtain an aggregated 1024-dimensional vector per firm-quarter to represent the TC modality. Similarly, to obtain the EC features, for each earnings call, we first collected a document of the employee-generated messages posted under the firm's branding in a time window between the dates of two quarterly earnings calls and then used the pre-trained BERT model to obtain an aggregated 1024-dimensional vector per firm-quarter to represent the EC modality.

To prepare the final dataset to train *RMGL*, we filtered the instances in two ways. First, if any of the five modalities is missing in an instance, we removed the instance. Second, in the setting of T time steps, we removed an instance if any of its previous T-1 quarterly instances is missing.

2.4.3 Experiments

We conducted three experiments to evaluate *RMGL* in the following aspects:

1) to compare *RMGL* against several benchmarks in terms of prediction performance,

2) to show consistent performance margins of *RMGL* against varying modalities in a modality ablation study,

3) to validate and gain further insights into the design choices of *RMGL* with an ablation study comparing *RMGL* against several variants.

In Experiment 1, we compared *RMGL* against nine benchmarks selected from state-of-the-art attention-based deep learning methods, multimodal deep learning methods, and temporal-spatial neural network methods. We first included two attention-based deep learning methods, i.e., *LSTM+Attention*, a long short-term memory network with an attention mechanism on the

sequential inputs, and *AutoInt* (Song et al. 2019), a deep learning method that learns high-order feature interactions with multi-head attention mechanism and residual connection. Next, in view of the design of modality-wise interrelationships incorporated by *RMGL*, we also identified five benchmarks from multimodal deep learning methods, which are aimed at synergizing multiple data modalities within a unified deep learning framework through various fusion methods. The multimodal deep learning benchmarks include: 1) *Soft-HGR* (Wang et al. 2019), a deep learning method that extracts informative features from multiple data modalities by modeling modality-wise correlations and makes predictions for the downstream prediction task by feeding the joint representation into a SoftMax layer, 2) *MAG* (Rahman et al. 2020), a multimodal adaptation gate that allows pre-trained deep neural networks to be fine-tuned with multimodal data, 3) *ARGF* (Mai et al. 2020b), an adversarial representation graph fusion framework for multimodal fusion, which learns a discriminative joint embedding space for various modalities via adversarial training and fuses modalities with a hierarchical graph network, 4) *MRACNN* (Zhang et al. 2020), a multimodal recurrent attention convolutional neural network that models high-order feature interactions with a multimodal factorized bilinear pooling approach, and 5) *BBFN* (Han et al. 2021), a bi-bimodal fusion network that performs fusion and separation on pairwise modality representations. Finally, we identified two deep learning methods that account for temporal and spatial dynamics when modeling modality-wise interrelationships. The two temporal-spatial benchmarks are: 1) *STAN* (Cheng et al. 2020), a spatial-temporal-attention-based neural network that measures the importance of temporal and spatial slices, and 2) *MAGNN* (Cheng et al. 2022), a multimodal graph neural network for forecasting temporal dynamics by incorporating sources of lead-lag relationships. In addition, we examined short-term (7-day) and long-term (15-day and 30-day) volatility predictions (i.e., n=7, 15, 30), as well as

varying number of time steps from one to eight (i.e., T= 1, 2, …, 8). Therefore, Experiment 1 has 10 (*RMGL* + 9 benchmarks) * 3 (n values) * 8 (numbers of time steps) = 240 settings.

In Experiment 2, we conducted a modality ablation study to evaluate the stability of *RMGL*'s performance. We are interested in investigating how the performance would change w.r.t. different settings of modalities and whether *RMGL* with all modalities can outperform *RMGL* with ablated modalities. Specifically, in each run, the input modalities excluded one of the FI, EA, ET, TC, and EC modalities, respectively. We conducted Experiment 2 using the short-term (i.e., 7-day) volatility prediction task (i.e., n=7) because it is a more challenging task. We chose the number of time steps as six (i.e., T=6) because *RMGL* yielded the best performance in Experiment 1 under this setting. Experiment 2, therefore, has 6 (five ablated sets of modalities + all modalities) settings.

In Experiment 3, we conducted an ablation study to investigate the utility of the design components introduced in *RMGL*. Specifically, we compared *RMGL*, which models *FMT2FMT* interrelationships and hence *FM2FM*, *FT2FT*, and *MT2MT* interrelationships, with its three variants, i.e., (1) *RMGL* without the effect of the feature-wise (*F2F*) graph, modeling *MT2MT* interrelationships, (2) *RMGL* without the effect of the modality-wise (*M2M*) graph, modeling *FT2FT* interrelationships, and (3) *RMGL* without the effect of the time-step-wise (*T2T*) graph, modeling *FM2FM* interrelationships. To ablate the effect of each graph (i.e., *F2F*, *M2M*, or *T2T*), we set the corresponding adjacency matrix (i.e., $\mathbf{A}^F$, $\mathbf{A}^M$, or $\mathbf{A}^T$) to an all-one matrix and kept the other two adjacency matrices still learnable. We conducted Experiment 3 using the short-term (i.e., 7-day) volatility prediction task with six time steps (i.e., n=7, T=6), the same as in Experiment 2. Therefore, Experiment 3 has 4 (*RMGL* + 3 ablated variants) settings.

For each setting in the experiments, we estimated the prediction performance through five independent runs of 10-fold cross validation, resulting in 50 estimates. We used root mean square error (RMSE) as the performance metric. Information on the implementation, execution, and parameter setting is available in Appendix D.

2.4.4. Experiment Results

2.4.4.1. Experiment 1: Performance Comparisons

Table 2.5 summarizes the prediction performance (RMSE) of *RMGL* vs the nine benchmarks. Tukey-Kramer test showed that *RMGL* consistently outperformed all benchmarks under every setting (see Appendix E). Among all comparisons, the maximum performance gain of *RMGL* was 49.3% when predicting 30-day volatility with two time steps (RMSE=0.352), compared to that of *AutoInt* (RMSE=0.694). Meanwhile, *RMGL* obtained the minimum gains of 6.3%, 21.4%, and 23.1%, when achieving the best performance in predicting 7-day volatility with six time steps (RMSE=0.507), 15-day volatility with one time step (RMSE=0.370), and 30-day volatility with two time steps (RMSE=0.352), compared to that of *MAG* (RMSE=0.541), *ARGF* (RMSE=0.471), and *BBFN* (RMSE=0.458), respectively. When *RMGL* achieved the best short-term prediction performance (7-day volatility, T=6), the minimum improvements were 6.5% over the attention-based methods, 6.3% over the multimodal deep learning methods, and 9.9% over the temporal-spatial neural network methods. For long-term prediction (30-day volatility, T=2), the minimum improvement over the benchmarks was 46.1% over the attention-based methods, 23.1% over the multimodal deep learning methods, and 44.3% over the temporal-spatial neural network methods, respectively. Consistent with the literature (J. Li et al. 2020, Qin and Yi 2019), *RMGL* obtained better performance for long-term prediction (RMSE=0.352), compared to that for short-term prediction (RMSE=0.507). 5) Finally, *RMGL* obtained the minimum prediction error at T=6 (RMSE=0.507), 1 (RMSE=0.435), and 2 (RMSE=0.384) for

the 7-, 15-, and 30-day volatility, respectively, indicating that *RMGL* can use a long or short MTS to improve the prediction performance.

Method	*n*	***T***							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
LSTM+Attention	7	0.559 (0.063)	0.574 (0.095)	0.555 (0.028)	0.589 (0.002)	0.552 (0.028)	0.542 (0.099)	0.603 (0.029)	0.598 (0.030)
AutoInt		0.561 (0.068)	0.561 (0.029)	0.555 (0.018)	0.597 (0.057)	0.561 (0.029)	0.547 (0.039)	0.584 (0.060)	0.583 (0.021)
Soft-HRG		0.577 (0.021)	0.559 (0.025)	0.549 (0.018)	0.597 (0.023)	0.565 (0.058)	0.601 (0.001)	0.570 (0.137)	0.595 (0.047)
MAG		0.570 (0.014)	0.569 (0.019)	0.569 (0.003)	0.592 (0.047)	0.597 (0.015)	0.542 (0.080)	0.565 (0.014)	0.575 (0.070)
ARGF		0.555 (0.022)	0.553 (0.113)	0.558 (0.017)	0.589 (0.009)	0.560 (0.064)	0.554 (0.028)	0.569 (0.043)	0.575 (0.011)
MARCNN		0.553 (0.059)	0.555 (0.043)	0.565 (0.021)	0.580 (0.052)	0.547 (0.030)	0.566 (0.005)	0.567 (0.087)	0.577 (0.017)
BBFN		0.579 (0.034)	0.595 (0.006)	0.585 (0.075)	0.597 (0.125)	0.568 (0.078)	0.568 (0.055)	0.588 (0.140)	0.578 (0.065)
STAN		0.552 (0.007)	0.559 (0.006)	0.573 (0.022)	0.615 (0.038)	0.552 (0.019)	0.569 (0.005)	0.591 (0.080)	0.597 (0.035)
MAGNN		0.559 (0.036)	0.635 (0.057)	0.605 (0.128)	0.580 (0.039)	0.548 (0.010)	0.563 (0.117)	0.576 (0.028)	0.609 (0.049)
RMGL		**0.526 (0.015)**	**0.513 (0.014)**	**0.518 (0.054)**	**0.546 (0.074)**	**0.512 (0.041)**	**0.507 (0.013)**	**0.515 (0.062)**	**0.535 (0.053)**
LSTM+Attention	15	0.479 (0.013)	0.487 (0.066)	0.692 (0.002)	0.482 (0.008)	0.680 (0.092)	0.501 (0.034)	0.687 (0.019)	0.690 (0.056)
AutoInt		0.491 (0.182)	0.474 (0.009)	0.668 (0.036)	0.676 (0.039)	0.671 (0.025)	0.510 (0.042)	0.484 (0.006)	0.473 (0.063)
Soft-HRG		0.691 (0.023)	0.676 (0.001)	0.527 (0.074)	0.508 (0.030)	0.671 (0.085)	0.540 (0.002)	0.502 (0.039)	0.691 (0.028)
MAG		0.503 (0.059)	0.494 (0.029)	0.669 (0.003)	0.671 (0.021)	0.686 (0.070)	0.493 (0.202)	0.504 (0.105)	0.533 (0.044)
ARGF		0.471 (0.033)	0.483 (0.021)	0.514 (0.019)	0.474 (0.024)	0.473 (0.034)	0.514 (0.069)	0.478 (0.024)	0.477 (0.017)
MARCNN		0.671 (0.026)	0.470 (0.017)	0.685 (0.061)	0.511 (0.022)	0.684 (0.065)	0.534 (0.013)	0.679 (0.007)	0.515 (0.003)
BBFN		0.490 (0.048)	0.515 (0.137)	0.493 (0.161)	0.496 (0.008)	0.494 (0.019)	0.496 (0.171)	0.486 (0.093)	0.515 (0.013)
STAN		0.528 (0.022)	0.678 (0.016)	0.659 (0.018)	0.667 (0.064)	0.698 (0.025)	0.688 (0.049)	0.686 (0.011)	0.561 (0.085)
MAGNN		0.484 (0.122)	0.497 (0.038)	0.524 (0.031)	0.476 (0.158)	0.474 (0.124)	0.544 (0.069)	0.668 (0.076)	0.482 (0.083)

RMGL		**0.370 (0.037)**	**0.439 (0.022)**	**0.44 9 (0.074)**	**0.424 (0.098)**	**0.381 (0.049)**	**0.411 (0.016)**	**0.428 (0.084)**	**0.395 (0.045)**
LSTM+Attention	30	0.443 (0.150)	0.654 (0.031)	0.61 9 (0.009)	0.629 (0.053)	0.462 (0.063)	0.635 (0.032)	0.635 (0.025)	0.627 (0.004)
AutoInt		0.663 (0.076)	0.694 (0.030)	0.44 7 (0.095)	0.631 (0.057)	0.636 (0.012)	0.463 (0.185)	0.442 (0.073)	0.451 (0.174)
Soft-HRG		0.641 (0.006)	0.642 (0.027)	0.61 2 (0.007)	0.669 (0.046)	0.600 (0.000)	0.697 (0.034)	0.613 (0.056)	0.639 (0.046)
MAG		0.451 (0.019)	0.471 (0.004)	0.45 5 (0.062)	0.444 (0.078)	0.434 (0.062)	0.664 (0.005)	0.652 (0.067)	0.648 (0.001)
ARGF		0.623 (0.039)	0.657 (0.047)	0.43 4 (0.049)	0.445 (0.065)	0.440 (0.103)	0.694 (0.009)	0.632 (0.003)	0.439 (0.066)
MARCNN		0.642 (0.072)	0.632 (0.010)	0.66 2 (0.025)	0.656 (0.015)	0.608 (0.001)	0.632 (0.067)	0.622 (0.011)	0.637 (0.031)
BBFN		0.446 (0.083)	0.458 (0.075)	0.46 2 (0.170)	0.454 (0.005)	0.449 (0.129)	0.460 (0.128)	0.458 (0.116)	0.442 (0.085)
STAN		0.652 (0.068)	0.632 (0.003)	0.68 7 (0.072)	0.637 (0.037)	0.618 (0.042)	0.649 (0.075)	0.660 (0.061)	0.671 (0.012)
MAGNN		0.641 (0.024)	0.668 (0.038)	0.66 4 (0.091)	0.437 (0.145)	0.450 (0.057)	0.670 (0.004)	0.670 (0.094)	0.641 (0.057)
RMGL		**0.399 (0.049)**	**0.352 (0.038)**	**0.38 6 (0.055)**	**0.387 (0.019)**	**0.362 (0.007)**	**0.397 (0.008)**	**0.397 (0.045)**	**0.368 (0.021)**

Table 2.5. Prediction Error (RMSE) of *RMGL* vs the Benchmarks

2.4.4.2 Experiment 2: Modality Ablation

Table 2.6 summarizes the results of Experiment 2. Under this setting of absenting a modality in each run (each column header indicates the modality being absented from the experiment), *RMGL* with full modalities achieved the best performance, outperforming *RMGL* with reduced modalities by a margin ranging from 4.3% to 8.6%, indicating that *RMGL* is quite powerful in terms of leveraging new modalities for prediction (See Appendix F for Tukey-Kramer test result). Furthermore, the result shows that compared to other modalities, the FI modality contributed the most to the 7-day volatility prediction task, since when ablating the FI modality from the experiment, the prediction error (RMSE) increased the most. On the other hand, the EC modality contributed the least to the 7-day volatility prediction task, as when ablating the EC modality, the prediction error (RMSE) increased the least.

Set of Modalities	RMSE
-FI	0.555 (0.024)
-EA	0.538 (0.049)
-ET	0.542 (0.043)
-TC	0.546 (0.020)
-EC	0.530 (0.038)
Full	**0.507 (0.038)**

Table 2.6. Prediction Error (RMSE) of *RMGL* with Ablated Sets of Modalities

2.4.4.3 Experiment 3: Design Ablation

Table 2.7 summarizes the results of Experiment 3. The removal of any of the three components in *RMGL* led to a significant decline in performance (see Appendix G for Tukey-Kramer test result). Specifically, the removal of modality-wise interaction led to the most significant decline of 23.9% in predicting the 7-day volatility, followed by the removal of temporal interaction, which led to a decline of 10.6%. Finally, the removal of feature-wise interaction led to a decline of 4.7%. The ablation study shows that by incorporating multilevel feature interrelationships in the design, *RMGL* improved the prediction performance significantly.

Ablated variant	RMSE
Without feature-wise interaction (*MT2MT*)	0.532 (0.023)
Without modality-wise interaction (*FT2FT*)	0.666 (0.042)
Without temporal interaction (*FM2FM*)	0.567 (0.024)
Full	**0.507 (0.013)**

Table 2.7. Prediction Error (RMSE) of *RMGL* vs Ablated Variants

2.4.5. Further Analyses

We conducted further analyses following the main experiments. We first evaluated the effectiveness of *RMGL*'s robust learning objective through a sensitivity analysis of the

hyperparameters of the penalty terms in the loss function (Equation 2.16). We then visualized the graph adjacency weights of *RMGL* to investigate the temporal and spatial variation patterns. We chose the same settings as previous experiments to conduct the analysis, i.e., the short-term (7-day) volatility prediction with six time steps.

2.4.5.1 Sensitivity of Hyperparameters of the Penalty Terms

Figure 2.4 shows that the prediction error (RMSE) of *RMGL* varied across different settings of the values of the four hyperparameters. Specifically, *RMGL* obtained the best performance with the following settings of the four hyperparameters: $\boldsymbol{\lambda_1} = 0.0001$, $\boldsymbol{\lambda_2} = 0.001$, $\boldsymbol{\lambda_3} = 0.01$, and $\boldsymbol{\lambda_4} = 0.00005$. When fixing $\boldsymbol{\lambda_2}$, $\boldsymbol{\lambda_3}$, and $\boldsymbol{\lambda_4}$ to their optimal settings and varying the value of $\boldsymbol{\lambda_1}$ to 0.000001, 0.00005, 0.0001, 0.001, 0.01, and 0.1 respectively, the prediction error of *RMGL* gradually decreased, achieving the minimum at $\boldsymbol{\lambda_1} = 0.0001$ (RMSE=0.507), and then started to increase ($\boldsymbol{\lambda_1} = 0.1$, RMSE=0.68). The pattern occurred similarly for $\boldsymbol{\lambda_2}$, $\boldsymbol{\lambda_3}$, and $\boldsymbol{\lambda_4}$. The results show that every penalty term in the loss function contributed to the prediction performance of *RMGL*, indicating the effectiveness of the robust learning objective designed for *RMGL*.

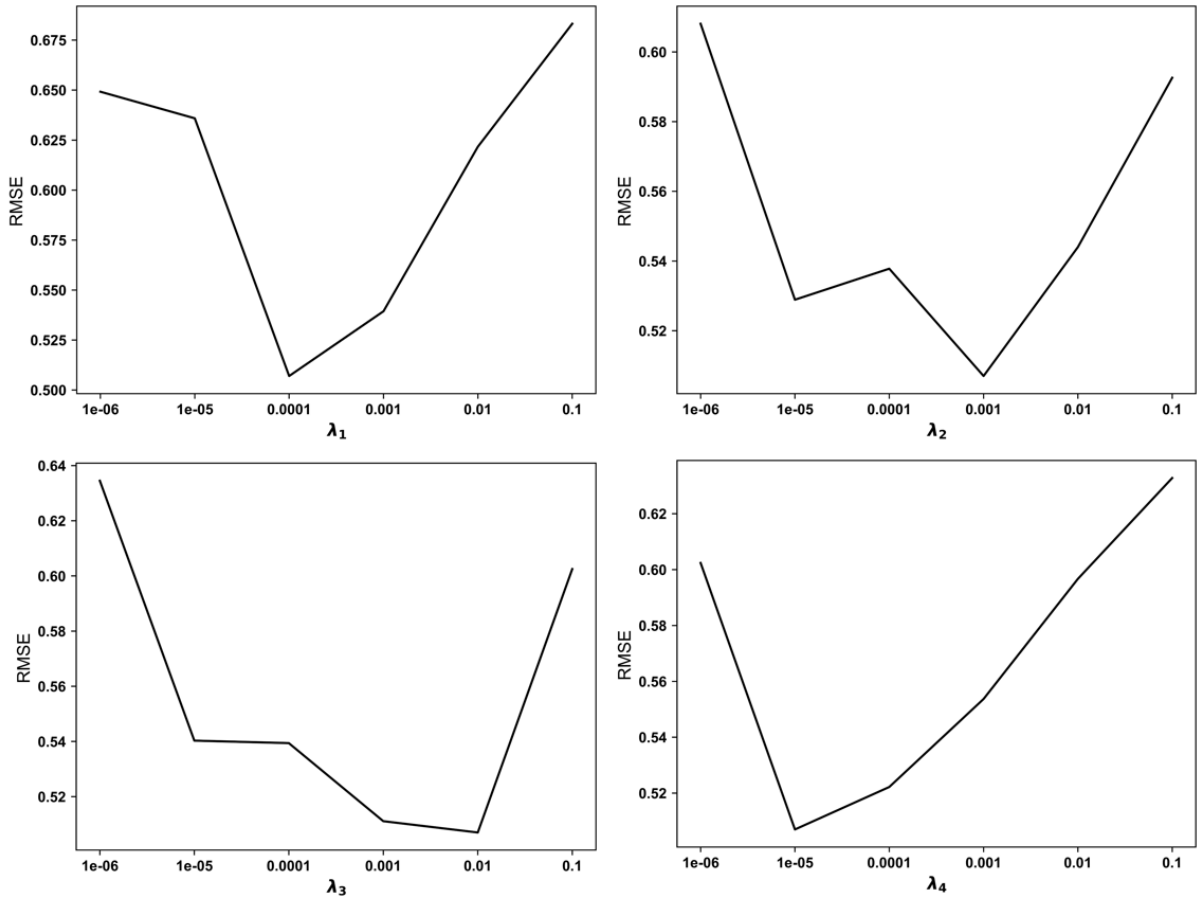


Figure 2.4. Sensitivity Analysis on Hyperparameters of the Penalty Terms

2.4.5.2 Visualization of the Learned Graph Weights

Figure 2.5 displays the adjacency weights of the *T2T* graph. *RMGL* learned *T2T* interrelationships that exhibited different directions and magnitudes across six time steps. For instance, the first time step had a positive and strong interrelationship with the sixth time step, whereas it had negative and weaker interrelationships with the second, fourth, and fifth time steps. The varying *T2T* interrelationships indicate that *RMGL* learned intricate *T2T* interrelationships.

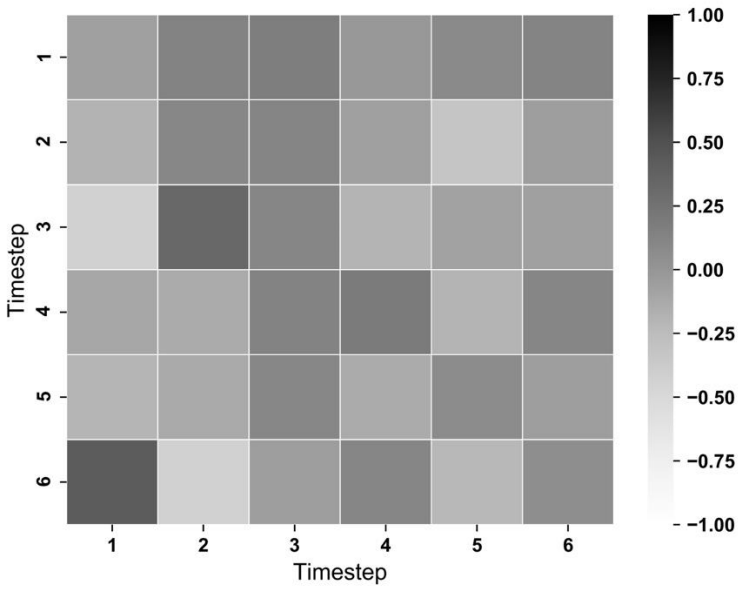


Figure 2.5. Adjacency Weights of the Time-step-wise Graph

Figure 2.6 displays the adjacency weights of the modality-wise graph extracted from six time steps. Within the same time step, the *M2M* interrelationships differed in their directions and magnitudes. Moreover, the variation patterns of the *M2M* interrelationships varied across time steps. For instance, in the first time step, the first modality is negatively related to the second modality and positively related to the third, fourth, and fifth modalities, whereas in the second time step, the first modality is positively related to the second, third, and fifth modalities, and negatively related to the fourth modality.

Figure 2.7 displays the full adjacency weights of the modality-wise graph. The varying *M2M* interrelationships within- and across-temporal dimensions indicate that *RMGL* learned intricate *M2M* interrelationships.

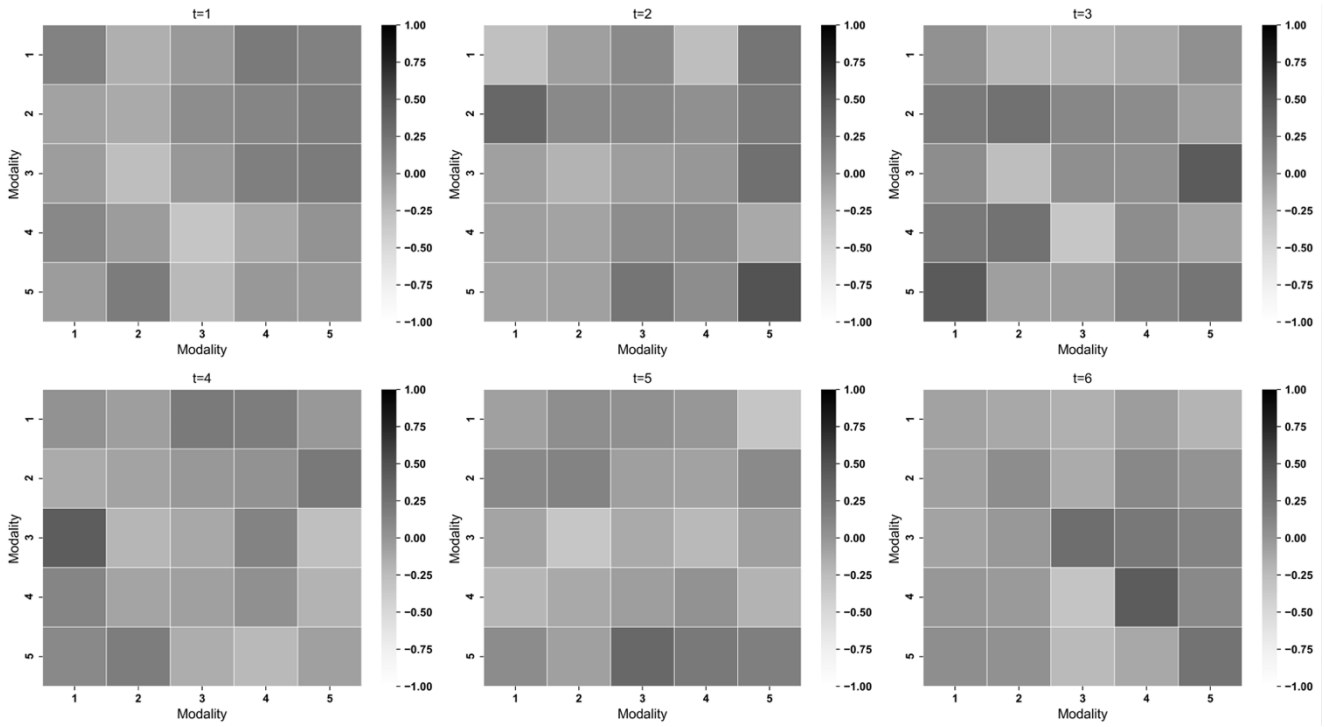


Figure 2.6. Adjacency Weights of the Modality-wise Graph Extracted from Six Time Steps

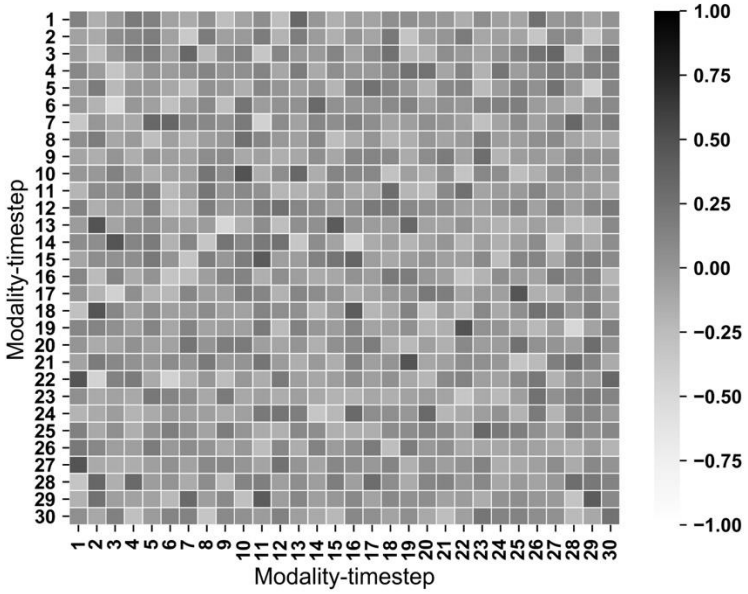


Figure 2.7. Adjacency Weights of the Modality-wise Graph

2.5. Contributions and Implications

This study makes several contributions to the literature. First, *RMGL* is rooted in theoretical foundations regarding GCNs (Kipf and Welling 2016, Schlichtkrull et al. 2017) and designed to model multilevel feature interrelationships in MTS. *RMGL* introduces a novel *meta-graph* to disentangle multilevel feature interrelationships for MTSP. The meta-graph is composed of three interconnected graphs, which simultaneously learn edge weights to represent three unit-level feature interrelationships (i.e., *F2F*, *M2M*, and *T2T*). To account for the binary and ternary superposed feature interrelationships in MTS (i.e., *FM2FM*, *FT2FT*, *MT2MT*, and *FMT2FMT*), *RMGL* incorporates a novel graph attention mechanism that allows the *meta-graph* to learn the superposed interrelationships through interactions among the interconnected graphs. Second, *RMGL* introduces a novel *robust learning objective* to ensure the effectiveness and stability of the learning process. Specifically, given the large number of learning parameters, *RMGL* adopts elementwise L1-norm-based and Frobenius-norm-based regularizations over the three graph adjacency matrices to induce weight sparsity and to prevent overfitting. Furthermore, drawing on the low-pass nature of GCN filters, *RMGL* introduces trace-norm-based regularizations over the low-rank space of the normalized graph Laplacian matrices to learn robust low-frequency components of GCN filters. In addition, *RMGL* introduces L21-norm-based regularization over the residual matrices of the normalized graph Laplacian matrices to optimize row-wise weights of GCN filters by reserving more important row-weight (interrelating) patterns. The proposed regularization scheme ensures that *RMGL* effectively learns multilevel feature interrelationships for MTSP. *RMGL* consistently outperformed state-of-the-art alternatives in the empirical evaluation, providing promising results for MTSP. Although we evaluated *RMGL* with a regression problem, its design can be extended to deal with classification problems too.

The proposed method (i.e., *RMGL*) has practical implications in various fields. For instance, in finance, we have demonstrated that *RMGL* can significantly improve the performance of risk predictions using data collected from multiple channels, including social media and financial platforms. Similarly, *RMGL* can be applied to other financial MTSP tasks, such as fraud detection, stock movement prediction, and market anomaly detection. In video analysis, *RMGL* can be applied to tasks such as emotion detection, sentiment analysis, or person re-identification using visual, acoustic, and textual data extracted from videos in temporal and spatial dimensions (Yang et al. 2020). In urbanization traffic management, *RMGL* can be applied to predict real-time traffic problems using MTS data collected from weather forecasting systems, social media platforms (e.g., Twitter and Facebook), and traffic sensor systems (Semwal et al. 2015). Other examples include pedestrian trajectory detection in autonomous car systems (Mohamed et al. 2020) and personalized recommendations (Tao et al. 2020), among others.

2.6. Conclusion

The big data era has provided unprecedented opportunities for researchers on data-centric studies. In this study, we explore multilevel feature interrelationships to enhance multimodal time-series prediction. We have designed a novel robust meta-graph learning method to disentangle multilevel feature interrelationships and shown its effectiveness in the empirical evaluation with a financial risk prediction task.

Our study has several limitations, which may be addressed in future research. First, although the proposed method can be extended to solve supervised classification problems, in this study, we only evaluated it with a supervised regression problem. The effectiveness of the proposed method for classification needs to be evaluated in future studies. Second, while we demonstrated the effectiveness of the proposed method using acoustic, textual, and numerical modalities,

future studies can examine its effectiveness with other (e.g., visual, video, and network) modalities. Third, while the proposed method falls into the category of supervised learning, future studies can consider extending the method to support unsupervised learning.

3. Essay 3: Measuring Employee Trust: A Deep Learning Approach

3.1. Introduction

Employee Trust Model (ETM), also known as Trust Index©, has been developed to evaluate and rank employee trust and satisfaction among companies (Mitchell 1985; Fulmer et al., 2003). ETM is operationalized through a comprehensive survey that consists of 57 items to address five dimensions of employee satisfaction, i.e., credibility, respect, fairness, pride, and camaraderie. Since employee trust and satisfaction are considered critical drivers of product quality, customer satisfaction, and productivity (Ibrahim et al., 2020), they can influence organizational performance from various aspects, such as leadership development, competitive advantage strengthening, product quality improvement, productivity enhancement, and financial performance optimization. For instance, companies perceived to have a high level of ETM dimensions tend to outperform similar ones in the same industry (Nold, 2012). Worker pride, employer reliability, and camaraderie are sources of competitive advantage (Butler et al., 2016). Respect and credibility are essential capabilities to establish leadership (Duggar, 2009; Verschoor, 2006) because the foundation of leadership requires a careful balance between respect and responsibility (Turknett, 2005). In corporate finance, ETM is also used to study corporate culture, stock value, and portfolio performance (Edmans, 2012; Mishra, 2018; Guiso et al., 2015).

Moreover, employee perception is also linked to environmental, social, and governance (ESG) indices, financial risk, and distress. ESG encompasses factors in the environment, society, and governance that may impact a firm's ability to execute its strategy and enhance corporate value (Jebe 2019). Specifically, the "S" (society) component evaluates whether a company effectively implements social responsibility, including aspects such as human rights, community contributions, labor practices, employment stability, and consumer safety and protection (Kotsantonis et al. 2016).

According to Morgan Stanley Capital International, when calculating ESG indices, employee treatment is taken into consideration, as worker pride, employer reliability, and camaraderie serve as sources of competitive advantage (Butler et al. 2016).

A recent survey of business professionals suggests that employees play a crucial role in a firm's development. Specifically, "talent and skill shortages" were identified as the second most significant risk faced by modern organizations, surpassed only by the risk of "customer loss" and ranking higher than factors like "changing legislation" (Lloyds 2011). Employees who have higher job satisfaction are less likely to turnover. The human capital risk borne by a firm increases with the costliness of replacing employees. Managing this resulting human capital risk is similar to managing other risks outside of risk management, such as debt (e.g., Bolton et al. 2011). To mitigate risk, firms seek to improve employee satisfaction. For instance, workers who feel respected are less likely to turnover, hence reducing human capital risks. Firms allocate resources to enhancing employee satisfaction, just as they invest in research and development, property, plant, and equipment, and organizational capital. Like other forms of investment, expenditures on employees need to be financed through either internal cash flow or externally raised capital. Some of these activities involve direct spending on salaries and pensions. Firms also allocate considerable resources to intangible activities that impact respect and fairness, including the working environment, policies and procedures, training, and supervision. These efforts can help prevent financial distress. However, other theories suggest that firms with better employee perception are less likely to experience financial distress. For example, having a competitive advantage, such as superior human capital, is a key factor explaining why some firms outperform others (Acedo et al. 2006; Barney et al. 2001;). Both individuals and society can benefit

economically by investing in people, such as through investments in education and respect (Sweetland 1996). Firms with a competitive advantage are less likely to face financial distress.

Existing measurement of employee trust and satisfaction requires members of an organization to complete a 10- to 15-minute survey anonymously. This makes it difficult to collect large samples of data over time (Chatman 2016). The use of small-size dataset has led to conflicting results in managerial and finance research (Iaffaldano & Muchinsky 1985; Edmans 2011), making the findings less appealing to practitioners. Furthermore, the absence of data in the time dimension has restricted analytical methods in use and limited the application of theoretical frameworks. In the meanwhile, the proliferation of digital platforms has provided researchers access to large amounts of secondary data, including employee-generated reviews that cover a wide range of organizations and time, providing opportunities for innovative methods to fill in the research gap.

In this study, we propose DeepEmployee, a novel design artifact based on automated text classification, to detect employee trust and satisfaction from employee-generated reviews. DeepEmployee stems from design science research (Hevner et al. 2004) and includes three cohesive and complementary parts: (1) domain-specific knowledge construction based on theoretical frameworks in the management field, (2) a state-of-the-art deep learning design artifact that incorporates domain-specific knowledge to improve performance, and (3) a rigorous two-part evaluation of improvements in ETM detection and increased predictive power of the derived variables.

The contribution of DeepEmployee is three-fold. First, DeepEmployee uses a new technical means to enrich managerial theories and extends their applications to various fields in

real-world settings. Second, DeepEmployee is among the first deep learning design artifacts to incorporate domain-specific knowledge for text classification with employee-generated reviews. Specifically, DeepEmployee includes three major components in its design: (1) an embedding model named EmpBERT, which transfers contextual word embeddings (Devlin et al. 2019) from the public text domain to employee-generated reviews, (2) a knowledge representation module that represents structural relationships, and (3) a triple attention mechanism, namely phrase-set semantic attention, transformer-based self-attention, and structural attention, to dynamically learn important feature weights for joint classification. Lastly, through a rigorous evaluation, we show that DeepEmployee outperforms baseline and state-of-the-art learning methods, and the more accurate detection can lead to higher predictive power in various downstream tasks.

Our work has important implications for information systems research. We show that new information technologies, natural language processing (NLP) in particular, have provided critical opportunities for research to enable the long-standing measurement of culture and management constructs with digital data logs, thereby contributing to computational management science. Furthermore, our work has practical implications for managers to facilitate their decision-making by including employee trust in a broader analytical context.

3.2. Background

3.2.1 ETM and organizational performance

ETM has commonly been used as a measurement method for employee trust and satisfaction in management and business decision-making studies through five dimensions: *credibility*, *respect*, *fairness*, *pride*, and *camaraderie*. It has a strong impact on organizational performance. Employee trust and satisfaction represent how employees feel about their working environment and can provide an indication of employees' emotional well-being (Spector,

1997). Among ETM dimensions, leadership with a high degree of credibility can enhance organizational performance (Williams et al., 2018). Respect is the way an organization treats its employees. The relationship between organizational justice and performance is moderated by organizational respect (Saboor et al., 2018). Furthermore, scholars suggest that organizational performance should include not only internal and external dimensions of efficiency and effectiveness but also fairness (Brewer and Selden, 2000). Managers must ensure that all employees are treated fairly and with respect. Employees get pride from knowing that they are doing something right. Pride has a positive impact on employee behavior, which consequently affects organizational performance (Gouthier and Rhein, 2011). Employee loyalty, trustworthiness, and camaraderie have been confirmed as factors contributing to competitive advantage and organizational performance (Butler et al., 2016).

3.2.3 Automated NLP-based ETM Detection

Employee-generated reviews encapsulate nuances of ETM (Swain et al. 2020). For instance, the review, "long hours, low direction provided by management" encapsulates two ETM dimensions. First, the mentioning of "long hours" refers to the reviewer's experiences with management's care for employees in terms of work-life balance and can be matched to the measurement item "people are encouraged to balance their work life and their personal life" under the "respect" dimension of ETM. Second, "low direction provided by management" indicates poor management communication and can be matched to the measurement item "management keeps me informed about important issues and changes" under the "credibility" dimension of ETM. In another case, the review "great team environment and good pay" has indicated that the organization's management style is characterized by teamwork and therefore can be categorized

as the clan culture type. In addition, the “team environment” and “good pay” are related to the ETM dimensions of camaraderie and fairness.

Accordingly, ETM detection consists of assigning multiple target labels to employee-generated reviews and therefore can be categorized as a multi-label text classification problem. Multi-label text classification is a fine-grained approach and a generalization of binary and multi-class text classification. Prior research has mainly used (1) problem transformation and (2) algorithm adaption to deal with multi-label classification problems. Problem transformation first transforms a multi-label classification problem into multiple binary or multi-class classification problems (e.g., *binary relevance* (Boutell et al. 2004), *label ranking* (Fürnkranz et al. 2008), and *label powerset* (Tsoumakas et al., 2011)) through data manipulation and then builds independent or chain of classifiers (Osojnik et al., 2017) to deal with classification problems. Algorithm adaption, on the other hand, adjusts existing machine learning algorithms, such as k-nearest neighbor, decision tree, support vector machine, and neural networks, to deal with multi-label problems. Examples of adapted algorithms are ML-kNN (multi-label k-Nearest Neighbor, Zhang 2007), MMAC (multi-class, multi-label associative classification, Thabtah et al., 2004), MLNB (multi-label naïve Bayes, Zhang, et al., 2009), RankSVM (Elisseeff et al., 2001), and MLTSVM (Chen et al, 2018).

Recent studies have fed linguistic features into end-to-end learning methods and obtained state-of-the-art performance in a variety of text classification tasks (Kim et al 2014; Devlin et al. 2019).

2.3 Feature Representations in Text Classification Tasks

Feature representation is of central importance to the performance of an end-to-end learning method, in which the model learns all parameters in a single processing step. The representation of text features can be broadly categorized as shallow and deep linguistic features. Shallow linguistic features, such as tfidf, one-hot embeddings, and n-grams, often ignore contextual information and word order in texts and are short of capturing the semantics of words. In addition, shallow linguistic features are subject to a feature selection process due to the data sparsity problem. On the other hand, deep linguistic features employing word embeddings and language models have shifted the paradigm for learning methods to incorporate meaningful linguistic features while greatly alleviating the data sparsity problem. For example, word embeddings (e.g., Word2Vec and GloVe (Mikolov et al., 2013; Pennington et al., 2014)) can capture meaningful semantic and syntactic regularities through training a deep neural network on large corpora of general-domain texts. Recent advances in language models have switched the focus to learning contextual representations. For instance, CoVe (Context Vectors, McCann et al. 2017) contextualizes word representations through a deep LSTM network trained on machine translation datasets and has achieved better performance than GloVe on various tasks. ELMo（deep contextualized word representations, Peters et al. (2018) uses Bi-LSTM networks and concatenates the outputs from Bi-LSTM to encode contextual information from both directions of input sequences. BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. (2019) uses Masked Language Model (MLM), which renders text representations the predictive power for randomly masked words in input sequences. When coupled with an end-to-end learning method, deep linguistic features have shown effectiveness in various text classification tasks. For instance, Kim et al. (2014) used pretrained Word2Vect embeddings as input features to a convolutional neural network (CNN) and achieved good results in a variety of sentence

classification tasks. In another case, Devlin et al. (2019) used a transformer-based deep learning network with BERT embeddings and obtained state-of-the-art performance in various NLP tasks. One limitation of end-to-end learning methods is the insufficient representation of domain knowledge (Marcus and Davis 2019, Marcus 2020, Nie et al. 2019, Jin et al. 2019). Specifically, deep linguistic features are commonly transferred from general-domain texts and do not capture the word distribution shift in the target domain. Therefore, when applied to domain-specific tasks, it often yields unsatisfactory results（Lee et al. 2019, Araci et al. 2019). Another reason is the lack of integration methods such that domain knowledge represented in end-to-end learning artifacts mainly pertains to opaque feature correlations (Marcus 2018), rather than abstractions like quantified statements.

In the meanwhile, rule-based methods, a different approach for automated text classification tasks, use hand-crafted features and rules to represent prior linguistic and domain-specific knowledge. For instance, in classifying consumer opinions from product reviews, Hu and Liu (2004) first extracted frequent terms of product features through part-of-speech tagging and association rule mining, then identified opinion words from frequent terms through syntactic patterns, and finally extracted infrequent terms of product features through a predefined set of rules. In another case, while classifying sentiments in financial texts, Chan and Chong (2017) used an ensemble method (meta-level decision tree) to build a linguistic constituent parser tree and proposed heuristic rules to propagate the sentiments from the leaves to the root node of the parser tree to derive an overall sentiment for a sentence. Leveraging prior knowledge, rule-based methods have served as vital approaches for text classification tasks when labeled corpus and computing resources are limited.

2.4 Attention Mechanism in Text Classification Tasks

First derived from human intuition and then adapted to machine translation for automatic token alignment, *attention* mechanism, a dynamic weight adjustment process of input features, has been widely applied to and attained significant improvement in various NLP tasks. In most cases, attention mechanisms can be formulated as

$$\text{Attended Input} = f_w(g_w(H), H),$$

where H represents hidden states of input features before the attention layer, g_w and f_w are parametric functions to be learned; g_w is an attention function to compute feature weights, and f_w is a fusion function to compute the attended input. A variety of choices on the functional form of g_w and f_w have led to diverse attention mechanisms. For instance, in self-attention (also referred to as Scaled Dot-Product Attention (Vaswani 2017)), H is linearly transformed to Q, K, and V, namely the query, key, and value matrix; g_w is a *softmax* function on the scaled inner product of Q and K; and f_w is an elementwise multiplication between g_w and V. In another case, addictive attention (Bahdanau & Bengio 2015) uses a one-hidden layer feed-forward network to calculate the attention alignment score. Other attention mechanisms include hierarchical attention (Yang et al. 2016; Ji et al. 2017), multi-scale multi-head attention (Zhang et al. 2019), and memory-based attention (Kumar et al. 2016), among others. In text classification tasks, attention mechanisms have been bundled with various deep learning artifacts, including RNN (Wang et al. 2016), CNN (Du et al. 2018), and transformer-based networks (Devlin et al. 2019), to learn weighting scores for input features, expecting key features to receive heavier weights, and weights of the features contribute to the classification task.

3.3. Proposed Method

We propose DeepEmployee, a deep-learning design artifact for ETM detection based on employee reviews. Extant automatic text classification methods can be broadly grouped into two

categories: (1) rule-based and (2) end-to-end learning methods. Rule-based methods represent prior knowledge explicitly through a set of hand-engineered features and rules, while it is difficult to ensure quality with manual rule selection and pruning (Liu et al. 2015). The applicability of rule-based methods is further limited because of the complex linguistic forms and information loss during the feature engineering process. On the other hand, end-to-end machine learning methods, which infer the correlations between input features and the output results based on the inherent associations embedded in data observations, are considered self-contained and isolated from potential useful prior knowledge (Marcus et al. 2018). In face of representational richness as an essential but challenging goal for learning models, DeepEmployee incorporates the following components in its design artifact:

(1) Prior knowledge construction is based on the central thesis of ETM.

(2) Feature representation including contextualized word representation and prior knowledge.

(3) Feature weighting and joint classification

We believe that incorporating prior knowledge of the central thesis of ETM into the learning artifact can benefit from the complementary strengths of both rule-based and end-to-end learning methods. Accordingly, we present the major components of our design artifact and compare its performance with various state-of-the-art methods.

3.3.1 Knowledge Construction

Prior studies have used survey items to operationalize the central thesis of ETM. An ETM survey consists of 57 items to address five dimensions of employee trust, i.e., credibility, respect, fairness, pride, and camaraderie. Each dimension of employee trust has its corresponding items to

assess fine-grained meanings under that dimension. For instance, the credibility dimension contains three sub-dimensions–communication, competence, and integrity–of employee trust in the management team, and in total, there are fourteen survey items to measure the three sub-dimensions in finer granularity. For example, the first four survey items– "management keeps me informed about important issues and changes", "management makes its expectations clear", "I can ask management any reasonable question and get a straight answer", and "management is approachable, easy to talk with"–are used to assess the informativeness and accessibility of the communication between employees and the management team.

To collect domain knowledge, we first identify dimensional structures based on theoretical frameworks of ETM (Great Place To Work® 2017; Edman 2012; Cameron and Quinn 1999;). We then match the survey items to the dimensions and sub-dimensions they measure. Next, for each survey item, we identify the key terms and phrases to be focused on during the labeling process.

3.3.2 Feature Representation

3.3.2.1 Contextualized Word Representation

DeepEmployee uses embeddings of BERT, a contextualized word representation technique, as the feature representation base for three main reasons. First, BERT embeddings have been used in transformer-based classifiers and obtained state-of-the-art performance in a variety of NLP tasks (Devlin et al. 2019). Second, features such as position embedding and MLM render BERT embeddings with bidirectional information flow to the target word simultaneously, unlike other contextualized representations that either consider the information flow from only one direction or use a shallow concatenation of the bidirectional context (e.g., ELMo). Third, BERT embeddings include both sentence-level and word-level representations, fitting our task of representing prior knowledge of the central thesis of ETM (see 3.2.2 and 3.2.3).

However, directly applying pre-trained BERT embeddings on new domain-specific tasks often yields unsatisfactory results due to word distribution shifts in the target domain. We observed that the texts from employee reviews are often terse and could imply a meaning different from the common interpretation. For instance, the sentence "not sure how to move up the ladder" in the context of employee reviews implies that the career path is unclear to the employee, which touches upon the fairness dimension of employee trust, and the implicit meaning is different from the direct interpretation of having no knowledge of climbing a ladder. Research has shown that BERT embeddings further trained with domain-specific corpora can improve performance. For instance, Lee et al. (Lee et al.) retrained BERT embeddings with a large-scale biomedical corpus, namely BioBERT, to understand complex biomedical texts. BioBERT outperformed previous state-of-the-art models on three representative biomedical text-mining tasks. In another case, Araci et al. (2017) introduced FinBERT, a BERT model retrained with a financial corpus including 29 million words and 400 thousand sentences. FinBERT obtained state-of-the-art performance in two financial sentiment analysis tasks.

In response to the domain adaption issue, we build EmpBERT, a BERT model further retrained with an employee-review corpus, to improve representations of employee-review texts. The domain-specific corpus contains large-scale employee-generated texts, enabling EmpBert to shift the word distribution to the specific domain.

DeepEmployee detects ETM at the review sentence level. An employee review can contain multiple sentences. We use *spacey*, a python NLP package, to split each review into sentences. For each review sentence, we tokenize it, add symbols *CLS* and *EOS* at the beginning and end of the review sentence, and then input the tokens to EmpBERT to obtain both sentence- and word-level

representations for that sentence. Let $S_i = (CLS, s_{i1}, \dots, s_{ij}, \dots, s_{iL_s}, EOS)$ denote the tokens of the i-th review sentence, where s_{ij} is the token for the j-th word in the sentence. Define

$$SR_i = \left(SR_{i1}, \dots, SR_{ij}, \dots, SR_{iL_s}\right) = EmpBERT(S_i), \quad (3.1)$$

where $EmpBERT\ (.)$ is the EmpBERT function that maps $\mathbb{R}^{L_{S_j}}$ to the embedding space $\mathbb{R}^{L_{S_j} \times \mathrm{d}}$, and SR_{i1} and SR_{iL_s}(the first and last in SR_i) are d-dimensional contextualized representations at the sentence level, and SR_{ij} is a d-dimensional contextualized representation for the j-th word in the sentence.

3.3.2.2 Knowledge Representation

DeepEmployee aims to learn a domain knowledge representation and integrate it with contextualized word representation of a review sentence to enhance the accuracy of ETM detection from the sentence. To this end, we break down the constructed domain knowledge into two categories and apply separate schemes for the representation: (1) a dimensional structure and phrase associations, which serve as a guideline for the labeling process, and (2) structural patterns of cooccurrence of dimensions and patterns of the cooccurrence of dimension and n-grams.

The dimensional structure of ETM and the association between dimensions and phrases can be considered as rules that contain a set of phrases so that if a sentence is semantically close to any of the phrases in the set then the sentence can be categorized as the dimension associated with the set of phrases. For each phrase, we tokenize it, add symbols *CLS* and *EOS* at the beginning and end of the phrase, and then input the tokens to EmpBERT to obtain both phrase- and word-level representations for that phrase. Formally, let $P_i = \{P_{ij}\}$ be the i-th phrase set governing dimension D_i, where P_{ij} is the j-th phrase in P_i, and $P_{ij} = \{CLS, p_{ij1}, \dots p_{ijk} \dots, p_{ijN_{R_{ij}}}, EOS\}$ where p_{ijk} is the token for the k-th word in the sentence. Define

$$PR_{ij} = \left(pr_{ij1}, \dots pr_{ijk} \dots, pr_{ijN_{R_{ij}}}\right) = EmpBERT(P_{ij}) \,,$$

(3.2)

where pr_{ij1} and $pr_{ijN_{R_{ij}}}$ are d-dimensional contextualized representations at the phrase level, and pr_{ijk} is a d-dimensional contextualized representation for the k-th word in phrase *j* and phrase set *i*.

To capture the structural pattern among dimensions, and dimension and n-grams, we apply a *graph neural network* (GNN), an effective approach to model structural relationships. Formally, consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{D})$, where $\mathcal{V}$ is a set of nodes including dimensions and n-grams; $\mathcal{E}$ is a set of edges between dimension to dimension and dimension to n-grams; $\mathcal{A}$ is the adjacency matrix built based on the PMI (Pointwise Mutual Information) function from the labeled corpus, where

$$\mathcal{A}_{ij} = \begin{cases} log\frac{p(v_i,v_j)}{p(v_i)p(v_j)} & if\ one\ or\ both\ of\ v_i\ and\ v_j\ are\ dimensions \\ 1 & if\ i = j\ and\ \ both\ v_i\ and\ v_j\ are\ dimensions \\ 0 & otherwise \end{cases}. \quad (3.3)$$

$\mathcal{D}$ is the degree matrix. We adopt the propagation rule of *graph convolutional network* (GCN, Kipf and Welling 2017):

$$H_{l+1}^{rel} = \sigma(\mathcal{D}^{-\frac{1}{2}}.\mathcal{A}.\mathcal{D}^{-\frac{1}{2}}.H_l^{rel}.W_{gl}^{rel}), \quad (3.4)$$

where W_{gl}^{rel} is the learnable weight matrix of the *l*-th layer, $H_l^{rel} \in \mathbb{R}^{N_v \times d_{gcn}}$ is the hidden state of the *l*-th layer, N_v is the number of nodes in the network, and d_{gcn} is the dimensionality of the network's hidden state. σ is the activation function. We use a 2-layer GCN whose nodes include dimensions of ETM and n-grams from labeled sentences. Specifically, we acquire unigram, bigram,

and 3-gram by removing stop words and then selecting the top (say, 100) tf-idf weighted n-grams from each dimension. We use one-hot encoding as the initial state of the nodes and ReLU as the activation function of the 2-layer GCN. Hence, the representation of a dimension can be written as follows.

$$H_i = \sigma\left(\mathcal{D}^{-\frac{1}{2}}.\mathcal{A}.\mathcal{D}^{-\frac{1}{2}}.\sigma\left(\mathcal{D}^{-\frac{1}{2}}.\mathcal{A}.\mathcal{D}^{-\frac{1}{2}}.X.W_{g1}^{rel}\right).W_{g2}^{rel}\right)[i] \quad (3.5)$$

3.3.3 Feature Weighting and Joint Classification

The feature representation process generates three groups of features, i.e., (1) the contextualized representations at the sentence and word levels, (2) the contextualized representations of structural dimension at the phrase and word level, and (3) the hidden state of the structural pattern between dimension and dimension, and dimension and n-grams. To apply the features for ETM detection, DeepEmployee uses a unique triple-attention-based framework for feature weighting and joint classification. Specifically, DeepEmployee adopts *scaled dot product attention* to compute a similarity score between features in the respective space. Intuitively, the triple attention mechanism serves as three different score functions to evaluate the dimension scores from three perspectives and then applies the scores to a joint classification.

The first attention mechanism, phrase-set semantic attention, is used to compute semantic/syntactic/concept relevance between phrases and sentence. First, for each phrase, an attended feature representation is acquired through the scaled dot product attention between the contextualized phrase representation, which serves as a query, and the contextualized sentence representation, which serves as an information base to be queried from. Next, the phrase-level attended representations of the phrases in the same phrase set are concatenated and fed into a feed-forward network to learn a semantic relevance score of the ETM dimension associated with the phrase set. Figure 3.1 shows the process of phrase set attention mechanism.

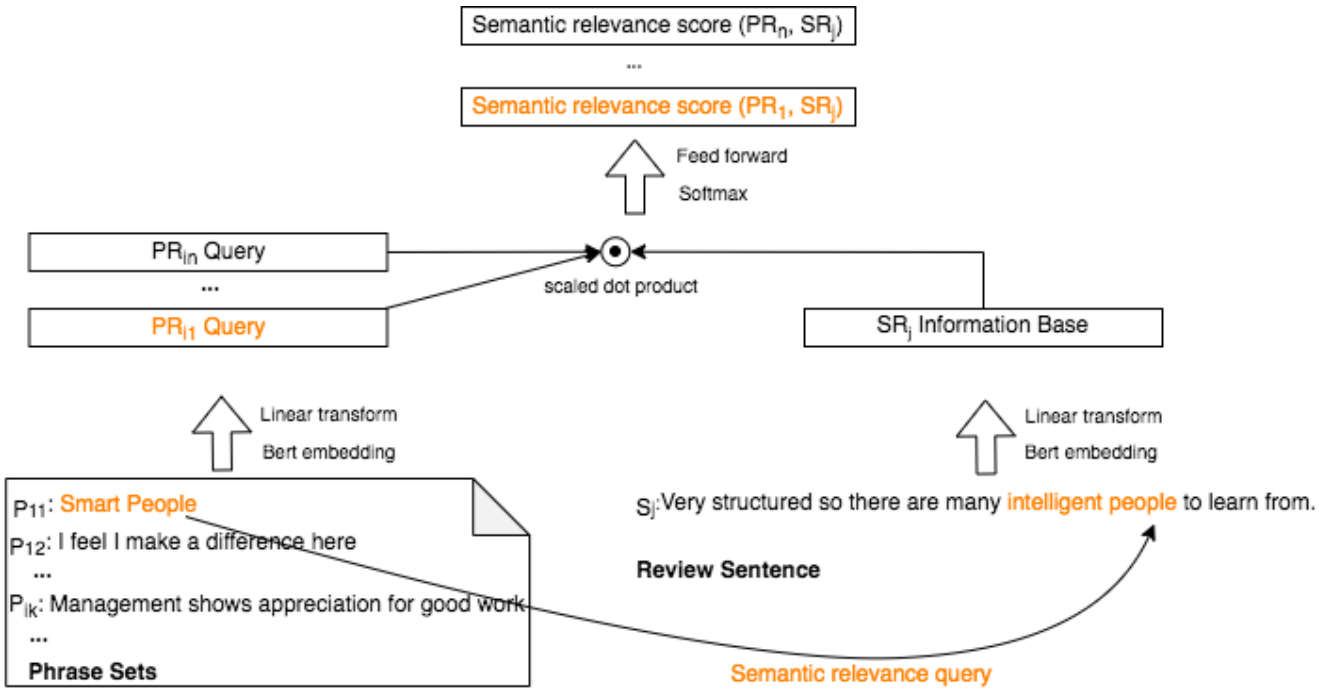


Figure 3.1. Phase-set attention

The second attention mechanism, the transformer-based self-attention, is used to compute contextual relevance between phrases and sentence. First, DeepEmployee adds position embeddings from the transformer network to the contextualized sentence representations. Next, self-attention is used to learn the weights of contextualized word representations in a sentence. Finally, the attended sentence representation is used in a feedforward network to learn a contextual relevance score for each ETM dimension. The third attention mechanism, structural attention, is used to compute structural relevance between the hidden states of ETM dimensions and a sentence. The scaled dot product attention is used to compute the attended structural representation of ETM dimensions, and then the attended structural representation is fed into a feedforward network to learn a structural relevance score for each ETM dimension. Finally, for each ETM dimension, its semantic relevance score, contextual relevance score, and structural relevance score are

concatenated and input to a feedforward network for the joint classification. Figure 3.2 shows the major components of DeepEmployee.

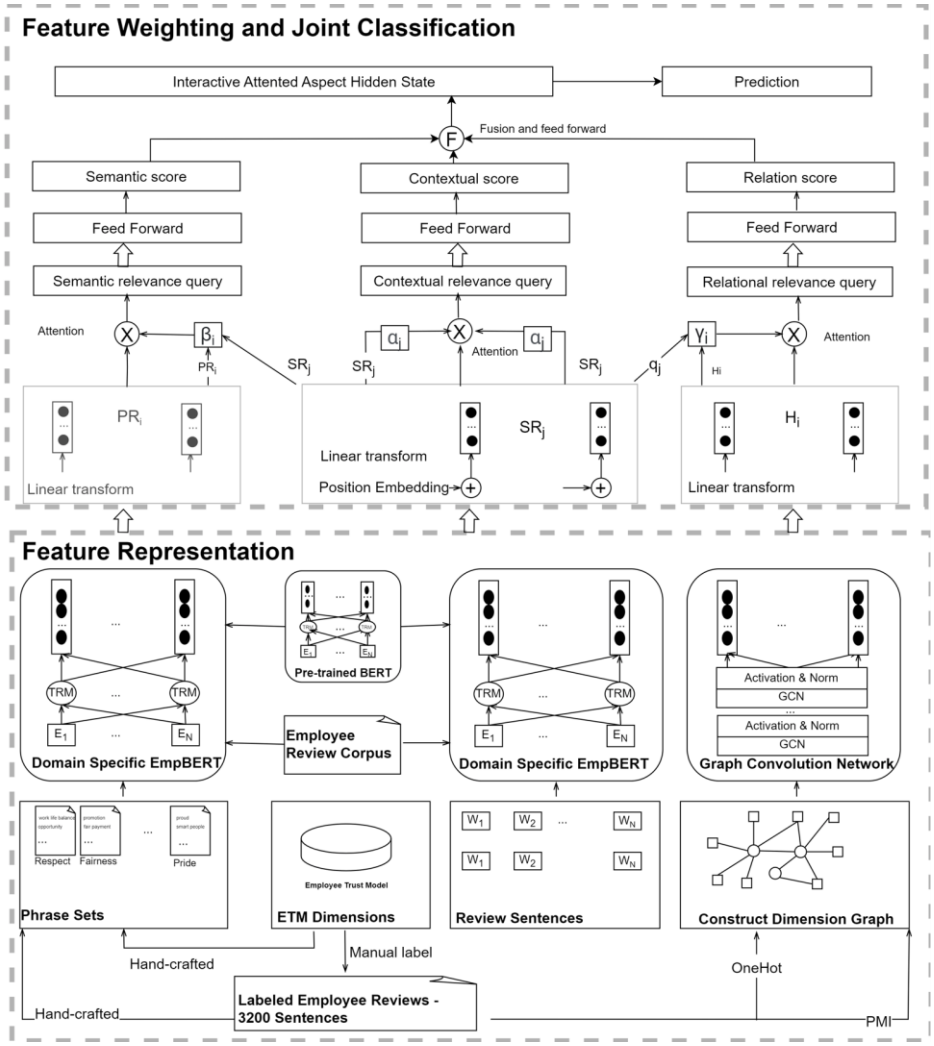


Figure 3.2. Design Overview

3.4. Empirical Evaluation

3.4.1 Data

We have collected a dataset containing employee-generated reviews of Standard & Poor's 1500 companies (S&P1500), spanning the period from 2009 to 2020, to evaluate the proposed method. On the employee review platform, when an anonymous user posts a review for a company, the user must write comments on the pros and cons of the company under a separate text area. The

domain-specific corpus contains large-scale employee-review sentences and words (over 9 million sentences and 90 million words) and covers employee reviews of S&P1500 companies from the year 2009 to 2020. We retrained EmpBERT with the employee-review corpus using an NVIDIA GPU with 10GB memory in three epochs for two weeks.

To generate a set of training and testing data, we obtained the pros and cons of reviews and split them into sentences. Out of the over 9 million review sentences, we sampled 3,615 sentences for manual labeling. Two researchers of this study manually and independently labeled the employee review sentences. We randomly sampled 3,615 review sentences from the full dataset for labeling. If a review sentence explicitly or implicitly conveys similar meanings represented by the phrases, the sentence will be labeled with the corresponding dimensions matched by the phrases. In the end, the dataset contains 3,615 review sentences, and 2,433 of them have at least one label. Table 3.1 presents the summary statistics of the ETM labels.

ETM	**respect**	**fairness**	**credibility**	**camaraderie**	**pride**
Number of labels	1019	409	446	348	414

Table 3.1. Summary Statistics of ETM labels

3.4.2 Experiment

We conducted three experiments to evaluate *DeepEmployee* in the following aspects:

1) to compare *DeepEmployee* against several benchmarks in terms of prediction performance,

2) to validate and gain further insights into the design choices of *DeepEmployee* with an ablation study comparing *DeepEmployee* against several ablated variants,

3) to assess the explanatory power of the derived ETM variables in a downstream explanatory task.

In experiment 1, we compared *DeepEmployee* against nine benchmarks selected from the multi-label text classification literature. We first included k-Nearest Neighbors (kNN), support vector machine (SVM), XGBoost (XGB), and random forest (RF) using the binary relevance transform for multi-label classification. We then included a support vector machine using the label powerset transform. Finally, we included Convolutional Neural Network (CNN), CNN+attention, Long Short-Term Memory (LSTM), bi-LSTM, and LSTM+attention for multi-label classification from the deep learning literature. For each method, we estimated the prediction performance, in terms of (micro) F1 and (micro) AUC, through 10 independent runs of 10-fold cross validation. Table 3.2 summarizes the result, which shows that *DeepEmployee* outperformed the benchmarks in terms of both F1 and AUC.

Method	**Respect**	**Fairness**	**Credible**	**Camaraderie**	**Pride**	**F1**	**AUC**
br-kNN	0.418	0.180	0.065	0.427	0.290	0.315	0.738
br-Svm	0.644	0.619	0.445	0.732	0.460	0.610	0.896
br-xgb	0.672	0.696	0.548	0.748	0.518	0.661	0.914
br-RF	0.402	0.339	0.251	0.309	0.125	0.352	0.673
lp-RF	0.318	0.197	0.164	0.252	0.117	0.275	0.694
cnn	0.697	0.675	0.499	0.762	0.480	0.663	0.913
cnn+attention	0.649	0.689	0.496	0.762	0.496	0.648	0.902
lstm	0.691	0.678	0.547	0.778	0.527	0.670	0.914
bi-lstm	0.687	0.687	0.533	0.774	0.534	0.671	0.914
lstm+attention	0.692	0.687	0.549	0.789	0.530	0.677	0.914
DeepEmployee	**0.737**	**0.745**	**0.609**	**0.829**	**0.648**	**0.718**	**0.933**

Table 3.2. Comparison of Detection Performance

In experiment 2, we conducted an ablation study to investigate the utilities of the design components introduced in *DeepEmployee*. Specifically, we compared *DeepEmployee* with its three variants, i.e., *(1) DeepEmployee* with only the knowledge representation module, *(2) DeepEmployee* with only structural relation, *(3) DeepEmployee* with only self-attention module, *(4) DeepEmployee* with only self-attention module and domain-specific BERT. Table 3.3 summarizes the result. The result of experiment 2 indicates that *DeepEmployee* achieved the best performance when all design components were combined.

Method	**F1**	**AUC**
Knowledge Representation	0.598	0.873
Structural Relation	0.510	0.851
Self-Attention	0.690	0.925
Self-Attention + Domain-specific BERT	0.693	0.927
DeepEmployee	**0.718**	**0.933**

Table 3.3. Ablation of DeepEmployee

In experiment 3, we obtained employee review ratings and texts from Glassdoor, quarterly accounting data from CompStat, and stock return data from CRSP for explaining financial risks with ETM indices. First, we calculated the mean of overall rating from Glassdoor for each quarter as a proxy of employee satisfaction. We used DeepEmployee for ETM detection and calculated the following ETM indices

$$camaraderie\ ratio = \frac{positive\ \text{camaraderie} - negative\ \text{camaraderie}}{review\ counts},$$

$$credibility\ ratio = \frac{positive\ \text{credibility} - negative\ \text{credibility}}{review\ counts},$$

$$fairness\ ratio = \frac{positive\ fairness - negative\ fairness}{review\ counts},$$

$$pride\ ratio = \frac{positive\ \text{pride}\ - negative\ \text{pride}}{review\ counts},$$

$$respect\ ratio = \frac{positive\ \text{respect} - negative\ \text{respect}}{review\ counts}.$$

We followed the methodology outlined by Ang et al. (2006), Edmans (2012), and Green et al. (2018) to control for firm size, book to market, ROA, and average ratings. In terms of risk measurement, we use volatility around the announcement period. Specifically, we calculated the standard deviation of stock returns in the subsequent 3, 5, and 7 trading days using the CRSP return data. Table 3.4 reports summary statistics of the data.

	Mean	Std. Dev.	Median	Min	Max
Risk [0,3]	0.021	0.022	.015	0	.999
Risk [0,5]	0.021	0.020	.016	0	.804
Risk [0,7]	.021	0.019	.016	0	1.081
Average rating	3.27	0.793	3.32	1	5
ROA	0.012	0.039	0.011	-2.319	2.402
Size	8.402	1.760	8.289	-.021	15.139
Book to market	2.321	10.262	1.006	0.006	1939.120
Camaraderie ratio	0.177	0.258	0.127	-2	4
Credibility ratio	-0.339	0.616	-0.250	-14	4
Fairness ratio	-0.133	0.462	-.056	-8	6
Respect ratio	0.224	0.642	0.235	-15	12
Pride ratio	0.266	0.359	0.211	-4	6

Table 3.4. Summary Statistics of Financial Risks Regression Variables

We estimated a regression model to explain the impact of ETM indices on the announcement volatility as a proxy of firm-level financial risks,

$$Risk = \beta_1 Average\ ratings + \beta_2 Cameradirie\ ratio + \beta_3 Credibility\ ratio + \beta_4 Fairness\ ratio + \beta_5 Respect\ ratio + \beta_6 Pride\ ratio + \beta_7 ROA + \beta_8 Size + \beta_9 B/M + \mu_i + \mu_t + \varepsilon_{it}$$

μ_i represents industry fixed effect and μ_t represents year fixed effect. To avoid unobservable industry differences and year shock, we also controlled for industry fixed effects and year fixed effects. The results (in Table 3.5) show that the coefficient for respect was negative and significant ($p<0.01$) for all three risk measurements, indicating that firms that manage their employees with respect can reduce financial risks. Respect can also enhance employee performance. Additionally, employees are more likely to remain with an organization when they feel respected. Interestingly, the coefficient for fairness ratio was positive and significant ($p<0.01$) for all three risk measurements. Compensation fairness, to some extent, can increase firm risks. First, struggles in meeting the competing demands of customers and managers can arise, leading to collusion between customers and employees against the firm's interests (Eddleston et al. 2002). Second, when compensation is linked to revenue, employees may be motivated to provide poor service to customers perceived as poor tippers, which can result in lawsuits and the loss of business from discriminated customers (Lynn 2004).

	Risk (3 day)	**Risk (5 day)**	**Risk (7 day)**
ROA	-.052*** (0.009)	-0.0534 *** (0.008)	-0.054 *** (0.008)
Firm size	-0.002*** (0.000)	-0.00161*** (0.000)	-0.002 *** (0.000)
B/M	0.001*** (0.000)	0.001 *** (0.000)	0.001 *** (0.000)
Rate mean	-0.001 (0.000)	-0.002+ (0.000)	-0.0004 (0.000)
Respect ratio	-0.008** (0.003)	-0.007** (0.002)	-0.007*** (0.002)
Fairness ratio	0.0127** (0.00435)	0.011 ** (0.004)	0.011** (0.003)
Credibility ratio	-0.0110* (0.00553)	-0.007 (0.005)	-0.00797+ (0.00455)
Camaraderie ratio	0.0161** (0.00553)	0.0148 ** (0.00468)	0.012** (0.004)

Pride ratio	0.001 (0.004)	-0.00153 (0.00317)	-0.002 (0.002)
Industry fixed effect	Yes	Yes	Yes
Year fixed effect	Yes	Yes	Yes
*Note: *** p < 0.001, ** p < 0.01, * p < 0.05, + p < 0.01*			

Table 3.5. Regression Result of Financial Risks on Employee Trust Model Indices

The coefficient for camaraderie was positive and significant ($p<0.01$) for all three risk measurements, indicating that a higher camaraderie ratio can increase firm risk during the announcement period. Firms with high camaraderie ratings resemble family firms, which maintain relationships based on family kinship. Family firms face additional instability due to the kinship sphere. Camaraderie relationships can also lead to an increase in organizational "entropy" and thus increase uncertainty (Coli 2013).

The results are important for firm policymakers to understand which aspects are crucial in internal risk management. Instead of relying on the abstract concept of employee satisfaction, we provide empirical evidence on which areas should be improved. For instance, firms should provide employees with positive feedback and align their work assignments with their skills to increase the respect rate and decrease risks. A novel finding we document is that camaraderie can increase risks.

3.5. Discussion

This study makes several contributions to the literature. Firstly, DeepEmployee uses innovative technical methods to enrich managerial theories and extend their applications to various real-world fields. By incorporating deep learning techniques, DeepEmployee enables the analysis of employee-generated reviews, contributing to a deeper understanding of employee experiences and sentiments. Secondly, DeepEmployee stands as one of the pioneering deep learning design artifacts that incorporate domain-specific knowledge for text classification in the context of employee-generated reviews. Its design comprises three major components: (1) EmpBERT, an

embedding model that transfers contextual word embeddings from the public text domain to employee-generated reviews, leveraging advancements like Devlin et al.'s (2019) work, (2) a knowledge representation module that captures structural relationships within the data, and (3) a triple attention mechanism, consisting of phrase-set semantic attention, transformer-based self-attention, and structural attention. These components work together to dynamically learn important feature weights for joint classification. Lastly, through rigorous evaluation, we demonstrate that DeepEmployee outperformed baseline and state-of-the-art learning methods. The improved detection accuracy provided by DeepEmployee contributes to higher predictive power in various downstream tasks. This underscores its potential value in practical applications.

3.6.Conclusion

Our work carries significant implications for information systems research. We showcase how new information technologies, specifically NLP, open critical opportunities for measuring culture and management constructs using digital data logs. This contributes to the emerging field of computational management science. Furthermore, our research has practical implications for managers, offering a broader analytical context that includes employee trust and satifaction. By considering these factors in decision-making processes, managers can enhance their understanding of the organizational climate and make more informed decisions.

Our study has several limitations that can be addressed in future research. Firstly, the detection model relies on manual labeling work, which necessitates the expertise of domain specialists and human labelers. Finding ways to reduce or automate the labeling process could enhance efficiency. Secondly, pretraining the EmpBert model demands a substantial amount of text data and computing resources. Exploring strategies to optimize the pretraining phase, such as using transfer learning techniques or leveraging smaller-scale resources, may facilitate broader adoption of the

model. Thirdly, although we demonstrated the effectiveness of the proposed method using supervised learning algorithms, future studies could delve into the efficacy of unsupervised or semi-supervised learning algorithms for detection purposes. Examining alternative learning approaches may provide valuable insights into the detection of employee sentiments. Addressing these limitations in future research will contribute to refining and expanding the scope of our findings, ultimately enhancing the applicability and robustness of the proposed method.

REFERENCES

Acedo, F. J., Barroso, C., & Galan, J. L. (2006). The resource‐based theory: dissemination and main trends. Strategic management journal, 27(7), 621-636.

Akbari H, Yuan L, Qian R, Chuang WH, Chang SF, Cui Y, Gong B (2021) VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text. Advances in Neural Information Processing Systems 34.

Armanious, K., et al., MedGAN: Medical image translation using GANs. Computerized Medical Imaging and Graphics, 2020. 79.

Badrinarayanan, V., A. Kendall, and R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. Ieee Transactions on Pattern Analysis and Machine Intelligence, 2017. 39(12): p. 2481-2495.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Barney, J., Wright, M., & Ketchen Jr, D. J. (2001). The resource-based view of the firm: Ten years after 1991. Journal of management, 27(6), 625-641.

Bolton, P., Chen, H., & Wang, N. (2011). A unified theory of Tobin's q, corporate investment, financing, and risk management. The journal of Finance, 66(5), 1545-1578.

Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. Pattern recognition, 37(9), 1757-1771.

Bernard VL, Thomas JK (1989) Post-Earnings-Announcement Drift: Delayed Price Response or Risk Premium? Journal of Accounting Research (27):1–36.

Bolton, P., Chen, H., & Wang, N. (2011). A unified theory of Tobin's q, corporate investment, financing, and risk management. The journal of Finance, 66(5), 1545-1578.

Boone, J., et al., Size-specific dose estimates (SSDE) in pediatric and adult body CT exams: Report of AAPM Task Group 204. 2011.

Brockwell PJ, Davis RA (2009) Time series: theory and methods (Springer Science & Business Media).

Butler, T. D., Armstrong, C., Ellinger, A., & Franke, G. (2016). Employer trustworthiness, worker pride, and camaraderie as a source of competitive advantage: Evidence from great places to work. Journal of Strategy and Management, 9(3), 322-343.

Caldwell, C., & Clapham, S. E. (2003). Organizational trustworthiness: An international perspective Wang et al., 2016, Attention-based lstm for aspect-level sentiment classification

Cao D, Wang Y, Duan J, Zhang C, Zhu X, Huang C, Tong Y, et al. (2020) Spectral temporal graph neural network for multivariate time-series forecasting. Advances in Neural Information Processing Systems (33):17766–17778.

Chambon S, Galtier MN, Arnal PJ, Wainrib G, Gramfort A (2018) A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. IEEE Transactions on Neural Systems and Rehabilitation Engineering 26(4):758–769.

Chatman, J. A., & O'Reilly, C. A. (2016). Paradigm lost: Reinvigorating the study of organizational culture. Research in Organizational Behavior, 36, 199-224.

Chen, Y., et al., Adversarial-learning-based image-to-image transformation: A survey. Neurocomputing, 2020. 411: p. 468-486.

Cheng C, Tan F, Hou X, Wei Z (2019) Success Prediction on Crowdfunding with Multimodal Deep Learning. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence.

Cheng D, Xiang S, Shang C, Zhang Y, Yang F, Zhang L (2020) Spatio-Temporal Attention-Based Neural Network for Credit Card Fraud Detection. Proceedings of the AAAI Conference on Artificial Intelligence 34(1):362–369.

Cheng D, Yang F, Xiang S, Liu J (2022) Financial time series forecasting with multi-modality graph neural network. Pattern Recognition 121(108218).

Coff, R. W. (1999). When competitive advantage doesn't lead to performance: The resource-based view and stakeholder bargaining power. Organization science, 10(2), 119-133.

Colli, A. (2013). Family firms between risks and opportunities: a literature review. Socio-Economic Review, 11(3), 577-599.

Das Swain, V., Saha, K., Reddy, M. D., Rajvanshy, H., Abowd, G. D., & De Choudhury, M. (2020, April). Modeling organizational culture with workplace experiences shared on glassdoor. In Proceedings of the 2020 CHI conference on human factors in computing systems (pp. 1-15).

De Gonzalez, A.B. and S. Darby, Risk of cancer from diagnostic X-rays: estimates for the UK and 14 other countries. The lancet, 2004. 363(9406): p. 345-351.

Deng A, Hooi B (2021) Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. Proceedings of the AAAI Conference on Artificial Intelligence 35(5).

Deng S, Huang Z (James), Sinha AP, Zhao H (2018) The interaction between microblog sentiment and stock returns: an empirical examination. MIS quarterly 42(3):895–918.

Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Division, I.M.I., 2019 CT Market Outlook Report. 2019.

(NCRP), N.C.o.R.P.M., Report no. 160: ionizing radiation exposure of the population of the United States. 2009.

Dong, C., et al. Learning a deep convolutional network for image super-resolution. in European conference on computer vision. 2014. Springer.

Du, J., Gui, L., Xu, R., & He, Y. (2018). A convolutional attention model for text classification. In Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6 (pp. 183-195). Springer International Publishing.

Duke B, Ahmed A, Wolf C, Aarabi P, Taylor GW (2021) SSTVOS: Sparse Spatiotemporal Transformers for Video Object Segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR):5912–5921.

Eddleston, K. A., Kidder, D. L., & Litzky, B. E. (2002). Who's the boss? Contending with competing expectations from customers and management. Academy of Management Perspectives, 16(4), 85-95.

Edmans A (2011) Does the stock market fully value intangibles? Employee satisfaction and equity prices. Journal of Financial Economics 101(3):621–640.

Edmans A (2012) The Link Between Job Satisfaction and Firm Value, With Implications for Corporate Social Responsibility. Academy of Management Perspectives 26(4):1–19.

Eldele E, Ragab M, Chen Z, Wu M Time-Series Representation Learning via Temporal and Contextual Contrasting.

Eldele E, Ragab M, Chen Z, Wu M, Kwoh CK, Li X, Guan C (2021) Time-Series Representation Learning via Temporal and Contextual Contrasting. arXiv:2106.14112 [cs.LG]

Fani H, Jiang E, Bagheri E, Al-Obeidat F, Du W, Kargar M (2020) User community detection via embedding of social network structure and temporal content. Information Processing & Management 57(2).

Farahani A, Genga L, Dijkman R (2021) Online multimodal transportation planning using deep reinforcement learning. IEEE International Conference on Systems, Man, and Cybernetics (SMC):1691–1698.

Gao, F., et al., Deep Residual Inception Encoder–Decoder Network for Medical Imaging Synthesis. IEEE journal of biomedical and health informatics, 2019. 24(1): p. 39-49.

Gao H, Wang Z, Ji S (2018) Large-scale learnable graph convolutional networks. Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining:1416–1424.

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. Proc. 13th Int. Conf. Artif. Intell. Stat.:249–256.

Goodfellow, I.J., et al., Generative adversarial networks. arXiv preprint arXiv:1406.2661, 2014.

Great Place To Work®, 2017, Trust Index© Employee Survey, http://teamhmh.com/wp-content/uploads/2017/10/Trust-Philosophy-Doc.pdf

Green TC, Huang R, Wen Q, Zhou D (2019) Crowdsourced employer reviews and stock returns. Journal of Financial Economics 134(1):236–251.

Gu Y, Yang K, Fu S, Chen S, Li X, Marsic I (2018) Multimodal affective analysis using hierarchical attention strategy with word-level alignment. Proceedings of the conference. Association for Computational Linguistics. Meeting. NIH Public Access 2018:2225.

Guo, Q., Qiu, X., Liu, P., Xue, X., & Zhang, Z. (2020, April). Multi-scale self-attention for text classification. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 05, pp. 7847-7854).

Han W, Chen H, Gelbukh A, Zadeh A, Morency L philippe, Poria S (2021) Bi-bimodal modality fusion for correlation-controlled multimodal sentiment analysis. Proceedings of the 2021 International Conference on Multimodal Interaction:6–15.

Han, X., MR-based synthetic CT generation using a deep convolutional neural network method. Medical Physics, 2017. 44(4): p. 1408-1419.

He, K., et al. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) Lightgcn: Simplifying and powering graph convolution network for recommendation. Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval:639–648.

Heng C, Rong Y, Xu T, Bian Y, Zhou S, Wang X, Huang J, Zhu W (2021) Not All Low-Pass Filters are Robust in Graph Convolutional Networks. Advances in Neural Information Processing Systems 34.

Henzler, P., et al., Single-image Tomography: 3D Volumes from 2D Cranial X-Rays. Computer Graphics Forum, 2018. 37(2): p. 377-388.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2008). Design science in information systems research. Management Information Systems Quarterly, 28(1), 6.

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation 9(8):1735–1780.

Hoye, J., et al., Organ doses from CT localizer radiographs: Development, validation, and application of a Monte Carlo estimation technique. Med Phys, 2019. 46(11): p. 5262-5272.

Jebe, R. "The convergence of financial and ESG materiality: Taking sustainability mainstream." American Business Law Journal 56.3 (2019): 645-702.

Ji, S., M. Yang, and K. Yu, 3D convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell, 2013. 35(1): p. 221-31.

Jin, C.B., et al., Deep CT to MR Synthesis Using Paired and Unpaired Data. Sensors, 2019. 19(10).

Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kosaraju V, Sadeghian A, Martín-Martín R, Reid I, Rezatofighi H, Savarese S (2019) Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks. Advances in Neural Information Processing Systems 32.

Kotsantonis, S., Pinney, C., & Serafeim, G. (2016). ESG integration in investment management: Myths and realities. Journal of Applied Corporate Finance, 28(2), 10-16.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., ... & Socher, R. (2016, June). Ask me anything: Dynamic memory networks for natural language processing. In International conference on machine learning (pp. 1378-1387). PMLR.

Kwok, R., Deep learning powers a motion-tracking revolution. Nature, 2019. 574(7776): p. 137-138.

Lahat D, Adali T, Jutten C (2015) Multimodal data fusion: an overview of methods, challenges, and prospects. Proceedings of the IEEE 103(9):1449–1477.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, February). Recurrent convolutional neural networks for text classification. In Proceedings of the AAAI conference on artificial intelligence (Vol. 29, No. 1).

Li J, Yang L, Smyth B, Dong R (2020) Maec: A multimodal aligned earnings conference call dataset for financial risk prediction. Proceedings of the 29th ACM International Conference on Information & Knowledge Management:3063–3070.

Li Y, Nair P, Lu XH, Wen Z, Wang Y, Dehaghi AAK, Miao Y, et al. (2020) Inferring multimodal latent topics from electronic health records. Nature communications 11(1):1–17.

Liu K, Li Y, Xu N, Natarajan P (2018) Learn to Combine Modalities in Multimodal Deep Learning. arXiv:1805.11730 [stat.ML] https://doi.org/10.48550/arXiv.1805.11730.

Liu, L., et al., Deep Learning for Generic Object Detection: A Survey. International Journal of Computer Vision, 2020. 128(2): p. 261-318.

Liu, Q., Gao, Z., Liu, B., & Zhang, Y. (2015, June). Automated rule selection for aspect extraction in opinion mining. In Twenty-Fourth international joint conference on artificial intelligence.

Liu, Y., et al. Self-improving generative adversarial reinforcement learning. in Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. 2019.

Lloyds. (2011) Lloyds Risk Index 2011

Luo Y, Cai X, Zhang Y, Xu J (2018) Multivariate Time Series Imputation with Generative Adversarial Networks. 32nd Conference on Neural Information Processing Systems.

Lynn, M., 2004. Ethnic differences in tipping: A matter of familiarity with tipping norms. Cornell Hotel and Restaurant Administration Quarterly, 45(1), pp.12-22.

Ma H, Li W, Zhang X, Gao S, Lu S (2019) AttnSense: Multi-level Attention Mechanism For Multimodal Human Activity Recognition. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence:3109–3115.

Mai S, Hu H, Xing S (2020a) Modality to Modality Translation: An Adversarial Representation Learning and Graph Fusion Network for Multimodal Fusion. 34(1):164–172.

Mai S, Hu H, Xing S (2020b) Modality to Modality Translation: An Adversarial Representation Learning and Graph Fusion Network for Multimodal Fusion. Proceedings of the AAAI Conference on Artificial Intelligence 34(01):164–172.

Marzullo, A., et al., Towards realistic laparoscopic image generation using image-domain translation. Computer Methods and Programs in Biomedicine, 2021. 200.

Mathis, A., et al., DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nat Neurosci, 2018. 21(9): p. 1281-1289.

Mohamed A, Qian K, Elhoseiny M, Claudel C (2020) Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR):14424–14432.

Montoya, J.C., et al. Volumetric scout CT images reconstructed from conventional two-view radiograph localizers using deep learning (Conference Presentation). in Medical Imaging 2019: Physics of Medical Imaging. 2019. International Society for Optics and Photonics.

NT H, Maehara T (2019) Revisiting graph neural networks: All we have is low-pass filters. arXiv preprint arXiv:1905.09550.

Pointer LV, Khoi PD (2019) Predictors of return on assets and return on equity for banking and insurance companies on Vietnam stock exchange. Entrepreneurial Business and Economics Review 7(4):185–198.

Pradhan, N., et al., Transforming view of medical images using deep learning. Neural Computing & Applications, 2020. 32(18): p. 15043-15054.

Radiation, U.N.S.C.o.t.E.o.A., Effects of Ionizing Radiation, United Nations Scientific Committee on the Effects of Atomic Radiation (UNSCEAR) 2006 Report, Volume I: Report to the General Assembly, Scientific Annexes A and B. 2008: United Nations.

Rahman W, Hasan MdK, Lee S, Zadeh A, Mao C, Morency LP, Hoque E (2020) Integrating multimodal information in large pretrained transformers. Proceedings of the conference.

Ronneberger, O., P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. Cham: Springer International Publishing.

Qin Y, Song D, Chen H, Cheng W, Jiang G, Cottrell G A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971:2017.

Qin Y, Yi Y (2019) What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics:390–401.

Rahman W, Hasan MdK, Lee S, Zadeh A, Mao C, Morency LP, Hoque E (2020) Integrating multimodal information in large pretrained transformers. Proceedings of the conference. Association for Computational Linguistics. Meeting:2359.

Ryu S, Lim J, Hong SH, Kim WY (2018) Deeply learning molecular structure-property relationships using attention-and gate-augmented graph convolutional network. arXiv preprint arXiv:1805.10988.

Sawhney R, Mathur P, Mangal A, Khanna P, Shah RR (2020) Multimodal multi-task financial risk forecasting. Proceedings of the 28th ACM international conference on multimedia:456–465.

Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M (2017) Modeling Relational Data with Graph Convolutional Networks. arXiv:1703.06103v4.

Schlichtkrull M, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M (2017) Modeling Relational Data with Graph Convolutional Networks

Seif, G. and D. Androutsos. Edge-Based Loss Function for Single Image Super-Resolution. in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018.

Semwal D, Patil S, Galhotra S, Arora A, Unny N (2015) STAR: Real-time Spatio-Temporal Analysis and Prediction of Traffic Insights using Social Media. Proceedings of the 2nd IKDD Conference on Data Sciences 7:1–4.

Shen, L., W. Zhao, and L. Xing, Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning. Nat Biomed Eng, 2019. 3(11): p. 880-888.

Shih SY, Sun FK, Lee H yi (2019) Temporal pattern attention for multivariate time series forecasting. Machine Learning 108(8):1421–1441.

Song C, Lin Y, Guo S, Wan H (2020) Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 34(1):914–921.

Song W, Chi C, Xiao Z, Duan Z, Xu Y, Zhang M, Tang J (2019) AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. Proceedings of the 28th ACM International Conference on Information and Knowledge Management:1161–1170.

Stimpel, B., et al., Projection-to-Projection Translation for Hybrid X-ray and Magnetic Resonance Imaging. Scientific Reports, 2019. 9.

Sweetland, S. R. (1996). Human capital theory: Foundations of a field of inquiry. Review of educational research, 66(3), 341-359.

Tan ZX, Soh H, Ong D (2020) Factorized inference in deep markov models for incomplete multimodal time series. Proceedings of the AAAI Conference on Artificial Intelligence:10334–10341.

Tao Z, Wei Y, Wang X, He X, Huang X (2020) MGAT: Multimodal Graph Attention Network for Recommendation. Information Processing & Management 57(5).

Tonekaboni S, Eytan D, Goldenberg A (2021) Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. arXiv preprint arXiv:2106.00750.

Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2017) Graph attention networks. arXiv preprint arXiv:1710.10903.

Wang G, Chen G, Zhao H, Zhang F, Yang S, Lu T (2021) leveraging multisource heterogeneous data for financial risk prediction: a novel hybrid-strategy-based self-adaptive method. MIS Quarterly 45(4):1949–1998.

Wang L, Wu J, Huang SL, Zheng L, Xu X, Zhang L, Huang J (2019) An Efficient Approach to Informative Feature Extraction from Multimodal Data. Proceedings of the AAAI Conference on Artificial Intelligence 33(1):5281–5288.

Wang W, Yang X, Ooi BC, Zhang D, Zhuang Y (2016) Effective deep learning-based multi-modal retrieval. The VLDB Journal 25(1):79–101.

Wang, Y.L., et al., CLCU-Net: Cross-level connected U-shaped network with selective feature aggregation attention module for brain tumor segmentation. Comput Methods Programs Biomed, 2021. 207: p. 106154.

Wei Y, Wang X, Guan W, Nie L, Lin Z, Chen B (2019) Neural Multimodal Cooperative Learning Toward Micro-Video Understanding. IEEE Transactions on Image Processing 29:1–14.

Whittle P (1951) Hypothesis Testing in Time Series Analysis. Almquist and Wiksell, Upssala 121:9.

Wolterink, J.M., et al. Deep MR to CT Synthesis Using Unpaired Data. 2017. Cham: Springer International Publishing.

Wu F, Souza A, Zhang T, Fifty C, Yu T, Weinberger K (2019) Simplifying Graph Convolutional Networks. einberger Proceedings of the 36th International Conference on Machine Learning:6861–6871.

Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-Based Recommendation with Graph Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence 33(1).

Xiang, L., et al., Deep embedding convolutional neural network for synthesizing CT image from T1-Weighted MR image. Med Image Anal, 2018. 47: p. 31-44.

Yan S, Xiong Y, Lin D (2018) Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In Thirty-second AAAI conference on artificial intelligence.

Yang J, Zheng WS, Yang Q, Chen Y, Tian Q (2020) Spatial-Temporal Graph Convolutional Network for Video-based Person Re-identification. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition:3289–3299.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 1480-1489).

Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. Proceedings of the AAAI conference on artificial intelligence 33(1):7370–7377.

Zhang C, Song D, Chen Y, Feng X, Lumezanu C, Cheng W, Ni J, Zong B, Chen H, Chawla NV (2019) A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. Proceedings of the AAAI Conference on Artificial Intelligence 33(01).

Zhang, L., et al., Generalizing Deep Learning for Medical Image Segmentation to Unseen Domains via Deep Stacked Transformation. IEEE Trans Med Imaging, 2020. 39(7): p. 2531-2540.

Zhang S, Tong H, Xu J, Maciejewski R (2019) Graph convolutional networks: a comprehensive review. Computational Social Networks 6(1):1–23.

Zhang X, Gao X, Lu W, He L, Li J (2020) Beyond Vision: A Multimodal Recurrent Attention Convolutional Neural Network for Unified Image Aesthetic Prediction Tasks. IEEE Transactions on Multimedia 23:611–623.

Zhao, C., et al. Whole Brain Segmentation and Labeling from CT Using Synthetic MR Images. 2017. Cham: Springer International Publishing.

Zhao, H., et al., Loss Functions for Image Restoration With Neural Networks. Ieee Transactions on Computational Imaging, 2017. 3(1): p. 47-57.

Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H (2019) T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems 21(9):3848–3858.

APPENDICES

Appendix A: Pseudo Codes (Essay 1)

Pseudo Code for Table Position Detection

```
def get_table_position(f: lateral localizer image):

  for j from 255 to -1: # search for the table position from the right side of the image

      pixel_count = sum(f[0:255, j] > 95) # count pixels with high intensity value

      percentage = pixel_count/256

      if percentage > 0.8: # intensity value is consistent across all image rows

          table_position = j

          break
```

Pseudo Code for Patient Boundary Detection

```
def detect_patient_boundary(f: localizer image):

    # detect the patient boundary at each row

    k = 7 # empirical threshold less than body width and greater than table width

  for i from 0 to 256: # iterate through image rows to find the boundary at each row

      for j from 0 to 256:

          if sum(f[i, j:j+k] > 95) == k:

              BDLT[i] = j

              break

      for j from 255 to -1:

          if sum(f[i, j:j+k] > 95) == k:

              BDRT[i] = j

              break
```

Pseudo Code for Patient Boundary Smoothening

```
def smooth_patient_boundary(f: localizer image, BDLT: left boundary, BDRT: right boundary):

    # set image intensity values to zero and patient boundary intensity values to 255

  for i from 0 to 256:

      for j from 0 to 256:

          if j != BDLT[i] and j !=BDRT[i]:

              f[i, j] = 0

          else:

              f[i, j] = 255

  # smooth the boundary

  for i from 0 to 255:

      f = 8 adjacency connect from BDLT[i] to BDLT[i+1] on f

      f = 8 adjacency connect from BDRT[i] to BDRT[i+1] on f

  scaleup(f) # apply scale-up on the image

    medianBlur(f) # apply median-blur on the image

  scaledown(f) # apply scale-down on the image

  for i from 0 to 256:

      for j from 0 to 256:

          if f[i, j] > 0:

              BDLT[i] = j

              break
```

Appendix B: Ablation Experiment on Loss Function (Essay 1)

Using a randomly sub-sampled dataset (with 615 image pairs), we performed an ablation experiment to select the loss function for our network. We evaluated six loss function options: MSE, SSIM, MSE+SSIM, Scaled MSE, Scaled SSIM, and Scaled Mixture (Proposed). We gauged the performance of these loss functions in terms of the three metrics, i.e., location accuracy, profile accuracy, and attenuation accuracy, as well as the perceived image quality (as illustrated in Figure 1.4).

The quantitative results are summarized as the following. The proposed scaled mixture loss achieved the best performance in terms of location prediction error and profile prediction error for predictions from both directions. MSE loss achieved the best performance in terms of attenuation prediction error. However, with MSE loss, the predicted images are blurry (Figure 1.4), and the model could not generate table lines in the lateral prediction. Therefore, we used the proposed scaled mixture loss to conduct experiments on the full dataset. Future work could consider adding a constraint loss at the image row level to improve the attenuation prediction performance.

Location Prediction Error

Loss	Orient	Mean (mm)	StdDev (mm)	Median (mm)	<2mm (%)	<5mm (%)	<15mm (%)	<20mm (%)
MSE	Lateral							
	AP	10.01	8.03	8.63	4.94	33.38	81.49	90.09
SSIM	Lateral							
	AP	8.99	8.33	**6.47**	4.94	40.74	81.49	90.09
MSE+SSIM	Lateral	16.28	18.01	8.63	19.05	42.86	59.05	61.90
	AP	8.19	**7.03**	**6.47**	9.88	43.21	82.72	90.06
Scaled MSE	Lateral	8.63	16.99	2.16	32.14	79.05	84.76	84.76
	AP	9.13	9.57	**6.47**	13.58	41.96	77.78	90.02
Scaled SSIM	Lateral							
	AP	8.70	7.54	**6.47**	7.41	37.04	83.95	90.09
Scaled Mixture *	Lateral	**3.94**	**9.04**	**2.15**	**32.38**	**88.57**	**95.24**	**95.24**
	AP	**8.12**	7.80	**6.47**	**14.81**	**45.68**	**85.18**	**90.12**

Profile Prediction Error

Loss	Orient	Mean (%)	StdDev (%)	Median (%)	5% (%)	<10% (%)
MSE	Lateral	7.05	2.78	6.36	25.61	85.37
	AP	5.46	2.20	5.10	47.56	96.34
SSIM	Lateral	5.61	2.24	5.25	43.90	**97.56**
	AP	5.66	2.99	4.66	53.66	92.68
MSE+SSIM	Lateral	5.01	2.79	4.72	**67.07**	95.12
	AP	5.84	3.30	5.03	50.00	89.02
Scaled MSE	Lateral	6.99	4.09	6.05	39.02	81.70
	AP	5.14	3.06	4.41	60.98	92.68
Scaled SSIM	Lateral	5.44	2.61	5.08	47.56	95.12
	AP	6.52	3.21	5.80	37.80	85.37
Scaled Mixture *	Lateral	**4.89**	**2.08**	**4.57**	59.76	**97.56**
	AP	**4.33**	**1.94**	**4.13**	**70.73**	**98.78**

Attenuation Prediction Error

Loss	Orientation	Mean (%)	StdDev (%)	Median (%)	<5% (%)	<10% (%)
MSE *	Lateral	**4.8**	**1.89**	**4.52**	**59.75**	**98.78**
	AP	**5.68**	**1.77**	**5.36**	**42.68**	**98.78**
SSIM	Lateral	6.60	3.18	5.92	36.59	86.59
	AP	5.94	3.26	4.85	52.44	93.90
MSE+SSIM	Lateral	6.98	3.96	6.07	30.48	84.15
	AP	6.87	4.09	5.58	39.02	80.49
Scaled MSE	Lateral	11.04	4.62	10.72	8.54	45.12
	AP	8.24	4.80	7.22	32.93	68.29
Scaled SSIM	Lateral	8.72	4.71	7.49	25.61	70.73
	AP	7.72	3.52	7.40	24.39	74.39
Scaled Mixture	Lateral	5.02	2.01	4.53	59.76	97.56
	AP	6.28	2.70	5.97	41.46	89.02

Appendix C: Ablation Experiment on Network Structure (Essay 1)

We conducted an ablation experiment to compare the proposed encoder-decoder network (6-block encoder-decoder network) with 5 alternatives, i.e., (1) 4-block encoder-decoder network, (2) 5-block encoder-decoder network, (3) 4-block encoder-decoder network with skip-connection, (4) 5-block encoder-decoder network with skip-connection, and (5) 6-block encoder-decoder network with a max pooling layer.

The results in terms of location prediction error, profile prediction error, and attenuation prediction error are summarized in Tables C.1, C.2, and C.3. The proposed network (6-block encoder-decoder network) achieved the best performance in terms of location prediction error and profile prediction error for predictions from both directions. The proposed network structure and its variant (6-block + max pooling) comparably achieved the best performance in terms of attenuation prediction error. However, adding a max pooling layer after each encoder block caused the predicted images to become blurry and failed to generate a straight table line in the lateral prediction. We therefore chose the proposed network to conduct the experiments on the full dataset.

The experiment on the various numbers of encoder-decoder blocks shows that the performance improves as the number of encoder-decoder blocks increases, although at the price of increased consumption of computing resources. The experiment on skip-connection shows that without proper transformation, connecting the outputs from the feature domain (encoder blocks) to the transformed domain (decoder blocks) simply consumes more GPU memory yet offers no help on the performance in our focal task.

Location Prediction Error

Network Structure	Orientation	Mean (mm)	Std Dev(mm)	Median (mm)	<2 mm (%)	<5 mm (%)	<15 mm (%)	<20 mm (%)
4-Block	Lateral							
	AP	11.23	8.13	8.63	3.70	32.10	64.20	85.19
5-Block	Lateral							
	AP	9.48	7.90	8.63	8.64	33.31	75.30	90.10
Max-pooling	AP	9.00	7.94	6.47	8.64	38.27	80.24	90.01

6-Block Proposed Network *	Lateral	**3.94**	**9.04**	**2.15**	**32.38**	**88.57**	**95.24**	**95.24**
	AP	**8.12**	**7.80**	**6.47**	**14.81**	**45.68**	**85.12**	**90.12**

Profile Prediction Error

Network Structure	Orient	Mean (%)	StdDev(%)	Median (%)	<5% (%)	<10% (%)
4-Block	Lateral	5.91	3.11	5.27	47.56	90.24
	AP	7.01	3.23	6.39	34.15	82.92
5-Block	Lateral	5.57	2.77	4.65	58.54	91.46
	AP	5.85	2.73	5.41	42.68	92.68
6-Block + Max pooling	Lateral	5.32	2.26	4.73	53.66	97.56
	AP	5.17	2.57	4.85	51.22	92.68
6-Block Proposed Network *	Lateral	**4.89**	**2.08**	**4.57**	**59.76**	**97.56**
	AP	**4.33**	**1.94**	**4.13**	**70.73**	**98.78**

Attenuation Prediction Error

Network Structure	Orient	Mean (%)	StdDev(%)	Median (%)	<5% (%)	<10% (%)
4-Block	Lateral	6.46	3.34	5.52	41.46	89.02
	AP	7.31	3.41	6.79	26.83	82.93
5-Block	Lateral	5.44	2.69	4.62	58.54	91.46
	AP	6.97	2.89	6.86	29.27	87.80
6-Block + Max pooling *	Lateral	5.98	2.68	5.35	43.90	91.46
	AP	**6.27**	3.65	**5.02**	**50.00**	**89.02**
6-Block Proposed Network *	Lateral	**5.02**	**2.01**	**4.53**	**59.76**	**97.56**
	AP	**6.27**	**2.70**	5.97	41.46	**89.02**

Appendix D: Implementation and Execution (Essay 2)

We implemented the proposed *RGML* and the benchmark methods based on the programs of tensorflow-gpu-2.0.0, keras-2.3.1, PyTorch-1.8.0, and scikit-learn-0.23.2. We used the pre-trained BERT model from https://github.com/google-research/bert, and the MFCC feature extraction module from https://librosa.org/doc/latest/feature.html. In the experiments, we performed parameter tuning for the benchmarks to ensure the fairness of the method comparisons. The following table summarizes the experiment settings of some key parameters.

All experiments were run on two servers, each with a NVIDIA RTX 2080Ti GPU. The deployment and execution of *RGML* (for multimodal data preprocessing, model pre-training, model training, and cross-validation) can be completed end-to-end within several hours on a GPU server (specifically, within 240 minutes on a single PC with a NVIDIA RTX 2080Ti GPU).

Parameter Setting in the Experiment.

Model	Parameter	Setting	Description
Pre-trained BERT	Text embedding size	1,024	The output dimensionality of the CLS hidden cell of the pre-trained BERT
MFCC	Audio embedding size	1,024	The feature dimensionality of MFCC for earnings call audios
LSTM+Attention	Representation output dimensionality	32	The output dimensionality of each hidden cell of LSTM for each data modality
	Number of heads	3	The number of heads in the attention mechanism
	Query dimensionality	32	The dimensionality of the query-vector in self-attention
	Key dimensionality	32	The dimensionality of the key-vector in self-attention

	Value dimensionality	32	The dimensionality of the value-vector in self-attention
AutoInt	Feature output dimensionality	96	Number of neurons of the representation layer for *AutoInt*
Soft-HRG	Feature output dimensionality	96	Number of neurons of the representation layer for *Soft-HRG*
MAG	Feature output dimensionality	160	Number of neurons of the representation layer for *MAG*
ARGF	Feature output dimensionality	96	Number of neurons of the representation layer for *ARGF*
MARCNN	Feature output dimensionality	96	Number of neurons of the representation layer for *MARCNN*
STAN	Feature output dimensionality	32	Number of neurons of the representation layer for *STAN*
MAGNN	Feature output dimensionality	96	Number of neurons of the representation layer for *MAGNN*
BBFN	Feature output dimensionality	192	Number of neurons of the representation layer for *BBFN*
RMGL	Feature output dimensionality	32	Number of neurons of each data modality for meta-graph learning
	Number of layers	1	Number of hidden layers in the GCN model
	Epochs	10	Number of epochs to train the model
	Batch size	128	Number of instances per gradient update
	k	80	Percentage of the sum of singular values
	$\boldsymbol{\lambda_1}$	0.0001	Threshold factor of L1 norm
	$\boldsymbol{\lambda_2}$	0.001	Threshold factor of Frobenius norm
	$\boldsymbol{\lambda_3}$	0.01	Threshold factor of trace norm
	$\boldsymbol{\lambda_4}$	0.000001	The trade-off factor of L21 norm

Appendix E. Tukey-Kramer Test (*t* and *p*) of *RMGL* vs the Benchmarks. (Essay 2)

Benchmark		*LSTM + Attention*	*Autolnt*	*Soft-HRG*	*MAG*	*ARGF*	*MARCNN*	*BBFN*
$n = 7$	$T=1$	5.86***	6.21***	9.05***	7.81***	5.15**	4.79**	9.41***
	$T=2$	8.00***	6.30***	6.04***	7.35***	5.25**	5.51***	10.76****
	$T=3$	4.98**	4.98**	4.17*	6.86***	5.38**	6.32***	9.01***
	$T=4$	5.30***	6.29***	6.29***	5.67**	5.30**	4.19*	6.29***
	$T=5$	6.59***	8.08***	8.74***	14.01****	7.91***	5.77***	9.23***
	$T=6$	4.16*	4.75**	11.16***	4.16*	5.58**	7.01***	7.24***
	$T=7$	7.80***	6.12***	4.88**	4.43*	4.79**	4.61**	6.47***
	$T=8$	10.08***	7.68***	9.60***	6.40***	6.40***	6.72***	6.88***
$n = 15$	$T=1$	10.10***	11.22****	29.76***	12.33****	9.36***	27.90***	11.12****
	$T=2$	6.52***	4.75**	32.19***	7.47***	5.98**	4.21*	10.32****
	$T=3$	26.09***	23.52****	8.38***	23.62****	6.98***	25.34***	4.72**
	$T=4$	6.28***	27.30****	9.10***	26.75****	5.42**	9.42***	7.80***
	$T=5$	31.38***	30.43****	30.43***	32.01****	9.65***	31.80***	11.86****
	$T=6$	6.89***	7.58***	9.88***	6.28***	7.89***	9.42***	6.51***
	$T=7$	30.89***	6.68***	8.82***	9.06***	5.96**	29.93***	6.92***
	$T=8$	40.61***	10.74****	40.74***	19.00****	11.29****	16.52***	16.52****
$n = 30$	$T=1$	4.41*	26.49****	24.28***	5.22**	22.47****	24.38***	4.72**
	$T=2$	57.99***	65.67****	55.69***	22.85****	58.57****	53.77***	20.36****

	T =3	21.06***	5.51**	20.42***	6.24***	4.34*	24.94***	6.87***
	T =4	26.55***	26.77****	30.94***	6.25***	6.36***	29.52***	7.35***
	T =5	11.15***	30.54****	26.53***	8.02***	8.69***	27.42***	9.70***
	T =6	21.19***	5.88**	26.71***	23.77****	26.44****	20.92***	5.61**
	T =7	26.08***	4.93*	23.67***	27.94****	25.75****	24.65***	6.68***
	T =8	26.26***	8.41***	27.47***	28.38****	7.20***	27.27***	7.50***

**** $p < 0.001$; *** $p < 0.01$; ** $p < 0.05$; * $p < 0.1$.

Appendix F: Tukey-Kramer Test (*t* and *p*) of *RMGL* with Full Modalities vs *RMGL* with Reduced Modalities. (Essay 2)

Ablated Set of Modalities	
-FI	9.95****
-EA	6.43****
-ET	7.26****
-TC	8.09****
-EC	4.77****

**** $p < 0.001$

Appendix G: Tukey-Kramer Test Result (*t* and *p*) of *RMGL* vs Ablated Variants. (Essay 2)

Ablated Variant	
Without feature-wise interaction	6.35****
Without modality-wise interaction	40.39****
Without temporal interaction	15.24****

**** $p < 0.001$