Neural Insights for Digital Marketing Content Design

Fanjie Kong*
Duke University, USA
fanjie.kong@duke.edu

Yuan Li
Amazon.com, Inc., USA
liayuan@amazon.com

Houssam Nassif*
Meta, USA
houssamn@meta.com

Tanner Fiez
Amazon.com, Inc., USA
fieztann@amazon.com

Ricardo Henao
Duke University, USA
KAUST, KSA
ricardo.henao@duke.edu

Shreya Chakrabarti
Amazon.com, Inc., USA
chashrey@amazon.com

ABSTRACT

In digital marketing, experimenting with new website content is one of the key levers to improve customer engagement. However, creating successful marketing content is a manual and time-consuming process that lacks clear guiding principles. This paper seeks to close the loop between content creation and online experimentation by offering marketers AI-driven actionable insights based on historical data to improve their creative process. We present a neural-network-based system that scores and extracts insights from a marketing content design. Namely, a multimodal neural network predicts the attractiveness of marketing contents, and a *post-hoc* attribution method generates actionable insights for marketers to improve their content in specific marketing locations. Our insights not only point out the advantages and drawbacks of a given current content, but also provide design recommendations based on historical data. We show that our scoring model and insights work well both quantitatively and qualitatively.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Information systems** → **Retrieval models and ranking**; **Users and interactive retrieval**; **Information retrieval query processing**; • **Computing methodologies** → *Natural language processing*; *Computer vision.*

KEYWORDS

Digital marketing, interactive system, deep learning, model interpretation, image and text recommendation.

ACM Reference Format:
Fanjie Kong, Yuan Li, Houssam Nassif, Tanner Fiez, Ricardo Henao, and Shreya Chakrabarti. 2023. Neural Insights for Digital Marketing Content Design. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599875

*Work done while at Amazon.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '23, August 6–10, 2023, Long Beach, CA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599875

1 INTRODUCTION

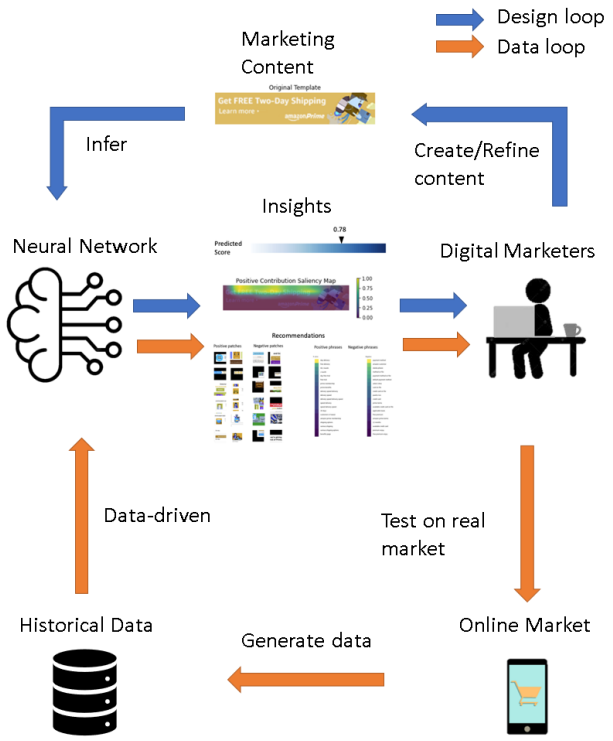

Figure 1: Diagram of AI-driven marketing content design.

Content experimentation plays an important role in driving key performance indicators as part of present-day online marketing [21, 38]. In a typical industrial workflow, digital marketers manually design content, launch controlled online experiments, and receive feedback through collected impression logs. While this process has proven to be reliable for measuring the incremental impact of content creation, it fails to provide insights to the marketer that can improve the likelihood of future experiments being successful. Indeed, unless treatments are deliberately designed relative to a control, it is difficult to establish the source of causality in an experiment outcome. This limits the opportunity to learn the preferences of a customer base. Similarly, the outcomes of online experiments do not immediately provide information to a marketer on how they should design novel content for future experiments.

As a result of the existing content experimentation paradigm, creating new marketing elements is a manual and time-consuming process with significant human involvement. Successful experiments are often the result of subject matter expertise among marketing teams, manual detection of patterns across campaigns, and sequential testing of ideas [44, 47]. Consequently, it is common that resulting insights suffer from cognitive and incentive bias by marketing teams who analyze the results [11].

An opportunity exists to significantly improve the efficiency and effectiveness of marketing content design through data-driven actionable insights. A fundamental challenge to this objective is that extracting actionable content creation insights from data-driven models requires methods that are interpretable by a human. Consequently, existing work in this direction has relied on simple machine learning techniques to model digital marketing content. In the closest related work on the topic [60], a generalized linear model with handcrafted features was developed to score marketing content and provide insights. Despite the promise of this approach, the technique suffers from several shortcomings in real-world marketing scenarios including: *i*) high prediction error, *ii*) a limited number of features, *iii*) the inability to generalize to content with novel features, and *iv*) unclear actionability from interpretation results.

In this paper, we develop a neural-network-based system that scores and extracts insights from a marketing content design to close the loop between content creation and online experimentation (see Figure 1). This approach is motivated by the remarkable success of deep neural nets in diverse application areas [12, 14, 27, 28, 30, 31, 40, 49, 59, 66, 67]. However, providing insights to improve content design is challenging and different from the traditional tasks where deep learning has proven to be successful. This is due to the fact that predictive performance is not the only objective, but it is also necessary to interpret the model, which is challenging given that deep learning models are generally difficult to interpret black-boxes.

We overcome this issue by using *post-hoc* model agnostic attribution methods. In summary, our paper makes the following contributions:

(1) To the best of our knowledge, we may be the first to apply deep learning in the digital marketing design process. We provide an analysis of how to leverage neural network interpretations to help in digital marketing design, and propose a novel image and text insight-generation framework based on attributions from deep neural nets.
(2) We present interpretable insights in an interactive visual format, with actionable insights overlaid with the content.

We validate the performance of the scoring model on an Amazon industry dataset. We also benchmark a variety of interpretation methods using a novel evaluation scheme. To the best of our knowledge, this is the first work to apply deep learning as a tool to model digital marketing content and provide insights to improve content design. Lastly, we publicly release the pseudo-code of algorithms described in this paper for researchers to easily reproduce the code and run our pipeline on their own datasets. Besides, to facilitate replications in other industrial settings, we do share images of our interactive dashboard in Figure 7.

Organization. In Section 2, we introduce the workflow that describes how digital marketers conduct experiments. In Section 3, we present the neural network model used to model the content data and the process used to train the neural network. In Section 4, we explain the method proposed to generate insights for content based on our multimodal neural network. In Section 5, we propose a three-step approach to quantitatively evaluate the performance of our insights with respect to the correlation between applying insights-guided modification and the observed outcome. Finally, in Section 6, we discuss our experiments and their results.

2 DATASET AND METRIC

Controlled experiments, also called randomized experiments or A/B tests, have had a profound influence in multiple fields, including medicine, agriculture, manufacturing, and advertising [21, 38]. Randomized and properly designed experiments can be used to establish causality, that is, to identify elements in marketing content likely to provide incremental impact [55]. In this paper, our goal is to use neural networks to model digital marketing experiments, and learn causal effects from interpreting the behavior of the model.

A typical marketing dataset consists of multiple sequences of controlled experiments conducted by marketers in different digital marketing locations. The dataset used in this paper contains tens of thousands of distinct content items and corresponding success rates. Each marketing content includes various modalities, for instance, an image I corresponding to the web-page screenshot of the content, a text T that contains all textual campaigns in the content, a string D that indicates the marketing content domain and location, and a set of categorical features F that are extracted from the raw content with (potentially) handcrafted functions.

The target metric we adopt is the success rate. In a binary setting, success can be defined as a click, a purchase, or other valuable customer action. Using clicks as an example, success rate is the number of clicks over the number of times the content is shown:

$$Y = N_{\text{clicks}}/N_{\text{total}}, \tag{1}$$

where N_{total} is the total number of people who viewed the content and N_{clicks} is the number of people who clicked on it. Our goal is to predict the success rate Y using the multimodal input X, while providing insights by interpreting the model and its predictions.

3 MARKETING CONTENT NEURAL MODEL

We now introduce the details and components of our marketing content scoring model. As a working example, we represent a marketing content using four of its modalities: image I, text T, content domain D, and feature vector F. We encode each modality using a corresponding widely-used and efficient neural architecture (see Equation 2). The image encoder is an RGB ResNet-18 [33] model without the fully-connected classifier. The text encoder is a standard BERT model [18] without the classification head. Fully-connected MLP neural networks [53] serve as the encoders for both domain and categorical features. The details of these networks are in Appendix B. We then use the most basic fusion strategy [25] by concatenating the embeddings from all modalities via their encoders (see Equation 3). Finally, we feed the concatenated embeddings into another fully-connected MLP neural network for regression.

Formally, given input content $X = \{I, T, D, F\}$, the corresponding embedding is given by $X_{\text{emb}} = \{I_{\text{emb}}, T_{\text{emb}}, D_{\text{emb}}, F_{\text{emb}}\}$, where

$$\begin{aligned} I_{\text{emb}} &= \text{ResNet}(I), \quad T_{\text{emb}} = \text{BERT}(T), \\ D_{\text{emb}} &= \text{MLP}_1(D), \quad F_{\text{emb}} = \text{MLP}_2(F). \end{aligned} \tag{2}$$

Then, denoting $C(\cdot)$ as the final module which takes all modalities as input, the success rate prediction $\widehat{y}$ is given as follows:

$$\widehat{y} = C(X_{\text{emb}}) = \text{MLP}_3(\{I_{\text{emb}}, T_{\text{emb}}, D_{\text{emb}}, F_{\text{emb}}\}). \tag{3}$$

To facilitate model convergence, each sub-network in the multi-modal model is pretrained separately. We begin by appending a classification head after each encoder to allow it to predict the success rate. Then, we train each module using a view of the dataset that only contains the respective modality. Importantly, the regression network $C(\cdot)$ is not trained since we do not have access to its input (concatenated embeddings of all modalities) at this (pretraining) stage. After pretraining each sub-network, the whole multi-modal network is trained on the multi-modality dataset. The encoders are initialized with the weights obtained in the pretraining stage. We report the single-modality sub-networks and final model performance metrics in Section 6.

Since we want to predict the continuous, but bounded, success rate Y, we append a sigmoid function $\sigma(\cdot)$ after the output $\widehat{y}$ of the final regression function $C(\cdot)$. Our optimization objective is the mean-squared error (MSE) between Y and $\sigma(\widehat{y})$:

$$L = \text{MSE}(Y, \sigma(\widehat{y})). \tag{4}$$

4 NEURAL INSIGHTS

In this section, we describe how we utilize *post-hoc* interpretation methods to produce insights from our scoring model. A key advantage of *post-hoc* interpretation is that it can be constructed from an arbitrary prediction model. This property alleviates the need to rely on customized model architectures for interpretable predictions [24, 65] or to train separate modules to explicitly produce model explanations [16, 29]. This section begins by motivating the utility of insights *post-hoc* attribution, then describes the attribution methods, and concludes by explaining how we develop insights from attribution techniques. Note that we are formulating a new problem in deep learning, where our insights aim to help marketers improve existing content.

4.1 Insights: guidance to improve current design

We start by addressing the attribution problem [7, 62], defined as the assignment of contributions to individual input features [20]. The aim of this subsection is illustrative; we seek to show in a near-ideal scenario that *post-hoc* attributions from a neural network can help improve the success rate of content that is being developed, whereas Section 6.3 verifies it empirically. Toward this goal, let us define the input content as a bag a features with a success rate.

Definition 1. *The input content X is a bag of features $X = \{x_i \in \mathbb{R}^n | i = 1, 2, ..., N\}$ with common success rate label $Y \in [0, 1]$.*

We now assume that the underlying success rate Y corresponding to content X can be represented as a linear combination of attribution scores for each feature in the representation of X.

Assumption 1. *Given a tuple $\{X, Y\}$, let $\{y_i \in \mathbb{R} | i = 1, 2, ..., N\}$ be the contribution of features of X to the ground-truth success rate Y, such that $\sum y_i = Y$. Each individual attribution y_i corresponds to an individual input feature x_i. We only have access to the bag label Y, while the ground-truth feature-level attribution y_i is unknown.*

We define an *attributor* as a function that estimates the contribution y_i of a feature $x_i \in X$ to the success rate prediction for the entire bag X. For example, a digital marketer has a set of promotional slogans $\{x_1, ..., x_r\}$, the contribution of each slogan to the success rate is $\{y_1, ..., y_r\}$. After adding these slogans to a blank content, the success rate of the blank content increased by an increment of $Y = \sum_{i=1,...,r} y_i$. Our attributor predicts the contribution of each slogan in the content such that $c(x_i) = y_i \forall i = 1, ..., r$.

Definition 2. *Given a prediction function $C(\cdot)$ such that $C(X)$ predicts Y, define an attributor $c(\cdot)$ as a function that estimates the contributions of each input feature $x_i \in X$ to the prediction $C(\cdot)$, which can be expressed as $C(X) = \sum_{x \in X} c(x)$.*

We now use this framework to show that using a feature with a higher attribution score than an existing feature would increase the overall success rate in a near-ideal scenario. This underscores that attribution methods can act as a guide for digital marketers to refine their existing content given that this effect can also be validated empirically (see Section 6.3).

Consider replacing a feature x in bag X with another feature $\bar{x}'$ such that $c(\bar{x}') \geq c(\bar{x})$, which is consistent with an A/B testing in which a treatment is derived from a control [10, 21]. Let X' be the treated content $X' = (X \backslash \{\bar{x}\}) \cup \{\bar{x}'\}$, where $\bar{x}$ is replaced by $\bar{x}'$. We now show that the treated content X' will have a higher success rate under certain assumptions.

Proposition 1. *Replacing a feature $\bar{x}$ in bag X with a feature $\bar{x}'$ such that $c(\bar{x}') \geq c(\bar{x})$ will increase the overall success rate from Y to Y' when $C(X') \geq C(X) \Leftrightarrow Y' \geq Y$, and under Assumption 1.*

Proof. By Definition 2, $C(X) = \sum_{x \in X} c(x)$. Thus, since $c(\bar{x}') \geq c(\bar{x})$ by construction, we have that

$$C(X') = \sum_{x \in X} c(x) + (c(\bar{x}') - c(\bar{x})) \geq C(X). \tag{5}$$

Since $C(X') \geq C(X) \Rightarrow Y' \geq Y$, we conclude that $Y' \geq Y$. □

The above example indicates that replacing features with higher $c(x)$ would increase Y when $C(X)$ is positively correlated with Y. In real-world datasets, the condition $C(X') \geq C(X) \Rightarrow Y' \geq Y$ may not always hold. However, we use pairwise accuracy to evaluate the accuracy of our predictor when comparing two content elements in Section 6.3 and validate the efficacy of the replacement. Below, we detail both the prediction function $C(\cdot)$ and attributor $c(\cdot)$.

4.2 Post-hoc attribution methods

There are three common trends in mechanisms behind *post-hoc* attribution. Back-propagation-based methods compute attributions according to the gradients with respect to the input [62]. Activation-based methods use a variety of ways to weigh activation maps of intermediate layers in neural network to assign attributions [57]. Perturbation-based methods treat the network as a black-box and assign importance by observing the change in the output after

perturbing the inputs. For instance, feature ablation [46] is done by replacing each input feature with a given baseline (zero vector), and computing the difference in the output. Another alternative is by approximating Shapley values in deep neural networks [6, 45]. Kernel SHAP leverages a kernel-weighted linear regression to estimate the Shapley values of each input as the attribution scores [45]. Langlois et al. [41] use PCA to aggregate a variety of attribution methods to estimate the *shared* component of the variance between different types of attention maps.

In our implementation, we borrow directly from the mentioned *post-hoc* attribution methods, namely, GradCam, Integrated Gradient, Kernel SHAP, Feature Ablation, and PCA, to approximate the attributor $c(\cdot)$. If the prediction function is a multimodal neural network $C(X)$ as defined in Section 3, the attributor is given by $c(x_i) = \text{attribution}(C(X))[i]$. Note that attributions are rescaled to satisfy $C(X) = \sum_{i=1}^{n} c(x_i)$, as in Definition 2.

4.3 Insights: recommending design elements

Given this attribution framework, our system leverages historical data to provide recommendations of visual and textual design element alterations (see Figure 2). The goal of our recommendation is to identify features that are highly likely to improve the success rate of a content being iterated on, and to provide hints for marketers as they embark on designing brand new creative content. We recommend features ranked by their mean attribution score across the whole dataset. We compute the rank score r of a feature x as:

$$r(x) = \frac{1}{N} \sum_{X \in \mathcal{X}} c(x), \tag{6}$$

where the rank score $r(x)$ is an estimate of the expected attribution score over the data distribution $\mathcal{X}$.

We split the implementation of our recommendation strategy according to its modality, whether text or image. We explain our recommendation strategy below and illustrate it in Figure 2.

Figure 3: Example of marketing recommended phrases.

Text Recommendation. For text data, we include word-level and phrase-level recommendations. In word-level recommendations, we simply recommend words that have high average attribution scores across all text contents in a marketing location. In phrase-level recommendations, *i*) we use phrasemachine [32] to extract phrases from each single text content; *ii*) we then compute the attribution score of a phrase by averaging the attribution scores of all its words; *iii*) finally, we recommend phrases that have high average attribution scores across all text contents within a domain. We define *positive* phrases as phrases with the top-10 rank scores while *negative* phrases are phrases with the bottom-10 rank scores.

Figure 3 shows an illustrative example of top and bottom scoring phrases. In the positive phrases, our model recommends using slogans about benefits such as "free game", "free trial", "free twitch", "unlimited access millions songs", *etc.* The negative phrases are about pricing, payment and legal terms, such as "prime 7. 99 month", "credit card", "applicable taxes", *etc.*

Image Recommendation. For image data, we overlay historical ground truth attributions on top of the image in consideration, recommending actions on patches (subsets) of the image. While recent works show the success of deep neural networks in image recommendations [12, 34, 48, 61], our image recommendation zooms into the salient patches inside images, aiming to provide users with key visual elements that contribute most to the label of the image. The goal of our image recommendation algorithm is very different from that of existing works. Moreover, our methodology to use attributions to find salient patches and cluster them to detect common patterns is innovative.

Our image recommendation consists of the following steps. *i*) We first run the attribution method on each single content in the whole dataset. Then, we extract patches with top-K attribution scores in an image. Once a patch is selected, we execute non-maximum suppression on the region of the selected patch to ensure each patch is distinct. *ii*) Subsequently, we cluster these patches based on their ResNet-18 embeddings using K-Means clustering [8] to uncover the design patterns of these patches. *iii*) Finally, patch recommendations are collected from each cluster.

In our work, we leverage K-means clustering to help us group similar image patches, as it has been successfully used for unsupervised image classification [5, 51]. We use the elbow method to select the number K of centroids [63]. In order to encourage a diverse set of suggestions, we randomly sample an equal number of patches from each cluster, as different clusters reflect distinct visual information. This procedure ensures the image recommendations have enough variety of patterns and avoids recommending repetitive patches. The positive patches are randomly sampled from clusters within the top-10 rank scores while the negative patches are randomly sampled from clusters within the 10 lowest rank scores.

Figure 4 shows an illustrative example of our visual design recommendation. The recommendations of images have some insights similar to text insights in Figure 3. Some positive patches are illustrations about benefits and some negative patches are illustrations about payment (row 2, column 10) and offers without revealing discounts and upgrades (row 2, column 2). This example seems to suggest that using the icon of prime (row 1, column 3) is more attractive than the generic Amazon icon (row 2, column 7). Moreover, negative patches shows a distorted Prime logo (row 2, column 4), an exaggerated human face (row 2, column 5) and an infantile cartoon (row 2, column 8), characters that resonate less with many customers [58], while positive patches recommend entertainment icons (row 1 in columns 2, 4, 8) and more favorable human illustrations such as upbeat smiling persons (row 1, columns 6, 9).

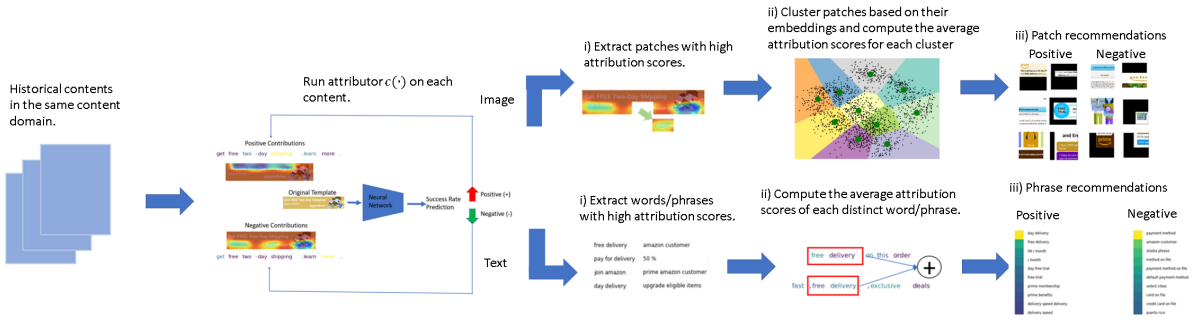

Figure 2: Generating image and text recommendations. For all content instances in the same marketing location D, we first run attribution methods on ResNet-18 and BERT separately to get the attribution maps for text and images. For text data, the scores are simply ranked by the average of the overlaid attribution values on the same words or phrases. For image data, we crop the salient area in every image, and then cluster them based on their embeddings in ResNet-18. Their scores are ranked by the average attribution scores in the same cluster. See Section 4 for details.

Figure 4: An example of visual insights. We show 10 positive patches within the top-10 rank scores and 10 negative patches within bottom-10 rank scores.

5 INSIGHTS EVALUATION

We now tackle the open-ended problem of evaluating insights. We need a practical insight-evaluation metric that marketers can track and trust, that captures the relationship between acting on an insight and its ensuing causal effect, and conveys the expected success rate increase if that insight is applied. Existing evaluation metrics of interpretation methods span faithfulness, stability and fairness [3], which do not satisfy our needs. Runge et al. [54] quantify the strength of causal relationships from observational time series data with pairwise correlations. In our work, we aim to examine the relationship between insights-guided modifications and the ensuing change in the actual success rate. However, evaluating our insights is an inherently difficult problem since no explicit ground truth feature-level attributions y exist.

In Section 4, we show that one can leverage insights to improve content attractiveness with an optimal prediction function $C(\cdot)$. However, in real-world situations, we may not be able to obtain a model $C(\cdot)$ satisfying the idealized properties. Further, a content change could span multiple features of the original design. Similar to [69], which computes the correlation of absolute neighbour differences to detect heteroscedastic relationships, we use the Pearson correlation between the predicted attribution difference and the actual success rate improvement to quantify the relative performance of an an insight.

Algorithm 1: A generic three-step approach to evaluate insights of attributor $c(\cdot)$.

Data: Input pairs of control bags and treatment bags (X, X'), $\forall X, X' \in \mathcal{X}$ and (Y, Y'), $\forall Y, Y' \in [0, 1]$ are the pairs of control labels and treatment labels respectively, and the evaluated attributor $c(\cdot) : \mathbb{R}^n \to [0, 1] \subset \mathbb{R}$.

Result: Correlation coefficient ρ.

Step *i*). Compute the distinct elements set S, such that the attributes in S can be only found in X or X'.

$$S := \{x | (x \in X \wedge x \notin X') \cup (x \notin X \wedge x \in X')\};$$

Step *ii*). Compute predicted attribution difference d_C and actual success rate improvement d_Y:

$$\mathbf{d_C} := \text{sign}(Y' - Y)\Big(\sum_{x \in (X' \cap S)} c(x) - \sum_{x \in (X \cap S)} c(x)\Big);$$

$$\mathbf{d_Y} := |\Delta Y|;$$

Step *iii*). Examine the linear relationship of variable $\mathbf{d_C}$ and variable $\mathbf{d_Y}$ by computing the Pearson Correlation ρ on the whole dataset.

Output ρ.

Specifically, we define the difference between two contents as the *difference set* $S = \{x | (x \in X \wedge x \notin X') \cup (x \notin X \wedge x \in X')\}$, the predicted attribution difference as $\Delta c(x) = \sum_{x \in (X' \cap S)} c(x) - \sum_{x \in (X \cap S)} c(x)$, and the actual success rate improvement as $\Delta Y =$

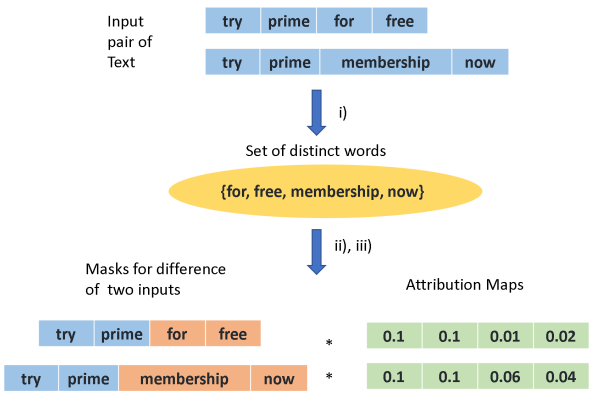

Figure 5: Visual explanation of text evaluation Algorithm 3.

$Y' - Y$. We postulate that a linear relationship exists between $(\Delta c(x), \Delta Y)$, which implies that marketers can improve the content's attractiveness by making modifications based on the insights. Hence, we propose a method that first finds $\Delta c(x)$ by computing the difference set of instances S, and then evaluates the Pearson correlation coefficients ρ across all possible control and treatment pairs within the same content domain in the dataset [9]. The Pearson Correlation Coefficient used in our evaluation is defined as:

$$\rho = \frac{\text{cov}(\Delta c(x), \Delta Y)}{\sigma_{\Delta c(x)} \sigma_{\Delta Y}}, \tag{7}$$

where $\text{cov}(\Delta c(x), \Delta Y)$ is the covariance between $\Delta c(x)$ and ΔY, $\sigma_{\Delta c(x)}$ is the standard deviation of $\Delta c(x)$, and $\sigma_{\Delta Y}$ the standard deviation of ΔY. In our implementation, since we do not have direct access to $\Delta c(x)$ and ΔY, we compute the Pearson Correlation Coefficient ρ of their surrogates. We denote the surrogates of $\Delta c(x)$ and ΔY as $\mathbf{d}_C$ and $\mathbf{d}_Y$, respectively.

Evaluation Algorithm Description. We propose a generic three-step approach to evaluate insights in Algorithm 1. We also provide the pseudo-code of our implementation to evaluate insights for real-world data structures including images and text in the Appendix (Algorithms 3 and 4). The general idea of Algorithm 1 is:

(1) First, we find a difference set S of two input samples, which represents the distinct elements that only appear in one of them. Based on the difference set S, we generate two masks for two input samples. Note that the masks retain the elements in the set S.
(2) Then, we use the masks to obtain the inner product of the corresponding attribution maps for two samples, and compute the difference d_C of the inner product results. We call it the predicted summed attributions of modifications. d_C represents the total attributions when sample one is modified to sample two, or *vice versa*. We also get d_y by computing the difference in ground truth success rates of two samples.
(3) Finally, we quantify the linear relationship between d_C and d_y by computing a correlation coefficient ρ on the whole test dataset.

The resulting correlation coefficient represents how well, when the input sample is modified based on the attribution insight, can it contribute to the change of its ground truth success rate. This metric is very useful in our digital marketing setting, where our goal is to provide deep insights generated by attributions to help digital marketers amend their content to improve its attractiveness. To ensure the algorithms operates accurately, each pair of samples used to compute d_c and d_y must be a pair of control and treatment instances from the same content experiment.

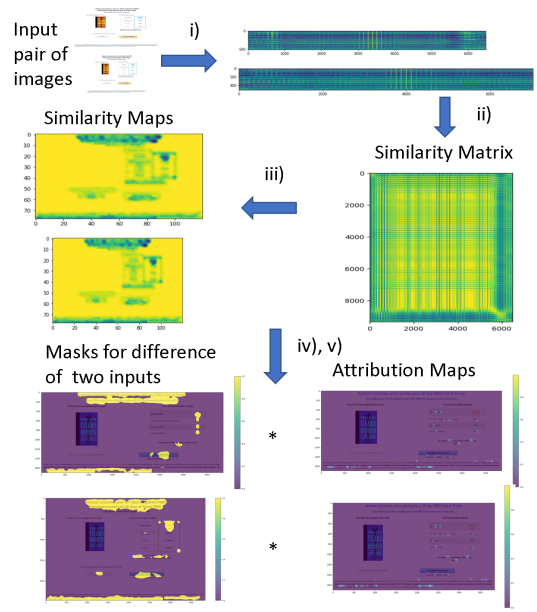

Figure 6: Visual explanation of image evaluation Algorithm 4.

Insight Examples. Figures 5 and 6 provide visual explanations of our insights evaluation algorithm in text and image settings, respectively. Figure 5, illustrating text evaluation, can be understood as follows. In step *i*), we extract a set of words that only appear either in the control sentence or the treatment sentence. In step *ii*), we use this set to create a mask for both sentences, where each element in the mask is 1 (orange color in the figure) if the word in that position belongs to the set S, or 0 (blue color in the figure) if the word in that position does not appear in set S. In step *iii*), we take the inner product of the masks with the attribution maps to produce d_C.

Figure 6, illustrating image evaluation, can be understood as follows. Step *i*) creates the feature maps of the control and treatment images. After properly reshaping the feature maps, step *ii*) computes the similarity between every pair of control-treatment feature vectors, creating a similarity matrix. Step *iii*) takes the matrix with maximum control-treatment similarity score for each location. Step *iv*) thresholds the similarity maps to create masks for differences between control and treatment, and reshapes them to the same size as their corresponding attribution maps. Step *v*) takes the inner product of the masks with the attribution maps to produce d_C.

6 EXPERIMENTS

We evaluate our algorithm on the dataset described in Section 2. If a modality is missing, we use a zero vector to substitute the missing

embedding. We split the dataset into training, validation and test sets with a ratio of 50:10:40. To evaluate the performance of our model on both existing and unseen content domains, we divide the test set into in-domain and out-of-domain subsets. In-domain only contains content domains present in the training set, and out-of-domain includes market domains absent from it.

6.1 Model Specifications

Here, we describe the training hyper-parameters used in our experiments. We use ResNet-18 as the image model. The image model is trained via an Adam optimizer with a batch size of 32, β_1 of 0.9, β_2 of 0.999 and a learning rate of 0.001 for 50 epochs. During pre-training, we randomly crop a 512×512 patch from the image as input in order to limit GPU memory usage. When we train the full multimodal model and infer new samples, we feed the whole image as the input. Due to the high memory consumption of processing full-size screenshots (size ranging from 1000×1000 to 6000×6000), we freeze the weights of image models at this stage, avoiding GPU out-of-memory issues. The text model uses BERT as its backbone, and is trained by an Adam optimizer with a batch size of 8, β_1 of 0.9, β_2 of 0.999 and a learning rate of 0.001 for 50 epochs.

The domain, feature and regression modules are four-layer fully-connected MLP neural networks, with each layer followed by batch normalization and ELU activations. ELU activation is often used in regression tasks [35]. The domain module and the feature module are trained via an Adam optimizer with a batch size of 512, β_1 of 0.9, β_2 of 0.999 and a learning rate of 0.0005 for 50 epochs. The regression network is trained when we optimize the complete multimodal model. After separately pretraining the image, text, domain and feature models, we train the whole multimodal model with a batch size of 32, β_1 of 0.9, β_2 of 0.999 and a learning rate of 0.001 for 50 epochs. See Appendix B for neural network architecture details.

6.2 Interactive Dashboard

For ease of use, we propose an interactive dashboard for digital marketers to visually work their content (see Figure 7). Our dashboard aims to provide similar functionality to [60], but our framework turns out to be more powerful and comprehensive. Specifically, our dashboard has merits that facilitate digital marketing design.

(1) In [60], the insights are restricted to the handcrafted features, which suffer from inefficient scalability and intuitiveness. For example, it is unclear what to do with the insights on a specific attribute like "lighting". Does it direct the marketer to increase the lighting of the whole page or a specific section? In contrast, our insights are directly overlaid on the original content as a saliency map, as in Figure 7.
(2) Our system provides recommendations of design elements based on historical data. When the marketers design a website content in a specific content domain, our dashboard shows the patches and words/phrases with the highest average interpretation scores on historical data with the same content domain D.
(3) Our system easily extends to new marketing content and novel features, thus our insights are not constrained to existing marketing content features.
(4) Our success rate prediction is more accurate. We compare several commonly used machine learning models with our proposed deep multi-modal method. The results in Table 1 show our model outperforming the rest. We also present an insights "Trust Score", which is based on the insights evaluation results in Table 2.

6.3 Evaluation

We evaluate our method using multiple methods. We quantitatively evaluate the success rate prediction, comparing our proposed multi-modal neural network to competing methods. We then report the predicted causal effect of applying our insights to improve content using our proposed correlation metric. Qualitatively, we exhibit some feedback of using our interactive dashboard in Section 6.2 to design marketing contents from real-world digital marketers.

Algorithm 2: Pairwise Accuracy.

Data: Predictions $\widehat{\mathbf{y}} = [\widehat{y}_1, \ldots, \widehat{y}_n]$, truth $\mathbf{y} = [y_1, \ldots, y_n]$
Result: Pairwise accuracy score s.
Initialize count = 0 and hit = 0 .
for every distinct pair $(\widehat{y}_i, \widehat{y}_j)$ **and** (y_i, y_j) **in dataset** do;
  if $\text{sign}(\widehat{y}_i - \widehat{y}_j) = \text{sign}(y_i - y_j)$;
    hit = hit + 1;
  end ;
  count = count + 1 ;
end ;
$s \leftarrow$ hit/count;
Output s

Success rate and pairwise prediction. Table 1 shows the success rate prediction results of different scoring models on our dataset. Here, we report the change in Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), both commonly used to evaluate the performance of regression models. We test the Generalized Linear Model used in [60]; MLP and XGBoost using only categorical features extracted from text and images, which are typically used in industrial applications; and deep learning models that take a single modality as input (BERT with text as input and ResNet-18 with image as input). The results show that our multi-modal neural network outperforms all competing methods.

During content experimentation, marketers often target a content to iterate on and improve. Then they conduct an experiment to compare the control content with its modified counterpart(s) (*i.e.* treatments). We use the pairwise ranking accuracy [1] between the control and each treatment counterpart to evaluate the performance of our models. Algorithm 2 details how pairwise accuracy is computed. The *Pairwise Accuracy* of our proposed model achieves a relative percentage increase of +38% on an out-of-domain test set when compared to GLM. This result shows that our neural network model is much more accurate for marketers in real-world use-cases.

Evaluating Insights. In Table 2, we evaluate the insights generated from the trained deep neural networks using our proposed evaluation scheme (see Section 5). In our dataset, we have multiple

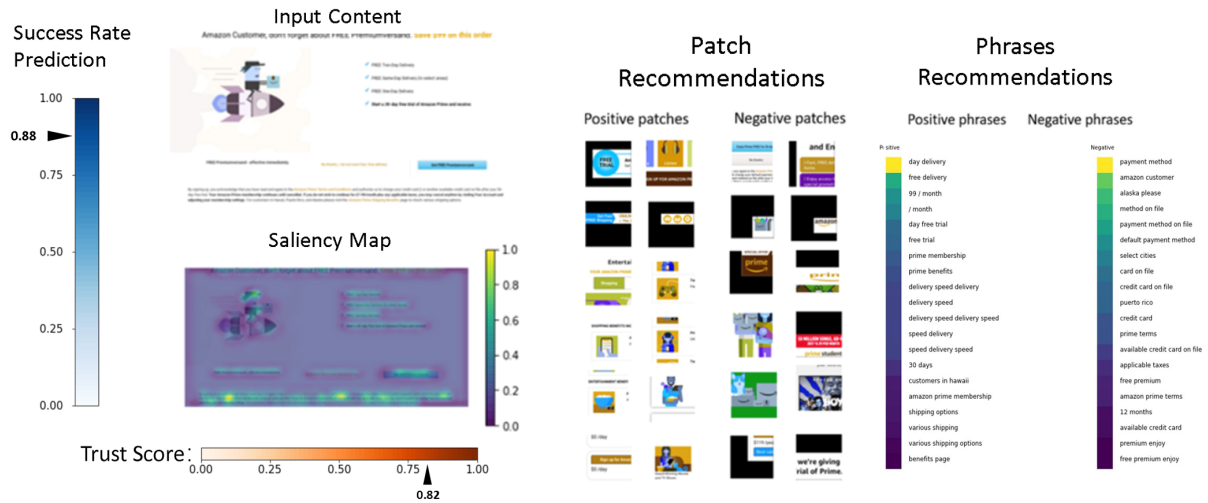

Figure 7: Exemplar dashboard of our interactive system to refine existing content or design new content.

Table 1: Success rate prediction results for different models and modality combinations. We show the percentage decrease of RMSE and MAE for each model compared to GLM.

Model	Modality	In-domain test set		Out-of-domain test set	
		RMSE change ⬇	MAE change ⬇	RMSE change ⬇	MAE change ⬇
GLM	Categorical Features	0 %	0%	0%	0%
MLP	Categorical Features	-42%	-31%	-44%	-35%
MLP	Domain	-54%	-33%	-50%	-29%
XGBoost	Categorical Features	-38%	-9%	-41%	-24%
ResNet-18	images	-25%	19%	-38%	-12%
BERT	Text	-59%	-64%	-59%	-66%
Multi-modal Neural Network	All modalities	**-68%**	**-65%**	**-66%**	**-75%**

Table 2: Results of insights evaluation. The performance metric is the percentage increase of Pearson Correlation Coefficient defined in Equation 7 for each attribution method compared to GradCam.

	GradCam	Integrated Gradient	Kernel SHAP	Feature Ablation	PCA
$\Delta\rho_{\text{text}}$ ⬆	0%	+288%	+109%	-14%	**+493%**
$\Delta\rho_{\text{image}}$ ⬆	0%	+55%	-18%	+0.5%	**+145%**

treatments related to a given control, requiring $O(n^2)$ time to compute d_C and d_y. We avoid such computational complexity by only comparing control with the best performing treatment in the same content domain.

For text data, Integrated Gradients performs the best among GradCam, Kernel SHAP and Feature Ablation. After we integrate these interpretation methods together by PCA, our method yields the highest correlation score. PCA returns a relative percentage increase of +493%, which is a very high correlation score. The result of PCA indicates a strong correlation between insights and success rate improvement, suggesting that the insights are trustworthy. Marketers should consider modifying their templates based on the insight attribution scores, and the insights-guided modification are highly likely to improve the success rate.

For image data, all above-mentioned attribution methods are too slow or intractable, as the size of image inputs is much larger than text inputs, taking too much time to compute attributions for all input pixels. To run the experiment in a reasonable time, we discard the very large images that has more than $5e + 06$ pixels and evaluate the insights of the remaining image data. From the results, we still see the pattern that Integrated Gradients and PCA methods outperform GradCam, with Integrated Gradient and PCA posting a correlation increase of +55% and +145% respectively. We hypothesize that Integrated Gradient is more accurate since it computes attributions on the original image, as opposed to computing it on the intermediate activations, as with GradCam. PCA integrates different aspects of attributions and captures the shared variance of attribution maps from GradCam, Integrated Gradients, KernelShap and Feature Ablations, leading to the best results.

User Experience. To further demonstrate the claims in our paper, we launched a demo of the functionality discussed in Section 6.2. The demo dashboard looks similar to Figure 7, including a saliency map that highlights which parts of the input content to keep or redesign, and recommended phrase and patch insights to act on.

The demo has been shown to tens of professional digital marketers, with mostly positive feedback.

Here is a positive feedback example, which highlights the usefulness of our framework in facilitating marketing content design: "The new demo visualization insights helped make analyzing our current templates faster - allowing marketers to spend more time identifying opportunities, create hypotheses, and test new experiences based on the results. In addition, the positive and negative contribution saliency maps enable marketers to select what areas of a template may have the highest impact during experimentation. We are looking forward to continue working to develop this tool and use it to help with successful experiments!"

In the above user's feedback, the marketer praises our positive and negative contribution saliency maps. In our implementation, the positive (negative) contribution map is based on the absolute value of the positive (negative) part of the attribution map. This visualization makes it easy for users to identify the positive and negative impact of the input content.

7 RELATED WORKS

In this section, we briefly discuss the existing works related to our topic, including modeling digital marketing contents, related deep learning approaches for text and image recommendations, and evaluation metrics for attribution methods. Note that none of these related works fully scales and solves our problem, especially as we define distinct tasks in Sections 4 and 5.

Modeling Digital Marketing Contents. The problem of modeling digital marketing content has triggered substantial research efforts [22, 60, 68, 70] over the past decade. Fong et al. [22] developed a machine learning pipeline to classify advertising images based on their quality. Wang [68] combines deep neural network and evolutionary algorithm to predict optimal personalized marketing strategy for better incomes. Zhou [70] proposes a recommendation algorithm based on recurrent neural network and distributed expression for recommending new products to consumers based on their browsing history. The above-mentioned works are out of our scope, as we focus on extracting insights from deep models to help digital marketers improve their content. The closest research to ours is Sinha et al. [60], which aims to improve the attractiveness of contents by providing AI insights. Nevertheless, they use a much simpler machine learning pipeline than ours, such that our framework has better prediction accuracy and more interpretable insights. Besides, they don't propose an insights evaluation metric, making us the first researchers to quantitatively examine the effectiveness of generated marketing AI insights.

Related Deep Learning Approaches. Among the reproducible deep learning approaches, our recommendation is quite similar to prototype learning. Prototype learning is a form of case-based reasoning [39, 56], which draws conclusions for new inputs by comparing them with a few exemplar cases (i.e prototypes) in the problem domain [17, 43]. It is a natural practice in our day-to-day problem-solving process. For example, physicians perform diagnosis and make prescriptions based on their experience with past patients [19, 26], and mechanics predict potential malfunctions by recalling vehicles exhibiting similar symptoms [27]. Prototype learning imitates human problem-solving processes for better interpretability. Recently the concept has been incorporated in convolutional neural networks to build interpretable image classifiers [17, 43]. Our framework is somehow similar to ProtoPNet [17], in the sense that we both first highlight the salient areas and then make recommendations. ProtoPNet outputs the recommendations that explain the image classification results, while our recommendations focus on improving the attractiveness scores of the current input. So far, prototype learning is not yet explored for modeling and improving digital marketing contents. Our method can be seen as learning prototypes that increase the regression scores, a new problem that we leave for future work.

Evaluating Attribution Methods. Recent research have proposed several metrics to evaluate attribution methods, which can be divided into two categories: Sanity Checks and Localization-Based Metrics. Sanity Checks [2, 3, 52] are designed to examine the basic properties of attribution methods according to faithfulness, stability and fairness. We aim at quantifying the effectiveness of attribution methods in real-world applications though. Hence our evaluation scheme examines the relationship between insights-guided modifications and the ensuing change in the actual success rate. Localization-Based Metrics measure how well attributions coincide with object bounding boxes or image grid cells that contains the key objects explaining the classification results [13, 15, 23]. In our scenario, we do not have the ground-truth bounding boxes, and our attribution methods explain the regression model. Thus localization-based metrics do not apply.

8 CONCLUSION

This paper constitutes the first attempt to use deep learning to facilitate the digital marketing design process. Our multimodal neural network outperforms competing methods in predicting success rates, and leverages neural attribution methods to provide insights that guide digital marketers to improve their existing design. Our approach is modular and generalizable, and individual neural components can be easily replaced as the state-of-the-art evolves. This work underscores the need to explore causal-aware models for modeling content experimentation, which we leave as future work. Additionally, our system's output insights can be further improved by high-capacity language and vision models such as ChatGPT [50] and SAM [37]. These models can provide clearer and more actionable instructions for human experts. Besides, our proposed insights evaluation methods may have broader impact on other real-world use-cases such as in healthcare, finance, bank sales etc. For example, quantifying the estimated contributions of biological risk factors on healthcare costs [42] or examining the effectiveness of a predicted business decision from an AI agent on the company's income/loss [4].

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their insightful comments. This research was supported by ONR N00014-18-1-2871-P00002-3. Special thanks to Amazon AWS for generously providing the computing resources necessary for this work.

REFERENCES

[1] Brian Ackerman and Yi Chen. 2011. Evaluating rank accuracy based on incomplete pairwise preferences. In *Proc. Workshop on UCERSTI Recsys*, Vol. 11.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems* 31 (2018).

[3] Chirag Agarwal, Eshika Saxena, Satyapriya Krishna, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. 2022. OpenXAI: Towards a Transparent Evaluation of Model Explanations. In *Conference on Neural Information Processing Systems (NeurIPS)*.

[4] Melda Alaluf, Giulia Crippa, Sinong Geng, Zijian Jing, Nikhil Krishnan, Sanjeev Kulkarni, Wyatt Navarro, Ronnie Sircar, and Jonathan Tang. 2022. Reinforcement Learning Paycheck Optimization for Multivariate Financial Goals. *Risk & Decision Analysis* (2022).

[5] Mohd Anas, Kailash Gupta, and Shafeeq Ahmad. 2017. Skin cancer classification using K-means clustering. *International Journal of Technical Research and Applications* 5, 1 (2017), 62–65.

[6] Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*. PMLR, 272–281.

[7] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research* 11 (2010), 1803–1831.

[8] Geoffrey H Ball and David J Hall. 1965. *ISODATA, a novel method of data analysis and pattern classification.* Technical Report. Stanford research inst Menlo Park CA.

[9] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 1–4.

[10] Antun Biloš, Davorin Turkalj, and Ivan Kelić. 2016. Open-rate controlled experiment in e-mail marketing campaigns. *Market-Tržište* 28, 1 (2016), 93–109.

[11] Aloïs Bissuel. 2020. Why your A/B-test needs confidence intervals. https://medium.com/criteo-engineering/why-your-ab-test-needs-confidence-intervals-bec9fe18db41.

[12] Ari Biswas, Thai T Pham, Michael Vogelsong, Benjamin Snyder, and Houssam Nassif. 2019. Seeker: Real-Time Interactive Search. In *International Conference on Knowledge Discovery and Data Mining (KDD)*. 2867–2875.

[13] Moritz Bohle, Mario Fritz, and Bernt Schiele. 2021. Convolutional dynamic alignment networks for interpretable classifications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10029–10038.

[14] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. 2017. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 661–670.

[15] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. 2015. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*. 2956–2964.

[16] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2019. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations (ICLR)*.

[17] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems* 32 (2019).

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT)*. 4171–4186.

[19] Inês Dutra, Houssam Nassif, David Page, Jude Shavlik, Roberta M Strigel, Yirong Wu, Mai E Elezaby, and Elizabeth Burnside. 2011. Integrating Machine Learning and Physician Knowledge to Improve the Accuracy of Breast Biopsy. In *American Medical Informatics Association Symposium (AMIA)*. 349–355.

[20] Bradley Efron. 2020. Prediction, estimation, and attribution. *International Statistical Review* 88 (2020), S28–S59.

[21] Tanner Fiez, Sergio Gamez, Arick Chen, Houssam Nassif, and Lalit Jain. 2022. Adaptive Experimental Design and Counterfactual Inference. In *Workshops of Conference on Recommender Systems (RecSys)*.

[22] Cher-Min Fong, Hui-Wen Wang, Chien-Hung Kuo, and Pei-Chun Hsieh. 2019. Image quality assessment for advertising applications based on neural network. *Journal of Visual Communication and Image Representation* 63 (2019), 102593.

[23] Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*. 3429–3437.

[24] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. 2019. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10705–10714.

[25] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. 2020. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. IEEE, 1–6.

[26] Sinong Geng, Zhaobin Kuang, Peggy Peissig, and David Page. 2018. Temporal poisson square root graphical models. *Proceedings of machine learning research* 80 (2018), 1714.

[27] Sinong Geng, Houssam Nassif, and Carlos A Manzanares. 2023. A Data-Driven State Aggregation Approach for Dynamic Discrete Choice Models. In *Uncertainty in Artificial Intelligence (UAI)*.

[28] Sinong Geng, Houssam Nassif, Carlos A Manzanares, A Max Reppen, and Ronnie Sircar. 2020. Deep PQR: Solving Inverse Reinforcement Learning using Anchor Actions. In *International Conference on Machine Learning (ICML)*. 3431–3441.

[29] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Counterfactual visual explanations. In *International Conference on Machine Learning*. PMLR, 2376–2384.

[30] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. 2020. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics* 37, 3 (2020), 362–386.

[31] Nicolas Grislain, Nicolas Perrin, and Antoine Thabault. 2019. Recurrent neural networks for stochastic control in real-time bidding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2801–2809.

[32] Abram Handler, Matthew Denny, Hanna Wallach, and Brendan O'Connor. 2016. Bag of what? simple noun phrase extraction for text analysis. In *Proceedings of the First Workshop on NLP and Computational Social Science*. 114–124.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[34] Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. 2016. Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *Proceedings of the 10th ACM conference on recommender systems*. 309–316.

[35] Andrew Jesson, Sören Mindermann, Uri Shalit, and Yarin Gal. 2020. Identifying causal-effect inference failure with uncertainty-aware models. *Advances in Neural Information Processing Systems* 33 (2020), 11637–11649.

[36] Keras. 2022. Global Average Pooling 2D. https://www.tensorflow.org/api_docs/python/tf/keras/layers/GlobalAveragePooling2D.

[37] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[38] Ron Kohavi and Roger Longbotham. 2017. Online Controlled Experiments and A/B Testing. *Encyclopedia of machine learning and data mining* 7, 8 (2017), 922–929.

[39] Janet L Kolodner. 1992. An introduction to case-based reasoning. *Artificial intelligence review* 6, 1 (1992), 3–34.

[40] Fanjie Kong and Ricardo Henao. 2022. Efficient Classification of Very Large Images with Tiny Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2384–2394.

[41] Thomas Langlois, Haicheng Zhao, Erin Grant, Ishita Dasgupta, Tom Griffiths, and Nori Jacoby. 2021. Passive attention in artificial neural networks predicts human visual selectivity. *Advances in Neural Information Processing Systems* 34 (2021), 27094–27106.

[42] Jiwoo Lee, Sakari Jukarainen, Padraig Dixon, Neil M Davies, George Davey Smith, Pradeep Natarajan, and Andrea Ganna. 2022. Quantifying the causal impact of biological risk factors on healthcare costs. *medRxiv* (2022), 2022–11.

[43] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[44] Zhaoqi Li, Lillian Ratliff, Houssam Nassif, Kevin Jamieson, and Lalit Jain. 2022. Instance-Optimal PAC Algorithms for Contextual Bandits. In *Conference on Neural Information Processing Systems (NeurIPS)*.

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

[46] Luke Merrick. 2019. Randomized ablation feature importance. *arXiv preprint arXiv:1910.00174* (2019).

[47] Sareh Nabi, Houssam Nassif, Joseph Hong, Hamed Mamani, and Guido Imbens. 2022. Bayesian Meta-Prior Learning Using Empirical Bayes. *Management Science* 68, 3 (2022), 1737–1755.

[48] Wei Niu, James Caverlee, and Haokai Lu. 2018. Neural personalized ranking for image recommendation. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 423–431.

[49] Mohammad Nuruzzaman and Omar Khadeer Hussain. 2018. A survey on chatbot implementation in customer service industry through deep neural networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. IEEE, 54–61.

[50] OpenAI. 2023. GPT-4 Technical Report. *ArXiv* abs/2303.08774 (2023).

[51] Sameer Ranjan, Deepak Ranjan Nayak, Kallepalli Satish Kumar, Ratnakar Dash, and Banshidhar Majhi. 2017. Hyperspectral image classification: A k-means

clustering based approach. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 1–7.

[52] Sukrut Rao, Moritz Böhle, and Bernt Schiele. 2022. Towards better understanding attribution methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10223–10232.

[53] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation*. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science.

[54] Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. 2019. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances* 5, 11 (2019), eaau4996.

[55] Neela Sawant, Chitti Babu Namballa, Narayanan Sadagopan, and Houssam Nassif. 2018. Contextual Multi-Armed Bandits for Causal Marketing. In *Workshops of International Conference on Machine Learning (ICML)*.

[56] Rainer Schmidt, Stefania Montani, Riccardo Bellazzi, Luigi Portinale, and Lothar Gierl. 2001. Cased-based reasoning for medical knowledge-based systems. *International Journal of Medical Informatics* 64, 2-3 (2001), 355–367.

[57] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[58] Steve Seymour. 2016. Why Resonance Is Vital For Your Digital Marketing Success. https://www.linkedin.com/pulse/why-resonance-vital-your-digital-marketing-success-steve-seymour/.

[59] Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, and Shikha Jain. 2017. Machine translation using deep learning: An overview. In *2017 international conference on computer, communications and electronics (comptelix)*. IEEE, 162–167.

[60] Moumita Sinha, Jennifer Healey, and Tathagata Sengupta. 2020. Designing with AI for digital marketing. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 65–70.

[61] A Razia Sulthana, Maulika Gupta, Shruthi Subramanian, and Sakina Mirza. 2020. Improvising the performance of image-based recommendation system using convolution neural networks and deep learning. *Soft Computing* 24, 19 (2020), 14531–14544.

[62] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[63] Robert L Thorndike. 1953. Who belongs in the family. In *Psychometrika*. Citeseer.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 6000—-6010.

[65] Lezi Wang, Ziyan Wu, Srikrishna Karanam, Kuan-Chuan Peng, Rajat Vikram Singh, Bo Liu, and Dimitris Metaxas. 2019. Sharpen Focus: Learning With Attention Separability and Consistency. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 512–521.

[66] Rui Wang, Tong Yu, Handong Zhao, Sungchul Kim, Subrata Mitra, Ruiyi Zhang, and Ricardo Henao. 2022. Few-Shot Class-Incremental Learning for Named Entity Recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 571–582.

[67] Tian Wang, Yuri M Brovman, and Sriganesh Madhvanath. 2021. Personalized embedding-based e-commerce recommendations at eBay. *arXiv preprint arXiv:2102.06156* (2021).

[68] Wei Wang. 2022. Data Marketing Optimization Method Combining Deep Neural Network and Evolutionary Algorithm. *Wireless Communications and Mobile Computing* 2022 (2022).

[69] Lifeng Zhang. 2021. Absolute Neighbour Difference based Correlation Test for Detecting Heteroscedastic Relationships. *Advances in Neural Information Processing Systems* 34 (2021), 25452–25462.

[70] Lichun Zhou. 2020. Product advertising recommendation in e-commerce based on deep learning and distributed expression. *Electronic Commerce Research* 20, 2 (2020), 321–342.

APPENDIX

A DETAILED INSIGHT EVALUATION ALGORITHMS

Algorithm 3: Evaluate attribution results of a text model.

Data: Input dataset $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ where $x_i \in \mathbb{R}^n$ is the input and $y_i \in \mathbb{R}$ is the label, and their corresponding attribution maps $\{C_1, C_2, ..., C_n\}$ where $C_i \in \mathbb{R}^n$.

Result: Correlation coefficient ρ.

Initialize $k \leftarrow 0$, $\mathbf{d_C} \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$ and $\mathbf{d_Y} \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$.

1) Find the difference:

For every pair of control and treatment $\{x_i, x_j\}$ **in dataset do**:

i. Compute the distinct elements set $S_{i,j}$, such that the attributes in $S_{i,j}$ can be only found in x_i or x_j.

$S_i \leftarrow \text{set}(x_i), \;\; S_j \leftarrow \text{set}(x_j)$;

$S_{i,j} \leftarrow S_i \cup S_j - S_i \cap S_j$;

ii. Compute P_i and P_j, indicator vectors where $P_i := \{p_i^s\}_{s=1,2,...n}$ such that $p_i^s = 1$ if $x_i^s \in S_{i,j}$ and $p_i^s = 0$ if $x_i^s \notin S_{i,j}$, and $P_j := \{p_j^s\}_{s=1,2,...n}$ such that $p_j^s = 1$ if $x_j^s \in S_{i,j}$ and $p_j^s = 0$ f $x_j^s \notin S_{i,j}$.

iii. Compute $d_C = \text{sign}(y_i - y_j)(P_i^T C_i - P_j^T Cj)$ as the sum of predicted attributions difference, and $d_y = |y_i - y_j|$ as the actual success rate improvements.

Update:

$\mathbf{d_C}[k] \leftarrow d_C, \;\; \mathbf{d_Y}[k] \leftarrow d_y$;

$k \leftarrow k + 1$;

end ;

2) Compute Pearson Correlation ρ between $\mathbf{d_C}$ and $\mathbf{d_Y}$.

Output ρ.

Algorithm 4: Evaluate attribution results of an image model.

Data: Input dataset $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ where $x_i \in \mathbb{R}^{m \times n \times Z}$ is an RGB input image and $y_i \in \mathbb{R}$ is the label of the image, their corresponding attribution maps $\{C_1, C_2, ..., C_n\}$ where $C_i \in \mathbb{R}^{m \times n}$, and the vision model $\Phi(\cdot)$ that can extract features of input images.

Result: Correlation coefficient ρ.

Initialize $k \leftarrow 0$, $\mathbf{d_C} \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$ and $\mathbf{d_Y} \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$.

1) Find the difference:

For every pair of control and treatment $\{x_i, x_j\}$ **in dataset do**:

i. Compute the feature maps of inputs:

$\mathbf{A}_i \leftarrow \Phi(x_i), \mathbf{A}_j \leftarrow \Phi(x_j)$;

reshape $\mathbf{A}_i$ and $\mathbf{A}_j$:

$\mathbf{A}_i \leftarrow \text{reshape}(\mathbf{A}_i, (m'n', Z'))$,

$\mathbf{A}_j \leftarrow \text{reshape}(\mathbf{A}_i, (Z', m'n'))$;

ii. Compute the Cosine Similarity matrix between every feature vector in $\mathbf{A}_i$ and every feature vector in $\mathbf{A}_j$:

$$\mathbf{S}_{i,j}[k, l] \leftarrow \frac{< \mathbf{A}_i[k, :], \mathbf{A}_j[:, l] >}{|\mathbf{A}_i[k, :]||\mathbf{A}_j[:, l]|},$$

$\forall k = 0, 1, 2, ..., m'n', \quad \forall l = 0, 1, 2..., m'n'$;

iii. Take the maximum similarity scores for each location in x_i and x_j:

$\mathbf{d}_{x_i}[i] \leftarrow \max_l \mathbf{S}[i, l], \quad \forall i = 0, 1, 2..., m'n'$;

$\mathbf{d}_{x_j}[j] \leftarrow \max_k \mathbf{S}[k, j], \quad \forall j = 0, 1, 2..., m'n'$

iv. Threshold $\mathbf{d}_{x_i}$, $\mathbf{d}_{x_j}$ and resize them to the same dimension as C_i, C_j:

$P_i \leftarrow \text{threshold}(\mathbf{d}_{x_i}), \;\; P_j \leftarrow \text{threshold}(\mathbf{d}_{x_j})$;

Resize P_i and P_j:

$P_i \leftarrow \text{resize}(P_i, (m, n)), \;\; P_j \leftarrow \text{resize}(P_j, (m, n))$;

v. Compute predicted attribution difference d_C and acutal success rate improvement d_y:

$d_C = \text{sign}(y_i - y_j)(\sum P_i \odot C_i - \sum P_j \odot Cj)$;

$d_y = |y_i - y_j|$;

Update:

$\mathbf{d_C}[k] \leftarrow d_C, \;\; \mathbf{d_Y}[k] \leftarrow d_y$;

$k \leftarrow k + 1$;

end ;

2) Compute Pearson Correlation ρ between $\mathbf{d_C}$ and $\mathbf{d_Y}$.

Output ρ.

B NEURAL NETWORK ARCHITECTURE DETAILS

In Table 3, the convolutional layer is denoted as "Conv", followed by the kernel size, stride, padding and number of filters. "fc" means fully-connected layer and the output hidden units is provided after the dash. "ELU", "ReLU" and "Sigmoid" represent the non-linear functions. "GlobalAveragePooling2D" is the global average pooling operation in the spatial dimension of the tensors, functioning the same as Keras' Global Average Pooling 2D [36]. "ResBlock" is the standard ResNet block [33]. In the brackets, we provide the kernel size, stride, and number of filters. "TransformerLayer" is the standard layer in a transformer [64]. In the brackets, we provide the size of hidden layers and the number of attention heads.

Table 3: The architecture of each component in our multimodal neural network.

ResNet$(\cdot)$

Layer	Type
1	Conv(3, 1, 1)-32 + ReLU()
2	ResBlock(3, 1, 32)
3	ResBlock(3, 2, 32)
4	ResBlock(3,2, 32)
5	ResBlock(3,2, 32)
6	BatchNorm()+ReLU()
7	GlobalAveragePooling2D()

BERT$(\cdot)$

Layer	Type
1-12	TransformerLayers(768, 12)

$\text{MLP}_1(\cdot)$

Layer	Type
1	fc-512 + BatchNorm + ELU()
2	fc-1024 + BatchNorm + ELU()
3	fc-1024 + BatchNorm + ELU()
4	fc-512 + BatchNorm + ELU()

$\text{MLP}_2(\cdot)$

Layer	Type
1	fc-512 + BatchNorm + ELU()
2	fc-1024 + BatchNorm + ELU()
3	fc-1024 + BatchNorm + ELU()
4	fc-512 + BatchNorm + ELU()

$\text{MLP}_3(\cdot)$

Layer	Type
1	fc-512 + BatchNorm + ELU()
2	fc-1024 + BatchNorm + ELU()
3	fc-1024 + BatchNorm + ELU()
4	fc-1 + Sigmoid()

C ADDITIONAL RESULTS

In this section, we offer an additional result to support the finding in our main paper. Specifically, Figure 8 compares RMSE and MAE of our multimodal neural network and the Generalized Linear Model (GLM) on each domain. Notably, each bar for RMSE and MAE is only computed on multimodal data from each respective domain. The objective here is to underscore the consistent error reduction achieved by our multimodal neural network across a variety of domains.

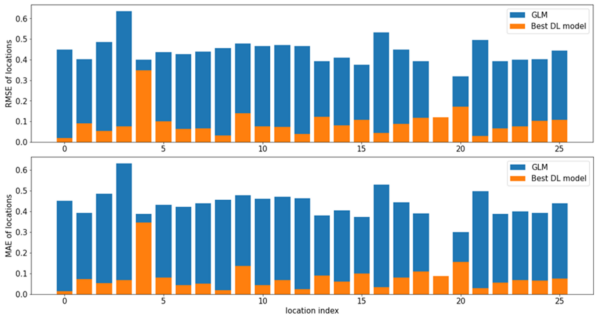

Figure 8: RMSE (top) and MAE (bottom) of GLM (blue) and our multimodal neural network (orange) evaluated on each domain.