ADVANCING VISION INTELLIGENCE THROUGH THE DEVELOPMENT OF EFFICIENCY, INTERPRETABILITY AND FAIRNESS IN DEEP LEARNING MODELS

by

Fanjie Kong

Department of Electrical and Computer Engineering
Duke University

Defense Date: April 1, 2024

Approved:

Ricardo Henao, Supervisor

Hai Li, Co-Supervisor

Yiran Chen

Stacy Tatum

David Carlson

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the Graduate School of Duke University
2024

ABSTRACT

ADVANCING VISION INTELLIGENCE THROUGH THE DEVELOPMENT OF EFFICIENCY, INTERPRETABILITY AND FAIRNESS IN DEEP LEARNING MODELS

by

Fanjie Kong

Department of Electrical and Computer Engineering
Duke University

Defense Date: April 1, 2024

Approved:

Ricardo Henao, Supervisor

Hai Li, Co-Supervisor

Yiran Chen

Stacy Tatum

David Carlson

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2024

Abstract

Deep learning has demonstrated remarkable success in developing vision intelligence across a variety of application domains, including autonomous driving, facial recognition, medical image analysis, *etc.* However, developing such vision systems poses significant challenges, particularly in relation to ensuring efficiency, interpretability, and fairness. Efficiency requires a model to leverage the least possible computational resources while preserving performance relative to more computationally-demanding alternatives, which is essential for the practical deployment of large-scale models in real-time applications. Interpretability demands a model to align with the domain-specific knowledge of the task it addresses while having the capability for case-based reasoning. This characteristic is especially crucial in high-stakes areas such as healthcare, criminal justice, and financial investment. Fairness ensures that computer vision models do not perpetuate or exacerbate societal biases in downstream applications such as web image search, text-guided image generation, *etc.* In this dissertation, I will discuss the contributions that I have made in advancing vision intelligence regarding to efficiency, interpretability and fairness in computer vision models.

The first part of this dissertation will focus on how to design computer vision models to efficiently process very large images. We propose a novel CNN architecture termed *Zoom-In Network* that leverages a hierarchical attention sampling mechanisms to select important regions of images to process. Such approach without processing the entire image yields outstanding memory efficiency while maintaining classification accuracy on various tiny object image classification datasets.

The second part of this dissertation will discuss how to build post-hoc interpretation method for deep learning models to obtain insights reasoned from the predictions. We propose a novel image and text insight-generation framework based on attributions from deep neural nets. We test our approach on an industrial dataset and demonstrate our method outperforms competing methods.

Finally, we study fairness in large vision-language models. More specifically, we examined gender and racial bias in text-based image retrieval for neutral text queries. In an attempt to

address bias in the test-time phase, we proposed post-hoc bias mitigation to actively balance the demographic group in the image search results. Experiments on multiple datasets show that our method can significantly reduce bias while maintaining satisfactory retrieval accuracy at the same time.

My research in enhancing vision intelligence via developments in efficiency, interpretability, and fairness, has undergone rigorous validation using publicly available benchmarks and has been recognized at leading peer-reviewed machine learning conferences. This dissertation has sparked interest within the AI community, emphasizing the importance of improving computer vision models through these three critical dimensions, namely, efficiency, interpretability and fairness.

Contents

List of Tables

List of Figures

Acknowledgements

Pursuing my Ph.D. at Duke University has been a journey filled with joy and excitement, and choosing Professor Ricardo Henao as my academic advisor is the best decision I have ever made. I want to express my heartfelt thanks to Professor Henao for his guidance and support. Transitioning from a master student in Biomedical Engineering to pursuing an academic career in machine learning was a difficult leap. Professor Henao trust in my potential to succeed in machine learning, despite my non-computer science background, which grants me the invaluable opportunity to delve into this field. His trust paved the way for my successful completion of the Ph.D. program in machine learning. His mentorship has been instrumental in my academic career, profoundly influencing my research and future career development.

Secondly, I would also like to express my sincere appreciation to Prof. Hai Li, Prof. Yiran Chen, and Prof. David Carlson, along with Prof. Stacy Tatum, for their invaluable contributions to my thesis committee. Additionally, my heartfelt thanks go to Prof. Henry Pfister and Prof. Zhengqiang Gong for graciously serving on in my preliminary exam committee. Their readiness to participate and their insightful contributions have been a tremendous source of encouragement and inspiration.

Thirdly, my gratitude extends to my colleagues Dr. Yanbei Chen, Dr. Jiarui Cai, Dr. Davide Modolo, Dr. Yuan Li, Dr. Houssam Nassif, Dr. Tanner Fiez and Dr. Shreya Chakrabarti at Amazon for their collaboration during my internship and for co-authoring two papers with me.

Lastly, my deepest appreciation goes to my family, whose unwavering support has been my stronghold throughout my academic endeavors. I am especially grateful to my parents, Mrs. Xiao and Mr. Kong, for their endless patience and selflessness in raising me. Studying in the pandemic presented unique challenges, but the emotional support from my parents made the journey more manageable. Furthermore, I wish to express my heartfelt appreciation to my cherished emotional support dog, Piggie. Her constant companionship throughout my Ph.D. journey fullfilled my days with joy and power. When I am depressed

by the heavy work, Piggie was there to rekindle my spirits. Her unwavering support was indispensable in my Ph.D. journey.

1. Introduction

Throughout the past decade, the field of computer vision has advanced substantially driven by new developments in deep learning methodology. Key innovations in model architectures and learning algorithms (Krizhevsky et al., 2012; He et al., 2016a) have enabled deep networks to achieve or even surpass human-level performance in various visual tasks, such as image classification (Dosovitskiy et al., 2020), object detection (Zou et al., 2023), image segmentation (Minaee et al., 2021), video understanding (Kong and Fu, 2022), and 3D perception (Guo et al., 2020). However, the requirements of real-world applications vastly differ from those of performance competitions or research settings. While models may achieve state-of-the-art accuracy in an ideal setting, challenges related to efficiency, interpretability, and fairness arise when applied in practical scenarios (Paleyes et al., 2022). For instance, recent vision models, such as vision transformers, face significant challenges in computational intensity and demands for annotated data (Wang et al., 2023b). Moreover, the intrinsic black-box nature of these models hinders interpretability (Guidotti et al., 2018), while their tendency to perpetuate or even amplify biases towards certain demographic groups in their predictions poses challenges for widespread adoption and trust (Lee et al., 2023). These issues related to efficiency, interpretability and fairness pose significant challenges when deploying deep computer vision models in real-world scenarios.

In pursuit of efficiency in computer vision models, current research efforts have been broadly categorized into four directions. Firstly, compression methods (Yu et al., 2018), aim to minimize the redundancy of weights in trained models. Secondly, lightweight model designs (Jaderberg et al., 2014) replace network components with computationally more efficient alternatives through for instance, low-rank approximation techniques. Thirdly, partial computation techniques (Larsson et al., 2017), focus on the selective activation of network units, which in turn modulates the computational load during forward propagation (inference). Lastly, the use of reinforcement learning and attention mechanisms (Ramapuram et al., 2018; Levi and Ullman, 2018; Uzkent and Ermon, 2020; Katharopoulos and Fleuret, 2019; Cordonnier et al., 2021), focuses on selectively processing input subsets based on their

relevance to a specific task. My work leverages attention mechanism to automatically select partial input, thereby achieving significant savings in memory usage during computation.

Recent research has increasingly focused on the interpretability of deep learning models, aiming to close the gap between model performance and human comprehension (Zhang et al., 2021). Interpretation methods can help identify data and model deficiencies (Anders et al., 2022; Hägele et al., 2020). Interpretable models allow users to verify that model predictions are generated using biologically concordant features supported by scientific evidence and identified by human experts. In decision-support use-cases involving a human expert, the algorithm needs to give a visual cue highlighting regions that require closer examination. In these applications, a predicted score is insufficient and needs to be complemented with a highlighted visual region associated with the model's prediction (Kong et al., 2023b).

The concept of fairness in vision intelligence has gained attention following revelations of bias in vision-language (VL) model applications (Wang et al., 2021a; Hall et al., 2023; Wang et al., 2021b). Approaches that encourage fairness in VL models range from pre-processing methods, which balance demographic groups in training data while maintaining model performance (Friedler et al., 2014; Calmon et al., 2017), to in-processing strategies that integrate fairness constraints within the training process (Berg et al., 2022; Wang et al., 2023a; Xu et al., 2021; Cotter et al., 2019). Post-processing techniques address biases in pre-trained models either through corrections Cheng et al. (2021); Calmon et al. (2017) or feature clipping based on mutual information in image-text encoders Wang et al. (2021a). Our work lies in the post-processing category of debiasing methods that encourage equal representation of diverse demographics. We extend the fair subset selection approach for text-based image retrieval proposed by Mehrotra and Celis (2021) that overcomes the need for precise demographic attributes annotations.

In the following sections of this dissertation, we focus on developing efficient, interpretable and fair computer vision models across various application contexts. In Chapter 2, we develop an efficient computer vision model to process very large images that significantly saves memory during inference, while maintaining high performance. Chapter 3 introduces an

innovative methodology for interpreting predictions from neural networks, which further aids in enhancing human understanding of input improvement. This chapter also demonstrates the application of this methodology in the context of digital marketing content design. In Chapter 4, we delve into the fairness problem within deep learning-based image search systems. Then, we propose a novel approach aimed at mitigating biases in image retrieval outputs, which has the potential to significantly influence the development of fair machine learning algorithms. The contributions in this dissertation have been presented in top peer-reviewed conferences in machine learning (Kong and Henao, 2022; Kong et al., 2023b,a).

2. Efficient Classification of Very Large Images with Tiny Objects

2.1 Introduction

Neural networks have achieved state-of-the-art performance in many image classification tasks Krizhevsky et al. (2012). However, there are still many scenarios where neural networks can still be improved. Using modern deep neural networks on image inputs of very high resolution is a non-trivial problem due to the challenges of scaling model architectures Tan and Le (2019). Such images are common for instance in satellite or medical imaging. Moreover, these images tend to become even bigger due to the rapid growth in computational and memory availability, as well as the advancements in camera sensor technology.

Specifically challenging are the so called *tiny object image classification* tasks, where the goal is to classify images based on the information of very small objects or regions of interest (ROIs), in the presence of a much larger and rich (*non-trivial*) background that is uncorrelated or non-informative of the label. Consequently, constituting an input image with a very low ROI-to-image ratio. Recent work Pawlowski et al. (2019) showed that with a dataset of limited size, convolutional neural networks (CNNs) have poor performance on very low ROI-to-image ratio problems. In these settings, the input resolution is increased from typical image sizes, *e.g.*, 224×224 pixels, to *gigapixel* images of size ranging from $45,056 \times 35,840$ to $217,088 \times 111,104$ pixels Litjens et al. (2018), which not only require significantly more computational processing power per image than a typical image given a fixed deep architecture, but in some cases, become prohibitive for current GPU-memory standards.

Figure 2.1 shows an example of a gigapixel pathology image, from which we see that manually annotated ROIs (with cancer metastases), not usually available for model training, constitute a small proportion of the whole slide image (WSI). Moreover, many tasks in satellite imagery Christie et al. (2018) and medical image analysis Litjens et al. (2018) are still challenging due to the scarce methodology available for such big images.

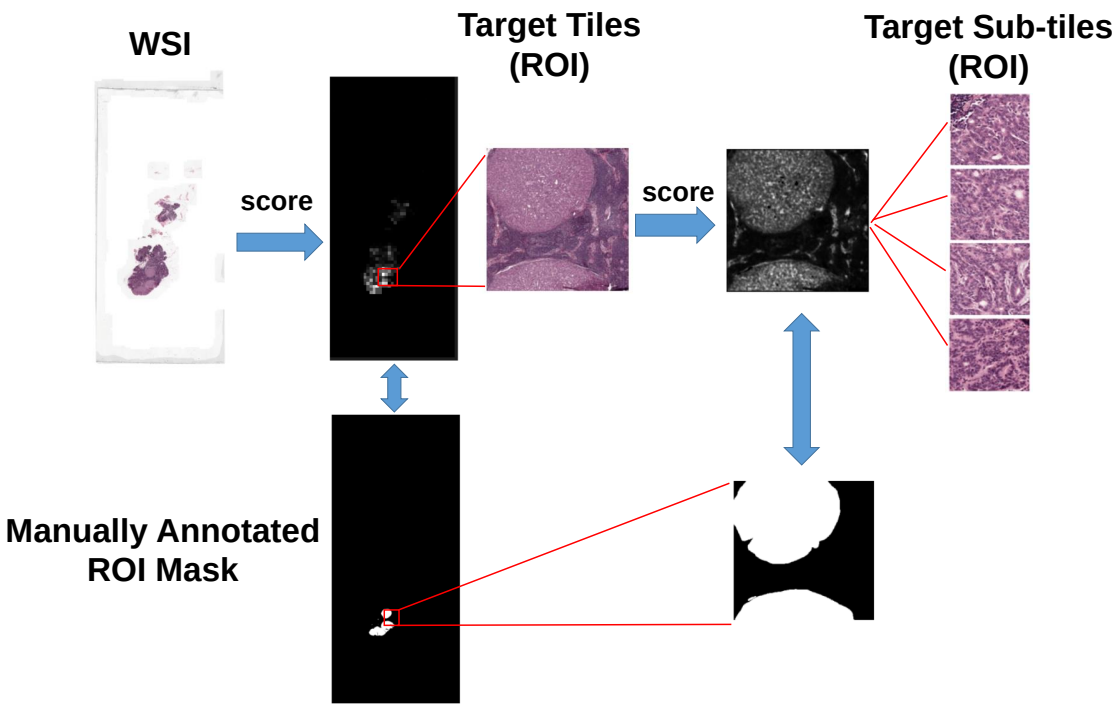


FIGURE 2.1: Illustration of processing a typical WSI using our zoom-in strategy. We see that *i*) there are large regions with little information (mostly background), and *ii*) small informative regions have high-resolution details. Leveraging the above characteristics of WSI, we derive a method that gradually zooms-in to the ROI. The proposed approach first process the down-sampled WSI to sample the target tiles, and then repeats this procedure to sample target sub-tiles. The sampled sub-titles contain the fine-grained information for classification. The bottom images show that the manually annotated ROIs are captured by the proposed approach without the need for pixel level annotations.

Other recent works have addressed the computational resource bottlenecks associated with models for very large images by proposing approaches such as the streaming neural network Pinckaers et al. (2019) and gradient checkpoint Marra et al. (2020). However, these methods do not take advantage of the characteristics of very large images in tiny object image classification tasks, *i.e.*, those in which only a small portion of the image input is informative for the classification label of interest. Alternatively, other approaches use visual attention models to exploit these characteristics and show that discriminative information may be sparse and scattered across various image scales Katharopoulos and Fleuret (2019); Fu et al. (2017); Papadopoulos et al. (2021), which suggests that in some scenarios, processing the entire input image is unnecessary, and specially true in tiny object image classification tasks.

For instance, Katharopoulos and Fleuret (2019) leverages attention to build image classifiers using a small collection of tiles (image patches) *sampled* from the matrix of attention weights generated by an attention network. Unfortunately, despite the ongoing efforts, existing approaches are either prohibitive or require severe resolution trade-offs that ultimately affect classification performance, for tasks involving very large (gigapixel) images.

The purpose of this work is to address these limitations simultaneously. Specifically, we propose a neural network architecture termed *Zoom-In network*, which as we will show, yields outperforming memory efficiency and classification accuracy on various tiny object image classification datasets. We build upon Katharopoulos and Fleuret (2019) by proposing a two-stage *hierarchical* attention sampling approach that is effectively able to process gigapixel images, while also leveraging contrastive learning as a means to improve the quality of the attention mechanisms used for sampling. This is achieved by building aggregated representations over a small fraction of high-resolution content (sub-tiles) that is selected from an attention mechanism, which itself leverages a lower resolution view of the original image. In this way, the model can dramatically reduce the data acquisition and storage requirements in real-world deployments. This is possible because low resolution views can be used to indicate which regions of the image should be acquired (attended) at higher resolution for classification purposes, without the need of acquiring the entire image at full resolution. Moreover, we show that the proposed approach can be easily extended to incorporate pixel-level-annotations when available for additional performance gains. Results on five challenging datasets demonstrate the capabilities of the Zoom-In network in terms of accuracy and memory efficiency.

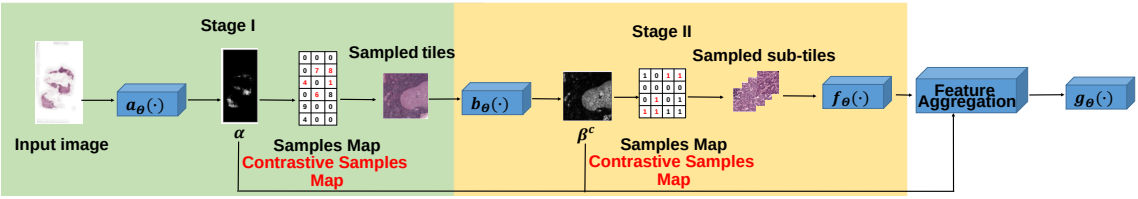


FIGURE 2.2: Illustration of the Zoom-In network. In Stage I, attention network $a_\Theta(\cdot)$ generates an attention map for the input image down-scaled by s_1, from which N tiles are sampled with replacement (see samples map). In Stage II, attention network $b_\Theta(\cdot)$ generates an attention map for each selected tile and selects a sub-tile, thus N sub-tiles are selected (without replacement). Then all sub-tiles are fed to feature extractor $f_\Theta(\cdot)$, feature maps are aggregated using their corresponding attention weights, and predictions are obtained from aggregated features using a classification module $g_\Theta(\cdot)$. Further, both attention maps are also used to draw contrastive samples with minimal computational overhead (during training).

2.2 Zoom-In Network

Below we present the construction of the proposed Zoom-In network model, which aims to efficiently process gigapixel images for classification of very large images with tiny objects. We start by briefly describing the one-stage attention sampling method proposed in Katharopoulos and Fleuret (2019), which we leverage in our formulation. Then, we introduce our strategy consisting in decomposing the attention-based sampling into two stages as illustrated in Figure 2.2. This *two-stage hierarchical sampling* approach enables computational efficiency without the need to sacrifice performance due to loss of resolution, when used in applications with very large images and small ROI-to-image ratios. In the experiments, we will show that the Zoom-In network results in improved performance relative to existing approaches on several tiny object image classification datasets, and importantly, without the need for any pixel-level annotation.

2.2.1 Attention Sampling

Let $T_{s_1}(x, c)$ denote a function that extracts a tile of size $h_1 \times w_1$ from the input, full-resolution, image $x \in \mathbb{R}^{H \times W}$ corresponding to the location (coordinates) $c = \{i, j\}$ in a lower resolution view $V(x, s_1) \in \mathbb{R}^{h \times w}$ of x at scale $s_1 \in (0, 1)$, so $h = \lfloor s_1 H \rfloor$ and $w = \lfloor s_1 W \rfloor$, where $\lfloor \cdot \rfloor$ is the floor operator. More specifically, $T_{s_1}(x, c)$ maps c to a location in x via

$\{\lfloor 1+(i-1)(W-1)/(w-1)\rfloor, \lfloor 1+(j-1)(H-1)/(h-1)\rfloor\}$, and returns a tile of size $h_1 \times w_1$. Note that *i*) the map of locations between $V(x, s_1)$ and x only depends on the size of x ($H \times W$) and s_1 and not on the tile size ($h_1 \times w_1$); *ii*) $h_1, w_1 > 1/s_1$, to guarantee full coverage of x; *iii*) this strategy requires to zero-pad x on all sides by $\lfloor h_1/2 \rfloor$ and $\lfloor w_1/2 \rfloor$ pixels accordingly; and *iv*) we have omitted the (color) channel dimension in x and $V(x, s_1)$ for notational simplicity, however, we consider color images (with an additional dimension) in our experiments. Then, let $\Psi_\Theta(x) = g_\Theta(f_\Theta(T_{s_1}(x, c)))$ be a neural network parameterized by Θ whose intermediate representation $z \in \mathbb{R}^K$ is obtained via feature extracting function $z = f_\Theta(T_{s_1}(x, c))$, *e.g.*, a convolutional neural network (CNN). Further, $g_\Theta(z)$ is a classification function also specified as a neural network and parameterized by Θ. We can provide $\Psi_\Theta(x)$ with an attention mechanism as follows

$$\alpha = a_\Theta(V(x, s_1)) : \mathbb{R}^{h \times w} \rightarrow \mathbb{R}^{h \times w} \tag{2.1}$$

$$\Psi_\Theta(x) = g_\Theta\left(\sum_{c \in C} \alpha_c f_\Theta(T_{s_1}(x, c))\right), \tag{2.2}$$

where α is the matrix of attention weights such that $\sum_{c \in C} \alpha_c = 1$, $a_\Theta(V(x, s_1))$ is the attention function, also specified as a neural network, and C (of length $|C| = h \cdot w$) is the collection of all index pairs for view $V(x, s_1)$. In order to avoid computing the features z from all $|C|$ tiles implied by view $V(x, s_1)$, which can be a very large number if x is big, as in our pathology and remote sensing scenarios, Katharopoulos and Fleuret (2019) proposed to leverage Monte Carlo estimation by only considering a small set of tiles from the original input image sampled via the attention function. This strategy leverages that α defines a discrete distribution over the set of $|C|$ tiles. Specifically, Katharopoulos and Fleuret (2019) approximates (2.2) by sampling from (2.1) via

$$\Psi_\Theta(x) \approx g_\Theta\left(\frac{1}{N}\sum_{c \in Q} f_\Theta(T_{s_1}(x, c))\right), \tag{2.3}$$

where Q is a collection of $N \ll |C|$ index pairs for view $V(x, s_1)$ drawn independently and identically distributed (iid) from the distribution defined by the attention weights, *i.e.*,

$Q = \{(i,j) \sim \ a_\Theta(V(x,s_1))|i = 1,2,...,N\}$. In Katharopoulos and Fleuret (2019), they consider tiles of size $h_1 = w_1 = 27$, $s_1 = 0.2$ and $N = 10$ for the colon cancer dataset. See Experiments below for additional details.

Using the approximation in (2.3), the attention mechanism uses a lower resolution view $V(x,s_1)$ of the original image x for computing the attention distribution and outputs an aggregated feature vector by averaging over the features $\{z_n\}_{n=1}^N$ of a small amount of N tiles. Unfortunately, this approach is still prohibitive for gigapixel images because feasible combinations of h_1, w_1 and s result in unrealistic memory needs for current GPU-memory standards. Below, we introduce the proposed two-stage hierarchical sampling to improve the memory efficiency of attention sampling.

2.2.2 Two-stage Hierarchical Attention Sampling

Multistage and hierarchical sampling strategies are often preferred in practice. For instance, the cost of interviewing or testing people are enormously reduced if these people are geographically or organizationally grouped, thus sampling is performed within groups (clusters). Such sampling design has many real-world applications such as household and mortality surveys, as well as high-resolution remote sensing applications Clark (2009); Galway et al. (2012); Xia et al. (2019). Motivated by this idea, we design a two-stage hierarchical sampling approach to reduce memory requirements when processing very large, gigapixel, images without severe resolution trade-offs.

Specifically, let $V(x,s_2,c) \in \mathbb{R}^{u\times v}$ be a view of $T_{s_1}(x,c)$ at scale $s_2 \in (0,1)$, so $u = \lfloor s_2 h \rfloor = \lfloor s_1 s_2 H \rfloor$ and $v = \lfloor s_2 w \rfloor = \lfloor s_1 s_2 W \rfloor$. Further, we define a function $T_{s_2}(T_{s_1}(x,c),c')$ that extracts a sub-tile of size $h_2 \times w_2$ at location $c' = \{i',j'\}$ in $V(x,s_2)$ from tile $T_{s_1}(x,c)$ at location $c = \{i,j\}$ in $V(x,s_1)$. The mapping function $T_{s_2}(T_{s_1}(x,c),c')$ is defined similarly to $T_{s_1}(x,c)$, but returns tiles of size $h_2 \times w_2$ instead of $h_1 \times w_1$, and is such that $h_2 < h_1$, $w_2 < w_1$, and $h_2, s_2 > 1/s_2$. Moreover, we can also define an attention mechanism for $V(x,s_2,c)$ as in (2.1) as follows

$$\beta^c = b_\Theta(V(x,s_2,c)) : \mathbb{R}^{u\times v} \to \mathbb{R}_+^{u\times v}, \tag{2.4}$$

where β is the matrix of attention weights for the tile at location c of $V(x, s_1)$ such that $\sum_{c' \in C'} \beta'_c = 1$, $b_\Theta(V(x, s_2, c))$ is the attention function, also specified as a neural network, and C' (of length $|C'| = u \cdot v$) is the collection of all index pairs for view $V(x, s_2, c)$ of $T_{s_1}(x, c)$. Provided that $\sum_{c \in C} \alpha_c = 1$ in (2.1) and $\sum_{c' \in C'} \beta'_c = 1$ in (2.4), it is easy to see that $\sum_{c \in C} \sum_{c' \in C'} a_\Theta(V(x, s_1)) b_\Theta(V(x, s_2, c)) = 1$ and that the attention for location $c' = \{i', j'\}$ in $V(x, s_2, c)$ relative to the entire image x is $\alpha_c \beta_{c'}$. Consequently, we can rewrite (2.2) as

$$\Psi_\Theta(x) = g_\Theta \left(\sum_{c \in C} \alpha_c \sum_{c' \in C'} \beta^c_{c'} f_\Theta(T_{s_2}(T_{s_1}(x, c), c')) \right), \tag{2.5}$$

where now, the aggregated representation is a weighted average of all tiles of size $h_2 \times w_2$ of x, and like in (2.3), we can approximate as

$$\Psi_\Theta(x) \approx g_\Theta \left(\frac{1}{N} \sum_{c \in Q} f_\Theta(T_{s_2}(T_{s_1}(x, c), c')) \right), \tag{2.6}$$

where $c' \sim b_\Theta(V(x, s_s, c))$ is drawn iid from distribution $b_\Theta(V(x, s_s, c))$ for every location $c \in Q$.

Note that the approximation in (2.6) uses *full-resolution* sub-tiles from x that are drawn hierarchically from the two-level discrete distribution implied by α and $\{\beta^c\}_{c=1}^{|C|}$, which are obtained from *low-resolution* views $V(x, s_1)$ and $\{V(x, s_2, c)\}_{c=1}^{|C|}$. Importantly, in practice we do not need to instantiate the tiles $T_{s_1}(x, c)$ but only $T_{s_2}(T_{s_1}(x, c), c')$, and the second-level attention matrix in (2.4) can be obtained as needed (*on the fly*). However, this can cause computational inefficiency if multiple samples from the same location c are selected for level-two sampling in (2.6). Inefficiency occurs because such procedure will require to instantiate view $V(x, s_2, c)$ multiple times to obtain β^c on a single model update (iterations), and then when a sub-tile is sampled multiple times when obtaining $f_\Theta(T_{s_2}(T_{s_1}(x, c), c')$. We can mitigate the inefficiency by ordering the samples in Q to prevent recalculating β^c, and we can reuse features $f_\Theta(T_{s_2}(T_{s_1}(x, c), c')$ for a given c and c' as needed. Alternatively, we can avoid reusing sub-tiles by sampling locations c' in (2.6) *without replacement*. However, such sampling strategy will not be iid and as a result, it will cause bias in the Monte Carlo

approximation in (2.6). Fortunately, using a formulation similar to that of Katharopoulos and Fleuret (2019), we can still obtain an unbiased estimator of the average (expectation) in (2.6) from a non-iid sample, without replacement, by leveraging the Gumbel-Top-k trick Kool et al. (2019), which is extended from the Gumbel-Max trick for weighted reservoir sampling Efraimidis and Spirakis (2006). Specifically, from (2.5) we can write

$$\mathbb{E}_{c' \sim b_\Theta(V(x,s_2,c))}[f_\Theta(T_{s_2}(T_{s_1}(x,c),c'))] = \sum_{c' \in C'} \beta^c_{c'} f_\Theta(T_{s_2}(T_{s_1}(x,c),c')), \tag{2.7}$$

from which we can see that the sum on the right is an unbiased estimator of the expectation on the left. Alternatively, we can write

$$\begin{aligned}\mathbb{E}_{c' \sim b_\Theta(V(x,s_2,c))}[f_\Theta(T_{s_2}(T_{s_1}(x,c),c'))] = \\ \sum_{c' \in C'} \sum_{i \neq c'} \beta^c_{c'} \frac{\beta^c_i}{1-\beta^c_{c'}} \left(\beta^c_{c'} f_\Theta(T_{s_2}(T_{s_1}(x,c),c'))\right) \\ + (1-\beta^c_{c'}) f_\Theta(T_{s_2}(T_{s_1}(x,c),i))),\end{aligned} \tag{2.8}$$

where $\beta^c_i/(1-\beta^c_{c'})$ is the attention weight for the i-th sub-tile reweighted to exclude sub-tile c', which is equivalent to having already sampled it. The proof of (2.8) can be found in the Supplementary Material (SM). We can then approximate (2.5) like in (2.6) but sampling without replacement using

$$\begin{aligned}\Psi_\Theta(x) \approx g_\Theta(\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{i-1} \beta^{c_i}_{c'_j} f_\Theta(T_{s_2}(T_{s_1}(x,c_i),c'_j)) \\ + \left[1 - \sum_{j=1}^{i-1} \beta^{c_i}_{c'_j}\right] f_\Theta(T_{s_2}(T_{s_1}(x,c_i),c'_j))),\end{aligned} \tag{2.9}$$

and c'_j is sampled via

$$c'_j \sim p(c'|c'_1,\ldots,c'_{j-1}) \propto \begin{cases} \beta^c_i & \text{if } i \notin \{c'_1,\ldots,c'_{j-1}\} \\ 0 & \text{otherwise} \end{cases},$$

where $p(c'|c'_1,\ldots,c'_{j-1})$ represents sampling location c'_j without replacement, by having already sampled locations $c'_1,\ldots,c'_{j-1}$.

Memory requirements In practice, the memory requirements of the attention sampling model are determined by the model parameters, feature maps, gradient maps and workspace variables Rhu et al. (2016). For neural-network-based image models, memory allocation is mainly dominated by the size of the input image, *i.e.*, H and W. Specifically, the *peak memory* usage at inference for N samples scales with $\mathcal{O}(s^2HW + Nh_2w_2)$ and $\mathcal{O}(s_1^2HW + N's_1^2s_2^2HW + Nh_2w_2)$ for both, the one-stage Katharopoulos and Fleuret (2019) and the proposed two-stage hierarchical model. Here, we use N' to denote the number of unique tiles in Q and s to indicate the scale of the view for the one-stage approach. In fact, we can show that our model requires significantly less GPU memory than one-stage attention sampling by choosing $s_1 < s$ and $s_2 = s$. Note that the number of selected tiles in the first stage decreases dramatically as the attention map is being optimized. We use the term peak memory to refer to the worse case scenario. Empirically, we have observed that the average number of selected tiles is $N' \approx N/2$. A detailed analysis of memory requirements is presented in the SM.

2.2.3 Efficient Contrastive Learning with Attention Sampling

Motivated by Ki et al. (2020), we introduce a contrastive learning objective for the proposed Zoom-In network consisting on encouraging the model to make predictions for cases ($y = 1$), *e.g.*, images with cancer metastases (see Experiments for details), but using sub-tiles with low attention weights while *inverting* the image labels ($y = 1 \rightarrow 0$). Conveniently, we can generate these (negative) contrastive samples without the need for additional modules or model parameters.

Specifically, we leverage the existing attention functions in (2.1) and (2.4). To generate the contrastive feature vectors for image x such that $y = 1$, we first sample (with replacement) tile locations via $1 - a_\Theta(V(x, s_1))$ similar to (2.1). Then, we sample N sub-tiles via $1 - b_\Theta(V(x, s_2, c))$ without replacement similar to (2.4).

The sampled contrastive sub-tiles are passed through the feature network and then processed by the classifier to make predictions $\Psi_\Theta(x|y = 1)$ using (2.9), where the conditioning

$y = 1$ is used to emphasize that we use images x of class $y = 1$ as contrastive examples. In general, the number of contrastive examples (per training batch) is equal to the number of samples such that $y = 1$. For these contrastive sample, we optimize the following objective, $\mathcal{L}_{\text{con}}(\Psi_\Theta(x|y = 1)) = \sum_n -\log(1 - \Psi_\Theta(x_n|y_n = 1))$. Note that $\mathcal{L}_{\text{con}}(\Psi_\Theta(x|y = 1))$ encourages contrastive samples for images x with label $y = 1$ to be predicted as $y = 0$. In multi-class scenarios, this contrastive learning approach can be readily extended by letting one of the classes be the reference, or in general, by using a complete, cross-entropy-based contrastive loss, in which contrastive samples are generated for both classes, *i.e.*, $y = \{0, 1\}$, instead of just one class (half the cross-entropy loss) as in our case.

2.2.4 Memory Cost Analysis

In Memory requirements Section 2.2.2, we empirically analyze the memory requirements of the proposed Zoom-In network and the closely related ATS model. Below we study the memory usage of these models using the colon cancer data.

The memory usage for the one-stage ATS and the proposed two-stage model are $O(s^2HW + Nh_2w_2)$ and $O(s_1^2HW + N's_1^2s_2^2HW + Nh_2w_2)$ respectively. Notations is inherited from the main paper, Zoom-In Network Section 2.2.2. In the colon cancer experiment, $s = 0.2$, $s_1 = 0.1$, $s_2 = 0.2$ and $N = 10$. Then, if we want the memory usage order for the two-stage model to be smaller than that of the one-stage model, we set $N' < 7.5$. In the experiment, $N' < N/2$ when the two-stage model has converged. It should be noted that in the beginning of training, the initial attention map α is approximately uniform because the weights of $a_\Theta(\cdot)$ are initialized at random. The number selected tiles N' is close to N, which implies that the memory consumption of the two-stage model is slightly larger than the one-stage model. However, after the attention network $a_\Theta(\cdot)$ is optimized, the number of selected tiles N' drops dramatically. At which time, the proposed two-stage model consumes much less memory than one-stage model as shown in Figure 2.3. Note that for very large images (gigapixel in size), the number of selected tiles is much smaller than the size of sample space $|C|$, which means that even at the beginning of training, the two-stage model does not need

to instantiate all tiles in an image and thus requires substantially less memory than the one-stage model (see right plot in Figure 2.3).

The above memory usage analysis of input entries can be reflected on the counts of FLOP, since the FLOPs is dominated by the size of the feature maps and model parameters, that is computed in the following way in agreement with Papadopoulos et al. (2021); Dong et al. (2017):

$$FLOPs = C_{in} \times k^2 \times H_{out} \times W_{out} \times C_{out} \tag{2.10}$$

where C_{in} is the number of channels of the input tensor, k^2 is the size of the convolution kernels in this layer, H_{out}, W_{out} and C_{out} are the heights, width and number of channels of the output tensor.

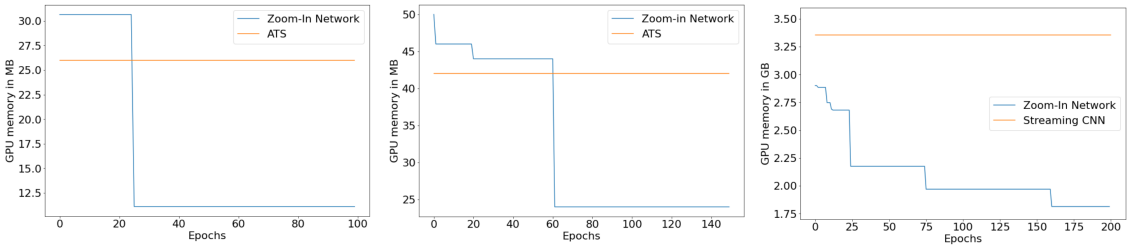


FIGURE 2.3: GPU memory usage (y-axis) versus training epoch (x-axis). We plot the GPU memory usage of the Zoom-In network and the one-stage ATS for the (left) colon cancer dataset, (middle) NeedleCamelyon dataset, and (right) Camelyon16 dataset.

2.3 Related Work

Below we discuss existing research work on classification of very large images with tiny objects, the most relevant body of work being on attention-based models, and general efforts toward computational efficiency for image classification models.

Tiny object classification Recent works have studied CNNs under different noise scenarios, either by performing experiments where label noise is introduced artificially Zhang et al. (2016); Arpit et al. (2017), or by directly working with noisy labels and annotations Mahajan et al. (2018); Han et al. (2018). While it has been shown that large amounts of label noise hinders the generalization ability of CNNs Zhang et al. (2016); Arpit et al. (2017), it has

been further demonstrated that CNNs can mitigate this label-corrupting noise by increasing the size of the data used for training Mahajan et al. (2018), tuning the parameters of the optimization procedure Jastrzębski et al. (2017), or re-weighting input training samples Han et al. (2018). However, all of these works focus on label corruption but do not consider the case of noiseless labels or label assignments with low-noise, in which alternatively, the region of interest (ROI) associated with the label is small or tiny, relative to the size of the image. Purposely, Pawlowski et al. (2019) analyzed the capacity of CNNs in precisely this context, *i.e.*, that of tiny object image classification tasks. Their results indicate that by using a training dataset limited in size, CNNs are unable to generalize well as the ROI-to-image ratio of the input decreases. Typically, the object associated with the label occupies a dominant portion of the image. However, in some real-world applications, such as medical imaging, remote sensing or traffic signs recognition, only a very tiny fraction of the image informs their labels, leading to a low ROI-to-image ratios.

Attention This technique has a long history in the neural networks literature Itti et al. (1998). In the modern era of deep learning, it has been used very successfully in various problems Denil et al. (2012); Fu et al. (2017); Zheng et al. (2019); Wang et al. (2017). Two main classes of attention mechanisms include: *soft attention*, which estimates a (continuous) weight for every location of the entire input Ilse et al. (2018), and *hard attention* which selects a fraction of the data, *e.g.*, a ROI in an image, for processing Papadopoulos et al. (2021), which is a harder problem that resembles object detection, but without ground-truth object boundaries. Note that the attention in Katharopoulos and Fleuret (2019); Ilse et al. (2018); Brendel and Bethge (2019) is defined as the weights of a bag of features from an image. Our formulation can also be interpreted in the same way, since α is the attention on bag of features of tiles and β^c is the attention on bag of features of sub-tiles.

Computational efficiency There are multiple ways to control the computational cost of deep neural networks. We categorize them into four groups: *i*) compression methods that aim to remove redundancy from already trained models Yu et al. (2018); *ii*) lightweight design strategies used to replace network components with computationally lighter counterparts

Jaderberg et al. (2014); *iii*) partial computation methods selectively utilize units of a network, thus creating forward-propagation paths with different computational costs Larsson et al. (2017); and *iv*) reinforcement learning and attention mechanisms that can be used to selectively process subsets of the input, based on their importance for the task of interest Ramapuram et al. (2018); Levi and Ullman (2018); Uzkent and Ermon (2020); Katharopoulos and Fleuret (2019); Cordonnier et al. (2021). The latter being the strategy we consider in the proposed Zoom-In architecture.

In-sample contrastive learning Conventional deep neural networks lack robustness to out-of-distribution data or naturally-occurring corruption such as image noise, blur, compression and label corruption. Contrastive learning Ki et al. (2020) has demonstrated great success with learning in noisy scenarios, *e.g.*, corrupted ImageNet Khosla et al. (2020). Here, we aim to leverage contrastive learning to mitigate the performance loss caused by images with low ROI-to-image ratio. Hence, the built-in attention mechanism facilitates the in-sample contrastive learning because contrastive samples can be obtained using the same attention mechanisms without the need for additional model components or parameters.

Weakly supervised training on gigapixel images Recent work has demonstrated that deep learning algorithms have the ability to predict patient-level attributes, *e.g.*, cancer staging from whole slide images (WSIs) in digital pathology applications Lee and Paeng (2018). Because these images are so large and no prior knowledge of which subsets of the image (tiles) are associated with the label, such task is known as weakly supervised learning Campanella et al. (2019); Lu et al. (2021). Specifically, the model has to estimate which regions within the image are relevant to the label, so predictions can be made using information from these regions alone; not the whole image. Importantly, with current hardware architectures, WSIs are too large to fit in GPU memory, so one commonly used technique is to build a model to select a subset of patches from the image Zhu et al. (2016); Naik et al. (2020). An alternative approach consists in using the entire WSI but in a compressed, much smaller, representation, at the cost of losing fine-grained details that may be important Tellez et al. (2019). Building representations that aggregate features from

selected image regions (tiles or patches) are also alternative approaches Campanella et al. (2019); Courtiol et al. (2018). We consider the performance of these approaches relative to the proposed Zoom-In network in the experiments.

2.4 Experiments

We evaluate the proposed approach in terms of accuracy and GPU memory requirements. In the results below, Zoom-In Network refers to the proposed method with a lightweight LeNet backbone, and Zoom-In Network (Res) refers to our method using a ResNet16 backbone. The details of the model architecture are presented in the SM. Moreover, we highlight the ability of the model to attend to a small amount of full-resolution sub-tiles (ROIs) of the image inputs, which results in a significantly reduced peak GPU memory usage and superior test accuracy relative to the competing approaches. We consider methods that can handle large images as inputs, *e.g.*, attention sampling models (ATS) Katharopoulos and Fleuret (2019), Differentiable Patch Selection (Top-K) Cordonnier et al. (2021), BagNet Pawlowski et al. (2019), EfficientNet Tan and Le (2019) and streaming CNNs Pinckaers et al. (2019). Simultaneously, we also compare our model with methods that apply strategies similar to zoom-in *e.g.*, PatchDrop Uzkent and Ermon (2020) and RA-CNN Fu et al. (2017). For the peak memory usage, we report inference memory in Mb per sample, *i.e.*, for a batch of size 1. We also report the floating point operations (FLOPs) and run time when inferring a single image. Details of the model architecture and some hyper-parameters not specified in each experiment, as well as an ablation study examining the impact on performance of N, λ, and using contrastive learning are presented in the SM. In-sample contrastive learning is applied after training without it for 10 epochs, and the entropy regularization parameter (for our model and ATS) is set to $\lambda = 1e^{-5}$.

Datasets We focus on datasets that have relatively large image sizes and feature tiny ROI objects that are scattered in large backgrounds, unlike natural images, in which objects are (usually) in the middle of the image due to the fact that photographers tend to center

Table 2.1: Test set results for colon cancer, NeedleCamelyon and fMoW data. Memory denotes average peak memory per sample usage at inference.

Colon Cancer

Method	Accuracy (%)	FLOPs (B)	Memory (Mb)	Time (ms)
PatchDrop Uzkent and Ermon (2020)	81.0	75.29	520.44	12.33
RA-CNN Fu et al. (2017)	86.4	135.88	4432.46	38.74
CNN Katharopoulos and Fleuret (2019)	90.8 $\pm$ 1.2	1.83	235.68	7.62
ATS Katharopoulos and Fleuret (2019)	90.7 $\pm$ 1.4	0.24	15.83	**2.81**
Zoom-In Network (ours)	**95.0 $\pm$ 2.6**	**0.24**	**2.55**	3.20

NeedleCamelyon

Method	Accuracy (%)	FLOPs (B)	Memory (Mb)	Time (ms)
BagNet Brendel and Bethge (2019)	70.0	222.72	3914.81	12.90
ATS Katharopoulos and Fleuret (2019)	72.5	1.66	37.97	**9.27**
Zoom-In Network (ours)	76.0	**0.52**	**11.78**	10.20
Zoom-In Network (Res) (ours)	**78.1**	0.84	14.22	11.42

Traffic Signs Recognition

Method	Accuracy (%)	FLOPs (B)	Memory (Mb)	Time (ms)
EfficientNet-B0 s0.5 Tan and Le (2019)	65.9	4.82	673.39	27.06
EfficientNet-B0 Tan and Le (2019)	79.1	19.26	2229.59	34.88
ATS-10 Katharopoulos and Fleuret (2019)	90.5	1.43	54.51	10.3
TopK-10 Cordonnier et al. (2021)	91.7	1.61	125.16	**9.8**
Zoom-In Network (ours)	91.2	**0.79**	**12.65**	12.28
Zoom-In Network (Res) (ours)	**92.6**	1.18	15.83	13.16

Functional Map of the World

Method	Accuracy (%)	FLOPs (B)	Memory (Mb)	Time (ms)
EfficientNet-B0 Tan and Le (2019)	70.2	8.22	1404.09	22.72
ATS-30 Katharopoulos and Fleuret (2019)	71.1	2.52	53.42	10.73
TopK-30 Cordonnier et al. (2021)	71.6	2.30	73.49	**10.12**
Zoom-In Network (ours)	72.9	**1.85**	**10.81**	**11.47**
Zoom-In Network (Res) (ours)	**74.3**	2.24	13.53	12.25

images around the object of interest (target) Touvron et al. (2019). Consequently, in the experiments, we do not consider datasets such as ImageNet, iNaturalist and COCO because in these the ROI-to-image ratio is close to 1 and also because image sizes are relatively small relative to some of the other datasets considered and described below.

We present experiments on five datasets: *i*) the colon cancer dataset introduced in Sirinukunwattana et al. (2016) aims to detect whether epithelial cells exist in a Hematoxylin and Eosin (H&E) stained images. This dataset contains 100 images of dimensions 500 $\times$ 500.

The images originate both from malignant and normal tissue, and contain approximately 22,000 annotated cells. Following Ilse et al. (2018); Katharopoulos and Fleuret (2019), we treat the problem as a binary classification task where the positive images are those containing at least one cell belonging to the epithelial cell class. *ii*) The NeedleCamelyon dataset Pawlowski et al. (2019) is built from cropped images from the original Camelyon16 dataset with specified ROI-to-image ratios. Specifically, we generate datasets for ROI-to-image ratios in the range of $[0.1, 1]\%$, and we crop each image with size of $1,024 \times 1,024$ pixels. Positive examples are created by taking 50 random crops from every annotated cancer metastasis area if the ROI-to-image ratio falls within the range $[0.1, 1]\%$. Negative examples are taken by randomly cropping normal whole-slide images and filtering out image crops that mostly contain background. Further, we ensure the class balance by sampling an equal amount of positive and negative crops. *iii*) Traffic Signs Recognition dataset Larsson and Felsberg (2011) consists of over $20,000$ road scene images with a size of 960×1280. Here, we use the same subset as Katharopoulos and Fleuret (2019); Cordonnier et al. (2021). The task is to classify whether the road-scene image contains a speed limit sign (50, 70 or 80 km/h) or not. The subset used in Katharopoulos and Fleuret (2019); Cordonnier et al. (2021) and our experiments includes 747 images for training and 684 images for testing. *iv*) The Functional Map of the World (fMoW) dataset Christie et al. (2018) aims to classify the functional purpose of buildings and infrastructures and land-use from high-resolution satellite images. The approximate range of image size in this dataset is 500×500 to $9,000 \times 9,000$ pixels. In our experiment, we extract a class-balanced subset from the original fMoW dataset to further illustrate the proposed method is applicable beyond digital pathology images. Our constructed subset consists of $15,000$ training images and $9,571$ test image from 10 classes. More details of constructing the subset is provided in SM. *v*) Finally, we utilize the Camelyon16 dataset to further demonstrate the utility of our model on gigapixel images. This dataset contains 400 WSIs, 270 WSIs with pixel-level annotations, and 130 unlabeled WSIs as test set. We split the 270 slides into train and validation sets; for hyperparameter tuning. Typically, only a small portion of a slide contains biological tissue of interest, with

background and fat encompassing the remaining areas, *e.g.*, see Figure 2.1 for a typical WSI (with pixel-level annotations).

Colon cancer We use the same experimental setup as Katharopoulos and Fleuret (2019); Ilse et al. (2018), namely, 10-fold-cross-validation, and five repetitions per experiment.

The approach most closely related to ours is the one-stage attention sampling model in Katharopoulos and Fleuret (2019). We refer to this method as ATS-N, where $N = 10$ indicates the number of tiles drawn from the attention weights, and we set $s = s_2$ in all experiments. The value of N was selected to maximize performance. To showcase the advantages of our model compared to traditional CNN approaches, we also include a ResNet He et al. (2016a) with 8 convolutional layers and 32 channels as naive baseline. For our model, we set $s_1 = 0.1$, $s_2 = 0.2$, $N = 10$, and $h_2 = w_2 = 27$.

Results are summarized in Table 2.1. The proposed two-stage attention sampling model results in approximately 4.3% higher test accuracy than (one-stage) ATS-10; presumably due to its ability to better focus on informative regions (sub-tiles) of the image via the hierarchical attention mechanism. Moreover, the baseline (CNN) and ATS-10 require at least $\times 90$ and $\times 6$ more memory relative to the Zoom-In network, respectively. Alternatively, PatchDrop Uzkent and Ermon (2020) and RA-CNN Fu et al. (2017) not only underperform the base CNN but also have higher memory requirements. The memory efficiency of the proposed approach is justified by the way in which the image is processed as a small collection of sub-tiles, thus resulting in a substantially reduced forward pass cost relative to the CNN and ATS-10 models. The FLOPs of ATS and Zoom-In Network are comparable because in the colon cancer experiments, the feature extractor $f_\Theta(\cdot)$ dominates the FLOP count. Since ATS and Zoom-In Network feed the same number of patches into $f_\Theta(\cdot)$, their resulting FLOPs in this step is the same. In fact, this step contributes 0.235 B FLOPs (96%). Further, our model takes slightly more run time due to the implementation inefficiency when extracting tiles and sub-tiles.

NeedleCamelyon Note that the NeedleCamelyon dataset uses larger images ($1,024 \times 1,024$) compared to the ones (up to 512×512) used in Pawlowski et al. (2019) to better

showcase the abilities of the models considered. Following Pawlowski et al. (2019), we split the NeedleCamelyon dataset into training set, validation set, test set with a ratio of 60:20:20. The number of images for each set are $6,000$, $2,000$ and $2,000$, respectively. Positive and negative samples are balanced in each set.

We compare our model with ATS-30 and an existing CNN architecture used for a similar NeedleCamelyon dataset in Pawlowski et al. (2019). BagNet Brendel and Bethge (2019) is a CNN model extracts features at tile-level, which is efficiently used in NeedleCamelyon experiments in Pawlowski et al. (2019). We use the BagNet as the CNN baseline in our experiment. For our model, we set $s_1 = 0.25$, $s_2 = 0.5$, $N = 30$, and $h_2 = w_2 = 32$. Due to the much smaller ROI-to-image ratio of NeedleCamelyon relative to the colon cancer dataset, we set a larger s_1 and s_2 to ensure the down-sampling will not wash out the discriminative information. We also increase N since the number of attention weights with large values is larger in this case.

Table 2.1 shows performance for the proposed model and the baselines. We observe that for larger images, the Zoom-In network has better GPU memory use at inference time because much less sub-tiles at scale s_2 tend to be instantiated. In terms of test accuracy, the proposed model results in approximately 3.5% higher test accuracy than one stage ATS-30 and 6.0% higher than the BagNet baseline in this experiment. Moreover, BagNet and one-stage attention sampling require at least $\times 500$ and $\times 7$ more memory, respectively, compared to the proposed approach. Here, we can see our Zoom-In Network consumes drastically less FLOPs than ATS. This is because the Zoom-In Network requires less pixels of the original input images to select a comparable amount of high resolution ROI patches for prediction relative to ATS.

Traffic Sign Recognition We compare the Zoom-In Network with a traditional CNN (EfficientNet-B0 Tan and Le (2019)) and the recently published one-stage zoom-in methods (ATS Katharopoulos and Fleuret (2019), TopK Cordonnier et al. (2021)). For EfficientNet-B0, we use both the original resolution images and images downsampled by half (denoted as s0.5) as inputs, to show the limitations of traditional CNNs. For ATS and TopK, we

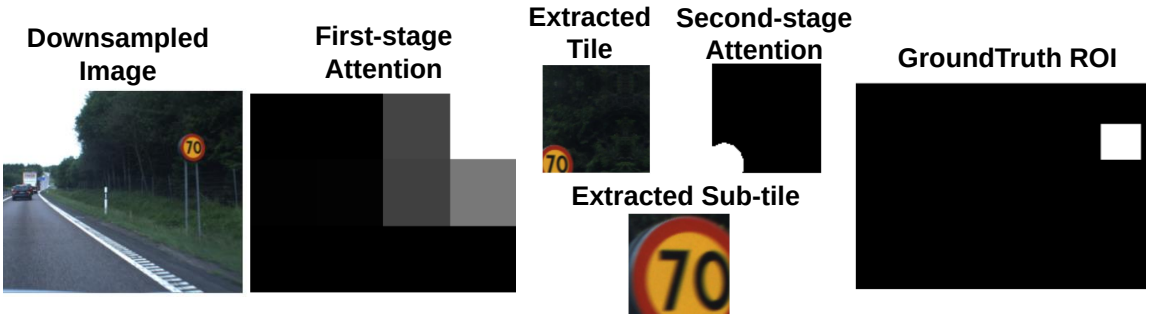


FIGURE 2.4: Illustration of the intermediate results for the Traffic Sign dataset using the Zoom-In Network. We show the tile and sub-tile with the highest first-stage and second-stage attention. More intermediate results are included in the SM.

use the same hyperparameter settings from Katharopoulos and Fleuret (2019); Cordonnier et al. (2021). For our Zoom-In Network, we tried two types of the backbones(LeNet and ResNet16) and we set $s_1 = 0.125$, $s_2 = 0.3$, $N = 10$, and $h_2 = w_2 = 100$. The details of the network architecture are shown in the SM.

In Table 2.1, we can see that our Zoom-In model achieves the highest accuracy and lowest GPU memory consumption among all tested methods. Also, our model requires less FLOPs by using less number of input pixels than ATS and Top-K. The merits of our model originate from accurate target object localization achieved by the attention mechanism and contrastive learning, as well as efficient GPU memory usage. Examples of attention maps for Traffic Sign dataset similar to those in Figure 2.4 are shown in the SM.

Functional Map of the World Functional Map of the World (fMoW) consists of a

Table 2.2: Results on Camelyon16 data. Memory denotes average peak memory per sample at inference.

Method	Pixel-level Annotation	Accuracy (%)	Memory (Mb)
Streaming CNN Pinckaers et al. (2019)	No	70.6	3,256.29
CLAM Lu et al. (2021)	No	78.0	206.88
MIL Campanella et al. (2019)	No	79.9	140.68
MRMIL Li et al. (2021)	No	81.1	568.00
Zoom-In Network	No	**81.3**	**71.76**
Zoom-In Network (Res)	No	**82.6**	**71.76**
Zoom-In Network	Yes	88.2	**71.76**
Zoom-In Network(Res)	Yes	90.8	**71.76**
Winning model Bejnordi et al. (2017)	Yes	92.2	395.77

large amount of high-resolution RGB images of various sizes ranging from 500×500 to $9,000 \times 9,000$. Following Papadopoulos et al. (2021), we choose EfficientNet-B0 Tan and Le (2019) as the baseline model. EfficientNet-B0 effectively scales with large images and has been proved to serve as a good baseline model on the fMoW dataset in Papadopoulos et al. (2021). We also implement the ATS and TopK with our best efforts for fMoW dataset. We set $s_1 = 0.25$, $s_2 = 0.5$, $N = 30$, and $h_2 = w_2 = 50$ for our model.

Results in Table 2.1 show that Zoom-In Network surpasses EfficientNet-B0 in accuracy and memory consumption at inference time. Examples of attention maps for fMoW similar to those in Figure 2.4 are shown in the SM.

Camelyon16 This is a gigapixel dataset consisting of whole-slide images (WSIs) with sizes ranging from $45,056 \times 35,840$ to $217,088 \times 111,104$. The objective here is to predict whether a WSI contains cancer metastases. As we described in the Related Work Section, existing works have attempted to train CNNs on very large images with only image-level labels, *i.e.*, via weakly supervised training on gigapixel images. We consider the streaming CNN Pinckaers et al. (2019), CLAM Lu et al. (2021), MIL Campanella et al. (2019) and MRMIL Li et al. (2021), which we previously briefly described. Details of these baselines are presented in the SM. Further, we also evaluate the model for the scenario in which pixel-level annotations (ROIs) are available. Here, we compare our results to the winning model of the Camelyon16 challenge Bejnordi et al. (2017). For our model, we set $s_1 = 0.03125$, $s_2 = 0.125$, $N = 100$, and $h_2 = w_2 = 50$.

In Table 2.2, we see that our Zoom-In network achieves the highest test accuracy when pixel-level annotations are not available. Even when there are pixel-level annotations, our model yields a test accuracy close to the winning model of the Camelyon16 challenge, without the need for comprehensive tuning and handcrafted features (*i.e.*, the minimum surrounding convex region, the average prediction values, and the longest axis of the lesion area). For the memory comparison, all methods require substantially more memory than the proposed approach, notable $\times 8$ more than MRMIL which has comparable performance. Moreover, we also consider the situation in which the pixel-level annotations are available. The extension

to leverage pixel-level annotations is described in the SM. The performance of the proposed model is relatively close to the winning model Bejnordi et al. (2017), which indicates that the proposed approach is flexible and also accurate when we have the manually annotated ROIs. Additional details including attention maps similar to that in Figure 2.4 are provided in the SM.

We also analyzed the correlation of the attention weights generated by the Zoom-In network with the ground-truth metastases-to-tile ratio obtained from pixel annotations. Specifically, the proportion of each sub-tile of size $h_2 \times w_2$ that is covered by cancer metastases. The Spearman correlation coefficient for all the tiles with pixel-level annotations is $\rho = 0.3570$, indicating a good agreement. Note that these attention weights are obtained from the model trained without pixel-level annotation information. In the SM, we visually present these correlations as scatter plots (attention weights *vs.* metastases-to-tile ratios).

From all the results above, we see Zoom-In Network reduces the GPU memory usage significantly. The reasons why we pursue a low GPU memory consumption are: *i*) GPUs with high memory are expensive and not widely available for applications in practice. A model using less GPU at inference time can be deployed with less expense; *ii*) the highly memory-efficient model allows training and inferring images larger than gigapixels possible; *iii*) With the growing use of neural network on mobile/edge devices, it is key to develop memory-light GPU models to allow locally running deep learning service on mobile/edge devices.

2.4.1 Time/Memory-Accuracy Trade-off

In our model, the main hyper-parameter that varies time and memory consumption is the sample size (N). Although the time/memory - accuracy trade-off can be inferred from the ablation study of sample size (N) in Table 2.4, we show additional details concerning the time-memory trade-off in the following table:

Table 2.3: Time/Memory-Accuracy Trade-off of Zoom-In Network compared with ATS in the colon cancer dataset experiment.

Method	Sample Sze (N)	Accuracy (%)	Memory (Mb)	Time (ms)
Zoom-In Network	5	93.2	2.43	3.04
	10	78.0	2.55	3.20
	50	79.9	5.03	4.33
ATS	10	90.7	15.83	2.81
	50	90.7	25.77	4.03

2.4.2 Ablation Study

In Table 2.4, we present and ablation study to evaluate the effects of the entropy regularization λ, sample size N and contrastive learning in our Zoom-In network. We examine these hyper-parameters on the colon cancer dataset to show the effects of varying N and λ, as well as to demonstrate the usefulness of the contrastive learning objective. Then, we further justify the contribution of contrastive learning on NeedleCamelyon and Camelyon16 datasets a similar ablation strategy.

Table 2.4: Ablation study results of λ, N and using contrastive learning. The first, seventh, and ninth rows are the standard hyper-parameter settings used in our experiments and the others are selected to show performance variations for different settings.

Dataset	Entropy Regularization (λ)	Sample Size (N)	Contrastive Learning	Test Accuracy (%)
Colon Cancer	1e-5	10	Yes	95.0
	0	10	Yes	94.0
	1.0	10	Yes	95.0
	1e-5	10	No	94.0
	1e-5	5	Yes	93.2
	1e-5	50	Yes	96.0
NeedleCamelyon	1e-5	30	Yes	76.0
	1e-5	30	No	74.3
Camelyon16	1e-5	100	Yes	81.3
	1e-5	100	No	80.6

Table 2.5: The architecture of the Zoom-In Network using a LeNet Structure.

$a_\Theta(\cdot)$

Layer	Type
1	Conv(3, 1, 1, 8) + Tanh()
2	Conv(3, 1, 1, 8) + Tanh()
3	Conv(3, 1, 1, 1) + Tanh()
4	GlobalAveragePooling2D()
5	SoftMax()

$b_\Theta(\cdot)$

Layer	Type
1	Conv(3, 1, 1, 8) + Tanh()
2	Conv(3, 1, 1, 8) + Tanh()
3	Conv(3, 1, 1, 1) + SoftMax()

$f_\Theta(\cdot)$

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

$g_\Theta(\cdot)$

Layer	Type
1	fc-n_{class}

2.4.3 Model Components

The components of the proposed Zoom-In network are summarized in Figure 2.2. We consider the LeNet and ResNet16 architecutres for the feature function $f_\Theta(\cdot)$. The choice of LeNet is consistent to the one used in Ilse et al. (2018); Sirinukunwattana et al. (2016). The mapping $f_\Theta(\cdot)$ is implemented by a LeNet5-like architecture; consistent to the one used in Ilse et al. (2018); Sirinukunwattana et al. (2016). We also considered a ResNet architecture to show our zoom-in strategies is also compatible with modern network architectures. The details of each subnetwork is listed in Table 2.5 and Table 2.6. In the table, the convolutional layer is denoted as "Conv" and the kernel size, stride, padding and number of filters are provided in the following brackets. "fc" means the fully-connected layer and the output hidden units is provided after the dash. n_{class} is the number of classes in the task that

Table 2.6: The architecture of the Zoom-In Network using a ResNet16 Structure.

$a_\Theta(\cdot)$

Layer	Type
1	Conv(3, 1, 1, 8) + ReLU()
2	Conv(3, 1, 1, 16) + ReLU()
3	Conv(3, 1, 1, 32) + ReLU()
4	Conv(3, 1, 1, 1) + ReLU()
5	GlobalAveragePooling2D()
6	SoftMax()

$b_\Theta(\cdot)$

Layer	Type
1	Conv(3, 1, 1, 8) + ReLU()
2	Conv(3, 1, 1, 16) + ReLU()
3	Conv(3, 1, 1, 32) + ReLU()
4	Conv(3, 1, 1, 1) + SoftMax()
5	SoftMax()

$f_\Theta(\cdot)$

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()

$g_\Theta(\cdot)$

Layer	Type
1	fc-n_{class}

the model is solving. "Tanh", "ReLU" and "SoftMax" represent the non-linear functions. "GlobalAveragePooling2D" is the global average pooling operation in the spatial dimension of the tensors, functioning the same as `https://www.tensorflow.org/api_docs/python/tf/keras/layers/GlobalAveragePooling2D`. "ResBlock" is the standard ResNet block He et al. (2016a). In the brackets, we provide the kernel size, stride, and number of filters.

The attention function $a_\theta(\cdot)$ in (2.1) is a smaller neural network consisting of a three-layer convolutional network with 8 kernels and ReLu activations, followed by average pooling and a softmax activation to obtain the matrix of attention weights. The attention function $b_\theta(\cdot)$

in (2.4) is defined similarly.

Finally, the classifier $g_\Theta(\cdot)$ is specified as a single fully connected layer with sigmoid activation. The complete objective for the Zoom-In network is

$$\mathcal{L}(y, x; \Theta) = \mathcal{L}_{\text{ce}}(y, x) + \mathcal{L}_{\text{con}}(x, y = 1) + \mathcal{L}_{\text{er}}(\alpha) + \textstyle\sum_{c \in Q} \mathcal{L}_{\text{er}}(\beta^c), \tag{2.11}$$

where $\mathcal{L}_{\text{ce}}(y, x)$ is the cross-entropy loss for the image-level binary classification, $\mathcal{L}_{\text{con}}(x, y = 1)$ is the contrastive loss introduced above, and $\mathcal{L}_{\text{er}}(\cdot)$ is the entropy regularization loss for attention matrices α and $\{\beta^c\}_c$. The regularization term Katharopoulos and Fleuret (2019); Mnih et al. (2016), $\mathcal{L}_{\text{er}}(p) = -\lambda H(p) = \lambda \sum_i p_i \log(p_i)$, where $H(\cdot)$ is the entropy of a discrete distribution and λ is the trade-off coefficient, is included in the overall objective to prevent overly sparse attention matrices that may result from overfitting or early converging during training.

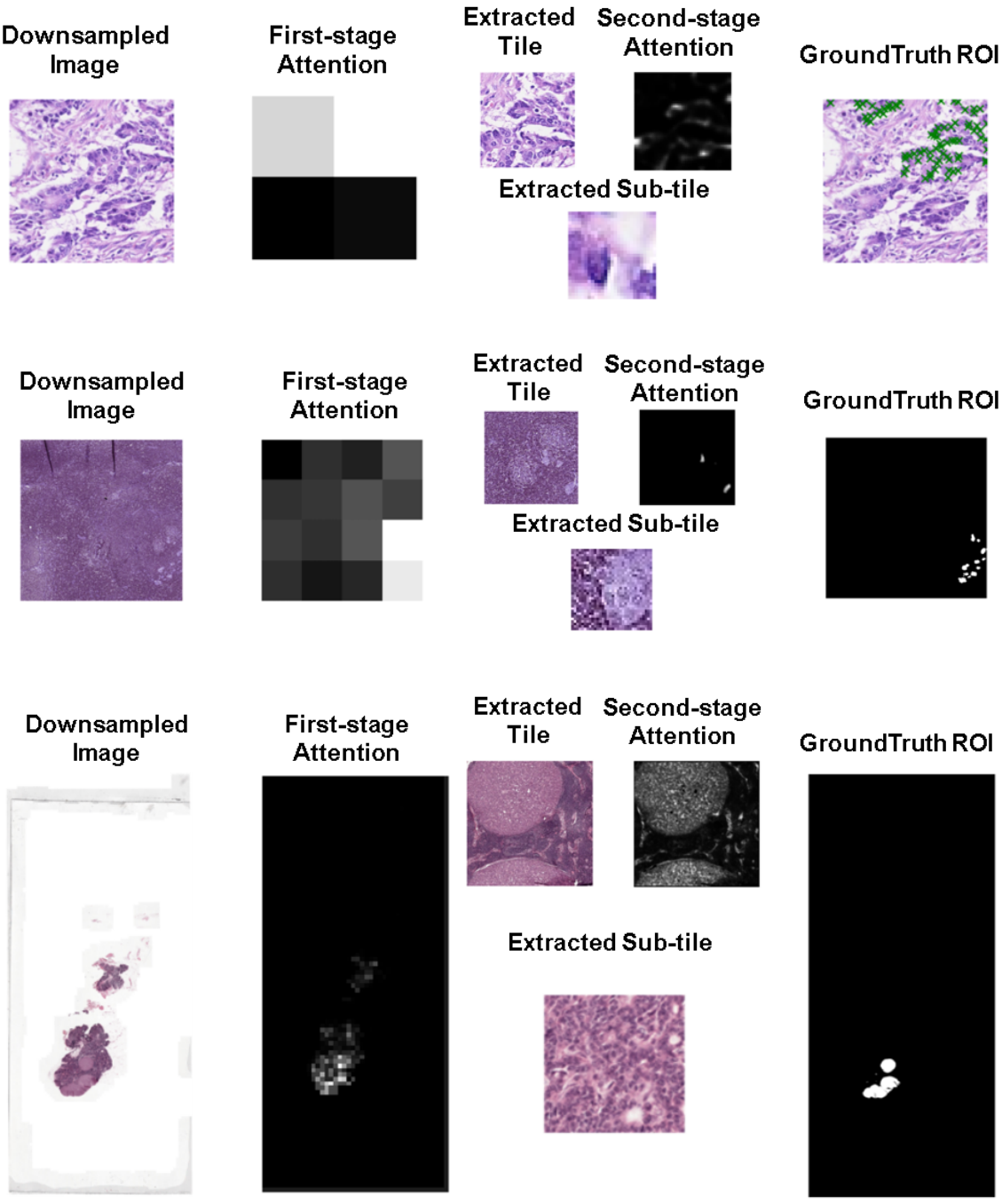


FIGURE 2.5: Intermediate results of the proposed Zoom-In network. For each row, we exhibit the visualization of *a*) colon cancer, *b*) NeedleCamelyon, *c*) Traffic Sign Recognition, *d*) fMoW and *e*) Camelyon16 datatsets. In each panel we show the downsampled original image, the ground truth ROI mask, the attention masks, the extracted tiles and sub-tiles with the highest first-stage and second-stage attention respectively.

2.5 Discussion

We presented the Zoom-In network that can efficiently classify very large images with tiny objects. We improved over the existing CNN-based baselines both in terms of accuracy and peak GPU memory use, by leveraging a two-stage hierarchical sampling strategy and a contrastive learning objective. We also considered the scenario in which pixel-level annotations (segmentation maps) are available during training. In the experiments, we demonstrated the advantage of the proposed model on five challenging classification tasks. We note that the images in the Camelyon16 dataset are all gigapixel in size and that we are likely the first ones training an end-to-end deep learning model for them. Our model achieved the best accuracy when pixel-level annotations are not available, while also using a small amount of GPU memory, which allows for training and inference on full-resolution gigapixel images using a single GPU. One limitation of the proposed model is the need to specify the number of sub-tile samples N, which can be potentially estimated from data.

3. Neural Insights for Digital Marketing Content Design

3.1 Introduction

Content experimentation plays an important role in driving key performance indicators as part of present-day online marketing Kohavi and Longbotham (2017); Fiez et al. (2022). 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 Nabi et al. (2022); Li et al. (2022b). Consequently, it is common that resulting insights suffer from cognitive and incentive bias by marketing teams who analyze the results Bissuel (2020).

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 Sinha et al. (2020), 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

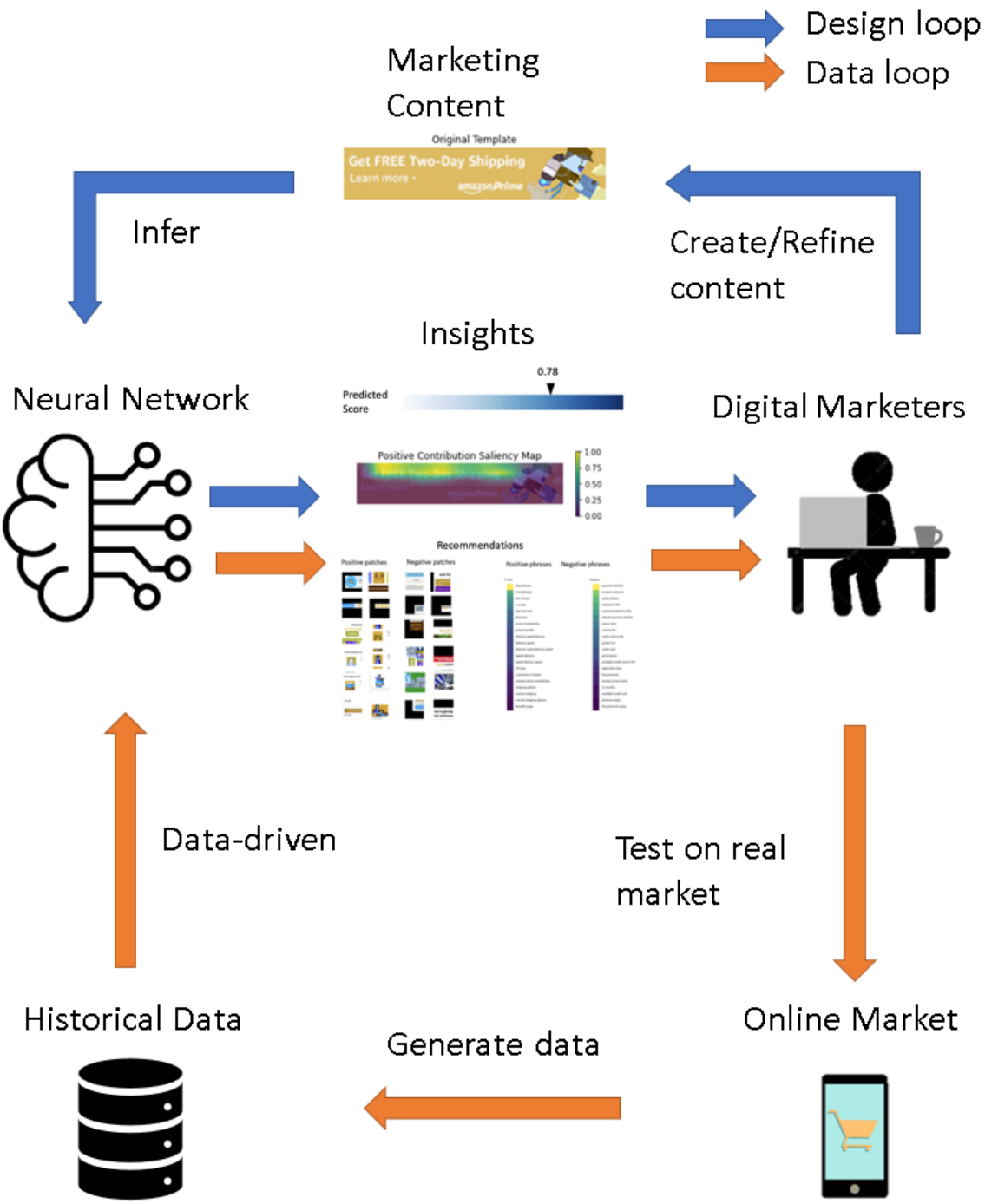


FIGURE 3.1: Diagram of AI-driven marketing content design.

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 3.1). This approach is motivated by the remarkable success of deep neural nets in diverse application areas Kong and Henao (2022); Grigorescu et al. (2020); Singh et al. (2017); Wang et al. (2022b); Geng et al. (2020, 2023); Cai et al. (2017); Grislain et al. (2019); Nuruzzaman and Hussain (2018); Wang et al. (2021c); Biswas et al. (2019). 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 3.6.

Organization. In Section 3.2, we introduce the workflow that describes how digital marketers conduct experiments. In Section 3.3, we present the neural network model used to model the content data and the process used to train the neural network. In Section 3.4, we explain the method proposed to generate insights for content based on our multimodal neural network. In Section 3.5, 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 3.6, we discuss our experiments and their results.

3.2 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 Kohavi and Longbotham (2017); Fiez et al. (2022). Randomized and properly designed experiments can be used to establish causality, that is, to identify elements in marketing content likely to provide incremental impact Sawant et al. (2018). 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:

$$Y = N_{\text{clicks}}/N_{\text{total}}, \tag{3.1}$$

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.

3.3 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 3.2). The image encoder is an RGB ResNet-18 He et al. (2016a) model without the fully-connected classifier. The text encoder is a standard BERT model Devlin et al. (2019) without the classification head. Fully-connected MLP neural networks Rumelhart et al. (1985) serve as the encoders for both domain and categorical features. We then use the most basic fusion strategy Gadzicki et al. (2020) by concatenating the embeddings from all modalities via their encoders (see Equation 3.3). 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

$$\begin{aligned} I_{\text{emb}} &= \text{ResNet}(I), \quad T_{\text{emb}} = \text{BERT}(T), \\ D_{\text{emb}} &= \text{MLP}_1(D), \quad F_{\text{emb}} = \text{MLP}_2(F). \end{aligned} \tag{3.2}$$

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

$$\widehat{y} = C(X_{\text{emb}}) = \text{MLP}_3(\{I_{\text{emb}}, T_{\text{emb}}, D_{\text{emb}}, F_{\text{emb}}\}). \tag{3.3}$$

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 3.6.

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 $\sigma(\widehat{y})$:

$$L = \text{MSE}(Y, \sigma(\widehat{y})). \tag{3.4}$$

3.4 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 Fukui et al. (2019); Wang et al. (2019) or to train separate modules to explicitly produce model explanations Chang et al. (2019); Goyal et al. (2019). 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.

3.4.1 Insights: guidance to improve current design

We start by addressing the attribution problem Sundararajan et al. (2017); Baehrens et al. (2010), defined as the assignment of contributions to individual input features Efron (2020). 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 3.6.3 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 3.6.3).

Consider replacing a feature x in bag X with another feature $\bar{x}'$ such that $c(\bar{x}') \geqslant c(\bar{x})$, which is consistent with an A/B testing in which a treatment is derived from a control Biloš et al. (2016); Fiez et al. (2022). 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 $c(\bar{x}') \geqslant c(\bar{x})$ will increase the overall success rate from Y to Y' when $C(X') \geqslant C(X) \Leftrightarrow Y' \geqslant Y$, and under Assumption 1.*

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

$$C(X') = \sum_{x \in X} c(x) + (c(\bar{x}') - c(\bar{x})) \geqslant C(X). \tag{3.5}$$

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

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 $C(X') \geqslant C(X) \Rightarrow Y' \geqslant Y$ may not always hold. However, we use pairwise accuracy to evaluate the accuracy of our predictor when comparing two content elements in Section 3.6.3 and validate the efficacy of the replacement. Below, we detail both the prediction function $C(\cdot)$ and attributor $c(\cdot)$.

3.4.2 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 Sun-

dararajan et al. (2017). Activation-based methods use a variety of ways to weigh activation maps of intermediate layers in neural network to assign attributions Selvaraju et al. (2017). 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 Merrick (2019) 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 Lundberg and Lee (2017); Ancona et al. (2019). Kernel SHAP leverages a kernel-weighted linear regression to estimate the Shapley values of each input as the attribution scores Lundberg and Lee (2017). Langlois et al. (2021) 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 3.3, 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.

3.4.3 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 3.2). 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:

$$r(x) = \frac{1}{N} \sum_{X \in \mathcal{X}} c(x), \tag{3.6}$$

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

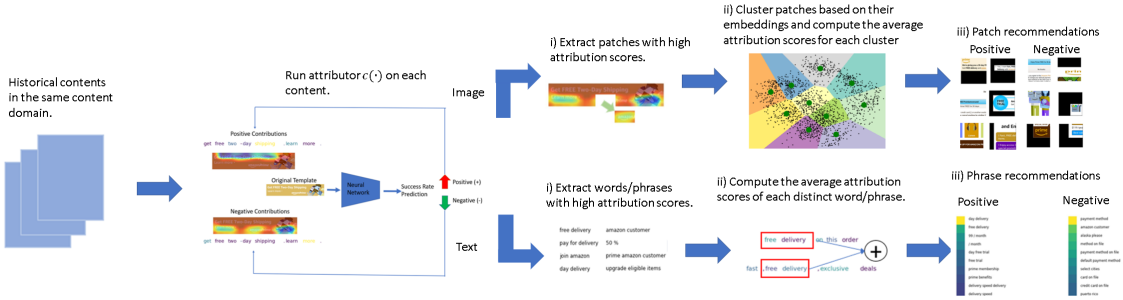


FIGURE 3.2: Generating image and text recommendations. For all content instances in the same marketing location $\boldsymbol{D}$, we first run attribution methods on ResNet-18 and BERT separately to get the attribution maps for text and images. For text data, the scores are simply ranked by the average 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 3.4 for details.

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 3.2.

figureExample 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 Handler et al. (2016) 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 3.4.3 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 He et al. (2016b); Sulthana et al. (2020); Niu et al. (2018); Biswas et al. (2019), 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*) Sub-

FIGURE 3.3: 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.

sequently, we cluster these patches based on their ResNet-18 embeddings using K-Means clustering Ball and Hall (1965) 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 Ranjan et al. (2017); Anas et al. (2017). We use the elbow method to select the number K of centroids Thorndike (1953). 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 3.3 shows an illustrative example of our visual design recommendation. The recommendations of images have some insights similar to text insights in Figure 3.4.3. 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 Seymour (2016), 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).

3.5 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 Agarwal et al. (2022), which do not satisfy our needs. Runge et al. (2019) 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.

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:

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

$d_Y := |\Delta Y|$;

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

Output ρ.

In Section 3.4, 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 Zhang (2021), 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.

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 $\Delta Y = Y' - Y$. 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 Benesty et al.

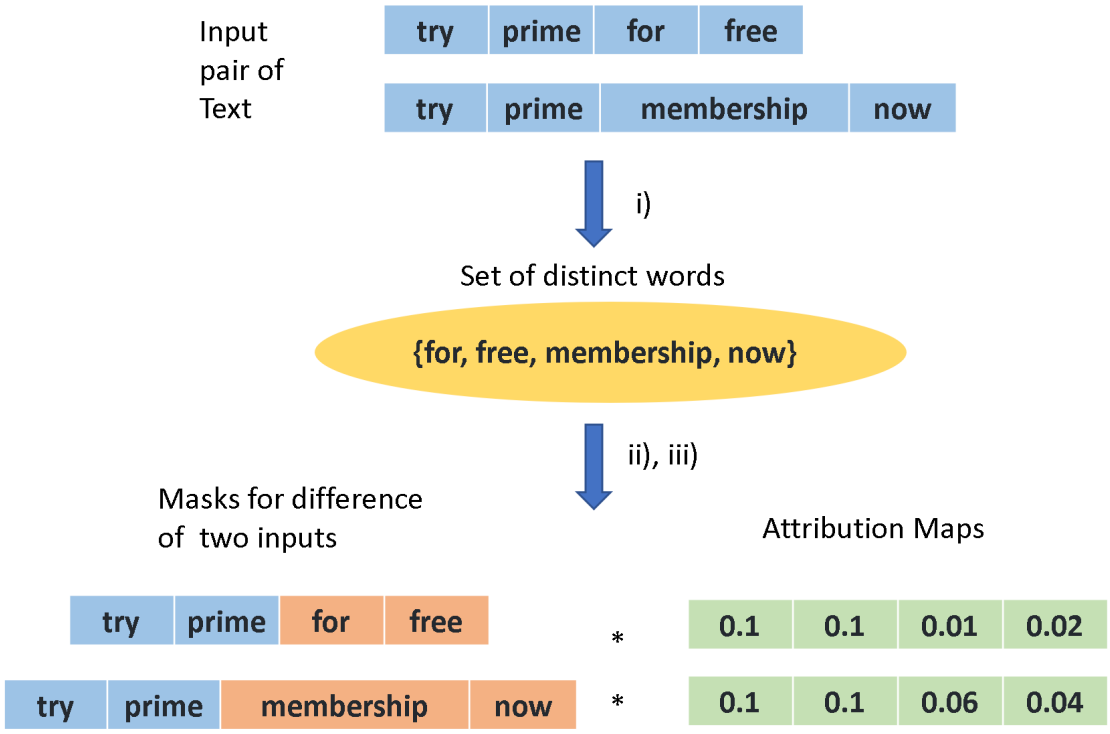


FIGURE 3.4: Visual explanation of text evaluation Algorithm 2.

Algorithm 2: 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 $k \leftarrow 0$, $d_C \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$ and $d_y \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$.

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.

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

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

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:

$d_C[k] \leftarrow d_C, \quad d_y[k] \leftarrow d_y$;

$k \leftarrow k + 1$;

end ;

2) Compute Pearson Correlation ρ between d_C and d_Y .

Output ρ.

(2009). The Pearson Correlation Coefficient used in our evaluation is defined as:

$$\rho = \frac{\text{cov}(\Delta c(x), \Delta Y)}{\sigma_{\Delta c(x)} \sigma_{\Delta Y}}, \tag{3.7}$$

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 d_C and 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 Algorithms 2

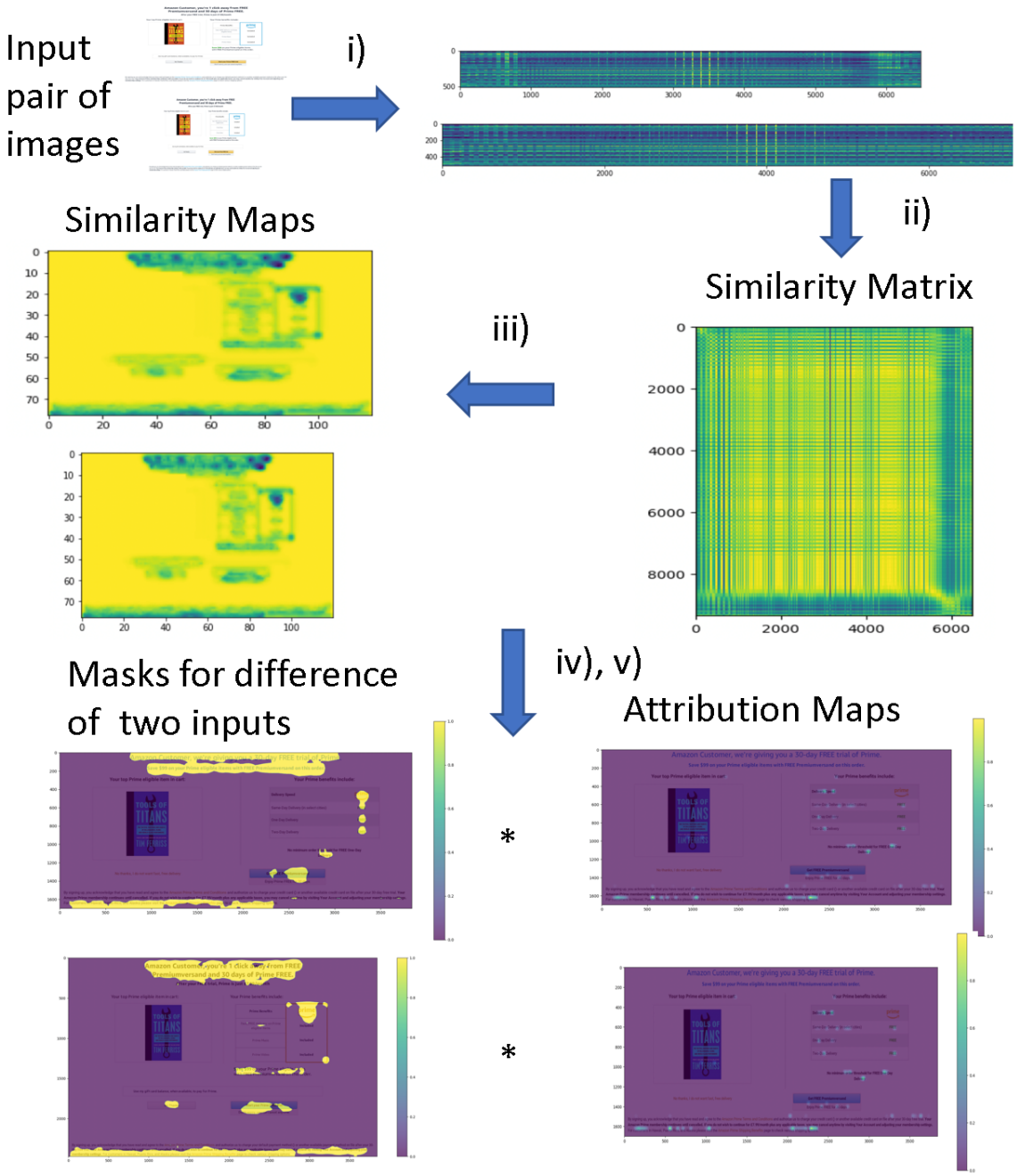


FIGURE 3.5: Visual explanation of image evaluation Algorithm 3.

and 3. 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

Algorithm 3: 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 $k \leftarrow 0$, $d_C \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$ and $d_y \leftarrow \vec{0}^{\frac{n*(n-1)}{2}}$.

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:

$A_i \leftarrow \Phi(x_i)$, $A_j \leftarrow \Phi(x_j)$;

reshape A_i and A_j:

$A_i \leftarrow \text{reshape}(A_i, (m'n', Z'))$,

$A_j \leftarrow \text{reshape}(A_i, (Z', m'n'))$;

ii. Compute the Cosine Similarity matrix between every feature vector in A_i and every feature vector in A_j:

$$S_{i,j}[k, l] \leftarrow \frac{< A_i[k, :], A_j[:, l] >}{|A_i[k, :]||A_j[:, l]|},$$

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

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

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

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

iv. Threshold d_{x_i}, d_{x_j} and resize them to the same dimension as C_i, C_j:

$P_i \leftarrow \text{threshold}(d_{x_i}), \quad P_j \leftarrow \text{threshold}(d_{x_j})$;

Resize P_i and P_j:

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

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:

$d_C[k] \leftarrow d_C, \quad d_y[k] \leftarrow d_y$;

$k \leftarrow k + 1$;

end ;

2) Compute Pearson Correlation ρ between d_C and d_Y .

Output ρ.

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.

Insight Examples. Figures 3.4 and 3.5 provide visual explanations of our insights evaluation algorithm in text and image settings, respectively. Figure 3.4, 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 3.5, 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.

3.6 Experiments

We evaluate our algorithm on the dataset described in Section 3.2. 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.

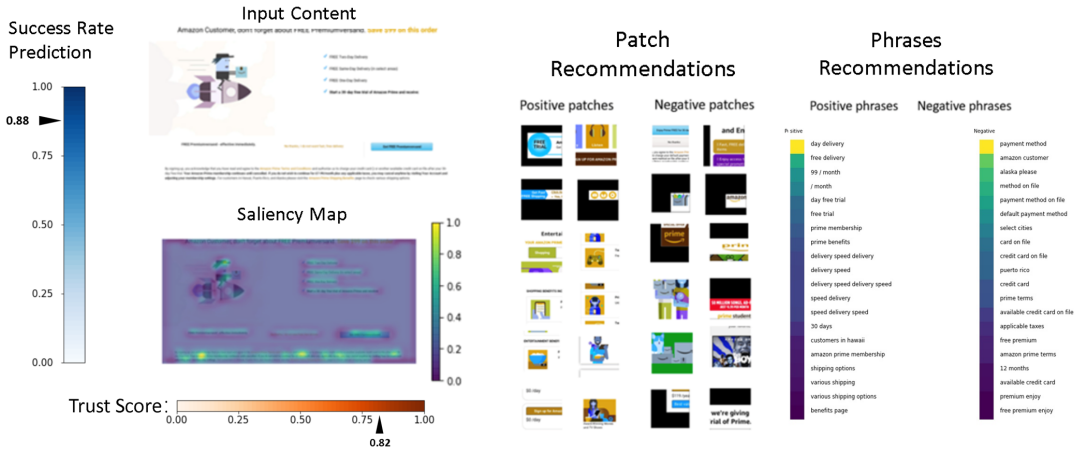


FIGURE 3.6: Exemplar dashboard of our interactive system to refine existing content or design new content.

3.6.1 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 Jesson et al. (2020). 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.

We include details of neural network architecture here. In Table 3.1, 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. "GlobalAverage-Pooling2D" is the global average pooling operation in the spatial dimension of the tensors, functioning the same as Keras' Global Average Pooling 2D Keras (2022). "ResBlock" is the standard ResNet block He et al. (2016a). In the brackets, we provide the kernel size, stride, and number of filters. "TransformerLayer" is the standard layer in a transformer Vaswani et al. (2017). In the brackets, we provide the size of hidden layers and the number of attention heads.

3.6.2 Interactive Dashboard

For ease of use, we propose an interactive dashboard for digital marketers to visually work their content (see Figure 3.6). Our dashboard aims to provide similar functionality to Sinha et al. (2020), but our framework turns out to be more powerful and comprehensive.

Table 3.1: The architecture of each component in our multimodal neural network.

ResNet(·)

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(·)

Layer	Type
1-12	TransformerLayers(768, 12)

$MLP_1(\cdot)$

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

$MLP_2(\cdot)$

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

$MLP_3(\cdot)$

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

Specifically, our dashboard has merits that facilitate digital marketing design.

1. In Sinha et al. (2020), 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 3.6.

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 3.2 show our model outperforming the rest. We also present an insights "Trust Score", which is based on the insights evaluation results in Table 3.3.

3.6.3 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 3.6.2 to design marketing contents from real-world digital marketers.

Success rate and pairwise prediction Table 3.2 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 Sinha et al. (2020); 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

Table 3.2: 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 3.3: Results of insights evaluation. The performance metric is the percentage increase of Pearson Correlation Coefficient defined in Equation 3.7 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%**

Algorithm 4: Pairwise Accuracy.

Data: Predictions $\widehat{y} = [\widehat{y}_1, \ldots, \widehat{y}_n]$, truth $y = [y_1, \ldots, y_n]$
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

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 Ackerman and Chen (2011) between the control and each treatment counterpart to evaluate the performance of our models. Algorithm 4 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 3.3, we evaluate the insights generated from the trained deep neural networks using our proposed evaluation scheme (see Section 3.5). In our dataset, we have multiple 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 3.6.2. The demo dashboard looks similar to Figure 3.6, 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.

Additional Result In this section, we offer an additional result to support the finding in our main paper. Specifically, Figure 3.7 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 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 domains.

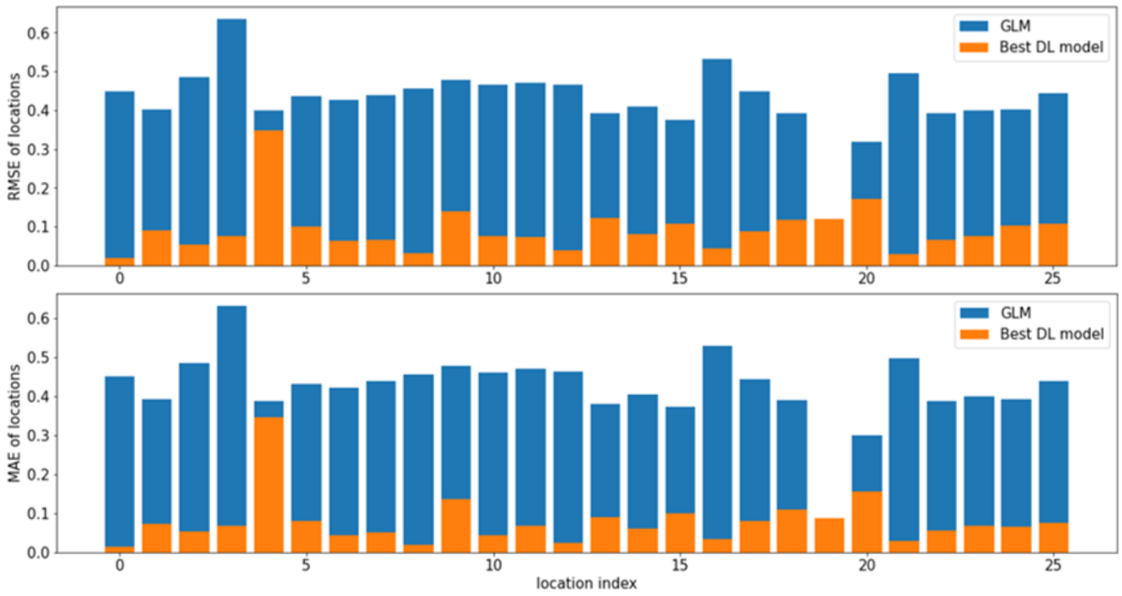


FIGURE 3.7: RMSE(top) and MAE(bottom) of GLM(blue) and our multimodal neural network(orange) evaluated on each domain.

3.7 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 3.4 and 3.5.

Modeling Digital Marketing Contents. The problem of modeling digital marketing content has triggered substantial research efforts Fong et al. (2019); Sinha et al. (2020); Wang (2022); Zhou (2020) over the past decade. Fong et al. (2019) developed a machine learning pipeline to classify advertising images based on their quality. Wang (2022) combines deep neural network and evolutionary algorithm to predict optimal personalized marketing strategy for better incomes. Zhou (2020) 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. (2020), 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 Kolodner (1992); Schmidt et al. (2001), which draws conclusions for new inputs by comparing them with a few exemplar cases (i.e prototypes) in the problem domain Chen et al. (2019); Li et al. (2018). 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 Dutra et al. (2011); Geng et al. (2018), and mechanics predict potential malfunctions by recalling vehicles exhibiting similar symptoms Geng et al. (2023). 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 Chen et al. (2019); Li et al. (2018). Our framework is somehow similar to ProtoPNet Chen et al. (2019), 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 Adebayo et al. (2018); Rao et al. (2022); Agarwal et al. (2022) 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 Bohle et al. (2021); Cao et al. (2015); Fong and Vedaldi (2017). 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.

3.8 Discussion

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 OpenAI (2023) and SAM Kirillov et al. (2023). 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 Lee et al. (2022) or examining the effectiveness of a predicted business decision from an AI agent on the company's income/loss Alaluf et al. (2022).

4. Mitigating Test-Time Bias for Fair Image Retrieval

4.1 Introduction

Image search on the web based on text-based image retrieval (TBIR) (Lew et al., 2006) involves interpreting a user's (text) query and returning corresponding images that are considered relevant in terms of semantic meaning (Chen et al., 2015). With recent advancements in multi-modal representation learning, Vision-Language (VL) models such as CLIP have been widely used to enhance the efficacy of text-based image retrieval (Radford et al., 2021; Cao et al., 2022). These models are usually trained on vast datasets that consist of millions of text-image pairs scrapped from the web, which inevitably manifest societal biases especially for neutral queries (Wang et al., 2021a; Hall et al., 2023; Wang et al., 2021b), *i.e.*, queries without explicit demographic (gender or race) connotations. In Figure 4.1, we show image retrieval results for the neutral query "Bus Driver" using CLIP and an unbiased alternative delivered by the proposed approach.

In our work, we adhere to *equal representation* as our fairness objective (Mehrotra and Celis, 2021; Kay et al., 2015), as an alternative to *proportional representation* (Jalal et al., 2021; Berg et al., 2022). The latter, which aligns the demographic proportions in the retrieval results with those in the dataset, which is susceptible to biases during collection (Kay et al., 2015). Instead, equal representation for all demographic groups of interest attempts to mitigate (obscure) the influence of any inherent biases.

Previous works have been dedicated to developing unbiased VL models that promote fairness in TBIR tasks (Wang et al., 2021a; Berg et al., 2022; Wang et al., 2022a; Kim et al., 2023; Chuang et al., 2023; Parraga et al., 2022). Berg et al. (2022) leveraged adversarial training for debiasing in which a learnable prompt is prepended to the input text queries, while an adversary seeks to discourage the image and text encoders from capturing gender or race information for retrieval. Alternatively, Wang et al. (2021a) discarded the components of the visual and text representations that are highly correlated with gender, by estimating the mutual information between these components and gender attributes. Further, Wang

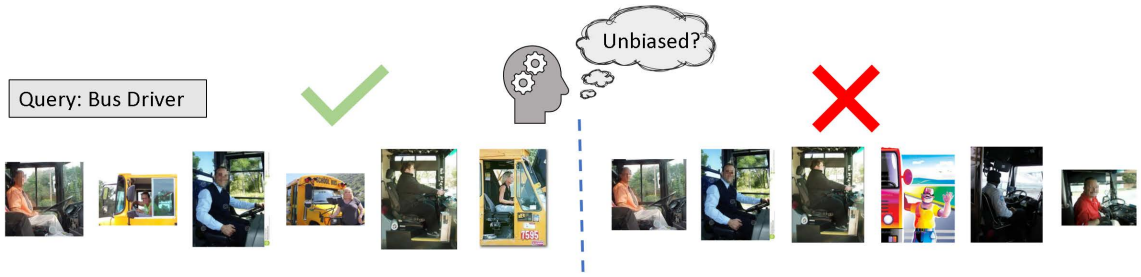


FIGURE 4.1: Text-based image retrieval (TBIR) results of a neutral query. Right: all returned images are male bus drivers, possibly pushing the message that only men (can) perform this job. Left: desirable unbiased result with equal gender representation obtained by the proposed PBM approach.

et al. (2022a) neutralized the gender contributions in image representations by enforcing equal contributions from gender (male and female) features using a bias contrast loss, while maximizing the contribution of gender-irrelevant features. These studies largely rely on demographic attributes that can be somewhat visually perceived. For instance, gender attributes are derived from perceived characteristics of masculinity or femininity, leading to labels such as "Male" or "Female". Similarly, race-related attributes are often characterized based on skin tones and typically categorized into groups such as "Fair Skin" and "Dark Skin" (Kay et al., 2015; Celis and Keswani, 2020). However, these methods primarily focus on debiasing the gender or race encoding within VL models, which may be insufficient due to bias or imbalance in the image retrieval candidate pool (test set). In contrast, though we also mitigate bias in TBIR using the same demographic attribute annotations as other works, the proposed solution is simpler in the sense that it forgoes the need for access to gradients or retraining the VL model.

In this paper, we start Section 4.3.1 by defining the text-based image retrieval task in the context of a fairness objective focused on equal representation. We then analyze the effectiveness of existing bias reduction methods in Section 4.3.4 and demonstrate how they fall short in achieving equal representation, mainly due to imbalances in the test-time image retrieval set. Based on this observation, in Section 4.3.5 we propose a simple yet effective post-processing debiasing method called post-hoc bias mitigation (PBM) that

creates fair retrieval subsets guided by predicted gender (or race) attributes obtained from either an off-the-shelf gender (or race) classifier or zero-shot inference using a pre-trained VL model. In Section 4.4, we evaluate PBM on real-world web image search (Kay et al., 2015; Celis and Keswani, 2020) and large-scale image-text datasets such as MS-COCO (Chen et al., 2015) and Flickr30K (Plummer et al., 2015). By comparing our approach to various existing bias-mitigation techniques, we show that our method achieves the lowest bias while maintaining satisfactory retrieval performance, as evidenced by both quantitative and qualitative results. Importantly, PBM strives to provide a more unbiased, fair and diverse visual representation of different groups in search results, ultimately promoting social welfare by challenging existing social perceptions of gender and race.

The summarized contributions of this work are:

- We present an analysis of the effect of existing debiasing methods on VL models, and highlight their insufficiency in achieving equal representation for text-based image retrieval.
- We propose PBM, a straightforward and efficient test-time post-processing debiasing algorithm that generates fair retrieval subsets, guided by predicted gender/race information obtained from an off-the-shelf classifier or inferred via zero-shot using the VL model itself.
- We evaluate PBM on two real-world web image search and two large-scale image-text datasets for text-based image retrieval, and compare with existing bias-mitigation techniques, demonstrating its effectiveness in achieving the lowest bias among all tested methods.

4.2 Related work

Text-based image retrieval Text-based image retrieval is the process of searching and retrieving images from a large database using textual descriptions or keywords as queries. The task is usually tackled by image-text feature alignment (Cao et al., 2022), which embeds

image and text inputs into a shared feature space such that relevant image and text features are close to each other. In recent years, substantial progress has been made due to the emergence of large-scale datasets, as well as the development of effective deep learning-based image-text models. Pioneering works include DeViSE (Frome et al., 2013), which bridges the *semantic gap* between image content and textual descriptors by aligning CNN-based features of images with textual embeddings from ImageNet labels. Subsequent approaches, such as VSE++ (Faghri et al., 2017) and SCAN (Lee et al., 2018), further refine these joint embeddings to improve retrieval performance. Recently, OpenAI's CLIP (Radford et al., 2021) has emerged as a powerful approach for image retrieval based on similarity matching between extracted image and text features. CLIP leverages a pre-trained transformer model, jointly optimized for both image and text understanding, which allows it to effectively match images and textual descriptions.

Fairness in machine learning Recent studies have highlighted numerous unfair behaviors in machine learning models (Angwin et al., 2016; Buolamwini and Gebru, 2018). For example, a risk assessment algorithm used in the United States criminal justice system predicted that Black defendants were more likely to commit future crimes than white defendants, even after controlling for criminal record (Angwin et al., 2016). Moreover, individuals with different gender and skin-tones are likely to receive disparate treatment in commercial classification systems such as Face++, Microsoft, and IBM systems (Buolamwini and Gebru, 2018). Consequently, there has been a surge in demand and interest for developing methods to mitigate bias, such as regularizing disparate impact (Zafar et al., 2015) and disparate treatment (Hardt et al., 2016), promoting fairness through causal inference (Kusner et al., 2017), and incorporating fairness guarantees in recommendations and information retrieval (Beutel et al., 2019; Morik et al., 2020).

Fairness in vision-language models After some studies revealed the bias of using VL models in downstream tasks (Wang et al., 2021a; Hall et al., 2023; Wang et al., 2021b), efforts to address and mitigate these biases in VL models have gained increasing attention. Existing

solutions for fair vision-language models can be generally classified into pre-processing, in-processing, and post-processing methods. Pre-processing techniques usually involve re-weighting or adjusting the training data to counter imbalances across demographic attributes, while preserving the utility of the dataset for the target task (Friedler et al., 2014; Calmon et al., 2017). In-processing methods focus on altering the training objective by incorporating fairness constraints, regularization terms or leveraging adversarial learning to obtain representations invariant to gender/race (Berg et al., 2022; Wang et al., 2023a; Xu et al., 2021; Cotter et al., 2019). Post-processing approaches achieve fairness by applying *post-hoc* corrections to a (pre-)trained model (Cheng et al., 2021; Calmon et al., 2017) or via feature clipping (Wang et al., 2021a) on the output of image-text encoders based on mutual information.

Our work lies in the post-processing category of debiasing methods that encourages equal representation of diverse demographics. We also identified the fair subset selection approach used in Mehrotra and Celis (2021) as a potential post-hoc debiasing method for TBIR. While Mehrotra and Celis (2021) shares our goal of ensuring equality of gender/race attributes in the set of results, their focus did not extend to the TBIR scenario with an underlying VL model nor detail an effective method for obtaining accurate demographic attributes for debiasing. More importantly, they assumed demographic attributes seen by their algorithm to be available ground truth labels subject to noise. This assumption creates difficulties when attempting to adapt their method to a real-world problem such as TBIR. Complementary, our approach is meant to address these deficiencies by providing a practical debiasing procedure, that includes acquiring demographic attributes.

Gender/Racial Bias in Web Image Search Our research is closely related to studies in the Human Computer Interaction community that demonstrated gender inequality issues in current online image search systems (Kay et al., 2015; Noble, 2018; Celis and Keswani, 2020). These studies revealed how gender bias in occupational image search results influences people's perceptions about the presence of men and women in various professions. Our

work builds upon these findings by examining the gender and racial bias in image search algorithm and offering innovative solutions for reducing bias in popular image retrieval framework using pre-trained VL model, such as CLIP (Radford et al., 2021).

4.3 Method

4.3.1 Problem formulation

Image retrieval Suppose $\mathcal{C}$ is the set of all text queries of interest (gender-neutral defined below), and $\mathcal{V} = \{v_i\}_{i=1}^{N}$ is a database of N images, from which we retrieve images given query inputs. Each query input $c \in \mathcal{C}$ is relevant to a ground-truth set of images $V_c^* \subseteq \mathcal{V}$ provided by human annotators.

The text-based image retrieval task aims at finding images from the database so they best match the text query inputs. Specifically, given any text query $c \in \mathcal{C}$ and a fixed retrieval size K ($K \ll N$) as inputs, the goal is to design an algorithm that returns a bag of K images $V_{c,K} \subseteq \mathcal{V}$, where $|V_{c,K}| = K$, such that $V_{c,K}$ contains as many relevant images from V_c^* as possible.

For evaluation purposes, a top-K recall score ("Recall@K") is usually computed to quantify whether the most relevant images are included in the retrieval output. Specifically, we write

$$\text{Recall@}K = \frac{\sum_{c \in \mathcal{C}} |V_{c,K} \cap V_c^*|}{\sum_{c \in \mathcal{C}} |V_c^*|} \times 100\%,$$

where K is selected specifically for each dataset so that $|V_c^*| \leqslant K$ for most queries $c \in \mathcal{C}$; this way, the recall can be realistically close to 100%.

Debiasing in image retrieval Using gender as a motivating scenario, we are interested in gender debiasing in cases where queries relate to human characteristics with no gender connotations, which we refer to as *gender-neutral queries*. For example, queries could be occupations that are widely open to all individuals, irrespective of gender (Organization, 2019), such as chef, nurse, social worker, *etc.*

We anticipate that gender-neutral queries should yield image retrieval sets that comprise equal representation of both male and female associated images, *i.e.*, in a 1:1 ratio, which is consistent with the equal representation goal discussed by Wang et al. (2021a, 2023a); Mehrotra and Celis (2021); Mehrotra and Vishnoi (2022). Note that we use gender debiasing as an example, but the same setting can be generalized to other debiasing issues such as racial discrimination, as we will demonstrate in the experiments.

Specifically, assume each image $v \in \mathcal{V}$ has a gender attribute $g(v) \in \{+1, -1\}$, which corresponds to the two genders manifested in the image, male and female, respectively. For each gender-neutral query c, the gender bias of the resulting retrieved bag of images $V_{c,K}$ is defined as the normalized absolute difference between the numbers of images of each gender, *i.e.*,

$$B(V_{c,K}) = \frac{1}{K}\left|\sum_{v \in V_{c,K}} 1\{g(v) = +1\} - \sum_{v \in V_{c,K}} 1\{g(v) = -1\}\right| = \frac{1}{K}\left|\sum_{v \in V_{c,K}} g(v)\right|, \tag{4.1}$$

which ranges from $\frac{1}{K}\text{mod}(K, 2)$ (minimum bias) to 1 (maximum bias). In our retrieval approach, we average $B(V_{c,K})$ across all gender-neutral queries $c \in C$ as an evaluation metric for fairness, *i.e.*,

$$\text{AbsBias@}K = \frac{1}{|C|}\sum_{c \in C} B(V_{c,K}) = \frac{1}{|C|}\sum_{c \in C}\frac{1}{K}\left|\sum_{v \in V_K^c} g(v)\right|.$$

The goal is to minimize AbsBias@K while maintaining a satisfactory retrieval Recall@K.

4.3.2 Similarity-based image-text matching

Overall framework As in previous works (Singh et al., 2003; Bai et al., 2014; Zaidi et al., 2019; Cao et al., 2022; Mukhoti et al., 2022), we use a similarity matching approach to tackle the image retrieval task. Such an approach involves aligning image and text features from largely pre-trained vision and language models during training time. At inference time, we rank images based on whether their features are similar to the input query text features

and use the top-ranked images as our retrieval output. Specifically, we use an image encoder network $f_\phi(\cdot)$ and a text encoder network $f_\psi(\cdot)$ to embed both image v and text c inputs into a shared d-dimensional feature space as $f_\phi(v), f_\psi(c) \in \mathbb{R}^d$. A cosine similarity score $S(v, c)$ is then computed to quantify the relevance between v and c (Singh et al., 2003; Bai et al., 2014; Wang et al., 2021a).

Training As in previous works (Chen et al., 2020; Radford et al., 2021), the training of the image and text encoders is achieved through optimizing an NT-Xent (Normalized Temperature-scaled Cross Entropy) loss with stochastic gradient descent on mini-batches.

Inference Given a new image database $\mathcal{V}^{\text{test}}$, for each new query input $c \in \mathcal{C}^{\text{test}}$, we compute its similarity score $S(v_i, c)$ with every image $v_i \in \mathcal{V}^{\text{test}}$ and then pick the top-K images that have the maximum similarity scores to form our retrieved set $V_{c,K}$.

4.3.3 Fairness criterion for image retrieval

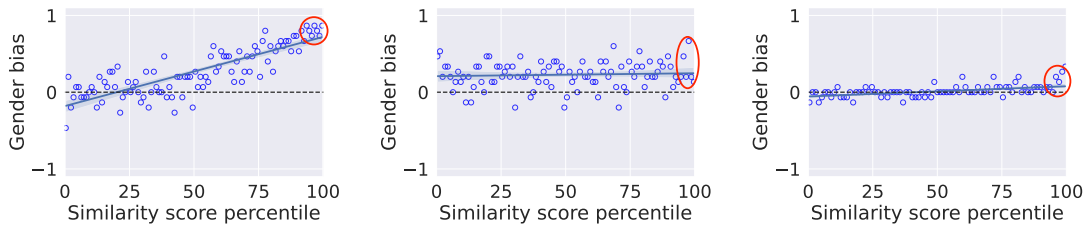
Fairness can be defined in numerous ways in favor of different situation (Saxena et al., 2020). In the context of fair image retrieval (Wang et al., 2021a; Berg et al., 2022), we define our fairness objective as *equal representation*, implying that the retrieved image set through the retrieval algorithm should encompass an equal number of samples from each demographic group. We formally define our criterion as follows:

Definition 3 (Equal Representation)**.** *An image retrieval algorithm satisfies equal representation with respect to binary demographic attributes $g(v)$ if,*

$$\mathbb{E}_{V_c \sim P}[\mathbb{E}_v[1(g(v) = +1)]] = \mathbb{E}_{V_c \sim P}[\mathbb{E}_v[1(g(v) = -1)]]$$

where P represents the distribution of retrieved image set V_c corresponding to neutral queries c.

The above definition implies $\mathbb{E}[B(V_c)]_{V_c \sim P} = 0$, *i.e.*, the expected bias is zero for any retrieved image sets. Our definition of *equal representation* is equivalent to "Equal



(a) CLIP (no debiasing) (b) MI debiasing (c) PBM

FIGURE 4.2: Gender bias distribution for different methods using "engineer" as query. We compute similarity scores for all images from the test image set and plot them against gender bias in 1% quantile increments. The red circle marks the top-K window covering the final retrieval output $V_{c,K}$.

opportunity for ranking distributions" introduced by Singh and Joachims (2017) when ranking position bias is constant.

4.3.4 Bias analysis

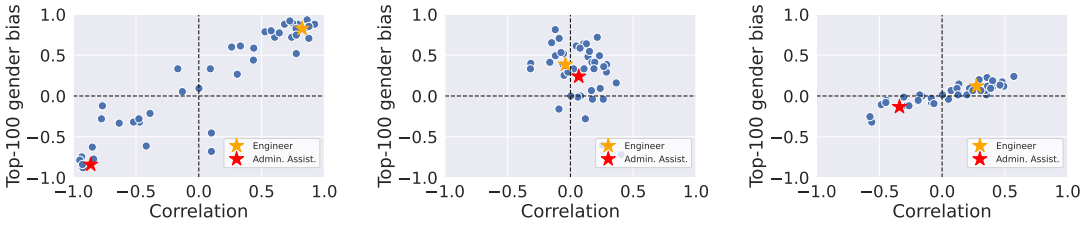
Using the image retrieval framework defined above as a basis, it is important to incorporate strategies to reduce gender (and race) biases of the retrieval output $B(V_{c,K})$. Below, we briefly analyze existing methods and discuss their limitations.

Existing methods Most methods address the gender fairness issue by enforcing model features to be less dependent on gender information. This is achieved by mainly two types of approaches, namely adversarial training (Edwards and Storkey, 2015; Berg et al., 2022; Xu et al., 2021) and mutual information (MI) minimization (Wang et al., 2021a, 2023a). Specifically, adversarial training involves training a separate (adversarial) classifier network by adding an adversarial loss so that the adversarial network cannot distinguish gender given the encoded image features (Edwards and Storkey, 2015; Berg et al., 2022; Xu et al., 2021). Alternatively, MI minimization aims at reducing the MI between feature distribution and their gender by clipping feature dimensions highly correlated with gender (Wang et al., 2021a). Both methods encourage the model to extract features that are *independent* of gender. However, we argue below that enforcing this kind of independence between image features and gender is not sufficient to effectively eliminate gender bias in image retrieval.

Illustrative example We start by showing debiasing results using mutual information minimization. Similar insights can be obtained via adversarial training, which is shown in the Supplementary Material (SM). Figure 4.2 shows an example for the query "engineer" comparing three methods, namely, CLIP (no debiasing), MI-based debiasing and the proposed PBM (described below). For each method, we first compute, sort and group the similarity scores of all images into 1% quantiles ($\sim$ 30 images each). Then, for each quantile, we compute the gender bias as defined in (4.1). This analysis shows the relationship between gender bias and similarity scores. Note that for retrieval, we use samples with the largest similarity scores to form our retrieved bag of images, thus only the right-most data points (highlighted in the figure) are part of the retrieval set. We show percentiles in the figure to emphasize similarity rank rather than the similarity scores *per se*.

As shown in Figure 4.2(a), the original image retrieval algorithm with no debiasing based on CLIP tends to assign higher scores (at larger percentiles) to samples associated with male attributes, as evidenced by the high correlation between gender and the similarity scores. This makes the final retrieval output largely biased towards male samples. In contrast, Figure 4.2(b) shows the result of debiasing via MI minimization (Wang et al., 2021a). Using the regression line as a visual guide, we see that the correlation between gender and similarity score is close to zero. However, the gender bias is consistently larger than zero across the range of similarity scores, so the final retrieval is still biased. A desirable debiasing outcome is that for which the regression line aligns with the dotted line (zero gender bias across the similarity range), as shown in Figure 4.2(c). This is accomplished by the proposed PBM (described below).

Example with multiple queries In order to get a more generalizable understanding of the behavior shown in Figure 4.2, we repeat the analysis with all 45 occupations from the dataset provided by Kay et al. (2015) and visualize the results in Figure 4.3. Specifically, for each query (occupation), we calculate the Spearman's rank correlation between all (test-set) similarity scores and gender bias similar to Figure 4.2. We use Spearman's



(a) CLIP (no debiasing) (b) MI debiasing (c) PBM

FIGURE 4.3: Comparing top-100 retrieval gender bias with full set similarity gender bias Spearman's correlation. For each query (occupation), we visualize the correlation between similarity score and gender bias against the top-100 retrieval gender bias. Two typical examples: "engineer" and "administrative assistant" are highlighted for illustration.

rank correlation because it is more appropriate to quantify associations between (variable) rankings. Moreover, we also calculate the gender bias of the final retrieval bag ($K = 100$) for each occupation. Consistent with the illustrative findings on the "engineer" query, in Figure 4.3(a), we see a large correlation between similarity scores and gender bias as well as a large retrieval-set gender bias. MI minimization in Figure 4.3(b) manages to reduce the correlation between similarity score and gender but suffers from considerable gender bias; mostly clustered around 0.4. The proposed PBM not only reduces the correlation between similarity scores and gender bias but also significantly pushes the bias of each retrieval result to zero, which is the sought target debiasing outcome.

Insights Our observations can be explained and understood from a theoretical perspective. Previous methods are mostly concerned with making their image features independent of gender information. For a fixed query c, suppose we sample a random image from the test image data distribution $v \sim \mathcal{V}^{\text{test}}$, its feature distribution $f_\phi(v)$ is independent of gender distribution $g(v)$. Since the query feature $f_\psi(c)$ is constant, the similarity score $S(v, c)$, defined in Section 4.3.2, is a deterministic function of $f_\phi(v)$ and is thus also independent of gender $g(v)$. This is why the results for MI-based unbiasing shown in Figure 4.2(b) and 4.3(b) show little correlation between similarity scores and gender bias. Similar results for debiasing with adversarial learning can be found in the SM.

However, as shown in Figure 4.2(b), there is still a consistent gender bias across similarity

Algorithm 4: Post-hoc Bias Mitigation (PBM).

Input: Text query c, retrieval size K, image database $\mathcal{V}^{\text{test}}$, similarity measure from pre-trained vision-language models $S(\cdot,\cdot)$, and gender prediction model $\hat{g}(\cdot)$.
Output: Image retrieval bag $V_{c,K}$.

Split $\mathcal{V}^{\text{test}}$ into $\mathcal{V}^{\text{test}}_{+1}$, $\mathcal{V}^{\text{test}}_{-1}$ and $\mathcal{V}^{\text{test}}_{\text{N/A}}$ using the gender prediction model $\hat{g}(\cdot)$;
Let $V_{c,K} = \emptyset$;
while $|V_{c,K}| < K$ **do**
  $v_{+1} =_{v \in \mathcal{V}^{\text{test}}_{+1}} S(v,c)$; $v_{-1} =_{v \in \mathcal{V}^{\text{test}}_{-1}} S(v,c)$; $v_{\text{N/A}} =_{v \in \mathcal{V}^{\text{test}}_{\text{N/A}}} S(v,c)$;
  if $[S(v_{+1},c) + S(v_{-1},c)]/2 > S(v_{\text{N/A}},c)$ **then**
    $V_{c,K} \leftarrow V_{c,K} \cup \{v_{+1}, v_{-1}\}$; $\mathcal{V}^{\text{test}}_{+1} \leftarrow \mathcal{V}^{\text{test}}_{+1} \backslash \{v_{+1}\}$; $\mathcal{V}^{\text{test}}_{-1} \leftarrow \mathcal{V}^{\text{test}}_{-1} \backslash \{v_{-1}\}$;
  else
    $V_{c,K} \leftarrow V_{c,K} \cup \{v_{\text{N/A}}\}$; $\mathcal{V}^{\text{test}}_{\text{N/A}} \leftarrow \mathcal{V}^{\text{test}}_{\text{N/A}} \backslash \{v_{\text{N/A}}\}$;
  end if
end while
return $V_{c,K}$

score values. This means that the gender distribution in each window tends to manifest the gender distribution of the whole test image set $\mathcal{V}^{\text{test}}$, which may not be balanced. This is also confirmed in Figure 4.3(b), as the final output bias of most occupations is clustered around 0.4, which is precisely the gender bias of the entire image set $\mathcal{V}^{\text{test}}$. Therefore, ensuring independence, quantified here in terms of correlation, between image features and gender may not guarantee zero gender bias if the test image set itself is biased due to imbalance.

Two types of bias From the examples above, we can differentiate two types of bias, namely, the model bias from training and the bias from the test-time image distribution. The former can be quantified based on the correlation between similarity score and gender and only depends on the training data distribution and the way in which the model is trained. This has been previously described and studied (Caliskan et al., 2017; Zhao et al., 2017). In comparison, the latter type of bias manifests in the test phase because the test image set (the database from which we retrieve images) does not necessarily have commensurate numbers of male and female samples. We refer this type of bias as test-time bias. In fact, such proportions are usually unknown for image databases.

These two sources of bias coexist and can be addressed separately during model training and inference. Previous methods (Edwards and Storkey, 2015; Wang et al., 2021a; Berg et al., 2022) have been fairly successful at addressing the first type of bias on the training side, but neglect the test-time bias that is specific to the test set. To tackle the test-time bias, it is necessary to find a strategy targeting the test image set, so that the retrieved image genders are balanced despite the gender imbalance in the training (source) set. This very insight motivates our approach called PBM, which we describe below. As previously shown in Figures 4.2(c) and 4.3(c), PBM achieves substantially smaller retrieval gender bias than MI debiasing. Other existing approaches will be considered in the experiments.

4.3.5 Post-hoc Bias Mitigation (PBM)

To address the second type of bias induced by the imbalanced image test set, one simple idea is sub-sampling. Specifically, we could first sub-sample the image set to make sure its gender ratio is balanced *before* doing retrieval. However, a clear limitation of such an approach is that some highly relevant images may be dropped during sub-sampling, which may negatively affect retrieval quality. This problem is especially exacerbated if the test set has a very large gender bias. Alternatively, our intuition is to sub-sample *after* computing and ranking similarity scores of all images using a post-hoc method to control gender bias while sampling from the image source set, which we call Post-hoc Bias Mitigation (PBM).

A general version of the PBM algorithm, which is straightforward, is presented in Algorithm 5. For each image $v_i \in \mathcal{V}^{\text{test}}$, we first predict its gender $\hat{g}(v_i) \in \{+1, -1\}$, which splits the images into two subsets, $\mathcal{V}^{\text{test}}_{+1}$ and $\mathcal{V}^{\text{test}}_{-1}$. We then rank images from the two subsets based on their similarity scores separately. While forming the retrieval bag, we sample the top of both subsets together in pairs. Specifically, we select the top $\lfloor K/2 \rfloor$ samples from each subset. If K is odd, we randomly pick one of the subsets and select one more top sample from it. This method ensures low gender bias of the retrieval output as long as gender predictions $\hat{g}(v_i)$ are accurate.

One extension of our method considers the case where gender may not be applicable

to some images. For instance, our image dataset may contain cartoon characters that are not easily associated with any gender, or maybe the gender cannot be determined from the image due to body or facial coverings. In this case, we allow our gender predictions to take N/A values, yielding a $\mathcal{V}_{\text{N/A}}^{\text{test}}$ subset. Images from $\mathcal{V}_{\text{N/A}}^{\text{test}}$ do not inherently exhibit visually perceptible bias. Thus, they could be selected in favor of other male-female pairs if their similarity scores are higher and are exempt from bias measurement.

Gender prediction To predict gender $\hat{g}(v)$, we consider the following two methods for different scenarios. Note that all options are considered in the experiments.

Supervised gender classification: For scenarios where ground-truth gender attributes are available, such as MS-COCO (Lin et al., 2014; Zhao et al., 2021), we can train a complementary small gender classification network. Using our encoded image feature $f_{\phi}(v)$ as input, we train a 3-layer MLP to classify gender. For datasets where gender may not be applicable to some samples, we add a N/A class to the set of labels for predictions.

Zero-shot inference using word embeddings or prompt: For scenarios where supervised training of gender is not possible, we can infer gender in a zero-shot manner (Radford et al., 2021; Li et al., 2022a) using the implicit knowledge embedded in the text features of large pre-trained vision-language models (Radford et al., 2021; Li et al., 2022a). Given that the semantics of gender and race attributes are already incorporated into the text encoder, we can extract them from word embeddings using words such as "Man" and "Woman". Alternatively, we could use prompts prepended to the occupation search query. For example, we add gender-specific adjectives like "Male" or "Female" in front of a query. Finally, we compute the (cosine) similarity score of each image v with them, *i. e.*, $S(v,$"Male " + c), $S(v,$"Female " + c). The plus operator (+) here refers to string concatenation. The gender prediction $\hat{g}(v)$ is then generated by comparing these two similarity scores.

Table 4.1: Results for debiased image retrieval from Occupation 1 and 2 datasets.

Method	Occupation 1 - Gender		Occupation 2 - Gender		Occupation 2 - Race	
	AbsBias@100 (↓)	Recall@100(↑)	AbsBias@100(↓)	Recall@100(↑)	AbsBias@100(↓)	Recall@100(↑)
Random Selection	.6370	-	.3155	-	.4171	-
CLIP Original (Radford et al., 2021)	.6231	58.3	.3566	46.2	.5002	46.2
MI-*clip* (Wang et al., 2021a)	.3769	47.0	.2539	42.2	.4099	42.3
Adversarial Training (Edwards and Storkey, 2015)	.2316	44.0	.2603	37.8	.4880	43.3
Debias Prompt (Berg et al., 2022)	.6373	**59.3**	.3564	**46.2**	.4946	**50.2**
CLIP-FairExpec (Mehrotra and Celis, 2021)	.2498	47.0	.2619	44.2	.2788	34.7
PBM - Zero-shot Embedding	.0969	49.8	.1150	42.1	.3133	40.2
PBM - Zero-shot Prompt	**.0560**	46.1	**.0443**	42.5	.2571	36.0
PBM - Supervised Classifier	.1404	50.3	.1171	42.1	**.0955**	37.9
PBM - Ground-truth Gender and Skin-tone	.0000	49.1	.0000	42.4	.0000	41.3

Table 4.2: Results for debiased image retrieval from MS-COCO and Flickr30k datasets.

Dataset	Method	Gender Bias			Recall		
		Bias@1 (↓)	Bias@5(↓)	Bias@10(↓)	Recall@1(↑)	Recall@5(↑)	Recall@10(↑)
COCO-1k	SCAN (Lee et al., 2018)	.1250	.2044	.2506	47.7	82.0	**91.0**
	FairSample (Wang et al., 2021a)	.1140	.1951	.2347	**49.7**	**82.5**	90.9
	CLIP (Radford et al., 2021)	.0900	.2024	.2648	48.2	77.9	88.0
	MI-*clip* (Wang et al., 2021a)	.0670	.1541	.2057	46.1	75.2	86.0
	Our PBM	**.0402**	**.0961**	**.1082**	37.3	73.6	84.8
COCO-5k	SCAN (Lee et al., 2018)	.1379	.2133	.2484	25.4	54.1	67.8
	FairSample (Wang et al., 2021a)	.1133	.1916	.2288	26.8	**55.3**	**68.5**
	CLIP (Radford et al., 2021)	.0770	.1750	.2131	**28.7**	53.9	64.7
	MI-*clip* (Wang et al., 2021a)	.0672	.1474	.1611	27.3	50.8	62.0
	Our PBM	**.0492**	**.1006**	**.1212**	22.3	50.5	61.9
Flickr30K	SCAN (Lee et al., 2018)	.1098	.3341	.3960	41.4	69.9	79.1
	FairSample (Wang et al., 2021a)	.0744	.2699	.3537	35.8	67.5	77.7
	CLIP (Radford et al., 2021)	.1150	.3150	.3586	**67.2**	**89.1**	**93.6**
	MI-*clip* (Wang et al., 2021a)	.0960	.2746	.2951	63.9	85.4	91.3
	Our PBM	**.0360**	**.1527**	**.1640**	41.2	85.3	92.6

4.4 Experiments

We first evaluate our image retrieval algorithm on two real-world web image search datasets, Occupation 1 (Kay et al., 2015) and Occupation 2 (Celis and Keswani, 2020). We also test on two large-scale image-text datasets, MS-COCO (Lin et al., 2014) and Flickr30k (Plummer et al., 2015) to further validate the effectiveness of PBM in handling more complex text-based image retrieval scenarios.

For comparison, we consider adversarial training (Edwards and Storkey, 2015; Berg et al., 2022) and mutual information minimization (Wang et al., 2021a) as baseline methods. We also include other types of debiasing methods such as FairSample (Wang et al., 2021a), which balances the gender distribution of image-text pairs within a training batch, and FairExpec (Mehrotra and Celis, 2021), a denoised selection algorithm designed to select fair subset based on noisy demographic attributes. We use AbsBias@KK and Recall@KK as evaluation metrics as described in Section 4.3.1.

Real-world web image search The first dataset, which we refer to as Occupation 1 (Kay et al., 2015), comprises the top 100 Google image search results for 45 gender-neutral occupation terms, such as "chef", "librarian", "primary school teacher", *etc.* Each image within this dataset is annotated with a crowd-sourced gender attribute (either "male" or "female") that characterizes the person depicted in the image. Occupation 2 (Celis and Keswani, 2020), the second dataset, includes the top 100 Google image search results for 96 occupations, where both gender and race (represented as skin-tone: fair and dark skin) annotations are provided. Notably, the gender and race attributes also include a N/A category in Occupation 2, where the annotators have chosen the option of "Not applicable" or "Cannot determine" for the gender or skin-tone represented in the image. Consequently, we treat the images labeled with N/A as neutral examples that do not contribute to the bias of retrieval, since in principle, the users cannot perceive gender or racial information from the image.

For these two datasets, we consider OpenAI's CLIP ViT-B/16 (Radford et al., 2021) as the VL model for all debiasing methods. The baselines for comparison are MI-*clip* from Wang et al. (2021a), adversarial training adapted from Edwards and Storkey (2015), and Debias Prompt from Berg et al. (2022). We have also tailored FairExpec (not originally proposed for TBIR) to our task by integrating it with CLIP and our proposed gender predictor $\hat{g}(\cdot)$. We refer to this adapted model as CLIP-FairExpec in our experiments. We test our PBM method with four variants of gender predictors $\hat{g}(\cdot)$, as discussed in Section 4.3.5. These variants include a supervised classifier, zero-shot inference, and ground truth from annotators. The implementation details are provided in the SM.

We summarize the experimental results in Table 4.1. Compared with other methods, PBM variants achieve significantly lower AbsBias@100 while maintaining comparable Recall@100 scores. The male:female ratio of images in Occupation 1 is approximately 61:49, thereby mitigating the first type of bias by MI-*clip*, adversarial training, debias prompt and random selection is not enough. The CLIP-FairExpec cannot achieve better results since the FairExpec algorithm relies on the independence assumption, while the samples in our test

dataset are correlated. This happens because images for occupations are collected from the same Google image search, which is biased. Further, FairExpec requires reliable probabilistic predictions for gender, however, in our setting the gender attributes are provided by an off-the-shelf MLP predictor based on visual features. In such a scenario, the label estimates yielded by the off-the-shelf MLP may not be always trustworthy, as the domain has shifted during inference. Consequently, these estimates could include misleading information, resulting in undesirable debiasing outcomes for CLIP-FairExpec. Moreover, it should be noted that Debias Prompt (Berg et al., 2022) always achieves the highest Recall@100, since we utilize their publicly accessible model, which has been fine-tuned on the Flickr30k dataset to enhance the performance of text-based image retrieval.

Large-scale text-based image retrieval We consider MS-COCO (Lin et al., 2014) and Flickr30k (Plummer et al., 2015). Our setup aligns with Wang et al. (2021a), where the gender attributes are directly inferred from the text captions of images. We consider the same baseline models as Wang et al. (2021a), which are SCAN (Lee et al., 2018) and CLIP, as well as their proposed approach, FairSample and MI-*clip*. It should be noted that FairSample is specifically designed to debias SCAN, a specialized in-domain VL model. We choose the best performing PBM model, which leverages the pre-trained classifier for gender attributes.

The performance metrics in Table 4.2 are consistent with Wang et al. (2021a), which resemble our AbsBias@K metric, with the omission of using absolute values in the sum. This bias metric is computed as Bias@$K = \frac{1}{|C|}\sum_{c\in C}\frac{1}{K}\sum_{v\in V_K^c} g(v)$. From the Table 4.2, we see that PBM maintains the bias to a minimum even when dealing with intricate text queries or images of complex scenes.

Bias-performance trade-off analysis In Figure 4.4, we show the trade-off between retrieval performance and bias for MI-*clip*, adversarial training and the four PBM variants. We choose MI-*clip* and adversarial training for comparison, as these models offer implementation simplicity of adjusting trade-off between AbsBias@100 and Recall@100. For

FIGURE 4.4: Trade-off between Recall@K and AbsBias@K for debiasing gender attributes within Occupation 1 (Middle) and race attributes using Occupation 2 (Right).

PBM, we introduce a trade-off parameter through a stochastic variable representing the probability of opting for a fair subset at any given time, as opposed to merely selecting the image with the highest similarity score. Additional details about this experiment are provided in the SM. Our results indicate that in a range of relatively high AbsBias@K values, MI-*clip* and adversarial training are still able to reduce bias while preserving Recall@K. However, in terms of lowering AbsBias@K, they struggle to maintain satisfactory TBIR performance. In contrast, the proposed PBM consistently succeeds in sufficiently reducing bias and maintaining performance, provided that predictions of attributes are reasonably accurate. Details of implementing this trade-off can be found in the SM.

Table 4.3: Group sensitivities and sensitivity ratios (ρ) for demographic attributes predicted by different classifiers on Occupation 1 - Gender and Occupation 2 - Race.

	Gender			Race		
Method	Male Sensitivity	Female Sensitivity	ρ	Light skin Sensitivity	Dark skin Sensitivity	ρ
PBM - Supervised Learning	0.97	0.88	1.10	**0.93**	**0.84**	1.11
PBM - Word Embedding	**0.98**	0.94	1.04	0.84	0.78	**1.08**
PBM - Zero-shot Prompt	**0.98**	**0.97**	**1.01**	0.88	0.81	1.09

Impact of Demographic Group Classifier on Debiased Results The demographic group classifier is an important module of our proposed method PBM. The debiasing result is intricately linked to the demographic group classifier's accuracy and prediction bias towards different demographic groups. Figure 4.5 showcases the relationship between the demographic group classifier's performance and the ensuing retrieval bias, by artificially introducing noise to the demographic group (logit) predictions via Gaussian noise with a standard deviation ranging from 0 (no noise) to 1. These results underscore that better

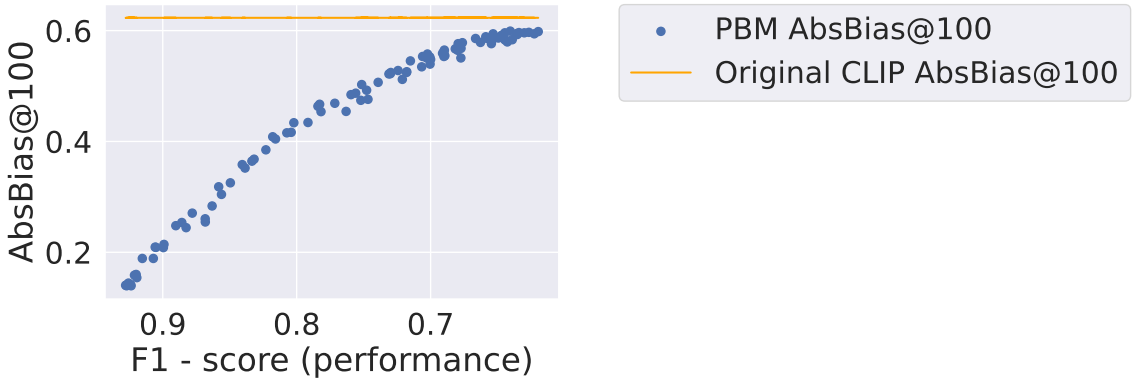


FIGURE 4.5: Relationship between the performance of the demographic group classifier (F1-score) and the retrieval bias (AbsBias@100) when utilizing PBM.

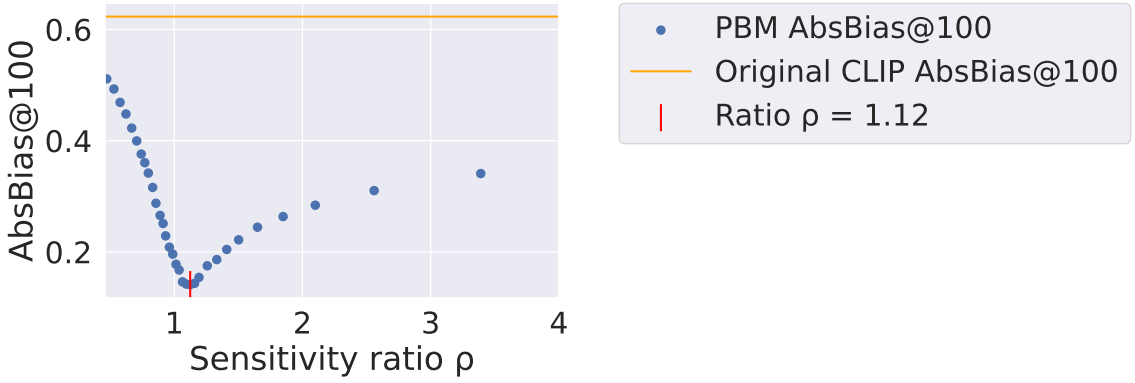


FIGURE 4.6: Relationship between the bias of the demographic group classifier (ratio of male sensitivity to female sensitivity) and the retrieval bias (AbsBias@100) when utilizing PBM.

group classifier performance yields lower bias, that bias converges to that of the original CLIP as the group classifier gets worse, and importantly, that the bias after PBM will be no worse than that of the original CLIP.

Further, Table 4.3 shows the individual demographic group sensitivities under three different scenarios, from which we can see that the group classifier is i) able to achieve good classification sensitivity (no lower than 0.81 and 0.90 in average), likely because demographic image attributes (gender and skin tone) are typically captured in images, and ii) that different scenarios exhibit different degrees of bias as measured by the group sensitivity ratio, which must be close to 1 for the model to be unbiased. Table 4.3 reveals

sensitivity discrepancies among different attributes. To delve deeper into the influence of classifier bias on PBM outcomes, we present the retrieval bias as a function of the sensitivity ratio in Figure 4.6. This is achieved by altering the gender classification threshold from 0 (maximizing male sensitivity) to 1 (minimizing male sensitivity). From Figure 4.6, we can conclude that the classifier bias does affect retrieval bias, however, only severely for more extreme sensitivity ratios, which is fortunately not the case in our results as shown in Table 4.3.

PBM on fine-tuned model In Table 4.4, we showcase the results of applying PBM to CLIp models that has been debiased by other approaches, such as MI-clip, Adversarial Learning, and Debias Prompt. When PBM is utilized in conjunction with other debiasing strategies, it exhibits a unique bias-recall trade-off, thus catering to a variety of application scenarios.

Table 4.4: Results of applying PBM - Supervised Learning on modified or fine-tuned CLIP.

Method	Occupation 1 - Gender		Occupation 2 - Race	
	AbsBias@100 (↓)	Recall@100(↑)	AbsBias@100(↓)	Recall@100(↑)
PBM	.1404	50.3	.0955	37.9
MI-*clip* - **PBM**	**.0780**	42.1	**.0737**	29.1
Adversarial Training - **PBM**	.1000	39.6	.0997	35.7
Debias Prompt - **PBM**	.1711	**52.1**	.1035	**40.6**

4.4.1 Neural Network Architectures

We summarize the details of the neural networks employed in our experiments in Table 4.5. For the Image Encoder, the Patch Extraction (dimensions: 16,16) extracts 196 non-overlapping 16×16 patches from the 224×224 image. These extracted patches are subsequently flattened. The subsequent Positional and Linear Embedding (768) maps these patch vectors onto a 768-dimensional space and adds 2D positional embeddings of patches to the 768-dimensional vectors. Next, 12 Vision Transformer Blocks (768, 12) processes the 768-dimensional embeddings. Each of these blocks features 12 self-attention heads. Lastly, the output embedding is obtained from a unique classification token ([CLS]) that we add

Table 4.5: The architecture of each component of CLIP and the MLP used in our experiments.

ImageEncoder(·)

Layer	**Type**
1	Patch Extraction(16, 16)
2	Positional and Linear Embedding(768)
4 - 15	Vision Transformer Blocks(768, 12)
16	[CLS] Token 1×768
17	Linear Projection (512)

TextEncoder(·)

Layer	**Type**
1	Positional and Token Embedding (512)
2 - 13	Transformer Blocks (512, 8)
14	[CLS] Token 1×512
15	Linear Projection (512)

MLP(·)

Layer	Type
1	fc-512 + BatchNorm + ReLU()
2	fc-512 + BatchNorm + ReLU()
3	fc-512 + BatchNorm + ReLU()
4	fc-n_class + Softmax()

to the input sequence of patch embeddings. The output from [CLS] Token 1×768 is then reduced from 768 dimensions to 512 dimensions using a Linear Projection (512).

Similarly in the Text Encoder, the initial phase involves Positional and Token Embedding (512). This step maps each token in the input text onto a 512-dimensional vector space and integrates positional embeddings into these vectors. Following this, the text encoder employs 12 Transformer Blocks (512, 8) to process these 512-dimensional embeddings. Each of these blocks contains 8 self-attention heads. Finally, the output embedding is derived from [CLS] Token 1×512. The subsequent Linear Projection (512) then maps the extracted text representation onto the multi-modal embedding space that aligns with the image embeddings.

4.5 Discussion

In this study, we examined gender and racial bias in text-based image retrieval (TBIR) for neutral text queries. In an attempt to identify bias in the test-time (inference) phase, we conducted an in-depth quantitative analysis on bias reduction, alongside existing debiasing methods and the proposed PBM. We concluded that solely addressing training-time model-encoded bias is not sufficient for obtaining equal representation results, because test-time bias also exists due to imbalance in the test image set used during retrieval. So motivated, we proposed Post-hoc Bias Mitigation (PBM), a straightforward post-processing method that aims to directly alleviate test-time bias. Experiments on multiple datasets show that our method can significantly reduce bias while maintaining satisfactory retrieval accuracy at the same time.

Moreover, the potential impact of PBM extends far beyond the initial scope of text-based image retrieval systems. The core concept of our methodology can be seamlessly adapted to a wide variety of information retrieval systems, such as image-based text retrieval or query-by-example image retrieval, as long as the demographic information of the test set is accessible or can be estimated, *e.g.*, via zero-shot inference. Overall, our approach is not limited to enhancing fairness in text-based image retrieval, thus can be extended to a broad range of VL model applications.

4.6 Limitations

Some limitations of the proposed method are duly acknowledged. Firstly, the efficacy of our approach is dependent upon the availability of a sufficient number of examples for each category (gender or racial attribute) within the test image set. Our work currently does not consider any techniques, such as using synthetic samples, to mitigate the issues arising from insufficient representations of certain demographic groups. Secondly, the debiasing effect of PBM pertains to the predictability and accessibility of demographic attributes. Attempts at debiasing religious representation, or other socio-cultural factors or identities, in images

or speech present significant challenges, because the predicting or securing annotations regarding religious information, can be exceptionally difficult. Thirdly, as we prioritize the "equal representation", our retrieval results sacrifice recall performance to ensure a retrieval bag that contains equal representations of each demographic group. This compromises fairness towards content providers in the image retrieval process. From the standpoint of content providers, fairness should imply that similar samples are treated similarly, regardless of the demographic group membership of their provided samples. Lastly, our work assumes all queries are neutral. We do not develop a technique to identify if a query is neutral or not, thereby limiting the applicability of our method in hybrid text query retrieval where text query can be biased like in "Male doctor". It is important to note, however, that the above challenges are not unique to our method but are a common issue encountered in other debiasing approaches. A more comprehensive discussion on the limitations of our work is available in the SM.

5. Final Remarks

In this dissertation, I focused on advancing vision intelligence via developing efficient, interpretable and fair deep learning models. Each chapter uniquely contributes to one of these critical aspects.

Chapter 2 presented the Zoom-In network, a groundbreaking approach for classifying extremely large images, notably reducing GPU memory usage while maintaining high accuracy. This achievement not only demonstrated my commitment to efficiency but also sets a new standard for processing large-scale image data.

In Chapter 3, the focus shifted to the interpretability of deep learning models in the digital marketing domain. My innovative multimodal neural network significantly improved prediction rates and provided actionable insights for enhancing digital marketing strategies. This chapter underlines the potential of using advanced neural models to provide clear, actionable guidance in various sectors, including healthcare and finance.

Chapter 4 addressed the critical issue of fairness in text-based image retrieval systems, specifically examining gender and racial biases. The post-hoc bias mitigation method introduced therein effectively reduces test-time bias while preserving retrieval accuracy. The principles established there extend beyond image retrieval, offering a versatile approach to enhancing fairness in a wide range of AI applications.

Overall, this dissertation significantly contributes to enhancing aspects of efficiency, interpretability, and fairness in deep learning models. Future work holds promising prospects, particularly in applying these approaches to current billion-scale vision-language models, such as GPT-4 (Achiam et al., 2023), BLIP-2 (Li et al., 2023), LLaVA (Liu et al., 2023) , to explore their scalability and impact in more complex real-world scenarios.

Bibliography

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023), "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*.

Ackerman, B. and Chen, Y. (2011), "Evaluating rank accuracy based on incomplete pairwise preferences," in *Proc. Workshop on UCERSTI Recsys*, vol. 11.

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018), "Sanity checks for saliency maps," *Advances in neural information processing systems*, 31.

Agarwal, C., Saxena, E., Krishna, S., Pawelczyk, M., Johnson, N., Puri, I., Zitnik, M., and Lakkaraju, H. (2022), "OpenXAI: Towards a Transparent Evaluation of Model Explanations," in *Conference on Neural Information Processing Systems (NeurIPS)*.

Alaluf, M., Crippa, G., Geng, S., Jing, Z., Krishnan, N., Kulkarni, S., Navarro, W., Sircar, R., and Tang, J. (2022), "Reinforcement Learning Paycheck Optimization for Multivariate Financial Goals," *Risk & Decision Analysis*.

Anas, M., Gupta, K., and Ahmad, S. (2017), "Skin cancer classification using K-means clustering," *International Journal of Technical Research and Applications*, 5, 62–65.

Ancona, M., Oztireli, C., and Gross, M. (2019), "Explaining deep neural networks with a polynomial time algorithm for shapley value approximation," in *International Conference on Machine Learning*, pp. 272–281, PMLR.

Anders, C. J., Weber, L., Neumann, D., Samek, W., Müller, K.-R., and Lapuschkin, S. (2022), "Finding and removing Clever Hans: using explanation methods to debug and improve deep models," *Information Fusion*, 77, 261–295.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016), "Machine bias," in *Ethics of data and analytics*, pp. 254–264, Auerbach Publications, Boca Raton, FL, USA.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. (2017), "A closer look at memorization in deep networks," in *International Conference on Machine Learning*, pp. 233–242, PMLR.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010), "How to explain individual classification decisions," *The Journal of Machine Learning Research*, 11, 1803–1831.

Bai, Y., Yu, W., Xiao, T., Xu, C., Yang, K., Ma, W.-Y., and Zhao, T. (2014), "Bag-of-words based deep neural network for image retrieval," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 229–232, New York, NY, USA, Association for Computing Machinery.

Ball, G. H. and Hall, D. J. (1965), "ISODATA, a novel method of data analysis and pattern classification," Tech. rep., Stanford research inst Menlo Park CA.

Bejnordi, B. E., Veta, M., Van Diest, P. J., Van Ginneken, B., Karssemeijer, N., Litjens, G., Van Der Laak, J. A., Hermsen, M., Manson, Q. F., Balkenhol, M., et al. (2017), "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *JAMA*, 318, 2199–2210.

Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009), "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*, pp. 1–4, Springer.

Berg, H., Hall, S. M., Bhalgat, Y., Yang, W., Kirk, H. R., Shtedritski, A., and Bain, M. (2022), "A prompt array keeps the bias away: Debiasing vision-language models with adversarial learning," *arXiv preprint arXiv:2203.11933*.

Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., Heldt, L., Zhao, Z., Hong, L., Chi, E. H., et al. (2019), "Fairness in recommendation ranking through pairwise comparisons," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2212–2220, New York, NY, USA, Association for Computing Machinery.

Biloš, A., Turkalj, D., and Kelić, I. (2016), "Open-rate controlled experiment in e-mail marketing campaigns," *Market-Tržište*, 28, 93–109.

Bissuel, A. (2020), "Why your A/B-test needs confidence intervals," `https://medium.com/criteo-engineering/why-your-ab-test-needs-confidence-intervals-bec9fe18db41`.

Biswas, A., Pham, T. T., Vogelsong, M., Snyder, B., and Nassif, H. (2019), "Seeker: Real-Time Interactive Search," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 2867–2875.

Bohle, M., Fritz, M., and Schiele, B. (2021), "Convolutional dynamic alignment networks for interpretable classifications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10029–10038.

Brendel, W. and Bethge, M. (2019), "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet," *arXiv preprint arXiv:1904.00760*.

Buolamwini, J. and Gebru, T. (2018), "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*, pp. 77–91, PMLR.

Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., and Guo, D. (2017), "Real-time bidding by reinforcement learning in display advertising," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 661–670.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017), "Semantics derived automatically from language corpora contain human-like biases," *Science*, 356, 183–186.

Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., and Varshney, K. R. (2017), "Optimized Pre-Processing for Discrimination Prevention," in *Advances in Neural Information Processing Systems*, eds. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, vol. 30, Curran Associates, Inc.

Campanella, G., Hanna, M. G., Geneslaw, L., Miraflor, A., Silva, V. W. K., Busam, K. J., Brogi, E., Reuter, V. E., Klimstra, D. S., and Fuchs, T. J. (2019), "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images," *Nature medicine*, 25, 1301–1309.

Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W., 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*, pp. 2956–2964.

Cao, M., Li, S., Li, J., Nie, L., and Zhang, M. (2022), "Image-text retrieval: A survey on recent research and development," *arXiv preprint arXiv:2203.14713*.

Celis, L. E. and Keswani, V. (2020), "Implicit diversity in image summarization," *Proceedings of the ACM on Human-Computer Interaction*, 4, 1–28.

Chang, C.-H., Creager, E., Goldenberg, A., and Duvenaud, D. (2019), "Explaining image classifiers by counterfactual generation," in *International Conference on Learning Representations (ICLR)*.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., and Su, J. K. (2019), "This looks like that: deep learning for interpretable image recognition," *Advances in neural information processing systems*, 32.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020), "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015), "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*.

Cheng, P., Hao, W., Yuan, S., Si, S., and Carin, L. (2021), "Fairfil: Contrastive neural debiasing method for pretrained text encoders," *arXiv preprint arXiv:2103.06413*.

Christie, G., Fendley, N., Wilson, J., and Mukherjee, R. (2018), "Functional map of the world," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6172–6180.

Chuang, C.-Y., Jampani, V., Li, Y., Torralba, A., and Jegelka, S. (2023), "Debiasing vision-language models via biased prompts," *arXiv preprint arXiv:2302.00070*.

Clark, R. G. (2009), "Sampling of subpopulations in two-stage surveys," *Statistics in Medicine*, 28, 3697–3717.

Cordonnier, J.-B., Mahendran, A., Dosovitskiy, A., Weissenborn, D., Uszkoreit, J., and Unterthiner, T. (2021), "Differentiable Patch Selection for Image Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2351–2360.

Cotter, A., Jiang, H., Gupta, M. R., Wang, S., Narayan, T., You, S., and Sridharan, K. (2019), "Optimization with Non-Differentiable Constraints with Applications to Fairness, Recall, Churn, and Other Goals." *J. Mach. Learn. Res.*, 20, 1–59.

Courtiol, P., Tramel, E. W., Sanselme, M., and Wainrib, G. (2018), "Classification and disease localization in histopathology using only global labels: A weakly-supervised approach," *arXiv preprint arXiv:1802.02212*.

Denil, M., Bazzani, L., Larochelle, H., and de Freitas, N. (2012), "Learning where to attend with deep architectures for image tracking," *Neural computation*, 24, 2151–2184.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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)*, pp. 4171–4186.

Dong, X., Huang, J., Yang, Y., and Yan, S. (2017), "More is less: A more complicated network with less inference complexity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5840–5848.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., and Unterthiner, T. (2020), "Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*.

Dutra, I., Nassif, H., Page, D., Shavlik, J., Strigel, R. M., Wu, Y., Elezaby, M. E., and Burnside, E. (2011), "Integrating Machine Learning and Physician Knowledge to Improve the Accuracy of Breast Biopsy," in *American Medical Informatics Association Symposium (AMIA)*, pp. 349–355.

Edwards, H. and Storkey, A. (2015), "Censoring representations with an adversary," *arXiv preprint arXiv:1511.05897*.

Efraimidis, P. S. and Spirakis, P. G. (2006), "Weighted random sampling with a reservoir," *Information Processing Letters*, 97, 181–185.

Efron, B. (2020), "Prediction, estimation, and attribution," *International Statistical Review*, 88, S28–S59.

Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. (2017), "Vse++: Improving visual-semantic embeddings with hard negatives," *arXiv preprint arXiv:1707.05612*.

Fiez, T., Gamez, S., Chen, A., Nassif, H., and Jain, L. (2022), "Adaptive Experimental Design and Counterfactual Inference," in *Workshops of Conference on Recommender Systems (RecSys)*.

Fong, C.-M., Wang, H.-W., Kuo, C.-H., and Hsieh, P.-C. (2019), "Image quality assessment for advertising applications based on neural network," *Journal of Visual Communication and Image Representation*, 63, 102593.

Fong, R. C. and Vedaldi, A. (2017), "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437.

Friedler, S., Scheidegger, C., and Venkatasubramanian, S. (2014), "Certifying and removing disparate impact," *arXiv preprint arXiv:1412.3756*.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013), "Devise: A deep visual-semantic embedding model," *Advances in neural information processing systems*, 26.

Fu, J., Zheng, H., and Mei, T. (2017), "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4438–4446.

Fukui, H., Hirakawa, T., Yamashita, T., and Fujiyoshi, H. (2019), "Attention branch network: Learning of attention mechanism for visual explanation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10705–10714.

Gadzicki, K., Khamsehashari, R., and Zetzsche, C. (2020), "Early vs late fusion in multimodal convolutional neural networks," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–6, IEEE.

Galway, L. P., Bell, N., Al Shatari, S. A., Hagopian, A., Burnham, G., Flaxman, A., Weiss, W. M., Rajaratnam, J., and Takaro, T. K. (2012), "A two-stage cluster sampling method using gridded population data, a GIS, and Google Earth TM imagery in a population-based mortality survey in Iraq," *International Journal of Health Geographics*, 11, 1–9.

Geng, S., Kuang, Z., Peissig, P., and Page, D. (2018), "Temporal poisson square root graphical models," *Proceedings of machine learning research*, 80, 1714.

Geng, S., Nassif, H., Manzanares, C. A., Reppen, A. M., and Sircar, R. (2020), "Deep PQR: Solving Inverse Reinforcement Learning using Anchor Actions," in *International Conference on Machine Learning (ICML)*, pp. 3431–3441.

Geng, S., Nassif, H., and Manzanares, C. A. (2023), "A Data-Driven State Aggregation Approach for Dynamic Discrete Choice Models," in *Uncertainty in Artificial Intelligence (UAI)*.

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., and Lee, S. (2019), "Counterfactual visual explanations," in *International Conference on Machine Learning*, pp. 2376–2384, PMLR.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020), "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, 37, 362–386.

Grislain, N., Perrin, N., and Thabault, A. (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*, pp. 2801–2809.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018), "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, 51, 1–42.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. (2020), "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 43, 4338–4364.

Hägele, M., Seegerer, P., Lapuschkin, S., Bockmayr, M., Samek, W., Klauschen, F., Müller, K.-R., and Binder, A. (2020), "Resolving challenges in deep learning-based analyses of histopathological images using explanation methods," *Scientific reports*, 10, 1–12.

Hall, M., Gustafson, L., Adcock, A., Misra, I., and Ross, C. (2023), "Vision-Language Models Performing Zero-Shot Tasks Exhibit Gender-based Disparities," *arXiv preprint arXiv:2301.11100*.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. (2018), "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *arXiv preprint arXiv:1804.06872*.

Handler, A., Denny, M., Wallach, H., and O'Connor, B. (2016), "Bag of what? simple noun phrase extraction for text analysis," in *Proceedings of the First Workshop on NLP and Computational Social Science*, pp. 114–124.

Hardt, M., Price, E., and Srebro, N. (2016), "Equality of opportunity in supervised learning," *Advances in neural information processing systems*, 29.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a), "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

He, R., Fang, C., Wang, Z., and McAuley, J. (2016b), "Vista: A visually, socially, and temporally-aware model for artistic recommendation," in *Proceedings of the 10th ACM conference on recommender systems*, pp. 309–316.

Ilse, M., Tomczak, J., and Welling, M. (2018), "Attention-based deep multiple instance learning," in *International Conference on Machine Learning*, pp. 2127–2136, PMLR.

Itti, L., Koch, C., and Niebur, E. (1998), "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 1254–1259.

Jaderberg, M., Vedaldi, A., and Zisserman, A. (2014), "Speeding up convolutional neural networks with low rank expansions," *arXiv preprint arXiv:1405.3866*.

Jalal, A., Karmalkar, S., Hoffmann, J., Dimakis, A., and Price, E. (2021), "Fairness for image generation with uncertain sensitive attributes," in *International Conference on Machine Learning*, pp. 4721–4732, PMLR.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. (2017), "Three factors influencing minima in SGD," *arXiv preprint arXiv:1711.04623*.

Jesson, A., Mindermann, S., Shalit, U., and Gal, Y. (2020), "Identifying causal-effect inference failure with uncertainty-aware models," *Advances in Neural Information Processing Systems*, 33, 11637–11649.

Katharopoulos, A. and Fleuret, F. (2019), "Processing megapixel images with deep attention-sampling models," in *International Conference on Machine Learning*, pp. 3282–3291, PMLR.

Kay, M., Matuszek, C., and Munson, S. A. (2015), "Unequal representation and gender stereotypes in image search results for occupations," in *Proceedings of the 33rd annual acm conference on human factors in computing systems*, pp. 3819–3828.

Keras (2022), "Global Average Pooling 2D," `https://www.tensorflow.org/api_docs/python/tf/keras/layers/GlobalAveragePooling2D`.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020), "Supervised contrastive learning," *arXiv preprint arXiv:2004.11362*.

Ki, M., Uh, Y., Lee, W., and Byun, H. (2020), "In-sample Contrastive Learning and Consistent Attention for Weakly Supervised Object Localization," in *Proceedings of the Asian Conference on Computer Vision*.

Kim, J. M., Koepke, A., Schmid, C., and Akata, Z. (2023), "Exposing and Mitigating Spurious Correlations for Cross-Modal Retrieval," *arXiv preprint arXiv:2304.03391*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023), "Segment anything," *arXiv preprint arXiv:2304.02643*.

Kohavi, R. and Longbotham, R. (2017), "Online Controlled Experiments and A/B Testing." *Encyclopedia of machine learning and data mining*, 7, 922–929.

Kolodner, J. L. (1992), "An introduction to case-based reasoning," *Artificial intelligence review*, 6, 3–34.

Kong, F. and Henao, R. (2022), "Efficient classification of very large images with tiny objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2384–2394.

Kong, F., Yuan, S., Hao, W., and Henao, R. (2023a), "Mitigating Test-Time Bias for Fair Image Retrieval," *arXiv preprint arXiv:2305.19329*.

Kong, F., Li, Y., Nassif, H., Fiez, T., Henao, R., and Chakrabarti, S. (2023b), "Neural insights for digital marketing content design," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4320–4332.

Kong, Y. and Fu, Y. (2022), "Human action recognition and prediction: A survey," *International Journal of Computer Vision*, 130, 1366–1401.

Kool, W., Van Hoof, H., and Welling, M. (2019), "Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement," in *International Conference on Machine Learning*, pp. 3499–3508, PMLR.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012), "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, 25, 1097–1105.

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017), "Counterfactual fairness," *Advances in neural information processing systems*, 30.

Langlois, T., Zhao, H., Grant, E., Dasgupta, I., Griffiths, T., and Jacoby, N. (2021), "Passive attention in artificial neural networks predicts human visual selectivity," *Advances in Neural Information Processing Systems*, 34, 27094–27106.

Larsson, F. and Felsberg, M. (2011), "Using Fourier descriptors and spatial models for traffic sign recognition," in *Scandinavian conference on image analysis*, pp. 238–249, Springer.

Larsson, G., Maire, M., and Shakhnarovich, G. (2017), "Ultra-deep neural networks without residuals," in *Int. Conf. on Learning Representations, arXiv, Toulon, France*, p. 1605.

Lee, B. and Paeng, K. (2018), "A robust and effective approach towards accurate metastasis detection and pn-stage classification in breast cancer," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 841–850, Springer.

Lee, J., Jukarainen, S., Dixon, P., Davies, N. M., Smith, G. D., Natarajan, P., and Ganna, A. (2022), "Quantifying the causal impact of biological risk factors on healthcare costs," *medRxiv*, pp. 2022–11.

Lee, K.-H., Chen, X., Hua, G., Hu, H., and He, X. (2018), "Stacked cross attention for image-text matching," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 201–216.

Lee, N., Bang, Y., Lovenia, H., Cahyawijaya, S., Dai, W., and Fung, P. (2023), "Survey of Social Bias in Vision-Language Models," *arXiv preprint arXiv:2309.14381*.

Levi, H. and Ullman, S. (2018), "Efficient coarse-to-fine non-local module for the detection of small objects," *arXiv preprint arXiv:1811.12152*.

Lew, M. S., Sebe, N., Djeraba, C., and Jain, R. (2006), "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2, 1–19.

Li, C., Liu, H., Li, L., Zhang, P., Aneja, J., Yang, J., Jin, P., Hu, H., Liu, Z., Lee, Y. J., et al. (2022a), "Elevater: A benchmark and toolkit for evaluating language-augmented visual models," *Advances in Neural Information Processing Systems*, 35, 9287–9301.

Li, J., Li, W., Sisk, A., Ye, H., Wallace, W. D., Speier, W., and Arnold, C. W. (2021), "A multi-resolution model for histopathology image classification and localization with multiple instance learning," *Computers in Biology and Medicine*, 131, 104253.

Li, J., Li, D., Savarese, S., and Hoi, S. (2023), "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*.

Li, O., Liu, H., Chen, C., and Rudin, C. (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.

Li, Z., Ratliff, L., Nassif, H., Jamieson, K., and Jain, L. (2022b), "Instance-Optimal PAC Algorithms for Contextual Bandits," in *Conference on Neural Information Processing Systems (NeurIPS)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014), "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer.

Litjens, G., Bandi, P., Ehteshami Bejnordi, B., Geessink, O., Balkenhol, M., Bult, P., Halilovic, A., Hermsen, M., van de Loo, R., Vogels, R., et al. (2018), "1399 H&E-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset," *GigaScience*, 7, giy065.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. (2023), "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*.

Lu, M. Y., Williamson, D. F., Chen, T. Y., Chen, R. J., Barbieri, M., and Mahmood, F. (2021), "Data-efficient and weakly supervised computational pathology on whole-slide images," *Nature Biomedical Engineering*, pp. 1–16.

Lundberg, S. M. and Lee, S.-I. (2017), "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, 30.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. (2018), "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European Conference on Computer Vision*, pp. 181–196.

Marra, F., Gragnaniello, D., Verdoliva, L., and Poggi, G. (2020), "A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection," *IEEE Access*, 8, 133488–133502.

Mehrotra, A. and Celis, L. E. (2021), "Mitigating bias in set selection with noisy protected attributes," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 237–248.

Mehrotra, A. and Vishnoi, N. (2022), "Fair Ranking with Noisy Protected Attributes," *Advances in Neural Information Processing Systems*, 35, 31711–31725.

Merrick, L. (2019), "Randomized ablation feature importance," *arXiv preprint arXiv:1910.00174*.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2021), "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 44, 3523–3542.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016), "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, pp. 1928–1937, PMLR.

Morik, M., Singh, A., Hong, J., and Joachims, T. (2020), "Controlling fairness and bias in dynamic learning-to-rank," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 429–438.

Mukhoti, J., Lin, T.-Y., Poursaeed, O., Wang, R., Shah, A., Torr, P. H., and Lim, S.-N. (2022), "Open Vocabulary Semantic Segmentation with Patch Aligned Contrastive Learning," *arXiv preprint arXiv:2212.04994*.

Nabi, S., Nassif, H., Hong, J., Mamani, H., and Imbens, G. (2022), "Bayesian Meta-Prior Learning Using Empirical Bayes," *Management Science*, 68, 1737–1755.

Naik, N., Madani, A., Esteva, A., Keskar, N. S., Press, M. F., Ruderman, D., Agus, D. B., and Socher, R. (2020), "Deep learning-enabled breast cancer hormonal receptor status determination from base-level H&E stains," *Nature Communications*, 11, 1–8.

Niu, W., Caverlee, J., and Lu, H. (2018), "Neural personalized ranking for image recommendation," in *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 423–431.

Noble, S. U. (2018), "Algorithms of oppression," in *Algorithms of oppression*, New York University Press.

Nuruzzaman, M. and Hussain, O. K. (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)*, pp. 54–61, IEEE.

OpenAI (2023), "GPT-4 Technical Report," *ArXiv*, abs/2303.08774.

Organization, I. L. (2019), "A quantum leap for gender equality: For a better future of work for all," .

Paleyes, A., Urma, R.-G., and Lawrence, N. D. (2022), "Challenges in deploying machine learning: a survey of case studies," *ACM Computing Surveys*, 55, 1–29.

Papadopoulos, A., Korus, P., and Memon, N. (2021), "Hard-Attention for Scalable Image Classification," *arXiv preprint arXiv:2102.10212*.

Parraga, O., More, M. D., Oliveira, C. M., Gavenski, N. S., Kupssinskü, L. S., Medronha, A., Moura, L. V., Simões, G. S., and Barros, R. C. (2022), "Debiasing Methods for Fairer Neural Models in Vision and Language Research: A Survey," *arXiv preprint arXiv:2211.05617*.

Pawlowski, N., Bhooshan, S., Ballas, N., Ciompi, F., Glocker, B., and Drozdzal, M. (2019), "Needles in haystacks: On classifying tiny objects in large images," *arXiv preprint arXiv:1908.06037*.

Pinckaers, H., van Ginneken, B., and Litjens, G. (2019), "Streaming convolutional neural networks for end-to-end learning with multi-megapixel images," *arXiv preprint arXiv:1911.04432*.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015), "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021), "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR.

Ramapuram, J., Diephuis, M., Lavda, F., Webb, R., and Kalousis, A. (2018), "Variational saccading: Efficient inference for large resolution images," *arXiv preprint arXiv:1812.03170*.

Ranjan, S., Nayak, D. R., Kumar, K. S., Dash, R., and Majhi, B. (2017), "Hyperspectral image classification: A k-means clustering based approach," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1–7, IEEE.

Rao, S., Böhle, M., and Schiele, B. (2022), "Towards better understanding attribution methods," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10223–10232.

Rhu, M., Gimelshein, N., Clemons, J., Zulfiqar, A., and Keckler, S. W. (2016), "vDNN: Virtualized deep neural networks for scalable, memory-efficient neural network design," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1–13, IEEE.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985), "Learning internal representations by error propagation," Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science.

Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., and Sejdinovic, D. (2019), "Detecting and quantifying causal associations in large nonlinear time series datasets," *Science advances*, 5, eaau4996.

Sawant, N., Namballa, C. B., Sadagopan, N., and Nassif, H. (2018), "Contextual Multi-Armed Bandits for Causal Marketing," in *Workshops of International Conference on Machine Learning (ICML)*.

Saxena, N. A., Huang, K., DeFilippis, E., Radanovic, G., Parkes, D. C., and Liu, Y. (2020), "How do fairness definitions fare? Testing public attitudes towards three algorithmic definitions of fairness in loan allocations," *Artificial Intelligence*, 283, 103238.

Schmidt, R., Montani, S., Bellazzi, R., Portinale, L., and Gierl, L. (2001), "Cased-based reasoning for medical knowledge-based systems," *International Journal of Medical Informatics*, 64, 355–367.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017), "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Seymour, S. (2016), "Why Resonance Is Vital For Your Digital Marketing Success," https://www.linkedin.com/pulse/why-resonance-vital-your-digital-marketing-success-steve-seymour/.

Singh, A. and Joachims, T. (2017), "Equality of opportunity in rankings," in *Workshop on Prioritizing Online Content (WPOC) at NIPS*, vol. 31.

Singh, K., Ma, M., and Park, D.-W. (2003), "A Content-based Image Retrieval using FFT & Cosine Similarity Coefficient." in *SIP*, pp. 315–319.

Singh, S. P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., and Jain, S. (2017), "Machine translation using deep learning: An overview," in *2017 international conference on computer, communications and electronics (comptelix)*, pp. 162–167, IEEE.

Sinha, M., Healey, J., and Sengupta, T. (2020), "Designing with AI for digital marketing," in *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 65–70.

Sirinukunwattana, K., Raza, S. E. A., Tsang, Y.-W., Snead, D. R., Cree, I. A., and Rajpoot, N. M. (2016), "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE Transactions on Medical Imaging*, 35, 1196–1206.

Sulthana, A. R., Gupta, M., Subramanian, S., and Mirza, S. (2020), "Improvising the performance of image-based recommendation system using convolution neural networks and deep learning," *Soft Computing*, 24, 14531–14544.

Sundararajan, M., Taly, A., and Yan, Q. (2017), "Axiomatic attribution for deep networks," in *International conference on machine learning*, pp. 3319–3328, PMLR.

Tan, M. and Le, Q. (2019), "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, pp. 6105–6114, PMLR.

Tellez, D., Litjens, G., van der Laak, J., and Ciompi, F. (2019), "Neural image compression for gigapixel histopathology image analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Thorndike, R. L. (1953), "Who belongs in the family," in *Psychometrika*, Citeseer.

Touvron, H., Vedaldi, A., Douze, M., and Jégou, H. (2019), "Fixing the train-test resolution discrepancy," *arXiv preprint arXiv:1906.06423*.

Uzkent, B. and Ermon, S. (2020), "Learning when and where to zoom with deep reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12345–12354.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017), "Attention is all you need," in *Advances in neural information processing systems*, pp. 6000—-6010.

Wang, J., Liu, Y., and Wang, X. E. (2021a), "Are gender-neutral queries really gender-neutral? mitigating gender bias in image search," *arXiv preprint arXiv:2109.05433*.

Wang, J., Liu, Y., and Wang, X. E. (2021b), "Assessing multilingual fairness in pre-trained multimodal representations," *arXiv preprint arXiv:2106.06683*.

Wang, J., Zhang, Y., and Sang, J. (2022a), "FairCLIP: Social Bias Elimination based on Attribute Prototype Learning and Representation Neutralization," *arXiv preprint arXiv:2210.14562*.

Wang, L., Wu, Z., Karanam, S., Peng, K.-C., Singh, R. V., Liu, B., and Metaxas, D. (2019), "Sharpen Focus: Learning With Attention Separability and Consistency," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 512–521, IEEE Computer Society.

Wang, R., Yu, T., Zhao, H., Kim, S., Mitra, S., Zhang, R., and Henao, R. (2022b), "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)*, pp. 571–582.

Wang, R., Cheng, P., and Henao, R. (2023a), "Toward Fairness in Text Generation via Mutual Information Minimization based on Importance Sampling," *arXiv preprint arXiv:2302.13136*.

Wang, T., Brovman, Y. M., and Madhvanath, S. (2021c), "Personalized embedding-based e-commerce recommendations at eBay," *arXiv preprint arXiv:2102.06156*.

Wang, W. (2022), "Data Marketing Optimization Method Combining Deep Neural Network and Evolutionary Algorithm," *Wireless Communications and Mobile Computing*, 2022.

Wang, Y., Han, Y., Wang, C., Song, S., Tian, Q., and Huang, G. (2023b), "Computation-efficient Deep Learning for Computer Vision: A Survey," *arXiv preprint arXiv:2308.13998*.

Wang, Z., Yin, Y., Shi, J., Fang, W., Li, H., and Wang, X. (2017), "Zoom-in-net: Deep mining lesions for diabetic retinopathy detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 267–275, Springer.

Xia, W., Ma, C., Liu, J., Liu, S., Chen, F., Yang, Z., and Duan, J. (2019), "High-Resolution Remote Sensing Imagery Classification of Imbalanced Data Using Multistage Sampling Method and Deep Neural Networks," *Remote Sensing*, 11, 2523.

Xu, H., Liu, X., Li, Y., Jain, A., and Tang, J. (2021), "To be robust or to be fair: Towards fairness in adversarial training," in *International Conference on Machine Learning*, pp. 11492–11501, PMLR.

Yu, R., Li, A., Chen, C.-F., Lai, J.-H., Morariu, V. I., Han, X., Gao, M., Lin, C.-Y., and Davis, L. S. (2018), "NISP: Pruning networks using neuron importance score propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9194–9203.

Zafar, M. B., Valera, I., Rodriguez, M. G., and Gummadi, K. P. (2015), "Learning fair classifiers," *arXiv preprint arXiv:1507.05259*, 1.

Zaidi, S. A. J., Buriro, A., Riaz, M., Mahboob, A., and Riaz, M. N. (2019), "Implementation and comparison of text-based image retrieval schemes," *International Journal of Advanced Computer Science and Applications*, 10, 611–618.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016), "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*.

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

Zhang, Y., Tiňo, P., Leonardis, A., and Tang, K. (2021), "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zhao, D., Wang, A., and Russakovsky, O. (2021), "Understanding and evaluating racial biases in image captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14830–14840.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2017), "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," *arXiv preprint arXiv:1707.09457*.

Zheng, H., Fu, J., Zha, Z.-J., and Luo, J. (2019), "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5012–5021.

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

Zhu, X., Yao, J., and Huang, J. (2016), "Deep convolutional neural network for survival analysis with pathological images," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 544–547, IEEE.

Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. (2023), "Object detection in 20 years: A survey," *Proceedings of the IEEE*.

Biography

Fanjie Kong is currently a Ph.D. candidate in the Electrical and Computer Engineering(ECE) Department at Duke University, working with Prof. Ricardo Henao. His primary research focus includes open-vocabulary object detection, the development of efficient vision models, and ensuring fairness in AI. His work in these domains underscores a commitment to tackling key challenges in vision intelligence regarding to efficiency, interpretability, and fairness.

During his PhD study, he has published multiple first-authored paper in the top tier venues, such as NeurIPS, CVPR, KDD, *etc.* He also served as reviewer for top AI conference such as EMNLP, CVPR, NeurIPS, *etc.*

Beyond academics, he demonstrated his skills and knowledge in industry as an Applied Scientist Intern at Amazon, where he focused on AI for digital marketing in 2021 and 2022. In 2023, he further interned at the Amazon AWS AI Lab as an Applied Scientist, where he worked on open-world object detection with vision-language foundation models.