


Massachusetts Institute of Technology
École Polytechnique Fédérale de Lausanne

Promoting Fairness in Computer Vision: A Study on the Effectiveness of Concept Bottleneck Models

Vincent Yuan

Master Thesis

Approved by the Examining Committee:

Schrasing Tong
Thesis Supervisor

Prof. Martin Jaggi
Principal Investigator, EPFL

Prof. Lalana Kagal
Principal Investigator, MIT

Decentralized Information Group
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
The Stata Center 5th floor room 32G-524
32 Vassar Street, Cambridge MA 02139

September 1, 2023

Science sans conscience n'est que ruine de l'âme
— François Rabelais

Dedicated to my parents, who left their home country to provide a better life to their children.

Abstract

With the growing responsibilities entrusted to Artificial Intelligence (AI) and Machine Learning (ML) systems, there is an increasing concern about these systems perpetuating and amplifying existing societal biases, such as gender bias. The Computer Vision field (CV) presents a unique set of challenges to achieve unbiased models due to the complexity of the image input, particularly in the field of visual recognition, consisting in identifying objects or activities in images. While existing solutions achieve some fairness guarantees for visual recognition, they often lack interpretability, a critical aspect of understanding the decision-making process of models in CV applications. This thesis aims to address this gap by exploring the effectiveness of ML Fairness of Concept Bottleneck Models (CBM), a novel type of model designed to map input images to an interpretable concept layer. By associating images with a set of gender-independent concepts, CBM can potentially eliminate gender-related biases from CV models. We conducted extensive experiments on three distinct classification cases: binary classification with the Doctor-Nurse dataset, multi-class classification with the imSitu dataset, and multi-label classification with the MS-COCO dataset. We created a new metric to assess fairness in multi-label classification. We manipulated the datasets by introducing different biases and found that CBM improve fairness while retaining or increasing performance compared to traditional models. Additionally, we studied the model's inner functioning, leveraging CBM interpretability to identify the steps at which the training set biases propagate to the model, the concept relevance, and the ideal density of the last layer in terms of performance and interpretability. Our work contributes to developing more responsible AI systems for Computer Vision by making models more interpretable, less biased, and more accurate, paving the way for a new area of unbiased Computer Vision models through Concept Bottleneck Models.

Acknowledgments

This Thesis concludes the end of my M.Sc. degree at EPFL, since the beginning of my B.Sc. degree 6 years ago. I've been extremely fortunate to meet multiple people who have shaped me into the person I am today through countless discussions, whom I would like to thank here.

I was lucky to complete my 6-month Thesis within the Decentralized Information Group at the Massachusetts Institute of Technology. Ph.D. student Schrasing Tong guided me from the start, supported me throughout the Thesis, and helped me countless times. His entrepreneurial aspirations greatly inspired me. I would also like to thank Prof. Lalana Kagal for her advice during the Thesis and for accepting me into the Decentralized Information Group. Her dedication to the ethical use of technology to address social issues greatly inspires me. Special thanks to Hengzhi Li for his collaboration towards the end of my Thesis and for our late-night work session together across two time zones.

I sincerely thank the Machine Learning & Optimization Laboratory of Prof. Jaggi and its subgroup intelligent Global Health, led by Prof. Mary-Anne Hartley (Annie), where I did all of my research at EPFL. I want to express my profound appreciation to Prof. Martin Jaggi for his supervision of my different research projects, including this Thesis, and for introducing me to the beautiful world of Machine Learning. I also want to express my sincere gratitude to Prof. Mary-Anne Hartley, who always inspired me with the impact of her projects. Finally, I thank Ph.D. student Lie He for supervising my two research projects at EPFL.

To all my friends who have been a constant source of support and encouragement, thank you for being there. I've had the chance to meet so many incredible people throughout the past six years. Although I cannot name everyone, I would like to acknowledge Louis, Oscar, Olivier, Maxime, and Kenyu for their exceptional support. Special thanks to Louis and Oscar for proofreading this Thesis.

Finally, I would like to thank my family for their unwavering love and support. To my parents, who have dedicated their lives to providing better opportunities for their children. They succeeded in passing the Gaokao, the Chinese national exam, in 1977, overcoming an 11-year-long ban, a pool of 5.8 million candidates, and an admission rate of just 4.8% to secure a university place. My parents were then among the 121 students selected to pursue a Ph.D. in France, where they settled to provide a better life for their children than the one they had. To my sisters, Elodie and Justine, who have always been my inspiration and guided me throughout my personal journey. And to my nephews, Nathan and Robin, for whom I am motivated to create a better future.

Cambridge, September 1, 2023 Vincent Yuan

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Machine Learning (ML) domain has achieved remarkable success and growth in recent years. The general audience is now knowledgeable about ML through its widespread application, while ML models are progressively deployed across all industries, automating manual work. As these models are increasingly deployed in critical and sensitive areas, there is a growing concern about their intrinsic biases. Studies found that some facial recognition systems misidentified African and East Asian faces 10 to 100 times more than Caucasian faces [23], that models used to identify and help patients with complex health needs are less likely to refer black people than white people with similar health conditions [36], or that Google ad-targeting system showed ads for high-income jobs to men much more often than to women [15]. The models often learn from biased data, leading to biased predictions that can have significant societal implications. Specifically, some applications, such as autonomous vehicles and medical diagnostics, are critical in computer vision, showing the need for models to be interpretable and as unbiased as possible.

The primary challenge in fairness in computer vision lies in the intrinsic complexity of the input. Each image is composed of many pixels, resulting in a large input feature space, making the internal functioning of computer vision models hard to interpret. This complexity gives rise to several challenges. Firstly, biases in the training data are not always apparent and can be easily overlooked by humans and inadvertently learned by the models. Secondly, the complexity of the input feature space makes it difficult to understand how the model is making decisions, which is critical for identifying and correcting biases. Lastly, the model's predictions can be influenced by confounding variables in the image that correlate with protected characteristics (e.g., gender, race, etc..), making it difficult to disentangle the model's decision-making process and ensure that it is learning meaningful representations unrelated to these characteristics during training.

Most existing research in computer vision fairness was centered around images where humans are the primary focus, such as in facial recognition [8]. Researchers can leverage standardized image formats (e.g., mugshots) to develop and test methods with a fair representation of the attributes

[41]. Significantly less research addresses fairness in scenarios where humans are present but not the main subject. This disparity is due to the complexity associated with the diversity of inputs in tasks like visual recognition, making it a more challenging area to research. Some previous works on fairness in visual recognition have debiased models by adding fairness constraints or using adversarial learning [50, 57]. While these methods achieved fairness objectives on the research datasets imSitu [54] and MS-COCO [32], they did not explain the underlying changes in the model's internal representation, making the results harder to interpret. Additionally, these methods require gender annotations in their data and have only used the original train-validation-test splits of MS-COCO and imSitu. The original splits have significant limitations, as they were not partitioned with respect to gender. This raises concerns about the validity of their results, as the classes can have different biases in their training and testing sets. It also limits the generalizability of these methods as they have not been tested on other scenarios with different biases. This raises questions about the robustness and adaptability of these approaches in real-world scenarios, and highlights that these methods needs gender-annotated data, which limits their generalizability.

In this thesis, we propose the use of Concept Bottleneck Models (CBM) as a way to address these challenges. CBM, introduced in 2020 by Koh et al.[31], are a new type of computer vision model with a penultimate concept layer, wherein each neuron is associated with a particular concept. This design is inherently interpretable, as it facilitates understanding the decision-making process through the lens of human-understandable concepts. CBM require finely annotated datasets to train the concept layer, which limits their use to specific datasets. Recently, researchers [37] have enabled the use of CBM on any classification dataset, given a model trained on this dataset. Label-Free CBM (LF-CBM) use GPT to generate concepts, CLIP to train the projection from the backbone model to the concept layer, and then train a final sparse classification layer. LF-CBM allow us to test CBM on datasets not finely annotated, presenting a potential solution to the previously mentioned challenge.

We believe that CBM could contribute to debiasing models for several reasons. First, mapping an image to concepts unrelated to gender should, in theory, remove biases in the model's internal representation of the images. For example, concepts such as 'a pan', 'indoor', or 'a stove', which are unrelated to gender, form a context from which the target is predicted , instead of directly extracting information from the person in the picture. Second, the interpretability of CBM enables us to understand the model's representation by examining the weights of the last sparse layer from the concept to the class, thereby providing insights into the model's inner workings. This is particularly important for addressing biases because it allows us to identify and correct any unfair associations or predictions made by the model. We can analyze how different scenarios, such as varying biases in the training set or the inclusion of gender as a concept, explicitly affect classification by examining model variations. Finally, CBM disentangles a complex input and maps it to a set of interpretable concepts akin to a tabular dataset scenario. Combining LF-CBM with gender prediction using CLIP can potentially lead to the development of an unsupervised visual recognition debiasing framework. Additionally, the fact that LF-CBM does not require gender information makes this technique possible for any dataset, thereby enhancing its potential for widespread applicability.

This thesis aims to rigorously assess the potential of Concept Bottleneck Models for enhancing fairness in computer vision models by evaluating how transitioning from traditional models to CBM impact performance, fairness and interpretability.

Our research reveals that CBM tend to reduce model bias while maintaining or improving performance metrics. Remarkably, this is achieved without imposing any fairness constraints but merely by transitioning from traditional models to CBM. The reliability of our results is reinforced by comprehensive evaluations undertaken on three unique datasets and different classification scenarios. Additionally, we utilized CBM interpretability to acquire a deeper understanding of the inner workings of the model and at which step of LF-CBM the training set biases propagate to the model.

The core contributions of this thesis are as follows:

1. We developed and applied a comprehensive evaluation framework that assesses a model on fairness and performance across different bias settings. We used this framework to reveal the potential of CBM in reducing bias while maintaining or even improving model performance compared to traditional models.
2. We applied this framework to three different classification tasks, each associated with a different dataset, thereby ensuring the robustness of our approach across different scenarios. We developed a data processing pipeline for the three datasets, enabling the manipulation of different dataset biases.
3. We utilized the interpretability of CBM to understand how LF-CBM associated concepts with classes in the presence or absence of biases in the training data and examined at which steps of the LF-CBM framework the training dataset biases propagate to the model.
4. We developed a novel metric to quantify fairness in multi-label classification tasks.

In Chapter 2, we will present an overview of the background of Bias and Fairness in Machine Learning and Visual Recognition, as well as a detailed explanation of CBM. In Chapter 3, we will detail the different datasets and provide a comprehensive explanation of our experimental design and methodology. In Chapter 4, we will present and analyze the results of our experiments, highlighting key findings and discussing about limitations and future work. In Chapter 5, we will briefly review related work in the domain, before concluding in Chapter 6.

The author acknowledges Hengzhi Li for his work on the implementation of the multi-label classification on MS-COCO, Prof. Mathias Payer for his Thesis template, and MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to the research results reported within this report. The code will be made available after publication at `https://github.com/Vinceyn/CV-Fairness`.

Chapter 2

Background

2.1 Bias and Fairness in Machine Learning

Machine Learning models seek to automate decision-making by leveraging vast amounts of data. While these algorithms can process more data and compute faster than humans, potentially removing human biases in decision-making, they are themselves not necessarily devoid of biases. A revealing example is the COMPAS software, used by US judges to evaluate the risk of recidivism, thereby helping judges to decide about prisoners' release. Notably, this tool demonstrated a significant bias against African Americans, showing a higher False Postive rate for this demographic, leading to inaccurate predictions of their likelihood to recommit a crime [5]. Likewise, studies have revealed that commercial facial recognition systems [42] and speech recognition technologies [30] underperform on black individuals compared to their white counterparts. Such discrepancies underscore the pressing need to address biases in machine learning systems to ensure fair and just outcomes.

Several studies have delved into the multifaceted origins of biases in machine learning, attempting to enumerate and understand every conceivable source [35] and conceiving solutions against these biases. We will explore these solutions in Related Work (see Section 5). One prominent source of bias is the training dataset of models. Biases in datasets can be inherited from historical patterns or emerge from inconsistent data collection practices, such as when certain demographic groups are over-sampled or under-sampled, especially when data from marginalized groups is underrepresented or omitted. For instance, historical biases are evident in Amazon's past recruiting algorithm, which was found to favor male candidates due to patterns in historical hiring data [14]. In another example, inconsistent data collection or lack of diverse data led some commercial gender recognition systems to underperform, particularly with dark-skinned women [8].

Research has demonstrated that machine learning models can not only inherit biases from

their training data but may also amplify or introduce new biases during the optimization phase [57]. These biases can manifest in various ways. For instance, when the loss function heavily fits classes with more samples, it can inadvertently skew the model predictions in favor of those classes, amplifying the distribution bias already present in the training data. Additionally, some regularization techniques, such as L1 or L2 regularization, might introduce biases by implicitly favoring specific features or groups over others [13].

In light of accumulating evidence of biases in machine learning models, the concept of fairness in machine learning has fostered significant attention in academic literature [39]. It encompasses efforts to define, evaluate, and enhance fairness within ML algorithms [47]. We note that the perception of what constitutes "fairness" can differ based on cultural, ethical, or domain-specific factors. The scientific community decided to formulate standardized fairness metrics, but the use or not of these metrics will be based on the abovementioned factors. We also note that there exists an inherent trade-off between most fairness definitions and accuracy: it is known as the fairness-accuracy trade-off.

Individual fairness asserts that similar individuals should receive similar outcomes from a decision-making algorithm. While individual fairness offers granular level equity, it may not always capture broader structural biases affecting entire groups. In contrast, this thesis will focus on group fairness, as individual fairness is less relevant in computer vision compared to tabular data. Under this paradigm, fairness is examined at the group level, aiming to treat distinct groups equitably. Particularly, underrepresented or historically marginalized groups are often designated as "protected" groups based on a sensitive attribute such as gender, race, or age.

To formalize the forthcoming discussion, we introduce some notation:

- S is the protected attribute, with $S = 1$ denoting the privileged group and $S \neq 1$ the underprivileged group.
- $\hat{Y} = 1$ indicates a positive classification rendered by the model.
- $Y = 1$ denotes an instance with a positive ground truth label.

With these notations in place, we will now inspect the predominant definitions of group fairness.

1. **Demographic Parity:** Demographic parity requires that positive predictions are equally distributed among different demographic groups. This is reminiscent of affirmative action policies that aim to ensure equal opportunities across different demographic groups [19].

$$|P(\hat{Y} = 1|S = 1)| = |P(\hat{Y} = 1|S = 0)| \tag{2.1}$$

This equality is often not the case. It is possible to compute a similar value, computing the difference between the right and left part of the equality. The closer this value is to zero, the better the demographic parity, indicating similar rates of positive predictions across groups. However, this metric can be problematic when the base rates differ between groups. Specifically, enforcing demographic parity will introduce bias if the actual rate of positive instances $Y = 1$ is different for $S = 1$ and $S = 0$. For example, if a specific demographic group has a higher rate of a particular disease, enforcing demographic parity in a medical diagnosis model could lead to incorrect diagnoses. This means that two individuals with similar qualifications could be treated differently solely based on their group membership

2. **Equalized Odds:** Equalized odds aim to ensure that a classifier exhibits similar accuracy regarding both true positives and false positives across different demographic groups. Introduced by Hardt et al. [24], this metric focuses on equating the False Postive rates (FPRs) and the True Positive rates (TPRs) between groups.

$$|P(\hat{Y} = 1|S = 1, Y = y)| = |P(\hat{Y} = 1|S \neq 1, Y = y)|, \ \forall y \in \{0, 1\} \tag{2.2}$$

This ensures that for both positive and negative true class labels Y, the predictions $\hat{Y}$ have similar distributions regardless of group membership. In other words, both the likelihood of correctly identifying a positive instance and of incorrectly classifying a negative instance as positive should be consistent across different groups. However, this metric also does not account for base rate differences, which may not be suitable for scenarios where the actual rates of positive and negative instances vary significantly between groups. It is also complex to extend this metric to multi-class classification.

3. **Equality of opportunity:** This criterion is particularly relevant in scenarios where it is essential to ensure that both privileged and underprivileged groups have equal chances of receiving positive outcomes when they are deserved (i.e., true positives).

$$|P(\hat{Y} = 1|S = 1, Y = 1)| = |P(\hat{Y} = 1|S \neq 1, Y = 1)| \tag{2.3}$$

Unlike equalized odds, which balance both false positives and true positives across groups, Equality of opportunity solely focuses on the latter. However, this focus on true positives could lead to a higher rate of false positives for one group than another, which may not be desirable in all contexts.

4. **Accuracy Parity:** One of the foundational goals in fairness is to ensure that the predictive accuracy of a model is consistent across different demographic or sensitive groups. The "Accuracy Parity" metric measures this consistency. It compares the accuracy of predictions for each group, aiming for a minimal difference between them:

$$\Delta_{Acc} = |Acc_{S=0} - Acc_{S=1}| \tag{2.4}$$

For an ideal scenario, $\Delta_{Acc} = 0$, meaning both groups experience the same accuracy. Accuracy parity is not always the most appropriate metric, especially when the underlying probability distribution of the true outcomes Y differs between groups or when the implications of false positives differ significantly from false negatives. Additionally, like the other metrics discussed, accuracy parity does not take into account the differences in base rates between groups, which can also lead to unfair outcomes.

5. **Fairness through awareness and unawareness**: Fairness through awareness involves actively using sensitive attributes to adjust for biases and ensure fair treatment across groups. On the other hand, unawareness involves deliberately ignoring the sensitive attributes to prevent the model from creating associations with them, even though some attributes can be correlated with the sensitive ones.

While the majority of fairness metrics have been designed for binary classification, there remains a lack of consensus on metrics for measuring fairness in multi-class classification, as research in this field is relatively recent [6, 16]. For multi-label classification, only a single paper has been published in this domain [33], which attempts to define some fairness metrics. However, this paper assumes the presence of a 'favorable set of outcomes' (e.g., 'being hired for a job'), which does not apply to our thesis. Throughout our thesis, we will employ a metric similar to Equality of Opportunity for binary classification, extend Accuracy Parity to a multi-class classification scenario, and define our own fairness metric for multi-label classification.

The research on Fairness in Machine Learning started to gain momentum in the early 2010s, parallel to the advances and democratization in Machine Learning. Much research has been done on Fairness for tabular datasets (see Section 5.1), and there is ongoing extensive research on Natural Language Processing (NLP) debiasing, especially on large language models, due to their widespread use in applications that affect people's lives. However, the research on Fairness in computer vision is less common due to the inherent complexity of images as input.

2.2 Bias and Fairness in Vision Recognition

Computer vision models are trained to perform classification tasks by extracting and internalizing patterns in the data. These models discern information at multiple levels of abstraction. For instance, at a lower level, they might recognize specific visual attributes, such as the azure hue of the sea. At a more contextual level, they might infer associations from broader cues, such as associating white coats with doctors. Computer vision models map input images to corresponding classes through this hierarchical extraction of information.

When computer vision systems are deployed on images featuring individuals, there is an inherent risk of introducing bias, especially concerning sensitive attributes such as gender or race. This is

because the model may inadvertently associate specific image components with these sensitive attributes in its attempt to find patterns, leading to biased outcomes [51]. A study by Zhao et al. [57] highlighted this issue, revealing gender biases in two widely used computer vision datasets for visual recognition: MS-COCO and imSitu. ImSitu [54], a dataset tailored for situation recognition, exhibited gendered distributions, such as a higher frequency of females cooking and males driving. Similarly, the MS-COCO dataset [32], designed for tasks like image recognition, segmentation, and captioning, was found to have a disproportionate number of images featuring male individuals in skating scenarios. These biases underscore the importance of critically examining training data in AI systems to ensure equitable outcomes.

Zhao et al.'s research underscores an alarming observation: models trained on these datasets not only adopt the existing biases but also amplify them. Their analysis across both datasets reveals that uncalibrated systems tend to exacerbate societal biases present in the training set. For example, there are twice as many pictures of females cooking than males cooking in the imSitu training set, but models trained predict cooking for women three times as often as for men. Further complicating matters, Wang et al. [50] highlight that even balanced datasets—where labels equally co-occur with each gender—are not immune to these challenges. Surprisingly, models trained on such datasets continue to learn associations between labels and gender, almost as if the data had not been balanced in the first place. Wang et al. propose a potential reason: models might be picking up on gender-correlated, yet unlabeled, variables that cannot be directly balanced. For instance, if there is a higher occurrence of children in scenes with cooking, and these scenes predominantly feature women, the model may inadvertently associate women with cooking through the presence of children.

Debiasing computer vision models is challenging due to the various potential sources of bias in an image. As highlighted in the Related Work section (see Section 5), there has been a significant increase in recent research on model interpretability in computer vision. However, there is still a lack of understanding of what information a computer vision model internalizes. That's why solutions from Zhao et al. and Wang et al. gravitate towards optimization methods for fairness instead of interpreting the model to remove biases. Specifically, Zhao et al. integrate constraints during optimization to ensure outputs align with a predefined distribution. This ensures that the model prediction distribution will be similar to the model training distribution. In contrast, Wang et al. employ an adversarial gender loss, forcing the model to ignore gender-coincident variables during its learning process. This approach indirectly forces the model to "blur out" the gender-specific characteristics of humans in the images (See Appendix C), thereby reducing gender bias in the model's predictions.

A significant gap exists in the literature regarding the debiasing of computer vision models for image recognition, specifically in tasks involving humans, through an explicit representation of concepts via feature disentanglement. Additionally, many existing debiasing methods in computer vision implicitly or explicitly assume prior knowledge of gender or other sensitive attributes. This assumption can be limiting, particularly in real-world scenarios where such information might not

always be available, accurate, or ethical to use. In such cases, models that rely on these assumptions can inadvertently amplify existing biases or introduce new ones.

2.3 Concept Bottleneck Models

Concept Bottleneck Models (CBM) are a new type of Computer Vision model developed by Koh et al. in 2020 [31]. These models enhance interpretability by incorporating an intermediate layer composed of neurons, each representing a human-understandable concept; this is referred to as the *concept layer*.

This section dives into the details of CBM, beginning with an examination of its foundational principles, implications, and limitations and then introducing the Label-Free Concept Bottleneck Model (LF-CBM) [37], a variant that we use during this thesis.

2.3.1 Concept Bottleneck Models: a Deep Dive

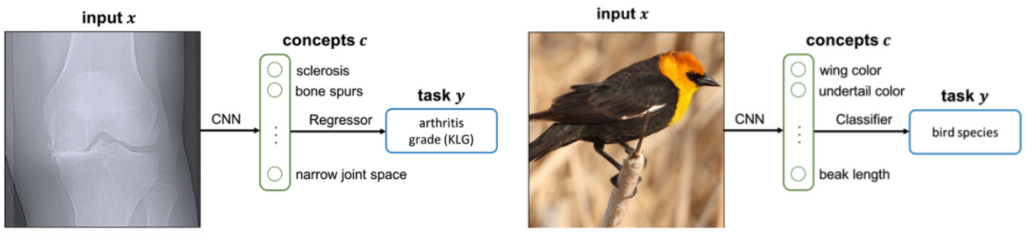


Figure 2.1: The Concept Bottleneck Model (CBM) workflow. First, the model predicts a set of human-specified concepts (c), then uses these concepts to classify the image into the final output (y), typically through a Logistic Regression.

Concept Bottleneck Models (CBM), developed by Koh et al. [31], constitute a specialized category of Computer Vision models intended for task classification. The *bottleneck layer* is the penultimate layer in which each neuron is associated with a distinct concept, such as an object or attribute that can be identified in an image. During the training phase, an intermediate loss is computed for each neuron in the bottleneck layer, which forces each neuron's prediction to align with the actual value of its associated concept in the training image. A classification model, typically a logistic regression, is then trained using this set of concept values to predict the final target. During testing, the model first deduces the values of the concepts from the input image and then uses these values to predict the final target.

Designed for interpretability, Concept Bottleneck Models aid in demystifying the black-box nature of deep learning models:

- Users have the capacity for post-hoc model analysis, inspecting the regression from concept to target to quantify the influence of individual concepts on specific outputs. This means that after the model has been trained and deployed, users can inspect the regression from concept to target to quantify the influence of individual concepts on specific outputs.
- Users can discern which concepts the model identifies during inference, enhancing model comprehension. They can pinpoint which concepts the model detects within an image, providing a semantic context to otherwise abstract representations.
- Users can also perform test-time intervention by adjusting the value of a mispredicted concept, potentially altering the prediction outcome. This is particularly useful in domains like medical imagery, where certain concepts might be easily identifiable by human experts but are not the primary target of the model.

Koh et al. also demonstrated that mapping images to intermediate concepts instead of the final target rendered the model more robust and agnostic to spurious associations. To test the model's ability to resist such correlations, they utilized the finely annotated CUB dataset for bird classification [49] and the Place365 [58] dataset composed of different backgrounds. They crafted a spurious correlation by cropping birds from their original backgrounds and associating each with a specific Place365 background during training. The mapping was shuffled during testing, realigning each bird class to an alternate background category. This tested the model's ability to correctly classify the birds based on their features rather than the background they were associated with during training. The outcome indicated that CBM outshone traditional models, demonstrating CBM robustness against factors unrelated to the target thanks to the concept mapping. The model learned to classifies birds based on their attributes rather than their background.

Compared to standard models, CBM present certain limitations. The primary limitation is the need for meticulously annotated datasets, which demand annotations for each concept across all data points. This requirement considerably restricts the datasets on which a CBM can be trained, limiting its applicability in real-world scenarios where such detailed annotations are not available. Moreover, CBM register marginally reduced accuracy relative to traditional computer vision models, as the concepts serve as an intermediary conduit linking the image and its target.

A critique paper [34] has also shown that CBM does not necessarily learn the semantic information associated with a concept. Using saliency maps, which visually represent which parts of an image the model is focusing on when making a prediction, they show that CBM can learn correlated features of the concept values but not necessarily the concept information itself. For example, the figure in Appendix C shows a saliency map for the concept "bird's leg" in which the bird's leg is not present in the saliency map. The model learned to distinguish features the bird body features, which is a confounding factor with the bird leg, instead of the bird leg.

In summary, CBM offers increased interpretability and robustness against spurious correlations, thanks to the penultimate layer associated with concepts. However, they have certain limitations,

including the requirement for finely annotated data, slightly lower accuracy than other models, and the risk that the concepts learned may be proxies rather than the actual intended information.

2.3.2 Label-Free CBM

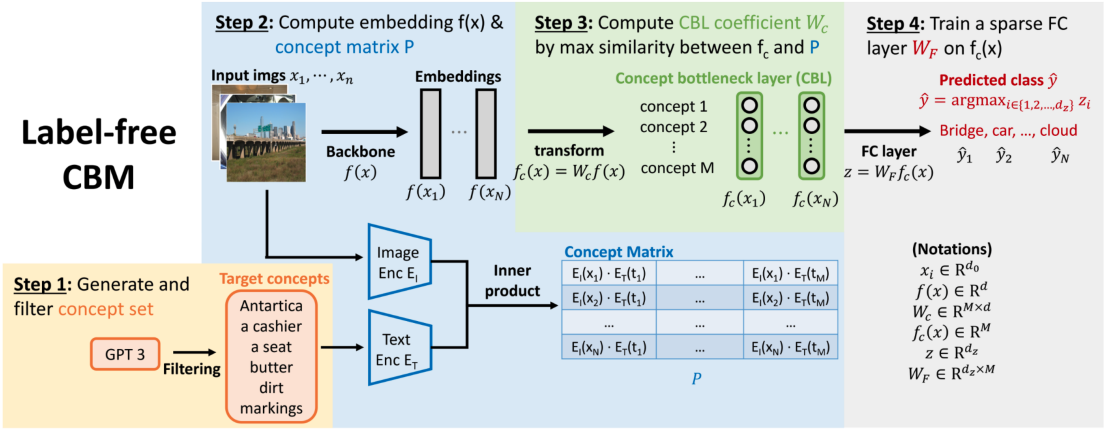


Figure 2.2: Label-Free Concept Bottleneck Model (LF-CBM) workflow overview. The framework generates a concept set using GPT, then creates a concept matrix using CLIP. Then, a projection from a backbone model to the concept layer is trained, followed by the training of a final sparse layer.

Oikarinen et al. [37] proposed a novel CBM framework, Label-Free Concept Bottleneck Models (LF-CBM), which overcomes the limitation of requiring a finely annotated dataset. This framework can transform any computer vision neural network into an interpretable CBM without needing labeled concept data while retaining high accuracy. It leverages GPT-3 to create the concepts and CLIP to train the projection from the backbone model to the concept layer. GPT-3 is a state-of-the-art language generation model, while CLIP is a vision-and-language model. Subsequently, the framework learns a sparse layer from the concepts to the target thanks to a regularization term.

Given a neural network pre-trained model called backbone, Label-Free CBM transforms it into an interpretable CBM in four steps, eliminating the need for annotated concepts:

1. **Concept set creation and filtering:**

   Oikarinen et al. generate the concepts using GPT-3 [7] with three prompts written in Appendix B. The generated concepts are then filtered to improve their quality and reduce the size of the concept set. This filtering process involves several steps, each of which helps to ensure that the final set of concepts is relevant, interpretable, and not redundant.

   (a) *Concept length*: Concepts longer than 30 characters are deleted. This helps to ensure that the concepts are concise and easy to interpret.

(b) *Remove concepts too similar to classes*: The authors use the text embedding space of the CLIP ViT-B/16 text encoder and the all-mpnet-base-v2 sentence encoder to remove concepts similar to the target classes. They delete concepts with a cosine similarity > 0.85 to any target class. This helps to ensure that the concepts are distinct from the target classes and do not overlap.

(c) *Remove concepts too similar to each other*: Using the same embedding space as above, the authors removed any concepts with a cosine similarity > 0.9 to another concept. This helps to ensure that the final set of concepts is not redundant and that each concept provides unique information.

(d) *Remove concepts not present in the training data:* The authors remove concepts that do not activate CLIP highly (e.g. value of clip is close to zero), with an average top-5 activation below an interpretability cutoff. This helps to ensure that the concepts are relevant to the training data and can be accurately projected from the backbone model to the concept layer.

(e) *Remove concepts not projecting accurately*: The authors remove the concepts that are not interpretable from the concept bottleneck layer (CBL), as described in the third step. This helps to ensure that the final set of concepts can be accurately and meaningfully interpreted in the context of the target classes.

2. **Compute embeddings from the backbone and CLIP on the training dataset:**

   With the initial set of concepts $C = \{t_1, ..., t_M\}$ created during step 1, and the training dataset $D = \{x_1, ..., x_N\}$ provided by the downstream task, the next step involves calculating the CLIP Concept activation matrix P.

   The matrix P is computed using the dot product between the CLIP image encoder E_I and text encoder E_T. Specifically, for each image x_i and concept t_j, the corresponding matrix value $P_{i,j}$ is computed as $P_{i,j} = E_I(x_i) \cdot E_T(t_j)$. Moreover, for one concept, its normalized vector with mean 0 and standard deviation 1 is defined as $\bar{P_{:,i}}$.

   In parallel, the backbone features $f(x)$, which have dimensionality $d0$ and denote the value of the last CNN layer of the backbone model before the classification DNN, are computed and saved for every image in the training dataset D.

3. **Learn projection W_c from the backbone to the concept layer:**

   The weight matrix W_c projects the backbone features to the Concept Bottleneck Layer (CBL) through the linear projection $f_c(x) = W_c f(x)$. W_c has dimension $M \times d_0$, and $f_c(x_i) \in \mathbb{R}_M$, M being the number of concepts.

   The authors use k to identify a particular neuron in the CBL, and $f_{c,k}(x) = [f_c(x)]_k$ corresponds to the neuron value on an input x. The activation pattern of a neuron q_k is defined as $q_k = [f_{c,k}(x_1), ..., f_{c,k}(x_N)]^T$, and its normalization with mean 0 and standard deviation 1 as $\bar{(q_k)}$.

The main objective is to make the CBL neurons interpretable by aligning their activation with the target concept. A new fully differentiable similarity function, $sim(t_i, q_i)$, is defined to measure the similarity between a neuron's activation pattern and the target concept. The objective function $L(W_c)$ optimizes the projection weights for interpretability.

$$L(W_c) = \sum_{i=1}^{M} -sim(t_i, q_i) := \sum_{i=1}^{M} -\frac{\bar{q_i}^3 \cdot \bar{P_{:,i}}^3}{\|\bar{q_i}^3\|_2 \|\bar{P_{:,i}}^3\|_2} \tag{2.5}$$

This equation accentuates the model activation's extremes by raising the values to the third power and then normalizing. This emphasis on extreme values facilitates the model learning from highly activated or non-activated P values, consequently enhancing the model's interpretability as it optimally tunes towards discernible and meaningful activations.

The Adam optimizer optimizes $L(W_c)$ using the pre-computed embeddings $f(x)$ from the training data D, with early stopping when the similarity on validation data starts to decrease. In the concept filtering step mentioned at step 1e, concepts not considered interpretable enough are discarded. Specifically, concepts j with $sim(t_j, q_j) <$ interpretability cutoff are discarded. The authors set the interpretability cutoff at 0.45, but different values of this hyperparameter will be experimented with during the experiment.

4. **Learn the weight W_F of the sparse final layer:**

   The final predictor is a fully connected layer $W_F \in \mathbb{R}^{d_z \times M}$, where d_z represents the number of classes. Citing the previous work of Wong et al. [52], the authors underline that sparse layers are more interpretable. Wong et al. developed a custom solver fitting elastic net [59] regularized linear models, thereby creating a sparse layer:

$$\min_{W_F, b_F} \sum_{i=1}^{N} L_{ce}(W_F f_c(x_i) + b_F, y_i) + \lambda R_\alpha(W_F) \tag{2.6}$$

   In this equation, $R_\alpha(W_F) = (1-\alpha)\frac{1}{2}||W_F||_F^2 + \alpha||W_F||_{1,1}$, $||\cdot||_F$ is the Frobenius norm, $||\cdot||_{1,1}$ is the element-wise matrix norm, L_{ce} is the cross-entropy loss, and y_i is the ground truth for the data point x_i. The authors set α at 0.99.

   The equation includes a constraint in its second term, which forces the model to be sparse. The λ hyperparameter controls the importance of this constraint: a higher λ results in a sparser matrix. This is crucial for interpretability, as a sparse matrix leads to a model that relies on a smaller set of features, thereby making it easier to understand and explain.

The results from Label-Free CBM are promising. The model was benchmarked across five diverse datasets: CIFAR-10, CIFAR-100, CUB, Places365, and ImageNet. Specifically, for the ImageNet dataset, with ResNet-50 as the backbone network, the model employed 4,505 concepts, and impressively, the training process was completed in under a day. Although the backbone model recorded a

top-1 accuracy of 76%, the Label-Free CBM experienced a slight decline, registering an accuracy of 72%.

The primary contribution of the Label-Free CBM framework is its capacity to generate Concept Bottleneck Models (CBM) from any dataset, provided there is a corresponding backbone model pre-trained on it. This is made possible by integrating both CLIP and GPT, automating the CBM construction process, and eliminating the need for manual data annotation. Additionally, incorporating a sparse layer enhances model intepretability since the set of concepts generated by GPT would be too vast to be fully comprehensible if the last layer was not sparse.

Chapter 3

Design

In this section, we first present the dataset used and their associated classification type. Then, we dive into the experiment framework with details about the models, metrics, and concepts. We then explore the project's intepretability aspect before diving into our experiment's details.

3.1 Datasets and classification type

In our research, we designed a series of experiments to unpack the capabilities and implications of CBM across various classification tasks. We sought publicly available datasets that feature humans but not as the primary classification objective (i.e., avoiding gender, race, skin tone detection, etc.) and annotated for gender.

We began with binary classification to get a foundational understanding of LF-CBM behavior. By focusing on this most basic form of classification, we aimed to establish a clear baseline, validate our methodology, and evaluate CBM fundamental behaviors without the intricacies of more complex scenarios. We employed the Doctor-Nurse dataset created in the Decentralized Information Group for this.

Building upon this understanding, we expanded to Multi-Class Classification. Using the ImSitu dataset [54], this experiment added complexity to our investigations, drawing them closer to real-world scenarios. Determining how effectively and robustly CBM scales across multiple classes was crucial.

Taking it a step further, we explored Multi-Label Classification using the MS-COCO dataset [32]. This experiment probed CBM capacity in contexts where images span multiple classifications. MS-Coco's extensive and diverse nature compared to ImSitu enhances our validation effort and expands the results to another classification type.

3.1.1 Doctor-Nurse for Binary Classification

We initiated our experiments with binary classification to assess whether LF-CBM operates as intended and if it can help debias the model. We also used this dataset to evaluate if the intepretability aspect of LF-CBM functions as designed. By confining our study to two categories, interpretability becomes more straightforward. Given our use of GPT to generate concepts, having only two classes simplifies the task of assessing the relevance of these concepts.

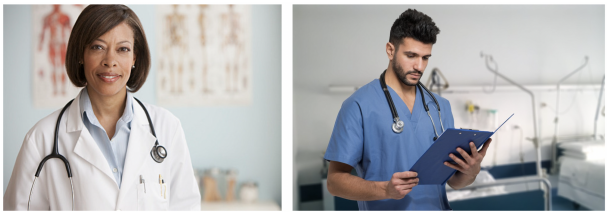


Figure 3.1: Images of a female doctor and a male nurse. The doctor wears a white coat, and the nurse wears a scrub.

We decided to use the doctor-nurse dataset established in the Decentralized Information Group for prior research [46]. This dataset features images of doctors and nurses, categorized by two sensitive attributes: gender (male or female) and skin color (light or dark). We decided to only use gender as the sensitive attribute, given its presence in both the ImSitu and MS-COCO datasets.

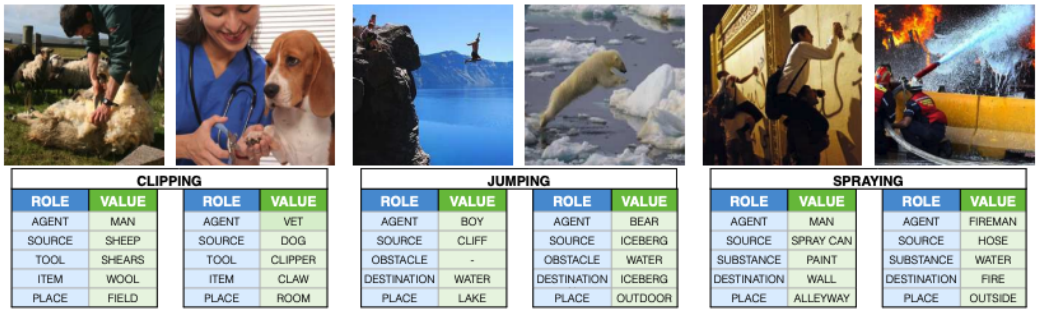
We anticipated that the results from this dataset would be relatively straightforward to interpret. This expectation originated from the distinct visual cues available to differentiate doctors from nurses, cues which can subsequently be transformed into concepts. Nurses typically wear scrubs, whereas doctors wear a white coat.

	Male	**Female**
Doctor	354	297
Nurse	262	295

Table 3.1: Number of samples in the doctor-nurse dataset by class and gender.

As the table indicates, each class and gender in the dataset has a comparable sample count. The preprocessing pipeline was straightforward: we downsampled the samples to the smallest count (262) to achieve a balanced dataset. Then, we partitioned this balanced dataset for training and testing purposes. We used the hyperparameters from the previous work [46], eliminating the need for a separate validation set. This approach allowed us to maximize our training and testing sample sizes. However, it is important to note that the sample count remains on the lower side, possibly affecting the generalizability of our test set results.

3.1.2 ImSitu Dataset for Multi-Class Classification



CLIPPING

ROLE	VALUE	ROLE	VALUE
AGENT	MAN	AGENT	VET
SOURCE	SHEEP	SOURCE	DOG
TOOL	SHEARS	TOOL	CLIPPER
ITEM	WOOL	ITEM	CLAW
PLACE	FIELD	PLACE	ROOM

JUMPING

ROLE	VALUE	ROLE	VALUE
AGENT	BOY	AGENT	BEAR
SOURCE	CLIFF	SOURCE	ICEBERG
OBSTACLE	-	OBSTACLE	WATER
DESTINATION	WATER	DESTINATION	ICEBERG
PLACE	LAKE	PLACE	OUTDOOR

SPRAYING

ROLE	VALUE	ROLE	VALUE
AGENT	MAN	AGENT	FIREMAN
SOURCE	SPRAY CAN	SOURCE	HOSE
SUBSTANCE	PAINT	SUBSTANCE	WATER
DESTINATION	WALL	DESTINATION	FIRE
PLACE	ALLEYWAY	PLACE	OUTSIDE

Figure 3.2: Sample images from three activities in the imSitu dataset. Below each picture are the annotations for the image: the verb is at the top, the left column (in blue) lists activity-specific roles, and the right column (in green) provides values for each role. These activities show variations in values for the same class, highlighting the complexity and variability in real-world activities. The *Agent* role often provides information on the gender of the primary character.

We aimed to expand the use case to multi-class classification using the imSitu dataset [54]. Multi-class classification, which classifies more than two classes, is more widespread and realistic than binary classification. We sought to assess if CBM generalizes to multi-class classification. When expanding the model to this new use case, we faced new operational challenges.

Designed for situation recognition, the imSitu dataset focuses on predicting activity and its associated semantic roles, encompassing actors, objects, substances, and locations that interact within a given situation. Each situation defines its unique activity-specific semantic roles. For instance, for *clipping*, the roles include "Agent, Source, Tool, Item, Place". Although values associated with these roles vary across samples, the roles themselves are consistent.

We determined whether a human is present in the situation through the "Agent" Role and its associated gender. Gender is inferred from the agent role if the values associated are 'female,' 'woman,' 'man,' or 'male.' Consequently, we had a count of male and female samples for all classes, simplifying the number of genders to two.

Furthermore, the role-value pair annotations provided annotated details of the image content. We used them to compare the GPT-generated concepts and the dataset's existing annotations. We also compared LF-CBM performance when used on a set of concepts fully GPT-Generated, compared to the same set of concepts into which we added the annotations.

After the data preprocessing (detailed in Appendix A), we observed a gender-based imbalance within classes and across the dataset. For a class c, we defined $n_{male,c}$ to represent the number of male samples of a class, $n_{female,c}$ to represent the number of female samples, and $n_{min,c}$ to be the smallest of the two. This imbalance is evident in specific cases. For instance, $n_{male,drumming} = 156$, while $n_{female,drumming} = 22$. A plot showing the difference between male and female count for

each class is present in the appendix (see Figure C.3).

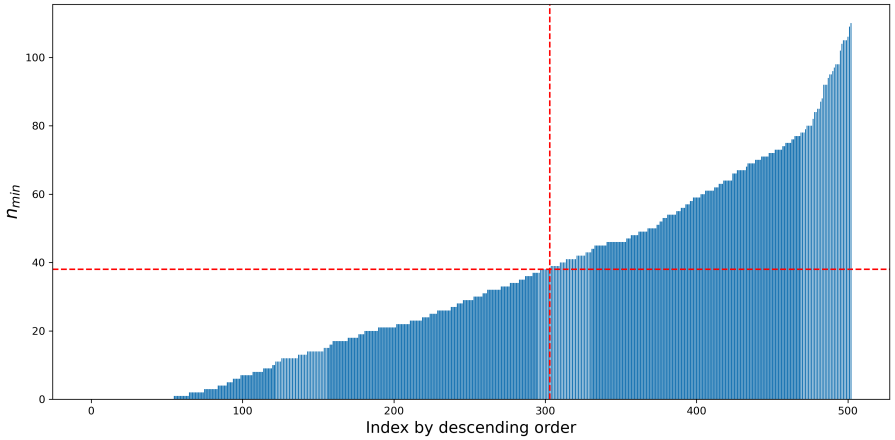


Figure 3.3: Distribution of minimum sample count (n_{min}) across 504 classes in the imSitu dataset. The x-axis represents the classes, sorted in decreasing order of n_{min}. The y-axis represents the count of samples corresponding to n_{min}. The red dotted lines mark the positions of the 200th classes, which correspond to n_{min} value of 39.

When considering different training distributions to simulate bias, we wanted to ensure that the original set, from which we split data into train, test, and validation sets, was balanced. Test and validation sets were always balanced. We downsampled the most prevalent gender for each class to attain n_{min}. We plotted the maximum count of n_{min} across the classes in figure 3.3.

To avoid excessive sample imbalance between the highest and lowest n_{min}, we constrained the classes to the 200 classes with the highest n_{min}. The highest n_{min} is 126 and the 200th n_{min} is 39. We decided to keep the sample imbalance per class for two reasons. First, we wanted to be closer to real-world scenarios with imbalanced classes. Second, having the same number of samples per class means either reducing every class sample to the lowest one, drastically reducing the dataset size, or sampling multiple times the same images that influence the concept mapping.

3.1.3 MS-COCO Dataset for Multi-Label Classification

We extended the work done on multi-class classification to the case of multi-label classification. In multi-label classification, each image can be assigned to multiple labels simultaneously. As such, it accommodates the complexity of objects and targets present in many real-world scenarios. Extending our work to multi-label classification is not a reproduction of the multi-class work, as it requires different techniques and considerations.

The Microsoft Common Objects in Context (MS-COCO) dataset [32] is an important dataset in computer vision research. This large-scale dataset can contain multiple objects by classes and

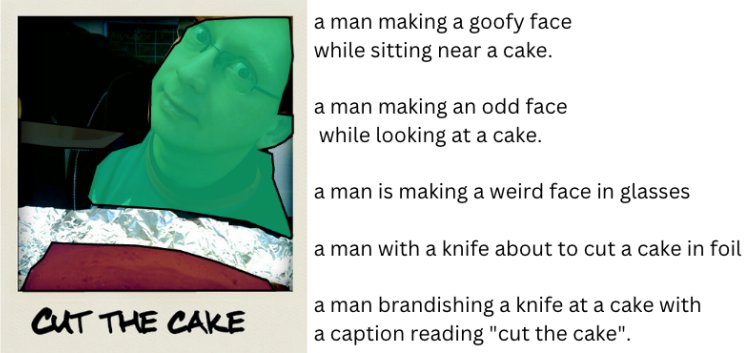


Figure 3.4: One image from MS-COCO with its associated captions. The three classes "person", "knife", and "cake" are present in the picture. The word "man" is present in the captions: the picture will be classified as having a male.

provides in-depth annotation: precise object bounding boxes, segmentation, and five captions per image providing context. With images often showcasing multiple objects, 7.7 per image on average, it becomes an excellent dataset for understanding the complexities and nuances of real-world scenes.

We determined whether or not a human was depicted in the picture by verifying if the entry included the class *person*. Then, we used the five captions to determine the gender based on whether the captions contained the words 'male,' 'female,' 'man,' or 'woman.' We also simplified the number of genders to two and discarded the entry with keywords from two genders.

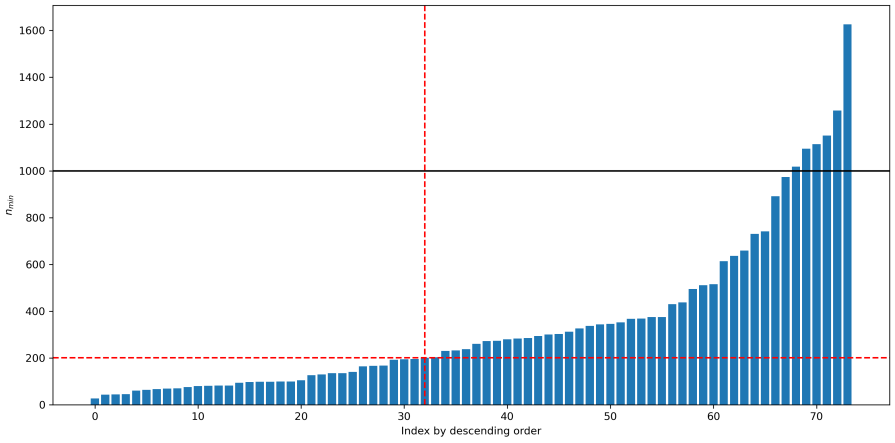


Figure 3.5: Distribution of minimum sample count (n_{min}) across 74 classes in the MS-COCO dataset. The x-axis represents the classes, sorted in decreasing order of n_{min}. The y-axis represents the count of samples corresponding to n_{min}. The red dotted line marks the 42nd class, the last one included in our experiment, having $n_{min} = 201$. The black horizontal line indicates the maximum threshold of n_{min}, set at 1000 samples.

We also observed a gender-based imbalance within classes and across the dataset for MS-COCO. The gender imbalance within classes is portrayed in a plot in Appendix (see Figure C.4). Similar to imSitu, we also downsampled the most prevalent gender for each class to attain n_{min}. We noticed that the n_{min} value is decreasing more sharply than for imSitu. We experimented on only the 42nd classes with the highest n_{min}. Yet, the 42th class has n_{min} = 201 compared to the 1st class having an n_{min} = 1626. To decrease the class imbalance in MS-COCO, which is higher than in ImSitu, we put a ceiling to the number of samples for a class, n_{min} = 1000.

We downsampled through a system of masks: when one class needed to be removed in a sample, we set both the prediction and the expected value to zero, creating a True Negative (see Appendix A for more details). This True Negative did not impact our fairness and performance metric. We kept every images that still had one class not masked. This led to keeping the initial bias in gender distribution in the dataset, with more male pictures than female pictures.

3.2 Framework: Models, Concepts, Sparsity and Metrics

3.2.1 Models

For imSitu and MS-COCO, we utilized the Resnet-50 [25] model with the Adam optimizer for baseline, a deep convolutional neural network known for its efficacy in image classification tasks. The initial set of weights for Resnet-50 is sourced from IMAGET1K_V2, which is fine-tuned on the ImageNet dataset. For our baseline experiments, we kept the Resnet-50 model frozen except for its final fully connected layer, which we adjusted based on the number of outputs required. When employed as the backbone in LF-CBM, we extracted activations from the model last convolutional layer, specifically layer 4, and then mapped them to the relevant concepts. For the multi-label classification, we changed the loss function from cross-entropy to binary cross-entropy. For the Doctor-Nurse dataset, we used the Alexnet model with SGD optimizer for baseline. Previous work [46] was done on this model, which yielded better result than ResNet. It is probably because the dataset has a small size, thus it is more suited to a simpler neural network.

3.2.2 Concept Generation

For concept generation, we used prompts from the LF-CBM paper (detailed in Appendix B). The original prompts were tilted towards image recognition, so we introduced a new prompt better suited for datasets like imSitu and the Doctor-Nurse dataset, on verb or job recognition: "List the things commonly seen around someone working as a class." For imSitu, we extracted concepts from role-value pairs, saving any value that appeared more than 15 times for a specific class as a potential concept. This allowed us to benchmark results between solely GPT-Generated concepts and a combination of GPT-Generated and imSitu-derived concepts. Before filtering during step 1d

and 1e of LF-CBM, the Doctor-Nurse dataset has 39 concepts, imSitu has 1798 concepts for the 200 classes, and MS-COCO has 520 concepts for 42 classes.

3.2.3 Sparsity and Interpretability

The sparsity of the final layer is an important hyperparameter. The number of concepts increases with the number of classes in LF-CBM, so the number of concepts can become too high to interpret. There was 4751 concepts for the 1000 classes of ImageNet: a classification layer flowing from all of the concepts to every classes is not interpretable as there would be too many concepts influencing the model, nor desirable as it's impossible that every concepts is important to classify a single class. The lower the λ would be, the sparser the model and the more interpretable it will be. However, if the model is too sparse, it will then be less accurate. The sparsity being done in the whole fully connected layer and not classes by classes, there is also the risk that all of the weights from concepts to a class become 0, the classifier for this class becoming dummy.

3.2.4 Metrics

Binary Classification

For the binary classification use case, we computed the True Positive Rates (TPR) (True Positive / (True Positive + False Negative)) for both Doctors and Nurses. We compute a metric called 'TPR Parity', which, for a given class (either doctors or nurses), measures the difference in the TPR for males and females in that class, with a positive value denoting a higher TPR for males. This metric is used to assess the fairness of our model. A value of zero would indicate that the model satisfies the criterion of Equality of Opportunity (see Equation 2.3) with respect to gender. The further this metric is from zero, the more biased the model is for a given class. A positive value indicates that the model has a higher TPR for males than females, whereas a negative value would indicate the opposite. We average the absolute value of the TPR Parity for both doctors and nurses as a metric to quantify the overall bias of our model.

Multi-Class Classification

For the multi-class classification task, we divided the classes into two groups, A and B, having a similar number of samples and each containing an equal number of classes (see Section 3.4.1 for more details on the division of classes and the introduction of gender biases). We computed the top-1 accuracy for each sample, and we aggregated the accuracy per group and for the entire dataset. Since the two groups are not exactly the same size, the overall dataset accuracy is not a simple average of the top-1 accuracies of groups A and B. As a fairness metric for multi-class classification,

we computed the difference in accuracy between male and female samples, which is equivalent to the Accuracy Parity metric defined in Equation 2.4. As with the binary classification task, a value closer to zero indicates a more unbiased model, with positive values indicating higher accuracy for males and negative values indicating higher accuracy for females. We computed the accuracy parity for groups A and B, and then averaged their absolute values to quantify the overall bias of our model. This approach was chosen because we introduced biases for different genders in groups A and B (see Section 3.4.1 for details).

Multi-Label Classification

For the multi-label classification task, we computed the performance and fairness metrics separately. We divided the classes into two groups, A and B, each with a similar number of samples, based on the methodology described in Section 3.4.1. To the best of our knowledge, there has been only one attempt to define metrics for fairness in multi-label classification [33], which assumed that there was an 'advantaged' label result where only 'favorable outcomes' are present (e.g., receiving a job offer), which does not apply to our case.

The performance metric is the micro-average F1-Score. We computed True Positives, False Positives, and False Negatives for every class. We aggregated these metrics to compute the F1-score for groups A, B, and for the entire dataset. This metric measures the model's overall performance and is consistent with imSitu measurements that compute the accuracy per group.

To compute the fairness metric, we considered each class as a binary classifier and computed the predicted probability for each sample containing this class in the testing set. The predicted probability is the post-logit value of the linear regression, mapping between 0 and 1. We then separately calculated the average predicted probabilities (APP) for males and females. We computed the difference, called 'Average predicted probability difference' (APPD). For example, for the class 'chair,' a model biased toward males would recognize a chair with a male in the picture more confidently, thus yielding, on average, a higher predicted probability for males than females. If the model were balanced, it would, on average, yield the same predicted probability for males and females, and the APPD would be close to zero. Like the two other classification tasks, positive values indicate a bias toward males, with a higher APP for males, while negative values indicate a higher APP for females. We computed the APPD separately for each class. Then, we averaged the classes' APPD for groups A and B to get the fairness metric of the groups. We averaged the absolute values of APPD of groups A and B to quantify the overall bias of our model, as groups A and B could have biases for different genders (see Section 3.4.1 for details). This final aggregation of groups A and B APPD is the metric used to evaluate the model fairness on the entire dataset.

Even though we treat every class as a binary classifier, we have chosen not to use binary classification fairness metrics for several reasons:

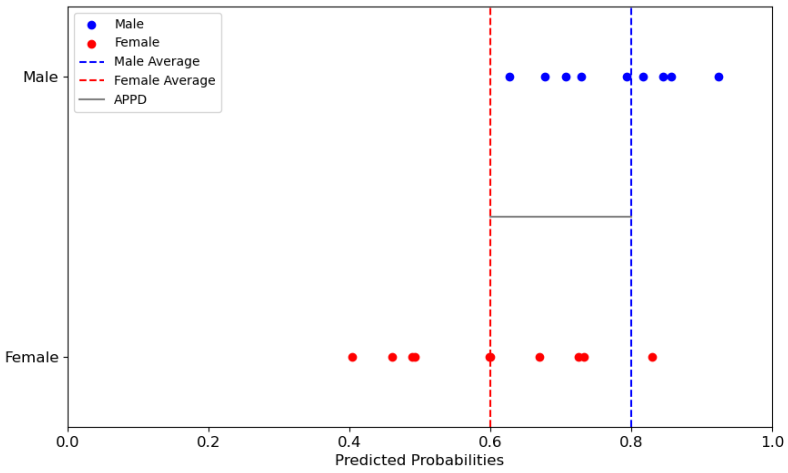


Figure 3.6: Example of predicted probabilities for a binary classifier biased towards males. The x-axis represents the predicted probabilities, the post-logit values of a binary classifier. All samples belong to the classifier class, and the predicted values display the model's confidence in its predictions. The classifier is biased towards males, as indicated by the higher average predicted probabilities for male samples compared to female samples.

1. **Nature of Multi-label Classification**: Precision, F1-Score, and Recall are computed at a specific threshold and focus on the values in the confusion matrix (True Positives, True Negatives, False Positives, False Negatives) at that threshold. This approach is suitable for binary classification, where the binary classifier will choose the other class if it does not predict the classifier class. However, it is less appropriate for multi-label classification, where each class has its own binary classifier. This is because in a multi-label classification task, a sample can belong to multiple classes simultaneously, which is fundamentally different from the binary classification case.

2. **Lack of Consensus on Threshold:** There is no consensus on an appropriate threshold value, which complicates the computation of these metrics, especially in multi-label classification, where a sample can have multiple labels, and selecting the highest predicted probability is not an option. This is because in multi-label classification, each class is considered as a separate binary classification problem, and hence, the predicted probabilities for different classes are not directly comparable.

3. **Obscuring Important Nuances:** Using a threshold obscures important nuances in the predicted probabilities, which is crucial when a model overfits one gender. For instance, there could be more true positives and false positives in the overfitted gender because the classifier more easily classifies samples of this gender as containing the class. This can actually decreases precision, as precision is defined as the number of true positives divided by the sum of true positives and false positives: if the model predicts 10 True Positive and 1 False

Postive for male, while only predicting 1 True Positive at the same threshold for female, the precision of female is higher than male. It consequently impacts the F1-score. As a result, a binary classifier that overfits one gender may more easily recognize a sample of the class associated with the overfitted gender (thus having a higher average predicted probability) but yield the same precision for both genders at a given threshold. This makes the comparison of the mean Average Precision (a metric commonly used in multi-label classification) between the two genders not relevant.

4. **Relevance of Predicted Probabilities in Multi-label Classification**: In multi-label classification, the predicted probabilities of classes can be compared, for example, to yield top-n accuracy. Therefore, predicted probability values are important, and comparing the predicted probabilities between genders is relevant. This is because it can help in understanding the model's behavior in terms of how confidently it predicts different classes for different genders, which is important for assessing the model's fairness.

3.3 Interpretability

3.3.1 Utilizing CBM interpretability

We leveraged the intrinsic interpretability of CBM to determine if LF-CBM operates as intended. Flowcharts were utilized to visually illustrate the relationship and display the weights from concepts to classes.

Although the backbone of LF-CBM remains consistent during the training process, the projection from the backbone to the concept layer is dependent on the training dataset. If the sets of concepts post LF-CBM training are identical, it is possible to compare two different projections using cosine similarity.

To understand individual sample predictions, we combined the activation metrics from the concept layer with the final layer's weights, providing a visualization of the most influential concepts.

3.3.2 Understanding Bias Propagation in LF-CBM

In LF-CBM, we identified four potential sources of bias:

1. **Concepts Generated by GPT:** The concepts generated by GPT might be biased.
2. **Values of CLIP Activation:** As underscored in research [3], the CLIP model may exhibit gender biases.

3. **Pre-trained models:** Our backbone model, Resnet50, is trained on ImageNet, shown by studies like [53] to be gender-imbalanced.

4. **Training Dataset Bias:** The training dataset bias can be picked up and propagated by the model during training.

Our observations revealed that the concepts from GPT did not explicitly mention gender. While there are potential biases in CLIP activations and in pre-trained models, they are beyond the scope of our study.

In this study, we examined LF-CBM to determine at which steps the biases from the training dataset are learned and propagated. While the training dataset is a root source of bias, it is crucial to understand how this bias is propagated through the model during training. This includes the projection from the backbone to the concept layer, and the final layer, which is the classifier from concept to classes.

3.3.3 Comparing Fairness through Awareness and Unawareness

When exclusively relying on GPT for concept generation, the gender-specific terms "male" and "female" are absent. In certain experiments, we integrated these gender-related concepts, enabling us to compare results with and without their inclusion. Moreover, the intepretability of CBM allowed us to investigate whether these gender-specific concepts influenced class predictions.

3.4 Experiment details

This section describes the various experiments conducted. The last subsection details which experiments were done on which classification type.

3.4.1 Dataset Distribution Experiments

Full Dataset Experiment

Datasets often mirror the societal gender biases prevalent at the time of data collection. As illustrated in the plots in Appendix C, there is a noticeable gender imbalance, particularly favoring males. This imbalance is apparent not only in gender-role associations in imSitu but also in the object-context biases in MS-COCO. Models are trained on these imbalanced datasets tend to perpetuate and amplify these biases. Consequently, we experimented with the entire dataset to study the behavior of LF-CBM under these biases.

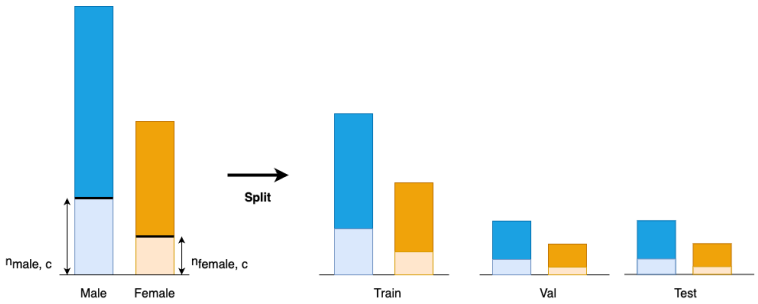


Figure 3.7: Data visualization of the train, validation, and test splits on the full dataset. The lighter parts represent one specific class. The stratified split retains the gender imbalance, both at the dataset and class levels.

For our experiments, we divided the dataset into training (60%), validation (20%), and test sets (20%), using a stratified split based on gender and classes. This approach ensured consistent gender imbalances across the splits, providing a uniform benchmark for evaluating LF-CBM behavior.

Bias Modification Experiments

We designed experiments to assess LF-CBM response to varying biases. To this end, we manually introduced specific biases and observed their impact on model performance and fairness. Our variations included:

1. One balanced dataset, with an equal number of male and female samples for each class.
2. One imbalanced dataset with half of the classes containing only male samples and the other half only female samples.
3. Another imbalanced dataset with an inversed gender bias.

For example, in the case of a binary classification on the Doctor-Nurse dataset, the balanced train set will contain both genders for doctors and nurses, one imbalanced train set will contain only male doctors and female nurses, and the other imbalanced train set will contain only female doctors and male nurses. All models were then tested on an identical test set.

Initially, we balanced all classes by setting the number of male and female samples for each class c to $n_{min,c}$, ensuring gender balance across classes despite different sample counts between classes (step (a) of Figure 3.8). We then divided the dataset into training (60%), validation (20%), and test sets (20%), using gender-based stratification to maintain balance within each class, as shown in step (b). We divided the dataset into training (75%) and testing (25%) for binary classification as we used hyperparameters from a previous experiment.

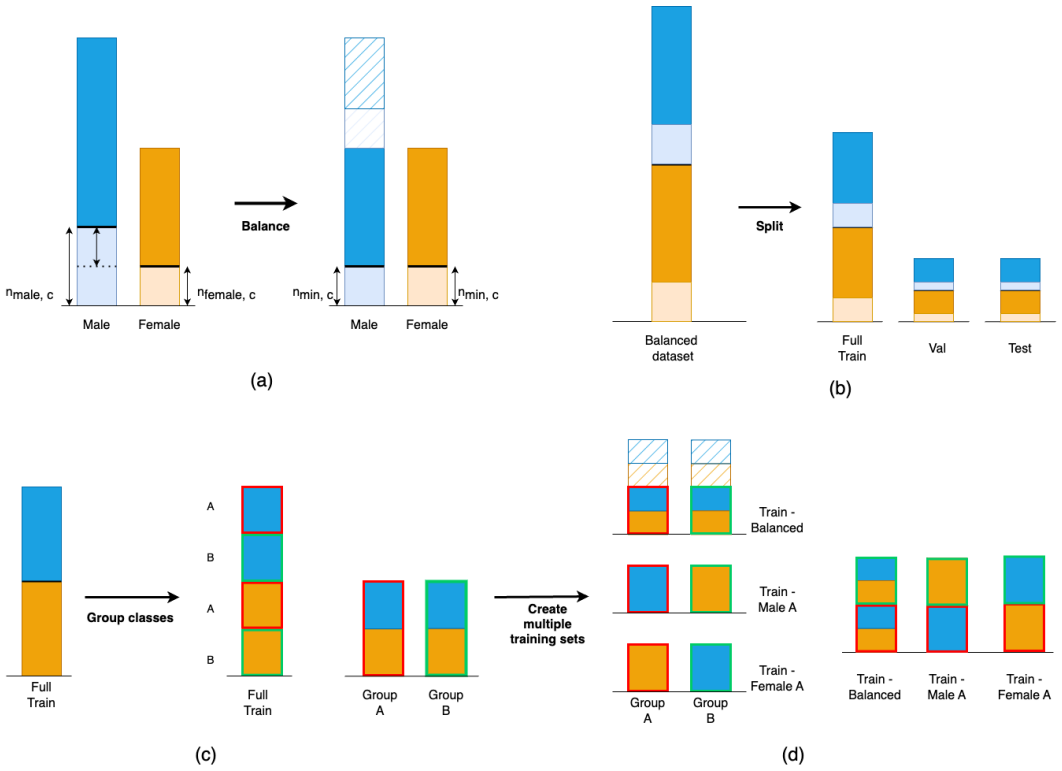


Figure 3.8: Data visualization of the different steps of the bias modification experiments. (a) represents the balancing of the dataset at a class level: each class gets balanced, taking $n_{min,c}$ samples for each class c. (b) represents the split into train, validation, and test, keeping the gender and the class balance among the splits. (c) represents the grouping of the train classes into two groups of similar size. (d) shows the split of groups A and B into three groups of the same size: Balanced, Male-A, and Female-A.

We divided classes into two roughly equal groups, A and B. For imSitu, given the nearly linear decrease in n_{min} values (see Figure 3.3), we grouped the first n/2 classes as group A and the remaining as group B. For MS-COCO, due to higher variance (see Figure 3.5), we sorted classes by decreasing n_{min} and alternated placements between groups A and B.

From this division, we created three distinct training datasets from the balanced training set (shown in step (d)). The **Train - Balanced** dataset evenly represents both genders in all classes by downsampling every class and gender to half its original size. The **Male-A** dataset exclusively features male samples in group A and female samples in group B, and vice versa for the **Female-A** dataset. This structure ensures that the models are exposed to gendered representations, with both genders present in every training set through an association between gender and classes. This association would be lost in a single-gender dataset.

For evaluation, we computed the metrics per group. While the dataset is gender-imbalanced at a group level, it is not the case at a dataset level: there is the same number of male and female

samples for every training set. Thus, we needed to inspect the bias of the groups A and B separately.

3.4.2 Interpretable Concepts Experiments

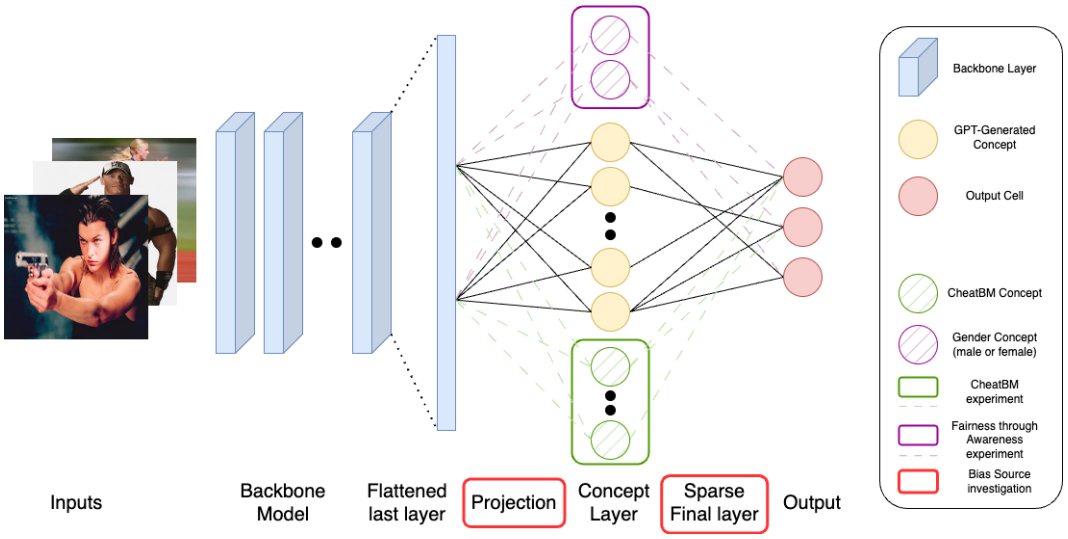


Figure 3.9: Diagram of LF-CBM illustrating the various experiments conducted. The presence or absence of augmented and gender concepts varied based on the experiments. The projection and sparse final layer steps, framed in red, were investigated to quantify the bias generated at these stages.

CBM vs "CheatBM" Comparison:

We aimed to assess the relevance of the concepts generated in the LF-CBM framework. Specifically, we inspected whether augmenting the GPT-generated concepts with extra annotations from imSitu led to significant improvements in model performance. We compared the performance and fairness of CBM using only GPT-generated concepts with those using the GPT-generated concepts complemented by imSitu annotation. GPT generated a total of 1798 concepts, and an additional 2853 were extracted from imSitu. We called 'CheatBM' the model with augmented concepts.

Awareness Experiment:

To determine the impact of explicit gender concepts on model comprehension, we introduced "Male" and "Female" concepts and evaluated the model both with and without these concepts. By evaluating the model with and without these concepts, we aimed to discern if they impacted model accuracy, especially regarding gender-specific associations. We decided to introduce both genders

instead of only one gender because the absence of a gender in a class would not necessarily mean that the opposite gender would have had a negative weight. We measured the accuracy and accuracy parity metrics and the prevalence of the gender concepts in the final layer.

3.4.3 Sparsity Experiment

Our sparsity experiment examined how sparsity, defined by the hyperparameter λ of the Elastic Net solver (see Equation 2.6), impacted the model.

We used the full dataset to explore how λ impacted a model based on a real-world dataset. Sparsity represents the trade-off between interpretability and performance: if the model is fully connected, it becomes harder to interpret as the number of concepts increases. Interpreting a model with 2000 concepts, for example, is challenging. Conversely, if the model becomes too sparse, the performance might drop, as insufficient concepts contribute to a class. When the regularization term becomes too strong, some classes can become fully sparse and have no more weight coming from the concept. They become dummy classifiers, which is not desirable.

With different values of λ ranging from 0 to 1, we trained models. We computed the density of non-zero elements of the last layer, the test performance, the number of classes having no more link to the concepts, and the average number of concepts per class.

3.4.4 Bias Propagation Investigation

Understanding how bias propagates through different layers of the model is crucial for developing strategies to debias models. This section details experiments conducted to investigate at which stages of the LF-CBM framework the biases from the training data propagate to the model.

Projection to concepts

Our objective in this experiment was to investigate if the training dataset biases were propagated in the projection from backbone to concept. The projection process uses the training dataset and CLIP to create the mapping and is unaware of the sample classes and gender. It was not possible to assess one projection's bias directly, so we compared different projections from different training datasets to extract information on what the projection mapped.

To this end, we trained multiple LF-CBM from the same backbone on five different datasets: Balanced, Male-A, Female-A, Male (only male pictures), and Female (only female pictures). We then compared the different projection weights by computing their cosine similarity, searching for patterns in the resulting values.

Last Layer

We aimed to measure how the training dataset bias propagated to the model's final layer, from concepts to classes. We first trained a common projection from the backbone to concepts to isolate this bias, using the full training set. Then, we trained five CBM across distinct training sets: Balanced, Male-A, Female-A, Male, and Female. We computed the results and compared them.

3.4.5 Experiment Application Across Classification Types

After performing different experiments on the three classification types, Binary, Multi-class, and Multi-label, we selected only some for deeper exploration based on the experiment's applicability to the classification type, as shown in Table 3.2.

	Binary	**Multi-class**	**Multi-label**
Full dataset		x	x
Bias Modification	x	x	x
Awareness vs Unawareness	x	x	x
Sparsity experiment		x	x
Bias Propagation		x	
CBM vs CheatBM		x	

Table 3.2: Experiments conducted across the different classification types

The Doctor-Nurse dataset is nearly balanced and has a low sample size, making the Full Dataset experiment not relevant. We used this dataset for an initial experiment but decided to not explore it further as the test size was only 66 samples per class and gender. The CheatBM experiment was only possible for the imSitu dataset.

For MS-COCO, we discarded the propagation bias investigation. As described in the implementation in Appendix A, we used every picture of MS-COCO in every training dataset, applying masks to remove classes only at the final layer. Thus, the projection from backbone to concepts is the same for every dataset in MS-COCO.

We also performed multi-class classification on MS-COCO, but the results were less interesting than on imSitu: the dataset was smaller regarding classes and samples, and the target was often ambiguous as there were multiple targets per image.

Chapter 4

Results

4.1 Results

4.1.1 Full dataset, Bias modification, Interpretable Concept Experiments

Note: In all the tables presented in this subsection, the rows shaded in light gray indicate the baseline models, which serve as a reference point for comparison. The metrics on the left measure the performance of the model; a higher value indicates better performance. On the right, the metrics measure fairness; a value closer to 0 indicates a fairer model. Positive values in the fairness metrics indicate a bias towards males, whereas negative values indicate a bias towards females. The 'Aggregation' column represents the average of the absolute values of the fairness metrics for groups A and B, quantifying the overall fairness. For imSitu and MS-COCO, the full training set was tested on a different test set than the Balanced, Male-A, and Female-A sets, making a comparison of its values less relevant. Values in gray are not relevant and are included only for completeness. These are the values for groups A and B on the Full dataset, which was not divided according to these groups.

Binary Classification

The Doctor-Nurse dataset, containing 262 entries per class and gender with a testing subset of 66 entries, is constrained in size and limited to binary classification. This implies results with high variance, making them less robust compared to the other experiments.

1. **CBM Balances TPR Between Classes:** In the balanced experiments, CBM Balanced achieves a TPR of 71.97% for doctors and 74.80% for nurses, in contrast to the Baseline Balanced, where the TPRs are 78.03% for doctors and 45.83% for nurses. This trend persists in unbalanced

	TPR		Accuracy	TPR Parity		
	Doctors	**Nurses**	**Total**	**Doctors**	**Nurses**	**Aggregation**
Baseline Balanced	78.03	45.83	61.93	-10.61	6.81	8.71
CBM Balanced	71.97	74.80	73.36	1.52	-1.91	1.72
CBM Balanced Gender	71.21	74.80	73.01	0	-1.91	0.96
Baseline Male-Doctor	77.27	55.70	66.49	0.03	-40.23	20.13
CBM Male-Doctor	61.36	66.31	63.84	25.76	-24.94	25.35
CBM Male-Doctor Gender	62.12	67.86	64.99	33.33	-21.86	27.60
Baseline Female-Doctor	79.55	59.69	69.62	-22.73	37.55	30.14
CBM Female-Doctor	70.45	58.88	64.67	-19.70	26.85	23.28
CBM Female-Doctor Gender	73.48	66.48	69.98	-16.67	17.81	17.24

Table 4.1: Doctor-Nurse results.

experiments: the baseline Male-Doctor model achieves a TPR of 77.27% for doctors and 55.70% for nurses. The CBM Male-Doctor model offers a more balanced outcome, with 61.36% for doctors and 66.31% for nurses.

2. **Balanced Experiments Have Lower TPR Parity:** In unbalanced training sets, the models lean towards the bias of the most represented class, such as male doctors and female nurses in the male-doctor training set.

3. **CBM Improves Fairness:** The CBM TPR Parity is lower than the baseline. For example, on the Female-Doctor training set, the baseline aggregated TPR Parity is 30.14%, while the CBM achieves 23.28% TPR Parity. However, there is an anomaly in the male-doctor experiment, where the baseline TPR Parity outperforms CBM. Our hypothesis is this result is due to the small dataset size, which can create high variance in the training and testing sets.

While the initial results are promising, suggesting that CBM effectively balances TPR across categories, the limitations of the dataset discourage definitive conclusions. These limitations also deter us from comparing gendered and non-gendered models.

Multi-Class Classification

The ImSitu dataset consists of 200 classes, making the results on this dataset more robust and having less variance compared to the binary classification scenario. In the Full Scenario, the training and testing datasets had the same bias as the original dataset for each class (see Section 3.4.1). For Balanced, Male-A, and Female-A, the 200 classes were divided into two groups, A and B, each with 100 classes and different biases (see Section 3.4.1 for details).

	Accuracy			Accuracy Parity		
	Group A	**Group B**	**Total**	**Group A**	**Group B**	**Aggregation**
Baseline Full	32.30	30.12	31.26	1.44	-4.83	3.13
CBM Full	33.15	29.92	31.62	-0.91	-5.18	3.05
CBM Full gender	33.27	30.21	31.81	-0.47	-5.15	2.65
CheatBM Full	34.26	30.18	32.32	-0.36	-5.46	2.91
Baseline Balanced	26.88	26.20	26.55	0.43	0.49	0.46
CBM Balanced	27.78	27.24	27.52	0.73	0.49	0.61
CBM Balanced Gender	27.85	27.41	27.63	-0.02	0.49	0.26
CheatBM Balanced	28.14	27.57	27.87	0.43	0.49	0.45
Baseline Male-A	24.71	23.77	24.26	13.57	-14.87	14.22
CBM Male-A	25.42	25.38	25.40	12.30	-14.23	13.27
CBM Male-A Gender	25.42	25.34	25.38	12.30	-14.63	13.47
CheatBM Male-A	24.23	25.75	24.96	12.00	-14.47	13.24
Baseline Female-A	23.07	23.32	23.19	-11.96	14.96	13.46
CBM Female-A	25.57	23.61	24.63	-10.40	13.74	12.07
CBM Female-A Gender	25.53	23.73	24.67	-10.62	13.99	12.31
CheatBM Female-A	25.23	24.00	24.65	-10.32	15.20	12.76

Table 4.2: ImSitu results.

1. **Expected Accuracy Trends Across Training Sets:** Full training sets consistently exhibit the highest accuracy, ranging between 31% and 32%. This is expected as these datasets possess consistent biases in both training and testing. Balanced datasets yield the second-highest accuracy before the biased datasets, approximately 26% to 27%. Male-A datasets achieve slightly superior accuracy (24% to 25%) compared to Female-A datasets, suggesting some disparities potentially due to their constituent classes.

2. **Coherent Accuracy Parity Observations:** As expected, balanced training sets have the lowest Accuracy Parity, around 0 to 1%, followed by the full training sets' Accuracy Parity, at 1 to 3%. The gender-biased training sets, Male-A and Female-A, show significantly higher Accuracy Parity values, between 12% and 14%. The directionality of this bias aligns with expectations for both groups. However, Female-A Accuracy Parity is slightly lower than Male-A, hinting at class disparities between groups.

3. **CBM Consistently Outperforms Baseline:** CBM, across different variations, consistently outperforms the baseline in terms of accuracy. This performance enhancement is more pronounced for the balanced and biased training sets than for the full one. We hypothesize that CBM might overfit less, with the concept layer projecting the model to a new dimension, and the sparse layer operating as a feature extractor, compared to the baseline Resnet-50 model.

4. **CBM Reduces Accuracy Parity in Biased Datasets:** For the imbalanced datasets (Male-A and Female-A), CBM successfully reduces Accuracy Parity regardless of bias directionality. For example, it shifts in Female-A CBM compared to Baseline from -11.96% to -10.40% for group A, and from 14.96% to 13.74% for group B. This could be attributed to the reduction of gender bias through the mapping of inputs to concepts. Even when there is a smaller bias with the full training set, there is still a decrease of a smaller scale in Accuracy Parity.

5. **Gender and Additional Concepts Do Not Significantly Alter Results:** CBM, with or without gender, behaves similarly across datasets. There are minor fluctuations in Accuracy and Accuracy Parity when gender concepts are incorporated, yet they do not significantly change outcomes. CheatBM performance is on par with CBM regarding both Accuracy and Accuracy Parity. The use of GPT-generated concepts does not substantially deviate from the results achieved with concepts extracted from the ImSitu dataset.

   We note that gender concepts are used with CBM 30 times for the balanced case, compared to 68 times for Male-A and 64 times for Female-A. The higher prevalence of gender concepts for imbalanced datasets shows that the model can use gender values to make its biases explicit, even though it does not alter the result.

Multi-Label Classification

	F1-Score			APPD		
	Group A	**Group B**	**Total**	**Group A**	**Group B**	**Aggregation**
Baseline Full	56.05	59.72	58.00	-2.44	1.84	2.14
CBM Full	56.01	58.52	57.35	-2.53	1.45	1.99
CBM Full gender	55.97	58.52	57.33	-2.54	1.43	1.99
Baseline Balanced	48.86	49.45	49.15	-4.02	4.21	4.11
CBM Balanced	52.75	52.59	52.67	-2.42	0.82	1.62
CBM Balanced Gender	52.63	52.71	52.67	-2.42	0.83	1.63
Baseline Male-A	48.25	48.25	48.25	-2.51	2.09	2.30
CBM Male-A	52.36	52.05	52.21	-2.42	1.34	1.88
CBM Male-A Gender	52.42	52.11	52.27	-2.37	1.40	1.89
Baseline Female-A	47.62	48.69	48.15	-4.89	3.72	4.31
CBM Female-A	52.54	52.20	52.37	-1.53	0.42	0.97
CBM Female-A Gender	52.46	52.19	52.32	-1.50	0.46	0.98

Table 4.3: MS-COCO Results.

In the MS-COCO dataset, we used two uncorrelated metrics to measure performance and fairness: the F1-Score and the Average Predicted Probabilities Difference (APPD) (see Section 3.2.4

for details). Similar to the ImSitu dataset, the Full scenario in MS-COCO has a different testing set than the Balanced, Male-A, and Female-A scenarios. The classes were again divided into groups A and B to generate different gender biases (see Section 3.4.1 for details).

1. **CBM F1-Score Similar or Better than Baseline:** The F1-Score is similar for the Full dataset between CBM and Baseline. For the other three datasets, the F1-Score of the CBM exceeds that of the baselines.

2. **Coherent F1-Score Distribution:** The F1-Score has the highest value for the full dataset with more samples at 58.15%, followed by the Balanced dataset at 49.15%, and then by the biased datasets with Male-A at 48.25% and Female-A at 48.15%. The baseline F1-scores are relatively small, but considering the multi-label classification with an imbalance in classes, they are not absurd - the F1-scores are similar to the experiments done in [50], in which the MS-COCO F1-Score was 52.52% using the full dataset.

3. **Fairness Metric Always Improves with CBM:** Transitioning from baseline to CBM always improves the APPD for every experiment. For example, in Balanced, we transition from 4.11% APPD in the baseline to 1.62% in CBM. In Female-A, the APPD is 4.31% in the baseline to 0.97% in CBM.

4. **Adding Gender as Concepts Does Not Alter Results:** The changes for both F1 and APPD between CBM and CBM with gender concepts are not significant for all the experiments.

5. **Coherent Results from Our Fairness Metric:** Comparing the APPD values of the baseline, we note that they all are as expected, except one value. The APPD of group A is -4.89% for Female-A, -4.02% for Balanced, and -2.51% for Male-A, thus having more bias towards females when trained on a female training set. For group B, it is 2.09% for Male-A and 3.72% for Female-A: the training set with a bias towards males in group B has a higher APPD value. However, there is an inconsistency for group B baseline APPD, which is higher, at 4.21%. We also note that Group A is always biased towards females, and group B is always biased towards males, probably due to the testing set.

Unified Observations

Consistent trends emerge across all datasets regarding the performance of CBM.

1. **CBM Either Maintains or Exceeds Baseline Performance:** This is true across all datasets. However, as pointed out in the Limitations (see Section 4.3.1), our baseline model was not optimized with regularization terms preventing overfitting. The results are still encouraging for the future.

2. **CBM Consistently Improves Fairness Metrics Compared to Baseline:** This improvement is achieved by simply switching from the baseline to a CBM, without any optimization towards the fairness metric. These results underline the CBM potential in rectifying biases and their robustness against gender distribution shift, encouraging future work.

3. **Gender and Extra Concepts Do Not Show Significant Differences in Both Performance and Fairness:** The results indicate negligible differences in both performance and fairness between CBM with gender and without gender. Adding concepts from imSitu annotation to those generated by GPT doesn't yield any marked improvement, suggesting potential saturation or redundancy in the combined set. It shows the relevance of the sparsity, which acts as a feature extraction of the concept layer to the class.

4.1.2 Sparsity Experiment

λ	Density (%)	Test Accuracy (%)	Zero Mapping Classes	Avg. Concepts/Class
0	99.98	29.83	0	1500
0.00001	96.63	30.21	0	1383
0.0001	44.33	31.36	0	635
0.0007	4.05	31.84	0	62
0.001	2.34	16.21	0	37
0.01	0.06	0.88	101	2
0.1	0	0	200	1

Table 4.4: ImSitu: Effect of varying the regularization parameter λ on model metrics.

λ	Density (%)	Test F1-Score (%)	Zero Mapping Classes	Avg. Concepts/Class
0	100.0	58.62	0	492
0.00001	99.93	58.57	0	492
0.0001	95.57	58.35	0	472
0.0007	37.22	58.00	0	183
0.001	26.93	56.50	0	134
0.01	3.38	44.89	0	17
0.1	0.14	3.76	34	1
0	0	0	42	0

Table 4.5: MS-COCO: Effect of varying the regularization parameter λ on model metrics.

This experiment illustrates the delicate balance required when selecting the sparsity level based on the λ value. The data reveals that both extremes of the spectrum—a very dense model (e.g., $\lambda = 0$) and a highly sparse one (e.g., $\lambda \geq 0.001$)—are suboptimal, leading to various undesirable effects.

When the model is overly dense, it creates connections between unrelated classes and concepts, hindering the training process and ultimately reducing the performance metric for imSitu, or

having no significant changes for MS-COCO. Conversely, an excessively sparse model needs more connections in the last layer to produce meaningful results, leading to a decline in performance metrics and, in extreme cases, resulting in classes with zero mappings.

For MS-COCO, the test F1-Score decreases with the density, but not linearly. The F1-Score is similar between a λ value of 0 at 58.62%, and a λ value of 0.0007 at 58.00%. However, the density then decreases from 100% to 37.22%. A sparser final layer makes the model more interpretable, going from an average concept per class of 492 to 183.

For imSitu, the best results are achieved when $\lambda = 0.0007$, resulting in a density of 4.05% and an average of 37 concepts per class. This observation of high-performing LF-CBM having relatively low sparsity aligns with our findings from the CBM vs. CheatBM comparison, where adding the 2853 concepts from imSitu to the 1798 generated by GPT did not yield significant model improvement. Given that the best-performing models retain only an average of 37 concepts, it is evident that adding more concepts is not a crucial factor in improving model performance. It also relates to our hyperparameters grid search, which had the best validation results with no concept filtering and a sparsity of $\lambda = 0.0007$. LF-CBM performs best when the concept filtering is done through the concept selection of the final sparse layer.

In conclusion, there is a trade-off between sparsity and performance. For multi-class classification, our results show an optimal value to achieve the highest accuracy with relatively low density. For multi-label classification, the density and F1-score decrease together, but not linearly. There is also an optimal density at which the F1-Score is still close to its maximum, with a lower density. A lower density, thus a higher sparsity, allows a better interpretation of the last layer weights. Interpreting the results can provide crucial information on the model's inner workings.

4.1.3 Results Interpretability

In this section, we explore the visualizations made possible by the interpretable nature of CBM. We first examine the last layer weights for Doctor-Nurse and some imSitu classes. We then investigate the interpretability of the predictions of some samples for imSitu. We discover how this last layer can provide useful information on the model's inner workings, whether through showing biases with gender concepts, showing that concepts may act as a proxy, or helping to understand some incorrect predictions.

Last Layer Weight Analysis:

Figure 4.1 reveals how the model differentiates between a nurse and a doctor. At a model level, the class 'nurse' is associated with concepts such as 'scrubs,' 'a receptionist,' or "a nurse's station," whereas 'doctor' is linked to "a doctor's office," 'a serious expression,' or 'a white coat.' This dia-

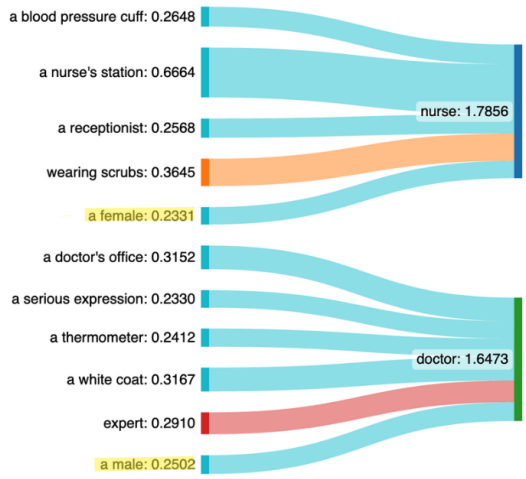


Figure 4.1: Sankey diagram of the Doctor-Nurse CBM weights for the Doctor-Male training set with gender. Highlighted values indicate the gender concepts.

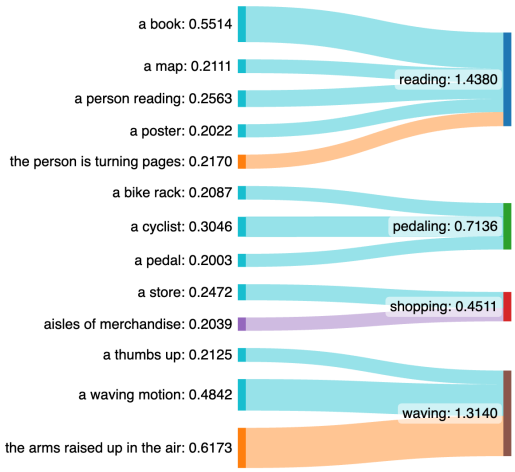


Figure 4.2: Sankey diagram of the CBM weights for four classes in the imSitu dataset trained on the balanced training set.

Figure 4.3: Sankey diagrams visualizing CBM weights for the Doctor-Nurse and imSitu datasets. The diagrams illustrate the concepts with the highest weight to ease interpretation.

gram was made from a biased model trained on the doctor-male dataset. The model establishes a connection between doctors and males, and between nurses and females.

Figure 4.2 represents the weights of four classes of a model trained on the imSitu dataset. It demonstrates that the concepts used for classification align coherently with the class. For example, the concepts with the highest weight for 'pedaling' are 'a bike rack,' 'a cyclist,' and 'a pedal.' This suggests that the model association between concepts and classes is functioning effectively.

However, when examining figure C.5 in Appendix C, which is trained on a balanced Doctor-Nurse dataset, we observe that the model has learned a different set of concepts, most of which are not visual. For example, the model associated the concept "professional" with nurses and "a medical degree" with doctors. This raises the question of whether the model is learning concepts genuinely related to the class or, instead, proxy information correlated with both the class and the concept. In figure 4.1, the concept with the highest weight for 'nurse' is 'a nurse's station,' yet most pictures in the dataset do not feature a nurse's station. This inconsistency makes us consider if we should apply a lower cutoff value in step 1b of LF-CBM, filtering more concepts too similar to classes.

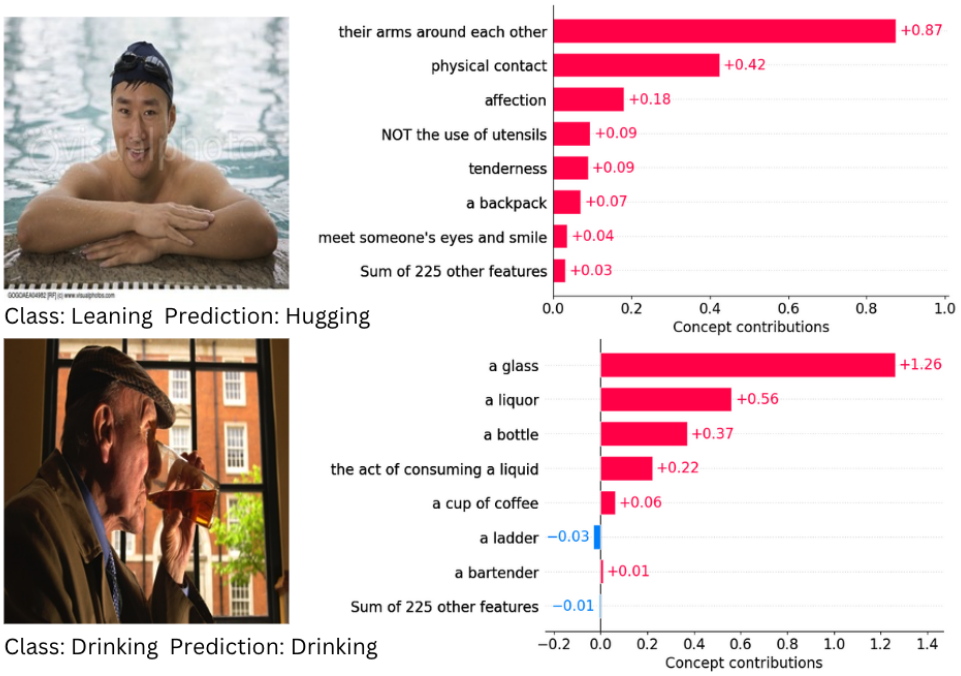


Figure 4.4: Two sample images from the imSitu dataset, their predictions made by the model, and the contribution of each concept in making those predictions.

Interpretation for Sample prediction:

The sample-level visualizations in Figure 4.4 enable us to understand the classification for two particular samples. For the 'drinking' image, the model accurately predicts the presence of a glass and liquor. However, the presence of other concepts not present in the picture but commonly associated with drinking (e.g., a bottle, a cup of coffee) might suggest that the model is learning proxies rather than the semantic concepts. Instead of associating 'a cup of coffee' with an actual cup of coffee, the model may learn correlated information, such as hand position. This observation aligns with our previous findings from the Doctor-Nurse CBM diagram.

These visualizations are particularly useful for understanding incorrect predictions. For example, the first image in Figure 4.4 depicts someone leaning in a swimming pool. The model accurately identifies that the individual's arms are around each other, leading to a prediction of 'hugging,' despite the image only featuring a single male. Such visualizations provide valuable insights into the model's behavior.

4.1.4 Bias Propagation Investigation

Projection to Concepts

	Balanced	**Male-A**	**Male-B**	**Male**	**Female**
Balanced	1.00	-	-	-	-
Male-A	0.3570	1.00	-	-	-
Male-B	0.3708	**0.1500**	1.00	-	-
Male	0.3573	0.3592	0.3556	1.00	-
Female	0.3560	0.3422	0.3570	**0.4059**	1.00

Table 4.6: Cosine similarities between the projections of different subsets of the imSitu dataset.

After computing the cosine similarities between the five datasets, we observed that the similarities between the models are relatively consistent, hovering around 0.35, except for two notable exceptions. These are highlighted in bold in Table 4.6: the cosine similarity between Male-A and Male-B, and the cosine similarity between Male and Female.

The cosine similarity between Male-A and Male-B is 0.15, significantly lower than the others. We hypothesize that this discrepancy arises because Male-A learns a biased projection of the concepts, associating Group A concepts with 'male' and Group B concepts with 'female', while the associations for Male-B are reversed. This difference in learned associations likely accounts for the lower similarity.

The highest cosine value, 0.4059, is observed between the Male and Female models. We hypothesize that because the Male and Female models are trained exclusively on one gender, they do not associate gender with concepts, but instead associate concepts with contextual information. This might lead to a more gender-independent dataset learned by each gender-specific model compared to the balanced dataset.

This observation suggests that even without knowing the gender or the classes of the images during the projection training from backbone to concepts, the training set influences the projection. More specifically, if the training set contains both genders and does not have an identical distribution of gender across classes, this bias gets propagated to the projection, which learns a gendered representation of the concepts.

Last Layer

The bias in the training dataset clearly propagates to the last layer, which is expected as this layer is a classifier that can be trained on a biased dataset. This is why, despite the identical image-to-concept projections across models, there is a significant disparity in accuracy results between each gender

	Accuracy					Accuracy Parity		
	Male A	**Male B**	**Female A**	**Female B**	**Total**	**Group A**	**Group B**	**Aggregation**
Balanced	34.09	30.76	37.29	34.13	34.03	-3.19	-3.38	3.22
Male	38.85	35.93	26.87	25.27	32.22	11.98	10.66	11.39
Female	26.53	23.14	42.31	43.13	33.16	-15.78	-19.99	17.77
Male-A	41.78	19.21	19.15	47.27	31.95	22.63	-28.05	25.34
Female-A	21.15	39.80	44.57	19.27	30.99	-23.42	20.53	21.98

Table 4.7: ImSitu Dataset: Results showcasing the last layer bias. The table includes Accuracy for for each gender subgroup (Male A, Male B, Female A, Female B) and the overall model accuracy across all groups, as well as the Accuracy Parity.

and group.

This trend is consistent across all rows of the table. For example, in the Male-A training set, the accuracy for Group A classes tested on male samples is 41.78%, which is more than twice the accuracy for female samples in Group A, which is 19.15%. Conversely, for the Female-A training set, the accuracy for Group A classes tested on female samples is 44.57%, more than double the 21.15% accuracy observed when the same classes are tested on male samples.

This pattern indicates that the model performs better on the gender that is overrepresented in each group of the dataset, thereby propagating the training set bias. It underscores that the last layer significantly learns the bias present in the training dataset. This is consistent with the fact that the last layer is trained as a classifier from a biased training set. It suggests that we can debias LF-CBM by focusing on this layer, and also provides us with more quantitative metrics to assess the result than the Projection Bias.

4.2 Discussion

Our experimentation on Concept Bottleneck Models (CBM) across three different datasets enabled us to assess the use of CBM in terms of fairness and performance. Our findings suggest that the use of CBM consistently enhances fairness metrics compared to traditional models across almost every classification case and training dataset bias. Additionally, the performance of CBM compared to baseline models is similar and can even be higher in some experiments, regardless of the presence of gender concepts and supplementary concepts. These findings are promising for future research.

We also observed the critical role of the last layer's sparsity, acting as a balance between performance and interpretability. A last layer too dense results in less interpretability due to a large number of concepts impacting the classification, while a too sparse last layer yields lower performance, even having all of the weights as 0 in extreme cases.

We used the interpretability of CBM to understand and visualize how LF-CBM performed the classification. The associations between classes and concepts were coherent, and we observed that the model learned to associate gender concepts with classes on a biased Doctor-Nurse dataset. However, we also observed that some concepts with a high degree of influence are not visual, raising suspicion that the model might learn to detect confounding factors instead of the concept's semantic value, as suggested in the CBM critique paper by Margeloiu et al. [34]. This underscores the importance of CBM's interpretability when working on fairness.

We examined at which step of LF-CBM the training dataset bias propagates to the model. Our results show that biases in the training dataset are picked up by both the projection and the final layer, underscoring the importance of addressing bias at multiple points in the model. We observed that the final classifier was an easier entry point to debias LF-CBM, as we can more easily measure the biases learned at this layer.

To enhance the robustness of our results, we developed data processing pipelines for imSitu and MS-COCO, generating multiple training datasets with distinct biases while maintaining balanced validation and testing sets. This approach enabled us to use various training set configurations instead of relying solely on the original one, achieving a broader benchmark compared to preceding studies [50, 57]. Past research employed the original datasets' train-test split, which lacked stratification by gender, thereby obfuscating the interpretation of results.

Overall, our results underscore the potential of CBM in enhancing fairness in computer vision. We observed that CBM improved fairness without any explicit fairness constraint in different datasets and different bias settings. This promising result on fairness can be extended by using tabular fairness methods on the concept layer (see Section 5.1). Additionally, CBM interpretability was valuable during our thesis to understand CBM's inner workings, which is a differentiating point compared to previous work on the subject (detailed in the Background, Section 2.2)).

4.3 Limitations and Future Work

4.3.1 Limitations

Our experiments aimed to provide a comprehensive study of the effects of CBM on both fairness and accuracy. However, there are several limitations to our experiment, which we describe below:

1. **Proxy Information**: Margeloiu et al. (2021) demonstrated that CBM can learn confounding variables instead of the actual concepts themselves, even with finely annotated data. This limitation might also occur in LF-CBM, where the concepts are recognized through CLIP instead of fine annotation. Inconsistencies observed in the interpretability experiments suggest this limitation, which could affect both the model's interpretability and performance.

2. **Regularization term for the sparse layer:** The regularization (referenced in equation 2.6) is applied at a layer level instead of a class level on the last layer. This leads to more parameters filtered for underrepresented classes in the training set, resulting in worse results than over-represented classes. Regularization should be applied at a class level to ensure similar filtering across classes.

3. **Bias sources:** Among the four potential sources of bias listed in section 3.3.2, we can only assess and mitigate the last one: the training dataset bias. The biases present in the backbone model and CLIP are challenging to assess and mitigate. Since CLIP is a critical part of the LF-CBM pipeline and has been shown to be biased [3], its biases inevitably propagate to the model.

4. **Statistical Significance**: Due to time constraints, conducting multiple iterations of all experiments to establish statistical significance was not feasible. More specifically, randomizing the classes within the groups (while still ensuring the groups are approximately the same size) and the train, validation, and test split of the datasets within the experiment would bring more statistical significance to the results. We have seen some inconsistencies in the results among Groups A and B, probably due to the class attribution between groups.

5. **Metrics:** Our metrics allow us to assess the performance and fairness of CBM, but we would need a variety of different metrics to evaluate the model comprehensively. During the experiments, one of our assumptions was that type I (False Positive) and type II (False Negative) errors had the same severity. This assumption is crucial for the validity of all of our metrics. The Average Predicted Probabilities Difference metric used to evaluate fairness on the multi-label dataset also has limitations. It considers every class as a binary classifier and does not test the classifier on negative samples. It also does not compare the different predicted probabilities across classes for a given sample.

6. **Baseline results:** We have benchmarked our result against a ResNet-50 fine-tuned on the training datasets. We could have put more effort into the baseline, for example, by applying regularization. We have also not benchmarked our solution compared to previous work optimizing fairness metrics for visual recognition [50, 57]. However, our work studied only the use of CBM and did not incorporate any fairness constraints or objectives.

7. **Difference in class size:** We decided to maintain the imbalance between the n_{min} of classes to align our experiment with real-world scenarios and to avoid discarding a large portion of our data through undersampling. Although this decision ensures consistency in class imbalances in training and testing, it also introduces the risk of the model overfitting to overrepresented classes. This decision could impact the model's performance.

It is essential to consider these limitations when interpreting the results of our experiments and when designing future studies.

4.3.2 Future Research Directions

To the best of our knowledge, this thesis is the first research on the fairness and debiasing aspect of CBM leveraging the capabilities of LF-CBM [37] to transform any backbone model into a CBM, even without finely annotated datasets. While our findings indicate that CBM tends to reduce model bias, several questions remain unanswered, presenting numerous possibilities for future research:

1. **Developing an Unsupervised Debiasing Method for Computer Vision:** LF-CBM does not require gender annotations, setting it apart from other debiasing methods. Although our experiments involved annotated datasets to generate gender bias, LF-CBM operates independently of gender annotations. CBM maps an image representation to a tabular representation with the concept layer, suggesting that incorporating tabular fairness methods (see Section 5.1) could enhance the debiasing process, using CLIP to assess gender. This approach would eliminate the need for gender annotations, facilitating the debiasing of any model used for visual recognition while improving interpretability. Our findings suggest that a significant portion of bias originates from the last layer; therefore, mitigating bias at this stage could lead to fairer models. However, this approach raises ethical concerns, as it involves using CLIP to determine gender and potentially altering model predictions based on this single CLIP prediction.

2. **Examining the Concept-Class Mapping Across Different Biases:** Our experiment on last layer bias (see Section 4.1.4) revealed that the last layer learns differently based on the training dataset. Analyzing the changes in the last layer across various training sets could show how model variations depend on the training set.

3. **Investigating the Co-Occurrence of Concepts and Gender:** Our analysis suggests that concepts may serve as proxies for co-occurring information, including gender. The experiment in Section 4.1.4 indicated that the model learned gendered class representations, even without gender concepts in the CBM. This finding implies that the model internalized gendered representations within the concepts, using them as gender proxies. Future research should examine the relationship between gender and concepts, correlating this information with concept importance in classification to elucidate the origins of bias in CBM and identify potential solutions.

4. **Exploring Transfer Learning in Concept Projection:** The performance in our last layer bias experiment (see Section 4.1.4) surpassed the ones of our 'Biased Dataset' experiment (Table 4.2), possibly due to a more accurate backbone-to-concepts projection achieved using the full dataset training set. As the projection and the last layer can operate independently, future research should investigate the efficacy of transfer learning on CBM by training the projection on a large dataset (e.g., ImageNet) before fine-tuning the final layer.

5. **Enhancing Concept Layer Interpretability Through Sample Analysis:** Instead of comparing

projection from backbone to concepts across different datasets, future research could compare concept values applied to the same image after training the projection on various datasets. This approach would involve generating the activation values of the concept layers for each model using the same dataset, followed by a comparison of these values. Such an analysis could provide deeper insights into model behavior during utilization.

In summary, this thesis lays the foundation for further research into the fairness and debiasing aspects of CBM. Our proposed future research directions aim to address the unanswered questions raised by our study and advance the field towards more fair and interpretable models through CBM.

Chapter 5

Related Work

This chapter provides a comprehensive review of existing literature and methodologies in three key areas relevant to this research: fairness in tabular datasets, interpretability in computer vision, and general computer vision bias mitigation. While the Background Chapter focused on foundational concepts and the specific use work directly relevant to our research, this chapter aims to provide a broader perspective by reviewing more general methods and approaches in the field. This will help in contextualizing our work within the larger landscape of ongoing research.

5.1 Fairness on Tabular Dataset

Tabular data, organized into rows and columns similar to spreadsheets, is a common format in various research and application domains. Ensuring fairness in tabular data is crucial to prevent the perpetuation of biases and to make fair decisions in various applications, such as credit scoring, criminal justice, and healthcare. Given our suggestion to treat the CBM concept layer as tabular data to apply fairness methods, we have a particular interest in tabular fairness. Methods for ensuring fairness in tabular data can be broadly categorized into three groups: pre-processing, in-processing, and post-processing methods.

1. **Pre-Processing**: These approaches primarily focus on modifying the training data to mitigate biases. Causal methods identify dependencies between sensitive and non-sensitive variables to eliminate biases [12, 20, 44]. Another technique involves partitioning the data into groups and training distinct classifiers for each, thereby optimizing accuracy for each subgroup [17, 38]. Data representation methods aim to create new, debiased feature spaces [9, 22]. Additionally, sampling methods modify the training dataset by oversampling instances near decision boundaries to balance class distributions [28]. However, pre-processing methods may lead to information loss and may not always be effective in completely eliminating biases.

2. **In-Processing**: These approaches incorporate fairness metrics directly into the model training process. Reweighing adjusts the weight of training instances, either by prioritizing sensitive samples [27] or by dynamically adjusting weights to meet fairness criteria during gradient descent [43]. Constraint-based methods aim to balance fairness and accuracy by introducing fairness terms into the optimization function [2, 10, 21]. Adversarial training includes an additional model that tests the primary model fairness by penalizing predictability of sensitive attributes [1, 48]. However, in-processing methods often involve a trade-off between fairness and model performance.

3. **Post-Processing**: These methods focus on adjusting the model output to meet fairness criteria. Probability calibration ensures that the rate of positive predictions is consistent across all subgroups [11, 40]. Similarly, threshold adjustment techniques modify decision boundaries to optimize fairness metrics such as equalized odds [24]. However, post-processing methods may sometimes lead to counter-intuitive results and may not be suitable for all applications.

In conclusion, ensuring fairness in tabular data is paramount and can be addressed at various stages of the model development process. Each category of method has its strengths and limitations, and the choice of method may depend on the specific application and the nature of the biases present in the data.

5.2 Interpretability in Computer Vision

The widespread adoption of Computer Vision (CV) solutions has made understanding CV models inner representations and learnings a crucial research area. Model interpretability is fundamental to ensure the fairness, robustness, and accountability of computer vision applications in various domains such as healthcare, autonomous vehicles, and surveillance. Researchers have made significant progress in this field, and we will briefly discuss the most common techniques. This understanding benefits our work as we can combine these interpretability techniques with CBM concepts. We can assess if the concepts are associated with the semantic value of the concept or if they are only proxies of confounding variables.

Gradient-weighted Class Activation Mapping (Grad-CAM) [45] visualizes the areas in an image that a convolutional neural network (CNN) focuses on when making a prediction. Grad-CAM computes the gradient of the target class score with respect to the feature maps of the final convolutional layer. These gradients are used to obtain the neuron importance weights, which result in a heatmap highlighting the significant regions in the image for the predicted class. This heatmap helps to understand which parts of the image were most important in the classification process. However, Grad-CAM may sometimes struggle to highlight the most relevant regions in images with multiple objects or complex backgrounds.

Testing with Concept Activation Vectors (TCAV) [29] provides an interpretation of a model in terms of concepts rather than at a pixel level. The authors define a concept activation vector for a concept by computing the activations from a set of images representing the concept. They can then compute the directional derivative with respect to a concept to quantify the model prediction sensitivity to the concept. This approach enables a more intuitive interpretation of the model, similar to CBM, linking the models to concepts. However, TCAV may sometimes require a considerable amount of labeled data to accurately compute the concept activation vectors.

Figure 5.1: Grad-CAM example. The left image is the original image with a cat and a dog. The right image is the Grad-CAM visualization for ResNet-18 and the target 'cat.' The Grad-CAM visualization helps to assess whether the model is focusing on the relevant regions of the image to make the prediction.

In conclusion, interpretability in computer vision is paramount to ensure the robustness and fairness of models in various applications. Techniques such as Grad-CAM and TCAV provide valuable insights into the inner workings of models. They can be combined with CBM concepts to assess their associations with semantic values or confounding variables. However, these techniques are limited and may require careful implementation and interpretation.

5.3 Computer Vision Bias Mitigation

Bias in computer vision models can lead to unfair and inaccurate predictions, affecting various applications such as facial recognition, object detection, and situation recognition. Therefore, it is crucial to incorporate fairness constraints and develop methods for bias mitigation in computer vision models. While the background section focused on visual recognition, it is important to note that the work related to debiasing computer vision models is wider than this field. This section discusses various techniques used for bias mitigation in computer vision beyond visual recognition.

Some techniques used for tabular data can also be applied to computer vision. For example, constraints to guarantee some fairness metrics can be incorporated during model optimization [2, 10, 21]. Data sampling methods can be used to create a more balanced dataset, more evenly distributed in terms of sensitive attributes [43]. Reweighting methods can be used to assign different weights to the training samples based on their importance [56]. However, it is important to note that these methods may not always fully address the unique challenges associated with bias in images, such as biased feature representations and imbalances in the visual data.

Other methods are more specific to computer vision, leveraging the unique input format of images to perform feature extractions. In these methods, an encoder extracts a latent representation of the image, for example, through an autoencoder. This latent representation can be debiased through an adversarial model predicting the sensitive attribute and masking it through inverse gradient update [18, 26, 55]. This representation of images in a latent space can be used for data generation: CycleGAN, for example, generates underrepresented samples [4] from the latent representation. However, it is important to note that these methods may introduce other challenges, such as the potential for adversarial attacks and the need for large amounts of labeled data to train the models effectively.

Chapter 6

Conclusion

In this thesis, we aimed to evaluate the fairness aspect of Concept Bottleneck Models (CBM) and assess their potential as debiasing techniques for visual recognition tasks. The recent research Label-Free Concept Bottleneck Models (LF-CBM) has enabled the transformation of any Computer Vision classification models into CBM by mapping them to a concept layer before classifying through a sparse layer.

Our comprehensive research spanned several critical areas. We started by evaluating the impact of LF-CBM on fairness and performance, creating an evaluation framework to manipulate bias and assess the model. We implemented our framework on three different datasets commonly used in fairness research, thereby improving the data processing pipeline compared to previous work. We investigated how the sparsity of the last layer impacts the model in terms of interpretability and performance. Additionally, we evaluated at which point of the LF-CBM pipeline the training set bias propagates to the model.

During our experiment, we encountered the unique challenge of evaluating fairness for multi-label classification without 'favorable outcomes,' a scenario that has not been previously addressed in the literature. Current fairness metrics are designed for binary or multi-class classification tasks and are unsuitable for multi-label classification scenarios. To address this gap, we developed a novel metric, the 'Average Predicted Probability Difference'. This metric enabled the evaluation of our model by measuring the differences in predicted probabilities across different genders, thereby quantifying model fairness in multi-label classification tasks.

Our findings suggest that the sole use of CBM tends to reduce bias in the model. The performances and fairness of CBM are comparable, if not better, than the baseline. Remarkably, CBM showed promising results in fairness improvement without imposing any fairness constraints but merely by transitioning from traditional models to CBM.

However, it is worth noting the limitations of our study, including that CBM sometimes learns

confounding factors rather than the actual concepts, the challenges of mitigating all potential sources of bias, and the difficulties in finding the correct regularization at a class level.

Looking forward, we proposed several directions for future research. One direction is creating an unsupervised computer vision debiasing method by combining the tabular representation of concepts with gender recognition through CLIP to perform tabular fairness. Other directions include analyzing how the last layer changes when trained on different datasets with various biases and investigating the co-occurrences of concepts with gender. Each of these directions offers a valuable opportunity to build on the work presented in this thesis and contribute to the ongoing efforts to create fairer and more interpretable machine learning models.

In conclusion, we believe that our work significantly contributes to the ongoing efforts to develop more responsible artificial intelligence (AI) systems. By providing a comprehensive framework to evaluate and understand biases in models and demonstrating the efficacy of Concept Bottleneck Models as a debiasing technique, we hope to encourage further research and development in this area. Ultimately, we hope that our work will help pave the way for the development of AI models that are not only high-performing but also fair and interpretable.

Bibliography

[1] Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. "One-network adversarial fairness". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 2412–2420.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. "A reductions approach to fair classification". In: *International conference on machine learning*. PMLR. 2018, pp. 60–69.

[3] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. "Evaluating clip: towards characterization of broader capabilities and downstream implications". In: *arXiv preprint arXiv:2108.02818* (2021).

[4] Amjad Almahairi, Sai Rajeshwar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. "Augmented cyclegan: Learning many-to-many mappings from unpaired data". In: *International conference on machine learning*. PMLR. 2018, pp. 195–204.

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. *Machine Bias*. Accessed: 14/08/2023. 2016. URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[6] Cody Blakeney, Gentry Atkinson, Nathaniel Huish, Yan Yan, Vangelis Metsis, and Ziliang Zong. "Measuring bias and fairness in multiclass classification". In: *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*. IEEE. 2022, pp. 1–6.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[8] Joy Buolamwini and Timnit Gebru. "Gender shades: Intersectional accuracy disparities in commercial gender classification". In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 77–91.

[9] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. "Optimized pre-processing for discrimination prevention". In: *Advances in neural information processing systems* 30 (2017).

[10] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. "Classification with fairness constraints: A meta-algorithm with provable guarantees". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 319–328.

[11] Irene Chen, Fredrik D Johansson, and David Sontag. "Why is my classifier discriminatory?" In: *Advances in neural information processing systems* 31 (2018).

[12] Silvia Chiappa and William S Isaac. "A causal Bayesian networks viewpoint on fairness". In: *Privacy and Identity Management. Fairness, Accountability, and Transparency in the Age of Big Data: 13th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Vienna, Austria, August 20-24, 2018, Revised Selected Papers 13* (2019), pp. 3–20.

[13] David Danks and Alex John London. "Algorithmic Bias in Autonomous Systems." In: *Ijcai*. Vol. 17. 2017. 2017, pp. 4691–4697.

[14] Jeffrey Dastin. *Amazon scraps secret AI recruiting tool that showed bias against women*. Accessed: 14/08/23. 2018. URL: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G.

[15] Amit Datta, Michael Carl Tschantz, and Anupam Datta. "Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination". In: *arXiv preprint arXiv:1408.6491* (2014).

[16] Christophe Denis, Romuald Elie, Mohamed Hebiri, and François Hu. "Fairness guarantee in multi-class classification". In: *arXiv preprint arXiv:2109.13642* (2021).

[17] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. "Decoupled classifiers for group-fair and efficient machine learning". In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 119–133.

[18] Harrison Edwards and Amos Storkey. "Censoring representations with an adversary". In: *arXiv preprint arXiv:1511.05897* (2015).

[19] Robert Fullinwider. "Affirmative action". In: (2001).

[20] Bruce Glymour and Jonathan Herington. "Measuring the biases that matter: The ethical and casual foundations for measures of fairness in algorithms". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 269–278.

[21] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. "Satisfying real-world goals with dataset constraints". In: *Advances in neural information processing systems* 29 (2016).

[22] Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. "Obtaining fairness using optimal transport theory". In: *International conference on machine learning*. PMLR. 2019, pp. 2357–2365.

[23] Patrick Grother, Mei Ngan, and Kayee Hanaoka. *Face recognition vendor test (fvrt): Part 3, demographic effects*. National Institute of Standards and Technology Gaithersburg, MD, 2019.

[24] Moritz Hardt, Eric Price, and Nati Srebro. “Equality of opportunity in supervised learning”. In: *Advances in neural information processing systems* 29 (2016).

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

[26] Sunhee Hwang, Sungho Park, Dohyung Kim, Mirae Do, and Hyeran Byun. “FairfaceGAN: Fairness-aware facial image-to-image translation”. In: *arXiv preprint arXiv:2012.00282* (2020).

[27] Heinrich Jiang and Ofir Nachum. “Identifying and correcting label bias in machine learning”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 702–712.

[28] Faisal Kamiran and Toon Calders. “Data preprocessing techniques for classification without discrimination”. In: *Knowledge and information systems* 33.1 (2012), pp. 1–33.

[29] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. “Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)”. In: *International conference on machine learning*. PMLR. 2018, pp. 2668–2677.

[30] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R Rickford, Dan Jurafsky, and Sharad Goel. “Racial disparities in automated speech recognition”. In: *Proceedings of the National Academy of Sciences* 117.14 (2020), pp. 7684–7689.

[31] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. “Concept bottleneck models”. In: *International conference on machine learning*. PMLR. 2020, pp. 5338–5348.

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. “Microsoft coco: Common objects in context”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[33] Tianci Liu, Haoyu Wang, Yaqing Wang, Xiaoqian Wang, Lu Su, and Jing Gao. “SimFair: A Unified Framework for Fairness-Aware Multi-Label Classification”. In: *arXiv preprint arXiv:2302.09683* (2023).

[34] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. “Do concept bottleneck models learn as intended?” In: *arXiv preprint arXiv:2105.04289* (2021).

[35] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. “A survey on bias and fairness in machine learning”. In: *ACM computing surveys (CSUR)* 54.6 (2021), pp. 1–35.

[36] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. “Dissecting racial bias in an algorithm used to manage the health of populations”. In: *Science* 366.6464 (2019), pp. 447–453.

[37] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. “Label-Free Concept Bottleneck Models”. In: *arXiv preprint arXiv:2304.06129* (2023).

[38] Luca Oneto, Michele Doninini, Amon Elders, and Massimiliano Pontil. “Taking advantage of multitask learning for fair classification”. In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 227–237.

[39] Dana Pessach and Erez Shmueli. “A review on fairness in machine learning”. In: *ACM Computing Surveys (CSUR)* 55.3 (2022), pp. 1–44.

[40] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. “On fairness and calibration”. In: *Advances in neural information processing systems* 30 (2017).

[41] Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. “Discovering fair representations in the data domain”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 8227–8236.

[42] Inioluwa Deborah Raji and Joy Buolamwini. “Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products”. In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 429–435.

[43] Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. “Fairbatch: Batch selection for model fairness”. In: *arXiv preprint arXiv:2012.01696* (2020).

[44] Babak Salimi, Bill Howe, and Dan Suciu. “Data management for causal algorithmic fairness”. In: *arXiv preprint arXiv:1908.07924* (2019).

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

[46] Schrasing Tong and Lalana Kagal. “Investigating bias in image classification using model explanations”. In: *arXiv preprint arXiv:2012.05463* (2020).

[47] Sahil Verma and Julia Rubin. “Fairness definitions explained”. In: *Proceedings of the international workshop on software fairness*. 2018, pp. 1–7.

[48] Christina Wadsworth, Francesca Vera, and Chris Piech. “Achieving fairness through adversarial learning: an application to recidivism prediction”. In: *arXiv preprint arXiv:1807.00199* (2018).

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. “The caltech-ucsd birds-200-2011 dataset”. In: (2011).

[50] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. “Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 5310–5319.

[51] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. “Towards fairness in visual recognition: Effective strategies for bias mitigation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8919–8928.

[52] Eric Wong, Shibani Santurkar, and Aleksander Mądry. *Leveraging Sparse Linear Layers for Debuggable Deep Networks*. 2021. arXiv: 2105.04857 [cs.LG].

[53] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. “Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy”. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 547–558.

[54] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. “Situation Recognition: Visual Semantic Role Labeling for Image Understanding”. In: *Conference on Computer Vision and Pattern Recognition*. 2016.

[55] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. “Mitigating unwanted biases with adversarial learning”. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 335–340.

[56] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. “Maintaining discrimination and fairness in class incremental learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 13208–13217.

[57] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. “Men also like shopping: Reducing gender bias amplification using corpus-level constraints”. In: *arXiv preprint arXiv:1707.09457* (2017).

[58] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. “Places: A 10 million image database for scene recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), pp. 1452–1464.

[59] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2 (2005), pp. 301–320.

Appendix A

Implementation

This section covers some of the implementation details of the thesis.

A.1 Data preprocessing

A.1.1 Doctor-Nurse

The preprocessing of the Doctor-Nurse dataset was relatively straightforward. We initially grouped the samples according to gender. To balance the dataset, we removed samples from overrepresented classes until each class and gender combination contained an equal number of samples, 262. This approach was chosen to maintain the originality of the dataset by avoiding duplication of underrepresented samples. Then, we performed a train-test split of 75% and 25%, resulting in 196 training samples and 66 test samples for each gender and class combination.

A.1.2 ImSitu

The preprocessing of the imSitu dataset involved several key steps to transform metadata have a comprehensive dataset ready for the experiments. Here is for example the metadata for one image:


```
{
    "hitchhiking_238.jpg": {
        "frames": [
            {"place": "n03519981", "agent": "n10287213"},
            {"place": "n03519981", "agent": "n10287213"},
            {"place": "n04096066", "agent": "n10287213"}
```

```
        ],
        "verb": "hitchhiking"
    }
}
```

Every image had three different annotations from three different annotators. Even if the role were the same, the value could be different. The values were also IDs instead of semantic values. Here is how we preprocessed the dataset

1. **Mapping IDs to Values**: Each image annotations under the form of IDs were mapped to their actual values using a predefined mapping.
2. **Filtering the gender**: We selected the most common agent among the three annotations, and kept the image if the agent revealed the gender. If the agent was 'a male' or 'a man', the image was labeled as male. Conversely, if the agent was 'a female' or 'a woman', the image was labeld as female
3. **Generating concepts from annotations**: For every image, we aggregated the annotations (excluding the 'agent' annotation) into a set. For each class, the count of each annotation was computed. If a annotation had more than 15 counts, it was retained for the CheatBM experiment.
4. **Aggregating sets:** The sets from the previous step for all three classes (train, validation, test) were concatenated to form the complete dataset with gender annotations and extra possible concepts.
5. **Selecting classes:** We took the 200 classes with the highest n_min, which are between 39 and 126, to not have a too strong gender imbalance.

By transforming the dataset in this manner, we successfully generated a comprehensive dataset, completed with gender annotations and additional potential concepts, which was subsequently used in the experiments described in the Design section (see Section 3). When selecting the 200 classes with the highest n_{min}, the dataset contains in total 33123 samples: 17552 samples with male, and 15571 samples with female.

A.1.3 MS-COCO

The preprocessing stage of MS-COCO consisted in a series of steps to ensure the data was adequatly prepared for the experiments:

1. **Gender Classifications from Captions:** There are 5 captions for every image. We selected image having "a person" as a target, and searched for some keywords in the caption. If a caption contained the words "man" or "male," the image was labeled as male. Conversely, if the caption had "woman" or "female," the image was labeled as female. Image ambiguously associated with both genders or had no gender associations were removed.

2. **Merging Train and validation Sets:** The training and validation images were combined to create a full dataset, from which we performed the different experiments.

3. **Filtering targets:** We removed the target 'a person' as every images in our refined dataset contained it. For each remaining target, we computed $n_{min,c}$. Any targets with $n_{min,c} \leq 200$ were filtered out, while we ceiled the maximum number of samples per class in the datast to 1000. This reduced the number of targets from 79 to 42

4. **Removing images with no targets**: Images that had no targets from the selected 42 were to be removed. In our case, no such images were found.

At the conclusion of the preprocessing stage, the dataset consisted of 42 targets and a total of 28,300 images. The original dataset before filtering contained 39505 samples with male, and 23046 samples with female.

A.2 Model

A.2.1 Fine-Tuning the model

In the imSitu dataset, there were several hyperparameters to consider for the Concept Bottleneck Model (CBM). These included the 'clip_cutoff' for selecting concepts activating CLIP during step 1d of LF-CBM, the 'interpretability cutoff' for selecting interpretable concepts during step 1e of LF-CBM, and the 'lambda' value defining the sparsity of the last layer (see Equation 2.6). These hyperparameters significantly influenced the number of concepts and sparsity. For instance, in scenarios where the cutoff values were excessively stringent, all concepts were filtered out. Additionally, for the ResNet baseline, we had to fine-tune hyperparameters such as the learning rate of the Adam Optimizer and the step size and gamma value of the learning rate scheduler.

We conducted a grid search across these hyperparameters to identify the combination that maximized Accuracy on the validation set. These optimal hyperparameters were utilized to compute the results on the testing set, a process that was carried out independently for each CBM and baseline model. The various hyperparameters tested in the grid search, along with the selected hyperparameters for each experiment, are detailed in Appendix D.2.

We observed a common pattern across almost all CBM: no concepts were filtered during the final two phases of concept filtering (1d, 1e), with both the 'clip_cutoff' and 'interpretability_cutoff' being zero or close to zero. The sparsity metric 'lambda' was always 0.0007, with a relatively low density of 4% from this extensive set of concepts. This combination yielded the best results, as demonstrated in Table 4.4. The model do not select the concepts in the projection step from the activation values, but perform feature selection in the last layer.

A.2.2 Protecting gender concepts:

To ensure the gender attributes were not inadvertently removed by the LF-CBM workflow during the 'fairness through awareness' experiment, we introduced a feature called 'protected concepts'. This feature allowed us to designate certain concepts as 'protected' when running the Python file, thereby excluding them from deletion during the interpretability and clip activation selection processes.

A.3 MS-COCO dataset manipulation

Manipulating the MS-COCO dataset proved to be more challenging than imSitu and Doctor-Nurse datasets. For the latter two, we applied undersampling to our train split to generate different training datasets with different bias, as each image only had one class. However, for MS-COCO, an image could be associated with multiple classes, rendering the undersampling technique unsuitable.

We developed our own solution, masking both prediction and ground truth for the MS-COCO dataset to address this challenge. After partitioning the data into train, validation, and test split, we implemented a quota mask system to achieve the desired class distribution. Each class had a separate male and female quotas: for example, the quotas were $n_{min,c}$ for both genders in the validation and test datasets, while they were 0 for male and $n_{min,c}$ for female when the training set was Male-A and the class was part of group B.

During training, the DataLoader checked the quotas for the image gender each time an image was used. If the quota for any class c was reached, then the model would not learn from class c even if the image included that class; we set both the expected and predicted value to 0, removing any backpropagation from the class. We then decremented the quotas by 1 for each class present in the image sample. At the end of each epoch, the DataLoader quotas were reset.

During validation or testing, when the quota was exhausted, we set both the model prediction and the target index to 0, resulting in the class being counted as a True Negative.

Appendix B

Label-Free CBM concepts prompts

The prompts used to generate concepts in the Label-Free CBM framework through GPT-3 are:

- List the most important features for recognizing something as class:
- List the things most commonly seen around a class:
- Give superclasses for the word class:

Appendix C

Extra Figures

C.1 Background / Related work figures

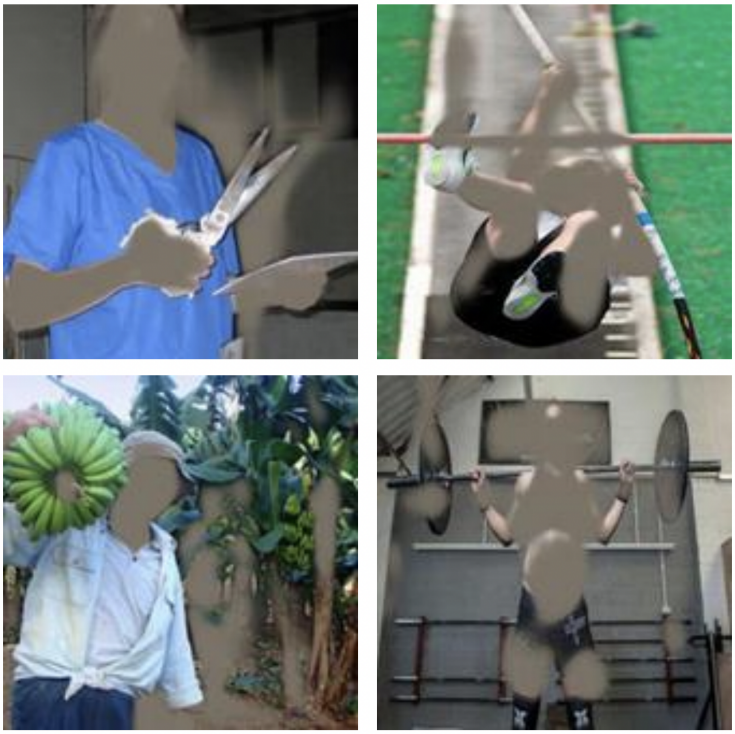


Figure C.1: This figure shows the effect of the 'Balanced datasets are not enough' technique. The adversarial model blurs out the gender characteristics of the human.

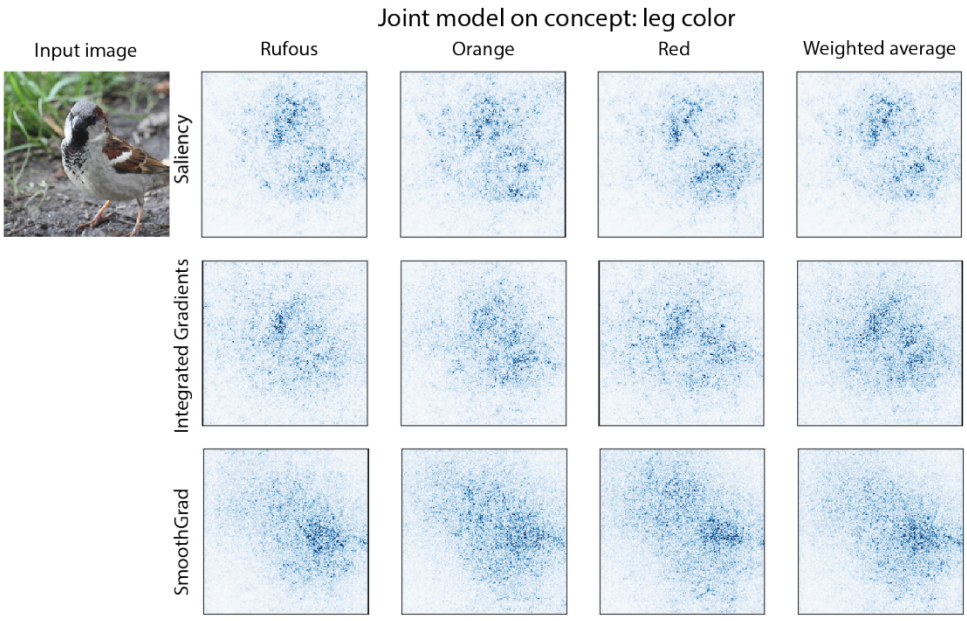


Figure C.2: This figure shows the saliency map associated with the concept "leg color". We can see that the model doesn't learn from the bird leg.

C.2 Design figures

C.2.1 Distribution of the difference between Male and Female count for each class

C.3 Results figures

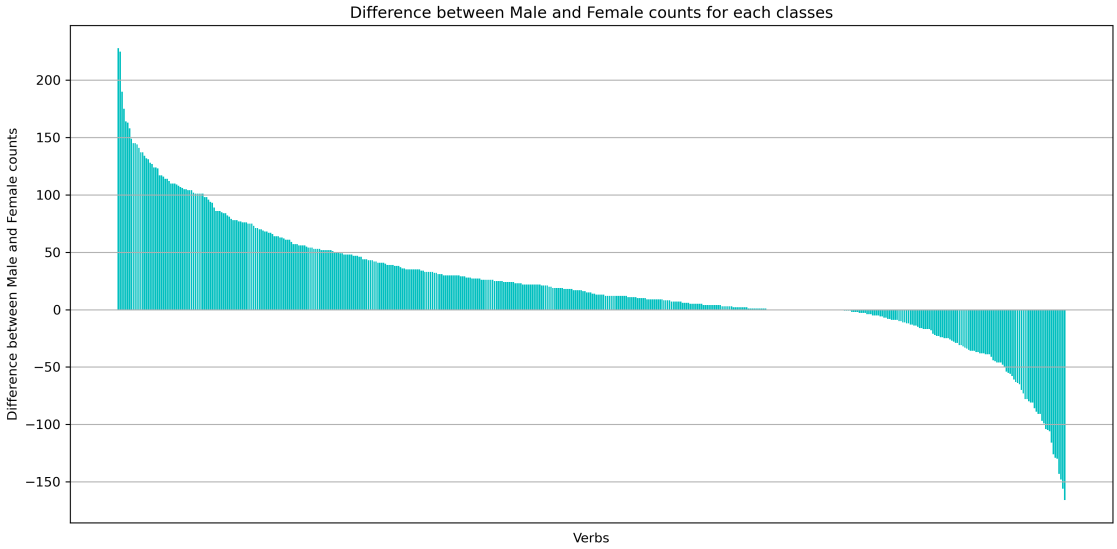


Figure C.3: Distribution of the difference between male and female counts for each of the 504 verbs. We note that the dataset is balanced towards male, as stated in previous work [57].

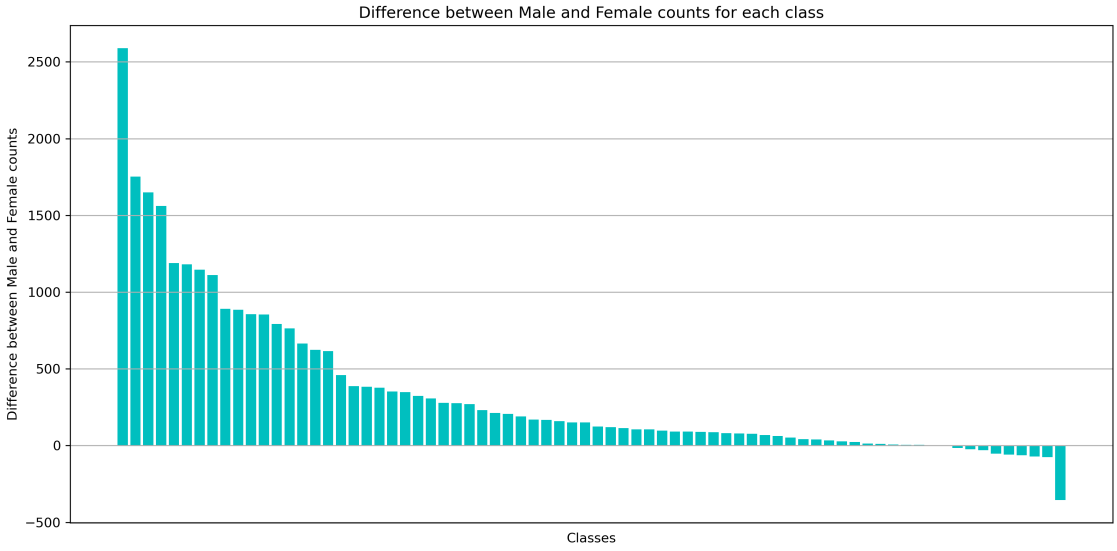


Figure C.4: Distribution of the difference between male and female counts for each of the 74 classes. We note that the dataset is balanced towards male, as stated in previous work [57].

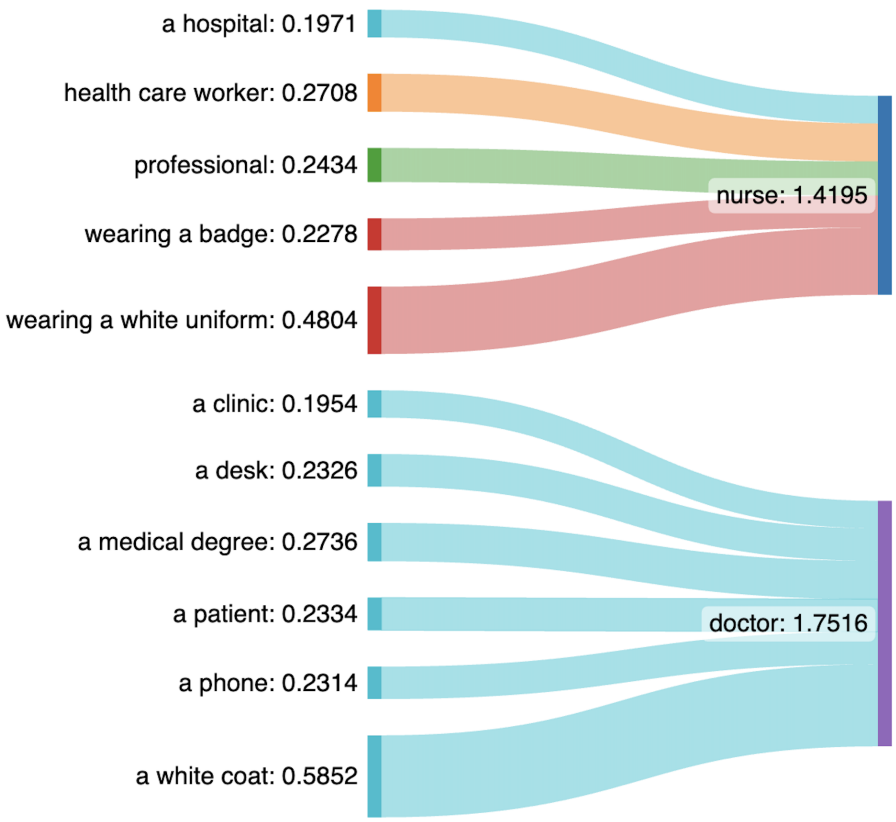


Figure C.5: Diagram of the Balanced CBM Doctor-Nurse Model, where only the concepts with the highest value are considered. We observe that some concepts, like the white coat for doctors and white uniform for nurses, are logical and help in differentiation. However, many concepts, such as a clinic, professional, phone, or patient, do not aid in differentiating the two professions. These concepts may be serving more as proxies rather than having semantic significance.

Appendix D

Detailed Results

D.1 Bias modification experiments

	Group A			Group B			Total
	Male	**Female**	**Total**	**Male**	**Female**	**Total**	
Baseline Full	32.95	31.51	32.30	27.84	32.67	30.12	31.26
CBM Full	32.74	33.65	33.15	27.48	32.67	29.92	31.62
CBM Full gender	33.05	33.52	33.27	27.78	32.93	30.21	31.81
CheatBM Full	34.10	34.46	34.26	27.60	33.07	30.18	32.32
Baseline Balanced	27.09	26.67	26.88	26.43	25.95	26.20	26.55
CBM Balanced	28.13	27.41	27.78	27.49	27.00	27.24	26.52
CBM Balanced Gender	27.84	27.86	27.85	27.65	27.16	27.41	27.63
CheatBM Balanced	28.36	27.93	28.14	27.81	27.32	27.57	27.87
Baseline Male-A	31.49	17.92	24.71	16.33	31.20	23.77	24.26
CBM Male-A	31.57	19.27	25.42	18.27	32.50	25.38	25.40
CBM Male-A Gender	31.57	19.27	25.42	18.03	32.66	25.34	25.38
CheatBM Male-A	30.22	18.22	24.23	18.51	32.98	25.75	24.96
Baseline Female-A	17.09	29.05	23.07	30.80	15.84	23.32	23.19
CBM Female-A	20.37	30.77	25.57	30.48	16.73	23.61	24.63
CBM Female-A Gender	20.22	30.84	25.53	30.72	16.73	23.73	24.67
CheatBM Female-A	20.07	30.40	25.23	31.61	16.41	24.00	24.65

Table D.1: ImSitu: Accuracy values for different classification types and different training datasets. We notice the overfitting of the model on the most prevalent gender in the training set.

	Group A			Group B			Total
	Male	**Female**	**Total**	**Male**	**Female**	**Total**	
Baseline Full	55.40	57.04	56.05	61.33	56.52	59.72	58.00
CBM Full	50.86	54.20	52.54	50.23	54.22	52.20	52.37
CBM Full gender	54.85	57.67	55.97	59.99	55.57	58.52	57.33
Baseline Balanced	45.73	51.97	48.86	49.57	49.33	49.45	49.15
CBM Balanced	50.86	54.60	52.75	51.24	53.93	52.59	52.67
CBM Balanced Gender	50.52	54.68	52.63	51.09	54.32	52.71	52.67
Baseline Male-A	46.55	49.95	48.25	47.47	49.06	48.25	48.25
CBM Male-A	50.58	54.12	52.36	51.71	52.39	52.05	52.21
CBM Male-A Gender	50.64	54.17	52.42	51.59	52.62	52.11	52.27
Baseline Female-A	44.76	50.57	47.62	47.70	49.74	48.69	48.15
CBM Female-A	50.86	54.20	52.53	50.23	54.22	52.20	52.37
CBM Female-A Gender	50.99	53.91	52.46	50.26	54.16	52.19	52.32

Table D.2: MS-COCO: F1-Score values for different classification types and different training datasets. We notice the overfitting of the model on the most prevalent gender in the training set.

D.2 Hyperparameters

For Doctor-Nurse, we used the hyperparameter from a previous work for the baseline. For the CBM hyperparameters, we decided to keep every concepts and to have a dense layer, as the concepts for doctor and nurse were similar.

For imSitu, we performed a grid search.

For MS-COCO, due to a lack of time, we couldn't perform a grid search. We tried empirically multiple hyperparameters combination until finding the one that yielded the highest performance on Validation. We selected 0.0007 for the sparsity as it was the value yielding the best tradeoff between a small amount of concepts and an high performance in Section 4.1.2.

D.2.1 imSitu: Grid Search Hyperparameters Tested

Baseline

- **Learning rate:** [0.001, 0.01, 0.1]
- **Step size:** [5, 7, 10]
- **Gamma:** [0.1, 0.5, 0.9]

CBM

- **Interpretability Cutoff:** [0, 0.1, 0.3, 0.5]
- **Lambda**: [0.00001, 0.0007, 0.007, 0.07]
- **Clip Cutoff:** [0, 0.1, 0.2, 0.3]

D.2.2 Hyperparameters Chosen

Doctor-Nurse

For the baseline, the learning rate is 0.001, the SGD momentum is 0.9, the stepLR step size is 7 and the stepLR gamma is 0.1.

The CBM hyperparameters for CBM were 0 for clip cutoff, 0 for interpretability cutoff, and 0.0007 for λ.

imSitu

	Learning Rate	Step Size	Gamma	Validation Accuracy
Baseline Full	0.001	5	0.1	26.19
Baseline Balanced	0.001	5	0.1	24.19
Baseline Male-A	0.001	5	0.1	23.21
Baseline Female-A	0.001	5	0.1	24.53

Table D.3: imSitu: Best hyperparameters and validation Accuracy for Baseline.

MS-COCO

Baseline hyperparameters are 0.01 for learning rate, and step size of 7 with gamma of 0.1 for the StepLR scheduler.

For CBM, the interpretability cutoff is 0.3, the clip cutoff 0.25, and λ is 0.0007.

	Interpretability Cutoff	Lambda	Clip Cutoff	Validation Accuracy
CBM Full	0	0.0007	0	31.37
CBM Full gender	0	0.0007	0	31.37
CBM Balanced	0	0.0007	0	28.82
CBM Balanced Gender	0.3	0.0007	0	28.84
CBM Male-A	0	0.0007	0	27.09
CBM Male-A Gender	0	0.0007	0	27.07
CBM Female-A	0.3	0.0007	0	25.29
CBM Female-A Gender	0.3	0.0007	0	25.35

Table D.4: imSitu: Best hyperparameters and validation Accuracy for every CBM. CheatBM was the same set of hyperparameters than CBM. We notice that the best result tends to have no concept filtering from the LF-CBM feature selection: the feature extraction is done by the sparsity of the last layer.