Title

Reliable Multimodal Models

Author

Petryk, Suzanne

Publication Date

2024

Reliable Multimodal Models

By

Suzanne Petryk

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Associate Professor Joseph E. Gonzalez, Co-chair
Professor Trevor Darrell, Co-chair
Professor Kate Saenko
Professor Jacob Steinhardt

Summer 2024

Reliable Multimodal Models

Abstract

Reliable Multimodal Models

by

Suzanne Petryk

Doctor of Philosophy in Computer Science

University of California, Berkeley

Associate Professor Joseph E. Gonzalez, Co-chair

Professor Trevor Darrell, Co-chair

Before deploying a machine learning model in a real application, it is important to ensure its reliability – this can take many forms, yet is broadly defined as operating without failure. For instance, an incorrect prediction from a model could have a myriad of negative downstream effects, especially if a user has placed trust in the model or if the error is consumed and propagated by automated agents. Multimodal models are growing in their capabilities and applications, yet research into the unique challenges they pose around reliability has been limited.

In this thesis, I cover my work towards improving reliability in the context of multimodal (vision + language) models. This is approached from three different axes: addressing visual biases via model explainability, learning better confidence estimates to abstain from answering questions with high uncertainty as well as reducing hallucinations in generated text, and investigating the contribution of language priors to caption error. In these works, I also present new evaluation frameworks that define particular areas of reliability. As machine learning models take a larger role in our society, carefully measuring and improving reliability becomes more important than ever.

Contents

Acknowledgments

It is strange to see the official single-authorship this thesis has, as it would not be even the slightest mark of ink on a page without help from a great deal of others.

The greatest thanks to my thesis committee – Joseph Gonzalez, Trevor Darrell, Kate Saenko, and Jacob Steinhardt – for their valuable guidance and broader perspective on my research.

My advisors – Joseph (Joey) Gonzalez and Trevor Darrell – have been enormous sources of wisdom throughout these years. Joey has patiently supported my research based in computer vision, an area not typically explored by students of his research group. His instinct for research nevertheless has led to valuable advice time and time again, and I've been continually impressed by his ability to almost instantly understand the technical details of a project. His devotion to communication has encouraged me to think deeply about the presentation of results when writing papers and giving talks. He's been a voice of support and reason throughout the highs and lows of my PhD, especially needed during the many, many months of the pandemic. Trevor has shaped my journey through computer vision, understanding what areas are truly exciting independent of the level of broader attention they might currently have. He's helped get a paper over the finish line more than once, knowing what results are valuable to the community and how to frame them beyond the myopia I have had while deep in experiments. Both Joey and Trevor's down-to-earth mannerisms and openness to collaboration have made me yet another fortunate student in their line of co-advisees, and I cannot express my gratitude to them enough.

Anna (Anja) Rohrbach and Marcus Rohrbach have essentially been my de facto third and fourth advisors for large portions of my PhD. Anja started working with me through the trials and tribulations of my first paper, and I was lucky to continue working with her for others. She is bold in her research philosophies, injecting both creativity and effectiveness into every project she works on. I worked with Marcus during my time as a visiting researcher at Facebook/Meta. He greatly shaped the trajectory of my research, with the title of his initial call for collaborators in 2021 becoming the title of my thesis three years later! I have learned a great deal from his thorough research process and instinct for "likelihood to work" when judging ideas.

Kate Saenko has been a fantastic collaborator at Meta, consistently advising on technical details of experiments from her vast knowledge of the field. It has been a blast co-authoring papers with David Chan and his infectious optimism. I'm proud to have collaborated with Lisa Dunlap. I'm also glad to have learned from Alvin Wan, Sayna Ebrahimi, and Spencer Whitehead. Thanks to the undergrad/master's students that have co-authored papers with me: Anish Kachinthaya, Haodi Zou, Kevin Miao, Keyan Nasseri, Nick Jiang, Raghav Singh, and Vedaad Shakib.

I've been lucky to intern under Raghav Subramaniam and the rest of the team at SafelyYou – seeing computer vision truly improve the lives of people has provided motivation and grounding to my technical pursuits.

My peers have given me a much broader education on AI than I could ever have hoped to achieve alone, and have my gratitude, whom I have not mentioned already: Ajay, Daniel, Eyal, Justin, Kevin, Paras, Peter, Sarah, Shishir, Sukrit, and Xin within the RISE/Sky lab. I also thank the members of the BAIR vision labs (and adjacent), especially Ale, Anastasios, Antonio, Baifeng, Boyi, Brent, Chung Min, Colorado, Devin, Ethan, Frederik, Grace, Ilija, Justin, Kartik, Matt, Medhini, Neerja, Norman, Parsa, Patrick, Ritwik, Roei, Rudy, Sanjay, Seth, Sheng, Tete, Vickie, Vongani, & Yossi. Thanks to Amir & Jathushan for being great desk-neighbors – I apologize to the former for my insults and the latter for failing to grasp his deadpan humor. I am still floored when realizing how joyful and privileged it has been to be around this brilliant group of peers.

The friends I have made have supported me more than they know. Medhini has kept me sane through tough times and has been a fantastic roommate and friend. Whether it's a phone call, walk, living room discussion, concert hall or mountain summit – she's been an incredible source of support, trip planning, and puns. I'm always flattered when Lisa finds me funny, as I generally live for her approval. She's been a great friend ever since I forgave her for stealing my desk the first time we met. My conversations with Rudy are never dull and leave me inspired. I would thank Daniel, but once someone owns a Dyson vacuum, there is little more I can do to further improve their self-worth. Amir, thanks for the great memories; and don't worry, I won't immortalize the giant poster I put up about your bad review by mentioning it here. I am thankful and surprised at the many more whom I have somehow convinced into friendship: Ale, Alvin (despite his laugh), Brent, Chung Min, Joseph, Julia, Justin, Kartik, Kenny, Neerja, and others whom I have unceremoniously forgotten and beg forgiveness.

My deepest thanks to Matt, whose patience and positivity have kept me grounded, dinners have kept me fed, and sense of adventure has kept me waking up in tents and getting shocked while feeding goats. And finally, a thanks to my family – my sister Margie and brother Mattie (who had finally convinced me to pursue this PhD: "It's such an exciting time in AI, how could you not be a part of it?"). Thanks to my parents, who have built a great life for us after they moved here from Poland. They depict the right balance of being hardworking and crazy (as evidenced by their racing in the Ironman world championship), and inspire me to be better.

Chapter 1

Introduction

Many fields such as software engineering, aviation, or agriculture have clearly-defined criteria for product reliability. For instance, these fields use terms such as "test cases", "quality assurance", "service-level agreements", "predictive maintenance", and many more to describe criteria that need to be met when a product is shipped. Once machine learning is involved, however, reliability becomes harder to define and ensure. Machine learning models can output incorrect predictions for many possible reasons that are hard to anticipate, such as an unexpected bias in training data or an out-of-domain input. In addition, a model may even output an incorrect prediction with high confidence – in other words, it may be "confidently wrong", making it difficult to know if a model is correct in the first place. Nevertheless, we must still endeavor to understand the reliability of machine learning models before they are used in a real application.

Why reliability? Should not model developers simply aim to increase the accuracy of models as much as possible? While model performance should always remain a critical goal, it should not be the only goal. There are many scenarios, particularly high-risk ones, where validating the level of model performance is just as important. For example, when a doctor uses a model's prediction to aid decision-making, explaining the features that led to the prediction is crucial for determining how much trust to place in it. When a person with visual impairments uses a model as a navigation assistant in a busy environment, properly communicating the model's uncertainty is also important for the user to assess trust. When automated tools are used to assess an area for damage after an earthquake, regions of high uncertainty may require human verification to best allocate relief resources. In these cases, it's even possible that a user may prefer a more reliable model at the cost of a drop in accuracy, although these goals are not necessarily at odds. The development of such reliability techniques, including better uncertainty estimation, explainability, evaluation metrics, online monitoring, and more, should thus be a focus of research alongside model performance.

The title of this thesis is *Reliable Multimodal Models*; before going further, we must be specific in defining what these terms mean. One possible definition of *reliability* is that from statistics: *"the overall consistency of a measure"*[1]. By this definition, a model may output the same prediction regardless of the input (e.g., always output '0'): not a useful framing for our purposes. Instead, we adopt the definition of **reliability** based in engineering: *"the probability that a [system] will operate in a defined environment without failure"*[2]. The chapters in this thesis will define an environment they operate in, as well as what it means to "fail". In practice, failure might have a strict, binary definition, such as "operating costs must not exceed $50,000 USD". In this academic work, however, we instead define frameworks under which reliability could be measured, and leave out specific thresholds that should instead be defined by a downstream user. For example, in Chapter 3, our framework parameterizes the percentage of error that a user can tolerate, and in Chapter 4, we describe systems where the level of hallucinated words in generated text should be as low as possible. Next, we use the common definition of **multimodal models** to mean neural networks that take more than one form of data as input – for this work, we specifically focus on *vision and language models* (VLMs) that operate over both images and text.

1.1 Thesis outline

This thesis covers three forms of multimodal reliability: (1) addressing an unwanted visual bias in training data (Chapter 2), (2) learning confidence estimators for visual question answering (Chapter 3) and image captioning (Chapter 4), and (3) investigating the language bias that contributes to hallucination in image captions (Chapter 5).

Addressing visual bias. In Chapter 2, we address the problem of *spurious correlations* in image classification – when there is a feature of an image that is correlated with one class during training, yet does not always hold during testing. For instance, consider a binary bird classification task: if the training set consists of photos of species A in the forest and species B by a swamp, the model may learn to classify species based on background of the image instead of the birds themselves. This becomes a problem when faced with test images of species A in a swamp, as the model will then predict incorrectly due to the bias. To address this, we propose a method using a high-level language specification of the task (e.g., "birds") and a pretrained vision-language model to *ground* the task-relevant features in the training images. We guide an image classifier's attention towards these features during training, supervising with a loss based on a visual explainability technique. We show that these image classifiers outperform baselines that do not use knowledge of the high-level task, and outperform the pretrained vision-language model as well.

Learning confidence estimators. Chapters 3 and 4 both focus on improving estimates of model confidence.[3] The baseline confidence estimates are *softmax scores*, sometimes erroneously interpreted as the likelihood that a prediction is correct, since these scores are

[1] https://en.wikipedia.org/wiki/Reliability_(statistics)

[2] https://en.wikipedia.org/wiki/Reliability_engineering

[3] We use "confidence" and "uncertainty" interchangeably, up to a sign flip. Although there may be cases where a distinction is needed, we do not address such cases here.

prone to miscalibration and overconfidence [58]. We develop alternative confidence estimators, based on learning an additional head on a base model. We train confidence estimators on a validation set, supervised to predict the loss of a given output of the base model. Chapter 3 describes such a confidence estimator in the task of classification-based visual question answering (VQA). We propose a framework for *abstention* in VQA, where a model can abstain from providing an answer in case of high uncertainty, and we show that our learned confidences outperform other baselines for abstention. Chapter 4 extends the idea of learned confidences to autoregressive image captioning, where the confidences are produced at a token-level. We show that these token-level confidences can be used to reduce the amount of hallucinations in image captions.

Investigating caption hallucination via language bias. Recent vision-language models are composed of a strong pretrained image encoder, such as CLIP [142], paired with a strong pretrained large language model (LLM), such as Vicuna [24]. These two models are combined via multimodal adaptation layers, often a small fully-connected network. While having a stronger grasp of language (e.g., the ability to follow instructions in natural language), these models generate image captions with considerably higher rates of hallucination than prior models that do not use an LLM. In Chapter 5, we explore the hypothesis that *language priors* explain some of these hallucinations, as they follow patterns that seem likely under language yet are clearly not grounded in the image (this can be thought of as a form of "language bias"). We find that a confidence measure based on agreement with the language prior is a better predictor of hallucinations than baselines such as softmax score or entropy. We also develop a new method for densely annotating captions for hallucinations.

This thesis presents significant developments into the lens of multimodal reliability, approaching it from several axes: addressing unwanted visual biases in training data, learning better confidence estimates for flagging and fixing incorrect outputs, and investigating the source of error in generated text from VLMs. A key principle of this research is its simplicity and practicality: each of the methods presented does not require significant annotation efforts or modifications to a base model. This becomes especially important as *scale*, in terms of data and compute, is dominating the field of AI more than ever. Being able to improve reliability without retraining a base model can often be a necessity. Speaking broadly, simple methods are easier to implement at scale, and are often more stable and likely to work. Incorporating multimodal reliability into the models of the future can thus become both possible and effective.

Chapter 2

On Guiding Visual Attention with Language Specification

2.1 Introduction

When trained with limited or biased data, visual models often learn unwanted correlations. For example, consider building a classifier to distinguish two fine-grained categories of birds: "landbird" and "waterbird". The background features from their corresponding habitats such as forests or beaches might be highly or perfectly correlated with the numerical class labels. A baseline model may mistakenly learn the unintended "location" task instead of the actual task, and fail on examples of birds out of their usual habitat (Fig. 2.1). However, knowledge that the task is *about birds* can disambiguate what the model is meant to learn.

Previous work has considered incorporating knowledge of the task as *language specifications* in the form of class names or class descriptions which can directly serve as a prior over visual model parameters [142, 189]. Several zero-shot methods condition models on attribute labels [90, 121, 198] (e.g., beak shape or wing color) or class descriptions [94, 42, 141, 217] (e.g., from Wikipedia) to enable transfer to unseen classes. However, this relies on the language specification being class-discriminative – an assumption which does not hold for some real-world tasks where only high-level task specification is given (e.g., in Fig. 2.1, we may only know that this is a "bird" dataset, without the class names being provided or even existing yet). Additionally, simply conditioning on language embeddings may not prevent a model from attending to spurious correlations in biased datasets.

Even when language specifications *are* class-discriminative, such models will perform poorly when there is insufficient image and text data to learn a multimodal model for rare or

This chapter is based on joint work with Lisa Dunlap (as a co-first author), Keyan Nasseri, Joseph E. Gonzalez, Trevor Darrell, and Anna Rohrbach. It is presented much as it appeared in the CVPR 2022 proceedings.

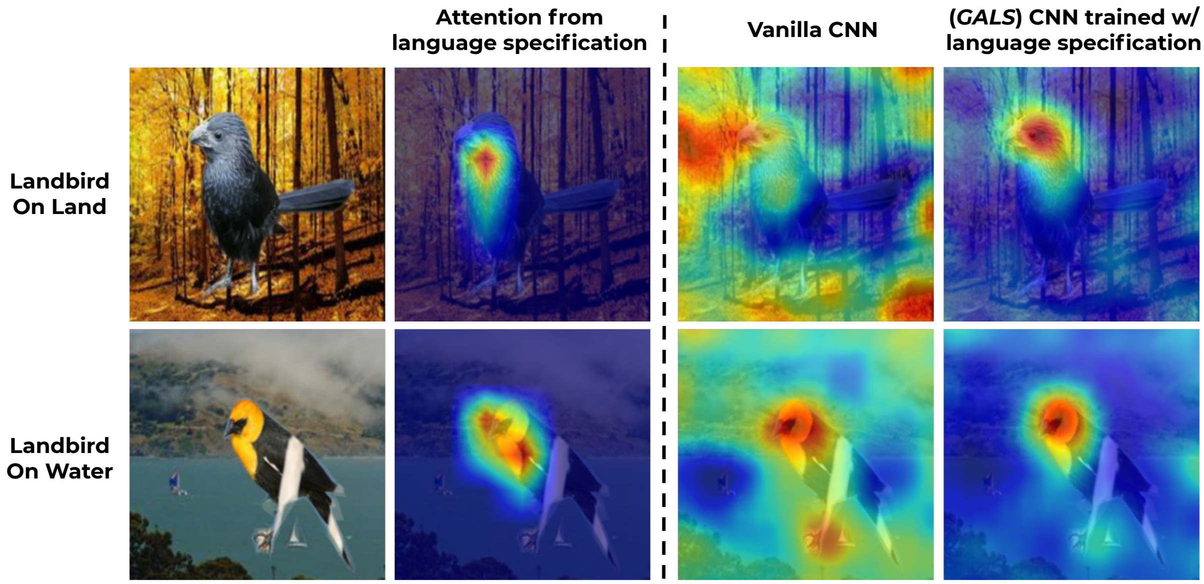


Figure 2.1: **Guiding attention with language.** Sample attention from the Waterbirds biased dataset. During training, Landbirds mostly appear on land backgrounds and Waterbirds mostly appear on water backgrounds. At testing, each class appears equally on land or on water. A CNN trained on this task learns to look at the background, but if we use a multimodal model to translate the language specification *"a photo of a bird"* into spatial supervision, we can ensure that our CNN learns task-relevant features.

fine-grained classes (e.g., a large-scale model such as CLIP [142] may not have seen enough examples of the relatively rare "landbird" or "waterbird" classes during pretraining to have good zero-shot performance).

To address these limitations, we propose a new framework called **Guiding visual Attention with Language Specification**, or ***GALS***, in which we translate available language specification provided by the task metadata into spatial attention that is used to supervise a CNN's attention during training. Fig. 2.1 displays how *GALS* is able to pull the model's attention away from the distractor features while retaining enough flexibility to pick up on fine-grained features which were not captured by the multimodal model.

Specifically, we first leverage an off-the-shelf pretrained vision-language model to *ground* textual information into each given image and obtain a respective saliency map. This is efficient and involves no additional overhead (i.e., no need for training or per-instance manual annotation). Next, we aim to leverage the obtained saliency map to inform the visual classifier. To do this, we *guide* the classifier's attention towards the area highlighted by the saliency from the language specification. Finally, the visual classifier still needs to solve the more fine-grained task, after obtaining the high-level attention guidance. It thus retains some flexibility, e.g. it may even attend to some useful (non-harmful) context. In practice, we use the recent powerful CLIP [142] model to ground textual information into images. We leverage the "Right for the Right Reasons" method [150] to enforce that the classifier indeed attends according to the given guidance. With this approach, we can incorporate language specification via an auxiliary loss during training, and thus the *vision-language model is not needed during inference.*

We show how *GALS* can assist in training on data with explicit and implicit bias. On the synthetic Waterbirds dataset [152] which contains a known, explicit bias (the image backgrounds), our method is able to achieve ∼2−7% per-group accuracy improvements over baselines, including a model which uses an unsupervised attention mechanism instead of guidance from language. *GALS* also shows a 15% improvement on the worst-group accuracy in the challenging scenario where class labels are perfectly correlated with the distracting backgrounds. For implicit bias, where training and test distributions differ in unknown ways, we see that *GALS* achieves ∼41−45% relative improvements on fairness metrics for apparent gender recognition. We also show a 2% accuracy improvement on a red-meat classification task from a subset of Food-101 [16], where an implicit bias emerges from noisy training labels. Lastly, we demonstrate that the quality of classifiers' explanations improves with the given advice (12.8% improvement in Pointing Game [208] accuracy). We provide our code and datasets to reproduce our experiments.

2.2 Related Work

Addressing bias with instance annotations. Most prior works that address bias in visual classifiers assume that some form of instance annotation is available. Some rely on expensive spatial annotations, such as object masks [63, 97, 147] or bounding boxes [27]. Hendricks et al.[63] address the image captioning task, where they want to reduce bias amplification and ensure a fair outcome for male and female genders by using person masks at training time. Others use slightly less expensive image-level annotations of the biased feature [2, 84, 152, 167, 186]. In contrast, in this work we do not assume that instance-level bias information is available. Instead, we rely on automatically generating attention guidance with readily-available language specification.

Addressing bias without instance annotations. Several works address bias without explicitly relying on instance-level bias annotations [31, 128, 175, 110]. Clark et al. [31] train an ensemble of low and high capacity models, forcing them to be conditionally independent, with the hope that the low capacity model will learn bias features, and the high capacity model will then learn the task-relevant features. Nam et al. [128] also train two models, one "biased" and the other "unbiased", by amplifying samples "aligned" with the bias for the first model (or easier to learn at the early stages), while amplifying the more difficult samples for the second model (where the first one fails). We view this line of research as complementary to our effort, and envision potentially combining these ideas with ours.

Language as information for visual tasks. We draw inspiration from prior work that leverages language in vision learning systems. Incorporating language in the zero/few-shot setting has been widely explored. Embedding language from class names or descriptions to obtain class "prototypes" is common in zero-shot learning, when no visual samples of the class are available [42, 141, 92, 44, 142]. Several works also aim to learn classes using their semantic attributes for better knowledge transfer [90, 121, 198]. Mu et al. [126] use image captions for regularizing few-shot representations to hold semantically meaningful information. Outside of zero/few-shot learning, Kim et al. [85] incorporate language advice into an autonomous driving controller, leading to a better performing and more explainable model. Rupprecht et al. [151] use language interactively to improve a pretrained CNN during inference time on semantic segmentation tasks. Ling et al. [107] use language feedback to improve an image captioning model. To the best of our knowledge, no works have explored using language specification to improve visual attention in biased scenarios.

Information grounding with vision-language models. One of the key components of our approach is to leverage an off-the-shelf vision-and-language model to ground textual information into an image. There is a large body of work on visual grounding, where the models are trained to localize textual expressions in an image with a bounding box [138, 148] or a segmentation mask [66]. Unfortunately, these methods are constrained by the cost of providing these extra labels for the training set. Others can handle more open-ended queries, but the size of the available training data is small as they require localization supervision which is costly to obtain, limiting the general application of these methods [81, 138]. A recent vision-and-language model CLIP [142] has demonstrated state-of-the-art image-text retrieval capabilities. CLIP is trained on 400M image-caption pairs sourced from the Web, making it a powerful general-purpose representation. We use CLIP and obtain grounding information with the help of salience visualization techniques [155].

Supervising spatial attention in visual classifiers. Another important component of our method is guiding the spatial attention within the visual classifier away from the spurious features. Several prior works have explored supervising spatial attention for, e.g., preventing catastrophic forgetting [41], fine-grained recognition [46], domain transfer [218] and generation of faithful explanations [150]. Specifically, the Right for Right Reasons approach [150] penalizes large input gradients in regions that are not allowed based on the user-defined "right reasons". We leverage this method to guide the classifier's attention towards the evidence pointed out by the language specification.

2.3 Guiding Attention with Language

In the following we outline *GALS*, our framework for incorporating language specification to guide a visual classifier; Fig. 2.2 provides an overview of our approach.

Problem Definition. In this work, we consider the learning problem in which we are given an image classification dataset $\{x_i, y_i\}_{i=1}^n$ for a prediction task $\mathcal{T}$ with C classes. Additionally, we assume we have a corresponding natural language specification $\mathcal{T}_s$ of the task or language descriptions of each class within the task $\mathcal{T}_s^c$. We also assume each image

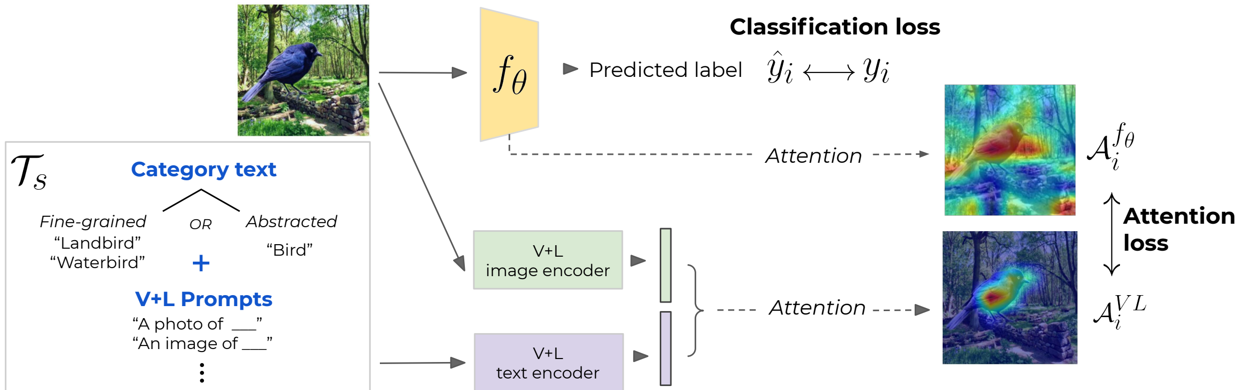


Figure 2.2: ***GALS* overview.** Our framework consists of three parts. First, we create a language specification $\mathcal{T}_s$ based on provided class names or a description of the task. Next, for every training image x_i, we use a pretrained vision and language model to ground the textual information into an image, in the form of an attention map A_i^{VL}. Finally, when we train the classifier f_θ, we incorporate A_i^{VL} as attention supervision. This encourages f_θ to align its attention $A_i^{f_\theta}$with task-relevant concepts, and away from distractors.

$x_i \in \mathbb{R}^{h\times w\times d}$ may contain a region of pixels that is irrelevant to $\mathcal{T}$, yet strongly correlated with y_i.

To model the distinction between the relevant and spuriously correlated pixels, we introduce a latent binary mask $\mathcal{Z}_i^{\mathcal{T}} \in \{0,1\}^{h\times w}$ for each image x_i, which encodes the relevance of each pixel to the task $\mathcal{T}$. That is, if $\mathcal{Z}_{i,(u,v)}^{\mathcal{T}} = 1$, then the value of pixel $x_{i,(u,v)}$ is informative for task $\mathcal{T}$ (and 0 if otherwise). Note that $\mathcal{Z}_i^{\mathcal{T}}$ is dependent on the prediction task. However, for notational convenience, we will omit $\mathcal{T}$ from $\mathcal{Z}_i^{\mathcal{T}}$ in the following.

Next, consider an image classification model f_θ with parameters θ. Our goal is to learn an optimal classifier f_{θ^*}, which outputs predictions $\hat{y}$ that rely only on task-relevant features (where $\mathcal{Z}_i = 1$). As $\mathcal{Z}_i$ is unobserved, we cannot learn f_{θ^*} by simply masking images according to locations of relevant features. Instead, we want to estimate a probability map over $\mathcal{Z}$, where each entry $x_{i,(u,v)}$ corresponds to the probability that pixel $x_{i,(u,v)}$ is relevant to $\mathcal{T}$.

Given this setup, our framework is three-fold: first, we create the high-level natural language specification $\mathcal{T}_s$ describing the semantic concepts relevant to $\mathcal{T}$. This is based on provided class names (e.g. "landbird") or description of the task (e.g. "bird species classification"). We then use a pretrained vision-language model and a spatial attention function to compute an estimate of the task attention $\mathcal{Z}$ for each image w.r.t. $\mathcal{T}_s$. Lastly, we use these estimates to supervise the spatial attention of f_θ, guiding it towards task-relevant features and away from unwanted biases.

Language specification. We assume access to natural language class names or a description of the task $\mathcal{T}$, but not necessarily access to what biases exist in the data. We argue that this is a safe assumption – in most real-world classification tasks, it is expected that a user has knowledge of what the categories are or what the task means. We then use the provided natural language to create $\mathcal{T}_s$ – words or phrases which are compatible with the choice of pretrained vision-language model, described below. For example, we preface task-relevant phrases with "a photo of" or "an image of" for compatibility with CLIP [142]. The language specification can be the same for each instance, or it can be class-specific by using the labels provided during training. Note that $\mathcal{T}_s$ is created once, prior to the training of f_θ, and does not require annotation of each image individually, allowing our framework to easily scale to large datasets.

Generating an estimate of $\mathcal{Z}$ from language specification. Consider a pretrained multimodal vision-and-language model VL, which has a joint understanding of image features and language phrases that correspond to them. For example, VL can be an image captioning or visual grounding model, or a model trained at scale with joint image-text supervision, such as OSCAR [101], VinVL [209], or CLIP [142], the latter of which we use in our experiments.

For every image x_i in the training dataset, we precompute a spatial attention map $\mathcal{A}_i^{VL} = Att^{VL}(\mathcal{T}_s^{y_i}, x_i)$, with $\mathcal{A}_i^{VL} \in \mathbb{R}^{h \times w}$. This serves as a probability map over $\mathcal{Z}_i$, where the attention value at location (u, v) estimates the likelihood that pixel $x_{i,(u,v)}$ is a task-relevant feature. The quality of $\mathcal{A}^{VL}$ as an estimate of $\mathcal{Z}$ depends on the ability of the pretrained vision and language model to ground text phrases in visual features. However, proper grounding within vision-and-language models is a research question on its own [109, 149]. Luckily, recent work on large-scale image-language pretraining has led to promising improvements [101, 209, 142]. Here, we use the saliency method GradCAM [155] to obtain reasonable attention maps.

Generating an estimate of the true task attention $\mathcal{Z}$ in this manner provides an automatic method for localizing per-instance, task-relevant features according to user specification $\mathcal{T}_s$. It requires only a high-level description of which semantic concepts are relevant to a task, which we view as a valid assumption for a user of a machine learning system.

Guiding the classifier with spatial attention. Next, for each image x_i, our objective is to guide the spatial attention of the classifier f_θ away from spurious correlations and towards task-relevant features. To do so, we would like to supervise the spatial attention of f_θ with the $\mathcal{T}_s$ attention maps $\mathcal{A}_i^{VL}$ computed in the previous step of our framework. This requires a function $Att^{f_\theta}(x_i, y_i)$, which computes a differentiable attention map $\mathcal{A}_i^{f_\theta}$. The attention map specifies spatial locations in x_i that were relevant to the prediction $\hat{y}_i$.

We supervise the classifier's attention for each training image x_i by computing a loss $\mathcal{L}_{att}$ between $\mathcal{A}_i^{VL}$ and $\mathcal{A}_i^{f_\theta}$. The final training loss $\mathcal{L}(\theta, X, y, \mathcal{A}^{VL})$ for a batch of training images X with m samples is given as:

$$\mathcal{L}(\theta, X, y, \mathcal{A}^{VL}) = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot log(\hat{y}_i) + \lambda \mathcal{L}_{att}(\mathcal{A}_i^{VL}, \mathcal{A}_i^{f_\theta}) \tag{2.1}$$

where λ is a hyperparameter that controls the strength of the attention supervision.

Our proposed framework does not require an architectural change to the classifier f_θ, and only incorporates language-guided spatial attention as an auxiliary loss term in Eq. (2.1) during training time. Therefore, our framework requires no additional knowledge at test time.

Model Design Choices

Vision-Language model. We use the CLIP (Constrastive Language-Image Pre-training model) [142] as our multimodal VL model. CLIP is trained on 400M image-caption pairs (x_{text}, x_{image}) sourced from the Web. It consists of two encoders for mapping x_{text} and x_{image} into a shared embedding space. The contrastive objective encourages image and text from the same pair to be close in the embedding space (as measured by cosine distance), and image and text from different pairs to be pushed apart.

Generating attention. For the language specification $\mathcal{T}_s$, we define a set of CLIP-style prompts. These are framed as short sentence descriptions, such as "an image of X" or "a photo with X," for the word or phrase X that describes task-relevant concepts. We generate multiple such prompts for each task and later combine (via average or max) the corresponding attention maps for each image, which serves as our estimate for $\mathcal{Z}$. Once the prompts are defined, they are embedded with CLIP's text encoder into the shared image-text latent space. For embedding images, we use the image encoder of CLIP with the ResNet50 backbone provided by Radford et al. [142]. For the attention function $Att^{VL}(\mathcal{T}_s, x_i)$, we use the saliency method GradCAM [155] between the image-text similarity score and the feature maps after the last convolutional block in the image encoder.

Attention incorporation. For supervising the classifier's attention, we adapt the framework Right for the Right Reasons, or RRR [150]. The original goal of RRR was to provide the correct explanation for each sample in addition to the correct prediction. This aligns well with our goal to prevent a model from learning unwanted feature correlations. In RRR, a user provides per-image binary masks of regions that are *irrelevant* to the task. It then penalizes the input gradients in those regions (the gradient of the output y with respect to the input x). In our work, the attention maps $\mathcal{A}_i^{VL}$ specify *relevant* regions to the task. Therefore, we take $1-\mathcal{A}_i^{VL}$ to specify *irrelevant* regions, and we compute the L1 loss between this and the input gradient. We normalize $\mathcal{A}_i^{VL}$ to contain values between 0 and 1 (instead of using a binary mask as in the original RRR method), as our intention is to estimate a probability map over the true task attention $\mathcal{Z}$.

Loss function. We apply GradCAM [155] to our chosen VL model (CLIP with a ResNet50 backbone), to provide $\mathcal{A}_i^{VL}$, the input gradients for a ResNet50 model pretrained on ImageNet as $\mathcal{A}_i^{f_\theta} = \frac{dy}{dX_i}$, and the RRR-based loss for $\mathcal{L}_{att}$. Thus, our loss function used in the experiments is:

$$\begin{aligned}\mathcal{L}(\theta, X, y, \mathcal{A}^{VL}) = & \overbrace{-\tfrac{1}{m}\sum_{i=1}^{m} y_i \cdot log(\hat{y}_i)}^{\text{Classification loss}} \\ & + \underbrace{\tfrac{\lambda}{m}\sum_{i=1}^{m}\left|\tfrac{dy}{dX_i}(1-\mathcal{A}_i^{VL})\right|}_{\text{Attention loss}}\end{aligned} \tag{2.2}$$

Our proposed framework is not restricted to a specific choice of pretrained model VL, classifier f_θ, mechanism of generating attention maps $\mathcal{A}^{VL}$ and $\mathcal{A}^{f_\theta}$, and attention loss function $\mathcal{L}_{att}$. *GALS* in the following section refers to the particular choices described above. We include ablations in Sec. 2.4 for other choices of VL, $\mathcal{A}^{f_\theta}$, and $\mathcal{L}_{att}$.

2.4 Experiments

Training. We use a ResNet50 [61] backbone pretrained on ImageNet for all classification models, with an input image resolution of $(224, 224)$. The GradCAM attention maps from CLIP are of size $(7, 7)$, which is the spatial resolution of the activations from the last convolutional block. We resize them up to the input resolution before computing the L1 loss. All error bars show standard deviation across 10 trials. We report further details on training parameters (such as the loss weight λ) and hyperparameter sweeps in the Appendix.

Baselines. We compare our work with several baselines that do not require per-instance knowledge of bias features. All baselines use the same ResNet50 backbone for consistency. *Vanilla* is trained in the same manner as f_θ in our framework, except without the attention loss $\mathcal{L}_{att}$. *UpWeight* is the same as Vanilla, except it uses class labels to address class imbalance. It computes a weighted average of per-sample cross entropy. The weights are inversely proportional to the frequency of the sample's class in the training data, assigning a weight of 1 to the class with fewest samples. *Attention Branch Network*, or *ABN* [46], learns a feed-forward attention map before the last convolutional block of ResNet50 and element-wise multiples it with the activations, which is added back into the activations before passing to the rest of the model. It also adds an additional cross-entropy loss term based on features in the attention branch, to encourage the spatial attention to be class-specific[1] We include tabular results of plots in the Appendix.

Visualizations. For all visualizations, the attention from language specification is generated with GradCAM (as in Sec. 2.3), and classifier attentions are generated with the black-box saliency method RISE [136]. More examples of attention for each dataset are in the Appendix.

Datasets

We evaluate our approach on datasets with explicit and implicit bias. Additional details are in the Appendix, including dataset size and creation. The license, PII, and consent details of each dataset are in the respective papers.

In the *explicit* bias setting, the distractor feature can be clearly defined and (potentially) labeled. We experiment with the synthetic Waterbirds dataset [152], where bias is easy to control. Specifically, the images of birds from the CUB dataset [181] are divided in two classes, landbirds and waterbirds. Next, birds are segmented out and pasted onto random land or water backgrounds from the Places dataset [213]. During training, most waterbirds appear on water backgrounds and landbirds on land backgrounds, while in validation/test

[1]We also experimented with supervising the attention of ABN with language specification. However, it under-performed the current formulation, and we include it in ablations in Tab. 2.3.

sets each class has an equal number of samples on land and water backgrounds. We consider two scenarios, one in which there is a small fraction of samples (5%) in the training data that go against the bias (***Waterbirds-95%***) and a more challenging one, where the bias and labels are perfectly correlated during training (***Waterbirds-100%***).

The Food-101 dataset [16] presents a case of *implicit* bias, as it was intentionally created such that the training images were not cleaned – for example, the images contain noise in the form of incorrect labels, bright colors, and visual confusion. Certain other foods appear more frequently with some classes than the others (e.g. sauce appears more often with baby back ribs than with steak). The evaluation set, on the other hand, was more thoroughly cleaned. We construct a 5-way ***Red Meat*** classification task between baby back ribs, filet mignon, pork chop, prime rib, and steak.

We present a second dataset with implicit bias, ***MSCOCO-ApparentGender***, which is constructed based on MSCOCO Captions [22] and prior work [63, 212]. In this dataset, apparent gender labels are defined based on the people's outward appearance as reflected in image captions. As defined in [63], when discussing people in captions, there are three options: "Man", "Woman" or a gender-neutral term, e.g. "Person". To follow that, we consider a three-way classification task for apparent gender, using the provided captions to generate labels for the classes "Man", "Woman", and "Person" (the latter when the annotators did not use gendered words in the captions). There are different types of spurious correlations in this dataset, e.g. women appearing in some environments more often than men, or a distractor object co-occurring with men but not with women, etc.

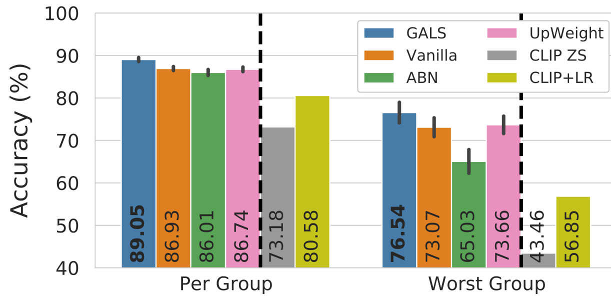
Explicit bias on Waterbirds

Since the Waterbirds dataset is constructed to encourage the model to pay attention to the background and not the bird, high-level language specification should give direction to attend to the bird, leaving the fine-grained discriminative image features up to the classifier to discover. Specifically, we generate attention from two CLIP prompts, to reduce noise – "an image of a bird" and "a photo of a bird." We average together these per-sample attentions to obtain $\mathcal{A}_i^{VL}$[2].

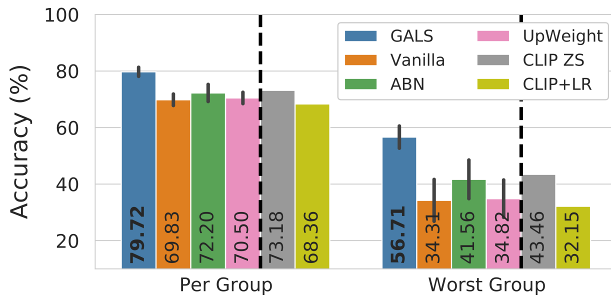
Following [152], we present test accuracy *per-group*, in which accuracy is weighted equally over the groups (specific combinations of class label and background, i.e. landbirds on land, landbirds on water, waterbirds on land, and waterbirds on water), and the *worst-group* accuracy. We are particularly interested in the worst-group (usually waterbird on land) performance, which suffers the most when a model makes use of the spurious background correlations.

The concepts "landbird" and "waterbird" are rare with respect to concepts that can be learned from the Web, as would often be the case in new, real-world classification tasks. To illustrate that large-scale models like CLIP may lack fine-grained task-specific knowledge, we

[2] In rare cases, the attention for a single prompt would be all-zero. Instead of averaging, we use the non-zero attention from the second prompt.



(a) Waterbirds 95%



(b) Waterbirds 100%

Figure 2.3: ***Waterbirds.*** Test accuracy on *Waterbirds-95%* and *Waterbirds-100%* datasets. Incorporating language specification results in a higher accuracy than all other baselines, including zero-shot CLIP and CLIP finetuned with logistic regression.

compare our method to zero-shot CLIP, as well as logistic regression trained on top of CLIP image-encoder features (following [142]). We find that CLIP often underperforms even the Vanilla baseline, demonstrating the value of taking the "best of both worlds" by combining large-scale multimodal model attention with CNNs on biased datasets with unfamiliar concepts.

Waterbirds-95%: As shown in Fig. 2.3a, our method outperforms all baselines on both per-group and worst-group accuracy. The strong bias in the data is evident when considering the worst-group accuracy, which drops the Vanilla performance by about 14%. Our model drives up the worst-group performance by 2.88% from the next-closest baseline of class weighting, without sacrificing per-group accuracy.

Waterbirds-100%: Because the class label and background are perfectly correlated, the performance of a classifier without any additional task information depends on whether it is easier to capture the true or bias signal. Surprisingly, the unsupervised attention mechanism in ABN provides ∼7% boost in worst-group performance as compared to upweighting by the class label. Our model improves on this, leading to a 15.15% improvement over ABN.

Using language specification to change the task

Since the "landbird" and "waterbird" labels in the *Waterbirds-100%* training set are perfectly correlated with land and water backgrounds, we can easily redefine the labels to create a background classification task. We investigate whether we can use language specification to choose which hypothesis a model learns during train time: the "bird" or the "background" task. To study this, we keep the training set the same, yet update the validation and test labels to reflect background classification. We use the phrases "nature scene", "outdoor scene", and "landscape", preceded with "a photo of" and "an image of" as in our other experiments. We ensemble the attention maps by taking the max value for each pixel. A vanilla ResNet50 baseline achieves 86.75% per-group accuracy on the test set, with a worst-group accuracy of 72.90%. Impressively, our method outperforms this baseline by **2.22%** and **7.32%** on per-group and worst-group accuracy respectively, demonstrating the flexibility of language specification to select the desired training signal.

Red Meat Classification with Noisy Data

Along with assisting in removing explicit contextual bias in datasets, in this experiment we will show how our approach can improve the learning process on implicit bias caused by noisy data. We train on 5 balanced classes from the Food-101 dataset pertaining to red meat, as discussed earlier. We generate attention from the CLIP prompts "an image of meat" and "a photo of meat". Our results displayed in Table 2.1 and visualized in Fig. 2.4 (top) show that our method is able to outperform the ABN model by $\sim 2\%$ overall.

Implicit bias on MSCOCO-ApparentGender

Next, we discuss how our approach performs in another implicit bias scenario on the *MSCOCO-ApparentGender* dataset. We follow the evaluation protocol from [63] and generate attention from the CLIP prompts "an image of a person" and "a photo of a person". Table 2.2 summarizes the quantitative results and Fig. 2.4 (bottom) displays a qualitative example of the attention maps. For each "Man" / "Woman" sample we separately report the % of the time they have been classified as a *Man*, *Woman*, or *Other*. We penalize gender misclassification, but do not penalize if the "Person" class was predicted. In this task, we care about several aspects. (1) The training data is imbalanced (with more men than women in it), thus we aim to reduce bias amplification at test time [63]. The metric "Ratio Delta" measures how close the

	GALS	**Vanilla**	**ABN** [46]
Accuracy (%)	$\mathbf{71.20 \pm 0.84}$	67.39 ± 0.88	69.44 ± 1.12

Table 2.1: ***Red Meat.*** Test accuracy of our method, vanilla, and ABN for *Red Meat* Classification (a subset of the Food-101 dataset).

	Man			Woman				
Method	*Man*	*Woman*	*Other*	*Woman*	*Man*	*Other*	**Ratio** Δ	**Outcome Divergence**
Vanilla	83.60	6.20	10.20	66.80	28.60	4.60	0.349	0.071
ABN [46]	**84.80**	**4.60**	10.60	68.80	25.40	5.80	0.339	0.068
UpWeight	80.20	11.20	8.60	68.00	28.60	3.40	0.272	0.040
GALS	79.80	11.80	8.40	**74.20**	**22.60**	3.20	**0.160**	**0.022**

Table 2.2: ***MSCOCO-ApparentGender.*** Performance of our approach and the baselines on *MSCOCO-ApparentGender* test set. The best result in each column is **bold**, and second-best is underlined.

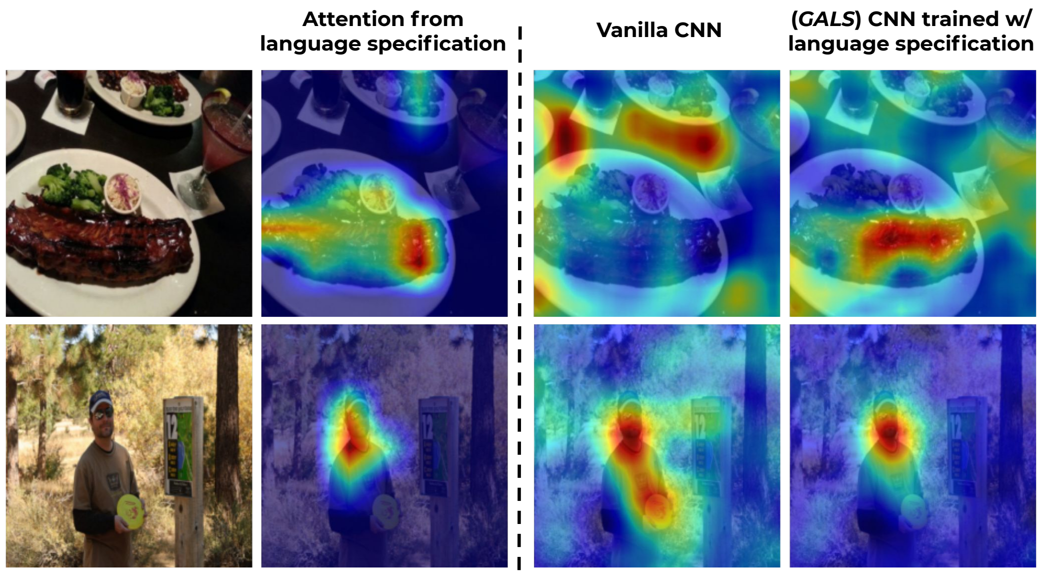


Figure 2.4: **Qualitative results for implicit bias.** Sample attention on *Red Meat* (top) and *MSCOCO-ApparentGender* (bottom). On these datasets the vanilla classifier may attend to non-task-relevant features due to implicit biases or noise. When we ground relevant features with language specification, we are able to move the classifier's attention away from the distractors.

predicted men/women ratio is to the true one (which is equal to 1.0), i.e. lower is better. Our approach performs the best in this metric. (2) We also aim to ensure an equal outcome for both men and women. In practice, we see that men tend to be recognized more accurately than women, as seen from the higher **Man**/*Man* values than the **Woman**/*Woman* values (e.g., the Vanilla baseline achieves 83.6% and 66.8% accuracy, respectively). As we see, women often get misclassified as men (22−28% across methods). The "Outcome Divergence" metric measures Jensen-Shannon divergence [104] between the two sets of scores across the two

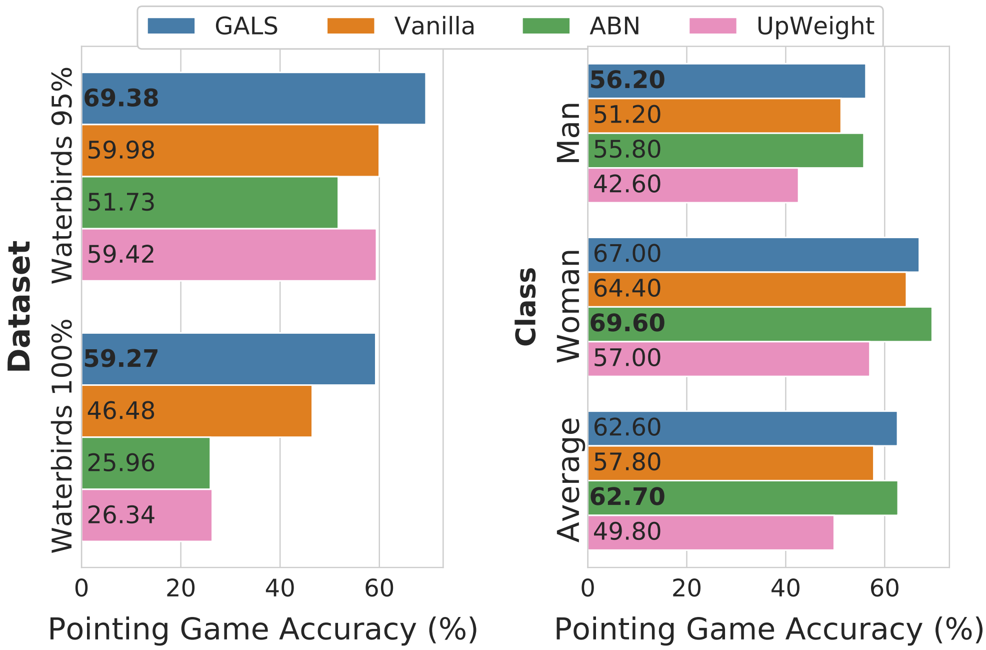


Figure 2.5: **Pointing game.** Pointing Game experiment [208] on *left.* Variants of Waterbirds datasets and *right.* MSCOCO-ApparentGender. We test whether the peak value of a black-box model explanation, generated with RISE [136], falls inside the segmentation label of the salient object.

classes, i.e. lower is better [63]. Again, our approach achieves the lowest outcome divergence, demonstrating the most fair behavior across all the compared methods.

Classifier Attention Method	**Language Attention Source**	**Validation Accuracy**		
		Per Class	**Landbird**	**Waterbird**
ABN	CLIP ViT	86.93	90.78	83.08
ABN	CLIP R50	86.10	90.25	81.95
GradCAM	CLIP ViT	87.20	<u>91.32</u>	83.08
GradCAM	CLIP R50	84.44	89.92	78.95
RRR	CLIP ViT	**88.25**	**92.28**	<u>84.21</u>
RRR	CLIP R50	<u>87.26</u>	89.17	**85.34**

(a) *Waterbirds-95%*

Cls. Att. Method	**Lang. Att. Source**	**Man**			**Woman**				
		Man	*Woman*	*Other*	*Woman*	*Man*	*Other*	**RΔ**	**OD**
ABN	CLIP ViT	84.40	10.60	5.00	68.40	29.40	2.20	<u>0.306</u>	**0.274**
ABN	CLIP R50	**90.60**	**5.40**	4.00	60.40	37.60	1.80	0.485	<u>0.280</u>
GradCAM	CLIP ViT	85.80	7.60	6.60	<u>70.20</u>	<u>27.00</u>	2.80	0.310	0.331
GradCAM	CLIP R50	83.40	<u>7.40</u>	9.20	66.20	29.80	4.00	0.311	0.298
RRR	CLIP ViT	<u>87.00</u>	8.40	4.60	68.60	29.80	1.60	0.341	0.305
RRR	CLIP R50	82.20	10.60	7.20	**72.20**	**26.00**	1.80	**0.235**	0.309

(b) *MSCOCO-ApparentGender*

Table 2.3: Comparison of different classifier attention methods and language attention sources on the (a) *Waterbirds 95%* and (b) *MSCOCO-ApparentGender* validation set. In (a), we report class instead of group scores, as we do not assume access to group labels at validation. The method indicated as "*GALS*" in Section 4 is placed at the bottom.

Attention Evaluation

We evaluate the quality of our model explanations to determine if language specification makes the model right for the right reasons in addition to improving accuracy. To do so, we use the Pointing Game [208], a common evaluation for model explanations. For each input x_i, the Pointing Game (PG) requires a corresponding model explanation a_i and binary mask $\mathcal{Z}_i$, both of the same dimensions as x_i. Recall that $\mathcal{Z}_i$ indicates the task-relevant pixels in an image. A model passes the PG on sample i if the maximum value of its explanation a_i falls inside $\mathcal{Z}$. In other words, the explanation is "pointing" to the correct region in the image.

For *Waterbirds-95%* and *Waterbirds-100%*, we use segmentation masks of the birds for $\mathcal{Z}$. On *MSCOCO-ApparentGender*, we use the available person segmentation masks, choosing the mask with the largest bounding box if multiple people are present to be consistent with our task. Segmentation masks for red meat in Food-101 are not available. For generating model explanations, we use the black-box saliency method RISE [136]. Fig. 2.5 presents our results. Our method matches the ABN baseline on *MSCOCO-ApparentGender*. However, we outperform all baselines by 9.4% on *Waterbirds-95%* and 12.8% on *Waterbirds-100%*.

Model Ablations

We explore several other design choices for the VL model and attention method in Tab. 2.3. We consider the Attention Branch Network (ABN) [46] as the classification model, while supervising its feed-forward attention map (similar to [125]). We also try supervising the GradCAM from the last convolutional layer of a ResNet50 classification model directly. For generating language specification, we experiment with the CLIP ViT-B/32 (CLIP ViT in the table). The method we denote as "*GALS*" corresponds to the row with RRR as the classifier attention method supervised with CLIP ResNet50 GradCAM attention. For both ABN and GradCAM classifier attention methods, we compute $\mathcal{L}_{att}$ as an L1 loss in a similar style as in RRR — penalizing A^{f_θ} where A^{VL} is low, as opposed to matching A^{f_θ} directly to A^{VL}, finding that this gives slightly better performance. We chose RRR+CLIP R50 since it had the most consistent performance in minority class accuracy and fairness.

2.5 Limitations and Broader Impacts

In this work, we focus on a scenario where a dataset bias during training time is not present at test time. This is an important issue with serious implications for high-risk domains such as autonomous driving or medical imaging. Generally, as machine learning methods become widespread and impact people's lives, reliance on biases may be harmful to entire populations. Thus, we envision potential positive impact from our work towards mitigating this issue.

One of the datasets used in this work (*MSCOCO-ApparentGender*) is derived from the image captioning MSCOCO-Bias and MSCOCO-Balanced splits introduced in [63]. Following [63], we consider three gender categories: male, female, and gender neutral (e.g., person) based on visual appearance. The gender labels were determined using a previously collected publicly released dataset in which annotators describe images [22]. Importantly, people in the images are not asked to identify their gender. Thus, we emphasize that we are not classifying biological sex or gender identity, but rather outward gender appearance. In particular, we are interested in reducing gender entanglement with contextual features and "equalizing" the outcome across male and female categories.

We also would like to point out that in our experiments, we use the off-the-shelf large-scale vision-language model (CLIP [142]) which may have encoded some internal biases, transferred from the data on which it was trained. Specifically, CLIP was trained on 400M image-caption pairs sourced from the Web, so we can not rule out the presence of biases or harmful (e.g. gender or racial) stereotypes in it. Practitioners who wish to use our approach should be mindful of such sources of bias.

As described in Sec. 2.3, our framework is limited to biases which can be pixel-wise separated from relevant features. As a counterexample, it would not apply to the task of classifying a person's age, with a confounding factor of race. Our framework also struggles when the vision and language model cannot ground the language specification (Fig. 2.6). In other scenarios, CLIP may struggle when the prompt is more compositional, such as "the person in the blue shirt sitting next to the table".

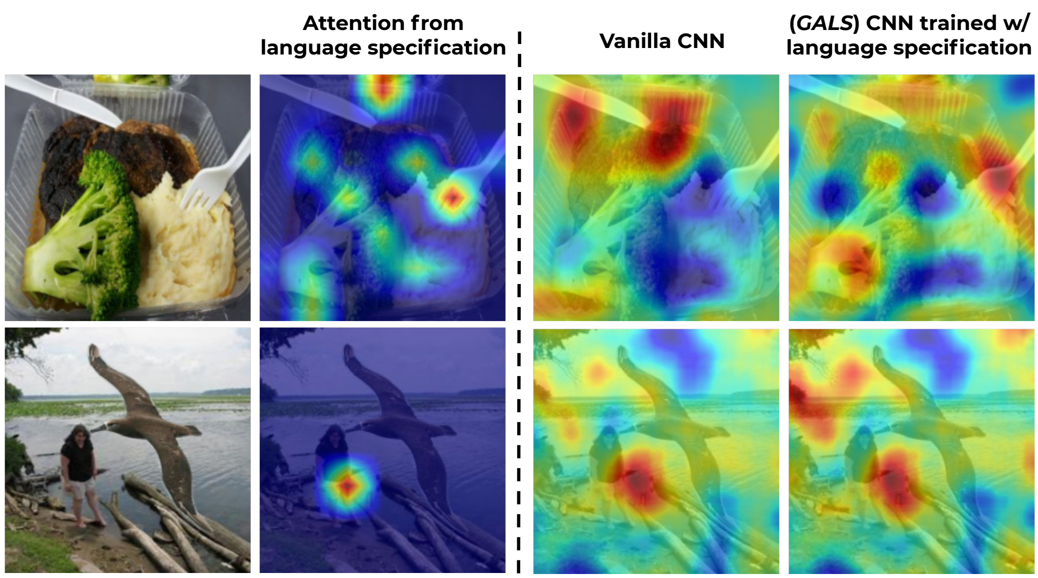


Figure 2.6: **Limitations.** Example of poor CLIP attention on *Red Meat* (top) and *Waterbirds* (bottom) dataset. Since *GALS* is supervised by the attention from language specification, our classifier's attention fails to ignore distractors when attention generated from language specification does not localize the task-relevant features.

2.6 Acknowledgements

This work was supported in part by DoD, including DARPA's LwLL, and/or SemaFor programs, and Berkeley Artificial Intelligence Research (BAIR) industrial alliance programs. In addition to NSF CISE Expeditions Award CCF-1730628, this research is supported by gifts from Amazon Web Services, Ant Group, Ericsson, Facebook, Futurewei, Google, Intel, Microsoft, Scotiabank, and VMware.

Chapter 3

Reliable Visual Question Answering: Abstain Rather Than Answer Incorrectly

3.1 Introduction

Visual Question Answering (VQA) is an important task and one core application of VQA is to provide a multimodal assistant, such as one that can answer questions to help with daily tasks for a user with visual impairments [10, 59]. To provide such utility, users must be able to trust the output of these tools as they may be basing decisions or actions on the output [11, 56, 118, 123]. While improving the accuracy of approaches may be an important factor for trusting models, models are imperfect and will inevitably produce some incorrect answers. In many scenarios, there is a price associated with a model giving an inaccurate answer as it may mislead the user and cause them to make a mistake that could be anywhere from mildly inconvenient to very serious. This is especially true for the example of helping users with visual impairments, since they likely do not have a method of verifying the outputs themselves.

One way to avoid providing incorrect information and misleading users is to *abstain* from making a prediction, as in the framework of selective prediction [29, 202, 48, 49]. Consider Fig. 3.1(a): when a model is correct, we naturally would like it to give us an answer. However, when it is unable to do so (e.g., cannot "read" the brand name) or is very uncertain, in many application we may prefer if the model communicated "*I don't know*", i.e., abstain [60, 82]. We say that VQA models are reliable, if they make highly accurate predictions when they choose to answer. Ideally, reliable models should also abstain as little as possible to be effective. Although reliability is often critical for the usage of VQA in real settings, this

This chapter is based on joint work with Spencer Whitehead (as a co-first author), Vedaad Shakib, Joseph E. Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach. It is presented much as it appeared in the ECCV 2022 proceedings.

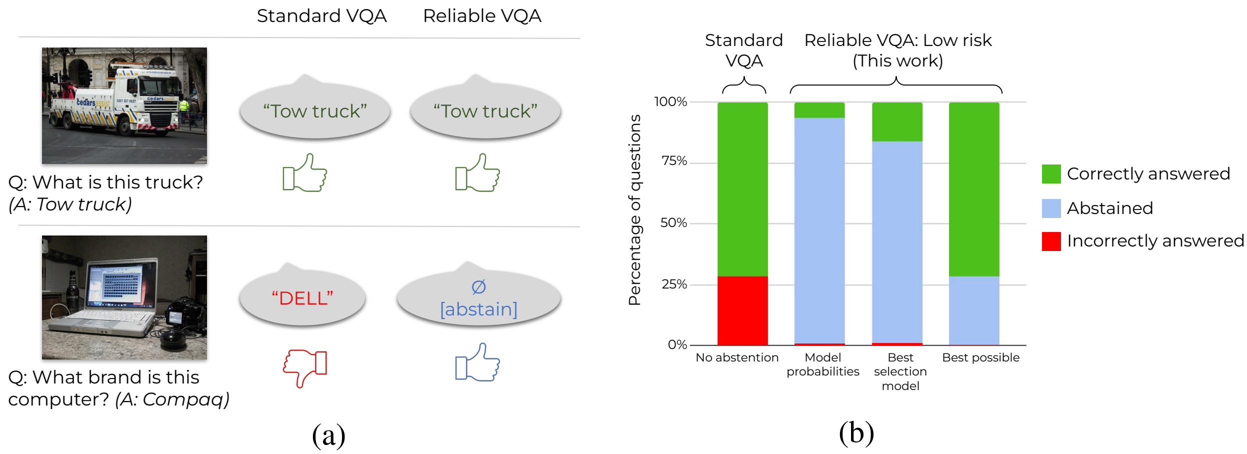


Figure 3.1: In the standard VQA problem, a model must answer all questions, even if it is likely to produce errors that could mislead a user, e.g., (a). A reliable VQA model, on the other hand, operates at *low risk* by having the option to abstain from answering if uncertain. In (b), at 1% risk of error, a SoTA model [160] can answer only ∼7% of questions when using vanilla model probabilities to choose when to abstain. Using a learned, multimodal selection function to estimate confidences can more than double the amount of questions answered, yet there remains much room for improvement (best possible, i.e., perfect abstention).

aspect has not received direct attention in the VQA literature aside from efforts to recognize difficult, unanswerable, or false premise questions [25, 59, 76, 143, 170]. Also, past efforts on selective prediction have not focused on the multimodal setting, where both an image and a question can be valid or in-distribution when considered independently, yet challenging in tandem.

In this work, we formalize and explore the notion of reliability in VQA. We propose to frame the task as a selective prediction problem [29, 202] in which models must either predict an answer or abstain from answering. This requires two techniques that have not been widely explored for VQA models: (1) gauging uncertainty of predictions and (2) learning when to abstain. To operationalize this framework, we measure performance with *coverage* (how many questions are answered) and *risk* (the error on these questions) [202, 78]. While low risk and high coverage are the goal, in practice there often is a trade-off between the two. To provide a scalar measure that captures this trade-off and allows for clearer model comparisons, we introduce a new *Effective Reliability* metric, which accounts for abstention while also introducing a cost for giving an incorrect answer. This also provides an alternative evaluation for domains where it may be more intuitive to specify the penalty for an individual error instead of a bound on risk.

Under this framework, we first show that existing VQA approaches leave much room for improvement. In particular, we demonstrate that, for a number of models, the common approach of using the maximum probability to determine abstention [64, 78] (by thresholding the softmax scores) limits the model to answering a small fraction of questions with a low risk of error (e.g., answering less than 7.5% of questions at 1% risk of error), despite having high standard VQA accuracy. This inability to answer a larger number of questions at low risk indicates low utility of the existing VQA models.

To address this, we explore two other approaches: calibration and training a multimodal selection function. We find that calibration often leads to a better risk-coverage trade-off compared to using the original model probabilities. We improve beyond this by training a multimodal selection function that can better learn to predict if a the model's answer is correct, based on intermediate representations as well as the answer from the VQA model. This selection function consistently improves the coverage of different VQA models across varying risks of error, particularly for low levels of risk. However, we show that there is still room to improve the effectiveness of these models (see Fig. 3.1(b)). Finally, we evaluate VQA models with our new Effective Reliability metric, and see that it correlates with risk and coverage in a meaningful way – the user-defined cost of an error impacts the risk at which the model operates.

In summary, our contributions are: (1) we are the first to analyze and operationalize reliability for multimodal VQA models; (2) we expose the issue of low coverage in VQA models when asked to operate at low risk levels; (3) we explore several methods for incorporating abstention, showing that a simple yet effective multimodal selection function outperforms other methods; (4) we propose a novel *Effective Reliability* metric for this problem, establishing a new benchmark for effective and reliable VQA models.

3.2 Related Work

VQA methods. VQA is a popular task with a plethora of methods proposed in recent years [5, 10, 23, 45, 47, 72, 73, 98, 101, 117, 160, 201, 206, 209]. To the best of our knowledge, there are no VQA models with a built-in abstention mechanism (i.e., they predict an answer for every image and question pair). We discuss a few exceptions with non-standard problem statements in the following. Our work analyzes VQA models' reliability by introducing the ability to abstain into several prominent VQA models [73, 98, 117, 160].

Detecting intrinsic difficulty. Some prior work on VQA involves the categorization and detection of questions that are intrinsically difficult to answer, regardless of model ability. For example, the VizWiz VQA dataset contains labels for questions which are unanswerable [59] and reasons for annotation entropy, such as low image quality or question ambiguity [12]. [34] define a similar categorization of unanswerable questions in VQA. [170] compute precision/recall based on VQA model confidences and show that these can be reflective of the ambiguities of the ground truth answers. Other work focuses on detecting whether the question incorrectly describes the visual semantics [76, 99, 120, 143]. Identifying intrinsically difficult examples has important implications in active learning, where such examples can stifle the ability of different methods to select useful examples to train on [79]. In this work, we focus on predicting uncertainty specific to a model as opposed to the intrinsic difficulty from data itself. However, in Sec. 3.5, we find that a subset of questions on which a model abstains from answering are ambiguous or unanswerable.

Calibration. In classification settings, calibration typically refers to probabilistic calibration, where the predicted confidence for a given class should be representative of the probability of the prediction being correct [58, 64, 89, 130, 137]. One popular parametric method is Platt scaling [137], in which a logistic regression model is trained on classifier outputs on the validation set to return calibrated probabilities. In our work, we explore the effectiveness of vector scaling, a multi-class extension of Platt scaling, for improving selective prediction performance.

Selective prediction. This refers to when models have the option to abstain from providing a prediction. It is also known as sample rejection [28, 29] or selective classification [202]. [35, 71, 174] propose various related evaluation metrics. [35] assigns cost coefficients to misclassified, abstained, and correctly classified samples. Concurrently with our work, [174] defines reliability as out-of-the-box performance for large-scale pretrained models across many unimodal vision or language tasks, including selective prediction. Other works integrate abstention in multi-stage networks or ensembles [15, 32, 83, 140, 188]. [75, 196] study selective prediction and transformer uncertainty within NLP tasks. [55, 78, 176] explore selective prediction performance on out-of-distribution data. [78] focuses on selective prediction for text-based question answering. However, they show that their method does not generalize to questions from the same domain which are intrinsically unanswerable, whereas this represents an important portion of difficult VQA samples. [48, 49] optimize selective models for specific coverage levels in image classification. We explore learned selection functions, but in the multimodal VQA setting, where the complex interaction between modalities must be modeled and more than one output may be considered correct to varying degrees. In the multimodal space, [63] addresses gender bias in image captioning, where the model can "abstain" by predicting gender-neutral words when it is uncertain. With our proposed metric, the cost of error (e.g., misclassifying gender) can be user-defined and potentially be made class-specific.

3.3 Visual Question Answering with Abstention

Visual question answering is currently formulated and evaluated in the literature [10, 52, 59, 69] as *always* predicting an answer from the answer space, $\mathcal{A}$, annotated in the dataset. So, a model $f : \mathcal{X} \mapsto \mathcal{A}$ predicts an answer $a \in \mathcal{A}$ for each input $x = (v, q) \in \mathcal{X}$, with image v and question q. This problem formulation forces the model to answer even if it is likely wrong, thus providing unreliable answers. To address this, we propose to extend the VQA problem formulation so that a model is given the option to *abstain* from answering a question (i.e., effectively saying "*I don't know*"). Outside VQA, this formulation has also been referred to as "*classification with a reject option*" [28, 35, 49, 60, 140] or "*selective prediction/classification*" [202, 48]. We first discuss the problem definition, and then the metrics to evaluate this problem.

Problem Definition

We extend the standard VQA formulation to the setting where a model can either provide an answer from $\mathcal{A}$ or choose to abstain (denoted by $\emptyset$): $h : \mathcal{X} \mapsto \mathcal{A} \cup \{\emptyset\}$. We refer to h as a *selective model*.

One way to formulate and achieve this is by decomposing h into two functions, f and g, which jointly comprise a selective model [202, 48, 49]. f denotes the VQA model that predicts answers and $g : \mathcal{X} \mapsto \{0, 1\}$ is the selection function that determines whether the model answers or abstains from answering:

$$h(x) = (f, g)(x) = \begin{cases} f(x) & \text{if } g(x) = 1, \\ \emptyset & \text{if } g(x) = 0. \end{cases} \tag{3.1}$$

Given an input x, the selective model yields an output from f when the selection function predicts that an answer should be given, or abstains if the selection function predicts that the model should not answer. One straightforward way to formulate the selection function g is based on a threshold γ, where the function $g' : \mathcal{X} \mapsto [0, 1]$ predicts a confidence in the correctness[1] of the model $f(x)$ [78]:

$$g(x) = \begin{cases} 1 & \text{if } g'(x) \geq \gamma, \\ 0 & \text{if } g'(x) < \gamma. \end{cases} \tag{3.2}$$

In general, a good function $g'(x)$ for abstention should yield high values when $f(x)$ is correct and low values when it is incorrect. In Sec. 3.4, we will further discuss how to define $g'(x)$.

Evaluation Metrics

To evaluate a VQA model with an ability to abstain, we consider two types of evaluation and discuss how we adapt them for VQA: first, *coverage* and *risk* [202] and, second, a cost-based metric for balancing the two.

Risk and Coverage. *Coverage* is the portion of questions that the model opted to answer, while *risk* is the error on that portion of questions [202]. Ideally, a reliable model should exhibit high coverage at low levels of risk, meaning it answers many questions with high accuracy and abstains on others. Concretely, coverage for dataset $\mathcal{D}$ with inputs x_i and ground truth answers y_i is given by:

$$\mathcal{C}(g) = \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} g(x_i), \tag{3.3}$$

and risk is defined as:

$$\mathcal{R}(f, g) = \frac{\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} \ell(f(x_i), y_i) \cdot g(x_i)}{\mathcal{C}(g)}, \tag{3.4}$$

[1]While we define the output space of g' as $[0, 1]$ as is the case for the common softmax, one can similarly define an output space which covers, e.g., all real values $\mathbb{R}$.

where ℓ is a cost function that measures the error between the predicted answer $f(x_i)$ and the corresponding ground truth answer y_i. Assuming g follows Eq. 3.2, if the threshold γ decreases, coverage will increase, but risk will increase as well. Hence, there is a risk-coverage trade-off that models can aim to optimize.

Applying this to VQA, the composite function (f, g) becomes our selective VQA model, where f produces an answer and g decides whether to abstain. However, the open-ended nature of the VQA task requires careful consideration for designing the risk-coverage metrics. A given question might have multiple possible answers which could all be considered correct to varying degrees. As a result, the error for a prediction on a given input is not necessarily binary.

When calculating risk, we must use a cost function that accurately represents this multi-class nature. We follow [10] to define VQA accuracy for a given model answer $f(x)$ as $Acc(f(x), y) = \min\left(\frac{\#\text{ annotations that match } f(x)}{3}, 1\right)$ and average these accuracies over all 10 choose 9 subsets of human annotated answers for the input question, similar to other VQA evaluations [52, 59, 166]. Under this, an answer is considered fully correct if it matches at least four of the human annotations, and receives partial credit for predicting an answer with one, two, or three humans in agreement. Thus, our risk measurement becomes:

$$\mathcal{R}(f, g) = \frac{\frac{1}{|\mathcal{D}|}\sum_{(x_i,y_i)\in\mathcal{D}}(1 - Acc(f(x_i), y_i)) \cdot g(x_i)}{\mathcal{C}(g)}. \tag{3.5}$$

In practice, the level of risk in model predictions that a user is willing to tolerate depends highly on the scenario. Therefore, we evaluate by computing coverage at a range of risk levels ($\mathcal{C}@\mathcal{R}$), such as coverage at 1% or 10% risk. We can also summarize this over the distribution of risk levels by plotting coverage versus corresponding risk, and computing the area under this risk-coverage curve (AUC) [78]. Moreover, for an evaluation that controls for how the threshold γ for g is chosen, we compute the maximum coverage for each risk level, allowing for a more direct comparison of the selection function design.

Effective Reliability. Recall the trade-off between risk and coverage: a standard VQA model may have high risk at 100% coverage, but a reliable model may have low risk yet abstain on a large portion of questions (see Fig. 3.1(b)). In practice, for a model to be reliable and effective, it should ideally achieve both low risk and high coverage. To jointly measure these two desirable qualities, we define a metric which assigns a reward to questions that are answered correctly, a penalty to those answered entirely incorrectly, and zero reward to those abstained on. We refer to this as *Effective Reliability*, or Φ_c for a given penalty c, inspired by the "effectiveness function" introduced by [35].

Formally, we define Effective Reliability for an input x as $\Phi_c(x)$ (Eq. 3.6), where c is the cost for answering incorrectly, g is the selection function, and Acc is a measure of a model's correctness. In this case, Acc is the VQA accuracy [10].

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ -c & \text{if } g(x) = 1 \text{ and } Acc(x) = 0, \\ 0 & \text{if } g(x) = 0. \end{cases} \tag{3.6}$$

We define the total score $\Phi_c = \frac{1}{n}\sum_x \Phi_c(x)$, a mean over all n samples x. This formulation assigns a reward to answers which are at least partially correct (i.e., $Acc(x) > 0$) – an important property of the VQA accuracy, where the correctness of answers can vary based on the number of human annotators in agreement. The choice of c depends on the deployment-specific cost of providing an incorrect answer. In Sec. 3.5, we report Φ_c with cost values of 1, 10, and 100 (Φ_1, Φ_{10}, Φ_{100}). While [35] suggest setting $\Phi_c(x) < 0$ for $g(x) = 0$, we set $\Phi_c(x) = 0$ (i.e., a score of 0 when abstaining). This enables our formulation to have the clear upper bound for models which abstain perfectly (Lemma 1). We provide a simple proof for this in Appendix B.11. It is also confirmed in our experiments in Tab. 3.2.

Lemma 1. *The Effective Reliability score is equal to the VQA Accuracy ($\Phi_c(x) = Acc(x)$) if a model abstains ($g(x) = 0$)* iff *it is incorrect ($Acc(x) = 0$).*

In our experiments, we choose a threshold γ which optimizes Φ_c on a validation set to compute a model's Effective Reliability with the form of the selection function g defined in Eq. 3.2. Additionally, the Effective Reliability score Φ_c can be evaluated for any model, even those which do not incorporate the option to abstain from providing a prediction (i.e., $g(x)$ is always 1).

Beyond its connection to VQA Accuracy (Lemma 1), Effective Reliability has several other advantages. We show that it meaningfully correlates with risk-coverage (Tab. 3.2), yet provides a single metric to compare models. This offers simpler comparisons that can be used to rank approaches (e.g., evaluating on a challenge server). It also provides an alternative evaluation for settings where it may be easier or more intuitive to define a cost for an incorrect answer as opposed to a target level of risk.

3.4 Selection Functions

We investigate three promising directions to extend VQA models to abstain by exploring different options for $g'(x)$ introduced in Sec. 3.3. Additional implementation details for the selection functions can be found in Appendix B.9.

MaxProb. Without any additional training, a model can be extended to abstain by defining g' as the softmax probability of the model's predicted class (i.e., maximum probability) and is thus refered to as MaxProb [64, 78, 89]. Essentially, MaxProb trusts that if the model gives a high probability to one class, it is quite certain that the answer is correct and should be given: $g'_{\text{MaxProb}}(x) = \max(f'(x))$, where $f'(x)$ represents the answer probabilities.

Calibration. Calibration techniques tune the absolute confidence values [137] to make the predicted probability for an output representative of the likelihood of that output being correct. Selective prediction has more to do with relative confidence rankings [202], but, nevertheless, a poorly calibrated model might also imply poor confidence rankings [78]. Temperature scaling [58, 137] is a popular calibration method, but it does not change the confidence rankings between examples and has no effect on the risk-coverage curve. Thus, we do not consider it in this work, but instead use vector scaling [58, 137] to calibrate the model logits. We then apply MaxProb on top of these calibrated logits. Appendix B.7 has evaluations of how well the scores are calibrated.

Multimodal selection function: Selector. Vector scaling essentially trains an additional component on top of the VQA model to refine the model confidences. We move beyond this by training a component (Selector) to predict whether the answer is correct [40, 78, 137]. Different from prior work on confidence estimation in other tasks [40, 49, 78, 188], the multimodal nature of VQA presents unique challenges where the model must consider the interaction between the image, question, and answer. To model this, we extract the image v, question q, multimodal r, and answer $f'(x)$ representations from the VQA model and input these to the Selector, which gives it access to representations of both the answer itself as well as the evidence on which the answer is based. The Selector is a multi-layered perceptron that takes these representations as input and predicts the correctness of an answer with respect to the image-question pair. To train this component, the simplest method may be to treat this as a binary classification problem (correct or incorrect). However, this does not account for answers that may be partially correct, or where one answer may be more correct than another, as is the case with VQA. Therefore, we propose to treat correctness prediction as a regression task where the target value is the VQA accuracy, allowing us to scale confidence scores with correctness.

3.5 Experiments

Data and Models

We experiment on the VQA v2 dataset [52] and require annotations for evaluation. As annotations for the test-dev and test-std sets of VQA v2 are not publicly available, we use questions from the official validation split for our evaluation as is common [3, 157, 193]. As a reminder, under our selective prediction setup, the VQA model is the function f, the selection function is g, and the composition of the two form a selective model h. We train the VQA models (f) on the training set of VQA v2. Meanwhile, we split the 214k examples in the VQA v2 validation set into three subsets: a split with 86k examples (40%) for validating VQA models as well as training selection functions (g), another with 22k examples (10%) for validating the selection functions, and a held out test split of 106k examples (50%) that we use strictly for evaluating the full models (h).

We benchmark the selection functions introduced in Sec. 3.4 in combination with VQA models with varying architectures and performance (test-std VQA v2 accuracy in parentheses): **Pythia** [73] (70.24%), an optimization of the widely used bottom-up top-down VQA model [5]; **ViLBERT** [117] (70.92%), a two-stream transformer, and **VisualBERT** [98] (71.00%), a single-stream transformer, both of which use multimodal pretraining [164]; **CLIP-ViL** [160] (74.17%), which is the MoVie+MCAN [129] model with a visual encoder from CLIP [142].

In Tab. 3.1, Tab. 3.2, and Fig. 3.2, we report mean results over 10 random seeds for Pythia and CLIP-ViL (standard deviations in Appendix B.10), while we report single runs

Model f	Selection function g	VQA Acc. ↑	$\mathcal{C}@\mathcal{R}$ ↑ $\mathcal{R} = 1\%$	$\mathcal{R} = 5\%$	$\mathcal{R} = 10\%$	$\mathcal{R} = 20\%$	AUC ↓
Pythia [73]	MaxProb	64.63	5.84	24.03	39.71	68.63	14.53
	Calibration	64.90	6.22	24.37	40.68	71.29	14.15
	Selector	64.63	**8.30**	**25.87**	**41.71**	**71.37**	**13.94**
	Best Possible ($\mathcal{C}$)	64.63	60.27	66.04	71.54	80.78	7.41
ViLBERT [117]	MaxProb	67.51	7.49	28.56	46.67	77.40	12.37
	Calibration	67.45	8.81	29.42	47.24	77.53	12.22
	Selector	67.51	**11.26**	**31.07**	**48.24**	**77.59**	**11.97**
	Best Possible ($\mathcal{C}$)	67.51	63.00	69.07	74.83	84.39	6.22
VisualBERT [98]	MaxProb	68.44	6.85	30.34	49.22	79.33	11.78
	Calibration	68.27	9.72	31.67	49.68	79.28	11.63
	Selector	68.44	**10.67**	**33.07**	**50.50**	**79.60**	**11.41**
	Best Possible ($\mathcal{C}$)	68.44	63.96	70.07	75.91	85.55	5.86
CLIP-ViL [160]	MaxProb	70.01	6.83	34.08	54.00	82.30	10.81
	Calibration	69.97	12.43	36.02	54.03	82.54	10.55
	Selector	70.01	**15.66**	**37.92**	**55.81**	**82.74**	**10.18**
	Best Possible ($\mathcal{C}$)	70.01	65.71	71.86	77.79	87.51	5.27

Table 3.1: Risk-coverage metrics for different selection functions. For coverage at risk ($\mathcal{C}@\mathcal{R}$) and VQA Acc., higher is better. For AUC, lower is better. All in %.

for ViLBERT and VisualBERT using existing pretrained and fine-tuned models. All other results are single runs from the same randomly chosen seed. Details of data and model setups are in Appendix B.8 and Appendix B.9.

Benchmarking Risk and Coverage

As discussed in Sec. 3.3, we measure the maximum coverage for a given risk ($\mathcal{C}@\mathcal{R}$) as well as AUC for the risk-coverage curves and overall accuracy for each model. We include the best possible performance on these metrics for each model, which would be a selective model that abstains only when the prediction is incorrect. Results are reported on the test test.
Selector outperforms other methods. From Tab. 3.1, we see that adding the Selector consistently outperforms MaxProb in coverage for all risk tolerances as well as AUC. The strongest improvements occur at lower risk tolerances (e.g., 1% and 5%), becoming smaller as the tolerance increases (e.g., 10% and 20%). Notably, CLIP-ViL with Selector can improve $\mathcal{C}@1\%$ to 2.3× that of CLIP-ViL with MaxProb. Fig. 3.2 illustrates how, for low risk levels, the addition of the selector maintains noticeably better risk as coverage increases compared to MaxProb. It generally appears that the more accurate a model is overall, the more it may potentially improve in coverage at low risk tolerances when using Selector. For instance,

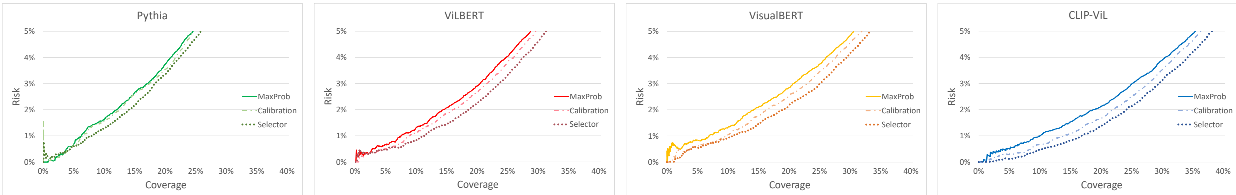


Figure 3.2: Risk-coverage plots for each model up to 5% risk.

when adding the Selector, we observe the largest improvements in $\mathcal{C}$@1% and $\mathcal{C}$@5% with CLIP-ViL (8.83% and 3.84%, respectively), which also has the highest accuracy. Meanwhile, Pythia has the lowest accuracy and exhibits the smallest improvements with the Selector at these tolerances (2.46% and 1.84%, respectively). Fig. 3.2 depicts this between 0-5% risk, where the gap between MaxProb and Selector appears to widen as we move to more accurate models (left to right). Lastly, we observe that Calibration can improve coverage beyond MaxProb as well, but less than the Selector, especially at low risk tolerances (e.g., 1%, 5%), and not as consistently. Because Calibration modifies the output logits, it also slightly changes model accuracy.

Better accuracy $\not\Rightarrow$ better coverage at low risk. While accuracy appears to positively correlate with a better risk-coverage trade-off, the results in Tab. 3.1 also imply that higher accuracy does not guarantee better coverage at low risk. For example, CLIP-ViL has 2.50% higher accuracy than ViLBERT, but, with default MaxProb, ViLBERT has 0.66% higher $\mathcal{C}$@1% than CLIP-ViL. Appendix B.2 also shows that augmenting the VQA model training data with the selection function training data and using MaxProb still has worse coverage at low risk than when using this data for Selector training, despite having higher accuracy. These results imply that improving upon the risk-coverage trade-off requires not only building more accurate models but also learning better abstention policies.

Still room for improvement. Though the evidence presented in Tab. 3.1 and Fig. 3.2 show that coverage at different risk tolerances can be improved, these approaches still fall short of the best possible. For example, in Tab. 3.1, the difference in $\mathcal{C}$@1% between each model with Selector and their respective best possibles is still >50%. Although achieving the best possible may not be realistic, more work is needed to have reliable models with high accuracy and wide coverage that shrink this gap further.

Thresholds generalize to test-time. Thus far, we have evaluated the maximum coverage at an exact risk level. In practice, however, a threshold γ must be chosen, e.g., on a validation set, and used at test-time. We evaluate how close the actual test-time risk is to the target risk when using the validation threshold with VisualBERT, with results in Appendix B.6. We find relatively small differences in risk, showing that the thresholds generalize reasonably well. This aligns with prior findings on other tasks [49]. However, since the actual risks are now slightly different between models, we can no longer compare the corresponding coverages directly. This motivates Effective Reliability, which compares models based on a predefined cost for wrong answers as opposed to an exact risk level.

Model f	Selection function g	c=1			c=10			c=100		
		$\Phi_1 \uparrow$	$\mathcal{R} \downarrow$	$\mathcal{C} \uparrow$	$\Phi_{10} \uparrow$	$\mathcal{R} \downarrow$	$\mathcal{C} \uparrow$	$\Phi_{100} \uparrow$	$\mathcal{R} \downarrow$	$\mathcal{C} \uparrow$
Pythia [73]	—	36.97	35.37	100.00	-211.96	35.37	100.00	-2701.25	35.37	100.00
	MaxProb	46.49	22.48	75.58	15.05	5.68	26.41	1.90	0.94	5.13
	Calibration	47.29	21.66	74.92	15.18	5.97	27.73	2.35	0.92	5.59
	Selector	**47.47**	21.02	73.52	**17.03**	6.34	30.16	**3.84**	1.01	8.23
	Best Possible (Φ_c)	64.63	10.66	72.34	64.63	10.66	72.34	64.63	10.66	72.34
ViLBERT [117]	—	42.91	32.49	100.00	-178.51	32.49	100.00	-2392.75	32.49	100.00
	MaxProb	51.50	21.15	79.92	17.94	6.45	34.50	1.67	1.36	10.18
	Calibration	51.50	19.34	76.08	18.59	4.99	29.39	2.56	1.26	10.97
	Selector	**51.78**	19.88	77.33	**20.90**	5.91	34.56	**5.38**	0.97	11.03
	Best Possible (Φ_c)	67.51	10.45	75.40	67.51	10.45	75.40	67.51	10.45	75.40
VisualBERT [98]	—	44.77	31.56	100.00	-168.30	31.56	100.00	-2299.01	31.56	100.00
	MaxProb	52.82	20.19	79.75	19.24	5.76	33.64	2.50	1.02	6.90
	Calibration	52.82	20.08	79.46	19.87	5.88	35.07	3.92	0.91	8.79
	Selector	**53.20**	19.69	78.95	**21.93**	5.45	34.60	**4.82**	1.07	11.34
	Best Possible (Φ_c)	68.44	10.33	76.33	68.44	10.33	76.33	68.44	10.33	76.33
CLIP-ViL [160]	—	47.68	29.99	100.00	-153.27	29.99	100.00	-2162.82	29.99	100.00
	MaxProb	54.77	19.84	81.98	21.93	5.93	38.47	2.82	0.98	7.27
	Calibration	55.00	18.91	80.24	23.16	5.20	36.73	5.29	0.78	9.96
	Selector	**55.47**	18.18	79.09	**25.93**	5.41	39.55	**8.00**	0.60	11.37
	Best Possible (Φ_c)	70.01	9.86	77.67	70.01	9.86	77.67	70.01	9.86	77.67

Table 3.2: Effective Reliability Φ_c for VQA models with and without abstention options. The best possible Φ_c is computed by only selecting correct predictions, and is equal to the model's VQA accuracy. All in %.

Effective Reliability

We evaluate Effective Reliability (Φ_c) defined in Sec. 3.3, which assigns a cost to incorrect predictions, a reward to correct predictions, and zero to questions on which a model abstained from answering. This provides a single measure to jointly consider reliability (i.e., low risk) and effectiveness (i.e., high coverage). In Tab. 3.2, we choose cost values c of 1, 10, and 100, to observe how models compare when the consequences for providing an incorrect prediction become high. Additionally, we can now directly compare to the original VQA formulation, where models do not have an option to abstain, denoted by a null selection function g. We also include Φ_c for the best possible g, where a model abstains exactly on those inputs which would result in incorrect predictions. As discussed in Sec. 3.3, this is equivalent to the model accuracy. Results are reported on the test set, with an abstention threshold selected to optimize Φ_c on the validation set. We include the corresponding risk and coverage for the selected threshold.

Selector still outperforms other methods. The Selector produces the highest Effective Reliability scores across all models and cost levels. As the penalty for wrong answers increases, the gap between the performance of Selector and the next best model generally increases as well. For example, the improvement of Selector over MaxProb for CLIP-ViL is 0.70% for Φ_1, yet it is 5.18% for Φ_{100}. Further, the gap between Selector and MaxProb for Φ_{100} generally increases as the VQA model itself has higher accuracy (or best possible performance). We observe a similar effect in Fig. 3.2, where more accurate models have larger gaps in risk between Selector and MaxProb at a given coverage.

Cost implicitly controls risk and coverage. When the penalty for a wrong answer is high, one might expect a selective model to operate in the low-risk regime. This is indeed reflected in Tab. 3.2, where the range of risk levels for selective models at Φ_{100} ($\mathcal{R} \approx 0.6$–1.3%) is much lower than the range of risk at Φ_1 ($\mathcal{R} \approx 18$–22%). This directly translates to a similar trend in coverage, where selective models answer about 5–11% of questions at Φ_{100}, and about 74–82% of questions at Φ_1. This shows that Effective Reliability behaves intuitively around the influence of a user-selected cost on model risk and coverage.

Human evaluation shows noise has little effect even with high cost values. For high costs (e.g., $c = 100$), models are strongly penalized for producing incorrect predictions. Given these strict penalties on errors, it becomes pertinent to ask to what degree noise in the annotations might be contributing to these penalties, though the potential impact of noise is certainly not unique to our evaluations and is a challenging problem in VQA [10, 77, 158]. To see if our results for Φ_{100} are significantly affected by annotation noise, in Appendix B.3, we manually examine each sample where the model predictions were marked incorrect (and thus heavily penalized when computing Φ_{100}). We annotate cases where models may have been unfairly penalized and recompute Φ_{100} when removing this penalty. We find that vast majority of incorrect predictions that contribute to these penalties are properly marked as incorrect. We also see that label noise does slightly change the Effective Reliability scores at high cost, but the rankings between models and selection functions are preserved.

All models without an abstention option perform poorly. When the cost of a wrong answer is equal to the reward of getting an answer entirely correct ($c = 1$), all models without a selection function g underperform their selective model counterparts. As c increases, this gap widens dramatically, with non-abstaining models reaching Φ_c values firmly in the negative range. Meanwhile, all selective models reach a positive Φ_c, even at high cost, illustrating the necessity of the abstention option for building models which are reliable and effective.

Selection Function Ablations

Tab. 3.3 provides ablations for the selection function design. In the following, we distill the main observations. Additional discussion is in Appendix B.1.

Selector requires multimodal input. Tab. 3.3 shows the importance of using multimodal information for coverage at low risk levels. When using each representation in isolation, we see that multimodal representations (r, v, and $f'(x)$) yield much stronger $\mathcal{C}$@1%, $\mathcal{C}$@5%, Φ_{10}, and Φ_{100} than unimodal representations (image $\tilde{v}$ or question q). For highly reliable models ($\mathcal{C}$@1%, Φ_{100}), unimodal selection functions fail (coverage $\leq$0.02%, $\Phi_{100} < 1\%$), suggesting that building reliable and effective VQA models is a truly multimodal problem. Combining all representations generally performs well, so we use this setup in all experiments.

Regressing to VQA accuracy is important. We find that formulating the objective as a regression of the answer accuracy, rather than classifying whether the answer is correct, offers significant improvements (Tab. 3.3), especially at low risk. This is likely because predicting

Features	Unimodal	Loss	$\mathcal{C}@\mathcal{R}$ ↑ $\mathcal{R}=1\%$	$\mathcal{R}=5\%$	$\mathcal{R}=10\%$	$\mathcal{R}=20\%$	AUC ↓	Φ_c ↑ c=1	c=10	c=100
$\tilde{v}$	✓	Regression	0.00	0.00	0.00	10.18	24.91	47.10	-0.01	-0.85
q	✓	Regression	0.02	10.78	33.97	76.33	14.06	51.81	10.25	0.94
$f'(x)$		Regression	5.08	34.61	54.32	81.98	10.77	55.05	22.99	5.87
v		Regression	11.41	35.34	51.45	80.57	11.01	53.75	23.79	6.55
r		Regression	**13.26**	32.88	51.26	80.23	11.11	53.37	22.17	**7.76**
$f'(x)+\tilde{v}$		Regression	3.67	34.97	54.49	82.06	10.76	54.94	23.47	4.59
$f'(x)+q$		Regression	8.97	35.89	55.13	82.13	10.55	55.01	24.18	5.32
$f'(x)+r$		Regression	10.17	35.89	<u>55.19</u>	<u>82.27</u>	10.49	<u>55.15</u>	24.19	5.51
$f'(x)+v$		Regression	12.34	**37.26**	55.12	**82.40**	<u>10.45</u>	**55.16**	**24.95**	7.02
$f'(x)+q+v+r$		Classification	6.51	34.87	55.16	81.58	10.69	54.69	23.14	4.36
$f'(x)+q+v+r$		Regression	<u>12.92</u>	<u>36.29</u>	**55.64**	<u>82.27</u>	**10.43**	55.13	<u>24.66</u>	<u>7.31</u>

Table 3.3: Ablations of Selector with CLIP-ViL [160] on our selection function validation set. The overall best performance is in bold and second best is underlined. $f'(x)$, q, $\tilde{v}$, and r are the answer, question, image, and multimodal representations, respectively. Note, v is a question conditioned image representation that is not unimodal (see Appendix B.1 for details). All in %.

the fine-grained accuracy allows the model to account for partially correct answers and learn to rank answers that are more correct higher, as opposed to classification where the distinction between partially correct answers is lost.

Selector Architecture. Appendix B.1 presents results using different Selector architectures, where a less complex architecture can degrade performance, but a more complex one does not necessarily improve it. Together with Tab. 3.3, we find that, rather than the network layout, the *input* to the Selector and optimization target are more critical to the performance when using the Selector.

Qualitative Analysis

Fig. 3.3 visualizes MaxProb and Selector decisions with CLIP-ViL for several examples on the test set (more in Appendix B.5). The abstention threshold is chosen to maximize Φ_{100} on validation. Fig. 3.3 (left) shows an example of a question that requires commonsense reasoning to answer that the VQA model may not be certain of (and gets wrong), so Selector abstains. Similarly, in Fig. 3.3 (middle), we see a false premise question [143] where Selector abstains again as the question does not make sense for the image, while MaxProb yields an incorrect answer. Fig. 3.3 (right) presents an example with synonymous answers where the model is correct yet MaxProb chooses to abstain and Selector chooses to answer. In a classification-based VQA model, synonyms can split the maximum softmax score used by MaxProb, whereas the Selector can potentially learn these answer similarities and adjust the confidence. These examples contribute to the higher coverage at low risk observed quantitatively in our experiments. We also find that MaxProb chooses to answer many simple

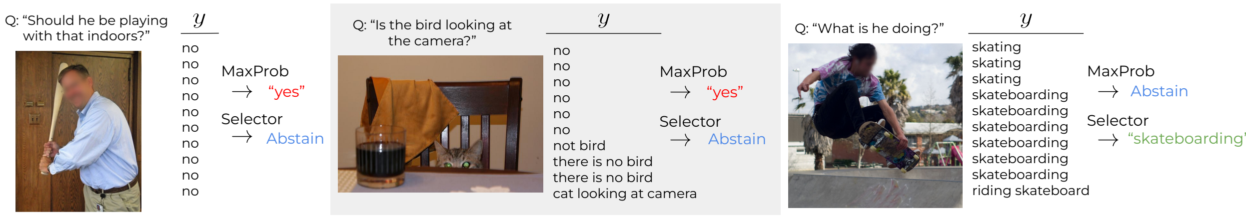


Figure 3.3: Qualitative test set examples with CLIP-ViL selective model predictions.

questions, while Selector additionally chooses to answer more difficult, multimodal ones as well (see Appendix B.4).

3.6 Conclusion

The standard VQA formulation does not include an option for models to abstain from answering if they are uncertain. However, for many applications, it is important that the model only provides an answer if there is a low risk of error. In this work, we promote a problem formulation for VQA which includes an option to abstain and discuss how to evaluate this, including a metric that rewards correct predictions but expects models to abstain if they are incorrect. We benchmark several VQA models in combination with approaches for abstention. If we want a reliable model with 1% risk of error, we find that a state-of-the-art VQA model [160] only answers less than 7.5% of the questions when using its softmax probabilities as estimates of model confidence. Using calibration can improve this, but we find that the best results are consistently achieved by training a multimodal selection function to estimate correctness directly. This increases the coverage from 6.8% to 15.6%. While this is a marked improvement, one has to consider that this model achieves 70% standard VQA accuracy on the same set of data. With our *Effective Reliability* metric, the performance drops from 70% (for perfect abstention) to 8% (our best abstention model) with high penalties for wrong answers. We believe this new framework and metric for VQA will encourage the community to build VQA models which are both reliable and effective, as well as offer an opportunity for many exciting directions to improve the self-awareness of models.

Acknowledgements: We thank Anastasios Angelopoulos and Kurt Shuster for helpful discussions. Authors, as part of their affiliation with UC Berkeley, were supported in part by the NSF CISE Expeditions Award CCF-1730628; DoD, including DARPA's LwLL, PTG, and/or SemaFor programs; the Berkeley Artificial Intelligence Research (BAIR) industrial alliance program as well as gifts from Amazon Web Services, Ant Group, Ericsson, Facebook, Futurewei, Google, Intel, Microsoft, Scotiabank, and VMware.

Chapter 4

Simple Token-Level Confidence Improves Caption Correctness

4.1 Introduction

For vision-and-language models, grounding and the ability to assess the correctness of a caption with respect to an image is critical for vision-language understanding. When models have difficulties with these, the outputs can be error prone [149] or rely on biases [3, 63]. State-of-the-art models, like CLIP [142] or OFA [184], demonstrate impressive capabilities in a variety of settings, in part, thanks to these properties. While these models have had much success, recent efforts for probing state-of-the-art models have revealed some weaknesses in these areas. For instance, the recent Winoground task [173] illustrates that these models, including large-scale pre-trained ones, can struggle to correctly associate image-caption pairs when the captions have differences in word order. Similarly, SVO-Probes [62] has shown that models can fail in situations that require understanding verbs compared to other parts of speech. The observations from these probing tasks suggest that existing models have difficulties discerning fine-grained details that can appear in multimodal data. This may hinder their accuracy and reliability when used in real settings, which presents significant issues in scenarios that require highly correct outputs, such as assisting people with visual impairments [59, 192].

We conjecture that these weaknesses may be related to the granularity with which models perform image-text matching (ITM). As shown in Fig. 4.1, many existing models often operate at a sequence-level, pooling the representations of the image and caption to assess whether the text correctly describes the image. This pretext task relies on sequence-level supervision and data with sufficient scale to learn finer-grained concepts, such as the difference between

This chapter is based on joint work with Spencer Whitehead, Joseph E. Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach. It is presented much as it appeared in the WACV 2024 proceedings.

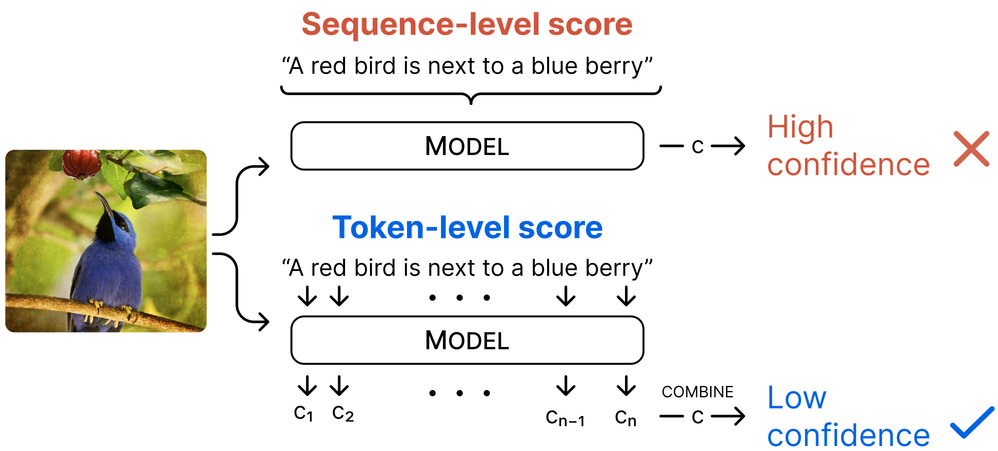


Figure 4.1: Judging caption correctness is still a challenge for large-scale models that operate at a sequence-level. We show that both algebraic and learned confidences at a token-level from a finetuned image captioning model improve fine-grained estimates of caption correctness.

Dataset & Task	Winoground [173]	SVO-Probes [62]	Hallucination in Captioning [149]
Metric	Acc Image (↑)	Accuracy (↑)	CHi (↓)
Prior SoTA	19.75 [145]	–	3.2 [102]
Baseline (ours)	10.25	81.23	2.0
Ours	**27.00**	**89.47**	**1.4**
Rel. Improvement	37%	10%	30%

Table 4.1: Summary of results. Despite its simplicity, the relative improvement over the next best approach highlights the significance of TLC for caption correctness.

"a cat jumping over a box" and "a box with a cat inside". Typical generative image captioning methods, on the other hand, generate words token-by-token and produce confidences for each one. They are supervised at a token-level rather than sequence-level, which may emphasize the consistency of each token in a sequence more explicitly.

Leveraging this observation, we explore Token-Level Confidence, or TLC, for assessing image-caption correctness. We input an image and proposed caption into a finetuned captioning model, which produces a distribution over the vocabulary at each time step. The base TLC method, TLC-A, uses algebraic confidence measures (*e.g.*, softmax score) to compute confidence for a given token. To produce a single score for image-caption correctness, we either aggregate token confidences over the sequence (*e.g.*, by taking the average value), or over particular words, such as verbs or objects. Next, we further investigate whether learned confidences can outperform algebraic ones. We propose a Learned confidence estimator, TLC-L, for use in the caption generation setting where training data is available. We use existing annotations to model the likelihood that a predicted token matches reference tokens, and an additional validation set to calibrate our estimated confidence to actual correct and incorrect concepts. Using TLC-L to re-rank candidate captions, we reduce hallucination rates in the final output captions.

Both TLC-A and TLC-L are simple to implement and can be applied on top of any autoregressive image captioning model with an encoder and decoder, an architecture found to scale well with data and multimodal tasks [21, 184, 191, 183]. In this work, we demonstrate the effectiveness of token-level confidence across multiple model sizes of OFA [184], a recent Transformer-based model [177] with strong performance on many vision-language tasks. As summarized in Tab. 4.1, on the challenging Winoground [173] benchmark evaluating compositional reasoning, we show that TLC-A more than doubles accuracy over pretrained ITM scores, *e.g.*, from 10.25% to 27% on image score. TLC-A additionally shows a relative improvement of image and group scores of 37% and 9%, respectively, over the prior state-of-the-art on Winoground [145], which used a regularization tailored for multimodal alignment. TLC-A also outperforms ITM on a fine-grained verb understanding task [62] by a relative 10%. When using TLC-L to re-rank candidate captions on MS COCO [22], we achieve a 30% relative reduction in object hallucination rate over the original captions and set a new state-of-the-art on a hallucination benchmark [149]. These results demonstrate that token-level confidence, whether algebraic (TLC-A) or learned (TLC-L) are a powerful yet simple resource for improving multimodal reliability.

4.2 Related Work

Caption correctness. One of the desired properties of a good caption is correctness, *i.e.*, being faithful to an image. [62, 133, 173] propose benchmarks to probe for sensitivity to hard negatives of different types, such as compositional reasoning or action understanding. We use probing benchmarks in our work to demonstrate the effectiveness of TLC-A. Within caption generation, [149] notes that in practice, image captioning models suffer from object hallucination [149], driven by visual misclassification and over-reliance on language priors. Several recent works addressed the issue of object hallucination [14, 102], in some cases relying on causal inference-based approaches [108, 200, 199]. Other recent works pose a slightly distinct problem of correcting errors in a caption provided for a given image (i.e., not as part of the caption generation process) [153, 154, 190]. Some works propose caption decoding methods such as constrained beam search [6], an uncertainty-aware beam search using prediction entropy [195], or a non-autoregressive caption decoding method [43] to target criteria such as correctness. However, the original formulation of beam search remains the dominant decoding method used in modern multimodal architectures [21, 96, 184, 187]. We apply our approach on top of captions generated with beam search and demonstrate that simply re-ranking beams based on token confidences can reduce hallucinations.

Correctness estimation in language models. Similar issues around correctness and hallucination are also relevant for many language-only tasks that require autoregressive prediction.

Hallucination in particular has been studied for tasks like abstractive summarization [122], e.g., one work performs token-level hallucination detection [215]. A number of works study model uncertainty and aim to improve model calibration for machine translation [51, 54, 185], dialog [124], question answering [210] and spoken language understanding [161], to name a few tasks. While our focus on image captioning is similarly a conditional generation task, estimating confidence in the multimodal setting can be challenging as errors are driven by factors from both modalities [192].

Image captioning. Image captioning has seen significant progress since the arrival of deep learning as a dominant methodology [5, 39, 68, 80, 146, 179]. In recent years Transformer-based architectures have gained particular prominence [101, 160, 209]. Many papers take the approach of pretraining large vision-and-language models and then adapting them to downstream tasks, including captioning [100, 191]. Recent efforts focus on further scaling these pretraining-based methods [4, 67, 183, 205], while many also aim to unify multiple vision-and-language tasks during pretraining [21, 26, 184, 187]. Despite steady improvements in image caption quality over the past years, even the best models still make mistakes. Here, we study the reliability of vision-language models, with the goal of assessing caption correctness.

Reliability in multimodal models. With the adoption of Large Language/Vision/Vision-and-Language Models (LLMs, LVMs, LVLMs), it is increasingly important to study their limitations and outline expectations regarding their *reliability.* One of the first efforts in doing that for LLMs and LVMs (unimodally) is [174], whose broad definition of reliability includes aspects from modeling uncertainty to robust generalization and adaptation. A recent work in multimodal learning outlines reliability of visual question answering [192], defining it as a model's ability to ensure a low risk of error by means of abstaining from answering. In our work, we approach reliability by improving assessments of caption correctness, and incorporating these estimates to reduce rates of error in generated captions.

4.3 TLC: Token-Level Confidence for Caption Correctness

Overview. Given an image and a caption, TLC produces a confidence score for each token and aggregates these scores to produce an estimate of caption correctness, *i.e.*, semantic consistency with the image. First, we describe two forms of confidences: algebraic (TLC-A) and learned (TLC-L). Next, we describe how to combine token confidences to measure caption correctness and use token confidences to re-rank captions during generation. In our experiments, we will then verify TLC-A primarily on out-of-domain probing benchmarks. We then evaluate TLC-L in a setting where in-domain training data is available.

Preliminaries. Let f_{pre} be a vision-language model pretrained on a large multimodal dataset, and f_{cap} be a model initialized with f_{pre} and subsequently finetuned for autoregressive image captioning. Given an image x, a caption consists of a sequence of n tokens $t_{1:n}$

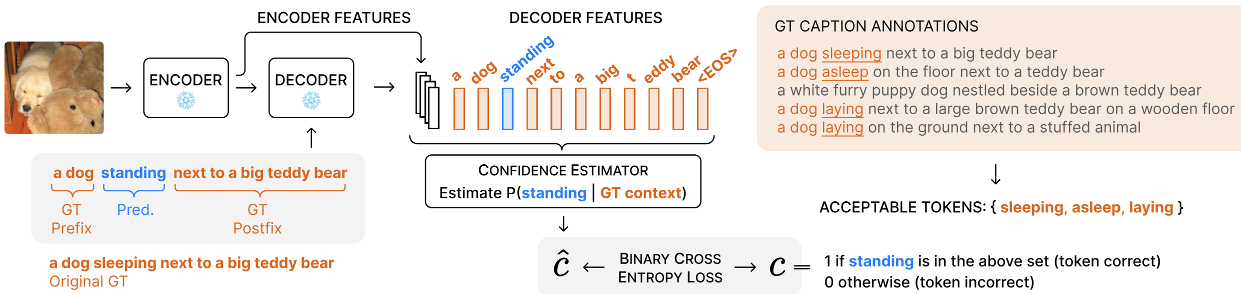


Figure 4.2: TLC-L: A framework to learn token-level confidence for a pretrained autoregressive encoder-decoder captioning model. We first use the captioning model to predict the next token (*e.g.*, "standing") after a partial reference caption (*e.g.*, "a dog"), shown in the bottom left. We input this sequence along with the image and the rest of the reference caption to the model, and obtain corresponding encoder and decoder features. These features become the inputs to our confidence estimator, a Transformer encoder. For supervising correctness, we create a binary classification task to learn whether or not the model's predicted token matched any reference token at the same time step with the same prefix.

describing the image. At each decoding time step $k \in \{1...n\}$, f_{cap} produces a distribution of token likelihoods $\vec{z}_k \in \mathbb{R}^{|V|}$ for a vocabulary V, conditioned on previous outputs $\vec{z}_{1:k-1}$. Autoregressive captioning models are typically trained with a token-level cross-entropy loss on $\vec{z}_k$, often followed by self-critical sequence training [146]. Decoding methods such as sampling or beam search can then be used to select tokens at inference time, typically aiming to maximize the image-conditional sequence likelihood.

TLC-A: Algebraic Confidences

A simple method for measuring token-level confidence is to use an algebraic function of the distribution $\vec{z}_k$ directly, such as taking the logit or softmax value at the selected token index. We refer to token confidences derived from algebraic functions of $\vec{z}_k$ as TLC-Algebraic, or TLC-A. Prior works find simple measures such as softmax to be unreliable in both vision and vision-language "one-of-K" classification tasks [58, 192]. In contrast, we find that softmax scores from autoregressively-generated tokens perform surprisingly well, even on data that is out-of-distribution from the image captioning training set used by f_{cap}. This is aligned with findings in the language-only setting [36, 168, 176], suggesting that token-level language modeling may be key for reliable confidence measures.

TLC-L: Learned Domain-Specific Confidences

Although we observe that TLC-A performs well on evaluation benchmarks out-of-distribution from the image captioning training data (Sec. 4.4), we would like to see whether *learning* a confidence estimator on in-distribution training data could improve estimates of correctness, similar to [192]. However, we do not have direct supervision to measure the correctness of a specific token in an arbitrary predicted caption with an image, aside from human evaluation. Instead, we leverage existing reference captions to learn a binary classification task, measuring whether a predicted token matches one or more reference tokens at the same time step. Fig. 4.2 presents an overview of this method, which we refer to as TLC-Learned, or TLC-L.

Forming the training set. We begin with a trained and frozen f_{cap} and use a heldout dataset $\mathcal{X}$ for training a confidence estimator g. Compared to the training set for f_{cap}, $\mathcal{X}$ provides a better estimate of the captioning model's performance on test data. In this work, we simply use the f_{cap} validation set. For each image in $\mathcal{X}$, paired with one or more references, we select one of the reference captions $t_{1:n}$ and time step k within the caption. We first input the *prefix*, or $t_{1:k-1}$, into the f_{cap} decoder to predict the next token, $\hat{t}_k$. We assign a binary label c to $\hat{t}_k$ – it is **correct** ($c = 1$) if it matches the reference token t_k or any token at k from other reference captions with the same prefix. Otherwise, $\hat{t}_k$ is labeled as **incorrect** ($c = 0$). For example, in Fig. 4.2, the original reference token t_k is "sleeping", yet "asleep" and "laying" are also considered correct, given that they share the same prefix "a dog". The predicted token $\hat{t}_k$ "standing" is therefore labeled as incorrect. This provides proxy for true consistency with the image, which may be noisy; for example, "resting" would be considered incorrect in Fig. 4.2. Nevertheless, these labels enable TLC-L to learn effective in-domain confidences (Sec. 4.4). At each epoch, we re-sample a reference caption and a time step k for each image in order to leverage all available ground-truth tokens.

Training a confidence estimator. The output of g is a scalar $\hat{c}$, trained with binary cross-entropy loss with c as supervision. As input, g receives image features from the model, such as those output by an encoder. It also receives token-level features from the decoder (*e.g.*, just before decoder features are projected into the vocabulary space). We find that including the reference *postfix*, or $t_{k+1:n}$, in addition to the prefix $t_{1:k}$ and predicted token $\hat{t}_k$ improves the confidence estimation. We pass the encoder features and position-encoded decoder sequence into a Transformer encoder [177], and pass the output embedding of token $\hat{t}_k$ into a small feed-forward network to produce $\hat{c}$. We provide details on our specific choice of architecture in Sec. 4.4. At inference time, we run our confidence estimator once per time step within a predicted caption $\hat{t}_{1:n}$.

A bidirectional confidence. Although we supervise confidence for a single token $\hat{t}_k$ at a time, the full caption context is given as input. Due to self-attention in the Transformer encoder within g, the final prediction $\hat{c}$ represents a bidirectional confidence estimate, in contrast to the original autoregressive token predictions. This enables a useful combination: generating tokens autoregressively scales well with data and model size [184, 21], whereas estimating token confidence bidirectionally uses future context to inform correctness.

From Confidence to Caption Correctness

Combining Confidences

In practice, we would like to measure correctness over an entire caption or particular span, such as a word or phrase. To obtain such a score from token-level confidences, we can simply aggregate the confidences over a specific span of tokens $t_{i:j}$ or the full sequence $t_{1:n}$ by taking, *e.g.*, the minimum or average confidence value. We exclude the end-of-sentence (EOS) token, as its confidence is often poorly calibrated relative to previous tokens [88]. In our experiments, we compare correctness between image-caption pairs by aggregating over the full sequence or specific words.

Confidence During Caption Generation

We can use token-level confidences to not only estimate correctness between an image and an *existing* caption but also between a *proposed* caption candidate during generation. By re-ranking candidates relative to estimated correctness, we can reduce errors in the final selected captions.

When generating a caption, it is common to first predict a set of B candidate captions using an autoregressive decoding method such as beam search. Initially, the beams are ranked according to their cumulative token log likelihoods from the captioning model:

$$\mathbb{P}(t_{1:n}) = \sum_{k=1}^{n} \log p(t_k \mid t_{1:k-1}, x) \tag{4.1}$$

However, token likelihood can fail to rank captions that are fully correct above those that contain an error. For example, a fluent and detailed sentence with a single-word hallucination may rank above a simpler, yet correct, caption. This is observed in [149], where captions with higher CIDEr [178] could also have higher hallucination rates. It is also similar to prior work in machine translation [54], which noted that errors can be "bad luck" from generation rather than inherent model failure.

To alleviate this, we first define a set of words or concepts $\mathcal{S}$ that we estimate correctness for. For example, in our experiments, we consider only the tokens that correspond to MS COCO [22] object categories, as we have annotations for their correctness during validation and evaluation. Beginning from the highest-likelihood beam, we estimate confidence $\hat{c}$ for each set of words in $\mathcal{S}$ that appear in the beam (*e.g.*, each MS COCO object that is mentioned). If any $\hat{c}$ are less than a threshold γ, we reject the beam, and continue to the next one until we reach a beam where all relevant tokens are predicted to be correct ($\hat{c} \geq \gamma$), or where there are no tokens from $\mathcal{S}$. If none of the beams satisfy these criteria, we output the original (highest-likelihood) caption. In that setting, we could alternatively choose to abstain from providing a caption in order to avoid misleading a user, similar to [192]. However, we instead choose the original caption in our experiments to simplify the comparison between methods.

We choose the threshold γ on a validation set to control the rate of false positives. This is captured by the precision: *"out of all samples predicted as correct, what fraction are* actually *correct?"* We define a target precision α, such as 99%, and select γ such that the binary decisions $\hat{c} \geq \gamma$ maximize the recall of correct samples in $\mathcal{S}$ on the validation set.

4.4 Experiments

After discussing the experimental setup, we demonstrate the effectiveness of TLC-A for identifying correct image-caption pairs that test understanding of compositionality and verbs. We then evaluate both TLC-A and TLC-L on reducing object hallucinations in generated captions.

Experimental Setup

As a captioning model, we choose to experiment with OFA [184], a recent open-source sequence-to-sequence multimodal transformer that achieves state-of-the-art captioning performance. OFA has a simple encoder-decoder architecture designed to unify multimodal tasks conditioned on an image and specific input instruction (e.g., "What does the image describe?" prompts the model to output a sequence of tokens for captioning). We use the official implementation and checkpoints (f_{pre}) for $\text{OFA}_{\text{Large}}$, OFA_{Base}, and OFA_{Tiny}, pretrained on a dataset with 20M publicly available image-text pairs. As image-text matching was included as a task in OFA pretraining, we use **ITM** in our results to denote the image-text matching score from f_{pre}. For f_{cap}, we finetune each scale of OFA model on MS COCO Captions [22], which has about 80k training images. We split the validation set of 40k images into three parts for training, validation, and testing of g, following [192]. Additional dataset details are in Appendix C.4.

For TLC-A, we use the softmax score at the selected token index. We experiment with several other choices of algebraic function and report results in Appendix C.2. For TLC-L, as input to the learned confidence estimator g, we use multimodal image and instruction features output from the OFA encoder, as well as token embeddings from the decoder just before they are projected onto the logit space by a linear layer. g itself is a 4-layer Transformer encoder [177], followed by a 2-layer MLP. We add a learned positional encoding to the token features, and train g for 200 epochs on 8 V100 GPUs. Additional details are in Appendix C.5.

Correctness Around Compositional Reasoning

First, we assess the ability of TLC-A to select corresponding image-caption pairs. We use Winoground [173], a dataset curated to test the compositionality of vision-language models. Each of the 400 examples contains two image-caption pairs (I_0, C_0) and (I_1, C_1). Captions C_0 and C_1 contain the same words and/or morphemes, yet differ in order; for example, "there is a mug in some grass" and "there is some grass in a mug". There are three evaluations per example: text score (given an image, select the correct caption), image score (given a caption,

Model	Conf.	Text	Image	Group
MTurk Human [173]	-	89.50	88.50	85.50
Random Chance [173]	-	25.00	25.00	16.67
UNITER$_{\text{Large}}$ [173]	ITM	38.00	14.00	10.50
VinVL [173]	ITM	37.75	17.75	14.50
CACR$_{\text{Base}}$ [131]	CACR	39.25	17.75	14.25
IAIS$_{\text{Large}}$ [131]	IAIS	*42.50	19.75	16.00
OFA$_{\text{Large}}$	ITM	**30.75**	10.25	7.25
	TLC-A	29.25	***27.00**	***17.50**
	(Δ)	(-1.5)	($+16.75$)	($+10.25$)
OFA$_{\text{Base}}$	ITM	**26.75**	10.75	6.50
	TLC-A	24.50	**23.50**	**13.75**
	(Δ)	(-2.25)	($+12.75$)	($+7.25$)
OFA$_{\text{Tiny}}$	ITM	**22.75**	7.75	4.50
	TLC-A	16.50	**15.75**	**6.75**
	(Δ)	(-6.25)	($+8.00$)	($+2.25$)

Table 4.2: Accuracy on text, image, and group score for the Winoground evaluation dataset [173]. Citations indicate where scores are reported, and * indicates state-of-the-art.

select the correct image), and group score (all text and image scores for an example must be correct). A pairing is considered correct if the image-caption matching score for the correct pair is greater than that of the incorrect pair (*i.e.*, $c_{POS} > c_{NEG}$). [173] find that the task is surprisingly difficult, with all models they test performing below random chance for image and group score.

As correctness estimates, [173] use image-text matching scores (ITM) from a range of pretrained vision-language models. Other works [131, 145] design training losses specifically targeting relation alignment. Using TLC-A, we produce a correctness estimate c by simply averaging token-level softmax scores for each proposed image-caption pair. We present results in Tab. 4.2.

TLC-A outperforms prior SOTA image and group performance. TLC-A with OFA$_{\text{Large}}$ reaches above random chance for both image and group score, improving over prior state-of-the-art. Despite its simplicity, with no additional training beyond standard image captioning, TLC-A outperforms IAIS (proposed in [145]), a training method optimized for multimodal attention alignment. Compared to ITM across OFA model sizes, TLC-A more than doubles the image and group scores in all but one case (OFA$_{\text{Tiny}}$ group).

Confidence	Model		
	$\text{OFA}_{\text{Large}}$	OFA_{Base}	OFA_{Tiny}
ITM	81.23	78.44	65.25
TLC-A	**89.47**	**89.64**	**81.34**
(Δ)	(+8.24)	(+11.20)	(+16.09)

Table 4.3: Image-caption matching accuracy for verb understanding with a subset of SVO-Probes [62]. TLC-A uses token-level softmax scores aggregated over the verb in each example.

Correctness Around Verb Understanding

Next, we consider caption correctness when aggregating token confidences over a single word, rather than over a full sequence. To evaluate this, we use SVO-Probes, a dataset designed by Hendricks and Nematzadeh [62] to test the verb understanding of vision-language transformers. Each example contains an image and a caption describing a ⟨subject, verb, object⟩ relation in the scene. It also contains a negative image, where only one part of the relation is different, such as ⟨person, swim, water⟩ and ⟨person, walk, water⟩. We use a publicly available subset of about 6,500 examples for verb understanding, and use a parser [65] to annotate the location of the verb in each caption. We aggregate token confidences over the verb tokens for TLC-A. Tab. 4.3 presents image-caption accuracy, where a score is 1 if the confidence is greater for the correct image (again, if $c_{POS} > c_{NEG}$).

TLC-A outperforms image-text matching scores. From Tab. 4.3, we see that TLC-A reaches higher image-caption matching accuracy compared to the ITM scores from pretrained models, across a range of model sizes (*e.g.*, 8.24% and 11.20% improvement for $\text{OFA}_{\text{Large}}$ and OFA_{Base} respectively). Therefore, when localized word or token positions are available, they can be leveraged for a finer-grained matching score than ITM operating on the full sequence.

Reducing Object Hallucinations

We now test our approach described in Sec. 4.3, where we select a caption from a set of candidates to lower the likelihood of error. We also evaluate learned confidences from TLC-L, now that we can use domain-specific training data for g with the image captioning validation set. Prior work [149] provides a framework for measuring object hallucination on MS COCO data. [149] provides a method to enumerate MS COCO objects mentioned in references for a given image and enumerate objects mentioned in an arbitrary, predicted caption. We also add part-of-speech taggers [13, 65] to exclude predicted words that are not nouns; however, when comparing directly to prior work, we use the original implementation. A hallucination is flagged when a prediction mentioned an object not present in the reference set. This is evaluated by sentence-level and object instance-level CHAIRs and CHAIRi metrics [149]:

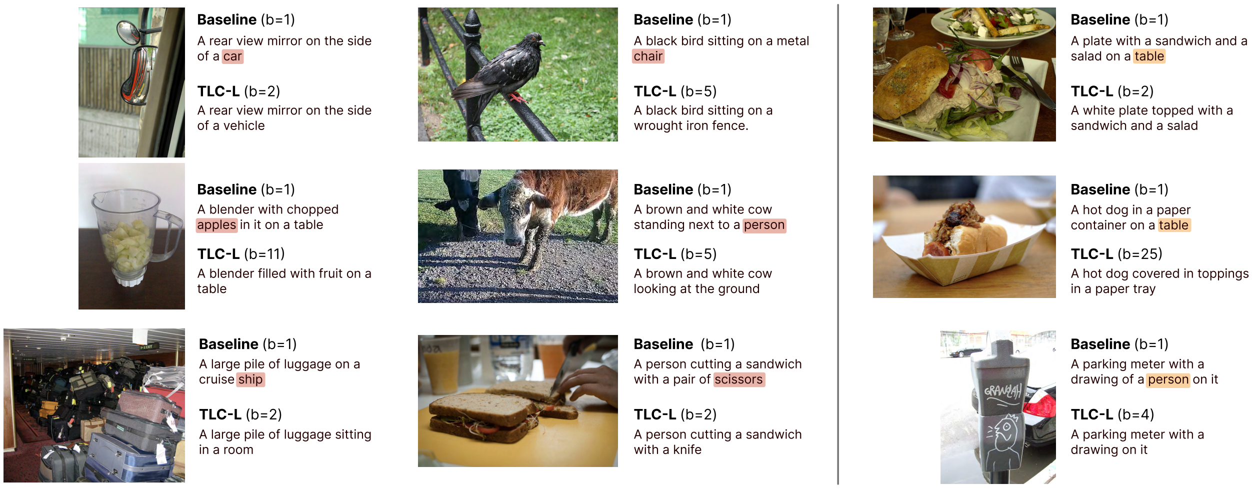


Figure 4.3: Qualitative examples from our test set in which TLC-L avoided hallucinations in the original (Baseline) captions. In the rightmost column, we show cases where the MS COCO object annotations did not exhaustively include all objects present. Captions are generated with $\text{OFA}_{\text{Large}}$ and a beam size of 25, and $(b = i)$ refers to the index i of the beam as ranked by the Baseline.

$$\text{CHAIR}_s = \frac{\#\text{ captions with} \geq 1 \text{ hallucination}}{\#\text{ captions}} \tag{4.2}$$

$$\text{CHAIR}_i = \frac{\#\text{ objects hallucinated}}{\#\text{ objects mentioned}} \tag{4.3}$$

We report standard captioning metrics [7, 178] as well as CHAIRs and CHAIRi (or CHs and CHi). We also report several caption diversity measures [162, 197] to examine whether captions with lower hallucination rates reduce caption diversity: *Vocab Size* measures unique unigrams across predictions, *% Novel* measures the percentage of generated captions which do not appear in the training set annotations, *Div-2* measures the ratio of unique bigrams to the number of generated words, and *Re-4* measures the repetition of four-grams.

For both TLC-A and TLC-L, we choose a threshold γ on the validation set. This threshold is used at test time to make binary decisions on the correctness of a given object in a predicted caption. We extract all objects from the validation set predictions, as well as corresponding token confidences and ground-truth hallucination scores. Then, we choose a confidence level γ that reaches at least 99% precision when separating correct vs. hallucinated objects. This precision is intentionally very high; the OFA captioning models have fairly low rates of hallucination on MS COCO already (as seen in Tab. 4.4), yet we are interested in pushing the caption reliability as far as possible. When aggregating token confidences over object words, we select the minimum value for TLC-A and the average value for TLC-L based on the validation set recall. We use a large beam size of $B = 25$ to observe the behavior of our caption selection method when given many possible candidates.

We show results from the following methods. **Standard** uses the original top caption, that is, the caption from the beam ranked highest by f_{cap}. **Standard-Aug** uses the top caption from a captioning model f'_{cap}, where its training set is augmented by the training set for g. This tests whether the improvements from TLC-L result from using token confidence itself or from additional training data. More details on Standard-Aug are in Appendix C.5. **ITM** uses f_{pre} to re-rank the B candidate captions from Standard based on their image-text matching score, and selects the highest-ranked caption as output. **TLC-A** and **TLC-L** use the respective algebraic or learned confidences over the MS COCO object words to re-rank captions as described in Sec. 4.3.

Learned confidences lead to the least hallucinations. From Tab. 4.4, we can see that both TLC-A and TLC-L lower the CHs and CHi hallucination rates across all model sizes compared to the original (Standard) captions. TLC-L reaches the lowest rates in each case; for example, it lowers CHs and CHi for $\text{OFA}_{\text{Large}}$ by a relative 37.6% and 34.3% respectively. Additionally, TLC-L lowers hallucination rates compared to Standard-Aug as well (*e.g.*, a relative 20.9% lower CHs for $\text{OFA}_{\text{Large}}$). This indicates that reserving a portion of data to train g can have a bigger impact on reducing hallucinations than does using the data for augmentation. Using ITM scores slightly lessens hallucination rates over Standard, yet at the cost of large degradation in CIDEr and SPICE, and underperforms TLC in all metrics. In Tab. 4.6, we further evaluate hallucination rates on the subset of images where the top beam from Standard was *not* selected by TLC-L with $\text{OFA}_{\text{Large}}$– in other words, samples where using TLC-L made a difference. This occurred in almost a quarter of the captions. Standard hallucination rates are much higher on this subset (*e.g.*, 6.78% CHs), whereas TLC-L reduces this by at least half.

Captioning metrics do not capture hallucinations. CIDEr and SPICE decrease across all TLC-based approaches, despite having dramatic reductions in hallucination rates. This effect was also observed by [149], which described how standard metrics can often fail to penalize hallucinations. For instance, the majority of a sentence might overlap with a reference caption, yet still, misclassify an object. [119] nevertheless find that some visually-impaired users of captioning systems prefer correctness above possibly-wrong detail, motivating the drive for low hallucination rates.

TLC improves caption diversity. From Tab. 4.5, our method achieves higher performance on diversity metrics across all model sizes. For instance, TLC-A consistently increases bigram uniqueness score *Div-2*, and decreases the repetition measure *Re-4*. Incorporating confidence into caption selection may help overcome language priors, where co-occurrence statistics from training influence token likelihoods. Diversity can improve as a result, where captions are driven more by consistency with the image rather than language. For example, the top center sample in Fig. 4.3 shows the baseline hallucinating a "metal chair", compared to the correct yet uncommon words "wrought iron fence" described by TLC-L.

Qualitative analysis. We show several qualitative examples in Fig. 4.3. In the left column, we see two examples where TLC-L "backed-off" to a more general concept, whereas the baseline was specific, yet the image did not contain enough information to determine whether the specificity was indeed correct (*e.g.*, "car" vs. "vehicle" and "apples" vs. "fruit"). A prior

Model	Confidence	Hallucination		Quality	
		CHs (↓)	CHi (↓)	CIDEr (↑)	SPICE (↑)
$\text{OFA}_{\text{Large}}$	Standard-Aug	2.20	1.38	153.3	26.7
	Standard	2.79	1.78	**144.4**	**25.8**
	ITM	2.57	1.76	126.5	24.4
	TLC-A	1.81	1.24	140.7	25.5
	TLC-L	**1.74**	**1.17**	141.8	25.4
OFA_{Base}	Standard-Aug	3.00	1.89	148.8	26.1
	Standard	3.78	2.39	**142.9**	**25.6**
	ITM	3.22	2.15	127.1	24.3
	TLC-A	2.47	1.75	137.5	25.2
	TLC-L	**2.05**	**1.48**	137.5	24.9
OFA_{Tiny}	Standard-Aug	10.58	6.83	119.8	22.1
	Standard	11.01	7.23	**117.4**	**21.7**
	ITM	9.42	6.51	106.6	20.6
	TLC-A	9.87	6.86	115.8	21.5
	TLC-L	**8.79**	**6.43**	113.9	21.3

Table 4.4: Hallucination rates and captioning metrics on our test set when generating captions with a beam size of 25.

work [53] explicitly optimized for this hierarchical generalization of unknown concepts, whereas here it emerges when considering confidence. TLC-L also avoids misclassification errors, such as "person" or "scissors" in the middle column. On the right column, we show examples influenced by incomplete object annotations. For example, the reference segmentations and captions might overlook the object "table". TLC-L rejects captions that mention "table" in some of these cases, reflecting its training objective where correctness was judged based on faithfulness to the reference distribution. We include additional examples, including several failure cases, in Appendix C.3.

TLC-L with $\text{OFA}_{\text{Large}}$ sets a new state-of-the-art. We compare to previous results on MS COCO object hallucination in Tab. 4.7. We re-train our captioning models and confidence estimators on a dataset split that does not overlap with the Karpathy test split used for evaluation [80]. [149] show that training with a self-critical (SC) loss after training with cross-entropy (XE) [146] can improve captioning metrics, yet worsen hallucination rates compared to training with XE alone. We find that the baseline $\text{OFA}_{\text{Large}}$ has similar hallucination rates for XE and SC, yet TLC-L indeed produces the least hallucinations on top of the XE model. This leads to a new state-of-the-art of 2.0% and 1.4% for CHs and CHi respectively. Notably, TLC-L reduces hallucination without requiring any architecture

Model	Conf.	Vocab Size (↑)	% Novel (↑)	Div-2 (↑)	Re-4 (↓)
$\text{OFA}_{\text{Large}}$	Std.	2822	77.07	6.97	66.34
	TLC-A	**2980**	**78.97**	**7.37**	**64.74**
	TLC-L	2915	77.70	7.13	65.54
OFA_{Base}	Std.	2272	75.43	5.68	71.14
	TLC-A	**2453**	**78.49**	**6.13**	69.28
	TLC-L	2452	77.53	6.03	**69.76**
OFA_{Tiny}	Std.	1130	74.80	2.73	83.29
	TLC-A	1211	75.71	2.91	82.68
	TLC-L	**1243**	**77.05**	**3.01**	**82.12**

Table 4.5: Caption diversity metrics, evaluated on our test set.

Subset	# I	Method	CHs (↓)	CHi (↓)
Full test set	20,252	Standard	2.79	1.78
		TLC-L	**1.74**	**1.17**
TLC-L, $b > 1$	5,401	Standard	6.78	3.22
		TLC-L	**2.81**	**1.61**

Table 4.6: Top: Results on the full test set reported in Tab. 4.4. Bottom: Hallucination rates on a subset of images where TLC-L did not choose the top beam. # I denotes the number of images in each set. Results are shown for $\text{OFA}_{\text{Large}}$.

changes to its captioning model, in contrast to the prior SOTA of COS-Net, where specific modules were introduced to capture image semantics.

4.5 Discussion and Limitations

While TLC-L provides effective confidence estimates for caption generation, it requires domain-specific training data for learning a confidence estimator from scratch on top of captioning model features. TLC-A, on the other hand, uses the captioning model outputs directly, which leverages generalization ability from large-scale pretraining. Thus, TLC-A can be effectively applied in settings where in-domain training data for captioning is not available. To combine these advantages, future research could explore unsupervised methods for learning correctness. Additionally, we use algebraic confidence estimates from uncalibrated output distributions, where output probabilities do not necessarily match actual probabilities of correctness. Potential future work may apply calibration methods to token-level confidence

Reported in	Method	Beam Size	XE Loss B@4	XE Loss S	XE Loss M	XE Loss C	XE Loss CHs (↓)	XE Loss CHi (↓)	SC Loss B@4	SC Loss S	SC Loss M	SC Loss C	SC Loss CHs (↓)	SC Loss CHi (↓)
[149] EMNLP 2018	NBT [116]	5	-	19.4	26.2	105.1	7.4	5.4	-	-	-	-	-	-
[149] EMNLP 2018	TopDown [5] (no Boxes)	5	-	19.9	26.7	107.6	8.4	6.1	-	20.4	27.0	117.2	13.6	8.8
[149] EMNLP 2018	TopDown [5]	5	-	20.4	27.1	113.7	8.3	5.9	-	21.4	27.7	120.6	10.4	6.9
[200] CVPR 2021	Transformer	unk	-	-	-	-	-	-	38.6	22.0	28.5	128.5	12.1	8.1
[200] CVPR 2021	Transformer+CATT	unk	-	-	-	-	-	-	39.4	22.8	29.3	131.7	9.7	6.5
[199] PAMI 2021	UD-DICv1.0	5	-	-	-	-	-	-	38.7	21.9	28.4	128.2	10.2	6.7
[14] WACV 2022	UD-L	no	34.4	20.7	27.3	112.7	6.4	4.1	37.7	22.1	28.6	124.7	5.9	3.7
[14] WACV 2022	UD-L + Occ	no	33.9	20.3	27.0	110.7	5.9	3.8	37.7	22.2	28.7	125.2	5.8	3.7
[108] CVPR 2022	CIIC$_G$	3	37.3	21.5	28.5	119.0	5.3	3.6	40.2	23.2	29.5	133.1	7.7	4.5
[102] CVPR 2022	COS-Net	3	39.1	22.7	29.7	127.4	4.7	3.2	42.0	24.6	30.6	141.1	6.8	4.2
This work	OFA$_{\text{Large}}$ [184]	5	**41.8**	**24.4**	**31.3**	**140.7**	3.1	2.0	**42.3**	**25.5**	**31.6**	**145.0**	3.1	2.0
This work	OFA$_{\text{Large}}$ + TLC-L	5	41.2	24.1	30.9	138.4	***2.0**	***1.4**	42.0	25.2	31.4	143.8	**2.3**	**1.5**

Table 4.7: Comparison to prior work for hallucination in image captioning on the MS COCO Karpathy test split. Although we add a noun parser for our results in Tables 4.4, 4.5, and 4.6, we remove this step here and use the original evaluation provided by [149] to be consistent with prior work. We show captioning metrics B@4 (BLEU [132]), S (SPICE [7]), M (METEOR [91]), and C (CIDEr [178]). * indicates state-of-the-art for hallucination rates.

for improving caption correctness. Finally, learned confidences may also be incorporated into decoding methods that are not autoregressive.

4.6 Conclusion

In this work, we have explored a simple method using Token-Level Confidence (TLC) for determining whether a caption correctly describes an image, a critical part of vision-language understanding. We find that judging caption correctness at a finer granularity than existing approaches leads to improvements in several settings, such as evaluating compositional reasoning with image-caption pairs or reducing object hallucinations in generated captions. To do so, TLC uses a vision-language model fine-tuned on image captioning to produce token confidences, and then aggregates either algebraic (TLC-A) or learned token confidences (TLC-L) over words or sequences to estimate image-caption consistency. Increasing the confidence granularity with TLC-A improves over prior state-of-the-art image and group scores on Winoground [173] by a relative 37% and 9%, respectively, and improves accuracy in verb understanding on SVO-Probes [62] by a relative 10%. When training data are available to learn and calibrate confidences with TLC-L, we reduce object hallucination rates on COCO Captions by a relative 30%, setting a new state-of-the-art. Overall, our results demonstrate that token-level confidence, whether algebraic or learned, can be a powerful yet simple resource for reducing errors in captioning output and assessing image-caption consistency.

Acknowledgements. We thank David Chan, Kate Saenko, and Anastasios Angelopoulos for helpful discussions and feedback. Authors, as part of their affiliation with UC Berkeley, were supported in part by the NSF CISE Expeditions Award CCF-1730628, DoD, including DARPA's LwLL, PTG, and/or SemaFor programs, the Berkeley Artificial Intelligence Research (BAIR) industrial alliance program, as well as gifts from Astronomer, Google, IBM, Intel, Lacework, Microsoft, Mohamed Bin Zayed University of Artificial Intelligence, Nexla, Samsung SDS, Uber, and VMware.

Chapter 5

How Much Do Language Priors Explain Image Caption Hallucinations?

5.1 Introduction

Generating a detailed description of an image requires a model to not only have a strong visual understanding, but also the linguistic skill to produce a coherent and complete summary of the visual information. As a result, many recent vision-language model (VLM) architectures leverage a powerful, pretrained large language model (LLM) directly. However, while the addition of LLMs has enabled the generation of long captions beyond a single sentence, it has also significantly increased the number of factual errors, also known as “hallucinations”.[1]

If one examines captions generated by recent VLMs (such as in Fig. 5.1 or Appendix D.8), one may notice that many hallucinations appear to be “obvious” errors that are clearly not supported by the image. For instance, in Fig. 5.1, there do not seem to be any pixels that could easily have been mistaken for a person. However, when considering only the caption context, it seems plausible that a language model trained for next-word prediction could start describing a person, given a person’s common occurrence around walkways. Additionally, one may notice many subjective statements or interpretations, such as “peaceful and serene atmosphere” or “perfect for a leisurely stroll.” While often not strictly falsifiable, and therefore not quite hallucination, they also seem likely to be grounded less in objective visual input and more in background knowledge from a language model.

In a previous analysis of simple, single-sentence captions, [149] found that object hallucinations were indeed more likely driven by *language priors* – the likelihood of a word conditioned only on preceding context, without any image input – rather than visual mis-

This chapter is based on joint work with Joseph E. Gonzalez, Trevor Darrell, and Kate Saenko.

[1]The term *hallucination* in captioning is sometimes limited to an object that is not present in the scene; here, however, we define it as any statement about the image that is factually incorrect.

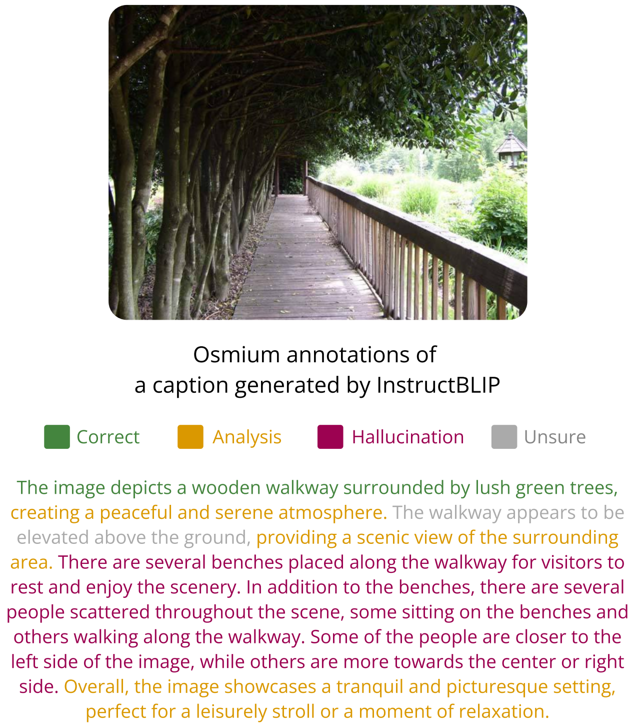


Figure 5.1: Recent VLMs such as InstructBLIP can produce long captions, yet with seemingly "obvious" hallucinations that may align with prior caption context but lack visual support, such as the imaginary benches or people, and subjective analyses such as "perfect for a leisurely stroll." Our work introduces Osmium, an automatic measure to densely label captions, allowing us to assess the impact of language priors on such errors.

classification errors. However, studying hallucinations in long captions from modern VLMs is difficult. Hallucinations are both more complex and verbose, beyond single erroneous objects or attributes. While prior work and qualitative observations support that language priors may impact hallucination [149, 95, 204], their full effect has yet to be explicitly and thoroughly measured.

To do so in this work, we propose **Osmium**,[2] a new measure to *densely* label words in captions on any dataset for statements that are either factually correct, subjective, or hallucinations. Whereas prior work has relied on human annotation for dense labels [57], we are inspired by research in other areas that automate annotation by leveraging powerful LLMs such as GPT-4 as a "judge" [1, 17, 20, 37, 114]. Osmium uses both GPT-4V (Vision) and GPT-4 to annotate spans of text in a caption into a relevant category. This achieves 85% precision for identifying correct statements, where previously no automated benchmark existed. Precision on identifying hallucinations is lower (51%, where random chance is 37%), and thus we do not use Osmium to provide an absolute measurement of hallucination.

[2]Our measure provides dense labels for caption hallucinations. It is named after Osmium, the densest naturally-occurring element.

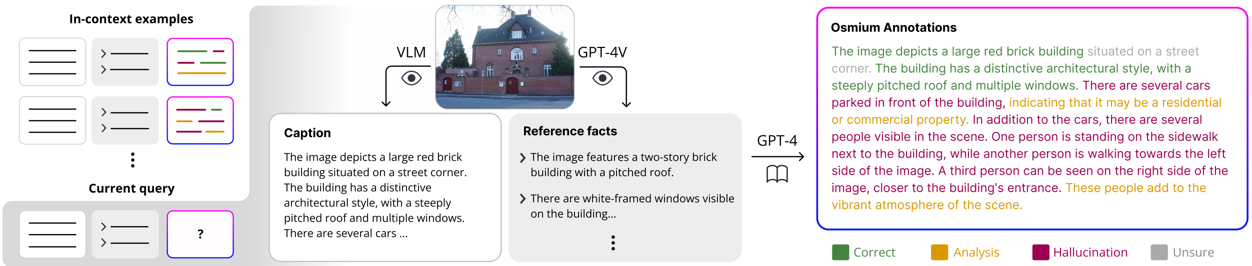


Figure 5.2: Osmium: Our proposed automatic measure to densely label spans of text within captions. Given an image, we first use GPT-4V as an oracle to extract a list of succinct, yet comprehensive, reference facts that describe the scene. These reference facts can be re-used for annotating different captions of the same image. Next, we give a set of completed, in-context example annotations to the language-only GPT-4 to demonstrate the dense labeling task, and append the VLM caption to be annotated and the GPT-4V references. GPT-4 then completes the query, using the reference facts to label spans of text within the caption into Correct, Analysis, Hallucination, or Unsure categories.

However, we find that our analyses around language priors nevertheless hold (Fig. 5.5), even providing underestimates of the true effect, as Osmium labels are conservative – the primary source of error is the misidentification of correct words as hallucinations. We encourage the reader to view examples of Osmium annotations in Appendix D.8.

The dense labels from Osmium unlock the ability to statistically analyze the correlation between a VLM's language prior and long caption hallucinations. To measure this correlation at a given token, we introduce Φ, a simple score that computes the difference in VLM output with and without image conditioning. We test Φ's effectiveness in distinguishing correct from hallucinated words, e.g., by measuring average precision across a range of thresholds. Intuitively, if removing the image does *not* change the VLM's prediction (a small Φ), it suggests that the prediction may have relied more on language cues than visual evidence. This does not necessarily need to be a hallucination (e.g., following the language prior is useful in cases such as syntax) – we discuss this further in Sec. 5.4. Nevertheless, *despite* these confounding variables and conservative labeling from Osmium that make it less likely for Φ to classify hallucinations, we *still* find that Φ reaches up to about 24% higher average precision on ADE20K than scores previously used to measure likelihood of correctness, such as softmax or entropy [75, 135]. We additionally find that Φ explains about *one-third* of hallucinations in recent VLMs, and nearly 60% of subjective, analytical statements.

Our work experimentally confirms that language priors strongly correlate with VLM hallucination on long captions through an explicit analysis. While LLMs can be useful tools for improving linguistic skill and instruction following in text generation, they must be incorporated more carefully into multimodal models to prevent language priors from introducing hallucination. We additionally develop the first automated measure to densely label long captions for hallucinations, opening new dimensions for hallucination research.

5.2 Related Work

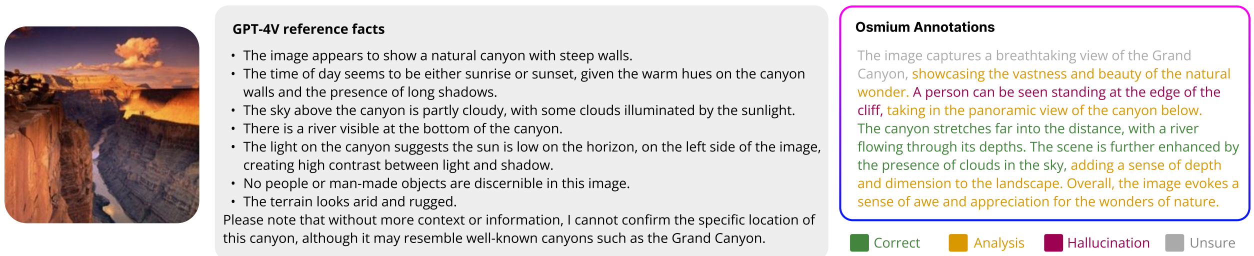


Figure 5.3: An example image from ADE20k captioned by LLaVA/Vicuna-7B. We show the outputs of Osmium: the list of reference facts from GPT-4V (grey box), and caption annotations from GPT-4 (displayed as colors in the caption). The reference facts express uncertainty about the location of the image, leading GPT-4 to annotate a span of text in the caption as Unsure.

Evaluating hallucination in VLMs. [149] pioneer a benchmark for object hallucination in captions, CHAIR, limiting evaluation to a fixed set of objects on COCO data [105]. More recently, [103] propose POPE to measure object hallucination levels of models, rather than captions. [93] evaluate user-defined criteria on long-form captions, yet produce a single caption-level score rather than dense word-level labels. Other works use LLMs such as GPT-4 [1] to evaluate captions [20, 111]; [207] use GPT-4 to localize object hallucination in long captions, yet the labels are not dense and require ground-truth object sets. Jing et al. [74] break generations into atomic facts and evaluate the correctness of each, and [204] extract a "visual knowledge base" to validate claims in captions; in contrast, we annotate captions directly and densely, and additionally include "unsure" and "analysis" categories. We also address that "detailedness", described as a desirable characteristic in [204] could actually be part of subjective, unwanted "analysis" text. [57] densely annotate spans of text within captions, yet with human annotators. Our evaluation measure, Osmium, localizes caption hallucinations beyond objects, , while producing dense annotations automatically.

Analysis of language priors. In text-only settings, several works improve the correctness of language models during decoding by leveraging the change in LLM output when a conditional input is removed (such as a document in summarization) [139, 163], a similar score as Φ in this work, measuring change in VLM output when the image is removed. [95] take a similar decoding approach to reduce object hallucinations in image captioning. [194] correlate VLM language priors with errors in compositional reasoning tasks, and [106] control the effect of language priors in image-text retrieval. [149] find that object hallucinations in short captions tend to follow language priors. Our work presents a thorough analysis of more recent VLMs on long captions, extending the definition of hallucination beyond objects to any factually inaccurate statement.

VLMs for image captioning. Vision-language models have improved in describing images, from early recurrent and convolutional networks [39, 179, 5] to Transformer-based [177] models [101, 184, 209, 4, 21]. With recent improvements in large-scale LLM pretraining, many VLM architectures combine a pretrained vision encoder, such as CLIP [142], with a pretrained LLM, combining them by training multimodal adaptation layers in between [96, 33, 112, 216, 203]. The linguistic skill from LLMs allowed models to be tuned for open-ended instruction following; yet, our work shows instruction-tuning correlates with the effect of language priors on hallucination and subjective analysis within captions.

5.3 Densely Labeling Captions for Hallucinations

Our goal is to annotate the correctness of spans of text within an image caption. Each annotation assigns a span to one of the following categories: **Correct** – consistent with the image, **Analysis** – subjective or represents an interpretation about the image, **Hallucination** – misleading or factually inaccurate description of the image, or **Unsure** – unclear if the description is correct. Note that we do not require all parts of a caption to be annotated. Specifically, given an image x and a caption $w_{1:n}$ with n words, we produce a list $\mathcal{Y}(x, w_{1:n})$. Each of i annotations per caption is of the form:

$$\begin{aligned} \mathcal{Y}_i(x, w_{1:n}) &= (w_{a:b}, y_i) \\ \text{s.t.} \quad 1 &\leq a < b \leq n, \\ y_i \in \{&\textsc{Correct}, \textsc{Analysis}, \textsc{Incorrect}, \textsc{Unsure}\} \end{aligned} \tag{5.1}$$

We are inspired by a prior work [57] that collected a dataset, MHal-Detect, containing dense labels from *human annotations* on the COCO dataset for captions from a single model [105]. Here, we present an *automated* evaluation measure, Osmium, based on a GPT-4 oracle to obtain these labels on any dataset. We use the human annotations from prior work to validate its accuracy.

Osmium consists of two stages, shown in Fig. 5.2. First, we use GPT-4 Vision (GPT-4V) to extract a set of detailed reference facts about the image. Next, we construct a system prompt to GPT-4, demonstrating the labeling task with instructions followed by several in-context examples – each with an image caption, reference facts, and annotations. We then include the caption and references for the current image, prompting for a final JSON output specifying spans of text with their corresponding labels. Details on the prompts and outputs are in Appendix D.5.

Model-based evaluation. While GPT-4V is itself a VLM, its capabilities far exceed that of the open-source models we use it to evaluate, and approach (if not also exceed) that of a standard human annotator in terms of accuracy. We do not study its own language prior as it is not open-source. Although GPT-4V and GPT-4 are powerful tools, note that they are not immune to hallucination themselves. However, we thoroughly compare Osmium labels to human annotations in Sec. 5.5, and find that they correlate well and preserve relationships in our language prior analysis. We discuss the use of GPT-4(V) further in Sec. 5.7.

Stage 1: Obtaining reference facts. We find that a simpler, single-stage of annotation does not work well in practice, perhaps due to the difficulty of in-context learning with multiple images and multimodal reasoning. Instead, for each image, we prompt GPT-4V to produce a list of succinct, yet comprehensive, facts describing the scene that will then be used by GPT-4 to annotate the correctness of the caption. We prompt the model to describe portions of the image that likely appear in image captions, such as the background and any objects or people that are present. We also emphasize the expression of uncertainty – e.g., in Fig. 5.3, GPT-4V suggests that the image may be of the Grand Canyon, but it is not sure. There are no in-context examples for this stage – we simply input the image and a short prompt (see Fig. D.7 in the Appendix).

Stage 2: Obtaining caption hallucination annotations. The second stage does not include images. We specify instructions for the labeling task in a system prompt to GPT-4 (Fig. D.8) to produce annotations in JSON format. We follow this with a sequence of eight annotated examples for in-context learning: as shown in Fig. 5.2, each example is a triplet of *(image caption, reference facts, caption annotations)*. The images and captions are sourced from the MHal-Detect Dataset proposed in [57]; we add reference facts from our Stage 1 procedure and re-annotate the captions ourselves, ensuring that the labels can be determined from the facts alone. We end the prompt with the current query: *(image caption, reference facts)*, letting the LLM perform the completion. We provide details of the output annotation format in Appendix D.5, and several in-context examples in Fig. D.12.

5.4 Defining an Analysis Framework

For each token within a caption, we measure how much the VLM prediction agrees with its own language prior. Then, using the dense labels from Osmium, we measure how well this score correlates with words in captions that are Correct, Analysis, or Hallucination.

Preliminaries. We define an image caption as a sequence of m tokens $t_{1:m}$ autoregressively generated by a vision-language model f – that is, $t_i = f(x, s, t_{1:i-1})$ for an image x and task specification s (such as *"describe the image in detail"*). We denote the language prior at a particular time step as $t_i^L = f(s, t_{1:i-1})$ – the output distribution without any image conditioning.

Measuring Language Priors per Token

To measure the amount that a language prior agrees with a VLM prediction at given token t_i, we propose a score Φ:

$$\Phi(t_i) = d(t_i, t_i^L) \tag{5.2}$$

d is an algebraic function that computes some measure of difference between the image-conditioned output t_i and language prior t_i^L. A small Φ indicates that removing the image does not cause much change in output, and thus the model prediction follows the language prior. On the other hand, if the output changed a lot (large Φ), the prediction is more likely to be visually grounded, and perhaps more likely to be correct.

We consider several choices of d, such as KL-Divergence or difference in logit or softmax scores. From our analysis in Appendix D.3, we select d to be the difference in logit scores at a given vocabulary index k_i corresponding to the token that was selected in the caption (Eq. 5.3). We use Φ throughout the work to refer to this difference in logit value. We include more details and further discussion on why the difference in logits may have performed best in Appendix D.3.

$$\Phi(t_i) = d(t_i, t_i^L) \equiv t_i[k_i] - t_i^L[k_i] \tag{5.3}$$

Correlating Language Priors with VLM Error

For a given set of captions from a VLM, we first compute $\Phi(t_i)$ for all predicted tokens t_i across all captions in a dataset. Then, we separate these values into those that correspond to each class $y \in \{$Correct, Analysis, Hallucination$\}$ using our word-level labels, y_j, for each of N words w_j in the dataset, excluding words without a label or Unsure. For words that are multiple tokens long (e.g., "furn" + "iture"), we take Φ from its first token to avoid effects from the language prior typically guiding subsequent tokens (see Sec. 5.7) – that is, if $w_j = [t_a, t_b]$, then $\Phi(w_j) = \Phi(t_a)$.

Recall that a large Φ means that adding the image caused a large *disagreement* with the language prior – we use this as a proxy for "visual groundedness" and thus "more likely to be correct". If this was indeed a perfect classifier for correctness, we could find a threshold γ such that $y_j \equiv$ Correct $\iff \Phi(w_j) \geq \gamma$ for each word w_j. While this is not the case in practice, measuring the degree to which Φ can classify Correct words is useful – the better it is, the more that "agreement with the language prior" is informative for "not Correct".

To measure this, we compute **Average Precision (AP)** for labeling Correct words using Φ as the classifier score. Specifically, for a label z_j, a word w_j is labeled $z_j = 1$ if it is Correct, and $z_j = 0$ if it belongs to a set of negative categories $\mathcal{N}_y$. In our main experiments, we use $\mathcal{N}_y = \{$Hallucination, Analysis$\}$ as both of these categories hint at being driven by language, yet we separately evaluate AP with $\mathcal{N}_y = \{$Hallucination$\}$ and $\mathcal{N}_y = \{$Analysis$\}$ as well. **AP(Φ)** measures the average precision across these binary labels using $\Phi(w_j)$ as the likelihood $\hat{z}_j$ that w_j has label 1. The **chance-level** AP is equivalent to the fraction of labels where $z_j = 1$, that is, the captioning **model accuracy** in generating Correct words in captions. We use the latter two terms interchangeably, and provide a definition of average precision in Appendix D.3.

Comparing across models

While a higher AP indicates a better confidence ranking given a single set of labels, it is important to consider how much this improves over chance level, or model accuracy. E.g., if VLM$_A$ has 90% accuracy and AP(Φ) = 91%, then Φ is not much better of a predictor than simply $\hat{z} = 1$ for all words. Yet, if VLM$_B$ has 50% accuracy and AP(Φ) = 91%, Φ is much more informative (and thus its language prior is more informative for correctness).

Therefore, in order to compare Φ across models with different accuracies, we compute the percentage of model error it is able to improve over chance level (where error = 1 − model accuracy). We define this in Eq. 5.4 as **Model Error Explained (MEE)**, similar to the Performance Gap Recovered proposed by [18].

$$\mathrm{MEE}(\Phi) = \frac{\overbrace{\mathrm{AP}(\Phi) - \text{model accuracy}}^{\text{Improvement from the language prior}}}{\underbrace{1 - \text{model accuracy}}_{\text{Total model error}}} \tag{5.4}$$

For example, MEE(Φ) for VLM$_A$ would be $(91-90)/(100-90) = 10\%$, and for VLM$_B$ $(91-50)/(100-50) = 82\%$. Note that when model accuracy is computed using word labels z_j estimated by Osmium, MEE is an estimate as well.

5.5 Experiments

Setup

VLM Models. We use several recent VLMs that all contain an explicit LLM component. BLIP-2 [96] and InstructBLIP [33] variants take a pretrained vision encoder (CLIP [142]) and LLM and train a Q-Former in between to map vision embeddings into the LLM token embedding space. We choose these models because the LLM remains frozen during the multimodal training – the language prior is exactly "intact" from the LLM's text-only pretraining. We additionally include LLaVA models [112], which similarly train a multimodal mapping between a pretrained vision encoder and LLM, yet the LLM itself is finetuned during this process. The VLMs we select include a range of LLM sizes, from 2.7B to 13B parameters.

Caption generation. We generate captions using beam search with five beams. BLIP-2 models generate short, usually single-sentence captions based on a fixed prompt. InstructBLIP and LLaVA models are trained for instruction-following abilities. Thus, we use prompts to specifically query for *long* captions, setting the maximum generation length to 200 tokens. Although most multimodal training data do not have long captions, InstructBLIP and LLaVA can nevertheless follow instructions to, e.g., "describe this image in detail" [33, 112]. Appendix D.6 provides additional details on experiment setup.

How Well Does Osmium Annotate Captions?

In this section, we validate the accuracy of Osmium annotations against a subset of human annotations from MHal-Detect, with about 2900 captions on 720 images in the COCO validation set. These captions were already generated in prior work using InstructBLIP [33] with Vicuna-7B as the LLM [24]. We ensure that none of the images appear in the examples we use for in-context learning. Fig. 5.4 shows the corresponding precision and recall. We exclude UNSURE and unlabeled words from MHal-Detect when computing precision, and include all classes when computing recall. Fig. D.3 in the Appendix shows the confusion matrix, on a word level, between MHal-Detect and Osmium labels.

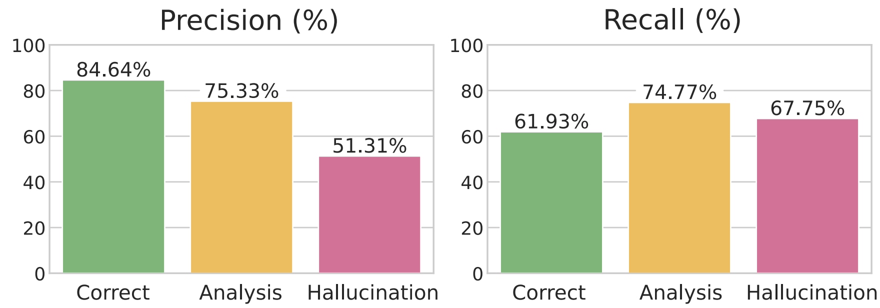


Figure 5.4: Precision and recall for Osmium labels on MHal-Detect, where the ground-truth labels are human annotations.

Osmium labels CORRECT and ANALYSIS words well. The task of densely labeling captions is inherently difficult and ambiguous; e.g., a hallucination could be annotated as just a single word, or include the whole phrase. Nevertheless, our measure reaches 85% precision on words it predicts as CORRECT and 75% precision for ANALYSIS.

Osmium labels are conservative. Our measure reaches 51% precision on HALLUCINATION (random chance is 37%). It confuses about 40% of its predicted HALLUCINATION with words that are actually CORRECT. It is therefore *conservative* in labeling words as CORRECT, with fewer false positives (7%, samples that are HALLUCINATION yet predicted as CORRECT) than false negatives (40%). This is also supported by the high percentage of UNSURE predictions that are actually CORRECT (69%). Because the precision for HALLUCINATION is low, we do not use Osmium to make claims about *absolute* levels of correctness. However, our predicted labels are nevertheless sufficient to make claims about AP and MEE, as we explain next.

Labels from Osmium underestimate language prior impact. Here, we test how error in our predicted labels affects our analyses. In Fig. 5.5, we plot the precision-recall curve for two classifier scores to measure $p(\text{CORRECT})$, as discussed in Sec. 5.4: logit and Φ, under both human annotation and Osmium labels. The two scores have similar curves between the two label sets, separated by a consistent drop in precision that preserves ranking. Next, Tab. 5.1 compares the difference in AP(Φ) and MEE(Φ). We find that Osmium underestimates both of these metrics that measure the impact of language priors on caption correctness – for instance, MEE(Φ) drops by a relative 26.25%. The drop is likely caused by the false negative rate, lowering the performance of Φ in classifying CORRECT samples. This suggests that our findings in the next section, where we show significant language prior impact on several VLMs, are likely underestimates of the true impact, and that Osmium's behavior is consistent with human annotations.

Metric	Annotation Source		(Relative %Δ)
	Humans	Osmium	
AP(Φ)	74.38	61.72	(−17.02%)
MEE(Φ)	44.26	32.64	(−26.25%)

Table 5.1: AP(Φ) and MEE(Φ), measures of language prior impact, are underestimated by Osmium labels compared to human annotations on a subset of MHal-Detect.

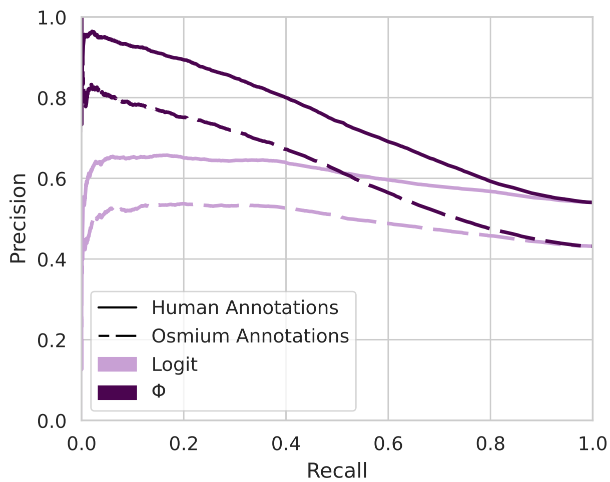


Figure 5.5: Under two sets of labels, we compare the behavior of two scores for classifying words as Correct: the token logit from the VLM, and Φ, the token-level disagreement with the language prior. Importantly, although the absolute scores vary, Osmium labels preserve the same relationship between the two scores as do human annotations.

Comparing Language Prior Effect Across Different VLMs

We now benchmark the effect of language priors on several recent VLMs. We select a subset of about 460 images from ADE20K [214], which contains cluttered, complicated scenes that provide interesting images to caption and is not used for training any VLMs we study. For Osmium, we run Stage 1 once per image and reuse the same reference facts for each caption in Stage 2. We compare Φ to several baseline measures that are typically used for confidence or model correctness: (negative) entropy, softmax score, and logit. See Appendix D.7 for more details. Note that while the relative relationships hold, as was shown in Fig. 5.5 and Tab. 5.1, we are careful not to claim that $1 - p(\text{Correct})$ in Tab. 5.2 is a VLM's exact rate of error, given Osmium's lower precision for Hallucination words.

Φ is a better predictor for correctness than baseline confidences. In Tab. 5.2, we compute the AP for identifying words labeled as Correct vs. $\mathcal{N}_y = \{\text{Hallucination, Analysis}\}$.

Confidence	BLIP-2		InstructBLIP				LLaVA	
	OPT (2.7B)	OPT (6.7B)	FlanXL (3B)	FlanXXL (11B)	Vicuna (7B)	Vicuna (13B)	Vicuna* (7B)	Vicuna* (13B)
Chance level (% of words labeled CORRECT)	84.29	82.39	37.58	46.00	35.20	36.66	37.96	38.98
−Entropy	86.29	82.66	39.90	48.81	36.55	37.73	35.12	36.13
Softmax	86.47	83.49	40.12	49.11	40.01	40.88	35.50	36.67
Logit	85.62	81.57	41.68	48.72	39.45	40.80	33.96	34.54
Φ (Ours)	**86.91**	**85.90**	**46.34**	**53.04**	**55.09**	**57.36**	**58.64**	**60.88**

Table 5.2: Average precision (%) for identifying words labeled as CORRECT, versus HALLUCINATION or ANALYSIS words, by Osmium on a subset of ADE20k. We also include the **chance level** of AP, i.e., the percent of samples that are labeled as CORRECT (this is the VLM's own accuracy for generating CORRECT words). * indicates that the LLM component of the VLM was finetuned during multimodal training. For all models, Φ is a better predictor of CORRECT words compared to typical scores used to measure model confidence – meaning that the language prior is informative for correctness.

For each VLM, Φ outperforms all baselines, reaching up to about 24% improvement over baselines for LLaVA/Vicuna-13B models.

Language priors explain about one-third of error in recent instruction-tuned VLMs. In Fig. 5.6, we plot MEE(Φ), the percentage of VLM error (Hallucination and Analysis words) that is explained by Φ. Going left to right in LLM model families in the figure, the amount of LLM instruction tuning roughly increases: OPT models are not instruction-tuned [211], FlanT5 models are tuned on many classification or multiple-choice benchmarks [30], Vicuna in InstructBLIP is tuned on open-ended conversations [24], and Vicuna in LLaVA is additionally finetuned on open-ended multimodal instruction data [112]. Fig. 5.6 shows that this correlates with the percent of model error explained by language priors. In fact, language priors explain a large 31-36% in error for the most recent Vicuna-based models, which is likely even an underestimate, as discussed earlier in this section.

LLM scale increases language prior impact, except for FlanT5-based models. For most model families tested, increasing the scale of LLM also increases the negative impact of language priors, albeit slightly. E.g., increasing Vicuna size by 5B parameters in InstructBLIP/LLaVA results in a 2.0-3.2% increase in MEE(Φ). However, FlanT5$_{\mathrm{XL}}$ and FlanT5$_{\mathrm{XXL}}$ have the largest parameter difference of 8B, yet MEE(Φ) actually *decreases* by 1.0% when scaling the LLM. This encouraging observation presents a direction for future work: exploring how VLMs can leverage scale without negatively driving multimodal generation, possibly with a certain form of instruction tuning.

Language priors have more effect on ANALYSIS words than HALLUCINATION. Whereas Fig. 5.6 compared MEE(Φ) with the classification of CORRECT words against a negative label set $\mathcal{N}_y$ = {HALLUCINATION, ANALYSIS}, Fig. 5.7 separates this into $\mathcal{N}_y$ = {HALLUCINATION} and $\mathcal{N}_y$ = {ANALYSIS}. We exclude BLIP-2 models, as very few ANALYSIS words arise without instruction tuning. We find that the effect of language priors is extremely pronounced on ANALYSIS words, even explaining over 50% of cases for

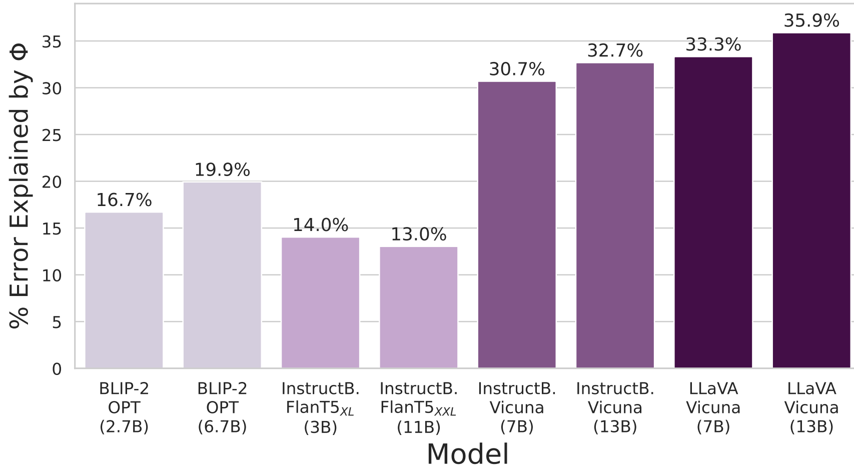


Figure 5.6: We show MEE(Φ) scores on ADE20K: the percentage of VLM error explained by Φ in predicting CORRECT words versus HALLUCINATION or ANALYSIS words. The higher MEE(Φ) for a specific VLM, the more its language prior explains its error.

Vicuna-based models.

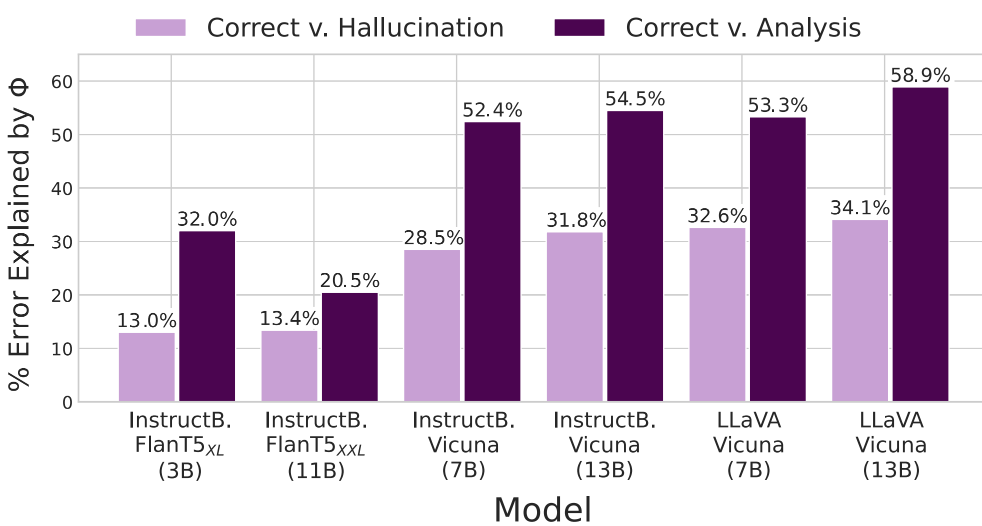


Figure 5.7: We compare MEE(Φ) on ADE20K for predicting CORRECT vs. HALLUCINATION words and CORRECT vs. ANALYSIS words. We omit BLIP-2 models as the captions contain very few ANALYSIS words.

5.6 Conclusion

In this paper, we explore the influence of the language model on VLM hallucinations. We propose the first automated measure, Osmium, to densely label not just objects but entire spans of text within captions for correct, subjective, and hallucinated words. We then propose a score Φ: the difference in token logit value between the original, image-conditioned output and when the image is removed. Φ can classify Osmium-labeled correct words with up to 24% AP over common confidence measures such as softmax or entropy. To compare across models, we propose MEE(Φ), the percent of model error that is explained by agreement with language priors. On ADE20K, language priors explain about one-third of hallucination for recent VLMs, and even about 60% of cases for subjective (analysis) phrases. Our dense annotation approach Osmium and our metric for the correlation of language priors on hallucinated or subjective words are valuable tools for driving new dimensions of hallucination research.

5.7 Limitations

Confounding variables. Several factors confound "following the language prior" with "high likelihood of hallucination". First, language priors aid in forming correct syntax or multi-token words, such as completing "furn" with "iture". Second, a prediction that is strongly grounded in the image may still agree with the language prior for common scenes – e.g., "a dog carrying a" is likely to be completed with "stick" under a language prior, which may be the same as a visually-grounded prediction. Similar to the conservative labeling by Osmium, these confounding variables make it more difficult for Φ to classify Correct words – yet, we find that it nevertheless explains a large fraction of VLM error. Even though the language prior can legitimately agree with a visually-grounded prediction, it is still useful to understand if it disproportionately affects errors.

What about hallucinations from GPT itself? GPT-4V may hallucinate a reference "fact" or omit a detail that the downstream image caption (correctly) includes. GPT-4 may misinterpret the task, overlook a relevant detail in the reference facts, or introduce false details itself. However, we validate the agreement of Osmium with human annotations in Sec. 5.5, finding that the largest source of error comes from labels that we predict to be HALLUCINATION, yet are actually CORRECT – this conservative behavior supports our analysis, giving lower bounds for the correlation of language priors on hallucination.

Why use closed-sourced models for annotation? While GPT-4 models are powerful evaluation tools, it is important to consider the limitations that they are not transparent, possibly inconsistent between runs, and have a cost associated with evaluation. Despite open-source models typically being free from these limitations, our dense caption annotation task is challenging, and recent open-source VLMs have high hallucinations themselves. Osmium could be modified to use stronger, open-source models in the future, perhaps as ensembles. We believe that GPT-4 models should currently be viewed as near-replacements for human annotation – humans themselves are not transparent in their decision-making, can give inconsistent labels, and should often be compensated for annotation effort. For Osmium, Stage 1 cost roughly \$0.00764 per image (producing reusable reference facts between models), and Stage 2 about \$0.0674 per caption – while individual resources may vary, Osmium is likely more cost-effective than human annotations.

Acknowledgements. Authors were supported in part by the NSF, DoD, and/or the Berkeley Artificial Intelligence Research (BAIR) industrial alliance program. Sky Computing Lab is supported by gifts from Accenture, AMD, Anyscale, Google, IBM, Intel, Microsoft, Mohamed Bin Zayed University of Artificial Intelligence, NVIDIA, Samsung SDS, SAP, and VMware.

Chapter 6

Conclusion

This thesis has made significant strides towards understanding and improving the reliability of vision and language models. My work has spanned reliability around the tasks of image classification, visual question answering, and image captioning. The methods presented are simple and thus practical for scaling, requiring little manual annotation effort, leveraging pretrained models, and using stable loss functions. These works have approached reliability from several axes: addressing visual bias, learning multimodal confidence estimators, abstaining from answering incorrectly, reducing hallucination, and investigating caption error via language bias.

In **Chapter 2**, we have reduced the impact of visual bias, or spurious correlations, in training data by guiding an image classifier using language specification of the task. We proposed a method, GALS, that grounds task-relevant words or phrases with attention maps from a pretrained large-scale model (CLIP). We then used this grounding to supervise a classifier's spatial attention away from distracting context. We showed that supervising spatial attention in this way improved performance on classification tasks with biased and noisy data, especially in image groups that had particularly strong biases in training.

The next two chapters focused on learned confidence estimators for multimodal tasks, as well as methods using confidences to improve reliability. First, **Chapter 3** proposed the first framework of *abstention* in the context of VQA: enabling a model to say *I don't know* to questions that it would have answered incorrectly. Otherwise, providing an incorrect answer could lead to negative consequences, especially if a user has placed trust in the model and makes decisions or actions based on its predictions. The framework evaluated abstention performance in two ways. One is risk and coverage, where the user specifies an acceptable level of risk, such as tolerating no more than a 5% error rate in predictions. A confidence threshold is then determined so that predictions with confidence levels below this threshold are abstained from, while those above it are accepted, ensuring that the overall error rate does not exceed the specified risk percentage. Then, models are evaluated based on coverage, or the fraction of questions answered to ensure the specified risk. The second is a metric we proposed called Effective Reliability, which assigns a cost to each question that is answered incorrectly or abstained, and a reward if answered correctly. We found that training an auxiliary head with regression to the base model's VQA Accuracy on a validation set produced better confidence estimates than softmax score, even after running a calibration procedure.

Second, **Chapter 4** proposed Token-Level Confidence, or TLC, as a simple yet surprisingly effective method to assess caption correctness. Specifically, we fine-tuned a vision-language model on image captioning, input an image and proposed caption to the model, and aggregated either algebraic or learned token confidences over words or sequences to estimate image-caption consistency. When training data are available, we proposed a method to learn token confidences over a validation set – although similar in spirit to the classification-based VQA method in the previous chapter, the autoregressive nature of captioning leads to quite a different training procedure. Along with this method, we proposed a simple procedure to select a caption beam from beam search that was less likely to contain a hallucination. We showed that this procedure combined with our learned confidence estimator set a new state-of-the-art in object hallucination rates on the MS COCO Captions benchmark.

Finally, in **Chapter 5**, we turned to investigating the source of hallucination itself within VLMs that have an LLM component. These recent VLMs produce image captions that are long and detailed, yet prone to errors known as hallucinations. The causes of hallucination in image captioning were under-explored and difficult to study due to limited evaluation tools. In this chapter, we developed Osmium, an automated measure to densely label words in image captions for hallucinations. Using these dense labels, we then explored the question: *how much do language priors within a VLM explain hallucinations?* To do so, we computed the VLM output without image conditioning (the language prior) and measured its difference with the image-conditioned output at each token. We found that using this score as a measure of confidence is significantly better than typical confidence measures such as logit or entropy. We also found that agreement with language priors explained around *one-third* of model error for recent instruction-tuned VLMs. Subjective statements within captions such as inferring aesthetics or emotion were especially affected, with language priors explaining nearly 60% of cases. Our analysis suggested that the increasingly common practices of instruction-tuning and incorporating large language models within VLMs must be used carefully to prevent hallucination in multimodal text generation.

Extensibility. Many of the methods I have presented are agnostic to the specific modalities used, or can be easily extended to other models; for instance, TLC can be used in a language-only setting, or with the addition of audio or video modalities, as long as the model autoregressively predicts tokens. GALS placed a loss over CNN attention maps, yet could also be used with differentiable attention maps for ViTs. This extensibility has been a core value of this thesis – methods should be applicable to a broad class of models, and "plug and play" with data or models as much as possible.

Broader limitations. Within each of these chapters, we have discussed the limitations of each proposed method and manners in which they can be extended or improved. There are also current limitations more broadly that will be promising and exciting for the field to address in the years to come. One clear issue is hallucination in generated text. Even industry-scale models can sometimes hallucinate details in image captions or responses to visual questions. While scale (data and compute) alleviates error rates, it is unclear whether it is enough to guarantee a level of factuality in high-risk scenarios; it will likely remain necessary to have specific quality checks, such as a human in the loop, in these settings. Another interesting and under-explored area is the improvement of the communication of uncertainty to users. Guiding a user to an appropriate level of trust for specific predictions can avoid negative consequences from incorrect outputs. We have introduced the framework of abstention to address this within VQA, yet there are many different forms this can take depending on the model and application – for example, communicating uncertainty in an image description to a person with visual impairments. There has been a growing trend to treat different modalities in a unified fashion; for instance, models may project image, video, audio, and/or language tokens into a shared space to reason over, even while interleaving different modalities. As we have began to investigate in Chapter 5, tracing sources of error back to a specific modality can uncover valuable insights into these models, despite having modalities in a unified space. One part of the input may be biasing the output in particular scenarios (e.g., the language prior in image captioning). As multimodal models expand in capabilities, this becomes an increasingly important area of analysis. Although I mention these limitations only briefly here, they each warrant extended discussion and research, and I look forward to the continued development of the field.

Closing thoughts. There are many applications involving machine learning where improving the capability of a model is not enough, as measured by performance of its intended task on some test set. Ensuring ongoing *reliability* – operation without failure – is often crucial, yet even defining the specific parameters of reliability for a given application might not be straightforward. Nevertheless, the incorporation of AI into society is only increasing, and care must be taken. In this thesis, I have presented many practical methods, evaluations, and frameworks for the reliability of vision and language models. The future of multimodal models is optimistic; I look forward to the continued development of the field, and beneficial, reliable AI.

Bibliography

[1] Josh Achiam et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[2] Ehsan Adeli et al. “Representation learning with statistical independence to mitigate bias”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 2513–2523.

[3] Aishwarya Agrawal et al. “Don’t just assume; look and answer: Overcoming priors for visual question answering”. In: *CVPR*. 2018.

[4] Jean-Baptiste Alayrac et al. “Flamingo: a visual language model for few-shot learning”. In: *arXiv preprint arXiv:2204.14198* (2022).

[5] Peter Anderson et al. “Bottom-up and top-down attention for image captioning and visual question answering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6077–6086.

[6] Peter Anderson et al. “Guided open vocabulary image captioning with constrained beam search”. In: *arXiv preprint arXiv:1612.00576* (2016).

[7] Peter Anderson et al. “Spice: Semantic propositional image caption evaluation”. In: *European conference on computer vision*. Springer. 2016, pp. 382–398.

[8] Anastasios N Angelopoulos and Stephen Bates. “A gentle introduction to conformal prediction and distribution-free uncertainty quantification”. In: *arXiv preprint arXiv:2107.07511* (2021).

[9] Anastasios N Angelopoulos et al. “Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control”. In: *arXiv preprint arXiv:2110.01052* (2021).

[10] Stanislaw Antol et al. “Vqa: Visual question answering”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433.

[11] Onur Asan, Alparslan Emrah Bayrak, Avishek Choudhury, et al. “Artificial intelligence and human trust in healthcare: focus on clinicians”. In: *Journal of medical Internet research* 22.6 (2020), e15154.

[12] Nilavra Bhattacharya, Qing Li, and Danna Gurari. “Why does a visual question have different answers?” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4271–4280.

[13] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[14] Ali Furkan Biten, Lluís Gómez, and Dimosthenis Karatzas. "Let there be a clock on the beach: Reducing Object Hallucination in Image Captioning". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 1381–1390.

[15] Emily Black, Klas Leino, and Matt Fredrikson. "Selective Ensembles for Consistent Predictions". In: *International Conference on Learning Representations*. 2022.

[16] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101 – Mining Discriminative Components with Random Forests". In: *European Conference on Computer Vision*. 2014.

[17] Sébastien Bubeck et al. "Sparks of artificial general intelligence: Early experiments with gpt-4". In: *arXiv preprint arXiv:2303.12712* (2023).

[18] Collin Burns et al. "Weak-to-strong generalization: Eliciting strong capabilities with weak supervision". In: *arXiv preprint arXiv:2312.09390* (2023).

[19] Jize Cao et al. "Behind the scene: Revealing the secrets of pre-trained vision-and-language models". In: *European Conference on Computer Vision*. Springer. 2020, pp. 565–580.

[20] David M Chan et al. "CLAIR: Evaluating Image Captions with Large Language Models". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore, Singapore: Association for Computational Linguistics, Dec. 2023.

[21] Xi Chen et al. "Pali: A jointly-scaled multilingual language-image model". In: *arXiv: 2209.06794* (2022).

[22] Xinlei Chen et al. "Microsoft COCO captions: Data collection and evaluation server". In: *arXiv preprint arXiv:1504.00325* (2015).

[23] Yen-Chun Chen et al. "UNITER: UNiversal Image-TExt Representation Learning". In: *ECCV*. ECCV. 2020.

[24] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. Mar. 2023. URL: `https://lmsys.org/blog/2023-03-30-vicuna/`.

[25] Tai-Yin Chiu, Yinan Zhao, and Danna Gurari. "Assessing image quality issues for real-world problems". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3646–3656.

[26] Jaemin Cho et al. "Unifying vision-and-language tasks via text generation". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 1931–1942.

[27] Jinwoo Choi et al. "Why Can't I Dance in the Mall? Learning to Mitigate Scene Bias in Action Recognition". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[28] C Chow. "On optimum recognition error and reject tradeoff". In: *IEEE Transactions on information theory* 16.1 (1970), pp. 41–46.

[29] Chi-Keung Chow. "An optimum character recognition system using decision functions". In: *IRE Transactions on Electronic Computers* EC-6.4 (1957), pp. 247–254.

[30] Hyung Won Chung et al. "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416* (2022).

[31] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. "Learning to Model and Ignore Dataset Bias with Mixed Capacity Ensembles". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020.

[32] Charles Corbière et al. "Addressing failure prediction by learning model confidence". In: *Advances in Neural Information Processing Systems* 32 (2019).

[33] Wenliang Dai et al. *InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning*. 2023. arXiv: `2305.06500 [cs.CV]`.

[34] Ernest Davis. "Unanswerable Questions About Images and Texts". In: *Frontiers in Artificial Intelligence* 3 (2020), p. 51.

[35] C. De Stefano, C. Sansone, and M. Vento. "To reject or not to reject: that is the question-an answer in case of neural classifiers". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 30.1 (2000), pp. 84–94. DOI: `10.1109/5326.827457`.

[36] Shrey Desai and Greg Durrett. "Calibration of pre-trained transformers". In: *arXiv preprint arXiv:2003.07892* (2020).

[37] Tim Dettmers et al. "Qlora: Efficient finetuning of quantized llms". In: *arXiv preprint arXiv:2305.14314* (2023).

[38] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[39] Jeffrey Donahue et al. "Long-term recurrent convolutional networks for visual recognition and description". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2625–2634.

[40] Li Dong, Chris Quirk, and Mirella Lapata. "Confidence Modeling for Neural Semantic Parsing". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 743–753. DOI: `10.18653/v1/P18-1069`. URL: `https://aclanthology.org/P18-1069`.

[41] Sayna Ebrahimi et al. "Remembering for the Right Reasons: Explanations Reduce Catastrophic Forgetting". In: *arXiv preprint arXiv:2010.01528* (2020).

[42] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. "Write a classifier: Zero-shot learning using purely textual descriptions". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 2584–2591.

[43] Zhengcong Fei. “Efficient Modeling of Future Context for Image Captioning”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 5026–5035.

[44] Andrea Frome et al. “DeViSE: a deep visual-semantic embedding model”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*. 2013, pp. 2121–2129.

[45] Akira Fukui et al. “Multimodal compact bilinear pooling for visual question answering and visual grounding”. In: *EMNLP*. 2016.

[46] Hiroshi Fukui et al. “Attention branch network: Learning of attention mechanism for visual explanation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10705–10714.

[47] Peng Gao et al. “Dynamic fusion with intra-and inter-modality attention flow for visual question answering”. In: *CVPR*. 2019.

[48] Yonatan Geifman and Ran El-Yaniv. “Selective classification for deep neural networks”. In: *Advances in neural information processing systems* 30 (2017).

[49] Yonatan Geifman and Ran El-Yaniv. “Selectivenet: A deep neural network with an integrated reject option”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2151–2159.

[50] *Getty Images API*. `https://www.gettyimages.com/`.

[51] Taisiya Glushkova et al. “Uncertainty-aware machine translation evaluation”. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2021.

[52] Yash Goyal et al. “Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6904–6913.

[53] Sergio Guadarrama et al. “Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition”. In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 2712–2719.

[54] Nuno M Guerreiro, Elena Voita, and André FT Martins. “Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation”. In: *arXiv preprint arXiv:2208.05309* (2022).

[55] Devin Guillory et al. “Predicting with confidence on unseen distributions”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 1134–1144.

[56] Varun Gulshan et al. “Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs”. In: *JAMA* 316.22 (Dec. 2016), pp. 2402–2410. ISSN: 0098-7484. DOI: `10.1001/jama.2016.17216`. eprint: `https://jamanetwork.com/journals/jama/articlepdf/2588763/joi160132.pdf`. URL: `https://doi.org/10.1001/jama.2016.17216`.

[57] Anisha Gunjal, Jihan Yin, and Erhan Bas. "Detecting and preventing hallucinations in large vision language models". In: *arXiv preprint arXiv:2308.06394* (2023).

[58] Chuan Guo et al. "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1321–1330.

[59] Danna Gurari et al. "Vizwiz grand challenge: Answering visual questions from blind people". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3608–3617.

[60] Blaise Hanczar and Edward R Dougherty. "Classification with reject option in gene expression data". In: *Bioinformatics* 24.17 (2008), pp. 1889–1895.

[61] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[62] Lisa Anne Hendricks and Aida Nematzadeh. "Probing image-language transformers for verb understanding". In: *arXiv preprint arXiv:2106.09141* (2021).

[63] Lisa Anne Hendricks et al. "Women also snowboard: Overcoming bias in captioning models". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 771–787.

[64] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *Proceedings of International Conference on Learning Representations*. 2017.

[65] Matthew Honnibal et al. "spaCy: Industrial-strength Natural Language Processing in Python". In: (2020). DOI: `10.5281/zenodo.1212303`.

[66] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. "Segmentation from natural language expressions". In: *European Conference on Computer Vision*. Springer. 2016, pp. 108–124.

[67] Xiaowei Hu et al. "Scaling up vision-language pre-training for image captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 17980–17989.

[68] Lun Huang et al. "Attention on attention for image captioning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 4634–4643.

[69] Drew A Hudson and Christopher D Manning. "Gqa: A new dataset for real-world visual reasoning and compositional question answering". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6700–6709.

[70] Xu Ji et al. "Test sample accuracy scales with training sample density in neural networks". In: *arXiv preprint arXiv:2106.08365* (2021).

[71] Heinrich Jiang et al. "To Trust Or Not To Trust A Classifier". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: `https://proceedings.neurips.cc/paper/2018/file/7180cffd6a8e829dacfc2a31b3f72ece-Paper.pdf`.

[72] Huaizu Jiang et al. "In defense of grid features for visual question answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10267–10276.

[73] Yu Jiang et al. "Pythia v0. 1: the winning entry to the vqa challenge 2018". In: *arXiv preprint arXiv:1807.09956* (2018).

[74] Liqiang Jing et al. "Faithscore: Evaluating hallucinations in large vision-language models". In: *arXiv preprint arXiv:2311.01477* (2023).

[75] Saurav Kadavath et al. "Language models (mostly) know what they know". In: *arXiv preprint arXiv:2207.05221* (2022).

[76] Kushal Kafle and Christopher Kanan. "An Analysis of Visual Question Answering Algorithms". In: *ICCV*. 2017.

[77] Kushal Kafle and Christopher Kanan. "Visual question answering: Datasets, algorithms, and future challenges". In: *Computer Vision and Image Understanding* 163 (2017), pp. 3–20.

[78] Amita Kamath, Robin Jia, and Percy Liang. "Selective Question Answering under Domain Shift". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5684–5696. DOI: 10.18653/v1/2020.acl-main.503. URL: https://aclanthology.org/2020.acl-main.503.

[79] Siddharth Karamcheti et al. "Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 7265–7281. DOI: 10.18653/v1/2021.acl-long.564. URL: https://aclanthology.org/2021.acl-long.564.

[80] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3128–3137.

[81] Sahar Kazemzadeh et al. "Referitgame: Referring to objects in photographs of natural scenes". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 787–798.

[82] Javed Khan et al. "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks". In: *Nature medicine* 7.6 (2001), pp. 673–679.

[83] Fereshte Khani, Martin Rinard, and Percy Liang. "Unanimous prediction for 100% precision with application to learning semantic mappings". In: *arXiv: 1606.06368* (2016).

[84] Byungju Kim et al. “Learning not to learn: Training deep neural networks with biased data”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9012–9020.

[85] Jinkyu Kim et al. “Advisable learning for self-driving vehicles by internalizing observation-to-action rules”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9661–9670.

[86] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *Proceedings of the International Conference on Learning Representations*. 2015.

[87] Ranjay Krishna et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International journal of computer vision* 123.1 (2017), pp. 32–73.

[88] Aviral Kumar and Sunita Sarawagi. “Calibration of encoder decoder models for neural machine translation”. In: *arXiv preprint arXiv:1903.00802* (2019).

[89] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in neural information processing systems*. Vol. 30. 2017.

[90] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. “Attribute-Based Classification for Zero-Shot Visual Object Categorization”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.3 (2014), pp. 453–465. DOI: `10.1109/TPAMI.2013.140`.

[91] Alon Lavie and Abhaya Agarwal. “METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments”. In: *Proceedings of the second workshop on statistical machine translation*. 2007, pp. 228–231.

[92] Yannick Le Cacheux, Hervé Le Borgne, and Michel Crucianu. “Using sentences as semantic representations in large scale zero-shot learning”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 641–645.

[93] Seongyun Lee et al. “Prometheus-Vision: Vision-Language Model as a Judge for Fine-Grained Evaluation”. In: *arXiv preprint arXiv:2401.06591* (2024).

[94] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. “Predicting deep zero-shot convolutional neural networks using textual descriptions”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4247–4255.

[95] Sicong Leng et al. “Mitigating object hallucinations in large vision-language models through visual contrastive decoding”. In: *arXiv preprint arXiv:2311.16922* (2023).

[96] Junnan Li et al. “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models”. In: *arXiv preprint arXiv:2301.12597* (2023).

[97] Kunpeng Li et al. "Tell me where to look: Guided attention inference network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9215–9223.

[98] Liunian Harold Li et al. "VisualBERT: A Simple and Performant Baseline for Vision and Language". In: *Arxiv*. 2019.

[99] Mengdi Li, Cornelius Weber, and Stefan Wermter. "Neural networks for detecting irrelevant questions during visual question answering". In: *International Conference on Artificial Neural Networks*. Springer. 2020, pp. 786–797.

[100] Wei Li et al. "Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. 2021.

[101] Xiujun Li et al. "Oscar: Object-semantics aligned pre-training for vision-language tasks". In: *European Conference on Computer Vision*. Springer. 2020, pp. 121–137.

[102] Yehao Li et al. "Comprehending and Ordering Semantics for Image Captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 17990–17999.

[103] Yifan Li et al. "Evaluating object hallucination in large vision-language models". In: *arXiv preprint arXiv:2305.10355* (2023).

[104] Jianhua Lin. "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information theory* 37.1 (1991), pp. 145–151.

[105] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[106] Zhiqiu Lin et al. "VisualGPTScore: Visio-Linguistic Reasoning with Multimodal Generative Pre-Training Scores". In: *arXiv preprint arXiv:2306.01879* (2023).

[107] Huan Ling and Sanja Fidler. "Teaching machines to describe images via natural language feedback". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017.

[108] Bing Liu et al. "Show, Deconfound and Tell: Image Captioning With Causal Inference". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18041–18050.

[109] Chenxi Liu et al. "Attention correctness in neural image captioning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 2017.

[110] Evan Z Liu et al. "Just train twice: Improving group robustness without training group information". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 6781–6792.

[111] Fuxiao Liu et al. "Aligning Large Multi-Modal Model with Robust Instruction Tuning". In: *arXiv preprint arXiv:2306.14565* (2023).

[112] Haotian Liu et al. "Visual instruction tuning". In: *arXiv preprint arXiv:2304.08485* (2023).

[113] Weitang Liu et al. "Energy-based out-of-distribution detection". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21464–21475.

[114] Yuan Liu et al. "Mmbench: Is your multi-modal model an all-around player?" In: *arXiv preprint arXiv:2307.06281* (2023).

[115] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[116] Jiasen Lu et al. "Neural Baby Talk". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[117] Jiasen Lu et al. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in neural information processing systems* 32 (2019).

[118] Bernd Lütkenhöner and Türker Basel. "Predictive modeling for diagnostic tests with high specificity, but low sensitivity: a study of the glycerol test in patients with suspected Meniere's disease". In: *PLoS One* 8.11 (2013), e79315.

[119] Haley MacLeod et al. "Understanding blind people's experiences with computer-generated captions of social media images". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 5988–5999.

[120] Aroma Mahendru et al. "The Promise of Premise: Harnessing Question Premises in Visual Question Answering". In: *EMNLP*. 2017.

[121] Utkarsh Mall, Bharath Hariharan, and Kavita Bala. "Field-Guide-Inspired Zero-Shot Learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9546–9555.

[122] Joshua Maynez et al. "On Faithfulness and Factuality in Abstractive Summarization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

[123] D. Harrison Mcknight et al. "Trust in a Specific Technology: An Investigation of Its Components and Measures". In: *ACM Transactions Management Information Systems* 2.2 (July 2011). ISSN: 2158-656X. DOI: 10.1145/1985347.1985353. URL: https://doi.org/10.1145/1985347.1985353.

[124] Sabrina J Mielke et al. "Reducing conversational agents' overconfidence through linguistic calibration". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 857–872.

[125] Masahiro Mitsuhara et al. "Embedding human knowledge in deep neural network via attention map". In: *arXiv preprint arXiv:1905.03540* 5 (2019).

[126] Jesse Mu, Percy Liang, and Noah Goodman. "Shaping Visual Representations with Language for Few-Shot Classification". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

[127] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. "Obtaining well calibrated probabilities using bayesian binning". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[128] Junhyun Nam et al. "Learning from failure: Training debiased classifier from biased classifier". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[129] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. "Movie: Revisiting modulated convolutions for visual counting and beyond". In: *Proceedings of the International Conference on Learning Representations*. 2021.

[130] Alexandru Niculescu-Mizil and Rich Caruana. "Predicting good probabilities with supervised learning". In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 625–632.

[131] Rohan Pandey et al. "Cross-modal Attention Congruence Regularization for Vision-Language Relation Alignment". In: *arXiv preprint arXiv:2212.10549* (2022).

[132] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[133] Jae Sung Park et al. "Exposing the limits of video-text models through contrast sets". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2022, pp. 3574–3586.

[134] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[135] Suzanne Petryk et al. "Simple token-level confidence improves caption correctness". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 5742–5752.

[136] Vitali Petsiuk, Abir Das, and Kate Saenko. "Rise: Randomized input sampling for explanation of black-box models". In: *arXiv preprint arXiv:1806.07421* (2018).

[137] John Platt et al. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in large margin classifiers* 10.3 (1999), pp. 61–74.

[138] Bryan A Plummer et al. "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2641–2649.

[139] Liam van der Poel, Ryan Cotterell, and Clara Meister. "Mutual information alleviates hallucinations in abstractive summarization". In: *arXiv preprint arXiv:2210.13210* (2022).

[140] P. Pudil et al. "Multistage pattern recognition with reject option". In: *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*. 1992, pp. 92–95. DOI: 10.1109/ICPR.1992.201729.

[141] Ruizhi Qiao et al. "Less is more: zero-shot learning from online textual documents with noise suppression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2249–2257.

[142] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *arXiv preprint arXiv:2103.00020* (2021).

[143] Arijit Ray et al. "Question Relevance in VQA: Identifying Non-Visual And False-Premise Questions". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, pp. 919–924.

[144] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[145] Shuhuai Ren et al. "Learning relation alignment for calibrated cross-modal retrieval". In: *arXiv preprint arXiv:2105.13868* (2021).

[146] Steven J Rennie et al. "Self-critical sequence training for image captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7008–7024.

[147] Laura Rieger et al. "Interpretations are useful: penalizing explanations to align neural networks with prior knowledge". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8116–8126.

[148] Anna Rohrbach et al. "Grounding of textual phrases in images by reconstruction". In: *European Conference on Computer Vision*. Springer. 2016, pp. 817–834.

[149] Anna Rohrbach et al. "Object Hallucination in Image Captioning". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018.

[150] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. "Right for the right reasons: Training differentiable models by constraining their explanations". In: *arXiv preprint arXiv:1703.03717* (2017).

[151] Christian Rupprecht et al. "Guide me: Interacting with deep networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8551–8561.

[152] Shiori Sagawa et al. "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2020.

[153] Fawaz Sammani and Mahmoud Elsayed. "Look and Modify: Modification Networks for Image Captioning". In: *British Machine Vision Conference (BMVC)*. 2019.

[154] Fawaz Sammani and Luke Melas-Kyriazi. "Show, edit and tell: a framework for editing image captions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 4808–4816.

[155] Ramprasaath R Selvaraju et al. "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.

[156] Glenn Shafer and Vladimir Vovk. "A tutorial on conformal prediction." In: *Journal of Machine Learning Research* 9.3 (2008).

[157] Meet Shah et al. "Cycle-consistency for robust visual question answering". In: *CVPR*. 2019.

[158] Himanshu Sharma and Anand Singh Jalal. "A survey of methods, datasets and evaluation metrics for visual question answering". In: *Image and Vision Computing* 116 (2021), p. 104327.

[159] Piyush Sharma et al. "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 2556–2565.

[160] Sheng Shen et al. "How Much Can CLIP Benefit Vision-and-Language Tasks?" In: *arXiv preprint arXiv:2107.06383* (2021).

[161] Yilin Shen, Wenhu Chen, and Hongxia Jin. "Modeling token-level uncertainty to learn unknown concepts in SLU via calibrated dirichlet prior RNN". In: *arXiv preprint arXiv:2010.08101* (2020).

[162] Rakshith Shetty et al. "Speaking the same language: Matching machine to human captions by adversarial training". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4135–4144.

[163] Weijia Shi et al. "Trusting Your Evidence: Hallucinate Less with Context-aware Decoding". In: *arXiv preprint arXiv:2305.14739* (2023).

[164] Amanpreet Singh, Vedanuj Goswami, and Devi Parikh. "Are we pretraining it right? digging deeper into visio-linguistic pretraining". In: *arXiv preprint arXiv:2004.08744* (2020).

[165] Amanpreet Singh et al. *MMF: A multimodal framework for vision and language research*. `https://github.com/facebookresearch/mmf`. 2020.

[166] Amanpreet Singh et al. "Towards VQA Models That Can Read". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8317–8326.

[167] Krishna Kumar Singh et al. "Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11070–11078.

[168] Elias Stengel-Eskin and Benjamin Van Durme. "Calibrated Interpretation: Confidence Estimation in Semantic Parsing". In: *arXiv preprint arXiv:2211.07443* (2022).

[169] Hao Tan and Mohit Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 2019, pp. 5100–5111. DOI: `10.18653/v1/D19-1514`. URL: `https://www.aclweb.org/anthology/D19-1514`.

[170] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. "Graph-structured representations for visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1–9.

[171] Damien Teney et al. "Tips and tricks for visual question answering: Learnings from the 2017 challenge". In: *CVPR*. 2018.

[172] Kento Terao et al. "Which visual questions are difficult to answer? Analysis with Entropy of Answer Distributions". In: *arXiv preprint arXiv:2004.05595* (2020).

[173] Tristan Thrush et al. "Winoground: Probing vision and language models for visio-linguistic compositionality". In: *CVPR*. 2022.

[174] Dustin Tran et al. "Plex: Towards reliability using pretrained large model extensions". In: *arXiv preprint arXiv:2207.07411* (2022).

[175] Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. "Towards debiasing NLU models from unknown biases". In: *arXiv preprint arXiv:2009.12303* (2020).

[176] Neeraj Varshney, Swaroop Mishra, and Chitta Baral. "Investigating selective prediction approaches across several tasks in IID, OOD, and adversarial settings". In: *arXiv preprint arXiv:2203.00211* (2022).

[177] Ashish Vaswani et al. "Attention is all you need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 6000–6010.

[178] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. "Cider: Consensus-based image description evaluation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4566–4575.

[179] Oriol Vinyals et al. "Show and tell: A neural image caption generator". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3156–3164.

[180] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

[181] Catherine Wah et al. *The caltech-ucsd birds-200-2011 dataset*. 2011.

[182] Dequan Wang et al. "Tent: Fully test-time adaptation by entropy minimization". In: *arXiv preprint arXiv:2006.10726* (2020).

[183] Jianfeng Wang et al. "GIT: A Generative Image-to-text Transformer for Vision and Language". In: *arXiv preprint arXiv:2205.14100* (2022).

[184] Peng Wang et al. "OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework". In: *Proceedings of the 39th International Conference on Machine Learning*. 2022.

[185] Shuo Wang et al. "On the Inference Calibration of Neural Machine Translation". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.

[186] Tianlu Wang et al. "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.

[187] Wenhui Wang et al. "Image as a foreign language: Beit pretraining for all vision and vision-language tasks". In: *arXiv preprint arXiv:2208.10442* (2022).

[188] Xin Wang et al. "Idk cascades: Fast deep learning by learning not to overthink". In: *arXiv preprint arXiv:1706.00885* (2017).

[189] Xin Wang et al. "TAFE-Net: Task-Aware Feature Embeddings for Low Shot Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[190] Zhen Wang et al. "Explicit Image Caption Editing". In: *European Conference on Computer Vision (ECCV)*. Springer. 2022, pp. 113–129.

[191] Zirui Wang et al. "Simvlm: Simple visual language model pretraining with weak supervision". In: *International Conference on Learning Representations (ICLR)*. 2022.

[192] Spencer Whitehead et al. "Reliable Visual Question Answering: Abstain Rather Than Answer Incorrectly". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2022.

[193] Spencer Whitehead et al. "Separating skills and concepts for novel visual question answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5632–5641.

[194] Chenwei Wu et al. "The Role of Linguistic Priors in Measuring Compositional Generalization of Vision-Language Models". In: *arXiv preprint arXiv:2310.02777* (2023).

[195] Yijun Xiao and William Yang Wang. "On hallucination and predictive uncertainty in conditional language generation". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. 2021.

[196] Ji Xin et al. "The Art of Abstention: Selective Prediction and Error Regularization for Natural Language Processing". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 1040–1051.

[197] Yilei Xiong, Bo Dai, and Dahua Lin. "Move forward and tell: A progressive generator of video descriptions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 468–483.

[198] Wenjia Xu et al. "Attribute Prototype Network for Zero-Shot Learning". In: *34th Conference on Neural Information Processing Systems*. Curran Associates, Inc. 2020.

[199] Xu Yang, Hanwang Zhang, and Jianfei Cai. "Deconfounded image captioning: A causal retrospect". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[200] Xu Yang et al. "Causal attention for vision-language tasks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9847–9857.

[201] Zichao Yang et al. "Stacked attention networks for image question answering". In: *CVPR*. 2016.

[202] Ran El-Yaniv and Yair Wiener. "On the Foundations of Noise-free Selective Classification". In: *Journal of Machine Learning Research* 11 (2010), pp. 1605–1641.

[203] Qinghao Ye et al. "mplug-owl: Modularization empowers large language models with multimodality". In: *arXiv preprint arXiv:2304.14178* (2023).

[204] Shukang Yin et al. "Woodpecker: Hallucination correction for multimodal large language models". In: *arXiv preprint arXiv:2310.16045* (2023).

[205] Jiahui Yu et al. "Coca: Contrastive captioners are image-text foundation models". In: *arXiv preprint arXiv:2205.01917* (2022).

[206] Zhou Yu et al. "Deep modular co-attention networks for visual question answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6281–6290.

[207] Bohan Zhai et al. "Halle-switch: Controlling object hallucination in large vision language models". In: *arXiv e-prints* (2023), arXiv–2310.

[208] Jianming Zhang et al. "Top-down neural attention by excitation backprop". In: *International Journal of Computer Vision* 126.10 (2018), pp. 1084–1102.

[209] Pengchuan Zhang et al. "VinVL: Making Visual Representations Matter in Vision-Language Models". In: *arXiv preprint arXiv:2101.00529* (2021).

[210] Shujian Zhang, Chengyue Gong, and Eunsol Choi. "Knowing more about questions can help: Improving calibration in question answering". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021.

[211] Susan Zhang et al. “Opt: Open pre-trained transformer language models”. In: *arXiv preprint arXiv:2205.01068* (2022).

[212] Jieyu Zhao et al. “Men also like shopping: Reducing gender bias amplification using corpus-level constraints”. In: *arXiv preprint arXiv:1707.09457* (2017).

[213] Bolei Zhou et al. “Places: A 10 million image database for scene recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), pp. 1452–1464.

[214] Bolei Zhou et al. “Scene parsing through ade20k dataset”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 633–641.

[215] Chunting Zhou et al. “Detecting hallucinated content in conditional neural sequence generation”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021.

[216] Deyao Zhu et al. “Minigpt-4: Enhancing vision-language understanding with advanced large language models”. In: *arXiv preprint arXiv:2304.10592* (2023).

[217] Yizhe Zhu et al. *A Generative Adversarial Approach for Zero-Shot Learning from Noisy Texts*. 2018. arXiv: `1712.01381 [cs.CV]`.

[218] Andrea Zunino et al. “Explainable deep classification models for domain generalization”. In: *arXiv preprint arXiv:2003.06498* (2020).

Appendix A

Chapter 2 Supplementary Material

We provide additional details on training and datasets. We also include several qualitative samples of attention from language specification, and compare Vanilla model attention to *GALS* model attention.

A.1 Training Details

All runs were performed on 1-4 NVIDIA GeForce RTX 2080 GPUs. All models were optimized with stochastic gradient descent with a momentum of 0.9. For simplicity, we do not perform any data augmentations. For hyperparameter tuning, we split each dataset into training, validation, and testing, choosing the final hyperparameters based on which maximize validation accuracy. Our final hyperparameters are summarized in Tab. A.2. Language specifications used in the experiments are show in Tab. A.3.

Waterbirds-95%. For the vanilla ResNet50 model, we perform a hyperparameter sweep with batch size 96, and run for 100 epochs. We sweep the backbone learning rate over 0.01, 0.005, 0.001, and 0.0001, and the linear classifier learning rate over 0.1, 0.01, 0.005, 0.001, and 0.0001. We chose a backbone learning rate of 0.01 and classifier learning rate of 0.001. For RRR, using the vanilla model learning rates, we first swept the attention loss weight (λ in Eq. (2.2)) over 1,000, 10,000, and 100,000, as well as the attention loss function over L1 and L2. From this, we chose a λ of 10,000 and an L1 loss. Next, we ran the same learning rate sweep as for the vanilla model, and chose a backbone learning rate of 0.005 and classifier learning rate of 0.0001.

Waterbirds-100%. We use the same hyperparameters found for *Waterbirds-95%.*

MSCOCO-ApparentGender. For the vanilla ResNet50 model, we run a hyperparameter sweep with a batch size of 96 for 100 epochs, testing backbone learning rates of 0.01, 0.005, and 0.001, and classifier learning rates of 0.1, 0.01, 0.005, and 0.001. We chose a backbone learning rate of 0.01 and classifier learning rate of 0.001. For attention weight λ, we test 1,000, 5,000, and 10,000, and choose 10,000 from validation.

Red Meat For the vanilla ResNet50 model, we run a hyperparameter sweep with a batch size of 96 for 50 epochs, testing backbone learning rates of 0.1, 0.01, 0.001, 0.005, 0.001, 0.0001, 0.0005, and classifier learning rates of 0.1, 0.01, 0.001, 0.005, 0.001, 0.0001, and 0.0005. For attention weight λ, we test 100, 1,000, 10,000, and 100,000, and choose 10,000 from validation.

A.2 Dataset Details

Waterbirds variants. For *Waterbirds-95%*, we use the same dataset as provided by the authors of [152]. For *Waterbirds-100%*, we follow the code provided by those authors for generating a new synthetic dataset. Land backgrounds are randomly chosen from the "bamboo forest" and "broadleaf forest" categories in the Places dataset, and water background are from the "ocean" and "natural lake" categories. These categories were determined in [152]. Both dataset variants have $4,795$ training images, $1,119$ validation images, and $5,794$ test images. Tables A.4 and A.5 show the number of samples per class, broken down further by the type of background.

MSCOCO-ApparentGender. For the training set, we begin by using the $22,966$ MSCOCO image ids defined in the Bias split in [212]. We next filter and label these images using a list of "male" words (such as "father", "man", or "groom"), a list of "female" words (such as "daughter", "lady", or "she"), and a list of "person" words which do not have a defined gender (such as "child", "surfer" or "employee") provided by [63]. From these provided lists, we filter out plural words. Next, we filter out images where the annotators do not agree on the gender (at least one caption mentions a male word and at least one caption mentions a female word). We label an image as "Man" if the majority of annotators (3 out of the 5 available captions per image) mention a male word, and "Woman" if the majority mention a female word. We label an image as "Person" if it has not been labeled as "Man" or "Woman" and if the majority of annotators have mentioned a "person" word. We use the same validation and test images for "Man" and "Woman" as in the "Balanced" split defined in [63]. Although these were not labeled in the same manner as our training set, we keep the splits the same to have consistent evaluation with prior work. The number of samples per class is summarized in Table A.1.

Food-101. We start by selecting the 5 red meat classes from the Food-101 dataset [16] and split the 750 training samples into 500 training samples and 250 validation samples,

Split	**Man**	**Woman**	**Person**
Training	10565	4802	2822
Validation	500	500	0
Test	500	500	0

Table A.1: Dataset sizes on *MSCOCO-ApparentGender*.

Dataset	**Method**	**Epochs**	**Batch Size**	**Base LR**	**Classifier LR**	λ
Waterbirds-95%	Vanilla	200	96	0.01	0.001	-
	ABN	200	96	0.01	0.001	-
	UpWeight	200	96	0.01	0.001	-
	GALS	200	96	0.005	0.0001	10,000
Waterbirds-100%	Vanilla	200	96	0.01	0.001	-
	ABN	200	96	0.01	0.001	-
	UpWeight	200	96	0.01	0.001	-
	GALS	200	96	0.005	0.0001	10,000
Waterbirds-100% Backgrounds	Vanilla	200	96	0.01	0.001	-
	GALS	200	96	0.005	0.0001	1,000
MSCOCO-ApparentGender	Vanilla	200	96	0.01	0.001	-
	ABN	200	96	0.01	0.001	-
	UpWeight	200	96	0.01	0.001	-
	GALS	200	96	0.01	0.001	10,000
Red Meat	Vanilla	150	96	0.01	0.001	-
	ABN	150	96	0.01	0.001	-
	GALS	150	96	0.001	0.001	10,000

Table A.2: Hyperparameter details. All models were optimized with SGD using a weight decay of 1*e*-5. “Base LR” refers to the learning rate used for the pretrained ResNet50 backbone, and “Classifier LR” refers to the learning rate used for the linear classifier. λ is the attention loss weight from in Eq. (2.2).

Dataset	**Language specifications**
Waterbirds-95%	“{a photo/an image} of a bird”
Waterbirds-100%	“{a photo/an image} of a bird”
Waterbirds-100% Backgrounds	“{a photo/an image} of a nature scene”, “{a photo/an image} of an outdoor scene”, “{a photo/an image} of a landscape”
MSCOCO-ApparentGender	“{a photo/an image} of a person”
Red Meat	“{a photo/an image} of meat”

Table A.3: Language specifications used for *GALS* in experiments. “{a photo/an image} of X” indicates that two prompts were used: “a photo of X” and “an image of X”.

keeping the 250 sample test set the same. The number of samples per class is summarized in Table A.6.

Attention Samples

In Figures A.1, A.2, and A.3, we show several qualitative examples of spatial attention. Specifically, for sample images from the *Waterbirds-100%*, *MSCOCO-ApparentGender*, and *Food-101* training sets, we show the CLIP ResNet50 GradCAM A^{VL} guidance, as well as the

Split	Landbirds, land	Landbirds, water	Waterbirds, land	Waterbirds, water
Training	3498	184	56	1057
Validation	467	466	133	133
Test	2255	2255	642	642

Table A.4: Dataset sizes on *Waterbirds-95%*. The two classes are "Landbird" and "Waterbird." Furthermore, each image can display either a land background or a water background.

Split	Landbirds, land	Landbirds, water	Waterbirds, land	Waterbirds, water
Training	3694	0	0	1101
Validation	467	466	133	133
Test	2255	2255	642	642

Table A.5: Dataset sizes on *Waterbirds-100%*. The validation and test splits have the same distribution as validation and test in Table A.4 for *Waterbirds-95%*.

Split	Filet Mignon	Filet Mignon	Pork Chop	Prime Rib	Steak
Training	500	500	500	500	500
Validation	250	250	250	250	250
Test	250	250	250	250	250

Table A.6: Dataset sizes on *Food-101*.

RISE attention for the vanilla model and ours. We show that in many cases, our model has learned to attend to similar image features as the language-guided attention. However, when the image is difficult for the language-guided attention to ground the object of interest, then our model can have more difficulty in localization as well.

	Waterbirds 95%		***Waterbirds 100%***	
Method	**Per Group**	**Worst Group**	**Per Group**	**Worst Group**
CLIP Zero-shot	73.18	43.46	73.18	43.46
CLIP Finetune, LogisticReg.	80.58	56.85	68.36	32.15
Vanilla	86.93 ± 0.46	73.07 ± 2.24	69.83 ± 2.04	34.31± 7.31
UpWeight Class	86.74 ± 0.54	73.66 ± 2.00	70.50 ± 2.00	34.82 ± 6.65
ABN	86.01 ± 0.70	65.03 ± 2.77	72.20 ± 3.02	41.56 ± 6.76
GALS	**89.05 ± 0.47**	**76.54 ± 2.40**	**79.72 ± 1.60**	**56.71 ± 3.92**

Table A.7: Test accuracy of approaches on the *Waterbirds-95%* and *Waterbirds-100%* datasets. The percentage indicates the proportion of training samples in each class which have a spurious correlation between the class label and the background.

A.3 Results Tables

Here we provide the tabular results for each of the figures in Sec. 2.4 of the main paper.

	Pointing Game Accuracy		
Method	**Man**	**Woman**	**Average**
Vanilla	51.20	64.40	57.80
ABN	55.80	**69.60**	**62.70**
UpWeight	42.60	57.00	49.80
GALS	**56.20**	67.00	62.60

Table A.8: Pointing game accuracy on *MSCOCO-ApparentGender*.

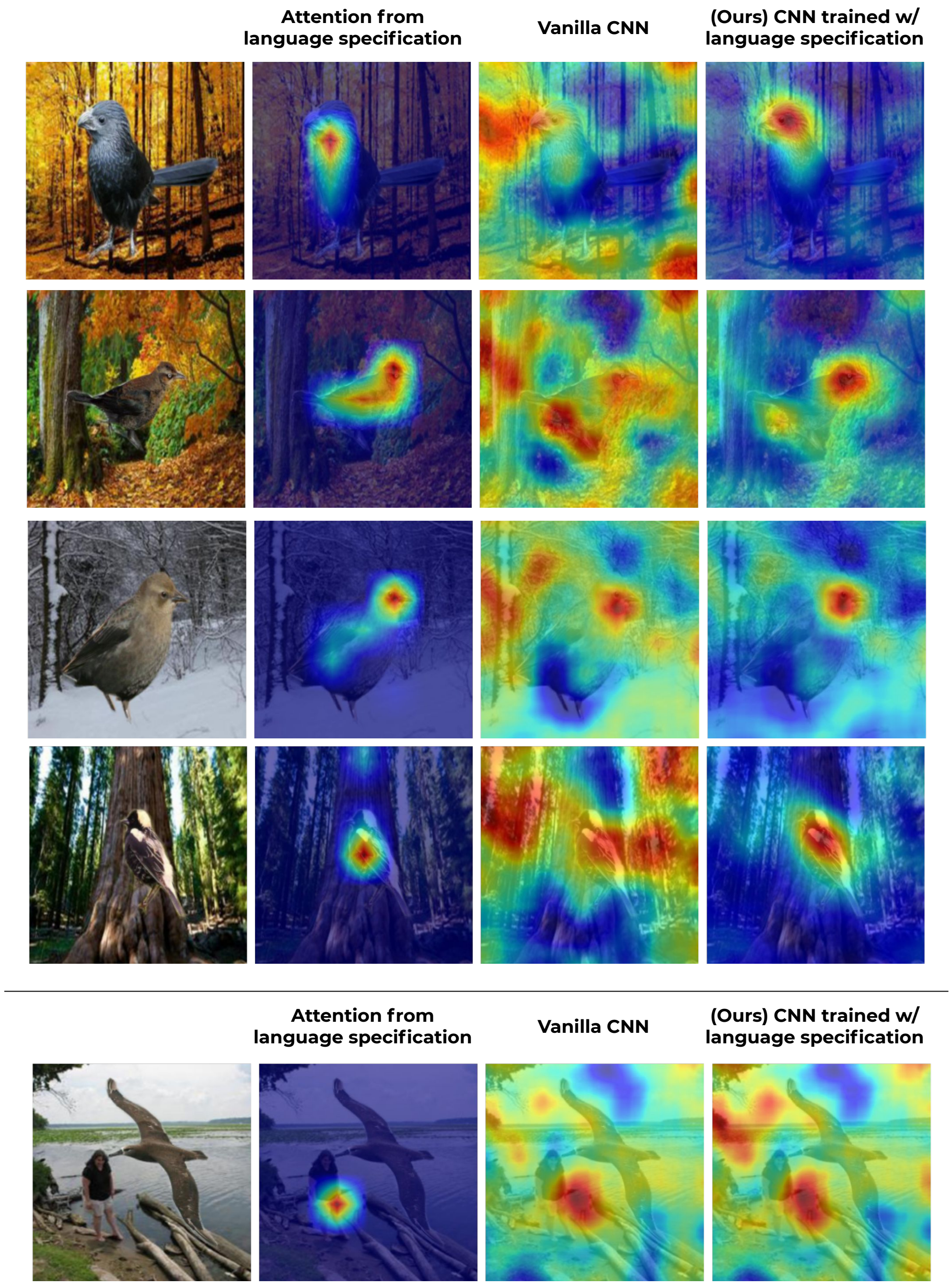


Figure A.1: Sample attention visualizations from the *Waterbirds-100%* training set. Our model places considerably less attention on the background features than did the Vanilla baseline. However, it can have difficulty localizing the bird in cases where the language-guided attention also has difficulty in grounding, as shown in the bottom row.

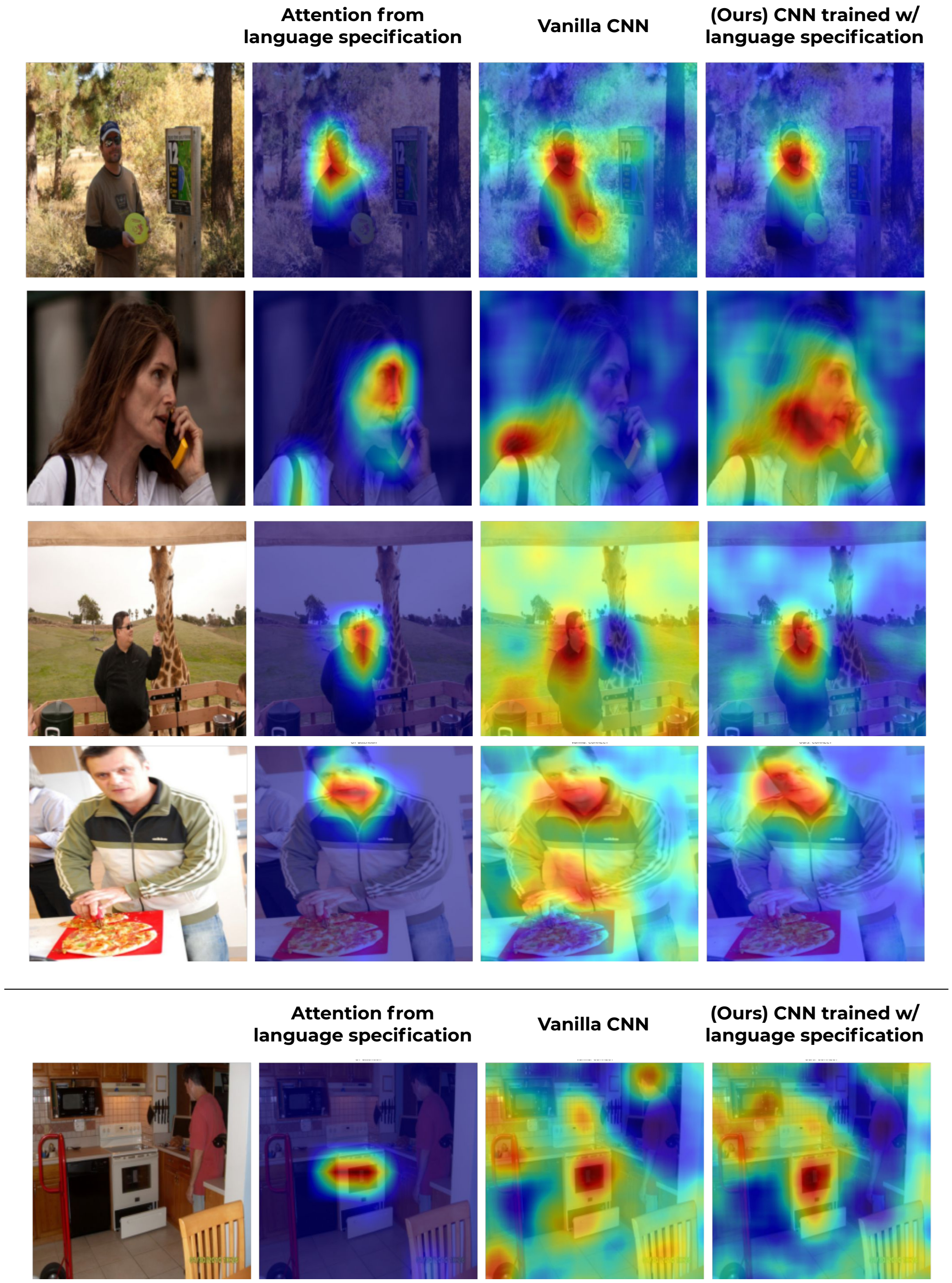


Figure A.2: Sample attention visualizations from the *MSCOCO-ApparentGender* training set. Like the attention from language specification, our model is proficient at identifying faces, and placing less attention on potentially biased context. However, the sample shown in the bottom row is an example where the language-guided attention does not localize the person correctly, and our model attends to similar features as the vanilla model.

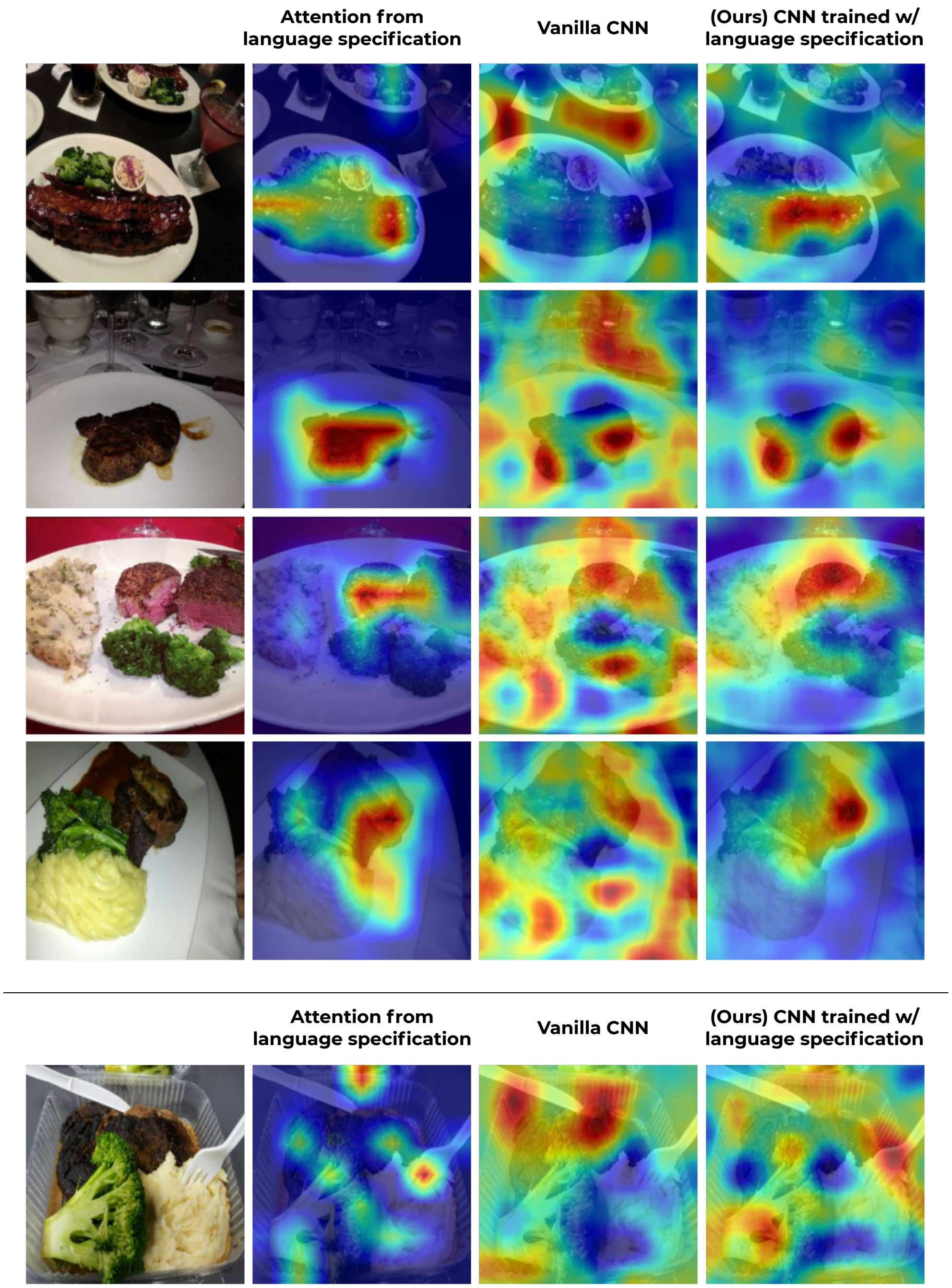


Figure A.3: Sample attention visualizations from the *Red Meat* training set. The images tend to show cluttered plates of food, where the meat is often not centered. *GALS* can learn to attend to the meat itself when guidance from the language specification is also able to localize the meat.

	Pointing Game Accuracy	
Method	***Waterbirds-95%***	***Waterbirds-100%***
Vanilla	<u>59.98</u>	<u>46.48</u>
ABN	51.73	25.96
UpWeight	59.42	26.34
GALS	**69.38**	**59.27**

Table A.9: Pointing game accuracy on *Waterbirds* datasets.

Appendix B

Chapter 3 Supplementary Material

Appendix B.1 has more discussion on Selector ablations.

Appendix B.2 shows an experiment with data augmentation for MaxProb.

Appendix B.3 provides a manual evaluation of the label noise.

Appendix B.4 gives further analysis comparing Selector versus MaxProb decisions.

Appendix B.5 provides more qualitative results.

Appendix B.6 presents results on threshold generalization.

Appendix B.7 looks at the calibration metric ECE.

Appendix B.8 has additional details on the dataset splits.

Appendix B.9 has additional model details.

Appendix B.10 provides standard deviations for results in Tab. 3.1 and Tab. 3.2.

Appendix B.11 provides a proof of Lemma 1, providing a motivation for the definition of the Effective Reliability score Φ_c.

Appendix B.12 discusses the relevance of related conformal prediction works.

B.1 Selector Design Ablations

Extending the discussion in Sec. 3.5, we are isolating the effects of different features/modalities on the risk-coverage trade-off when using Selector. In this direction, we experiment with different input representation variants from CLIP-ViL [160] in Tab. 3.3 by ablating the question q, multimodal r, and answer $f'(x)$ representations as well as different image representations. For image representations, we ablate the usage of the visual representation $\tilde{v}$ directly from the

Architecture	$\mathcal{C}$@1% ↑	AUC ↓	Φ_{100} ↑
1-layer Linear	10.38	11.32	**7.47**
2-layer MLP (ours)	12.92	10.43	7.31
4-layer Transformer	**13.25**	**10.41**	7.34

Table B.1: Different Selector architectures with CLIP-ViL on our selection function validation split (Val in Tab. B.6). All in %.

CLIP visual encoder [142], as well as the visual representation v that is the concatenation of the respective pooled outputs from MCAN's self-guided attention module [206] and MoVie's modulated convolutional bottleneck [129], which are visual representations that also contain multimodal information from the question. Question representations are taken from the output of MCAN's self-attention module. The multimodal representation is the concatentation of the multimodal representations that are used as inputs to the softmax output (i.e., classification) layer of CLIP-ViL. For the answer representation, we use the logits just before the softmax in the output layer.

The results in Tab. 3.3 show the importance of using multimodal information for coverage at low risk levels. When comparing using each representation in isolation, we see that multimodal representations (r, v, and $f'(x)$) yield much stronger $\mathcal{C}$@1%, $\mathcal{C}$@5%, Φ_{10} and Φ_{100} than unimodal representations ($\tilde{v}$ and q). We also observe that the answer representation achieves the best performance for $\mathcal{C}$@10% and $\mathcal{C}$@20% when each input representation is used in isolation. Overall, we find that considering multimodal information (i.e., combinations of multimodal representations and unimodal representations from different modalities) to be most effective, with the top performers being the models that incorporate the answer representation alongside multimodal representations ($f'(x)+r$, $f'(x)+v$, and $f'(x)+q+v+r$).

Lastly, we also experiment with other architectures for the Selector using the same features as above. Our Selector is a 2-layer multi-layered perceptron (MLP) (Appendix B.9). In Tab. B.1, we see that a simpler, 1-layer Selector has slightly higher Φ_{100}, yet lowers $\mathcal{C}$@1% by about 2.5%. A more complex Transformer yields comparable performance to our 2-layer Selector. Given these results as well as those in Tab. 3.3, we observe that the input representations and training objectives appear to be most important, and efforts for improving learned selection function performance can potentially focus on these.

B.2 Comparing to Data Augmentation

In our experiments, we use a separate set to validate VQA models and train the selection functions (Dev in Tab. B.6). However, one could use this data to augment the VQA training data, which could potentially improve performance for MaxProb as there is a relationship between accuracy and these reliability metrics (Sec. 3.5). Tab. B.2 presents these results where we see that using this data to train the Selector is more effective for improving coverage at low risk levels and Φ_c with a high cost. Since the extra data helps improve accuracy, as

Model f	Selection function g	Acc. ↑	$\mathcal{C}@\mathcal{R}$ ↑ $\mathcal{R}=1\%$	$\mathcal{R}=5\%$	$\mathcal{R}=10\%$	$\mathcal{R}=20\%$	AUC ↓	Φ_c ↑ c=1	c=10	c=100
	MaxProb	69.70	3.32	31.30	52.57	81.21	11.13	54.05	20.17	1.60
CLIP-ViL	MaxProb-Aug	70.52	6.57	33.01	54.72	**83.25**	10.73	**55.61**	22.05	2.76
	Selector	69.70	**12.92**	**36.29**	**55.64**	82.27	**10.43**	55.13	**24.66**	**7.31**

Table B.2: Comparison between augmenting the training data of CLIP-ViL with our Dev set for MaxProb versus utilizing it for training Selector. Results are on our selection function validation split (Val in Tab. B.6). All in %.

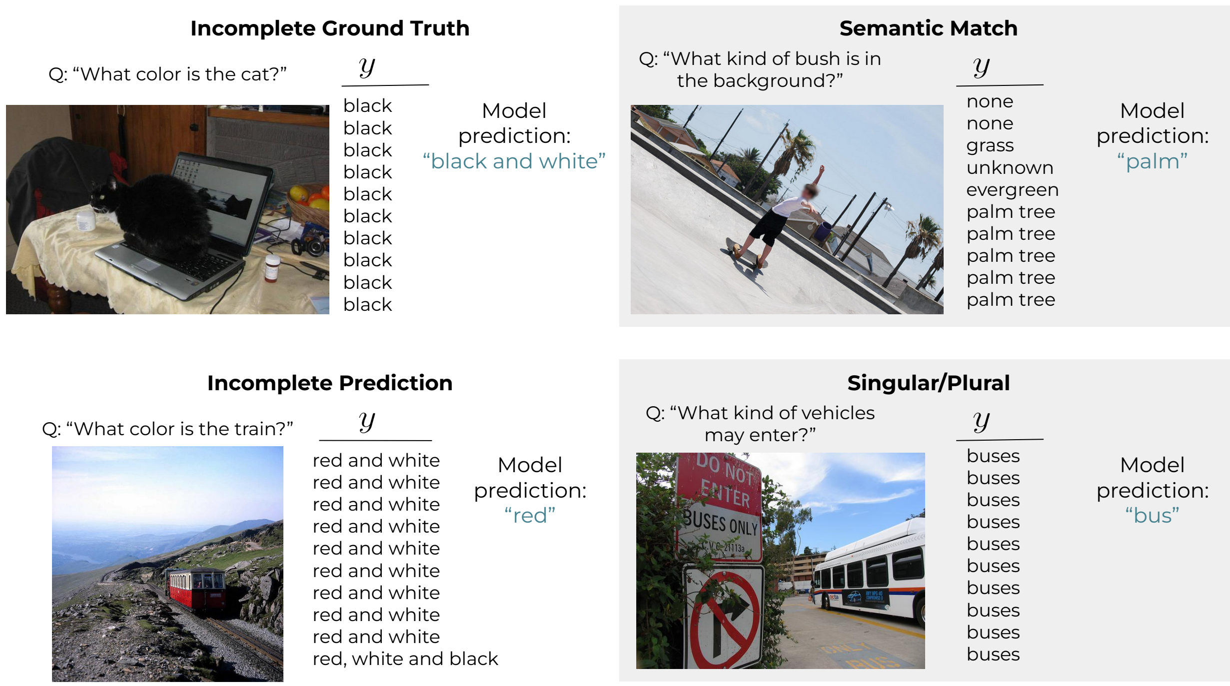


Figure B.1: Example questions, images, annotations, and model predictions for each category of label noise we discover.

the risk tolerance nears the error rate of the model and coverage approaches 100%, MaxProb surpasses Selector in coverage (i.e., $\mathcal{C}@20\%$) and Effective Reliability (i.e., Φ_1). However, overall, these results suggest that using this data to train a Selector can be more beneficial to model reliability than using it for augmentation.

B.3 Manual Evaluation of Label Noise

As discussed in Sec. 3.5, we provide further details on our manual annotation for label noise as well as Φ_{100} when accounting for cases where the model may have been unfairly penalized. We specifically annotate image-question-answer triples, and discovered the following cases

(Fig. B.1 provides examples of each):
Incomplete Ground Truth: The ground truth is in some way incomplete and simply misses the predicted answer.
Semantic Match: The predicted answer is semantically correct but does not exactly match the ground truth.
Incomplete Prediction: The predicted answer is incomplete but has part of the correct answer.
Singular/Plural: The predicted answer is singular/plural while the ground truth is plural/singular (though only if providing the opposite singular/plural version is still correct).

We do these annotations for each considered VQA model and selection function trained to optimize Φ_{100} (i.e., the strongest penalty for wrong answers) and focus our efforts on questions with VQA accuracy of 0, meaning questions that contribute negatively to Φ_{100}. Once we have the annotations of unfairly penalized questions, we recompute the Effective Reliability score Φ'_{100} when counting those questions as either abstentions or as answered questions that achieved a VQA accuracy of 100%. Although the selection function decided to answer each of the unfairly penalized questions that we annotated, we compute Φ'_{100} under these two cases because it is unclear exactly how correct these non-matching answers should be considered. Counting them as abstentions serves as a lower bound for Φ'_{100}, whereas assigning a VQA accuracy of 100% is an upper bound.

We present the results before (Φ_{100}) and after (Φ'_{100}) controlling for noise in Tab. B.3. We find that while this noise does contribute to some differences in performance, it does not affect the rankings between selection functions. For example, relative to each Φ_{100} with CLIP-ViL, and counting unfairly penalized questions as abstentions, Φ'_{100} yields an increase of 0.37% for MaxProb, 0.47% for Calibration, and 0.57% for Selector, yet the rankings remain the same. Qualitatively, we observe that there tends to be a very significant overlap in unfairly penalized examples between selection functions, which is likely part of why the rankings remain the same. Moreover, the amount of these label errors tends to be small, and the vast majority of questions contributing to the penalties in Φ_{100} across all models are properly marked as incorrect ($\sim$93%). Since the score for an incorrect sample (-100) is considerably lower than a sample marked as 100% correct (+1), there is also little difference in Φ'_{100} when considering these few unfairly penalized questions as abstentions versus as correct answers. These results imply that the comparisons between different selection functions at high cost (or low risk) for a given model are still meaningful despite the potential presence of noise.

B.4 Analysis of Selector Decisions

We would like to understand any differences in the types of questions that the Selector chooses to abstain or answer as compared to MaxProb. We compare decisions on our test split for the two selective models, where thresholds were chosen to optimize Φ_{100} on validation. We use labels from [172] which assign one of the following categories to each question, in order of difficulty: unimodal (Level 1), where the question could be answered without looking

Model f	Selection function g	% Correct GT	$\Phi_{100} \uparrow$	$\Phi'_{100} \uparrow$ *Abstain*	*Correct*
Pythia [73]	MaxProb	91.30	1.76	1.95	1.95
	Calibration	93.55	2.19	2.37	2.38
	Selector	87.50	**4.11**	**4.48**	**4.49**
ViLBERT [117]	MaxProb	97.75	1.67	1.86	1.86
	Calibration	94.94	2.56	2.93	2.94
	Selector	88.14	**5.38**	**6.32**	**6.33**
VisualBERT [98]	MaxProb	100.00	2.50	2.50	2.50
	Calibration	97.92	3.92	4.01	4.01
	Selector	85.29	**4.82**	**5.29**	**5.30**
CLIP-ViL [160]	MaxProb	94.74	1.32	1.69	1.70
	Calibration	93.44	5.32	5.79	5.80
	Selector	87.23	**8.74**	**9.31**	**9.31**

Table B.3: Effect of label noise on Φ_{100}. % Correct GT indicates the percentage of answered samples with a VQA accuracy of 0, where the ground truth and resulting VQA accuracy was considered correct based on the question, image, annotations, and model prediction. Φ_{100} indicates the original score, whereas Φ'_{100} indicates the score when counting answered questions where label errors led to a VQA accuracy of 0 as abstentions (*Abstain*) or having a VQA accuracy of 100% (*Correct*) instead of being counted as incorrect. Although there is a small amount of label noise, it does not affect the ranking between selection functions with respect to Effective Reliability. All in %.

at the image, "simple-multimodal" (Level 2), where the question is simple to answer when additionally considering the image, and "difficult-multimodal" (Level 3), where the question is difficult to answer even when considering both modalities. Fig. B.2 compares the number of questions answered in each difficulty level by the MaxProb and Selector models. We find that the Selector not only answers 1.1× more unimodal questions than MaxProb, but also 1.4× more "simple-multimodal" and, impressively, 2.4× more "difficult-multimodal" questions.

B.5 More Qualitative Analysis

In Fig. B.3, we show several more examples of cases from our test split that illustrate Selector and MaxProb decisions, where we use CLIP-ViL with selection functions optimized for Φ_{100} on the validation set (same as Fig. 3.3). In particular, we show cases where the decisions of Selector and MaxProb differed — where Selector chooses to answer while MaxProb abstains, and vice-versa. We see some cases where the MaxProb decision to abstain may have been influenced by variability in possible answers that may cause model confidence values to be

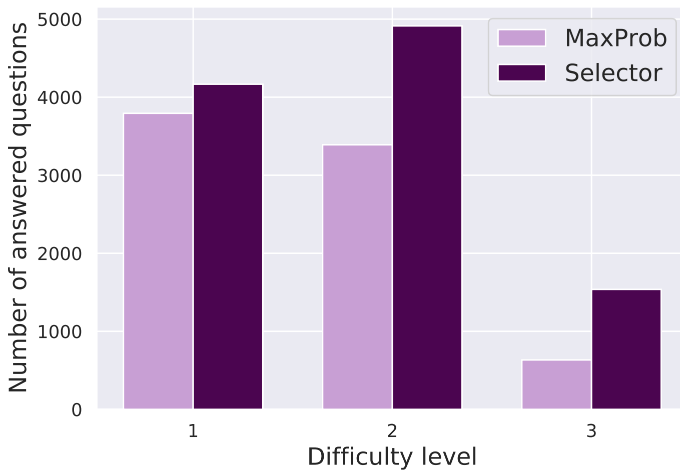


Figure B.2: Number of questions in our test split that the MaxProb and Selector selection functions chose to answer, grouped by difficulty level [172]. Level 1 corresponds to simple questions that could be answered without the image, Level 2 questions are simple to answer when considering both the question and image, and Level 3 questions are difficult to answer even when considering both modalities. Thresholds for the selection functions are chosen on the validation set to maximize Φ_{100}.

split, yet the annotations themselves have underlying semantic agreement (e.g., Fig. B.3 top left, where "*sunny*" weather conditions are also described as "*nice*" or "*clear*"). On the other hand, we also see cases where the model was incorrect on questions which may have been unclear or surprising, and Selector chose to abstain whereas MaxProb chose to answer (e.g., the second example on row (c) asks the unusual question "*Is the bear wearing a helmet?*"). In these cases, we would expect a selective VQA model to abstain from answering to avoid providing an incorrect answer. Additionally, we show several failure cases of Selector, which chose to answer on an incorrect question while MaxProb chose to abstain.

B.6 Threshold Generalization

As discussed in Sec. 3.5, we evaluate how well a threshold selected for a target risk level on validation can achieve a similar level of risk on our test split. Experimenting with VisualBERT, comparing MaxProb and Selector, we see in Tab. B.4 that the differences in risk for both selection functions tend to be at most 0.26%. Likewise, we observe corresponding differences in achieved coverage between the validation threshold and the maximum coverage ($\Delta\mathcal{C}$). This demonstrates that the thresholds can generalize reasonably well, although it does not allow for a direct comparison of coverage for the same risk. Effective Reliability, on the other hand, can use thresholds chosen from validation and still result in a clear comparison of models as

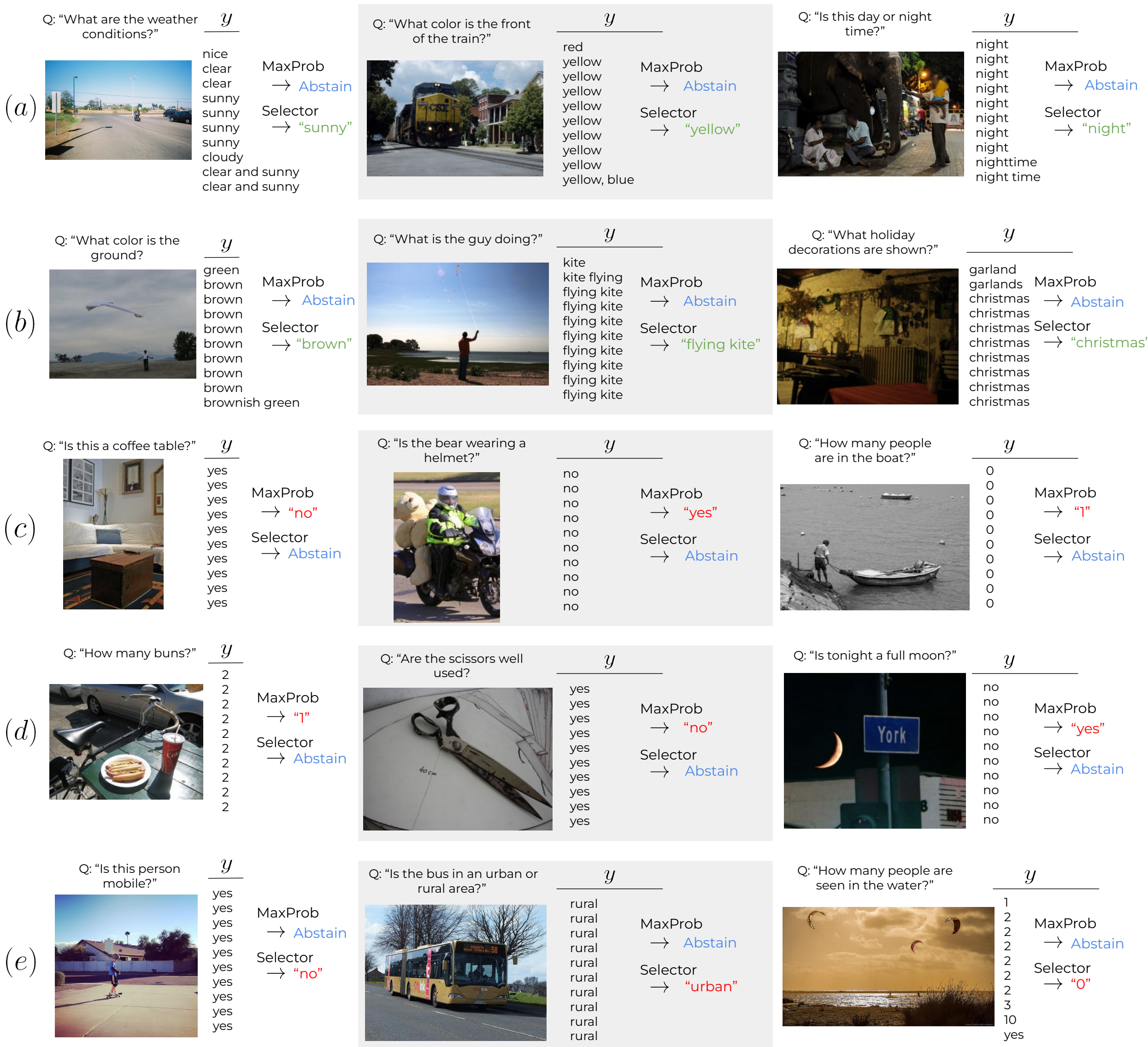


Figure B.3: More qualitative test set examples with CLIP-ViL selective model predictions, when optimized for Φ_{100} on validation. Rows (a) and (b) show cases where the model was correct, yet MaxProb chose to abstain and Selector chose to answer. Rows (c) and (d) show examples of the opposite case, where the model was wrong, yet MaxProb chose to answer (contributing to the risk) and Selector chose to abstain. Row (e) shows failure cases of Selector, which chose to answer on an incorrect sample when MaxProb chose to abstain.

it is a single metric.

Selection function g $\mathcal{R} =$	$\Delta\mathcal{R}$ 1%	5%	10%	20%	$\Delta\mathcal{C}$ 1%	5%	10%	20%
MaxProb	+0.11	−0.20	+0.26	−0.15	+0.91	−0.68	+0.82	−0.34
Selector	+0.22	+0.26	+0.22	−0.17	+2.79	+0.89	+0.74	−0.36

Table B.4: Generalization of abstention thresholds γ from validation to test, with VisualBERT. $\Delta\mathcal{R}$ and $\Delta\mathcal{C}$ are the differences in risk and coverage percentages, respectively, when using γ selected for the target risk $\mathcal{R}$ on validation vs. γ with maximum $\mathcal{C}@\mathcal{R}$.

	Pythia		ViLBERT		VisualBERT		CLIP-ViL	
	MaxProb	Calib.	MaxProb	Calib.	MaxProb	Calib.	MaxProb	Calib.
ECE ↓	0.1701	**0.0938**	0.1457	**0.1121**	0.1458	**0.1169**	0.1974	**0.1522**

Table B.5: ECE of different models with (Calibration, denoted Calib.) and without (MaxProb) the vector scaling calibration on our test split. Lower is better.

B.7 Effect of Model Calibration

We report the calibration performance of the vector scaling. Specifically, we measure the expected calibration error (ECE) [58, 127], which measures the expected difference between the model confidence and accuracy. The lower the ECE, the more that the model's confidence scores correspond to the actual accuracy of the predictions. Note that the ECE metric is designed for single label classification problems. To use the ECE metric for VQA, where there can be multiple possible answers for a question, we simply consider the most frequent human annotated answer as the ground truth for each question.

We see in Tab. B.5 that vector scaling does indeed improve calibration for all models. Taking this observation in combination with the improvements over MaxProb on $\mathcal{C}@\mathcal{R}$, AUC, and Effective Reliability seen in Tab. 3.1 and Tab. 3.2, it appears that improving model calibration can help improve the risk-coverage trade-off. However, as discussed in Sec. 3.4, it is necessary to use calibration techniques that can change the relative confidence rankings, such as vector scaling.

B.8 Additional Dataset Split Details

We experiment on the VQA v2 dataset [52], which contains a large amount of human-annotated image-question-answer triplets. Tab. B.6 lays out the data splits we use in our experiments. We create splits of the VQA v2 validation set since we require answer annotations to evaluate risk, coverage, and Effective Reliability. These splits are created such that no images (and therefore no question-answer annotations) are shared between them. Note that the data in

Source	Split Name	Usage	% src	#I	#Q	#A
VQA v2 train	Train	Train f	100%	82,783	443,757	4,437,570
VQA v2 val	Dev	Validate f / Train g	40%	16,202	86,138	861,380
	Val	Validate g	10%	4,050	21,878	218,780
	Test	Test h	50%	20,252	106,338	1,063,380

Table B.6: Table of statistics for the dataset splits used for training as well as validating VQA models (f), training as well as validating selection functions (g), and testing full selective models ($h = (f, g)$). % src indicates the percentage of the source data (Source) that each split represents. #I, #Q, and #A indicate the number of images, questions, and answers, respectively.

Hyperparameters	Pythia	ViLBERT†	VisualBERT†	CLIP-ViL
Batch Size	512	896	896	32
Hidden Size	5,000	1,024	768	1,024
# Layers	L-1, V-1	L-12, V-6	12	6 / 4
Optimizer	Adamax[86]	AdamW[115]	AdamW[115]	AdamW[115]
Adam ϵ	1e-8	1e-8	1e-8	1e-9
Adam β_1	0.9	0.9	0.9	0.9
Adam β_2	0.999	0.98	0.98	0.98
Learning rate	0.01	5e-5	5e-5	5e-5
Dropout	–	0.1	0.1	0.1
# Steps	22,000	88,000	88,000	236,000
# Warmup Steps	1,000	2,000	2,000	54,000
Max Grad. L2-Norm	0.25	–	–	5

Table B.7: Hyperparameters of each model used in our experiments. Max Grad. L2-Norm is used for gradient clipping. L and V indicate language and vision layers, respectively. The 6 / 4 for CLIP-ViL indicates that the model has 6 MCAN layers and 4 MoVie layers. † indicates that the hyperparameters are reported directly from [164].

the held out test set (Test in Tab. B.6) is never seen during the training or validation of any component (f or g) and is only used for evaluations. All presented results are on our test set unless otherwise specified.

B.9 Model Details

In this section, we present the details of the models used in our experiments.

VQA Models

We use the open-source MMF framework [165] for all our experiments, which contains implementations of each VQA model.[1] For training VQA models, we follow the hyperparameters from MMF, which we list in Tab. B.7. All models treat VQA as a classification task and are trained with VQA accuracy as soft target scores via a binary cross-entropy loss [171]. We briefly discuss the models and settings used in our experiments, extending Sec. 3.5:

Pythia [73]: A previous state-of-the-art model that won the 2018 VQA challenge and is an optimization of the widely used bottom-up top-down (BUTD) VQA model [5]. This model uses BUTD object detection features [5] trained on Visual Genome [87], but the features are extracted from a ResNext-152 based FasterRCNN [144]. Pythia's implementation further uses grid features from a ResNet-152 [61] as additional inputs to improve performance [73]. GloVe embeddings [134] are used to initialize the word representations. We train this model from scratch on the VQA v2 training data.

ViLBERT [117]: A two-stream vision-and-language transformer model [19, 169] that also uses object detection features. The same object detection features from Pythia are used, but without the addition of grid features. We use the pretrained and fine-tuned model provided by MMF.[2] The MMF version of this model is from [164] is pretrained on the VQA v2 training data [52] using self-supervised objectives (masked language modeling and masked image modeling). The VQA model is initialized with the pretrained encoder weights, and then fine-tuned on the VQA v2 training data.

VisualBERT [98]: This model is a single-stream transformer architecture, like BERT [38]. Here, the setup is very similar to ViLBERT and we use the same visual features as ViLBERT. We again use the pretrained and fine-tuned model provided by MMF.[2] This MMF version of VisualBERT [164] is pretrained on MSCOCO captions [22] using a masked language modeling objective. Just like ViLBERT, the VQA model is also initialized with the pretrained encoder weights and fine-tuned on VQA v2.

CLIP-ViL [160]: This represents a state-of-the-art model that is trained from scratch on the VQA data whose visual encoder is from the CLIP model [142]. The visual representations are grid features that are obtained from the visual encoder of the CLIP model [142]. We use the implementation provided by the authors of [160] to extract the visual features.[3] The VQA architecture, MoVie+MCAN [129], is an ensemble of a transformer encoder-decoder [206] and modulated convolutional [129] model, which won the 2020 VQA challenge. GloVe embeddings [134] are also used to initialize the word representations. Like Pythia, we train this VQA model from scratch on VQA v2 training data.

[1] `https://mmf.sh/`

[2] `https://github.com/facebookresearch/mmf/tree/main/projects/pretrain_vl_right`

[3] `https://github.com/clip-vil/CLIP-ViL/tree/master/CLIP-ViL-Direct/vqa`

Selection Functions

We detail the Calibration and Selector selection functions here. We do not cover MaxProb as no additional training is required. While training each selection function, we freeze the weights of the VQA model.

Calibration. The inputs to the calibration are the unnormalized answer logits (i.e., answer representation just before the softmax) of the VQA model, and the outputs are the calibrated logits. Since we use vector scaling [58, 137], we input the logits from the VQA model into a linear layer with a diagonal weight matrix and a bias term. During training, after the linear layer, we apply a sigmoid activation and, in contrast to [58], use these as input to a binary cross entropy loss with the soft VQA labels [171]. We train the linear layer using the AdamW optimizer [115] with a learning rate of 0.01 and a weight decay of 1e-4. At test time, we use the output of this linear layer as our calibrated logits, apply a softmax, and use the same abstention procedure as MaxProb (Sec. 3.4).

Selector. The inputs to Selector are the answer, question, image, and multimodal representations. For each input, we have a specific 1-layer MLP with a ReLU activation and hidden size of 512. We then concatenate the outputs of these layers and input them to a 2-layer MLP with ReLU activations and hidden size of 1,024, followed by a binary output layer to produce a confidence value. This architecture remains exactly the same for all models. However, if a model produces a set of representations for the image or question, then we max pool these features to collapse them to a single representation. For optimization, we employ the AdamW optimizer [115] with a learning rate of 1e-4, a batch size of 256, and gradient clipping with a max gradient L2 norm of 0.25.

B.10 Extended Results

Tab. B.8 and Tab. B.9 provide the mean and standard deviation over the 10 random seeds for Pythia and CLIP-ViL results. Due to difficulties reproducing the pretrained and fine-tuned performance of ViLBERT and VisualBERT, we simply use existing checkpoints in MMF[2] and report single run metrics for these VQA models.

B.11 Proof of Lemma 1

Lemma 1 states that if a model abstains "perfectly", the introduced Effective Reliability score is equal to the VQA Accuracy. In this section, we provide a proof of Lemma 1 in the main paper, which we repeat here for ease of understanding the proof:

Lemma 1. *The Effective Reliability score is equal to the VQA Accuracy ($\Phi_c(x) = Acc(x)$) if a model abstains ($g(x) = 0$)* iff *it is incorrect ($Acc(x) = 0$).*

Distilling this to the mathematical notation:

$$(g(x) = 0 \leftrightarrow Acc(x) = 0) \longrightarrow \Phi_c(x) = Acc(x) \tag{B.1}$$

Model f	Selection function g	Acc. ↑	$\mathcal{C}@\mathcal{R}$ ↑ $\mathcal{R}=1\%$	$\mathcal{R}=5\%$	$\mathcal{R}=10\%$	$\mathcal{R}=20\%$	AUC ↓
Pythia	MaxProb	64.63 ± 0.10	5.84 ± 0.36	24.03 ± 0.41	39.71 ± 0.34	68.63 ± 0.33	14.53 ± 0.08
	Calibration	64.90 ± 0.09	6.22 ± 0.47	24.37 ± 0.43	40.68 ± 0.29	71.29 ± 0.25	14.15 ± 0.08
	Selector	64.63 ± 0.10	8.30 ± 0.36	25.87 ± 0.35	41.71 ± 0.41	71.37 ± 0.22	13.94 ± 0.07
	Best Possible ($\mathcal{C}$)	64.63 ± 0.10	60.27 ± 0.11	66.04 ± 0.12	71.54 ± 0.13	80.78 ± 0.13	7.41 ± 0.05
CLIP-ViL	MaxProb	70.01 ± 0.13	6.83 ± 1.93	34.08 ± 1.17	54.00 ± 0.38	82.30 ± 0.20	10.81 ± 0.12
	Calibration	69.97 ± 0.11	12.43 ± 0.69	36.02 ± 0.29	54.03 ± 0.37	82.54 ± 0.20	10.55 ± 0.06
	Selector	70.01 ± 0.13	15.66 ± 0.71	37.92 ± 0.25	55.81 ± 0.41	82.74 ± 0.24	10.18 ± 0.07
	Best Possible ($\mathcal{C}$)	70.01 ± 0.13	65.71 ± 0.14	71.86 ± 0.15	77.79 ± 0.14	87.51 ± 0.16	5.27 ± 0.05

Table B.8: Mean and standard deviations for risk-coverage metrics for different selection functions from Tab. 3.1. All in %. See Sec. B.10.

Model f	Selection function g	c=1 Φ_1 ↑	$\mathcal{R}$ ↓	$\mathcal{C}$ ↑	c=10 Φ_{10} ↑	$\mathcal{R}$ ↓	$\mathcal{C}$ ↑	c=100 Φ_{100} ↑	$\mathcal{R}$ ↓	$\mathcal{C}$ ↑
Pythia	|	36.97 ± 0.19	35.37 ± 0.10	100.00 ± 0.00	-211.96 ± 1.00	35.37 ± 0.10	100.00 ± 0.00	-2701.25 ± 9.16	35.37 ± 0.10	100.00 ± 0.00
	MaxProb	46.49 ± 0.13	22.48 ± 0.18	75.58 ± 0.44	15.05 ± 0.34	5.68 ± 0.61	26.41 ± 1.88	1.90 ± 0.55	0.94 ± 0.31	5.13 ± 1.79
	Calibration	47.29 ± 0.15	21.66 ± 0.45	74.92 ± 0.90	15.18 ± 0.39	5.97 ± 0.77	27.73 ± 2.48	2.35 ± 0.63	0.92 ± 0.25	5.59 ± 1.28
	Selector	47.47 ± 0.14	21.02 ± 0.55	73.52 ± 1.12	17.03 ± 0.24	6.34 ± 0.25	30.16 ± 0.75	3.84 ± 0.39	1.01 ± 0.20	8.23 ± 1.33
	Best Possible (Φ_c)	64.63 ± 0.10	10.66 ± 0.06	72.34 ± 0.09	64.63 ± 0.10	10.66 ± 0.06	72.34 ± 0.09	64.63 ± 0.10	10.66 ± 0.06	72.34 ± 0.09
CLIP-ViL	|	47.68 ± 0.24	29.99 ± 0.13	100.00 ± 0.00	-153.27 ± 1.32	29.99 ± 0.13	100.00 ± 0.00	-2162.82 ± 12.26	29.99 ± 0.13	100.00 ± 0.00
	MaxProb	54.77 ± 0.15	19.84 ± 0.38	81.98 ± 0.81	21.93 ± 0.50	5.93 ± 0.24	38.47 ± 1.01	2.82 ± 0.78	0.98 ± 0.24	7.27 ± 2.00
	Calibration	55.00 ± 0.16	18.91 ± 0.50	80.24 ± 1.09	23.16 ± 0.33	5.20 ± 0.47	36.73 ± 1.79	5.29 ± 0.71	0.78 ± 0.20	9.96 ± 2.35
	Selector	55.47 ± 0.17	18.18 ± 0.54	79.09 ± 1.07	25.93 ± 0.28	5.41 ± 0.48	39.55 ± 1.96	8.00 ± 0.68	0.60 ± 0.17	11.37 ± 2.11
	Best Possible (Φ_c)	70.01 ± 0.13	9.86 ± 0.08	77.67 ± 0.12	70.01 ± 0.13	9.86 ± 0.08	77.67 ± 0.12	70.01 ± 0.13	9.86 ± 0.08	77.67 ± 0.12

Table B.9: Mean and standard deviation for Effective Reliability Φ_c over 10 trials from Tab. 3.2. All in %. See Sec. B.10.

Extending Eq. 6 to both cases, $Acc(x) = 0$ and $Acc(x) > 0$ (note, that Acc cannot be smaller than 0):

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ -c & \text{if } g(x) = 1 \text{ and } Acc(x) = 0, \\ 0 & \text{if } g(x) = 0 \text{ and } Acc(x) > 0, \\ 0 & \text{if } g(x) = 0 \text{ and } Acc(x) = 0. \end{cases} \tag{B.2}$$

To prove Lemma 1, we must show that the condition $(g(x) = 0 \leftrightarrow Acc(x) = 0)$ implies $\Phi_c(x) = Acc(x)$. The condition $(g(x) = 0 \leftrightarrow Acc(x) = 0)$ simplifies Eq. B.2 as the second and third line contradict the condition:

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ 0 & \text{if } g(x) = 0 \text{ and } Acc(x) = 0. \end{cases} \tag{B.3}$$

As the $Acc(x) = 0$, the second line can be re-written as:

$$\Phi_c(x) = \begin{cases} Acc(x) & \text{if } g(x) = 1 \text{ and } Acc(x) > 0, \\ Acc(x) & \text{if } g(x) = 0 \text{ and } Acc(x) = 0. \end{cases} \tag{B.4}$$

Now, in both cases $\Phi_c(x) = Acc(x)$ $\square$

B.12 Relation to Conformal Prediction

Conformal prediction aims to predict a set of outputs, with a guarantee that the set contains the correct output with a specified probability [180, 156]. In VQA, the criterion of a set containing the "correct output" is harder to define. For example, two distinct answers might be both be true (*"yellow","brown"*) for *"What color are the bananas?"*, but others sets might be contradictory (*"yes","no"*). Further research might focus on how to best convey answer sets to users in VQA and how semantic similarity of answers should be modeled, or on the design of better criteria to determine a set-based risk. More generally, the field of risk control, which does not require variable-size output sets, provides theoretical guarantees that a given error measure is below a tolerance level with some specified probability [8, 70]. [9] describes how to choose a prediction threshold to satisfy a guarantee on error bound. [70] relates these guarantees to test sample accuracy based on training sample density. We view these probabilistic guarantees on error bounds as complementary to our framework, with opportunities for future work to incorporate them both.

Appendix C

Chapter 4 Supplementary Material

C.1 Overview

Appendix C.2 presents an ablation showing several alternative algebraic confidence estimates, and compares the precision-recall curve for the learned TLC-L to that of algebraic confidences when separating correct and hallucinated objects. Appendix C.3 presents additional qualitative examples of both success and failure cases, comparing TLC-L to the Baseline model. Appendix C.4 and Appendix C.5 provide further details on datasets and models respectively.

C.2 Alternative Confidence Estimates

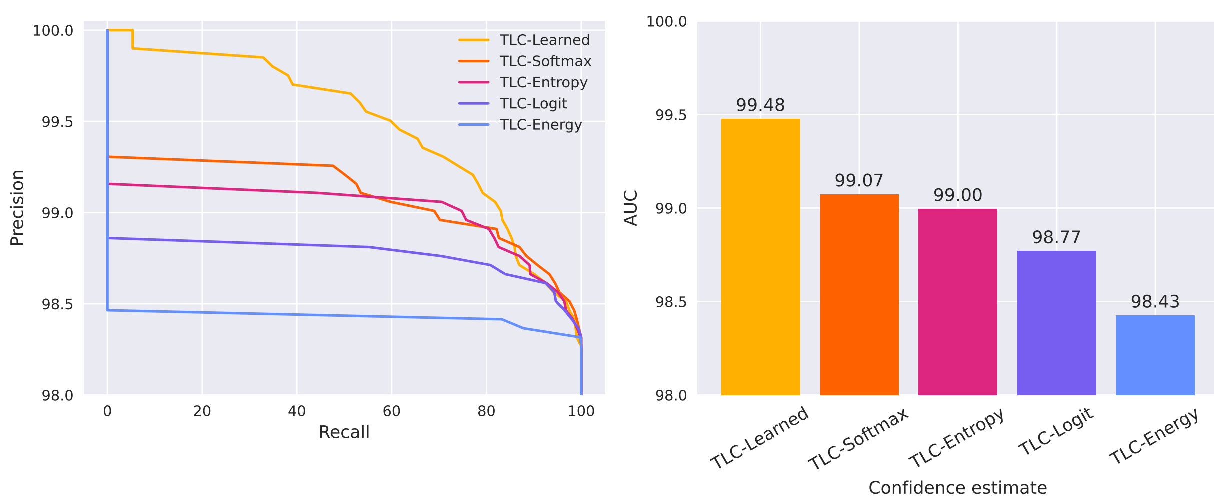


Figure C.1: Precision-recall curve (left) and AUC (right) with different confidence estimates for separating correct and hallucinated objects. Results are shown on our validation set using $\text{OFA}_{\text{Large}}$.

We compare several other choices of algebraic confidence estimates for TLC-A besides softmax score used in the main paper. All are derived from the likelihood (logit) distribution $\vec{z}_k$, as mentioned in Sec. 4.3. **Logit** is the logit value for the selected token directly from $\vec{z}_k$, whereas **Softmax** is the corresponding value after a softmax function. Again, in our main paper, TLC-A is based on this softmax score confidence. **Entropy** is the negative entropy of the log-softmax distribution, as a higher entropy should indicate higher uncertainty. Entropy has been previously used as a direct estimate of model uncertainty [182] as well as a penalty in image caption decoding [195]. Finally, we consider the **Energy** score [113], originally proposed as a measure for OOD detection that theoretically correlates with the probability density of the in-domain samples. We use a temperature of 1, and negate the energy score so positive values indicate confident samples.

In Fig. C.1, we show the precision-recall curve for various confidence estimates to separate correct and hallucinated objects. We compute these results on our MSC-Main validation set for g (see Tab. C.1). Specifically, we are not interested in the exact values of confidence estimates themselves, but rather how well they can *rank* correct objects over those that are hallucinated. When using confidence estimates in practice, we need a threshold to make a binary decision about whether an object in a caption is considered hallucinated or not (Sec. 4.3). We choose this threshold for a specific precision level, above the accuracy that the model achieves on its own. For instance, on the validation set for g, about 98.3% of the captioning model's predicted objects are correct (and the rest hallucinated). To push reliability further, we choose a threshold γ for each method that achieves a precision of 99%. In Fig. C.1 (left), we therefore only show recall rates above 98% precision, yet show the overall area-under-the-curve (AUC) in Fig. C.1 (right).

From Fig. C.1, we can see that TLC-Learned (*i.e.*, TLC-L) achieves the highest AUC of 99.48%, and TLC-Softmax achieves the second-highest of 99.07%. The precision-recall plot shows that all algebraic confidences reach 0% recall before 99.5% precision, whereas TLC-L still retains about 60% recall at this high precision rate. In our main paper, we use TLC-A to denote TLC-Softmax, as it performed the best among the algebraic confidences.

C.3 Additional Qualitative Examples

In Fig. C.2, we present qualitative examples (in addition to those in Fig. 4.3) where the Baseline model caption contained a hallucination, yet the caption selected by TLC-L did not. Note that "Baseline" refers to "Standard" as in Tab. 4.4. In Fig. C.3, we show several failure cases of TLC-L. On the left is a case where the Baseline model selects a more general caption, whereas TLC-L erroneously rejects it for one with a hallucinated "carrot". On the middle and right, TLC-L selects captions that include other hallucinations of objects. Nevertheless, TLC-L corrected 44.5% (252/566) of captions that contained a hallucination from the Baseline model, whereas TLC-L introduced a hallucination in only 0.2% ($38/19,686$) of captions that did not contain a hallucination from the Baseline model.

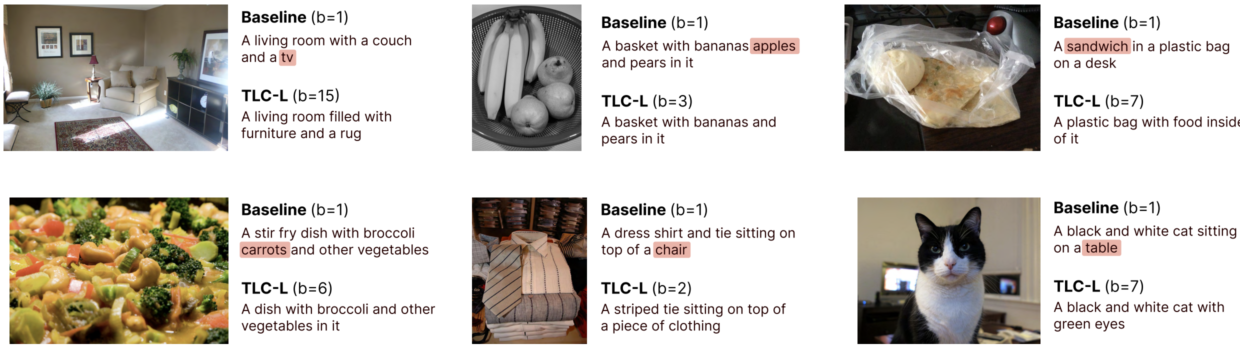


Figure C.2: Additional qualitative examples on our test set for TLC-L on $\text{OFA}_{\text{Large}}$, where the Baseline model caption contained a hallucination, yet the caption selected by TLC-L did not.

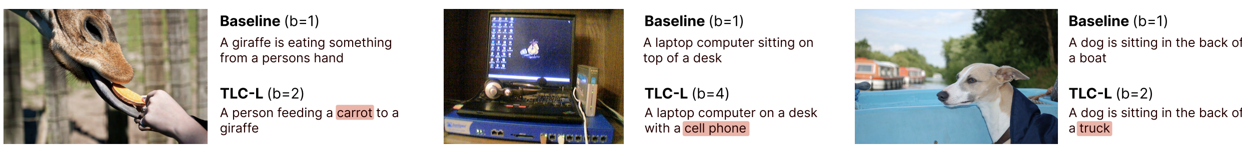


Figure C.3: Failure cases on our test set for TLC-L on $\text{OFA}_{\text{Large}}$, where TLC-L selected a caption with a hallucination, yet the Baseline did not.

C.4 Dataset details

MS COCO Captions. We use the same dataset splits as [192] for training and validating the captioning model f_{cap} and confidence estimator g, as [192] similarly reserves validation data in MS COCO for training a confidence estimator (yet for the visual question answering task, rather than image captioning). For the Standard-Aug model f'_{cap} in Tab. 4.4, we include the training set for g as part of the training set for f'_{cap}. In Tab. C.1, we refer to these splits as MSC-Main (for MS COCO Main), and use them for results in Tabs. 4.4, 4.5, and 4.6, and Figs. 4.3, C.1, C.2, and C.3. For comparison to prior work that uses the Karpathy test split (Tab. 4.7), we re-split the validation set to prevent overlap. These details are presented as MSC-Prior in Tab. C.1.
Winoground. We use the original data and evaluation setup for Winoground as in the original paper [173], which consisted of 800 unique images and captions. This leads to 400 examples, each consisting of two image-caption pairs, where the captions contain the same words and/or morphemes yet a different word order.
SVO-Probes. For SVO-Probes [62], we use the authors' public code to access a subset of data where the images were available. As discussed in Sec. 4.4, each image is annotated with a ⟨subject, verb, object⟩ relation, *e.g.*, ⟨girl, sit, shore⟩ relation. We take the available data that contrasts two verbs, *e.g.*, a "positive" or image-consistent relation ⟨girl, sit, shore⟩ and a "negative" or inconsistent relation ⟨girl, walk, shore⟩. For each image, we take the provided "positive" caption (*e.g.*, "A girl sits on the shore"), and use a part-of-speech tagger [65] to

Dataset	Use Case	# Images	# Captions
MSC - Main	Train f_{cap} and f'_{cap}	82,783	414,113
	Validate f_{cap}, Train g and f'_{cap}	16,202	81,065
	Validate g and f'_{cap}, Select g thresholds	4,050	20,268
	Evaluation	20,252	101,321
MSC - Prior	Train f_{cap}	82,783	414,113
	Validate f_{cap}, Train g	28,403	142,120
	Validate g, Select g thresholds	7,101	35,524
	Evaluation	5,000	25,010
Winoground	Evaluation	800	800
SVO-Probes	Evaluation	12,958	6,479

Table C.1: Overview of datasets used in our work. MSC indicates MS COCO Captions [22].

localize the verb (*e.g.*, "sit") in the sentence. We do not use images where the tagger failed to identify the verb, often in cases where the verb did not appear in the caption itself (*e.g.*, a triplet of ⟨person, wear, glasses⟩ with a caption of "The glasses fogged up"). The final split contains about 6,500 image-caption pairs (Tab. C.1), half of which are correct pairs. This evaluation is not directly comparable to prior work [62], which used the full set of data, chose a threshold of 0.5 to indicate whether or not an individual sample matched an image, and was performed at a sequence-level rather than word-level. In our work, we contrast a positive and negative image for a given caption, and label a sample as correct if the confidence for the positive pair is larger than the confidence for the negative pair, similar to Winoground.

Overlap with training data. All OFA models were not exposed to any MS COCO validation or test data during pretraining [184]. Winoground was hand-curated from the Getty Images API [50, 173], which is not used by OFA pretraining. Data from SVO-Probes was collected via the Google Image Search API and de-duplicated against Conceptual Captions [62, 159]. As OFA models used Conceptual Captions during pretraining, we assume there is no further overlap.

C.5 Model details

Captioning. To complement the details in Sec. 4.4, we provide additional experimental details for the captioning models. We use publicly available checkpoints for pretrained models provided by [184]. Parameter counts are 930M for $\text{OFA}_{\text{Large}}$, 180M for OFA_{Base}, and 33M for OFA_{Tiny} [184]. To finetune the pretrained models on MS COCO Captions, we follow the same settings from [184], where we train with cross entropy loss for 2 epochs for $\text{OFA}_{\text{Large}}$, and 5 epochs for OFA_{Base} and OFA_{Tiny}. We then train with CIDEr optimization for 3 epochs.

TLC-L. In addition to details in Sec. 4.4, we provide further information on the learned confidence estimator g. We use a 4-layer Transformer encoder [177] with 4 attention heads each. The embedded output corresponding to the token of interest t_k (Sec. 4.3) is passed to a 2-layer MLP, with hidden dimensions of size 512. The embedding dimension is 1024 for $\mathrm{OFA}_{\mathrm{Large}}$, 768 for $\mathrm{OFA}_{\mathrm{Base}}$, and 512 for $\mathrm{OFA}_{\mathrm{Tiny}}$. We train g for 200 epochs, with a batch size of 256, starting learning rate of 0.001, warm up ratio of 0.06 and polynomial learning rate decay to $2e$-7. We use the Adam optimizer [86] and clip gradients over 1.0. For aggregating tokens over objects for caption generation (Sec. 4.3), we use the minimum score for softmax and average for TLC-L, found on our validation set.

Appendix D

Chapter 5 Supplementary Material

D.1 Table of Contents

D.2 Analysis on Word Position

In Fig. D.1, we present an analysis of the impact of word position within captions on two key measures: Φ and the likelihood of various labeling outcomes. This analysis is conducted using the LLaVA-13B model on the ADE20K dataset.

Φ versus word position. In Fig. D.1 (Top), Φ is plotted against the word position, revealing that as the length of the caption increases, there is a higher agreement with the language prior, indicated by a lower Φ. This suggests that longer captions tend to align more closely with the expected language patterns.

Likelihood of labels. In Fig. D.1 (Bottom), we analyze the likelihood of different labels (Correct, Analysis, and Hallucination) at each word position within the captions. We observe general trend where the likelihood of Analysis and Hallucination labels increases as the caption progresses. This indicates that initial words in the caption are more likely to be labeled as Correct, while subsequent words have a higher tendency to be labeled as either Analysis or Hallucination.

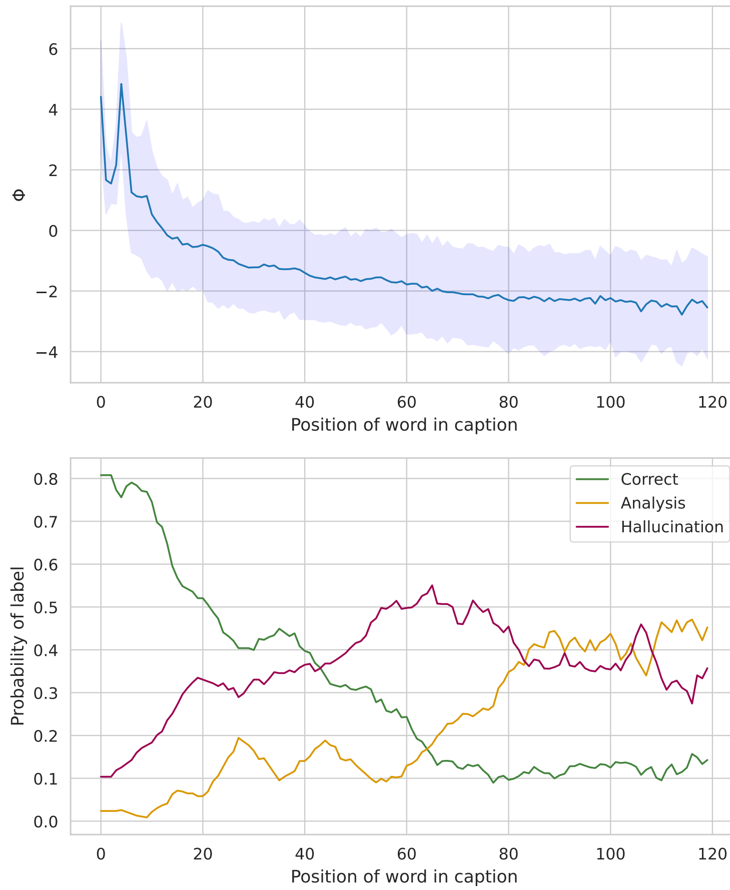


Figure D.1: **(Top)** We plot Φ versus the position of a word in a caption, computed for LLaVA-13B on ADE20K. As caption length increases, the caption outputs have higher agreement (lower Φ) with the language prior. **(Bottom)** We plot the likelihood of CORRECT, ANALYSIS, and HALLUCINATION labels at each word position. We observe the general trend that ANALYSIS and HALLUCINATION become more likely as the caption progresses.

D.3 Choosing a Distance Function for Φ

In our work, we measured how much a generated output agrees with the language prior at each token position. To do so, we needed a distance function between the original output and language prior. We show the performance of different distance functions in Fig D.2, and ultimately choose difference in logit score for the rest of our work. Here, we describe these options in more detail, and discuss why difference in logits may have performed best.

In Sec. 5.4, we defined Φ as a measure of difference $d(t_i, t_i^L)$ between the original (image-conditioned) output distribution t_i and language prior (non-image-conditioned) output distribution t_i^L, at position i in the generated output. Let t_i and t_i^L denote the logit distributions at position i, k_i be the token index from the vocabulary that was selected in the caption at position i, $|V|$ be the vocabulary size, and $\sigma(q)[k_i]$ be the softmax score for a given logit distribution q and token index k_i. We define m_i as the pointwise mean between t_i and t_i^L, used in Jensen-Shannon Divergence. For logit distributions X, Y, we define the entropy $H(X)$ and relative entropy $H(X,Y)$ (i.e., KL-Divergence) as:

$$\begin{aligned} H(X) &= -\sum_{j=1}^{|V|} \sigma(X) \cdot \log_2(\sigma(X)) \\ H(X,Y) &= \sum_{j=1}^{|V|} \sigma(X) \cdot \log_2(\sigma(X)/\sigma(Y)) \end{aligned}$$

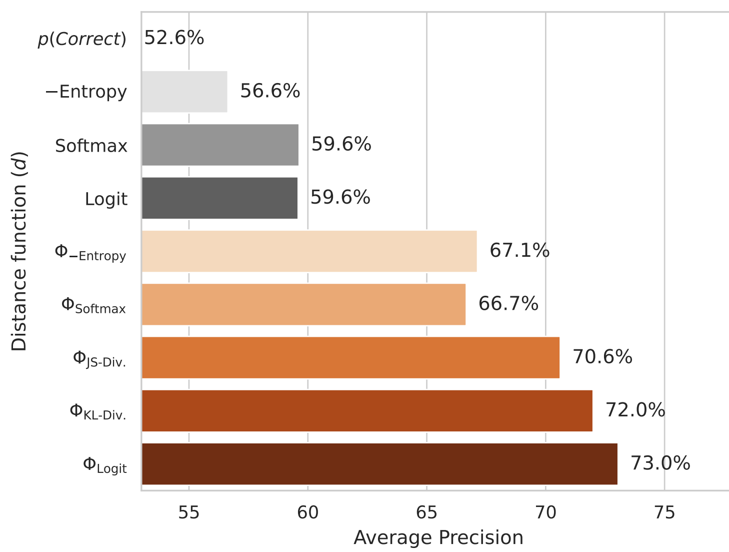


Figure D.2: Average precision for CORRECT words with different distance functions d on MHal-Detect. p(CORRECT) is 52.6% AP. We use Φ throughout the paper to refer to Φ_{Logit}, the difference in logit score.

We additionally explore baseline confidence measures that depend only on t_i, and not the language prior t_i^L – these are Logit, Softmax, and −Entropy (we negate the entropy, as low entropy corresponds to high confidence). We define these, as well as Φ using several different choices of d, as follows:

$$
\begin{aligned}
\text{Logit} &= t_i[k] \\
\text{Softmax} &= \sigma(t_i)[k] \\
-\text{Entropy} &= -H(t_i) \\
\Phi_{\text{Logit}} &= t_i[k_i] - t_i^L[k_i] \\
\Phi_{\text{Softmax}} &= \sigma(t_i)[k_i] - \sigma(t_i^L)[k_i] \\
\Phi_{-\text{Entropy}} &= H(t_i^L) - H(t_i) \\
\Phi_{\text{KL-Div.}} &= H(t_i^L, t_i) \\
\Phi_{\text{JS-Div.}} &= \sqrt{[H(t_i, m_i) + H(t_i^L, m_i)]/2}
\end{aligned}
$$

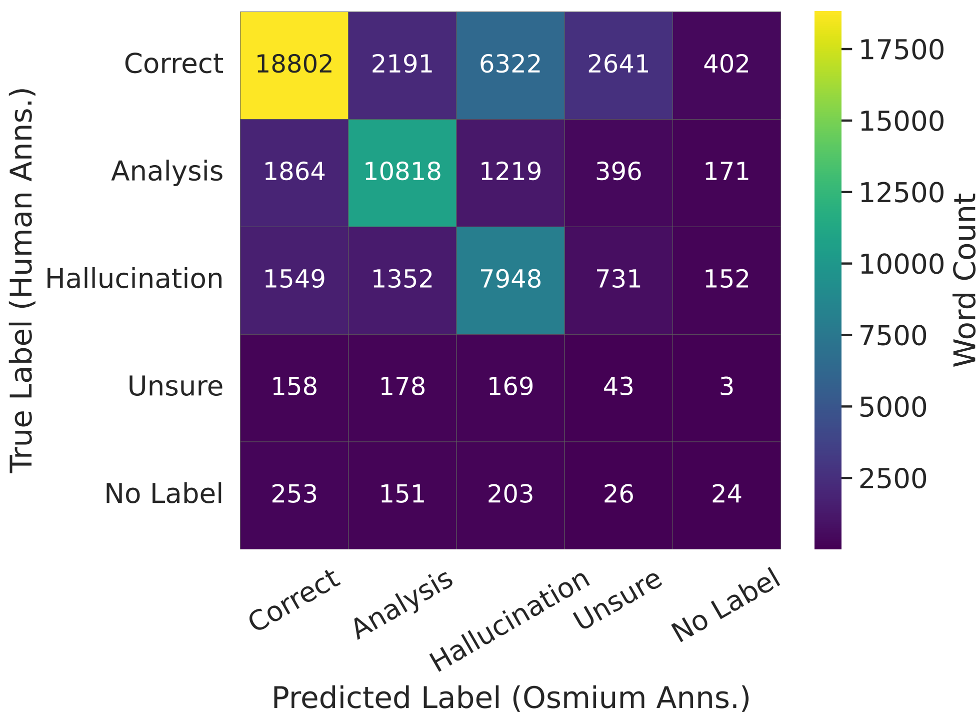


Figure D.3: Confusion matrix between annotation categories, comparing human annotations from MHal-Detect (left) to our predicted annotations from Osmium (bottom). Our labels are conservative in labeling words as CORRECT, with the main source of error coming from false negatives.

Φ_{Logit} is the score for Φ that we use throughout the main paper, simply measuring the difference in logit values at the selected token index k_i. Φ_{Softmax} is the corresponding difference in softmax scores. $\Phi_{-\text{Entropy}}$ is the difference in negative entropy between the two distributions. $\Phi_{\text{KL-Div.}}$ is the Kullback-Leibler Divergence, measuring how much the original distribution

$\sigma(t_i)$ approximates the language prior $\sigma(t_i^L)$. $\Phi_{\text{JS-Div.}}$ measures the symmetric Jensen-Shannon Divergence.

In Fig. D.2, we plot $\text{AP}(\Phi_d)$ using the various distance functions d on MHal-Detect. We use the provided human annotations as ground-truth, where $p(\text{CORRECT}) = 52.6\%$. We see that any choice of Φ outperforms all of the baseline confidence measures. Φ_{Logit} reached the highest AP, and so we select it as the choice of d in the main paper.

Why might Φ_{Logit} have performed best? Note that Φ_{Logit} is the only choice of d where the distribution t_i is not normalized with softmax. This may be an important reason for why the difference in logit score, somewhat surprisingly, performs best. The language prior probability distribution $\sigma(t_i^L)$ often has higher variance than that of the image-conditioned probability distribution $\sigma(t_i)$. For instance, there are many continuations of "The image contains a" under language – the next token could be "person", "tree", or many others. However, t_i has strictly more constraints (the image); there are often fewer possible continuations and lower variance in t_i. Thus, even if the logits $t_i[k_i]$ and $t_i^L[k_i]$ themselves were similar, taking $\sigma(t_i)[k_i]$ and $\sigma(t_i^L)[k_i]$ can differentiate them dramatically due to the different spread in distribution. Therefore, the logit scores before softmax normalization may be a better measure of similarity at a specific choice of token index k_i.

Influence of parts of speech. In Fig. D.4, we subdivide results from Fig. D.2 by the parts of speech (PoS) of each word. We aggregated "grounded" parts of speech: nouns, verbs, adjectives, and numbers (listed these in order of sample size in Fig. D.4). Next, we divided results further into each of those PoS. Note that each chart has a different chance level p(Correct), set as the left-hand side of the x-axis. All Φ-based distance measures reached higher AP for identifying CORRECT words over all baseline confidence scores in each evaluation, with the exception of the number PoS, where only Φ_{Softmax} and Φ_{Logit} outperformed each baseline. This PoS had the smallest sample size of only 517 words. When aggregating the grounded PoS categories (second chart), Φ with JS-Div, KL-Div, and Logit performed similarly (from 75.5-76.6%). For verbs, $\Phi_{\text{JS-Div.}}$ and $\Phi_{\text{KL-Div.}}$ outperformed Φ_{Logit}; yet, for nouns, Φ_{Logit} performed best. Given these results, we recommend using logit score as the distance function in Eq. 5.2.

D.4 Sources of Misalignment Between Osmium and Human Annotations

We further investigate the reasons for any misalignment between Osmium and human annotations that differ. First, we notice that some Osmium errors seem to arise from incomplete reference facts from GPT-4V in Stage 1. For instance, Osmium may label a statement that correctly describes an object as HALLUCINATION when the object is not mentioned by GPT-4V. To account for this, we select a 56-image subset of MHal-Detect and manually edit Stage 1 reference facts to be consistent with the human annotations, adding or changing statements in the original GPT-4V output. Next, we re-label the captions in Stage

2 using the edited Stage 1 reference facts. We re-compute the precision and recall scores using the human annotations as ground-truth. Fig. D.6 shows the difference in scores when using the original versus edited reference facts. All metrics improve, especially the precision of HALLUCINATION, which increases by 15.83%.

However, Osmium labels do not perfectly match human annotations, even with manually edited Stage 1 reference facts. There are 3 common reasons for mismatches in annotations, with examples shown in Fig. D.5: **Human Label Error**, **Subjectivity of Granularity**, and **Subjectivity of Analysis**. These lead to additional drops in precision and recall in cases where Osmium nevertheless reflects a correct interpretation of images and captions.

Human Label Error. The human annotations themselves may contain errors, as illustrated in Fig. D.5 (top). Because human annotations in MHal-Detect are used as ground-truth, Osmium may reflect a correct interpretation of a scene and caption, yet have low precision and recall.

Subjectivity of Granularity. The labeling task is inherently subjective, with multiple possible interpretations that are equally correct. One area of subjectivity is the granularity at which hallucinations are labeled. Both Osmium and human annotations are free to select any span of text within a caption when assigning a label; e.g., neither is limited to sentence-level labeling. However, the localization of an error is subjective. For instance, in Fig. D.5 (middle), there are four birds in the scene. The entire phrase "There are six birds in total" is labeled as HALLUCINATION by Osmium, whereas the human annotations label only the word "six" as HALLUCINATION. Both choices of labels represent correct interpretations of the scene. As another example, consider an image of two children, where one is holding a basketball and another is holding a tennis racket, and the phrase "the children are holding basketballs". The entire phrase may be annotated as HALLUCINATION, or only "children" and "basketballs" as the plural forms are incorrect, or only "basketballs" as it could have instead been "a basketball and a tennis racket". Because natural language is long-tailed, it is difficult to define specific rules around the granularity of labeling. While we consider multiple interpretations to be equally correct, we can only use human annotations as absolute ground-truth in our automated evaluation of Osmium on MHal-Detect, leading to lower scores despite generating a valid labeling.

Subjectivity of Analysis. In our motivation for examining language priors, our hypothesis that analysis-type statements, such as inferences about emotions or aesthetics, may be especially language-driven (which was later validated by our experiments). Our hypothesis applied to all such analysis statements, whether or not they referred to information that was correct or a hallucination. Many ANALYSIS labels from Osmium reflect this behavior. However, human annotations often label analysis statements that refer to hallucinations as HALLUCINATION themselves. For example, in Fig. D.5 (bottom), Osmium appropriately labels *inferences* about a car and handbags as ANALYSIS, despite labeling statements about the *existence* of the car and handbags as HALLUCINATION. On the other hand, the human annotations label all of those statements as HALLUCINATION. We view either interpretation of the ANALYSIS category as correct.

D.5 Osmium Implementation

We use a 42-image subset of the MHal-Detect validation set to perform prompt tuning, and exclude these images from our later analyses. We access GPT-4 models through their API using `gpt-4-1106-preview` for GPT-4 and `gpt-4-vision-preview` for GPT-4V.

Prompts. We provide the prompt for Stage 1 with GPT-4V in Fig. D.7. For Stage 2 with GPT-4, the system prompt is shown in Fig. D.8, followed by the format of in-context learning examples in Fig. D.9.

Output format. We follow the same annotation format as human annotations in MHal-Detect [57]. Specifically, each entry $\mathcal{Y}_i$ (Eq. 5.1) is a JSON with keys `"start"` (holding the character index in the caption where the span annotation begins), `"end"` (holding the character index where the annotation ends), `"text"` (holding the span, a subset of the caption), and `"label"` (holding one of "ACCURATE" [Correct], "ANALYSIS", "INACCURATE" [Hallucination], or "UNSURE"). We find that it is difficult for GPT-4 to produce the exact start and end characters of the span, yet the text itself is almost always a valid subset of the caption. Therefore, we post-process the start and end indices to match with the span predicted by GPT-4, and discard any annotations that are not a valid substring of the caption – this is only a small fraction in practice.

In-context examples. We initially selected several examples from our 42-image prompt-tuning subset of MHal-Detect to use as in-context examples directly. However, when pairing the annotations with reference facts from GPT-4V, some images did not contain enough information in the reference facts to directly explain each human annotation. Therefore, we re-annotate eight captions ourselves to use as in-context examples with the corresponding reference facts in mind, ensuring that the reference facts contained sufficient information for annotation. We annotated spans as Unsure if they were correct, yet the reference facts did not mention the detail needed. Several of the in-context examples are shown in Fig. D.12.

D.6 Additional Details on VLMs and Caption Generation

For all models, we use public checkpoints provided by the authors of BLIP-2, InstructBLIP and LLaVA [96, 33, 112]. We use BLIP-2 checkpoints that are finetuned on COCO captioning [105], using the prompt "this is a picture of". As caption prompts for the InstructBLIP and LLaVA models, select a subset of the prompts for detailed caption generation from Liu et al. [112], shown in Fig. D.10. Terms of use for models are in their respective citations. We use 8 NVIDIA A100 GPUs and 8 NVIDIA RTX 2080 GPUs to run experiments.

Language prior computation. To compute language priors, we first generate the caption with beam search, using a beam size of 5 and a maximum length of 200 tokens. Next, given the selected tokens $t_{1:m}$, we run the VLM m times, progressively teacher-forcing more context $t_{1:a}$ for each $1 \leq a \leq m$, and taking the token distributions t_{a+1} (and t^L_{a+1}) for Φ computation. To compute language priors for LLaVA-based models, we simply remove the image embedding inputs to the LLM. For BLIP-2 and InstructBLIP, we remove the cross-attention between the query tokens and the image. We find that this produces slightly more informative language priors than removing the input embeddings to the LLM as we do for LLaVA.

D.7 Additional Details on Datasets

For MHal-Detect experiments, we use about 200 captions on 50 images (about 4 captions per image) from the validation set for prompt tuning. For the MHal-Detect evaluations we show in our work, we use 2,924 captions on 726 images. We compute token distributions by teacher-forcing the captions provided with the dataset using InstructBLIP/Vicuna-7B, to follow the model they were originally generated by [57]. We ensure that none of our evaluation images were used in our prompt tuning. Licenses for data in COCO [105] and ADE20K [214] can be found in the respective papers.

For ADE20K, we use a subset of 463 images. The Softmax score for LLaVA/Vicuna-13B in Tab. 5.2 was computed on 96 images due to computational constraints. When subsampling the larger ADE20K dataset, we select images from the following diverge range of categories: "cultural", "home or hotel", "industrial", "nature landscape", "shopping and dining", "sports and leisure", "transportation", "urban", "work place".

D.8 Qualitative Examples

We include many qualitative results of caption annotations: Fig. D.11 comparing Osmium labels to human annotations from MHal-Detect, Fig. D.12 displaying a few of our manually-annotated in-context learning examples, and randomly-selected samples from LLaVA/Vicuna-7B (Fig. D.13) and LLaVA/Vicuna-13B (Fig. D.14) on ADE20k labeled by Osmium.

D.9 Average Precision Computation

Eq. D.1 provides a definition of average precision as discussed in Sec. 5.4. P_j is the precision when using the jth score as a threshold to predict the positive label. It is multiplied by $(R_j - R_{j-1})$, the increase in recall over the previous threshold, and normalized by the number of true positives in the dataset, $\sum_{j=1}^{N} \mathbf{1}(y_j = 1)$.

$$\mathrm{AP} = \frac{\sum\limits_{j=1}^{N} (P_j \times (R_j - R_{j-1}))}{\sum\limits_{j=1}^{N} \mathbf{1}(y_j = 1)} \tag{D.1}$$

D.10 Impact and Ethical Considerations

As VLMs and AI models more broadly are becoming increasingly incorporated into society, the need for reliability becomes increasingly important as well. For example, VLMs may be used for automatically describing surrounding environments or online visual content to people with visual impairments. In these settings, getting a detail wrong can be misleading or even harmful for a user who may not be able to verify such output themselves.

Given the high rate of hallucination in open-source VLMs, these models are not yet well-suited for tasks that require reliable outputs. Although models such as GPT-4V may have high-quality visual understanding, as we have seen in our work, this type of powerful tool may not be accessible to those who need it, e.g., due to the cost of inference. Thus, research around the prevention of hallucinations is incredibly useful for advancing the accessibility of strong VLMs. Our work presents an important contribution to this area, experimentally confirming the degree to which language priors explain hallucination, and hopefully influencing the design of future VLMs with this in mind. We hope that our analysis framework, as well as our measure to automatically label captions much more densely than before, can drive research around the prevention of hallucination.

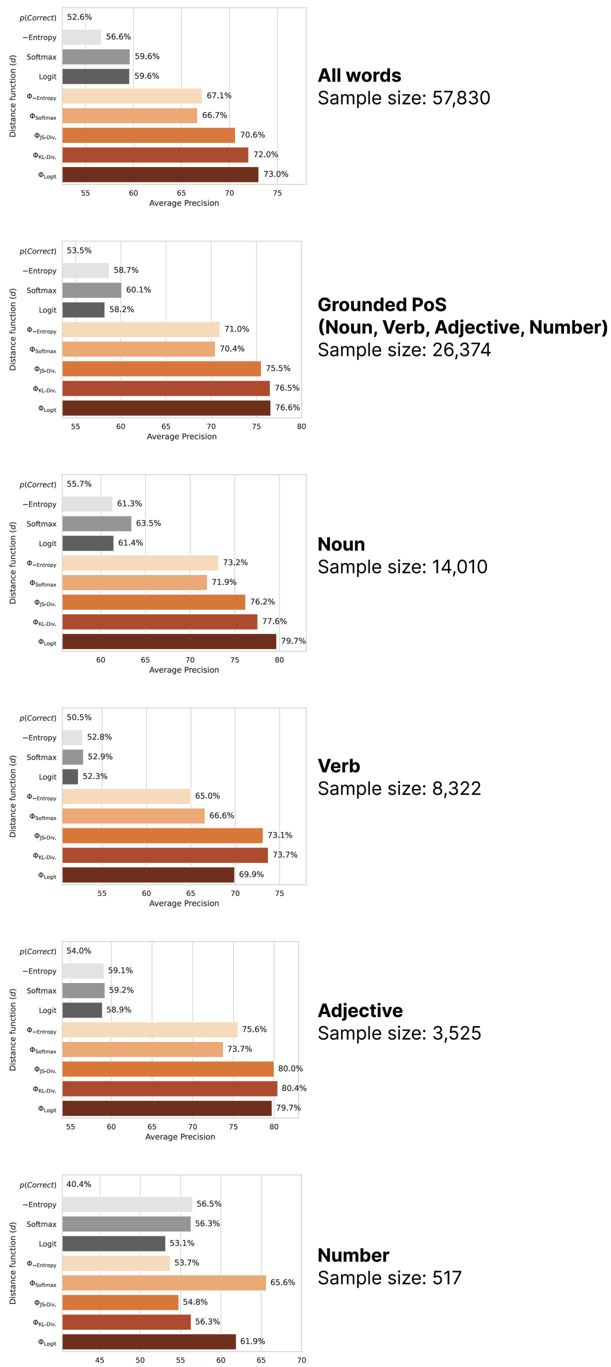


Figure D.4: Average precision results from Fig. D.2 on the MHal-Detect subset, subdivided by parts of speech (PoS).

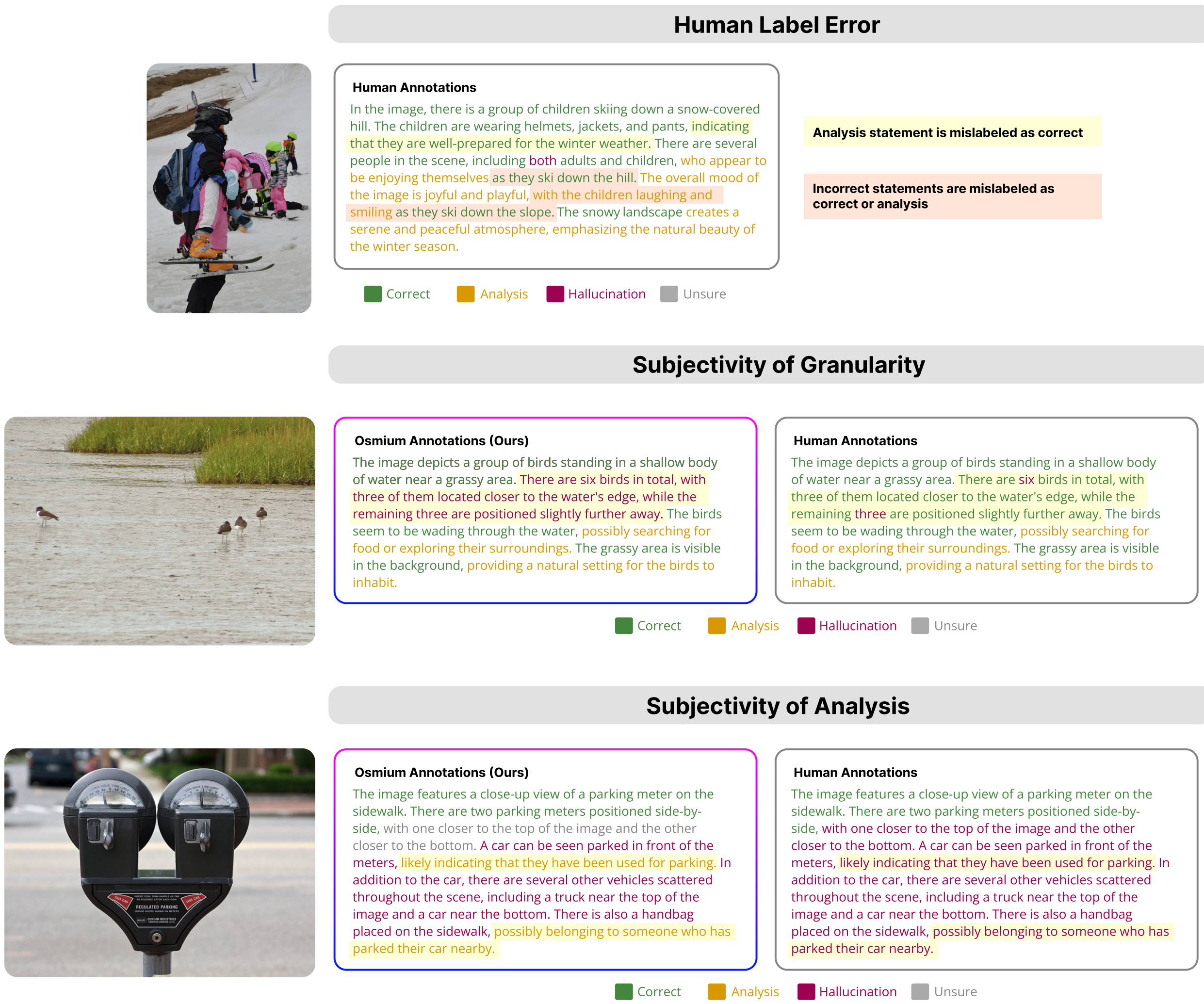


Figure D.5: Examples of sources of misalignment between Osmium and human annotations. **(Top)** The human annotations may contain errors themselves; e.g., the children are standing with their faces obscured, and are not skiing down the slope while laughing and smiling, yet these statements are mislabeled in the human annotations as CORRECT or ANALYSIS. **(Middle)** There may be subjectivity in deciding the granularity of HALLUCINATION statements, leading to a mismatch between Osmium and human annotations. In this example, Osmium labels an entire phrase as incorrect, whereas the human annotations label the singular words (number of birds) that contribute to the error. **(Bottom)** Within an incorrect statement, Osmium often continues to separate terms that are ANALYSIS, whereas human annotations often label the entire phrase as HALLUCINATION, as the analysis terms refer to erroneous information.

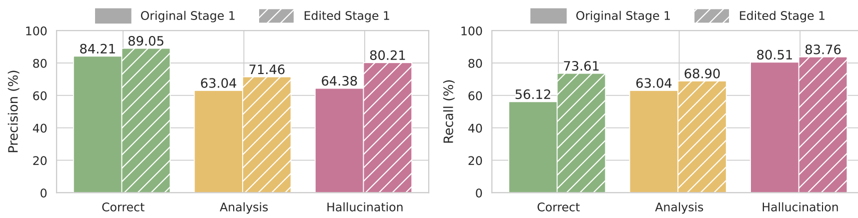


Figure D.6: We manually edit Stage 1 reference facts to match information in human annotations on a 56-image subset of MHal-Detect. We compute precision (left) and recall (right) of Osmium labels from Stage 2, showing the difference in scores using the original reference facts (solid bars) and edited reference facts (hatched bars). All metrics improve, with many remaining errors due to task subjectivity or human label error (see Fig. D.5).

```
Provide a bulleted list of succinct yet
comprehensive facts describing this image,
including the background and any objects that
are present.  If any people are present, make
sure to describe what they are wearing and doing.
Do not include any subjective statements or
inferences, only facts.  If you include a detail
that you are not sure about, be sure to identify
that it's uncertain.
```

Figure D.7: GPT-4V system prompt to obtain reference facts in Stage 1 of Osmium.

```
Given a list of known reference facts about an image, you will
annotate spans of text within a candidate caption into one of
the following four classes:

1. "ACCURATE": Span is DIRECTLY supported by evidence in the
reference. Do NOT label anything as accurate if there is no
explicit evidence in the reference to support it.

2. "INACCURATE": Span contradicts the reference, OR no
evidence in the reference to support it.

3. "ANALYSIS": Span includes complex reasoning or
interpretations about the image. These are portions of the
data that are SUBJECTIVE and not grounded visually within the
image, such as describing mood, assumptions, or emotions.

4. "UNSURE": You are uncertain as to which of the above 3
categories the span should be annotated as.

For each candidate caption, produce a list of annotations in
JSON-format, where each annotation has the following keys:

"text": a span of text within the predicted caption
"label": the corresponding label for the span
```

Figure D.8: GPT-4 system prompt to obtain dense caption annotations in Stage 2 of Osmium.

```
{System prompt}

For i = 1 to M:

 ** Sample i **
 REFERENCE FACTS: {reference facts}
 CANDIDATE CAPTION: {candidate caption}
 ANNOTATIONS: {JSON-format annotations}
 ...

** Sample M + 1 **
REFERENCE FACTS: {reference facts}
CANDIDATE CAPTION: {candidate caption}
ANNOTATIONS:
```

Figure D.9: Format of M in-context learning examples that appear after the system prompt, for GPT-4 in Stage 2 of Osmium.

"Provide a detailed description of what is presented in the photo."
"A detailed image description:"
"Explain the visual content of the image in great detail."
"Using language, provide a detailed account of the image."
"A detailed image caption:"
"Write a detailed description of the given image."
"Analyze the image in a comprehensive and detailed manner."
"Write a detailed description for the photo."
"Describe the content of the image in detail."
"Write a detailed description for the image."
"Please provide a detailed depiction of the picture."

Figure D.10: Prompts for long captions used with InstructBLIP and LLaVA models, taken or slightly modified from prompts in Dai et al. [33] and Gunjal et al. [57].

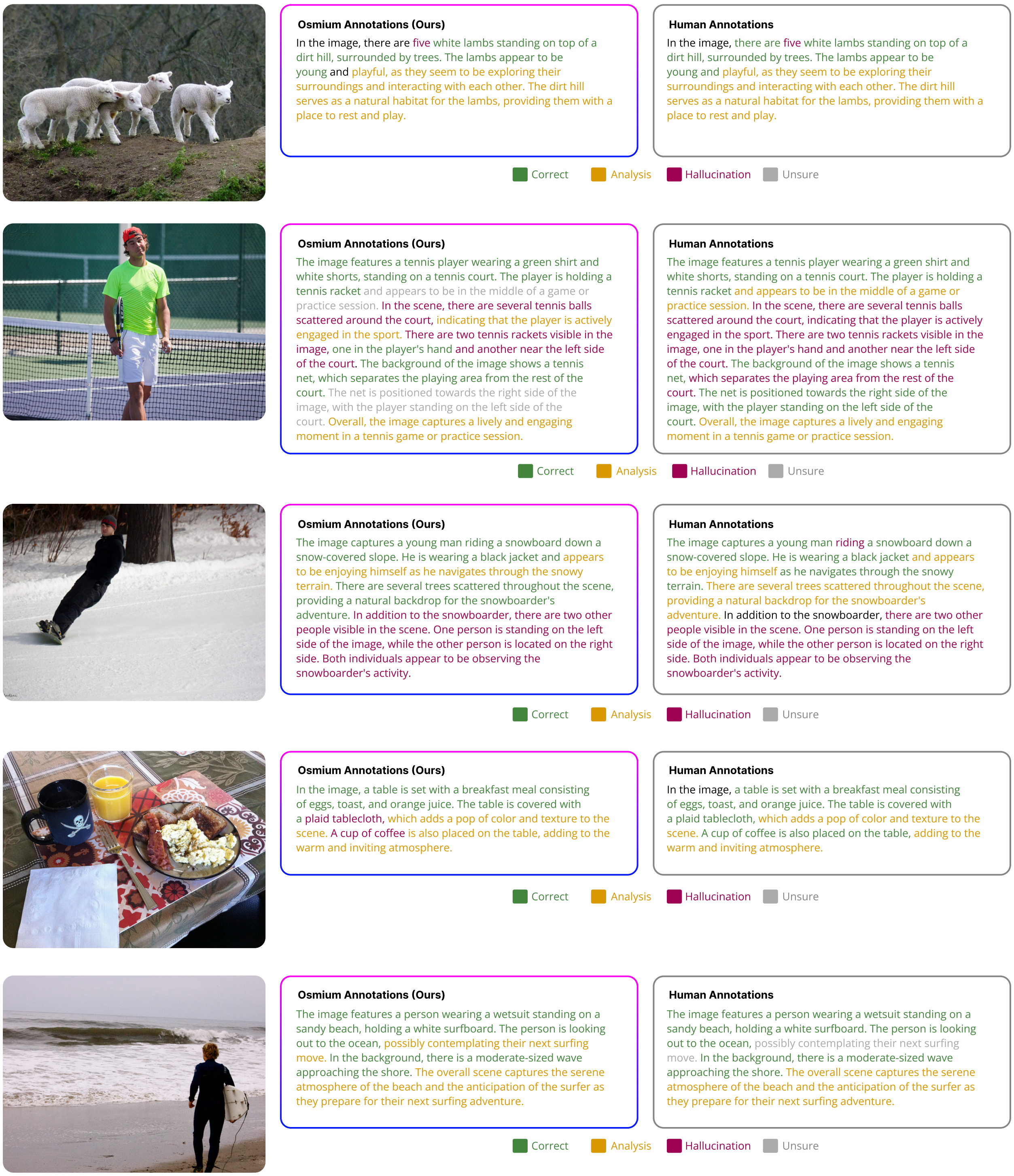


Figure D.11: We show several examples from the MHal-Detect dataset (and COCO images), comparing Osmium annotations to the provided human annotations. Note that black text indicates unlabeled words.

GPT-4V reference facts

- There are four doughnuts visible in an open box.
- The doughnut in the foreground is a glazed doughnut with a light brown, shiny glaze.
- To the right of the glazed doughnut is a chocolate frosted doughnut with visible textures that suggest it might be a cake doughnut.
- Behind the glazed doughnut, there appears to be a doughnut coated with powdered sugar.
- Next to the powdered sugar doughnut, there's another doughnut that seems to have a plain or lightly glazed surface.
- The box containing the doughnuts has a white interior.
- The text visible on the doughnut box is not clear enough to read due to the image focus.
- The background is blurred and indistinct, making it difficult to make out any specific details.
- The image is focused on the doughnuts in the box, with a shallow depth of field that blurs the background and part of the box itself.

Our Manual Annotations

The image features a white cardboard box filled with four different types of doughnuts. Three of the doughnuts are located on the left side of the box, while one doughnut is situated on the right side. The doughnuts come in various flavors and colors, such as chocolate, vanilla, and glazed. They are arranged in a visually appealing way, showcasing the diversity of the doughnut selection.

Correct Analysis Hallucination Unsure

GPT-4V reference facts

- The image shows a well-lit room that appears to be a living area.
- There are two windows with wooden blinds partially closed, and white window trim.
- The walls are painted a light color, possibly white.
- An assortment of furniture is present, including:
  - A black sofa, with several cushions
  - A red and white checked armchair with a matching ottoman
  - A dark-colored coffee table with various items including magazines
  - A dark grey sofa or daybed with an orange-red patterned throw
- A red shelving unit stands near the left wall, holding objects that cannot be identified with certainty.
- Plants can be seen on the windowsill in the background.
- A water cooler is positioned in the corner of the room between the two windows.
- Art and decorative items adorn the walls and the top of the red shelving unit.
- No people are present in the image.
- The room appears tidy and maintained.

Our Manual Annotations

The image depicts a cozy living room filled with various furniture and decorations. There is a large couch placed in the center of the room, surrounded by several potted plants. A dining table can be seen in the corner of the room, with chairs placed around it. In addition to the potted plants, there are several bottles scattered throughout the room. One bottle is placed on the left side of the room near the couch, while another can be found on the right side of the room, closer to the dining table. Two more bottles are positioned at the back of the room, one on the left side and the other on the right side. The living room also features a television mounted on the wall, providing entertainment for those spending time in the space. Overall, the room exudes a warm and inviting atmosphere, perfect for relaxation and social gatherings.

Correct Analysis Hallucination Unsure

GPT-4V reference facts

- There are assorted vegetables and a leaflet spread out on a wooden pallet.
- The vegetables include carrots with green tops, green onions with white bulbous bases and green shoots, yellow squash, and several green zucchinis.
- There is also a bunch of dark leafy greens that may be kale, with curly edges.
- Two red beets with green leaves attached are on the right side of the pallet.
- A portion of dark green herbs resembling dill is situated on the left side.
- There is a leafy vegetable with broad green leaves that could be chard or a similar green.
- A small plastic container labeled with "Humble House" and some text beneath that appears to be a branded product, potentially a type of sauce or condiment, is placed among the vegetables.
- A plastic bag containing a leafy substance is lying near the bottom left corner.
- A yellow leaflet is advertising a "CSA" with a website address listed and a logo that seems to be from "Suzie's Farm".
- The background includes sunlight and shadows indicating the photo may have been taken outdoors during the daytime.
- No people are present in the image.

Our Manual Annotations

The image features a wooden pallet filled with a variety of fresh vegetables, including broccoli, carrots, and radishes. The vegetables are neatly arranged on the pallet, creating a colorful display. There are at least 10 different types of vegetables visible in the scene. Some of the vegetables are placed closer to the center of the pallet, while others are positioned towards the edges.

Correct Analysis Hallucination Unsure

GPT-4V reference facts

- There are two people standing on a grassy hill.
- One person appears to be controlling a kite that is flying in the sky.
- The sky is blue with white clouds near the horizon.
- The kite has a red tail and seems to be mostly yellow with some red and possibly other colors.
- Both individuals are wearing dark-colored clothing, but specific details are not clear.
- The grass is a mix of green and brown, suggesting it may be partially dry.
- There are no other distinguishable objects or people in the immediate vicinity.

Our Manual Annotations

The image depicts a grassy field with two people flying a kite. One person is standing on the left side of the field, while the other person is on the right side. Both individuals are actively engaged in flying the kite, which appears to be a large and colorful one. The kite is flying high up in the sky, creating a vibrant display against the cloudy blue backdrop. In addition to the two people flying the kite, there are several smaller kites visible in the scene, adding to the festive atmosphere.

Correct Analysis Hallucination Unsure

Figure D.12: We show a few of our in-context learning examples, annotated ourselves with the GPT-4V reference captions in mind. We ensure that the reference facts are sufficient for determining our selected label.

GPT-4V reference facts

- The image shows an urban setting.
- There are no people present in the image.
- There are tall buildings on either side of a pedestrian walkway.
- The building on the leftappears to have a reflective glass facade.
- The building on the right has a series of windows and is labeled with the number 300.
- There is a lamppost in the center of the walkway.
- The sidewalk is made of rectangular paving stones.
- Young trees line the walkway, each supported by wooden stakes.
- An orange traffic cone is visible on the leftside of the walkway.
- There is a yellow bollard on the right side of the walkway.
- The sky is overcast, and the lighting suggests it may be either morning or evening.
- There are no visible moving vehicles or animals.
- The architecture and design imply a modern and planned urban space.

Osmium Annotations

The image depicts a city street with a row of trees lining the sidewalk. There are several potted plants placed along the sidewalk, adding a touch of greenery to the urban environment. The street appears to be empty, with no cars or pedestrians visible in the scene.
In addition to the trees and potted plants, there is a fire hydrant located near the center of the sidewalk,and a stop sign can be seen further down the street on the left side. The overall atmosphere of the scene is calm and peaceful.

Correct Analysis Hallucination Unsure

GPT-4V reference facts

- The image depicts a modern building with a large overhanging section and a recessed entrance area.
- The building has large glass windows and beige-colored walls with distinct horizontal lines and geometric shapes.
- The sign above the entrance reads "The University of Toledo".
- There are multiple people in front of the building, including:
- A person in a green shirt and black pants, walking towards the right.
- Several individuals in business attire, including suits and dresses; some are standing, while at least one person is walking.
- A group of people standing in a cluster, appearing to converse, near the entrance.
- The sky is clear and the lighting suggests it is daytime.
- There are trees and greenery visible around the building, indicating landscaped surroundings. There appear to be concrete planters with flowers near the entrance.
- One person is wearing sunglasses, associated with sunny weather.
- People in the background seem to be wearing identification badges or tags.

Please note that any assumptions about the building's function or people's specific activities are uncertain without additional context.

Osmium Annotations

The image depicts a group of people standing in line outside of a building, likely waiting to enter. There are at least 13 people visible in the scene, with some standing closer to the building and others further away. They appear to be patiently waiting for their turn to enter the building.
In addition to the people, there are two handbags visible in the scene. One handbag is located near the middle of the line, while the other is closer to the right side of the image.

Correct Analysis Hallucination Unsure

GPT-4V reference facts

- The image features a two-story house that appears aged and weathered.
- The architectural style of the house could be considered Victorian based on features such as bay windows and decorative trims, although that classification cannot be confirmed.
- The house appears to be in a state of disrepair, with visible damage to the roof and other parts of the structure.
- There are no visible people in the image.
- The sky is clear blue, suggesting a day with no visible clouds.
- The ground surrounding the house is barren, with no visible vegetation, implying a desert-like environment.
- There are no other buildings or structures immediately adjacent to the house.
- Shadows indicate that the sunlight is coming from the le
- There are no clear indications of the house's location or historical context.

Osmium Annotations

The image depicts a large, old, and run-down house situated in a desert-like environment. The house appears to be in a state of disrepair, possibly abandoned or neglected for a long time. The exterior of the house is made of stone, giving it a rustic and aged appearance.
There are two people visible in the scene, one standing closer to the left side of the house and the other closer to the right side, They seem to be observing the house, possibly admiring its architectural features or contemplating its history.

Correct Analysis Hallucination Unsure

GPT-4V reference facts

- The image features a large drilling rig.
- The sky is clear with no visible clouds.
- The drilling rig is predominantly red and white.
- There is snow on the ground, indicating cold weather conditions.
- No people are visible in the image.
- The rig is equipped with various industrial machinery and components. There are metal stairs leading up to parts of the rig.
- There are several large cylindrical tanks or containers near the base of the rig. A series of pipes or hoses appear to connect different parts of the operation.

Osmium Annotations

The image depicts a large oil rig in the middle of a snow-covered field. The oil rig is surrounded by several pipes and towers, with one of the towers reaching high into the sky. The field is covered in snow, giving the scene a wintry atmosphere.
In addition to the oil rig, there are two trucks visible in the scene. One truck is located on the left side of the oil rig, while the other truck is situated on the right side. The presence of these trucks suggests that they might be involved in the operation or maintenance of the oil rig.

Correct Analysis Hallucination Unsure

Figure D.13: We show several randomly-selected samples from ADE20K, with captions generated by LLaVA/Vicuna-7B and labels generated by Osmium.

GPT-4V reference facts

- The photo shows an indoor setting, likely a restaurant or pub, with multiple patrons seated at wooden tables.
- Vaulted ceilings with arches suggest a traditional or historic architecture.
- Walls are adorned with framed pictures or artworks.
- Lighting is provided by wall-mounted lantern-style fixtures.
- A woman in the foreground is seated with her back to the camera; she is wearing a burgundy tank top and has her hair styled in an updo.
- A blue denim jacket is draped over the back of the wooden bench the woman is sitting on. Patrons appear engaged in conversation or looking at menus.
- Glasses and what seem to be napkins or coasters are visible on the tables.
- In the background, a waiter or another patron is standing, although their activity is not discernible.
- No individual faces are clearly visible, preserving anonymity.
- The setting suggests a social atmosphere, as people are gathered in small groups.

Osmium Annotations

The image depicts a lively scene at a restaurant, where a group of people is gathered around a long wooden dining table, enjoying their meal and drinks. There are several people sitting and standing around the table, engaged in conversation and having a good time.
Various cups and bottles are placed on the table, indicating that the guests are enjoying beverages along with their meal. In addition to the cups and bottles, there are a couple of forks visible on the table, suggesting that the guests are eating as well.
The atmosphere appears to be casual and friendly, with everyone enjoying their time together at the restaurant.

Correct Analysis Hallucination Unsure

GPT-4V reference facts

- Two buses are parked in what appears to be a bus depot or parking area for buses.
- The bus on the left has a sign on the front indicating "310" which could be its route number.
- The bus on the right has "1010" written near its rear wheel and on its side, possibly representing a bus number or identifier.
- Both buses have a similar color scheme with shades of white, green, and blue.
- There is a noticeable amount of dirt and wear on the buses, indicating they may be well-used or older models.
- A partially obscured gas station is visible in the background, with a blue and white color theme. The sky is overcast, suggesting it might be a cloudy day.
- There do not appear to be any people in the immediate vicinity of the buses in the visible area of the image.
- The pavement is wet, which could mean recent rain or cleaning activities. No traffic or movement is evident in the image; the scene is static.

Osmium Annotations

The image features a parking lot with three buses parked next to each other. The first bus is on the left side of the parking lot, the second bus is in the middle, and the third bus is on the right side. The buses are lined up in a row, and they appear to be waiting for passengers or taking a break.
In addition to the buses, there are two people visible in the scene. One person is standing near the middle of the parking lot, while the other person is located closer to the right side of the parking lot.

Correct Analysis Hallucination Unsure

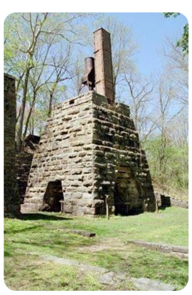
GPT-4V reference facts

- There are two individuals standing in a locker room.
- The floor appears to be wooden.
- The walls between and above the lockers are painted a pale yellow.
- The lockers themselves are yellow and have ventilation slits.
- The person on the left is wearing a white t-shirt with dark sleeves, dark pants, and is leaning against the lockers with one foot raised behind them.
- The person on the right is dressed in a burgundy top with dark sleeves, dark pants and is holding a smartphone at eye level, possibly taking a photo or looking at the screen.
- The person on the left is wearing a white t-shirt with dark sleeves, dark pants, and is leaning against the lockers with one foot raised behind them.
- There is a bench centered along the row of lockers.
- No other objects are visible in the locker room besides the lockers and the bench.
- The lighting in the room is bright and appears to be fluorescent based on the ambiance.
- The image has a watermark with the word "superstock".

Osmium Annotations

The image depicts two young women standing in a locker room, looking into their respective lockers. They are both wearing sports uniforms, indicating that they might be athletes or participating in a sports event.
The locker room is filled with numerous lockers, some of which are open, while others are closed. A bench can be seen in the room, providing a place for the women to sit or rest. The overall atmosphere of the locker room suggests that it is a place for athletes to store their belongings and prepare for their sports activities.

Correct Analysis Hallucination Unsure



GPT-4V reference facts

- Historical stone structure: The image features a large, old stone structure with a prominent chimney stack.
- Ruins appearance: The stone structure appears to be ruins of a larger building, with missing walls and roof.
- Arched openings: There are arched openings at the base of the structure, suggesting former doorways or furnace openings.
- Vegetation: The structure is surrounded by greenery, including grass and trees, which indicates it is located in a natural, outdoor setting.
- No visible people: There are no people discernible in the image.
- Chimney detail: The chimney is tall, narrow, and appears to be made of bricks, much of which is intact compared to the rest of the structure.
- Sky: It is a clear day with a bright blue sky in the background.
- Uncertain detail: The exact purpose of the structure and its historical context are not identifiable from the image alone.

Osmium Annotations

The image depicts an old stone structure, possibly a chimney or an oven, located in a grassy area. The structure is made of bricks and appears to be a part of an old building. The chimney is situated in the middle of the scene, surrounded by grass.
In addition to the stone structure, there is a bird perched near the top of the chimney, adding a touch of life to the scene. The overall atmosphere of the image conveys a sense of history and the passage of history and the passage of time.

Correct Analysis Hallucination Unsure

Figure D.14: We show several randomly-selected samples from ADE20K, with captions generated by LLaVA/Vicuna-13B and labels generated by Osmium.