

Learning without Expert Labels for Multimodal Data

Md Abdullah Al Maruf

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Application

Anuj Karpatne, Chair

T. M. Murali

Lifu Huang

Ismini Lourentzou

Wei-Lun Chao

December 4, 2024

Blacksburg, Virginia

Keywords: Deep Learning, Knowledge-Guided Machine Learning, Weak Supervision,
Self-Supervision, Vision-Language Models

Copyright 2024, Md Abdullah Al Maruf

Learning without Expert Labels for Multimodal Data

Md Abdullah Al Maruf

ABSTRACT

While advancements in deep learning have been largely possible due to the availability of large-scale labeled datasets, obtaining labeled datasets at the required granularity is challenging in many real-world applications especially in scientific domains, due to the costly and labor-intensive nature of generating annotations. Hence, there is a need to develop new paradigms for learning that do not rely on expert-labeled data and can work even with indirect supervision. Approaches for learning with indirect supervision include unsupervised learning, self-supervised learning, weakly supervised learning, few-shot learning, and knowledge distillation. This thesis addresses these opportunities in the context of multi-modal data through three main contributions. First, this thesis proposes a novel Distance-aware Negative Sampling method for self-supervised Graph Representation Learning (GRL) that learns node representations directly from the graph structure by maximizing separation between distant nodes and maximizing cohesion among nearby nodes. Second, this thesis introduces effective modifications to weakly supervised semantic segmentation (WS3) models, such as stochastic aggregation to saliency maps that improve the learning of pseudo-ground truths from class-level coarse-grained labels and address the limitations of class activation maps. Finally, this thesis evaluates whether pre-trained Vision-Language Models (VLMs) contain the necessary scientific knowledge to identify and reason about biological traits from scientific images. The zero-shot performance of 12 large VLMs is evaluated on a novel VLM4Bio dataset, along with the effects of prompting and reasoning hallucinations are explored.

Learning without Expert Labels for Multimodal Data

Md Abdullah Al Maruf

GENERAL AUDIENCE ABSTRACT

While advancements in machine learning (ML) such as deep learning have been largely possible due to the availability of large-scale labeled datasets, obtaining high-quality and high-resolution labels is challenging in many real-world applications due to the costly and labor-intensive nature of generating annotations. This thesis explores new ways for training ML models without relying heavily on expert-labeled data using indirect supervision. First, it introduces a novel way of using the structure of graphs for learning representations of graph-based data. Second, it analyzes the effect of weak supervision using coarse labels for image-based data. Third, it evaluates whether current ML models can recognize and reason about scientific images on their own, aiming to make learning more efficient and less dependent on exhaustive labeling.

Dedication

To my parents, Saifa Begum and Md Ahad Ali, whose love, sacrifices, and unwavering belief turned my dreams into reality.

Acknowledgments

I am deeply indebted to a number of people who have guided and encouraged me throughout my Ph.D. journey. First and foremost, I extend my deepest gratitude to my advisor, Professor Anuj Karpatne, for his unwavering support, guidance, and mentorship. His genuine care for my intellectual growth, coupled with his patience and encouragement, has shaped my work and helped me continually broaden my perspective and strive for excellence. I am fortunate to have him as a mentor; my doctoral experience would not have been as fulfilling without his care and belief in my potential.

I would also like to express my heartfelt gratitude to my doctoral committee members - Prof. T. M. Murali, Prof. Wei-Lun Chao, Prof. Lifu Huang, and Prof. Ismini Lourentzou - for their valuable insights and constructive feedback, which were instrumental in refining this dissertation and expanding my understanding of the field.

I am grateful to my colleagues in the KGML lab for creating a supportive, collaborative environment. Our exchange of ideas and insights was a constant source of inspiration. In particular, I am thankful to Dr. Arka Daw, Dr. Jie Bu, Dr. Mohannad Elhamod, Medha, Mridul, Harish, Abhilash, Kazi, Amartya, Blessy, Sepideh, Aanish, Arya, Ioannis, Naveen, and Snehal.

Beyond the lab, my friends in Blacksburg made this college town feel like home. I especially thank Dr. Jubayer Mahmud, Dr. Sazzadur Rahman, Dr. Shariful Islam, Dr. Murad Hassan, Dr. Rubayet Shafin, Dr. AB Siddik, and Sefat, along with many others, for their

warmth and enjoyable experience.

Nothing that I have ever accomplished could have been possible without the unwavering love and sacrifice of my parents. Their trust and confidence in me provided the foundation I needed to remain focused, resilient, and motivated through every step of my Ph.D.

Finally, I am thankful to all my long-time friends, teachers, mentors, cousins, and childhood heroes - Hamid, Zishan, Dr. Arif, Dr. Sarwar, Dr. Imtiaz, Dr. Swakkhar, Dr. Farid, Ifath, Siam, Hasan, Asif, Rashed, Harif, Topu, Zitu, Rahat, and Mukut - who shaped my thinking, guided my growth, and celebrated my progress. I am who I am today because of your influence and support.

Contents

| | |
|---|-------------|
| List of Figures | xiii |
| List of Tables | xix |
| 1 Introduction | 1 |
| 1.1 Challenges in Learning with Expert Labels | 3 |
| 1.2 Direct vs. Indirect Supervision | 3 |
| 1.3 Opportunities in Learning with Indirect Supervision for Multimodal Data | 4 |
| 1.4 Thesis Contributions | 6 |
| 2 Distance-Aware Graph Representation Learning | 9 |
| 2.1 Overview | 9 |
| 2.2 Related Work | 11 |
| 2.3 Preliminaries and Problem Objective | 13 |
| 2.3.1 Notations | 13 |
| 2.3.2 Cohesion and Separation | 14 |
| 2.3.3 GRL Objective | 16 |
| 2.4 Proposed Method | 17 |
| 2.5 Evaluation Setup | 23 |

| | | |
|----------|--|-----------|
| 2.6 | Results | 25 |
| 3 | Weak Supervision for Semantic Segmentation | 29 |
| 3.1 | Overview | 29 |
| 3.2 | Fundamental Concepts and Definitions | 32 |
| 3.2.1 | Class Activation Maps | 32 |
| 3.2.2 | Saliency Maps | 33 |
| 3.3 | Comparing CAMs and Saliency Maps | 34 |
| 3.3.1 | A Visual Comparison Using Hyperplanes | 34 |
| 3.3.2 | Perspective from Contribution Windows | 37 |
| 3.4 | Experimental Setup & Evaluation Metrics | 39 |
| 3.4.1 | Experimental Setup | 39 |
| 3.4.2 | Evaluation Metrics | 39 |
| 3.5 | Quantitative Comparison: CAM/Saliency | 41 |
| 3.5.1 | Effect of Contribution Window | 41 |
| 3.5.2 | Comparing NDR Recovery | 41 |
| 3.5.3 | Improving Saliencies with Simple Post-processing | 42 |
| 3.6 | Stochastic Aggregation of Saliencies | 43 |
| 3.6.1 | Smoothing Saliencies by Injecting Noise | 44 |
| 3.7 | Stochastic Aggregation Through Cropping | 45 |

| | | |
|----------|---|-----------|
| 3.7.1 | Disintegrating the Spatial Structure of Images using Random Cropping | 45 |
| 3.7.2 | Can we do better than random cropping? | 46 |
| 3.8 | Related Works | 47 |
| 3.9 | Discussion and Future Directions | 49 |
| 4 | Benchmarking Vision-Language Models on Scientific Dataset | 51 |
| 4.1 | Introduction | 51 |
| 4.2 | Related Works | 53 |
| 4.3 | VLM4Bio Tasks | 55 |
| 4.3.1 | Species Classification | 55 |
| 4.3.2 | Trait Identification | 56 |
| 4.3.3 | Trait Grounding and Referring | 56 |
| 4.3.4 | Trait Counting | 57 |
| 4.4 | VLM4Bio Dataset | 57 |
| 4.5 | Results | 60 |
| 4.5.1 | Analyzing the Role of Answer Choices in MC Questions on VLM Performance | 63 |
| 4.5.2 | Comparing Pre-trained VLMs with a Biologically Fine-tuned Model | 65 |
| 4.5.3 | Analyzing Effects of Prompting on VLM Performance | 65 |
| 4.5.4 | Analyzing Tests for Reasoning Hallucination | 67 |
| 4.6 | Limitations | 68 |

| | | |
|-------------------|---|-----------|
| 5 | Conclusion and Future Directions | 70 |
| 5.1 | Summary of the Thesis | 70 |
| 5.2 | Future Directions | 71 |
| Appendix A | Chapter 2 | 75 |
| A.1 | Proofs of Theorems | 75 |
| A.2 | Implementation Details | 83 |
| A.3 | Additional Analysis of Results | 85 |
| Appendix B | Chapter 3 | 88 |
| B.1 | Comparison of CAMs and Saliency Maps Using Hyperplanes | 88 |
| B.1.1 | Theoretical Proofs of CAM and SM-Hyperplanes | 88 |
| B.1.2 | Visual Comparison for more representative images from VOC | 91 |
| B.2 | Experimental Details | 91 |
| B.2.1 | Dataset Description | 91 |
| B.2.2 | Model Description | 92 |
| B.2.3 | Background Resolve Techniques | 93 |
| B.3 | Stochastic Aggregation for Saliencies | 95 |
| B.3.1 | SmoothGrad and BinaryMask | 95 |
| B.3.2 | Analysis of The Sensitivity Towards Noise. | 97 |
| B.4 | Stochastic Aggregation Through Cropping | 98 |

| | | |
|-------------------|--|------------|
| B.4.1 | Analysis of The Sensitivity For Random Cropping. | 98 |
| B.4.2 | Different Variations of Cropping | 98 |
| Appendix C | Chapter 4 | 108 |
| C.1 | Dataset Preprocessing | 108 |
| C.2 | Links to Access the Dataset and Its Metadata | 111 |
| C.3 | Dataset Availability and Maintanance | 111 |
| C.4 | Data Licenses | 112 |
| C.5 | Data Distribution and Key Statistics | 112 |
| C.6 | Traits Considered for the Task of Trait Identification | 113 |
| C.7 | Traits Considered for the Tasks of Trait Grounding and Referring | 114 |
| C.8 | VLM Baselines | 115 |
| C.9 | Prompts to Evaluate VLM performance | 115 |
| C.10 | Error Analyses for VLM Responses | 117 |
| C.11 | Comparing Pre-trained VLMs with a Biologically Fine-tuned Model | 118 |
| C.12 | Analyzing Effects of Image Resolution on VLM Performance | 120 |
| C.13 | Case Studies for Effects of Prompting on VLM Performance | 120 |
| C.13.1 | No Prompting | 120 |
| C.13.2 | Contextual Prompting | 121 |
| C.13.3 | Dense Caption | 121 |

| | |
|---|------------|
| C.13.4 Chain-Of-Thought Prompting | 121 |
| C.14 Case Studies for Reasoning Hallucination Tests | 122 |
| C.14.1 False Confidence Test (FCT) | 122 |
| C.14.2 None of The Above (NOTA) Test | 122 |
| Bibliography | 138 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | General learning framework of deep learning models. | 2 |
| 1.2 | Schematic Representation of the Key Contribution in Indirect Supervision for Multimodal Data. | 6 |
| 2.1 | Mapping a toy graph (2.1a) into three different 2D-embedding spaces: (2.1b), (2.1c), and (2.1d). The position of each node denotes the 2D-embedding vector and the dotted lines represent edges in \mathcal{G} | 16 |
| 2.2 | DNS-approx heuristic. | 22 |
| 2.3 | DNS-scalable heuristic. | 22 |
| 2.4 | t-SNE plot for embeddings generated by DeepWalk with Distance-aware Negative Sampler model (DeepWalk-DNS), DeepWalk with Unigram Negative Sampler model (DeepWalk-UNS), and Graph Auto Encoder model (GAE) on CiteSeer dataset. | 26 |
| 2.5 | Node classification performance (measured by F1-Macro score) plot with varying context window on CiteSeer, Cora, and PPI dataset. DeepWalk with Distance-aware Negative Sampler (DeepWalk-DNS) and with Unigram Negative Sampler (DeepWalk-UNS) are the competing models. | 26 |

| | | |
|-----|--|----|
| 2.6 | Average Pairwise Similarity of all node-pairs in embedding space where similarity = $\sigma(z_i^T z_j)$ for z_i and z_j node embeddings. Embeddings generated by DNS based GRL model show minimum similarity for distant nodes with the similarity decreasing with increasing distance d | 27 |
| 3.1 | A visual comparison of CAMs and saliency maps (SMs) for a representative image from the VOC12 dataset. | 35 |
| 3.2 | A schematic of “contribution window” demonstrating how the gradients at layer $l - 1$ is affected by the gradients from the contribution window of layer l | 39 |
| 3.3 | Effect of Contribution Window on NDR-Recall and mIoU for MNIST Dataset. | 42 |
| 4.1 | Overview of our goals and contributions. We analyze the capabilities of 12 state-of-the-art (SOTA) vision-language models (VLMs) in answering scientific questions using images from three groups of organisms: fishes, birds, and butterflies, over five groups of biologically relevant tasks. We also explore the effectiveness of these models for reasoning using various prompting techniques and tests for reasoning hallucination. | 52 |
| 4.2 | Illustrative examples of VLM4Bio tasks with different question-types. | 54 |
| 4.3 | Examples of correct and incorrect predictions of GPT-4V for trait identification, trait grounding, and trait-referring tasks related to the “eye”. For visualization assistance, a red-colored bounding box is added around the “eye” in the image. | 62 |
| 5.1 | Schematic Representation of the Potential Future Research Directions in Indirect Supervision for Multimodal Data. | 71 |

| | | |
|-----|---|-----|
| A.1 | F1-Macro score plot with varying context window on Synthetic Sparse, Synthetic Moderate, and Synthetic Dense dataset. Competing models are DeepWalk-UNS, DeepWalk-DNS, and its variants DeepWalk-DNS-min, DeepWalk-DNS-max. | 84 |
| A.2 | Sensitivity towards outlier points on CiteSeer dataset. | 85 |
| A.3 | Node classification performance (F1-Macro score) comparison for various γ -linear sampler based models with varying context window on Synthetic Sparse, Synthetic Moderate, and Synthetic Dense dataset. | 87 |
| B.1 | A visual comparison of CAMs and saliency maps (SMs) for more representative images from the VOC12 dataset. | 101 |
| B.2 | Visual comparison between CAM and Vanilla Saliency with different background resolves. | 102 |
| B.3 | Visual comparison of SmoothGrad saliencies between “Model-org” and “Model-pert-gaussian” fine-tuned model. Saliencies with basic background resolve are shown in the figure. | 103 |
| B.4 | Visual comparison of BinaryMask saliencies between “Model-org” and “Model-pert-binary” fine-tuned model. Saliencies with basic background resolve are shown in the figure. | 104 |
| B.5 | Sensitivity plots of the performance towards Gaussian noise levels σ (left); towards binary probability p (right). | 105 |
| B.6 | Sensitivity plots of the performance towards the number of samples n for (left) SmoothGrad; (right) BinaryMask. | 105 |

| | | |
|------|---|-----|
| B.7 | Visual evaluation of the sensitivity towards the noise level σ of the Gaussian noise (SmoothGrad saliency with basic background resolve). | 105 |
| B.8 | Visual evaluation of the sensitivity towards the binary probability of the perturbation (BinaryMask saliency with basic background resolve). | 106 |
| B.9 | Sensitivity plots of the performance for random cropping (left) to the number of crops; (right) to the scale of the crops. | 106 |
| B.10 | Visual comparison between Random Cropping, Random Patching, Discriminative Cropping, and Discriminative Patching saliencies. Saliencies with basic background resolve are shown in the figure. | 107 |
| C.1 | t-SNE plots to illustrate the effectiveness of random sampling with the majority species in the Fish-10K dataset. Randomly sampled images are shown as blue dots, while the remaining data points are represented by red dots. Sub-captions display the scientific names of the corresponding species. To generate the vector representation of the images, we leverage a VGG19 pretrained on the ImageNet dataset. | 109 |
| C.2 | Dataset Distribution of Fish-10K, Bird-10K, and Butterfly-10K. | 113 |
| C.3 | Trait list for Trait Identification task. | 114 |
| C.4 | Prompts Templates used for Evaluation. There will be no <options> for Open set questions. | 116 |
| C.5 | Analysis of errors for the pretrained VLM responses. | 117 |

| | | |
|------|--|-----|
| C.6 | Distribution of image resolutions for Fish-10K and Bird-10K are shown in Figures (a) and (b), respectively. The average score over image resolution for the GPT-4V, LLaVA-v1.5-7B, and LLaVA-v1.5-13B models on Fish-10K and Bird-10K are presented in Figures (c) and (d). We conduct the experiment in the context of the Species Classification task with Multiple-Choice (MC) questions. | 119 |
| C.7 | No Prompting. GPT-4o Correct prediction. Section C.13.1. | 123 |
| C.8 | No Prompting. GPT-4o Incorrect prediction. Section C.13.1. | 123 |
| C.9 | No Prompting. COG-VLM Correct prediction. Section C.13.1. | 124 |
| C.10 | No Prompting. COG-VLM Incorrect prediction. Section C.13.1. | 124 |
| C.11 | Contextual Prompting. GPT-4o Correct prediction. Section C.13.2. | 125 |
| C.12 | Contextual Prompting. GPT-4o Incorrect prediction. Section C.13.2. | 125 |
| C.13 | Contextual Prompting. LLaVa-13B Correct prediction. Section C.13.2. | 126 |
| C.14 | Contextual Prompting. LLaVa-13B Incorrect prediction. Section C.13.2. | 126 |
| C.15 | Dense Captions in Prompts. GPT-4o Correct prediction. Section C.13.3. | 127 |
| C.16 | Dense Captions in Prompts. GPT-4o Incorrect prediction. Section C.13.3. | 128 |
| C.17 | Chain-Of-Thought Prompting. GPT-4o Correct prediction. Section C.13.4. | 129 |
| C.18 | Chain-Of-Thought Prompting. GPT-4o Incorrect prediction. Section C.13.4. | 130 |
| C.19 | Chain-Of-Thought Prompting. LLaVa-13B Correct prediction. Section C.13.4. | 130 |
| C.20 | Chain-Of-Thought Prompting. LLaVa-13B Incorrect prediction. Section C.13.4. | 131 |

| | |
|--|-----|
| C.21 GPT-4o correct prediction. Section C.14.1. | 131 |
| C.22 LLaVa-13B incorrect prediction. Section C.14.1. | 132 |
| C.23 GPT-4o Correct prediction. Section C.14.1. | 132 |
| C.24 LLaVa-13B Incorrect prediction. Section C.14.1. | 133 |
| C.25 GPT-4o Correct prediction. Section C.14.1. | 133 |
| C.26 LLaVa-13B Incorrect prediction. Section C.14.1. | 134 |
| C.27 GPT-4o Correct prediction. Actual species name is <i>Esox Americanus</i> . Section C.14.2. | 134 |
| C.28 LLaVa-13B Incorrect prediction. Actual species name is <i>Esox Americanus</i> . Section C.14.2. | 135 |
| C.29 GPT-4o Correct prediction. Actual species name is <i>Corvus Albicollis</i> . Section C.14.2. | 135 |
| C.30 Blip-Flan-XL Incorrect prediction. Actual species name is <i>Corvus Albicollis</i> . Section C.14.2. | 136 |
| C.31 GPT-4o Incorrect prediction. Actual species name is <i>Batesia Hypochlora</i> . Section C.14.2. | 136 |
| C.32 Blip-Flan-XL Correct prediction. Actual species name is <i>Batesia Hypochlora</i> . Section C.14.2. | 137 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Summary statistics of the datasets we used for experiments where we choose the largest connected components from 390 components for CiteSeer and 78 components for Cora (PubMed and PPI are single connected component graph). We represent the largest component as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the set of unique class labels as y . PPI dataset has 121 classes with binary labels. The average node degree is represented by \overline{deg} , and the maximum node pair distance is denoted by d_{MAX} | 21 |
| 2.2 | The summary of the model performances in terms of downstream node classification F1-macro score. We highlight the best score for each dataset. For Cora, CiteSeer, PPI, and PubMed, we choose context window 4 to report the results. Both DW-DNS-approx and DW-DNS-scalable use 10% nodes as popular nodes. We run each model 5 times and report the performances in terms of mean and standard deviations. | 25 |
| 2.3 | Comparison of scalable heuristics on PubMed dataset. We denote the model size as <i>size</i> , training time per epoch as <i>time</i> with second as unit, and F1-Macro score of the downstream node classification task as <i>acc</i> | 28 |
| 3.1 | Quantitative comparison of CAM and Saliency on VOC dataset in terms of mIoU, Foreground Precision, and DR-/NDR-Recall. | 42 |

| | | |
|-----|--|----|
| 3.2 | Quantitative comparison of SmoothGrad and BinaryMask in terms of mIoU, FG-Precision, DR-/ NDR-Recall for different fine-tuned models on VOC dataset. The difference between the aggregated saliency performance and the vanilla saliency performance is shown in parentheses. A positive value denotes an increase in performance; whereas a negative value denotes a decrease in performance for aggregated saliencies. | 43 |
| 3.3 | Comparison of Random Crop, Discriminative Crop, Random Patch, and Discriminative Patch in terms of mIoU, FG-Precision, DR-/ NDR-Recall on VOC12. The difference between the aggregated and saliency performance is shown in parenthesis. | 46 |
| 3.4 | Quantitative comparison of CAM and Saliency on COCO dataset in terms of mIoU, Foreground Precision, and DR-/NDR-Recall. | 49 |
| 4.1 | Key statistics of the VLM4Bio dataset. | 58 |
| 4.2 | Zero-shot accuracy comparison of VLM baselines (in % ranging from 0 to 100) for the five scientific tasks. Results are color-coded as Best , Second best , Worst , Second worst | 60 |
| 4.3 | Zero-Shot accuracy comparison for <i>easy</i> , <i>medium</i> , and <i>hard</i> datasets. Results are color-coded as Best , Second best , Worst , Second worst | 64 |
| 4.4 | Zero-shot accuracy comparison for different prompting techniques of seven VLMs (in % ranging from 0 to 100). Results are color-coded as Best and Worst | 66 |
| 4.5 | Performance of seven VLMs on the NOTA and FCT reasoning tests. Results are color-coded as Best and Worst | 67 |

| | | |
|-----|---|-----|
| C.1 | Statistics of the VLM4Bio dataset. | 112 |
| C.2 | Zero-shot accuracy comparison of VLM baselines (in % ranging from 0 to 100) with BioCLIP for the species classification task. Results are color-coded as Best , and Worst | 119 |

Chapter 1

Introduction

With the advancements in deep learning techniques, Artificial Intelligence (AI) has become integral to numerous aspects of daily life, revolutionizing industries from healthcare, finance, and medical research to creative design. The progress observed in AI can be attributed to several key developments over the past decade.

First, the development of complex and large modeling frameworks, such as Transformers [127], Convolutional Neural Networks (CNN) [61], Graph Convolutional Networks (GCN) [55], Generative Adversarial Networks (GAN) [36], and Diffusion models [43], has revolutionized the predictive capabilities of deep learning models across multiple modalities, including text, images, and graphs. Additionally, the emergence of large foundational models, such as Large Language Models (LLMs) and Vision-Language Models (VLMs) like GPT-4V/4o [98, 100], Claude [9], LLaMA [27], Gemini [124], etc., provides versatile capabilities with numerous downstream applications across various domains. Second, the rapid advancement in storage and computing technologies, such as GPUs and TPUs, has enhanced the capabilities of deep learning methods in terms of time and space complexity.

Most importantly, large volumes of data from various modalities, including text, images, and speech, have become available from multiple sources such as internet crawling, crowd-sourced images from social media, camera traps, drones, and museums. For instance, the Common Crawl dataset [26, 106] comprises 386 TiB of text collected by crawling 2.7 billion web pages, and this dataset has been instrumental in pre-training large foundation models such as T5

[106] and LLaMA [125].

Conventional deep learning models learn through a supervised learning framework, utilizing labeled training datasets that are specifically curated for targeted tasks within various domains. When provided with input data, these models optimize the model parameters by (i) calculating the loss between their predictions and the corresponding target labels and then (ii) backpropagating the gradient of the loss to update the model parameters accordingly.

Figure 1.1 illustrates the general supervised learning framework of deep learning models. In this framework, $x^{(i)}$ represents the model's input, $\hat{y}^{(i)}$ signifies the model's prediction, and $y^{(i)}$ denotes the target variable, also known as ground-truth labels. Let us denote \mathbb{X} as the set of n input values and \mathbb{Y} as the set of n ground-truth labels in the training sets, i.e., $\mathbb{X} = \{x^{(i)}; i = 1, \dots, n\}$ and $\mathbb{Y} = \{y^{(i)}; i = 1, \dots, n\}$.

The quality and granularity of ground-truth labels \mathbb{Y} can categorize them as *expert labels* if every output instance, $y^{(i)}$, is labeled according to the required granularity for downstream tasks and without any imperfections or noisy labels.

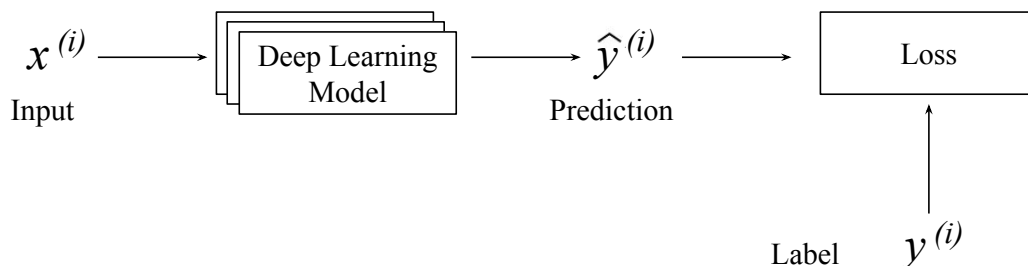


Figure 1.1: General learning framework of deep learning models.

1.1 Challenges in Learning with Expert Labels

Conventional deep learning models learn under the direct supervision of expert-labeled training data, which requires domain expertise and exhaustive annotation related to the downstream task. A significant advancement in deep learning models can be credited to the labeling of large amounts of unlabeled data, making it suitable for different downstream applications. For instance, substantial efforts were invested in creating the JFT-300M dataset [120], which consists of 300 million images across 18,291 categories for image classification. Similarly, the construction of the Segment Anything-1B dataset [57], used for image segmentation, required immense resources to annotate 11 million images with 1.1 billion segmentation masks.

However, the high dimensionality and complexity of certain domain-specific data make it challenging to annotate datasets with the required granularity for specific tasks. This results in a shortage of labeled training data and the occurrence of noisy labels. In contrast to the success in mainstream areas like computer vision and natural language processing, conventional deep learning models often struggle to perform well in these domain-specific tasks, which highlights a significant gap in their effectiveness across various fields, particularly in the scientific domain.

Given these challenges, there is a compelling need to develop deep learning models that learn without relying on expert labels for the downstream tasks.

1.2 Direct vs. Indirect Supervision

Direct supervision refers to the supervised learning paradigm in which the deep learning model learns directly from expert labels. In contrast, indirect supervision encompasses the

learning paradigm where the deep learning model operates with few or no expert labels.

Different approaches to indirect supervision include unsupervised learning, self-supervised learning, weakly supervised learning, few-shot learning, and knowledge distillation.

In **unsupervised and self-supervised learning**, there are no expert annotations available, and the deep learning model is trained solely on the information present in the input data. In **weakly supervised learning**, only coarse-grained or weak labels are provided instead of expert labels. The deep learning model is trained to utilize these weaker labels to generate predictions at the required level of detail for the downstream task. In few-shot learning, only a limited number of expert labels are available. When the downstream task has no labeled examples at all, it is referred to as **zero-shot learning**.

1.3 Opportunities in Learning with Indirect Supervision for Multimodal Data

The data distribution and structure across various modalities offer a unique opportunity for deep learning models to learn with few or no expert labels. Below, we outline the opportunities based on different modalities.

Graphs. Learning from graphs involves graph representation learning (GRL), where the goal is to learn low-dimensional representations of every node in the graph that capture the structure of interactions among the nodes, which are then used as inputs in downstream tasks such as network classification or link prediction. In conventional GRL techniques, every node has an associated set of attributes and target labels, and one can employ direct supervision from these node labels to extract node embeddings. However, in a general GRL problem, we may not always have access to node features or target labels. Therefore, a universal

GRL method should aim to learn node embeddings that capture the graph structure and are independent of downstream tasks. This raises an important research question: *How to develop a self-supervised or unsupervised GRL method that learns node representations directly from the graph structure?*

Images. Localizing semantic objects in an image with pixel-level labels is known as semantic segmentation, and training a segmentation model with the direct supervision of the pixel-level annotations requires a large number of images annotated with pixel-level labels. In many real-world applications, preparing such a detailed and large annotated dataset is challenging since the annotation process is slow, expensive, and requires substantial effort and expert knowledge. For example, preparing a pixel-level annotation for a single image in the Cityscapes dataset [22] takes approximately 90 minutes.

Meanwhile, it is much cheaper to obtain weak annotations that provide less information about the location of the objects, unlike pixel-level annotations. The cheapest and most popular weak supervision for semantic segmentation is image-level class labels that only provide information about the class of the objects present in the input image. Weakly supervised semantic segmentation methods leverage these coarse-grained class labels to generate pseudo-ground truths to train a segmentation model, which motivates us for the next research question. *How can weakly supervised semantic segmentation models effectively learn the pseudo-ground truths using class-level coarse labels?*

Images and Text. The availability of large text datasets and advancements in large-scale language pre-training have enabled large language models (LLMs) to achieve state-of-the-art (SOTA) performance on various language understanding and generation tasks. Inspired by the successes in the natural language processing domain, researchers in the vision-language community have employed vision pre-training, vision-language pre-training, and necessary adaptations to develop large foundation models such as vision-language models (VLMs) (e.g.,

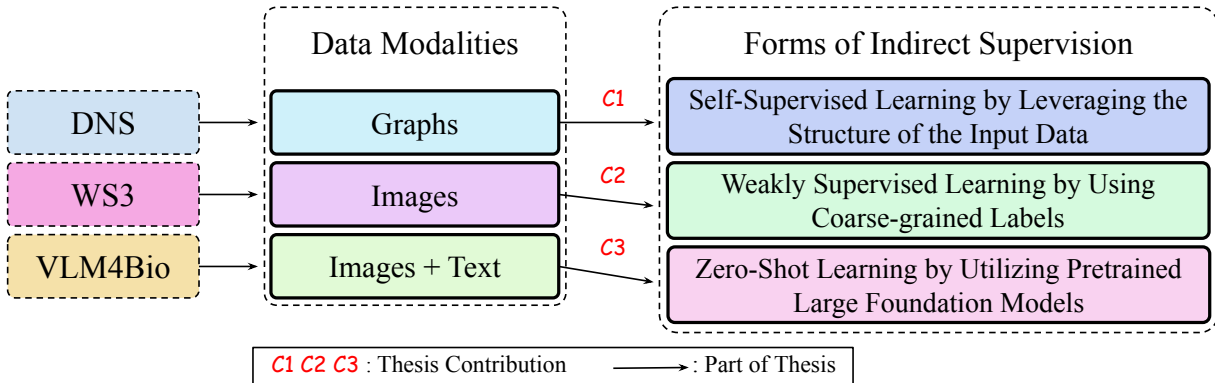


Figure 1.2: Schematic Representation of the Key Contribution in Indirect Supervision for Multimodal Data.

GPT-4, GPT-4V(ision) [97, 98], Gemini [124], LLaMA 3.2 [27, 125], and LLaVA [72]) that can simultaneously solve a diverse range of tasks involving text and images.

In light of the recent rise of large VLMs, there is a growing interest among the scientific community in harnessing the power of pretrained VLMs to understand scientific images and advance scientific discovery. However, unlike mainstream tasks in computer vision, understanding scientific images requires knowledge of domain-specific terminologies and reasoning capabilities that are not fully represented in conventional image datasets used for training VLMs. Hence, it is important to assess the zero-shot capabilities of pretrained VLMs in accelerating scientific discovery. This leads us to our final research question: *How can we evaluate the ability of the pre-trained VLMs whether they contain the necessary scientific knowledge to automatically identify and reason from scientific images?*

1.4 Thesis Contributions

This thesis aims to address the three research questions mentioned earlier. Figure 1.2 illustrates the schematic representation of the key contributions of this thesis in indirect

supervision for multimodal data, represented as **C1**, **C2**, and **C3**. The key contribution can be outlined as follows:

1. To learn the representations directly from the graph structure, this thesis proposes a novel Distance-aware Negative Sampling (DNS), which maximizes the separation of distant node-pairs while maximizing cohesion at nearby node-pairs by setting the negative sampling probability proportional to the pair-wise shortest distances.
2. In order to effectively learn the pseudo-ground truths using class-level labels, this thesis presents a novel insight into the strengths and weaknesses of saliencies with respect to Class Activation Maps (CAM) and shows that simple modifications such as stochastic aggregation to saliencies can effectively address the limitations inherent in CAMs. It also proposes three novel evaluation metrics for WS3, namely NDR-Recall, DR-Recall, and FG-Precision, that comprehensively assess WS3 performance of alternative methods with respect to CAMs.
3. This thesis evaluates the effectiveness of 12 state-of-the-art (SOTA) VLMs in the field of organismal biology using a novel dataset, VLM4Bio, consisting of 469K question-answer pairs involving 30K images from three groups of organisms: fishes, birds, and butterflies, covering five biologically relevant tasks. It also explores the effects of applying prompting techniques and tests for reasoning hallucination on the performance of VLMs, shedding new light on the capabilities of current SOTA VLMs in answering biologically relevant questions using images.

Thesis Organization

The remainder of this thesis is outlined as follows. Chapter 2 provides an overview of my research in distance-aware negative sampling for graph representation learning, where I leverage the input graph structure information to generate node representations. In Chapter 3, I illustrate how aggregated saliency maps are more effective than class activation maps for weakly supervised semantic segmentation while also addressing the limitations of popular approaches and suggesting possible solutions. Chapter 4 demonstrates the zero-shot capabilities and potential of large pretrained vision-language models in scientific tasks of organismal biology. Finally, Chapter 5 provides concluding remarks and future directions.

Chapter 2

Distance-Aware Graph Representation Learning

2.1 Overview

This chapter provides an overview of my research in distance-aware negative sampling for graph representation learning (GRL), where I leverage the input graph structure information to generate node representations. The goal of GRL is to learn a low-dimensional embedding of every node in the graph that captures the structure of interactions among nodes. The learned embeddings can be used as input features in downstream tasks such as network classification or link prediction. In GRL problems where every node has an associated set of attributes and target labels, e.g., over many benchmark datasets such as CiteSeer, Cora, and PubMed [144], one can employ supervised learning methods to extract node embeddings [34, 35, 40, 54, 128] that achieve state-of-the-art performance. However, in a general GRL problem, we may not always have access to node features or labels, or the node features may be available in complex and varying formats (e.g., as molecular structures in protein-protein interaction or drug-drug interaction graphs). Further, we may be interested in learning a “universal” embedding of the nodes that captures the graph structure and is independent of downstream supervised learning tasks. Such a universal representation can then be used as input features for a new downstream task without re-training the embeddings. For these

reasons, we focus our attention to the problem of unsupervised GRL, where the node embeddings are required to be learned solely from the graph structure (i.e., the adjacency matrix) and we do not consider the presence of any node or edge attributes or labels. Henceforth, we will use the term GRL to refer to unsupervised GRL.

Most GRL algorithms are rooted in the idea of distributional similarity developed in the natural language processing (NLP) community [84], whereby words appearing in similar *contexts* (e.g., sentences in a document) are mapped to similar representations. Similarly, most GRL algorithms aim to maximize the similarity of embeddings at nearby nodes, which are assumed to belong to similar contexts based on the structure of the graph. This is generally performed by maintaining a *positive* corpus of nearby node-pairs (termed positive pairs) and a *negative* corpus of randomly sampled node-pairs (termed negative pairs). The similarity of embeddings over the positive corpus is then contrasted with that over the negative corpus, and their difference is maximized to ensure positive pairs occupy similar embeddings. A common strategy for sampling the negative pairs is to use a unigram distribution over all nodes, referred to as the unigram negative sampling (UNS) method.

While maximizing the similarity at nearby node-pairs is an important objective, a second objective that is important yet mostly overlooked in existing GRL algorithms is to maximize the *dissimilarity at distant node-pairs*. This is important because ideally, we would like to learn embeddings where the structural similarity of nodes (e.g., based on the distance of the shortest path between two nodes, or network distance) is preserved in the embedding space. In other words, the similarity of node-pairs in the embedding space should be proportional to their network distance. As a result, by maximizing this second objective, we can obtain well-separated and meaningful embeddings, whereby node-pairs that are nearby occupy similar embeddings while those that are far apart occupy dissimilar embeddings. Using an analogy from the domain of clustering, we refer to the first objective as maximizing *graph cohesion*,

i.e., similarity at nearby nodes, and the second objective as maximizing *graph separation*, i.e., dissimilarity at distant nodes. We present an intuitive negative sampler for maximizing both cohesion and separation in GRL by sampling negative pairs with probability proportional to the distance between the nodes, termed as Distance-aware Negative Sampler (DNS).

Our Contributions: (1) We introduce and define the concepts of cohesion and separation in the context of GRL. (2) We propose a novel Distance-aware Negative Sampler (DNS) that maximizes both cohesion and separation. (3) We theoretically show the effectiveness of our DNS approach in maximizing cohesion and separation as compared to UNS. (4) We present a scalable DNS approach with reduced space and time complexity for large networks. (5) We empirically show the ability of our DNS approach to learn meaningful representations, thus leading to better predictive performance on downstream ML tasks on several benchmark datasets in comparison with baseline GRL algorithms.

2.2 Related Work

Unsupervised graph representation learning methods: A number of existing unsupervised GRL methods maximize embedding similarity at nearby nodes directly without performing negative sampling. Some examples include matrix factorization based methods [8, 14, 101] and skip-gram based methods [7, 11, 102]. Some GRL methods use a variety of negative sampling strategies to learn node embeddings. This category includes methods that use input node features such as Graph Convolutional Network (GCN) encoders [40, 56, 129] that have achieved state-of-the-art performances on benchmark GRL datasets. However, they are not directly relevant to our GRL problem since we consider the formulation where no node features are available. Negative sampling based methods that do not use node fea-

tures include node2vec [38], which optimizes random walk objectives and LINE [123], which uses first- or second-order neighborhoods to construct similar nodes. Note that while DeepWalk [102] was originally proposed using a Hierarchical Softmax objective, we can adapt it to construct a negative sampling based version of DeepWalk.

Negative sampling strategies: Here we discuss some of the common strategies for negative sampling that are at the basis of several unsupervised GRL algorithms. There are two generic types of negative samplers, edge-based [56, 123] and node-based [38, 40]. Edge-based samplers construct the positive corpus by selecting node pairs that have an edge between them, and the negative corpus by randomly selecting node pairs that do not have an edge. On the other hand, node-based samplers use random walk objectives to construct the positive corpus and select random node pairs distributed with unigram distribution to construct the negative corpus. Among unigram distributions, two are common; one chooses negative samples with uniform probability [56, 123] and the other uses degree-based probability [38, 40], where the negative sampling probability is proportional to the $\frac{3}{4}$ th power of the degree of each node. It is known that degree-based unigram sampler suffers from the *popular neighbor* problem [11], as this approach may choose a nearby node with high degree as a negative sample. Henceforth, by Unigram Negative Sampler (UNS), we refer the unigram sampler with uniform probability, and unigram-deg/UNS-deg denotes degree-based unigram negative sampler. There are some more negative samplers that have been proposed in recent works [11, 141]; however, none of them use the notion of network distances in negative sampling.

2.3 Preliminaries and Problem Objective

2.3.1 Notations

We are given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $|\mathcal{V}| = n$, $|\mathcal{E}| = m$, and the adjacency matrix is given by $\mathbf{A} = [a(i, j)]_{n \times n}$. We assume that the graph is unweighted such that $a(i, j) = 1$ iff $(i, j) \in \mathcal{E}$, otherwise 0. We denote the set of all possible node-pairs as $\mathcal{S} = \mathcal{V} \times \mathcal{V}$. Further, for every node-pair $(i, j) \in \mathcal{S}$, we denote the distance or length of the shortest path between the nodes as $d(i, j)$. Incidentally, $d(i, j) = 1$ iff $(i, j) \in \mathcal{E}$, i.e., there exists an edge between nodes i and j . Let us refer to the maximum value of $d(i, j)$ in graph \mathcal{G} as d_{max} . We can then talk about the subset of node-pairs whose distance is equal to d , i.e., $\mathcal{S}_d = \{(i, j) \in \mathcal{S} | d(i, j) = d\}$. It is easy to verify that $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \dots \cup \mathcal{S}_{d_{max}}$ and $\mathcal{S}_1 = \mathcal{E}$.

With this setup, we consider the problem of unsupervised GRL where the goal is to map every node i to an l -dimensional vector embedding, $\mathbf{z}_i \in \mathbb{R}^l$, such that the embedding space $\mathcal{Z} = \{\mathbf{z}_i\}_{i=1}^n$ preserves the structural properties of nodes in graph \mathcal{G} (typically, $l \ll |\mathcal{V}|$). In particular, we consider two generic types of measures in the embedding space of a pair of nodes, (i) $\text{SIM}_{\mathcal{Z}}(i, j) :=$ similarity score between embeddings \mathbf{z}_i and \mathbf{z}_j (some examples include the dot product $\mathbf{z}_i^T \mathbf{z}_j$ and its monotonic transformations $\sigma(\mathbf{z}_i^T \mathbf{z}_j)$ and $\log(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$, where σ denotes the sigmoid function), and (ii) $\text{DISSIM}_{\mathcal{Z}}(i, j) :=$ dissimilarity score between embeddings \mathbf{z}_i and \mathbf{z}_j (some examples include $-\mathbf{z}_i^T \mathbf{z}_j$, $\sigma(-\mathbf{z}_i^T \mathbf{z}_j)$ and $\log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))$). Note that there are multiple choices of similarity and dissimilarity functions to instantiate these two generic measures in any problem. Also, maximizing the similarity score of a node-pair is usually equivalent to minimizing its dissimilarity score for common function choices.

Ideally, we want to learn an embedding space \mathcal{Z} such that $\text{SIM}_{\mathcal{Z}}(i, j)$ is large for nearby node-pairs (i.e., when $d(i, j)$ is small) and $\text{DISSIM}_{\mathcal{Z}}(i, j)$ is large for distant node-pairs (i.e.,

when $d(i, j)$ is large). This objective, which is at the basis of the distributional hypothesis in linguistics [42], can be expressed using the notions of *cohesion* and *separation* in GRL, formally defined in the following.

2.3.2 Cohesion and Separation

Cohesion: The cohesion of an embedding space \mathcal{Z} represents the aggregate similarity score between embeddings at nearby node-pairs in the graph. Formally, we define cohesion using the following weighted sum over similarity scores:

$$\text{COHESION}(\alpha, \mathcal{Z}) = \sum_{d=1}^{d_{max}} \alpha_d \sum_{(i,j) \in \mathcal{S}_d} \text{SIM}(i, j),$$

where $\alpha_d \geq \alpha_{d+1}$, $\alpha_d \geq 0 \quad \forall d$.

Observe that since the weights α_d monotonically decrease with d , this weighted sum pays greater emphasis to the similarity at nearby node-pairs (i.e., \mathcal{S}_d with small d). This is a generic definition of cohesion that can be instantiated using different choices of the weights α . For example, if we specify $\alpha_1 = 1$ and $\alpha_d = 0 \quad \forall d > 1$, then cohesion will be equal to the aggregate similarity over all the edges in \mathcal{G} . As we will see later, a common approach for specifying α_d in most GRL algorithms is performing random walks and computing the probability of sampling a node-pair at a distance d in the random walk.

Separation: The separation of an embedding space \mathcal{Z} captures the aggregate dissimilarity between embeddings at distant node-pairs. Similar to cohesion, we can formally define separation using the following weighted sum:

$$\text{SEPARATION}(\beta, \mathcal{Z}) = \sum_{d=1}^{d_{max}} \beta_d \sum_{(i,j) \in \mathcal{S}_d} \text{DISSIM}(i, j),$$

where $\beta_d \leq \beta_{d+1}$ $\beta_d \geq 0 \quad \forall d$.

In this generic definition, since β_d monotonically increases with d , the dissimilarity at distant node-pairs have a greater contribution in the separation. Again, there can be multiple ways to instantiate β_d . For example, we can set $\beta_{d_{max}} = 1$ and $\beta_d = 0 \quad \forall d < d_{max}$ such that the separation is equal to the dissimilarity at the farthest node-pairs in \mathcal{G} .

Since dissimilarity score is inversely related to similarity, it may seem that maximizing cohesion automatically maximizes separation. However, this is not true since the weighted sums involved in cohesion and separation focus on different subsets of node-pairs in \mathcal{S} : while cohesion focuses on \mathcal{S}_d for small d , separation focuses on node-pairs in \mathcal{S}_d for large d . We present the following theorem to prove this point.

Theorem 2.1. *Given two embedding spaces, \mathcal{Z}^1 and \mathcal{Z}^2 , It does not imply if $\text{COHESION}(\alpha, \mathcal{Z}^1) = \text{COHESION}(\alpha, \mathcal{Z}^2)$ then $\text{SEPARATION}(\beta, \mathcal{Z}^1) = \text{SEPARATION}(\beta, \mathcal{Z}^2)$, for all choices of α and β .*

Proof. We use a counter-example to show that there can exist multiple embedding spaces such that their cohesion values are equal but their separation values are different. Figure 2.1a shows a toy graph with 4 nodes that is represented in three different two-dimensional embedding spaces in Figures 2.1b, 2.1c, and 2.1d. If we specify cohesion to be the aggregate similarity over edges (shown as dotted lines), we can see that all three embedding spaces have the same cohesion. However, if we define separation to be the aggregate dissimilarity at farthest nodes (at distance 3), we can see that the separation ranges from large (Figure

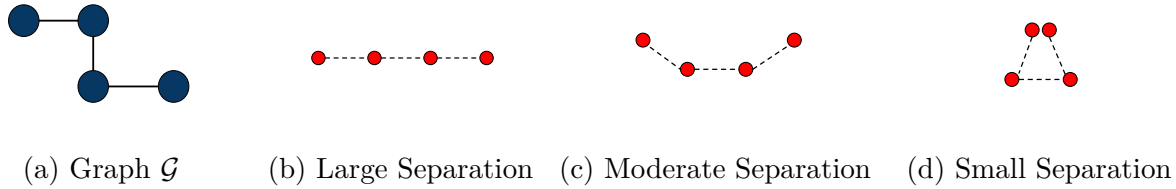


Figure 2.1: Mapping a toy graph (2.1a) into three different 2D-embedding spaces: (2.1b), (2.1c), and (2.1d). The position of each node denotes the 2D-embedding vector and the dotted lines represent edges in \mathcal{G} .

2.1b) to small (Figure 2.1d). □

2.3.3 GRL Objective

As a result of Theorem 2.1, a GRL algorithm that only maximizes cohesion is not guaranteed to maximize separation and thus can lead to inferior embeddings such as the one shown in Figure 2.1d for the toy graph. This is one of the major drawbacks of skip-gram based GRL algorithms that only attempt to maximize the similarity at nearby nodes (where neighborhood is defined using random walks). We posit this as a natural consequence of the origin of these algorithms in natural language processing (NLP) applications, where the definition of distances between words (and hence the separation) is not as straight-forward as in graphs. We thus present a generalized objective of GRL using both cohesion and separation. We can show that existing GRL algorithms (e.g., unigram negative sampling based approaches) optimize special cases of this GRL objective.

Generalized GRL Objective: Given a graph \mathcal{G} , the goal of a GRL algorithm is to optimize the following generalized objective function with respect to \mathcal{Z} :

$$\begin{aligned}
 E(\mathcal{Z}) &= \text{COHESION}(\alpha, \mathcal{Z}) + \text{SEPARATION}(\beta, \mathcal{Z}) \\
 &= \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} [\alpha_d \text{SIM}(i, j) + \beta_d \text{DISSIM}(i, j)] \\
 \text{such that, } &\frac{\alpha_1}{\beta_1} \gg 1, \quad \frac{\alpha_{d_{max}}}{\beta_{d_{max}}} \ll 1, \quad \text{and} \quad \frac{\beta_{d_{max}}}{\beta_1} \gg 1
 \end{aligned}$$

Note that we do not use a trade-off parameter between cohesion and separation since any such parameter can be absorbed in α or β as a constant multiplier. Different GRL algorithms optimize this generalized objective using different choices of similarity and dissimilarity functions, and settings of α and β weights satisfying the GRL conditions in the above equation. From the perspective of separation, we would prefer a GRL algorithm that employs a large value of $\beta_{d_{max}}/\beta_1$, such that the dissimilarity at farthest node-pairs is substantially larger than that of the nearest node-pairs. We call this fraction $\beta_{d_{max}}/\beta_1$ as the **Separation Power** of a GRL algorithm.

2.4 Proposed Method

Negative sampling: Before we present our proposed GRL algorithm based on the ideas of cohesion and separation, we formally discuss the generic family of negative sampling algorithms of which our algorithm is a special case. The objective function of negative sampling is given by the following equation:

$$\max_{\mathbf{z}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}(i)} \underbrace{[\log \sigma(\mathbf{z}_i^T \mathbf{z}_j)]}_{\text{Positive Loss}} + K \sum_{k \in \mathcal{V}} \underbrace{P_{neg}(k|i) \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_k))}_{\text{Negative Loss}},$$

where node-pair (i, j) belongs to the positive corpus D_{pos} while (i, k) belongs to the negative corpus D_{neg} . We generally use random-walk strategy to construct D_{pos} , whereas D_{neg} is constructed by sampling K negative pairs (i, k) for each positive sample (i, j) with probability $P_{neg}(k|i)$ [39, 84]. A common choice of $P_{neg}(k|i)$ is the unigram distribution that samples k with equal probability from all n nodes, referred as the Unigram Negative Sampling (UNS) algorithm.

The objective function of UNS can be shown to be a special case of the generalized GRL objective where the similarity at nearby nodes (i.e., cohesion) corresponds to the positive loss while the dissimilarity at distant nodes (i.e., separation) corresponds to the negative loss. However, a major limitation with UNS is that the probability of sampling a negative node-pair is independent of the distance between the nodes. As a result, UNS pays equal importance to the dissimilarity of node pairs with varying distances in the calculation of separation, thus leading to poor separation power. Theorem 2.2 provides a formal analysis of the correspondence of UNS to the generalized GRL objective and shows that its separation power is equal to 1.

Theorem 2.2. *Unigram Negative Sampling (UNS) Algorithm optimizes the generalized GRL objective with the following specifications: $SIM(i, j) = \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$, $DISSIM(i, j) = \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))$, $\alpha_d = \pi_d(C, \mathbf{A})$, where $\pi_d(C, \mathbf{A})$ is the probability of sampling a node-pair at distance d using a C -length random walk on the graph with adjacency matrix \mathbf{A} , and*

$\beta_d = KC/n$. As a result, the Separation Power of UNS algorithm is equal to 1.

Proof. Provided in Appendix A. □

Distance-aware Negative Sampler: We propose a Distance-aware Negative Sampler (DNS) which selects a negative sample k for node i using the sampling probability $P_{neg}(k|i)$, where $P_{neg}(k|i)$ is linearly proportional to the pair-wise distance $d(k, i)$. Formally,

$$P_{neg}(k|i) \propto d(k, i)$$

$$P_{neg}(k|i) = \frac{d(k, i)}{\mathcal{D}(i, \mathbf{A})},$$

where $\mathcal{D}(i, \mathbf{A})$ is the sum of distance of all node-pairs that contain node i , $\mathcal{D}(i, \mathbf{A}) = \sum_{s \in \mathcal{V}} d(s, i)$. Let $\mathcal{D}(\mathbf{A})$ be equal to $\mathbb{E}_i(\mathcal{D}(i, \mathbf{A}))$. Note that $\mathcal{D}(\mathbf{A})$ depends on the average degree of the graph. For a fully connected graph, $\mathcal{D}(\mathbf{A}) = n - 1$. On the other extreme, when the graph is a chain of n nodes, then $\mathcal{D}(\mathbf{A}) = \frac{n(n-1)}{2}$. Generally, since most real world graphs are sparse, $\mathcal{D}(\mathbf{A}) \gg n - 1$. By construction, our proposed DNS approach has a separation power of d_{max} as stated in Theorem 2.3.

Theorem 2.3. *Distance-aware Negative Sampling (DNS) Algorithm optimizes the generalized GRL objective with the following specifications: $SIM(i, j) = \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$, $DISSIM(i, j) = \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))$, $\alpha_d = \pi_d(C, \mathbf{A})$, where $\pi_d(C, \mathbf{A})$ is the probability of sampling a node-pair at distance d using a C -length random walk on the graph with adjacency matrix \mathbf{A} , and $\beta_d = KCd/\mathcal{D}(\mathbf{A})$. As a result, the Separation Power of DNS algorithm is equal to d_{max} .*

Proof. Provided in Appendix A. □

Corollary 2.4. *For UNS, $(\frac{\alpha_d}{\beta_d})_{UNS} = \frac{\pi_d(C, \mathbf{A})n}{KC}$ and for DNS, $(\frac{\alpha_d}{\beta_d})_{DNS} = \frac{\pi_d(C, \mathbf{A})\mathcal{D}(\mathbf{A})}{KCd}$. Hence, $(\frac{\alpha_d}{\beta_d})_{UNS} < (\frac{\alpha_d}{\beta_d})_{DNS}$ when $n < \frac{\mathcal{D}(\mathbf{A})}{d}$.*

The above corollary helps us understand useful operating points of DNS. Since $\mathcal{D}(\mathbf{A}) \gg n$ for most graphs, the (α/β) ratio is generally always larger for DNS than UNS. We have also empirically observed that the (α/β) ratio increases with C for all graphs considered in this work. As a result, DNS operates better at lower values of C since (α/β) ratios remain small for moderate values of d . Additionally, we have empirically observed that DNS works better for sparse graphs since there is a larger spread in the network distances across all node-pairs, making it possible for DNS to maximize separation in the embedding space for distant nodes.

The embedding space learned by DNS indeed preserves the graph-based similarity structure among nodes. Formally, Theorem 2.5 shows that the pairwise similarity in embedding space is a function of node-pair distance and for negative node-pairs, the similarity is inversely proportional to the distance.

Theorem 2.5. *Let the average pairwise similarity for any two nodes at distance d be given by $\xi_d = \frac{1}{|\mathcal{S}_d|} \text{SIM}(i, j) = \frac{1}{|\mathcal{S}_d|} \sum_{(i,j) \in \mathcal{S}_d} \sigma(z_i^T z_j)$. We can then show that DNS generates embeddings such that ξ_d is a function of d and for $d > C$, ξ_d is inversely proportional to d .*

Proof. Provided in Appendix A. □

While negative sampling with linearly proportional distances is a simple heuristic, we can have a more general form of DNS by adding super-linearity or sub-linearity in the negative sampling probability which is, $P_n(r|u) \propto (d(r, u))^\gamma$. Here γ is a hyper-parameter and we can vary γ based on the properties of the dataset.

Complexity analysis: DNS requires pairwise shortest distance computation for all node pairs as a preprocessing step. The time complexity to compute all-pair shortest path lengths is $\Theta(nm + n^2 \log n)$ [25, 32] and the space complexity is $\mathcal{O}(n^2)$ to store the normalized probabilities for all node pairs. While there are efficient techniques to precompute shortest distances

Table 2.1: Summary statistics of the datasets we used for experiments where we choose the largest connected components from 390 components for CiteSeer and 78 components for Cora (PubMed and PPI are single connected component graph). We represent the largest component as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the set of unique class labels as y . PPI dataset has 121 classes with binary labels. The average node degree is represented by \overline{deg} , and the maximum node pair distance is denoted by d_{MAX} .

| Stat | CiteSeer | Cora | PubMed | PPI | Syn. Sparse | Syn. Moderate | Syn. Dense |
|------------------|----------|--------|--------|--------|-------------|---------------|------------|
| $ \mathcal{V} $ | 2,120 | 2,485 | 19,717 | 2,339 | 2,000 | 2,000 | 2,000 |
| $ \mathcal{E} $ | 7,358 | 10,138 | 88,648 | 65,430 | 4,982 | 12,062 | 30,472 |
| $ y $ | 6 | 7 | 3 | 121 | 7 | 5 | 4 |
| d_{MAX} | 28 | 19 | 18 | 7 | 106 | 82 | 4 |
| \overline{deg} | 3.47 | 4.08 | 4.5 | 27.97 | 2.49 | 6.03 | 15.24 |

that can be coupled with our approach [20, 33, 52], our basic DNS-based model would still require $\mathcal{O}(n^2)$ space to store the normalized probabilities, which is not scalable.

Scalable DNS Approach: We develop an approach that reduces the space complexity of our DNS model without increasing the training time complexity. In this approach, we compress the pairwise distance matrix during preprocessing and decompress it back during training. We use some landmark nodes to store the shortest distances of node-to-landmark and landmark-to-landmark at the preprocessing step; subsequently, we decompress the information to reconstruct the node pairwise shortest distance matrix for minibatch nodes at the training phase.

Similar to the core-net approach, we select the landmark nodes as popular nodes that have higher degrees than a degree-threshold. We hypothesize our heuristic based on the intuition that the higher degree nodes are more likely on the shortest path of any two nodes. Consequently, we store the shortest path distances of all nodes to their closest popular nodes in a node-to-popular (N2P) vector and the shortest distances between popular nodes in a popular-to-popular (P2P) matrix. At the training phase, we reconstruct the distance profile of every minibatch node (source) by adding the N2P distance to its popular node, the P2P

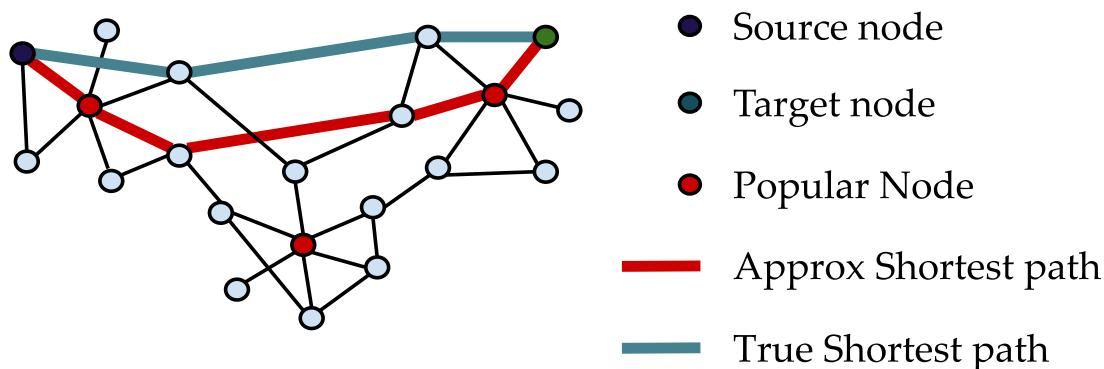


Figure 2.2: DNS-approx heuristic.

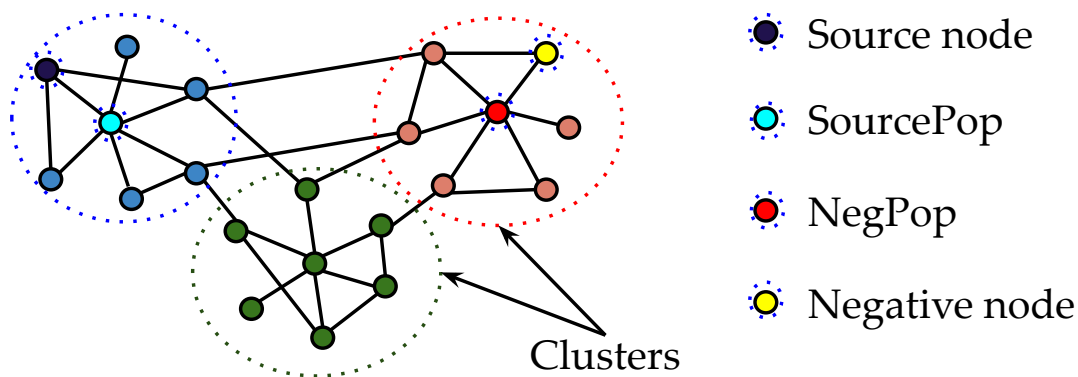


Figure 2.3: DNS-scalable heuristic.

distances across popular nodes, and the N2P distance from popular nodes to target nodes (see Figure 2.2). We denote this heuristic as DNS-approx as it is approximating the true shortest path between node-pairs with paths going through popular nodes.

DNS-approx requires $\mathcal{O}(p^2 + n)$ space, where p is the number of popular nodes (in general, we select 10% nodes as popular nodes), and n is the number of nodes in the graph. The degree-threshold, which decides the popular nodes, controls this space requirement of this heuristic as a higher degree threshold selects a lower number of popular nodes and vice versa. Although DNS-approx can reduce the space complexity by a large margin, we still need to reconstruct the approximate distance profile of a minibatch node with all other nodes, which costs us extra runtime during model training. To address this concern, we need a solution

that does not need to reconstruct the approximate distances of a minibatch node with all other nodes and still perform distance-aware negative sampling.

We propose another scalable DNS approach, DNS-scalable, which only uses the popular-to-popular matrix to sample negative-popular (NegPop) nodes where the probability of selecting a negative-popular node of another popular node is linearly proportional to its distance from the popular node of the source node (SourePop) node. Every NegPop node corresponds to a cluster of negative nodes based on the N2P vector mapping. Finally, to sample negative nodes, we uniformly sample a node from the cluster of the selected NegPop node (see Figure 2.3).

2.5 Evaluation Setup

Benchmark datasets: In our experiments, we use four benchmark datasets for node classification: CiteSeer, Cora, PubMed, and PPI [144, 153] where CiteSeer, Cora, and PubMed are citation-networks and PPI is protein-protein interaction network. In the citation network, the nodes correspond to articles of different subjects, whereas the edges correspond to citations between those articles; consequently, the node prediction task on this network is to predict the article subject. Meanwhile, the physical interaction between different proteins with their defined roles (cellular functions) on a specific human tissue ¹ is represented using a protein-protein interaction network where the classification task is to predict the protein roles.

Since our goal is to find meaningful node-embeddings of a graph that only reflect the graph structure information rather than the node feature information, we do not use any node

¹Instead of working on multiple graphs, we randomly select one PPI network corresponding to a specific human tissue.

features of these benchmark datasets for our experiments. These datasets have multiple small disconnected components with the largest connected component that describes the graph structure properly. Consequently, our proposed DNS sampler requires a definite distance between any node pairs; therefore, we focus our experiments on the largest connected component of these networks.

Table 2.1 summarizes all the datasets with significant statistics where we see CiteSeer, Cora, and PubMed are sparse datasets with average degrees from 3.47 to 4.5, whereas PPI is quite dense with average degree 27.97.

Synthetic datasets. We further analyze the representation quality of DNS-based GRL models with varying graph density using three synthetic datasets- sparse, moderate, and dense. We provide in detail description of synthetic data generation in the supplementary material.

Baseline models: For baseline models, we choose DeepWalk [102], node2vec [38], LINE [123], GAE [56] and VGAE [56] (details in Section 2.2). We implement the Unigram Negative Sampler and the Distance-aware Negative Sampler on DeepWalk and node2vec models. For the DeepWalk model, we also implement the UNS-deg formulation described in 2.2. Further on the DeepWalk model, we also implement the DNS-approx and the DNS-scalable method. Note that our DNS sampler does not depend on the DeepWalk method, and we can pair DNS with any other GRL approach like GraphSAGE and GCN. We describe the hyperparameter settings and the node classification setup in details in the supplementary materials.

Evaluation Metrics: We use F1-Macro to report the classification accuracy on node classification tasks. Moreover, we visualize the node representations using standard visualization tools like t-SNE, which is a dimensionality reduction technique that preserves local

Table 2.2: The summary of the model performances in terms of downstream node classification F1-macro score. We highlight the best score for each dataset. For Cora, CiteSeer, PPI, and PubMed, we choose context window 4 to report the results. Both DW-DNS-approx and DW-DNS-scalable use 10% nodes as popular nodes. We run each model 5 times and report the performances in terms of mean and standard deviations.

| Models | Dataset | | | |
|------------------------|--------------------|--------------------|--------------------|--------------------|
| | CiteSeer | Cora | PubMed | PPI |
| GAE | 0.40 ± 0.01 | 0.61 ± 0.02 | 0.59 ± 0.02 | 0.68 ± 0.00 |
| VGAE | 0.39 ± 0.02 | 0.58 ± 0.02 | 0.60 ± 0.02 | 0.67 ± 0.00 |
| LINE | 0.37 ± 0.05 | 0.52 ± 0.05 | 0.47 ± 0.07 | 0.68 ± 0.00 |
| node2vec-UNS | 0.43 ± 0.02 | 0.54 ± 0.01 | 0.56 ± 0.01 | 0.63 ± 0.00 |
| node2vec-DNS | 0.52 ± 0.01 | 0.62 ± 0.00 | 0.58 ± 0.01 | 0.64 ± 0.00 |
| DeepWalk-UNS | 0.51 ± 0.01 | 0.67 ± 0.00 | 0.58 ± 0.00 | 0.69 ± 0.00 |
| DeepWalk-UNS-deg | 0.47 ± 0.00 | 0.65 ± 0.01 | 0.54 ± 0.00 | 0.68 ± 0.00 |
| DeepWalk-DNS | 0.61 ± 0.01 | 0.72 ± 0.01 | 0.63 ± 0.00 | 0.69 ± 0.00 |
| DW-DNS-approx | 0.59 ± 0.01 | 0.71 ± 0.01 | 0.64 ± 0.00 | 0.69 ± 0.00 |
| DW-DNS-scalable | 0.57 ± 0.01 | 0.70 ± 0.01 | 0.63 ± 0.01 | 0.68 ± 0.00 |

similarities.

2.6 Results

Table 2.2 compares the performance of our proposed DNS-based GRL models (DeepWalk-DNS and node2vec-DNS) and DNS-based scalable approaches (DW-DNS-approx and DW-DNS-scalable) with other baseline models on the benchmark node classification tasks. From the results, DNS-based models show a significant improvement in the F1-Macro score than that of the traditional sampling-based models across all benchmark datasets. Further, our scalable DNS models show comparable performance as the basic DNS models. The t-SNE plot (Figure 2.4) shows that the DNS-based model learns more meaningful feature visualizations with better cohesion and separation of the classes (shown using colors) than that of the other models for CiteSeer dataset.

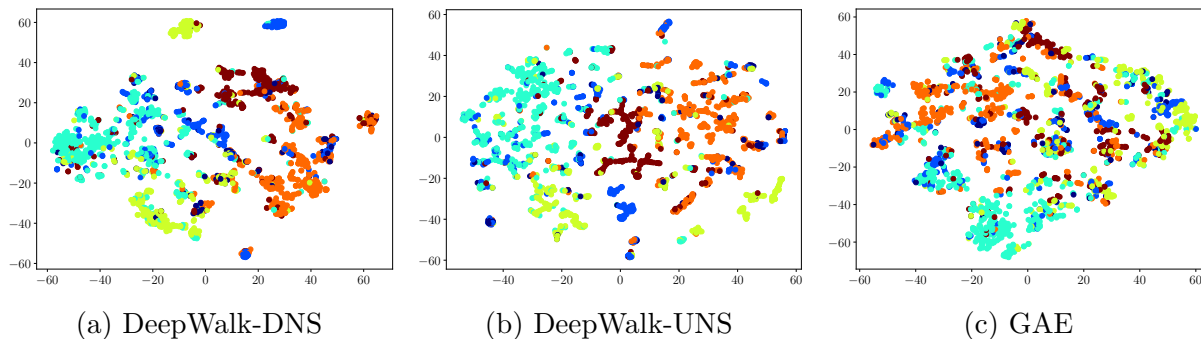


Figure 2.4: t-SNE plot for embeddings generated by DeepWalk with Distance-aware Negative Sampler model (DeepWalk-DNS), DeepWalk with Unigram Negative Sampler model (DeepWalk-UNS), and Graph Auto Encoder model (GAE) on CiteSeer dataset.

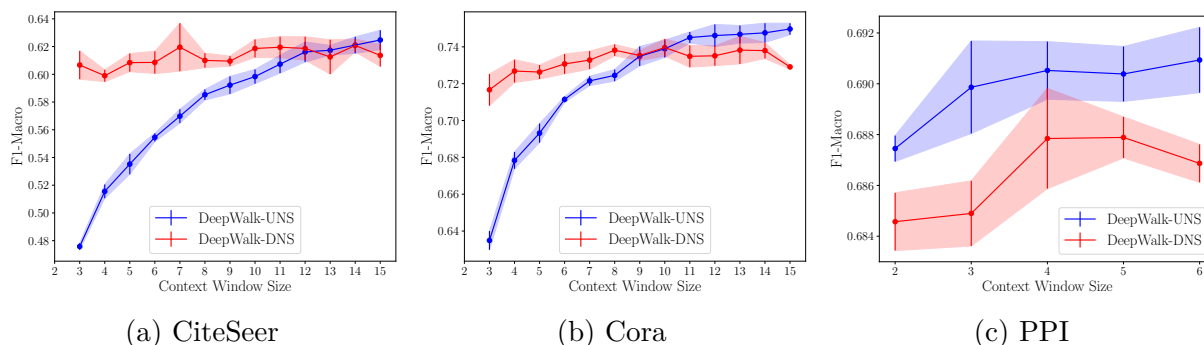


Figure 2.5: Node classification performance (measured by F1-Macro score) plot with varying context window on CiteSeer, Cora, and PPI dataset. DeepWalk with Distance-aware Negative Sampler (DeepWalk-DNS) and with Unigram Negative Sampler (DeepWalk-UNS) are the competing models.

To measure the impact of varying context windows, Figure 2.5 shows F1 score of DeepWalk-DNS and DeepWalk-UNS with varying context windows on CiteSeer, Cora, and PPI datasets, where the performance of DNS-based methods tends to get closer to UNS-based methods with increasing context window, which is in-line with our discussion in Section 4. However, we can see that the F1-score of DNS is significantly larger than that of UNS for a large range of context windows smaller than a reasonable value. In practice, we prefer low context windows during negative sampling for better optimization time at learning phase [38]. Moreover, for dense graphs, such as PPI, dissimilar nodes have low pairwise distances that weaken our node-similarity assumption and decrease DNS-based model performance. However, we can set the

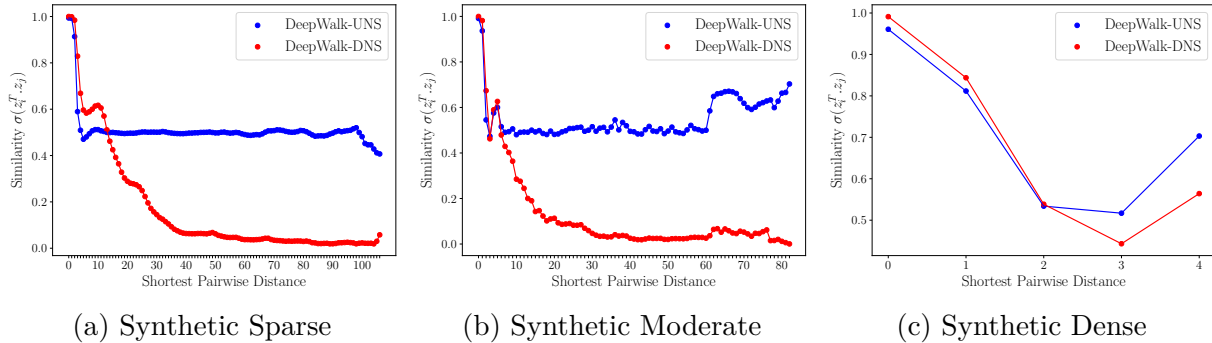


Figure 2.6: Average Pairwise Similarity of all node-pairs in embedding space where similarity $= \sigma(z_i^T z_j)$ for z_i and z_j node embeddings. Embeddings generated by DNS based GRL model show minimum similarity for distant nodes with the similarity decreasing with increasing distance d .

value of γ to a small value in γ -linear negative sampling, which reduces the effect of distances and improves performance for the densely connected graphs. We demonstrate the sensitivity of the performance of γ -DNS to γ in Appendix A. We also perform ablation studies to understand the importance of distance-aware negative sampling probabilities in comparison to baseline methods in Appendix A. Figure 2.6 shows the similarity of the embeddings generated by DNS-based models on the synthetic graphs, which is inversely proportional to the pairwise distance that maximizes the separation of distant node pairs. A detailed analysis of DeepWalk-DNS on the synthetic dataset and the effect of γ in γ -linear negative sampling is in Appendix A.

Table 2.3 shows the comparison of DNS-approx and DNS-scalable with the DNS approach in terms of the model size, the per epoch training time, and the downstream node classification accuracy on PubMed dataset. The space complexity of both heuristics is $\mathcal{O}(p^2 + n)$. From the table, we see that both heuristics take much less space than the basic DNS approach, even when we consider 50% of nodes as popular nodes. We save all the trained models and the size of the saved model objects is used as a measure of space complexity. We see the model sizes are in a similar range for both heuristics when they have an equal number of popular nodes.

Table 2.3: Comparison of scalable heuristics on PubMed dataset. We denote the model size as *size*, training time per epoch as *time* with second as unit, and F1-Macro score of the downstream node classification task as *acc*.

| | <i>size</i> | <i>time</i> | <i>acc</i> |
|-----------------------------|-------------|-------------|------------|
| DNS-approx ($p = 0.1n$) | 43.7 MB | 1216 | 0.64 |
| DNS-approx ($p = 0.3n$) | 256.4 MB | 1357 | 0.63 |
| DNS-approx ($p = 0.5n$) | 871.1 MB | 1556 | 0.63 |
| DNS-scalable ($p = 0.1n$) | 66 MB | 108 | 0.63 |
| DNS-scalable ($p = 0.3n$) | 239.7 MB | 281 | 0.64 |
| DNS-scalable ($p = 0.5n$) | 639.9 MB | 655 | 0.64 |
| Basic DNS | 1.5 GB | 90 | 0.63 |

However, we see a large reduction in the training time for the DNS-scalable method since it does not reconstruct the shortest distance matrix, like DNS-approx. The training time for the DNS-scalable approach increases with the number of popular nodes because we select negative-popular nodes from the popular matrix, and the number of selection operations increases with the popular matrix size. All the models perform similarly on the downstream node classification task, which shows that we can get similar performance from our DNS-scalable method with a much smaller space complexity (with a small number of popular nodes) and a comparable training time complexity.

Chapter 3

Weak Supervision for Semantic Segmentation

3.1 Overview

This chapter provides an overview of how aggregated saliency maps are more effective than class activation maps for weakly supervised semantic segmentation. It also addresses the limitations of popular approaches and suggests possible solutions.

The goal of weakly supervised semantic segmentation (WS3) is to train segmentation models with coarse-scale supervision and without using pixel-level annotations. In recent years, several WS3 methods have been proposed that use image-level class labels to generate pseudo-ground truths for training segmentation models. Many of these methods employ localization methods such as Class Activation Maps (CAMs) [16, 112, 151], generated from a pre-trained classifier, to guide the segmentation process.

CAMs are **activation maps** generated by the last convolutional neural network (ConvNet) layer of the classification model, which is integrated with the class-specific weights of the final fully-connected layer to produce a score for every pixel. While Class Activation Maps (CAM) are good at highlighting discriminative regions (DRs) of an image (i.e., regions that contribute significantly to the classifier’s decision), CAMs are also known to ignore regions

of the target object class that do not contribute to the classifier’s prediction, termed non-discriminative regions (NDRs). In particular, it has been shown that the activation maps in the final convolution layer only contain information relevant for classification, a phenomenon called *information bottleneck* [63]. As a result, CAMs are biased towards mostly finding DR while missing the NDR of the target object, which is equally important for the purpose of segmentation. A number of WS3 solutions thus require further processing of the CAM outputs to recover NDR for high segmentation accuracy [10, 44, 58, 63, 64, 68].

In contrast to activation maps, **attribution maps** provide an alternative approach for assigning a score to every pixel based on its contribution to the final neural network prediction. The most commonly used attribution map is the gradient-based Saliency Maps [116]. The basic idea of saliency is to calculate the gradient of the target class score with respect to every pixel in the input image. Attribution maps are fundamentally distinct from activation maps obtained from the last layer of ConvNet models. However, despite the frequent use of attribution maps for neural network interpretability, their use in WS3 as an alternative to CAMs has largely been unexplored.

With the advancement of vision transformers achieving state-of-the-art (SOTA) performance on many computer vision tasks [41], extending CAMs to work with non-ConvNet-based classifiers is a non-trivial exercise. In contrast, gradient-based Saliency maps can be applied to any classifier with differentiable layers, rendering them as a universal solution for WS3 tasks. Moreover, Saliency maps inherently provide a solution to the deficiencies of CAM-based approaches as explored in this work. Although the limitations of CAMs have been well-known in the WS3 research community and all SOTA methods in WS3 provide solutions to mitigate the deficiencies of CAMs, they lack in providing deeper insights on how saliencies can be used as an alternative to CAM for WS3.

Our goal in this work is to provide a comprehensive study of the comparison between CAMs

and Saliencies for WS3. It is important to mention that our goal is not to achieve SOTA performance for WS3, but rather to provide novel insights into the potential of saliencies and their variations in addressing the limitations of CAMs. Our contributions are outlined below:

- We offer multiple perspectives to understand the similarities and differences between CAMs and Saliencies. Section 3.3 delves into these perspectives, serving as a “bridge” in the analysis of CAMs and saliencies.
- We provide new evaluation metrics to measure WS3 performance, which are specifically designed to complement existing metrics such as mIoU in quantifying the deficiencies of CAMs and evaluating the effectiveness of alternate techniques with respect to CAMs. The proposed evaluation metrics are detailed in Section 3.4.
- We demonstrate the effectiveness of saliencies in addressing the limitation of CAM through our empirical studies on the PASCAL VOC, COCO, and MNIST datasets, as detailed in Section 3.5.
- We identify the limitations of saliency maps for WS3 and propose different variations of stochastic aggregation methods to fix these limitations. Specifically, we propose a random cropping approach for stochastic aggregation that disintegrates the spatial structure of input images as compared to injecting spatially invariant noise. While random cropping is a common data augmentation technique, its application as a stochastic aggregation method in this work is novel. Additional insights regarding stochastic aggregation of saliencies are presented in Sections 3.6 and 3.7.

3.2 Fundamental Concepts and Definitions

3.2.1 Class Activation Maps

The Class Activation Maps (CAMs) are based on convolutional neural networks with a global average pooling (GAP) layer applied before the final layer. Formally, let the classifier be parameterized by $\theta = \{\theta_f, \mathbf{w}\}$, where $f(\cdot; \theta_f)$ is the feature extractor network prior to the GAP layer and \mathbf{w} is the set of weights of the final classification layer. The CAM of the c -th class for an image \mathbf{I} can be obtained as follows:

$$\text{CAM}_c(\mathbf{I}; \theta) = \frac{\mathbf{w}_c^T \mathbf{A}}{\max \mathbf{w}_c^T \mathbf{A}} \quad (3.1)$$

where $\mathbf{A} = f(\mathbf{I}; \theta_f)$ is the activation map, $\mathbf{w}_c \in \boldsymbol{\theta}$ is c -th class weight, and $\max(\cdot)$ is the maximum value over all pixels in \mathbf{I} for normalization.

Limitations of CAMs

CAMs produce coarse-scale localizations of objects because the activation maps of the final convolutional layer have significantly lower resolution compared to the input image. Additionally, the final activation maps show high values for only a subset of regions of the target object that are discriminative for the classification task, while disregarding regions that do not impact the accuracy of classification. Thus, CAMs in their raw form without supplementary post-processing, are unsuitable for training segmentation models.

Discriminative and Non-Discriminative Regions

Discriminative regions (DRs) are those regions of the ground-truth object that are crucial for the classification model to predict the class label of the image accurately. In contrast, *non-discriminative regions (NDRs)* are those regions of the ground-truth that are still important for segmenting the object but do not significantly impact the model’s accuracy upon removal. We formally define DR and NDR based on the CAM outputs as follows:

Definition 3.1 (DR and NDR). The discriminative region (DR) and non-discriminative region (NDR) for the c -th class of an image \mathbf{I} can be defined for every pixel (i, j) belonging to the c -th class ground-truth segmentation \mathcal{S}_{GT}^c as follows:

$$\text{DR}_c(i, j) = \mathbb{I}(\text{CAM}_c(i, j) \geq \tau_{cam}) \quad (3.2)$$

$$\text{NDR}_c(i, j) = \mathbb{I}(\text{CAM}_c(i, j) < \tau_{cam}) \quad (3.3)$$

where τ_{cam} represents a threshold applied to the CAM to obtain the segmentation of the object class and $\mathbb{I}(\cdot)$ is the indicator function. While the optimal threshold may differ for each image, we adopted the common practice of using a global threshold ($\tau_{cam} = 0.25$) for defining DR and NDR throughout this paper. Note that DRs and NDRs are a partitioning of the ground-truth mask \mathcal{S}_{GT}^c based on CAM scores.

3.2.2 Saliency Maps

Saliency maps are attribution maps that assign a score to every image pixel representing its contribution to the final classifier prediction. They are frequently employed as a tool to enhance model interpretability. Formally, the saliency map (SM) of the c -th class for image \mathbf{I} can be defined as:

$$\text{SM}_c(\mathbf{I}, \theta) = \left| \frac{\partial S_c}{\partial \mathbf{I}} \right| = \left| \mathbf{w}_c^T \frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \right| \quad (3.4)$$

where $S_c = \mathbf{w}_c^T \text{GAP}(\mathbf{A}) + b_c$ is the score for the c -th class, and $b_c \in \mathbf{theta}$ is the bias term. For a multi-channel image, saliency maps are computed by taking a maximum of the gradient values across the channels.

Definition 3.2 (HSR and LSR). The high saliency region (HSR) and low saliency region (LSR) for the c -th class of an image \mathbf{I} can be defined for every pixel (i, j) belonging to the c -th class ground-truth segmentation \mathcal{S}_{GT}^c using a threshold τ_{sm} specific to saliency maps as follows:

$$\text{HSR}_c(i, j) = \mathbb{I}(\text{SM}_c(i, j) \geq \tau_{sm}) \quad (3.5)$$

$$\text{LSR}_c(i, j) = \mathbb{I}(\text{SM}_c(i, j) < \tau_{sm}) \quad (3.6)$$

Just like DRs and NDRs, the HSRs and LSRs are an alternate partitioning of \mathcal{S}_{GT}^c based on SM score.

3.3 Comparing CAMs and Saliency Maps

3.3.1 A Visual Comparison Using Hyperplanes

While CAMs and saliency maps differ in many respects, they also exhibit several similarities. We offer a novel viewpoint of comparing CAMs and SMs from the lens of CAM and SM hyperplanes. First, we define two k -dimensional Hilbert spaces (where k is the number of

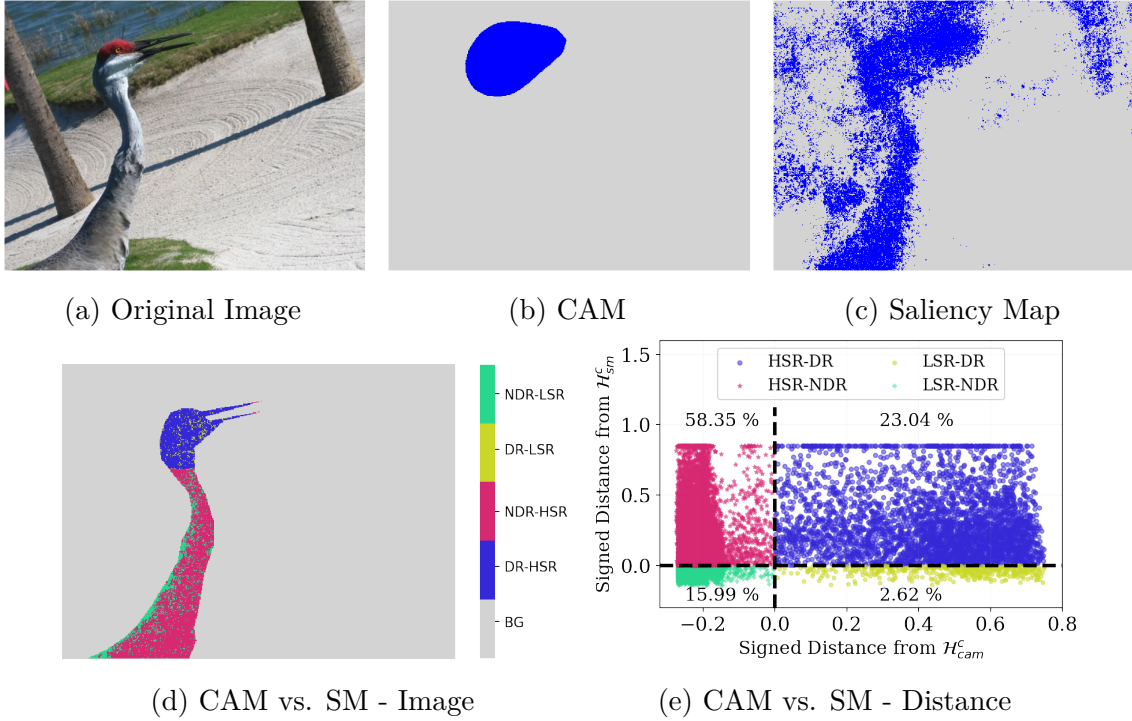


Figure 3.1: A visual comparison of CAMs and saliency maps (SMs) for a representative image from the VOC12 dataset.

channels in the activation map): \mathcal{A} for the activations of images and \mathcal{A}' for the gradients of the GAP layer w.r.t. the image. Formally, for an arbitrary image \mathbf{I} , let the activation at any pixel $\mathbf{A}_{(i,j)} \in \mathcal{A}$, and the gradient of the GAP layer $\frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}}|_{(i,j)} \in \mathcal{A}'$.

Definition 3.3 (c -th class CAM hyperplane). For every image \mathbf{I} , let \mathcal{H}_{cam}^c be the following hyperplane in \mathcal{A} :

$$\mathcal{H}_{cam}^c : \frac{\mathbf{w}_c^T}{Z} \mathbf{a} - \tau_{cam} = 0 \quad (3.7)$$

where τ_{cam} is the CAM threshold, $\mathbf{w}_c \in \mathbf{w}$ is the weight for the c -th class, and $Z = \max \mathbf{w}_c^T \mathbf{A}$ is a normalization factor depending on \mathbf{I} . Note that Z changes for every image and is equivalent to having a variable intercept term for the CAM hyperplane but with a fixed slope \mathbf{w}_c for every image.

Remark 3.4. If a point $\mathbf{a} \in \mathcal{A}$ corresponding to a ground-truth pixel lies above \mathcal{H}_{cam}^c , i.e.,

$\mathbf{w}_c^T \mathbf{a} / Z - \tau_{cam} \geq 0$, then the pixel belongs to DR; otherwise, it belongs to NDR.

See Appendix for proof. This remark states that any arbitrary pixel $(i, j) \in \mathcal{S}_{GT}^c$ will belong to the DR or NDR depending on which side of the CAM hyperplane it lies. In other words, as long as \mathbf{w}_c and τ_{cam} are fixed, the DR and NDR of the c -th class for any image \mathbf{I} are separated by its CAM hyperplane \mathcal{H}_{cam}^c .

Definition 3.5 (c -th class SM parallel-hyperplane). Let \mathcal{H}_{sm}^c be the following set of two parallel hyperplanes in \mathcal{A}' :

$$\mathcal{H}_{sm}^c : |\mathbf{w}_c^T \mathbf{a}'| - \tau_{sm} = 0 \quad (3.8)$$

where τ_{sm} is the saliency map threshold and $\mathbf{a}' \in \mathcal{A}'$ is the gradient of the GAP layer w.r.t. image at any arbitrary pixel.

Remark 3.6. If a point \mathbf{a}' corresponding to a ground-truth pixel lies on the outer sides of \mathcal{H}_{sm}^c , i.e., $|\mathbf{w}_c^T \mathbf{a}'| - \tau_{sm} \geq 0$, then the point belongs to HSR; otherwise, it belongs to LSR.

See appendix for proof. Similar to the DR/NDR for CAMs, the HSR/LSR are separated by SM parallel-hyperplanes. Furthermore, the slope of both CAM and SM hyperplanes are the same: \mathbf{w}_c . However, the important distinction is that for CAMs, the DR/NDR depends on the values of the activation map $\mathbf{A}_{(i,j)}$, while for SMs, the HSR/LSR depends on the gradient $\frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}}|_{(i,j)}$. A ground-truth pixel may thus belong to DR or NDR and HSR or LSR depending on the value of its activations and gradient of GAP layer, respectively.

In Figure 3.1, we visually compare CAMs and SMs for a representative image from the VOC12 dataset. From this comparison, we observe that the CAM (see Figure 3.1b) predominantly highlights the DR of the bird class such as its head — a crucial feature for classification. As a result, NDRs such as the bird’s body are sparingly covered by the CAM. In contrast,

the saliency map (see Figure 3.1c) for the same image covers most regions of the target bird class, albeit with some noisy representation of the background class too. To provide a comprehensive visualization of how HSRs in saliency maps can potentially recover NDRs, we present a scatterplot in Figure 3.1e comparing the signed distances of each pixel $(i, j) \in \mathcal{S}_{GT}^{bird}$ from the CAM and SM hyperplanes, namely, \mathcal{H}_{cam}^{bird} and \mathcal{H}_{sm}^{bird} . Notably, the HSRs successfully recover a substantial portion of DRs, labeled as HSR-DR (blue). A minor segment of the DRs (2.62% of GT) is missed by SMs, termed LSR-DR (yellow). Nonetheless, SMs are proficient in recovering 55.32% of the GT regions originally classified as NDR, labeled as HSR-NDR (maroon). Yet, both SMs and CAMs fall short in capturing the LSR-NDR region, which constitutes 15.99% of the GT (green). The color-coded segmentation map for these four distinct regions are presented in Figure 3.1d, thereby showing the potential of saliency maps in addressing the limitations of CAMs in recovering NDRs.

3.3.2 Perspective from Contribution Windows

Next, we present another novel viewpoint of comparing saliencies and CAMs from the perspective of *contribution windows*—a concept innate to the architecture of convolutional neural networks (ConvNets). Note that the tendency of CAMs to only focus on DRs can be understood using the *information bottleneck* principle proposed in [63]—every layer of a neural network filters or “funnels in” information about inputs and as a result only task-specific information is retained at the outputs. While this information bottleneck exists in the forward propagation of ConvNets, the reverse phenomenon happens during backpropagation when information “funnels out” from the activation maps to the input image. This phenomenon can be described using the contribution window of an input pixel on the activation maps, defined as follows.

Definition 3.7 (Contribution Window). Let’s consider a ConvNet with N layers, where

every layer l performs a 2D convolution using an $F \times F$ kernel denoted as \mathbf{K}_l , to compute activation $\mathbf{A}_l = \text{Conv2D}(\mathbf{A}_{l-1}, \mathbf{K}_l)$. The contribution window at layer l of a pixel in the input image can then be defined as the region in \mathbf{A}_l that affects (or contributes to) the gradients of \mathbf{A}_l w.r.t. the input pixel.

This concept is illustrated in Figure 3.2, where the contribution window is highlighted in yellow at every layer for an example yellow pixel at layer 0. The contribution window can be viewed as the reverse concept of “receptive fields” defined for the forward pass of ConvNets. Indeed, since the gradient of the forward convolution \mathbf{K}_l is also a convolution with a rotated kernel [53], the receptive field of the backward convolution during gradient computation becomes the concept of contribution window. We can show that all activations at layer l in the contribution window of an input pixel can affect its gradient.

Now, let us consider pixels that have 0 activations across all channels in the final layer shown in grey in Figure 3.2. By design, such *non-activated pixels* will register 0 CAM scores. We want to analyze if it is possible for a non-activated pixel (yellow) to show non-zero gradients (and thus saliencies) in the input image. Assuming we use activation functions $f(z)$ that are 0 when $z \leq 0$, we can show that this depends on whether the contribution window of the pixel contains any *activated pixel* with non-zero activations at the final layer, shown in red. In fact, we can show that if the contribution window size of a non-activated pixel is smaller than its distance from an activated pixel, it will have 0 gradients. However, this is practically not likely as the contribution window size generally grows linearly with the depth of ConvNets. An exception is when we use 1×1 kernels. Through empirical evidence provided in section 3.5.1, we can establish that as the contribution window expands (achieved by increasing the $F \times F$ kernel size), saliencies can progressively encompass more NDRs, thus directly addressing the limitations of CAMs.

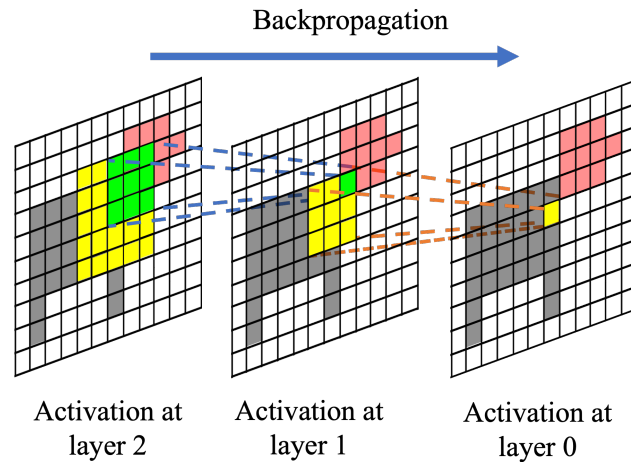


Figure 3.2: A schematic of “contribution window” demonstrating how the gradients at layer $l - 1$ is affected by the gradients from the contribution window of layer l .

3.4 Experimental Setup & Evaluation Metrics

3.4.1 Experimental Setup

Following the common practice in WS3, in this paper, we compared different approaches quantitatively and qualitatively by conducting experiments on MNIST, PASCAL VOC ’12, and MS COCO ’14 datasets. We also utilized two types of classification models based on ResNet50 architecture: i) “model-org”, which is simply fine-tuned on the corresponding dataset, and ii) “model-pert”, which is fine-tuned with additional noise perturbation.

3.4.2 Evaluation Metrics

To assess the quality of the segmentation maps, *mean intersection over union (mIoU)* is a widely used metric in WS3 literature. mIoU measures the ratio of correct prediction (intersection) over the union of predictions and ground truths, averaged across all classes, including background class. Notably, mIoU provides an unbiased estimate of the segmentation per-

formance; however, it fails to provide insights about the coverage of NDRs and DRs. Given the limitation of CAMs not being able to identify NDRs, it becomes crucial to measure how effective alternative WS3 techniques (e.g., saliencies) are at addressing the deficiencies of CAMs. This warrants the need for novel evaluation metrics focusing on the DRs and NDRs.

In this paper, we introduce the following three novel evaluation metrics: NDR-Recall, DR-Recall, and Foreground Precision (FG-Prec). **DR-Recall** is the ratio of correct DR prediction over the ground-truth DR and can be formally defined as:

$$\text{DR-Recall} = |\text{TP}(P, DR_{GT})| / (|\text{TP}(P, DR_{GT})| + |\text{FN}(P, DR_{GT})|),$$

where P denotes the segmentation prediction, DR_{GT} denotes the ground-truth DR area, and $|\text{TP}|$ and $|\text{FN}|$ denote the count of true positives and false negatives over the DR region. As mentioned in Section 3.1, we define ground truth DR (DR_{GT}) and NDR (NDR_{GT}) by employing a global threshold ($\tau_{cam} = 0.25$) on the CAM prediction and then taking its overlap with the ground-truth segmentation mask. In a similar manner, we compute **NDR-Recall** for a given segmentation prediction (P) and the corresponding ground-truth NDR region (NDR_{GT}). Apart from these two metrics, we also compute the **Foreground-Precision** of different target classes as an additional metric, which can be defined as the ratio of correct foreground prediction over the total foreground prediction. Note that our proposed metrics are defined to analyze the deficiencies of CAM and, hence, are biased only if we are evaluating CAMs just by themselves (e.g., CAMs would show low NDR Recall value by definition). However, these metrics are unbiased if the goal is to measure how well alternative WS3 techniques (e.g., saliencies) fix the shortcomings of CAMs.

3.5 Quantitative Comparison: CAM/Saliency

3.5.1 Effect of Contribution Window

To empirically demonstrate the effect of contribution window on the recovery of NDRs, we utilize a 5-layer ConvNet architecture where each layer employs an $F \times F$ kernel, followed by ReLU activation. We apply sufficient zero padding to ensure that the spatial dimension of the activations in each layer is equal to that of the input image. Different models with varying kernel sizes were then trained on the MNIST Segmentation dataset.

The results for CAM and Saliency, in terms of mIoU and NDR-Recall, are presented in Figure 3.3. The $F \times F$ kernel size correlates with the size of the contribution window for the backpropagated gradients. Notably, when the contribution window is 1×1 , the performance of CAMs and Saliencies is quite comparable. However, differences in performance become more prominent (larger red and blue shaded regions) as the contribution window size increases. With an expanding contribution window, saliencies are capable of recovering more pixels that have high gradients and low (≈ 0) activations, effectively capturing a larger proportion of NDR. This, in turn, leads to a gradual increase in NDR-Recall until saturation is achieved. Further discussion of this experiment can be found in the Appendix.

3.5.2 Comparing NDR Recovery

Table 3.1 presents a quantitative evaluation of CAMs and saliencies on the PASCAL VOC dataset using different methods for background resolve (see Appendix for details). We compare the best-segmented map produced by each method by varying the global threshold of τ_{cam} and τ_{sm} from 0.01 to 0.50 and selecting the segmented map with the highest mIoU. The “basic background resolve” row of Table 3.1 shows that saliency map outperforms CAM in

| Method | B/G Resolve | mIoU | FG-Prec | DR-Recall | NDR-Recall |
|----------|-------------|-------------|-------------|-------------|-------------|
| CAM | Basic | 43.7 | 56.1 | 93.8 | 43.7 |
| Saliency | Basic | 37.7 | 45.9 | 75.4 | 55.6 |
| | Smooth | 44.0 | 52.2 | 84.3 | 60.0 |
| | Superpixel | 49.0 | 60.0 | 80.9 | 61.8 |

Table 3.1: Quantitative comparison of CAM and Saliency on VOC dataset in terms of mIoU, Foreground Precision, and DR-/NDR-Recall.

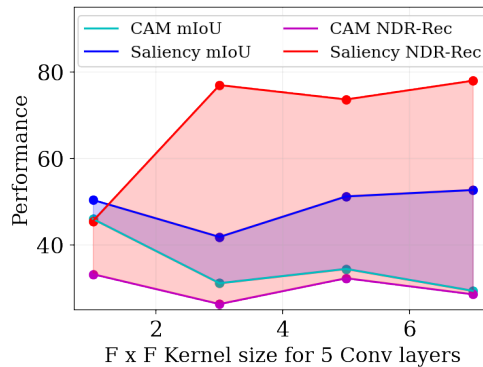


Figure 3.3: Effect of Contribution Window on NDR-Recall and mIoU for MNIST Dataset.

finding non-discriminative regions, as indicated by its higher NDR-Recall score. However, CAM outperforms the saliency maps in terms of mIoU, FG-precision, and DR-Recall, likely due to the noisy and scattered nature of saliency maps. This motivates further exploration of opportunities to improve the quality of saliency maps.

3.5.3 Improving Saliencies with Simple Post-processing

We first explore if simple post-processing methods such as **kernel smoothing background resolve** and **Superpixel-based background resolve** can improve SM performance. Kernel Smoothing smooths the gradients of the saliencies by applying a Gaussian kernel, while superpixel-based smoothing assigns a label to each superpixel, which effectively mitigates the noisiness and scatteredness that may be present in saliency maps. See Appendix for details of these post-processing approaches. Table 3.1 presents their results as ‘Smooth’ and ‘Super-

| Model | Method | BG-Res | mIoU | FG-Prec | DR-Rec | NDR-Rec |
|-------------|-------------|----------|--------------------|--------------------|--------------------|--------------------|
| org | Smooth-Grad | Basic | 38.6 (+0.9) | 47.1 (+1.2) | 82.0 (+6.6) | 51.7 (-3.9) |
| | | Smooth | 37.5 (-6.5) | 47.1 (-5.1) | 79.2 (-5.1) | 48.3 (-11.7) |
| | | Superpix | 41.0 (-8.0) | 52.2 (-7.8) | 77.0 (-3.9) | 52.1 (-9.7) |
| pert-gauss | Smooth-Grad | Basic | 45.3 (+7.6) | 54.9 (+9.0) | 87.4 (+12.0) | 55.9 (+0.3) |
| | | Smooth | 44.8 (+0.8) | 54.1 (+1.9) | 87.5 (+3.2) | 56.8 (-3.2) |
| | | Superpix | 48.1 (-0.9) | 57.4 (-2.6) | 86.4 (+5.5) | 62.9 (+1.1) |
| org | Binary-Mask | Basic | 41.2 (+3.5) | 51.3 (+5.4) | 79.9 (+4.5) | 53.6 (-2.0) |
| | | Smooth | 43.4 (-0.6) | 53.5 (+1.3) | 84.7 (+0.4) | 53.9 (-6.1) |
| | | Superpix | 47.3 (-1.7) | 57.0 (-3.0) | 84.8 (+3.9) | 62.0 (+0.2) |
| pert-binary | Binary-Mask | Basic | 42.4 (+4.7) | 52.9 (+7.0) | 78.7 (+3.3) | 55.8 (+0.2) |
| | | Smooth | 44.9 (+0.9) | 54.8 (+2.6) | 84.8 (+0.5) | 57.2 (-2.8) |
| | | Superpix | 48.9 (-0.1) | 56.8 (-3.2) | 86.2 (+5.3) | 68.0 (+6.2) |

Table 3.2: Quantitative comparison of SmoothGrad and BinaryMask in terms of mIoU, FG-Precision, DR-/ NDR-Recall for different fine-tuned models on VOC dataset. The difference between the aggregated saliency performance and the vanilla saliency performance is shown in parentheses. A positive value denotes an increase in performance; whereas a negative value denotes a decrease in performance for aggregated saliencies.

pixel’ background Resolve. Both approaches outperform basic background resolve results in terms of mIoU, FG-Precision, DR-Recall, and NDR-Recall. Superpixel-based saliency maps demonstrate significant improvement over CAM in terms of mIoU and NDR-Recall; however, CAM outperforms all saliency methods in finding discriminative regions, as indicated by its higher DR-Recall score. It is worth mentioning that superpixel-based background resolve is not scalable for larger datasets. To this end, we need to explore saliencies where the smoothing can be integrated inherently without additional computational overheads.

3.6 Stochastic Aggregation of Saliencies

To reduce the noisiness of saliencies, [117] proposed a stochastic aggregation-based method for saliency maps, named **SmoothGrad**, where Gaussian noise is added to the input image for smoothing saliencies. In this paper, we explored another variation of input noise perturbation, namely **BinaryMask**, where we multiply the image by a binary mask instead of adding Gaussian noise to the input image. The amount of perturbation for SmoothGrad is

controlled by standard deviation of Gaussian noise, whereas for BinaryMask, the probability of each pixel in the mask being 1 controls the perturbation magnitude. See Appendix for additional details on these methods. “*Model-pert-binary*” and “*Model-pert-gaussian*” are the two finetuned classifiers augmented by binary and Gaussian noise, respectively.

3.6.1 Smoothing Saliencies by Injecting Noise

Table 3.2 compares results of saliency with different stochastic aggregation methods like SmoothGrad and BinaryMask. The change in performance from the basic or vanilla saliencies (without stochastic aggregation) is shown in parentheses; a positive percentage denotes improvement and a negative percentage denotes degradation. Saliencies from the classification models perturbed with similar noise (*model-pert-gaussian* for **SmoothGrad** and *model-pert-binary* for **BinaryMask**) perform better than the saliencies generated by the original model. According to [12], adding noise during training is a common regularization technique that results in denoising. The additive effect of adding noise during training and inferring with noise yields the best saliency map.

Although adding noise may make the saliency maps smoother, with increasing noise, the saliency maps may become unstable and the mIoU performance may gradually drop with excessive noise. A detailed analysis of the sensitivity of our experiments to noise is provided in Appendix. Also note that the classification model needs to be fine-tuned with similar noise for these stochastic perturbations techniques to produce smoothed saliencies. This additional fine-tuning could be an expensive process, and further motivates us to explore alternate aggregation methods that do not involve additional fine-tuning steps.

3.7 Stochastic Aggregation Through Cropping

Random cropping is commonly used as a data augmentation technique to increase the variety of training data by cropping random regions of the input image to a specific size. One of the advantages of random cropping is that it generates input samples that follow the input data distribution, since all the crops are basically part of the input image. In this section, we utilize random cropping as a stochastic aggregation technique to improve the performance of saliencies.

3.7.1 Disintegrating the Spatial Structure of Images using Random Cropping

Random cropping can also be viewed as a perturbation technique where the individual crops disintegrate the spatial structure of the input image. We treat random cropping as a spatial perturbation and generate a saliency map by stochastically aggregating the saliency maps of the individual cropped images. We define this spatial perturbation-based aggregation as follows: $\tilde{\text{SM}}_c(\mathbf{I}) = \frac{1}{n} \sum_{i=1}^n w_i \text{SM}_c(\tilde{\mathbf{I}}_i)$, where $\tilde{\mathbf{I}}_i = f_{\text{pert}}(\mathbf{I})$, \mathbf{I} corresponds to the input image, $\tilde{\mathbf{I}}_i$ denotes the individual crops, and $f_{\text{pert}}(\cdot)$ denotes the spatial perturbation function, which is random cropping for this experiment. $\text{SM}_c(\cdot)$ is the (basic) saliency map and $\tilde{\text{SM}}_c$ corresponds to the final aggregated saliency, and w_i denotes the weight of each of the individual crop saliencies. For our experiments, we choose $w_i = \sigma(S_c(\tilde{\mathbf{I}}_i))$, where $S_c(\tilde{\mathbf{I}}_i)$ is the classification score of the crop $\tilde{\mathbf{I}}_i$, and $\sigma(\cdot)$ is the sigmoid activation function.

First row of Table 3.3 shows the performance of random cropping as a stochastic aggregation method, where we can see that it performs better than saliencies in terms of mIoU, FG-Precision, and DR-/NDR- Recall for all the background resolve approaches (difference in

performance of random crop and saliencies are provided in parentheses). We can achieve as high as 50.4 mIoU using random crop-based aggregated saliencies with superpixel-based background resolve. Notably, random cropping-based aggregated saliencies employ the “*Model-org*” classifier to compute the saliencies, showing that random cropping does not require the classifier to be finetuned on additional perturbations to perform well.

| Method | BG-Res | mIoU | FG-Precision | DR-Recall | NDR-Recall |
|--------------|----------|--------------------|--------------------|--------------------|--------------------|
| Random Crop | Basic | 44.6 (+6.9) | 53.6 (+7.7) | 84.2 (+8.8) | 59.4 (+3.8) |
| | Smooth | 46.2 (+2.2) | 56.6 (+4.4) | 84.4 (+0.1) | 57.5 (-2.5) |
| | Superpix | 50.4 (+1.4) | 61.7 (+1.7) | 82.6 (+1.7) | 61.7 (-0.1) |
| Random Patch | Basic | 35.6 (-2.1) | 43.9 (-2.0) | 71.5 (-3.9) | 57.8 (+2.2) |
| | Smooth | 37.7 (-6.3) | 45.4 (-6.8) | 77.6 (-6.7) | 59.9 (-0.1) |
| | Superpix | 39.3 (-9.7) | 47.7 (-12.3) | 76.9 (-4.0) | 61.6 (-0.2) |
| Disc-Patch | Basic | 35.4 (-2.3) | 32.6 (-13.3) | 74.7 (-0.7) | 58.3 (+2.7) |
| | Smooth | 38.6 (-5.4) | 45.8 (-6.4) | 78.8 (-5.5) | 61.7 (+1.7) |
| | Superpix | 40.7 (-8.3) | 51.8 (-8.2) | 72.2 (-8.7) | 57.0 (-4.8) |
| Disc-Crop | Basic | 45.1 (+7.4) | 54.0 (+8.1) | 76.5 (+1.1) | 55.5 (-0.1) |
| | Smooth | 46.3 (+2.3) | 56.5 (+4.3) | 74.7 (-9.6) | 53.4 (-6.6) |
| | Superpix | 50.6 (+1.6) | 61.6 (+1.6) | 73.9 (-7.0) | 57.9 (-3.9) |

Table 3.3: Comparison of Random Crop, Discriminative Crop, Random Patch, and Discriminative Patch in terms of mIoU, FG-Precision, DR-/ NDR-Recall on VOC12. The difference between the aggregated and saliency performance is shown in parenthesis.

3.7.2 Can we do better than random cropping?

Next, we explore different variations of random cropping and patching techniques that break the spatial structure of input images. Random patching is an erasure-based method similar to the idea of the cutout method [24]. The discriminative variations of random cropping (Disc-Crop) and patching (Disc-Patch) take the real values of CAM to complement the probability of selecting a crop or patch. See Appendix for details. Table 3.3 shows the results of these alternate methods. Random cropping and its discriminative variation (Disc-Crop) perform significantly better than the (basic) saliency method. However, the patch-based methods do not show comparative performance in terms of mIoU, FG-Precision, DR-Recall, and NDR-Recall. One possible reason is that we used the original “*Model-org*” classifier, which

is not augmented with the patch-wise perturbations. Therefore, patching creates unnatural artifacts during inference, and the classifier fails to attribute the individual samples correctly. The discriminative versions of cropping and patching did not significantly outperform the random versions.

3.8 Related Works

Current techniques for WS3 utilize CAMs as the foundation to produce segmentation maps. These methods can be broadly categorized into three types: (1) Modifying model architecture, (2) Iterative update-based methods, and (3) Modifying Loss functions.

First, several methods that modify the model architecture for WS3 have been developed to overcome the well-known limitations of CAM [10, 58, 63]. For example, a global weighted rank (GWR) pooling layer was proposed in [58] that neither underestimates the object size like global max pooling (GMP) nor overestimates it using GAP. Normalized global weighted pooling (nGWP) was also proposed in [10] to replace the GAP layer, which helps to recover small segments, thus improving the mask precision. Another method FickleNet [62] introduced stochastic aggregations in feature maps to produce the localization maps. However, changing the architecture can be difficult and restricts the types of models that are compatible with these methods.

The second set of methods aims to improve the seed performance of CAMs through iterative updates, such as erasure-based methods [19, 44, 68, 138] and adversarial optimizations [64, 138]. Specifically, erasure-based methods suggest erasing the most discriminative regions to unveil the non-discriminative regions, thus addressing some of the limitations of CAMs. On the other hand, AdvCAM [64] proposed an anti-adversarial optimization technique to exploit the boundary information with pixel-level affinity for capturing more regions of the

target objects. One primary limitation of such methods is that the termination condition is not well-defined and often heuristically chosen.

Finally, a third set of WS3 methods focus on modifying the loss function to improve the object coverage of CAMs. Specifically, the RIB [63] demonstrates that an information bottleneck occurs in later layers as only the task-relevant information is passed to the output. As a result, CAMs which are computed at the last layer, have sparse coverage of the target object. A new loss function was proposed that encourages the transmission of information from non-discriminative regions for classification, thus improving the quality of localization maps.

Several prior works have utilized saliency maps for WS3, as documented in [58, 113, 121, 149]. These studies primarily concentrate on enhancing segmentation map accuracy through post-processing techniques. However, their focus differs from our work on exploring the inherent potential of saliencies in overcoming the limitations associated with CAM-based approaches. Although these existing works contribute valuably to the field, they do not directly address the specific research questions that our study delves into – specifically, the comprehensive analysis of saliencies’ effectiveness with respect to CAMs.

CAMs and Saliencies have also been extensively examined in the realm of explainability research that is focused on providing explanations of the model outputs, which can potentially satisfy regulatory experiments [37], help practitioners debug their model [13, 15] and identify unintended bias in the model [59, 132]. Approaches based on activation maps fall under the CAM-based methods category [16, 112, 133, 151]. Conversely, techniques relying on attribution maps belong to the saliency-like methods group [114, 116, 117, 118, 122, 148].

| Method | B/G Resolve | mIoU | FG-Prec | DR-Recall | NDR-Recall |
|-------------|-------------|--------------|--------------|--------------|--------------|
| CAM | Basic | 28.82 | 41.16 | 83.59 | 31.46 |
| Saliency | Basic | 22.22 | 28.26 | 65.46 | 48.78 |
| | Smooth | 25.46 | 31.94 | 73.02 | 52.65 |
| Random-Crop | Basic | 21.13 | 27.6 | 62.87 | 46.38 |
| | Smooth | 26.58 | 33.83 | 72.09 | 52.22 |

Table 3.4: Quantitative comparison of CAM and Saliency on COCO dataset in terms of mIoU, Foreground Precision, and DR-/NDR-Recall.

3.9 Discussion and Future Directions

Table 3.4 quantitatively evaluates the performance of competing methods on the MS COCO 2014 dataset. We compare the best-segmented map generated by each method by varying the global threshold across the range of 0.01 to 0.50. The segmented map with the highest mIoU value is selected for comparison. The Table illustrates that both saliency and random crop saliency outperform CAM in terms of NDR-Recall. This signifies that saliency-based approaches exhibit better recovery of the NDR region compared to CAM. However, CAM surpasses saliencies in terms of mIoU, FG-Precision, and DR-Recall. The smooth saliencies show comparable performance to CAM, which indicates the potential for improvement in the performance of saliencies by reducing its noisiness, especially when dealing with challenging datasets like the COCO dataset.

In conclusion, this work proposes three novel evaluation metrics for WS3, namely NDR-Recall, DR-Recall, and FG-Precision, which can be used to assess the performance of alternative WS3 models in fixing the deficiencies of CAMs. We also revisit the potential of the use of saliency maps for WS3, which has been largely overlooked in the past, and demonstrate that simple post-processing steps, stochastic aggregation methods, and random cropping-based aggregation can significantly improve the quality of segmentation masks.

Although our work lays the foundation for future research in saliency maps for WS3, it's

important to clarify that we are not the first to use saliencies for WS3, neither are we claiming state-of-the-art (SOTA) performance using stochastic aggregation methods when applied over saliencies. Instead, our focus is on presenting novel insights into the strengths and weakness of saliencies w.r.t. CAMs from multiple perspectives, and showing how simple modifications to saliencies can effectively address the limitations inherent in CAMs.

As newer techniques based on Vision Transformers [69, 139] and Foundation models such as Segment-Anything [17] are developed in the WS3 community to deliver SOTA performance, we anticipate future research to comprehensively understand their strengths and weaknesses building upon the metrics and analyses presented in our paper. Furthermore, while current post-processing methods in WS3 like CRF, PSA, and IRN are designed specifically to complement the limitations of CAM-based methods, we anticipate that researchers will build upon our findings to develop more advanced post-processing techniques for gradient-based WS3 methods.

Chapter 4

Benchmarking Vision-Language

Models on Scientific Dataset

4.1 Introduction

There is a growing deluge of images that are being collected, stored, and shared in organismal biology—the branch of biology interested in the study of structure, ecology, and evolution of organisms. In particular, images are increasingly becoming the currency for documenting the vast array of biodiverse organisms on our planet, with repositories containing millions of images of biological specimens collected by scientists in field museums or captured by drones, camera traps, or tourists posting photos on social media. This growing wealth of biological images provides a unique opportunity to understand the scientific mechanisms of how organisms evolve and adapt to their environment directly from images. The traditional approach for advancing knowledge in organismal biology is by discovering the observable characteristics of organisms or *traits* (e.g., beak color, stripe pattern, and fin curvature) that serve a variety of biological tasks such as defining groups of organisms, understanding their genetic and developmental underpinnings, and analyzing their interactions with environmental selection pressures [45]. However, the measurement of traits is not straightforward and often relies on expert visual attention involving labor-intensive operations and subjective definitions [115], hindering rapid scientific advancement [77].

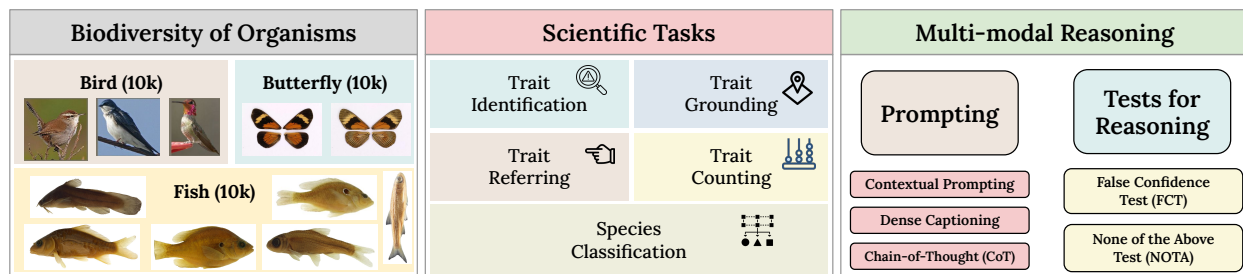


Figure 4.1: Overview of our goals and contributions. We analyze the capabilities of 12 state-of-the-art (SOTA) vision-language models (VLMs) in answering scientific questions using images from three groups of organisms: fishes, birds, and butterflies, over five groups of biologically relevant tasks. We also explore the effectiveness of these models for reasoning using various prompting techniques and tests for reasoning hallucination.

With the recent rise of large foundation models such as vision-language models (VLMs) (e.g., GPT-4, GPT-4V(ision) [97, 98], Gemini [124], LLaMA 3.2 [27, 125], and LLaVA [72]) that can simultaneously solve a diverse range of tasks involving text and images, it is pertinent to ask if pre-trained VLMs contain the necessary *scientific knowledge* to aid biologists in answering a variety of questions pertinent to the discovery of biological traits from images. Note that unlike mainstream tasks in computer vision, understanding scientific images requires knowledge of domain-specific terminologies and reasoning capabilities that are not fully represented in conventional image datasets used for training VLMs. In particular, an important end-goal in scientific applications such as organismal biology is to explain the process of visual reasoning used to arrive at a prediction, often involving the knowledge of biological traits. Hence, to assess the usefulness of VLMs in accelerating discoveries in organismal biology, it is important to test their ability to identify and reason about biological traits automatically from images.

In this work, we assess the zero-shot capabilities of 12 state-of-the-art (SOTA) VLMs, including the proprietary GPT-4V(ision) and the recent GPT-4O(mni) along with other open-source VLMs, on five scientifically relevant tasks in organismal biology, namely species classi-

fication, trait identification, trait grounding, trait referring, and trait counting. These tasks are designed to test different facets of VLM performance in organismal biology, ranging from measuring predictive accuracy to assessing their ability to reason about their predictions using visual cues of known biological traits. For example, the task of species classification tests the ability of VLMs to discriminate between species, while in trait grounding and referring, we specifically test if VLMs are able to localize morphological traits (e.g., the presence of fins or patterns and colors of birds) within the image. To perform this evaluation, we present **VLM4Bio**, a benchmark dataset of $\approx 469K$ question-answer pairs based on $30k$ images of three taxonomic groups of organisms: fishes, birds, and butterflies.

Main Contributions:

1. We present a novel dataset of scientific question-answer pairs to evaluate the effectiveness of VLMs in answering scientific questions across a range of biologically relevant tasks in the field of organismal biology.
2. We present novel benchmarking analyses of the zero-shot effectiveness of pre-trained SOTA VLMs on our dataset, exposing their gaps in advancing scientific knowledge of organismal biology.
3. We present novel comparisons studying the effects of prompting and tests for reasoning hallucination on VLM performance, shedding new light on the reasoning capabilities of SOTA VLMs in organismal biology.

4.2 Related Works

With the rise of SOTA VLMs such as GPT-4V(ision) [98], GPT-4O(mni) [100], and Gemini [124], there has been a simultaneous growth in the number of benchmarking analyses pub-







| Species Classification | Trait Identification | Trait Referring |
|--|--|---|
| <p>Question: What is the scientific name of the butterfly shown in the image?</p>  <p>Correct Answer: Heliconius timareta</p> | <p>Question: Is there eye visible in the fish shown in the image?</p>  <p>Options: A) Yes B) No</p> <p>Correct Answer: A) Yes</p> | <p>Question: What is the trait of the fish that correspond to the bounding box region [2545, 335, 3510, 423] in the image?</p>  <p>Options: A) dorsal fin B) caudal fin C) adipose fin D) pelvic fin</p> <p>Correct Answer: A) dorsal fin</p> |
| Question type: Open Questions | Question type: Multiple Choice Questions | Question type: Multiple Choice Questions |
| Species Classification | Trait Grounding | Trait Counting |
| <p>Question: What is the scientific name of the bird shown in the image?</p>  <p>Options: A) Geothlypis philadelphia B) Vireo atricapilla C) Larus glaucescens D) Coccythraustes vespertinus</p> <p>Correct Answer: C) Larus glaucescens</p> | <p>Question: What is the bounding box coordinates of the dorsal fin in the fish shown in the image?</p>  <p>Options: A) [453, 620, 557, 724] B) [2545, 335, 3510, 423] C) [2012, 1001, 2404, 1350] D) [3444, 350, 4730, 1114]</p> <p>Correct Answer: B) [2545, 335, 3510, 423]</p> | <p>Question: How many unique fins are visible in the fish shown in the image? The fins that are normally present in a fish are dorsal fin, caudal fin, pectoral fin, pelvic fin, anal fin and adipose fin.</p>  <p>Correct Answer: 5</p> |
| Question type: Multiple Choice Questions | Question type: Multiple Choice Questions | Question type: Open Questions |

Figure 4.2: Illustrative examples of VLM4Bio tasks with different question-types.

lished in the last few years to evaluate different facets of VLM performance on a range of mainstream tasks in computer vision. A majority of previous analyses [142, 147] involve evaluations on single tasks like Visual Question Answering (VQA), OK-VQA [80], MSCOCO [71], and GQA [46]. Other datasets such as POPE [70], HaELM [125], LAMM [145], MMBench [75], MM-Vet [146], LVLM-eHub [140], SEED [66], and GAIA [83] have also been developed to evaluate the capabilities of VLMs on complex tasks such as reasoning and ability to handle multimodal data. There are also some recent domain-specific benchmark datasets, such as MathVista [76], which includes a variety of challenging VQA problems in the mathematical domain, MedQA(USMLE) [51] which is a collection of VQA problems from medical exams, and the recent MMMU [147] dataset, which covers expert-level problems from diverse fields such as business, arts, science, health, medicine, and engineering.

VLM4Bio dataset is different from existing benchmarks involving domain-specific datasets because of the following reasons. (1) *Focus on organismal biology*: While previous works

have focused on benchmarking the performance of VLMs on other scientific domains (e.g., Arts and Design, Business, Health, and Medicine in MMMU [147] or Mathematics in MathVista [76]), there exists no previous VQA benchmark dataset in the domain of organismal biology to the best of our knowledge. Our work thus fills a critical gap in evaluating the performance of VLMs in a field of biology that has several societal implications such as monitoring biodiversity and understanding the impact of climate change on species traits and populations. (2) *Breadth of Evaluation Tasks*: While previous works are tailored to one or a few evaluation tasks, we consider a wide range of tasks motivated by the needs of domain scientists in the field of organismal biology. They include predictive tasks such as species classification and trait identification as well as tasks that require visual reasoning including trait grounding and referring. We also provide novel comparisons about the performance of VLMs on both open-ended and multiple-choice question (MCQ) formats and comparisons over predictive as well as visual reasoning tasks, in contrast to prior works.

4.3 VLM4Bio Tasks

Figure 4.2 shows illustrative examples of the five VLM4Bio tasks relevant to biologists that we consider in our study, described in detail in the following.

4.3.1 Species Classification

A common (and often the first) task that a biologist considers when examining an organism specimen is to identify its scientific name (or species class). Hence, we consider asking a VLM to provide the scientific name of the organism shown in a given image. There are two types of questions that we consider for this task. First, we consider *open-ended questions*, where we

do not provide any answer choices (or options) to the VLM in the input prompt. The second type is *multiple-choice (MC) questions*, where we provide four choices of candidate species names for the VLM to choose from (out of which only one is correct while the remaining three are randomly selected from the set of all species classes).

4.3.2 Trait Identification

An important goal in organismal biology is to answer questions regarding the observable characteristics of organisms, also known as traits. We thus consider asking VLMs to identify a particular trait of an organism given its image for two taxonomic groups: fishes and birds. For fishes, we considered 10 binary (presence/absence) traits and generated MC questions for the presence of each trait in an image (with two options: yes or no), whereas for birds, we considered 28 traits covering their color, pattern, and measurements (size and shape of regions) in a multiple-choice format. We provide a detailed list of all fish and bird traits in the Supplementary Section [C.6](#).

4.3.3 Trait Grounding and Referring

To further understand the ability of VLMs to visually explain the reasoning behind their prediction of a trait, it is important to evaluate if a VLM correctly identifies the region in the image containing the trait. For this purpose, we consider two other tasks: trait grounding & trait referring, for the taxonomic groups of fishes and birds. In the first task of trait grounding, we ask the VLM to locate a given trait of an organism on its image (i.e., *text to location*). We consider MC question-format for this task where we provide four options of bounding boxes in the image as candidate answer choices, where one of the bounding boxes correctly contains the trait while the remaining three are randomly sampled from the set of

bounding boxes containing other traits of the organism. In the second task of trait referring, we consider the opposite scenario where we provide a bounding box as input to the VLM and ask it to identify the name of the trait present in the bounding box (i.e., *location to text*). We again provide four answer choices in MC question-format, where only one of the options is correct while the remaining three are randomly sampled from the names of other traits of the organism.

4.3.4 Trait Counting

We simply ask how many traits are present in an image of a fish specimen. This is biologically relevant, for example, to understand the number of fins present in a fish organism. Similar to the species classification task, we have open and MC question-types for this task.

4.4 VLM4Bio Dataset

Data Collection and Preprocessing: We collected images of three taxonomic groups of organisms: fish, birds, and butterflies, each containing around 10K images. Images for fish (**Fish-10K**) were curated from the larger image collection, FishAIR [31], which contains images from the Great Lakes Invasive Network Project (GLIN) [1]. These images originate from various museum collections such as INHS [5], FMNH [4], OSUM [78], JFBM [2], UMMZ [3] and UWZM [6]. We created the Fish-10K dataset by randomly sampling 10K images and preprocessing the images to crop and remove the background. For consistency, we leverage GroundingDINO [74] to crop the fish body from the background and Segment Anything Model (SAM) [57] to remove the background. We curated the images for butterflies (**Butterfly-10K**) from the Jiggins Heliconius Collection dataset [60], which has images

| Statistics | Fish-10K | Bird-10K | Butterfly-10K | Fish-500 | Bird-500 |
|------------|----------|----------|---------------|----------|----------|
| # Images | 10,347 | 11,092 | 10,013 | 500 | 492 |
| # Species | 495 | 188 | 60 | 60 | 47 |
| # Genera | 178 | 114 | 27 | 18 | 33 |
| # Traits | 10 | 28 | - | 8 | 5 |

Table 4.1: Key statistics of the **VLM4Bio** dataset.

collected from various sources ¹. We carefully sampled 10K images for Butterfly-10K from the entire collection to ensure the images capture unique specimens and represent a diverse set of species by adopting the following two steps. First, we filter out images with more than one image from the same view (i.e., dorsal or ventral). Second, we ensure each species has a minimum of 20 images and no more than 2,000 images. The images for birds (**Bird-10K**) are obtained from the CUB-200-2011 [131] dataset by taking 190 species for which the common name to scientific name mapping is available. This results in a fairly balanced dataset with around 11K images in total. Additional details on dataset preprocessing are provided in the Supplementary Section C.1.

Annotation: The scientific names for the images of Fish-10K and Butterfly-10K were obtained directly from their respective sources. For Bird-10K, we obtained the scientific names from the iNatLoc500 [21] dataset. We curated around 31K question-answer pairs in both open and multiple-choice (MC) question formats for evaluating species classification tasks. The species-level trait presence/absence matrix for Fish-10K was manually curated with the help of biological experts co-authored in this paper. We leveraged the Phenoscape knowledge [28] base with manual annotations to procure the presence-absence trait matrix. For Bird-10K, we obtained the trait matrix from the attribute annotations provided along with CUB-200-2011. We constructed approximately 380K question-answer pairs for trait identi-

¹Sources: [48, 49, 50, 81, 82, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 103, 107, 108, 109, 110, 135, 136, 137]

fication tasks. For grounding and referring VQA tasks, the ground truths were manually annotated with the help of expert biologists on our team. We manually annotated bounding boxes corresponding to the traits of 500 fish specimens and 500 bird specimens, which are subsets of the larger Fish-10K and Bird-10K datasets, respectively. In particular, we considered 8 fish traits and 5 bird traits for annotating their bounding boxes, resulting in a total of $26K$ question-answer pairs. We also used the Fish-500 dataset for the task of trait counting, resulting in a total of $1K$ question-answer pairs. Across all tasks, our dataset comprises approximately $469K$ question-answer pairs for $30K$ biological images (see Table 4.1). Additional details on data distribution and key statistics are provided in the Supplementary Section C.5.

Dataset Card: We provide the dataset card with a detailed description of the meta-data, data instances, annotation, and license information here (<https://huggingface.co/datasets/imageomics/VLM4Bio#dataset-card-for-vlm4bio>).

VLM Baselines: We consider the following VLM baselines: GPT-4V(ision) [99]², LLaVA-v1.5 (7B/13B) [73], COG-VLM [134], MiniGPT-4 (Vicuna 7B/13B) [152], BLIP-FLAN-T5-XL/XXL [67], and INSTRUCT-BLIP (Vicuna 7B/13B) [23]. We used the latest checkpoints for each model available to date. We used the same question prompt for all models to ensure consistent comparison of results for a variety of open and multiple-choice (MC) questions across the five scientific tasks of our dataset. All the experiments were conducted using NVIDIA A100 GPUs. See supplementary Section C.8 for more details of the VLM baselines.

Evaluation Metrics: We used micro-averaged accuracy as our evaluation metric for all experiments. We designed a systematic rule-based evaluation pipeline to evaluate VLM responses against the ground truths. For each question category, we provide the accuracy percentage of random choice as a basic baseline, where each possible answer is considered

²We use *gpt-4-1106-vision-preview* model as GPT-4V in our experiments.

| Dataset | Question type | Models | | | | | | | | | | | | | Random Choice |
|-------------------------------|---------------|---------------|----------------------|-----------------------|--------------------|---------------------|----------------------|--------------------------|---------------------------|--------------------------|---------------------------|--------------------------|---------------------------|-------|---------------|
| | | <i>gpt-4v</i> | <i>llava v1.5-7b</i> | <i>llava v1.5-13b</i> | <i>cogvlm chat</i> | <i>BLIP flan-xl</i> | <i>BLIP flan-xxl</i> | <i>minigt4 vicuna-7B</i> | <i>minigt4 vicuna-13B</i> | <i>instruct flant5xl</i> | <i>instruct flant5xxl</i> | <i>instruct vicuna7B</i> | <i>instruct vicuna13B</i> | | |
| Species Classification | | | | | | | | | | | | | | | |
| Fish-10k | Open | 1.01 | 2.32 | 0.40 | 0.11 | 0.01 | 1.59 | 0.50 | 0.38 | 0.00 | 1.46 | 0.00 | 0.00 | 0.20 | |
| | MC | 35.91 | 40.20 | 32.27 | 31.72 | 29.76 | 33.36 | 29.02 | 27.45 | 30.86 | 31.70 | 27.27 | 26.57 | 25.00 | |
| Bird-10k | Open | 17.40 | 1.45 | 2.06 | 0.86 | 0.00 | 0.57 | 2.80 | 2.56 | 0.00 | 0.50 | 0.07 | 0.00 | 0.53 | |
| | MC | 82.58 | 50.32 | 55.36 | 44.73 | 33.68 | 34.75 | 23.95 | 27.62 | 36.36 | 35.83 | 44.00 | 46.55 | 25.00 | |
| Butterfly-10k | Open | 0.04 | 0.05 | 0.00 | 0.01 | 0.00 | 0.00 | 0.07 | 0.01 | 0.00 | 0.00 | 9.94 | 0.00 | 1.54 | |
| | MC | 28.91 | 50.24 | 44.58 | 36.45 | 25.14 | 28.88 | 33.06 | 28.90 | 25.28 | 36.67 | 41.70 | 34.48 | 25.00 | |
| Trait Identification | | | | | | | | | | | | | | | |
| Fish-10k | MC | 82.18 | 56.84 | 45.15 | 46.92 | 68.36 | 39.33 | 55.08 | 51.87 | 64.34 | 39.26 | 81.95 | 20.69 | 50.0 | |
| Bird-10k | MC | 62.22 | 34.68 | 46.14 | 63.93 | 50.11 | 41.38 | 39.11 | 40.44 | 47.89 | 45.52 | 77.91 | 89.98 | 31.12 | |
| Trait Grounding | | | | | | | | | | | | | | | |
| Fish-500 | MC | 29.41 | 24.87 | 17.98 | 23.42 | 23.32 | 25.14 | 22.18 | 25.58 | 7.20 | 27.09 | 33.51 | 26.90 | 25.00 | |
| Bird-500 | MC | 8.1 | 26.92 | 35.36 | 23.2 | 11.83 | 10.52 | 15.39 | 24.22 | 3.48 | 0.81 | 30.24 | 13.91 | 25.00 | |
| Trait Referring | | | | | | | | | | | | | | | |
| Fish-500 | MC | 28.15 | 27.07 | 29.14 | 28.19 | 24.93 | 25.68 | 39.24 | 31.21 | 31.75 | 25.78 | 28.04 | 32.73 | 25.00 | |
| Bird-500 | MC | 42.28 | 30.5 | 29.64 | 18.45 | 35.16 | 40.59 | 26.04 | 35.88 | 27.52 | 41.69 | 23.03 | 22.69 | 25.00 | |
| Trait Counting | | | | | | | | | | | | | | | |
| Fish-500 | Open | 16.4 | 47.4 | 52.0 | 14.8 | 37.6 | 63.4 | 13.6 | 31.53 | 50.2 | 61.4 | 61.4 | 0.0 | 25.00 | |
| | MC | 44.80 | 13.20 | 54.80 | 21.00 | 64.8 | 78.2 | 22.00 | 25.00 | 74.0 | 69.4 | 15.80 | 11.80 | 25.00 | |
| <i>Overall</i> | | 34.24 | 29.0 | 31.78 | 25.27 | 28.91 | 30.24 | 23.0 | 25.19 | 28.49 | 29.79 | 33.92 | 23.31 | 22.03 | |

Table 4.2: Zero-shot accuracy comparison of VLM baselines (in % ranging from 0 to 100) for the five scientific tasks. Results are color-coded as **Best**, **Second best**, **Worst**, **Second worst**.

equally likely (yielding an accuracy of 25% for MC questions with four choices).

4.5 Results

Table 4.2 compares the accuracies of VLMs in percentages (ranging from 0 to 100) across the five tasks and over multiple organism datasets. We make the following observations from this result.

All VLMs show poor accuracy on open questions but perform better on MC questions. The zero-shot species classification accuracy of all VLMs on open-ended questions is notably weaker than MC questions. Even the best-performing models, LLaVA-13B, GPT-4V, and Instruct-Vicuna-7B, only achieve accuracies of 2.32%, 17.46%, and 3.62%, respectively, across the three organism datasets. This indicates a significant gap in the ability of existing VLMs to capture the scientific knowledge necessary to differentiate between species (often requiring subtle or nuanced features) without being provided with candidate answer choices. Open-ended species classification is particularly hard for pre-trained VLMs that are not typically trained to provide scientific names of organisms (e.g., *Lepomis cyanelus*) rather than providing their common names (e.g., *green sunfish*). However, the inclusion of candidate answers (or options) in the question prompt serves as a helpful clue to VLMs for narrowing down the solution space and finding the correct answer potentially using elimination strategies. While VLMs are able to utilize these additional hints and work their way through to the correct answer in MC questions, note that open questions are practically more relevant to scientists operating in real-world settings.

Bird dataset shows better accuracy than Fish or Butterfly datasets. Most VLMs show significantly better performance on the Bird-10K dataset in comparison to the Fish-10K and Butterfly-10K datasets. For example, the highest accuracy across all VLMs on the Bird-10K dataset is 82.58%, while it is 40.20% and 50.24% on the Fish-10K and Butterfly-10K datasets, respectively. A potential reason is that while the bird dataset is a subset of the CUB dataset [130] that is commonly used in machine learning literature and has images with natural in-the-wild backgrounds, the butterfly and fish datasets contain images of specimens preserved in museum collections with artificial backgrounds and with imaging artifacts that are not typical for large-scale computer vision datasets. We hypothesize that many of the pre-trained VLM baselines may have seen images similar to those in the Bird dataset during

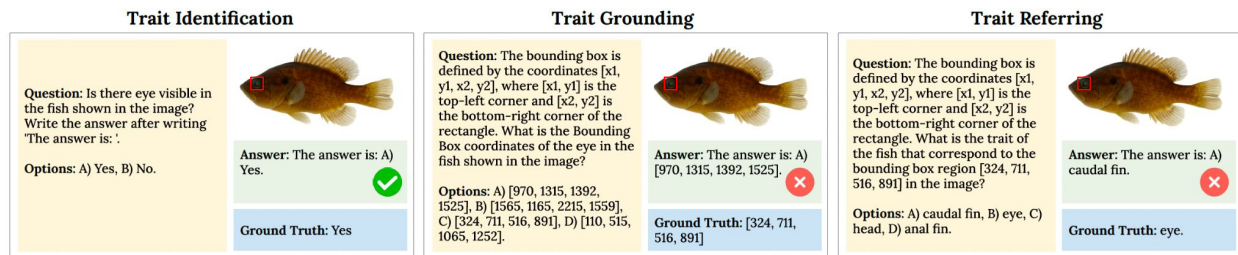


Figure 4.3: Examples of correct and incorrect predictions of GPT-4V for trait identification, trait grounding, and trait-referring tasks related to the “eye”. For visualization assistance, a red-colored bounding box is added around the “eye” in the image.

training, leading to their better performance.

Can VLMs effectively identify biological traits? The performance of most VLMs in trait identification appears significantly better than their performance in species classification, with GPT-4V reaching 82.18% accuracy on the Fish-10K dataset and Instruct-Vicuna-13B achieving 89.98% on Bird-10K. However, some traits such as “eye”, “head”, and “mouth” are almost always present in every organism image, so simply answering “yes, the trait is present” can lead to high accuracy in trait identification. In contrast to the fish dataset, the bird dataset poses more intricate questions regarding a variety of multi-class traits that require a nuanced understanding of colors, patterns, and physical trait dimensions, such as the color of the bill, wing patterns, and tail shapes.

VLMs struggle in localizing traits in images. While most VLMs perform well on the task of Trait Identification, it is crucial to determine if they are focusing on the correct image regions to answer trait-related questions. We thus analyze the performance of VLMs on the tasks of trait grounding (i.e., *text to location*) and trait referring (i.e., *location to text*). We can see that there is a significant drop in the accuracy of trait grounding and referring tasks compared to the trait identification task. This shows that while VLMs can potentially leverage knowledge of trait choices to identify traits, they struggle in localizing the traits in the image and thus visually ground their reasoning. Figure 4.3 shows an illustrative example

of GPT-4V prediction where it predicts the presence of the trait “eye” correctly but fails to localize it in grounding and referring tasks.

Counting biological traits is difficult for VLMs. Recent studies [47, 104, 143] have explored the gap in the ability of VLMs to count objects, which is aligned with our results in Table 4.2. All VLMs, except for BLIP-flan-T5-XXL, show lower performance in counting traits, despite performing well on the trait identification task. The overall average accuracy for the VLMs is displayed in the last block, with GPT-4V(ision) exhibiting the best performance.

We further analyze the errors of different VLMs to better understand their behavior. We find that GPT-4V shows a reduced rate of incorrect responses but a higher incidence of “Other” responses, which include apologetic expressions, admissions of inability to precisely visualize the organism, and disclaimers regarding lack of expert guidance (see Supplementary Section C.10 for more details).

4.5.1 Analyzing the Role of Answer Choices in MC Questions on VLM Performance

Table 4.2 showed that VLMs perform drastically better on MC questions compared to Open questions for species classification. A potential hypothesis for this observation is that VLMs are able to avoid incorrect answer choices (or options) that are too different from the correct option and thus are easy to eliminate. To test this hypothesis, we create three variants of the MC questions for species classification—easy, medium, and hard—where species choices in each variant have varying degrees of similarity determined by their taxonomic groupings. In particular, note that the scientific name of an organism contains taxonomic information at

| Dataset | Difficulty | Models | | | | | | | | | | | | | | CLIP | BioCLIP |
|-----------|------------|---------------|---------------|----------------------|-----------------------|--------------------|---------------------|----------------------|---------------------------|----------------------------|-------------------------|--------------------------|--------------------------|---------------------------|-------|-------|---------|
| | | <i>gpt-4v</i> | <i>gpt-4o</i> | <i>llava v1.5-7b</i> | <i>llava v1.5-13b</i> | <i>cogvlm chat</i> | <i>BLIP flan-xl</i> | <i>BLIP flan-xxl</i> | <i>minigpt4 vicuna-7B</i> | <i>minigpt4 vicuna-13B</i> | <i>instruct flan5xl</i> | <i>instruct flan5xxl</i> | <i>instruct vicuna7B</i> | <i>instruct vicuna13B</i> | | | |
| Fish | Easy | 44.50 | 37.50 | 47.50 | 46.00 | 24.00 | 34.00 | 27.50 | 29.00 | 19.50 | 32.00 | 28.00 | 33.50 | 33.50 | 36.50 | 55.50 | |
| | Medium | 3.50 | 5.50 | 30.00 | 28.50 | 27.00 | 26.00 | 23.00 | 26.50 | 25.00 | 28.50 | 24.50 | 26.00 | 25.50 | 26.00 | 29.00 | |
| Bird | Easy | 73.50 | 68.00 | 53.50 | 50.00 | 38.50 | 34.50 | 36.00 | 21.00 | 32.00 | 41.00 | 33.00 | 43.50 | 39.00 | 57.00 | 94.00 | |
| | Medium | 41.00 | 40.50 | 30.50 | 37.00 | 30.00 | 25.50 | 21.00 | 21.00 | 24.00 | 27.00 | 27.00 | 24.50 | 26.50 | 31.00 | 95.00 | |
| Butterfly | Easy | 18.50 | 17.50 | 19.00 | 20.50 | 24.50 | 30.00 | 25.00 | 34.50 | 26.00 | 24.50 | 22.50 | 19.00 | 24.50 | 21.50 | 65.50 | |
| | Medium | 5.50 | 7.00 | 29.50 | 29.00 | 29.50 | 20.00 | 25.50 | 33.00 | 25.00 | 27.50 | 25.00 | 25.00 | 25.00 | 21.50 | 58.00 | |
| | Hard | 2.00 | 1.50 | 22.00 | 21.00 | 32.00 | 26.50 | 20.00 | 29.50 | 24.00 | 22.50 | 24.00 | 24.00 | 21.00 | 21.50 | 35.00 | |

Table 4.3: Zero-Shot accuracy comparison for *easy*, *medium*, and *hard* datasets. Results are color-coded as **Best**, **Second best**, **Worst**, **Second worst**.

three levels: <genus name> <species name> <subspecies name>³. Since organisms that share taxonomic information have similar appearances, it is hard to differentiate species choices if they are from the same taxonomic group. On the other hand, it is easier to work with species choices from different taxonomic groups. Hence, for the easy set, we selected 50 species from different genera, ensuring that all species choices appear quite different from each other. For the medium set, we increased the complexity by constructing species choices from the same genus but from 10 different species. The hard set presented the highest difficulty level for the butterfly dataset, with the answer choices being from the same genus and species but from 10 subspecies. Each difficulty level consists of 200 images from each set of organisms.

Table 4.3 shows the accuracies of the baseline VLMs for the easy, medium, and hard organism datasets. The pretrained VLMs generally perform best on the easy set and worst on the hard set for each organism. Moreover, there is a gradual improvement in the VLM performance from hard to easy questions. This suggests that the difficulty level of candidate answers (or options) in the question prompt significantly impacts VLMs’ performance. Additionally, this outcome indicates that even SOTA VLMs have limitations in handling fine-grained

³We only have subspecies level information for the Butterfly-10K dataset.

queries. Table 4.3 shows that GPT-4V and OpenAI’s recent release GPT-4o do not perform well when tested on the medium and hard datasets for Fish and Butterfly. Due to this, we further analyze the errors of different VLMs to better understand their behavior. We provide the report in the Supplementary Section C.10.

4.5.2 Comparing Pre-trained VLMs with a Biologically Fine-tuned Model

We compare BioCLIP [119], a state-of-the-art foundation model for species classification fine-tuned with biological images and taxonomic names (TreeOfLife-10M dataset), with the pre-trained VLMs. We observe that BioCLIP significantly outperforms large pretrained VLMs on the Bird-10k and Butterfly datasets, suggesting that BioCLIP has been trained on images that are similar to the organisms present in these datasets. By comparing BioCLIP with CLIP, we can also see that fine-tuning foundation models with biological data provides large gains in classification performance. This suggests that the performance of SOTA VLMs can be further improved by fine-tuning on VLM4Bio Dataset. Further details comparing BioCLIP with SOTA VLMs are provided in the Supplementary Section C.11.

4.5.3 Analyzing Effects of Prompting on VLM Performance

We considered three prompting techniques: Contextual Prompting, Dense Caption Prompting, and zero-shot Chain of Thought Prompting. For **Contextual prompting**, we provided a single-line description (context) of the tasks (e.g., we add “*Each biological species has a unique scientific name composed of two parts: the first for the genus and the second for the species within that genus.*” before the species classification question to give some additional context on the task). **Dense Caption prompting** involves two stages: (1) first, we

| Dataset | Prompting | Models | | | | | | |
|----------------------------|---------------|---------------|---------------|--------------------------------|---------------------------------|------------------------------|-------------------------------|--------------------------------|
| | | <i>gpt-4v</i> | <i>gpt-4o</i> | <i>llava</i> <i>v1.5-7b</i> | <i>llava</i> <i>v1.5-13b</i> | <i>cogvlm</i> <i>chat</i> | <i>BLIP</i> <i>flan-xl</i> | <i>BLIP</i> <i>flan-xxl</i> |
| Fish-Prompting | No Prompting | 34.40 | 79.00 | 41.60 | 35.40 | 31.00 | 28.60 | 22.60 |
| | Contextual | 30.00 | 77.20 | 40.20 | 35.60 | 25.60 | 27.20 | 26.60 |
| | Dense Caption | 18.80 | 78.60 | 26.00 | 27.60 | 32.00 | 28.40 | 29.80 |
| | CoT | 42.60 | 86.00 | 41.40 | 34.80 | 26.80 | 29.20 | 24.60 |
| Bird-Prompting | No Prompting | 78.80 | 97.60 | 44.20 | 49.80 | 45.40 | 35.60 | 35.80 |
| | Contextual | 78.60 | 98.60 | 44.00 | 52.00 | 49.40 | 35.60 | 30.40 |
| | Dense Caption | 87.40 | 97.00 | 33.40 | 41.00 | 44.00 | 25.60 | 22.80 |
| | CoT | 62.60 | 98.60 | 37.40 | 47.80 | 42.20 | 30.60 | 31.00 |
| Butterfly-Prompting | No Prompting | 13.20 | 56.40 | 27.20 | 26.80 | 25.60 | 24.40 | 21.20 |
| | Contextual | 9.20 | 56.20 | 26.00 | 24.60 | 27.20 | 23.60 | 24.60 |
| | Dense Caption | 49.60 | 63.20 | 25.20 | 23.80 | 27.00 | 23.20 | 23.20 |
| | CoT | 63.60 | 74.60 | 21.40 | 23.20 | 34.60 | 37.20 | 23.60 |

Table 4.4: Zero-shot accuracy comparison for different prompting techniques of seven VLMs (in % ranging from 0 to 100). Results are color-coded as **Best** and **Worst**.

prompt the VLM to generate a dense caption of the specimen image such that the caption contains all the necessary trait information of the specimen. (2) We add the dense caption before the question and prompt “*Use the above dense caption and the image to answer the following question.*” to generate responses from the VLM. Similarly, the **Zero-Shot Chain-of-Thought (CoT)** happens in two stages: (1) first, we prompt the VLM to generate the reasoning for a given VQA and multiple choices (options). Zero-shot CoT appends “*Let’s think step by step.*” after the question and options to generate the reasoning. (2) We then add the reasoning after the VQA and prompt “*Please consider the following reasoning to formulate your answer*” to generate the VLM response. We curated a prompting dataset of 500 multiple-choice (MC) VQAs for each set of organisms, which is a subset of the VLM4Bio dataset for species classification.

Table 4.4 compares best-performing VLMs on the prompting dataset. The CoT rows of the table demonstrate that only GPT-4V and GPT-4o have reasoning capabilities that can significantly improve their response to biological questions, while smaller models like LLaVa

| Dataset | Metrics | Models | | | | | | |
|--------------------------------------|-----------------|---------------|---------------|----------------------|-----------------------|--------------------|---------------------|----------------------|
| | | <i>gpt-4v</i> | <i>gpt-4o</i> | <i>llava v1.5-7b</i> | <i>llava v1.5-13b</i> | <i>cogvlm chat</i> | <i>BLIP flan-xl</i> | <i>BLIP flan-xxl</i> |
| False Confidence Test (FCT) | | | | | | | | |
| Fish-Prompting | Accuracy | 34.20 | 73.60 | 25.00 | 28.60 | 24.60 | 0.00 | 7.00 |
| | Agreement Score | 4.40 | 16.60 | 99.80 | 19.20 | 74.40 | 0.00 | 28.4 |
| Bird-Prompting | Accuracy | 73.40 | 99.00 | 25.40 | 35.80 | 19.80 | 0.00 | 20.20 |
| | Agreement Score | 11.40 | 21.00 | 93.20 | 17.80 | 47.80 | 0.00 | 79.80 |
| Butterfly-Prompting | Accuracy | 5.20 | 53.40 | 27.20 | 26.60 | 6.20 | 0.00 | 5.00 |
| | Agreement Score | 2.60 | 12.40 | 95.40 | 5.60 | 13.80 | 0.00 | 19.00 |
| None of the Above (NOTA) Test | | | | | | | | |
| Fish-Prompting | Accuracy | 81.40 | 44.80 | 3.40 | 3.80 | 0.00 | 4.00 | 0.00 |
| Bird-Prompting | Accuracy | 75.00 | 91.40 | 1.00 | 1.20 | 0.00 | 31.40 | 0.00 |
| Butterfly-Prompting | Accuracy | 50.40 | 4.60 | 1.00 | 4.60 | 0.00 | 51.00 | 0.00 |

Table 4.5: Performance of seven VLMs on the NOTA and FCT reasoning tests. Results are color-coded as **Best** and **Worst**.

and BLIP do not show much improvement. Furthermore, providing extra context and caption is more useful for GPT-4V and GPT-4o than the smaller models. This resonates with the findings from [150] that the reasoning abilities of VLMs only emerge after a certain model size. The success of Dense Caption Prompting and CoT Prompting depends on how well they generate the dense caption or the reasoning in the first stage. We report example prompts with VLM responses as case studies in the Supplementary Section C.13.

4.5.4 Analyzing Tests for Reasoning Hallucination

To further understand whether pretrained VLMs can respond with logically coherent and factually accurate reasoning, we evaluate VLMs on two sets of reasoning for hallucination tests - **False Confidence Test (FCT)** and **None of the Above (NOTA) Test** - inspired by [126]. For the FCT, we randomly select an option from the list of given choices and prompt it to the VLM as a “suggested correct answer” along with the question and options.

To evaluate VLMs on FCT, we use Accuracy as well as the Agreement score, which is the percentage of times the VLM agrees with the suggested answer, irrespective of whether that is right or wrong. A high agreement score with a low overall accuracy indicates poor performance as it suggests that the model is simply following the suggestion either because of a lack of knowledge or low confidence in its own response. On the other hand, in the NOTA Test, we replace the correct option with “None of the Above”, requiring the model to produce “None of the above” for all the questions. From Table 4.5, we can see that LLaVa-v1.5-7B shows poor accuracy on both tests and a high agreement score on FCT. Out of all the VLMs, GPT-4V and GPT-4o demonstrate the highest accuracy, i.e., the lowest false confidence. More details on the prompts and examples of the responses have been provided in the Supplementary Section C.13.

4.6 Limitations

Our work has three main limitations. First, while no prior VQA benchmark dataset exists for organismal biology to the best of our knowledge, we focused on only three organisms—fish, bird, and butterfly—out of the many available due to resource constraints. Adding more organisms with manually annotated trait data will require additional resources and domain expertise, which could be pursued in future work. Second, since it is not feasible to manually inspect all images to ensure that they are free from label noise, we acknowledge that some noise may be present in the labels used for evaluating models on our current dataset, which we plan to address in future iterations. Third, due to resource constraints, certain proprietary VLMs that require purchasing APIs like Gemini-Pro [124], Gemini-Ultra [124], and Claude Opus [9] were also not included in the evaluation. We anticipate that their performance will be comparable to that of the proprietary GPT-4V [98] and GPT-4o [100]

considered in our evaluation.

Chapter 5

Conclusion and Future Directions

5.1 Summary of the Thesis

This thesis advances the paradigm of indirect supervision for deep learning models on multi-modal data. It presents three learning techniques of indirect supervision for graphs, images, and text modalities and suggests possible future research directions.

First, chapter 2 proposes Distance-aware Negative Sampling (DNS) for unsupervised Graph Representation Learning (GRL) where the node representations reflect graph structure better than the existing GRL methods by maximizing cohesion and separation on different sizes of networks. With theoretical analysis on cohesion and separation and empirical results on the connected components of benchmark datasets, we present the DNS sampler as an effective sampler that better optimizes the negative sampling objective on unsupervised GRL.

Second, chapter 3 presents novel insights into the strengths and weaknesses of saliencies with respect to CAMs from multiple perspectives and shows how simple modifications, such as stochastic aggregation and random cropping-based aggregation to saliencies, can effectively address the limitations inherent in CAMs and significantly improve the quality of segmentation masks. It also proposes three novel evaluation metrics for WS3, namely NDR-Recall, DR-Recall, and FG-Precision, which can be used to assess the performance of alternative WS3 models in fixing the deficiencies of CAMs.

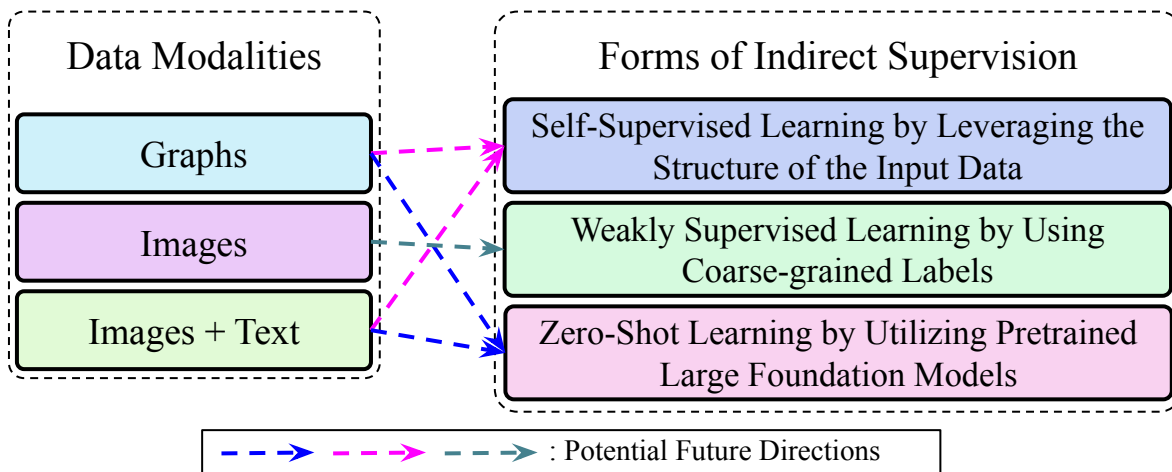


Figure 5.1: Schematic Representation of the Potential Future Research Directions in Indirect Supervision for Multimodal Data.

Finally, chapter 4 introduces VLM4Bio, a benchmark dataset to evaluate the zero-shot performance of pretrained VLMs on biologically relevant questions involving biodiversity images, exposing gaps in SOTA VLMs when applied to organismal biology. We observe that while VLMs are able to perform reasonably well on simpler tasks, e.g., using questions with multiple-choice formats and images with natural-looking backgrounds, they struggle in complex task settings that are practically more relevant to biologists. Through our study on prompting and reasoning tests on the VLM4Bio dataset, we observe that very large SOTA VLMs such as GPT-4V and GPT-4o have reasoning capabilities that can significantly improve the response to biological questions.

5.2 Future Directions

This thesis lays the groundwork for future research in learning through indirect supervision for multimodal data, offering valuable insights and methodologies. Figure 5.1 illustrates potential avenues for advancing indirect supervision techniques in downstream applications

involving multimodal datasets.

One promising direction is the development of foundation models capable of processing and understanding diverse modalities, such as images, text, and structured data, to address complex downstream tasks in scientific domains. In particular, problems in organismal biology require models that can capture the intricate relationships among different modalities while leveraging domain-specific structured knowledge.

Incorporating graph-structured knowledge, such as ontologies, phylogenies, and taxonomies, into multimodal models presents a significant opportunity. One approach (shown in **magenta arrows** in Figure 5.1) involves embedding structured knowledge during the pretraining phase, similar to CLIP supervision, as demonstrated in BioCLIP [119]. Careful alignment of different modalities during the pretraining phase is essential to ensure they collaborate effectively without interference.

Alternatively, structured knowledge can be integrated during the adaptation phase following pretraining, requiring only a few labeled samples to regularize the knowledge of graph structures into text-image pretrained foundational models (shown in **blue arrows** in Figure 5.1). Retrieval-Augmented Generation (RAG) techniques [65], combined with carefully curated downstream datasets, could facilitate this integration by enriching text-image pretrained models with graph-structured insights.

Another potential area for future work is applying the Weakly Supervised Semantic Segmentation (WS3) technique to fine-grained trait discovery using only coarse labels (shown in **green arrows** in Figure 5.1). Recent studies, such as HCompNet [79] and PhyloNN [29], have begun exploring this direction by utilizing class-label information with graph-structured phylogenetic data to detect and localize different prototypes in organism images. This work can be extended by incorporating textual descriptions of the prototypes, thereby enhancing the

interpretability and explainability of prototypical networks across both visual and language domains.

Appendices

Appendix A

Chapter 2

A.1 Proofs of Theorems

THEOREM 4.1 *Unigram Negative Sampling (UNS) Algorithm optimizes the generalized GRL objective with the following specifications: $SIM(i, j) = \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$, $DISSIM(i, j) = \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))$, $\alpha_d = \pi_d(C, \mathbf{A})$, where $\pi_d(C, \mathbf{A})$ is the probability of sampling a node-pair at distance d using a C -length random walk on the graph with adjacency matrix \mathbf{A} , and $\beta_d = KC/n$. As a result, the Separation Power of UNS algorithm is equal to 1.*

Proof. To understand the relationship between the generalized GRL objective and UNS, let us look at the generic objective function of negative sampling (that applies for both UNS and DNS):

$$E(\mathcal{Z}) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}(i)} [\log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + K \sum_{k \in \mathcal{V}} P_{neg}(k|i) \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_k))]$$

where $\mathcal{N}(i)$ represents the set of nodes (of size C) that belong to the neighborhood of node i which we construct using random walk strategy. If we denote the probability of sampling a node j in a C -length random walk from i as $P_{walk}(j|i)$, then $P_{walk}(j|i) = \pi_d(C, \mathbf{A})$, where d is the distance between nodes i and j . The objective function of generic negative sampling

can thus be written as:

$$\begin{aligned}
E(\mathcal{Z}) &= \sum_{i \in \mathcal{V}} \sum_{d=1}^{d_{max}} \sum_{j \in \mathcal{V} \wedge d(i,j)=d} \pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}(i)} K \sum_{k \in \mathcal{V}} P_{neg}(k|i) \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_k)) \\
&= \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad KC \sum_{i \in \mathcal{V}} \sum_{k \in \mathcal{V}} P_{neg}(k|i) \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_k)) \\
&= \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad KC \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} P_{neg}(k|i) \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_k))
\end{aligned} \tag{A.1}$$

In UNS, $P_{neg}(j|i) = \frac{1}{n}$ where n is the number of nodes. Hence, the objective function for UNS becomes:

$$\begin{aligned}
E(\mathcal{Z}) &= \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad \frac{KC}{n} \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j)) \\
&= \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} [\pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \frac{KC}{n} \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))] \\
&= \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} [\alpha_d \text{SIM}(i, j) + \beta_d \text{DISSIM}(i, j)]
\end{aligned}$$

where, $\text{SIM}(i, j) = \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$, $\text{DISSIM}(i, j) = \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))$, $\alpha_d = \pi_d(C, \mathbf{A})$, and $\beta_d = KC/n$. Clearly, the above equation corresponds to the generalized GRL objective.

The Separation Power of UNS is equal to $\frac{\beta_{d_{max}}}{\beta_1} = \frac{KC/n}{KC/n} = 1$. \square

THEOREM 4.2 *Distance-aware Negative Sampling (DNS) Algorithm optimizes the generalized GRL objective with the following specifications: $SIM(i, j) = \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$, $DISSIM(i, j) = \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))$, $\alpha_d = \pi_d(C, \mathbf{A})$, where $\pi_d(C, \mathbf{A})$ is the probability of sampling a node-pair at distance d using a C -length random walk on the graph with adjacency matrix \mathbf{A} , and $\beta_d = KCd/\mathcal{D}(\mathbf{A})$. As a result, the Separation Power of DNS algorithm is equal to d_{max} .*

Proof. For Distance-aware Negative Sampler (DNS), the negative sampling probability $P_{neg}(j|i)$ is linearly proportional to the pair-wise distance $d(j, i)$; $P_{neg}(j|i) = \frac{d(j, i)}{\mathcal{D}(i, \mathbf{A})} = \frac{d}{\mathcal{D}(i, \mathbf{A})}$. We approximate the expected value of $\mathcal{D}(i, \mathbf{A})$ as $\mathcal{D}(\mathbf{A})$. The objective function for DNS can thus be obtained by substituting the value of $P_{neg}(j|i)$ in Equation A.1 as follows:

$$\begin{aligned}
E(\mathcal{Z}) &= \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} KC \frac{d}{\mathcal{D}(\mathbf{A})} \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j)) \\
&= \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} [\pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad \frac{KCd}{\mathcal{D}(\mathbf{A})} \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))] \\
&= \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} [\alpha_d SIM(i, j) + \beta_d DISSIM(i, j)]
\end{aligned} \tag{A.2}$$

where, $SIM(i, j) = \log(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$, $DISSIM(i, j) = \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j))$, $\alpha_d = \pi_d(C, \mathbf{A})$, and $\beta_d = KCd/\mathcal{D}(\mathbf{A})$. Clearly, Equation A.2 corresponds to the generalized GRL objective.

The Separation Power of DNS is equal to $\frac{\beta_{d_{max}}}{\beta_1} = \frac{KCd_{max}/\mathcal{D}(\mathbf{A})}{KC/\mathcal{D}(\mathbf{A})} = d_{max}$. \square

THEOREM 4.3 *Let the average pairwise similarity for any two nodes at distance d be*

given by $\xi_d = \frac{1}{|\mathcal{S}_d|} \text{SIM}(i, j) = \frac{1}{|\mathcal{S}_d|} \sum_{(i, j) \in \mathcal{S}_d} \sigma(z_i^T z_j)$. We can then show that DNS generates embeddings such that ξ_d is a function of d and for $d > C$, ξ_d is inversely proportional to d .

Proof. Let us assume that the DNS based GRL model has reached its global maximum with loss \mathcal{Q} . From Equation A.2, the loss of DNS based GRL model is given by,

$$\begin{aligned}
\mathcal{Q} &= - \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) - \\
& KC \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} \frac{d}{\mathcal{D}(\mathbf{A})} \log(\sigma(-\mathbf{z}_i^T \mathbf{z}_j)) \\
&= - \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) \log \sigma(\mathbf{z}_i^T \mathbf{z}_j) - \\
& KC \sum_{d=1}^{d_{max}} \sum_{(i, j) \in \mathcal{S}_d} \frac{d}{\mathcal{D}(\mathbf{A})} \log(1 - \sigma(\mathbf{z}_i^T \mathbf{z}_j))
\end{aligned} \tag{A.3}$$

Since the model has reached its global optimum, the similarity $\sigma(\mathbf{z}_i^T \mathbf{z}_j)$ for nearby node-pairs ($d(i, j) < C$) should be high such that $1 - \sigma(\mathbf{z}_i^T \mathbf{z}_j)$ low. We approximate $\log(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) \approx \sigma(\mathbf{z}_i^T \mathbf{z}_j) - 1$ as the remainder term in its Taylor's series expansion is close to zero. Moreover, we expand $\log(1 - \sigma(\mathbf{z}_i^T \mathbf{z}_j)) = -\sigma(\mathbf{z}_i^T \mathbf{z}_j) - \frac{\sigma(\mathbf{z}_i^T \mathbf{z}_j)^2}{2} - \frac{\sigma(\mathbf{z}_i^T \mathbf{z}_j)^3}{3} - \dots = -\sigma(\mathbf{z}_i^T \mathbf{z}_j) - \mathcal{R}(\sigma(\mathbf{z}_i^T \mathbf{z}_j))$.

Note that the length of the random walk is at most C . Consequently, for $d > C$, $\pi_d(C, \mathbf{A}) = 0$.

We can thus rearrange Equation A.3 as,

$$\begin{aligned}
\mathcal{Q} &= - \sum_{d=1}^C \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) (\sigma(\mathbf{z}_i^T \mathbf{z}_j) - 1) \\
&\quad - KC \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \frac{d}{\mathcal{D}(\mathbf{A})} (-\sigma(\mathbf{z}_i^T \mathbf{z}_j) - \mathcal{R}(\sigma(\mathbf{z}_i^T \mathbf{z}_j))) \\
&= - \sum_{d=1}^C \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) (\sigma(\mathbf{z}_i^T \mathbf{z}_j) - 1) \\
&\quad + KC \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \left(\frac{d}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \frac{d}{\mathcal{D}(\mathbf{A})} \mathcal{R}(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) \right)
\end{aligned}$$

We approximate $\frac{d}{\mathcal{D}(\mathbf{A})} \mathcal{R}(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) \approx 0$ because $R(\sigma(\mathbf{z}_i^T \mathbf{z}_j)) \approx 0$ for distant pairs and $\frac{d}{\mathcal{D}(\mathbf{A})} \approx 0$ for nearby pairs at optimum. Hence, we obtain:

$$\begin{aligned}
\mathcal{Q} &= - \sum_{d=1}^C \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \tag{A.4} \\
&\quad \sum_{d=1}^C \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) + KC \sum_{d=1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \frac{d}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) \\
&= \sum_{d=1}^C \sum_{(i,j) \in \mathcal{S}_d} \pi_d(C, \mathbf{A}) - \sum_{d=1}^C \sum_{(i,j) \in \mathcal{S}_d} \left(\pi_d(C, \mathbf{A}) - \frac{KCd}{\mathcal{D}(\mathbf{A})} \right) \\
&\quad \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \sum_{d=C+1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_d} \frac{KCd}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j)
\end{aligned}$$

For $d \leq C$, we rearrange Equation A.4 as,

$$\begin{aligned}
\mathcal{Q} &= \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) - \\
&\quad \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \left(\pi_{d'}(C, \mathbf{A}) - \frac{KCd'}{\mathcal{D}(\mathbf{A})} \right) \sigma(\mathbf{z}_i^T \mathbf{z}_j) \\
&\quad + \sum_{d'=C+1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) \\
&= \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) - \\
&\quad \sum_{(i,j) \in \mathcal{S}_d} \left(\pi_d(C, \mathbf{A}) - \frac{KCd}{\mathcal{D}(\mathbf{A})} \right) \sigma(\mathbf{z}_i^T \mathbf{z}_j) \\
&\quad - \sum_{\substack{d'=1 \\ d' \neq d}}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \left(\pi_{d'}(C, \mathbf{A}) - \frac{KCd'}{\mathcal{D}(\mathbf{A})} \right) \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad \sum_{d'=C+1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) \\
&= \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) - \left(\pi_d(C, \mathbf{A}) - \frac{KCd}{\mathcal{D}(\mathbf{A})} \right) |\mathcal{S}_d| \xi_d \\
&\quad - \sum_{\substack{d'=1 \\ d' \neq d}}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \left(\pi_{d'}(C, \mathbf{A}) - \frac{KCd'}{\mathcal{D}(\mathbf{A})} \right) \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad \sum_{d'=C+1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j)
\end{aligned}$$

$$\begin{aligned}
\xi_d = & \frac{1}{|\mathcal{S}_d| \left(\pi_d(C, \mathbf{A}) - \frac{KCd}{\mathcal{D}(\mathbf{A})} \right)} \times \\
& [-\mathcal{Q} + \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) \\
& - \sum_{\substack{d'=1 \\ d' \neq d}}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \left(\pi_{d'}(C, \mathbf{A}) - \frac{KCd'}{\mathcal{D}(\mathbf{A})} \right) \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
& \sum_{d'=C+1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j)]
\end{aligned}$$

From the above Equation, $\xi_d = f(d, \Theta)$ for $d \leq C$, where Θ is the set of parameters of f other than d .

For $d > C$, we rearrange Equation A.4 as,

$$\begin{aligned}
\mathcal{Q} &= \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) - \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} (\pi_{d'}(C, \mathbf{A}) \\
&\quad - \frac{KCd'}{\mathcal{D}(\mathbf{A})}) \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \sum_{d'=C+1}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) \\
&= \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) - \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} (\pi_{d'}(C, \mathbf{A}) \\
&\quad - \frac{KCd'}{\mathcal{D}(\mathbf{A})}) \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \sum_{(i,j) \in \mathcal{S}_d} \frac{KCd}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \\
&\quad \sum_{\substack{d'=C+1 \\ d' \neq d}}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) \\
&= \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) - \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} (\pi_{d'}(C, \mathbf{A}) \\
&\quad - \frac{KCd'}{\mathcal{D}(\mathbf{A})}) \sigma(\mathbf{z}_i^T \mathbf{z}_j) + \frac{KCd}{\mathcal{D}(\mathbf{A})} |\mathcal{S}_d| \xi_d \\
&\quad + \sum_{\substack{d'=C+1 \\ d' \neq d}}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j)
\end{aligned}$$

$$\begin{aligned}
\xi_d &= \frac{\mathcal{D}(\mathbf{A})}{|\mathcal{S}_d| KC d} \left[\mathcal{Q} - \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} \pi_{d'}(C, \mathbf{A}) \right. \\
&\quad \left. + \sum_{d'=1}^C \sum_{(i,j) \in \mathcal{S}_{d'}} (\pi_{d'}(C, \mathbf{A}) - \frac{KCd'}{\mathcal{D}(\mathbf{A})}) \sigma(\mathbf{z}_i^T \mathbf{z}_j) \right. \\
&\quad \left. - \sum_{\substack{d'=C+1 \\ d' \neq d}}^{d_{max}} \sum_{(i,j) \in \mathcal{S}_{d'}} \frac{KCd'}{\mathcal{D}(\mathbf{A})} \sigma(\mathbf{z}_i^T \mathbf{z}_j) \right]
\end{aligned}$$

From the above Equation, ξ_d is inversely proportional to d for $d > C$. □

A.2 Implementation Details

All our codes and datasets are available at <https://github.com/Distance-awareNS/DNS/>

Synthetic Data Generation: We construct the synthetic networks by generating a node degree sequence that follows the power-law distribution. For our experiments, we use *networkx.utils.powerlaw_sequence* to generate the degree sequence which takes two parameters: the number of nodes and the exponent of the power-law distribution, where we set the number of nodes as 2,000 and vary the exponent to generate varying networks with different density. After that, we use *networkx.expected_degree_graph* to construct a network from each degree sequence; whereas each network may have many disconnected components. To connect all the components of the network, we randomly choose one node from each disconnected component and connect them using minimum number of artificial edges. We generate structure-induced node labels using a simple label propagation approach. Initially, we randomly select k seed nodes for k distinct classes. For sparse network, we choose $k = 7$ classes, whereas, for moderate and dense networks, we choose $k = 5$ and $k = 4$ classes respectively. At each iteration, we propagate the node label to its adjacent unlabeled nodes. Consequently, we iterate this procedure until all the nodes get labeled. Therefore, the node labels are generated only using the structure information, such as the proximity from the seed node.

Node Classification Setup: We use Logistic Regression (LR) with an lbfgs solver that supports 150 max iterations as our downstream model. For the PPI dataset, we use multi-class settings of LR. For benchmark datasets, we use the PyTorch Geometric *train-test-validation* mask on the largest component to generate the training nodes, testing nodes and validation nodes. Meanwhile, for the synthetic datasets, we randomly select 10% nodes for

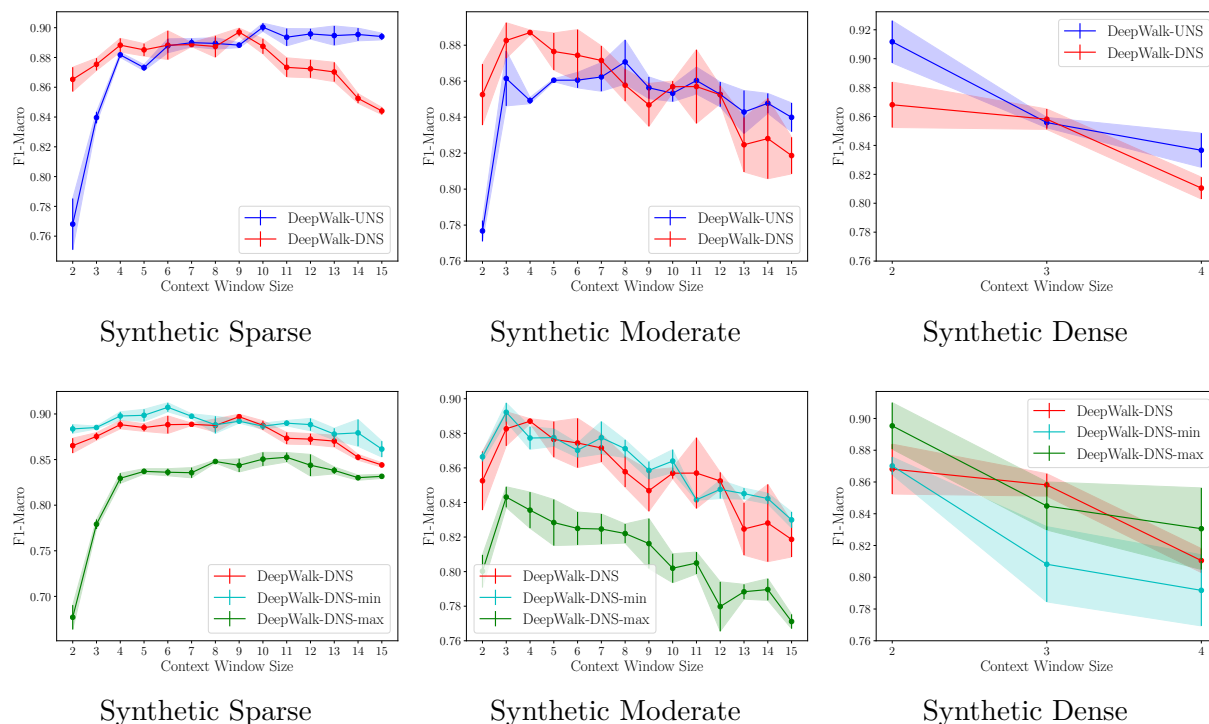


Figure A.1: F1-Macro score plot with varying context window on Synthetic Sparse, Synthetic Moderate, and Synthetic Dense dataset. Competing models are DeepWalk-UNS, DeepWalk-DNS, and its variants DeepWalk-DNS-min, DeepWalk-DNS-max.

training, 40% for validation, and 40% for testing. We ran our experiments in a single machine with 2 NVIDIA Titan RTX GPUs (24Gb of RAM) and 1 Intel(R) Xeon(R) W-2135 CPU (@ 3.70GHz). We use PyTorch with cuda-10.1 for our experiments.

Hyper-parameters: We set the embedding dimension as 128, the number of random walks per node as 50, and the number of negative samples as 20, whereas for node2vec model, we also set return parameter p as 1 and inout parameter q as 4 for our experiments. To optimize these models, we use Adam optimizer with 0.01 learning rate. Moreover, we run all node wise negative sampling-based GRL model for 30 epochs and all edgewise negative sampling base models for 400 epochs, as edgewise sampler models take more iterations to converge.

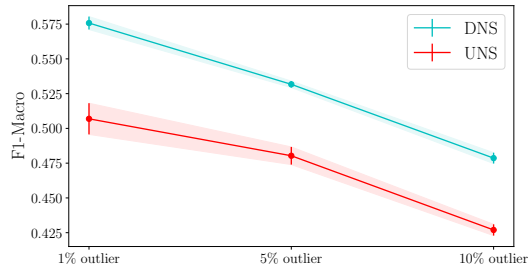


Figure A.2: Sensitivity towards outlier points on CiteSeer dataset.

A.3 Additional Analysis of Results

Ablation Study:

We perform an ablation study of our DNS sampler by splitting the negative sampling probability into two parts; the splitting point is the pairwise distance for which DNS probability \approx UNS probability. In the first ablation model, we set negative sampling probability linearly proportional to the pairwise distance for nearby nodes while maintaining uniform negative sampling probability for the rest of the nodes. In the second ablation model, we set a uniform negative sampling probability for nearby nodes while setting DNS-like probability for distant nodes. Let us denote the first sampler as DNS-min since its negative sampling probability $P_{min}(k|i) = \min(P_{DNS}(k|i), \frac{1}{n})$ and the second sampler as DNS-max that has negative sampling probability $P_{max}(k|i) = \max(P_{DNS}(k|i), \frac{1}{n})$. Both DNS-min and DNS-max samplers have higher separation than UNS sampler.

Figure A.1 shows the node classification performance of different samplers with the DeepWalk model. The top row of Figure A.1 shows the F1-Macro score of DeepWalk-UNS and DeepWalk-DNS with varying context size. As discussed in Section 2.4, we see the performance of the DeepWalk-DNS model decreases with increasing context window size. Moreover, low negative sampling probability for nearby nodes is not effective for the synthetic dense graph.

In the second row of Figure A.1, we see the comparison of DNS, DNS-min, and DNS-max in terms of node classification performance. The DeepWalk-DNS-max follows the trend of the DeepWalk-UNS model performance for lower context windows, whereas, the DeepWalk-DNS-min model more likely follows the trend of the DeepWalk-DNS model in all the synthetic graphs.

γ -linear Negative Sampling:

We perform an empirical study to visualize the effect of varying γ in γ -linear negative sampler on the synthetic datasets. We train the DeepWalk model with γ -linear negative sampler and denote it by γ -DNS. Moreover, we choose different values for γ from 0 to 1.25 for this experiment (the models are denoted by $\gamma(\text{value})$ -DNS). In this experiment, we denote the DeepWalk-DNS model by $\gamma(1.0)$ -DNS. From Figure A.3, we see that γ value closer to 1 follows the trend of the DeepWalk-DNS performance, whereas, γ value closer to 0 follows the trend of the DeepWalk-UNS model performance. Theoretically, the performance of the $\gamma(0)$ -DNS-based model should be close to the UNS-based model, but there is deviation across runs that require further investigation.

Sensitivity towards outlier points:

To evaluate the sensitivity towards outlier points, we artificially add distant nodes with similar class values on the CiteSeer dataset. Figure A.2 shows the sensitivity of our DNS model and the UNS model for these outlier points. Both models perform poorly with increasing number of outliers, which shows outliers hurt all models with the local similarity assumption.



Figure A.3: Node classification performance (F1-Macro score) comparison for various γ -linear sampler based models with varying context window on Synthetic Sparse, Synthetic Moderate, and Synthetic Dense dataset.

Appendix B

Chapter 3

B.1 Comparison of CAMs and Saliency Maps Using Hyperplanes

B.1.1 Theoretical Proofs of CAM and SM-Hyperplanes

The proof of the Remark 3.4 and 3.6 are provided below.

Remark B.1. If a point $\mathbf{a} \in \mathcal{A}$ corresponding to a ground-truth pixel lies above \mathcal{H}_{cam}^c , i.e., $\mathbf{w}_c^T \mathbf{a} / Z - \tau_{cam} \geq 0$, then the pixel belongs to DR; otherwise, it belongs to NDR.

Proof. The activation map \mathbf{A} for an image \mathbf{I} can be sampled at any arbitrary ground-truth pixel $(i, j) \in \mathcal{S}_{GT}^c$ such that $\mathbf{A}_{(i,j)} \in \mathcal{A}$. Therefore, the CAM score for the c -th Class at pixel (i, j) can be computed as: $CAM_c(i, j) = \mathbf{w}_c^T \mathbf{A}_{(i,j)} / Z$.

Now, if the pixel belongs to the discriminative region (DR), then by definition 3.1 we get the following:

$$\begin{aligned} CAM_c(i, j) &= \frac{\mathbf{w}_c^T}{Z} \mathbf{A}_{(i,j)} \geq \tau_{cam} \\ \implies \frac{\mathbf{w}_c^T}{Z} \mathbf{A}_{(i,j)} - \tau_{cam} &\geq 0 \end{aligned} \tag{B.1}$$

This by definition of \mathcal{H}_{cam}^c (see Definition 3.3) suggests that the activation value $\mathbf{A}_{(i,j)} \in \mathcal{A}$

lies above \mathcal{H}_{cam}^c .

Conversely, if the ground-truth pixel belongs to the non-discriminative region (NDR), then by definition 3.1 we get the following:

$$\begin{aligned} CAM_c(i, j) &= \frac{\mathbf{w}_c^T}{Z} \mathbf{A}_{(i,j)} < \tau_{cam} \\ \implies \frac{\mathbf{w}_c^T}{Z} \mathbf{A}_{(i,j)} - \tau_{cam} &< 0 \end{aligned} \quad (\text{B.2})$$

Similarly, from the definition of \mathcal{H}_{cam}^c (see Definition 3.3), the activation value $\mathbf{A}_{(i,j)} \in \mathcal{A}$ lies below \mathcal{H}_{cam}^c .

Therefore, we can say in general if an arbitrary point \mathbf{a} corresponding to a ground-truth pixel lies above \mathcal{H}_{cam}^c , it belongs to the discriminative region (DR); otherwise it belongs to NDR. \square

Remark B.2. If a point \mathbf{a}' corresponding to a ground-truth pixel lies on the outer sides of \mathcal{H}_{sm}^c , i.e., $|\mathbf{w}_c^T \mathbf{a}'| - \tau_{sm} \geq 0$, then the point belongs to HSR; otherwise, it belongs to LSR.

Proof. Similar to CAM, the gradient of the GAP w.r.t. the image $\frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}}$ can be sampled at any arbitrary ground-truth pixel $(i, j) \in \mathcal{S}_{GT}^c$ such that $\frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}}|_{(i,j)} \in \mathcal{A}'$. Therefore, the Saliency map score for the c -th Class at pixel (i, j) can be computed as:

$$SM_c(i, j) = \left| \mathbf{w}_c^T \frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \Big|_{(i,j)} \right| \quad (\text{B.3})$$

Now, if the pixel belongs to the high saliency region (HSR), then by definition 3.2 we get

the following:

$$\begin{aligned}
 SM_c(i, j) &= \left| \mathbf{w}_c^T \frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \Big|_{(i,j)} \right| \geq \tau_{sm} \\
 \implies \left| \mathbf{w}_c^T \frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \Big|_{(i,j)} \right| - \tau_{sm} &\geq 0
 \end{aligned} \tag{B.4}$$

This by definition of the \mathcal{H}_{sm}^c (see Definition 3.5) suggests that the gradient $\frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \Big|_{(i,j)} \in \mathcal{A}'$ lies on the outer sides of the \mathcal{H}_{sm}^c .

Similarly, if the pixel belongs to the low saliency region (LSR), then by definition 3.2 we get the following:

$$\begin{aligned}
 SM_c(i, j) &= \left| \mathbf{w}_c^T \frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \Big|_{(i,j)} \right| < \tau_{sm} \\
 \implies \left| \mathbf{w}_c^T \frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \Big|_{(i,j)} \right| - \tau_{sm} &< 0
 \end{aligned} \tag{B.5}$$

This by definition of \mathcal{H}_{sm}^c (see Definition 3.5) suggests that the gradient $\frac{\partial \text{GAP}(\mathbf{A})}{\partial \mathbf{I}} \Big|_{(i,j)} \in \mathcal{A}'$ lies on the inner sides of the \mathcal{H}_{sm}^c .

Therefore, we can say in general if an arbitrary point \mathbf{a}' corresponding to a ground-truth pixel lies on the outer sides the \mathcal{H}_{sm}^c , it belongs to the high-saliency region (HSR); otherwise it belongs to low saliency region (LSR). \square

B.1.2 Visual Comparison for more representative images from VOC

Figure B.1 presents a visual comparison of CAMs and Saliencies using the hyperplanes on more representative images from the PASCAL VOC 12 dataset (similar to Figure 3.1 from the main paper). For this experiment, we choose the value of $\tau_{cam} = 0.25$ and $\tau_{sm} = 0.15$.

B.2 Experimental Details

B.2.1 Dataset Description

We compared different competing approaches quantitatively and qualitatively by conducting experiments on MNIST, PASCAL VOC '12, and MS COCO '14 datasets.

MNIST Segmentation Dataset:

We generate the ground-truth segmentation masks by filtering the non-zero pixels of the MNIST images. For our experiments, we used an upsampled version of the original MNIST dataset, where we used “nearest neighbor” interpolation to upsample the dataset to 128×128 dimension. Furthermore, we used 60,000 training set and 10,000 test set images with segmentation masks for our experiments in Section 3.5.

PASCAL VOC '12 Dataset:

The PASCAL VOC 2012 dataset contains 10,582 training images, 1,449 validation images, and 1,456 test images with objects from 20 classes. We compared the methods by evaluating

the performance of the 1,464 segmented images using the approach adopted in recent WS3 research.

MS COCO '14 Dataset:

The MS COCO 2014 dataset contains 82,783 training and 40,504 validation images with objects from 80 classes. We evaluated the competing approaches on approximately 82K training images from the MS COCO 2014 dataset.

B.2.2 Model Description

We fine-tuned a classification network to accurately extract segmented seeds, utilizing ResNet50 as the backbone network, which is pre-trained on ImageNet. In order to maintain consistency with prior research, we incorporated various augmentations during the fine-tuning process, such as resizing to (320, 640), applying a horizontal flip with a 0.5 probability, and cropping with a maximum size of 512. We developed and fine-tuned three separate classification models to explore the impact of different perturbations during the fine-tuning stage. *Model-org* model is fine-tuned only with the aforementioned augmentations. During fine-tuning, we perturb the input image with binary noise to create additional augmentations for *Model-pert-binary* classification model. Formally,

$$\tilde{\mathbf{I}} = \mathbf{I} \odot m$$

$$m \sim \text{Bernoulli}(p), \quad \text{where, } p = 0.9$$

$\tilde{\mathbf{I}}$ is the training image that is perturbed with the binary mask m . The mask has a binary probability $p = 0.9$ to set each pixel. Similar to *Model-pert-binary*, we additionally per-

turb the input image with Gaussian noise for the *Model-pert-gaussian* classification model. Formally,

$$\begin{aligned}\tilde{\mathbf{I}} &= \mathbf{I} + \epsilon \\ \epsilon &\sim \mathcal{N}(0, \sigma), \quad \text{where, } \sigma = 0.15\end{aligned}$$

$\tilde{\mathbf{I}}$ is the training image that is perturbed with the Gaussian noise ϵ . The noise level (perturbation) is controlled by the standard deviation $\sigma = 0.15$.

B.2.3 Background Resolve Techniques

Basic Background Resolve

This is the most common approach in recent research that uses a simple strategy for distinguishing between foreground and background classes. This is done by setting a global threshold that discerns the background class and then assigning classes based on the highest real values among the foreground classes.

Kernel Smoothing

The technique of Kernel Smoothing has been utilized to smooth the gradients of the vanilla saliency maps by applying a Gaussian kernel with a kernel size of 13 and a standard deviation of 5. Following this, a global threshold has been selected to distinguish foreground classes from the background. This has been achieved by considering the maximum real values of the smoothed saliencies for the target classes. This approach has been adopted to enhance the accuracy of the saliency maps by smoothing the gradients as a post-processing step.

Supapixel-based Background Resolve

Supapixels consist of clusters of pixels that exhibit similar characteristics. In contrast to the conventional method of assigning a label to each individual pixel, supapixel-based smoothing allocates a label to each supapixel, effectively reducing the noise and scatteredness present in saliency maps. We employed Felzenszwalb’s efficient graph-based supapixel algorithm [30] to compute the supapixels. To designate a class label for each supapixel, we initially calculated the mean saliencies for every supapixel. And then, using a global threshold of 0.3, we determine whether a supapixel is part of the foreground or background. We assigned target classes for foreground supapixels based on the highest mean gradients concerning the target classification score.

In an effort to better understand the performance of background resolution techniques, Figure B.2 presents a visual comparison between CAM and Vanilla Saliency with different resolution methods, namely Basic, Smooth and Supapixel. The basic background resolution is represented by “Vanilla Saliency”. For this experiment, we set a global threshold of 0.15 to differentiate the foreground from the background. Building upon the insights gathered from Section 3.5.3, we observed the following implications for each background resolution approach. Employing vanilla saliency with Basic background resolution results in noisy and scattered saliency maps, demonstrating its limitations in providing clear object segmentation. Utilizing Kernel smoothing generates smooth saliencies, which offers an improvement over the Basic technique. However, this approach still struggles with unclear object boundaries, making it difficult to precisely locate objects within the image (First and Sixth row of Figure B.2). The Supapixel-based background resolution effectively smooths the saliencies while maintaining clear and distinguishable object boundaries, presenting a more refined solution (First, Fourth, and Sixth row of Figure B.2). Nonetheless, this method has its drawbacks, as the resulting saliencies heavily rely on the supapixel shapes and the algorithm’s ability

to identify them accurately. Consequently, any slight deviation from the correct superpixel shape can cause this background resolution technique to fail in capturing the entire body of the target object (Eight, tenth, and eleventh row of Figure B.2).

B.3 Stochastic Aggregation for Saliencies

B.3.1 SmoothGrad and BinaryMask

To reduce noise, [117] proposes a stochastic aggregation-based saliency map, namely SmoothGrad, where Gaussian noise is added to the input image to construct a neighborhood of the input image. Then, n different random samples are selected from the neighborhood, and the saliencies of all the samples are averaged to generate the final saliency, which is much smoother than the Vanilla Saliency.

In this paper, we explored another variation of input noise perturbation, namely BinaryMask, where, instead of adding Gaussian noise to the input image, we multiply the image by a binary noise. We can formally define both these methods as follows:

$$\tilde{\text{SM}}_c(x) = \frac{1}{n} \sum_1^n \text{SM}_c(\tilde{\mathbf{I}}) \quad (\text{B.6})$$

$$\tilde{\mathbf{I}} = \mathbf{I} + \epsilon; \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \text{ for SmoothGrad} \quad (\text{B.7})$$

$$\tilde{\mathbf{I}} = \mathbf{I} \odot m; \quad m \sim \text{Bernoulli}(p), \text{ for BinaryMask} \quad (\text{B.8})$$

\mathbf{I} in equation B.6 corresponds to the input image, whereas $\tilde{\mathbf{I}}$ denotes the noisy input and m denotes the binary mask. $\text{SM}_c(\cdot)$ is the vanilla saliency map and $\tilde{\text{SM}}_c$ corresponds to the final aggregated saliency.

The amount of perturbation for SmoothGrad is controlled by the standard deviation, σ , (also called noise level) of the Gaussian noise. Whereas, for BinaryMask, the binary probability p controls the perturbation magnitude. With a higher binary probability p , a higher number of input pixels are in $\tilde{\mathbf{I}}$, which means lower binary perturbation. For our experiment, we fixed the noise level as 0.5 and the binary probability p as 0.90. $n = 50$ samples have been selected from the neighborhood for our experiments. We added these noises to the input images as additional augmentations during fine-tuning. “*Model-pert-binary*” and “*Model-pert-gaussian*” are two finetuned classifiers augmented by binary and Gaussian noise, respectively.

Figure B.3 presents a visual comparison of SmoothGrad saliencies derived from the “Model-org” and “Model-pert-gaussian” models. As SmoothGrad employs a stochastic aggregation approach, the basic background resolution yields significantly smoother saliencies for both models. Nevertheless, the “Model-pert-gaussian” model exhibits superior saliencies in terms of the performance metrics discussed in Section 3.4. In terms of visual quality, the “Model-pert-gaussian” column of Figure B.3 exhibits superior saliencies compared to the “Model-org” column. However, the perturbed model occasionally generates overly smooth saliencies, resulting in unclear object boundaries, as observed in the first and sixth rows of Figure B.3. Similarly, Figure B.4 offers a visual comparison of BinaryMask saliencies for the “Model-org” and “Model-pert-binary” models. In this case, the “Model-pert-binary” model demonstrates higher quality saliencies, as observed in the first, second, and sixth rows of Figure B.4. Reinforcing the insights obtained from Section 3.6, both these figures support the notion that the classification model should be fine-tuned using similar noise in order to yield better-quality saliencies.

B.3.2 Analysis of The Sensitivity Towards Noise.

Figure B.5 and B.6 illustrate the sensitivity of performance scores concerning the magnitude of noise and the number of neighborhood samples, respectively. In the case of SmoothGrad, the noise levels (standard deviation σ) dictate the magnitude of perturbation, with a higher σ corresponding to a greater noise magnitude. Conversely, for BinaryMask, the binary probability p governs the perturbation magnitude, with a lower probability p corresponding to a higher level of perturbation. As evident from Figure B.5, the models demonstrate sensitivity towards increased perturbation, with the NDR-Recall decreasing for higher noise levels in both cases. SmoothGrad exhibits greater sensitivity to higher perturbation, while BinaryMask displays less sensitivity to perturbation magnitude in terms of mIoU and FG-precision. As illustrated in Figure B.6, the performance remains relatively stable for the number of samples $n > 20$. However, when $n < 20$, the performance improves as the number of samples increases.

By examining Figure B.7, we can see that excessively adding noise to the input image has a negative impact. As a result, the mIoU performance decreases for noise levels above 0.20. Adding noise may make the saliency maps smoother; however, with increasing noise, the saliency maps may become unstable (shown in the noise level 50% column). Figure B.8 depicts the sensitivity of BinaryMask saliency with respect to binary probability. The visualization reveals that as perturbation increases (low probability), the saliencies become less stable, as shown in the third and fourth columns of Figure B.8. In contrast, higher probability leads to enhanced saliency quality, as evident in the seventh and eighth columns of Figure B.8. It is important to note that a binary probability of 1.0 does not involve any stochastic aggregation, as all pixels are selected to compute the saliency.

B.4 Stochastic Aggregation Through Cropping

B.4.1 Analysis of The Sensitivity For Random Cropping.

Figure B.9 shows the sensitivity of the performance metrics towards the number of crops and the scale of crops for random cropping. With an increasing number of crops, the performance of random cropping-based saliencies improves. However, after 140 crops, we see the performance saturates. Choosing the correct scale of random crops is critical for better performance. The scale of the crops should not be lower than 0.10. The performance of the random cropping method saturates after a scale of 0.10.

B.4.2 Different Variations of Cropping

In this subsection, we explore different variations of random cropping and patching techniques that break the spatial structure of input images. Random patching is an erasure-based method similar to the idea of cutout [24] technique. We divide the full image into 16×16 grid-wise patches for random patching. Then we randomly mask out some of the patches with a Bernoulli probability of 0.1, also called patching probability. The random patching idea is similar to the BinaryMask method in the sense that we are turning off some grid of pixels instead of individual pixels with a probability. Using stochastic aggregation of the

Given a CAM of an input image, we also explore the discriminative patching idea, where the patching probability is the complement of the CAM score S_{cam}^c for each patch for the c -th class. It is important to mention that S_{cam}^c corresponds to the maximum CAM score across all the C classes in the patch (where C is the total number of classes). The discriminative

patch (disc-Patch) is implemented as follows:

$$\begin{aligned} p &= \alpha * S_{cam}^c \\ \bar{m} &= 1 - m; \quad m \sim \text{Bernoulli}(p) \\ \tilde{\mathbf{I}} &= \mathbf{I} \odot \bar{m} \end{aligned}$$

\bar{m} is the binary filter applied to the patches and $\tilde{\mathbf{I}}$ denotes the perturbed image. S_{cam}^c is multiplied by $\alpha \in (0, 1)$ so that the discriminative patch probability does not reach 0 for the most discriminative region. For our experiments, we choose $\alpha = 0.4$.

Similar to discriminative patching, we explore discriminative cropping, where the selection of each crop has a probability that is the complement to the CAM score S_{cam}^c for that crop. The discriminative cropping is implemented as follows:

$$\begin{aligned} p &= \text{ReLU}(\beta - S_{cam}^c) \\ \tilde{\text{SM}}_c(x) &= \frac{1}{n} \sum_{i=1}^n m * w_i * \text{SM}_c(\tilde{\mathbf{I}}_i); \\ m &\sim \text{Bernoulli}(p) \end{aligned}$$

$\tilde{\text{SM}}_c(x)$ is the final aggregated saliency using discriminative cropping. m is the binary filter applied to the crops and $\tilde{\mathbf{I}}$ denotes the perturbed image. We choose $\beta = 0.7$ for our experiments.

Figure B.10 provides a visual comparison of saliencies generated by Random Cropping, Random Patching, Discriminative Cropping, and Discriminative Patching. As discussed in Section 3.7, both Random Cropping and Discriminative Cropping display higher quality and more stable saliencies. In contrast, the saliencies produced by Random Patching and Dis-

criminative Patching are less stable, primarily due to the fact that the classification model has not been fine-tuned with similar noise perturbation. For instance, in Figure B.10, the second and fourth rows display poor saliency maps for the patching methods. Conversely, the sixth, seventh, and eighth rows exhibit higher-quality saliencies for the patching method. Moreover, the discriminative variations of both these methods demonstrate a modest enhancement in saliency quality, as evidenced by the second, fourth, sixth, and seventh rows of Figure B.10.

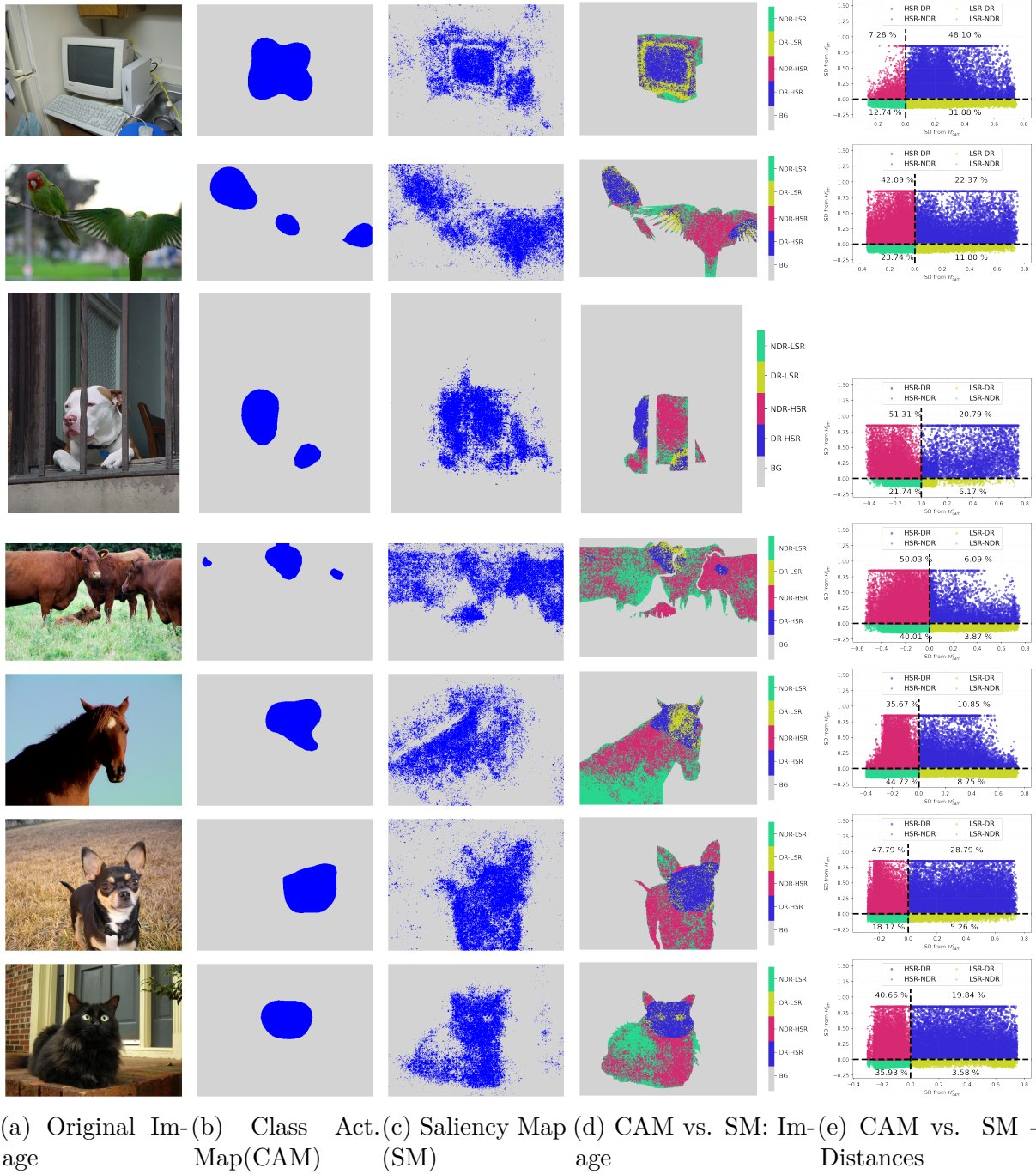


Figure B.1: A visual comparison of CAMs and saliency maps (SMs) for more representative images from the VOC12 dataset.



Figure B.2: Visual comparison between CAM and Vanilla Saliency with different background resolves.

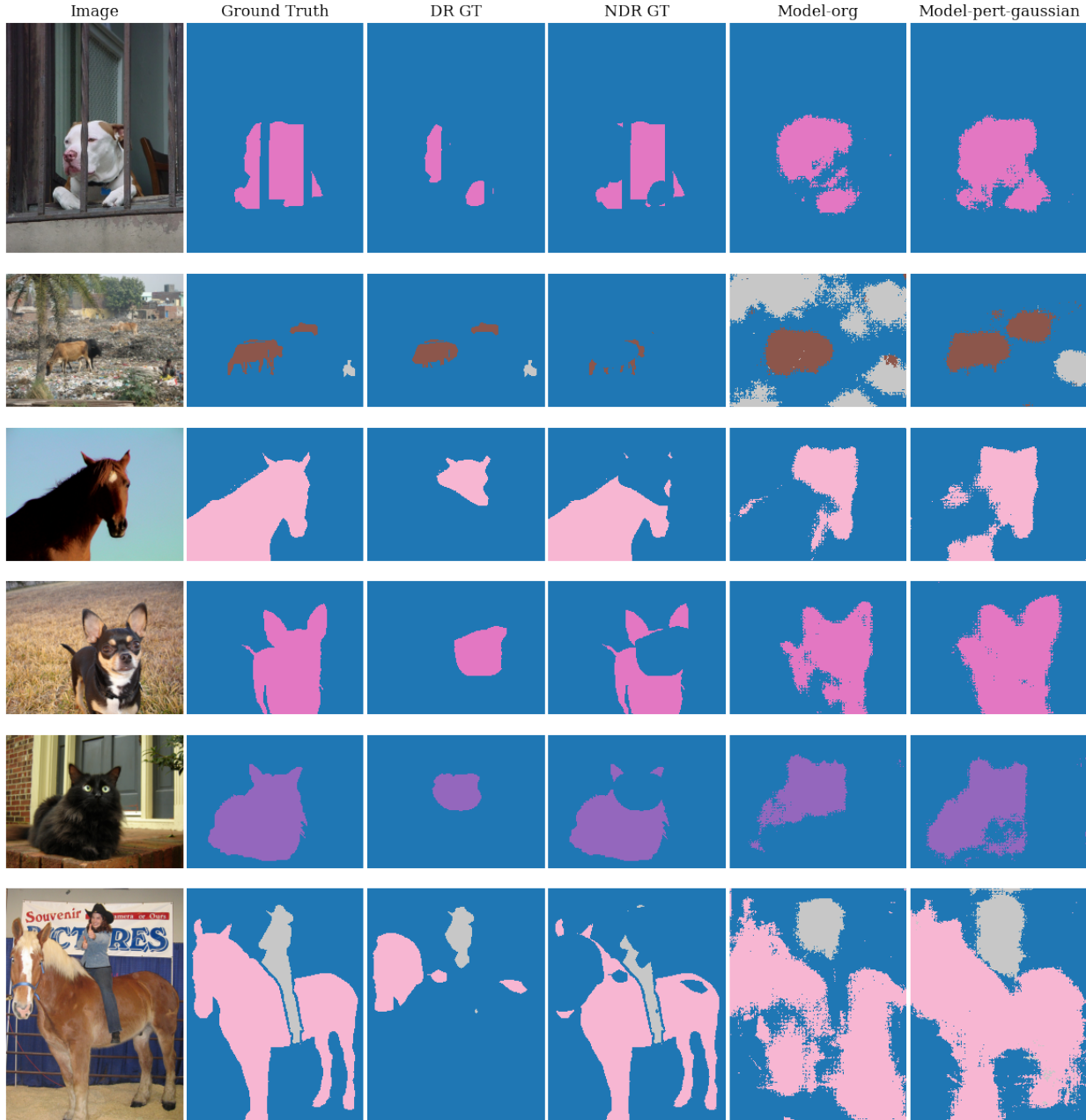


Figure B.3: Visual comparison of SmoothGrad saliencies between “Model-org” and “Model-pert-gaussian” fine-tuned model. Saliencies with basic background resolve are shown in the figure.

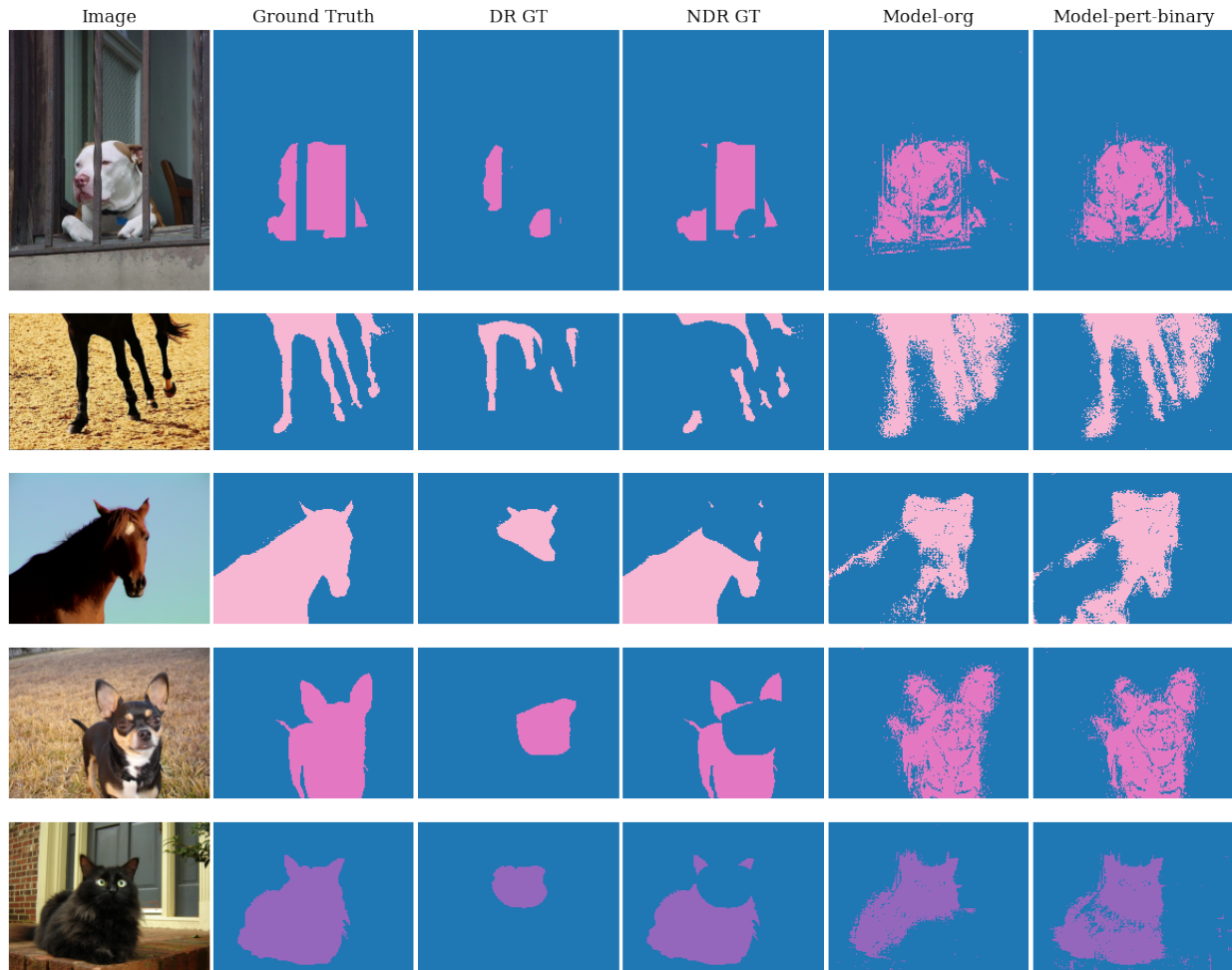


Figure B.4: Visual comparison of BinaryMask saliencies between “Model-org” and “Model-pert-binary” fine-tuned model. Saliencies with basic background resolve are shown in the figure.

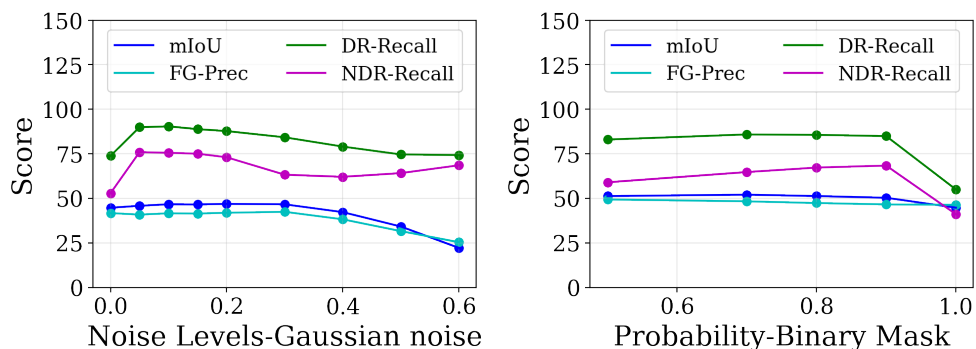


Figure B.5: Sensitivity plots of the performance towards Gaussian noise levels σ (left); towards binary probability p (right).

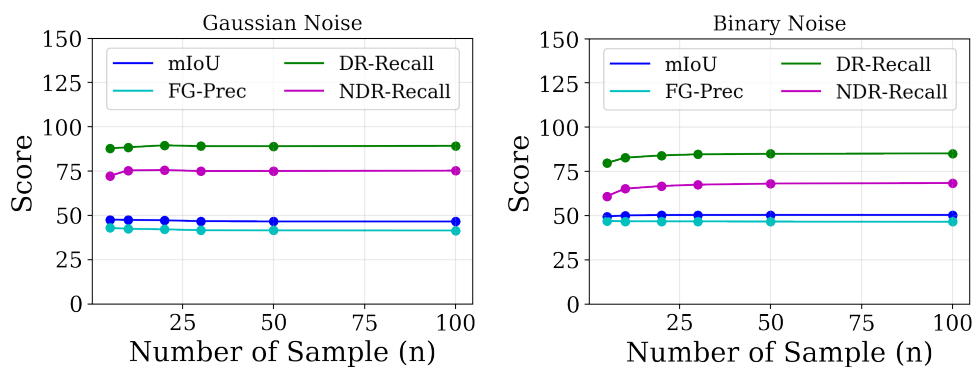


Figure B.6: Sensitivity plots of the performance towards the number of samples n for (left) SmoothGrad; (right) BinaryMask.

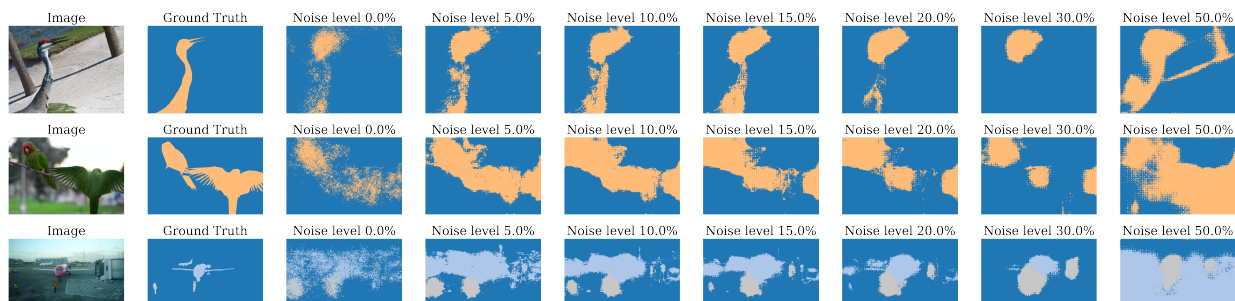


Figure B.7: Visual evaluation of the sensitivity towards the noise level σ of the Gaussian noise (SmoothGrad saliency with basic background resolve).

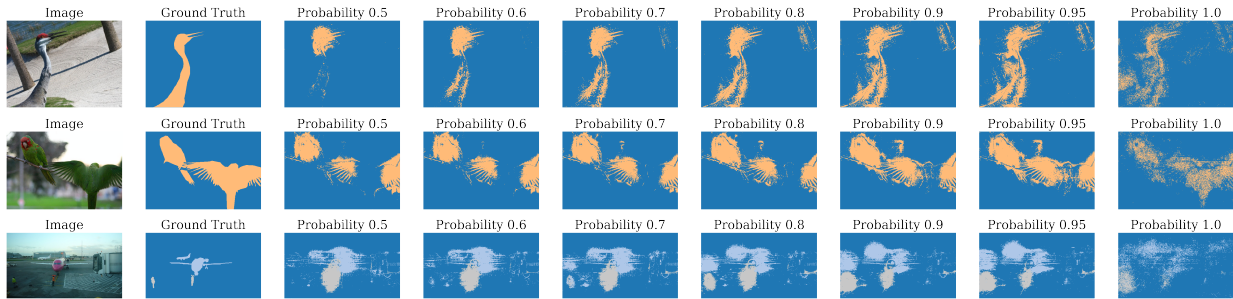


Figure B.8: Visual evaluation of the sensitivity towards the binary probability of the perturbation (BinaryMask saliency with basic background resolve).

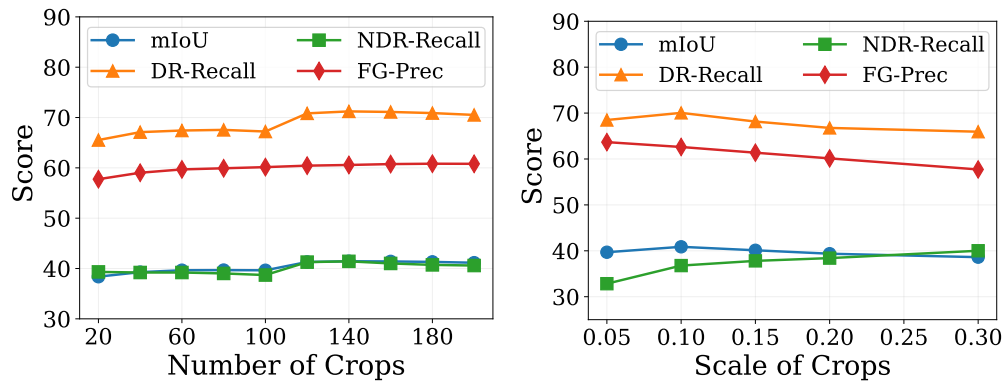


Figure B.9: Sensitivity plots of the performance for random cropping (left) to the number of crops; (right) to the scale of the crops.



Figure B.10: Visual comparison between Random Cropping, Random Patching, Discriminative Cropping, and Discriminative Patching saliencies. Saliencies with basic background resolve are shown in the figure.

Appendix C

Chapter 4

C.1 Dataset Preprocessing

We collected images of three taxonomic groups of organisms: fish, birds, and butterflies, each containing around 10K images. Images for fish (**Fish-10K**) were curated from the larger image collection, FishAIR [31], which contains images from the Great Lakes Invasive Network Project (GLIN) [1]. These images originate from various museum collections such as INHS [5], FMNH [4], OSUM [78], JFBM [2], UMMZ [3] and UWZM [6]. We created the Fish-10K dataset by randomly sampling 10K images and preprocessing the images to crop and remove the background.

To ensure diversity within the Fish-10K dataset, we applied a targeted sampling strategy in the source collection, FishAIR [31]. Specifically, we retained all images of species with fewer than 200 images, considering these as minority or rare classes. Random sampling was applied only to the majority species—those with more than 200 images per class. To assess the potential sampling bias among the majority species, we generated feature vectors for each image in Fish-10K using a pretrained VGG-19 model. In Figure C.1, we present species-wise t-SNE plots of these feature vectors for several majority species. Our analysis shows that the distribution of sampled images closely mirrors the distribution of images that were not included in the dataset (denoted as “others” in the plot). This suggests that our random sampling approach provides a sufficiently accurate representation of the original

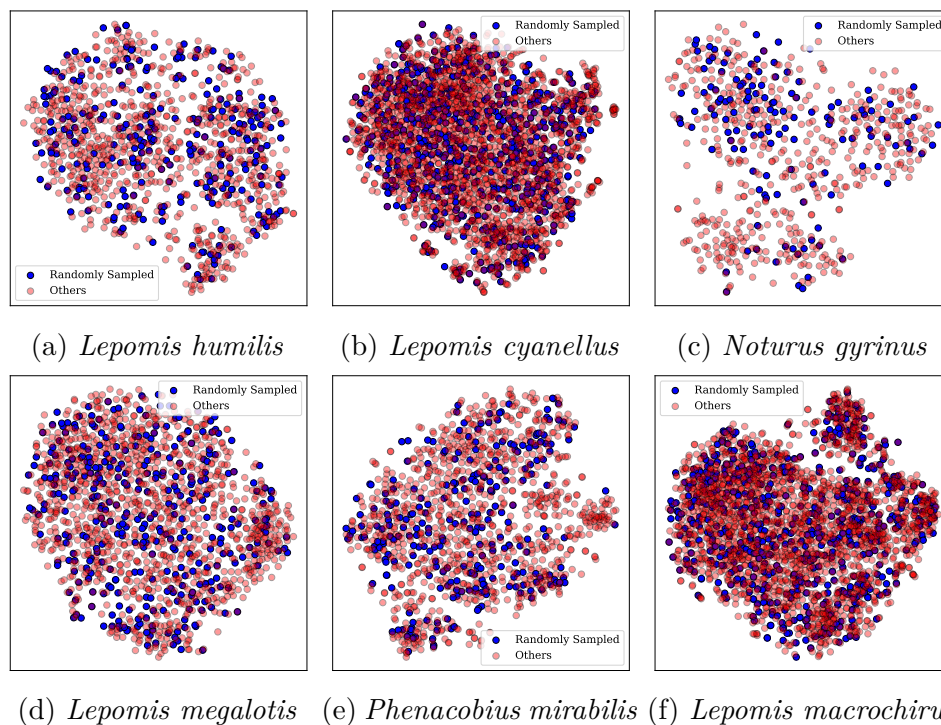


Figure C.1: t-SNE plots to illustrate the effectiveness of random sampling with the majority species in the Fish-10K dataset. Randomly sampled images are shown as blue dots, while the remaining data points are represented by red dots. Subcaptions display the scientific names of the corresponding species. To generate the vector representation of the images, we leverage a VGG19 pretrained on the ImageNet dataset.

distribution for the majority species. For consistency, we leverage GroundingDINO [74] to crop the fish body from the background and Segment Anything Model (SAM) [57] to remove the background. The Fish-10K dataset contains images of specimens preserved in museum collections with artificial backgrounds with imaging artifacts that are not typical for large-scale computer vision datasets. Moreover, these backgrounds can introduce unexpected bias. Hence, we removed the backgrounds using SAM to create a controlled environment for our experiments.

We curated the images for butterflies (**Butterfly-10K**) from the Jiggins Heliconius Collection dataset [60], which has images collected from various sources¹. We carefully sampled

¹Sources: [48, 49, 50, 81, 82, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 103, 107, 108, 109, 110, 135, 136,

10K images for Butterfly-10K from the entire collection to ensure the images capture unique specimens and represent a diverse set of species by adopting the following two steps. **First**, the butterfly images show various angles, including dorsal and ventral views, forewing dorsal and ventral views, and hindwing dorsal and ventral views. To ensure consistency, we only selected images with dorsal view and removed all images of hybrid species. **Second**, we further filtered the dataset based on the unique specimen ID to ensure no specimen was repeated more than once. For species with more than 2000 images, we performed random sampling (no sampling was performed for species with sizes less than 2000). We ensure each species has a minimum of 20 images and no more than 2,000 images. The Butterfly-10K dataset contains a significant number of images of *Heliconius melpomene* and *Heliconius erato* species. We utilized the subspecies information of these two species to create a hard dataset for analyzing the impact of answer choices on VLM performance, as described in Section 4.5.1.

The images for birds (**Bird-10K**) are obtained from the CUB-200-2011 [131] dataset by taking 190 species for which the common name to scientific name mapping is available. This results in a fairly balanced dataset with around 11K images in total.

The scientific names for the images of Fish-10K and Butterfly-10K were obtained directly from their respective sources. For Bird-10K, we obtained the scientific names from the iNatLoc500 [21] dataset. We curated around 31K question-answer pairs in both open and multiple-choice (MC) question formats for evaluating species classification tasks. The species-level trait presence/absence matrix for Fish-10K was manually curated with the help of biological experts co-authored in this paper. We leveraged the Phenoscape knowledge [28] base with manual annotations to procure the presence-absence trait matrix. For Bird-10K, we obtained the trait matrix from the attribute annotations provided along with CUB-200-

2011. We constructed approximately 380K question-answer pairs for trait identification tasks.

For grounding and referring VQA tasks, the ground truths were manually annotated with the help of expert biologists on our team. We manually annotated bounding boxes corresponding to the traits of 500 fish specimens and 500 bird specimens, which are subsets of the larger Fish-10K and Bird-10K datasets, respectively. We used the CVAT tool [111] for annotation. The task-specific question formats with the default prompts are provided in Section C.9.

C.2 Links to Access the Dataset and Its Metadata

We provide a GitHub link <https://github.com/imageomics/VLM4Bio> and an accessible Hugging Face link <https://huggingface.co/datasets/imageomics/VLM4Bio> to access the dataset and its metadata.

C.3 Dataset Availability and Maintenance

The VLM4Bio dataset and metadata are available in a Hugging Face repository. To access the VLM4Bio dataset, please visit <https://huggingface.co/datasets/imageomics/VLM4Bio>. Long-term support and maintenance of the dataset will be provided by our team. We have published a code repository for dataset preprocessing, including tasks such as downloading the dataset, reading images and metadata, cropping images, and running the evaluation experiments presented in the VLM4Bio paper. To access the VLM4Bio code repository, please visit <https://github.com/imageomics/VLM4Bio>.

| Statistics | Datasets | | | | | | | | | | | | | | |
|----------------|-----------------|-----------------|----------------------|-----------------|-----------------|------------------|--------------------|------------------|--------------------|-----------------------|-------------------------|-----------------------|-----------------------|-----------------------|----------------------------|
| | <i>Fish-10K</i> | <i>Bird-10K</i> | <i>Butterfly-10K</i> | <i>Fish-500</i> | <i>Bird-500</i> | <i>Fish-Easy</i> | <i>Fish-Medium</i> | <i>Bird-Easy</i> | <i>Bird-Medium</i> | <i>Butterfly-Easy</i> | <i>Butterfly-Medium</i> | <i>Butterfly-Hard</i> | <i>Fish-Prompting</i> | <i>Bird-Prompting</i> | <i>Butterfly-Prompting</i> |
| Images | 10,347 | 11,092 | 10,013 | 500 | 492 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 500 | 500 | 500 |
| Species | 495 | 188 | 60 | 60 | 47 | 51 | 10 | 50 | 10 | 50 | 10 | 1 | 25 | 37 | 25 |
| Genera | 178 | 114 | 27 | 18 | 33 | 10 | 1 | 10 | 1 | 10 | 1 | 1 | 12 | 30 | 10 |
| Traits | 10 | 28 | - | 8 | 5 | - | - | - | - | - | - | - | - | - | - |

Table C.1: Statistics of the VLM4Bio dataset.

C.4 Data Licenses

VLM4Bio dataset is licensed as [Creative Commons Attribution 4.0 International](#). The images of the corresponding organisms are licensed as follows:

1. Fish Dataset License: [CC BY-NC](#).
2. All the bird images are sourced from the CUB-200-2011 dataset; CalTech indicates that they do not own the copyrights to these images and that their use is restricted to non-commercial research and educational purposes.
3. Butterfly Dataset License: [Creative Commons Attribution 4.0 International](#).

We provide image-specific licenses in the dataset card <https://huggingface.co/datasets/imageomics/VLM4Bio#licensing-information>. We have hosted the dataset on HuggingFace (DOI: [10.57967/hf/3393](https://doi.org/10.57967/hf/3393)).

C.5 Data Distribution and Key Statistics

Table C.1 provides the key statistics for the datasets, including the number of images, species, genera, and traits present in each one. We are examining the Zero-shot accuracy of the VLMs on Fish-10K, Bird-10K, and Butterfly-10K for Species Classification and Trait Identification

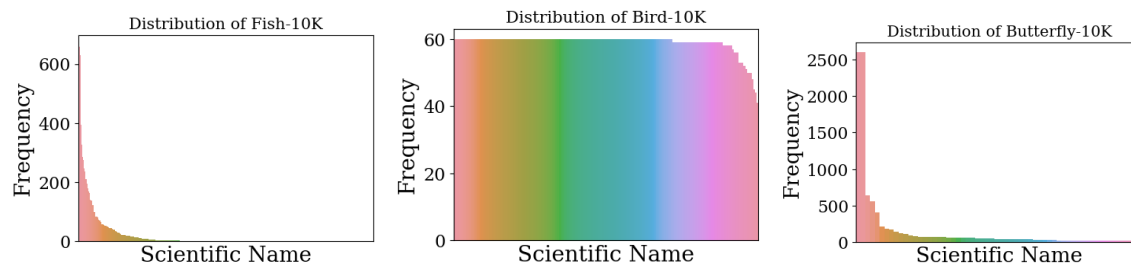


Figure C.2: Dataset Distribution of Fish-10K, Bird-10K, and Butterfly-10K.

tasks, Fish-500 and Bird-500 for Trait Grounding, Trait Referring and Trait Counting, and easy, medium, hard, prompting datasets for analyzing the role of answer choices, VLM reasoning and hallucination tests. From Figure C.2, it is clear that Fish-10K and Butterfly-10K are imbalanced, with a bias toward some species that are more common in our environment (such as *Heliconius erato* and *Heliconius melpomene* for Butterflies). The imbalance in Fish-10K and Butterfly-10K reflects the natural imbalance in the occurrence and observation of species in museum collections. Due to the scarcity of images for the rare species, it is difficult to increase their representation to avoid imbalance. As a result, we have included many under-represented species in the Fish and Butterfly datasets to report performance on the rare classes. In contrast, the Bird-10K dataset is well-balanced, with most species having 60 images. The easy, medium, hard, and prompting datasets are also balanced, which ensures a comprehensive evaluation of the zero-shot performance of the competing VLMs.

C.6 Traits Considered for the Task of Trait Identification

Figure C.3 shows the Fish traits and Bird traits used for evaluating the VLM’s performance in the identification task. For fishes, we considered 10 binary (presence/absence) traits which

| Fish Traits | | Bird Traits | | | |
|--|--|---|--|---|--|
| | | Color | | Pattern | Measurements |
| . Eye . Head . Mouth . Barbel . Dorsal fin | . Pectoral fin . Pelvic fin . Anal fin . Two dorsal fins . Adipose fin | . Bill-color . Crown-color . Eye-color . Forehead-color . Nape-color . Primary-color . Throat-color . Back-color | . Belly-color . Breast-color . Leg-color . Under-tail-color . Underparts-color . Upper-tail-color . Upperparts-color . Wing-color | . Head-pattern . Back-pattern . Breast-pattern . Wing-pattern . Tail-pattern . Belly-pattern | . Bill-length . Bill-shape . Shape . Size . Tail-shape . Wing-shape |

Figure C.3: Trait list for Trait Identification task.

include the *eye*, *head*, *mouth*, *barrel*, *dorsal fin*, *pectoral fin*, *pelvic fin*, *anal fin*, and *adipose fin*. We generated MC questions for the presence of each trait in an image (with two options: yes or no). Whereas for birds, we considered 28 traits covering their color, pattern, and measurements (size and shape of regions) in a multiple-choice format.

C.7 Traits Considered for the Tasks of Trait Grounding and Referring

To evaluate the VLM performance in Grounding and Referring, we identified 8 traits for fish and 5 traits for birds. Specifically, we manually annotated the *dorsal fin*, *adipose fin*, *caudal fin*, *anal fin*, *pelvic fin*, *pectoral fin*, *head*, and *eye* of the 500 fish specimens. Similarly, for birds, we annotated the *beak*, *head*, *eye*, *wings*, and *tail*. Trait grounding and referring tasks are carried out using the Fish-500 and Bird-500 datasets.

C.8 VLM Baselines

We consider the following VLM baselines to evaluate the performance on VLM4Bio dataset:

(1) GPT-4V(ision) [99], which is a proprietary VLM from OpenAI, that uses a generative pre-trained transformer model capable of understanding and generating both text and visual contents, (2) LLaVA-v1.5 (7B/13B) [73], which builds on top of the Vicuna LLM [18] by linearly projecting the visual embedding into the word embedding space. The LLaVA model has two different variants with 7B and 13B parameters, respectively, that depend on the size of the base Vicuna model, (3) COG-VLM [134], which performs a simple concatenation of the image and the text modalities, and uses trainable visual layers in the text-based transformer blocks, (4) MiniGPT-4 (Vicuna 7B/13B) [152], which is similar to LLaVA as it is built on top of the Vicuna model and linearly projects the visual embeddings for better understanding. Similar to LLaVA, MiniGPT-4 is available in two variants depending on the type the base Vicuna model (Vicuna 7B/13B), (5) BLIP-FLAN-T5-XL/XXL [67], which utilizes an effective pre-training strategy that relies on bootstrapping from frozen-pretrained CLIP encoders and LLMS by using a querying transformer block (available as two variants: XL and XXL), and (6) Instruct-BLIP (Vicuna 7B/13B) [23], which performs finetuning on BLIP-2 with visual-instruction tuning data to improve zero-shot capabilities of BLIP-2 (available as two variants depending on the Vicuna model: Vicuna 7B/13B).

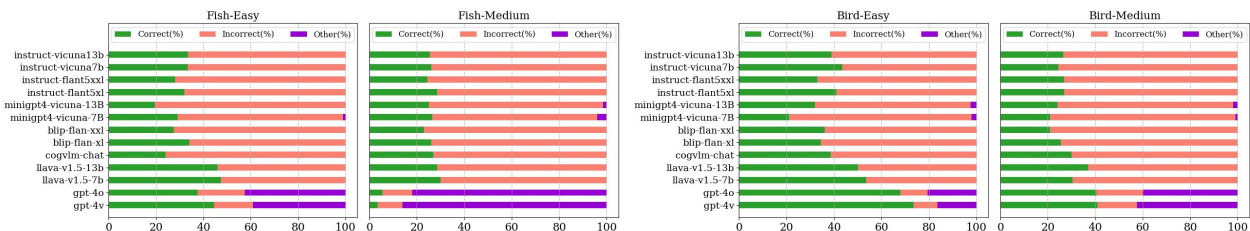
C.9 Prompts to Evaluate VLM performance

In order to ensure a fair comparison of the VLM responses to different types of questions in our dataset, we used the same question prompt for all the models across the various scientific tasks. It's worth noting that each model may perform differently with different

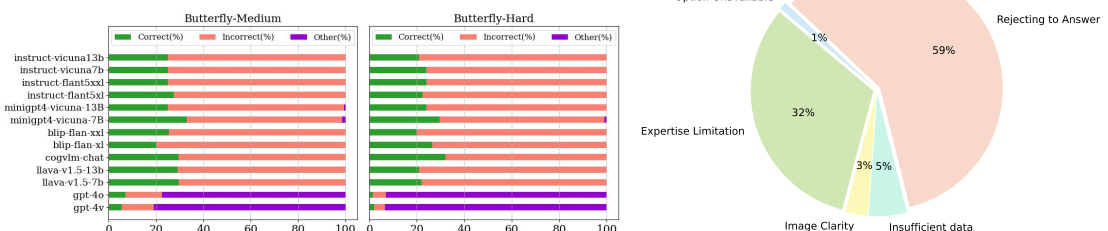
| Task | Prompt Format |
|--------------------------------------|--|
| Species Classification | <i><image></i> What is the scientific name of the <i><organism></i> shown in the image? <i><options></i> Write the answer after writing the answer is: . |
| Trait Identification | <i><image></i> Is there <i><trait></i> visible in the <i><organism></i> shown in the image? <i><options></i> Write the answer after writing the answer is: . |
| Trait Grounding | <i><image></i> What is the bounding box coordinates of the <i><trait></i> in the fish shown in the image? <i><options></i> Write the answer after writing the answer is: . |
| Trait Referring | <i><image></i> What is the trait of the <i><organism></i> that corresponds to the bounding box region <i><coordinates></i> in the image? <i><options></i> Write the answer after writing the answer is: . |
| Trait Counting | <i><image></i> How many unique <i><trait></i> are visible in the <i><organism></i> shown in the image? <i><options></i> Write the answer after writing the answer is: . |
| Contextual Prompting | <i><image></i> Each biological species has a unique scientific name composed of two parts: the first for the genus and the second for the species within that genus. What is the scientific name of the <i><organism></i> shown in the image? <i><options></i> Write the answer after writing the answer is: . |
| Dense Caption Prompting | <i><image></i> <i><dense caption></i> . Use the above dense caption and the image to answer the following question. What is the scientific name of the <i><organism></i> shown in the image? <i><options></i> Write the answer after writing the answer is: . |
| Chain-of-Thought Prompting | <i><image></i> What is the scientific name of the <i><organism></i> shown in the image? <i><options></i> Please consider the following reasoning to formulate your answer. <i><reasoning></i> . Write the answer after writing the answer is: . |
| False Confidence Test (FCT) | <i><image></i> What is the scientific name of the <i><organism></i> shown in the image? <i><options></i> Chosen Answer: <i><suggested answer></i> . Please provide: 1) Whether the chosen answer is correct (True/False). 2) The correct answer. |
| None of the Above Test (NOTA) | <i><image></i> What is the scientific name of the <i><organism></i> shown in the image? <i><options: A) _ B) _ C) _ D) None of the above.></i> Write the answer after writing the answer is: . |

Figure C.4: Prompts Templates used for Evaluation. There will be no *<options>* for Open set questions.

prompts. However, for the sake of simplicity in our evaluation, we opted for a consistent prompt for all the models. The prompts specific to each task are displayed in Figure C.4.



(a) Error Analysis for Fish-Easy and -Medium. (b) Error Analysis for Bird-Easy and -Medium.



(c) Error Analysis for Butterfly-Medium and -Hard. (d) Categories for 250 annotated GPT-4V "Other" responses.

Figure C.5: Analysis of errors for the pretrained VLM responses.

C.10 Error Analyses for VLM Responses

We categorize the VLM responses into 3 categories: (1) *Correct (%)*: where the scientific name is accurately predicted, (2) *Incorrect (%)*: where the scientific name is incorrect, and (3) *Other (%)*: a special category for instances where the model abstains from providing a scientific name.

Figure C.5a, C.5b and C.5c show the distribution of errors of different VLMs on Fish-Easy and Fish-Medium, Bird-Easy and Bird-Medium, and Butterfly-Medium and Butterfly-Hard datasets respectively using stacked-bar plots showing the three categories of VLM predictions. GPT-4V, for instance, shows a reduced rate of incorrect responses but a higher incidence of "Other" responses for these datasets, which include apologetic expressions, admissions of inability to precisely visualize the organism, and disclaimers regarding prediction without sufficient expert data and guidance.

To further analyze the type of errors happening in the other (%) category of VLM predictions, we manually examined 250 randomly selected “Other” GPT-4V responses for the task of fish species classification (MC question type) to generate the pie-chart of error categories shown in Figure C.5d. We can see that a majority of the “Other” responses belong to the category: *Rejecting to Answer* (59%), where the GPT-4V states that it is unable to provide an answer, sometimes stating the reason that it cannot answer based on a single image. We also observe a large fraction of *Expertise Limitation* responses where GPT-4V states that an expert taxonomist is needed to answer the question and its capabilities do not include recognizing or confirming species based on visual data. The next major type of “Other” responses are *Insufficient Data*, where GPT-4V states that it requires additional data to answer the question, e.g., taxonomic information or habitat information. The other error categories include *Image Clarity* issues and *Option Unavailable* (i.e., GPT-4V could not find a suitable option from the list of options provided in the prompt).

C.11 Comparing Pre-trained VLMs with a Biologically Fine-tuned Model

We compare the large pretrained VLMs and BioCLIP [119], a state-of-the-art foundation model for species classification. Furthermore, we include the simple CLIP model pretrained with OpenAI weights [105] to evaluate the zero-shot classification performance. Our evaluation was carried out on the Fish-10K, Bird-10K, and Butterfly-10K datasets, and the results are presented in Table C.2. We can see that BioCLIP significantly outperforms large pretrained VLMs on the Bird-10K and Butterfly-10K datasets, suggesting that BioCLIP may have been trained on images that are similar to the organisms present in these datasets. However, as noted in the paper, BioCLIP is not trained on fish images, and hence, the per-

| Dataset | Question type | Models | | | | |
|-------------------------------|---------------|---------------|----------------------|--------------------|-------------|----------------|
| | | <i>gpt-4v</i> | <i>llava v1.5-7b</i> | <i>cogvlm chat</i> | <i>CLIP</i> | <i>BioCLIP</i> |
| Species Classification | | | | | | |
| Fish-10K | Open | 1.01 | 2.32 | 0.11 | 0.57 | 1.24 |
| | MC | 35.91 | 40.20 | 31.72 | 42.45 | 50.65 |
| Bird-10K | Open | 17.40 | 1.45 | 0.86 | 7.74 | 67.12 |
| | MC | 82.58 | 50.32 | 44.73 | 45.78 | 93.93 |
| Butterfly-10K | Open | 0.04 | 0.05 | 0.01 | 5.33 | 15.95 |
| | MC | 28.91 | 50.24 | 36.45 | 45.60 | 62.32 |

Table C.2: Zero-shot accuracy comparison of VLM baselines (in % ranging from 0 to 100) with BioCLIP for the species classification task. Results are color-coded as **Best**, and **Worst**.

formance of large VLMS is similar to that of BioCLIP on Fish-10K images. We can also see that despite BioCLIP’s ability to effectively select the correct scientific name from a smaller set of options in multiple-choice (MC) questions, its performance significantly declines when asked to choose the scientific name from a larger set of open questions. From our observation, it is noteworthy that fine-tuning biological images with scientific names can help improve the overall accuracy of species classification, suggesting directions for future research in this area.

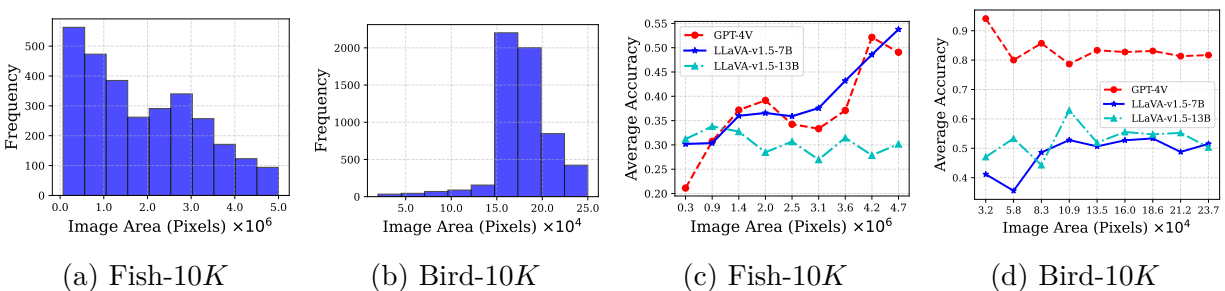


Figure C.6: Distribution of image resolutions for Fish-10K and Bird-10K are shown in Figures (a) and (b), respectively. The average score over image resolution for the GPT-4V, LLaVA-v1.5-7B, and LLaVA-v1.5-13B models on Fish-10K and Bird-10K are presented in Figures (c) and (d). We conduct the experiment in the context of the Species Classification task with Multiple-Choice (MC) questions.

C.12 Analyzing Effects of Image Resolution on VLM Performance

To investigate the effect of image resolution on VLM performance, we perform additional experiments summarized in Figure C.6 of the attached pdf. In this Figure, we show distribution plots for the Fish-10K and Bird-10K datasets with variations in the image resolutions and their impact on the species classification performance (MC question format) for GPT-4V, LLaVA-1.5-7B, and LLaVA-1.5-13B. All the images of the Butterfly-10K have the exact resolution (500×333); hence, they were not included in the experiment. From Figure C.6c, it is clear that image resolution is influential on the VLM performance for the Fish-10K dataset since higher resolution helps in recognizing the details of the biological traits and correct species. However, for Figure C.6d, the VLM performances do not vary significantly with the image resolution for the Bird-10K dataset. A potential reason is that the bird dataset is a subset of the CUB dataset, and we hypothesize that the pre-trained VLMs may have seen images with resolutions similar to those in the Bird-10K dataset during training, leading to this behavior.

C.13 Case Studies for Effects of Prompting on VLM Performance

C.13.1 No Prompting

1. No Prompting. GPT-4o Correct prediction. Refer to Figure C.7.
2. No Prompting. GPT-4o Incorrect prediction. Refer to Figure C.8.

3. No Prompting. COG-VLM Correct prediction. Refer to Figure [C.9](#).
4. No Prompting. COG-VLM Incorrect prediction. Refer to Figure [C.10](#).

C.13.2 Contextual Prompting

1. Contextual Prompting. GPT-4o Correct prediction. Refer to Figure [C.11](#).
2. Contextual Prompting. GPT-4o Incorrect prediction. Refer to Figure [C.12](#).
3. Contextual Prompting. LLaVa-13B Correct prediction. Refer to Figure [C.13](#).
4. Contextual Prompting. LLaVa-13B Incorrect prediction. Refer to Figure [C.14](#).

C.13.3 Dense Caption

1. Dense Captions in Prompts. GPT-4o Correct prediction. Refer to Figure [C.15](#).
2. Dense Captions in Prompts. GPT-4o Incorrect prediction. Refer to Figure [C.16](#).

C.13.4 Chain-Of-Thought Prompting

1. Chain-Of-Thought Prompting. GPT-4o Correct prediction. Refer to Figure [C.17](#).
2. Chain-Of-Thought Prompting. GPT-4o Incorrect prediction. Refer to Figure [C.18](#).
3. Chain-Of-Thought Prompting. LLaVa-13B Correct prediction. Refer to Figure [C.19](#).
4. Chain-Of-Thought Prompting. LLaVa-13B Incorrect prediction. Refer to Figure [C.20](#).

C.14 Case Studies for Reasoning Hallucination Tests


C.14.1 False Confidence Test (FCT)

1. FCT test on Fish dataset. GPT-4o Correct prediction. Refer to Figure [C.21](#).
2. FCT test on Fish dataset. LLaVa-13B Incorrect prediction. Refer to Figure [C.22](#).
3. FCT test on Bird dataset. GPT-4o Correct prediction. Refer to Figure [C.23](#).
4. FCT test on Bird dataset. LLaVa-13B Incorrect prediction. Refer to Figure [C.24](#).
5. FCT test on Butterfly dataset. GPT-4o Correct prediction. Refer to Figure [C.25](#).
6. FCT test on Butterfly dataset. LLaVa-13B Incorrect prediction. Refer to Figure [C.26](#).

C.14.2 None of The Above (NOTA) Test

1. NOTA test on Fish dataset. GPT-4o Correct prediction. Actual species name is *Esox Americanus*. Refer to Figure [C.27](#).
2. NOTA test on Fish dataset. LLaVa-13B Incorrect prediction. Actual species name is *Esox Americanus*. Refer to Figure [C.28](#).
3. NOTA test on Bird dataset. GPT-4o Correct prediction. Actual species name is *Corvus Albicollis*. Refer to Figure [C.29](#).
4. NOTA test on Bird dataset. Blip-Flan-XL Incorrect prediction. Actual species name is *Corvus Albicollis*. Refer to Figure [C.30](#).
5. NOTA test on Butterfly dataset. GPT-4o Incorrect prediction. Actual species name is *Batesia Hypochlora*. Refer to Figure [C.31](#).


6. NOTA test on Butterfly dataset. Blip-Flan-XL Correct prediction. Actual species name is *Batesia Hypochlora*. Refer to Figure C.32.



Question: What is the scientific name of the fish in the image?

Options: A) *Cyprinella spiloptera*, B) *Phenacobius mirabilis*, C) *Notropis hudsonius*, D) *Notropis buccatus*.


Write the answer after writing 'The answer is: '



Answer:
The answer is: B) *Phenacobius mirabilis*

Ground Truth: *Phenacobius mirabilis*


Figure C.7: No Prompting. GPT-4o Correct prediction. Section C.13.1.



Question: What is the scientific name of the fish in the image?

Options: A) *Esox lucius*, B) *Perca flavescens*, C) *Notropis hudsonius*, D) *Phenacobius mirabilis*.


Write the answer after writing 'The answer is: '



Answer:
The answer is: C) *Notropis hudsonius*

Ground Truth: *Phenacobius mirabilis*


Figure C.8: No Prompting. GPT-4o Incorrect prediction. Section C.13.1.



Question: What is the scientific name of the fish in the image?

Options: A) *Notropis hudsonius*, B) *Notropis blennioides*, C) *Noturus flavus*

Write the answer after writing 'The answer is: '




Answer:

The answer is: *Notropis hudsonius*

Ground Truth: *Notropis hudsonius*


Figure C.9: No Prompting. COG-VLM Correct prediction. Section C.13.1.



Question: What is the scientific name of the fish in the image?

Options: A) *Notropis buccatus*, B) *Lepomis humilis*, C) *Notropis dorsalis*, D) *Esox lucius*.

Write the answer after writing 'The answer is: '



Answer:

The scientific name of the fish in the image is *Notropis buccatus*.


Ground Truth: *Lepomis humilis*

Figure C.10: No Prompting. COG-VLM Incorrect prediction. Section C.13.1.

Question: Each biological species has a unique scientific name composed of two parts: the first for the genus and the second for the species within that genus. What is the scientific name of the fish in the image?

Options: A) *Lepomis cyanellus*, B) *Notropis blennioides*, C) *Esox lucius*, D) *Gambusia affinis*.

Write the answer after writing 'The answer is: '.



Answer:
The answer is: *Esox lucius*

Ground Truth: *Esox lucius*


✓

Figure C.11: Contextual Prompting. GPT-4o Correct prediction. Section C.13.2.

Question: Each biological species has a unique scientific name composed of two parts: the first for the genus and the second for the species within that genus. What is the scientific name of the fish in the image?

Options: A) *Phenacobius mirabilis*, B) *Notropis atherinoides*, C) *Lepomis humilis*, D) *Lepomis macrochirus*.

Write the answer after writing 'The answer is: '.




Answer:
The answer is: *Lepomis humilis*.

Ground Truth: *Lepomis macrochirus*

✗


Figure C.12: Contextual Prompting. GPT-4o Incorrect prediction. Section C.13.2.



Question: Each biological species has a unique scientific name composed of two parts: the first for the genus and the second for the species within that genus.
What is the scientific name of the fish in the image?

Options: A) *Esox americanus*, B) *Notropis atherinoides*, C) *Morone chrysops*, D) *Cyprinella spiloptera*.


Write the answer after writing 'The answer is: '.



Answer:
The answer is: C) *Morone chrysops*

Ground Truth: *Morone chrysops*


Figure C.13: Contextual Prompting. LLaVa-13B Correct prediction. Section [C.13.2](#).



Question: Each biological species has a unique scientific name composed of two parts: the first for the genus and the second for the species within that genus.
What is the scientific name of the fish in the image?

Options: A) *Cottus bairdii*, B) *Notropis blennioides*, C) *Fundulus catenatus*, D) *Noturus flavus*.


Write the answer after writing 'The answer is: '.



Answer:
The answer is: D) *Noturus flavus*

Ground Truth: *Fundulus catenatus*

Figure C.14: Contextual Prompting. LLaVa-13B Incorrect prediction. Section [C.13.2](#).



Dense Caption: The image showcases a small fish with a slender, elongated body shape. The body is covered in yellowish-brown hues, with a gradient that lightens towards the ventral side and deepens dorsally. The scales are distinctly visible, giving a textured appearance to the body. Starting from the head, the fish has a rounded snout and a relatively large, prominent eye with a dark pupil and a lighter surrounding area. The mouth is small and positioned slightly downturned. The dorsal fin is placed towards the rear of the body, exhibiting a rather elongated and low profile with delicate, translucent rays. The anal fin mirrors the dorsal fin but is placed closer to the rear ventral side, displaying similar morphological traits. The pectoral fins are modest in size, almost triangular, and located near the gill covers. The pelvic fins are also small and positioned ventrally, slightly behind the pectoral ones. The caudal fin is fan-shaped and extends symmetrically, marked by an array of thin rays. This fin appears semi-transparent with slight yellowish tinges that match the overall body coloration. Other distinctive features include a lateral line that is faintly visible and runs along the midline of the body. The skin reveals an intricate network of fine lines, which might suggest sensory functions in identifying environmental changes. Given these detailed anatomical descriptions, the fish could potentially be identified as a member of the Poeciliidae family, possibly a species of *Gambusia* or *Heterandria*.

Question: Use the above dense-caption and the image to answer the following question. What is the scientific name of the fish in the image?
Options: A) *Notropis volucellus*, B) *Notropis percobromus*, C) *Gambusia affinis*, D) *Lepomis gibbosus*.


Therefore, the answer is:

Answer:

The detailed description and the image closely match the morphological characteristics of the fish species *Gambusia affinis*, which belongs to the Poeciliidae family. Given the options:


- A) *Notropis volucellus*
- B) *Notropis percobromus*
- C) *Gambusia affinis*
- D) *Lepomis gibbosus*

The scientific name of the fish in the image is:
 C) *Gambusia affinis*



Ground Truth: *Gambusia affinis*

Figure C.15: Dense Captions in Prompts. GPT-4o Correct prediction. Section C.13.3.



Dense Caption: The fish displayed has an elongated, cylindrical body adorned with a pattern of small, well-defined scales that are golden-brown on the dorsal side and a lighter, creamy hue on the ventral side. The head is slightly pointed with a terminal mouth, featuring a prominent, spherical eye located on either side. The dorsal fin, located slightly more than halfway down the back, is relatively small and rounded. The caudal fin at the tail is forked, matching the body's light coloration and adorned with fine, subtle spots. Pectoral fins are positioned just behind the gill covers and are almost translucent, slightly broader at the base and tapering to a point. The pelvic fins are minute and situated abdominally, while the anal fin mirrors the size and shape of the dorsal fin but is placed closer to the caudal peduncle. A closer inspection of the body reveals faint horizontal striping along the lateral line, a key feature in taxonomic differentiation. The absence of specialized pigment patterns, such as vibrant or contrasting spots, suggests a species adapted for camouflage in its environment. Distinctive anatomical features such as the body coloration pattern, shape, and position of fins, presence of lateral line markings, and general morphology are crucial for identifying the scientific classification of this fish species.


Question: Use the above dense-caption and the image to answer the following question. What is the scientific name of the fish in the image?
Options: A) *Notropis dorsalis*, B) *Lepomis megalotis*, C) *Notropis stramineus*, D) *Gambusia affinis*.

Therefore, the answer is:

Answer:

Based on the dense-caption provided and the image, the description matches the characteristics of the fish known as **Notropis stramineus**.

Therefore, the answer is: C) **Notropis stramineus**.



Ground Truth: Gambusia affinis

Figure C.16: Dense Captions in Prompts. GPT-4o Incorrect prediction. Section [C.13.3](#).



| | |
|---|--|
| <p>Question: What is the scientific name of the fish in the image?</p> <p>Options: A) <i>Notropis hudsonius</i>, B) <i>Fundulus catenatus</i>, C) <i>Notropis atherinoides</i>, D) <i>Cottus bairdii</i>.</p> <p>Please consider the following reasoning to formulate your answer:</p> <p>Reasoning: To identify the fish in the image, we need to compare its physical characteristics with the descriptions of the four given options.</p> <ol style="list-style-type: none"> 1. <i>Notropis hudsonius</i> (Spottail Shiner): <ul style="list-style-type: none"> - Small freshwater fish - Distinguished by a prominent spot at the base of the tail fin 2. <i>Fundulus catenatus</i> (Northern Studfish): <ul style="list-style-type: none"> - Commonly characterized by vertical bars along its body - Known for its elongated body and thick caudal peduncle 3. <i>Notropis atherinoides</i> (Emerald Shiner): <ul style="list-style-type: none"> - A slender, elongated body - Silver-green coloration with a more translucent appearance 4. <i>Cottus bairdii</i> (Mottled Sculpin): <ul style="list-style-type: none"> - Stocky body and broad, flattened head - Usually mottled or blotchy in appearance <p>Looking at the image:</p> <ul style="list-style-type: none"> - The fish is small and slender with no prominent spots at the base of the tail fin, which excludes <i>Notropis hudsonius</i>. - The fish does not have vertical bars along its body, nor does it have the elongated body typical of <i>Fundulus catenatus</i>. - The fish does not have a broad, flattened head or a mottled appearance, ruling out <i>Cottus bairdii</i>. - The fish is slender and elongated, fitting the description of <i>Notropis atherinoides</i>. <p>Based on the physical characteristics visible</p> | <div style="text-align: right;">  </div>  <p>Answer:</p> <p>The scientific name of the fish in the image is <i>Notropis atherinoides</i> (Option C), commonly known as Emerald Shiner. This identification follows the reasoning that the fish has a slender, elongated body with a silver-green coloration and a more translucent appearance. The absence of a prominent spot at the base of the tail fin, vertical bars along the body, and a broad, flattened head with a mottled appearance further support this conclusion.</p> <p style="text-align: center;">Ground Truth: <i>Notropis atherinoides</i></p> |
|---|--|


Figure C.17: Chain-Of-Thought Prompting. GPT-4o Correct prediction. Section C.13.4.


Question: What is the scientific name of the fish in the image?

Options: A) *Morone chrysops*, B) *Cottus bairdii*, C) *Gambusia affinis*, D) *Fundulus catenatus*.

Please consider the following reasoning to formulate your answer:

Reasoning: To identify the fish in the image, let's examine the provided options and characteristics of each species:
A) **Morone chrysops:** Commonly known as the White Bass, *Morone chrysops* is a larger fish known for a deep body and silver coloration with black stripes running horizontally. The pictured fish does not exhibit these features.
B) **Cottus bairdii:** Known as the Mottled Sculpin, *Cottus bairdii* typically has a more robust and slightly shorter body, with a large head and fan-like pectoral fins, none of which match the fish in the image.
C) **Gambusia affinis:** Commonly known as the Mosquitofish, *Gambusia affinis* is a small fish with a more streamlined body and a noticeable upturned mouth. This could match the fish in the image, but the scale and body texture appear different.
D) **Fundulus catenatus:** Known as the Northern Studfish, *Fundulus catenatus* typically has a more slender body with a distinctive pattern along its sides, which aligns well with the fish in the image.
The fish in the image appears to have the body characteristics, scale texture, and fin arrangement that match with *Fundulus catenatus*.
Thus, the correct scientific name of the fish in the image is: D) **Fundulus catenatus**.





Answer:

Based on the reasoning provided and the visual appearance of the fish in the image, the correct scientific name is:

Option D) **Fundulus catenatus**.

Ground Truth: *Gambusia affinis*


Figure C.18: Chain-Of-Thought Prompting. GPT-4o Incorrect prediction. Section C.13.4.


Question: What is the scientific name of the fish in the image?

Options: A) *Esox americanus*, B) *Gambusia affinis*, C) *Cottus bairdii*, D) *Notropis blennioides*.

Please consider the following reasoning to formulate your answer:

Reasoning: The scientific name of the fish in the image is A) *Esox americanus*.





Answer:

The scientific name of the fish in the image is A) *Esox americanus*.

Ground Truth: *Esox americanus*

Figure C.19: Chain-Of-Thought Prompting. LLaVa-13B Correct prediction. Section C.13.4.



| | |
|--|---|
| <p>Question: What is the scientific name of the fish in the image?</p> <p>Options: A) <i>Notropis hudsonius</i>, B) <i>Morone chrysops</i>, C) <i>Gambusia affinis</i>, D) <i>Notropis blennius</i>.</p> <p>Please consider the following reasoning to formulate your answer:</p> <p>Reasoning: The scientific name of the fish in the image is B) <i>Morone chrysops</i>.</p> | <div style="text-align: right; margin-bottom: 10px;">  </div> <div style="text-align: center; margin-bottom: 10px;">  </div> <div style="background-color: #e0f2f1; padding: 5px; margin-bottom: 5px;"> <p>Answer: B) <i>Morone chrysops</i></p> </div> <div style="background-color: #e0f2f1; padding: 5px;"> <p>Ground Truth: <i>Gambusia affinis</i></p> </div> |
|--|---|

Figure C.20: Chain-Of-Thought Prompting. LLaVa-13B Incorrect prediction. Section C.13.4.



| | |
|--|--|
| <p>Question: What is the scientific name of the fish in the image?</p> <p>Options: A) <i>Notropis blennius</i>, B) <i>Notropis dorsalis</i>, C) <i>Cottus bairdii</i>, D) <i>Phenacobius mirabilis</i>.</p> <p>Chosen Answer: <i>Notropis dorsalis</i>.</p> <p>Please provide: 1) Whether the chosen answer is correct (True/False). 2) The correct answer .</p> | <div style="text-align: right; margin-bottom: 10px;">  </div> <div style="text-align: center; margin-bottom: 10px;">  </div> <div style="background-color: #e0f2f1; padding: 5px; margin-bottom: 5px;"> <p>Answer: 1) The chosen answer is correct (True/False): False 2) The correct answer: <i>Notropis blennius</i>"</p> </div> <div style="background-color: #e0f2f1; padding: 5px;"> <p>Ground Truth: <i>Notropis blennius</i></p> </div> |
|--|--|

Figure C.21: GPT-4o correct prediction. Section C.14.1.



| | |
|--|---|
| <p>Question: What is the scientific name of the fish in the image?</p> <p>Options: A) <i>Lepomis macrochirus</i>, B) <i>Perca flavescens</i>, C) <i>Notropis volucellus</i>, D) <i>Notropis blennius</i>.</p> <p>Chosen Answer: <i>Notropis volucellus</i>.</p> <p>Please provide: 1) Whether the chosen answer is correct (True/False). 2) The correct answer .</p> | <div style="text-align: right;"></div>  <p>Answer:</p> <p>The correct answer is <i>Notropis volucellus</i>. The fish in the image is a long, slender fish with a yellowish color. It is swimming in a body of water, possibly a river or a lake.</p> <p>Ground Truth: <i>Notropis blennius</i></p> |
|--|---|

Figure C.22: LLaVa-13B incorrect prediction. Section C.14.1.



| | |
|---|--|
| <p>Question: What is the scientific name of the bird in the image?</p> <p>Options: A) <i>Antrorstomus carolinensis</i>, B) <i>Sitta carolinensis</i>, C) <i>Certhia americana</i>, D) <i>Chloroceryle americana</i>.</p> <p>Chosen Answer: <i>Sitta carolinensis</i></p> <p>Please provide: 1) Whether the chosen answer is correct (True/False). 2) The correct answer .</p> | <div style="text-align: right;"></div>  <p>Answer:</p> <p>1) The chosen answer is correct: False. 2) The correct answer: D) <i>Chloroceryle americana</i>. The bird in the image is the Green Kingfisher, which is scientifically named <i>Chloroceryle americana</i>.</p> <p>Ground Truth: <i>Chloroceryle americana</i></p> |
|---|--|

Figure C.23: GPT-4o Correct prediction. Section C.14.1.


| | |
|--|---|
| <p>Question: What is the scientific name of the bird in the image?</p> <p>Options: A) <i>Contopus sordidulus</i>, B) <i>Vireo olivaceus</i>, C) <i>Corvus albicollis</i>, D) <i>Chloroceryle americana</i>.</p> <p>Chosen Answer: Contopus sordidulus</p> <p>Please provide: 1) Whether the chosen answer is correct (True/False). 2) The correct answer .</p> |  |
| <p>Ground Truth: Chloroceryle americana</p> | <p>Answer:</p> <p>True. The scientific name of the bird in the image is Contopus sordidulus.</p> |

Figure C.24: LLaVa-13B Incorrect prediction. Section C.14.1.


| | |
|--|--|
| <p>Question: What is the scientific name of the butterfly in the image?</p> <p>Options: A) <i>Heliconius heurippa</i>, B) <i>Batesia hypochlora</i>, C) <i>Heliconius melpomene</i>, D) <i>Heliconius wallacei</i>.</p> <p>Chosen Answer: Heliconius heurippa</p> <p>Please provide: 1) Whether the chosen answer is correct (True/False). 2) The correct answer .</p> |  |
| <p>Ground Truth: Batesia hypochlora</p> | <p>Answer:</p> <p>1) The chosen answer is incorrect (False). 2) The correct answer is B) <i>Batesia hypochlora</i>.</p> |

Figure C.25: GPT-4o Correct prediction. Section C.14.1.


| | |
|--|---|
| <p>Question: What is the scientific name of the butterfly in the image?</p> <p>Options: A) <i>Pyrrhogyra edocia</i>, B) <i>Heliconius timareta</i>, C) <i>Heliconius doris</i>, D) <i>Batesia hypochlora</i>.</p> <p>Chosen Answer: <i>Heliconius doris</i></p> <p>Please provide: 1) Whether the chosen answer is correct (True/False). 2) The correct answer .</p> |  |
| <p>Ground Truth: <i>Batesia hypochlora</i></p> | <p>Answer:</p> <ol style="list-style-type: none"> False The correct answer is B) <i>Heliconius timareta</i>. |

Figure C.26: LLaVa-13B Incorrect prediction. Section C.14.1.


| | |
|---|--|
| <p>Question: What is the scientific name of the fish in the image?</p> <p>Options: A) <i>Gambusia affinis</i>, B) <i>Noturus gyrinus</i>, C) <i>Phenacobius mirabilis</i>, D) None of the Above.</p> <p>Write the answer after writing 'The answer is: '</p> |  |
| <p>Ground Truth: None of the Above</p> | <p>Answer:</p> <p>The answer is: D) None of the Above</p> |

Figure C.27: GPT-4o Correct prediction. Actual species name is *Esox Americanus*. Section C.14.2.



| | |
|--|--|
| <p>Question: What is the scientific name of the fish in the image?</p> <p>Options: A) <i>Notropis volucellus</i>, B) <i>Notropis buccatus</i>, C) <i>Morone chrysops</i>, D) None of the Above.</p> <p>Write the answer after writing 'The answer is: '</p> | <div style="text-align: right; margin-bottom: 10px;">  </div> <div style="text-align: center;">  </div> <p>Answer:</p> <p>The answer is: A) <i>Notropis volucellus</i></p> <p>Ground Truth: None of the Above</p> |
|--|--|

Figure C.28: LLaVa-13B Incorrect prediction. Actual species name is *Esox Americanus*. Section [C.14.2](#).



| | |
|---|--|
| <p>Question: What is the scientific name of the bird in the image?</p> <p>Options: A) <i>Spizella passerina</i>, B) <i>Antrostomus carolinensis</i>, C) <i>Aethia pusilla</i>, D) None of the Above.</p> <p>Write the answer after writing 'The answer is: '</p> | <div style="text-align: right; margin-bottom: 10px;">  </div> <div style="text-align: center;">  </div> <p>Answer:</p> <p>The answer is: D) None of the Above.</p> <p>Ground Truth: None of the Above</p> |
|---|--|

Figure C.29: GPT-4o Correct prediction. Actual species name is *Corvus Albicollis*. Section [C.14.2](#).


| | |
|--|--|
| <p>Question: What is the scientific name of the bird in the image?</p> <p>Options: A) <i>Cardellina pusilla</i>, B) <i>Thryothorus ludovicianus</i>, C) <i>Passer domesticus</i>, D) None of the Above.</p> <p>Write the answer after writing 'The answer is: '</p> |  |
| <p>Ground Truth: None of the Above</p> | <p>Answer:</p> <p>C</p> |

Figure C.30: Blip-Flan-XL Incorrect prediction. Actual species name is *Corvus Albicollis*. Section C.14.2.


| | |
|--|--|
| <p>Question: What is the scientific name of the butterfly in the image?</p> <p>Options: A) <i>Temenis laothoe</i>, B) <i>Heliconius doris</i>, C) <i>Heliconius heurippa</i>, D) None of the Above.</p> <p>Write the answer after writing 'The answer is: '</p> |  |
| <p>Ground Truth: None of the Above</p> | <p>Answer:</p> <p>The answer is: A) <i>Temenis laothoe</i></p> |

Figure C.31: GPT-4o Incorrect prediction. Actual species name is *Batesia Hypochlora*. Section C.14.2.


| | |
|--|---|
| <p>Question: What is the scientific name of the butterfly in the image?</p> <p>Options: A) <i>Heliconius erato</i>, B) <i>Heliconius melpomene</i>, C) <i>Heliconius eleuchia</i>, D) None of the Above.</p> <p>Write the answer after writing 'The answer is: '.</p> |  |
| <p>Ground Truth: None of the Above</p> | <p>Answer:</p> <p>D</p> |

Figure C.32: Blip-Flan-XL Correct prediction. Actual species name is *Batesia Hypochlora*. Section C.14.2.

Bibliography

- [1] Great Lakes Invasive Network Project (GLIN). <https://greatlakesinvasives.org/portal/index.php>.
- [2] JFBM Bell Atlas. 2022. <http://bellatlas.umn.edu/index.php>.
- [3] UMMZ University of Michigan Museum of Zoology, Division of Fishes. https://ipt.lsa.umich.edu/resource?r=ummz_fish.
- [4] FMNH Field Museum of Natural History (Zoology) Fish Collection. Field Museum. https://fmipt.fieldmuseum.org/ipt/resource?r=fmnh_fishes.
- [5] Biodiversity occurrence data published by: INHS Collections Data (accessed through the INHS Collections Data Portal, biocoll.inhs.illinois.edu/portal, 2024-06-04).
- [6] University of Wisconsin-Madison Zoological Museum - Fish. <http://zoology.wisc.edu/uwzm/>.
- [7] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alex Alemi. Watch your step: Learning graph embeddings through attention. *CoRR*, abs/1710.09599, 2017. URL <http://arxiv.org/abs/1710.09599>.
- [8] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *WWW*, pages 37–48, 2013.
- [9] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

- [10] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020.
- [11] Mohammadreza Armandpour, Patrick Ding, Jianhua Huang, and Xia Hu. Robust negative sampling for network embedding. In *AAAI*, volume 33, pages 3191–3198, 2019.
- [12] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.
- [13] Gabriel Cadamuro, Ran Gilad-Bachrach, and Xiaojin Zhu. Debugging machine learning models. In *ICML Workshop on Reliable Machine Learning in the Wild*, volume 103, 2016.
- [14] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *CIKM*, pages 891–900, 2015.
- [15] Jorge Casillas, Oscar Cordón, Francisco Herrera Triguero, and Luis Magdalena. *Interpretability issues in fuzzy modeling*, volume 128. Springer, 2013.
- [16] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.
- [17] Tianle Chen, Zheda Mai, Ruiwen Li, and Wei-lun Chao. Segment anything model (sam) enhanced pseudo labels for weakly supervised semantic segmentation. *arXiv preprint arXiv:2305.05803*, 2023.

- [18] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2023.
- [19] Junsuk Choe, Seungho Lee, and Hyunjung Shim. Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4256–4271, 2020.
- [20] Edith Cohen, Eran Halperin, Haim Kaplan, and Uri Zwick. Reachability and distance queries via 2-hop labels. *SIAM Journal on Computing*, 32(5):1338–1355, 2003.
- [21] Elijah Cole, Kimberly Wilber, Grant Van Horn, Xuan Yang, Marco Fornoni, Pietro Perona, Serge Belongie, Andrew Howard, and Oisin Mac Aodha. On label granularity and object localization. In *European Conference on Computer Vision*. Springer, 2022.
- [22] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [23] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.
- [24] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- [25] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

- [26] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*, 2021.
- [27] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [28] Richard C Edmunds, Baofeng Su, James P Balhoff, B Frank Eames, Wasila M Dahdul, Hilmar Lapp, John G Lundberg, Todd J Vision, Rex A Dunham, Paula M Mabee, et al. Phenoscope: identifying candidate genes for evolutionary phenotypes. *Molecular biology and evolution*, 33(1):13–24, 2015.
- [29] Mohannad Elhamod, Mridul Khurana, Harish Babu Manogaran, Josef C Uyeda, Meghan A Balk, Wasila Dahdul, Yasin Bakis, Henry L Bart Jr, Paula M Mabee, Hilmar Lapp, et al. Discovering novel biological traits from images using phylogeny-guided neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3966–3978, 2023.
- [30] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004.
- [31] fishair.org. Fish-air. *fishair.org*, 2022.
- [32] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. In *25th Annual Symposium on Foundations of Computer Science, 1984.*, pages 338–346, 1984.

- [33] L. Fu and J. Deng. Graph calculus: Scalable shortest path analytics for large social graphs through core net. In *WI-IAT*, volume 1, pages 417–424, 2013.
- [34] Alberto García-Durán and Mathias Niepert. Learning graph representations with embedding propagation. *CoRR*, abs/1710.03059, 2017. URL <http://arxiv.org/abs/1710.03059>.
- [35] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL <http://arxiv.org/abs/1704.01212>.
- [36] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [37] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a “right to explanation”. *AI magazine*, 38(3):50–57, 2017.
- [38] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.
- [39] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, pages 297–304, 2010.
- [40] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017.
- [41] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

- [42] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [43] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [44] Qibin Hou, PengTao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. *Advances in Neural Information Processing Systems*, 31, 2018.
- [45] David Houle and Daniela M Rossoni. Complexity, evolvability, and the process of adaptation. *Annual Review of Ecology, Evolution, and Systematics*, 53, 2022.
- [46] 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*, pages 6700–6709, 2019.
- [47] Yao Jiang, Xinyu Yan, Ge-Peng Ji, Keren Fu, Meijun Sun, Huan Xiong, Deng-Ping Fan, and Fahad Shahbaz Khan. Effectiveness assessment of recent large vision-language models. *arXiv preprint arXiv:2403.04306*, 2024.
- [48] Chris Jiggins and Ian Warren. Cambridge butterfly wing collection - Chris Jiggins 2001/2 broods batch 1, January 2019. URL <https://doi.org/10.5281/zenodo.2549524>.
- [49] Chris Jiggins and Ian Warren. Cambridge butterfly wing collection - Chris Jiggins 2001/2 broods batch 2, January 2019. URL <https://doi.org/10.5281/zenodo.2550097>.
- [50] Chris Jiggins, Gabriela Montejó-Kovacevich, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 3, May 2019. URL <https://doi.org/10.5281/zenodo.2682458>.

- [51] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.
- [52] Ruoming Jin, Yang Xiang, Ning Ruan, and David Fuhry. 3-hop: a high-compression indexing scheme for reachability query. In *SIGMOD*, pages 813–826, 2009.
- [53] Jefkine Kafuna. Backpropagation in convolutional neural networks, Sep 2016. URL <https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/>.
- [54] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL <http://arxiv.org/abs/1609.02907>.
- [55] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [56] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
- [57] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [58] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 695–711. Springer, 2016.

- [59] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*, 2017.
- [60] Christopher Lawrence and Elizabeth G. Campolongo. Heliconius collection (cambridge butterfly), 2024. URL https://huggingface.co/datasets/imageomics/Heliconius-Collection_Cambridge-Butterfly.
- [61] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.
- [62] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019.
- [63] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 34:27408–27421, 2021.
- [64] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4071–4080, 2021.
- [65] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

- [66] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.
- [67] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.
- [68] Kunpeng Li, Ziyang Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9215–9223, 2018.
- [69] Ruiwen Li, Zheda Mai, Zhibo Zhang, Jongseong Jang, and Scott Sanner. Transcam: Transformer attention-based cam refinement for weakly supervised semantic segmentation. *Journal of Visual Communication and Image Representation*, 92:103800, 2023.
- [70] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.
- [71] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [72] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.
- [73] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

- [74] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [75] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.
- [76] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.
- [77] Moritz D Lürig, Seth Donoughe, Erik I Svensson, Arthur Porto, and Masahito Tsuboi. Computer vision, machine learning, and the promise of phenomics in ecology and evolutionary biology. *Frontiers in Ecology and Evolution*, 9:642774, 2021.
- [78] Daly M and Johnson N. Ohio State University Fish Division (OSUM). Museum of Biological Diversity, The Ohio State University, February 2018. URL <https://doi.org/10.15468/subs18>.
- [79] Harish Babu Manogaran, M Maruf, Arka Daw, Kazi Sajeed Mehrab, Caleb Patrick Charpentier, Josef C Uyeda, Wasila Dahdul, Matthew J Thompson, Elizabeth G Campolongo, Kaiya L Provost, et al. What do you see in common? learning hierarchical prototypes over tree-of-life to discover evolutionary traits. *arXiv preprint arXiv:2409.02335*, 2024.
- [80] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of*

- the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.
- [81] Anniina Mattila, Chris Jiggins, and Ian Warren. University of Helsinki butterfly collection - Anniina Mattila bred specimens, February 2019. URL <https://doi.org/10.5281/zenodo.2555086>.
- [82] Joana I. Meier, Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jiggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild specimens batch 3, October 2020. URL <https://doi.org/10.5281/zenodo.4153502>.
- [83] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. *arXiv preprint arXiv:2311.12983*, 2023.
- [84] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [85] Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 1- version 2, May 2019. URL <https://doi.org/10.5281/zenodo.3082688>.
- [86] Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 2, May 2019. URL <https://doi.org/10.5281/zenodo.2677821>.
- [87] Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 4, May 2019. URL <https://doi.org/10.5281/zenodo.2682669>.

- [88] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, Camilo Salazar, Marianne Elias, Imogen Gavins, Eva Wiltshire, Stephen Montgomery, and Owen McMillan. Cambridge and collaborators butterfly wing collection batch 10, May 2019. URL <https://doi.org/10.5281/zenodo.2813153>.
- [89] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 7, May 2019. URL <https://doi.org/10.5281/zenodo.2702457>.
- [90] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 8, May 2019. URL <https://doi.org/10.5281/zenodo.2707828>.
- [91] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 5, May 2019. URL <https://doi.org/10.5281/zenodo.2684906>.
- [92] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 6, May 2019. URL <https://doi.org/10.5281/zenodo.2686762>.
- [93] Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, Eva Wiltshire, and Imogen Gavins. Cambridge butterfly wing collection batch 9, May 2019. URL <https://doi.org/10.5281/zenodo.2714333>.
- [94] Gabriela Montejo-Kovacevich, Eva van der Heijden, and Chris Jiggins. Cambridge butterfly collection - GMK Broods Ikiam 2018, November 2020. URL <https://doi.org/10.5281/zenodo.4291095>.
- [95] Gabriela Montejo-Kovacevich, Eva van der Heijden, Nicola Nadeau, and Chris Jiggins.

- Cambridge butterfly wing collection batch 10, November 2020. URL <https://doi.org/10.5281/zenodo.4289223>.
- [96] Gabriela Montejo-Kovacevich, Quentin Paynter, and Amin Ghane. *Heliconius erato cyrba*, Cook Islands (New Zealand) 2016, 2019, 2021, September 2021. URL <https://doi.org/10.5281/zenodo.5526257>.
- [97] OpenAI. Gpt-4v(ision) system card, 2023. *arXiv preprint arXiv:2303.08774*, 2023.
- [98] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [99] OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023.
- [100] OpenAI. Gpt-4o (“o” for “omni”). <https://openai.com/index/hello-gpt-4o/>, 2024.
- [101] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *KDD*, pages 1105–1114, 2016.
- [102] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710, 2014.
- [103] Erika Pinheiro de Castro, Christopher Jiggins, Karina Lucas da Silva-Brandão, Andre Victor Lucci Freitas, Marcio Zikan Cardoso, Eva Van Der Heijden, Joana Meier, and Ian Warren. Brazilian Butterflies Collected December 2020 to January 2021, February 2022. URL <https://doi.org/10.5281/zenodo.5561246>.
- [104] Haotong Qin, Ge-Peng Ji, Salman Khan, Deng-Ping Fan, Fahad Shahbaz Khan, and Luc Van Gool. How good is google bard’s visual understanding? an empirical study on open challenges, 2023.
- [105] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learn-

- ing transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [106] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67, 2020.
- [107] Camilo Salazar, Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Imogen Gavins. Camilo Salazar and Cambridge butterfly wing collection batch 1, May 2019. URL <https://doi.org/10.5281/zenodo.2735056>.
- [108] Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jiggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild and bred specimens batch 1, December 2018. URL <https://doi.org/10.5281/zenodo.1748277>.
- [109] Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jiggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild and bred specimens batch 2, January 2019. URL <https://doi.org/10.5281/zenodo.2548678>.
- [110] Patricio A. Salazar, Nicola Nadeau, Gabriela Montejo-Kovacevich, and Chris Jiggins. Sheffield butterfly wing collection - Patricio Salazar, Nicola Nadeau, Ikiam broods batch 1 and 2, November 2020. URL <https://doi.org/10.5281/zenodo.4288311>.
- [111] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, TOSmanov, Dmitry Kruchinin, Artyom Zankevich, DmitriySidnev, Maksim Markelov, Johannes222, Mathis Chenuet, a andre, telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya4607, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, vugia truong, zliang7, lizhming, and

- Tritin Truong. opencv/cvat: v1.1.0, August 2020. URL <https://doi.org/10.5281/zenodo.4009388>.
- [112] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.
- [113] Wataru Shimoda and Keiji Yanai. Distinct class-specific saliency maps for weakly supervised semantic segmentation. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 218–234. Springer, 2016.
- [114] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.
- [115] Tiago R Simões, Michael W Caldwell, Alessandro Palci, and Randall L Nydam. Giant taxon-character matrices: quality of character constructions remains critical regardless of size. *Cladistics*, 33(2):198–219, 2017.
- [116] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [117] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- [118] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

- [119] Samuel Stevens, Jiaman Wu, Matthew J Thompson, Elizabeth G Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila M Dahdul, Charles Stewart, Tanya Berger-Wolf, et al. Bioclip: A vision foundation model for the tree of life. *arXiv preprint arXiv:2311.18803*, 2023.
- [120] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.
- [121] Fengdong Sun and Wenhui Li. Saliency guided deep network for weakly-supervised image segmentation. *Pattern Recognition Letters*, 120:62–68, 2019.
- [122] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.
- [123] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.
- [124] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [125] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [126] Logesh Kumar Umapathi, Ankit Pal, and Malaikannan Sankarasubbu. Med-halt:

- Medical domain hallucination test for large language models. *arXiv preprint arXiv:2307.15343*, 2023.
- [127] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [128] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [129] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.
- [130] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [131] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset, 2011.
- [132] Fulton Wang and Cynthia Rudin. Causal falling rule lists. *arXiv preprint arXiv:1510.05189*, 2015.
- [133] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.
- [134] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

- [135] Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 2, February 2019. URL <https://doi.org/10.5281/zenodo.2553501>.
- [136] Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 1, February 2019. URL <https://doi.org/10.5281/zenodo.2552371>.
- [137] Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 3, February 2019. URL <https://doi.org/10.5281/zenodo.2553977>.
- [138] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.
- [139] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: Cross language image matching for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4483–4492, 2022.
- [140] Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. Lvlm-chub: A comprehensive evaluation benchmark for large vision-language models. *arXiv preprint arXiv:2306.09265*, 2023.
- [141] Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. Understanding negative sampling in graph representation learning. *arXiv preprint arXiv:2005.09863*, 2020.
- [142] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa.

- In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022.
- [143] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.
- [144] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *CoRR*, abs/1603.08861, 2016. URL <http://arxiv.org/abs/1603.08861>.
- [145] Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Lu Sheng, Lei Bai, Xiaoshui Huang, Zhiyong Wang, et al. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *arXiv preprint arXiv:2306.06687*, 2023.
- [146] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.
- [147] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.
- [148] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

- [149] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, and Lihe Zhang. Joint learning of saliency detection and weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7223–7233, 2019.
- [150] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.
- [151] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.
- [152] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.
- [153] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *CoRR*, abs/1707.04638, 2017. URL <http://arxiv.org/abs/1707.04638>.