1 Introduction

Accurate counting of items of interest within large image datasets is pivotal in fields ranging from ecology to medical imaging, where it enables essential quantitative analyses and supports empirical research. Traditionally, this task has involved training specialized detectors to segment images and identify relevant items. These detectors, however, require access to high-quality labeled training data and are often restricted to their specific training domains. While large-scale generalist models like the Segment Anything Model (Kirillov et al., 2023) offer a broadened scope by proposing to segment any visible object, they still necessitate a subsequent counting mechanism or further labeling to identify and count the items of interest.

As a compelling alternative, Vision-Language Models (VLMs) offer the direct use of natural language to query a corpus of images and identify ones with the presence of desired items. This capability not only simplifies the process of querying and counting specific objects within large datasets but also bypasses the limitations posed by domain-specific detectors. By leveraging natural language, VLMs facilitate a more intuitive and flexible approach to the counting problem. This research aims to explore augmenting the out-of-the-box performance of VLMs, primarily CLIP by OpenAI (Radford et al., 2021), in order to develop a flexible, query-based count estimator over large-scale, unlabeled (or sparsely labeled) image datasets across various domains.

In ecological and environmental studies, for instance, dynamic counting based on queries such as identifying images with a specific species is crucial for understanding and preserving biodiversity. Accurate image counts in datasets that span a geographical region or a particular series of time can inform studies on wildlife populations, habitat health, and ecological balance. Similarly, in healthcare, particularly in radiology and pathology, the method could estimate the count of images showing specific medical conditions. Furthermore, in the industrial sector, this method could be used to estimate the number of product images or fashion items fitting certain descriptions. For example, a retailer might query specific descriptions within their product database. These counts can help in inventory management, trend analysis, and customer de-

mand prediction. Overall, fields wherein the speed and accessibility of generating these counts is of importance, as opposed to complete text-based image retrieval, serve as the primary target of this research.

Consider the example of an ecology dataset where the task is to count images corresponding to the query "yellow beaked birds." We define a *true count* as one that represents the actual number of images that depict birds with yellow beaks, reflecting an exact correspondence to the query specifications. Achieving an exact true count or a *ground-truth* value for all relevant queries necessitates comprehensive and descriptive labeling of each image in the dataset.

This labeling process for large-scale modern datasets, which generally contain upwards of 10,000 images, requires significant human effort. As mentioned earlier, for our task, these labels must be exceptionally descriptive to cover any possible natural language query, adding to the complexity and labor intensity of the task. Despite such detailed labeling, it remains uncertain whether these annotations alone can suffice to return true counts for all possible queries using simple keyword matching or semantic similarity calculations. This motivates the use of AI models to generate estimates of true counts, which ideally are *unbiased* and accurate within a desired degree of error.

The prevailing approach to this challenge leverages the capabilities of VLMs, which are trained on extensive collections of image-text pairs. These models assess the presence or absence of specified objects, colors, or attributes in each image, relative to a given query. This method, akin to a binary classification within the framework of Visual Question Answering (VQA), provides a scalable solution to derive estimated counts across extensive datasets.

Solely relying on these models to generate these estimates, however, is not ideal as they lack a deeper domain knowledge for zero-shot image classification based on specific fine-grained details and complex reasoning queries. A common method to mitigate this is to perform *fine-tuning*, wherein a model is further trained on a specific set of labeled data pertaining to the objective at hand to improve its domain-specific task accuracy. For example, we could fine-tune CLIP on a labeled set of domain-specific images for it to perform better on our ecol-

ogy dataset. Fine-tuning, while having the potential of largely improving model accuracy, once again requires labeled data, hence the expenditure of significant human effort.

These factors motivate a *human-in-the-loop* approach to counting. Instead of spending significant human effort in labelling, manually identifying true matches, or screening outputs for all images, one can spot-check only a small subset and then employ statistical techniques to obtain *unbiased estimates* of true counts across the entire dataset.

Therefore, this research assesses the effectiveness of a VLM-based Importance sampling technique with human screening (that provides true counts for the samples) to produce unbiased estimates of object counts in images that match a proposed natural language query in large-scale datasets. The overarching goal is to examine the hypothesis that our statistical approach provides a significantly lower expenditure of human effort and obtains a comparable level of accuracy when compared to extensive fine-tuning or outright manual screening. The work is motivated by problems where improving the VLM requires serious effort, but counts from the model are likely correlated with true counts and can be used as a proposal distribution for our sampling.

Furthermore, there exist a number of secondary objectives and areas of inquiry. When performing human spot-checking, it is important to identify the minimum images required to be sampled in order to generate counts with a desired accuracy. This number is inversely proportional to the effectiveness of our statistical method. To determine success, we desire it to be a reasonably small fraction of the entire data. Identifying trends in minimum samples required to achieve desired estimates for different datasets is also of practical use, since it can potentially guide deployment in the real world.

Our work is largely influenced by Perez et al., 2024, a paper that introduces a Detector-Based Importance Sampling approach - DISCount - to generate count estimates catering to the image modality. We extend their research to a broader use case by allowing for user queries through VLMs, instead of using a computer vision model to detect and count only a specific item in the images. Furthermore, we propose an Adaptive method on top of DISCount that

allocates a portion of the sampled images to train a logistic regression model and iteratively update the importance sampling proposal distribution. We evaluate our method on a modified version of the Caltech 256 dataset (Griffin et al., 2022) and the Caltech-UCSD Birds (CUB) dataset (Wah et al., 2011), for direct class-level, broad category-level and finer attribute-level queries. Our results indicate that our method improves estimate error rates for classification and broad category-level queries by roughly $30-40\%$ on modified Caltech and $50-65\%$ on CUB. We also showcase the ineffectiveness of using DISCount for granular attribute-level queries by comparing it to a Monte-Carlo sampler and improve its performance through our adaptive approach by $10-20\%$ in most cases on CUB [1]. Furthermore, we demonstrate that our newly generated estimates have a lower variance, hence better confidence intervals.

The novelty of our study lies in its application of a multimodal AI model to generate a suitable proposal distribution for Importance sampling, alongside the incorporation of an Adaptive loop to augment DISCount. For this research the primary modality is vision and the secondary modality is chosen to be natural language, however, it intends to serve as a proof-of-concept for the same method to be replicated in image-audio, audio-text or other relevant multimodal contexts.

[1] Our demo is available on GitHub at advaitgosai/count_anything_demo

2 Related Work

Visual Question Answering (VQA) is a research area at the intersection of computer vision and natural language processing. The primary objective of VQA is to enable AI models to provide accurate answers to questions posed about visual content. Early works in this field focused on simplistic interpretations of images but gradually evolved to handle complex queries. Originally, the paper "VQA: Visual Question Answering" (Agrawal et al., 2016) laid the foundation for these tasks by introducing the concept of free-form and open-ended VQA. This involves providing an accurate natural language answer to a natural language question about a given image. It further presented a dataset containing approximately 250,000 images, 760,000 questions, and 10 million answers, offering a resource for developing and comparing VQA methods.

The paper "Yin and Yang: Balancing and Answering Binary Visual Questions" (Zhang et al., 2016) introduced a novel approach specifically for binary VQA, which is the scope of our research. It employed a two-step method: Language Parsing, where binary questions are converted into a tuple format <P R S> (Primary object, Relation, Secondary object), and Visual Verification, which aligns these tuple elements with corresponding objects in an image. This method was unique in its ability to handle complex queries, even when certain objects are not visible in the image. The study used two models, the Q-model and Tuple-model, which process different language features from the question but use the same image features, leading to a fused language-image representation for answering binary VQA tasks. It demonstrated that models trained on a balanced dataset, where language biases are minimized, outperform those trained on unbalanced datasets.

Other studies from the same time show similar results, wherein inherent structures in our world and bias in our language tended to be a simpler signal for learning than visual modalities. For example, the language model could directly leverage its statistical priors to answer questions such as "What is the color of the banana?", without having to rely on its vision capabilities. Goyal et al. (2017) argued that this phenomenon resulted in models that ignore visual information, leading to an inflated sense of their capability. To formally test this, they balanced

the original VQA dataset presented in Agrawal et al. 2016 by collecting complementary images such that every question is associated with a pair of complementary images that result in two different answers to the question. They released this dataset as VQA v2.0, finding that all state-of-the-art models performed significantly worse when tested on it, thus providing concrete evidence for their hypothesis (Goyal et al., 2017). VQA v2.0 ends up serving as a staple benchmark in the field, driving model development for the next few years.

In the early stages of VQA, models primarily relied on Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks, a type of recurrent neural network (RNN) capable of handling sequences of data. These LSTM-based models were adept at processing the sequential nature of language in questions. A typical architecture involved using a convolutional neural network (CNN) to extract features from images, and an LSTM to process the question. The image features and text features were then combined, often through simple concatenation or more complex interaction mechanisms, to predict the answer. This approach was seen in early works such as the paper "Exploring Models and Data for Image Question Answering" (Ren et al., 2015), which demonstrated the effectiveness of combining CNNs and LSTMs for basic VQA tasks.

Over time, as the limitations of LSTM-based models became apparent, especially in terms of handling more complex, nuanced questions and larger datasets, newer architectures were developed. This led to the adoption of attention mechanisms and Transformer models, which offered more flexibility and power in modeling the relationships between visual elements and textual queries. A significant milestone in this evolution was the introduction of the Transformer model in the paper "Attention Is All You Need" (Vaswani et al., 2017), which laid the foundation for subsequent developments in VQA.

The latest and one of the most influential advancements in VQA has been the development of CLIP (Contrastive Language–Image Pretraining) in the work "Learning Transferable Visual Models From Natural Language Supervision" (Radford et al., 2021). CLIP represents a paradigm shift in how models are trained and how they perform VQA tasks. It is trained on a

vast dataset of 400 million images and corresponding text captions crawled from the internet, learning to associate images with textual descriptions in a more generalized and robust manner. This pretraining allows CLIP to showcase strong zero shot capabilities on benchmarks such as ImageNet classification dataset that was first introduced in Deng et al., 2009.

Despite its groundbreaking zero shot performance on such vision-based tasks, for VQA specifically, using CLIP out-of-the-box was found to be ineffective as many such tasks require complex multimodal reasoning (Kim et al., 2021). This was eventually mitigated in the paper "How Much Can CLIP Benefit Vision-and-Language Tasks?" where the authors proposed to integrate CLIP with existing VQA models by simply replacing their visual encoder with CLIP's visual encoder, given its superior performance on image classification benchmarks (Shen et al., 2021). Their research became the then state-of-the-art for VQA and other Vision and Language tasks.

One of the techniques employed by the authors to achieve such performance was to fine-tune CLIP's vision encoder on task-specific data to improve its domain knowledge. Fine-tuning involves adjusting the pretrained model on a dataset tailored to the particular task, allowing the model to adapt its generalized learning to more specific contexts (Dai and Le, 2015). This refinement improves the model's understanding and accuracy within that domain. Further studies with CLIP such as Gao et al., 2021 confirmed these results with fine-tuning significantly improving performance for VQA tasks, boosting its popularity and modern relevance. However, it was also quickly seen that fine-tuning has its drawbacks, primarily the cost and effort involved in annotating task-specific datasets, which can be substantial. Moreover, fine-tuning also risks overfitting the model to the particular characteristics of the training data, potentially reducing its generalizability and robustness, a concern highlighted in Pham et al., 2021.

The architecture of CLIP has inspired modern multimodal Vision-Language Models (VLMs), which represent the current state of the art in VQA and other vision-language tasks. For example, Florence, introduced by Yuan et al., 2021, builds on the CLIP architecture to create a more robust and versatile model capable of handling a wide range of visual tasks, including VQA. It

integrates advanced features such as more extensive pretraining on diverse datasets and sophisticated cross-modal interaction mechanisms. Furthermore, recent advancements have also seen the integration of large language models (LLMs) into vision-language tasks, including VQA. LLMs, with their vast knowledge base and sophisticated text processing capabilities, complement vision-focused models by enhancing their understanding of complex language queries. This synergy of visual and textual understanding in models like GPT-4 (OpenAI, 2023), LLaVa (Liu et al., 2023) and CogVLM (Wang et al., 2023) extends the frontier of multimodal AI, enabling more nuanced and context-aware interpretations of visual data in relation to natural language queries, serving as contemporary VQA state-of-the-art.

Despite the impressive zero shot results obtained using VLMs, the issue of their performance being significantly less impressive in highly specialized or niche domains such as identifying animals of specific species, or distinguishing medical image scans with or without lesions is still persistent. Fine-tuning stands to offer a significant performance boost, but at the cost of further model training and domain-specific data curation, cleaning and labeling effort that needs to be replicated for each desired use case. This motivates the use of alternate approaches to fine-tuning, hence alternate methods to allocate human effort in order to boost model performance. For example Wah et al., 2014 show that even with a less accurate model, human-in-the-loop recognition and screening strategies reduce annotation costs and improve performance by integrating human validation with noisy predictions.

Such techniques find relevance when it comes to counting within large image collections, since they can be coupled with statistical methods to provide unbiased estimates that are accurate within certain confidence intervals. Such was the case in the paper "DISCOUNT: Counting in Large Image Collections with Detector-Based Importance Sampling" (Perez et al., 2024), whose results form a fundamental basis for ours. The paper contributes counting techniques for large image collections that build on IS-count, a previously established covariate-based method (Meng et al., 2021) to count specific items of interest within a set of images. Their setup where it is possible to train a detector to run on all images, but the detector is not reliable enough for the final counting task, or its reliability is unknown is similar to the one for our application where

we replace the detector with a VLM. Their primary contribution is that of devising human-in-the-loop methods for count estimation using the detector to construct a proposal distribution for sampling. Furthermore, they prove the conditional unbiasedness of this novel sampling approach, and design confidence intervals, which are important practically to know how much human effort is needed. We directly borrow these findings for our multimodal application.

Mathematically, each of their findings can be expressed and applied to our case as follows. Given an approximate count - 1 or 0 (yes or no) - $g(s)$ for each sample image s in an entire dataset S (instead of regions as defined in Perez et al., 2024), produced by a trained detection model, the total approximate count for S is $G(S) = \sum_{s \in S} g(s)$. The DISCount estimator uses a proposal distribution proportional to g on dataset S, defined as:

$$\bar{g}_S(s) = \frac{g(s) I[s \in S]}{G(S)} \tag{1}$$

The Importance sampling estimator is then expressed as:

$$\hat{F}_{\text{DIS}}(S) = G(S) \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{f(s_i)}{g(s_i)}, \quad s_i \sim \bar{g}_S \tag{2}$$

where $f(s_i)$ represents the true count for the i^{th} sample. The ratio $w_i = \frac{f(s_i)}{g(s_i)}$ serves as the importance weight, and DISCount reweights the detector-based total count $G(S)$ by the average weight $\bar{w}$, acting as a correction factor for over- or under-counting.

Confidence intervals for the DISCount estimator are derived similarly to standard importance sampling. The importance weight variance $\sigma^2(S)$ is estimated first. An approximate $1-\alpha$ confidence interval is then:

$$\hat{F}_{DIS}(S) \pm z_{\alpha/2} \cdot \frac{G(S) \cdot \sigma(S)}{\sqrt{n(S)}} \tag{3}$$

Here, z_γ represents the $1 - \gamma$ quantile of the standard normal distribution. This approach enables constructing confidence intervals around count estimates, enhancing the reliability of and practical use of our analysis.

In order to assess the effectiveness the statistical counting described by DISCount for our task, our research could also benefit by performing preliminary active testing to identify the best VLM to use. Active testing in model evaluation involves training a surrogate model and implementing an acquisition proposal, which is a strategy for selecting the most informative test points for labeling. This process iteratively selects test data points based on their expected contribution to the model's learning, observes their labels, and updates the surrogate model accordingly. The final step involves computing the test loss estimate, thus providing a sample-efficient approach to model evaluation by prioritizing test points that are most impactful for the loss estimate (Kossen et al., 2021).

3 Methods

The two datasets chosen for evaluating a DISCount-based approach to generate estimated counts based on the VQA task include the entire Caltech-UCSD Birds-200-2011 (CUB) dataset (Wah et al., 2011) and a modified version of the Caltech 256 image dataset (Griffin et al., 2022) that we name as modified Caltech.

Since our scope is that of answering arbitrary natural language queries, generating ground-truth labels is not as simple as assigning a static true count (1 or 0) for each image. For each image, it was necessary to use the dataset annotations, class labels, and any relevant metadata to provide an elaborate description of its visual scene to eventually determine a set of counts each corresponding to a specific predetermined query. For CUB, a large corpus of annotations is provided with the dataset. These annotations include detailed descriptions of 312 attributes, encompassing a variety of features such as color patterns, beak shapes, and sizes. Each of the 200 species (classes) is represented by approximately 60 images, allowing for diverse queries regarding the appearance and attributes of birds. This detailed annotation facilitated the generation of ground-truth counts for texture, pattern and shape-based queries such as inquiring for birds with red feathers or a rounded bill.

On the other hand, the Caltech 256 dataset does not come with annotations beyond its class names. Hence, the dataset was modified to aid manual ground truth generation for the feature and attribute-based queries. The modifications included reducing the dataset size to 128 classes with approximately 30 images randomly sampled from each class. The selected subset was further annotated manually to include category-level details about the image items and identifiable object parts. The images were classified in a total of 10 high-level classes such as ‘animal’ or ‘sporting equipment’, with the 128 original classes (per human discretion) being matched to the new ones. This dataset - modified Caltech, provided a baseline to assess performance on broader, semantic visual question answering.

Within each dataset, our ground truth function F assigns a binary label to each image i,

indicating the presence (1) or absence (0) of the queried text:

$$F(i) = \begin{cases} 1 & \text{if queried text corresponds to image } i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

To embed the datasets and queries, CLIP with the Vision Transformer (ViT-L/14) architecture (Radford et al., 2021) was the chosen VLM. Both CUB and modified Caltech are embedded using this model, producing a feature vector v_i for each image within its respective dataset. For each query, a corresponding textual embedding t is similarly generated using CLIP. [2] The CLIP model and its tokenizer were obtained using the open-source Hugging Face Sentence Transformers library. [3]

3.1 DISCount using CLIP

We calculate the cosine similarity (s) between the query embedding (t) and each image embedding (v_i) to serve as a similarity score, a measure of how closely CLIP represents the content of each image with the queried text:

$$s_i = \frac{\mathbf{t} \cdot \mathbf{v}_i}{\|\mathbf{t}\|\|\mathbf{v}_i\|} \tag{5}$$

The proposal distribution for importance sampling, q, is derived as the *softmax* of the cosine similarities, regulated by a temperature parameter τ that modulates the entropy of the distribution. A lower value of τ leads to a distribution that is more concentrated around the images with the highest cosine similarities, thereby reducing the diversity of samples by favoring those that are most similar to the query. Conversely, a higher value of τ increases the entropy, resulting in a more uniform distribution approximates the characteristics of random sampling.

[2] It is highly possible that either or both of these datasets could have been part of the training data for CLIP. Despite this, from experiment results and existing literature, it is evident that these models perform mediocrely at best for our target task out of the box.

[3] Outside of obtaining the CLIP model, the datasets and utilizing PI sanctioned compute from the Massachusetts Green High Performance Computing Center, there have been no additional physical or virtual resources or materials required for my project. Moreover, no specialized training was required to conduct my research, since it did not involve animal or human testing, interviewing or any laboratory work.

Mathematically, we the define *softmax* function as:

$$softmax(i) = \frac{e^{\frac{s_i}{\tau}}}{\sum_j e^{\frac{s_j}{\tau}}} \tag{6}$$

The probabilities vector generated by Equation 6 serves as our initial detector (VLM) counts g from Equation 1. While in the original implementation of DISCount, these counts are normalized by dividing them by their sum to create the proposal distribution q for Importance Sampling, in our case, $g = q$ due to the nature of it being a probability distribution for a binary classification in the first place.

We initially compare our generated proposal distribution q to a Uniform one, wherein we have equal probabilities for each image, which is equivalent to a Monte Carlo sampling approach. This serves as a baseline to assess not only the effectiveness of our VLM to match a query with appropriate images using cosine similarity, but also the process of the proposal generation with different τ values. Thereby, for each query, an instance of the DISCount estimator with CLIP generated probabilities is employed, and compared to Monte Carlo Sampling (labeled as Uniform).

For various sample sizes n, we sample images according to these distributions, compute the estimated count $\hat{F}$ as seen in Equation 2 and subsequently the confidence intervals. Per Perez et al., 2024, to calculate these intervals for a dataset S (instead of sub regions), we first estimate the importance weight variance $\sigma^2(S)$ as:

$$\hat{\sigma}^2(S) = \frac{1}{n(S)} \sum_{i:s_i \in S} \left(\frac{f(s_i)}{g(s_i)} - \frac{\hat{F}(S)}{G(S)} \right)^2 \tag{7}$$

We finally substitute $\hat{\sigma}^2(S)$ obtained in Equation 7 in Equation 3 to obtain the upper and lower confidence interval bounds.

3.2 Adaptive DISCount using Logistic Regression

The adoption of an adaptive approach within the DISCount framework is motivated by the desire to refine our proposal distribution iteratively, enhancing the accuracy and efficiency of our estimates. As outlined in Owen, 2013, Adaptive Importance Sampling (AIS) can offer an advantage over static sampling strategies by continually adjusting the sampling distribution based on the data acquired thus far, leading to reduced variance and improved estimate convergence.

The adaptive estimator generates and utilizes a dynamically updated proposal distribution. Our approach leverages logistic regression to perform this update and refine the distribution based on a subset of the sampled (screened) images iteratively. Logistic regression is particularly well-suited for this task since it requires minimal resources, training and prediction time. We leverage its ability to provide probabilistic outputs, which directly inform the likelihood of each image containing the queried object based on its features. Each iteration of logistic regression training updates the proposal distribution to be more selective, focusing subsequent sampling efforts on images that are more likely to be relevant. The goal for the adaptive method using regression is to reduce the spread and variance of the estimator by decreasing the likelihood of sampling irrelevant images in future iterations.

Configuration of Hyperparameters The adaptive process is configured with several hyperparameters or choices as described in Owen, 2013 that control the adaptive behavior. These include determining the stopping criterion (k) for the adaptive update, deciding on sample sizes for each updated proposal (n_i), and establishing the appropriate weights (w) for calculating the final estimate ($\hat{F}$). Our logistic regression training also requires us to decide on a regularization term (C). Finally, we also add a smoothing factor (ϵ) to generate our updated proposals. These choices are further described as follows:

- Stopping criterion - choosing k: The question of when to stopping the adaptive update by training additional regression models to generate a new proposal does not have a straightforward answer. Section 4.2 explores several fixed values of k in order to determine trends that help make a stopping decision.

- Sample sizes n: For each updated version of the proposal, we need to determine the number of images to sample in order to calculate $\hat{F}$ or serve as training data for the next update. We might want to utilize most of our samples on the last stage with presumably the best $q^{(k)}$, or we might want to allocate a lot of samples to $q^{(1)}$, predicting high sensitivity to a bad start. Different values of n_i where $i = 0 \dots k$ were tested, the results of which are reported throughout Section 4.

- Weights w: We have to decide how to weigh the estimates and confidence intervals generated from all the iterations. One approach is to only use the very last proposal, however, depending on n_k, this would mean estimating $\hat{F}$ using a considerably small subset of sampled images. Another approach is to weigh from all iterations equally, but this is arbitrary. With the aim of refining the choice of w, the two main approaches utilized in this research include:

  – Sample ratio-based: For k iterations, the combined estimate $\hat{F}$ is calculated as a weighted sum of the estimates $\hat{F}_i$ from each iteration, where weights are based on the sample ratios. This can be expressed as:

$$\hat{F} = \sum_{i=1}^{k} w_i \hat{F}_i \tag{8}$$

  where w_i is set to be the ratio of samples from each proposal.

  – Variance-based: The estimates $\hat{F}_i$ from each proposal are weighted by the inverse of their variances. This approach is advantageous in theory because it allocates more weight to estimates with lower variances, which are typically more reliable. Denoting σ_i^2 as the variance of the estimate from the i-th iteration, the combined estimate $\hat{F}$ is calculated using the variance-weighted average formula:

$$\hat{F} = \frac{\sum_{i=1}^{k} \frac{\hat{F}_i}{\sigma_i^2}}{\sum_{i=1}^{k} \frac{1}{\sigma_i^2}} \tag{9}$$

  Without access to ground-truth in a real world setting, the variance of the estimates

generated from a proposal distribution becomes the primary evaluation criteria for its quality (with Importance sampling and AIS both being variance-reduction techniques in essence).

- C: Different values of regularization terms C_i, where $i = 1 \dots k$, used for each of the logistic regression models were tested. While dataset and query specific, the effect of this term on the estimator performance is explored in Section 4.2.1. Generally, regularization is used to prevent overfitting, ensuring that the model generalizes well on unseen data, which is crucial especially when the training inputs are a sampled from a probability distribution instead of being fixed and known.

- ϵ: A constant smoothing factor ϵ was applied to the probabilities generated per image from the trained logistic regression classifier, before turning them into a normalized vector that served as the updated q. Without this smoothing, any zero or near-zero values for g in Equation 2 (which is the same as q for our case as described earlier) can lead to infinitely large weights in the importance sampling estimation process, which severely distorts the computed estimates. By adding ϵ, we ensure that every sample has atleast an ϵ non-zero probability of being chosen, reducing the variance of the estimator and preventing the dominance of a few samples with disproportionately high weights to a certain extent. For our experiments, ϵ was set to $1\mathrm{e}{-4}$ empirically.

Iterative Update With the hyperparameters as described above, given a total number of samples N and a query, we define the iterative update for our Adaptive procedure as follows. For each iteration $i \leq k$, we:

1. Obtain the number of samples n_i, perform the sampling (with replacement) from the entire dataset using current proposal $q^{(i)}$ and screen them to load the ground truth.

    - Initially, at $i = 0$, we set $q^{(0)}$ to be the proposal generated by applying *softmax* to the CLIP cosine similarities as described in Equation 6.

2. Estimate and store $\hat{F}_i$ and the confidence intervals based on the current proposal and loaded ground truth values for the n_i samples.

3. Extract the CLIP embedding corresponding to the sampled images, serving as our X_{train}.

4. Retrieve the ground truth labels for these images, serving as our y_{train}.

5. Train the logistic regression model on X_{train} and y_{train} using the regularization parameter C_i.

    - For the purpose of finer experimentation, this was implemented by using a linear layer and utilizing the Adam optimizer (Kingma and Ba, 2015) and a Binary Cross Entropy loss function.

6. Use the trained model to predict the positive class probabilities for the remaining unscreened images.

7. Concatenate the ground truth values for the screened images, with the predicted positive class probabilities of the unscreened images to generate an updated probability vector p.

8. Generate the updated $g = q$ by adding ϵ and normalizing as follows:

$$q_j^{(i)} = \frac{p_j + \epsilon}{\sum(p + \epsilon)} \tag{10}$$

Final Estimation We calculate the final estimate of the count of images that match our query $\hat{F}$ by computing a dot product with our weights w, once the stopping criterion (k iterations) is reached. This estimate is based on N, the total number of samples we allowed ourselves to work with. These estimates are compared with directly performing DISCount using the CLIP proposal that serves as the starting point for our adaptive approach and the uniform proposal as described earlier.

3.3 Evaluation

We assess the accuracy of our estimates by measuring the fractional error between the true counts F and the estimated counts $\hat{F}$. This error is computed as follows:

$$\text{Error} = \frac{|F - \hat{F}|}{F} \tag{11}$$

In addition, we calculate the average confidence interval width normalized by F. The width of the 95% confidence intervals for each estimate $\hat{F}_i$ is computed as:

$$\hat{F}_i \pm 1.96 \times \sqrt{\frac{\hat{\sigma}^2}{n_i}} \tag{12}$$

where $\hat{\sigma}^2$ is the estimated variance of $\hat{F}_i$, and n_i is the number of samples used for the estimation. The factor of 1.96 is derived from the standard normal distribution, where ± 1.96 standard deviations from the mean account for approximately 95% of the data.

4 Results

In this section, we present the results using DISCount and the Adaptive approach for a vast range of representative queries on both datasets. For each of the experiments reported in this section, we run 100 trials and plot the average metrics over them in order to make our results robust to noise.

4.1 CLIP-based DISCount

4.1.1 Direct class-level queries

We initially compare using the CLIP generated g and q for DISCount to Monte Carlo (or Uniform) Sampling, wherein we assign equal probabilities to each image for sampling.

Direct classification refers to identifying the counts of images that match a particular class in the dataset. CUB contains 200 classes, with each class having ≈ 60 representative images. For modified Caltech, we have 128 classes with 30 images each as described in Section 3. Radford et al., 2021 describe prompt engineering techniques for queries to improve CLIP's zero-shot performance. Borrowing from their findings, the queries used to append the phrase "A photo of a/an" to the class names from the datasets, eg: "A photo of an Acadian Flycatcher" for CUB and "A photo of a backpack" for modified Caltech.

We first observe that using CLIP directly without any screening results in high error rates - approximately 65% and 125% for modified Caltech and CUB respectively, despite reasonably careful classification threshold selection. Next, Figure 1 shows a side by side comparison between DISCount using the CLIP-ViT-L-14 VLM and Monte Carlo (uniform) Sampling. For both datasets, initially, DISCount achieves lower error rates with fewer screened samples, however in some cases we observe that its performance converges to that of a uniform sampling proposal over time.

The results obtained in Figure 1 were further scrutinized by modifying the temperature parameter τ of the *softmax* function that was used to generate the proposal distribution q for

Figure 1: CLIP-based DISCount compared to a Uniform proposal for direct classification queries

DISCount. Figure 2 shows the quantitative impact of τ on the performance of DISCount. Across both datasets, it is evident that lowering τ provides lower error rates for DISCount, regardless of the sample size. From the tested values, the best performance is achieved when $\tau = 0.1$ (which was the value used in Figure 1), where DISCount significantly outperforms the uniform sampler, averaging nearly a 12% error on CUB at 2000 samples ($\approx 17\%$ of the dataset) and an approximate 9% error on modified Caltech when a similar fraction of the dataset is sampled (≈ 650 images). The lower temperature curves also depict a lower initial error rate with lower samples, asserting that the VLM correctly identifies instances of the query, and that they are more effectively sampled by the peakier distribution generated with a smaller τ. It is to note that lowering τ below 0.1 led to a further decrease in error rates, however, finding the optimal value of τ is highly dataset and query specific, hence not meaningful once a certain acceptable performance threshold or trend is observed.

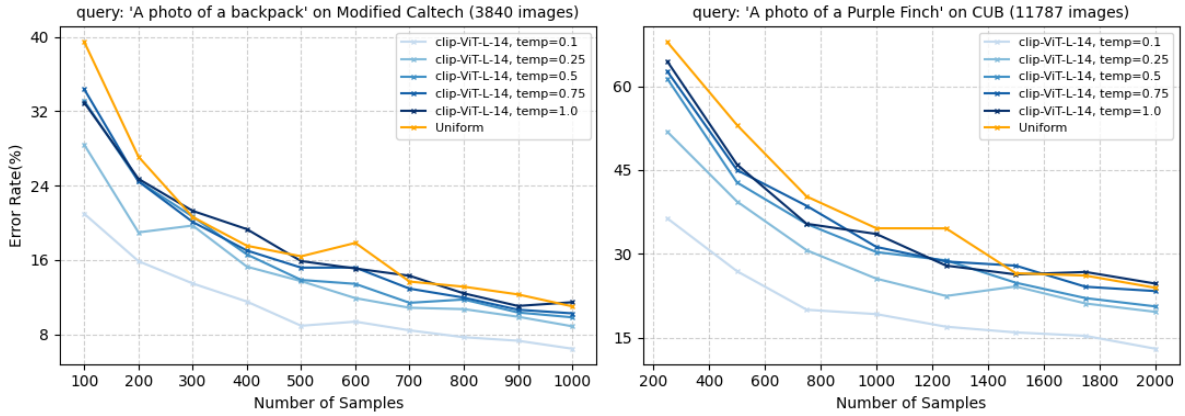

Figure 2: The effect of τ on CLIP-based DISCount for direct classification queries

In order to examine the reliability of the estimated counts, we calculate the confidence intervals (CIs) as depicted in Equation 12. Figure 3 plots their width for the representative queries.

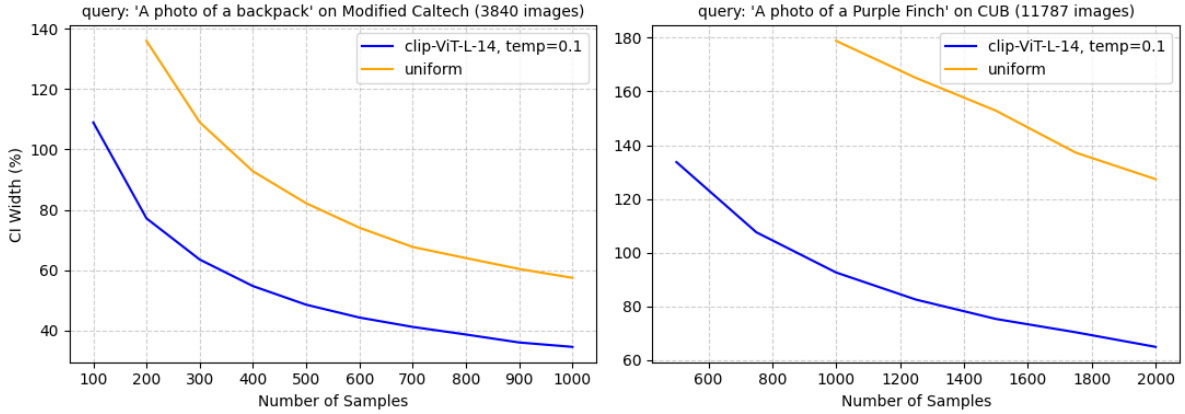

Figure 3: Confidence Interval widths as percentages from the count estimates at $\tau = 0.1$

The width of the CIs is inherently linked to the variance of the importance weights; a smaller variance results in tighter CIs, suggesting that our estimates are consistent across different samples drawn from the proposal distribution. As seen in Figure 3, the counts provided by DISCount have a much lower CI width across all samples, especially for CUB, indicating that they are much more reliable.

For all of the classes (and their corresponding queries) tested, the results follow a similar trend as seen in the representative figures, establishing DISCount with a lower softmax temperature value to improve sampling to outperform the baseline of a Monte Carlo method for direct class-level queries.

4.1.2 Broad category-level queries on modified Caltech dataset

The next set of queries that were examined include classifying images based on their broader level categories rather than their individual dataset assigned classes. This experimentation was performed on modified Caltech which contains a diverse set of images of everyday, well-known items. Examples of such categories include ‘animals’, ‘food items’, ‘sporting equipment’ etc. This task provided an increased level of difficulty from direct classification, helping evaluate CLIP-based DISCount on simpler versions of high-level queries that it would receive in the real

world.

Figure 4 below showcases how the approach performs in this setting. We see the lower τ values continue to outperform, however the margin of outperformance is much lower, especially for identifying 'sports equipment'.

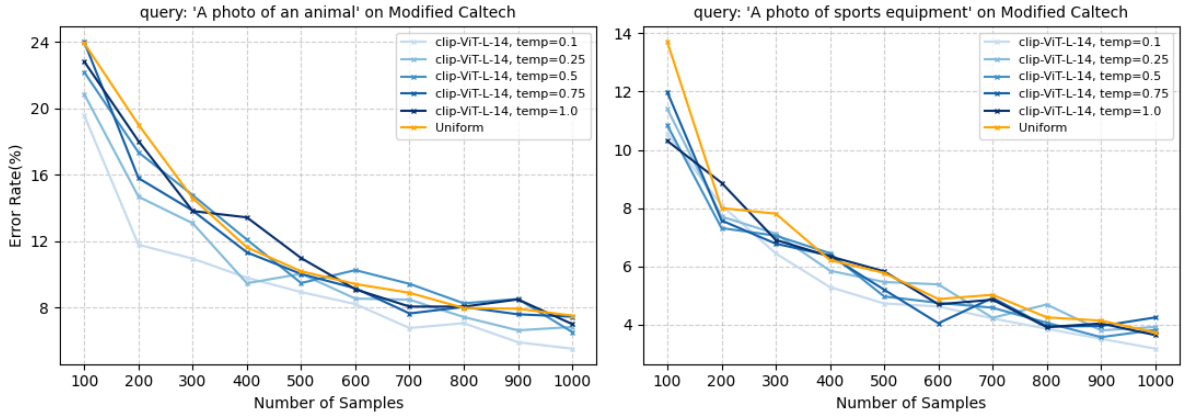

Figure 4: The effect of τ for high-level queries on modified Caltech

It is also observed that most higher temperature DISCount estimators do not offer a substantial improvement in count estimation compared to the Uniform estimator, with $\tau = 0.1$ and lower being the best performing parameter once again. Results from other high, category-level queries follow the same trend of only the lower temperatures justifying the CLIP-based proposal, as we start to see initial signs of the limitations of this approach.

4.1.3 Attribute-level queries on CUB dataset

The next type of queries we evaluate are those requesting for classification based on finer-grained image features or attributes in CUB. CUB comes with annotations of a large corpus attributes pertaining to the shape, pattern, color and length of either the entire bird or specific parts of its body. Each of these type of attributes were tested individually to identify the scope at which, or if ever, the simple CLIP-based DISCount proposal does not offer improvement over its Uniform counterpart, regardless of tuning τ. Figures 5, 6, 7, 8 and 9 below show the results for these local (pertaining to specific parts of the bird) and global (pertaining to the entire bird) fine-grained queries.

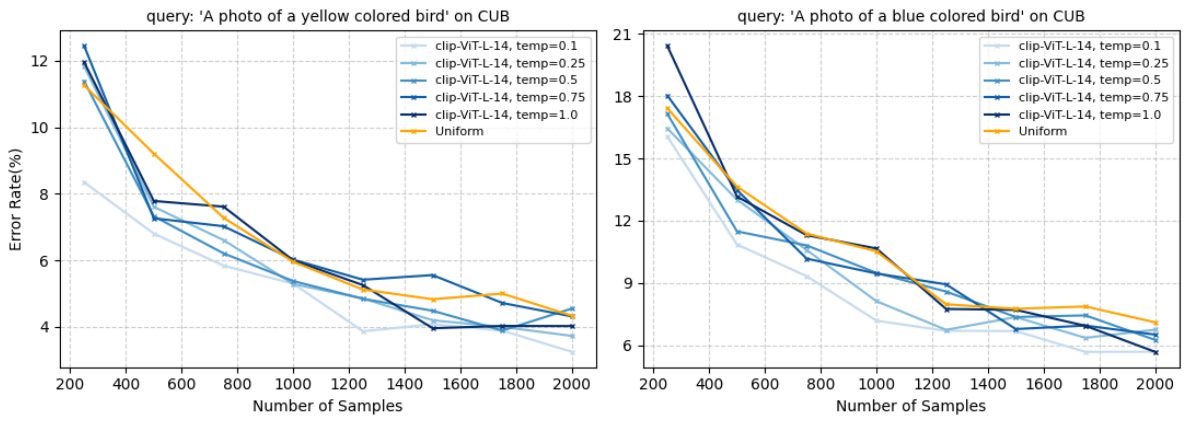

Figure 5: The effect of τ for whole bird color based queries on CUB

Figure 5 follows the pattern we have observed so far, with DISCount configured with lower τ values, leading to increasingly improved performance. The best configuration at $\tau = 0.1$ achieves an $\approx 8\%$ error rate with only 250 samples which amounts to roughly 2% of the dataset, compared to vanilla DISCount with $\tau = 0.1$. [4] The trend of all of the estimators converging at higher sample values also remains intact, indicating diminishing returns over a considerably large sample size.

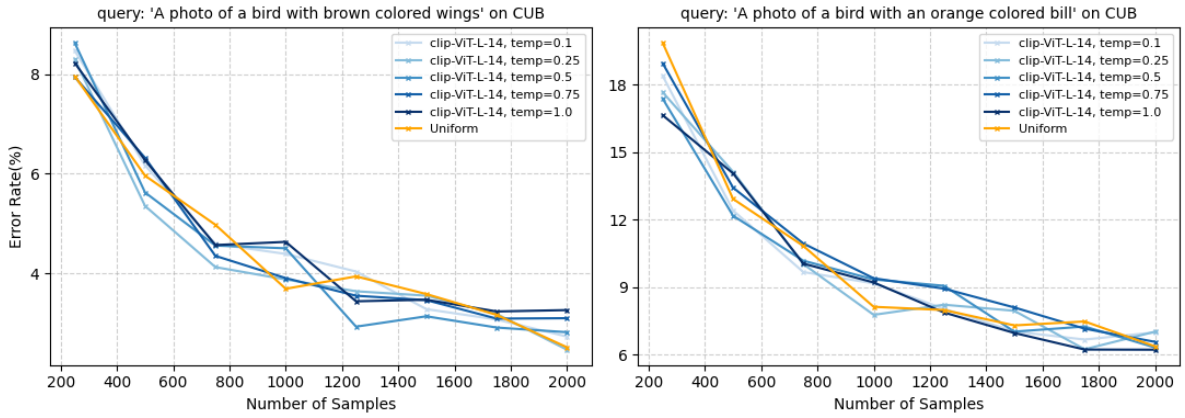

Figure 6: The effect of τ for localized color based queries on CUB

As seen in the remaining figures however, DISCount's performance on this task matches the Monte Carlo method, meaning that the proposal generated by our VLM insufficiently distinguishes between images that match the query to the ones that do not. While the overall error rate

[4] For all of the aforementioned and subsequent results, it is important to note that the error rate value for a particular sample size N is significantly affected by the actual number of images that match the corresponding query in the given dataset. Generally speaking, higher ground-truth values lead to a lower error rate, since positives are easier to sample and the samples represent a more descriptive distribution of the entire dataset. This is another reason why majority of the analysis is done comparing each method and hyperparameter configuration to the baseline of a Monte Carlo sampling estimator, rather than solely looking at the error-rate values out of context.

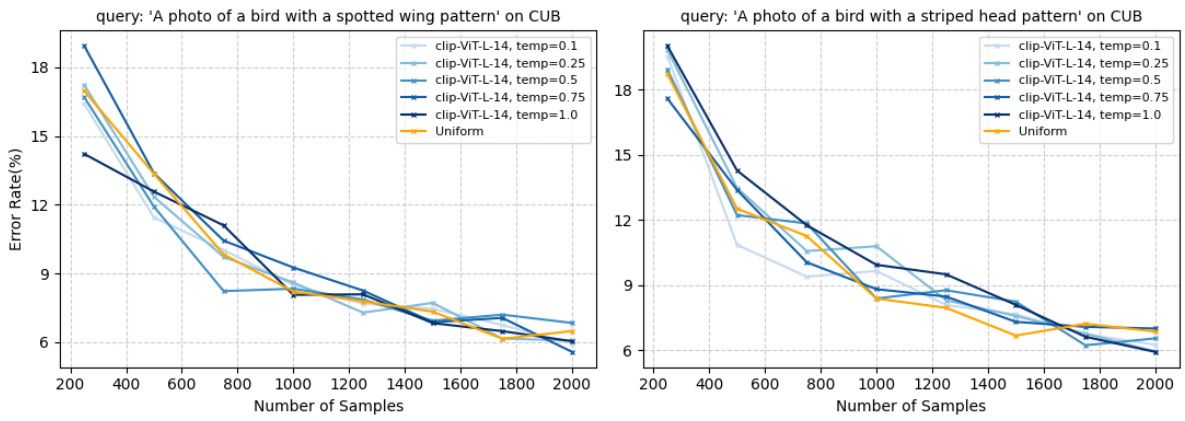

Figure 7: The effect of τ for localized pattern/texture based queries on CUB

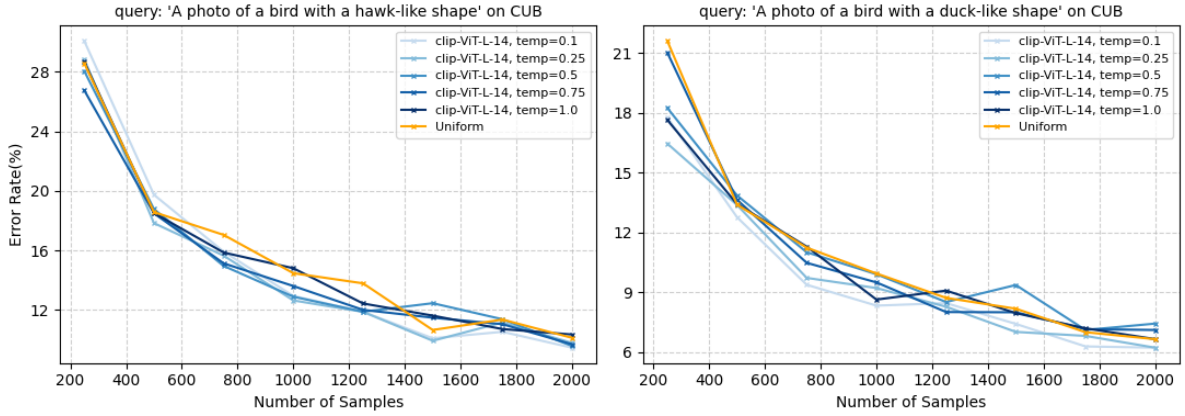

Figure 8: The effect of τ for whole bird shape based queries on CUB

is lower compared to that of the direct classification task, this can be attributed to the number of images that match the query - the ground truth being a larger fraction of the entire dataset. The tuning of τ does not help improve performance either, with all the values tested providing similar results within a margin of error.

This performance of DISCount likely stems from the limitations of VLMs like CLIP in capturing subtle attribute-based distinctions. These models are trained on broad data, emphasizing general category recognition rather than the nuanced detection of fine-grained attributes. This potentially explains why, even with adjustments to the temperature parameter τ, the VLM's proposal distribution does not surpass Monte Carlo in distinguishing closely related categories.

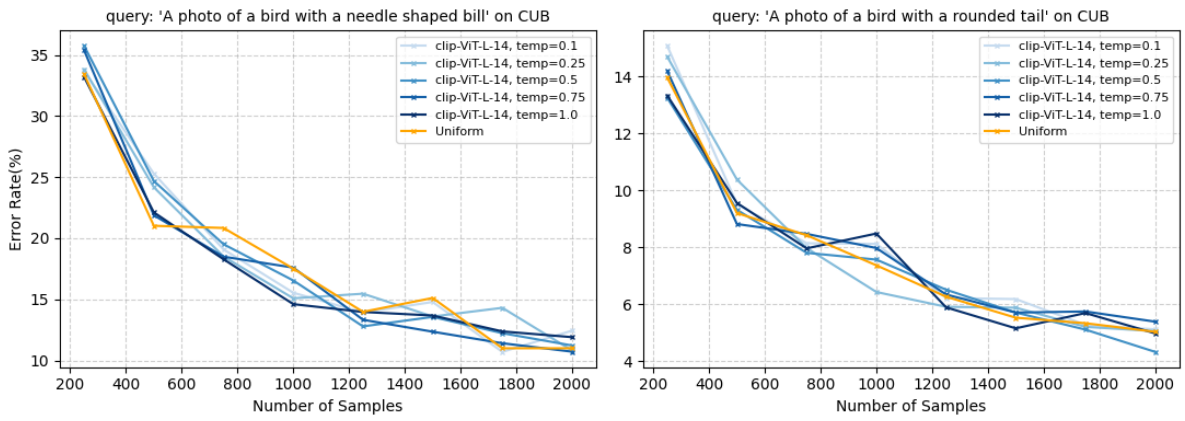

Figure 9: The effect of τ for localized shape based queries on CUB

4.2 Adaptive DISCount

For the Adaptive process, we train a logistic regression (LR) model on the sampled images and update our proposal distribution based on the model's prediction on the remaining data as described in Section 3.2.

The standard regularization parameter for training the regression models was set to $C = 1\mathrm{e}{-2}$, however due its sensitive nature, its impact was further studied in Section 4.2.1. For initial experiments the stopping criteria was manually set to only allowing 1 update, thus having an original CLIP-based proposal, and a subsequent adaptive proposal using the regression model. The effect of increasing the number of update iterations was studied in Section 4.2 below.

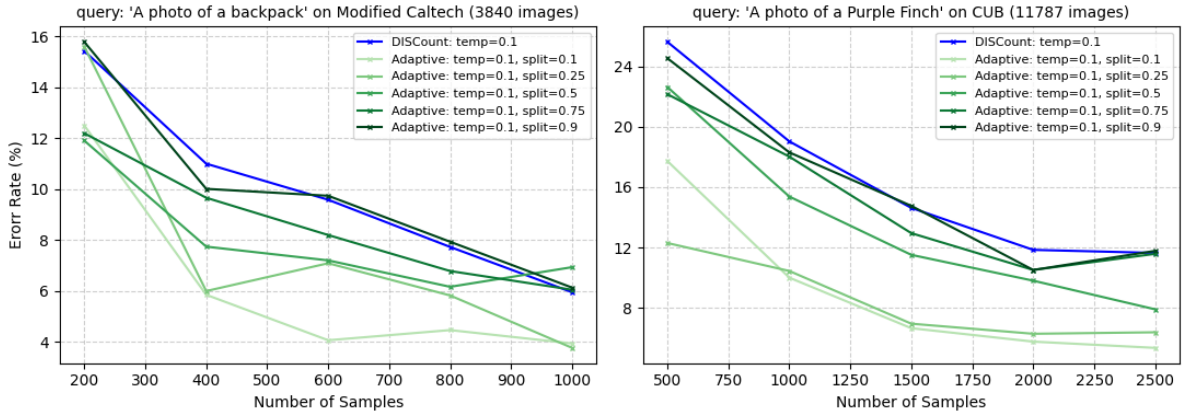

Figure 10: Sample ratio-based Adaptive DISCount for direct classification using different n values

Figure 10 shows the performance of the initial Adaptive Importance Sampling method tested for various *split* values for the direct classification queries. Here, *split* indicates the fraction of

samples from the original proposal that were used to train the linear classifier. Therefore, with only 1 adaptive iteration we have have $n_1 = \lfloor N \times split \rfloor$ and $n_2 = N - n_1$ with N being our total number of samples. For the sample ratio-based method, as described in Section 3.2, our final estimate $\hat{F}$ is calculated as $split \times \hat{F}_0 + (1 - split) \times \hat{F}_1$.

From Figure 10, we observe that our Adaptive method outperforms vanilla CLIP-based DISCount with an optimal τ value for almost every value of the $split$ threshold tested. It is also seen that lower split values offer a better performance, indicating that the logistic regression needs only a few samples for this direct classification task to be effectively trained, after which, most of the effort should be reserved for sampling from the newer proposal.

This sample ratio-based method is directly compared with the variance-based weighting method that was described in Section 3.2. Figure 11 below shows the results when compared to vanilla CLIP-based DISCount with $\tau = 0.1$ for the direct classification task.

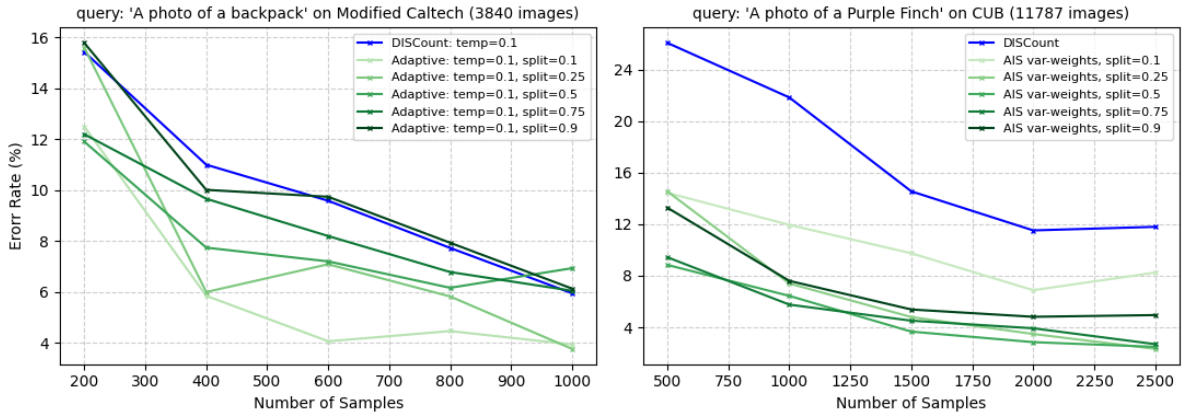

Figure 11: Variance based Adaptive DISCount for direct classification using different n values

Comparing Figures 10 and 11, we see that the weights inversely proportional to the proposal distribution's variance lead to significantly improved results, especially for the query tested on CUB. While the results for the sample ratio weights remain similar to the variance ones on Caltech, for CUB, the best *split* ratio for the variance weights achieves an improved 9% error rate with $N = 500$, compared to 12% in Figure 10 and nearly 26% for vanilla CLIP-based DISCount. This showcases a huge improvement in estimator performance, which our experiments have found to replicate over all of the direct classification queries tested.

The variance weights also significantly improve estimator performance for broader category-level classification on modified Caltech which was discussed in Section 4.1.2. We notice similar trends as observed so far continue to be true for this task, as depicted in Figure 12 below.

Figure 12: Variance based Adaptive DISCount for high-level queries on modified Caltech

For example, from both Figures 11 and 12, we notice that the best performing $split$ value for most queries is now 0.5, with 0.1 and 0.9 being the worst performers, indicating that for variance-based weights, the ideal sample-ratio tends towards an even split of N to calculate n_i per iteration.

Determining stopping criteria As discussed previously in Section 3.2, determining a stopping criteria for the Adaptive method is extremely nuanced and does not have a direct formulation. Instead of attempting to identify a one-size fits all solution for this problem, we experimented with increasing values of k from 2 to 10 in order to assess the effect of increasing the iteration counts on the overall performance. Figure 13 below showcases the results for different values of k, or the number of updates to our proposal. This is done over a variety of sampling vectors n, with *split* values set to $0.1, 0.5$ and 0.9 in order also examine how choosing n affects this process. For example, a *split* value of 0.1 in this case indicates 10% of the samples N were used for the initial estimate from the vanilla DISCount proposal, after which, for each update, the remaining samples were divided equally. The same process applies for *split* $= 0.5$ and 0.9.

The results from Figure 13 unanimously depict that a higher number of iterations k give us better estimates for our counts. This implies that increasing the number of updates to the

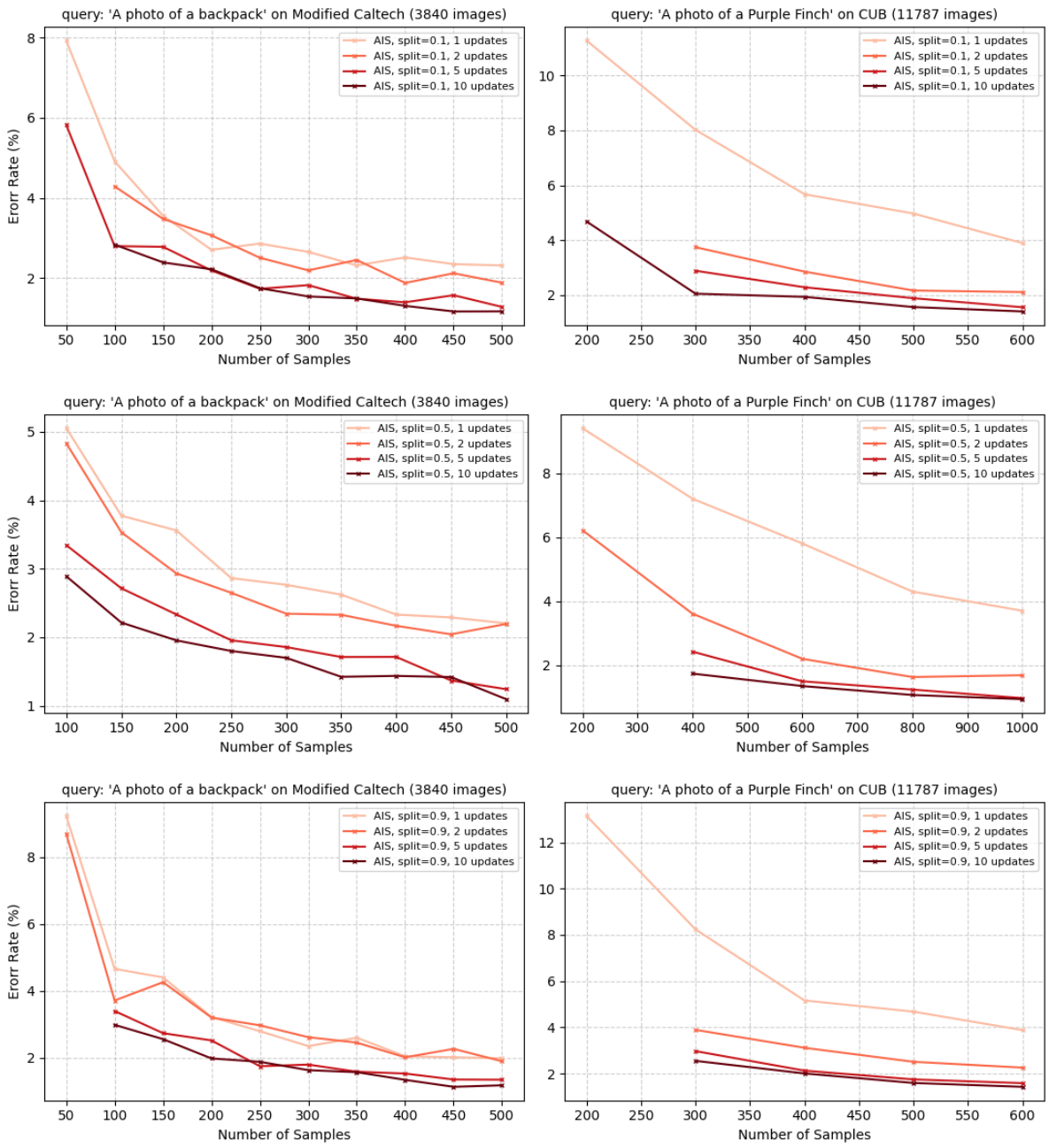

Figure 13: Effect of increasing updates/iterations (k) for Adaptive DISCount

proposal distribution enhances the accuracy and stability of the estimator. A possible reason for this could be the fact that at every iteration, the sampled data, albeit a smaller fraction, contain newer images that are easier to tailor the model to, rather than attempting to generalize over a larger quantity. Furthermore, it is important to note that the increased iterations are highly susceptible to a 'false start' as each proposal influences the next iteration's sampled images. We also empirically observe that higher iterations work better when the total sample size N is sufficiently large, since each individual subset of samples n_i must contain enough positive and

negative examples to properly train the logistic regression at each iteration.

4.2.1 Attribute-level queries

Despite promising results shown in Figure 5 and partly in Figure 8 for whole bird based attribute level queries, applying the Adaptive scheme with the default parameters actually worsens estimator performance when compared to vanilla CLIP-based DISCount. The results for this can be seen in Figure 14 below.

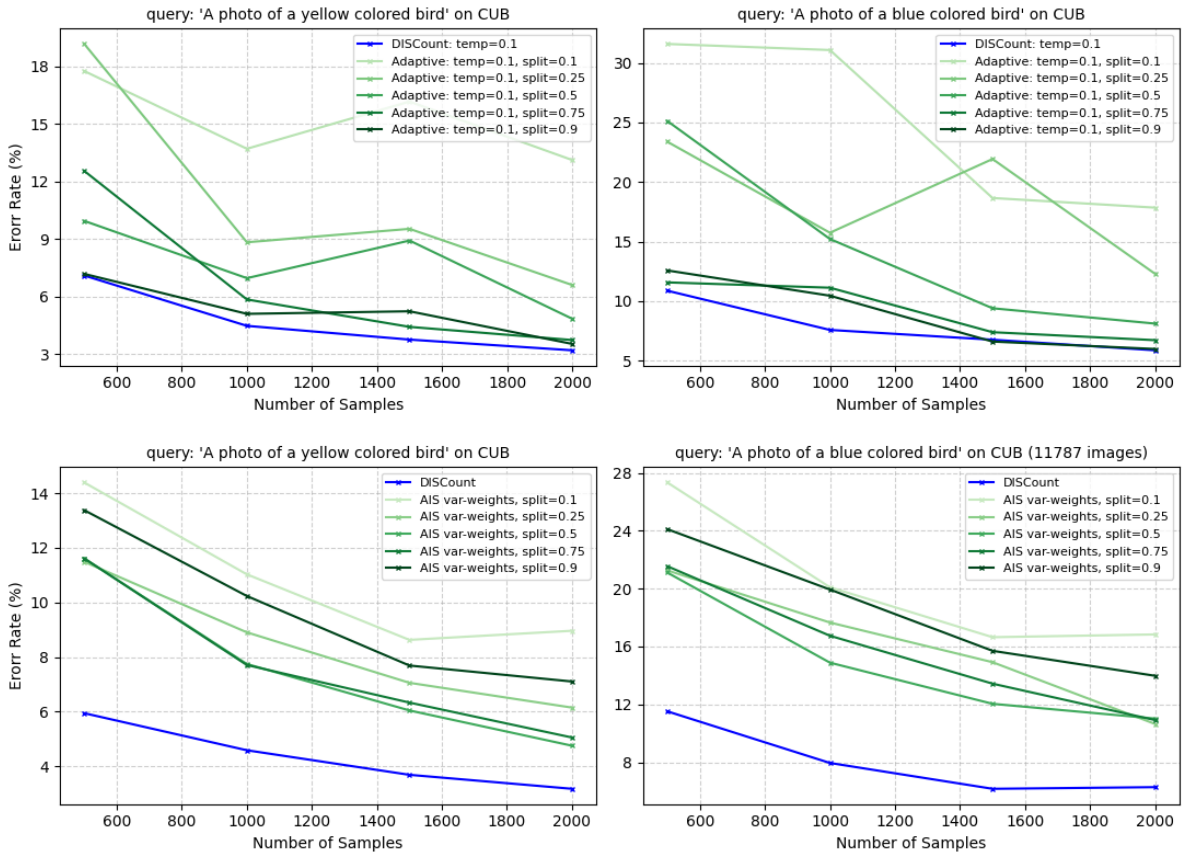

Figure 14: Adaptive DISCount for attribute-level queries (whole bird color) on CUB

The top section of Figure 14 utilizes the sample-ratio based weighting, whereas the bottom section utilizes variance weights. Despite variance-based w outperforming its sample-ratio counterpart, which is consistent with our findings, we see that all of the Adaptive configurations significantly underperform vanilla DISCount. This indicates that while the original proposal is suitable for importance sampling and estimation, the subsequent one generated by the logistic regression does not encapsulate the overall distribution of data for these queries of higher difficulty. In order to analyze the cause behind this, we examine the performance of the lo-

gistic regression in isolation for this task, alongside tuning its primary hyperparameter - the regularization terms C.

Determining C for logistic regression The default C_i value for each iteration utilized in the experiments demonstrated so far was set to $1\mathrm{e}-2$. However, as seen in Figure 14, the logistic regression models trained with this configuration for attribute-level queries do not seem to improve CLIP-based probabilities. This motivates further scrutiny and tuning of the regularization parameter.

Section A.1 of the Appendix shows the Precision-Recall (PR) Curves of the probability distribution generated by the logistic regression model trained across different regularization values, with a fixed sample size of $N = 1000$. It is seen in Figure 16 that for direct class queries, lower regularization values lead to significantly better performance. However, Figure 18 shows that for attribute-level queries, we have the opposite scenario wherein higher regularization values are preferred.

It is important to note that when attribute-level queries were tested for rarer attributes with only $100 - 200$ positives on CUB (similar to an individual class size), we observed the same trend with higher regularization values improving performance (up until $C = 0.1 - 0.5$).

Generating these PR curves requires access to ground-truth. For this, we can have a subset of our sampled images serve as a validation set for regularization tuning on-the-fly in our adaptive loop. However, this approach (a) reduces the training and estimation sample sizes which has a significant effect for low values of N, and (b) requires sufficient images originally sampled from both classes to allocate a fraction for validation. As an alternative, we find that the variance of the estimates generated serves as a reasonable proxy for performance. Figures 17 and 19 in the Appendix show how the variances change based on regularization values for direct class and attribute level queries respectively. We observe that lower variances correspond to better PR curves.

With these findings, we modify the regularization term C to be equal to 0.1 for our local

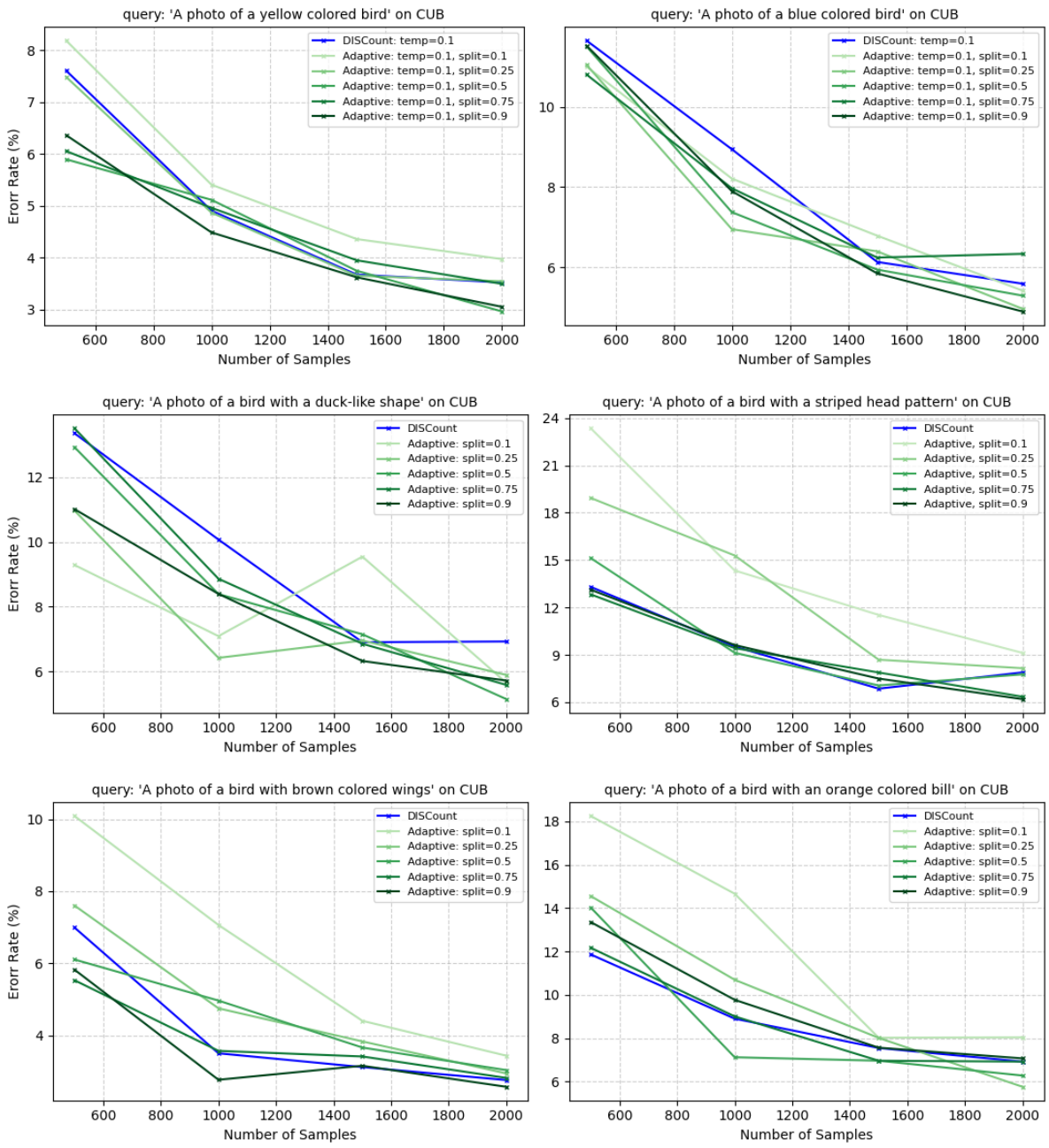

Figure 15: Adaptive DISCount for several attribute-level queries on CUB, $C = 0.1$

and global attribute-level queries on CUB and achieve similar or better performance than DISCount as seen in Figure 15. These results were calculated using sample-ratio based weights, instead of variance-based ones which performed worse in this scenario. In most cases we see that higher $split$ values $(0.75, 0.9)$ consistently outperform DISCount, with the range of improvement varying based on the granularity of the query. While certain cases do not improve estimate error rates significantly, we are assured a reduced variance with the Adaptive method (as seen in Figures 17, 19) which makes it more desirable in real-world settings.

5 Discussion and Conclusions

DISCount outperforms zero-shot baseline significantly We observe that despite careful threshold selection for the binary classification of images, DISCount outperforms the zero-shot CLIP baseline by a significant margin (a minimum of $50-80\%$) with as few as $20-50$ samples depending on dataset size.

DISCount outperforms Monte-Carlo sampling for direct classification and broad high--level queries We observe that DISCount outperforms a Monte-Carlo (Uniform) sampler for such queries by 20-60% based on sample size, with lower samples leading to a higher margin of outperformance. It only requires DISCount $\approx 0.1-0.5\%$ of the dataset to achieve this result. DISCount, however, struggles to surpass Monte-Carlo for attribute-level queries on CUB.

Adaptive DISCount outperforms zero-shot, Monte-Carlo and DISCount We see that the Adaptive DISCount proposed in our paper outperforms all previously tested methods across all three types of queries with careful hyperparameter tuning. In particular, we see an approximately $30-40\%$ improvement in estimate error rates for classification and broad category-level queries on modified Caltech and $50-65\%$ on CUB, alongside $10-20\%$ improvement for attribute-level queries in most cases on CUB.

We discuss several experiments and derive strategies to perform this hyperparameter tuning, as listed below:

- It is beneficial to use lower samples initially (n_0) to train the first logistic regression if there is a higher confidence in CLIP performance for the query.
- Variance-based weighting is superior for direct classification and high-level queries
- Increasing iterations (k) to greater than 1 increases performance in cases, achieving error rates as low as $2-5\%$ for class-level queries. However, it is sensitive to a false start and requires sufficient samples with adequately balanced classes, hence is not always justified, especially for the other, harder query types.

- Lower regularization values for the logistic regression model ($1\mathrm{e}-3, 1\mathrm{e}-2$) are better suited for direct classification or broad-level queries, whereas attribute-level queries require better generalization, hence higher C values ($0.1-0.5$).

- To identify suitable a regularization term to use, we can use variances of estimates generated as a proxy - lower variance values are preferred.

- Although we cannot directly correlate model performance with variance, we perform sample-ratio based weighting across pre and post update iterations to obtain a similar or better estimate than DISCount (in terms of error rate) while simultaneously achieving variance reduction.

Through this research, we have created a flexible, computationally efficient mechanism for counting images in large unlabeled datasets that requires minimal human supervision. Our approach is limited by the accuracy of human spot-checking to generate ground-truth from samples and the extent of hyperparameter tuning for the Adaptive method. While we achieve variance reduction, we still note that variance values are relatively high. Furthermore, ground-truth generation for queries that require domain-knowledge (such as identifying bird species) might be more human-error prone than ones describing visual features in the image.

6 Future Work

There are several avenues of exploration that remain pertinent for future research. The following suggestions cater towards improving overall estimation error for descriptive queries within the Adaptive DISCount framework, better predicting the quality of generated proposals and quantifying human effort to calculate the time and resources saved by human-in-the-loop techniques such as ours compared to fine-tuning or other labeling-intensive approaches.

State-of-the-art VLMs Future research should consider exploring a wider array of Vision-Language Models (VLMs) beyond CLIP, particularly those that might exhibit superior performance in specific domains or scenarios. Since the best performing model has the potential to change based on the target dataset, it is also practically necessary to investigate methods to test model performance without complete access to labels. One could perform Active testing (Kossen et al., 2021) to determine out-of-the-box model accuracies using only a small number of labeled examples.

Datasets spanning a variety of domains Expanding the variety of datasets can demonstrate the generalizability of our method. These datasets could pertain to medical imaging, satellite imagery, or specialized industrial contexts, where the visual elements significantly differ from those in everyday objects and scenes.

Transductive learning With a fixed prediction dataset for each iteration in our adaptive loop i.e. the unscreened images, one could consider replacing the logistic regression model with transductive learning methods (Gammerman et al., 2013) in order to make the learning process more data-centric.

Fine-tuning While our current framework leverages pre-trained models, there is substantial potential in exploring fine-tuning strategies. Future work could investigate the impact of fine-tuning VLMs on a subset of task-specific data before employing them in the Adaptive DISCount framework. In addition, it would also be beneficial to know the performance of fine-tuned mod-

els and compare them with our method with regards to both error rates of estimates and expenditure of human effort.

Quantifying human effort To quantify and compare labeling effort to that of screening, Perez et al., 2024 utilize statistics generated by Su et al., 2012 for median times to draw bounding boxes versus verification. Generating similar quantitative estimates for the reduction in labeling effort resulting from our method would be beneficial to calculate across different categories of queries and datasets. In real-world settings, we expect our method to only get faster once deployed, since repeated querying would lead to an ever-increasing corpus of labeled images that can be reused.

Count multiple instances per image Future versions of the framework could also explore the ability to count multiple instances of objects within a single image, a common requirement in many practical applications. Leveraging Open Vocabulary object detection models could be suitable for this task.

References

Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C. L., Batra, D., & Parikh, D. (2016). Vqa: Visual question answering.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Gammerman, A., Vovk, V., & Vapnik, V. (2013). Learning by transduction.

Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., & Qiao, Y. (2021). Clip-adapter: Better vision-language models with feature adapters.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2017). Making the v in vqa matter: Elevating the role of image understanding in visual question answering.

Griffin, G., Holub, A., & Perona, P. (2022, April). Caltech 256. https://doi.org/10.22002/D1.20087

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Kim, W., Son, B., & Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings*. http://arxiv.org/abs/1412.6980

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything.

Kossen, J., Farquhar, S., Gal, Y., & Rainforth, T. (2021). Active testing: Sample-efficient model evaluation.

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning.

Meng, C., Liu, E., Neiswanger, W., Song, J., Burke, M., Lobell, D., & Ermon, S. (2021). Iscount: Large-scale object counting from satellite images with covariate-based importance sampling.

OpenAI. (2023). Gpt-4 technical report.

Owen, A. B. (2013). *Monte carlo theory, methods and examples*. https://artowen.su.domains/mc/.

Perez, G., Maji, S., & Sheldon, D. (2024). Discount: Counting in large image collections with detector-based importance sampling. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(20), 22294–22302. https://doi.org/10.1609/aaai.v38i20.30235

Pham, H., Dai, Z., Ghiasi, G., Kawaguchi, K., Liu, H., Yu, A. W., Yu, J., Chen, Y.-T., Luong, M.-T., Wu, Y., Tan, M., & Le, Q. V. (2021). Combined scaling for zero-shot transfer learning.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 july 2021, virtual event* (pp. 8748–8763, Vol. 139). PMLR.

Ren, M., Kiros, R., & Zemel, R. S. (2015). Exploring models and data for image question answering. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, 2953–2961.

Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., & Keutzer, K. (2021). How much can clip benefit vision-and-language tasks?

Su, H., Deng, J., & Fei-Fei, L. (2012). Crowdsourcing annotations for visual object detection. *Workshops at the twenty-sixth AAAI conference on artificial intelligence*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011, July). *The caltech-ucsd birds-200-2011 dataset*.

Wah, C., Van Horn, G., Branson, S., Maji, S., Perona, P., & Belongie, S. (2014). Similarity comparisons for interactive fine-grained categorization. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 859–866. https://doi.org/10.1109/CVPR.2014.115

Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., Xu, J., Xu, B., Li, J., Dong, Y., Ding, M., & Tang, J. (2023). Cogvlm: Visual expert for pretrained language models.

Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., … Zhang, P. (2021). Florence: A new foundation model for computer vision.

Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., & Parikh, D. (2016). Yin and yang: Balancing and answering binary visual questions.

A Appendix

A.1 Logistic Regression Tuning

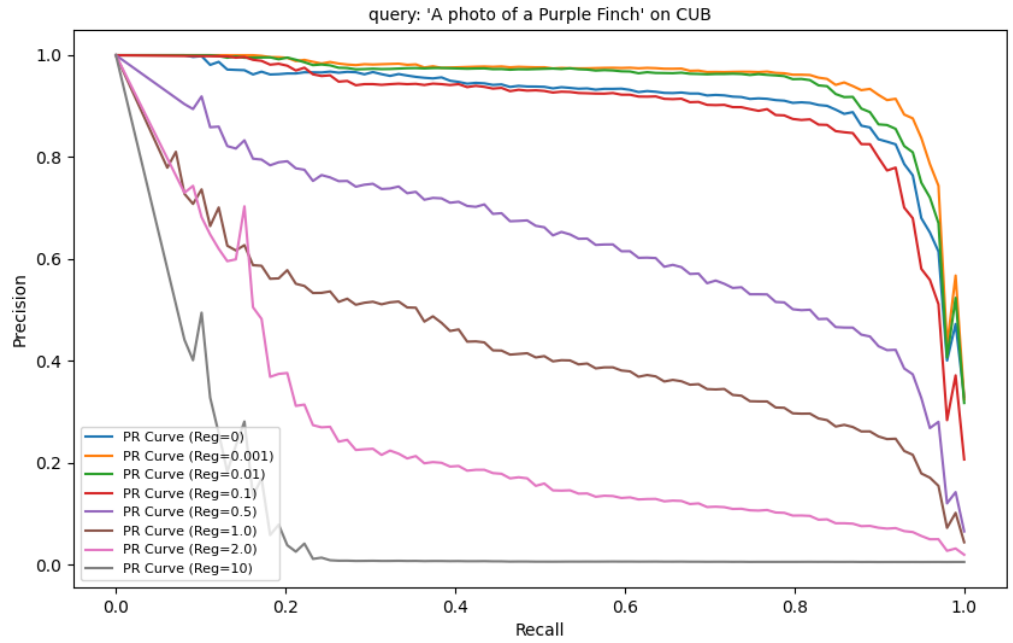

Figure 16: $C = 0.1$

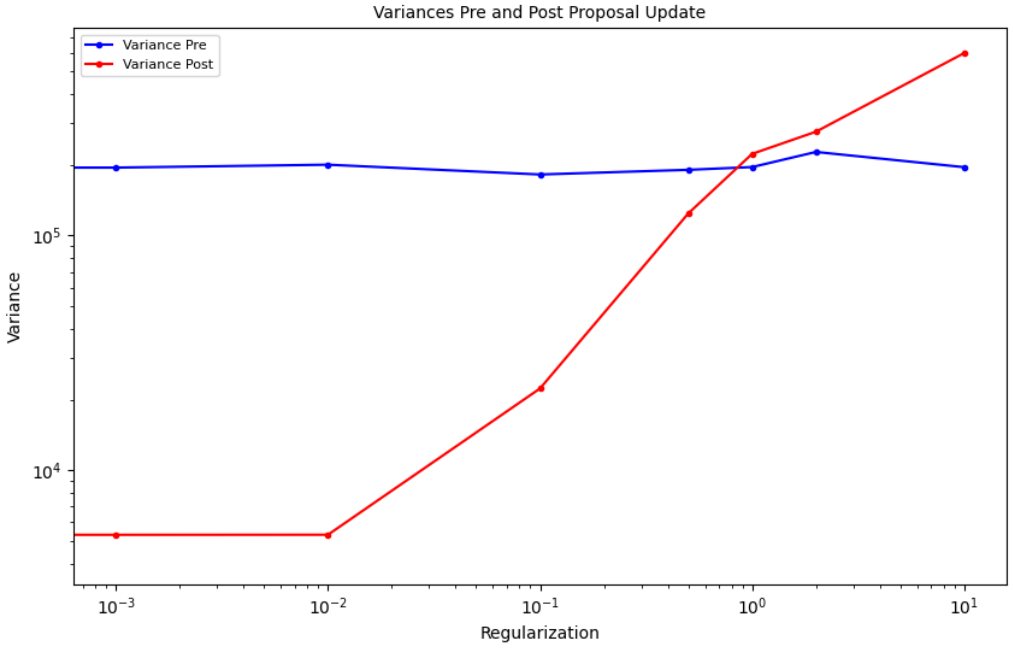

Figure 17: $C = 0.1$

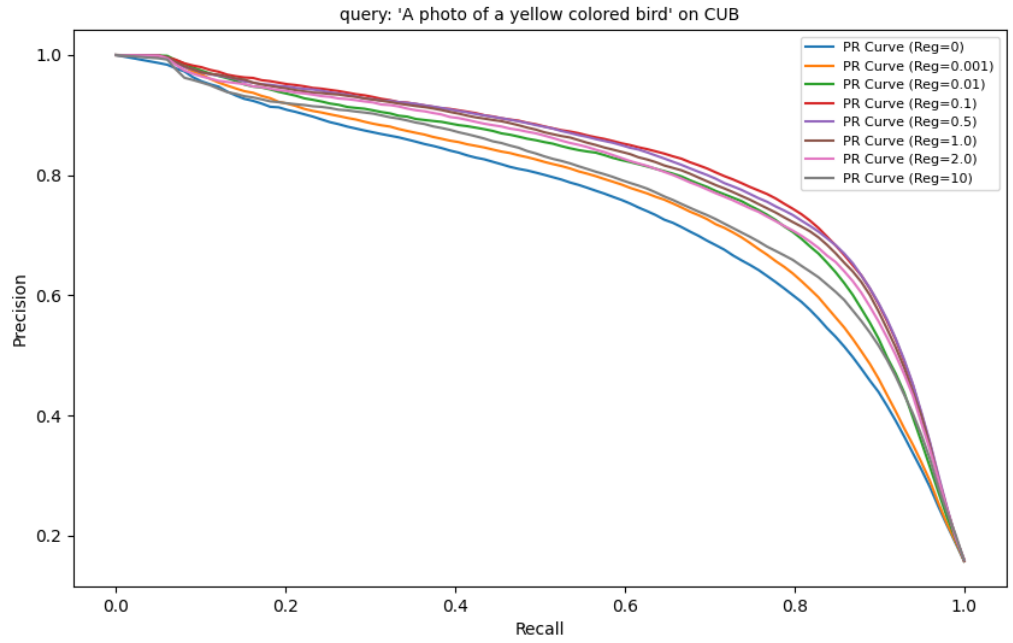

Figure 18: $C = 0.1$

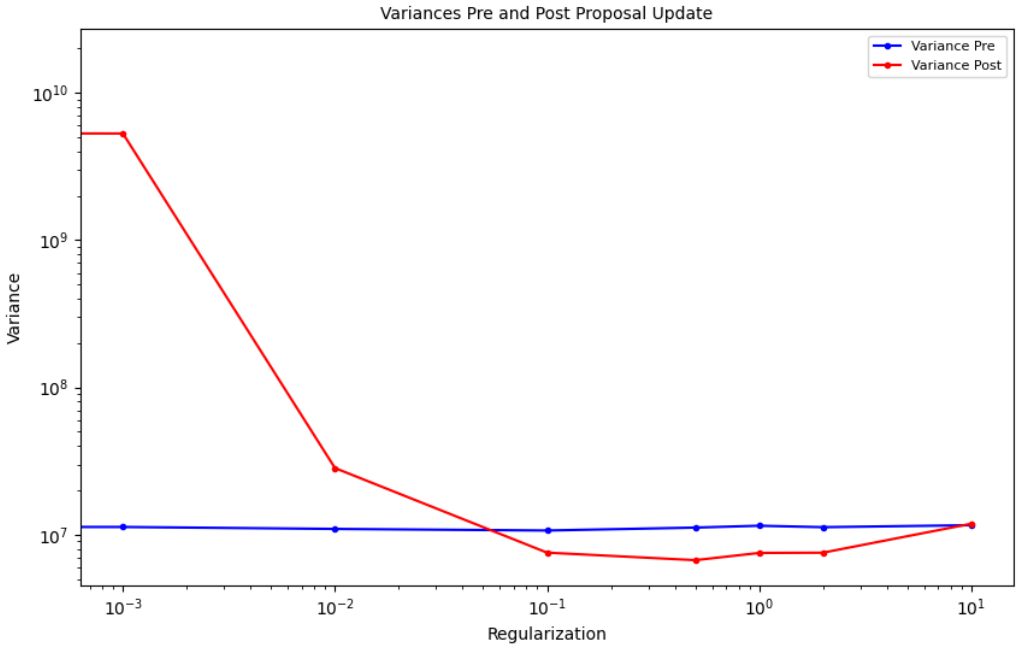

Figure 19: $C = 0.1$