Seeing Only What Exists: Visibility-Aware Contrastive Learning for Concept-Level Hallucination in Vision–Language Models

Hikaru Shijo[1] Yutaka Yoshihama[1] Yasunori Ishii[2] Takayoshi Yamashita[3]
[1]Panasonic Automotive Systems Co., Ltd. [2]Panasonic Holdings Corporation [3]Chubu University
{shijo.hikaru, yoshihama.yutaka, ishii.yasunori}@jp.panasonic.com takayoshi@fsc.chubu.ac.jp

Abstract

Vision–Language Models (VLMs) such as CLIP achieve strong performance in zero-shot recognition, but often suffer from hallucination, assigning high confidence to visual attributes absent from the image. This issue arises because conventional contrastive learning optimizes only relative similarity between positive and negative pairs and does not explicitly penalize alignment with absent concepts. In this work, we propose a fine-tuning framework to mitigate hallucination in VLMs. Based on concept visibility scores predicted by a large vision–language model (LVLM), we introduce a contrastive learning objective with soft concept visibility that aligns image embeddings with visually supported concepts while suppressing responses to likely invisible ones. Experiments on CUB and Oxford Pets show that the proposed method effectively reduces hallucinated attributes while improving classification performance.

1. Introduction

Vision–Language Models (VLMs) such as CLIP [8] and ALIGN [4] achieve strong zero-shot and downstream performance by contrastively aligning images and text in a shared space, and now serve as general-purpose foundations for vision–language understanding.

However, despite their strong performance, their predictions are often not sufficiently grounded: CLIP can assign high confidence to concepts weakly supported—or entirely absent—in the image [5, 12, 14]. A key cause is over-reliance on global features and language co-occurrence, conflating visually observed evidence with language priors [5, 12, 14]. To improve downstream performance, CLIP is often adapted through fine-tuning, with methods such as FLYP [3] showing the benefit of continuing contrastive learning. More recently, several approaches incorporate external concept knowledge—e.g., LLM/LVLM-generated prompts or explicitly defined part-level concepts [9, 11]—to provide finer-grained semantic guidance during adaptation. However, introducing such concepts without verifying their visual evidence can itself lead to concept-level hallucination, where the model confidently predicts concepts that are not actually visible. Prior work mainly focuses on object-level hallucination, while part-level concepts (e.g., ears, legs, tails) remain underexplored. However, such parts are often occluded or absent in real images, making them particularly prone to hallucination. We hypothesize that part-level hallucination is partly caused by a structural issue in contrastive learning: class-associated concepts are treated as positive for all images of the class, regardless of visibility, reinforcing non-visible parts. To overcome this limitation, we propose visibility-aware contrastive learning, which incorporates per-image concept visibility to modulate supervision. Our method promotes alignment with visually supported concepts while suppressing responses to non-visible ones. As there are no suitable benchmarks for evaluating concept visibility, we additionally construct new evaluation datasets for this task. Experimental results demonstrate that our method significantly improves concept separation (AUROC and FPR95) while improving classifi-

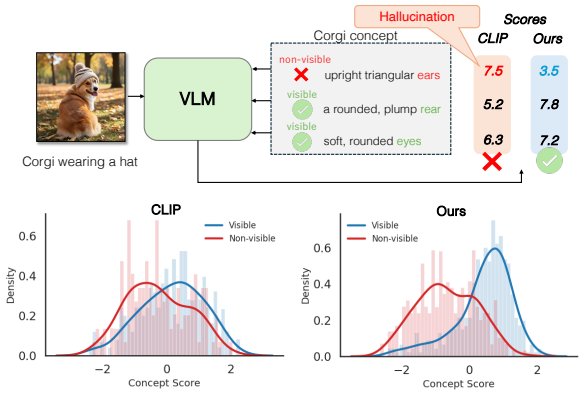

Figure 1. Concept-level hallucination in VLMs. Models assign high confidence to non-visible concepts and ignore visibility. Our method suppresses such errors via visibility-aware learning.

Code is available at https://github.com/shijo-hikaru/visible-concept-learning.

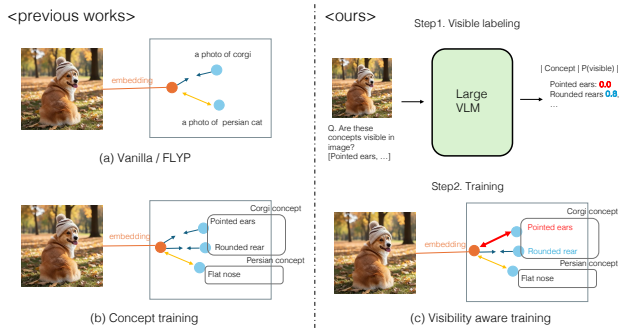

Figure 2. Overview of our visibility-aware concept learning framework. Conventional approaches either rely on class-level alignment or treat all concepts equally, leading to incorrect emphasis on non-visible attributes. In contrast, our method estimates per-image concept visibility, and incorporates it into contrastive learning to promote alignment with visible concepts while suppressing non-visible ones.

cation accuracy. Our contributions are summarized as follows: (1) We propose visibility-aware contrastive learning that weights concept supervision using per-image visibility probabilities. (2) We construct new evaluation datasets for concept visibility prediction, enabling reliable evaluation of visible and non-visible concepts. (3) Experiments demonstrate that our method significantly improves AUROC and FPR95 while improving classification accuracy.

2. Method

2.1. Concept Generation

For each class y, we generate a set of semantic concepts $\mathcal{C}_y = \{c_{y,k}\}_{k=1}^{K}$, where each concept describes an attribute or part-level property of the class. Following prior work [11], these concepts are automatically generated using a large language model (LLM) given the class name. The generated concepts are shared across all images of the class and are encoded using the CLIP text encoder.

2.2. Concept Visibility Labeling

While the generated concepts provide rich semantic descriptions for each class, not all concepts are visually supported in every image. To model this discrepancy, we introduce *concept visibility scores*, which estimate how likely a concept is visible in a given image. Given an image x_i and a class-specific concept $c_{y_i,c}$, we obtain a visibility probability $p_{i,c} \in [0,1]$, indicating the likelihood that the concept is visually present. These scores are obtained by querying a large vision–language model (LVLM), which evaluates whether visual evidence corresponding to each concept exists in the image. The resulting probabilities are treated as fixed pseudo-labels and are used only during training. Details of the prompting strategy and label generation procedure are provided in the Appendix 8.1.

2.3. Contrastive Loss with Soft Concept Visibility

We consider a mini-batch of size B. Let $\mathbf{v}_i \in \mathbb{R}^D$ denote the image embedding of the i-th sample, and $\mathbf{t}_{j,c} \in \mathbb{R}^D$ denote the text embedding of the c-th concept of class j. The ground-truth class index of image i is denoted by y_i. The similarity between an image and a concept is defined as $s_{i,j,c} = \frac{\mathbf{v}_i^\top \mathbf{t}_{j,c}}{\tau}$, where τ is a temperature parameter.

Concept Visibility. Each image i is associated with a visibility probability $p_{i,c} \in [0,1]$, which represents the likelihood that the c-th concept of the target class y_i is visually present in the image.

Negative Concepts. For each image i, inter-class negatives are defined using only the concepts associated with other samples in the mini-batch whose class labels differ from y_i:

$$\mathcal{N}_i^{\text{inter}} = \{(b,c) \mid b \in \{1,\dots,B\},\ y_b \neq y_i,\ c \in \{1,\dots,C\}\}. \tag{1}$$

Our objective explicitly promotes alignment with visible target concepts weighted by $p_{i,c}$, while suppressing invisible ones through an intra-class penalty. Specifically, we define the per-sample loss as

$$\mathcal{L}_{\text{ctr}}^{(i)} = -\frac{1}{Z_i} \sum_{c=1}^{C} p_{i,c} \log \frac{\exp(s_{i,y_i,c})}{\mathcal{D}_{i,c}}, \tag{2}$$

where the denominator $\mathcal{D}_{i,c}$ incorporates both inter-class negatives and intra-class suppression:

$$\begin{aligned} \mathcal{D}_{i,c} = \exp(s_{i,y_i,c}) + \sum_{(j,c') \in \mathcal{N}_i^{\text{inter}}} \exp(s_{i,j,c'}) \\ + \lambda_{\text{inv}} \sum_{c'=1}^{C} (1 - p_{i,c'}) \exp(s_{i,y_i,c'}). \end{aligned} \tag{3}$$

Here, λ_{inv} controls the strength of penalizing similarities to concepts that are likely invisible. The normalization term is defined as $Z_i = \sum_{c=1}^{C} p_{i,c}$. If Z_i becomes too small (e.g., when all concepts are predicted invisible), we skip such samples during training by discarding cases where $Z_i < \epsilon$, with a small constant $\epsilon > 0$. In practice, such samples are filtered out prior to training and therefore do not affect the training process. Finally, the overall contrastive objective is computed by averaging the per-sample loss over the mini-batch:

$$\mathcal{L}_{\text{ctr}} = \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}_{\text{ctr}}^{(i)}. \tag{4}$$

Discussion. Eq. (2) introduces two key design choices. First, concept contributions are weighted by the visibility

score $p_{i,c}$, allowing the model to learn from uncertain visibility supervision. Second, the formulation includes an intra-class suppression term weighted by $(1 - p_{i,c})$, which discourages high similarity to concepts predicted to be invisible. Together, these mechanisms reduce concept hallucination while preserving standard inter-class discrimination.

2.4. Part-Aware Augmentation

In many fine-grained datasets, discriminative object parts are often clearly visible, and cases where class-related concepts are absent are relatively rare. To increase such cases during training, we introduce a strong augmentation strategy based on Open-Vocabulary Object Detector (OVOD) [2]. OVOD detects semantically meaningful object parts, which are used to guide region-aware augmentations such as masking and cropping. These transformations intentionally create samples where certain concepts become partially or completely invisible. Concept visibility scores are assigned after augmentation using a large vision–language model (LVLM), allowing newly invisible concepts to be treated as negative signals during training. Further details are provided in the Appendix 8.2.

2.5. Inference Score

At inference time, we compute class scores using the similarities between the image and concept embeddings. Let $\{S_m^k\}_{m=1}^{l_k}$ denote the similarity logits between the input image and the l_k concept prompts of class k. The predicted class is obtained by averaging the top-5 logits within each class, which reduces sensitivity to noisy concepts:

$$\hat{k} = \arg\max_k \frac{1}{5} \sum_{m \in \mathrm{Top5}(\{S_m^k\}_{m=1}^{l_k})} S_m^k. \tag{5}$$

3. Experiments

We evaluate the proposed method from two perspectives: (i) concept visibility prediction, and (ii) generalization to unseen classes.

Concept Visibility Evaluation on CUB. To evaluate suppression of visually non-visible concepts, we conduct experiments on the CUB-200-2011 dataset [13]. Using part-level keypoint annotations, we construct semantic concepts corresponding to bird parts and evaluate concept-level predictions. In addition to the original test set (CUB-Vis), we construct a new evaluation dataset, *CUB-Vis50*, where object parts are intentionally occluded via cropping and masking, followed by human annotation. We report AUROC and FPR95 computed from concept prediction scores, together with classification accuracy. This setting allows us to measure whether the model suppresses concepts that are not visually supported. Details of the dataset construction and evaluation protocol are provided in the Appendix 6.

	CUB / CUB-Vis			CUB-Vis50	
Method	Acc ↑	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
Zero-shot	57.52	58.16	85.01	69.45	60.05
FLYP	74.95	56.01	86.39	68.16	62.55
DEAL	57.44	58.11	85.00	65.46	68.57
Ours($p = 1.0$)	79.17	49.97	92.19	57.79	74.82
Ours	**80.74**	**68.18**	**74.24**	**83.28**	**41.84**

Table 1. Concept visibility prediction performance on the CUB datasets. Higher AUROC and lower FPR95 indicate better separation between visible and non-visible concepts.

Concept Visibility Evaluation on Oxford-IIIT Pet. We additionally construct a concept visibility evaluation dataset on the Oxford-IIIT Pet dataset [7]. Specifically, we select two representative images from each of the 37 classes and manually annotate concept visibility for each image, resulting in a total of 74 annotated images. To increase the number of non-visible concepts, we apply region-based masking and cropping operations, similar to the CUB setting, to intentionally remove visual evidence for specific attributes. This dataset enables evaluation of whether models can correctly suppress non-visible concepts in a different domain. Further details of the dataset construction are provided in the Appendix 7.

Zero-Shot Evaluation on Unseen Classes. We evaluate generalization by testing on unseen classes. For each dataset, classes are split into seen and unseen subsets, and models are evaluated on unseen categories. We report classification accuracy together with AUROC and FPR95 on the corresponding visibility benchmarks. This setting verifies whether visibility-aware learning improves concept-level reasoning beyond the training categories.

Implementation Details We use a large language model (GPT-5 API) to generate class-specific textual concepts for each dataset. For concept visibility estimation, we employ a large VLM (Qwen3-VL-4B [1]), which predicts the probability that each concept is visually present in a given image. We adopt CLIP ViT-B/16 as the backbone model and fine-tune it on downstream datasets using the proposed method. All models are trained for 10 epochs. Unless otherwise specified, we set $\lambda_{inv} = 6.0$ for all experiments. To analyze the effect of visibility modeling, we compare our method with a variant that treats all concepts as fully visible (denoted as *Ours (p = 1)*). Detailed implementation is provided in Appendix 9.

3.1. Accuracy and Concept Grounding

We evaluate our method's performance on downstream recognition tasks, focusing on both overall classification ac-

Method	Pet / Pet-Vis		
	ACC ↑	AUROC ↑	FPR95 ↓
Zero-shot	89.51	60.86	71.27
FLYP	91.31	60.88	69.91
DEAL	89.81	60.20	71.76
Ours (p=1)	94.11	58.42	75.21
Ours	**94.66**	**83.24**	**44.56**

Table 2. Results on Pet dataset.

Method	CUB / CUB-Vis50			Pet / Pet-Vis		
	ACC ↑	AUROC ↑	FPR95 ↓	ACC ↑	AUROC ↑	FPR95 ↓
Zero-shot	67.28	69.45	60.05	92.34	60.86	71.27
FLYP	67.60	69.13	59.36	93.70	58.68	70.67
DEAL	50.07	62.51	68.69	90.11	54.83	75.03
Ours ($p = 1$)	67.55	61.14	68.93	94.92	55.13	78.70
Ours	**67.67**	**77.53**	**47.71**	**95.19**	**78.53**	**47.57**

Table 3. Zero-shot evaluation on unseen classes. For each dataset, classes are evenly split, where the first half is treated as unseen and the remaining half as seen. ACC is measured on the standard test splits of CUB and Pet (unseen classes). AUROC and FPR95 are evaluated on dedicated visibility benchmarks (CUB-Vis50 and Pet-Visible (Unseen)), where concept visibility is annotated.

curacy and the model's ability to ground predictions in visible evidence. Tables 1 and 2 summarize the results on the CUB and Pet datasets.

Classification Performance. Our visibility-aware fine-tuning achieves the best classification accuracy among all compared methods. FLYP improves classification accuracy over zero-shot CLIP by leveraging class-name supervision, but shows limited performance gains. Incorporating concept-level supervision leads to further improvements. Explicitly modeling concept visibility yields the best classification performance.

Mitigation of Concept-Level Hallucination. The zero-shot CLIP baseline shows limited ability to distinguish visible and non-visible concepts. FLYP improves classification accuracy, but because it does not perform conceptual-level learning, AUROC and FPR95 do not change compared to the zero-shot baseline. Methods that incorporate concept-level supervision without visibility modeling (e.g, DEAL and our variant with $p = 1.0$) actually degrade visibility separation performance. This confirms that forcing a model to align image embeddings with non-visible parts reinforces hallucination and weakens semantic consistency. Our proposed method significantly outperforms all benchmarks in AUROC and FPR95. On the CUB-Vis50, our method improves AUROC by +13.83 over zero-shot CLIP (69.45 → 83.28). On the Pet-Vis benchmark, it also achieves a consistent improvement of +22.38 in AUROC (60.86 → 83.24).

These results demonstrate that incorporating concept visibility enables interpretable concept-level predictions grounded in visible evidence, while improving both concept grounding and classification accuracy.

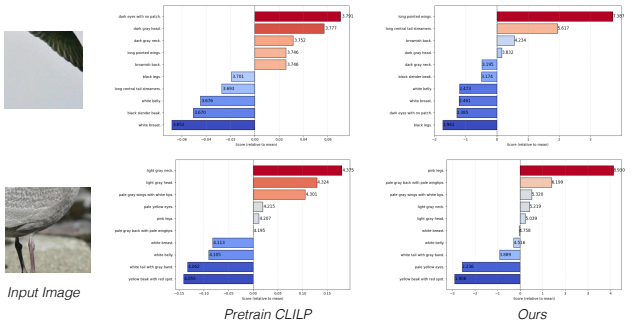

Figure 3. Comparison of concept scores between pretrained CLIP and our method. CLIP assigns high scores to non-visible concepts, while our method suppresses them and focuses on visible concepts.

3.2. Zero-Shot Performance on Unseen Classes

To assess the robustness of our visibility-aware representations, we conduct zero-shot evaluation on unseen categories. As shown in Table 3, our method achieves higher AUROC and lower FPR95 than prior approaches. This demonstrates that modeling concept visibility enables more reliable semantic grounding even for novel categories. Furthermore, our method achieves higher classification accuracy on unseen classes, indicating that visibility-aware learning leads to more robust and transferable visual-language representations.

3.3. Qualitative evaluation

Figure 3 shows that the pretrained CLIP assigns high scores to concepts that are not visually present. In the top example, where only the wing is visible, CLIP produces high responses for unrelated concepts such as eyes. Similarly, in the bottom example, although only the body and legs are visible, CLIP incorrectly assigns the highest scores to concepts such as neck and head. In contrast, our method assigns higher scores to concepts that are actually visible in the image. This demonstrates that our visibility-aware training successfully aligns concept scores with visual evidence, reducing concept-level hallucination.

4. Conclusion

In this paper, we proposed a visibility-aware concept learning framework for VLM. We additionally construct two visibility annotation benchmarks. Experiments demonstrate consistent improvements in AUROC and FPR95, highlighting the importance of visibility-aware learning for reliable interpretation at the concept-level.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. 3, A6

[2] Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14987–14997, 2025. 3

[3] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19338–19347, 2023. 1, A1

[4] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1

[5] Yufang Liu, Tao Ji, Changzhi Sun, Yuanbin Wu, and Aimin Zhou. Investigating and mitigating object hallucinations in pretrained vision-language (clip) models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18288–18301, Miami, Florida, USA, 2024. Association for Computational Linguistics. 1, A1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. A6

[7] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505. IEEE, 2012. 3, A3

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1

[9] Oindrila Saha, Grant Van Horn, and Subhransu Maji. Improved zero-shot classification by adapting vlms with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17542–17552, 2024. 1, A1

[10] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. A8

[11] Li Tang, Mengmeng Ma, and Xi Peng. Deal: Disentangle and localize concept-level explanations for vlms. In *European Conference on Computer Vision*, pages 383–401. Springer, 2025. 1, 2, A1

[12] Yingtian Tang, Yutaro Yamada, Yoyo Zhang, and Ilker Yildirim. When are lemons purple? the concept association bias of vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14333–14348. Association for Computational Linguistics, 2023. 1, A1

[13] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology Technical Report*, 2011. 3, A1

[14] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. Oral Presentation. 1, A1

[15] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023. A7

Seeing Only What Exists: Visibility-Aware Contrastive Learning for Concept-Level Hallucination in Vision–Language Models

Supplementary Material

5. Related Work

5.1. Concept-Aware Adaptation of Vision–Language Models

To improve the downstream performance of vision–language models (VLMs), a number of works have explored concept-aware adaptation strategies that leverage additional textual descriptions beyond class names. Early studies showed that continuing contrastive learning on downstream datasets can be more effective than fine-tuning image encoders with standard cross-entropy loss, as exemplified by FLYP [3].

More recent approaches incorporate large language models (LLMs) to enrich textual supervision. Methods such as CLIP-A and CLIP-A-self [9] generate attribute-level descriptions or expanded prompts to better capture fine-grained semantic information, leading to improved zero-shot and few-shot classification performance. Other works introduce pseudo-captions or auxiliary textual descriptions to enhance the alignment between visual features and semantic concepts.

DEAL [11] explicitly decomposes class representations into multiple semantic concepts and encourages alignment between image regions and concept-level text embeddings. By modeling concept-level interactions, DEAL improves fine-grained recognition and interpretability of VLM predictions. However, it implicitly assumes that the introduced concepts are visually grounded, and does not explicitly address whether a concept is actually observable in a given image.

Despite their effectiveness, most existing concept-aware adaptation methods treat all introduced concepts as positive supervision, regardless of their visual presence. In contrast, our approach explicitly models concept visibility during training, enabling the model to suppress non-visible concepts and produce more reliable concept-level predictions.

5.2. Hallucination in Vision–Language Models

Hallucination in vision–language models (VLMs) refers to the phenomenon where models assign high confidence to objects, attributes, or concepts that are not visually present in an image. Recent studies have shown that this issue arises not merely from missing visual evidence, but from structural biases induced by contrastive learning and statistical co-occurrence between concepts.

Tang et al. [12] demonstrated that VLMs are susceptible to concept association bias, leading to visually inconsistent attribute predictions. Liu et al. [5] analyzed object-level hallucinations in CLIP and reported that frequently co-occurring objects tend to receive high confidence even when absent from the image.

Yuksekgonul et al. [14] showed that VLMs often behave in a bag-of-words manner, being insensitive to word order and compositional structure, and proposed NegCLIP by extending contrastive learning. While these approaches improve compositionality or mitigate specific hallucination patterns, they do not explicitly consider whether a concept is visually observable in the image.

Overall, prior work has established the presence and causes of hallucination in VLMs, but has largely focused on object- or attribute-level errors. In contrast, part-level concepts are often partially or fully unobservable due to occlusion, viewpoint, or resolution, yet remain strongly associated with object categories. Our work addresses this gap by explicitly modeling concept visibility and suppressing non-visible concepts during training.

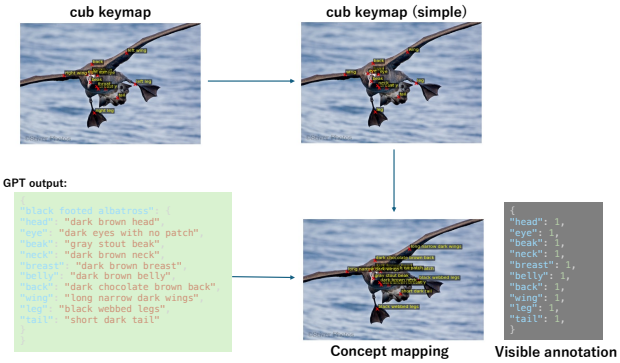

Figure 4. Creation of CUB-Vis with concept-level visibility annotations.

6. Dataset Analysis and Construction

6.1. Concept Visibility Evaluation on CUB

The CUB dataset [13] provides detailed part annotations based on keypoints. We used these annotations to construct concept-level visibility labels.

Figure 4 illustrates the process of linking keypoint annotations with concept descriptions.

Step 1: Part Unification The original CUB dataset contains multiple redundant keypoints (e.g., *left wing* and *right wing*). To simplify the concept space, we merge symmetric or redundant parts into unified categories.

After merging, we define the following ten semantic parts:

```
[head, eye, beak, neck, breast, belly
 (underparts), back, wing, leg, tail]
```

This unified representation allows us to map fine-grained keypoints into a consistent set of concept-level parts.

Step 2: Concept Generation using GPT-5 For each bird class in CUB, we generate semantic descriptions for the unified parts using GPT-5 (text-only). Given a class name, GPT-5 produces one short visual concept for each semantic part. This procedure enables scalable concept-level annotation without requiring manual concept design for every class. The prompt used for concept generation is shown below.

Prompt

You are an expert ornithologist and fine-grained vision researcher.

Task. Given a bird class name, generate *visual concepts* for that bird, conditioned on the following 10 body parts:

> [”head”, ”eyes”, ”beak”, ”neck/throat”, ”breast”, ”belly”, ”back”, ”wings”, ”legs”, ”tail”]

Rules.

1. For each body part, always output a value.
2. Each value must be a short natural phrase including the body part name.
3. Each description should be at most 7 words.
4. Avoid redundancy and emphasize the most distinctive parts.
5. Output strictly in JSON format with all 10 keys present.
6. Output only a single JSON object and nothing else. No explanations and no backticks.

Example (class: `"Black footed albatross"`):

```
{
  "head": "dark brown head",
  "eyes": "dark eyes with no patch",
  "beak": "grayish stout beak",
  "neck/throat": "dark brown neck",
  "breast": "dark brown breast",
  "belly": "dark brown belly",
  "back": "dark chocolate brown back",
  "wings": "long narrow dark wings",
  "legs": "black webbed legs",
  "tail": "short dark tail"
}
```

Now generate the output for: `{cls}`

For example, for the class *Black-footed Albatross*, GPT-5 generates concepts such as *dark brown head*, *grayish stout beak*, and *long narrow dark wings*.

Step 3: Mapping Keypoints to Concepts Finally, we map the original CUB keypoint annotations to the generated concepts.

Each generated concept corresponds to one unified semantic part. If the corresponding part annotation exists and is visible in an image, the concept is labeled as *visible*; otherwise, it is labeled as *non-visible*.

Through this mapping, we obtain concept-level visibility labels from the original part annotations, enabling evaluation of whether a model can correctly determine the visibility of class-specific concepts.

6.2. Bias in the Original CUB Test Set (CUB-Vis)

We first analyze the visibility distribution of part annotations in the original CUB test set (5,794 images). The number of images in which each part is visible is summarized in Table 4.

Part	Visible Ratio
Head	5774 / 5794
Eye	5623 / 5794
Beak	5776 / 5794
Neck	5777 / 5794
Breast	5394 / 5794
Belly	5091 / 5794
Back	4410 / 5794
Wing	5633 / 5794
Leg	4824 / 5794
Tail	5397 / 5794

Table 4. Visibility statistics of part annotations in the original CUB test set.

These statistics show that most parts are visible in the majority of images. For example, *head*, *beak*, and *neck* are visible in almost all images, while parts such as *legs* are more frequently occluded. This leads to a strong imbalance across parts, where several parts are almost always visible.

As a result, evaluating concept visibility prediction using the original dataset becomes difficult, since a model can achieve high performance simply by predicting most parts as visible.

6.3. Constructed Evaluation Dataset (CUB-Vis50)

To mitigate this bias, we construct a new evaluation dataset.

From the first 50 classes of CUB, we randomly select two images per class, resulting in a total of 100 images. Each image is manually annotated for concept visibility.

To increase the number of occluded parts, we generate modified images by cropping or masking regions based on ground-truth keypoints from the original CUB annotations.

Each part is annotated using the following labels:

- 1 : clearly visible

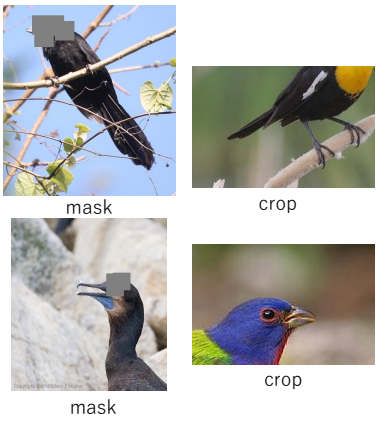

Figure 5. Examples of the constructed evaluation dataset. Images are generated by cropping or masking regions based on ground-truth keypoints to control part visibility.

- 0 : completely invisible
- 0.5 : ambiguous or partially visible

Ambiguous labels (0.5) are excluded from evaluation.

6.4. Dataset Statistics

The visibility statistics of the constructed dataset are shown in Table 5.

Part	Visible Ratio
Head	50 / 100
Eye	44 / 100
Beak	53 / 100
Neck	59 / 100
Breast	51 / 100
Belly	49 / 100
Back	39 / 100
Wing	69 / 100
Leg	40 / 100
Tail	47 / 100

Table 5. Visibility statistics of the constructed evaluation dataset.

A total of 27 ambiguous annotations (0.5) appear among 1,000 part labels.

Compared to the original CUB test set, the new dataset contains significantly more occluded parts and exhibits a more balanced visibility distribution across parts.

This makes the dataset more suitable for evaluating concept visibility prediction. Figure 5 shows samples of new dataset. The constructed dataset will be publicly released on GitHub.

6.5. Training and Evaluation Protocol

Using the class-specific concepts constructed as described above, we train the model to align each image with its corresponding semantic concepts. Each class is associated with 10 concepts, corresponding to the unified parts: *head*, *eye*, *beak*, *neck*, *breast*, *belly*, *back*, *wing*, *leg*, and *tail*.

At test time, given an input image, we compute confidence scores for the 10 concepts belonging to its ground-truth class. We then examine whether the model assigns:

- high confidence to concepts whose corresponding parts are visible, and
- low confidence to concepts whose corresponding parts are non-visible.

More formally, for each test image, the concepts corresponding to visible parts are treated as **positive** samples, while the concepts corresponding to non-visible parts are treated as **negative** samples. Ambiguous labels (0.5) are excluded from evaluation.

We evaluate concept visibility prediction using **AUROC** and **FPR95**. A good model should produce higher confidence scores for visible concepts than for non-visible concepts, resulting in high AUROC and low FPR95.

This evaluation protocol directly measures whether the model can distinguish between visually grounded concepts and concepts that are not supported by visible evidence.

7. Concept Visibility Evaluation on Pet

To verify the generality of our concept visibility evaluation framework, we extend it to the Oxford-IIIT Pet dataset [7].

Unlike CUB, the Pet dataset does not provide part-level keypoint annotations. Therefore, we construct concept visibility labels manually.

Semantic Attributes To capture visually distinguishable characteristics in pet images, we define the following ten semantic attributes:

```
[ear shape, muzzle length, face shape,
 eye shape, fur length, coat pattern,
 coat color, tail shape, leg length,
          body proportion]
```

These attributes reflect fine-grained appearance differences across dog and cat breeds and are directly observable from images.

Concept Generation For each class, we generate visual concepts for each attribute using GPT-4. Given a class name, the model outputs a short, human-interpretable description per attribute.

Examples include *small upright ears*, *short smooth coat*, and *curled tail*. This results in 10 concepts per class.

Visibility Bias and Its Mitigation In the original Pet dataset, face-related attributes (e.g., ears, eyes, and muzzle) are visible in most images.

Under this setting, a model can achieve high performance by simply predicting these attributes as always visible. Thus, it is not suitable for evaluating concept visibility.

To address this issue, we control the visibility distribution.

Specifically, we apply cropping and masking operations based on object regions predicted by LLMDet. These operations remove visual evidence for selected attributes, increasing the number of non-visible concepts.

As a result, the distribution of visible and non-visible attributes becomes more balanced, allowing us to evaluate whether a model can recognize concept visibility based on visual evidence rather than dataset bias.

Manual Annotation We construct the evaluation dataset by selecting two images per class from all 37 classes, resulting in 74 images.

These include both original and modified (cropped/masked) images.

For each image, we manually annotate visibility for all 10 attributes. Labels are defined as:

- 1: visible
- 0: not visible
- 0.5: ambiguous

Ambiguous labels are excluded from evaluation.

Compared to CUB, Pet includes more appearance-based attributes (e.g., color and texture), making visibility judgment more subjective. However, this also enables evaluation of concept understanding without relying on explicit spatial annotations.

Concept	Visible Count
Ear Shape	41 /74
Muzzle Length	35 / 74
Face Shape	34 / 74
Eye Shape	28 / 74
Fur Length	71 / 74
Coat Pattern	53 / 74
Coat Color	60 / 74
Tail Shape	21 / 74
Leg Length	36 / 74
Body Proportion	33 / 74

Table 6. Visibility statistics of the constructed evaluation dataset

Dataset Characteristics Table 6 summarizes the visibility statistics of the constructed dataset. A total of 58 ambiguous annotations (0.5) appear among 740 part labels. Compared to the original Pet dataset, our dataset contains more occluded attributes, resulting in a less biased visibility distribution. In contrast to CUB, where visibility is defined based on the presence of parts, our definition is attribute-aware. That is, an attribute is considered *non-visible* not only when it is occluded, but also when the observed attribute does not match the class-specific concept (e.g., a different coat color). This makes the task more challenging, as it requires models to capture fine-grained visual attributes rather than merely detecting the existence of parts.

Figure 6. Examples of the constructed Pet-Vis. Images are generated by cropping or masking to control part visibility.

Evaluation Protocol We follow the same evaluation protocol as in CUB. For each image, concepts corresponding to visible attributes are treated as positive, while those corresponding to non-visible attributes are treated as negative. We evaluate performance using AUROC and FPR95. This evaluation measures whether a model can correctly distinguish between visually grounded concepts and those not supported by visual evidence.

8. Detail of Methods

8.1. Create visible label.

LVLM for Visibility Estimation We obtain concept visibility scores using a large vision–language model (LVLM). In our implementation, we use **Qwen3-VL-4B**, which provides a good trade-off between accuracy and inference speed. Given an input image and a set of class-specific concepts, the LVLM estimates the probability that each concept is visually present.

Prompt Design We design a prompt that explicitly instructs the model to rely only on visual evidence and to output a probability for each concept. The prompt used in our experiments is shown below.

Prompt

You are given an image and a list of possible visual attributes:
{concept_list}

Your task is to estimate the probability that each attribute is **visibly present** in the image.
Base your decision strictly on visible pixels. Do not guess based on prior knowledge about the object class.

Task
For each attribute, output the probability that it is visible in the image.

Visibility rules
- Clearly visible → probability close to 1.0
- Clearly not visible → probability close to 0.0
- Uncertain → intermediate probability

Output format (STRICT)
Return a table with two columns:

```
|Attribute|P(visible)|
```

- P(visible) must be a decimal between 0.0 and 1.0
- Output only the table

8.2. Concept-Aware Data Augmentation Details

To generate diverse concept visibility conditions, we introduce a region-based data augmentation strategy using an Open-Vocabulary Object Detector (OVOD). This procedure is applied offline to the training data prior to model training. First, OVOD (LLMDet) detects semantically meaningful regions in an image. Next, we apply cropping and masking operations to the detected regions. We further apply mild appearance transformations to the entire image. Finally, we recompute concept visibility scores for the augmented images using an LVLM.

OVOD-based Region Detection To obtain semantically meaningful regions, we employ LLMDet. Under the open-vocabulary setting, LLMDet outputs bounding boxes corresponding to objects or their parts. For each image, we input multiple class-specific concepts as queries, and use the resulting bounding boxes as candidate regions for augmentation. We use only the region information and do not utilize detection labels.

Region-aware Data Augmentation We apply the following augmentations to the detected regions in a stochastic manner. The overall procedure is summarized in Algorithm 1.

(1) Masking Selected regions are filled with a constant color to remove the corresponding visual evidence.

(2) Cropping We crop the image such that selected regions are partially or fully removed, eliminating visual evidence associated with specific concepts.

(3) Appearance Transformation We apply color jitter to the entire image with a certain probability. This improves robustness to variations in illumination and color. We do not apply local appearance transformations, as they may degrade image realism.

Each augmentation is applied independently, and multiple operations may be combined for a single image.

Algorithm 1: Concept-Aware Region Augmentation

Input: Image x, detected boxes $\mathcal{B}$ with labels $\mathcal{L}$, number of augmentations N
Output: Augmented images $\{\tilde{x}_i\}_{i=1}^{N}$
Initialize $\mathcal{X} \leftarrow \emptyset$
for $i = 1$ *to* N **do**
  Initialize $\tilde{x}_i \leftarrow x$
  Sample operation $a \in \{\text{mask}, \text{crop}\}$
  if $a =$ *mask* **then**
    Select a concept c and its boxes $\mathcal{B}_c$
    Fill $\mathcal{B}_c$ in $\tilde{x}_i$ with a constant color
  else
    Sample a crop region r
    Crop $\tilde{x}_i$ using r such that at least one concept remains visible and at least one is removed
  Sample $u \sim \mathcal{U}(0, 1)$
  if $u < p$ **then**
    Apply global color jitter to $\tilde{x}_i$
  Add $\tilde{x}_i$ to $\mathcal{X}$
return $\mathcal{X}$

Visibility Re-labeling After augmentation, we recompute concept visibility scores using an LVLM. This allows concepts that become invisible due to augmentation to be treated as negative samples during training.

Effect Our approach enables the generation of training samples where concepts are intentionally non-visible, a scenario that is rare in the original dataset. As a result, the distribution of visible and non-visible concepts becomes more diverse, forcing the model to rely on visual evidence rather than dataset bias. Moreover, since our method selectively manipulates regions corresponding to concepts, it can generate samples in which specific concepts are intentionally occluded. This leads to semantically meaningful changes in concept visibility, rather than arbitrary corruption of visual information. Table 7 compares our augmentation with a fully random crop-and-mask strategy. Our concept-aware augmentation leads to substantial improvements in classification accuracy, AUROC, and FPR95, demonstrating the effectiveness of selectively manipulating concept-relevant regions.

9. Implementation Details

Concept-Aware Data Augmentation. For each input image, we generate $N = 3$ augmented samples by randomly

Method	CUB / CUB-Vis50		
	Seen ↑	AUROC ↑	FPR95 ↓
Random	77.46	77.38	46.41
LLMdet-based	80.74	83.28	41.84

Table 7. Comparison between random augmentation and our LLMDet-based concept-aware augmentation. Our method significantly improves classification accuracy, AUROC, and FPR95.

applying region-based operations (masking or cropping) guided by concept-level detections. We additionally apply global color jitter with probability $p = 0.3$. The augmented samples are combined with the original training images for optimization. This augmentation explicitly increases the diversity of visible and non-visible concepts, which is critical for learning visibility-aware representations.

Training Setup We adopt CLIP ViT-B/16 as the backbone model. All models are trained using the AdamW [6]. Unless otherwise specified, all experiments are conducted using a mixture of original training images and augmented samples. We generate concept visibility labels using Qwen3-VL-4B [1] for both the original training images and the augmented images. For training, the dataset is split into training and validation sets with a ratio of 8:2. The best model is selected based on validation accuracy. We train the model for 10 epochs and evaluate the selected checkpoint on the test set. We use a cosine learning rate schedule with a linear warm-up over the first 10% of total steps. During both training and evaluation, textual concepts are formatted as *"{classname} with {concept}."*. This prompt formulation encourages the model to associate class-specific attributes with visual evidence. The temperature parameter is initialized to 0.1 and optimized during training. We use different hyperparameters for each dataset, as summarized below.

	CUB	PET
lr main	2×10^{-6}	1×10^{-6}
lr proj	5×10^{-6}	2×10^{-6}
weight decay (main)	1×10^{-3}	1×10^{-3}
weight decay (proj)	1×10^{-2}	1×10^{-2}
weight decay (temperature)	1×10^{-6}	
temperature init	0.1	
epochs	10	
warm-up ratio	10% of total steps	
β_1, β_2	$(0.9, 0.98)$	
ϵ	1×10^{-6}	
optimizer	AdamW [6]	

Table 8. Training hyperparameters for each dataset

Ablation Setting. To analyze the effect of visibility modeling, we compare our method with a variant that treats all concepts as fully visible, denoted as *Ours (p = 1)*.

10. Additional Results

10.1. Analysis of LVLM Visible-Label Quality

Model	CUB-Vis50		Pet-Vis	
	AUROC(%)	ACC(%)	AUROC(%)	ACC(%)
Qwen3-VL-2B	70.91	2.00	83.96	18.91
Qwen3-VL-4B	79.53	17.00	86.08	29.73
Qwen3-VL-8B	76.89	15.00	87.82	24.32

Table 9. LVLM performance on concept visibility benchmarks. Accuracy (ACC) is computed per image as a strict separation metric: a prediction is considered correct only if the minimum score of visible concepts exceeds the maximum score of non-visible concepts.

Table 9 reports the performance of the large vision–language models (LVLMs) used to generate pseudo-labels for concept visibility. To ensure a fair comparison, we generate visibility scores for the benchmark images using the same prompt as in the main experiments (sec 8.1). We evaluate the LVLM predictions using AUROC and ACC. Here, ACC is defined as the proportion of images for which the minimum score among visible concepts is greater than the maximum score among non-visible concepts, indicating a strict separation between visible and non-visible concepts for the entire image. The results show that the prediction quality varies with model size, and notably, the 4B model outperforms the 8B model in terms of both AUROC and ACC. While the LVLMs achieve relatively high AUROC scores (around 80%), the ACC remains low (approximately 10–30%). This large gap suggests that, although the LVLMs provide a coarse ranking between visible and non-visible concepts, the generated pseudo-labels still contain substantial noise and are far from perfectly reliable.

Table 10 shows that the performance of our method depends on the quality of the generated visibility labels. Although the influence is not strictly proportional, better label quality generally leads to improved performance.

Interestingly, the trained model can outperform the quality of the pseudo-labels used for supervision. For example, while the LVLM (Qwen3-VL-4B) achieves an AUROC of 79.53 on the CUB-Vis50, the model trained with these labels reaches a higher AUROC of 83.28. This indicates that the proposed learning framework is robust to noise in the pseudo-labels and can extract more reliable signals during training.

Overall, these results suggest that, despite the limited accuracy of current LVLMs, visibility-aware learning remains effective and can achieve strong performance. As LVLMs continue to improve, further gains in both concept grounding and classification performance can be expected.

Visible Label	CUB / CUB-Vis50			Pet / Pet-Vis		
	ACC ↑	AUROC ↑	FPR95 ↓	ACC ↑	AUROC ↑	FPR95 ↓
Qwen3-VL-2B	78.18	77.36	48.25	94.45	78.99	50.28
Qwen3-VL-4B	**80.74**	**83.28**	**41.84**	**94.66**	**83.24**	44.56
Qwen3-VL-8B	79.75	82.88	42.62	94.62	81.85	**43.39**

Table 10. Effect of visible label quality generated by different LVLMs on classification and concept visibility prediction performance using CLIP (ViT-B/16).

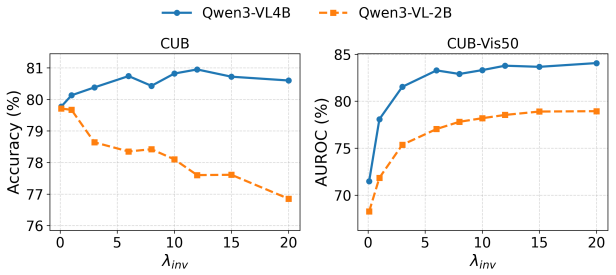

Figure 7. Effect of λ_{inv} on classification accuracy (CUB) and concept visibility separation (CUB-VIS).

10.2. Effect of λ_{inv}.

Figure 7 shows the effect of the hyperparameter λ_{inv}, varied from 0.1 to 20, on both classification accuracy (ACC) and concept-level discrimination (AUROC) on the CUB dataset (CUB-test and CUB50-Vis50). The left panel presents ACC, while the right panel shows AUROC.

We observe that AUROC consistently improves as λ_{inv} increases. This trend is expected, as Eq. 2 explicitly increases the penalty on non-visible concepts, thereby enhancing the separation between visible and non-visible concepts.

In contrast, ACC exhibits different behaviors depending on the quality of visibility labels. When using high-quality labels (4B), increasing λ_{inv} leads to improved classification performance. However, with lower-quality labels (2B), ACC degrades as λ_{inv} increases. This is because noisy visibility labels introduce incorrect supervision, causing the model to suppress features corresponding to actually visible concepts.

These results highlight that the effectiveness of the proposed visibility-aware suppression critically depends on the quality of visibility estimation. When accurate visibility labels are available, a larger λ_{inv} is beneficial for both concept-level discrimination and classification performance.

10.3. Effect of Concept-Aware Data Augmentation

We evaluate the impact of our LLMDet-based data augmentation by comparing performance with and without augmentation. Tables 11 and 12 present the results on CUB and Oxford-IIIT Pet without data augmentation.

Method	**CUB / CUB-Vis**			**CUB-Vis50**	
	Seen ↑	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
Zero-shot	57.52	58.16	85.01	69.45	60.05
FLYP	70.24	56.68	85.53	69.02	59.74
Ours ($p = 1$)	74.36	53.90	89.31	62.78	67.90
Ours	**75.10**	**65.59**	**77.71**	**68.23**	**58.03**

Table 11. Comparison on CUB **without data augmentation**.

Method	**Pet / Pet-Vis**		
	Seen ↑	AUROC ↑	FPR95 ↓
Zero-shot	89.51	60.86	71.27
FLYP	92.16	61.25	70.25
Ours ($p = 1$)	94.30	59.59	70.31
Ours	**94.66**	**66.71**	**57.12**

Table 12. Comparison on Oxford-IIIT Pet and Pet-Vis benchmarks **without data augmentation**.

Method	CUB / CUB-Vis50			Pet / Pet-Vis		
	Acc ↑	AUROC ↑	FPR95 ↓	Acc ↑	AUROC ↑	FPR95 ↓
Zero-Shot	54.09	55.61	88.29	86.56	61.90	73.84
FLYP	66.64	64.21	66.24	89.38	65.11	63.41
Ours (p=1)	72.28	55.57	74.43	91.97	59.84	75.30
Ours	**73.41**	**80.40**	**45.86**	**92.33**	**84.63**	**47.26**

Table 13. CLIP(Vit-B/32)

Even without augmentation, our method consistently achieves the best performance across all metrics, including classification accuracy (ACC), AUROC, and FPR95, outperforming all baselines. This indicates that the proposed visibility-aware learning framework is effective on its own and does not rely on augmentation to yield improvements.

However, when comparing these results with those obtained using data augmentation (Tables 1 and 2), we observe that the overall performance is consistently lower in the absence of augmentation. This suggests that concept-aware data augmentation further enhances the model by exposing it to diverse visibility conditions, leading to improved generalization and more robust concept grounding.

These results demonstrate that while our method is inherently effective, incorporating concept-aware data augmentation provides additional gains and is crucial for achieving optimal performance.

10.4. Other VLMs

We further evaluate our method on different VLM architectures, including CLIP (ViT-B/32) and SigLIP [15]. As shown in Tables 13 and 14, our method consistently exhibits the same trend as observed with CLIP (ViT-B/16).

In all cases, our method achieves the highest classification accuracy while also outperforming baseline methods in AUROC and FPR95. This demonstrates that the effective-

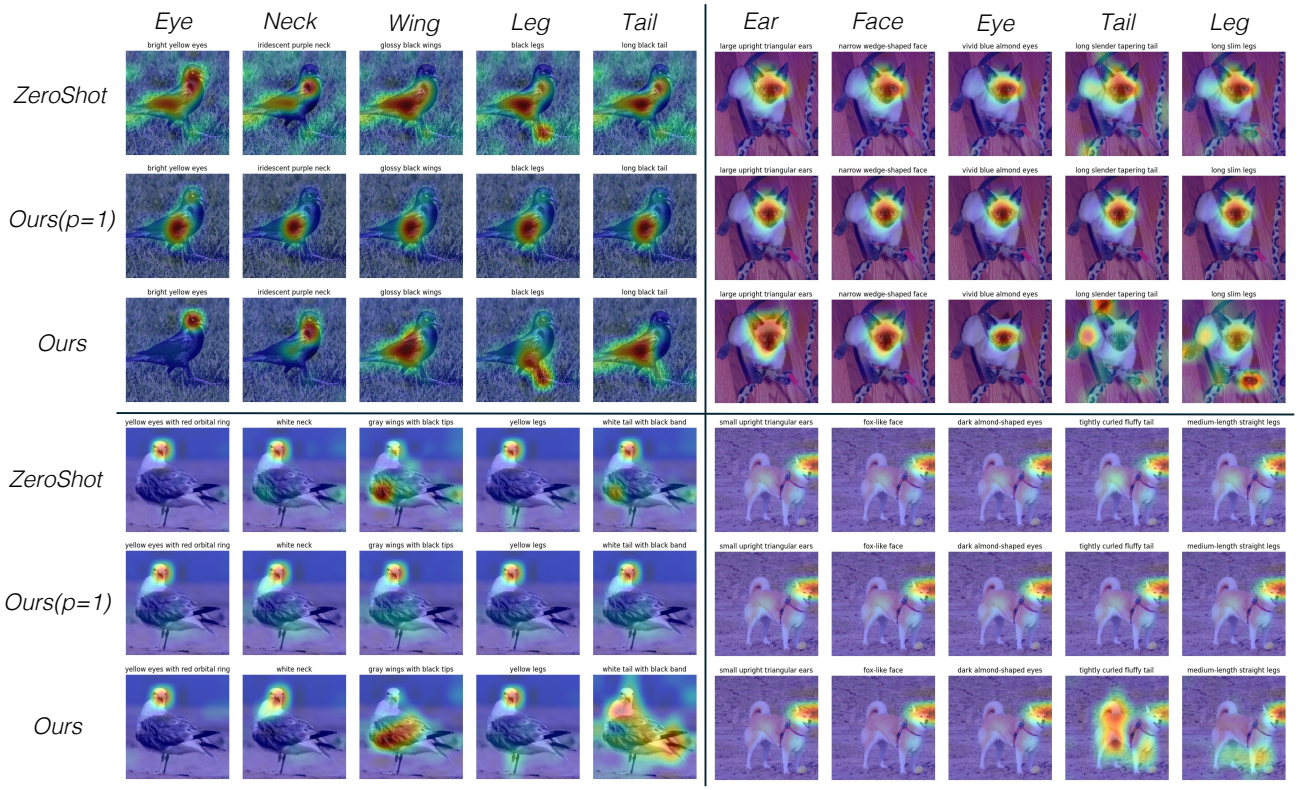

Figure 8. Grad-CAM visualizations of **CLIP (ViT-B/32)** on the test sets of CUB and Oxford-IIIT Pet. Although our model is trained with 10 semantic concepts, we visualize 5 representative concepts for clarity.

	CUB / CUB-Vis50			Pet / Pet-Vis		
Method	Acc ↑	AUROC ↑	FPR95 ↓	Acc ↑	AUROC ↑	FPR95 ↓
Zero-Shot	71.19	67.15	63.44	94.19	66.75	65.13
FLYP	81.39	64.98	66.13	95.39	67.61	66.72
Ours (p=1)	82.53	54.00	78.85	95.23	49.65	86.73
Ours	**83.28**	**83.57**	**42.05**	**95.56**	**89.17**	**30.79**

Table 14. SigLIP

ness of visibility-aware learning is not limited to a specific backbone architecture.

Notably, on the Pet-Vis benchmark, SigLIP achieves a particularly high AUROC of 89, further highlighting the strong generalization capability of our approach across different model architectures

10.5. Visualization of GradCAM

Figure 8 presents Grad-CAM [10] visualizations for pretrained CLIP, Ours ($p = 1$), and the proposed method. When trained with Ours ($p = 1$), the model tends to align with non-visible concepts, leading to degraded localization compared to the pretrained model. In contrast, our method encourages alignment with visually supported concepts, resulting in localization that is preserved or even improved relative to the pretrained CLIP.