Discovering Fine-Grained Visual-Concept Relations by Disentangled Optimal Transport Concept Bottleneck Models

Supplementary Material

A. Dataset

Part-Detection Task dataset description.

A.1. Part-ImageNet

PartImageNet[7] selects 158 classes from ImageNet and groups them into 11 super-categories. For example, `quadruped` super-category contains many animal categories. The part annotations are based on these super-categories, as shown in Table 1.

Id	Name	Part Taxonomy
1	Quadruped	head, body, foot, tail
2	Biped	head, body, hand, foot, tail
3	Fish	head, body, fin, tail
4	Bird	head, body, wing, foot, tail
5	Snake	head, body
6	Reptile	head, body, foot, tail
7	Car	body, tier, side mirror
8	Bicycle	head, body, seat, tier
9	Boat	body, sail
10	Aeroplane	head, body, wing, engine, tail
11	Bottle	body, mouth

Table 1. **PartImageNet part taxonomy** from [7].

Our method focuses on finer-grained part-concept alignment. We use CDL[15] to generate a general concept space shared across all categories and leverage the distribution differences between concepts of different categories to distinguish the fine-grained alignment of different parts. Therefore, we select the super-categories `Quadruped`, `Biped`, `Fish`, `Bird`, `Reptile` from Part-ImageNet, as they share a large portion of the local concept space.

A.2. PASCAL-Part

PASCAL VOC 2010 dataset[6] is a popular dataset used to benchmark Object Detection models in which each image has an annotation file containing bounding box coordinates and class labels for each object. There are 20 classes present in the dataset which can be categorized into 4 super categories namely Person, Animal, Vehicle, Indoor. Training and validation contain 10,103 images while testing contains 9,637 images. The twenty object classes in Pascal VOC 2010 dataset are:

- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv-monitor

The PASCAL-Part Dataset[4]is a set of additional annotations for PASCAL VOC 2010. It provides segmentation masks for each body part of the object. Corresponding to Part-ImageNet, we only select the Animal categories for testing. Due to the overly fine-grained local annotations in PASCAL-Part, we merge some part annotations into a single label. For example, left eye and right eye are merged into eye. The merged parts are as shown in Table.2:

Name	Part Taxonomy
Bird	beak, eye, head, neck, wing, torso, leg, foot, tail
Dog	eye, muzzle, ear, head, neck, torso, leg, foot, tail
Cat	eye, ear, head, neck, torso, leg, foot, tail
Cow	eye, muzzle, ear, head, neck, torso, leg, tail
Sheep	eye, muzzle, ear, head, neck, torso, leg, tail

Table 2. **PASCAL-Part taxonomy**.

B. Metric

To quantitatively evaluate the performance of CBMs in concept inversion, we treat concept inversion as the prediction for a part object detection task. We use two common metrics in object detection, *mAP* and *mIOU*, to quantify the prediction results, thereby characterizing the effectiveness of concept inversion.

mAP(mean Average Precision) is a widely used evaluation metric in object detection tasks, designed to provide a comprehensive assessment of a model's performance across multiple categories. It combines both precision and recall metrics to evaluate how well a model can detect objects and classify them correctly. In our task, we use $mAP_{0.5}$ as the metric to measure the precision of the concept inversion correspondence. In concept inversion where IOU$\geq$0.5, mAP effectively reflects the part-part spurious correlation.

mIOU(mean Intersection over Union) is a widely used evaluation metric in computer vision tasks, particularly in semantic segmentation and object detection. It provides a quantitative measure of the overlap between the predicted and ground truth regions, reflecting the accuracy of the model's localization and segmentation capabilities. The mIOU metric is particularly useful because it balances the

trade-off between precision and recall, ensuring that both the coverage and the accuracy of the predicted regions are evaluated. In our task, we use mIOU as the metric to measure the precision of the inversion relative to the target region. IOU effectively reflects the overlap ratio between the concept inversion and the ground truth.

For each CBM to be evaluated, we use interpretability methods to obtain the activation heatmaps of concepts in the original image space. For example, we use GradCAM in the baseline methods and the assignment matrix in our proposed method to generate these heatmaps.

From the heatmap H_c, we select the top 60% of the highest values to identify the most salient regions. We then compute the minimum bounding rectangle (MBR) that encloses these selected regions. This MBR serves as the predicted bounding box B_c for concept c. Formally, let S_c be the set of coordinates corresponding to the top 60% of the values in H_c. The predicted bounding box B_c is defined as:

$$B_c = \text{MBR}(S_c) \tag{1}$$

where $\text{MBR}(S_c)$ denotes the minimum bounding rectangle that encloses all points in S_c.

C. Construction of Concept Set

According to previous work [8, 15], sharing a generic concept space can improve model performance by encouraging the model to learn more discriminative features from the input local features, while also being more friendly for human intervention. Building on the work of [15], which uses the mutual information between VLM and LLM predictions to filter concepts generated by LLMs, we adopt a similar approach but with a key difference. We use human-defined concepts[13, 14] as the input prompt for In-Context Learning (ICL)[1, 2] to promote the generation of more generic concept sets by the LLM. We've added this part of the code to the supplementary material as well. The basic steps are as follows:

- **Setp 1:** Use the Stanford CoreNLP[9] tool to split the captions in the CC3M[11] dataset into `objects`.
- **Step 2:** Use the `objects` as `category` names to prompt the LLM, with the prompt being(where we can use human defined concepts for In-Context Learning):
- **Step 3:** Encode the concepts using the text encoder of CLIP ViT-L/14 and filter out duplicate concepts with a similarity score greater than 0.9.
- **Step 4:** Given a concept and an image-caption dataset, let the X variable denotes the image-concept similarity as measured by a CLIP, and the Y variable is a binary indicator on the caption-concept correspondence according to an LLM. Compute the Mutual Information(MI) between X and Y as:

$$MI = \sum_{y \in Y} \sum_{x \in X} P_{x,y}(x,y) \log \frac{P_{x,y}(x,y)}{P_x(x) P_y(y)} \tag{2}$$

- **Step 5:** Based on the desired size of the concept set, select the top-K concepts with the highest mutual information.We use twice the number of categories for each dataset as the size of the concept set.
- **Step 6:** For the target dataset, we obtain the category-concept mapping matrix by prompting the LLM with the following prompt:
*Please just answer "yes" or "no". Does the {**category name**} usually have the visual attribute "{**concept**}"?*

Q: What are useful visual features for distinguishing a lemur in a photo? A: There are several useful visual features to tell there is a lemur in a photo: - four-limbed primate - black, grey, white, brown, or red-brown body color - wet and hairless nose with curved nostrils - long tail - large eyes - furry bodies - clawed hands and feet Q: What are useful visual features for distinguishing a television in a photo? A: There are several useful visual features to tell there is a California Gull in a photo: - Gray body color - White head and neck - Yellow bill color - Yellow legs - Red spot near bill - Long neck - Dark eyes - Gray wing color Q: What are useful features for distinguishing a {`category name`} in a photo? A: There are several useful visual features to tell there is a {`category name`} in a photo: -

Table 3. In-Context Learning Prompt used for generating concepts.

D. Motivation of disentanglement

Visual Prior Pre-trained ViT models typically use the [CLS] token to aggregate global image features. After pre-training, the [CLS] token embedding effectively focuses on the salient regions of the image. This has been extensively discussed in previous works such as [3, 5, 10], and [12] utilizes the relationship between [CLS] embeddings and patch embeddings to achieve object localization. Inspired by this,

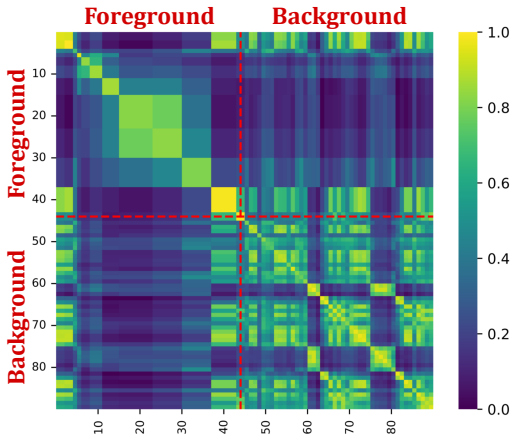


Figure 1. Pairwise dot product similarity matrix of patch embeddings extracted from pretrained ViT.

in our DOT-CBM framework, since the [CLS] embedding already possesses self-organizing capabilities for patch embeddings trained on large datasets, we explicitly use this organization as the prior distribution for the patch embedding set in the OT process. This approach not only retains the inductive biases learned by the pre-trained model from large datasets but also effectively integrates the global information of each image into the OT process through the prior distribution.

Disentanglement on Local representations Building upon the **Visual Prior**, we further investigate the internal characteristics of the patch embedding set. Specifically, we compute the dot product similarity between each patch embedding and all other patch embeddings, and we visualize these similarities as a heatmap, as shown in Figure 1. By distinguishing patches within the foreground from those in the background using the foreground mask generated by SAM (Segment Anything Model), we can clearly observe the distinct differences between *foreground* and *background* patches. Additionally, it is evident that both the intra-set similarity among foreground patches and the intra-set similarity among background patches are relatively high. From a linear algebra perspective, if we consider the [CLS] embedding vector as an aggregation of the patch embedding vectors, then:

$$e_{CLS} = \sum_{i=1}^{n} a_i e_{patch_i} \tag{3}$$

where n denotes the number of patches, and a_i represents the coefficients in the linear representation of the vectors. This approach reveals, as a representation of global information, the set of patch embeddings lacks sufficient disentanglement internally. Therefore, we propose an orthogonality disentanglement loss specifically designed for the internal structure of patch embeddings.

E. Comparison Experiment

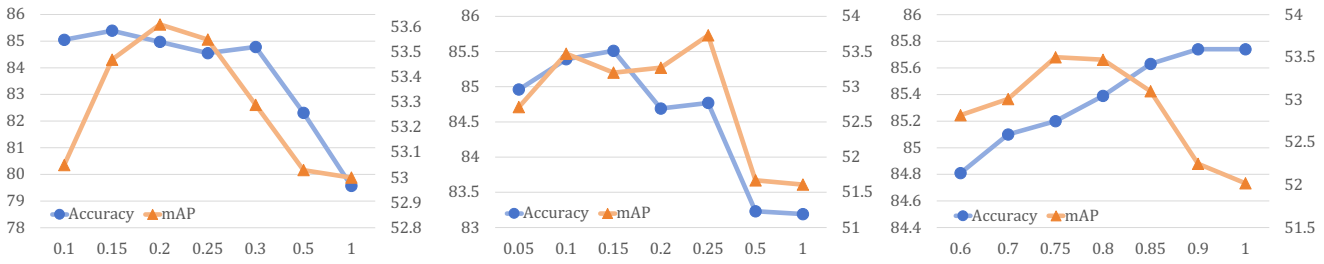
In DOT-CBM, both modalities incorporate predefined components: a pre-trained Backbone and a predefined concept set. Therefore, we conduct separate comparison experiments on different methods for these two components.

	ImageNet	CUB	Part-ImageNet		CUB	
			mAP	mIOU	mAP	mIOU
	Only Change Concept Set					
Human	/	80.94	/	/	48.21	0.59
LaBo	83.16	84.73	40.05	0.30	40.32	0.37
CDL	83.84	85.39	50.12	0.52	53.47	0.66
	Only Change Visual BackBone					
CLIP	82.77	84.67	49.14	0.41	53.08	0.54
MAE	83.19	84.34	47.83	0.50	52.26	0.60
DINOv2	83.84	85.39	50.12	0.52	53.47	0.66

Table 4. Comparison experiments on Visual Backbone and concept set.

When the concept set changes, LaBo's concept set is also generated using an LLM. However, these concepts rarely correspond to local visual features. Comparing the concept generation processes of CDL and LaBo, we identify two main reasons: (1) Overly Simple Prompting; (2) Lack of Visual Guidance. Despite LaBo's high overall performance, its class-specific concept set, which corresponds to the Section 3.4 in the main text, has a 100% co-occurrence rate within each class. This leads to a lack of discrimination among similar image information, resulting in low mAP and mIOU scores in the concept inversion process. In contrast, the human-defined concept set, although having lower overall performance, shows better correspondence in inversion. The performance of the human-defined concept set may be limited by its smaller size and potential errors in manual labeling. This phenomenon supports our hypothesis that a shared part concept space can enhance the correspondence of CBM concept inversion. Inspired by this, we use human-defined concepts as prompts for ICL when generating concepts with LLMs, treating the LLM-generated concepts as an extension of the human-defined concepts.

When the backbone changes, CLIP outperforms MAE in both classification performance and mAP. CLIP's large-scale language-image alignment pre-training results in better feature generalization and separability. In the inversion process, particularly for IOU ≥ 0.5, CLIP's image-text alignment properties lead to higher prediction precision. On the other hand, MAE, which is pre-trained using a Mask and Generate approach, has better fine-grained features compared to CLIP's single-label supervision. This results in better precision in the inversion process.



(a) λ_1 variation (b) λ_2 variation (c) λ_3 variation

Figure 2. Three hyperparameters variation

When the attribution changes. In our main experiments, the visual set prior of DOT-CBM is derived from the [CLS] token's attention map. Additionally, our framework supports advanced attribution methods (e.g., Rollout [1]) to generate this prior. Experimental results demonstrate that leveraging such methods yields superior part detection performance. Among the baseline methods, ViT-based approaches (e.g., LaBo [2] and SparseCBM [3]) were also evaluated using Rollout for part detection. Notably, our proposed method consistently exhibited the best performance across all comparisons.

	Part-ImageNet	CUB	RIVAL
LaBo (Rollout)	40.21	40.58	41.33
SparseCBM (Rollout)	40.93	42.35	43.06
DOT-CBM (Attention)	50.12	53.47	50.93
DOT-CBM (Rollout)	**52.94**	**56.18**	**52.83**

Table 5. Part Detection comparison with Rollout attribution.

F. Computational Complexity Analysis

We also compare the computational complexity of CBMs based on pre-trained models. Our method has slightly higher FLOPs due to the OT optimization process and prior computation, but this only accounts for 2.2% of the total FLOPs. Additionally, the number of learnable parameters in our model remains within an acceptable range.

Method	FLOPs ($\times 10^{12}$)	Estimated Memory (GB)
LaBo	20.84(20.45)	16.45
DOT-CBM	26.61(26.04)	16.84

Table 6. Computational Complexity Comparison for Pretrained model based CBM with Batch Size of 256. Gray numbers indicate the FLOPs of pre-trained ViT and text encoder.

G. Places365 Experiments

Experiments on complex dataset Places365 validate that DOT-CBM still perform well and shows correct concept inversion. Adopting task-specific (eg. SSL or SFT) ViTs can further enhance both prior accuracy and task performance.

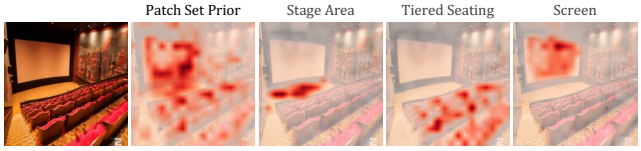


Figure 3. Prior and concept inversion for sample in Places365.

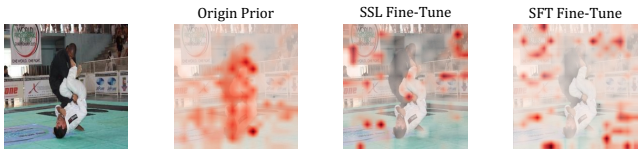


Figure 4. Prior in more complex cases (object + background).

S-CBM	LF-CBM	DOT-CBM	DOT-CBM(SSL)	DOT-CBM(SFT)
41.34	43.68	**44.07**	46.35	47.21

Table 7. Experiments on Places365-standard datasets.

H. Hyperparameter

The training process of DOT-CBM involves three tunable hyperparameters. The impact of varying these parameters on the model's classification accuracy and inversion correspondence metric (mAP) on the CUB dataset is shown in Figure.2. Among them:

λ_1: The mAP metric is highest at 0.2, but the classification performance is generally negatively correlated with λ_1. Therefore, we choose a balanced value of 0.15.

λ_2: A high value of λ_2 negatively affects both the model's accuracy and inversion performance. Hence, we set it to 0.1.

λ_3: The classification accuracy is positively correlated with λ_3, but the inversion performance peaks around 0.75. Therefore, we set it to 0.8 for a balanced trade-off.

For other implementation details, in addition to those mentioned in the main text, we design the Adapter as a three-layer MLP with a hidden layer size of 1096.

I. Visualization

Figure 5. Visualization.

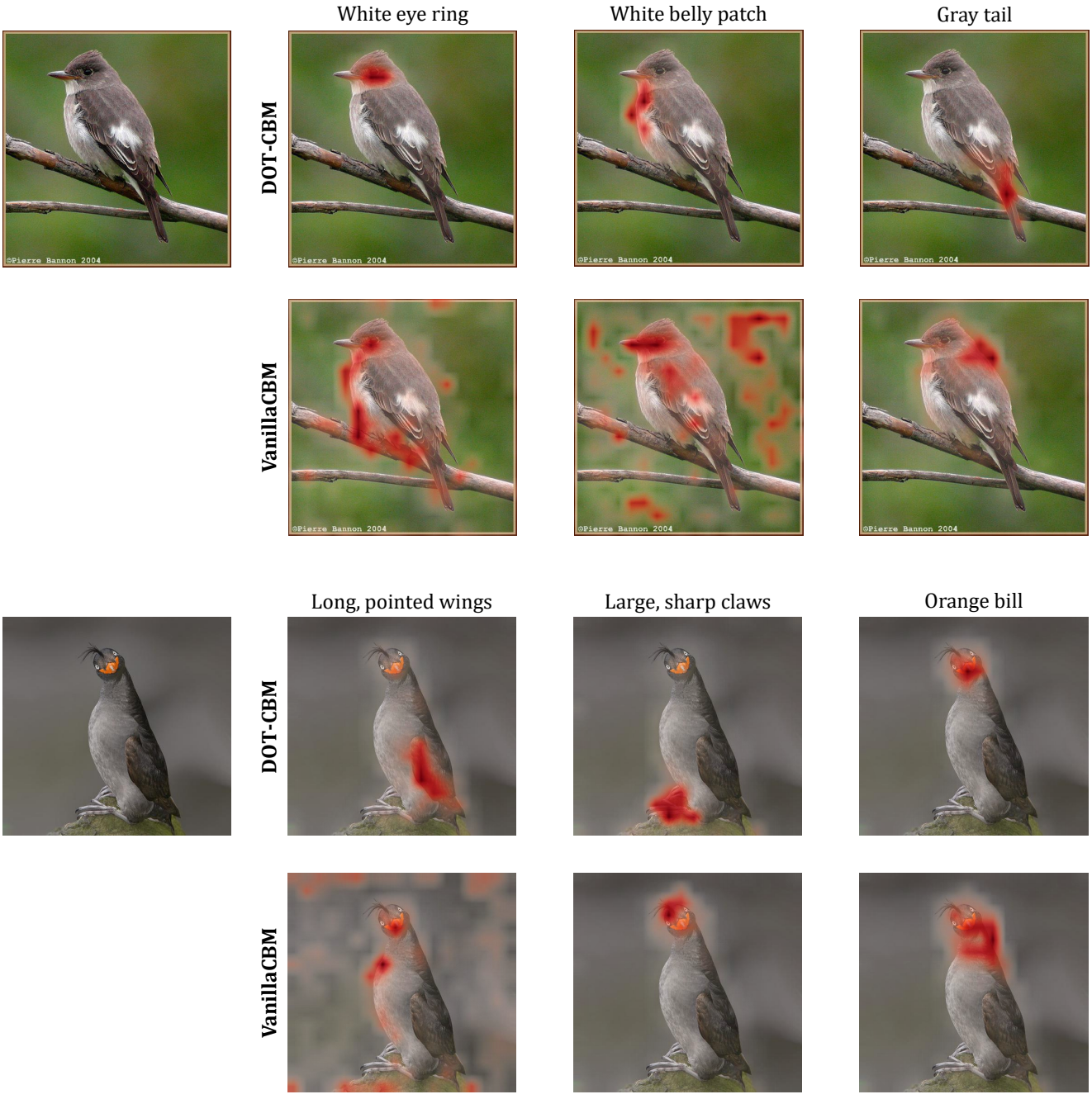


Figure 6. Visualization.

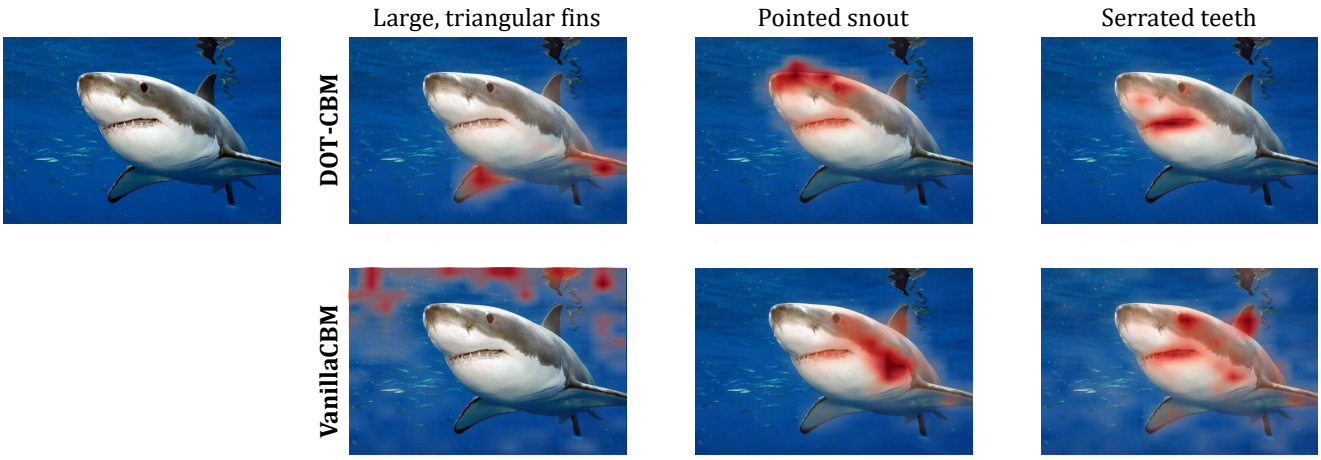


Figure 7. Visualization.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2

[4] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 1

[5] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 2

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html. 1

[7] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022. 1

[8] Tang Li, 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. 2

[9] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. 2

[10] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[11] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 2

[12] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021. 2

[13] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2

[14] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. 2

[15] Yuan Zang, Tian Yun, Hao Tan, Trung Bui, and Chen Sun. Pre-trained vision-language models learn discoverable visual concepts. *arXiv preprint arXiv:2404.12652*, 2024. 1, 2