# Unsupervised Part Discovery via Descriptor-Based Masked Image Restoration with Optimized Constraints (Supplementary Material)

Jiahao Xia[1], Yike Wu[1], Wenjian Huang[2], Jianguo Zhang[2], Jian Zhang[*,1]
[1]Faculty of Engineering and IT, University of Technology Sydney
[2] Dept. of Comp. Sci. and Eng., Southern University of Science and Technology
Jiahao.Xia-1@uts.edu.au, Yike.Wu@student.uts.edu.au, {huangwj, zhangjg}@sustech.edu.cn, Jian.Zhang@uts.edu.au

## Appendix A.1 Dataset Details

**PartImageNet OOD dataset**. We use the OOD variant of PartImageNet [6] to validate MPAE, following the setting used in [2, 14]. This variant comprises 110 classes distributed across 11 super-classes, including 14,865 samples for training and 1,658 samples for testing. Each sample is annotated with pixel-level part masks.
**PartImageNet Segmentation dataset**. This dataset is another variant of PartImageNet [6]. Compared to the OOD variant, it is more challenging because the number of categories increases from 109 to 158, comprising 20,457 images for training and 2,405 images for testing.
**CelebA dataset**. CelebA dataset [10] contains 200,000 unaligned face images with 5 labeled keypoints, representing the eye centers, the tip of the nose, and the corners of the mouth, for 10,000 different identities. Following the setting in [8], we retain the images where face covers more than 30% of the area, resulting in 45,609 images for training, 5,397 images for validation and 283 images for testing. This ensures the face to be the salient object in each image for subsequent part discovery.
**CUB dataset**. CUB dataset [15] consists of 200 different bird species with 5,994 images for training and 5,794 images for testing. Each image is annotated with 15 keypoints and their visibility, representing 15 different bird parts.

## Appendix A.2 Implementation Details

As in [7, 14], we set the input size to $448 \times 448$ for CUB dataset and $224 \times 224$ for other datasets to ensure fair comparisons. For the datasets with multiple categories (PartImageNet OOD and PartImageNet Segmentation), the mini-group size is set to 64. For the datasets with single category, the mini-group size is set to 8. All models employ a frozen ViT-B/14 pre-trained using DINO v2 [11] and register tokens [4] to extract dense feature maps. Other parts of MPAE are trainable and are optimized using Adam optimizer [5]. The learning rate, batch size, and feature dimension $C$ are set to $5 \times 10^{-3}$, 64 and 256 respectively. The number of both MPAE encoder layers and decoder layers is set to 2. In all experiments, $\lambda_{\mathrm{p}}$ is set to 1.0, $\lambda_{\mathrm{d}}$ to 0.5, and $\lambda_{\mathrm{s}}$ to 0.25. Referring to [16], we set $s$ and $m$ to 20 and 0.5, respectively, to ensure that each part descriptor is well aligned with the pixel-level features that have very high similarity.

## Appendix B.1 Influence of Structural Difference Penalty Component of $\mathcal{L}_{\mathrm{r}}$

| Structural Difference Penalty Component | With | Without |
|---|---|---|
| NMI (%) ↑ | **55.10** | 19.65 |
| ARI (%) ↑ | **73.52** | 49.72 |

Table 1. Performance comparisons of MPAE with/without the structural difference penalty component of $\mathcal{L}_{\mathrm{r}}$ on the PartImageNet Segmentation dataset in the setting of $K = 50$.

*Corresponding Author

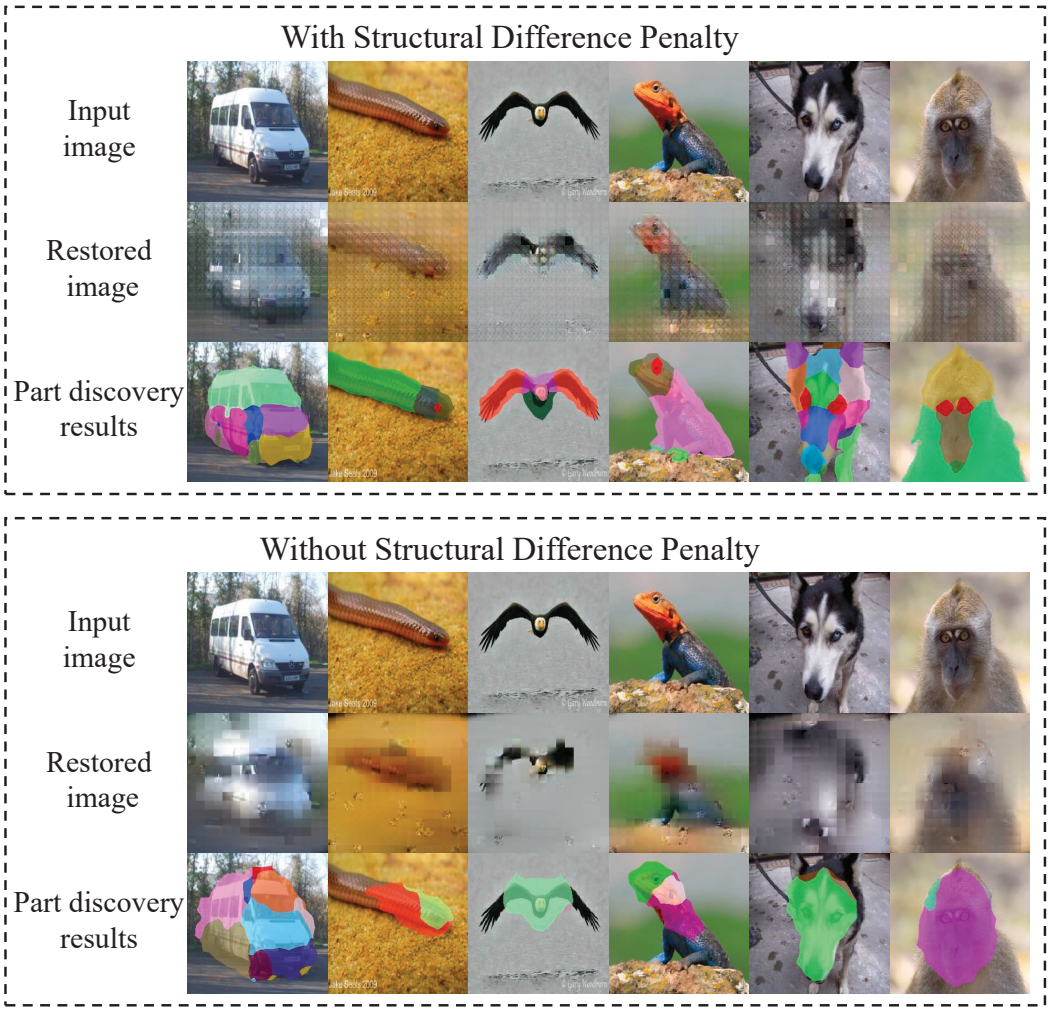


Figure 1. Some qualitative results of image restoration and part discovery predicted by the MPAE trained with/without structural difference penalty component in $\mathcal{L}_{\mathrm{r}}$.

In addition to minimizing the differences between masked image patches and their corresponding restored patches, we employ a frozen pretrained VGG-19 [13] to penalize the structural difference between the masked and restored images. The quantitative and qualitative impact of this structural difference penalty component is demonstrated in Table 1 and Fig. 1 respectively in Appendix B.1. This penalty plays an essential role in the training of MPAE, increasing the NMI and ARI metrics from 19.65 and 49.72 to 55.10 and 73.52 respectively. From Fig. 1 in this supplementary file, we observe that the part discovery results are highly consistent with the images restored using their learned part descriptors: restored patches with similar views tend to be identified as the same part. This further supports the conclusion that MPAE implicitly clusters the filled part descriptors and unmasked patch features within the same part regions by utilizing them to generate image patches with similar appearances. Consequently, the low-level appearance features of the unmasked patches further align the high-level semantics of the part descriptors with the corresponding part shapes. Without the structural penalty, significant structural deviations can be observed between the input images and the restored images. This further results in a misalignment between the part descriptors and the shapes of their corresponding parts, as well as similarity maps that do not closely follow the part boundaries. Consequently, the MPAE fails to discover meaningful parts with consistent semantics, resulting in performance degradation in all metrics.

## Appendix B.2 Influence of Encoder Layer Number

| MPAE encoder layer number | 1 | 2 | 4 | 6 |
|---|---|---|---|---|
| NMI (%) ↑ | 50.71 | 55.10 | 55.12 | **55.43** |
| ARI (%) ↑ | 70.86 | 73.52 | **73.74** | 73.59 |

Table 2. Performance comparison of MPAE with different number of encoder layers on the PartImageNet Segmentation dataset in the setting of $K = 50$. The number of MPAE decoder layers is fixed at 2.

We conduct ablation studies on PartImageNet Segmentation dataset ($K = 50$) to investigate the influence of the number of MPAE encoder layers. The results are reported in Table 2 in Appendix B.2. When the number of MPAE encoder layers increases from 1 to 2, the performance of MPAE improves from 50.71 to 55.10 in NMI. This is mainly because the expressive ability of the MPAE encoder with only a single layer is insufficient to effectively encode the appearance of unmasked patches in the latent space. Consequently, the high-level semantics from part descriptors and the low-level appearances are not well aligned, leading to performance degradation. Nevertheless, when the number of encoder layers exceeds two, the MPAE encoder can coherently encode the appearance of unmasked patches in the latent space. Therefore, the improvement gained from increasing the number of encoder layers is relatively slight.

## Appendix B.3 Influence of Decoder Layer Number

| MPAE Decoder layer number | 1 | 2 | 4 | 6 |
|---|---|---|---|---|
| NMI (%) ↑ | 54.27 | **55.10** | 53.89 | 52.69 |
| ARI (%) ↑ | 73.32 | **73.52** | 72.43 | 71.81 |

Table 3. Performance comparison of MPAE with different number of decoder layers on the PartImageNet Segmentation dataset in the setting of $K = 50$. The number of MPAE encoder layers is fixed at 2.

We also conduct ablation studies on PartImageNet Segmentation dataset ($K = 50$) to investigate the influence of the number of MPAE decoder layers. The results are reported in Table 3 in Appendix B.3. Similarly, when the number of MPAE decoder layers increases from 1 to 2, we observe improvements in all metrics. The primary reason is that increasing the number of decoder layers from one to two improves image restoration results. Consequently, high-level semantics from part descriptors become better aligned with the shapes of their corresponding parts. However, when the number of decoder layers exceeds 2, we observe slight performance degradation. This is because more decoder layers encourages the model to rely more on the unmasked patch features for image restoration rather than part descriptors. As a result, the features within the same part region on the filled feature map $\boldsymbol{R}$ are not well aligned with the part shapes.

## Appendix B.4 Influence of $\lambda_{\mathrm{d}}$, $\lambda_{\mathrm{p}}$ and $\lambda_{\mathrm{s}}$

| $\lambda_{\mathrm{d}}$ | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| NMI↑(%) | 35.88 | 53.21 | 55.10 | 55.45 | 55.36 |
| ARI↑(%) | 31.39 | 72.89 | 73.52 | 74.85 | 73.93 |

| $\lambda_{\mathrm{p}}$ | 0.50 | 0.75 | 1.00 | 1.25 | 1.50 |
|---|---|---|---|---|---|
| NMI↑(%) | 53.50 | 55.55 | 55.10 | 53.46 | 52.68 |
| ARI↑(%) | 74.62 | 73.61 | 73.52 | 73.08 | 72.45 |

| $\lambda_{\mathrm{s}}$ | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 |
|---|---|---|---|---|---|
| NMI↑(%) | 55.45 | 55.23 | 55.10 | 53.00 | 51.28 |
| ARI↑(%) | 74.59 | 75.48 | 73.52 | 69.34 | 68.75 |

Table 4. Influence of $\lambda_{\mathrm{d}}$, $\lambda_{\mathrm{p}}$ and $\lambda_{\mathrm{s}}$ on PartImageNet Segmentation ($K = 50$)

We carry out ablation studies on PartImageNet Segmentation dataset ($K = 50$) to further investigate the influence of hyper-parameters $\lambda_{\mathrm{d}}$, $\lambda_{\mathrm{p}}$ and $\lambda_{\mathrm{s}}$, as shown in Table 4 in Appendix B.4. Across a wide range of $\lambda_{\mathrm{d}}$, $\lambda_{\mathrm{p}}$ and $\lambda_{\mathrm{s}}$, MPAE consistently maintains comparable performance, indicating that it is not sensitive to the selection of hyperparameters. Moreover, we keep the hyperparameters fixed across all datasets in our paper, and MPAE still achieves competitive performance compared to other state-of-the-art methods, illustrating its robustness.

## Appendix B.5 Influence of Different Self-supervised Pretrained Backbones

| Backbone | K=8 (%) | | K=25 (%) | | K=50 (%) | |
|---|---|---|---|---|---|---|
| | NMI | ARI | NMI | ARI | NMI | ARI |
| DINO v1 [3] | 26.86 | 63.70 | 32.89 | 69.75 | 39.12 | 70.08 |
| DINO v2 [11] with [4] | **32.90** | **66.17** | **39.28** | **68.87** | **53.65** | **74.22** |

Table 5. Performance comparison of MPAE with different self-supervised pretrained backbones on the PartImageNet OOD dataset in the setting of $K = 50$. The number of MPAE encoder and decoder layers is fixed at 2.

We implement MPAE with DINO v1 (ViT-S/16) on PartImage OOD, and the results are reported in Table 5 in Appendix B.5. Compared to direct clustering [1] and Xia et al. [16], which use similar backbone (DINO v1, ViT-S/8), our MPAE with DINO v1 still outperforms them by a significant margin, demonstrating the effectiveness of our method. However, the features produced by DINO V1 are not as fine-grained as those produced by DINO v2. Consequently, MPAE with DINO V1 fails to outperform MPAE with DINO v2.

## Appendix B.6 Comparison with Supervised Pretrained Backbones

| Backbone | DINO v2 [11] with [4] | SAM [9] | CLIP [12] |
|---|---|---|---|
| NMI (%) ↑ | **55.10** | 17.16 | 33.16 |
| ARI (%) ↑ | **73.32** | 55.26 | 72.90 |

Table 6. Performance comparison of MPAE with different supervised pretrained backbones on the PartImageNet Segmentation dataset in the setting of $K = 50$. The number of MPAE encoder and decoder layers is fixed at 2.

We also implement MPAE on the PartImageNet Segmentation dataset using backbones pretrained in a fully supervised manner, including the encoder of Segment Anything (SAM) and CLIP. The results are shown in Table 6 of Appendix B.6. The training of SAM focuses on object boundaries rather than semantics, while CLIP mainly aligns instance-level descriptions with global ViT features in the latent space. Neither of them can produce finer-grained part-level features compared to DINO v2. Therefore, MPAE with DINO v2 achieves better performance, even though it is pretrained without any manual labels.

## Appendix B.7 Influence of Mini-group Size

| Dataset | CelebA | | | | PartImageNet-S | | | |
|---|---|---|---|---|---|---|---|---|
| Mini-group size | 4 | 8 | 16 | 32 | 16 | 32 | 64 | 128 |
| NMI (%) ↑ | 59.50 | **59.64** | 53.89 | 25.52 | 43.94 | 47.28 | **55.10** | 53.63 |
| ARI (%) ↑ | **41.78** | 41.72 | 35.06 | 10.34 | 59.71 | 66.75 | **73.52** | 72.63 |

Table 7. Performance comparison of MPAE with different mini-group size on CelebA ($K = 8$) and PartImageNet Segmentation ($K = 50$).

We report the performance of MPAE with different mini-group sizes on a single-category dataset (CelebA) and a multi-category dataset (PartImageNet Segmentation) in Table 7 in Appendix B.7. With a very large mini-group size, MPAE tends to identify rarely appearing regions as independent parts. When the mini-group size is set to 32 on the CelebA dataset, the model identifies sunglasses and hands as independent parts instead of decomposing the face region into the target number of parts. Since we calculate the metrics using facial landmarks on the CelebA dataset, this results in a significant degradation in NMI and ARI on CelebA. Therefore, we set the mini-group size to 8 for datasets containing only a single category. However, different categories in PartImageNet Segmentation consist of various parts. A mini-group size that is too small forces MPAE to focus only on highly similar regions shared across different categories. Consequently, we observe a performance degradation when the mini-group size is less than 32 on datasets with multiple categories. As a result, we set the mini-group size to 64 for datasets with multiple categories.

## Appendix B.8 Average Number of Discovered Parts per Image with/without $\mathcal{L}_{\mathrm{s}}$

| Model | with $\mathcal{L}_{\mathrm{s}}$ | without $\mathcal{L}_{\mathrm{s}}$ |
|---|---|---|
| Average number of discovered parts per image | 9.27 | 3.94 |

Table 8. Average number of discovered parts per image on PartImageNet Segmentation ($K = 50$) with/without $\mathcal{L}_{\mathrm{s}}$

We compute the average number of discovered foreground parts per image with/without the constraint of $\mathcal{L}_{\mathrm{s}}$ to further investigate its influence, as shown in Table 8 in Appendix B.8. Without $\mathcal{L}_{\mathrm{s}}$, the average number of discovered foreground parts per image is only 3.94, indicating that each object is parsed into one or several coarse parts. This is because the MPAE without $\mathcal{L}_{\mathrm{s}}$ ssigns the $K$ parts primarily based on instance-level similarity rather than exploring the shared parts. $\mathcal{L}_{\mathrm{s}}$ encourages each part descriptor to respond only to regions with high semantic similarity on the feature map$\boldsymbol{F}$. As a result, the MPAE with $\mathcal{L}_{\mathrm{s}}$ can better discover the shared parts across multiple categorie, and the average number of the discovered foreground parts per image increases to 9.27.

## Appendix C.1 Discovered Parts across Multiple Categories

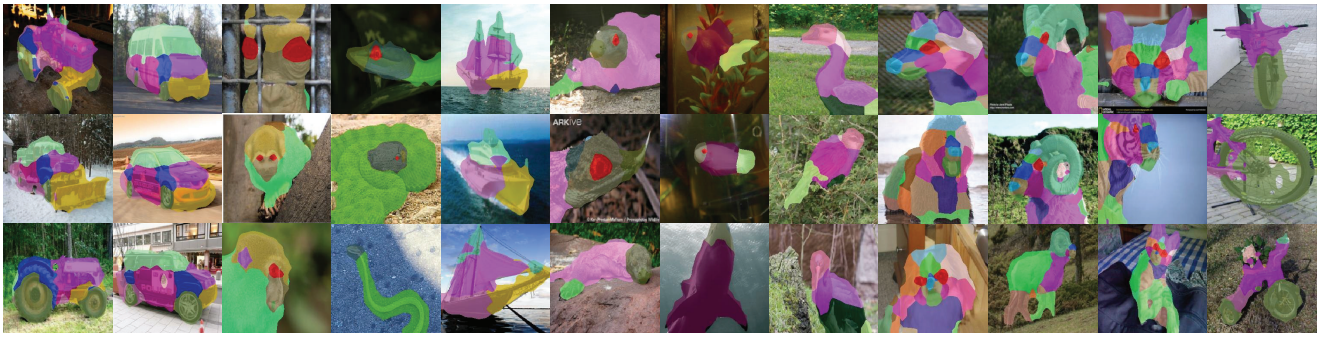

Figure 2. Parts unsupervisedly discovered by MPAE across multiple categories on PartImageNet Segmentation ($K = 50$). The same color indicates that these discovered parts share similar semantics, even if they belong to different categories.

## Appendix C.2 Visualized Attention Maps in the Trainable ViT

In Fig. 3, we present some pixel-level masks of the discovered parts predicted by MPAE, along with their corresponding attention maps from the trainable ViT used for descriptor extraction. The image restoration process performs implicit clustering, encouraging the features from the same part region to be similar. As a result, the region of each part on $\boldsymbol{S}$ is filled with the same part descriptor. By using these part descriptors to restore the masked patches, the ViT successfully learns to extract features from the corresponding part regions as part descriptors through the attention mechanism, as shown in Fig. 3 in Appendix C.2. This explains why the learned results can be robustly generalized to test images.

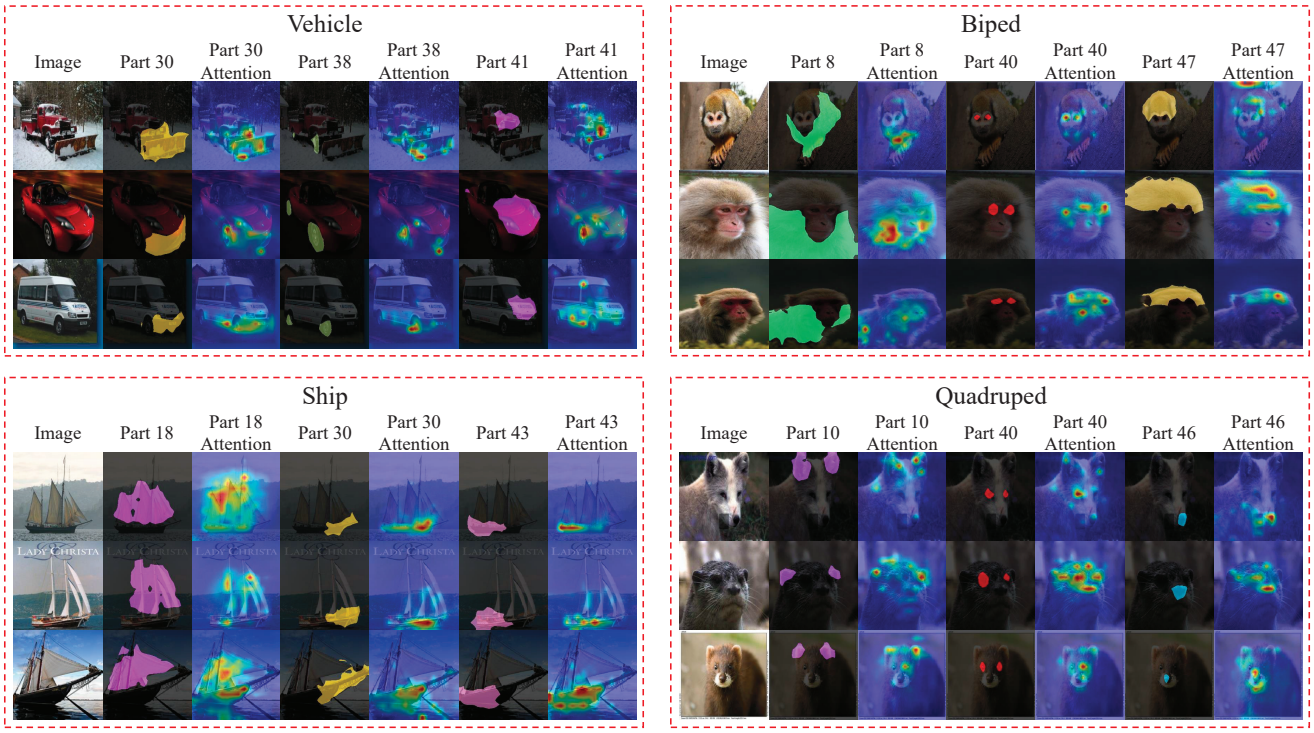


Figure 3. Pixel-level masks of discovered parts predicted by MPAE and their corresponding attention maps on PartImageNet Segmentation dataset ($K = 50$).

## Appendix C.3 More Visualized Part Discovery Results

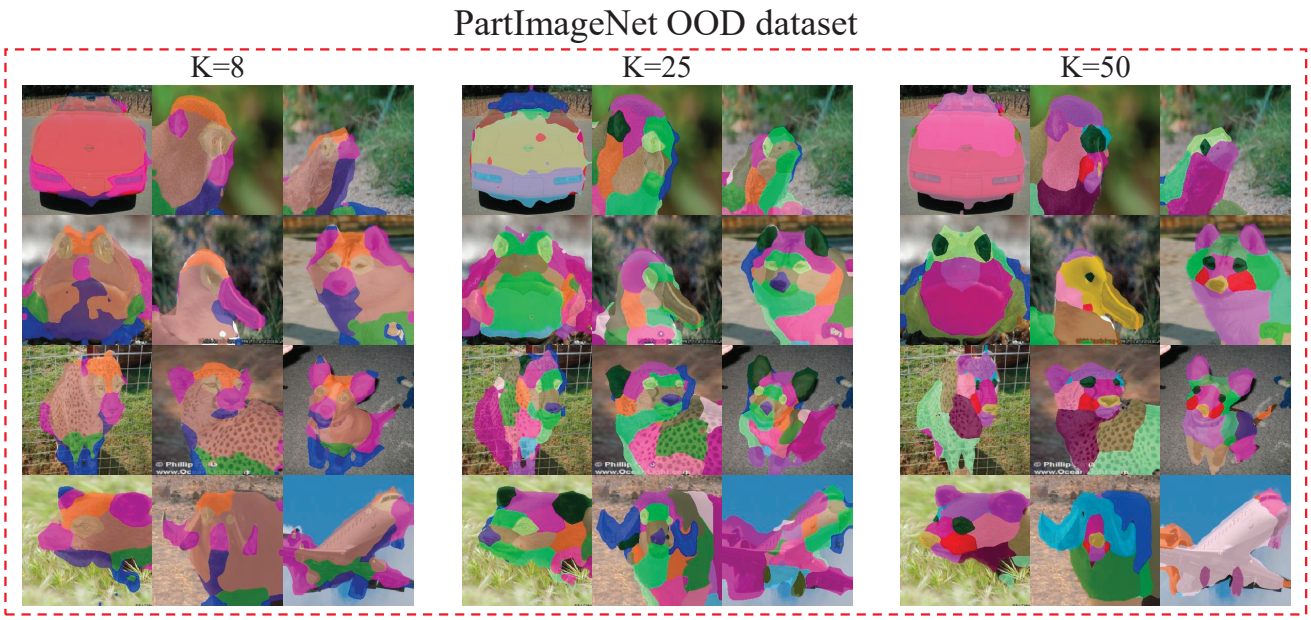


Figure 4. Examples of unsupervised part discovery results on PartImageNet OOD dataset predicted by MPAE in the setting of $K = 8, 25, 50$.

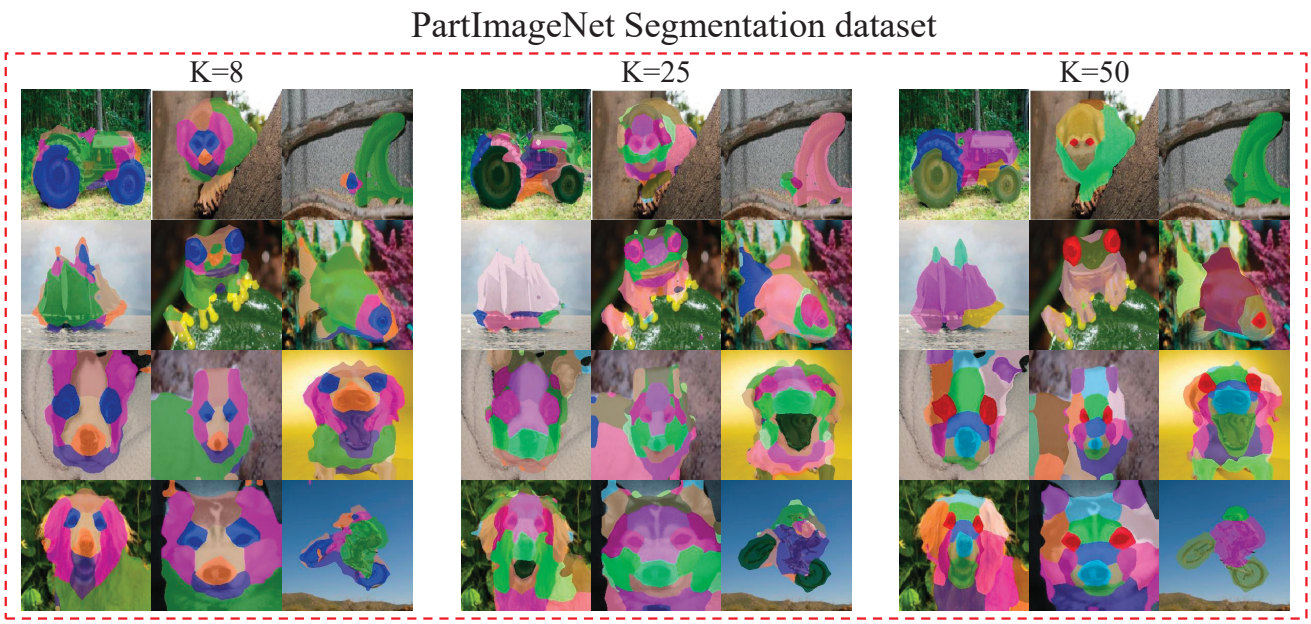


Figure 5. Examples of unsupervised part discovery results on PartImageNet Segmentation dataset predicted by MPAE in the setting of $K = 8, 25, 50$.

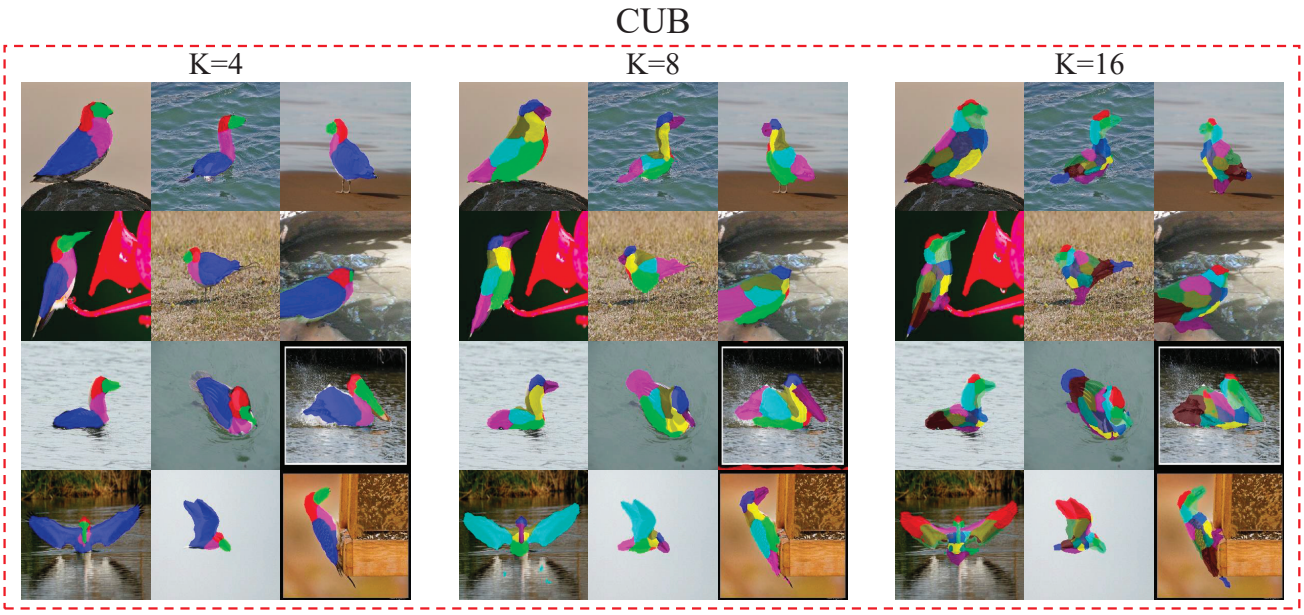


Figure 6. Examples of unsupervised part discovery results on CUB dataset predicted by MPAE in the setting of $K = 4, 8, 16$.

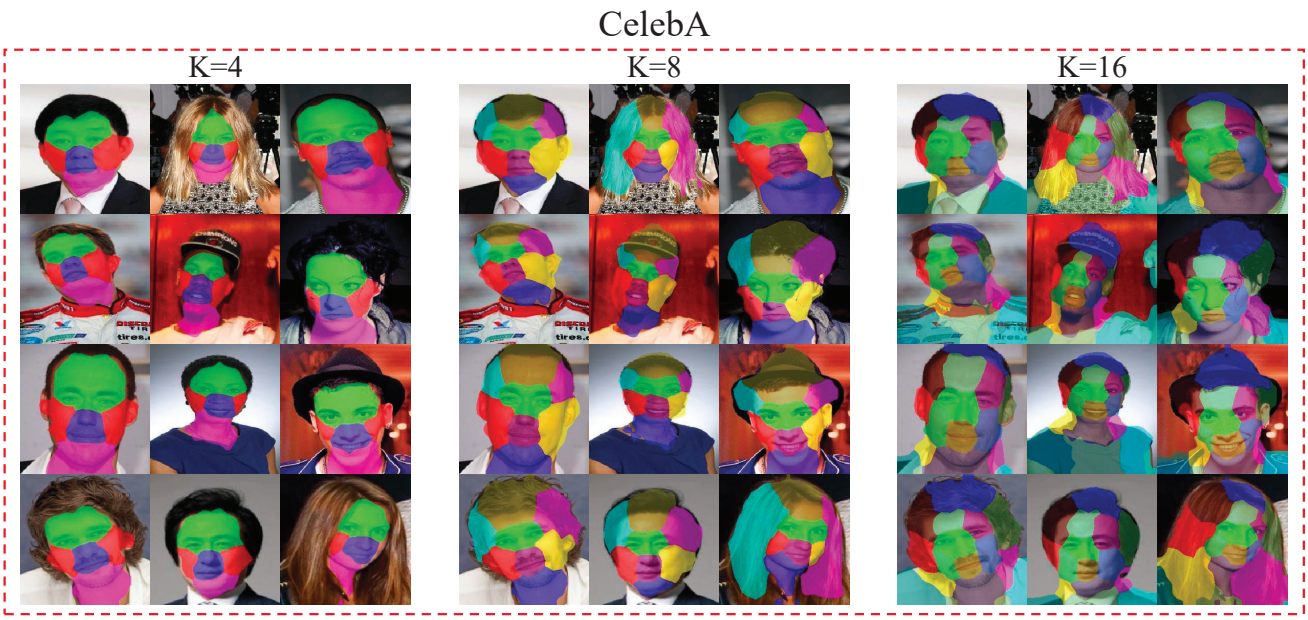


Figure 7. Examples of unsupervised part discovery results on CelebA dataset predicted by MPAE in the setting of $K = 4, 8, 16$.

## References


[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. In *ECCV Workshops*, 2022. 4

[2] Ananthu Aniraj, Cassio F Dantas, Dino Ienco, and Diego Marcos. Pdiscoformer: Relaxing part discovery constraints with vision transformers. In *ECCV*, 2024. 1

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9630–9640, 2021. 4

[4] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR.*, 2024. 1, 4

[5] Kingma Diederik and Ba Jimmy. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[6] 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 *ECCV*, pages 128–145, 2022. 1

[7] Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *CVPR*, pages 8659–8669, 2020. 1

[8] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *CVPR*, pages 869–878, 2019. 1

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, pages 3992–4003, 2023. 4

[10] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 1

[11] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 1, 4

[12] 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, 2021. 4

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[14] R. van der Klis, S. Alaniz, M. Mancini, C. F. Dantas, D. Ienco, Z. Akata, and D. Marcos. Pdisconet: Semantically consistent part discovery for fine-grained recognition. In *ICCV*, pages 1866–1876, 2023. 1

[15] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1

[16] Jiahao Xia, Wenjian Huang, Min Xu, Jianguo Zhang, Haimin Zhang, Ziyu Sheng, and Dong Xu. Unsupervised part discovery via dual representation alignment. *IEEE TPAMI*, pages 1–18, 2024. 1, 4