LINPART: UNSUPERVISED PART DISCOVERY WITH A SINGLE LINEAR HEAD

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised part discovery segments objects into parts and assigns consistent identities across images without part annotations. Recent methods build on self-supervised features by adding trainable modules or partially fine-tuning the backbone. Two analyses of frozen DINOv2 features motivate a simpler design. First, nearest-centroid clustering already recovers part structure using a linear assignment rule. Second, cross-image feature neighbours agree on part identity more often than randomly reconnected pairs. Based on these observations, LinPart trains only a shared linear head on the frozen features. Its objective combines feature reconstruction and regularization to guide foreground selection, division into parts, and consistent part assignments across images. With at most 39,270 trainable parameters, LinPart outperforms prior methods. We also derive a merge score by relating the expected correspondence loss to graph modularity and incorporating changes in reconstruction and warp consistency. Applied to a trained head's predictions, the score produces partitions with fewer parts without annotations or retraining. In multiple evaluated settings, these partitions achieve higher mean agreement with annotated parts than direct training at the target part count.

1 INTRODUCTION

Unsupervised part discovery identifies regions such as a bird's head, wings, and tail and assigns them consistent identities across images. Such parts support fine-grained recognition and correspondence between instances (Van Der Klis et al., 2023; Amir et al., 2021). Prior work shows that this part structure is already present in frozen self-supervised features. Principal components of DINOv2 patch features reveal corresponding regions across images (Oquab et al., 2024), and clustering DINO features recovers parts without labels (Amir et al., 2021). Recent methods build on these features by adapting the backbone or adding trainable components. PDiscoFormer (Aniraj et al., 2024) partially fine-tunes the backbone with image class labels, and MPAE (Xia et al., 2025) trains a prototype transformer and a reconstruction decoder on frozen features. These additions raise the question of what a shared linear head alone can achieve on the frozen features. Keeping the backbone frozen and the head linear allows us to examine the role of the training objective.

Two analyses in Table 1 examine whether the frozen features support this design. The first asks how much part information an affine rule can read from the features, using k-means without part annotations and a linear probe with them. Fitting the centroids only on tokens selected by an unsupervised foreground estimate improves normalized mutual information (NMI) on all three benchmarks. The probe reaches 88.1 NMI on CUB against 66.1 for foreground-restricted k-means (Table 1a). A linear head on the same features can therefore reach higher agreement with annotated parts than clustering. The second asks whether feature similarity across images provides a signal for learning consistent part identities. Feature neighbours across images agree on part identity more often than randomly reconnected pairs (Table 1b). The control preserves each token's incoming and outgoing connection counts. These two results support a shared linear head for part prediction and cross-image feature matches for consistent identities.

Learning part assignments requires selecting the foreground, dividing it into parts, and maintaining consistent identities across images. Feature similarity provides a grouping criterion within the foreground but does not ensure agreement with semantic parts. A shared head applies the same assignment rule across images, but appearance differences can lead to different assignments for cor-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Part information in frozen DINOv2 features. (a) NMI of clustering and supervised linear probes. (b) Cross-image part-label agreement, averaged over source-part classes (%).

(a) Part prediction

Rule	CUB	NABirds	OOD
k-means	58.3	46.5	40.4
+ foreground	66.1	56.4	53.3
Linear probe (supervised)	88.1	88.3	74.0

(b) Cross-image correspondence

Pairs	CUB	NABirds	OOD
Feature neighbours	67.37	70.67	35.01
Reconnected control	9.23	13.01	17.18

responding regions. Assigning all foreground tokens to one part would give consistent labels but remove distinctions between parts. The objective must therefore encourage cross-image consistency while avoiding this collapse.

LinPart addresses these requirements with six loss terms applied to a shared linear head on frozen DINOv2 features. A foreground anchor guides foreground selection using an unsupervised objectness estimate. Reconstruction measures how well per-image slot means approximate the token features. Slot usage discourages concentration in a few parts. Spatial smoothness penalizes differences in assignment probabilities between neighbouring tokens, and warp consistency penalizes assignment changes under token-grid resampling. Correspondence encourages matched tokens across images to be assigned to the same part.

To reuse a trained head at smaller part counts, we derive a merge criterion from its objective. In expectation, minimizing the correspondence loss maximizes a modularity score that rewards matches within parts beyond what their sizes alone would predict. Reconstruction penalizes merging co-occurring parts with different mean features. For each candidate merge, the score combines changes in correspondence, reconstruction, and warp consistency, computed from frozen features and the head's predictions on training images. Successive merges map the head's predicted parts to coarser groups without retraining. The resulting partitions achieve higher mean agreement with annotated parts than direct training at the target part count in most evaluated settings.

- We introduce LinPart, whose design is motivated by analyses of linearly accessible part information and cross-image correspondence. It trains only a shared linear head on frozen DINOv2 features without labels. With at most 39,270 trainable parameters, LinPart achieves higher NMI and adjusted Rand index (ARI) than prior methods in multiple evaluated settings.
- We derive a merge criterion from correspondence, reconstruction, and warp consistency, connecting correspondence to graph modularity. Applied to a trained head, it gives coarser partitions without retraining, with higher mean NMI and ARI than direct training at the target part count in five of six evaluated settings.

2 RELATED WORK

Part discovery. Prior work discovers parts or landmarks without annotations through CNN feature factorization (Collins et al., 2018), spatial embeddings (Thewlis et al., 2017; Zhang et al., 2018), or assignment networks with spatial and background constraints (Hung et al., 2019). Reconstruction-based methods assemble parts (Gao et al., 2021), disentangle shape and appearance (Liu et al., 2021), or use contrastive reconstruction (Choudhury et al., 2021). GANSeg (He et al., 2022b) discovers parts through hierarchical image generation. Class-supervised methods learn parts within image classifiers (Huang & Li, 2020; Van Der Klis et al., 2023). PDiscoFormer (Aniraj et al., 2024) partially fine-tunes a self-supervised transformer using image class labels and regularizes part assignments with spatial and consistency losses. Label-free methods also align part descriptors with image content through reconstruction (Xia et al., 2024). MPAE (Xia et al., 2025) freezes the backbone and trains a prototype transformer and a masked-reconstruction decoder. In object-centric learning, DINOSAUR (Seitzer et al., 2023) reconstructs frozen features using slot attention and a decoder. PDiscoFormer's (Aniraj et al., 2024) distance-based assignments can also be represented by affine logits in the token features. LinPart differs in what is trained, learning only a shared linear

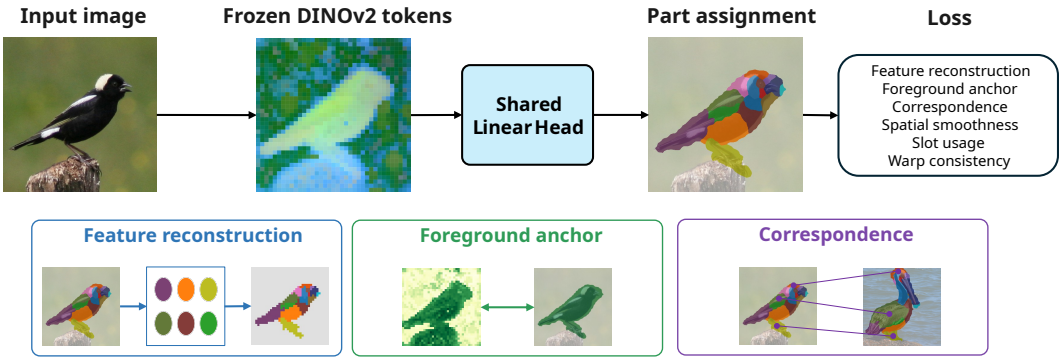

Figure 1: Overview of LinPart. Only the shared linear head is trained.

head on fully frozen features without image class or part annotations. Reconstruction uses per-image prototypes computed from the assignments, without a learned decoder.

Part information in pretrained features. Prior studies identify object and part structure through attention maps (Caron et al., 2021), principal components (Oquab et al., 2024), cross-image clustering (Amir et al., 2021), and supervised linear segmentation on frozen features (Zhu et al., 2023). PARTICUL learns part detectors from recurring patterns in pretrained CNN features (Xu-Darme et al., 2022). Leopart (Ziegler & Asano, 2022) and PARTICLE (Saha & Maji, 2023) further learn part-aware features through self-supervised dense clustering and part-level contrastive learning, respectively. Frozen features also support foreground discovery through patch similarities (Siméoni et al., 2021) and spectral partitioning (Wang et al., 2022). Feature similarities provide supervision at the image level in SCAN (Van Gansbeke et al., 2020) and at the patch level in STEGO (Hamilton et al., 2022) and Hidden Positives (Seong et al., 2023). FOUND (Siméoni et al., 2023) trains a single 1×1 convolution on frozen features using automatically generated binary masks to separate foreground from background. LinPart trains a shared linear head to discover K foreground parts with consistent identities across images. Its objective uses foreground guidance to select object tokens and cross-image feature matches to encourage consistent part assignments.

Partition coarsening. Modularity measures graph community structure relative to a chance baseline (Newman, 2006; Reichardt & Bornholdt, 2006), and differentiable modularity has been used to train graph clustering models (Tsitsulin et al., 2023). Leopart groups part clusters through community detection on a spatial co-occurrence graph (Ziegler & Asano, 2022). In object-centric video learning, SOLV merges similar slots through agglomerative clustering as part of training (Aydemir et al., 2023). LinPart connects its correspondence loss to modularity and scores candidate merges using changes in correspondence, reconstruction, and warp consistency. The resulting criterion groups a trained head's predicted part labels without updating its parameters.

3 METHOD

LinPart uses patch tokens from a frozen DINOv2-B/14 backbone with registers (Oquab et al., 2024; Darcet et al., 2024). A shared linear head assigns each token to one of K parts or to background. Its $K + 1$ output channels are called slots.

3.1 ARCHITECTURE

Figure 1 shows an overview of LinPart. The frozen backbone produces an $H_p \times W_p$ grid of patch features, where H_p and W_p are the numbers of patches along the image height and width. Flattening the grid gives $N = H_p W_p$ token features $\boldsymbol{z}_n \in \mathbb{R}^{768}$. Let K denote the predefined number of part slots. A shared linear head maps each token feature to logits over the K part slots and one background slot. The logits are $\boldsymbol{\ell}_n = \boldsymbol{W}\boldsymbol{z}_n + \boldsymbol{b}$, where $\boldsymbol{W} \in \mathbb{R}^{(K+1)\times 768}$ is the weight matrix and $\boldsymbol{b} \in \mathbb{R}^{K+1}$ is the bias vector. During training, Gumbel-softmax gives assignment probabilities

$\boldsymbol{q}_n = \mathrm{softmax}(\boldsymbol{\ell}_n + \boldsymbol{\epsilon}_n)$, where the components of $\boldsymbol{\epsilon}_n$ are independent standard Gumbel noise variables. At inference, the noise is omitted.

3.2 Loss

For each image, we collect the assignment probabilities in the matrix $\boldsymbol{A} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N] \in \mathbb{R}^{(K+1)\times N}$. Its entry $A_{s,n} = q_{n,s}$ is the probability of assigning token n to slot s. The foreground probability of token n is $m_n = 1 - q_{n,\mathrm{bg}}$, where $q_{n,\mathrm{bg}}$ is its background assignment probability. The objective combines six terms,

$$L = \lambda_{\mathrm{rec}} L_{\mathrm{rec}} + \lambda_{\mathrm{fg}} L_{\mathrm{fg}} + \lambda_{\mathrm{corr}} L_{\mathrm{corr}} + \lambda_{\mathrm{tv}} L_{\mathrm{tv}} + \lambda_{\mathrm{use}} L_{\mathrm{use}} + \lambda_{\mathrm{cons}} L_{\mathrm{cons}}, \tag{1}$$

where the λ coefficients weight the terms. Hyperparameters and implementation details are provided in Appendix B.

Feature reconstruction. Feature reconstruction evaluates a partition by how well its per-image slot prototypes reconstruct the frozen token features. For each image, the prototype of slot s is $\boldsymbol{v}_s = \sum_n \bar{A}_{s,n} \boldsymbol{z}_n$, where $\bar{A}_{s,n} = \frac{A_{s,n}}{\sum_{n'} A_{s,n'}}$ normalizes the assignments over tokens. Each token is reconstructed as $\hat{\boldsymbol{z}}_n = \sum_s A_{s,n} \boldsymbol{v}_s$, using all $K+1$ slots, including background. The loss normalizes the reconstruction error by the total feature variance within the batch,

$$L_{\mathrm{rec}} = \frac{\sum_{b,n} \|\boldsymbol{z}_{b,n} - \hat{\boldsymbol{z}}_{b,n}\|^2}{\sum_{b,n} \|\boldsymbol{z}_{b,n} - \bar{\boldsymbol{z}}\|^2}, \tag{2}$$

where b indexes images and $\bar{\boldsymbol{z}}$ is the mean feature vector over all tokens in the batch.

Foreground anchor. The foreground anchor uses an objectness target from the frozen features to guide the predicted foreground probability m_n. For each image, projecting the centered token features onto their first principal component gives an initial score. Its sign is chosen so that border tokens have a lower mean score than interior tokens. The scores are then scaled to $[0, 1]$ to obtain p_n and pooled across images,

$$u_n = (1 - \alpha) p_n + \frac{\alpha}{k_{\mathrm{fg}}} \sum_{j \in \mathcal{N}_{\mathrm{fg}}(n)} p_j, \tag{3}$$

where $\mathcal{N}_{\mathrm{fg}}(n)$ contains the k_{fg} nearest tokens from other images in the batch under feature cosine similarity, and $\alpha \in [0, 1]$ is the mixing weight.

Targets $u_n > 0.5$ indicate foreground, while $u_n < 0.5$ indicate background. We define a signed prediction $\delta_n = 2m_n - 1$, a target sign $\sigma_n = \mathrm{sign}(2u_n - 1)$, and a confidence weight $\omega_n = |2u_n - 1|$. The confidence-weighted hinge loss is

$$L_{\mathrm{fg}} = \frac{\sum_n \omega_n \max(0, \tau - \sigma_n \delta_n)}{\sum_n \omega_n}, \tag{4}$$

Correspondence. The correspondence loss encourages matched foreground tokens across images to share a part assignment (Hamilton et al., 2022; Seong et al., 2023). We sample tokens from the batch whose predicted foreground probability m_n exceeds a threshold. For each sampled token, its k_{corr} nearest neighbours from other images within the sample form positive pairs, using feature cosine similarity. Negative pairs are formed by drawing partner tokens uniformly from the same sample with replacement. Pair selection is not differentiated, but gradients flow through both tokens' assignments.

Let $\mathcal{P}^+$ and $\mathcal{P}^-$ denote the collections of positive and negative pairs, including repeated draws. The foreground-conditional part distribution is $\hat{\boldsymbol{q}}_i = \boldsymbol{q}_i^{\mathrm{parts}} / m_i$, where $\boldsymbol{q}_i^{\mathrm{parts}}$ contains the K part probabilities. For a pair (i, j), the agreement $\Pi_{ij} = \hat{\boldsymbol{q}}_i^\top \hat{\boldsymbol{q}}_j$ is the probability that independent assignments from these distributions give the same part. The loss rewards agreement on positive pairs beyond the agreement of random pairs,

$$L_{\mathrm{corr}} = 1 - \frac{1}{|\mathcal{P}^+|} \sum_{(i,j) \in \mathcal{P}^+} \Pi_{ij} + \frac{1}{|\mathcal{P}^-|} \sum_{(i,j) \in \mathcal{P}^-} \Pi_{ij}. \tag{5}$$

Spatial smoothness. Total variation regularization encourages spatially coherent part assignments (Rudin et al., 1992; Aniraj et al., 2024). Edge weights reduce the penalty for assignment changes across image edges. Let $\mathcal{E}_{\mathrm{h}}$ contain horizontally adjacent token pairs and $\mathcal{E}_{\mathrm{v}}$ vertically adjacent ones. The loss is

$$L_{\mathrm{tv}} = \frac{1}{B}\sum_{b=1}^{B}\sum_{d\in\{\mathrm{h},\mathrm{v}\}}\frac{1}{|\mathcal{E}_d|}\sum_{(i,j)\in\mathcal{E}_d} w_{b,ij}\sum_{s=1}^{K}|q_{b,i,s}-q_{b,j,s}|, \tag{6}$$

where B is the batch size and $q_{b,i,s}$ is the probability of assigning token i in image b to part slot s. The weights $w_{b,ij}$ decrease with the Euclidean distance between the mean RGB values of neighbouring image patches and are normalized to mean one over all neighbour pairs in each image.

Slot usage. The usage loss discourages assignments from concentrating in a few part slots and sharpens the assignments of foreground tokens. The fraction of foreground assignment mass held by slot s across the batch is

$$r_s = \frac{\sum_{b,n} q_{b,n,s}}{\sum_{s'=1}^{K}\sum_{b,n} q_{b,n,s'}}, \qquad s = 1,\dots,K.$$

The loss combines a floor penalty, a cap penalty, and a commitment term,

$$L_{\mathrm{use}} = K\sum_{s=1}^{K}\left(\frac{f}{K}-r_s\right)_+^2 + \left(\max_{s=1,\dots,K} r_s - c\right)_+^2 + L_{\mathrm{commit}}, \tag{7}$$

where $(x)_+ = \max(0,x)$, and f/K and c are the thresholds of the soft floor and cap penalties.

The commitment term L_{commit} uses cross-entropy to match the foreground-conditional distribution $\hat{\boldsymbol{q}}_n$ to a detached target obtained by sharpening the same distribution and applying Sinkhorn normalization (Caron et al., 2020). It is averaged over tokens whose foreground probability m_n exceeds a threshold.

Warp consistency. Warp consistency encourages stable part assignments under token-grid resampling. For image b, a random affine transformation T_b resamples the feature grid. The shared head predicts assignments on the transformed features using independent Gumbel noise. The resulting maps are resampled back to the original grid using T_b^{-1}. Let $\boldsymbol{q}_{b,s} = (q_{b,1,s},\dots,q_{b,N,s})^\top$ be the original assignment map for part slot s and $\tilde{\boldsymbol{q}}_{b,s}$ the map obtained after transformation and inverse resampling. The foreground-weighted cosine loss is

$$L_{\mathrm{cons}} = 1 - \frac{1}{BK}\sum_{b=1}^{B}\sum_{s=1}^{K}\cos(\boldsymbol{m}_b\odot\boldsymbol{q}_{b,s}, \boldsymbol{m}_b\odot\tilde{\boldsymbol{q}}_{b,s}), \tag{8}$$

where $\boldsymbol{m}_b$ contains the original foreground probabilities, $\odot$ denotes elementwise multiplication, and the cosine is computed over spatial positions.

4 Experiments

4.1 Setup

Benchmarks. CUB-200-2011 (Wah et al., 2011) and NABirds (Van Horn et al., 2015) are fine-grained bird benchmarks with annotated keypoints. PartImageNet-OOD (He et al., 2022a) provides pixel-level part annotations across multiple object categories. A separate head is trained for each benchmark and each value of K, using only training images without annotations. Input settings and training details are provided in Appendices B.1 and B.3.

Evaluation metrics. Normalized mutual information (NMI) and adjusted Rand index (ARI) measure agreement between predicted and annotated part labels. They are computed at visible keypoints on CUB and NABirds, and over pixels, including background, on PartImageNet-OOD. Normalized mean error (NME) evaluates landmark localization on the bird benchmarks using a linear regressor from predicted part centroids to annotated keypoints.

Table 2: Quantitative comparisons at different values of K. Bold marks the best reported value. *Our reproduction. ‡Retrained at the listed resolution.

Method	Res.	CUB ($K = 4$)			CUB ($K = 8$)			CUB ($K = 16$)		
		NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑
DINO	518	–	31.18	11.21	–	47.21	19.76	–	50.57	26.14
PDiscoNet + ViT-B	518	7.70	52.59	26.66	6.34	65.01	37.90	5.95	68.63	43.41
PDiscoFormer	518	**7.41**	58.13	25.11	5.99	69.87	43.49	5.74	73.38	55.83
MPAE	448	7.75	**58.93**	**33.73**	**5.53**	68.24	45.85	**5.14**	79.00	66.56
LinPart (Ours)	518	8.61	58.66	27.33	6.14	**71.49**	**46.05**	5.62	**79.97**	**69.37**
		±0.95	±1.20	±3.34	±0.21	±1.96	±3.46	±0.02	±0.30	±0.85

Method	Res.	NABirds ($K = 4$)			NABirds ($K = 8$)			NABirds ($K = 11$)		
		NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑
DINO	518	–	26.50	11.30	–	39.45	23.20	–	39.23	23.37
PDiscoNet + ViT-B	518	**9.76**	43.02	22.77	9.17	56.50	34.10	9.34	68.92	54.65
PDiscoFormer	518	11.22	48.24	27.73	8.84	60.39	46.74	8.36	72.04	63.35
MPAE*	448	10.43	54.81	**33.36**	**7.99**	**72.71**	**62.64**	7.62	70.25	59.40
LinPart (Ours)	518	10.30	**54.95**	30.49	8.20	68.04	47.88	**7.42**	**76.97**	**70.66**
		±0.07	±5.05	±4.02	±0.17	±2.08	±2.22	±0.08	±2.94	±5.16

Method	Res.	OOD ($K = 8$)		OOD ($K = 25$)		OOD ($K = 50$)	
		NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑
DINO	224	19.17	7.59	31.46	14.16	37.81	16.50
PDiscoNet + ViT-B	224	19.28	34.72	28.23	50.35	29.48	27.80
PDiscoFormer	224	29.00	52.40	44.71	59.27	46.29	62.21
PDiscoFormer‡	448	39.55	68.85	48.06	69.61	46.06	46.86
MPAE	224	32.90	66.17	39.28	68.87	53.65	74.22
MPAE‡	448	16.25	38.08	33.79	73.43	34.11	75.58
LinPart (Ours)	224	58.12	71.54	59.74	72.77	58.04	71.71
		±1.59	±0.53	±1.96	±1.01	±0.93	±0.90
LinPart (Ours)	448	**58.46**	**80.35**	**63.01**	**80.94**	**64.14**	**81.16**
		±1.11	±0.65	±1.12	±1.12	±1.92	±1.05

Quantitative results. Table 2 shows that LinPart achieves the highest mean NMI and ARI at $K = 16$ on CUB, $K = 11$ on NABirds, and $K = 25$ on PartImageNet-OOD while training only a shared linear head. The largest margins are on PartImageNet-OOD. At 448 pixels, LinPart achieves the highest NMI and ARI at every evaluated K, exceeding the strongest compared result by 10.49–18.91 NMI points. It also outperforms MPAE and PDiscoFormer retrained at the same resolution (Appendix B.6). At 224 pixels, LinPart retains the highest NMI among methods evaluated at that resolution for every K, although MPAE achieves higher ARI at $K = 50$. On NABirds at $K = 11$, LinPart leads on all three metrics, exceeding PDiscoFormer by 4.93 NMI and 7.31 ARI points. On CUB, it achieves the highest mean NMI and ARI at $K = 8$ and 16, without the image class labels used by PDiscoNet and PDiscoFormer. The advantage is less consistent at smaller part counts. MPAE achieves higher NMI and ARI on CUB at $K = 4$ and NABirds at $K = 8$, and higher ARI on NABirds at $K = 4$. LinPart also does not consistently achieve the lowest NME.

Qualitative results. Figure 2 compares the predicted parts on the same images. In several CUB and NABirds examples, LinPart produces contiguous wing and body regions, while MPAE divides these regions into smaller segments. On PartImageNet-OOD, LinPart groups most of the dog's head into one part, whereas MPAE and PDiscoFormer divide it into several parts. In the primate example, LinPart also avoids the large foreground region that PDiscoFormer predicts on the surrounding wall.

Training efficiency. Table 3 compares training costs on PartImageNet-OOD at $K = 50$ and 448×448 resolution. LinPart trains $25\times$ fewer parameters than PDiscoFormer and $226\times$ fewer than

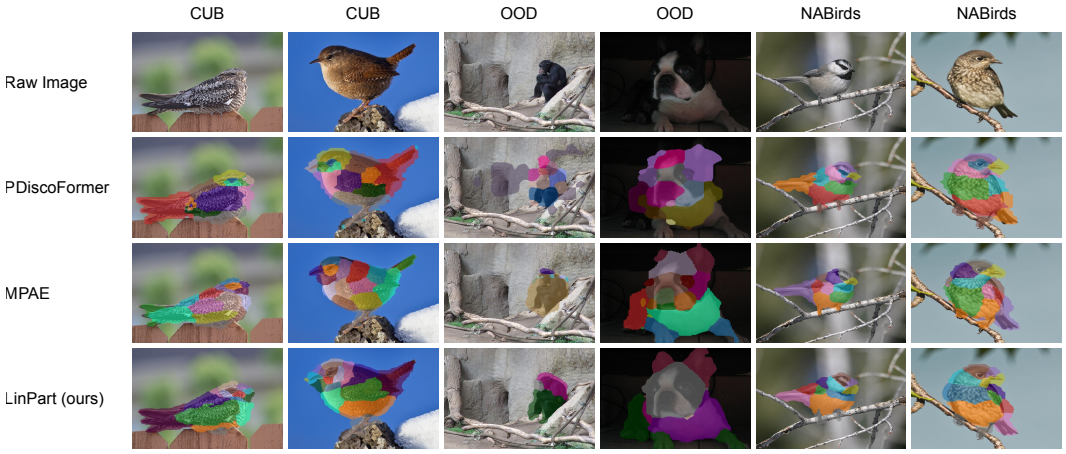

Figure 2: Qualitative comparisons on CUB ($K = 16$), PartImageNet-OOD ($K = 50$), and NABirds ($K = 11$).

Table 3: Training efficiency on OOD.

Method	Trainable params. (K)	Epochs	Time (s/step)	Peak alloc. (GiB)
PDiscoFormer	991.5	28	2.028	56.70
MPAE	8879.8	70	1.034	44.32
LinPart	**39.3**	**10**	**0.486**	**2.96**

Table 4: Effect of head width and depth.

Head	CUB (K=16)	NABirds (K=11)	OOD (K=50)
Linear	**79.97 / 69.37**	**76.97 / 70.66**	64.14 / 81.16
MLP (256)	68.63 / 47.76	63.23 / 46.26	64.82 / 81.86
MLP (512)	62.18 / 36.60	67.57 / 55.14	**66.56 / 83.51**
MLP (1024)	62.22 / 39.32	60.39 / 42.72	65.24 / 82.77
MLP (1024×2)	73.27 / 57.86	61.86 / 44.66	63.75 / 82.33
MLP (1024×3)	23.37 / 16.95	54.56 / 35.02	60.76 / 80.10

MPAE. On the same GPU with FP32 computation and batch size 64, its training step is $4.2\times$ faster than PDiscoFormer's and $2.1\times$ faster than MPAE's. Its peak allocated memory is 15–19$\times$ smaller. LinPart also uses a 10-epoch training schedule, compared with 28 epochs for PDiscoFormer and 70 for MPAE. Measurement details are provided in Appendix C.1.

4.2 ABLATION STUDY

Head capacity. Table 4 compares linear and MLP heads of varying width and depth over three seeds. All heads use the same frozen backbone, objective, and training schedule. MLP learning rates are calibrated to match the magnitude of the first-step logit change of a linear reference. The linear row repeats the LinPart results of Table 2 and achieves the highest mean NMI and ARI on CUB and NABirds. The MLP with three 1024-unit hidden layers collapses to background in two of the three CUB runs (Appendix B.7). On PartImageNet-OOD, all single-hidden-layer MLPs achieve higher mean NMI and ARI than the linear head. The 512-unit head achieves the highest scores, improving NMI by 2.42 points and ARI by 2.35 points. Under this training protocol, increasing head width or depth does not consistently improve performance.

Loss ablation. Table 5 compares the full objective with each term removed. Removing warp consistency produces the largest mean NMI drops on CUB and NABirds, at 17.59 and 26.52 points, respectively. On PartImageNet-OOD, removing the foreground anchor has the largest effect, reducing mean NMI from 64.14 to 3.05. Removing spatial smoothness or slot usage lowers mean NMI and ARI on all three benchmarks. Removing correspondence also lowers mean NMI throughout, although ARI increases on PartImageNet-OOD.

Reconstruction has a benchmark-dependent effect. On CUB, removing it leaves mean NMI nearly unchanged but increases its standard deviation from 0.30 to 3.89 across the three seeds. On NABirds, both mean scores decrease, whereas both increase on PartImageNet-OOD. Removing the foreground anchor also improves mean scores on CUB. These results show that the terms contribute differently

Table 5: Loss ablation. Mean ± s.d. over three seeds.

	CUB ($K = 16$)		NABirds ($K = 11$)		OOD ($K = 50$)	
Objective	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑
Full objective	79.97 ±0.30	69.37 ±0.85	76.97 ±2.94	70.66 ±5.16	64.14 ±1.92	81.16 ±1.05
w/o reconstruction	79.96 ±3.89	69.62 ±7.03	73.64 ±1.33	64.58 ±1.25	65.68 ±1.28	81.68 ±0.45
w/o foreground anchor	80.85 ±1.33	71.19 ±2.34	71.25 ±4.02	58.55 ±12.93	3.05 ±0.16	2.38 ±0.17
w/o correspondence	76.78 ±1.56	61.29 ±2.75	71.12 ±1.19	56.40 ±2.06	60.51 ±2.14	82.50 ±1.00
w/o spatial smoothness	70.54 ±1.55	47.96 ±6.41	66.54 ±0.89	46.15 ±2.25	54.17 ±0.49	67.67 ±0.28
w/o slot usage	76.68 ±6.96	60.11 ±16.27	76.56 ±1.92	68.40 ±5.69	61.26 ±1.51	80.48 ±0.54
w/o warp consistency	62.38 ±4.56	35.63 ±5.27	50.45 ±3.04	31.05 ±6.29	64.74 ±1.05	80.97 ±0.48

Table 6: Comparison of merge criteria and direct training. Mean scores over seeds 1–3.

Dataset	Parts	Cosine		Rec. only		Corr. + rec.		Final score		Direct training	
	Source → target	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
CUB	16 → 8	70.66	41.00	69.88	38.24	70.54	43.42	74.41	53.37	71.49	46.05
	16 → 4	47.31	16.33	55.88	23.33	55.07	23.20	54.80	21.47	58.66	27.33
NABirds	11 → 8	69.03	49.83	70.33	50.90	68.29	46.76	71.79	55.36	68.04	47.88
	11 → 4	29.22	6.60	58.57	32.46	52.37	27.71	60.52	36.76	54.95	30.49
OOD	50 → 25	64.20	81.09	63.99	81.15	63.58	81.04	63.31	81.05	63.01	80.94
	50 → 8	52.02	71.70	56.97	79.36	62.39	80.77	61.22	80.72	58.46	80.35

across benchmarks, with the largest effects arising from warp consistency on the bird benchmarks and foreground guidance on PartImageNet-OOD.

5 Coarsening parts without retraining

A trained head can provide partitions with fewer parts by merging its predicted labels. Candidate merges are ranked using a score derived from the correspondence, reconstruction, and warp-consistency terms of the objective, while the head parameters and foreground predictions remain fixed.

5.1 A merge score from the objective

The positive pairs of the correspondence loss define edges between sampled foreground tokens. Under hard assignments, positive-pair agreement is the fraction of edges within parts. Large parts can contain many such edges by chance. Subtracting the expected agreement of uniformly sampled token pairs corrects for this chance agreement and gives a modularity score (Newman, 2006; Reichardt & Bornholdt, 2006). Unlike standard degree-based modularity, its chance baseline depends only on part sizes. For fixed assignments, sample, and graph, the correspondence loss averaged over uniform negative sampling equals one minus this score.

Merging parts r and s increases modularity by $e_{rs} - 2p_r p_s$. Here e_{rs} is the fraction of positive edges connecting the two parts in either direction, and p_r and p_s are their shares of the sampled tokens. The product $2p_r p_s$ is the probability that a uniformly sampled pair connects these parts. Let B_{rs} denote the normalized increase in reconstruction error after recomputing the merged per-image prototypes. This cost is positive when the parts co-occur and have different mean features. Combining these changes with the change in warp consistency gives

$$s_{rs} = e_{rs} - 2p_r p_s - \frac{\lambda_{\text{rec}}}{\lambda_{\text{corr}}} B_{rs} - \frac{\lambda_{\text{cons}}}{\lambda_{\text{corr}}} \Delta L_{\text{cons}}, \tag{9}$$

where ΔL_{cons} is the consistency loss after merging minus the loss before merging. The score measures the decrease in the expected hard-assignment objective formed by these three terms, divided by λ_{corr}. Foreground guidance remains unchanged, while slot usage and spatial smoothness are omitted. A positive score therefore does not guarantee a decrease in the full training loss or an increase in annotation agreement.

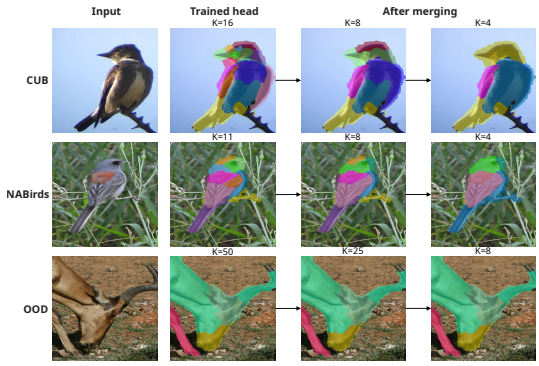

Figure 3: Coarsening predicted parts without retraining.

5.2 Results

Table 6 compares four merge criteria with direct training at the target K. The final score of Eq. 9 yields higher mean NMI and ARI than direct training in five of the six settings. On CUB at eight parts and at both NABirds targets, the gains range from 2.92 to 5.57 NMI points and from 6.27 to 7.48 ARI points. Direct training remains better on CUB at four parts. On PartImageNet-OOD, reducing the 50-part head to 25 parts lowers mean NMI and ARI by 0.83 and 0.11 points. Reducing it to eight parts lowers them by 2.92 and 0.44 points, compared with a 5.68 point NMI difference between direct training at 50 and eight parts.

The comparison among merge orders isolates the criterion by holding the source heads, training images, and foreground predictions fixed. The final score achieves the highest mean NMI and ARI on CUB at eight parts and at both NABirds targets. Adding warp consistency to correspondence plus reconstruction raises mean NMI by 3.50–8.15 points in these settings. Simpler criteria achieve higher means in the remaining settings. On PartImageNet-OOD at 25 parts, the four criteria differ by at most 0.89 NMI and 0.11 ARI points.

Predicted slot usage provides context for these differences. On the sampled training images, PartImageNet-OOD uses an average of 4.7–6.0 of the 50 slots per image, whereas the bird benchmarks use nearly all slots on average. Counting label changes relative to the largest constituent slot of each merged group, merging to 25 parts changes only 3.5%–5.6% of foreground token labels on PartImageNet-OOD. Merging to eight parts changes 12%–36% on the bird benchmarks. Thus, a large reduction in the global part count can involve relatively few token-label changes. Per-seed results and standard deviations are provided in Appendix D.3.

6 Conclusion and Limitations

LinPart shows that part information in frozen DINOv2 features can support learning consistent assignments without part annotations. Foreground guidance and cross-image correspondence provide training signals for organizing this information, allowing a shared linear head to achieve competitive part discovery without adapting the representation. The results support focusing on how existing features are assigned to parts, while the capacity experiments show that the benefit of a nonlinear head depends on the benchmark. Beyond training, changes in the objective provide a criterion for regrouping the predicted labels. A trained partition can therefore be reused at several granularities without learning a separate head for each target.

LinPart relies on the part information available in the frozen backbone features and cannot refine the representation for part discovery. Its linear head restricts assignments to affine decision regions. Shallow MLPs achieve higher mean scores than the linear head on PartImageNet-OOD. The target number of parts must be specified in advance. Coarsening only merges existing part labels and leaves foreground–background assignments unchanged. It cannot split parts into finer regions, correct foreground errors, or recover distinctions already lost in the initial predictions.

REPRODUCIBILITY STATEMENT

The architecture and objective are described in Section 3, and the benchmarks and evaluation metrics in Section 4.1. Appendix B specifies input settings, training hyperparameters, loss implementation, and inference. Appendix A gives the clustering and probe settings, and Appendix B.2 the sampling and reconnected control of the correspondence analysis. Per-seed results underlying the main comparison are listed in Appendix C, and the cost measurement in Appendix C.1. The merge-score derivation, training-image statistics, and comparison of merge criteria are provided in Appendix D.

AI USE STATEMENT

In this work, we did not use generative AI tools for any of the tasks with required disclosure; all research ideas, methods, experiments, and analyses were developed and conducted entirely by the authors. We used generative AI tools for creating or editing software code (writing utility scripts, fixing bugs, and refactoring existing code) and for editing the research paper to improve readability (rephrasing sentences, correcting grammar, and improving the overall flow of the text). We have reviewed all AI-assisted work. LLM-assisted code was verified and tested for correctness by the authors, and all LLM-edited text was checked by the authors to ensure that the original meaning and technical claims were preserved and that no new content was introduced. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

Ananthu Aniraj, Cassio F Dantas, Dino Ienco, and Diego Marcos. Pdiscoformer: Relaxing part discovery constraints with vision transformers. In *European Conference on Computer Vision*, pp. 256–272. Springer, 2024.

Görkay Aydemir, Weidi Xie, and Fatma Guney. Self-supervised object-centric learning for videos. *Advances in Neural Information Processing Systems*, 36:32879–32899, 2023.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. *Advances in Neural Information Processing Systems*, 34:28104–28118, 2021.

Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 336–352, 2018.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024.

Qingzhe Gao, Bin Wang, Libin Liu, and Baoquan Chen. Unsupervised co-part segmentation through assembly. In *International Conference on Machine Learning*, pp. 3576–3586. PMLR, 2021.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=SaKO6z6Hl0c.

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*, pp. 128–145. Springer, 2022a.

Xingzhe He, Bastian Wandt, and Helge Rhodin. Ganseg: Learning to segment by unsupervised hierarchical image generation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1215–1225. IEEE, 2022b.

Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8659–8669. IEEE, 2020.

Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 869–878. IEEE, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Unsupervised part segmentation through disentangling appearance and shape. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8351–8360. IEEE, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

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. *Transactions on Machine Learning Research*, 2024. arXiv:2304.07193.

Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 74(1):016110, 2006.

Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

Oindrila Saha and Subhransu Maji. Particle: Part discovery and contrastive learning for fine-grained recognition. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 167–176. IEEE, 2023.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=b9tUk-f_aG.

Hyun Seok Seong, WonJun Moon, SuBeen Lee, and Jae-Pil Heo. Leveraging hidden positives for unsupervised semantic segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19540–19549. IEEE, 2023.

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.

Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonín Vobeckỳ, Éloi Zablocki, and Patrick Pérez. Unsupervised object localization: Observing the background to discover objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3176–3186, 2023.

James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3229–3238. IEEE, 2017.

Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023.

Robert Van Der Klis, Stephan Alaniz, Massimiliano Mancini, Cassio F Dantas, Dino Ienco, Zeynep Akata, and Diego Marcos. Pdisconet: Semantically consistent part discovery for fine-grained recognition. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1866–1876. IEEE, 2023.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pp. 268–285. Springer, 2020.

Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 595–604, 2015.

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

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14523–14533. IEEE, 2022.

Jiahao Xia, Wenjian Huang, Min Xu, Jianguo Zhang, Haimin Zhang, Ziyu Sheng, and Dong Xu. Unsupervised part discovery via dual representation alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10597–10613, 2024.

Jiahao Xia, Yike Wu, Wenjian Huang, Jianguo Zhang, and Jian Zhang. Unsupervised part discovery via descriptor-based masked image restoration with optimized constraints. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8668–8677. IEEE, 2025.

Romain Xu-Darme, Georges Quénot, Zakaria Chihani, and Marie-Christine Rousset. Particul: part identification with confidence measure using unsupervised learning. In *International Conference on Pattern Recognition*, pp. 173–187. Springer, 2022.

Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2694–2703. IEEE, 2018.

Jie Zhu, Jiyang Qi, Mingyu Ding, Xiaokang Chen, Ping Luo, Xinggang Wang, Wenyu Liu, Leye Wang, and Jingdong Wang. Understanding self-supervised pretraining with part-aware representation learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=HP7Qpui5YE.

Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14482–14491. IEEE, 2022.

A PART INFORMATION IN FROZEN FEATURES

Table 7: Clustering results by restart seed on PartImageNet-OOD. FG denotes foreground-based centroid fitting. Highest reports per-metric maxima, which may occur at different seeds.

	$K = 8$				$K = 25$				$K = 50$			
	k-means		+ FG		k-means		+ FG		k-means		+ FG	
Seed	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑
0	35.16	7.50	49.37	65.02	40.26	7.71	54.70	65.97	36.70	2.95	52.31	65.96
1	38.84	17.36	53.09	65.14	39.06	6.82	52.71	65.94	36.87	3.17	52.56	66.00
2	45.63	24.17	47.33	63.16	40.18	6.21	53.54	65.94	36.53	3.17	52.59	66.03
3	41.09	27.35	48.14	64.23	35.79	5.44	52.69	65.98	36.75	3.58	50.65	65.89
4	38.28	13.24	46.74	64.60	40.63	7.14	53.49	66.01	37.15	3.24	52.67	65.99
5	44.26	18.60	56.90	65.69	40.06	5.91	51.42	65.79	36.20	3.40	52.47	66.00
6	44.53	19.31	52.19	64.55	36.56	5.81	53.14	65.83	36.10	3.46	53.20	66.00
7	40.36	12.46	52.85	65.27	38.25	5.44	52.77	65.92	36.06	3.22	52.95	66.01
8	38.76	18.62	54.46	65.48	38.09	6.88	52.83	65.88	37.01	3.36	52.88	66.06
9	38.53	13.91	51.66	65.13	37.93	5.98	52.48	65.89	36.93	3.18	52.73	66.03
10	38.83	18.36	50.86	64.76	34.74	4.50	53.19	65.98	36.63	3.20	52.95	65.98
11	38.28	13.26	53.98	65.13	36.79	5.12	52.91	65.80	37.36	3.77	52.48	65.96
12	39.07	18.62	49.37	65.01	38.57	7.20	52.84	65.97	36.47	3.27	53.14	66.04
13	36.75	8.54	51.48	65.14	39.48	6.98	53.45	66.00	36.92	3.31	51.70	65.95
14	38.79	18.75	49.49	64.73	38.14	6.15	53.27	65.96	37.02	3.60	52.45	65.96
15	37.45	12.87	54.01	65.03	40.39	6.47	50.52	65.83	37.95	3.60	51.05	65.93
16	39.13	19.45	50.95	65.07	40.43	6.29	51.70	65.85	34.33	2.93	52.68	65.98
17	42.82	27.24	46.99	64.51	38.04	6.71	53.71	66.00	37.41	3.18	52.98	66.01
18	38.80	18.85	51.22	65.12	38.29	7.65	54.29	66.04	38.07	3.54	52.22	65.96
19	32.25	10.76	52.86	65.27	39.23	7.86	52.84	65.86	36.47	3.13	52.50	65.98
Mean	39.38	16.96	51.20	64.90	38.54	6.41	52.92	65.92	36.75	3.31	52.46	65.99
S.d.	3.16	5.46	2.72	0.54	1.64	0.90	0.94	0.08	0.78	0.22	0.65	0.04
Highest	**45.63**	**27.35**	**56.90**	**65.69**	**40.63**	**7.86**	**54.70**	**66.04**	**38.07**	**3.77**	**53.20**	**66.06**

Table 8: Clustering results by restart seed on CUB. FG denotes foreground-based centroid fitting. Highest reports per-metric maxima, which may occur at different seeds.

	$K = 4$				$K = 8$				$K = 16$			
	k-means		+ FG		k-means		+ FG		k-means		+ FG	
Seed	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑
0	0.37	0.02	0.85	0.04	50.55	20.96	58.10	24.85	58.29	24.91	58.17	24.85
1	28.42	7.12	0.99	0.05	43.97	13.31	28.16	7.22	50.30	20.93	57.83	24.81
2	0.56	0.03	0.41	0.02	28.23	7.13	28.14	7.18	43.48	13.02	58.17	24.85
3	0.56	0.03	1.05	0.05	28.20	7.12	32.79	7.81	41.74	13.90	62.36	26.86
4	0.28	0.01	0.79	0.04	0.50	0.02	55.71	22.92	56.55	23.48	63.02	27.43
5	0.56	0.03	28.35	7.28	0.48	0.02	50.34	20.91	55.83	23.09	62.63	30.78
6	0.42	0.02	0.85	0.04	43.83	13.27	28.15	7.20	28.28	7.13	63.80	32.89
7	28.38	7.18	1.05	0.05	28.27	7.10	55.71	22.98	58.32	24.91	61.84	26.62
8	0.28	0.01	28.35	7.28	28.25	7.11	32.26	7.67	50.24	20.91	62.24	27.07
9	0.51	0.02	0.41	0.02	28.22	7.13	55.70	22.98	55.82	23.08	59.21	27.96
10	0.51	0.02	1.05	0.05	28.27	7.10	28.12	7.21	55.88	23.19	58.13	24.84
11	0.51	0.02	28.35	7.28	28.23	7.13	28.22	7.21	54.84	22.80	62.35	26.84
12	0.51	0.02	0.72	0.03	0.48	0.02	60.31	31.99	43.49	13.05	66.12	37.00
13	0.40	0.02	1.05	0.05	28.23	7.13	33.27	7.91	55.84	23.09	66.02	36.95
14	28.37	7.19	1.05	0.05	0.50	0.02	58.14	25.33	58.29	24.91	57.97	27.26
15	41.00	12.87	28.36	7.25	0.55	0.03	50.22	20.88	49.88	20.98	42.35	14.22
16	28.37	7.19	28.35	7.28	28.20	7.12	62.64	26.92	28.26	7.13	57.46	24.58
17	0.49	0.02	0.99	0.05	0.52	0.02	50.34	20.91	43.73	13.23	44.17	13.47
18	0.49	0.02	28.36	7.25	28.23	7.13	28.23	7.20	50.28	20.93	60.51	32.47
19	0.51	0.02	1.05	0.05	28.20	7.12	44.39	13.52	41.67	13.80	65.60	36.90
Mean	8.08	2.09	9.12	2.21	22.60	6.30	43.45	16.04	49.05	18.92	59.50	27.43
S.d.	13.77	3.86	12.92	3.40	16.17	5.41	13.41	8.68	9.17	5.90	6.23	6.26
Highest	**41.00**	**12.87**	**28.36**	**7.28**	**50.55**	**20.96**	**62.64**	**31.99**	**58.32**	**24.91**	**66.12**	**37.00**

Table 9: Clustering results by restart seed on NABirds. FG denotes foreground-based centroid fitting. Highest reports per-metric maxima, which may occur at different seeds.

	$K = 4$				$K = 8$				$K = 11$			
	k-means		+ FG		k-means		+ FG		k-means		+ FG	
Seed	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑
0	22.09	5.99	1.12	-0.02	22.32	5.92	47.13	23.83	47.03	23.64	56.78	31.54
1	0.54	-0.02	1.12	-0.02	41.97	17.59	40.07	19.46	21.60	5.71	59.30	36.65
2	0.54	-0.02	1.12	-0.02	22.37	5.94	46.94	23.77	2.80	0.00	56.03	31.11
3	0.54	-0.02	1.12	-0.02	22.37	5.94	44.03	19.50	41.16	16.89	56.35	31.34
4	0.54	-0.02	26.06	9.08	1.42	-0.04	46.31	23.30	55.99	30.97	34.03	16.54
5	0.54	-0.02	22.50	6.40	24.12	7.11	56.24	31.04	46.45	23.21	59.62	37.16
6	21.60	6.06	22.50	6.40	22.06	5.91	42.35	17.18	21.80	5.73	56.60	31.56
7	0.54	-0.02	22.50	6.40	21.84	5.90	24.00	7.78	55.94	30.86	56.67	31.60
8	1.02	-0.01	22.54	6.31	3.00	-0.03	42.38	17.95	21.69	5.71	39.95	14.77
9	1.01	-0.01	22.55	6.31	22.37	5.93	56.98	31.57	47.00	23.65	56.85	31.53
10	1.27	-0.04	22.50	6.40	1.75	-0.05	57.08	31.53	40.30	14.77	56.85	31.54
11	0.54	-0.02	22.50	6.40	42.23	16.32	56.97	31.56	56.75	30.78	56.68	31.51
12	1.27	-0.04	1.12	-0.02	21.88	5.91	46.31	23.30	21.99	5.76	56.68	31.57
13	0.54	-0.02	22.50	6.40	22.40	5.87	60.34	36.19	21.90	5.74	59.77	37.30
14	0.54	-0.02	1.12	-0.02	21.94	5.90	39.19	17.63	21.65	5.71	56.83	31.54
15	0.54	-0.02	1.12	-0.02	22.37	5.93	47.10	23.81	21.60	5.71	54.71	31.26
16	0.54	-0.02	1.12	-0.02	21.90	5.92	45.01	22.70	46.47	23.22	52.54	30.06
17	1.03	-0.01	22.50	6.40	1.47	-0.04	57.19	31.59	41.43	17.26	56.07	31.08
18	1.03	-0.01	22.50	6.40	1.72	-0.04	44.29	19.76	46.44	23.20	52.58	30.08
19	0.54	-0.02	46.50	23.19	21.84	5.90	41.75	19.70	21.56	5.72	56.32	31.44
Mean	2.84	0.58	15.33	4.80	19.17	5.59	47.08	23.66	35.08	15.21	54.56	30.56
S.d.	6.51	1.86	13.01	5.47	11.88	4.71	8.58	6.85	15.54	10.33	6.35	5.54
Highest	**22.09**	**6.06**	**46.50**	**23.19**	**42.23**	**17.59**	**60.34**	**36.19**	**56.75**	**30.97**	**59.77**	**37.30**

Clustering baselines. Clustering baselines. Euclidean k-means is applied to frozen DINOv2-B/14 patch features with one set of centroids shared across images. Centroids are fitted to at most 400,000 tokens sampled without replacement from the evaluation images. Each of 20 restarts uses k-means++ initialization and 50 Lloyd iterations. Table 1 reports the restart with the lowest fitting inertia, which is marked in Tables 7–9 together with the mean, standard deviation, and maximum over restarts. No annotations are used for fitting or restart selection.

Foreground-restricted clustering. The foreground variant fits centroids only to tokens whose unsupervised objectness score exceeds 0.5. CUB and PartImageNet-OOD use PCA-based scores pooled across images, while the recorded NABirds baseline uses per-image PCA scores. On CUB and NABirds, the fitted centroids assign every token to one of K clusters. On PartImageNet-OOD, K centroids are fitted to foreground tokens, and tokens outside the estimated foreground are assigned to background. The all-token baseline on this benchmark instead uses $K + 1$ clusters.

Supervised linear probes. Supervised linear probes assess whether annotated part identities can be recovered from the frozen features by an affine classifier. Unlike the clustering baselines, these probes use part annotations to learn the assignment rule. The PartImageNet-OOD probe predicts 40 annotated part classes and background, whereas its clustering baseline uses $K = 25$ foreground clusters. This comparison therefore measures accessible part information rather than performance at a matched output count.

Evaluation protocol. The clustering diagnostics use token-level predictions. On CUB and NABirds, the predicted label is read from the token containing each visible keypoint. On PartImageNet-OOD, token labels are upsampled by nearest-neighbor interpolation to the annotation grid, and all pixels, including background, are evaluated. NMI and ARI are computed over the aggregated predictions within each benchmark. CUB uses square-frame features, NABirds uses the recorded approximately 6,000-image evaluation subset, and PartImageNet-OOD uses the crop-frame features at 448×448. These diagnostic settings differ from the main evaluation protocol, which bilinearly upsamples logits before assigning labels.

B IMPLEMENTATION DETAILS

B.1 INPUT SETTINGS AND ARCHITECTURE

Input resolution is 518×518 for CUB and NABirds and 448×448 for PartImageNet-OOD, producing patch grids of 37×37 and 32×32, respectively. The main comparisons evaluate $K \in \{4, 8, 16\}$ on CUB, $K \in \{4, 8, 11\}$ on NABirds, and $K \in \{8, 25, 50\}$ on PartImageNet-OOD. A separate head is trained for each benchmark and part count. All parameters of the DINOv2-B/14 backbone remain frozen, including the position, class, and register tokens. We use the patch features after the final layer normalization and discard the class and register outputs. The head contains $770(K + 1)$ trainable parameters, including the linear weights and two bias vectors.[1]

B.2 CROSS-IMAGE CORRESPONDENCE

Samples and neighbours. The correspondence diagnostic in Table 1 uses 512 fixed evaluation images per benchmark, sampled with seed 20260923 and divided into eight batches of 64. CUB and NABirds use tokens containing visible annotated keypoints. Keypoints that share a token are retained as separate annotated instances. CUB uses square-frame features at 518×518, and NABirds uses its cached evaluation frame at the same resolution. PartImageNet-OOD uses crop-frame features at 448×448. Each batch samples up to 2,048 tokens without replacement from those with an external PCA foreground score above 0.5, and reads their annotations at token centres. Each sampled token connects to its five nearest sampled tokens from other images under feature cosine similarity. These graphs are diagnostic samples, distinct from the head-dependent graphs used during training. No head is fitted or updated.

Reconnected control. The control randomly swaps destination tokens between edges ending in the same image. This preserves each token's incoming and outgoing connection counts and each source token's number of neighbours in each destination image. Swaps that create duplicate neighbours for a source token are rejected. Each batch uses 20 control graphs, each generated by eight passes of candidate swaps. The sampled tokens remain fixed, and annotations are not used to select neighbours or swaps.

Part-label agreement. An edge agrees when its two tokens have the same annotated part label. Edges with a background source are excluded, while background destinations count as disagreements. Counts are pooled across the eight batches, and the agreement rate is computed for each source-part class. Table 1 reports the mean of these rates over the observed part classes. For the control, edge counts are first averaged over its 20 graphs within each batch.

B.3 TRAINING SETTINGS

Augmentation. At every step each image is augmented by a horizontal flip with probability 0.5 and a random affine map with rotation up to $45°$, translation up to 20% of the image side, and scale in $[0.8, 1.2]$, with reflection padding and no color jitter. The backbone is run on the augmented image, and the tokens, the objectness score, and the two neighbor graphs of Section 3.2 are recomputed from that forward pass; no features are cached during training.

Optimization. Training uses AdamW (Loshchilov & Hutter, 2019) with weight decay 0.01, batch size 64, and a 10-epoch schedule. The learning rate starts at 0.02 and follows cosine decay to zero, updated once per epoch. Computation uses bfloat16 mixed.

During training, Gumbel-Softmax sampling (Jang et al., 2016) gives $\boldsymbol{q}_n = \mathrm{softmax}(\boldsymbol{\ell}_n + \boldsymbol{\epsilon}_n)$, where each component of $\boldsymbol{\epsilon}_n$ is sampled independently from a standard Gumbel distribution. The Gumbel temperature is 1 throughout, and no term of the loss is warmed up or scheduled. The optimizer and sampling settings are shared across benchmarks. Input resolutions, part counts, and loss coefficients are specified separately.

[1] The linear-layer bias and an additional per-slot bias act as a single effective bias; both are included in the parameter count.

B.4 LOSS IMPLEMENTATION

Table 10 lists the loss coefficients. All six weights are 1 on CUB and NABirds. On PartImageNet-OOD at 448 pixels, the foreground, correspondence, and spatial smoothness weights are 2, and the remaining weights are 1. The 224-pixel runs of LinPart in Table 2 keep the 448-pixel coefficients and change only the input size. The remaining loss settings are shared across benchmarks and are detailed below.

Table 10: Loss coefficients. Other loss settings are shared across benchmarks.

Term	CUB / NABirds	PartImageNet-OOD (448 px)
Feature reconstruction	1	1
Foreground anchor	1	2
Correspondence	1	2
Spatial smoothness	1	2
Slot usage	1	1
Warp consistency	1	1

Foreground target construction. For each image, we center the frozen token features and approximate their first principal direction using 15 power iterations initialized with the centered token of largest norm, and project onto that direction. We choose the sign so that the mean projection is lower at the image border than in the interior, then min-max scale the projections to obtain $p_n \in [0, 1]$. Cross-image pooling uses $k_{\text{fg}} = 20$ neighbors and mixing weight $\alpha = 0.5$. The hinge margin is $\tau = 0.6$. The scores and neighbors are computed without gradient propagation.

Correspondence sampling. Let B denote the number of images in a batch. We uniformly sample without replacement up to $32B$ tokens from all batch tokens with $m_n > 0.5$, rather than taking a fixed number per image. Thus $S = \min(32B, N_{\text{fg}})$, where N_{fg} is the number passing this threshold. For each anchor, same-image feature similarities are set to -2 before selecting the $k_{\text{corr}} = 5$ largest similarities. Cross-image selection assumes at least five other-image candidates. It returns zero loss when fewer than seven foreground tokens are available. Each anchor also draws 20 negative indices uniformly with replacement from the sample, without excluding itself, same-image tokens, or positive neighbors. Selection is performed without gradients, while assignment agreement differentiates both endpoints. The uniform negative expectation is $\|S^{-1} \sum_i \hat{\boldsymbol{q}}_i\|_2^2$.

Spatial smoothness weights. We average the augmented input image within each token cell. Let $d_{b,ij}^2$ be the squared Euclidean difference between the mean RGB vectors of neighboring cells, and let σ_b^2 be the median of these squared differences over both horizontal and vertical pairs, clamped below at 10^{-6}. We compute $\tilde{w}_{b,ij} = \exp[-d_{b,ij}^2/(2\sigma_b^2)]$ and divide by their mean over all neighbor pairs of image b to obtain $w_{b,ij}$. The weights are computed without gradients. The loss sums absolute differences across the K part rows, excluding background, averages horizontal and vertical pairs separately, adds the two means, and averages images.

Usage target construction. The slot shares are $r_s = \sum_{b,n} q_{b,n,s} / \sum_{s'=1}^{K} \sum_{b,n} q_{b,n,s'}$ for $s = 1, \ldots, K$, excluding the background slot. The floor parameter is $f = 0.16$ and the cap threshold is $c = 0.15$. For tokens with $m_n > 0.3$, we form the foreground-conditional distribution $\hat{\boldsymbol{q}}_n = \boldsymbol{q}_n^{\text{parts}}/m_n$. We raise its entries to the power $1/\epsilon_{\text{SK}}$, with $\epsilon_{\text{SK}} = 0.07$, and apply three Sinkhorn-Knopp iterations toward uniform slot marginals over these tokens. The resulting target Q_n^* is detached from the gradient. The commitment loss is the mean cross-entropy $-\sum_s Q_{n,s}^* \log \hat{q}_{n,s}$ over the selected tokens. It is set to zero when fewer than $2K$ tokens pass the foreground threshold. Before sharpening, the conditional distributions are renormalized across slots. Sinkhorn first normalizes each sharpened token row, then alternates slot and token normalization for three iterations.

Warp consistency settings. For each image, we independently sample a rotation uniformly in $[-90°, 90°]$, two translation coordinates uniformly in $[-0.4, 0.4]$ in normalized grid coordinates, and a scale uniformly in $[0.8, 1.2]$. The translation range corresponds to 20% of the grid side. Both feature resampling and inverse assignment resampling use bilinear interpolation, zero padding, and

`align_corners=False`. The shared head draws fresh Gumbel noise on the resampled features. Only the K part rows enter the cosine comparison, and the original foreground probability weights both maps without detachment. Gradients flow through both assignment branches. The configuration used in our experiments does not apply a valid-region mask or slot-mass weighting. Resampling and inverse resampling are not exact inverses at the discrete grid level, especially near padded boundaries.

Numerical implementation. Prototype pooling uses $\bar{A}_{s,n} = A_{s,n}/(\sum_{n'} A_{s,n'} + 10^{-8})$. Reconstruction is implemented as the mean squared residual over tokens and feature dimensions divided by the mean per-dimension batch variance, clamped below at 10^{-8}, with population variance (`unbiased=False`). Away from this clamp, it equals Eq. 2. The foreground confidence denominator, slot-mass denominator, conditional-probability denominators, and probabilities inside the commitment logarithm are clamped below at 10^{-6}. Correspondence renormalizes the sampled part rows by their own sum. Sinkhorn uses 10^{-9} for its probability and normalization safeguards. The spatial-weight normalization denominator is clamped below at 10^{-6}. Cosine similarity uses the implementation's 10^{-8} norm safeguard.

B.5 Inference

At inference, the Gumbel noise is omitted. The head is applied to the tokens of the unaugmented image. The $K + 1$ logit maps are upsampled bilinearly to the image resolution and the label of a pixel is the argmax; on keypoint benchmarks the label is read at the keypoint. Interpolating the logits of a linear head is the same as interpolating the features and applying the head, provided the same interpolation weights are used. Each image is labeled on its own; the cross-image quantities of Section 3.2 are used only in training.

B.6 Sources of comparison results

The CUB and 224-pixel PartImageNet-OOD results for DINO, PDiscoNet + ViT-B, and PDiscoFormer in Table 2 are taken from Table 1 of Aniraj et al. (2024). Their NABirds results are taken from the authors' model zoo.[2] MPAE's CUB and 224-pixel PartImageNet-OOD results are taken from Tables 2 and 1 of Xia et al. (2025), respectively. These values retain the evaluation protocols of their sources.

MPAE on NABirds is our reproduction at 448 pixels. At each K, the table reports the run with the highest evaluation NMI among four reproductions, with NME and ARI from the same run. Each run uses its final checkpoint. Images are resized to a square, and part maps are bilinearly upsampled before labels are read at visible keypoints.

The 448-pixel PartImageNet-OOD rows marked ‡ use our retrained models and their final checkpoints. Both methods use whole-image square resizing and are scored against the same 224×224 annotation grid as LinPart. Part maps are bilinearly upsampled to 448×448, converted to labels, and sampled at every second pixel for evaluation.

B.7 Head capacity experiments

Table 4 uses $K = 16$, 11, and 50 on CUB, NABirds, and PartImageNet-OOD, respectively. For each benchmark and seed, an independently trained linear control and five randomly initialized MLPs share the augmented images, frozen features, objective, and ten-epoch schedule. The MLPs have one hidden layer of 256, 512, or 1024 units, or two or three hidden layers of 1024 units each, with GELU activations and no hidden-layer normalization. On the first training batch, each MLP's learning rate is calibrated against the linear control's root-mean-square change in centered logits after one AdamW update. The weights and optimizer state are reset before training. The linear controls are separate from the main-result models. Seeds 1–3 and final checkpoints are used for all heads, including failed runs.

Tables 11–13 report seeds 1–3 for every head in Table 4 under square-frame evaluation. Means and sample standard deviations are computed over all three runs. For the CUB MLP with three 1024-unit

[2] `https://github.com/ananthu-aniraj/pdiscoformer/blob/main/model_zoo.md`

hidden layers, seeds 2 and 3 have zero NMI and ARI, and the final training diagnostics record no tokens assigned to foreground. These runs remain included in the mean.

Table 11: Head capacity on CUB ($K = 16$), by seed.

Head	Seed 1		Seed 2		Seed 3		Mean ± s.d.	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
MLP (256)	61.77	35.42	66.22	42.44	77.90	65.42	68.63 ± 8.33	47.76 ± 15.69
MLP (512)	60.44	34.09	63.05	37.10	63.05	38.62	62.18 ± 1.51	36.60 ± 2.31
MLP (1024)	52.87	25.87	72.17	54.50	61.64	37.61	62.22 ± 9.66	39.32 ± 14.39
MLP (1024×2)	76.79	62.74	76.78	63.48	66.23	47.36	73.27 ± 6.10	57.86 ± 9.10
MLP (1024×3)	70.10	50.84	0.00	0.00	0.00	0.00	23.37 ± 40.47	16.95 ± 29.35

Table 12: Head capacity on NABirds ($K = 11$), by seed.

Head	Seed 1		Seed 2		Seed 3		Mean ± s.d.	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
MLP (256)	72.31	61.66	58.34	38.20	59.04	38.91	63.23 ± 7.87	46.26 ± 13.35
MLP (512)	70.39	61.82	61.92	44.77	70.41	58.84	67.57 ± 4.90	55.14 ± 9.11
MLP (1024)	70.70	59.55	54.10	33.21	56.39	35.41	60.39 ± 8.99	42.72 ± 14.61
MLP (1024×2)	65.39	51.32	51.51	33.35	68.68	49.33	61.86 ± 9.11	44.66 ± 9.85
MLP (1024×3)	66.41	50.31	52.82	30.07	44.46	24.67	54.56 ± 11.08	35.02 ± 13.51

Table 13: Head capacity on PartImageNet-OOD ($K = 50$), by seed.

Head	Seed 1		Seed 2		Seed 3		Mean ± s.d.	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
MLP (256)	63.96	81.83	64.09	81.79	66.41	81.97	64.82 ± 1.38	81.86 ± 0.09
MLP (512)	67.28	83.95	67.75	84.83	64.65	81.75	66.56 ± 1.67	83.51 ± 1.58
MLP (1024)	66.67	82.24	64.12	83.04	64.94	83.03	65.24 ± 1.30	82.77 ± 0.46
MLP (1024×2)	62.45	81.44	62.35	81.55	66.45	84.01	63.75 ± 2.34	82.33 ± 1.45
MLP (1024×3)	62.05	79.87	61.55	80.62	58.68	79.80	60.76 ± 1.82	80.10 ± 0.45

C MAIN RESULTS BY SEED

Tables 14–16 report the LinPart results for seeds 1–3 underlying Table 2, together with their means and sample standard deviations.

Table 14: LinPart results by seed on CUB (518×518).

Seed	$K = 4$			$K = 8$			$K = 16$		
	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑
1	8.18	58.99	25.34	6.33	70.24	42.71	5.63	79.82	68.93
2	9.70	57.33	31.19	6.18	70.46	49.63	5.59	79.78	68.84
3	7.96	59.66	25.46	5.91	73.75	45.81	5.63	80.31	70.35
Mean	8.61	58.66	27.33	6.14	71.49	46.05	5.62	79.97	69.37
s.d.	0.95	1.20	3.34	0.21	1.96	3.46	0.02	0.30	0.85

C.1 COMPUTATIONAL-COST MEASUREMENT

Table 3 uses one NVIDIA RTX PRO 6000 Blackwell Max-Q Workstation Edition GPU, 448×448 inputs, $K = 50$, and batch size 64. All methods use FP32 with TF32 disabled and their original training loops and losses. After ten warmup updates, 30 consecutive updates in the first epoch are

Table 15: LinPart results by seed on NABirds (518×518).

Seed	$K = 4$			$K = 8$			$K = 11$		
	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑
1	10.24	57.42	32.58	8.40	67.00	45.67	7.47	73.70	64.86
2	10.29	49.14	25.86	8.13	70.43	50.11	7.33	77.81	72.41
3	10.38	58.29	33.04	8.08	66.67	47.85	7.45	79.40	74.73
Mean	10.30	54.95	30.49	8.20	68.04	47.88	7.42	76.97	70.66
s.d.	0.07	5.05	4.02	0.17	2.08	2.22	0.08	2.94	5.16

Table 16: LinPart results by seed on PartImageNet-OOD.

Res.	Seed	$K = 8$		$K = 25$		$K = 50$	
		NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑
224	1	56.56	71.50	58.09	71.64	56.97	70.72
	2	58.04	71.03	61.90	73.06	58.62	72.48
	3	59.75	72.09	59.23	73.61	58.54	71.92
	Mean	58.12	71.54	59.74	72.77	58.04	71.71
	s.d.	1.59	0.53	1.96	1.01	0.93	0.90
448	1	57.20	80.48	62.22	79.87	62.31	80.06
	2	59.29	79.64	62.53	80.84	63.96	81.24
	3	58.89	80.92	64.29	82.10	66.14	82.16
	Mean	58.46	80.35	63.01	80.94	64.14	81.16
	s.d.	1.11	0.65	1.12	1.12	1.92	1.05

timed with CUDA synchronization before and after each iteration. Timings include host-to-device transfer, GPU augmentation, live backbone computation, all losses, backward computation, and optimizer updates. They exclude CPU data loading and augmentation, initialization, evaluation, and checkpoint writing. Peak memory is the maximum allocated PyTorch CUDA memory during these updates, reported in GiB. The runs measure training-step cost, not complete training time. Epoch counts are the original training schedules. Trainable parameter counts include resolution-dependent positional parameters in the compared models and exclude frozen parameters.

D Coarsening details

D.1 Training-image statistics

Coarsening uses 24 fixed batches of 32 training images on CUB and NABirds, and 24 batches of 64 on PartImageNet-OOD. Features are extracted in the unaugmented center-crop frame. In each batch, up to $32B$ tokens are sampled uniformly without replacement from those with foreground probability above 0.5 and a foreground argmax label. Each token connects to its five nearest sampled tokens from other images under feature cosine similarity. The sample, graph, features, and foreground remain fixed during merging. Scores are computed within each batch and then averaged.

D.2 Merge-score calculation

For hard assignments on a fixed sample and graph, uniform negative sampling with replacement gives

$$\mathbb{E}_{-}[L_{\mathrm{corr}}] = 1 - Q_u, \qquad Q_u = \sum_k (e_{kk} - p_k^2), \tag{10}$$

where e_{kk} is the fraction of positive edges internal to part k and p_k its share of sampled tokens. Positive agreement is $\sum_k e_{kk}$, while expected negative agreement is $\sum_k p_k^2$, including self-pairs. This gives a modularity score with a uniform chance baseline (Newman, 2006; Reichardt & Bornholdt, 2006). Merging r and s increases Q_u by $e_{rs} - 2p_r p_s$. The product $p_r p_s$ is computed before averaging over batches.

Let n_{br} and $\boldsymbol{\mu}_{br}$ denote the count and mean feature of part r in image b. The within-between sum-of-squares identity gives the normalized reconstruction cost

$$B_{rs} = \frac{1}{TDv} \sum_{b:\, n_{br} n_{bs} > 0} \frac{n_{br} n_{bs}}{n_{br} + n_{bs}} \|\boldsymbol{\mu}_{br} - \boldsymbol{\mu}_{bs}\|^2, \tag{11}$$

where T is the number of reconstructed tokens in the batch, $D = 768$, and v is the mean per-dimension population variance, clamped below at 10^{-8}. Counts and feature sums are updated after each merge; normalization remains fixed.

For warp consistency, original and transformed-view predictions are converted to one-hot assignments before bilinear inverse resampling. The original hard foreground mask weights both maps. One affine transformation per image is drawn from the training ranges and reused for all candidates. Gumbel noise is omitted. Each candidate sums the two part maps in both views, then recomputes the cosine loss at the new part count. Correspondence, reconstruction, and consistency retain their training coefficients. Merging continues to the specified target even when all remaining scores are nonpositive. Ties after rounding scores to 12 decimal places are resolved by original slot indices. This criterion evaluates a hard-assignment partial objective, not the complete stochastic training loss.

D.3 Comparison of merge criteria

The four merge orders in Table 6 share the same source heads, training-image batches, foreground predictions, and target part counts. Cosine merges the pair with the highest cosine similarity between their mean feature vectors, pooled over all tokens assigned to each part in the training batches. After a merge, the mean is updated by weighting each constituent part by its token count. Reconstruction only merges the pair with the smallest normalized per-image reconstruction cost B_{rs}. Correspondence plus reconstruction uses Eq. 9 without the consistency contribution. The final score includes all three contributions with their training coefficients. All orders follow the same greedy procedure, update their statistics after each merge, preserve background predictions, and continue until the target count is reached. Merge mappings are fixed without annotations before evaluation.

Results are means over seeds 1–3 under the same square-frame evaluation. Reconstruction only has higher means on CUB at four parts. On PartImageNet-OOD at 25 parts, cosine gives the highest mean NMI and reconstruction only gives the highest mean ARI. At eight parts, correspondence plus reconstruction gives the highest means on both metrics. These comparisons describe the existing checkpoint cohort and do not establish a uniformly superior merge order.

Table 17: Coarsening results on CUB. Individual seeds and mean ± s.d.

	Seed 1		Seed 2		Seed 3		Mean ± s.d.	
Criterion	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
Source $K = 16$, target $K = 8$								
Cosine	71.78	42.82	69.79	37.12	70.42	43.05	70.66 ± 1.02	41.00 ± 3.36
Rec. only	67.04	32.34	72.51	44.34	70.08	38.03	69.88 ± 2.74	38.24 ± 6.01
Corr. + rec.	68.31	37.56	71.76	44.05	71.55	48.65	70.54 ± 1.93	43.42 ± 5.57
Final score	75.09	51.97	74.01	53.32	74.14	54.81	74.41 ± 0.59	53.37 ± 1.42
Direct training	70.24	42.71	70.46	49.63	73.75	45.81	71.49 ± 1.96	46.05 ± 3.46
Source $K = 16$, target $K = 4$								
Cosine	55.86	22.45	50.15	19.20	35.93	7.35	47.31 ± 10.26	16.33 ± 7.95
Rec. only	55.61	22.45	56.50	22.97	55.52	24.58	55.88 ± 0.54	23.33 ± 1.11
Corr. + rec.	57.70	24.80	58.36	25.03	49.16	19.76	55.07 ± 5.13	23.20 ± 2.98
Final score	51.08	16.82	58.36	25.03	54.96	22.57	54.80 ± 3.65	21.47 ± 4.21
Direct training	58.99	25.34	57.33	31.19	59.66	25.46	58.66 ± 1.20	27.33 ± 3.34

Table 18: Coarsening results on NABirds. Individual seeds and mean ± s.d.

	Seed 1		Seed 2		Seed 3		Mean ± s.d.	
Criterion	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
Source $K = 11$, target $K = 8$								
Cosine	65.12	42.39	72.33	58.62	69.65	48.48	69.03 ± 3.64	49.83 ± 8.20
Rec. only	69.75	53.37	70.11	49.42	71.13	49.92	70.33 ± 0.71	50.90 ± 2.15
Corr. + rec.	65.12	42.39	70.11	49.42	69.65	48.48	68.29 ± 2.75	46.76 ± 3.82
Final score	71.10	56.45	70.17	50.03	74.10	59.59	71.79 ± 2.05	55.36 ± 4.88
Direct training	67.00	45.67	70.43	50.11	66.67	47.85	68.04 ± 2.08	47.88 ± 2.22
Source $K = 11$, target $K = 4$								
Cosine	24.46	4.92	39.81	10.33	23.37	4.56	29.22 ± 9.19	6.60 ± 3.23
Rec. only	56.61	30.99	59.10	33.02	60.00	33.37	58.57 ± 1.75	32.46 ± 1.28
Corr. + rec.	51.99	26.98	61.70	40.16	43.43	16.00	52.37 ± 9.14	27.71 ± 12.09
Final score	59.83	36.51	61.73	40.42	60.00	33.37	60.52 ± 1.05	36.76 ± 3.53
Direct training	57.42	32.58	49.14	25.86	58.29	33.04	54.95 ± 5.05	30.49 ± 4.02

Table 19: Coarsening results on PartImageNet-OOD. Individual seeds and mean ± s.d.

	Seed 1		Seed 2		Seed 3		Mean ± s.d.	
Criterion	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
Source $K = 50$, target $K = 25$								
Cosine	62.56	79.87	63.90	81.24	66.14	82.16	64.20 ± 1.81	81.09 ± 1.16
Rec. only	62.24	80.06	63.68	81.23	66.05	82.16	63.99 ± 1.93	81.15 ± 1.05
Corr. + rec.	61.78	79.94	63.16	81.12	65.80	82.06	63.58 ± 2.04	81.04 ± 1.06
Final score	61.61	79.98	62.97	81.13	65.34	82.06	63.31 ± 1.89	81.05 ± 1.04
Direct training	62.22	79.87	62.53	80.84	64.29	82.10	63.01 ± 1.12	80.94 ± 1.12
Source $K = 50$, target $K = 8$								
Cosine	52.16	71.02	51.90	71.50	51.99	72.57	52.02 ± 0.13	71.70 ± 0.79
Rec. only	54.82	77.86	58.04	80.33	58.05	79.88	56.97 ± 1.86	79.36 ± 1.32
Corr. + rec.	60.93	79.64	61.68	80.80	64.56	81.88	62.39 ± 1.91	80.77 ± 1.12
Final score	60.69	79.68	59.91	80.72	63.05	81.77	61.22 ± 1.63	80.72 ± 1.05
Direct training	57.20	80.48	59.29	79.64	58.89	80.92	58.46 ± 1.11	80.35 ± 0.65