Which Tokens to Use? Investigating Token Reduction in Vision Transformers Supplementary Materials

Joakim Bruslund Haurum[1] Sergio Escalera[2,1] Graham W. Taylor[3] Thomas B. Moeslund[1]

[1] Visual Analysis and Perception (VAP) Laboratory, Aalborg University & Pioneer Centre for AI, Denmark
[2] Universitat de Barcelona & Computer Vision Center, Spain [3] University of Guelph & Vector Institute for AI, Canada

joha@create.aau.dk, sescalera@ub.edu, gwtaylor@uoguelph.ca, tbm@create.aau.dk

A. Overview of Supplementary Materials

In these supplementary materials we describe in further detail aspects of the training process, performance of the tested models, per-dataset results of the in-depth analysis, and visual examples of the reduction patterns. We refer to both the relevant sections in the main manuscript as well as sections in the supplementary materials. Specifically the following will be described:

- Hyperparameters for the four classification datasets (Section 3.3 / B).
- Performance of token reduction methods using the DeiT-T and DeiT-B backbones (Section 4 / C).
- Per-dataset analysis of the dynamic keep rate in the ATS method (Section 4 / D).
- Description of the pattern reduction similarity measures (Section 5.1–5.2 / E).
- Description of the lower bound IoA and IoU computation (Section 5.1–5.2 / F).
- Per-dataset results for varying the keep rate r and backbone capacity (Section 5.1–5.2 / G).
- Additional saliency metrics for the cross-dataset reduction pattern comparison (Section 5.3 / H).
- Results of the ℓ_p reduction pattern comparison (Section 5.4 / I).
- Results using CKA and PWCCA as proxies of model performance (Section 5.5 / J).
- Per-dataset scatter plots of the proxy measures and model performance (Section 5.5 / K).
- Per-dataset visualization of the averaged reduction patterns (Section 5.3 / L).
- Per-dataset example visualization of the reduction patterns (Section 5 / M).

Table 1: **Hyperparameter grid search.** We conduct a gird search over a subset of the hyperparameters. For ImageNet the search is conducted over the token reduction methods (restricted to the warmup epochs, backbone scale, and how many the backbone weights are frozen), whereas for NABirds, COCO, and NUS-WIDE it is conducted for the DeiT-S baseline. We note that for the ImageNet dataset we restrict the backbone LR scale factor to only 1 or 0.01, following Rao *et al.* [17].

Hyperparameter	Grid Values
Learning Rate (LR)	$[0.01, 0.001, 0.0001]$
LR Normalization Factor (LR-Norm)	$[512, 1024]$
Warmup Epochs (W-E)	$[5, 20]$
Backbone LR Scale (B-LR)	$[1, 0.1, 0.01]$
Backbone Freeze Epochs (B-FE)	$[0, 5]$

B. Hyperparameters

In this section we further elaborate on the training details in Section 3.3 and describe the hyperparameters used during training in detail. The hyperparameters can be split into two groups: 1) the static hyperparameters per dataset and 2) the hyperparameters which we conducted a search on per method. The static hyperparameters were selected based on what have been used in prior methods applied on each dataset [9, 13, 17, 18], as well as training guidelines from the DeiT paper [21]. For all datasets we used the AdamW optimizer [15] with a momentum of 0.9 weight decay of 0.05, a Cosine learning rate schedule [14] with a decay rate of 0.1, and stochastic depth of 0.1 [11]. We train all methods on 2 V100 GPUs with mixed precision, repeated augmentations (x3) [2, 10], and gradient accumulation if the batch cannot fit onto the GPUs. For the K-Medoids and Sinkhorn methods we perform three iterations for the clustering, set the entropy regularization ϵ in the Sinkhorn method to 1, and set the number of neighbours $k = 5$ for the DPC-KNN method. The remaining static hyperparameters are shown in Table 2.

Table 2: **Dataset-specific hyperparameters.** We fix a large set of the hyperparameters based on prior work. For ImageNet we are inspired by the DynamicViT and DeiT papers [17, 21], NABirds is based on the hyperparameters used in the TransFG work [9], and COCO and NUS-WIDE are based on the hyperparameters from the ASL work [18].

Dataset	ImageNet	NABirds	COCO	NUS-WIDE
Epochs	30	50	40	40
Batch size	1024	1024	512	512
Loss	Cross-Entropy	Cross-Entropy	ASL [18]	ASL [18]
Label Smoothing	0.1	0	0	0
ASL γ_-	-	-	4	4
ASL Clip	-	-	0.05	0.05
Model EMA	0.9999	0.9999	0.9997	0.9997
Augmentations	Random Resize and Crop Horizontal Flip (50%) RandAugment [4] Normalization Random Erasing (25%) [24] Mixup/CutMix [22, 23]	Random Resize and Crop Horizontal Flip (50%) Normalization	Resize Cutout (50%) [5] RandAugment [4] Normalization	Resize Cutout (50%) [5] RandAugment [4] Normalization

Table 3: **Selected token reduction method hyperparameters - ImageNet.** We present the selected hyperparameters when searching on ImageNet for each token reduction method.

r (%)	25			50			70			90		
	W-E	B-LR	B-FE	W-E	B-LR	B-FE	W-E	B-LR	B-FE	W-E	B-LR	B-FE
ℓ_1	5	1	0	5	0.01	5	5	0.01	5	5	0.01	5
ℓ_2	5	1	0	5	0.01	5	5	0.01	5	5	0.01	5
ℓ_∞	5	1	0	5	0.01	5	5	0.01	5	5	0.01	5
Top-K	5	1	0	5	0.01	5	5	0.01	5	20	1	0
EViT	5	1	0	5	0.01	5	5	0.01	5	20	1	0
DynamicViT	20	0.01	5	5	0.01	5	20	0.01	5	20	0.01	5
ATS	5	1	0	5	0.01	5	20	0.01	5	5	0.01	5
ToMe	-	-	-	5	0.01	5	5	0.01	5	5	0.01	5
K-Medoids	5	1	0	5	0.01	5	20	0.01	5	20	1	0
DPC-KNN	5	0.01	5	20	0.01	5	5	0.01	5	5	0.01	5
SiT	5	0.01	5	5	0.01	5	5	0.01	5	20	0.01	5
PatchMerger	5	0.01	5	5	0.01	5	5	0.01	5	5	0.01	5
Sinkhorn	5	1	0	5	1	0	5	1	0	5	1	0

Table 4: **Selected DeiT baseline hyperparameters.** We present the selected hyperparameters for the DeiT baselines on NABirds, COCO, and NUS-WIDE.

Dataset	LR	LR-Norm	W-E	B-LR	B-FE
NABirds	0.001	1024	5	0.1	5
COCO	0.0001	512	5	1	0
NUS-WIDE	0.0001	512	5	1	0

For a subset of the hyperparameters we perform a grid search per token reduction method with the DeiT-S backbone on the ImageNet dataset, and for the DeiT-S baseline on the NABirds, COCO, and NUS-WIDE datasets. The grid searched hyperparameters are: the learning rate, the number of warmup epochs in the cosine scheduler, the number of epochs where the backbone weights should be fixed, the backbone weights learning rate scaling factor, and a normalization factor of the learning rate [8]. The hyperparameter value ranges are shown in Table 1. On ImageNet we fix the learning rate to 0.001 and the normaliza-

Table 5: **Hyperparameter indicator matrix.** We illustrate below for each method and keep rate r whether the hyperparameter settings from the ImageNet dataset ($\mathcal{I}$) or the dataset specific DeiT-S baseline ($\mathcal{D}$) are used.

	NABirds				COCO				NUS-WIDE			
r (%)	25	50	70	90	25	50	70	90	25	50	70	90
ℓ_1	$\mathcal{I}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
ℓ_2	$\mathcal{I}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
ℓ_∞	$\mathcal{I}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
Top-K	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
EViT	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
DynamicViT	$\mathcal{I}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{I}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
ATS	$\mathcal{I}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
ToMe	-	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	-	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	-	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
K-Medoids	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
DPC-KNN	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
SiT	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$
PatchMerger	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$	$\mathcal{D}$
Sinkhorn	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$	$\mathcal{I}$

tion factor to 1024 (*i.e*. the batch size). On the NABirds, COCO, and NUS-WIDE datasets we determine the final per-method hyperparameters by comparing models trained with the method-specific hyperparameters obtained on ImageNet and the dataset-specific DeiT-S baseline hyperparameters. The best hyperparameters per token reduction method and keep rate r on the ImageNet dataset is shown in Table 3. For NABirds, COCO, and NUS-WIDE we show the best hyperparameters for the DeiT-S baseline in Table 4, and an indicator matrix in Table 5 indicating whether the dataset fine-tuned DeiT-S hyperparameters or the ImageNet hyperparameters are used per token reduction method and r.

C. Token Reduction Performance using DeiT-T and DeiT-B Backbones

In this section we present the results with the DeiT-T and DeiT-B baselines as mentioned in Section 4; see Table 6.

D. Analysis of ATS Keep Rates

As discussed in Section 3.1.3, the ATS [7] method is a dynamic keep rate pruning method, and therefore the meaning of the keep rate r makes a subtle but important change. Instead of being the ratio of kept tokens, it instead represents the upper bound of the ratio of tokens to be kept, which the ATS method cannot exceed. In order to better understand the ATS method we plot the per-dataset average keep rate at each ViT stage for the different values of r; see Figure 1. We observe that when 90% and 70% of the tokens may be kept the actual keep rate is much lower, especially during the later stages of the ViT.

E. Reduction Pattern Similarity Metrics

In this section we describe in more detail the metrics used to compare reduction patterns in Sections 5.1–5.2. For the Intersection over Area (IoA) and Intersection over Union (IoU) metrics used to compare pruning-based methods, each method produces a reduction pattern M with keep rate r, where M consists of the kept tokens after applying the reduction method. Using set notation the IoA and IoU can then be defined as in Equations 1-2.

$$\text{IoA} = \frac{M_1 \cap M_2}{M_2} \text{ s.t. } r_1 \geq r_2 \tag{1}$$

$$\text{IoU} = \frac{M_1 \cap M_s}{M_1 \cup M_2} \tag{2}$$

For clustering-based methods, we utilize two information theoretic metrics: Homogeneity [19] and Normalized Mutual Information (NMI) [20]. Homogeneity measures the class distribution within the constructed clusters, where the optimal value is obtained if all data points from the same class are assigned to the same cluster. This can be expressed using entropy as in Equations 3-5.

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases} \tag{3}$$

$$H(C|K) = -\sum_{k=1}^{|K|} \sum_{c=1}^{|C|} \frac{n_{c,k}}{N} \log \frac{n_{c,k}}{n_k} \tag{4}$$

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{N} \log \frac{n_c}{N} \tag{5}$$

Table 6: **Performance of Token Reduction methods.** Measured across varying keep rates, r, and backbone capacities. Scores exceeding the DeiT baseline are noted in **bold**, measured as Top-1 accuracy for ImageNet & NABirds and mean Average Precision for COCO and NUS-WIDE. The three best performing methods per keep rate are denoted in descending order with red , orange , and yellow , respectively. Similarly, the three worst performing methods are denoted in descending order with light blue , blue , and dark blue

(a) Performance comparison of token reduction methods trained with a DeiT-Base backbone.

	ImageNet				NABirds				COCO				NUS-WIDE			
DeiT-B	81.85				83.32				80.93				64.37			
r (%)	25	50	70	90	25	50	70	90	25	50	70	90	25	50	70	90
ℓ_1	71.23	74.96	78.94	81.04	59.79	71.57	78.92	82.42	58.28	69.27	76.23	79.65	53.01	60.10	63.25	64.14
ℓ_2	71.41	75.40	79.07	81.18	61.55	73.24	79.52	82.55	59.69	70.33	76.56	79.75	54.00	60.37	63.29	64.28
ℓ_∞	71.67	74.40	78.95	81.20	59.96	70.51	79.73	82.59	58.48	68.50	76.54	79.89	53.00	59.59	63.12	64.25
Top-K	73.63	78.97	80.91	**82.03**	74.71	82.22	83.20	**83.40**	67.63	76.91	79.95	**80.97**	58.51	62.78	63.92	**64.40**
EViT	75.26	79.22	80.99	**82.00**	74.73	82.00	83.19	**83.33**	68.93	76.92	79.87	80.92	59.00	62.88	63.90	**64.43**
DynamicViT	27.94	74.58	80.68	81.76	49.23	82.30	83.16	83.23	24.88	62.79	76.54	80.64	28.56	55.51	60.73	63.83
ATS	73.89	78.94	80.78	81.57	71.00	80.10	82.58	83.26	68.17	76.38	79.35	80.50	59.49	63.17	64.21	**64.48**
ToMe	-	78.89	81.05	**82.00**	-	73.67	81.59	82.98	-	74.11	78.82	80.48	-	62.38	64.06	64.35
K-Medoids	69.12	76.86	79.98	81.76	57.54	75.29	80.62	82.57	61.79	73.60	77.58	80.32	56.67	62.18	63.53	64.35
DPC-KNN	69.40	75.87	79.06	81.05	58.16	67.36	72.83	78.29	65.99	73.32	77.03	79.76	58.58	61.39	62.96	63.87
SiT	68.39	75.53	76.63	77.26	65.09	70.75	70.36	68.96	54.86	53.27	53.16	52.73	56.12	59.76	60.64	61.08
PatchMerger	58.78	70.63	74.52	76.76	40.38	57.21	62.20	67.06	54.25	66.22	70.97	73.72	51.80	58.83	60.79	62.09
Sinkhorn	63.37	63.33	63.36	63.50	42.89	42.33	41.72	42.86	52.57	52.33	52.21	52.12	47.55	47.41	47.26	47.48

(b) Performance comparison of token reduction methods trained with a DeiT-Tiny backbone.

	ImageNet				NABirds				COCO				NUS-WIDE			
DeiT-T	72.20				74.16				71.09				59.27			
r (%)	25	50	70	90	25	50	70	90	25	50	70	90	25	50	70	90
ℓ_1	58.58	62.27	67.91	71.06	51.82	59.25	68.36	73.47	49.09	58.27	67.03	70.24	44.81	52.64	57.30	58.73
ℓ_2	58.85	62.91	67.91	71.13	53.10	60.87	69.20	73.42	50.46	60.00	67.33	69.98	45.73	53.45	57.34	58.54
ℓ_∞	59.08	61.92	67.79	71.38	52.60	57.70	69.25	73.34	50.08	57.30	67.22	69.89	45.32	51.91	57.41	58.30
Top-K	62.19	68.55	70.96	71.85	62.14	73.19	**74.57**	**74.64**	60.31	67.47	70.20	**71.65**	52.20	57.02	58.60	**59.50**
EViT	64.11	68.69	71.06	71.83	64.13	73.24	**74.49**	**74.53**	61.44	67.62	70.25	**71.63**	53.09	57.26	58.64	**59.49**
DynamicViT	36.93	67.40	70.94	72.14	57.38	72.54	73.97	**74.30**	24.67	61.70	68.83	**71.30**	28.09	49.36	56.79	58.95
ATS	62.63	68.61	70.77	71.71	64.53	71.07	73.71	**74.43**	60.97	67.37	69.88	**71.10**	52.85	57.30	58.55	59.20
ToMe	-	69.72	71.74	72.16	-	66.61	73.65	**74.50**	-	65.66	69.70	**71.16**	-	55.32	57.78	58.98
K-Medoids	57.50	65.82	69.90	71.50	44.62	66.52	72.09	74.04	54.08	64.83	69.09	71.05	49.13	55.92	58.38	59.07
DPC-KNN	64.56	69.68	71.10	71.88	64.23	71.05	73.02	74.05	63.32	68.03	69.55	70.84	55.37	57.33	58.08	58.88
SiT	63.43	67.98	68.99	68.90	36.35	36.65	34.00	35.07	48.01	47.50	46.98	46.48	36.67	38.15	36.98	37.70
PatchMerger	60.38	64.80	66.81	68.09	38.83	54.20	59.94	62.60	52.49	59.69	62.63	64.30	47.56	52.69	54.33	55.37
Sinkhorn	53.61	53.49	53.19	53.51	36.94	35.98	37.29	36.19	50.47	49.52	49.12	49.01	45.77	44.81	44.52	44.20

where K is the set of generated clusters, C is the set of ground truth classes, $n_{c,k}$ is the number of data points from class c in cluster k, n_c is the number of data points in class c, n_k is the number of data points in cluster k, and N is the total number of data points.

In our analysis we define C to be the constructed clusters in M_1 and K to be the clusters in M_2, given $r_1 \geq r_2$. Thereby $|K| \leq |C|$, and the Homogeneity measures how well each cluster in M_1 maps to the reduced amount of clusters in M_2.

Similarly the NMI can be expressed using the Mutual Information (MI) between the constructed clusters in M_1 and M_2 (denoted as C and K to keep consistency with the Homogeneity notation), normalized by the averaged entropy of C and K, see Equations 6-7.

$$\text{NMI}(C, K) = \frac{I(C, K)}{(H(C) + H(K))/2} \tag{6}$$

$$I(C, K) = \frac{n_{c,k}}{N} \log \frac{N n_{c,k}}{n_c n_k} \tag{7}$$

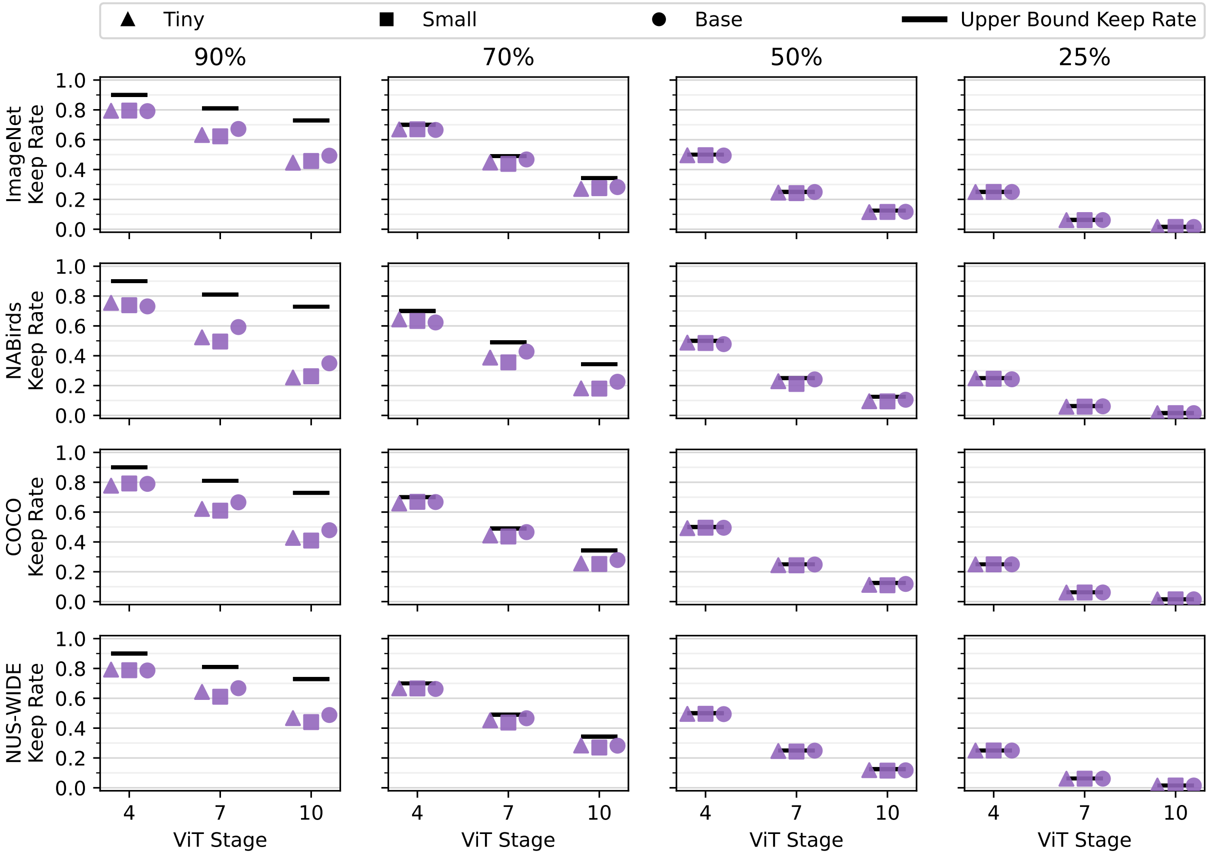


Figure 1: **Actual ATS keep rates (Section D).** The ATS method is a dynamic keep rate pruning method, meaning the amount of tokens kept at each reduction stage can be variable. This means the keep rate r is interpreted as an upper bound. We find that with an r value of 90% and 70% the actual keep rates are dramatically lower at the later reduction stages.

Furthermore, using the Homogeneity h, and the symmetric metric "Completeness", c, a combined metric called the "V-Measure" can be defined as the harmonic mean of h and c [19]. It has been shown by Becker [1] that the V-Measure is equivalent to the Normalized Mutual Information, when the arithmetic mean is used for normalizing the MI.

F. Lower Bound of IoA and IoU

When comparing pruning-based token reduction methods in Section 5.1–5.2 the keep rates, r_1 and r_2, may be selected such that a subset of tokens will be selected by both models. This can skew the interpretation of the IoA and IoU metrics, as the metrics may have high values but in fact only due to the inherently overlapping subset. In order to account for this we determine the minimum IoA and IoU for the reduction stage given r_1, r_2, and number of spatial tokens in the input image, P, using the Algorithms 1-2. These lower bounds are only true for pruning-based methods with a static keep rate and may therefore be broken by the ATS method.

It is not necessary to derive similar lower bounds for the clustering-based Homogeneity and NMI metrics, as both metrics can reach a value of 0. Homogeneity reaches 0 when the clustering provides no new information, *i.e.* when the class distribution in each cluster is equal to the overall class distribution [19]. Similarly, it can be inferred the same is true for the NMI, since NMI can be expressed in terms of Homogeneity and Completeness.

G. Per-Dataset Results when Varying r and backbone capacity

In this section, we extend the analysis conducted in Sections 5.1–5.2 by presenting per-dataset results when testing the consistency of reduction patterns under varying keep rate r and backbone capacity; see Figures 2-5. For all datasets we find that fixed rate pruning-based reduction patterns are consistent when varying r, but inconsistent when varying the backbone capacity. We also observe that the hard-merging methods have a high Homogeneity when

Algorithm 1 Lower bound of IoA

```
Input: $P$, $r_1$, $r_2$, s.t. $r_1 \geq r_2$
Output: $LB$
  $LB \leftarrow \varnothing$
  for $s \in \{1, 2, 3\}$ do
    $P_{s,r_1} \leftarrow \lfloor Pr_1^s \rfloor$
    $P_{s,r_2} \leftarrow \lfloor Pr_2^s \rfloor$
    $P_{s,r_1,r_2} \leftarrow P_{s,r_1} + P_{s,r_2}$
    if $P_{s,r_1,r_2} \geq P$ then
      $LB_s \leftarrow \frac{P_{s,r_1,r_2} - P}{P_{s,r_2}}$
    else
      $LB_s \leftarrow 0$
    end if
    $LB \leftarrow LB \cup LB_s$
  end for
```

Algorithm 2 Lower bound of IoU

```
Input: $P$, $r_1$, $r_2$
Output: $LB$
  $LB \leftarrow \varnothing$
  for $s \in \{1, 2, 3\}$ do
    $P_{s,r_1} \leftarrow \lfloor Pr_1^s \rfloor$
    $P_{s,r_2} \leftarrow \lfloor Pr_2^s \rfloor$
    $P_{s,r_1,r_2} \leftarrow P_{s,r_1} + P_{s,r_2}$
    if $P_{s,r_1,r_2} \geq P$ then
      $LB_s \leftarrow \frac{P_{s,r_1,r_2} - P}{P}$
    else
      $LB_s \leftarrow 0$
    end if
    $LB \leftarrow LB \cup LB_s$
  end for
```

varying r indicating the constructed clusters are very consistent, while DPC-KNN and K-Medoids have a low IoU indicating varying cluster centers, similar to the observations made in Section 5.1–5.2. We also observe the Homogeneity to be lower for soft-merging methods for all datasets. Similar to the observations made in Section 5.2, we found that the hard-merging method have consistent reduction patterns when varying the backbone as long as r is above 25% and 50% for PatchMerger, while the constructed clusters are inconsistent for the Sinkhorn and SiT methods. These findings match the findings made when analyzing the data aggregated across datasets in Sections 5.1–5.2.

H. Expanded Cross-Dataset Reduction Pattern Metric Suite

We extend our analysis of the cross-dataset pruning-based reduction patterns in Section 5.3, by reporting results when using additional metrics from the saliency domain [3]. Specifically, we report results using the Spearman's ranked correlation coefficient, Jensen-Shannon Divergence, Earth Mover's Distance, and histogram similarity; see Figure 6. We observe that for all metrics there is a high similarity between reduction patterns from different datasets. Specifically, we note that the results observed when using the Earth's Mover Distance are similar to results obtained with all other metrics. This is noteworthy, as the Earth Mover's Distance is the only metric which incorporates the spatial distance between the tokens, whereas the other metrics interpret the reduction patterns as 1D distributions.

I. Comparison of Pruning-based and ℓ_p Reduction Patterns

In this section, we present more detailed results of the comparison of learned reduction patterns and the ℓ_p reduction patterns in Section 5.4. We report the IoU between the different ℓ_p fixed pattern reduction methods and the learned pruning-based reduction methods as well as the DPC-KNN and K-Medoids cluster centers; see Figure 7. It is clear that all methods have a low IoU score across all reduction stages for all three ℓ_p methods, indicating that the learned reduction patterns are very different from the fixed image-centered radial patterns applied by the ℓ_p.

J. Extended feature alignment metric suite - CKA and PWCCA analysis

We extend the analysis of whether feature alignment is a good proxy for model performance conducted in Section 5.5, by considering the commonly used metrics: Centered Kernel Alignment (CKA) [12] and Projection-Corrected Canonical Correlation Analysis (PWCCA) [16]. We follow the procedure laid out by Ding *et al*. [6] and make pairwise comparisons between all methods to the three anchor methods: Top-K, K-Medoids, and the baseline DeiT. The results are presented in Figure 8, and we observe that the CKA and PWCCA metrics are as good proxies for model performance as the orthogonal Procrustes Distance, with no noticeable differences in the results.

K. Model Performance Proxies - Scatter Plots

As described in Section 5.5, we find that reduction pattern similarity and CLS token feature alignment are moderate-to-strong proxies of model performance. We present scatter plots comparing the metric difference between the anchor model and all other models against the orthogonal Procrustes distance, IoU, and NMI; see Figures 9-23. Note that for the sake of brevity the results from different keep rates are plotted in the same plot, but do report separate results per backbone model capacities.

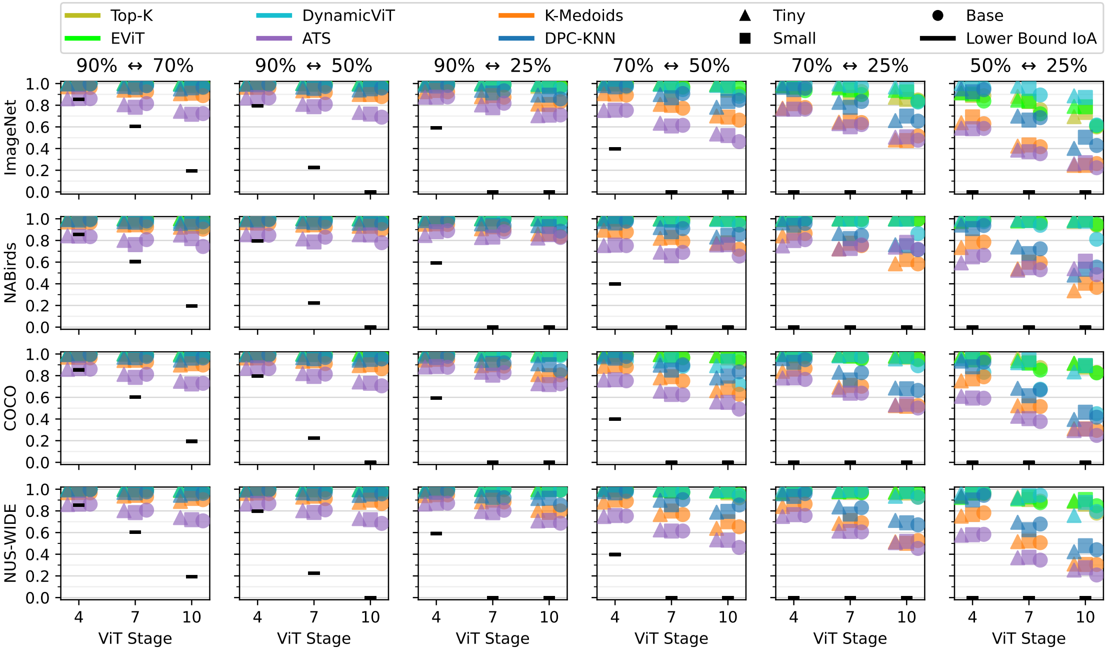


Figure 2: **Per-Dataset IoA results (Section G).** We observe that across all dataset the fixed rate pruning-based methods achieves high IoA scores across all keep rates r, indicating the reduction patterns are consistent. On the contrary, the ATS method and the cluster centers of the K-Medoids and DPC-KNN methods have low IoAs, indicating more inconsistent reduction patterns.

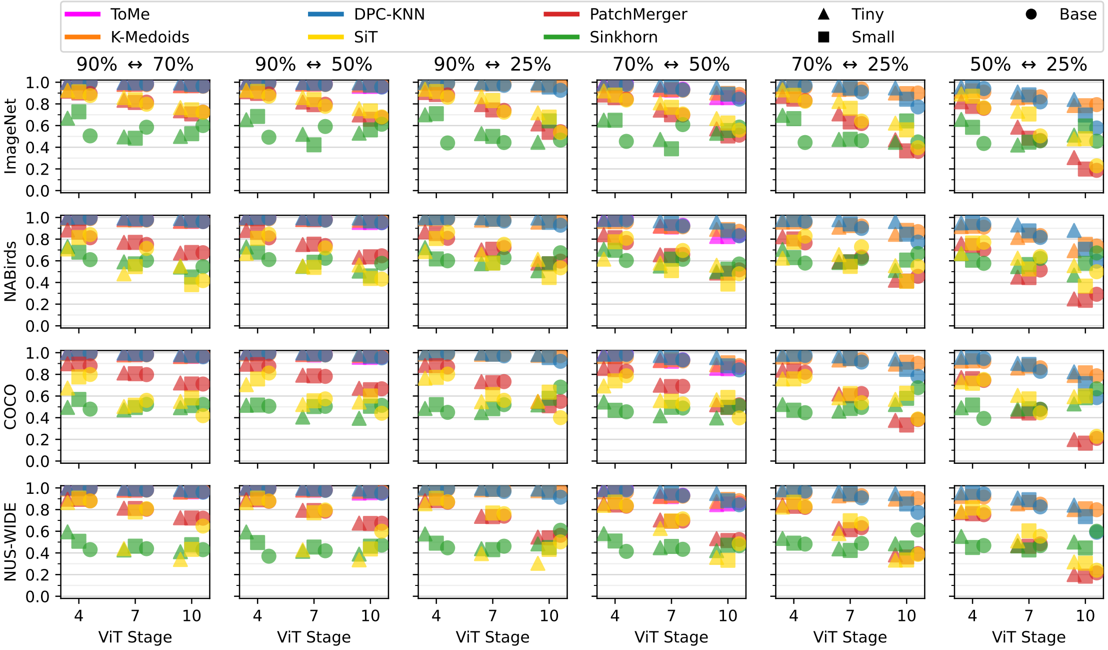


Figure 3: **Per-Dataset Homogeneity results (Section G).** We observe across all datasets that the hard-merging methods achieve a high Homogeneity score, indicating a high consistency of the constructed clusters when varying r. We also observe that soft-merging methods generally have lower Homogeneity scores, indicating less consistent clusters. We note that the PatchMerger and SiT methods have high scores at the earlier reduction stages, but that the scores reduces dramatically at later stages.

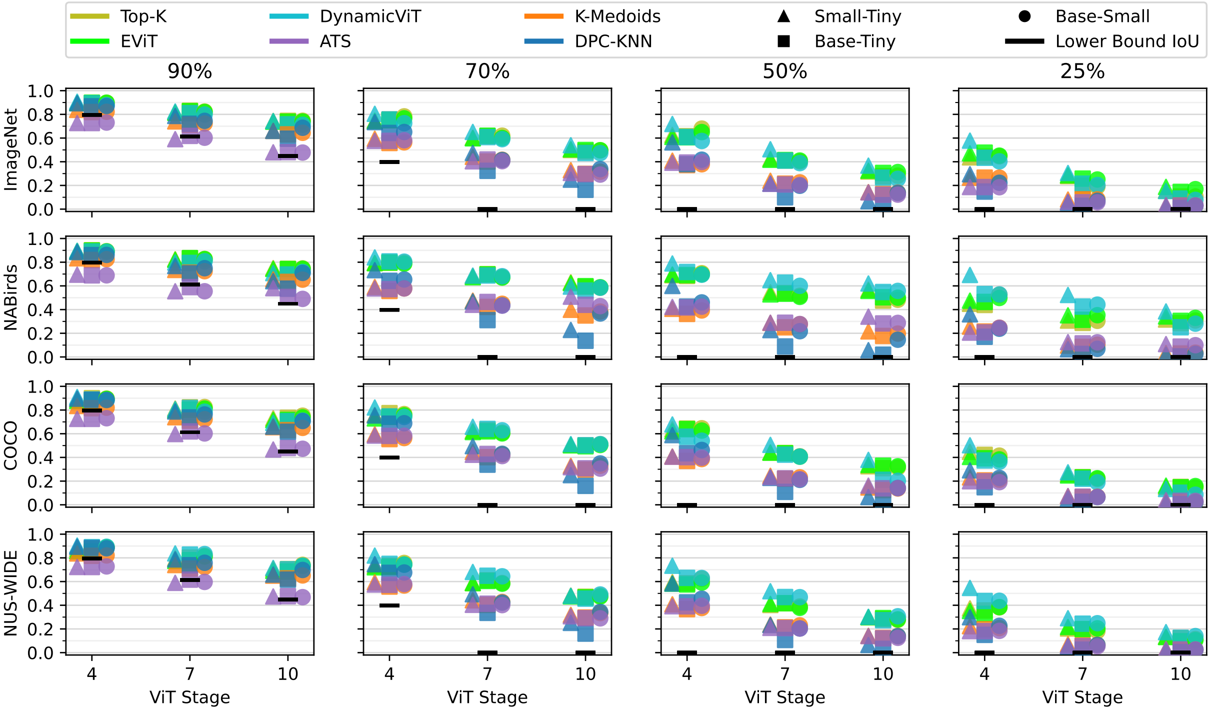


Figure 4: **Per-Dataset IoU results (Section G).** For all datasets we observe that the IoU score is very low for all tested methods. This indicates that the reduction patterns are not consistent when varying the backbone capacity.

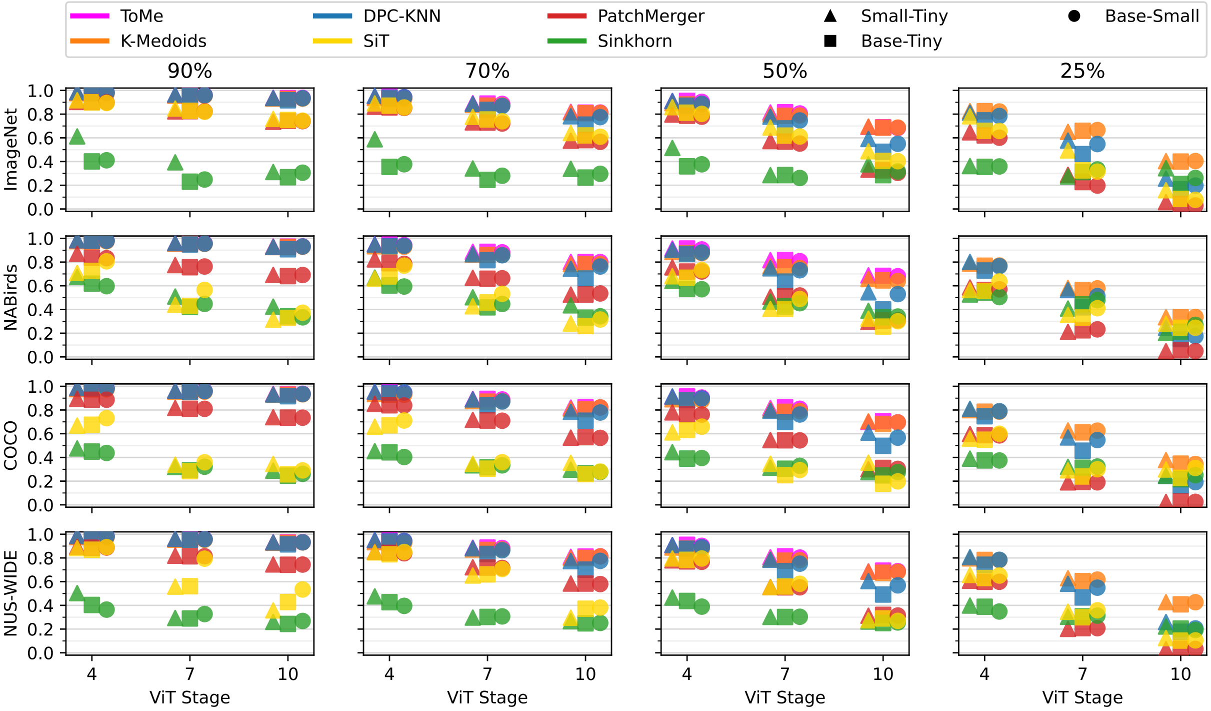


Figure 5: **Per-Dataset NMI results (Section G).** We observe that the hard-merging methods achieve high NMI scores when varying the backbone capacity as long as r is above 25%, where after it lower dramatically. We observe similar behaviour for the PatchMerger method as long as r is above 50%.

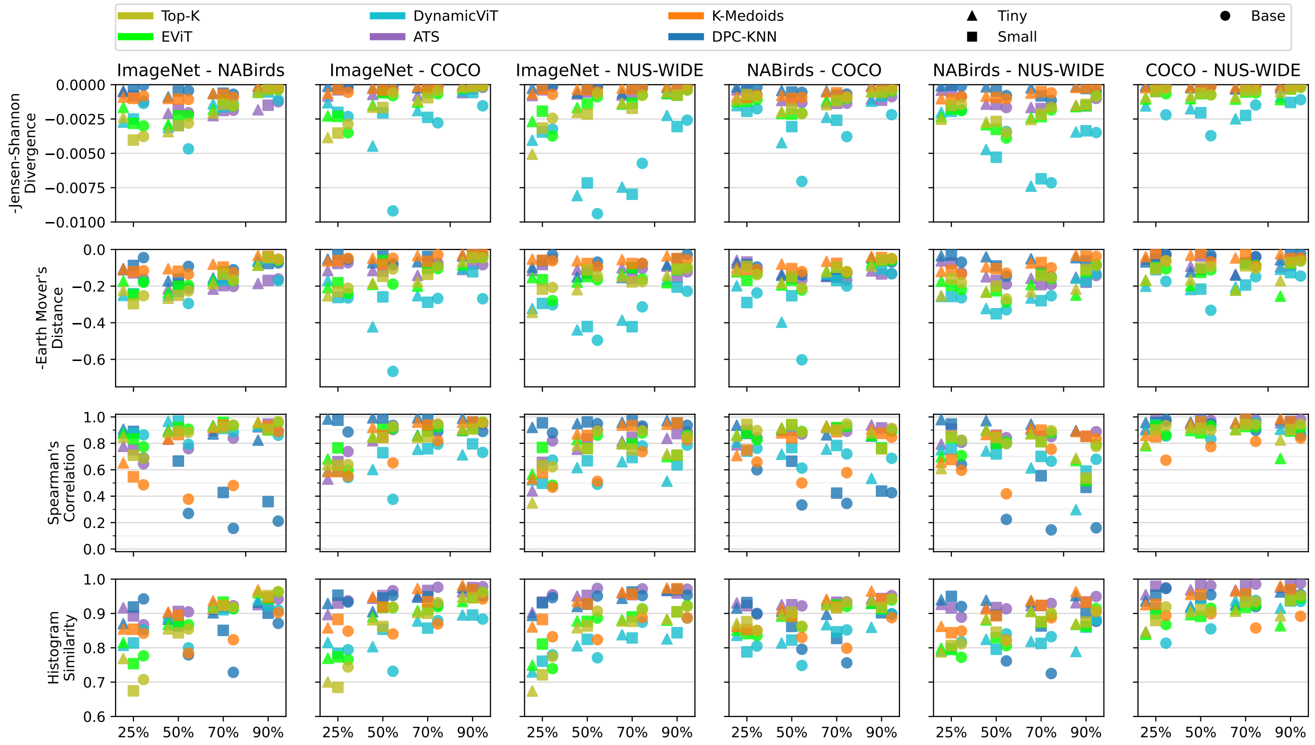


Figure 6: **Results with expanded averaged reduction pattern metric suite (Section H).** We compare with an expanded set of saliency metrics [3]. Across all metrics we observe high similarity when comparing the dataset-averaged reduction patterns.

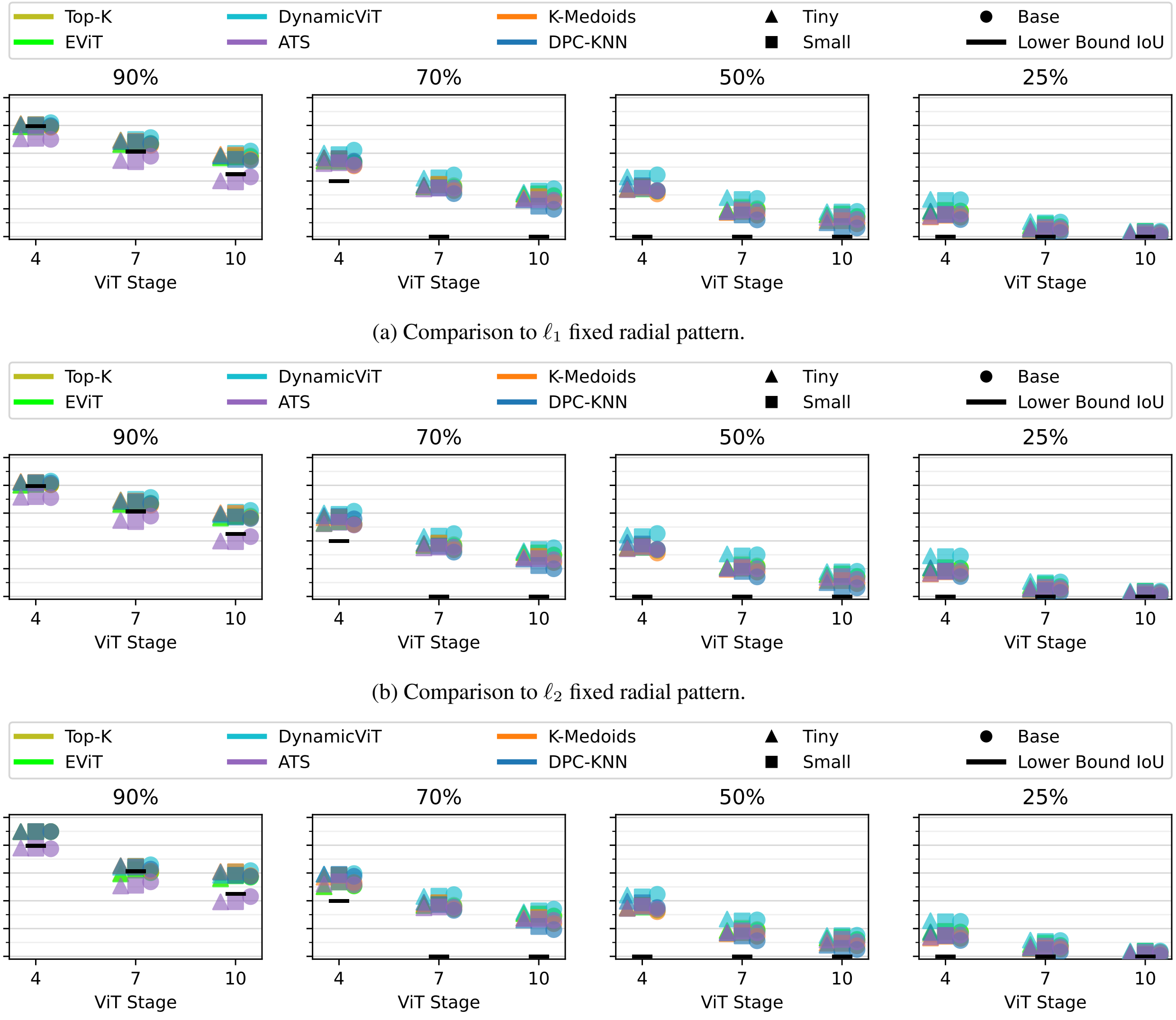
L. Visualization of Dataset Averaged Reduction Patterns

In this section, we present visualization of the dataset averaged reduction patterns used in Section 5.3 from the learned pruning-based methods as well as DPC-KNN and K-Medoids; see Figures 24-29. We observe that the reduction patterns of the Top-K and EViT methods are visually very similar, which is intuitive given both methods use the same pruning technique. Comparatively, we observe that the DynamicViT tends to more often select tokens closer to the image center, whereas Top-K and EViT instead select tokens along the border. This behavior is also exhibited by the ATS method, where we can also observe that the tokens in general have a lower average depth (*i.e.* the tokens are on average processed by fewer ViT layers) due to the dynamic keep rate nature of the ATS method. For the K-Medoids and DPC-KNN methods we see that the tokens are less centered on the image center. Instead the average depth of the tokens is much more uniform, corresponding to what we can visually observe (see Section M) as well as determined when investigating the reduction pattern consistency.

M. Per-Dataset Reduction Pattern Visualization of Randomly Selected Samples

As mentioned in Section 5, we present the reduction patterns obtained from a random sample of each dataset; see Figures 30-37. For brevity, we only show the reduction patterns obtained using the DeiT-S backbone. For the merging-based token reduction methods we observe that the hard-merging methods extract clusters which appear to be semantically coherent, whereas the soft-merging based methods more often extract clusters that are either incoherent or collapsed to a single cluster.

For the pruning-based methods we observe that the Top-K, EViT, and DynamicViT methods manage to focus in on the distinguishing features in a similar manner. Comparatively, the reduction patterns of the ATS method are distinctively different as they maintain a more global diversity of tokens, instead of larger coherent regions. Lastly, we observe the K-Medoids and DPC-KNN methods tend to not select tokens with distinctive features as cluster centers. This makes sense as these tokens may not be the best representative of a larger region.



(a) Comparison to ℓ_1 fixed radial pattern.

(b) Comparison to ℓ_2 fixed radial pattern.

(c) Comparison to ℓ_∞ fixed radial pattern.

Figure 7: **Comparing pruning-based reduction patterns to ℓ_p fixed reduction patterns (Section I).** We find that across all reduction stages the IoU between token reduction methods and the ℓ_p pruning baselines is very low. This indicates the learned reduction patterns are very different from the fixed radial patterns.

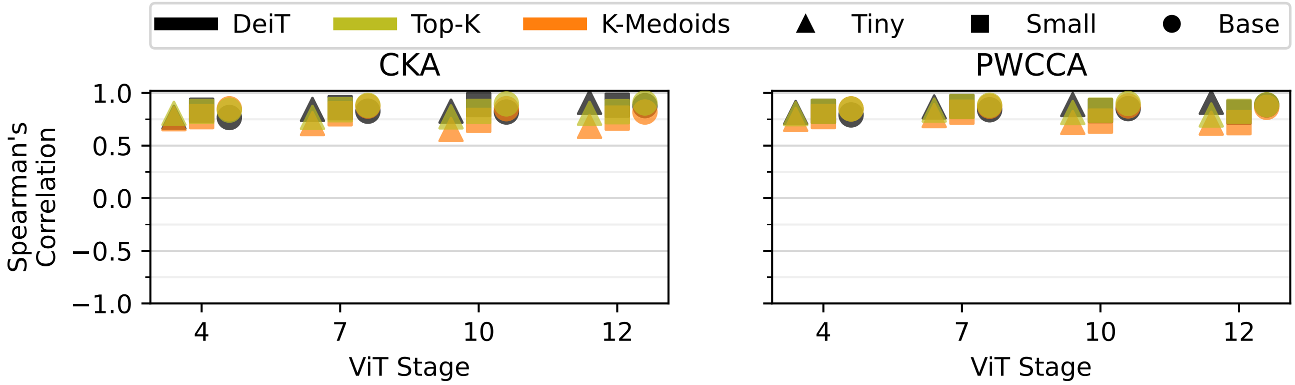


Figure 8: **Results with extended feature alignment metric suite (Section J).** We present results when comparing feature alignment using CKA and PWCCA with the difference in model performance. We find a high correlation, similar to what was observed using the orthogonal Procrustes distance.

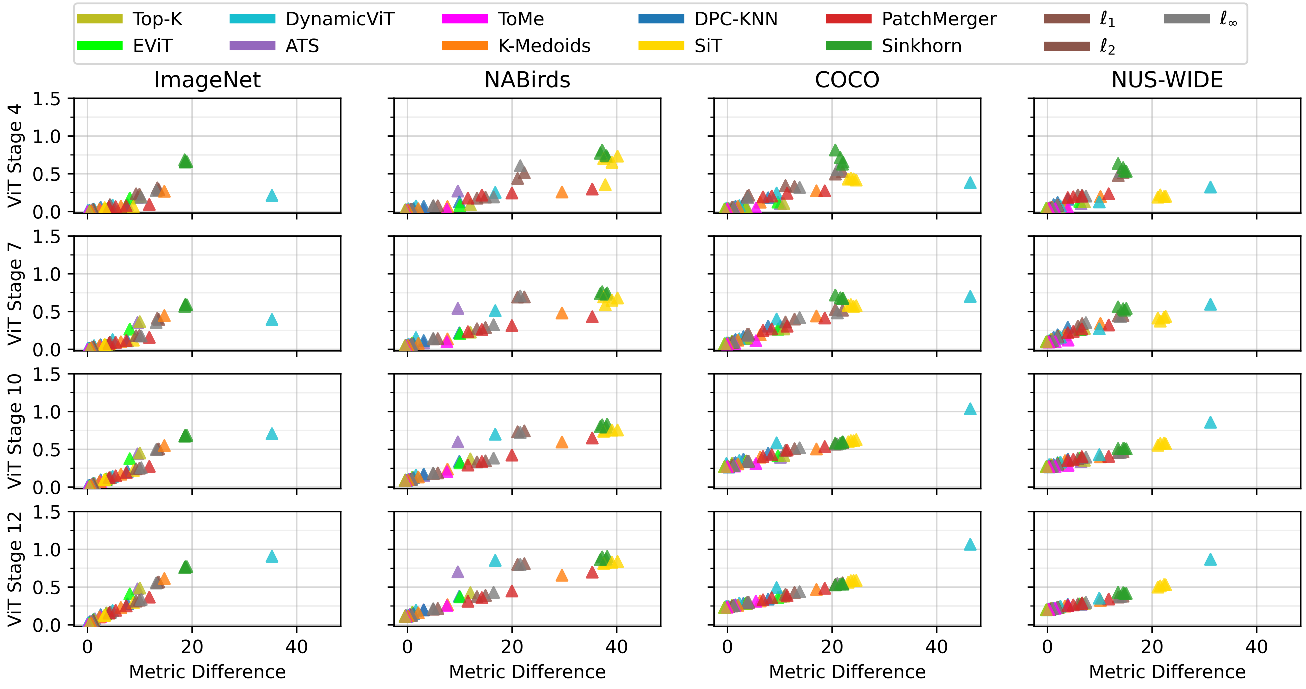


Figure 9: **Procrustes and DeiT-T anchor model as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with DeiT-T as anchor model.

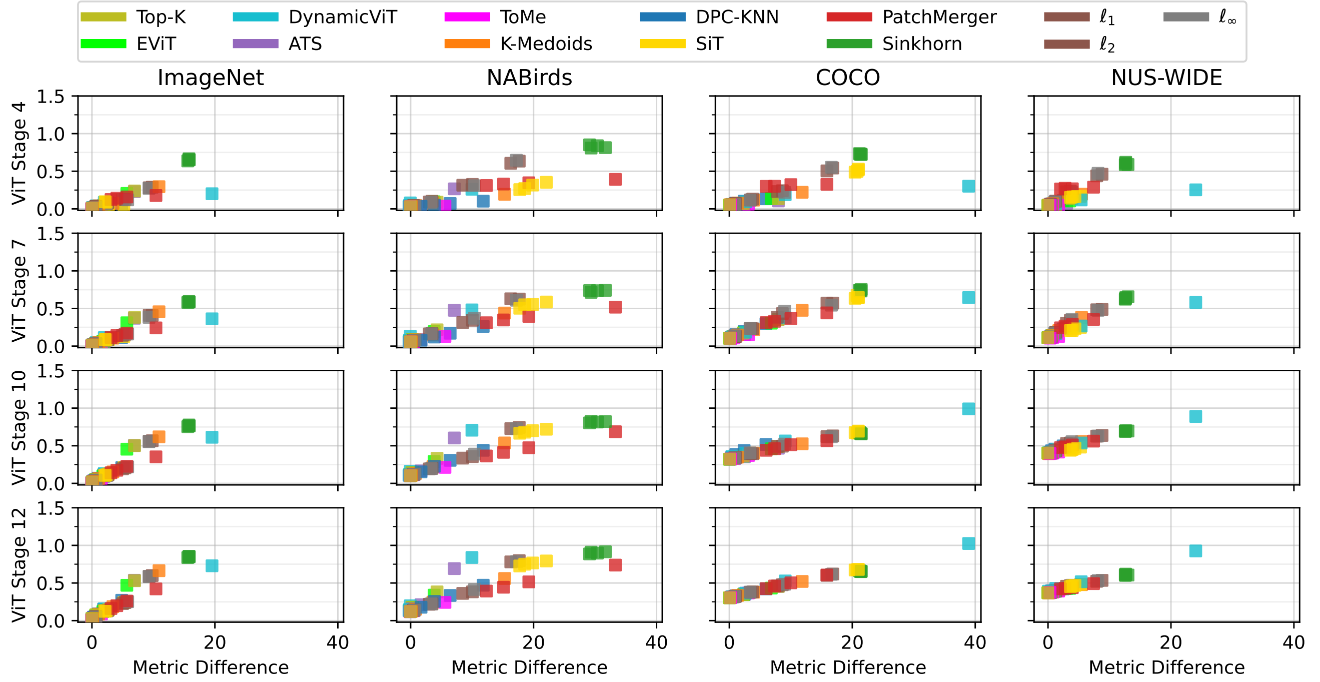


Figure 10: **Procrustes and DeiT-S anchor model as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with DeiT-S as anchor model.

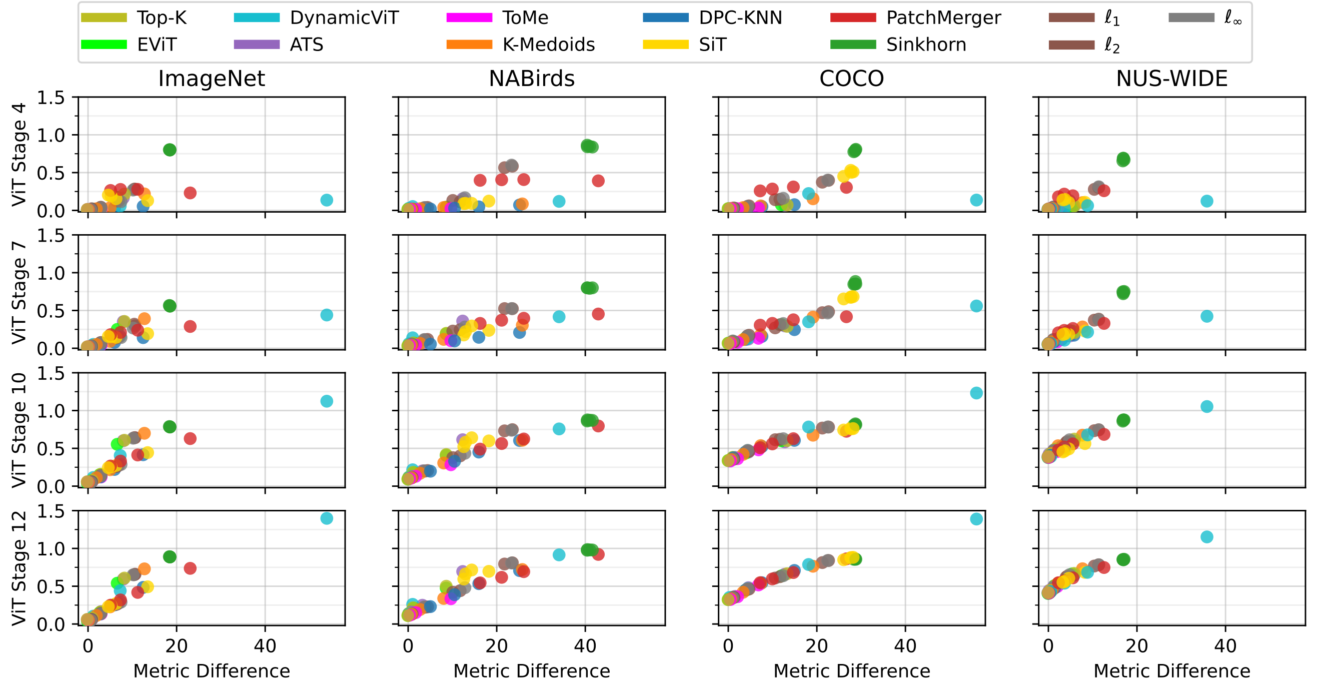


Figure 11: **Procrustes and DeiT-B anchor model as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with DeiT-B as anchor model.

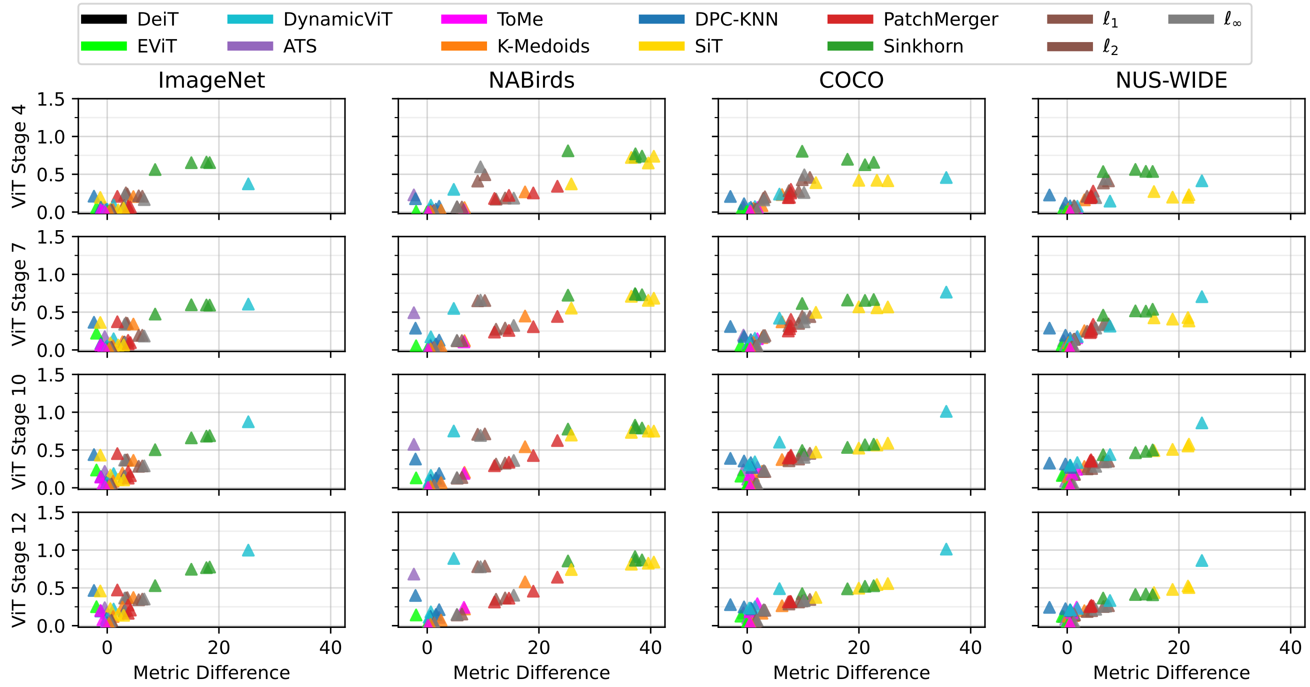


Figure 12: **Procrustes and Top-K anchor model with a DeiT-T backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with Top-K as anchor model.

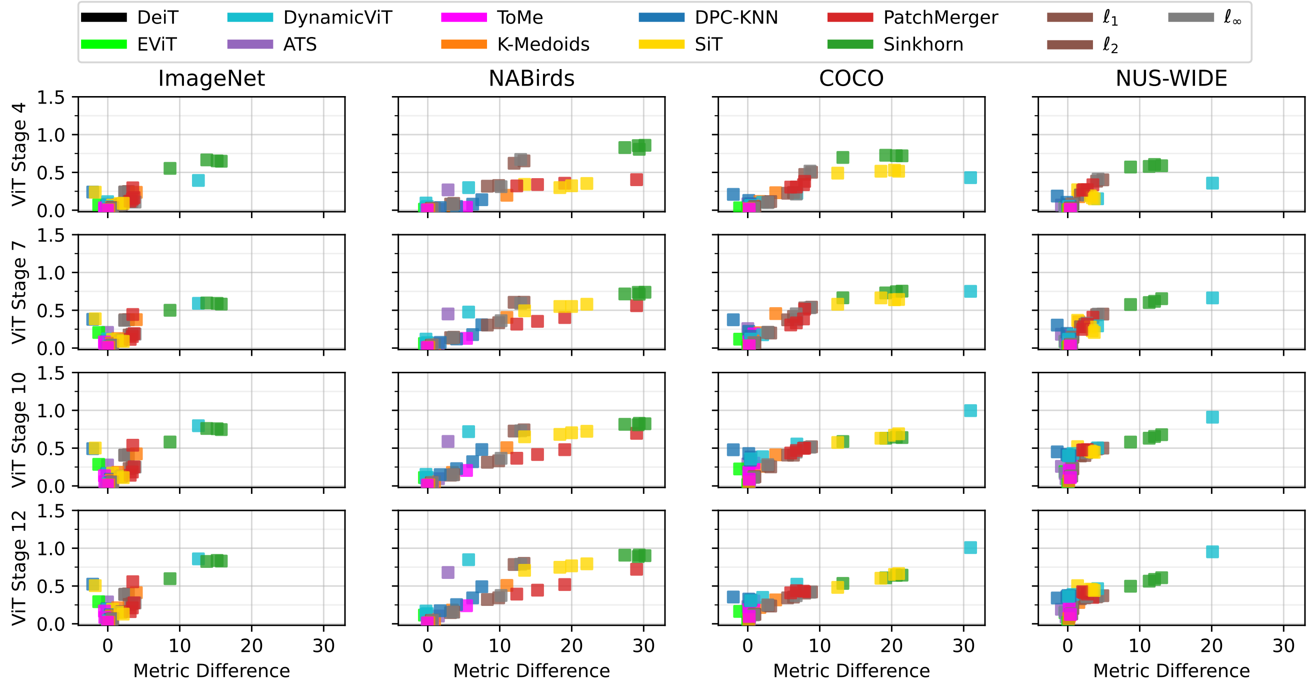


Figure 13: **Procrustes and Top-K anchor model with a DeiT-S backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with Top-K as anchor model.

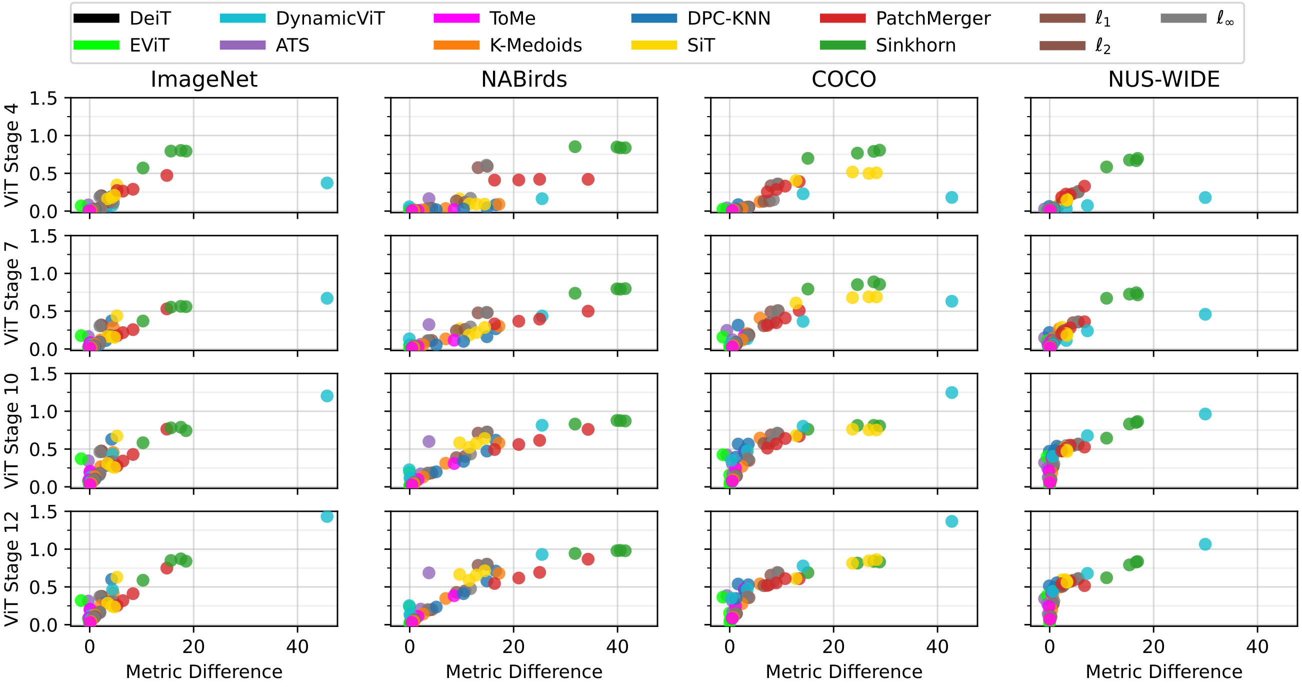


Figure 14: **Procrustes and Top-K anchor model with a DeiT-B backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with Top-K as anchor model.

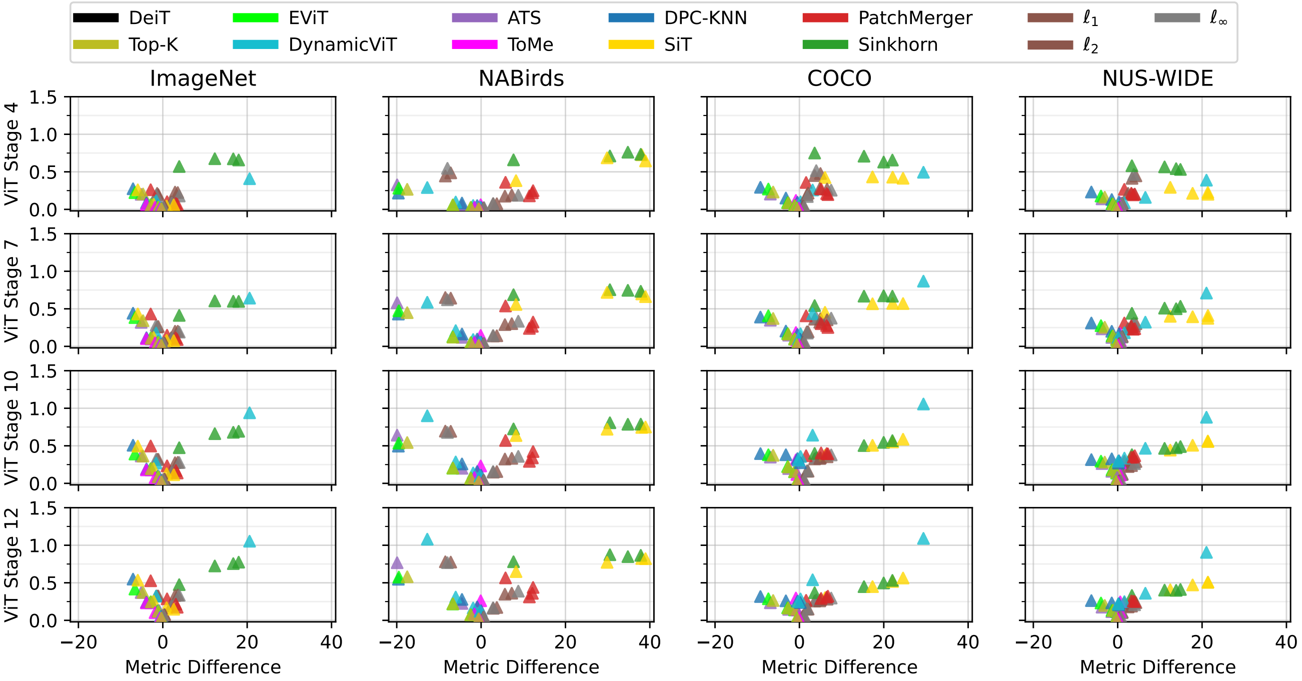


Figure 15: **Procrustes and K-Medoids anchor model with a DeiT-T backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with K-Medoids as anchor model.

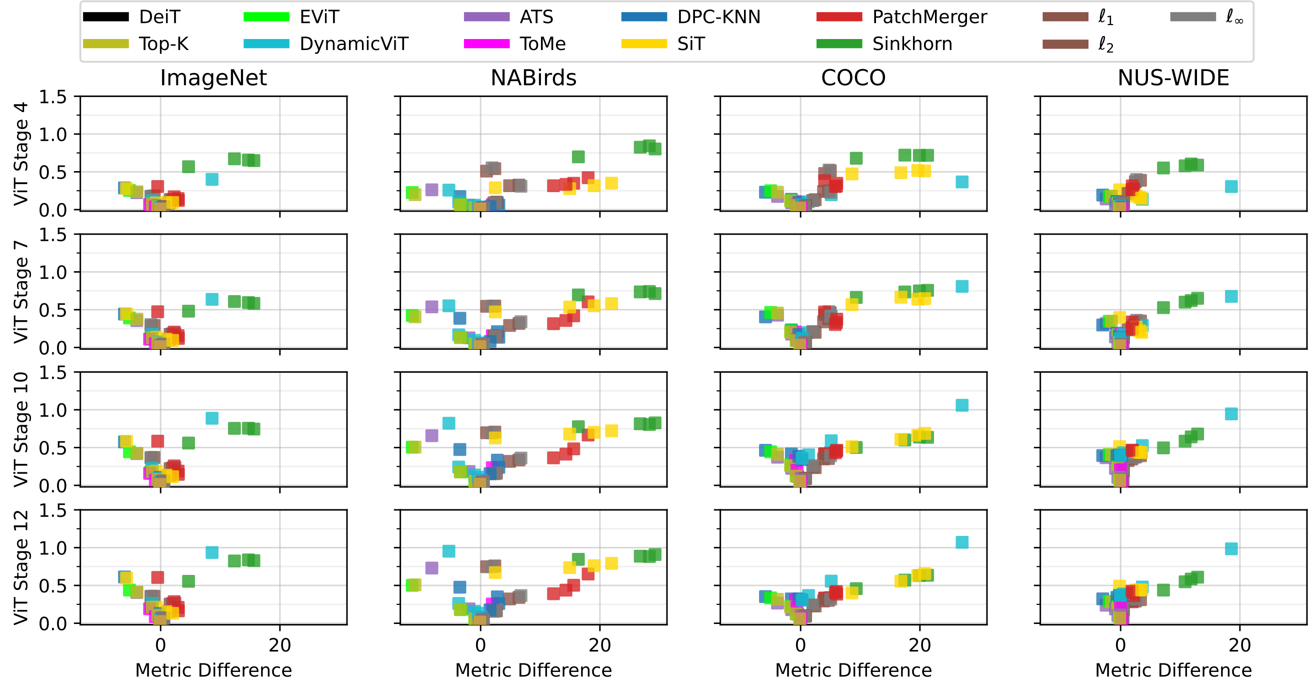


Figure 16: **Procrustes and K-Medoids anchor model with a DeiT-S backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with K-Medoids as anchor model.

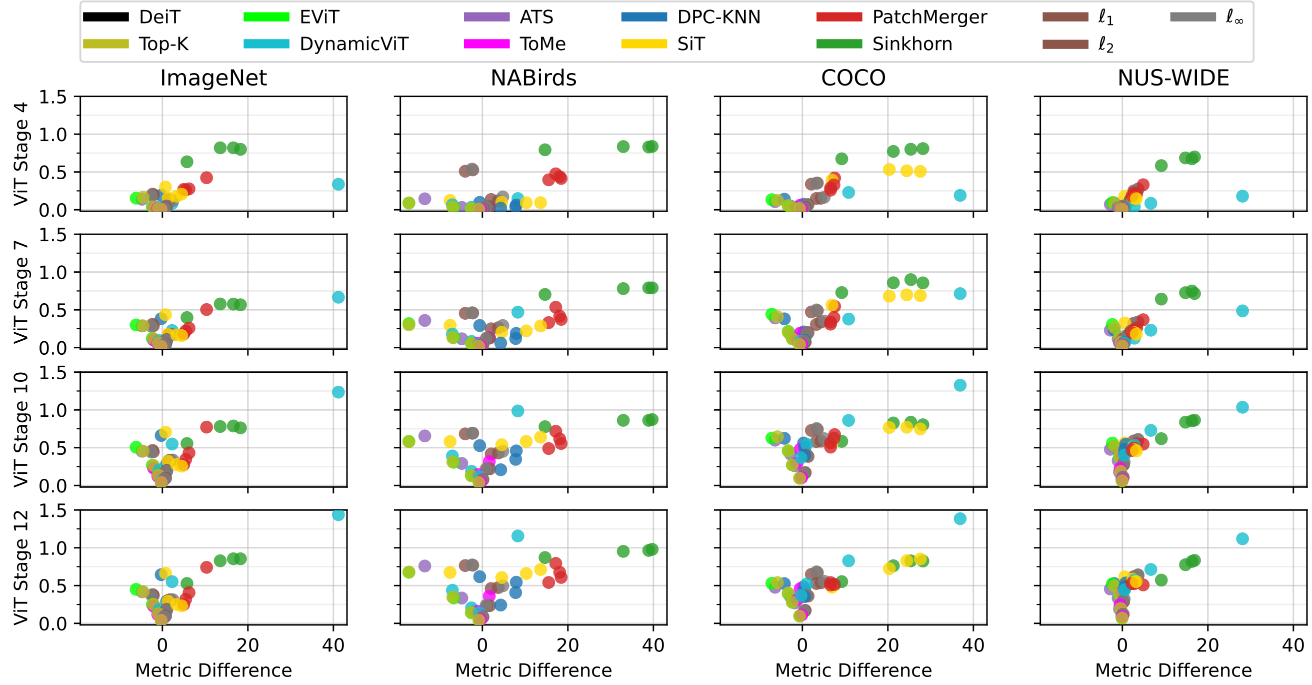


Figure 17: **Procrustes and K-Medoids anchor model with a DeiT-B backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the orthogonal Procrustes distance with K-Medoids as anchor model.

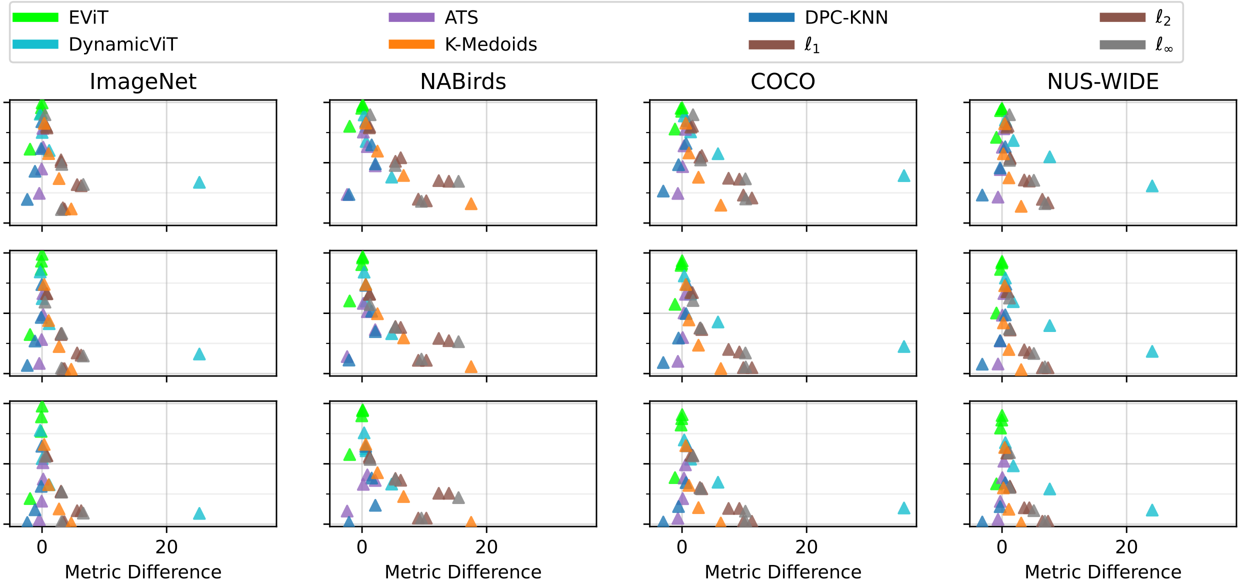


Figure 18: **IoU and Top-K anchor model with a DeiT-T backbone as model performance proxy (Section K).** Scatter plot between difference in performance and the IoU with Top-K as anchor model.

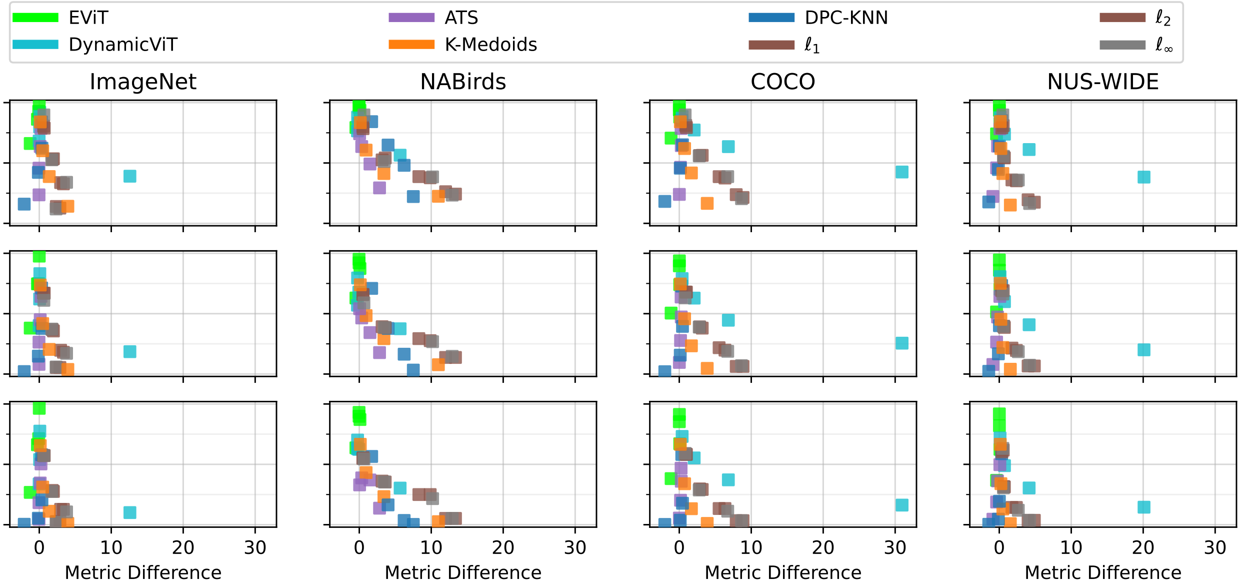


Figure 19: **IoU and Top-K anchor model with a DeiT-S backbone as model performance proxy (Section K).** Scatter plot between difference in performance and the IoU with Top-K as anchor model.

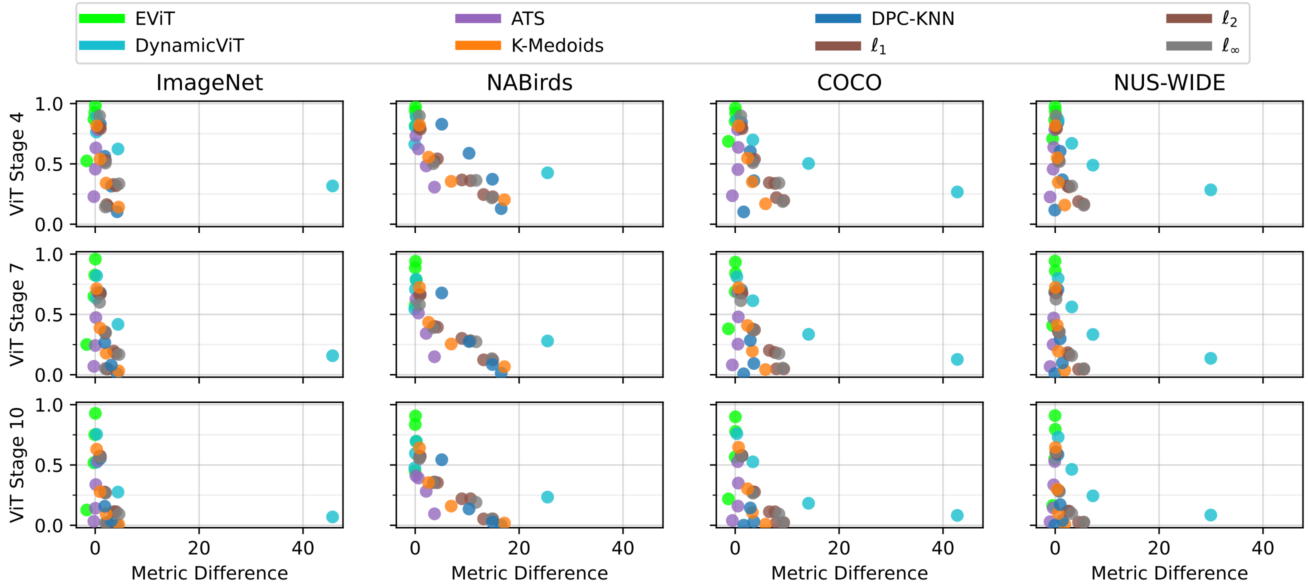


Figure 20: **IoU and Top-K anchor model with a DeiT-B backbone as model performance proxy (Section K).** Scatter plot between difference in performance and the IoU with Top-K as anchor model.

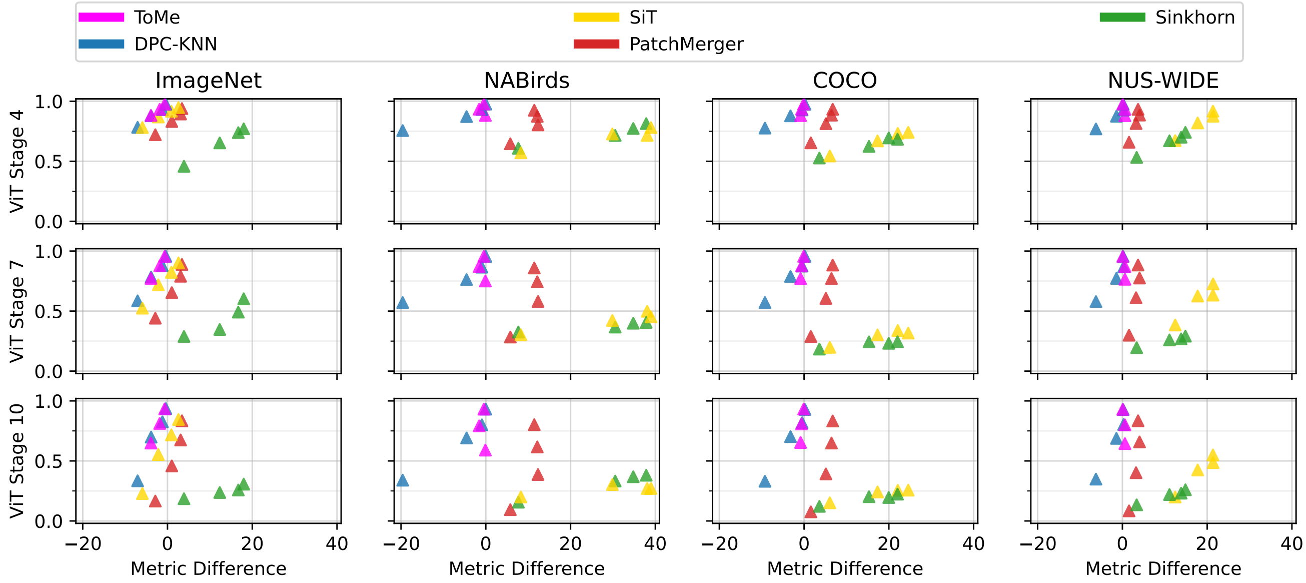


Figure 21: **NMI and K-Medoids anchor model with a DeiT-T backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the NMI with K-Medoids as anchor model.

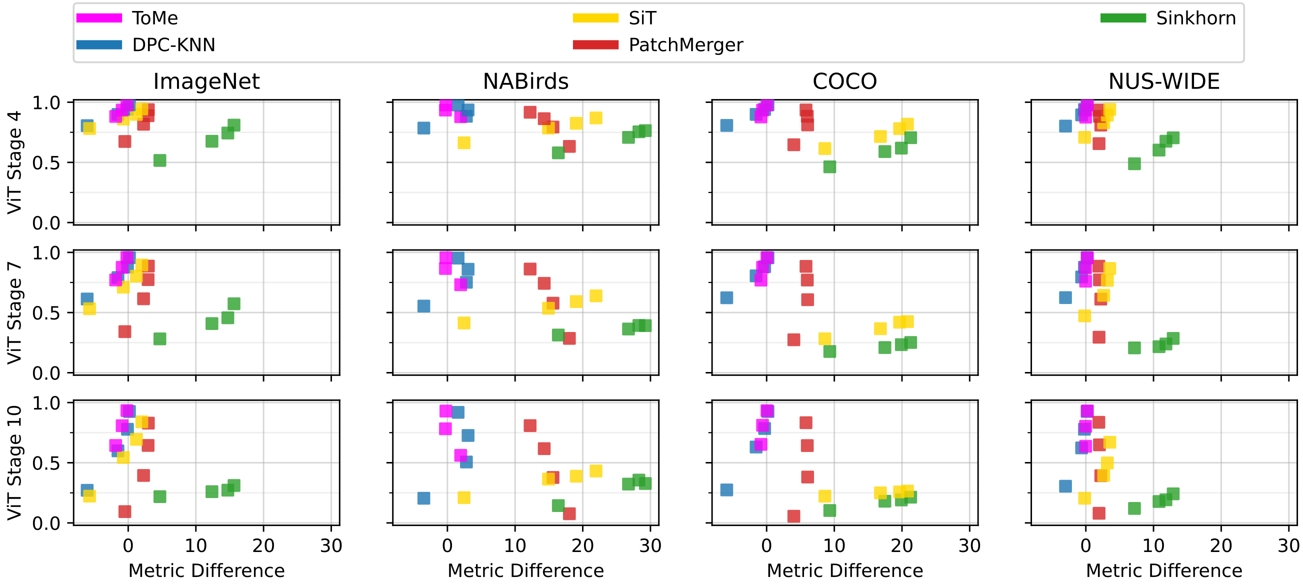


Figure 22: **NMI and K-Medoids anchor model with a DeiT-S backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the NMI with K-Medoids as anchor model.

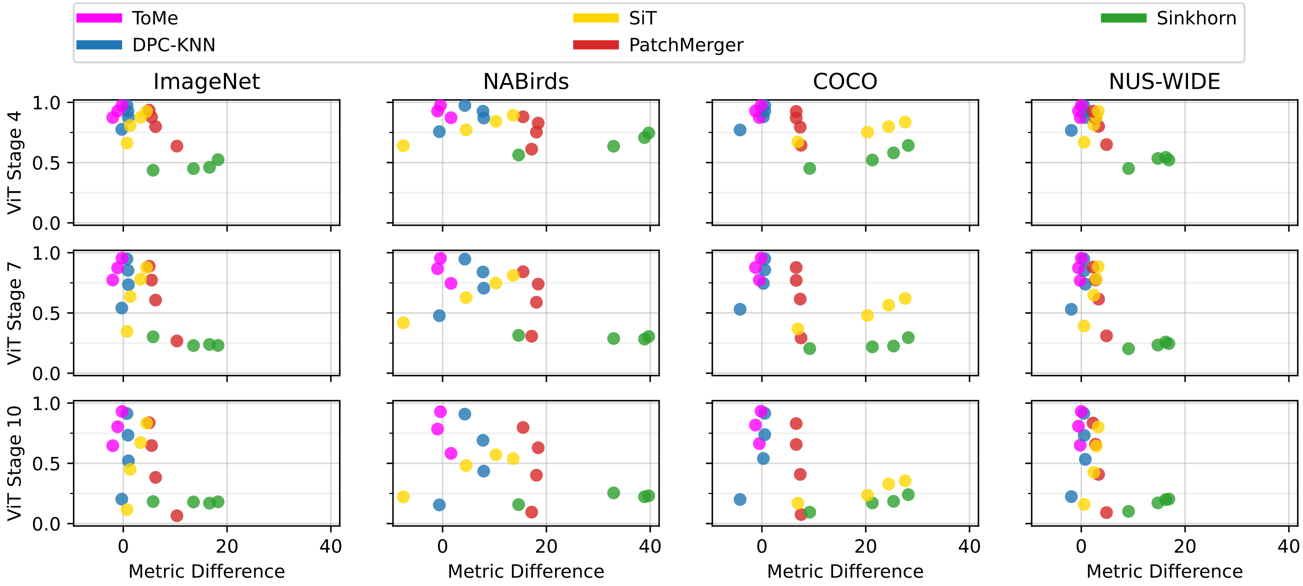
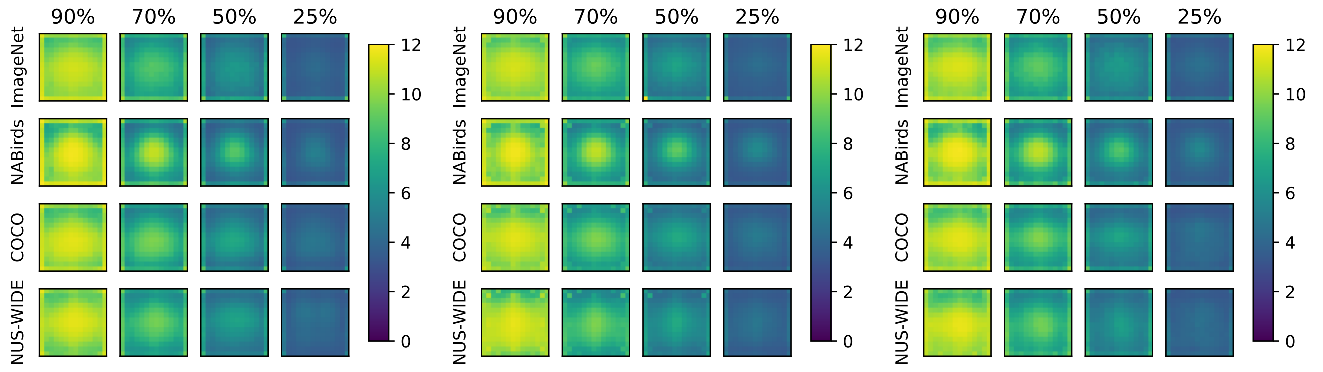
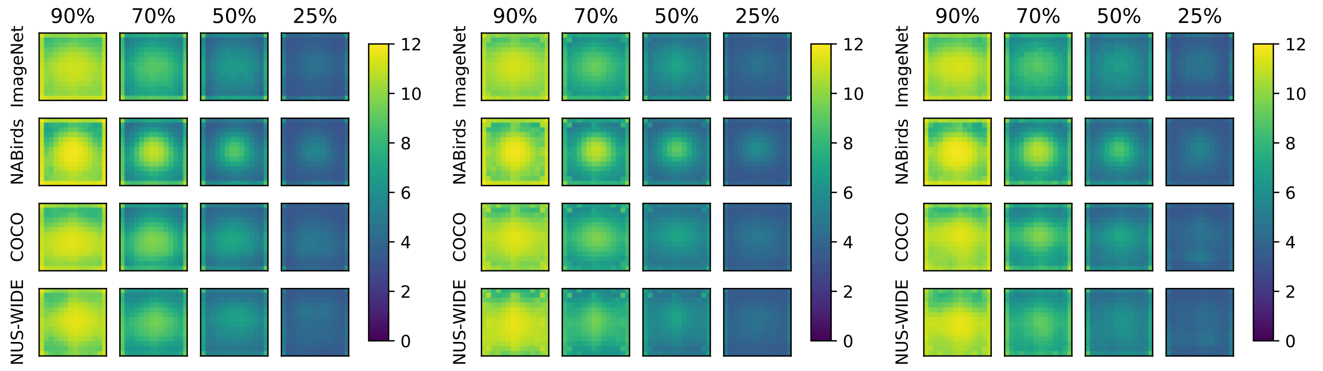


Figure 23: **NMI and K-Medoids anchor model with a DeiT-B backbone as model performance proxy (Section K).** Scatter plot between difference in model performance and the NMI with K-Medoids as anchor model.



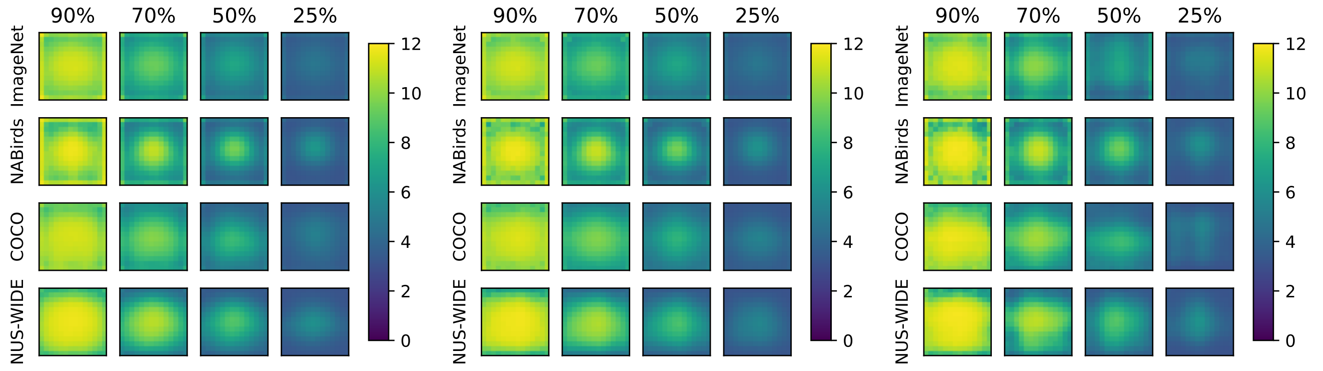
(a) DeiT-T Backbone (b) DeiT-S Backbone (c) DeiT-B Backbone

Figure 24: **Top-K Dataset-averaged reduction patterns (Section L).** We find that the Top-K method on average select tokens from both the image center and border.



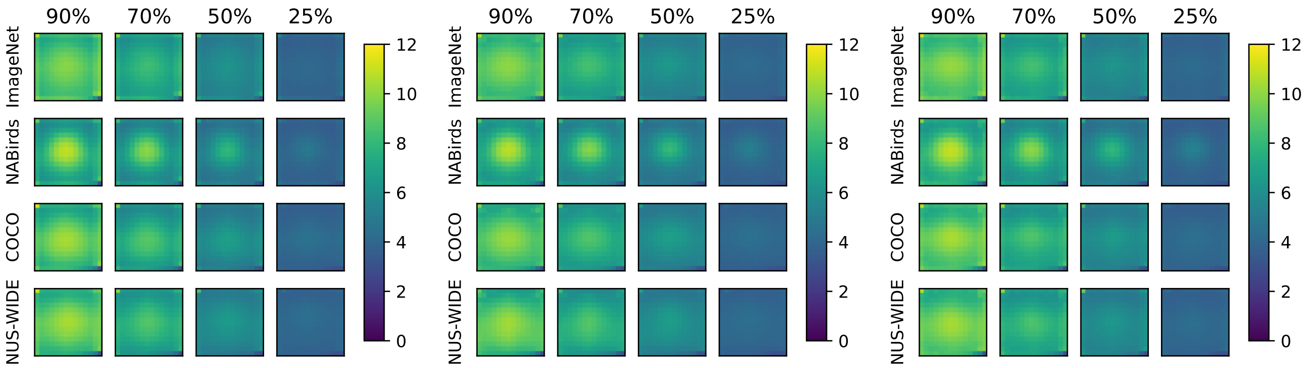
(a) DeiT-T Backbone (b) DeiT-S Backbone (c) DeiT-B Backbone

Figure 25: **EViT Dataset-averaged reduction patterns (Section L).** We find that the EViT method on average select tokens from both the image center and border.



(a) DeiT-T Backbone (b) DeiT-S Backbone (c) DeiT-B Backbone

Figure 26: **DynamicViT Dataset-averaged reduction patterns (Section L).** We find that the DyanmicViT method on average selects tokens primarily from the image center, ignoring the borders.

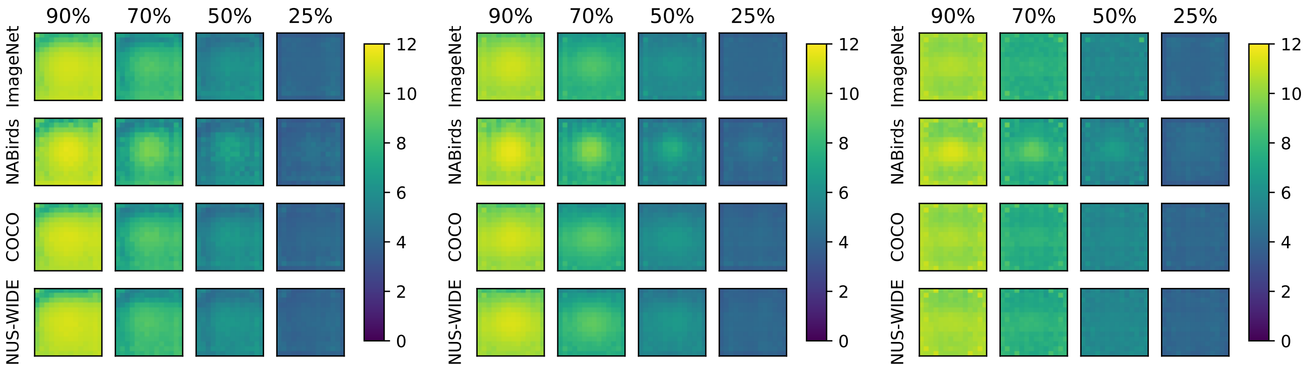


(a) DeiT-T Backbone

(b) DeiT-S Backbone

(c) DeiT-B Backbone

Figure 27: **ATS Dataset-averaged reduction patterns (Section L).** We find that the ATS method on average select tokens from both the image center and border, and that the tokens have a lower depth on average due to its dynamic keep rate.

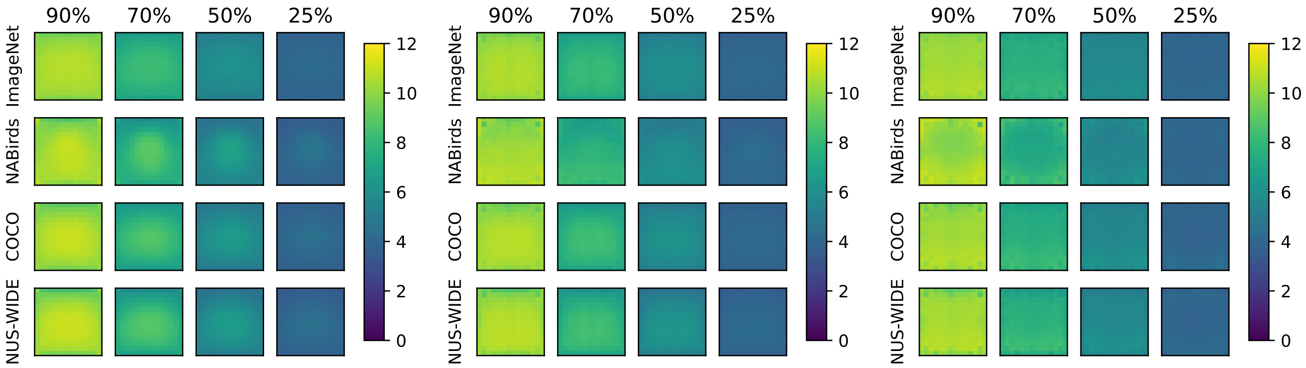


(a) DeiT-T Backbone

(b) DeiT-S Backbone

(c) DeiT-B Backbone

Figure 28: **K-Medoids Dataset-averaged reduction patterns (Section L).** We find that the K-Medoids method select tokens in a more uniform manner than the pruning-based methods.

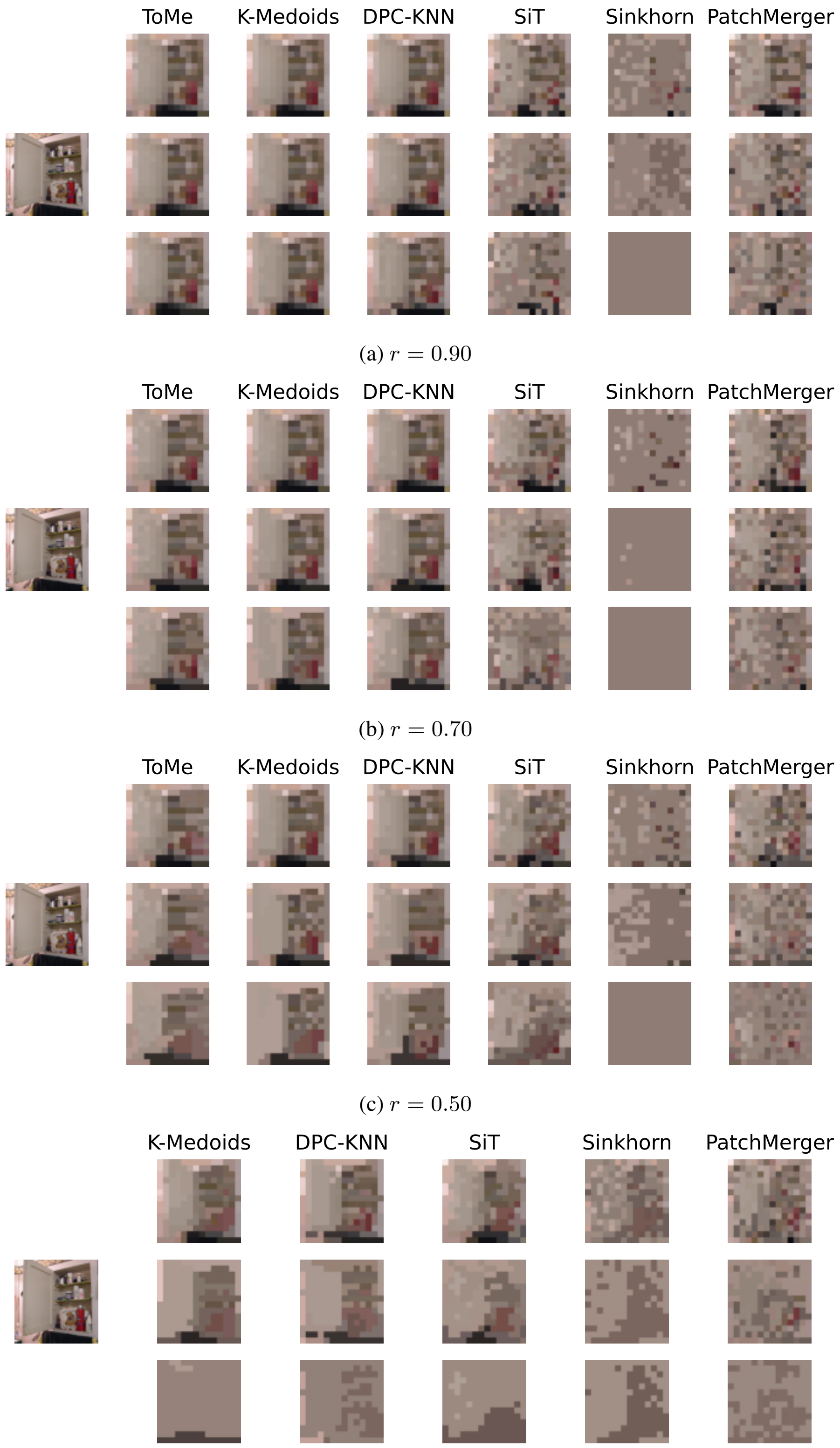


(a) DeiT-T Backbone

(b) DeiT-S Backbone

(c) DeiT-B Backbone

Figure 29: **DPC-KNN Dataset-averaged reduction patterns (Section L).** We find that the DPC-KNN method select tokens in a more uniform manner than the pruning-based methods.



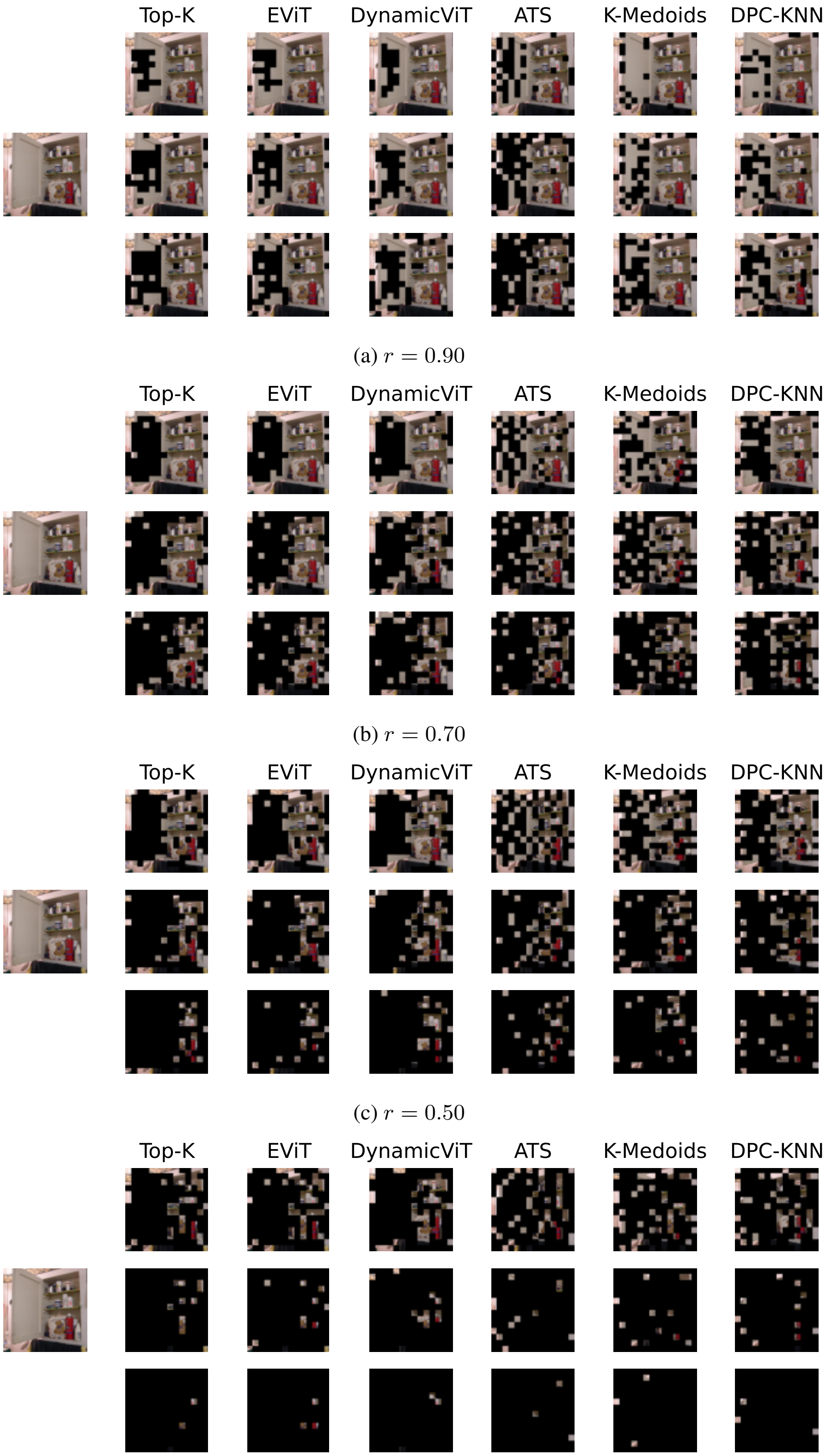
(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 30: **Cluster Reduction Patterns - ImageNet (Section M).** Example of constructed clusters obtained at different keep rate r values, on a random image from the ImageNet dataset.



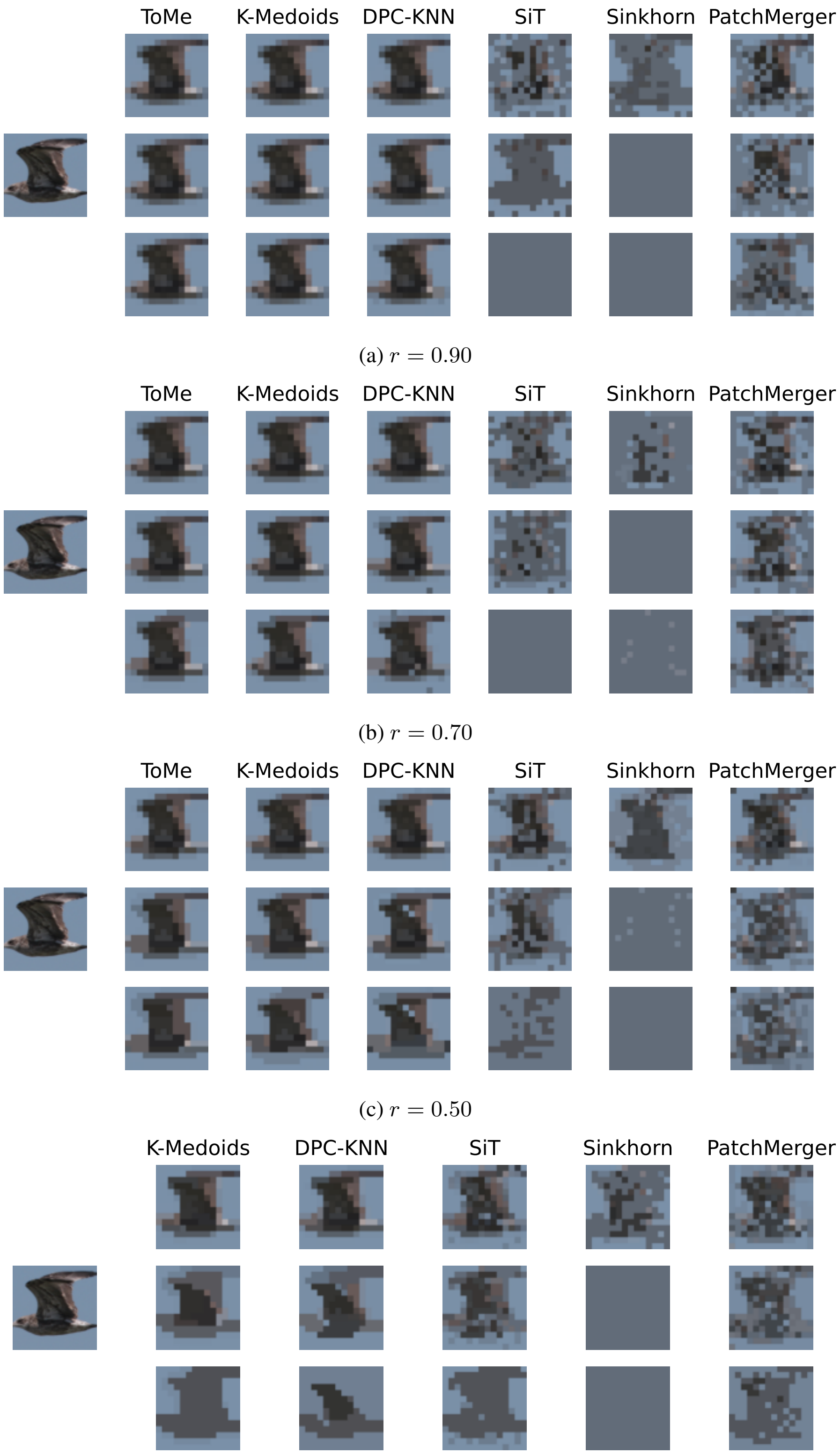
(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 31: **Pruning Reduction Patterns - ImageNet (Section M).** Example of pruning reduction patterns obtained at different keep rate r values, on a random image from the ImageNet dataset.



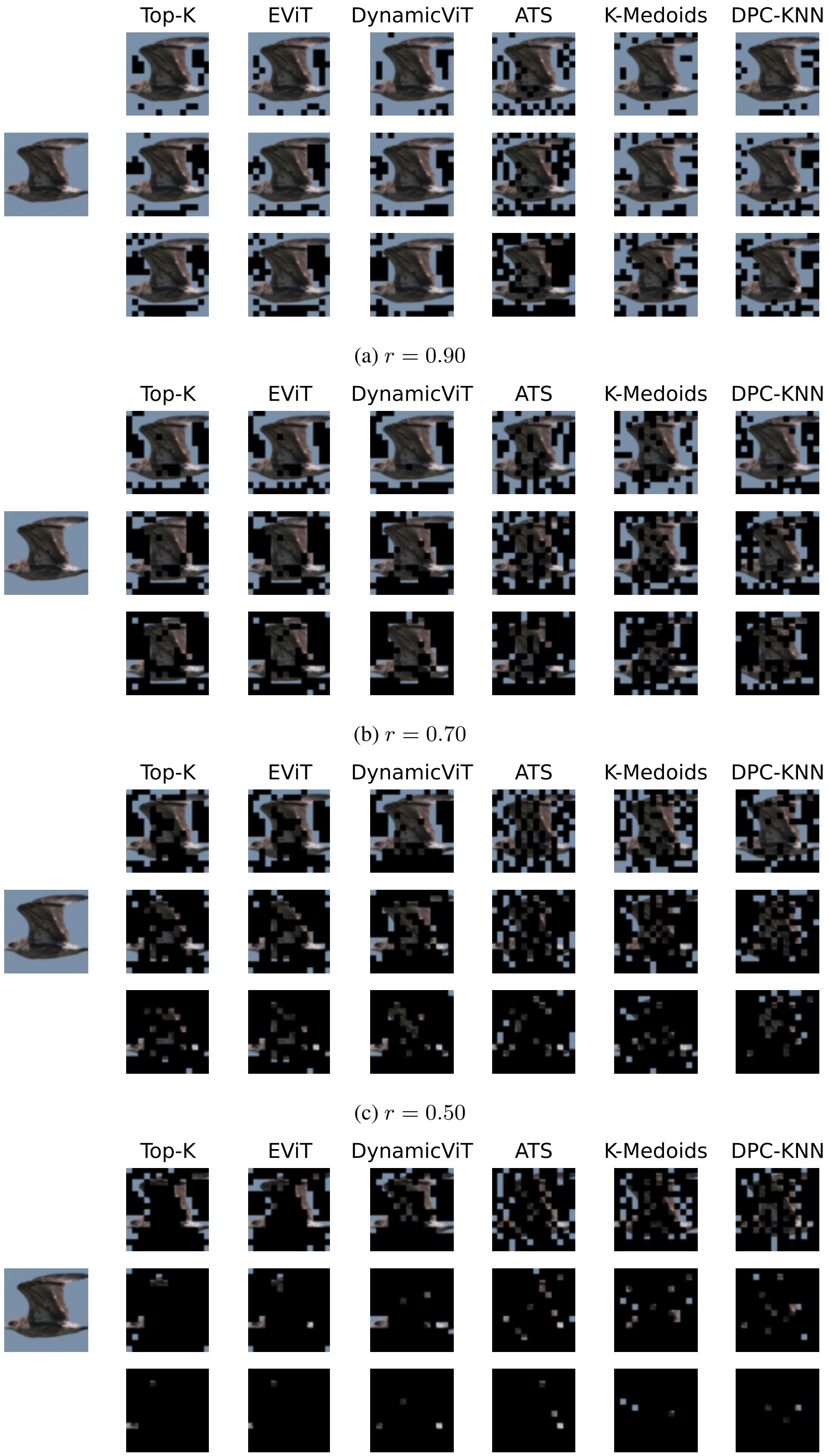
(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 32: **Cluster Reduction Patterns - NABirds (Section M).** Example of constructed clusters obtained at different keep rate r values, on a random image from the NABirds dataset.



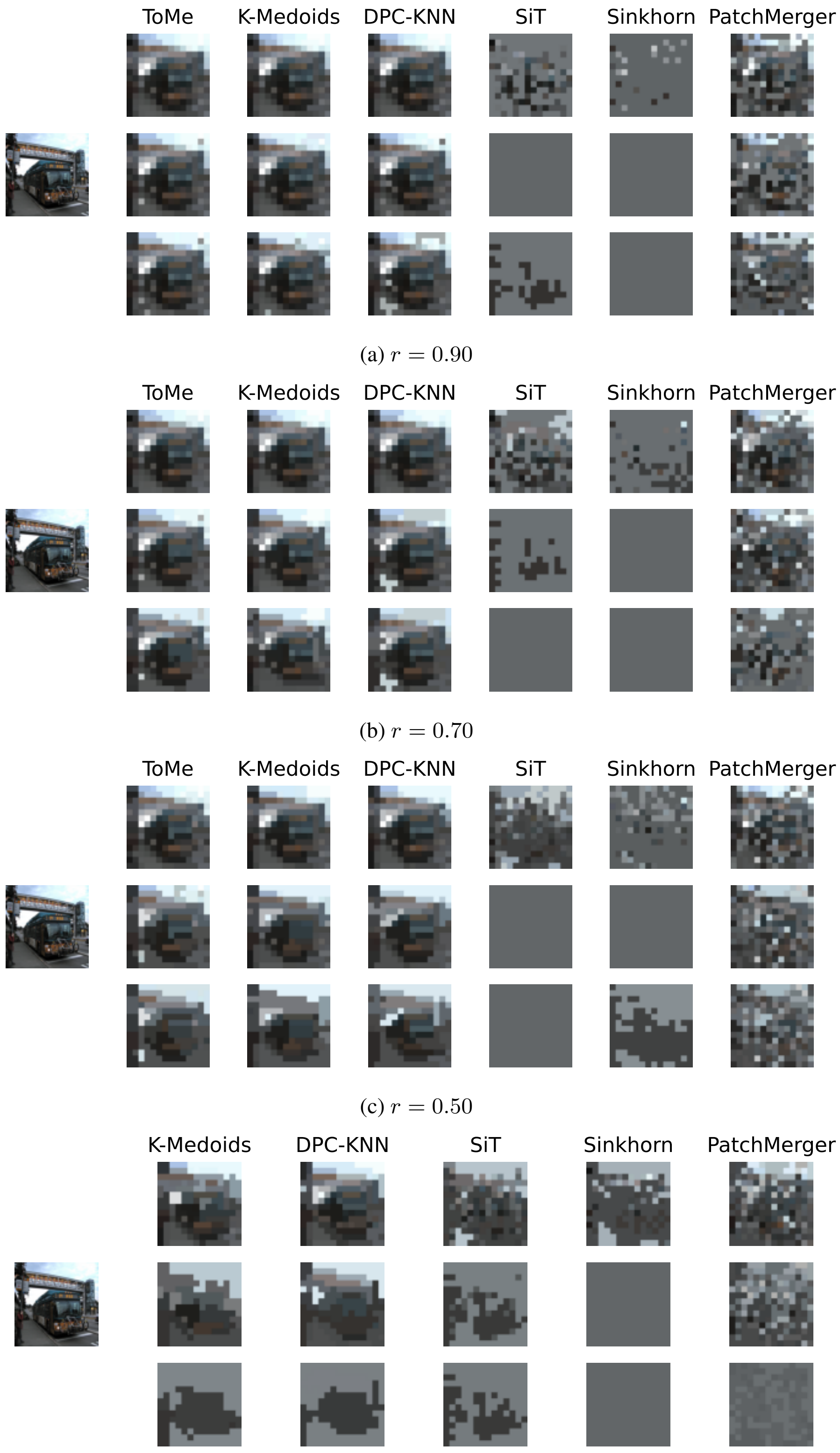
(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 33: **Pruning Reduction Patterns - NABirds (Section M).** Example of pruning reduction patterns obtained at different keep rate r values, on a random image from the NABirds dataset.



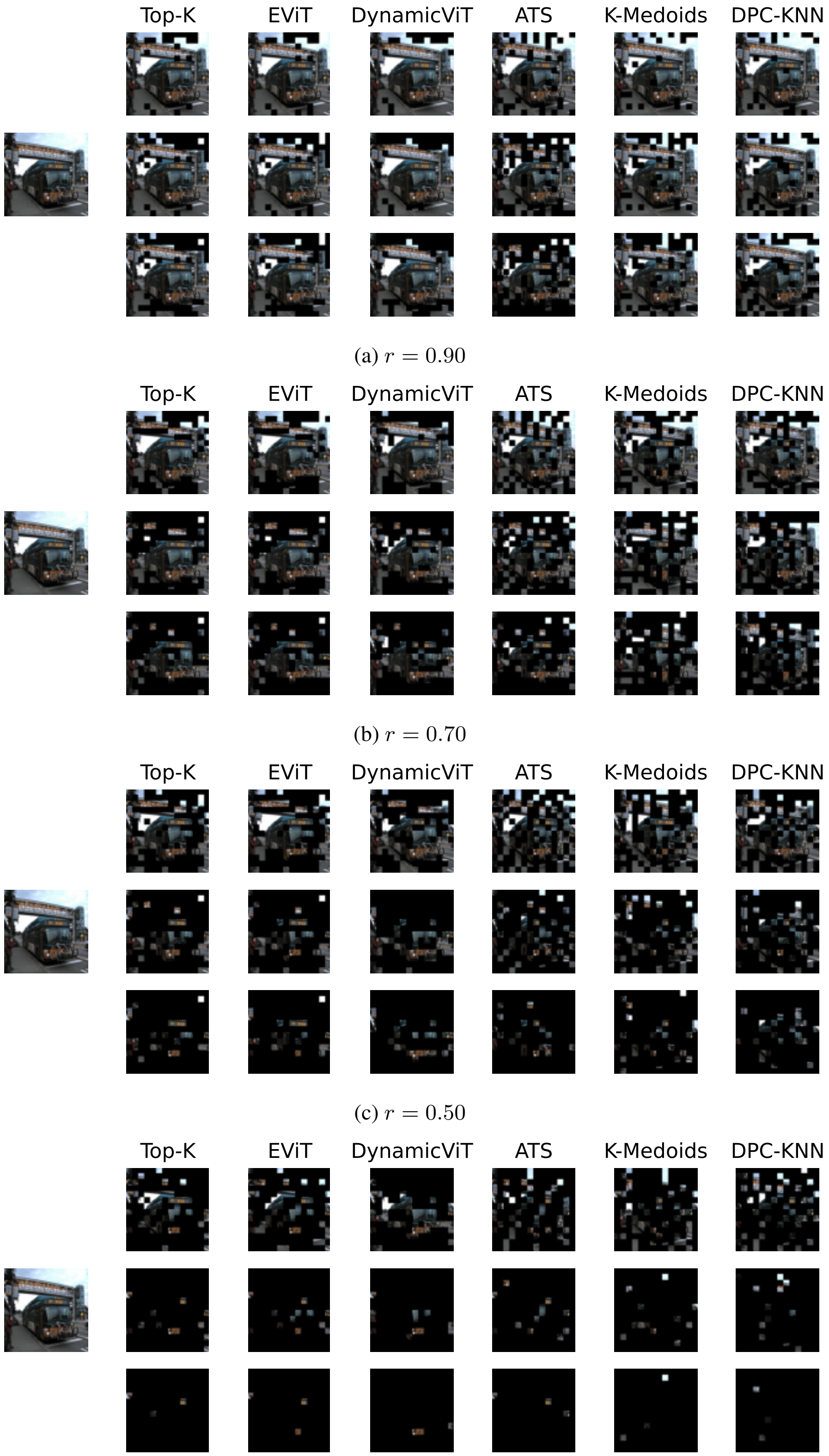
(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 34: **Cluster Reduction Patterns - COCO (Section M).** Example of constructed clusters obtained at different keep rate r values, on a random image from the COCO dataset.



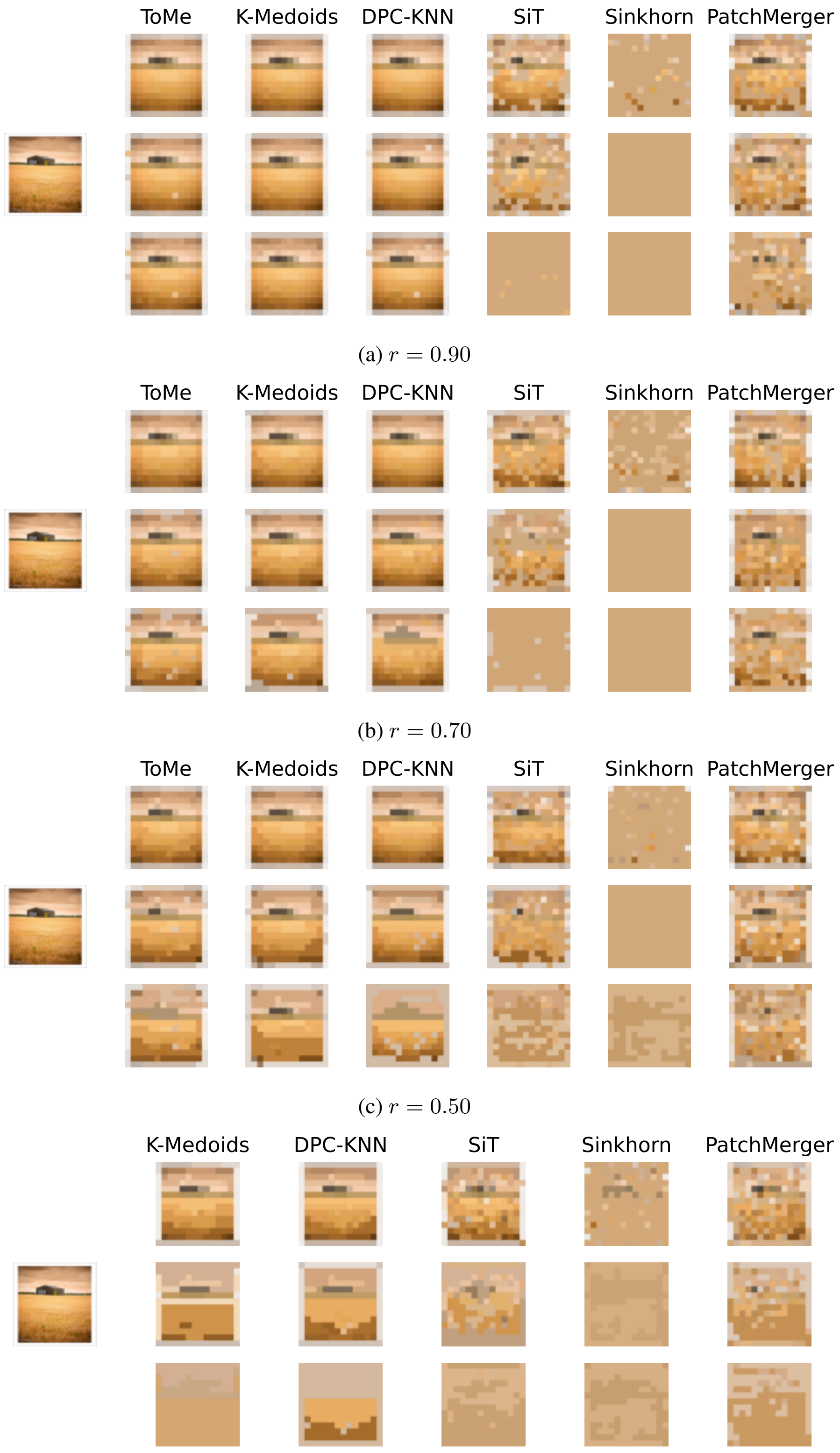
(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 35: **Pruning Reduction Patterns - COCO (Section M).** Example of pruning reduction patterns obtained at different keep rate r values, on a random image from the COCO dataset.



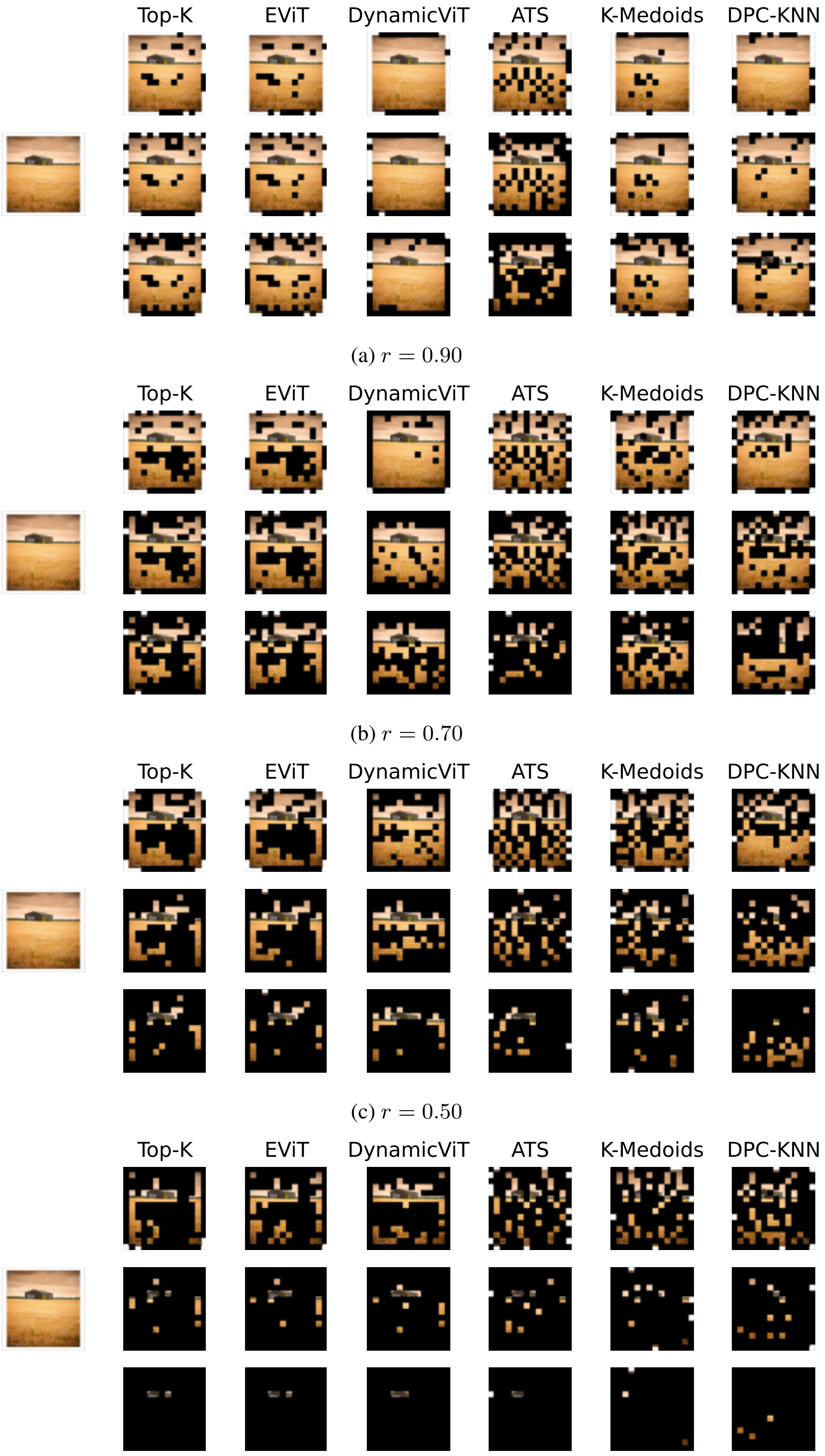
(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 36: **Cluster Reduction Patterns - NUS-WIDE (Section M).** Example of constructed clusters obtained at different keep rate r values, on a random image from the NUS-WIDE dataset.



(a) $r = 0.90$

(b) $r = 0.70$

(c) $r = 0.50$

(d) $r = 0.25$

Figure 37: **Pruning Reduction Patterns - NUS-WIDE (Section M).** Example of pruning reduction patterns obtained at different keep rate r values, on a random image from the NUS-WIDE dataset.

References

[1] Hila Becker. Identification and characterization of events in social media. 2011. 5

[2] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019. 1

[3] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3), 2019. 6, 9

[4] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020. 2

[5] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2

[6] Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 6

[7] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. Adaptive token sampling for efficient vision transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 3

[8] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 2

[9] Ju He, Jie-Neng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, and Changhu Wang. Transfg: A transformer architecture for fine-grained recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):852–860, Jun. 2022. 1, 2

[10] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8126–8135, 2020. 1

[11] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–661, Cham, 2016. Springer International Publishing. 1

[12] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019. 6

[13] Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 1

[14] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 1

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 1

[16] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 6

[17] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 1, 2

[18] Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 1, 2

[19] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 3, 5

[20] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, mar 2003. 3

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 18–24 Jul 2021. 1, 2

[22] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[23] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2

[24] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13001–13008, Apr. 2020. 2