A. Appendix Overview

In our primary results, we focused on comparing ViT-B/16 models trained with different supervision methods. In this Appendix, we present additional results for a wider range of ViT variants including different architecture sizes and patch sizes. This analysis includes 20 models in total, summarized in Table 2. This Appendix is organized following the same three major domains: Attention, Features, and Downstream Tasks. We summarize all the key findings of our work in Table 3.

B. Additional Experimental Details

B.1. ViT Variants Examined

To provide a uniform platform for comparison, our primary results focus on ViT-B/16 models trained with six methods: Full Supervision, CLIP [51], DINO [9], MoCo-v3 [12], MAE [26], and BEiT [4]. In this Appendix, we present additional results examining different ViT variants for the above methods as provided by their original authors. In total, this expanded ViT collection contains 20 models including instances of ViT Small, Base, Large, and Huge as well as patch sizes 8, 14, 16, and 32. We continue to process all images at a 224 × 224 input resolution, meaning the number of spatial tokens for a given model will vary by patch size, from 47 spatial tokens for models with patch size 32 up to 784 spatial tokens for patch size 8. Overall, the models evaluated are summarized in Table 2. Note that the MoCo ViT-Small model is a modified variant with 12 heads per layer instead of 6.

B.2. ViT Training Details

In this section, we briefly outline the training protocols that were used to train each of the ViTs examined in this work. The CLIP, DINO, MoCo, MAE, and BEiT models are all official pre-trained models released by their original authors.

Fully Supervised (FS). For FS, we work with models from the TIMM repository [69]. The FS models are pretrained on ImageNet21k and fine-tuned on ImageNet1k. The FS models are the only models in this study that are fine-tuned with ImageNet-1k labels. They are trained following the augmentation protocols of [58]. Specifically, the ViT-Base and Large models are trained using a combination of RandAugment [15] and Mixup [74], while the ViT-Small models use only RandAugment. All FS models also use weight decay [38].

CLIP. The goal of CLIP (**C**ontrastive **L**anguage-**I**mage **P**re-Training) is to train models with open-ended supervision provided by paired captions. The learning objective is simply to match images with their corresponding captions. CLIP models are joint vision and language models, which include separate encoder networks for the image and text inputs. For our analysis, we focus only on the properties of the visual encoding network. The authors pretrain the model on 400M image-text pairs with a batch size of 32768 and mixed-precision to accelerate training and reduce memory usage. The only augmentation used is taking a random square crop from the resized image. They also use a cosine learning rate decay schedule.

Table 2. Summary of all ViT Variants used in Appendix analysis. *MoCo S/16 uses 12 heads per layer instead of 6.

Model	**Layers**	**Heads**	**Spatial Token Grid Size**
FS S/32	12	6	7x7
FS S/16	12	6	14x14
FS B/32	12	12	7x7
FS B/16	12	12	14x14
FS B/8	12	12	28x28
FS L/16	24	16	14x14
CLIP B/32	12	12	7x7
CLIP B/16	12	12	14x14
CLIP L/14	24	16	16x16
DINO S/16	12	6	14x14
DINO S/8	12	6	28x28
DINO B/16	12	12	14x14
DINO B/8	12	12	28x28
MoCo S/16*	12	12	14x14
MoCo B/16	12	12	14x14
MAE B/16	12	12	14x14
MAE L/16	24	16	14x14
MAE H/14	32	16	16x16
BEiT B/16	12	12	14x14
BEiT L/16	24	16	14x14

MoCo. The **Mo**mentum **Co**ntrast (MoCo) method trains using contrastive learning with a momentum encoder, which is an exponential moving average of previous versions of the encoder. Under the contrastive objective, the encoder must generate representations for query image views that are similar to corresponding representations of key image views as generated by the momentum encoder. This strategy was proposed for CNNs and extended to ViTs. For MoCo v3 [12], the authors pretrain the model on ImageNet-1k without labels with a batch size of 4096. They follow a cosine learning rate decay. They use data augmentations like random resized cropping, horizontal flipping, color jittering, grayscale conversion, blurring, and solarization. They take two 224 × 224 crops for each image for each iteration.

DINO. The authors of DINO [9] describe their method as a form of self-**di**stillation with **no** labels. Their training strategy is based on MoCo [27] and they also use a momentum encoder, though they instead view their method as a student/teacher knowledge distillation framework. They pretrain the models on ImageNet-1k without labels with batch

Table 3. A comprehensive summary of the key observations of this work, including both the main paper and appendix results.

Key Observations	**Analysis Methods**	**Sections**	**Figures & Tables**
The attention maps of explicitly supervised ViTs devolve into Sparse Repeating Attention Patterns in the mid-to-late layers.	Average CLS Attention Maps	4.1	Figure 2
All ViTs studied learn to use Offset Local Attention Heads, suggesting they are fundamentally necessary in ViTs.	Aligned Aggregated Attention Maps	4.2	Figure 3
ViTs learn to process local and global information in different orders depending on their method of supervision.	Average Attention Distance	4.3	Figure 4
All ViTs studied differentiate salient foreground objects by the early-to-mid layers.	Attention Saliency IoU	4.4	Figure 5
Reconstruction-based self-supervised methods can learn semantically meaningful CLS representations, even when the CLS token is only a placeholder.	CKA Feature Similarity, Image Clustering by CLS Features	5.1, 5.2	Figures 6, 7
Supervised method's features are the most semantically rich, but contrastive self-supervised methods are comparable or even superior in some cases.	Image-, Object-, and Part-Level Feature Clustering	5.2, 5.3	Figures 7, 8
For localized tasks, the best performance often comes from a mid-to-late layer.	Local Downstream Tasks	6.2	Figure 10 Table 1
There is no single "best" training method or layer for all downstream tasks.	Local & Global Downstream Tasks	6.3	Figures 9, 10 Table 1
The positions of maximal activation in the Sparse Repeating Attention Patterns vary by input.	CLS Attention Maps	C.1	Figures 11-13
All models studied learn to use Offset Local Attention Heads, and some larger models even learn ones with diagonal offsets.	Aligned Aggregated Attention Maps	C.1	Figure 14
The order of local *vs*. global information processing in a ViT is primarily determined by the method of supervision and is largely unaffected by changes in architecture and patch size.	Average Attention Distance	C.2	Figure 15
For the expanded ViT collection, again all models differentiate salient foreground objects by the early-to-mid layers.	Attention Saliency IoU	C.3	Figure 16
Explicitly supervised ViTs with patch size 32 are less impacted by Sparse Repeating Attention Patterns, suggesting they may be an indication of overfitting.	Averaged CLS Attention Maps	C.1, C.3	Figure 17
The last layer spatial representations of self-supervised methods are similar across changes in architecture size and patch size, but this is not consistently true for explicitly supervised methods.	CKA Feature Similarity	D.2	Figure 19
Both MAE and BEiT show X patterns in their depth-wise feature CKAs, suggesting an encoder/decoder internal structure.	CKA Feature Similarity	D.3	Figures 21, 22
For larger MAE ViTs, the later layers appear to act more like decoder layers, even thought MAE has a separate decoder.	CKA Feature Similarity	D.3	Figure 21
Residual connection analysis provides further evidence of a fundamental shift in information processing in the mid-to-late layers of explicitly supervised ViTs.	Residual Connection Analysis	D.4	Figure 23
BEiT L/16 learns extremely expressive part-level features compared with BEiT B/16.	Part-Level Feature Clustering	D.5	Figure 27
Larger architectures tend to give better feature quality and downstream performance.	Feature Clustering, Local & Global Downstream Tasks	D.5, E.2	Figures 24-31
ViTs with smaller patch sizes unsurprisingly perform better at localized downstream tasks.	Local Downstream Tasks	E.2	Figures 30, 31
Reconstruction-based ViTs show the largest variance in their performance characteristics on downstream tasks.	Local & Global Downstream Tasks	E.2	Figures 28-31
For the expanded group of ViTs, once again there is no single "best" training method or layer for all downstream tasks.	Local & Global Downstream Tasks	E.2	Figures 28-31 Table 5

size 1024. They follow a cosine learning rate and weight decay. They use data augmentations like color jittering, gaussian blur, and solarization similar to BYOL [22]. Multi-crop [8] is also used.

MAE. The **M**asked **A**uto**e**ncoder (MAE) [26] method is a reconstruction-based training objective where a large portion of input patches/tokens are masked out. The rational of MAE is that, because a large percentage of the image content is missing, the network must learn representations that embed meaningful high-level semantics to reconstruct the missing regions. MAE uses both an encoder and decoder network, though the decoder is discarded after pretraining. The authors pretrain the model on ImageNet-1k without labels with a batch size of 4096. They do not use color jittering, drop path or gradient clipping and only apply random resized crop augmentation. They use a masking ratio of 0.75 which also improves the efficiency of training by significantly decreasing the token count in the encoder. They also use a cosine learning rate decay schedule.

BEiT. BEiT [4] stands for **B**idirectional **E**ncoder representation from **I**mage **T**ransformers, and it is based on BERT [17], a well-known masked reconstruction learning method for NLP. In contrast to MAE, BEiT does not perform pixel-level reconstruction, but instead uses a tokenizer to convert image patches into discrete tokens. The BEiT learning objective is to predict the token values for the masked patches. Unlike MAE, BEiT does not include a separate decoder network. BEiT is trained with a masking ratio of roughly 0.4, though they also employ a block-masking method which masks out larger adjacent groups of tokens. They pretrain BEiT on ImageNet-1k with a batch size of 2048, and they include random resized cropping, horizontal flipping, and color jittering augmentations. They also utilize cosine learning rate decay. Note that the authors have provided both BEiT models before and after fine-tuning with ImageNet labels. For our analysis, we work with the non-fine-tuned versions, in order to focus on just the effects of the BEiT pretraining method.

For more details and exact parameters, please refer to the corresponding papers and codebases for each of the models.

B.3. Random Chance Scores

During our Feature Clustering and Downstream Task Analysis, we present random chance scores for both the clustering metrics and downstream task scores. To evaluate these scores, we repeat the task analysis replacing all ViT features with uniformly distributed Gaussian noise. To be more specific, we generate arrays of Gaussian noise with the exact same dimensions as the extracted feature arrays of a ViT B/16 model. These random chance scores are heavily influenced by the underlying data. For example, we see that the random chance score is quite high for the object-level clustering purity scores on COCO. This can be attributed to the dataset's highly imbalanced object distribution. Still, this method of random chance evaluation is informative as it effectively acts as a baseline model where all feature vectors contain absolutely no useful information.

B.4. Dense Feature Extraction

Certain local tasks, like object segmentation, benefit from having a denser array of high quality features. For an input image of size 224×224, a ViT with patch size 16 produces a feature array with size 14×14. This low resolution can be very limiting for localized tasks, like DAVIS Video Segmentation Propagation. While some of the ViTs evaluated do support variable input size, others are hard-coded to operate at a fixed size. To generate a denser feature grid while also providing a level playing field, we propose a simple dense feature extraction strategy using image tiling.

We begin by rescaling the smaller image dimension to size 448 (twice the input resolution). We then scale the larger image dimension to the nearest integer multiple of the model patch size, in order to preserve the image aspect ratio as best as possible. Finally we slice the image into non-overlapping tiles of size 224×224. Then we extract ViT features for each of the tiles and concatenate the features together. Unless the image is exactly square, this leaves some leftover image content along the larger image dimension. For these areas, we take two additional crops which do overlap other image tiles, but for these areas the features are discarded, while the non-overlapping features are concatenated to the rest. The final product is a feature array that is twice as dense as the original while also respecting the original image aspect ratio.

In Table 4, we present an ablative analysis comparing the DAVIS Video Segmentation performance of all models with and without dense feature extraction enabled. All models see a significant performance boost with dense feature extraction, especially models with patch size 32.

B.5. Code Release

Our analysis codebase includes complete scripts for replicating the experiments and figures in the main work and appendix. Our source code is available at `www.github.com/mwalmer-umd/vit_analysis` and our project page can be found at `www.cs.umd.edu/~sakshams/vit_analysis`.

C. Attention Analysis

C.1. Expanded Attention Visualizations

Figure 11 and Figure 12 provide additional visualizations of CLS token attention maps in ViT-B/16 models for single input images. These visualizations show all layers (rows) and all heads (columns). From these views, we can see clear signs of the Sparse Repeating Attention Patterns

Table 4. Comparison of Dense vs. Normal feature extraction for the DAVIS Video Object Segmentation task. Results show for the best layer per model, with layer number in parenthesis.

Model	J and F Mean	
	Normal	Dense
FS S/32	0.18 (12)	0.39 (9)
FS S/16	0.34 (10)	0.58 (8)
FS B/32	0.17 (10)	0.38 (9)
FS B/16	0.34 (10)	0.59 (8)
FS B/8	0.51 (9)	0.68 (9)
FS L/16	0.34 (19)	0.56 (13)
CLIP B/32	0.19 (12)	0.41 (9)
CLIP B/16	0.35 (9)	0.60 (9)
CLIP L/14	0.38 (17)	0.60 (17)
DINO S/16	0.32 (11)	0.61 (11)
DINO S/8	0.52 (11)	0.73 (12)
DINO B/16	0.32 (12)	0.60 (12)
DINO B/8	0.51 (11)	0.73 (10)
MoCo S/16	0.34 (11)	0.6 (10)
MoCo B/16	0.33 (12)	0.61 (11)
MAE B/16	0.29 (12)	0.54 (12)
MAE L/16	0.31 (24)	0.55 (23)
MAE H/14	0.36 (30)	0.59 (30)
BEiT B/16	0.31 (7)	0.58 (9)
BEiT L/16	0.36 (17)	0.61 (15)

in the mid-to-late layers of the FS and CLIP models. These patterns are strongly repeated across the head and layer axes. Note that the specific token positions that give strong activations are different for the different input images.

Figure 13 shows one ViT attention map per model for 10 sample images over a wide array of ViT variants. The final row displays the average attention over 5000 images. The head selected is the first head of the final layer of each ViT. For the explicitly supervised models, FS and CLIP, we again see mainly Sparse Repeating Attention Patterns. However, for the models with patch size 32, we also see some attention on object-centric regions. This holds true for FS S/32, FS B/32, and CLIP B/32. Because these models use larger patches, their token grids are a quarter of the size, making these models four times narrower than the models with patch size 16. The fact that Sparse Repeating Attention Patterns do not emerge as strongly for these smaller models may suggest that they are an indicator of overfitting in ViTs. For the other FS and CLIP models, we sometimes see faint traces of salient objects highlight in the attention maps, but this occurs alongside the Sparse Repeating Attention Patterns. For DINO, MoCo, and MAE, all models produce attention maps that tend to align well with the salient object. For BEiT, the attention maps do not correlate well. As we previously noted, the final layers of BEiT must serve as a built-in decoder, which may explain why its final layers are dissimilar to DINO, MoCo, and MAE.

We find that every model learns instances of Offset Local Attention Heads, and some larger models even have ones with a diagonal offset. We present one example per model in Figure 14, but be aware that all models have many Offset Local Attention Heads with different offsets. For the explicitly and contrastively supervised models, Offset Local Attention Heads typically only occur in the first 3 to 6 layers, but for the reconstruction-based models, MAE and BEiT, we can find them in deeper layers too.

To provide the reader a complete view of the size and number of attention heads in each ViT, we present two plots that visualize all layers and all heads of all 20 ViTs. Figure 17 presents the average CLS token attention over 5000 sample images. When viewed this way, it is clear how widespread the Sparse Repeating Activation Patterns are over all of the mid-to-later layer heads of the FS and CLIP models. For FS S/32 and B/32 we can see clear signs of Sparse Repeating Activation Patterns, but for CLIP B/32 we instead see far more centered circular blobs, similar to those we observe in the later layers of DINO and MoCo. We also note that some of the early-layer heads (layers 1-3) of the DINO models produce semi-repetitive grid-like patterns that somewhat resemble the Sparse Repeating Attention Patterns seen in CLIP and FS. However, on closer inspection, we believe these heads represent a different phenomenon. The attention patterns in these layers have more variations across heads and layers, and they are not identically repeated as is seen in the FS and CLIP models. Also, these heads come in the early layers, not the mid-to-late layers. For this reason, we hypothesize these heads are learning to extract an initial sparse down-sampling of the image, which would be especially beneficial for the DINO models with patch size 8 due to their much larger token counts. Finally, Figure 18 shows the Aligned Aggregated Attention Maps for all the spatial tokens. This view highlights the great variety of local attention heads used in each ViT. These figures are best viewed digitally and in color.

C.2. Attention Distance for ViT Variants

In Figure 15 we present the Average Attention Distances per-layer for the full ViT collection, broken out by supervision type. For our distance computations, we have normalized the distances such that the token grids are within a 1×1 square, which allows us to compare models with different patch sizes and hence different token grid sizes. We see that the trends of local-vs-global processing order is consistent within supervision groups. For FS, CLIP, DINO, and MoCo we again see an intial high distance, a dip to lower distances, and an increase again in the later layers. Meanwhile, for MAE we again see lower attention distance in the later layers. This result shows that the order of local-vs-global information processing in a ViT is primarily impacted by the method of supervision and is largely unaffected by changes in architecture size and patch size.

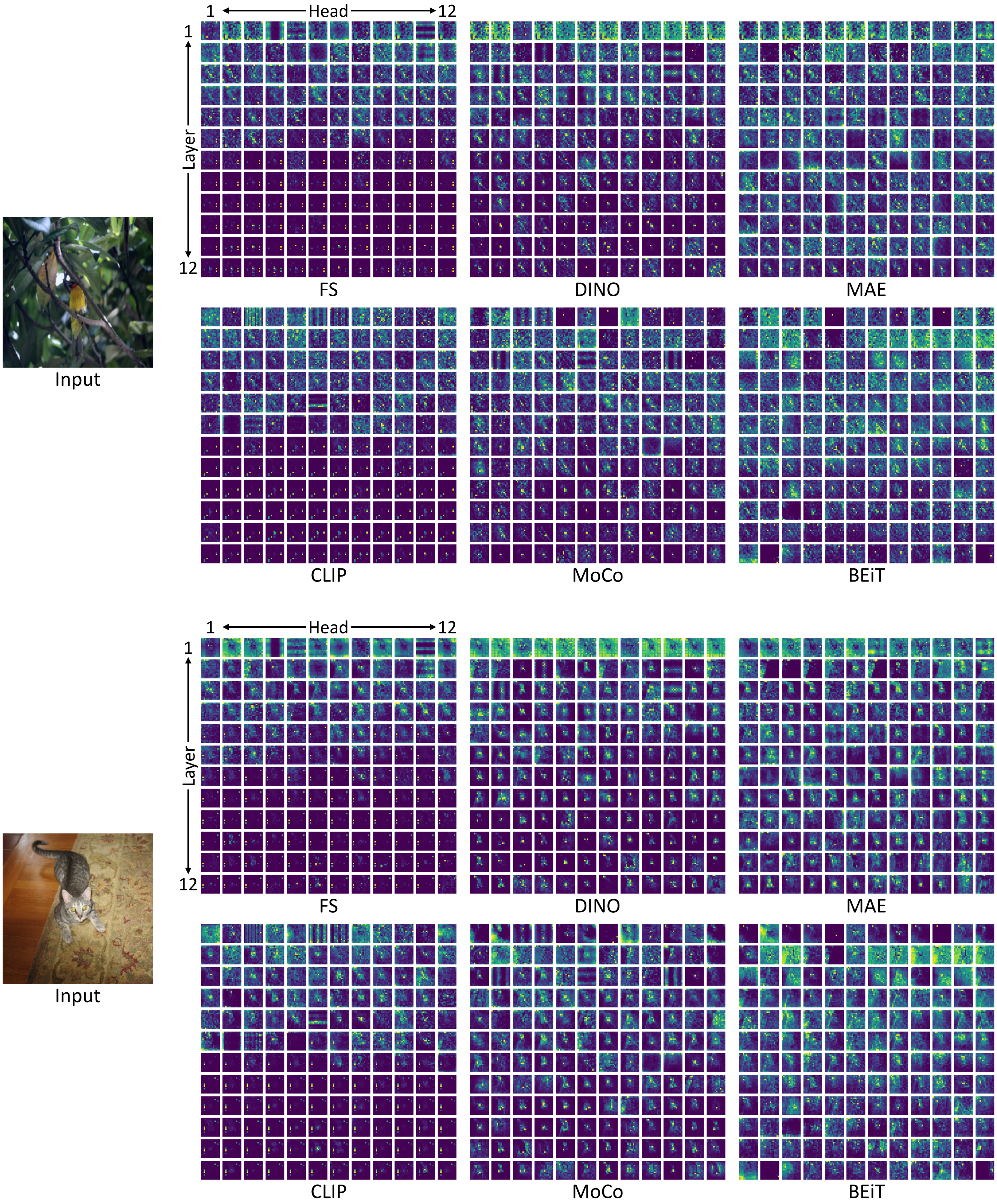

Figure 11. Visualizing all CLS token attention maps for all heads and all layers in ViT B/16 models for single input images (left). The FS and CLIP models show Sparse Repeating Attention Patterns in the mid-to-late layers, where a small group of spatial token positions at seemingly arbitrary positions have strong and consistent activations shared across both heads and layers. Note that the positions of these strong repetitive activations are different for the two inputs.

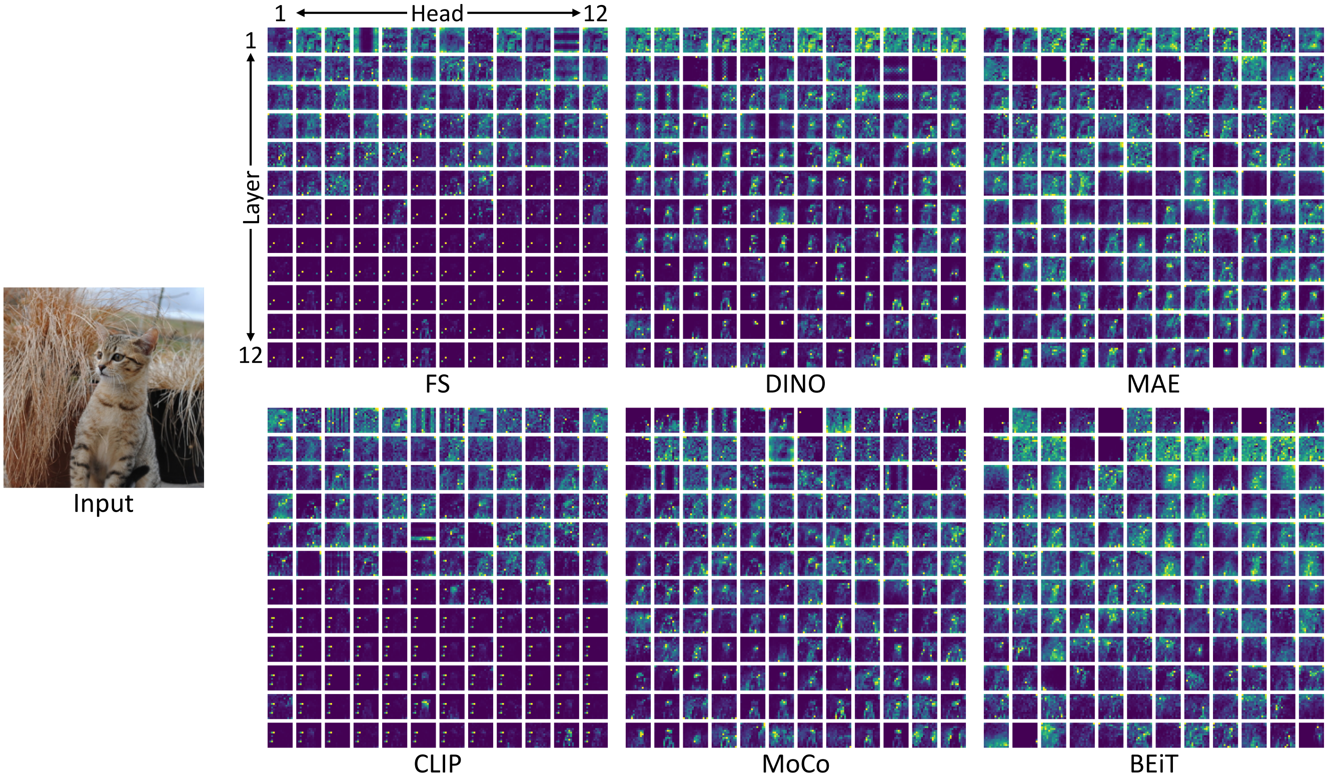

Figure 12. Visualizing all CLS token attention maps for all heads and all layers in ViT B/16 models for a single input image (left).

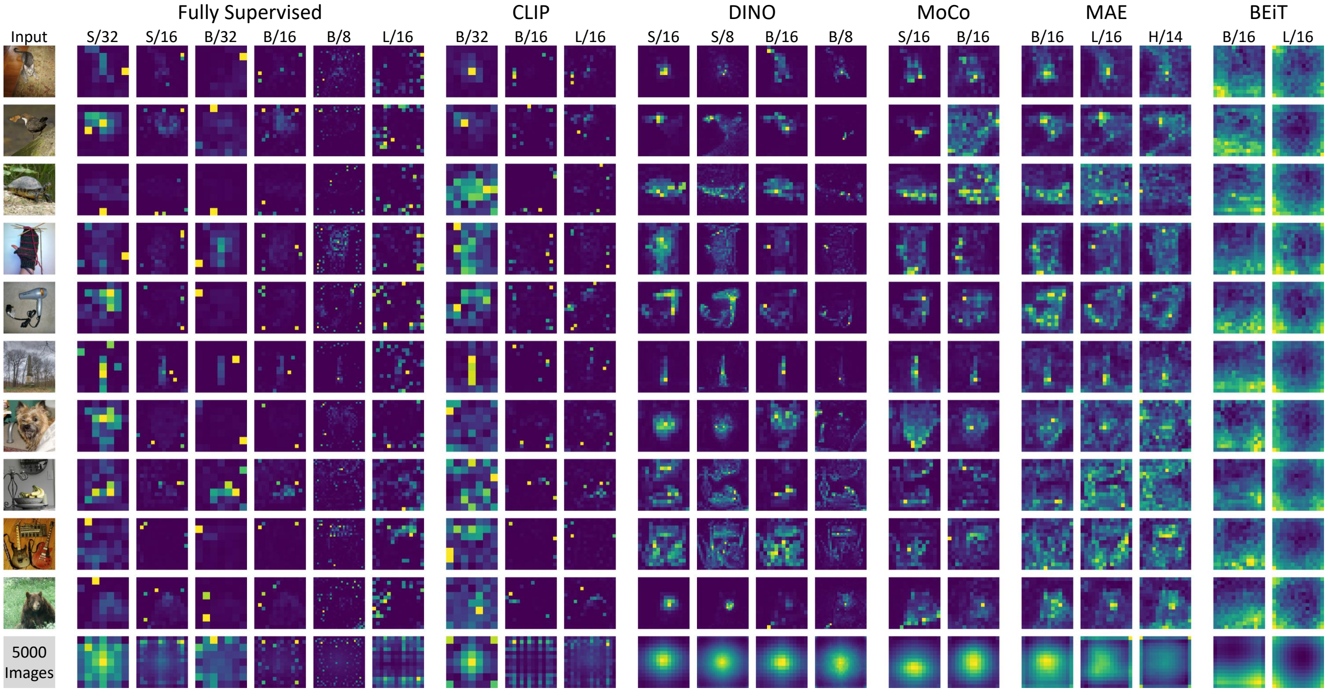

Figure 13. **Sample CLS token attention maps for a wide range of ViT variants.** For each ViT and input image, we show the the attention map of the CLS token of the first head of the final layer. The bottom row shows the averaged activation over 5000 ImageNet images.

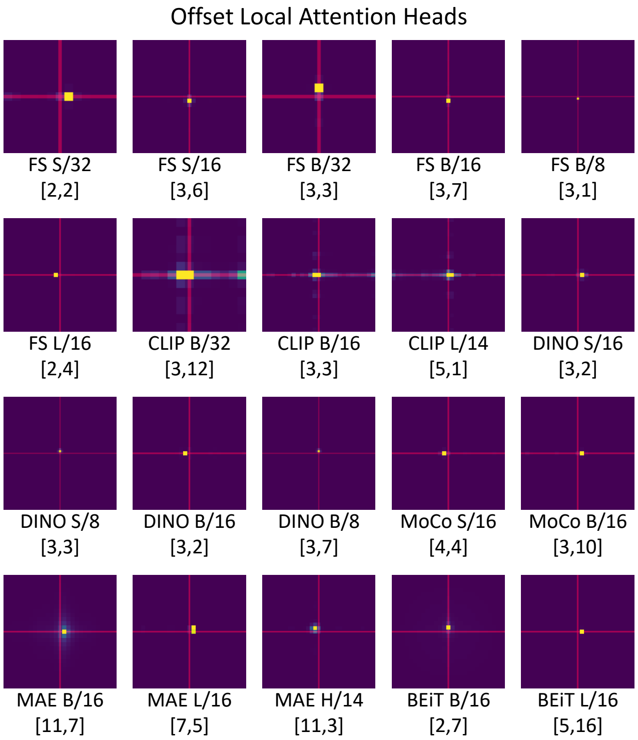

Figure 14. **Examples of Offset Local Attention Heads in all ViT Variants.** We find that all ViTs examined learn to use Offset Local Attention Heads, and some larger models even use ones with diagonal offsets. Midlines are drawn in red as a visual aid.

C.3. Attention Salience for ViT Variants

In this section, we present additional experimental details for our Attentional Saliency Analysis, followed by results for the full ViT collection. In addition to PartImageNet [25], we also performed this analysis with COCO [36], however the results are extremely similar for the two datasets. The PartImageNet dataset contains 11 superclasses, whose members contain similar part structures (biped, quadruped, car). When sampling from PartImageNet, we take 500 samples per superclass, or all samples for ones with less than 500. Within superclasses, we evenly sample from each of the subclasses, or if a subclass is fully sampled we continue to sample evenly from the remaining classes. This yields a mostly balanced collection of 5294 images. PartImageNet divides different subclasses into the train, validation, and test partitions, but for our analysis we work with all three partitions together. For COCO, we simply sample the first 5000 images of the 2017 validation set.

Figure 16 summarizes the results for both PartImageNet and COCO with both CLS token attention and average spatial token attention. The patterns of scores are very similar for PartImageNet and COCO. We see that the explicitly supervised methods again face a decrease in IoU in the mid-to-late layers with the emergence of Sparse Repeating Attention Patterns. This is with the exception of CLIP B/32 which has a much better IoU in the later layers. This result matches Appendix C.1, where we observed that CLIP B/32 is less impacted by the Sparse Repeating Attention Patterns. For the other self-supervised methods, we see that the same general trends hold, usually with slightly higher IoUs from larger models or models with smaller patches.

D. Feature Analysis

D.1. The CKA metric

In our Feature Analysis section, we compare learned representations through Centered Kernel Alignment (CKA) [14, 31], which is able to align and rescale neural features to enable similarity measurements. Specifically, we used batched CKA [42], which can be represented as follows:

$$CKA_{batched} = \frac{\frac{1}{k}\sum_{i=1}^{k} HSIC_1(\mathbf{X}_i\mathbf{X}_i^T, \mathbf{Y}_i\mathbf{Y}_i^T)}{\left(\begin{array}{c}\sqrt{\frac{1}{k}\sum_{i=1}^{k} HSIC_1(\mathbf{X}_i\mathbf{X}_i^T, \mathbf{X}_i\mathbf{X}_i^T)} \\ * \sqrt{\frac{1}{k}\sum_{i=1}^{k} HSIC_1(\mathbf{Y}_i\mathbf{Y}_i^T, \mathbf{Y}_i\mathbf{Y}_i^T)}\end{array}\right)}, \quad (1)$$

where X_i and Y_i are the feature representation matrices of the i^{th} batch from the two models, k is the number of batches and $HSIC_1$ is as follows:

$$HSIC_1(\mathbf{K},\mathbf{L}) = \frac{1}{n(n-3)}\left(tr(\hat{\mathbf{K}}\hat{\mathbf{L}}) + \frac{\mathbf{1^T\hat{K}11^T\hat{L}1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1^T\hat{K}\hat{L}1}\right), \quad (2)$$

where $\hat{\mathbf{K}}$ and $\hat{\mathbf{L}}$ are $\mathbf{K}, \mathbf{L}$ with diagonals set to 0 and n is the batch size. We use the implementation from Subramanian [59] for our analysis.

D.2. Last Layer Comparisons for ViT Variants

In the main paper we analyzed the last layer CKA between the CLS and spatial tokens across the B/16 models separately. Here we expand our analysis to the wider collection of ViT variants. As can be seen from Figure 19 (left), for the CLS tokens, similar supervision strategies create similar representations. Groups emerge with DINO and MoCo forming one subset while MAE and BEiT form another. The FS and CLIP models form their own sub-groups. Some of the FS models also show comparatively high similarity with MoCo and DINO models. Again we see that the MAE CLS representations have a moderate similarity with explicitly and constrastively supervised methods.

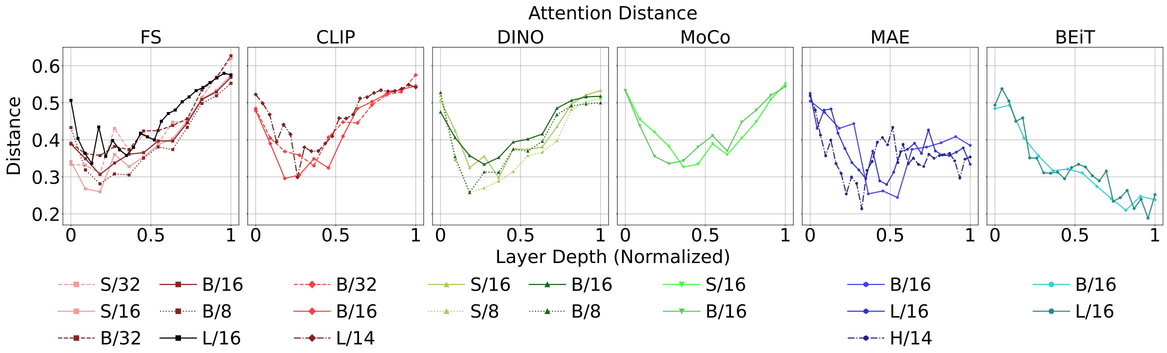

Figure 15. **Average Attention Distance for all ViT Variants.** Results are plotted against the normalized layer depth.

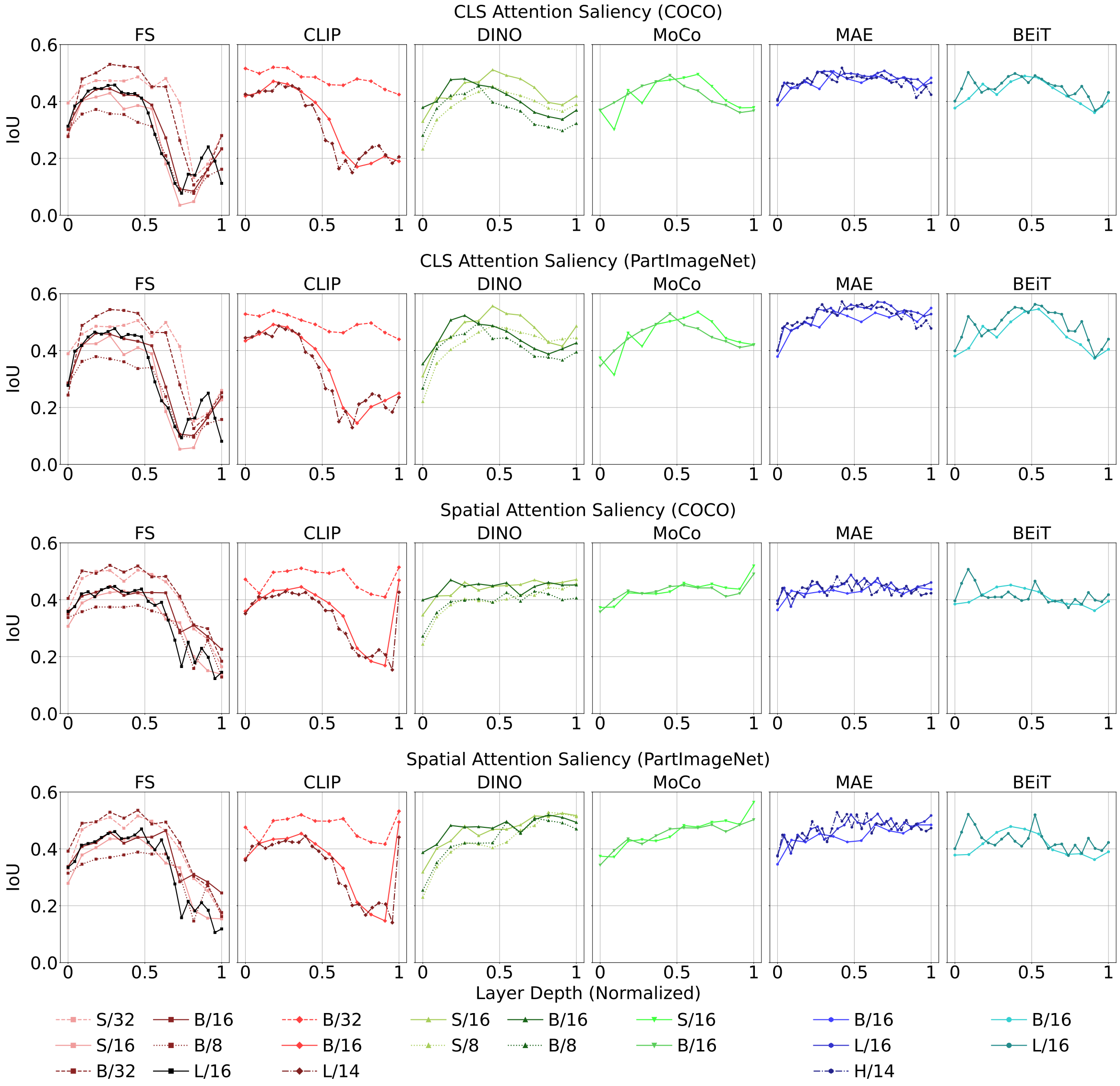

Figure 16. **Attention alignment with salient image content for all ViT Variants.** Results are shown for both CLS token attention and average spatial token attention on both COCO and PartImageNet. We see that the results are highly similar for the two datasets.

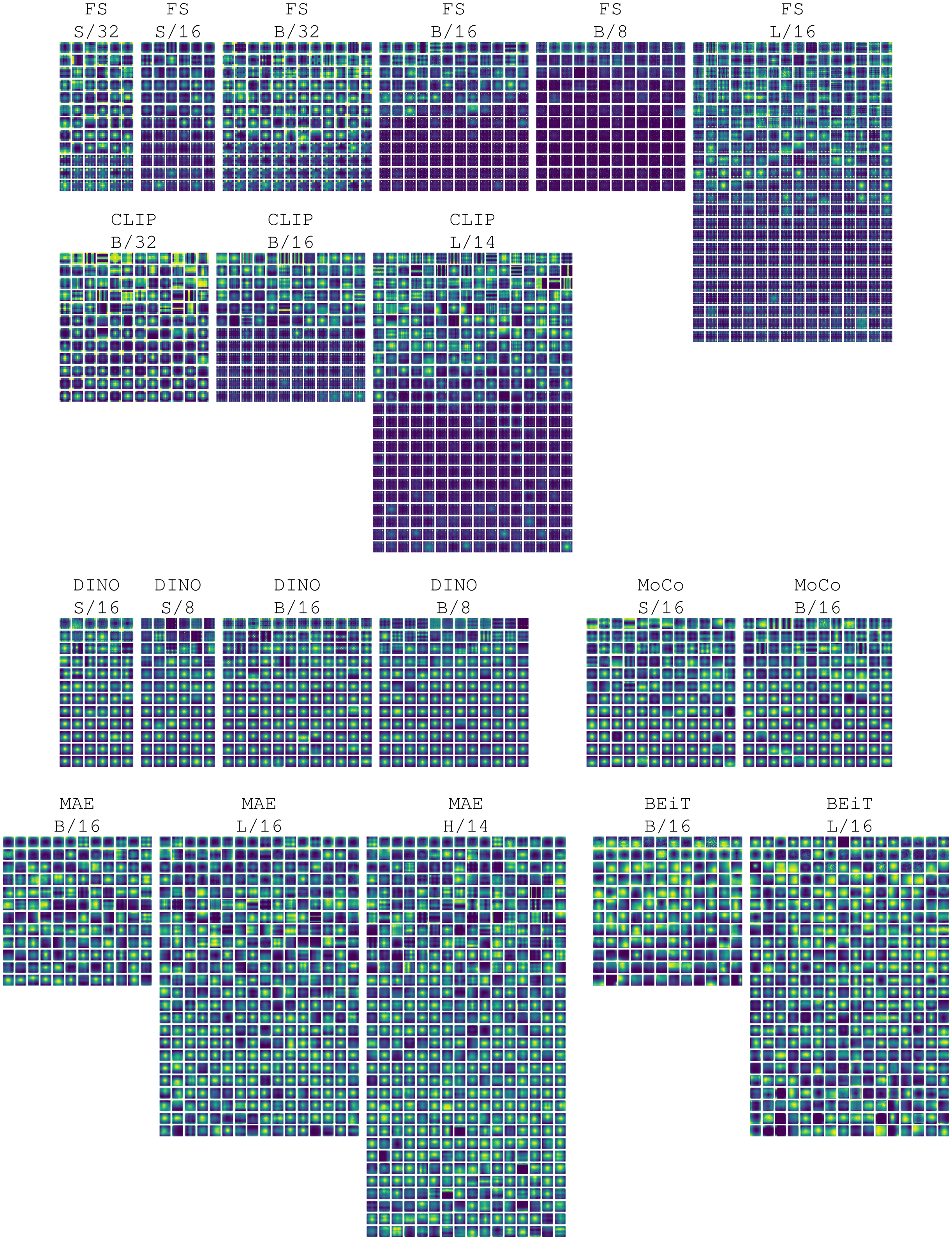

Figure 17. Average CLS token attention over 5000 images for every head of every ViT Variant.

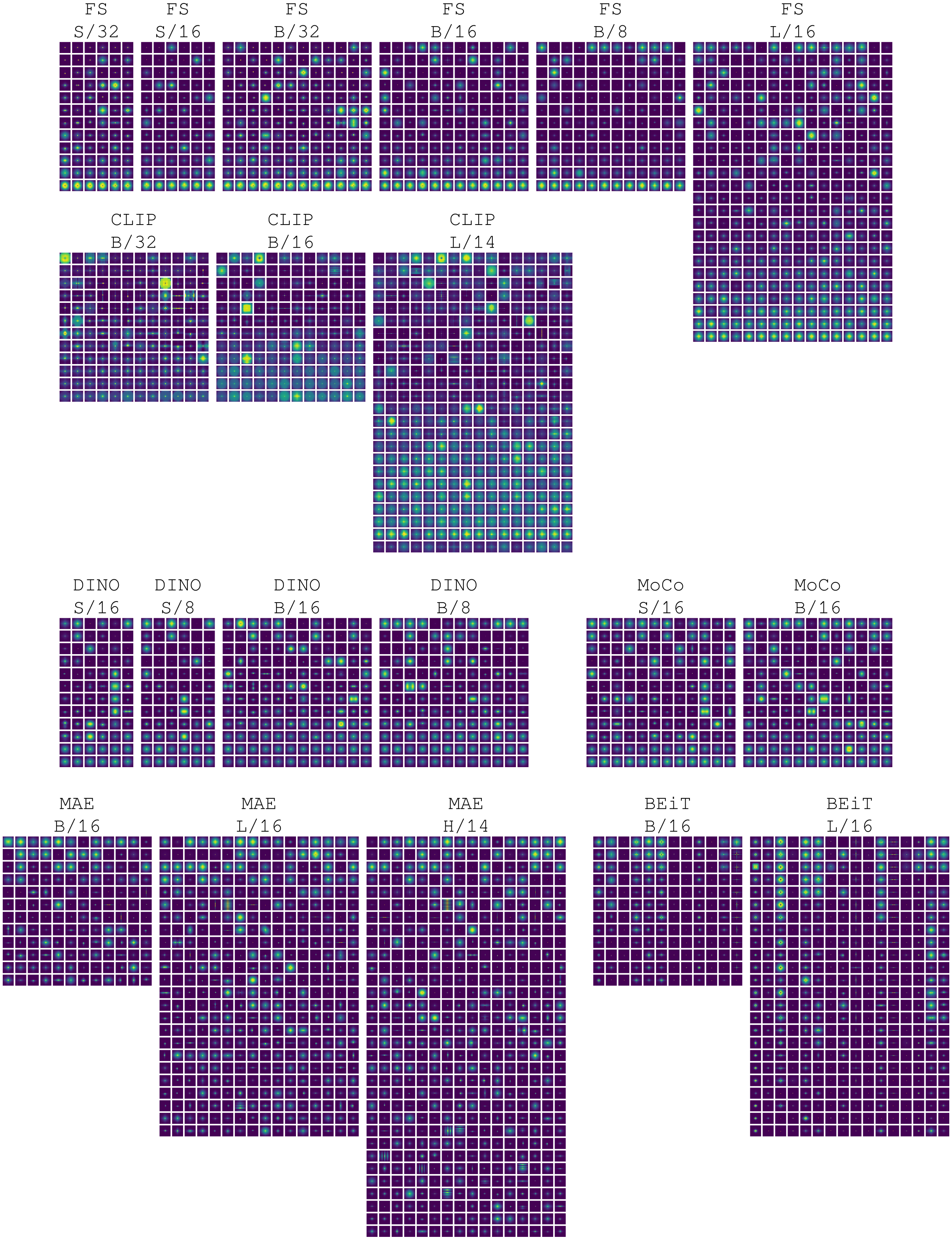

Figure 18. Aligned Aggregated Attention Maps over 5000 images for every head of every ViT Variant.

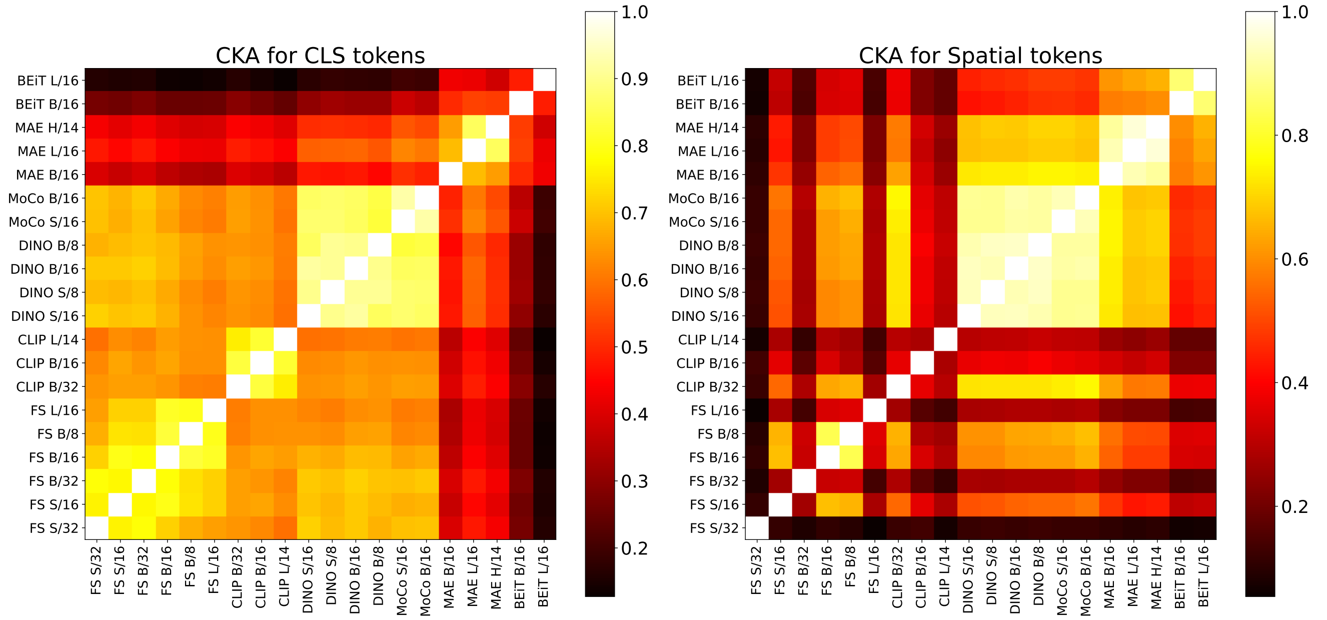

Figure 19. CKA similarity between final layer features of different ViTs for their CLS tokens (left) and spatial tokens (right).

From Figure 19 (right), we see that the internal similarity within the FS and CLIP groups are more fragmented. Meanwhile, the self-supervised models show more consistency, having higher spatial feature similarity within and between self-supervision methods. DINO and MoCo show very high similarity amongst themselves due to their similar training methods. The MAE spatial features also show high similarity to those of DINO and MoCo. BEiT shows comparatively high similarity with these other self-supervised methods. FS and CLIP are not too similar with each other or with the other models with the exception of CLIP B/32 and a few FS models like B/16 and B/8 which show comparatively high similarity with DINO and MoCo models. This separation of CLIP and FS can be attributed to the supervision which is applied to only the CLS token, which may make their final layer spatial representations less consistent.

D.3. Depth-Wise CKA Analysis

Self-Comparison of ViT B/16 Models. Figure 20 shows the CKA plots across multiple layers of the same model for different training methods. For brevity and consistency we focus on the ViT B/16 models. For all the CKA plots we use the features from the batch norm layers due to their well-behaved outputs. We see that the different models show variations in the development of information. For FS, the final layers have a lower similarity to the earlier layers, as compared with CLIP, DINO, and MoCo. For MAE, we see two distinct blocks of similarity divided around the middle layer. For BEiT, we see a clear X pattern, which we analyze more in a subsequent section.

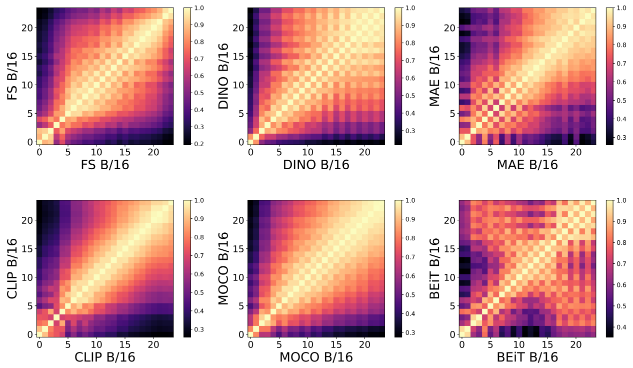

Figure 20. CKA for all ViT B/16 models showing their feature similarity across layers. Different supervision techniques result in different patterns.

Going deeper into MAEs. Figure 21 shows the CKA plot for MAE models Base, Large and Huge from left to right. It can be seen that as we move from a smaller model to a larger model (for example from Base to Large), the bottom left quadrant of larger models CKA matches the full CKA for the smaller model. This indicates that a larger model in this case encodes information in a similar way as the smaller model in its initial layers but ends up having more specialized later layers at the end. A similar trend can be observed when going from Large to Huge.

X pattern for MAE and BEiT. As shown in Figure 21 and Figure 22, MAE and BEiT show an X-like pattern in their CKA plot (with the exception of MAE B/16). This

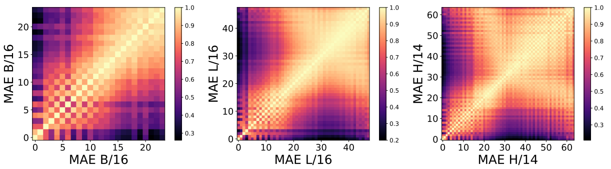

Figure 21. CKA across Base, Large and Huge MAE models.

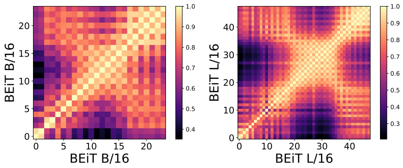

Figure 22. CKA across Base and Large BEiT models.

indicates that the late layer features of these models are similar to the early layers but not the middle layers. We hypothesize that this is due to the reconstruction-oriented nature of the training losses, showing that in their final layers MAE and BEiT are trying to recreate the same local information that is present in the initial layers. It should be noted that this X-pattern arises for all sizes of BEiT, but not in MAE B/16. We attribute this to the fact that MAE has a separate decoder module which is discarded after training, while BEiT does not have a separate decoder. This means that BEiT needs to inherently learn a decoder which leads to emergence of this X pattern at both sizes. For MAE, the fact that the X pattern emerges more clearly for the larger ViT variants suggests that the additional layers of these models start specializing for the task of decoding. This has important implications for the MAE training method, as it suggests that, for larger MAEs, late layers learn to act in a more decoder-like way, which may limit the usefulness of these layers for downstream tasks.

D.4. Residual Connection Analysis

Previously works [52] have contrasted CNNs and ViTs by comparing the features propagated through the skip connections and normal connections. We extend this analysis to ViTs trained with varying supervision techniques. Figure 23 shows the CKA between the features coming from the skip connection (Y-axis) and the normal pathway (X-axis) for the MHA layer of each block. For CLIP and FS we can see a similar trend, initially the skip connection carries similar information as the normal path but after a certain point the information it carries becomes very different (dark regions). This also correlates with the emergence of the Sparse Repeating Patterns observed in Appendix C.1, providing further evidence that a fundamental shift in information processing behavior occurs in the mid-to-late layers of explicitly supervised ViTs. For MoCo and DINO, this shift in behavior does not happen and as the depth increases the skip connections and normal pathways still have similar representations. MAE is another special case where, given the reconstruction nature of the loss, at multiple depth locations the normal pathway representation is similar to the skip connection representation. For BEiT, there is again an X-like pattern, likely because it needs to start reconstructing the complete input. MAE does not show an X-like pattern, despite its similar reconstruction objective. Again we theorize that this occurs because MAE has a separate decoder that is discarded after training.

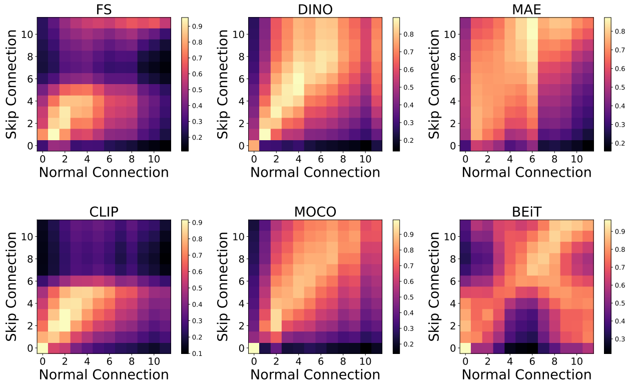

Figure 23. **Residual connection analysis.** We show the CKA similarity between the features coming from the skip-connect (Y-axis) and normal pathway (X-axis) for each MHA layer of each block. Each cell indicates the similarity between the skip connection features and output of normal pathway at that location.

D.5. Additional Clustering Analysis

In this section, we expand our feature clustering analysis to the full collection of models. In addition to Cluster Purity, we also report results for Normalized Mutual Information (NMI) and Adjusted Random Index (ARI). We see similar trends for all three clustering metrics.

Image-Level CLS Feature Clustering Results shown in Figure 24. For FS, CLIP, DINO, and MoCo the same general trends hold. Cluster quality rises faster for DINO and MoCo, but FS and CLIP catch up rapidly and overtake at the end. For the deeper model variants (FS L/16 and CLIP L/14) the trends are consistent when plotted against normalized block depth, meaning that semantic information actually emerges half as quickly. For MAE, the larger model variants lead to significantly better cluster purity, which also rises earlier as the models get larger. In contrast, for BEiT cluster purity is generally worse for the larger L/16 model.

Image-Level Spatial Feature Clustering Results shown in Figure 25. For FS, CLIP, DINO, and MoCo the same general trends show, though with larger models tending to do slightly better. Interestingly, the spatial features of BEiT L/16 have an increase in cluster purity, which is surprising as its CLS tokens saw a decrease in the previous section.

Object-Level Spatial Feature Clustering Results shown in Figure 26. For FS, CLIP, DINO, and MoCo again the general trends hold, with larger models or those with smaller patch size doing slightly better. However, for FS the L/16 variant did worse than B/16. For FS the best scores are achieved by the B/8 variant. Like the previous section, we see a significant boost for BEiT L/16 over B/16.

Part-Level Spatial Feature Clustering Results shown in Figure 27. For part-level feature clustering, we previously observed that, for the B/16 models, the self-supervised methods are much more competitive with the explicitly supervised methods. This trend still holds here, with BEiT L/16 performing particularly well, seeing a huge boost over BEiT B/16. For all metrics, BEiT L/16 is on par with the best explicitly and constrastively supervised methods. Larger models and those with smaller patch size generally provide better part-level feature clusters, with the exception of MAE, where the large models actually do worse.

E. Downstream Task Analysis

E.1. Keypoint Correspondence Additional Details

As part of our Downstream Task Analysis, we present results for Keypoint Correspondence as an additional local-focused task. Given an input image with a set of human annotated keypoints, a model must predict the position of corresponding keypoints in a second paired image with the same type of object. Challenges in this task include changes in scale, size, and large intraclass variations. Correspondence is a prerequisite step in applications such as pose estimation [54], 3D reconstruction [54], and edit propagation in images and videos [71]. We use the SPair-71k [39] dataset consisting of 1800 images from 18 categories. Following the evaluation protocol used by Amir et al. [1], we randomly sample 20 image pairs from each category of the test set and compute PCK [72] (percentage of correct keypoints) for each of the 18 categories. Given the dense ViT spatial token features of a source image, a target image, and source keypoint, we (1) get the corresponding feature vector of the keypoint in the source image, (2) find the nearest neighbor of this feature vector in the target image, and (3) get the 2D location of the nearest neighbor in the target image. The keypoint prediction is considered correct if it is within, a threshold $\alpha \cdot \max(H, W)$ of the groundtruth correspondence, where α is a constant and (H, W) are the height and width of the target image. We report the average PCK@0.1, PCK@0.05, and PCK@0.01.

E.2. Results for ViT Variants

ImageNet Classification As seen in Figure 28, the general trends as reported in the main paper for this task hold for each training method. The FS B/8 has the highest Top-1 and Top-5 performance. For FS, CLIP, DINO and MoCo the trends show that performance improves as we go to later layers. It is also interesting to see that under normalized depth, the larger MAE models peak earlier than the smaller ones and show a higher peak performance. For BEiT, the peak performance occurs at a similar relative depth but is higher for the L/16 model.

Image Retrieval As shown in Figure 29, the Base and Large models for FS and CLIP perform well for this image-level task. This aligns with our observations in the main paper. The FS B/16 performs the best on ROxford5k while the FS B/8 and L/16 are the best performers on RParis6k. For the FS, CLIP, MoCo and DINO models, performance improves as we go to later layers in most cases. MAE and BEiT again peak early in the mid-to-late layers. The general observations and trends for this task are similar to the k-NN task as they are both image-level global tasks. The only difference being that, for this task, the later layers of FS and CLIP show a sudden improvement while the earlier layers are flatter when compared to the trends for k-NN.

DAVIS Segmentation Propagation As shown in Figure 30 many models peak at an earlier layer. The best performance comes from DINO B/8 and S/8, followed closely by FS B/8. Meanwhile, the models with patch size 32 see a significant drop in performance. Given the dense-prediction-based nature of this task, these methods which are trained with a smaller patch size have finer features which give them a boost in performance. The reconstruction-based models, BEiT and MAE, are also very competitive in this task, performing on par with FS, CLIP, DINO, and MoCo among the models with patch size 16.

Keypoint Correspondence We show comparisons on this task in Figure 31. It should be highlighted that BEiT L/16 performs the best for PCK@0.1, again showing a massive improvement over BEiT B/16. This large boost correlates with the its improved part-level feature purity observed in Appendix D.5. In general the smaller patch size models like DINO B/8, DINO S/8, CLIP L/14 and FS B/8 are also good performers. Due to their finer feature grids, these methods perform much better at the stricter thresholds (PCK@0.05 and PCK@0.01). All models on this task peak around mid-to-late layers.

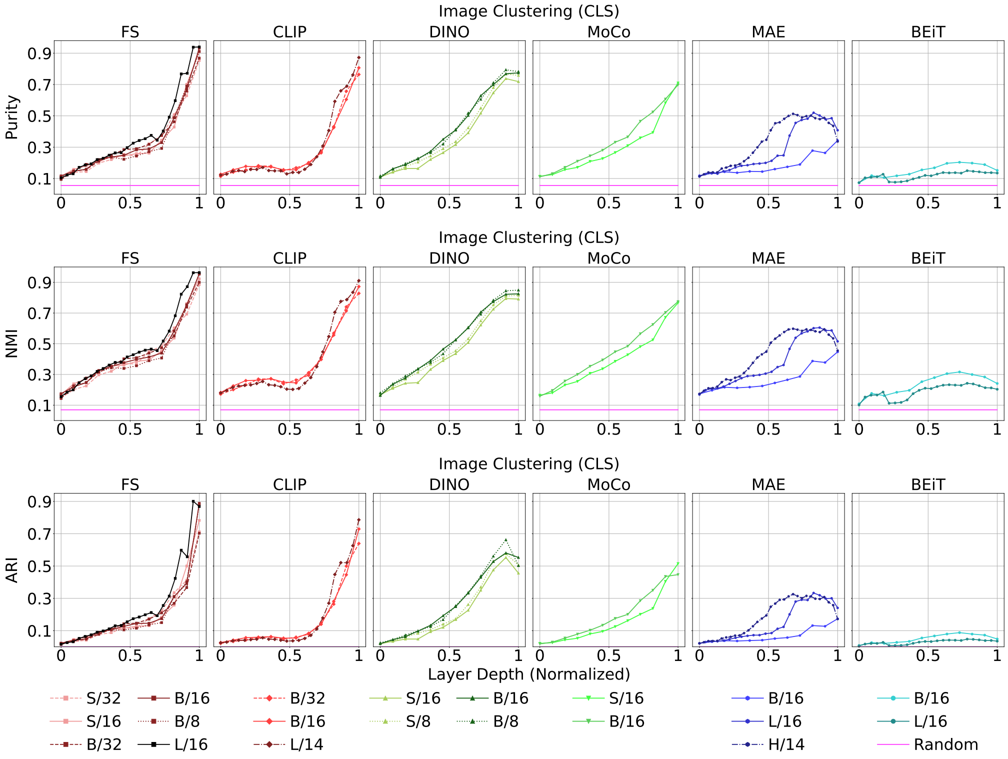

Figure 24. Expanded CLS feature clustering for image-level labels with ImageNet-50.

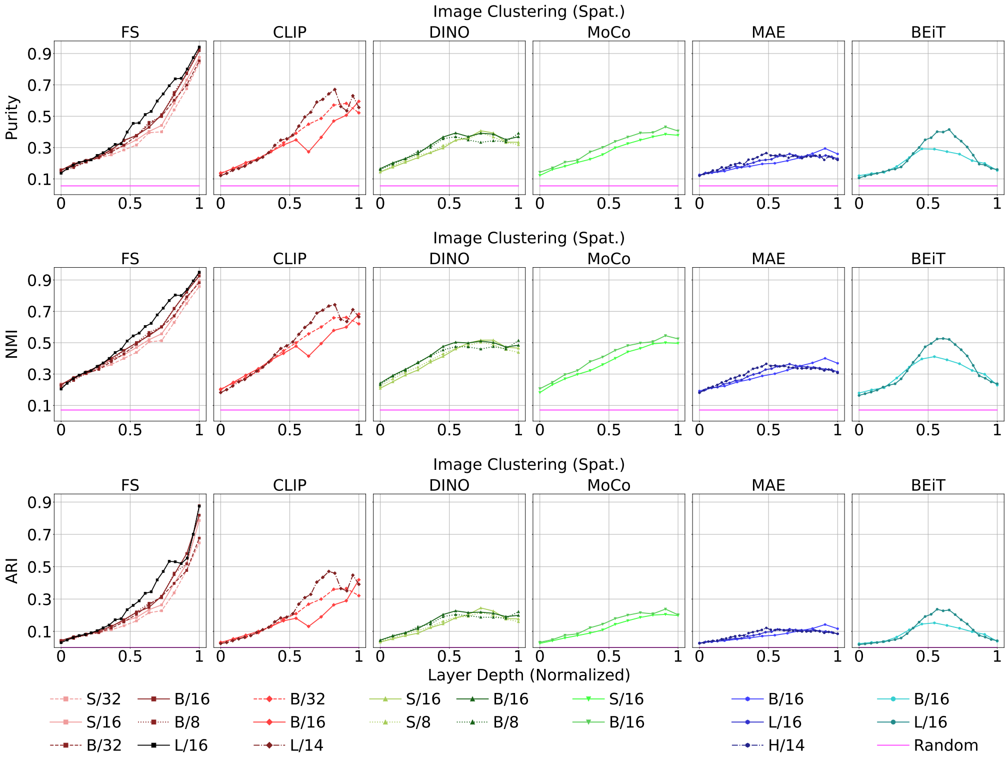

Figure 25. Averaged spatial feature clustering for image-level labels with ImageNet-50.

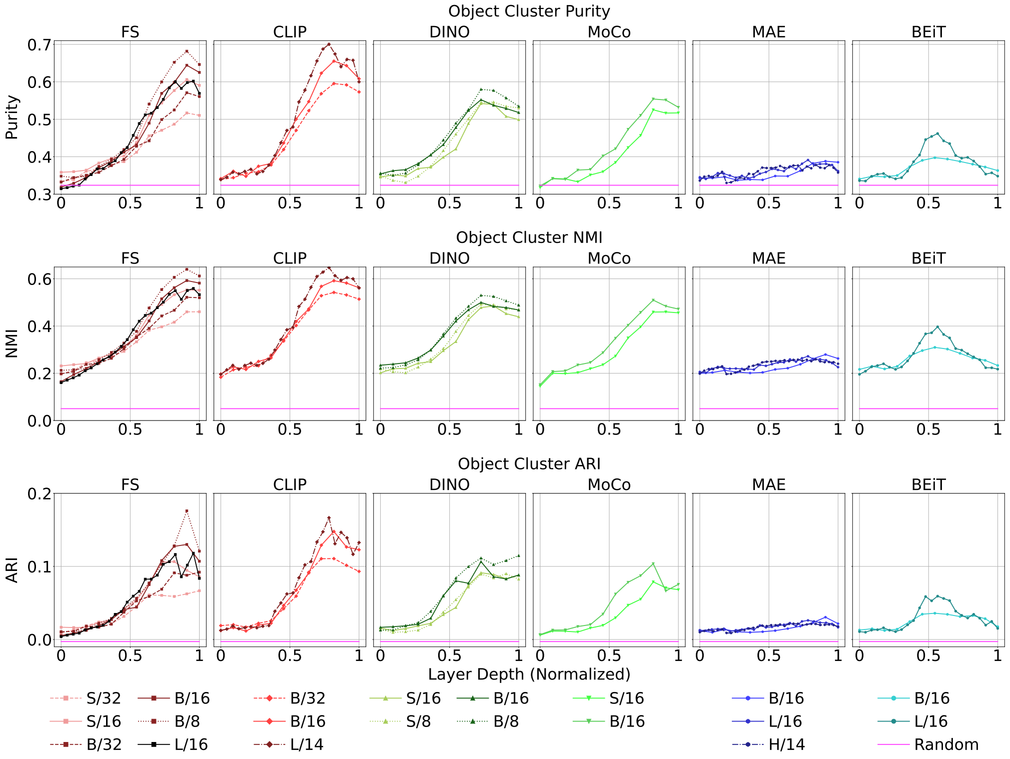

Figure 26. Expanded spatial feature clustering for object-level labels with COCO.

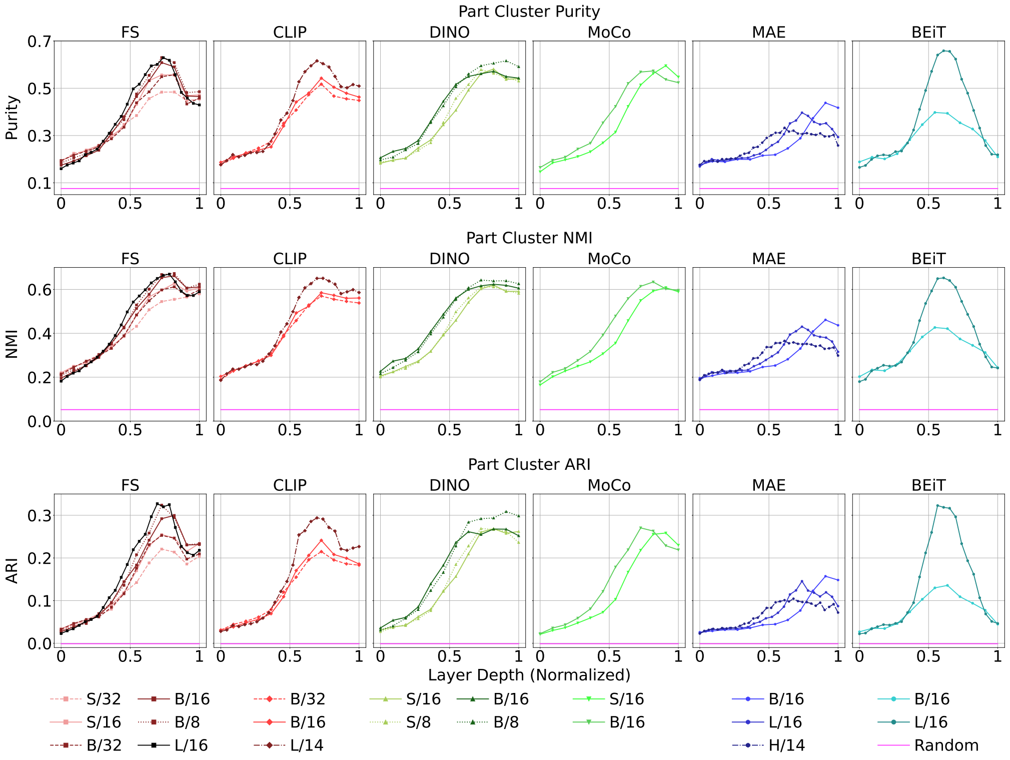

Figure 27. Expanded spatial feature clustering for part-level labels with PartImageNet.

Table 5. Best performance for each ViT on each downstream task with the corresponding best layer in parenthesis.

Model		**Task Performance (Best Performing Layer)**											
Dataset	Layers	ImageNet		ROxford5k		RParis6k		DAVIS			SPair-71k		
Metric		Top-1↑	Top-5↑	mAP↑ (M)	mAP↑ (H)	mAP↑ (M)	mAP↑ (H)	J Mean↑	F Mean↑	J and F Mean↑	PCK@0.1↑	PCK@0.05↑	PCK@0.01↑
FS S/32	12	74.48 (12)	90.33 (12)	0.33 (12)	0.12 (12)	0.63 (12)	0.38 (12)	0.43 (8)	0.35 (9)	0.39 (9)	15.28 (8)	4.34 (8)	0.13 (8)
FS S/16	12	80.64 (12)	93.71 (12)	0.34 (12)	0.1 (11)	0.66 (12)	0.40 (12)	0.57 (8)	0.60 (9)	0.58 (8)	26.37 (8)	11.62 (8)	0.68 (9)
FS B/32	12	79.08 (12)	92.8 (12)	0.33 (12)	0.12 (12)	0.67 (12)	0.43 (12)	0.42 (9)	0.34 (9)	0.38 (9)	15.49 (9)	4.19 (8)	0.15 (7)
FS B/16	12	83.79 (12)	95.01 (12)	0.45 (12)	0.19 (12)	0.72 (12)	0.51 (12)	0.58 (8)	0.60 (8)	0.59 (8)	28.56 (9)	12.33 (8)	0.63 (7)
FS B/8	12	85.58 (12)	95.71 (12)	0.45 (12)	0.16 (12)	0.73 (12)	0.51 (12)	0.66 (7)	0.70 (9)	0.68 (9)	36.09 (9)	21.97 (9)	1.61 (9)
FS L/16	24	85.03 (24)	95.39 (24)	0.42 (24)	0.15 (24)	0.73 (24)	0.51 (24)	0.56 (13)	0.57 (13)	0.56 (13)	30.99 (17)	13.49 (15)	0.79 (14)
CLIP B/32	12	70.90 (12)	89.54 (12)	0.38 (12)	0.10 (12)	0.66 (12)	0.41 (12)	0.44 (9)	0.37 (9)	0.41 (9)	18.55 (8)	5.37 (8)	0.26 (8)
CLIP B/16	12	75.75 (12)	92.27 (12)	0.40 (12)	0.11 (12)	0.71 (12)	0.48 (12)	0.58 (9)	0.62 (9)	0.60 (9)	30.70 (8)	13.61 (8)	0.98 (6)
CLIP L/14	24	80.24 (24)	94.15 (24)	0.45 (24)	0.17 (24)	0.70 (24)	0.49 (24)	0.57 (14)	0.62 (17)	0.60 (17)	36.04 (15)	16.72 (15)	1.15 (13)
DINO S/16	12	74.61 (12)	90.08 (12)	0.38 (12)	0.14 (12)	0.61 (12)	0.33 (12)	0.60 (11)	0.63 (11)	0.61 (11)	26.72 (9)	11.68 (9)	0.55 (9)
DINO S/8	12	74.34 (12)	90.15 (12)	0.36 (12)	0.12 (12)	0.59 (12)	0.30 (12)	0.70 (12)	0.77 (12)	0.73 (12)	31.14 (8)	18.15 (8)	1.48 (8)
DINO B/16	12	76.06 (12)	91.40 (12)	0.37 (12)	0.11 (12)	0.62 (12)	0.35 (12)	0.59 (12)	0.61 (12)	0.60 (12)	28.28 (9)	12.00 (7)	0.65 (6)
DINO B/8	12	77.70 (12)	92.24 (12)	0.40 (12)	0.13 (11)	0.65 (12)	0.37 (12)	0.69 (10)	0.77 (10)	0.73 (10)	33.17 (8)	19.04 (8)	1.66 (5)
MoCo S/16	12	68.71 (12)	86.36 (12)	0.27 (12)	0.07 (12)	0.50 (12)	0.22 (12)	0.58 (10)	0.62 (10)	0.6 (10)	24.88 (9)	10.92 (9)	0.39 (11)
MoCo B/16	12	71.59 (12)	88.37 (12)	0.31 (12)	0.08 (12)	0.51 (12)	0.22 (12)	0.59 (11)	0.62 (11)	0.61 (11)	25.85 (9)	10.64 (8)	0.43 (10)
MAE B/16	12	45.19 (12)	65.32 (12)	0.15 (10)	0.02 (10)	0.28 (10)	0.08 (10)	0.54 (11)	0.54 (12)	0.54 (12)	22.65 (11)	10.59 (11)	0.44 (11)
MAE L/16	24	60.80 (20)	78.9 (20)	0.19 (21)	0.03 (21)	0.35 (21)	0.11 (21)	0.55 (23)	0.56 (23)	0.55 (23)	27.65 (19)	13.02 (22)	0.60 (21)
MAE H/14	32	63.16 (23)	79.87 (23)	0.20 (23)	0.03 (30)	0.39 (23)	0.13 (23)	0.58 (31)	0.61 (30)	0.59 (30)	27.50 (26)	13.65 (26)	1.35 (26)
BEiT B/16	12	26.84 (8)	45.12 (8)	0.14 (8)	0.02 (8)	0.20 (8)	0.05 (10)	0.57 (10)	0.59 (9)	0.58 (9)	24.11 (8)	11.02 (8)	0.54 (7)
BEiT L/16	24	41.24 (18)	62.79 (18)	0.16 (18)	0.02 (18)	0.25 (17)	0.06 (17)	0.58 (17)	0.64 (15)	0.61 (15)	37.52 (15)	18.22 (16)	1.04 (16)
Random	-	0.10	0.49	0.02	0.01	0.04	0.03	0.03	0.08	0.06	1.32	0.34	0.02

E.3. Summary of Downstream Tasks

We report the best result for each model along with the layer at which it occurs in Table 5. This table captures all downstream tasks and summarizes all metrics for each task. We would like to highlight that **[1]** different models peak at different layers, based on type of task, local vs. global, and **[2]** no one model is the best model for all tasks.

E.4. ImageNet-1k Classification with Linear Probes

We present additional results for ImageNet classification with Linear Probes in Table 6. For each model, we trained a linear layer on the last layer features for 20 epochs on the ImageNet-1k training set, and report results on the validation set. For BEiT we instead use layer 8, which gave the best k-NN classification results. This analysis includes variations based on protocols from the compared works. We present results for CLS token features in row 2 and average-pooled spatial token features (proposed by BEiT) in row 3. We also test the addition of a batch normalization layer before the linear layer (proposed by MAE) in rows 4 & 5. As FS is trained with ImageNet labels, it performs best in all settings. For approaches with explicit CLS supervision (FS, CLIP, DINO, MoCo), the CLS token features give higher accuracy. For MAE and BEiT, due to the local nature of their supervision, their spatial features give better performance. Batchnorm is generally beneficial for all models and features.

Table 6. Accuracy@1 of ViT-B/16 models for Linear Probing on ImageNet-1k val. *required reduced LR for stable training.

Feature	FS	CLIP	DINO	MoCo	MAE	BEiT
CLS	83.86	65.63	73.03	74.26	49.52	9.87*
Spat.	82.31	52.53	37.37	62.47	52.01	29.81
CLS+BN	**84.40**	**78.63**	**76.39**	**74.51**	59.31	41.68
Spat.+BN	82.83	74.59	68.09	68.58	**59.86**	**44.27**

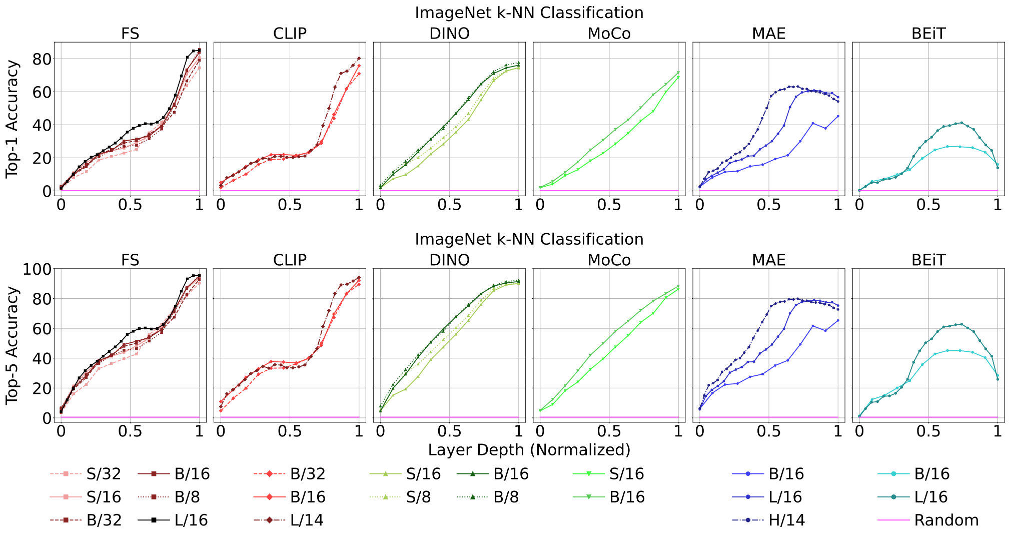

Figure 28. k-NN ImageNet classification results for all ViT variants.

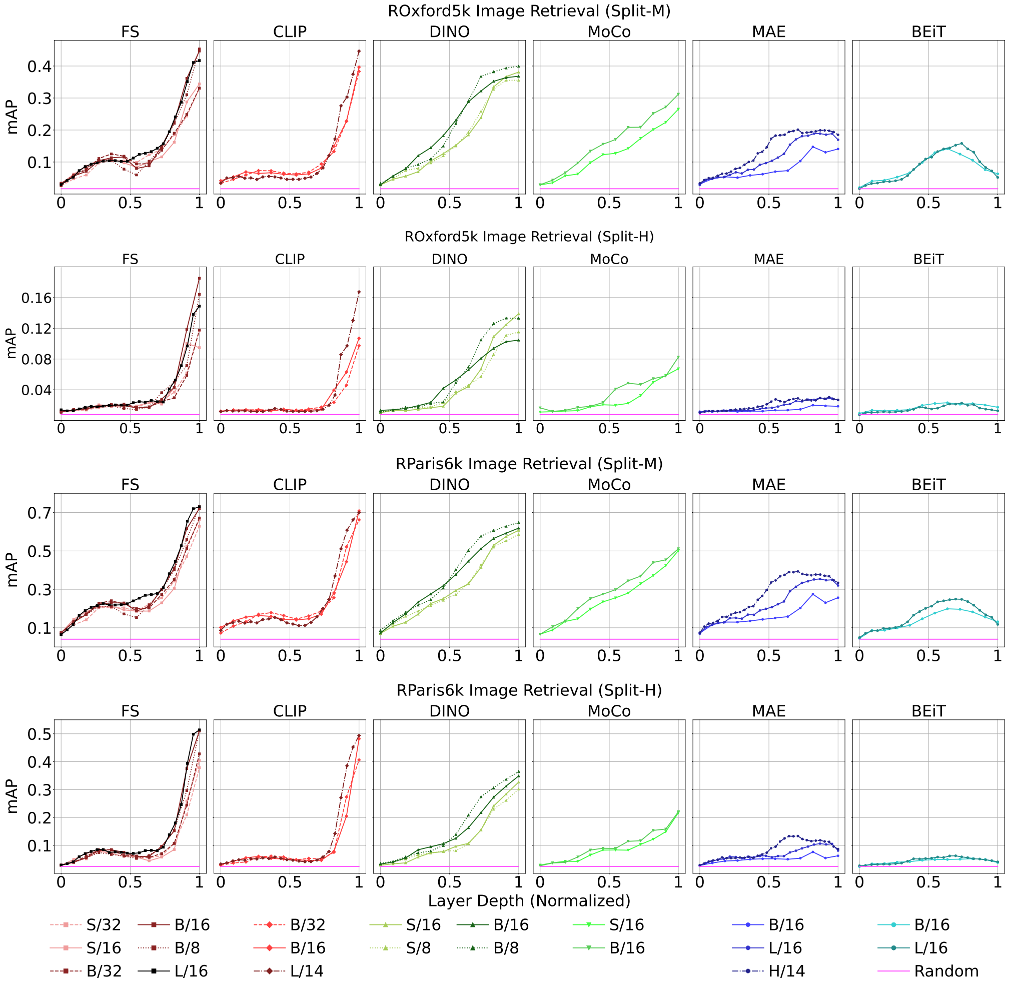

Figure 29. ROxford5k and RParis6k retrieval results for all ViT variants.

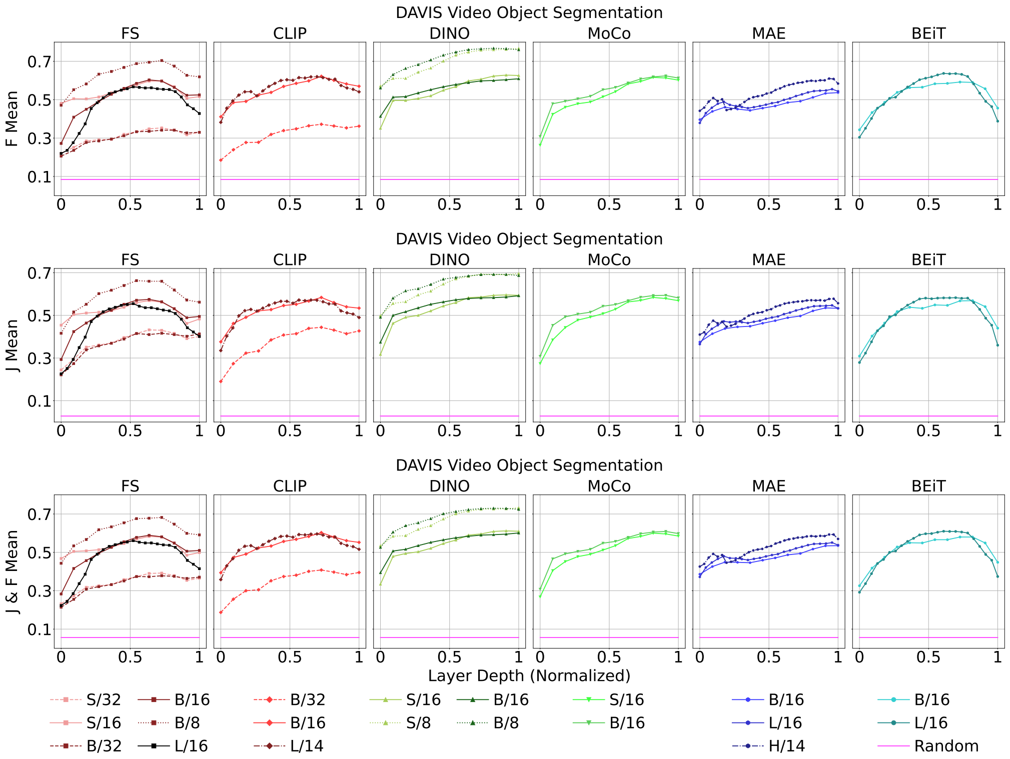

Figure 30. DAVIS Video Segmentation Propagation comparison for all ViTs

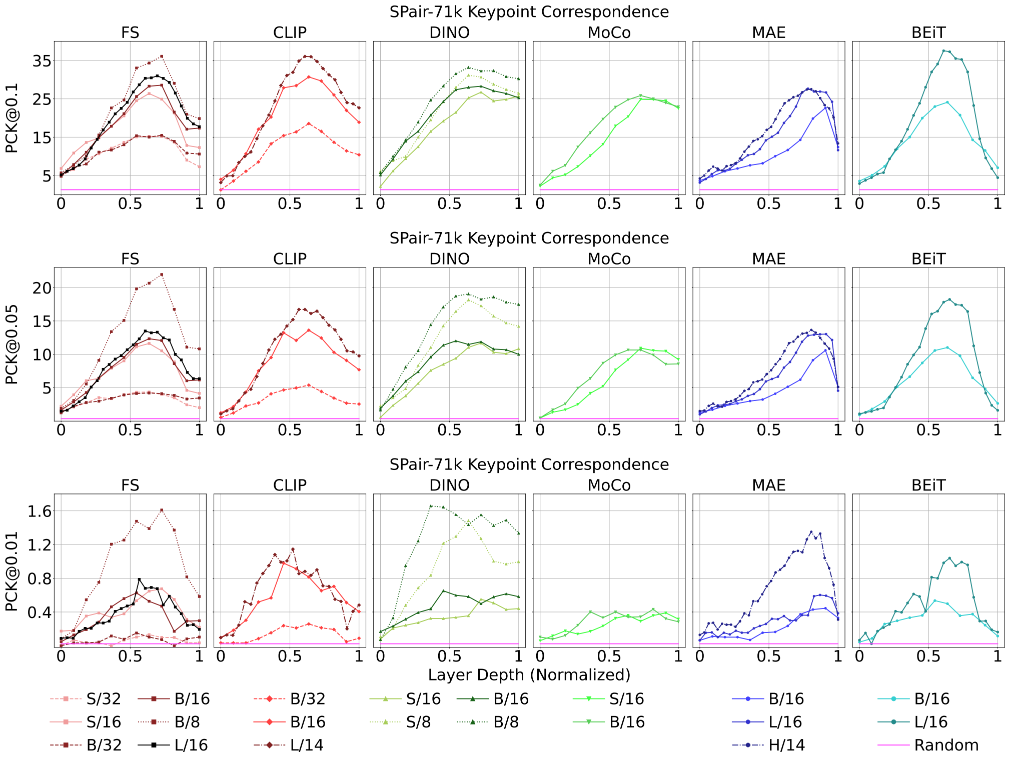

Figure 31. SPair-71k Keypoint Correspondence comparison for all ViTs