Understanding contrastive versus reconstructive self-supervised learning of Vision Transformers

Shashank Shekhar [1] **Florian Bordes** [1, 2, 3] **Pascal Vincent** [1, 2, 3]

Ari Morcos [1]

[1]Meta AI (FAIR)
[2]MILA - Quebec AI Institute
[3]Université de Montréal, DIRO

Abstract

While self-supervised learning on Vision Transformers (ViTs) has led to state-of-the-art results on image classification benchmarks, there has been little research on understanding the differences in representations that arise from different training methods. We address this by utilizing Centred Kernel Alignment for comparing neural representations learned by contrastive learning and reconstructive learning, two leading paradigms for self-supervised learning. We find that the representations learned by reconstructive learning are significantly dissimilar from representations learned by contrastive learning. We analyze these differences, and find that they start to arise early in the network depth and are driven mostly by the attention and normalization layers in a transformer block. We also find that these representational differences translate to class predictions and linear separability of classes in the pre-trained models. Finally, we analyze how fine-tuning affects these representational differences, and discover that a fine-tuned reconstructive model becomes more similar to a pre-trained contrastive model.

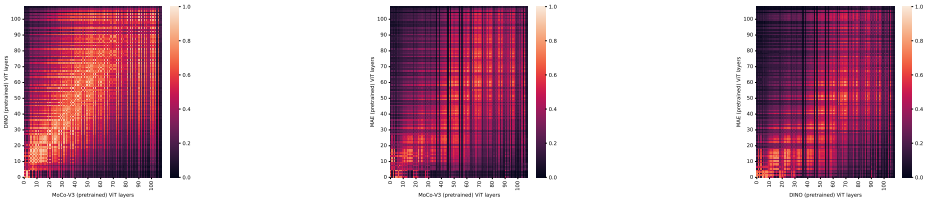
1 Introduction

Self-supervised learning (SSL) has emerged as the state-of-the-art learning paradigm for learning visual representations for tasks such as image classification, object detection, etc. SSL does not require data labels, which gives it an advantage over supervised learning when it comes to learning representations from large scale data. Among visual SSL, two broad categories of training have emerged in contrastive learning [1, 2, 3] and reconstructive learning [4].

While both contrastive and reconstructive learning have demonstrated strong IID classification results, there are several open questions in terms of *how* each training learns to solve these visual tasks. Are the pre-trained representations learned by contrastive and reconstructive methods similar? How does supervised fine-tuning affect theirs representations? Are the similarities and differences in the representations learned by these methods affected by depth and layer-types? Understanding the answers to these questions is important to address several theoretical and practical questions about visual SSL; like why frozen contrastive representations perform better for transfer learning with a linear probe classifier, while reconstructive learning representations transfer better when the ViT is fine-tuned end-to-end [4].

In this paper, we study these questions by comparing the representations of a standard ViT-Base model [5] trained with 16x16 image patches (ViT-B/16) on the ImageNet [6] dataset across popular contrastive (MoCo-V3 [2] and DINO [3]) and reconstructive (MAE [4]) methods using Centred

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) MoCo-V3 and DINO (b) MoCo-V3 and MAE (c) DINO and MAE

Figure 1: CKA similarity between pairs of transformer layers in ViT-B/16 trained with contrastive and reconstructive learning. Similarity between contrastive models is higher and shows strong correspondence in early and intermediate layers. Similarity between reconstructive and contrastive models is lower, and shows correspondence in blocks of layers (Section 2.1)

Kernel Alignment (CKA) [7] – a vector similarity index that has been demonstrated its utility for comparing neural network representations [8, 9, 10]. Further details on the background and our experimental setup are provided in Appendices A and B. Our findings are as follows:

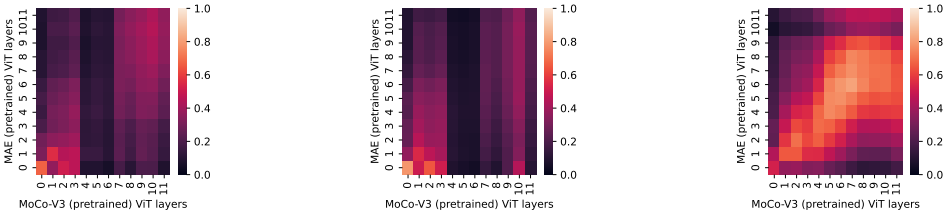
- Representations between contrastive methods are highly similar under CKA, except in the final few layers. On the other hand, differences start to arise early in the architecture across contrastive and reconstructive representations and the representations are markedly different.
- Differences in representations across methods are not uniformly distributed across layer types and are driven primarily by attention and normalization layers.
- Dissimilar representations across contrastive and reconstructive methods translated to the class separability of intermediate as well as final representations. Similar representations across contrastive methods leads to similar class predictions (both correct and incorrect).
- Upon fine-tuning, we find that the representations learned by reconstructive methods move towards the pre-trained representations learned by contrastive learning in embedding space.

2 Results

2.1 How does representational structure of MoCo-V3, DINO, and MAE compare?

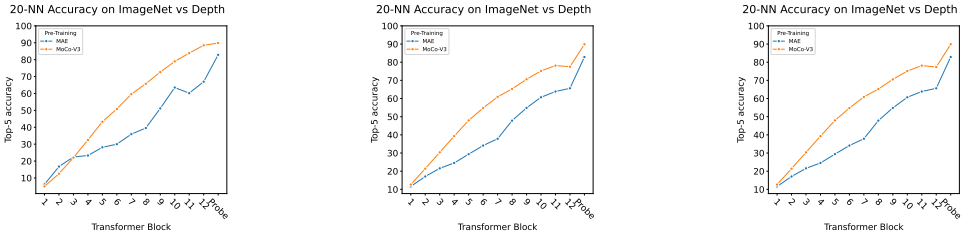
We begin by performing pairwise comparisons of the representational structures of MoCo-V3, DINO, and MAE. In Fig 1, we plot the CKA matrices of the transformer block layers across these pairs.

We observe that two contrastive learning procedures (MoCo-V3 and DINO) have very similar representations (Fig 1a). We also observe that the early and intermediate layers show a strong correspondence in their representations across the two contrastive methods, and these layers are quite dissimilar from the later layers (final three transformer blocks). The later layers are comparatively less similar across both contrastive methods, which could be explained by each contrastive method learning different view invariances based on its augmentations [10]. In comparison, the reconstructive learning method (MAE) has representations that are very dissimilar to both the contrastive methods (Fig 1b and 1c). The representations between corresponding layers of a reconstructive and contrastive model are much less similar. We also observe emergence of block-wise correspondences in layer similarities: the first quarter of MAE layers are similar to the first half of layers in MoCo-V3 and DINO, while the last three-quarters are similar to the last half. This implies differences in how spatial information is aggregated and localized across contrastive and reconstructive learning [8].



(a) MHSA-QKV (b) LayerNorm-1 (c) MLP-FC-1

Figure 2: Layer-wise CKA simlarity between MoCo-V3 and MAE. Within a transformer block, attention and normalization layers are much more dissimilar than fully-connected layers



(a) CLS (b) GAP (c) GAP w/o CLS

Figure 3: 20-Nearest Neighbour classification accuracy across the ViT-B/16 transformer blocks and linear probe in MoCo-V3 and MAE. The class separability diverges after a quarter of ViT-B/16 layers, and the last layers of MoCo-V3 already contain significant amount of class information (3a).

2.2 Which layers drive these differences in representations?

Next, we focus on the differences in the representations of MoCo-V3 and MAE across various types of layers in a transformer block. In Figure 2, we plot the CKA similarity across a subset of three layers from each block: the multi-head self-attention layer (MHSA-QKV), the layer normalization before the attention layer (LayerNorm-1), and first linear layer after the residual connection (MLP-FC-1).

We observe that CKA similarity between attention and normalization layers across MAE and MoCo-V3 are much lower than fully connected layers. In fact, the first and intermediate attention layers of MoCo-V3 are entirely dissimilar from any of the attention layers in MAE. This could imply that the attention layers are learning different order of features in the intermediate layers of the ViT, or one learning more texture/shape features than the other [11].

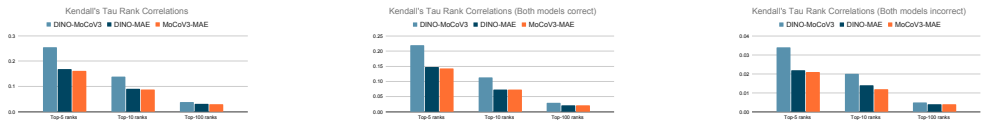
2.3 How do differences in representations lead to differences in class separability?

Next, we look at how the class separability of the ViT-B/16 model evolves in the intermediate representations across network depth based on the pre-training method. We calculate the 20 nearest-neighbour classification accuracy ([3, 1] after each transformer block (12 in total in ViT-B/16) and a linear probe trained on top of the pre-trained representation. Following [8], three different representations are used: the CLS token features, Global Average Pooled (GAP) features from all tokens, as well as Global Average Pooled features from all tokens except the CLS token (GAP w/o CLS). We plot the classification accuracy results in Fig 2.3.

From the CLS token feature classification in Fig 3a two important observations arise: the class separability of the MAE model starts to diverge from the MoCo-V3 model after the third transformer block, corresponding to the first quarter of layers that were observed to be similar to the first half of the MoCo-V3 model in Section 2.1. Secondly, the MoCo-V3 model already contains signficant amount of class information in the last transformer layers, as the gap between the probe and final ViT block accuracy is quite small. For the global features, there is less class information in the last layers of MoCo-V3, but the class separability gap with MAE persists throughout the ViT-B/16 depth.

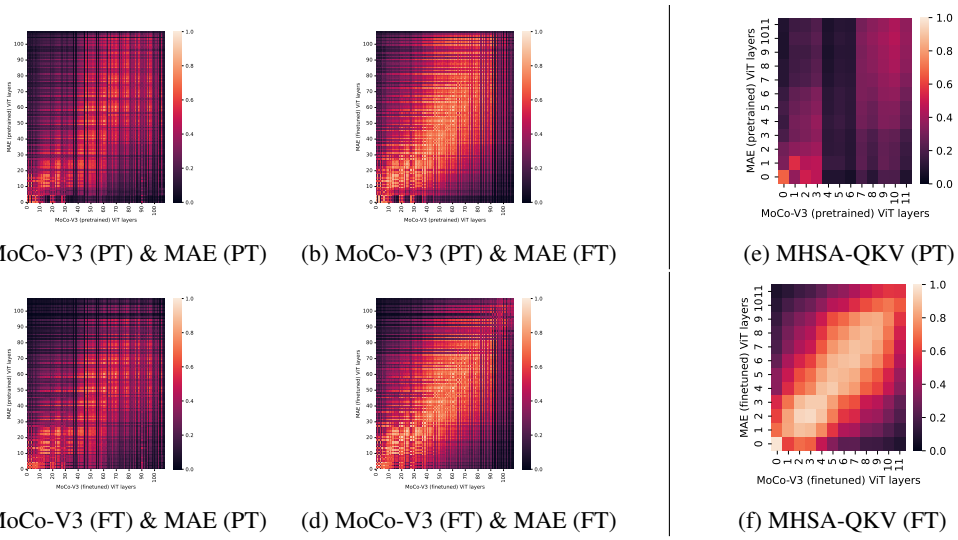
2.4 Do the differences in representations and class separability translate to class predictions?

We also consider whether the representational similarity in contrastive models and the differences in class separability across contrastive and reconstructive models translates to the predictions made by these models. In order to evaluate this, we consider the Kendall's Tau rank correlation coefficient of the top-5, top-10, and top-100 class predictions averaged across the ImageNet validation set from MoCo-V3, DINO, and MAE in Fig 4. We observe that the ranking predictions generated by MoCo-V3 and DINO are consistently more correlated across all predictions, as well as both correct and incorrect



(a) All Predictions (b) Correct predictions (c) Incorrect predictions

Figure 4: Kendall's Tau rank correlation of linear probe ranks (Top-5/10/100 ranks averaged across ImageNet test set). Contrastive models generate more similar rankings across all predictions (4a), correct predictions (4b), and incorrect predictions (4c).



(a) MoCo-V3 (PT) & MAE (PT) (b) MoCo-V3 (PT) & MAE (FT) (e) MHSA-QKV (PT)

(c) MoCo-V3 (FT) & MAE (PT) (d) MoCo-V3 (FT) & MAE (FT) (f) MHSA-QKV (FT)

Figure 5: CKA similarity between MoCo-V3 and MAE before (PT) and after fine-tuning (FT) (5a, 5b, 5c, 5d). An MAE (FT) ViT-B/16 becomes very similar to a MoCo-V3 (PT), (5c), and the similarity persists with the MoCo-V3 (FT) ViT-B/16 (5d). Attention layers (5e, 5f) show strong linear correspondence as well as block correspondence in CKA similarity after fine-tuning.

predictions. We also calculate the F-1 score of top-1 predictions for DINO and MoCo-V3 (0.9341) and confirm that it is higher than the F-1 score for MAE and MoCo-V3 (0.88). Our results verify that contrastive models make similar class predictions which are also right and wrong in similar ways.

2.5 What happens to self-supervised ViT representations post fine-tuning?

Lastly, we consider what happens to the layer-wise CKA similarity of MoCo-V3 and MAE models after fine-tuning in Fig 2.5. We find that the layers of a ViT pre-trained with MAE and fine-tuned end-to-end are highly similar to that of a pre-trained MoCo-V3 ViT (Fig 5b), implying that instance discriminative contrastive pre-training learns very similar representations to class discriminative fine-tuning. This correspondence remains after fine-tuning MoCo-V3 except in later layers (Fig 5c).

We repeat the experiment from Section 2.2 to analyze which layers drive this increase in similarity after fine-tuning. We find that the layers which were initially most dissimilar after SSL (multi-head self-attention and layer normalization) become the most similar after fine-tuning (Fig 2.5, Fig 6 in Appendix C), while the most similar pre-trained layers (fully-connected) become more similar only in the initial and intermediate layers, but become more dissimilar in the later layers of the ViT (Fig 6 in Appendix C). We conclude that during fine-tuning, the way spatial features are attended to in ViTs pre-trained with reconstructive learning changes significantly to align with how ViTs pre-trained with contrastive learning attends to spatial features; which is largely consistent before and after fine-tuning.

3 Conclusions and Future Work

We compared ViT representations across contrastive (MoCo-V3, DINO) and reconstructive (MAE) SSL methods and demonstrated that DINO and MoCo-V3 representations are similar except in the later layers and show a strong linear correspondence in layer similarity, while MAE and MoCo-V3 are more dissimilar and show a block correspondence. MoCo-V3 and DINO also make similar ImageNet class predictions, both correct and incorrect. We found that the representational differences between MoCo-V3 and MAE are driven by the attention and normalization layers, and lead to different class separability across network depth. Finally, we found that a fine-tuned MAE model becomes similar to a pretrained MoCo-V3 model, driven by increased similarity in attention layers.

Our findings lead to several important questions, especially around the mechanistic explanations of how the representations change after fine-tuning. How does finetuning a masked model lead to better scaling in performance versus finetuning a contrastive model? Is there a way to combine both learning mechanisms to learn representations that are linearly separable and also amenable to fine-tuning? What happens in the training dynamics of these models during fine-tuning which causes them to become more similar? We leave these questions for future work.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[2] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[8] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

[9] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020.

[10] Tom George Grigg, Dan Busbridge, Jason Ramapuram, and Russ Webb. Do self-supervised and supervised methods learn similar visual representations? *arXiv preprint arXiv:2110.00528*, 2021.

[11] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022.

[14] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[15] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

[16] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

[17] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[19] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[20] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Standard ImageNet train and val splits are used for training and validation. Training hyper-parameters are given in B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] MoCo-V3 and MAE pre-trained models were obtained from GitHub under Attribution-NonCommercial 4.0 International license. DINO pre-trained models were obtained from GitHub under Apache-2.0 license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix: Background

Vision Transformers Transformers [12] were introduced as sequence-to-sequence models for natural language translation. Vision Transformers (ViTs) [5] adapted them for vision by tokenizing image patches as transformer inputs, and adding an extra `CLS` token to represent object class, which

is later used to train an image classifier. Besides image classification, ViTs have demonstrated state-of-the-art empirical results across a variety of visual tasks such as object detection [13], semantic segmentation [14], and strong results across a variety of other visual tasks [15]. Recent works [8, 16] have tried to understand how ViTs work and how they differ from convolutional neural networks, which used to be standard architecture for modelling visual tasks.

Self-Supervised Learning Of Visual Representations SSL from visual data learns by exploiting known invariances present in images. In contrastive learning [1, 2, 17, 3], this is achieved by learning view invariance of views from the same image. For example, MoCo-V3 [17] takes two crops (views) of images, and encodes each crop through two parallel encoders, f_q and f_k to generate embeddings q and k. A contrastive loss function InfoNCE loss [18] is minimized:

$$\mathcal{L}_q = -\log \frac{\exp\left(q \cdot k^+/\tau\right)}{\exp\left(q \cdot k^+/\tau\right) + \sum_{k^-} \exp\left(q \cdot k^-/\tau\right)} \tag{1}$$

where k^+ represents the embedding from a different crop of the same image as q, while $\{k^-\}$ is the set of embeddings from other images.

Reconstructive learning exploits the strong spatial correlation present locally in natural images to reconstruct the image in the input space from representations learned only on occluded views of the input. This approach towards SSL with ViTs involves masking/corruption of input tokens, learning a visual representation from this noisy input, and then predicting the masked values [19, 5, 4]. Among such approaches, Masked Autoencoder (MAE) [4], learns an SSL representation by passing the highly (70+ %) masked input through a ViT, and then reconstruct the input by predicting the pixel values of each masked patch. The Mean Squared Error between the reconstructed and original image serves as the loss function for MAE.

Representation comparison across neural networks and layers Centred Kernel Alignment (CKA) [7] has been demonstrated [7, 8, 9] to be a useful metric for comparison of neural network representations across layer dimensionality, model initialization, and neural architectures. The CKA value between two p_1 and p_2 dimensional representational matrices of m examples $\mathbf{X} \in \mathbb{R}^{m \times p_1}$ and $\mathbf{Y} \in \mathbb{R}^{m \times p_2}$ is the normalized Hilbert-Smith Independence Criteria [20] of the Gram similarity matrices $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ and $\mathbf{L} = \mathbf{Y}\mathbf{Y}^T$ given as:

$$CKA(\mathbf{K}, \mathbf{L}) = \frac{\mathrm{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\mathrm{HSIC}(\mathbf{K}, \mathbf{K})\ \mathrm{HSIC}(\mathbf{L}, \mathbf{L})}} \tag{2}$$

We adapt the formalization from [9] which approximates the linear CKA metric by averaging over k minibatches to obtain the minibatch CKA metric. Minibatch CKA over two sets of activation matrices $\mathbf{X}_i \in \mathbb{R}^{n \times p_1}$ and $\mathbf{Y}_i \in \mathbb{R}^{n \times p_2}$ of the i^{th} minibatch of n examples is given as:

$$\mathrm{CKA}_{minibatch} = \frac{\frac{1}{k}\sum_{i=1}^{k} \mathrm{HSIC}_1\left(\mathbf{X}_i\mathbf{X}_i^\top, \mathbf{Y}_i\mathbf{Y}_i^\top\right)}{\sqrt{\frac{1}{k}\sum_{i=1}^{k} \mathrm{HSIC}_1\left(\mathbf{X}_i\mathbf{X}_i^\top, \mathbf{X}_i\mathbf{X}_i^\top\right)}\sqrt{\frac{1}{k}\sum_{i=1}^{k} \mathrm{HSIC}_1\left(\mathbf{Y}_i\mathbf{Y}_i^\top, \mathbf{Y}_i\mathbf{Y}_i^\top\right)}} \tag{3}$$

where HSIC_1 is an unbiased estimator of the Hilbert-Smith Independence Criteria such that the CKA value is independent of batch size. The HSIC_1 between two similarity matrices $\mathbf{K}$ and $\mathbf{L}$ ($\tilde{\mathbf{K}}$ and $\tilde{\mathbf{L}}$ are obtained by setting the respective diagonal entries to zeros) is given as:

$$\mathrm{HSIC}_1(\mathbf{K}, \mathbf{L}) = \frac{1}{n(n-3)}\left(tr(\tilde{\mathbf{K}}\tilde{\mathbf{L}}) + \frac{\mathbf{1}^\top\tilde{\mathbf{K}}\mathbf{1}\mathbf{1}^\top\tilde{\mathbf{L}}\mathbf{1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1}^\top\tilde{\mathbf{K}}\tilde{\mathbf{L}}\mathbf{1}\right) \tag{4}$$

B Appendix: Experimental Setup Details

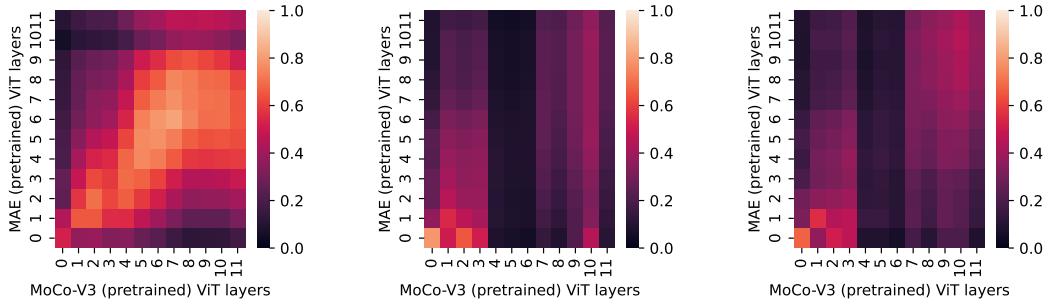
Pre-training We use a ViT-B/16 [5] architecture as the ViT backbone in all our comparisons. It takes as input 16×16 image patches as input tokens, and is made up of 12 transformer blocks. Each

transformer block consists 12-head self-attention layers and a 768-dimensional token embedding. Our ViT models were pre-trained on ImageNet using the procedures outlined in [17, 3, 4] respectively, and we utilize the pre-trained weights publicly released by the authors.

Fine-Tuning All models are fine-tuned under the same hyper-parameter settings. We train on 224×224 ImageNet images using a batch size of 64, and train for 150 epochs on a total of 16 GPUs across 2 nodes. Standard supervised ImageNet data augmentation is used. An Adam optimizer is used with a learning rate of 5e-4, momentum of 0.9, and weight decay of 0.05. A cosine learning rate scheduler is used with a minimum learning rate of 1e-5. The first five epochs are used for learning rate warm-up with a learning rate of 1e-6.

Minibatch Centred Kernel Alignment For our mini-batch CKA computations, we use a batch size of 32 and sample a total of 1024 examples without replacement for computing the representations. Like [8], we compared our mini-batch CKA values across a large range of mini-batch sizes (2^5 to 2^{10}) as well as a large range of examples (10^3 to 10^6) and found no noticeable differences.

C Appendix: Layer-Wise impact on Representational Differences Before and After Fine-Tuning



(a) MLP-FC-1 (Pre-trained) (b) LayerNorm-1 (Pre-trained) (c) Attention-QKV (Pre-trained)

(d) MLP-FC-1 (Fine-tuned) (e) LayerNorm-1 (Fine-tuned) (f) Attention-QKV (Fine-tuned)

Figure 6: CKA similarity between MoCo-V3 and MAE before and after fine-tuning by layer type.

In addition to Fig 2.5 we include additional comparisons of layer-wise CKA similarity between MoCo-V3 and MAE layers before and after fine-tuning in 6. We can observed that the similarity between the fully-connected layers (MLP-FC1) increases for the initial and intermediate ViT layers but decreases for the later layers. However, the similarity between multi-head self-attention layers (MHSA-QKV) and layer normalization layers after attention (LayerNorm) of both models increases remarkably post fine-tuning. There is also a strong linear correspondence (layers at similar depth learn similar features) as well as strong block correspondence (groups of layers learn similar features) in the initial and intermediate MHSA-QKV and LayerNorm layers after fine-tuning.