OBJECTIVES MATTER: UNDERSTANDING THE IMPACT OF SELF-SUPERVISED OBJECTIVES ON VISION TRANSFORMER REPRESENTATIONS

Shashank Shekhar[1*] **Florian Bordes**[1,2,3] **Pascal Vincent**[1,2,3] **Ari S. Morcos**[1]

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

ABSTRACT

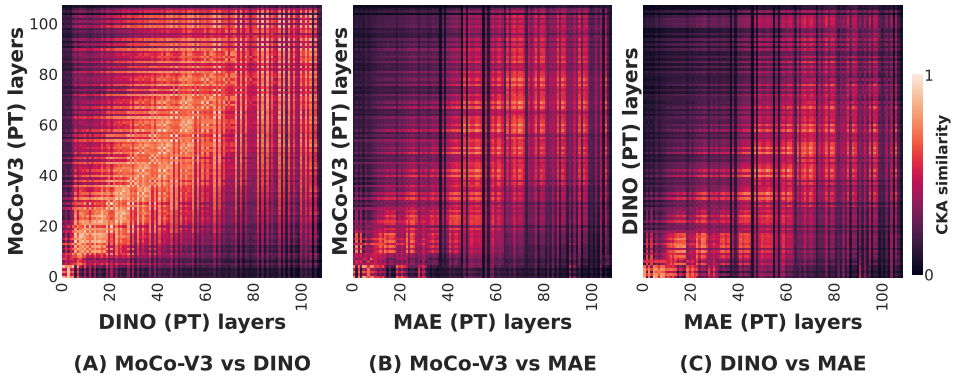
Joint-embedding based learning (e.g., SimCLR, MoCo, DINO) and reconstruction-based learning (e.g., BEiT, SimMIM, MAE) are the two leading paradigms for self-supervised learning of vision transformers, but they differ substantially in their transfer performance. Here, we aim to explain these differences by analyzing the impact of these objectives on the structure and transferability of the learned representations. Our analysis reveals that reconstruction-based learning features are significantly dissimilar to joint-embedding based learning features and that models trained with similar objectives learn similar features even across architectures. These differences arise early in the network and are primarily driven by attention and normalization layers. We find that joint-embedding features yield better linear probe transfer for classification because the different objectives drive different distributions of information and invariances in the learned representation. These differences explain opposite trends in transfer performance for downstream tasks that require spatial specificity in features. Finally, we address how fine-tuning changes reconstructive representations to enable better transfer, showing that fine-tuning re-organizes the information to be more similar to pre-trained joint embedding models.

1 INTRODUCTION

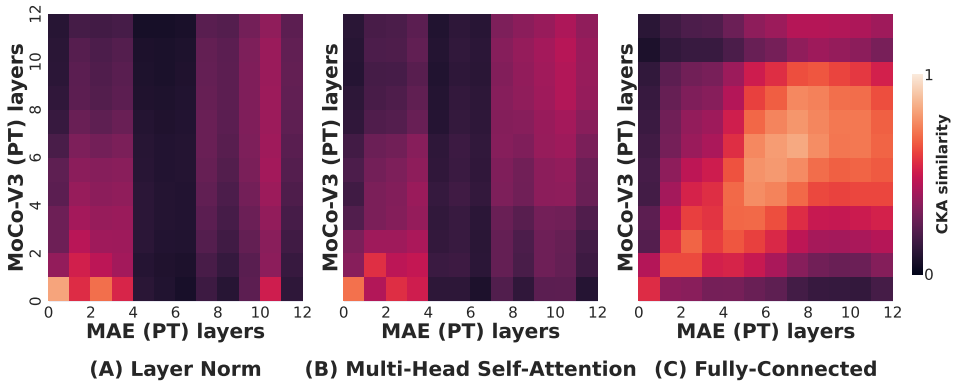
Among Self-Supervised Learning (SSL) methods for learning Vision Transformer (ViT) representations, two broad categories have emerged: joint embedding based learning (Chen et al., 2020b; Caron et al., 2021) and reconstruction-based learning (Zhou et al., 2021; He et al., 2022) (*referred to as* **JE** *and* **REC** *respectively hereafter*). JE training maximizes view invariance between handcrafted augmentations of the same image via a joint-embedding (Siamese) Chen & He (2020). In contrast, REC objectives train models to reconstruct images in pixel space from a masked input. JE learning demonstrates stronger linear probe transfer than REC but requires more inductive biases. While some methods (El-Nouby et al., 2021; Assran et al., 2022b) have tried to combine these objectives, the reason why such difference arise across methods remains unclear. We seek to better understand the differences in representations learned across SSL ViT methods in order to diagnose what information is learned and discarded during SSL pre-training. We approach differences between SSL methods from the perspectives of representational (dis)similarity, accessibility of information for transfer, as well as changes that arise during fine-tuning, leading to the following contributions:

- JE representations are more similar to each other than REC representations and vice versa (even across architectures). These differences arise early in the network, and are concentrated in the Layer Norm and Self-Attention Layers (Sec 2.1).
- JE models contain more linearly decodable representations because all relevant class discriminative information is available in final pre-projector layer `CLS` token. In contrast, REC models lack key invariances and distribute class discriminative information across layers, leading to poor downstream transfer w/o fine-tuning (Sec 2.2).

*Work done during an AI Residency at FAIR. Corresponding author: sshkhr@meta.com



(a) JE models show high similarity, and are less similar to REC models.



(b) Attention, Norm layers = more dissimilar (than FC layers.)

Figure 1: CKA similarity between pre-trained ViTs (Fig 1a), broken down by layer-types (Fig 1b)

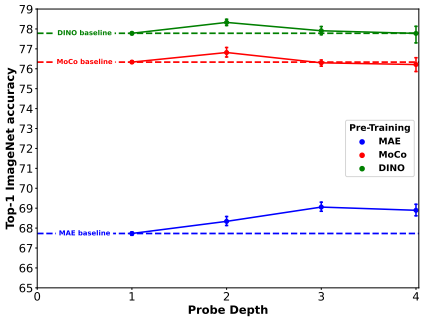
- Training probes on multiple layers from REC models improves transfer. The downstream task also plays a role in transfer from frozen pre-trained representation, as we discover that reconstructive features transfer better to tasks requiring spatial specificity (Sec 2.2).
- Fine-tuning REC models makes them similar to JE models by re-organizing class information into the final layer. During fine-tuning, REC models take a more efficient path through parameter space than JE models (Sec 2.3).

2 Experiments and Results

2.1 How does representational structure of ViTs trained with different SSL objectives compare?

Representation Similarity Analyses as a Lens for Model Understanding To analyze why reconstructive models transfer differently than JE models without any fine-tuning, we perform pairwise comparisons of the representational structures of MoCo-V3, DINO, and MAE using CKA (Fig. 1a). The two JE learning procedures (MoCo-V3 and DINO) have very similar representations (Fig. 1aA), especially in the early and intermediate layers. In comparison, the REC learning method (MAE) has representations that are very dissimilar to both JE methods (Fig. 1aB,C).

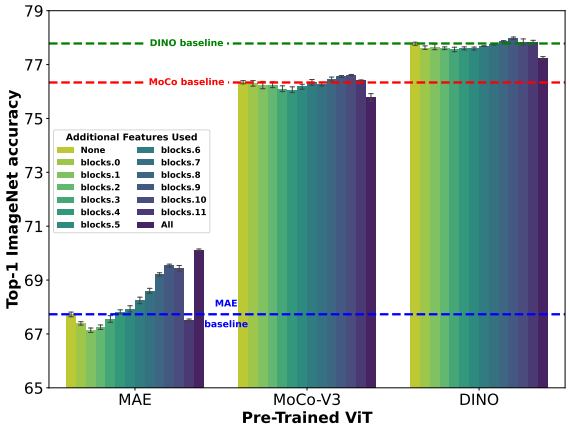
Which layers drive differences in representations? We aim to understand whether the differences are higher in attention layers which encode global shape features, or in MLP layers which encode local texture features (Naseer et al., 2021; Anonymous, 2023). In Fig. 1b, we plot the CKA similarity across a subset of each ViT block: the layer normalization before the attention layer (Layer Norm), the multi-head self-attention layer (Multi-Head Self-Attention), and first linear layer after the residual connection (Fully-Connected). We observe that CKA similarity between attention and normalization layers across MAE and MoCo-V3 are much lower than fully connected layers.



(a) Supervised transfer with non-linear probes.

	Final Layer	Final + (Best) Intermediate	All layers
DINO	77.78 ± 0.07%	77.97 ± 0.05% (+0.19%)	77.24 ± 0.07% (-0.54%)
MoCo-V3	76.33 ± 0.07%	76.61 ± 0.02% (+0.32%)	75.77 ± 0.18% (-0.56%)
MAE	67.73 ± 0.10%	69.53 ± 0.07% (+1.8%)	70.12 ± 0.04% (+2.4%)

(b) Linear Probing w/ feats. at different depths.



(c) Using additional intermediate layer CLS token features for linear probe transfer.

Figure 2: Evaluating distribution of information across pre-trained ViTs by probing ImageNet top-1 accuracy (mean ± std for best 5 runs)

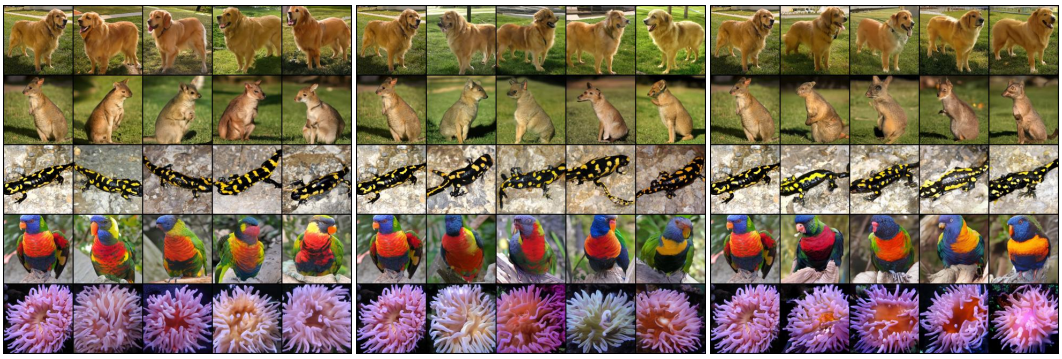
Does objective or architecture drive representational structure? We compare the CKA values between ViT-CNN model pairs learned with the same objective against ViT-ViT model pairs learned with different objectives[1] Fig. 11 plots the CKA similarity between pairs of models as a function of the distance between two layers in each model pair. In Fig. 11a, we show that CKA similarity for two JE models trained on different architectures is consistently higher than for a JE and a REC ViT. In Fig. 11b, when the layer depths are similar, the inter-layer CKA for REC CNN-ViT models is of similar order of magnitude as the CKA for a JE and a REC ViT. As the distances between the layers being compared increases, the CKA across REC ViT-CNN pair stays high while the CKA across pre-training objectives in a ViT-ViT pair falls off. Hence, we conclude that the SSL objective governs representational similarity more than architecture choice for both REC and JE learning.

How do representational differences manifest when utilizing self-supervised features for class predictions? We consider how class discriminative information diverges between layers. In order to do so, we calculate the 20 nearest-neighbour classification accuracy after each transformer block (12 in total in ViT-B) as well as for a linear probe trained on top of the SSL representation. Following (Raghu et al., 2021), two different representations are used: the CLS token features, as well as Global Average Pooled features from all tokens except the CLS token (GAP w/o CLS), in order to ensure we utilize class information present in the CLS token as well as outside the CLS token. We plot the classification accuracy in Fig. 12. The predictions are highly consistent across JE models, demonstrating that similar pre-training objectives lead to features that represent different object classes similarly leading to class predictions which are also right and wrong in similar ways. (See Appendix F for details.)

2.2 Does the SSL objective impact information distribution in representations?

Is class discriminative information in pre-trained MAEs non-linearly decodable? While previous work on JE and REC learning has consistently demonstrated that the final CLS token in the former contains features which can more easily be decoded by a linear probe, there has been little exploration into utilizing non-linear probes for transfer. In Fig. 2a, we show the results on Top-1 ImageNet accuracy when using non-linear probes of increasing depth on the final CLS token features of an SSL pre-trained ViT. Both JE models do not see any improvement in performance from using a deeper probe. However, an MAE pre-trained ViT reports a performance improvement of 1.03% on top-1 accuracy when using a three layer non-linear probe. While this improvement is not enough to match the performance of JE final layer features, it demonstrates that there is additional class-specific information available in final CLS token features, but it is inaccessible with a linear probe.

[1] See Appendix B for details of CNN models used.



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

Figure 3: RCDM Bordes et al. (2022) samples generated by a diffusion model conditioned on pre-trained ViT features shows differences in invariances learned

What features invariances have been learned during self-supervised learning in the final layer of a ViT? Since the information present in the final `CLS` token is more suited for classification in JE models than REC models, it becomes important then to characterize it. To do so, we used RCDM (Bordes et al., 2022), a conditional diffusion model that uses pre-trained SSL representations as conditioning. For training, we used the face-blurred version of ImageNet (Yang et al., 2022). As RCDM is a stochastic generative model, the information that varies across samples (because of the noise) is not contained in the representation while the information that remain constant across many samples is contained in the representation. We visualize the samples for both JE and REC models Fig. 3. We find that JE representations yield images with horizontal flip invariance, whereas REC representations do not.

Does REC learning discard class discriminative information in its final layer? In this section, we try to establish whether class discriminative information is discarded across network depth by REC SSL. We find (Fig. 2c, Table 2b) that training a linear probe on both the final and an intermediate layer `CLS` token features leads to an improvement in classification accuracy for REC representations, but provides no marginal utility for JE representations. In Appendix C we consider different transfer tasks like object detection and segmentation, and demonstrate how distribution of information varies for pre-trained objectives.

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

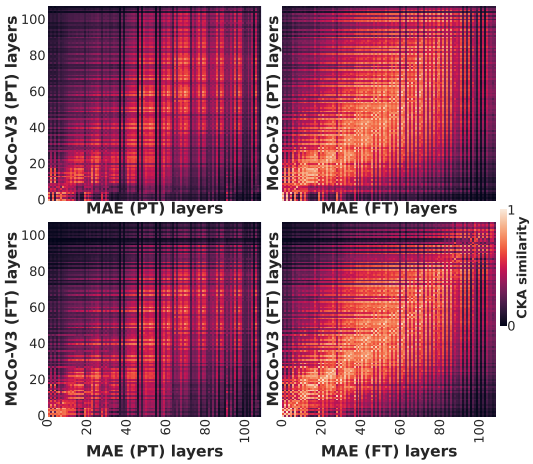


Figure 4: CKA similarity between MoCo-V3 and MAE before (PT) and after fine-tuning (FT).

Thus far, we have focused on pre-trained representations, but how does fine-tuning impact representational structure? Given the importance of fine-tuning to the downstream performance of REC models, this is a critical question.

How does representational similarity change post fine-tuning? How does the layer-wise CKA similarity changes as a result of fine-tuning? We find (Fig. 4) that fine-tuned MAE features are highly similar to that of a pre-trained MoCo-V3, implying that instance discriminative JE pre-training learns very similar representations to class discriminative fine-tuning. This correspondence remains after fine-tuning MoCo-V3 except in later layers (See Fig. 8 for qualitatively similar results with DINO). In addition, 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. 9, Fig. 10 in Appendix D), .

A mechanistic understanding of increased similarity To understand why fine-tuned MAEs can quickly exceed transfer performance of fine-tuned JE models, we look at the the L_2 norm of the total difference between the pre-trained and fine-tuned model parameters, and normalize it by the L_2 norm of the difference between parameters after each epoch. This gives us a measure of the efficiency of the path taken by the ViT model during fine-tuning, a score of 1 implies a perfectly straight path from pre-trained to fine-tuned model versus a score closer to 0 implies a very inefficient path. A visualization of the quantity we measure is shown in Fig. 14a. We find that the relative displacement of the MAE pre-trained model attention layers is the noticeably lower than the MoCo-V3 and DINO models when we observe the fine-tuning dynamics of the attention layers in Fig. 14b.

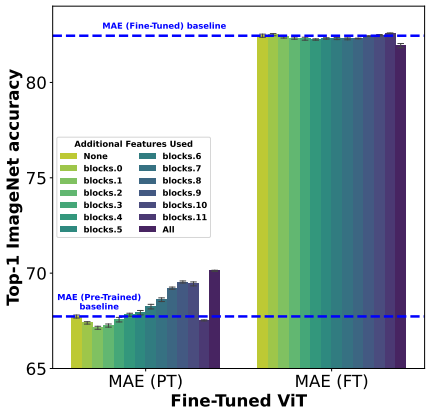


Figure 5: Marginal class information in intermediate feats. for linear probing pre-trained (PT) vs fine-tuned (FT) MAE.

Does representational similarity translate to functional similarity for finetuned MAEs? To answer this, we repeat the experiments from Section 2.2 on a fine-tuned MAE model to visualize how the invariances learned in the final `CLS` token features change. In Fig. 17, we visualize the samples from an RCDM trained on the final `CLS` token features of a fine-tuned MAE model. The samples generated from a particular horizontal orientation of the object do not preserve this information i.e. during fine-tuning the ViT learns to be invariant to horizontal flip. Thus, fine-tuning enables MAE pre-trained ViTs to learn invariances that are similar to pre-trained JE models and informative for classification.

In Fig. 5, we show that the marginal utility of training a linear probe on the intermediate `CLS` token features in a fine-tuned MAE. Unlike a pre-trained MAE (Fig. 2c) where not all class specific information was available in the final layer features, we find that the fine-tuned MAE does not perform any better when a linear probe is additionally trained on its intermediate features. Thus, fine-tuning with supervision leads to a re-organization of information in the ViT layers, and the class discerning information becomes readily available in the final `CLS` features.

3 Discussion

Conclusion We analyzed ViT representations and their transferability when trained via two popular self-supervised approaches: (1) Joint-Embedding (JE) methods (MoCo-V3, DINO), and (2) Reconstruction-Based (REC) methods (MAE). We reveal key differences learned across both representations and how these differences are localized by layer types while being distributed across network depth. We explained why JE models transfer better with a linear probe, as their final layer `CLS` tokens contain all pertinent information for class discriminative learning. We also presente ways to extract the relevant information distributed across REC layers *without fine-tuning*. Finally, we show how fine-tuning modifies REC features to be more linearly decodable by re-organizing class information into the final layer.

Limitations and Future Work Our pre-training dataset, ImageNet is a balanced large-scale dataset, SSL ViT methods have demonstrated poor empirical performance and transfer when trained on imbalanced datasets (Assran et al., 2022a). We also focused on understanding the ViT-Base model representations in this study. Seeing how different SSL pre-training methods scale with model size and dataset size and diversity is an interesting avenue for future research. Another potential future study could look into quantifying the notion behind 'information' available in SSL representations from a mathematical perspective instead of our treatment of representational information as its impact on downstream transfer. It would also be interesting to see how both JE and REC representations transfer to other downstream tasks beyond classification, detection, and segmentation. Lastly, we are very interested in exploring the impact of SSL objectives on multi-modal representation learning methods such as CLIP (Radford et al., 2021) and Omni-MAE (Girdhar et al., 2022).

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl.

Anonymous. What do self-supervised vision transformers learn? In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=azCKuYyS74. under review.

YM Asano, C Rupprecht, and A Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2019.

Mahmoud Assran, Randall Balestriero, Quentin Duval, Florian Bordes, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, and Nicolas Ballas. The hidden uniform cluster prior in self-supervised learning. *arXiv preprint arXiv:2210.07277*, 2022a.

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pp. 456–473, Berlin, Heidelberg, 2022b. Springer-Verlag. ISBN 978-3-031-19820-5. doi: 10.1007/978-3-031-19821-2_26. URL https://doi.org/10.1007/978-3-031-19821-2_26.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=urfWb7VjmL.

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*, pp. 9650–9660, 2021.

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

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2020.

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*, pp. 248–255. Ieee, 2009.

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.

Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021.

Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2Toe: Utilizing intermediate representations for better transfer learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6009–6033. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/evci22a.html.

Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Omnimae: Single model masked pretraining on images and videos. *arXiv preprint arXiv:2206.08356*, 2022.

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.

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.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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*, pp. 9729–9738, 2020.

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*, pp. 16000–16009, 2022.

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

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

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.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

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.

Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=azCKuYyS74.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

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.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9653–9663, 2022.

Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. A study of face obfuscation in imagenet. In *International Conference on Machine Learning (ICML)*, 2022.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction, June 2021. URL `http://arxiv.org/abs/2103.03230`. arXiv:2103.03230 [cs, q-bio].

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2021.

A Related Work

We study the questions raised by us in Section 1 by comparing the representations of a standard ViT-Base model (Dosovitskiy et al., 2020) trained with 16x16 image patches (ViT-B/16) on the ImageNet (Deng et al., 2009) dataset across popular JE (MoCo-V3 He et al. (2020), DINO Caron et al. (2021)) and REC methods (MAE He et al. (2022)). We summarize the most pertinent directions of research related to our study below.

Self-Supervised Learning of Vision Transformers Self-supervised learning (for ViTs) can be broadly categorized into two families of algorithms. First is the JE SSL family (Chen et al., 2020a; Grill et al., 2020; Zbontar et al., 2021; Bardes et al., 2021; Chen & He, 2020) which rely on training criteria that encourage the representations learned from different augmentations of a given image to be close together. Second is REC SSL family which rely on a reconstruction loss in the pixel space that doesn't require handcrafted data augmentations but instead utilizes a decoder to reconstruct from the noisy representation (Zhou et al., 2021; Xie et al., 2022; He et al., 2022). He et al. (2022) showed that a simple masking approach (MAE) tailored for ViTs followed by a pixel-level reconstruction objective outperforms all other methods for fine-tuning and scaling with dataset and ViT size. However, the performance of MAEs with linear probes was much poorer than that of JE models. Research on combining both methods has focused on sample efficient learning and transfer. Assran et al. (2022b) tried to utilize masked image modelling to improve efficiency for JE learning and better few-shot transfer. El-Nouby et al. (2021) combined joint-embedding learning with REC learning across disjoint subsets of patches, as well as utilizing feature space augmentations and contrastive loss to improve training sample efficiency. Park et al. (2023) performed an extensive study on the differences in pre-trained feature diversity, scale of features, and texture versus shape bias across both methods, and showed that a simple linear combination of losses outperforms individual objectives.

Representation Similarity Analyses as a Lens for Model Understanding Representational Similarity metrics provide a method for comparison of neural network representations across layer dimensionality, model initialization, and neural architectures (Raghu et al., 2017; Morcos et al., 2018; Kornblith et al., 2019). Among these, Centered Kernel Alignment (CKA) (Kornblith et al., 2019) between two representation matrices is given as the normalized Hilbert-Smith Independence Criteria (Gretton et al., 2007) of the Gram similarity matrices. We adapt the formalization from (Nguyen et al., 2020) which approximates the linear CKA metric by averaging over k minibatches to obtain the minibatch CKA metric. Raghu et al. (2021) utilized CKA to demonstrate that information is localized and distributed differently across CNNs and ViTs, and that training set size plays an important role in the scale of features learned by supervised ViTs. Grigg et al. (2021) used it to analyze how supervised and SSL representations defer while controlling for model architecture and training datasets. Park et al. (2023) showed low feature diversity in attention heads in pre-trained JE models versus REC models, by showing high CKA values across depth, attention heads, and tokens.

Transfer Learning from SSL representations Different SSL methods can have very different downstream performances. To visualize how invariances differ between SSL and supervised-trained representations, Bordes et al. (2022) trained a Representation Conditioned Diffusion Model (RCDM) to generate images conditioned on a given pretrained representation. While most SSL methods analyze how intermediate probes (Alain & Bengio, 2017) perform for linear transfer, Evci et al. (2022) showed that probes trained on intermediate layers in addition to final layer features improve transfer performance and robustness. In the supervised setting, Neyshabur et al. (2020) showed that the scale of features being transferred during fine-tuning depends on the relation between the pre-training and transfer tasks. Asano et al. (2019) showed that (older) SSL methods for CNNs cannot match supervised performance irrespective of amount of data and augmentation used, while El-Nouby et al. (2021) showed that REC SSL is more robust to type and size of dataset versus JE learning.

B Additional Experimental Details

B.1 Mini-batch CKA details

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 (Gretton et al., 2007) of the Gram similarity matrices $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ and $\mathbf{L} = \mathbf{Y}\mathbf{Y}^T$ given as:

$$\text{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\text{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\text{HSIC}(\mathbf{K}, \mathbf{K})\ \text{HSIC}(\mathbf{L}, \mathbf{L})}} \tag{1}$$

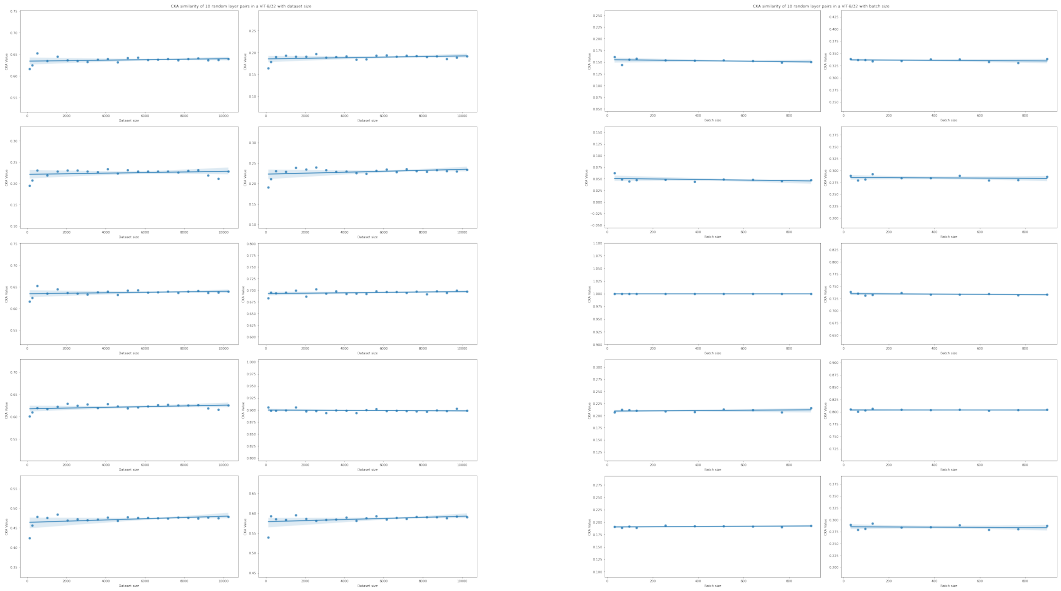
We adapt the formalization from (Nguyen et al., 2020) which approximates the linear CKA metric by averaging over k minibatches to obtain the minibatch CKA metric. Minibatch CKA over two sets of activation matrices $\mathbf{X}_i \in \mathbb{R}^{n \times p_1}$ and $\mathbf{Y}_i \in \mathbb{R}^{n \times p_2}$ of the i^{th} minibatch of n examples is given as:

$$\text{CKA}_{\text{minibatch}} = \frac{\frac{1}{k}\sum_{i=1}^{k} \text{HSIC}_1\left(\mathbf{X}_i\mathbf{X}_i^\top, \mathbf{Y}_i\mathbf{Y}_i^\top\right)}{\sqrt{\frac{1}{k}\sum_{i=1}^{k} \text{HSIC}_1\left(\mathbf{X}_i\mathbf{X}_i^\top, \mathbf{X}_i\mathbf{X}_i^\top\right)}\sqrt{\frac{1}{k}\sum_{i=1}^{k} \text{HSIC}_1\left(\mathbf{Y}_i\mathbf{Y}_i^\top, \mathbf{Y}_i\mathbf{Y}_i^\top\right)}} \tag{2}$$

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:

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

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 Raghu et al. (2021), 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 (Fig. 6).



(a) Minibatch CKA vs samples used

(b) Minibatch CKA vs batch size

Figure 6: Comparisons showing the impact of batch size and number of data samples used to compute minibatch CKA values across a random subset of 10 layers in ViT-B. Minibatch CKA values remain consistent above batch sizes of 2^5 and sample sizes of 1024.

Config	Value
optimizer	AdamW
base learning rate	$1.5\mathrm{e}-4$
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	4096
learning rate schedule	cosine decay
warmup epochs	40
training epochs	800
augmentation	RendomResizedCrop

Table 1: Pre-Training Details for MAE.

Config	Value
optimizer	AdamW
base learning rate	$1.5\mathrm{e}-4$
weight decay	0.1
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	4096
learning rate schedule	cosine decay
warmup epochs	40
training epochs	300
momentum encoder momentum	0.99
momentum rate schedule	cosine
augmentation	RandomResizedCrop ColorJitter RandomGrayscale GaussianBlur Solarize RandomHorizontalFlip

Table 2: Pre-training details for MoCo-V3

B.2 SSL PRE-TRAINING DETAILS

For each of MoCo-V3 Chen et al. (2020b), DINO (Caron et al., 2021), MAE He et al. (2022) we utilize pre-trained models provided by the original authors with the exact pre-training setup as mentioned in the original papers. We summarize these details in Table 1 for REC model MAE, and in Table 2 and 3 for JE models MoCo-V3 and DINO respectively. We pretrain replicate models for all three ourself to verify that our observations also hold on replicates. During this pre-training, Linear *lr scaling* rule is used for large batch training where = *lr = base.lr x batch size / 256*.

Config	Value
optimizer	AdamW
base learning rate	$5\mathrm{e}-4$
weight decay	0.04
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	4096
learning rate schedule	cosine decay
warmup epochs	10
training epochs	300
teacher momentum	0.996
teacher temperature	0.07
teacher temperature warmup epochs	30
augmentation	RandomResizedCrop--Multi (96x96 and 224x244) ColorJitter RandomGrayscale GaussianBlur Solarize RandomHorizontalFlip

Table 3: Pre-training details for DINO

Config	Value
optimizer	LARS
base learning rate	$\{0.1, 1\mathrm{e}-2, 1\mathrm{e}-3\}$
weight decay	$\{0, 5\mathrm{e}-2, 0.1\}$
L-1 regularization α	$\{0, 1\mathrm{e}-\{1, 2, 3, 4\}, 5\mathrm{e}-4\}$
optimizer momentum	0.9
batch size	4096
learning rate schedule	cosine decay
warmup epochs	$\{10, 40\}$
training epochs	$\{100, 200\}$
augmentation	RandomResizedCrop

Table 4: Linear and Non-Linear Probe Transfer details. A hyper-parameter grid search was performed on the cross-product of config values within {}.

B.3 Linear Probe details

Details of our linear probe transfer settings are given in Table 4. Similar to He et al. (2022), we utilize an extra BatchNorm layer without affine transformation before the linear classifier to calibrate feature magnitudes across different layer features for our experiments involving intermediate features. We perform extensive hyper-parameter sweeps by performing a grid search over cross product of values given in Table 4, and report the mean and standard deviation in accuracy for the 5 best performing models in each experiment.

Config	Value
optimizer	AdamW
base learning rate	$1e-3$
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
layer-wise lr decay $[10, 2]$	0.75
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100
augmentation	RandAug $(9, 0.5)$[12]
label smoothing [52]	0.1
mixup [69]	0.8
cutmix [68]	1.0
drop path [30]	0.1

Table 5: Fine-Tuning Transfer details

B.4 Non-Linear Probe details

For our experiments with non-linear probes trained on the last layer `CLS` token features, each non-linear probe layer block is made up of a linear layer (same as linear probe), followed by a BatchNorm layer, followed by a non-linear ReLU activation. The training related hyperparameters remain the same as given in Table 4. Similar to Section B.3, we perform extensive hyper-parameter sweeps, and report the mean and standard deviation in accuracy for the 5 best performing models in each experiment.

B.5 Fine-Tuning details

Details of our fine-tuning transfer settings are given in Table 5.

B.6 Alternate neural architecture details

For our comparisons of representation similarity across types of architectures, we require convolutional models pre-trained with similar objectives as our ViT models. For this purpose, we take a standard ResNet50 (He et al., 2016) pre-trained with DINO and MoCo-V3 objectives as our candidate JE CNN model, and a ConvNextv2-Base (Woo et al., 2023) pre-trained with MAE objective as our candidate REC CNN model. The readers may refer to the original papers for exact model specifications.

B.7 MS COCO Object Detection and Segmentation

We utilize the ViTDet framework introduced by Li et al. (2022), which uses the final `CLS` token features and then uses strided convolutions and deconvolutions to upsample/downsample the single-scale features into a simple hierarchical feature pyramid. The feature pyramid generate uses strides of 4, 8, 16, and 32 - consistent with ResNet based detection/segmentation models.

Once this feature pyramid is built from a ViT backbone, a standard Mask R-CNN (He et al., 2017) is applied on top of the feature pyramid to perform bounding box regression, classification, as well as instance segmentation. In order to evaluate the utility of pre-trained ViT representations for detection and segmentation, we keep the backbone model parameters frozen when we train our Mask R-CNN in Section C.

	Detection			Segmentation		
	AP	AP^{large}	AP^{small}	AP	AP^{large}	AP^{small}
MAE	**30.25**	39.93	18.69	**28.56**	41.73	14.37
MoCo-V3	28.75	40.21	15.49	26.67	41.92	10.89
DINO	32.57	44.14	19.89	30.04	45.77	14.37
DINO (w/o multi-crop)	29.97	40.82	17.55	28.16	42.53	12.99

Table 6: Downstream transfer for object detection and segmentation on MS COCO using a ViTDet based Mask R-CNN with a frozen backbone ViT. **MAE outperforms JE models when transferring from frozen pre-trained features. Due to the scale of its features, it does worse than JE models on larger objects, but performs better on small objects.**

C Segmentation and Detection Results

How do SSL pre-trained ViTs perform on downstream tasks requiring spatial specificity? In Section 2.2, we examined the presence of information in SSL ViT features for linear probe transfer. However, image classification is just one possible downstream task, and does not require a model to preserve exact spatial information about object location. Other downstream transfer tasks, such as object detection and instance segmentation (Lin et al., 2014), require the availability of precise object location in the pre-trained features for transfer. In order to test how SSL objectives influence ViT features for learning location-preserving information, we evaluate the performance of frozen pre-trained ViTs as backbone feature extractors on the MS-COCO detection and segmentation tasks. We utilize the ViTDet framework introduced by Li et al. (2022) to perform these experiments (see Appendix B.7 for details).

Our results are shown in Table 6. Contrary to image classification from frozen representations, we find that `CLS` token features from MAE actually outperform MoCo-V3 features on detection as well as segmentation. While DINO intially outperforms MAE (Table **??**), we hypothesize that it benefits from its unique multi-crop training setup since DINO is specifically trained to be invariant to both local and global scale of objects. In order to verify this, we train a DINO ViT without multi-crops and indeed find that a frozen MAE outperforms a frozen DINO for detection and segmentation.

For the detection and segmentation transfer, we also observe an interesting difference between REC and JE models that pertains to the scale of objects. The frozen REC model performs worse than JE models on localizing larger objects (AP^{large}), but performs better for localizing smaller objects (AP^{small}). Thus, the final `CLS` token features from a MAE are informative for localizing smaller objects, but lack global context to correctly detect larger objects. This observation is consistent with the Park et al. (2023) observations that the receptive field of REC models is more local in the last layer features versus JE models. Our results establish that REC features can be useful out-of-the-box vs JE features when the downstream task requires spatial specificity.

For completeness, we also provide the corresponding detection and segmentation results when the Mask R-CNN with a ViT backbone is fine-tuned end-to-end. With supervised fine-tuning, the discrepancy noted in the detection and segmentation for REC models on larger objects vanishes, since they acquire global context and perform better on AP^{large} versus JE models.

Is there information relevant to detection and segmentation in the intermediate layers of SSL pre-trained ViTs? How is information relative to spatially sensitive transfer tasks distributed across layers in SSL pre-trained models? We repeat our experiment in Table 2b and Fig. 2c, utilizing features from the final and intermediate layer `CLS` tokens, and concatenating them to build the feature pyramid for Mask R-CNN training. However, due to the computational constraints of training Mask R-CNN we limit ourselves to using intermediate features from ViT blocks 9, 10, 11.

We find that both a Mask R-CNN trained on both final and intermediate `CLS` token features outperforms a similar model trained only on the final layer features for both REC models as well as JE

	Detection			Segmentation		
	AP	AP^{large}	AP^{small}	AP	AP^{large}	AP^{small}
MAE	51.57	66.36	35.27	45.84	63.84	27.27
MoCo-V3	48.81	64.83	32.79	43.18	62.86	23.78
DINO	47.73	62.49	32.17	30.04	60.03	24.28

Table 7: Downstream transfer for object detection and segmentation on MS COCO using a ViTDet based Mask R-CNN with a backbone ViT and end-to-end fine-tuning.**With fine-tuning, MAE outperforms both MoCo-V3 and DINO on both large and small objects, implying that REC models learn global scale features with fine-tuning.**

models. While intermediate features offered no marginal utility in linear probe classification for JE models, they contain information relevant to detection/segmentation which is not present in the final `CLS` token. While the JE objective lends itself better to classification by concentrating useful information in its last pre-projector layer, it loses some relevant spatial information.

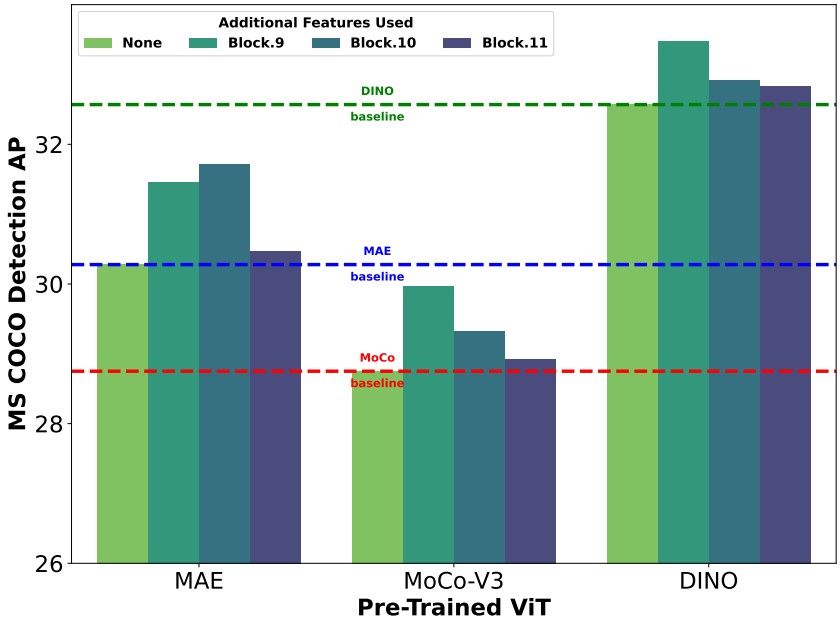
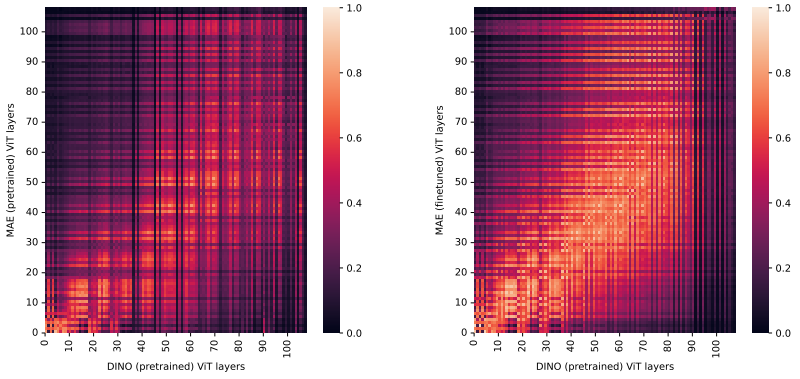


Figure 7: Object detection results on MS-COCO using frozen pre-trained ViT features. **ViTs trained with both kinds of SSL objectives show improved performance when additional intermediate features are used for detection, unlike for classification.**

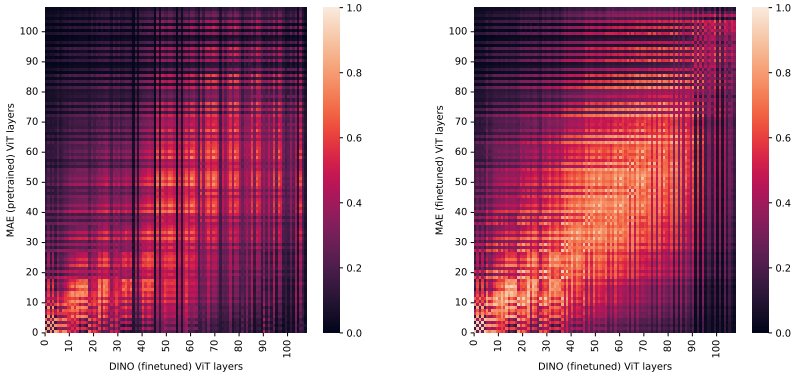
D ADDITIONAL CKA PLOTS

D.1 CKA BETWEEN PRE-TRAINED AND FINE-TUNED MAE AND DINO

For completeness, we provide the plots comparing CKA between DINO and MAE both before and after fine-tuning, analogous to Fig. 4 in the main text with MoCo-V3.



(a) DINO (PT) & MAE (PT) (b) DINO (PT) & MAE (FT)



(c) DINO (FT) & MAE (PT) (d) DINO (FT) & MAE (FT)

Figure 8: CKA similarity between DINO and MAE before (PT) and after fine-tuning (FT). Similar to the MoCo-V3 comparisons 4, an MAE (FT) ViT-B becomes very similar to a DINO (PT), (8c), and the similarity persists with the DINO (FT) ViT-B/16 (8d).

D.2 CKA BETWEEN PRE-TRAINED AND FINE-TUNED JE AND REC MODELS BY LAYER TYPE

In addition to Fig. 4 we include additional comparisons of layer-wise CKA similarity between MoCo-V3 and MAE layers before and after fine-tuning in 10. 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.

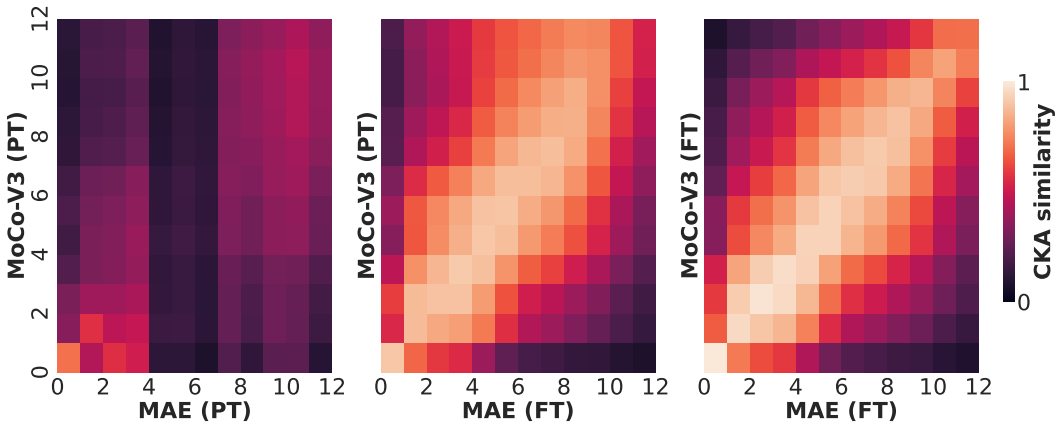
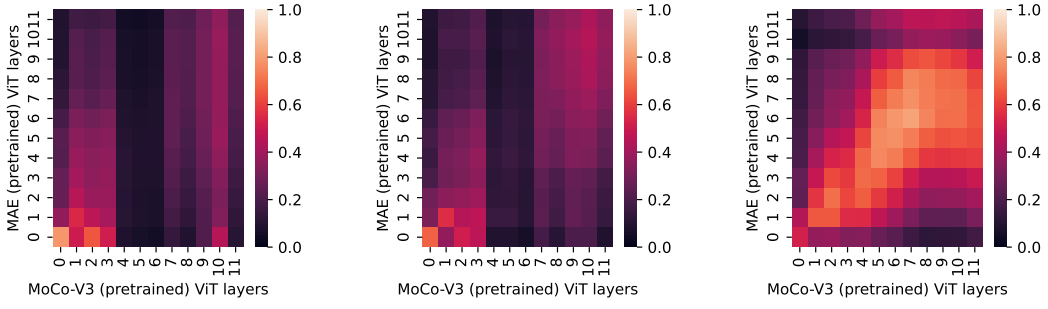
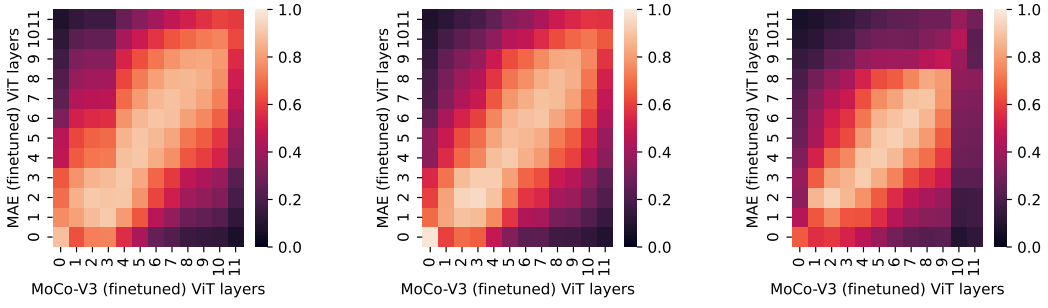


Figure 9: **Attention layers show strong linear correspondence as well as block correspondence in CKA similarity after fine-tuning JE and REC models.** NOTE: Attention layer indices shown (one per ViT block, ViT-B made up of 12 ViT blocks.)



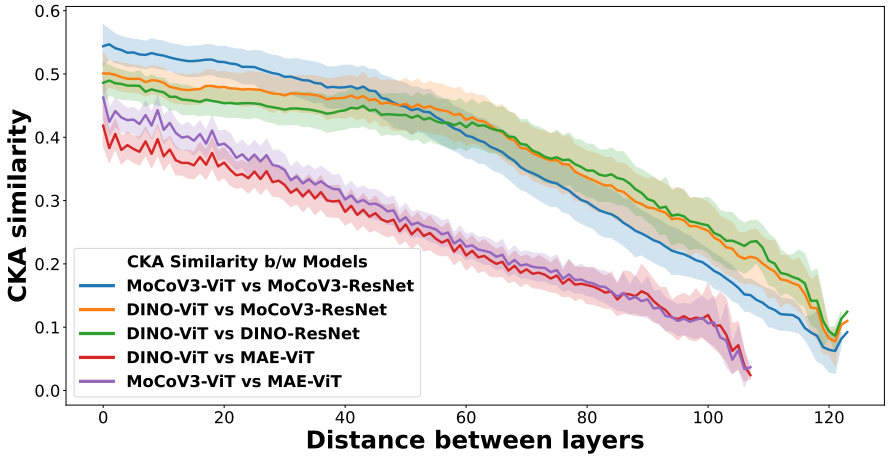
(a) Layer Norm (PT) (b) Self-Attention (PT) (c) Fully-Connected (PT)



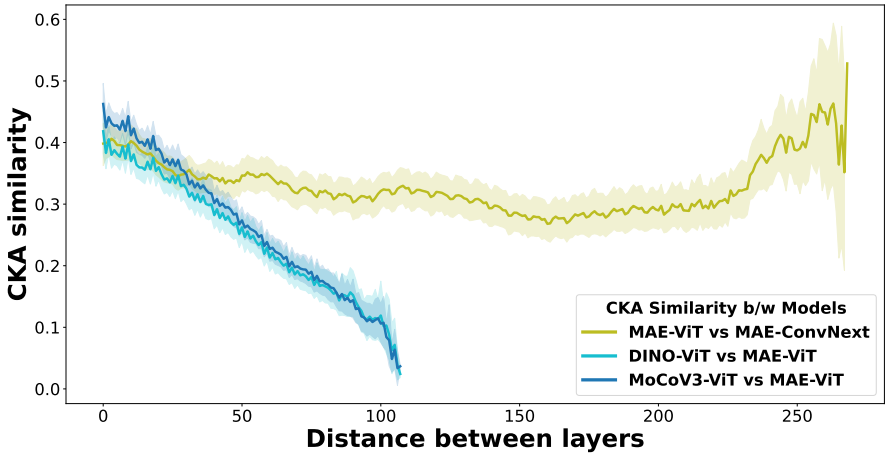
(d) Layer Norm (FT) (e) Self-Attention (FT) (f) Fully-Connected (FT)

Figure 10: CKA similarity between MoCo-V3 and MAE before and after fine-tuning by layer type.

E SUPPLEMENTARY RESULTS: DOES OBJECTIVE OR ARCHITECTURE DRIVE REPRESENTATIONAL STRUCTURE?



(a) JE objectives across architectures



(b) REC objectives across architectures

Figure 11: CKA similarity vs layer distance across ViT and CNN architectures. **Similar SSL objectives yield similar representations even across models with very different architectures.**

We visualize above the results discussed in Section 2.1, showing that CKA values across CNN-ViT model pairs trained with the same self-supervised objective tend to be higher than CKA values acros ViT-ViT model pairs trained with different types of objectives.

F SUPPLEMENTARY RESULTS: HOW DO REPRESENTATIONAL DIFFERENCES MANIFEST WHEN UTILIZING SELF-SUPERVISED FEATURES FOR CLASS PREDICTIONS

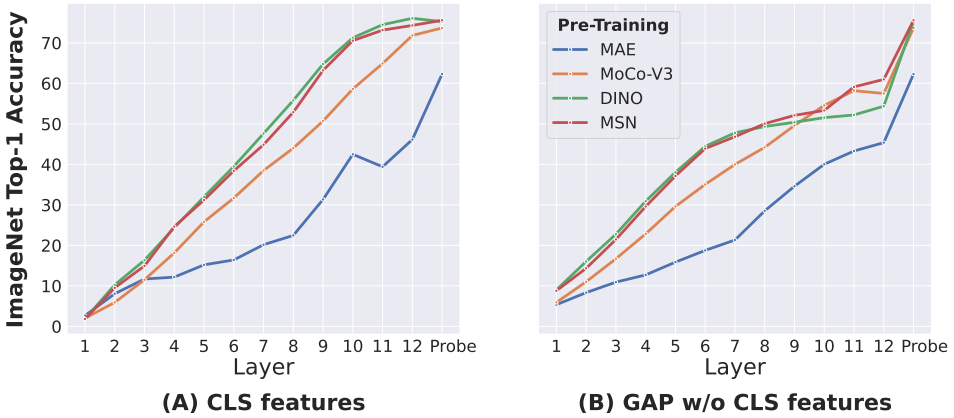


Figure 12: 20-Nearest Neighbour classification accuracy across ViT blocks and linear probe. **Class separability starts to decrease in MAEs after the first few ViT layers. The `CLS` token features in the last ViT layers of JE models contain a significant amount of class information**.

We visualize above the results discussed in Section 2.1, showing that class separability starts to diverge early between JE and REC trained ViTs (by ViT block 3). We also observe that the last layer `CLS` token features in JE ViTs contain all information necessary for classification, as demonstrated by their nearest-neighbor classification accuracy being comparable to linear probe.

We also consider whether the differences in representational similarity and in class separability across JE and REC models translates to the class predictions made by these models. While we have observed higher linear and k-NN transfer performance in JE models in Section 2.1, we do not know whether similar representations and performance are driven by consistent object classification results, or inconsistent results across different classes. In order to evaluate this, we consider the Kendall's Tau rank correlation coefficient of the top-5 and top-10 class predictions made across the ImageNet validation set from MoCo-V3, DINO, and MAE in Fig. 13. 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 predictions. We also calculate the F-1 score of top-1 predictions for DINO and MoCo-V3 (0.93) and confirm that it is higher than the F-1 score for MAE and MoCo-V3 (0.88). Our results verify that not only does the training objective determine representation content, similar pre-training objectives lead to features that represent different object classes similarly leading to class predictions which are also right and wrong in similar ways.

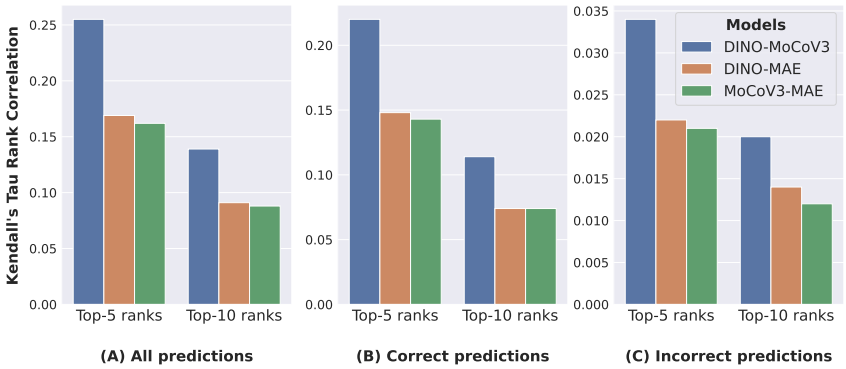
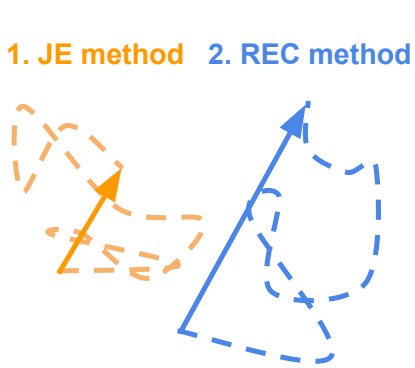
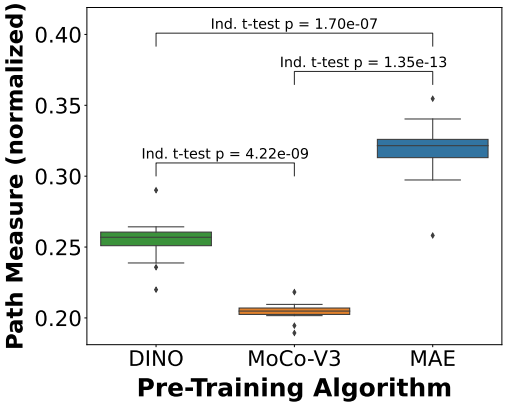


Figure 13: Kendall's Tau rank correlation of linear probe ranks (Top-5 and Top-10 ranks averaged across ImageNet val set). **JE models generate more similar rankings across all predictions** (5A), **and are also incorrect in similar ways** (5C).

G SUPPLEMENTARY RESULTS: A MECHANISTIC UNDERSTANDING OF INCREASED SIMILARITY



(a) Sketch visualizing difference in fine-tuning dynamics of JE and REC models



(b) Fine-Tuning Path efficiency

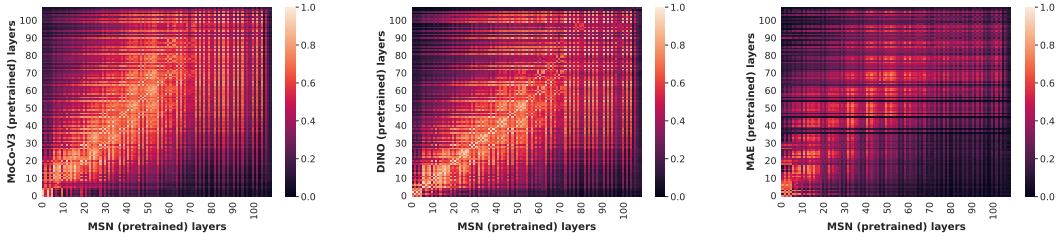
Figure 14: Fine-tuning dynamics of attention layers of SSL ViTs. **MAE fine-tuning follows a more efficient path integral, and attention layer parameters converge more directly towards new values. On the contrary, JE parameters do not follow an efficient path, and go through much higher displacement relative to the actual distance covered in parameter space.**

We visualize above in Figure G the results discussed in Section 2.3 demonstrating that MAE fine-tuning follows a more efficient path in the ViT parameter space during fine-tuning.

H Comparisons of Reconstruction-Based and Joint-Embedding Learning with Masked Siamese Networks

We also compare the representations in JE and REC ViTs to a training procedure that incorporates elements of both: Masked Siamese Networks (MSN) (Assran et al., 2022b). Masked Siamese Networks use a joint embedding approach similar to DINO Caron et al. (2021) as their objective, however they also sample input patches from the image like MAE He et al. (2022) for learning their anchor view embeddings in the joint embedding framework.

We hypothesize that since the training objective of MSN does not invoke reconstruction-based losses, the representations learned will be similar to joint-embedding approaches despite their use of masking-based feature learning. Indeed, our representation similarity analysis in Figure 15 shows that pre-trained MSN representations are much more similar to pre-trained JE representations (DINO, MoCo-V3) than REC representation (MAE). Thus, we conclude that the reconstruction based objective plays a much stronger role in the features learned by MAE versus the modelling of masked image features which MSN shares with MAEs, while the JE objective of modelling similarity between pairs of views dominates features learned by MSN.

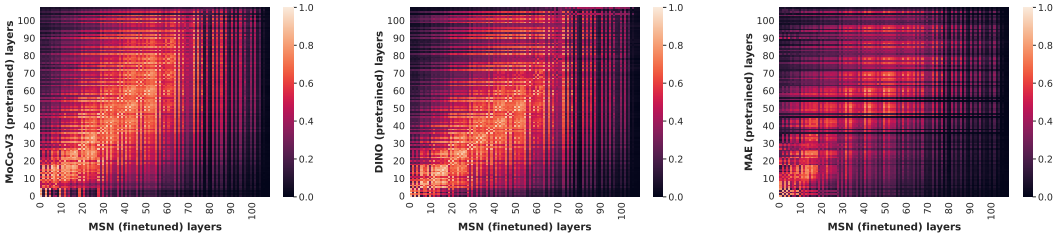


(a) MoCo-V3 (PT) vs MSN (PT) (b) DINO (PT) vs MSN (PT) (c) MAE (PT) vs MSN (PT)

(d) MoCo-V3 (FT) vs MSN (PT) (e) DINO (FT) vs MSN (PT) (f) MAE (FT) vs MSN (PT)

Figure 15: CKA similarity between pre-trained and fine-tuned JE and REC models and a pre-trained MSN model.

Fine-tuning ViTs pre-trained with MSN also gives results consistent with fine-tuning JE models as outlined in Section 2.3. Fine-tuned MSN models continue to remain similar to pre-trained and fine-tuned JE ViTs, as well as fine-tuned REC ViTs, as shown in in Figure 16.



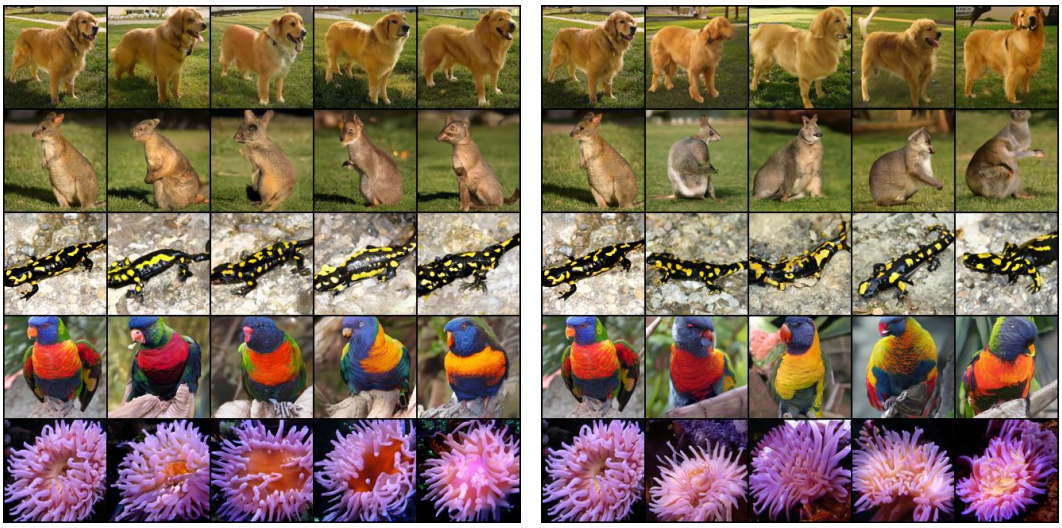
(a) MoCo-V3 (PT) vs MSN (FT) (b) DINO (PT) vs MSN (FT) (c) MAE (PT) vs MSN (FT)

(d) MoCo-V3 (FT) vs MSN (FT) (e) DINO (FT) vs MSN (FT) (f) MAE (FT) vs MSN (FT)

Figure 16: CKA similarity between pre-trained and fine-tuned JE and REC models and a fine-tuned MSN model.

I ADDITIONAL RCDM EXAMPLES

We visualize RCDM[2] samples conditioned on a fine-tuned MAE model below to demonstrate that fine-tuning imparts new invariances (like horizontal flip invariance) that improve transfer performance on classification.

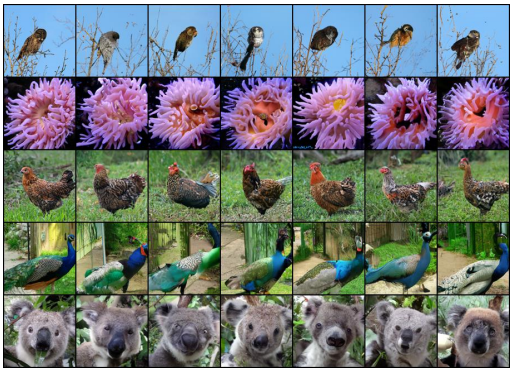


(a) MAE (PT) (b) MAE (FT)

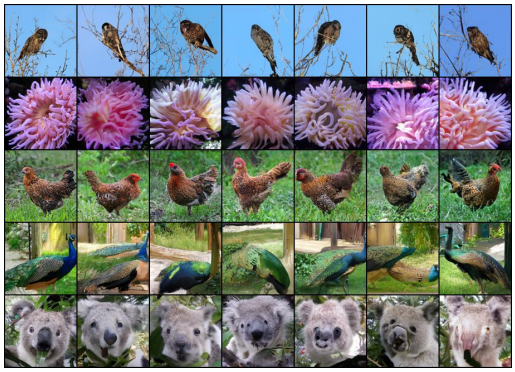
Figure 17: Visualization of samples generated from an RCDM[2] conditioned on fine-tuned MAE and trained on the face-blurred version of ImageNet (Yang et al., 2022). **Unlike pre-trained MAEs, fine-tuned MAE features generate objects oriented differently (horizontally) to the source image, demonstrating that these features are invariant to horizontal flips.**

We also provide additional RCDM samples visualized for each of the 4 models (Pretrained: MAE, DINO, MoCo-V3 and Finetuned: MAE) for readers to identify additional invariances.

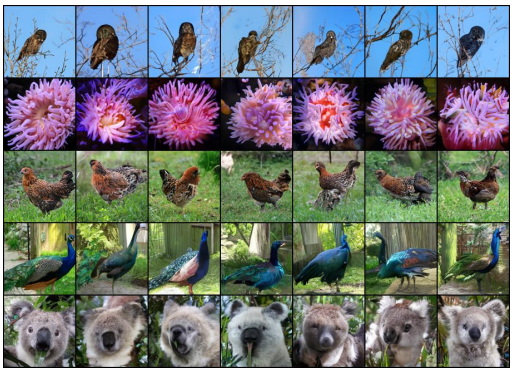
[2]We train RCDM on the face-blurred version of ImageNet (Yang et al., 2022), which enhances privacy.



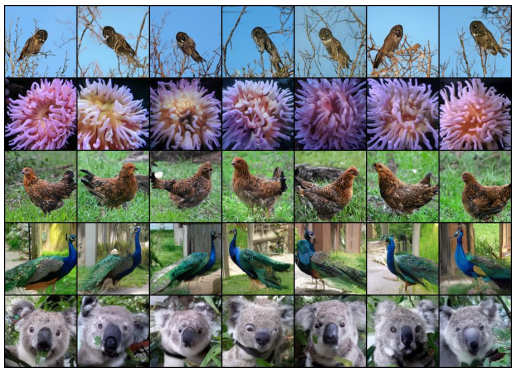
(a) MAE (PT)



(b) MoCo-V3 (PT)



(c) MAE (FT)



(d) DINO (PT)

Figure 18: Additional RCDM^2 samples generated using representations from pre-trained MAE, DINO, MoCo-V3 and fine-tuned MAE when training RCDM on the face-blurred version of ImageNet (Yang et al., 2022).