WHAT DOES POST-TRAINING PRESERVE IN SELF-SUPERVISED REPRESENTATIONS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning produces transferable visual representations, yet it remains unclear which properties are preserved during post-training. We study how full fine-tuning and LoRA reshape self-supervised vision models across network depth and within their final embedding spaces. Our evaluation covers 39 model variants and more than 64,000 images across seven widely used benchmarks: CIFAR-10, CIFAR-100, DTD, Flowers-102, Food-101, OxfordPets, and Tiny-ImageNet, yielding over two million embedding vectors. Our analysis is threefold. First, we use layerwise CKA to trace representational changes throughout the network. Second, we use embedding-level CKA and Gromov-Wasserstein distance to analyze final-embedding alignment and relational geometry. Third, we assess local neighborhood preservation through nearest-neighbor consistency. Our results reveal recurring stability patterns across the tested model configurations and adaptation targets, while alignment and geometric displacement do not always evolve together. In our rank sweep, the highest tested rank produces substantially earlier layerwise divergence than the lowest tested rank, suggesting that rank can affect the depth-wise localization of adaptation. Overall, our findings indicate that different notions of representational preservation do not always evolve together under post-training. These structured forms of specialization are shaped by model family and adaptation capacity, motivating future work on controlling where and how pretrained representations are reorganized. The code used for all our experiments is released in the supplementary material.

1 INTRODUCTION

Large-scale pretraining (Chen et al., 2020; Caron et al., 2021; He et al., 2022; Oquab et al., 2023; Radford et al., 2021) provides general-purpose visual representations, while full fine-tuning and low-rank adaptation (LoRA) (Hu et al., 2021) adapt them to downstream tasks. Accuracy measures performance on the target task but does not reveal how post-training reorganizes the pretrained representation. LoRA is primarily motivated by computational and storage efficiency, but it is also expected to preserve more of the pretrained representation than full fine-tuning. Existing comparisons mainly examine forgetting (Biderman et al., 2024), weight-space structure (Shuttleworth et al., 2026), or individual notions of representational similarity. Weight-space analyses establish that LoRA and full fine-tuning are not equivalent at the parameter level, but do not determine whether pretrained representations, relational geometry, or local neighborhoods are preserved. We therefore study adaptation directly in representation space, across network depth, embedding geometry, and neighborhood structure.

Our study covers 39 model variants and more than 64,000 images across seven datasets. Global alignment, relational geometry, and local neighborhood preservation sometimes lead to different conclusions. On the two backbones evaluated under both regimes, LoRA preserves high matched-depth similarity until later blocks than full fine-tuning, although the subsequent full-fine-tuning profile remains backbone-dependent. On MocoV3, both regimes produce comparable local changes on the datasets used for post-training, while LoRA preserves more local structure on the remaining datasets. Across the tested configurations, local relationships change more on the datasets used for post-training, while nearby samples become more consistent with the downstream class labels. Rank analyses on MocoV3, MAE, and I-JEPA suggest that greater LoRA capacity can produce greater and earlier representational change, with high-rank outcomes also shaped by optimization.

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

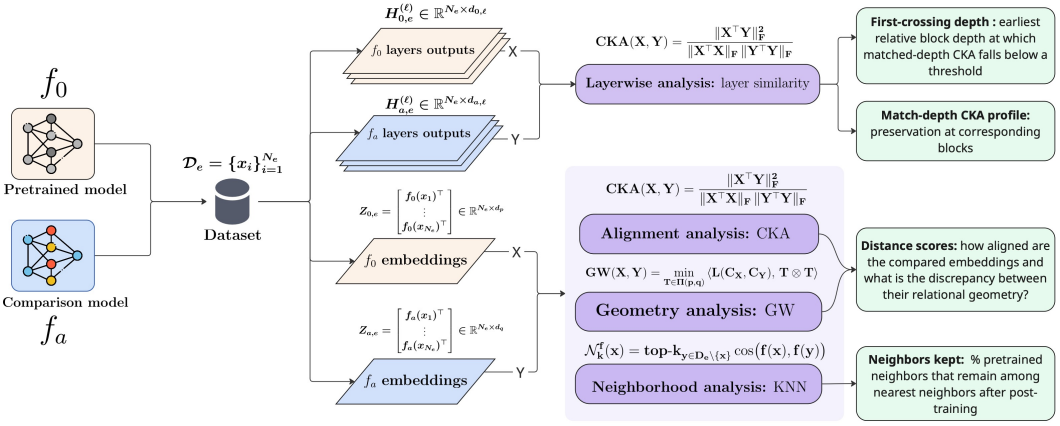


Figure 1: **Overview of the analysis.** Pretrained and comparison models from the same family are compared on identical images. Layer-wise CKA characterizes representational change across network depth. Final-embedding CKA, Gromov–Wasserstein distance, and nearest-neighbor consistency evaluate alignment, relational geometry, and local neighborhood preservation, respectively.

2 Related Work

Self-supervised visual representations. Visual representations are pretrained with contrastive and clustering objectives (Chen et al., 2020; He et al., 2020; Caron et al., 2021), masked reconstruction (He et al., 2022), latent and feature prediction (Baevski et al., 2022; Assran et al., 2023), self-distillation (Oquab et al., 2023) and large-scale language supervision (Radford et al., 2021). All yield transferable features, but they differ substantially in layerwise organization and transfer behavior. In this study, we focus on what happens to this pretrained structure after supervised adaptation.

What fine-tuning changes. Transfer learning analyses of feature reuse, representation similarity and parameter space behavior show that the benefits of pretraining are not reducible to downstream accuracy (Neyshabur et al., 2020). Full fine-tuning can distort useful pretrained features and hurt out-of-distribution (OOD) generalization even as in-distribution accuracy improves, an effect mitigated by linear-probe initialization (Kumar et al., 2022). Interpolating pretrained and fine-tuned weights can recover robustness lost during adaptation (Wortsman et al., 2022), and selectively updating parts of the network can outperform full fine-tuning under distribution shift (Lee et al., 2022). These results expose a stability plasticity tension: adaptation must acquire task-specific information without needlessly overwriting reusable pretrained structure. Rather than proposing a new fine-tuning algorithm, we measure where representational change occurs and what kind of structure survives.

Parameter-efficient fine-tuning (PEFT). PEFT restricts the adaptation of large backbones to few trainable parameters, as in LoRA's low-rank updates to frozen pretrained weights Hu et al. (2021) or, in vision, prompt tuning Jia et al. (2022), lightweight adapters Chen et al. (2022) and feature scaling and shifting Lian et al. (2022). These methods are designed to reduce the computational and storage costs of adaptation while retaining competitive downstream performance.

Comparing neural representations. Representational similarity measures expose changes that downstream accuracy can obscure. Centered Kernel Alignment (CKA) robustly compares activation and embedding structure across layers and models (Kornblith et al., 2019) and has been used to characterize differences between convolutional networks and Vision Transformers (Raghu et al., 2021). CKA, however, summarizes global alignment and cannot tell whether individual samples retain their local neighborhoods; nearest-neighbor overlap captures this sample-level preservation (Doimo et al., 2020). Gromov-Wasserstein distance adds a coordinate-invariant comparison of relational geometry through within-space distances (Peyré et al., 2016). Together, these measures separate global alignment, geometric preservation and local neighborhood stability, and their agreement or disagreement reveals whether adaptation preserves a representation along all three axes or only a subset.

3 METHODOLOGY: ANALYSES CONDUCTED

Our study investigates which properties of self-supervised visual representations are preserved or reorganized during post-training. We consider both full fine-tuning and PEFT, focusing on changes across network depth and within the final embedding space. More specifically, our analyses address the following research questions: **RQ1:** Where across network depth do representations diverge from their pretrained counterparts, and how do the post-training strategy and LoRA rank affect this divergence? **RQ2:** To what extent do post-training variants preserve the alignment and relational geometry of the final pretrained embedding space? **RQ3:** To what extent are local nearest-neighbor relationships preserved after post-training? **RQ4:** How do these patterns vary across SSL families, adaptation targets, and evaluation datasets, and which properties of pretrained representations are preserved together under adaptation versus evolve independently?

As illustrated in Fig. 1, we organize the study into two levels of analysis: layer-wise analyses characterize where representational divergence emerges throughout the network (RQ1), while embedding-level analyses examine the alignment, relational geometry (RQ2), and local neighborhood structure of the final representations (RQ3). RQ4 is evaluated across the layer-wise, embeddings, and nearest-neighbor analyses. Further details on metric computation are provided in App. A.

3.1 LAYER-WISE ANALYSES

We use **layer-wise CKA** to examine how post-training changes intermediate representations across network depth. For every transformer block, we record the token-mean output on a fixed set of images.

Let $\mathcal{D}_e = \{x_i\}_{i=1}^{N_e}$ denote the set of N_e images sampled from evaluation domain e. Let f_0 be a pretrained encoder and f_a a post-training variant derived from it, where a identifies the adaptation target or fine-tuning configuration. For a layer ℓ, token outputs are averaged to obtain one representation vector per image. Stacking these representations gives an activation matrix $H_{0,e}^{(\ell)} \in \mathbb{R}^{N_e \times d_{0,\ell}}$ for the pretrained model and $H_{a,e}^{(\ell)} \in \mathbb{R}^{N_e \times d_{a,\ell}}$ for its post-training variant. Here, $d_{0,\ell}$ and $d_{a,\ell}$ denote the respective hidden dimensions; the number and identity of the images must be identical. For two activation matrices X and Y, centered across samples, CKA is computed as

$$\mathrm{CKA}(X, Y) = \frac{\|X^\top Y\|_F^2}{\|X^\top X\|_F \, \|Y^\top Y\|_F}.$$

We compute CKA between every pair of pretrained and post-trained layers, producing a layer-to-layer similarity matrix. Because each variant retains its parent's architecture and block indexing, the matrix diagonal defines the *matched-depth CKA profile*, which measures preservation at corresponding blocks; off-diagonal entries capture possible cross-depth correspondences.

We summarize divergence by the *first-crossing depth*, defined as the earliest relative block depth at which matched-depth CKA falls below a threshold τ. Relative depth normalizes the block index by the final block index, enabling comparisons across architectures. If no crossing occurs, we report it explicitly; because CKA may subsequently recover, this measure does not imply permanent divergence. We use these profiles and crossing depths to compare LoRA with full fine-tuning, analyze LoRA rank at a fixed α/r, and contrast evaluations on adaptation and non-adaptation domains.

3.2 EMBEDDING-LEVEL ANALYSES

The layer-wise analysis characterizes where representational change occurs, but does not fully describe the structure of the resulting final embeddings. We therefore analyze the final representations through complementary global and local views.

Let f_0 and f_a denote any two encoders under comparison on the aforementioned datasets. They may correspond to two pretrained models, a pretrained model and one of its post-training variants, or two post-training variants. For each image x_i, the encoders produce final embeddings $f_p(x_i) \in \mathbb{R}^{d_p}$ and$f_q(x_i) \in \mathbb{R}^{d_q}$, where d_p and d_q are the respective embedding dimensions. Applying both encoders to the same evaluation images gives the two matrices below; row i of both matrices corresponds to the same image x_i.

$$Z_{0,e} = \begin{bmatrix} f_0(x_1)^\top \\ \vdots \\ f_0(x_{N_e})^\top \end{bmatrix} \in \mathbb{R}^{N_e \times d_p}, \qquad Z_{a,e} = \begin{bmatrix} f_a(x_1)^\top \\ \vdots \\ f_a(x_{N_e})^\top \end{bmatrix} \in \mathbb{R}^{N_e \times d_q}.$$

CKA and Gromov-Wasserstein. Given two embedding matrices $Z_{0,e}$ and $Z_{a,e}$ computed on the same evaluation dataset, we use CKA to measure the alignment of their sample-wise similarity structures and Gromov-Wasserstein distance to measure the discrepancy between their relational geometries. The embedding dimensions d_0 and d_a need not be equal for CKA or Gromov-Wasserstein.

Nearest-neighbor Consistency. Nearest-neighbor consistency measures whether individual images remain surrounded by the same samples after post-training. In the subsequent explanation, f_0 denotes a pretrained encoder, f_a one of its post-training variants, and $\mathcal{D}_e = \{x_i\}_{i=1}^{N_e}$ an evaluation dataset containing N_e images. For a query image $x \in \mathcal{D}_e$, we define its neighborhood under an encoder f as the k other images in the same evaluation set whose embeddings have the highest cosine similarity with $f(x)$:

$$\mathcal{N}_k^f(x) = \text{top-k}_{y \in \mathcal{D}_e \setminus \{x\}} \cos\big(f(x), f(y)\big).$$

The query itself is excluded. We use $k = 20$ in the main analysis and evaluate sensitivity to other values of k separately. For the nearest neighbors consistency, we report:

(1) Neighbors kept. For each query, we measure the proportion of its pretrained neighbors that remain among its k nearest neighbors after post-training:

$$K_k(x; f_0, f_a) = \frac{\left|\mathcal{N}_k^{f_0}(x) \cap \mathcal{N}_k^{f_a}(x)\right|}{k}.$$

The score equals one when the neighborhood is unchanged and zero when all neighbors are replaced. We average this quantity over all query images in $\mathcal{D}_e$ to obtain the dataset-level score $\overline{K}_{a,e}$. For reference, a uniformly random neighborhood retains an expected fraction $k/(N_e - 1)$ of a fixed reference neighborhood. Although being a monotonic transformation of the Jaccard index, we report it as a proportion because it directly expresses the share of the pretrained neighborhood preserved.

(2) Seed consistency. Training randomness can modify local neighborhoods even when the post-training configuration remains unchanged. When replicate runs are available, we compute *seed consistency* as the average neighbors-kept score between post-trained encoders that differ only in their training seed. If $\mathcal{S}$ is the set of available seeds, we denote this baseline by

$$S_{a,e} = \operatorname*{avg}_{s<s'} \overline{K}_{k,e}\left(f_a^{(s)}, f_a^{(s')}\right).$$

A high value of $S_{a,e}$ indicates that neighborhoods are stable across seeds. We use this quantity as the seed-noise baseline and report it when seed replicates are available.

(3) Class purity. We measure normalized class purity, averaged over queries as $\overline{P}_{f,e}$. Comparing $\overline{P}_{f_0,e}$ and $\overline{P}_{f_a,e}$ indicates whether neighborhood reorganization makes the local representation more or less class coherent. Further details about the purity score are available in App. A.

(4) Adaptation and evaluation domains. For a model adapted on dataset a and evaluated on dataset e, we distinguish *on-target* evaluations, for which $a = e$, from *elsewhere* evaluations, for which $a \neq e$. The aggregation method over multiple seeds is available in App. A.

3.3 Comparison Design

Within-family model-to-model comparisons. For each SSL family, we compare all compatible pairs of model variants belonging to that family. Depending on the available *post-training variants*, these include pretrained-to-pretrained (e.g, ijepa-vitb16 vs ijepa-vith16), pretrained-to-fine-tuned, pretrained-to-adapted, fine-tuned-to-adapted, and adapted-to-adapted comparisons. We consider variants compatible only when they share the same architecture class; in particular, we exclude ViT–ConvNeXt and ViT–ResNet comparisons.

Exact parent-child comparisons. Holding the pretrained backbone fixed avoids attributing architectural or pretraining differences to post-training. These comparisons therefore provide a more targeted characterization of the alignment and geometric displacement associated with fine-tuning or adaptation. An example of such comparison would be *ijepa-vitb16* vs *ijepa-vitb16-lora-dtd*.

4 EXPERIMENTS

Our experiments are designed to isolate the effect of post-training on the representation itself. We first describe the shared evaluation protocol, then the datasets and model selection, then the adaptation procedure. Details of our computational infrastructure are provided in App. C.

Datasets selection. We evaluate the representations on seven widely used image-classification benchmarks: CIFAR-10 and CIFAR-100 Krizhevsky et al. (2009), Tiny-ImageNet Le et al. (2015), DTD Cimpoi et al. (2014), Flowers-102 Nilsback & Zisserman (2008), Food-101 Bossard et al. (2014), and Oxford-IIIT Pet Parkhi et al. (2012). CIFAR-10, CIFAR-100, and Tiny-ImageNet cover general object recognition at different levels of granularity, DTD focuses on texture recognition, and Flowers-102, Food-101, and Oxford-IIIT Pet involve fine-grained category distinctions across flowers, food dishes, and pet breeds, respectively. Together, these benchmarks provide broad coverage across visual domains, task types, and levels of classification granularity.

Model selection. We consider five widely used self-supervised vision model families: I-JEPA Assran et al. (2023), MocoV3 Chen et al. (2021), DINOv3 Siméoni et al. (2026), data2vec2 Baevski et al. (2023), and MAE He et al. (2022). Some models such as SimCLR Chen et al. (2020) were not included because the required pretrained weights were inaccessible. For each selected family, we include the available pretrained backbones together with their fine-tuned and LoRA adapted variants. We also generated additional models through LoRA adaptation on selected downstream datasets. This selection yields 39 pretrained and post-training model variants. The complete list of pre-trained models and their post-training configurations is provided in App. B.

LoRA adaptation. Each pretrained backbone is adapted independently on three downstream targets, Flowers-102, DTD and Oxford-IIIT Pet, chosen because they emphasize different visual information: fine-grained shape and color, texture, and breed-level structure. Keeping the backbone fixed while varying only the target lets us separate what an adapter learns that is generic from what is specific to its domain. Full details on adapter placement, initialization, training procedure, and hyperparameter selection are provided in App. D.

5 RESULTS

In this section, unless stated otherwise, comparisons use all seven evaluation datasets. Analyses that require an adaptation target use DTD, Flowers-102, and Oxford-IIIT Pet.

5.1 LAYER-WISE ANALYSIS (RQ1)

Divergence begins later under LoRA. MocoV3 ViT-B and MAE ViT-L/16 are the two backbones evaluated under both LoRA and full fine-tuning (Tab. 1). At $\tau = 0.95$, the mean first-crossing depth among profiles that cross is 0.70 for LoRA versus 0.00 for full fine-tuning on MocoV3, and 0.81 versus 0.24 on MAE. One MocoV3 LoRA profile never crosses this threshold. At $\tau = 0.99$, full fine-tuning crosses at the first block on both backbones, compared with mean relative depths of 0.15 and 0.40 for LoRA. Across these matched comparisons, LoRA therefore retains high matched-depth similarity until later in the network.

The depth profile of full fine-tuning depends on the backbone. For both backbones, LoRA's matched-depth CKA decreases toward the output. Under full fine-tuning, MocoV3 reaches its lowest similarity in the early blocks and partially recovers toward the output, whereas MAE continues to decline. Final-layer similarity alone cannot reveal this difference in where change occurs. Because the regime comparison covers only two backbones, we treat these profile shapes as observations

about the tested models, not as a general rule for full fine-tuning. The full layer-to-layer CKA matrices for these backbone adaptation pairs are shown in Fig. 2.

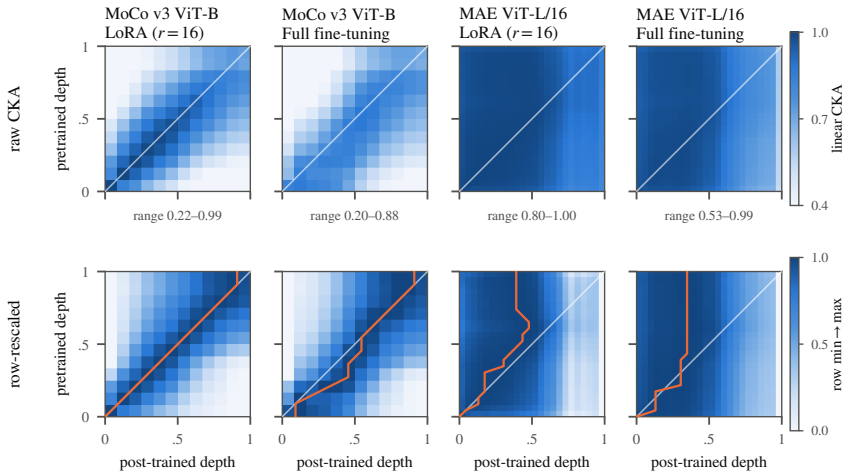


Figure 2: **Layer-to-layer CKA after post-training.** Each matrix compares pretrained blocks (rows) with post-trained blocks (columns) on the same evaluation images. The columns show MocoV3 ViT-B and MAE ViT-L/16 under LoRA ($r = 16$) and full fine-tuning. Axes indicate relative block depth; the white line marks matched depths. The top row displays linear CKA on a shared color scale from 0.4 to 1. Values below 0.4 are clipped in color, while the ranges printed beneath the panels report the unclipped matrix extrema. In the bottom row, each matrix row is independently rescaled to $[0, 1]$; the orange trace marks its highest-CKA column. This rescaling highlights within-row differences and does not preserve absolute CKA differences across rows or panels.

Table 1: **First-crossing depth.** Earliest relative depth at which matched-depth CKA falls below τ, averaged over the profiles that cross, with the count that never do. Each row aggregates nine profiles (three adaptation targets $\times$ three evaluation datasets) at r=16. Lower values indicate that divergence begins closer to the input. A superscript gives the number of profiles that never cross that threshold, which are excluded from the mean.

Backbone	**Regime**	τ=0.99	τ=0.95	τ=0.90	τ=0.80
MAE ViT-L/16	LoRA	0.40	0.81	0.86^{2}	0.90^{4}
DINOv3 ViT-H+/16	LoRA	0.29	0.71^{4}	0.87^{6}	0.84^{8}
MoCo v3 ViT-B	LoRA	0.15	0.70^{1}	0.82^{4}	0.91^{6}
I-JEPA ViT-H/14	LoRA	0.09	0.50	0.74	0.90^{1}
data2vec2 ViT-L/16	LoRA	0.05	0.27	0.54	0.83
MAE ViT-L/16	Full FT	**0.00**	0.24	0.48^{1}	0.89^{1}
MoCo v3 ViT-B	Full FT	**0.00**	**0.00**	**0.00**	**0.03**

Tab. 1 quantifies the onset of divergence. At $\tau = 0.99$, full fine-tuning crosses at the first block for both MoCo v3 and MAE. At lower thresholds, MoCo v3 continues to cross at or near the input, whereas MAE crosses progressively later, at relative depths of 0.24 and 0.48 for $\tau = 0.95$ and $\tau = 0.90$, respectively. LoRA on the same backbones holds above 0.95 until 0.70 and 0.81 of relative depth, and on MocoV3 four of nine profiles never fall below 0.90 at all. Reporting non-crossings separately matters here: folding them onto the final block would place the best-preserved and the worst-preserved profiles at adjacent values.

Robustness across seeds. The layerwise patterns are stable across replicate runs. Across seeds 42, 43, and 44, the standard deviation of mean matched-depth CKA ranges from 0.0006 to 0.0069, while the largest block-level standard deviation ranges from 0.0078 to 0.0427. These variations represent only 3–9% of the corresponding profile ranges, indicating that the observed depth-wise structure is substantially larger than seed-induced variation. Full results are provided in App. E.2.

5.2 EMBEDDING-LEVEL CKA AND GROMOV-WASSERSTEIN ANALYSIS (RQ2)

Within family vs Others. Tab. 2 shows that post-training reduces CKA alignment across all families. The effect on relational geometry is more family-dependent: GW increases substantially for data2vec2 and MAE, increases moderately for I-JEPA, and decreases for MoCoV3. MoCoV3 best preserves both relational geometry and representational alignment, achieving the lowest original-to-variant GW and the highest CKA. Although data2vec2 and I-JEPA are both representation-predictive SSL families, they respond differently to post-training: data2vec2 undergoes substantially greater geometric restructuring, whereas I-JEPA better preserves its pretrained relational geometry. Overall, our results indicate that (1) post-training sensitivity varies substantially across the tested family-specific configurations, and (2) reduced embeddings alignment does not necessarily imply an equivalent deformation of relational geometry. As a reference, comparisons between different seeds of the same post-training configuration yield CKA values close to 0.99 and Gromov-Wasserstein discrepancies on the order of 10^{-9}, indicating near-perfect representational and geometric agreement.

Table 2: **Pretrained models versus post-training variants.** Relative changes express the Original $\leftrightarrow$ Variant value with the within-family pretrained Original $\leftrightarrow$ Original value as the reference. Lower GW indicates greater preservation of relational geometry, higher CKA indicates stronger representational alignment. DINOv3 is excluded because only one compatible variant is available.

Metric	Model family	Original $\leftrightarrow$ Original	Original $\leftrightarrow$ Variant	Relative change
GW $\downarrow$	data2vec2	$0.0382 \pm < 0.000$	0.1932 ± 0.0731	5.06× higher
	I-JEPA	**0.0107 ± 0.0030**	0.0158 ± 0.0051	1.48× higher
	MAE	0.0237 ± 0.0143	0.0802 ± 0.0424	3.38× higher
	MocoV3	0.0125 ± 0.0072	**0.0104 ± 0.0046**	**1.20× lower**
CKA $\uparrow$	data2vec2	0.2508 ± N/A	0.1800 ± 0.0961	1.39× lower
	I-JEPA	0.6226 ± 0.0580	0.4615 ± 0.1015	1.35× lower
	MAE	0.6083 ± 0.1803	0.4707 ± 0.1309	1.29× lower
	MocoV3	**0.6850 ± 0.0529**	**0.6096 ± 0.0466**	**1.12× lower**

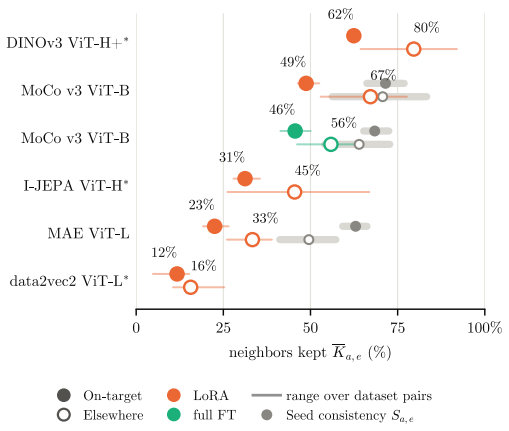
Parent-child analysis. Tab. 3 reveals strong family- and target-dependent effects of adaptation. MoCoV3 preserves its pretrained representation most strongly, combining the lowest mean GW with the highest and least variable mean CKA. By contrast, data2vec2 exhibits the largest geometric displacement and the lowest retained alignment. Although I-JEPA and MAE attain comparable mean CKA, MAE undergoes substantially greater geometric restructuring, confirming that alignment and relational geometry capture complementary effects. The adaptation target also has a consistent influence. Flowers-102 produces the highest parent-child CKA, while also yielding the largest GW for data2vec2, MAE, and MoCoV3; I-JEPA is the only exception, with DTD producing a slightly larger GW. Flowers-102 adaptation therefore produces substantial relational displacement while retaining comparatively strong alignment with the pretrained parent. Consistent with the layer-wise analysis in Sec. 5.1 and the adapted-to-adapted comparisons in App. E.1, MocoV3 exhibits the strongest preservation according to both CKA and GW.

5.3 NEAREST NEIGHBORS ANALYSIS (RQ3)

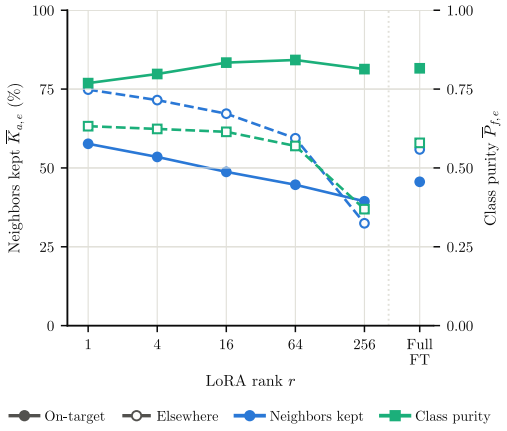
Per family. In Fig. 3a, we plot the per-family proportion of neighbors kept $\overline{K}_{a,e}$ on-target and elsewhere, alongside the seed consistency $S_{a,e}$ where available. The clearest pattern is the ordering across pretrained backbones: neighbors-kept score decreases monotonically from DINOv3 to data2vec2, both on-target and elsewhere, pointing to noticeably different levels of representational stability under adaptation. Within every backbone, the elsewhere score consistently exceeds the on-target one: whether adapting with LoRA or full fine-tuning, adaptation concentrates its effect on the target dataset and leaves elsewhere representations comparatively untouched. This is confirmed by the seed-consistency baseline $S_{a,e}$ (available for three of the six configurations): on-target scores sit well below it, while elsewhere scores stay close, confirming that adaptation rewrites the on-target representation measurably more than seed-to-seed noise alone would. Numerical results, including purity, are available in App. E.3.

Table 3: Gromov–Wasserstein distance (GW) and Centered Kernel Alignment (CKA) between the final embeddings of each pretrained parent and its adapted variants, averaged over five seeds.

Model family	Adaptation target	GW	Mean GW	CKA	Mean CKA
data2vec2	DTD	**0.1119**	0.13 ± 0.02	0.2745	0.28 ± 0.07
	Flowers-102	0.1643		**0.3728**	
	OxfordPets	0.1184		0.1879	
I-JEPA	DTD	0.0140	0.01 ± 0.00	0.4498	0.54 ± 0.10
	Flowers-102	0.0125		**0.6774**	
	OxfordPets	**0.0106**		0.4832	
MAE	DTD	0.0578	0.06 ± 0.01	0.4536	0.54 ± 0.09
	Flowers-102	0.0787		**0.6655**	
	OxfordPets	**0.0604**		0.5123	
MocoV3	DTD	0.0070	$<$ **0.01 ± 0.00**	0.6675	**0.67 ± 0.03**
	Flowers-102	0.0094		**0.7184**	
	OxfordPets	**0.0064**		0.6333	



(a) Neighbors-kept score $\overline{K}_{a,e}$ after post-training, on-target and elsewhere, alongside the seed-consistency baseline $S_{a,e}$ (absent, *, for single-seed configurations).



(b) Effect of LoRA rank on neighborhood preservation (MocoV3 ViT-B): neighbors-kept $\overline{K}_{a,e}$ and class purity $\overline{P}_{f,e}$, on-target and elsewhere, across rank r, with a full fine-tuning reference.

Figure 3: Neighborhood preservation after post-training, per family (left) and by LoRA rank (right).

Per adaptation rank. Fig. 3b shows $\overline{K}_{a,e}$ and normalized class purity $\overline{P}_{f,e}$ as a function of LoRA rank r for MocoV3 ViT-B/16, alongside the corresponding full fine-tuning values for reference. Neighbors-kept decreases with rank in both, showing that higher ranks rewrite the representation more thoroughly. Meanwhile, purity rises on-target and drops elsewhere. One possible interpretation is that low-rank adaptation concentrates changes in directions sufficient for the downstream task, whereas high-rank adaptation permits more extensive reorganization. Our measurements do not directly identify the affected feature directions. Other results in this section use the default rank ($r = 16$). Numerical results are in Tab. 10.

Per adaptation dataset. Flowers-102 stands out across tasks: neighborhoods change the least, both on-target and elsewhere, while DTD shows the opposite behavior. Comparing neighborhoods between adapted embeddings tells the same story: DTD and Pets share the fewest kept neighbors. Adapted embeddings retain at least 25% of their neighbors in every tested comparison, indicating a non-negligible shared local structure across adaptation targets. Numerical results are in Tab. 11.

Neighbor overlap reproduces the family ordering of Sec. 5.2 and supports its reading of Flowers-102 adaptation, since the target that displaces relational geometry the most is also the one that preserves local neighborhoods best.

6 Ablation: LoRA rank sweep

Table 4: **LoRA rank sweep** on MoCoV3 ViT-B, averaged over the three adaptation targets, with α=2r throughout, with variation reported across targets. We define *trunk CKA* as the mean matched-depth CKA across all transformer blocks, from relative depth 0 to 1, inclusive.

Configuration	Trainable (M)	Top-1	CKA trunk	CKA final	$\overline{K}_{a,e}$
r=1	0.20	88.4	0.944 ± 0.021	0.829 ± 0.056	0.612 ± 0.090
r=4	0.64	88.6	0.924 ± 0.015	0.765 ± 0.044	0.583 ± 0.100
r=16	2.41	**88.9**	0.911 ± 0.020	0.725 ± 0.062	0.476 ± 0.110
r=64	9.49	87.8	0.877 ± 0.027	0.650 ± 0.074	0.492 ± 0.100
r=256	37.80	84.6	0.804 ± 0.036	0.525 ± 0.037	0.313 ± 0.081
Full FT	85.8	87.0	0.798 ± 0.009	0.696 ± 0.062	0.376 ± 0.076

We sweep $r \in \{1, 4, 16, 64, 256\}$ on MocoV3 ViT-B with $\alpha/r = 2$ (Tab. 4). Accuracy is highest at $r = 16$, but only 0.5 points above $r = 1$, and drops to 84.6% at $r = 256$. Trunk and final-layer CKA decrease monotonically with rank, while neighborhood preservation declines overall but is non-monotonic. At $r = 256$, final-layer CKA, neighborhood preservation, and accuracy are all lower than under full fine-tuning. Moreover, the first crossing below $\tau = 0.95$ shifts from relative depth 0.67 at $r = 1$ to 0.09 at $r = 256$, suggesting that high-rank adaptation can affect earlier blocks. This comparison does not establish a monotonic depth-wise shift across the full sweep.

The broader rank analysis (App. E.4) shows similar capacity–preservation trade-offs on MAE and I-JEPA: mean layerwise CKA and neighborhood preservation generally decrease as rank increases. However, several high-rank configurations fail to converge under the learning rate selected at $r = 16$, reinforcing that the observed rank effects depend partly on optimization.

7 Limitations

Our comparison between LoRA and full fine-tuning is limited to two backbones, and our evaluation focuses on image classification. Moreover, the rank sweeps hold α/r and the remaining optimization protocol fixed. The observed effects may therefore reflect both adaptation capacity and rank-dependent optimization dynamics. Finally, our study characterizes representational change rather than its downstream consequences. Relating these measurements to retrieval or transfer performance, and extending the analysis to additional backbones, adaptation protocols, and downstream tasks, are important directions for future work.

8 Conclusion

We examined how post-training reorganizes self-supervised representations across network depth, final-embedding geometry, and local neighborhoods. Across the tested backbones, LoRA generally preserves early representations more strongly than later ones, while on the two backbones evaluated under both regimes, full fine-tuning produces measurable divergence from the earliest blocks. Our rank analyses further suggest that increasing LoRA capacity can extend representational change toward earlier layers. Post-training consistently reduces alignment with the pretrained representation, but the preservation of relational geometry and local neighborhoods varies across configurations. Changes concentrate on the adaptation dataset, where neighborhoods become less stable but more class-coherent. Taken together, our results suggest that representational preservation is multi-dimensional: alignment, relational geometry, and local neighborhood structure do not always evolve together under adaptation. These observations complement prior weight-space analyses by showing where representational change occurs within the network and which properties of the pretrained representation are preserved or reorganized.

AI DISCLOSURE

In this work, we used generative AI tools to assist with the analysis of experimental failure logs and to verify the consistency between our interpretations of specific experiments and their raw results. Additionally, we used generative AI tools to improve the readability of selected passages through minor corrections, rephrasing, and synonym suggestions, and to generate docstrings for software functions.

We did not use generative AI tools to generate datasets, develop theoretical or conceptual frameworks, formulate mathematical claims or proofs, propose or refine hypotheses, design the research methodology or experiments, implement the proposed analysis methods, translate content, clean or reformat datasets, or conduct qualitative or thematic analysis; the remaining required disclosure tasks are not applicable to this work. Generative AI was not used to summarize or survey the literature, identify research gaps, source references, format citations, suggest the paper structure, or propose keywords. All AI-assisted text was reviewed and edited by the authors. Failure-log analyses were checked against the original logs, and consistency checks were verified against the corresponding raw experimental results. Generated docstrings were reviewed against the relevant function implementations and did not determine the behavior of the software.

We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code used for the experiments is provided in the supplementary material. Sections 3 and 4 describe the evaluation datasets, pairwise comparisons, token-mean layer activations, and analysis metrics. App. B lists the model variants. Adapter placement, initialization, rank scaling, and learning-rate selection are documented in App. D, and hardware and random-seed values in App.. C. Paired models are evaluated on identical images. The main nearest-neighbor analysis uses $k = 20$, with sensitivity results for other values of k in Tab. 12. Seed consistency estimates are reported only where replicate runs are available. The rank-specific nearest-neighbor analysis uses a single seed, its reported standard errors reflect variation across queries rather than across training runs.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pp. 1298–1312. PMLR, 2022.

Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International conference on machine learning*, pp. 1416–1429. PMLR, 2023.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

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

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in neural information processing systems*, 35:16664–16678, 2022.

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, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629, 2021. doi: 10.1109/ICCV48922.2021.00950.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Diego Doimo, Aldo Glielmo, Alessio Ansuini, and Alessandro Laio. Hierarchical nucleation in deep neural networks. *Advances in Neural Information Processing Systems*, 33:7526–7536, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9726–9735. IEEE, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 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.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35: 109–123, 2022.

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

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pp. 2664–2672. PMLR, 2016.

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, 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.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *Advances in Neural Information Processing Systems*, 38: 174627–174662, 2026.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2NlGyqNjns. Featured Certification.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7949–7961. IEEE, 2022.

A METRICS COMPUTATION DETAILS

A.1 DATA SPLIT

For all seven datasets, embeddings were computed on held-out evaluation images. We used the official test split of each dataset, except for Tiny-ImageNet, for which we used the validation split because the official test set is unlabeled. Consequently, none of the images used to compute the representation metrics were seen during post-training.

A.2 EMBEDDINGS EXTRACTION

Final embeddings are extracted from the pooled backbone representations prior to both the LayerNorm layer and the classification head. Let

$$\hat{z} = \frac{z}{\|z\|_2}$$

denote the ℓ_2-normalized representation. Embeddings are stored without normalization, while ℓ_2 normalization is applied at evaluation time, i.e., cosine similarities are computed using $\hat{z}$. Pooling follows the linear-probe protocol associated with each method: the class token is used for MAE, MocoV3, and DINOv3, whereas the mean of patch tokens is used for I-JEPA and data2vec2. For the layer-wise analysis, representations are obtained by averaging patch-token embeddings only, excluding all prefix tokens. The number of excluded prefix tokens is 0 for I-JEPA, 1 for MAE, data2vec2, and MocoV3, and 5 for DINOv3, where four register tokens follow the class token. The same extraction and normalization procedure is applied to both the pretrained parent model and its corresponding post-trained variant.

A.3 GROMOV-WASSERSTEIN AND EMBEDDINGS NORMALIZATION

We use GW to assess whether two representation spaces preserve similar relational geometry independently of their coordinate systems and without requiring a predefined correspondence between individual samples.

For each model pair, we initialize a random-number generator with a fixed seed. In each evaluation, we sample without replacement up to 256 embeddings independently from each representation space. Consequently, the source and target batches do not need to contain the same images or follow the same ordering.

We first ℓ_2-normalize each embedding. For every sampled source and target batch, we construct the within-space pairwise Euclidean distance matrices C_s and C_t. To reduce differences in distance scale, each matrix is independently normalized by its maximum value:

$$C_s \leftarrow \frac{C_s}{\max(C_s) + \varepsilon}, \qquad C_t \leftarrow \frac{C_t}{\max(C_t) + \varepsilon},$$

where $\varepsilon = 10^{-12}$ ensures numerical stability. We then assign uniform probability masses p_s and p_t to the sampled embeddings and compute the balanced, non-entropic squared Gromov–Wasserstein discrepancy using the `gromov_wasserstein2` solver from the Python Optimal Transport library, with the squared loss

$$L(a, b) = (a - b)^2.$$

Unless otherwise specified, the remaining solver parameters are left at their library defaults. We report the mean and standard deviation of the GW discrepancy across the batches. Because GW optimizes over couplings between the two sampled distributions, it does not require identity-paired images and is invariant to the ordering of samples within each batch.

A.4 LAYER-WISE CKA

Activations are recorded by a forward hook on each transformer block of the encoder, indexed from the first to the last: 12 for ViT-B, 24 for ViT-H and ViT-H+. the patch embedding, the positional embedding and the final normalization are not hooked and therefore never enter any reported quantity.

Each block output is reduced to one vector per image by averaging the patch tokens and excluding the prefix tokens, whose number is model-specific (none for I-JEPAm one for MAE, data2vec2 and MoCo v3 and five for DINOv3, whose class token is followed by four register tokens. The same reduction is applied to the pretrained parent and to its post-training variant.

Because a post-training variant retains its parent's block count, entry ℓ of the profile compares block ℓ of the two models directly. When two different pretrained models of unequal depth are compared, the profile is resampled at matched relative depths instead.

A.5 Purity score formula

Let $c(x)$ denote the class of query x, and let $n_{c(x),e}$ be the number of images from that class in $\mathcal{D}_e$. For encoder f, purity is defined as:

$$P_k(x; f) = \frac{\left|\left\{y \in \mathcal{N}_k^f(x) : c(y) = c(x)\right\}\right|}{\min\left(k, n_{c(x),e} - 1\right)}.$$

The denominator accounts for classes containing fewer than $k + 1$ images, so that a score of one represents the highest class purity permitted by the class size.

We average this per-query score over all N_e queries in $\mathcal{D}_e$ to obtain the dataset-level purity score:

$$\overline{P}_{f,e} = \frac{1}{N_e} \sum_{x \in \mathcal{D}_e} P_k(x; f).$$

This score is computed separately for the pretrained encoder f_0 and for each post-training variant f_a.

A.6 Score aggregation

Let $\overline{K}_{a,e}$ denote the neighbors-kept score averaged over queries and, where applicable, over training seeds. We aggregate the scores as:

$$\overline{K}_{\text{target}} = \frac{1}{3} \sum_{a} \overline{K}_{a,a}, \qquad \overline{K}_{\text{elsewhere}} = \frac{1}{6} \sum_{a \neq e} \overline{K}_{a,e}.$$

Each adaptation-evaluation dataset pair receives equal weight, regardless of the number of images in the evaluation dataset.

B Model variants details

Our study includes pretrained, fully fine-tuned, and LoRA-adapted variants from five SSL model families: data2vec2, DINOv3, I-JEPA, MAE, and MocoV3. The LoRA variants were adapted on DTD, Flowers-102, or Oxford-IIIT Pet. Selected adapted models were additionally evaluated at multiple LoRA ranks to study the effect of adaptation capacity on representational change.

For the LoRA rank analysis, selected model-target configurations were trained at ranks $r \in \{1, 4, 16, 64, 256\}$ while keeping α/r constant. Rank-specific checkpoints are treated as experimental configurations and are not repeated in Tab. 5.

C Experimental setup

All experiments were conducted on a shared compute infrastructure comprising one NVIDIA RTX 6000 Ada Generation GPU with 49 GB of VRAM and two NVIDIA RTX 5000 GPUs with 32 GB of VRAM each. The system possessed an Intel Xeon w5-3435X (4.70 GHz) CPU featuring 32 threads.

We used various random seeds throughout the experiments: 42, 123, 1234, 10, and 1. Tables 2, 3, and 7 average results over five training seeds. Seed consistency is reported in Tables 8 and 11 only for configurations with replicate runs. Tables 4, 9, and 10 use one seed per configuration.

Table 5: Pretrained backbones and post-training variants included in the study. A model variant is defined by its pretrained parent, post-training regime, and, where applicable, adaptation target. Target-specific adapters are therefore counted as separate variants, whereas seed replicates and rank-specific checkpoints are treated as experimental instances and are excluded from the reported variant count.

Model family	Pretrained backbones	Post-training variants
data2vec2	data2vec2-vitb16 data2vec2-vith14 data2vec2-vitl16	data2vec2-vitb16-ft data2vec2-vith14-ft data2vec2-vitl16-ft data2vec2-vitl16-lora-dtd data2vec2-vitl16-lora-flowers-102 data2vec2-vitl16-lora-oxfordpets
DINOv3	dinov3-vit7b16 dinov3-vith16plus dinov3-convnext-base dinov3-convnext-large	dinov3-vith16-lora-dtd dinov3-vith16-lora-flowers-102 dinov3-vith16-lora-oxfordpets
I-JEPA	ijepa-vitg16-in22k ijepa-vith14-in1k-300e ijepa-vith14-in22k ijepa-vith16-in1k-448px	ijepa-vith14-in1k-300e-lora-dtd ijepa-vith14-in1k-300e-lora-flowers-102 ijepa-vith14-in1k-300e-lora-oxfordpets
MAE	mae-vitb16 mae-vitl16 mae-vith14	mae-vitl16-lora-dtd mae-vitl16-lora-flowers-102 mae-vitl16-lora-oxfordpets mae-vitl16-ft
MoCoV3	mocov3-vit-s mocov3-vit-b mocov3-r50-100ep mocov3-r50-300ep mocov3-r50-1000ep	mocov3-vit-b-lora-dtd mocov3-vit-b-lora-flowers-102 mocov3-vit-b-lora-oxfordpets mocov3-vit-b-ft

D LoRA details

Linear probing freezes the backbone entirely and trains only the classifier on the training split of each target dataset: its CKA against the pretrained model is 1 by construction, which makes it a null condition rather than a competitive baseline. Full fine-tuning updates all backbone parameters and therefore provides a high-capacity reference regime for representational change. LoRA (Hu et al., 2021) lies between them: the pretrained weights are frozen and a trainable low-rank update is added in parallel, so the function changes while the original weights do not.

D.1 Training procedure

Adapter placement. For a frozen linear layer $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA adds a parallel low-rank branch so that the layer computes

$$h = W_0 x + \frac{\alpha}{r} BA\, x, \qquad A \in \mathbb{R}^{r \times d_{\text{in}}},\ B \in \mathbb{R}^{d_{\text{out}} \times r},$$

with A initialized from a kaiming-uniform distribution He et al. (2015) and B initialized to zero, so that the adapted model is exactly the pretrained model at step 0 and any measured representational change is attributable to training. We insert these branches at the (query, key, value) projection, the attention output projection, and both feed-forward projections of every transformer block, leaving patch embedding, positional embeddings and final normalization frozen.

Adapter coverage is not uniform across families, and we state this explicitly because it bears on the cross-family comparison. DINOv3 uses a SwiGLU feed-forward block whose projections do

not match the naming convention of the other four families, so its adapters cover attention only. DINOv3 is consequently the most lightly adapted family in our study, and its apparent advantage in representational preservation should be read as partly a consequence of adapter coverage rather than of the pretraining objective alone. Within family comparisons, LoRA against full fine-tuning, and the rank sweep are unaffected, since the coverage is held fixed there.

Table 6: LoRA coverage per backbone at r=16. Sites counts the linear layers wrapped with a low-rank branch. DINOv3's SwiGLU feed-forward block is not matched by the projection names used by the other families, so its adapters cover attention only.

Backbone	Blocks	Adapted modules	Sites	Trainable	% of model
MoCo v3 ViT-B	12	attn + MLP	48	2.40M	2.79
MAE ViT-L/16	24	attn + MLP	96	6.34M	2.09
data2vec ViT-L/16	24	attn + MLP	96	6.34M	2.05
I-JEPA ViT-H/14	32	attn + MLP	128	10.55M	1.67
DINOv3 ViT-H+/16	32	attn only	96	3.99M	0.48

Image preprocessing. We applied only the preprocessing transformations prescribed by each backbone's original implementation, without introducing any additional image preprocessing.

D.2 Hyperparameters

First, the scaling factor is tied to the rank as $\alpha = 2r$ throughout. Holding α/r constant means that a rank ablation varies adapter capacity without simultaneously varying the effective step size of the low-rank update, so that the two are not confounded. Second, each regime is trained at its own optimal learning rate, selected by sweep: 10^{-3} for LoRA and linear probing, 10^{-4} for full fine-tuning. We report the sweep rather than fixing one rate across regimes because the ranking between them inverts at matched rates at 10^{-4} full fine-tuning leads, at 10^{-3} LoRA leads by 14 points. A comparison between adaptation regimes is therefore only defensible at respective optima, and must say so.

Our LoRA comparison characterizes two regimes at their separately selected learning rates and parameter-update scopes. It therefore measures practical regime-level behavior rather than isolating the effect of low-rank parameterization under identical optimization and parameter coverage.

Optimisation. All regimes are trained for 20 epochs with a batch size of 32, using AdamW under a one-cycle schedule with a warm-up fraction of 0.1, a weight decay of 0.05, and cross-entropy with label smoothing 0.1, gradient-norm clipping at at 1.0. The backbone runs under bfloat16 autocast while the classifier head is evaluated in float32. A class-stratified 10% of each training split is held out for validation; the checkpoint with the best validation accuracy is retained and is the one scored on the official test split, which is read exactly once and never enters a selection decision. Unless a rank, learning rate, or seed is explicitly ablated, every run uses $r = 16$, the rate given above, and seed 42, with seeds 43 and 44 providing replicates where reported.

Classifier head. All three regimes share the same head: a layer normalization over the pooled backbone output, followed by a single linear layer mapping it to the class logits, with weights initialized from a truncated normal of standard deviation 0.01 and a zero bias. Both the layer normalization and the linear layer are trainable in every regime, so that the regimes differ only in how much of the backbone is allowed to move; under full fine-tuning the trainable count therefore slightly exceeds the backbone's own parameter count. The embeddings analyzed throughout this paper are the pooled backbone output, taken before this layer normalization and before the linear layer.

E Additional Results

E.1 Adapted-to-adapted analysis

Tab. 7 shows that cross-target similarity remains strongly family-dependent. MoCoV3 variants are the most consistent across adaptation targets, achieving the lowest mean GW (< 0.01) and highest mean CKA (0.72 ± 0.05). Conversely, data2vec2 exhibits the lowest cross-target alignment, while I-JEPA and MAE occupy intermediate positions. Adaptation-target effects are less uniform across families. Flowers-102 and OxfordPets yield the strongest alignment for data2vec2 and MoCoV3, whereas DTD and Flowers-102 are most aligned for I-JEPA and MAE. Nevertheless, OxfordPets paired with DTD consistently produces the lowest CKA within each family. Together with the parent-child results in Tab. 3, these findings indicate that MoCoV3 preserves both its pretrained structure and its cross-target consistency, while the effect of the adaptation target depends on the pretrained family.

Table 7: Cross-target comparison of post-training representations across SSL families. For each family, GW and CKA measure the similarity between variants adapted to DTD, Flowers-102, and OxfordPets. Each target-pair value is averaged over five random seeds. The Mean GW and Mean CKA columns report the mean ± sample standard deviation across the three target pairs. Lower GW indicates stronger preservation of relational geometry, while higher CKA indicates stronger representational alignment.

Family	Target pair	GW	Mean GW	CKA	Mean CKA
data2vec2	DTD vs Flowers-102	0.0228	0.02 ± 0.00	0.3134	0.29 ± 0.06
	DTD vs OxfordPets	**0.0147**		0.2295	
	Flowers-102 vs OxfordPets	0.0160		**0.3401**	
I-JEPA	DTD vs Flowers-102	0.0233	0.02 ± 0.00	**0.4836**	0.42 ± 0.09
	DTD vs OxfordPets	0.0149		0.3186	
	Flowers-102 vs OxfordPets	**0.0126**		0.4652	
MAE	DTD vs Flowers-102	0.0141	0.01 ± 0.00	**0.5386**	0.47 ± 0.11
	DTD vs OxfordPets	**0.0125**		0.3461	
	Flowers-102 vs OxfordPets	0.0145		0.5248	
MocoV3	DTD vs Flowers-102	0.0121	**$< 0.01 \pm 0.00$**	0.7383	**0.72 ± 0.05**
	DTD vs OxfordPets	0.0087		0.6666	
	Flowers-102 vs OxfordPets	**0.0070**		**0.7555**	

E.2 CKA extended analysis

Table 8: **Layer-wise CKA across training seeds.** Each configuration was retrained with seeds 42, 43 and 44 on all three adaptation targets and compared to its pretrained parent, for 27 parent–child comparisons in total. The three seed columns give the mean matched-depth CKA of each seed, averaged over the three targets, and s.d. their standard deviation. Max block s.d. is the largest per-block standard deviation across seeds, to be read against profile range, the peak-to-trough amplitude of the mean profile: seed variation accounts for three to nine percent of the depth-wise structure it is compared to.

		Mean matched-depth CKA					
Backbone	**Regime**	**42**	**43**	**44**	**s.d.**	**Max block s.d.**	**Profile range**
MoCo v3 ViT-B	LoRA	0.9114	0.9112	0.9124	0.0006	0.0078	0.268
	Full FT	0.7979	0.7947	0.8024	0.0039	0.0122	0.220
MAE ViT-L/16	LoRA	0.9000	0.9098	0.8963	0.0069	0.0427	0.455

E.3 KNN EXTENDED ANALYSIS

Throughout this section, for reference, the expected neighbors-kept score under a uniformly random neighborhood is $k/(N_e - 1)$ and depends on the size of the evaluation dataset.

Table 9: Neighborhood preservation by pretrained family ($k = 20$): neighbors kept $\overline{K}_{a,e}$, seed consistency $S_{a,e}$, in %, and class purity $\overline{P}_{f,e}$ (pretrained → post-trained), on-target and elsewhere. * single training seed, so no seed baseline is available.

		$\overline{K}_{a,e}$ (%)		$S_{a,e}$ (%)	$\overline{P}_{f,e}$	
Model	Post-tr.	On-target	Elsewhere	On-target / Elsewhere	On-target	Elsewhere
DINOv3 ViT-H+/16	LoRA*	62	80	–	0.83 → 0.90	0.83 → 0.82
MoCo v3 ViT-B/16	LoRA	49	67	72 / 71	0.62 → 0.83	0.62 → 0.61
MoCo v3 ViT-B/16	Full FT	46	56	68 / 64	0.62 → 0.82	0.62 → 0.58
I-JEPA ViT-H/14	LoRA*	31	45	–	0.48 → 0.84	0.48 → 0.52
MAE ViT-L/16	LoRA	23	33	63 / 50	0.33 → 0.85	0.33 → 0.47
data2vec 2.0 ViT-L/16	LoRA*	12	16	–	0.18 → 0.85	0.18 → 0.43

Tab. 9 reports the values plotted in Fig. 3a, together with class purity $\overline{P}_{f,e}$. Pretrained class purity follows the same ordering across families as neighbors-kept, from DINOv3 (0.83) to data2vec2 (0.18), consistent with class-coherent neighborhoods having less to reorganize. After adaptation, on-target purity reaches 0.82 to 0.90 in every family, regardless of this starting point. Elsewhere, LoRA leaves purity nearly unchanged for DINOv3 and MocoV3 and raises it for the three other families. On MocoV3, both regimes score more than 20 points below seed consistency on-target, but only four points below it elsewhere for LoRA, against eight for full fine-tuning. Full fine-tuning also produces the largest purity drop elsewhere ($0.62 \rightarrow 0.58$) and is less reproducible across seeds (seed consistency of $68/64$ vs $72/71\%$). For MAE, the gap elsewhere reaches 17 points: how much change spreads beyond the adaptation target is itself family-dependent. We omit standard errors from this table: since each score averages over thousands of query images, they remain below 0.5 points for $\overline{K}_{a,e}$ and 0.003 for $\overline{P}_{f,e}$ (Tab. 10 and 11), which is finer than the precision reported here, and would reduce readability without affecting any comparison.

Table 10: Effect of the LoRA rank on neighborhood preservation (MoCo v3 ViT-B/16, single seed, $k = 20$, mean ± 1 SEM over queries throughout): neighbors kept $\overline{K}_{a,e}$, in %, and class purity $\overline{P}_{f,e}$.

	$\overline{K}_{a,e}$ (%)		$\overline{P}_{f,e}$	
Post-training	On-target	Elsewhere	On-target	Elsewhere
LoRA, $r = 1$	57.7±0.2	74.8±0.1	0.769±0.003	0.632±0.002
LoRA, $r = 4$	53.5±0.2	71.5±0.1	0.798±0.003	0.624±0.002
LoRA, $r = 16$ (default)	48.7±0.1	67.2±0.1	0.834±0.002	0.614±0.001
LoRA, $r = 64$	44.7±0.2	59.4±0.1	0.842±0.003	0.570±0.002
LoRA, $r = 256$	39.4±0.2	32.4±0.1	0.813±0.003	0.369±0.002
Full fine-tuning	45.6±0.1	55.9±0.1	0.816±0.002	0.579±0.001

Tab. 10 reports the values plotted in Fig. 3b. On-target, neighbors-kept declines by four to five points per rank step, and LoRA falls below full fine-tuning from r=64 (44.7 vs 45.6%). Elsewhere, the decline is gradual up to r=64 and then steep ($59.4 \rightarrow 32.4\%$), making r=256 markedly more disruptive than full fine-tuning in both neighbors-kept (32.4 vs 55.9%) and purity (0.369 vs 0.579). This rank is also the only one at which the elsewhere score falls below the on-target one, and the first at which on-target purity decreases ($0.842 \rightarrow 0.813$): at this capacity, change is no longer concentrated on the adaptation target. All ranks use a single seed, so the reported SEMs reflect query sampling, not seed variability.

Tab. 11 reports neighbors-kept for each adaptation dataset, together with the overlap between differently adapted models. Beyond the trends discussed in Sec. 5.3, three observations stand out. First, the advantage of Flowers-102 holds elsewhere, where it preserves the most neighbors in all five families (by 1 to 19 points), but not on-target, where it leads in three families, ties with DTD for data2vec2 and ranks last for I-JEPA. Second, in data2vec2, the DTD adapter keeps fewer neighbors

Table 11: Effect of the adaptation dataset on neighborhood preservation (LoRA, default rank, single seed, $k = 20$): neighbors kept $\overline{K}_{a,e}$ (%, mean $\pm$ 1 SEM over queries) on the adaptation dataset (Tgt.) and the two others (Els.), for each of the three adaptation datasets, alongside the overlap between two differently-adapted models of the same family (Adapted↔adapted) and the seed consistency $S_{a,e}$ for reference. Bold (left): the adaptation dataset that preserves the most neighbors in each family. Bold (right): the most distant pair of adapters. Mean: equal-weight average over the five families (†: over the two families with a seed replicate, MoCo v3 and MAE).

	Adapted on DTD		Adapted on Flowers-102		Adapted on Oxford Pets		Adapted ↔ adapted			
Family	Tgt.	Els.	Tgt.	Els.	Tgt.	Els.	DTD/Fl.	DTD/Ox.	Fl./Ox.	Seed
DINOv3	60.7±0.5	67.5±0.2	**64.0±0.2**	**87.4±0.1**	62.5±0.2	84.0±0.1	62.8±0.2	**58.6±0.2**	71.7±0.1	–
MoCo v3	47.1±0.5	62.3±0.2	**53.8±0.3**	**75.0±0.2**	45.9±0.3	64.2±0.2	55.4±0.2	**46.2±0.2**	56.6±0.2	71.0±0.1
I-JEPA	30.2±0.5	38.4±0.2	27.9±0.3	**58.6±0.2**	**35.5±0.3**	39.3±0.2	40.5±0.2	**32.6±0.2**	42.0±0.2	–
MAE	21.3±0.4	27.8±0.2	**26.6±0.2**	**37.0±0.2**	19.5±0.2	35.7±0.2	36.0±0.2	**28.4±0.2**	39.0±0.2	54.0±0.1
data2vec 2.0	**15.2±0.3**	11.4±0.1	**15.2±0.2**	**18.8±0.2**	4.8±0.1	16.7±0.2	28.2±0.2	**24.6±0.2**	30.4±0.2	–
Mean	34.9±0.2	41.5±0.1	37.5±0.1	55.4±0.1	33.6±0.1	48.0±0.1	44.6±0.1	38.1±0.1	47.9±0.1	62.5±0.1†

Table 12: Sensitivity of neighborhood preservation to the neighborhood size k: neighbors kept $\overline{K}_{a,e}$ and seed consistency $S_{a,e}$, in %, on-target / elsewhere. The main text uses $k = 20$. * single training seed, so no seed baseline is available.

Model	Post-tr.	$k = 10$	$k = 20$	$k = 50$	$k = 100$
Neighbors kept $\overline{K}_{a,e}$ (%)					
DINOv3 ViT-H+/16	LoRA*	55 / 75	62 / 80	71 / 86	74 / 87
MoCo v3 ViT-B/16	LoRA	46 / 65	49 / 67	51 / 69	50 / 69
MoCo v3 ViT-B/16	Full FT	42 / 53	46 / 56	49 / 58	51 / 59
I-JEPA ViT-H/14	LoRA*	28 / 42	31 / 45	35 / 49	36 / 51
MAE ViT-L/16	LoRA	22 / 32	23 / 33	23 / 35	23 / 37
data2vec 2.0 ViT-L/16	LoRA*	11 / 14	12 / 16	13 / 18	13 / 21
Seed consistency $S_{a,e}$ (%)					
MoCo v3 ViT-B/16	LoRA	66 / 68	72 / 71	76 / 73	76 / 74
MoCo v3 ViT-B/16	Full FT	62 / 61	68 / 64	76 / 67	77 / 68
MAE ViT-L/16	LoRA	55 / 47	63 / 50	70 / 52	70 / 54

elsewhere than on-target (11.4 vs 15.2%), and the OxfordPets adapter keeps only 4.8% of on-target neighbors. Finally, in the two families with seed replicates, adapters trained on different targets share fewer neighbors than two seeds of the same adapter (46–57 vs 71% for MocoV3, 28–39 vs 54% for MAE), consistent with each target moving the representation in its own direction rather than along a shared one.

Tab. 12 repeats the per-family analysis of Tab. 9 for $k \in \{10, 20, 50, 100\}$. Absolute values vary with k, by up to 19 points (DINOv3, on-target), but the major qualitative trends remain stable across neighborhood sizes. In particular, the family ordering is unchanged on-target and elsewhere, and the elsewhere score exceeds the on-target one in every configuration. On MocoV3, LoRA also preserves more neighbors than full fine-tuning under elsewhere evaluation for all tested values of k. Quantitative differences between LoRA and full fine-tuning can nevertheless vary with k; for example, under on-target evaluation on MocoV3, the neighbors-kept ordering reverses at $k = 100$. On-target scores remain 20 to 47 points below seed consistency, against 3 to 5 points elsewhere for MocoV3 with LoRA.

E.4 LoRA EXTENDED ANALYSIS

Table 13: **LoRA rank sweep across three backbones**, with $\alpha{=}2r$ throughout. Top-1 and the CKA columns average the three adaptation targets under on-target evaluation; $\overline{K}$ averages the nine target $\times$ evaluation-domain pairs. Dispersion is the spread across those cells, which reflects how much the three targets differ in difficulty rather than measurement uncertainty: seed noise, measured over three seeds at $r{=}16$, is 0.31 to 0.50 accuracy points. Rows are shown only where at least two targets converged. † Oxford-IIIT Pet collapsed at this rank and is excluded from the row. MAE at $r{=}256$ and I-JEPA at $r{=}64$ and $r{=}256$ are absent for the same reason: at the learning rate selected at $r{=}16$ they diverge, seems to be an optimisation effect rather than a property of the rank.

Backbone	**Config.**	**Train. (M)**	**Top-1**	**CKA trunk**	**CKA final**	$\overline{K}_{a,e}$
MAE ViT-L/16 (303M)	$r{=}1$	0.46	87.9	0.929 ± 0.063	0.689 ± 0.253	0.358 ± 0.083
	$r{=}4$	1.64	89.2	0.922 ± 0.047	0.656 ± 0.205	0.338 ± 0.083
	$r{=}16$	6.36	89.2	0.900 ± 0.057	0.536 ± 0.296	0.298 ± 0.064
	$r{=}64^{\dagger}$	25.24	85.1	0.883 ± 0.080	0.586 ± 0.311	0.234 ± 0.048
	Full FT	303.37	87.4	0.834 ± 0.067	0.424 ± 0.176	0.253 ± 0.043
I-JEPA ViT-H/14 (631M)	$r{=}1$	0.74	87.5	0.907 ± 0.021	0.482 ± 0.035	0.472 ± 0.145
	$r{=}4$	2.70	87.4	0.889 ± 0.017	0.353 ± 0.091	0.451 ± 0.139
	$r{=}16$	10.57	87.3	0.858 ± 0.021	0.266 ± 0.139	0.378 ± 0.117