PAFT: AUXILIARY BOTTLENECKS MAKE REPRESENTATIONS MORE PROJECTABLE

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual analysis of learned representations plays an important role in interpreting deep neural networks. Current practice predominantly focuses on applying post hoc dimensionality-reduction techniques, such as PCA, t-SNE, and UMAP. However, the strength and prevalence of these tools largely frame dimensionality reduction for visualization as a downstream stage, leaving the geometry learned during training unaddressed. In this work, we propose PAFT (Projection-Aware Fine-Tuning), a fine-tuning method built around an auxiliary branch consisting of a linear bottleneck and a classification head attached to the penultimate layer of a classifier. The classifier and the auxiliary branch are jointly trained with three additional terms in the training objective: prediction consistency between the main and auxiliary heads, together with CKA alignment and neighborhood preservation between the penultimate and bottleneck representations. During training, PAFT learns a linear projection from the penultimate representation to a low-dimensional bottleneck that preserves predictive and geometric structure, while simultaneously reshaping the high-dimensional representation itself so that its structure is more faithfully preserved in lower dimensions. At $d = 2$, this bottleneck also provides a directly inspectable two-dimensional view without requiring a separate post-hoc projection. Across five image-classification datasets and three classifier backbones, we evaluate two-dimensional projections using 20 established projection-quality measures. PAFT gives the strongest results, outperforming both the untreated baseline and rank-reduction alternatives overall, while maintaining comparable classification accuracy.

1 INTRODUCTION

The representations learned by deep neural networks live in high-dimensional spaces that are difficult to inspect directly. To make them interpretable, dimensionality-reduction techniques such as Principal Component Analysis (PCA) (Pearson, 1901), Multidimensional Scaling (MDS) (Kruskal, 1964), t-SNE (van der Maaten & Hinton, 2008) and UMAP (McInnes et al., 2018) are routinely applied, and the resulting two-dimensional projections are used to visually inspect their structure.

Despite their widespread use, these approaches are inherently post hoc: nothing in the model's objective encourages its representation to survive the projection that will later be used for analysis. How faithful such a projection is can itself be measured. Projection-quality criteria compare points before and after projection to check whether neighbors, relative distances and cluster structure are preserved. The literature developing these measures is mature (Venna & Kaski, 2006; Lee & Verleysen, 2009; Espadoto et al., 2021; Jeon et al., 2022). We adopt them directly to evaluate how well high-dimensional structure is retained in low-dimensional projections.

To address this, we propose PAFT (Projection-Aware Fine-Tuning), which augments classifier fine-tuning with a low-dimensional auxiliary bottleneck and a second classification head attached to the penultimate representation (Figure 1). The classifier and auxiliary branch are jointly trained with three additional objectives: prediction consistency between the two heads, CKA alignment (Kornblith et al., 2019) between the penultimate and bottleneck representations, and neighborhood preservation between those representations. During training, PAFT learns a bottleneck that preserves predictive and geometric structure while adapting the penultimate representation itself so that its structure is more faithfully retained in lower dimensions.

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.

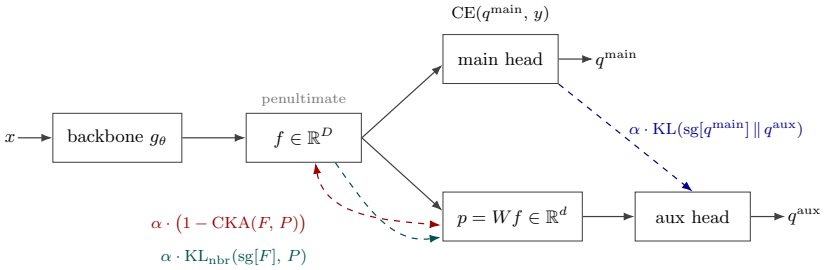


Figure 1: The objective. A linear bottleneck $p = Wf$ and a second classifier are attached to the penultimate representation f; prediction consistency aligns the bottleneck's predictive distribution with the main head, CKA alignment preserves broader similarity structure between f and p, and neighborhood preservation matches their neighborhood distributions.

Across five image-classification datasets and three backbones, PAFT achieves the strongest results among the methods compared, improving, on average, 13.6 of the 20 projection-quality measures per projection over the baseline while degrading only 1.0. These gains span every measure family we report in Section 4 and are concentrated in the local geometry and global geometry families.

In summary, our main contributions include:

- We introduce PAFT, a fine-tuning approach that jointly learns a low-dimensional linear bottleneck and reshapes the penultimate representation, and show that this improves the fidelity of post-hoc PCA, t-SNE, UMAP, and PaCMAP views.
- We benchmark PAFT across five datasets and three classifier backbones under four standard post-hoc projection methods, PCA, t-SNE, UMAP, and PaCMAP, using 20 established projection-quality measures spanning local and global geometry, label structure, and cluster-level structure. We compare PAFT with the untreated fine-tuning baseline and three rank-reduction alternatives: a nuclear-norm penalty, a top-d variance penalty, and a narrow-penultimate baseline. Across these evaluations, PAFT gives the strongest overall projection-quality performance while maintaining comparable classification accuracy.

2 Related Work

Measuring what a projection preserves. Projection quality has been studied using a range of criteria that compare structure in the original and projected spaces. These include neighborhood-based measures such as trustworthiness and continuity (Venna & Kaski, 2006), the co-ranking framework (Lee & Verleysen, 2009) and derived rank-based summaries (Lee et al., 2013), the Local Continuity Meta-Criterion (LCMC) (Chen & Buja, 2009), and large comparative studies evaluating projection techniques across collections of such measures (Espadoto et al., 2021). Complementary criteria assess cluster- and label-level structure: steadiness and cohesiveness measure how well inter-cluster structure is preserved between the original and projected spaces (Jeon et al., 2022), while label-trustworthiness and label-continuity evaluate how class structure changes between the two spaces without assuming that each class forms a single cluster in the original representation (Jeon et al., 2024). We adopt these established measures directly as our primary evaluation framework; Section 4 provides the full metric grouping and evaluation setup.

The reliability of neighbor embeddings. t-SNE (van der Maaten & Hinton, 2008) and UMAP (McInnes et al., 2018) are widely used nonlinear tools for two-dimensional visualization, including in single-cell analysis (Becht et al., 2019). PaCMAP (Wang et al., 2021) was designed to balance local and global structure preservation. These nonlinear embeddings can be sensitive to design and optimization choices: t-SNE cluster sizes and inter-cluster distances need not reflect those in the original space (Wattenberg et al., 2016), initialization can strongly affect global structure in both t-SNE and UMAP (Kobak & Linderman, 2021), and the attraction-repulsion balance changes whether continuous or cluster structure is emphasized (Böhm et al., 2022). In single-cell settings,

Chari & Pachter (2023) further show that reduction to two dimensions can introduce substantial distortions and apparent false structure. Rather than modifying these post-hoc algorithms, we train representations to be faithfully reducible. PAFT improves projection quality across PCA, t-SNE, UMAP, and PaCMAP, with the largest improvement observed under PCA.

Learning the projection. A parallel line of research learns the projection function itself, as in parametric t-SNE (van der Maaten, 2009) and parametric UMAP (Sainburg et al., 2021). Parametric t-SNE and parametric UMAP learn mappings to preserve fixed neighborhood structures constructed from an input representation. PAFT instead jointly learns the classifier representation and its auxiliary linear projection without prescribing a fixed neighborhood graph. Accordingly, our empirical comparisons focus on alternatives that act directly on the classifier representation during training, including the rank-reduction alternatives discussed below, rather than on methods that fit a parametric embedding to a preconstructed neighborhood target. Appendix H examines the contribution of joint training by comparing PAFT with an auxiliary branch fitted to a fixed baseline representation.

Rank reduction as an alternative. Rank reduction provides an alternative way to encourage more compact representations. Shi et al. (2024) penalize the nuclear norm of penultimate features to encourage low-rank representations for domain generalization. Hu et al. (2013) proposed truncated nuclear-norm regularization for matrix completion, leaving the leading singular values unpenalized while penalizing the remainder to better approximate low rank. We adapt these ideas to our setting through a scale-normalized nuclear-norm penalty and a top-d variance penalty whose target dimension d is selected from the same $\{16, 64\}$ grid as PAFT's bottleneck width. We also include a narrow-penultimate baseline that directly replaces the original penultimate representation with a lower-dimensional one and trains the classifier on that representation without an auxiliary branch. Appendix B defines all three alternatives. These alternatives test whether encouraging lower rank or directly restricting representation dimensionality is sufficient to account for PAFT's projection-quality gains.

Designing for inspectability. A broader principle in interpretable machine learning is to build inspectability into the model rather than relying exclusively on post hoc explanations (Rudin, 2019). Concept bottleneck models exemplify this idea by routing predictions through intermediate variables aligned with human-specified concepts, making those variables directly interpretable and amenable to intervention (Koh et al., 2020). PAFT follows the same broad design principle, but its bottleneck is trained to preserve predictive and geometric structure rather than to encode human-specified concepts. At $d = 2$, this bottleneck is directly visualizable during both training and inference, without assigning semantic meaning to its individual coordinates.

3 PAFT: Projection-Aware Fine-Tuning

Problem setting. We are given a standard classification setup: a backbone that produces a penultimate representation $f \in \mathbb{R}^D$ and a classification head trained via cross-entropy loss. Our goal is to jointly learn a low-dimensional linear projection that preserves predictive and geometric structure while adapting the representation on which it acts.

3.1 Objective

Let g_θ be a backbone producing a penultimate representation $f = g_\theta(x) \in \mathbb{R}^D$ and let h_{main} be the standard classification head on top of f. We introduce two linear mappings: an auxiliary bottleneck $p = Wf \in \mathbb{R}^d$ with $d \ll D$, and a second classification head h_{aux} acting on p. Figure 1 summarizes the resulting architecture and training objective. A single coefficient α weights the three auxiliary objectives, keeping their weighting to one tuning dimension rather than introducing term-specific coefficients. The shared coefficient is a tuning-design choice rather than an assumption that the three auxiliary losses take equal raw values or produce equal gradient magnitudes: their relative scaling is held fixed while α controls the overall strength of the auxiliary objective. Writing f_i and p_i for the representations of the i-th example in a minibatch of size B, and defining its class probability distributions as $q_i^{\text{main}} = \text{softmax}(h_{\text{main}}(f_i))$ and $q_i^{\text{aux}} = \text{softmax}(h_{\text{aux}}(p_i))$, the overall training

loss for that minibatch is

$$\mathcal{L} = \underbrace{\frac{1}{B}\sum_{i=1}^{B} \mathrm{CE}\big(h_{\mathrm{main}}(f_i),\, y_i\big)}_{\text{task objective}} + \alpha \Bigg[\underbrace{\frac{1}{B}\sum_{i=1}^{B} \mathrm{KL}\big(\,\mathrm{sg}[q_i^{\mathrm{main}}] \,\big\|\, q_i^{\mathrm{aux}}\big)}_{\text{prediction consistency}} + \underbrace{\big(1 - \mathrm{CKA}(F, P)\big)}_{\text{CKA alignment}} + \underbrace{\mathrm{KL}_{\mathrm{nbr}}\big(\mathrm{sg}[F],\, P\big)}_{\text{neighborhood preservation}} \Bigg], \tag{1}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator, $F \in \mathbb{R}^{B\times D}$ and $P \in \mathbb{R}^{B\times d}$ collect the minibatch representations, and CKA is linear centered kernel alignment (Kornblith et al., 2019),

$$\mathrm{CKA}(F, P) = \frac{\big\|P_c^\top F_c\big\|_{\mathrm{F}}^2}{\big\|F_c^\top F_c\big\|_{\mathrm{F}}\,\big\|P_c^\top P_c\big\|_{\mathrm{F}}}, \tag{2}$$

with column-centered matrices F_c and P_c.

Neighborhood preservation uses the parametric t-SNE objective (van der Maaten, 2009), applied between the two representations instead of between a fixed input space and a learned map. Let Π be the symmetrized Gaussian neighborhood distribution over pairs in $\mathrm{sg}[F]$, with a per-point bandwidth set by the usual perplexity search, and let Q be the Student-t distribution over pairs in P, with $Q_{ij} \propto \big(1 + \|p_i - p_j\|^2\big)^{-1}$. Then

$$\mathrm{KL}_{\mathrm{nbr}}\big(\mathrm{sg}[F],\, P\big) = \sum_{i\neq j} \Pi_{ij}\, \log \frac{\Pi_{ij}}{Q_{ij}}. \tag{3}$$

Prediction consistency uses knowledge-distillation-style distribution matching (Hinton et al., 2015), with stop-gradient applied to the main-head target. Stop-gradient is also applied to F when constructing the neighborhood-preservation target; CKA alignment remains differentiable in both F and P. This makes the main-head predictions and the neighborhood distribution one-sided targets within their respective auxiliary terms, rather than allowing gradients to propagate through both sides of those objectives. Keeping CKA differentiable in both F and P allows not only the projection W to adapt to the representation, but also the representation to adapt so that its similarity structure is better preserved by the low-dimensional linear projection.

4 Experiments

4.1 Experimental setup

Models and data. Three pretrained backbones are fine-tuned: ResNet-18 and ResNet-50 (He et al., 2016) and the AugReg ViT-S/16 (Dosovitskiy et al., 2021; Steiner et al., 2022) (`vit_small_patch16_224.augreg_in1k`), all from `timm` (Wightman, 2019) and all pretrained on ImageNet-1k, on five image-classification datasets: STL-10 (Coates et al., 2011), SVHN (Netzer et al., 2011), CIFAR-100 (Krizhevsky, 2009), TinyImageNet (Le & Yang, 2015) and CUB-200-2011 (Wah et al., 2011), covering all 15 combinations of three backbones and five datasets. Fine-tuning hyperparameters are held constant across all configurations; they are listed in Appendix A.

Treatments. For each combination, we train the untreated baseline, the nuclear-norm penalty, the top-d variance penalty, the narrow-penultimate baseline, and PAFT, all with 10 shared seeds. PAFT and the two penalties each search six coefficients; PAFT and the top-d penalty additionally search $d \in \{16, 64\}$, denoting bottleneck width and target dimension, respectively, and the narrow-penultimate baseline searches its penultimate width. The primary cross-method comparisons use validation-selected settings from these grids, while additional PAFT bottleneck widths are examined separately in explicitly identified analyses. Appendix A details the search grids and validation-based candidate selection, while Appendix B defines the comparison methods.

Projection evaluation. Each projection method is fitted on the validation split with library-default parameters, and test points are then embedded into this fixed fit: PCA applies its linear map, while t-SNE, UMAP and PaCMAP use their out-of-sample transforms (Appendix C). For the main comparisons, all four projection methods are applied to each model's penultimate representation. Projection quality is evaluated with 20 measures from four families.

- **local geometry** (7): trustworthiness and continuity (Venna & Kaski, 2006), LCMC (Chen & Buja, 2009), both mean relative rank errors (Lee & Verleysen, 2009), AUC $R_{\mathrm{NX}}(K)$ (Lee et al., 2013; 2015), and Q_{local};
- **global geometry** (5): Q_{global}, normalized stress, Shepard goodness, Pearson distance correlation, and random-triplet accuracy as defined by Wang et al. (2021);
- **label structure** (4): silhouette, distance consistency, k-NN accuracy, neighborhood hit;
- **cluster level** (4): steadiness and cohesiveness (Jeon et al., 2022), and label-trustworthiness and label-continuity (Jeon et al., 2024).

Taxonomy. Our metric grouping follows the local / cluster-level / global distinction of Jeon et al. (2023), with an additional label-structure family that separates label-based measures from geometric measures. This separation reflects that class separation and representational geometry are distinct properties that should not be conflated (Jeon et al., 2024). The label-structure measures are computed from the projection and the class labels alone; the other sixteen measures compare the projection with the high-dimensional reference. Label-trustworthiness and label-continuity also use labels, but compare class separability between the two spaces, and therefore remain in the cluster-level family. Neighborhood sizes and the remaining measure parameters are given in Appendix C.

Significance criteria. We evaluate each method's two-dimensional projections using the measures above. When a measure requires a high-dimensional reference, we use that method's own penultimate representation. When comparing two methods, each projection–measure pair is compared across the 10 shared seeds with a paired t-test, with differences oriented so that positive values favor the first method, including for measures where lower is better. The 80 tests of one backbone–dataset combination and comparison pair form a single Benjamini–Hochberg family at $q = 0.05$. A measure counts as a win for the first method when its difference is significant and reaches a 1% relative-change threshold, as a loss in the symmetric case, and as unchanged otherwise. Separately, under the classification-accuracy criterion, a candidate is rejected when its validation accuracy is significantly below the baseline or drops by at least 1% relative to the baseline. Appendix D gives the full procedures.

Win–loss difference. For each direct comparison, we subtract the number of measures favoring the second method from the number favoring the first, counting only differences that satisfy the criteria above. For example, 14 wins, two losses, and four unchanged measures yield a win–loss difference of $+12$ out of 20. This is a score for the stated comparison pair, not an absolute score assigned to either method; scores obtained against different comparators therefore cannot be subtracted to recover a direct comparison between those comparators.

4.2 Main results

Comparison with the baseline. Across 15 backbone–dataset combinations and four projection methods, PAFT achieves an average win–loss difference of $+12.6$ out of 20 measures against the untreated baseline.

Table 1 breaks this result down by projection and measure family. Local and global geometry carry the largest win–loss differences under every projection, 5.3–6.3 out of seven and 4.0–5.0 out of five measures. Label and cluster-level gains are concentrated under PCA, reaching 3.1 and 1.7 out of four measures, compared with at most 1.1 and 0.7 under t-SNE, UMAP, and PaCMAP.

Comparison with related techniques. We compare PAFT with the nuclear-norm penalty, the top-d variance penalty, and the narrow-penultimate baseline, each at its validation-selected operating point. Table 2 reports all 20 projection-quality measures under PCA for ResNet-50/CIFAR-100. The three alternatives improve many of these measures over the untreated baseline, while PAFT

Table 1: Win–loss difference for PAFT versus the untreated baseline, averaged over 15 backbone–dataset combinations. Entries count wins minus losses within each measure family; the total covers all 20 measures. The final row averages the four projection methods. Values are rounded independently.

Projection	Local (7)	Global (5)	Label (4)	Cluster (4)	Total (20)
PCA	+6.3	+4.6	+3.1	+1.7	+15.8
t-SNE	+5.3	+5.0	+0.8	+0.2	+11.3
UMAP	+5.4	+4.0	+0.9	+0.3	+10.6
PaCMAP	+5.9	+4.9	+1.1	+0.7	+12.6
Mean	+5.7	+4.6	+1.5	+0.8	+12.6

Table 2: Raw PCA projection-quality measures on ResNet-50 with CIFAR-100. Entries are means $\pm$ sample standard deviations over 10 seeds. Columns are the untreated baseline, the narrow-penultimate baseline, the nuclear-norm penalty, the top-d variance penalty and PAFT, each at its validation-selected operating point. PCA is applied to each model's penultimate representation, except in the $d = 2$ PAFT column, a separate run whose bottleneck is read directly. Best values per row are bold; ties at the reported precision are left unmarked; $\downarrow$ indicates lower is better.

Metric	Untreated	Narrow	Nuclear	Top-d	**PAFT** d=16	**PAFT** d=2
Local geometry						
Trustworthiness	0.744 ± 0.021	0.793 ± 0.015	0.770 ± 0.024	0.800 ± 0.022	0.849 ± 0.020	**0.884** ± 0.015
Continuity	0.930 ± 0.010	0.956 ± 0.003	0.934 ± 0.010	0.934 ± 0.014	0.953 ± 0.003	**0.970** ± 0.003
$\mathrm{AUC}_{R_{NX}}$	0.140 ± 0.012	0.178 ± 0.009	0.160 ± 0.009	0.183 ± 0.013	0.209 ± 0.015	**0.248** ± 0.019
Q_{local}	0.049 ± 0.005	0.065 ± 0.004	0.054 ± 0.004	0.068 ± 0.007	0.069 ± 0.004	**0.124** ± 0.018
LCMC	0.037 ± 0.004	0.051 ± 0.003	0.041 ± 0.003	0.053 ± 0.006	0.051 ± 0.003	**0.091** ± 0.013
$\mathrm{MRRE}_{h\to l}\downarrow$	0.062 ± 0.009	0.038 ± 0.003	0.058 ± 0.009	0.058 ± 0.013	0.042 ± 0.003	**0.027** ± 0.002
$\mathrm{MRRE}_{l\to h}\downarrow$	0.256 ± 0.021	0.208 ± 0.015	0.230 ± 0.024	0.200 ± 0.022	0.152 ± 0.019	**0.117** ± 0.015
Global geometry						
Dist. correlation	0.564 ± 0.028	0.696 ± 0.017	0.673 ± 0.112	**0.845** ± 0.090	0.746 ± 0.051	0.777 ± 0.033
Shepard goodness	0.543 ± 0.018	0.668 ± 0.020	0.629 ± 0.058	0.722 ± 0.059	0.733 ± 0.057	**0.748** ± 0.031
Norm. stress $\downarrow$	0.448 ± 0.026	0.374 ± 0.010	0.466 ± 0.036	0.492 ± 0.040	0.425 ± 0.014	**0.372** ± 0.007
Triplet accuracy	0.687 ± 0.011	0.747 ± 0.010	0.717 ± 0.012	0.747 ± 0.019	0.779 ± 0.027	**0.789** ± 0.020
Q_{global}	0.338 ± 0.019	0.448 ± 0.018	0.406 ± 0.031	0.483 ± 0.035	0.523 ± 0.051	**0.537** ± 0.039
Label structure						
kNN accuracy	0.158 ± 0.020	0.194 ± 0.009	0.166 ± 0.017	0.185 ± 0.016	0.232 ± 0.018	**0.391** ± 0.050
Neighborhood hit	0.106 ± 0.011	0.122 ± 0.006	0.112 ± 0.009	0.124 ± 0.011	0.152 ± 0.012	**0.273** ± 0.040
Silhouette	-0.241 ± 0.007	-0.230 ± 0.006	-0.247 ± 0.017	-0.259 ± 0.025	-0.228 ± 0.009	**-0.139** ± 0.026
Distance consistency	0.142 ± 0.015	0.196 ± 0.010	0.144 ± 0.022	0.156 ± 0.018	0.197 ± 0.012	**0.361** ± 0.045
Cluster-level						
Steadiness	0.373 ± 0.014	0.364 ± 0.007	0.380 ± 0.010	0.373 ± 0.015	0.384 ± 0.007	**0.459** ± 0.016
Cohesiveness	0.534 ± 0.010	0.512 ± 0.010	0.539 ± 0.020	0.544 ± 0.022	0.559 ± 0.010	**0.633** ± 0.016
Label continuity	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
Label trustworthiness	0.911 ± 0.015	0.936 ± 0.003	0.906 ± 0.022	0.905 ± 0.024	0.940 ± 0.003	**0.962** ± 0.004

improves most of them further. The table also includes a separate $d = 2$ PAFT run whose bottleneck is read directly, without PCA. Appendix E reports the corresponding values for all datasets, averaged over backbones and projection methods.

Table 3 reports PAFT's win–loss difference against the baseline and each alternative by measure family. PAFT keeps a positive win–loss difference against every alternative in all four families. Against every alternative, global geometry shows the largest difference, 3.6–3.8 out of five measures, whereas the alternatives narrow the gap most in local geometry. For label structure, the difference against the nuclear-norm and top-d variance penalties exceeds that against the baseline, while the narrow-penultimate baseline shows a smaller one. In cluster-level measures, all three alternatives narrow the gap slightly relative to the baseline.

Table 3: Win–loss difference of PAFT against the comparator named in each row, from their direct comparison, by measure family. Entries count wins minus losses within each family, averaged over all 15 backbone–dataset combinations and four projections; the total covers all 20 measures. Values are rounded independently.

Comparator	Local (7)	Global (5)	Label (4)	Cluster (4)	Total (20)
Untreated baseline	+5.7	+4.6	+1.5	+0.8	+12.6
Narrow-penultimate baseline	+0.9	+3.8	+1.0	+0.3	+6.0
Nuclear-norm penalty	+3.2	+3.8	+2.1	+0.6	+9.7
Top-d variance penalty	+2.0	+3.6	+2.1	+0.4	+8.2

A directly readable two-dimensional representation. PAFT not only improves the quality of post-hoc projections of the penultimate representation relative to the comparison methods above, but also admits an additional configuration that is not available to the alternatives considered here. When the bottleneck width is set to $d = 2$, the auxiliary bottleneck is itself a two-dimensional representation and can therefore be inspected directly, without fitting a separate post-hoc projection method. Because the bottleneck is a linear map of the penultimate representation, $p = Wf$, and is trained to preserve its predictive and geometric structure, we compare this directly read view with the corresponding linear views obtained by applying PCA to the penultimate representations of the untreated baseline, the narrow-penultimate baseline, the nuclear-norm penalty, the top-d variance penalty, and PAFT. We also include PCA of the penultimate representation from the same $d = 2$ PAFT run. For context, the $d = 2$ direct-read results are also reported alongside the corresponding PCA results in Table 2 and Figure 2, allowing the directly available view to be compared with the post-hoc linear projection used throughout the main evaluation.

Using the standard classification-accuracy gate and fallback procedure, the directly read configuration gives a positive average win–loss difference against every PCA view in Table 4. The differences are $+15.2$ against the untreated baseline, $+12.3$ against the narrow-penultimate baseline, $+10.9$ against the nuclear-norm penalty, $+10.1$ against the top-d variance penalty, $+7.4$ against the validation-selected PAFT representation, and $+5.5$ against PCA of the same $d = 2$ run's penultimate representation. Thus, in addition to making the penultimate representation more amenable to standard post-hoc projection, PAFT can be configured to expose a two-dimensional linear view directly as part of the trained model, without a separate projection-fitting stage; where the $d = 2$ configuration is admissible, this view can therefore also be inspected during training.

Table 4: Projection quality of the accuracy-gated directly read d=2 PAFT view relative to post-hoc PCA views. Entries are average win–loss differences, out of 20 projection-quality measures, over all 15 backbone–dataset combinations; positive values favor the direct-view procedure. The PAFT column uses the validation-selected PAFT configuration for each combination. Where the dedicated d=2 configuration fails the validation accuracy gate, the left view falls back to the untreated baseline under PCA; this occurs in 3 of the 15 combinations. For the same-run d=2 penultimate comparison, both arms use the same fallback. Appendix J reports the gate failures and a pass-only sensitivity analysis.

Comparison view	Untreated	Narrow	Nuclear	Top-d	PAFT	d=2 penult.
Win–loss difference	+15.2	+12.3	+10.9	+10.1	+7.4	+5.5

4.3 Using faithful low-dimensional views

The purpose of projection-quality evaluation is to determine how much of the structure present in a model's high-dimensional representation remains visible after projection. The measures in Section 4.1 quantify neighborhoods, distances, label structure, and cluster structure in the resulting views, including their preservation from the high-dimensional reference where applicable. Higher projection fidelity therefore makes the resulting two-dimensional view a more reliable basis for inspecting the representation.

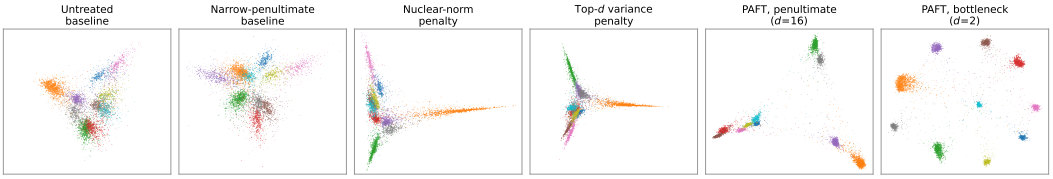


Figure 2: ResNet-18 on SVHN, class-colored, one seed (Appendix A). The first five panels show PCA of each method's penultimate representation at its validation-selected operating point; the final panel shows the directly read bottleneck of the separate PAFT run with $d = 2$.

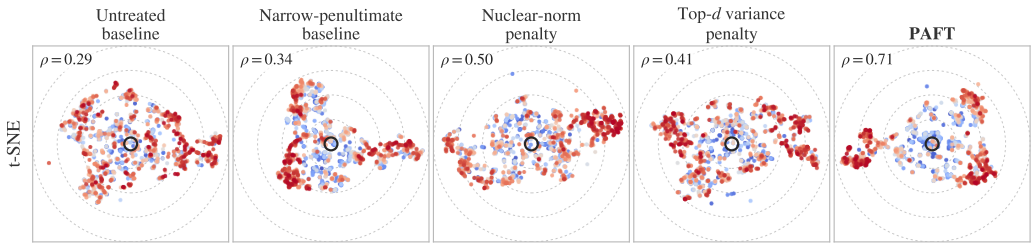


Figure 3: The high-dimensional representation radius on t-SNE maps of ResNet-50/CUB-200, one seed (Appendix A). Each point is colored by its distance from the center of the model's penultimate representations on the test set, expressed as a percentile; each panel reports the Spearman correlation ρ between high- and two-dimensional radius for the displayed samples. The narrow-penultimate panel shows an accuracy-ineligible candidate for illustration only (Appendix A).

When the number of classes is small enough for class coloring to remain legible, class structure can be inspected directly. Figure 2 uses SVHN to illustrate this setting. PAFT shows the clearest class separation among the compared penultimate representations under PCA, consistent with its stronger preservation of label and cluster structure; the directly read $d = 2$ bottleneck separates the classes further without a post-hoc projection.

With many classes, class coloring becomes difficult to interpret. Figure 3 instead visualizes a geometric property of the high-dimensional representation: each test sample is colored by its distance from the center of that model's penultimate representations on the test set. This provides a complementary view of the geometric organization reflected in the global-geometry results, asking whether the central-to-peripheral structure of the high-dimensional representation remains visible after projection. Across the 15 backbone–dataset combinations, four projection methods, and 10 seeds, the mean Spearman correlation between high- and two-dimensional radius is 0.446 for PAFT, compared with 0.110 for the untreated baseline, 0.286 for the narrow-penultimate baseline over its 11 accuracy-eligible combinations, 0.310 for the nuclear-norm penalty, and 0.345 for the top-d variance penalty. PAFT gives the highest correlation under each of the four projection methods.

We further test whether this improvement preserves structure that is useful for interpreting model behavior. For each model, we compare the geometry of class centroids with its class-confusion structure. PAFT preserves the high-dimensional between-class geometry most faithfully under all four projection methods. The high-dimensional class geometries carry a similar relationship with class confusion across the compared methods; after projection, PAFT has the highest mean geometry–confusion correlation and leads under PCA, t-SNE, and PaCMAP. Appendix F reports the full results.

4.4 Ablations and Controls

We use three sets of controls to examine what drives PAFT's gains. First, we ablate the three auxiliary objectives, both individually and in combination. Second, we separate learning the low-dimensional projection from shaping the source representation. Third, we examine whether the

gains can be explained by representation compression alone. Appendices G, H, and I report the detailed results.

Objective terms. The full objective gives the strongest overall result. The two-term variants give average win–loss differences of -1.73, -3.77, and -11.18 relative to full PAFT when prediction consistency, CKA, and neighborhood preservation are removed, respectively; the single-term variants range from -5.58 to -12.30. Despite these differences in projection quality, ablating the auxiliary terms has little effect on the main classifier: the largest mean test-accuracy reduction relative to full PAFT is 0.25 percentage points. Appendix G reports the complete objective-term ablations.

Learning the projection vs. shaping the representation. We compare PCA of the untreated baseline representation with PCA of auxiliary bottlenecks fitted to frozen baseline and PAFT representations. Both fitted branches use the same optimization procedure (Appendix H). The PAFT-representation control achieves an average win–loss difference of $+14.1$ out of 20 PCA measures against the fitted-baseline control.

Rank reduction and compression. The random-projection control separates the general benefit of compression from the advantage observed under structured projections. All three rank-reduction alternatives improve more than PAFT over the untreated baseline under an arbitrary two-dimensional projection, whereas PAFT gives the largest gain under each of PCA, t-SNE, UMAP, and PaCMAP. Together with the spectral analysis in Appendix I, this indicates that compression contributes to projection quality but does not by itself account for PAFT's advantage under the projection methods used in our evaluation.

5 Limitations

Scope. Our experiments cover image classification on five datasets and three backbone architectures. Although the objective is not specific to vision, its effectiveness for other modalities and task types remains to be established.

Role of rank reduction. The spectral and random-projection controls show that compression magnitude alone does not account for PAFT's gains. However, the rank-reduction alternatives also show that reducing representation rank can substantially improve projection quality, and our experiments do not fully isolate how much of PAFT's advantage arises from this effect.

6 Conclusion

We introduced PAFT, a fine-tuning method that jointly learns a low-dimensional bottleneck and adapts the penultimate representation through prediction consistency, CKA alignment, and neighborhood preservation. Across five datasets, three backbones, and four projection methods, PAFT improves projection quality over the untreated baseline and all three rank-reduction alternatives while maintaining predictive accuracy. The analyses in Section 4.3 further show that these gains correspond to better preserved class structure, stronger preservation of radial and between-class geometry, and projected class geometry that remains more informative about class confusion. The directly read $d = 2$ bottleneck also provides a low-dimensional view without post-hoc projection. The controls in Section 4.4 show that the gains depend on joint adaptation of the representation and bottleneck and are not explained by compression alone. Overall, PAFT provides more faithful low-dimensional views by jointly adapting the representation and a structure-preserving auxiliary bottleneck during fine-tuning.

A promising direction for future work is to use the learned bottleneck as the working space for geometric or statistical objectives whose estimates become difficult in high-dimensional representations. Covariance-based objectives are a natural example: when representation dimension is large relative to the number of samples in a minibatch, estimating such quantities can become statistically demanding. Computing them in a lower-dimensional bottleneck that retains task and geometric structure may provide a better-matched space for estimation. Whether this improves the stability and effectiveness of such objectives without sacrificing task performance remains to be investigated.

AI USE STATEMENT

In this work, we used generative AI tools to assist with conceptual development, hypothesis refinement, experimental design and feedback, method implementation, data reformatting, and result interpretation. We did not use generative AI tools to generate synthetic datasets or to assist with translation; mathematical proof tasks and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for software-code assistance, literature search and summarization, and manuscript editing. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The experimental protocol is specified in full in the appendices. Appendix A gives the training hyperparameters, the data splits and seeds, the search grids, and the validation-based candidate selection rule together with the accuracy criterion and the fallback it triggers; Appendix B defines the three comparison methods; Appendix C gives the projection and measure parameters, including library versions and the out-of-sample transform protocol; and Appendix D gives the statistical procedure behind every win–loss difference reported in the paper. All comparisons use the same 10 shared seeds, and every number in the paper is produced by a script from stored per-seed artifacts rather than transcribed. Code for training, projection, evaluation, and for generating every table and figure is available at https://anonymous.4open.science/r/paft-0C15/.

REFERENCES

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, 2019.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction-repulsion spectrum in neighbor embeddings. *Journal of Machine Learning Research*, 23(95):1–32, 2022.

Tara Chari and Lior Pachter. The specious art of single-cell genomics. *PLOS Computational Biology*, 19(8):e1011288, 2023.

Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485): 209–219, 2009.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 215–223, 2011.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2153–2173, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, 2013.

Hyeon Jeon, Hyung-Kwon Ko, Jaemin Jo, Youngtaek Kim, and Jinwook Seo. Measuring and explaining the inter-cluster reliability of multidimensional projections. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):551–561, 2022.

Hyeon Jeon, Aeri Cho, Jinhwa Jang, Soohyun Lee, Jake Hyun, Hyung-Kwon Ko, Jaemin Jo, and Jinwook Seo. ZADU: A Python library for evaluating the reliability of dimensionality reduction embeddings. In *IEEE Visualization and Visual Analytics (VIS)*, pp. 196–200, 2023.

Hyeon Jeon, Yun-Hsin Kuo, Michaël Aupetit, Kwan-Liu Ma, and Jinwook Seo. Classes are not clusters: Improving label-based evaluation of dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):781–791, 2024.

Dmitry Kobak and George C. Linderman. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology*, 39(2):156–157, 2021.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*, pp. 5338–5348, 2020.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Joseph B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Technical report, Stanford University, CS231N, 2015.

John A. Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7–9):1431–1443, 2009.

John A. Lee, Emilie Renard, Guillaume Bernard, Pierre Dupont, and Michel Verleysen. Type 1 and 2 mixtures of Kullback–Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, 112:92–108, 2013.

John A. Lee, Diego H. Peluffo-Ordóñez, and Michel Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Karl Pearson. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907, 2021.

Zhenmei Shi, Yifei Ming, Ying Fan, Frederic Sala, and Yingyu Liang. Domain generalization via nuclear norm regularization. In *Conference on Parsimony and Learning (CPAL)*, volume 234 of *Proceedings of Machine Learning Research*, pp. 179–201, 2024.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? Data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022.

Laurens van der Maaten. Learning a parametric embedding by preserving local structure. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 384–391, 2009.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Jarkko Venna and Samuel Kaski. Local multidimensional scaling. *Neural Networks*, 19(6–7):889–899, 2006.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021.

Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-SNE effectively. *Distill*, 2016.

Ross Wightman. PyTorch image models. `https://github.com/huggingface/pytorch-image-models`, 2019.

A Training hyperparameters

All configurations use Adam with a learning rate of 10^{-4}, batch size 256, 224×224 input resolution, and the same data-augmentation pipeline. Training images are resized to 256 pixels on the shorter side, randomly cropped to 224×224 and randomly flipped horizontally, then normalized with the ImageNet mean and standard deviation; horizontal flipping is disabled for SVHN. Validation and test images are resized to 256 pixels and center-cropped to 224×224. The pretrained ImageNet classifier is removed and a randomly initialized linear classification head is trained. Training runs for at most 150 epochs with early stopping on validation accuracy using patience 10, and the checkpoint with the best validation accuracy is restored for evaluation.

Data splits and seeds. For every dataset, 10% of each class in the official training split is held out as the validation split, using a fixed random seed shared by all runs; the remaining images are used for training. Testing uses the official test split (for TinyImageNet, the official validation split). Runs use seeds 0–9, which set the Python, NumPy and PyTorch random number generators, with deterministic PyTorch algorithms enabled. Seed s of one method is paired with seed s of another in all comparisons. For Figures 2–4, we rank the ten seeds by PAFT's margin over the strongest competing method among those displayed and show the upper of the two middle seeds. The margin uses test-set PCA silhouette for Figure 2 and the sum of test-set Spearman-correlation margins between high- and two-dimensional radius over t-SNE, UMAP, and PaCMAP for Figures 3–4.

Neighborhood preservation. The target distribution Π in Eq. (3) is computed within each training minibatch with perplexity 30, and each point's bandwidth is found with 32 bisection steps.

Computational overhead. For minibatch size B, penultimate width D, bottleneck width d and C classes, the bottleneck and the auxiliary classifier add $O(BDd)$ and $O(BdC)$ work, and prediction consistency adds $O(BC)$. Neighborhood preservation forms pairwise geometry in $O(B^2(D+d))$ time with $O(B^2)$ pairwise storage, and its target bandwidths are obtained with 32 vectorized bisection iterations under `no_grad`. Our current linear-CKA implementation forms $B \times B$ Gram

matrices and applies centering through explicit matrix products, which introduces an $O(B^3)$ operation; this is a property of the implementation rather than an inherent requirement of linear CKA. All runs use $B = 256$. Across the validation-selected PAFT settings, per-epoch training time exceeds untreated fine-tuning by 2.0% to 17.5%; these jobs ran on shared compute with unmatched hosts, so the range indicates the scale of the overhead rather than a controlled benchmark. We report per-epoch rather than whole-run time because early stopping changes the number of epochs. The auxiliary branch adds at most 0.62% of the backbone parameters.

Search grids. PAFT searches $\alpha \in \{0.01, 0.05, 0.1, 0.25, 0.5, 1\}$ with bottleneck width $d \in \{16, 64\}$, and the top-d variance penalty searches the same coefficient set with target dimension $d \in \{16, 64\}$. The search caps α at 1 for both PAFT and the top-d variance baseline; earlier exploratory runs with larger coefficients degraded classification accuracy. The nuclear-norm penalty searches $\{0.001, 0.005, 0.01, 0.025, 0.05, 0.1\}$. The narrow-penultimate baseline searches widths $\{2, 16, 64, 128, 256\}$, with 512 and 1024 additionally included for ResNet-50.

Candidate selection. Selection uses validation data only. For each method and backbone–dataset combination, every candidate setting is first checked against the classification-accuracy criterion (Appendix D), and settings that fail it are discarded. Each remaining setting is compared with the untreated baseline on the validation split under the significance criteria of Section 4.1, over all 80 projection–measure pairs of the four projection methods, and the setting with the highest validation win–loss difference is selected. A single setting is therefore chosen per method and combination and shared by the four projection methods, rather than one per projection. The selected setting is fixed for test evaluation, where PAFT is compared directly with each method. If no setting of a method satisfies the accuracy criterion, that method falls back to the untreated baseline for the combination. Table 5 lists the selected settings.

Narrow-penultimate reporting. No narrow-penultimate setting satisfies the accuracy criterion on four backbone–dataset combinations, marked as fallback in Table 5. Primary aggregate comparisons use the untreated baseline for these combinations. Tables 19 and 20 instead report, for descriptive purposes, the narrow-penultimate width with the smallest mean validation-accuracy drop. Figures 3 and 4 likewise display the descriptive ResNet-50/CUB-200 narrow candidate (width 512) rather than the aggregate fallback.

Additional analyses. The objective-term ablations, fitted-branch control, and bottleneck-width sweep use the validation-selected PAFT setting for each backbone–dataset combination. The width sweep additionally evaluates $d \in \{2, 8, 32\}$ using the same coefficient; widths 16 and 64 come from the main search grid. These additional runs, including the $d = 2$ runs in Table 2 and Figure 2, are not used for candidate selection.

B Comparison methods

The comparison methods use the same backbones, training recipe and selection procedure as PAFT and have no auxiliary branch. The two penalties act on the minibatch matrix $F \in \mathbb{R}^{B\times D}$ of penultimate representations. Let $\sigma_1 \geq \sigma_2 \geq \dots$ be the singular values of its column-centered version F_c; they are computed at every training step, and gradients flow through them.

Nuclear-norm penalty. The training loss is

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B} \mathrm{CE}\big(h_{\mathrm{main}}(f_i), y_i\big) + \beta\,\frac{\|F_c\|_*}{\|F_c\|_{\mathrm{F}}}, \qquad \frac{\|F_c\|_*}{\|F_c\|_{\mathrm{F}}} = \frac{\sum_i \sigma_i}{\big(\sum_i \sigma_i^2\big)^{1/2}}.$$

Dividing by the Frobenius norm makes the penalty invariant to the scale of F; it ranges from 1 for a rank-one batch to at most $\sqrt{\min(B, D)}$, attained when all singular values are equal. The penalty has no target dimension. Its coefficient grid is one tenth of the PAFT grid (Appendix A).

Table 5: Validation-selected settings of the four training methods for each backbone–dataset combination: the width n for the narrow-penultimate baseline, β for the nuclear-norm penalty, (β, d) for the top-d variance penalty and (α, d) for PAFT. Fallback marks combinations where no narrow-penultimate width satisfies the accuracy criterion.

Backbone	Dataset	Narrow width	Nuclear-norm	Top-d variance	**PAFT**
ViT-S/16	SVHN	16	0.005	1, 16	0.1, 64
	STL-10	16	0.01	0.5, 16	0.5, 16
	CIFAR-100	128	0.01	1, 64	0.1, 64
	TinyImageNet	256	0.005	0.1, 64	0.1, 16
	CUB-200	256	0.01	1, 64	0.25, 64
ResNet-50	SVHN	16	0.001	0.25, 64	1, 64
	STL-10	256	0.01	0.25, 16	0.5, 16
	CIFAR-100	128	0.005	0.25, 64	0.05, 16
	TinyImageNet	128	0.005	0.01, 64	0.05, 16
	CUB-200	fallback	0.025	0.25, 64	1, 64
ResNet-18	SVHN	16	0.01	0.5, 16	1, 16
	STL-10	16	0.025	1, 64	1, 16
	CIFAR-100	fallback	0.01	0.25, 64	0.5, 16
	TinyImageNet	fallback	0.01	0.25, 64	0.25, 16
	CUB-200	fallback	0.01	0.25, 16	0.5, 64

Top-d variance penalty. The training loss is

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B} \mathrm{CE}\big(h_{\mathrm{main}}(f_i), y_i\big) + \beta\Big(1 - \frac{\sum_{i\le d}\sigma_i^2}{\sum_i \sigma_i^2}\Big),$$

the fraction of minibatch variance outside the d leading principal directions of F_c. The penalty is invariant to the scale of F and lies in $[0, 1]$. Here d is only the target dimension of the penalty: no bottleneck layer is added, and the penultimate representation keeps its full width D.

Narrow-penultimate baseline. A linear layer (PyTorch `nn.Linear`) from $\mathbb{R}^D$ to $\mathbb{R}^n$ is inserted between the backbone's pooled output and the classification head, which then maps $\mathbb{R}^n$ to the class logits. Its n-dimensional output replaces the penultimate representation: it is the representation used by the classifier, projected by the four projection methods, and used as the high-dimensional reference of the projection-quality measures. No nonlinearity or normalization is added, so the representation has rank at most n. Both new layers are randomly initialized with the PyTorch defaults, and the model is trained with cross-entropy only. Unlike the two penalties, which encourage lower rank, this baseline restricts representation dimensionality through the architecture.

C Measure and projection parameters

Rank-based measures use a neighborhood of $K = 15$. Q_{local} and Q_{global} split the $R_{\mathrm{NX}}(K)$ curve at a *fixed* $K = 50$ rather than at each run's own curve peak: the peak-based split is undefined for a non-negligible fraction of PCA runs, and a split that moves with the run being scored is not comparable across arms. AUC $R_{\mathrm{NX}}(K)$ is counted as local because its $1/K$ weighting makes small neighborhoods dominate. k-NN accuracy and neighborhood hit use the 15 nearest neighbors in the projection, random-triplet accuracy samples 50 triplets per point with a fixed seed, and silhouette uses Euclidean distances in the projection. Steadiness, cohesiveness, label-trustworthiness and label-continuity use the ZADU 0.4.2 implementations (Jeon et al., 2023) with default parameters, except that steadiness and cohesiveness use 600 iterations; their random sampling is seeded with the training seed, so that paired runs share the same samples. Label-trustworthiness and label-continuity use the default class-pair distance-consistency criterion.

Projection methods. We use scikit-learn 1.6.0 for PCA, openTSNE 1.0.4 for t-SNE, umap-learn 0.5.11 for UMAP and PaCMAP 0.9.1, with two output dimensions and library-default parameters;

only the random state is set, to the training seed. Each trained model receives its own fit of each method on the full validation split, and the validation measures used for selection are computed on this fitted embedding. Test points are embedded without refitting. PCA applies its fitted linear map. The t-SNE, UMAP and PaCMAP transforms, with their library-default settings, optimize only the coordinates of the test points relative to the fixed validation embedding. Test measures use all test points, except for SVHN, where a fixed random subset of 10 000 points is used.

Bottleneck-side projections as a sanity check. For PAFT, a post-hoc projection can in principle be applied to the learned bottleneck p rather than to the penultimate representation f. At the validation-selected PAFT settings, comparing the two choices within the same run gives average win–loss differences of $+5.93$ for PCA, -6.33 for t-SNE, -6.47 for UMAP and -5.93 for PaCMAP, out of 20 measures per combination. Projecting the bottleneck is therefore a viable alternative view, and bottleneck-side PCA in particular performs well.

The headline post-hoc comparison nevertheless uses the penultimate representation, because it is the representation that every method in the comparison exposes, and it therefore keeps a common evaluation interface across methods. Granting PAFT access to its auxiliary bottleneck in the cross-method comparison alone would change that interface for one method only. The directly read $d = 2$ analysis of Section 4 asks a different question, namely whether a two-dimensional view is available without fitting any post-hoc projection at all, and is reported separately. This sanity check concerns the jointly trained bottleneck and is unrelated to the separately fitted branches of Appendix H, which are not the same object.

D Statistical procedure

Projection-quality measures. Within a backbone–dataset combination, each of the 80 (projection, measure) combinations is compared between the two arms over the 10 shared seeds. The arms use the same seeds, so the comparison is paired seed by seed. Differences are sign-corrected, so that a positive value means improvement for measures where higher is better and for those where lower is better alike, and tested with a paired t-test. The 80 p-values of that combination are then corrected together for false discovery rate with Benjamini–Hochberg at $q = 0.05$ (Benjamini & Hochberg, 1995). Correcting within the combination keeps the correction independent of how many combinations the study contains.

A significant difference also has to be large enough to matter, so the second condition is relative. The absolute mean difference must reach 1% of the absolute value of the measure's mean, averaged over the two arms. The requirement is there because several measures are close to saturated in the baseline. Label-continuity, for instance, sits above 0.999 before treatment: with 10 matched seeds a shift in its fourth decimal is statistically detectable and visually meaningless. A measure is counted as a win for the first arm when both conditions hold, as a loss in the symmetric case, and as unchanged otherwise. *Unchanged* records that we detected no change of the size we set out to detect; it is not a claim that the two arms are equal.

Classification accuracy. The accuracy criterion runs on the validation split, over the same 10 seeds, and decides whether a candidate setting may be selected at all. A setting whose mean validation accuracy is at or above the baseline's is admitted without further testing. A setting below it is ruled out if its shortfall is at least 1% in relative terms, or if the shortfall is significant. Either condition alone is disqualifying. For accuracy drops below 1%, we perform a one-sided one-sample t-test on the baseline's ten seed accuracies, using the candidate's mean as the reference ($p \leq 0.05$). We use the baseline's seed variability so that a candidate's higher seed variability cannot increase the test's standard error and make an accuracy drop less likely to be significant. The criterion is applied to PAFT and to every competing method in the same form, and a method left with no admissible setting for a combination falls back to the baseline for that combination. Combining the two conditions with *or* rather than *and* places the burden of proof on the side of preserving accuracy: a setting has to clear both to be used, which is the stricter reading of the constraint and the opposite of how the two conditions are combined for the projection measures.

Accuracy-gate sensitivity. Across all 808 candidate configurations to which we applied the accuracy gate during the study, including configurations from the primary search grids and additional

auxiliary and ablation analyses, the accuracy criterion rejects 216, including 134 rejected solely by the significance condition despite a relative validation-accuracy drop below 1%. Among these 134 candidates, the median observed drop is 0.28 percentage points. The significance test's effective drop threshold depends on the untreated baseline's accuracy variability across seeds; its median across the 15 backbone–dataset combinations is approximately 0.15 percentage points. This condition is more restrictive than the 1% relative-drop threshold in 14 of the 15 combinations.

Accuracy-gate sensitivity. Average win–loss difference of PAFT against each comparator, per projection and out of 20 measures, under the original accuracy criterion and an alternative using only the 1% relative validation-accuracy drop threshold. The alternative reselects configurations using validation data.

Comparator	Original gate	1%-only gate
Untreated baseline	+12.6	+12.8
Narrow-penultimate baseline	+6.0	+4.8
Nuclear-norm penalty	+9.7	+9.4
Top-d variance penalty	+8.2	+8.2

Under the 1%-only rule, 18 of the 75 method–combination settings change, including 4 PAFT settings, and the number of narrow-penultimate fallback combinations decreases from 4 to 2. The changes in the win–loss comparisons are reported in the table above.

E RAW PROJECTION-QUALITY MEASURES BY DATASET

Tables 6–10 report all 20 projection-quality measures separately for each dataset, comparing the untreated baseline, the three rank-reduction alternatives, and PAFT. Each entry averages the corresponding raw test metric over the four projection methods and three backbones, with each backbone–projection combination contributing equally. For combinations without an eligible narrow-penultimate setting, its averages include the untreated-baseline results under the fallback rule in Appendix A.

Table 6: Raw projection-quality measures on SVHN, averaged over the three backbones and four projection methods. Each backbone–projection value is the 10-seed test mean at the method's validation-selected operating point. The four projection methods and three backbones are macro-averaged with equal weight. Best values per row are bold; ties at the reported precision are left unmarked; ↓ indicates lower is better.

Metric	Untreated baseline	Narrow-penultimate baseline	Nuclear-norm penalty	Top-d variance penalty	**PAFT**
Local geometry					
Trustworthiness	0.939	0.959	0.963	0.971	**0.973**
Continuity	0.967	0.983	0.979	**0.984**	0.980
$\mathrm{AUC}_{R_{NX}}$	0.249	0.337	0.327	**0.362**	0.349
Q_{local}	0.117	0.214	0.174	**0.226**	0.161
LCMC	0.093	0.183	0.143	**0.191**	0.132
$\mathrm{MRRE}_{h\to l}\downarrow$	0.031	0.015	0.020	0.015	0.019
$\mathrm{MRRE}_{l\to h}\downarrow$	0.061	0.041	0.037	0.029	**0.027**
Global geometry					
Dist. correlation	0.540	0.666	0.685	0.701	**0.827**
Shepard goodness	0.416	0.598	0.642	0.670	**0.822**
Norm. stress ↓	0.400	0.343	0.346	0.340	**0.283**
Triplet accuracy	0.658	0.727	0.757	0.764	**0.823**
Q_{global}	0.306	0.434	0.478	0.496	**0.592**
Label structure					
kNN accuracy	0.913	0.918	0.926	0.933	**0.947**
Neighborhood hit	0.877	0.883	0.895	0.903	**0.922**
Silhouette	0.573	0.507	0.546	0.524	**0.593**
Distance consistency	0.907	0.910	0.913	0.918	**0.928**
Cluster-level					
Steadiness	0.733	0.742	0.768	0.789	**0.800**
Cohesiveness	0.711	**0.760**	0.726	0.755	0.735
Label continuity	1.000	1.000	0.999	1.000	0.999
Label trustworthiness	0.977	0.982	0.982	0.984	**0.991**

Table 7: Raw projection-quality measures on STL-10, averaged over the three backbones and four projection methods. Each backbone–projection value is the 10-seed test mean at the method's validation-selected operating point. The four projection methods and three backbones are macro-averaged with equal weight. Best values per row are bold; ties at the reported precision are left unmarked; ↓ indicates lower is better.

Metric	Untreated baseline	Narrow-penultimate baseline	Nuclear-norm penalty	Top-d variance penalty	**PAFT**
Local geometry					
Trustworthiness	0.958	**0.971**	0.963	0.967	0.967
Continuity	0.973	**0.985**	0.975	0.977	0.974
$\text{AUC}_{R_{NX}}$	0.299	**0.364**	0.324	0.331	0.320
Q_{local}	0.147	**0.233**	0.168	0.176	0.152
LCMC	0.116	**0.197**	0.135	0.142	0.121
$\text{MRRE}_{h\to l}$ ↓	0.026	**0.014**	0.024	0.022	0.024
$\text{MRRE}_{l\to h}$ ↓	0.042	**0.029**	0.037	0.033	0.033
Global geometry					
Dist. correlation	0.650	0.707	0.700	0.700	**0.741**
Shepard goodness	0.608	0.666	0.688	0.689	**0.719**
Norm. stress ↓	0.402	**0.347**	0.385	0.381	0.363
Triplet accuracy	0.731	0.758	0.761	0.764	**0.772**
Q_{global}	0.429	0.483	0.488	0.493	**0.502**
Label structure					
kNN accuracy	0.918	0.927	0.914	0.921	**0.934**
Neighborhood hit	0.889	0.899	0.883	0.892	**0.908**
Silhouette	0.640	0.639	0.619	0.620	**0.669**
Distance consistency	0.912	0.922	0.901	0.905	**0.925**
Cluster-level					
Steadiness	0.750	0.772	0.760	**0.774**	0.773
Cohesiveness	0.714	**0.741**	0.698	0.713	0.709
Label continuity	1.000	1.000	1.000	0.999	1.000
Label trustworthiness	0.988	0.989	0.985	0.985	0.989

Table 8: Raw projection-quality measures on CIFAR-100, averaged over the three backbones and four projection methods. Each backbone–projection value is the 10-seed test mean at the method's validation-selected operating point. The four projection methods and three backbones are macro-averaged with equal weight. Best values per row are bold; ties at the reported precision are left unmarked; ↓ indicates lower is better.

Metric	Untreated baseline	Narrow-penultimate baseline	Nuclear-norm penalty	Top-d variance penalty	**PAFT**
Local geometry					
Trustworthiness	0.919	0.928	0.940	0.944	**0.965**
Continuity	0.962	0.968	0.968	0.970	**0.978**
$\text{AUC}_{R_{NX}}$	0.264	0.284	0.301	0.315	**0.354**
Q_{local}	0.277	0.297	0.287	0.298	**0.305**
LCMC	0.221	0.240	0.231	0.242	**0.244**
$\text{MRRE}_{h\to l}\downarrow$	0.035	0.030	0.030	0.028	**0.021**
$\text{MRRE}_{l\to h}\downarrow$	0.081	0.072	0.060	0.056	**0.035**
Global geometry					
Dist. correlation	0.322	0.375	0.492	0.516	**0.646**
Shepard goodness	0.285	0.343	0.472	0.497	**0.644**
Norm. stress ↓	0.443	0.427	0.418	0.422	**0.379**
Triplet accuracy	0.608	0.629	0.678	0.686	**0.749**
Q_{global}	0.208	0.245	0.334	0.350	**0.452**
Label structure					
kNN accuracy	0.612	0.621	0.614	0.619	**0.647**
Neighborhood hit	0.547	0.554	0.546	0.548	**0.575**
Silhouette	0.077	0.089	0.069	0.069	**0.096**
Distance consistency	0.532	0.546	0.530	0.536	**0.555**
Cluster-level					
Steadiness	0.736	0.739	0.735	0.737	**0.756**
Cohesiveness	0.732	0.732	0.735	**0.738**	0.730
Label continuity	1.000	1.000	1.000	1.000	1.000
Label trustworthiness	0.959	0.963	0.962	0.963	**0.971**

Table 9: Raw projection-quality measures on TinyImageNet, averaged over the three backbones and four projection methods. Each backbone–projection value is the 10-seed test mean at the method's validation-selected operating point. The four projection methods and three backbones are macro-averaged with equal weight. Best values per row are bold; ties at the reported precision are left unmarked; ↓ indicates lower is better.

Metric	Untreated baseline	Narrow-penultimate baseline	Nuclear-norm penalty	Top-d variance penalty	**PAFT**
Local geometry					
Trustworthiness	0.901	0.917	0.913	0.911	**0.940**
Continuity	0.948	0.955	0.952	0.952	**0.964**
$\text{AUC}_{R_{NX}}$	0.242	0.267	0.259	0.255	**0.309**
Q_{local}	0.334	0.346	0.333	0.334	**0.351**
LCMC	0.291	0.303	0.291	0.292	**0.307**
$\text{MRRE}_{h\to l}$ ↓	0.047	0.040	0.043	0.044	**0.033**
$\text{MRRE}_{l\to h}$ ↓	0.097	0.082	0.086	0.088	**0.059**
Global geometry					
Dist. correlation	0.250	0.342	0.321	0.299	**0.543**
Shepard goodness	0.241	0.335	0.317	0.294	**0.548**
Norm. stress ↓	0.444	0.428	0.437	0.438	**0.411**
Triplet accuracy	0.596	0.631	0.623	0.614	**0.715**
Q_{global}	0.176	0.237	0.223	0.209	**0.379**
Label structure					
kNN accuracy	0.530	0.534	0.527	0.527	**0.541**
Neighborhood hit	0.471	0.473	0.467	0.467	**0.478**
Silhouette	-0.108	**-0.101**	-0.113	-0.111	-0.109
Distance consistency	0.331	**0.342**	0.329	0.329	0.337
Cluster-level					
Steadiness	0.662	0.657	0.669	0.669	**0.685**
Cohesiveness	0.765	**0.767**	0.762	0.762	0.728
Label continuity	1.000	1.000	1.000	1.000	1.000
Label trustworthiness	0.939	0.945	0.941	0.941	**0.949**

Table 10: Raw projection-quality measures on CUB-200, averaged over the three backbones and four projection methods. Each backbone–projection value is the 10-seed test mean at the method's validation-selected operating point. The four projection methods and three backbones are macro-averaged with equal weight. Best values per row are bold; ties at the reported precision are left unmarked; ↓ indicates lower is better.

Metric	Untreated baseline	Narrow-penultimate baseline	Nuclear-norm penalty	Top-d variance penalty	**PAFT**
Local geometry					
Trustworthiness	0.906	0.910	0.920	0.925	**0.954**
Continuity	0.941	0.944	0.947	0.948	**0.965**
$\text{AUC}_{R_{NX}}$	0.241	0.247	0.263	0.271	**0.335**
Q_{local}	0.229	0.237	0.235	0.242	**0.263**
LCMC	0.204	0.212	0.209	0.216	**0.233**
$\text{MRRE}_{h\to l}\downarrow$	0.053	0.051	0.048	0.047	**0.032**
$\text{MRRE}_{l\to h}\downarrow$	0.093	0.089	0.080	0.075	**0.046**
Global geometry					
Dist. correlation	0.418	0.436	0.496	0.518	**0.728**
Shepard goodness	0.393	0.416	0.484	0.504	**0.738**
Norm. stress ↓	0.436	0.431	0.435	0.430	**0.397**
Triplet accuracy	0.643	0.651	0.678	0.682	**0.778**
Q_{global}	0.262	0.276	0.328	0.335	**0.503**
Label structure					
kNN accuracy	0.326	0.332	0.327	0.331	**0.353**
Neighborhood hit	0.244	0.249	0.243	0.246	**0.267**
Silhouette	-0.277	-0.273	-0.283	-0.286	**-0.264**
Distance consistency	0.230	0.233	0.228	0.228	**0.255**
Cluster-level					
Steadiness	0.657	0.650	0.667	0.670	**0.705**
Cohesiveness	**0.727**	0.726	0.721	0.721	0.717
Label continuity	1.000	1.000	1.000	1.000	1.000
Label trustworthiness	0.946	0.947	0.947	0.946	**0.959**

F REPRESENTATION RADIUS AND CLASS-LEVEL GEOMETRY

Figure 4 extends the representation-radius visualization of Figure 3 across all four projection methods.

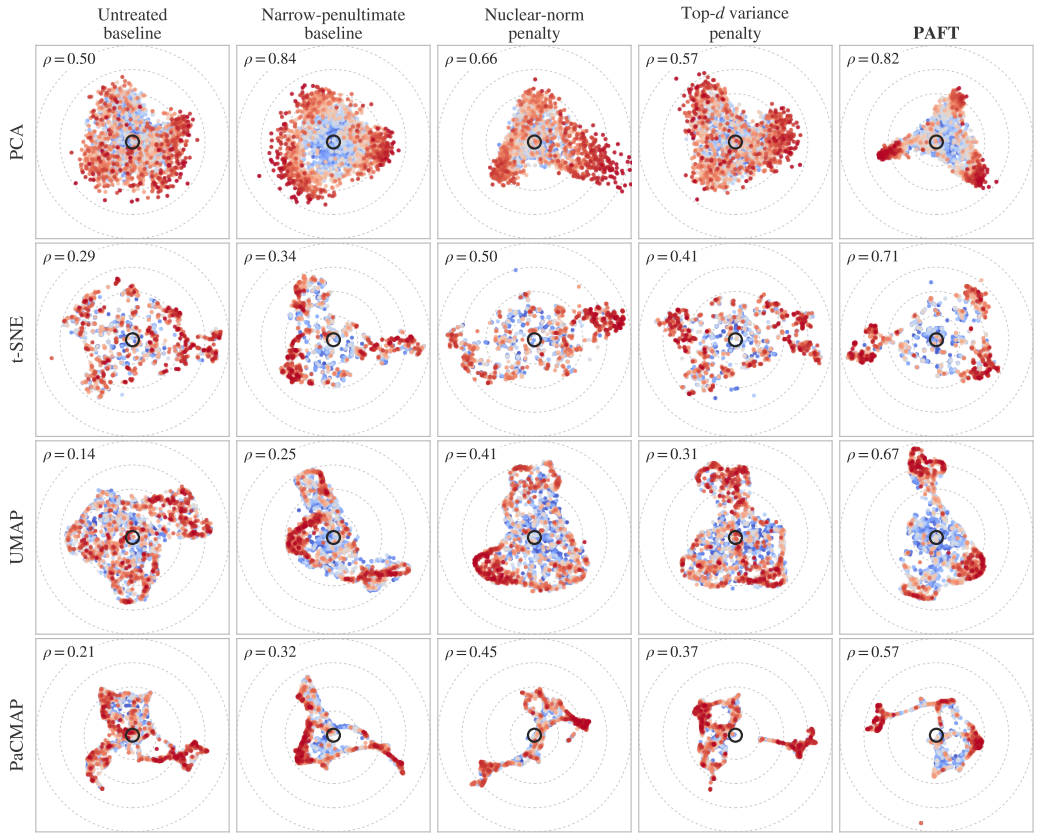


Figure 4: The high-dimensional representation radius on PCA, t-SNE, UMAP, and PaCMAP maps for the same backbone–dataset combination, method configurations, and seed as Figure 3, including the same illustrative narrow-penultimate candidate. Each point is colored by its distance from the center of the model's penultimate representations on the test set, expressed as a percentile; each panel reports the Spearman correlation ρ between high- and two-dimensional radius for the displayed samples.

To quantify this preservation beyond the illustrated example, we compute the Spearman correlation between high-dimensional and two-dimensional radius across the full evaluation. PAFT gives the highest correlation under every projection method, with a mean of 0.446, compared with 0.110 for the untreated baseline, 0.286 for the narrow-penultimate baseline, 0.310 for the nuclear-norm penalty, and 0.345 for the top-d variance penalty.

Class-level geometry and class confusion. The radius analysis concerns sample-level geometry. We next ask whether a two-dimensional view also preserves relationships between classes that can be useful when inspecting model behavior. For each model, we compute class centroids over the test samples in its penultimate representation and in each two-dimensional view, form the corresponding pairwise Euclidean distance matrices, and compare them using Spearman correlation, denoted ρ_{geom}. This comparison is entirely within the same model and therefore measures how much of its high-dimensional between-class geometry survives projection.

PAFT preserves this geometry most faithfully under all four projection methods. The ordering is the same in every case: untreated baseline, narrow-penultimate baseline, nuclear-norm penalty, top-d variance penalty, and PAFT.

Table 11: Spearman correlation between high-dimensional representation radius and the corresponding two-dimensional radius. Entries are averaged over the 10 shared seeds and backbone–dataset combinations. The narrow-penultimate baseline uses the 11 combinations with an accuracy-eligible setting; the other methods use all 15.

Method	PCA	t-SNE	UMAP	PaCMAP	Mean
Untreated baseline	0.209	0.101	0.018	0.111	0.110
Narrow-penultimate baseline	0.524	0.235	0.126	0.260	0.286
Nuclear-norm penalty	0.504	0.271	0.187	0.278	0.310
Top-d variance penalty	0.511	0.311	0.239	0.318	0.345
PAFT	**0.652**	**0.360**	**0.319**	**0.452**	**0.446**

Table 12: Preservation of high-dimensional between-class geometry. Entries report the Spearman correlation ρ_{geom} between pairwise class-centroid distances in the penultimate representation and in each two-dimensional view, averaged over the 10 shared seeds and all 15 backbone–dataset combinations. For the narrow-penultimate baseline, the four combinations without an accuracy-eligible setting use the untreated-baseline fallback of Appendix A.

Method	PCA	t-SNE	UMAP	PaCMAP
Untreated baseline	0.519	0.533	0.458	0.558
Narrow-penultimate baseline	0.647	0.556	0.499	0.593
Nuclear-norm penalty	0.737	0.603	0.552	0.614
Top-d variance penalty	0.749	0.608	0.579	0.633
PAFT	**0.909**	**0.715**	**0.675**	**0.773**

We then ask whether the preserved class geometry reflects a property of the classifier itself. For classes $i \neq j$, we define M_{ij} as the mean predicted probability of class j over test samples whose true label is i, and symmetrize the matrix as $M + M^\top$. We compute ρ_{conf} as the Spearman correlation between the upper-triangular off-diagonal entries of this class-confusion matrix and the negated corresponding pairwise class-centroid distances, so that positive values indicate greater confusion between geometrically closer classes. The high-dimensional representations already contain a substantial geometry–confusion relationship across all methods; the relevant question here is how much of that relationship remains accessible after projection.

Table 13: Association between between-class geometry and class confusion. Entries are averaged over the 10 shared seeds and all 15 backbone–dataset combinations. For the narrow-penultimate baseline, the four combinations without an accuracy-eligible setting use the untreated-baseline fallback of Appendix A. The penultimate column reports ρ_{conf} in the high-dimensional representation; the remaining columns report the same quantity in each two-dimensional view.

Method	Penultimate	PCA	t-SNE	UMAP	PaCMAP
Untreated baseline	**0.757**	0.472	0.557	0.526	0.548
Narrow-penultimate baseline	0.732	0.476	0.538	0.549	0.536
Nuclear-norm penalty	0.720	0.521	0.586	**0.553**	0.578
Top-d variance penalty	0.720	0.528	0.571	0.549	0.580
PAFT	0.725	**0.606**	**0.602**	0.541	**0.598**

The high-dimensional geometry–confusion correlations lie in a relatively narrow range across methods, from 0.720 to 0.757. After projection, PAFT has the highest mean correlation across the four views (0.587), compared with 0.526 for the untreated baseline, 0.525 for the narrow-penultimate baseline, 0.559 for the nuclear-norm penalty, and 0.557 for the top-d variance penalty. PAFT leads under PCA, t-SNE, and PaCMAP, while UMAP is the exception.

Under the same statistical procedure used in the main evaluation, the combined family of four ρ_{geom} and four ρ_{conf} measures gives PAFT positive average win–loss differences against all four com-

parators: $+5.6$ against the untreated baseline, $+5.8$ against the narrow-penultimate baseline, $+4.8$ against the nuclear-norm penalty, and $+4.9$ against the top-d variance penalty.

G OBJECTIVE-TERM ABLATIONS

We report the detailed objective-term ablations underlying Section 4.4. Table 14 compares the full objective with each single term and each pair of terms at the PAFT setting described in Appendix A.

Table 14: Aggregate objective-term ablation over all 15 backbone–dataset combinations, at the validation-selected PAFT settings. Projection-quality entries are win–loss differences against full PAFT over four projection methods applied to the penultimate representations, averaged per projection; negative values favor full PAFT. Main-head test accuracy is macro-averaged over the 15 combinations after averaging each combination over its 10 seeds, and Δ is relative to full PAFT in percentage points. The accuracy columns report the trained ablation arms directly; the projection-quality comparison additionally applies the accuracy-gate fallback of Appendix A, which affects 9 of the 90 arm–combination cells.

Objective terms retained	Win–loss difference	Main-head test acc. (%)	Δ vs. full PAFT (pp)
Full PAFT	–	85.77	0.00
CKA + Neighborhood preservation	−1.73	85.68	−0.09
Prediction consistency + Neighborhood preservation	−3.77	85.76	−0.01
Neighborhood preservation only	−5.58	85.66	−0.11
Prediction consistency + CKA	−11.18	85.52	−0.25
Prediction consistency only	−10.90	85.58	−0.19
CKA only	−12.30	85.62	−0.15

Prediction consistency plays a different role for the auxiliary classifier. The variants that retain this term reach a mean auxiliary-head test accuracy of 83–84%, close to full PAFT at 83.65%, whereas the variants without it average about 5%. This is consistent with the optimization graph, in which the auxiliary classifier receives a training signal only through the prediction-consistency term.

H LEARNING THE PROJECTION VS. SHAPING THE REPRESENTATION

We compare PCA of the untreated baseline's penultimate representation with PCA of bottlenecks from auxiliary branches fitted to frozen baseline and PAFT representations. Both branches are optimized on the frozen training-set representations using the three auxiliary objectives at unit weight, Adam with learning rate 10^{-3}, and validation-loss early stopping with patience 30; the checkpoint with the lowest validation loss is restored. The baseline branch is randomly initialized, whereas the PAFT branch starts from its jointly trained weights, which also remain a candidate checkpoint. Both bottlenecks have the width of the validation-selected PAFT setting described in Appendix A. Each arm is evaluated against its own penultimate representation, and the fitted-baseline control leaves the original classifier unchanged. Comparisons are paired over the 10 seeds, with Benjamini–Hochberg correction applied to the 20 PCA measures within each backbone–dataset combination (Appendix D).

Table 15: Raw PCA projection-quality measures for the projection-learning control on ResNet-18/CUB-200, averaged over 10 seeds. PCA is applied to the untreated baseline's penultimate representation and to bottlenecks from auxiliary branches fitted to frozen baseline and PAFT representations. Both fitted branches use the same optimization procedure; the PAFT branch is initialized from its jointly trained weights. Each arm is evaluated against its own penultimate representation. Best values per row are bold; ties at the reported precision are left unmarked; ↓ indicates lower is better.

Metric	Baseline	Frozen baseline + fitted branch $d=64$	Frozen **PAFT** + fitted branch $d=64$
Local geometry			
Trustworthiness	0.793	0.807	**0.883**
Continuity	0.939	0.946	**0.959**
$\text{AUC}_{R_{NX}}$	0.170	0.180	**0.248**
Q_{local}	0.062	0.072	**0.091**
LCMC	0.046	0.054	**0.070**
$\text{MRRE}_{h\to l}$ ↓	0.053	0.047	**0.036**
$\text{MRRE}_{l\to h}$ ↓	0.208	0.193	**0.118**
Global geometry			
Dist. correlation	0.600	0.597	**0.835**
Shepard goodness	0.585	0.581	**0.830**
Norm. stress ↓	0.386	0.395	**0.358**
Triplet accuracy	0.722	0.721	**0.832**
Q_{global}	0.392	0.393	**0.616**
Label structure			
kNN accuracy	0.078	0.094	**0.111**
Neighborhood hit	0.053	0.062	**0.072**
Silhouette	-0.311	**-0.301**	-0.307
Distance consistency	0.104	0.116	**0.124**
Cluster-level			
Steadiness	0.549	0.568	**0.615**
Cohesiveness	0.661	**0.678**	0.673
Label continuity	1.000	1.000	1.000
Label trustworthiness	0.940	0.947	**0.950**

I SPECTRAL CHARACTERIZATION AND RANK-REDUCTION CONTROLS

To characterize how PAFT and the comparison methods reorganize the penultimate representation, we examine both scalar rank summaries and the cumulative explained-variance profile of the centered penultimate features. For singular values $\{\sigma_i\}$, we use effective rank $r_{\text{eff}} = \exp(-\sum_i p_i \log p_i)$, where $p_i = \sigma_i / \sum_j \sigma_j$, and stable rank $r_{\text{stable}} = \sum_i \sigma_i^2 / \sigma_1^2$. Table 16 reports PAFT's same-seed ratios to the untreated baseline for every backbone–dataset combination. PAFT is evaluated on its penultimate representation f, not on the auxiliary bottleneck.

PAFT reduces both summaries relative to the baseline, but to different degrees: the mean effective-rank ratio across the 15 pairs is 0.85, whereas the mean stable-rank ratio is 0.38. To examine how this compression is distributed across the spectrum, let $k(\tau)$ denote the minimum number of principal components required to explain a fraction τ of the variance. We report $k(\tau)$ relative to the corresponding untreated baseline within each seed, together with the fraction of variance captured by the first two principal components, again relative to the baseline.

The four interventions produce qualitatively different spectral profiles. PAFT concentrates variance toward the leading directions while retaining much of the spectral tail: relative to the baseline, it requires a median of 0.43 times as many components to explain 50% of the variance and 0.88 times as many to explain 99%, while the first two components carry $2.5\times$ their baseline variance share. The nuclear-norm penalty produces a comparatively smoother compression at its selected operating points. The top-d variance penalty preferentially suppresses the middle and upper part of the spectrum around its target subspace, whereas the narrow-penultimate baseline strongly truncates the tail, as expected when the classifier representation itself is made low-dimensional. These profiles

Table 16: Effective- and stable-rank ratios of PAFT's penultimate representation relative to the untreated baseline. Ratios are computed within each shared seed on centered test features and then averaged over the 10 seeds of each backbone–dataset combination; values below one indicate lower rank than the baseline. The final two rows report the mean and median across the 15 combinations. PAFT is evaluated at its selected operating point and on its penultimate representation f, not on the auxiliary bottleneck.

Backbone–dataset combination	Effective rank / baseline	Stable rank / baseline
ResNet-18 / SVHN	0.745	0.348
ResNet-18 / STL-10	0.885	0.623
ResNet-18 / CIFAR-100	0.879	0.279
ResNet-18 / TinyImageNet	0.936	0.254
ResNet-18 / CUB-200	0.925	0.498
ResNet-50 / SVHN	0.694	0.245
ResNet-50 / STL-10	0.926	0.765
ResNet-50 / CIFAR-100	0.954	0.579
ResNet-50 / TinyImageNet	0.962	0.503
ResNet-50 / CUB-200	0.889	0.418
ViT-S/16 / SVHN	0.655	0.260
ViT-S/16 / STL-10	0.643	0.343
ViT-S/16 / CIFAR-100	0.861	0.138
ViT-S/16 / TinyImageNet	0.930	0.249
ViT-S/16 / CUB-200	0.882	0.233
Mean	0.851	0.382
Median	0.885	0.343

Table 17: Cumulative explained-variance profiles of the penultimate representation at the selected operating points. Each $k(\tau)$ entry is the number of principal components required to explain τ of the variance, normalized to the corresponding untreated baseline within each seed; values below one therefore indicate that the same variance fraction is concentrated in fewer directions. The final column gives the variance captured by the first two components relative to the baseline. Entries are medians across the 15 backbone–dataset combinations, except for the narrow-penultimate baseline, which uses the 11 combinations with an eligible setting.

	Relative to baseline					
	k_{50}	k_{80}	k_{90}	k_{95}	k_{99}	First-2 variance
Narrow-penultimate baseline	0.47	0.25	0.10	0.05	0.03	1.9×
Nuclear-norm penalty	0.63	0.56	0.60	0.71	0.88	1.4×
Top-d variance penalty	0.71	0.50	0.36	0.49	0.80	1.4×
PAFT	0.43	0.54	0.59	0.69	0.88	2.5×

characterize how the interventions reshape the representation spectrum; they do not by themselves establish which spectral change causes the differences in projection quality.

Random-projection control. To separate the general effect of concentrating a representation into fewer directions from the fidelity obtained by the projection methods used in our evaluation, we introduce an arbitrary two-dimensional linear map as a negative control. For each penultimate dimension D, we draw a Gaussian $D \times 2$ matrix and orthonormalize its columns by QR decomposition. The map is not fitted to the representation: centering uses the validation mean, after which the fixed map is applied to the test features. The same random draw is shared by methods with the same penultimate dimensionality within a backbone. We use eight draws, analyzing each draw independently over the 10 training seeds, and report the mean of the eight resulting analyses. All conditions in this control are evaluated on the same deterministic subsample of 2,000 test points.

The random map is deliberately weak as a visualization method. Within each intervention, the corresponding fitted projection outperforms the random map on 16 to 19 of the 20 projection-quality measures. Its role is instead to measure how much improvement is obtained when an arbitrary two-dimensional subspace is exposed to a representation whose variance has become more concentrated.

Table 18: Win–loss difference relative to the untreated baseline under an arbitrary random two-dimensional map and under the four projection methods. All columns are recomputed on the same 2,000-point metric subsample used for this control. Random-map values are means over eight complete analyses, each using the 10 shared training seeds. The narrow-penultimate baseline is aggregated over its 11 accuracy-eligible combinations; the other methods use all 15.

Method	Random	PCA	t-SNE	UMAP	PaCMAP
Nuclear-norm penalty	$+13.57$	$+11.00$	$+7.33$	$+7.20$	$+6.53$
Top-d variance penalty	$+14.33$	$+12.33$	$+7.87$	$+7.27$	$+7.20$
Narrow-penultimate baseline	$\mathbf{+17.92}$	$+14.18$	$+10.18$	$+7.82$	$+9.64$
PAFT	$+13.47$	$\mathbf{+16.00}$	$\mathbf{+12.00}$	$\mathbf{+11.13}$	$\mathbf{+12.67}$

All four interventions improve the arbitrary view, indicating that compression contributes to projection quality. The ordering under this control, however, differs from that obtained with the projection methods used in the main evaluation. PAFT has the smallest random-map gain of the four interventions, whereas it has the largest gain under each of PCA, t-SNE, UMAP, and PaCMAP. Thus, the amount of improvement available to an arbitrary two-dimensional projection does not account for PAFT's advantage under the fitted projection methods.

This interpretation is also consistent with the spectral statistics. Across the 56 method–combination points, random-map gain is strongly associated with effective-rank ratio (Spearman $\rho = -0.892$) and k_{99} ratio ($\rho = -0.871$), while its association with stable-rank ratio is weaker ($\rho = -0.398$). Lower ratios denote stronger compression, so the arbitrary projection primarily rewards representations in which variance has been concentrated into fewer directions. The same ordering of these associations remains after excluding the narrow-penultimate baseline, removing its dimensionality difference from the comparison.

The nuclear-norm penalty provides an informative same-width comparison. It has the same median k_{99} ratio as PAFT (0.88) and essentially the same gain under the random map ($+13.57$ versus $+13.47$). Nevertheless, PAFT exceeds its gain over the untreated baseline by 5.00, 4.67, 3.93, and 6.13 points under PCA, t-SNE, UMAP, and PaCMAP, respectively. This comparison further illustrates that similar tail compression and similar behavior under an arbitrary view need not imply similar fidelity under the fitted projections.

A stricter spectrum-matched control cannot be constructed from the existing accuracy-eligible candidate pool. Joint matching of stable-rank and k_{99} ratios within 5% yields no match for any of the three comparison methods in any of the 15 backbone–dataset combinations. We therefore do not interpret these results as isolating PAFT from rank reduction. Rather, the spectral and random-projection analyses jointly show that rank reduction contributes to projection quality, while the magnitude of compression alone does not determine the fidelity obtained under the projection methods considered here.

J Classification accuracy

Tables 19 and 20 report validation and test top-1 classification accuracy, respectively, across all backbone–dataset combinations, as means $\pm$ sample standard deviations over the 10 shared seeds.

At $d = 2$, auxiliary-head accuracy can fall sharply on the many-class datasets, while main-head test accuracy differs from the untreated baseline by -1.06 to $+0.64$ percentage points across the 15 combinations, with a median difference of $+0.00$ percentage points.

Accuracy gate for the $d = 2$ configuration. The validation accuracy gate of Appendix A rejects three of the 15 dedicated $d = 2$ runs: ViT-S/16 on STL-10 by 0.200 percentage points, ResNet-18 on TinyImageNet by 0.113 and ResNet-18 on CUB-200 by 1.667. In Table 4 the left view falls

Table 19: Validation top-1 classification accuracy (%) across all 15 backbone–dataset combinations. Entries are means ± sample standard deviations over the 10 shared seeds. Each method is at its validation-selected, accuracy-eligible operating point, except the narrow-penultimate entries marked †: no narrow-penultimate width satisfies the accuracy criterion for those combinations, so the table reports the width with the smallest mean validation-accuracy drop. These candidates are descriptive only; aggregate comparisons use the untreated baseline for those combinations.

Backbone	Dataset	Untreated baseline	Narrow-penultimate baseline	Nuclear-norm penalty	Top-d variance penalty	**PAFT**
ViT-S/16	SVHN	96.20 ± 0.19	96.16 ± 0.06	96.18 ± 0.13	96.21 ± 0.10	96.27 ± 0.16
	STL-10	98.02 ± 0.33	97.86 ± 0.49	97.94 ± 0.25	97.98 ± 0.37	98.02 ± 0.33
	CIFAR-100	84.71 ± 0.35	84.52 ± 0.32	84.90 ± 0.42	84.64 ± 0.25	84.62 ± 0.17
	TinyImageNet	81.02 ± 0.14	81.02 ± 0.31	81.08 ± 0.13	81.01 ± 0.19	81.05 ± 0.21
	CUB-200	77.68 ± 1.05	79.25 ± 1.10	78.92 ± 1.26	79.38 ± 1.04	78.53 ± 1.02
ResNet-50	SVHN	94.57 ± 0.14	94.59 ± 0.14	94.60 ± 0.17	94.69 ± 0.15	95.06 ± 0.09
	STL-10	97.78 ± 0.26	97.94 ± 0.19	97.90 ± 0.27	97.76 ± 0.41	97.92 ± 0.21
	CIFAR-100	82.48 ± 0.18	82.52 ± 0.20	82.53 ± 0.25	82.50 ± 0.28	82.74 ± 0.20
	TinyImageNet	80.98 ± 0.11	81.02 ± 0.13	80.95 ± 0.19	80.96 ± 0.07	80.97 ± 0.12
	CUB-200	76.85 ± 1.00	$74.55 \pm 1.07^{\dagger}$	77.27 ± 1.04	77.17 ± 0.79	77.32 ± 1.14
ResNet-18	SVHN	94.65 ± 0.20	94.59 ± 0.09	94.71 ± 0.21	94.80 ± 0.14	95.27 ± 0.11
	STL-10	95.14 ± 0.48	95.08 ± 0.49	95.00 ± 0.37	94.98 ± 0.55	95.32 ± 0.30
	CIFAR-100	78.21 ± 0.25	$77.91 \pm 0.25^{\dagger}$	78.22 ± 0.33	78.27 ± 0.41	78.08 ± 0.23
	TinyImageNet	71.99 ± 0.12	$71.84 \pm 0.17^{\dagger}$	71.93 ± 0.22	71.98 ± 0.16	72.02 ± 0.16
	CUB-200	70.05 ± 1.32	$65.07 \pm 1.46^{\dagger}$	70.37 ± 0.98	69.63 ± 1.09	70.18 ± 1.27

Table 20: Test top-1 classification accuracy (%) across all 15 backbone–dataset combinations, for the same configurations as Table 19. Entries are means ± sample standard deviations over the 10 shared seeds. The narrow-penultimate entries marked † did not satisfy the validation-based accuracy criterion; among the tested widths, they have the highest mean validation accuracy and were selected for descriptive reporting without using test accuracy. Aggregate comparisons use the untreated baseline for those combinations.

Backbone	Dataset	Untreated baseline	Narrow-penultimate baseline	Nuclear-norm penalty	Top-d variance penalty	**PAFT**
ViT-S/16	SVHN	96.90 ± 0.22	96.82 ± 0.21	96.94 ± 0.21	96.89 ± 0.22	96.93 ± 0.13
	STL-10	97.16 ± 0.42	96.94 ± 0.41	96.94 ± 0.40	96.74 ± 0.23	97.08 ± 0.27
	CIFAR-100	85.21 ± 0.30	84.87 ± 0.16	85.41 ± 0.40	85.19 ± 0.28	85.23 ± 0.26
	TinyImageNet	81.04 ± 0.28	80.83 ± 0.26	81.03 ± 0.28	81.16 ± 0.16	81.03 ± 0.42
	CUB-200	77.30 ± 0.71	77.70 ± 0.53	78.18 ± 0.65	78.07 ± 0.51	77.60 ± 0.41
ResNet-50	SVHN	95.53 ± 0.17	95.48 ± 0.16	95.62 ± 0.12	95.68 ± 0.15	96.03 ± 0.10
	STL-10	97.28 ± 0.17	97.63 ± 0.08	97.24 ± 0.16	97.12 ± 0.09	97.49 ± 0.19
	CIFAR-100	83.02 ± 0.25	82.79 ± 0.21	82.90 ± 0.27	82.86 ± 0.26	83.32 ± 0.27
	TinyImageNet	80.89 ± 0.21	80.98 ± 0.20	80.91 ± 0.12	80.89 ± 0.17	80.97 ± 0.20
	CUB-200	77.76 ± 0.53	$75.19 \pm 0.59^{\dagger}$	77.18 ± 1.00	77.56 ± 0.60	78.31 ± 0.66
ResNet-18	SVHN	95.52 ± 0.12	95.47 ± 0.13	95.71 ± 0.12	95.60 ± 0.15	96.11 ± 0.11
	STL-10	94.68 ± 0.33	94.51 ± 0.21	94.50 ± 0.15	94.59 ± 0.19	94.81 ± 0.38
	CIFAR-100	78.73 ± 0.15	$78.35 \pm 0.27^{\dagger}$	78.92 ± 0.22	78.83 ± 0.19	78.82 ± 0.14
	TinyImageNet	71.64 ± 0.23	$71.66 \pm 0.15^{\dagger}$	71.62 ± 0.22	71.56 ± 0.23	71.55 ± 0.19
	CUB-200	71.12 ± 0.67	$67.66 \pm 0.82^{\dagger}$	71.37 ± 0.48	70.84 ± 0.93	71.29 ± 0.94

Table 21: Auxiliary-head test top-1 classification accuracy (%) across PAFT bottleneck widths. Entries are means ± sample standard deviations over the 10 seeds. For each backbone–dataset combination, all widths use the coefficient selected for PAFT for that combination. These accuracies describe the auxiliary classifier operating on the bottleneck, not the main classifier operating on the high-dimensional penultimate representation. No width is selected using these auxiliary-head test accuracies.

Backbone	Dataset	$d = 2$	$d = 8$	$d = 16$	$d = 32$	$d = 64$
ViT-S/16	SVHN	94.44 ± 4.93	97.04 ± 0.14	97.12 ± 0.26	96.89 ± 0.19	96.96 ± 0.13
	STL-10	50.42 ± 10.30	97.07 ± 0.24	97.05 ± 0.33	97.11 ± 0.35	97.03 ± 0.20
	CIFAR-100	11.29 ± 5.28	81.60 ± 0.72	83.97 ± 0.70	84.12 ± 0.64	84.68 ± 0.55
	TinyImageNet	2.44 ± 0.81	67.03 ± 6.15	76.97 ± 1.06	78.37 ± 1.14	79.89 ± 0.61
	CUB-200	1.52 ± 0.45	35.18 ± 10.78	52.66 ± 16.02	65.39 ± 11.99	73.16 ± 2.65
ResNet-50	SVHN	94.22 ± 0.63	95.93 ± 0.16	95.95 ± 0.15	96.04 ± 0.11	96.05 ± 0.11
	STL-10	49.71 ± 15.93	97.20 ± 0.24	97.47 ± 0.19	97.50 ± 0.21	97.49 ± 0.19
	CIFAR-100	9.36 ± 1.76	77.23 ± 1.94	81.17 ± 0.53	82.17 ± 0.45	82.69 ± 0.24
	TinyImageNet	4.57 ± 0.61	69.99 ± 1.37	76.74 ± 0.49	79.38 ± 0.26	80.36 ± 0.22
	CUB-200	2.59 ± 0.73	40.05 ± 11.02	66.41 ± 4.70	69.67 ± 3.75	74.01 ± 1.85
ResNet-18	SVHN	95.27 ± 0.25	96.13 ± 0.11	96.13 ± 0.11	96.15 ± 0.10	96.10 ± 0.14
	STL-10	54.65 ± 12.26	93.57 ± 3.27	94.67 ± 0.47	95.03 ± 0.49	95.01 ± 0.30
	CIFAR-100	13.41 ± 1.80	74.31 ± 0.67	76.77 ± 0.27	77.95 ± 0.27	78.27 ± 0.48
	TinyImageNet	5.15 ± 0.32	60.94 ± 0.51	66.98 ± 0.43	69.22 ± 0.25	70.55 ± 0.30
	CUB-200	2.11 ± 0.56	28.56 ± 7.24	52.19 ± 5.62	62.22 ± 3.33	65.97 ± 2.15

back to the untreated baseline under PCA on those 3 combinations, and in the same-run $d = 2$ penultimate comparison both arms use that fallback, so those combinations contribute unchanged comparisons rather than evidence. Restricting the analysis to the 12 combinations whose $d = 2$ configuration passes the gate gives $+19.00$ against the untreated baseline, $+16.50$ against the narrow-penultimate baseline, $+16.92$ against the nuclear-norm penalty, $+16.33$ against the top-d variance penalty, $+13.42$ against the validation-selected PAFT representation and $+6.92$ against PCA of the same run's penultimate representation, all out of 20; every aggregate remains positive. The aggregate result does not hold for every measure family: in the same-run comparison, the global-geometry mean win–loss difference is -0.93 out of five, or -0.19 normalized by the number of measures in that family, while the overall difference remains positive.