Width Confounding in Label-Free Cross-Architecture Model Selection: Diagnosis and Subspace Correction

Qin Liu **Chen Zhong** **Fengshan Zhao***
State Key Laboratory of Novel Software Technology
Nanjing University, Nanjing, China

Abstract

Choosing a pretrained encoder without target labels is a practical problem in transfer learning. RankMe, validated primarily within architecture families, can be confounded by feature width when architectures differ. Under an isotropic random-feature model at fixed aspect ratio, its effective rank scales as $\Theta(d)$. We propose Subspace-Corrected Ranking (SCR), which augments the target spectrum with the alignment between target and reference features extracted by the same candidate encoder. Its default member $\mathrm{SSR}^{\ddagger}$ subtracts log subspace alignment from log effective rank. Across 22 pool–task units from three heterogeneous model pools (MV/VTAB/EX), $\mathrm{SSR}^{\ddagger}$ reduces mean selection regret from 0.133 to 0.069 relative to RankMe (paired difference: -0.0645; target-name-clustered 95% CI: $[-0.122, -0.020]$). On MV, the mean score–width correlation decreases from 0.609 to 0.305, indicating that width dependence is reduced but remains present. On five MV tasks, a two-stage pipeline using $\mathrm{SSR}^{\ddagger}$ for architecture selection and RankMe for within-family selection requires fewer candidate evaluations while achieving regret similar to evaluating $\mathrm{SSR}^{\ddagger}$ over all candidates.

1 Introduction

Transfer learning starts with a choice: which pretrained encoder to transfer to the target task. When candidate encoders share the same architecture but differ in pretraining recipe, the choice can be made cheaply via a label-free score on the encoder's penultimate features. The dominant such score, *RankMe* [16], computes the effective rank $r_{\mathrm{eff}}(H)$ of the feature matrix and ranks candidates by $\log r_{\mathrm{eff}}$. In the within-architecture setting where RankMe was validated, the penultimate-layer dimension d is constant across candidates, so width alone cannot explain their ranking.

A practitioner using timm [54] or torchvision compares heterogeneous backbones, including ResNet [18], ViT [13], ConvNeXt [31], and EfficientNet [49], whose feature widths vary substantially. The within-family setting in which RankMe was validated does not establish its reliability across these architectures. On the 119-candidate multi-variant (MV) pool (Section 3), RankMe is width-associated (mean per-task $\rho_{\mathrm{Sp}} = 0.609$), with regret 0.144: better than random (0.191), but worse than our relational score $\mathrm{SSR}^{\ddagger}$ (0.098; Section 5). On the Visual Task Adaptation Benchmark (VTAB) and extended (EX) pools, however, RankMe has higher mean regret than random. Taken together, these results show that width association alone does not determine whether selection is better or worse than random.

The mechanism is a consequence of RankMe's target-only spectral form under feature isotropy. A Marchenko–Pastur argument shows that RankMe effective rank scales as $\Theta(d)$ at fixed positive

*Corresponding author: Fengshan Zhao, fs.zhao@nju.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

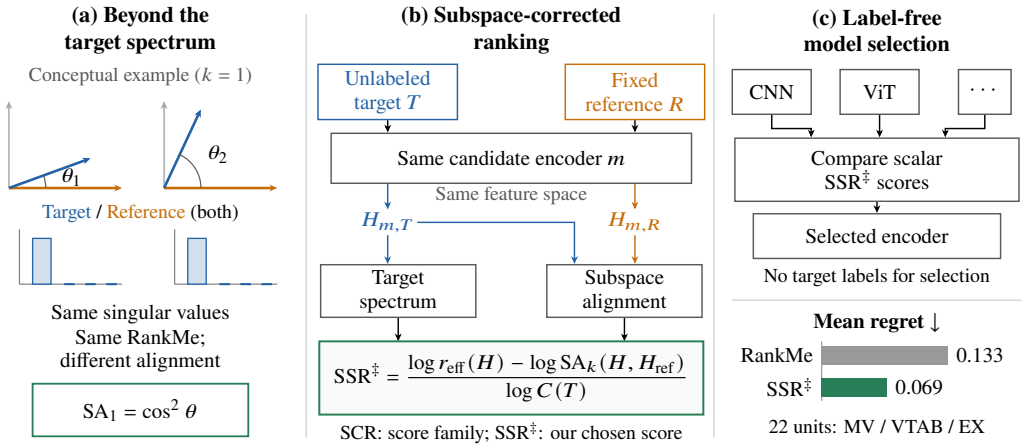

Figure 1: Overview of subspace correction. (a) Equal target spectra can have different reference-relative angles; this is not an accuracy ordering. (b) Alignment uses both feature matrices from the same candidate, with $H = H_{m,T}$ and $H_{\text{ref}} = H_{m,R}$. Class-count metadata $C(T) > 1$ only rescales within-task scores; omitting it leaves ranking unchanged. (c) Scalar scores enable zero-target-label selection. Bars summarize 22 common-support units (Table 9); target labels are used only for evaluation.

aspect ratio, so $\log r_{\text{eff}}$ equals $\log d$ plus an aspect-ratio-dependent constant asymptotically. This formal statement is specific to RankMe; rotation invariance alone is insufficient to conclude that an arbitrary score is monotone in d.

Figure 1 summarizes our approach. Given unlabeled target images and a fixed, target-disjoint reference dataset, each candidate encoder separately maps both datasets into its own feature space. SCR augments the target spectrum with the relative geometry of these two subspaces; its default member $\text{SSR}^{\ddagger}$ subtracts log alignment from log effective rank. Only the resulting scalar scores are compared across encoders, so their feature dimensions need not match. This provides a label-free way to shortlist pretrained backbones before labeled downstream evaluation. Across 22 common-support pool–task units from MV, VTAB, and EX, $\text{SSR}^{\ddagger}$ reduces mean selection regret from 0.133 for RankMe to 0.069, showing that target–reference subspace geometry can improve upon target-spectrum-only selection on these pools.

Contributions. **(i) Diagnosis.** We identify feature-width confounding in RankMe across heterogeneous architectures and explain its effective-rank scaling under an isotropic random-feature model. **(ii) Subspace correction.** We introduce Subspace-Corrected Ranking (SCR), a family of label-free relational scores that complements the target spectrum with within-encoder target–reference geometry. Its default member, $\text{SSR}^{\ddagger}$, reduces mean selection regret relative to RankMe on the evaluated common support. **(iii) Budgeted selection.** We evaluate a two-stage pipeline that uses $\text{SSR}^{\ddagger}$ for architecture selection and RankMe for within-family variant selection, which on five MV tasks uses fewer candidate evaluations at regret similar to flat $\text{SSR}^{\ddagger}$.

2 Related Work

Transferability scores. RankMe [16] ranks pretrained encoders by the effective rank of penultimate features, using $p_i = \sigma_i / \sum_j \sigma_j$; our focus is the heterogeneous-pool setting beyond its original within-family validation. VNE [25] uses the entropy of normalized squared singular values, but we do not extend our RankMe-specific theorem to VNE. Official classification DiSCO [60] uses target ground-truth labels to construct class-conditional quantities and is therefore reported as a label-dependent comparator, not a zero-label baseline. Other label-requiring scores include LogME [58], H-score [4], NCTI [53], LEEP [38], N-LEEP [29], PACTran [12], OTCE [50], GBC [42], TransRate [21], NCE [51], and Task2Vec [2]. Transfer learning more broadly is surveyed in Pan and Yang [41].

Subspace alignment and random matrix theory. Subspace alignment [15] was introduced for unsupervised domain adaptation via principal-angle geometry on the Grassmann manifold [17]. Neural network representation similarity via CKA and SVCCA [26, 45] uses related geometric ideas to compare layers; we use principal-angle geometry differently, as a label-free measure of generic-vs-specific feature structure relative to a fixed reference dataset encoded by each candidate. The Marchenko–Pastur law [36] has been applied to characterize weight matrices in deep networks [35]; we apply it directly to feature matrices to derive the width-confound mechanism.

Two-stage and budgeted model selection. Staged screening followed by precise evaluation is a standard design in neural architecture search and Bayesian hyperparameter optimization. In source-free transferability estimation [53, 58], the same logic supports a cross-architecture stage at low cost followed by a within-family stage; we evaluate this design and a mixed-budget extension as five-task proofs of concept.

3 Setup

Pools. Table 1 summarizes the four pools (Appendices B–E). MV uses timm/torchvision [54]; its 119 variants in 35 multi-variant groups form the headline and two-stage support. VTAB [59] has 24 complete architectures; excluding the STL-10 reference leaves seven targets. SSL has eight encoders, including CLIP [44], DINOv2 [40], DINO [7], and MoCo [19], on 29 tasks with incomplete support. EX has 51 supervised timm variants from 35 architectures on ten tasks. The inventories cover 52 task units and $d \in [192, 4096]$.

Table 1: Four architecturally distinct evaluation pools used throughout the paper. Width ranges describe the candidate pools; they are not a validated score-selection threshold.

Pool	Archs	Variants	Tasks	d range	$d_{\max}/d_{\min}$
MV pool (multi-variant)	35 / 48 full	119 (multi-var.) / 132 (full)	5	$[192, 4096]$	$21.3\times$
VTAB pool	24	24	8	$[192, 2048]$	$10.7\times$
SSL/multimodal pool	8	8	29	$[384, 2048]$	$5.3\times$
EX pool (supv.)	35	51	10	$[192, 4096]$	$21.3\times$

Scores. Effective rank r_{eff} follows RankMe [16]: $p_i = \sigma_i / \sum_j \sigma_j$ and $r_{\text{eff}} = \exp(-\sum_i p_i \log p_i)$. For each candidate encoder m in the main 22-unit comparison, SA_k compares its target features with its STL-10 features; both are produced by encoder m and therefore share dimension d_m. We use uncentered feature matrices, default $k = 20$, and clip quantities to at least 10^{-12} before taking logarithms. Different candidates' subspaces are never compared directly. Official classification DiSCO and the other label-requiring methods are reported separately from the zero-label RankMe–$\text{SSR}^{\ddagger}$ comparison.

Linear-probe ground truth. We use held-out accuracy of ℓ_2-regularized logistic regression ($C = 1$) on an $80/20$ split. Regret is the gap to the oracle over the same candidate set. The 22-unit paired and target-name-clustered bootstrap and the five-task pipeline intervals each use 100,000 resamples with seed 42.

Feature extraction and preprocessing. For torchvision VGG16/VGG16-BN, we retain the hidden fully connected layers to obtain 4096-dimensional features and apply the official weights transform once to target and STL-10 images. The other 23 torchvision variants in the 119-candidate support retain cached features associated with an extractor that applies loader resizing, cropping, and normalization before the weights transform, including a second normalization. These include the SVHN oracle, DenseNet-169. Individual historical cache provenance is incomplete. A VGG control retaining this two-transform pipeline gives identical 22-unit RankMe/$\text{SSR}^{\ddagger}$ regrets, but does not validate preprocessing for the remaining candidates. The benchmark is not a uniformly preprocessed comparison; its oracles and uncertainty are conditional on the retained features (Appendices B and G).

4 The Width Confound

We isolate the failure mechanism with two formal statements. Both proofs are in Appendix A.

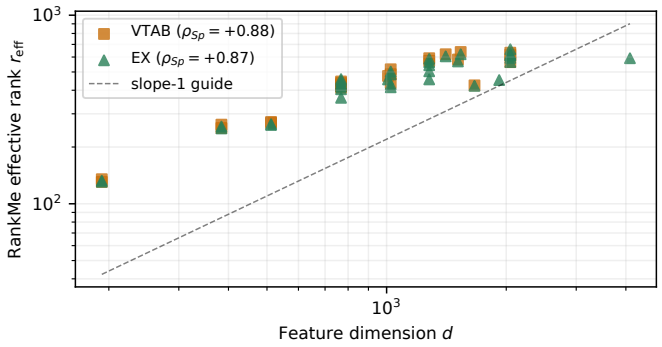

Figure 2: Effective rank versus feature width for the 24 VTAB and 35 EX architecture groups. The dashed slope-1 line is a visual guide, not a fit. VTAB includes its eight cached datasets for this spectral diagnostic only; STL-10 is excluded from all target-selection comparisons.

Lemma 1 (Marchenko–Pastur scaling)**.** *Let $H \in \mathbb{R}^{n\times d}$ have i.i.d. rows from $\mathcal{N}(0, \sigma^2 I_d)$, with $\sigma > 0$. As $n, d \to \infty$ with $\gamma = n/d \in (0, \infty)$ fixed, $r_{\mathrm{eff}}(H)/d \to c(\gamma) > 0$ almost surely, where $c(\gamma)$ depends only on the aspect ratio. In particular $r_{\mathrm{eff}}(H) = \Theta(d)$.*

Why width enters. Let $\lambda_i = \sigma_i^2/(n\sigma^2)$ include all d eigenvalues, and set $S_d = d^{-1}\sum_i \sqrt{\lambda_i}$. RankMe's singular-value normalization gives the exact identity

$$\log r_{\mathrm{eff}}(H) = \log d + \log S_d - \frac{1}{2S_d d}\sum_i \sqrt{\lambda_i}\log\lambda_i.$$

At fixed aspect ratio, the last two terms converge to a finite constant under the stated Gaussian model. The continuous extension $\sqrt{x}\log x \to 0$ as $x \downarrow 0$ handles zero eigenvalues; no spectral gap from zero is assumed. Thus width contributes the explicit $\log d$ term. This is an explanatory model, not an assertion that every pretrained representation is isotropic; equal classwise total variances do not establish directional isotropy. Appendix A gives the full derivation.

Proposition 2 ($\mathrm{SSR}^{\ddagger}$ is relational and jointly basis-invariant)**.** *For candidate encoder m, take the retained right-singular subspaces to be uniquely specified (or transport a chosen subspace consistently through a tied singular-value block). For $Q \in O(d_m)$, jointly replacing $(H_{m,T}, H_{m,R})$ by $(H_{m,T}Q, H_{m,R}Q)$ leaves their principal angles, SA_k, and $\mathrm{SSR}^{\ddagger}$ unchanged. In general, $\mathrm{SSR}^{\ddagger}$ is not determined by the singular values of $H_{m,T}$ alone: it additionally depends on the relative geometry of the target and reference right-singular subspaces.*

Why the correction adds information. Under the common basis change, $V_T' = Q^\top V_T$ and $V_R' = Q^\top V_R$, so $(V_T')^\top V_R' = V_T^\top V_R$. By contrast, Figure 1(a) holds a rank-one reference fixed while rotating a rank-one target without changing its singular value. With one retained direction ($k = 1$), alignment is $\cos^2\theta$: RankMe cannot distinguish the two targets, whereas $\mathrm{SSR}^{\ddagger}$ can. This illustrates information beyond the target spectrum, not an ordering of their downstream accuracies.

Finite-pool illustration. Figure 2 uses RankMe's singular-value normalization consistently with the headline scores. After averaging within architecture groups, effective rank and width have Spearman correlations $+0.877$ on VTAB and $+0.871$ on EX. These finite-pool associations are distinct from per-task correlations and do not constitute a fitted asymptotic law.

Empirical association across pools. Mean per-task $\rho_{\mathrm{Sp}}(\mathrm{RankMe}, d)$ is 0.609 on MV, 0.879 on VTAB, and 0.766 on EX; width–LP correlations are -0.049, $+0.233$, and -0.070. Thus width association is not equivalent to quality prediction. MV RankMe is better than random, whereas VTAB and EX have higher mean regret than their respective random baselines (Table 2). On the same 119-candidate MV support, $\mathrm{SSR}^{\ddagger}$ reduces mean score–width correlation to 0.305 (task range 0.156–0.495) and increases mean score–LP correlation from 0.352 to 0.512. This is descriptive evidence of reduced, not eliminated, width dependence; it does not isolate a causal effect of width. On EX, mean Kendall correlation between RankMe and d is $\tau = +0.618$ across 10 tasks.

Table 2: Width and quality diagnostics by pool. MV uses 119 candidates; VTAB excludes STL-10. Correlations are per-task Spearman means.

Pool	Groups/tasks	$\rho(\mathrm{RankMe}, d)$	$\rho(d, \mathrm{LP})$	RM regret	Random regret
MV	35/5	0.609	−0.049	0.144	0.191
VTAB	24/7	0.879	0.233	0.110	0.106
EX	35/10	0.766	−0.070	0.144	0.131

VTAB per-task breakdown. Table 3 uses the same 24 candidates for every non-reference target. The RankMe–width correlation ranges from 0.838 to 0.915, while regret varies substantially by task.

Table 3: VTAB per-task width confound on the complete 24-architecture support. STL-10 is excluded because it is the reference dataset.

Task	$\rho_{\mathrm{Sp}}(\mathrm{RankMe}, d)$	Regret	RankMe pick
CIFAR-10	+0.865	+0.050	efficientnet_b3 (d=1536)
CIFAR-100	+0.869	+0.235	efficientnet_b3 (d=1536)
DTD	+0.877	+0.095	resnet101 (d=2048)
Fashion-MNIST	+0.886	+0.045	wide_resnet50_2 (d=2048)
Flowers-102	+0.838	+0.165	efficientnet_b3 (d=1536)
Pets	+0.902	+0.070	mobilenetv3_large (d=1280)
SVHN	+0.915	+0.110	efficientnet_b3 (d=1536)
Mean	**+0.879**	**+0.110**	—

VGG sensitivity and task variation. Removing the three VGG variants ($3/119 \approx 2.5\%$) leaves RankMe and $\mathrm{SSR}^{\ddagger}$ regrets unchanged at 0.144 and 0.098. RankMe selects a ResNeXt variant on four tasks and Inception on SVHN. $\mathrm{SSR}^{\ddagger}$ improves over RankMe on four tasks, but is worse on SVHN; it is within 0.07 LP of the oracle on two tasks (Table 4).

Table 4: Per-task LP on 119 MV candidates. Random is the exact expectation. $\mathrm{SSR}^{\ddagger}$ improves over RankMe on four tasks, but is worse on SVHN. EuroSAT also has a tied ViT-large oracle (LP 0.955).

Task	Oracle	Random	RM	$\mathrm{SSR}^{\ddagger}$	RM group	$\mathrm{SSR}^{\ddagger}$ group	Oracle group
CIFAR-10	0.970	0.787	0.845	0.870	resnext50	resnet50	convnext_base
CIFAR-100	0.825	0.486	0.535	0.670	resnext50	resnet152	convnext_base
DTD	0.740	0.564	0.625	0.720	resnext50	convnext_base	convnext_base
EuroSAT	0.955	0.882	0.880	0.905	resnext50	efficientnet_b3	densenet201 (tie)
SVHN	0.615	0.428	0.500	0.450	inception_v3	efficientnet_b3	densenet169

Spectral truncation diagnostic. One CIFAR-10 cell lacks truncated and auxiliary scores. For these comparisons only, we remove that variant uniformly, giving 118 candidates on all five tasks. On this complete support, truncation at $D = 128/256/512/768$ gives regrets $0.134/0.134/0.151/0.142$, versus full-spectrum RankMe 0.144 and $\mathrm{SSR}^{\ddagger}$ 0.098. Some cutoffs improve RankMe, but all four have higher regret than $\mathrm{SSR}^{\ddagger}$. The cutoff is an additional design choice (Appendix E).

Auxiliary comparisons. Table 5 reports regret and Table 6 mean LP on 118 candidates, or 115 without VGG. Removing VGG lowers LogME accuracy, while H-score is unchanged. Direct dimension normalization, $\log r_{\mathrm{eff}} - \log d$, gives regret 0.200, versus random 0.192 and $\mathrm{SSR}^{\ddagger}$ 0.098 on the same 118 candidates; subtracting $\log d$ is insufficient here. The LEEP column is a clustering proxy, not an official LEEP implementation. These diagnostics do not extend our RankMe-specific theorem.

5 Subspace-Corrected Ranking

Geometric premise. For candidate encoder m, write $H = H_{m,T}$ and $H_{\mathrm{ref}} = H_{m,R}$ for its target and fixed-reference feature matrices. Let $V_{m,\mathrm{ref},1:k}$ contain the top-k right singular vectors of H_{ref}, and write $P = V_{m,\mathrm{ref},1:k} V_{m,\mathrm{ref},1:k}^{\top}$, $Q = I - P$. Let Σ_B be the class-frequency-weighted covariance

Table 5: Complete auxiliary support: 118 candidates after excluding one variant with a missing score; 115 after also removing three VGG variants. Entries are five-task mean regret.

Method	With VGG	Without VGG	Δ	Type
Random	0.192	0.191	−0.001	reference
RankMe	0.144	0.144	0.000	zero-label
RankMe ($-\log d$)	0.200	0.200	0.000	zero-label
LogME	0.219	0.550	0.331	labels
H-score	0.470	0.470	0.000	labels
NCTI	0.155	0.155	0.000	labels
SSR‡	0.098	0.098	0.000	zero-label
LEEP clustering proxy	0.038	0.038	0.000	labels

Table 6: Complete auxiliary support: 118 candidates after excluding one variant with a missing score; 115 after also removing three VGG variants. Entries are five-task mean LP.

Method	With VGG	Without VGG	Δ	Type
Random	0.629	0.630	0.001	reference
RankMe	0.677	0.677	0.000	zero-label
RankMe ($-\log d$)	0.621	0.621	0.000	zero-label
LogME	0.602	0.271	−0.331	labels
H-score	0.351	0.351	0.000	labels
NCTI	0.666	0.666	0.000	labels
SSR‡	0.723	0.723	0.000	zero-label
LEEP clustering proxy	0.783	0.783	0.000	labels

of target class means, computed with target labels only for this diagnostic. Define

$$f_{\mathrm{spe}}(H, H_{\mathrm{ref}}) := \frac{\mathrm{tr}(Q\Sigma_B Q)}{\mathrm{tr}(\Sigma_B)},$$

the fraction of between-class covariance lying outside the same candidate's reference-data subspace. This residual fraction does not, by itself, establish architecture-specific information or compare different encoders. On the complete VTAB support ($24\times7 = 168$ architecture–task pairs), the subspace alignment $\mathrm{SA}_k(H, H_{\mathrm{ref}})$ is nearly perfectly anti-correlated with f_{spe} ($\rho_{\mathrm{Sp}} = -0.959$, Figure 3). Within each task, f_{spe} is positively associated with LP accuracy (mean Spearman $\rho = +0.420$). Subtracting subspace alignment is therefore an empirically motivated, label-free penalty for overlap with the reference-data subspace; this observation is mechanistic evidence rather than a downstream guarantee. We exclude `resnet152` uniformly because reference and target features are available for only one of the eight VTAB datasets; all 168 pairs on the resulting 24-architecture support are computed successfully (see `results/camera_ready/vtab_fspe_24arch_uncentered.csv`).

Alignment and algorithm. For candidate m, let the uncentered feature matrices be $H_{m,T} \in \mathbb{R}^{n_T \times d_m}$ and $H_{m,R} \in \mathbb{R}^{n_R \times d_m}$, both produced by encoder m. Let $V_T, V_R \in \mathbb{R}^{d_m \times k_m}$ contain their leading right-singular vectors, where $k_m = \min\{k, n_T, n_R, d_m\}$ and $k = 20$ by default. With principal angles $\theta_1, \ldots, \theta_{k_m}$,

$$\mathrm{SA}_k(H_{m,T}, H_{m,R}) = \frac{1}{k_m}\|V_T^\top V_R\|_F^2 = \frac{1}{k_m}\sum_{i=1}^{k_m}\cos^2\theta_i.$$

Algorithm 1 gives the complete computation. Candidate feature dimensions may differ, because no cross-candidate subspaces are compared.

Method family. Subspace-Corrected Ranking is the family

$$\mathrm{SCR}(H, T; \beta, g) := \frac{\log r_{\mathrm{eff}}(H) + \beta \cdot g\big(\mathrm{SA}_k(H, H_{\mathrm{ref}})\big)}{\log C(T)},$$

where g is a monotonically decreasing function and $\beta \geq 0$. Choosing $g(s) = -\log s$ with $\beta = 1$ gives

$$\mathrm{SSR}^{\ddagger}(H, T) := \frac{\log r_{\mathrm{eff}}(H) - \log \mathrm{SA}_k(H, H_{\mathrm{ref}})}{\log C(T)},$$

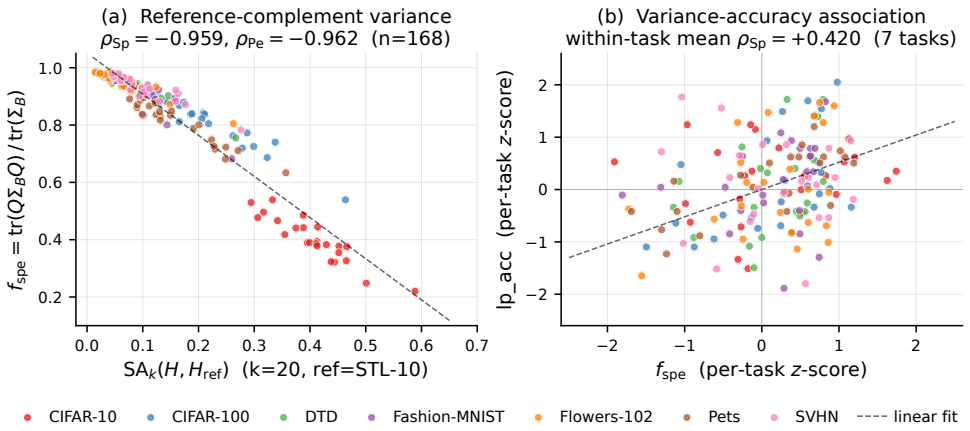

Figure 3: (a) Alignment between target and STL-10 features from the same candidate is nearly perfectly anti-correlated with the fraction of class-mean variance outside that candidate's reference-data subspace, f_{spe}. (b) Within-task, f_{spe} is positively associated with linear-probe accuracy (mean Spearman $\rho = +0.420$ over seven tasks).

Algorithm 1 Zero-label $\mathrm{SSR}^{\ddagger}$ scoring for one candidate

Require: Encoder m; nonempty unlabeled target T and target-disjoint reference R with nonzero features; $k = 20$; $\epsilon = 10^{-12}$

1: Extract $H_{m,T}$ and $H_{m,R}$ with the same encoder and its fixed preprocessing; cache reference features.
2: Compute uncentered thin SVDs of both matrices, retaining target singular values $\{\sigma_i\}$.
3: Set $k_m = \min\{k, n_T, n_R, d_m\}$ and retain right-singular bases V_T, V_R with k_m columns.
4: $s \leftarrow \|V_T^\top V_R\|_F^2 / k_m$.
5: $p_i \leftarrow \sigma_i / \sum_j \sigma_j$; $r \leftarrow \exp(-\sum_{i:p_i>0} p_i \log p_i)$.
6: $a \leftarrow \log\max(r, \epsilon) - \log\max(s, \epsilon)$.
7: **return** $a / \log C(T)$ if class-count metadata $C(T) > 1$ is provided; otherwise return a for identical per-task ranking.

and $g(s) = 1 - s$ gives an additive variant. For per-task ranking, the common positive factor $1/\log C(T)$ may be omitted without changing the selected model. Both forms use target–reference relational geometry and are jointly basis-invariant as in Proposition 2. We adopt $\mathrm{SSR}^{\ddagger}$ as the default because $g(s) = -\log s$ and $\beta = 1$ introduce no tuned correction weight.

Implementation choices. We fix the correction weight at $\beta = 1$; k, reference data, and preprocessing remain design choices. We use $k = 20$ without target-label tuning, although sensitivity to k under this evaluation protocol remains untested (Appendix D). Main scores use uncentered SVD, with alignment clipped at 10^{-12} to keep the logarithm finite. This numerical safeguard does not ensure statistical stability. Ties at the truncation boundary follow the consistent subspace convention in Proposition 2.

Choice of reference dataset. We use STL-10 because it is small, diverse, supported by all candidates, and disjoint from the targets in the 22 evaluated pool–task units. We recommend one generic, target-disjoint reference chosen before inspecting target labels or accuracy. Other domains require modality-compatible, diverse data and enough samples to estimate the subspace. On the 24-architecture VTAB support, single-reference regret is 0.061; a leave-one-target-out median over four predetermined references (protocol below) gives 0.055. Given the limited seven-task evidence and additional reference-side computation, we retain a single reference as the default. This is a heuristic choice.

Target-disjoint reference protocol. The ensemble ablation uses the predetermined set STL-10, CIFAR-10, Flowers-102, and Pets. For each evaluated target, we first remove the matching dataset from this set, then compute each remaining alignment using the same candidate encoder on target

and reference data. We take the median of these alignment values before applying the logarithmic correction; we do not select a reference by target accuracy or average scores from different encoders. Each candidate's reference features can be cached and reused across targets with unchanged preprocessing. Target labels enter only the post-hoc LP evaluation and between-class-variance diagnostic, not this selection rule. This procedure excludes direct target/reference dataset overlap; sensitivity to the reference domain and sample composition remains an open question (Appendix D).

Cross-pool evaluation. We use common support across the zero-label methods: MV (5 tasks), VTAB (7 non-reference tasks), and EX (10 tasks), for 22 pool–task units in total. STL-10 is excluded from every method because it is the reference for $\mathrm{SSR}^{\ddagger}$. On this support (Table 9), $\mathrm{SSR}^{\ddagger}$ has mean regret 0.069, versus 0.133 for RankMe and 0.137 for uniform random selection. The paired $\mathrm{SSR}^{\ddagger}$-minus-RankMe difference is -0.0645 (unit bootstrap 95% CI $[-0.107, -0.028]$, $p = 0.00006$). Official classification DiSCO is reported separately because it uses target labels.

Dependence-aware uncertainty. The 22 units contain 10 unique target names, eight repeated across pools. Resampling target-name clusters and retaining their pool units gives 95% CI $[-0.122, -0.020]$ and two-sided bootstrap sign $p = 0.00194$. Paired Wilcoxon gives $p = 0.00328$ (differences rounded to 10 decimal places; zero method `wilcox`, normal approximation, no continuity correction; SciPy 1.15.3). The advantage remains under this repeated-target analysis.

Interpreting the aggregate. Within each pool, methods share the same candidate set and per-task oracle; candidate sets differ across pools. The target-name bootstrap keeps all occurrences of a resampled target together to account for repeated targets. Its intervals are conditional on the retained feature caches and LP protocol, and do not include variation from new pretraining runs, reference choices, or train/test splits. We therefore report per-pool results alongside the aggregate, without extrapolating to unseen tasks or model zoos.

Per-pool evidence and its limits. Table 8 disaggregates the same 22 units; its aggregate weights pool means by task count, not equally by pool. The separate SSL diagnostics cover only 209 of 232 possible model–task cells, and the historical SSR reference manifest is unavailable. SSL is excluded from the headline target-disjoint comparison.

Label-dependent comparison. DiSCO uses target ground-truth labels to form class-conditional covariances, so we report it separately from the zero-label methods. Table 7 uses the official PCA-128 protocol with $G = 6$ and $G = 8$ consecutive, near-equal spectral groups (Appendix C). DiSCO has lower regret than $\mathrm{SSR}^{\ddagger}$ on EX, while their ordering on VTAB depends on G. Its SSL values, $0.045/0.042$, use a separate diagnostic support and are excluded from the headline aggregate. These comparisons describe performance under different label budgets.

Table 7: Official PCA-128 classification DiSCO on each pool; MV uses the same 119 candidates as the headline comparison. G is the number of consecutive, near-equal spectral groups. DiSCO uses target labels and is therefore not included in the zero-label comparison.

Pool	Tasks	G=6 regret	G=8 regret
MV	5	0.145	0.145
VTAB (STL-10 excluded)	7	0.067	0.040
EX	10	0.045	0.049
SSL	29	0.045	0.042

Table 8: Per-pool mean regret on the headline common support (lower is better). These are the same observations as Table 9, not additional experiments. STL-10 is excluded from all methods.

Pool	Tasks	RankMe	$\mathrm{SSR}^{\ddagger}$	Difference
MV	5	0.1440	0.0980	−0.0460
VTAB	7	0.1100	0.0614	−0.0486
EX	10	0.1445	0.0595	−0.0850

Within-family ranking. Across 173 nonconstant architecture–task groups in MV, mean within-group Spearman correlations are 0.527 for RankMe and 0.519 for $\mathrm{SSR}^{\ddagger}$. The scores are close on

Table 9: Zero-label cross-pool comparison on 22 common-support pool–task units. DiSCO is excluded because official classification DiSCO uses target labels.

Method	Mean regret	Label budget	Role
Random	$+0.137$	zero	reference
Flat-RankMe	$+0.133$	zero	target spectrum
Flat-$\mathrm{SSR}^{\ddagger}$	$\mathbf{+0.069}$	zero	target–reference geometry
Paired $\mathrm{SSR}^{\ddagger}$ vs. RankMe: $\Delta = -0.0645$; target-clustered 95% CI $[-0.122, -0.020]$, $p = 0.00194$.			

this descriptive measure, giving little basis for preferring one within families. Appendix F reports group-level and stage-specific diagnostics.

6 Two-Stage and Mixed-Budget Pipelines

Algorithm 2 Two-stage source-free model transferability estimation (SF-MTE) pipeline

1: **Input:** candidate pool $\{(a, V_a)\}$; target features H_T; reference H_{ref}.
2: **Stage 1 (architecture).** For each architecture a, compute stage-1 score s_1 on a canonical representative $\bar{v}_a \in V_a$. Set $a^* = \arg\max_a s_1(\bar{v}_a, T)$.
3: **Stage 2 (variant).** For $v \in V_{a^*}$, compute stage-2 score s_2. Return $v^* = \arg\max_v s_2(v, T)$.

Cost and quality. Algorithm 2 selects an architecture using canonical representatives, then ranks its variants. Table 10 compares strategies S1–S8. Flat scoring evaluates 119 candidates per MV task; $\mathrm{SSR}^{\ddagger}$ →RankMe (S5) averages 38 evaluations (35 representatives plus 3 variants), or 37 with reuse. Its regret is 0.102, versus 0.098 for flat $\mathrm{SSR}^{\ddagger}$. Five tasks support a proof of concept, not a general cost or quality guarantee.

Table 10: Five-task MV pipelines on 119 candidates. CIs use 100,000 task bootstrap resamples. Counts are without/with canonical reuse, excluding reference setup. K counts labeled architecture evaluations.

Strategy	Mean LP	Regret	95% CI	Evaluations
S1 Random	0.630	0.191	$[0.118, 0.276]$	$1/1$
S2 Flat-RankMe	0.677	0.144	$[0.093, 0.220]$	$119/119$
S3 Flat-$\mathrm{SSR}^{\ddagger}$	0.723	0.098	$[0.048, 0.148]$	$119/119$
S4 RankMe→RankMe	0.677	0.144	$[0.093, 0.220]$	$38/37$
S5 $\mathrm{SSR}^{\ddagger}$ →RankMe	0.719	0.102	$[0.044, 0.153]$	$38/37$
S6 $\mathrm{SSR}^{\ddagger}$ →$\mathrm{SSR}^{\ddagger}$	0.720	0.101	$[0.051, 0.151]$	$38/37$
S7 R_{UID} →RankMe	0.763	0.058	$[0.005, 0.111]$	$37.4/36.4$
S8 H-score→RankMe (labels; $K = 35$)	0.803	0.018	$[0.000, 0.052]$	$37/36$

Stage-specific diagnostics. On the same 119 candidates, $\mathrm{SSR}^{\ddagger}$ →$\mathrm{SSR}^{\ddagger}$ gives regret 0.101, close to $\mathrm{SSR}^{\ddagger}$ →RankMe (0.102). We use RankMe at stage 2 because it requires only target features, avoiding reference extraction for newly evaluated variants. A zero-label alternative, R_{UID} →RankMe (S7), gives 0.058 with $37.4/36.4$ evaluations without/with reuse. Here UID is a TwoNN intrinsic-dimension estimate [14], and $R_{\mathrm{UID}} = \log r_{\mathrm{eff}}/(\mathrm{UID} \log C)$ divides log effective rank by this estimate and $\log C$; it does not rank candidates directly by UID (Appendix F). All 35 canonical scores are available. This alternative outperforms our pipeline in the post-hoc five-task comparison.

Mixed-budget variant. We replace stage 1 with H-score on K randomly sampled architecture representatives and retain RankMe at stage 2. Here K counts label-dependent architecture evaluations, not labeled samples. Over 100 repeated subsets per budget on the same five tasks, mean regret is 0.089 at $K = 5$ and 0.018 at $K = 35$, with 8.324 and 37 mean candidate evaluations without reuse. Figure 4 compares strategy regret against evaluation count (left) and the mixed-budget curve against zero-label S5 and S7 (right). This remains an exploratory information–compute trade-off.

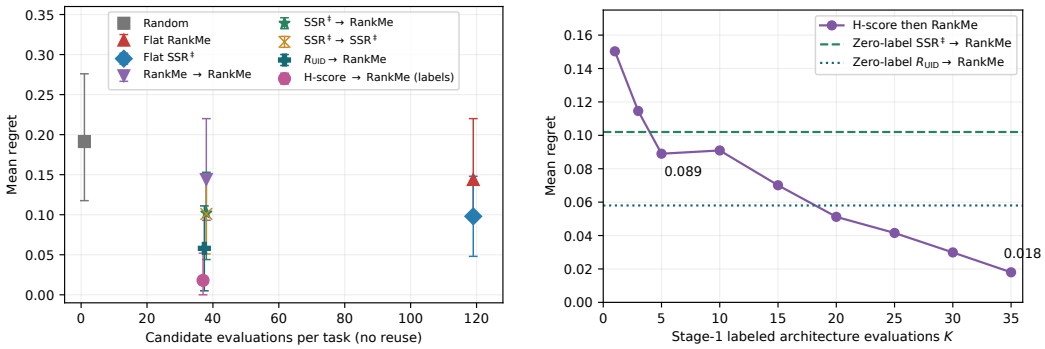

Figure 4: Five-task MV proof of concept. (left) Regret versus candidate evaluations without reuse, with task-bootstrap intervals. (right) Mean regret versus the number K of label-dependent architecture evaluations, not labeled samples.

7 Discussion

Applicability. The strongest evidence for SCR comes from selection across heterogeneous architectures. Within families, RankMe and $\mathrm{SSR}^{\ddagger}$ have similar mean correlations with accuracy. Our two-stage design uses RankMe to avoid extra reference extraction in the second stage, rather than assuming it is the better within-family score.

What the correction captures. Alignment measures relative target–reference geometry within each candidate's feature space and preserves joint basis invariance. Low alignment alone need not indicate useful features: directions outside the reference subspace may be unrelated to the target labels, while shared directions can be discriminative. The VTAB diagnostic (Figure 3) provides empirical support for this geometric interpretation.

Compute accounting. The documented MV/EX feature-extraction launcher uses four Huawei Ascend NPUs; VGG extraction uses an NVIDIA L20 GPU. Scoring and statistical analysis use CPU routines. Candidate-evaluation counts exclude reference setup and do not measure wall-clock or energy savings. Appendix G gives the recorded environments and their coverage.

Reference caches can be reused across target tasks for the same encoder, but do not remove setup cost or transfer across encoders. With M reference datasets, the approximately M-fold cost applies to reference-side extraction and alignment, not end-to-end runtime.

Limitations. Reference selection remains heuristic, and we have no formal bound linking alignment or $\mathrm{SSR}^{\ddagger}$ to downstream accuracy. The ensemble comparison covers only seven VTAB tasks (0.061 single-reference regret, 0.055 ensemble), leaving its broader value uncertain. Preprocessing differences within MV can affect both scores and LP oracles; the two VGG controls do not resolve this for other candidates. The two-stage evidence is limited to five tasks, where a different zero-label pipeline achieves lower regret. These limitations call for matched preprocessing and broader evaluation before general deployment claims.

8 Conclusion

RankMe can confound feature width with representation quality when selecting among different architectures. Under the stated isotropic model at fixed aspect ratio, its effective rank scales as $\Theta(d)$. SCR addresses this problem by adding target–reference subspace geometry, with $\mathrm{SSR}^{\ddagger}$ as its default score. Across 22 common-support pool–task units, $\mathrm{SSR}^{\ddagger}$ reduces mean selection regret from 0.133 to 0.069 relative to RankMe (paired difference: -0.0645; target-name-clustered 95% CI: $[-0.122, -0.020]$). On MV, the mean score–width correlation decreases from 0.609 to 0.305: width dependence is reduced, though not eliminated. The central finding is that target–reference geometry improves selection beyond the target spectrum on these pools. Uniform preprocessing across the full model pool, reference design, downstream guarantees, and broader two-stage validation remain as open problems.

Acknowledgments and Disclosure of Funding

This work was supported by the "111 Center" (No. B26023) and the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118). Backed by Huawei's AI Hundred Schools Program, the experimental and computational work in this research leveraged AI compute resources alongside the Huawei Cloud AI Compute Service.

Competing interests: The authors declare no competing interests.

References

[1] Andrea Acevedo, Anna Merino, Santiago Alférez, Ángel Molina, Laura Boldú, and José Rodellar. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in Brief*, 30:105474, 2020.

[2] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. Task2Vec: Task embedding for meta-learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6429–6438, 2019. doi: 10.1109/ICCV.2019.00653.

[3] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in Brief*, 28:104863, 2020.

[4] Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *IEEE International Conference on Image Processing (ICIP)*, pages 2309–2313, 2019.

[5] Patrick Bilic, Patrick Christ, et al. The liver tumor segmentation benchmark (LiTS). *Medical Image Analysis*, 84:102680, 2023. doi: 10.1016/j.media.2022.102680.

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, volume 8694 of *Lecture Notes in Computer Science*, pages 446–461, 2014. doi: 10.1007/978-3-319-10599-4_29.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951.

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014. doi: 10.1109/CVPR.2014.461.

[9] Adam Coates, Honglak Lee, and Andrew Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, 2011.

[10] Noel Codella, Veronica Rotemberg, Philipp Tschandl, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[12] Nan Ding, Xi Chen, Tomer Levinboim, Soravit Changpinyo, and Radu Soricut. PACTran: PAC-Bayesian metrics for estimating the transferability of pretrained models to classification tasks. In *European Conference on Computer Vision (ECCV)*, volume 13694 of *Lecture Notes in Computer Science*, pages 252–268, 2022. doi: 10.1007/978-3-031-19830-4_15.

[13] 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.

[14] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7:12140, 2017.

[15] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2960–2967, 2013.

[16] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 10929–10974, 2023.

[17] Jihun Hamm and Daniel D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *International Conference on Machine Learning (ICML)*, pages 376–383, 2008.

[18] 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)*, pages 770–778, 2016.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. doi: 10.1109/CVPR42600.2020.00975.

[20] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

[21] Long-Kai Huang, Junzhou Huang, Yu Rong, Qiang Yang, and Ying Wei. Frustratingly easy transferability estimation. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 9201–9225, 2022.

[22] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

[23] Jakob Nikolas Kather, Johannes Krisam, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLOS Medicine*, 16 (1):e1002730, 2019.

[24] Daniel S. Kermany, Michael Goldbaum, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, 2018.

[25] Jaeill Kim, Suhyun Kang, Duhun Hwang, Jungwook Shin, and Wonjong Rhee. VNE: An effective method for improving deep representation by manipulating eigenvalue distribution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3799–3810, 2023. doi: 10.1109/CVPR52729.2023.00370.

[26] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529, 2019.

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

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. Ranking neural checkpoints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2662–2672, 2021. doi: 10.1109/CVPR46437.2021.00269.

[30] Ruhan Liu, Xiangning Wang, Qiang Wu, et al. DeepDRiD: Diabetic retinopathy—grading and image quality estimation challenge. *Patterns*, 3(6):100512, 2022. doi: 10.1016/j.patter.2022.100512.

[31] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022. doi: 10.1109/CVPR52688.2022.01167.

[32] Vebjorn Ljosa, Katherine L. Sokolnicki, and Anne E. Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature Methods*, 9(7):637, 2012.

[33] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[34] Makerere AI Lab. iBean: Bean disease dataset. Original dataset repository, 2020. URL https://github.com/AI-Lab-Makerere/ibean. Accessed November 4, 2026.

[35] Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

[36] Vladimir A. Marčenko and Leonid A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.

[37] 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.

[38] Cuong V. Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 7294–7305, 2020.

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

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[41] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[42] Michal Pándy, Andrea Agostinelli, Jasper Uijlings, Vittorio Ferrari, and Thomas Mensink. Transferability estimation using Bhattacharyya class separability. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9162–9172, 2022. doi: 10.1109/CVPR52688.2022.00896.

[43] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763, 2021.

[45] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, 2017.

[46] Semeion. Semeion Handwritten Digit [dataset]. UCI Machine Learning Repository, 1998. URL https://doi.org/10.24432/C5SC8V.

[47] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, 2013.

[48] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1453–1460, 2011.

[49] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, 2019.

[50] Yang Tan, Yang Li, and Shao-Lun Huang. OTCE: A transferability metric for cross-domain cross-task representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15774–15783, 2021. doi: 10.1109/CVPR46437.2021.01552.

[51] Anh T. Tran, Cuong V. Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1395–1405, 2019.

[52] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5:180161, 2018.

[53] Zijian Wang, Yadan Luo, Liang Zheng, Zi Huang, and Mahsa Baktashmotlagh. How far pre-trained models are from neural collapse on the target dataset informs their transferability. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5526–5535, 2023. doi: 10.1109/ICCV51070.2023.00511.

[54] Ross Wightman. PyTorch Image Models. Software, 2019. URL https://github.com/rwightman/pytorch-image-models.

[55] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[56] Xuanang Xu, Fugen Zhou, Bo Liu, Dongshan Fu, and Xiangzhi Bai. Efficient multiple organ localization in CT image using 3D region proposal network. *IEEE Transactions on Medical Imaging*, 38(8):1885–1898, 2019. doi: 10.1109/TMI.2019.2894854.

[57] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2—a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[58] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 12133–12143, 2021.

[59] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

[60] Tengxue Zhang, Yang Shu, Xinyang Chen, Yifei Long, Chenjuan Guo, and Bin Yang. Assessing pre-trained models for transfer learning through distribution of spectral components. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(21):22560–22568, 2025. doi: 10.1609/aaai.v39i21.34414.

Supplementary Material

Width Confounding in Label-Free Cross-Architecture Model Selection: Diagnosis and Subspace Correction

Qin Liu, Chen Zhong, Fengshan Zhao

Appendices A–G accompany the main paper. Reference numbers refer to the main paper's bibliography.

A Marchenko–Pastur and Isotropy

Proof of Lemma 1. Write $r_{\text{eff}}(H) = \exp\big(H_{\text{ent}}(\mathbf{p})\big)$, where RankMe uses $p_i = \sigma_i / \sum_j \sigma_j$. Let $\lambda_i = \sigma_i^2/(n\sigma^2)$ and let the empirical measure include all d eigenvalues, including the zero eigenvalues when $d > n$. By the Marchenko–Pastur law [36], this measure converges almost surely to μ_γ. Define $S_d = d^{-1}\sum_j \sqrt{\lambda_j}$ and $m_{1/2} = \int \sqrt{x}\,\mathrm{d}\mu_\gamma(x)$. The common scale $\sigma\sqrt{n}$ cancels from RankMe, and the exact finite-d identity is

$$H_{\text{ent}}(\mathbf{p}) = \log d + \log S_d - \frac{1}{2S_d}\frac{1}{d}\sum_i \sqrt{\lambda_i}\log\lambda_i.$$

Consequently, with $0\log 0 = 0$, the Marchenko–Pastur limit gives

$$H_{\text{ent}}(\mathbf{p}) = \log d + \log m_{1/2} - \frac{1}{2m_{1/2}}\int \sqrt{x}\log x\,\mathrm{d}\mu_\gamma(x) + o(1).$$

The required moments are finite for every fixed positive aspect ratio, including regimes with support reaching zero or an atom at zero. Hence $r_{\text{eff}}(H)/d \to c(\gamma) > 0$ and $r_{\text{eff}}(H) = \Theta(d)$. □

Proof of Proposition 2. If $H = U\Sigma V^\top$, then $HQ = U\Sigma(Q^\top V)^\top$. Thus both retained bases transform by $V_T' = Q^\top V_T$ and $V_R' = Q^\top V_R$, so $(V_T')^\top V_R' = V_T^\top Q Q^\top V_R = V_T^\top V_R$. Singular values and the clipped score are also unchanged. To see that the target spectrum alone is insufficient, take a fixed rank-one reference along e_1 and a rank-one target with the same singular value but right direction $\cos\theta\, e_1 + \sin\theta\, e_2$. For $k_m = 1$, alignment is $\cos^2\theta$, which changes the score without changing the target spectrum. A tie crossing the truncation boundary requires a consistent subspace choice; arbitrary independent SVD tie-breaking is not covered by the invariance statement. □

Classwise total-variance diagnostic. On the VTAB architecture–task pairs for which the diagnostic is available, $\eta_\Sigma = \mathrm{Var}_c[\mathrm{tr}(\Sigma_c)]/(\mathbb{E}_c[\mathrm{tr}(\Sigma_c)])^2$ has median 0.022, mean 0.035, and 93.3% of values below 0.1. This measures similarity of total variance across classes, not directional isotropy: identical anisotropic class covariances can also give $\eta_\Sigma = 0$. We therefore do not use this statistic to validate the isotropic assumption of Lemma 1; that assumption defines an explanatory model rather than a verified property of every pretrained representation.

Width-plot protocol. The VTAB series uses uncentered RankMe singular-value ranks. Architecture means use eight cached datasets for this spectral diagnostic only; STL-10 is excluded from target selection. EX uses the same RankMe scores as its selection evaluation. MV is omitted from this architecture-mean plot because widths can differ within recorded grouping keys; its correlations are reported per variant.

B MV Pool (Multi-Variant Pool)

Dataset sources. CIFAR-10/100 originate from Krizhevsky [27], DTD from Cimpoi et al. [8], EuroSAT from Helber et al. [20], and SVHN from Netzer et al. [37]. ImageNet [11] is the pretraining dataset for the supervised candidates, not a target in the headline aggregate.

Setup. The MV inventory contains 132 pretrained variants in 48 recorded groups; the main architecture-selection support uses 119 variants in 35 groups with at least two members. Models are sourced from timm [54] and torchvision, spanning VGG, ResNet, DenseNet, EfficientNet, MobileNet, RegNet, ViT, DeiT, ConvNeXt, and Swin. The five tasks are CIFAR-10, CIFAR-100, DTD, EuroSAT, and SVHN. LP ground truth uses ℓ_2-regularized logistic regression ($C = 1$) on an 80/20

split, as in VTAB. Table 11 reports actual recorded group counts rather than conflating broad backbone classes with grouping keys.

Architecture families and variant counts.

Table 11: Recorded MV architecture-group keys and variant counts on the fixed multi-variant support. There are 35 groups and 119 variants; broad classes such as ResNet contain several keys.

Group key	Variants	Group key	Variants
convnext_base	2	regnet	8
convnext_small	3	resnet101	4
convnext_tiny	3	resnet152	4
deit_base	2	resnet18	3
deit_small	2	resnet34	3
deit_tiny	2	resnet50	11
densenet121	3	resnext50	3
densenet169	3	swin_small	4
densenet201	3	swin_tiny	3
efficientnet_b0	3	tf_efficientnet	2
efficientnet_b2	3	vgg16	3
efficientnet_b3	3	vit_b32	2
efficientnet_b4	2	vit_base	9
inception_v3	3	vit_large	2
mixer_b16	2	vit_small	4
mnasnet	3	vit_tiny	2
mobilenet_v3_large	4	wide_resnet50_2	3
mobilenet_v3_small	3		

Width diagnostic. Per-task MV RankMe–width correlations range from 0.588 to 0.631 (mean 0.609). RankMe regret is 0.144, lower than random (0.1914) but higher than $\mathrm{SSR}^{\ddagger}$ (0.098). Removing VGG leaves both score-based regrets unchanged. On SVHN the oracle is DenseNet-169. EuroSAT has two tied oracle variants: DenseNet-201 and ViT-large, both with LP 0.955.

Multi-variant structure. The inventory has 132 variants in 48 groups; the headline support has 119 variants in 35 multi-variant groups. VGG16-BN is a singleton outside the headline support. Canonical representatives are the lexicographically first variant in each group. Flat evaluation uses 119 candidates. Full-132 diagnostic regrets are 0.152 for RankMe and 0.082 for $\mathrm{SSR}^{\ddagger}$, with their own oracle; they are separate from the headline comparison.

Provenance and LP sensitivity. Each torchvision VGG model has target and STL-10 caches for three extraction settings: 25088-dimensional flattened features, 4096 dimensions with two transforms, and 4096 dimensions with one official transform. Manifests record paths, timestamps, cache/weight/script hashes, LP splits, predictions, and solver versions. Independent checks verify all 36 caches and reproduce predictions for all 20 LP cells in the two 4096-dimensional settings. Those settings and the tested scikit-learn versions (1.6.1 and 1.2.2) give identical 22-unit RankMe/$\mathrm{SSR}^{\ddagger}$ regrets, though individual VGG LP accuracies can differ. This sensitivity test does not establish protocol equivalence for auxiliary metrics or untested candidates.

Version provenance and corrections. Earlier versions used flattened 25088-dimensional torchvision VGG features while describing 4096-dimensional penultimate features. The present inventory replaces the two VGG models' ten target rows; the other 650 rows are retained. The 22-unit SSR-minus-RankMe difference changes from -0.1043 to -0.0645, and MV RankMe is better than random. Earlier below-random MV claims, unsupported CNN-only/Transformer-only effects, and general within-family superiority statements are not supported. Effective-rank normalization, official label-dependent DiSCO, target-disjoint reference comparisons, and candidate supports are specified in the present protocols. Original affected cache provenance is incomplete; new manifests document replacement data rather than reconstructing that history.

C SSL/Multimodal Pool

Additional dataset sources. Besides the datasets cited in Appendices B, D, and E, the SSL inventory includes MNIST [28], USPS [22], GTSRB [48], Semeion [46], and Beans [34]. CIFAR-100-coarse uses the superclass labels of CIFAR-100 [27]. Rendered-SST2 is the image-rendered variant used by CLIP [44] of the Stanford Sentiment Treebank [47]. Beans and Semeion are credited through original dataset release records rather than an assumed standalone paper.

The eleven biomedical tasks use MedMNIST [57]: PathMNIST (colorectal histology [23]), DermaMNIST (HAM10000/ISIC [10, 52]), OCTMNIST and PneumoniaMNIST [24], RetinaMNIST (DeepDRiD [30]), BreastMNIST [3], BloodMNIST [1], TissueMNIST [32], and OrganAMNIST/OrganCMNIST/OrganSMNIST (LiTS images and organ annotations [5, 56]). The citations identify both the standardized tasks and their original image sources.

Setup. We extract penultimate-layer features from 8 SSL/multimodal encoders: ResNet-50/101, MoCo-v3-R50, ConvNeXt-B, ViT-B/16, CLIP-ViT-L/14, DINOv2-ViT-B/14, DINO-ViT-S/16. Feature dimensions $d \in \{384, 768, 1024, 2048\}$; no VGG-class architectures. Evaluation uses 29 downstream tasks covering VTAB, MedMNIST variants, and standard transfer benchmarks. LP ground truth uses the same $80/20$ logistic-regression protocol as MV pool.

Available support and scope. The retained SSL table has 209 of $8 \times 29 = 232$ possible model–task cells. There are eight candidates on 23 tasks, four on FGVC-Aircraft, five each on Flowers-102, Food-101, Oxford-Pets, and Rendered-SST2, and one on Semeion. The oracle is over available candidates; Semeion necessarily has zero selection regret. Official DiSCO scores match these same 209 LP cells. These historical diagnostics are not a complete rectangular benchmark and are excluded from the 22-unit headline comparison. The historical SSR script specifies ImageNet reference features with a per-model CIFAR-10 fallback, but a per-run reference manifest was not retained. We therefore do not treat its SSR columns as a verified target-disjoint STL-10 evaluation or use them to support the headline correction claim.

Diagnostic limitation. The incomplete SSL support and missing per-run reference manifest prevent a fully aligned RankMe/$\mathrm{SSR}^{\ddagger}$ comparison under the main target-disjoint protocol. Official DiSCO results remain descriptive on the available 209 cells and are separate from the headline zero-label comparison.

Label-dependent comparison. Official PCA-128 DiSCO gives regret 0.045 for $G = 6$ and 0.042 for $G = 8$ on the available 29 SSL tasks. Full-feature scoring gives $0.057/0.047$; these are different dimensionality settings. Official classification DiSCO uses target ground-truth labels for class-conditional covariances and is reported separately from the zero-label $\mathrm{SSR}^{\ddagger}$–RankMe comparison.

Table 7 in the main text reports these label-dependent results alongside MV, VTAB, and EX.

D VTAB Pool

Dataset sources. In addition to the shared sources in Appendix B, this pool uses Fashion-MNIST [55], Flowers-102 [39], Oxford-IIIT Pets [43], and STL-10 [9]. The names `pets` and `oxford_pets` denote the same dataset. "VTAB pool" names our retained evaluation pool, not a claim to reproduce the full original VTAB suite [59].

Setup. The final common pool contains the 24 timm-pretrained architectures with complete features and LP values on all 8 VTAB tasks (CIFAR-10/100, DTD, Fashion-MNIST, Flowers-102, Pets, STL-10, SVHN). The original manifest also listed ResNet-152, but only one of its eight feature files was available, so it is excluded uniformly rather than allowing the candidate set to vary by task. STL-10 features serve as the reference for SA_k and are excluded as a target in the common-support analysis.

Reference protocol and fixed k. The protocol uses uncentered SVD, RankMe's singular-value normalization, and excludes STL-10 as a target. With $k = 20$, the STL-10 single reference gives regret 0.061 over seven targets on the 24-architecture support. Reference sensitivity is evaluated only on target-disjoint comparisons. We use a fixed $k = 20$ without target-label tuning. Sensitivity to k has not been established under this final evaluation protocol.

Evaluation support. The inventories contain 660 MV, 510 EX, and 192 VTAB model–task cells; 168 VTAB cells correspond to non-reference targets. The 22-unit aggregate uses the multi-variant

MV support, all seven non-reference VTAB targets, and ten EX targets. The seven-task reference regrets are $0.061/0.055$ for the single reference and ensemble, respectively.

Multi-reference robustness. For a target-disjoint evaluation, we omit any reference whose dataset name equals the current target and take the median alignment across the remaining members of the predetermined set (STL-10, CIFAR-10, Flowers-102, Pets). This leave-one-target-out ensemble gives regret 0.055, versus 0.061 for STL-10 alone. Given the limited seven-task evidence and approximately M-fold reference-side cost, we retain a single reference as the default. Reference design remains an open problem.

E EX Pool and Stage-1 Comparisons

Dataset sources. FGVC-Aircraft [33] and Food-101 [6] supplement the eight target datasets credited in Appendices B and D; STL-10 is not an EX target.

EX pool (51-variant $\times$ 10-task, supervised). 51 ImageNet-1k pretrained variants from 35 canonical architectures (including VGG-16, d=4096; $d_{\max}/d_{\min}$=21.3×) on 10 target tasks spanning fine-grained recognition (FGVC-Aircraft, Flowers-102, Food-101, Oxford-Pets), texture/scene (DTD, EuroSAT), digit/fashion (SVHN, Fashion-MNIST), and general object (CIFAR-10/100); features extracted on the full dataset. Flat-$\mathrm{SSR}^{\ddagger}$ regret is 0.0595, compared with 0.1445 for RankMe. Official, target-label-dependent DiSCO obtains $+0.045$ with G=6 and $+0.049$ with G=8; it is not part of the zero-label comparison.

Table 5 gives the complete-support auxiliary comparison.

Spectral truncation. On the complete 118-candidate auxiliary support, top-256 RankMe has regret 0.134, between full RankMe (0.144) and $\mathrm{SSR}^{\ddagger}$ (0.098). The corresponding two-stage truncated-score→RankMe diagnostic has regret 0.136, versus 0.102 for $\mathrm{SSR}^{\ddagger}$ →RankMe. These are setting-specific observations; they do not establish that truncation is intrinsically unsuitable.

F Ablations and Negative Results

Information budgets. DiSCO requires target labels; $\mathrm{SSR}^{\ddagger}$ does not. Width range is not a validated selection rule. The default relational weight is one, without cross-pool tuning.

Algebraic form of the correction. On MV, $-\alpha \log \mathrm{SA}_k$ gives regrets 0.098, 0.098, and 0.093 at the tested $\alpha = 0.5, 1, 2$, respectively. These three points do not establish stability over a continuous interval. On the 118-candidate support, the additive variant $g(s) = 1 - s$, $\beta = 1$ gives regret 0.117. The implementation $(\log r_{\mathrm{eff}} - s)/\log C$ omits the task-constant $1/\log C$; its ranking and regret, but not raw scores, equal the family definition. The inverted-direction control $-\log(1 - \mathrm{SA}_k)$ gives 0.388. We keep $\alpha = 1$ without target-based tuning; these controls show sensitivity to the form of the relational term.

Per-task selection. Table 4 gives the selections and oracles. RankMe selects a ResNeXt variant on four tasks and Inception on SVHN. $\mathrm{SSR}^{\ddagger}$ wins four paired task comparisons but loses on SVHN.

UID-normalized score. $R_{\mathrm{UID}} = \log r_{\mathrm{eff}}/(\mathrm{UID} \log C)$ divides log effective rank by the TwoNN intrinsic-dimension estimate UID [14] and $\log C$, rather than ranking candidates directly by UID. On 118 complete candidates its flat regret is 0.364 (versus $\mathrm{SSR}^{\ddagger}$'s 0.098), but R_{UID} →RankMe gives 0.058 on both this support and the headline 119: all 35 canonical scores exist despite one missing noncanonical auxiliary cell. Poor flat ranking need not imply poor screening. This post-hoc five-task result outperforms the evaluated $\mathrm{SSR}^{\ddagger}$ pipelines, without establishing a general UID-based selection rule.

Narrow-pool scope. The available evidence does not identify a universally preferable score from CNN/Transformer membership alone. Claims about homogeneous sub-pools require matched candidate and task support.

Within-family ranking and task scaling. Excluding two constant-LP groups from 35×5 architecture–task groups leaves 173 with mean Spearman correlations 0.527 (RankMe) and 0.519

($\mathrm{SSR}^{\ddagger}$): descriptive, not significant superiority evidence. Common positive task scaling, including $1/\log C(T)$, preserves candidate ordering. We aggregate per-task regrets, not raw scores.

G Experimental Environment and Compute Accounting

Feature extraction used both NVIDIA GPU and Huawei Ascend NPU platforms; cached-feature analyses ran on CPUs.

NPU feature extraction. The Huawei Cloud ModelArts instance (`modelarts.bm.npu.arm.8snt9b1.d`) provides eight Ascend 910B NPUs with 32 GB device memory each, 192 vCPUs, 1536 GiB system memory, and 1200 GB SSD storage. The MV/EX DiSCO feature-extraction runs used four of these NPUs (devices 0–3), not all eight. The launcher uses PyTorch with `torch_npu`/Ascend, one model shard per active device, batch size 64, inference mode, and float32 caches. Hardware provenance is not complete for every historical cached feature file.

Cached-feature analysis. DiSCO, linear probes, reference comparisons, and bootstrap use CPU-based NumPy/SciPy/scikit-learn routines. The MV/EX DiSCO launcher uses 16 workers; caching avoids repeating encoder inference for each score.

VGG extraction environment. The two VGG models use an NVIDIA L20 (about 46 GB), batch size 64, float32 caches, PyTorch 2.5.1+cu124, torchvision 0.20.1, Python 3.11.11, NumPy 1.26.4, SciPy 1.15.3, and scikit-learn 1.6.1. The host exposes 16 CPU threads and about 123 GiB RAM; scoring is capped at eight threads. LP uses seed-0 stratified $800/200$ splits, train-only StandardScaler, $C = 1$, lbfgs, and max_iter=1000. An isolated sklearn 1.2.2 comparison changes some VGG LPs but not the headline regrets. Cache and script hashes and generation timestamps are recorded.

Reporting scope. Section 6 counts candidate–task evaluations, not runtime, energy, or hardware speedups; reference setup is additional. Historical per-run timing records remain incomplete, preventing execution-time and total-compute reporting for all experiments (Checklist: "No").

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sections 1 and 7 state the RankMe-specific width-confounding claim, the cross-family scope of SSR‡, the 22-unit result, and the principal limitations. The five-task two-stage result is explicitly described as a proof of concept rather than a general guarantee.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses heuristic reference selection, pool-dependent performance, the absence of a formal downstream-accuracy guarantee, limited within-family evidence and incomplete historical provenance, the limited five-task evidence for the two-stage pipeline, and preprocessing heterogeneity among retained MV candidates. The five-task diagnostics also include a lower-regret zero-label pipeline.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 4 states the isotropic random-feature and aspect-ratio assumptions for the RankMe result, and Appendix A supplies the Marchenko–Pastur derivation using RankMe's singular-value normalization. The paper does not extend this formal conclusion to arbitrary rotation-invariant scores.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: Section 5 gives the exact effective-rank, subspace-alignment, and $\text{SSR}^{\ddagger}$ definitions and a complete algorithm, including uncentered SVD, $k = 20$, and numerical clipping. Section 3 and Appendices B–G document supports, VGG extraction settings, fixed LP parameters, and cache/script manifests. Frozen result tables permit aggregate verification, but some original caches and historical environments are unavailable; full end-to-end historical reproducibility is not claimed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The accompanying repository at https://github.com/kenny67nju/crossarchitecturetransferability provides scoring and analysis code, final result tables, provenance records, and verification instructions. Raw VGG feature caches are not distributed, and some historical caches and environments are unavailable; complete open-data reproduction is therefore not provided.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3 describes the pools, linear-probe protocol, sampling, and per-candidate preprocessing. Section 5 specifies the reference, k, clipping, DiSCO settings, and aggregate uncertainty analysis; Section 6 specifies the two-stage evaluation. Appendices B–G provide pool-specific supports, extraction and LP settings, ablations, and the recorded compute environment. Encoders are frozen; only the evaluation linear probes are fitted. Historical reproducibility and compute-reporting gaps are disclosed in items 4 and 8.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main 22-unit comparison reports paired unit-bootstrap and target-name-clustered 95% confidence intervals and two-sided tests in Section 5; the clustered analysis uses 100,000 resamples and is corroborated by a paired Wilcoxon test. The five-task two-stage intervals and their limited evidential scope are reported in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix G documents feature extraction on both NVIDIA L20 GPUs and Huawei Ascend 910B NPUs, along with CPU-based cached-feature analyses. It specifies the ModelArts instance, four active NPUs with 32 GB device memory each, 192 vCPUs, 1536 GiB system memory, and 1200 GB SSD storage, as well as the separate VGG host configuration. Historical per-run timing records remain incomplete, so execution times and total compute cannot be reported for all experiments. Candidate-evaluation counts are not wall-clock speedups.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The work evaluates pretrained encoders on established image-classification benchmarks and introduces a model-selection score. It involves no human subjects, private data, or deployment intervention, and the authors have reviewed the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This is foundational methodology for selecting existing pretrained encoders rather than a deployed application. It may reduce unnecessary model evaluation, but the work does not introduce a direct application-specific societal impact beyond those of the underlying pretrained models and benchmark datasets.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released artifacts are scoring and analysis code plus numerical result tables; no new pretrained model, scraped dataset, generative system, or other high-risk asset is released.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: Dataset source publications or original release records are cited in Appendices B–E, including the underlying sources of derived datasets. We do not provide a complete asset-by-asset inventory of licenses and terms for all datasets, model checkpoints, and libraries, so we answer No rather than claim complete licensing documentation.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: The released scoring code, result tables, verification scripts, and feature-extraction instructions are documented in the repository, but the current release does not include a separate explicit software-license file.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or research involving human subjects was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects or study participants were involved, so IRB approval and participant-risk disclosures are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not a component of the proposed scoring method. AI assistance was used for manuscript editing, preparation of the conceptual overview figure, and code/audit support; the mathematical scores and reported evaluations are computed by the documented numerical procedures, not by LLM judgments. The authors remain responsible for verification of the text, figures, code, and results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.