PER-SAMPLE ROUTING OF PRE-TRAINED ENCODERS VIA EMBEDDING TYPICALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Practitioners typically deploy pre-trained vision encoders by choosing a single backbone and layer, fitting a probe on its embeddings, and using that representation for every test sample. This fixed choice can be suboptimal because the best encoder and layer can vary across inputs in heterogeneous tasks. We study unsupervised per-sample routing over a pool of frozen pre-trained encoders, treating each (backbone, layer) pair as a candidate. Using only embedding geometry, the router either assigns a test sample to one candidate or combines predictions across the pool. It requires neither supervised router training, test labels, nor per-candidate calibration. For hard routing, we repurpose the existing Levy radial-typicality score: it reads typicality from the embedding norm and selects one candidate per sample. For soft ensembling, we introduce the local spectral normalized gap (LSNG), which measures local spectral typicality and produces dimension-comparable weights over the full pool. On a pool of 192 candidates from 2 general-purpose backbones and 6 finetuned experts, LSNG soft ensembling gains +7.8 to +54.1 percentage points (pp) in accuracy over the DINOv2-L last-layer default under a linear probe. This is within 1.0 pp of the test-selected best fixed candidate on 3/5 evaluations and above it on the other 2. Levy hard routing evaluates 1 probe per sample, matching the fixed default's probe cost, and improves accuracy by up to +31.1 pp on 3 evaluations. It fails when strongly finetuned off-domain experts imitate typical embedding norms and attract the argmax; soft ensembling is less sensitive to this failure. Across this heterogeneous library, training-free embedding geometry provides enough signal to route between frozen encoders.

1 INTRODUCTION

Pre-trained vision encoders span general-purpose self-supervised backbones (Oquab et al., 2023; He et al., 2022; Radford et al., 2021) and domain-specific specialists such as BiomedCLIP (Zhang et al., 2023a) and RemoteCLIP (Liu et al., 2024). The fixed-backbone approach selects a single backbone, fits a probe on its last layer, and applies that probe everywhere. Yet no single (backbone, layer) candidate is best for every sample of a heterogeneous task. The best backbone shifts with the input domain (Goldblum et al., 2023), while intermediate layers tend to transfer better under distribution shift and late layers specialize more strongly to the pre-training objective (Evci et al., 2022; Kumar et al., 2022; Skean et al., 2025). Choosing one candidate up front fixes a single trade-off between backbone and depth. A per-sample router can use the full candidate pool instead.

We pose this as an unsupervised inference-time model-selection problem: given a library of frozen encoders with probes fit once on the train set, decide which candidate handles each test sample without using test labels. The fixed-backbone default applies 1 model, such as DINOv2, to every sample, while the per-dataset oracle selects the best single candidate by test accuracy. Our router instead makes per-sample decisions at inference time without labels (Figure 1).

Contributions. We study label-free per-sample routing over frozen encoders and layers using embedding typicality. We adapt the existing Levy score (Maes et al., 2025) for hard routing with 1 probe evaluation per sample and introduce LSNG, a dimension-normalized local spectral score for full-pool posterior averaging. Across 192 candidates from 8 backbones, LSNG improves accuracy over the DINOv2-L last-layer default by $+7.8$ to $+54.1$ pp under linear probes. It comes within

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

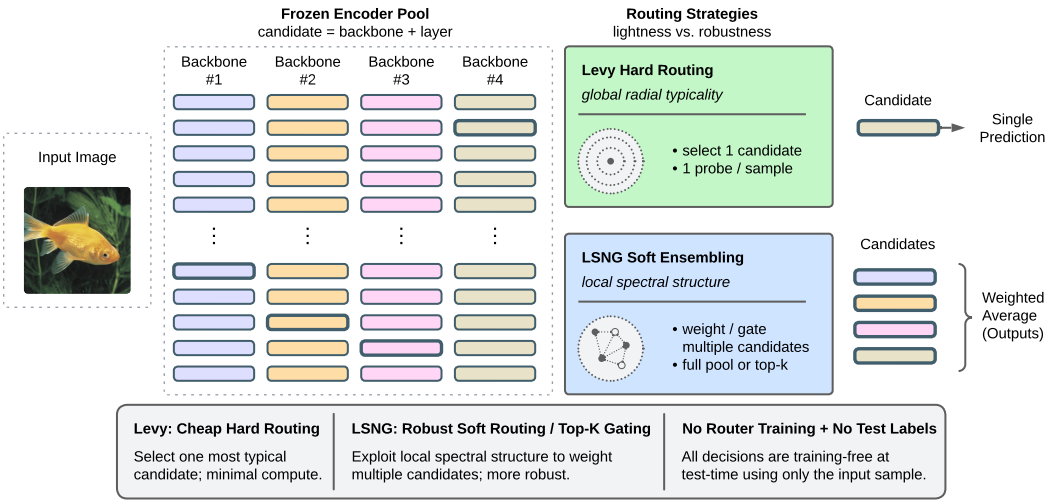

Figure 1: We route each sample over a frozen pool of (backbone, layer) candidates using embedding geometry. At test time, an unsupervised score ranks candidates using only their embedding geometry, with no supervised router training, no test labels, and no per-candidate calibration. Levy (hard routing) reads global radial typicality from the embedding norm and selects one candidate per sample (one probe). LSNG (soft ensembling) reads local spectral structure and weights every candidate's posterior by its score (full pool).

1.0 pp of the per-dataset oracle on $3/5$ evaluations and exceeds it on the remaining 2. Levy improves over the default by up to $+31.1$ pp on $3/5$ evaluations while evaluating 1 probe per sample.

2 Related Work

Routing and mixtures of experts. Mixture-of-experts methods route inputs to specialized sub-models, but the router and the experts are trained jointly on the same task (Shazeer et al., 2017; Fedus et al., 2022). Our setting is different: the experts are frozen, independently pre-trained encoders, the router is unsupervised, and routing happens at inference time on held-out samples without supervised router training.

Backbone selection and transferability scores. A line of work scores how well a pre-trained model will transfer to a downstream task using only embeddings and labels from a small probe set. LogME (You et al., 2021) marginalizes over a Bayesian linear probe, LEEP (Nguyen et al., 2020) uses the source-classifier distribution evaluated on target labels, and NCE-style transferability (Tran et al., 2019) measures conditional entropy between source and target labels. PARC (Bolya et al., 2021) and Model Spider (Zhang et al., 2023b) scale this per-task selection to large heterogeneous model zoos. These scores answer "which model for this task" using labeled data. We answer "which model for this sample" without labels.

Per-input model routing. Another line of work selects models per query rather than per dataset, almost exclusively in the language-model setting. Supervised large language model (LLM) routers use per-model classifiers trained on benchmark labels (Shnitzer et al., 2024), preferences between a strong and a weak model (Ong et al., 2025), or query-conditioned Bradley–Terry models (Frick et al., 2025). RouterBench (Hu et al., 2024) standardises evaluation on a 405k-query benchmark of LLM responses. Smoothie (Guha et al., 2024) removes the label requirement by scoring generations against the pool consensus in one shared embedding space. These methods either require labeled supervision or assume a shared output space. We instead score the embeddings directly, without supervised router training, across heterogeneous encoder backbones and layers.

Embedding typicality and out-of-distribution detection. Sample-wise scores from a fitted classifier include the max-softmax baseline (Hendrycks & Gimpel, 2017) and energy scores (Liu et al., 2020), while embedding-side scores include Mahalanobis distance to per-class Gaussians (Lee et al., 2018). ViM (Wang et al., 2022) scores a sample by the feature residual off the train principal subspace,

a global version of the tangent-versus-normal split that LSNG reads locally per cluster. These scores are designed to flag a single test sample as in- or out-of-distribution against one fitted model. Levy (Maes et al., 2025) is a per-sample confidence score for joint-embedding models, built from the chi distribution of embedding norms under the isotropic-Gaussian assumption. We use the same statistic, but as an unsupervised router score across candidates rather than as a confidence filter for one model.

Self-supervised learning and embedding geometry. Joint-embedding methods promote hyperspherical uniformity and variance-controlled isotropy (Wang & Isola, 2020; Bardes et al., 2022; LeCun, 2022; Assran et al., 2023; Balestriero & LeCun, 2025). This motivates the Gaussian-norm assumption behind the hard-routing with the Levy score. RankMe (Garrido et al., 2023) predicts representation quality from the global spectrum without labels. LSNG instead reads a local spectral gap around each test sample, normalized to be comparable across embedding dimensions.

3 LEVY AND LSNG: TRAINING-FREE ROUTING SCORES

We use two training-free routing scores: Levy (Section 3.2), a norm-based typicality score for hard routing, and LSNG (Section 3.3), a local spectral score for score-weighted soft ensembling. Here, training-free means no supervised router training; LSNG fits unlabeled geometric statistics on the train embeddings. A score-weighted plurality variant is reported in Appendix M.

3.1 SETTING AND ROUTING RULES

We consider routing over a model library of frozen pre-trained encoders. Let $\mathcal{B} = \{b_1, \dots, b_M\}$ be a set of backbones, where backbone b has L_b layers. A *candidate* is a backbone–layer pair $c = (b, l)$, and the candidate set is

$$C = \{(b, l) : b \in \mathcal{B},\ l \in \{1, \dots, L_b\}\}. \tag{1}$$

For an input x, candidate c produces an embedding $z_c(x) \in \mathbb{R}^{K_c}$, where K_c is its embedding dimension.

For each candidate $c \in C$, we fit a fixed probe $p_c(y \mid x)$ on the train split of the downstream task. The probe outputs a posterior distribution over labels $y \in \mathcal{Y}$. We report linear probes and kNN probes with $k = 20$ throughout. Let

$$\hat{y}_c(x) = \arg\max_{y \in \mathcal{Y}} p_c(y \mid x) \tag{2}$$

denote the prediction of candidate c. All probes are trained once and kept fixed at routing time.

At test time, the router receives per-candidate unsupervised scores $\{s_c(x)\}_{c \in C}$ computed from the embeddings $\{z_c(x)\}_{c \in C}$. These scores may not use labels, test targets, or any additional supervised calibration. We consider two ways to convert these scores into predictions.

Hard score routing. Hard routing selects the highest-scoring candidate and returns its prediction:

$$c^\star(x) = \arg\max_{c \in C} s_c(x), \qquad \hat{y}_{\text{route}}(x) = \hat{y}_{c^\star(x)}(x). \tag{3}$$

This rule evaluates 1 probe per sample after the scores have been computed.

Score-weighted soft ensembling. Soft ensembling converts scores into weights following

$$w_c(x) = \frac{\exp(s_c(x)/\tau)}{\sum_{c' \in C} \exp(s_{c'}(x)/\tau)}, \quad \tau > 0. \tag{4}$$

It then averages candidate posteriors and predicts the most likely label:

$$\hat{y}_{\text{ens}}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{c \in C} w_c(x)\, p_c(y \mid x). \tag{5}$$

This rule combines classifier confidence through $p_c(y \mid x)$ with embedding-based candidate weights through $w_c(x)$.

3.2 SCORING EMBEDDINGS WITH LEVY

Hyperspherical uniformity (Wang & Isola, 2020), variance–covariance regularization (Bardes et al., 2022), and isotropic-Gaussian regularization (Balestriero & LeCun, 2025) motivate well-spread embedding geometry. This supports Levy's Gaussian radial reference as a typicality proxy (Maes et al., 2025), although it may not hold exactly for every DINOv2 or MAE layer.

For a candidate c with embedding $z_c(x) \in \mathbb{R}^{K_c}$, the Levy score is the log-density of the Chi distribution with K_c degrees of freedom evaluated at the embedding norm,

$$\mathrm{Levy}(z_c(x)) \;=\; \log \mathrm{pdf}_{\chi_{K_c}}\big(\|z_c(x)\|_2\big). \tag{6}$$

Under the standard isotropic Gaussian assumption, $\|z_c(x)\|_2$ follows exactly χ_{K_c}, so a sample whose norm is far from the mode of χ_{K_c} is unusual under this reference; we use this deviation as a proxy for candidate reliability. Computing Levy requires only the embedding norm and its Chi log-density (Figure 5).

3.3 SCORING EMBEDDINGS WITH LSNG

Full-pool soft ensembling requires scores that weight every candidate's posterior without train-time calibration and that remain comparable across embedding dimensions. LSNG is the score we propose for this rule. For a candidate c, we partition its train embeddings into G clusters via mini-batch k-means. For each cluster g, we compute the local covariance spectrum $\{\lambda_{g,i}\}_{i=1}^{K_c}$. The cluster-level LSNG score is the difference between the mean log-eigenvalues of two sets: the tangent set (the t leading eigenvalues, or data-supporting directions) and the normal set (the $r = K_c - t$ trailing eigenvalues, or orthogonal residual):

$$\mathrm{LSNG}_c(z_c(x)) \;=\; \frac{1}{t}\sum_{i=1}^{t} \log \lambda_{g^\star(x),\,i} \;-\; \frac{1}{r}\sum_{i=t+1}^{K_c} \log \lambda_{g^\star(x),\,i}, \tag{7}$$

where $g^\star(x)$ is the cluster of candidate c's train set whose centroid is closest to $z_c(x)$. The unregularized score is unitless and scale-invariant: a common eigenvalue rescaling cancels between the two log-averages. Regularization preserves exact invariance when scaled with the covariance (Appendix B). Two candidates whose tangent and normal blocks have the same per-coordinate eigenvalue scale therefore produce comparable LSNG values regardless of K_c, allowing their scores to be used directly as soft-ensembling weights.

Remark 1. *Equation* (5) *is the Bayes predictive distribution when $w_c(x)$ and p_c are the candidate and candidate-conditional label posteriors (Proposition 9). LSNG is motivated as a structural proxy for candidate quality, not as an exact posterior.*

4 WHY THESE TWO SCORES: A MODEL-BASED RATIONALE

The two scores in Section 3 target different geometric properties: Levy tests global radial typicality, whereas LSNG measures local spectral structure. We derive both quantities under surrogate models. Full statements and proofs are in Appendix B.

4.1 LEVY: GLOBAL RADIAL TYPICALITY

Levy is a radial typicality score for hard routing. It does not use the direction of the embedding, only its norm. Under the isotropic reference model $Z \sim \mathcal{N}(0, I_K)$, the radius $R = \|Z\|_2$ follows a χ_K distribution (Lemma 2), whose log-density is

$$\log p_{\chi_K}(r) = (K-1)\log r - r^2/2 + \mathrm{const.}$$

Thus, (6) scores whether candidate c places x near the typical high-dimensional radius. This differs from Gaussian energy: the shell-volume term $(K-1)\log r$ makes the score maximal near $r = \sqrt{K-1}$, not at the origin (Corollary 4).

A latent-candidate model clarifies the role of the score. If $C^\star$ denotes the candidate appropriate for x, and $\rho_c(x) = \|\tilde{z}_c(x)\|_2$, then a Bayes selector based on radial statistics has the form

$$\hat{c}(x) = \arg\max_{c \in \mathcal{C}} \left[\log \pi_c + \log f_{1,c}(\rho_c(x)) - \log f_{0,c}(\rho_c(x))\right].$$

When the matched candidate satisfies $\tilde{z}_c(x) \mid C^\star = c \sim \mathcal{N}(0, I_{K_c})$, the matched-candidate term is exactly Levy (Proposition 3). Thus Levy is a matched radial likelihood term, not a complete posterior: priors, off-candidate densities, and cross-candidate calibration are not modeled unless explicitly added.

Levy comparisons require comparable embedding norm scales. Equal embedding dimension gives the same χ_K reference distribution, but if candidates produce embeddings with different coordinate scales, the radii should be standardized or calibrated before comparing scores. Both tested calibration transforms lower hard-routing accuracy on every vision evaluation. Per-candidate calibration removes differences relative to the common reference (Appendix N).

4.2 LSNG: LOCAL SPECTRAL STRUCTURE FOR SOFT ENSEMBLING

Section 3.3 defines LSNG as a normalized log-volume gap between tangent and normal covariance eigenvalue blocks, and Section 3.1 uses it to weight candidate posteriors in the full-pool ensemble. We justify this choice through three properties: scale comparability, local block interpretability, and compatibility with score-weighted posterior averaging.

Comparability and tangent–normal interpretation. LSNG is designed to be comparable across candidates because it is dimension-normalized and, before regularization, invariant to positive scalar rescaling of each candidate's embeddings. If a candidate's embeddings are transformed as $z \mapsto az + b$, $a > 0$, all local covariance eigenvalues are multiplied by a^2. The resulting $\log a^2$ shift appears in both the tangent and normal log-averages and cancels. In the block-isotropic case

$$\Sigma_g = \sigma_T^2 P_T + \sigma_N^2 P_N,$$

where P_T, P_N are tangent and normal projectors of ranks t, r, LSNG reduces to

$$\mathrm{LSNG}_g = \log \sigma_T^2 - \log \sigma_N^2.$$

Thus LSNG is a per-coordinate log ratio between tangent and normal variance. The ridge-regularized score is stable under small local covariance-estimation error when the regularized eigenvalues are bounded away from zero (Propositions 5–6).

Local block model. To connect this spectral ratio to candidate quality, consider the local model

$$Z = (\alpha(M_Y + \Xi_T),\, \beta\Xi_N) \in \mathbb{R}^t \times \mathbb{R}^r, \qquad L := \mathrm{LSNG} = \log\big(\alpha^2(\sigma_B^2 + \tau_T)\big) - \log(\beta^2 \tau_N),$$

inside the nearest train cluster $g_c^\star(x)$. Here M_Y is label-relevant tangent signal, Ξ_T is label-irrelevant tangent variation, and Ξ_N is orthogonal residual variation. The expression for L follows under the covariance assumptions of Theorem 7. Thus LSNG measures total tangent variation, including label-relevant signal, relative to orthogonal residual variation.

The same model gives a monotone relation with the label-aligned clustering gain

$$D_Y := \mathbb{E}\|Z - \mathbb{E}[Z \mid Y]\|_2^2, \qquad D_1 := \mathbb{E}\|Z - \mathbb{E}Z\|_2^2, \qquad \Gamma := \tfrac{D_1 - D_Y}{D_Y}.$$

Under the assumptions of Theorem 8,

$$\Gamma(L) = \frac{t\sigma_B^2}{t\tau_T + r(\sigma_B^2 + \tau_T)e^{-L}}, \qquad \frac{d\Gamma}{dL} > 0.$$

The derivative is positive for fixed t, r, σ_B^2, τ_T with $\sigma_B^2 > 0$. Higher LSNG corresponds to better label-aligned clusterability under this conditional model. The relation does not hold unconditionally, since increasing tangent nuisance variation can raise leading eigenvalues without improving semantic separation.

Score-weighted averaging. The posterior-mean rule (5) is decision-theoretically justified when the score-derived weights are calibrated candidate posteriors. If each $p_c(y \mid x)$ is the candidate-conditional label posterior and

$$w_c(x) = \mathbb{P}(C = c \mid x, \{s_j(x)\}_{j \in \mathcal{C}}), \qquad p(y \mid x, \{s_j(x)\}_{j \in \mathcal{C}}) = \sum_{c \in \mathcal{C}} w_c(x) p_c(y \mid x),$$

then the second identity is the Bayes predictive distribution and minimizes posterior expected log loss (Proposition 9). This identity requires calibrated weights; LSNG supplies a structural proxy for candidate quality.

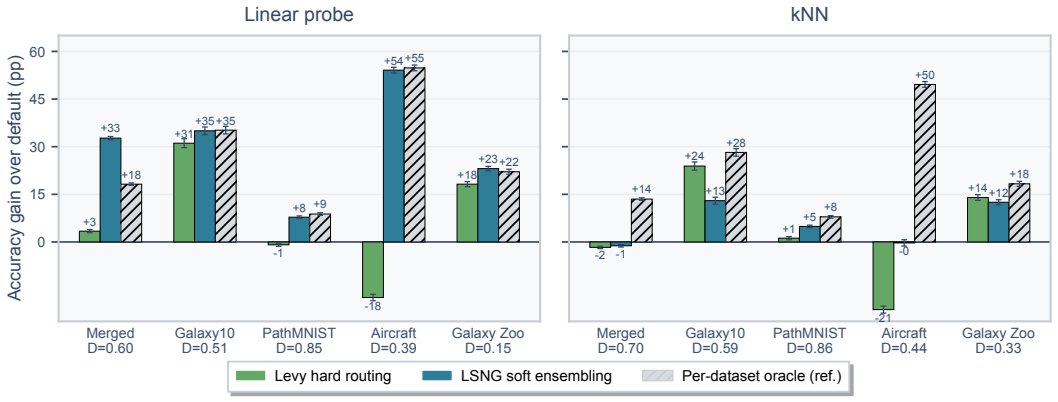

Figure 2: We report accuracy gains (pp) over the DINOv2-L last-layer default for Levy hard, LSNG soft, and the per-dataset oracle. The left panel uses linear probes; the right uses kNN with 20 neighbours. Error bars show paired bootstrap standard deviations over 1000 resamples. Default accuracy D appears below each dataset.

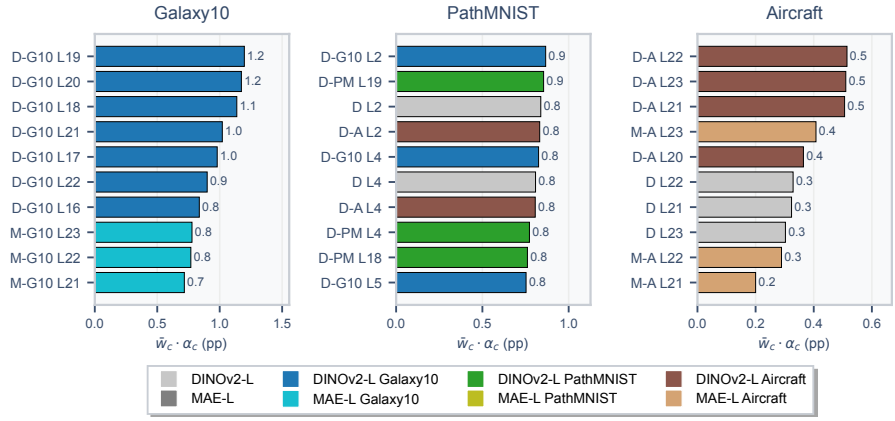

Figure 3: We show the 10 candidates with the largest LSNG contributions $\bar{w}_c \alpha_c$ (pp; Section 5.3). Colors indicate backbone family: D = DINOv2-L, M = MAE-L. Expert codes: G10 = Galaxy10, PM = PathMNIST, A = Aircraft. Appendices F and F.2 report Merged, Galaxy Zoo, and Levy hard contributions.

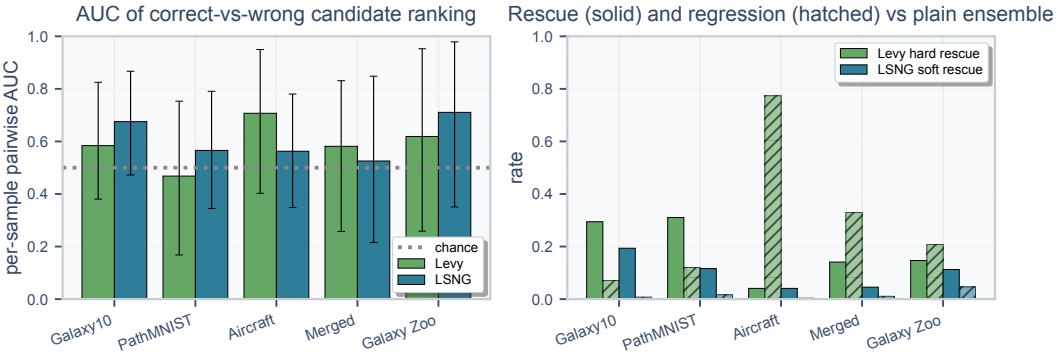

Figure 4: The left panel shows mean per-sample candidate-ranking AUC; error bars span the 10th–90th percentiles across samples. The right panel shows conditional rescue (solid) and regression (hatched) rates against uniform ensembling. Section 5.3 defines the statistics and their conditioning subsets.

5 EXPERIMENTS

We evaluate training-free embedding typicality as a per-sample routing signal on a curated library of pre-trained encoders.

The candidate pool contains 8 backbones: 2 general-purpose (DINOv2 ViT-L/14 (Oquab et al., 2023), MAE ViT-L (He et al., 2022)) and 6 finetuned experts trained on Galaxy10 (Leung & Bovy, 2021), PathMNIST (Yang et al., 2023), and Aircraft (Maji et al., 2013) (2 backbones $\times$ 3 tasks), each contributing 24 layers for $|C| = 192$ candidates; experts beat their general backbone on the target task by $+8.0$ to $+54.8$ pp (Appendix C.3). We group the 5 evaluations into 3 categories: (i) *Single-task*, comprising Galaxy10, PathMNIST, and Aircraft, each with task-matched experts finetuned on its train split; (ii) *Merged*, combining the 3 test sets into a 119-class task; and (iii) *Same-domain unseen*, using Galaxy Zoo (Walmsley et al., 2023) morphology classes that no expert saw during finetuning. These categories test routing within a task, across domains, and to unseen classes. We report a linear probe and a kNN head (k=20, cosine) side-by-side throughout, against the fixed-backbone default (DINOv2-L last layer) and the per-dataset oracle.

5.1 BASELINE AND REFERENCE POINTS

We compare every routing method with 3 reference points. (i) The *fixed-backbone default* fits a linear probe on the final layer of 1 general-purpose backbone, DINOv2-L, and applies it to every input: standard deployment when no routing is used. (ii) The *per-dataset oracle* selects the single best *(backbone, layer)* candidate per dataset by labeled evaluation accuracy. It is the best fixed-candidate reference, not an upper bound: per-sample routing can outperform it by selecting different candidates for different samples, which we observe on Merged and Galaxy Zoo in Section 5.2. It is also distribution-specific, so its advantage need not transfer under distribution shift. (iii) A *random* router selects a candidate uniformly at random per sample: a useful router should beat it and the fixed default. We also compare with task-matched finetuned encoders (Appendix C.3), uniform posterior averaging, and max-confidence selection (Appendix M).

5.2 DOES TRAINING-FREE ROUTING BEAT THE FIXED BACKBONE?

We compare Levy hard routing and LSNG soft ensembling with the fixed-backbone default across 5 evaluations using linear probes (Figure 2). Levy hard routing uses 1 probe per sample, matching the fixed-backbone probe cost. It gains $+31.1$ pp on Galaxy10, $+18.2$ pp on Galaxy Zoo, and $+3.4$ pp on Merged, but loses 0.9 pp on PathMNIST and 17.5 pp on Aircraft. On Aircraft, off-domain experts produce embedding norms that look typical to Levy, and these norm decoys pull the argmax away from the matching expert on 83% of samples (Appendix N). At $|C|$ probes per sample, LSNG soft ensembling gains $+7.8$ to $+54.1$ pp over the fixed-backbone default. On the 3 Single-task evaluations, it comes within 1.0 pp of the per-dataset oracle. On Merged, it exceeds the oracle by 14.5 pp. On the Same-domain unseen evaluation, Galaxy Zoo, it exceeds the oracle by 1.0 pp. On Merged, combining predictions therefore achieves higher accuracy than selecting any single candidate for the entire evaluation. We next replace the linear probes with kNN heads. LSNG soft accuracy drops by up to 49 pp relative to the linear head. Selecting candidates before averaging improves the kNN results, as we examine in Section 5.4. The aggregation rule therefore matters when the probe head changes.

Figure 5 (Appendix E) reports component fit-and-evaluation totals for an 8-backbone pool, normalized by test-set size. These include offline fitting and are not online inference latencies. Routing embeds every test sample with all 8 backbones, about $8\times$ the default's encoder cost. Hard routing then evaluates one probe, whereas LSNG soft evaluates all $|C|$ probes. LSNG-lite reduces fit-and-score time by a median $13.4\times$ in the computational-cost evaluation (Appendix I). Fitting and evaluating the full 192-probe pool takes 5.9 min to 1.55 h, compared with a mean 1.8 to 29.1 s per probe; feature extraction is a separate one-time cost (Appendix E.2). Without router training, test labels, or per-candidate calibration, soft ensembling based on embedding geometry improves consistently over the fixed-backbone default under the linear head. On Merged, the full-pool posterior average beats the per-dataset oracle with either uniform or LSNG weights, while a Levy top-k gate recovers this accuracy with fewer probe evaluations (Appendix K). Appendices O–S report additional baselines, probe-capacity and text-modality studies, a larger-encoder control, and a heterogeneous-pool control.

5.3 WHICH CANDIDATES DOES THE SCORE PICK?

To see where each score concentrates its weight, we measure candidate c's contribution as $\bar{w}_c \cdot \alpha_c$: its mean softmax weight times its accuracy on the samples receiving that weight (Appendix F). Figure 3 shows the candidates with the largest contributions. LSNG concentrates on the matching DINOv2-L expert family on Galaxy10 and Aircraft. On PathMNIST it mixes the matching expert with early generic layers, where most candidates are already correct. On Merged it splits weight between the DINOv2-L PathMNIST and Galaxy10 experts, and on Galaxy Zoo, whose classes the expert never saw, it concentrates on the DINOv2-L Galaxy10 family. Levy hard concentrates similarly where it succeeds but is drawn to off-domain experts on Aircraft (Appendix N).

To assess candidate ranking, we measure the *per-sample pairwise AUC* (Figure 4): for each sample, the fraction of (correct, wrong) candidate pairs where the score ranks the correct one higher (computed only when both populations exist). AUC > 0.5 indicates useful cross-candidate ordering, but does not by itself distinguish input-dependent ranking from fixed candidate preferences. LSNG exceeds chance on every evaluation, reaching 0.68 on Galaxy10 and 0.71 on Galaxy Zoo. Levy exceeds chance on 4/5 evaluations, peaking at 0.71 on Aircraft and 0.62 on Galaxy Zoo. Levy is slightly below chance on PathMNIST (0.47). These measurements support cross-candidate ranking as a routing signal, under the evaluated candidate pools.

We examine prediction changes by comparing each rule against the *plain ensemble* (uniform posterior average then argmax). The *rescue rate* is the fraction of plain-ensemble mistakes a rule recovers when at least one candidate is correct; the *regression rate* is the fraction of plain-ensemble correct samples a rule misclassifies. At $\tau = 1$, LSNG rescues 19% of eligible uniform-ensemble mistakes on Galaxy10. It changes 1% of correct uniform-ensemble predictions into errors. On Galaxy Zoo, the rescue rate is 11%. The regression rate is 5%. Appendix M.1 reports the corresponding Levy rates. The rates condition on different subsets; net accuracy gains depend on the corresponding counts. Plain full-pool ensembling is the $\tau \to \infty$ endpoint of (4), providing a matched baseline that isolates the benefit of score weighting (Lakshminarayanan et al., 2017). Under the linear head, it matches LSNG weighting to within 1 pp on PathMNIST, Aircraft, and Merged (Appendix M). LSNG weighting adds gains of $+2.0$ pp on Galaxy10 and $+1.7$ pp on Galaxy Zoo, together with a record of the candidates weighted for each input. In the heterogeneous 372-candidate specialist pool, these gains over uniform averaging reach $+4.4$ and $+4.7$ pp, respectively (Appendix S.2).

5.4 CAN THE RANKING SUPPORT A SMALLER ENSEMBLE?

Hard routing uses only the highest-scoring candidate. To examine whether useful candidates remain below that first choice, we use the same Levy ranking in 2 steps: (i) select the k highest-scoring candidates for each sample, and (ii) average their posteriors with equal weights. This separates candidate selection from score weighting within the selected set. At $k = 1$, the rule is hard routing; at $k = 192$, it averages the full pool (Appendix K).

On Aircraft with linear probes, accuracy rises from 0.212 for hard routing to 0.789 when averaging the top 4 candidates. At $k = 64$, accuracy reaches 0.934, compared with 0.935 for the best fixed candidate. On Merged, the same $k = 64$ rule reaches 0.937, exceeding the best fixed candidate's 0.784. Thus, the ranking contains useful candidates beyond its first choice, and combining their predictions can recover accuracy when the argmax selects an unsuitable candidate.

Under kNN heads, this rule exceeds the fixed-backbone default on all 5 evaluations at every tested $k \in \{4, 8, 16, 32, 64\}$ (Appendix L). At $k = 16$, it comes within 2.5 pp of the best fixed candidate on the 3 Single-task evaluations. It exceeds that reference on both Merged and Same-domain unseen. This setting evaluates 16 of the 192 probes per sample, while still computing embeddings from all 8 backbones. The score therefore supports an intermediate probe budget between hard routing and full-pool averaging.

5.5 ROBUSTNESS

We evaluate robustness to candidate-pool corruption, pool enlargement, and class imbalance.

Pool corruption. We inject 1, 2, or 4 random-init DINOv2-L backbones into the candidate pool (Table 1b top). Random-init candidates can match typical norms without useful semantics; LSNG

Table 1: Robustness checks. **(a)** Balanced accuracy (Bal) and worst-class recall (Worst); all methods have zero worst-class recall on Galaxy Zoo (Appendix H). **(b)** Accuracy after adding random backbones on Merged or weight-perturbed copies on Galaxy10 (mild: σ=0.01; harsh: 0.05). Uniform averages posteriors over the same pool. Values $\pm$ are bootstrap standard deviations (1000 resamples). Full results and pool sizes: Appendices H and G.

(a) Class-balanced metrics (best in bold).

Eval.	Method	Bal	Worst
Galaxy10	DINOv2-L	$0.515_{\pm 0.014}$	$0.198_{\pm 0.023}$
	Levy	$0.814_{\pm 0.011}$	$\mathbf{0.637}_{\pm 0.034}$
	LSNG	$\mathbf{0.852}_{\pm 0.010}$	$0.615_{\pm 0.046}$
Aircraft	DINOv2-L	$0.387_{\pm 0.007}$	$0.000_{\pm 0.000}$
	Levy	$0.212_{\pm 0.007}$	$0.000_{\pm 0.000}$
	LSNG	$\mathbf{0.928}_{\pm 0.004}$	$\mathbf{0.424}_{\pm 0.074}$
Merged	DINOv2-L	$0.399_{\pm 0.005}$	$0.000_{\pm 0.000}$
	Levy	$0.296_{\pm 0.005}$	$0.000_{\pm 0.000}$
	LSNG	$\mathbf{0.910}_{\pm 0.004}$	$\mathbf{0.455}_{\pm 0.068}$

(b) Pool corruption (best in bold).

Pool	Levy	LSNG	Uniform
Merged, random-init injection			
Clean	$0.636_{\pm 0.004}$	$0.929_{\pm 0.002}$	$\mathbf{0.937}_{\pm 0.002}$
+1 random	$0.316_{\pm 0.004}$	$0.862_{\pm 0.003}$	$\mathbf{0.931}_{\pm 0.002}$
+2 random	$0.316_{\pm 0.004}$	$0.790_{\pm 0.004}$	$\mathbf{0.910}_{\pm 0.002}$
+4 random	$0.316_{\pm 0.004}$	$0.707_{\pm 0.004}$	$\mathbf{0.851}_{\pm 0.003}$
Galaxy10, weight-perturbation			
Clean	$0.824_{\pm 0.009}$	$\mathbf{0.863}_{\pm 0.008}$	$0.843_{\pm 0.009}$
+mild	$0.822_{\pm 0.009}$	$\mathbf{0.861}_{\pm 0.008}$	$0.845_{\pm 0.008}$
+both	$0.822_{\pm 0.009}$	$\mathbf{0.861}_{\pm 0.008}$	$0.848_{\pm 0.008}$

instead measures local spectral structure. The experiment tests how these distinct signals respond. Levy hard accuracy falls from 0.636 to 0.316 at +1 random and remains at this level, while LSNG soft falls from 0.929 on the clean pool to 0.862 at +1 and 0.707 at +4 (mass-allocation breakdown in Appendix G). Plain ensembling degrades the least (0.937 to 0.851): equal weights dilute the injected candidates, at the full $|C|$-probe cost and with no way to exclude them. Levy therefore needs a roughly homogeneous curated pool; LSNG soft retains higher accuracy under the tested corruption.

Pool enlargement. We test sensitivity to redundant candidates by adding one mild (σ=0.01) and one harsh (σ=0.05) Gaussian-noise copy of each of the 8 backbones (Table 1b bottom): both rules are unchanged within one bootstrap standard deviation (Levy hard 0.824 to 0.822, LSNG soft 0.863 to 0.861 on Galaxy10). Tripling the pool with noisy copies neither helps nor destabilizes either rule (Appendix G). *Class imbalance.* LSNG soft improves balanced accuracy over DINOv2-L by +33.7 to +54.1 pp, with worst-class recall rising from 0.20 to 0.62 on Galaxy10 and from 0.00 to 0.42 on Aircraft (Table 1a; full per-class metrics in Appendix H). LSNG soft remains effective under class imbalance, random-init injection, and pool enlargement. Levy hard is stable under pool enlargement but requires a vetted pool: random-init backbones and heavily finetuned off-domain experts can both capture its argmax.

6 Conclusion

We studied per-sample routing over a heterogeneous library of frozen pre-trained vision encoders, using Levy for hard routing and LSNG for soft ensembling without per-candidate calibration. Across finetuned experts and general-purpose backbones, LSNG soft ensembling improves on the DINOv2-L last-layer default by +7.8 to +54.1 pp under a linear probe. Levy hard routing improves on the same default by up to +31.1 pp on 3/5 evaluations while using 1 probe per sample. Pairwise AUC supports cross-candidate ordering, and top-k averaging uses this ranking to recover accuracy with fewer probe evaluations than full-pool ensembling. Random-init candidates and finetuned off-domain experts can imitate typical norms and dominate Levy's argmax; LSNG soft is more robust to both in the tested pools.

AI USE STATEMENT

We used generative AI tools to help revise and edit the manuscript and assist with implementation, primarily code improvement. AI assistance also supported the review of methodological wording and LaTeX formatting. AI tools were not used to develop proofs. The authors thoroughly reviewed all AI-assisted work and take responsibility for the final text, claims, and research artifacts.

ETHICS STATEMENT

This work studies representation selection using existing image and text datasets. The medical-image evaluations assess classification performance, not clinical validity. We report class-balanced metrics and worst-class recall to make performance disparities visible, and examine failures caused by unsuitable candidates in the model pool.

REPRODUCIBILITY STATEMENT

Section 3 specifies the routing rules and scores, with theoretical assumptions and proofs in Appendix B. Appendix C describes the model library, datasets, and expert fine-tuning; Appendix D specifies embedding extraction, probe fitting, LSNG configuration, seeds, and uncertainty estimation. Appendix E reports computational resources and costs, and subsequent appendices provide baseline comparisons and ablations. Code will be released after acceptance.

REFERENCES

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

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics, 2025. URL https://arxiv.org/abs/2511.08544.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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.

Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C. Mozer. Head2Toe: Utilizing intermediate representations for better transfer learning. In *International Conference on Machine Learning (ICML)*, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 2022.

Evan Frick, Connor Chen, Joseph Tennyson, Tianle Li, Wei-Lin Chiang, Anastasios N. Angelopoulos, and Ion Stoica. Prompt-to-leaderboard. In *International Conference on Machine Learning (ICML)*, 2025.

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*, 2023.

Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, Rama Chellappa, Andrew Gordon Wilson, and Tom Goldstein. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Neel Guha, Mayee F. Chen, Trevor Chow, Ishan S. Khare, and Christopher Ré. Smoothie: Label free language model routing. In *Advances in Neural Information Processing Systems*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yann LeCun. A path towards autonomous machine intelligence. *OpenReview*, 2022.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Henry W. Leung and Jo Bovy. Galaxy10 DECaLS: A galaxy image dataset. `https://github.com/henrysky/Galaxy10`, 2021. astroNN dataset.

Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. RemoteCLIP: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Lucas Maes, Damien Scieur, and Randall Balestriero. Levyscore: A fast sample-wise confidence score of pretrained joint embedding model. In *UniReps: 3rd Edition of the Workshop on Unifying Representations in Neural Models*, 2025. URL `https://openreview.net/forum?id=sKRhHq9QyZ`.

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

Cuong V. Nguyen, Tal Hassner, Matthias Seeger, and Cédric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning (ICML)*, 2020.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations (ICLR)*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. In *Conference on Language Modeling (COLM)*, 2024.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *International Conference on Machine Learning*, 2025.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Anh T. Tran, Cuong V. Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *International Conference on Computer Vision (ICCV)*, 2019.

Mike Walmsley, Tobias Géron, Sandor Kruk, Anna M. M. Scaife, Chris Lintott, Karen L. Masters, James M. Dawson, Hugh Dickinson, Lucy Fortson, Izzy L. Garland, Kameswara Mantha, David O'Ryan, Jürgen Popp, Brooke Simmons, Elisabeth M. Baeten, and Christine Macmillan. Galaxy zoo DESI: Detailed morphology measurements for 8.7m galaxies in the DESI legacy imaging surveys. *Monthly Notices of the Royal Astronomical Society*, 526(3):4768–4786, 2023.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-of-distribution with virtual-logit matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning (ICML)*, 2020.

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*, 2023.

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)*, 2021.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. *arXiv preprint arXiv:2303.00915*, 2023a.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model spider: Learning to rank pre-trained models efficiently. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

A LIMITATIONS

Levy hard routing assumes a curated candidate pool. A random-init backbone can produce norms close to the mode of the Levy reference, and even one such candidate can dominate its argmax (Section 5.5, Appendix G). Heavily finetuned off-domain experts distort the same signal: on Aircraft, the argmax abandons the matching expert on 83% of samples and falls below the default (Appendix N). Uniform averaging over the Levy top-k recovers accuracy at k probes per sample (Appendix K). LSNG soft ensembling degrades more gradually under the same corruption, combining mild spectral down-weighting with the dilution of posterior averaging (Appendix G), but it requires the full $|C|$-probe cost. For an unvetted candidate pool, LSNG soft is more robust than Levy hard routing in our experiments.

Both scores judge embedding geometry only. They cannot detect the synthetic bad probes in Appendix G.6: a bad probe attached to a real embedding still passes both tests, and the score distributes weight equally between the bad probe and a good probe attached to the same embedding. Probe-quality measures, such as held-out accuracy, provide information that embedding geometry alone cannot capture.

The pairwise routing AUC analysis (Section 5.3) identifies a cross-candidate ranking signal: given a sample, the score tends to rank competent candidates above incompetent ones. Neither score estimates the reliability of one candidate on one sample. Cross-candidate ranking and within-candidate reliability are distinct quantities.

The main experiments use a model library of 2 general-purpose ViT-L backbones (DINOv2-L/14, MAE-L/16) and six experts produced by finetuning each on Galaxy10, PathMNIST, and Aircraft. A control pool in Appendix S mixes architecture families, pretraining paradigms, and embedding dimensions (384 to 2048). LSNG remains dimension-comparable there and Levy confirms its homogeneous-scale requirement, with neither routed operator beating uniform averaging in the pool without specialists. With specialists added, LSNG beats uniform averaging on 3/5 evaluations. The vision and text experiments evaluate each modality separately.

B THEORETICAL RESULTS FOR THE LEVY SCORE AND LSNG

B.1 THE LEVY SCORE: GLOBAL RADIAL TYPICALITY FOR HARD ROUTING

The Levy score used in Section 3 is not the ambient Gaussian density in $\mathbb{R}^K$. It is the density induced on the radius after integrating out the spherical volume element.

Lemma 2 (Gaussian shell density)**.** *Let $Z \sim \mathcal{N}(0, I_K)$ and $R = \|Z\|_2$. Then R has the χ_K density*

$$p_{\chi_K}(r) = \frac{1}{2^{K/2-1}\Gamma(K/2)} r^{K-1} \exp\left(-\frac{r^2}{2}\right), \qquad r > 0. \tag{8}$$

Equivalently,

$$\log p_{\chi_K}(r) = (K-1)\log r - \frac{1}{2}r^2 + \text{const}. \tag{9}$$

Proof. The density of Z is $\phi_K(z) = (2\pi)^{-K/2}\exp(-\|z\|_2^2/2)$. Write $z = ru$ with $u \in S^{K-1}$. The Euclidean volume element becomes $dz = r^{K-1}dr\, d\omega(u)$. Integrating $\phi_K(ru)$ over the sphere gives

$$p_R(r) = (2\pi)^{-K/2} e^{-r^2/2} r^{K-1} |S^{K-1}| = \frac{1}{2^{K/2-1}\Gamma(K/2)} r^{K-1} e^{-r^2/2}.$$

Taking logarithms gives the second display. □

The next proposition separates the exact Bayes rule from the score used in the implementation. Levy is the matched-candidate radial likelihood term. A fully specified Bayes router also contains candidate priors and off-candidate radial densities.

Proposition 3 (Radial Bayes router)**.** *Let $C^\star \in \mathcal{C}$ be the latent candidate whose probe is appropriate for a test input x. For each candidate c, define*

$$\rho_c(x) = \|\tilde{z}_c(x)\|_2,$$

where $\tilde{z}_c$ is the representation used for radial scoring: the embedding in the main experiments, or a train-statistic standardized embedding in calibrated variants. Assume that, conditional on $C^\star = c$, the radii are conditionally independent and have positive densities

$$\rho_c \sim f_{1,c}, \qquad \rho_j \sim f_{0,j} \quad (j \neq c),$$

with prior $\pi_c = \mathbb{P}(C^\star = c)$. Then the Bayes hard router is

$$\hat{c}_{\mathrm{Bayes}}(x) = \arg\max_{c \in \mathcal{C}} \left[\log \pi_c + \log f_{1,c}(\rho_c(x)) - \log f_{0,c}(\rho_c(x)) \right]. \tag{10}$$

If the matched candidate satisfies

$$\tilde{z}_c(x) \mid C^\star = c \sim \mathcal{N}(0, I_{K_c}),$$

then $f_{1,c} = p_{\chi_{K_c}}$, and the matched-candidate term is

$$\mathrm{Levy}_c(x) = \log p_{\chi_{K_c}}(\|\tilde{z}_c(x)\|_2). \tag{11}$$

Proof. By Bayes' rule and conditional independence,

$$\mathbb{P}(C^\star = c \mid \rho_1, \ldots, \rho_{|\mathcal{C}|}) \propto \pi_c f_{1,c}(\rho_c) \prod_{j \neq c} f_{0,j}(\rho_j).$$

Multiplying and dividing by $f_{0,c}(\rho_c)$ gives

$$\pi_c \frac{f_{1,c}(\rho_c)}{f_{0,c}(\rho_c)} \prod_{j \in \mathcal{C}} f_{0,j}(\rho_j).$$

The final product is independent of the candidate index being maximized over. Taking logarithms gives (10). The Gaussian case follows from Lemma 2. □

Thus Levy is an exact Bayes hard-router score only after additional calibration assumptions: for example, if the prior and off-candidate terms $\log \pi_c - \log f_{0,c}(\rho_c(x))$ are constant across candidates, or have already been removed by a calibration step. In the general case, Levy is the matched radial likelihood term in the Bayes likelihood ratio. We use this radial likelihood term as a routing statistic.

Corollary 4 (Radial typicality is not Gaussian energy)**.** *For $Z \sim \mathcal{N}(0, I_K)$,*

$$\log \phi_K(z) = -\frac{1}{2}\|z\|_2^2 + \mathrm{const},$$

whereas

$$\log p_{\chi_K}(\|z\|_2) = (K-1)\log \|z\|_2 - \frac{1}{2}\|z\|_2^2 + \mathrm{const}.$$

The difference is the shell-volume term $(K-1)\log\|z\|_2$. The ambient Gaussian density is maximized at the origin, while radial typicality is maximized at $\|z\|_2 = \sqrt{K-1}$ for $K > 1$.

Proof. The ambient expression is the Gaussian density. The radial expression follows from Lemma 2. The derivative of $(K-1)\log r - r^2/2$ is $(K-1)/r - r$, whose positive root is $\sqrt{K-1}$. The ambient Gaussian density decreases with $\|z\|_2$ and is therefore maximized at $z = 0$. □

Levy values are directly comparable only when candidates share a common radial reference. Equal embedding dimension removes the deterministic χ_K dimension offset; common coordinate scale is the remaining modeling assumption. The main pool fixes $K = 1024$; common coordinate scale remains a modeling assumption. Per-candidate calibration is not a substitute for this assumption. Train-statistic standardization and the empirical radial CDF transform both reduce hard-routing accuracy on every vision evaluation (Appendix N). Mapping each candidate's scores onto its own train statistics removes the cross-candidate comparison that routing reads. On the strongly anisotropic text pool, where the raw chi-norm assumption fails outright, we instead report the quantile transform (Appendix Q).

B.2 LSNG: LOCAL SPECTRAL STRUCTURE FOR SOFT ENSEMBLING

Levy uses one scalar radial statistic. LSNG uses local second-order structure. For candidate c, let $g_c^\star(x)$ be the train-set cluster whose centroid is closest to $z_c(x)$, and let the local covariance eigenvalues in that cluster be

$$\lambda_{g,1} \geq \cdots \geq \lambda_{g,K_c} \geq 0.$$

For tangent rank t and normal rank $r = K_c - t$, with $1 \leq t < K_c$, for strictly positive eigenvalues define the population score

$$\mathrm{LSNG}_c(x) = \frac{1}{t}\sum_{i=1}^{t} \log \lambda_{g_c^\star(x),i} - \frac{1}{r}\sum_{i=t+1}^{K_c} \log \lambda_{g_c^\star(x),i}. \tag{12}$$

For finite-sample analysis, consider the ridge form $\log(\lambda_i + \delta_g)$. The implementation additionally applies covariance shrinkage and an eigenvalue floor (Appendix D). Exact scale invariance is preserved when the ridge scales with the covariance, for example $\delta_g = \epsilon \,\mathrm{tr}(\widehat{\Sigma}_g)/K_c$.

Proposition 5 (Scale invariance and block interpretation)**.** *Consider the no-ridge population score in* (12)*. If all train and query embeddings for a candidate are transformed as $z \mapsto az + b$ with $a > 0$, then nearest-centroid assignments are unchanged and $\mathrm{LSNG}_c(x)$ is unchanged.*

Moreover, if the local covariance decomposes as

$$\Sigma_g = \sigma_T^2 P_T + \sigma_N^2 P_N,$$

where P_T and P_N are orthogonal projectors of ranks t and r, and $\sigma_T^2 > \sigma_N^2 > 0$, then

$$\mathrm{LSNG}_g = \log \sigma_T^2 - \log \sigma_N^2. \tag{13}$$

Thus, in the block-isotropic case, LSNG is the log ratio between tangent and normal geometric variance per coordinate.

Proof. Translation does not change covariance. Scaling by a multiplies the covariance by a^2, so each eigenvalue is multiplied by a^2. The term $\log a^2$ is added to both averages in (12) and cancels. Nearest-centroid assignments are unchanged because all Euclidean distances are multiplied by the same positive factor a. The block formula follows by substituting t eigenvalues equal to σ_T^2 and r eigenvalues equal to σ_N^2. □

The normalization by t and r is the precise sense in which LSNG is dimension-normalized: it compares average log-volume per coordinate, rather than total log-volume. The next bound records that the plug-in estimate is stable when the local covariance estimate is accurate and the log is protected away from zero.

Proposition 6 (Plug-in stability)**.** *Let Σ_g be a local covariance with $\lambda_{\min}(\Sigma_g) \geq m \geq 0$, and let $\widehat{\Sigma}_g$ satisfy*

$$\|\widehat{\Sigma}_g - \Sigma_g\|_{\mathrm{op}} \leq \varepsilon < m + \delta.$$

Define $\mathrm{LSNG}_{t,\delta}$ by replacing each $\log \lambda_i$ in (12) *with $\log(\lambda_i + \delta)$. Then*

$$\left|\mathrm{LSNG}_{t,\delta}(\widehat{\Sigma}_g) - \mathrm{LSNG}_{t,\delta}(\Sigma_g)\right| \leq \frac{2\varepsilon}{m + \delta - \varepsilon}. \tag{14}$$

Proof. By Weyl's inequality, $|\widehat{\lambda}_i - \lambda_i| \leq \varepsilon$ for every eigenvalue. Both $\lambda_i + \delta$ and $\widehat{\lambda}_i + \delta$ are at least $m + \delta - \varepsilon > 0$. On this interval, the derivative of the log is at most $(m + \delta - \varepsilon)^{-1}$. Therefore each log-eigenvalue changes by at most $\varepsilon/(m + \delta - \varepsilon)$. The tangent average and the normal average each change by at most this amount, and the difference of the two averages gives the factor two. □

B.3 A LOCAL SEMANTIC–NOISE MODEL

We now give a local model in which LSNG measures the ratio between semantic tangent volume and nuisance normal volume. All quantities are interpreted inside the nearest train cluster $g_c^\star(x)$.

Theorem 7 (Local semantic–noise decomposition)**.** *Let a local representation have the form*

$$Z = (\alpha(M_Y + \Xi_T),\ \beta\Xi_N) \in \mathbb{R}^t \times \mathbb{R}^r,$$

where Y is a latent semantic state, $M_Y \in \mathbb{R}^t$ is a mean-zero semantic prototype, and

$$\mathrm{Cov}(M_Y) = \sigma_B^2 I_t, \qquad \mathrm{Cov}(\Xi_T) = \tau_T I_t, \qquad \mathrm{Cov}(\Xi_N) = \tau_N I_r.$$

Assume M_Y, Ξ_T, Ξ_N are mutually independent, the noises Ξ_T, Ξ_N are independent of Y, $\mathbb{E}[\Xi_T] = 0$, $\mathbb{E}[\Xi_N] = 0$, and

$$\alpha^2(\sigma_B^2 + \tau_T) > \beta^2\tau_N > 0.$$

Then the local covariance of Z has t leading eigenvalues equal to $\alpha^2(\sigma_B^2 + \tau_T)$ and r trailing eigenvalues equal to $\beta^2\tau_N$. Consequently,

$$L := \mathrm{LSNG} = \log\big(\alpha^2(\sigma_B^2 + \tau_T)\big) - \log(\beta^2\tau_N). \tag{15}$$

Proof. Independence and centering give

$$\mathrm{Cov}(Z) = \alpha^2\,\mathrm{Cov}(M_Y + \Xi_T) \oplus \beta^2\,\mathrm{Cov}(\Xi_N) = \alpha^2(\sigma_B^2 + \tau_T)I_t \oplus \beta^2\tau_N I_r.$$

The assumed inequality orders the eigenvalues, and substituting them into (12) gives (15). □

The same model links LSNG to a simple label-aligned clustering objective. Let

$$D_Y = \mathbb{E}\|Z - \mathbb{E}[Z \mid Y]\|_2^2, \qquad D_1 = \mathbb{E}\|Z - \mathbb{E}Z\|_2^2$$

be the distortion obtained by using the semantic conditional mean and the distortion obtained by using one global mean.

Theorem 8 (Conditional monotonicity of label-aligned clustering gain)**.** *Under the assumptions of Theorem 7,*

$$D_Y = \alpha^2 t\tau_T + \beta^2 r\tau_N, \qquad D_1 - D_Y = \alpha^2 t\sigma_B^2.$$

Hence the relative label-aligned gain

$$\Gamma = \frac{D_1 - D_Y}{D_Y}$$

can be written as

$$\Gamma(L) = \frac{t\sigma_B^2}{t\tau_T + r(\sigma_B^2 + \tau_T)e^{-L}}. \tag{16}$$

For fixed t, r, σ_B^2, τ_T, this is a strictly increasing function of the LSNG value L:

$$\frac{d\Gamma}{dL} = \frac{t\sigma_B^2\, r(\sigma_B^2 + \tau_T)e^{-L}}{\left[t\tau_T + r(\sigma_B^2 + \tau_T)e^{-L}\right]^2} > 0 \tag{17}$$

whenever $\sigma_B^2 > 0$.

Proof. Since $\mathbb{E}[Z \mid Y] = (\alpha M_Y, 0)$,

$$Z - \mathbb{E}[Z \mid Y] = (\alpha\Xi_T, \beta\Xi_N),$$

and therefore

$$D_Y = \alpha^2 t\tau_T + \beta^2 r\tau_N.$$

The law of total covariance gives

$$D_1 - D_Y = \mathrm{tr}\,\mathrm{Cov}(\mathbb{E}[Z \mid Y]) = \alpha^2 t\sigma_B^2.$$

From (15),

$$\beta^2\tau_N = \alpha^2(\sigma_B^2 + \tau_T)e^{-L}.$$

Substituting this identity into D_Y and cancelling the common factor α^2 gives (16). Differentiating gives the displayed positive derivative. □

The theorem assumes a fixed tangent noise profile and does not establish exact label recovery by k-means. In particular, increasing tangent nuisance τ_T can increase the leading eigenvalues without improving semantic separation. In a local block model with fixed tangent noise profile, higher LSNG means lower normal nuisance volume relative to semantic tangent volume, and this increases the relative gain of using local semantic means instead of one global mean. If Y has finitely many states, the optimal k-means objective with $k = |\mathcal{Y}|$ centers can only improve on these label-aligned means, so the gain above is a conservative clusterability certificate under the model.

B.4 Information-geometric view and score-weighted averaging

For a Gaussian block $N(0, \Sigma)$ in dimension d,

$$h(N(0,\Sigma)) = \frac{d}{2}\log(2\pi e) + \frac{1}{2}\log\det\Sigma.$$

Therefore, for tangent and normal covariance blocks,

$$\frac{2}{t}h(N(0,\Sigma_T)) - \frac{2}{r}h(N(0,\Sigma_N)) = \frac{1}{t}\log\det\Sigma_T - \frac{1}{r}\log\det\Sigma_N.$$

Thus LSNG is twice the per-coordinate Gaussian entropy gap between the tangent and normal blocks. Equivalently, it is a normalized local log-volume gap: high LSNG means that the neighborhood has free volume in data-supporting directions and low residual volume in orthogonal directions.

The full-pool rule in Section 3 averages candidate posteriors with score-derived weights. The following identity states the decision-theoretic assumption under which this rule is exactly Bayesian model averaging.

Proposition 9 (Score-weighted posterior mean)**.** *Let C be a latent candidate index. Suppose that, conditional on the observed scores, its posterior is*

$$\mathbb{P}(C = c \mid x, \{s_j(x)\}_{j\in\mathcal{C}}) = \frac{\exp(s_c(x)/\tau)}{\sum_{j\in\mathcal{C}}\exp(s_j(x)/\tau)} = w_c(x), \qquad \tau > 0.$$

Assume also that candidate c supplies a calibrated conditional label distribution $p_c(y \mid x)$ and that

$$p(y \mid x, C = c, \{s_j(x)\}_{j\in\mathcal{C}}) = p_c(y \mid x).$$

Then the Bayes predictive distribution is

$$p(y \mid x, \{s_j(x)\}_{j\in\mathcal{C}}) = \sum_{c\in\mathcal{C}} w_c(x)p_c(y \mid x). \tag{18}$$

This predictive distribution minimizes posterior expected log loss under the assumed model.

Proof. The predictive identity follows from the law of total probability:

$$p(y \mid x, \{s_j\}) = \sum_{c\in\mathcal{C}} p(y \mid x, C = c, \{s_j\})\mathbb{P}(C = c \mid x, \{s_j\}).$$

Substituting the assumed calibrated candidate posterior and the assumed softmax posterior over C gives the formula. For log loss, the posterior risk of a predictive distribution q is $-\sum_y p(y \mid x, \{s_j\})\log q(y)$, which is minimized by setting q equal to the posterior predictive distribution. □

The proposition requires calibrated candidate weights.

Theorems 7–8 justify LSNG as a local structural quality proxy for these weights.

B.5 Summary

The two scores capture different geometric signals. Levy is a radial shell-likelihood term under a centered isotropic Gaussian embedding null. The shell-volume correction explains why Levy differs from ambient Gaussian energy: typical Gaussian mass lies near a shell, not at the origin. Under a radial latent-candidate model, the Bayes hard router is a likelihood-ratio rule whose matched-candidate term is Levy.

LSNG is a local normalized log-volume gap. It is scale-invariant in the population score, stable under small covariance-estimation error, and in a block semantic–noise model equals the log ratio between tangent semantic volume and normal nuisance volume. In the same model, higher LSNG conditionally increases the relative gain of local semantic means over one global mean. These results support the empirical design choice: radial shell typicality for cheap hard routing, and local spectral structure for full-pool score-weighted ensembling.

C EXPERIMENTAL SETUP

C.1 MODEL LIBRARY

The candidate pool contains 8 backbones (Table 2): 2 general-purpose ViT-L (Dosovitskiy et al., 2021) backbones (DINOv2-L (Oquab et al., 2023) and MAE-L (He et al., 2022)) and the 6 experts produced by finetuning each general backbone on Galaxy10, PathMNIST, and Aircraft (Appendix C.3). Both general backbones have embedding dimension $D = 1024$ and 24 transformer blocks. We treat each block as a separate candidate, for $|C| = 8 \times 24 = 192$ candidates.

Table 2: Model library used in all main experiments. Each backbone contributes the class (CLS) token at every one of its 24 transformer blocks, for 192 candidates in total. Patch size is 14 for the DINOv2 family and 16 for the MAE family. Input resolution is 224×224 in both cases.

Backbone	**Family**	**Layers**	**Dim**	**Source**
DINOv2-L	general	24	1024	torch.hub `facebookresearch/dinov2:dinov2_vitl14`
MAE-L	general	24	1024	HuggingFace `facebook/vit-mae-large`
DINOv2-L Galaxy10	expert	24	1024	DINOv2-L finetuned on Galaxy10
DINOv2-L PathMNIST	expert	24	1024	DINOv2-L finetuned on PathMNIST
DINOv2-L Aircraft	expert	24	1024	DINOv2-L finetuned on Aircraft
MAE-L Galaxy10	expert	24	1024	MAE-L finetuned on Galaxy10
MAE-L PathMNIST	expert	24	1024	MAE-L finetuned on PathMNIST
MAE-L Aircraft	expert	24	1024	MAE-L finetuned on Aircraft

C.2 DATASETS

Galaxy10 (10 classes, from Leung & Bovy (2021)) uses the `matthieulel/galaxy10_decals` train/test splits (15,962/1,774 samples). PathMNIST (9 classes, from Yang et al. (2023)) uses its standard train/test splits (89,996/7,180 samples). FGVC-Aircraft (Maji et al., 2013) provides 100 variant classes; we use 6,667 train/validation images for training and 3,333 test images. Galaxy Zoo, used as the *Same-domain unseen* evaluation, is built from the Galaxy Zoo DESI vote fractions (Walmsley et al., 2023) by following the Galaxy Zoo decision tree to assign a composite morphology label per galaxy and dropping classes with fewer than 20 samples. The evaluated subset contains 32 fine-grained morphology classes and 4,321 test images; no expert sees these composite labels at finetuning time. Merged is a virtual concatenation of the Galaxy10, PathMNIST, and Aircraft test sets with per-dataset label remapping ($10 + 9 + 100 = 119$ classes).

C.3 EXPERT FINETUNING

We finetune DINOv2 ViT-L/14 and MAE ViT-L on Galaxy10 (Leung & Bovy, 2021) (10-class galaxy morphology), PathMNIST (Yang et al., 2023) (9-class colon pathology), and Aircraft (Maji et al., 2013) (100-class fine-grained aircraft variants). Finetuning uses the standard ViT recipe: layer-wise learning-rate decay (Bao et al., 2022) ($\lambda \in [0.65, 0.75]$), AdamW (Loshchilov & Hutter, 2019) with cosine schedule (Loshchilov & Hutter, 2017), label smoothing (Szegedy et al., 2016) ($\epsilon = 0.1$), Mixup (Zhang et al., 2018) ($\alpha = 0.8$), RandAugment (Cubuk et al., 2020), and class-weighted cross-entropy. We hold out a stratified 10% split of each training set (seed 42) and pick the checkpoint with the best balanced accuracy on that holdout, with early-stopping patience 10 epochs. The test split is never loaded during finetuning. We save backbone weights only and register the six experts as candidates alongside the two original backbones.

Last-layer gains over the general backbone are $+8.0$ to $+54.8$ pp on DINOv2-L and $+25.7$ to $+50.3$ pp on MAE-L. The largest gap on both backbones is on Aircraft, where the general embedding is far from the fine-grained variant structure. On PathMNIST the MAE expert reaches 94.3% while the best general MAE layer trails the best general DINOv2 layer by 18 pp, so the best expert is not the one a single-backbone heuristic would select.

Table 3: Linear-probe accuracy (%) of finetuned experts against the corresponding general backbones. *Last* uses the final layer; *Best* reports the best layer across all 24. **Bold** marks the best value per row.

		General		Expert	
Dataset	**Backbone**	Last	Best	Last	Best
Galaxy10	DINOv2-L	51.3	56.0	**86.5**	**86.5**
	MAE-L	34.0	36.0	67.5	**67.6**
PathMNIST	DINOv2-L	85.5	88.7	**93.5**	**93.5**
	MAE-L	66.0	70.7	91.7	**94.3**
Aircraft	DINOv2-L	38.7	53.3	**93.5**	**93.5**
	MAE-L	12.2	13.0	**62.5**	**62.5**

D Implementation details

D.1 Embedding extraction

For every candidate (b, l) in the main vision pool, the embedding $z_c(x)$ is the CLS token at the output of block l of backbone b. DINOv2 layer outputs are pulled via `get_intermediate_layers(n=24, return_class_token=True, norm=True)`, so the trained backbone's final LayerNorm is applied to each selected block output. MAE layer outputs are taken from the output of each transformer block before the decoder. All inputs are resized to 224×224 and normalized with each backbone's pretraining mean and standard deviation (read from the HuggingFace `ImageProcessor` when available, ImageNet defaults otherwise).

D.2 Linear probe

We fit a separate linear probe per candidate on cached, ℓ_2-normalized embeddings. The optimizer is stochastic gradient descent (SGD) with momentum 0.9, learning rate 0.5, weight decay 10^{-4}, and cosine annealing over 1000 epochs. Training is full-batch over the cached embedding matrix, with class-weighted cross-entropy to balance classes.

D.3 kNN probe

We use scikit-learn's `KNeighborsClassifier` with $k = 20$ neighbours and cosine distance over ℓ_2-normalized embeddings. The reference set is the same train split as the linear probe.

D.4 LSNG configuration

Unless otherwise specified, LSNG uses the same configuration on every candidate and evaluation, with no per-dataset tuning: $G = 64$ clusters from mini-batch k-means with 100 refinement iterations, tangent rank $t = 8$, an eigenvalue floor of 10^{-4}, and covariance shrinkage of 10^{-2} toward the isotropic covariance with the same trace. Specifically, raw eigenvalues λ_i enter the log as $\max\{(1 - \alpha)\lambda_i + \alpha \operatorname{tr}(\widehat{\Sigma})/D, \epsilon\} + \epsilon$, with $\alpha = 10^{-2}$ and $\epsilon = 10^{-4}$. Appendix I ablates the cheaper variants, which change only the number of k-means iterations, the fit-subset size, and the projection dimension.

Table 4 varies the cluster count and tangent rank around the default. Halving or doubling G moves soft-ensembling accuracy by at most 0.7 pp. Halving t moves it by at most 0.5 pp, and doubling t costs up to 2.0 pp on Galaxy10. We use the same defaults across all evaluations.

Two further settings control the mini-batch k-means fit, not the model: the random seed, which sets the initialization and the mini-batch draws, and the mini-batch size. Table 5 varies each in isolation at the default G=64, t=8. Across 5 seeds, soft-ensembling accuracy moves by at most 0.3 pp on any evaluation. Across mini-batch sizes from 512 to 4096 it moves by at most 0.2 pp. Accuracy is therefore stable across the tested seeds and mini-batch sizes.

Table 4: LSNG sensitivity to the cluster count G and tangent rank t: linear soft-ensembling accuracy per evaluation. G=64, t=8 is the default configuration.

Dataset	G=32, t=8	G=64, t=8	G=128, t=8	G=64, t=4	G=64, t=16
Galaxy10	0.868	0.863	0.857	0.868	0.843
PathMNIST	0.931	0.933	0.940	0.932	0.938
Aircraft	0.930	0.928	0.932	0.930	0.928
Merged	0.929	0.929	0.932	0.925	0.934
Galaxy Zoo	0.386	0.386	0.384	0.387	0.370

Table 5: LSNG sensitivity to the two mini-batch k-means fitting settings at G=64, t=8: linear soft-ensembling accuracy. Seed varies the initialization and mini-batch draws (seeds 0 to 3 and the default 42). Batch varies the mini-batch size over $\{512, 1024, 2048, 4096\}$. The default column (seed 42, batch 4096) is the default configuration.

	Paper	**Seed**		**Batch**	
Dataset	default	min	max	min	max
Galaxy10	0.863	0.861	0.863	0.862	0.864
PathMNIST	0.933	0.933	0.935	0.933	0.935
Aircraft	0.928	0.928	0.929	0.928	0.928
Merged	0.929	0.929	0.932	0.928	0.930
Galaxy Zoo	0.386	0.385	0.386	0.385	0.387

D.5 Bootstrap and seeds

Accuracy standard deviations use 1000 test-set bootstrap resamples. Gain error bars use paired resamples; the standalone accuracy summaries in Table 22 use method-specific bootstrap streams. All random number generators are seeded from a single global seed (default 42): probe initialization, k-means refinement in LSNG, and bootstrap resampling.

E Compute

E.1 Hardware

All experiments run on a workstation with 8 NVIDIA GeForce RTX 3090 GPUs (24 GB each), inside a PyTorch 2.6 + CUDA 12.6 container. The estimated computational cost for the main 8-backbone study is approximately 100 GPU-hours, covering one-time feature extraction across the eight backbones, expert finetuning, probe fitting across all 5 evaluations, and scoring with bootstrap aggregation. Code will be released after acceptance. Figure 5 and the timing tables report an 8-backbone computational-cost evaluation, using different expert checkpoints from the main accuracy evaluations. Linear and kNN totals include fitting and evaluation across the pool; scorer totals include fitting and scoring. These normalized pipeline costs are not online inference latencies. Table 6 reports the normalized component times for every component on every evaluation. Single-probe and full-pool training costs are compared in Appendix E.2 below.

E.2 Training cost: single probe vs. full pool

Table 6 normalizes component totals by test-set size. We separately report the one-time costs of extracting features and fitting probes. The fixed-backbone baseline trains a single linear probe on top of frozen features; per-sample routing trains $|C|$ probes on top of the same features. Table 7 reports both points side by side.

The mean per-candidate time to fit and evaluate a linear probe ranges from 1.8 s to 29.1 s across evaluations, and fitting the full $|C| = 192$-probe pool takes 5.9 min to 1.55 h on the same hardware. Feature extraction is a separate one-time cost for the backbones used by each method; it does not

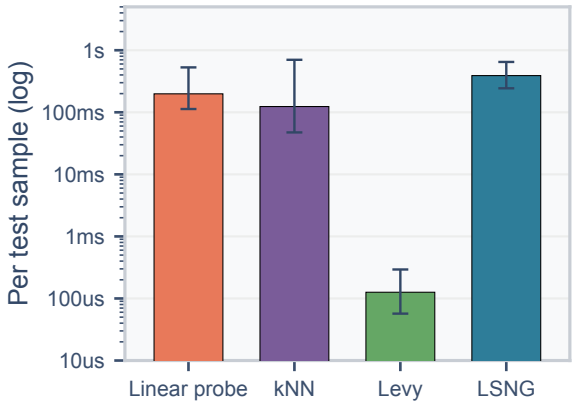

Figure 5: 8-backbone computational-cost evaluation: component fit-and-evaluation totals divided by test-set size, log scale. Bars are the median across the 5 evaluations, and error bars are the min-max range.

Table 6: 8-backbone computational-cost evaluation: normalized component fit-and-evaluation time (milliseconds) on every evaluation. Times are total component seconds divided by N_{test}. The LSNG column reports the original implementation, which uses singular value decomposition (SVD). See Appendix I for cheaper LSNG variants.

Dataset	Linear probe	kNN	Levy	LSNG (Original)
Galaxy10	198	107	0.293	609
PathMNIST	529	615	0.117	648
Aircraft	135	47	0.126	244
Merged	455	701	0.057	390
Galaxy Zoo	113	124	0.167	283

dominate full-pool probe fitting in these timings. Probe fitting can be parallelized across candidates.[1] The full $|C|$-probe pool is shared across all routing rules in the paper: Levy hard, LSNG soft, plain ensembling, plain voting, and the per-sample max-confidence baseline of Appendix M all read from the same pool, so the numbers in Table 7 describe a reusable probe pool shared by these operators.

Table 7: Training-cost summary for an 8-backbone computational-cost evaluation. **Extract** is the wall-clock to extract per-layer features for every backbone in the pool (one-time, for methods using the full pool). **1 probe** is the wall-clock to fit and evaluate a single linear probe on those features (per-candidate, mean±standard deviation across the 192 candidates). **All probes ($|C|$)** is the total wall-clock to fit and evaluate every probe in the pool. The Merged Extract value is a cache-hit number; see footnote in the text.

Eval.	Extract	1 probe (mean ± std)	All probes ($\|C\|$)	Pool size
Galaxy10	3.1 min	1.8 s ± 247 ms	5.9 min	192
PathMNIST	31.0 min	19.8 s ± 11.5 s	1.06 h	192
Aircraft	10.2 min	2.3 s ± 1.0 s	7.5 min	192
Merged	cached	29.1 s ± 10.4 s	1.55 h	192
Galaxy Zoo	4.2 min	2.5 s ± 158 ms	8.1 min	192

F PER-CANDIDATE CONTRIBUTION

We quantify how routing weight and candidate correctness combine for each (backbone, layer) pair under both routing rules.

[1]The Merged feature-extraction value (4 ms) reflects a cache-hit run that reuses embeddings already extracted for the constituent datasets; on a cold cache the wall-clock would match the sum across the three constituents.

F.1 LSNG SOFT ENSEMBLING

For LSNG soft ensembling, the per-candidate contribution is the product of mean softmax weight and weighted slice accuracy, $\bar{w}_c \cdot \alpha_c$. Here $\bar{w}_c = (1/N)\sum_i w_{c,i}$ is the average weight assigned to candidate c across the test set, and $\alpha_c = \sum_i w_{c,i}\,\mathbf{1}[\hat{y}_{c,i} = y_i]/\sum_i w_{c,i}$ is its accuracy on the LSNG-weighted samples. The product is large only when LSNG both uses the candidate and the candidate is correct on its weighted slice.

The bar values in Figure 3 are in percentage points and are small in absolute terms (0.5 to 1.5 on most evaluations) by construction: with $|C| = 192$ and softmax weights summing to one per sample, the average $\bar{w}_c$ across candidates is $1/|C| \approx 0.005$. Candidates that LSNG concentrates on exceed this baseline, and candidates LSNG ignores fall below it. The values do not sum to the soft-ensembling accuracy in Figure 2, because the ensemble's argmax benefits from cross-candidate variance reduction that no single $\bar{w}_c \cdot \alpha_c$ captures. The figure shows which candidates carry the soft weight, not how much each contributes additively to the ensemble accuracy.

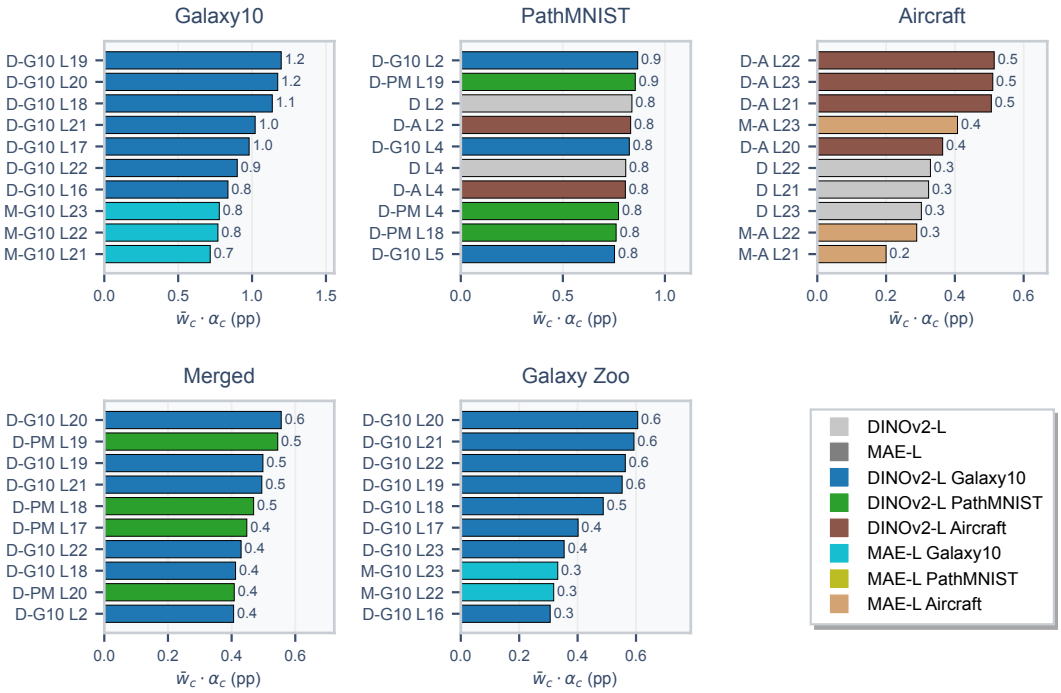

Figure 6: LSNG soft ensembling per evaluation: top-10 (backbone, layer) candidates by per-candidate contribution $\bar{w}_c \cdot \alpha_c$ (in pp). The panels cover the 3 Single-task evaluations, Merged, and Galaxy Zoo. Bar color encodes the backbone family; codes follow Figure 3.

F.2 LEVY HARD ROUTING

Under hard routing each test sample is sent to one candidate. The per-candidate contribution is the fraction of test samples that this candidate received and classified correctly, expressed in percentage points. Summed over all candidates, these contributions equal the overall hard-routing accuracy; the figure displays the top contributors.

On every per-task evaluation the per-sample argmax concentrates on a small number of late-layer candidates (Figure 7), in line with the LSNG picture in Figure 6. On Galaxy Zoo and on the merged multi-task evaluation the top contributors are the matching expert family. On the per-task evaluations the chosen candidate is not always the matching expert by name. The score therefore concentrates on a small set of candidates that are competent on the routed slice, which is consistent with the radial-router interpretation of Levy (Proposition 3).

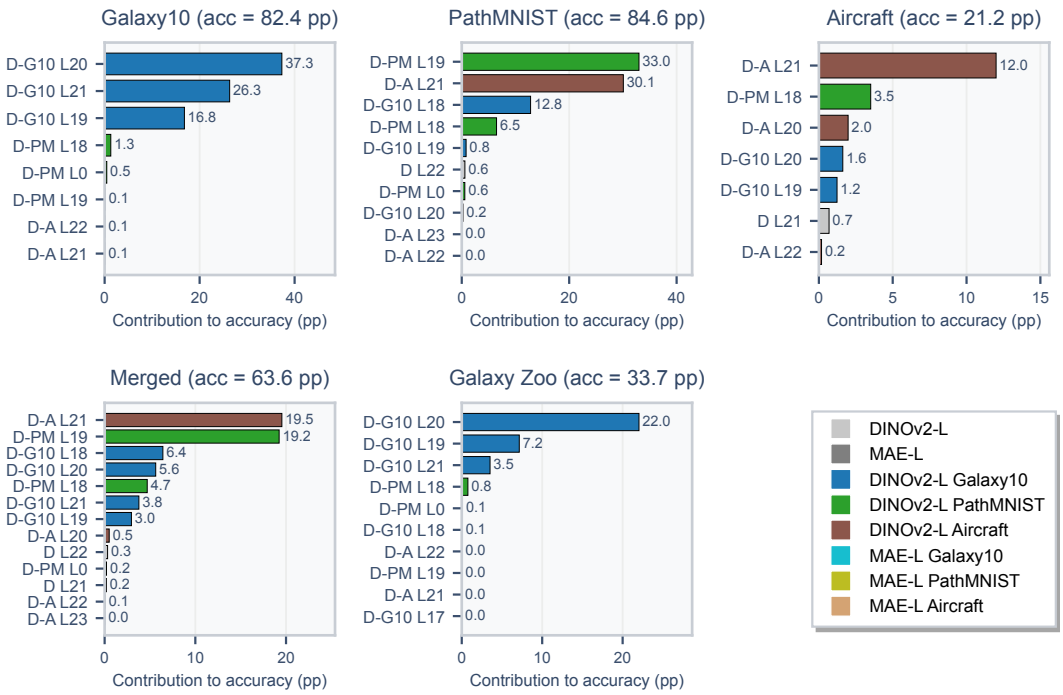

Figure 7: Levy hard routing per evaluation: per-candidate contribution to routing accuracy (selection frequency $\times$ slice accuracy, in pp). Contributions summed over all candidates equal the overall Levy hard routing accuracy; the figure shows the top contributors. Bar color encodes the backbone family; codes follow Figure 3.

G Robustness to pool corruption

We characterize how each routing rule responds when the candidate pool is corrupted by randomly initialized backbones. We inject $k \in \{1, 2, 4\}$ random-initialized DINOv2 ViT-L/14 backbones (full backbone reset, 24 layers each) into the clean 8-backbone pool, raising the candidate count from 192 to $192 + 24k$. Table 8 reports merged accuracy across the two routing rules and the plain-ensemble reference. The two rules respond differently to the same corruption.

Table 8: Pool-corruption robustness on the random-init pool. Random-init candidates are full-backbone weight resets of DINOv2 ViT-L/14 added on top of the clean 192-candidate Merged pool. Routing rules: Levy hard sends each sample to one candidate, LSNG soft weights every candidate's posterior by the softmax of its LSNG score. Plain ensembling (uniform posterior mean over the same pool) is reported as a reference. **Bold** marks the best value per row.

Eval.	Pool	Levy hard	LSNG soft	Plain ens.
Merged	Clean (192 cand.)	0.636	0.929	**0.937**
	+1 random (216)	0.316	0.862	**0.931**
	+2 random (240)	0.316	0.790	**0.910**
	+4 random (288)	0.316	0.707	**0.851**

G.1 Why Levy hard routing collapses on the corrupted pools

Random-init candidates can have norms close to the mode of the Chi reference without providing useful representations. In this experiment, they receive 99.5% of hard-routing decisions. Proposition 3 identifies Levy as a matched radial likelihood term under its assumptions; a high radial score alone does not establish semantic competence or probe quality.

G.2 Why LSNG soft ensembling absorbs the corruption

Theorem 7 decomposes LSNG into total tangent variation, including signal and nuisance, relative to normal variation. This distinguishes its spectral criterion from radial typicality, but does not guarantee that random backbones receive low scores. The observed mass allocation below measures their treatment directly. The measured down-weighting is nonetheless mild: LSNG places 0.101, 0.175, and 0.286 of its soft mass on the random candidates against neutral shares of 0.111, 0.200, and 0.333, so most of the tolerance is the dilution that any posterior average provides. Plain ensembling, which is pure dilution, degrades less still (Table 8).

G.3 Mass placed on the random candidates

The asymmetry between the two routers reduces to one quantity, the share of routing mass that the score places on the injected random candidates. A neutral score would spread its mass uniformly over the $192 + 24N$ candidates, so the $24N$ injected random candidates would jointly receive the share $N/(8 + N)$, which is 0.111, 0.200, and 0.333 at $N{=}1, 2, 4$. Levy hard routing instead sends 0.995 of its argmax decisions to a random backbone at every injection size, accompanied by a 32 pp accuracy decrease. LSNG places 0.101, 0.175, and 0.286 of its soft mass on the random candidates at temperature one across the same three pool sizes, slightly below the neutral expectation in every case. The remaining mass sits on the 192 candidates from the eight pretrained or finetuned backbones. Figure 8 reports the curves across pool sizes and temperatures.

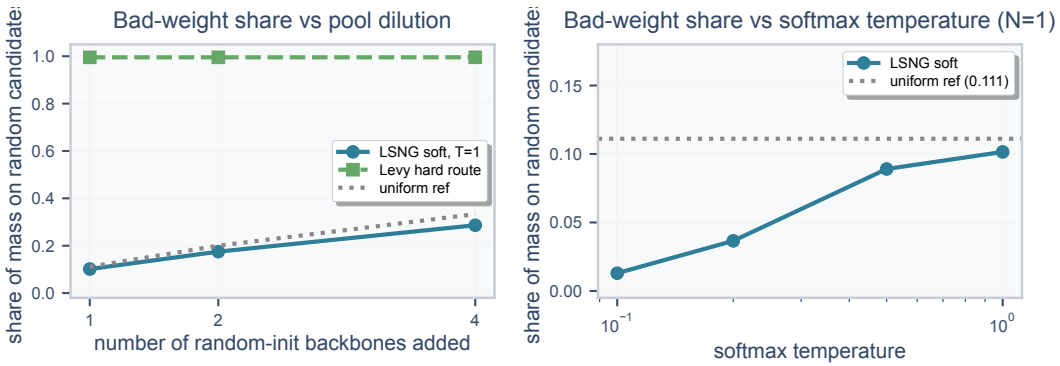

Figure 8: Share of routing mass placed on random-init candidates on the Merged set. The dotted gray line is the neutral expectation $N/(8 + N)$, the joint share assigned to the random candidates by uniform weighting. Left panel varies the number of random backbones added at fixed temperature one. Right panel varies the softmax temperature at fixed $N = 1$.

G.4 Weight-perturbed pool

A second corruption setting adds weight-perturbed copies of every existing backbone, simulating noise on the candidate pool itself rather than an unvetted backbone. We add one mild copy of each backbone (Gaussian weight noise $\sigma = 0.01$, pool size from 192 to 384) and one further harsh copy of each ($\sigma = 0.05$, pool size from 384 to 576). Table 9 reports the two routing rules across evaluations. Both rules remain within one bootstrap standard deviation as perturbed copies are added: Levy hard

Table 9: Weight-perturbed-pool experiment on Galaxy10. Each pool is the clean 8-backbone pool plus one mild ($\sigma = 0.01$) and/or one harsh ($\sigma = 0.05$) weight-perturbed copy of each backbone. Plain ensembling (uniform posterior mean over the same pool) is reported as a reference. **Bold** marks the best value per row.

Eval.	Pool	Levy hard	LSNG soft	Plain ens.
Galaxy10	Clean (192 cand.)	0.824	**0.863**	0.843
	+mild perturb (384 cand.)	0.822	**0.861**	0.845
	+all perturb (576 cand.)	0.822	**0.861**	0.848

moves from 0.824 to 0.822 and LSNG soft from 0.863 to 0.861 while the pool triples from 192 to 576 candidates. Plain ensembling is equally stable (0.843 to 0.848). Proposition 6 gives a related conditional result: LSNG is stable when the local covariance error is small. The table measures prediction stability under the tested weight perturbations.

G.5 Routing cost and robustness

Levy hard routing matches DINOv2-L's 1-probe-per-sample cost. LSNG soft ensembling pays the full $|C|$-probe cost. Both rules still evaluate all backbones. In these corruption experiments, soft ensembling is more robust than hard routing. A pool enlarged with mildly noisier copies of existing backbones leaves both routers stable. The matched comparison to plain voting and plain ensembling on the clean pool is in Appendix M.

G.6 Sensitivity to probe corruption

Both Levy and LSNG score only a candidate's embedding geometry; neither observes the attached linear probe. We test this limitation directly by attaching bad probes to real embeddings. Because candidates with identical embeddings receive identical scores, the scores cannot distinguish their probe quality.

A bad-probe candidate simulates sharp predictions at random classes; no head is actually retrained on shuffled labels. We synthesize K bad-probe twins per real backbone without retraining. Each twin reuses the per-sample score of a randomly chosen real layer of the same backbone (so the score sees real embedding geometry by construction) and outputs a softened one-hot prediction at a uniformly random class per sample. We sweep $K \in \{0, 1, 2, 4, 8\}$, which adds up to 64 bad candidates on top of the 192 real ones. Table 10 reports soft accuracy at $K{=}0$ and $K{=}8$ on three evaluations together with the mean weight LSNG places on bad-probe candidates; *cluster* denotes the original LSNG implementation and *lite* the reduced variant, both defined in Appendix I. On Galaxy10 the accuracy decreases are 0.7 pp for LSNG cluster and 1.3 pp for lite (from 0.863 to 0.856 for cluster, from 0.851 to 0.838 for lite), beating uniform soft (from 0.843 to 0.816) at every K. On Merged and Aircraft both LSNG and uniform degrade together, with LSNG cluster placing 0.235 to 0.260 of its mass on bad-probe candidates at $K{=}8$ across the three evaluations, near the neutral expectation of 0.250.

Table 10: Bad-probe injection on three evaluations. Soft accuracy at $K{=}0$ and $K{=}8$ for uniform soft averaging, LSNG cluster soft, and LSNG lite soft. The last column reports the mean weight LSNG cluster places on bad-probe candidates at $K{=}8$, with the neutral expectation $64/256{=}0.250$ in parentheses. **Bold** marks the best value per row.

Eval.	K	Uniform	LSNG cluster	LSNG lite	LSNG bad-weight at $K{=}8$
Galaxy10	0	0.843	**0.863**	0.851	0.249 (0.250)
	8	0.816	**0.856**	0.838	
Merged	0	**0.937**	0.929	0.933	0.235 (0.250)
	8	**0.782**	0.679	0.736	
Aircraft	0	**0.930**	0.928	0.928	0.260 (0.250)
	8	**0.454**	0.403	0.314	

The bad twins inherit the per-sample score of their real source layer by construction, so the softmax divides mass equally among candidates with equal scores. Their observed total mass is near the neutral expectation in this experiment; equal scores alone do not imply uniform total mass across groups. Probe-quality measures such as held-out accuracy provide information distinct from the embedding-geometry score.

H Class-balanced metrics

Accuracy can be misleading on class-imbalanced datasets. Table 11 reports class-balanced and per-class metrics for the fixed-backbone default DINOv2-L last layer, Levy hard routing, and LSNG soft ensembling on every evaluation. Balanced accuracy is the average of per-class recall. Macro

F1 is the unweighted mean of per-class F1. Worst-class accuracy is the recall on the lowest-recall class. Recall standard deviation captures the spread of per-class recall and is a direct measure of how unevenly the classifier distributes its skill across classes. LSNG soft ensembling improves balanced accuracy over DINOv2-L last layer by $+8.2$ to $+54.1$ pp on every evaluation. Macro F1 improves on every evaluation; worst-class recall improves on four and remains zero on Galaxy Zoo. On Galaxy10, LSNG soft ensembling raises worst-class recall from 0.198 to 0.615 ($+41.7$ pp) and cuts the per-class recall standard deviation from 0.199 to 0.104. On Aircraft, where DINOv2-L has zero recall on the worst-recall class, LSNG soft ensembling raises worst-class recall to 0.424 and cuts the per-class recall standard deviation from 0.321 to 0.094. The routing improvement is therefore not driven by majority classes alone. On the imbalanced merged evaluation (119 classes from three sub-tasks of unequal size), DINOv2-L last layer has zero recall on the worst-recall class and balanced accuracy of 0.399. Levy hard routing raises top-1 accuracy but lowers balanced accuracy to 0.296, consistent with its off-domain decoy failure (Appendix N). LSNG soft ensembling raises balanced accuracy to 0.910 and worst-class recall to 0.455, exceeding even the per-dataset oracle on every reported class-balanced and per-class metric.

Table 11: Classification and per-class metrics on each evaluation. **Acc**: top-1 accuracy. **Bal**: balanced accuracy (average per-class recall). **F1**: macro F1. **wF1**: weighted F1. **Worst**: worst-class recall. **rStd**: per-class recall standard deviation (lower means more even recall). **Bold** marks the best value per (dataset, metric) across the four methods (largest for every metric except rStd, where bold marks the smallest). The Oracle row is the metrics of the best single (backbone, layer) candidate by accuracy.

Dataset	Method	Acc	Bal	F1	wF1	Worst	rStd
Galaxy10	DINOv2-L last layer	0.513	0.515	0.476	0.498	0.198	0.199
	Levy hard routing	0.824	0.814	0.802	0.824	**0.637**	0.113
	LSNG soft ens.	0.863	0.852	0.843	0.863	0.615	0.104
	Oracle	**0.865**	**0.853**	**0.848**	**0.866**	0.624	**0.099**
PathMNIST	DINOv2-L last layer	0.855	0.821	0.815	0.856	0.523	0.144
	Levy hard routing	0.846	0.822	0.809	0.847	0.580	0.130
	LSNG soft ens.	0.933	0.903	0.907	0.930	0.544	0.135
	Oracle	**0.943**	**0.921**	**0.919**	**0.942**	**0.620**	**0.110**
Aircraft	DINOv2-L last layer	0.387	0.387	0.379	0.364	0.000	0.321
	Levy hard routing	0.212	0.212	0.216	0.216	0.000	0.169
	LSNG soft ens.	0.928	0.928	0.929	0.929	**0.424**	0.094
	Oracle	**0.935**	**0.935**	**0.935**	**0.935**	**0.424**	**0.093**
Merged	DINOv2-L last layer	0.602	0.399	0.393	0.582	0.000	0.328
	Levy hard routing	0.636	0.296	0.298	0.633	0.000	0.265
	LSNG soft ens.	**0.929**	**0.910**	**0.912**	**0.928**	**0.455**	**0.100**
	Oracle	0.784	0.844	0.837	0.776	0.073	0.194
Galaxy Zoo	DINOv2-L last layer	0.155	0.164	0.079	0.128	0.000	**0.224**
	Levy hard routing	0.337	0.271	0.198	0.350	0.000	0.239
	LSNG soft ens.	**0.386**	0.310	**0.230**	**0.399**	0.000	0.255
	Oracle	0.376	**0.321**	0.228	**0.399**	0.000	0.240

I FAST-LSNG AND LSNG-LITE ABLATION

The LSNG soft ensembling numbers in Figure 2 use the original LSNG implementation, which fits each per-cluster covariance by SVD and refines the k-means clustering for 100 iterations. We compare 4 lower-cost variants that preserve the routing rule and configuration and trade scoring time for accuracy. The first variant, *Exact*, replaces the per-cluster SVD with the Gram-matrix eigendecomposition. When the cluster size m is at most the embedding dimension D, eigendecomposing the (m, m) Gram matrix recovers the same non-zero covariance eigenvalues exactly, and the missing $D - m$ eigenvalues are exactly zero. The variant is therefore mathematically identical to the SVD path. The second and third variants, *Fast-20* and *Fast-10*, keep the Gram trick and reduce the k-means refinement to 20 and 10 iterations respectively. The fourth variant, *Lite*, keeps the Gram trick and 10 k-means iterations but also fits on a deterministic 4096-sample train subset and a deterministic Gaussian random projection of the embeddings to 256 dimensions before clustering and scoring.

Table 12 reports the linear soft ensembling accuracy of each variant on every evaluation, Table 13 reports normalized fit-and-score times, and Figure 9 summarizes the accuracy and timing measurements.

Table 12: Fast-LSNG accuracy ablation: linear soft ensembling accuracy per evaluation. *Exact* is mathematically identical to the original SVD path. *Fast-20* and *Fast-10* reduce k-means refinement to 20 and 10 iterations. *Lite* adds a 4096-sample fit subset and a 256-dimensional Gaussian random projection. **Bold** marks the best value per row.

Eval	Original	Exact	Fast-20	Fast-10	Lite
Galaxy10	0.863	0.863	**0.864**	0.863	0.851
PathMNIST	0.933	0.933	0.933	0.934	**0.938**
Aircraft	**0.928**	**0.928**	**0.928**	**0.928**	**0.928**
Merged	0.929	0.929	0.930	0.929	**0.933**
Galaxy Zoo	0.386	0.386	**0.387**	**0.387**	0.382

Table 13: Fast-LSNG timing ablation on an 8-backbone computational-cost evaluation: LSNG fit-and-score time divided by test-set size (milliseconds) on every evaluation, computed as total component seconds over the 192-candidate pool divided by N_{test} to match Table 6, and the median per-evaluation speedup over the original. **Bold** marks the best speedup.

Eval	Original	Exact	Fast-20	Fast-10	Lite
Galaxy10	609	426	170	117	48
PathMNIST	648	437	379	355	41
Aircraft	244	198	62	41	23
Merged	390	252	219	207	23
Galaxy Zoo	283	204	67	49	21
Median speedup	1.0×	1.4×	3.6×	5.2×	**13.4**×

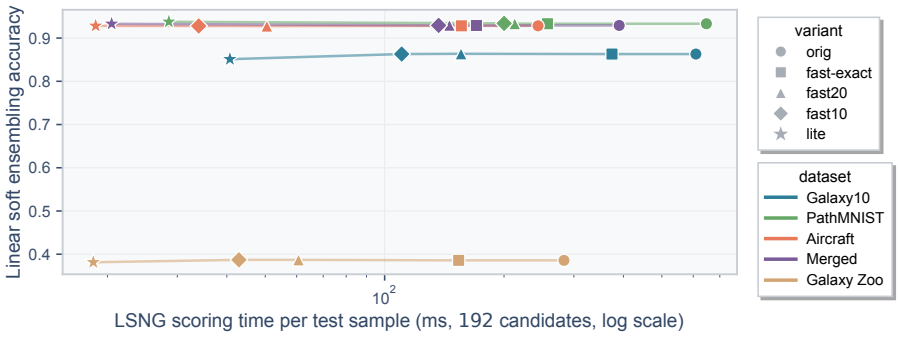

Figure 9: Fast-LSNG accuracy and timing summary. Each color is one evaluation; markers identify the variants of Table 12. Timing includes fitting and scoring; the timing and accuracy evaluations use different expert checkpoints, so these are not paired measurements from the same runs.

Exact matches the SVD implementation accuracies on every evaluation while reducing scoring time by 1.2 to 1.5×. Fast-20 and Fast-10 retain or improve accuracy while reducing time by 1.7 to 6.0×. Lite reduces time by 10.5 to 17.0×, with accuracy changes of −1.2 pp on Galaxy10, −0.4 pp on Galaxy Zoo, zero on Aircraft, +0.4 pp on Merged, and +0.5 pp on PathMNIST.

Fast-10 preserves accuracy while reducing fit-and-score cost in the computational-cost evaluation. Lite has the lowest fit-and-score cost and the highest PathMNIST and Merged accuracies among the tested LSNG variants.

J TEMPERATURE AND TOP-K PEAKING ABLATION

The score-weighted soft-ensembling rule of (5) has two control parameters: the softmax temperature τ in (4), and a top-k mask that restricts averaging to the per-sample k highest-scoring candidates. The LSNG soft results in Figure 2 fix $\tau = 1$ and $k = |C|$. Hard routing is invariant to τ, so the ablation concerns the soft rule only.

We restrict the temperature sweep to $\tau \in (0, 1]$. The opposite direction $\tau > 1$ flattens the softmax toward the uniform distribution and the rule degenerates to plain ensembling, which is already analysed in Appendix M; the $\tau \to 0^+$ limit recovers hard routing. The top-k sweep covers $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 192\}$ at $\tau = 1$ and is reported here for LSNG. Appendix K reports the corresponding Levy comparison, whose soft accuracies plateau at the tested $k \geq 4$ values. We report both standard probe heads in parallel: a linear classifier and a k-nearest-neighbours classifier ($k = 20$, cosine distance) over the same per-candidate embeddings.

J.1 TEMPERATURE ABLATION

Table 14 reports soft-ensembling accuracy as a function of τ at $k = |C|$ for both scorers and both heads.

Table 14: Soft-ensembling accuracy under the temperature sweep $\tau \in (0, 1]$ at $k = |C| = 192$, for the linear and kNN probe heads. The LSNG $\tau = 1$ column matches Figure 2; $\tau \to 0^+$ recovers hard routing. **Bold** marks the best value per row.

Scorer	Head	Dataset	τ=0.05	τ=0.1	τ=0.25	τ=0.5	τ=0.75	τ=1
Levy	linear	Galaxy10	0.823	0.824	0.826	0.829	0.833	**0.835**
		PathMNIST	0.847	0.848	0.850	0.853	0.856	**0.859**
		Aircraft	0.222	0.233	0.258	0.294	0.339	**0.378**
		Merged	0.639	0.642	0.653	0.666	0.683	**0.698**
		Galaxy Zoo	0.338	0.338	0.340	0.342	0.344	**0.346**
	kNN	Galaxy10	0.829	0.829	0.831	0.834	0.835	**0.839**
		PathMNIST	0.869	0.869	0.871	0.873	0.876	**0.878**
		Aircraft	0.239	0.244	0.266	0.292	0.314	**0.331**
		Merged	0.691	0.693	0.700	0.709	0.717	**0.723**
		Galaxy Zoo	0.470	0.471	0.471	0.473	0.472	**0.474**
LSNG	linear	Galaxy10	0.802	0.835	0.858	**0.868**	0.865	0.863
		PathMNIST	0.631	0.688	0.824	0.905	0.928	**0.933**
		Aircraft	0.362	0.616	0.869	0.922	0.927	**0.928**
		Merged	0.434	0.502	0.677	0.851	0.920	**0.929**
		Galaxy Zoo	0.356	0.365	0.379	0.386	**0.389**	0.386
	kNN	Galaxy10	0.790	0.830	**0.856**	0.814	0.753	0.720
		PathMNIST	0.782	0.811	0.852	0.885	0.900	**0.905**
		Aircraft	0.210	0.292	0.416	**0.447**	0.442	0.437
		Merged	0.542	0.575	0.633	0.663	0.682	**0.693**
		Galaxy Zoo	0.465	0.477	**0.505**	0.496	0.476	0.453

The behaviour under Levy is qualitatively identical across heads: the softmax weights concentrate on the per-sample argmax and accuracy generally increases with τ. The spread between $\tau = 0.05$ and $\tau = 1$ reaches $+15.6$ pp on Aircraft (linear), where diffusing even a little mass away from the decoys helps. On Galaxy10 and PathMNIST the spread stays within 1.2 pp on both heads.

LSNG, by contrast, is head-dependent. Under the linear head, accuracy is monotone or near-monotone in τ on every evaluation and the $\tau = 1$ default sits within 0.5 pp of the per-row optimum. Under the kNN head, the optimum sits at intermediate τ on 3/5 evaluations: Galaxy10 ($\tau = 0.25$, $+13.6$ pp over $\tau = 1$), Aircraft ($\tau = 0.5$, $+1.0$ pp), and Galaxy Zoo ($\tau = 0.25$, $+5.2$ pp). These results are consistent with the lower kNN posterior entropy measured in Appendix L. Lowering τ concentrates the routing weights, reducing the contribution of lower-scoring candidates before averaging. Sharpening closes most of the gap to the kNN per-dataset oracle on Galaxy10 (0.856 vs 0.872) and Galaxy Zoo (0.505 vs 0.511); on Aircraft the recovery is partial (0.447 vs 0.936).

J.2 TOP-K ABLATION UNDER LSNG

Table 15 reports LSNG soft accuracy as a function of k at $\tau = 1$, for both heads.

Table 15: LSNG soft-ensembling accuracy under the top-k sweep at $\tau = 1$, for the linear and kNN probe heads. $k = 1$ is LSNG hard routing; $k = 192$ is the full pool of Figure 2. **Bold** marks the best value per row.

Head	Dataset	k=1	k=2	k=4	k=8	k=16	k=32	k=64	k=128	k=192
linear	Galaxy10	0.692	0.737	0.812	0.850	0.863	0.866	**0.867**	0.862	0.863
	PathMNIST	0.583	0.600	0.621	0.649	0.702	0.774	0.917	0.929	**0.933**
	Aircraft	0.111	0.142	0.193	0.303	0.461	0.626	0.752	0.885	**0.928**
	Merged	0.296	0.313	0.352	0.446	0.494	0.553	0.776	0.843	**0.929**
	Galaxy Zoo	0.318	0.328	0.357	0.369	0.383	0.389	**0.391**	0.386	0.386
kNN	Galaxy10	0.745	0.781	0.822	0.856	**0.863**	0.857	0.803	0.739	0.720
	PathMNIST	0.747	0.772	0.797	0.819	0.834	0.851	0.884	0.904	**0.905**
	Aircraft	0.137	0.177	0.216	0.299	0.380	0.429	0.434	**0.443**	0.437
	Merged	0.509	0.529	0.553	0.615	0.631	0.638	0.661	0.681	**0.693**
	Galaxy Zoo	0.439	0.445	0.471	0.492	0.506	**0.509**	0.492	0.466	0.453

The k-curves split along the same head boundary as the τ-curves. Under the linear head, accuracy grows with k up to a plateau on every evaluation and the full pool is optimal or within 0.5 pp of optimal everywhere; the largest argument for soft averaging over the full pool is Aircraft, where moving from $k = 1$ to $k = 192$ gains $+81.7$ pp. Under the kNN head, the curve is non-monotone: the optimum sits at $k = 16$ on Galaxy10 ($+14.3$ pp over the full pool), $k = 128$ on Aircraft ($+0.6$ pp), and $k = 32$ on Galaxy Zoo ($+5.6$ pp). The interpretation is again posterior fragmentation: kNN posteriors are sharp, and once the active candidate set exceeds the band of competent experts, additional candidates contribute confidently wrong vote-quantised mass and the soft argmax flips. Restricting to a moderately sized top-k removes these contributions before they enter the average.

J.3 SENSITIVITY ACROSS PROBE HEADS

The soft rule defaults of the main paper, $\tau = 1$ and $k = |C|$, are robust under the linear head: they fall within 1 pp of the per-row optimum on every evaluation and every scorer, and varying either parameter does not change the qualitative picture in Section 5.2. Under the kNN head, the same defaults are not Pareto-optimal for LSNG; a moderate temperature ($\tau \approx 0.25$) or a moderate top-k ($k \approx 16$ to 32) recovers 5 to 14 pp on Galaxy10 and Galaxy Zoo, where the kNN posterior fragmentation is most pronounced. Both phenomena are consistent with the head analysis of Appendix L: the soft rule integrates classifier confidence implicitly, and the optimal configuration of the rule depends on whether that confidence is dispersed (linear) or concentrated (kNN).

K DECOMPOSING TOP-K SOFT ENSEMBLING: GATING VS. WEIGHTING

The soft-ensembling rule of (5) uses the per-sample score in two ways: a *gate* selects the top-k candidates for a sample, and a *weighting* combines the selected candidates through a softmax. The peaking ablation in Appendix J varies the joint operator over (τ, k) without separating these roles. We isolate them by holding the gate fixed and replacing its softmax weights with uniform $1/k$ weights:

$$\hat{y}_{\text{uni-top-k}}(x) \;=\; \arg\max_{y \in \mathcal{Y}} \; \frac{1}{k} \sum_{c \,\in\, \text{Top}_k(s(x))} p_c(y \mid x), \tag{19}$$

where $\text{Top}_k(s(x))$ is the set of k candidates with the highest per-sample scores $s_c(x)$. The aggregation operator is the same posterior mean used by plain ensembling and LSNG soft ensembling; only the weights differ.

The gap between (19) and plain ensembling (Appendix M, $k = |C|$ with $w_c = 1/|C|$) isolates the per-sample *gate*, because both sides use uniform weights. Comparing (19) with score-weighted top-k from Appendix J instead isolates the softmax *weighting* within the same gate. Plain ensembling

cannot implement a per-sample top-k: equal weights over a fixed pool contain no selection signal. The scores from Levy and LSNG supply that signal.

The boundary cases agree by construction. At $k = 1$, uniform top-k is hard score routing; at $k = |C|$, it is plain ensembling; and as $\tau \to \infty$ it coincides with score-weighted top-k. In Tables 16 and 17, the $k = 1$ columns reproduce their respective hard-routing results, while the $k = 192$ columns reproduce the Ensemble column of Table 22.

K.1 LSNG: LITTLE BENEFIT FROM GATING ALONE

Table 16 compares uniform top-k against score-weighted top-k at $\tau = 1$, both using the same LSNG gate, on every evaluation under the linear head.

Table 16: LSNG top-k ensembling under uniform ($1/k$) and score-weighted (softmax, $\tau = 1$) aggregation, linear probe head. Same gate in both column groups. The Δ_w column is the matched-gate difference of softmax over uniform weighting at $k = 64$.

	Uniform $1/k$					**Softmax** ($\tau = 1$)					Δ_w
Dataset	k=1	k=4	k=16	k=64	k=192	k=1	k=4	k=16	k=64	k=192	at k=64
Galaxy10	0.692	0.815	0.862	0.859	0.843	0.692	0.812	0.863	0.867	0.863	+0.008
PathMNIST	0.583	0.621	0.706	0.933	0.942	0.583	0.621	0.702	0.917	0.933	−0.016
Aircraft	0.111	0.193	0.461	0.754	0.930	0.111	0.193	0.461	0.752	0.928	−0.002
Merged	0.296	0.352	0.494	0.778	0.937	0.296	0.352	0.494	0.776	0.929	−0.002
Galaxy Zoo	0.318	0.358	0.383	0.388	0.369	0.318	0.357	0.383	0.391	0.386	+0.003

Within a fixed gate, softmax and uniform weighting are nearly interchangeable: the matched-gate difference Δ_w stays within 1.6 pp at $k = 64$ and within 2.0 pp across all shown k values. Small LSNG gates inherit the weakness of the LSNG argmax, so accuracy at $k \leq 16$ remains far below the full pool on PathMNIST, Aircraft, and Merged. LSNG's residual gain over plain ensembling comes from full-pool weighting: at $k = 192$, the score-weighted column beats uniform averaging by $+2.0$ pp on Galaxy10 and $+1.7$ pp on Galaxy Zoo and stays within 1 pp elsewhere.

K.2 LEVY: UNIFORM TOP-k AGGREGATION UNDER NORM DECOYS

We apply the same gate–weighting decomposition to Levy (Table 17). Levy's softmax weights at $\tau = 1$ are sharply peaked, so score-weighted top-k remains at the $\tau = 1$ full-pool value for every $k \geq 4$. Uniform averaging over the same gate instead dilutes the few decoy candidates that capture the argmax, and recovers the competent candidates that Levy's ranking places just below them.

Table 17: Levy top-k ensembling under uniform ($1/k$) and score-weighted (softmax, $\tau = 1$) aggregation, linear probe head. The Δ_w column is the matched-gate gain of uniform over softmax weighting at $k = 64$.

	Uniform $1/k$					**Softmax** ($\tau = 1$)					Δ_w
Dataset	k=1	k=4	k=16	k=64	k=192	k=1	k=4	k=16	k=64	k=192	at k=64
Galaxy10	0.824	0.863	0.868	0.859	0.843	0.824	0.835	0.835	0.835	0.835	+0.024
PathMNIST	0.846	0.939	0.938	0.947	0.942	0.846	0.859	0.859	0.859	0.859	+0.088
Aircraft	0.212	0.789	0.924	0.934	0.930	0.212	0.378	0.378	0.378	0.378	+0.556
Merged	0.636	0.878	0.927	0.937	0.937	0.636	0.698	0.698	0.698	0.698	+0.239
Galaxy Zoo	0.337	0.366	0.389	0.381	0.369	0.337	0.346	0.346	0.346	0.346	+0.035

Uniform top-k nearly eliminates Levy's Aircraft failure. The argmax abandons the matching expert on 83% of samples and scores 0.212, but uniform averaging over the Levy top-4 reaches 0.789 and top-64 reaches 0.934, within 0.1 pp of the per-dataset oracle. The decoy layers occupy the highest ranks but are few, so a gate wider than the decoy set admits enough competent candidates to dilute them. This aggregation uses the same Levy ranking, requires k probes per sample, and uses no labels for selection.

K.3 Differences between the scoring rules

LSNG spreads its mass across many candidates, so its softmax at $\tau = 1$ is already a soft mixture and truncating the pool only removes useful posterior mass. Levy has concentrated softmax weights, but its soft and hard predictions differ. Uniform averaging makes greater use of candidates below the top ranks. Both observations are consistent with the pairwise AUC analysis of Section 5.3: Levy's ranking is reliable in aggregate but not at the very top, and LSNG's is reliable in aggregate but too diffuse for a small gate.

K.4 Aggregation across probe budgets

LSNG benefits from weighting the full pool ($\tau = 1$, $k = |C|$). For Levy, uniform top-k with k from 16 to 64 recovers accuracy under norm decoys and approaches or exceeds plain ensembling, using k probes instead of $|C|$. The same operator carries over to the kNN head, where it beats the kNN default on every evaluation (Appendix L).

L kNN heads

The main paper uses a linear probe on each candidate's embedding. The kNN classifier ($k = 20$, cosine distance) is a non-parametric head used in the self-supervised representation literature. To examine sensitivity to the probe head, Table 18 reports every reference and every routing method under both heads.

L.1 Reference points behave the same way under both heads

On every evaluation, the DINOv2-L last-layer default is below the per-dataset oracle, and Random is below both, under both linear and kNN probes. The reference ordering therefore agrees across the two probe choices.

L.2 Routing strategies under kNN

Levy hard routing improves over the kNN DINOv2-L last-layer baseline on 3/5 evaluations ($+1.2$ to $+23.9$ pp); the Aircraft decoy failure persists under this head (-21.3 pp) and Merged loses -1.7 pp. LSNG soft ensembling improves on three of five ($+4.9$ to $+13.0$ pp) and sits below the default on Aircraft (-0.3 pp) and Merged (-1.2 pp). On Aircraft the absolute LSNG soft accuracy drops from 0.928 on linear to 0.437 on kNN. We measure lower posterior entropy under kNN than under linear probes (below). This is consistent with confidently wrong predictions from off-task candidates contributing to the loss under full-pool averaging. Sharpening improves Galaxy10 and Galaxy Zoo substantially, but Aircraft by only 1.0 pp (Appendix J). The matching DINOv2-L Aircraft expert under kNN (0.936) is therefore largely unreachable by the flat soft-ensembling rule on this head. The hard routing rule, which selects one candidate per sample, is unaffected by posterior fragmentation but keeps its own decoy failure.

L.3 Sensitivity to the number of kNN neighbours

The kNN head fixes the neighbour count at k=20, a common default in self-supervised evaluation rather than a tuned value. Table 19 sweeps k from 5 to 100 with the routing rule unchanged. Levy hard stays within 1.2 pp of its k=20 value on the single-task and multi-domain evaluations; only Galaxy Zoo shows a larger swing ($+4.5$ pp from k=5 to k=20) as extra neighbours stabilize the routed generalist. LSNG soft is flat except on Aircraft, where additional neighbours de-fragment the near-one-hot votes and raise accuracy from 0.31 at k=5 to 0.51 at k=50. The k=20 result (0.44) is therefore mid-range rather than a tuned peak. On Aircraft, the fixed-backbone baseline spans 16.3 pp across k, compared with 19.5 pp for LSNG soft and 1.2 pp for Levy hard.

L.4 Posterior entropy and top-k aggregation

We quantify posterior concentration using normalized entropy. The normalized posterior entropy, averaged over candidates and test samples, sits between 0.84 and 0.99 under the linear head and

Table 18: Routing reference points and routing strategies under linear and kNN probe heads. Δ is kNN minus linear accuracy; positive values mean kNN exceeds linear on that row.

Dataset	Method	Linear	kNN	Δ
Galaxy10	Random	0.344	0.431	+0.087
	DINOv2-L last layer	0.513	0.590	+0.077
	Per-dataset oracle	0.865	0.872	+0.007
	Levy hard routing	0.824	0.829	+0.005
	LSNG soft ens.	0.863	0.720	−0.143
PathMNIST	Random	0.664	0.830	+0.166
	DINOv2-L last layer	0.855	0.856	+0.001
	Per-dataset oracle	0.943	0.935	−0.008
	Levy hard routing	0.846	0.868	+0.022
	LSNG soft ens.	0.933	0.905	−0.028
Aircraft	Random	0.099	0.135	+0.036
	DINOv2-L last layer	0.387	0.440	+0.053
	Per-dataset oracle	0.935	0.936	+0.002
	Levy hard routing	0.212	0.227	+0.015
	LSNG soft ens.	0.928	0.437	−0.491
Merged	Random	0.323	0.585	+0.261
	DINOv2-L last layer	0.602	0.705	+0.103
	Per-dataset oracle	0.784	0.840	+0.056
	Levy hard routing	0.636	0.688	+0.052
	LSNG soft ens.	0.929	0.693	−0.236
Galaxy Zoo	Random	0.136	0.252	+0.117
	DINOv2-L last layer	0.155	0.328	+0.173
	Per-dataset oracle	0.376	0.511	+0.135
	Levy hard routing	0.337	0.468	+0.132
	LSNG soft ens.	0.386	0.453	+0.067

Table 19: Sensitivity to the number of kNN neighbours k, with the routing rule fixed: accuracy under the kNN head at $k \in \{5, 10, 20, 50, 100\}$. k=20 is the value used throughout. Baseline is the fixed DINOv2-L last layer.

Dataset	Method	k=5	k=10	k=20	k=50	k=100
Galaxy10	DINOv2-L last	0.536	0.573	0.590	0.586	0.579
	Levy hard	0.826	0.830	0.829	0.833	0.830
	LSNG soft	0.719	0.715	0.720	0.719	0.721
PathMNIST	DINOv2-L last	0.854	0.860	0.856	0.854	0.849
	Levy hard	0.865	0.869	0.868	0.866	0.858
	LSNG soft	0.905	0.905	0.905	0.906	0.904
Aircraft	DINOv2-L last	0.480	0.464	0.440	0.369	0.317
	Levy hard	0.217	0.223	0.227	0.229	0.221
	LSNG soft	0.311	0.372	0.437	0.506	0.465
Merged	DINOv2-L last	0.706	0.712	0.705	0.684	0.666
	Levy hard	0.683	0.688	0.688	0.687	0.679
	LSNG soft	0.677	0.686	0.693	0.697	0.688
Galaxy Zoo	DINOv2-L last	0.268	0.300	0.328	0.347	0.348
	Levy hard	0.423	0.451	0.468	0.463	0.468
	LSNG soft	0.438	0.449	0.453	0.455	0.453

drops to 0.19 to 0.62 under kNN. On Merged the mean top-1 posterior mass rises from 0.06 to 0.60, and on PathMNIST 42% of kNN posteriors are exact one-hots. A 20-neighbour vote also places mass on at most 20 classes, so unsmoothed geometric-mean averaging can give zero mass to every class; zero votes require smoothing for log-space aggregation.

We also evaluate uniform top-k aggregation (Appendix K) under kNN heads. Table 20 sweeps uniform averaging over the Levy top-k across the full k range. Every tested k from 4 to 64 beats the kNN fixed-backbone default on every evaluation, showing gains across multiple probe budgets. At $k = 16$ the operator reaches 0.866 on Galaxy10, 0.933 on PathMNIST, 0.912 on Aircraft, 0.917 on Merged, and 0.519 on Galaxy Zoo. This is within 2.5 pp of the kNN per-dataset oracle on the 3 Single-task evaluations and above that oracle on Merged and Galaxy Zoo. At $k = 16$, the gate also beats plain kNN ensembling ($k = 192$) on every evaluation, by up to $+50$ pp on Aircraft, because it excludes most of the quantized wrong votes before averaging. These results support using the Levy ranking to select a subset of kNN predictions before averaging.

Table 20: Uniform averaging over the Levy top-k under the kNN head. $k = 1$ is Levy hard routing and $k = 192$ is plain kNN ensembling, matching Tables 18 and 22. **Bold** marks the best value per row.

Dataset	k=1	k=2	k=4	k=8	k=16	k=32	k=64	k=128	k=192
Galaxy10	0.829	0.855	0.863	**0.872**	0.866	0.835	0.754	0.629	0.614
PathMNIST	0.868	0.912	0.934	**0.937**	0.933	0.930	0.934	0.913	0.912
Aircraft	0.227	0.392	0.787	0.908	0.912	**0.916**	0.811	0.434	0.409
Merged	0.688	0.763	0.882	**0.919**	0.917	0.911	0.876	0.743	0.732
Galaxy Zoo	0.468	0.484	0.499	0.513	**0.519**	0.506	0.466	0.396	0.387

L.5 Per-backbone breakdown

Table 21 reports the best-layer accuracy under both heads for every backbone on every evaluation. kNN improves many cross-domain candidates, consistent with the reference-point gains in Table 18.

Table 21: Best-layer accuracy under linear-probe and kNN heads, per backbone, on every evaluation. **Bold** marks the per-dataset oracle. The bottom row repeats the per-dataset oracles from Table 18.

	Galaxy10		**PathMNIST**		**Aircraft**		**Merged**		**Galaxy Zoo**	
Backbone	lin	kNN	lin	kNN	lin	kNN	lin	kNN	lin	kNN
DINOv2-L (general)	0.560	0.603	0.887	0.903	0.533	0.600	0.679	0.766	0.166	0.343
DINOv2-L Galaxy10 expert	0.865	0.872	0.819	0.882	0.136	0.191	0.535	0.666	0.376	0.511
DINOv2-L PathMNIST expert	0.478	0.545	0.935	0.933	0.078	0.116	0.608	0.641	0.134	0.292
DINOv2-L Aircraft expert	0.564	0.621	0.887	0.904	0.935	0.936	0.784	0.840	0.162	0.352
MAE-L (general)	0.360	0.449	0.707	0.863	0.130	0.185	0.410	0.617	0.142	0.261
MAE-L Galaxy10 expert	0.676	0.692	0.707	0.866	0.130	0.193	0.477	0.642	0.289	0.394
MAE-L PathMNIST expert	0.391	0.448	0.943	0.935	0.117	0.167	0.617	0.649	0.138	0.277
MAE-L Aircraft expert	0.393	0.453	0.729	0.866	0.625	0.608	0.583	0.707	0.141	0.269
Oracle	**0.865**	**0.872**	**0.943**	**0.935**	**0.935**	**0.936**	**0.784**	**0.840**	**0.376**	**0.511**

M Comparison to plain voting and plain ensembling baselines

Two natural no-routing baselines on a pool of pre-trained encoders are plain voting and plain ensembling. Both aggregate the full pool and evaluate $|C|$ probes per sample. **Plain voting**, the decision-level operator, has each candidate cast one vote for its argmax prediction $\hat{y}_c(x)$ and predicts the most-voted class:

$$\hat{y}_{\text{plur}}(x) \;=\; \arg\max_{y \in \mathcal{Y}} \sum_{c \in C} w_c(x) \cdot \mathbf{1}[\hat{y}_c(x) = y], \qquad w_c(x) = 1/|C|. \tag{20}$$

No sample-dependent weighting enters this rule. A score-weighted variant of (20) (Levy-weighted plurality) replaces the uniform $w_c = 1/|C|$ with the softmax weights of (4). We report it alongside Levy hard routing in Table 22. **Plain ensembling** is the soft-ensembling operator of (5) with $w_c = 1/|C|$: every candidate's posterior is averaged with equal weight, and the argmax of the average is the prediction. Classifier confidence enters the prediction implicitly because peaked posteriors contribute more probability mass than flat ones. Plain ensembling is the $\tau \to \infty$ endpoint of the soft rule in

(4). By integrating classifier confidence implicitly, it reaches accuracy comparable to the per-dataset oracle on every linear evaluation. We isolate the contribution of the routing signal alone, holding the operator and pool fixed (the *matched-aggregation* comparison).

M.1 Rescue and regression rates

We compare Levy hard routing with uniform posterior averaging using the conditional rates defined in Section 5.3. On Galaxy10, the rescue rate is 29% and the regression rate is 7%. On PathMNIST, the respective rates are 31% and 12%. On Aircraft, they are 4% and 77%. On Merged, they are 14% and 33%. On the Same-domain unseen evaluation, Galaxy Zoo, they are 15% and 21%. Because rescue and regression condition on different subsets, their effect on overall accuracy depends on the corresponding sample counts.

M.2 Matched-aggregation accuracies

Table 22 compares uniform aggregation with Levy-weighted plurality and LSNG-weighted posterior averaging, for both probe heads. Δ_{vote} is the matched-aggregation gain over plain voting (Levy-weighted plurality minus uniform plurality); Δ_{ens} is the analogous quantity over plain ensembling for the posterior-mean operator.

Table 22: Matched-aggregation comparison: plain voting (uniform plurality), plain ensembling (uniform posterior mean), and per-sample max-confidence (pick the candidate with the highest top-class posterior) against Levy-weighted plurality and LSNG-weighted posterior mean, under the linear and kNN probe heads. Δ_{vote} is the gain over plain voting; Δ_{ens} is the gain over plain ensembling. The *Max-conf* column is a $|C|$-probe routing operator that needs every probe at test time but no separate scoring head: per sample, predict from the candidate whose probe is most confident on that sample. Bootstrap standard deviations (1000 resamples of the test set) are shown as $\pm$ for the scored operators and uniform posterior ensemble. With Levy scores the softmax weights are sharply peaked, so the Levy-weighted plurality column is numerically equal to Levy hard score routing (Figure 2); the LSNG posterior-mean column matches the main LSNG soft ensembling configuration.

		Plurality			**Posterior mean**			**Per-sample**
Head	**Dataset**	Voting	Levy	Δ_{vote}	Ensemble	LSNG	Δ_{ens}	Max-conf
linear	Merged	0.330	$0.636_{\pm 0.004}$	$+0.306$	$0.937_{\pm 0.002}$	$0.929_{\pm 0.002}$	-0.008	$0.925_{\pm 0.002}$
	Galaxy10	0.419	$0.824_{\pm 0.009}$	$+0.405$	$0.843_{\pm 0.009}$	$0.863_{\pm 0.008}$	$+0.020$	$0.868_{\pm 0.008}$
	PathMNIST	0.747	$0.846_{\pm 0.004}$	$+0.099$	$0.942_{\pm 0.003}$	$0.933_{\pm 0.003}$	-0.009	$0.946_{\pm 0.003}$
	Aircraft	0.088	$0.212_{\pm 0.007}$	$+0.124$	$0.930_{\pm 0.004}$	$0.928_{\pm 0.004}$	-0.002	$0.936_{\pm 0.004}$
	Galaxy Zoo	0.137	$0.337_{\pm 0.007}$	$+0.200$	$0.369_{\pm 0.008}$	$0.386_{\pm 0.007}$	$+0.017$	$0.373_{\pm 0.007}$
kNN	Merged	0.669	$0.688_{\pm 0.004}$	$+0.019$	$0.732_{\pm 0.004}$	$0.693_{\pm 0.004}$	-0.039	$0.917_{\pm 0.002}$
	Galaxy10	0.539	$0.829_{\pm 0.009}$	$+0.290$	$0.614_{\pm 0.012}$	$0.720_{\pm 0.010}$	$+0.105$	$0.860_{\pm 0.008}$
	PathMNIST	0.906	$0.868_{\pm 0.004}$	-0.038	$0.912_{\pm 0.003}$	$0.905_{\pm 0.004}$	-0.007	$0.923_{\pm 0.003}$
	Aircraft	0.215	$0.227_{\pm 0.008}$	$+0.012$	$0.409_{\pm 0.008}$	$0.437_{\pm 0.009}$	$+0.028$	$0.937_{\pm 0.004}$
	Galaxy Zoo	0.356	$0.468_{\pm 0.008}$	$+0.112$	$0.387_{\pm 0.008}$	$0.453_{\pm 0.008}$	$+0.066$	$0.497_{\pm 0.008}$

First, under the linear head, the matched-aggregation gain against plain voting is large on every evaluation ($+9.9$ to $+40.5$ pp): plain voting discards probe confidence and votes equally, so without a routing signal the matching expert is one of $|C|$ candidates and is out-voted by the rest. Second, under the linear head, the matched-aggregation gain against plain ensembling is small in magnitude (-0.9 to $+2.0$ pp) on every evaluation. Plain ensembling already provides implicit per-sample weighting through classifier confidence (Section 3.1), leaving little additional gain for LSNG weighting. Third, the kNN head reorders the operators. Plain voting is stronger under kNN because the 20-neighbour vote produces near-one-hot per-candidate posteriors and the cross-candidate plurality concentrates on the modal class. Plain ensembling weakens because averaging quantised posteriors loses the variance-reduction benefit of smooth posteriors (Appendix L). Accordingly, Levy-weighted plurality gains $+1.2$ to $+29.0$ pp over plain voting on 4/5 evaluations but loses 3.8 pp on PathMNIST, where the matching expert is already the modal voter. LSNG-weighted posterior mean gains $+2.8$ to $+10.5$ pp on Galaxy10, Aircraft, and Galaxy Zoo, but loses 0.7 to 3.9 pp on PathMNIST and Merged, where posterior fragmentation dominates the embedding-routing gain.

M.3 PER-SAMPLE MAX-CONFIDENCE AS A NO-SCORER BASELINE

The Max-conf column reports a third $|C|$-probe operator: per sample, route to the candidate whose probe assigns the highest top-class posterior. It needs the same probe pool but no separate scoring head, making it the simplest use of per-sample probe confidence as a routing signal. On the curated clean pool it is strong. Under the linear head it matches LSNG soft to within 1.3 pp on all 5 evaluations, beating it slightly on Galaxy10 ($+0.5$ pp), PathMNIST ($+1.3$ pp), and Aircraft ($+0.8$ pp). Under the kNN head it exceeds the other operators in Table 22 on every evaluation; the top-k comparisons appear separately in Appendix L. The synthetic bad-probe experiment separates the methods (Table 27): max-confidence is particularly vulnerable to confidently wrong predictions, while LSNG degrades less in those runs. Neither geometry score directly tests probe quality (Appendix G.6).

M.4 MATCHED AGGREGATION UNDER POOL PERTURBATION

Table 23 repeats the matched-aggregation comparison on the weight-perturbed Galaxy10 pools of Appendix G. For every pool size and probe head we report plain voting, Levy-weighted plurality, plain ensembling, and LSNG-weighted posterior mean, all on the same pool. This isolates whether the routing signal becomes more or less valuable as the pool is enlarged with weight-perturbed copies of the same backbones.

Table 23: Matched-aggregation comparison on the weight-perturbed pools, under the linear and kNN probe heads. Each row is one pool size for one evaluation: clean (192 candidates, no perturbed copies), +mild (384, +8 copies at $\sigma = 0.01$), and +all (576, +8 copies each at $\sigma \in \{0.01, 0.05\}$). Δ_{vote} is the gain of Levy-weighted plurality over plain voting; Δ_{ens} is the gain of LSNG-weighted posterior mean over plain ensembling.

			Plurality			**Posterior mean**		
Head	**Eval**	**Pool**	Voting	Levy	Δ_{vote}	Ensemble	LSNG	Δ_{ens}
linear	Galaxy10	Clean (192)	0.419	0.824	+0.405	0.843	0.863	+0.020
		+mild (384)	0.420	0.822	+0.402	0.845	0.861	+0.016
		+all (576)	0.421	0.822	+0.401	0.848	0.861	+0.012
kNN	Galaxy10	Clean (192)	0.539	0.829	+0.290	0.614	0.720	+0.105
		+mild (384)	0.534	0.830	+0.296	0.614	0.720	+0.105
		+all (576)	0.528	0.830	+0.302	0.613	0.720	+0.107

Both matched-aggregation gains are stable while the pool triples. Under the linear head, Δ_{vote} stays near $+40$ pp ($+0.405$ to $+0.401$) and Δ_{ens} shrinks from $+2.0$ to $+1.2$ pp as plain ensembling absorbs the extra mildly noisy copies. Under kNN, Δ_{vote} grows slightly (from $+29.0$ to $+30.2$ pp) and Δ_{ens} remains stable (from $+10.5$ to $+10.7$ pp). The latter is the contribution that score weighting adds when plain ensembling cannot resolve quantised kNN posteriors (Appendix L). As on the clean pool, the routing signal is most visible in the plurality comparison.

N WHEN HARD ROUTING FAILS: NORM DECOYS

On the Aircraft evaluation Levy hard routing scores 0.212, below the 0.387 fixed default. The geometry score ranks off-domain expert layers above the in-domain Aircraft expert, so the per-sample argmax leaves the Aircraft expert on 82.9% of samples. Table 24 reports the fraction of samples assigned to each backbone.

An off-domain DINOv2 expert layer outscores every Aircraft-expert layer on 82.8% of samples, at a mean score gap of $+8.01$ in favor of the decoy; this per-sample ranking deficit produces the 82.9% routed-away share above. The Aircraft expert best layer (L23, 0.935 accuracy) has the highest accuracy but a much lower score, and is rarely selected. LSNG soft ensembling reaches 0.928 on the same pool despite the candidates that attract the Levy argmax.

We test 2 per-candidate calibration transforms on the same pool. Standardized and train-quantile Levy, the two transforms suggested for pools without a common coordinate scale (Appendix B), lower Aircraft hard routing from 0.212 to 0.148 and 0.172, and lower every other evaluation as

Table 24: Routed-share breakdown of Levy hard routing on Aircraft (linear head). **Share** is the fraction of test samples the argmax sends to each backbone family. **Decoy layer** is the highest-traffic layer of that family, with its full-test accuracy.

Routed to	Share	Decoy layer	Layer acc
DINOv2 Galaxy10 expert	42.9%	L20	0.060
DINOv2 PathMNIST expert	39.2%	L18	0.077
DINOv2 Aircraft expert	17.1%	L21	0.893
DINOv2-L base	0.8%	L21	0.533

well (Galaxy10 0.824 to 0.388 and 0.429, Merged 0.636 to 0.475 and 0.446). Calibration maps each candidate's scores onto that candidate's own train distribution, which removes the cross-candidate typicality comparison the router reads. These tested calibration transforms do not resolve the decoy failure.

O Additional baselines

LogME selection and per-sample max-confidence match the routed operators on clean, single-domain pools. However, LogME needs labeled training data for each task, and max-confidence collapses when the pool contains a corrupted probe.

Table 25: Per-dataset selection. LogME (You et al., 2021) picks one candidate per task from labeled train embeddings; we report the selected candidate's test accuracy against the per-dataset oracle and LSNG soft ensembling.

Eval	LogME selection	Oracle single	LSNG soft
Galaxy10	0.865	0.865	0.863
PathMNIST	0.933	0.943	0.933
Aircraft	0.935	0.935	0.928
Merged	0.755	0.784	0.929
Galaxy Zoo	0.369	0.376	0.386

Table 26: Supervised learned router against Levy hard and LSNG soft. The router is a multinomial logistic gate on the candidate score vectors, trained on a stratified 10% labeled split of the test set and evaluated on the remaining 90%, averaged over five splits. Levy and LSNG use no labels and are evaluated on the same 90% splits, with accuracy computed on those held-out subsets.

Eval	Learned router	Levy hard	LSNG soft
Galaxy10	0.848	0.824	0.864
PathMNIST	0.895	0.847	0.933
Aircraft	0.814	0.214	0.928
Merged	0.850	0.635	0.929
Galaxy Zoo	0.336	0.337	0.385

Max-confidence collapses on Merged (from 0.925 to 0.023) and Aircraft (from 0.936 to 0.095) once eight synthetic bad probes per backbone enter, while LSNG soft degrades without collapsing (Aircraft 0.928 to 0.403, Merged 0.929 to 0.679). The learned router needs labels and still trails LSNG soft on every evaluation.

P Probe-capacity ablation

The linear and kNN results differ despite using the same embedding scores. We therefore examine whether probe capacity changes the comparison by fixing the candidate pool and evaluating additional heads on 3 evaluations. The heads are the linear probe, the kNN probe, and one- and two-hidden-layer MLPs (MLP-1, MLP-2). Table 28 reports top-1 accuracy.

Table 27: Per-sample max-probe-confidence routing: clean pool against a bad-probe pool (K=8, 64 synthetic bad-probe twins). LSNG soft is shown for contrast.

Eval	Max-conf clean	Max-conf +bad	LSNG clean	LSNG +bad
Galaxy10	0.868	0.778	0.863	0.856
PathMNIST	0.946	0.933	0.933	0.922
Aircraft	0.936	0.095	0.928	0.403
Merged	0.925	0.023	0.929	0.679
Galaxy Zoo	0.373	0.030	0.386	0.323

Table 28: Probe-capacity ablation. Top-1 accuracy of each operator under four probe heads on three evaluations. kNN max-confidence predictions use the same lexicographic candidate-key tie order as Table 22.

Eval	Head	Levy hard	LSNG soft	Uniform	Max-conf	Oracle
Galaxy10	linear	0.824	0.863	0.843	0.868	0.865
	kNN	0.829	0.720	0.614	0.860	0.872
	MLP-1	0.826	0.820	0.746	0.864	0.867
	MLP-2	0.833	0.797	0.727	0.864	0.870
Aircraft	linear	0.212	0.928	0.930	0.936	0.935
	kNN	0.227	0.437	0.409	0.937	0.936
	MLP-1	0.230	0.879	0.891	0.933	0.937
	MLP-2	0.266	0.862	0.872	0.931	0.935
Merged	linear	0.636	0.929	0.937	0.925	0.784
	kNN	0.688	0.693	0.732	0.917	0.840
	MLP-1	0.621	0.900	0.929	0.927	0.758
	MLP-2	0.662	0.839	0.875	0.930	0.808

Levy hard improves from the linear head to MLP-2 on all three evaluations, although Merged first dips with MLP-1. Its Aircraft failure persists across heads, consistent with the scoring failure. LSNG soft performs best with linear posteriors and loses 2.9 to 9.0 pp under the MLP heads.

Q TEXT MODALITY: ROUTING OVER FROZEN LANGUAGE-MODEL LAYERS

The preceding experiments evaluate vision representations. We next examine whether the same geometric scores remain useful for routing over frozen language-model layers. The pool holds 48 candidates: Pythia-410M and Pythia-1.4B, 24 layers each, mean-pooled over non-padding tokens, with per-dimension standardization fit on the train split. The vision protocol of per-sample L_2 normalization collapses the mid-depth probes, because two or three massive-activation dimensions carry almost all of the squared feature mass. Fitting on raw standardized features, as in the layer-wise protocol of Skean et al. (2025), restores the known mid-depth accuracy peak. Two operator variants appear only here: quantile-Levy maps each candidate's test scores through the empirical CDF of that candidate's train scores before the cross-candidate argmax, and log-space LSNG soft averages log-posteriors (a geometric mean) instead of posteriors.

Table 29: Text single-task pools (homogeneous), top-1 accuracy. Frozen Pythia layers, linear probes, no labels for routing.

Method	Banking77	Emotion
Pythia-1.4B last layer (default)	0.769	0.602
Levy hard routing	0.772	0.626
LSNG soft ensembling	0.808	0.650
Uniform average	0.809	0.656
Oracle best single	0.808	0.680

Table 30: Text merged evaluation (Banking77 + Emotion, 83 classes, same 48-candidate pool), top-1 accuracy. Log-space LSNG soft ensembling beats uniform averaging on both parts and edges the best single candidate overall.

Method	Overall	Banking77 part	Emotion part
Default 1.4B last layer	0.648	0.750	0.490
quantile-Levy hard	0.658	0.762	0.497
LSNG hard	0.671	0.770	0.518
Uniform average	0.682	0.792	0.513
LSNG soft (τ=1)	0.686	0.791	0.525
LSNG soft (log-space)	0.689	0.796	0.526
Oracle best single (1.4B L13)	0.688	0.783	0.544

On the single-task pools (Table 29), LSNG matches the oracle on Banking77 and comes within 0.1 pp of uniform averaging on that task. On the merged evaluation (Table 30), log-space LSNG beats uniform averaging on both subsets and marginally exceeds the best single candidate overall. Levy hard routing gains at most +2.4 pp over the default and trails uniform averaging on every text run, so the chi-norm geometry carries far less routing signal on anisotropic language-model states than on the vision pool.

R A LARGER SINGLE ENCODER IS NOT A SUBSTITUTE

We add a frozen DINOv2 ViT-g/14 (1.1B parameters, 40 layers, dimension 1536) under the same protocol and ask whether a larger single encoder removes the need to route. Table 31 reports its last and test-selected best layer against the default and the routed operators.

Table 31: Larger single encoder control. Frozen DINOv2 ViT-g last layer and test-selected best layer against the DINOv2-L default, Levy hard, and LSNG soft.

Eval	ViT-g last	ViT-g best	DINOv2-L last	Levy hard	LSNG soft
Galaxy10	0.519	0.612	0.513	0.824	0.863
PathMNIST	0.880	0.914	0.855	0.846	0.933
Aircraft	0.352	0.607	0.387	0.212	0.928
Merged	0.601	0.745	0.602	0.636	0.929
Galaxy Zoo	0.169	0.207	0.155	0.337	0.386

The ViT-g last layer moves the default by −3.5 pp (Aircraft) to +2.5 pp (PathMNIST) and trails LSNG soft on every evaluation. Granting ViT-g a test-selected best layer, it still trails LSNG soft on every evaluation, and its own best layers sit mid-depth (L26 to L33 of 40).

S HETEROGENEOUS POOLS, WITH AND WITHOUT SPECIALISTS

S.1 WITHOUT SPECIALISTS

The main results use candidates with 1 embedding dimension and 1 architecture family. To examine generality beyond this setup, we vary both and first evaluate a pool without specialists. The pool holds 8 frozen backbones with no finetuning: DINOv2-S/14, DINOv2-L/14, MAE-L, CLIP ViT-B/16, BEiT-B, MoCo v3 ViT-B, I-JEPA ViT-H/14, and a supervised ResNet-50. Embedding dimensions span 384 to 2048, a 5.3× range, across two architecture families and multiple pretraining paradigms, for 144 candidates. The protocol matches Section 5.2.

Levy hard routing collapses, falling 5.9 to 40.2 pp below the default. Mixed dimensions and coordinate scales remove its common radial reference (Appendix B), reproducing the norm-decoy failure of Appendix N. LSNG remains within 2.9 pp of the default on every evaluation (from −1.0 on Aircraft to +2.9 on Merged), so the dimension normalization of Section 3.3 survives both the 5.3× dimension range and the architecture change. In this pool without specialists, neither routed operator beats plain

Table 32: Heterogeneous pool without experts, linear head. Plain ens. is the uniform posterior mean over the same pool. Oracle is the best single candidate, named in the last column. **Bold** marks the best routed or averaged operator per row.

Eval	Default	Levy hard	LSNG soft	Plain ens.	Oracle	Oracle candidate
Galaxy10	0.514	0.194	0.526	**0.536**	0.574	CLIP ViT-B L11
PathMNIST	0.856	0.618	0.867	**0.887**	0.900	MoCo v3 L10
Aircraft	0.387	0.048	0.377	**0.410**	0.535	DINOv2-L L21
Merged	0.602	0.200	0.631	**0.661**	0.681	DINOv2-L L21
Galaxy Zoo	0.156	0.097	0.153	**0.158**	0.186	MoCo v3 L11

ensembling: the oracle is only $+3.0$ to $+14.8$ pp above the default, and a different generalist wins each task (CLIP on Galaxy10, MoCo v3 on PathMNIST and Galaxy Zoo, DINOv2-L on Aircraft and Merged). Routing benefits from complementary competence, as in the expert pools of the main experiments, rather than pool diversity alone.

S.2 The same pool with specialists added

To evaluate heterogeneous specialist pools, we extend the frozen-backbone control with task-finetuned candidates from multiple architecture families. We finetune 6 further experts on architectures outside the main pool, CLIP ViT-B/16 (768) and a supervised ResNet-50 (2048), on Galaxy10, PathMNIST, and Aircraft with the recipe of Appendix C.3. Added to the 8 frozen backbones and the six DINOv2-L and MAE-L experts, the pool holds 20 backbones and 372 candidates, spans 384 to 2048 dimensions and seven pretraining paradigms, and carries specialists at three different dimensions.

Table 33: Heterogeneous pool with specialists, linear head. **Bold** marks the best routed or averaged operator per row. Bootstrap standard deviations over 1000 paired test-set resamples are at most 0.012 on every entry.

Eval	Default	Levy hard	LSNG soft	Plain ens.	Oracle	Oracle candidate
Galaxy10	0.514	0.274	**0.847**	0.803	0.865	DINOv2-L Gal10 L21
PathMNIST	0.856	0.660	**0.932**	0.929	0.943	MAE-L Path L22
Aircraft	0.387	0.069	0.918	**0.923**	0.935	DINOv2-L Airc L23
Merged	0.602	0.289	0.922	**0.932**	0.784	DINOv2-L Airc L21
Galaxy Zoo	0.156	0.120	**0.380**	0.333	0.376	DINOv2-L Gal10 L17

Adding specialists changes the result. LSNG soft ensembling gains 7.6 to 53.1 pp over the default, comes within 1.8 pp of the per-dataset oracle on the 3 Single-task evaluations, exceeds it on Galaxy Zoo by 0.4 pp in point accuracy, and exceeds it on Merged by 13.8 pp. The score concentrates on the specialists rather than diffusing over the pool: experts are 61% of the candidates but receive 66 to 70% of the softmax mass. Dimension comparability therefore holds across both generalists of different widths and specialists at 768, 1024, and 2048 dimensions. Relative to the 288-candidate pool before adding the 84 CLIP and ResNet expert candidates, LSNG accuracy changes by at most 0.7 pp on any evaluation. Levy hard routing still requires a common radial scale and collapses without one, although uniform averaging over its top-k recovers most of the loss (Aircraft 0.069 at k=1 to 0.902 at k=64). Together, these controls show stronger routing gains in the heterogeneous pool containing complementary specialists.