NUISANCE REMOVAL IS A ROTATION PROBLEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Frozen encoders are reused in settings their training never anticipated, and their representations carry nuisance factors such as the layout a document was rendered in or the camera it passed through. Cleaning them up is almost always done one dimension at a time, by thresholding, masking, reweighting or selecting the largest entries of a sparse code. Practitioners assume that this per-dimension filtering separates content from nuisance.

We show that this family has a measurable ceiling. Every per-dimension operation weights a dimension by that dimension's own statistics, so it can only exploit differences between dimensions, and its headroom is governed by one quantity: the spread of the per-dimension content-to-nuisance variance ratio. The ceiling is measured rather than proved: our derivation is idealised and over-estimates what a real encoder yields, so we use it to predict ordering and check every prediction against the measurement, over eight encoders and six domains.

This analysis motivates a complementary strategy: mix dimensions rather than reweight them. A fitted diagonal escapes the ceiling, at a price we quantify: up to seven times more ratio out of sample, but less retrieval in eight of the twelve settings. The ratio is therefore a diagnostic, not a training target. Our remedy mixes dimensions instead of weighting them: a closed-form rotation estimates the nuisance subspace by PCA on the differences between two renderings of the same content and projects it out, using pairs only at fit time and no nuisance labels, and improves held-out retrieval by about ten points on the document grid. Finally, an input-dependent per-dimension readout can be worse than neutral: in a released sparse autoencoder the sparse code its global top-k produces retrieves far less than the dense head it replaces. To separate content from nuisance, rotate; do not rescale.

1 INTRODUCTION

Frozen encoders are reused far beyond the objectives they were trained on. Retrieval, editing and interpretation of their representations all require the same property: the representation should follow the content and ignore the nuisance—the layout a paragraph was rendered in, the settings a form was scanned with, the conditions a card was photographed under, the texture a shape sits on. The standard toolkit for this is per-dimension: threshold activations, keep the features that are stable across nuisance variation, reweight coordinates by inverse frequency, rescale them, or read out the top-k entries of a sparse code (Gao et al., 2025).

Sparse autoencoders, feature masking and representation-cleaning pipelines all rest on the assumption that picking the right per-dimension statistic removes the nuisance. We show that this family carries a structural limitation, and that the limitation can be measured.

Our analysis has two parts, and they are worth separating at the outset. The first is structural and does not depend on any measurement. Its methods — IDF weighting, masking, denoising, and the variance-ratio rescaling we take as their ceiling — derive each dimension's weight from that dimension's own statistics, so they can only exploit differences between dimensions. When every dimension carries content and nuisance in the same proportion, no such weighting separates them. The family's ceiling is the diagonal fitted with knowledge of the answer, and the sufficient statistic is the *spread* of the per-dimension ratio c_d/v_d, where c_d and v_d are the content and nuisance variance along dimension d. The second part is empirical: the ceiling is derived under idealisations, so it has to be measured on real encoders, and that measurement is what the rest of the paper reports. Figure 1 previews both parts.

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.

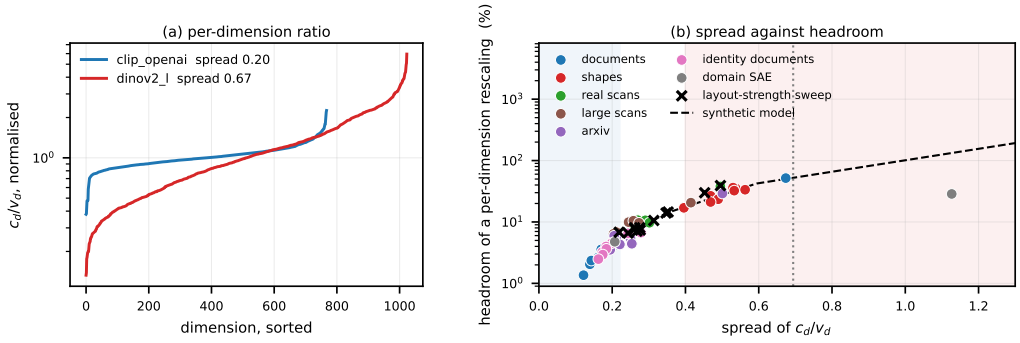

Figure 1: **Per-dimension methods are bounded, and the bound is measurable.** (a) The object of study: the per-dimension content/nuisance variance ratio that any fixed rescaling can exploit, for two encoders on the document grid. For CLIP it is nearly flat – no dimension is mostly content or mostly nuisance – while DINOv2 spreads it over decades. (b) The consequence: the spread of the ratio against the headroom of an oracle diagonal rescaling, over the 50 (encoder, domain) points of Section 4, the layout-strength sweep and the synthetic model of Section 3.2. Where the ratio is nearly flat a per-dimension rescaling gains almost nothing of the ratio; where it is uneven it gains tens of percent. The ceiling is on the ratio, not on the task: a plug-in mask can still raise retrieval by nine points on the flat document grid (Table 8). Mixing dimensions with the closed-form rotation of Section 3.4 does not have this ceiling.

We measure the spread and its consequences on a two-factor benchmark, in which content and nuisance are varied independently. The benchmark covers six domains (rendered documents, two real-scan grids, photographed identity documents, arXiv first pages, shapes on textures) and eight frozen encoders. Across 50 (encoder, domain) points the bound holds and the regimes separate: where the spread is at most 0.22, an oracle diagonal rescaling fitted on the evaluation grid itself changes the global ratio by at most 6.8%; where the spread is at least 0.40 it changes it by 21–52%. The heuristics in current use — IDF weighting, stable-feature masking, activation denoising — stay inside that bound, and most of them do nothing or worse. The spread is not a proxy for nuisance strength. In a controlled sweep of one document domain, a 1.8× wider layout range raises the nuisance distance while making the per-dimension ratios *more* uniform; the quantity therefore has to be measured rather than inferred.

We also locate the limit of the bound. A diagonal that is *fitted* rather than derived escapes it. Optimising the weights directly, within a bounded dynamic range and scored on held-out contents, yields up to seven times the ratio gain of the plug-in family (Table 11). That larger number is not a better representation. The same weights reduce retrieval in eight of the twelve settings, five of the six domains (Table 28), and cost content-probe accuracy, which the plug-in weights leave unchanged. The ratio is therefore a diagnostic rather than a target, which is an argument for mixing dimensions rather than tuning them.

Mixing dimensions escapes the bound. We estimate the nuisance subspace by PCA on the differences between two renderings of the same content and remove it by projection onto the orthogonal complement: a closed-form rotation whose only hyperparameter, the rank, is chosen on held-out contents. It uses content pairs only at fit time, needs no nuisance labels, and must be fitted in the domain it is applied to; eight paired contents are already enough to beat the raw representation. Fitted on documents whose contents the evaluation grid never sees, it raises held-out single-shot retrieval from 0.845 to 0.945, matching a mixing head trained on the same pairs and clearly ahead of a head constrained to act per dimension with the same objective and budget. On the same data it is competitive with label-supervised concept erasure (Ravfogel et al., 2020; Belrose et al., 2023), while INLP-style iterative projection, which requires labels, is far weaker.

The family also has an input-dependent member. A top-k sparse code chooses which dimensions to keep for each input, so it is a per-dimension operation whose weights depend on the image rather than on a dimension's marginal statistics. It can therefore fail in a way the bound does not describe, by

discarding content outright. In a released aligned sparse autoencoder (Nasiri-Sarvi et al., 2026) whose selection is global across a CLIP and a DINO stream, the dense pre-selection head matches the CLIP embedding it was built on. The code the model actually produces retrieves far less. Sweeping the budget restores accuracy monotonically. An autoencoder trained in-domain on an order of magnitude more data loses far less at the same budget, so the cost belongs to the readout and its budget rather than to sparse coding as such.

Contributions.

1. A two-factor benchmark of six domains and eight frozen encoders, and the identification of the per-dimension ratio spread as the sufficient statistic for any fixed per-dimension rescaling (50 points, with error bars and a measured resolution limit).
2. A measured bound for the plug-in family with its controls — the heuristics in use, the layout-strength sweep, the held-out oracle and the negative control — together with its limit: a fitted diagonal escapes the bound in ratio but pays for it in retrieval and content-probe accuracy.
3. A closed-form rotation that needs no nuisance labels, compared against label-supervised erasure, INLP and an adversarial head, and a case study in which a shared top-k readout loses most of the content.

Section 2 positions the work, Section 3 states the bound and gives the rotation, Section 4 reports the geometry, retrieval and readout experiments, and Section 5 states what the results do not cover and what we would do next. All experiments were run on a single 6 GB GPU; the anonymous release builds every grid from the public sources listed in Appendix F and reruns every analysis, and the two checkpoints it needs are the released ones listed in Appendix G.

2 Related work

Document encoders. Document models are trained to be sensitive to exactly the factor we treat as nuisance, page structure and rendering: LayoutLM (Xu et al., 2020) reads the layout explicitly, and Donut (Kim et al., 2022) reads the page image OCR-free and so is sensitive to it implicitly. We do not train any encoder, and the question we ask is what can be done with the frozen weights of general vision encoders—CLIP (Radford et al., 2021), SigLIP (Zhai et al., 2023), EVA-02 (Fang et al., 2023), ConvNeXt (Liu et al., 2022) and DINOv2 (Oquab et al., 2024)—when the same document appears under several renderings, scans or captures.

Invariance and concept erasure. A large literature removes a labelled attribute from a representation: INLP iteratively projects out probe directions (Ravfogel et al., 2020), LEACE removes the attribute linearly in closed form (Belrose et al., 2023), adversarial erasure and adversarial heads train against a classifier (Ravfogel et al., 2022; Elazar & Goldberg, 2018; Zhang et al., 2018), and invariant risk minimisation asks for predictors that agree across environments (Arjovsky et al., 2019). These methods can mix dimensions, but each needs a nuisance label for every example, and they are judged by whether the attribute is still decodable. A further family removes the leading principal directions instead, which needs no labels and no pairs but deletes variance blindly (Table 8, row "PCA removal"). Our rotation also mixes dimensions, needs only two renderings of the same content, has a single rank hyperparameter chosen on held-out contents, and is evaluated by what erasure does to the content geometry.

Per-dimension operations and sparse readouts. IDF weighting, stable-feature masking and activation denoising are the standard per-dimension recipes; SPARC (Nasiri-Sarvi et al., 2026) and other aligned autoencoders (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024) are dictionary learners. The readout we study is the input-dependent top-k selection that SPARC applies on top of the per-dimension weights (Gao et al., 2025). That literature evaluates the dictionary—what the latents mean, whether they transfer. We bound the per-dimension family, and then evaluate the readout: with identical weights and images, the dense pre-selection head matches the embedding it was built on while the sparse code retrieves substantially less.

Table 6 (Appendix A) places these families on the three axes that matter here: what supervision each method needs, whether it needs content pairs, and whether it acts per dimension or mixes dimensions. The per-dimension heuristics need no labels and their ceiling is the oracle diagonal rescaling, which is the bound we measure; the erasure methods can mix dimensions but need a nuisance label per example; the rotation needs only content pairs.

3 METHOD

3.1 GRIDS, NOTATION AND METRICS

A *grid* is a set of contents crossed with a set of nuisance realisations—layout, scan pass, capture event, render setting, background—with one image per cell, so "same content, different nuisance" and "same nuisance, different content" pairs both exist by construction. Without that pairing the quantity below is not defined, which is why we build grids instead of reusing retrieval benchmarks: an existing retrieval set varies content and nuisance together, so it cannot separate a content-driven match from a layout-driven one, and the six grids are listed in Section 4.1.

For a grid we report S_{content} and S_{layout}, the mean cosine distances over content pairs and nuisance pairs, and their ratio $R = S_{\text{content}}/S_{\text{layout}}$, which we write R and never r, because r is the rank of the subspace removed below; **NN_R@1**, single-shot cross-layout retrieval with one gallery rendering per content; the per-dimension variance decomposition into content c_d and nuisance v_d; and the **spread**, the coefficient of variation of the ratio profile, which is the quantity the rest of the paper turns on:

$$\text{spread} \;=\; \text{CV}\left(\frac{c_d}{v_d}\right) \;=\; \frac{\sigma_d\,(c_d/v_d)}{\mu_d\,(c_d/v_d)}, \tag{1}$$

where σ_d and μ_d are the standard deviation and mean over dimensions. In words, a small spread means that every dimension carries the two factors in nearly the same proportion, so a per-dimension reweighting has almost nothing to exploit. Concretely: if some dimensions were pure content and others pure nuisance the spread would be large, and reweighting would separate them; if content and nuisance were mixed in the same proportion everywhere the spread would be near zero, and no choice of weights could tell them apart. Figure 4 shows both cases on real features: the profile is nearly horizontal for CLIP on the document grid and spans decades for DINOv2 on the same grid, which is exactly the encoder whose retrieval there has already collapsed. Three things should be kept apart. Eq. equation 1 is a *definition*; the closed form of Section 3.2 is an *idealised derivation* that assumes independent coordinates and no clipping and over-estimates what a real encoder yields, so we use it only for direction and ordering; and the boundary we report is an *empirical observation*, measured for eight encoders over six domains and to be re-measured before it is assumed for another. R is a diagnostic, never a training objective (Section 3.3). The **oracle diagonal** weights each dimension by $(c_d/v_d)^{1/2}$ using the answer, so it is an analysis tool rather than a deployable method or a baseline.

3.2 THE PLUG-IN FAMILY IS BOUNDED BY THE SPREAD

The per-dimension operations in practice — IDF weighting, stable-feature masking, quantile denoising, variance-ratio rescaling — derive each dimension's weight from that dimension's own marginal statistics. Call these *plug-in* weights: each dimension's weight depends only on that dimension's own statistics, formally $w_d = f(\hat{c}_d, \hat{v}_d)$ for one fixed function f that every dimension shares. No dimension can therefore be treated as content in one image and as nuisance in another. Concretely: IDF down-weights dimensions that are active in most gallery images, masking zeroes the dimensions a layout change moves most (we use the least stable 25%), and quantile denoising zeroes the near-zero residue of each image. If every dimension carries the two factors in the same proportion, no plug-in weighting separates them, and the family's ceiling is the weighting that knows the answer and concentrates on the largest ratios, which no plug-in weight can do. Writing that ceiling as a function of the profile:

$$\frac{R_{\text{ceiling}}}{R} \;=\; \max_d\left(\frac{c_d}{v_d}\right)\cdot\frac{\sum_d v_d}{\sum_d v_d(c_d/v_d)} \;-\; 1, \tag{2}$$

which is an idealised ceiling: it assumes independent coordinates and no clipping, and on real grids it over-predicts, by a median factor of 19.6, the headroom actually measured for the oracle diagonal

Table 1: **The bound as a boundary.** Below a spread of 0.22 – every document-like setting – a per-dimension rescaling gains at most a few percent, even with weights fitted on the evaluation grid. Above 0.40 the same oracle recovers 21 to 52%.

regime	points	spread	oracle headroom
small spread	24	≤ 0.22	$\leq 6.8\%$
large spread	14	≥ 0.40	21–52%

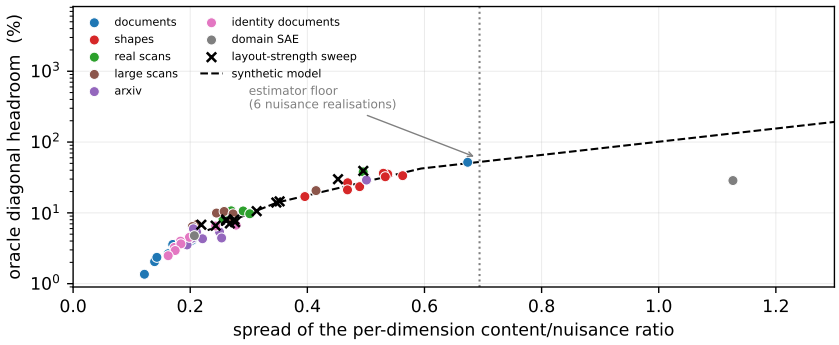

Figure 2: **The bound.** One marker per (encoder, domain) point; crosses are the layout-strength sweep; the dashed line is the synthetic model. The dotted vertical line is the estimator floor at six nuisance realisations, so the document-like spreads are upper bounds.

that is fitted from the answer's own variances, while preserving the ordering (log-log correlation 0.93, Appendix B). We therefore quote measured headroom throughout and read Eq. 2 as a statement about the direction and the ordering. Its first-order expansion in the deviation of the profile from its mean,

$$\text{headroom} \;\approx\; \mathrm{CV}^2(c_d/v_d) \;\text{ up to a } v_d\text{-weighted projection}, \tag{3}$$

matches the oracle weights at correlation 0.998 over all 50 measured points.

The consequence is a measured boundary (Table 1). Below a spread of 0.22 — every content-dominant setting — a per-dimension rescaling gains at most 6.8% even with weights fitted on the evaluation grid, over 24 points; above 0.40 the same oracle recovers 21 to 52%, over 14 points. Figure 2 plots all 50 (encoder, domain) points plus the 12 of the layout-strength sweep, and the regimes separate; the heuristics in use stay inside that bound (one entry out of 168 measured exceeds it, by 1.3 points), and most do nothing or worse: across the seven blocks and eight encoders the three heuristics move the ratio by -9.6 to $+9.1\%$ (IDF), -14.9 to $+4.1\%$ (masking) and -23.7 to $+9.8\%$ (denoising), with the per-block ranges in Table 7.

The bound constrains the ratio, not the task metrics, and the difference is visible: on the document grid the spread is 0.20 and the oracle diagonal yields 4.8% of ratio, yet a plug-in mask whose ratio gain is 1.9% raises retrieval from 0.845 to 0.938 (Table 8). A per-dimension operation can therefore move a task metric a long way inside the bound, which is why every claim we make below is made about the ratio and is reported next to task metrics.

The measurement has a resolution, and estimation noise can only inflate a measured spread: every spread reported here is an upper bound on the true one, and the true profiles are at least as uniform as measured. How much inflation is possible depends on how the nuisance varies. A synthetic model with a known zero true spread, in which one fixed nuisance vector is applied to every content, estimates the per-dimension nuisance variance from the L realisations alone and its profile carries a floor near $1.7/\sqrt{L}$ (1.04 at $L = 4$, 0.69 at 6, 0.23 at 48), which is why the sweep of Figure 2 uses 48 realisations. Our grids redraw the nuisance for every content, so the same estimate also averages over contents and that floor does not bind: on the document grid the measured spread is 0.217 at $L = 3$ and 0.201 at $L = 6$, and on shapes 0.559 at $L = 3$ against 0.392 at $L = 8$ (Appendix E).

Algorithm 1 Fit the nuisance subspace, then project it out.

Require: paired renderings of K calibration contents from the target domain, $\{(x_i^+, x_i^-)\}_{i=1}^K$, rank r

Ensure: a linear map to apply to any unpaired encoding

1: **for** $i = 1$ to K **do**
2: $\quad d_i \leftarrow x_i^+ - x_i^-$ ▷ same content, two nuisances
3: **end for**
4: $D \leftarrow [d_1, \ldots, d_K]$ ▷ one column per content pair
5: $B \leftarrow$ top-r left singular vectors of D ▷ PCA on the differences; r chosen on held-out contents
6: $P \leftarrow I - BB^\top$ ▷ projection onto the orthogonal complement
7: **return** $x \mapsto Px$ ▷ one matrix product per encoding, unpaired

3.3 Fitted weights escape the bound and pay for it

The bound is a statement about plug-in weights, not about everything one can do one dimension at a time. Optimising the per-dimension weights directly on the ratio, under a bounded dynamic range and scored on held-out contents, yields much more of it: $+14.9\%$ against $+3.1\%$ on rendered documents, $+30.6\%$ against $+11.9\%$ on shapes, $+21.6\%$ against $+2.9\%$ on identity documents, and up to seven times the plug-in gain overall (Appendix B.3).

That larger number is not a better representation. The fitted weights cost retrieval in eight of the twelve settings — five of the six domains, by up to 2.8 points on arxiv/CLIP and 1.3 on identity documents (Table 28) — and cost up to seventeen points of content-probe accuracy ($0.567 \to 0.488$ on documents, $0.701 \to 0.533$ and $0.391 \to 0.255$ on identity documents), while the plug-in weights leave the probe unchanged. Outside the bounded range the ratio can be inflated without limit: the unconstrained maximiser reaches hundreds of percent while reducing retrieval to zero, so we treat the ratio as a *diagnostic* and judge every method by task metrics as well.

3.4 The rotation

Mixing dimensions escapes the bound. Two renderings of the same content differ in nuisance and noise, so the principal directions of their differences estimate the nuisance subspace; projecting that subspace out removes the directions in which the representation is unstable. With K calibration contents and their pairs,

$$B = \text{top-}r\ \text{PCA}\{x_i^+ - x_i^-\}_{i=1}^K, \qquad x \ \mapsto\ x - BB^\top x, \tag{4}$$

Algorithm 1 is the whole method (Eq. 4 is the same two steps in one line). The fit needs content pairs but no nuisance labels and no knowledge of what the nuisance is — on the document grid the paired renderings differ in font, width, margin, spacing and alignment simultaneously. Inference is unpaired: one matrix product per sample. Two properties are limits rather than features: the subspace must be fitted in the domain it is applied to, and the rank is a real hyperparameter. We use $r = 4$ throughout; choosing it instead on a calibration half and scoring it on held-out contents matches or beats the fixed choice (0.957 against 0.948 on documents), while the energy rule (the directions that carry 95% of the difference variance) selects 21–343 and does not help.

Four conditions bound the method. It must be fitted on pairs from the domain it is applied to, since a subspace fitted elsewhere transfers nothing; its rank is set on held-out contents, where the held-out rule matches or beats our $r = 4$; with no pairs the augmentation fallback recovers about 60% of the gain for a rendering nuisance and almost nothing for a capture event, so it must be checked first; and where the nuisance is weak there is little to remove — below a spread of 0.22 even the oracle diagonal yields at most 6.8% of ratio.

The controls are in Appendix C. Fitting the same procedure on mismatched pairs gives 0.798 on the document grid, worse than doing nothing (0.845), while matched pairs give 0.917: the gain comes from the pairing, not from low-rank structure, and the subspace does not transfer between domains. Where a domain has no pairs, two augmentations of one image can stand in, and the two conditions for that fallback are measurable before fitting: the augmentation subspace must cover the

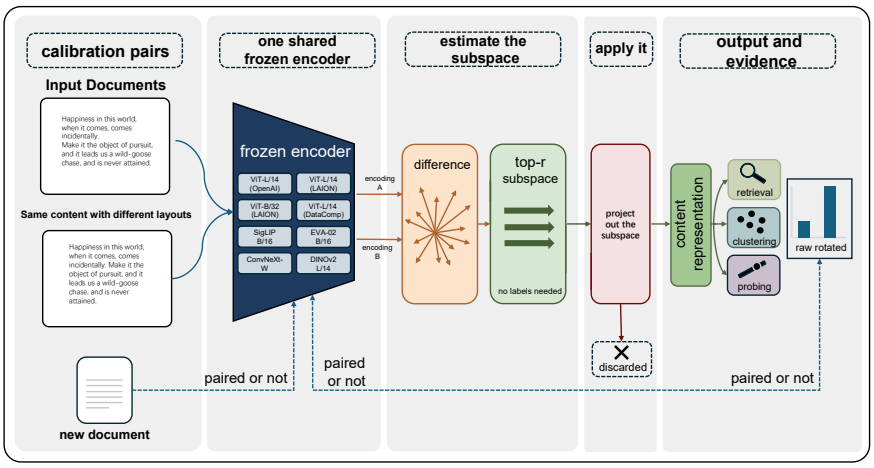

Figure 3: **The method in one picture.** Left: two renderings of one calibration content differ only in nuisance, so the difference of their encodings lies in the nuisance subspace, and the top-r principal directions of those differences span it; the estimate uses content pairs only, with no nuisance labels and no fine-tuning. Right: the same frozen encoder embeds a later document, and projecting the estimated subspace out of that encoding – paired or not – corrects it with a single matrix product, while the discarded directions leave the pipeline.

nuisance subspace (0.66 principal cosine on documents, 0.86 on identity) and there must be nuisance to remove ($+0.084$ recall from projecting the true subspace out on documents, $+0.012$ on identity documents).

4 Experiments

4.1 Grids and encoders

The six grids are the rendered documents (84 contents under six layouts), shapes (16, eight), two real-scan grids (199, four) and (919, four), identity documents (500, five) and arXiv pages (95, four), plus a disjoint 560-content training grid. Eight frozen encoders are used (CLIP ViT-L/14 and ViT-B/32, SigLIP, EVA-02, ConvNeXt, DINOv2); grid construction, splits and checkpoints are in Appendices F and G.

4.2 Where the encoders sit

Table 2 summarises the 50 (encoder, domain) points and Table 22 gives them one by one. Two regimes appear. The content-dominant domains — rendered documents, both scan grids, identity documents and arXiv pages — sit at the bottom of the spread range, the identity grid lowest of all, with oracle headroom of a few percent. The nuisance-dominated settings — shapes on textures, DINOv2 on the document grid, and the sparse readout of Section 4.4 — sit at the top, with headroom of tens of percent. DINOv2 is the one encoder that crosses the boundary inside a content-dominant domain, and it is also the encoder whose retrieval on the document grid has already collapsed: a per-dimension fix has most to offer exactly where the representation has lost the content. The spread is not a proxy for nuisance strength. Widening the layout parameters of one document domain by a factor of 1.8 raises the nuisance distance monotonically while making the profile *more* uniform (Appendix B.5).

4.3 Retrieval

Fitted on document contents the evaluation grid never sees, the rotation raises held-out single-shot retrieval from 0.845 to 0.945 at $r = 4$, against 0.950 for a mixing head trained on the same pairs with

Table 2: **The 50 points, by domain.** Content/nuisance ratio, spread, oracle headroom and cross-layout retrieval, each as a range over the encoders of that domain; the last row treats the released and in-domain SAE readouts as representations. The point-by-point table is Table 22 (Appendix I).

domain	points	ratio	spread	oracle headroom	NN_R@1
documents	8	0.25–2.36	0.122–0.674	+1.4–+51.8%	0.033–0.971
shapes	8	0.68–0.97	0.396–0.563	+17.0–+36.3%	0.473–0.795
real scans	8	3.22–6.98	0.202–0.495	+4.8–+37.9%	0.970–0.998
large scans	8	4.72–7.91	0.202–0.415	+4.3–+20.6%	0.919–0.991
arxiv	8	3.53–5.21	0.195–0.501	+3.5–+29.1%	0.839–0.954
identity documents	8	3.89–6.80	0.162–0.278	+2.5–+6.7%	0.790–0.863
in-domain SAE	2	1.61–2.18	0.207–1.127	+4.8–+28.6%	0.419–0.855

Table 3: **Closed form against trained heads**, fitted on the same disjoint training contents and scored on the document grid. The closed form has no objective and no learned parameters beyond the rank; the per-dimension head, trained with the same objective and budget, recovers almost none of the gain.

encoder	representation	ratio	NN_R@1
clip_openai	raw	2.116	0.845
clip_openai	closed form r2	2.220	0.921
clip_openai	closed form r4	2.223	0.945
clip_openai	closed form r8	2.200	0.945
clip_openai	trained per-dimension	2.127	0.876
clip_openai	trained mixing (linear)	2.369	0.950
clip_openai	trained mixing (MLP)	2.437	0.933
siglip	raw	2.360	0.971
siglip	closed form r2	2.527	0.998
siglip	closed form r4	2.557	0.998
siglip	closed form r8	2.541	1.000
dinov2_l	raw	0.247	0.033
dinov2_l	closed form r2	0.310	0.040
dinov2_l	closed form r4	0.295	0.043
dinov2_l	closed form r8	0.287	0.036

the same budget, and 0.876 for a head constrained to act per dimension — the closed form therefore recovers 95% of the gain without an objective (Table 3). On the same data it is competitive with label-supervised erasure: averaged over twelve settings, LEACE-style erasure gains $+5.1$ retrieval points and wins in 7 of 12, INLP $+0.7$ in 6 of 12, and the rotation $+4.3$ in 12 of 12, with the largest mean ratio gain ($+1.11$ against $+0.68$; the twelve settings are listed one by one in Table 29); it does not beat label-supervised erasure on retrieval, but it needs pairs rather than labels and never loses to the raw representation (Appendix C).

Under the retrieval protocol with unknown queries the rotation improves R@1 and mAP and, on documents, the separation of known from unknown queries (AUROC 0.909 to 0.933; Appendix I). Downstream it improves k-means clustering wherever the nuisance is real (adjusted Rand index 0.080 to 0.466 on shapes, 0.561 to 0.677 on arXiv) while pooling three sources into 794 contents is flat, as the transfer control predicts.

4.4 A TOP-k READOUT CAN BE WORSE THAN NEUTRAL

The operations bounded above are *fixed* functions of the representation. A top-k sparse code is not: it chooses its dimensions per input, so it can fail in a way the boundary does not describe, by discarding content outright — and it is the readout current document and vision systems use. In a released aligned sparse autoencoder whose latent space is shared by a CLIP image stream, a CLIP text stream and a DINO stream under a *global* top-k selection, one budget of 64 slots is divided between the streams, so the content stream does not receive a budget of its own.

Table 4: **Dense against sparse, same weights and same images.** The released aligned autoencoder on four domains: the embedding it was built from, its dense pre-selection head, the code its global top-k actually produces, and how many of the 64 slots reach the content stream.

domain	CLIP	dense head	top-k code	released budget
documents	0.924	0.921	0.112	17 of 64
shapes	0.795	0.750	0.161	16 of 64
real scans	0.997	0.997	0.777	19 of 64
arxiv	0.937	0.940	0.596	22 of 64

Reading the same weights over the same images two ways isolates the selection. The dense pre-selection head of the CLIP stream is at parity with the CLIP embedding the autoencoder was built on, on every domain; the code the model actually produces is far worse, by 22% of the retrieval on real scans and 88% on rendered documents, where it keeps about 17 of the 64 slots and retrieves 0.112 against the dense head's 0.921. Sweeping the budget after the fact recovers monotonically — 0.486 at the operating point to 0.917 at $k = 1024$ — and at the same budget a per-stream rule (0.510) beats the global one (0.112), so most of the damage is done by sharing the budget, not by sparsifying, and a nucleus rule that chooses its own count returns to the dense level after keeping about 1,300 of the 8,192 latents. The control that isolates the dictionary is an autoencoder trained in-domain on 24,000 images (4,000 contents, six fixed layouts, texts disjoint from the evaluation grid): its top-k loss at the same nominal budget is 10 points (0.890 to 0.800) instead of 44, and it is monotone in the budget (Table 9, Figure 7).

5 Limitations and conclusion

Limitations. The boundary we report is a measurement, not a theorem: it holds for the eight frozen encoders and six domains of this paper, and a new encoder or a new domain should be measured before the same statement is assumed. Three further limits matter in practice. A fitted diagonal escapes the bound, and we do not claim otherwise: what we claim is about the per-dimension methods people derive and about the ratio as a diagnostic. The rotation must be fitted on pairs from the domain it is applied to, since a subspace fitted elsewhere transfers nothing; where no pairs exist, the augmentation fallback recovers about 60% of the gain for a rendering nuisance and almost nothing for a capture event, so it has to pass the two measurable checks of Section 3.4 first. And where the nuisance is weak there is little to remove. Two honest caveats on the numbers: the estimator's resolution, the compromises in each domain (a digital second factor, a single capture device, synthetic fitting grids) and the statistical power of the comparisons are discussed in Appendix E, and nuisance is never fully removed — every method we compare leaves it linearly decodable at 0.42–0.80 against a chance level of 0.17.

Future work. Three directions follow: a dictionary trained with a per-stream budget, domains where the same document is captured twice by different devices, and a statement about fitted per-dimension maps under correlated coordinates, where our first-order form already tracks the oracle at correlation 0.998.

Conclusion. For a frozen representation, the per-dimension family — threshold, mask, reweight, rescale, select — is bounded by the spread of the content/nuisance variance ratio, and that spread is small exactly where retrieval is content dominated. Mixing dimensions escapes the bound: a closed-form rotation on content-paired differences needs no nuisance labels, has one hyperparameter, recovers within a point of a trained mixing head and is competitive with label-supervised erasure, provided the pairs come from the domain it is applied to. An input-dependent per-dimension readout can be worse than neutral, and the case study says why: the loss belongs to the budget the content stream receives rather than to the autoencoder. To separate content from nuisance, rotate; do not rescale.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the manuscript text and debugging the experiment code. We have not used generative AI tools for generating experimental result, number, table or figure in this paper. We have reviewed all AI-assisted work, we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our experiments use public research corpora of documents and identity documents. No human subjects were involved and we collected no new data. The identity-document grid comes from a filtered distribution of MIDV-500 (`MIDV-500-filter`) (Arlazarov et al., 2019), a public research corpus of real personal documents; we use it only to measure the geometry of frozen representations, we train no recogniser, we make no attempt to read any personal information from the images, and we redistribute no document images. The finding that a shared top-k readout can discard content is reported as a measurement and the practical recommendation is a diagnostic one. Like any tool that removes a factor from a representation, the rotation we propose could be misused to remove content rather than nuisance; we therefore report task metrics alongside every ratio and state the conditions under which the method should not be applied. The encoders we study are used exactly as released, so any bias they carry is inherited unchanged, and our measurements characterise it rather than correct it.

REPRODUCIBILITY STATEMENT

All experiments run on a single 6 GB GPU and a desktop CPU. Each grid is built from a public source and described in Appendix F, together with the fitting grid; the encoders are the released checkpoints listed in Appendix G, used as published and never fine-tuned. The pairing, fitting and evaluation protocols are given in Sections 3 and 4, and the contents used for fitting are disjoint from the contents that are evaluated. Autoencoder configurations are given in Appendix H and runtimes in Appendix J, and every table states the setting its numbers were obtained under. Figure 3 is a schematic rather than a data figure. An anonymous release of the code and of the grid-construction scripts is available at `https://github.com/AAAnonymous-aaa/nuisance-rotation`. It rebuilds every grid from the public corpora listed in Appendix F, reruns every analysis, and reproduces the tables and data figures of this paper; the two checkpoints it expects, the OpenAI CLIP weights and the released autoencoder, are not redistributed and must be obtained from their public sources.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Vladimir V. Arlazarov, Konstantin Bulatov, Timofey Chernov, and Vladimir L. Arlazarov. MIDV-500: A dataset for identity document analysis and recognition on mobile devices in video stream. *Computer Optics*, 43(5):818–824, 2019. doi: 10.18287/2412-6179-2019-43-5-818-824.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, et al. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023.

Konstantin B. Bulatov, Elena V. Emelianova, Daniil V. Tropin, Natalya S. Skoryukina, Yulia S. Chernyshova, Alexander V. Sheshkus, Sergey A. Usilin, Zuheng Ming, Jean-Christophe Burie, Muhammad Muzzamil Luqman, and Vladimir V. Arlazarov. MIDV-2020: A comprehensive benchmark dataset for identity document analysis. *Computer Optics*, 46(2), 2022. doi: 10.18287/2412-6179-CO-1006.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, 2023. doi: 10.1109/CVPR52729.2023.00276.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022. The FLORES-200 evaluation set is released with this report.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024.

Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, et al. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 27092–27112, 2023. doi: 10.52202/075280-1179.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations (ICLR)*, 2025.

Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

Guillaume Jaume, Hazım Kemal Ekenel, and Jean-Philippe Thiran. FUNSD: A dataset for form understanding in noisy scanned documents. In *International Conference on Document Analysis and Recognition Workshops (ICDARW)*, 2019.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jisang Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022.

Jens Lehmann, Robert Isele, Max Jakob, et al. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 2015.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

Ali Nasiri-Sarvi, Hassan Rivaz, and Mahdi S. Hosseini. SPARC: Concept-aligned sparse autoencoders for cross model and cross-modal interpretability. *Transactions on Machine Learning Research (TMLR)*, 2026. Published in the 3/2026 issue; preprint arXiv:2507.06265.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

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

Table 5: **The six domains.** Each varies content and nuisance independently. The last two rows are the auxiliary capture grid (the same content type as the identity grid, with longer capture series, used by the fallback and rank controls) and the synthetic training grid, whose texts are disjoint from the evaluation grid and which is used only to fit the method of Appendix C. "Realisation" names the content and nuisance factors of each pair: a digital pass is a transformation we apply to a real scan, a capture is a second physical photograph of the same document.

domain	content axis	contents	nuisance	realisation
documents	topic × doc	84	6	synthetic
shapes	shape, colour	16	8	synthetic
real scans	scanned form	199	4	real content, digital pass
large scans	scanned page	919	4	real content, digital pass
identity documents	identity document	500	5	real content, real capture
arxiv	paper	95	4	real content, render setting
real captures	identity document (long series)	10	30	real content, real capture
training grid	topic × doc	560	6	synthetic

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning (ICML)*, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 25278–25294, 2022. doi: 10.52202/068431-1833.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, et al. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. Transformer Circuits Thread, 2024.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. LayoutLM: Pre-training of text and layout for document image understanding. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2020.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2018.

A THE TWO-FACTOR BENCHMARK

Everything here is measured on grids in which two factors vary independently. A *content* is an object whose identity should be preserved; a *nuisance realisation* is a way of rendering, scanning or photographing it that should not change that identity. Every content appears under every realisation of its domain, so "same content, different nuisance" and "same nuisance, different content" pairs both exist by construction. Without that pairing the quantity of Appendix B is not defined, which is why we build grids rather than reuse retrieval benchmarks. Table 5 lists the six evaluation domains together with the auxiliary capture grid and the synthetic training grid.

Two synthetic grids. 14 topic-labelled corpora contribute six documents each, each rendered under six fixed layouts (font size, page width, margin, line spacing, alignment, font); and 16 shape-and-colour contents under eight nuisance tuples (texture, rotation, scale, offset). Both are ours, which is the weakness the real domains address.

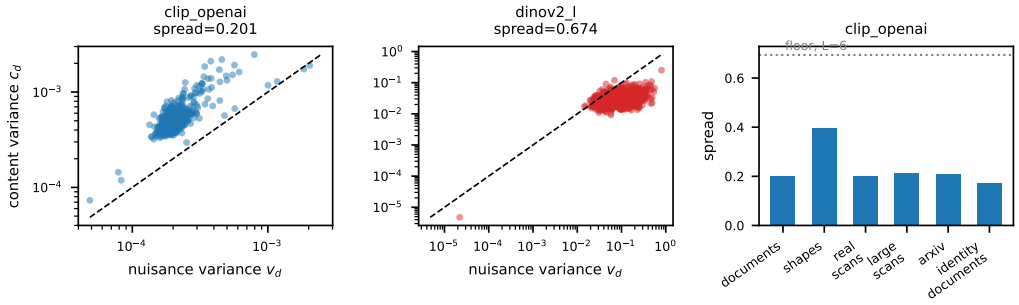

Figure 4: **The geometry this paper is about.** Left and middle: nuisance against content variance per dimension on the document grid, with the identity line. CLIP sits near it; DINOv2 does not. Right: measured spread per domain, with the resolution limit of Appendix B marked.

Three real-content grids. 199 real scanned forms (FUNSD) and 919 real scanned archive pages from DocVQA (Mathew et al., 2021) are put through the same four digital second-scan passes — identity, JPEG re-compression, half resolution, and a small skew with uneven illumination — and 95 arXiv first pages under four render settings. The pages, their layouts and their scanner noise are real; the second factor is ours, because no public corpus of this size contains two physical scans of the same page. We say so wherever these numbers appear.

One real-capture grid. The only domain where both sides of the pair are real events: 500 identity documents from the filtered MIDV-500 distribution (`MIDV-500-filter`), five photographs of each document (2,500 images over 50 document types, taken at different pose and framing; the device does not change within a document, see Appendix E). A second mirror of the same corpus carries longer capture series for ten documents (28–30 captures each); it is the auxiliary capture grid behind the rank curves of Appendix I.1 and the capture rows of Table 16.

Encoders and metrics. Eight frozen encoders spanning the contrastive and self-supervised families (architectures and weights in Table 10); nothing is fine-tuned, and all features are cached, which is why the study fits on one 6 GB GPU. Every table reports the metrics defined in Section 3.1: the content and nuisance distances and their ratio, single-shot cross-layout retrieval (NN_R@1), the per-dimension content and nuisance variances, the spread of their ratio, the oracle diagonal, and the rotation.

Figure 4 shows why the decomposition matters: for CLIP the per-dimension points lie near the identity line — dimensions with more content carry proportionally more nuisance — while DINOv2 spreads over two orders of magnitude and loses the content entirely.

Design requirements. Two choices are load-bearing. A fixed canvas height removes the content signal for most encoders, because page geometry is the layout-invariant cue these models use (Table 19), so images are sized to their text. And at least four nuisance realisations are needed: with only two, the content/nuisance ratio inflates badly on real content (arXiv pages read 26.5 instead of 3.6), which Appendix B quantifies as an estimator floor. Two of the six domains are synthetic by construction, three put real content through a digital second factor (both scan grids) or a render setting (arXiv), and one, the identity grid, pairs real content with real capture events. The set is useful because it disagrees about everything except the measured quantity.

A.1 WHERE THIS PAPER SITS

Table 6: **Where this paper sits.** The per-dimension heuristics need no labels, and their ceiling is the oracle diagonal rescaling, which needs the answer and is the bound we measure. The erasure methods can mix dimensions but require a nuisance label for every example, and they are evaluated by whether the attribute is still decodable; the rotation proposed here mixes dimensions as well, but needs only two renderings of the same content, has a single rank hyperparameter chosen on held-out contents, and is therefore outside the per-dimension family whose limit Appendix B measures rather than assumes.

Method	Input it needs	Per-dim.?	Fitted by
IDF weighting	corpus statistics	yes	closed form
Stable masking	nuisance statistics	yes	closed form
Quantile denoising	activation quantiles	yes	closed form
Oracle diagonal	the variances	yes	closed form
INLP (Ravfogel et al., 2020)	nuisance label	no	iterative
LEACE (Belrose et al., 2023)	nuisance label	no	closed form
Linear adversarial (Ravfogel et al., 2022)	nuisance label	no	closed form
Adversarial head	nuisance label	no	alternating
Rotation (ours)	**two renderings**	no	**closed form**

B THE PER-DIMENSION BOUND

B.1 WHAT A PLUG-IN RESCALING CAN DO

We start with the per-dimension methods in actual use. IDF weighting, stable-feature masking, quantile denoising and the variance-ratio rescaling we use as their ceiling all share one property, which we make the definition of the family:

Definition (plug-in rescaling). Let $\hat{c}_d$ and $\hat{v}_d$ be the content and nuisance variance of dimension d estimated from content pairs, or any marginal statistic of dimension d alone (its corpus frequency, its activation quantiles). A rescaling is *plug-in* if there is a fixed function f, the same for every dimension, with $w_d = f(\hat{c}_d, \hat{v}_d)$.

Every heuristic in Appendix B.4 is plug-in in this sense, as is the variance-ratio oracle $f = (\hat{c}_d/\hat{v}_d)^{1/2}$. The definition is what makes the bound a statement about a family rather than about a list.

Cosine geometry. Cosine distance between unit-norm encodings equals half the squared Euclidean distance, so the additive-model derivation below applies to $S_{\text{content}}/S_{\text{layout}}$ exactly as written; what has to be argued is only that reweighting does not invalidate it. Reweighting multiplies each coordinate and then renormalises, which perturbs the norm profile across points; the induced change in the ratio enters at second order, and we measure it: under the fitted weights of Appendix B.3 the norm profile has relative standard deviation 0.02–0.10, and the first-order formula predicts the realised gain with correlation 0.998 and a median error of 5% (Appendix B).

A fixed per-dimension rescaling multiplies dimension d by some weight w_d, which leaves the per-dimension ratio of the two factors unchanged: an operation of this family can only exploit *differences between dimensions*. In the additive two-factor model with squared distances, where an encoding is $u_{\text{content}} + w_{\text{nuisance}} + \varepsilon$ with independent coordinates, the ratio of any such rescaling is

$$\text{ratio}(a) \;=\; \frac{\sum_d a_d^2 c_d}{\sum_d a_d^2 v_d} \;=\; \frac{\sum_d \omega_d \rho_d}{\sum_d \omega_d}, \qquad \omega_d = a_d^2 v_d, \quad \rho_d = c_d/v_d, \tag{5}$$

a plain weighted average of the per-dimension ratios. Two consequences follow directly. First, $\text{ratio}(a) \le \max_d \rho_d$, attained by concentrating weight on the largest ratios: that is the *definitional*

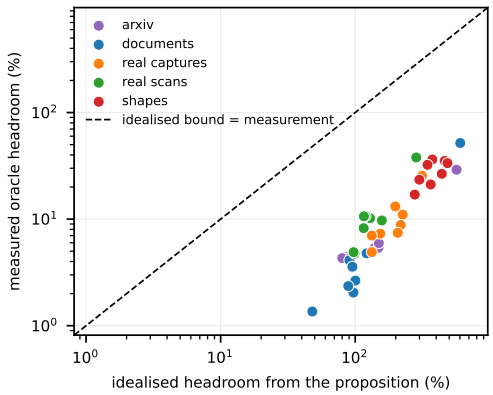

Figure 5: **Idealised ceiling against measurement.** Headroom predicted by Eq. 2 from the measured per-dimension variances, against the headroom actually measured for the diagonal fitted from those same variances. Ordering preserved, magnitude not.

ceiling written out as Eq. 2 in the main text, which no plug-in weighting reaches, not even the variance-ratio oracle $(\hat{c}_d/\hat{v}_d)^{1/2}$ of Section 3.1. Second, writing any weighting as $\omega_d = v_d(1 + s_d)$ and expanding to first order,

$$\frac{\text{ratio}(a)}{\text{ratio}(1)} - 1 \;=\; \frac{\text{Cov}_v(s, \rho)}{\bar{\rho}_v} \;+\; O(s^2)\,: \tag{6}$$

a rescaling gains to first order exactly to the extent that its weighting covaries with the per-dimension ratio. We use the coefficient of variation of that profile as the summary statistic and call it the *spread*. We are explicit that Eq. 5 is close to definitional for this family, and that what matters empirically is the gap between the ceiling, the ceiling a *fitted* diagonal reaches, and the measured one.

Two numbers size that gap. Evaluated on the oracle weights of the 50 points, Eq. 6 predicts the measured gain with correlation 0.998 and a median error of 5%; evaluated on weights fitted on half the contents it still ranks the settings perfectly (0.999) but over-predicts their gain by a median factor of 1.9, because estimation noise attenuates the covariance term. The definitional ceiling itself is loose on real features: it exceeds the measured oracle by a median factor of 20 (range 7.5 to 47, Figure 5), since cosine geometry, correlated dimensions and clipped weights all stop a rescaling from isolating one dimension's variance. We therefore quote measured headroom throughout and use the equations for the direction and the ordering.

B.2 Where encoders sit

Figure 2 plots all 50 points plus the 12 of the layout-strength sweep, and the regimes separate. The content-dominant domains (rendered documents, both scan grids, identity documents, arXiv pages) run from spread 0.12 to 0.42 with headroom between $+1.4$ and $+20.6\%$, the top of that range being DINOv2 on the large scan grid; the identity grid, whose nuisance is the most real of all, sits at the bottom (spread 0.16–0.28, headroom $+2.5$ to $+6.7\%$). Only DINOv2 goes past that range inside those domains — 0.50 on arXiv, 0.50 on the form-scan grid, 0.67 on the document grid — and it is the encoder whose retrieval has already collapsed (0.033 on the document grid). The domains whose nuisance dominates (shapes on textures, DINOv2 on the document grid, the sparse readout of Appendix D) run from 0.40 to 1.13 with headroom $+17$ to $+52\%$. The same statement read from the other end: a per-dimension fix has most to offer exactly where the representation has already lost the content.

B.3 THE BOUND COVERS PLUG-IN WEIGHTS, NOT THE WHOLE FAMILY

The distinction is not cosmetic. Optimising the per-dimension weights directly on the ratio, under a bounded dynamic range (weights confined to $[1/c, c]$ so that the solution cannot degenerate), and scoring on held-out contents, gives much more headroom than the plug-in family: $+14.9\%$ against $+3.1\%$ on documents, $+30.6\%$ against $+11.9\%$ on shapes, $+21.6\%$ against $+2.9\%$ on the identity documents, and in the same direction on the other three grids (Table 11). Three seeds move these figures by at most 2.6 points in eleven of the twelve settings and by 7.5 on shapes/SigLIP, fitting on even realisations and scoring on odd ones gives the same answer ($+9$ to $+46\%$ depending on the domain), unclipped plug-in weights give exactly the plug-in figures, and random weights of the same dynamic range move the ratio by only 0.0–2.8% — so the effect is neither a split artefact nor a clipping artefact nor a property of the ratio being fragile.

Whether the larger number is a better representation depends on the domain, and that is the useful part. Table 28 gives the retrieval of the raw representation, the plug-in reweighting and the fitted diagonal in all twelve settings. On the shapes grid the fitted weights improve every metric we have: ratio $+31$–58% and retrieval $+11$ points for CLIP and $+13$ points for SigLIP. On the document-like grids they buy ratio (up to $+33\%$) at a cost of up to 2.8 retrieval points, and where a fine-grained topic label exists they cost a great deal of content accuracy: $0.567 \to 0.488$ on rendered documents, $0.701 \to 0.533$ and $0.391 \to 0.255$ on the identity documents, while the plug-in weights leave the probe unchanged or slightly better. And beyond the bounded range the ratio can be inflated without limit: the unconstrained maximiser reaches several hundred percent while reducing retrieval to zero (Table 12). We therefore treat the ratio as a *diagnostic* and judge every method by task metrics as well, which is what the rest of the paper does. The reading that survives all controls is: the weights people derive rather than fit are bounded by the spread, fitting them moves the ratio by several times that bound, and whether the fitted version is worth having is a question about the task, not about the ratio.

Where the escape comes from. The fitted weights are only partly explained by the marginal profile that the definition is built on: their correlation with the plug-in weights is 0.54–0.70 across the twelve (domain, encoder) settings we checked, so roughly half of the fitted solution's variance lies outside the plug-in family. That is the escape route — a fitted diagonal exploits the joint structure of the grid, which dimensions' differences are aligned with which other dimensions', and no function of the marginals can see it — and it is why the claim is stated as a definition plus a bound rather than as an impossibility result.

B.4 THE HEURISTICS IN USE

The three per-dimension heuristics are defined here, because the point of the panel is that they are ordinary implementations rather than straw men. All of them act on one encoding x of a frozen encoder and are plug-in in the sense of Section B: each derives dimension d's weight from that dimension's own marginal statistic over the gallery, and renormalises the weighted vector. *IDF weighting* uses $w_d = \log \frac{N+1}{n_d+1} + 1$, where N is the number of gallery images and n_d counts those in which dimension d is positive, so dimensions that are active almost everywhere are down-weighted; the fair table of Appendix B.3 uses the same formula with n_d counting non-zero entries instead. *Masking* zeroes a subset of dimensions: the panel below keeps the dimensions active in at least 25% of the gallery images, and the stable mask of the fair table keeps the 25% whose per-layout mean varies least across the nuisance realisations, which keeps the dimensions that a change of layout does not move. *Quantile denoising* zeroes every entry of an image below 1% of the 99th percentile of that image's positive entries, i.e. it removes the near-zero residue rather than reweighting anything. Because the three act on continuous embedding coordinates rather than on discrete features, the panel reports the content/nuisance ratio after each operation, so a negative entry means the operation moved the two distances in the wrong direction.

Table 7 applies the three heuristics a practitioner reaches for to all seven blocks: seven blocks times three heuristics times eight encoders is 168 measured entries, which the table shows as the range over the encoders of each block, so every cell is a pair of numbers and not a single measurement. With one exception — IDF weighting on the large scan grid, which beats that point's bound by 1.3 points – every entry stays inside the oracle bound of its block, and most do nothing or worse: quantile

Table 7: **Heuristics, all blocks** (percentage change in the content/nuisance ratio). Every entry is inside the oracle bound of its block.

domain	IDF	masking	denoising
documents	-6.5/+0.7	-0.3/+4.1	-10.5/+9.8
documents (in-domain SAE)	-9.4/-1.9	-14.9/+8.3	-23.7/-0.0
shapes	-4.1/+3.6	-1.9/+3.6	-0.1/+4.9
real scans	-9.6/+3.2	-1.0/+3.2	-13.4/-1.9
large scans	-6.2/+9.1	-3.5/+2.4	-18.2/-9.6
arxiv	-1.9/+3.5	-3.3/+0.4	-12.4/-7.4
identity documents	-2.7/+3.1	-0.5/+2.1	-11.8/-7.9

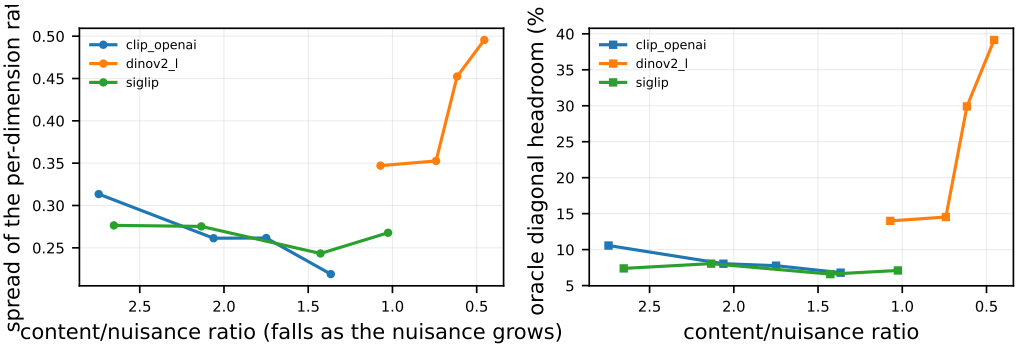

Figure 6: **Spread against nuisance strength.** As the layout span grows the ratio falls (right to left) but the spread falls too: the spread is a property of the geometry, not a proxy for nuisance strength.

denoising is harmful in five of the seven blocks, by up to -23.7%, and masking never exceeds $+8.3\%$. This is the negative result in its most practical form — the operations people actually run are bounded by a quantity that is small exactly where they are used.

B.5 The spread is not nuisance strength

The obvious reading of the above is "weak nuisance is safe, strong nuisance is dangerous"; a controlled sweep falsifies it. Scaling the span of one document domain's layout parameters by 1.8 grows the nuisance monotonically (the mean distance between realisations of a content rises from 0.139 to 0.246) and the content/nuisance ratio falls monotonically (2.74 to 1.37), but the spread *falls* too (0.313 to 0.219) and the headroom with it ($+10.6\%$ to $+6.8\%$ for CLIP, Figure 6). The most content-dominant level of the sweep is the one with the largest headroom. The mechanism is that a weak nuisance leaves many dimensions with a near-zero v_d, which makes the ratio ill-conditioned and gives a diagonal method something to exploit. The spread therefore cannot be inferred from how strong the nuisance looks — and a rotation fitted where there is little nuisance to remove can lose to the oracle, which is what happens on the arXiv pages (Appendix C.5).

B.6 Resolution: how much of the spread is signal

The spread is estimated from finitely many contents and realisations, so part of it is estimation noise. In a synthetic model whose content and nuisance profiles are proportional by construction — true spread zero — the measured spread depends on the number of realisations L and not on the number of contents: 1.04 at $L = 4$, 0.69 at 6, 0.46 at 12, 0.31 at 24 and 0.23 at 48, i.e. roughly $1.7/\sqrt{L}$, while growing the content count from 84 to 672 moves it from 0.69 to 0.76. Our grids use four to six realisations, so the document-like spreads of 0.12–0.42 are at or below the floor: they are upper bounds and the underlying uniformity is understated. The same model, given enough realisations, reproduces the measured curve (spread 0.23 gives $+5.7\%$, 0.59 gives $+42\%$, 1.35 gives $+213\%$), which is what makes us willing to read the scatter of Figure 2 as a relationship.

Resampling eight 80% subsets per (domain, encoder) moves the spread by at most 9.6% relative and the headroom by at most 5.3 points, against the 14-point separation the boundary relies on (Tables 24 and 23 in Appendix I).

C A CLOSED-FORM ROTATION

C.1 FITTING AND INFERENCE

This appendix gives the details behind Algorithm 1 in the main text, which is the whole method. Two renderings of the same content differ only in nuisance plus noise, so the principal directions of their differences estimate the nuisance subspace; projecting it out removes exactly the directions in which the representation is unstable across nuisance variation. The fit needs content pairs but no nuisance labels and no knowledge of what the nuisance is — on the document grid the paired renderings differ in font, width, margin, spacing and alignment simultaneously. Inference is unpaired: one matrix product per sample. Two properties are limits rather than features: the subspace must be fitted in the domain it is applied to, and the rank is a real hyperparameter. Figure 3 in the main text is the same two steps as a picture.

C.2 THE METHOD'S OWN HYPERPARAMETERS

Rank (Figure 8, left): most of the gain arrives by $r = 2$–4 — documents go from 0.845 to 0.921 at $r = 4$ for CLIP and from 0.971 to 1.000 by $r = 1$ for SigLIP — while at $r = 32$ the curve turns over in nine of the fifteen settings. Can the rank be chosen without a sweep on the test data? The energy rule, which keeps the directions that carry 95% of the difference variance and is the rule reported as *energy* r in Table 14, selects r between 21 and 343 over the eighteen settings there and does not help, because that spectrum is not effectively low-rank; selecting the rank by retrieval on a calibration half, then scoring it on the held-out half, matches or beats the fixed choice (0.957 against 0.948 on documents; equal at 0.696 on shapes). We use $r = 4$ throughout and report the held-out rule as the principled way to set it.

Calibration size and pairing (middle and right): eight paired contents already beat the raw representation (0.883 against 0.845) and the gain saturates at about $+0.09$, so this is a ceiling rather than a data limitation. Degrading the pairs deliberately costs little — 0.933 at 0% mismatched, 0.902 at 100% — but the stricter control below shows that the pairing still has to be content-preserving. The three curves are in Appendix I.1.

C.3 WHAT IT RECOVERS

The closed form reaches 0.945 at $r = 4$ against 0.950 for a trained mixing head, i.e. it recovers 95% of the gain without an objective; a per-dimension head reaches 0.876 and its ratio gain ($+0.5\%$) stays under the oracle bound of that grid ($+4.8\%$). Three sampler seeds give 0.950, 0.950, 0.950, so the gap is not optimisation noise: learning does not buy a per-dimension method anything.

Controls. Fitting the same procedure on mismatched pairs drawn from the evaluation grid gives 0.798, worse than doing nothing (0.845), while matched pairs give 0.917 — the gain comes from the pairing, not from low-rank structure. The subspace does not transfer: fitted on one domain and evaluated on another, off-diagonal entries land at or barely above the raw row (Table 20). The oracle diagonal fitted on half the contents and scored on the other half is 1–6 points smaller than the in-grid oracle and keeps the same ordering, so the bound does not depend on the oracle seeing the test data. With five fits and 2,000 clustered bootstrap resamples (Table 13), the retrieval gain is significant on the document grid for both encoders ($+0.066$ and $+0.023$, effect sizes 4.3 and 3.1) and for CLIP on shapes, and inside the noise for SigLIP on shapes and on the arXiv pages; we mark those two as non-significant rather than aggregating them. A full retrieval protocol with unknown queries (Table 21) improves R@1 and mAP and, on documents, the separation of known from unknown queries (AUROC 0.909 to 0.933).

Without paired data. Two random augmentations of a single image can stand in for the pairs (Table 16). We can say exactly when that works, because both conditions are measurable before

Table 8: **One fair table for the debiasing families.** All methods are fitted on the same half of the contents and scored with the same three metrics; the last column is balanced accuracy of a fresh probe for the nuisance label (chance $1/L$).

setting	method	ratio	NN_R@1	probe
documents/clip_openai	raw	2.12	0.845	0.802
	IDF weighting	2.05	0.352	0.282
	stable mask (25%)	2.16	0.938	0.754
	PCA removal (unsupervised)	1.93	0.845	0.782
	mean matching (labels)	2.39	0.895	0.167
	ridge residualisation (labels)	2.35	0.933	0.270
	label-PCA removal (labels)	2.00	0.883	0.837
	paired-difference PCA (pairs, ours)	2.33	0.917	0.627
documents/siglip	raw	2.36	0.971	0.929
	IDF weighting	2.28	0.319	0.266
	stable mask (25%)	2.49	1.000	0.881
	PCA removal (unsupervised)	2.32	0.993	0.917
	mean matching (labels)	2.80	1.000	0.167
	ridge residualisation (labels)	2.76	1.000	0.321
	label-PCA removal (labels)	2.29	0.988	0.925
	paired-difference PCA (pairs, ours)	2.73	1.000	0.639
real scans/clip_openai	raw	5.42	0.987	0.595
	IDF weighting	4.53	0.843	0.370
	stable mask (25%)	5.56	0.992	0.537
	PCA removal (unsupervised)	4.61	0.990	0.638
	mean matching (labels)	6.00	0.988	0.250
	ridge residualisation (labels)	5.82	0.995	0.302
	label-PCA removal (labels)	4.74	0.985	0.618
	paired-difference PCA (pairs, ours)	5.94	0.992	0.440
real scans/siglip	raw	5.03	0.990	0.588
	IDF weighting	5.33	0.822	0.345
	stable mask (25%)	5.71	0.997	0.535
	PCA removal (unsupervised)	5.06	1.000	0.618
	mean matching (labels)	6.02	0.997	0.250
	ridge residualisation (labels)	5.99	0.995	0.357
	label-PCA removal (labels)	5.29	1.000	0.615
	paired-difference PCA (pairs, ours)	6.50	1.000	0.420

fitting anything. First, the augmentation subspace has to cover the nuisance subspace: the largest principal cosine between them is 0.66 on rendered documents and 0.86 on the identity documents, and both are large enough in principle. Second, there has to be nuisance to remove at all: projecting the *true* subspace out gains $+0.084$ recall on documents but only $+0.012$ on the identity documents, whose geometry is content-dominated (both conditions are measured in Table 26). On documents both conditions hold and the fallback recovers about 60% of the gain of the paired fit ($0.845 \rightarrow 0.888$ against 0.917, Table 16); on identity documents the first holds and the second does not, and the fallback gains nothing ($0.790 \rightarrow 0.790$). A practitioner can therefore check the two numbers — subspace overlap and the gain of the paired fit on a handful of pairs — before trusting the fallback, rather than assuming it. Note also that the paired fit itself needs very little data: eight paired contents already beat the raw representation (Appendix C).

Downstream tasks. Clustering and a content probe behave as the geometry predicts (Table 17): clustering improves in every domain and dramatically where the nuisance dominates (ARI 0.080 to 0.466 on shapes, 0.561 to 0.677 on arXiv), retrieval improves in five of six domains, and the probe is at 0.96–1.00 throughout. On a pooled set of 794 real contents from three sources all three metrics are flat, which is what the transfer control predicted: a rotation fitted per domain does not help across sources.

C.4 AGAINST OTHER METHODS

Table 8 compares every family we can implement, including the two that need neither labels nor pairs. Three key interpretations. On the saturated scan grids retrieval no longer discriminates — three to six of the eight methods sit at 0.99 or above, depending on the encoder — so the ratio and the probe carry the comparison. The best *erasers* are label-based moment matching and ridge residualisation (probe 0.17–0.36 against 0.59–0.93 raw), while the rotation sits between the families on decodability (0.42–0.64). And an unsupervised rotation, plain PCA removal, is a strong retrieval baseline on these grids (0.99–1.00 on the scans), which a pair-free method must clear.

Averaged over twelve settings, LEACE-style erasure (Belrose et al., 2023) gains $+5.1$ retrieval points and wins in 7 of 12, INLP (Ravfogel et al., 2020) $+0.7$ in 6 of 12, and the rotation $+4.3$ in **12 of 12**, with the largest mean ratio gain ($+1.11$ against $+0.68$). The rotation does not beat label-supervised erasure on retrieval; it comes within a point of it while needing pairs instead of labels, and it is the only one of the three that never loses to the raw representation (Table 8).

The same comparison with the decodability column, for the family that needs nuisance labels, is Table 15.

The same pairs can also fit a *diagonal* end to end, which Appendix B.3 shows is a stronger ratio baseline than the plug-in weights and a weaker task baseline in most domains ($+18\%$ ratio but 0.916 retrieval against the plug-in's $+3\%$ and 0.952 on documents). It appears in that section rather than here because its point is about the bound, not about the method.

C.5 WHERE IT DOES NOT PAY

On the arXiv pages the rotation loses to the oracle for six of eight encoders, and the same happens at the weak end of the layout-strength sweep. The reason is Appendix B.5: a mild photometric nuisance leaves little to remove, and projecting directions out then costs more than it saves. The method is worth applying when the nuisance is real — re-layout, a second scan, a separate photograph — and we say so rather than quote an average that hides it.

D CASE STUDY: WHAT A TOP-k READOUT COSTS

The operations the previous sections bound are fixed functions of a representation. This section studies the one that is not: a top-k sparse code, whose selection depends on the input. The setting is a released aligned sparse autoencoder (Nasiri-Sarvi et al., 2026) whose latent space is shared by a CLIP image stream, a CLIP text stream and a DINO stream under a *global* top-k selection: one budget of 64 slots is divided between the three streams for every image, so the content stream does not receive a budget of its own.

D.1 DENSE HEAD AGAINST SPARSE CODE

Table 9: **Dense against sparse, same weights and same images.** The released aligned autoencoder on four domains: the embedding it was built from, its dense pre-selection head, the code its global top-k actually produces, and how many of the 64 slots reach the content stream.

domain	CLIP	dense head	top-k code	released budget
documents	0.924	0.921	0.112	17 of 64
shapes	0.795	0.750	0.161	16 of 64
real scans	0.997	0.997	0.777	19 of 64
arxiv	0.937	0.940	0.596	22 of 64

Reading the same weights over the same images two ways isolates the selection. The dense activations of the CLIP stream are at parity with the CLIP embedding the autoencoder was built on, on every domain (Table 9); the sparse code the model actually produces is far worse, by 22% of the retrieval on real scans and 88% on rendered documents. Its per-dimension spread — the coefficient of variation of the ratio profile c_d/v_d defined in Section 3.1 — explodes at the same time: 4.16 on rendered

documents and 30.1 on real scans, against 0.14 and 0.20 for the dense head on the same two domains (Table 27). A code that is zero almost everywhere has no well-defined per-dimension ratio, which is also why the oracle diagonal has nothing to exploit there (-0.0% to $+4.3\%$).

D.2 The budget, not the sparsity

The model's logits are dense pre-selection activations, so a budget can be imposed after the fact without retraining (Figure 7). On rendered documents retrieval rises monotonically with the budget, from 0.486 at $k = 16$ — the budget the model's own selection works out to, 16.8 active CLIP latents — to 0.917 at $k = 1024$, where it meets the dense head (0.921); the same monotone recovery appears on the other domains. Two details sharpen this (Table 18). At the same budget the *global* rule is far worse than a rule inside the content stream: 0.112 against 0.510 on documents, with a value threshold giving the same 0.510. The 0.112 is what the released model's global selection actually produces on the same images, i.e. the shared budget costs the content stream more than sparsifying it does; the 0.486 quoted above is what the same budget yields when it is spent inside the content stream. And when the rule chooses its own count, as a nucleus rule does, retrieval returns to the dense level (0.921) after keeping about 1,300 of the 8,192 latents.

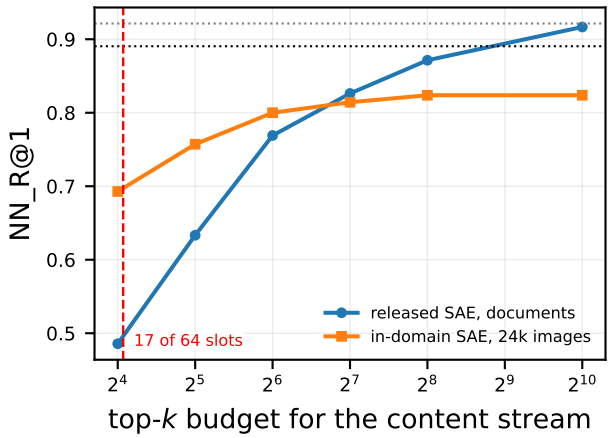

Figure 7: **Retrieval against the top-k budget** for the content stream, on rendered documents. The released model recovers monotonically towards its own dense head from the 17 slots it actually receives; an autoencoder trained in-domain on 24k images loses much less at the same budget.

D.3 A better dictionary, and what it changes

The control in an earlier version of this work was an in-domain autoencoder trained on 2,856 images, which reached a reconstruction error of 1.0×10^{-5} with 81% of its latents dead — it had largely memorised its training set, and its sparse code lost 51% of the retrieval of its own dense head. That is not a fair test of sparse coding, so we retrain the same architecture on 24,000 images (4,000 contents, six fixed layouts, texts disjoint from the evaluation grid): 120 epochs, 99 seconds on the GPU, final error 5.35×10^{-5}, 2.6% dead latents, explaining 95.9% of the feature variance.

The result (Table 25 in Appendix I) corrects the picture: with a dictionary that is not memorising its training set, the top-k loss at the same nominal budget is 10 points (0.890 to 0.800) instead of 44, and it is monotone in the budget. Sparse coding is not intrinsically destructive; what is destructive is a budget too small for the content stream, and a dictionary trained on too little data behaves as if its budget were small. The released aligned checkpoint remains the extreme case and the only one that drops below half.

D.4 READING

A fixed per-dimension operation cannot separate content from nuisance; an input-dependent one can additionally drop the content. Both failures are measurable before deployment — the spread predicts the first (Appendix B), the budget curve predicts the second — and the practical recommendation is narrow but concrete: when a shared top-k is used across streams, measure the content stream's retrieval as a function of its own budget, and do not read the dense head's quality as evidence about the code.

E DETAILED LIMITATIONS AND DISCUSSION

The bound is measured, not proved. Equations 5 and 6 are close to definitional: both sides are functions of the same profile. What is empirical is where real encoders sit, the resolution of the measurement, and the fact that the spread is not a nuisance-strength proxy. We have made those claims falsifiable rather than presenting the relationship as a theorem.

The bound covers plug-in weights, not the whole family. A diagonal fitted end to end on the same pairs escapes it by up to seven times in ratio (Appendix B.3), sometimes with better task metrics (shapes) and sometimes with clearly worse ones (identity documents, where the content probe loses 17 points). Our claim is therefore about the per-dimension methods people derive — IDF, masking, denoising, variance-ratio rescaling — and about the ratio as a diagnostic, not about the ceiling of everything one could do one dimension at a time.

What the estimator can and cannot resolve. Estimation noise inflates a measured spread, never deflates it, so every spread in this paper is an upper bound on the true spread and the true profiles are at least as uniform as measured. How large that inflation can be depends on how the nuisance varies across contents. A synthetic model with a known zero true spread in which one fixed nuisance vector is applied to *every* content estimates the per-dimension nuisance variance from the L realisations alone, and its measured spread then sits on a floor near $1.7/\sqrt{L}$ (1.04 at $L = 4$, 0.69 at 6, 0.23 at 48). Our grids redraw the nuisance for every content, so the same estimate averages over contents as well and that floor does not bind: dropping realisations on the document grid leaves the measured spread at 0.217 ($L = 3$), 0.202 ($L = 4$) and 0.201 ($L = 6$), and on the shapes grid it moves from 0.392 at $L = 8$ to 0.559 at $L = 3$, far below the synthetic floor of 1.04 at $L = 4$. Resolving differences much below these values therefore needs a different nuisance design, not more contents.

The domains are compromises in three different ways. The scan grids have real content but a digital second factor, because no public corpus of this size contains two physical scans of one page. The identity documents have real captures on both sides — 500 documents photographed five times each — but every capture in the `MIDV-500-filter` distribution is a camera photograph, so we see a change of capture event but not of device. And the two document grids used to fit and train are synthetic. We tried to do better: a public document-dewarping release we inspected is entirely synthetic — its second factors are digital warps of a page rather than new captures of it — MIDV-2020 (Bulatov et al., 2022) has no mirror we could reach, and the official distribution was unreachable from this machine.

The method needs paired data from the domain it is applied to. The transfer table is unambiguous — a subspace fitted elsewhere gives nothing — and the pooled downstream evaluation is flat for the same reason. Where pairs are unavailable, augmentation pairs recover about 60% of the gain when the true nuisance is a rendering transform and almost nothing when it is a capture event; the practical fallback is to measure the spread first and decide whether a per-dimension fix is worth attempting.

Statistical resolution. The paired intervals for the retrieval gain span about ten points on the document grid, and two of six settings are inside the noise and marked as such. Nuisance is never fully removed: every method we compare, including label-supervised erasure and an adversarial head, leaves it linearly decodable at 0.42–0.80 against a chance level of 0.17, and erasure in the closed-form sense is a statement about a linear probe rather than about meaning. Content matching

is also script-local in these encoders: same-topic retrieval between languages sharing a script is 0.51–0.94, while retrieval into or out of Chinese is 0.09–0.38.

Scope. Eight models at the scale that fits one consumer GPU, no fine-tuning, and no encoders trained with explicit invariance objectives: the question is what a practitioner can do with the weights as given. The practical takeaway is correspondingly small and testable — measure the spread before reaching for a per-dimension tool, prefer pairs and a rotation when they exist, and measure the content stream's budget when a shared top-k is in use.

F Domains, grids and pair construction

Every domain is a grid: a set of contents, a set of nuisance realisations, and one image for each pair. A *content pair* is two images of the same content under two realisations; a *nuisance pair* is two images of different contents under one realisation. Both are used by the decomposition of Appendix A, and the first is what the method of Appendix C consumes at fit time.

Rendered documents. Fourteen topic-labelled corpora contribute six documents each (84 contents); six layouts are drawn once from the six-parameter layout space (font size, page width, margin, line spacing, alignment, and the font itself) and reused for every content, so that a shared layout is a controlled contrast. Images are sized to their text; text length is held in a narrow band. The same generator produces the training grid listed in Table 4 (560 contents, $3{,}360$ images), of which 476 contents are used to fit the method and the trained heads of Appendix B.3, with different texts and a different layout draw. The autoencoder of the case study is trained separately on the larger grid of Appendix H ($4{,}000$ contents, $24{,}000$ images), and no text used to fit anything appears in the evaluation grid.

Shapes on textures. Sixteen contents (shape family crossed with colour) under eight nuisance tuples (background texture, rotation, scale, offset).

Real scans. FUNSD (Jaume et al., 2019), 199 scanned forms from the forms class of RVL-CDIP (Harley et al., 2015), each under four passes: identity, JPEG re-compression at quality 30, rescaling to half resolution and back, and a $0.6°$ rotation with an illumination gradient and a border.

Identity documents. The filtered distribution of MIDV-500 (`MIDV-500-filter`) (Arlazarov et al., 2019), 500 identity documents over 50 document types, five photographs of each document (2,500 images, differing in pose and framing). The distribution keeps the original `<type>/images/<group>/<doc>_<capture>.tif` file naming; we store downscaled JPEG copies and keep the mapping from each stored image to its document and capture index, so document identity and capture index are exact. The five images of one document come from one capture series, so they differ in pose and framing rather than in device.

Auxiliary capture grid. A second, HuggingFace mirror of the same corpus (`Noaman/midv500`) carries longer capture series: 297 photographs of ten documents, grouped by the document identifier preserved in the file names (28–30 captures each). This is the grid behind the rank curves, the fallback condition and the capture rows of Table 16; it is the one labelled *real captures* in Figures 5 and 8.

arXiv first pages. 95 papers drawn from twelve arXiv categories (cs.CL, cs.CV, cs.LG, math.OC, math.NT, astro-ph, cond-mat, hep-th, q-bio.QM, econ.EM, stat.ME, eess.SP), eight per category where the download succeeded and seven for one of them, rendered from the publisher PDF at two resolutions and in colour and greyscale, with and without a border.

Training texts. The method is fitted on texts drawn from a corpus of DBpedia abstracts (Lehmann et al., 2015) and, for the language check in Appendix E, from FLORES-200 (Costa-jussà et al., 2022). All texts used for fitting are disjoint from the texts of the evaluation grid.

Table 10: The eight frozen encoders. All are used with their released preprocessing and are never fine-tuned; image encodings are cached once per (encoder, grid).

name	architecture and weights	dimension
clip_openai	ViT-L/14, OpenAI	768
clip_laion	ViT-L/14, LAION-2B	768
clip_b32	ViT-B/32, LAION-2B	512
clip_datacomp	ViT-L/14, DataComp-XL	768
siglip	ViT-B/16 SigLIP, WebLI	768
eva02_b	EVA-02-B/16, merged-2B	512
convnext	ConvNeXt-Base-W, LAION-2B	640
dinov2_l	DINOv2 ViT-L/14 with registers (Darcet et al., 2024)	1024

G ENCODERS

The LAION-2B checkpoints (`clip_laion`, `clip_b32` and `convnext`) and the DataComp-XL checkpoint (`clip_datacomp`) are the `open_clip` releases (Cherti et al., 2023), trained on LAION-5B (Schuhmann et al., 2022) and DataComp (Gadre et al., 2023); the other rows are the weights of the models cited in Section 2.

H AUTOENCODERS

The case study of Appendix D uses three checkpoints, all with the same architecture family: an encoder affine map, a pre-bias subtracted before encoding, a global or stream-local top-k, and a decoder tied to the encoder weights with unit-norm columns.

- **Released aligned model** (Nasiri-Sarvi et al., 2026): 8,192 latents per stream, $k = 64$ selected globally across the CLIP image, CLIP text and DINO streams, used exactly as released. Its selection leaves 16–22 slots to the CLIP stream depending on the domain.
- **In-domain control, small**: 8,192 latents, $k = 64$, 120 epochs on 2,856 images, Adam, learning rate 10^{-3}; final training error 1.0×10^{-5} with 81% dead latents, i.e. largely memorised.
- **In-domain model, scaled**: the same configuration on 24,000 images (4,000 contents $\times$ six layouts, texts disjoint from the evaluation grid); 120 epochs, 99 seconds, final training error 5.35×10^{-5}, 2.6% dead latents.

I ADDITIONAL TABLES AND FIGURES

This appendix collects the tables and figures that support the sections above: the method curves and their controls, the rank and rank-selection curves, and the point-by-point tables behind the numbers quoted in the main text.

Four fits appear in these tables and are not interchangeable, which is why the same method carries different numbers in different places. A *held-out fit* fits the subspace on one half of a domain's contents and scores the whole grid (Tables 8, 16 and 23); the seed sweep, the rank selection and Table 28 go further and score the held-out contents alone, whose raw baseline is higher because a query is then compared with fewer distractors (0.922 against 0.845 on documents/CLIP). A *transferred fit* fits on the disjoint synthetic training grid and scores the domain (Tables 3 and 15). An *in-sample fit* fits and scores the whole grid, and so is optimistic (the rank sweep of Appendix C). The *unknown-query protocol* of Table 21 removes each query's own gallery entry, which is why its raw baseline is neither of the two above. On documents/CLIP the closed-form rotation reads 0.945, 0.929, 0.921 and 0.935 under the four.

I.1 METHOD CURVES AND CONTROLS

Table 11: **Plug-in against fitted diagonal weights**, fitted on one half of the contents (three seeds; the seed column is the standard deviation of the fitted gain across seeds) and scored on the other half, plus a split that fits on even nuisance realisations and scores the odd ones, and the topic-probe accuracy of the raw and fitted representations. The probe is reported as unavailable where the grid carries a single topic label, or one label per content, so that it would measure nothing.

domain	encoder	plug-in	fitted	seed sd	realisation	probe raw/fitted
documents	clip_openai	+3.4%	+18.1%	2.4	+16.4%	0.57/0.49
documents	siglip	+2.7%	+11.0%	1.3	+9.5%	0.54/0.54
shapes	clip_openai	+12.5%	+31.4%	0.6	+34.1%	n/a
shapes	siglip	+31.0%	+57.7%	7.5	+45.9%	n/a
real scans	clip_openai	+3.3%	+14.5%	1.3	+15.4%	n/a
real scans	siglip	+6.7%	+20.4%	1.1	+14.3%	n/a
large scans	clip_openai	+6.8%	+27.4%	2.6	+28.1%	n/a
large scans	siglip	+4.0%	+21.8%	0.5	+16.5%	n/a
arxiv	clip_openai	+2.6%	+7.4%	1.0	+19.4%	0.18/0.18
arxiv	siglip	+1.6%	+9.2%	1.6	+15.8%	0.14/0.16
identity documents	clip_openai	+3.0%	+20.4%	0.9	+24.1%	0.70/0.53
identity documents	siglip	+6.3%	+32.6%	0.5	+34.9%	0.39/0.26

Table 12: **The frontier of the ratio objective** on the whole grid, for weights whose dynamic range is bounded by c. The unconstrained end (c large) inflates the ratio by hundreds of percent while collapsing retrieval; the paper's plug-in oracle corresponds to a modest range and to the small gains quoted in Appendix B.

domain	encoder	clip $c=1$		clip $c=2$		clip $c=4$		clip $c=10$		clip $c=40$		clip $c=200$	
		gain	R@1	gain	R@1	gain	R@1	gain	R@1	gain	R@1	gain	R@1
documents	clip_openai	+0%	0.845	+28%	0.876	+68%	0.612	+113%	0.221	+126%	0.095	+126%	0.081
shapes	clip_openai	+0%	0.562	+52%	0.723	+119%	0.598	+288%	0.223	+289%	0.188	+289%	0.188
real scans	clip_openai	+0%	0.987	+26%	0.980	+60%	0.879	+98%	0.466	+102%	0.258	+102%	0.209
large scans	clip_openai	+0%	0.993	+35%	0.985	+93%	0.857	+149%	0.428	+157%	0.139	+157%	0.083
arxiv	clip_openai	+0%	0.881	+27%	0.881	+62%	0.744	+96%	0.435	+104%	0.179	+104%	0.165
identity documents	clip_openai	+0%	0.790	+25%	0.768	+76%	0.620	+114%	0.215	+139%	0.091	+140%	0.055

Table 13: **Effect sizes and significance** for the retrieval gain, over five fits and 2,000 clustered bootstrap resamples per setting; the effect size is the mean paired improvement over its standard deviation. Settings whose interval contains zero are marked not significant and are not aggregated into any claim.

setting	Δ R@1	95% interval	effect size	significant
documents/clip_openai	+0.066	[+0.036, +0.095]	4.28	yes
documents/siglip	+0.023	[+0.008, +0.036]	3.13	yes
shapes/clip_openai	+0.075	[+0.000, +0.143]	2.07	yes
shapes/siglip	+0.037	[-0.034, +0.101]	1.16	no
arxiv/clip_openai	+0.027	[+0.010, +0.041]	3.33	yes
arxiv/siglip	+0.001	[+0.000, +0.002]	0.86	no

Table 14: **Choosing the rank.** The energy rule — the rank that keeps 95% of the difference variance — selects very large ranks and does not help; choosing the rank on a calibration half of the contents and scoring it on the other half matches or beats the fixed $r = 4$.

domain	encoder	energy r	held-out r	fixed r=4	score
documents	clip_openai	139	32	0.948	0.957
documents	siglip	143	1	1.000	1.000
documents	dinov2_l	21	32	0.081	0.110
shapes	clip_openai	34	16	0.696	0.696
shapes	siglip	26	32	0.893	0.768
shapes	dinov2_l	27	32	0.679	0.679
real scans	clip_openai	139	8	0.993	1.000
real scans	siglip	113	1	1.000	1.000
real scans	dinov2_l	69	2	0.990	0.987
large scans	clip_openai	343	16	0.991	0.992
large scans	siglip	339	32	0.991	0.994
large scans	dinov2_l	244	32	0.983	0.991
arxiv	clip_openai	79	32	0.938	0.944
arxiv	siglip	81	2	1.000	1.000
arxiv	dinov2_l	49	16	0.979	1.000
identity documents	clip_openai	284	32	0.842	0.843
identity documents	siglip	297	32	0.899	0.901
identity documents	dinov2_l	279	32	0.908	0.911

Table 15: **Erasure with nuisance labels on the document grid.** Every method here is fitted on the disjoint synthetic training grid of Appendix F and scored on the whole document grid, so its closed form is the *transferred* version of Table 8's row of the same name and not the same fit (0.929 against 0.917 on documents/CLIP); the last column is balanced accuracy of a fresh probe for the nuisance label (chance $1/L$), so lower is better. The label-supervised methods retrieve better than the closed form, which is the closest method that uses neither nuisance labels nor a training objective.

representation	supervision	ratio	NN_R@1	nuisance probe
raw	–	2.12	0.845	0.802
closed form r=4	content pairs	2.21	0.929	0.750
LEACE-style	nuisance labels	2.15	0.952	0.714
INLP	nuisance labels	2.12	0.862	0.798
adversarial head	nuisance labels	2.36	0.950	0.583

Table 16: **Paired data against augmentations.** Same rank, same split, same evaluation grid; only the pairs used to estimate the subspace differ.

setting	pairs used at fit time	ratio	NN_R@1
documents/clip_openai	raw	2.12	0.845
	real pairs	2.33	0.917
	augmentations only	2.10	0.888
documents/siglip	raw	2.36	0.971
	real pairs	2.73	1.000
	augmentations only	2.37	0.990
real captures/clip_openai	raw	1.63	0.551
	real pairs	1.76	0.812
	augmentations only	1.56	0.571
real captures/siglip	raw	2.34	0.704
	real pairs	2.78	0.882
	augmentations only	2.42	0.714

Table 17: **Downstream tasks**, before and after the rotation: a content probe (capped at 60 classes for cost where a grid has more), k-means clustering and cross-layout retrieval. The pooled row uses 794 real contents from three sources and one rotation fitted on the pool.

encoder	setting	acc	NMI	ARI	NN_R@1
clip_openai	documents, raw	1.000	0.901	0.554	0.845
	documents, rotation	0.994	0.929	0.687	0.917
clip_openai	shapes, raw	0.906	0.440	0.080	0.562
	shapes, rotation	0.922	0.768	0.466	0.643
clip_openai	real scans, raw	0.975	0.976	0.817	0.987
	real scans, rotation	0.992	0.980	0.827	0.992
clip_openai	large scans, raw	1.000	0.946	0.461	0.981
	large scans, rotation	1.000	0.948	0.464	0.981
clip_openai	arxiv, raw	0.958	0.931	0.561	0.881
	arxiv, rotation	0.967	0.947	0.677	0.905
clip_openai	identity documents, raw	0.975	0.907	0.426	0.790
	identity documents, rotation	1.000	0.915	0.445	0.798
clip_openai	pooled (794 contents), raw	0.975	0.924	0.429	0.840
	pooled (794 contents), rotation	0.975	0.925	0.419	0.845

Table 18: **Selection rules at a comparable operating point**, from the released model's cached logits. "Mean kept" is the average number of entries left in the CLIP stream against 8,192 latents.

domain	selection rule	mean kept	NN_R@1
documents	dense (no selection)	8192	0.921
	global top-k (as released)	17	0.112
	per-stream top-k	17	0.510
	value threshold	17	0.510
	nucleus (90% mass)	1339	0.921
shapes	dense (no selection)	8192	0.750
	global top-k (as released)	16	0.161
	per-stream top-k	16	0.634
	value threshold	16	0.634
	nucleus (90% mass)	1169	0.768
real scans	dense (no selection)	8192	0.997
	global top-k (as released)	19	0.777
	per-stream top-k	19	0.915
	value threshold	19	0.915
	nucleus (90% mass)	1305	0.997
arxiv	dense (no selection)	8192	0.940
	global top-k (as released)	22	0.596
	per-stream top-k	22	0.758
	value threshold	22	0.758
	nucleus (90% mass)	1380	0.933

Table 19: **Rendering controls** (content/nuisance ratio). Sizing each image to its text keeps the content signal; fixing the canvas height removes it for CLIP and DINOv2, and the last row is degenerate for DINOv2 (no measurable nuisance variance).

rendering setting	clip_openai	siglip	dinov2_l
short text, sized to text	1.47	1.46	0.74
medium text, sized to text	1.63	1.63	0.29
fixed length, sized to text	1.56	1.55	0.17
fixed canvas height	0.74	1.30	0.04
fixed canvas, short text	1.18	1.85	n/a

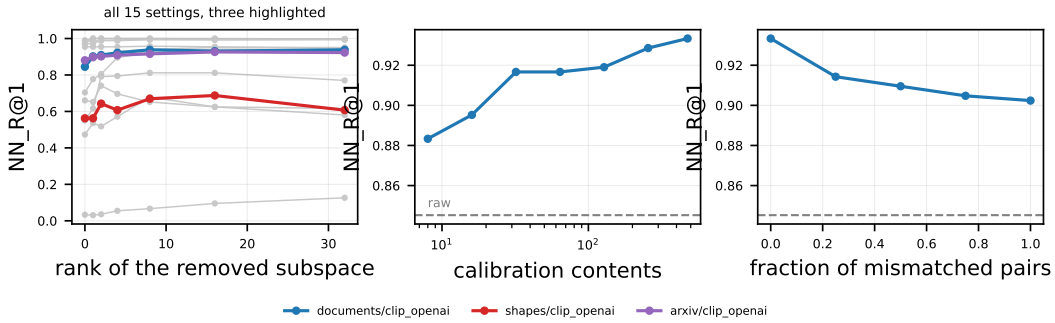

Figure 8: **The method's behaviour in its own hyperparameters.** Left: rank of the removed subspace (dashed lines are the raw representation). Middle: number of calibration contents. Right: fraction of deliberately mismatched pairs. Rank and calibration size saturate early; pairing noise degrades gracefully.

Table 20: **Transfer.** Ratio / NN_R@1 when the subspace is fitted on the row domain and evaluated on the column domain, against the raw row (2.12 / 0.845 for documents, 0.93 / 0.562 for shapes, 5.42 / 0.987 for scans, 3.63 / 0.881 for arXiv). Only the diagonal does anything.

fit \ evaluate	documents	shapes	real scans	arXiv
documents	2.33 / 0.917	0.93 / 0.598	5.35 / 0.992	3.57 / 0.884
shapes	2.11 / 0.850	1.48 / 0.643	5.41 / 0.988	3.63 / 0.881
real scans	2.12 / 0.848	0.93 / 0.580	5.94 / 0.992	3.62 / 0.881
arxiv	2.12 / 0.855	0.93 / 0.580	5.42 / 0.988	3.85 / 0.905

Table 21: **Retrieval with unknown queries.** Gallery = one rendering per content; unknown contents are absent from the gallery. The last column is the clustered bootstrap interval for the improvement in R@1.

setting		R@1	mAP	AUROC	bootstrap Δ R@1
documents/clip_openai	raw	0.894	0.931	0.909	+0.057 [+0.033, +0.082]
	rotation	0.935	0.960	0.933	
documents/siglip	raw	0.974	0.983	0.957	+0.023 [+0.008, +0.036]
	rotation	1.000	1.000	0.992	
arxiv/clip_openai	raw	0.890	0.931	0.821	+0.024 [+0.005, +0.040]
	rotation	0.921	0.952	0.818	
arxiv/siglip	raw	0.969	0.985	0.929	+0.002 [+0.000, +0.006]
	rotation	0.974	0.987	0.938	

Table 22: **All 50 points**, sorted by spread. *oracle* is the percentage change in the content/nuisance ratio from the diagonal rescaling fitted on the evaluation grid; *rotation* is the same for projecting out the top four content-paired principal directions.

domain	encoder	ratio	NN_R@1	spread	oracle	rotation
documents	eva02_b	1.52	0.757	0.122	+1.4%	+5.6%
documents	clip_b32	1.70	0.910	0.139	+2.0%	+4.5%
documents	clip_datacomp	1.81	0.924	0.143	+2.4%	+5.0%
identity documents	eva02_b	4.39	0.811	0.162	+2.5%	+0.4%
documents	clip_laion	1.92	0.940	0.162	+2.6%	+2.6%
documents	convnext	1.98	0.938	0.170	+3.6%	+3.7%
identity documents	clip_openai	4.56	0.790	0.173	+3.2%	+0.8%
identity documents	convnext	5.09	0.829	0.174	+2.9%	+5.5%
identity documents	clip_b32	3.89	0.798	0.183	+4.0%	+4.8%
identity documents	clip_datacomp	4.67	0.791	0.184	+3.7%	+6.5%
arxiv	siglip	3.78	0.954	0.195	+3.5%	+6.7%
identity documents	clip_laion	5.13	0.835	0.199	+4.5%	+9.4%
real scans	clip_openai	5.42	0.987	0.202	+4.8%	+5.9%
large scans	siglip	7.06	0.980	0.202	+4.3%	+12.4%
documents	clip_openai	2.12	0.845	0.203	+4.8%	+8.2%
documents	siglip	2.36	0.971	0.203	+4.1%	+18.6%
large scans	clip_datacomp	6.13	0.991	0.204	+6.4%	+25.0%
arxiv	clip_laion	4.21	0.944	0.204	+4.4%	-3.3%
arxiv	clip_datacomp	3.84	0.937	0.205	+6.0%	-5.8%
real scans	clip_datacomp	5.63	0.997	0.207	+4.9%	+21.7%
in-domain SAE	dense head	2.18	0.855	0.207	+4.8%	+14.2%
arxiv	clip_openai	3.63	0.881	0.211	+5.3%	-9.7%
large scans	clip_openai	6.60	0.981	0.211	+6.8%	+4.6%
arxiv	convnext	4.32	0.933	0.221	+4.3%	-10.5%
identity documents	siglip	4.25	0.863	0.243	+6.5%	-0.4%
large scans	clip_laion	6.12	0.988	0.245	+9.9%	+29.4%
arxiv	clip_b32	4.08	0.916	0.250	+5.4%	-9.1%
arxiv	eva02_b	3.53	0.849	0.254	+4.4%	-2.9%
real scans	siglip	5.03	0.990	0.256	+8.2%	+25.0%
large scans	clip_b32	7.54	0.986	0.258	+10.5%	+33.5%
real scans	clip_laion	4.96	0.998	0.263	+10.2%	+40.8%
real scans	clip_b32	6.50	0.997	0.270	+10.7%	+29.0%
large scans	convnext	6.41	0.989	0.273	+9.7%	+31.4%
large scans	eva02_b	7.91	0.959	0.273	+7.5%	+23.6%
identity documents	dinov2_l	6.80	0.857	0.278	+6.7%	-1.8%
real scans	convnext	4.75	0.997	0.290	+10.6%	+36.6%
real scans	eva02_b	6.98	0.993	0.302	+9.7%	+24.8%
shapes	clip_openai	0.93	0.562	0.396	+17.0%	+41.2%
large scans	dinov2_l	4.72	0.919	0.415	+20.6%	+57.5%
shapes	dinov2_l	0.75	0.473	0.469	+21.1%	+65.1%
shapes	clip_laion	0.81	0.643	0.469	+26.6%	+54.2%
shapes	eva02_b	0.97	0.679	0.489	+23.5%	+82.8%
real scans	dinov2_l	3.22	0.970	0.495	+37.9%	+115.4%
arxiv	dinov2_l	5.21	0.839	0.501	+29.1%	+42.9%
shapes	siglip	0.68	0.661	0.530	+36.3%	+104.5%
shapes	clip_datacomp	0.85	0.795	0.533	+32.4%	+77.4%
shapes	clip_b32	0.82	0.723	0.537	+35.2%	+78.0%
shapes	convnext	0.74	0.732	0.563	+33.5%	+98.3%
documents	dinov2_l	0.25	0.033	0.674	+51.8%	+336.5%
in-domain SAE	top-k code	1.61	0.419	1.127	+28.6%	+8.0%

Table 23: **Resampling error bars**, eight 80% subsets of the contents per (domain, encoder): mean and standard deviation of the spread and of the oracle headroom.

domain	encoder	spread mean ± sd	oracle mean ± sd
documents	clip_openai	0.209 ± 0.004	+5.1 ± 0.2%
documents	in-domain SAE, dense head	0.217 ± 0.003	+5.2 ± 0.2%
documents	in-domain SAE, top-k code	1.246 ± 0.062	+29.3 ± 1.6%
documents	clip_laion	0.172 ± 0.003	+2.9 ± 0.1%
documents	clip_b32	0.148 ± 0.004	+2.3 ± 0.1%
documents	siglip	0.212 ± 0.003	+4.5 ± 0.1%
documents	clip_datacomp	0.154 ± 0.002	+2.7 ± 0.1%
documents	convnext	0.183 ± 0.004	+4.0 ± 0.2%
documents	eva02_b	0.127 ± 0.002	+1.5 ± 0.1%
documents	dinov2_l	0.676 ± 0.005	+52.0 ± 0.7%
shapes	clip_openai	0.409 ± 0.012	+18.3 ± 1.2%
shapes	clip_laion	0.483 ± 0.013	+28.6 ± 1.3%
shapes	clip_b32	0.559 ± 0.027	+38.2 ± 3.8%
shapes	siglip	0.549 ± 0.019	+38.4 ± 3.2%
shapes	clip_datacomp	0.538 ± 0.009	+33.4 ± 1.1%
shapes	convnext	0.560 ± 0.016	+34.4 ± 1.4%
shapes	eva02_b	0.511 ± 0.017	+26.2 ± 1.4%
shapes	dinov2_l	0.470 ± 0.017	+21.2 ± 1.5%
arxiv	clip_openai	0.222 ± 0.005	+5.8 ± 0.4%
arxiv	clip_laion	0.221 ± 0.002	+5.1 ± 0.1%
arxiv	clip_b32	0.254 ± 0.007	+6.0 ± 0.3%
arxiv	siglip	0.209 ± 0.004	+4.2 ± 0.2%
arxiv	clip_datacomp	0.220 ± 0.005	+6.6 ± 0.4%
arxiv	convnext	0.225 ± 0.005	+5.0 ± 0.2%
arxiv	eva02_b	0.250 ± 0.009	+5.6 ± 0.4%
arxiv	dinov2_l	0.512 ± 0.010	+30.2 ± 1.3%
real scans	clip_openai	0.210 ± 0.003	+5.2 ± 0.2%
real scans	clip_laion	0.271 ± 0.003	+10.7 ± 0.3%
real scans	clip_b32	0.277 ± 0.003	+10.9 ± 0.5%
real scans	siglip	0.258 ± 0.004	+8.5 ± 0.3%
real scans	clip_datacomp	0.214 ± 0.001	+5.3 ± 0.1%
real scans	convnext	0.298 ± 0.003	+11.1 ± 0.2%
real scans	eva02_b	0.302 ± 0.005	+10.2 ± 0.3%
real scans	dinov2_l	0.500 ± 0.005	+38.4 ± 1.0%
real captures	clip_openai	0.257 ± 0.025	+6.1 ± 1.1%
real captures	clip_laion	0.361 ± 0.019	+12.7 ± 1.3%
real captures	clip_b32	0.327 ± 0.017	+10.4 ± 1.6%
real captures	siglip	0.414 ± 0.020	+17.4 ± 1.8%
real captures	clip_datacomp	0.294 ± 0.020	+8.4 ± 1.2%
real captures	convnext	0.323 ± 0.011	+9.8 ± 0.8%
real captures	eva02_b	0.310 ± 0.022	+8.7 ± 1.4%
real captures	dinov2_l	0.576 ± 0.041	+30.3 ± 5.3%

Table 24: **Resampling noise, summarised.** Worst within-encoder standard deviation of the oracle headroom and worst relative standard deviation of the spread, per domain.

domain	encoders	oracle sd (max)	spread rel. sd (max)
documents	10	1.6	5.0%
shapes	8	3.8	4.9%
arxiv	8	1.3	3.7%
real scans	8	1.0	1.5%
real captures	8	5.3	9.6%

Table 25: **The properly trained in-domain dictionary**, evaluated on the document grid: dense pre-selection head against top-k at several budgets.

readout	ratio	NN_R@1
top-k, $k = 16$	1.73	0.693
top-k, $k = 32$	1.68	0.757
top-k, $k = 64$	1.67	0.800
top-k, $k = 128$	1.70	0.814
top-k, $k = 256$	1.73	0.824
top-k, $k = 1024$	1.73	0.824
dense (pre-selection)	2.17	0.890

Table 26: **The two conditions for the augmentation fallback**, for the two settings where both subspaces were measured: the largest principal cosine between the augmentation subspace and the true nuisance subspace, then the retrieval of the raw representation, of the true subspace projected out, and of the augmentation fallback.

setting	max cosine	raw	true subspace	augmentations
documents/clip_openai	0.66	0.845	0.929	0.888
identity documents/clip_openai	0.86	0.790	0.802	0.790

Table 27: **The spread of the three readouts** of Table 9, on the same images: coefficient of variation of the ratio profile c_d/v_d, the quantity defined in Section 3.1.

domain	CLIP	dense head	top-k code
documents	0.143	0.136	4.157
shapes	0.524	0.539	5.579
real scans	0.206	0.204	30.082
arxiv	0.207	0.196	8.552

Table 28: **Retrieval of the fitted diagonal**, the escape route of Appendix B.3, in all twelve settings: means over the three seeds of the raw representation, the plug-in reweighting and the fitted diagonal, and the last two in retrieval points. Eight of the twelve settings lose retrieval. The fit and the score use the held-out half of the contents that this appendix's seed sweep uses, so the raw column is higher than in Tables 2 and 8, which score the whole grid (0.922 against 0.845 on documents/CLIP); only the differences are comparable across protocols.

setting	raw	plug-in	fitted	Δ pts
documents/clip_openai	0.922	0.952	0.916	-0.6
documents/siglip	0.976	0.998	0.978	+0.2
shapes/clip_openai	0.643	0.696	0.756	+11.3
shapes/siglip	0.756	0.875	0.881	+12.5
real scans/clip_openai	0.998	0.998	0.987	-1.1
real scans/siglip	0.996	0.997	0.998	+0.2
large scans/clip_openai	0.994	0.994	0.988	-0.6
large scans/siglip	0.996	0.996	0.994	-0.3
arxiv/clip_openai	0.933	0.924	0.905	-2.8
arxiv/siglip	0.993	0.993	0.991	-0.2
identity documents/clip_openai	0.833	0.836	0.820	-1.3
identity documents/siglip	0.889	0.891	0.885	-0.4

Table 29: **The twelve settings behind the erasure summary of Section 4.3.** Change in cross-layout retrieval, in points, against the raw representation of the same setting; every method is fitted on one half of the contents and scored on the whole grid. LEACE wins in seven settings, INLP in six, and the rotation is positive in all twelve.

setting	raw	LEACE	INLP	rotation
documents/clip_openai	0.845	+11.4	+2.4	+7.1
documents/siglip	0.971	+2.9	+1.2	+2.9
documents/dinov2_l	0.033	+1.7	+0.0	+2.9
shapes/clip_openai	0.562	+17.9	+2.7	+8.0
shapes/siglip	0.661	+20.5	+1.8	+2.7
shapes/dinov2_l	0.473	+12.5	+0.0	+14.3
real scans/clip_openai	0.987	-1.0	+0.0	+0.5
real scans/siglip	0.990	-1.2	+0.2	+1.0
real scans/dinov2_l	0.970	+1.2	+0.3	+2.2
arxiv/clip_openai	0.881	-2.8	+0.0	+2.5
arxiv/siglip	0.954	-1.8	+0.0	+0.4
arxiv/dinov2_l	0.839	+0.0	+0.0	+7.0

J COMPUTE AND RUNTIMES

Every experiment in this paper was run on a single 6 GB consumer GPU and a desktop CPU; no experiment required more. Feature extraction is the dominant cost: the CLIP ViT-L/14 encoder runs at 20.6 images per second on grids up to 504 images, and the largest grid used here (24,000 rendered documents) takes about 20 minutes to render, 35 minutes to encode and 99 seconds to train the autoencoder. Peak GPU memory is 2.4 GB with an encoder resident, and the autoencoder training itself uses 183 MB. All analyses after feature extraction (the decomposition, the bound, the rank and calibration curves, the bootstrap) run on cached features in seconds to minutes.