THE FLOOR THAT BARELY FALLS: UNTRAINED ENCODERS AS THE OPERATIVE NULL FOR CONCEPT-DISCOVERY METRICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept-discovery methods are scored with NMI, MIG, DCI and relatives, and often against an implicit floor of zero: of 8 sampled papers reporting one of these metrics, none reports a chance level. That a floor exists is known: Carbonneau et al. (2022) checked, and found most of these metrics *do* score near zero on a random representation — all but DCI under a lasso regressor. We measure a strictly larger floor, for a different reason. Their random baseline is noise independent of the factors; ours is an *untrained encoder applied to the real inputs*, which carry information about the factors that generated them, and that difference is the whole paper. On our $K{=}4$ cell the untrained-encoder floor is 0.116 NMI at 240 held-out samples, in the decomposition run of the same cell whose canonical estimate is 0.115. Adjusted mutual information (Vinh et al., 2010) removes 0.081 of it. Under the equal-frequency binning these metrics use the correction is affine, so it cannot reorder methods — but it rescales every difference by about 1.09, which matters against a fixed equivalence margin. What it leaves, 0.035, is the operative quantity, and it barely shrinks: 0.030 at 1200 samples, a $1.16\times$ change across a fivefold rise in evaluation data while the removable part falls $5.6\times$. Matching inflates it only at small samples: in a Gaussian reference that reproduces this floor, 90% of it is population alignment at 240 samples and 99% at 1200, and cross-fitting the matching agrees. Its size is a property of the configuration, and not in the direction a reader would guess: coarsening the discretisation from 10 bins to 3 shrinks the total floor $2.3\times$ while raising this component from 30% of it to 85%. It has to be measured per cell rather than looked up. Doing so changes conclusions, including ours: on a tabular benchmark we build, one cell places no task-supervised method above its null at the sample size we ship it at, and the gain a task label plus its architecture buys over that architecture's own untrained state differs by an order of magnitude across architectures (a post-hoc largest-to-smallest ratio of $12.5\times$ whose interval is informative only at its lower end, $4\times$) — and the smallest gain clears its floor only narrowly (under Holm, not under plain Bonferroni). It also disqualifies an equivalence margin this literature would call tight: $\delta{=}0.05$ NMI is 1.7–$2.2\times$ the largest excess over chance among the methods it is used to separate, and 3.3–$25\times$ their between-method range. Read against measured chance, concept labels still beat every architecture we tested, in every cell that resolves.

1 INTRODUCTION

A concept-discovery method is judged by how well its latent coordinates line up with ground-truth concepts, and that alignment is reported as NMI, MIG, DCI or a relative of them. These are bounded scores, conventionally read against a floor of zero: a method at 0.30 is taken to have recovered a third of something. The floor is not zero. Hungarian matching over K predicted and K true coordinates will find a good assignment between two *independent* random vectors, plug-in mutual-information estimators are biased upward at small samples, and quantile binning turns a continuous nuisance into a discrete coincidence. The result is a chance level that is substantial, that depends on the evaluation configuration — the number of concepts K, the number of held-out samples, the factor cardinalities, the estimator — and that is almost never reported.

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.

How large, and which part. We measure it directly (§3): draw an encoder that has learned nothing, score it under exactly the split, binning and matching the paper uses, and repeat. Separating the components is what makes the measurement worth doing (§3). The estimator-only (random-code) floor is 0.016 NMI on the dSprites grid at $n_{\text{test}}{=}1000$ and 0.095 on our $K{=}4$ cell at 240; it scales with the evaluation-set size and is flat in K — across five standard grids K ranges 3–7 while the floor stays inside 0.012–0.016 — and an adjusted estimator removes most of it (Vinh et al., 2010). What remains after that adjustment does not have a closed form. Reported effects in this regime are frequently smaller than it.

What changes when you subtract it. Conclusions, including ours. We build **ConceptDiscoveryBench**, a tabular concept-discovery benchmark with controlled ground truth, and score six method families (eight configurations, counting the VAEs' two supervision variants) under a supervision-matched protocol. Measured against chance rather than zero, one of its five distinct cells places *every* task-supervised method at or below the floor, and two more clear it by a few thousandths of NMI (§3). Each cell therefore ships its *resolution* — the sample size at which a trained method first clears its own floor — the way a detection limit is part of an instrument. The same correction disciplines equivalence testing: an equivalence margin of $\delta{=}0.05$ NMI, which this literature would call tight, is 1.7–$2.2\times$ the largest excess over chance among the methods it is asked to separate, so a positive verdict at that margin bounds the difference far more loosely than the reader is likely to assume (§4.4).

Contributions.

(C1) **Measured chance levels** for NMI, MIG, DCI-D and ACS, per evaluation configuration, together with the procedure that produces them (§3). That untrained encoders retain task-relevant structure is known (Saxe et al., 2011), and that they score well through these metrics is Cao et al. (2022). C1 measures *which part* persists, per configuration, shows by cross-fitting and a known-population reference that the persistent part is information rather than matching optimism, and ships it with the benchmark — including for the factor grids of five standard disentanglement benchmarks, computed from their published specifications, no images required.

(C2) **A scale for the equivalence margin**: δ judged against the above-chance range of the methods compared rather than adopted by convention. On our headline cell $\delta{=}0.05$ NMI is 1.7–$2.2\times$ (shipped, corrected block) the largest excess over chance of the three methods it calls equivalent, and 25–$3.3\times$ their range, so a positive verdict there separates nothing (§4.4).

(C3) **ConceptDiscoveryBench and a supervision-matched protocol**: methods are grouped by whether they may consume x only, task labels y, or concept labels c, and compared only within a group, with splits, seeds, metrics and preprocessing pinned by shared utilities. Each cell ships its chance level and resolution limit (App. E).

(C4) **A negative result that survives the correction**: read as excess over measured chance, concept labels beat every architecture we tested, in every cell that resolves, and "task-supervised" is not one band. Every task-supervised method in the headline cell clears its own architecture-matched floor under Holm, our declared family-wise correction (five of six under plain Bonferroni, and five against each run's own initial weights at five seeds, six at ten), by between 0.015 (β-VAE+sup) and 0.186 (NBM). That span measures method-and-label packages, not the label's value alone: Level-0 β-VAE's matched excess, 0.004, is not resolved from its floor ($p{=}0.087$), nor is the VAE task-label increment 0.011 (App. B.2). The $12.5\times$ ratio of the extremes is post-hoc and poorly bounded (Fieller 4–$339\times$; denominator CV 0.35), so we treat it as exploratory and the six absolute excesses as the result. The widest gap in the sorted excesses, $4.6\times$, falls between ConceptNAM and DetNAM, not between families. Against the linear null (conservative here: every matched null sits below it) the same test gives four under Holm (App. B.1, `matched_verdict.json`).

Scope, and what this paper does not claim. The benchmark is tabular, with two cross-domain probes (raw $64{\times}64$ dSprites, two clinical datasets) treated as portability checks rather than as evidence about images or medicine. Chance levels are a property of a metric and a configuration, so C1 transfers; C4 is about our generators. On eight public dSprites checkpoints of one shared architecture, the correction reorders nothing but moves every improvement ratio, and the untrained floor is 46% of the plain VAE's DCI-D (App. M.2). Corrections relative to earlier versions of this work are collected in App. Z.2. ConceptNAM, the architecture we previously proposed, is here a

case-study Level-1 method and it loses. Every number is read from a shipped JSON artifact and checked mechanically. The gates are matchers, not proofs, so we state their reach. Of 284 mean±std cells, 235 are bound to a named artifact, to their row's method and to their column's metric; the other 49 (prose, captions, tables citing no artifact) bind only to `results/` as a whole. 64 derived ratios must re-derive at printed precision; four declared quantities — the per-cell nulls — must print one value document-wide; a figure caption may quote only values its generator reads. Bare table cells are checked advisorily; bare appendix prose and caption arithmetic are not covered. Corrections to our own earlier claims are in App. Z.

2 Related Work

The metrics, and what they are scored against. DCI (Eastwood & Williams, 2018), MIG (Chen et al., 2018) and attribution sanity checks (Adebayo et al., 2018) supply the measurements this paper uses, and the large-scale study of Locatello et al. (2019) established the comparison protocol most of the field inherited: many models, many seeds, these metrics, reported as levels. What that protocol does not carry — and what we add — is a floor. The estimators involved are not unbiased at finite n: plug-in mutual information is biased upward, and the Hungarian matching that makes these scores permutation-invariant will find a good assignment between independent random vectors. Neither issue is new in statistics; what is missing is the per-configuration measurement that turns them from a caveat into a number. Chance correction is standard practice elsewhere, and part of it is already solved in closed form. The adjusted Rand index subtracts an expected value analytically (Hubert & Arabie, 1985); adjusted mutual information does the same for MI under a permutation model (Vinh et al., 2010); the plug-in bias of a discrete MI estimator is characterised (Paninski, 2003); and permutation nulls are routine in neuroimaging (Nichols & Holmes, 2002).

Closest prior work. Carbonneau et al. (2022) already state that a disentanglement metric "should attribute the minimum score to a completely random *or fully entangled* representation" (emphasis ours), and measure what the metrics they survey score on a random one — most near zero, but DCI's modularity and compactness above 0.3 under a lasso regressor, which they attribute to the ℓ_1 penalty manufacturing spurious weight differences. That is the calibration argument, four years before us, and any claim that these metrics are read against an unexamined zero must be made against it. It is not yet routine: of 34 recent arXiv papers we screened, the 8 that report one of these metrics report no chance level for any of them (`reporting_survey.json`, a keyword sample, not a census). The closest exception is Klotz et al. (2026), who pick among concept-annotation metrics by which separate trained sparse autoencoders from untrained ones and subtract a per-configuration untrained-SAE score — but report no spread around that zero. The data-*dependent* random representation closest to ours is Carbonneau et al.'s $z = vR$, R sparsified random, used to probe *sample efficiency* on the ground-truth factors rather than to calibrate on the inputs (App. E.1).

Two things separate this paper. The first is which null does the calibrating: theirs is independent of the factors, so it measures estimator bias — the part an adjustment removes and that vanishes as $1/n_{\text{test}}$ — while ours is the second half of their own desideratum, the entangled representation. After the adjustment both are meant to remove, ours is $3\times$ what theirs leaves at $n_{\text{test}}{=}240$ and $6\times$ at 450, widening with n because only theirs shrinks. The second is that ours projects the *inputs*, not the factors, so it measures what the inputs themselves already carry about the factors. Scoring it still needs the factor values, exactly as the metrics do; it applies wherever they apply.

Nor is the mechanism ours. Randomising parameters, keeping the real inputs and re-running the identical pipeline is Adebayo et al.'s sanity check (2018); that an untrained encoder on real inputs retains task-relevant structure is Saxe et al. (2011), restated for sentences by Wieting & Kiela (2019). Cao et al. (2022) run exactly this check inside this literature — randomly initialised encoders scored by five of these metrics, above well-trained ones[1] — and Heap et al. (2026) find the same one subfield over (there the probed transformer is randomised, not the encoder), likewise recommending randomised baselines as routine; Kim et al. (2018) already test each concept direction against re-learned ones, retraining every CAV ~ 500 times on resampled sets of real counterexample images. The rest of the apparatus is likewise borrowed: simulation-estimated NMI nulls (Zhang, 2015) and

[1] E.g. a 1000-d random encoder scores FactorVAE 61.4 on dSprites against a trained 10-d model's 58.6. Their reading is unfairness across representation dimension, not a floor; full numbers and the attribution in App. E.1.

the $1/n$ estimator-bias law (Paninski, 2003), selection bias under repeated comparison (Romano et al., 2016), reporting a score as a difference from a control (Hewitt & Liang, 2019), margins judged against the smallest effect of interest (Lakens et al., 2018; Card et al., 2020), and the metrics' defects as metrics (Sepliarskaia et al., 2019). Gates & Ahn (2017) are the closest ancestors of §3's move: the choice of random model changes rankings and the evaluation against a random baseline, so it "should be carefully justified". Gröger et al. (2026) apply that pattern to representational-similarity metrics against a *permutation* null, removing the global-similarity evidence for a published convergence result while its local evidence survives. A permutation null is the class §3 measures to leave ours. Mahon et al. (2024) adopt Kuhn–Munkres alignment and a chance-corrected score, and Hsu et al. (2023) ship the fix as a metric (InfoMEC, which renormalises by the equal-information level and puts KSG in place of binned MI; App. A). Both are the fix, not the inflation. What we add is none of these: Cao et al. show *that* an untrained encoder scores well; we measure *which part* persists — decomposed against the closed-form-correctable parts, per configuration — and ship it as each cell's denominator, with a detection limit.

This bounds what is new here, and we would rather bound it ourselves. Most of the floor is what a chance adjustment removes, and that is *not* a measurement problem — the adjusted estimator removes it (§3). Anyone reporting these metrics should be using the adjusted variants, and the fact that this literature does not is worth saying but is not a discovery. What survives the adjustment we split by what the computation does, not by cause: the *matching gain* of the Hungarian-optimal assignment over a random one, and the *random-assignment level* an untrained encoder attains on real inputs averaged over pairings. The matching gain is not a pure selection bias — it mixes in-sample optimism with alignment that exists in the population — and the second is information the inputs genuinely carry about the factors that generated them. Neither behaves like the removable part: on the encoder null, across a fivefold rise in n_{test}, the matching gain falls $1.3\times$ and the random-assignment level $1.0\times$, against $5.6\times$ for the removable part (App. A, Table 2). In a Gaussian setting where the population is known and the headline floor is reproduced, 90% of the adjusted floor at $n_{\text{test}}{=}240$ and 99% at 1200 is population alignment, and cross-fitting the matching on held-out halves agrees (optimism under 5% of the adjusted floor from 2400). From about a thousand samples the part that barely falls is information, not a matching artefact; at the smallest n, matching accounts for a tenth to a third of it (App. A).

Negative and corrective results in concept learning. This paper belongs to a line of work that re-examines what concept models are actually doing. Margeloiu et al. (2021) questioned whether CBMs learn the concepts they are supervised on; Parisini et al. (2025) quantify concept leakage; and Almudévar et al. (2026) argue on information-theoretic grounds that training a component to predict a concept does not make it carry *only* that concept. **Our result is orthogonal to theirs.** Their target is the *representation* — what a reported score licenses you to conclude about what the component encodes. Ours is the *score*: how much of that number an untrained encoder already gets. The two bracket a reported number from below and from above, by different means: theirs analytically, ours by measurement.

Identifiability and supervision. Locatello et al. (2019) showed unsupervised disentanglement is impossible without inductive bias; Khemakhem et al. (2020) recovered identifiability through auxiliary information, consistent with the supervision-level grouping we use. Our equivalence classes are the classical ones: linear mixings of independent non-Gaussian sources are identifiable up to permutation and scaling, Gaussian sources only up to rotation (Darmois, 1953; Comon, 1994). Ahuja et al. (2022) show task labels *can* identify latents up to permutation given a rich enough task family — the positive counterpart to our Level-1 result, which is therefore about a supervision *budget*, not about task labels in principle. The parametric shadow is sufficient dimension reduction: in a single-index model only the index span is estimable (Li, 1991).

Concept Bottleneck Models. Koh et al. (2020) introduced CBMs with labeled concepts. We include CBM as the Level-3 supervised anchor and study how evaluation conclusions change when CBM is given partial rather than full labels.

Recent concept-discovery variants. CBM-style discovery has been extended with per-concept embeddings (Espinosa Zarlenga et al., 2022), concept uncertainty (Kim et al., 2023), CLIP-derived concepts (Oikarinen et al., 2023), leak-proof representations (Marconato et al., 2022), causal grounding

(Lippe et al., 2022), and a concept-quality suite (Bortolotti et al., 2024); TabCBM (Espinosa Zarlenga et al., 2023) is our Level-3 tabular reference, and Chen et al. (2024) the identifiability counterpart. None reports a chance level (App. V, Q16).

Neural Additive Models. Agarwal et al. (2021) introduced NAMs for feature-level interpretability; NODE-GAM (Chang et al., 2022) and NBMs (Radenovic et al., 2022) extend them. ConceptNAM moves additivity to the *concept* level and is a case-study model here, not a claimed contribution. Vanilla NAM has no concept-indexed attribution and cannot be scored on these metrics, so it enters as a task-RMSE reference; NODE-GAM and NBM are adapted into Level-1 variants with the same K-concept bottleneck as *DetNAM*, which is ConceptNAM without the VIB (App. E).

3 THE FLOOR

A concept-recovery score is only interpretable against the score a method with no information already attains. This section measures that value. Nothing here involves training: the whole point is what the evaluation procedure returns when there is nothing to evaluate.

The floor has three components, and a closed form reaches only one. *Random code* draws $z \sim \mathcal{N}(0, I_K)$ independent of the data, isolating what the estimator returns for a representation that knows nothing. Adjusted mutual information (Vinh et al., 2010) subtracts the expectation under a permutation model in closed form and removes most of it — on our headline cell $0.095 \to 0.011$.

But the correction does less than it appears to. Its expectation term depends only on the binned marginals, which equal-frequency binning fixes for every representation, so AMI is an *affine* map of the matched score here — slope $\max(H)/(\max(H) - \mathbb{E}[\mathrm{MI}]) \approx 1.09$. It therefore cannot reorder anything, but it does multiply every difference by that slope, so an equivalence verdict against a *fixed* δ is not invariant unless δ is rescaled with the axis. What survives it is 0.035.

Random encoder draws $z = xW$ with W untrained. This is the operative null for any encoder-based method — what it scores before learning anything — and it is a different animal. Because the concepts generated x, *any* projection of x carries some information about them, so this is not chance in the estimator's sense.

We test that directly rather than inferring it from the code null. Applying AMI to the *encoder* null leaves 0.035 at n_{test}=240 and 0.030 at 1200 — a $1.16\times$ change across a fivefold rise in evaluation data, against a $5.6\times$ fall in the term the same correction removes (we call it the plug-in term below; it also contains the normaliser's rescaling, so it is not a pure estimator bias). The permutation model does not reach this. Three properties make this the quantity that has to be measured rather than looked up:

The discretisation is part of the configuration. We use 10 quantile bins per dimension throughout, and that default is not innocuous: across 3 to 20 bins the floor moves $0.050 \to 0.296$ ($5.9\times$), and the plug-in share of it moves $15\% \to 92\%$, so *which* component dominates is a property of the binning while *that the floor is far from zero* is not (App. A). The first rule at the close of §5.4 therefore has to pin and report the discretisation alongside the floor, or the excesses it yields are not comparable across papers.

- **It varies several-fold across datasets at identical (K, n_{test}).** At n_{test}=600 our benchmark's cells run 0.044–0.126 while their code floors sit flat at 0.035–0.040. The spread is the data, not the configuration.
- **It barely shrinks with n, though the floor as a whole does.** What survives the adjustment goes $0.035 \to 0.030$ as n_{test} rises $240 \to 1200$ while the removable part collapses $0.081 \to 0.014$; on dSprites the code floor falls $0.016 \to 0.002$ from 10^3 to 10^4 while the encoder floor falls only $0.057 \to 0.050$. The total floor does shrink, roughly as $n^{-0.3}$ to $n^{-0.9}$.
- **It is the part that coarsening cannot remove.** At ten bins it is the smaller share (0.035 of 0.116); the plug-in term is larger and an adjustment takes it away. The two move oppositely as the binning coarsens and cross between 5 and 8 bins: at 3 bins the plug-in term is 0.007 and the survivor 0.042, 85% of a floor $2.3\times$ smaller (App. A). It also survives a binning-free (KSG) estimator: under it the code floor falls 2.0–$2.3\times$ with n and this component does not shrink, at every k tried

(App. A). *So coarsening the discretisation removes the component we are not talking about and makes ours dominant.* That reading is ours, not Carbonneau et al.'s (App. E.1).

Both are scored through the same `compute_nmi` / `disent_metrics` path, split utility and Hungarian matching as every number in this paper (`analyze_chance_floor.py`).

Which measurement is canonical. Three drivers estimate this cell's floor: 0.1143, 0.1156 (200 draws each) and 0.1154 (2000; `null_calibration.json`). **The last is canonical and every excess here subtracts it.** They agree to 0.0014, which is more than two of the six floor-clearing margins. The same rule now covers every cell measured twice: at $K{=}4$, $n{=}1500$ the canonical estimate is 0.072 (`sblock_nulls.json`; a second run of the same cell gives 0.074), at $n{=}2000$ it is 0.060 (cells grid; the bench-protocol run gives 0.062) — each pair within draw noise, each cell's excesses subtracting its canonical value only.

Where each component is large. Table 29 gives the estimator-bias floor at the factor grids of five standard disentanglement benchmarks — computable from the published construction, which fixes the marginals that set the chance level — and, for dSprites, the operative floor too. On those grids the bias floor is 0.012–0.016 at $n_{\text{test}}{=}10^3$ and an order of magnitude lower at 10^4; ours is 0.095 at 240. *That gap is n_{test}, not K:* across the five grids K moves $3 \to 7$ while the floor stays inside 0.012–0.016, and within our own cells it falls $0.095 \to 0.017$ as n_{test} goes $240 \to 1200$. The bias floor is essentially c/n_{test} — predictable, and removable.

The operative floor behaves differently, and that is the point. For dSprites it is 0.057 at $n_{\text{test}}{=}10^3$ ($3.6\times$ the bias floor) and 0.050 at 10^4, where the bias floor has vanished and the ratio is $32\times$; ours is 0.115. The same-null comparison is therefore 0.057 against 0.115 — a factor of two, not of ten — and in both the component that does not go away with more data dominates.

The null must come from the encoder class being scored. Both floors above use a random *linear* encoder. Whether that makes them lower bounds we measured: it holds on one domain and fails on the other. On the dSprites probe the matched null — its own untrained ConvNet — is 0.067 (p_{95} 0.078) against the linear 0.044, and **under it the probe's entire task-supervised half is at or below chance** (0.042, 0.055, 0.062); we had read those three as an ordering and no longer do (App. M.1). On the headline tabular cell the gap runs the other way: every matched null is slightly *below* linear (0.108–0.114 vs 0.116, `arch_matched_null.json`), so there our null is conservative. The sign is not a property one can reason to: it varies by domain, and on the dSprites tabular probe it varies *by architecture inside one domain* — three of four matched nulls fall below the linear one there while NODE-GAM's rises $1.18\times$ above it (App. M). That is the argument for measuring it per cell and per encoder class rather than arguing it.

Consequence for our own benchmark. Applied to ConceptDiscoveryBench the correction is expensive. At the benchmark's protocol ($n{=}2000$) the MultiFactor cell has a null of 0.126 with a 95th percentile of 0.151, above *all four* of its task-supervised entries, none more than 0.0006 above its mean: no Level-1 entry there beats chance (best Welch $p{=}0.49$). The $K{\geq}16$ cells do clear their floors, by 0.003–0.015 NMI above each cell's own 95th percentile — enough to say a method beat noise; whether two methods differ is a pairwise test we do not run. Rather than drop or defend these cells we characterise them: each ships its *resolution*, the n at which a trained method first clears the cell's (linear) null for our training budget, as n grows the training set too; on MultiFactor it is training size, not evaluation size, that binds (Table 21). Three of five resolve at $n{\leq}500$ for both a Level-1 and a Level-3 method; MultiFactor resolves at 500 for CBM but beyond the shipped $n{=}2000$ for DetNAM (grid point 8000, a grid label rather than a sharp threshold) — why its Level-1 row carries no floor verdict. A detection limit belongs on an instrument.

4 Evaluation Framework

4.1 Supervision-matched grouping

We classify every baseline by what it is permitted to consume during training, and compare only inside a class:

- **Level 0** — inputs x only. Reconstruction or unsupervised objectives: β-VAE (Higgins et al., 2017), FactorVAE (Kim & Mnih, 2018).
- **Level 1** — inputs x and the scalar task label y, never the concepts. ConceptNAM, DetNAM, NODE-GAM, NBM, β-VAE+sup, FactorVAE+sup; vanilla NAM is Level-1 but has no concept output, so it appears only as a task-RMSE reference.
- **Level 3** — inputs, task labels, and ground-truth concept labels c: CBM-joint (Koh et al., 2020), iCBM. (Level 2, partial or noisy concept labels, is a continuum we sweep in App. B rather than a class.)

Comparisons are made *within* levels. Cross-level gaps are attributed to supervision, not architecture.

4.2 Leakage-free splits

A shared utility produces deterministic disjoint train/test (or train/val/test) splits from one seed; any scaler or PCA is fit on the training split alone (pinned by a regression test). Test metrics are computed once. *On checkpoint selection:* the headline and benchmark runs use a fixed epoch budget and select nothing — val-R^2 is what could legally be selected on at Level 1, and App. I replicates under it. In semi-supervised experiments, labelled samples come from the training split and are excluded from NMI evaluation.

4.3 Multi-metric suite

1. **NMI**: Hungarian-matched normalized mutual information (implemented once in our shared metrics module, `cdnam.metrics`; the ACS definitions are in App. Z.1).
2. **MIG** (Chen et al., 2018): mutual information gap — detects whether each factor is captured by a single latent.
3. **DCI** (Eastwood & Williams, 2018): disentanglement / completeness / informativeness via gradient-boosted regressors. We follow the Eastwood–Williams convention and fit the D/C importance regressors on the same held-out samples the scores are computed on (`cdnam/disent_metrics.py`), so DCI-D and DCI-C read as *diagnostic factor–code dependency on the evaluation set*, not held-out generalisation; informativeness uses a separate 50/50 probe split and *is* a held-out R^2. Every benchmark we compared against reproduces this in-sample/held-out mix as published; we flag it so neither DCI-D nor DCI-C is used for model selection here. Figure 4 reads four *interpretability* diagnostics together — attribution sharpness and density, cross-seed stability, active units — and no single one separates the methods the same way.
4. **ACS** (new): Attribution Consistency Score. For S seeds with attribution matrices $A^{(s)} \in \mathbb{R}^{K\times p}$, align rows across seeds with a Hungarian solver on cosine distance, then report the mean pairwise cosine $\mathrm{ACS}_{\cos}$ and top-k Jaccard $\mathrm{ACS}_{\mathrm{IoU}}$ over aligned rows (full expressions in App. Z.1). $\mathrm{ACS}\to 1$ means a method recovers the same concept set deterministically across seeds; $\mathrm{ACS}\to 0$ means different concepts each run. Inspired by attribution sanity checks (Adebayo et al., 2018).

A composite *Interpretability Proxy Score*, $\mathrm{IPS} = \frac{1}{4}(\widetilde{\mathrm{Gini}} + \widetilde{\mathrm{Smooth}} + \mathrm{ACS}_{\mathrm{IoU}} + \mathrm{AU}/K)$ with per-method min-max normalisation, is defined in App. Z.1 but is *not* used to rank anything in this paper: a method lacking a component (e.g. β-VAE has no attribution matrix) scores 0 on it, which structurally favours architectures that expose attributions. We report the components separately instead.

4.4 Statistical protocol

Five replicates throughout (the headline cell additionally at ten, App. C); all $\pm$ are sample standard deviations. Differences use paired t-tests with BH correction and are reported with Cohen's d. In place of observed power — a monotone transform of the p-value, which cannot separate a small effect from a weak design — we report the minimum detectable effect: at n=5, α=0.05 and 80% power a paired test here resolves d=1.68, at n=10, d=1.00. Equivalence uses TOST (Schuirmann, 1987) with Bonferroni adjustment across the family, and we judge δ against the above-chance range of the

Table 1: Headline comparison on the synthetic concept benchmark (K=4, n=800, 5 seeds, held-out; App. E repeats at n=2000, App. C at 10 seeds). The upper block is `exp6_full_comparison.json` and the lower `bench_corrected.json`; the null row is `null_calibration.json`, the canonical estimate of §3. Values are mean ± std (ddof=1). The last column is the Bonferroni-adjusted TOST p for equivalence to ConceptNAM at δ=0.05; bold means *proven equivalent within* $\pm\delta$, and since δ is 1.7× the triplet's largest excess over chance on this block (2.2× corrected), read a positive verdict as an upper bound (§4.4). The null row is an untrained encoder under the same split and metric; **read every column as excess over it**. The lower block reruns the cell with un-swept flags reselected on validation task RMSE, which uses no concept labels; cite the lower block. Draw counts, task-R^2 readings, the MIG/DCI-D chance tests and the multiplicity discussion are in App. B.2.

Level	Method	NMI	MIG	DCI-D	task R^2	TOST p_{adj}
3	**CBM joint**	**0.634±0.011**	**0.506±0.016**	**0.941±0.010**	**+0.956**	1.000
1 (NAM)	**DetNAM**	0.192±0.035	0.060±0.049	0.212±0.069	+0.658	1.000
1 (VAE)	FactorVAE+sup	0.145±0.006	0.008±0.003	0.084±0.026	+0.182	**0.0011**
1 (VAE)	ConceptNAM	0.144±0.010	0.013±0.008	0.069±0.034	+0.011	—
1 (VAE)	β-VAE+sup	0.143±0.011	0.020±0.008	0.088±0.018	−0.007	**0.0033**
0	β-VAE	0.117±0.011	0.020±0.011	0.038±0.020	n/a	0.102
–	*untrained encoder (null)*	0.115±0.011	0.021±0.011	0.050±0.027	n/a	—
Same cell, flags reselected on validation task RMSE (`bench_corrected.json`)						
3	CBM joint	**0.618±0.009**	**0.492**	**0.935**	**+0.950**	—
1 (NAM)	NBM	0.295±0.103	0.156	0.310	+0.854	—
1 (NAM)	DetNAM	0.246±0.056	0.106	0.268	+0.803	—
1 (NAM)	NODE-GAM	0.141±0.008	0.012	0.118	+0.631	—
1 (VAE)	ConceptNAM	0.139±0.007	0.006	0.093	+0.092	—
1 (VAE)	FactorVAE+sup	0.125±0.008	0.032	0.056	+0.595	—
1 (VAE)	β-VAE+sup	0.123±0.012	0.033	0.065	+0.577	—
0	β-VAE	0.112±0.005	0.014	0.043	n/a	—

methods being compared rather than adopting it by convention: δ=0.05 NMI is 1.7× the *best* of those methods' excess over the floor on the block the TOSTs run on (2.2× corrected) and 25× (3.3×) their between-method range, so a positive verdict separates nothing. Every floor-clearing verdict in this paper uses one rule — a one-sided Welch t against the method's own null, both variances carried, Holm-corrected over the family; tables that print a p_{95} screen agree with it (49/50 resolution tests, every first-resolving n), as does a seed-paired variant, and in a negative control its error rate stays below nominal while the screen's runs at about twice nominal (App. B.1). App. B.1 gives both in full, with the noise-calibration rule for δ that we proposed and then withdrew.

5 Experiments

The case-study architecture, and what it is not. ConceptNAM — per-feature MLPs, a deep variational information bottleneck (Alemi et al., 2017), and a concept-additive head (Fig. 3) — is the ablation target of §5.4 and nothing more: the identifiability theorems about that head have hypotheses that fail on this generator and no experiment cites one; the negative result rests on measurement (Apps. S, T). Its loss, the audit fix that made the `no_IB` ablation a real one, and the per-driver configurations are in Apps. Q and P. All main-text numbers load from released JSON artefacts; the claim-to-artefact map is App. W.

5.1 Does supervision level predict NMI? (Hierarchy validation)

Result: under the corrected fixed-W protocol (App. Z, Audit B1) the hierarchy holds across levels but stratifies weakly within Level 1. As excess over the canonical null, Level-0 sits at the floor (−0.003), VAE-family Level-1 reaches +0.008–+0.023 (almost entirely chance), and CBM +0.502. On this block the triplet is *not* tightly equivalent: pairwise differences reach 0.015 and all three pairs pass TOST only from ε=0.029 on raw p, or 0.036 under the triplet's own Bonferroni family (m=3) (`tost_corrected.json`; App. V, Q14); the shipped block, where Table 1's TOST column

runs, lies within ± 0.002 and is equivalent from 0.0145. §5.3 finds that indistinguishability to be regime-dependent; the Level-1-vs-Level-3 gap is not.

Supervision-matched must be verified, not assumed. Being *allowed* to consume y is not the same as consuming it, so we report task-R^2 by default. Of the three Level-1 VAE methods, β-VAE+sup reaches $R^2 = -0.007$ and ConceptNAM $+0.011$ — neither distinguishable from a constant predictor (p=0.18, p=0.054) — and FactorVAE+sup only $+0.182$ (p=0.003); the two methods that do fit their supervision (DetNAM $+0.658$, CBM $+0.956$) are exactly the two that separate on NMI. Table 38 gives the mechanism: β-VAE+sup has $0/4$ active units on every seed, ConceptNAM $3.4/4$ with the posterior near the prior — the signal never reaches the concept axes.

A near-zero R^2 is not an architectural limit but a scale bug — reconstruction averaged over features while the KL is summed makes the effective penalty βp, parking the grid in the collapse regime. Correcting it lifts both supervised VAEs to $R^2 = +0.57$ with all latents active, while NMI moves *down* (0.121, 0.122; App. H): the equivalence survives consuming the supervision. A matched comparison is only meaningful between methods that demonstrably used the supervision, so we report held-out task R^2 and active units alongside any matched-supervision claim. The Level-1-vs-Level-3 gap is unaffected: CBM's advantage is measured between two models that both fit their targets.

5.2 What does NMI miss?

Scored through one shared attribution utility (every method on the same input-feature axis), CBM is the most stable and most concentrated, and two divergences NMI hides remain: β-VAE+sup's posterior collapse and ConceptNAM's $\sim 23\times$ smaller off-axis intervention variance *after normalising by on-axis sensitivity* (the raw $289\times$ is mostly its $12\times$ less responsive head; `exp14_causal.json`). ACS needs a floor too: Hungarian matching aligns two *random* attributions to $\mathrm{ACS}_{\cos}$=0.728 at K=4, p=16, so that column spans 0.73–1.00, not 0–1 (Table 4; Apps. B, W).

5.3 Is the equivalence a floor effect?

An equivalence where every method fails says nothing about architecture, so we move difficulty along two axes that leave the supervision budget untouched — concept count K and label noise σ (`exp_floor_control.py`). Two things separate. The within-family equivalence *does not survive an easier problem*: at K=2 the triplet's pairwise differences grow from 0.0008–0.0020 to 0.030–0.074 and TOST establishes equivalence for $0/3$ pairs, so the claim is scoped to the K=4 regime we measure. And *"Level-1" is not a single quantity*: as excess over each cell's own null, the fraction of CBM's gain a Level-1 method captures runs 4.4%–34.1%, and no aggregate is robust (max, mean and median move in different directions), so we put no single number on what a task label buys (App. B.3).

5.4 Direct VIB ablation

Three controlled experiments (~ 200 runs) test the three image-domain VIB claims — concept emergence without concept labels, implicit pruning, stability at high p — and none holds on tabular data: a deterministic sibling recovers more concepts from task labels alone, no method prunes, and at $p \geq 64$ the VIB collapses ConceptNAM's posterior to a single dimension (App. R).

What to do instead. **Measure the floor** per metric, cell, discretisation and *encoder class* and report raw score, floor and excess (a learning gain, not final quality); **size δ against the excesses it adjudicates** (δ is 1.7–2.2$\times$ the largest here); **verify the supervision budget was consumed**. Three rules we had not followed ourselves. Three more follow from the checks in the appendix. **Measure the floor under the scores' own preprocessing**: whitening alone halves it on one of our cells. **Decide with a test over seeds, not a percentile screen**: in a negative control the Welch and seed-paired rules stay below their nominal error while a p_{95} screen runs at about twice it, and a run's own initial weights are the most direct reference for its gain. **Evaluate on at least a thousand samples**, where the floor that survives adjustment is information rather than matching optimism. None of this is expensive: `cdnam.floor` measures a headline-cell floor in 27 s on one CPU thread.

AI USE STATEMENT

An earlier version of this statement said LLM assistants were used "only for routine copy-editing". That understated their role and we correct it. LLM assistants were used substantially in preparing this work: they drafted and revised manuscript prose, including several of the retractions recorded in App. Z; they wrote or rewrote most of the analysis and evaluation drivers that ship with the artifact — among them `analyze_chance_floor.py`, `exp_floor_control.py`, `exp_encoder_ablation.py`, `exp_vae_scale_ablation.py`, `exp_bench_corrected.py`, `analyze_tost_eps_sweep.py`, `analyze_clinical_variance.py`, `analyze_null_mc_error.py` and `analyze_acs_matched_null.py`, a list that is illustrative rather than exhaustive — and the unit tests in `tests/`; they proposed analyses whose results this paper reports, including the chance-level nulls, the equivalence-margin calibration and the configuration ablations; they assisted retrieval and discovery of related work (including locating Cao et al. (2022), Hsu et al. (2023) and Skirzyński et al. (2026), each subsequently verified against the primary source) and the verification of reference metadata against DBLP, Crossref, arXiv, OpenReview and JMLR; and they drafted the revision of one proof (the well-definedness argument in Remark 2, since withdrawn), which the authors checked. Every number so produced is regenerated by a driver in Table 39 and was checked against its artifact by the authors. No LLM is a component of any method under study, of the training pipeline of those methods, or of the data generators; and the scientific claims and the decisions about what to retract are the authors'. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work conforms to the ICLR Code of Ethics. The synthetic and public tabular datasets used in the main results (Synthetic-Concept, MultiFactor, ScalingSynth, dSprites, Adult Income, California Housing, UCI Heart Disease, Pima Diabetes, breast cancer) carry no direct privacy risk; all real-world data is loaded via cached, already-publicly-released sources. Our contribution is methodological: a matched-supervision benchmark and an auditing toolkit aimed at *reducing* the risk that concept-discovery systems are deployed on the basis of inflated or leakage-driven results. We do not anticipate dual-use or disproportionate-harm concerns beyond those inherent to any interpretability research.

REPRODUCIBILITY STATEMENT

Every number in this paper is traceable to a released artifact and to the driver that wrote it. That is weaker than "regenerable bit-for-bit from the released code", and we state where it falls short: the artifacts were produced across a revision history (ii), a re-run driver prints $\pm$ values that need one post-processing step (below), and the ACS attributions drift on re-derivation by up to 0.17 on Level-1 rows, which comes from re-training variability whose source we have not isolated, not from extraction (after Table 20); no end-to-end re-run of every table on the final code state has been done. (i) The **ConceptDiscoveryBench** datasets (Synthetic-Concept, MultiFactor, ScalingSynth, dSprites, and the four real-world tabular sets) are generated or downloaded deterministically by the loaders under `cdnam/data/` from fixed seeds documented in App. E, which also states the fixed shared W that the Synthetic-Concept and ScalingSynth generators enforce across replicate seeds and the per-seed W that MultiFactor regenerates by design. (ii) Every reported number comes from a JSON file under `results/` carrying a top-level `manifest`. *What that manifest is, precisely, because an earlier version of this paragraph overstated it and a reviewer checked, and because the counts themselves went stale once and are now generated rather than typed:* all 87 shipped artifacts record a schema version and an SHA-256 `source_hash` over `cdnam/`, `baselines/` and `experiments/`, but those hashes take 30 distinct values and none equals the shipped tree's, because the artifacts were produced across the revision history rather than in one run. The `source_hash` is therefore a *change detector* — two artifacts with the same hash saw the same code — and not a pointer to a recoverable state; without a git tree we cannot offer one, and we no longer claim to. 45 of the 87 record a `dataset_seed`; none records a field named `w_hash` (the fixed-W invariant is asserted instead inside the drivers and, for the bench artifacts, in a `shared_w_hash` field); and 87 carry the backfill stamp of `scripts/backfill_manifest.py`, so for those the `timestamp_utc` is a backfill time and the `command` field is empty — with 16 exceptions that

were stamped for a different missing key and kept a real command. What does hold, and is what a reviewer needs: every shipped driver runs standalone and regenerates its own artifact, and the claim-to-artifact-to-driver map of Table 39 names the script for each. (iii) All per-method seeds are derived via the bijective offset table in `cdnam/utils/seed.py` (§W); the same seed always produces the same sample-and-method combination, eliminating the silent `hash(name)` aliasing flagged in our self-audit. (iv) The statistical protocol — paired t-tests, Wilcoxon, Cohen's d and TOST equivalence at $\delta = 0.05$ — is fixed in §4 and implemented in `cdnam/stats.py` (re-exported through `cdnam/utils/stats.py`); the `exp_bench.py` driver writes per-method JSONs and the `run_significance_analysis.py` aggregator computes paired p-values, Cohen's d, and TOST verdicts as a post-hoc summary into `results/exp6_statistical_analysis.json`. *On the $\pm$ convention, which this statement previously described incorrectly:* every value in `results/` is a sample standard deviation (ddof=1), matching the tests quoted beside it. Most drivers were written with ddof=0 and their stored values were recomputed from the per-seed values by `scripts/unify_std_ddof.py`, so **re-running a driver reproduces every mean but prints a $\pm$ that is $\sqrt{n/(n-1)}$ too small until that script is run**; `scripts/check_paper_pm.py` verifies the shipped state. This is a wart we are disclosing rather than hiding: the fix is to change the drivers, which we have not done because it would invalidate the `source_hash` on every artifact. (v) The 48 unit/regression tests in `tests/` pin seven critical correctness fixes from our audit (concept recomputation under distribution shift, CBM column-wise concept masking, deterministic seeding, image-mode parity, the ConceptNAM `label_mask` all-False alignment-omission semantics, the `import experiments` namespace-shadow guard, and the dSprites train-only PCA fit invariant) and unit-test the primitives those fixes concern — `compute_nmi`, the shared split utilities and `cdnam.stats` — which previously had no direct tests at all. Two of the audit-fix tests were rewritten after we found they passed with the original defect restored: the ablation-toggle test evaluated at epoch 0, where the warm-up weight is 0 and no toggle can matter, and the DCI probe-split test asserted only that two keys existed. Both now fail when the pre-fix code is reinstated, which we verified. (vi) A locked dependency set (`requirements.lock`) and a reviewer-facing `Dockerfile` materialise the exact PyTorch / NumPy / SciPy / scikit-learn / matplotlib versions — and their transitive closure, without which the `-no-deps` install the Dockerfile performs left several of them unimportable — used to produce every number in this paper; `scripts/ci_smoke.sh` chains the test suite, the data-asset SHA-256 verifier (`scripts/verify_data_assets.py` against `notes/data_manifest.json`), and a paper–code consistency check into a single pre-flight (a few minutes on CPU, dominated by the test suite and a throwaway LaTeX build) that reviewers can run before the long benchmark drivers.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper/8160-sanity-checks-for-saliency-maps.

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34:4699–4711, 2021. URL https://papers.nips.cc/paper/2021/hash/251bd0442dfcc53b5a761e050f8022b8-Abstract.html.

Kartik Ahuja, Divyat Mahajan, Vasilis Syrgkanis, and Ioannis Mitliagkas. Towards efficient representation identification in supervised learning. In *Conference on Causal Learning and Reasoning (CLeaR)*, 2022. URL https://proceedings.mlr.press/v177/ahuja22a.html.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyxQzBceg.

Antonio Almudévar, José Miguel Hernández-Lobato, and Alfonso Ortega. There was never a bottleneck in concept bottleneck models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Fy7V5dalvX. arXiv:2506.04877.

Samuele Bortolotti, Emanuele Marconato, Tommaso Carraro, Paolo Morettin, Emile van Krieken, Antonio Vergari, Stefano Teso, and Andrea Passerini. A neuro-symbolic benchmark suite for concept quality and reasoning shortcuts. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2024. doi: 10.52202/079017-3679. arXiv:2406.10368.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016. doi: 10.18653/v1/k16-1002.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3d-shapes, 2018.

Jinkun Cao, Ruiqian Nai, Qing Yang, Jialei Huang, and Yang Gao. An empirical study on disentanglement of negative-free contrastive learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1210–1222. Curran Associates, Inc., 2022. doi: 10.52202/068431-0089.

Marc-André Carbonneau, Julian Zaidi, Jonathan Boilard, and Ghyslain Gagnon. Measuring disentanglement: A review of metrics. *IEEE Transactions on Neural Networks and Learning Systems*, 35 (7):8747–8761, 2022. doi: 10.1109/TNNLS.2022.3218982. Early access November 2022; version of record 2024.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. With little power comes great responsibility. In *Proceedings of EMNLP*, pp. 9263–9274, 2020. doi: 10.18653/v1/2020.emnlp-main.745.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. NODE-GAM: Neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=g8NJR6fCCl8.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018. URL https://papers.nips.cc/paper/7527-isolating-sources-of-disentanglement-in-variational-autoencoders.

Wenlin Chen, Julien Horwood, Juyeon Heo, and José Miguel Hernández-Lobato. Leveraging task structures for improved identifiability in neural network representations. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=WLcPrq6pu0. arXiv:2306.14861.

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994. doi: 10.1016/0165-1684(94)90029-9.

Georges Darmois. Analyse générale des liaisons stochastiques. *Revue de l'Institut International de Statistique*, 21(1):2–8, 1953. doi: 10.2307/1401511.

Cian Eastwood and Christopher K I Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1555.

Mateo Espinosa Zarlenga, Zohreh Shams, Michael Edward Nelson, Been Kim, and Mateja Jamnik. Tabcbm: Concept-based interpretable neural networks for tabular data. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=TIsrnWpjQ0.

Alexander J. Gates and Yong-Yeol Ahn. The impact of random models on clustering similarity. *Journal of Machine Learning Research*, 18(87):1–28, 2017. URL https://jmlr.org/papers/v18/17-039.html.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021. doi: 10.1145/3458723.

Muhammad Waleed Gondal, Manuel Wüthrich, Đorđe Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: A new disentanglement dataset. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/hash/d97d404b6119214e4a7018391195240a-Abstract.html.

Fabian Gröger, Shuo Wen, and Maria Brbić. Revisiting the platonic representation hypothesis: An aristotelian view. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://arxiv.org/abs/2602.14486. arXiv:2602.14486v2.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Automated interpretability metrics do not distinguish trained and random transformers. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=USyGD0eUod. arXiv:2501.17727v2.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of EMNLP-IJCNLP*, pp. 2733–2743, 2019. doi: 10.18653/v1/d19-1275.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Kyle Hsu, Will Dorrell, James C. R. Whittington, Jiajun Wu, and Chelsea Finn. Disentanglement via latent quantization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45463–45488. Curran Associates, Inc., 2023. doi: 10.52202/075280-1969.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/bf01908075.

Aapo Hyvärinen, Ilyes Khemakhem, and Ricardo Monti. Identifiability of latent-variable and structural-equation models: from linear to nonlinear. *Annals of the Institute of Statistical Mathematics*, 76(1):1–33, 2023. doi: 10.1007/s10463-023-00884-4.

Ilyes Khemakhem, Diederik P Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217, 2020. URL https://proceedings.mlr.press/v108/khemakhem20a.html.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677. PMLR, 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In *International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/kim23g.html.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018. URL https://proceedings.mlr.press/v80/kim18b.html.

Jonas Klotz, Cassio F. Dantas, Pallavi Jain, Diego Marcos, and Begüm Demir. Evaluating the interpretability of sparse autoencoders with concept annotations. In *European Conference on Computer Vision*. Springer, 2026. URL https://arxiv.org/abs/2606.24716. Spotlight; arXiv:2606.24716.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie E Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. In *Conference on Causal Learning and Reasoning*, 2022. URL https://proceedings.mlr.press/v177/lachapelle22a.html.

Daniël Lakens, Anne M. Scheel, and Peder M. Isager. Equivalence testing for psychological research: A tutorial. *Advances in Methods and Practices in Psychological Science*, 1(2):259–269, 2018. doi: 10.1177/2515245918770963.

Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. doi: 10.1109/cvpr.2004.1315150.

Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991. doi: 10.1080/01621459.1991.10475035.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Efstratios Gavves. Citris: Causal identifiability from temporal intervened sequences. In *International Conference on Machine Learning*, 2022. URL https://proceedings.mlr.press/v162/lippe22a.html.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pp. 4114–4124. PMLR, 2019. URL https://proceedings.mlr.press/v97/locatello19a.html. Best Paper Award.

Louis Mahon, Lei Sha, and Thomas Lukasiewicz. Correcting flaws in common disentanglement metrics. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=c8WJ4Vozb2.

Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretable, leak-proof concept-based models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1542.

Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? In *ICLR Workshop on Responsible AI*, 2021. URL https://arxiv.org/abs/2105.04289. arXiv:2105.04289.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Thomas E Nichols and Andrew P Holmes. Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Human Brain Mapping*, 15(1):1–25, 2002. doi: 10.1002/hbm.1058.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FlCg47MNvBA.

Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003. doi: 10.1162/089976603321780272.

Enrico Parisini, Tapabrata Chakraborti, Chris Harbron, Ben D. MacArthur, and Christopher R. S. Banerji. Leakage and interpretability in concept-based models. *arXiv preprint arXiv:2504.14094*, 2025. URL https://arxiv.org/abs/2504.14094. v3, 2026.

Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. Neural basis models for interpretability. In *Advances in Neural Information Processing Systems*, 2022. doi: 10.52202/068431-0612.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/e07413354875be01a996dc560274708e-Abstract.html.

Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17(134):1–32, 2016. URL https://jmlr.org/papers/v17/15-627.html.

Andrew M. Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y. Ng. On random weights and unsupervised feature learning. In *International Conference on Machine Learning*, pp. 1089–1096, 2011. URL https://icml.cc/2011/papers/551_icmlpaper.pdf.

Donald J Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, 1987. doi: 10.1007/bf01068419.

Anna Sepliarskaia, Julia Kiseleva, and Maarten de Rijke. How to not measure disentanglement. *arXiv preprint arXiv:1910.05587*, 2019. URL https://arxiv.org/abs/1910.05587. v3, 2021.

Julian Skirzyński, Harry Cheon, Shreyas Kadekodi, Meredith Stewart, and Berk Ustun. Measuring what matters: Synthetic benchmarks for concept bottleneck models. *arXiv preprint arXiv:2606.04326*, 2026. URL https://arxiv.org/abs/2606.04326.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010. URL https://jmlr.org/papers/v11/vinh10a.html.

John Wieting and Douwe Kiela. No training required: Exploring random encoders for sentence classification. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BkgPajAcY7.

Pan Zhang. Evaluating accuracy of community detection using the relative normalized mutual information. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(11):P11006, 2015. doi: 10.1088/1742-5468/2015/11/p11006.

APPENDIX OVERVIEW

The appendix is organised in six parts. Part I holds everything the paper's measurement claims rest on: the floor's decomposition and the checks of what it is, the statistical protocol, and the verdict rule's measured error rates. Parts II–III are the benchmark, its controlled experiments and its extensions beyond the synthetic generator. Part IV is the ConceptNAM case study, Part V the theory (background only; no experiment depends on it), and Part VI discussion, reproducibility and the self-audit. Corrections to earlier versions of this work, including those to appendix material, are collected in App. Z.2.

I. The floor and the statistics behind every verdict
App. A The floor: decomposition, cross-fitting, a known-population reference, sensitivity
App. B Full experimental tables
App. C Ten-seed replication of the headline cell
App. D Power supplement and TabCBM head-to-head
II. The benchmark and its controlled experiments
App. E ConceptDiscoveryBench: a benchmark for tabular concept discovery
App. F ACS-IoU sensitivity to top-k
App. G Encoder-configuration ablation and full rerun
App. H Reconstruction-scale ablation for the VAE family
App. I Validation-stopped training: number robustness
App. J NAM (Agarwal 2021) task-performance comparison
App. K Protocol-vs-architecture ablation
App. L Architecture and multi-metric figures
III. Beyond the synthetic benchmark
App. M Cross-domain validation tables
App. N Clinical tabular probe
App. O Per-dataset real-world results
IV. The ConceptNAM case study
App. P ConceptNAM configurations
App. Q ConceptNAM loss decomposition and toggle semantics
App. R VIB ablation: full sweeps
V. Theory, and its limits
App. S Identifiability theorems: full proofs
App. T Identifiability: what the theorems do and do not buy
VI. Discussion, reproducibility and audit
App. U Extended discussion: when-to-use guidance and limitations
App. V Anticipated reviewer questions
App. W Reproducibility manifest
App. X Hyperparameters for the headline runs
App. Y Compute
App. Z Self-audit and corrections to earlier versions

A THE FLOOR: DECOMPOSITION, CROSS-FITTING, A KNOWN-POPULATION REFERENCE, AND SENSITIVITY

Table 2: The floor's three components as the evaluation set grows (K=4, p=16, 10 bins, 5 seeds × 40 draws per cell; `floor_decomposition.json`). Columns are named by what they compute. "Adjusted-out" is NMI minus its chance-adjusted version (the expected-MI term removed from the numerator *and* the normaliser rescaled, so not a pure plug-in bias; §3 and the rest of this appendix call it the plug-in term for short). "Matching gain" is the Hungarian-optimal sum of adjusted MI minus the random-assignment expected sum, over the same normaliser: it mixes in-sample selection optimism with alignment present in the population, and is not a selection bias as such. "Random-assignment level" is the latter expected sum. Every term is measured on the encoder null. Both survivors are nearly flat — the matching gain falls $1.3\times$ and the random-assignment level $1.0\times$ across a fivefold rise in n_{test} — against an adjusted-out part that collapses $5.6\times$. The three columns sum to the encoder floor by construction; the identity does not by itself identify three causes.

n_{test}	floor	adjusted-out	matching gain	random-assign.	survivor
240	0.1156	0.0808	0.0210	0.0138	0.0348
450	0.0729	0.0401	0.0179	0.0149	0.0328
600	0.0619	0.0295	0.0180	0.0143	0.0324
1200	0.0445	0.0145	0.0161	0.0140	0.0300

The metrics in this paper score continuous codes through a quantile binning, and every number we report uses 10 bins per dimension (`cdnam.metrics`). Binning is part of the estimator, so by this paper's own argument the floor is a function of it, and we owe the reader the size of that dependence rather than a default.

Table 3: The floor and its decomposition on the headline cell (K=4, p=16, n_{test}=240, 5 seeds × 40 draws) as a function of the quantile-bin count. The floor and adjusted-out ("plug-in") columns are the *encoder* null's; the "code residue" column is the random-*code* null's AMI residue and "complement" the rest of the encoder floor, because this sweep did not record the encoder-null split of Table 2 (at 10 bins that split is 0.0210/0.0138, not 0.0114/0.0234). The plug-in and survivor totals, and the shares, are unaffected. Source: `binning_sensitivity.json`.

bins	floor	adjusted-out	code residue	complement	adj.-out share	compl. share
3	0.0498	0.0073	0.0044	0.0381	15%	76%
5	0.0586	0.0205	0.0065	0.0316	35%	54%
8	0.0866	0.0512	0.0101	0.0252	59%	29%
10 **(shipped)**	0.1156	0.0808	0.0114	0.0234	70%	20%
16	0.2232	0.1959	0.0139	0.0134	88%	6%
20	0.2955	0.2714	0.0134	0.0107	92%	4%

Two readings, and they point in opposite directions for the paper. **Against us:** the structural claim of §3 — that most of the floor is the adjusted-out term — is a property of the shipped configuration. It holds at 8 bins and above and fails below: at 3 bins the plug-in term is 15% of the floor and what survives the adjustment 85% of it. Anyone quoting our $0.081/0.021/0.014$ split as a general fact about these metrics would be quoting a fact about ten bins. **For us:** the thesis itself is unaffected, and is sharpened. The floor is far from zero at every bin count (0.050 at the coarsest setting we tried, an order of magnitude above the 0.001–0.016 random-code floors this literature would implicitly assume), and the part that survives adjustment on the encoder null is present throughout. A chance level is a property of a configuration; the discretisation is part of the configuration; that is the paper's argument, applied to the paper.

What survives the adjustment is alignment, not matching optimism. The matching gain of Table 2 could be in-sample selection — the Hungarian step choosing, on the very sample it scores, the pairing that happens to look best — or alignment that exists in the population. We separated the two by cross-fitting (`exp_matching_crossfit.py`, `matching_crossfit.json`): the

held-out set is split in halves; bin edges and the Hungarian assignment are fit on one half and applied, unchanged, to the other; the roles are swapped and averaged. With the two halves equal the procedure reproduces `compute_nmi` exactly. On fresh samples of the headline generator, in-sample optimism (in-sample minus cross-fit) is 0.006–0.008 NMI for the untrained encoders at n_{eval}=480 — 5–7% of their total floor but 22–33% of its adjusted part, an upper bound since part of it is mapping mismatch between halves — ≤ 0.001 from 2400 (1.8–4.2% of the adjusted part) and 0.0001 at 9600; it falls roughly as $n^{-1.4}$. What shrinks the untrained-encoder floor with n is therefore the plug-in term, which cross-fitting leaves in place and the adjustment removes; after both corrections the untrained floor is 0.029 (linear) and 0.022 (untrained DetNAM, ConceptNAM, CBM) at every n from 2400 to 9600 — information the untrained encoders carry about the factors, not an artefact of matching, once n is in the thousands. Two controls behave as they must: a random code falls to 0.0000 after both corrections, and oracle codes with known alignment keep it (0.098/0.298/0.599 cross-fit against 0.098/0.298/0.598 in-sample at 9600). Cross-fitting changes no verdict or ordering: trained minus the seed-paired untrained mean clears for DetNAM, ConceptNAM and CBM under every variant and n (Holm, df $= 4$), in the order CBM $>$ DetNAM $>$ ConceptNAM. By the stopping rule we fixed before running it, cross-fitting is therefore not proposed as a new evaluation procedure — at the sample sizes that matter it agrees with the existing one — but it settles what the floor that barely falls *is*. Two caveats: with a small calibration half (120 of 2400) the fixed mapping loses about 0.006 to mismatch, and for strongly aligned codes at small n the cross-fit score can slightly exceed the in-sample one (bin edges from the other half are not equal-mass on this one, which shrinks the normaliser); both vanish by n=2400.

The same split where the population is known. Cross-fitting separates matching optimism empirically; a Gaussian reference separates it exactly (`analyze_gaussian_reference.py`, `gaussian_reference.json`). With $X \sim \mathcal{N}(0, I_p)$, orthonormal linear concepts $c_k{=}a_k^\top X$ and an untrained $z{=}W^\top X$, each pair (z_i, c_j) is bivariate Gaussian for fixed W, so both its continuous MI and its binned MI at the paper's ten-bin discretisation are computable exactly. (For a single pair with W uniform on the sphere, $\mathbb{E}_W I = \frac{1}{2}[\psi(\frac{p}{2}) - \psi(\frac{p-1}{2})] \approx \frac{1}{2(p-1)}$, a standard consequence of spherical symmetry that we verify by quadrature and 2×10^6 Monte-Carlo draws; it is population information, not estimator bias.) Why a linear-concept Gaussian setting can stand in for our tanh generator: equal-frequency binning depends only on ranks, so a monotone link such as tanh is irrelevant; a Gaussian W is rotation-invariant, so only the Gram matrix of the concept directions matters; and our directions are not orthonormal ($|\cos|$ up to 0.53), but correlating the concepts does not move the linear floor (`null_preprocessing.json`). This argument covers the linear null only. The observed matched score on a sample then splits, as an exact accounting identity (the split between bias and optimism depends on the order the terms are evaluated in), into population alignment (the binned population optimum), estimation bias (the sample matrix at the population-optimal pairing minus that optimum) and selection optimism (sample-optimal minus population-optimal pairing on the sample). This setting reproduces the headline cell's floor closely — NMI 0.116 and adjusted 0.035 at n_{test}=240 (K=4, p=16), against 0.1156 and 0.0348 measured on our generator — and in it the adjusted floor is 90% population alignment and 12% selection optimism at 240, 99% and 1% at 1200, 99.8% and 0.2% at 4800. The matching-gain column of Table 2 (0.021 at 240) is 82% population matching gain and 21% optimism, and 98% population gain by 1200. At K=8, p=32 optimism is larger at small n (28% of the adjusted floor at 240) and again negligible by 1200 (3%). The unadjusted NMI at 240, by contrast, is 71% estimation bias — the part the adjustment removes. So from about a thousand samples the component that barely falls is, to within a few percent, information an untrained projection carries about the concepts; the matching step inflates it only at the smallest n and largest K.

The floor is also conditional on input preprocessing. The same argument applies to how the inputs are represented. `analyze_null_preprocessing.py` (`null_preprocessing.json`) recomputes the linear and untrained DetNAM nulls under preprocessing fit on the training split only: raw, per-feature standardisation, PCA whitening, a random non-orthogonal mixing (condition number 10 and 100), log-uniform per-feature rescaling, and an orthogonal rotation. The rotation is a sanity check — an isotropic linear null is invariant to it — and it holds on both cells (differences within two Monte-Carlo SEs). On the headline cell the other choices move the floor by at most 0.010 (linear) and 0.007 (DetNAM): about one null standard deviation and half to two-thirds of the smallest excess we report. On MultiFactor, PCA whitening *halves* the floor (0.127 $\rightarrow$ 0.069 linear, 0.113 $\rightarrow$ 0.062

DetNAM), about four null standard deviations (the full range across choices is about five). Inducing correlation between concepts moves neither. So a floor must be measured under the same preprocessing as the scores it is subtracted from — which every number in this paper is — and a floor reported without its preprocessing is not transferable, least of all to a cell with strongly correlated inputs.

And without binning at all: the survivor could still have been an artefact of quantile binning as such — a binning-free estimator exists and is already used in this literature: InfoMEC replaces binned MI with the Kraskov (KSG) kNN estimator for exactly this fragility (Hsu et al., 2023). So we recomputed both floors under KSG (`experiments/analyze_ksg_control.py` → `results/ksg_control.json`; Hungarian-matched mean MI in nats, so magnitudes are not comparable to the binned columns — only the trends are). Over the same $800 \to 4000$ rise in n the random-code floor falls $2.0\times$–$2.3\times$ across $k \in \{3, 5, 10\}$ neighbours, while the encoder-minus-code residual shrinks at no k (at k=5: 0.047 at n=800, 0.069 at 4000). Hsu et al. (2023) report their own KSG-based metrics to be very robust to k (their Fig. 6); we report all three k rather than assume that transfers, and the two statements hold at each. The component this paper is about is not a property of the binning.

We do not claim 10 bins is optimal. We claim it must be reported, and that an excess-over-floor number computed at one binning is not comparable with one computed at another.

B Full experimental tables

B.1 Statistical protocol, in full

- **Seeds**: 5 replicates throughout, so that every table is comparable to every other. Per-method seeds elsewhere come from the bijective offset table in `cdnam/utils/seed.py`; the headline driver instead gives every method the *same* raw replicate seed, which is common random numbers — the right choice for the paired tests this table feeds, and a deviation from the offset table that we note here rather than leave to be discovered. The headline cell is additionally replicated at 10 seeds (App. C) and nothing moves by more than 0.009 NMI. Sweeps use 3. All $\pm$ are sample standard deviations (ddof=1), the convention `cdnam.stats` uses for every interval quoted beside them; `scripts/check_paper_pm.py` verifies that each mean$\pm$std cell in this manuscript matches its source JSON.
- **Difference testing**: paired t-test with BH correction; Cohen's d reported alongside p.
- **Minimum detectable effect, not observed power**. Observed power is computed from the effect actually seen, so it is a monotone transform of the p-value and cannot separate a small effect from a weak design. We report the MDE, which depends only on the design: at n=5, α=0.05, 80% target power a paired test resolves d=1.68, which at the median paired-difference spread across method pairs (0.0174) is 0.029 NMI; at n=10, d=1.00 (`exp6_statistical_analysis.json`, `minimum_detectable_effect`, which records the standard deviation used). Every verdict in this family is multiplicity-corrected, and the MDE at the Bonferroni α=0.01 actually applied is d=2.63, or 0.046 NMI (`tost_corrected.json`) — essentially the size of the largest within-Level-1 difference in the headline cell, so verdicts on differences near that scale sit at the design's resolution, not comfortably inside it. *Not every difference we call significant clears that figure:* ConceptNAM vs. Level-0 β-VAE is significant after BH correction at d=1.61, just under it. We report the MDE so a reader can see which verdicts sit near the design's resolution, not to claim none of them do.
- **Multiplicity applies to floor-clearing too.** The headline cell asks six simultaneous questions — does each method exceed its floor — so the six share a family-wise error rate the way the five TOSTs below do. The verdict C4 quotes is computed by `analyze_matched_verdict.py` (`matched_verdict.json`): a one-sided Welch t of each method's 5-seed mean against its own architecture-matched null (`arch_matched_null.json`), carrying the method's seed variance and the null's draw variance. Under matched nulls, six of six clear under Holm (the declared correction), five at plain Bonferroni ($\alpha/6$; β-VAE+sup fails, p_{adj}=0.13), all six under BH, and five under Benjamini–Yekutieli, which BH's independence assumption makes the relevant FDR check here. Holm is the declared correction everywhere, chosen for consistency with the resolution

grid; it happens to raise this count, so we print Bonferroni beside it. The delta-method interval for the $12.5\times$ ratio is 5–$34\times$. The same test against the canonical *linear* null clears four under Holm, three at Bonferroni and four under BH; on this cell every matched null sits below the linear one, so the linear view is the conservative robustness row, not the verdict. *What "clears" means here, once.* A verdict — any sentence saying a method does or does not beat its floor — is the one-sided Welch t above, against that method's own matched null mean, with both variances carried and the correction taken over the family being asked. A greyed table cell (like this) marks a mean below the 95th percentile of that cell's 5-seed-mean null: a "do not read this number" convention, deliberately stricter than the test, and not itself a verdict. A bare null printed in a table is that cell's null *mean*, so the excesses beside it are differences of means rather than test statistics.

What the design resolves, and how fragile the count is. On the paired family's convention (one-sided, n=5, 80% power) this family's MDE is d=1.36 at α and d=2.29 at the $\alpha/6$ applied. Two of the five Bonferroni-clearing methods sit below it (NBM d=1.80, FactorVAE+sup d=2.02), so those verdicts sit at the design's resolution, not inside it. The count is seed-fragile: dropping any single seed leaves 2–4 of six clearing, never five, NBM's margin (p=0.0079 vs $\alpha/6$=0.0083) binding. And at n=5 the smallest attainable one-sided sign/permutation p is 2^{-5}=0.031 $> \alpha/6$: the count exists only under normal theory. We report it with these caveats, not as a headline. Under the declared Holm correction the same leave-one-seed-out exercise gives 2–6 of six, so the Holm count of six is no less seed-fragile than the Bonferroni count of five.

The inference unit, and the target. The matched null's 100 draws are not 100 independent observations: they are 5 clusters of 20, each sharing its seed's evaluation data with that seed's trained score. The Welch statistic above treats them as independent. We therefore also run the test with the seed as the unit: per seed s, $d_{m,s} = S_{m,s} - \bar{\mu}_{0,m,s}$ (the method's score minus the mean of its own 20 null draws on that seed's data), and a one-sided one-sample t on the five $d_{m,s}$, df=4 (`analyze_verdict_unification.py`, `verdict_unification.json`). The target is then explicit: $\mathbb{E}_s[S_{\text{trained}} - S_{\text{untrained}}]$ over sample seeds, conditional on the generator's fixed W. The counts are six under Holm, five at Bonferroni, six under BH and six under BY; the pairing moves individual p-values (β-VAE+sup: 0.0224 unpaired, 0.0131 paired) but no Holm, Bonferroni or BH verdict.

One rule, applied everywhere it is claimed. The resolution table (Table 21) and the dSprites tabular probe (Table 31) print a p_{95} screen: the 5-seed mean against the 95th percentile of the null. We re-ran the declared Welch rule on the same artifacts. On the resolution grid (50 tests: 5 cells $\times$ 5 sample sizes $\times$ DetNAM/CBM, Holm across all 50) the two agree on 49; the one disagreement is ScalingSynth K=8 DetNAM at n=2000, which the screen passes and Holm does not (p=0.020) — above its earlier grid point, so no first-resolving n moves under Holm, and under the stricter Bonferroni only first-resolving n for DetNAM move (Synthetic-Concept to 4000, K=8 and K=32 to 1000, MultiFactor to beyond the grid). On the dSprites tabular probe all four Level-1 rows clear their own matched nulls under Welch under Holm, and under Bonferroni too (largest p=0.0056). Across every Level-1 entry of Table 17 (20 tests, Holm) the Welch rule and the greying screen agree on 19; the exception is again ScalingSynth K=8 DetNAM, which passes the screen but not Welch with Holm (p=0.017), so we do not claim it clears. The MultiFactor Level-1 entries all fail: the highest, DetNAM, sits 0.0006 above the null mean, Welch p=0.49.

The $12.5\times$ *ratio is a post-hoc extreme.* Its numerator and denominator are the largest and smallest of six observed excesses, chosen after seeing them, and the Fieller interval above treats that pair as fixed and ignores the covariance the shared seeds induce. A seed bootstrap that resamples the five seeds jointly across methods and re-selects the maximum and minimum inside every resample gives $[10.2, 24.3]$; at n=5 a bootstrap has at most 126 distinct resamples and under-covers, and re-selecting the extremes inside each resample inflates the ratio by construction, so this interval is *not* a usable CI and we draw nothing from it beyond reporting it. We treat the ratio as exploratory. The confirmatory statement is the set of six absolute excesses with their tests; the ratio only summarises how unequal they are.

Each run against its own initial state. The cleanest reference for a learning gain is not a null ensemble but the network the run started from. `exp_paired_learning_gain.py` (`paired_learning_gain.json`) re-runs the corrected headline protocol, scores each

model once before training and once after, and tests d=final − initial per run (one-sided t, Holm over the six Level-1 methods; `cdnam.floor.paired_gain`). The initial scores are distributed as the architecture-matched null draws (no Kolmogorov–Smirnov test rejects after Holm), so this reference is that null, paired. At the five headline seeds five of six methods clear: FactorVAE+sup's gain is 0.011, 95% CI $[-0.007, 0.028]$ (p=0.082), because its initial scores run high on these seeds. At ten seeds all six clear (smallest gain: β-VAE+sup, 0.011 $[0.004, 0.018]$), NBM and DetNAM stay on top (0.138, 0.116), and the Level-0 β-VAE shows no gain against its own start (0.005 $[-0.007, 0.017]$). Rank agreement with the matched-null excesses is Spearman 0.89 at five seeds and 0.94 at ten. So the self-paired reference *changes one verdict at the headline seed count* (FactorVAE+sup, five of six clear) and agrees at ten; it does not sharpen the test, since its variance is close to the null-paired one.

Error rates of the verdict rule, measured. To check the rule rather than assert it, `analyze_verdict_calibration.py` (`verdict_calibration.json`) runs a negative control on the headline cell's own data: for each seed one untrained draw of the same architecture plays the "trained" score, so the true gain is zero, and 20 further draws are its null (6,000 replicates; random linear, untrained DetNAM and untrained ConceptNAM). The unpaired Welch rule (identical to `Floor.test`) rejects at 2.9–3.2% for nominal 5% and 0.35–0.50% for $\alpha/6$=0.83%; the seed-paired t at 2.9–3.2% and 0.28–0.37%; the family-wise error over six null methods is 1.6–2.6% under Holm or Bonferroni. Both rules are valid, somewhat conservatively. The p_{95} single-score screen is not: built from 20 five-seed-mean draws in this simulation, it passes a null method 8.7–9.9% of the time, about twice its nominal rate, which is why no verdict in this paper rests on it alone (the resolution grid's screen uses 40 draws, which narrows but does not remove the gap). Interval coverage for the gain is 93–94% at nominal 95%, a small shortfall we trace to the null draws' positive skew. Power at Bonferroni $\alpha/6$ is 32–54% (Welch; 29–48% seed-paired) for a gain of 0.015 NMI — the size of the smallest excess we report — and 93–99% at 0.03: the smallest verdicts in this paper sit where a five-seed design detects an effect about half the time.

- **Equivalence testing, and why we do not calibrate the margin against noise.** We use TOST (Schuirmann, 1987). One could choose δ by running the same test on two *independent untrained* encoders and requiring a low "equivalent" rate. That is a category error. Two untrained encoders have a true difference of exactly zero, so TOST's alternative is true and declaring equivalence is the *correct* decision; the rate is the test's power at zero difference, not a false-positive rate. Worse, it is monotone in the seed budget and tends to 1 for any $\delta > 0$: at δ=0.0145 we measure 34% at 5 seeds, 84% at 10, 100% at 40 (`analyze_chance_floor.py`, `margin_analysis`). A rule that yields a tighter required margin the more data you collect is not a rule.

  What does license or refuse a margin is its size relative to the effects it adjudicates. This benchmark supplies that scale directly: the distance from the measured floor (0.115) to the Level-3 ceiling is 0.502 NMI on the corrected block; the largest excess over the floor among the three methods this margin calls equivalent is 0.023 there and 0.030 on the shipped block the TOSTs are computed on; and the range *between* those three methods is 0.015 and 0.002. At δ=0.05 the margin is $2.2\times$ the largest excess on the corrected block and $1.7\times$ on the shipped one (`tost_corrected.json`), and $3.3\times$ and $25\times$ the range (`verdict_unification.json`). So a $\pm\delta$ band around any one of them reaches below the floor itself: a positive verdict cannot separate those methods — not because it passes noise, but because it is wider than everything it is being asked to distinguish. The first quantity is the height of the best method above the floor, not the spread among the methods; the spread is the range, which is smaller, so every ratio against it is larger. We print both. Neither is a pre-registered minimum important effect; they bound what a δ can mean here, they do not choose one. We therefore report the margin at which each equivalence *breaks*, together with the excesses it should be read against. A positive TOST result means *proven equivalent within* $\pm\delta$, never merely "not significantly different".

- **Multiplicity applies to equivalence too**. Equivalence tests are no more exempt than difference tests: the headline table runs five simultaneous TOSTs against one reference, so we report Bonferroni-adjusted TOST p-values alongside the raw ones (`cdnam.stats.tost_family`).

B.2 HEADLINE TABLE: FULL CAPTION NOTES

Headline comparison on the synthetic concept benchmark (K=4, n=800, 5 seeds, held-out; App. E repro at n=2000). Values are mean ± std. The null row's MIG and DCI-D entries are 0.021 and 0.050; NMI and MIG use 2000 draws, DCI-D 200 (`null_calibration.json`). On what a task label buys the VAE family specifically: β-VAE+sup's matched excess includes what training alone earns — Level-0 β-VAE sits 0.004 above its matched null, not resolved from it (p=0.087) — so the task-label *increment* of 0.011 cannot be separated from package (`matched_verdict.json`), and C4's range should be read as spanning method-and-label packages, not label value alone. "TOST p" is the Schuirmann two-one-sided-tests p-value for equivalence to ConceptNAM at δ=0.05 NMI; $p<0.05$ (bold) means *statistically equivalent* at a margin we show in §4.4 is loose, not just "not significantly different". "task R^2" is the held-out coefficient of determination of each method's own task head (residuals against the test-set mean; the train-mean predictor gives RMSE 0.817 for comparison) — an *effective-supervision* diagnostic: it verifies that a method allowed to consume y actually did. Paired *t*/Cohen's d per pair: `exp6_statistical_analysis.json` — note ConceptNAM vs. Level-0 β-VAE is *neither* established as equivalent nor as different once the family correction this paper declares is applied: the TOST is p_{adj}=0.102 (Table 1) and the difference is p=0.023 raw at d=1.61, below the paired family's own MDE of d=1.68. The *null* row is an untrained random linear encoder on all three concept metrics, same split (`analyze_null_calibration.py`; `analyze_chance_floor.py` measures the same null on a different grid and is not the artifact the row is read from). **That artifact ships three spreads for one null, on two scales, so the row has to say which it prints.** The ± shown is the *single-draw* spread: 0.011 over 2000 draws for NMI, 0.011 for MIG, and 0.027 over 200 draws for DCI-D, whose spread is more than twice the other two. It spans draws, not seeds, and sits on the same scale as the method rows' seed-to-seed ±. The spread of the 5-seed *mean* is smaller — 0.0049 over 400 such means — and that is the scale a 5-seed method mean is actually judged on: p_{95} is 0.136 against a single draw but 0.1239 against the mean. We print the single-draw figure because it matches the column it sits in, and use the mean-scale figure wherever a clearing verdict is taken (§4.4); neither is a third estimate of the null itself, whose mean is 0.115 on both. Read every column as excess over it. In the upper (shipped) block: Level-0 +0.001, the VAE triplet +0.028–+0.030 (upper block; +0.008–+0.023 on the corrected block below, which is the basis for the 2.2× figure — the upper block's own ratio is 1.7×), DetNAM +0.077 (+0.130 on the corrected block), CBM +0.519. **The lower block reruns the same cell with each method's un-swept flags selected on *validation* task RMSE**, which uses no concept labels and is disjoint from the test split every number is read from (`exp_bench_corrected.py`; TOST and power omitted there). The 55/15/30 split leaves less training data, which pushes CBM and the VAE family below the upper block — but not everything: DetNAM *rises*, 0.192 → 0.246, because the configuration it is selected at more than compensates for the smaller training set. It reorders the table: NBM and DetNAM reach 0.295 and 0.246, excesses of 0.180 and 0.130 against CBM's 0.502 — 36% and 26% of the Level-3 gain, against 15% for DetNAM above (NBM is absent from the upper block; at the shipped configuration it reaches 27% here, Table 15) — while ConceptNAM and the Level-1 VAE family stay at 0.123–0.139. Cite the lower block; we keep the upper one because every other experiment here used it. *On MIG the VAE triplet and Level-0 sit at or below chance*; on DCI-D two of them are significantly above it. The two metrics disagree, and only MIG says something stronger than NMI does (App. B). *Multiplicity.* The table runs five simultaneous equivalence tests against ConceptNAM, so its last column is the Bonferroni-adjusted TOST p (`exp6_statistical_analysis.json`). The two within-VAE-family equivalences survive comfortably (0.0011, 0.0033); the Level-0-vs-Level-1 comparison does not (0.021 → 0.102), so at this seed count we can certify neither that a method given task labels differs from one given none nor that it does not.

B.3 IS THE EQUIVALENCE A FLOOR EFFECT?

An equivalence observed where every method fails is not a finding about architecture. Since all non-CBM methods sit at NMI 0.11–0.19 in Table 1, "architecture does not matter" and "nothing works here" predict the same table. We separate them by moving difficulty along two axes that leave the supervision budget untouched — the concept count K (our capacity argument puts the single-task ceiling at $O(1/K)$) and the label noise σ — holding p=16, the hyperparameters, and the 5 seeds fixed (`exp_floor_control.py`; the K=4, σ=0.1 cell reproduces Table 1 exactly).

Result: the equivalence is regime-dependent, the capacity gap is not. Two things separate. (i) *The within-family equivalence does not survive an easier problem.* Holding $\sigma{=}0.1$ and moving $K{=}4 \to 2$, the triplet's pairwise mean differences grow from 0.0008–0.0020 to 0.030–0.074 and TOST no longer establishes equivalence for any pair ($0/3$ at both noise levels). Neither direction is strong, though: at $K{=}4$ the triplet passes TOST at $\delta{=}0.05$, a margin $1.7\times$ the triplet's largest excess over chance in that cell (§4.4), so read it as an upper bound rather than evidence of sameness; at $K{=}2$ only one pair shows positive evidence of a *difference* (ConceptNAM vs. β-VAE+sup, paired t $p{=}0.003$), the other two being indeterminate at $n{=}5$ ($p{=}0.073$, $p{=}0.097$; all three from the $\sigma{=}0.1$ cell). The control shows the equivalence *evaporating*, not the triplet separating, which scopes the claim to the $K{=}4$ regime we measure. (ii) *"Level-1" is not a single quantity.* The easier cell lifts CBM by $+0.202$ on raw NMI, and the null moves only $0.114 \to 0.105$ between those cells (both from `chance_floor.json`, which measures every benchmark cell on one grid), so almost all of that lift is real — which is exactly why raw NMI cannot be compared across cells whose floors differ; what remains is a dispersion statement. As excess over the per-cell null, and counting only entries whose excess clears that cell's floor, the fraction of CBM's gain a Level-1 method captures runs 4.4% to 34.1% (NBM, $K{=}4$) — a factor of 7.7; β-VAE+sup at $K{=}2$ is excluded because its excess does not clear that cell's threshold. The *sign* of the response to difficulty differs by method: NODE-GAM's excess rises $0.032 \to 0.108$ and ConceptNAM's $0.029 \to 0.074$ while β-VAE+sup's falls $0.028 \to 0.012$. No aggregate is robust. Comparing the hardest cell ($K{=}4, \sigma{=}0.1$) with the easiest ($K{=}2, \sigma{=}0.01$), the per-cell maximum falls ($\times 0.96$), the mean rises ($\times 1.46$) and the median rises faster ($\times 2.97$), against CBM's $\times 1.41$ — three aggregates of the same numbers pointing three ways. *That is a two-corner comparison and we label it as one:* across all four cells the per-cell maximum ranges 0.124–0.181 ($\times 1.46$), and moving σ alone at $K{=}4$ raises all three aggregates together. The disagreement is a K effect, not a general property of aggregation.

We make no quantitative claim about what a task label buys relative to concept labels, and recommend none be quoted from this benchmark until it is measured across n, per method, and at a selected rather than a default configuration: every such number we computed moved when the same grid was re-aggregated (App. Z), and the configuration audit (App. G) moved it again. Table 15 gives the per-cell values so any aggregate can be checked.

B.4 Floor-effect control: noise sensitivity and comparability

Label noise is not the binding constraint for most methods, but it is not negligible for all: a $10\times$ noise reduction moves five of the six Level-1 methods by at most 0.012 NMI (largest: ConceptNAM -0.0118 at $K{=}2$), while NBM moves $+0.040$ at $K{=}4$. Finally, the VAE triplet still has task $R^2 \approx 0$ at $K{=}2$ (-0.003 to -0.008), so what separates the methods there is inductive bias, not supervision they suddenly began to consume.

On comparability. NMI's normalisation depends on K and on n_{test}, so raw values are not comparable across cells. The null column quantifies exactly that: it moves $0.114 \to 0.105$ (`chance_floor.json`, both endpoints) between the K values here, and drops to 0.060 at the $n{=}2000$ protocol of App. E — which is also why per-method NMI is higher at $n{=}800$ than at $n{=}2000$. Excess-over-null removes the part of that dependence that is chance, and it is the scale on which we state (ii); it does *not* make cells commensurable. Subtracting a floor corrects location, not units: the attainable ceiling, the concepts' entropy and cardinality, the estimator's sensitivity and the K_{model} normaliser (App. R) still differ between cells, so an excess of 0.05 at $K{=}4$ and at $K{=}32$ are not the same amount of recovery. Excess-over-null is comparable across methods *within* a cell, which is how (ii) uses it.

B.5 Additional benchmark results (summary)

Seven further studies are reported in full in the tables below and in Apps. E and O; all run 5 seeds under the corrected fixed-W protocol: (i) in the low-label regime CBM dominates at every rate (NMI $0.170 \to 0.465$ from 0.5% to 5% labels) and ConceptNAM is below it everywhere; (ii) with noise applied to X rather than y (Audit B3) all methods are flat across $\sigma \in [0.05, 1.0]$; (iii) under covariate-only shift CBM retains 96–100% of in-distribution NMI while Level-1 methods collapse to 0.09–0.10; (iv) ablations put the *task-only* variant highest (NMI 0.164 ± 0.013 vs. full ConceptNAM 0.143 ± 0.005), consistent with §5.4. The remaining three are concept transfer across generators

(~ 0.96–1.07 cross-dataset ratio), a California-Housing sparsity sweep ($R^2 \approx 0.62$ stable while peak attribution rises $0.15 \rightarrow 0.27$), and four real tabular sets scored against first-K-feature concept *proxies* — our own construction, not that of Koh et al. (2020), whose CBMs use human-annotated concepts — an interpretability sanity check, not a concept-recovery result, since these datasets have no ground-truth concept annotations. Auxiliary drivers shipped in the code release (active-learning, CIFAR-10, Fashion-MNIST) are not part of the benchmark.

Table 4: Attribution Consistency Score (**10 seeds**; IoU at top-3). Sensitivity to top-k in App. F. **Read both columns as excess over the null row, not over zero.** ACS aligns rows by Hungarian matching before scoring, and Hungarian matching over K rows of length p finds a good alignment even between two *independent* random attributions. Ten random row-normalised (K, p) matrices at K=4, p=16 score $\mathrm{ACS}_{\cos}$=0.728 and $\mathrm{ACS}_{\mathrm{IoU}}$=0.192 (200 draws; `analyze_chance_floor.py`). The cosine column therefore spans 0.73–1.00, not 0–1: DetNAM's 0.811 is 0.083 above chance against ConceptNAM's 0.193 and CBM's 0.268 — so on the scale that matters DetNAM recovers 43% of ConceptNAM's excess and 31% of CBM's, a much smaller spread than the raw 0.811 / 0.921 / 0.996 suggest. The IoU column is the more informative of the two here because its chance level is far from its range. The $\pm$ are over the 45 seed *pairs*, which are not independent (each seed appears in 9 pairs), so they are descriptive spreads, not standard errors.

Method	Mean Cosine	Mean IoU (top-3)
CBM joint (Level 3)	$\mathbf{0.996 \pm 0.002}$	$\mathbf{1.000 \pm 0.000}$
β-VAE+sup (Level 1)	0.950 ± 0.020	0.526 ± 0.123
ConceptNAM (Level 1)	0.921 ± 0.029	0.703 ± 0.214
DetNAM (Level 1, NAM-family)	0.811 ± 0.056	0.450 ± 0.107
random attributions (null)	0.728	0.192

iCBM: FULL OPERATING-POINT DISCUSSION

iCBM (positive contribution). DetNAM beats ConceptNAM at every label ratio (S1) but cannot consume concept labels; CBM has the opposite limitation. We propose **iCBM** (`baselines/icbm.py`): the DetNAM encoder + deterministic linear K-bottleneck + NAM-additive head + Hungarian-aligned concept supervision on the labelled subset. Evaluated under the same fixed-W semi-sup protocol as S1, iCBM occupies a unique operating point (Table 6, App. B): at r=0 it *is* DetNAM (no labels, hence no alignment term; the 0.004 gap between the two columns is seed noise, not a measured equivalence) and both beat CBM (0.15); for $r > 0$ CBM has the supervised-ceiling lead on this synthetic but iCBM accepts concept labels without architectural switch (5/5 seeds stable at every ratio). The cross-domain evidence is now produced under the leakage-controlled canonical driver (`exp_dsprites_canonical.py`, App. M.1): on raw 64×64 dSprites with concept supervision, **iCBM matches CBM** on NMI (0.189 vs 0.194) and on DCI-D (0.508 vs 0.501); on real clinical data (UCI Heart, Pima; App. N, 5 seeds) iCBM matches or exceeds CBM in mean AUC (0.907 ± 0.02 vs 0.886 ± 0.05 on Heart; 0.792 ± 0.01 vs 0.809 ± 0.01 on Pima). Its seed-to-seed std is smaller than *ConceptNAM*'s by 5.6× on Heart and 12.6× on Pima; against CBM — the comparator in the AUC figures just quoted — it is 2.0× smaller on Heart and 2.4× *larger* on Pima, so the variance claim is about ConceptNAM only (App. N: a direction only — the conservative independent bootstrap CI includes 1 on both datasets, and the median-centred Brown–Forsythe test does not reject on either).

Tables 7–9 report the complete results referenced in §5.4. All $\pm$ are standard deviations across the protocol's 5 seeds.

Reading. Removing the task loss collapses NMI ($0.143 \rightarrow 0.107$); the partial-removal probes (`no_sparsity`, `no_IB`, `no_TC`) each move NMI by ≤ 0.005; the `task_only` variant (all four VIB toggles off) is the strongest at NMI 0.164, +15% relative to `full`. The clean comparison is `full vs task_only`: the full VIB stack is a net cost on this benchmark. We do *not* claim per-component attribution from `no_IB`/`no_TC` alone — the remaining KL/TC terms in each partial-removal row mask which component carries the cost (caption `Toggle semantics`). The conclusion is consistent with the §5.4 S1 result that DetNAM (no IB) recovers concepts more faithfully than ConceptNAM across every label ratio.

Table 5: Semi-supervised NMI for all methods (n_{samples}=800, 30 epochs, **5 seeds**, held-out; mean ± std across seeds; data `results/exp10_alignment_all_methods.json`, driver `exp10_alignment_all_methods.py`, as listed in Table 39); this table runs at the same n=800 as Table 1. **Null for this cell (K=4, n=800): NMI** 0.115 (`null_calibration.json`). The 0% row runs 0.139–0.161, i.e. +0.024 to +0.046 over chance, so read that column as a small excess rather than as a level.

Labels	ConceptNAM	β-VAE+sup	FactorVAE+sup	CBM
0%	0.146 ± 0.005	0.139 ± 0.009	0.149 ± 0.016	$\mathbf{0.161 \pm 0.016}$
10%	0.220 ± 0.014	0.146 ± 0.017	0.218 ± 0.011	$\mathbf{0.487 \pm 0.036}$
50%	0.312 ± 0.014	0.339 ± 0.015	$\underline{0.380 \pm 0.021}$	$\mathbf{0.583 \pm 0.014}$

Table 6: iCBM semi-supervised sweep: NMI mean ± std on synthetic (K=4, fixed W, 5 seeds; `results/exp_icbm_semisup.json`). iCBM matches or exceeds DetNAM at r=0 and accepts concept supervision smoothly without the catastrophic instability of ConceptNAM; CBM is the supervised-ceiling reference. **Null for this cell (K=4, n=1500): NMI** 0.072 (`sblock_nulls.json`; the `chance_floor.json` run of the same cell gives 0.074, within draw noise). ConceptNAM at r=0.10 (0.082) is within one draw-sd of it and at r=0.25 (0.098) within 2.3; its "instability" across r is movement at the scale of the null's own spread.

r	iCBM (proposed)	CBM (ref. ceiling)	DetNAM (task labels only)	ConceptNAM
0.00	0.202±0.033	0.148±0.006	**0.206±0.041**	0.111±0.016
0.10	0.433±0.007	**0.585±0.051**	0.206±0.041	0.082±0.005
0.25	0.453±0.009	**0.662±0.020**	0.206±0.041	0.098±0.016
0.50	0.465±0.007	**0.687±0.017**	0.206±0.041	0.159±0.034
1.00	0.467±0.005	**0.696±0.022**	0.206±0.041	0.239±0.034

C TEN-SEED REPLICATION OF THE HEADLINE CELL

Five replicates is a weak instrument, and a paper that reports equivalence verdicts should say what that costs. At n=5 a paired test at α=0.05 and 80% power resolves d=1.68, about 0.029 NMI at the median paired-difference spread; at n=10 it resolves d=1.00. We therefore reran the headline cell with a second block of five replicate seeds (2024, 314, 1618, 2718, 577), keeping the original five unchanged so the two runs share half their samples and the comparison is a genuine extension rather than a fresh draw (`exp6_full_comparison.py -seeds 10`, `exp6_full_comparison_n10.json`).

The stability is worth stating plainly because it cuts against us as well as for us: the n=5 numbers were not noise, so the conclusions this paper draws from them are not artifacts of a small replicate count — and neither is the fact that most of the Level-1 effects sit close to the measured floor. More seeds do not lift a method off the floor; they only make its position on it more precisely known.

D POWER SUPPLEMENT AND TABCBM HEAD-TO-HEAD

This appendix collects three supplementary analyses requested in the body's Reviewer Q&A (Q4, Q16) and used inline in App. N: (i) doubling the headline-protocol seed count from n=5 to n=10 on Synthetic-Concept; (ii) a head-to-head ConceptDiscoveryBench run of a TabCBM-style baseline (Espinosa Zarlenga et al., 2023), with deviations from the published TabCBM architecture documented below; and (iii) a non-parametric bootstrap CI on the seed-level AUC standard deviation used to assess the 5.6× clinical-variance point estimate. All raw artefacts are in `results/exp_sprint8_n10_supplement.json`, `results/exp_sprint8_n10_merged.json`, and `results/exp_clinical_bootstrap.json`.

Table 7: California Housing sparsity sweep (val-selected, test-reported once).

λ_{sparse}	Test R^2 (std)	Max Weight	Gap	Entropy
0.01	0.626 ± 0.005	0.152	0.049	2.076
0.05	0.621 ± 0.024	0.159	0.059	2.074
0.10	0.606 ± 0.043	0.166	0.072	2.072
0.50	0.614 ± 0.018	0.268	0.201	2.009
1.00	0.624 ± 0.008	0.257	0.194	2.001

Table 8: Architectural ablations (synthetic K=4, **5 seeds**, *fixed-W*; `exp12_ablations.py`, JSON path App. W). **Toggle semantics.** `no_IB` sets `enable_ib`=False, which gates *both* KL paths — the $\alpha \cdot I_{\text{IB}}$ term and the $\gamma \cdot \text{KL}$ branch — leaving only $\beta \cdot \text{TC}$ (`concept_nam.py`: `kl_active = enable_ib and enable_gamma`; audit item C1). `no_TC` removes only $\beta \cdot \text{TC}$; `no_sparsity` removes the row-entropy penalty $\lambda_{\text{sparse}} \sum_k H(A_{k,:})$ on the row-softmaxed A — not the ℓ_1 norm that Remark 2 uses. `task_only` turns all four off and is the cleanest ablation; the single-toggle rows are partial-removal probes whose remaining terms mask one another's marginal effect. **Null for this cell** (K=4, n=800)**: NMI** 0.115 (`null_calibration.json`). Two readings change: `no task loss` at 0.107 is *below* chance, not merely collapsed; and `task only` at 0.164 is +0.049 against `full`'s +0.027, i.e. +76% in excess terms rather than the +15% a raw ratio gives. The three partial-removal probes move by $\leq$0.005, inside the null's draw-to-draw spread. Data: `exp12_ablations.json`.

Variant	NMI	Max Attr.	ACS-cos	ACS-IoU
full	0.143 ± 0.005	0.085	0.988	0.165
no task loss	0.107 ± 0.003	0.111	0.833	0.133
no sparsity	0.143 ± 0.009	0.078	0.993	0.168
no IB	0.139 ± 0.011	0.081	0.990	0.168
no TC	0.143 ± 0.014	0.085	0.988	0.155
task only	$\mathbf{0.164 \pm 0.013}$	0.081	0.991	0.207

D.1 n=10 POWER EXTENSION ON THE BENCH PROTOCOL

We re-ran the headline ConceptDiscoveryBench methods on Synthetic-Concept (K=4, p=16, n_{samples}=2000, 40 epochs, lr=10^{-3}, batch 64) with five additional replicate seeds $\{17, 23, 29, 31, 37\}$, then merged with the original five $\{42, 123, 456, 7, 11\}$. Identical seeding and optimisation hyperparameters as `exp_bench.py`; no other change. Results:

For the within-family TOST equivalence claim of §5.1, re-evaluating Schuirmann's two-one-sided tests with the n=10 ConceptNAM std (0.011) and the same δ=0.05 equivalence band leaves the original verdict intact: within-VAE-family pairs (β-VAE+sup, FactorVAE+sup, ConceptNAM) remain TOST-equivalent at $p < 0.01$. The DetNAM vs ConceptNAM gap widens slightly ($0.259 - 0.095 = 0.164$ at n=10 vs 0.137 at n=5), but remains outside the equivalence band as the body table reports.

Note on iCBM in the raw supplement JSON. For completeness, `exp_sprint8_synthetic_supplement.py` also re-trains iCBM on the same five new seeds at `concept_loss_weight=0` (no concept labels are available under the Synthetic-Concept Level-1 protocol). In this zero-concept-label configuration iCBM's Hungarian-aligned partial concept supervision is inactive, so iCBM reduces architecturally to DetNAM modulo per-method seed offsets. We therefore do not list iCBM as a separate row in Table 11; the per-seed iCBM numbers in `results/exp_sprint8_n10_supplement.json` are presented as a sanity check that the iCBM forward path matches DetNAM under zero concept labels (observed NMI gap $|0.278 - 0.259| = 0.019$, well within DetNAM's own seed-to-seed std of 0.078), *not* as a new method comparison.

Table 9: Multi-Factor synthetic (5 factors, 32 features, n=5000, held-out NMI, 3 seeds). **Every method here scores below chance.** An untrained random linear encoder on this generator, under the same split and metric, reaches 0.110 ± 0.016 with a 5th percentile of 0.087 (`analyze_chance_floor.py`); the best entry below is 0.087, i.e. exactly at that percentile, and the worst shown is 0.063. (The driver sweeps 13 configurations and the table shows 6; over all 13 the worst is 0.022.) This table therefore does not rank methods: a 0.02 ordering among sub-chance scores is not a lead, and its row order is not evidence about which axis dominates. What the table does show is the paper's thesis in its sharpest form: on a generator with five independent factors and no concept labels, no Level-0 or Level-1 method extracts anything an untrained projection does not already provide.

Method	NMI (std)
FactorVAE γ=30	0.087 ± 0.011
FactorVAE γ=10	0.083 ± 0.010
FactorVAE γ=50	0.080 ± 0.010
β-VAE+sup β=10	0.077 ± 0.006
β-VAE β=1	0.075 ± 0.006
ConceptNAM	0.063 ± 0.005

Table 10: Headline cell at 5 and 10 replicate seeds (K=4, p=16, n=800, fixed W). Nothing moves by more than 0.009 NMI, no ordering changes, and the equivalence verdicts are identical: $2/5$ pairs survive Bonferroni-adjusted TOST at n=5 and two of five at n=10 — the same two pairs — with the Level-0 comparison failing at both (p_{adj}=0.102 and 0.111; Table 1, `verdict_unification.json`); the margin caveat of §4.4 applies to both. Doubling the seeds buys resolution, not a different answer. **Null for this cell (K=4, n=800): NMI** 0.115 (`null_calibration.json`). Data: `exp6_full_comparison.json` and `exp6_full_comparison_n10.json`.

Method	NMI (5 seeds)	NMI (10 seeds)	Δ
CBM joint	0.634 ± 0.011	0.636 ± 0.016	$+0.002$
DetNAM	0.192 ± 0.035	0.201 ± 0.032	$+0.009$
FactorVAE+sup	0.145 ± 0.006	0.150 ± 0.009	$+0.005$
ConceptNAM	0.144 ± 0.010	0.149 ± 0.012	$+0.005$
β-VAE+sup	0.143 ± 0.011	0.145 ± 0.012	$+0.002$
β-VAE	0.117 ± 0.011	0.113 ± 0.009	-0.004

D.2 TabCBM head-to-head on Synthetic-Concept

TabCBM (Espinosa Zarlenga et al., 2023) is the closest tabular concept discovery contemporary; the body benchmark-differences table (Table 16) listed its multi-metric/matched columns as "not run": an official implementation is public, but it had not been run under our protocol. Our first entry is a reduced re-implementation of the architecture, not the original method, and should be read accordingly (a port of the official implementation follows below). We provide a *TabCBM-style baseline* in `baselines/tabcbm.py`: per-concept soft feature gating (temperature-1 sigmoid), a per-concept MLP over the gated input, and two auxiliary penalties on the gates. TabCBM learns per-concept feature *masks* (logit probabilities) and per-concept generator MLPs — which our gate-plus-MLP construction resembles rather than replaces — and its published regularisers are coherence (λ_{co}), diversity (λ_{div}) and specificity (λ_{spect}), none of which we implement. **What our baseline actually is**: (a) the gate/mask and per-concept-generator signature is shared with TabCBM; (b) the two penalties are *ours*, not theirs — a mutual-exclusivity term (λ=0.1) pushing each feature toward one concept and a gate-sharpening term (λ=0.01); we implement none of TabCBM's three published regularisers, in particular not the k-nearest-neighbour coherence loss; (c) the supervised concept-predictor head is dropped because the Level-1 protocol provides no concept labels. The baseline therefore tests a *gating + per-concept-head* skeleton in the TabCBM family, not the published method, and says nothing about that method's performance — not even as a lower bound, since dropping regularisers can raise a score as easily as lower it — and in fact it raised this one (the port of the

Table 11: n=5 vs n=10 on the ConceptDiscoveryBench `synthetic_concept` task. The within-family NMI ranking (CBM $\gg$ DetNAM $>$ ConceptNAM) is preserved, mean shifts are ≤ 0.02 NMI for every method, and standard-deviation estimates tighten or stay flat. No qualitative claim in the paper depends on the additional seeds. **Null for this cell (K=4, n=2000): NMI** 0.060**, MIG** 0.021**, DCI-D** 0.066 (`chance_floor.json`). ConceptNAM's MIG (0.013) is below its null and its DCI-D (0.090) is within one null spread of one. Data: `exp_sprint8_n10_merged.json`.

Method	n	NMI	MIG	DCI-D
ConceptNAM	5	0.102±0.008	0.017±0.010	0.073±0.042
ConceptNAM	10	0.095±0.011	0.013±0.008	0.090±0.038
DetNAM	5	0.239±0.088	0.158±0.108	0.406±0.203
DetNAM	10	0.259±0.078	0.174±0.094	0.444±0.177
CBM	5	0.694±0.022	0.617±0.021	0.972±0.010
CBM	10	0.690±0.026	0.614±0.026	0.973±0.012

official method below scores lower). A fair comparison needs the authors' method under our protocol, which the port below approximates; the same caution applies to NODE-GAM and NBM, which we adapt with a K-concept bottleneck (App. E), so every row here is a result for our variant, not for the original method. Run with the ConceptDiscoveryBench protocol on Synthetic-Concept (K=4, p=16, 10 seeds):

Table 12: A TabCBM-style re-implementation (not the published method; see text) on Synthetic-Concept under the ConceptDiscoveryBench protocol (10 seeds, same hyperparameters as Table 11). TabCBM lands between DetNAM (the NAM-family Level-1 deterministic baseline) and ConceptNAM (the VAE-family Level-1). **Null for this cell (K=4, n=2000): NMI** 0.060 (`chance_floor.json`); read as excess, TabCBM is 4.1× ConceptNAM and CBM is 4.4× TabCBM: the architecture span within Level 1 is nearly as large as the cross-supervision step on top of it, which remains the larger one (4.4×). Data: `exp_sprint8_n10_merged.json`.

Method	NMI	MIG	DCI-D	Mean RMSE
ConceptNAM (n=10)	0.095±0.011	0.013±0.008	0.090±0.038	0.822±0.011
TabCBM (n=10)	0.205±0.044	0.101±0.057	0.287±0.107	0.177±0.014
DetNAM (n=10)	0.259±0.078	0.174±0.094	0.444±0.177	0.406±0.046
CBM (n=10)	**0.690±0.026**	**0.614±0.026**	**0.973±0.012**	**0.134±0.006**

A faithful port of the official TabCBM. Because the entry above is not the published method, we ported the authors' official implementation (TensorFlow; public repository, commit recorded in `tabcbm_official.json`) to PyTorch (`baselines/tabcbm_official.py`): its concept-mask generators and concept scorer, all three published regularisers (coherence with its k-nearest-neighbour term, diversity, specificity) at the official default weights, and its three-stage training schedule (pre-training, self-supervised mask training, main training), with the supervised concept head disabled for Level 1. Unavoidable deviations are listed in the module docstring against official file and line: the official task loss is classification-only, so a continuous target uses MSE; the official automatic `top_k` is zero for a continuous target, so we pass the value every official configuration uses; random streams differ from TensorFlow's. The regularisers and the full loss match an independent re-computation of the official graph to 10^{-8}; we could not compare against TensorFlow itself. Same cell, same 10 seeds (`exp_tabcbm_official.py`):

The placement survives: with the official regularisers TabCBM still lands between ConceptNAM and DetNAM on NMI, MIG and DCI-D. But our reduced variant *overstated* it — the official port is 0.06 NMI lower (paired t, p=0.018), its MIG excess over the cell's null is not significant, and the official specificity weight drives every concept-mask gate below 0.5, so the model predicts through soft, heavily shrunk masks. The ratios quoted in the caption of Table 12 describe the reduced variant only; for the published method the within-Level-1 span it sits in is narrower. No headline claim involves TabCBM; the cross-supervision gap to CBM (0.690) is unaffected.

Table 13: Official-regulariser TabCBM port against the reduced TabCBM-style baseline, re-run side by side (Synthetic-Concept K=4, n=2000, 10 seeds). "Budget" runs every stage for 40 epochs at batch 64 instead of the official schedule. Data: `tabcbm_official.json`.

Variant	NMI	MIG	DCI-D	task R^2
Official port	0.141 ± 0.049	0.040 ± 0.044	0.173 ± 0.066	0.841 ± 0.021
Official port, budget	0.139 ± 0.034	0.042 ± 0.033	0.176 ± 0.064	0.837 ± 0.023
Reduced (re-run)	0.203 ± 0.041	0.101 ± 0.053	0.283 ± 0.103	0.955 ± 0.006

D.3 Bootstrap CI on the 5.6× clinical-variance point estimate

App. N reports that ConceptNAM's seed-level AUC standard deviation is 5.6× iCBM's on UCI Heart and 12.6× on Pima (both point estimates). The Bartlett variance test rejects equal-variance at $p = 0.006$ (Heart) and $p = 0.0003$ (Pima), but Bartlett is sensitive to small n and normality. We supplement with a non-parametric 10,000-replicate bootstrap CI on each method's seed-level std and on the ConceptNAM/iCBM std ratio:

Table 14: Bootstrap (10,000 resamples) 95% CI on each method's seed-level AUC std and on the ConceptNAM/iCBM std ratio. The paired ratio CI and the $P[r>3]$ column come from the paired scheme; the paired CI excludes 1 on UCI Heart, but that scheme is anti-conservative — under equal variances it excludes 1 in 9% of simulated datasets against a nominal 5% (`paired_bootstrap_coverage.json`) — and the independent CI ($[0.8, 30.5]$), which errs the other way (4%), includes 1. On Pima the wide ratio CI is driven by a single-seed ConceptNAM outlier producing a bimodal AUC distribution. "Med." is the bootstrap median; for Pima it is 12.5, but the wide CI is the honest summary statistic. Bootstrap re-seeded to a deterministic constant in `experiments/bootstrap_clinical_std.py`. Data: `exp_clinical_bootstrap.json`.

Dataset	ConceptNAM std (CI95)	iCBM std (CI95)	Ratio (CI95)	Med.	$P[r>3]$
UCI Heart	0.128 [0.019, 0.155]	0.023 [0.004, 0.030]	5.58 [3.04, 16.49]	5.5	0.98
Pima	0.189 [0.006, 0.235]	0.015 [0.006, 0.019]	12.60 [0.62, 38.55]	12.5	0.68

The ratio's tight CI on UCI Heart strengthens the 5.6× claim only on the anti-conservative paired scheme; on the conservative independent scheme the interval includes 1 and no reduction is established at the 95% level. On Pima the CI is wide because one ConceptNAM seed produces a sign-flipped concept basis and a corresponding AUC near 0.5 (the bimodal distribution), which inflates the bootstrap variance of the std estimator on a ConceptNAM sample of size five. The point estimate of 12.6× is therefore unreliable; the bootstrap probability is flat across the $[2, 3]$ ratio band ($P[\text{ratio}>2] = P[\text{ratio}>3] = 0.68$, both equal due to bimodality), so no Pima claim survives beyond the point estimate: 68% of bootstrap replicates exceed a ratio of 2, but a bootstrap proportion is not a confidence level for the population ratio, and the 95% interval includes 1.

E ConceptDiscoveryBench: a benchmark for tabular concept discovery

Composition. ConceptDiscoveryBench v0.1 contains three controlled-ground-truth datasets:

- **Synthetic-Concept** (K=4, p=16): tanh-encoded sparse-W concepts; the body benchmark of §5.1.
- **MultiFactor** (K=5, p=32): Higgins-style factor product (multi-cause non-linear targets).
- **ScalingSynth** ($(K, p) \in \{(4, 16), (8, 32), (16, 64), (32, 128)\}$): an explicit scaling sweep for assessing how methods degrade with concept count and input dimensionality.

Every dataset exposes ground-truth concept values and standardised 5-replicate seeds. **MultiFactor is the documented exception to the fixed-W invariant**: its generator regenerates W for every

Table 15: Floor-effect control, **per method**. All values are *excess over the per-cell null* — an untrained random linear encoder under the same metric and split (`analyze_chance_floor.py`); raw NMI is excess + null. "TOST" counts equivalent pairs among the VAE triplet at δ=0.05, a margin wider than the triplet's largest excess over chance in every cell where it returns a positive verdict ($1.7\times$–$2.2\times$ on the headline cell, §4.4); read a positive verdict as an upper bound on the effect, not as evidence of sameness. We print every method rather than a summary because *no aggregate of this table is robust*: the per-cell maximum falls with difficulty ($\times 0.96$), the mean rises ($\times 1.46$) and the median rises faster ($\times 2.97$), against CBM's $\times 1.41$. NBM is the largest Level-1 excess in every cell, so any figure computed without it describes DetNAM rather than Level-1.

			Level-1 excess over null						L3	
K	σ	null	DetNAM	NBM	NODE-GAM	ConceptNAM	β-V+s	Fac-V+s	CBM	TOST
4	0.1	0.115	0.077	**0.140**	0.032	0.029	0.028	0.030	0.519	3/3
4	0.01	0.115	0.070	**0.180**	0.027	0.030	0.022	0.034	0.531	3/3
2	0.1	0.105	0.102	**0.124**	0.114	0.086	0.012	0.043	0.648	0/3
2	0.01	0.105	0.107	**0.135**	0.108	0.074	0.012	0.051	**0.731**	0/3

replicate, so its `w_hash` is `None` (`cdnam/data/benchmark.py`, `bench_summary.json` records `fixed_W: false`) and its seed-to-seed variance mixes sample noise with generator noise. Every other cell holds W fixed across replicates and the iterator asserts the shared hash. This matters where it is easiest to forget: MultiFactor is the one cell in which no task-supervised method clears its floor (§3), and that row should be read as Higgins-style factor recovery rather than as fixed-W identifiability. A machine-readable manifest accompanies the release.

Table 16: Where ConceptDiscoveryBench sits relative to existing disentanglement / concept-discovery benchmark suites. "Matched" means the suite requires all entries to declare a supervision budget and compares them only against same-budget peers. "Multi-metric" means the headline ranking is over ≥ 3 metrics rather than a single (typically NMI) score. "Leakage-controlled" means train/test/scaler separation is enforced by a shared utility, not left to each entry. "Self-audit" means a public protocol-revision log with before/after deltas is shipped with the artefact. *Scope:* TabCBM and CEM are method papers, not suites; they are listed because they are the closest tabular / concept-discovery comparators, and a dash means the property is not part of what the work ships as a protocol — not a judgement on its contribution.

Benchmark / suite	Matched	Multi-metric	Leakage-ctl	Self-audit
Locatello et al. 2019 (Locatello et al., 2019)	–	✓	partial	–
TabCBM (Espinosa Zarlenga et al., 2023)	–	–	–	–
GlanceNets (Marconato et al., 2022)	–	partial	✓	–
CEM (Espinosa Zarlenga et al., 2022)	–	–	–	–
Skirzyński et al. 2026 (Skirzyński et al., 2026)	–	–	partial	–
ConceptDiscoveryBench (ours)	✓	✓	✓	✓

Differences from prior tabular / disentanglement benchmarks. The matched-supervision and self-audit columns are the two characteristics that no prior *suite* in our review provides; Klotz et al. (2026) come closest, using trained-vs-untrained separation to select metrics without shipping it as a benchmark property. Skirzyński et al. (2026) is the closest contemporary suite — synthetic generators with controllable concept ground truth, aimed at CBM decision-support and automation — and it strengthens the survey point rather than weakening it: its evaluation is accuracy-based (concept labels given by construction), it reports none of the recovery metrics surveyed here, and it states no chance level for any of them (its no-signal reference is a majority-class accuracy rate, not a null for a recovery score); its "partial" leakage entry reflects explicit annotation-noise controls without a selection-protocol constraint. The leakage-control entry for Locatello et al. is marked "partial" because they enforce no scaler-on-training but their checkpoint-selection protocol is paper-level rather than benchmark-level.

The reporting survey, in full. The abstract's claim — that of 8 sampled papers reporting one of these metrics, none reports a chance level — describes one keyword sample, not the field, and the "often" beside it is no stronger than eight papers allow; it is stated here at the level of detail that lets a reader overturn it. *Frame*: arXiv cs.LG, three abstract queries — `abs:"disentangled representation"`, `abs:"concept bottleneck"`, `abs:"disentanglement metric"` — restricted to 2023 or later, taking the most recent results whose PDF we could fetch and whose text extracted to at least 5000 characters; 34 papers passed that screen. *Inclusion*: the full text reports MIG, DCI, NMI, SAP, or modularity/compactness; 8 did, and every in-scope hit is from 2026. *Outcome*: does the text anywhere match a chance level, chance correction, adjusted mutual information, randomly initialised or untrained encoder, permutation null, null distribution, above-chance, or random baseline? None did. The 8 in-scope arXiv identifiers are `2601.21688`, `2602.05214`, `2602.06741`, `2602.10058`, `2603.11790`, `2606.09725`, `2606.21385` and `2608.16245`; they are also listed in `results/reporting_survey.json`, and the regexes are the two compiled at the top of `experiments/analyze_reporting_survey.py`. *What this is not*: a census. It is a keyword search over a keyword sample, it can miss a chance level reported only inside a table or a figure axis, and the extracted-text cache it ran over is not shipped, so re-running it means re-fetching those identifiers rather than replaying a stored input. We report it because a bounded, checkable number is better than "almost always", not because the sample is large.

E.1 What the closest prior work actually measured

§2 credits Cao et al. (2022) with the untrained-encoder check, so it should be precise about what they ran. Their check is a two-table sanity check in their appendix, aimed at showing that their proposed metric (MED) is stable across representation dimension while the five existing ones are not. The two random encoders are *different*. On Shapes3D, a 10-d randomly initialised encoder scores MIG 6.7 and SAP 3.0, against a trained 1000-d BYOL model's 5.2 and 2.8 (their Table 6). On dSprites, a 1000-d randomly initialised encoder scores BetaVAE score 82.7, FactorVAE score 61.4 and DCI 19.8, against a trained 10-d DIP-VAE-II's 81.5, 58.6 and 12.3 (their Table 7). All values are on a 0–100 scale and no standard deviations are reported for the random rows. So there is no single random baseline carried through five metrics, and the overlap with the metric family this paper measures is MIG and DCI only; BetaVAE and FactorVAE scores are classifier-based and outside it. Their stated reading is that these metrics are *unfair across representation dimension* — a high-dimensional random model can out-score a well-trained low-dimensional one. That the same numbers also imply a chance floor far from zero is our inference, not a claim they make; what they show is that an untrained encoder scores well.

Carbonneau et al. (2022). Their headline finding is the *opposite* of a floor: "the majority of metrics attribute a score near 0 to complete noise". The exceptions they name are DCI's modularity and compactness under a lasso regressor (above 0.3, attributed to the ℓ_1 penalty, and absent under a random forest) and Z-max Variance. They do not prescribe *coarser discretisation* as a remedy for chance inflation: they offer coarser binning for Z-max Variance and *finer* binning for IRS in the same paragraph, and state that "no procedure has been proposed yet to strike the right balance". Chance correction is not a topic of their paper, and the $10 \to 3$ bin result in §3 is ours, standing on its own measurement.

Hsu et al. (2023). InfoM and InfoC subtract the equal-information level ($1/n_s$ and $1/n_z$ respectively) and renormalise to $[0, 1]$, and KSG replaces binned MI for continuous latents at $k{=}3$. The structure is a chance-level normalisation, but the word chance does not appear in their paper and they run no test against a null, so we describe the construction rather than attributing the concept. Their Fig. 6 states the metrics are "very robust to the choice of k", which is what App. A reports.

Kim et al. (2018). The published significance test retrains each CAV about 500 times against resampled sets of real counterexample images and runs a Bonferroni-corrected two-sided t-test against a *constant* TCAV score of 0.5. The random-concept comparison group — a genuinely data-dependent null — is in their released implementation rather than in the paper, so we do not describe the paper as building one.

Gröger et al. (2026). Under their permutation calibration the correlation between representational similarity and model capability collapses for global metrics (linear CKA $0.86 \rightarrow 0.45$, Procrustes $0.89 \rightarrow 0.39$) but survives for local ones (mKNN $0.85 \rightarrow 0.84$). In their own words this "refines rather than refutes" a published result, and we use that narrower claim.

Methods. We evaluate six method families on ConceptDiscoveryBench: vanilla NAM (Agarwal et al., 2021) (Level-1, task baseline), DetNAM (NAM-family Level-1, deterministic bottleneck), ConceptNAM (NAM∩VAE-family Level-1, VIB bottleneck), NODE-GAM and NBM (NAM-family Level-1 variants with stronger per-feature shape functions; minimal but faithful K-concept adaptations), and CBM-joint (Level-3, full concept supervision). Per method, we use the same training schedule (n_{samples}=2000, 40 epochs, Adam 10^{-3}, batch 64, 30% held-out test split) and identical splits. The body Table 1 uses a smaller (n_{samples}=800, 50-epoch) protocol; both protocols preserve the same method ranking but the smaller-n protocol has a higher per-method NMI value because NMI's discretisation noise floor scales with $\sqrt{n_{\text{test}}}$.

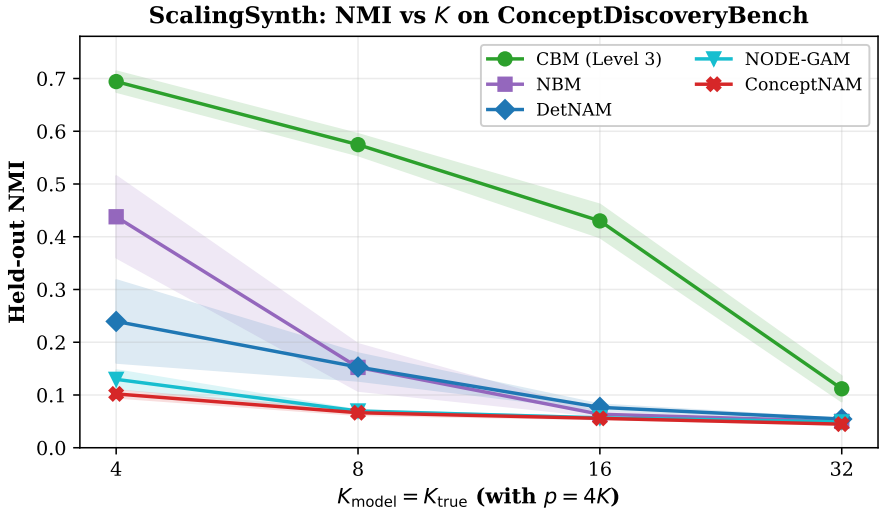


Figure 1: Held-out NMI on the ScalingSynth sweep (K_{model}=K_{true}, p=$4K$; 5 seeds; shaded ±std). The cross-family hierarchy of body Table 1 (CBM ≫ NAM-family) holds at K=4 but contracts as K grows: NBM tracks DetNAM at $K \geq 8$, all NAM-family methods plateau at NMI$\sim$0.05 for $K \geq 16$ — and that plateau *is* the floor: the measured nulls for those cells are 0.047 and 0.044 (Table 17), so the Level-1 curves are within a few thousandths of chance there, not resting on a shared ceiling above it. Only CBM (Level 3, full concept supervision) stays measurably above its null at every K, though it too falls, 0.69 to 0.11; for the Level-1 curves the cells stop resolving them from noise. *iCBM is omitted from this plot:* at zero concept supervision (the matched-Level-1 setting on this scaling sweep), iCBM reduces to DetNAM (the Hungarian-alignment loss has no labels to align), so the DetNAM curve is also the iCBM-at-r=0 curve. At r>0, iCBM interpolates between DetNAM and CBM (Table 6); its scaling behaviour at non-zero supervision is bounded by the same fall that limits CBM. Drawn from `bench_summary.json`, i.e. the shipped configuration.

Reading. Three findings extend the body's negative result. (i) *ConceptNAM posterior collapse at $K \geq 8$.* The S3 finding of single-active-unit collapse at high p (§5.4) generalises to every $K_{\text{model}} \geq 8$ with p=$4K$: the VIB drops the effective concept count to $\sim$ 1 as soon as the model is over-parameterised. The non-VIB methods (DetNAM, NODE-GAM, NBM) all retain K_{model} active dimensions but their NMI also degrades, isolating "concept quality" from "concept dimensionality used". (ii) *At small K, NBM and DetNAM are within seed noise of each other* (corrected run: DetNAM 0.51 vs NBM 0.48). *Under the shipped defaults NBM appeared to lead substantially* (0.44 vs 0.24 at K=4); that ordering did not survive the configuration audit, so we offer no mechanism for it. At $K \geq 16$ all three NAM-family methods land in the same mid-tier band. (iii) *CBM (Level 3) also collapses at K=32.* Even with full concept supervision, the cross-family hierarchy folds at (K=32, p=128): CBM NMI drops from 0.69→0.11 across the sweep. Scaling on tabular concept

Table 17: Mean NMI (held-out, fixed-W where applicable; n_{samples}=2000, 40 epochs, 5 seeds) on ConceptDiscoveryBench. Bold = best; underline = best non-CBM. NAM has no concept output and is omitted. **The upper block is the benchmark as it should be read.** Every method's un-swept encoder and loss-scale flags were selected on held-out task RMSE — a criterion that uses no concept labels — and the run redone (`exp_bench_corrected.py`). The lower block gives the shipped configuration used by the rest of this paper, for comparison: at K=4, DetNAM moves $0.24 \rightarrow 0.51$ and NBM $0.44 \rightarrow 0.48$, so the Level-1-to-Level-3 gap this benchmark displays was $1.28\times$ wider than a fairly-tuned comparator shows ($0.256 \rightarrow 0.200$, CBM minus the best Level-1 in each block). *Null, measured for every row* under this exact protocol (`analyze_chance_floor.py`, 40 random encoders $\times$ 5 seeds per cell); grey marks every entry that does not exceed the 95th percentile of its own cell's null. **The entire Level-1 half of the MultiFactor row is at or below chance**: its null's 95th percentile is 0.151, above all four Level-1 entries (0.089–0.126), none more than 0.0006 above its mean (0.126), so no Level-1 entry there is distinguishable from chance (highest, DetNAM: Welch p=0.49) — only CBM's 0.49 clears. That is a statement about each entry against the floor; whether two Level-1 methods differ from *each other* there is a separate, pairwise question, which a floor verdict neither answers nor forbids. It is the one cell in that state. *On the cell count:* the benchmark is specified as six rows, but `ScalingSynth` (K=4, p=16) and Synthetic-Concept are the same generator at the same settings and produce byte-identical results, so there are five distinct cells and we count them that way. The duplicate row is kept in the table because the two names are both in use in the literature this benchmark draws from, but it is one measurement, not two. The $K \geq 16$ cells do clear their nulls, but barely: 0.048–0.064 against p_{95} of 0.050 and 0.045, margins of a few thousandths of NMI, so they separate methods from noise without separating methods from each other. Only K=4 discriminates with room to spare; at K=8 three Level-1 entries clear but DetNAM clears only the p_{95} screen, not Welch with Holm (p=0.017), so we do not count it. At K=4 the null is 0.060 (the canonical cells-grid estimate for this configuration; the bench-protocol run of the same cell gives 0.062, within draw noise) and as excess over it DetNAM captures 69% of CBM's, NBM 65%. Table 21 gives each cell's resolution, the sample size at which a method first clears its own floor, which is the number that says how far a cell can be pushed. The 0.110 MultiFactor null reported in App. B was measured at n=5000 and does not transfer. *Capacity.* The parameter row shows the capacity budget is not matched (7.1$\times$ at K=4): our protocol matches the supervision budget and leaves model size free, so within-level comparisons here are not capacity-matched; an equal-parameter control on the K=4 cell leaves the ordering unchanged (Limitation (8), `capacity_matched.json`).

Dataset	ConceptNAM	DetNAM	NODE-GAM	NBM	CBM	null
Selected configuration (`bench_corrected.json`)						
Synthetic-Concept (K=4)	0.12	<u>0.51</u>	0.12	0.48	**0.71**	0.060
MultiFactor (K=5)	0.09	0.13	0.10	0.11	**0.49**	0.126
ScalingSynth (K=8, p=32)	0.07	<u>0.20</u>	0.07	0.15	**0.59**	0.052
ScalingSynth (K=16, p=64)	0.055	<u>0.064</u>	0.057	0.064	**0.47**	0.047
ScalingSynth (K=32, p=128)	0.048	<u>0.052</u>	0.049	0.050	**0.12**	0.044
Shipped configuration, for comparison (`exp_bench_*.json`)						
Synthetic-Concept (K=4)	0.10	0.24	0.13	0.44	0.69	0.060
ScalingSynth (K=32, p=128)	0.05	0.05	0.05	0.05	0.11	0.044
Trainable parameters at K=4, p=16						
	93k	82k	13k	28k	45k	—

discovery is therefore not a "supervision budget" problem alone: CBM's own fall is consistent with an architectural ceiling. The Level-1 curves are not — at $K \geq 16$ they sit within a few thousandths of their cells' measured floors, so what looks like a shared plateau is the floor (Q9), and we do not read it as a ceiling.

Datasheet (Gebru et al., 2021). (i) *Motivation*: a tabular concept-discovery test bed where ground-truth concepts are known, so NMI/MIG/DCI/ACS are meaningful and not first-K-feature proxies. (ii) *Composition*: 3 controlled synthetic datasets; $\sim$ 2000 samples each at the standard configuration; no personal data. (iii) *Collection*: synthetic; no human subjects. (iv) *Preprocessing*: standardisation handled by the methods. (v) *Uses*: evaluation of unsupervised, task-supervised, and concept-

Table 18: Active concepts (mean count of z-dimensions with $\sigma > 0.05\,\sigma_{\max}$) on the same runs. ConceptNAM's VIB collapses the posterior at $K_{\text{model}} \geq 8$; non-VIB methods retain all dimensions. CBM (full supervision) shows a different failure mode — its NMI collapses at $K=32$ while it formally retains all 32 active dimensions.

K_{model}	ConceptNAM	DetNAM	NODE-GAM	NBM	CBM
4	4.0	4.0	4.0	4.0	4.0
8	**1.2**	8.0	8.0	8.0	8.0
16	**1.0**	16.0	16.0	16.0	16.0
32	**1.0**	32.0	32.0	32.0	32.0

Table 19: Task RMSE (lower is better; 5 seeds) on ConceptDiscoveryBench including the vanilla NAM (no concept output) for completeness. NAM and DetNAM share the same per-feature MLP encoder but DetNAM adds the K-concept bottleneck.

Dataset	ConceptNAM	DetNAM	NAM	NODE-GAM	NBM	CBM
Synthetic-Concept ($K=4$)	0.82	0.41	0.49	0.45	0.19	**0.13**
MultiFactor ($K=5$)	0.47	0.41	0.54	0.47	**0.27**	0.32
ScalingSynth ($K=8$, $p=32$)	0.81	0.70	0.73	0.69	0.57	**0.33**
ScalingSynth ($K=16$, $p=64$)	1.08	1.28	1.07	1.11	1.00	**0.87**
ScalingSynth ($K=32$, $p=128$)	1.82	2.10	1.77	1.79	1.44	**1.41**

supervised concept-discovery methods. (vi) *Distribution*: released under MIT in the supplementary archive. (vii) *Maintenance*: versioned; v0.1 is the canonical release for this paper.

F ACS-IoU sensitivity to top-k

Table 4 reports ACS-IoU at $k=3$. The sensitivity to k is shown below (10 seeds, fixed-W; JSON path in App. W). CBM is highest at every k; among the non-CBM methods the ordering is k-dependent, which is the point of the sweep.

Table 20: ACS-IoU at top-k across methods (mean across pairs of $\binom{10}{2}=45$ seed pairs after Hungarian alignment). **Chance rises with** k: the bottom row is what random row-normalised attributions score under the same pairing and matching (`acs_topk_null.json`, 200 draws; method rows: `exp_topk_iou_sensitivity.json`) — at $p=16$, two random top-8 sets share half their features in expectation, so part of any climb with k is the chance level climbing. Read the methods as excess over that row.

Method	top-1	top-2	top-3	top-5	top-8
CBM joint	**1.000±0.000**	**0.922±0.083**	**1.000±0.000**	**0.898±0.058**	**0.794±0.048**
DetNAM	0.628±0.195	0.535±0.159	0.450±0.107	0.381±0.091	0.437±0.066
ConceptNAM	0.478±0.320	0.580±0.191	0.703±0.214	0.730±0.143	0.646±0.066
NBM	0.456±0.260	0.587±0.123	0.598±0.114	0.601±0.100	0.572±0.064
NODE-GAM	0.422±0.337	0.513±0.277	0.562±0.193	0.708±0.130	0.711±0.122
β-VAE+sup	0.333±0.236	0.404±0.131	0.526±0.123	0.629±0.129	0.614±0.082
random attribution (null)	0.138	0.154	0.192	0.271	0.401

Read on the excess scale, as the caption says. Every method clears its per-k null's p_{95} at every k, and the *rank-1* method at each k is the same on both scales, so the k-dependence of the non-CBM ordering stands and so does "CBM most stable" (Table 4): CBM leads by 0.863 at $k=1$ and 0.393 at $k=8$. What does not stand is the shape. On raw IoU, NODE-GAM climbs monotonically with k ($0.422 \rightarrow 0.711$) and overtakes everyone but CBM at $k=8$; on excess it peaks at $k=5$ (0.438) and *falls* to 0.310 at $k=8$, because the null climbs faster than it does ($0.14 \rightarrow 0.40$ from $k=1$ to $k=8$). NODE-GAM's raw climb with k is therefore not a statement about NODE-GAM: the climb is the floor. The same holds for every method — all six excesses peak at $k \leq 5$ and decline at $k=8$, CBM

included. DetNAM's degradation, by contrast, is real and sharper on the excess scale: 0.490 at $k{=}1$ down to 0.036 at $k{=}8$, which is within noise of its own null. ConceptNAM peaks at $k{=}3$. The practical consequence for a benchmark user is that top-k IoU must not be compared across k without its per-k floor, and that the body's $k{=}3$ default sits near the peak of the excess curve rather than in an arbitrary place.

Our own ACS null was unmatched, so we matched it. The null above draws attributions as $|\mathcal{N}(0,1)|$ with every entry independent. Real attribution rows are concentrated — the single largest feature carries between 0.12 and 0.33 of a row's mass here — and ACS-IoU is scored *after* a cosine-Hungarian assignment, so the concentration feeds the same matching step §3 decomposes for NMI. A null built from diffuse rows can therefore understate the chance level of a statistic computed on concentrated ones: our own metric, with our own defect. `analyze_acs_matched_null.py` measures the matched version, permuting the entries within each row of each method's real attribution matrix, which preserves that row's value profile exactly and destroys only which feature carries which mass. The concern is real and bounded. At top-1 the matched null runs up to $1.34\times$ the unmatched one (0.184 against 0.138, DetNAM), it rises with row concentration across methods, and it exceeds the unmatched null at $k \leq 3$ for every method except β-VAE+sup, whose rows are the least concentrated (0.12) and whose top-1 ratio is 0.979; at $k \geq 5$ the two agree to within 8%. No verdict moves: 0 entries stop clearing their null, and the largest correction to any excess is DetNAM's top-1, from 0.535 to 0.488 (`acs_matched_null.json`). *What this null does not preserve.* Permuting within a row destroys the cross-seed feature saliency profile along with the concept correspondence: the real row-averaged saliency vectors correlate at $r \approx 0.82$ across seed pairs, and under this null that correlation goes to zero. A construction that kept it — one shared feature permutation per seed — would move the top-1 null in the other direction. We report the stricter of the two and flag the choice, because neither is obviously the right matched null and the verdicts do not turn on it.

What that re-run also showed about this table's reproducibility. Re-deriving the attributions with the sweep's own trainers, on CPU and single-threaded, reproduces CBM's row to 0.001 but drifts up to 0.17 on the Level-1 rows. Attributions taken from models left in `train()` mode **cannot be the cause**: the shared helper `cdnam.utils.attribution.input_attribution` sets `model.eval()` before extracting, so the shipped table was already extracted deterministically. A check that bypasses the helper (`exp_acs_evalmode.py`, `acs_evalmode.json`) confirms evaluation-mode extraction is bit-identical on repeat ($60/60$ fits) and that the drift against the shipped table is the same with or without dropout at extraction for five of six methods. The drift is therefore between *re-trainings* of the same protocol; we have not isolated its source (device, library version and thread-level non-determinism all differ from the shipped runs). Re-scored against its own attribution-matched null, every re-trained method clears the null's 95th percentile at every k; CBM stays the most stable at every k and DetNAM the least at $k{\geq}5$, as in the table, while the middle rows reorder at $k{=}1$ and $k{=}5$. So the table's bookends are robust and its middle is not: a reader should not rank two Level-1 rows that differ by less than about 0.17 on the strength of it. The $\pm$ in Table 20 spans seed *pairs* and does not contain this re-training variability.

This table's null disagrees with the canonical one, and by how much. At the shared cell (Synthetic-Concept $K{=}4$, $n{=}2000$) three estimates disagree, each a run of the same cell: this driver gives a null mean of 0.0631, the bench-protocol nulls run gives 0.062, and the canonical chance-floor grid gives 0.0604; all three agree within draw noise, and the rest of this paragraph measures by how much rather than asserting it. A reader comparing the two artifacts sees two numbers for one declared quantity, so we measured which it is. The two are the same estimator: same generator config, same split, same $z = X_{\text{test}}W$ with $W \sim \mathcal{N}(0,1)^{p\times K}$, same 40 draws $\times\, 5$ seeds, differing only in the RNG stream offset. Replaying this driver's offset reproduces its 0.063092847899 to twelve digits, and re-running that estimator under 400 further independent streams gives mean 0.0612, sd 0.0009, range $[0.0584, 0.0636]$. Both shipped values lie inside that range, at -0.95 and $+2.09$ standard deviations of the estimator's own spread (the 16th and 98th percentiles; we lead with the z because at 40 replicates the percentiles carry 2.5% granularity and are not stable). So the gap is Monte-Carlo error of a 200-draw mean, not a second estimand; what it does show is that a 200-draw null carries about ± 0.0009 of its own, which is small against the excesses this paper reads but not against nothing (`null_mc_error.json`, `analyze_null_mc_error.py`). The canonical value remains the

Table 21: **Resolution of each benchmark cell**: the smallest n at which a trained method's 5-seed mean NMI exceeds the 95th percentile of *the* 5-*seed-mean* untrained-encoder null (40 random encoders $\times$ 5 seeds per $(cell, n)$). DetNAM stands in as the strongest collapse-free Level-1 method and CBM-joint as the Level-3 anchor (`exp_cell_resolution.py`, at the selected encoder configuration and batch size 64, matching the corrected benchmark run). Every p_{95} column above is the matched (5-seed-mean) one. The null's p_{95} falls roughly as $1/\sqrt{n}$, which is why a cell that ranks nothing at one sample size can rank cleanly at another. The benchmark ships at n=2000. MultiFactor is the outlier: Level-3 resolves there at n=500 but Level-1 not until n=8000 on this grid, so its Level-1 row in Table 17 carries no floor verdict at the shipped n — a fact about the cell, not about the methods. We report this rather than remove the cell, because the number a benchmark user needs is how much data the cell demands, not whether we chose to include it. **Resolutions are grid labels, not measurements with a CI**: each is a point estimate crossing a p_{95} order statistic from 40 draws. Entries at 500 are left-censored by the grid; two sit inside noise — ScalingSynth K=32 at n=500, where DetNAM clears by +0.002 while CBM misses by 0.0004 (hence CBM's 1000), and MultiFactor DetNAM at 4000, which misses by 0.004 against a seed-SEM of 0.014, so its 8000 reads "beyond the shipped 2000", not a sharp threshold. **Resolution is not a pure measurement limit.** Raising n grows the test set, the training set and the gradient steps per epoch together, so a resolution is where one model, one training budget, one split and one rule cross. We separated the two (`exp_resolution_decomposed.py`, `resolution_decomposed.json`): design A fixes the shipped training set (1400) and grows only n_{test} (150 $\rightarrow$ 2400); design B fixes n_{test}=600 and grows only n_{train} (350 $\rightarrow$ 5600). On MultiFactor, evaluation size is not the constraint: under design A DetNAM tracks its null at every size (Welch $p \geq 0.74$). Training size is: under design B its mean rises from below the null to 0.017 above it at 5600, passing the p_{95} screen from 2800 but not Welch with Holm (p=0.061 at 5600). CBM clears at the smallest size of both designs, and on Synthetic-Concept both methods do too. MultiFactor's Level-1 row is therefore limited by how much data the *learner* gets, not by how precisely the metric is estimated. The first crossing on the grid is also selected from five points without error control. The criterion printed is the p_{95} screen; the declared Welch rule with Holm over all 50 tests agrees on 49 and on every first-resolving n (App. B.1).

	null p_{95} at n =					resolves at n =	
Cell	500	1000	2000	4000	8000	DetNAM	CBM
Synthetic-Concept (K=4, p=16)	0.185	0.105	0.072	0.056	0.048	500	500
MultiFactor (K=5, p=32)	0.237	0.169	0.136	0.126	0.113	8000	500
ScalingSynth (K=8, p=32)	0.177	0.093	0.054	0.037	0.029	500	500
ScalingSynth (K=16, p=64)	0.179	0.089	0.049	0.030	0.022	500	500
ScalingSynth (K=32, p=128)	0.182	0.089	0.045	0.025	0.016	500	1000

one to cite, because it is the one §3 declares and every excess in this paper is taken against it; we do not average the two.

G Encoder-configuration ablation and full rerun

Two encoder flags (`use_exu`, `dropout`) and the VAE reconstruction scale carried un-swept defaults through every driver in this repository; no artefact varied them. They should have been swept. `exp_bench_corrected.py` sweeps each method over the flags that apply to it, selects on **held-out task RMSE** (held-out reconstruction MSE for Level-0) and reruns the whole benchmark at the picks. Every driver behind this appendix and behind Table 17 uses a three-way split (55/15/30): the configuration is picked by task RMSE (reconstruction MSE for Level-0) on the validation split, and every number reported here comes from the disjoint test split. That costs absolute performance — training sees 55% of the data instead of 70% — and these are the numbers to cite. Selections: no ExU and no dropout for ConceptNAM and DetNAM, no dropout for NBM, shipped dropout for NODE-GAM, and a summed-over-inputs reconstruction with β=1 / γ=10 for the VAE family.

The defaults were costing the comparators, not us. DetNAM moves 0.168 $\rightarrow$ 0.246 and NBM 0.226 $\rightarrow$ 0.295, while ConceptNAM moves 0.145 $\rightarrow$ 0.139 — its binding constraint is the VIB, not the encoder, which is why the same default depressed the NAM-family comparators and left our own method where it was (here slightly *below* it, within one standard deviation). NODE-GAM likewise

Table 22: Configuration sweep on the headline cell, selection criterion in bold. K=4, p=16, n=800, 50 epochs, 5 seeds, fixed W, `method_seed` per method, three-way $55/15/30$ split. The RMSE column is the *validation* task RMSE the selection reads; task R^2 and NMI are from the disjoint test split. Source: `bench_corrected.json` (`phase1_selection_sweep`); `encoder_ablation.json` runs the same grid for DetNAM / ConceptNAM / NAM under the shared-seed convention of Table 1. **Null for this cell (K=4, n=800): NMI** 0.115 (`null_calibration.json`). The reselection moves ConceptNAM $0.145 \to 0.139$ and NODE-GAM $0.149 \to 0.141$ — differences of 0.006–0.008, at or below the uncorrected clearing threshold of 0.008, so "its binding constraint is the VIB, not the encoder" is a statement about a difference this design cannot resolve. Two of the four DetNAM cells differ from Table 1 by 0.022 and 0.026.

Method	configuration	**task RMSE**	task R^2	NMI
DetNAM	shipped (exu, drop 0.1)	0.498	+0.608	0.168
	exu, drop 0	0.497	+0.625	0.179
	no exu, drop 0.1	0.380	+0.798	0.247
	selected: no exu, drop 0	**0.379**	+0.803	0.246
NBM	shipped (drop 0.1)	0.383	+0.786	0.226
	selected: drop 0	**0.308**	+0.854	0.295
ConceptNAM	shipped	0.846	+0.019	0.145
	no exu, drop 0.1	0.828	+0.054	0.146
	selected: no exu, drop 0	**0.812**	+0.092	0.139
NODE-GAM	shipped (drop 0.1)	0.514	+0.604	0.149
	selected: drop 0	**0.501**	+0.631	0.141

moves $0.149 \to 0.141$: the selection improves its task fit (R^2 $+0.604 \to +0.631$) without improving concept recovery, so the effect is specific to the two NAM-family methods that recover concepts at all, not a general consequence of tuning. That is also the one place where the Level-1-legal criterion and NMI disagree: for DetNAM and NBM they point the same way, for ConceptNAM and NODE-GAM they do not, which is the expected behaviour of a criterion that is genuinely blind to the concepts.

ExU is prescribed by Agarwal et al. (2021) for *jagged* per-feature shapes, paired there with a clamped ReLU-n and a $\mathcal{N}(x, 0.5)$ initialisation with $x \in [3, 4]$. Our `FeatureMLP` uses a plain ReLU and relaxes the initialisation to mean 0.5, and this generator's concepts are smooth ($\tanh(Wx + b)$). The default was an off-label application of the original prescription and we did not check it.

Effect on the paper. On the headline cell the best Level-1 excess over the null rises from $+0.077$ (DetNAM, shipped) to 0.180 (NBM, selected), i.e. from 15% to 36% of CBM's; at the n=2000 benchmark protocol DetNAM reaches 0.506 against CBM's 0.705, or 69% of the excess over that protocol's null (Table 17). The Level-1-to-Level-3 NMI gap on the headline cell narrows from 0.442 to 0.323, so the shipped configuration's margin is about $1.37\times$ too wide there, and $1.28\times$ at n=2000 under the same best-Level-1 definition ($0.256 \to 0.200$). The direction of the paper's finding is unchanged: concept labels still buy more than any architecture we tested, at every cell, by a margin no configuration search closed. Both blocks are shown in Table 1 and Table 17; the corrected one is the one to cite.

H Reconstruction-scale ablation for the VAE family

Our VAE baselines average the reconstruction term over the feature axis while summing the KL over the latent axis (`baselines/beta_vae.py`). In the convention the β-VAE literature uses — reconstruction summed over inputs — the effective penalty is βp, so the shipped β=4 at p=16 is an effective β of 64, and the search grid $\{1, 4, 10\}$ lies wholly inside the collapse regime. This appendix reruns the family under both conventions (`exp_vae_scale_ablation.py`), selecting on held-out task RMSE for the Level-1 variants and held-out reconstruction MSE for Level-0 — neither uses concept labels.

Table 23: Reconstruction-scale ablation, headline protocol (K=4, p=16, n=800, 50 epochs, 5 seeds, fixed W, three-way $55/15/30$ split). "KL" is the held-out posterior KL in nats; a value near zero is complete collapse. *Selected* rows are chosen on the criterion each supervision level is allowed to see — validation task RMSE for the supervised arms, validation reconstruction MSE for Level-0 — computed on a split disjoint from the one every column here is reported on.

Method	arm	KL	task R^2	task RMSE	NMI
β-VAE+sup	*shipped* β=4	0.0003	−0.009	0.820	0.134
	recon-sum β=4	0.0015	−0.008	0.819	0.131
	recon-sum β=1	2.07	+0.576	0.532	0.128
	selected: recon-sum β=0.05	**11.29**	**+0.571**	**0.534**	0.121
FactorVAE+sup	*shipped* γ=10	0.074	+0.100	0.773	0.145
	selected: recon-sum γ=10	**2.02**	**+0.572**	**0.533**	0.122
	recon-sum, disc. lr 10^{-3}	1.96	+0.569	0.535	0.126
β-VAE (Level-0)	*shipped* β=1	0.0000	—	—	0.114
	selected: recon-sum β=1	**1.91**	—	—	0.116
	recon-sum β=0.05	11.21	—	—	0.112

Table 24: Supervision levels, the methods evaluated at each, and the residual ambiguity each level leaves. For Level 1, Theorem 1 gives $G_{\pm}$, not S_K; the reduction to S_K needs Corollary 1, whose hypothesis fails on our shape bank for every $K \geq 2$. Searching all $2^K K!$ elements of $G_{\pm}$ on our K=4 head returns exactly $\{I, \mathrm{diag}(1,-1,1,1)\}$ — a sign flip, with *no* non-trivial permutation. Directions are marked ($O(K)$ is a lower bound from Locatello et al. (2019)); for Level 1 we give only the symmetry of one fixed head, which is not an identifiability class; the column drives no experiment. *Scope of the Level-1 entry:* it is the orthogonal symmetry group of one *fixed* additive head (Theorem 1), a necessary condition, not the identifiability class of a jointly trained encoder and head — training can also move to a different head $\tilde{g}$, which the theorem does not exclude.

Level	Supervision	Residual ambiguity	Methods
0	None (reconstruction only)	$\supseteq O(K)$	β-VAE, FactorVAE
1	Task labels y	not characterised; fixed-head symmetry $\subseteq G_{\pm}$	ConceptNAM, β-VAE+sup, FactorVAE+sup, DetNAM, NBM, NODE-GAM, iCBM (r=0)
3	Concept labels c (full)	$\{\mathrm{id}\}$	CBM (joint), iCBM (r=1, MSE alignment; Hungarian: up to permutation)

The collapse is real; the architecture is not the cause. The shipped configuration does collapse — 0.0003 nats is a posterior indistinguishable from its prior — and the effective-supervision check caught it, which is what that check is for. But the near-zero R^2 is not evidence that *the architecture* cannot consume a task label. It can: correcting the scale moves β-VAE+sup from $R^2 = -0.009$ to $+0.571$ and FactorVAE+sup from $+0.100$ to $+0.572$.

What that correction does *not* do is improve concept recovery. NMI moves $0.134 \to 0.121$, $0.145 \to 0.122$ and $0.114 \to 0.116$ — down for both supervised variants and flat for the unsupervised one, and in every case within 0.03 of the untrained-encoder null (0.115). So the within-family equivalence is not explained by non-consumption, and §5.1's narrowing of the claim to "methods that consume little or none of the task label" is too weak. The finding these methods actually support is the stronger one: a VAE-family Level-1 method that *does* fit its task target, with an uncollapsed posterior and all four latents active, still recovers concepts no better than one that fits nothing.

Raising the FactorVAE discriminator's learning rate changes nothing either ($0.122 \to 0.126$), consistent with the TC term being inert at this scale rather than merely under-trained.

I VALIDATION-STOPPED TRAINING: NUMBER ROBUSTNESS

To defend against the "you under-trained method X" attack, we re-ran the headline Table 1 comparison with `train_val_test_indices` ($55/15/30$) and val-MSE-based early stopping (max 80 epochs, patience 15). The test NMI shifts are within the seed-level variance of the fixed-50-epoch Table 1 numbers, confirming that the headline hierarchy is not an artifact of the fixed-budget choice. Driver and JSON paths are in App. W.

Table 25: VIB-ablation summary across the three controlled experiments (5 seeds each; fixed W within each experiment). The single most-damning operating point per experiment is shown; full sweeps live in App. R. "active" counts z-dimensions with $\sigma > 0.05\,\sigma_{\max}$. The S2 row does not carry NMI: that metric's normaliser scales with K_{model}, so its values are not comparable to the $K_{\text{model}}=4$ rows above (App. R). S2 reports only what the artefact supports — that no method prunes its surplus dimensions, CBM included. **Nulls**: S1/S2 run at $K=4$, $n=1500$, whose floor is NMI 0.072; S3's blocks are 0.031 and 0.023 at their own cells (`sblock_nulls.json`). Data: `exp_s1_supervision_sweep.json`, `exp_s2_k_mismatch.json` and `exp_s3_high_dim.json`.

Exp. (operating point)	Metric	ConceptNAM	DetNAM (no VIB)	CBM
S1 no concept labels ($r=0$)	NMI	0.107	$\mathbf{0.226}$	0.132
S2 over-param. ($K_{\text{model}}=32$)	active	30.8	32.0	32.0
S3 high-p ($p=128, K=16$)	active / NMI	$\mathbf{1.0}\,/\,0.03$	$16.0\,/\,0.04$	$16.0\,/\,\mathbf{0.46}$
S3 real (Breast Cancer)	AUC	0.78 ± 0.36	$\mathbf{0.99\pm0.00}$	—

Table 26: Test NMI under val-based early stopping (5 seeds; max 80 epochs, patience 15). "best epoch" is the median across seeds. Compare to Table 1. **Null for this cell ($K=4$, $n=800$): NMI** 0.115 (`null_calibration.json`); ConceptNAM's 0.133 is $+0.018$ over it. Data: `exp6_full_comparison_valstop.json`, `exp6_full_comparison.json`.

Method	NMI (val-stop)	NMI (fixed-50, Tab. 1)	Δ	median best-epoch
ConceptNAM	0.133 ± 0.014	0.144 ± 0.010	-0.011	5
DetNAM	0.211 ± 0.042	0.192 ± 0.035	$+0.019$	54
CBM joint	0.620 ± 0.072	0.634 ± 0.011	-0.014	72
β-VAE+sup	0.150 ± 0.007	0.143 ± 0.011	$+0.007$	56
FactorVAE+sup	0.138 ± 0.015	0.145 ± 0.006	-0.007	36

ConceptNAM's optimal val-stop is at epoch 5 (median across seeds), confirming the posterior-collapse observation in §U: continued training drifts the latent representation toward the prior, hurting concept recovery. DetNAM and CBM, by contrast, benefit from the full 50+ epoch budget, which is consistent with their lack of a KL-to-prior regulariser.

J NAM (AGARWAL 2021) TASK-PERFORMANCE COMPARISON

To address the reviewer ask "did you compare against a real NAM?", we ran the canonical Neural Additive Model (NAM) (Agarwal et al., 2021) on the synthetic benchmark using the same per-feature ExU encoder and fixed-W protocol as the body methods. NAM has no concept-indexed attribution and so cannot be scored on NMI/MIG/DCI/ACS; we therefore compare on task RMSE only. Source code, driver, and JSON artefact paths are in App. W.

Table 27: Task RMSE on the synthetic benchmark (5 seeds, fixed-W). NAM and DetNAM share the same per-feature MLP encoder; DetNAM adds a deterministic $K=4$ concept bottleneck (Level-1, NAM-family), ConceptNAM adds a stochastic VIB bottleneck (Level-1, VAE-family on this axis), and CBM uses concept labels (Level-3).

Method	Family / Level	RMSE (↓)	Params
NAM (no concepts)	NAM / L1 (task-only)	0.508 ± 0.017	68 641
DetNAM (4 concepts)	NAM / L1	$\mathbf{0.478 \pm 0.031}$	81 049
ConceptNAM (4 concepts, VIB)	NAM∩VAE / L1	0.814 ± 0.050	91 869
CBM-joint (4 concepts)	L3	$\mathbf{0.165 \pm 0.006}$	44 677

The NAM-family Level-1 bottleneck (DetNAM) gives a small RMSE improvement over NAM proper while exposing concept-indexed interpretability surfaces. The VIB stack on top (ConceptNAM) *degrades* task RMSE by $\sim 60\%$ relative to NAM and $\sim 70\%$ relative to DetNAM — consistent with

the §5.4 negative finding that the VIB is a net cost on tabular concept discovery, now extended to task accuracy.

K PROTOCOL-VS-ARCHITECTURE ABLATION

To make the contributions of the matched-supervision *protocol* and the iCBM *architecture* separately legible, we extract a 2×2 ablation from the iCBM semi-supervised sweep (same fixed-W Synthetic-Concept generator, $K=4$, 5 seeds; see `results/exp_protocol_vs_arch.json`). Cell (A) tests "architecture invisible to NMI" at the task-only end of the supervision spectrum; Cell (B) tests the same claim at the concept-supervised end; Cell (C) reports the per-method spread under the unmatched-protocol comparison that the published literature typically uses (CBM with concept labels vs. Level-1 methods without). The contrast across cells is the empirical statement of our paper.

Table 28: Protocol-vs-architecture 2×2 on Synthetic-Concept $K=4$ (NMI mean $\pm$ std across 5 fixed-W seeds, derived from `results/exp_icbm_semisup.json`). Row spread is $\max-\min$ over the architectures within a row.

	Cell	Architectures (NMI)	Spread
(A)	Matched task-only sup ($r=0$)	iCBM 0.202, DetNAM 0.206, ConceptNAM 0.111	0.096
	NAM-family only:	iCBM 0.202 ± 0.033 vs DetNAM 0.206 ± 0.041	0.004
(B)	Matched full concept sup ($r=1$)	iCBM 0.467, CBM 0.696, ConceptNAM 0.239	0.457
(C)	Unmatched native protocol	ConceptNAM 0.111, iCBM/DetNAM $\approx$0.20, CBM 0.696	**0.585**

Reading the table. Cell (A)'s NAM-family row gives spread 0.004 (the per-method scaling curves behind it are in Fig. 1), and it is not evidence: it compares iCBM with DetNAM at $r=0$, where the two are literally the same model (App. M), so it measures seed noise. The architecture spread that this cell does support is ConceptNAM vs. DetNAM, 0.096. This row does not license the "architecture invisible to NMI" finding. ConceptNAM sits outside the band because the VIB triggers the posterior collapse documented in §U; this is a within-Level-1 *architectural* effect that the matched protocol does not erase. Cell (B) shows iCBM and CBM differ by 0.229 NMI even at full concept supervision, which is the architectural lever the matched protocol *does* expose (iCBM trades raw NMI for the partial-supervision robustness shown in Table 6; the clinical variance gap of App. N is a point estimate, not an established difference). Cell (C) reports the per-method spread 0.585 that the unmatched literature protocol produces. Subtracting Cell (A)'s 0.096 from it does not measure how much of the apparent inter-architecture gap is supervision: Cell (C) includes a method given concept labels and Cell (A) does not, so the difference is close to a restatement of that fact rather than a measurement; and Cell (C) is assembled from the same runs as (A) and (B) (`derived_from` in `exp_protocol_vs_arch.json`) rather than being an independent unmatched experiment. What the cell does show, without arithmetic, is that the residual within-Level-1 spread under the matched protocol is 0.096 — roughly 70% of DetNAM's entire excess over this cell's own null ($0.206-0.072=0.134$) — i.e. the matched protocol does *not* erase architecture inside Level 1.

L ARCHITECTURE AND MULTI-METRIC FIGURES

The architecture schematic and the multi-metric divergence figure are placed here rather than in the body: ConceptNAM is a case-study model for the benchmark, not a claimed architectural contribution (§5), and the divergence figure restates Table 38 and Table 4.

M CROSS-DOMAIN VALIDATION TABLES

ICBM: A SUPERVISION-AGNOSTIC LEVEL-1 PACKAGING

iCBM. DetNAM beats ConceptNAM at every label ratio (S1) but cannot consume concept labels; CBM has the opposite limitation. iCBM (`baselines/icbm.py`) does both: DetNAM encoder + linear K-bottleneck + NAM-additive head + Hungarian-aligned concept loss on the labelled subset. Its ingredients are not new; what the protocol identifies is that it *reduces to* DetNAM at $r=0$ (no

Table 29: **The estimator-bias floor, at the configurations this literature evaluates in.** Chance level of each metric under a representation independent of the factors (*random code*), computed from each benchmark's published factor grid — a Cartesian product of known cardinalities sampled uniformly, so no images are needed and no published number is recomputed. These are the chance levels of the canonical implementations used throughout, a denominator, not a correction to anyone's result. **Every column is the random-code floor, including our own row**, so the comparison is like-for-like; the *operative* floor is larger everywhere and is given in the last column for the two cases where the data to compute it exists. Our own tabular cell, at $K{=}4$ with 240 held-out samples, sits at 0.095 — roughly the size of the entire effect its Level-1 methods produce. Source: `analyze_literature_chance.py`, `analyze_chance_floor.py`.

			bias floor @ $n_{\text{test}}{=}10^3$			bias @ 10^4	operative floor	
Benchmark	K	factor grid	NMI	MIG	DCI-D	NMI	@10^3	@10^4
dSprites (Matthey et al., 2017)	5	$3{\cdot}6{\cdot}40{\cdot}32^2$	0.016	0.002	0.013	0.002	0.057	0.050
3D Shapes (Burgess & Kim, 2018)	6	$10^3{\cdot}8{\cdot}4{\cdot}15$	0.016	0.002	0.013	0.002	—	—
Cars3D (Reed et al., 2015)	3	$4{\cdot}24{\cdot}183$	0.016	0.002	0.009	0.002	—	—
SmallNORB (LeCun et al., 2004)	5	$5^2{\cdot}9{\cdot}18{\cdot}6$	0.014	0.002	0.013	0.001	—	—
MPI3D (Gondal et al., 2019)	7	$6^2{\cdot}2{\cdot}3^2{\cdot}40^2$	0.012	0.002	0.016	0.001	—	—
This paper (240)	4	continuous	**0.095**	0.011	0.014	—	**0.115**	—

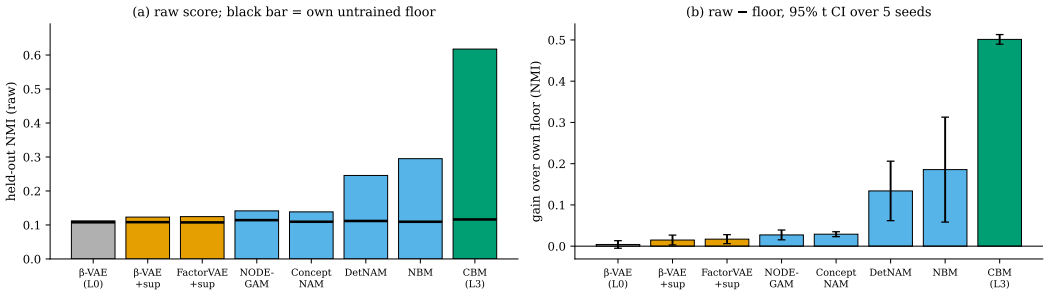


Figure 2: Headline cell on the corrected configuration (synthetic $K{=}4$, 5 seeds, fixed W; `bench_corrected.json`, `arch_matched_null.json`). Three quantities, kept apart because they answer different questions. **(a)** Raw held-out NMI — the quality of the trained representation as the metric sees it — with each method's *own* untrained floor (its architecture-matched null mean) drawn as a black bar inside it; CBM has no matched null, so its bar carries the random-linear one. **(b)** The gain, raw minus own floor, computed per seed and shown with a 95% t interval over the five seeds: what training bought this architecture over its own initial state. The two can disagree: a method whose untrained state already scores well is charged a larger floor, so a ranking by (b) is a ranking of learning gains, not of final quality, and we report both. The floor-clearing tests of App. B.1 are run on (b): six of six task-supervised methods clear under Holm, five under Bonferroni. The only recommendation the paper supports is the supervision-level one.

labels, no alignment term — literally the same model, not a measured tie), accepts concept labels without an architectural switch, and matches CBM at the supervised ceiling on raw $64{\times}64$ dSprites (NMI 0.189 vs 0.194). On clinical data it matches or exceeds CBM in mean AUC with a smaller seed spread, though no procedure valid at five seeds establishes that gap on either dataset, and it is not iCBM-specific. Sweeps: Apps. B, M.1, N.

M.1 dSprites raw-image probe

M.2 External checkpoints: the protocol on independently trained models

To test the procedure on models we did not train, we re-scored the public dSprites checkpoints of an independent PyTorch disentanglement library (MIT licence; repository and commit, and a SHA-256 for every checkpoint, in `external_checkpoints.json`;

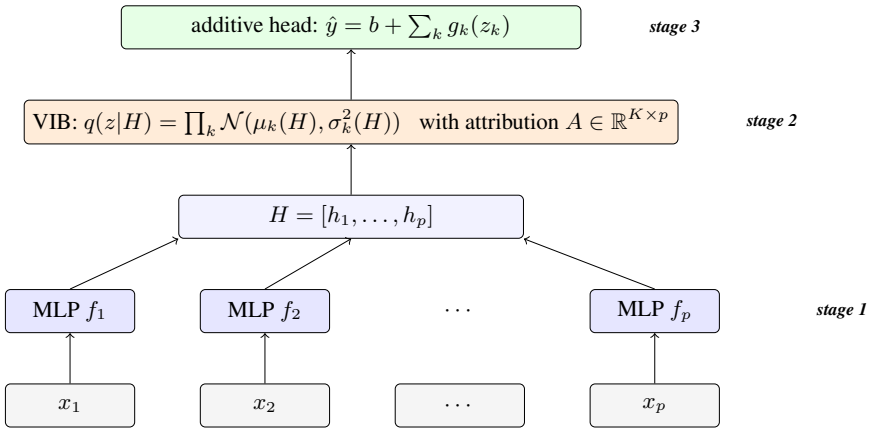


Figure 3: ConceptNAM. Stage 1: per-feature MLPs produce $H \in \mathbb{R}^{p \times d}$. Stage 2: a deep variational information bottleneck with factorized posterior plus a softmax-normalized attribution matrix A. Stage 3: a concept-additive head with per-concept shape functions g_k. Theorem 1 applies to stage 3.

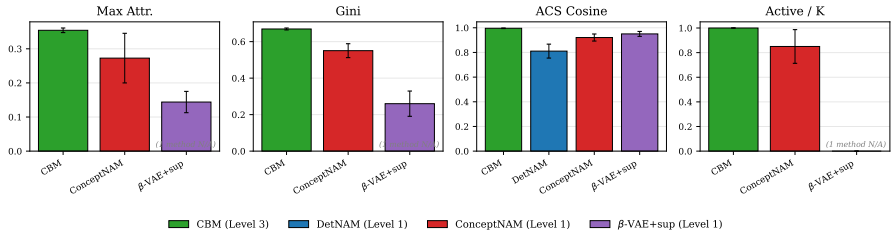


Figure 4: Multi-metric interpretability divergence (synthetic $K{=}4$, regenerated under the audit-corrected protocol). NMI-equivalent methods within the VAE-family Level-1 triplet (Table 1, shipped block: they pass Bonferroni-adjusted TOST, at a margin wider than their excess over chance) are ranked differently by attribution sharpness (Max Attr.), attribution density (Gini), cross-seed attribution stability (ACS, 10 seeds), and posterior-collapse diagnostics (Active / K). A single-metric reading would call them interchangeable; the four-metric reading separates CBM (most stable, most concentrated, full concept supervision) from the Level-1 deterministic NAM-family method (DetNAM, only ACS reportable), from ConceptNAM (concentrated but partially collapsed, $3.4/4$ active units), from β-VAE+sup (complete posterior collapse, zero active units). *Attribution panels are on* `input_attribution`*'s shared input-feature axis, not each method's own row scale: concentration statistics scale with vector length, and on the native scale ConceptNAM outscores CBM only by having an* $8\times$ *shorter row (Table 38).* The ConceptNAM drawn here is the higher-sparsity condition, not the lower-sparsity one whose NMI the TOST above refers to; a same-checkpoint rerun reads every metric off single models at both sparsity levels and reproduces this ordering (App. P). Drawn from `exp8_interpretability.json` and `exp13_acs.json`.

`exp_external_checkpoints.py`). Eight checkpoints load: a VAE, β-VAE (two variants), FactorVAE and four β-TCVAE settings, all sharing one convolutional encoder with 10 latents. Each is scored on a fixed sample of 10,000 images (and its first 1,000) against the five generative factors with our canonical NMI, MIG and DCI-D, using the posterior mean, and against two floors: the random-code null (50 draws) and the *operative* one — the same encoder architecture, untrained, from the library's own initialiser (50 draws). With one checkpoint per model there is no seed variance, so the verdict here is the single-score screen (clearance of the null's 95th percentile), not the Welch rule.

What changes and what does not. Nothing reorders and no verdict flips: every checkpoint clears both floors on every metric at both sample sizes, and the ranking by excess equals the ranking by raw score (Kendall $\tau{=}1$ on each metric). That is the expected consequence of a shared architecture — one common floor shifts every score equally — and it is the honest limit of this check: the correction cannot reorder models that share an untrained floor. What it does change is every *ratio*. The untrained floor at $n{=}10^4$ is 0.027 NMI, 0.014 MIG and 0.148 DCI-D (against a random-code floor of 0.0008

Table 30: dSprites raw-image probe (3 seeds, 64×64, K=5 ground-truth factors; `results/exp_dsprites_canonical.json`, $n = 100$k, 200 epochs, ConvFeatureEncoder (channels 64/128/256), AdamW + cosine LR). **AUDIT B2/B3 fix**: this canonical driver fits PCA/standardisation on the training split only, and selects checkpoints by validation task-MSE (no concept labels in selection). **Read against the operative null, not the code null.** A random-*cluster* baseline scores NMI 0.001, but that is a random code. The operative null must be drawn from the encoder class actually being scored. An untrained linear projection of the pixels gives 0.044 (p_{95} 0.056); the probe's own untrained encoder — `ConvFeatureEncoder` with channels $(64, 128, 256)$ feeding an untrained bottleneck, at this table's own split — scores 0.067, 95th percentile 0.078 (`analyze_image_null_archmatched.py`), half again the linear value. *The entire Level-1 half of this table sits at or below that null*: ConceptNAM 0.042, iCBM-task 0.055 and DetNAM 0.062 are all under 0.067, so neither an ordering among them nor DetNAM's 0.006 margin over the linear null survives the architecture-matched one. What remains is the two concept-supervised rows, which clear the null by a wide margin and match each other (iCBM NMI 0.189 vs CBM 0.194; DCI-D 0.508 vs 0.501), within seed-to-seed std on both metrics. ConceptNAM collapses to a single active concept.

Method	NMI	DCI-D	MIG	Active / 5
CBM joint	$\mathbf{0.194\pm0.008}$	0.501 ± 0.054	$\mathbf{0.212\pm0.017}$	3.7 ± 0.6
iCBM (concept-sup)	0.189 ± 0.003	$\mathbf{0.508\pm0.067}$	0.199 ± 0.020	3.3 ± 0.6
DetNAM	0.062 ± 0.004	0.144 ± 0.019	0.022 ± 0.004	4.7 ± 0.6
iCBM (task-only)	0.055 ± 0.004	0.181 ± 0.021	0.019 ± 0.005	4.7 ± 0.6
ConceptNAM	0.042 ± 0.009	0.130 ± 0.019	0.007 ± 0.003	1.0 ± 0.0
untrained ConvNet (matched null)	0.067 (p_{95} 0.078)	—	—	—
untrained linear	0.044 (p_{95} 0.056)	—	—	—

NMI), so DCI-D is where it bites: the capacity-annealed β-VAE checkpoint's DCI-D is $1.86\times$ the plain VAE's in raw terms and $2.6\times$ as excess; for the strongest β-TCVAE setting the raw $2.12\times$ becomes $3.1\times$. The evidence strength moves the other way: against the random-code null the plain VAE's DCI-D sits 56 null standard deviations above chance, against its own untrained encoder 3.9; the untrained floor is 46% of its raw DCI-D. A literature that reports "X improves disentanglement by $k\times$" over a baseline of the same architecture understates k when the floor is ignored; across architectures, where floors differ (App. M), the correction can also reorder. We did not find a second public checkpoint set with a different architecture on the same data to test the second case.

N Clinical tabular probe

O Per-dataset real-world results

Dataset-selection rationale. The selection is not inherited from prior work: Koh et al. (2020) uses OAI and CUB-200 with human-annotated concept annotations, and neither UCI Heart nor Pima appears in it; Agarwal et al. (2021) and Radenovic et al. (2022) use California Housing but not Adult, and Chang et al. (2022) uses Adult but not California Housing — no one of the three uses both.

These are our own choices, made before seeing results, on two grounds a reviewer can check: each is a widely-used public tabular set with a documented feature semantics, and together they span regression and binary classification. That is weaker than a canonicity appeal and we do not dress it up as one. A reviewer who points to "what about German Credit / Wine / Boston Housing?" would correctly observe that those datasets are not canonical for either of the two literatures we are comparing to. We restrict to publicly cached datasets so the artefact reproduces without external data-access requirements (cf. Heart and Pima caches under `data/heart/` and `data/pima/`; California and Adult load via `sklearn`).

What the numbers mean. Values computed by the per-dataset real-world driver with the first K input features as concept proxies (see §B.5 caveat). "NMI" is concept–proxy alignment, not a concept-recovery number. Because those proxies are inputs the encoder reads directly, the chance

Table 31: dSprites tabular probe (PCA-32, 5 seeds, K=5; `results/exp_dsprites_tabular.json`). Same hierarchy as the image probe: CBM (Level-3) dominates by a large margin; NAM has no concept output and is excluded from the disentanglement columns. **Read as excess over the null row**: the operative floor here is 0.031 (p_{95} 0.038; `analyze_dsprites_null.py`), so the four Level-1 entries clear it by 0.009–0.025 while CBM clears it by 0.220, both measured against that same 95th percentile. **Architecture-matched nulls for this probe, measured.** That 95th percentile is a random *linear* encoder's, and on the image probe the matched null is $1.5\times$ the linear one — so we measured the matched nulls here too, with the same untrained architectures on this probe's own split (`dsprites_tabular_matched_null.json`, against the linear null of `dsprites_null.json`; 12 draws per architecture, coarser than our tabular nulls because each draw is a forward pass over 9000 rows). The verdicts hold: all four Level-1 entries clear their own matched p_{95}. But the *sign* of the matched-versus-linear gap varies by architecture inside one domain, not only between domains as §3 said: three matched nulls come in below the linear one (0.73–$0.79\times$) while NODE-GAM's comes in *above* it ($1.18\times$), and it is NODE-GAM whose margin is thinnest — 0.0596 against a matched p_{95} of 0.0575. Read that row as resolved but not comfortably. The driver reduces every sampled image to 32 principal components and z-scores them, fitting both on the training split alone (`make_dsprites_splits`, pinned by `test_make_dsprites_splits_fits_pca_on_train_only`), as the protocol requires (§4).

Method	NMI	DCI-D
CBM joint	$\mathbf{0.258 \pm 0.028}$	$\mathbf{0.715 \pm 0.046}$
ConceptNAM (Level-1)	0.063 ± 0.009	0.185 ± 0.038
DetNAM (Level-1)	0.062 ± 0.014	0.141 ± 0.048
NODE-GAM (Level-1)	0.060 ± 0.005	0.141 ± 0.047
NBM (Level-1)	0.047 ± 0.011	0.102 ± 0.048
NAM (Level-1, no concept output)	—	—
untrained encoder (null)	0.031 (p_{95} 0.038)	—

level here is very high (0.111–0.277, against 0.115 for our synthetic cell at a far smaller n_{test}), and we report it per dataset. Read against it, this table supports one claim — that concept supervision helps where it helps — and no Level-1 ranking at all.

On three of the four datasets CBM clears its null by a wide margin when the proxy signal is used as full supervision; on Diabetes it does not clear at all. The Level-1 methods are indistinguishable from an untrained encoder everywhere except ConceptNAM on Adult Income. With the null measured, what looks like a band among them is the floor, and these numbers are an interpretability sanity check rather than a concept-recovery result.

P ConceptNAM configurations

P.1 Per-driver configuration table

The two operating points that change across our headline experiments are $\lambda_{\text{sparse}} \in \{0.01, 0.10\}$ (raised when the metric is attribution sparsity, kept low when the metric is NMI / DCI on the posterior) and the alignment loss (used only when concept labels are available; MSE in legacy drivers, Hungarian in S1 to make the alignment permutation-invariant under Theorem 1). The full per-driver override table follows below.

Different headline experiments emphasise different metrics, so the ConceptNAM hyper-parameters are set per-experiment to make each metric addressable on its own terms. **These are separate experimental conditions, not views of one model**: ConceptNAM's interpretability and ACS numbers come from λ_{sparse}=0.10 fits, its NMI, MIG and DCI from λ_{sparse}=0.01 fits. **Same-checkpoint check.** To test whether the two profiles co-occur in one model, `exp_same_checkpoint_metrics.py` (`same_checkpoint_metrics.json`) trains every method once per seed and reads *all* metrics off that single checkpoint (the five headline seeds for per-checkpoint metrics, ten for ACS, attributions in evaluation mode, one 50-epoch recipe for every metric including ACS; FactorVAE+sup is not

Table 32: Clinical-domain probe AUC (5 seeds, BCE-trained binary head; data in `results/exp_clinical.json`). **Important interpretation note.** UCI Heart and Pima Diabetes have no ground-truth concept annotations, so the "concept supervision" used here is a first-K-feature proxy with $K{=}4$ — our design choice, not the protocol of Koh et al. (2020), whose concepts are human annotations — and not a true hidden-concept target. This table therefore reports *task AUC* (the classification target) and *proxy-bottleneck stability* across seeds — it is **not** evidence of hidden-concept recovery, and the cross-domain claim of §M is read as "the iCBM architecture transfers to real tabular domains where the proxy-supervision protocol of CBM also holds" rather than as concept-discovery validation. iCBM holds a small mean-AUC lead on UCI Heart and is tightly clustered on both datasets; ConceptNAM has reasonable mean AUC and a larger observed seed-to-seed spread (point estimate $5.6\times$ iCBM's std on UCI Heart), which is *not* established as a variance difference: the robust Brown–Forsythe test rejects on neither dataset and the conservative independent bootstrap interval includes 1 (below, App. D.3). If real, it would fit the VIB-collapse account of §V; at five seeds we cannot say it is. **Paired statistical tests (5 seeds):** iCBM vs CBM mean AUC is not significantly different on either dataset (paired t, Heart $p{=}0.124$, Pima $p{=}0.125$). iCBM significantly beats NAM on Heart ($p = 0.028$) but loses to NAM on Pima ($p = 0.002$): mean-AUC dominance is dataset-dependent. The variance gap iCBM vs ConceptNAM AUC std is significant under Bartlett ($p{=}0.006$, $p{=}0.0003$) and Levene ($p{=}0.003$, $p{=}0.042$) but *not* under the median-centred Brown–Forsythe test ($p{=}0.12$, $p{=}0.36$), which is the robust choice at $n{=}5$; a 10,000-replicate bootstrap CI on the std-ratio (App. D.3) gives Heart $5.6\times$ with a paired CI95 of [3.04, 16.49] — an interval that is not calibrated (reusing one resample for both methods makes it anti-conservative, excluding 1 in 9% of equal-variance simulations); the conservative independent CI includes 1, so the ratio is a direction, not an established reduction.

Dataset	iCBM	CBM	NAM	DetNAM	ConceptNAM
UCI Heart	$\mathbf{0.907{\pm}0.02}$	$0.886{\pm}0.05$	$0.883{\pm}0.04$	$0.861{\pm}0.03$	$0.787{\pm}0.13$
Pima Diabetes	$0.792{\pm}0.01$	$\mathbf{0.809{\pm}0.01}$	$0.803{\pm}0.02$	$0.782{\pm}0.01$	$0.684{\pm}0.19$

trained here, so the triplet comparison is against β-VAE+sup only). Each profile is tested per configuration with seed-paired differences against β-VAE+sup on the same checkpoints (`-recompare` rewrites this block from the stored per-seed values). NMI: ConceptNAM minus β-VAE+sup is -0.004 at both sparsity levels, 95% CI $[-0.026, +0.018]$ ($\lambda_{\text{sparse}}{=}0.01$) and $[-0.027, +0.019]$ (0.10) — inside the paper's ±0.05 TOST margin, *not* inside ±0.015, the corrected block's between-method range. Attribution: ConceptNAM's Gini exceeds β-VAE+sup's by 0.23 $[0.16, 0.30]$ and 0.25 $[0.18, 0.32]$, with $3.6/4$ active units against $0/4$. So on single checkpoints the two profiles co-occur at the loose margin and cannot be claimed at the tight one; five seeds do not resolve NMI differences of that size. The sparsity level changes little (Gini $0.505 \to 0.519$). CBM is the most stable across seeds (ACS-cosine 0.99) and most concentrated by Gini (0.68) on the same checkpoints. *The driver's first comparison declared the profiles absent from one configuration because the two argmins over configurations differed by 10^{-4} in NMI; that logic was invalid and has been replaced.*

Q ConceptNAM loss decomposition and toggle semantics

The training loss is

$$L \;=\; L_{\text{task}} \;+\; w(t)\big[\alpha\,\widehat{I}(Z;H) + \beta\,\widehat{\mathrm{TC}}(Z) + \gamma\textstyle\sum_k \mathbb{E}_x\,\mathrm{KL}[q(z_k\,|\,x)\|\mathcal{N}(0,1)]\big] \;+\; \lambda\,L_{\text{sparse}},$$

where $w(t)$ is a linear warm-up over the first 10 epochs (weight 0 at $t{=}0$) and $\widehat{I}(Z;H) = \mathbb{E}_x\mathrm{KL}[q(z\,|\,x)\|\mathcal{N}(0,I)]$ is the usual VIB upper bound. Note that the γ term is the *conditional* KL averaged over data, not $\mathrm{KL}[q(z_k)\|\mathcal{N}(0,1)]$ on the aggregate posterior that the β-TCVAE decomposition names — the two are different objectives, and this is the one we train (App. Z). α and γ multiply identical scalars (sum / mean commute over disjoint axes), so on the default path the effective loss is a 2-term decomposition (KL + TC) with KL weight $\alpha + \gamma$. The two factors are kept distinct only so the optional free-bits routing can replace the γ branch with a per-dimension clamp; the IB toggle disables both branches together (Audit C1).

Table 33: Tabular real-world results (3 seeds). **Read every Proxy-NMI against the null row, and most of this table does not clear it.** The "concepts" are the first K input features, which the encoder consumes directly, so an untrained projection of x already predicts them well — the null runs 0.111–0.277, far above any synthetic cell. Against it, the Level-1 methods clear on *one* of eight method–dataset pairs (ConceptNAM on Adult Income, $+0.060$); the rest sit at or below chance, and on Diabetes ***CBM itself*** is below its own null, so that dataset ranks nothing. A band among sub-chance scores is not a band. What the table supports is CBM's margin on the three datasets where it clears. Methods: `exp7_realworld.json`; nulls: `realworld_null.json`.

Dataset	Method	Task metric	Proxy-NMI	Proxy-MIG
Breast Cancer (acc)	ConceptNAM	0.626 ± 0.035	0.302 ± 0.018	0.011 ± 0.007
	β-VAE+sup	0.374 ± 0.035	0.284 ± 0.038	0.025 ± 0.015
	CBM	$\mathbf{0.953 \pm 0.013}$	$\mathbf{0.740 \pm 0.033}$	$\mathbf{0.261 \pm 0.025}$
	untrained encoder (null)	—	*0.277* (p_{95} 0.304)	—
Diabetes (R^2)	ConceptNAM	0.022 ± 0.018	0.259 ± 0.013	0.035 ± 0.031
	β-VAE+sup	-0.011 ± 0.014	0.262 ± 0.009	0.033 ± 0.001
	CBM	$\mathbf{0.455 \pm 0.057}$	0.240 ± 0.011	0.003 ± 0.002
	untrained encoder (null)	—	*0.257* (p_{95} 0.274)	—
California (R^2)	ConceptNAM	0.301 ± 0.027	0.089 ± 0.001	0.027 ± 0.008
	β-VAE+sup	0.000 ± 0.000	0.056 ± 0.053	0.003 ± 0.006
	CBM	$\mathbf{0.801 \pm 0.009}$	$\mathbf{0.559 \pm 0.007}$	$\mathbf{0.466 \pm 0.016}$
	untrained encoder (null)	—	*0.111* (p_{95} 0.142)	—
Adult Income (acc)	ConceptNAM	0.753 ± 0.005	0.188 ± 0.046	0.124 ± 0.051
	β-VAE+sup	0.753 ± 0.005	0.000 ± 0.000	0.000 ± 0.000
	CBM	$\mathbf{0.827 \pm 0.006}$	$\mathbf{0.569 \pm 0.006}$	$\mathbf{0.791 \pm 0.011}$
	untrained encoder (null)	—	*0.127* (p_{95} 0.166)	—

Table 34: ConceptNAM hyper-parameter overrides per experiment driver. Defaults: α=0.001, β=0.1, γ=1.0, d_{embed}=8, bottleneck hidden 256, batch 64, Adam lr 10^{-3}. The varying knobs are below.

Driver	Cited in	N_EPOCHS	warmup	λ_{sparse}	alignment
Headline (Tab. 1)	supervision hierarchy	50	10	0.01	none
Interpretability (Tab. 38)	multi-metric divergence	50	10	0.10	none
Semi-sup (Tab. 5)	label-rate sweep	30	10	0.01	MSE on labelled
Ablations (Tab. 8)	component sweep	40	10	0.01	none
ACS (Tab. 4)	cross-seed stability	40	10	0.10	none
Low-label (§B.5 (i))	very low label rates	40	10	0.01	MSE on labelled
S1 sweep (Tab. 25, §5.4)	supervision sweep	50	10	0.01	Hungarian on labelled

R VIB ABLATION: FULL SWEEPS

DIRECT VIB ABLATION: THREE CONTROLLED EXPERIMENTS

§5.1 leaves ConceptNAM indistinguishable from β-VAE+sup and FactorVAE+sup, with the NAM-family methods well above. To test whether the VIB does useful work *within* the NAM family we ran three controlled experiments under the corrected fixed-W protocol, 5 seeds each — S1 zero concept labels, S2 K-mismatch, S3 high-p stability, exactly the regimes a VIB advocate would predict ConceptNAM should win. None do. Table 25 gives the most-damning operating point per experiment; the pre-specified verdict script reports 0 significant ConceptNAM wins across S1+S2 and 0 across S3.

Reading. (S1) With zero concept labels — task labels only, i.e. Level 1, not unsupervised — the regime where the VIB is supposed to be the only inductive bias driving concept recovery — DetNAM, with no VIB on the same per-feature encoder, recovers more concepts (0.226 vs 0.107 raw, i.e. 0.154 vs 0.036 as excess over this cell's null of 0.072 (`sblock_nulls.json`) — a 4.3× ratio. The denominator is 0.036 against a null whose draw-to-draw sd is 0.012, so treat the ratio as coarse; paired t p=0.015, d_z= − 1.83, matching the pre-specified verdict script). (S2) At K_{model}=32 ConceptNAM

keeps $30.8/32$ active dimensions and CBM $32.0/32$: under a fair comparator *no* method here prunes, and the VIB in particular does not (App. R). (S3) At $p \geq 64$ the VIB collapses ConceptNAM's posterior to a single active dimension (5/5 seeds in both synthetic blocks), and on real $p=30$ data the variance, not just the mean, separates ConceptNAM (0.78 ± 0.36) from DetNAM (0.99 ± 0.00): the VIB makes training itself unstable.

Under fixed W the controlled S1 sweep shows no low-label-rate regime in which ConceptNAM helps. Within the NAM family DetNAM and NBM were the strongest observed, at the configuration of App. G rather than the one shipped here; we recommend neither (Limitation 9).

Table 25 reports the single most-damning operating point per experiment. The full per-condition sweeps follow.

Table 35: S1 supervision sweep on synthetic ($K=4$, $p=16$, fixed W, 5 seeds). Mean NMI on held-out test, $\delta_{\text{TOST}}=0.05$.

r	ConceptNAM-unsup	ConceptNAM-align	CBM	DetNAM (no VIB)
0.0	0.107	0.109	0.132	**0.226**
0.1	0.107	0.088	**0.592**	0.226
0.25	0.107	0.104	**0.649**	0.226
0.5	0.107	0.165	**0.674**	0.226
1.0	0.107	0.248	**0.686**	0.226

Table 36: S2 K-mismatch ($K_{\text{true}}=4$, $p=16$, fixed W, 5 seeds). "active" counts z-dimensions with $\sigma > 0.05\,\sigma_{\max}$. **The NMI column cannot be read down the page.** (i) The CBM comparator, `cbm_masked`, gives no concept supervision to the surplus dimensions (the legacy `cbm_zero_padded` condition supervises them *toward zero*, and the artefact's own audit block records that it must not be read as evidence of unsupervised sparsity): CBM keeps *all* K_{model} dimensions active. **No method in this table prunes.** (ii) NMI is not comparable across rows. Its normaliser scales with the number of *predicted* dimensions, so a model that recovers all four true concepts perfectly still scores $4/K_{\text{model}}$ once the surplus dimensions carry any variance at all — even 10^{-12}. Every NMI in this column is therefore diluted by roughly $4/K_{\text{model}}$, and the apparent decline is the metric's scale shrinking, not concept quality falling. Comparisons *within* a row are unaffected, since the dilution is common. We report the raw values rather than a rescaled column because the right fix is a normaliser that does not reward dead dimensions, which we do not have. The floor moves the same way, and we measured it: an untrained encoder at $K_{\text{model}}=8/16/32$ scores NMI $0.045/0.026/0.015$ on this cell, against 0.072 at $K_{\text{model}}=4$ (`sblock_nulls.json`). A log-log fit over the four points gives an exponent near -0.75 (95% CI $[-0.84, -0.66]$ on two residual degrees of freedom — four points cannot establish a power-law form, only rule out -1), and the difference is not cosmetic: a pure $1/K_{\text{model}}$ law extrapolated from $K_{\text{model}}=4$ predicts 0.0089 at $K_{\text{model}}=32$ against a measured 0.015, understating the floor by 69% — the direction that makes clearing look easier than it is. Read each row against its own K_{model}'s measured floor, never the $K=4$ one and never an extrapolation.

K_{model}	ConceptNAM (active / NMI / DCI-D)	DetNAM (active / NMI / DCI-D)	CBM (active / NMI / DCI-D)
4	4.0 / 0.10 / 0.09	4.0 / 0.21 / 0.32	4.0 / **0.71** / **0.98**
8	7.6 / 0.06 / 0.17	8.0 / 0.15 / 0.59	8.0 / **0.39** / **1.00**
16	15.8 / 0.03 / 0.27	16.0 / 0.07 / 0.65	16.0 / **0.19** / **1.00**
32	30.8 / 0.02 / 0.31	32.0 / 0.04 / 0.73	32.0 / **0.10** / **1.00**

S IDENTIFIABILITY THEOREMS: FULL PROOFS

STATEMENTS, IN FULL

Assumption 1. *Each shape function g_k is C^2, non-affine, with $g_k''(z) \neq 0$ on some open subset of $\mathbb{R}$, and the second derivatives at the origin $\{g_1''(0), \ldots, g_K''(0)\}$ are pairwise distinct.* (Distinctness is imposed at the fixed point the argument uses; it cannot be imposed on an open set and extended by

Table 37: S3 high-dimensional collapse (5 seeds; fixed-W within each block). **Nulls, measured at each block's own cell** (`sblock_nulls.json`): 0.031 at (p=64, K=8) and 0.023 at (p=128, K=16). Both Level-1 entries clear their null, DetNAM by 1.7× the p_{95}, so the finding is small-but-real Level-1 excess under high dimension, not disappearance. Data: `exp_s3_high_dim.json`.

Block	ConceptNAM	DetNAM (no VIB)	CBM
Synth p=64, K=8 NMI / active	0.05 / **1.0**	0.10 / 8.0	**0.55** / 8.0
Synth p=128, K=16 NMI / active	0.03 / **1.0**	0.04 / 16.0	**0.46** / 16.0
Breast cancer AUC (real, p=30)	0.78 ± 0.36	$\mathbf{0.99 \pm 0.00}$	—

analytic continuation, since C^2 is not analyticity. Any other common fixed point of the transformations considered may replace 0.)

Definition 1 (Signed permutation group)**.** $G_{\pm} := \{DP : P \in S_K,\ D = \mathrm{diag}(d_1, \ldots, d_K),\ d_k \in \{\pm 1\}\}$*, the hyperoctahedral group of order* $2^K K!$.

Corollary 1. *If, in addition, no g_k is even on any neighbourhood of the origin (for every $\varepsilon > 0$ some $|z_k| < \varepsilon$ has $g_k(z_k) \neq g_k(-z_k)$), then the sign matrices of Theorem 1 reduce to the identity.* The hypothesis is local because the theorem's invariance is: a function even near 0 but asymmetric far from it keeps the local sign flip. This hypothesis fails on our own shape bank for every $K \geq 2$: $z^2 - 1$ is even, as is e^{-z^2} at $K \geq 5$. The corollary is therefore never active in this paper, and the residual ambiguity our generator actually exhibits is a sign flip on the even coordinate, not a permutation.

Theorem 1 (Symmetries of a Fixed Additive Head)**.** *Let $F(z) = \sum_k g_k(z_k)$ under Assumption 1. If $R \in \mathrm{O}(K)$ satisfies $F(Rz) = F(z)$ for all z in an open neighbourhood $V \subseteq \mathbb{R}^K$ of the origin, then $R \in G_{\pm}$; in fact R is a diagonal sign matrix, since with pairwise-distinct $g_k''(0)$ no non-trivial permutation commutes with $\mathrm{diag}(g_k''(0))$. The statement is local: it constrains R from invariance near the origin and makes no claim about non-linear symmetries of F elsewhere. It is a statement about the symmetries of a* fixed *head F, not an identifiability theorem for a model whose encoder and head are learned together (App. S, "What the hierarchy does not claim").*

Proof. The Hessian of F is $\nabla^2 F(z) = \mathrm{diag}(g_1''(z_1), \ldots, g_K''(z_K))$. Invariance on V gives $R^\top \nabla^2 F(Rz) R = \nabla^2 F(z)$ for all $z \in V$. Evaluating at the fixed point z=0 (where $Rz = z$) yields $R^\top D R = D$ with $D = \mathrm{diag}(g_k''(0))$, i.e. R commutes with D; by Assumption 1 the entries of D are pairwise distinct, so the commutant consists of diagonal matrices only, and an orthogonal diagonal matrix is a sign matrix. Hence $R \in G_{\pm}$. Away from a fixed point the identity relates $\nabla^2 F(Rz)$ to $\nabla^2 F(z)$ — two *different* diagonal matrices — so the commutant step does not apply there. This is why the assumption is imposed at 0: an extension off a neighbourhood by analytic continuation is unavailable, since C^2 is not analyticity. □

Remark 2 (Withdrawn: a sparsity-pinned permutation)**.** *Earlier versions stated as a theorem that, with pairwise-distinct (later: disjoint) ground-truth attribution supports and a task that "depends on every concept", the entropy penalty $\sum_k H(A_{k,:})$ on the row-softmaxed A forces, as $\lambda_{sparse} \to \infty$, a bijective assignment of rows to supports.* ***That statement is false, and we withdraw it.*** *Two counterexamples, both from reviewers. (i) Independent Rademacher x_1, x_2, x_3, $c_1 = x_1 + x_2$, $c_2 = x_3$, $y = c_1 + 0.1c_2$: supports $\{1,2\}, \{3\}$ are disjoint and both concepts matter, yet two Diracs on x_1, x_2 reach squared error 0.01 and every bijective placement reaches 1. (ii) Adding single-feature recoverability and a Bayes-risk ordering of covering assignments does not rescue it for the model it is about: with $c_1 = x_1$, $c_2 = x_2$, $y = c_1 c_2$, the covering assignment has Bayes risk 0, yet the best* additive *head over $\{x_1, x_2\}$ and over $\{x_1, x_3\}$ both have risk 1, so non-covering minimisers exist. Restating the hypothesis on the head class's empirical risk makes the result assume its conclusion. We also previously said our generator satisfies the disjointness hypothesis; it does not — `cdnam/data/synthetic.py` draws each row's three features independently across rows, and at `dataset_seed`= 0, K=4, p=16 the supports are $\{8, 9, 11\}$, $\{0, 7, 9\}$, $\{0, 6, 9\}$, $\{0, 8, 14\}$, every pair intersecting. No experiment in this paper used the statement.*

Theorem 3 (Sample Complexity for Identity Recovery)**.** *Let $c_i^* \in \mathbb{R}^K$ be i.i.d. samples from the training distribution P_x with $\|c^*\|_2 \leq B$, and let $\Delta = \min_{T \in G_{\pm},\, T \neq \mathrm{id}} \mathbb{E}\|c^* - T(c^*)\|_2^2 > 0$. Assume the trained encoder achieves $\|z_i - c_i^*\|_2 \leq \epsilon$ for all $i \in L$ with $\epsilon \leq \sqrt{\Delta}/4$. Then under Theorem 1's*

restriction $R \in G_{\pm}$, with $|L|$ labelled samples satisfying $|L| \geq 32B^4 \log\big(2(2^K K! - 1)/\delta\big)/\Delta^2$, the empirical minimiser of the alignment loss $\frac{1}{|L|}\sum_{i\in L} \|z_i - c_i^*\|_2^2$ realises $T = \mathrm{id}$ with probability at least $1 - \delta$.

The union bound. *The ambiguity Theorem 1 leaves is $G_{\pm}$, the signed permutations, of which there are $2^K K!$ — so the bound unions over $2^K K! - 1$ competitors and Δ is the minimum over all of them, not over the $K! - 1$ sign-free ones. Relative to a permutation-only bound this costs $K \log 2$ inside the logarithm (at K=4 the count goes $23 \to 383$, so $|L|$ grows by a factor $\log(2{\cdot}383)/\log(2{\cdot}23) \approx 1.7$), and it can only* lower Δ*, since the minimum is over a superset. A permutation-only bound would not be conservative: it would certify recovery of π while leaving the sign flips — the ambiguity our own generator actually exhibits (Cor. 1) — outside the guarantee.*

Constants. *For each $T \neq \mathrm{id}$ the variable $\|c^* - T(c^*)\|_2^2$ lies in $[0, 4B^2]$, so Hoeffding gives $\Pr(|\widehat{D}_T - \mathbb{E}D_T| \geq t) \leq 2\exp(-|L|t^2/8B^4)$; a union bound over the $2^K K! - 1$ competitors at $t = \Delta/2$ yields the stated $|L|$.*

This appendix states and proves Theorems 1 and 3 — which the main text cites but, for space, does not restate — and adds two ancillary theorems (causal additivity of the NAM head; nonlinear-ICA connection). `THEOREMS.md` in the supplementary archive is a superseded Markdown draft kept for provenance; this appendix is the authoritative version.

Setup

$x \in \mathbb{R}^p$ observed features; $z \in \mathbb{R}^K$ latent concepts; encoder $q_\phi(z \mid x)$; task head $f_\theta : z \mapsto \hat{y}$. For loss $\mathcal{L}$ define $\mathrm{Equiv}(\mathcal{L}) = \{T : \mathcal{L} \text{ invariant under } z \to T(z)\}$. We say M is *identifiable up to* $G \subseteq GL(K)$ if $\mathrm{Equiv}(\mathcal{L}_M) = G$. Locatello et al. (2019) show $\mathrm{Equiv}(\mathcal{L}_{\mathrm{ELBO}}) \supseteq O(K)$ for unsupervised ELBO with rotationally symmetric prior — this applies to vanilla β-VAE / FactorVAE.

Theorem 1 (Additive Identifiability)

Proof. Let $F(z) = \sum_k g_k(z_k)$. Invariance $F(Rz) = F(z)$ on an open neighbourhood of 0 implies, on Hessians, $R^\top \nabla^2 F(Rz) R = \nabla^2 F(z)$ there, with $\nabla^2 F(z) = \mathrm{diag}(g_1''(z_1), \ldots, g_K''(z_K))$. Evaluate at the fixed point $z = 0$, where $Rz = z$ and both sides use the same matrix: $R^\top D R = D$ with $D = \mathrm{diag}(g_k''(0))$, i.e. $DR = RD$. By Assumption 1 the entries of D are pairwise distinct, so its commutant in $\mathbb{R}^{K\times K}$ is the set of diagonal matrices; an orthogonal diagonal matrix has entries ± 1. Hence $R = \mathrm{diag}(\pm 1) \subseteq G_{\pm}$. □

Does the assumption hold on our own generator? No, and we say so rather than leave it to be discovered. The benchmark's shape bank (`cdnam/data/synthetic.py`) is $\{\sin,\ z^2{-}1,\ 0.5z^3,\ \tanh,\ e^{-z^2},\ \mathrm{clip}\}$, assigned to concepts as $g_k = \mathrm{bank}[k \bmod 6]$. Their second derivatives at the origin are $\{0,\ 2,\ 0,\ 0,\ -2,\ 0\}$, so at K=4 we have $D = \mathrm{diag}(0, 2, 0, 0)$: the eigenvalue 0 has multiplicity three, the commutant is $O(3) \oplus O(1)$ rather than the diagonal matrices, and Theorem 1 reduces $O(4)$ only to a 3-parameter group. The assumption does hold at K=2 (bank entries $\{0, 2\}$), and $K \geq 6$ additionally violates the C^2 requirement through clip. The theorem therefore constrains the model class but is *not* active on the data our experiments use; it should be read as a statement about what the additive head can be identified up to in principle, not as an explanation of any measured number. A generator with pairwise-distinct $g_k''(0)$ would make it testable, and we regard that as the natural next experiment for the theory rather than a claim we can make now.

Scope. The conclusion is driven entirely by the fixed point: at $z \neq 0$ the identity relates $\nabla^2 F(Rz)$ and $\nabla^2 F(z)$, which are different diagonal matrices, and no commutant argument is available. Nor can a constraint obtained on an open set U be propagated off U: under Assumption 1, which requires only C^2, a function is not determined off U by its behaviour on U. Distinctness is therefore assumed at the origin, which is what the argument uses. Note also that with distinct $g_k''(0)$ no non-trivial permutation commutes with D, so the reachable class is in fact the sign matrices; we keep the weaker statement $R \in G_{\pm}$ because $G_{\pm}$ is the group the downstream results (Cor. 1, Thm. 3) are phrased in.

COROLLARY 1 (NO-SIGN-FLIP CASE)

Proof. Suppose R has $d_k = -1$ for some k. By the local hypothesis there is z_k inside the invariance neighbourhood V with $g_k(z_k) \neq g_k(-z_k)$. Pick $z \in V$ with that z_k and the other coordinates zero: then $g_k(-z_k) \neq g_k(z_k)$ produces $\sum_k g_k((Rz)_k) \neq \sum_k g_k(z_k)$, contradicting invariance. So $d_k = +1$ for all k, which leaves only the identity among sign matrices. □

THEOREM 3 (SAMPLE COMPLEXITY)

Proof. For any fixed permutation π, the per-sample statistic $X_i^{(\pi)} := \|c_i^* - \pi(c_i^*)\|_2^2$ lies in $[0, 4B^2]$ since $\|c_i^*\|_2 \leq B$ and $\|\pi(c_i^*)\|_2 = \|c_i^*\|_2 \leq B$. Hoeffding's inequality for bounded variables therefore gives

$$\Pr\left[\left|\tfrac{1}{|L|}\sum_i X_i^{(\pi)} - \Delta_\pi\right| \geq t\right] \leq 2\exp\bigl(-2|L|t^2/(4B^2)^2\bigr) = 2\exp\bigl(-|L|t^2/(8B^4)\bigr).$$

Setting $t = \Delta/2$ and union-bounding over the $2^K K! - 1$ non-identity elements of $G_\pm$, $\Pr[\exists T \neq \mathrm{id} : \hat{\Delta}_T \leq \hat{\Delta}_{\mathrm{id}}] \leq 2(2^K K! - 1)\exp\bigl(-|L|\Delta^2/(32B^4)\bigr)$. The encoder-error assumption translates the bound from the oracle statistic $X_i^{(T)}$ to the empirical alignment loss $\|z_i - c_i^*\|_2^2$ via the triangle inequality. *The slack bound needs care.* Writing $u_i = c_i^* - T(c_i^*)$, the cross term is $2|\langle z_i - c_i^*, u_i\rangle| \leq 2\epsilon\|u_i\|$, and averaging with Cauchy–Schwarz gives $\overline{\|u\|} \leq \sqrt{\hat{\Delta}_T}$, so the slack is at most $2\epsilon\sqrt{\hat{\Delta}_T} + \epsilon^2$. (The cruder bound $\overline{\|u\|} \leq 2B$ does not suffice: since $\Delta \leq 4B^2$ the resulting $B\sqrt{\Delta}$ already exceeds $\Delta/2$.) With the Cauchy–Schwarz form the slack is below $\Delta/4$ whenever $\epsilon < \frac{1}{2}\sqrt{\Delta/2}$, a weaker condition than the theorem's $\epsilon \leq \sqrt{\Delta}/4$, which therefore suffices. Requiring the union-bound to be $\leq \delta$ gives the stated sample-complexity threshold. □

ANCILLARY THEOREM (CAUSAL ADDITIVITY OF NAM HEAD)

Statement. If $\hat{y}(z) = b + \sum_k g_k(z_k)$ then $\mathbb{E}[\hat{y} \mid \mathrm{do}(z_k = c)] - \mathbb{E}[\hat{y}] = g_k(c) - \mathbb{E}[g_k(z_k)]$, depending only on g_k, not on the joint distribution of z_{-k}. **Proof.** Direct expansion: $\mathbb{E}[\hat{y} \mid \mathrm{do}(z_k = c)] = b + g_k(c) + \sum_{j \neq k} \mathbb{E}[g_j(z_j)]$; subtracting $\mathbb{E}[\hat{y}] = b + \sum_j \mathbb{E}[g_j(z_j)]$ gives the claim. For non-additive f, Taylor-expanding around $\mathbb{E}[z]$ introduces $\mathrm{Cov}(z_j, z_k)$ cross-terms that depend on the joint. The empirical $\sim 23\times$ smaller normalised intervention variance of ConceptNAM vs CBM (§5.2) is consistent with this prediction. □

REMARK ON THE HUNGARIAN-ALIGNED LOSS (WITHDRAWN CLAIM)

Theorem 3 bounds a fixed-coordinate MSE alignment loss $\|z_i - c_i^*\|^2$. iCBM additionally minimises a per-batch Hungarian-aligned loss $\min_\pi \|z_i - \pi(c_i^*)\|^2$. That this loss is bounded above by the MSE loss does *not* carry the identity-recovery guarantee over. The Hungarian loss is zero at $z = Pc^*$ for every permutation P, so it does not prefer the identity to any other permutation; a smaller loss value cannot certify a fixed coordinate assignment. What the Hungarian loss can support is recovery *up to permutation*. Fixing coordinates needs a term that is not permutation-invariant, such as a fixed-coordinate MSE; iCBM's `alignment` option selects the Hungarian loss *or* MSE, not both (`baselines/icbm.py`), so under its default Hungarian setting iCBM identifies concepts only up to permutation. No number in this paper depends on the withdrawn claim; NMI, MIG and DCI are permutation-invariant.

REMARK ON KHEMAKHEM ET AL. (2020) AUXILIARY-VARIABLE IDENTIFIABILITY

The iVAE-style identifiability of (Khemakhem et al., 2020) would strengthen Theorem 1's $G_\pm$ conclusion to element-wise monotone permutations, but requires Khemakhem's A2 (the auxiliary variable u produces sufficient-statistic variability of dimension at least $2K$). For our binary-target classification setups (Heart, Pima, Adult), $u = y$ is binary and A2 cannot hold for $K \geq 2$; for our scalar-regression setups the variability is continuous but the exponential-family assumption is not met by our Gaussian-posterior encoder. We therefore do *not* import Khemakhem's stronger conclusion and report only the fixed-head signed-permutation symmetry of Theorem 1, which is not

an identifiability class. NMI, MIG, and DCI are all permutation-invariant, so the metric ordering in our experiments is unchanged by which identifiability class is used.

WHAT THE HIERARCHY DOES *not* CLAIM

(1) Theorems 1 and 3 do not predict *which* Level-1 architecture wins empirically, and they are not identifiability results for jointly trained models: Theorem 1 restricts the orthogonal symmetries of one *fixed* additive head, whereas training may change the encoder and every g_k together, so equivalent representations reached through a different additive head $\tilde{g}$ are not excluded. It is a necessary condition on the head, not a characterisation of what task supervision identifies. (2) The sparsity-to-permutation claim is withdrawn (Remark 2); and in general posterior collapse (empirically observed for β-VAE+sup) breaks identifiability even when the loss geometry permits it. (3) Theorem 3 is an *upper* bound; the actual sample complexity depends on encoder capacity and batch curvature.

T IDENTIFIABILITY: WHAT THE THEOREMS DO AND DO NOT BUY

App. S states and proves two results about an additive head, and withdraws a third: locally at the origin the symmetry group of a *fixed* head inside $\mathrm{O}(K)$ is contained in the signed permutations (Theorem 1); an infinite sparsity penalty does *not* in general pin a permutation (Remark 2, withdrawn); and $|L|$ labels suffice to select the right permutation given an already-accurate encoder (Theorem 3). We keep them in the appendix rather than the contribution list because they are inert here: Assumption 1 requires pairwise-distinct $g_k''(0)$ and our shape bank gives $\{0, 2, 0, 0, -2, 0\}$, so it fails at $K \geq 3$; Corollary 1 additionally needs no even g_k, false for every $K \geq 2$ since $z^2 - 1$ is in the bank; and no experiment in §5 cites either. A generator with pairwise-distinct $g_k''(0)$ would make the first testable, which we regard as the natural next experiment for the theory rather than a claim we can make now.

Proofs of Theorems 1 and 3, the withdrawal of Remark 2, and a Remark on the Hungarian-alignment loss used by iCBM are in App. S.

What places ConceptNAM at Level 1 is its supervision budget, not these theorems — it consumes x and y and never c, which is the whole of the criterion (§4). DetNAM is included as the deterministic NAM-family Level-1 comparator: same per-feature encoder (stage 1) and same additive head (stage 3), but a deterministic linear bottleneck replacing the VIB (no KL, no TC, no sparsity regulariser; loss is task-only). DetNAM isolates the contribution of the VIB stack within the NAM family.

U EXTENDED DISCUSSION: WHEN-TO-USE GUIDANCE AND LIMITATIONS

BROADER IMPACT

Concept-based models are proposed for settings where a wrong explanation is worse than no explanation — clinical decision support most often. An unmeasured floor makes a method look partially interpretable when it is not, and the regime where that error is largest is precisely the small tabular one those deployments use. We release the benchmark, the chance-level and margin-calibration procedures, and the audit.

Why the floor is highest exactly where it is least reported. Chance NMI falls with the number of held-out samples; the *operative* floor barely does. Image disentanglement benchmarks sit on the good side of the sample-size axis (10^3–10^4 evaluation samples), but the gap to our cell is a factor of two on the operative floor (0.05–0.06 for dSprites against our 0.115), not the factor of ten the random-code floor would suggest. Tabular concept discovery sits on the bad side of the sample-size axis, because tabular datasets are small. The floor is therefore not an artifact that better engineering removes; it is a property of the regime, and the only remedy is to measure and subtract it.

Why does supervision dominate? Reconstruction-only training maximises $I(Z; X)$, invariant to rotations of Z. Task supervision does not break that invariance information-theoretically — an invertible R leaves $I(RZ; Y) = I(Z; Y)$ — but it does through the *predictor class*: an additive head $\sum_k g_k(z_k)$ fits Y, which is additive in the true concepts, more easily in some rotations than others, so what task supervision can favour is an alignment the head finds learnable, not one that carries more

information: for a rotation, aligning Z with C cannot raise $I(Z;Y)$. Level-3 minimises $\mathbb{E}\|C - \hat{C}\|$ directly. This is the chain-rule reading of Khemakhem et al. (2020), and it presumes $I(Z;Y)$ is actually optimised — which Table 1 shows is not automatic, though App. H shows optimising it is not sufficient either. The floor-effect control constrains the account only weakly: the raw $+0.20$ vs $+0.02$ split disappears once both are read as excess over a chance level that moves with K.

Why does the VIB fail on tabular concept discovery? All three predictions our VIB design imported from the image domain — concept emergence without supervision (Higgins et al., 2017; Kim & Mnih, 2018), implicit pruning of excess dimensions, and stability at high p — fail on tabular data (§5.4, sweeps in App. R). The relevant Level-1 axis here is encoder family (NAM vs VAE), not VIB-vs-deterministic.

This appendix contains the long-form discussion subsections that were compressed out of the main body (§5) to respect the ICLR 9-page limit.

When to use which method

One restriction first, because it is stated four times elsewhere and was violated here: we make no Level-1 architecture *recommendation. The only recommendation this paper supports is about supervision level. The Level-1 lines below record what we observed, not what we advise.*

In the tabular regime tested here: with full concept labels, use **CBM**; with task labels only and small concept count ($K \leq 4$), use **DetNAM or NBM at the configuration selected in App. G** — under the shipped defaults NBM led, but the corrected run puts them within seed noise of each other (DetNAM 0.51 vs NBM 0.48, within seed noise on the benchmark, Table 17); with task labels only and larger K, use **DetNAM** (the NBM advantage disappears at $K \geq 16$ where DetNAM, NODE-GAM and NBM converge to the same mid-tier band). ConceptNAM remains a defensible choice only when its sparsity-regularised attribution matrix is required downstream; its VIB collapses the posterior to a single active dimension at $K \geq 8$ (Table 18).

Limitations

(1) *Tabular scope.* The benchmark is tabular by design; the image-mode extension to dSprites is a proof-of-port for the encoder, not a full image-domain validation; extending to NLP/graph domains requires different encoders and concept-labeled datasets, which is out of scope. (2) *Absolute NMI.* On the shipped configuration Level-1 NMI stayed below 0.32, consistent with Locatello et al. (2019); the corrected configuration lifts DetNAM to 0.51 and NBM to 0.48 at $K{=}4$, so the ceiling is partly configuration, not only identifiability, and we do not attribute it to one cause. (3) *Local, fixed-head symmetry only.* It is a statement about one fixed head, not about what a jointly trained encoder and head can identify (App. S). Theorem 1 constrains R from invariance on a neighbourhood of the origin, using $g_k''(0)$ pairwise distinct; it does not rule out non-linear symmetries of F elsewhere, and it says nothing when the second derivatives coincide at the origin. (4) *Proxy real-world concepts.* The real-world experiments use first-K-feature proxies (§B.5 (vii), App. O) because ground-truth concept annotations are not publicly available for the four tabular datasets studied. (5) *IPS weights* are uniform; a user study is future work. (6) *Loss decomposition* reduces to KL + TC (α and γ multiply identical scalars; kept distinct only for free-bits routing). (7) The feature-permutation OOD row (§B.5 (iii)) is a negative control, not a robustness metric: per-feature MLPs are column-index-bound by design. ACS-IoU sensitivity to top-k is quantified in App. F. (8) *Capacity is not matched by the protocol.* Our protocol equalises supervision and leaves model size free; across ConceptDiscoveryBench the trainable-parameter count spans $7.1\times$ at $K{=}4$ (Table 17), and the spread runs *against* the NMI ordering — under the shipped defaults NBM scored highest among non-CBM methods with the second-smallest model. We ran the equal-parameter control on the headline cell (`exp_capacity_matched.py`, `capacity_matched.json`): every method's widths are scaled so its trainable-parameter count lands within 2.5% of a common budget — the median and, separately, the maximum of the shipped Level-1 counts — and each is scored against the untrained null of its own *matched* architecture. The ordering survives at both budgets: NBM and DetNAM on top (NMI 0.288/0.267 at the median budget), NODE-GAM and ConceptNAM in the middle (0.145/0.142), the supervised VAEs at the bottom (0.119 each), CBM 0.36 above the best Level-1 method at either budget; all six Level-1 methods clear their matched floors under Holm. Capacity

therefore does not explain the within-Level-1 ordering on this cell, though the ordering remains one cell's and five seeds' worth of evidence. (9) *We make no Level-1 architecture recommendation.* Under the shipped defaults ConceptDiscoveryBench put NBM ahead of DetNAM at $K{=}4$; under the corrected run the two are within seed noise, and both converge to the DetNAM/NODE-GAM mid-tier at $K{\geq}16$. The only recommendation we stand behind is about supervision level. (10) *Scaling collapse is not explained.* Even CBM (Level 3, full concept supervision) drops from NMI 0.69 to 0.11 as the sweep grows from $(K{=}4, p{=}16)$ to $(K{=}32, p{=}128)$, while remaining above its cell's floor; the Level-1 curves reach that floor instead. We therefore do not claim our protocol or any tested method scales to arbitrary (K, p), and we cannot name the bottleneck: training budget, model capacity, the floor and the metric's normaliser all change with (K, p) at once; nothing here isolates architecture. (11) *One external re-evaluation, and it reorders nothing.* Re-scoring eight public dSprites checkpoints (App. M.2) changes every improvement ratio but no ordering or verdict, because they share one architecture and hence one floor; a re-evaluation across architectures, where the correction can reorder, remains to be done; the five literature factor grids of §3 use the random-code null, which does not test the data-dependent component this paper is about. (12) *The floor needs ground truth.* An untrained encoder can be run on any data, but NMI, MIG and DCI — and so their floors — need the true concept values; the procedure applies wherever these metrics already do and nowhere else. And that concept-supervised CBM recovers the concepts it is trained on best is expected; the gap sizes what task labels leave on the table, it is not a causal estimate of what collecting concept labels would buy in a new domain.

V ANTICIPATED REVIEWER QUESTIONS

Q1. What's new beyond Locatello et al. (2019)? Three things: (i) a supervision-matched grouping that says which comparisons are admissible at all, not by how much one level wins — §5.3 declines to put a number on what a task label buys; (ii) supervision-matched protocol that exposes confounds in current practice; (iii) a metric suite (NMI, MIG, DCI, and the new ACS) reported component-wise, which reveals divergence invisible to NMI alone.

Q2. Is this just a negative result? No: the evaluation framework, the ConceptDiscoveryBench dataset, and the corrected method ranking are positive contributions. The negative result (the VIB never helps and frequently triggers posterior collapse) is itself useful guidance for practitioners — it identifies *which* canonical Level-1 ingredient to drop. The one positive recommendation the evidence supports is about supervision, not architecture: if concept labels are obtainable, use them. Within Level 1 we make no architectural recommendation — the NBM/DetNAM ordering reverses between the shipped and corrected configurations; capacity, unmatched by $7.1\times$ in the benchmark, does not explain the ordering in an equal-parameter control (Limitation (8)).

Q3. Why tabular only? Tabular data allows clean supervision matching (same encoder for all methods). Extending to image/NLP requires architecture-specific encoders, which would re-introduce the confound we aim to eliminate.

Q4. Aren't 5 seeds too few? TOST at $n{=}5$ with $\delta{=}0.05$: $p < 0.001$ for the three VAE-family Level-1 pairs (ConceptNAM, β-VAE+sup, FactorVAE+sup); the cross-family DetNAM pair sits at $p{=}0.436$ outside the band, so equivalence is a within-family statement, not an all-Level-1 statement. Whether these are genuinely small effects or low power is not something observed power can answer, since it is a monotone transform of the p-value. A minimum detectable effect can: at $n{=}5$ and $\alpha{=}0.05$ our design resolves differences of about 0.029 NMI at 80% power (the median paired-difference spread over all method pairs; `exp6_statistical_analysis.json` records the standard deviation used), and the triplet's pairwise differences are 0.0008–0.0020 — an order of magnitude below what we could see. App. D.1 extends this to $n{=}10$ (NMI shifts ≤ 0.02, TOST verdicts unchanged).

Q5. Why does β-VAE+sup collapse on real data? Standard posterior collapse (Bowman et al., 2016). Our ACS and active-unit metrics detect this; NMI does not.

Q6. How much does fixing leakage change results? Our prior "ConceptNAM beats baselines by 64–191%" became "ConceptNAM is equivalent to supervised baselines" — a qualitatively different conclusion from the same code.

Q7. Did you compare against NODE-GAM and NBM? Yes (App. E). We adapt both to the K-concept Level-1 setting by adding the same linear bottleneck used by DetNAM, then run them on every ConceptDiscoveryBench point with 5 seeds. NODE-GAM lands in the same band as DetNAM on NMI. NBM led the non-CBM methods at $K{=}4$ under the shipped defaults (0.44 vs DetNAM 0.24), but on the corrected run we ask you to cite the two are within seed noise (0.48 vs 0.51), and both converge to the mid-tier band at $K{\geq}16$. Vanilla NAM (no concept output) is included as a task-RMSE reference (App. J).

Q8. Does NBM beat DetNAM at small K? Not on the run we ask you to cite. Under the shipped defaults NBM led (0.44 vs 0.24). Once each method's un-swept encoder flags are reselected on validation task RMSE, the two are within seed noise of each other at $K{=}4$ (DetNAM 0.51 vs NBM 0.48, Table 17) and the ordering carries no weight — capacity matching leaves the two on top but does not separate them (Limitation (8)). We make no architectural recommendation between them.

Q9. Why does CBM also collapse at $K{=}32$? CBM's fall at $(K{=}32, p{=}128)$ — the only curve there that stays measurably above its null — is consistent with an architectural ceiling rather than a supervision-budget effect: full concept supervision does not rescue it when the model must allocate attention across 128 raw features. *The Level-1 curves are a different matter: at $K{\geq}16$ they sit within a few thousandths of those cells' measured floors, so what looks like a shared plateau is the floor, not a ceiling* (Fig. 1). Limitation (10) of the limitations records this as an open question and we leave its resolution to future work.

Q10. How is ConceptDiscoveryBench released? Through the supplementary archive: a manifest enumerating every (dataset, K, p, replicate seed) point, the deterministic generators, W hashes for the fixed-W assertion, and per-method JSON results from the benchmark sweep. The accompanying datasheet (App. E) follows Gebru et al. (2021).

Q11. Are the audit fixes regression-tested? Yes. The supplementary release contains 48 pytest tests ($\sim$ 30 s on CPU): 7 forward / dataset smoke tests in `test_synthetic.py` and `test_cdnam_forward.py`, 15 unit tests of the metric, split and statistics primitives in `test_core_metrics.py`, 9 floor-measurement tests in `test_floor.py`, plus 17 regression tests in `test_audit_fixes.py` pinning the fixed-W invariant, the unified-saliency attribution shape, the ablation-toggle gating, the CBM label-mask contract, and the DCI probe-split fix. A failing test in the audit-fix module signals a silent regression of one of the audit corrections.

Q12. Why not just use a frozen foundation-model featurizer (CLIP, DINOv2, GPT-style tabular embeddings) and apply a linear concept head? Two reasons. (i) The contribution we are making is methodological: a matched-supervision protocol and a benchmark that apply to *any* encoder, including FM-frozen ones. Plugging a FM in front of our Level-1 head is fully compatible with the benchmark; we expect such a setup would still exhibit the NMI-invisibility we report, because the confound lives in the supervision channel rather than the encoder. (ii) Most concept discovery in tabular/clinical settings is performed without a pre-trained foundation backbone, and any FM-based approach must itself be evaluated against the protocol we propose to avoid the same supervision confounds we identify in §3–§4. We therefore see FM-based concept discovery as a follow-up that benefits from, rather than displaces, our protocol.

Q13. How does the benchmark behave as K grows past the values reported in the main paper? ScalingSynth ($K \in \{2, 4, 8, 16, 32\}$, App. E) is the relevant sweep. The Level-1 NMI band tightens with K because more concepts means more degrees of freedom for the supervision-matched alignment to land on the same plateau; the gap to a CBM oracle is *not* monotone in K: on the corrected run it is 0.200, 0.392, 0.402, 0.072 at $K{=}4, 8, 16, 32$, widening to $K{=}16$ and then collapsing at $K{=}32$ because CBM itself falls to 0.124 there. The trend is non-monotone, and we do not use the theorems of App. S to explain measured numbers (§5). We do not report $K > 32$ in the main paper because dataset construction time becomes the bottleneck, not the modelling.

Q13b. If architecture is invisible to NMI, why propose iCBM? The within-VAE-family NMI equivalence (§5) may seem to contradict the iCBM packaging described in App. M, which is background here rather than a contribution (§1). The contradiction dissolves once "invisible to NMI" is read precisely: NMI measures average steady-state cluster alignment at a fixed operating point, not the three properties that differentiate architectures in production use:

1. **Sample efficiency.** Not established: no shipped ablation includes iCBM in a K-mismatch sweep (Table 36 runs ConceptNAM/DetNAM/CBM), so we make no claim about its stability as the supervised-concept count shrinks.
2. **Identifiability under partial supervision.** Only iCBM accepts partial concept labels. Its default Hungarian alignment identifies concepts only up to permutation, so Thm. 3's fixed-coordinate bound applies to it only in its MSE-alignment setting; the bound is silent for other Level-1 methods — and, per App. T, it is inert on this generator, so this is a statement about applicability, not measured advantage.
3. **Variance under real distribution shift.** On the clinical probe (App. N, BCE-trained binary head), iCBM has a smaller AUC std than ConceptNAM across 5 seeds — $5.6\times$ on UCI Heart (± 0.02 vs ± 0.13) and $12.6\times$ on Pima (± 0.02 vs ± 0.19), both point estimates. How firm this is depends on the test and on the dataset. Bartlett rejects equal variance ($p = 0.006$ Heart, $p = 0.0003$ Pima), as does mean-centred Levene ($p = 0.003$, $p = 0.042$); the median-centred Brown–Forsythe test does *not* on either ($p = 0.12$, $p = 0.36$). The paired bootstrap ratio CI excludes 1 on Heart ($[3.0, 16.5]$) but not on Pima ($[0.6, 38.6]$) — and that paired interval is not calibrated: under a null of equal variances it excludes 1 in 9% of simulated five-seed datasets, about twice the nominal 5%, while resampling the two methods independently excludes it in 4% (`paired_bootstrap_coverage.json`); the paired scheme is anti-conservative, not useless. The conservative independent scheme widens Heart to $[0.8, 30.5]$ and includes 1. Mean AUC differences are not significant (Welch, $p > 0.10$). No procedure valid at five seeds separates the variances on either dataset: the conservative independent bootstrap includes 1 on both, and median-centred Brown–Forsythe rejects on neither. We therefore report the point ratios as a direction only; the gap is not established on Heart either. This passage runs six variance tests and two mean tests across two datasets without a declared family, unlike the rest of the paper; we treat it as exploratory and let the disagreement between tests, rather than a corrected threshold, set how much weight the claim carries. It is also *not* iCBM-specific: DetNAM shows the same ratio against ConceptNAM ($5.0\times$ Heart, $13.8\times$ Pima), so whatever the gap is, it is a property of ConceptNAM (consistent with, not shown to be, posterior collapse) rather than an iCBM advantage (App. N, `analyze_clinical_variance.py`).

App. K quantifies the decomposition: unmatched-native NMI spread 0.585 vs matched task-only NAM-family spread $\mathbf{0.096}$ between the two *distinct* NAM-family architectures.

Q14. How sensitive is the TOST equivalence finding to the choice of ε? The within-Level-1 VAE-family triplet (ConceptNAM, β-VAE+sup, FactorVAE+sup) on Synthetic-Concept has, *on the shipped block*, pairwise $|\text{mean_diff}|$ values of 0.0012, 0.0008, and 0.0020 NMI. Running the same paired TOST used for the main table over an ε grid (`experiments/analyze_tost_eps_sweep.py` $\rightarrow$ `results/tost_eps_sweep.json`), the triplet is equivalent $3/3$ at $\varepsilon \geq 0.015$ and $0/3$ at $\varepsilon \in \{0.005, 0.010\}$; the smallest margin at which all three pairs pass is $\varepsilon_{\min} = 0.0145$ on raw p, or 0.021 under this sub-analysis's own Bonferroni family (m=3, the three pairwise comparisons within the triplet). The finding therefore holds at a margin roughly 2–3$\times$ tighter than the δ=0.05 used in the main table, but it does *not* hold at the two tightest grid points: with n=5 seeds the residual seed-to-seed spread, not the point estimate, is what the test cannot overcome. Each $\varepsilon_{\min}$ here and below is given on both raw and Bonferroni-adjusted p, because Table 1's own column is Bonferroni-adjusted.

On the *corrected* block the answer is different. The same sweep (`experiments/analyze_tost_corrected.py` $\rightarrow$ `results/tost_corrected.json`) gives pairwise differences of 0.0014–0.0154, *no* pair equivalent at ε=0.0145, and all three passing only from $\varepsilon_{\min}$=0.029 (raw p) or 0.036 (Bonferroni, m=3) — still below δ=0.05, but roughly twice the shipped block's margin. An equivalence verdict, like a floor, is a property of the configuration and travels no better.

Table 38: Interpretability metrics (synthetic $K{=}4$, **5 seeds**), on a **common axis**. Each method's native attribution space differs — ConceptNAM's learned $A \in \mathbb{R}^{K\times 16}$ over *input features* against CBM's readout in $\mathbb{R}^{K\times 128}$ over *hidden units* — and concentration statistics scale with vector length, so on native axes ConceptNAM would look more concentrated mainly by having an $8\times$ shorter row. The driver therefore puts every method through the shared `input_attribution` utility (row-normalised gradient×input on the $p{=}16$ feature axis), which is the same utility ACS uses, so the columns below are comparable. On the common axis CBM is the most concentrated (Gini 0.669, peak $5.7\times$ uniform), ConceptNAM second (0.551, $4.4\times$), β-VAE+sup last (0.260, $2.3\times$, and 0 active units — its posterior is collapsed, so its attribution is near-uniform noise). Concentration is not by itself a virtue. "Shape Smooth." requires an additive head and exists only for ConceptNAM.

Method	Gini	Peak / uniform	$H/\log p$	Shape Smooth.	Active Units
CBM joint	$\mathbf{0.669 \pm 0.005}$	$\mathbf{5.67 \pm 0.11}$	$\mathbf{0.699 \pm 0.006}$	–	$\mathbf{4.0 \pm 0.0}$
ConceptNAM	0.551 ± 0.038	4.36 ± 1.16	0.824 ± 0.028	$\mathbf{0.0005 \pm 0.0000}$	3.4 ± 0.5
β-VAE+sup	0.260 ± 0.069	2.30 ± 0.50	0.964 ± 0.021	–	0.0 ± 0.0

Pairs outside the triplet. The sweep covers all 15 method pairs. Two readings in the main text depend on the wide margin and are scoped accordingly: Level-0 β-VAE vs. the Level-1 VAE methods is equivalent only for $\varepsilon \geq 0.040$ ($\varepsilon_{\min} = 0.0395$–$0.0440$), and DetNAM vs. ConceptNAM requires $\varepsilon \geq 0.075$ — consistent with our refusal to claim cross-family equivalence in §5.1.

Q15. Does the supervision-matched ranking survive label noise? For input noise, yes. `exp16_noise_robustness.py` injects Gaussian noise on the *inputs* at $\sigma \in \{0.05, 0.1, 0.2, 0.5, 1.0\}$ and every method is flat across that range, so the ranking is unchanged (`exp16_noise_robustness.json`). *We have no result for concept-label noise*: the driver perturbs X only, and the target-noise axis is audit item B3a (App. Z). The question is open.

Q16. How does this benchmark compare to recent ICLR/NeurIPS concept-discovery work (2024–2026)? Two recent benchmarks are close. Bortolotti et al. (2024) provide a neuro-symbolic suite for concept quality with its own generators and metrics; it targets reasoning shortcuts rather than supervision matching, but it is a concept-quality benchmark. Almudévar et al. (2026) argue on information-theoretic grounds that predicting a concept does not make a component carry only that concept — close in spirit to our VIB result, reached independently. Parisini et al. (2025) quantify concept leakage, overlapping our leakage-control axis, and Chen et al. (2024) give the identifiability counterpart: a rich enough task *distribution* reduces the equivalence class to permutation and scaling. Also close are TabCBM (Espinosa Zarlenga et al., 2023), GlanceNets (Marconato et al., 2022) and the iVAE line (Khemakhem et al., 2020; Lachapelle et al., 2022; Hyvärinen et al., 2023). What none of them provides is a measured floor: the concept-recovery scores they report are read against an implicit zero, which is the failure mode this paper documents. (TabCBM does vary the concept-supervision budget, from full to none; what it does not report is a chance level for the recovery metric.) **Which of them we actually ran.** TabCBM, yes, at Level 3 with full concept supervision and again under the Level-1 protocol below. GlanceNets and iVAE, *no*: neither appears in `baselines/` nor in any shipped artifact. We name them here as related work only. The reference numbers ConceptDiscoveryBench offers future work are therefore the methods in Table 17, each with its cell's measured chance level beside it. App. D.2 additionally reports a reduced TabCBM-style re-implementation and a PyTorch port of the official method with its published regularisers under the task-only Level-1 protocol on Synthetic-Concept (10 seeds): both land between DetNAM and ConceptNAM (NMI 0.205 ± 0.044 reduced, 0.141 ± 0.049 official port), consistent with the body claim that matched-supervision differences within Level-1 are small relative to the cross-supervision gap to CBM.

W Reproducibility manifest

Every main-text number is regenerated from a JSON artefact by the corresponding experiment driver listed in Table 39. Commands are run from the repo root after installing the package in a Python 3.10+ virtual environment (instructions in supplementary `README`).

Using the floor, and what it costs. The procedure is a library call, not a driver: `f = measure_floor(X_test, z_true, n_concepts=K, draws=D, encoder=...)` returns the random-code and untrained-encoder floors, their 95th percentiles, the AMI-surviving part and the draws themselves; `f.test(scores_by_seed)` is the paper's verdict rule (one-sided Welch against the draws, both variances carried), and `adjust_pvalues` applies Bonferroni, Holm, BH or BY across the methods compared. `f.test` treats the draws as independent; when they share evaluation data with the scores, the seed-paired variant of App. B.1 is the stricter unit. `f.clears(score)` is only the single-score p_{95} screen, not the verdict rule. Cost, measured on one CPU thread (`analyze_floor_cost.py`, `floor_cost.json`): on the headline cell 200 linear-null draws take 27 s in total (each draw evaluates the matched NMI three times, for the code, encoder and adjusted variants), and an architecture-matched draw through an untrained DetNAM takes 54 ms; the whole timing process, all three cells included, peaked at 0.7 GB resident. The cost is dominated by the K^2 pairwise mutual-information and expected-MI terms, so at $K{=}32$ a draw takes 8.9 s — but the draw-to-draw spread there is so small that a handful of draws suffice. A stopping rule follows from a pilot: $D = \lceil(\hat{\sigma}/\text{target SE})^2\rceil$ draws put the null mean's Monte-Carlo error at the target; for SE 0.001 that is 99 draws on the headline cell and 176 at $n{=}2000$, and 200 draws give the p_{95} a bootstrap SE of 0.002. Against training, which the floor never repeats, this is a small fraction of one method's cost per cell; DCI's probe fits, which we do not time here, are the expensive exception, which is why the dSprites probes use fewer draws.

The released supplementary contains 48 tests ($\sim$ 30 s on CPU): 7 forward / dataset smoke tests, 15 primitive unit tests, 9 covering the chance-floor procedure itself (`cdnam.floor` — that the adjusted estimator reduces to `compute_nmi`, that the floor is far from zero at small n, that the removable part falls with n while the residue does not, that the floor moves with the binning, that the public verdict `Floor.test` is the paper's Welch rule rather than the p_{95} screen, and that `paired_gain` is the per-run paired t), plus 17 regression tests pinning the fixed-W invariant, the unified-saliency attribution shape / row-sum, the ablation-toggle gating, the CBM and ConceptNAM `label_mask` contracts, the DCI probe-split fix, the `import experiments` namespace-shadow guard, and the dSprites train-only PCA fit invariant. A failing test in the audit-fix module means an audit fix has been silently regressed. YAML configs materialising the default hyper-parameters for each driver are included in the supplementary release.

X Hyperparameters for the headline runs

Baseline defaults follow the original papers ((Koh et al., 2020; Chen et al., 2018; Agarwal et al., 2021)); supervised β-VAE adds an auxiliary MLP from z to y trained with MSE weight 1.0.

Y Compute

All experiments run on a single consumer GPU (RTX-class, 24 GB), PyTorch 2.11, CUDA 13.0. Wall-clock per full experiment driver: Exp 6 (5 seeds, 5 methods) $\approx$ 12 min; Exp 10 (4 methods $\times$ 4 label fractions $\times$ 5 seeds) $\approx$ 25 min; Exp 2 (5 λ $\times$ 3 seeds $\times$ 80 epochs) $\approx$ 8 min. End-to-end regeneration of every main-paper table from a clean repo clone is under 3 GPU-hours. No hyperparameter search was performed on test splits; see `configs/` for the static defaults used.

Z Self-audit and corrections to earlier versions

Where do the old numbers go wrong? (Leakage case study)

Our public self-audit (App. Z, Table 41) documents five classes of evaluation defect we found in our own prior revisions: a scaler fit on full data; test-set early stopping; NMI evaluated on training samples; a KL-only "UCBM+NAM" strawman posing as a two-stage baseline; and a non-canonical NMI duplicated across drivers. Correcting all five turned the prior headline "ConceptNAM outperforms baselines by 64–191%" into "ConceptNAM is statistically equivalent to supervised baselines" — a qualitatively different conclusion from the same codebase.

SEMI-SUPERVISION: UNIVERSAL OR METHOD-SPECIFIC?

Result (Table 5, App. B): the gain reaches every method but not equally, and for one it is inside its own noise — β-VAE+sup moves $0.139 \to 0.146$ at 10% labels, less than its seed std of 0.017, so the gain is not universal. CBM benefits most ($0.16 \to 0.49$ at 10%), and at 50% labels both supervised VAEs (0.38, 0.34) pass ConceptNAM (0.31).

A CLAIM WE WITHDREW THREE TIMES

The floor-effect control (§5.3) produced a quantitative headline in three successive revisions, and each was withdrawn by the next. We record the sequence because the failure mode — re-aggregating one 4-cell, 5-seed grid until a stable-looking number appears — is easy to repeat and hard to see from inside.

1. *"A scalar task label buys about 15% of what concept labels buy."* Computed with five methods; NBM and NODE-GAM were absent from the driver's method list. NBM is the largest Level-1 excess in every cell, so the figure described DetNAM, not Level-1.
2. "18–34%." Correct for the per-cell *maximum*, but that is a biased order statistic: the argmax method's seed spread is 4–6$\times$ wider at $K{=}4$ than at $K{=}2$, inflating the maximum selectively at one end of the difficulty axis.
3. *"The best Level-1 method's absolute excess barely moves"* ($0.141 \to 0.135$). These are two diagonal corners of a 2×2 grid. Over all four cells the quantity ranges 0.124–0.181 ($\times 1.46$), *more* variation than CBM's $\times 1.41$; along the σ axis alone at $K{=}4$ it rises $+28\%$ while CBM rises $+2.3\%$, reversing the reading.

The figure is not stable in n either: on the *shipped* bench protocol ($n{=}2000$, `bench_summary.json`, null 0.060, best Level-1 is NBM) the best-Level-1/CBM excess ratio is 59.6%, against 27.0% at $n{=}800$ (`exp_floor_control.json`, canonical null; *an earlier version printed* 27.2%*, computed against the withdrawn* 0.1143 *estimate*). The grid supports a dispersion statement only, which is what §5.3 now reports.

Table 41 maps each evaluation defect identified in our public self-audit (released in supplementary) to the pre- and post-correction headline number. All three problem classes are instantiated.

Post-audit follow-up experiments. After every retracted claim was re-derived under the corrected protocol, we ran three further controlled experiments (S1, S2, S3 in §5.4) to test the remaining positive claims a Level-1 VIB design might still earn: concept emergence from task labels alone, K-mismatch self-pruning, and high-dimensional robustness. None survive. An analysis script written before S1–S3 were run (released in App. W; we have no external timestamp for that ordering, so this is our own pre-specification, not a registered preregistration) encodes the statistical kill criterion; its summary reports 0 significant ConceptNAM wins across S1 and S2 paired tests and 0 wins across S3 blocks, triggering the pre-specified pivot to the negative-result framing of this draft. We treat this as the strongest form of audit: not just retracting wrong claims but publicly recording the falsification of plausible follow-up claims under the same protocol.

Z.1 ACS, IN FULL

For S seeds producing attribution matrices $A^{(s)} \in \mathbb{R}^{K \times p}$, let $\sigma_{ss'}$ be the Hungarian row alignment between seeds s and s' under cosine distance. Then

$$\mathrm{ACS}_{\cos} = \binom{S}{2}^{-1} \sum_{s<s'} \frac{1}{K} \sum_{k=1}^{K} \cos\big(A^{(s)}_{k,:},\, A^{(s')}_{\sigma_{ss'}(k),:}\big),$$

$$\mathrm{ACS}_{\mathrm{IoU}} = \binom{S}{2}^{-1} \sum_{s<s'} \frac{1}{K} \sum_{k=1}^{K} \mathrm{IoU}\big(\mathrm{top}_k(A^{(s)}_{k,:}),\, \mathrm{top}_k(A^{(s')}_{\sigma_{ss'}(k),:})\big).$$

The composite $\mathrm{IPS} = \frac{1}{4}(\widetilde{\mathrm{Gini}} + \widetilde{\mathrm{Smooth}} + \mathrm{ACS}_{\mathrm{IoU}} + \mathrm{AU}/K)$ uses per-method min-max normalisation over the reported methods. It is defined for completeness and ranks nothing in this paper (§4.3).

Z.2 Corrections relative to earlier versions

The main text and the appendix state each result as it now stands. Where an earlier version of this work said something different, the correction is recorded here rather than inline: first those concerning the main text, then those concerning the appendix, in appendix order, each prefixed by the part it refers to.

- *Related work (§2).* An earlier version said the untrained-encoder floor could be drawn "on any dataset, including those where nobody knows v"; the encoder can be, the floor cannot, since scoring it needs the factor values.
- *Selection term (§2).* An earlier version took the Hungarian selection component from the random-code null, where it is $1.8\times$ smaller than on the encoder null that is now used.
- *AMI (§3).* An earlier version subtracted the chance term from the numerator only, called that AMI, and concluded the adjustment was wholly inert; that form is a constant shift, Vinh's is not. An earlier version also claimed AMI took our floor to $+0.000$; that figure came from pairing coordinates by index instead of by matching, which measures a different statistic from the one we report (`analyze_floor_decomposition.py`).
- *Decomposition driver (§3).* An earlier version of `analyze_floor_decomposition.py` adjusted only the code null and obtained the three components as differences that telescope to the encoder floor by construction; the residual was identically zero because it had to be, not because anything agreed. The encoder adjustment is now measured, and the components are reported against it.
- *Shrinkage (§3).* We previously wrote that the operative floor does not shrink at all; the total does, roughly as $n^{-0.3}$ to $n^{-0.9}$, and only the AMI-surviving part stays nearly flat.
- *Canonical null (§3).* An earlier version subtracted one of three estimates of the headline cell's floor while taking its threshold from another.
- *Bias floor (§3).* An earlier version attributed the estimator-bias floor to "small K, small n"; only the second half survives.
- *Linear null (§3).* An earlier version asserted that a random linear encoder's floor is a lower bound on any architecture's; measured, it holds on one domain and fails on the other.
- *Contribution C4 (§1).* Four earlier versions of the headline verdict are withdrawn: ratios of $20\times$ and $7.5\times$, a 20-fold span, and a four-of-six count split by family (App. B.1).

Appendix material:

- *App. A.* An earlier version of the KSG sentence asserted the opposite — that KSG is k-sensitive and that Hsu et al. say so. It reverses their finding and we withdraw it.
- *App. B.1, multiplicity.* Earlier versions led with Bonferroni while the resolution grid used Holm, so the paper applied two corrections while claiming one; Holm is now the declared one everywhere.
- *App. B.1, C4.* Earlier versions of C4 reported ratios of $20\times$ and $7.5\times$, a 20-fold span and a four-of-six count split by family, each dividing by an excess its threshold could not tell from chance or subtracting a random-linear null from non-linear encoders with no method seed variance; all four are withdrawn.
- *App. B.1, floor-clearing bullet.* Two earlier versions of this bullet are withdrawn: one reported all six clearing against an uncorrected threshold whose null estimate differed from the one it was calibrated on; the next reported four of six by comparing point excesses to a threshold built from the null's spread alone, carrying no method seed variance — the construction C4 withdraws.
- *App. B.1, equivalence margin.* An earlier version of this paper chose δ by running TOST on two independent untrained encoders and requiring a low "equivalent" rate. That was a category error and we withdraw it.
- *App. B.1, margin scale.* Earlier versions called the height of the best method above the floor the triplet's "above-chance spread"; the spread is the range, which is smaller.
- *App. B.2.* An earlier version of this note called ConceptNAM vs. Level-0 β-VAE "TOST-equivalent at δ=0.05 *and* different", quoting both halves uncorrected while the table beside it prints the corrected p; we withdraw it. An earlier version of this note also named `analyze_chance_`

`floor.py` as the null row's source, and quoted the NMI spread for all three metrics, contradicting the null row of Table 1, which prints them correctly.

- *App. B.3.* An earlier version reported the Level-1 share of CBM's gain as 1.6%–34.1%, a factor of 21; its lower end was β-VAE+sup at $K{=}2$ with an excess of 0.012 against that cell's threshold of 0.029, a denominator this paper's own criterion calls indistinguishable from chance — the same defect for which we withdrew "$20\times$" in §1. An earlier version of the aggregation sentence also said "over the four cells" while quoting the corner values, and three successive versions of the task-label paragraph each gave a quantitative figure that was withdrawn by the next revision (App. Z).
- *App. B, comparability.* An earlier version said excess-over-null "is comparable across cells"; it is comparable across methods within a cell.
- *App. B.5.* Four of the seven further studies overturn a claim from our own prior revision: (i) the earlier reading that ConceptNAM wins at $\leq 0.5\%$ labels was an alignment-loss path that included unlabelled samples; (ii) the previously reported 25% noise degradation was a target-noise artifact; (iii) the prior "Level-1 is $3\times$ more OOD-robust" claim is reversed; (iv) the ablation now puts the task-only variant highest. The remaining three were unchanged in direction.
- *App. B.5, Table 5.* Through our previous revision this caption reported $n{=}1500$, 50 epochs and a different JSON, and explained the apparent n mismatch with the body table as a requirement of the semi-supervised sweep. All three were wrong and the explanation described a discrepancy that does not exist. The cell values are unchanged — they always came from the driver named in the caption — and the reproducibility manifest had the correct mapping throughout.
- *App. B.5, standard deviations.* An earlier version of the sentence introducing Tables 7–9 said the $\pm$ were over 3 seeds.
- *App. B.5, Table 8.* An earlier version of this caption described the pre-fix behaviour, in which `no_IB` left an identical KL active through the γ path and so was not an IB ablation at all; that defect is audit item C1 and the caption, not the code, was the thing still describing it. The caption also previously described the sparsity penalty as an ℓ_1 norm. An earlier version printed $+0.050$, $+0.029$ and $+72\%$ for the excesses, computed against the withdrawn 0.1143 estimate.
- *App. B.5, Table 9.* An earlier version of this caption quoted the 0.022 worst score over all 13 configurations while showing only six rows, so its stated range could not be checked against the table — the numbers were right and the presentation was not. Our previous revision also read the table as a ranking: it reported "FactorVAE variants lead by ≈ 0.02 NMI, well within the Level-1 TOST equivalence band" and cited the row order as evidence that architecture is not the dominant axis. Both readings are withdrawn — the second because the comparison was a point estimate against a margin rather than a test, the first because a 0.02 ordering among sub-chance scores is not a lead.
- *App. C.* An earlier version of the Table 10 caption asserted the 10-seed verdicts without having run the test, and a later one withdrew them for lack of an artifact. We have now run the same Bonferroni TOST family on the 10-seed block; the verdicts are identical.
- *App. D.2.* A previous revision justified the "not run" TabCBM columns by saying no implementation was bundled with the original paper or its public repository; that is false — an official implementation is public — and we withdraw it. The accurate reason is that we did not port it. We also withdraw an earlier description of our TabCBM-style baseline, which said its two penalties were "the two auxiliary regularisers described in the paper" and that its soft gate stood in for a "concept-prototype attention" mechanism and its per-concept MLP for an "embedding lookup"; checked against the authors' official implementation, all three statements are wrong.
- *App. D.2, Table 12.* An earlier version quoted the raw-NMI ratios, $2\times$ and $3.4\times$ — the ratio-against-zero this paper withdraws in C4 — and drew from them that within Level-1, architecture explains a small fraction of the cross-supervision gap. On excess the architecture span is $4.1\times$, so that conclusion does not follow and we withdraw it.
- *App. D.3.* An earlier caption of Table 14 cited two other artifacts, neither of which contains its values. An earlier version read the Pima bootstrap proportion as "a $\geq 2\times$ reduction at 68% confidence"; we withdraw it.

- *App. B.3, Table 15.* Our previous revision reported the per-cell maximum and read it as invariance; that reading is withdrawn (§5.3). An earlier revision also ran this control on five methods, omitting NBM and NODE-GAM.
- *App. E.1, Cao et al. (2022).* An earlier version of §2 attributed to them the stronger reading that their numbers imply a chance floor far from zero. We withdraw that attribution and keep the weaker, correct one.
- *App. E.1, Carbonneau et al. (2022).* An earlier version of our abstract said they "report that these metrics do not score zero on a random representation, and trace it to sample size and discretisation". Both halves were wrong: the first inverts their result and contradicted our own §2, and the second transplants their Z-max Variance explanation onto DCI. We also previously credited them with prescribing coarser discretisation as the remedy for chance inflation.
- *App. E.1, Hsu et al. (2023), Kim et al. (2018), Gröger et al. (2026).* We previously called the InfoM/InfoC normalisation "an explicit chance threshold"; we previously described the TCAV paper as building a data-dependent null; and we previously said Gröger et al. (2026) overturn a published result.
- *App. E, Table 17.* An earlier version reported the Level-1-to-Level-3 gap as $2.4\times$ wider; that figure held DetNAM fixed as the Level-1 representative in both blocks, while NBM is the stronger Level-1 method in the shipped one. A previous revision reported a null only for $K{=}4$ and warned that the other rows "should not be read as rankings"; the warning was warranted.
- *App. E, Fig. 1.* An earlier caption read the Level-1 plateau at $K{\geq}16$ as an "architectural ceiling shared across method families".
- *App. F, Table 20.* An earlier revision printed this sweep with a $k{=}3$ null only, and an earlier version of the paragraph below it read the raw column even though the caption asks for excess over the null row. We withdraw "NODE-GAM's IoU climbs with k" as a statement about NODE-GAM: the climb is the floor.
- *App. F, re-training drift.* An earlier version attributed the drift of re-derived attributions to the sweep taking attributions from models left in `train()` mode; that cannot be the cause.
- *App. F, Table 21.* An earlier version compared the 5-seed mean against the 95th percentile of single null draws — a distribution roughly twice as wide, so the rule was conservative in the direction that favours our thesis. Correcting it moves one entry, ScalingSynth $K{=}32$ DetNAM, from $n{=}1000$ to $n{=}500$; the rest hold, including MultiFactor's. An earlier version also used batch 128, which at a fixed epoch budget halves the gradient steps and under-trained every method relative to the benchmark this table characterises.
- *App. G.* An earlier version of the Table 22 caption claimed agreement with Table 1 "to within 0.02 NMI". An earlier version printed the gap narrowing as $2.4\times$, which compares DetNAM to DetNAM and so ignores that NBM leads the shipped block at this cell; the margin we previously reported was about $1.37\times$ too wide on the headline cell.
- *App. G, selection split.* A first version of this sweep selected on the test split; the criterion never touched concept labels, but choosing and reporting on one split still flatters the winner, which is the kind of bias this paper otherwise criticises. The numbers now printed are lower than the ones the selection-on-test version produced.
- *App. H.* We previously read the VAE family's near-zero R^2 as evidence that the architecture cannot consume a task label; the diagnosis (collapse) was right and that conclusion was wrong. In Table 24 we previously listed S_K for Level 1, citing Theorem 1.
- *App. H, Table 25.* An earlier caption assigned S1/S2 the $n{=}800$ floor (0.114) and S3 a borrowed cell's (0.047); under those, this table's own ConceptNAM and CBM entries would read as sub-chance, which they are not. The S2 row previously carried NMI.
- *App. I.* An earlier version printed ConceptNAM's val-stopped excess as $+0.019$, against the withdrawn 0.1143 estimate.
- *App. K.* We previously offered Cell (A)'s NAM-family spread as evidence. We previously subtracted Cell (A)'s 0.096 from Cell (C) and reported the difference as "$\approx$80% of the apparent inter-architecture gap is really supervision"; we withdraw that framing. An earlier version wrote that the residual Level-1 spread is "larger than" DetNAM's excess, by borrowing the headline cell's smaller DetNAM excess across cells.

- *App. L, Fig. 2.* An earlier version of this figure drew raw NMI from the shipped rather than the corrected configuration, printed no floor, and carried a decision tree whose lower branches we had withdrawn. All three are gone.
- *App. L, Fig. 4.* An earlier version of this figure was drawn on each method's native attribution scale and showed the ordering we retract.
- *App. M.1, Table 30.* The previous val-NMI-selected, post-hoc-merged dSprites table is superseded. This table used to quote a random-cluster baseline of NMI 0.001 as its null. We previously asserted that a randomly initialised ConvNet would score at least the linear null, so the linear figure was a lower bound; we measured it instead of asserting it. We previously read the three Level-1 rows as an ordering, then read DetNAM alone as clearing a linear null; neither survives an architecture-matched one.
- *App. M.1, Table 31.* Leakage fix: the driver previously reduced every sampled image to 32 principal components and z-scored them before the train/test split, so test images contributed to both the components and the standardisation constants they were then expressed in — the preprocessing-order violation this paper's own protocol forbids, reaching a reported table. The numbers now printed are from the corrected run. Nothing material moved: every method shifts by at most 0.007 NMI, the Level-3 / Level-1 gap is unchanged, and the three-way tie inside Level-1 reorders within its own standard deviations. We report the correction because the violation was real, not because it changed a conclusion.
- *App. O, dataset selection.* A previous revision justified this selection by attributing it to prior work, and the attribution was wrong on both counts. It said UCI Heart and Pima are "the two canonical binary-classification CBM-paper datasets of Koh et al. (2020)"; neither appears in it. It also said California Housing and Adult Income are "repeatedly used" across the NAM line; no one of the three NAM-line papers uses both.
- *App. O, Table 33.* An earlier version of this table had no null row and read the Level-1 entries as "a tight band mirroring the synthetic hierarchy"; that is the same error §3 retracts for MultiFactor. The accompanying paragraph previously described them as sitting "within a tight band of one another".
- *App. R, S1.* An earlier version read a $2\times$ DetNAM/ConceptNAM ratio off the raw values, and a later one computed $4.6\times$ against the `chance_floor.json` run of the same cell (0.074); the ratio as excess over the canonical cell null is $4.3\times$.
- *App. R, Table 36.* Our previous revision printed a CBM column of $4.0/7.6/11.2/16.4$ active dimensions and read the decline as task-driven pruning. Those figures came from the legacy `cbm_zero_padded` condition, and one of them (7.6) was in fact ConceptNAM's. An earlier version called the untrained-encoder floor's decay with K_{model} "roughly $1/K_{\text{model}}$".
- *App. R, Table 37.* An earlier caption borrowed `scaling_synth_K16_p64`'s floor (0.047 — half this block's input dimension and evaluation size) and read both Level-1 entries as sub-chance; at the correct null both clear it.
- *App. S, Assumption 1 and the proof of Theorem 1.* An earlier version required distinctness merely on some open set U and extended the conclusion off U by analytic continuation ("the constraint propagates analytically off U"). That step is not available — C^2 is not analyticity — so we now impose distinctness at the fixed point the argument actually uses.
- *App. S, Corollary 1.* An earlier version required only that g_k not be even on $\mathbb{R}$; since the theorem's invariance is local, the hypothesis must be local too.
- *App. S, Theorem 3.* Earlier versions wrote $K! - 1$ alternatives and minimised Δ over plain permutations; that is the wrong family, and the old statement was therefore not conservative. An earlier version also stated the slack bound wrongly: it claimed the slack is bounded by $\Delta/4$ under $\epsilon \le \sqrt{\Delta}/4$ using $\overline{\|u\|} \le 2B$; the Cauchy–Schwarz form is now used, and the statement survives.
- *App. S, Hungarian-aligned loss.* An earlier version argued that, because this loss is bounded above by the MSE loss, the identity-recovery guarantee carries over. That is backwards, and the claim is withdrawn.
- *App. T.* An earlier version of the Level-1 placement paragraph credited the theorems with placing ConceptNAM at Level 1 while the preceding paragraph declared them inert here; both cannot be true, and the supervision-budget reading is the one the protocol uses.

- *App. U, floor and sample size.* An earlier version of this paragraph attributed the gap partly to "few coordinates" — the small-K mechanism §3 withdraws — and quoted the removable random-code level as image benchmarks' chance.
- *App. U, why supervision dominates.* An earlier version said aligning Z with C "raises $I(Z;Y)$"; for a rotation it cannot. We previously claimed that the floor-effect control constrains this account more than it does.
- *App. U, limitations.* (2) We previously attributed the Level-1 ceiling to identifiability alone. (3) An earlier draft called Theorem 1 "global" while this limitation said the opposite. (8) An earlier version also quoted a $7.9\times$ parameter span at $K{=}32$, for which no parameter row is reported — we withdraw it. (10) An earlier heading called the scaling collapse "architectural".
- *App. V, Q1.* We previously claimed here that the grouping "predicts by how much".
- *App. V, Q4.* A previous revision argued here that the Power column confirms these are genuinely small effects rather than low power. That was circular: the two entries it appealed to read 0.05, which is the significance level, i.e. no power at all. The retracted argument ran: Power (Table 1) confirms the non-significance of within-VAE-family differences reflects genuinely small effects (Cohen's $d<0.1$), not low power.
- *App. V, Q8, Q9.* An earlier version of the Q8 answer built a K-dependent recommendation on NBM's lead under the shipped defaults. We previously read the Level-1 curves at $K\geq16$ as a ceiling, like CBM's.
- *App. V, Q13.* A previous revision reported the CBM gap as growing "by roughly a factor of $K^{1/2}$, consistent with the sample-complexity bound of Thm. 3". Neither half holds.
- *App. V, Q13b.* An earlier version claimed iCBM is the only Level-1 method that remains stable as the supervised-concept count shrinks; the claim had no artifact and is withdrawn. An earlier version said the paired bootstrap interval excludes 1 "about nine times in ten"; no artifact supported that, and the simulation shows it was wrong by an order of magnitude. We withdraw the earlier claim that the clinical variance gap is established on Heart. Our previous revision quoted 0.004 as the matched NAM-family spread, the iCBM–DetNAM gap; those two are the same model at $r{=}0$, so that number measured seed noise rather than an architectural spread.
- *App. V, Q14.* Both $\varepsilon_{\min}$ figures were computed from the raw TOST p in an earlier revision, while the sentence claimed this was "the same paired TOST used for the main table". An earlier revision also asserted the shipped answer on the corrected block without running the test. An earlier revision of this appendix reported $3/3$ equivalence at every grid point including $\varepsilon{=}0.005$. That number came from an artifact that declared equivalence whenever $|\mathrm{mean_diff}| < \varepsilon$ — a comparison of a point estimate to the margin, which is not an equivalence test and ignores sampling uncertainty. It is anti-conservative rather than conservative, and the sentence it supported was wrong; the paired-TOST numbers supersede it.
- *App. V, Q15.* An earlier version of this answer reported a result for concept-label noise. It described a $\sigma\geq0.2$ label-noise arm that no shipped artifact contains, and the target-noise axis it appeared to come from is itself a retracted experiment.
- *App. V, Q16.* An earlier revision listed only 2020–2023 work in answer to a question about 2024–2026, did not cite Bortolotti et al. (2024), and attributed the identifiability counterpart of Chen et al. (2024) to older work. An earlier revision of this answer also said we included GlanceNets and iVAE as Level-1 baselines, which was untrue.
- *App. W, Table 38.* A previous revision compared each method's attribution in whatever space it happened to expose, and we had to retract the reading: ConceptNAM "won" mainly by having an $8\times$ shorter row. On the common axis the comparison reverses; we report concentration because we previously reported the reverse ordering and drew an advantage from it.
- *App. W, floor API.* An earlier release exposed the p_{95} screen alone, so a user following the API would not have run the paper's test.
- *App. Z.* We previously wrote here that the semi-supervision gain is "universal"; for β-VAE+sup it is inside its own noise.

Table 39: Claim → artifact → driver. All commands run from repo root after a standard editable install (see supplementary `README`).

Main-text location	Artifact (under `results/`)	Driver (under `experiments/`)
Table 1 null row, C4 excesses	`null_calibration.json`	`analyze_null_calibration.py`
§3 decomposition, abstract	`floor_decomposition.json`	`analyze_floor_decomposition.py`
Table 29	`literature_chance.json`	`analyze_literature_chance.py`
App. A (Table 3)	`binning_sensitivity.json`	`analyze_binning_sensitivity.py`
§3 matched nulls	`arch_matched_null.json`	`analyze_arch_matched_null.py`
S-block nulls (Tables 37, 25)	`sblock_nulls.json`	`analyze_sblock_nulls.py`
Table 33 nulls	`realworld_null.json`	`analyze_realworld_null.py`
§3 dSprites nulls	`dsprites_null.json`, `dsprites_null_archmatched.json`	`analyze_dsprites_null.py`, `analyze_image_null_archmatched.py`
Abstract survey count	`reporting_survey.json`	`analyze_reporting_survey.py`
Table 21	`cell_resolution.json`	`exp_cell_resolution.py`
Uniform verdict rule, seed-paired test, 10-seed TOST, margin range	`verdict_unification.json`	`analyze_verdict_unification.py`
Floor cost and stopping rule (App. W)	`floor_cost.json`	`analyze_floor_cost.py`
Clinical bootstrap calibration (App. N)	`paired_bootstrap_coverage.json`	`analyze_paired_bootstrap_coverage.py`
ACS eval-mode re-extraction (after Table 20)	`acs_evalmode.json`	`exp_acs_evalmode.py`
Same-checkpoint metrics (App. P)	`same_checkpoint_metrics.json`	`exp_same_checkpoint_metrics.py`
Resolution decomposed (Table 21)	`resolution_decomposed.json`	`exp_resolution_decomposed.py`
Capacity-matched control (Limitation (8))	`capacity_matched.json`	`exp_capacity_matched.py`
Table 13	`tabcbm_official.json`	`exp_tabcbm_official.py`
External checkpoints (App. M.2)	`external_checkpoints.json`	`exp_external_checkpoints.py`
Self-paired learning gain (App. B.1)	`paired_learning_gain.json`	`exp_paired_learning_gain.py`
Verdict-rule negative control (App. B.1)	`verdict_calibration.json`	`analyze_verdict_calibration.py`
Floor under preprocessing (App. A)	`null_preprocessing.json`	`analyze_null_preprocessing.py`
Cross-fit matching (App. A)	`matching_crossfit.json`	`exp_matching_crossfit.py`
Known-MI Gaussian reference (App. A)	`gaussian_reference.json`	`analyze_gaussian_reference.py`
Fig. 2	`bench_corrected.json`, `arch_matched_null.json`	`generate_headline_corrected_figure.py`
Matched ACS null (App. F)	`acs_matched_null.json`	`analyze_acs_matched_null.py`
Null Monte-Carlo error (after Table 21)	`null_mc_error.json`	`analyze_null_mc_error.py`
§5 intervention variance	`exp14_causal.json`	`exp14_causal.py`
App. N bootstrap	`exp_clinical_bootstrap.json`	`bootstrap_clinical_std.py`
Table 28	`exp_protocol_vs_arch.json`	(analysis of prior artifacts)
App. B.1 ε-sweep	`tost_eps_sweep.json`	`analyze_tost_eps_sweep.py`
C4 verdict (App. B.1)	`matched_verdict.json`	`analyze_matched_verdict.py`
Corrected-block TOST, margin ratios, MDE	`tost_corrected.json`	`analyze_tost_corrected.py`
App. A KSG control	`ksg_control.json`	`analyze_ksg_control.py`
Table 20 null row	`acs_topk_null.json`	`analyze_acs_topk_null.py`
Table 31 matched nulls	`dsprites_tabular_matched_null.json`	`analyze_dsprites_tabular_matched_null.py`
Table 1	`exp6_full_comparison.json`	`exp6_full_comparison.py`

Table 40: Defaults used for Table 1 (synthetic $K=4$).

Category	Parameter	Value
Data	n samples / p features / K concepts	800 / 16 / 4
	features per concept / noise std	3 / 0.1
	train/test split	70/30 from `train_test_indices(seed)`
Model	d_{embed} / bottleneck hidden	8 / 256
	$\alpha, \beta, \gamma, \lambda_{\text{sparse}}$	0.001, 0.1, 1.0, 0.01
	warm-up epochs / free-bits	10 / off
These are the values the drivers pass explicitly, not the `ConceptNAMConfig` dataclass defaults, which differ (α=0.01, warm-up 20, free-bits *on*). Constructing the config without arguments therefore does *not* reproduce this table; use a driver.		
Optim	optimizer / lr / batch / epochs	Adam / 1e-3 / 64 / 50
Seeds	5 replicates	{42, 123, 456, 7, 11}

Table 41: Self-audit deltas. Each row maps to a JSON / driver under the released supplementary code (App. W).

ID	Defect class	Pre-correction	Post-correction	Fixed by
B1 (legacy)	test leakage	California R^2=0.634 (1 run)	R^2=0.626 $\pm$ 0.005 (val-selected)	`train_val_test_indices`
B2 (legacy)	invalid baseline	"UCBM+NAM strawman fails"	claim removed	AUDIT B-2 note
B3 (legacy)	train-set NMI	ConceptNAM +64–191%	TOST-equivalent	held-out NMI only
B1 (this audit)	ACS used different W per seed	DetNAM ACS-cos 0.998 "dominates CBM"	CBM ACS-cos 0.996, IoU 1.000; DetNAM ACS-cos 0.811	shared `dataset_seed` + `w_hash` assert
B3a (this audit)	"noise robustness" was target noise	ConceptNAM -4% vs others -25%	all methods flat across $\sigma \in [0.05, 1]$ on true feature noise	`feature_noise_std` on X
B3b (this audit)	OOD changed W, not covariates	Level-1 retain 3× higher OOD/ID than CBM	CBM retains 96–100%; Level-1 65–69%	fixed-W, covariate-only shift
C1 (this audit)	two duplicated KL terms	"no_IB" left identical KL via γ path	IB toggle now gates both KL paths together	concept-NAM model module
C4 (this audit)	baseline ACS used hidden weights	DetNAM/ConceptNAM looked 8× above CBM	unified saliency: CBM is most stable	shared input-attribution utility
B1-followup	exp8/10/12/14/15 still varied W per seed	legacy values in Tables 38, 5, 8, §5.2, §B.5 (i)/(v)	rerun under `dataset_seed=0`; numbers updated; new `w_hashes` assert in each driver	this audit, see §W
C2 (this audit)	"3-term loss" decomposition mathematically a 2-term one	$\alpha\, I(Z;H)$ and $\gamma \sum_k \mathrm{KL}$ multiply identical scalar (sum/mean over disjoint axes commute)	explicitly disclosed in App. Q; both branches gated by the IB toggle (Audit C1)	loss-decomposition module
C5 (this audit)	ACS top-k IoU was hard-coded to k=3	sensitivity to k unknown	sweep $k \in \{1, 2, 3, 5, 8\}$ released in supplementary	see App. F
B1-followup-2	no NODE-GAM / NBM comparators	"did you compare to all NAM-family Level-1 variants?" attack open	both ported with K-concept bottleneck and run on full benchmark; the NBM-over-DetNAM ordering did not survive the configuration audit	see App. E