MATRYOSHKA CONCEPT BOTTLENECK MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept bottleneck models predict through named attributes that can be inspected and replaced. Several readouts can give a frozen encoder different input budgets. We study joint training that makes those budgets part of learning the encoder itself. **Matryoshka Concept Bottleneck Models (MCBMs)** optimize one concept encoder with task losses on nested prefixes of a fixed concept order. Each prefix has a trained head and a stable set of concept identities. At deployment, a query policy operates within the selected prefix. On CUB, the same model reaches 80.05% accuracy with 64 concepts and 80.14% with all 112, reducing visible inputs by 42.9%. Single-seed studies on CelebA and the animal domain of LAD show similar retention at moderate budgets. Readout studies on the same CUB encoders examine a separate property: response to concept replacements. Active queries improve probability-head curves, while intervention-aware training improves raw-head responses. These results support joint multi-budget training as a practical design for named concept interfaces. They also separate task accuracy at reduced budgets from the ability of a particular head to use replacement values.

1 INTRODUCTION

Concept Bottleneck Models (CBMs) make predictions through named attributes, such as a bird's bill shape or wing color (Koh et al., 2020). These attributes give users a defined set of inputs to inspect and correct. The same task may, however, call for different concept interfaces. One deployment may use the full vocabulary, while another requires a shorter, fixed attribute list. Supporting these choices within one model would let users select a concept budget without maintaining a separate image encoder for every setting.

A simple approach is to train a standard CBM and fit separate readouts for different concept subsets. This already supports multiple budgets. Yet full-budget task training does not explicitly require each smaller subset to support prediction on its own. Readouts fitted afterward can only use the concept predictions that the frozen encoder provides. This leaves a concrete question: *can supervision at several concept budgets shape one shared encoder into useful predictors at each budget?*

Our starting observation is that useful predictions need not depend on the entire concept vector. In our CUB study, the same trained models achieve 80.05% accuracy with 64 concepts and 80.14% with all 112, averaged over three seeds. Removing 42.9% of the task head's inputs changes the mean accuracy by only 0.09 percentage points. This motivates learning useful moderate budgets together, with each budget exposing a consistent set of named concepts. The reduction concerns inputs visible to the task head; the image encoder still predicts the full vocabulary.

We introduce **Matryoshka Concept Bottleneck Models (MCBMs)** to implement this design. MCBM uses minimum redundancy maximum relevance (mRMR) to construct a fixed concept order from training annotations (Peng et al., 2005). Each task head reads a nested prefix of that order. Following the idea of Matryoshka Representation Learning (Kusupati et al., 2022), task losses from all selected prefixes update the shared encoder alongside concept supervision. Each coordinate is trained against its assigned attribute label, while several prediction tasks shape how the coordinates are used together. Thus, the chosen budgets enter encoder training itself. Figure 1 shows the architecture.

Prediction at a reduced budget and response to concept queries are separate requirements. We distinguish the *deployment budget* d, which fixes the concepts visible to a head, from the *query budget* b, which counts queried concepts within that set. A fixed deployment prefix does not require a fixed

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.

query sequence. Existing policies can select different concepts for different inputs (Chauhan et al., 2023; Shin et al., 2023). We evaluate uncertainty-based querying and CooP within each deployed prefix. Separately, we examine intervention-aware readout training, motivated by work on learning to receive corrections (Espinosa Zarlenga et al., 2023). These comparisons test how the interface works with established intervention methods.

The results show why both requirements matter. For CUB probability heads at $d = 64$, CooP improves the normalized area under the intervention accuracy curve by 6.66 percentage points over fixed-order querying of the same head. Intervention-aware training also substantially improves raw-readout responses, with small observed changes in natural accuracy. These studies reuse the same encoder sources but evaluate distinct heads, each with its own natural starting point. Additional single-seed studies on CelebA and the animal domain of LAD show similar prediction retention at moderate budgets. These results support joint multi-budget training as a practical design. They also show that budget retention alone does not establish reliable intervention response.

Our contributions are:

- **Joint multi-budget concept learning.** We train one concept encoder with task losses on nested, named prefixes, making the selected deployment budgets part of representation learning.
- **Prediction with moderate concept budgets.** We evaluate native budget retention with three-seed CUB comparisons and single-seed studies on CelebA and the animal domain of LAD.
- **Interventions within fixed interfaces.** We compare query policies and readout adaptation on the same CUB encoder sources, separating prediction accuracy from the response to concept replacements.

2 Related Work

2.1 Concept interfaces and semantic supervision

CBMs connect task predictions to named intermediate variables (Koh et al., 2020). Independent training fits the task predictor on annotated concepts; sequential training uses predictions from a frozen concept encoder; joint training also updates the encoder through the task loss. Post-hoc and label-free methods broaden how concept interfaces can be obtained (Yuksekgonul et al., 2022; Oikarinen et al., 2023). These choices affect task performance and the meaning of a correction. Continuous concept values can carry information beyond their binary labels (Havasi et al., 2022). We retain fixed concept identities and supervision while studying multiple deployment budgets. Task retention, concept alignment, and intervention response remain separate properties, and measurements of one do not establish the others.

2.2 Intervention policies and models that receive corrections

Intervention research studies which concepts to query and how to use their corrected values. Interactive CBMs introduce policies such as CooP for selecting useful queries (Chauhan et al., 2023). Intervention analysis also highlights the role of protocol choices (Shin et al., 2023). IntCEM trains models to benefit from interventions (Espinosa Zarlenga et al., 2023), while stochastic CBMs model concept dependencies when propagating corrections (Vandenhirtz et al., 2024). Minimal studies representation constraints for prediction and intervention (Almudévar et al., 2026). These methods address distinct parts of the correction process. MCBM defines the concept set visible at a chosen deployment budget. An existing policy can then operate inside that set. We evaluate this combination directly and distinguish query count, correction accuracy, and harm, rather than treating a smaller interface as sufficient evidence of better interventions.

2.3 Nested representations and concept selection

Matryoshka Representation Learning trains prefixes of a shared representation for several output dimensions (Kusupati et al., 2022). Feature-selection methods choose informative variables; mRMR

balances target relevance against redundancy among selected features (Peng et al., 2005). Several prefix readouts can also be fitted to one frozen encoder. MCBM evaluates the additional choice of updating that encoder through multiple prefix losses, while each coordinate retains a fixed annotation target and correction rule. The resulting interfaces can be specified before deployment. Every sample at budget d exposes the same concept names, although its query sequence can differ. We compare this joint training design with native and refitted predictors, and evaluate query policies within each fixed interface.

3 Matryoshka Concept Bottleneck Models

3.1 A Concept Interface with Two Budgets

Let x be an image, $y \in \{1, \dots, C\}$ its class, and $c \in \{0, 1\}^K$ its concept-supervision targets. A concept encoder g_θ predicts logits $z = g_\theta(x) \in \mathbb{R}^K$. Each coordinate has a fixed concept identity and annotation target; its predicted value remains continuous. A permutation π defines an ordered vector $z_\pi = (z_{\pi_1}, \dots, z_{\pi_K})$. We choose deployment budgets from a set $\mathcal{D} \subseteq \{1, \dots, K\}$. At budget d, the task head receives only $z_{\pi,1:d}$. Its visible set $S_d = \{\pi_1, \dots, \pi_d\}$ is the same for every image.

The query budget b is a different quantity. It counts queried concept coordinates inside S_d for one image, with $0 \le b \le d$. A query may confirm an already correct concept. Choosing d changes the task interface. Increasing b replaces values inside that fixed interface. We keep the selected head fixed throughout a query curve. This prevents a change of head from being mistaken for a benefit of a correction.

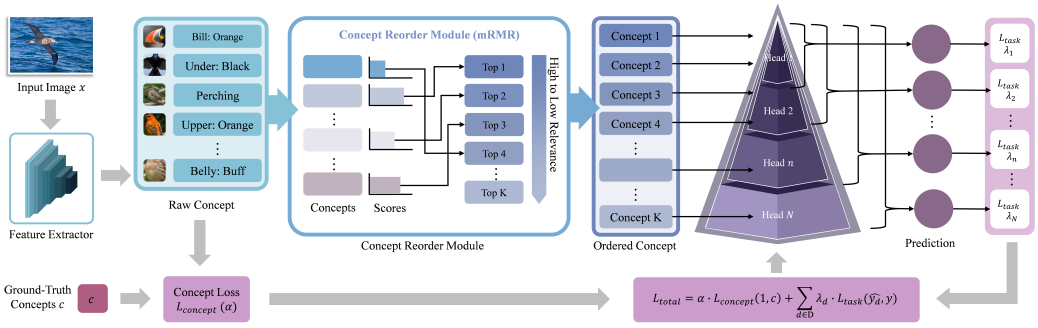

Figure 1: **Matryoshka Concept Bottleneck Models Architecture.** A fixed mRMR order routes raw concept logits to nested task heads. All heads and the encoder train jointly. The diagram uses λ_d for w_d in Eq. 3.

3.2 A Shared Concept Order

We use mRMR to build nested sets from training annotations (Peng et al., 2005). The first concept maximizes its estimated mutual information with the class. After selecting S_{t-1}, the next concept is

$$\pi_t = \arg\max_{j \notin S_{t-1}} \left[\widehat{I}(c_j; y) - \frac{1}{|S_{t-1}|} \sum_{k \in S_{t-1}} \widehat{I}(c_j; c_k) \right], \qquad t > 1. \tag{1}$$

The first term favors useful concepts; the second reduces repeated information. Mutual information is estimated from discrete training labels, without validation or test labels. The resulting order is fixed before image training. It is a greedy feature-selection rule, not a learned permutation or a causal order. Its role is to provide one consistent set of prefixes. Ordering controls test whether it is preferable to other choices in the evaluated settings.

3.3 Joint Training at Several Deployment Budgets

Each budget has a linear task head with its own parameters:

$$q_d(y \mid x) = \mathrm{softmax}(W_d z_{\pi,1:d} + a_d), \qquad W_d \in \mathbb{R}^{C \times d}. \tag{2}$$

All heads share the image encoder. We train them with

$$\mathcal{L}_{\text{joint}} = \alpha \mathcal{L}_c + \sum_{d \in \mathcal{D}} w_d \ \text{CE}(q_d, y), \qquad w_d \geq 0, \quad \sum_{d \in \mathcal{D}} w_d = 1. \tag{3}$$

Here $\mathcal{L}_c$ is binary cross-entropy on annotated concept coordinates. Missing annotations, when present, are masked rather than treated as negative labels. The coefficient α controls concept supervision. We use equal head weights in the main model. The encoder and the task heads receive gradients; the concept order does not. Equation 3 describes the main branch; the training-only Inception auxiliary branch applies the same objective with the coefficient given in Appendix A.

Several prefix heads can also be fitted after a conventional encoder has been trained. In MCBM, their task losses update the shared encoder during image training. The selected budgets therefore enter representation learning itself. Early coordinates receive task supervision from more heads, while larger heads can use later concepts. This specifies the coupling between budget tasks; it does not make joint training the only way to obtain multiple readouts. We evaluate native heads and frozen-encoder refits as separate comparisons.

More precisely, the task gradient at ordered coordinate j is

$$\frac{\partial \mathcal{L}_{\text{task}}}{\partial z_{\pi_j}} = \sum_{d \in \mathcal{D}:\, d \geq j} w_d \frac{\partial \, \text{CE}(q_d, y)}{\partial z_{\pi_j}}. \tag{4}$$

Only heads that can see a coordinate contribute to its task gradient. With six equal-weight heads, the first eight coordinates serve all six tasks, while coordinates beyond 64 serve only the full head. The gradients can differ in direction. The equation explains how the training objectives share the encoder; empirical comparisons assess the resulting predictors.

Once trained, any $d \in \mathcal{D}$ can be selected without further image training. The chosen interface fixes the permitted concept inputs in advance, as a fixed attribute checklist would. Query policies may then adapt within that set. The full encoder remains active. Our deployment budget measures visible inputs, while the independent linear heads require $C \sum_{d \in \mathcal{D}} d$ weights, excluding biases. Figure 1 shows the shared encoder, fixed permutation, and nested task heads.

3.4 Querying inside a deployed prefix

Let u_d denote the representation used by a deployed head. For a probability head, $u_d = \sigma(z_{\pi,1:d})$. For a raw head, $u_d = z_{\pi,1:d}$. Let $A_b \subseteq \{1, \ldots, d\}$ contain the b queried positions. An intervention replaces only those positions:

$$\big[T(u_d, c, A_b)\big]_j = \begin{cases} v_j(c_{\pi_j}), & j \in A_b, \\ (u_d)_j, & j \notin A_b. \end{cases} \tag{5}$$

For probability heads, $v_j(0) = 0$ and $v_j(1) = 1$. For raw heads, these values are the fifth and ninety-fifth percentiles of the training logits for that concept. This keeps replacements in the units received by the head. Both percentiles are estimated from training predictions and fixed for evaluation. They need not lie on opposite sides of zero, so raw replacement does not guarantee repair of a binary concept error. The two representations require separately fitted heads and have separate natural prediction baselines.

We compare three query rules. *Fixed order* queries the next visible concept in the mRMR prefix. *Entropy* selects the most uncertain unqueried concept. *CooP* combines concept uncertainty with an estimate of the effect of a replacement on the task prediction (Chauhan et al., 2023). We adapt the released implementation to the deployed prefix. Its score uses expected absolute change, as in that implementation. Neither the true class nor an unqueried concept label is available to the selection rule. All queries have unit cost. The CooP weight is selected on validation curves before test evaluation; Appendix A gives the grid and tie rule.

Restricting a policy to S_d preserves the deployed interface. It does not require the policy to follow π within that set. This distinction lets us test a global deployment order and an input-dependent query order without changing both at once. We use complete curves, including cases in which further corrections reduce accuracy. We do not stop when a prediction first matches its true label, since this information is unavailable during deployment.

3.5 TRAINING A READOUT TO RECEIVE CORRECTIONS

A task head trained only on predicted concepts may respond poorly to replacements. We test a simple adaptation motivated by intervention-aware learning (Espinosa Zarlenga et al., 2023). The image encoder remains frozen. We fit raw readouts with both natural and intervened inputs:

$$\mathcal{L}_{\text{aware}} = \sum_{d \in \mathcal{D}} w_d \left[(1-\beta)\, \text{CE}(h_d(u_d), y) + \beta\, \mathbb{E}_A\, \text{CE}\left(h_d(T(u_d, c, A)), y\right) \right]. \tag{6}$$

For every training example, step, and head, we sample b uniformly from $\{1, \ldots, d\}$ and set $A = \{1, \ldots, b\}$. A separate natural branch preserves supervision at $b = 0$. We evaluate $\beta \in \{0, 0.1, 0.5\}$ and give the same adaptation to the strong baselines. The $\beta = 0$ readout is the natural-training control. Nonzero settings share the intervention sampling stream within each seed and head.

This experiment tests adaptation to a specified replacement distribution while keeping the concept encoder fixed. We apply the same training to MCBM and the baselines, and evaluate all raw heads with fixed-prefix replacement. The probability-head query study evaluates a different route. Each route uses its own head and natural prediction baseline.

4 EXPERIMENTS

We test four questions: whether native heads preserve prediction quality at a reduced concept budget; how design and representation choices affect prediction; whether active queries work within a fixed prefix; and whether a readout can learn to use queried values. This separates budget retention from the operational response of a particular head and query policy.

4.1 EXPERIMENTAL SETUP

Data and budgets. CUB-200-2011 (Wah et al., 2011) is our main benchmark, with 200 classes and a 112-concept vocabulary. We train heads at $d \in \{8, 16, 24, 32, 64, 112\}$ and emphasize $d = 64$ and 112. The former retains 57.14% of the vocabulary. Intervention answers are training-derived class attribute prototypes. Our curves measure task responses to these replacements, rather than independent image-specific human corrections. All task results use the official test split. Ranking, calibration, and readout fitting use training data; query-policy selection uses a separate validation split. The test split has been observed during development, so these are benchmark observations rather than a new untouched confirmation study.

We also test natural prediction on CelebA (Liu et al., 2015), using *Attractive* as the binary target and excluding it from the remaining 39 concepts, and on the animal subset of LAD (Zhao et al., 2019). The latter covers 50 classes and 123 attributes, with 560 directly annotated training images, 140 validation images, and 3,960 test images. These are single-seed dataset studies. They do not constitute full-LAD or multi-dataset intervention evaluations.

Models and comparisons. The visual experiments use an Inception-v3 concept encoder. MCBM shares this encoder across nested task heads; FullJoint trains a full-budget task head; Joint64 trains a dedicated 64-concept task head. The current CUB native-head and intervention studies use final checkpoints after 100 epochs, with a learning-rate reduction at epoch 50. Their FullJoint and Joint64 sources use initial learning rate 0.02. We compare these reported configurations; training and protocol details appear in Appendix A.

Table 3 separates the encoder cohorts and head types. FullJoint at $d = 64$ uses a fitted prefix readout, not its native 112-input classifier with missing coordinates. Fitting and query rules are shared within each comparison. CooP (Chauhan et al., 2023) selects queries inside the deployed prefix; its coefficient is chosen from a fixed eight-value grid using validation data only. Entropy and fixed-order queries provide policy controls. The appendix describes the separate native evaluations of IntCEM, SCBM, and Minimal (Espinosa Zarlenga et al., 2023; Vandenhirtz et al., 2024; Almudévar et al., 2026), whose training and query units differ from this comparison.

Measurements. CUB summaries use seeds 101, 102, and 103 and report the mean and sample standard deviation. Accuracy is measured at every integer query budget $b \in \{0, \ldots, d\}$ while holding the deployed head fixed. We compute nAUC by trapezoidal integration over this complete curve,

divided by d. Thus, nAUC summarizes performance within a deployment budget; nAUC at $d = 64$ and 112 does not represent equal absolute query expenditure. Repair and harm are conditional on initially wrong and initially correct predictions, respectively. Curves retain decreases and reversals. No cumulative maximum or first-success stopping rule is used.

4.2 RQ1: Does a shared model retain performance with fewer concepts?

Table 1: Native accuracy (%). CUB reports three-seed mean±SD; the other datasets use one seed. A dash marks an unrun comparison.

		MCBM native		Separate models	
Dataset	d/K	Reduced d	Full K	Joint-d	FullJoint
CUB	64/112	$80.05 \pm .25$	$80.14 \pm .35$	$79.29 \pm .33$	$79.61 \pm .28$
CelebA	20/39	71.97	70.21	—	66.86
LAD animals	64/123	72.15	72.10	—	65.53

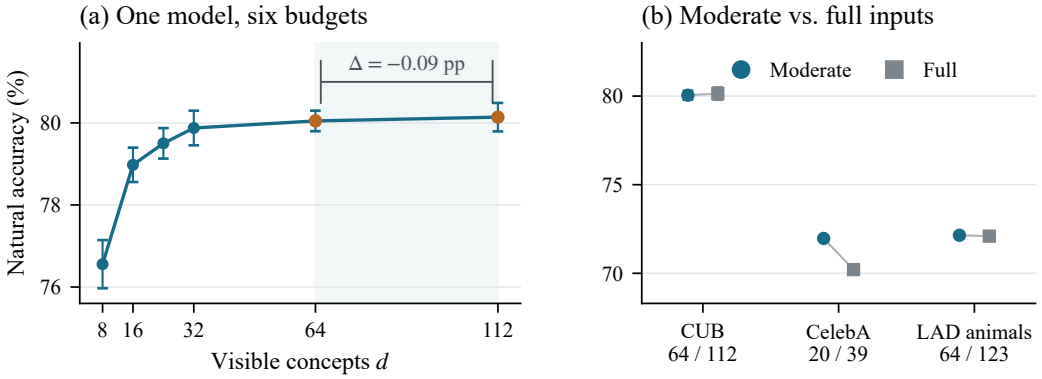

Figure 2: Native budget retention. (a) All six trained CUB heads, with three-seed SD. (b) Moderate and full inputs across datasets; only CUB has multi-seed error bars. Lines connect measured budgets, and the panels use different accuracy ranges.

MCBM *preserves its own full-budget accuracy* with substantially fewer task-head inputs (Table 1, Figure 2). On CUB, moving from 112 to 64 concepts changes accuracy by -0.092 ± 0.132 percentage points across paired seeds. Both native heads belong to the same trained model, so selecting the smaller interface requires no further fitting. This establishes native budget retention. Very small heads remain useful diagnostics, but our practical comparison is near half of the vocabulary.

Joint64 provides a separate budget-specific training alternative. In Table 1, MCBM is higher by 0.76 points at $d = 64$ than Joint64 and by 0.54 points at $d = 112$ than FullJoint. Its within-model comparison separately measures the effect of reducing the visible interface. The reported gap describes paired seed outcomes; we do not make a statistical equivalence claim.

The two additional datasets also retain full-budget accuracy with moderate inputs. LAD's $+0.051$-point difference corresponds to two more correct predictions out of 3,960 test images. Both studies use one seed, so they provide limited evidence about stability. These results concern visible inputs, not measured savings in encoder computation, training annotations, or human time.

4.3 RQ2: How do design and representation choices affect prediction?

Earlier control cohorts examine subset constraints, concept order, and numeric representation. In a finite subset-selection experiment, a concept-only encoder supplies the same predictions to all candidates. We fit 36 distinct subsets drawn from seven orders and six budgets, then use validation data to select one global order or separate subsets at each budget. The global mRMR order obtains

73.455% at $d = 64$, versus 72.730% for independent budget-wise selection. Relaxing the shared-order constraint does not improve test accuracy in this candidate pool. The result supports the feasibility of useful nested subsets, without establishing zero nesting cost or optimal selection.

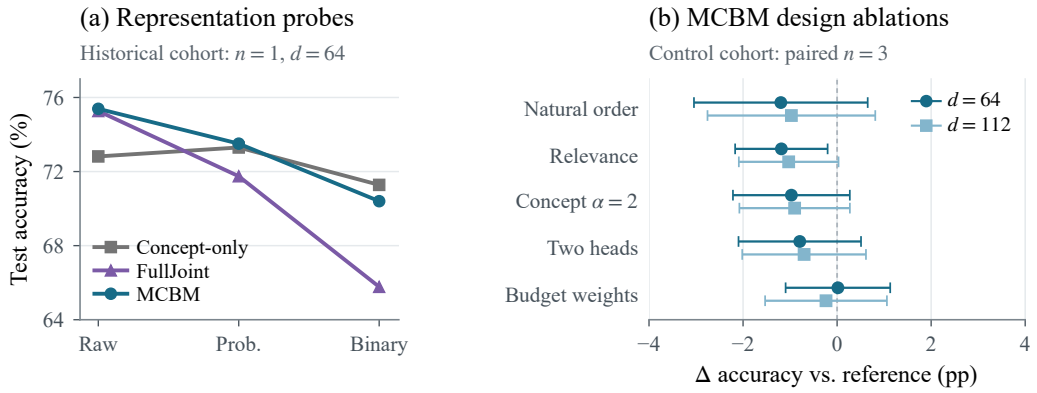

Figure 3: Separate CUB controls: (a) matched representations of earlier single-seed encoders at $d = 64$; (b) paired changes from the constant-rate, six-head MCBM reference, with three-seed SD. Neither panel uses the optimized native-model cohort.

The constant-rate control cohort tests relevance and natural orders, concept-loss weighting, a two-head variant, and budget-proportional head weights. None consistently improves both native budgets, 64 and 112, over that cohort's six-head reference (Figure 3b; Appendix B). These comparisons explain design choices within their training setting. They do not establish a universally best order; an early low score for a random prefix also cannot establish that random ordering must collapse.

The historical matched readouts in Figure 3a show representation dependence. For that MCBM encoder, converting raw logits to binary concepts reduces $d = 64$ accuracy from 75.39% to 70.40%; at $d = 112$, the values are 75.35% and 70.85%. Probability inputs fall between them. A fixed 128-unit nonlinear probe does not outperform the longer-trained linear raw readout. These finite probes, together with the source-specific concept diagnostics in the appendix, characterize earlier encoders; they do not certify the semantics of the current native models. RQ3–4 instead test operational responses using readouts of the current encoder sources. Task prediction, concept alignment, and response to replacements remain separate measurements.

4.4 RQ3: Can active queries operate within a fixed deployment prefix?

Table 2: Probability-head performance (%, three-seed means). Each row uses one head across all policies. FullJoint at $d = 64$ uses a prefix refit.

			Intervention nAUC		
Model	d	Natural	Fixed order	Entropy	CooP
MCBM	64	76.44	86.27	92.58	92.93
FullJoint	64	73.86	87.35	87.35	87.80
Joint64	64	74.52	85.89	85.96	86.32
MCBM	112	77.46	91.84	95.55	95.91
FullJoint	112	75.85	90.83	91.27	91.58

FullJoint prefix readouts form the post-hoc comparison: one encoder is trained for full-budget prediction and its frozen outputs support several refitted heads. Both families use the same representation format, fitting procedure, and query rules.

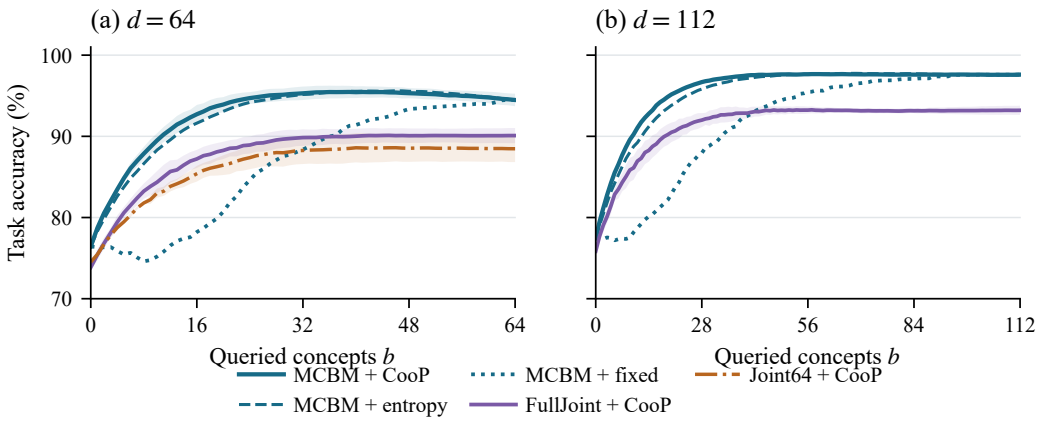

Figure 4: Fixed-prefix query curves, measured at every integer b. Lines show three-seed means; CooP bands show sample SD. Every curve starts from its own probability head. The panels cover different absolute query costs.

For MCBM, the nAUC gains from CooP over fixed-order queries are 6.661 points at $d = 64$ and 4.076 at $d = 112$ (Table 2, Figure 4). These comparisons hold the head and visible concept set fixed. Entropy accounts for most of the gain; CooP adds another 0.36–0.37 points.

Under the same CooP protocol, the paired MCBM nAUC differences are $+5.129 \pm 1.089$ points against FullJoint and $+6.610 \pm 1.551$ against Joint64 at $d = 64$. At $d = 112$, the difference against FullJoint is $+4.334 \pm 0.566$. All three seeds favor MCBM for these encoder sources. This comparison assesses the learned inputs through a common readout and query procedure.

With fixed-order queries, MCBM at $d = 64$ instead has nAUC 1.083 ± 0.949 points below FullJoint, with a negative difference in each seed. More queries can also reduce task accuracy: MCBM+CooP reaches 95.300% after 32 queries in the 64-concept prefix, compared with 94.477% after all 64. Its mean maximum conditional harm is 4.334%. Complete curves and harm therefore reveal behavior that a high endpoint would miss, consistent with broader analyses of intervention procedures (Shin et al., 2023).

4.5 RQ4: Can a readout learn to use queried concept values?

A raw readout can predict well before intervention yet respond poorly to replaced inputs. Without intervention training, the current MCBM raw heads have natural accuracies of 79.00% and 79.43%, but nAUC values of only 47.67% and 62.72% (Figure 5). *Good natural prediction does not establish a useful response to queried values.*

We freeze each encoder and fit raw readouts with $\beta \in \{0, .1, .5\}$. Training uses the percentile replacements from Section 3.4 and 1,000 full-batch Adam steps at learning rate 0.001. The zero-weight head is the natural-training control. All weights are reported, with no test-based selection.

At $\beta = .1$, MCBM improves $d = 64$ nAUC by 45.479 ± 1.174 points, while natural accuracy changes by $+0.058 \pm 0.251$ points. Raising the weight to .5 further improves the curve but reduces natural accuracy by 0.276 ± 0.301 points relative to the zero-weight head. At $d = 112$, $\beta = .5$ reaches nAUC 96.72% and a 100% full-query endpoint under the prototype targets. The readout can therefore learn a much better task response to these calibrated replacements with a small observed change in natural accuracy.

The same training also helps the baselines. At $\beta = .5$, MCBM retains higher natural accuracy, but its $d = 64$ nAUC is 0.802 points below FullJoint and 0.251 below Joint64. At $d = 112$, its nAUC is 0.024 points below FullJoint. The result is a trade-off rather than a nesting-specific advantage, and a perfect endpoint does not remove intermediate harm. These raw readouts and the probability+CooP study use different heads; their best numbers cannot be combined into one model result.

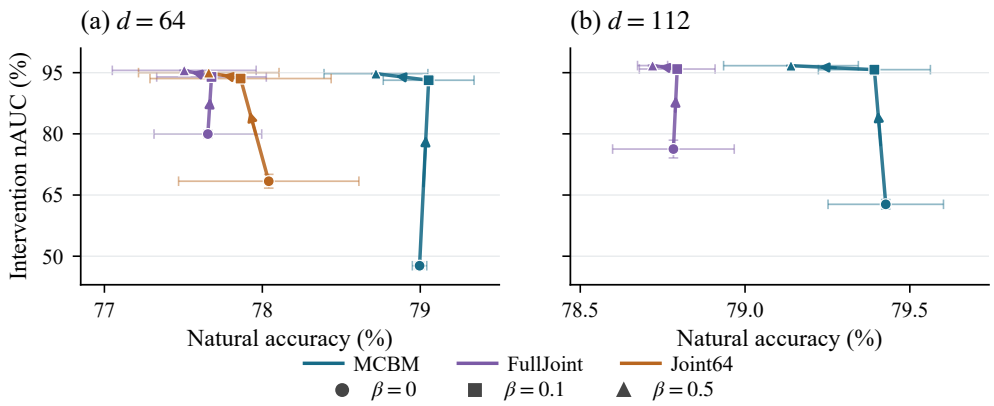

Figure 5: Natural accuracy and task response to calibrated replacements in raw readouts. Points and error bars show three-seed mean±SD. Arrows connect increasing β, not training steps. The panels share the nAUC scale but use different accuracy ranges. Full values are in Table 6.

5 Discussion and Limitations

Our results concern named, continuously valued concept inputs at predefined deployment budgets. Native predictions measure budget retention. Separate readout studies on the same CUB encoders measure response to specified replacements. Historical representation and concept-quality diagnostics explain design choices and are kept distinct from evidence about the current encoders. This distinction prevents task accuracy from standing in for concept alignment or semantic purity.

Table 3: Roles of the evidence. Current-encoder results and earlier diagnostics answer different questions. Refitted heads are kept separate from native heads.

Evaluation	Encoder cohort	Evaluated head
Budget retention	Main CUB / pilots	Native
Query policies	Main CUB	Probability refit
Replacement training	Main CUB	Raw refit
Design diagnostics	Earlier CUB cohorts	Native / matched probes

The image encoder still predicts all concepts; visible-input reduction is not a measured reduction in total computation or human time. CUB uses class-prototype answers, while CelebA and LAD provide single-seed natural-prediction studies. The test sets have been evaluated during development, so the results describe the reported configurations rather than a fresh confirmation study. Within this scope, query policies and readout training change intervention quality, including harm. Appendix E explains why replacing values in a trained head differs from adding information to an ideal predictor.

6 Conclusion

MCBM makes predefined concept budgets part of encoder training through joint losses on nested heads. On CUB, the 64-input native head retains accuracy close to the 112-input head of the same model. Readout comparisons then evaluate how the learned inputs support active queries and replacement-aware training. The results establish useful operating points for this design and distinguish budget retention from intervention response. They leave open how far these benefits extend to other concept annotations and deployment constraints.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, literature discovery, and limited code development and debugging. All suggested references were verified by the authors against the original publications, and all AI-assisted code was manually reviewed and tested for correctness. Generative AI tools were not used to generate experimental data, fabricate results, or autonomously determine the scientific claims or conclusions of this work. The authors designed the methodology, conducted the experiments, analyzed the results, and verified all reported findings. We take full responsibility for the final content of this work, including any text, code, or other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

We use existing research datasets and do not conduct a human study. The CelebA task predicts a dataset label for facial attractiveness; it should not be read as an objective measure of a person's value or used for decisions about individuals. Concept interfaces can expose prediction errors, but they do not by themselves ensure fairness or safe deployment. Our simulated corrections should not be treated as evidence of clinical or other high-stakes reliability.

REFERENCES

Antonio Almudévar, José Miguel Hernández Lobato, and Alfonso Ortega. There was never a bottleneck in concept bottleneck models. In *International Conference on Learning Representations*, volume 2026, pp. 114146–114165, 2026.

Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In *Proceedings of the aaai conference on artificial intelligence*, volume 37, pp. 5948–5955, 2023.

Mateo Espinosa Zarlenga, Katie Collins, Krishnamurthy Dvijotham, Adrian Weller, Zohreh Shams, and Mateja Jamnik. Learning to receive help: Intervention-aware concept embedding models. *Advances in Neural Information Processing Systems*, 36:37849–37875, 2023.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. *Advances in Neural Information Processing Systems*, 35:23386–23397, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *International conference on machine learning*, pp. 31504–31520. PMLR, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Moritz Vandenhirtz, Sonia Laguna, Ričards Marcinkevičs, and Julia E Vogt. Stochastic concept bottleneck models. *Advances in Neural Information Processing Systems*, 37:51787–51810, 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

Bo Zhao, Yanwei Fu, Rui Liang, Jiahong Wu, Yonggang Wang, and Yizhou Wang. A large-scale attribute dataset for zero-shot learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 398–407. IEEE, 2019.

A PROTOCOLS AND ADDITIONAL DIAGNOSTICS

A.1 DATA AND MODEL IDENTITIES

CUB-200-2011 has 200 classes (Wah et al., 2011). Our fixed protocol uses 112 canonical concepts, 4,796 training images, 1,198 validation images, and 5,794 official test images. Concept supervision and intervention answers use training-derived class-attribute prototypes. These answers are not independent per-image human corrections. Sorting and calibration use training data only. Concept IDs, sample IDs, source checkpoints, and permutations are retained across exported representations. The test set has been observed in earlier project experiments; the present comparisons are not a fresh holdout study.

CelebA (Liu et al., 2015) uses Attractive as the binary target and excludes this attribute from the concept vocabulary, leaving 39 concepts. The official train/validation/test counts are 162,770/19,867/19,962. The trained budgets are $\{4, 8, 16, 20, 39\}$. The LAD experiment (Zhao et al., 2019) is an animal-domain pilot with 50 classes and 123 attributes, not the complete 359-attribute benchmark. Its directly annotated development population has 560 training and 140 validation images; natural accuracy uses 3,960 official animal test images. Only 300 test images have direct concept labels. Missing annotations remain masked and do not become negative labels. Both cross-dataset studies use seed 101 and are reported without multi-seed uncertainty estimates.

The image models use an ImageNet-pretrained Inception-v3 backbone (Szegedy et al., 2016), 299-pixel inputs, batch size 64, and a fixed 100-epoch horizon. SGD uses momentum .9 and weight decay 4×10^{-5}; the auxiliary branch coefficient is .4. The current CUB sources use ImageNet input normalization and a factor-.1 StepLR schedule at epoch 50. Cross-dataset pilots retain their separately frozen preprocessing and constant-rate recipe; their absolute accuracies are not a matched extension of the latest CUB optimization study. We evaluate the final checkpoint. No checkpoint is chosen by test accuracy.

MCBM has six CUB heads at $\{8, 16, 24, 32, 64, 112\}$ with equal task weights. FullJoint has a native 112-concept task head; Joint64 is trained with a 64-concept task head. A FullJoint-to-64 readout is fitted on the frozen encoder's visible prefix. It is not a truncation of the native 112-head. Native, probability-refitted, and raw-refitted results keep separate model identities and natural prediction points. All encoders still produce the full concept vector.

CUB preparation and ordering. We sort the 5,994 official training image IDs and apply one class-stratified shuffle split with 1,198 validation samples, random seed 101, and scikit-learn 1.7.2. Each class prototype is then computed from the 4,796 training images only. Certainty level 1 is unobserved; levels 2–4 contribute binary votes. A nonempty tied vote is assigned 1; a class–concept pair with no votes remains unknown. The retained CUB vocabulary has no such unknown pair. Canonical columns preserve the following zero-based indices in the original 312-attribute vocabulary:

> 1, 4, 6, 7, 10, 14, 15, 20, 21, 23, 25, 29, 30, 35, 36, 38, 40, 44, 45, 50, 51, 53, 54, 56, 57, 59, 63, 64, 69, 70, 72, 75, 80, 84, 90, 91, 93, 99, 101, 106, 110, 111, 116, 117, 119, 125, 126, 131, 132, 134, 145, 149, 151, 152, 153, 157, 158, 163, 164, 168, 172, 178, 179, 181, 183, 187, 188, 193, 194, 196, 198, 202, 203, 208, 209, 211, 212, 213, 218, 220, 221, 225, 235, 236, 238, 239, 240, 242, 243, 244, 249, 253, 254, 259, 260, 262, 268, 274, 277, 283, 289, 292, 293, 294, 298, 299, 304, 305, 308, 309, 310, 311.

The fixed mRMR order uses unsmoothed empirical joint counts, natural logarithms, and uniform training-sample mass. Relevance uses observed concept–class pairs; redundancy uses jointly observed concept pairs. Equal selection scores are resolved by lexicographic concept ID. This defines the ranking before image training; validation and test predictions do not enter its construction. For the CUB class prototypes, c_j is a deterministic function of y, so $I(c_j; y) = H(c_j)$ on the training distribution. Here mRMR balances variation across prototype attributes against their pairwise redundancy. The image encoder still has to predict these targets from images.

A.2 FIXED-PREFIX QUERY EVALUATION

We invoke the published CooP implementation (Chauhan et al., 2023) from Google Research commit `e6890f8`. The deployment prefix is fixed before evaluation. Its columns are projected into canonical order, and a separate permutation restores the trained head's column order. The policy cannot inspect hidden suffix predictions, unqueried concept answers, or the task label. Every query reveals one previously unqueried scalar concept. The same deployed head is used for every $b = 0, \ldots, d$.

Probability readout fitting. The frozen encoder supplies sigmoid-transformed training logits. We fit fresh linear heads at $\{8, 16, 24, 32, 64, 112\}$ for 1,000 full-batch Adam steps, with learning rate .001, moments $(.9, .999)$, $\epsilon = 10^{-8}$, zero weight decay, and AMSGrad disabled. The objective is task cross-entropy with uniform sample mass, head weights $1/6$, total task coefficient one, and no intervention term. There is no hidden layer. For model seed $s \in \{101, 102, 103\}$, head d is initialized with seed $s + d$ using the linear layer's uniform weight and bias initialization on $[-1/\sqrt{d}, 1/\sqrt{d}]$. The last step is used. The encoder, ranking, and concept IDs remain fixed; no head is fitted on validation or test data. These existing heads are exported unchanged for query evaluation.

Probability-head queries use binary 0/1 replacement, without percentile calibration. CooP combines concept entropy with expected absolute change in the current predicted class probability. This follows the public code's $\mathbb{E}[|\Delta|]$, rather than substituting $|\mathbb{E}[\Delta]|$. Entropy has coefficient one, scalar cost is one, cost weight is zero, and no score normalization is added. For each source and budget, $\lambda \in \{0, .1, .3, 1, 3, 10, 30, 100\}$ is selected by the full validation accuracy-curve area. Exact integer trapezoidal numerators determine ties, which favor smaller λ. Selection is frozen before test access. The selected policy, entropy-only policy, and fixed mRMR order are evaluated; if $\lambda = 0$, the identical selected/entropy trajectory is reused.

This is a scalar, fixed-prefix adaptation of the released implementation. We compare the configurations named in each experiment under the same downstream fitting and query protocol.

A.3 INTERVENTION-AWARE RAW READOUTS

Nine encoders are frozen: three families and seeds 101, 102, and 103. For each encoder, the six raw-logit linear heads use Adam with learning rate .001, zero weight decay, and 1,000 full-batch steps. Head weights are $1/6$ and sample weights are uniform. We retain all $\beta \in \{0, .1, .5\}$, using

$$\mathcal{L}_\beta = \frac{1}{6} \sum_d \left[(1 - \beta)\, \mathrm{CE}(h_d(z_{:d}), y) + \beta\, \mathrm{CE}(h_d(\widetilde{z}_{:d}), y) \right].$$

For every step, example, and head, the queried count is sampled uniformly from $1, \ldots, d$ and the corresponding fixed prefix is replaced. Each head uses a separate generator seeded by the model seed plus d. The two nonzero-β fits share the same draw stream. Beta-zero parameters and loss histories are reproduced exactly before the existing scientific model is reused. Thus the batch contains 18 new fits, not 27 new source models.

For concept j, a binary answer maps to that encoder's training-set 5th or 95th raw-logit percentile. Calibration is unconditional and per concept. Percentiles are not adjusted using validation/test data or forced across zero. A query therefore need not resolve a thresholded concept error. Task accuracy and nAUC measure response to this replacement distribution; they do not count every queried concept as repaired. Both the raw natural branch and its corrected branch use the same head. Complete validation and test curves cover 64/112 concepts for MCBM and FullJoint and 64 for Joint64. This fixed-query raw route is separate from probability+CooP; their best numbers are never assembled into a single model.

A.4 METRICS, EXTERNAL PROTOCOLS, AND VERIFICATION

Normalized AUC is the trapezoidal accuracy area over the complete grid $b = 0, \ldots, d$, divided by d. It is reported as a percentage. Repair is conditional on an initially wrong prediction; harm is conditional on an initially correct prediction. Empty denominators are undefined, not zero. We also retain exact query counts, prediction flips, and corrected/new concept-error counts. Curves use cur-

rent predictions, with neither cumulative maxima nor label-dependent stopping. Equal normalized positions at $d = 64$ and $d = 112$ imply different absolute query costs.

External methods retain their native intervention mechanisms. IntCEM (Espinosa Zarlenga et al., 2023) uses ResNet-18 (He et al., 2016), 16-dimensional concept embeddings, seed 42, and 150 epochs. Its 28-group actions are projected by actual scalar acquisition costs; unused allowance is preserved. SCBM (Vandenhirtz et al., 2024) uses ResNet-18, 100 Monte Carlo samples, seed 42, and 300 epochs. Its existing random and probability-uncertainty curves are source-verified reuse, with the original output rounding. Minimal (Almudévar et al., 2026) uses its original Inception model/loss, seed 42, 250 epochs, and 112 singleton concepts. Unknown interventions remain unknown; a revealed binary value uses its native latent mapping $6c - 3$. Confidence and mRMR query orders use the same checkpoint. These different training and query protocols support contextual comparisons, not a strict pooled leaderboard.

Saved logits, predictions, query identities, and curve metrics were independently recomputed. Cross-precision argmax differences are retained rather than corrected by altering saved predictions. Means use three seeds and sample SD, not confidence intervals. No significance or noninferiority claim is made. Test-access events are recorded and identical reused evaluations are deduplicated.

A.5 Limits of mechanism diagnostics

An earlier single-seed frozen concept-only encoder compared seven orderings over six budgets, yielding 36 distinct fitted subsets. At $d = 64$, the global mRMR prefix achieved 73.455% versus 72.730% for validation-selected independent subsets. This finite search did not reveal a benefit from relaxing nesting; it does not prove zero nesting cost or global subset optimality. On an earlier MCBM encoder, 5,000-step readouts gave raw/probability/hard accuracies of 75.39/73.51/70.40% at 64 and 75.35/73.77/70.85% at 112. A width-128 ReLU head did not outperform the linear probe. These bounded optimization and representation checks are not Bayes-risk estimates, convergence proofs, or evidence that all retained information is semantic. They are distinct from the latest encoder results.

B Additional budget, design, and intervention results

B.1 All native MCBM budgets

Table 4 reports every trained head of the MCBM configuration used for the main natural-prediction result. Each column of observations is derived from the same three final checkpoints. These are native heads, not probability or intervention-aware refits. The smaller heads remain available, but the main deployment comparison concerns 64 versus 112 concepts. Increasing the budget improves the observed mean in this configuration; nesting alone does not require this ordering for other trained models or every individual example.

Table 4: All native CUB heads: three-seed mean±SD (%). The fraction measures visible head inputs.

Budget d	Visible vocabulary (%)	Task accuracy (%)
8	7.14	76.556 ± 0.587
16	14.29	78.978 ± 0.419
24	21.43	79.502 ± 0.371
32	28.57	79.876 ± 0.422
64	57.14	80.048 ± 0.251
112	100.00	80.140 ± 0.347

B.2 Controlled design ablations

Table 5 compares MCBM variants against their own constant-rate, 100-epoch reference. The reference uses mRMR, six heads, equal head weights, and concept-loss weight one. Each alternative

changes the indicated factor. All runs use the same corrected input preprocessing and three seeds. The two-head variant retains only the 64- and 112-concept heads, with equal weights; the budget-weighted variant retains six heads and assigns weight $d/256$ to each head. The native heads are evaluated after 100 epochs.

These are design ablations under the constant-rate recipe, not single-factor comparisons with the learning-rate-decay configuration in Table 4. Natural and relevance ordering, increased concept-loss weight, and removing the smaller heads do not improve the mean native accuracy at either main budget in this setting. Budget weighting changes the $d = 64$ mean by only $+0.017$ percentage points and lowers the full-budget mean. These observations do not establish that mRMR or equal weighting is optimal across training regimes. They also do not imply the same ranking for refitted readouts.

Table 5: Native accuracy under the same constant-rate, 100-epoch recipe (three-seed mean±SD, %). Each variant changes one factor from the first-row reference.

Variant	$d = 64$	$d = 112$
mRMR, six equal-weight heads	75.135 ± 0.777	74.848 ± 0.902
Natural concept order	73.939 ± 1.730	73.875 ± 1.419
Relevance order	73.950 ± 0.682	73.818 ± 0.838
Concept-loss weight $\alpha = 2$	74.163 ± 1.924	73.944 ± 1.714
Two heads: $d = 64, 112$	74.341 ± 0.916	74.146 ± 0.967
Six heads weighted by $d/256$	75.152 ± 0.337	74.612 ± 0.402

B.3 Every intervention-aware readout condition

Table 6 reports every family–budget–weight combination, with three seeds per condition. The raw readouts and replacement protocol are fixed across weights.

Table 6: Complete raw-readout results on CUB: three-seed mean±SD (%). Each row starts from its own natural prediction.

Encoder	d	β	Natural accuracy	nAUC
MCBM	64	0	78.996 ± 0.046	47.670 ± 0.883
MCBM	64	0.1	79.053 ± 0.287	93.149 ± 0.344
MCBM	64	0.5	78.719 ± 0.329	94.764 ± 0.143
FullJoint	64	0	77.655 ± 0.341	79.957 ± 0.881
FullJoint	64	0.1	77.678 ± 0.347	93.953 ± 0.377
FullJoint	64	0.5	77.505 ± 0.455	95.566 ± 0.093
Joint64	64	0	78.041 ± 0.571	68.387 ± 1.690
Joint64	64	0.1	77.862 ± 0.574	93.563 ± 0.057
Joint64	64	0.5	77.661 ± 0.445	95.014 ± 0.058
MCBM	112	0	79.427 ± 0.175	62.724 ± 1.163
MCBM	112	0.1	79.392 ± 0.170	95.741 ± 0.107
MCBM	112	0.5	79.139 ± 0.204	96.718 ± 0.107
FullJoint	112	0	78.783 ± 0.185	76.273 ± 2.176
FullJoint	112	0.1	78.794 ± 0.115	95.893 ± 0.177
FullJoint	112	0.5	78.719 ± 0.046	96.742 ± 0.164

Table 6 contains all 15 family–budget–weight combinations for the frozen-encoder intervention experiment. Each combination averages the same three seed identities. Each encoder family is fitted with all three weights, $\beta \in \{0, 0.1, 0.5\}$. Its readout protocol, calibration rule, and query cost remain fixed across weights. The zero-weight natural head is reused after exact numerical agreement; its intervention curve is evaluated under the same replacement protocol. FullJoint at $d = 64$ is a prefix refit, and Joint64 is evaluated at its native deployment budget. All rows retain the encoder sources used in the corresponding natural-prediction study.

All curves use raw logits and calibrated fixed-prefix replacements. They are distinct from the probability+CooP curves. Increasing the intervention weight improves nAUC for all three families here. It can also reduce natural accuracy, and MCBM does not have the highest nAUC at every common weight and budget. At $d = 64$, the $\beta = 0.5$ MCBM readout has higher natural accuracy than both baselines but lower nAUC. At $d = 112$, its nAUC is close to FullJoint. Thus the repair is useful but is not exclusive to nested training. Means and sample SDs are descriptive, not significance or noninferiority tests.

C External reference results and concept diagnostics

Table 7 reports completed external routes on the same 5,794 CUB test IDs and training-derived class prototypes. Every model deploys 112 concepts. These are single-seed reference experiments with different backbones, training objectives, horizons, and intervention mechanisms. They are not a pooled ranking or a matched comparison against the three-seed prefix experiments. The natural column reports each model's separate natural evaluation; the same checkpoint is reused across its policy rows.

Table 7: Completed external reference routes. Natural accuracy and nAUC are percentages. Source training details appear in Appendix A.4. IntCEM nAUC uses actual scalar allowance; other rows use scalar queries. SCBM retains the precision of its rounded native outputs.

Model	Query policy	Natural	nAUC
IntCEM	Learned groups	72.5406	96.0533
IntCEM	Random groups	72.5406	94.2778
SCBM	Probability uncertainty	68.9	98.1112
SCBM	Random	68.9	93.5893
Minimal	Confidence	73.4553	90.7506
Minimal	Fixed mRMR	73.4553	94.5173
Concept-only	CooP ($\lambda = 100$)	73.2654	97.7910
Concept-only	Entropy	73.2654	97.3351
Concept-only	Fixed mRMR	73.2654	94.6919

IntCEM uses the completed recovery run, which restored the prescribed loader without changing its scientific parameters. Its 28-group trajectory is projected by each example's actual cumulative concept cost. Groups are neither split nor converted using average size; unused allowance remains unused. SCBM uses native conditional propagation with 100 Monte Carlo samples. Its random and uncertainty curves begin at 69.0% and 68.7%, respectively, rather than the separate 68.9% natural score. Their AUC values integrate these original rounded curves; extra displayed AUC digits do not imply unrounded sample-level precision.

Minimal retains its native latent intervention and singleton adaptation; only the frozen query order changes between rows. The concept-only CooP experiment uses a frozen Inception-v3 encoder trained for 100 epochs with seed 101, a probability head, binary replacement, and the unmodified public generator. Its $\lambda = 100$ was chosen on validation and lies at the search-grid boundary. Within this fixed source, CooP improves nAUC by 0.4560 points over entropy and 3.0991 over fixed mRMR. These gains belong to the external concept-only source, not to MCBM. The class-prototype setting also prevents interpreting high corrected endpoints as demonstrated human-intervention effectiveness.

C.1 Earlier validation concept quality

Table 8 uses an earlier seed-101, final-epoch-100 CUB cohort and the same 1,198 validation examples. It does not describe the latest approximately 80%-accuracy encoders. Concept F1 is the mean positive-class F1 over supported concepts, not task macro-F1. The 112-column measures the entire encoder vocabulary; Joint8 still predicts and receives supervision for all 112 concepts. ECE uses 15 equal-width bins. FullJoint has no native 8-concept task head, hence the dash.

Table 8: Earlier validation diagnostics, not current-model test scores. Task accuracy is a percentage; concept metrics are on $[0, 1]$. F1 columns refer to the first eight ranked concepts and all 112 concepts.

Source	Task acc. (8)	F1 (8)	F1 (112)	Brier (8)	ECE (8)
Concept-only	59.77	.9181	.8823	.0576	.0273
MCBM	70.95	.7955	.8578	.1584	.1315
Joint8	69.53	.6282	.8205	.2922	.2656
FullJoint	—	.8724	.8064	.0857	.0330

In this cohort, MCBM improves task accuracy over concept-only readouts while reducing agreement and calibration against concept prototypes. Relative to Joint8, both task accuracy and prefix concept quality are better. FullJoint also has higher prefix concept F1 than MCBM despite lacking a native eight-concept task interface. These observations reveal a task–concept tradeoff; they neither prove leakage nor establish semantic purity. Scores were independently checked from saved final predictions, without new image inference or test access.

D Representation control values

The following values accompany Figure 3a. The historical frozen encoders and representation-matched readouts are distinct from the optimized native models.

Table 9: Historical single-seed encoders at $d = 64$: accuracy (%) after 5,000 steps of representation-matched linear readout fitting.

Frozen encoder	Raw logits	Probabilities	Binary concepts
Concept-only	72.82	73.30	71.28
FullJoint	75.27	71.75	65.77
MCBM	75.39	73.51	70.40

E Nested observations and replacement measurements

E.1 Nested observations and trained predictors

Let R^*_d be the Bayes classification risk when the full observation is the first d true concepts. If $d_1 < d_2$, any classifier using $c_{\pi,1:d_1}$ can be implemented on $c_{\pi,1:d_2}$ by ignoring the added coordinates. Taking the infimum over all measurable classifiers gives $R^*_{d_2} \leq R^*_{d_1}$. This is a statement about nested observations and optimal classifiers. Separately trained neural heads need not attain those risks. Their observed errors need not be ordered by budget.

Replacement has a different structure. The vectors $T(u, c, A_b)$ and $T(u, c, A_{b+1})$ need not contain nested information: the second vector replaces a previous predicted value rather than retaining it and adding an annotation. That predicted value may carry information beyond its binary label. Hence nesting of query sets does not establish an increase in mutual information with the class. It also does not establish a decrease in the risk of a fixed task head. This is why the experiments measure the complete curve and harm instead of imposing monotonicity.

E.2 Scope of the intervention measurements

Accuracy at a fixed query count measures the trained head on that altered input distribution. Repair conditions on examples that were initially incorrect. Harm conditions on examples that were initially correct. Their denominators differ, so they should not be added or directly compared as if they described the same population. A zero harm value at the final point also says nothing about harm at intermediate budgets. The normalized area summarizes a complete curve over one deployed prefix; areas for $d = 64$ and $d = 112$ cover different absolute query ranges.