CABM: EFFICIENT CONCEPT AUGMENTATION FOR INTERPRETABLE DEEP MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept Bottleneck Models (CBMs) offer interpretability by aligning latent representations with semantic concepts. However, standard CBMs suffer from a static concept space, limiting their adaptability to evolving knowledge. Expanding this space typically necessitates full retraining, which is computationally prohibitive and risks semantic drift in pre-existing concepts. In this work, we propose **Concept Augmentation in Bottleneck Models (CABM)**, a novel framework for post-hoc concept addition that avoids iterative end-to-end retraining. CABM formulates concept addition as a parameter perturbation problem, addressed by influence-inspired local updates. To overcome the scalability bottleneck of dense curvature inversion in high-dimensional spaces, we leverage Eigenvalue-corrected Kronecker-Factored Approximate Curvature (EK-FAC) to perform structured approximate second-order updates. We further introduce a trust-region constraint and concept-conditional initialization to improve numerical stability. We bound its local error relative to a re-optimized appended concept block, including the effect of trust-region projection. Extensive experiments show that our framework achieves predictive performance on par with full retraining while reducing computational costs by 10.8–$61.5\times$ relative to full retraining and 1.21–$1.53\times$ relative to a matched converged frozen-feature probe, offering a robust solution for dynamic interpretable modeling.

1 INTRODUCTION

Concept Bottleneck Models (CBMs) (Koh et al., 2020) have recently become an attractive framework for interpretable learning in medical AI. By introducing a human-understandable concept layer between the inputs and the predictions, CBMs allow model decisions to be expressed through clinically meaningful features rather than opaque latent representations (Gong et al., 2025; Hu et al., 2024). This aligns well with medical practice, where diagnostic reasoning depends on observable attributes and requires transparent justification (Wu et al., 2025; Kopf & Claassen, 2021).

In real clinical workflows, medical knowledge evolves continuously as new diagnostic cues, risk factors, and biomarkers are uncovered through ongoing research and advances in imaging and testing technologies. As a result, some clinically relevant attributes may emerge after a CBM has already been deployed, meaning that its original concept set no longer fully captures the factors driving medical decision-making. For example, during the early stages of the COVID-19 pandemic, obesity was only later recognized as a key predictor of severe outcomes (Sattar et al., 2020), and demographic characteristics such as age, gender, and ethnicity were subsequently identified as important determinants of infection risk (Rozenfeld et al., 2020). A straightforward solution is to retrain the expanded CBM, but this is computationally costly and may change previously learned concept predictions, motivating more targeted updates for clinical workflows (Wiens et al., 2019; Finlayson et al., 2021).

We study the problem of **post-hoc concept addition** (see details in Fig. 1) for CBMs in medical applications. Once a model has been trained on an initial set of k clinical concepts, an important question is how to efficiently add newly discovered concepts while preserving interpretability and predictive performance. We consider a fixed downstream label space, with training inputs, task labels, and new concept annotations available. This setting complements label-efficient learning (Oikarinen et al., 2023; Yuksekgonul et al., 2022; Lai et al., 2024) and utility gap reduction (Sheth & Ebrahimi Kahou,

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

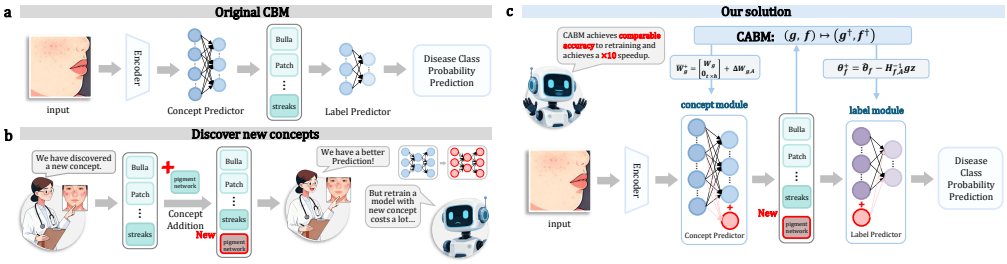


Figure 1: **Post-hoc concept addition in CBMs.** (a) Standard CBMs use a concept set specified at training time. (b) Full retraining can incorporate new concepts, but is costly and may change existing concept predictions. (c) CABM adds new concepts through localized updates, preserving existing concept scores with accuracy comparable to full retraining at lower update cost.

2023; Espinosa Zarlenga et al., 2022) by focusing on structural concept expansion after training. Addressing this goal raises several key challenges: *(i)* The concept predictor needs to extend its output space, and the label predictor must adapt to higher-dimensional concept representations; *(ii)* New concepts need to be integrated while preserving existing concept predictions and limiting changes in their downstream effects; *(iii)* Finally, the update should reduce computation without sacrificing new-concept learning.

When the new concepts can be predicted from the pretrained features, concept addition can focus on the added predictors and their downstream connections rather than relearning the representation. We therefore investigate whether concept-conditional initialization allows a single local curvature correction to approach a fully optimized appended concept predictor on the same frozen features.

To address these challenges, we propose **Concept Augmentation in Bottleneck Models (CABM)**, a framework that integrates new medical concepts *without iterative end-to-end retraining*. CABM expands both the concept and label modules, freezes the encoder and old concept pathway, and initializes the added concept predictors with concept-conditional feature statistics. It then applies a single influence-inspired approximate second-order update using EK-FAC (George et al., 2018), with a trust-region constraint to control the update magnitude. The downstream label predictor uses a one-step approximate update for the default softmax head while retaining its old coefficients and bias, and an exact fixed-design solution for the ridge variant. The frozen concept pathway preserves existing scores exactly; downstream intervention stability is evaluated separately. Experiments across five benchmarks, including two medical datasets, show performance comparable to full retraining, with lower update time than both full retraining and a matched converged frozen-feature probe. Our contributions are summarized as follows:

- We introduce *post-hoc concept addition* for CBMs with a fixed label space as a setting for adaptive and interpretable modeling in healthcare.
- We propose **Concept Augmentation in Bottleneck Models (CABM)**, an efficient framework that combines concept-conditional initialization and localized second-order updates to expand both modules while preserving existing concept scores.
- We provide local approximation guarantees for the appended concept block and empirical evidence for efficient expansion, including matched-probe comparisons and a separate evaluation of old-concept intervention stability.

2 Related Work

2.1 Dynamic and Continual Concept Bottleneck Models

Concept Bottleneck Models (CBMs) (Koh et al., 2020) provide interpretable decision pathways by predicting human-defined concepts before the final label, making them attractive for high-stakes and medical decision support (Ahmad et al., 2018; Yu et al., 2018; Rudin, 2019; Hu et al., 2025). However, standard CBMs assume a fixed concept vocabulary, whereas clinical knowledge evolves as new diagnostic findings, imaging cues, and risk factors are identified (Sattar et al., 2020; Wiens et al., 2019; Finlayson et al., 2021). Recent work relaxes this assumption through post-hoc or open-

vocabulary CBMs (Yuksekgonul et al., 2022; Oikarinen et al., 2023; Yang et al., 2023; Tan et al., 2024), editable or controllable CBMs (Hu et al., 2025; Lin et al., 2026), and continual CBMs such as CONCIL, CLG-CBM, and CI-CBM (Lai et al., 2025; Yu et al., 2025; Javadi et al., 2026). CABM differs from these continual-learning approaches: they typically receive a stream of tasks or classes and mitigate forgetting through recursive analytic updates, replay, distillation, prototype generation, or regularization, whereas CABM addresses *post-hoc structural concept addition* in a deployed CBM with a fixed old label space and fixed old concept semantics. CABM directly appends new concept outputs and updates only the corresponding parameter blocks through a localized second-order edit.

2.2 Model Editing and Influence-Based Updates

Model editing methods update model behavior through lightweight parameter modifications (Meng et al., 2022a; Mitchell et al., 2022; Meng et al., 2022b; Mazzia et al., 2024), but most target factual association editing in language models rather than architectural expansion of a structured concept layer. Influence functions (Koh & Liang, 2017; Robins et al., 2017; Guo et al., 2021) estimate parameter changes induced by small objective perturbations, and scalable curvature approximations such as K-FAC and EK-FAC (Martens & Grosse, 2015; George et al., 2018) make second-order updates tractable. CABM adapts this perspective to concept-space expansion: it appends new concept nodes and performs structured, blockwise second-order edits on the new concept branch and its corresponding label-head parameters, rather than editing arbitrary internal facts or globally fine-tuning the network. Alternative solvers include dense curvature, Hessian-vector products, and L-BFGS; MesaNet (von Oswald et al., 2026) and Gated KalmaNet (Peng et al., 2026) illustrate iterative local solves in sequence models. EK-FAC is a practical choice, not a universally superior solver.

2.3 Medical AI and Semantic Preservation

Clinical AI systems require not only high average accuracy, but also stable and auditable reasoning after deployment. Guidelines and deployment studies emphasize that healthcare models must be monitored under dataset shift, integrated into clinical workflows, and updated without undermining trust or safety (Wiens et al., 2019; Finlayson et al., 2021; Lekadir et al., 2025). For interpretable medical models, this requirement is stronger: a concept such as *ulcer*, *pigment network*, or *vascular structures* should retain its clinical meaning after a model update unless clinicians explicitly redefine it. CABM is designed around this semantic-preservation principle by freezing the old concept pathway and applying a localized, trust-region-controlled update to newly added concept blocks.

3 Preliminaries

3.1 Concept Bottleneck Models and Concept Addition

A Concept Bottleneck Model (CBM) (Koh et al., 2020) decomposes prediction into two stages: a concept predictor $g : \mathbb{R}^{d_i} \to \mathbb{R}^k$ maps an input x to k human-understandable concepts, and a label predictor $f : \mathbb{R}^k \to \mathbb{R}^{d_o}$ maps these concepts to the target label. Given a trained CBM (g, f) with concept vector $c \in \mathbb{R}^k$, post-hoc concept addition aims to insert t newly discovered concepts after deployment:

$$g^+ : \mathbb{R}^{d_i} \to \mathbb{R}^{k+t}, \qquad f^+ : \mathbb{R}^{k+t} \to \mathbb{R}^{d_o}. \tag{1}$$

A naive solution is to retrain the expanded model (g^+, f^+) from scratch. This is computationally expensive and may change the behavior of old concepts and old concept–label relations.

We distinguish this setting from continual learning. Continual learning usually receives a sequence of tasks or classes and optimizes the model repeatedly to balance plasticity and forgetting. In our problem, the old task and label space are fixed; only the concept space is structurally expanded. A desirable update preserves old concept predictions, $g^+(x)_{1:k} \approx g(x)_{1:k}$, while maintaining stable downstream effects of old concepts. CABM preserves the first property exactly by construction and evaluates the second through intervention diagnostics.

3.2 Influence Functions for Local Parameter Updates

Influence functions estimate how the optimum of a model changes under a small perturbation to the training objective (Koh & Liang, 2017). Let $\hat{\theta}$ minimize $L(\theta) = n^{-1}\sum_{i=1}^{n}\ell(x_i,\theta)$ and consider the perturbed objective $L_\epsilon(\theta) = L(\theta) + \epsilon\Delta L(\theta)$. If $\hat{\theta}_\epsilon$ minimizes L_ϵ, differentiating the stationarity condition around $\epsilon = 0$ gives

$$\frac{d\hat{\theta}_\epsilon}{d\epsilon}\Big|_{\epsilon=0} = -H_{\hat{\theta}}^{-1}\nabla_\theta\Delta L(\hat{\theta}), \tag{2}$$

where the Hessian $H_{\hat{\theta}} = \nabla_\theta^2 L(\hat{\theta})$ is assumed invertible at the original optimum. This expression gives a curvature-corrected estimate of how parameters should move without full retraining.

In CABM, this influence perspective motivates curvature-corrected updates to the appended concept block and the corresponding label-head block. We formulate the finite update directly on the appended-block objective, with the encoder and old concept pathway fixed. This block-localized formulation is the basis for efficient concept expansion and exact preservation of old concept outputs; its local approximation guarantee is stated in Section 5.

4 Concept Augmentation in Bottleneck Models (CABM)

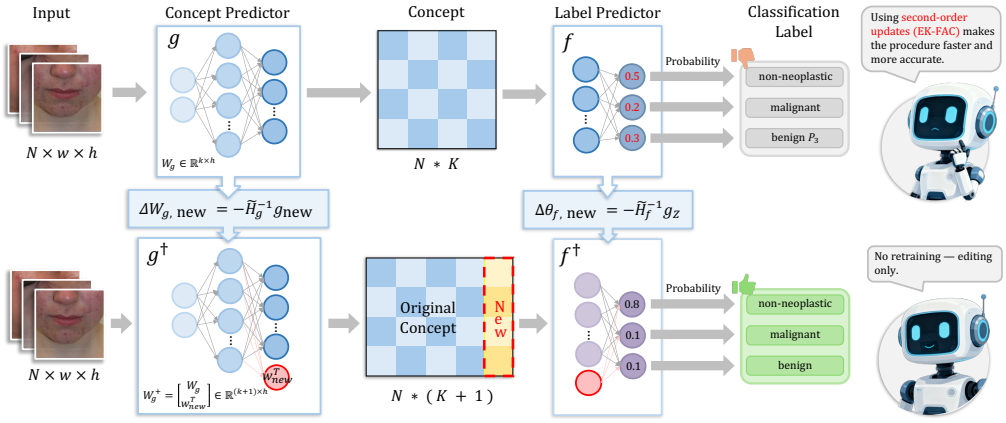


Figure 2: **CABM workflow:** new concepts are added via curvature-aware edits, enabling rapid model update without iterative end-to-end retraining.

Fig. 2 provides an overview of the CABM workflow. Consider a trained Concept Bottleneck Model (CBM) trained on a dataset $\mathcal{D} = \{(x_i, c_i, y_i)\}_{i=1}^n$, with concept labels $c_i \in \{0,1\}^k$ and target y_i. The training objective is given by:

$$\min_{g,f}\mathcal{L}(\theta) = \sum_{i=1}^{n}\Big(L_C(g(x_i), c_i) + L_Y(f(g(x_i)), y_i)\Big), \tag{3}$$

where L_C and L_Y denote the concept and label losses, respectively.

4.1 Concept Addition Formulation

We address the problem of extending a trained CBM from k to $k{+}t$ concepts. Instead of retraining the expanded model (g^+, f^+) from scratch, we derive a lightweight update rule $(g, f) \mapsto (g^\dagger, f^\dagger)$. Our goal is to incorporate new concepts via a single curvature-corrected step while retaining the encoder and old concept pathway.

4.2 Concept Predictor Update via Influence Functions

To accommodate new concepts, we append a block $u_{\mathcal{A}}(x) = W_{\mathcal{A}}\bar{\phi}(x)$ to g, where $\bar{\phi}(x) = [\phi(x)^\top, 1]^\top$ includes the bias. For binary concepts, $W_{\mathcal{A}} \in \mathbb{R}^{t\times(h+1)}$ with a sigmoid readout;

embedding variants can have greater editable width (Appendix A.2). On fixed features, the objective is

$$J_{\mathcal{A}}(W) = \frac{1}{n}\sum_{i=1}^{n} \ell_{\mathcal{A}}(W\bar{\phi}_i, \tilde{c}_{i,\mathcal{A}}) + R_{\mathcal{A}}(W), \tag{4}$$

where $R_{\mathcal{A}}$ is explicit regularization. Let $G_{\mathcal{A}} = \nabla_W J_{\mathcal{A}}(W_{\mathcal{A}}^{(0)})$. Motivated by influence functions (Koh & Liang, 2017), CABM approximates the quadratic-model step $-H_0^{-1}\,\mathrm{vec}(G_{\mathcal{A}})$, where H_0 is the appended-objective Hessian at initialization.

Tractable Approximation via EK-FAC. We use Eigenvalue-corrected Kronecker-Factored Approximate Curvature (EK-FAC) (George et al., 2018). With $s_i = \nabla_{u_{\mathcal{A}}} \ell_{\mathcal{A},i}$, define $D = n^{-1}\sum_i \bar{\phi}_i\bar{\phi}_i^\top$ and $B = n^{-1}\sum_i s_i s_i^\top$. K-FAC (Martens & Grosse, 2015) uses $D \otimes B$; EK-FAC corrects its spectrum in the factor eigenbases Q_D, Q_B:

$$M_i = Q_B^\top s_i \bar{\phi}_i^\top Q_D, \qquad \Lambda^* = \frac{1}{n}\sum_i M_i \odot M_i. \tag{5}$$

This approximates the empirical Fisher, which generally differs from the loss Hessian. For $R_{\mathcal{A}}(W) = \rho\|W\|_F^2/2$, including $\rho = 0$, the update is

$$\Delta W_{\mathcal{A}}^{\mathrm{raw}} = -Q_B\left[(Q_B^\top G_{\mathcal{A}} Q_D) \oslash (\Lambda^* + \rho + \lambda)\right] Q_D^\top. \tag{6}$$

Numerical damping λ is distinct from objective regularization ρ; other regularizers enter the solve separately. We enforce the update radius explicitly:

$$\Delta W_{\mathcal{A}} = \beta_g \Delta W_{\mathcal{A}}^{\mathrm{raw}}, \quad \beta_g = \min\left\{1, \frac{r_g}{\|\Delta W_{\mathcal{A}}^{\mathrm{raw}}\|_F}\right\}, \quad W_{\mathcal{A}}^{\dagger} = W_{\mathcal{A}}^{(0)} + \Delta W_{\mathcal{A}}, \tag{7}$$

with $\beta_g = 1$ for a zero step. Factorization and implementation details appear in Appendix A.2.

4.3 Label Predictor Adaptation

Following the concept update, the label predictor f must adapt to the expanded input dimension. Let $C \in \mathbb{R}^{n\times d_C}$ and $Z \in \mathbb{R}^{n\times d_A}$ denote the original and new head inputs; $d_C = k, d_A = t$ for scalar concepts.

Linear/Ridge Head. For a ridge regression head, the optimization problem $\min_V \|[C\ Z]V - Y\|_F^2/(2n) + \rho\|V\|_F^2/2$, with $\rho > 0$, admits a closed-form solution. We reuse the original factorization through the Schur complement.

Proposition 1 (Efficient Ridge Update). *Define $A = C^\top C/n + \rho I$, $U = C^\top Z/n$, $T = Z^\top Z/n + \rho I$, and $S = T - U^\top A^{-1}U$. The augmented solution is*

$$\begin{aligned} V_A &= S^{-1}(Z^\top Y/n - U^\top A^{-1} C^\top Y/n), \\ V_C &= A^{-1}(C^\top Y/n - U V_A). \end{aligned} \tag{8}$$

Remark 1. After forming the new Gram blocks, the additional factorization cost is $O(d_C^2 d_A + d_C d_A^2 + d_A^3)$. This exact fixed-design solution reoptimizes both old and new head coefficients; the derivation and data-dependent costs are in Appendix A.3.

Softmax Head. Our default head predicts $\mathrm{softmax}(V_C^\top c + V_A^\top z + b)$. We retain V_C, b, initialize $V_A = 0$, and use the label block's own EK-FAC factors for one damped curvature-corrected step, radially clipped to radius r_f.

4.4 Initialization with Concept-Conditional Priors

Standard initialization (e.g., random or zero) can require a larger correction. For each binary concept a, our prior-informed initialization is

$$w_a^{(0)} = \alpha(\mu_{a,1} - \mu_{a,0}), \qquad b_a^{(0)} = \log\frac{\pi_a}{1-\pi_a}, \tag{9}$$

where $\mu_{a,b} = \mathbb{E}[\phi(x) \mid \tilde{c}_a = b]$ and $\pi_a = \Pr(\tilde{c}_a = 1)$ are estimated on training data, with both groups present. This centroid-difference warm start uses concept presence and absence, with a prevalence bias, and is evaluated in the initialization ablation.

5 Theoretical Analysis of Concept Addition

We analyze approximation accuracy against a matched optimum of $J_{\mathcal{A}}$ on the same fixed features and with the same regularization, followed by semantic stability.

5.1 Approximation Guarantee

Theorem 1 (Local Approximation Error Bound). *Let $\vartheta = \operatorname{vec} W_{\mathcal{A}}$. Suppose $J_{\mathcal{A}}$ is μ-strongly convex with an M-Lipschitz Hessian on a convex neighborhood containing initialization ϑ_0 and stationary point ϑ^*. Write $g = \nabla J_{\mathcal{A}}(\vartheta_0)$, $H_0 = \nabla^2 J_{\mathcal{A}}(\vartheta_0)$, and $\Delta^{\mathrm{raw}} = -Pg$, where P is CABM's damped inverse-curvature operator. If $\|P - H_0^{-1}\|_2 \le \delta$, the clipped step Δ satisfies*

$$\|\vartheta_0 + \Delta - \vartheta^*\|_2 \le \delta\|g\|_2 + \frac{M}{2\mu^3}\|g\|_2^2 + \|\Delta - \Delta^{\mathrm{raw}}\|_2. \tag{10}$$

The bound separates curvature approximation, local linearization, and clipping error. This guarantee concerns the appended concept block, not unrestricted retraining or the label head. Here δ includes empirical-Fisher/Hessian mismatch, EK-FAC approximation, and damping. Appendix B.1 gives the proof.

5.2 Stability and Semantic Fidelity

Fixing the encoder and old concept pathway preserves $g^\dagger(x)_{1:k} = g(x)_{1:k}$ exactly. We further characterize local model error and downstream intervention stability.

Proposition 2 (Decoupling and Stability). *For an M-Lipschitz Hessian on the radius-r_g ball and $\|\widetilde{H} - H_0\|_2 \le \epsilon_H$, define $q(\Delta) = J_{\mathcal{A}}(\vartheta_0) + g^\top\Delta + \frac{1}{2}\Delta^\top\widetilde{H}\Delta$. Every $\|\Delta\|_2 \le r_g$ satisfies*

$$|J_{\mathcal{A}}(\vartheta_0 + \Delta) - q(\Delta)| \le \tfrac{1}{2}\epsilon_H r_g^2 + \tfrac{M}{6}r_g^3. \tag{11}$$

For the default softmax head with fixed V_C, b and $\|V_A\|_F \le r_f$, let $I_j(V_A)$ be the label-probability difference when old concept j is set to one versus zero, holding other inputs, including z, fixed. Then $\|I_j(V_A) - I_j(0)\|_2 \le \|z\|_2 r_f$.

The intervention bound concerns the frozen-old-coefficient softmax update. Proofs and the distinction from ridge are in Appendix B.2.

6 Experiments

We evaluate CABM across five concept bottleneck benchmarks to address four core research questions:

RQ1: *Effectiveness.* Does CABM achieve predictive performance comparable to full retraining and modern continual-learning baselines under both single-concept and multi-concept addition?

RQ2: *Efficiency & Scalability.* Can CABM substantially reduce update time compared with full retraining and continual-learning methods as the number of added concepts increases?

RQ3: *Semantic Preservation.* Does CABM preserve existing concept predictions and maintain stable downstream intervention behavior after expanding the concept space?

RQ4: *Design Rationality.* How do concept-conditional priors, joint updates to g and f, and EK-FAC–based second-order curvature contribute to CABM?

Our central empirical claim is not that CABM always maximizes accuracy by a large margin, but that it achieves *accuracy comparable to retraining and modern continual-learning baselines* while providing two practical advantages: **lower update time** and **exact preservation of old concept scores**. We report averaged performance, stability analyses, component ablations, and qualitative case studies, with per-setting results in Appendix E.6.

Table 1: Overall concept addition performance averaged over all addition settings within each dataset. L denotes label accuracy (%), C denotes concept accuracy (%), and T denotes runtime in seconds. CABM achieves comparable predictive accuracy with lower update time; the matched Probe-LR comparison isolates the gain under frozen features. Detailed per-concept and per-group results are reported in Appendix E.6 and Appendix E.8.

Method	SkinCon			Derm7pt			CUB			CelebA			AwA2		
	L	C	T	L	C	T	L	C	T	L	C	T	L	C	T
Original Base	72.97	88.77	0.0	80.24	74.45	0.0	79.17	96.55	0.0	41.91	51.08	0.0	89.78	97.50	0.0
Retraining	72.99	87.79	346.1	80.23	74.70	343.0	**81.35**	**96.72**	11931	42.78	53.54	1452	**90.20**	97.61	7196
FastIF	70.74	86.59	65.4	78.04	72.41	59.2	79.77	96.05	377	42.44	52.75	69.4	89.89	97.21	806.2
Partial-FT	71.79	86.97	124.7	78.49	72.95	114.0	80.28	96.32	717	42.66	53.29	892.3	89.96	94.92	4183
DER++	72.45	87.50	315.2	79.31	73.58	289.4	80.86	96.44	2210	42.69	53.32	1214	89.97	97.28	5773
CONCIL	72.91	88.10	242.8	80.08	74.16	226.7	81.12	96.58	1760	42.82	53.92	1108	90.13	97.50	5559
CLG-CBM	72.87	88.94	211.6	80.50	74.78	253.0	81.16	96.63	2395	43.03	53.56	890.6	90.11	97.47	3815
CI-CBM	72.93	88.20	236.4	80.22	74.38	198.5	81.22	96.66	2653	43.28	55.10	965.9	90.14	**97.62**	3221
Probe-LR	72.95	89.31	49.20	**80.81**	**75.25**	39.08	81.09	96.28	289	43.40	56.17	70.11	90.05	97.32	476.5
CABM	**73.01**	**89.33**	**32.15**	80.55	74.96	**27.7**	81.34	96.70	**194**	**43.66**	**56.77**	**46.8**	90.15	97.53	**395.3**

6.1 Experimental Setup

We evaluate CABM on five concept bottleneck benchmarks spanning medical, fine-grained, and large-scale settings: SkinCon (Daneshjou et al., 2022), Derm7pt (Kawahara et al., 2018), CUB-200-2011 (Wah et al., 2011), CelebA (Liu et al., 2015), and AwA2 (Xian et al., 2018). The evaluation covers leave-one-out single-concept addition on SkinCon and Derm7pt, semantic group addition on CUB, simultaneous two-concept addition on CelebA, and larger-scale 5/10/15-concept addition on AwA2. We compare CABM with the original base model, full retraining, FastIF (Guo et al., 2021), Partial-FT, DER++ (Buzzega et al., 2020), CONCIL (Lai et al., 2025), CLG-CBM (Yu et al., 2025), and CI-CBM (Javadi et al., 2026). Unless otherwise specified, concept accuracy averages the available outputs (k before and $k + t$ after expansion); on SkinCon, Derm7pt, CUB, and AwA2, thresholds are recalibrated on validation data after editing. We report label accuracy, concept accuracy, and wall-clock update time, and further evaluate semantic preservation through noise robustness, intervention, sequential addition, and downstream-drift analyses. All methods start from the same trained base CBM and are evaluated on the same test split. Appendix E gives dataset, protocol, baseline, and training details.

We add *Converged Probe-LR*: the same frozen features, annotations, appended objective, regularization, and label-head update, with L-BFGS optimization to convergence. CelebA has 256 classes defined by eight binary attributes: four are initially explicit, two are added, and two remain unannotated (Appendix D.2).

6.2 Main Results: Averaged Concept Addition Performance

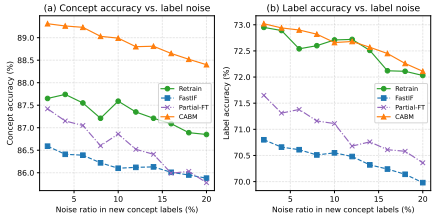
Overall takeaway. Table 1 shows that CABM provides a favorable trade-off between predictive performance and update time. Full retraining remains a strong reference because it globally re-optimizes the expanded model, and it obtains the highest label accuracy on CUB and AwA2. However, the accuracy gaps are very small: CABM reaches 81.34% on CUB compared with 81.35% for retraining, and 90.15% on AwA2 compared with 90.20%. In return, CABM reduces update time from hours to seconds or minutes. Compared with retraining, CABM is approximately $10.8\times$ faster on SkinCon, $12.4\times$ faster on Derm7pt, $61.5\times$ faster on CUB, $31.0\times$ faster on CelebA, and $18.2\times$ faster on AwA2. This is the key practical advantage of CABM: it enables deployed CBMs to absorb new concepts without repeatedly taking the model offline for full retraining.

Matched update cost. The speedup is 1.21–$1.53\times$ over Converged Probe-LR and 3.7–$19.1\times$ over Partial-FT. The larger 10.8–$61.5\times$ gain over full retraining also includes the cost of re-optimizing the encoder. These comparisons distinguish the benefit of a one-step correction from the benefit of freezing the feature extractor (Appendix D.7).

Effectiveness across datasets. On the two medical datasets, CABM achieves competitive label and concept accuracy. On SkinCon, CABM improves concept accuracy from 87.79% under retraining to 89.33%, while also slightly improving label accuracy. On Derm7pt, CABM similarly improves both

Method	Concept Acc (↑,%)	Label Acc (↑,%)	Runtime (↓,s)
Zero-init	52.31	58.72	32.74
Global prior	70.19	65.24	32.11
Concept-cond. prior only	80.49	68.28	27.40
Concept-cond. priors (CABM)	**89.54**	**73.01**	32.15
Only edit g	89.33	70.12	—
Only edit f	88.77	73.04	—
CABM-EKFAC	**89.33**	**73.01**	32.15
CABM-Diag	89.18	72.56	29.47
CABM-FirstOrder	88.95	70.78	**26.32**

(a) **Design component ablation.**



(b) **Label noise robustness.**

Figure 3: **Ablation and robustness analysis.** (a): the first four rows use new-concept accuracy; the remaining rows use all-concept accuracy. (b): CABM remains stable when synthetic noise is injected into the labels of the newly added concept.

label and concept accuracy, suggesting that localized updates are especially useful when concept annotations are scarce and noisy. On CelebA, CABM obtains 43.66% label accuracy and 56.77% concept accuracy, with a slightly higher reported mean than retraining; the setting-wise variation is reported in Appendix D.2. On CUB and AwA2, CABM remains nearly tied with retraining while requiring substantially less computation. These results support our main claim: CABM achieves predictive accuracy comparable to retraining at lower update cost. Its small accuracy advantages do not imply that one-step editing systematically outperforms full optimization.

Comparison with modern continual-learning baselines. DER++, CONCIL, CLG-CBM, and CI-CBM provide additional continual-learning comparisons alongside FastIF and Partial-FT. CONCIL uses recursive analytic updates; DER++, CLG-CBM, and CI-CBM use iterative training with replay, language guidance, or pseudo-concept mechanisms. By contrast, CABM performs a one-shot structural edit: it appends the new concept block, keeps the old concept pathway fixed, and updates only the corresponding concept and label-head parameters. Empirically, CLG-CBM and CI-CBM obtain slightly lower accuracy than CABM on most datasets and require substantially longer update time. This difference highlights the distinction between our setting and standard continual learning. Continual learning mitigates forgetting during sequential updates, whereas CABM directly performs post-hoc structural concept addition in an already deployed CBM.

Semantic preservation under concept expansion. Accuracy alone is insufficient for evaluating post-hoc CBM editing. A useful edit should also preserve existing concept predictions and maintain stable downstream concept–label reasoning. CABM is designed around this requirement: old concept parameters are frozen, the newly introduced concept block is isolated, and the label-head update is controlled by a trust-region second-order correction. In contrast, retraining and iterative continual-learning baselines perform global or semi-global optimization, which can change the learned dependencies between old concepts and labels. Frozen old concept scores are exactly preserved, but the expanded label head can change their downstream effects. CABM achieves new-concept accuracies of 89.54% and 75.37% on SkinCon and Derm7pt, with old-concept intervention drift of 0.006 and 0.009, versus 0.068 and 0.091 for retraining. Thus exact score preservation and stable downstream intervention effects are evaluated separately. We therefore evaluate label-noise robustness, intervention sensitivity, sequential concept addition, runtime scaling, and KL-based downstream prediction drift. Appendix D.1 separately measures threshold recalibration and changes in old-concept intervention effects.

6.3 Stability, Ablations, and Case Studies

Robustness to noisy new concepts. Fig. 3(b) evaluates whether CABM remains reliable when the newly added concept labels are corrupted. Retraining is sensitive to noisy new-concept labels because corrupted supervision can influence the optimization trajectory of the expanded model. Partial-FT also exhibits unstable behavior because it still relies on iterative gradient updates. CABM is more stable because the update is localized and its magnitude is controlled by a curvature-aware trust region. As a result, CABM maintains higher concept and label accuracy across increasing noise ratios.

Intervention, sequential addition, and downstream drift. Fig. 4(a) shows that CABM remains compatible with standard concept intervention: replacing predicted concepts with ground-truth values

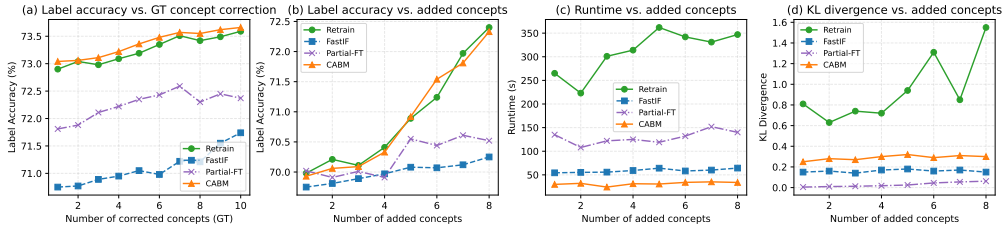


Figure 4: **Stability under concept-space expansion and expert intervention (SkinCon).** (a) Test-time intervention. (b) Sequential concept addition. (c) Runtime scaling. (d) Downstream prediction drift.

improves downstream label accuracy. This indicates that the edited bottleneck remains actionable after concept expansion. Fig. 4(b)–(c) further shows that CABM supports sequential concept addition with stable accuracy growth and low runtime. Unlike gradient-based methods, CABM avoids repeated multi-epoch optimization at each edit. Finally, Fig. 4(d) measures downstream prediction drift using KL divergence, rather than a change in the frozen old concept scores. Partial-FT has low drift but limited plasticity, while retraining can introduce larger shifts through global re-optimization. CABM maintains a moderate and stable divergence, suggesting a balance between learning the new concept and preserving downstream behavior. The separate intervention-effect diagnostic in Appendix D.1 tests the behavior of old concepts after label-head expansion.

Design validation. Fig. 3(a) validates the main components of CABM. Zero initialization severely underfits the new concept, showing that an uninformed starting point is insufficient. Global priors improve performance, while concept-conditional priors use feature statistics conditioned on concept presence or absence to provide a better warm start. The prior-only ablation reaches 68.28% label accuracy, compared with 73.01% for the full correction. The "Only edit g" control updates the appended concept predictor but freezes the new label-head connections at their fan-in uniform initialization. The new concept-to-label connections therefore remain unadapted (Appendix E.7). Editing only f improves the classifier but cannot learn the new concept predictor. The full CABM update jointly edits both modules. Finally, diagonal and first-order approximations are slightly faster, while EK-FAC gives the strongest label accuracy in this ablation. The advantage over diagonal curvature is modest; broader curvature and local-solution diagnostics appear in Appendices D.4 and D.5.

6.4 Qualitative Case Studies

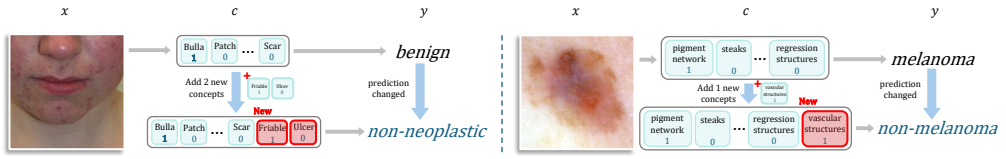


Figure 5: **Case studies on SkinCon and Derm7pt.**

Fig. 5 shows two examples where newly added concepts introduce clinically meaningful decision changes. After incorporating the new risk factor, CABM correctly separates high-risk from low-risk cases without full retraining. This behavior matches the intended clinical workflow: when physicians identify a new imaging sign, such as vascular structures or emergent morphological cues, CABM can incorporate the new concept while preserving the original concept scores.

7 Conclusion

We presented CABM, an efficient framework for post-hoc concept addition in Concept Bottleneck Models. CABM introduces new concepts through a single curvature-corrected update, avoiding iterative end-to-end retraining while exactly preserving existing concept scores. Experiments on dermatology benchmarks show that CABM achieves accuracy comparable to retraining, remains stable under the evaluated noise levels, and scales to batch and sequential concept additions.

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

This work evaluates updates to concept-based models on existing benchmark datasets. Medical-image and facial-attribute results are research evaluations and do not establish clinical efficacy, fairness across populations, or suitability for deployment. Preserving a concept predictor can also preserve its existing errors and biases. New concepts and model updates require appropriate data permissions, expert assessment, and evaluation under the intended conditions of use.

REFERENCES

Muhammad Aurangzeb Ahmad, Carly Eckert, Ankur Teredesai, and Greg McKelvey. Interpretable machine learning in healthcare. *BCB*, 18:559–560, 2018.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Roxana Daneshjou, Mert Yuksekgonul, Zhuo Ran Cai, Roberto Novoa, and James Y Zou. Skincon: A skin disease dataset densely annotated by domain experts for fine-grained debugging and analysis. *Advances in Neural Information Processing Systems*, 35:18157–18167, 2022.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in neural information processing systems*, 35:21400–21413, 2022.

Samuel G Finlayson, Adarsh Subbaswamy, Karandeep Singh, John Bowers, Annabel Kupke, Jonathan Zittrain, Isaac S Kohane, and Suchi Saria. The clinician and dataset shift in artificial intelligence. *New England Journal of Medicine*, 385(3):283–286, 2021.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in neural information processing systems*, 31, 2018.

Shizhan Gong, Huayu Wang, Xiaofan Zhang, and Qi Dou. Concepts from neurons: Building interpretable medical image diagnostic models by dissecting opaque neural networks. In *International Conference on Information Processing in Medical Imaging*, pp. 3–18. Springer, 2025.

Han Guo, Nazneen Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. Fastif: Scalable influence functions for efficient model interpretation and debugging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10333–10350, 2021.

Lijie Hu, Songning Lai, Wenshuo Chen, Hongru Xiao, Hongbin Lin, Lu Yu, Jingfeng Zhang, and Di Wang. Towards multi-dimensional explanation alignment for medical classification. *Advances in Neural Information Processing Systems*, 37:129640–129671, 2024.

Lijie Hu, Chenyang Ren, Zhengyu Hu, Hongbin Lin, Cheng-Long Wang, Zhen Tan, Weimin Lyu, Jingfeng Zhang, Hui Xiong, and Di Wang. Editable concept bottleneck models. In *Forty-second International Conference on Machine Learning*, 2025.

Amirhosein Javadi, Tuomas Oikarinen, Tara Javidi, and Tsui-Wei Weng. Ci-cbm: Class-incremental concept bottleneck model for interpretable continual learning. *arXiv preprint arXiv:2604.14519*, 2026.

Jeremy Kawahara, Sara Daneshvar, Giuseppe Argenziano, and Ghassan Hamarneh. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE journal of biomedical and health informatics*, 23(2):538–546, 2018.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

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.

Andreas Kopf and Manfred Claassen. Latent representation learning in biology and translational medicine. *Patterns*, 2(3), 2021.

Songning Lai, Lijie Hu, Junxiao Wang, Laure Berti-Equille, and Di Wang. Faithful vision-language interpretation via concept bottleneck models. In *International Conference on Learning Representations*, volume 2024, pp. 23756–23779, 2024.

Songning Lai, Mingqian Liao, Zhangyi Hu, Jiayu Yang, Wenshuo Chen, Hongru Xiao, Jianheng Tang, Haicheng Liao, and Yutao Yue. Learning new concepts, remembering the old: Continual learning for multimodal concept bottleneck models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 12314–12322, 2025.

Karim Lekadir, Alejandro F Frangi, Antonio R Porras, Ben Glocker, Celia Cintas, Curtis P Langlotz, Eva Weicken, Folkert W Asselbergs, Fred Prior, Gary S Collins, et al. Future-ai: international consensus guideline for trustworthy and deployable artificial intelligence in healthcare. *bmj*, 388, 2025.

Hongbin Lin, Chenyang Ren, Juangui Xu, Zhengyu Hu, Cheng-Long Wang, Yao Shu, Hui Xiong, Jingfeng Zhang, Di Wang, and Lijie Hu. Controllable concept bottleneck models. *arXiv preprint arXiv:2601.00451*, 2026.

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.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. A survey on knowledge editing of neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11759–11775, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022a.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International conference on machine learning*, pp. 15817–15831. PMLR, 2022.

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

Liangzu Peng, Aditya Chattopadhyay, Luca Zancato, Elvis Nunez, Wei Xia, and Stefano Soatto. Gated kalmanet: A fading memory layer through test-time ridge regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20212–20222, 2026.

James Robins, Lingling Li, Rajarshi Mukherjee, Eric Tchetgen Tchetgen, and Aad van der Vaart. Higher order estimating equations for high-dimensional models. *Annals of statistics*, 45(5):1951, 2017.

Yelena Rozenfeld, Jennifer Beam, Haley Maier, Whitney Haggerson, Karen Boudreau, Jamie Carlson, and Rhonda Medows. A model of disparities: risk factors associated with covid-19 infection. *International journal for equity in health*, 19(1):126, 2020.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Naveed Sattar, Iain B McInnes, and John JV McMurray. Obesity is a risk factor for severe covid-19 infection: multiple potential mechanisms. *Circulation*, 142(1):4–6, 2020.

Ivaxi Sheth and Samira Ebrahimi Kahou. Auxiliary losses for learning generalizable concept-based models. *Advances in Neural Information Processing Systems*, 36:26966–26990, 2023.

Andong Tan, Fengtao Zhou, and Hao Chen. Explain via any concept: Concept bottleneck model with open vocabulary concepts. In *European Conference on Computer Vision*, pp. 123–138. Springer, 2024.

Johannes von Oswald, Nino Scherrer, Seijin Kobayashi, Luca Versari, Songlin Yang, Maximilian Schlegel, Kaitlin Maile, Yanick Schimpf, Oliver Sieberling, Alexander Meulemans, et al. Mesanet: Sequence modeling by locally optimal test-time training. In *International Conference on Learning Representations*, volume 2026, pp. 26397–26447, 2026.

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

Jenna Wiens, Suchi Saria, Mark Sendak, Marzyeh Ghassemi, Vincent X Liu, Finale Doshi-Velez, Kenneth Jung, Katherine Heller, David Kale, Mohammed Saeed, et al. Do no harm: a roadmap for responsible machine learning for health care. *Nature medicine*, 25(9):1337–1340, 2019.

Yifan Wu, Yang Liu, Yue Yang, Michael S Yao, Wenli Yang, Xuehui Shi, Lihong Yang, Dongjun Li, Yueming Liu, Shiyi Yin, et al. A concept-based interpretable model for the diagnosis of choroid neoplasias using multimodal data. *Nature communications*, 16(1):3504, 2025.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *International Conference on Learning Representations*, volume 2024, pp. 6944–6966, 2024.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19187–19197. IEEE, 2023.

Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare. *Nature biomedical engineering*, 2(10):719–731, 2018.

Lu Yu, Haoyu Han, Zhe Tao, Hantao Yao, and Changsheng Xu. Language guided concept bottleneck models for interpretable continual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14976–14986. IEEE, 2025.

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

A Detailed Theoretical Derivations

A.1 The Appended Objective and the Influence-Function Connection

The optimization variable is $\vartheta = \operatorname{vec} W_{\mathcal{A}} \in \mathbb{R}^p$, where $p = q_{\mathcal{A}}(h+1)$ includes every optimized bias coordinate. The objective $J_{\mathcal{A}}$ in Eq. 4 uses a mean over samples, with a fixed convention for summing the losses of concepts added together. All gradients, curvature estimates, regularizers, and matched-reference objectives must use this same convention. The parameter norm and trust-region radius are unnormalized Euclidean quantities. The normalized gradient reported in diagnostics is $\|\nabla J_{\mathcal{A}}\|_2/\sqrt{p}$; this diagnostic normalization is not part of the update.

At an initialization ϑ_0, a quadratic expansion has the form

$$J_{\mathcal{A}}(\vartheta_0 + \Delta) \approx J_{\mathcal{A}}(\vartheta_0) + g^\top \Delta + \tfrac{1}{2}\Delta^\top H_0 \Delta, \tag{12}$$

where $g = \nabla J_{\mathcal{A}}(\vartheta_0)$ and $H_0 = \nabla^2 J_{\mathcal{A}}(\vartheta_0)$. An invertible positive-definite quadratic model gives the Newton step $-H_0^{-1} g$. CABM replaces its curvature with a structured estimate and adds numerical damping. This derivation does not require ϑ_0 to minimize an old objective, and does not integrate an infinitesimal influence derivative to a unit perturbation.

For comparison, influence-function differentiation requires a fixed-dimensional stationary path and an invertible old Hessian. Appending parameters that do not affect the old loss can instead create a singular old Hessian. Even when an influence derivative exists, applying it at a finite perturbation needs additional control along the solution path. These are different assumptions from the local appended-objective conditions of Theorem 1.

For a row-wise sigmoid head with fixed features, binary cross-entropy is convex in the appended affine parameters. Strict positive curvature still requires conditions on the features, predictions, or explicit regularization; unregularized separable logistic regression need not have a finite minimizer. For embedding-based readouts, neither convexity nor strong convexity follows merely from freezing the encoder. Theorem 1 is conditional on its stated assumptions, and the reported solution-gap diagnostic concerns the row-wise SkinCon block.

A.2 EK-FAC Construction, Dimensions, and Approximation Scope

Let $d = h + 1$, $q = q_{\mathcal{A}}$, and use column-major vectorization. Each sample's data-loss gradient has the outer-product form $s_i \bar{\phi}_i^\top \in \mathbb{R}^{q \times d}$. Thus

$$v_i = \bar{\phi}_i \otimes s_i, \qquad F = \frac{1}{n} \sum_i (\bar{\phi}_i \bar{\phi}_i^\top) \otimes (s_i s_i^\top). \tag{13}$$

The K-FAC approximation replaces this joint second moment by the product of its two empirical second moments, $D \otimes B$. It is an approximation to the empirical Fisher, not an equality with the Hessian. In particular, using observed-label gradient outer products is different from taking an expectation over labels sampled from the predictive distribution. The observed empirical Fisher and the loss Hessian can differ even before factorization.

With $Q = Q_D \otimes Q_B$, EK-FAC replaces the diagonal of the curvature in this orthonormal basis by

$$\operatorname{diag}(Q^\top F Q) = \operatorname{vec} \Lambda^*, \qquad \Lambda^* = \frac{1}{n} \sum_i (Q_B^\top s_i \bar{\phi}_i^\top Q_D) \odot (Q_B^\top s_i \bar{\phi}_i^\top Q_D). \tag{14}$$

Off-diagonal entries in this basis are discarded. This eigenvalue correction distinguishes EK-FAC (George et al., 2018) from using only the inverse factors $B^{-1} G D^{-1}$ of K-FAC (Martens & Grosse, 2015). For an isotropic objective penalty, G includes its gradient ρW, while its Hessian contributes ρI separately from damping λI. For a non-isotropic penalty (for example, one excluding bias coordinates), use its actual Hessian in the solve; adding a single ρ to all eigenvalues is not generally equivalent.

The dense matrix is $p \times p$ with $p = qd$. Storing the two eigensystems and corrected spectrum requires $O(d^2 + q^2 + qd)$ values rather than $O(q^2 d^2)$. Forming factors costs $O(n(d^2 + q^2))$ and their eigendecompositions cost $O(d^3 + q^3)$. Forming the corrected spectrum also requires transforming per-sample features and output gradients and accumulating qd entries; these operations are not

eliminated by the factorization. Cached feature-side quantities may be reused when the encoder and sample set are unchanged. For $q = 1$, dense feature-side storage remains $O(d^2)$, so single-concept experiments do not establish an asymptotic storage advantage.

The reported CelebA configuration uses $t = 2$, $h = 512$, embedding size $m = 32$, and actual appended width $q_{\mathcal{A}} = 128$, hence $p = 65{,}664$. The binary-row formula $p = t(h + 1)$ applies only when $q_{\mathcal{A}} = t$.

Approximation can be poor when feature/output-gradient dependence is strong, off-diagonal curvature in the Kronecker eigenbasis is substantial, or the empirical Fisher poorly represents the Hessian. Ill-conditioned factors and large or redundant appended blocks can also make the update sensitive. A concept absent from the fixed representation cannot generally be recovered by a head-only edit. Neither damping nor clipping removes these limitations.

A.3 Label Predictor Adaptation

For ridge regression, the normal equations for the expanded design are

$$\begin{bmatrix} A & U \\ U^\top & T \end{bmatrix} \begin{bmatrix} V_C \\ V_A \end{bmatrix} = \begin{bmatrix} C^\top Y/n \\ Z^\top Y/n \end{bmatrix}, \qquad S = T - U^\top A^{-1} U. \tag{15}$$

With $\rho > 0$, the full Gram matrix, A, and S are positive definite. Eliminating V_C gives Eq. 8. Equivalently,

$$\begin{bmatrix} A & U \\ U^\top & T \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1} U S^{-1} U^\top A^{-1} & -A^{-1} U S^{-1} \\ -S^{-1} U^\top A^{-1} & S^{-1} \end{bmatrix}. \tag{16}$$

Here A is $d_C \times d_C$, whereas $Z^\top Z$ is $d_A \times d_A$; these matrices cannot be added to describe column augmentation. In practice, cached factorizations and linear solves can replace explicit inverses. The complexity in the main text excludes construction of $C^\top Z$ and $Z^\top Z$, which costs $O(n d_C d_A + n d_A^2)$, and the output-dependent right-hand-side calculations. The ridge solution reoptimizes both old and new coefficients. It is exact for this fixed-design ridge objective, not for a model whose representation is retrained.

The default linear-softmax variant instead fixes V_C, b and optimizes the newly appended V_A, initialized at zero. Its per-sample gradient is an outer product of the new input z_i and the softmax residual, so its own empirical-Fisher EK-FAC factors can be constructed by the same procedure. Apply numerical damping and radial clipping to this block separately. A single such update need not equal a converged logistic regression solution.

B Theoretical Proofs

B.1 Proof of Theorem 1

Let $e = \vartheta_0 - \vartheta^*$. Strong convexity and stationarity give

$$\mu \|e\|_2^2 \le \langle \nabla J_{\mathcal{A}}(\vartheta_0) - \nabla J_{\mathcal{A}}(\vartheta^*), e \rangle = \langle g, e \rangle \le \|g\|_2 \|e\|_2, \tag{17}$$

so $\|e\|_2 \le \|g\|_2/\mu$. The exact Newton step at ϑ_0 satisfies

$$\vartheta_0 - H_0^{-1} g - \vartheta^* = H_0^{-1} \int_0^1 [H_0 - \nabla^2 J_{\mathcal{A}}(\vartheta^* + se)] e \, ds, \tag{18}$$

$$\|\vartheta_0 - H_0^{-1} g - \vartheta^*\|_2 \le \frac{M}{2\mu} \|e\|_2^2 \le \frac{M}{2\mu^3} \|g\|_2^2. \tag{19}$$

The line segment is contained in the assumed convex neighborhood. Replacing H_0^{-1} by P adds at most $\delta \|g\|_2$. Replacing the raw step by the clipped step adds at most $\|\Delta - \Delta^{\mathrm{raw}}\|_2$. The triangle inequality yields Eq. 10. No assumption that the clipped iterate is stationary is used.

What the inverse-curvature error includes. Let $K = \widetilde{F} + \nabla^2 R_{\mathcal{A}}(\vartheta_0) + \lambda I$ be positive definite with $K \succeq \nu I$, so $P = K^{-1}$. The inverse identity gives

$$\|P - H_0^{-1}\|_2 \le \frac{\|H_0 - K\|_2}{\mu\nu} \le \frac{\|H_{\mathrm{data}} - F\|_2 + \|F - \widetilde{F}\|_2 + \lambda}{\mu\nu}, \tag{20}$$

where H_{data} is the exact data-loss Hessian at the initialization. If the explicit-regularizer Hessian is itself approximated, its approximation error must also be included. Consequently, a dense empirical-Fisher comparator isolates neither the complete Hessian approximation error nor the damping error. A measured solution gap evaluates the actual approximation against a matched solution but does not verify every assumption or constant of the theorem.

Warm-start conditions. The bound is informative when the same-objective stationary reference is in a neighborhood with controlled curvature and $\|g\|_2$ is small relative to μ and M. Concept-conditional initialization is a heuristic for reaching such a regime; it is not a proof that this happens. Under Gaussian class-conditional features with common isotropic covariance, a discriminant slope is proportional to the centroid difference, but its intercept also contains a difference of squared centroid norms. The prevalence-only intercept in Eq. 9 omits this term. We therefore do not identify this initializer with an exact Bayes or logistic optimum.

B.2 Proof of Proposition 2

Taylor's theorem with an M-Lipschitz Hessian gives

$$J_{\mathcal{A}}(\vartheta_0 + \Delta) - J_{\mathcal{A}}(\vartheta_0) = g^\top \Delta + \tfrac{1}{2}\Delta^\top H_0 \Delta + R_3, \qquad |R_3| \le \tfrac{M}{6}\|\Delta\|_2^3. \tag{21}$$

Subtracting the quadratic model using $\widetilde{H}$ and applying the spectral-norm bound yields Eq. 11. This controls model error; it does not assert that every feasible step decreases the objective.

For the intervention claim, write $\ell_b = V_C^\top c^{(j \leftarrow b)} + b_0$ for the unchanged old logits under intervention $b \in \{0, 1\}$, and set $a = V_A^\top z$. Then

$$I_j(V_A) = \mathrm{softmax}(\ell_1 + a) - \mathrm{softmax}(\ell_0 + a). \tag{22}$$

The softmax Jacobian is $\mathrm{diag}(p) - pp^\top$ and has spectral norm at most $1/2$: its maximum absolute row sum is $2p_i(1 - p_i) \le 1/2$, and it is symmetric. Thus each probability vector changes by at most $\|a\|_2/2$. The triangle inequality and $\|V_A^\top z\|_2 \le \|V_A\|_F \|z\|_2$ prove

$$\|I_j(V_A) - I_j(0)\|_2 \le \|a\|_2 \le \|z\|_2 r_f. \tag{23}$$

This holds with new concept predictions fixed during the old-concept intervention. It does not apply unchanged to a nonlinear head, an intervention that also changes z, or the ridge variant that refits V_C. The empirical normalized aggregate intervention metric is a related diagnostic, not numerically the same quantity as this per-input absolute bound.

Finally, old concept-output invariance requires no curvature argument: evaluating the same fixed encoder, old parameters, and old readout gives the same old output. This structural fact alone says nothing about preservation of downstream probabilities. Radial clipping enforces $\|\Delta\|_2 \le r$ explicitly. A fixed damped solve can violate this radius, and radial rescaling is generally not the exact minimizer of a non-isotropic quadratic under a Euclidean radius constraint.

C Algorithm and Pseudocode

Algorithm 1 CABM with blockwise EK-FAC and explicit radius enforcement

Require: Trained model; training inputs, labels, and new concept labels; concept-block initialization; explicit objective regularizers; numerical damping λ; radii r_g, r_f.
Ensure: Expanded concept predictor and adapted label head.

```
Freeze the encoder and old concept pathway; cache features φ̄_i and old head inputs C.
Initialize W_A^(0); use Eq. 9 for binary concept rows.
Compute G_A = ∇J_A(W_A^(0)) with the specified objective regularizer.
Compute sample output gradients s_i, factors D, B, and eigenvectors Q_D, Q_B.
Compute corrected diagonal Λ* using Eq. 5.
Solve for the damped raw step; Eq. 6 applies to isotropic regularization.
Clip the raw concept step using Eq. 7; update only W_A.
Compute the appended label-head inputs Z from the updated concept block.
if ridge-head variant then
    Form U, T, S; compute V_A, V_C by Eq. 8 with the ridge penalty ρ.
else
    Keep old softmax coefficients and bias fixed; initialize new coefficients V_A = 0.
    Compute the label-loss gradient and label-block EK-FAC factors on Z.
    Compute a damped raw step and radially clip it to Frobenius radius r_f.
    Set V_A to the clipped label-head update.
end if
Return the expanded model.
```

D Additional Evaluation and Diagnostics

D.1 Concept Accuracy and Preservation

For SkinCon, Derm7pt, CUB, and AwA2, *All Concept Acc.* is the macro-average of binary prediction accuracy over the concept outputs available in the evaluated model. This is an average over k outputs for the base model and $k + t$ outputs after expansion. We distinguish this metric from accuracy on the appended concepts alone. We also distinguish old-concept accuracy evaluated with the original thresholds from old-concept accuracy after validation-set threshold recalibration. In Table 2, the expanded-set accuracy is $A_{\text{all}} = (kA_{\text{old,recal}} + tA_{\text{new}})/(k + t)$, with $(k, t) = (21, 1)$ for SkinCon and $(6, 1)$ for Derm7pt.

The encoder and all pre-existing concept-prediction parameters are frozen. Consequently, with evaluation mode, the same input features, and the original thresholds, old concept scores and binary predictions are unchanged up to floating-point precision. Changes in recalibrated accuracy are evaluation effects rather than changes to the old predictor.

Table 2: Separating old, new, and all-concept accuracy (%). The fixed-threshold columns directly test old prediction preservation.

Dataset	Model	Old fixed	Old recalibrated	New	All
SkinCon	Base	88.77	88.77	–	88.77
	CABM	88.77	89.32	89.54	89.33
Derm7pt	Base	74.45	74.45	–	74.45
	CABM	74.45	74.89	75.37	74.96

Exact score preservation does not imply unchanged downstream effects. We therefore evaluate the normalized L_1 change in the probability-level intervention effects of old concepts after label-head expansion, denoted D_{int}. Table 3 reports this behavior together with new-concept and label accuracy. CABM's drift is lower than retraining in both medical datasets; the old predictor's structural invariance and the label head's empirical intervention stability are separate properties.

Table 3: Old-concept intervention stability and learning of appended concepts. Accuracy is in percent; smaller D_{int} means less change in intervention effects. Old accuracy uses recalibrated thresholds.

Dataset	Method	Old acc.	New acc.	D_{int}	Label acc.
SkinCon	Retraining	87.85	86.53	0.068	72.99
	Partial-FT	89.20	40.14	0.011	71.79
	CABM	89.32	89.54	0.006	73.01
Derm7pt	Retraining	74.76	74.35	0.091	80.23
	Partial-FT	74.86	61.49	0.014	78.49
	CABM	74.89	75.37	0.009	80.55

D.2 CelebA: Incomplete Bottleneck and Decoded Attributes

Eight approximately balanced binary attributes define an eight-bit vector and a 256-way downstream class. The base model uses four annotated attributes $a_{1:4}$, two more annotated attributes $a_{5:6}$ are added together, and two task-relevant attributes $a_{7:8}$ remain unannotated. The uniform chance accuracy is $1/256 \approx 0.39\%$. Both CABM and retraining receive the same six annotated attributes after expansion.

Concept accuracy is the macro-average of binary accuracy at the explicitly modeled concept outputs. The base model obtains 51.08% over four concepts; its old concept predictions remain unchanged under CABM. The two appended concepts obtain 68.15%, giving $(4{\times}51.08{+}2{\times}68.15)/6 = 56.77\%$ over the expanded six-concept set. These concept-head measurements differ from decoding the predicted downstream class into its eight constituent bits.

Table 4 reports exact-match accuracy for the two added bits and the two unannotated bits decoded from the predicted 256-way class. The uniform chance level for either two-bit group is 25%. The two added bits are recovered substantially more accurately than the still-unannotated bits. A possible explanation for CABM's small test-accuracy advantage is finite-sample regularization from its localized update under an incomplete bottleneck; this is not a guarantee that a one-step update outperforms full optimization.

Table 4: CelebA label and decoded-attribute results (%). Label standard deviations are across three concept-selection settings. CABM is deterministic conditional on the base checkpoint and setting.

Method	256-way label acc.	Added bits exact match	Unannotated bits exact match
Base	41.91 ± 0.35	48.81	47.60
Retraining	42.78 ± 0.61	95.54	48.81
Partial-FT	42.66 ± 0.47	95.16	47.27
CABM	43.66 ± 0.32	96.03	48.92

D.3 Warm Start and Trust-Region Behavior

For each binary concept a, concept-conditional initialization uses training-feature means conditioned on the presence or absence of that concept:

$$\mu_{a,1} = \mathbb{E}[\phi(x) \mid \widetilde{c}_a = 1], \qquad \mu_{a,0} = \mathbb{E}[\phi(x) \mid \widetilde{c}_a = 0],$$
$$w_a^{(0)} = \alpha(\mu_{a,1} - \mu_{a,0}), \qquad b_a^{(0)} = \log \frac{\pi_a}{1-\pi_a}, \qquad \pi_a = \Pr(\widetilde{c}_a = 1). \tag{24}$$

This is an isotropic-covariance LDA-inspired centroid-difference heuristic, with a prevalence-based intercept, rather than an optimal solution. The conditioning groups are defined by the concept annotation, not the downstream class.

For the SkinCon analysis, the editable block contains one concept row and its bias, so $p_A = 2049$. We normalize its gradient norm by $\sqrt{p_A}$. With radius $r = 0.1$, the raw step is projected radially:

$$\Delta\vartheta_A^{\text{clip}} = \alpha_{\text{clip}} \Delta\vartheta_A^{\text{raw}}, \qquad \alpha_{\text{clip}} = \min\left\{1, \frac{r}{\|\Delta\vartheta_A^{\text{raw}}\|_2}\right\}, \tag{25}$$

where $\alpha_{\text{clip}} = 1$ for a zero raw step. Tables 5 and 6 aggregate 110 concept–seed edits. The prior-only ablation reports 68.28% label accuracy, compared with 73.01% for full CABM. Concept-conditional initialization reduces both the gradient magnitude and the severity of projection. Its post-update gradient need not vanish because the curvature is approximate and the step can be clipped.

Table 5: SkinCon warm-start ablation. Gradient norms include the appended bias coordinates and are divided by $\sqrt{p_A}$. Runtime uses synchronized measurements with the same deterministic batched solver.

Initialization and update	Pre-gradient	Post-gradient	Label acc. (%)	Time (s)
Zero + EK-FAC	1.916	0.5081	58.72	32.74
Global prior + EK-FAC	0.2613	0.1130	65.24	32.11
Concept-conditional only	0.1702	–	68.28	27.40
Concept-conditional + EK-FAC	0.1702	0.08914	73.01	32.15

Table 6: Trust-region activation and severity on SkinCon. The raw norm and radial scale are medians; the normalized clipping residual is a mean. Clipping frequency alone does not describe its severity.

Initialization	Clipped / total	Clipped (%)	Raw norm	α_{clip}	Residual
Zero	110 / 110	100.0	0.428	0.234	0.0076
Global prior	92 / 110	83.6	0.184	0.543	0.0022
Concept-conditional	61 / 110	55.5	0.108	0.926	0.0005

D.4 Curvature Approximation and Update Dimensions

The number of added concepts t need not equal the output width q_A of the optimized block. With feature dimension h and a bias coordinate, dense curvature has dimension $p = q_A(h + 1)$. In the CelebA embedding configuration, $m = 32$ and $q_A = 128$ for two appended concepts. In the row-wise configurations, $q_A = t$. Table 7 lists representative dimensions. For single-concept SkinCon and Derm7pt edits, the feature-side factor is as large as the dense block, so these settings do not provide an asymptotic factor-storage advantage.

Table 7: Appended-block dimensions and FP32 curvature storage. The last column counts the two factor matrices, not peak memory. EK-FAC additionally stores $q_A(h + 1)$ corrected eigenvalues; feature caches, eigensystems, and solver workspaces are excluded. Units are decimal MB and GB.

Setting	t	h	q_A	p	Dense	Factors
SkinCon	1	2048	1	2049	16.79 MB	16.79 MB
Derm7pt	1	2048	1	2049	16.79 MB	16.79 MB
CelebA	2	512	128	65664	17.25 GB	1.12 MB
AwA2, +5	5	512	5	2565	26.32 MB	1.05 MB
AwA2, +15	15	512	15	7695	236.85 MB	1.05 MB
CUB, +15	15	2048	15	30735	3.78 GB	16.79 MB

On AwA2 with five added concepts, we construct the unfactorized empirical Fisher

$$F_{\text{dense}} = \frac{1}{n} \sum_{i=1}^{n} v_i v_i^\top, \qquad v_i = \text{vec}(s_i \bar{\phi}_i^\top). \tag{26}$$

We compare its damped update with EK-FAC and Diagonal using the same cached features, initialization, gradient, damping $\lambda = 10^{-3}$, trust-region radius $r = 0.1$, and label-head adaptation. Dense construction is feasible at this scale. The comparison isolates the effect of factorization relative to the empirical Fisher; it does not measure the empirical-Fisher-to-Hessian discrepancy in a Hessian-based approximation guarantee.

Table 8: AwA2 +5 curvature comparison. Direction error and cosine are measured against the dense empirical-Fisher update. Accuracy is in percent; times are in seconds. Curvature storage excludes caches and workspaces, and the EK-FAC factor count excludes its corrected diagonal.

Method	Relative error	Cosine	Concept acc.	Label acc.	Curvature + solve	Total edit	Storage
Dense empirical Fisher	0.000	1.000	97.54	90.16	4.80	336.2	26.32 MB
EK-FAC	0.17	0.984	97.52	90.15	0.24	331.6	1.05 MB
Diagonal	0.39	0.921	97.46	90.10	0.07	331.4	0.010 MB

The end-to-end accuracy advantage over Diagonal is modest. Table 9 shows that Diagonal is a competitive alternative, while EK-FAC attains slightly higher accuracy at a small additional runtime. We do not infer that structured curvature is universally necessary from these results.

Table 9: Curvature trade-offs in single- and multi-concept edits. Concept accuracy is averaged over the expanded concept set.

Setting	Method	Concept acc. (%)	Label acc. (%)	Time (s)
SkinCon, +1	EK-FAC	89.33	73.01	32.15
	Diagonal	89.18	72.56	29.47
	First-order	88.95	70.78	26.32
AwA2, +15	EK-FAC	97.58	90.16	452.00
	Diagonal	97.34	89.93	449.31

D.5 Matched-Scope Solution Diagnostics

For a parameter-space comparison, the converged reference must use the same feature basis and editable parameters. We therefore compare the SkinCon appended concept block, including its bias, with Converged Probe-LR on the same objective J_A. Explicit objective regularization is shared by both methods and is distinct from numerical damping. Unrestricted full retraining changes the encoder and additional parameters, so it remains a predictive-performance comparator rather than a parameter-distance reference for this diagnostic.

Table 10 reports normalized distances to the converged reference for the raw and clipped solutions (E_{raw} and E_{final}), the mean normalized clipping residual $\overline{E}_{\mathrm{clip}}$, and the relative objective gap E_{obj}. Across the 110 SkinCon concept–seed runs, final solution distance has Pearson correlations 0.74 and 0.81 with the pre-update normalized gradient and its square, respectively. These are empirical associations, not a measurement of all constants or error terms in the local bound.

Table 10: SkinCon solution diagnostics for the appended concept-predictor block. The reference row has zero gap values by definition; its residual normalized gradient is nonzero because convergence is numerical.

Method	E_{raw}	E_{final}	$\overline{E}_{\mathrm{clip}}$	E_{obj}	Post-gradient
Converged Probe-LR	0.0000	0.0000	0.0000	0.0000	0.0008
CABM–EK-FAC	0.0214	0.0217	0.0005	0.0029	0.08914
Diagonal	0.0481	0.0486	0.0008	0.0081	0.1287

D.6 Correlated and Redundant Concepts

For each appended concept a, we compute its maximum absolute correlation with an existing annotation,

$$\rho_a = \max_{j \le k} \left| \mathrm{corr}(\widetilde{c}_a, c_j) \right| . \tag{27}$$

The 22 SkinCon leave-one-out additions are divided into low-, medium-, and high-correlation tertiles containing 8, 7, and 7 additions. We also evaluate one random concept and one exact duplicate.

Correlation cannot alter the frozen old concept scores, but it can affect fitting of the appended block and identifiability of downstream attribution.

Table 11: SkinCon correlation and redundancy diagnostics. Accuracy is in percent. The last column is the reported weight assigned to the appended concept in the label head.

Added concept	Count	Mean ρ_a	D_{int}	New acc.	Label acc.	Weight
Random	1	0.04	0.001	50.10	72.94	0.002
Low correlation	8	0.18	0.004	90.42	73.05	0.031
Medium correlation	7	0.43	0.006	89.63	73.02	0.026
High correlation	7	0.71	0.011	88.54	72.98	0.019
Exact duplicate	1	1.00	0.014	89.32	72.97	0.008

Across the correlation tertiles, new-concept accuracy decreases from 90.42% to 88.54% and intervention drift rises from 0.004 to 0.011, while label accuracy varies by 0.07 percentage points. The exact duplicate returns to the base label accuracy of 72.97%. Perfect redundancy can preserve predictions while making separate concept-to-label attribution non-identifiable without additional constraints or supervision.

D.7 Timing and Repeated-Edit Limitations

The converged frozen-feature baseline includes both appended-block optimization and label-head adaptation. It uses the same cached feature representation and downstream update as CABM. Table 12 separates speedups against this matched baseline, Partial-FT, and full retraining. The comparison with full retraining includes the additional cost of optimizing the encoder and is therefore distinct from the frozen-feature comparisons.

Table 12: CABM runtime and speedup by comparator. Ratios are computed from the reported mean times and rounded to one decimal place.

Dataset	CABM (s)	vs. Retraining	vs. Partial-FT	vs. Probe-LR
SkinCon	32.15	10.8×	3.9×	1.5×
Derm7pt	27.7	12.4×	4.1×	1.4×
CUB	194.0	61.5×	3.7×	1.5×
CelebA	46.8	31.0×	19.1×	1.5×
AwA2	395.3	18.2×	10.6×	1.2×

Repeated additions do not accumulate score-level changes in previously introduced concepts, since their predictor parameters are frozen at each later edit. This does not guarantee bounded cumulative downstream drift or uniformly accurate local approximations. An incomplete frozen representation, near-duplicate concepts, a large pre-update gradient, ill-conditioned curvature, repeated severe clipping, or very large appended blocks can all make one-step expansion unreliable.

Useful diagnostics include old–new concept correlation, normalized gradient magnitude, curvature conditioning, raw step norm, clipping scale, downstream prediction drift, and old-concept intervention-effect drift. If the local approximation is inadequate, iterative block optimization, representation adapters, or full retraining are appropriate alternatives. Existing sequential accuracy and timing experiments provide finite-length empirical evidence, not a guarantee for arbitrarily many edits.

E Dataset and Experiment Details

E.1 Datasets

SkinCon. SkinCon (Daneshjou et al., 2022) is a dermatology benchmark containing high-resolution clinical images annotated with fine-grained dermatologic concepts and lesion diagnoses. Although the full dataset is larger, only a subset of **2,250 images** includes expert-verified dense concept annotations. These concept-labeled samples contain binary clinical concepts describing morphological and

Table 13: Summary of datasets and concept addition protocols.

Dataset	Domain	Images	Concept Setting	Addition Protocol
SkinCon	Dermatology	2,250 concept-labeled	22 clinical concepts	Leave-one-out single concept addition
Derm7pt	Dermoscopy	1,011	7 dermoscopic concepts	Leave-one-out single concept addition
CUB	Birds	11,788	112 visual attributes	Batch semantic group addition
CelebA	Faces	>200K	8 defining attributes / 256 labels	Add 2 held-out concepts simultaneously
AwA2	Animals	37,322	85 attributes	Add back 5/10/15 removed concepts

chromatic attributes, together with a 3-way disease label: *non-neoplastic*, *benign*, and *malignant*. We use a curated subset of **22 concepts** chosen for sufficient frequency:

> Papule, Plaque, Pustule, Bulla, Patch, Nodule, Ulcer, Crust, Erosion, Atrophy, Exudate, Telangiectasia, Scale, Scar, Friable, Dome-shaped, Brown, White, Purple, Yellow, Black, and Erythema.

Derm7pt. Derm7pt (Kawahara et al., 2018) is a dermoscopic benchmark based on the classical 7-point checklist used in melanoma diagnosis. It contains **1,011** dermoscopic images annotated with seven clinically meaningful dermoscopic criteria:

> pigment network, streaks, pigmentation, regression structures, dots and globules, blue whitish veil, and vascular structures.

The downstream task is binary disease classification: *melanoma* vs. *non-melanoma*.

CUB-200-2011. CUB (Wah et al., 2011) contains **11,788 images** across **200 bird species**. Following standard CBM preprocessing (Koh et al., 2020), we denoise instance-level attributes by aggregating them into class-level majority votes and retain **112 concepts**. For batch concept addition, we isolate semantic concept groups such as *Wing Color*, *Bill Shape*, *Tail Pattern*, *Upperparts Color*, and *Underparts Color*.

CelebA. CelebA (Liu et al., 2015) is a large-scale facial attribute dataset with over 200K images. Following the concept embedding model protocol (Espinosa Zarlenga et al., 2022), eight binary attributes encode a 256-way downstream class. Six attributes are available as explicit concept annotations; two further task-relevant attributes remain unannotated. The base CBM is trained on 4 randomly selected concepts, and the remaining 2 concepts are inserted simultaneously.

AwA2. Animals with Attributes 2 (AwA2) (Xian et al., 2018) contains **37,322 images**, **50 animal classes**, and **85 semantic attributes**. Following the protocol used by energy-based CBMs (Xu et al., 2024), we remove 5, 10, and 15 concepts from the base model and evaluate adding them back.

E.2 Concept Addition Protocols and Metrics

Concept addition protocols. For **single-concept addition** on SkinCon and Derm7pt, we use a leave-one-out protocol. A base CBM is trained with one concept removed, and the held-out concept is later inserted post hoc. This yields 22 addition settings for SkinCon and 7 addition settings for Derm7pt.

For **batch concept addition** on CUB, we remove semantic concept groups from the base CBM and add each group back in one update. The main group-wise evaluation uses *Wing Color*, *Bill Shape*, and *Tail Pattern*; the sequential addition experiment further evaluates *Upperparts Color* and *Underparts Color*.

For **large-scale concept addition** on AwA2, we remove 5, 10, or 15 concepts from the base model and evaluate adding them back. For **CelebA**, the base model is trained with 4 concepts, and the remaining 2 concepts are inserted simultaneously.

Metrics. We report downstream classification accuracy (**Label Acc.**), concept prediction accuracy (**Concept Acc.**), and wall-clock update time in seconds (**Runtime**). Runtime includes feature/statistics

caching, curvature estimation, and parameter update for CABM, and full optimization time for gradient-based baselines. All methods start from the same trained base CBM and are evaluated on the same test split. For the main table, we report averages over all concept-addition settings within each dataset. Detailed per-concept and per-group results are reported in Appendix E.6.

E.3 Baselines and Adaptation Details

Original Base Model. The original base model is trained with the held-out concept or concept group removed. It is not given access to the newly added concept during training and therefore serves as the pre-edit reference model.

Retraining. Retraining expands the CBM architecture to include the new concept outputs and then re-optimizes the expanded concept predictor and label predictor. It serves as a strong reference solution, but incurs the largest computational cost. We refer to it as a retraining reference rather than a strict upper bound, because an optimization reference need not give the highest held-out accuracy. A regularization effect is one possible explanation for small differences, not an established mechanism.

FastIF. FastIF (Guo et al., 2021) approximates influence-function updates without explicit Hessian inversion. We adapt it by applying the influence update to the newly appended concept parameters and the corresponding label-head block.

Partial-FT. Partial-FT freezes the feature extractor and old concept parameters, appends the new concept outputs, and fine-tunes only the newly introduced concept parameters and the corresponding label-head weights using gradient descent.

Converged Probe-LR. The matched frozen-feature baseline optimizes the same appended-concept objective with L-BFGS to convergence, then applies the same downstream update as CABM. Data splits, features, new concept annotations, and explicit objective regularization are shared. Numerical damping is specific to the one-step solver and is not used as a substitute for objective regularization.

DER++. DER++ (Buzzega et al., 2020) is adapted as a replay-based continual learning baseline. The newly added concept labels define the new task, while replay/distillation terms are used to reduce changes to the old concept and label predictions.

CONCIL. CONCIL (Lai et al., 2025) uses recursive analytic updates for the concept and decision layers, formulated as linear regression problems. We retain the original recursive update procedure, treating each newly inserted concept or concept group as a new concept-learning phase while keeping the downstream label space fixed.

CLG-CBM. CLG-CBM (Yu et al., 2025) is adapted as a modern CBM-specific continual learning baseline. Each added concept or concept group is treated as a continual concept task. The class label set remains fixed, while the concept layer is expanded to include the newly added concepts.

CI-CBM. CI-CBM (Javadi et al., 2026) is originally designed for class-incremental interpretable continual learning. We adapt it to our concept-incremental setting by replacing class expansion with concept-output expansion while preserving the original downstream label set. Its concept regularization and pseudo-concept generation terms are used to preserve previous concept-level reasoning during the new concept addition phase.

CABM. CABM performs a one-shot curvature-corrected structural edit. It freezes the old feature extractor and old concept pathway, initializes the newly added concept block with concept-conditional initialization, and applies trust-region-controlled EK-FAC updates to the new concept and default softmax blocks. The ridge variant uses the exact augmented fixed-design solution.

E.4 Training Details

All experiments are implemented in PyTorch and executed on a single NVIDIA A800 GPU. For full retraining and gradient-based baselines, including Partial-FT, DER++, CLG-CBM, and CI-CBM,

we use Adam with learning rate 1×10^{-4} and batch size 32 unless otherwise specified by the original baseline implementation. For these gradient-based methods, training is run for 100 epochs on SkinCon, 80 epochs on Derm7pt, and 800 epochs on CUB. For CelebA and AwA2, we follow the corresponding CEM and ECBM-style protocols. CONCIL instead uses its original recursive analytic updates for the concept and decision layers; the Adam settings and epoch budgets above do not apply to its incremental updates.

SkinCon and Derm7pt detailed tables report mean $\pm$ standard deviation over 5 random seeds. CelebA results average three concept-selection settings, not three stochastic CABM edits. AwA2 results average three runs. All methods are evaluated on the same train/test split under the same concept-addition protocol.

For CABM, no iterative end-to-end retraining is performed. All edits are computed directly from the trained base model using the blockwise EK-FAC updates or the exact ridge solution described in the main paper. Numerical damping is $\lambda = 10^{-3}$, distinct from explicit objective regularization. The approximate concept and softmax-head updates use separate raw parameter radii $r_g = r_f = 0.1$. No learning rate, training epoch, optimizer schedule, or replay buffer is required for CABM.

E.5 Implementation Details

Curvature and normalization. We use the sample-mean objective of Eq. 4. The per-sample gradients define an empirical Fisher; Eqs. 5–6 include its EK-FAC eigenvalue correction. Biases are included by augmenting each feature vector with one. Explicit objective regularization is distinct from numerical damping and is shared with the matched converged-probe objective. The normalized gradient diagnostic divides the Euclidean gradient norm by $\sqrt{p}$, where p counts all optimized appended parameters including biases.

Damping and radius enforcement. The reported configuration uses numerical damping $\lambda = 10^{-3}$ and separate radii $r_g = r_f = 0.1$. Each raw approximate update is radially rescaled when its unnormalized Euclidean norm exceeds its radius. For matrix-shaped blocks this is the Frobenius norm. Clipping frequency, radial scale, and normalized clipping residual are reported separately: frequent clipping can still be mild. The initialization ablation uses a deterministic batched direct solve and synchronized GPU timing with common timing boundaries.

Label-head adaptation. The default predictor is linear-softmax. Its old coefficients and bias stay fixed, while the new input coefficients receive one damped, clipped EK-FAC update from zero. The "Only edit g" control instead uses a randomly initialized, frozen head extension, as specified in Appendix E.7 together with the matched zero-initialized no-head-update control. The ridge variant instead computes the exact solution of its expanded fixed-design ridge objective through the Schur complement in Eq. 8. This can change old head coefficients and has a different scope from the default appended-only softmax update. The ridge-head validation is reported in Table 20.

E.6 Detailed Concept Addition Results

This section reports the per-concept and per-group measurements. For SkinCon and Derm7pt, we report leave-one-out concept addition results over 5 random seeds. For CUB, we report semantic group addition results. For AwA2, we report larger-scale 5/10/15-concept addition results.

Medical leave-one-out concept addition. Tables 14 and 15 provide detailed SkinCon and Derm7pt results. Each column corresponds to a held-out concept that is removed from the base CBM and then added back post hoc.

Table 14: SkinCon: detailed leave-one-out concept addition performance over 5 random seeds. We report label accuracy, concept accuracy, and runtime for each newly added concept as mean ± standard deviation.

	Papule	Plaque	Pustule	Bulla	Patch	Nodule	Ulcer	Crust	Erosion	Atrophy	Exudate	Telang.
Label accuracy (↑, %)												
Original	72.94	73.41	72.94	72.63	73.10	73.25	72.94	72.96	72.78	73.10	72.63	72.62
Retrain	73.10±0.348	72.81±0.021	73.26±0.039	72.72±0.085	73.15±0.070	73.48±0.071	72.93±0.069	72.93±0.220	72.64±0.010	73.21±0.070	72.52±0.121	72.79±0.139
FastIF	70.15±0.054	69.51±0.080	71.02±0.017	71.80±0.255	68.28±0.075	70.53±0.067	71.05±0.031	70.05±0.587	69.80±0.132	72.27±0.071	71.72±0.189	70.87±0.008
Partial-FT	72.03±0.267	72.17±0.290	71.88±0.181	71.67±0.081	71.55±0.139	72.40±0.291	71.92±0.201	71.93±0.227	71.67±0.158	72.14±0.302	70.59±0.244	71.24±0.172
DER++	72.81±0.210	72.01±0.350	71.97±0.411	71.36±0.235	72.24±0.285	74.52±0.269	72.95±0.322	72.71±0.074	73.34±0.425	72.20±0.160	72.94±0.148	72.00±0.072
CONCIL	73.43±0.175	73.02±0.252	73.70±0.018	73.71±0.139	71.76±0.209	72.75±0.295	73.28±0.154	72.83±0.137	72.97±0.158	73.06±0.268	72.17±0.378	72.72±0.148
CLG-CBM	71.87±0.225	72.57±0.236	72.75±0.197	72.37±0.109	72.38±0.310	72.81±0.256	73.43±0.215	72.72±0.252	74.18±0.195	72.70±0.201	73.80±0.162	72.80±0.235
CI-CBM	73.24±0.150	73.30±0.159	73.28±0.192	72.85±0.110	72.42±0.114	73.26±0.245	73.23±0.124	72.90±0.165	72.75±0.159	73.29±0.113	71.69±0.119	72.23±0.196
CABM	73.25±0.072	73.39±0.135	73.10±0.272	72.89±0.165	72.77±0.030	73.62±0.066	73.14±0.233	73.15±0.207	72.89±0.005	73.36±0.120	71.81±0.063	72.46±0.087
Concept accuracy (↑, %)												
Original	90.06	90.34	87.65	88.49	88.12	87.88	88.10	88.94	88.59	88.67	88.65	88.71
Retrain	88.32±0.235	87.92±0.079	88.54±0.371	87.68±0.184	87.27±0.193	86.90±0.368	87.25±0.303	87.91±0.616	87.40±0.576	87.86±0.612	87.69±0.251	87.58±0.411
FastIF	87.16±0.255	87.92±0.278	88.21±0.057	87.20±0.309	83.89±0.468	87.45±0.265	87.90±0.282	86.38±0.141	85.05±0.327	87.28±0.165	87.18±0.422	86.24±0.411
Partial-FT	88.76±0.367	88.06±0.256	86.12±0.269	87.02±0.335	86.97±0.317	86.22±0.247	86.85±0.284	87.34±0.210	87.01±0.193	87.20±0.317	87.16±0.199	87.32±0.360
DER++	88.13±0.242	88.93±0.269	85.18±0.109	88.99±0.154	88.60±0.182	87.98±0.352	86.31±0.228	88.18±0.177	86.18±0.221	88.99±0.141	86.70±0.287	86.52±0.299
CONCIL	89.76±0.209	89.84±0.283	86.92±0.133	87.53±0.145	88.40±0.134	87.21±0.438	87.64±0.162	87.69±0.247	88.43±0.194	88.71±0.539	88.89±0.122	88.74±0.210
CLG-CBM	90.25±0.270	88.06±0.238	88.30±0.235	88.27±0.137	88.70±0.240	87.56±0.212	87.50±0.329	87.50±0.207	87.39±0.137	90.40±0.243	90.87±0.315	89.61±0.233
CI-CBM	89.94±0.249	89.12±0.300	87.09±0.318	88.03±0.185	88.15±0.134	87.81±0.308	87.70±0.249	88.04±0.285	88.34±0.281	88.14±0.187	87.99±0.205	88.23±0.226
CABM	91.12±0.157	90.42±0.308	88.48±0.147	89.38±0.401	89.33±0.287	88.58±0.425	89.21±0.270	89.70±1.253	89.37±0.422	89.56±0.176	89.52±0.294	89.68±0.060
Runtime (↓, seconds)												
Retrain	475±32	255±14	455±78	421±52	455±61	353±39	258±19	386±42	315±33	269±38	337±26	221±61
FastIF	63.91±3.378	74.14±2.443	61.21±1.226	67.45±1.513	69.22±3.204	68.37±3.464	65.66±6.014	58.44±4.814	71.91±5.347	64.05±4.213	63.82±3.365	75.03±2.818
Partial-FT	123.17±9.257	133.40±9.415	120.47±8.327	126.71±9.358	128.48±6.507	127.63±9.052	124.92±7.367	117.70±13.751	131.17±8.988	123.31±9.058	123.08±7.683	134.29±5.901
DER++	257.44±43.90	423.70±44.80	360.26±42.50	268.69±43.70	263.07±41.80	321.62±42.10	385.96±42.00	319.30±44.30	241.07±43.80	300.28±44.70	351.06±44.70	392.19±43.80
CONCIL	216.12±15.26	203.60±28.09	207.10±27.94	250.44±16.36	218.30±10.02	262.59±19.13	281.32±25.54	235.50±22.31	291.84±20.20	229.01±15.48	264.12±17.98	297.70±13.31
CLG-CBM	251.56±21.38	207.15±33.57	254.23±21.21	254.34±16.65	222.05±19.46	255.03±25.11	180.84±15.01	169.07±31.66	235.09±27.20	176.11±21.38	184.38±28.10	174.10±23.17
CI-CBM	222.13±31.55	250.20±38.85	245.33±22.19	229.40±32.91	236.82±28.31	233.24±31.57	219.66±32.19	224.30±40.21	231.52±36.00	230.70±29.76	246.04±37.00	258.94±31.47
CABM	29.85±2.900	32.17±3.314	35.07±0.817	30.94±2.167	35.22±3.567	29.48±1.932	31.77±0.933	33.68±2.756	29.02±1.703	31.44±1.503	34.11±2.079	30.58±1.438

	Scale	Scar	Friable	Dome	Brown	White	Purple	Yellow	Black	Erythema	Avg.
Label accuracy (↑, %)											
Original	73.25	72.47	72.94	72.78	73.10	73.87	73.41	72.78	72.47	72.94	72.97
Retrain	73.38±0.311	72.60±0.209	73.09±0.110	72.92±0.303	72.94±0.211	73.64±0.303	73.69±0.622	72.63±0.330	72.40±0.069	73.01±0.414	72.99±0.188
FastIF	72.45±0.273	70.69±0.061	70.17±0.106	73.00±0.289	70.16±0.195	68.81±0.060	71.60±0.352	71.78±0.227	70.55±0.183	70.12±0.293	70.74±0.164
Partial-FT	72.13±0.168	71.56±0.220	71.64±0.212	71.87±0.195	72.02±0.237	71.79±0.169	72.33±0.080	71.64±0.249	71.41±0.244	71.80±0.140	71.79±0.203
DER++	72.41±0.111	72.08±0.235	73.00±0.189	72.57±0.220	72.75±0.212	72.61±0.270	72.55±0.428	72.00±0.388	70.65±0.438	72.23±0.450	72.45±0.268
CONCIL	73.52±0.160	73.27±0.143	73.51±0.225	72.67±0.158	73.33±0.121	71.94±0.210	73.33±0.293	73.21±0.180	71.65±0.173	72.19±0.098	72.91±0.186
CLG-CBM	73.25±0.133	72.98±0.170	72.90±0.150	73.56±0.216	73.51±0.184	72.53±0.172	73.40±0.280	72.80±0.168	71.61±0.132	72.22±0.223	72.87±0.201
CI-CBM	73.04±0.188	72.70±0.132	73.03±0.217	73.15±0.068	73.25±0.217	73.05±0.194	73.70±0.105	72.92±0.219	72.52±0.234	72.66±0.165	72.93±0.163
CABM	73.35±0.251	72.78±0.263	72.86±0.199	73.09±0.045	73.24±0.480	73.01±0.164	73.55±0.015	72.86±0.316	72.63±0.049	73.00±0.068	**73.01±0.150**
Concept accuracy (↑, %)											
Original	89.81	88.47	88.80	88.37	89.48	87.81	88.61	88.70	88.56	90.05	88.77
Retrain	88.62±0.434	87.56±0.211	88.78±0.380	87.40±1.374	88.31±0.090	86.62±0.538	87.53±0.183	87.77±0.230	87.55±0.354	88.97±0.070	87.79±0.367
FastIF	86.23±0.409	84.19±0.234	87.39±0.183	88.09±0.182	87.86±0.151	83.28±0.147	86.16±0.055	86.29±1.392	87.19±0.068	86.46±0.159	86.59±0.293
Partial-FT	86.32±0.251	87.06±0.189	87.35±0.180	86.95±0.208	87.06±0.251	85.45±0.281	85.20±0.343	87.23±0.275	87.04±0.314	87.65±0.361	86.97±0.273
DER++	85.93±0.475	87.65±0.181	87.45±0.211	87.87±0.238	88.32±0.189	87.32±0.241	84.31±0.111	88.94±0.106	87.58±0.358	88.94±0.178	87.50±0.225
CONCIL	88.08±0.170	88.65±0.167	87.85±0.504	87.97±0.178	87.68±0.114	86.37±0.145	86.78±0.107	87.68±0.130	88.46±0.126	88.92±0.120	88.10±0.208
CLG-CBM	89.98±0.288	89.61±0.141	89.82±0.234	89.97±0.209	89.68±0.298	88.37±0.168	88.78±0.220	87.68±0.218	89.46±0.240	88.92±0.294	88.94±0.232
CI-CBM	87.50±0.280	88.80±0.257	88.30±0.239	87.80±0.248	88.36±0.211	87.40±0.195	86.99±0.218	88.93±0.189	88.57±0.315	89.21±0.222	88.20±0.241
CABM	88.68±0.453	89.42±0.199	89.71±0.171	89.31±0.260	89.42±0.134	87.81±0.062	87.56±0.048	89.59±0.059	89.40±0.244	89.92±0.140	**89.33±0.271**
Runtime (↓, seconds)											
Retrain	402±19	229±22	362±30	443±59	319±27	452±40	331±55	279±14	347±41	249±23	346.1±37.5
FastIF	61.77±3.188	58.46±1.621	60.13±2.674	61.56±2.731	68.26±2.088	64.11±5.122	73.40±1.982	58.52±4.623	66.63±2.287	63.72±1.151	65.4±3.107
Partial-FT	121.03±8.635	117.72±12.596	119.39±5.326	120.82±8.262	127.52±7.080	123.37±6.907	132.66±4.265	117.78±7.775	125.89±7.117	122.89±7.773	124.7±8.12
DER++	281.34±42.60	335.09±44.00	323.47±43.40	241.74±47.60	232.08±53.40	387.22±43.80	408.21±42.30	313.03±43.20	289.09±42.30	238.49±43.30	315.2±39.5
CONCIL	194.94±26.92	215.19±26.54	231.78±12.72	221.72±26.93	221.92±12.37	233.66±15.94	263.74±20.59	269.69±17.07	272.38±11.35	258.94±9.34	242.8±18.6
CLG-CBM	257.23±37.10	185.21±26.61	240.94±22.80	173.11±27.13	187.25±22.52	162.18±26.11	180.34±30.73	243.45±27.24	254.57±21.54	206.97±29.50	211.6±23.7
CI-CBM	215.33±45.70	249.13±35.52	224.99±31.40	233.31±35.92	228.40±21.41	240.01±34.99	242.02±39.65	247.44±26.14	257.32±30.46	234.57±28.40	236.4±32.8
CABM	32.89±1.777	28.74±1.555	34.65±1.250	35.43±1.986	32.08±2.954	34.97±1.396	31.36±0.738	30.11±1.320	33.92±0.994	29.85±2.499	**32.15±1.890**

Table 15: Derm7pt: detailed leave-one-out concept addition performance over 5 random seeds. We report label accuracy, concept accuracy, and runtime for each newly added concept as mean ± standard deviation.

	PigNet	Streaks	Pigment	Regr.	Dots	Veil	Vasc.	Avg.
Label accuracy (↑, %)								
Original	81.99	78.88	80.75	79.75	79.50	80.76	80.03	80.24
Retrain	80.10±1.430	79.52±1.811	81.29±1.607	80.40±1.288	80.18±0.833	80.72±0.810	79.37±1.483	80.23±1.323
FastIF	79.22±0.652	76.40±1.450	78.20±1.313	78.01±1.139	77.02±1.181	78.32±0.791	79.10±1.046	78.04±1.082
Partial-FT	79.87±0.091	75.92±0.135	79.43±0.089	77.88±0.072	78.17±0.085	79.20±0.132	78.93±0.107	78.49±0.101
DER++	80.74±0.090	78.46±0.176	80.45±0.098	79.63±0.096	78.10±0.068	79.11±0.114	78.68±0.093	79.31±0.105
CONCIL	81.18±0.079	79.00±0.069	81.03±0.075	79.86±0.079	79.23±0.151	80.21±0.110	80.05±0.088	80.08±0.093
CLG-CBM	81.45±0.104	79.42±0.052	81.38±0.099	80.68±0.075	79.98±0.109	80.51±0.091	80.08±0.079	80.50±0.087
CI-CBM	81.72±0.103	79.16±0.097	81.23±0.136	79.94±0.025	79.68±0.099	79.89±0.099	79.92±0.106	80.22±0.095
CABM	81.88±0.960	79.61±1.273	81.46±0.709	80.38±0.924	79.71±0.982	80.42±0.627	80.37±0.595	**80.55±0.868**
Concept accuracy (↑, %)								
Original	75.78	74.02	75.36	74.12	77.43	72.32	72.11	74.45
Retrain	75.69±1.062	74.61±1.206	75.92±0.785	74.73±1.412	78.01±0.757	71.98±1.515	71.95±0.733	74.70±1.067
FastIF	73.36±0.511	72.21±0.838	73.51±0.607	72.31±0.976	75.05±0.502	70.61±1.156	69.82±0.699	72.41±0.756
Partial-FT	74.58±0.082	71.81±0.076	74.12±0.115	71.92±0.077	76.29±0.098	71.78±0.120	70.17±0.108	72.95±0.097
DER++	75.03±0.127	73.32±0.079	74.44±0.142	73.26±0.072	76.97±0.062	71.15±0.077	70.89±0.085	73.58±0.092
CONCIL	75.48±0.072	73.83±0.103	75.30±0.071	73.77±0.098	77.22±0.103	71.96±0.079	71.56±0.056	74.16±0.083
CLG-CBM	76.02±0.066	74.31±0.053	75.72±0.108	74.48±0.069	77.70±0.115	72.79±0.081	72.44±0.054	74.78±0.078
CI-CBM	72.97±0.063	74.14±0.088	75.60±0.105	74.20±0.107	75.71±0.027	75.22±0.071	72.82±0.079	74.38±0.077
CABM	76.12±0.815	74.56±0.894	75.98±0.655	74.68±0.548	78.12±0.582	72.77±0.809	72.51±0.374	**74.96±0.668**
Runtime (↓, seconds)								
Retrain	403±54	295±32	231±23	230±61	520±40	343±41	379±72	343.0±46.1
FastIF	59.28±3.121	57.34±2.424	58.63±5.433	58.17±4.615	60.34±1.711	59.84±3.099	60.71±4.207	59.2±3.516
Partial-FT	115.4±10.26	109.8±11.86	112.5±11.75	114.2±9.66	121.6±9.85	110.3±12.59	113.9±7.88	114.0±10.55
DER++	330.5±17.3	248.2±26.1	200.2±24.0	187.7±26.7	442.4±33.2	288.5±28.9	328.3±31.4	289.4±26.8
CONCIL	276.9±12.4	204.4±12.7	155.9±15.0	240.1±20.2	229.2±24.4	245.1±22.8	235.3±24.8	226.7±18.9
CLG-CBM	277.4±19.2	194.6±17.5	283.3±15.1	307.0±13.2	283.9±16.4	237.0±20.8	187.8±15.4	253.0±16.8
CI-CBM	212.9±15.5	191.5±14.5	162.2±11.2	147.4±13.1	287.2±18.6	181.7±16.2	206.6±17.3	198.5±15.2
CABM	**27.70±2.114**	**26.94±1.557**	**27.93±2.916**	**27.51±1.094**	**28.43±3.296**	**27.13±1.941**	**28.07±1.928**	**27.7±2.121**

Table 16: CUB-200: detailed batch concept addition performance. We compare all baselines on adding semantic groups: **Wing Color**, **Bill Shape**, and **Tail Pattern**.

Method	Label Acc. (↑, %)				Concept Acc. (↑, %)				Runtime (↓, s)			
	Wing	Bill	Tail	Avg.	Wing	Bill	Tail	Avg.	Wing	Bill	Tail	Avg.
Original	78.72	79.65	79.14	79.17	96.48	96.63	96.54	96.55	–	–	–	–
Retrain	80.51	**81.51**	**82.04**	**81.35**	**96.70**	**96.74**	**96.72**	**96.72**	8795	14517	12481	11931
FastIF	79.15	80.10	80.05	79.77	95.98	96.10	96.07	96.05	325	415	390	377
Partial-FT	79.85	79.90	81.10	80.28	96.25	96.38	96.33	96.32	650	780	720	717
DER++	80.20	80.91	81.47	80.86	96.37	96.49	96.46	96.44	2100	2240	2290	2210
CONCIL	80.40	81.18	81.78	81.12	96.55	96.60	96.59	96.58	1650	1810	1820	1760
CLG-CBM	80.47	81.25	81.76	81.16	96.60	96.64	96.65	96.63	2260	2440	2486	2395
CI-CBM	80.52	81.31	81.83	81.22	96.62	96.68	96.68	96.66	2520	2700	2739	2653
CABM	**80.65**	81.45	81.92	81.34	96.67	96.72	96.71	96.70	**186**	**205**	**192**	**194**

Variation across settings and seeds. Tables 14 and 15 report per-setting means and standard deviations over five random seeds.

CUB semantic group addition. Table 16 reports detailed CUB results for adding semantic concept groups such as *Wing Color*, *Bill Shape*, and *Tail Pattern*. This evaluates whether CABM can handle batch concept expansion rather than only single-concept insertion.

AwA2 multi-concept addition. Table 17 reports AwA2 results when 5, 10, or 15 concepts are removed from the base model and added back. This setting evaluates larger-scale concept expansion and shows how update cost changes as the number of newly added concepts increases.

Table 17: AwA2 concept addition results. We remove 5, 10, or 15 concepts from the base model and evaluate adding them back.

Added Concepts	Method	Label Acc. (↑, %)	Concept Acc. (↑, %)	Runtime (↓, s)
5	Original Base Model	90.07	97.11	–
	Retrain	**90.20**	**97.73**	6291.0
	FastIF	90.07	97.61	712.0
	CONCIL	90.13	97.52	4977.0
	DER++	90.11	97.49	4870.0
	Partial-FT	90.12	97.44	3751.0
	CLG-CBM	90.12	97.49	3500.0
	CI-CBM	90.15	97.63	2980.0
	CABM	90.15	97.52	**331.6**
10	Original Base Model	89.93	97.72	–
	Retrain	**90.18**	97.40	7218.0
	FastIF	89.98	97.13	929.1
	CONCIL	90.14	97.54	6488.0
	DER++	90.09	97.32	6414.0
	Partial-FT	90.08	97.33	4008.0
	CLG-CBM	90.11	97.46	3815.0
	CI-CBM	90.14	**97.62**	3210.0
	CABM	90.13	97.50	**402.3**
15	Original Base Model	89.35	97.68	–
	Retrain	**90.21**	**97.71**	8080.0
	FastIF	89.62	96.90	777.5
	CONCIL	90.11	97.44	5211.0
	DER++	89.70	97.03	6034.0
	Partial-FT	89.67	89.99	4790.0
	CLG-CBM	90.10	97.46	4130.0
	CI-CBM	90.13	97.61	3473.0
	CABM	90.16	97.58	**452.0**

E.7 Stability and Ablation Details

This section provides detailed interpretations for the stability and ablation experiments summarized in the main paper.

Robustness to label noise. We inject synthetic noise into the labels of the newly added concept, increasing the corruption ratio from 2% to 20%. Retraining is sensitive to noisy new-concept labels because corrupted supervision can influence the optimization trajectory of the expanded model. Partial-FT also exhibits unstable behavior because it still relies on iterative gradient updates. In contrast, CABM localizes its update and constrains its magnitude through a curvature-aware trust-region step, making noisy supervision less likely to disrupt the pretrained concept space.

Expert intervention. Following standard CBM intervention protocols, we replace 1–10 predicted concepts with ground-truth values at test time. CABM consistently benefits from these corrections and matches or slightly exceeds the gains of retraining. This indicates that the edited concept bottleneck remains actionable: expert corrections to concepts still translate into downstream label improvements. We further report targeted intervention on newly added concepts in Table 18.

Sequential concept addition. We incrementally add multiple new concepts to evaluate whether approximation error accumulates over repeated edits. In the SkinCon setting of Fig. 4(b)–(c), CABM maintains stable accuracy growth with approximately constant runtime per edit. In contrast, gradient-based baselines require iterative optimization at each addition step. Additional sequential results on CUB are reported in Table 21.

Downstream prediction drift. The KL divergence in Fig. 4(d) measures changes in downstream model outputs. It does not measure changes in old concept scores, which CABM preserves exactly. A low label-output drift alone does not establish that a method learns new concepts effectively; we therefore report new-concept accuracy and intervention-effect drift separately.

Design components. The component comparisons examine concept-predictor initialization, curvature correction, and label-head adaptation. In the initialization ablation (the first four rows of Fig. 3(a)), concept accuracy evaluates the newly appended concept only; the 21 frozen old concepts are excluded. The remaining rows report all-concept accuracy. The additional diagnostics report clipping frequency and severity, the prior-only label result, and diagonal versus EK-FAC curvature. Their empirical differences do not imply that EK-FAC is necessary for every concept edit.

Concept-only editing with a random head extension. The "Only edit g" row in Fig. 3(a) uses a nonzero, unadapted extension of the default linear-softmax label head. We freeze the encoder and the pre-existing concept predictor, and apply the CABM update to the appended concept-predictor block. The label head is expanded to accept the new concepts, while its pre-existing coefficients V_C and bias b are copied from the base model without reinitialization. Only the appended coefficients are randomly initialized:

$$\left(V_A^{\text{init}}\right)_{aj} \overset{\text{i.i.d.}}{\sim} \mathcal{U}\left(-\frac{1}{\sqrt{d_{\text{in}}}}, \frac{1}{\sqrt{d_{\text{in}}}}\right), \qquad d_{\text{in}} = d_C + d_A. \tag{28}$$

Here, d_{in} is the full input dimension of the expanded label head; for scalar concepts, $d_{\text{in}} = k + t$. The expanded label head is then held fixed, with no subsequent CABM label-head update. Given the unchanged old concepts $c(x)$ and the updated new concepts $z^\dagger(x)$, its prediction is

$$p_{\text{rand}}(y \mid x) = \text{softmax}\left(V_C^\top c(x) + \left(V_A^{\text{init}}\right)^\top z^\dagger(x) + b\right). \tag{29}$$

Thus, "only" refers to the module receiving the CABM parameter update, rather than to whether the label head is structurally expanded. This control evaluates concept editing with unadapted nonzero connections. Since its head initialization differs from full CABM, the comparison does not isolate the label-head update under a shared initialization.

Zero-initialized no-head-update control. For the matched-initialization control, we retain the zero initialization $V_A^{(0)} = 0$ used by full CABM and omit only the subsequent label-head update. The appended concept predictor still receives the CABM update, while the old concept pathway, V_C, and b remain fixed. Consequently,

$$p_{\text{zero}}(y \mid x) = \text{softmax}\left(V_C^\top c(x) + 0^\top z^\dagger(x) + b\right) = p_{\text{base}}(y \mid x). \tag{30}$$

This control preserves the base model's label predictions by construction. It is distinct from the randomly initialized extension reported as "Only edit g" in Fig. 3(a).

E.8 Additional Diagnostic Results

This section reports targeted intervention on newly added concepts, irrelevant concept insertion, and ridge-head validation.

Targeted intervention on newly added concepts. We evaluate whether newly added concepts remain actionable after editing. For each of the 22 SkinCon concepts, we train a base model with that concept held out, add it back, and then correct the predicted value of the target concept to its ground truth at test time. We compare intervening on the newly added concept against intervening on a random old concept.

Robustness to irrelevant concepts. To test robustness to irrelevant concept insertion, we add a pure random binary concept to SkinCon. CABM assigns a near-zero weight to this irrelevant concept in the label predictor, and the overall disease accuracy remains stable.

Ridge vs. softmax label heads. The main experiments use a softmax label head. To validate the ridge-head derivation, we replace the softmax predictor with a ridge regression head on SkinCon. Disease labels are converted to one-hot targets and optimized with MSE.

Table 18: Targeted intervention on SkinCon. Correcting the newly added concept in CABM yields a larger downstream gain than correcting a random old concept and performs comparably to retraining.

Method	**Target Concept for Intervention**	**Avg. Label Acc. (↑, %)**
Baseline	None	72.97
CABM edited	Random old concept	73.06
Retrain	Newly added concept	73.07
CABM edited	Newly added concept	**73.11**

Table 19: Irrelevant concept insertion on SkinCon. CABM suppresses the effect of a pure random noise concept.

Setup	**Added Concept Type**	**Label Acc. (↑, %)**	**Assigned Label-Head Weight**
Original CBM	None	72.97	–
CABM	Pure random noise	72.94	0.002

Table 20: Label-head ablation on SkinCon. Ridge regression is faster but slightly less accurate than the default softmax head.

Label Predictor Head	**Label Acc. (↑, %)**	**Runtime (↓, s)**
Ridge regression	71.85	**26.77**
Softmax	**73.01**	32.15

Table 21: Sequential concept addition on CUB-200. We sequentially add five semantic concept groups and report label accuracy and stage-level runtime. CABM achieves competitive accuracy with substantially lower stage-level update time.

Sequential Addition Step	**Label Accuracy (↑, %)**		**Stage Runtime (↓, s)**		**Speedup**
	Retrain	CABM	Retrain	CABM	
Step 1: + *Bill Shape*	**73.27**	72.81	11152	**145**	76.9×
Step 2: + *Wing Color*	**76.60**	76.18	12307	**192**	64.1×
Step 3: + *Tail Pattern*	**79.12**	78.45	13058	**297**	44.0×
Step 4: + *Upperparts Color*	**80.95**	80.53	12140	**338**	35.9×
Step 5: + *Underparts Color*	**82.10**	82.04	14144	**393**	36.0×

E.9 Sequential Concept Addition

We conduct a sequential editing experiment on CUB to simulate a realistic user workflow. In this setting, the model receives semantic concept groups one by one. We sequentially append five groups in the following order: *Bill Shape*, *Wing Color*, *Tail Pattern*, *Upperparts Color*, and *Underparts Color*.

We evaluate CABM against retraining at every addition step. Table 21 reports label accuracy and stage-level runtime. Retraining achieves marginally higher accuracy because it globally re-optimizes the expanded model. However, this comes at a prohibitive computational cost. CABM reaches 82.04% after the final step, close to 82.10% for retraining, while reducing the final stage-level update time from 14144 seconds to 393 seconds. This confirms that CABM enables continuous concept-space maintenance without repeated full retraining.