Dynamic-V2C: Editable and Continual Vision-to-Concept Bottleneck Models via Influence Functions

Anonymous authors
Paper under double-blind review

Abstract

Concept Bottleneck Models (CBMs) offer intrinsic interpretability by grounding predictions in human-understandable concepts. However, contemporary CBMs face three key limitations: reliance on ungrounded text-driven concepts, vulnerability to spurious correlations that require costly retraining to rectify, and severe catastrophic forgetting during continual learning. We introduce Dynamic-V2C, a unified framework that constructs a Vision-to-Concept (V2C) tokenizer and incorporates an Influence Function (IF) engine for rapid, retraining-free model editing. By isolating approximate inverse-curvature updates to the low-dimensional bottleneck and leveraging damping, our method performs rapid head-level data removal and spurious concept debugging in seconds. Furthermore, Dynamic-V2C extends to continual learning through influence-triggered vocabulary expansion and damped protection of historical directions. Experiments across six visual benchmarks demonstrate competitive predictive performance while reducing editing time by over two orders of magnitude (over 131× on CUB) relative to full retraining. Dynamic-V2C corrects worst-group spurious correlations in under two seconds, limits catastrophic forgetting to -3.1% Backward Transfer, and reduces the output-level privacy signal of removed data while leaving frozen-backbone leakage explicitly out of scope.

1 Introduction

Deep visual models achieve strong predictive performance, but their limited transparency continues to hinder deployment in high-stakes domains such as medical diagnosis, autonomous driving, and financial forecasting. To bridge the gap between predictive performance and human-level interpretability, Concept Bottleneck Models (CBMs) Koh et al. (2020) require the network to first predict intermediate, human-understandable concepts (e.g., "red wings", "calcification") before mapping these concepts to a final target label. Recently, Label-Free CBMs Oikarinen et al. (2023) and Post-hoc CBMs Yuksekgonul et al. (2023) have reduced the burden of manual concept annotation by leveraging the open-vocabulary knowledge of Large Language Models (LLMs) and Vision-Language foundation models such as CLIP.

While these advancements alleviate the annotation burden, the real-world scalability of modern CBMs is still hindered by three intertwined challenges, as illustrated in Figure 1(A). First, LLM-generated concepts are frequently verbose, abstract, and lack strict visual grounding (e.g., "the historical significance of the bird"), effectively introducing non-visual noise into the bottleneck He et al. (2025). Second, CBMs can exhibit concept leakage Havasi et al. (2022) and remain vulnerable to spurious correlations and data poisoning. For example, a model might spuriously associate a "water background" with the concept of "waterbirds". When such biases or poisoned training data are discovered post-deployment, a dependable correction often falls back to expensive retraining Bourtoule et al. (2021). Third, real-world visual systems are rarely static; they must learn continuously. Existing CBMs suffer from catastrophic forgetting when exposed to sequential tasks, as modifying the linear projection layer overwrites previously acquired conceptual knowledge Yu et al. (2025).

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.

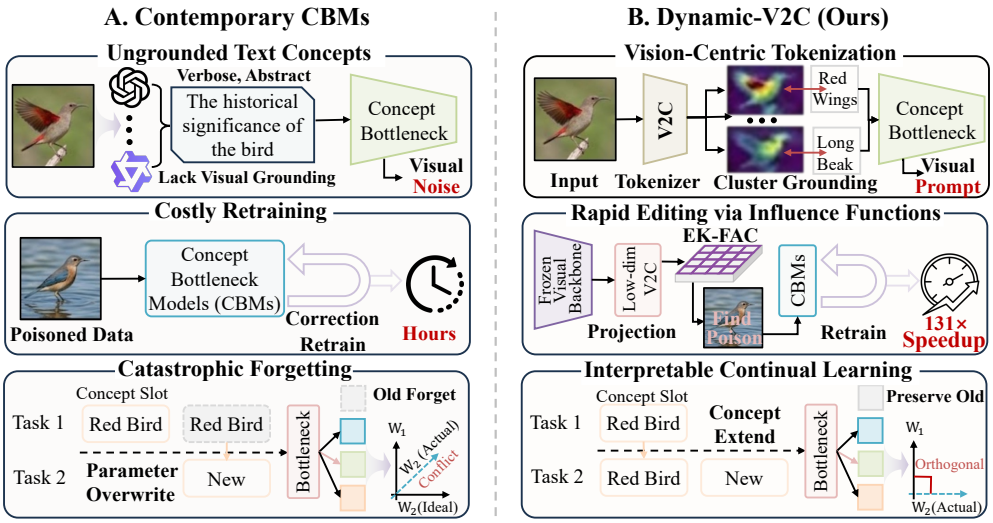


Figure 1: Motivation of Dynamic-V2C. (A) Contemporary CBMs suffer from three intertwined limitations: (i) ungrounded, verbose text concepts that inject visual noise into the bottleneck; (ii) costly full retraining when poisoned data or spurious correlations are discovered; and (iii) catastrophic forgetting during sequential tasks due to parameter overwriting conflicts between W_1 and W_2. (B) Dynamic-V2C addresses each challenge with vision-centric tokenization via a frozen backbone and cluster-grounded V2C projection, rapid IF-based editing accelerated by EK-FAC (over 131× on CUB), and continual vocabulary expansion with damped protection of historically active directions.

To systematically address these challenges, we introduce Dynamic-V2C, an editable and continual Vision-to-Concept Bottleneck Model (Figure 1(B)). The framework has three components:

1. Vision-Centric Tokenization: Inspired by V2C-CBM's vision-oriented objective He et al. (2025), but instantiated differently, our tokenizer clusters dense CLIP patch embeddings into visual prototypes. Post-hoc text labels are used only for display, avoiding arbitrary LLM-generated training concepts.

2. Rapid Editing via Influence Functions (IFs): Building on influence-based CBM editing Koh & Liang (2017); Hu et al. (2025), we couple edits to an evolving visual-prototype bottleneck: after updating the V2C projection, the label predictor is corrected for the induced shift $c \to \bar{c}$. A direct dense inverse over d parameters costs $\mathcal{O}(d^3)$ time and $\mathcal{O}(d^2)$ memory and is fragile in deep non-convex networks Basu et al. (2021). We instead freeze the backbone and apply damped EK-FAC George et al. (2018) only to the low-dimensional projection, enabling rapid output-level editing without full retraining.

3. Interpretable Continual Learning: We extend the same influence signal to sequential tasks. Dynamic-V2C initializes visual prototypes for poorly represented features and uses a damped form of orthogonal gradient protection Farajtabar et al. (2020) to reduce interference with historical concepts, complementing CBM-specific continual learning Yu et al. (2025).

We conduct extensive evaluations on established benchmarks including CUB Wah et al. (2011), CelebA Liu et al. (2015), and the Osteoarthritis Initiative (OAI) Nevitt et al. (2006); Koh et al. (2020). Our results show that Dynamic-V2C approaches the predictive utility of full retraining while accelerating model editing from hours to seconds (over $10^2\times$ on CUB). Furthermore, the Robust Membership Inference Attack (RMIA) Zarifzadeh et al. (2024) shows that editing reduces output-score membership leakage, whereas a separate feature-space attack remains unchanged. We therefore describe the procedure as output/head-level editing, not backbone or certified machine unlearning.

2 Related Work

Concept Bottleneck Models. CBMs route predictions through an explicit concept layer Koh et al. (2020). Label-free and post-hoc variants reduce annotation cost with multimodal representations Oikarinen et al. (2023); Yuksekgonul et al. (2023), while V2C-CBM learns a vision-to-concept mapping over a filtered textual vocabulary He et al. (2025). Editable CBM derives influence-based concept–label, concept, and data edits Hu et al. (2025), and Controllable CBM extends this editing view to data addition and removal Lin et al. (2026). Dynamic-V2C builds on these lines with clustered patch prototypes, concept-shift-aware local editing, and continual vocabulary expansion. Appendix A provides an extended discussion of interactive, probabilistic, counterfactual, and intervention-oriented CBMs.

Machine Unlearning and Influence Functions. Influence functions trace predictions to training samples through inverse-Hessian-vector products Koh & Liang (2017), but their deep-network approximations can be fragile Basu et al. (2021). Machine-unlearning methods range from sharded retraining Bourtoule et al. (2021) to Hessian and optimization-based approximations Mehta et al. (2022); Thudi et al. (2022). We restrict curvature to the shallow bottleneck and use Kronecker-factored approximations Grosse & Martens (2016); George et al. (2018); this provides fast local editing rather than certified deletion of frozen-backbone information. The full unlearning taxonomy and historical IF discussion are retained in Appendix A.

Continual Learning. Interpretable continual learners preserve concept structure across tasks Yang et al. (2024); Yu et al. (2025); CI-CBM additionally studies class-incremental concept regularization and pseudo-concept generation Javadi et al. (2026). Dynamic-V2C operates on a visually grounded vocabulary, adding capacity when a new task strongly perturbs historical weights and protecting previously active directions during the update. We evaluate a task-aware protocol and leave class-incremental evaluation to future work; extended discussion appears in Appendix A.

3 Preliminaries

We consider a classification task with a concept set denoted as $\mathcal{C} = \{p_1, \ldots, p_k\}$, where each p_j is a visual attribute or prototype. The training dataset is represented as $\mathcal{D} = \{z_i\}_{i=1}^n$, where $z_i = (x_i, y_i, c_i)$. Here, $x_i \in \mathbb{R}^{d_i}$ represents the input image, $y_i \in \{0,1\}^{d_o}$ denotes the one-hot target label, and $c_i = (c_i^1, \ldots, c_i^k) \in \mathbb{R}^k$ represents a concept target or activation vector. Conventional CBMs obtain c_i from manual labels or multimodal zero-shot predictions Oikarinen et al. (2023); Dynamic-V2C instead sets $k = K$ and obtains c_i from fixed patch-to-prototype assignments, without using benchmark concept annotations for training.

CBMs factorize the prediction into two distinct mapping functions: the concept predictor $g : \mathbb{R}^{d_i} \to \mathbb{R}^k$, and the label predictor $f : \mathbb{R}^k \to \mathbb{R}^{d_o}$. The label predictor f is strictly constrained to be a linear transformation to preserve the exact weight contributions of each concept. The empirical losses for the concept predictor $\hat{g}$ and label predictor $\hat{f}$ are minimized as follows:

$$\hat{g} = \arg\min_g \sum_{i=1}^{n} \sum_{j=1}^{k} L_C^j(g(x_i), c_i^j) \tag{1}$$

$$\hat{f} = \arg\min_f \sum_{i=1}^{n} L_Y\big(f(\hat{g}(x_i)), y_i\big) \tag{2}$$

where L_C is the concept/prototype prediction loss and L_Y is the label cross-entropy loss.

Influence Functions. The influence function mathematically measures the dependence of a trained estimator $\hat{\theta} = \arg\min_\theta \sum_{i=1}^n \ell(z_i; \theta)$ on an individual data point z_m. By introducing a perturbation where z_m is upweighted by an infinitesimally small ϵ, the newly optimal parameters $\hat{\theta}_{\epsilon, z_m}$ can be approximated via a first-order Taylor expansion:

$$\mathcal{I}_{\hat{\theta}}(z_m) \triangleq \left. \frac{\mathrm{d}\hat{\theta}_{\epsilon, z_m}}{\mathrm{d}\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \cdot \nabla_\theta \ell(z_m; \hat{\theta}) \tag{3}$$

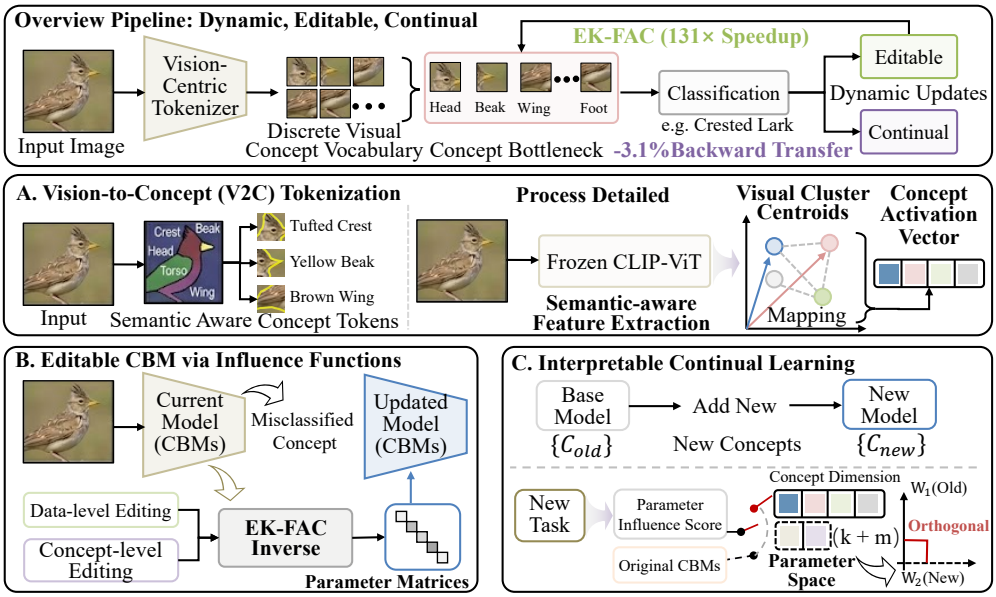


Figure 2: The overall framework of Dynamic-V2C. (A) Vision-to-Concept (V2C) Tokenization: Input images are mapped to a visually grounded prototype bottleneck. (B) EK-FAC Influence Engine: Computes approximate inverse-curvature updates exclusively on the bottleneck, utilizing damping for stability. (C) Interpretable Continual Learning: Dynamically expands the concept vocabulary for novel tasks while protecting historical knowledge via influence-guided gradient projection.

where $H_{\hat{\theta}} = \nabla^2_\theta \sum_{i=1}^{n} \ell(z_i; \hat{\theta})$ is the Hessian of the summed loss. Because a neural-network Hessian need not be positive definite, our implementation substitutes the damped positive-definite curvature approximation described in Section 4. To approximate removal of z_m, we set its weight to zero via $\epsilon = -1$ in this upweighting convention, yielding the local change $\Delta\theta \approx +H_{\hat{\theta}}^{-1} \nabla_\theta \ell(z_m; \hat{\theta})$ Koh & Liang (2017).

4 Methodology: Dynamic-V2C

In this section, we formulate the Dynamic-V2C framework, as illustrated in Figure 2. We first define the Vision-to-Concept (V2C) Tokenizer. We then adapt influence-based CBM edits Hu et al. (2025) to a visual-prototype bottleneck whose concept activations can shift after an edit, and detail EK-FAC acceleration and the continual-learning extension. A formal step-by-step summary is provided in the appendix.

4.1 Vision-to-Concept (V2C) Tokenization

As depicted in Figure 2(A), given an image $x \in \mathcal{X}$, we extract dense visual features using a frozen CLIP image encoder. Instead of querying an LLM for arbitrary text concepts, we construct a discrete visual-prototype vocabulary $\mathcal{P} = \{p_1, \ldots, p_K\}$ by projecting image patches directly into the CLIP visual embedding space and clustering them using a large auxiliary unlabeled dataset. Specifically, we extract dense patch embeddings from the penultimate layer of the frozen CLIP-ViT, apply L_2 normalization, and run k-means to establish K cluster centroids. The resulting patch-to-centroid assignments are pooled into the prototype target $c_i \in \mathbb{R}^K$ used in Equation 1; the trainable projection g_{proj} predicts the corresponding prototype-activation vector from frozen visual features. Each unit is therefore grounded in clustered patch evidence rather than generated text. For qualitative presentation, we query each visual centroid against the frozen CLIP text encoder using a dictionary of domain-specific attributes and assign the nearest-neighbor label (e.g., "yellow beak", "heavy makeup"). The model's routing and editing use the clusters, not these labels. As k-means-derived "visual words," they need not have the semantic purity or compositionality of manually verified concepts; the design instead avoids using LLM-generated text as training concepts.

4.2 Editable CBM via Influence Functions

As shown in Figure 2(B), we encounter two primary editing scenarios in the lifecycle of a deployed visual CBM: Data-Level Editing (removing poisoned/biased images at the bottleneck and head) and Concept-Level Editing (removing a spurious concept route). Hu et al. Hu et al. (2025) introduced influence-based data and concept edits for CBMs; our formulation specializes this idea to the patch-prototype tokenizer, couples the label-head correction to the induced concept shift, and supports the same vocabulary as it expands across tasks.

4.2.1 Data-Level Editable CBM.

Suppose we identify a subset of training images indexed by $G \subset [n]$ that contain spurious biases or privacy-sensitive data. Our objective is to estimate the retrained concept predictor $\hat{g}_{-z_G}$ and label predictor $\hat{f}_{-z_G}$ trained exclusively on $[n] \setminus G$, entirely bypassing the heavy computational cost of retraining.

Theorem 1. Let $\hat{\phi}$ denote the parameters of g_{proj}. Given a dataset $\mathcal{D}$ and a removal set G, the parameters obtained by retraining the concept projection without G can be locally approximated by a single damped Newton step:

$$\hat{\phi}_{-G} \approx \bar{\phi}_{-G} \triangleq \hat{\phi} + (H_{\hat{\phi}} + \delta I)^{-1} \sum_{r \in G} \sum_{j=1}^{K} \nabla_{\phi} L_C^j(g_{\hat{\phi}}(x_r), c_r^j) \tag{4}$$

where $H_{\hat{\phi}} = \nabla_{\phi}^2 \sum_{i,j} L_C^j(g_{\phi}(x_i), c_i^j)|_{\phi=\hat{\phi}}$ is computed strictly over the tokenizer projection parameters; the edited predictor is $\bar{g}_{-G} = g_{\bar{\phi}_{-G}}$.

Proof Sketch. We reformulate the empirical risk after removing G using a subtraction weight ϵ (complete removal is $\epsilon = 1$) and differentiate the optimality condition $\nabla_g \mathcal{L} = 0$ at $\epsilon = 0$. The resulting first-order inverse-curvature update is exact for strictly convex quadratics and locally approximates our cross-entropy bottleneck using Levenberg–Marquardt damping. Appendix E gives the full derivation.

Following the approximation of $\bar{g}_{-z_G}$, we must correspondingly update the linear label predictor $\hat{f}$. Because the inputs to $\hat{f}$ (the concept activations) have now shifted from $c = \hat{g}(x)$ to $\bar{c} = \bar{g}_{-z_G}(x)$, applying standard Influence Functions directly fails. We resolve this by splitting the approximation into two explicit sequential steps: 1) removing the direct influence of G on $\hat{f}$ assuming concepts remain fixed, 2) compensating for the downstream representational shift from $c \to \bar{c}$. The full derivation of this two-step update is detailed in the supplementary material.

4.2.2 Concept-Level Editable CBM.

If a specific concept p_r (where $r \in M \subset [k]$) is discovered to be a spurious shortcut (e.g., "water background"), we must completely excise its influence from the downstream decision-making process. Retraining the label predictor $\hat{f}$ from scratch on the remaining concepts requires altering its input dimension from k to $k - |M|$. To execute this edit without causing architectural dimension mismatch, we formulate concept deletion as a linearly constrained local optimization in which the weights for concept set M are forced to zero.

Theorem 2. The initial label predictor $\hat{f}$ can be edited to ignore concept set M while adjusting its remaining weights through a dimension-consistent local approximation:

$$f^+ = \hat{f} - (H_M + \delta I)^{-1} \nabla_f \sum_{i=1}^{n} L_Y\left(\hat{f}(P_M \hat{c}_i), y_i\right), \qquad \bar{f}_{\setminus M} = \Pi_M(f^+), \tag{5}$$

where $P_M \in \mathbb{R}^{k \times k}$ zeros the M-th concept dimensions, H_M is curvature of the masked objective, and Π_M exactly zeros the corresponding classifier columns. Masking keeps the update dimension-consistent, while the final projection prevents removed coordinates from re-entering at inference (Appendix F).

This allows Dynamic-V2C to safely sever spurious concept routes while adjusting the remaining network weights to compensate for the lost feature space.

4.3 Acceleration via EK-FAC and Complexity Analysis

Inverse-curvature products for the linear label predictor can be solved independently of the visual backbone, whereas applying dense curvature to all parameters of the deep concept predictor is intractable ($\mathcal{O}(d^3)$). We therefore strictly freeze the deep visual backbone during

editing and apply Eigenvalue-corrected Kronecker-Factored Approximate Curvature (EK-FAC) George et al. (2018) solely to the shallow projection layers of the V2C tokenizer head. The head update likewise uses damped inverse-curvature–vector products rather than materializing a dense full-backbone Hessian.

For the l-th projection layer, K-FAC first forms activation and pre-activation-gradient factors $\hat{\Omega}_{l-1}$ and $\hat{\Gamma}_l$, whose eigenvectors define the Kronecker basis $Q_l = Q_\Omega \otimes Q_\Gamma$. Let $v_i = \mathrm{vec}(\nabla_{W_l} \ell_i)$. Unlike ordinary K-FAC, EK-FAC replaces the Kronecker-product eigenvalues by the empirical second moment in this basis:

$$d_{\mathrm{EK},l} = \mathbb{E}_i\left[(Q_l^\top v_i) \odot (Q_l^\top v_i)\right], \qquad \widehat{H}_l^{-1} \approx Q_l(\mathrm{diag}(d_{\mathrm{EK},l}) + \delta I)^{-1} Q_l^\top, \tag{6}$$

where δ is Levenberg–Marquardt damping Martens (2020). Using $\Lambda_\Omega \otimes \Lambda_\Gamma$ directly would instead recover ordinary K-FAC. As with empirical-Fisher and factored-curvature methods, this approximation need not equal the true Hessian Kunstner et al. (2019); Eschenhagen et al. (2023); restricting it to the shallow projection and using damping improves numerical stability. In our experiments, edits take seconds to a few minutes depending on dataset and backbone.

Computational & Memory Complexity: Full retraining requires repeated passes over N examples for E epochs, while a dense curvature inverse over d parameters costs $\mathcal{O}(d^3)$ time and $\mathcal{O}(d^2)$ storage. EK-FAC stores layerwise activation/gradient factors and a diagonal correction rather than a dense backbone matrix. At the start of an editing request, we refresh the curvature factors and reuse them across all requested samples; the reported update time includes this refresh and the projection/head corrections. The gain therefore comes from avoiding backbone optimization and repeated full-dataset epochs, not from claiming that curvature estimation is independent of N.

4.4 Interpretable Continual Learning Framework

As illustrated in Figure 2(C), we extend the IF-driven editing mechanism to Continual Learning over sequential tasks $T_1, T_2, \ldots, T_N$. Catastrophic forgetting in standard CBMs occurs because the weights in f associated with early concepts are severely disrupted by the gradient updates of later tasks. In Dynamic-V2C, when Task T_t arrives, we define the parameter influence score as the norm of the expected parameter shift $\mathcal{I}(T_t) = \|H_{\hat{f}}^{-1} \nabla_f L_{T_t}\|_2$, measuring how much the new task disrupts the existing historical concept weights W_{t-1}. If this influence exceeds a threshold τ, forcing the new task into the old bottleneck may cause substantial interference, so the V2C tokenizer expands the concept dimension from k to $k + m$. High-quantization-error patch residuals from T_t are clustered into m added prototypes. Gradients for the historical k concepts are attenuated in directions with high historical activation covariance through a damped variant of Orthogonal Gradient Descent Farajtabar et al. (2020), while the new slots remain trainable (Appendix G.5). Thus influence triggers expansion, whereas covariance controls historical-direction protection.

5 Experiments

In this section, we rigorously evaluate the utility, efficiency, and interpretability of Dynamic-V2C against full retraining and strong baselines.

5.1 Experimental Setup

Datasets. We utilize six established datasets to evaluate different capabilities under the Label-Free (Concept-Unsupervised) paradigm, where the concept bottleneck is trained entirely without manual ground-truth concept annotations (detailed implementation and hyperparameters are provided in the supplementary material): (1) Datasets with Ground-Truth Concepts (used exclusively for evaluation, not for training the V2C bottleneck): Bird Identification (CUB) Wah et al. (2011), comprising 11,788 images across 200 bird species with 312 concepts (for evaluation); Large-scale CelebFaces Attributes (CelebA) Liu et al. (2015), containing over 200,000 celebrity images with 40 attributes; and the Osteoarthritis Initiative (OAI) Nevitt et al. (2006), a medical dataset with 36,369 X-rays and 10 clinical indicators. (2) Standard Classification Datasets (lacking concept annotations entirely): CIFAR-100 Krizhevsky (2009), Flowers-102 Nilsback & Zisserman (2008), and Food-101 Bossard et al. (2014). For continual learning, we utilize Split-CIFAR100.

Table 1: Standard classification accuracy (%). All rows are our reimplementations under matched splits, backbones, hardware, and seeds; citations identify the methods.

Method	CUB	CelebA	OAI	CIFAR-100	Flowers	Food-101
Standard Black-Box (ViT-B/16)	82.1 ± 0.3	95.6 ± 0.2	89.2 ± 0.2	86.4 ± 0.3	97.1 ± 0.1	88.5 ± 0.2
Label-Free CBM Oikarinen et al. (2023)	77.4 ± 0.5	94.2 ± 0.3	86.5 ± 0.4	81.3 ± 0.6	92.4 ± 0.5	84.2 ± 0.4
V2C-CBM He et al. (2025)	80.5 ± 0.4	94.5 ± 0.2	88.3 ± 0.3	84.1 ± 0.4	95.8 ± 0.2	86.7 ± 0.3
Editable CBM Hu et al. (2025)	79.1 ± 0.4	93.8 ± 0.3	87.1 ± 0.3	82.5 ± 0.5	93.5 ± 0.4	85.1 ± 0.4
Dynamic-V2C (Ours)	80.1 ± 0.3	94.0 ± 0.2	88.4 ± 0.2	84.5 ± 0.3	95.9 ± 0.2	86.5 ± 0.3

Table 2: Data-level editing utility and efficiency (3% removal). All methods are run in our shared setup and hardware; results are averaged over 3 seeds.

Method	CUB (Bird Identification)		OAI (Medical Imaging)	
	F1 Score (↑)	Runtime (s) (↓)	F1 Score (↑)	Runtime (s) (↓)
Original Model (No Unlearning)	0.8012 ± 0.003	-	0.8845 ± 0.002	-
Retrain (Ground Truth)	0.7971 ± 0.002	5133	0.8825 ± 0.003	17866
CBM-IF (Standard Hessian)	0.7610 ± 0.015	1240	0.8310 ± 0.012	4120
V2C-CBM He et al. (2025) (Retrain)	0.7985 ± 0.004	5050	0.8830 ± 0.002	17500
Editable CBM Hu et al. (2025)	0.7850 ± 0.005	68	0.8690 ± 0.004	210
CLG-CBM Yu et al. (2025) (Retrain)	0.7920 ± 0.004	5200	0.8810 ± 0.003	18000
Dynamic-V2C (Ours)	0.7963 ± 0.003	39	0.8808 ± 0.002	141

Baselines. We reimplement baseline rows with matched data splits, seed counts, and hardware; concept-based methods within each comparison use the same frozen backbone unless a table explicitly states otherwise. The ResNet-50 row is a black-box reference, and the Food-101 scale study explicitly uses ViT-L/14. Citations identify source methods rather than previously published values: Standard ResNet-50 / ViT-B/16: Standard black-box models without concept bottlenecks. Retrain (Ground Truth): Full retraining of the CBM bottleneck from scratch after removing data or concepts. To ensure strict fairness, this baseline uses the exact same frozen visual backbone as Dynamic-V2C, meaning the runtime difference purely isolates our IF-engine's efficiency. CBM-IF: The vanilla influence function applied to CBM without EK-FAC acceleration (computationally heavy). Label-Free CBM Oikarinen et al. (2023): The standard zero-shot multimodal CBM relying on LLM text generation. V2C-CBM He et al. (2025): A recent CBM that utilizes a Vision-to-Concept tokenizer without editing capabilities. Editable CBM Hu et al. (2025): A recent method for editing CBMs, primarily focused on non-continual settings. CLG-CBM Yu et al. (2025): A recent framework for language-guided continual learning in CBMs, adapted to our task-aware split.

Metrics. Performance: F1 Score, Accuracy. Efficiency: Runtime (s). Privacy: RMIA Zarifzadeh et al. (2024). Continual Learning: Final Average Accuracy (FAA), the mean final accuracy over all learned tasks, and Backward Transfer (BWT), the average final-minus-initial accuracy change on previous tasks.

5.2 Standard Classification Performance

Before evaluating editing capabilities, we first establish that Dynamic-V2C preserves competitive standard classification accuracy relative to black-box models and static CBM baselines. As shown in Table 1, Dynamic-V2C consistently outperforms the Label-Free CBM Oikarinen et al. (2023). Its central values remain within 0.5 accuracy points of the static V2C-CBM He et al. (2025) on all six datasets, indicating that the modifications for editing preserve comparable base utility under these benchmarks.

5.3 Editing Efficiency and Machine Unlearning

To evaluate Data-Level Editing (Machine Unlearning), we randomly select 3% of the training data points across the CUB and OAI datasets and issue an editing request to remove their contribution from the bottleneck and prediction head. We evaluate the resulting F1 score strictly on the unseen test set to ensure that generalization is preserved. As shown in Table 2, Dynamic-V2C achieves F1 scores close to full retraining on both datasets at substantially lower runtime. On CUB, full retraining takes 5,133s, whereas Dynamic-V2C updates the bottleneck and head in 39s—over 131× faster. In this matched implementation, it also

Table 3: Our CelebA Gender-prediction protocol (spurious attribute: Blond Hair). All baseline values and times are matched reimplementations; cited works define the methods. Results are averaged over 3 seeds.

Method	Avg Acc (%)	Worst-Group Acc (%)	Update Time
Standard ResNet-50	95.6 ± 0.2	39.4 ± 1.2	$\sim$ 6 Hours
Group DRO Sagawa et al. (2020)	93.1 ± 0.4	80.5 ± 0.8	$\sim$ 8 Hours
JTT Liu et al. (2021)	93.5 ± 0.3	78.2 ± 0.9	$\sim$ 12 Hours
DFR Kirichenko et al. (2023)	94.0 ± 0.2	81.0 ± 0.6	$\sim$ 1 Hour
Label-Free CBM Oikarinen et al. (2023)	94.2 ± 0.3	41.2 ± 1.0	$\sim$ 6 Hours
V2C-CBM He et al. (2025)	94.5 ± 0.2	43.1 ± 1.1	$\sim$ 6 Hours
Editable CBM Hu et al. (2025)	93.8 ± 0.4	76.5 ± 1.5	$\sim$ 3 Seconds
Dynamic-V2C (Ours)	94.0 ± 0.2	81.5 ± 0.7	$<$ 2 Seconds

outperforms Editable CBM Hu et al. (2025) in utility and speed; this comparison does not isolate prototype grounding as the sole cause. A ViT-L/14 Food-101 stress test retains accuracy within 0.6 points of retraining in 112s instead of 3.2h (Appendix Table 4). The first-order approximation uses complete subtraction at $\epsilon = 1$ under our convention and remains local. Figure 9 reports the trend through 10% removal; for larger structural shifts ($|G| > 10\%$), we conservatively recommend full retraining.

5.4 Output-Level Membership Leakage Evaluation

To test whether editing reduces the output-level membership signal, we perform the Robust Membership Inference Attack (RMIA) Zarifzadeh et al. (2024). We randomly select 200 CUB training samples for removal and average over 5 independent runs. Before editing, these members yield a mean RMIA score of 0.0635 ($\pm$0.0012); afterward it falls to 0.0521 ($\pm$0.0008), near the unseen non-member baseline of 0.0494, and output RMIA ROC-AUC drops from 0.89 to 0.51. A separate white-box boundary probe on frozen embeddings remains essentially unchanged: CUB 0.54 $\rightarrow$ 0.55 and OAI 0.56 $\rightarrow$ 0.56. Dynamic-V2C therefore reduces output/head leakage in this evaluation, but is neither certified deletion nor backbone erasure.

5.5 Spurious Correlation Debugging (CelebA)

CBMs are frequently plagued by spurious correlations. We use a CelebA Gender-prediction protocol in which "Blond Hair" is the spurious attribute (the reverse orientation of the frequently used Blond-Hair target protocol). The worst group is Blond Males; in our matched reimplementation, the Label-Free CBM Oikarinen et al. (2023) obtains 41.2% worst-group accuracy.

To rectify this via Dynamic-V2C without relying on dense group annotations across the entire training set (unlike Group DRO Sagawa et al. (2020)), we adapt standard debugging protocols Liu et al. (2021); Kirichenko et al. (2023) and use a small balanced validation set with group labels to identify worst-group errors. Concept–label influence then ranks the training images contributing to this bias, and a damped approximate update penalizes the corresponding routes without another training epoch. Table 3 and Figure 3 show a modest, statistically overlapping worst-group margin over our DFR reimplementation (81.5% vs. 81.0%); the operational gain is the $<$ 2-second correction versus roughly one hour, with the same 94.0% average accuracy.

5.6 Qualitative Editing Case Study (CUB)

Complementary to the aggregate CelebA experiment, Figure 3 examines a separate CUB bird-classification case with a "water background" shortcut. Within two seconds, that route is zeroed, attention shifts to intrinsic bird evidence, and confidence rises from 45% to 89%.

5.7 Interpretable Continual Learning (Split-CIFAR100)

We evaluate ten task-aware splits of CIFAR-100. Dynamic-V2C reaches $78.5 \pm 0.6\%$ Final Average Accuracy (FAA) and $-3.1 \pm 0.4\%$ Backward Transfer (BWT), compared with $72.8 \pm 0.9\%$ and $-8.2 \pm 0.8\%$ for CLG-CBM Yu et al. (2025); its vocabulary grows only from 1,000

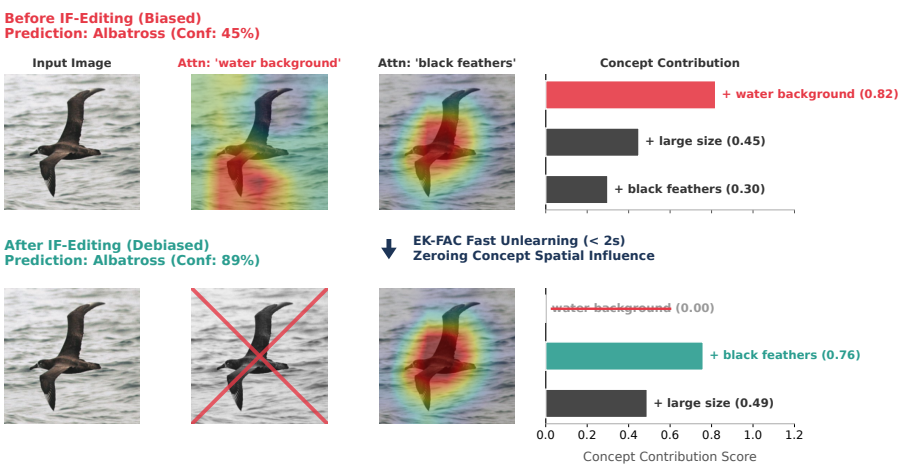


Figure 3: Case study: Dynamic-V2C removes a spurious visual route. Before editing (top), "water background" dominates both spatial attention and concept contribution. After the $<$ 2-second IF edit (bottom), that route is zeroed, attention shifts to intrinsic bird evidence, and confidence increases from 45% to 89%.

to about 1,350 concepts. Under a mixed-granularity CUB/CIFAR stream, fixed/adaptive expansion thresholds obtain FAA 75.1/76.4, BWT $-5.4/-4.6$, and vocabulary growth $1.39\times/1.32\times$. The complete baseline table, per-task curve, setup, and threshold analysis are in Appendix D.2.

5.8 Ablation and Qualitative Analysis

Vocabulary expansion alone reaches 58.4% FAA and IF-OGP alone reaches 65.2%; together they reach 78.5% FAA and -3.1% BWT. EK-FAC reduces the corresponding CUB edit from 1,240s to 39s. Appendix Table 8 gives the complete component ablation; Appendix D.3 provides additional prototype visualizations, and Appendix Figure 9 reports the removal-ratio analysis. Broader scope and limitations are discussed in Appendix B.

6 Conclusion

We present Dynamic-V2C, an Editable and Continual Concept Bottleneck Model grounded by a visual-prototype tokenizer tied to inspectable patch evidence. Its concept-shift-aware IF editing and EK-FAC local curvature updates enable fast data removal and spurious-route debugging without full retraining. Influence-triggered vocabulary expansion with damped historical-direction protection mitigates sequential catastrophic forgetting. Across six visual benchmarks, Dynamic-V2C provides a dynamically correctable architecture while retaining explicit limitations in prototype semantics, frozen-backbone privacy, and bias.

Reproducibility statement. The appendix reports implementation and hyperparameters, derivations, complete ablations, stress tests, privacy protocols, qualitative results, seed counts, and evaluation protocols.

Ethics statement. The study uses established benchmarks and no new human-subject data. CelebA gender prediction can encode harmful binary or demographic assumptions, and OAI results are not intended for clinical deployment. The reported edits are neither a fairness guarantee, certified deletion, legal compliance, nor frozen-backbone erasure.

AI use statement. Generative AI assisted prose, LaTeX/figure layout, reference formatting, and exposition/derivation critique; it generated no research data or experimental results. The authors verified all content and take full responsibility.

References

Aisha Aamir, Minija Tamosiunaite, and Florentin Wörgötter. Interpreting the decisions of cnns via influence functions. Frontiers in Computational Neuroscience, 17:1172883, 2023. doi: 10.3389/fncom.2023.1172883.

S Basu, P Pope, and S Feizi. Influence functions in deep learning are fragile. In International Conference on Learning Representations (ICLR), 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In European Conference on Computer Vision, pp. 446–461. Springer, 2014.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In 2021 IEEE Symposium on Security and Privacy (SP), pp. 141–159. IEEE, 2021.

Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In International Conference on Machine Learning, pp. 1092–1104. PMLR, 2021.

Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37(5), pp. 5948–5955, 2023.

R Dennis Cook. Detection of influential observation in linear regression. Technometrics, 42 (1):65–68, 2000.

R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. Technometrics, 22(4):495–508, 1980.

Gabriele Dominici, Pietro Barbiero, Francesco Giannini, Martin Gjoreski, Giuseppe Marra, and Marc Langheinrich. Counterfactual concept bottleneck models. In International Conference on Learning Representations, 2025.

Runa Eschenhagen, Alexander Immer, Richard E. Turner, Frank Schneider, and Philipp Hennig. Kronecker-factored approximate curvature for modern neural network architectures. In Advances in Neural Information Processing Systems, volume 36, 2023.

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

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, volume 108 of Proceedings of Machine Learning Research, pp. 3762–3773. PMLR, 2020. URL https://proceedings.mlr.press/v108/farajtabar20a.html.

Rui Gao and Weiwei Liu. Defying catastrophic forgetting via influence function. Artificial Intelligence, 339:104261, 2025. doi: 10.1016/j.artint.2024.104261.

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.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9304–9312, 2020.

Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 792–801, 2021.

Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In International Conference on Machine Learning, pp. 573–582. PMLR, 2016.

Han Guo, Nazneen Fatema 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. URL https://aclanthology.org/2021.emnlp-main.808/.

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

Hangzhou He, Lei Zhu, Xinliang Zhang, Shuang Zeng, Qian Chen, and Yanye Lu. V2c-cbm: Building concept bottlenecks with vision-to-concept tokenizer. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39(3), pp. 3401–3409, 2025. doi: 10.1609/aaai.v39i3.32352.

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 Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pp. 24678–24726. PMLR, 2025. URL https://proceedings.mlr.press/v267/hu25u.html.

Amirhosein Javadi, Tuomas Oikarinen, Tara Javidi, and Tsui-Wei Weng. CI-CBM: Class-incremental concept bottleneck model for interpretable continual learning. Transactions on Machine Learning Research, 2026. URL https://arxiv.org/abs/2604.14519.

Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. In Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pp. 16521–16540. PMLR, 2023. URL https://proceedings.mlr.press/v202/kim23g.html.

Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=THOOBy1uWVH.

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.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In Advances in Neural Information Processing Systems, volume 32, 2019.

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. doi: 10.1145/3746027.3758157.

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. URL https://arxiv.org/abs/2601.00451.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pp. 6781–6792. PMLR, 2021.

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.

Luca Salvatore Lorello, Marco Lippi, and Stefano Melacci. Continual learning for unsupervised concept bottleneck discovery. In Proceedings of the 3rd Conference on Lifelong Learning Agents, volume 274 of Proceedings of Machine Learning Research, pp. 597–619. PMLR, 2025. URL https://proceedings.mlr.press/v274/lorello25a.html.

James Martens. New insights and perspectives on the natural gradient method. Journal of Machine Learning Research, 21(146):1–76, 2020.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10422–10431, 2022.

Michael Nevitt, David Felson, and Gayle Lester. The osteoarthritis initiative: Protocol for the cohort study. Technical report, National Institute of Arthritis and Musculoskeletal and Skin Diseases, 2006. Version 1.1.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In Sixth Indian Conference on Computer Vision, Graphics and Image Processing, pp. 722–729. IEEE, 2008.

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Lily Weng. Label-free concept bottleneck models. In International Conference on Learning Representations, 2023.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In International Conference on Learning Representations, 2020.

Ivaxi Sheth and Samira Ebrahimi Kahou. Auxiliary losses for learning generalizable concept-based models. In Advances in Neural Information Processing Systems, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/555479a201da27c97aaeed842d16ca49-Abstract-Conference.html.

Ao Sun, Qingyue Wang, Yuanyuan Yuan, Pingchuan Ma, and Shuai Wang. Hi-cbm: Mitigating information leakage via hierarchical concept bottleneck modeling. Neurocomputing, 675:132916, 2026. doi: 10.1016/j.neucom.2026.132916.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), pp. 303–319. IEEE, 2022.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10962–10971, 2023. doi: 10.1109/CVPR52729.2023.01055.

Nuoye Xiong, Anqi Dong, Ning Wang, Cong Hua, Guangming Zhu, Lin Mei, Peiyi Shen, and Liang Zhang. Intervening in black box: Concept bottleneck model for enhancing human neural network mutual understanding. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2836–2845, 2025.

Sin-Han Yang, Tuomas P. Oikarinen, and Tsui-Wei Weng. Concept-driven continual learning. Transactions on Machine Learning Research, 2024.

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

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In The Eleventh International Conference on Learning Representations, 2023.

Valentina Zaccaria and Gian Antonio Susto. Enhancing usability in concept bottleneck models via grouped concept interventions. IFAC-PapersOnLine, 59(26):73–78, 2025. doi: 10.1016/j.ifacol.2025.12.013.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 58244–58282. PMLR, 2024. URL https://proceedings.mlr.press/v235/zarifzadeh24a.html.

A Extended Related Work

Concept bottleneck models. CBMs Koh et al. (2020) introduce a structural bottleneck where inputs are first mapped to predefined concepts, which subsequently vote on the final classification. Chauhan et al. Chauhan et al. (2023) enable efficient test-time interaction with predicted concepts, while Oikarinen et al. Oikarinen et al. (2023) use multimodal embeddings to build Label-Free CBMs without dense concept annotation. Subsequent work has explored probabilistic formulations Kim et al. (2023), auxiliary losses Sheth & Ebrahimi Kahou (2023), concept embeddings Espinosa Zarlenga et al. (2022), and post-hoc implementations Yuksekgonul et al. (2023). He et al. He et al. (2025) learn a vision-to-concept mapping over a filtered textual vocabulary to reduce LLM-induced non-visual noise, and Dominici et al. Dominici et al. (2025) introduced Counterfactual CBMs. Wang et al. Wang et al. (2023) studied learning bottleneck concepts from images, Xiong et al. Xiong et al. (2025) explored interventions for black-box CBMs, and Zaccaria and Susto Zaccaria & Susto (2025) studied grouped concept interventions. Editable CBM Hu et al. (2025) supports concept–label, concept, and data edits with influence-function approximations, while Controllable CBM adds data insertion to this editing family Lin et al. (2026). Dynamic-V2C combines this editing perspective with clustered patch prototypes and continual expansion.

Machine unlearning and influence functions. The influence function, initially a tool in robust statistics Cook & Weisberg (1980); Cook (2000), was introduced to deep learning by Koh and Liang Koh & Liang (2017) to trace predictions to training samples. IFs are useful for debugging Guo et al. (2021); Aamir et al. (2023), but their application to deep non-convex models is hindered by fragile Hessian approximations Basu et al. (2021) and high computational cost. Machine-unlearning strategies include sharded retraining Bourtoule et al. (2021), exact statistical removal Brophy & Lowd (2021), Hessian-based updates Mehta et al. (2022), unrolled optimization Thudi et al. (2022), and selective forgetting Golatkar et al. (2020; 2021). K-FAC and related natural-gradient approximations reduce inverse-curvature cost Grosse & Martens (2016); Martens (2020); George et al. (2018). Dynamic-V2C confines this approximation to the shallow bottleneck and treats its output-level edits as distinct from certified or backbone-level removal; Hi-CBM Sun et al. (2026) instead addresses hierarchical concept-information leakage.

Continual learning. Continual learning aims to acquire new knowledge without catastrophic forgetting. CLG-CBM Yu et al. (2025) uses language-guided CBMs to maintain semantic consistency across sequential tasks, concept-driven continual learning manages explicit concepts over time Yang et al. (2024), and CI-CBM studies class-incremental concept regularization and pseudo-concept generation Javadi et al. (2026). Other work explores concept expansion Lai et al. (2025), unsupervised concept discovery Lorello et al.

(2025), IF-guided mitigation Gao & Liu (2025), and orthogonal-gradient protection Farajtabar et al. (2020). Dynamic-V2C operates on a visually grounded V2C bottleneck and combines influence-triggered expansion with damped protection of historical activation directions.

B Broader Impacts and Limitations

Broader Impacts. Fast, inspectable bottleneck edits can support model debugging, subgroup auditing, and incident response, including in high-stakes settings. They can also encourage premature deployment if an output-level correction is mistaken for certified deletion, legal compliance, or a fairness guarantee. Dynamic-V2C should therefore complement, rather than replace, domain-specific validation and full retraining when required.

Limitations and Backbone Bias. Dynamic-V2C relies on a frozen pre-trained Vision-Language backbone such as CLIP to initialize the V2C tokenizer. Consequently, its edits operate at the concept bottleneck and prediction head; they do not mathematically erase data footprints from frozen backbone features. If the CLIP embedding space collapses relevant distinctions or encodes demographic biases (e.g., incorrectly conflating medical anomalies in OAI or entangling facial features with gender in CelebA), the shallow tokenizer cannot necessarily recover them. Correcting these representation-level failures would require unfreezing the backbone and would forgo much of the current speed and memory advantage. Scaling the vocabulary to extreme sizes (e.g., $K > 100{,}000$) may also reintroduce memory bottlenecks during factored eigendecomposition.

Practical scope. The scope of these claims is intentionally bounded. Because the visual backbone remains frozen, our editing and unlearning results apply to concept routing, bottleneck parameters, and output behavior rather than certified erasure of all feature-level traces. Likewise, the V2C vocabulary should be interpreted as a set of visually grounded prototypes with post-hoc semantic labels, not as a complete substitute for manually verified causal concepts. These design choices trade perfect semantic intervention for fast, auditable correction, and they leave open future work on backbone-level unlearning, stronger privacy attacks, and adaptive vocabulary expansion under broader domain shifts.

In practice, Dynamic-V2C is best viewed as a diagnostic layer complementary to full retraining: it previews how removing data, suppressing concepts, or expanding the vocabulary changes predictions before expensive backbone updates.

C Implementation Details and Hyperparameters

Experiments use PyTorch on a single NVIDIA A100 (80GB) GPU. Unless a table states otherwise, concept-based methods use a frozen CLIP ViT-B/16; the Food-101 scale test uses ViT-L/14, and the ResNet-50 CelebA row is a separate black-box reference. The V2C tokenizer is initialized using 200,000 unlabeled images randomly sampled from ImageNet. The default static vocabulary size is $K = 10,000$, while Split-CIFAR100 starts from $K = 1,000$ to measure growth. EK-FAC uses Levenberg–Marquardt damping $\delta = 10^{-3}$, and the fixed continual-learning expansion threshold is $\tau = 0.15$.

D Supplementary Experiments

D.1 Additional Fidelity, Privacy-Boundary, and Stream Stress Tests

We include the additional stress tests used to evaluate scale, approximation fidelity, the boundary of head-level unlearning, and threshold robustness.

Table 4: IF fidelity and scale in our shared implementation and hardware. The first two rows report retrain/edit F1; the Food-101 row reports the absolute accuracy gap recorded by the fixed stress-test protocol.

Setting	Removal	Retrain / Dynamic-V2C utility	Retrain / edit time
CUB, ViT-B/16	3%	0.7971 / 0.7963 F1	5,133s / 39s
OAI, ViT-B/16	3%	0.8825 / 0.8808 F1	17,866s / 141s
Food-101, ViT-L/14	5%	< 0.6 accuracy-point gap	3.2h / 112s

Table 5: Privacy boundary before/after editing. Output leakage drops toward chance, whereas frozen-feature leakage remains essentially unchanged.

Attack view	Before	After
Output RMIA AUC (CUB)	0.89	0.51
Feature MIA AUC (CUB)	0.54	0.55
Feature MIA AUC (OAI)	0.56	0.56

Table 6: Threshold robustness. The adaptive rule uses running statistics from the observed stream, without future test labels.

Stream / rule	FAA	BWT	Growth
Split-CIFAR100, fixed	78.5±0.6	−3.1±0.4	1000→1350
Mixed, fixed τ	75.1	−5.4	$K\times1.39$
Mixed, adaptive τ	76.4	−4.6	$K\times1.32$

D.2 Complete Continual-Learning Results and Discussion

We evaluate our method on the Split-CIFAR100 benchmark, dividing the 100 classes into 10 sequential tasks. Task identity is provided during training and evaluation, and the concept vocabulary dynamically expands across tasks without re-initialization. For EK-FAC, we use a mini-batch size of 256, expanding the vocabulary when the influence score exceeds $\tau = 0.15$.

As shown in Table 7, traditional CBMs degrade significantly in this regime (FAA: 46.2±1.4%, BWT: $-42.5 \pm 1.8\%$) due to unrestricted overwriting of the linear label predictor f. While EWC, LwF, and our task-aware reimplementation of CLG-CBM Yu et al. (2025) mitigate this, Dynamic-V2C combines influence-triggered vocabulary expansion with covariance-based attenuation of historical-direction updates. Vocabulary expansion alone improves FAA to 58.4%, while historical-direction protection alone reaches 65.2%; their combination reaches 78.5%. The vocabulary grows from 1,000 to approximately 1,350 concepts over 10 tasks. Dynamic-V2C obtains FAA $78.5 \pm 0.6\%$ and BWT $-3.1 \pm 0.4\%$, the strongest result among the evaluated baselines.

Table 7: Continual-learning performance on task-aware Split-CIFAR100 (10 tasks). All rows are our matched reimplementations; results are averaged over 3 seeds.

Method	Final Avg Acc (FAA) % ↑	Backward Transfer (BWT) % ↑
Standard CBM	46.2 ± 1.4	-42.5 ± 1.8
CBM + EWC	62.4 ± 1.1	-18.6 ± 1.4
CBM + LwF	65.1 ± 1.3	-15.4 ± 1.5
CLG-CBM Yu et al. (2025)	72.8 ± 0.9	-8.2 ± 0.8
Dynamic-V2C (Ours)	78.5 ± 0.6	-3.1 ± 0.4

Component ablations. Table 8 separates the V2C tokenizer, dynamic vocabulary, historical-direction protection, and EK-FAC acceleration. The two single-component rows report FAA

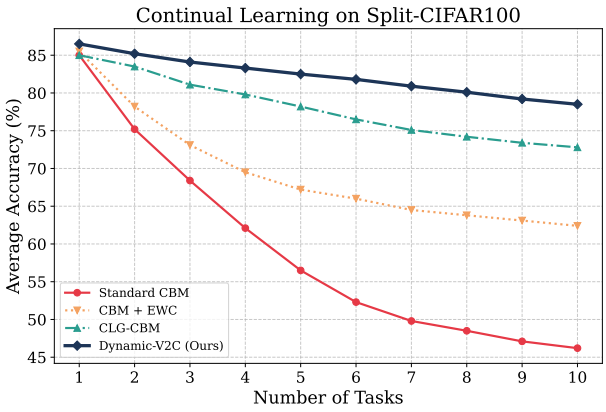


Figure 4: Continual learning on Split-CIFAR100 over 10 tasks. Dynamic-V2C reduces forgetting by protecting historically active directions.

from the corresponding continual-learning checks; BWT and edit time were not recorded for those isolated checks. Combining expansion and protection yields the largest FAA gain, while EK-FAC reduces the CUB edit from 1,240s to 39s with similar continual-learning metrics.

As shown in Table 3, Dynamic-V2C elevates worst-group accuracy to 81.5% in our Gender-prediction protocol, reducing the measured association between hair color and the target. The edit executes in under 2 seconds and outperforms our Editable-CBM reimplementation; the result is consistent with the benefit of the patch-prototype bottleneck, although this comparison does not isolate visual grounding as the sole causal factor.

D.3 Additional qualitative visualizations

The main paper now presents the editing case study in Figure 3. We retain the continual-learning curve and broader per-example prototype visualizations here as complementary evidence.

To provide transparent decision-making insights, we visualize prototype contributions for individual predictions on the CUB dataset. Given an input image, a unit's contribution is the product of its bottleneck activation and linear-classifier weight. Figure 5 annotates the resulting visual prototypes with their nearest post-hoc text labels. For a "Wood Duck," highly contributing units receive labels such as duck shape and crested head, while a negatively contributing unit is labeled curved beak. On CelebA, examples include heavy makeup and wavy hair. These labels are inspection aids rather than supervised semantic concepts; together with the quantitative interventions below, they make the learned visual units auditable.

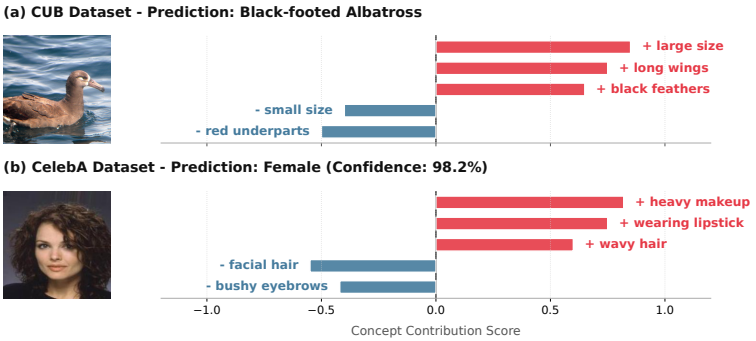


Figure 5: Qualitative concept visualization on CUB and CelebA. Bar charts show the top positive and negative concept contributions driving each prediction.

D.4 Ablation Study: V2C Tokenizer Vocabulary Size

Table 8: Component ablation on task-aware Split-CIFAR100 and CUB editing. A dash denotes a metric not recorded in the isolated component check.

Components					Continual (CIFAR100)		Edit Runtime
Base	V2C	Dyn. vocab.	Protection	EK-FAC	FAA ↑	BWT ↑	CUB (s) ↓
✓					46.2	-42.5	5,133 (Retrain)
✓	✓				46.8	-41.8	5,050 (Retrain)
✓	✓	✓			58.4	–	–
✓	✓		✓		65.2	–	–
✓	✓	✓	✓		78.1	-3.3	1,240
✓	✓	✓	✓	✓	78.5	-3.1	39

To understand the impact of tokenizer capacity, we vary the base vocabulary from 1,000 to 30,000. Figure 6 shows CUB accuracy increasing from 76.5% to approximately 80% by $K = 10{,}000$, after which both curves saturate; no degradation is observed within the tested range.

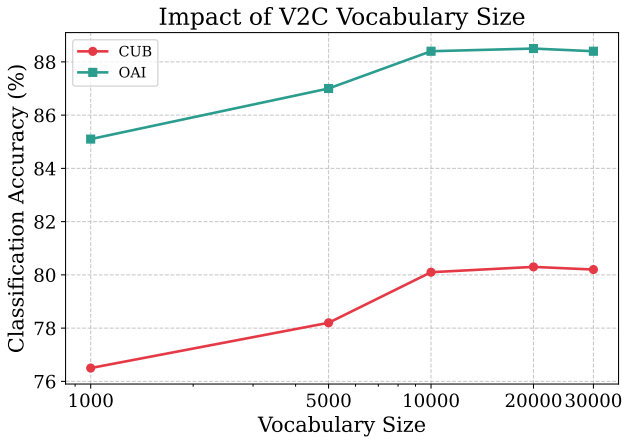


Figure 6: Classification accuracy versus V2C vocabulary size on CUB and OAI. Performance saturates around 10,000 prototypes.

D.5 Concept-Level Interpretability Validation

A core motivation of the influence function is to estimate the local importance of each parameter or prototype. We validate our concept-level formulation by ranking CUB prototypes by their computed influence score and editing the top- and bottom-ranked sets.

Upon removing the Top-10 influential concepts (e.g., "brown feathers", "spotted pattern"), the model's F1 score drops by approximately 0.10. Conversely, removing the Bottom-10 concepts changes F1 by less than 0.005. The F1 divergence between Dynamic-V2C's local edit and full retraining after these concept removals is below 0.001. These interventions quantitatively support the ranking's local faithfulness; Figure 7 provides the complete curve.

Table 9: Quantitative faithfulness check for concept-level edits.

Edit	Observed effect	Interpretation
Top-10 prototypes	~ 0.10 F1 drop	Decision-relevant
Bottom-10 prototypes	< 0.005 F1 change	Low influence
Edit vs. retrain	< 0.001 F1 divergence	Faithful local approximation

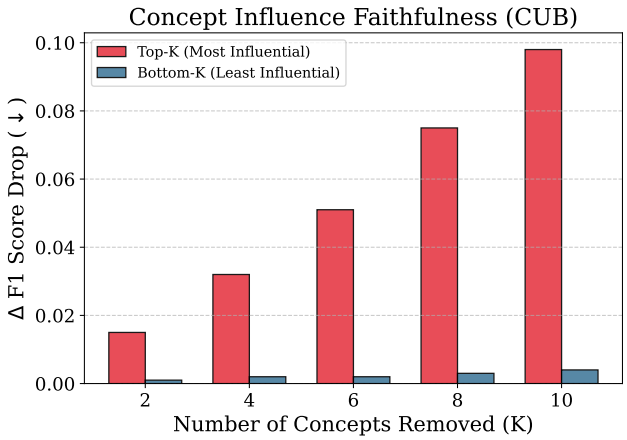


Figure 7: Concept faithfulness. Removing Top-K influential prototypes causes a larger performance drop than removing Bottom-K prototypes, supporting the local influence ranking.

D.6 Supplementary Concept Interpretability Visualizations

The main-paper CUB case study in Section 5.6 demonstrates a concept-level edit. Here we provide complementary CUB and CelebA prototype visualizations across additional classes.

As shown in Figures 10 and 11, Dynamic-V2C extracts inspectable visual prototypes across domains. Their nearest text labels include 'curved beak' for the Groove-billed Ani, 'yellow eye' for the Brewer Blackbird, and 'blue feathers' for the Indigo Bunting. CelebA examples include 'mustache', 'arched eyebrows', and '5 o clock shadow' for units used in gender and age predictions. These qualitative examples show the patch evidence associated with each prototype; they do not turn every unsupervised cluster into a manually validated semantic concept.

D.7 Detailed Setup for Robust Membership Inference Attack (RMIA)

In the main-paper privacy evaluation, RMIA measures the output-level membership signal after editing. We use the score-based black-box setting: the attacker observes output confidence scores but not model parameters or hidden representations. Independently trained reference models estimate the population score distribution, and a likelihood-ratio statistic

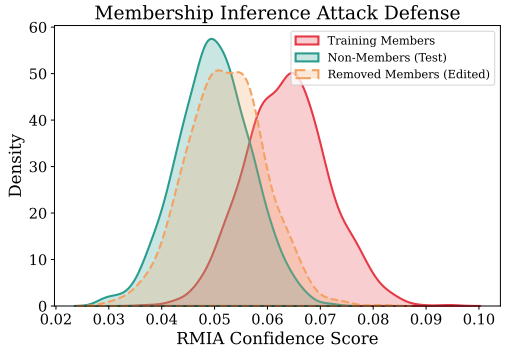


Figure 8: RMIA density. Post-editing, the "Removed Members" distribution closely aligns with entirely unseen Test data, supporting reduced output-level membership leakage in the evaluated head.

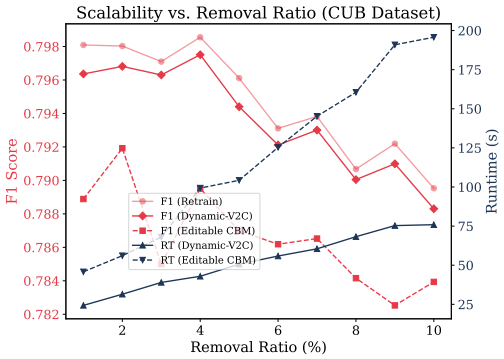


Figure 9: Scalability study on the CUB dataset. We show F1 score and runtime scaling as the proportion of removed/unlearned data increases.

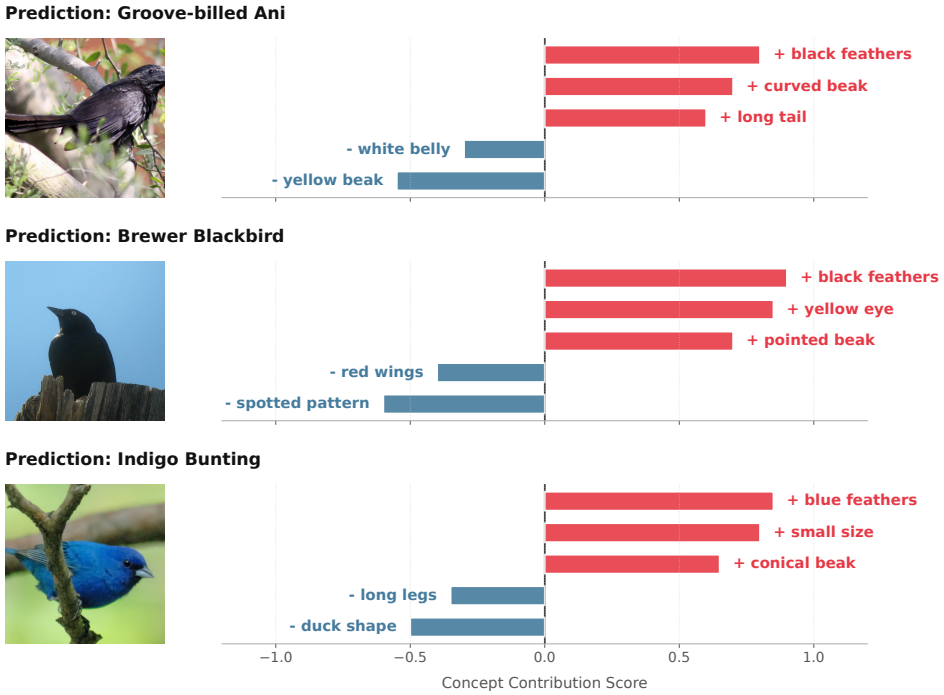


Figure 10: Additional CUB visualizations. Bars report contributions of clustered patch prototypes annotated with nearest post-hoc text labels.

ranks samples as members or non-members. We evaluate a removal set of $n = 200$ samples across 5 independent runs. The edited-member and non-member distributions in Figure 8 become closely aligned, supporting reduced membership leakage at the evaluated head. Separately, we expose frozen embeddings to a feature-space MIA as a white-box boundary probe; its unchanged AUC in Table 5 shows that the experiment does not establish feature-level erasure.

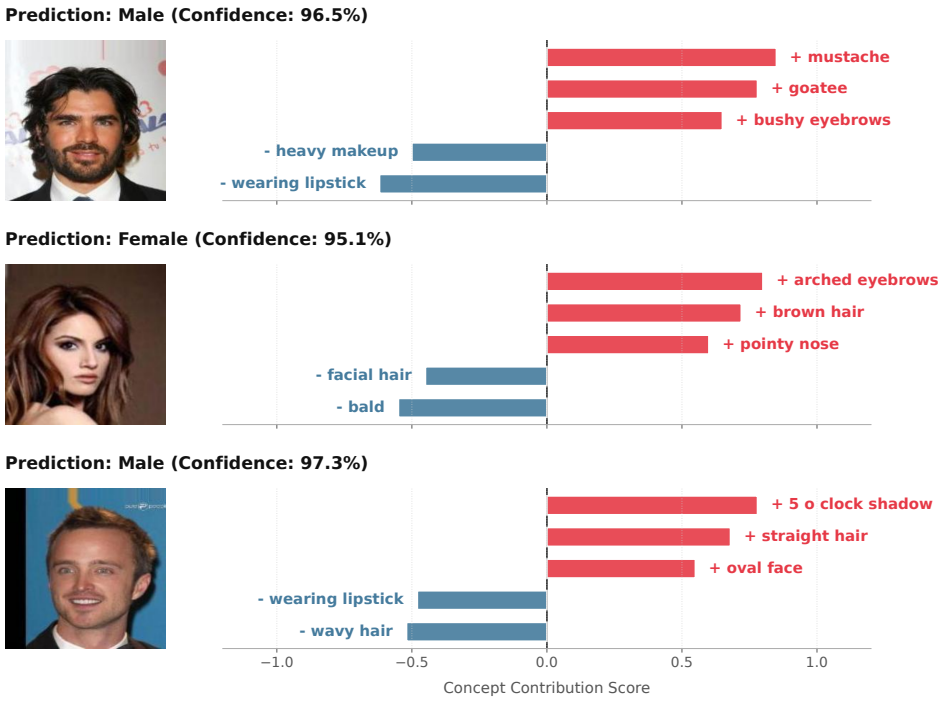


Figure 11: Additional CelebA visualizations. Bars report contributions of clustered patch prototypes annotated with nearest post-hoc text labels; labels are inspection aids, not supervised attributes.

E Theoretical Proofs for Data-Level Editing

In this section, we provide the formal proof for the data-level editing theorem in the main paper, demonstrating how the retrained concept predictor $\hat{g}_{-z_G}$ can be approximated via the Influence Function without full retraining.

Proof. We rewrite the optimal concept predictor $\hat{g}_{-z_G}$ trained after removing the dataset G as:

$$\hat{g}_{-z_G} = \arg\min_{g} \left[\sum_{i=1}^{n} L_C(g(x_i), c_i) - \sum_{r \in G} L_C(g(x_r), c_r) \right]$$

To leverage the influence function, we construct an ϵ-parameterized response function $\hat{g}_\epsilon$, where the removed data G is instead upweighted by a factor ϵ:

$$\hat{g}_\epsilon \triangleq \arg\min_{g} \left[\sum_{i=1}^{n} L_C(g(x_i), c_i) - \epsilon \cdot \sum_{r \in G} L_C(g(x_r), c_r) \right] \tag{7}$$

Because $\hat{g}_\epsilon$ is the minimizer of this objective, the first-order optimality condition dictates:

$$\nabla_{\hat{g}_\epsilon} \sum_{i=1}^{n} L_C(\hat{g}_\epsilon(x_i), c_i) - \epsilon \cdot \nabla_{\hat{g}_\epsilon} \sum_{r \in G} L_C(\hat{g}_\epsilon(x_r), c_r) = 0$$

As $\epsilon \to 0$, $\hat{g}_\epsilon \to \hat{g}$. We perform a first-order Taylor expansion around the original unedited parameters $\hat{g}$:

$$\nabla_g \sum_{i=1}^{n} L_C(\hat{g}(x_i), c_i) - \epsilon \cdot \nabla_g \sum_{r \in G} L_C(\hat{g}(x_r), c_r) + \nabla_g^2 \sum_{i=1}^{n} L_C(\hat{g}(x_i), c_i) \cdot (\hat{g}_\epsilon - \hat{g}) \approx 0$$

By the definition of the original model, $\nabla_g \sum_{i=1}^{n} L_C(\hat{g}(x_i), c_i) = 0$. Therefore, solving for the derivative with respect to ϵ evaluated at $\epsilon = 0$ yields:

$$\left.\frac{\mathrm{d}\hat{g}_\epsilon}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = H_{\hat{g}}^{-1} \cdot \sum_{r \in G} \nabla_g L_C(\hat{g}(x_r), c_r)$$

where $H_{\hat{g}} = \nabla_g^2 \sum_{i=1}^{n} L_C(\hat{g}(x_i), c_i)$ is the Hessian matrix evaluated on the full dataset.

When $\epsilon = 1$, the response function mathematically equates to complete removal ($\hat{g}_\epsilon = \hat{g}_{-z_G}$). By executing a single Newton step of size 1 at $\hat{g}$, we obtain the approximated update (relying on accurate curvature estimation and damping):

$$\hat{g}_{-z_G} \approx \bar{g}_{-z_G} \triangleq \hat{g} + H_{\hat{g}}^{-1} \cdot \sum_{r \in G} \nabla_g L_C(\hat{g}(x_r), c_r)$$

which mirrors the formulation in the main-paper data-level editing theorem. Under the local Taylor approximation and damped curvature model, this update approximates retraining the concept projection layer without G; it is not an exact or certified erasure guarantee.

E.1 Two-Step Update for the Label Predictor $\hat{f}$

When the concept predictor $\hat{g}$ shifts to $\bar{g}_{-z_G}$, the concept activations for the entire dataset shift from $c_i = \hat{g}(x_i)$ to $\bar{c}_i = \bar{g}_{-z_G}(x_i)$. Consequently, the optimal label predictor $\hat{f}$ must be updated not only to "forget" the data G, but also to adapt to the new concept representations $\bar{c}_i$.

We resolve this via a two-step sequential approximation:

1. Data Removal Step: First, we apply the standard influence function to remove the direct influence of G on $\hat{f}$, assuming the concepts c_i remain fixed:

$$\tilde{f} = \hat{f} + H_{\hat{f}}^{-1} \cdot \sum_{r \in G} \nabla_f L_Y(\hat{f}(c_r), y_r) \tag{8}$$

   where $H_{\hat{f}} = \sum_{i=1}^{n} \nabla_f^2 L_Y(\hat{f}(c_i), y_i)$.

2. Representation Shift Step: Next, we account for the shift $\Delta c_i = \bar{c}_i - c_i$. By applying a first-order Taylor expansion to the gradient of the loss with respect to the input features, the shift in the optimality condition induces a corresponding parameter shift. To prevent this step from scaling with the full dataset size n, we approximate the sum using a uniformly sampled mini-batch $\mathcal{B} \subset \mathcal{D} \setminus G$. The final updated predictor is:

$$\bar{f}_{-z_G} = \tilde{f} - \frac{n - |G|}{|\mathcal{B}|} H_{\hat{f}}^{-1} \cdot \sum_{i \in \mathcal{B}} \nabla_{f,c}^2 L_Y(\hat{f}(c_i), y_i) \cdot \Delta c_i \tag{9}$$

   where $\nabla_{f,c}^2$ represents the mixed partial derivatives of the loss with respect to the weights f and the input concepts c.

This sequential formulation ensures that the label predictor remains optimally calibrated to the newly updated bottleneck, avoiding the severe performance degradation that would occur if we only updated $\hat{g}$. □

F Theoretical Proofs for Concept-Level Editing

When a concept p_r (for $r \in M$) is determined to be entirely spurious, we must remove its influence from the label predictor $\hat{f}$. We demonstrate that applying an Influence Function update on the original parameter space with a masked projection provides a principled first-order approximation to retraining $\hat{f}_{\setminus M}$ on the reduced dimension space.

Proof. Let $W \in \mathbb{R}^{d_o \times k}$ be the weight matrix of the full linear label predictor. Removing concept set $M \subset [k]$ would ordinarily produce a predictor in $\mathbb{R}^{d_o \times (k-|M|)}$.

To avoid dimension mismatch during an influence update, let $P_M \in \mathbb{R}^{k \times k}$ be diagonal with $(P_M)_{j,j} = 0$ for $j \in M$ and 1 otherwise. We retain the original parameter space and impose $W(I - P_M) = 0$:

$$\min_{W \in \mathbb{R}^{d_o \times k}} \sum_{i=1}^{n} L_Y(W P_M c_i, y_i) \quad \text{subject to} \quad W(I - P_M) = 0.$$

Any reduced-dimensional linear predictor has an equivalent full-dimensional representation satisfying this constraint, obtained by inserting zero columns on M. Conversely, $W P_M c_i$ depends only on retained coordinates.

Consequently, we evaluate the masked objective in the original full-dimensional space and take one damped Newton step toward its optimum:

$$W^+ \triangleq \hat{W} - (H_M + \delta I)^{-1} \sum_{i=1}^{n} \nabla_W L_Y \left(\hat{W} P_M \hat{c}_i, y_i \right),$$
$$\bar{W}_{\setminus M} \triangleq \Pi_M(W^+) = W^+ P_M.$$

The final projection exactly enforces zero classifier weights on the removed coordinates, while the Newton step is a local approximation to retraining the retained coordinates. Working in the original parameter space bypasses architectural dimension mismatch; it does not erase the corresponding visual pattern from the frozen backbone. □

G Extended Experimental and Implementation Details

G.1 Pseudo-code for Dynamic-V2C Editing Engine

To facilitate the reproducibility and practical implementation of our framework, we provide the step-by-step pseudo-code for the rapid Data-Level head-editing process in Algorithm 1.

G.2 Hyperparameter Calibration, Damping, and Variance

To ensure the reproducibility of our results, we report key hyperparameter calibrations and variance metrics that were omitted from the main text due to space constraints. For EK-FAC, we set the Levenberg–Marquardt damping factor $\delta = 10^{-3}$. Adding δI makes the positive-semidefinite factored Fisher approximation invertible. We selected it from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$: 10^{-4} produced ill-conditioned, unstable updates, whereas 10^{-1} behaved like a weakly preconditioned gradient step. The factors and eigendecomposition are refreshed once at the start of each editing request and reused across its requested samples; reported edit time includes this refresh. Classification, targeted-removal, spurious-correlation, and Split-CIFAR100 results report means and standard deviations over 3 independent random seeds. The RMIA/privacy experiment uses a removal set of 200 samples and 5 independent runs. The Food-101 ViT-L/14 and mixed-stream stress tests use the fixed protocols reported above.

G.3 Expansion-threshold sensitivity

We further analyzed the sensitivity of the dynamic expansion threshold τ. Setting τ too low leads to excessive vocabulary growth, while an overly high τ fails to allocate capacity for genuinely novel visual features. Through ablation across datasets, we determined that $\tau \in [0.10, 0.15]$ optimally balances compactness with backward-transfer preservation on coarse-grained tasks, while fine-grained tasks benefit from $\tau \approx 0.08$. In a mixed CUB/CIFAR stream, a fixed τ remains bounded (FAA 75.1, BWT -5.4, $K\times1.39$), and an adaptive τ based on running stream statistics further improves stability (FAA 76.4, BWT -4.6, $K\times1.32$); no vocabulary explosion is observed.

Algorithm 1 Dynamic-V2C Rapid Data-Level Head Editing

Require: Original trained Concept Predictor $\hat{g}$ and Label Predictor $\hat{f}$, Damping factor δ, Data subset G to be unlearned.
Ensure: Edited Predictors $\bar{g}_{-z_G}$ and $\bar{f}_{-z_G}$.

Phase 1: Per-request EK-FAC curvature refresh
Extract activation covariance $\hat{\Omega}$ and gradient covariance $\hat{\Gamma}$ from the V2C projection layer.
Compute Eigen-decompositions: $\hat{\Omega} = Q_\Omega \Lambda_\Omega Q_\Omega^\top$, $\hat{\Gamma} = Q_\Gamma \Lambda_\Gamma Q_\Gamma^\top$.
Construct inverse Hessian approx $\hat{H}_{\hat{g}}^{-1}$ via the EK-FAC formulation.
Phase 2: Concept Predictor Update
Initialize gradient accumulator: $\nabla_{total} \leftarrow 0$.
for each sample $r \in G$ **do**
    Compute $\nabla_{g_{proj}} L_C(x_r, c_r; \hat{g})$.
    $\nabla_{total} \leftarrow \nabla_{total} + \nabla_{g_{proj}} L_C$.
end for
Apply Newton step: $\bar{g}_{-z_G} \leftarrow \hat{g} + \hat{H}_{\hat{g}}^{-1} \cdot \nabla_{total}$.
Phase 3: Label Predictor Update (Two-Step)
Prepare damped inverse-curvature–vector products for the linear head $\hat{f}$.
// Step 3a: Remove data influence
$\tilde{f} \leftarrow \hat{f} + \hat{H}_{\hat{f}}^{-1} \cdot \sum_{r \in G} \nabla_f L_Y(\hat{f}(c_r), y_r)$.
// Step 3b: Compensate for concept activation shift
Sample a fixed-size mini-batch $\mathcal{B} \subset \mathcal{D} \setminus G$ to avoid a full retained-data pass.
Calculate activation shift $\Delta c_i \leftarrow \bar{g}_{-z_G}(x_i) - \hat{g}(x_i)$ for $i \in \mathcal{B}$.
$\bar{f}_{-z_G} \leftarrow \tilde{f} - \frac{n-|G|}{|\mathcal{B}|} \hat{H}_{\hat{f}}^{-1} \cdot \sum_{i \in \mathcal{B}} \nabla^2_{f,c} L_Y(\hat{f}(c_i), y_i) \cdot \Delta c_i$.
return $\bar{g}_{-z_G}$ and $\bar{f}_{-z_G}$.

G.4 Dynamic Vocabulary Expansion via Residual Features

In Section 4.4, we introduced dynamic expansion of the V2C tokenizer. For each normalized patch embedding v, let p_{j^*} be its nearest normalized historical prototype. We define the quantization residual as $r(v) = v - p_{j^*}$ and its novelty score as $\|r(v)\|_2$. When task influence exceeds τ, high-novelty residuals from the observed task are clustered with k-means into an expansion budget of m new prototypes, which are appended without reinitializing the historical slots. This nearest-prototype residual remains meaningful even when the complete prototype matrix spans the encoder space; finite-sample clustering does not guarantee semantic novelty.

G.5 Damped Orthogonal Gradient Projection

During continual learning, we protect historical concepts using the activation covariance $\Sigma_{old} = \mathbb{E}[\hat{c}\hat{c}^\top]$. A hard projector onto the empirical null space would require choosing the rank of Σ_{old}. We instead use the ridge-stable soft projector

$$P_\alpha = I - \Sigma_{old}(\Sigma_{old} + \alpha I)^{-1} = \alpha(\Sigma_{old} + \alpha I)^{-1}, \qquad \widetilde{\nabla} W_{old} = \nabla W_{old} P_\alpha,$$

where $\alpha > 0$ is a smoothing term. In the eigenbasis of Σ_{old}, directions with large historical variance receive proportionally smaller updates; as $\alpha \to 0$, P_α approaches the projector onto the exact null space. For finite α it is a soft protection mechanism rather than an idempotent projector or a formal no-forgetting guarantee.

G.6 RMIA Attacker Specification

Following Zarifzadeh et al., output-score RMIA uses black-box confidence to compare each sample with a population distribution estimated by five reference models trained on independently sampled splits. ROC-AUC and density plots average five runs. This tests

output-score leakage; the feature-space MIA in Table 5 is the separate white-box boundary probe.