Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Bayesian Concept Consolidation for Concept-Level Stability in Continual Learning

Simachew Alamneh

simachewalamneh3@gmail.com

Bonga University

mohammed Abebe

Arba Minch University

Chernet Erdachew

Bonga University

Research Article

Keywords: continual learning, catastrophic forgetting, Bayesian neural networks, representation stability, elastic weight consolidation, concept drift

Posted Date: September 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10766033/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Bayesian Concept Consolidation for Concept-Level Stability in Continual Learning

Simachew Alamneh
Department of Computer Science
Bonga University
Bonga, Ethiopia
Email: simachewalamneh3@gmail.com
ORCID: 0009-0008-7415-5642

Chernet Erdachew Yirgu
Department of Computer Science
Bonga University
Bonga, Ethiopia
Email: cherneterdachew@gmail.com
ORCID: 0009-0001-9415-7484

Mohammed Abebe, Ph.D.
Arba Minch Institute of Technology
Arba Minch University
Arba Minch, Ethiopia
Email: mohammed.abebe@amu.edu.et
ORCID: 0000-0003-0622-4841

Corresponding author:
simachewalamneh3@gmail.com

***Abstract*—Continual learning methods commonly mitigate catastrophic forgetting by constraining parameter change (e.g., Elastic Weight Consolidation) or by rehearsing stored samples (replay). Neither approach directly measures or protects the semantic content encoded in a model's internal representations, leaving open the question of whether accuracy retention implies that a class's learned *concept* remains stable. We propose Bayesian Concept Consolidation (BCC), a continual learning framework that introduces a concept-stability loss operating on a Bayesian concept bottleneck layer, together with a posterior-variance-derived adaptive weighting term and an uncertainty-prioritized replay buffer. To evaluate concept-level drift independently of task accuracy, we introduce the Knowledge Stability Index (KSI), defined as the cosine similarity between a class's concept representation at first acquisition and at a later evaluation point. On Split-MNIST (5 tasks, 2 classes each), a controlled ablation over 3 random seeds shows that concept-stability regularization improves KSI from 0.467 ± 0.069 (no concept loss) to 0.997 ± 0.0004 (full method) at statistically indistinguishable task accuracy (94.3%–94.7% across conditions). Under the evaluated protocol, a diagonal-Fisher Elastic Weight Consolidation (EWC) baseline without replay collapses to near-chance accuracy ($19.6\% \pm 0.1\%$), an outcome we trace to an early task-boundary gradient shock that outpaces Fisher-penalty accumulation; adding replay to EWC restores accuracy ($93.2\% \pm 0.7\%$) but not concept stability (KSI 0.476 ± 0.050, statistically indistinguishable from replay alone), isolating concept-stability regularization – rather than replay – as the mechanism responsible for concept preservation. These results indicate that accuracy retention and semantic concept stability are dissociable quantities under the studied conditions, and that existing regularization and replay mechanisms address the former without necessarily addressing the latter.**

***Index Terms*—continual learning, catastrophic forgetting, Bayesian neural networks, representation stability, elastic weight consolidation, concept drift**

I. Introduction

A. Background and Motivation

Continual learning requires a model to acquire new tasks sequentially without catastrophically forgetting previously learned ones. The dominant families of mitigation strategies are (i) regularization-based methods, which constrain change in parameters deemed important for prior tasks (e.g., Elastic Weight Consolidation, EWC [1]), (ii) Bayesian methods, which treat network weights as distributions and propagate posterior uncertainty across tasks (e.g., Variational Continual Learning, VCL [5]), and (iii) replay-based methods, which rehearse stored or generated samples from earlier tasks.

B. Problem and Existing Limitations

The standard evaluation protocol in continual learning is accuracy-centric. Catastrophic forgetting was originally characterized by McCloskey and Cohen [7] and later demonstrated at scale in modern connectionist networks by Goodfellow et al. [8] purely in terms of the drop in task performance after sequential training; this framing has persisted into the standard metrics used today. Backward Transfer (BWT) and Forward Transfer (FWT), introduced by Lopez-Paz and Ranzato alongside Gradient Episodic Memory (GEM) [11], formalize forgetting as the signed change in a previously learned task's accuracy after training on subsequent tasks, and remain the de facto standard for reporting continual learning results. Rehearsal methods such as iCaRL [10], Experience Replay [12], and Dark Experience Replay [13], and distillation-based methods such as Learning without Forgetting [9], are likewise validated primarily by their effect on these accuracy-derived quantities, as are architectural approaches such as Progressive Neural Networks [26] and PackNet [27], and class-incremental surveys such as Masana et al. [25] adopt accuracy, BWT, and FWT as the reporting standard across the field.

Why accuracy, BWT, and FWT are an incomplete lens. All three quantities are computed exclusively from the model's output layer: accuracy is the fraction of correct predictions, and BWT/FWT are differences of accuracy across task boundaries. None of them inspects what happens between the input and the output – specifically, whether the internal representation the network uses to recognize a class has stayed put or has reorganized. This is not a hypothetical gap. Kornblith

et al. [14] introduced Centered Kernel Alignment (CKA) precisely because two networks (or two snapshots of the same network) can differ substantially in internal representation while agreeing almost perfectly at the output. Ramasesh et al. [16] used representational analysis to show that catastrophic forgetting is disproportionately a deeper-layer phenomenon, with earlier-task representational subspaces measurably erased even when this is not fully visible from output accuracy alone. Davari et al. [15] made the dissociation explicit by showing that *representation forgetting* – the drop in linear-probe accuracy on frozen features – can be small even when a model's own classifier has ostensibly forgotten a task, and, conversely, that low output forgetting does not guarantee the representation has not moved; standard continual learning methods can score well on accuracy-based metrics while leaving the question of representational reorganization unanswered. Madaan et al. [17] extended this observation to unsupervised continual learning, and most directly, Filus et al. [23] recently proposed a concept-anchored diagnostic (using sparse-autoencoder-derived concept proxies) and found that concept-level information is frequently not erased but *hidden* – its recoverability and decodability change even when task-level accuracy narratives would suggest it is simply gone or simply retained. Taken together, this line of work indicates that accuracy, BWT, and FWT can be insensitive to exactly the kind of internal reorganization that a concept-level metric is designed to detect.

Existing continual learning evaluations predominantly emphasize task performance, parameter stability, or output behavior, leaving concept-level representational stability comparatively under-characterized. Two models can exhibit identical parameter drift, or identical accuracy retention, while differing substantially in whether their internal representation of "class c" has rotated in representation space; accuracy is silent on this distinction because a classifier can remain correct even as its underlying representation reorganizes. This is consistent with recent evidence from Semantic Drift Compensation [24], which shows that class-prototype drift in embedding space is a measurable and compensable quantity distinct from classification accuracy in class-incremental settings, and with broader continual learning surveys [18]–[20] that catalog regularization, replay, and architectural strategies primarily by their accuracy-preservation properties rather than by their effect on the underlying representation.

C. What We Mean by "Concept," and Why It Carries More Weight Than Parameters

We use *concept* in the sense developed by the concept-based interpretability literature: a concept is a compact, class-associated direction or region in a model's internal representation space that the model consistently activates for instances of that class, in the way Testing with Concept Activation Vectors (TCAV) [22] defines a concept as a direction in activation space, and Concept Bottleneck Models [21] define it as an explicit intermediate layer whose units are intended to correspond to human-nameable or class-associated attributes. In this paper, concretely, the concept for class c_k is the mean concept-layer activation vector $c_k^{(0)} \in \mathbb{R}^{32}$ computed over that class's training examples at the point the class is first learned (Section IIID) – *not* a raw parameter, and *not* a class label or logit.

It is useful to be explicit about the distinction between a *parameter* and a *concept*, since the two are easy to conflate:

- A **parameter** (θ, e.g., a single weight w_{ij} connecting unit i to unit j) is a scalar coordinate of the model. **Parameter stability** means $\theta_t \approx \theta_0$, typically measured as $\|\theta_t - \theta_0\|_2$ (as in Section VB) or protected via a per-parameter importance penalty, as in EWC [1] and Synaptic Intelligence [2]. A single parameter, in isolation, carries no semantic meaning: the same class can be represented by many different parameter configurations (the network is over-parameterized relative to the task), so two networks with very different parameter values can implement the identical function.
- A **concept** (here, $c_k \in \mathbb{R}^{32}$) is an aggregate, class-associated direction in *activation* space, computed from many parameters jointly via a forward pass. **Concept stability** means the concept vector's direction is preserved over time, $s_k(t) = \cos(c_k^{(0)}, c_k^{(t)}) \approx 1$ (Eq. 5), regardless of how the underlying parameters that produced it have moved.

Worked example. Suppose class "7" is first learned at Task 4 with concept vector $c_7^{(0)}$. Between Task 4 and Task 5, thousands of parameters in the encoder, Bayesian latent layer, and concept layer are updated to accommodate the new classes in Task 5. Parameter drift $\|\theta_5 - \theta_4\|_2$ is necessarily large and, by construction, monotonically non-decreasing as more tasks are learned (Section VB): every task changes some parameters. The question concept stability asks is different: *after* all of that parameter movement, does the network, when shown a new "7," still produce a concept vector $c_7^{(5)}$ that points in essentially the same direction as $c_7^{(0)}$? Our results (Section VB, Fig. 3) show these two quantities are empirically dissociated: parameter drift for the full method grows monotonically across all five tasks while KSI stays within $[0.986, 0.999]$, i.e., the concept for "7" is preserved almost perfectly even though none of the parameters that compute it are the same as they were at acquisition time.

Why concept stability is the quantity that carries more explanatory weight here. Parameter stability is neither necessary nor sufficient for what a practitioner or downstream system actually needs preserved, namely, the semantic identity of previously learned classes. It is not *necessary*: a network can move substantially in parameter space (e.g., via any transformation the task head can compensate for, such as a rotation absorbed by a later linear layer) without the class concept degrading, exactly the pattern in Fig. 3. It is not *sufficient*: constraining parameters, as EWC does, does not stop concept-level drift if the penalty is too weak to prevent early representation reorganization, or, as we find in Section VF, if the penalty is too slow to activate relative to a task-boundary gradient

shock; our EWC-baseline result (KSI $= -0.198$, Table III) is a direct empirical instance of a parameter-regularization method failing to protect the concept it was nominally trying to protect. Accuracy sits between these two failure modes and, per the discussion in the previous subsection, can mask both: a linear task head can remain correct through the compensation route described above (parameter movement without concept damage) and can also, per Davari et al. [15] and Filus et al. [23], appear to have forgotten while relevant structure is still linearly recoverable. Concept stability, as operationalized by KSI, is therefore the more direct proxy for the property we actually care about – whether the model's learned notion of a class has stayed the same – because it is measured at the representational level where that notion lives, rather than at the parameter level (too low-level, semantically uninterpretable on its own) or the output level (too coarse, compensable, and, as shown above, not always reflective of what the representation is doing).

D. What Concept Stability Adds to Continual Learning

Concept stability, as a target and as a measurement, contributes something accuracy-based evaluation cannot: it distinguishes *why* a method preserves accuracy from *whether* it does. Two methods that both preserve 94% cumulative accuracy (Table III: full method and EWC+replay) can be relying on structurally different mechanisms – one preserving the class's internal representation directly, the other preserving only the classifier's ability to route inputs to correct labels, potentially through a reorganized or compensating representation. This distinction matters for at least three practically-motivated reasons, consistent with the broader representation-forgetting literature [14]–[16], [23]: (i) representations that have been repeatedly reorganized may be less reusable for downstream tasks such as transfer learning or feature reuse, even if current-task accuracy looks unaffected; (ii) a semantically drifted concept is a plausible risk factor for brittle or unpredictable behavior under distribution shift or adversarial-style perturbation of the input, since the decision boundary may now rely on a coincidental, rather than principled, arrangement of features; and (iii) concept-level diagnostics offer a finer-grained failure signal during method development than a single scalar accuracy number, letting a practitioner see *which* classes are drifting and by how much, well before that drift becomes large enough to move the output-level accuracy metric.

E. Research Gap and Central Question

This gap motivates a specific empirical question, which we treat as the paper's central research question:

> *Can a continual-learning model preserve the internal representation of a previously learned concept even as its parameters continue adapting to new tasks?*

Bayesian Concept Consolidation (BCC) is proposed as one candidate answer, and the Knowledge Stability Index (KSI) is introduced as the measurement mechanism used to evaluate it.

F. Proposed Approach

We introduce Bayesian Concept Consolidation (BCC): a Bayesian concept bottleneck layer whose per-class representation is anchored, after first acquisition, by a concept-stability loss; a posterior-variance-derived adaptive weighting mechanism that modulates this loss; and an epistemic-uncertainty-prioritized replay buffer. We additionally propose KSI, a cosine-similarity-based metric that isolates concept-level drift from accuracy and parameter drift, enabling direct empirical comparison of methods on this axis.

G. Contributions

- We formulate concept-stability regularization as a loss operating on a Bayesian concept bottleneck, distinct from parameter-space regularization (Section III).
- We introduce the Knowledge Stability Index (KSI), a metric for concept-level representational drift, independent of task accuracy (Section IIIE).
- We propose a posterior-variance-derived adaptive weighting mechanism for the concept-stability loss, and report its exact realized form as implemented, including a discrepancy between its per-concept design intent and its scalar realization in the current implementation (Section IIIG, Section VI).
- We empirically demonstrate, via a controlled ablation against a diagonal-Fisher EWC baseline with and without replay, that concept-stability regularization – not replay – is responsible for concept preservation, while accuracy retention is statistically indistinguishable across replay-based conditions (Section V).
- We report an observed failure mode of diagonal EWC without replay under the evaluated class-incremental protocol, in which Fisher-penalty accumulation lags an early task-boundary gradient shock; we frame this as a protocol-specific observation rather than a general claim about EWC (Section VF).

H. Paper Organization

Section II positions this work relative to regularization-based, Bayesian, and replay-based continual learning. Section III formalizes the problem setting and describes the proposed method, including the training algorithm. Section IV describes the experimental setup. Section V reports results and discussion jointly. Section VI states limitations and threats to validity, and Section VII concludes.

II. Related Work

We organize this review by method family, and for each family state explicitly what it protects, its strongest supporting evidence, and its principal limitation with respect to concept-level stability – the axis this paper is concerned with. General surveys [18]–[20], [25] provide broader coverage of the field than is reproduced here; our focus is the specific dimension along which each family does or does not make claims about internal representations.

A. Regularization-Based Continual Learning

Regularization-based approaches constrain movement of parameters identified as important to previous tasks. EWC [1] estimates parameter importance via a diagonal Fisher Information approximation and penalizes squared deviation from the post-task parameter values, weighted by that importance; its strength is architectural simplicity and a single, interpretable penalty term, and it has been validated across supervised and reinforcement learning benchmarks. Its principal limitation, both generally documented and directly reproduced by our own EWC-baseline result (Section VF), is that the Fisher penalty is identically zero at the moment a new task boundary is registered and only accumulates strength as parameters subsequently move away from the anchor $\theta^{\star}$; if the task-boundary gradient shock is large relative to this accumulation rate, EWC can fail at the accuracy level well before its parameter-level protection engages. Synaptic Intelligence [2] addresses part of this by computing an online, per-parameter importance measure from the trajectory of updates during training rather than a post-hoc Fisher snapshot, giving it protection that begins accruing immediately rather than only after a task boundary; however, like EWC, it remains a purely parameter-space penalty and is agnostic to what a given parameter represents semantically – two networks can have identical Synaptic-Intelligence-style importance-weighted parameter trajectories while differing in whether their class concepts have rotated, because the penalty has no access to activation-space geometry. Architectural approaches such as Progressive Neural Networks [26], which allocate a new sub-network per task and reuse frozen previous columns via lateral connections, and PackNet [27], which iteratively prunes and freezes task-specific parameter subsets within a fixed-capacity network, sidestep the accumulation-timing failure mode entirely by making forgetting structurally impossible for frozen parameters – their limitation is instead a growing parameter or capacity cost with the number of tasks, and, more relevant to this paper, that "no parameter change" does not by itself constitute a measurement of concept stability; it is a sufficient but architecturally expensive condition for it, and neither method reports a representation-level stability metric.

B. Bayesian Neural Networks and Uncertainty Estimation

Bayes-by-Backprop [3] and related variational approaches treat network weights as distributions optimized via the reparameterization trick, providing calibrated epistemic uncertainty estimates; their strength is a principled, closed-form regularizer (the KL term against a prior) and posterior variance as a reusable signal for downstream uncertainty-aware decisions, which this paper exploits directly in Section III-G. Monte Carlo Dropout [4] offers a computationally cheaper approximate alternative with weaker theoretical guarantees on the resulting posterior. Variational Continual Learning (VCL) [5] extends variational Bayesian inference to the continual setting by treating the posterior after task t as the prior for task $t{+}1$, which is an elegant way to propagate uncertainty forward without EWC's reliance on a fixed post-hoc importance snapshot, and has shown strong empirical forgetting resistance on small-scale benchmarks; its limitation, as with the regularization-based methods above, is that the propagated posterior is a distribution *over weights*, not over class-level concepts, so VCL's protection is still, ultimately, parameter-space protection, applied more gracefully than EWC's fixed penalty but not evaluated against representational drift. Uncertainty-guided approaches such as UCB [6] use posterior variance to modulate per-parameter learning rate or regularization strength – a mechanism structurally related to, but applied at a different representational level than, the adaptive weighting introduced in this work (Section IIIG): UCB's variance signal governs how much a *parameter* may move, whereas our λ_2 governs how strongly a *concept vector* is anchored to its prototype, decoupling the two by design.

C. Replay and Knowledge Retention

Rehearsal-based methods store or generate a subset of previously seen data and interleave it with current-task training. iCaRL [10] pairs a bounded exemplar memory with a nearest-class-mean classifier and knowledge distillation, and remains a standard class-incremental baseline; its strength is directly counteracting the gradient updates that would otherwise overwrite prior-task decision boundaries, and it explicitly maintains class prototypes in feature space, which is conceptually close to, but not the same measurement as, our prototype-anchored concept-stability loss (Section IIIF) – iCaRL's prototypes are used for nearest-mean classification, not as an explicit stability regularization target evaluated by a dedicated drift metric. Experience Replay [12] shows that even simple reservoir-sampled rehearsal, inspired by the Complementary Learning Systems view of hippocampal replay, is a surprisingly strong and general-purpose baseline across task-incremental and class-incremental settings. Dark Experience Replay (DER) [13] extends this by additionally matching logits from earlier training via a reservoir of past network outputs, improving on plain rehearsal without task boundaries at inference time. Learning without Forgetting [9] instead distills the old model's output distribution on new-task data, avoiding a stored exemplar buffer altogether at the cost of relying entirely on knowledge encoded in old-model predictions on new inputs, which can drift for classes visually dissimilar from the new task's data. Across all four methods, the shared limitation relevant to this paper is the same one demonstrated empirically in Section V: replay (with or without distillation) directly counteracts output-level forgetting, and, as our EWC+replay condition shows (accuracy 93.2%, KSI 0.476, Table III), can restore accuracy to a level statistically indistinguishable from a concept-regularized method while leaving concept-level drift essentially unaddressed – replay was not designed to, and in our results does not, guarantee that the internal representation of a previously learned class remains stable; it guarantees only that classification accuracy on it is preserved. Semantic Drift Compensation [24] is a partial exception: it explicitly tracks and compensates for drift in class-prototype embeddings during class-incremental metric learning, which is closer in

spirit to concept-level protection than accuracy-only replay, though it operates in a non-Bayesian metric-learning setting without an uncertainty-derived regularization strength or a KSI-style bounded, per-class stability index reported over the full task trajectory.

D. Representation-Level and Concept-Level Analysis of Forgetting

A growing, largely diagnostic (rather than method-proposing) body of work measures forgetting directly at the representation level rather than through output accuracy. Kornblith et al. [14] introduced Centered Kernel Alignment (CKA) as a representation similarity index and showed that high functional (output) similarity between two networks does not imply high representational similarity, and vice versa – a direct precedent for the accuracy/concept dissociation this paper reports empirically in Section VC. Ramasesh et al. [16] applied CKA-style analysis specifically to continual learning and found forgetting concentrated in deeper layers, with earlier-task representational subspaces measurably eroded by later training. Davari et al. [15] proposed representation forgetting, measured via the drop in an optimal linear probe's accuracy on frozen features before and after a new task, and showed it can be small even when task-level (classifier) forgetting is large, i.e., that a model's representation can retain more task-relevant structure than its own classifier currently exposes; their finding that supervised contrastive representation learning combined with prototype-based querying is a strong representation-preserving baseline is conceptually adjacent to our prototype-anchored concept loss, though their metric is a probe-based accuracy measure rather than a bounded, per-class directional-drift index. Madaan et al. [17] extended representation-forgetting analysis to unsupervised continual learning, finding self-supervised representations to be comparatively more continuity-preserving than their supervised counterparts. Most directly relevant to the present paper, Filus et al. [23] recently proposed a diagnostic framework using sparse-autoencoder-derived concept proxies to decompose forgetting into apparent concept deletion, recoverability, and decodability, and found that a substantial share of seemingly lost concept-level information is not erased but rather rendered less *accessible* – concepts are, in their framing, often "hidden" rather than "lost." This body of work establishes, across several independent measurement approaches (linear probing, CKA, sparse-autoencoder concept proxies), that representation-level and concept-level dynamics are not fully captured by output accuracy; it is, however, predominantly diagnostic rather than prescriptive – these papers characterize representational drift after the fact using post-hoc analysis tools, rather than proposing a training-time regularizer, driven by the model's own uncertainty, that directly targets concept-level stability as an optimization objective. The concept-based interpretability literature that defines what a "concept" is in the first place – TCAV [22] and Concept Bottleneck Models [21] – was developed for post-hoc and intrinsic interpretability respectively, not for continual learning, and neither addresses temporal (across-task) stability of the concepts they define.

TABLE I
POSITIONING OF REPRESENTATIVE PRIOR WORK AGAINST THREE QUESTIONS RELEVANT TO CONCEPT-LEVEL STABILITY.

Method family	Protects concepts directly?	Measures concept drift?
EWC / SI [1], [2]	No (parameters)	No
VCL / UCB [5], [6]	No (weight posterior)	No
iCaRL / ER / DER / LwF [9], [10], [12], [13]	Indirectly (via accuracy)	No
Semantic Drift Comp. [24]	Partially (prototypes)	Partially
CKA / probing / SAE diagnostics [14]–[16], [23]	No (post-hoc only)	Yes (diagnostic)
This work (BCC + KSI)	**Yes**	**Yes**

E. Positioning of the Proposed Method

Table I summarizes the positioning argued above. Existing regularization methods (EWC [1], SI [2], and their architectural relatives [26], [27]) protect parameters without reference to what those parameters represent, and, in EWC's case, can fail at the accuracy level before parameter protection is even engaged (Section VF). Existing Bayesian continual learning methods (VCL [5], UCB [6]) propagate or exploit weight-level uncertainty but do not measure representation-level semantic drift. Existing replay and distillation methods (iCaRL [10], Experience Replay [12], DER [13], LwF [9]) preserve accuracy through rehearsal or distillation but, as our results show empirically, do not thereby guarantee representational stability; Semantic Drift Compensation [24] is a partial exception, targeting prototype drift directly but without a Bayesian, uncertainty-derived adaptive weighting mechanism. Existing representation-forgetting diagnostics [14]–[17], [23] establish that concept-level and output-level forgetting are dissociable but are predominantly analytical rather than corrective. This work occupies the resulting gap: a Bayesian, uncertainty-derived regularization mechanism operating directly on a concept-level representation, evaluated with a bounded, per-class, per-task stability metric (KSI) designed specifically to detect drift that accuracy-based (§IIA footprint via BWT/FWT [11]) and parameter-based metrics cannot.

III. PROBLEM FORMULATION AND BAYESIAN CONCEPT CONSOLIDATION

A. Continual Learning Setting

We consider a sequence of T tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T$, each associated with a dataset $\mathcal{D}_t = \{(x_i, y_i)\}$ and a disjoint set of classes C_t, such that $C_i \cap C_j = \emptyset$ for $i \neq j$ (class-incremental setting). At task t, the model has access only to $\mathcal{D}_t$ and a bounded replay buffer $\mathcal{M}$ populated from prior tasks; it is evaluated on the cumulative test set $\bigcup_{i=1}^{t} \mathcal{D}_i^{\text{test}}$.

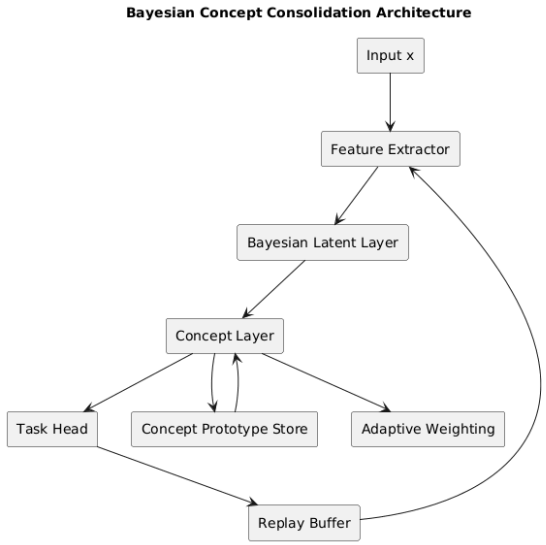


Fig. 1. Overall framework: input → feature extractor → Bayesian latent layer → concept layer → task head, with the concept prototype store, adaptive weighting module, and uncertainty-prioritized replay buffer operating alongside the forward path.

B. Model Architecture

Let $x \in \mathbb{R}^d$ denote an input. The model (Fig. 1) is composed of:

$$h = f_{\text{enc}}(x) \in \mathbb{R}^{128} \quad \text{(feature extractor)} \tag{1}$$

$$z = \text{ReLU}\big(f_{\text{bayes}}(h)\big) \in \mathbb{R}^{128} \quad \text{(Bayesian latent)} \tag{2}$$

$$c = f_{\text{concept}}(z) \in \mathbb{R}^{32} \quad \text{(concept vector)} \tag{3}$$

$$\hat{y} = f_{\text{head}}(c) \in \mathbb{R}^{|C|} \quad \text{(task head logits)} \tag{4}$$

where f_{bayes} and f_{concept} are Bayesian linear layers (Section IIIC) and f_{head} is a deterministic linear classifier over the cumulative class set observed so far.

C. Bayesian Parameterization

Each Bayesian linear layer parameterizes a Gaussian posterior over its weights, $w \sim \mathcal{N}(\mu, \sigma^2)$ with $\sigma = \text{softplus}(\rho)$, using the reparameterization trick $w = \mu + \sigma \odot \epsilon,\ \epsilon \sim \mathcal{N}(0, I)$ for differentiable sampling. The KL divergence against a zero-mean Gaussian prior $\mathcal{N}(0, \sigma_p^2)$ is computed in closed form per layer and summed:

$$\mathcal{L}_{\text{KL}} = \sum_{\ell \in \{\text{bayes,concept}\}} \sum_i \left[\ln \frac{\sigma_p}{\sigma_i} + \frac{\sigma_i^2 + \mu_i^2}{2\sigma_p^2} - \frac{1}{2} \right]. \tag{5}$$

The role of this Bayesian parameterization in the proposed method follows a specific causal chain, made explicit here rather than left implicit: *posterior variance* of the concept layer's weights is treated as an *uncertainty estimate* over how settled the model's current concept representation is; this uncertainty estimate is converted into an *adaptive regularization strength* (Section IIIG); and that regularization strength directly modulates the *concept-stability loss* (Section IIIF), which is the mechanism ultimately responsible for concept preservation (Section V). Bayesian modeling is used here not for its predictive-uncertainty output per se, but as the source signal for this regularization-strength chain.

D. Concept Representation and Prototype Memory

The concept vector $c \in \mathbb{R}^{32}$ (Eq. 3) is the output of the Bayesian concept layer. A *Concept Prototype Store* maintains, for each class c_k, a frozen prototype $c_k^{(0)}$ set once – as the mean concept vector over that class's training data – at the task in which the class is first observed. This prototype is not updated by subsequent gradient steps; it serves solely as the anchor for the concept-stability loss and the reference point for KSI.

E. Knowledge Stability Index

For each class c_k, let $c_k^{(0)}$ denote its prototype (Section IIID) and $c_k^{(t)}$ its concept vector re-evaluated at a later task boundary t using stored exemplars. Concept stability for class c_k at task t is the cosine similarity

$$s_k(t) = \frac{c_k^{(0)} \cdot c_k^{(t)}}{\|c_k^{(0)}\| \, \|c_k^{(t)}\|}. \tag{6}$$

The Knowledge Stability Index at task t is the mean over all classes with a recorded re-evaluation:

$$\text{KSI}(t) = \frac{1}{|K_t|} \sum_{k \in K_t} s_k(t), \tag{7}$$

where K_t is the set of classes with both an original prototype and at least one later re-evaluation by task t.

Why cosine similarity. Cosine similarity measures the change in *direction* of a class's concept vector, independent of its scale. We consider direction the more meaningful quantity for representational identity here because the task head (Eq. 4) is a linear classifier operating on c: a rotation of c_k relative to other classes' concept vectors changes the decision geometry the network relies on for class k, whereas a pure rescaling of c_k does not, provided the head's weights can compensate. Cosine similarity isolates the former.

What KSI captures and does not capture. KSI captures directional drift of a class's mean concept vector between its acquisition point and a later evaluation point. It does *not* capture: (i) drift in vector magnitude, since cosine similarity is scale-invariant; (ii) intra-class dispersion, since $c_k^{(0)}$ and $c_k^{(t)}$ are both class means, not per-sample statistics; (iii) trajectory shape between the two measured points, since only original-vs.-latest is compared by the summary statistic (per-task trajectories, Section VB, partially address this by reporting KSI at every task boundary rather than only the endpoint).

Range and aggregation. By construction, $s_k(t) \in [-1, 1]$, so $\text{KSI}(t)$, as an unweighted mean of bounded quantities, is likewise bounded in $[-1, 1]$; no additional normalization or transformation is applied. KSI $= 1$ indicates no directional drift; KSI $\to -1$ indicates the concept vector has rotated toward the opposite direction from its original orientation. Aggregation across classes is an unweighted arithmetic mean, giving each class with a recorded re-evaluation equal influence

TABLE II
CONCEPTUAL COMPARISON OF EVALUATION QUANTITIES.

Quantity	Measures	Does not measure
Accuracy	Task prediction correctness	Internal representation
Parameter drift	Weight change ($\|\theta_t - \theta_0\|_2$)	Concept preservation
KSI	Concept-vector directional stability	Magnitude / intra-class dispersion

regardless of sample count.Because a single severely drifted class can be masked by an otherwise-high mean across the remaining classes, we additionally report the per-class median, minimum, and standard deviation of $s_k(t)$ alongside the mean wherever KSI is reported (Section VB).

Relation to accuracy and parameter drift. Table II summarizes the distinction motivating KSI's introduction as a separate metric.

F. Concept-Stability Regularization

The concept-stability loss penalizes the cosine distance between a sample's current concept vector and its class's frozen prototype, for classes with an existing prototype:

$$\mathcal{L}_{\text{concept}} = \frac{1}{|B'|} \sum_{i \in B'} \left(1 - \frac{c_i \cdot c_{y_i}^{(0)}}{\|c_i\| \, \|c_{y_i}^{(0)}\|} \right), \tag{8}$$

where B' is the subset of the current batch (or replay batch) whose label y_i already has a stored prototype. Because task boundaries in the class-incremental setting introduce only previously unseen classes, $\mathcal{L}_{\text{concept}}$ evaluated on the current task's own batch is identically zero for that task; the loss receives gradient signal for a given class only when that class's data reappears via replay in a later task (Section IIIH). $\mathcal{L}_{\text{concept}}$ in Eq. 12 is therefore computed as the sum of the current-batch and replay-batch terms.

G. Uncertainty-Adaptive Weighting

The design intent of the adaptive weighting mechanism is a per-concept-dimension coefficient derived from that dimension's posterior variance, $\lambda_{2,i} = 1/(\sigma_i^2 + \epsilon)$, so that concept dimensions the model is confident about (low variance) are protected more strongly than dimensions it remains uncertain about. Although this formulation motivates dimension-specific uncertainty weighting, the current implementation aggregates posterior variance across the concept dimensions, yielding a global scalar coefficient rather than a per-dimension one:

$$\lambda_2 = \text{clip}\left(\frac{1}{32} \sum_{i=1}^{32} \frac{1}{\sigma_i^2 + \epsilon},\ 0,\ \lambda_{\max} \right), \tag{9}$$

with σ_i^2 the concept layer's per-output-unit weight variance (mean over the input dimension), $\epsilon = 10^{-3}$, and $\lambda_{\max} = 10$. Consequently, the experiments in Section V evaluate uncertainty-adaptive concept regularization at the global rather than per-concept-dimensional level; we report this explicitly rather than presenting the mechanism as fully per-concept in its current form (Section VI). The upper clip was found necessary empirically: without it, near-zero initial posterior variance produces $\lambda_2 \sim 10^3$, which dominates the task loss and prevents new-task learning.

H. Replay Mechanism

An *Uncertainty Replay Buffer* stores, per class, up to M exemplars selected by highest Monte Carlo predictive entropy at the end of the task in which the class was learned:

$$u(x) = -\sum_j \bar{p}_j(x) \log \bar{p}_j(x), \quad \bar{p}(x) = \frac{1}{S} \sum_{s=1}^{S} \text{softmax}\big(f_\theta^{(s)}(x)\big), \tag{10}$$

with S stochastic forward passes. Replay samples are drawn uniformly from the buffer at each training step and contribute an uncertainty-weighted cross-entropy term,

$$\mathcal{L}_{\text{replay}} = \frac{1}{|B_r|} \sum_{i \in B_r} \tilde{u}_i \cdot \ell_{\text{CE}}(x_i, y_i), \quad \tilde{u}_i = \frac{u(x_i)}{\sum_j u(x_j)} |B_r|, \tag{11}$$

prioritizing samples the model is currently least confident about. Buffer population is performed for every experimental condition regardless of whether replay is used in the training loss, so that KSI's later re-evaluation of old-class concept vectors is available uniformly across conditions.

I. Overall Objective, Optimization, and Algorithm

Combining the above, the complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_1 \mathcal{L}_{\text{KL}} + \lambda_2 \mathcal{L}_{\text{concept}} + \lambda_3 \mathcal{L}_{\text{replay}}, \tag{12}$$

with $\lambda_1 = 10^{-4}$ fixed, λ_2 as defined in Section III-G (or a fixed constant of 1.0 in the *fixed-weight* ablation condition), and $\lambda_3 = 1.0$. Only terms present in the released implementation are included. All parameters (deterministic and variational) are optimized jointly with Adam (lr $= 10^{-3}$). For the EWC baseline condition (Section IVB), gradients are additionally clipped to a maximum global norm of 5.0 as a numerical stability safeguard; this clipping is not required by, and does not materially affect, the proposed method's default configuration.

Algorithm 1 summarizes the full training procedure.

IV. EXPERIMENTAL SETUP

A. Datasets and Task Construction

We use Split-MNIST: the standard MNIST handwritten digit dataset (60,000 training / 10,000 test images, 28×28 grayscale, 10 classes), partitioned into $T = 5$ class-incremental tasks of 2 disjoint classes each: $\{0,1\}, \{2,3\}, \{4,5\}, \{6,7\}, \{8,9\}$. No additional preprocessing beyond tensor conversion (pixel values in $[0,1]$) is applied.

B. Baselines

We compare five conditions, all sharing the identical backbone architecture and differing only in the loss terms and replay configuration applied:

- **Full method**: adaptive concept-stability loss (Section III-G) with replay.

Algorithm 1 Bayesian Concept Consolidation training

Initialize model f_θ, optimizer, prototype store $P \leftarrow \emptyset$, replay buffer $\mathcal{M} \leftarrow \emptyset$
for task $t = 1, \ldots, T$ **do**
  for epoch $= 1, \ldots, E$ **do**
    for minibatch $(x, y) \sim \mathcal{D}_t$ **do**
      Sample replay batch $(x_r, y_r) \sim \mathcal{M}$ (if $\mathcal{M} \neq \emptyset$)
      $z \leftarrow \text{BayesLatent}(f_{\text{enc}}(x))$; $c \leftarrow \text{ConceptLayer}(z)$
      $\hat{y} \leftarrow f_{\text{head}}(c)$
      Compute $\mathcal{L}_{\text{task}}$, $\mathcal{L}_{\text{KL}}$
      Compute $\mathcal{L}_{\text{concept}}$ on (c, y) and on replay batch via P
      Compute λ_2 from concept-layer posterior variance (Sec. III-G)
      Compute $\mathcal{L}_{\text{replay}}$ via uncertainty-weighted CE on replay batch
      $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}} + \lambda_1 \mathcal{L}_{\text{KL}} + \lambda_2 \mathcal{L}_{\text{concept}} + \lambda_3 \mathcal{L}_{\text{replay}}$
      Backpropagate and update θ
    end for
  end for
  Prototype registration: for each new class $k \in C_t$, set $c_k^{(0)} \leftarrow$ mean concept vector over $\mathcal{D}_t$'s samples of class k; add to P
  Cross-task KSI check: for each previously seen class $k \notin C_t$ with buffered exemplars, re-embed and log $s_k(t)$
  Buffer update: select top-M highest-uncertainty exemplars per class in C_t (Eq. 9) and add to $\mathcal{M}$
  Evaluate cumulative test accuracy on $\bigcup_{i \leq t} \mathcal{D}_i^{\text{test}}$; compute KSI$(t)$
end for

- **No concept loss**: replay only, $\mathcal{L}_{\text{concept}}$ disabled ($\lambda_2 \equiv 0$); isolates the contribution of concept-stability regularization against replay alone.
- **Fixed-weight**: concept-stability loss with $\lambda_2 \equiv 1$ (adaptive weighting disabled); isolates the contribution of adaptive weighting against a fixed-coefficient version of the same loss.
- **EWC baseline**: diagonal-Fisher EWC applied to all model parameters, $\lambda_{\text{EWC}} = 1000$, no replay, no concept loss; a literature baseline representing parameter-space regularization without our proposed mechanism.
- **EWC + replay**: identical to the EWC baseline with replay additionally enabled; isolates whether EWC's observed failure mode (Section VF) is attributable to the absence of replay, and whether replay alone (independent of concept-stability regularization) is sufficient for concept preservation.

The Fisher Information matrix for the EWC conditions is estimated via the empirical diagonal approximation, sampled from the model's own predictive distribution over 5 minibatches at each task boundary, and accumulated additively across tasks.

TABLE III
MAIN RESULTS: FINAL ACCURACY AND KSI, MEAN ± STD OVER $n = 3$ SEEDS.

Condition	Accuracy	KSI
Full method	0.9427 ± 0.0087	$\mathbf{0.9966 \pm 0.0004}$
No concept loss	0.9435 ± 0.0057	0.4674 ± 0.0686
Fixed-weight	0.9470 ± 0.0019	0.9294 ± 0.0160
EWC baseline	0.1955 ± 0.0011	-0.1975 ± 0.0113
EWC + replay	0.9317 ± 0.0068	0.4759 ± 0.0499

C. Implementation Details

The model is implemented in PyTorch. The feature extractor is a two-layer CNN (32 and 64 channels, 3×3 kernels, ReLU, 2×2 max-pooling) followed by a linear projection to a 128-dimensional feature vector; the Bayesian latent layer and concept layer are Bayes-by-Backprop linear layers (128→128 and 128→32 respectively); the task head is a deterministic linear layer over the cumulative class count. Replay buffer capacity: 50 exemplars per class. Monte Carlo samples for predictive uncertainty: 5 (replay weighting) / 10 (evaluation-time uncertainty, where used). All experiments were run on CPU. Full source code is available at https://github.com/simachewalamneh/BayesianConceptCL.

D. Evaluation Metrics

Accuracy: mean classification accuracy over the cumulative test set $\bigcup_{i \leq t} \mathcal{D}_i^{\text{test}}$, reported after the final task. **KSI**: as defined in Section III-E, reported after the final task and as a per-task trajectory. **Parameter drift**: $\|\theta_t - \theta_0\|_2$, reported for descriptive purposes only (Section VB). We did not evaluate additional continual-learning metrics (BWT, FWT, per-task forgetting) or calibration metrics (ECE, NLL) in the present study (Section VI).

E. Experimental Protocol

Optimizer: Adam, learning rate 10^{-3}, batch size 128, 1 epoch per task in the main reported results. All metrics are averaged over $n = 3$ random seeds $\{0, 1, 2\}$, with standard deviation reported. A partial check at 3 epochs per task was run on a subset of seeds and did not indicate a qualitatively different pattern, but was not completed across all conditions and is not included in the main results (Section VI).

V. RESULTS AND DISCUSSION

A. Overall Performance

Table III reports final cumulative test accuracy and KSI across all five conditions. The three replay-enabled, non-EWC conditions (full method, no concept loss, fixed-weight) achieve statistically indistinguishable accuracy, ranging 94.27%–94.70% with overlapping standard deviations. EWC+replay achieves comparable accuracy ($93.17\% \pm 0.68\%$), while EWC without replay collapses to $19.55\% \pm 0.11\%$, near the level expected from retaining only the most recently trained task in a 5-task, 10-class cumulative evaluation.

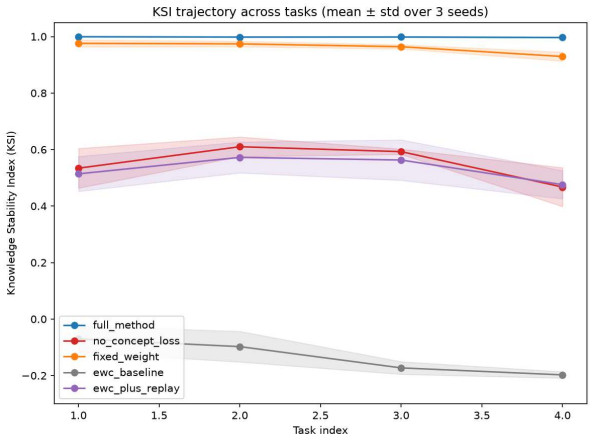


Fig. 2. KSI trajectory across tasks, mean ± std over 3 seeds, for all five conditions. The full method remains flat near 1.0; fixed-weight declines gradually; no-concept-loss and EWC+replay overlap closely throughout; EWC-without-replay drops sharply after the first task transition into negative territory.

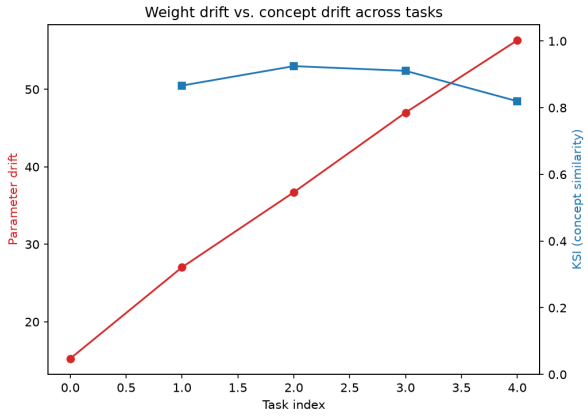


Fig. 3. Illustrative single-run comparison of parameter-space drift ($\|\theta_t - \theta_0\|_2$) against KSI across tasks, for the full method. Parameter drift increases monotonically across tasks while KSI remains near 1.0.

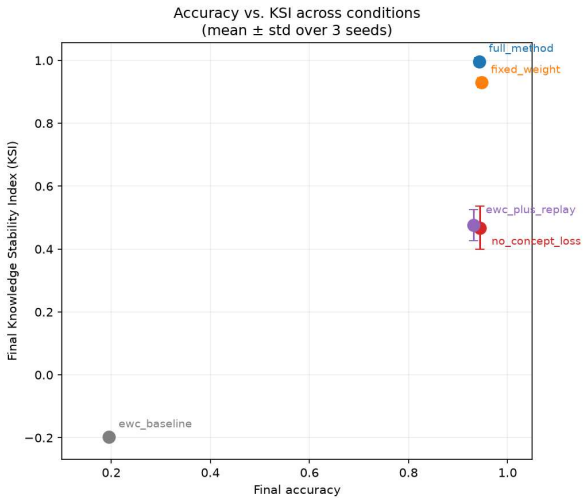


Fig. 4. Final accuracy vs. final KSI for all five conditions. The four replay-enabled conditions cluster within a narrow accuracy band (∼93–95%) while spanning KSI from ∼0.47 to ∼0.997, visually demonstrating that prediction accuracy and concept stability are not equivalent quantities under the evaluated protocol.

B. Concept Stability

Fig. 2 shows the full per-task KSI trajectory. The no-concept-loss and EWC+replay conditions' KSI trajectories overlap closely at every task index (bands substantially intersecting throughout), despite representing different underlying mechanisms (no regularization vs. Fisher-based regularization), indicating that replay – not the presence or absence of a Fisher penalty – determines the trajectory shape when concept-stability regularization is absent. Both of these conditions exhibit a non-monotonic pattern, rising from task index 1 to task indices 2–3 before declining by task index 4; we report this descriptively and did not investigate it further (Section VI).

Per-class KSI can hide localized drift behind a healthy mean. The averaged KSI reported above and in Table III can mask class-specific variation. In a representative single run of the full method, per-class similarity at the final task boundary ranged from 0.987 (class 5) to above 0.999 for the least-drifted classes, against a cohort mean of 0.9965 – a roughly $9\times$ larger deviation from perfect stability for the worst class than the mean alone suggests. This did not indicate a failure in this run (0.987 remains high), but it demonstrates that the mean KSI, taken alone, can understate the degree to which any single concept has drifted; we report per-class mean, median, minimum, and standard deviation (not only the aggregate mean) for this reason (Section III-E).

For the full method, parameter drift increases monotonically across all five tasks (from a representative single run: $15.4 \rightarrow 27.0 \rightarrow 37.0 \rightarrow 47.5 \rightarrow 57.2$ in ℓ_2 norm) while KSI remains within $[0.986, 0.999]$ over the same interval (Fig. 3). This dissociation – substantial, continued parameter-space movement co-occurring with near-total concept-space stability – indicates that continued parameter-space adaptation, as required to learn new classes, does not necessarily disturb the semantic content associated with previously learned classes when a concept-level anchor is maintained.

C. Accuracy–Concept Stability Dissociation

This is, in our view, the central empirical result of the paper, and Fig. 4 is intended as its primary supporting figure: three conditions with statistically indistinguishable accuracy (full method, no-concept-loss, fixed-weight) span KSI values from 0.467 to 0.997, and a fourth (EWC+replay) matches the full method's accuracy within 1.1 percentage points while differing in KSI by more than a factor of two. Accuracy alone, as conventionally reported in continual learning evaluation, would not have distinguished these conditions; only KSI does.

D. Ablation Study

Comparing the three non-EWC conditions in Table III isolates two components. Removing the concept-stability loss entirely (no-concept-loss vs. full method) reduces KSI from

0.997 to 0.467 at statistically indistinguishable accuracy, indicating the loss term is responsible for concept preservation rather than for the small accuracy differences observed (which are within one another's standard deviation). Disabling adaptive weighting in favor of a fixed coefficient (fixed-weight vs. full method) yields an intermediate KSI (0.929), indicating adaptive weighting contributes additional concept preservation beyond a fixed-coefficient version of the same loss.

E. Effect of Adaptive Bayesian Weighting

Of the total KSI range spanned by the ablation ($0.467 \rightarrow 0.997$), the majority ($0.467 \rightarrow 0.929$) is attributable to the presence of the concept-stability loss itself, with adaptive weighting responsible for the remaining improvement ($0.929 \rightarrow 0.997$) over a fixed-coefficient version of the same loss. As noted in Section III-G, this adaptive coefficient is, in the current implementation, a global scalar rather than a per-concept-dimension value; the improvement attributed to adaptive weighting here should accordingly be read as the effect of a global posterior-variance-derived coefficient, not of dimension-specific weighting.

F. Comparison with EWC and Replay

Instrumented per-batch logging at the first task transition (Task $0 \rightarrow$ Task 1) shows task loss spiking to ~ 35 (versus an expected $\sim \ln 2 \approx 0.69$ for a freshly initialized 2-class head) with an associated pre-clip gradient norm of ~ 124, while the EWC penalty remains $< 10^{-3}$ for the first several batches, since the penalty term $\mathcal{F}_i(\theta_i - \theta_i^\star)^2$ is identically zero at the moment of task-boundary registration and grows only as parameters subsequently move away from $\theta^\star$. This ordering was observed to persist across a four-order-of-magnitude sweep of $\lambda_{\text{EWC}} \in \{1, 10, 100, 1000\}$, with Task 0 accuracy remaining at 0.0 in every case tested. We report this as an observed failure mode under the evaluated experimental protocol – Split-MNIST's abrupt class-incremental task boundaries, one epoch per task, and the specific diagonal-Fisher estimator and hyperparameter range tested here – rather than as a general property of EWC; a different task-boundary structure, per-task (rather than accumulated) EWC penalties, or a warm-start schedule may not exhibit the same timing mismatch, and we did not test these variants.

G. Sensitivity and Computational Analysis

We did not systematically measure wall-clock time, memory usage, or sensitivity to concept dimensionality (d_c) in the present study. The proposed method requires maintaining a concept prototype store and a replay buffer, incurring memory overhead proportional to the number of classes and buffer capacity; this overhead is shared with, and no larger than, the replay-based baselines evaluated here. Systematic measurement of these quantities, and of sensitivity to $d_c \in \{16, 32, 64, 128\}$, is left to future work (Section VI).

VI. Limitations and Threats to Validity

- **Single dataset**: all results are on Split-MNIST. Generalization to higher-resolution or more visually complex benchmarks (e.g., Split-CIFAR10) has not been evaluated; this is the single most valuable extension for establishing external validity.
- **Task count and duration**: five tasks, one epoch per task, in the reported main results; a partial check at three epochs per task on a subset of seeds did not indicate a qualitatively different pattern but was not completed across all conditions.
- **Seed count**: $n = 3$ random seeds per condition. Standard deviations are consequently coarse estimates; we did not perform significance testing beyond reporting mean $\pm$ std, and a larger seed count (e.g., 5) would strengthen the reported comparisons.
- **Adaptive weighting granularity**: as implemented, the per-concept-dimension posterior-variance weighting (Section III-G) is averaged into a single scalar before being applied to the loss, rather than applied per concept dimension.
- **KSI is direction-only**: being cosine-based, KSI does not detect drift in concept vector magnitude, and reports a class-mean statistic rather than a per-sample or intra-class-variance statistic (Section III-E).
- **EWC hyperparameter and protocol coverage**: the EWC baseline was swept over $\lambda_{\text{EWC}} \in \{1, 10, 100, 1000\}$ under a single task-boundary protocol; alternative Fisher estimators, per-task (non-accumulated) penalty variants, or different task-boundary structures were not evaluated, and the observed failure mode (Section VF) should not be generalized beyond the tested protocol.
- **Unexplained non-monotonicity**: the rise-then-fall KSI pattern observed for replay-only conditions (Section VB) was noted but not diagnosed.
- **Per-class KSI statistics from a single run**: the per-class mean/median/min/std breakdown (Section VB) is reported from one representative run of the full method, not aggregated across all 3 seeds or all 5 conditions; whether the same class-specific pattern (e.g., which class shows the largest deviation from the mean) is consistent across seeds and conditions was not systematically checked.
- **No calibration or standard CL metrics**: backward transfer, forward transfer, per-task forgetting, expected calibration error, and negative log-likelihood were not evaluated.
- **No computational cost or dimensionality-sensitivity measurements**: wall-clock time, memory usage, and sensitivity to concept dimensionality d_c were not systematically measured (Section VG).
- **Single architecture**: all conditions share one fixed backbone; sensitivity to architecture choice (depth, width) beyond concept dimensionality was not studied.

VII. Conclusion

We introduced Bayesian Concept Consolidation, a continual learning framework that regularizes a Bayesian concept-space representation directly, together with the Knowledge Stability Index, a metric that isolates concept-level representational drift from task accuracy and parameter drift. On Split-MNIST, a controlled ablation shows that concept-stability regularization improves concept preservation substantially (KSI $0.467 \rightarrow 0.997$) at statistically indistinguishable task accuracy, and that this improvement is not attributable to replay alone: an EWC-plus-replay baseline matching the proposed method's accuracy achieves less than half its KSI. Under the evaluated protocol, a diagonal-Fisher EWC baseline without replay was found to fail at the accuracy level due to a diagnosed timing mismatch between an early task-boundary gradient shock and the gradual accumulation of the Fisher penalty; we report this as an observation specific to the tested protocol rather than a general claim about EWC. These results support the conclusion that, under the studied conditions, task accuracy and semantic concept stability are dissociable, and that concept-level preservation benefits from a mechanism operating directly on the concept representation rather than on parameters or on classification decisions alone.

The limitations enumerated in Section VI directly suggest next steps: extending evaluation to Split-CIFAR10 and standard continual-learning metrics (BWT, FWT, forgetting), increasing seed count, implementing a genuinely per-concept-dimension realization of the adaptive weighting mechanism, diagnosing the observed non-monotonic KSI trajectory via per-class rather than class-averaged logging, and systematically measuring computational cost and sensitivity to concept dimensionality. We view the present results as establishing the core empirical distinction between accuracy retention and concept-level stability, with these extensions as the natural path toward a more broadly validated claim.

Funding Declaration

This research received no external funding.

References

[1] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[2] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. 34th Int. Conf. Machine Learning (ICML)*, 2017.

[3] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. 32nd Int. Conf. Machine Learning (ICML)*, 2015.

[4] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Machine Learning (ICML)*, 2016.

[5] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *Int. Conf. Learning Representations (ICLR)*, 2018.

[6] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, "Uncertainty-guided continual learning with Bayesian neural networks," in *Int. Conf. Learning Representations (ICLR)*, 2020.

[7] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24, Elsevier, 1989, pp. 109–165.

[8] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," in *Int. Conf. Learning Representations (ICLR)*, 2014.

[9] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[10] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010.

[11] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6467–6476.

[12] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[13] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: A strong, simple baseline," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 15920–15930.

[14] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, 2019.

[15] M. Davari, N. Asadi, S. Mudur, R. Aljundi, and E. Belilovsky, "Probing representation forgetting in supervised and unsupervised continual learning," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16712–16721.

[16] V. V. Ramasesh, E. Dyer, and M. Raghu, "Anatomy of catastrophic forgetting: Hidden representations and task semantics," in *Int. Conf. Learning Representations (ICLR)*, 2021.

[17] D. Madaan, J. Yoon, Y. Li, Y. Liu, and S. J. Hwang, "Representational continuity for unsupervised continual learning," in *Int. Conf. Learning Representations (ICLR)*, 2022.

[18] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5362–5383, 2024.

[19] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[20] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[21] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *Proc. 37th Int. Conf. Machine Learning (ICML)*, 2020, pp. 5338–5348.

[22] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, 2018, pp. 2668–2677.

[23] K. Filus, K. Faber, R. Corizzo, and C. Kanan, "Lost or hidden? A concept-level forgetting in supervised continual learning," *arXiv preprint arXiv:2605.16374*, 2026.

[24] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. van de Weijer, "Semantic drift compensation for class-incremental learning," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5513–5533, 2022.

[26] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[27] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7765–7773.

Note: citation details above (venue, year, page numbers) should be independently verified against the original publications before submission.