Hierarchical Bayesian Causal Modular Learning: A Two-Level Columnar Architecture for Continual Learning

Ben Goertzel[1,2], Charlie Derr[1], Yohannes Taye[1,2]

[1] SingularityNET Foundation
ben@singularitynet.io
[2] iCog Labs

Abstract. We describe a brain-inspired columnar neural architecture for continual learning, called Columnar Bayesian Causal Coding (ColBaC), together with a general theoretical framework that subsumes it, called Hierarchical Bayesian Causal Modular Learning (HBCML). The central idea is to decompose continual learning into two coupled probabilistic processes operating at different structural scales: a top-level controller selecting a sparse subset of structurally restricted component learners (columns) for each context, and an internal probabilistic process inside each column distinguishing reusable causal structure from task-local residue. We motivate the columnar inductive bias via the actual diversity of cortical columnar microstructure across brain regions — from rigid V1-like orientation columns through the discrete barrel cortex through the more flexible recruitment patterns of prefrontal cortex — and argue that a single architectural template with adjustable rigidity (a protected hard kernel surrounded by adaptable shells, plus typed microcolumns) can accommodate this range. We summarize the HBCML theory and its main theorems — architectural adequacy implying causal modularity, exact one-swap teacher monotonicity, and a Rao–Blackwell guarantee for internal certificates — then present the concrete ColBaC MNIST architecture and report preliminary experimental findings, including a dense offline selector audit showing that the current learned controller is meaningfully suboptimal but in a way the architecture itself is well positioned to repair. We close with a scaling protocol for Split-CIFAR and a more speculative outline of how the framework extends to reinforcement learning and to transformer-style architectures.

Keywords: Continual learning · Columnar networks · Causal coding · Bayesian modularity · Catastrophic forgetting · Brain-inspired architectures · AGI.

1 Introduction

Continual learning — the ability to acquire new skills and concepts over time without overwriting previously acquired ones — is one of the most stubbornly unsolved problems in modern machine learning, and one of the most clearly

necessary capabilities for anything that aspires to general intelligence. A system that forgets task A as soon as it begins learning task B is not a general learner but at best a sequence of narrow-learner invocations glued together into an application architecture. Yet catastrophic forgetting [1,2] has proved remarkably persistent: standard end-to-end training, which works so well in single-task or jointly trained settings, fails badly when tasks are presented sequentially without rehearsal.

The cleanest theoretical handle on this problem comes from the observation that catastrophic forgetting is fundamentally an interference problem. When the parameter updates induced by different tasks are not approximately commuting — when learning task B moves the same parameters that encoded task A, in roughly the same directions — forgetting follows almost mechanically. The general causal-continual-learning (CCL) theorem [3] makes this precise: if a learner's parameters can be modularized so that each task's update field acts mainly on a small task-specific support, with small gradients elsewhere and small cross-Hessian coupling, then the pairwise commutators of update fields are correspondingly small, and sequential learning approximates joint learning up to a controllable error.

The CCL theorem is structural, not constructive. It says when continual learning works; it does not say how to build a learner satisfying its hypotheses. The natural next question is: what kind of architecture makes those hypotheses easy to satisfy, rather than something one merely hopes a generic dense network will discover by itself? There are going to be many answers to this question, and this paper proposes *one*, in two parts. The general part, called *Hierarchical Bayesian Causal Modular Learning* (HBCML), defines an architecture class that, by construction, is well positioned to satisfy the CCL hypotheses. The concrete part, called *Columnar Bayesian Causal Coding* (ColBaC), is a specific instance that we have been developing experimentally, motivated by the columnar organization of biological cortex.

Among the closest neighbors in the literature are progressive networks [13], which add new columns for new tasks but without internal probabilistic structure or a sparse selection mechanism, and modern sparse mixture-of-experts [10,11], which have learned routing among many experts but without explicit continual-learning mechanisms or internal certificates. Elastic Weight Consolidation [14] and related regularization-based approaches address forgetting at the parameter level but without architectural modularity. Capsule networks [12] introduce a different kind of grouped representation but again without the two-level Bayesian decomposition. The predictive coding tradition [16,17] provides much of the local-update philosophy underlying causal coding but does not on its own supply continual-learning guarantees. The general theory of general intelligence [22] and TransWeave [21] provide the broader operator-theoretic context in which the present architecture sits. What distinguishes the framework presented here is the explicit two-level Bayesian decomposition: not just sparse routing among modules, but probabilistic state inside each module that informs the routing,

plus structural restrictions inside each module specifically designed to preserve causal modularity through sequential learning.

The organization of the paper is as follows. Section 2 reviews columnar organization in the brain and in machine learning, and argues that the diversity of cortical columnar microstructure across regions is itself an important design constraint. Section 3 presents the HBCML theory concisely: the architecture class, the adequacy assumptions linking it to the CCL theorem, and three of its main results. Section 4 describes the ColBaC architecture as concretely instantiated for Split-MNIST. Section 5 reports preliminary experimental findings, including a dense offline selector audit with concrete numbers. Section 6 describes the protocol for scaling to Split-CIFAR. Section 7 sketches extensions to reinforcement learning and transformers. Section 8 concludes.

2 Columnar Organization in Brains and Machines

The hypothesis that cortex is organized into columns goes back to Mountcastle's recordings in cat somatosensory cortex [4], which found that vertical penetrations encountered neurons sharing modality and receptive-field properties, while horizontal penetrations crossed sharply into different ones. Mountcastle proposed the cortical column as a basic computational unit. Hubel and Wiesel's later work on cat and monkey V1 [5,6] produced what is still the textbook image of cortical columnar organization: orientation columns within ocular dominance bands, with smooth tangential maps and abrupt vertical coherence.

It is important for the design of brain-inspired architectures that this picture, though real, is far from uniform across cortex. The differences are themselves informative.

V1: highly stereotyped. Primary visual cortex has narrow functional differentiation axes — orientation, ocular dominance, spatial frequency, retinal position — and the columnar organization is correspondingly rigid. Neighboring columns differ in only a few well-understood ways, and the same columnar template is replicated across the entire surface of V1.

Barrel cortex: discretely modular. In rodent primary somatosensory cortex, each facial vibrissa projects to a distinct, cytoarchitecturally identifiable barrel [7]. This is arguably the cleanest example of one-to-one mechanism-to-module mapping in any cortical area: there is a definite barrel for each whisker, and the structural correspondence is essentially perfect.

Prefrontal and higher association cortex: flexibly recruited. Higher-order areas retain minicolumnar architecture but the functional columns are recruited far more flexibly, often combinatorially across tasks [8]. The same physical column may participate in different functional roles in different contexts, and the columnar boundaries are correspondingly less crisp.

Hippocampus and cerebellum: not columnar at all. Hippocampus follows a lamellar rather than columnar organization, and the cerebellum is exquisitely stereotyped but in microzones rather than columns. So columnar organization is one option in the brain's repertoire, not a universal one.

This regional variation is, on its own, a useful design constraint. Whatever a "column" should mean in an artificial neural network, it should not be a single fixed template imported wholesale from V1; it should be a parameterized template that ranges across the same spectrum of rigidity. The HBCML/ColBaC design is built precisely around this: each column has a protected, stereotyped hard kernel (closer to the V1 or barrel-cortex end) surrounded by adaptable shells with controllable rigidity (closer to the PFC end), plus typed microcolumns capturing distinct intra-column processing roles. The amount of structure absorbed by the kernel versus the shells is a free parameter, to be tuned per problem domain.

The history of columnar architectures in machine learning is older than often recognized. Jacobs et al.'s adaptive mixtures of local experts [9] already had a probabilistic gating network choosing among specialized experts; modern sparse mixture-of-experts work [10,11] has rediscovered much of this at scale. Capsule networks [12] introduce a different kind of columnar grouping, organized around vector-valued part representations. Modular and progressive continual-learning architectures [13,14,15] have explored related ideas under different names. What distinguishes HBCML/ColBaC from these is the explicit two-level Bayesian decomposition: not just sparse routing among modules, but a probabilistic state *inside* each module that informs the routing, together with structural restrictions inside each module specifically designed to preserve causal modularity through sequential learning.

3 HBCML: A General Two-Level Architecture Class

3.1 The two-level idea

A HBCML system contains a collection of structurally restricted component learners. For each context, a sparse top-level controller chooses which components to activate, how to combine them, and whether to reuse, diversify, or recruit additional structure. Inside each active component, a second probabilistic process maintains beliefs about the role of its internal structure: shared abstraction, stabilizing semi-general material, task-local residue, or stale material due for recycling. Those internal beliefs are then allowed to influence the top-level controller. Figure 1 illustrates the architecture.

The essential insight is that different kinds of uncertainty belong at different structural scales [18]. The top level faces a combinatorial selection problem under sparse, noisy signals. The inside of each module faces a structural-hygiene problem requiring continuous, fine-grained adjustments. HBCML insists that these two problems be addressed by distinct but coupled probabilistic mechanisms, rather than collapsed into a single undifferentiated learner.

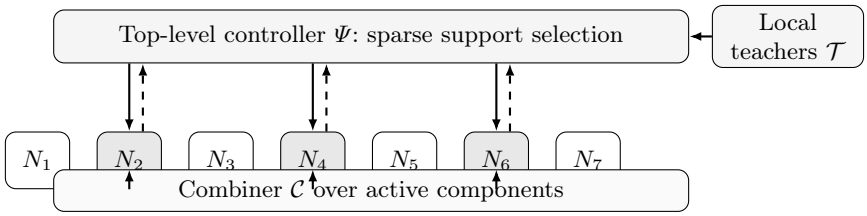

Fig. 1. The HBCML two-level architecture. A sparse top-level controller Ψ selects an active subset of structurally restricted components N_m (shaded). Each active component emits internal certificates upward (dashed arrows) informing future selection. A combiner $\mathcal{C}$ produces the output. Local counterfactual teachers $\mathcal{T}$ correct controller decisions by auditing nearby alternatives.

3.2 Formal definition

Definition 1 (HBCML). *A* hierarchical Bayesian causal modular learner *is a tuple*

$$\mathcal{H} = (\mathcal{N}, \mathcal{C}, \Phi, \Psi, \mathcal{T}),$$

where $\mathcal{N} = \{N_1, \ldots, N_M\}$ *is a family of structurally restricted component learners,* $\mathcal{C}$ *is a combiner mapping the outputs of an active subset to a final prediction,* $\Phi = (\phi_1, \ldots, \phi_M)$ *is the family of internal probabilistic states of the components,* Ψ *is a top-level probabilistic controller determining support selection, reuse, diversification, and recruitment, and* $\mathcal{T}$ *is a family of local counterfactual teachers used to evaluate nearby support changes and nearby internal reorganizations. For a context* t*, the controller induces a (random or soft) support set* $S_t \subseteq \{1, \ldots, M\}$*, and the system output has the form*

$$\widehat{y}_t = \mathcal{C}\big((N_m(x_t; \theta_m, \phi_m))_{m \in S_t}; \Psi\big).$$

The components are not arbitrary dense subnetworks; they come with built-in restrictions meant to match recurring structure in the environment — columns, shells, typed motifs, multiresolution bands, and so on. The claim is not that arbitrary restrictions help, but that restrictions help *when they align with recurrent causal structure in the task family.*

3.3 Adequacy and the link to CCL

Definition 2 (Architectural adequacy). *A HBCML system is* $(\eta_{\text{app}}, \eta_g, \eta_h)$*-adequate for an environment if for each task* t *there exists an intended support set* S_t^* *such that (i) the task predictor can be approximated within error* η_{app} *using only those components, (ii) gradients outside* S_t^* *have norm at most* η_g *in the relevant region, (iii) cross-Hessian blocks between* S_t^* *and its complement have norm at most* η_h *there, and (iv) the combiner is Lipschitz in the component outputs with bounded Jacobian.*

The first key result reduces HBCML to the CCL theorem.

Theorem 1 (Adequacy implies approximate causal modularity). *Let $\mathcal{H}$ be a HBCML system satisfying architectural adequacy, with combiner Jacobian operator norm at most J_C and Lipschitz constant L_C. Then the induced family of task losses is $(\varepsilon_g, \varepsilon_h)$-causally modular, with*

$$\varepsilon_g \le J_C\, \eta_g + \mathcal{O}(\eta_{\mathrm{app}}), \qquad \varepsilon_h \le J_C\, \eta_h + L_C\, \mathcal{O}(\eta_{\mathrm{app}}).$$

Consequently, by the general CCL theorem, the pairwise commutators of update fields satisfy $\big\|[\widetilde{X}_i, \widetilde{X}_j]\big\| \le C_{\mathcal{H}}(\eta_h G_{\max} + \eta_g H_{\max} + \eta_{\mathrm{app}} + \delta)$, and sequential learning is correspondingly close to joint learning, with forgetting bounded by the same right-hand side [3,18].

The proof is a routine chain-rule argument together with adequacy bounds; we omit it for space and refer to the companion technical note. The qualitative content is exactly what one wants: if the architecture's components and combiner align well with the latent mechanism structure of the task family, then the CCL forgetting guarantees automatically apply to the system.

3.4 Local theorems for selection and certificates

The next two results are about the controller and the certificate channel. Both are local rather than global, but both have direct architectural consequences.

Theorem 2 (Exact one-swap teaching is locally monotone). *Suppose a HBCML system uses an exact one-swap auditor at task start, evaluating every $S' \in \mathcal{N}_1(S) := \{(S \setminus \{a_{\mathrm{out}}\}) \cup \{a_{\mathrm{in}}\} : a_{\mathrm{out}} \in S, a_{\mathrm{in}} \notin S\}$ on an audited local objective J_t, and applies a swap only when the best audited neighbor strictly improves J_t. Then the post-teacher support S^{teach} satisfies $J_t(S^{\mathrm{teach}}) \le J_t(S)$, with strict inequality whenever any improving one-swap exists.*

Proof. By construction the algorithm replaces S by an improving one-swap neighbor (strict gain) or leaves S unchanged (zero gain).

This justifies the exact one-swap teacher forcing step used in ColBaC: whatever its global limitations, it cannot worsen the audited local objective. The next theorem makes precise the claim that internal probabilistic state inside a component should be allowed to influence top-level routing.

Theorem 3 (Internal certificates strictly improve reuse-utility estimation). *Let U be a latent reuse utility for a candidate component, B a base score measurable from external fit signals alone, and C an internal certificate derived from the component's internal probabilistic state. Then the Bayes estimators $\widehat{U}_B := \mathbb{E}[U \mid B]$ and $\widehat{U}_{B,C} := \mathbb{E}[U \mid B, C]$ satisfy*

$$\mathbb{E}\big[(U - \widehat{U}_{B,C})^2\big] \;\le\; \mathbb{E}\big[(U - \widehat{U}_B)^2\big],$$

with equality if and only if U is conditionally independent of C given B.

Proof. Standard Rao–Blackwell: conditional expectation on a richer σ-algebra minimizes mean-squared error.

The content of this is that if internal certificates contain *any* genuine information about reuse utility beyond the base fit score, the optimal top-level controller should use them. The architecture is therefore designed so internal certificates — shared-abstraction mass, specificity load, demotion pressure, saturation, similarity signatures — flow upward into selection.

A fourth theorem proven elsewhere [18] shows that representing reusable internal structure with higher posterior precision strictly reduces the expected squared parameter displacement caused by a mistakenly opened gate; this is the formal version of the radial shell semantics in which inner motifs, having been consolidated over more tasks, are more resistant to perturbation.

4 The ColBaC Architecture for Split-MNIST

We now describe a concrete HBCML instance, the Columnar-Bayesian (ColBaC) architecture, in the Split-MNIST configuration we have been running experimentally [19,20].

4.1 Task setting

We use task-incremental Split-MNIST with five binary tasks

$$(0,1),\ (2,3),\ (4,5),\ (6,7),\ (8,9).$$

Each image is converted into a sequence of sixteen 7×7 non-overlapping patches in raster order. Each patch is linearly embedded, coordinates are appended, and the resulting patch sequence is processed by the columnar recurrent/predictive-coding stack. For the present stage of work, each task receives its own local two-way readout head, so that forgetting in the shared representation is not confounded with classifier-head interference. Class-incremental evaluation is the long-run target, but starting there would conflate too many failure modes with the architectural questions we are trying to isolate.

4.2 Columns and microcolumns

The current MNIST-scale configuration uses $N_{\text{col}} = 20$ total columns, partitioned into 2 always-on shared columns, 15 adaptive columns, and 3 reserve columns. Each example uses 5 active columns: the 2 shared plus $k_{\text{nonshared}} = 3$ selected from the adaptive/reserve pool. The selector therefore searches a space of $\binom{18}{3} = 816$ candidate non-shared support sets.

Each column contains $R = 3$ typed microcolumns labelled K, L, B:

- K (kernel center) carries stable kernel-style processing with moderate context memory;

- L (lateral refinement) carries same-region local discriminative refinement;
- B (bridge) carries cross-region or cross-scale context, integrating information across spatial extent.

Each microcolumn carries a hard kernel of width d_m (set to 8–12 for MNIST) plus three concentric shell tiers $S^{(1)}, S^{(2)}, S^{(3)}$ of sizes $|S^{(1)}| = 4$, $|S^{(2)}| = 6$, $|S^{(3)}| = 8$. The shell semantics are radial: inner shells hold reusable abstraction, middle shells hold stabilizing semi-general structure, and outer shells hold task-local exploratory residue. Pruning is outside-in. Within-shell inhibition suppresses double-counting of overlapping causal footprints. Promotion from outer to inner shells reflects consolidation; demotion outward reflects controlled forgetting. Critically, the hard kernel is non-prunable: it is the protected substrate the architecture is willing to defend across all tasks.

4.3 The combiner and hierarchical bias

A small attention-style composer combines the outputs of the five active columns into the per-token representation that is then read out by the task-local head. To bias columns toward local-to-global compositional structure, fine-scale patch tokens are accompanied by coarser auxiliary targets at the quadrant and global levels, with consistency losses tying them together. This is a soft hierarchical-bias signal, not a hard architectural constraint.

4.4 Top-level controller: the teacher-first regime

In the regime currently under experimental evaluation, the top-level support controller uses *exact combinatorial search* at task boundaries rather than a learned selection policy. A multi-objective boundary criterion couples current-task fit with worst-old-task retention, and a switching penalty suppresses gratuitous support churn. During each task, the controller performs frequent one-swap audits over the $3 \times 15 = 45$ neighbors of the current support, applying only swaps that improve the audited objective (Theorem 2). At task scale (five binary MNIST tasks, 816 candidate supports) exact search is computationally cheap, and it produces the strongest possible supervision for any future learned controller. The intended trajectory is a curriculum-for-control ladder: exact search at small scale, heuristic/symbolic search at medium scale, and learned policies at large scale, where the lower rungs supply training data for the upper ones.

4.5 Internal certificates and counterfactual teachers

Each column emits a compact internal certificate upward: shared-abstraction mass (how much of its content is reusable across tasks), specificity load (how much is task-local residue), saturation pressure (whether the column is approaching its capacity for new structure), demotion pressure (whether some inner motifs are stale enough to recycle), and a similarity signature relative to other

Table 1. Offline selector audit, 15 contexts. Numbers below are computed by exhaustive evaluation of all 816 candidate non-shared support sets per context. "Rank" = position of chosen support among all 816, lower is better; "Improvement" = audited-objective gain of best vs. chosen support; "one-swap recovery" = fraction of best-vs-chosen gain captured by the best one-swap neighbor of the chosen support.

Quantity	Value
Contexts where chosen = best support	1 / 15
Mean rank of chosen support (out of 816)	47.7
Median rank of chosen support	44
Worst rank of chosen support	136
Mean best-vs-chosen improvement	0.0267
Median improvement	0.0200
Maximum improvement	0.1076
Mean number of supports within 0.01 of best	15.87
Mean number of supports within 0.05 of best	120
Contexts with at least one improving one-swap	14 / 15
Contexts where best one-swap reaches global best	7 / 15
Mean fraction of full gain captured by best one-swap	≈ 0.86

columns. The top-level controller is permitted, by Theorem 3, to use these certificates in addition to its base fit scores; in practice this is realized by a small score-combination layer that the exact teacher then audits.

The local teacher family $\mathcal{T}$ contains two members. The *support one-swap audit* evaluates, for a chosen support S, every $S' \in \mathcal{N}_1(S)$ by measuring current-task held-out loss, worst-old-task held-out loss, and combined improvement. The *demotion swap audit* evaluates candidate inner/outer shell swaps within columns, measuring whether a proposed swap preserves current-task performance, preserves worst-old-task performance, and improves causal-replaceability structure. The first teacher governs support; the second governs internal organization. Both are conservative by design: a swap is applied only if it strictly improves the audited objective.

5 Preliminary Split-MNIST Results

The current ColBaC system is at the architecture-development stage rather than the empirical-victory stage. The most informative results so far come from a dense *offline selector audit* run across multiple training checkpoints, in which every one of the 816 candidate non-shared supports is exhaustively evaluated against a fixed audited objective at each audited context. This produces a rich picture of how well the online selector is doing relative to what is possible. Table 1 summarizes the main findings across 15 audited contexts (three runs $\times$ five checkpoints).

Several things stand out from this audit. First, the online selector is meaningfully suboptimal: the chosen support equals the global best in only 1 of 15 contexts, with a mean rank near 48. Second, the absolute regret is moderate because the support landscape is broad rather than needle-in-a-haystack sharp: typically 15–120 supports lie within reasonable epsilon-bands of the optimum. The selector problem is therefore less "find the unique correct support" than "identify a good neighborhood and avoid clearly bad regions." Third — and most encouragingly — the local one-swap neighborhood around the chosen support is highly informative: 14 of 15 contexts contain at least one improving one-swap, and the best one-swap captures roughly 86% of the full possible gain on average.

This last finding is the empirical complement to Theorem 2. The teacher mechanism the architecture is built around is not just provably safe; it is, in the actual experimental regime, capable of recovering most of the realizable gain. Two contexts illustrate the pattern at its strongest: in one, the chosen support was $\{14, 16, 19\}$ and the global best $\{8, 17, 19\}$, with improvement 0.1076; in another, the chosen support $\{11, 14, 16\}$ and the global best $\{11, 16, 17\}$, with improvement 0.1070 and full recovery from a single $14 \to 17$ swap.

The audit also reveals systematic bias in the current learned selector. Across the 15 contexts, only 9 distinct chosen supports appear, against 14 distinct best supports. At the column level, columns 16 and 19 are overused by the chosen selector relative to the true best, while columns 6, 9, 10, and 12 are underused (column 12 appears zero times in chosen supports but four times in best supports). This is exactly the kind of structural bias the certificate channel and one-swap teacher are designed to correct, and gives a concrete target for the next round of selector training.

The internal certificates show modest but consistent positive correlations with audited support quality (causal-pair mean: ≈ 0.19; probe-influence mean: ≈ 0.18; certificate shared-mass min: ≈ 0.17). These are not strong enough to support a global parametric ranking rule on their own, but they are clearly informative enough to be valuable in a retrieval- or kNN-style scoring layer, which is consistent with Theorem 3's prediction that any genuine information helps.

We treat these results as architectural validation rather than as a continual-learning leaderboard claim. The structural pieces — exact teacher, certificate channel, sparse selection over a structurally restricted column pool — are all functioning and producing the qualitative behavior the theory predicts. Final continual-learning accuracy numbers, with the recent multi-objective old-task loss accounting fixes properly integrated, will be reported in a forthcoming companion note. The current paper's contribution is the architecture-and-theory pairing, not a new state-of-the-art on Split-MNIST.

6 Scaling to Split-CIFAR

Split-CIFAR is one more small step along a longer scaling path that runs through CIFAR-100, ImageNet-scale benchmarks, multimodal and embodied settings,

and ultimately reinforcement-learning environments. We dwell on it here because it is the next step where the architecture's main scaling decisions can be made deliberately rather than reactively, and because the choices made for CIFAR will largely determine the shape of everything beyond. Moving from Split-MNIST to Split-CIFAR is not a matter of scaling everything up uniformly. The visual manifold is richer, nuisance variation is far larger, and low-level visual abstraction matters much more. The CIFAR version of the architecture should therefore enlarge the stable hard-kernel substrate more than the shell fringe, increase the column pool substantially so sparse support remains meaningful, and introduce a shallow shared visual stem before the columns.

Recommended starting configuration. For Split-CIFAR-10 with $T = 5$ binary tasks and task-local heads, we propose

$$N_{\text{col}} = 40, \quad N_{\text{shared}} = 4, \quad N_{\text{adaptive}} = 30, \quad N_{\text{reserve}} = 6,$$

with active support $k_{\text{nonshared}} = 5$ (so $k_{\text{total}} = 9$ active per example). Each column retains $R = 3$ microcolumns $\{K, L, B\}$, with widened hard kernel $d_m = 32$ and shell sizes $|S^{(1)}| = 10$, $|S^{(2)}| = 20$, $|S^{(3)}| = 30$ (the same 1:2:3 shell ratio as MNIST, scaled up). The active fraction $5/36 \approx 0.14$ remains clearly sparse.

Shared visual stem. A shallow convolutional stem extracts low-level visual primitives before support selection begins:

$$\text{Conv}(3, 32, 3{\times}3) \to \text{Conv}(32, 64, 3{\times}3, \text{stride } 2) \to \text{Conv}(64, 64, 3{\times}3),$$

followed by light patch pooling to 64–96 tokens of width 64–96. This is not a violation of the columnar idea: the columns still carry the continual-learning modularity burden, while the stem provides a better front-end so the columnar machinery is not overwhelmed by raw-pixel work.

Shell dynamics changes. CIFAR produces more nuisance redundancy than MNIST, so same-tier inhibition should be slightly stronger ($\gamma_1 = 0.35, \gamma_2 = 0.22, \gamma_3 = 0.10$) and inward anti-overlap weighting more pronounced ($\omega_1 : \omega_2 : \omega_3 = 4 : 2.5 : 1$, in units of 10^{-3}). Pruning is more conservative: outer shells prune their bottom 10–15% only after a substantial warmup, middle shells only at task boundaries, inner shells not at all in the initial regime. Promotion remains rare and requires multi-task evidence rather than only current-task usefulness.

Selector at CIFAR scale. The CIFAR selector should still be teacher-first, with exact or exhaustive boundary search where feasible (the candidate space $\binom{36}{5} = 376{,}992$ is at the edge of exhaustive evaluation but feasible with light prefiltering), and frequent one-swap audits during training. CIFAR justifies a smarter selector and a larger column pool, not denser support; the active fraction stays sparse.

Onward to CIFAR-100. Split-CIFAR-100 (typically $T = 10$ tasks of 10 classes each, or $T = 20$ of 5) requires further enlargement, particularly of N_{adaptive} and N_{reserve}. Our preliminary plan is $N_{\text{col}} \approx 80$–$120$ with $k_{\text{nonshared}} \in \{6, 7, 8\}$ and a deeper shared stem. The much larger task count also makes the case for moving past pure exact search: the curriculum-for-control ladder (exact $\rightarrow$ heuristic $\rightarrow$ learned) should enter its second rung here, with learned selectors trained on the CIFAR-10 exact teacher's audit traces. Split-CIFAR-100 is also where the systematic bias patterns of the selector (overuse of certain columns, underuse of others) are most likely to matter operationally and where certificate-based correction will be tested at meaningful scale.

7 Beyond Vision: RL and Transformers

The HBCML/ColBaC framework is not committed to vision tasks. We now briefly review two natural extensions of AGI relevance.

Reinforcement learning. The selector itself is already a controller in the RL sense: at each context it chooses a sparse subset of columns, receiving feedback in the form of one-swap audit gains and downstream task performance. Lifting this from supervised continual learning to RL is conceptually a small step. The main mechanical change is that columns now carry option-like structure: a column may encode a reusable behavioral motif (subpolicy, skill, world-model fragment) rather than a perceptual feature, and the support controller becomes an option-selection mechanism along the lines familiar from hierarchical RL. The certificate channel naturally generalizes: instead of (or in addition to) shared-abstraction mass and specificity load, columns can report skill-success statistics, exploration novelty, and option-value estimates. The teacher-first design also generalizes naturally: at an option-boundary, an exact (or heuristic) audit of nearby option-set substitutions can supply training signal for the option-selection policy. The connection to TransWeave [21] becomes most visible here: Bellman–Darboux operator intertwining gives an explicit transfer mechanism from a source RL task to a target one, and the selector certificates are exactly the kind of compact, transferable summary that TransWeave is built around.

Transformers. The relation between ColBaC columns and transformer attention heads is genuinely close, even though they arose from very different traditions. A transformer layer is already a kind of sparse mixture: each attention head computes a different relational pattern, and the layer normalizes and combines them. ColBaC columns can be thought of as more structurally restricted attention modules, with explicit hard kernel + shell organization replacing the dense W_Q, W_K, W_V projections, and with a separate top-level Bayesian controller deciding which heads are active for which contexts. A practical integration path is to interleave conventional attention layers with ColBaC column layers, using transformers for fluid relational mixing and ColBaC columns for the modularity-preserving substrate that survives sequential learning. A wavelet-transformer

integration along these lines is sketched in the broader CBCCN program [19], where wavelet basis structure provides natural multiresolution scaffolding for the column microstructure. The long-run picture is not "ColBaC replaces transformers" but "ColBaC supplies the continual-learning-respecting substrate that transformer-style flexibility can run on top of without paying catastrophic forgetting costs."

8 Conclusion

We have presented a brain-inspired columnar neural architecture (ColBaC) for continual learning, together with a general theoretical framework (HBCML) that subsumes it. The central design move is the two-level decomposition: a sparse top-level controller selecting structurally restricted columns under uncertainty, and an internal probabilistic process inside each column distinguishing reusable causal structure from task-local residue, with the two levels coupled by an upward certificate channel and a family of local counterfactual teachers. The framework inherits its forgetting bounds from the general CCL theorem under a precise architectural-adequacy condition. The exact one-swap teacher is provably locally monotone, and informative internal certificates strictly improve top-level reuse-utility estimation. The current MNIST-scale system is at the architecture-development stage and our preliminary offline audits show both that the system is meaningfully suboptimal in its current form and, more interestingly, that it is suboptimal in exactly the way the architecture is built to repair: 14/15 contexts admit an improving one-swap, 86% of the realizable gain on average is reachable through the local teacher mechanism the architecture already contains.

The most important next steps are empirical: completing the multi-objective old-task accounting fixes, running the Split-CIFAR protocol of Section 6, and beginning the second rung of the curriculum-for-control ladder by training learned selectors on the exact teacher's audit traces. The longer-term picture is more ambitious: HBCML is intended not as a continual-learning trick but as a candidate cognitive substrate compatible with reinforcement learning, transformer-style attention, and (via the certificate channel and TransWeave-style operator transfer) symbolic reasoning and cross-task knowledge transport. Whether all of those compose into something AGI-shaped is an open question; what we have aimed to establish here is that the underlying theory is structurally well-defined and the architectural pieces individually behave as predicted, at least in simple cases.

References

1. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: the sequential learning problem. Psychology of Learning and Motivation **24**, 109–165 (1989)
2. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences **3**(4), 128–135 (1999)

3. Goertzel, B.: Causal coding and the general causal-continual-learning theorem. Manuscript (2025)
4. Mountcastle, V.B.: Modality and topographic properties of single neurons of cat's somatic sensory cortex. Journal of Neurophysiology **20**(4), 408–434 (1957)
5. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. Journal of Physiology **160**(1), 106–154 (1962)
6. Hubel, D.H., Wiesel, T.N.: Functional architecture of macaque monkey visual cortex. Proceedings of the Royal Society B **198**(1130), 1–59 (1977)
7. Woolsey, T.A., Van der Loos, H.: The structural organization of layer IV in the somatosensory region (SI) of mouse cerebral cortex. Brain Research **17**(2), 205–242 (1970)
8. Buxhoeveden, D.P., Casanova, M.F.: The minicolumn hypothesis in neuroscience. Brain **125**(5), 935–951 (2002)
9. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3**(1), 79–87 (1991)
10. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. In: ICLR 2017 (2017)
11. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: scaling to trillion-parameter models with simple and efficient sparsity. Journal of Machine Learning Research **23**(120), 1–39 (2022)
12. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems 30, pp. 3856–3866 (2017)
13. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
14. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)
15. van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint arXiv:1904.07734 (2019)
16. Rao, R.P.N., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience **2**(1), 79–87 (1999)
17. Friston, K.: The free-energy principle: a unified brain theory? Nature Reviews Neuroscience **11**(2), 127–138 (2010)
18. Goertzel, B.: Toward a general theory of hierarchical Bayesian causal modular learning. Manuscript (2026)
19. Goertzel, B.: Columnar Bayesian causal coding networks: a two-level architecture for causal modularity, continual learning, universality, and wavelet–transformer integration. Manuscript (2026)
20. Goertzel, B.: Teacher-first columnar control for continual learning: a two-level architecture with exact support search. Manuscript (2026)
21. Goertzel, B.: TransWeave: transfer learning and cognitive synergy via geodesic cognition guidance and compositional solution transfer using Bellman–Darboux operator intertwining. Manuscript (2026)
22. Goertzel, B.: The general theory of general intelligence: a pragmatic patternist perspective. arXiv preprint arXiv:2103.15100 (2021)