UNSUPERVISED SPARSE EXPANSION FOR CONTINUAL LEARNING IN FLY-INSPIRED SPIKING NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Artificial neural networks trained on a sequence of tasks tend to overwrite the knowledge of earlier tasks, whereas animals continue to acquire memories throughout life. The fruit-fly mushroom body motivates sparse expansion as one way to reduce this interference, but computational models usually freeze randomly sampled projection-neuron–Kenyon-cell connections. We study whether shaping these connections from unlabeled input improves later class-incremental learning in a spiking network. A global orthogonalizing update is unattractive at Kenyon-cell populations of thousands, where the residual would have to be accumulated over the whole population for every unit and the rule itself is non-local, so we apply it within small groups whose composition is redrawn during training; in expectation the pairwise interaction then becomes symmetric — the form of anti-Hebbian decorrelation — instead of the sequential recursion of the original rule. The learned connections are given the same number of inputs per cell and the same weight norm as the random baseline, frozen, and only a shared readout is trained. Across a synthetic odor stream and two image benchmarks, the learned mapping improves final class-incremental accuracy without labeled replay or test-time task identity. Feature measurements show lower class and task correlation on the image inputs, although active-set overlap increases slightly on the odor stream. The method combines an unsupervised feature stage with a biologically inspired spiking circuit, while requiring access to unlabeled inputs from future classes.

1 INTRODUCTION

Fruit flies can form new odor associations through a small circuit in which projection neurons (PNs) drive a much larger population of Kenyon cells (KCs). Sparse KC activity supports odor discrimination (Turner et al., 2008; Lin et al., 2014; Caron et al., 2013), and plasticity at the KC-to-output synapses is the substrate of odor learning in this circuit (Hige et al., 2015). This circuit has therefore motivated machine-learning models of both odor recognition and sequential learning (Dasgupta et al., 2017; Zou et al., 2025a). Yet the circuit analogy leaves a practical question open: if the input-to-KC connections are fixed before supervised learning, does it matter whether they were sampled at random or shaped by unlabeled sensory experience?

The question matters for class-incremental learning (CIL), where a model encounters classes in sequence and must classify among all classes seen so far without a task label. Neural networks commonly forget old classes when new-class updates change shared weights (McCloskey & Cohen, 1989; French, 1999; De Lange et al., 2022). Replay, parameter regularization and task-specific allocation mitigate this problem, but bring different storage, optimization or inference requirements (Rebuffi et al., 2017; Kirkpatrick et al., 2017; Zenke et al., 2017; Serrà et al., 2018). A sparse expansion layer can reduce overlap before the readout is trained; whether an unsupervised, data-dependent expansion helps beyond an equally sparse random one is the focus here.

We examine this question in a leaky-integrate-and-fire spiking neural network (SNN) based on the PN–KC–output motif. During a separate, unlabeled feature-learning phase, we update sparse PN–KC connections with a grouped version of the established generalized Hebbian algorithm (Sanger, 1989). We then freeze these connections and train a single KC-to-output classifier on sequential labeled tasks. A global orthogonalizing update is unattractive here for two structural reasons: the residual has to be accumulated over the whole KC population for every unit, so its cost grows with

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

population size, and the rule itself is non-local, which is hard to reconcile with synaptic plasticity. We therefore restrict it to small groups whose composition is redrawn during training. This is not only an approximation: with random regrouping the sampling coefficient of the pairwise term becomes the same for every pair in expectation, which turns the ordered recursion of the original rule into a symmetric anti-Hebbian term (Appendix D), and each update uses only the units of one group rather than the whole KC population. The learned connections are held to the same number of inputs per cell and the same weight norm as the random baseline, so the two mappings differ in how their connections are chosen rather than in how many there are. Sparseness, inhibition, adaptation and fixed adult input connections retain features of the circuit model. We treat this as a biologically inspired engineering approximation, not as evidence that flies implement this learning rule.

We begin with a synthetic odor stream to test the circuit in its motivating domain, then apply the same motif to raw MNIST pixels and frozen Vision Transformer (ViT) features from CIFAR-100 (LeCun et al., 1998; Krizhevsky, 2009; Dosovitskiy et al., 2021). The image experiments test transfer of the computational motif; they are not a model of fly vision. Across these three streams the learned mapping improves final CIL accuracy relative to the random mapping. The odor advantage is smaller and depends on noise level, whereas the image results show larger reductions in measured class and task correlation. Alternative task partitions, a second unsupervised rule and synaptic-intelligence (SI) regularization delimit the scope of the claims.

Our contributions are twofold. First, we turn an unsupervised expansion stage into a usable design for a fly-inspired SNN: we limit the interaction range of the orthogonalizing update by grouping and reshuffling, which in expectation gives it a symmetric anti-Hebbian form, and we match the resulting connections to the random baseline in synaptic budget, so that the two maps are compared under the same connection budget rather than a different one. Second, we relate the resulting accuracy to measured properties of the KC code and test it across odor noise, task partitions, grouping choices and readout regularization. The Hebbian rule itself is prior work; our contribution is the circuit-level design, its biological grounding and its controlled evaluation for CIL.

2 Related Work

Fly-inspired learning. Anatomical studies report that a small PN population projects divergently onto a much larger KC population, with each KC sampling a handful of PNs and the projection approximating a random draw (Caron et al., 2013; Lin et al., 2014); sparse KC activity is then regulated by feedback and lateral inhibition. This makes random sparse connectivity a substantive biological baseline rather than a straw man. Computational accounts have used this expansion for separating overlapping inputs and for similarity search (Dasgupta et al., 2017). Li et al. (2025) studied lateral inhibition and spike-frequency adaptation in a spiking fly olfactory model for noisy odor discrimination; we retain those circuit mechanisms and ask what changes when the PN–KC map itself is shaped by unlabeled input. Zou et al. (2025a) studied a fly-inspired module for continual learning with sparse random expansion and ANN-based downstream learning. Their later Fly-CL framework addresses decorrelation and training time in pretrained-model continual representation learning (Zou et al., 2026), while FlyLoRA uses a frozen sparse random projection within parameter-efficient adaptation and multi-task merging (Zou et al., 2025b). These methods keep the expansion random and act on the downstream model; ours adapts the sparse input map before the task sequence and evaluates the resulting representation in a spiking circuit with a single shared readout. These comparisons motivate the design; they do not establish PN–KC plasticity in the fly.

Continual learning in ANNs and SNNs. Sequential training of a shared model overwrites earlier solutions, and the standard remedies differ in what they require at inference time. Replay methods store or generate past examples (Rebuffi et al., 2017; Prabhu et al., 2020); regularization methods penalize changes to parameters that mattered for earlier tasks (Kirkpatrick et al., 2017; Zenke et al., 2017); architectural methods isolate parameters per task and therefore usually need a task identifier at inference (Mallya & Lazebnik, 2018; Serrà et al., 2018); gradient-projection methods constrain updates to directions that do not disturb earlier tasks (Farajtabar et al., 2020; Saha et al., 2021). All of these act during supervised task training. A parallel line keeps the representation usable while the model continues to adapt, without retaining raw samples from past tasks. EFC targets the cold-start regime, in which the first task has too little data to fix a backbone, and regularizes drift of the features themselves during backbone updates (Magistri et al., 2024); AdaGauss studies

both randomly initialized and pretrained backbones, trains the extractor, and adapts class statistics in the latent space to counter task-recency bias (Rypeść et al., 2024). Both therefore adapt the representation during the task sequence, whereas we shape it once, before the first labeled task, and then measure what a frozen code can support. In spiking networks, continual learning has been approached with task-agnostic local plasticity (Soures et al., 2024) and with Hebbian orthogonal projections during sequential learning (Xiao et al., 2024). Our protocol places the unsupervised stage before the labeled sequence: the expansion is learned once from unlabeled inputs, frozen, and then evaluated with no task identifier, no stored examples and no prototype rehearsal. This ordering buys clarity of comparison at the cost of access to the future classes' unlabeled training inputs, which is the main assumption we have to state.

Unsupervised feature learning. Oja's rule and Sanger's generalized Hebbian algorithm extract principal directions from input statistics (Oja, 1982; Sanger, 1989). Reducing redundancy as an organising principle of sensory coding goes back further (Barlow, 1961), and a local anti-Hebbian rule that implements it was given by Földiák (1990); sparse coding and related objectives follow the same logic. We use a grouped, sparse implementation of an existing rule rather than propose a new principal-component algorithm. Two properties make the question empirical rather than settled. First, principal directions need not preserve the detail that distinguishes classes, so a representation that is better compressed is not automatically better for classification. Second, a sparse grouped update need not produce globally orthogonal KC activity, since the connections are constrained to a fixed fan-in and norm, only a small fraction of units is updated per example, and the resulting code is measured rather than assumed. The experiments are therefore organised around measuring the code a fixed unsupervised stage produces, and around comparing it to an equally sparse random map under identical budgets.

3 Model and Evaluation Protocol

3.1 Spiking Circuit and Feature Learning

Figure 1 shows the model. Odor inputs drive 51 olfactory receptor neurons (ORNs) and 51 PNs, followed by 2,000 KCs and one output unit per class. PN lateral inhibition, KC feedback inhibition and spike-frequency adaptation follow the circuit design; KC threshold homeostasis targets sparse activity. All units use leaky-integrate-and-fire dynamics (Appendix C). For MNIST, the PN and KC populations are 784 and 15,680; for CIFAR-100 ViT features, they are 768 and 15,360. The visual inputs are mapped into the same computational motif with dataset-specific input scales and connection budgets (Table C.1).

Each normalized input component enters the ORN layer as an analog current, together with the small constant bias listed in Table C.1; ORN dynamics generate spikes, without an additional rate- or latency-encoding scheme. Every unit is a leaky integrate-and-fire neuron with a layer-specific membrane time constant, integrated with a step of $\Delta t = 1$ ms,

$$V_j[t+1] = \beta^X V_j[t] + I_j[t+1], \qquad \beta^X = \exp(-\Delta t/\tau_{\mathrm{m}}^X), \tag{1}$$

where V_j is the membrane potential of unit j in layer X and I_j its input current; a unit spikes when its potential reaches its threshold, after which the potential is reset to zero, or reduced by the threshold in the KC layer. The lateral-inhibition and adaptation currents that enter I_j, and the homeostatic adjustment of the firing thresholds, are specified in Appendix C.

The classical generalized Hebbian update, shown here as a reference form for PN activity x_j, KC response y_i, weight w_{ij} and step size η, is

$$\Delta w_{ij} = \eta y_i \left(x_j - \sum_{k \le i} y_k w_{kj} \right). \tag{2}$$

Our implementation uses this residual update within randomly formed groups of ten KCs, with the partition redrawn during training (Appendix C), plus a separate self-decay and nonnegative shadow weights. Keeping the largest K values per KC and normalizing the retained vector enforce the same fan-in and norm budget as the random-projection (RP) condition; only the learned-map condition

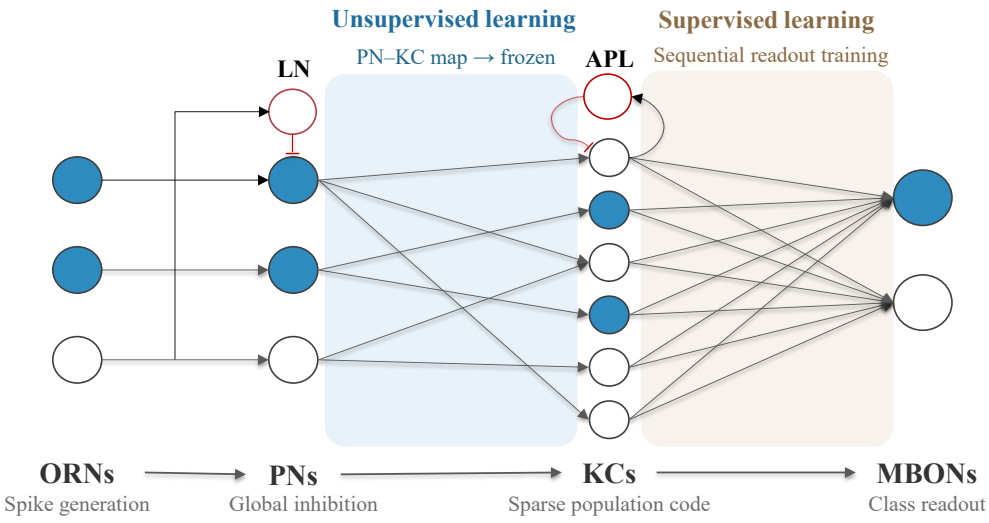


Figure 1: Two-stage fruit-fly-inspired spiking circuit. Sparse PN–KC connections are learned from unlabeled training inputs or sampled randomly, then frozen, and only the KC-to-output readout is trained on the labeled tasks. Connections are representative; filled and open circles mark active and inactive units for the illustrated input, and the LN and APL units are drawn singly and project uniformly to their layer (Appendix C).

adapts PN–KC weights. The two maps therefore differ in which connections they use, but not only in that: the KC thresholds are adapted towards the same target rate in both conditions before Phase 2 (Appendix C), so they also differ in that adaptation history. Reshuffling and whitening remain algorithmic operations, and online whitening, active-KC selection, top-K projection and nonnegative clipping all separate the spiking system from the classical generalized Hebbian algorithm. We therefore call the design *biologically inspired*, without claiming a synapse-local implementation of Eq. 2 or global orthogonality; Section 3.2 states what the grouping does imply in expectation.

3.2 Expected Form of the Grouped Update

Grouping is not only a way of limiting the interaction range. Let $\xi_{ik}(t) = 1$ if unit k shares a group with unit i in regrouping round t and precedes it in the order of that group, and 0 otherwise; Appendix C gives the interval at which the partition is redrawn in each set of experiments. The single-unit form of Eq. 2 then becomes

$$\Delta w_i \propto y_i \widetilde{x} - \sum_{k \neq i} \xi_{ik}\, y_i y_k w_k - \lambda y_i^2 w_i, \tag{3}$$

with whitened PN activity $\widetilde{x}$ and self-decay coefficient λ. For a uniformly random partition of M units into groups of G, the indicator has the same expectation for every ordered pair, $\mathbb{E}[\xi_{ik}] = (G-1)/[2(M-1)]$ (Appendix D). Under the slow-learning approximation — weights change little between two regroupings, so the indicator is approximately independent of them — the expected update replaces the sum over co-active units by that single coefficient multiplying $\sum_{k \neq i} \mathbb{E}[y_i y_k] w_k$. The coefficient multiplying each pairwise term is therefore the same for every pair, so the expected update is symmetric between units: each unit's change is driven by one uniform coefficient over all other units, rather than by the ordered recursion of the sequential algorithm, and this is the form of anti-Hebbian decorrelation (Földiák, 1990). What is equal is the sampling coefficient, not the realised strength of each pair's interaction, which still depends on the co-activity $y_i y_k$ and the current weight w_k; and the statement concerns the learning-time update, not the inhibition the APL delivers at run time. The symmetry is one of the sampling coefficient rather than of the interaction geometry: no anatomical neighbourhood is implied. Reshuffling is what makes this expectation pair-symmetric; with a fixed partition, the expected interaction would depend on the particular assignment instead.

3.3 Two-Stage Class-Incremental Protocol

Phase 1 uses unlabeled *training* inputs from every class in the stream to learn the PN–KC map; the RP map is fixed. Labels and task boundaries are unavailable to this phase, but its inputs are. Both maps are then frozen. Phase 2 trains the shared KC-to-output readout with cross-entropy and surrogate gradients, one labeled task at a time. No previous-task examples are stored or replayed in Phase 2, and evaluation supplies no task identifier. Predictions use the output units of all classes seen so far. Thus the claim is rehearsal-free *supervised* CIL after transductive unlabeled feature learning, not fully online learning from an unseen future distribution.

The synthetic odor stream contains 110 classes in ten tasks of eleven classes, evaluated at four noise levels. Each class has four training indices and eight independently generated test examples; training noise is resampled when an index is accessed. Phase 1 uses the 440 training indices only. MNIST uses ten one-class tasks on raw pixels. CIFAR-100 uses 25 four-class tasks on features from a frozen ImageNet-pretrained ViT-B/16; the ViT is an external pretrained model and is shared across mapping conditions. Evaluation uses the output units of all classes seen so far and supplies no task identifier, so an error on an old class can only be corrected by the readout; restricting prediction to the current task's classes would change the setting we target. We report incremental accuracy after each task, and the accuracy over all classes seen so far after the final task; we write this average accuracy as AAC. Both are given as mean $\pm$ standard deviation over five seeds. BWT compares each old task's accuracy after the last task with its accuracy immediately after it was learned; more negative BWT indicates more forgetting. BWT is reported for the main comparison (Table 4.1); the control experiments report AAC. Appendix A gives the definitions of both metrics and of the representation measures; Appendix C lists the parameters. Comparisons are within each dataset and shared protocol, not across sensory modalities or against fully online CIL methods.

4 Continual Learning Results

4.1 Odor Learning: AAC and Representations

Table 4.1 reports the odor results as AAC, measured after the complete sequence, at each noise level. We report AAC rather than incremental curves: the learned map is ahead at the final task in every condition reported here; the margin is smallest at $\sigma = 0.035$ (6.14 points against a run-to-run spread of 10.74 points for the fixed map) and BWT is less negative for the learned map in all four conditions.

The frozen KC representation provides a narrower interpretation than a general "task mask." On odor inputs the learned code separates classes better than the fixed map: the Fisher ratio rises from 1.66 to 2.15 at $\sigma = 0.025$ and from 1.53 to 1.65 at $\sigma = 0.040$, driven by a larger inter-class distance (6.20 to 7.96 at $\sigma = 0.025$) rather than by tighter classes (3.73 to 3.70). The separation improvement shrinks monotonically as the noise level grows, whereas the accuracy margin does not follow the same trend (7.78, 11.37, 6.14 and 10.41 points at the four noise levels), so better separation is associated with the gain without tracking it. The silhouette coefficient is negative for the fixed map at every noise level and for the learned map at three of the four, turning slightly positive at $\sigma = 0.025$ ($+0.035 \pm 0.008$); separation therefore remains limited throughout, which is what near-zero silhouette values indicate, even though the learned code is relatively better separated. The dimension needed to explain 90% of the variance of the code falls only slightly (165.0 to 152.2 at $\sigma = 0.025$, 165.2 to 160.0 at $\sigma = 0.040$, Figure 2).

The two interference measures move the other way, and this is worth stating plainly. Class and task correlation are small in both maps – below 0.05 throughout – and slightly *higher* for the learned map (task correlation 0.0236 to 0.0492 at $\sigma = 0.025$); active-set task overlap behaves the same way (0.036 to 0.062 at $\sigma = 0.025$). On the odor stream, then, the accuracy gain comes with better class separation at essentially unchanged sparsity and with slightly more, not less, overlap between tasks. These measurements therefore do not support reduced task overlap as the explanation for the accuracy gain, although they are static and cannot exclude a mechanism that acts during training; noise removal and class separation remain hypotheses rather than isolated causal mechanisms, and the Oja control in Section 4.3 shows that part of the gain appears without a cross-unit term at all.

Table 4.1: AAC and BWT after the complete task sequence (mean ± s.d. over five seeds); more negative BWT means more forgetting. All rows share the protocol, readout and full-output argmax; the odor stream runs ten tasks of eleven classes, MNIST ten one-class tasks, CIFAR-100 25 four-class tasks on frozen ViT features.

Stream	Tasks	Fixed RP AAC (%)	Learned map AAC (%)	Fixed RP BWT	Learned map BWT
Odor, $\sigma = 0.025$	10×11	79.02 ± 1.43	86.80 ± 3.66	-23.18 ± 1.35	-12.32 ± 1.87
Odor, $\sigma = 0.030$	10×11	60.61 ± 3.78	71.98 ± 2.70	-40.45 ± 5.39	-26.84 ± 2.46
Odor, $\sigma = 0.035$	10×11	50.36 ± 10.74	56.50 ± 6.38	-45.98 ± 12.03	-39.34 ± 3.26
Odor, $\sigma = 0.040$	10×11	36.93 ± 8.57	47.34 ± 10.03	-49.37 ± 18.24	-42.32 ± 10.71
MNIST	10×1	23.39 ± 0.47	58.64 ± 1.37	-65.26 ± 1.68	-29.85 ± 2.32
CIFAR-100	25×4	19.09 ± 0.51	31.41 ± 1.79	-29.29 ± 3.86	-15.73 ± 4.80

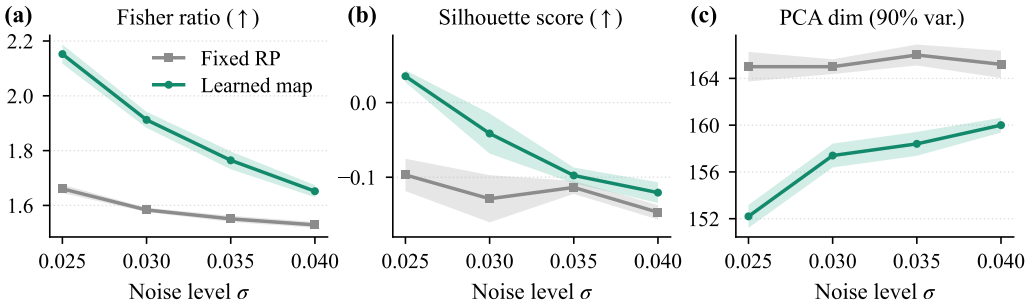


Figure 2: Frozen KC representation on the odor stream across noise levels: Fisher ratio, silhouette coefficient and number of principal components explaining 90% of variance. These measures characterize the codes; they do not isolate the cause of the accuracy difference. Bands show the standard deviation over five seeds.

4.2 Transfer of the Circuit Motif to Image Inputs

On MNIST and CIFAR-100, the learned sparse map also improves AAC (Table 4.1). Figure 3 reports the incremental accuracy of these two streams, where the two maps separate early and the learned map never falls behind. On MNIST the final difference is 35.25 percentage points. On CIFAR-100 it is 12.32 points, although absolute accuracy after all 100 classes remains low. The absolute level has a mechanical component: with 100 classes arriving four at a time, a late task extends a readout that already covers most of them, over a fixed transformation of ImageNet features. Both maps use the same features and the same readout, so the comparison is internally controlled; this does not show that the gap is independent of task difficulty, and it is not a state-of-the-art CIL comparison. Forgetting is smaller for the learned map on both streams as well.

The representation measurements in Figure 4 describe a different picture, much larger in magnitude and in the direction expected of reduced interference. Task-level measures group classes into tasks. Only MNIST differs from its main experiment, using five two-class tasks (with one class per task the task and class centres would coincide); the odor and CIFAR-100 analyses use their main groupings. On MNIST, class correlation falls from 0.55 to 0.17, task correlation from 0.73 to 0.19 and active-set overlap from 0.47 to 0.20; on CIFAR-100 the corresponding values fall from 0.47 to 0.13, 0.78 to 0.17 and 0.58 to 0.25. Class separation improves in the same direction: the Fisher ratio rises from 0.55 to 0.67 on MNIST and from 0.42 to 1.07 on CIFAR-100, and the silhouette coefficient, negative for both maps, moves from -0.25 to -0.14 and from -0.09 to -0.05. The dimension needed for 90% of the variance collapses on MNIST (1,694 to 132) and falls by about a third on CIFAR-100 (4,002 to 2,624).

The two image streams differ in how far the code is compressed, and compression is not automatically useful. On MNIST the learned map keeps about one thirteenth of the dimensions of the random code, alongside the larger accuracy gap (35.25 points); on CIFAR-100 the reduction is smaller and so is the gap (12.32 points). Lower variance dimension does not by itself show that only nuisance

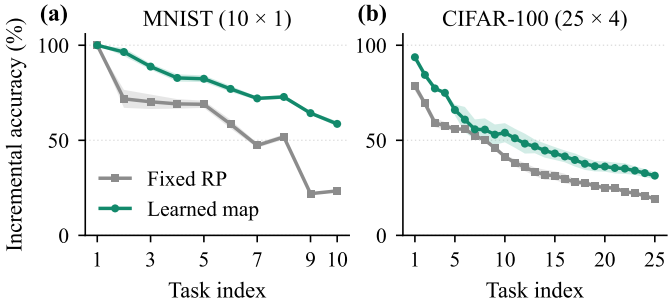


Figure 3: Incremental accuracy over image task sequences: (a) MNIST raw pixels and (b) frozen ViT features from CIFAR-100. All previously seen classes are included at each evaluation point. Externally pretrained ViT features are fixed in both CIFAR-100 conditions. Bands show the standard deviation over five seeds.

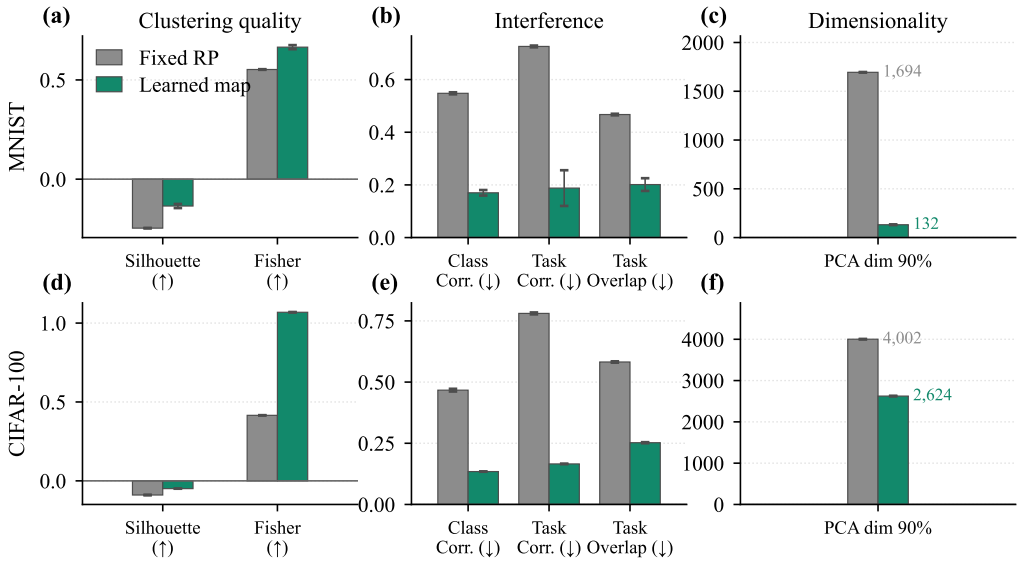


Figure 4: Frozen KC representation for MNIST (top) and CIFAR-100 (bottom): clustering, class/task correlation and active-set overlap, and dimension for 90% explained variance. For MNIST, task-level measures group the ten classes into five two-class tasks. Error bars show the standard deviation over five seeds. Silhouette scores remain negative in both image settings; correlations and overlaps are descriptive measures.

features were removed, and the silhouette values remain negative, so even the better code does not form clean class clusters in absolute terms. The measurements are best read as a description of the code rather than as an identified mechanism: correlation, overlap, dimension and class separation all change together, the odor stream shows a gain without the correlation reduction, and the controls in Section 4.3 show that neither grouping nor orthogonalization is indispensable. On the image inputs, the learned map lowers both class and task correlation while improving class separation, which is the combination a shared readout can exploit to keep old classes apart while new ones arrive.

4.3 Controls: Rule, Task Partition and Readout Regularization

The gains are not exclusive to one group size or even to cross-unit orthogonalization. On the odor stream at $\sigma = 0.030$, an Oja update reaches $65.27 \pm 5.65\%$ AAC against $60.61 \pm 3.78\%$ for the fixed RP baseline; the grouped update reaches $71.41 \pm 6.75\%$, $71.98 \pm 2.70\%$ and $68.50 \pm 8.42\%$

Table 4.2: Odor rule and group-size control at $\sigma = 0.030$, ten tasks of eleven classes. AAC (%, mean $\pm$ s.d. over five seeds). All learned-map conditions used unlabeled feature learning before the task sequence.

Feature map	AAC (%)
Fixed RP	60.61 ± 3.78
Oja update	65.27 ± 5.65
Grouped update, $G = 5$	71.41 ± 6.75
Grouped update, $G = 10$	71.98 ± 2.70
Grouped update, $G = 20$	68.50 ± 8.42

Table 4.3: AAC (%) under an alternative task partition, keeping all other settings unchanged (mean $\pm$ s.d. over five seeds). The default partitions are those in Table 4.1.

Stream	Partition	Fixed RP AAC (%)	Learned map AAC (%)
Odor, $\sigma = 0.030$	5×22	63.64 ± 7.64	78.52 ± 9.63
MNIST	5×2	33.20 ± 1.48	53.68 ± 5.33
CIFAR-100	10×10	22.89 ± 2.62	33.61 ± 2.53

for groups of 5, 10 and 20, respectively (Table 4.2). The standard deviations do not justify a unique optimal group size or a claim that orthogonalization alone causes the gain.

The Oja comparison matters because it improves the mean without a cross-unit residual term, and it also bounds how the code measurements may be read: since a rule without a cross-unit term gains on the odor stream as well, better class separation there cannot be attributed to the symmetric interaction alone. It is compatible with a benefit from unsupervised adaptation more broadly, not a demonstration that orthogonalization is necessary. Across the three grouped sizes, mean accuracy peaks at ten units but the standard deviations overlap substantially; a larger sweep would be needed to select G reliably, and at the scales we can afford the group size and the regrouping scheme both move the mean by less than the seed-to-seed spread. The reshuffling experiment in Table B.1 changes the mean by at most 5.4 points, comparable to the seed-to-seed spread at these group sizes, so dynamic grouping should not be presented as a separately established accuracy contribution.

A second task partition preserves the learned-map advantage in each stream (Table 4.3): margins are between 2.4 and 35.3 points over the settings tested, where the range spans the rows of Tables 4.1 and E.1, so the ordering is not an artifact of one partition. Readout regularization is complementary rather than redundant. Adding a synaptic-intelligence (SI) penalty (Zenke et al., 2017) to the readout loss, with the coefficient tuned per dataset, improves the mean in every configuration of Table 4.4, and the learned map with the penalty has the best mean on every dataset — but the gain is distributed differently. On MNIST it is 7.60 points for the fixed RP map against 4.38 for the learned map; on CIFAR-100 it is 1.62 against 0.18, and the latter is smaller than the seed-to-seed spread of those two rows (s.d. 1.79 and 1.83), so there is no clear evidence of an improvement there. The unregularized learned map therefore still exceeds the regularized fixed RP baseline on both streams (58.64 against 30.99, and 31.41 against 20.71); on the odor stream both gain substantially (+19.78 and +15.11 points), and fixed RP with SI (80.39%) exceeds the unregularized learned map (71.98%). The feature stage is therefore not a substitute for readout-level continual-learning methods, and the two combine. At some stronger penalty settings the fixed RP readout learned poorly; those settings are not the basis of the comparison. No broad hyperparameter comparison is claimed.

5 Discussion and Limitations

In this two-stage SNN, learning a sparse PN–KC map from unlabeled inputs improves shared-readout accuracy over an equally sparse fixed map. The accompanying image-code correlations and overlaps are consistent with reduced interference, but the odor overlap result and the Oja control preclude a single "task-mask" explanation. The grouped Hebbian rule specifies the update; the

Table 4.4: AAC (%) with and without a synaptic-intelligence penalty on the readout loss, under the default partitions (mean ± s.d. over five seeds; SI coefficient 0.03 for the odor stream and 0.5 for the vision datasets). The columns without SI repeat the corresponding rows of Table 4.1.

Dataset (partition)	Fixed RP	Fixed RP + SI	Learned map	Learned map + SI
Odor (10×11, $\sigma = 0.030$)	60.61 ± 3.78	80.39 ± 3.15	71.98 ± 2.70	87.09 ± 3.12
MNIST (10×1)	23.39 ± 0.47	30.99 ± 1.03	58.64 ± 1.37	63.02 ± 1.22
CIFAR-100 (25×4)	19.09 ± 0.51	20.71 ± 0.55	31.41 ± 1.79	31.59 ± 1.83

representation measurements describe outputs of the complete circuit and do not isolate one causal mechanism.

Several limits matter. Phase 1 has access to unlabeled examples of classes that arrive later in Phase 2, so the protocol does not cover a stream with no prior exposure to future classes. The grouped update and whitening increase training cost: Phase 1 runs for 1000–1500 epochs over the training set before the first labeled task, which is substantially slower than sampling the random map. The reduction in variance dimension reported above may also discard details that fine-grained discrimination needs, so we expect this design to be weaker on fine-grained streams than the aggregate accuracies suggest. A ViT trained on ImageNet supplies the CIFAR-100 input representation; this external training matters when comparing against methods using pixels alone. Finally, random PN–KC convergence has direct anatomical support, while our grouped learning rule remains an algorithmic hypothesis. Biological plausibility of individual ingredients does not establish that this plasticity occurs in the fly.

The two streams differ in how the code changes, and the difference is informative rather than a nuisance. On the image inputs the learned map lowers class and task correlation and reduces task overlap at the same time as it improves class separation, which is the combination a shared readout can exploit. On the odor stream separation improves while correlation and overlap stay small and rise slightly, so the property that separates the two maps on images is not what separates them there. One reading is that the learned stage removes input noise rather than re-organising the class geometry: separation improves most where the noise is lowest, while the accuracy margin peaks at the second-lowest level ($\sigma = 0.030$) rather than at the lowest, so the two quantities are related without being proportional. Distinguishing the two would require inputs whose noise level is controlled independently of class separability, which the synthetic stream does not provide.

The cost of Phase 1 is paid once per input domain: the map is frozen, reused for every task of the stream, and not retrained for a further stream over the same input statistics. Where inputs arrive continuously and are never labeled, this is the natural place to spend unsupervised computation; adapting the map during the labeled sequence is the alternative we excluded and did not cost here.

Future work should measure pretraining cost, test no-future-data protocols and fine-grained streams, and isolate which components of the update preserve class information, which would distinguish a useful engineering intervention from a circuit-level biological mechanism.

6 Conclusion

An unlabeled feature-learning stage improves a sparse fruit-fly-inspired spiking circuit for sequential class learning in the tested settings. The learned map has higher AAC than a matched random map on odor, MNIST and CIFAR-100, and combines with readout regularization, in a two-stage protocol that uses unlabeled inputs from future classes. The measured code supports the design in the image settings, where class and task correlation fall while class separation improves, and it does not support the same reading on the odor stream, where separation improves without a fall in task overlap; we report both. The learning rule and native PN–KC plasticity are not claimed as discoveries; we contribute the circuit-level design that admits an orthogonalizing update into a sparse spiking expansion, its expected form, and a controlled evaluation against an equally sparse random map.

AI USE STATEMENT

Generative AI tools assisted manuscript editing, simulation-code development and debugging, figure-script writing and reference checking, and took part in developing the derivation in Appendix D. The authors verified that derivation, reviewed the text, code, results and cited sources, and take responsibility for all content and claims. The research questions, the experimental design and the interpretation of the results are the authors' own work.

REPRODUCIBILITY STATEMENT

Section 3 and Appendices A–D describe the circuit, protocol, parameters and measures. Appendices B and E report the rule and grouping controls and the odor stream under a second partition. Reported accuracies summarize five seeds.

ETHICS STATEMENT

The experiments use public image benchmarks and synthetic odor inputs; no human subjects or personal data are involved. The fly circuit is used as computational inspiration, not as a claim of a verified biological intervention.

REFERENCES

Horace B. Barlow. Possible principles underlying the transformation of sensory messages. In Walter A. Rosenblith (ed.), *Sensory Communication*, pp. 217–234. MIT Press, 1961.

Sophie J. C. Caron, Vanessa Ruta, L. F. Abbott, and Richard Axel. Random convergence of olfactory inputs in the *Drosophila* mushroom body. *Nature*, 497(7447):113–117, 2013.

Sanjoy Dasgupta, Charles F. Stevens, and Saket Navlakha. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796, 2017.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *AISTATS*, 2020.

Péter Földiák. Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, 64(2):165–170, 1990.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

Toshihide Hige, Yoshinori Aso, Mehrab N. Modi, Gerald M. Rubin, and Glenn C. Turner. Heterosynaptic plasticity underlies aversive olfactory learning in Drosophila. *Neuron*, 88(5):985–998, 2015. doi: 10.1016/j.neuron.2015.11.003.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Haiyang Li, Liao Yu, Qiang Yu, and Yunliang Zang. Seemingly redundant modules enhance robust odor learning in fruit flies. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Andrew C. Lin, Alexei M. Bygrave, Alix de Calignon, Tzumin Lee, and Gero Miesenböck. Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. *Nature Neuroscience*, 17(4):559–568, 2014.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.08840.

Simone Magistri, Tomaso Trinci, Albin Soutif-Cormerais, Joost van de Weijer, and Andrew D. Bagdanov. Elastic feature consolidation for cold start exemplar-free incremental learning. In *International Conference on Learning Representations*, 2024.

Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Gordon H. Bower (ed.), *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017.

Grzegorz Rypeść, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Task-recency bias strikes back: Adapting covariances in exemplar-free class incremental learning. In *Advances in Neural Information Processing Systems*, 2024.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989.

Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, 2018.

Nicholas Soures, Peter Helfer, Anurag Daram, Tej Pandit, and Dhireesha Kudithipudi. TACOS: Task agnostic continual learning in spiking neural networks, 2024. arXiv:2409.00021.

Glenn C. Turner, Maxim Bazhenov, and Gilles Laurent. Olfactory representations by *Drosophila* mushroom body neurons. *Journal of Neurophysiology*, 99(2):734–746, 2008.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Hebbian learning based orthogonal projection for continual learning of spiking neural networks. In *International Conference on Learning Representations*, 2024.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

Heming Zou, Yunliang Zang, and Xiangyang Ji. Structural features of the fly olfactory circuit mitigate the stability-plasticity dilemma in continual learning, 2025a. URL https://arxiv.org/abs/2502.01427.

Heming Zou, Yunliang Zang, Wutong Xu, Yao Zhu, and Xiangyang Ji. FlyLoRA: Boosting task decoupling and parameter efficiency via implicit rank-wise mixture-of-experts. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://arxiv.org/abs/2510.08396.

Heming Zou, Yunliang Zang, Wutong Xu, and Xiangyang Ji. Fly-CL: A fly-inspired framework for enhancing efficient decorrelation and reduced training time in pre-trained model-based continual representation learning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.16877.

A Evaluation Metrics

Continual-learning metrics. Let $a_{i,j}$ be the accuracy on the test set of task j after training on task i, for a stream of T tasks. After a model has learned task t we record the incremental accuracy, i.e. the accuracy over all classes seen up to and including task t, evaluated under the full class-incremental protocol with no task identifiers. AAC is the same quantity measured after the last task and averaged over all tasks of the stream,

$$\mathrm{AAC} = \frac{1}{T}\sum_{j=1}^{T} a_{T,j}, \tag{4}$$

and BWT (Lopez-Paz & Ranzato, 2017) averages, over the tasks learned before the last one, the change in that task's accuracy between the moment it was learned and the end of the stream,

$$\mathrm{BWT} = \frac{1}{T-1}\sum_{j=1}^{T-1} \left(a_{T,j} - a_{j,j}\right). \tag{5}$$

A more negative BWT means more forgetting; both quantities are reported as mean $\pm$ standard deviation over the five seeds.

We quantify the quality of KC-layer representations with the following metrics:

Intra-class distance:

$$D_{\mathrm{intra}} = \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{N_c}\sum_{i\in \mathrm{Class}_c} \|\boldsymbol{z}_i - \boldsymbol{\mu}_c\|_2\right) \tag{6}$$

$\boldsymbol{z}_i$ is the activation vector of the i-th sample, and $\boldsymbol{\mu}_c$ is the center of the c-th class.

Inter-class distance:

$$D_{\mathrm{inter}} = \frac{2}{C\left(C-1\right)}\sum_{i=1}^{C}\sum_{j=i+1}^{C} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2 \tag{7}$$

The mean Euclidean distance over all pairs of class centers.

Fisher ratio:

$$\mathrm{FR} = \frac{D_{\mathrm{inter}}}{D_{\mathrm{intra}}} \tag{8}$$

The ratio of the inter-class distance to the intra-class distance, measuring the separability of classes.

Silhouette coefficient:

$$s\left(i\right) = \frac{b\left(i\right) - a\left(i\right)}{\max\{a\left(i\right), b\left(i\right)\}} \tag{9}$$

$a(i)$ is the mean distance from sample i to the other samples of its own class, excluding i itself, and $b(i)$ is the smallest over the other classes of the mean distance from i to all samples of that class.

The final score is the mean over all samples.

The Fisher ratio and the silhouette coefficient are classical clustering metrics that directly measure the geometric cluster structure of representations.

Class correlation:

$$\mathrm{Corr}_{\mathrm{class}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} |\rho\left(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j\right)| \tag{10}$$

$$\rho\left(\boldsymbol{u}, \boldsymbol{v}\right) = \frac{(\boldsymbol{u} - \overline{\boldsymbol{u}}) \cdot (\boldsymbol{v} - \overline{\boldsymbol{v}})}{\|\boldsymbol{u} - \overline{\boldsymbol{u}}\|_2 \|\boldsymbol{v} - \overline{\boldsymbol{v}}\|_2} \tag{11}$$

Measures whether the representations of different classes are similar.

$\mathcal{P}$ denotes the set of all off-diagonal class pairs, and ρ is the Pearson correlation coefficient.

Task correlation:

$$\mathrm{Corr}_{\mathrm{task}} = \frac{1}{|\mathcal{T}|} \sum_{(t_a,t_b)\in\mathcal{T}} |\rho\left(\boldsymbol{\mu}_{t_a}, \boldsymbol{\mu}_{t_b}\right)| \tag{12}$$

Measures the similarity between the centers of different tasks (sets of classes); the sum runs over all unordered pairs of distinct tasks.

$\boldsymbol{\mu}_{t_a}$ is the combined center of all classes in task a.

Task overlap: measures whether different tasks use the "same set" of neurons. For task t, a KC is included in the active set $\mathcal{S}_t$ when it emits at least one spike on more than 5% of that task's test samples; this is a sample-response fraction rather than a firing rate.

$$\mathcal{S}_t = \left\{ j \,\middle|\, \frac{1}{N_t} \sum_{i\in\mathrm{Task_t}} \mathbf{1}[z_{\mathrm{ij}} > 0] > 0.05 \right\} \tag{13}$$

For all task pairs, compute the Jaccard index.

$$J\left(\mathcal{S}_a, \mathcal{S}_b\right) = \frac{|\mathcal{S}_a \cap \mathcal{S}_b|}{|\mathcal{S}_a \cup \mathcal{S}_b|} \tag{14}$$

Finally, average over all task pairs.

PCA dimension (90%):

$$k = \min\left\{ k^{'} \,|\, \sum_{i=1}^{k^{'}} \lambda_i \geq 0.90 \times \sum_{m=1}^{D} \lambda_m \right\} \tag{15}$$

Perform PCA on the activation matrix Z and compute the cumulative explained variance ratio; then find the smallest k such that the cumulative variance is $\geq 90\%$

B Alternative Rules and Group Sizes on the Odor Stream

We test whether the benefit of orthogonalization is specific to Sanger's rule at the group size used in the main experiments. We repeat the odor experiment (10 tasks $\times$ 11 classes, noise σ = 0.030; all other settings unchanged) with (i) a plain Oja rule (Oja, 1982), which performs PCA-like compression without any cross-unit orthogonalization, and (ii) Sanger's rule at group sizes G = 5, 10, and 20. The AAC values are reported with the other controls in Table 4.2 and are not repeated here.

Oja learning also improves over the fixed RP baseline in this condition, so the experiment does not isolate cross-unit orthogonalization as the sole cause of the gain. The grouped results are non-monotonic and variable across seeds; G = 5–10 has the highest means among the three tested sizes, but no unique optimum is established. Every tested configuration has a higher mean than the fixed RP baseline. Group size changes which residual terms each update can use, not how many connections a KC has: fan-in, weight norm and sparsity are held at the RP values in all conditions.

Table B.1: Effect of epoch-wise reshuffling of the groups (odor, 10×11, $\sigma = 0.030$, AAC in %).

Grouping scheme	G = 5	G = 10	G = 20
Reshuffled each epoch (default)	71.41 ± 6.75	71.98 ± 2.70	68.50 ± 8.42
Fixed partition	72.82 ± 5.79	66.61 ± 10.59	70.05 ± 2.46

Whether the groups are re-randomized at every epoch changes AAC by at most 5.4 points, comparable to the run-to-run variability at these group sizes (s.d. 2.5–10.6), and every variant remains above the RP baseline (60.61%). Reshuffling therefore shapes the expected form of the lateral interaction (Appendix D) without producing a reliable accuracy difference at the scales studied here; the local orthogonalization is effective with either grouping scheme.

C Neuronal Mechanisms and Parameters

C.1 Neuron model

All neurons in the network are leaky integrate-and-fire (LIF) units. In addition to the circuit topology described in Section 3.1, the model embeds two functionally complementary neural mechanisms: lateral inhibition and spike-frequency adaptation. Lateral inhibition is present in both the PN layer, where it is delivered by LNs, and in the KC layer, where it is delivered by the APL neuron. The KC layer additionally exhibits SFA and threshold homeostasis.

C.2 Lateral inhibition

Lateral inhibition is implemented by a feedforward sub-circuit within a processing layer. Excitatory signals from the input ORNs project not only to the corresponding PNs but also to the LNs. The LN population then inhibits the PN layer. In the model a single LN unit receives input from the ORN layer and projects to all PNs with a uniform weight, so the inhibition is global within the layer and carries no spatial structure; when a strong signal drives several PNs, the LN activation suppresses the weaker ones, and only the strongest inputs drive sustained spiking. The mechanism sharpens the representation of the input and makes the boundary between distinct input patterns more pronounced. The inhibitory current received by PN j is given by

$$I_{\mathrm{LI},j}^{\mathrm{PN}}(t) = \sum_{k} w_{\mathrm{LN}\to\mathrm{PN,jk}} T_k^{\mathrm{LN}}(t) \tag{16}$$

$$\frac{dT_k^{\mathrm{LN}}(t)}{dt} = -\frac{T_k^{\mathrm{LN}}(t)}{\tau_{\mathrm{trace,\ LN}}} + S_k^{\mathrm{LN}}(t) \tag{17}$$

$$S_k^{\mathrm{LN}}(t) = \sum_{l} \delta(t - t_{\mathrm{kl}}^{\mathrm{LN}}) \tag{18}$$

In the implementation the trace is advanced as $T \leftarrow e^{-\Delta t/\tau} T + s$ after each step, so that a spike adds one to the trace; the same scheme is used for the adaptation trace below. Here $I_{\mathrm{LI},j}^{\mathrm{PN}}(t)$ is the inhibitory current received by the j-th PN; $w_{\mathrm{LN}\to\mathrm{PN,jk}}$ is the synaptic weight from the k-th LN to the j-th PN, which is non-positive because the synapse is inhibitory; $T_k^{\mathrm{LN}}(t)$ is the spike trace of the k-th LN, which records its activity and decays over time; $\tau_{\mathrm{trace,\ LN}}$ is the decay time constant; $S_k^{\mathrm{LN}}(t)$ is the spike train of the k-th LN, whose spikes occur at times $t_{k,l}^{\mathrm{LN}}$; and δ denotes the Dirac delta function, used here as the mathematical representation of a spike.

C.3 SPIKE-FREQUENCY ADAPTATION

Spike-frequency adaptation is implemented as an intrinsic dynamical property of the PNs, LNs, and KCs rather than through circuit connectivity. When a neuron fires repeatedly under sustained excitatory input, every spike triggers an internal, slowly decaying inhibitory adaptation current that accumulates. This accumulating current reduces the neuron's excitability, so that a sustained stimulus drives it less strongly and the firing rate declines; the spiking threshold itself is unchanged by adaptation. The neuron therefore behaves like a high-pass filter or a change detector: it responds most strongly to the onset of, or change in, a signal, and progressively adapts to a constant background. The adaptation current is governed by

$$I^X_{\mathrm{SFA},j}(t) = w^X_{\mathrm{SFA}} A^X_j(t) \tag{19}$$

$$\frac{dA^X_j(t)}{dt} = -\frac{A^X_j(t)}{\tau^X_{\mathrm{SFA}}} + S^X_j(t) \tag{20}$$

$$S^X_j(t) = \sum_l \delta(t - t^X_{j,l}) \tag{21}$$

where $I^X_{\mathrm{SFA},j}(t)$ is the adaptation current of the j-th neuron in layer X, which accumulates recent spiking activity and decays with time constant τ^X_{SFA}; $w^X_{\mathrm{SFA}} \leq 0$ sets the strength of the inhibitory adaptation current; and $S^X_j(t)$ is the spike train of neuron j.

C.4 POPULATION SPARSITY IN THE KC LAYER

The KC layer employs a comparable lateral-inhibition mechanism with the APL neuron in place of the LN population: APL receives excitatory input from the KC population and returns feedback inhibition to it. The two inhibitory populations therefore differ in where they take their input, the LN population from the ORN layer and APL from the KCs. Together with SFA and threshold homeostasis, this feedback regulates the average firing rate of the KC layer towards the target used throughout the model (0.05 Hz), yielding the sparse population code that the readout layer exploits. This target is a sparsity level chosen for the model and not a measured biological rate. Threshold homeostasis is implemented as a per-neuron adjustment of the KC firing thresholds, applied after every training batch on the batch mean: each KC threshold is updated by

$\theta_j \leftarrow \mathrm{clip}(\, \theta_j + \eta_\theta \cdot (r_j - r^*), \theta_{\min}, \theta_{\max}\,)$,

where r_j is the mean firing rate of KC j over the batch and $r^* \equiv 0.05$ Hz the target rate, $\eta_\theta = 10^{-3}$ is the threshold-adaptation learning rate (Table C.1), and the clamp is $\theta_{\min} = 0.1$ with $\theta_{\max} = 3.0$ for the odor stream and 10.0 for the vision streams. The update applies to KC neurons only.

The group partition is redrawn during training, every epoch in the odor experiments and every batch in the vision experiments; the two pipelines were written independently and this interval is where they differ. We did not run a systematic study of the interval. The related comparison we ran, on the odor stream only, is redrawing the partition every epoch against fixing it once: it changes AAC by at most 5.4 points, comparable to the seed-to-seed spread of those runs (Table B.1). The vision pipelines were also run with the epoch interval and gave comparable accuracy; we did not record that sweep as a separate table.

C.5 PARAMETERS

The synthetic odor generator constructs 110 prototypes over 51 receptor coordinates. For each prototype it selects 30–50% of receptors and samples their affinity distances from a three-component Gaussian mixture whose fitted parameters are averaged from rows 20–65 of `fit_parameters.txt`; affinity is mapped by $\exp(-d^2/\mathbb{E}[D]^2)$ and normalized. A training access adds fresh Gaussian noise of standard deviation σ and clips to $[0, 1]$; the separate test examples use a fixed independent noise draw. The five seeds generate distinct prototypes. This is a synthetic benchmark, not measured fruit-fly odor responses.

Network and training parameters are summarized in Table C.1.

Input preparation. All inputs are min-max normalized to [0, 1] and then multiplied by the dataset-specific input scale (Table C.1), to set the circuit input current. The CIFAR-100 inputs are extracted with an ImageNet-pretrained ViT-B/16 (vit_base_patch16_224, 224×224 images) and taken as the 768-dimensional pooled pre-logits output. Inputs are injected as an analog current into the ORN layer, whose LIF dynamics produce the spikes; no explicit rate or latency encoding is used.

Table C.1: Network and training parameters. The upper table lists the settings that differ between the three input configurations; the lower table lists the parameters shared by all of them. Values describe the reported configurations. The RP baseline requires no pretraining: its PN–KC weights are sparse random connections (0 epochs).

Parameter	Odor	MNIST	CIFAR-100
ORN / PN units	51	784	768
KC units	2,000	15,680	15,360
PN → KC fan-in per KC	6	40	40
Expansion ratio (PN : KC)	$\approx 1:40$	$1:20$	$1:20$
MBON units	110	10	100
PN → KC initial weight	0.1	0.017	0.017
ORN bias current	0.072	0.072	0
Input scale	0.9	0.52	0.304
Gradient scaling factor	2×10^6	5×10^2	1×10^3
Batch size	44	256	256
Readout epochs per task	50	20	50
Phase-1 pretraining epochs	1500	1000	1500
Group-partition redraw	every epoch	every batch	every batch
Threshold clamp range	[0.1, 3.0]	[0.1, 10.0]	[0.1, 10.0]

Group	Shared parameter values
Neuron dynamics	membrane time constant τ_{m} = 10 ms (ORN, PN, LN, MBON), 4 ms (KC), 50 ms (APL); lateral-inhibition trace $\tau_{\mathrm{trace,\ LN}}$ = 5 ms; spike-frequency adaptation τ_{SFA} = 400 ms; initial firing threshold 0.8, except 0.49 for KC (the KC threshold is then adapted, Appendix C.4); simulation step 1 ms; stimulus / silence window 1000 / 100 ms
Circuit	LN and APL units 1/1; target L2 norm per KC $\sqrt{(\text{fan-in} \cdot \text{w}^2)}$; target KC mean firing rate 0.05 Hz
Plasticity	atan surrogate gradient; Hebbian learning rate 5×10^{-4}; threshold-adaptation learning rate 1×10^{-3}; readout learning rate 1×10^{-3} (weight decay 0); initial KC→MBON weight (max) 0.01; grouped-update group size G = 10; KC thresholds updated every batch
Protocol	Phase-1 training indices per epoch: 440 odor (four per class; noise resampled on access), 60,000 MNIST and 50,000 CIFAR-100; the separate odor test set has 880 samples (eight per class)

D GROUPED HEBBIAN UPDATE: IMPLEMENTATION AND SCOPE

The classical generalized Hebbian algorithm (GHA) (Sanger, 1989) motivates the feature stage but is not identical to the spiking implementation. PN spike counts are centered and whitened with an exponential moving estimate of the mean and covariance (decay 0.01, transform refreshed every 50 steps). For each training example the 5% most active KCs are eligible for the update. KCs are partitioned into groups of size $G = 10$; the partition is redrawn during training, every epoch in the odor experiments and every batch in the vision experiments (Appendix C). The update inside a group uses a running PN residual. For group order i,

$$\Delta \boldsymbol{w}_i \propto \mathbb{E}\left[y_i \left(\widetilde{\boldsymbol{x}} - \sum_{k<i} y_k \boldsymbol{w}_k \right) \right] - \lambda \mathbb{E}[y_i^2] \boldsymbol{w}_i, \tag{22}$$

where $\widetilde{\boldsymbol{x}}$ is whitened PN activity and λ is the implementation's norm-dependent self-decay coefficient. The code averages this quantity over a batch, scales it with a decaying learning rate, adds decaying Gaussian perturbations to dense shadow weights and clips those weights to be nonnegative. The weights used for a forward pass are formed by keeping the K largest *values* per KC (not the largest absolute magnitudes) and normalizing each retained vector to a target Euclidean norm. Thus the physical PN–KC map has the same fan-in and norm budget as RP; the shadow matrix retains potential future connections. The forward map is refreshed from the shadow matrix before the next batch, so the last shadow update need not appear in the last forward map of an epoch.

The group assignment has one exact combinatorial property. For two distinct KCs i and k in a uniformly random partition of M units into groups of G, the probability that they share a group is $(G-1)/(M-1)$. If their within-group order is also random, the probability that k precedes i is half of that. Consequently, the *indicator* that k enters i's residual has expectation

$$\mathbb{E}[\xi_{ik}] = \frac{G-1}{2(M-1)}. \tag{23}$$

This makes the pairwise interaction symmetric in expectation over new assignments. It does not make the realized update synapse-local, show that the spiking circuit converges to principal components, or prove global orthogonality or lower task interference. Online whitening, nonnegative clipping, top-K projection, active-KC selection and homeostatic thresholds alter the dynamics of classical GHA. We therefore use Eq. 2 as a reference form and rely on measured representations and accuracy for the claims in the main text.

E Odor Stream under a Second Partition

The main experiments use a fixed partition of each dataset and a fixed set of readout regularizers. The alternative partitions are reported in Table 4.3 and the regularization control in Table 4.4; this appendix adds the odor stream under the second partition across the same four noise levels (Table E.1).

Table E.1: Odor stream with the 5×22 partition at four noise levels (AAC, mean $\pm$ s.d. over five seeds).

Noise σ	Fixed RP AAC (%)	Learned map AAC (%)
0.025	81.11 ± 1.65	83.80 ± 4.94
0.030	63.64 ± 7.64	78.52 ± 9.63
0.035	57.80 ± 8.92	60.23 ± 10.38
0.040	47.41 ± 10.09	54.18 ± 9.90

Under this coarser partition the learned map is ahead at every noise level. The advantage narrows at the lowest and intermediate noise levels ($\sigma = 0.025$ and 0.035, ≤ 2.7 points) but remains positive, and both mappings degrade monotonically with noise: the qualitative ordering of the main experiments is therefore not an artifact of a particular task partition.