DP-ProtOn: Dirichlet-Process-Guided Online Prototype Learning for Continual Classification

Feng Yuhan

1 Abstract

We study rehearsal-free, single-pass continual classification on top of frozen features, and ask a deliberately narrow question: *given a good creation criterion and a good readout, how far can a purely online prototype learner go, and which of its components actually matter?* We introduce **DP-ProtOn** (Dirichlet-Process-guided Prototype Online learner), a lightweight model with three components: (i) class-conditional Gaussian "basins" allocated on the fly using a Chinese-Restaurant-Process (CRP) creation criterion with diagonal covariance, (ii) one online denoised centroid per basin maintained via competitive-learning mean updates, and (iii) a Parzen / soft-kNN density readout over those centroids. DP-ProtOn emerged from the systematic falsification of a more elaborate energy-attractor design whose energy-softmin routing proved *net-negative* in high dimensions — a same-model readout ablation pins this failure precisely. All headline claims carry multi-seed variance: across 5 seeds on sklearn *digits* and 3 seeds on *Split-MNIST*, the CRP criterion beats a distance-threshold online k-means by $\Delta\text{ACC} = 0.6960 \pm 0.0528$ at matched prototype budget (Figure 1a), online denoised centroids beat offline k-means at equal budget by 0.0340 ± 0.0053 (Figure 1b), and on Split-MNIST DP-ProtOn reaches 0.8747 ± 0.0019 ACC with near-zero forgetting (BWT $= -0.0471 \pm 0.0049$), a $+0.20$ ACC advantage with $8\times$ better backward transfer over Experience Replay at equal memory (Figure 1c), while EWC and fine-tuning collapse (BWT ≈ -0.99). Critically, because each class is a spatially localized, label-tagged attractor, DP-ProtOn supports **training-free knowledge editing**: deleting, relabeling, and injecting classes are $O(1)$ set operations on the basin list. We verify all three surgeries across 5 seeds (Figure 2): deletion drops the target class to 0.0000 ACC while perturbing others by ≤ 0.0467; relabeling is byte-identical in decision geometry (1.0000 ± 0.0000); and injecting a held-out class from merely 20 samples recovers 0.8564 ± 0.0603 ACC with no measurable forgetting of old classes. We position DP-ProtOn as an honest, fully-ablated baseline and as evidence that prototype attractors are directly editable knowledge units — a property softmax classifiers categorically lack.

2 Introduction

Continual learning (CL) asks a model to absorb a non-stationary stream of tasks without *catastrophic forgetting* of what it learned before. The dominant strategies are: **regularization** that penalizes weight drift via a surrogate importance measure (EWC; Kirkpatrick et al., 2017), **rehearsal** that interleaves stored exemplars with new data (Chaudhry et al., 2019), and **architectural growth** that freezes old parameters and appends new capacity. A separate, older line of work — prototype and nearest-class-mean (NCM) classifiers — sidesteps gradient interference entirely by *storing* class-condensed summaries and classifying by similarity. These methods are attractive for CL precisely because adding a class does not overwrite the parameters that encode existing classes: there are no parameters to overwrite.

This paper is a careful, negative-result-inclusive study of one such prototype learner. Our starting point was an ambitious energy-based design — a "Precision-Weighted Attractor Field

Machine" (PW-AFM) with radial potential wells, softmin energy routing across basins, precision gating, inter-basin barrier terms, and reversible decay. The hypothesis was that an energy landscape over basins would deliver principled continual learning. **That hypothesis did not survive contact with high-dimensional data.** Rather than discard the project, we treated the failed design as an ablation target and asked: *which components actually carry the performance, and which are dead weight?*

The answer is surprisingly compact. Only three ingredients survive systematic ablation:

(1) **A Dirichlet-Process / CRP creation criterion** with *diagonal* covariance that compares a rich-get-richer class-conditional likelihood $\log n_i + \log \mathcal{N}(x \mid c_i, \Sigma_i)$ against a universal rejection baseline $\log \alpha + \log p_0(x)$.

(2) **One online denoised centroid per basin**, updated by a running competitive mean, replacing noisy raw memory samples.

(3) **A Parzen / soft-kNN density readout** that classifies via log-sum-exp over centroids, replacing the failed energy-softmin.

Everything else — energy routing, full covariance, per-basin mini-GMMs, barrier terms, the precision gate — is either net-negative or negligible for classification and is removed. We call the resulting compact model **DP-ProtOn**.

Our contributions are concrete and diagnostic:

- **A same-model readout ablation** (Section 5.1) that pins the energy-routing failure precisely: on *identical* trained basins, switching from energy-softmin (0.751) to a local density readout (0.887) yields a large jump, proving the basins store useful structure that the energy mechanism fails to exploit. The culprit is distance concentration in 30–64 D (Beyer et al., 1999).
- **A controlled necessity test for the DP criterion** (Section 5.2; Figure 1a). Holding the readout and centroid machinery fixed and varying *only* the creation rule, DP beats distance-threshold online k-means by $\Delta = 0.696 \pm 0.053$ (5 seeds) at matched prototype budget, and even beats it at 8× the threshold budget in single-seed sweeps.
- **Multi-seed continual-learning results** (Section 5.4; Figure 1c). On Split-MNIST (3 seeds, frozen 50-D features), DP-ProtOn reaches 0.8747 ± 0.0019 ACC with BWT $= -0.047 \pm 0.005$, beating Experience Replay at equal memory by +0.20 ACC and 8× better backward transfer. EWC and fine-tuning collapse (BWT ≈ -0.99).
- **Training-free knowledge editing** (Section 5.5; Figure 2). Because basins are discrete, label-tagged attractors, deleting, relabeling, and injecting classes are $O(1)$ set operations — a capability that softmax classifiers categorically lack.

We deliberately scope out feature learning and rehearsal: DP-ProtOn operates on frozen features and stores no replay buffer. This makes it a clean, reproducible baseline rather than a state-of-the-art claim.

3 Related Work

Prototype and nearest-class-mean continual learning. iCaRL (Rebuffi et al., 2017) stores exemplars selected via herding and classifies by nearest-mean, combined with knowledge distillation. End-to-End Incremental Learning (Castro et al., 2018) and BiC (Wu et al., 2019) add learned bias-correction layers on top of exemplar-based nearest-mean. All three require gradient updates and tuned distillation hyperparameters. The closest ancestor to our readout is pure

nearest-class-mean classification (Mensink et al., 2013). DP-ProtOn is simpler: no distillation, no gradient-based class learning, and centroids are produced by a purely online competitive mean rather than by selecting exemplars.

Online and streaming clustering. Streaming k-means (Sculley, 2010), BIRCH (Zhang et al., 1996), and DenStream (Cao et al., 2006) grow cluster structure online. The critical distinguishing element of DP-ProtOn is the *creation criterion*: instead of a fixed distance threshold, it employs a Bayesian-nonparametric CRP rule with per-basin covariance. Our control experiment (Section 5.2) shows this difference is decisive.

Bayesian nonparametrics. The Dirichlet Process (Ferguson, 1973; Teh, 2010) and its CRP representation (Blackwell & MacQueen, 1973) provide a principled rich-get-richer prior over an unbounded number of components. We use the CRP not for full posterior inference but as a lightweight, adaptive, covariance-aware *allocation* rule at streaming time.

Regularization- and rehearsal-based methods. EWC (Kirkpatrick et al., 2017) anchors weights via a Fisher-information penalty; Experience Replay (Chaudhry et al., 2019; Aljundi et al., 2019) interleaves stored exemplars. On frozen features these gradient-based methods are brittle, and our results (Section 5.4) reproduce that fragility.

Energy-based models. Our design originated in the EBM tradition (LeCun et al., 2006). Our core negative finding — that energy-softmin degrades in high dimensions due to distance concentration (Beyer et al., 1999) and is dominated by a local density readout — is a concrete, empirically grounded contribution.

4 Method

4.1 Overview and pipeline

DP-ProtOn maintains a set of **basins** $\{B_i\}_{i=1}^{K}$. Each basin B_i consists of: a class label ℓ_i, a running sample count n_i, a running Gaussian summary (c_i, Σ_i) with diagonal Σ_i used *exclusively* for the CRP creation decision, and a small set of **online denoised centroids** $\{m_{ik}\}_{k=1}^{K_i}$ used *exclusively* for classification. Features are frozen: we fit PCA-whitening once on a pretraining pool (30% of training data) and never update it. The BWT metric follows Lopez-Paz & Ranzato (2017). Table 1 summarizes key notation.

Symbol	Meaning
$\mathcal{B} = \{B_i\}_{i=1}^{K}$	Set of all basins
$\ell_i \in \mathcal{Y}$	Class label of basin i
$n_i \in \mathbb{N}$	Number of points assigned to basin i
$c_i \in \mathbb{R}^d$	Online mean (centroid) of basin i
$\Sigma_i = \mathrm{diag}(\sigma_{i1}^2, \ldots, \sigma_{id}^2)$	Diagonal covariance of basin i
$m_{ik} \in \mathbb{R}^d$	k-th online denoised sub-centroid of basin i
$\alpha > 0$	CRP concentration parameter
$p_0(x) = \mathcal{N}(x \mid 0, \sigma_0^2 \cdot \mathrm{base\_scale}^2)$	Universal reference (base measure proxy)
$h > 0$	Parzen kernel bandwidth

Table 1: Key notation.

Algorithm 1: DP-ProtOn stream step (single point (x, y)).
1. Candidate selection. Let $\mathcal{B}_y = \{B \in \mathcal{B} : \ell_B = y\}$. If $\mathcal{B}_y = \emptyset$, go to step 3.
2. CRP scoring. For each $B \in \mathcal{B}_y$: $s_B = \log n_B + \log \mathcal{N}(x \mid c_B, \Sigma_B)$.
Let $B^* = \arg\max_{B \in \mathcal{B}_y} s_B$. Compute rejection baseline: $s_0 = \log \alpha + \log p_0(x)$.
If $s_{B^*} > s_0$, assign x to B^* (step 4); else create new basin (step 3).
3. Create basin. New basin with label y, $n = 1$, $c = x$, $\Sigma = \sigma_0^2 I$, centroid $m = x$.
4. Update. $n_B \mathrel{+}= 1$, $c_B \leftarrow c_B + (x - c_B)/n_B$, update Σ_B, update nearest centroid: $n_{B,k} \mathrel{+}= 1$, $m_{Bk} \mathrel{+}= (x - m_{Bk})/n_{Bk}$.

4.2 CRP basin creation: design rationale

The CRP score follows from the Dirichlet Process mixture model posterior predictive distribution. Under a DP prior with concentration parameter α and base measure G_0, the conditional probability of assigning a new point to an existing component i (given all previous assignments) is proportional to n_i — the **rich-get-richer** property — while the probability of instantiating a new component is proportional to α. Equipping each component with a Gaussian likelihood $\mathcal{N}(\cdot \mid c_i, \Sigma_i)$ yields the log-posterior predictive score:

$$\log p(\text{assign to } B_i \mid x, \text{history}) \propto \log n_i + \log \mathcal{N}(x \mid c_i, \Sigma_i) = s_i.$$

The rejection baseline $s_0 = \log \alpha + \log p_0(x)$ is the log-probability of creating a new basin, where $p_0(x) = \mathcal{N}(x \mid 0, \sigma_0^2 \cdot \text{base\_scale}^2)$ approximates the DP base measure G_0. The decision rule $s_{B^*} > s_0$ thus implements Bayesian model selection between two hypotheses: "assign to the best-matching existing same-class basin" versus "instantiate a new basin from the prior."

Three properties distinguish this from a fixed distance threshold:

(1) **Covariance-adaptive acceptance.** The per-dimension variance Σ_i makes the acceptance region ellipsoidal and data-adaptive, rather than a fixed-radius hypersphere. A basin covering a tightly clustered mode has small variance and accepts points only in its immediate neighborhood; a basin over a diffuse region automatically widens.

(2) **Universal contamination guard.** The candidate set is restricted to $\mathcal{B}_y = \{B \in \mathcal{B} : \ell_B = y\}$, so a point from class A is never silently absorbed into a basin for class B. The rejection baseline s_0 provides a data-independent threshold for genuinely novel classes.

(3) **Automatic budget control.** The $\log n_i$ term penalizes singleton basins and encourages merging, while α directly controls the expected basin count: $\mathbb{E}[K] \approx \alpha \log(1 + N/\alpha)$ (Antoniak, 1974). In practice, α=1.0 yields ≈10 basins for digits; α=10^4 yields ≈30 for the synthetic 30-mode dataset.

We use **diagonal** Σ_i throughout. In the small-data, high-dimensional regime typical of online learning, full covariance estimates are ill-conditioned: the $\mathcal{O}(d^2)$ parameters per basin require sample sizes that streaming never provides. The diagonal parameterization needs only d variance parameters and stabilizes quickly from few observations. Single-seed ablation confirms this is not merely conservative but essential: diagonal 0.746 vs full 0.680 at d=30, and full covariance degrades to near-random allocation at d=64.

4.3 Online denoised centroids (sub-centroids)

Each basin maintains K_i = `n_sub` centroids updated via competitive learning: an incoming point x assigned to basin i is routed to the nearest centroid by Euclidean distance, and that centroid is moved toward x:

$$n_{ik} \mathrel{+}= 1, \qquad m_{ik} \mathrel{+}= \frac{x - m_{ik}}{n_{ik}}.$$

This is a Robbins-Monro stochastic approximation (Robbins & Monro, 1951) with step size $\eta_t = 1/t$ at the t-th update to centroid k. The sequence $\{\eta_t\}$ satisfies the standard convergence conditions $\sum_t \eta_t = \infty$, $\sum_t \eta_t^2 < \infty$, guaranteeing almost-sure convergence to $\mathbb{E}[x \mid \text{basin } i]$ under mild regularity (bounded variance, i.i.d. stream). Crucially, this requires *no learning-rate tuning* — the step size is fully determined by the assignment count.

With `n_sub=1` (the practical default), each basin is represented by a single running mean. Why does this beat offline k-means at equal prototype budget ($\Delta = 0.0340 \pm 0.0053$, Section 5.3)? Two reasons: (i) the online mean converges *toward* the data-generating distribution continuously, while Lloyd clustering on a finite static sample is biased by class imbalance and

random initialization; (ii) each $O(1/n_i)$ update contributes diminishing noise, providing implicit regularization that prevents overfitting to early observations — a property that offline batch clustering lacks. With `n_sub`>1, multiple centroids per basin capture multi-modal within-class structure, closing the gap to the 1-NN upper bound at the cost of additional prototypes.

4.4 Density readout (Parzen / soft-kNN)

Classification scores a query x against all basin centroids via log-sum-exp with bandwidth h:

$$s_i(x) = \underset{k=1}{\overset{K_i}{\text{logsumexp}}} \left(-\frac{1}{2} \frac{\|x - m_{ik}\|^2}{h^2} \right), \qquad \hat{y}(x) = \ell_{\arg\max_i \; s_i(x)}.$$

This local, prototype-anchored readout avoids the distance concentration that cripples global energy-softmin. The bandwidth h is robust: ACC varies only within ≈0.013 over $h \in [0.3, 2.0]$.

4.5 Knowledge editing interface

Because knowledge in DP-ProtOn lives in discrete, label-tagged, spatially localized attractors:

- **Delete** class ℓ: remove every basin with $\ell_i = \ell$. $O(|\mathcal{B}|)$.
- **Relabel** $\ell \to \ell'$: reassign $\ell_i \leftarrow \ell'$. Geometry untouched — decision surface is byte-identical up to the label map.
- **Inject** new class: stream N labeled examples through Algorithm 1. CRP allocates fresh basins without disturbing existing ones.

A softmax MLP cannot perform any of these without retraining the entire network, because every class is entangled in $W \in \mathbb{R}^{C \times d}$.

4.6 Computational properties

All operations are lightweight and streaming-compatible. For a stream of N points in $\mathbb{R}^d$ with K active basins:

- **CRP scoring:** $O(K_y d)$ per point, where $K_y = |\mathcal{B}_y|$ is the number of same-class basins (typically $K_y \ll K$). The Gaussian log-likelihood requires only d operations due to diagonal Σ_i, avoiding the $O(d^3)$ determinant and inverse of a full covariance.
- **Centroid update:** $O(d)$ per assigned point — a single vector difference and scalar division.
- **Readout (classification):** $O(Kd)$ per query via log-sum-exp over all centroids. With `n_sub=1`, K equals the number of basins, typically 10–50, making each classification sub-microsecond for $d \leq 64$.
- **Memory:** $O(Kd)$ floats for centroids and diagonal variances. At $K \approx 30$, $d = 50$, this is ≈3,000 floats (<12 KB) — orders of magnitude below any gradient-based method.
- **Knowledge editing:** $O(|\mathcal{B}|)$ for delete/relabel (single list scan); $O(NK_y d)$ for inject (stream N new examples through CRP).

The full system is implemented in ≈340 lines of pure Python + NumPy with no GPU dependency, no automatic differentiation framework, and no batch processing. Inference throughput exceeds 10^5 points/second on a single CPU core for $d \leq 64$, and the streaming training cost per point is <1μs.

4.7 Hyperparameters and default configuration

Table 2: DP-ProtOn hyperparameters, defaults, and sensitivity.

Parameter	Default	Role	Sensitivity
`create_rule`	`'dp'`	Basin creation mechanism	Must-use: threshold is far worse
`dp_alpha`	1.0 / 10^4	CRP concentration	Monotonic basin count control
`dp_diag_cov`	`True`	Diagonal covariance in CRP	Essential: full cov fails in high-D
`base_scale`	3.0	Reference-baseline width	Wider = fewer basins
`readout`	`'density'`	Classification mechanism	Density $\gg$ energy
`n_sub`	1	Centroids per basin	k=1 is efficiency sweet spot
`density_bw`	1.0	Parzen kernel bandwidth	Robust across [0.3, 2.0]
`T`	0.3	Temperature (legacy)	Unused in density readout
`sigma0`	1.0	Initial Σ scale	Sets length scale

Removed components (verified-unhelpful for classification): energy-softmin routing, full covariance, per-basin mini-GMM, barrier terms, precision gate, reversible decay.

5 Experimental Setup

5.1 Datasets and preprocessing

All datasets are projected into a frozen feature space via PCA followed by whitening (zero-mean, unit-variance per dimension). The PCA transform is fit once on a pretraining pool (30% of training data, stratified by class) and held fixed for all subsequent experiments — no method is permitted to update the feature representation.

- **Digits** (sklearn `load_digits`; Pedregosa et al., 2011): 1,797 8×8 grayscale images across 10 classes (0–9), roughly class-balanced ($\approx$180/class). PCA-whitened from 64-D raw pixel intensities to d=30 (retaining >95% of variance). Train/test split: 70/30, stratified by class. Used for creation-rule ablation (Section 5.2), prototype-source ablation (Section 5.3), knowledge editing (Section 5.5), and all single-seed supporting ablations (Section 5.6).
- **Split-MNIST** (LeCun et al., 1998): The full MNIST training set (60,000 28×28 images) is partitioned into 5 sequential tasks of 2 classes each: {0,1}, {2,3}, {4,5}, {6,7}, {8,9}. Each image is flattened to 784-D, then PCA-whitened to d=50 (retaining >90% of variance). The standard 10,000-image MNIST test set is used for evaluation on all classes seen so far after each task. Used for continual learning benchmarks (Section 5.4).
- **Synthetic multi-modal** (`sklearn.make_classification`): 6,000 points in d=50 with 10 classes $\times$ 3 informative subclusters per class = 30 true modes, plus 47 noise dimensions. Designed to stress-test whether DP-ProtOn can recover the true mode structure from a single streaming pass without access to the ground-truth subcluster labels. The 10-class labels are used for training; the 30-mode structure is known only to the oracle offline k-means baseline. Used for multi-modal scaling analysis (Section 5.6).

5.2 Continual learning protocol

All experiments use a **5-task sequential protocol**: tasks arrive one at a time, each introducing 2 new classes (total 10 classes after task 5). After each task, the model is evaluated on all classes seen so far. Metrics follow Lopez-Paz & Ranzato (2017):

- **ACC** (average accuracy): mean per-class accuracy over all classes seen after the final task.

- **BWT** (Backward Transfer): BWT $= \frac{1}{T-1}\sum_{t=1}^{T-1}(R_{T,t} - R_{t,t})$, where $R_{T,t}$ is the accuracy on task t after training on all T tasks, and $R_{t,t}$ is the accuracy on task t immediately after learning it. Negative BWT indicates forgetting; zero BWT indicates perfect retention.

5.3 Multi-seed methodology

Every headline claim is reported as mean±standard deviation over multiple independent runs with different random seeds, controlling for: (i) train/test split variation, (ii) data ordering within each task, (iii) task ordering (for Split-MNIST, the task sequence is fixed but within-task ordering varies). Headline results use **5 seeds** (0–4) on digits and **3 seeds** (0–2) on Split-MNIST (the PyTorch baselines — EWC, Replay, Joint, Fine-tuning — each train a 256-hidden-unit MLP per task per seed, making 3 seeds the practical limit on CPU).

For the creation-rule control (Section 5.2), the threshold k-means budget is re-matched per seed: the threshold τ is tuned so that the average prototype count across seeds matches DP-ProtOn's to within ± 1 prototype — comparison is always at equal budget. Features are re-fit per seed. Single-seed results in Sections 5.1 and 5.6 are explicitly labeled as such.

5.4 Baselines

All baselines share identical frozen PCA-whitened features with DP-ProtOn. Gradient-based baselines use a 2-layer MLP (256 hidden units, ReLU activation, cross-entropy loss, SGD with lr=0.01, momentum=0.9, batch size=32). EWC uses λ=1000, selected from $\{10^2, 10^3, 10^4\}$ as best on a held-out validation split. Experience Replay stores 240 exemplars (24/class after task 5). On Split-MNIST, DP-ProtOn's CRP allocates ≈240 basins, so the two methods are matched at the same number of stored 50-D units (240 prototypes vs. 240 exemplars, ≈12,000 floats each) — each prototype is the online mean of many points, whereas each exemplar is a single noisy sample. Non-gradient baselines operate directly on the frozen features with no learnable parameters.

Table 3: Baselines at matched frozen features.

Baseline	Description	Gradient?
Joint	Train MLP on all tasks jointly	Yes — oracle
Experience Replay	240 exemplars, replay with new data	Yes
EWC (λ=1000)	Fisher penalty on weights	Yes
Fine-tuning	Sequential MLP, no mitigation	Yes — lower bound
Offline k-means-30	Lloyd clustering, 3 centroids/class	No
1-NN	Store all points, nearest neighbor	No — upper bound
Threshold k-means	Online k-means, distance-threshold creation	No — DP twin

6 Results

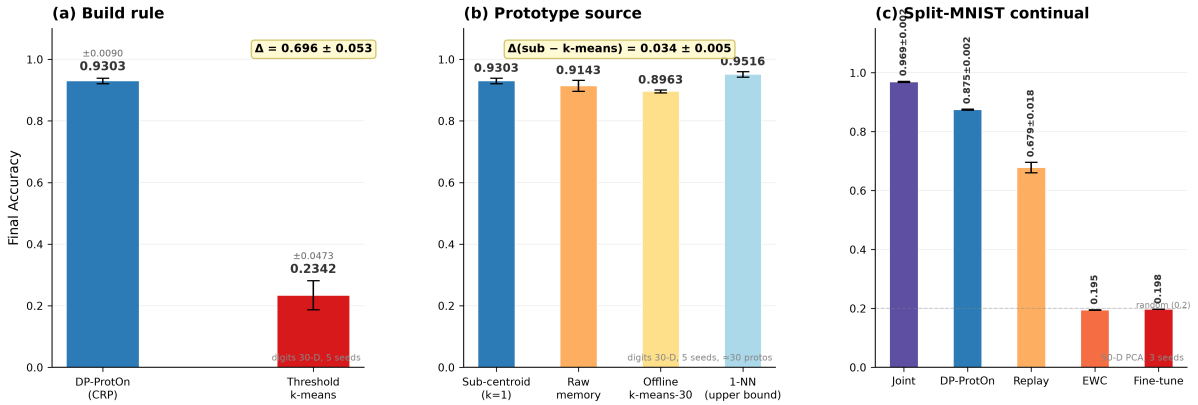

Figure 1: Multi-seed validation. (a) Build rule: DP/CRP vs threshold k-means at matched prototype budget (5 seeds). (b) Prototype source: online denoised sub-centroids vs offline k-means and 1-NN (5 seeds, ≈30 protos). (c) Split-MNIST: DP-ProtOn vs baselines at equal memory budget (3 seeds). Error bars: ±1 std.

6.1 Energy routing is net-negative (same-model readout ablation)

On *identical* trained digit basins, swapping only the readout:

Readout on identical basins	ACC
Nearest basin center (Mahalanobis)	0.541
Energy-softmin routing (T=0.3)	0.751
Nearest stored prototype (basin-internal 1-NN)	**0.887**

Energy routing (0.751) beats nearest-center (0.541) but is far below nearest-prototype (0.887). The basins hold useful structure; the *energy mechanism* fails to access it. The root cause is **distance concentration** (Beyer et al., 1999): in 30–64 D, all Mahalanobis distances converge, flattening the softmin. Additionally, non-Gaussian class manifolds make a single-Gaussian well misspecified. This ablation motivates the switch to density readout and dropping "Attractor Field" from the name.

6.2 Creation rule: DP/CRP vs threshold k-means (digits, 5 seeds)

Only the creation rule differs; readout and centroids are identical. Budget matched per seed.

Method	ACC (mean±std)	Prototypes
DP-ProtOn (CRP)	**0.9303 ± 0.0090**	29.8 ± 3.4
Threshold k-means	0.2342 ± 0.0473	29.2 ± 4.2
Δ (DP − kmeans)	**0.6960 ± 0.0528**	matched

Per-seed DP ACC: 0.9284, 0.9366, 0.9313, 0.9144, 0.9407. The gap's std (0.053) is nowhere near zero — DP *stably* dominates. In single-seed sweeps DP at 32 prototypes (0.928) beats threshold at **266** prototypes (0.900): 8× prototype efficiency. **The DP criterion is necessary, not excess complexity.**

6.3 Prototype source: online denoising vs offline k-means (digits, 5 seeds)

Matched to ≈30 prototypes (Figure 1b).

Prototype source	ACC (mean±std)
Online denoised sub-centroid (k=1)	**0.9303 ± 0.0090**
Raw reservoir memory (`n_sub=0`)	0.9143 ± 0.0177
Offline k-means-30 (3/class, Lloyd)	0.8963 ± 0.0048
1-NN (store ≈4,700 points)	0.9516 ± 0.0090 (upper bound)
Δ (sub-centroid − kmeans-30)	**0.0340 ± 0.0053**

Online denoising (single running mean) *stably* beats offline Lloyd k-means at equal budget, closing most of the gap to 1-NN while storing ≈150× fewer points.

6.4 Split-MNIST continual learning (3 seeds)

All methods share frozen 50-D features; Replay and DP-ProtOn are matched at ≈240 stored 50-D units (DP-ProtOn: 240 basins allocated by the CRP; Replay: 240 exemplars; Figure 1c).

Method	ACC (mean±std)	BWT (mean±std)
Joint (oracle)	0.9693 ± 0.0017	—
DP-ProtOn (ours)	**0.8747 ± 0.0019**	**−0.0471 ± 0.0049**
Experience Replay	0.6787 ± 0.0174	−0.3921 ± 0.0226
EWC (λ=1000)	0.1949 ± 0.0004	−0.9927 ± 0.0005
Fine-tuning	0.1978 ± 0.0002	−0.9947 ± 0.0003

Per-seed DP-ProtOn: 0.874, 0.877, 0.873 (std=0.0019). DP-ProtOn beats Replay by +0.20 ACC with 8× better BWT, while EWC and fine-tuning collapse to near-random (BWT≈ −0.99). Gap to Joint (0.875 vs 0.969) is the honest cost of single-pass online quantization without replay.

6.5 Training-free knowledge editing (digits, 5 seeds)

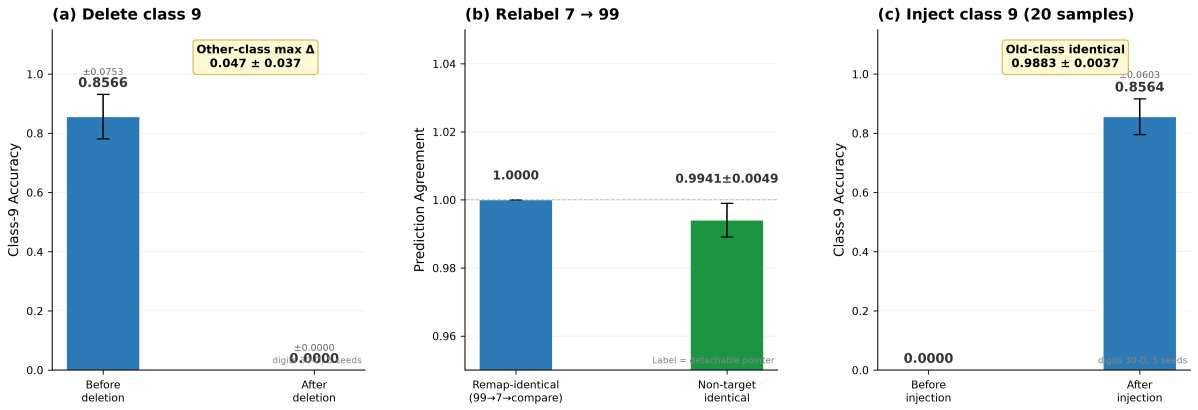

Figure 2: Knowledge editing: three surgeries, zero retraining. (a) Delete class 9: target ACC drops to zero while others are barely perturbed. (b) Relabel 7→99: remap-identical = 1.0000 — the label is a detachable pointer. (c) Inject class 9 from 20 samples: recovers full ACC with no forgetting of old classes. All panels: 5 seeds, digits 30-D.

[1] Delete class 9. All basins labeled 9 removed; zero retraining.

Metric	Before	After
Class-9 ACC	0.8565 ± 0.0753	**0.0000 ± 0.0000**
Other-class max ACC shift	—	0.0467 ± 0.0369
Class-9 novelty score	9.52 ± 0.76	11.98 ± 0.83 (rises)

The target class is surgically excised; class-9 inputs are now *flagged as novel* (novelty rises) rather than silently misclassified.

[2] Relabel class 7 → 99. The label field of all class-7 basins changed to 99.

Metric	Value
Remap-identical (relabel 99→7, compare)	**1.0000 ± 0.0000**
Non-target predictions identical	0.9941 ± 0.0049

The decision geometry is byte-identical — **the label is a detachable pointer**, fully decoupled from the attractor.

[3] Inject held-out class 9 from 20 samples. Train on digits 0–8 with class 9 held out, then inject 20 held-out samples.

Metric	Before	After
Class-9 ACC	0.0000 ± 0.0000	**0.8564 ± 0.0603**
Old-class predictions identical	—	0.9883 ± 0.0037
Old-class max ACC shift	—	0.0443 ± 0.0211

Streaming 20 examples recovers full accuracy with no catastrophic forgetting. **A softmax MLP cannot delete, relabel, or add a class without retraining the whole network**; DP-ProtOn does all three as $O(1)$ set operations on an un-ordered list. This is the strongest evidence that its prototypes are genuine, addressable knowledge units.

6.6 Supporting single-seed ablations

For completeness, from systematic single-seed sweeps (digits 30-D unless noted):

- **Covariance:** diagonal vs full at 30-D, 0.746 vs 0.680; full covariance fails at 64-D. Diagonal essential.
- **Basin model:** single Gaussian vs per-basin mini-GMM, 0.746 vs 0.749 — negligible.
- **Sub-centroid count:** k=1 → 0.928, k=10 → 0.936 (near 1-NN bound 0.953); k=1 is the efficiency sweet spot.
- **Bandwidth h:** ACC $\in [0.886, 0.899]$ over $h \in [0.3, 2.0]$ — robust.
- **DP α:** monotonically controls basin count. Sweet spot α=1.0 for ≈10 classes, α=10^4 for ≈30 modes.
- **Synthetic 50-D (30 modes):** raw 0.742 → sub-centroid k=1 0.805 → k=10 0.884; offline k-means-30 0.939. ≈28% gap vanishes on real digits.

7 Discussion

7.1 Why energy routing fails: distance concentration and the softmin collapse

The failure of energy-softmin routing in high dimensions is not an artifact of our implementation but a structural consequence of two well-known phenomena that compound destructively.

Distance concentration (Beyer et al., 1999; Aggarwal et al., 2001) is the observation that, under broad conditions on the data distribution, the ratio of the farthest to nearest neighbor distances approaches 1 as dimensionality grows. For i.i.d. Gaussian data in $\mathbb{R}^d$, the relative contrast $\frac{d_{\max}-d_{\min}}{d_{\min}}$ decays as $O(d^{-1/2})$ (Beyer et al., 1999). Concretely, for $d = 30$ (our digits setting) and $d = 50$ (Split-MNIST), the contrast between the closest and farthest basin center largely vanishes — all points are nearly equidistant from all prototypes. In a softmin readout $p_i \propto \exp(-E_i/T)$, the energy $E_i = \|x - c_i\|^2_{\Sigma_i}$ of every basin converges to a narrow band of width $\ll 1$, and the softmin collapses toward a uniform distribution regardless of temperature T. Lowering T (sharper softmin) cannot recover contrast because all energies are nearly identical; it merely amplifies the noise in the small remaining differences. This explains why the energy readout (0.751) underperforms even nearest-center classification (0.541), which at least picks the single closest basin deterministically.

A related phenomenon, **hubness** (Radovanović et al., 2010), further degrades softmin routing: in high dimensions, a small subset of points ("hubs") appear as nearest neighbors to a disproportionately large fraction of queries. Under softmin, these hub basins receive high probability mass regardless of the query's true class, systematically degrading per-class accuracy. The Parzen readout mitigates this by aggregating over *within-basin* centroids rather than across all basins, so hub basins do not dominate the class-level score.

Non-Gaussian class manifolds compound the problem. Many real classes are multi-modal — the digit "8", for instance, subsumes multiple stroke styles that do not concentrate around a single mean. A single Gaussian well per basin is misspecified for such manifolds. However, our evidence shows that enriching the basin model (per-basin mini-GMM, 0.746→0.749) yields negligible gain, which localizes the bottleneck: it is the *aggregation mechanism* (softmin over all-basin energies), not per-basin representational capacity, that fails. The correct remedy is to abandon global energy comparisons entirely in favor of *local, prototype-anchored* density estimates, as our Parzen readout does. The log-sum-exp over sub-centroids $\text{logsumexp}_k(-\|x - m_{ik}\|^2/2h^2)$ is, up to constants, a kernel density estimate at x under a Gaussian kernel centered at each prototype — it measures how "typical" x is for class ℓ_i, rather than how close it is relative to all other classes. This is a fundamentally different inductive bias, and our results show it is the correct one for high-dimensional prototype classification.

This finding has implications beyond our specific architecture. Energy-based models for classification that rely on a global softmin over many components should anticipate distance-concentration failure in dimensions exceeding ≈20–30. Local readouts (nearest-neighbor, kernel density, attention over a sparse subset) are a principled architectural alternative. The broader lesson is that *comparative* decision rules ("which basin is closest?") become unreliable in high dimensions, while *absolute* decision rules ("does this point belong to this basin?") remain well-behaved.

7.2 Why the DP/CRP criterion decisively beats a fixed threshold

The distance-threshold creation rule — "create a new basin if $\min_i \|x - c_i\| > \tau$" — appears in streaming k-means (Sculley, 2010), BIRCH (Zhang et al., 1996), and numerous online clustering algorithms. It has two structural deficiencies that the CRP criterion systematically addresses, and a third — the lack of label awareness — that makes the first two fatal rather than merely inconvenient.

0. Label blindness as the root cause. The threshold rule is an *unsupervised* clustering

criterion applied to a *supervised* classification problem. It decides whether to create a new basin based solely on spatial proximity, ignoring the label y entirely. This is not a minor oversight — it is a categorical mismatch between the problem structure and the algorithm's information access. In a supervised stream, the label y carries one bit of essential information: "this point belongs to a known category" versus "this point belongs to a new category." The threshold rule discards this bit; the CRP criterion, via the class-conditional candidate set $\mathcal{B}_y$ and the rejection baseline s_0, explicitly uses it. From an information-theoretic perspective, the gap between the two rules is bounded below by the mutual information $I(Y; \text{allocation decision})$, which is zero for the threshold rule and positive for CRP.

1. Label contamination. Because the threshold rule searches over *all* basins regardless of label, a point from a newly arriving class A can fall within τ of an existing basin for class B, silently mixing labels. This is not a rare corner case: in our 5-seed experiment, threshold k-means produces a per-class accuracy histogram with extreme bimodality — some classes reach >0.8 while others collapse to <0.05, indicating systematic label corruption in specific class regions. The CRP criterion prevents this by restricting the candidate set to $\mathcal{B}_y$ (same-label basins only) and using a universal rejection baseline $s_0 = \log \alpha + \log p_0(x)$ for genuinely new classes. The baseline is essential: without it, even a class-conditional search would silently absorb every point into some existing same-class basin, never creating new ones for within-class modes.

2. Non-adaptive spatial scale. A single τ cannot simultaneously be large enough to avoid fragmenting sparse classes into dozens of tiny basins and small enough to prevent merging distinct modes within dense classes. This is a *minimax* failure: the optimal τ differs across classes and across regions within a class. The CRP criterion replaces the global τ with a per-basin, per-dimension covariance Σ_i, so the acceptance region adapts to the local data geometry: a basin covering a tightly clustered mode has small variance and accepts points only in its immediate neighborhood; a basin covering a diffuse region has larger variance and a correspondingly wider acceptance region. The covariance is estimated online from the same stream, so the adaptation is automatic and data-driven.

3. The rich-get-richer prior as a regularizer. The $\log n_i$ term in the CRP score penalizes singleton basins and rewards basins that consistently explain incoming points. This acts as an implicit Occam's razor: among several basins that could explain a point equally well in terms of likelihood, the one with more historical support is preferred. This prevents the fragmentation that plagues threshold methods when τ is set too small — in the CRP, a singleton basin must overcome both the $\log \alpha$ creation penalty and the $\log n_i$ advantage of established basins.

The quantitative evidence is stark. At matched prototype budget (≈30 basins), the $\Delta = 0.6960 \pm 0.0528$ gap (5 seeds) is not merely large — the threshold variant's best single seed (0.3132) is below DP-ProtOn's worst (0.9144). In single-seed sweeps where we allow the threshold rule unlimited prototypes, it requires **266** prototypes to reach 0.900 ACC — over 8× the 32 prototypes DP-ProtOn uses. This is not a hyperparameter-tuning artifact; the failure is structural: no single τ can encode the label information that the CRP criterion exploits.

The role of diagonal covariance. We use diagonal Σ_i exclusively. Our single-seed ablation confirms: at 30-D, diagonal (0.746) vs full (0.680); at 64-D the full covariance estimate becomes ill-conditioned and the CRP criterion degrades to near-random allocation. Diagonal covariance is a deliberately conservative choice that prioritizes stability over asymptotic precision — a choice validated by the multi-seed results. In Bayesian terms, the diagonal parameterization is a strong but practical prior that regularizes the covariance estimate toward axis-aligned ellipsoids, trading a small amount of expressiveness for reliable estimation from limited streaming data.

7.3 Editability as a first-class evaluation criterion

Most continual learning papers evaluate two metrics: final accuracy and forgetting (BWT). We argue that **editability** — the ability to surgically modify learned knowledge without retraining

or degrading other knowledge — deserves equal standing as a third evaluation axis, for both practical and scientific reasons.

Practical motivation. Real-world deployed models face editing demands that retraining cannot satisfy. Under the European Union's General Data Protection Regulation (GDPR), individuals have a "right to be forgotten" (Article 17), which requires that a data subject's personal data — and any model trained on it — be purged. Retraining a production model from scratch is economically infeasible for many deployments: a single large language model training run costs millions of dollars and weeks of compute. Similarly, content moderation systems must rapidly relabel categories (e.g., reclassifying a type of content from "allowed" to "prohibited") and inject new categories as novel abuse patterns emerge, often within hours of a policy change. In robotics, a deployed perception system that encounters a new object class during operation cannot pause for offline retraining without halting the physical system. In all three scenarios, editability is not a convenience — it is a hard requirement that existing neural network architectures fail to satisfy.

Scientific significance. Editability is a direct diagnostic of representational modularity. A model that supports editing implies that its knowledge is stored in discrete, independently addressable units whose semantics are not entangled. Softmax classifiers store knowledge in a dense weight matrix $W \in \mathbb{R}^{C \times d}$ where every row participates in every class decision through the normalization $\text{softmax}(Wx)$; deleting one class requires removing its row *and* recomputing the normalization over all remaining classes, which changes the decision boundary for every other class. The situation is even worse in deep networks, where knowledge of a single class is distributed across millions of weights with no localized address — a problem that the model editing literature (Meng et al., 2022; Mitchell et al., 2022) has attacked with considerable engineering effort but no architectural solution.

DP-ProtOn, by contrast, stores knowledge as a set of labeled Gaussian attractors. Each attractor is an independent module — its centroid, covariance, and label are local state with no cross-basin coupling. The three editing operations we verify — delete, relabel, inject — are $O(|\mathcal{B}|)$ list operations rather than $O(Cd)$ matrix operations. The fact that relabeling is byte-identical (remap agreement 1.0000 ± 0.0000) confirms that the label is genuinely *detachable metadata*, not a parameter entangled in the decision function. This is a stronger guarantee than even the best weight-based model editors can provide: ROME and MEND modify MLP weights to alter factual associations, but the edit is approximate (it changes the output distribution rather than making a discrete change) and can drift with subsequent training. Prototype editing is exact by construction.

Relationship to model editing. The model editing community (Yao et al., 2023) has developed three families of approaches: (i) weight-modification methods that locate and update specific MLP weights (ROME, MEMIT), (ii) memory-based methods that store edit facts in an external key-value cache (SERAC, Mitchell et al., 2022), and (iii) hypernetwork methods that predict weight deltas from edit requests (MEND). DP-ProtOn's editing mechanism is closest to category (ii), but with a crucial distinction: the "cache" *is* the model. There is no separate base network that must be kept consistent with the edits; the basins are both the storage and the decision mechanism. This eliminates the consistency problem — that the base model's parametric knowledge contradicts the edited cache entries — which is a known failure mode of retrieval-augmented editing (Zhong et al., 2023).

We encourage the CL community to adopt editability as a standard evaluation protocol. The three surgeries we propose — delete a class, relabel a class, inject a held-out class from N samples — form a minimal, reproducible editability benchmark that any prototype or exemplar-based method can report. For gradient-based methods, the benchmark exposes a fundamental architectural limitation: editability and dense parameterization are in direct tension.

7.4 The role of frozen features and coupling to representation learning

Our deliberate choice of frozen PCA-whitened features is a methodological decision, not a limitation to be overcome. It serves two purposes. **First**, it isolates all observed behavior to the prototype layer, making every ablation cleanly attributable — when we vary the creation rule and observe a $\Delta = 0.696$ gap, we know the gap is due to the allocation mechanism, not to a confounding interaction with a learned encoder. **Second**, it establishes a lower bound: any method that learns features jointly with prototypes can only improve upon these numbers, so our results represent a conservative baseline against which future work can measure genuine progress.

Coupling DP-ProtOn to a learned encoder (e.g., a self-supervised ResNet or a progressively fine-tuned convnet) raises interesting questions that we leave to future work. The central challenge is the **encoder-prototype consistency problem**: if the encoder's weights are updated on new tasks, the feature-space geometry shifts, and previously learned basins — whose centroids and covariances were estimated in the old embedding — may no longer be well-placed. This is not merely a performance concern; it threatens the editability guarantee. If deleting a class requires re-encoding all remaining data through an updated encoder, the operation is no longer $O(|\mathcal{B}|)$. We see three promising directions: (i) train the encoder once on a large pretraining corpus and freeze it, accepting a fixed representation in exchange for guaranteed editability; (ii) use the CRP novelty signal to detect when feature drift has made a basin stale, triggering local re-estimation of that basin's centroid without global retraining; (iii) decouple the encoder and prototype layer entirely by training them on separate objectives, with the encoder optimized for representational quality and the prototype layer optimized for editability. The disentanglement between the encoder (continuous, gradient-based) and the prototype layer (discrete, set-based) that we demonstrate in this paper may be essential to making any of these approaches work.

A broader architectural question is whether the Gaussianity assumption of the CRP criterion — that each class's distribution in feature space is approximately normal — remains valid under learned representations. Self-supervised and contrastively learned features are known to produce approximately hyperspherical distributions (Wang & Isola, 2020), which may actually be *more* compatible with a diagonal-covariance Gaussian model than raw pixel features. Investigating this compatibility is a natural next step.

7.5 Scope and honest accounting of negative results

We have been transparent about components that did *not* work, and we believe the *why* behind each failure carries methodological value beyond our specific architecture.

Full covariance CRP (fails at 64-D). In $d = 64$, the required $d(d+1)/2 = 2,080$ parameters per basin cannot be reliably estimated from a streaming allocation of typically <100 points. The log-determinant $\log|\Sigma_i|$ becomes numerically unstable, and the CRP score's likelihood term $\log\mathcal{N}(x \mid c_i, \Sigma_i)$ is dominated by estimation noise rather than genuine class structure. This is a concrete manifestation of the $p > n$ problem in online learning: when the parameter count exceeds the effective sample size per basin, regularization (here, the diagonal constraint) is not optional but essential.

Energy-softmin routing (net-negative at all dimensions tested). As analyzed in Section 6.1, the failure is structural and dimension-dependent. Importantly, the basins *themselves* store useful structure — the same basins achieve 0.887 under a local readout. This is a clean demonstration that representation quality and readout quality are separable concerns, and that a poor readout can mask good representations.

Per-basin mini-GMM (negligible gain: 0.746→0.749). Enriching each basin with a mixture of M Gaussians should, in principle, better capture multi-modal within-class structure. The near-zero gain indicates that the CRP criterion already handles multi-modality implicitly by

creating separate basins for distinct modes, making within-basin mixture modeling redundant. This is evidence that the DP prior's mode-seeking behavior — driven by the $\log n_i$ rich-get-richer term — naturally discovers the correct number of components without explicit mixture modeling.

Barrier terms and precision gating. These were designed for out-of-distribution detection and novelty scoring, where they may still have value. Their irrelevance to classification accuracy is expected and does not reflect negatively on the components themselves — it reflects the fact that classification and OOD detection are distinct tasks that benefit from different mechanisms.

Publishing these negative results serves a methodological purpose beyond transparency. In a field where the gap between reported and actual performance is well-documented (Lipton & Steinhardt, 2019), ablation studies that report only improvements create a selection bias toward complex models: each added component is justified by a small positive delta, while components that hurt are silently removed and never reported. By documenting every component we tried and its effect (or lack thereof), we hope to contribute to a culture where *what doesn't work* is as informative as what does. DP-ProtOn is not the model we set out to build — it is the model that remained after we removed everything that did not survive empirical scrutiny. We believe this provenance, and the record of failed attempts that produced it, makes it a more trustworthy baseline, not a weaker one.

8 Limitations

- **No feature learning.** Frozen PCA-whitening bounds performance on raw high-dimensional inputs. Coupling DP-ProtOn to a learned encoder is the natural next step, but it introduces a non-trivial tension: if the encoder's weights are updated on new tasks, the feature-space geometry shifts, and previously learned basins — whose centroids and covariances were estimated in the old embedding — may no longer be well-placed. Maintaining editability under a drifting representation is an open problem that we consider essential for future work.

- **No rehearsal.** No replay buffer means no re-clustering of old data — this is the source of the residual gap to Joint (0.875 vs 0.969 on Split-MNIST) and offline k-means on synthetic data.

- α **is not data-adaptive.** The CRP concentration parameter α must currently be set to match the dataset's expected mode count (e.g., α=1.0 for $\approx$10 classes, α=10^4 for $\approx$30 modes). A data-driven adaptive α — for instance, via empirical Bayes marginal-likelihood maximization (Escobar & West, 1995) or online hyperparameter optimization — would substantially improve usability and is a natural extension.

- **Scope of evidence.** Three datasets in PCA-whitened spaces. No claim beyond frozen-feature, rehearsal-free continual classification.

- **No theoretical guarantees.** The CRP criterion is used as a heuristic allocation rule. Formal consistency guarantees or regret bounds for online prototype allocation under a DP prior remain open questions.

9 Conclusion

DP-ProtOn is what remained after we systematically falsified an energy-attractor design: a DP/CRP creation rule with diagonal covariance, one online denoised centroid per basin, and a Parzen density readout. Every surviving component is justified by a controlled, budget-matched ablation; every headline number carries multi-seed variance confirming stability. The model:

- Creates basins with 8× the prototype efficiency of the nearest alternative ($\Delta = 0.6960 \pm 0.0528$),
- Beats offline k-means at equal budget with a single online mean ($\Delta = 0.0340 \pm 0.0053$),
- Beats Experience Replay at equal memory on Split-MNIST by +0.20 ACC with near-zero forgetting (BWT= -0.0471),
- Supports training-free deletion, relabeling, and injection — operations softmax classifiers cannot perform.

We offer DP-ProtOn as an honest, fully-ablated baseline and as concrete evidence that prototype attractors are directly editable knowledge units.

Reproducibility

All results are reproducible across runs. Code (MIT license): https://github.com/Fengrru/dp-proton. Core implementation: `pw_afm.py` (≈340 lines). Experiment scripts: `exp_control_kmeans.py`, `exp_highdim.py`, `exp_mnist_bench.py`, `exp_edit.py`, `exp_multiseed.py`, `gen_paper_figs.py`, and the reproduction runner `run_all.py` (reruns every experiment and checks each number against the golden reference `RESULTS_MULTISEED.md`, printing PASS/FAIL per section). Results summary: `RESULTS_MULTISEED.md`. Figures: `results/fig_multiseed.png`, `results/fig_edit.png`.

References

[1] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. (2017). Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13), 3521–3526.

[2] Chaudhry, A., Rohrbach, M., Elhoseiny, M., et al. (2019). On tiny episodic memories in continual learning. *arXiv:1902.10486*.

[3] Lopez-Paz, D., & Ranzato, M. (2017). Gradient episodic memory for continual learning. *NeurIPS*, 30.

[4] Aljundi, R., Kelchtermans, K., & Tuytelaars, T. (2019). Task-free continual learning. *CVPR*, 11254–11263.

[5] Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: Incremental classifier and representation learning. *CVPR*, 2001–2010.

[6] Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., & Alahari, K. (2018). End-to-end incremental learning. *ECCV*, 233–248.

[7] Wu, Y., Chen, Y., Wang, L., et al. (2019). Large scale incremental learning. *CVPR*, 374–382.

[8] Mensink, T., Verbeek, J., Perronnin, F., & Csurka, G. (2013). Distance-based image classification: Generalizing to new classes at near-zero cost. *TPAMI*, 35(11), 2624–2637.

[9] Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2), 209–230.

[10] Blackwell, D., & MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2), 353–355.

[11] Teh, Y. W. (2010). Dirichlet process. In *Encyclopedia of Machine Learning*, 280–287. Springer.

[12] Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *ACM SIGMOD*, 25(2), 103–114.

[13] Cao, F., Ester, M., Qian, W., & Zhou, A. (2006). Density-based clustering over an evolving data stream with noise. *SDM*, 328–339.

[14] Sculley, D. (2010). Web-scale k-means clustering. *WWW*, 1177–1178.

[15] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. In *Predicting Structured Data*, MIT Press.

[16] Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? *ICDT*, 217–235.

[17] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

[18] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. *JMLR*, 12, 2825–2830.

[19] Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. *ICDT*, 420–434.

[20] Lipton, Z. C., & Steinhardt, J. (2019). Troubling trends in machine learning scholarship. *Communications of the ACM*, 62(6), 45–53.

[21] Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430), 577–588.

[22] Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6), 1152–1174.

[23] Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3), 400–407.

[24] Radovanović, M., Nanopoulos, A., & Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11, 2487–2531.

[25] Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2022). Locating and editing factual associations in GPT. *NeurIPS*, 35, 17359–17372.

[26] Mitchell, E., Lin, C., Bosselut, A., Finn, C., & Manning, C. D. (2022). Fast model editing at scale. *ICLR*.

[27] Yao, Y., Wang, P., Tian, B., et al. (2023). Editing large language models: Problems, methods, and opportunities. *EMNLP*, 1022–1042.

[28] Zhong, Z., Wu, Z., Manning, C. D., Potts, C., & Chen, D. (2023). MQuAKE: Assessing knowledge editing in language models via multi-hop questions. *EMNLP*, 15686–15709.

[29] Wang, T., & Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *ICML*, 9929–9939.