PRIOR TO MEMORY: BAYESIAN ASSIGNMENT PROJECTION FOR CONTINUAL GENERALIZED CATEGORY DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual generalized category discovery (C-GCD) requires discovering new classes from sequentially arriving unlabeled data containing both historical and new classes while retaining recognition of historical classes. However, parameter estimation under fixed pseudo-labels does not automatically correct sample assignments, so initial assignment errors may persist in class memory. We propose Bayesian Assignment Projection (BAP) to jointly refine old/new separation and assignments among new classes before memory construction. BAP treats the classes observed in the initial labeled stage (base classes) as sampled from the class population with equal inclusion probability. Their mean count estimates the population mean sample count per class. Under comparable observation scales for base and future classes, a shared-rate count model provides Bayesian predictions of future new-class sizes, whose predictive intervals define global assignment constraints. Cardinality constraints support reassessing initial assignments, while local geometric compatibility guides individual sample reassignments. BAP alternates regularized information projection with auxiliary Gaussian posterior estimation and constructs class memory from the refined assignments. We prove uniqueness of the fixed-cost projection solution and a non-increasing auxiliary objective under exact alternating updates within a fixed stage. Controlled comparisons use identical feature representations, data streams, and Gaussian memory models. Across multiple initializations, BAP improves mean final overall recognition accuracy on all four standard C-GCD benchmarks, with an average gain of 2.68 percentage points across datasets.

1 INTRODUCTION

Continual generalized category discovery (C-GCD) extends generalized category discovery (GCD; Vaze et al., 2022) to continual learning. It requires models to discover new classes from successive unlabeled batches containing both old and new classes while retaining recognition of previously encountered classes (Ma et al., 2024). Unlike static GCD, sample assignments at the current stage affect not only current recognition but also the construction of class models used in subsequent stages.

However, C-GCD methods that fit class distributions under fixed pseudo-labels primarily update class parameters conditional on existing sample memberships, without jointly reassessing those memberships (Dai & Chauhan, 2025). Parameter updates may therefore remain influenced by initially misassigned samples, preserving assignment errors in class memory and affecting recognition at subsequent stages.

In the absence of incremental labels, per-class sample counts can provide supplementary evidence for membership revision. In practice, the class composition of some natural communities and everyday activities can remain relatively stable over a given time span. For example, a long-term ecological survey found that the relative abundance structure of a particular shallow-water fish community remained stable across repeated observations in the same season (Hoff & Bronte, 1999). Observations of temporal stability in vehicle-type proportions and brand sales shares have also been reported for road traffic and mature consumer markets, respectively (Coropulis et al., 2025; Bron-

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.

nenberg et al., 2007). These observations suggest that historical class counts can provide a useful statistical reference for subsequent data when the observation scope, sampling procedure, and statistical scale are comparable.

From this perspective, we revisit the initial labeled stage, which we call the *base stage*; the classes observed in this stage are the *base classes*. Beyond class representations, the base stage provides direct observations of per-class sample counts. When classes are sampled with equal inclusion probability and count scales are comparable, the mean count of the base classes can be used to estimate the population mean sample count per class; Appendix A.2 provides the statistical interpretation. To further predict new-class sample counts, we introduce a shared-rate count model. Under its assumptions, we construct Bayesian posterior predictions for both the sample count of an individual new class and the total sample count of all new classes, then convert the predictive intervals into global cardinality constraints for the current mixed batch.

We propose *Bayesian Assignment Projection* (BAP) to jointly revise old/new separation and sample assignments among new classes before constructing class memory. When the class counts induced by the initial assignments fall outside the predictive ranges, updating distribution parameters under fixed memberships cannot change those counts. BAP treats initial pseudo-labels as revisable assignment variables. Through regularized information projection, it combines cardinality priors calibrated from the labeled data of the base stage with the local geometric compatibility of current samples: the former constrains assignments through overall class sizes, while the latter guides the membership choices of individual samples.

BAP then alternates assignment revision with auxiliary Gaussian posterior estimation. Revised memberships update class statistics, which in turn guide subsequent assignments; the refined memberships ultimately determine class memory. In controlled comparisons using identical feature representations, data streams, and Gaussian memory models, experiments across multiple initializations show that BAP improves final overall recognition accuracy over VB-CGCD by an average of 2.68 percentage points across four standard C-GCD benchmarks.

Our main contributions are as follows:

- **Statistical modeling.** We construct a Bayesian cardinality model calibrated by base classes, translating posterior predictions for individual and aggregate new-class sample counts into marginal constraints for continual class assignments.
- **Algorithm.** We propose BAP, which alternates information projection and analytic class estimation to iteratively revise the sample assignments used to construct class memory.
- **Theory.** We prove uniqueness and a finite class-potential representation of the fixed-cost projection solution, and establish auxiliary-objective descent for exact alternating inference within a fixed stage.

2 Related Work

Continual generalized category discovery. Existing methods primarily address representation and pseudo-label quality, together with continual memory modeling. Happy mitigates prediction bias through debiased learning (Ma et al., 2024), while OPCR combines confidence-aware label refinement with orthogonal prototype learning to improve pseudo-label reliability (Hu et al., 2026). For memory modeling, VB-CGCD combines self-corrective relabeling with variational Gaussian distribution estimation to construct class memory (Dai & Chauhan, 2025), whereas DYDM separates discovery and memory branches to balance new-class adaptation and retention of historical knowledge (Yu et al., 2026).

Class priors and sample counts. Class priors provide a global reference for sample assignments under unknown or imbalanced distributions. BYOP iteratively estimates class proportions from model predictions in novel class discovery (Yang et al., 2023). For generalized category discovery, ImbaGCD combines prior estimation with optimal-transport assignment (Li et al., 2023), while BaCon uses distribution estimates from its contrastive-learning branch to regularize pseudo-label predictions (Bai et al., 2023). These methods estimate class proportions from predictions or features of the current data. BAP uses labeled base-class counts to predict individual and aggregate

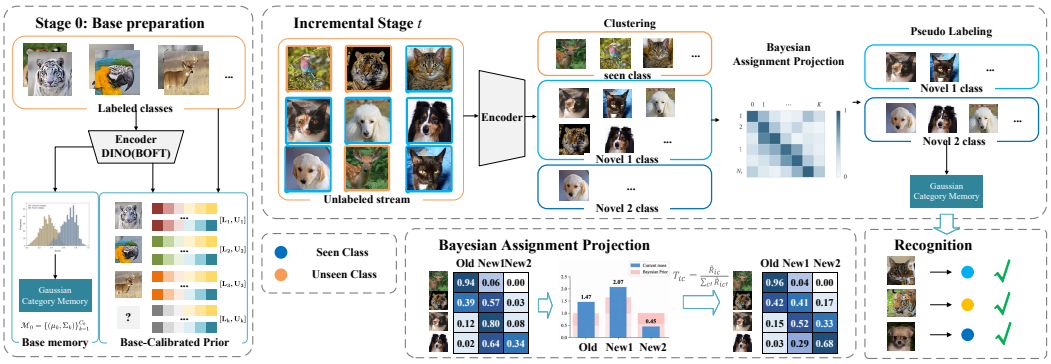


Figure 1: Overview of BAP. The base stage estimates geometric and cardinality priors; incremental inference alternates assignment refinement and auxiliary class estimation, then constructs new-class memory from global hard assignments. Images and probabilities are illustrative.

future new-class sample counts, converting predictive intervals into constraints for joint assignment refinement before memory construction.

Information projection and constrained inference. Information projection uses KL divergence to characterize the distributional adjustment needed to satisfy constraints (Csiszár, 1975). Posterior regularization introduces structural constraints into latent-variable inference (Ganchev et al., 2010). Entropy-regularized optimal transport provides computational tools for global marginal constraints (Cuturi, 2013), while unbalanced optimal transport further relaxes exact mass matching through marginal penalties (Chizat et al., 2018). BAP incorporates count constraints predicted from base classes into assignment projection and alternates updates of sample memberships and Gaussian class statistics.

3 Bayesian Assignment Projection

At the base stage, labeled data D_0^L are used to learn a representation and initialize class memory. The representation is then fixed. At stage t, the model receives historical memory M_{t-1} and an unlabeled batch X_t containing old classes and a known number K_t of new classes. Before constructing new-class memory, BAP uses base-class priors to jointly refine old/new separation and assignments among new classes (Figure 1). Below, we write $N = N_t$ and $K = K_t$.

3.1 From base-class counts to marginal constraints

Under uniform random class partitioning and a common observation protocol, the base-class mean count is unbiased for the population-average class count, supporting our scale calibration (Appendix A.2). Let $n_c^{(0)}$ denote the deduplicated labeled sample count of base class c, and let K_0 denote the number of base classes. To predict future new-class counts, BAP uses a shared-rate Poisson model for the sample counts of both base and future new classes. We place a Jeffreys prior on the shared count rate λ. The rate posterior is $\mathrm{Gamma}(a_0, b_0)$ in the shape–rate parameterization, with $a_0 = \frac{1}{2} + \sum_{c=1}^{K_0} n_c^{(0)}$ and $b_0 = K_0$. Integrating out the shared rate gives the individual and aggregate new-class count predictions

$$\widetilde{n}_k \mid D_0^L \sim \mathrm{NB}\left(a_0, \frac{b_0}{b_0+1}\right), \qquad \widetilde{N}_t^{\mathrm{new}} \mid D_0^L \sim \mathrm{NB}\left(a_0, \frac{b_0}{b_0+K}\right). \tag{1}$$

Here $\mathrm{NB}(r,p)$ has mean $r(1-p)/p$; the shared rate induces correlated predictions. Raw single-class and total-count intervals $[\widetilde{L}_1, \widetilde{U}_1]$ and $[L_\Sigma, U_\Sigma]$ each have nominal 95% model-based coverage. Known new-class presence imposes $L_1 = \max(1, \widetilde{L}_1)$, with $U_1 = \widetilde{U}_1$. Column 0 represents the historical-class background mixture, while the other columns represent current new classes. The

column-mass bounds are

$$\begin{aligned} l_k &= L_1, \qquad U_k = U_1 \quad (k \geq 1), \\ l_0 &= \max\{0, N - U_\Sigma, N - KU_1\}, \\ U_0 &= \min\{N, N - L_\Sigma, N - KL_1\}. \end{aligned} \tag{2}$$

Individual predictive intervals bound each new-class mass, while the aggregate interval constrains the background mass through conservation of the batch total. Their intersection has no joint 95% credible-region guarantee (Appendix A.5). Let $T \in \mathbb{R}_+^{N\times(K+1)}$ be the soft assignment matrix and $q = T^\top \mathbf{1}$ its column masses. The feasible domain and reference masses are

$$\begin{aligned} &\mathcal{C} = \{T \geq 0 : T\mathbf{1} = \mathbf{1},\ l \leq T^\top \mathbf{1} \leq U\}, \\ &\nu = (N - K\mu_\lambda, \mu_\lambda, \ldots, \mu_\lambda), \qquad \mu_\lambda = a_0/b_0, \qquad \pi = \nu/N. \end{aligned} \tag{3}$$

We require $\nu_0 > 0$ and $\mathcal{C} \neq \emptyset$ (Appendix C.2).

3.2 Refining local assignments with global cardinality information

Given matching costs $C^{(r)}$ at iteration r, BAP constructs local assignments $R^{(r)}$ and jointly refines the entire mixed batch through regularized information projection (Csiszár, 1975; Ganchev et al., 2010):

$$\begin{aligned} R_{ic}^{(r)} &= \frac{\pi_c \exp(-C_{ic}^{(r)}/\epsilon)}{\sum_{j=0}^{K} \pi_j \exp(-C_{ij}^{(r)}/\epsilon)}, \\ T^{(r+1)} &= \arg\min_{T\in\mathcal{C}} \epsilon D_{\mathrm{KL}}(T\|R^{(r)}) + \rho D_{\mathrm{KL}}(T^\top\mathbf{1}\|\nu). \end{aligned} \tag{4}$$

Matrix KL sums row-wise divergences; the marginal term is generalized KL, with $\epsilon = \rho = 1$. Equation (4) is an entropy-regularized transport problem with Bayesian-calibrated mass bounds and reference masses (Cuturi, 2013); the marginal KL term regularizes class masses around these references (Appendix A.6).

Proposition 1 (Projection uniqueness). *If costs are finite, $\epsilon, \rho > 0$, all components of π, ν are positive, $0 \leq l \leq U$, and $\mathcal{C} \neq \emptyset$, the projection in Eq. (4) has a unique solution. Its class potentials β_c are shared across all samples in the same column; for any two reachable columns a, b,*

$$\log \frac{T_{ia}^*}{T_{ib}^*} = \log \frac{R_{ia}^{(r)}}{R_{ib}^{(r)}} - \frac{\beta_a - \beta_b}{\epsilon}. \tag{5}$$

At the projection optimum, each reachable column satisfies

$$\beta_c = \rho \log \frac{q_c^*}{\nu_c} + \eta_c^+ - \eta_c^-. \tag{6}$$

Here $q_c^* = \sum_i T_{ic}^*$ is the optimal column mass and $\eta_c^\pm \geq 0$ are upper- and lower-bound multipliers, respectively. The marginal KL term continuously adjusts class preferences according to deviations from reference mass, even inside the admissible interval. Active bounds contribute additional corrections through their multipliers. Together, these terms produce shared class-level adjustments to each sample's relative assignments in Eq. (5) (Appendices B.1–B.2).

3.3 Alternating updates and class-memory construction

BAP initializes the one-hot $T^{(0)}$ from GMM labels a^0 and the first Gaussian fit's new-class predictions A_0. It may be infeasible; the first cost update and projection yield feasible $T^{(1)}$, where descent comparisons start. Auxiliary inference uses base-calibrated whitened features $u_i \in \mathbb{R}^d$ with $u_i \mid \theta_k \sim \mathcal{N}(\theta_k, I)$ and $\theta_k \sim \mathcal{N}(0, \kappa_0^{-1} I)$. Geometric parameters are estimated at the base stage and then fixed (Appendix C.1).

Let $q_k = \sum_i T_{ik}$ and $h_k = \sum_i T_{ik} u_i$. Using T_{ik} as the power weight of each sample likelihood gives the auxiliary posterior mean and covariance, and the expected negative log-likelihood cost for the next iteration:

$$\begin{aligned} m_k &= \frac{h_k}{\kappa_0 + q_k}, \qquad V_k = \frac{I}{\kappa_0 + q_k}, \\ C_{ik} &= \frac{1}{2}\|u_i - m_k\|^2 + \frac{d}{2(\kappa_0 + q_k)} + \frac{d}{2}\log(2\pi). \end{aligned} \tag{7}$$

Algorithm 1 Bayesian Assignment Projection

Require: D_0^L, $\{(X_t, K_t)\}_{t=1}^T$, and a fixed representation.
Ensure: Class memories $\{M_t\}_{t=1}^T$.
MM: alternate Eqs. (4) and (7); Harden: round capacities and solve Eq. (10).
Initialize base priors, M_0, and (n, h) from D_0^L.
for $t = 1, \ldots, T$ **do**
  $(a^0, A_0) \leftarrow$ GMM and first Gaussian fit from M_{t-1}.
  Construct $(\mathcal{C}_t, \nu, \pi, g_0)$; $T^{(0)} \leftarrow \mathrm{OneHot}(A_0, a^0)$.
  $(T_b, C_b) \leftarrow \mathrm{MM}(T^{(0)})$; $\mathcal{Q} \leftarrow \{\mathrm{Harden}(T_b, C_b)\}$.
  for each nondegenerate $T_k^{(0)} = \mathrm{SplitMerge}(T_b, k)$, $k = 1, \ldots, K_t$ **do**
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathrm{Harden}(\mathrm{MM}(T_k^{(0)}))\}$.
  end for
  $\widehat{Y} \leftarrow \arg\min_{Y \in \mathcal{Q}} F(Y)$; $Y^* \leftarrow \mathrm{Align}(\widehat{Y}, a^0)$.
  $A' \leftarrow \{i : Y^*_{i0} = 0\}$; obtain M_t by the second Gaussian fit from M_{t-1} on (A', Y^*).
  Store $n_k = \sum_i Y^*_{ik}$ and $h_k = \sum_i Y^*_{ik} u_i$, $k = 1, \ldots, K_t$.
end for

The historical background is an equally weighted mixture of the auxiliary predictive distributions of the O_t historical classes:

$$g_0(u) = \frac{1}{O_t} \sum_{c=1}^{O_t} \mathcal{N}(u; m_c, I + V_c), \qquad C_{i0} = -\log g_0(u_i). \tag{8}$$

Equation (7) supplies the negative log-evidence gradient, while Eq. (8) provides the fixed predictive background. The auxiliary model provides closed-form class-parameter updates during assignment search, avoiding a full-Gaussian recognition fit for each candidate.

Let $G_k(T)$ be the log evidence obtained by integrating out the auxiliary mean, with its closed form given in Appendix B.3. The iterations correspond to the auxiliary objective

$$\begin{aligned} F(T) = & -\sum_{k=1}^{K} G_k(T) - \sum_i T_{i0} \log g_0(u_i) + \epsilon \sum_{i,c} T_{ic} \log \frac{T_{ic}}{\pi_c} \\ & + \rho D_{\mathrm{KL}}(T^\top \mathbf{1} \| \nu) + I_{\mathcal{C}}(T), \end{aligned} \tag{9}$$

where $I_{\mathcal{C}}$ is the feasible-set indicator. Each $G_k(T)$ is convex in the assignments, so the first-order expansion of $-G_k$ gives a global upper bound tight at the current iterate. Retaining the entropy, marginal regularization, fixed historical background, and feasible domain yields a majorization–minimization (MM) update: exact projection in Eq. (4) minimizes this bound, ensuring a non-increasing F.

Proposition 2 (Within-stage descent)**.** *Fix the stage, priors, and historical background. Starting from feasible assignments, exact alternating updates satisfy* $F(T^{(r+1)}) \leq F(T^{(r)})$ *(Appendix B.3).*

Because the complete objective is generally nonconvex, BAP optimizes a finite candidate bank containing the baseline initialization and spectral split-and-merge initializations.

For each candidate, soft masses are converted to integer capacities $\bar{q}$ that satisfy the bounds and conserve total mass. BAP then solves the global hard-assignment problem

$$\min_{Y \in \{0,1\}^{N \times (K+1)}} \sum_{i,c} Y_{ic} C_{ic}, \qquad Y\mathbf{1} = \mathbf{1}, \qquad Y^\top \mathbf{1} = \bar{q}. \tag{10}$$

Candidates are selected by the recomputed complete hard objective $F(Y)$, and anonymous new-class indices are aligned by overlap with GMM assignments, without incremental ground-truth labels (Appendices B.4 and C.3). The selected new-class samples and hard labels drive the second, independent full-covariance Gaussian fit from the pre-stage memory M_{t-1}. Only new-class memory is updated; historical parameters remain fixed. Each stage performs exactly two such fits using the full Gaussian memory model of VB-CGCD (Dai & Chauhan, 2025). Algorithm 1 summarizes this continual update; Appendix C.2 provides the implementation in Algorithm S1. At test time, recognition directly uses class Gaussian densities, without batch-cardinality projection (Appendix C.4).

Table 1: Stage-wise recognition results on four benchmarks.

Dataset	Method	S_0	S_1			S_2			S_3			S_4			S_5		
		All	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
C100	PromptCCD	85.20	63.71	65.03	57.13	61.80	62.86	55.46	60.14	61.81	53.54	52.65	52.89	50.72	55.53	55.64	54.54
	Happy	90.10	83.62	85.08	76.30	75.83	80.85	45.70	70.03	73.50	45.70	64.70	65.06	61.80	60.77	62.02	49.50
	GOAL	91.00	82.80	86.40	65.10	77.80	84.00	62.40	71.80	82.80	53.50	66.60	83.80	45.10	61.50	79.20	43.80
	VC-CGCD	90.64	83.22	83.56	**81.50**	76.33	85.26	54.00	70.55	79.74	55.23	64.99	81.94	43.80	61.28	62.70	48.50
	VB-CGCD†	**92.10**	88.25	89.82	80.40	86.14	86.68	82.90	84.20	84.67	80.90	83.08	83.51	79.60	81.49	82.61	71.40
	BAP	**92.10**	**89.02**	**90.56**	81.30	**87.87**	**87.80**	**88.30**	**86.29**	**86.63**	**83.90**	**86.00**	**85.74**	**88.10**	**84.32**	**85.36**	**75.00**
Tiny	PromptCCD	79.23	64.10	64.72	61.02	61.43	62.07	57.58	56.39	57.08	51.53	55.38	55.52	54.27	50.74	50.82	50.00
	Happy	85.76	77.80	82.68	53.40	72.26	76.02	49.70	65.29	70.06	31.90	57.79	61.40	28.90	54.93	56.08	44.60
	GOAL	86.10	77.10	81.90	52.80	72.10	80.30	51.60	67.30	77.00	51.10	61.50	76.40	42.80	57.20	74.50	39.90
	VC-CGCD	85.15	77.15	80.88	58.50	70.54	81.38	43.45	63.52	78.88	37.93	57.56	77.68	32.40	53.25	75.98	30.52
	VB-CGCD†	**88.56**	84.90	86.34	**77.70**	81.99	83.95	70.20	79.66	81.06	69.90	76.66	78.61	61.00	74.29	75.99	59.00
	BAP	**88.56**	**85.07**	**87.24**	74.20	**83.27**	**83.98**	**79.00**	**81.45**	**82.14**	**76.60**	**79.17**	**80.76**	**66.40**	**77.37**	**78.40**	**68.10**
IN100	PromptCCD	91.18	82.11	82.36	80.84	80.03	80.36	78.02	77.78	78.03	76.05	67.37	67.54	65.99	63.97	64.08	62.95
	Happy	96.20	91.20	95.36	70.40	87.83	90.83	69.80	85.22	86.40	77.00	81.93	83.00	73.40	78.58	79.11	73.80
	GOAL	96.20	**93.90**	**95.90**	83.80	88.80	**94.90**	73.70	86.40	**94.40**	72.90	81.20	**92.40**	68.40	79.30	**93.00**	66.60
	VC-CGCD	**96.21**	91.10	95.52	69.00	86.31	94.72	65.30	83.33	93.96	65.60	78.60	92.24	61.55	76.72	91.88	61.55
	VB-CGCD†	94.12	93.10	93.04	93.40	90.86	91.90	84.60	88.53	89.49	81.80	86.00	87.38	75.00	85.00	85.09	84.20
	BAP	94.12	93.23	92.96	**94.60**	**92.43**	92.37	**92.80**	**91.13**	90.89	**92.80**	**90.40**	90.15	**92.40**	**89.98**	89.73	**92.20**
CUB	PromptCCD	82.10	63.21	63.91	59.71	51.00	52.17	43.98	51.80	52.57	41.43	46.21	46.86	41.61	50.95	51.16	49.06
	Happy	86.00	77.83	82.44	55.37	68.19	76.24	21.51	63.44	67.47	34.63	58.39	61.90	31.14	54.37	56.70	33.67
	GOAL	**90.40**	**81.10**	83.10	**71.30**	73.10	76.40	**65.00**	69.00	76.10	57.30	64.20	74.60	51.40	61.90	76.00	47.90
	VC-CGCD	90.11	79.99	**87.34**	44.37	72.15	**83.91**	43.05	67.31	**82.77**	41.70	62.94	**79.99**	41.79	58.75	**78.29**	39.38
	VB-CGCD†	86.41	78.35	83.99	49.30	73.31	76.98	51.45	70.77	71.60	**64.86**	66.85	69.95	41.37	64.76	65.98	53.51
	BAP	86.41	79.04	85.18	47.36	**74.24**	78.12	51.11	**71.31**	73.51	55.59	**68.72**	70.73	**52.29**	**66.98**	68.24	**55.44**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and protocol. We evaluate on CIFAR-100 (C100) (Krizhevsky et al., 2009), TinyImageNet-200 (Tiny) (Le et al., 2015), ImageNet-100 (IN100) (Deng et al., 2009), and CUB-200-2011 (CUB) (Wah et al., 2011). The labeled base stage contains half the classes: 50 for C100/IN100 and 100 for Tiny/CUB. Five incremental stages introduce 10 or 20 new classes each, respectively. Each receives an unlabeled mixture of historical and current new classes, with the current new-class count known. Stage-wise testing covers all classes introduced so far. Appendix D specifies the streams; Appendix F details the ten-stage extension.

Baselines and implementation. Comparisons include PromptCCD (Cendra et al., 2024), Happy (Ma et al., 2024), GOAL (Han et al., 2026), VC-CGCD (Xiong et al., 2026), and VB-CGCD (Dai & Chauhan, 2025). BAP and our reproduced VB-CGCD (†) share representations, data streams, Gaussian memory models, and evaluation mappings; other baseline rows retain literature-reported results. A DINO-pretrained ViT-B/16 (Caron et al., 2021) is adapted with BOFT (Liu et al., 2024) at the base stage, then frozen together with the base-fitted PCA transform. Initial GMM clustering uses 768-dimensional features; auxiliary inference and class memory use 384 dimensions. Unless ablated, $\epsilon = \rho = 1$, the nominal predictive level is 0.95, and candidates receive at most 50 alternating updates. Appendix C gives optimization budgets and stopping criteria.

Evaluation. Final All measures final-stage overall accuracy; Old and New cover previously observed and current new classes, respectively. Their harmonic mean is H, and MeanAll averages All across incremental stages. M_f measures the base-class accuracy decrease from the base to final stage; M_d averages current-new accuracy across increments. Anonymous categories receive an evaluator-side mapping from their initial GMM assignments at introduction, held fixed thereafter. Incremental ground truth does not enter learning or assignment optimization. Tables 1–3 report fixed-initialization results; Appendix D separately summarizes multiple initializations.

Table 2: Forgetting and new-class discovery.

Method	C100		Tiny		IN100		CUB	
	$M_f \downarrow$	$M_d \uparrow$	$M_f \downarrow$	$M_d \uparrow$	$M_f \downarrow$	$M_d \uparrow$	$M_f \downarrow$	$M_d \uparrow$
PromptCCD	21.66	54.28	23.46	54.88	18.76	72.77	19.10	47.15
Happy	12.08	55.80	9.16	41.70	10.13	72.88	3.22	35.26
VB-CGCD	6.48	79.04	5.88	67.56	5.52	83.80	6.66	52.10
BAP	5.08	83.32	4.80	72.86	4.52	92.96	3.99	52.36
UpperBound	5.14	83.76	3.48	77.04	3.16	89.04	5.13	79.65

Table 3: Ten-stage results.

Dataset	Method	Final All	Mean All
C100	VB-CGCD	77.70	82.62
C100	BAP	82.23	85.28
Tiny	VB-CGCD	72.94	79.59
Tiny	BAP	75.88	82.15

4.2 Recognition Performance

Overall and stage-wise recognition. Table 1 shows that BAP outperforms the listed methods in final All on all four datasets. Against reproduced VB-CGCD, the strongest listed baseline, the mean gains are 3.28 percentage points with fixed initialization and 2.68 across initializations (Appendix D), with the overall advantage persisting across stages. Final-stage gains on C100, Tiny, and IN100 cover both historical and current new classes, with larger gains for new classes. Across datasets, BAP follows the same design: global count information adjusts class allocations, local matching guides sample memberships, and alternating updates re-estimate new-class statistics as memberships change. The refined models support current recognition and become historical memory for later stages, linking assignment optimization to continual recognition.

Forgetting and discovery. Table 2 separates retention from current-new recognition using the reproduced VB-CGCD entries. BAP reduces base-class forgetting and increases mean discovery accuracy on all four datasets in these runs. On CUB, the fixed-initialization results show stage-dependent changes in new-class recognition (Table 1). Averaged over three incremental initializations, final All and Old accuracy improve by 1.38 and 1.82 percentage points, respectively, while final New accuracy and H decrease by 2.69 and 1.26 points (Appendix D). The overall gain on CUB therefore primarily reflects improved recognition of previously observed classes.

Ten-stage extension. Keeping the base-class ratio at 50% and the final class set unchanged, ten increments introduce fewer new classes per stage and require more successive memory updates. Per-class size predictions remain calibrated by base-class counts, while aggregate predictions adjust to the current new-class count; the same inference procedure therefore handles finer stage partitions. Table 3 shows Final All and MeanAll gains over reproduced VB-CGCD of 4.53 and 2.66 percentage points on C100, and 2.94 and 2.56 on Tiny, respectively. The recognition advantage thus persists over a longer stage sequence.

4.3 Ablation Studies

Marginal constraints. Table 4 contains separate matched control blocks, each retaining its own reference configuration. In group (a), removing per-class bounds reduces Final All by 1.03 points, while removing the additional aggregate bound reduces it by 0.15. Both controls retain other marginal information and regularization. Per-class limits contribute more in this block than the extra total-mass restriction.

Alternating inference. Group (b) compares one-shot projection with repeated assignment and class-statistic updates under matched initial structures and marginal conditions. Neither configuration uses spectral candidate search. The displayed endpoints improve by approximately 2.01 points on C100 and 2.10 on CUB, supporting repeated reassessment using updated statistics.

Table 4: Component ablations and alternative design comparisons.

Setting	All	Old	New	Δ
(a) Class-marginal constraints: C100				
Full	84.25	85.31	74.70	0.00
w/o class bound	83.22	84.24	74.00	-1.03
w/o total bound	84.10	85.23	73.90	-0.15
(b) Alternating inference: C100 / CUB				
C100/one-shot	82.31	83.19	74.40	0.00
C100/alternating	84.32	85.36	75.00	$+2.01$
CUB/one-shot	65.38	67.38	47.02	0.00
CUB/alternating	67.48	68.68	56.49	$+2.10$
(c) Prediction and labels: C100				
Full	84.25	85.31	74.70	0.00
Poisson	84.35	85.33	75.50	$+0.10$
row argmax	84.50	85.43	76.10	$+0.25$

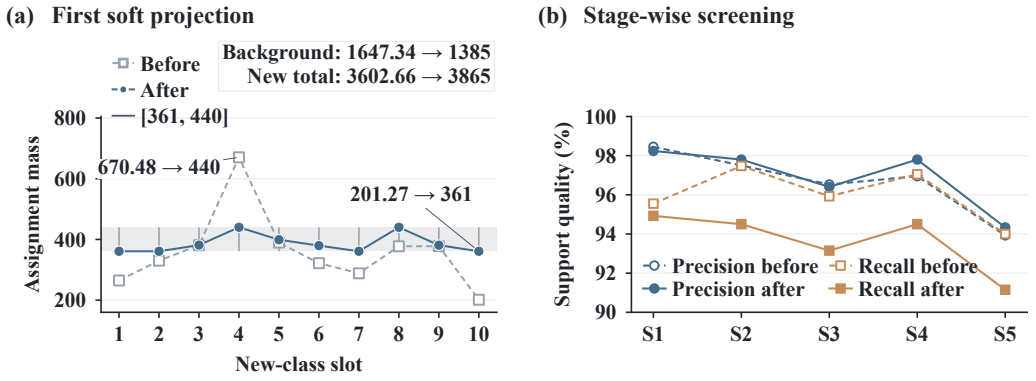


Figure 2: First-projection assignment masses and stage-wise new-class screening precision/recall on CIFAR-100.

Separately, applying BAP only at S_1 and reverting to VB-CGCD thereafter increases final All accuracy by 0.34 percentage points on C100 and 0.05 on CUB over VB-CGCD throughout (Appendix F, Table F1). These gains are smaller than those from applying BAP at every stage (Table 1), supporting continued assignment revision throughout the sequence.

Alternative designs. In group (c), the plug-in Poisson and row-argmax variants exceed Full by 0.10 and 0.25 percentage points, respectively. Both retain marginal constraints and alternating inference. Appendix E defines the controls. Appendix G compares alternative scoring rules on the trained memories.

Count-prior sensitivity. On C100, Final All varies by 0.13 percentage points across 90–99% prediction levels. Under class-dependent subsampling, BAP gains 1.91 points on C100 but loses 3.02 on CUB relative to reproduced VB-CGCD (Appendix F), illustrating a limitation of transferring count priors across observation scales.

4.4 Assignment Diagnostics and Error Analysis

Quantity constraints and screening. Figure 2 shows soft-mass adjustment and stage-wise screening quality in the CIFAR-100 seed-0 diagnostic run. In the first projection, the extreme slot masses move from 670.48 to 440 and from 201.27 to 361. Other slots retain different masses within the interval: the bounds limit excessive concentration and under-allocation without requiring equal class masses.

Training assignment gains and losses. Figure 3 evaluates the balance of assignment changes over all true current-new training samples after full revision. After full revision, recall decreases, and precision does not improve consistently across stages. Within-new repair accounts for most of the positive contribution at S_1–S_4. At S_5, internal relabeling still yields more repairs than harmful changes, but excluding previously correct samples reverses the overall balance. Correct coverage increases by 4.44 percentage points on average across stages but falls by 2.18 at S_5. Figures 2 and 3 show that lower screening recall can coexist with higher correct coverage when net gains from within-new relabeling outweigh screening losses. Screening precision and recall alone therefore do not capture the net effect of assignment revision.

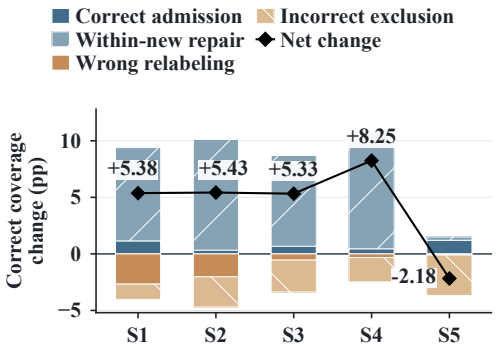


Figure 3: Training correct-coverage change on CIFAR-100 (all 4,000 true current-new samples per stage). The four signed contributions sum to the black net-change curve (pp).

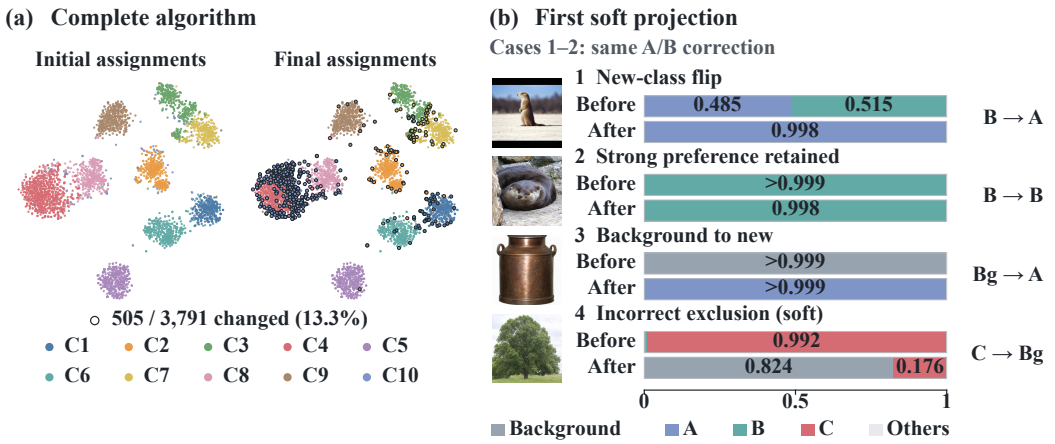


Figure 4: Initial assignments and final hard assignments in fixed coordinates, alongside four cases from the first soft projection on CIFAR-100. Thumbnails are AI-enhanced for display; assignment values are computed from the original experiment inputs.

Overall changes and local responses. Figure 4 uses identical samples and two-dimensional coordinates to show changed assignments rather than feature movement. Its scatterplots compare initial membership with the complete algorithm's final hard assignments. The four cases instead come from the first soft projection. Cases 1 and 2 receive the same multiplicative increase in their A/B odds. This reverses the nearly balanced preference in Case 1 but leaves the strong B preference in Case 2 intact. Their different responses arise from the interaction between a shared class-level correction and the pre-projection local preferences. Cases 3 and 4 illustrate background-to-new admission and loss of a correct preference to background.

Final test-error sources. Table 5 decomposes final errors relative to reproduced VB-CGCD. Same-cohort confusion decreases most on C100, Tiny, and IN100. For CUB, the 3.94-point reduction in later-cohort errors is partly offset by a 2.24-point increase in earlier-cohort errors; same-cohort errors decrease by 0.54 points. In contrast, IN100's 4.20-point same-cohort reduction faces only a 0.04-point increase in earlier-cohort errors. With paired, equally weighted test samples and fixed evaluation mappings, the unrounded components sum to the accuracy gain.

Table 5: Final-test error reduction relative to reproduced VB-CGCD.

Dataset	Error reduction (pp)			Net ΔACC
	Same	Earlier	Later	(pp)
C100	+2.57	−0.69	+0.95	+2.83
Tiny	+2.57	−0.22	+0.73	+3.08
IN100	+4.20	−0.04	+0.82	+4.98
CUB	+0.54	−2.24	+3.94	+2.23

Earlier and Later denote predictions into classes introduced before and after the true class, respectively. Positive/negative values indicate fewer/more errors. Independent rounding may yield a 0.01-point display difference.

5 Conclusion

We proposed Bayesian Assignment Projection (BAP), which jointly revises old/new separation and assignments among novel classes before category memory construction. By converting posterior predictions from base-class counts into marginal constraints and alternating information projection with auxiliary Gaussian posterior estimation, BAP brings global scale information and local geometric evidence into the assignment optimization. Under identical feature representations, data streams, and Gaussian memory models, BAP improves mean final overall recognition accuracy by 2.68 percentage points across four standard benchmarks, indicating that explicitly revising sample assignments can improve continual category discovery. Under class-dependent subsampling, BAP improves final overall accuracy on C100 but reduces it on CUB relative to reproduced VB-CGCD (Appendix F). Future work will extend count calibration to accommodate heterogeneous class observation scales and assignment inference to settings with unknown numbers of novel classes.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and stylistic refinement of this manuscript. These tools did not replace the authors in defining the problem, formulating technical claims, designing experiments, interpreting results, or conducting final verification. All text generated with AI assistance was reviewed and revised by the authors, who take full responsibility for the final content of this paper.

ETHICS STATEMENT

This study uses existing image benchmarks, including CIFAR-100, CUB-200-2011, ImageNet-100, and TinyImageNet, to investigate sample assignments and class-memory construction in continual generalized category discovery. These benchmarks may contain sampling biases, and the performance improvements reported in this paper do not imply equal reliability across all classes or real-world environments. In particular, when sampling scales or class composition differ between historical and current data, constraints derived from historical counts may affect the discovery and recognition of rare classes. Practical applications should therefore assess the method's suitability in light of data sources, class distributions, and the potential consequences of misclassification, while respecting the licenses and terms of use of the data and models employed.

REPRODUCIBILITY STATEMENT

Section 3 presents the main mathematical definitions and algorithmic procedure of BAP, and Section 4 describes the experimental settings and evaluation metrics. Appendix A provides the count model, posterior predictions, and statistical basis for scale calibration; Appendix B states the assumptions and proofs of the theoretical results; and Appendix C details the implementation, solver settings, stopping criteria, and numerical validity checks. Appendix D reports the data-stream protocols and results across multiple initializations; Appendices E–F provide supplementary ablations, extensions, and sensitivity experiments; and Appendix G records computational costs and reproducibility information. Together, these materials specify the key conditions needed to implement the method, reproduce the experimental comparisons, and verify the results.

REFERENCES

Jianhong Bai, Zuozhu Liu, Hualiang Wang, Ruizhe Chen, Lianrui Mu, Xiaomeng Li, Joey Tianyi Zhou, Yang Feng, Jian Wu, and Haoji Hu. Towards distribution-agnostic generalized category discovery. *Advances in Neural Information Processing Systems*, 36:58625–58647, 2023.

Bart J Bronnenberg, Sanjay K Dhar, and Jean-Pierre Dubé. Consumer packaged goods in the united states: National brands, local branding. *Journal of Marketing Research*, 44(1):4–13, 2007.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Fernando Julio Cendra, Bingchen Zhao, and Kai Han. Promptccd: Learning gaussian mixture prompt pool for continual category discovery. In *European conference on computer vision*, pp. 188–205. Springer, 2024.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of computation*, 87(314):2563–2609, 2018.

Stefano Coropulis, Paolo Intini, Nicola Introcaso, and Vittorio Ranieri. Credible variable speed limits for improving road safety: A case study based on italian two-lane rural roads. *Sustainability*, 17(11):4833, 2025.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Hao Dai and Jagmohan Chauhan. Continual generalized category discovery: Learning and forgetting from a bayesian perspective. *arXiv preprint arXiv:2507.17382*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049, 2010.

Jizhou Han, Chenhao Ding, SongLin Dong, Yuhang He, Shaokun Wang, Qiang Wang, and Yihong Gong. Goal: Geometrically optimal alignment for continual generalized category discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4565–4573, 2026.

Haitham Hindi. A tutorial on convex optimization. In *Proceedings of the 2004 American control conference*, volume 4, pp. 3252–3265. IEEE, 2004.

Michael H Hoff and Charles R Bronte. Structure and stability of the midsummer fish communities in chequamegon bay, lake superior, 1973–1996. *Transactions of the American Fisheries Society*, 128(2):362–373, 1999.

Ningge Hu, Xiao Li, Zihao Zhao, Bo Xu, Jiaqing Liu, and Xuezheng Fan. Opcr: Continual generalized category discovery via orthogonal prototypes and confidence-aware label refinement. *Neurocomputing*, pp. 133011, 2026.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Ziyun Li, Ben Dai, Furkan Simsek, Christoph Meinel, and Haojin Yang. Imbagcd: Imbalanced generalized category discovery. *arXiv preprint arXiv:2401.05353*, 2023.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization. In *International Conference on Learning Representations*, volume 2024, pp. 38317–38350, 2024.

Shijie Ma, Fei Zhu, Zhun Zhong, Wenzhuo Liu, Xu-Yao Zhang, and Cheng-Lin Liu. Happy: A debiased learning framework for continual generalized category discovery. *Advances in Neural Information Processing Systems*, 37:50850–50875, 2024.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. *arXiv preprint arXiv:2201.02609*, 2022.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Jiahui Xiong, Qiuxia Lai, and Hongsong Wang. Virtual category-guided continual generalized category discovery. In *European Conference on Computer Vision*, pp. 58–75. Springer, 2026.

Muli Yang, Liancheng Wang, Cheng Deng, and Hanwang Zhang. Bootstrap your own prior: Towards distribution-agnostic novel class discovery. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3459–3468. IEEE, 2023.

Jiawei Yu, Zijian Gao, Xingxing Zhang, Xuan Liu, Huaimin Wang, and Kele Xu. Decouple your discovery and memory in continual generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25022–25031, 2026.

A NOTATION AND THE BAYESIAN CARDINALITY MODEL

This appendix gives the notation and derivations used by Bayesian Assignment Projection (BAP). The appendix notation κ corresponds to κ_0 in the main text.

A.1 NOTATION

Table S1: Notation used in the Bayesian model and optimization proofs.

Symbol	Type or dimension	Meaning
D_0^L	set	labeled base-stage observations
K_0, N_0	scalars	number of base classes and base observations
X_t, N_t	set, scalar	unlabeled stream at stage t and its size
O_t, K_t	scalars	old classes already in memory and new slots at stage t
z_i, u_i	vectors in $\mathbb{R}^d$	PCA feature and whitened auxiliary feature
$n_c^{(0)}$	count	raw deduplicated labeled observations of base class c
λ	positive scalar	shared class-cardinality rate
a_0, b_0	positive scalars	Gamma shape and rate for λ
μ_λ	scalar	posterior mean a_0/b_0
$\widetilde{n}_k$	random count	future count of a single new class
$\widetilde{N}_t^{\text{new}}$	random count	total future count over the K_t new classes
R, T, Y	$N_t \times (K_t + 1)$	local assignment, soft assignment, hard assignment
$q, \bar{q}$	vectors in $\mathbb{R}_+^{K_t+1}$	soft column mass and integer capacity
l, U	vectors in $\mathbb{R}_+^{K_t+1}$	lower and upper column-mass bounds
ν, π	vectors in $\mathbb{R}_+^{K_t+1}$	soft target mass and prior probability, $\pi = \nu/N_t$
β	vector in $\mathbb{R}^{K_t+1}$	column potential in the projection dual
ϵ, ρ	positive scalars	row-entropy and marginal-KL weights
κ	positive scalar	auxiliary Gaussian prior precision
b_{geom}	nonnegative scalar	implementation variance parameter, $b_{\text{geom}} = 1/\kappa$
m_k, V_k	vector, covariance	auxiliary posterior mean and variance for new slot k
h_k, s_k	vector, scalar	weighted feature sum and weighted squared-norm sum
$A^{(0)}, A'$	binary masks	initial and final current-stage support masks

All assignment matrices are row-stochastic: $T\mathbf{1} = \mathbf{1}$ and $T_{ic} \geq 0$. Column masses are $q = T^\top \mathbf{1}$. The feasible set at stage t is

$$\mathcal{C}_t = \{T \geq 0 : T\mathbf{1} = \mathbf{1},\ l \leq T^\top \mathbf{1} \leq U\}. \tag{A.1}$$

We omit the stage subscript when no confusion is possible.

A.2 RANDOM CLASS PARTITIONING AND COUNT-SCALE CALIBRATION

Consider a finite population of $M_{\text{pop}} \geq 2$ classes with fixed nonnegative counts $n_1, \ldots, n_{M_{\text{pop}}}$ under a common observation protocol. A common protocol means comparable observation, labeling, and sampling scales for base and target new classes. This comparability is an assumption, not a consequence of random class selection; it does not make benchmark sampling caps additional inputs to BAP. Define the population mean and finite-population variance by

$$\mu_{\text{pop}} = \frac{1}{M_{\text{pop}}} \sum_{c=1}^{M_{\text{pop}}} n_c, \qquad S_{\text{pop}}^2 = \frac{1}{M_{\text{pop}} - 1} \sum_{c=1}^{M_{\text{pop}}} (n_c - \mu_{\text{pop}})^2. \tag{A.2}$$

Choose a base-class set $\mathcal{S}_0$ by simple random sampling without replacement, with fixed size $1 \leq K_0 < M_{\text{pop}}$, and let

$$\widehat{\mu}_0 = \frac{1}{K_0} \sum_{c \in \mathcal{S}_0} n_c. \tag{A.3}$$

Taking the first K_0 classes of a uniformly random class permutation is equivalent to this design. Sampling expectations and variances below concern the class partition with the counts held fixed, not clustering initialization or optimization randomness. The set notation $\mathcal{S}_0$ is distinct from the scalar count sum S_0 used in Appendix A.3; M_{pop} is the number of classes, not the memory M_t.

Proposition 3 (Count-scale calibration under random class partitioning)**.** *Under the sampling design above,*

$$\mathbb{E}[\widehat{\mu}_0] = \mu_{\text{pop}}, \qquad \text{Var}(\widehat{\mu}_0) = \frac{1 - K_0/M_{\text{pop}}}{K_0} S^2_{\text{pop}}. \tag{A.4}$$

These identities do not require equal class counts. For any fixed $1 \le K_t \le M_{\text{pop}} - K_0$, let $\mathcal{S}_t$, conditional on $\mathcal{S}_0$, be a uniformly chosen K_t-class subset of the remaining classes. Then, over the complete random partition,

$$\mathbb{E}\left[K_t \widehat{\mu}_0 - \sum_{c \in \mathcal{S}_t} n_c\right] = 0. \tag{A.5}$$

Proof. Write $M = M_{\text{pop}}$ within this proof and let $I_c = \mathbf{1}\{c \in \mathcal{S}_0\}$. Simple random sampling without replacement gives

$$\mathbb{E}[I_c] = p = \frac{K_0}{M}, \qquad \mathbb{E}[I_c I_d] = \frac{K_0(K_0 - 1)}{M(M-1)} \quad (c \neq d). \tag{A.6}$$

Substituting $\widehat{\mu}_0 = K_0^{-1} \sum_c I_c n_c$ yields

$$\mathbb{E}[\widehat{\mu}_0] = \frac{1}{K_0} \sum_c \mathbb{E}[I_c] n_c = \frac{1}{M} \sum_c n_c = \mu_{\text{pop}}.$$

This step needs only fixed sample size and equal inclusion probabilities. For the variance, the pairwise probabilities in Eq. (A.6) additionally imply

$$\text{Var}(I_c) = p(1-p), \qquad \text{Cov}(I_c, I_d) = \frac{K_0(K_0-1)}{M(M-1)} - p^2 = -\frac{p(1-p)}{M-1}.$$

Consequently,

$$\begin{aligned} \text{Var}(\widehat{\mu}_0) &= \frac{1}{K_0^2} \left[\sum_c n_c^2 \,\text{Var}(I_c) + \sum_{c \neq d} n_c n_d \,\text{Cov}(I_c, I_d) \right] \\ &= \frac{p(1-p)}{K_0^2 (M-1)} \left[M \sum_c n_c^2 - \left(\sum_c n_c \right)^2 \right] \\ &= \frac{p(1-p)M}{K_0^2} S^2_{\text{pop}} = \frac{1 - K_0/M}{K_0} S^2_{\text{pop}}. \end{aligned} \tag{A.7}$$

Here the second line uses $\sum_{c \neq d} n_c n_d = (\sum_c n_c)^2 - \sum_c n_c^2$. Thus the exact variance uses the simple-random-sampling design, rather than equal inclusion probabilities alone.

For the future set, each class has marginal inclusion probability

$$\Pr(c \in \mathcal{S}_t) = \mathbb{E}\left[\mathbf{1}\{c \notin \mathcal{S}_0\} \frac{K_t}{M - K_0}\right] = \frac{M - K_0}{M} \frac{K_t}{M - K_0} = \frac{K_t}{M}.$$

Therefore $\mathbb{E}[\sum_{c \in \mathcal{S}_t} n_c] = (K_t/M) \sum_c n_c = K_t \mu_{\text{pop}} = K_t \mathbb{E}[\widehat{\mu}_0]$, proving Eq. (A.5). □

The total-count result expresses absence of systematic bias across random class partitions. Conditional on a realized $\mathcal{S}_0$, the expected future count is instead K_t times the mean of the remaining classes, which need not equal $K_t \widehat{\mu}_0$.

Random class partitioning thus supports the population representativeness of the base-class mean. Reindexing the observed base counts as $n_c^{(0)}$, the Gamma posterior in Appendix A.3 has mean

$$\mu_\lambda = \frac{\sum_{c \in \mathcal{S}_0} n_c + 1/2}{K_0} = \widehat{\mu}_0 + \frac{1}{2K_0}. \tag{A.8}$$

The exact unbiasedness statement concerns $\widehat{\mu}_0$, not μ_λ, which includes the Jeffreys correction. The shared-rate Poisson model is a further working assumption used to obtain computable predictive intervals. Random-partition unbiasedness alone establishes neither Poisson model correctness nor joint 95% coverage of the final constraint intersection (Appendix A.5).

A.3 JEFFREYS GAMMA POSTERIOR

BAP calibrates class cardinality only from labeled S0 observations. Let $n_c^{(0)}$ denote the raw deduplicated observation count of base class c; data augmentation, training epochs, pseudo-labels, query labels, and hidden test labels do not enter this count. Dataset sampling caps define the benchmark stream but are not additional inputs to the count prior, reference masses, or assignment bounds. The implemented working model is

$$n_c^{(0)} \mid \lambda \overset{\text{iid}}{\sim} \text{Poisson}(\lambda), \qquad p(\lambda) \propto \lambda^{-1/2}, \tag{A.9}$$

where the prior is Jeffreys' prior for a Poisson rate. With $S_0 = \sum_{c=1}^{K_0} n_c^{(0)}$, the posterior density is

$$p(\lambda \mid D_0^L) \propto \lambda^{S_0 - 1/2} \exp(-K_0 \lambda). \tag{A.10}$$

Under the shape–rate parameterization,

$$\lambda \mid D_0^L \sim \text{Gamma}(a_0, b_0), \qquad a_0 = S_0 + \tfrac{1}{2}, \quad b_0 = K_0, \qquad \mu_\lambda = \frac{a_0}{b_0}. \tag{A.11}$$

This is the only rate posterior used to calibrate class cardinality in the BAP implementation. It is not an empirical-Bayes grid search, a class-specific rate model, or a hyperparameter selected according to downstream accuracy.

A.4 POSTERIOR PREDICTIVE COUNTS UNDER A SHARED RATE

For one future new class,

$$\widetilde{n}_k \mid \lambda \sim \text{Poisson}(\lambda). \tag{A.12}$$

Integrating the Gamma posterior gives the negative-binomial predictive distribution

$$\widetilde{n}_k \mid D_0^L \sim \text{NB}\left(a_0, \frac{b_0}{b_0 + 1}\right), \tag{A.13}$$

where $\text{NB}(r, p)$ denotes the distribution over nonnegative counts with probability mass function

$$\Pr(X = x) = \frac{\Gamma(x + r)}{\Gamma(r) x!} p^r (1 - p)^x, \qquad x = 0, 1, 2, \ldots. \tag{A.14}$$

Therefore,

$$\mathbb{E}[\widetilde{n}_k \mid D_0^L] = \frac{a_0}{b_0}, \qquad \text{Var}(\widetilde{n}_k \mid D_0^L) = \frac{a_0}{b_0} + \frac{a_0}{b_0^2}. \tag{A.15}$$

The K_t future new classes share the same posterior rate parameter. Conditional on λ, their counts are independent; after integrating λ, however, they are positively correlated:

$$\text{Cov}(\widetilde{n}_k, \widetilde{n}_j \mid D_0^L) = \text{Var}(\lambda \mid D_0^L) = \frac{a_0}{b_0^2}, \qquad k \neq j. \tag{A.16}$$

Consequently, the total new-class count satisfies

$$\widetilde{N}_t^{\text{new}} = \sum_{k=1}^{K_t} \widetilde{n}_k \mid D_0^L \sim \text{NB}\left(a_0, \frac{b_0}{b_0 + K_t}\right), \tag{A.17}$$

with moments

$$\mathbb{E}[\widetilde{N}_t^{\text{new}} \mid D_0^L] = \frac{K_t a_0}{b_0}, \qquad \text{Var}(\widetilde{N}_t^{\text{new}} \mid D_0^L) = \frac{K_t a_0}{b_0} + \frac{K_t^2 a_0}{b_0^2}. \tag{A.18}$$

The first parameter in Eq. (A.17) remains a_0, not $K_t a_0$, because the model integrates one shared rate parameter rather than independent rate parameters for the future classes.

A.5 PREDICTIVE MARGINAL DOMAIN

Let $\widetilde{L}_1 = F_1^{-1}(0.025)$ and $\widetilde{U}_1 = F_1^{-1}(0.975)$ be generalized quantiles of the unconditional single-class predictive cumulative distribution in Eq. (A.13). Define L_Σ and U_Σ analogously for the total predictive cumulative distribution. Under the stated model, the raw discrete equal-tail intervals have posterior-predictive coverage at least 0.95, although not necessarily exactly 0.95. The implemented single-class bounds are

$$L_1 = \max(1, \widetilde{L}_1), \qquad U_1 = \widetilde{U}_1. \tag{A.19}$$

The maximum with 1 encodes the known presence of a new class as a hard feasibility constraint; it does not renormalize the predictive distribution by conditioning on class presence. The effective total range is determined by intersecting the total predictive interval, the single-column range, and the current stream size:

$$L_\Sigma^* = \max(L_\Sigma, K_t L_1), \qquad U_\Sigma^* = \min(U_\Sigma, K_t U_1, N_t). \tag{A.20}$$

The background-column bounds follow from conservation of total mass:

$$l_0 = \max(0, N_t - U_\Sigma^*), \qquad U_0 = \min(N_t, N_t - L_\Sigma^*), \tag{A.21}$$

and, for new-class columns $k \geq 1$,

$$l_k = L_1, \qquad U_k = U_1. \tag{A.22}$$

The implementation checks $l \leq U$, $\sum_c l_c \leq N_t$, and $\sum_c U_c \geq N_t$, and requires the soft reference vector to be positive:

$$\nu_k = \mu_\lambda \quad (k \geq 1), \qquad \nu_0 = N_t - K_t \mu_\lambda > 0, \qquad \pi = \nu / N_t. \tag{A.23}$$

Reference positivity and domain feasibility are distinct requirements. The condition $\nu_0 = N_t - K_t\mu_\lambda > 0$ ensures positive reference masses and well-defined logarithms, whereas $\mathcal{C} \neq \emptyset$ ensures that the assignment constraints admit a solution. A nonpositive background reference does not by itself imply that the constraint set is empty. BAP requires both conditions; the implementation checks and failure behavior are described in Appendix C.2.

The predictive intervals define the feasible domain, and the posterior mean defines the soft reference. The final integer counts are obtained by optimization and rounding within the feasible domain, not by directly rounding μ_λ.

Three interpretations must be distinguished. First, the raw intervals have marginal predictive meaning under the adopted Poisson working model. Second, restricting the count lower bound to a positive value and intersecting with N_t are deterministic restrictions, not a recomputation of the predictive distribution conditional on all new classes being present and on the observed batch total. Third, the intersection of several marginal intervals does not have joint 95% coverage. For example, when the S_0 counts are $1, 1$, $a_0 = 2.5$ and $b_0 = 2$: the raw single-class interval is $[0, 5]$, whereas the implemented interval $[1, 5]$ contains only about 62.50% of the unconditional predictive probability mass. For the recorded CIFAR prior, the raw lower quantile is 361, so this positivity restriction does not change that run.

The residual $\nu_0 = N_t - K_t\mu_\lambda$ is a constructed background reference, not an independently learned posterior mean for the old-class population and not a predictive mean conditioned on the resulting feasible domain. This work does not specify a probabilistic model for the current background population. Observing balanced base-class counts alone does not establish that the Poisson observation scale transfers; the predictive interpretation above is conditional on this working model.

A.6 CALIBRATION IS NOT PREDICTIVE-LIKELIHOOD MINIMIZATION

For a nonnegative integer vector of novel counts $\boldsymbol{n}$, let $Q = \sum_{k=1}^{K} n_k$. Its actual shared-rate joint predictive probability mass is

$$p(\boldsymbol{n} \mid D_0^L) = \frac{\Gamma(a_0 + Q)}{\Gamma(a_0) \prod_k n_k!} \frac{b_0^{a_0}}{(b_0 + K)^{a_0 + Q}}. \tag{A.24}$$

This expression is obtained by integrating the product of the K conditionally independent Poisson likelihoods against the Gamma posterior. Conditional on Q, the novel counts follow a multinomial

distribution with class probabilities $1/K$; summing over all vectors with total Q recovers Eq. (A.17). The negative logarithm of Eq. (A.24) contains $\sum_k \log(n_k!)$ and a coupled $\log\Gamma(a_0 + Q)$ term. It is not $D_{\mathrm{KL}}(q\|\nu)$, which is a continuous-mass regularizer, contains a constructed background reference, and is centered at the predictive mean. Thus BAP uses Bayesian prediction to calibrate a constrained optimization problem. It does not minimize the exact count-predictive negative log likelihood, nor does it perform exact joint Bayesian inference over partitions and current counts.

B Information Projection and Optimization Proofs

B.1 Regularized information projection

For fixed current costs $C \in \mathbb{R}^{N\times J}$ with $J = K_t + 1$, define the local row distribution

$$R_{ic} = \frac{\pi_c \exp(-C_{ic}/\epsilon)}{\sum_j \pi_j \exp(-C_{ij}/\epsilon)}. \tag{B.1}$$

Let $Z_i = \sum_c \pi_c \exp(-C_{ic}/\epsilon)$. For any row-stochastic T,

$$\langle T, C\rangle + \epsilon \sum_{ic} T_{ic} \log \frac{T_{ic}}{\pi_c} = \epsilon \sum_{ic} T_{ic} \log \frac{T_{ic}}{R_{ic}} - \epsilon \sum_i \log Z_i. \tag{B.2}$$

The last term is independent of T. Therefore, the fixed-cost transport subproblem implemented by BAP is equivalent to

$$\min_{T\in\mathcal{C}} \epsilon D_{\mathrm{KL}}(T\|R) \;+\; \rho D_{\mathrm{KL}}(T^\top \mathbf{1}\|\nu), \tag{B.3}$$

up to the same additive constant. Here,

$$D_{\mathrm{KL}}(q\|\nu) = \sum_c \{q_c \log(q_c/\nu_c) - q_c + \nu_c\}. \tag{B.4}$$

If T/N is written as a joint distribution, the same identity holds as long as the corresponding factor N is kept throughout the derivation. The marginal KL term is part of the objective and can be omitted only in special cases, such as $\rho = 0$ or a feasible set in which the marginal q is always fixed.

The fixed-cost projection assumes that all entries of the cost matrix are finite, $\epsilon > 0$, $\rho > 0$, every component of π and ν is positive, $0 \le l \le U$, and the feasible set $\mathcal{C}$ is nonempty.

Existence, uniqueness, and feasible support. Use the convention $0 \log 0 = 0$. On a nonempty $\mathcal{C}$, the finite-valued objective is continuous and the row-stochastic polytope is compact, so a minimizer exists. The function $x \log x$ is strictly convex on $[0, \infty)$; therefore, the positive-weight entropy term is strictly convex in the entire assignment matrix, including boundary points. Adding a convex mass term preserves strict convexity on every convex feasible subset, which proves uniqueness.

A column can be forced to zero by mass conservation even when its specified upper bound is positive. The maximum attainable mass for column c and the maximal feasible column support are

$$\overline{U}_c = \min\left(U_c, N - \sum_{d\neq c} l_d\right), \qquad \mathcal{I} = \{c : \overline{U}_c > 0\}. \tag{B.5}$$

For example, $N = 1$, $l = (1, 0)$, and $U = (1, 1)$ force the second column to zero. In BAP, every row can be assigned to every column, so any feasible mass vector q can be realized by $T_{ic} = q_c/N$. For each column in $\mathcal{I}$, choose a feasible vector with positive mass in that column; averaging these vectors gives a feasible matrix T^+ that is positive on every element of this support.

If an optimizer had a zero entry in $\mathcal{I}$, mixing it with T^+ would give a negative contribution of order $s \log s$ from that zero entry in the entropy term. Changes from positive entries and finite cost terms are $O(s)$; when a formerly zero-mass column becomes positive, it also gives an additional negative $s \log s$ contribution from the mass KL term. For sufficiently small $s > 0$, the objective would therefore decrease, a contradiction. Hence $T^*_{ic} > 0$ for every row i and every $c \in \mathcal{I}$.

On this reduced support, the objective is differentiable at T^* and the constraints are polyhedral. The polyhedral normal-cone optimality condition (Hindi, 2004, Section IV) therefore supplies finite

equality multipliers and active-bound multipliers. Equivalently, a positive feasible point on the reduced support satisfies the refined Slater condition: affine inequalities need not be strict. Combining the derivative of the mass term with the bound multipliers gives

$$\beta_c = \rho \log(q_c/\nu_c) + \eta_c^+ - \eta_c^-, \quad \eta_c^+(q_c - U_c) = 0, \quad \eta_c^-(l_c - q_c) = 0, \quad \eta_c^\pm \geq 0. \tag{B.6}$$

Eliminating the row multipliers gives, on the reduced support,

$$T_{ic}^* = \frac{R_{ic}\exp(-\beta_c/\epsilon)}{\sum_{j\in\mathcal{I}} R_{ij}\exp(-\beta_j/\epsilon)} \quad (c \in \mathcal{I}), \qquad T_{ic}^* = 0 \quad (c \notin \mathcal{I}). \tag{B.6a}$$

The normalization in Eq. (B.6a) is restricted to $\mathcal{I}$, with zero entries outside it; taking log ratios between reachable columns yields the relative-assignment correction in Eq. (5) of the main text. Positive active bounds, including fixed-positive faces, do not preclude finite potentials. The primal problem has a unique minimizer; the dual representation can have gauge freedom, for example when all column masses are fixed. This proves Proposition 1, while keeping distinct the notions of interior row entries and interior column masses.

B.2 Fenchel dual and finite potentials

For each column, define

$$\phi_c(q) = \rho\{q\log(q/\nu_c) - q + \nu_c\} + I_{[l_c,U_c]}(q), \tag{B.7}$$

where I is the indicator function of the interval. Its Fenchel conjugate is

$$\phi_c^*(\beta_c) = \beta_c q_c^*(\beta_c) - \rho\{q_c^*(\beta_c)\log(q_c^*(\beta_c)/\nu_c) - q_c^*(\beta_c) + \nu_c\}, \tag{B.8}$$

where

$$q_c^*(\beta_c) = \mathrm{clip}\{\nu_c\exp(\beta_c/\rho),\, l_c,\, U_c\}. \tag{B.9}$$

Using the same sign convention as the implementation, the differentiable clipped dual function minimized by the solver is

$$\mathcal{D}(\beta) = \epsilon\sum_i \log\sum_c \pi_c \exp\left[-\frac{C_{ic}+\beta_c}{\epsilon}\right] + \sum_c \phi_c^*(\beta_c). \tag{B.10}$$

Given β, the associated row-stochastic assignment is

$$T_{ic}(\beta) = \frac{\pi_c\exp[-(C_{ic}+\beta_c)/\epsilon]}{\sum_j \pi_j\exp[-(C_{ij}+\beta_j)/\epsilon]}. \tag{B.11}$$

Its gradient is

$$\nabla_c\mathcal{D}(\beta) = q_c^*(\beta_c) - \sum_i T_{ic}(\beta). \tag{B.12}$$

Away from clipping thresholds, its Hessian is

$$\nabla^2\mathcal{D}(\beta) = \frac{\mathrm{diag}(q) - T^\top T}{\epsilon} + \mathrm{diag}(d_c), \qquad d_c = \frac{q_c^*}{\rho}\mathbf{1}\{l_c < \nu_c e^{\beta_c/\rho} < U_c\}. \tag{B.13}$$

The covariance term is positive semidefinite. The conjugate function is continuously differentiable, but its second derivative need not exist at a clipping threshold. At such points, a generalized derivative can be chosen from $[0, q_c^*/\rho]$; the implementation uses the zero endpoint. This is the matrix used in the semismooth refinement, not a statement that the objective is globally twice differentiable. Damping changes the numerical search direction, not the objective. A positive β_c lowers the row logit of that column and increases its unclipped conjugate mass, and the two effects balance at a stationary point. Relative odds are governed by differences $\beta_c - \beta_d$.

Dual sign and boundary constants. Introduce $q = T^\top\mathbf{1}$ and use the Lagrangian term $\beta^\top(T^\top\mathbf{1} - q)$. This gives the dual objective $-\mathcal{D}(\beta)$. If every column is attainable, strong duality gives

$$P^* = -\min_\beta \mathcal{D}(\beta), \qquad P(T) + \mathcal{D}(\beta) \geq 0 \quad (T \in \mathcal{C}), \tag{B.14}$$

where P is the fixed-cost primal objective. A residual such as $\|\nabla\mathcal{D}\|_\infty/N$ is not itself an objective-gap bound.

For forced-zero columns, finite potentials on the original full softmax cannot produce exact zeros. The problem should then be reduced to $\mathcal{I}$ as above. Retain the original positive weights π_c for $c \in \mathcal{I}$ (they need not sum to 1 on this subset), and retain the omitted mass-KL constant $\rho \sum_{c\notin\mathcal{I}} \nu_c$. Thus $P^* = \rho \sum_{c\notin\mathcal{I}} \nu_c - \min \mathcal{D}_{\mathcal{I}}$. If the remaining π is renormalized, an additional row-constant shift is introduced and must be included when objective values are compared. Alternatively, the full-space dual may approach its infimum only as the potentials of the removed columns diverge. The directly audited CIFAR feasible domains contain no forced-zero columns, so the reported solver calls do not require this reduction.

These identities require mathematical primal feasibility. A floating-point plan accepted within mass tolerances can still contain tiny boundary violations. In that case, evaluating the objective without the indicator function and adding $\mathcal{D}$ is only a diagnostic quantity, not a certified nonnegative duality gap for the exact constrained problem.

B.3 Gaussian evidence integration and MM descent

For a new-class slot k, write

$$q_k = \sum_i T_{ik}, \qquad h_k = \sum_i T_{ik} u_i, \qquad s_k = \sum_i T_{ik}\|u_i\|^2. \tag{B.15}$$

BAP's auxiliary model is

$$u_i \mid \theta_k \sim \mathcal{N}(\theta_k, I), \qquad \theta_k \sim \mathcal{N}(0, \kappa^{-1} I), \tag{B.16}$$

where the implementation parameter b_{geom} satisfies $b_{\text{geom}} = 1/\kappa$. For fixed real-valued responsibilities, this is a power-likelihood auxiliary posterior proportional to $p(\theta_k)\prod_i p(u_i \mid \theta_k)^{T_{ik}}$, not an exact posterior obtained by integrating over all discrete assignments. Completing the square and integrating θ_k gives

$$\begin{aligned} G_k(T) &= \log \int p(\theta_k) \exp\left(\sum_i T_{ik} \log p(u_i \mid \theta_k)\right) d\theta_k \\ &= -\frac{q_k d}{2}\log(2\pi) - \frac{s_k}{2} + \frac{\|h_k\|^2}{2(\kappa+q_k)} - \frac{d}{2}\log\left(1+\frac{q_k}{\kappa}\right). \end{aligned} \tag{B.17}$$

The auxiliary posterior mean and covariance of θ_k are

$$m_k = \frac{h_k}{\kappa+q_k}, \qquad V_k = \frac{I}{\kappa+q_k}. \tag{B.18}$$

The derivative of $-G_k$ with respect to responsibility T_{ik} is

$$\frac{\partial[-G_k(T)]}{\partial T_{ik}} = \frac{1}{2}\|u_i - m_k\|^2 + \frac{d}{2(\kappa+q_k)} + \frac{d}{2}\log(2\pi). \tag{B.19}$$

Writing $p_k(\theta_k \mid T)$ for the auxiliary mean posterior, differentiation of the log evidence gives

$$C_{ik} = \mathbb{E}_{p_k(\theta_k\mid T)}[-\log p(u_i \mid \theta_k)] = -\frac{\partial G_k(T)}{\partial T_{ik}}.$$

This is an expectation of negative log likelihood, rather than the negative log of a posterior-predictive density. The latter has covariance $I + V_k$; V_k itself is uncertainty in the auxiliary mean, not the full covariance of the final Gaussian class memory. The evidence gradient explains the trace-uncertainty term and supplies the new-class costs for the MM tangent surrogate below.

The complete auxiliary objective $F(T)$ is Eq. (9) in the main text, with $G_k(T)$ given by Eq. (B.17) and $I_{\mathcal{C}}$ the feasible-domain indicator. The historical background term g_0 is defined by Eq. (8) from statistics saved before the current stage and remains fixed throughout this stage. Hence its contribution is linear in T_{i0}, with cost $C_{i0} = -\log g_0(u_i)$; no tangent approximation is needed for this term. New-class evidence-gradient costs and fixed historical predictive costs therefore serve the same

auxiliary objective, but are not a common posterior-predictive classification score for every class. The uniform historical weights $1/O_t$ in Eq. (8) are a background-model choice. Stored counts and feature sums affect the class posteriors, but stored counts are not estimates of current-stream class arrival frequencies. These mixture weights are not implied by Proposition 3.

The map $T_{\cdot k} \mapsto G_k(T)$ is the log integral of an exponential family whose natural parameters are affine in $T_{\cdot k}$; it is therefore convex by convexity of the log-partition function. Hence $-G_k$ is concave, and its first-order Taylor expansion is a global upper bound. Let $Q(T \mid T^{(r)})$ denote the objective obtained by replacing each $-G_k(T)$ with this tangent upper bound while keeping the entropy, marginal KL, historical background term, and feasible domain unchanged. If the initial iterate is feasible and the projection subproblem is solved over the same $\mathcal{C}$, then

$$F(T^{(r+1)}) \le Q(T^{(r+1)} \mid T^{(r)}) \le Q(T^{(r)} \mid T^{(r)}) = F(T^{(r)}). \tag{B.20}$$

The upper bound is tight at the current feasible iterate. More generally, if an inexact update satisfies $Q(T^{(r+1)} \mid T^{(r)}) \le Q(T^{(r)} \mid T^{(r)}) + \delta_r$, then $F(T^{(r+1)}) \le F(T^{(r)}) + \delta_r$. A small transport stationarity residual alone is not enough to establish this inequality. The implementation instead checks the computed full objective value and allows it to increase by at most $10^{-9}\max(1, |F(T^{(r)})|)$; this is a floating-point acceptance rule, not an exact theorem. The initial one-hot proposal $T^{(0)}$ need not belong to $\mathcal{C}$. Its class statistics define $C^{(0)}$, and the first constrained projection produces $T^{(1)} \in \mathcal{C}$. The implementation then evaluates the first finite auxiliary objective; its non-increase comparisons begin with $F(T^{(2)})$ versus $F(T^{(1)})$. For exact updates, Eq. (B.20) applies from feasible iterates $r \ge 1$. When $T^{(0)}$ is infeasible, $I_{\mathcal{C}}(T^{(0)})$ is infinite, so the initial transition is not a finite-objective descent comparison. The same convention applies to each candidate initialization.

This proves the exact-update descent conclusion in Proposition 2 under fixed stage, fixed prior, fixed background statistics, fixed candidate initialization, and feasible starting point. The conclusion does not extend automatically to capacity rounding, soft-to-hard assignment conversion, the final full-Gaussian fit, objectives across stages, or test accuracy. The separate finite-bank guarantee for the recomputed hard auxiliary objective remains valid, as established in Appendix B.4.

B.4 Integer rounding, hard assignment, and finite-bank selection

Soft feasibility and integer feasibility are different conditions. Integer capacities require

$$\sum_c \lceil l_c \rceil \le N_t \le \sum_c \lfloor U_c \rfloor. \tag{B.21}$$

For example, $N = 1$ and $l = U = (1/2, 1/2)$ are soft feasible but admit no integer capacity vector. All BAP endpoints in the implementation are integers because they are derived from discrete quantiles and integer conservation; therefore, the checked soft feasible domains used by BAP also satisfy Eq. (B.21).

For an exactly feasible q within these integer bounds, initialize $\bar{q}_c = \lfloor q_c \rfloor$ and let $r = N_t - \sum_c \lfloor q_c \rfloor$. If $r > 0$, increase by one the r components with the largest fractional remainders, breaking ties by column index. Every positive remainder satisfies $\lfloor q_c \rfloor < U_c$, and there are enough such components because their fractional parts sum to the integer r and each fractional part is less than one. Therefore, $l \le \bar{q} \le U$ and $\sum_c \bar{q}_c = N_t$. The implementation also clips and checks the result to handle small numerical feasibility errors. This proof applies to the integer feasible domains actually used; it does not claim that a one-pass clipped remainder rule solves quota allocation for arbitrary real-valued bounds.

The subsequent hardening step solves

$$\min_{Y \in \{0,1\}^{N_t \times J}} \sum_{ic} Y_{ic} C_{ic} \quad \text{s.t.} \quad Y\mathbf{1} = \mathbf{1}, \qquad Y^\top \mathbf{1} = \bar{q}. \tag{B.22}$$

After reversing the signs of the column-balance constraint rows and omitting one redundant balance equation, this equality-constraint matrix is the node–edge incidence matrix of a directed bipartite transportation network. The matrix is totally unimodular and the right-hand side is integral. Therefore, every extreme point of the linear relaxation is integral, and a linear objective has an optimal integral extreme-point solution. An arbitrary convex combination of optimal extreme points can still

be fractional, so the actual solver output must be checked rather than assuming that every optimal assignment is integral. This is why the implementation solves a global linear program and rejects fractional outputs instead of taking row-wise argmax decisions.

Hardening optimizes a linear assignment problem under fixed integer capacities and fixed costs. It does not preserve the nonlinear soft-assignment optimum. BAP therefore recomputes the full hard-assignment objective for every candidate in the finite candidate bank, including the integrated Gaussian term, the entropy-to-prior term, the marginal KL, and the same feasible-domain check. The selected candidate has a hard-assignment auxiliary objective no greater than that of the baseline candidate within the same candidate bank. This is an optimization statement over a finite candidate bank, not a guarantee of test accuracy.

The subsequent anonymous alignment only permutes new-class columns. In the BAP feasible domains used in the implementation, all new-class columns share the same π_k, ν_k, l_k, U_k and geometric prior, while the background column is fixed. Hence the sums of evidence, entropy, and mass KL, as well as the feasible domain, are invariant under this permutation. Alignment therefore preserves the selected hard-assignment objective and feasibility. If unequal class-specific references or bounds were used, this invariance would no longer hold unless those parameters were permuted consistently as well.

B.5 Boundary of the mathematical claims

The full auxiliary objective is generally nonconvex. Uniqueness of the fixed-cost projection does not imply uniqueness of the joint category solution, global optimality, or an accuracy guarantee. In the implementation limit $b_{\mathrm{geom}} = 0$, the mean prior degenerates to a point mass at zero, so $m_k = 0$ and $V_k = 0$, and the last two terms of Eq. (B.17) vanish. The evidence is then linear, and the same majorization argument still applies in this limit.

The Bayesian count-predictive model, the auxiliary Gaussian mean posterior, the soft information projection, and the final native full-Gaussian writer/reader are distinct objects. BAP does not perform exact Bayesian integration over all possible category partitions, and the auxiliary posterior variance in Eq. (B.18) is not the native class covariance used by the final reader. The proofs above justify the implemented auxiliary inference and hardening steps under the stated feasibility and fixed-stage assumptions.

C Implementation and Reproducibility

Figure 1 summarizes the stage update used by Bayesian Assignment Projection (BAP). The implementation reuses the existing representation assets and the native writer and reader, and changes only the assignment inference supplied to the second native fit in each stage.

C.1 Assets and Fixed Representations

All reported BAP runs start from the frozen S0 memory and cached stream assets. The native Gaussian writer receives the fixed 384-dimensional PCA writer features used by the reference implementation. Initial clusters for the current stage are produced by the cached Gaussian-mixture initialization in the corresponding raw 768-dimensional representation. The auxiliary projection is performed in the same PCA space and uses the S0 geometry file for whitening: if z_i is a PCA feature, then

$$u_i = L^{-1} z_i - m_0.$$

Here $C_0 = LL^\top$ is the shared within-class covariance of the base classes, and m_0 is the average of the base-class means in the whitened space. The auxiliary likelihood and mean prior use Eq. (B.16); the main-text precision κ_0 and the appendix precision κ denote the same parameter. The auxiliary mean posterior is used for assignment inference, while the final class memory continues to use the native full-covariance model.

The auxiliary geometry is estimated once using only labeled S0 observations. Let $n_\star$ be the common per-S0-class sample count used in the geometry estimate, let $\bar{z}_c$ be the class mean in the whitened

PCA coordinate system, and let m_0 be the mean of these class means. The implementation uses

$$b_{\text{geom}} = \max\left\{0, \frac{1}{dK_0}\sum_{c=1}^{K_0}\|\bar{z}_c - m_0\|^2 - \frac{1}{n_\star}\right\}, \qquad \kappa = b_{\text{geom}}^{-1}. \tag{C.1}$$

The limiting case $b_{\text{geom}} = 0$ is treated in the auxiliary model as an infinite-precision mean prior; the released CIFAR/CUB geometry checks have $b_{\text{geom}} > 0$. This geometry estimator is independent of the S0 cardinality posterior in Appendix A.

C.2 Stage Execution

At stage t, let the pre-stage native parent contain classes $0, \ldots, O_t - 1$ and let K_t new-class slots be available. The runner first performs one native fit using the current-stage writer features and cached GMM labels. The resulting native model is used only to decide which sample rows are initially admitted as current new-class candidates. BAP search then constructs a row-stochastic initialization: admitted rows are assigned to their GMM new-class slots, and rejected rows are assigned to the background column.

The base stage initializes the native memory M_0, the geometric and cardinality priors, and the auxiliary counts and feature sums for base classes. For an already written class c, let n_c and h_c denote its auxiliary count and feature sum. Its auxiliary mean posterior is

$$m_c = \frac{h_c}{\kappa + n_c}, \qquad V_c = (\kappa + n_c)^{-1} I. \tag{C.2}$$

Substituting these statistics into Eq. (8) gives the fixed historical background for the current stage. After each stage, the final hard assignment appends new auxiliary counts and feature sums for use in later background computations.

The count inputs for each stage are the fixed S0 posterior, the observed batch size N_t, and the known number of new classes K_t. Predictive intervals, nonempty new classes, and mass conservation determine l, U; the posterior mean determines ν, π as in Appendix A.5. Per-class benchmark sampling caps are not passed to this construction.

The prior constructor checks the effective total interval, positivity of $\nu_0 = N_t - K_t\mu_\lambda$, and the column bounds $l \le U$, $\sum_c l_c \le N_t \le \sum_c U_c$. Invalid intervals or column bounds raise an infeasibility error; a nonpositive reference raises a separate incompatible-prior error. The runner records the error and stops the affected stage. It does not construct a fallback prior or automatically relax the bounds. BAP is defined on inputs satisfying both reference positivity and constraint feasibility, rather than arbitrary batch partitions or arbitrarily sparse inputs.

The second native fit is rebuilt independently from the pre-stage parent, rather than updated from the result of the first fit. Historical native parameters are checked by content hashes before and after fitting. A POST checkpoint is written after every native call; if a fit intent already exists without the corresponding POST checkpoint, the runner refuses to repeat that fit automatically.

The defaults were adopted as design settings: $\epsilon = 1$ uses the Gaussian negative-log-density cost without temperature rescaling, while $\rho = 1$ assigns unit weight to the marginal KL penalty. The nominal predictive level is specified as 95%; labeled base-class counts determine the predictive distributions and the corresponding interval endpoints. Each candidate receives a fixed computational budget of at most 50 MM updates, with early stopping when the relative change in the auxiliary objective is at most 10^{-8}. Appendix F reports sensitivity analyses for ϵ and the predictive level.

The following procedure is repeated for each incremental stage.

C.3 Search, Transport Solver, and Hardening

BAP uses the same candidate-search rule at every stage. The base candidate starts from the original one-hot $T^{(0)}$; MM first projects it into the feasible domain and then refines it. For each new-class column of the resulting base solution, the implementation proposes one weighted spectral split and one compensating merge of the closest pair among the other active new-class columns. The split direction is the leading eigenvector of the weighted scatter matrix of the source column, followed

Table S2: Implementation settings for BAP.

Item	Value	Role
Writer feature dimension	384	Input to the native full-Gaussian writer and reader
GMM feature dimension	768	First-fit initialization in the raw space
Base / increment / total classes	50 / 10 / 100	CIFAR-100 T5 protocol
Native optimization steps	1000	Official SVI loop budget for each fit
Native learning rate	4×10^{-6}	Initial value for the Adam schedule
Projection weights	$\epsilon = 1, \rho = 1$	Weights of the entropy and marginal KL terms
MM budget	50 iterations	Per candidate; relative tolerance 10^{-8}
Transport tolerance	2×10^{-8}	Normalized stationarity-residual threshold
Row-sum tolerance	10^{-10}	Soft-assignment check
Soft mass tolerance	$\max(10^{-5}, 2N\,10^{-8})$	Bound-violation check
Integer capacity tolerance	0	Exact preservation of inferred integer counts

Algorithm S1 One BAP stage

Require: Pre-stage native parent P_{t-1}, current features X_t, GMM labels q_0, S0 geometry $(L, m_0, b_{\text{geom}}, n, h)$, and S0 count posterior.

1: $\theta_{\text{first}} \leftarrow$ first guarded native fit from P_{t-1} using q_0.
2: $A^{(0)} \leftarrow$ rows predicted by θ_{first} as current-stage classes.
3: Derive π, ν, l, U from the S0 posterior, N_t, and K_t.
4: $u_i \leftarrow L^{-1} z_i - m_0$; $g_0(u_i) \leftarrow$ old-class auxiliary mixture.
5: $T^{(0)} \leftarrow$ one-hot background/new initialization from $A^{(0)}$ and q_0.
6: Run Algorithm S2 to alternate updates, round capacities, solve each hard-assignment LP, and select Y by the complete hard auxiliary objective.
7: $Y_t \leftarrow$ selected hard labels after aligning anonymous new-class columns by overlap with the fixed q_0.
8: $\theta_{\text{second}} \leftarrow$ second guarded native fit from the same P_{t-1} using rows with $Y_t > 0$ and their hard labels.
9: Append auxiliary counts and sum statistics from Y_t; predict the fixed query set with the updated memory.

Ensure: Updated native parent θ_{second} and auxiliary history statistics.

by ten weighted Lloyd updates. Degenerate zero-mass columns, zero scatter, unavailable merge pairs, and empty weighted subclusters are skipped directly rather than repaired through additional thresholds.

For a nonsaturated domain, every fixed-cost transport solve initializes $\beta = 0$ and minimizes the clipped dual by L-BFGS-B for at most 10000 iterations. It then applies up to 50 semismooth Newton refinements using the Hessian in Appendix B; each Newton step tries at most 30 backtracking steps. After reconstructing the plan by a row-wise softmax, the solver checks the normalized stationarity residual, row sums, and mass bounds. A saturated subproblem with fixed column masses uses the log-domain Sinkhorn solver. This numerical branch changes neither the S0 calibration nor the information used to construct the bounds.

After MM terminates, capacities are rounded by the largest-remainder rule after clipping to the integer domain. The hard assignment is one sparse transportation linear program over all sample rows and fixed capacities; fractional output from the linear program is treated as an error. The final winning candidate is selected by the complete hard auxiliary objective, not by row-wise argmax or evaluation accuracy.

Computation and storage. Each incremental stage performs two native Gaussian fits. The candidate search only adds auxiliary inference. The soft-assignment matrix requires $O(N_t(K_t + 1))$ storage, and each projection has at most $K_t + 1$ dual variables. If J_t classes have been accumulated, the implementation stores one count and one d-dimensional feature sum per class in addition to the native Gaussian memory, giving $O(J_t d)$ extra persistent storage.

Algorithm S2 Finite-bank BAP search

Require: Auxiliary features u, old-class log density $\log g_0(u_i)$, initialization $T^{(0)}$, prior π, ν, and domain (l, U).

1: Run MM from $T^{(0)}$ to obtain the baseline soft assignment; round capacities, solve the global hard-assignment LP, and compute $F(Y)$.
2: Add the baseline hard assignment and its complete objective value to the finite bank.
3: **for** each new-class column j **do**
4: Construct the weighted scatter matrix of column j and its leading split direction.
5: Run ten weighted Lloyd updates for the two children obtained by the split.
6: Merge the closest pair among the other active new-class columns.
7: If no skip condition is triggered, run MM, capacity rounding, and the global hard-assignment LP with the same budget; recompute $F(Y)$ and add the result to the finite bank.
8: **end for**
9: Select the candidate with the smallest recomputed complete hard auxiliary objective.

C.4 Native Writer, Guards, and Reader

The second fit starts from the pre-stage memory M_{t-1}, uses the novel support $A' = \{i : Y_{i0} = 0\}$ and its hard new-class labels, and adds only current-stage class parameters. With one-based mathematical class numbering, the memory update and test-time reader are

$$M_t = M_{t-1} \cup \{(\mu_c, \Sigma_c)\}_{c=O_t+1}^{O_t+K_t}, \qquad \hat{y}(x) = \arg \max_{1 \le c \le O_t+K_t} \log \mathcal{N}(z; \mu_c, \Sigma_c). \tag{C.3}$$

The second fit does not inherit first-fit new-class parameters, and old-class parameters are kept fixed. The final reader uses full class covariances. At test time, it scores each feature directly and does not project the test batch according to class-size constraints.

The native writer is the official full-Gaussian VB-CGCD writer wrapped by domain guards. Every first or second fit uses the native SVI/Adam path, the official slot parameterization, the native schedule, and the stage parent as the global parameters. The guarded loop records the last valid SVI state; when a candidate update produces non-finite values, non-positive-definite active covariances, a loss-computation failure, or a changed historical prefix, the loop stops or rolls back. Before reader scoring, all class covariances are validated again. This learning path has no access to ground-truth labels; ground-truth labels are used only by the independent evaluator after prediction files have been sealed.

D Multi-Initialization Results and Comparison Sources

The following statistics keep the feature representation and data stream fixed and vary only the incremental initialization. Table 1 corresponds to the fixed-initialization run; means and standard deviations are collected in this appendix.

Table D1: Final overall recognition accuracy across incremental initializations.

Dataset	VB-CGCD	BAP	Mean gain
C100	79.79 ± 1.47	82.79 ± 1.33	$+3.00$
Tiny	74.23 ± 0.22	76.57 ± 0.81	$+2.33$
IN100	85.49 ± 1.64	89.48 ± 0.97	$+3.99$
CUB	65.94 ± 1.03	67.32 ± 0.78	$+1.38$

The paired mean gain across the four datasets is 2.68 percentage points, which is the multi-initialization statistic reported in the abstract; the fixed-initialization mean gain in Table 1 is 3.28 percentage points. On CUB, Old increases by 1.82 percentage points on average, whereas New and H decrease by 2.69 and 1.26 percentage points, respectively, showing that an overall gain does not imply improvement in every component.

The control is the reproduced VB-CGCD run that shares the representation, data stream, and evaluation mapping with BAP. Table D4 repeats the matched measurements in Table 2 as a reference

Table D2: Mean gains in final-stage recognition metrics across incremental initializations.

Dataset	ΔAll	ΔOld	ΔNew	ΔH
C100	+3.00	+2.78	+5.00	+4.03
Tiny	+2.33	+1.91	+6.10	+4.43
IN100	+3.99	+3.70	+6.60	+5.15
CUB	+1.38	+1.82	−2.69	−1.26

Table D3: Sources and evaluation settings for the main comparison table.

Method	Result source	Evaluation note
PromptCCD (Cendra et al., 2024); Happy (Ma et al., 2024)	Table 2 of Dai & Chauhan (2025), PDF p. 7.	We keep the stage-wise values from the cited source and do not replace them with results for methods with the same names in other papers.
GOAL (Han et al., 2026)	Table 1 of the AAAI 2026 official paper, p. 4569 (PDF p. 5).	We use the S0–S5 stage columns; the original "1–5 Stages Avg" column is not used as an endpoint score.
VC-CGCD (Xiong et al., 2026)	Author-released full text, arXiv:2607.04984v1: Table 1 (PDF p. 12) and CUB Table D3 (p. 22).	We use stage-wise results; the cross-stage mean in Table D2 is not used as the S5 score.
VB-CGCD† (Dai & Chauhan, 2025); BAP	The fixed-initialization run in this paper and the corresponding multi-initialization records.	The runs share the fixed representation, data stream, and birth-stage semantic mapping; incremental stages maintain their own historical memories.

Table D4: Matched forgetting and discovery metrics for VB-CGCD and BAP.

Dataset	M_f (control)	M_f (BAP)	M_d (control)	M_d (BAP)
C100	6.48	5.08	79.04	83.32
Tiny	5.88	4.80	67.56	72.86
IN100	5.52	4.52	83.80	92.96
CUB	6.66	3.99	52.10	52.36

for the comparison summaries in this appendix. M_f is measured in percentage points and M_d in percent. The corresponding ΔM_d values for the paired runs are 4.28, 5.30, 9.16, and 0.26. Upper-Bound is separately quoted from Table 3 (p. 8) of Dai & Chauhan (2025); its two-part, two-stage supervised setting is described in their Section 4.2 (p. 7).

The literature-baseline rows in Table 1 preserve source-reported values and their original training and evaluation settings. The VB-CGCD reproduction and BAP rows identified as our runs report results from this paper. The VB-CGCD/BAP forgetting and discovery comparisons in Tables 2 and D4 use our runs under the same representation, data stream, and evaluation mapping.

D.1 Stage-Wise Multi-Initialization Results

Table D5: Stage-wise recognition results on C100 across incremental initializations.

Method	Stage	All	Old	New	H
VB-CGCD	S1	87.01 ± 1.09	89.61 ± 0.58	74.00 ± 6.72	80.95 ± 4.01
	S2	83.96 ± 1.90	85.23 ± 1.29	76.30 ± 5.74	80.47 ± 3.71
	S3	82.18 ± 1.75	82.66 ± 1.76	78.83 ± 2.84	80.69 ± 2.06
	S4	81.38 ± 1.47	81.54 ± 1.71	80.10 ± 0.46	80.81 ± 0.62
	S5	79.79 ± 1.47	80.85 ± 1.53	70.30 ± 1.15	75.20 ± 1.27
BAP	S1	89.00 ± 0.08	90.60 ± 0.04	81.00 ± 0.61	85.53 ± 0.33
	S2	85.51 ± 2.04	87.77 ± 0.09	71.97 ± 14.15	78.61 ± 8.18
	S3	84.31 ± 1.72	84.33 ± 1.99	84.13 ± 0.21	84.22 ± 0.88
	S4	84.26 ± 1.51	83.77 ± 1.70	88.13 ± 0.06	85.89 ± 0.88
	S5	82.79 ± 1.33	83.63 ± 1.50	75.30 ± 0.36	79.24 ± 0.55

Table D6: Stage-wise recognition results on Tiny across incremental initializations.

Method	Stage	All	Old	New	H
VB-CGCD	S1	85.17 ± 0.31	86.57 ± 0.20	78.20 ± 1.23	82.17 ± 0.72
	S2	81.98 ± 0.16	83.94 ± 0.10	70.20 ± 0.50	76.46 ± 0.34
	S3	79.75 ± 0.16	81.03 ± 0.12	70.73 ± 0.76	75.53 ± 0.45
	S4	76.43 ± 0.19	78.71 ± 0.22	58.20 ± 2.91	66.89 ± 1.86
	S5	74.23 ± 0.22	75.83 ± 0.15	59.90 ± 2.01	66.92 ± 1.23
BAP	S1	86.34 ± 1.11	87.21 ± 0.03	82.00 ± 6.75	84.43 ± 3.67
	S2	83.61 ± 0.29	85.29 ± 1.13	73.53 ± 4.73	78.90 ± 2.18
	S3	81.79 ± 0.29	82.51 ± 0.32	76.70 ± 0.10	79.50 ± 0.20
	S4	78.39 ± 0.67	81.10 ± 0.30	56.70 ± 8.41	66.50 ± 5.54
	S5	76.57 ± 0.81	77.74 ± 0.57	66.00 ± 4.17	71.35 ± 2.60

Table D7: Stage-wise recognition results on IN100 across incremental initializations.

Method	Stage	All	Old	New	H
VB-CGCD	S1	92.50 ± 1.10	92.81 ± 0.33	90.93 ± 4.98	91.82 ± 2.73
	S2	90.58 ± 2.03	91.06 ± 1.88	87.73 ± 4.59	89.33 ± 2.93
	S3	88.85 ± 2.13	89.30 ± 1.89	85.73 ± 5.03	87.45 ± 3.29
	S4	86.24 ± 1.13	87.67 ± 1.83	74.80 ± 4.50	80.64 ± 1.87
	S5	85.49 ± 1.64	85.47 ± 1.28	85.67 ± 4.97	85.54 ± 3.10
BAP	S1	93.21 ± 0.02	92.93 ± 0.02	94.60 ± 0.00	93.76 ± 0.01
	S2	92.49 ± 0.06	92.44 ± 0.07	92.73 ± 0.12	92.59 ± 0.06
	S3	91.17 ± 0.07	90.94 ± 0.06	92.80 ± 0.20	91.86 ± 0.12
	S4	89.79 ± 1.10	90.21 ± 0.06	86.40 ± 10.39	88.04 ± 5.60
	S5	89.48 ± 0.97	89.17 ± 1.05	92.27 ± 0.31	90.69 ± 0.67

Table D8: Stage-wise recognition results on CUB across incremental initializations.

Method	Stage	All	Old	New	H
VB-CGCD	S1	79.35 ± 0.96	83.86 ± 0.34	56.10 ± 5.96	67.10 ± 4.37
	S2	74.55 ± 1.07	77.86 ± 0.86	54.86 ± 3.79	64.32 ± 2.72
	S3	71.67 ± 0.83	73.03 ± 1.26	61.95 ± 2.62	67.00 ± 1.05
	S4	67.96 ± 1.00	70.75 ± 0.70	45.01 ± 3.61	54.98 ± 2.90
	S5	65.94 ± 1.03	67.15 ± 1.02	54.85 ± 1.41	60.38 ± 1.18
BAP	S1	80.19 ± 1.02	85.21 ± 0.02	54.34 ± 6.19	66.22 ± 4.73
	S2	75.54 ± 1.26	79.31 ± 1.08	53.09 ± 6.29	63.46 ± 4.44
	S3	72.62 ± 1.16	74.68 ± 1.18	57.87 ± 6.34	65.07 ± 3.88
	S4	69.53 ± 0.70	72.02 ± 1.16	49.12 ± 4.06	58.32 ± 2.61
	S5	67.32 ± 0.78	68.97 ± 0.63	52.16 ± 8.55	59.12 ± 5.69

D.2 Protocols and Evaluation

CIFAR-100 and ImageNet-100 both start from 50 base classes and are followed by five increments of 10 new classes each; CUB-200 and TinyImageNet both start from 100 base classes and are followed by five increments of 20 new classes each. The benchmark builders use old/new per-class sampling caps of 25/400, 60/1000, 5/25, and 25/400, respectively. These settings define the data stream only. BAP calibrates cardinality from the S0 observations and the current N, K through the posterior predictive construction in Appendices A.3–A.5; the sampling caps are not additional inputs to its priors, reference masses, or assignment bounds. The ImageNet class list is distributed with the source tables. Representation preparation is fixed, including seed-42 ImageNet and TinyImageNet feature assets.

Table S3 reports the results for each incremental initialization, the single-random-seed Spectral control, and a paired CIFAR count-model ablation block containing the BAP reference and the mean-only variant. Table S4 gives the same-condition VB and BAP stage metrics from that block. All is evaluated on test samples from all classes introduced up to stage t; Old covers earlier introduced class groups, and CurrentNew covers the current class group. Anonymous categories receive a fixed evaluator-side birth-stage mapping from their original GMM assignments. This mapping is shared by the methods compared and is preserved in later stages. The learner-side overlap alignment acts independently on anonymous assignments.

MeanAll averages All over S1–ST. For birth-cohort accuracy $a_{b,t}$,

$$M_f = a_{0,0} - a_{0,T}, \qquad M_d = T^{-1} \sum_t \mathrm{CurrentNew}_t.$$

Each row of Figure S2 tracks a fixed class cohort, while Figure S1 separately shows discovery and retention over the sequence. The source CSVs preserve correct counts and denominators. Figures 2–4 and S2, and Table 5, use the fixed-initialization runs with seed 0.

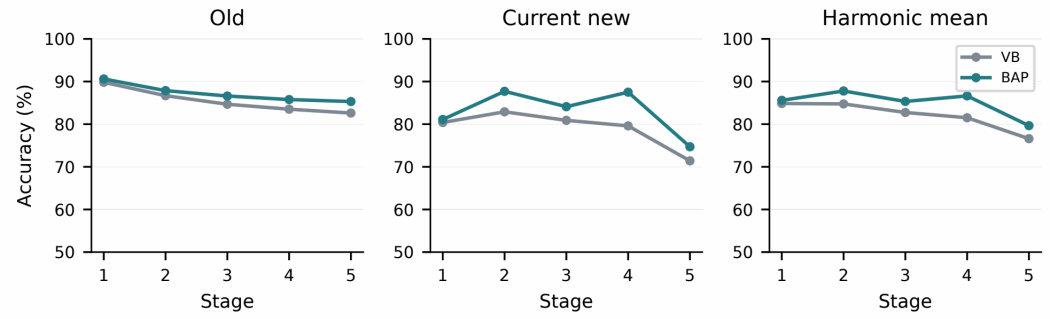


Figure S1: Old, current-new, and harmonic-mean accuracy for VB and BAP in the CIFAR sequence. In all three panels, the horizontal axis is stage and the vertical axis is accuracy (%).

Table S3: Final recognition metrics for individual initializations and the paired CIFAR count-model ablation block.

Dataset	Method	Seed	S5 All	MeanAll	Old	New	H
cifar	BAP	0	84.32	86.70	85.36	75.00	79.84
cifar	BAP	1	82.12	84.49	82.83	75.70	79.11
cifar	BAP	2	81.94	84.33	82.69	75.20	78.77
cifar	Spectral	0	83.48	85.49	84.37	75.50	79.69
cifar	VB	0	81.49	84.63	82.61	71.40	76.60
cifar	VB	1	78.84	81.92	79.92	69.10	74.12
cifar	VB	2	79.05	82.05	80.01	70.40	74.90
cub	BAP	0	66.98	72.06	68.24	55.44	61.18
cub	BAP	1	68.21	73.64	69.26	58.60	63.48
cub	BAP	2	66.76	73.42	69.41	42.46	52.69
cub	Spectral	0	64.22	69.98	65.43	53.16	58.66
cub	VB	0	64.76	70.81	65.98	53.51	59.10
cub	VB	1	66.43	72.39	67.53	56.32	61.42
cub	VB	2	66.62	72.49	67.92	54.74	60.62
imagenet	BAP	0	89.98	91.43	89.73	92.20	90.95
imagenet	BAP	1	88.36	90.77	87.96	92.00	89.93
imagenet	BAP	2	90.10	91.48	89.82	92.60	91.19
imagenet	Spectral	0	85.38	87.99	84.80	90.60	87.60
imagenet	VB	0	85.00	88.70	85.09	84.20	84.64
imagenet	VB	1	87.32	90.31	86.89	91.20	88.99
imagenet	VB	2	84.14	87.19	84.42	81.60	82.99
tiny	BAP	0	77.37	81.26	78.40	68.10	72.89
tiny	BAP	1	75.75	81.29	77.37	61.20	68.34
tiny	BAP	2	76.58	81.46	77.46	68.70	72.82
tiny	Spectral	0	75.21	80.05	76.84	60.50	67.70
tiny	VB	0	74.29	79.50	75.99	59.00	66.43
tiny	VB	1	73.99	79.58	75.71	58.50	66.00
tiny	VB	2	74.42	79.46	75.78	62.20	68.32
cifar	BAP (count-model block)	0	84.25	86.67	85.31	74.70	79.65
cifar	Mean-only	0	82.02	84.71	83.01	73.10	77.74

Table S4: Stage-wise recognition metrics in the paired CIFAR count-model ablation block.

Method	Stage	All	Old	New	H	Correct/total
VB	0	92.10	92.10	–	–	4605/5000
VB	1	88.25	89.82	80.40	84.85	5295/6000
VB	2	86.14	86.68	82.90	84.75	6030/7000
VB	3	84.20	84.67	80.90	82.74	6736/8000
VB	4	83.08	83.51	79.60	81.51	7477/9000
VB	5	81.49	82.61	71.40	76.60	8149/10000
BAP (count-model block)	0	92.10	92.10	–	–	4605/5000
BAP (count-model block)	1	89.02	90.60	81.10	85.59	5341/6000
BAP (count-model block)	2	87.84	87.87	87.70	87.78	6149/7000
BAP (count-model block)	3	86.29	86.60	84.10	85.33	6903/8000
BAP (count-model block)	4	85.97	85.78	87.50	86.63	7737/9000
BAP (count-model block)	5	84.25	85.31	74.70	79.65	8425/10000

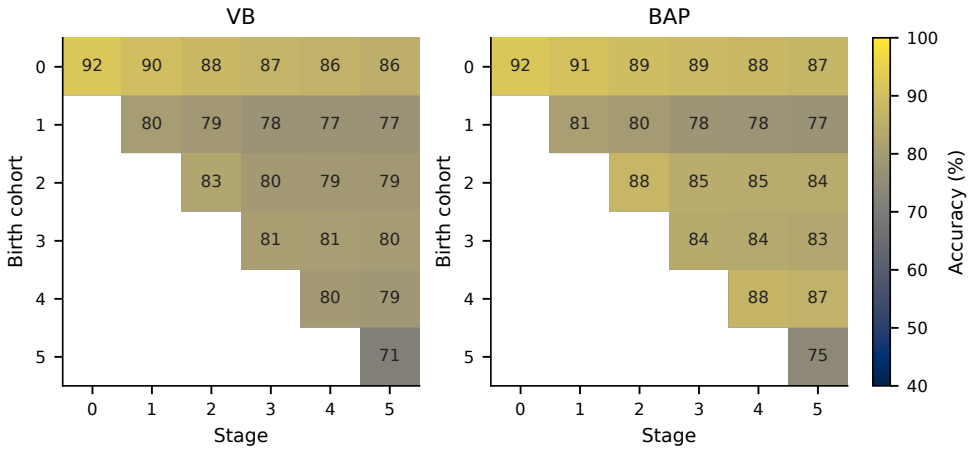


Figure S2: Birth-cohort accuracy under a fixed semantic mapping. Blank cells indicate classes that have not yet been introduced. The left and right panels show VB and BAP, respectively; the horizontal axis is stage, the vertical axis is birth cohort, and the colorbar shows accuracy (%).

D.3 Literature Comparisons

Table S5 summarizes reported S5 All values by source: VB Table 2, GOAL Table 1, and VC Table 1/Appendix D3. VB Table 2 also reports GM, MetaGCD, Happy, and PromptCCD. Published results follow the representation, mapping, and evaluation conventions of their cited sources. VC Appendix D2 reports stage averages, which are stored separately in the MeanAll table. The matched comparisons in the main text share the representation, data stream, and birth-stage mapping described above.

Table S5: Literature-reported final recognition accuracy grouped by evaluation setting.

Reported method	Source	CIFAR	Tiny	ImageNet	CUB
FRoST	GOAL Table 1	48.00	40.20	63.80	34.60
GM	GOAL Table 1	54.10	46.90	71.10	52.00
GOAL	GOAL Table 1	61.50	57.20	79.30	61.90
Happy	GOAL Table 1	60.00	54.60	78.60	59.40
MetaGCD	GOAL Table 1	55.80	48.10	66.70	49.60
SimGCD	GOAL Table 1	43.50	41.60	52.90	44.70
SimGCD+	GOAL Table 1	50.50	43.00	57.60	45.80
VanillaGCD	GOAL Table 1	51.40	45.90	64.50	46.80
GM	VB Table 2	54.11	46.90	71.08	51.96
Happy	VB Table 2	60.77	54.93	78.58	54.37
MetaGCD	VB Table 2	55.78	48.14	66.68	49.60
PromptCCD	VB Table 2	55.53	50.74	63.97	50.95
VB-CGCD	VB Table 2	81.23	74.52	87.32	66.44
Happy	VC Appendix D3	–	–	–	58.53
MetaGCD	VC Appendix D3	–	–	–	49.60
VC-CGCD	VC Appendix D3	–	–	–	58.75
FRoST	VC Table 1	48.03	40.15	63.82	–
GM	VC Table 1	54.11	46.90	71.08	–
Happy	VC Table 1	59.99	51.99	74.98	–
KMeans	VC Table 1	35.69	34.62	53.92	–
MetaGCD	VC Table 1	55.78	48.14	66.68	–
SimGCD	VC Table 1	43.53	41.59	52.90	–
VC-CGCD	VC Table 1	61.28	53.25	76.72	–
VanillaGCD	VC Table 1	51.36	45.94	64.54	–

E SUPPLEMENTARY ABLATIONS AND INTERVENTION DEFINITIONS

All controls modify the BAP method defined in Section 3 and Appendices A–B. "Full" denotes BAP with all its components. Table 4(a,c) reports the matched C100 count-model and label controls, whose Full reference obtains a Final All of 84.25%. Removing the per-class bounds replaces the new-class intervals by $[1, N]$ while retaining the background bounds that encode the total new-class constraint. Removing the aggregate bound replaces the background interval by $[0, N]$ while retaining the individual new-class intervals. Both controls preserve ν and π. The plug-in Poisson control fixes the shared rate at its posterior mean, replacing the predictive count distributions by $\mathrm{Poisson}(\mu_\lambda)$ and $\mathrm{Poisson}(K\mu_\lambda)$ before constructing the intervals; all remaining terms are unchanged. The final-row argmax control retains candidate optimization and selection by the complete hard objective, and changes only the labels used for the final Gaussian fit.

In Table 4(b), BASE_ONLY and CAP_ONESHOT use the same marginal conditions and neither runs spectral candidates. They differ only in whether class statistics and assignments are updated alternately. The remaining structural and support controls are listed in Table E1.

Table E1: Final recognition accuracy under structural and support controls.

Setting	Intervention	C100	CUB
BAP (Full)	Complete inference with base and spectral candidates	84.32	66.98
BASE_ONLY	Base candidates only, with the original single-candidate budget	84.32	67.48
CAP_ONESHOT	One assignment under the initial class statistics	82.31	65.38
RANDOM_SPLIT	Replace the spectral split direction with a random direction	84.32	65.76
NO_UPPER	Relax hard count bounds to nonempty classes and mass conservation; retain marginal references	81.88	63.93
FIXED_SOURCE_NO_UPPER	Use the same relaxed bounds and fix the initial background/new-class support	83.38	63.89

Table E2: Results and completion status of additional CUB controls.

Setting	Final All (%)	Status
COUNT_PLUGIN_POISSON	Not completed	S1 soft solve stopped
NO_INDIVIDUAL_BOUNDS	Not completed	Saved only through S3
NO_AGGREGATE_BOUND	65.15	Completed
FINAL_ROW_ARGMAX	Not completed	Stopped before entering the label intervention

The BAP references in Table E1 and Table 4(a,c) belong to the structural/support and count-model/label experimental blocks, respectively. Their reported scores are retained within their own comparisons; differences across these blocks are not used to estimate component effects. The interventions are defined below.

Table E2 reports absolute endpoints and completion states for the additional CUB controls. A matched BAP reference is not reported for this block, so no paired gains or component rankings are inferred from it. An unfinished run has no final endpoint; missing results are not replaced by zero or by scores from another comparison.

E.1 S0 POSTERIOR-PREDICTIVE CONSTRAINTS

The S0 data for CIFAR contain 50 classes, each with 400 distinct labeled observations. Under the shared-rate model, this gives $a_0 = 20000.5$ and $b_0 = 50$, with a single-class prediction interval of $[361, 440]$ and a ten-class prediction interval of $[3865, 4137]$. Table S6 lists the corresponding background domain and the inferred hard-assignment totals. At every stage, the total number of new-class assignments reaches the aggregate lower bound of 3865, showing that the predictive domain is an active constraint in the inference.

Table S6: Bayesian marginal bounds and inferred hard-assignment totals on CIFAR.

Stage	N	New-class interval	Background interval	ν_0	Hard new-class total	Hard background total
1	5250	$[3865, 4137]$	$[1113, 1385]$	1249.9	3865	1385
2	5500	$[3865, 4137]$	$[1363, 1635]$	1499.9	3865	1635
3	5750	$[3865, 4137]$	$[1613, 1885]$	1749.9	3865	1885
4	6000	$[3865, 4137]$	$[1863, 2135]$	1999.9	3865	2135
5	6250	$[3865, 4137]$	$[2113, 2385]$	2249.9	3865	2385

E.2 Projection and support-set diagnostics

In Figure 4(b), A/B/C denote slots 10/6/4, matching C10/C6/C4 in panel (a). The original 11-slot weights are retained; Others aggregates omitted slots, and preferences use full-row maxima.

The mechanism measurement for S1 uses the first base-candidate subproblem. Its local background and new-class masses are 1647.34 and 3602.66, respectively, which become 1385 and 3865 after projection. The normalized stationarity residual is 4.31×10^{-13}. In this solve, columns whose pre-projection local masses exceed their upper bounds receive positive potentials, whereas columns whose local masses fall below their lower bounds receive negative potentials. The relative-odds relation describes the resulting change.

The total-variation statistic $\frac{1}{2}\sum_c |T_{ic} - R_{ic}| \geq 0.05$ identifies 1040 observations whose assignments are substantially revised, of which 750 change their preferred class. Figure 1 gives the algorithmic pipeline, Figure 2(a) shows the measured first projection, and the final-stage diagnostics use the complete selected assignment.

For the true new-class set V, precision is $|A' \cap V|/|A'|$, and recall is $|A' \cap V|/|V|$. Joint correct coverage is the number of accepted new-class observations whose class is also identified correctly, divided by $|V|$. Repairs and harms are evaluated within $V \cap A \cap A'$, whose coverage is $|V \cap A \cap A'|/|V|$. Table S7 explicitly reports the metrics corresponding to these denominators and the associated changes in class attribution. The initial and final joint correct coverage values are computed directly from the corresponding assignments under the same evaluation mapping.

Table S7: Support-set and assignment-refinement diagnostics on CIFAR.

Stage	Precision	Recall	Joint correct coverage	Repairs	Harms	Coverage
1	98.24	94.92	85.55	330	107	93.50
2	97.80	94.50	87.80	392	81	94.08
3	96.40	93.15	89.33	321	22	92.35
4	97.80	94.50	91.75	411	13	93.95
5	94.33	91.15	89.90	12	4	89.83

E.3 Controlled cardinality and inference configurations

BAP uses a single cardinality construction. Deduplicated labeled S0 counts determine the shared-rate Gamma posterior. Individual and aggregate posterior-predictive intervals determine $L_1, U_1, L_\Sigma, U_\Sigma$, and class presence and batch-mass conservation give the effective bounds in Eqs. (A.19)–(A.22). The reference masses are $\nu_k = \mu_\lambda$ for new classes and $\nu_0 = N - K\mu_\lambda$, with $\pi = \nu/N$, subject to the positivity and feasibility conditions in Appendix A.5. Dataset sampling caps do not enter this construction.

The mean-only control replaces the predictive-interval constraints and background-mass regularizer with a penalty on new-class counts, $\sum_{k\geq 1}[q_k \log(q_k/\bar{n}_0) - q_k + \bar{n}_0]$, where $\bar{n}_0 = K_0^{-1}\sum_c n_c^{(0)}$ is the empirical mean S0 count. It retains the auxiliary geometry, background matching costs, row entropy, and the nonempty-class lower bound of 1, but omits the row-prior weights π. This is a count-model ablation of BAP. The other count and label controls change only the components specified at the start of this appendix.

The iteration and candidate-search controls retain the BAP posterior-derived bounds and reference masses. BASE_ONLY uses only the base initialization with the same per-candidate budget.

CAP_ONESHOT solves one bounded assignment under the initial class statistics and does not alternate assignment and posterior updates. Neither uses spectral candidates. RANDOM_SPLIT replaces the spectral direction with a deterministic random direction while retaining the split-magnitude rule, ten Lloyd steps, class count, and candidate budget. Its random seed is determined by the dataset, stage, and initial class index.

NO_UPPER sets all upper bounds to N and replaces the raw lower bounds by 0 for background and 1 for new classes, then applies total-mass conservation. It therefore relaxes the hard predictive intervals, including their lower endpoints, while retaining the BAP reference masses ν and prior weights π. FIXED_SOURCE_NO_UPPER uses the same relaxed bounds but constrains initially accepted observations to new-class columns and initially rejected observations to background. This pair isolates support-set reversibility under a shared feasible domain. Table E1 reports all controls, including the higher BASE_ONLY result on CUB.

F CONTINUAL INTERVENTIONS, EXTENSIONS, AND SENSITIVITY

Table F1: Later-stage recognition performance with BAP applied only at S1.

Dataset	Update strategy	Final All (%)	S1-class accuracy at S5 (%)
C100	VB-CGCD throughout	81.49	77.10
C100	S1 BAP, then VB	81.83	75.70
CUB	VB-CGCD throughout	64.76	43.84
CUB	S1 BAP, then VB	64.81	43.31

Revising only the first stage produces limited final overall gains and does not improve the endpoint accuracy of the fixed S1 class group. Together with the continual-use comparison in the main text, this result shows that the overall gains cannot be attributed to a single early correction.

Table F2: Parameter sensitivity of final recognition accuracy on C100.

Experiment group	Parameter value	Final All
Nominal prediction-interval level	0.90	84.24
Nominal prediction-interval level	0.95	84.25
Nominal prediction-interval level	0.99	84.12
Entropy coefficient ϵ	0.5	84.36
Entropy coefficient ϵ	1.0	84.32
Entropy coefficient ϵ	2.0	84.33

Prediction-level and entropy-coefficient sensitivity are reported as separate experimental blocks, each with its own BAP reference. The former changes the predictive quantiles while preserving the S0 posterior and reference masses; the latter varies ϵ while holding the count bounds and reference masses fixed. The observed ranges are 0.13 and 0.04 percentage points, respectively. These results are not a joint parameter sweep, and the reference scores are not compared across blocks.

Upper-bound relaxation and predictive-interval level are distinct interventions. The CUB sensitivity block does not report a matched 95% endpoint; the available 90% and 99% endpoints are reported without substituting a BAP result from another experimental block.

The extension studies use the same S0-calibrated BAP construction as the main experiments, with the named protocol or component interventions. T10 uses the official ten-stage data loader, smaller class increments, independent GMM initialization, and continual memory within each branch. The posterior prediction uses the number K_t of new classes in each stage and the observed batch size N_t. The class-order experiment fixes the base classes and applies one PCG64(2026) permutation to the new-class list. Table S8 reports Final All and MeanAll under these settings.

For the underfilled stream, each observed class keeps, in stable UID order, a class-determined fraction selected from $\{0.4, 0.6, 0.8, 1.0\}$. This rule constructs the benchmark observations; its retention fractions and per-class sampling caps are not supplied to BAP as count priors or constraints. BAP uses the S0 count posterior, the resulting batch size, and the known number of new classes. CIFAR

Table F3: Final recognition accuracy under additional experimental settings.

Experiment	Dataset	VB-CGCD	BAP or corresponding variant
Class order 2026	C100	78.59	82.96
Class order 2026	CUB	65.90	Not completed
Underfilled mixed stream	C100	80.18	82.09
Underfilled mixed stream	CUB	65.03	62.01
Upper-bound relaxation $\times 1.10$	C100	Not applicable	84.23
Upper-bound relaxation $\times 1.25$	C100	Not applicable	82.88
Upper-bound relaxation $\times 1.10$	CUB	Not applicable	66.22
Upper-bound relaxation $\times 1.25$	CUB	Not applicable	63.95
$\epsilon = 0.5$	CUB	Not applicable	Not completed
$\epsilon = 2.0$	CUB	Not applicable	67.21
Prediction level 90%	CUB	Not applicable	66.02
Prediction level 95%	CUB	Not applicable	Not reported
Prediction level 99%	CUB	Not applicable	64.93

improves from 80.18% to 82.09%, whereas CUB decreases from 65.03% to 62.01%. These opposite outcomes identify acquisition-scale variation as a limitation of transferable cardinality modeling.

The upper-bound relaxation controls use multipliers of 1.10 and 1.25 with exact upward rounding of the column upper bounds supplied to the assignment solver. Bounds implied by total-mass conservation are recomputed, while ν and π remain unchanged. These controls perturb the assignment domain; they do not recalibrate the count posterior from the data-generation rules or change the nominal predictive-interval level.

Table S8: Final and mean recognition accuracy for BAP extensions and sensitivity controls.

Study	Dataset	Method/variant	Final All	MeanAll	Status
Underfilled	cifar	VB-CGCD	80.18	83.86	Completed
Underfilled	cifar	BAP	82.09	85.55	Completed
Underfilled	cub	VB-CGCD	65.03	70.75	Completed
Underfilled	cub	BAP	62.01	69.23	Completed
T10	cifar	VB-CGCD	77.70	82.62	Completed
T10	cifar	BAP	82.23	85.28	Completed
T10	tiny	VB-CGCD	72.94	79.59	Completed
T10	tiny	BAP	75.88	82.15	Completed
Order 2026	cifar	VB-CGCD	78.59	83.63	Completed
Order 2026	cifar	BAP	82.96	85.87	Completed
Order 2026	cub	VB-CGCD	65.90	72.64	Completed
Order 2026	cub	BAP	–	–	Not completed
Entropy coefficient	cifar	BAP ($\epsilon = 0.5$)	84.36	86.74	Completed
Entropy coefficient	cifar	BAP ($\epsilon = 2.0$)	84.33	86.72	Completed
Entropy coefficient	cub	BAP ($\epsilon = 0.5$)	–	–	Not completed
Entropy coefficient	cub	BAP ($\epsilon = 2.0$)	67.21	73.03	Completed
Upper-bound relaxation	cifar	BAP ($\times 1.10$)	84.23	86.61	Completed
Upper-bound relaxation	cifar	BAP ($\times 1.25$)	82.88	85.12	Completed
Upper-bound relaxation	cub	BAP ($\times 1.10$)	66.22	72.50	Completed
Upper-bound relaxation	cub	BAP ($\times 1.25$)	63.95	71.04	Completed

The CUB $\epsilon = 0.5$ run terminates at S2 with a transport residual of approximately 4.93×10^{-8}, above the fixed threshold of 2×10^{-8}. The CUB class-order experiment terminates at the MM non-increase check in S3. The completed stages saved for both runs remain available, but the final endpoints are marked as missing.

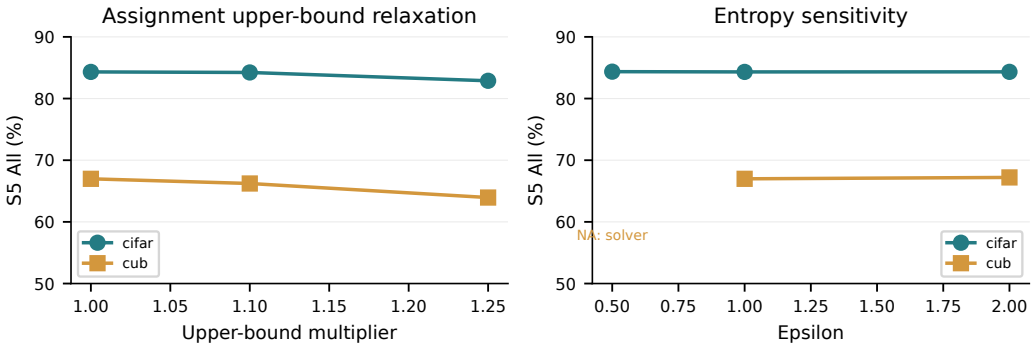


Figure S3: Sensitivity to assignment upper-bound relaxation (left) and the entropy coefficient (right). Points show final S5 All accuracy for complete runs. The CUB $\epsilon = 0.5$ endpoint is missing because the solver terminated, as marked by "NA: solver". The upper-bound multiplier changes solver constraints, whereas ϵ changes the row-entropy weight.

Table G1: Mean gains from alternative classification rules.

Dataset	Full Gaussian	Diagonal Gaussian	Euclidean
C100	+3.00	-1.41	+3.09
Tiny	+2.33	-1.17	+1.63
IN100	+3.99	+0.91	+4.52
CUB	+1.38	-2.84	+1.93

G Classification rules and computational record

Paired differences for each scoring rule are computed using the same class means, test samples, and evaluation mapping. The diagonal Gaussian and Euclidean variants are scoring replacements on the trained memory and do not retrain another class model.

Table G2: Computational costs of matched C100 runs.

Method	Fit (s)	Auxiliary (s)	Test (s)	Extra statistics (KiB)
VB-CGCD	327.6	0.0	0.32	0.0
BAP	242.2	154.8	1.43	300.8

Fit, auxiliary, and test times are reported in seconds, and extra statistics are reported in KiB. Fit and auxiliary times are not cross-spliced across groups, and the original measurement conditions are preserved.

G.1 Per-stage cost and memory

Table S9 breaks down computational costs by dataset, method, and processing stage. BAP denotes the same S0-calibrated method used throughout this paper; Spectral remains a separate control. Native fitting, auxiliary inference, and query measurements are reported separately. GPU load differs across runs, so costs from different experimental blocks are not pooled. Persistent native arrays, updated auxiliary statistics, fixed geometric transformations, and temporary optimizer memory are recorded separately. The CIFAR implementation uses ten native fits, 1109 fixed-cost transport solves, and 55 integer-linear-program hardening steps. The final-stage n, h statistics occupy 300.8 KiB. The fixed L, m_0, κ geometry is reused throughout the sequence. Candidate comparison takes place in the auxiliary space, followed by one final native fit per stage.

G.2 NUMERICAL AND REPRODUCIBILITY MATERIALS

Mathematical definitions and proofs are given in Appendices A–B; numerical budgets, stopping criteria, and validity checks are given in Appendix C. Each experiment records random seeds, data streams, configurations, candidate objective values, inference quality, hard assignments, and the corresponding historical model states. Tables and quantitative figures are generated from source CSV files. The independent evaluator uses semantic labels for scoring and support-set analysis; these diagnostics do not enter the assignment optimization.

The manuscript package includes editable LaTeX, figure scripts, numerical source tables, and the official ICLR style file. Reproducing training additionally requires the benchmark feature resources and Gaussian-model dependencies. The supplied configuration, prior, and data manifests identify the inputs associated with the reported measurements.

Table S9: Runtime and persistent memory usage across datasets and methods.

Dataset	Method	First fit (s)	Auxiliary (s)	Final fit (s)	Query (s)	Native (MiB)	Auxiliary (KiB)
cifar	BAP	242.5	342.3	199.1	3.09	56.40	300.8
cifar	Spectral	140.6	8.5	119.7	–	56.40	0.0
cifar	VB	178.7	0.0	148.9	–	56.40	0.0
cub	BAP	161.6	469.9	125.7	3.56	112.79	601.6
cub	Spectral	154.1	6.6	131.7	–	112.79	0.0
cub	VB	157.6	0.0	136.3	–	112.79	0.0
imagenet	BAP	243.7	592.9	199.5	2.66	56.40	300.8
imagenet	Spectral	149.5	27.2	121.8	1.87	56.40	0.0
imagenet	VB	139.7	0.0	121.7	1.70	56.40	0.0
tiny	BAP	169.6	4671.9	129.4	3.44	112.79	601.6
tiny	Spectral	298.2	75.1	246.6	6.75	112.79	0.0
tiny	VB	298.4	0.0	267.7	8.02	112.79	0.0

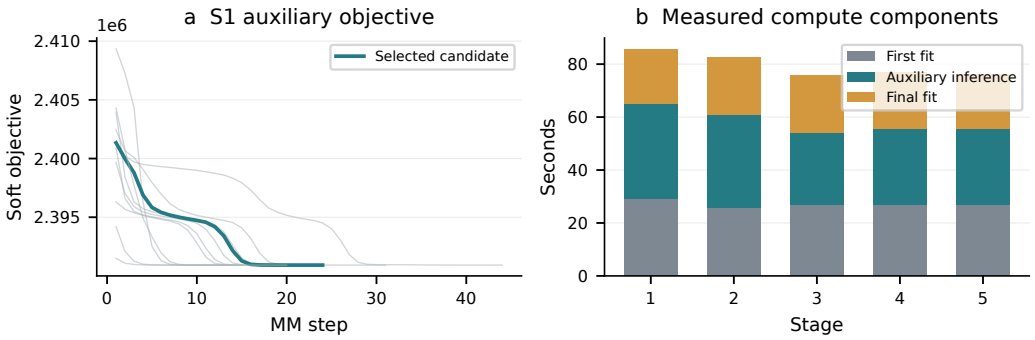


Figure S4: BAP optimization and computation. (a) Saved auxiliary-objective traces for S1 candidates; the highlighted candidate is selected by the complete hard objective, not by the plotted soft objective alone. (b) Measured first-fit, auxiliary-inference, and final-fit times across the five stages. These components exclude input/output and query scoring and are not total wall time.