SEE-K: Seen-Constrained Model-Order Selection for Unknown-K Generalized Category Discovery

Mingfu Yan[1,2], **Jiancheng Huang**[2], **Haipeng Luo**[3], **Yi Huang**[2], **Yuqi Peng**[4], **Qiang Zhou**[1,5,6*], **Shifeng Chen**[2,7*]
[1]Southeast University
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[3]Shenzhen International Graduate School, Tsinghua University
[4]University of Arizona
[5]Purple Mountain Laboratories
[6]Engineering Research Center of Blockchain Application, Supervision and Management (Southeast University), Ministry of Education
[7]Shenzhen University of Advanced Technology
mfyan99@gmail.com, zhouqiang@u.nus.edu, shifeng.chen@siat.ac.cn

Abstract

Generalized category discovery (GCD) becomes substantially more challenging when the number of novel classes is unknown. We argue that, after the backbone is fixed, this stage should be treated as seen-constrained model-order selection rather than as an auxiliary clustering detail. The selector must estimate the class count without novel labels while remaining consistent with the labeled seen categories available in the standard GCD setting. We introduce SEE-K, Seen-category Evidence for Estimating K, which instantiates this view by constructing a nested merge path from a seen-constrained overclustering and selecting the coarsest near-optimal cut under observable surrogate decision risks. These risks combine seen-class consistency, cross-seed stability, unlabeled-side compactness, and a tiny-cluster degeneracy penalty. The resulting protocol separates class-count error, downstream clustering quality, and robustness. Where an aligned public unknown-K baseline is available, the selector reduces class-count error on all four fine-grained datasets, including a reduction from 342 to 15 classes on Herbarium-19. The selector also applies to additional GCD feature sources without retuning. Broader-domain diagnostics further distinguish ImageNet100, where the default selector can underestimate the class count toward the seen-class regime, from CIFAR100, where the selector landscape exposes an accuracy-count operating-point trade-off. Under long-tailed sampling of the unlabeled discovery pool, All accuracy rises while rare-class recovery declines. These results show that count fidelity and assignment quality can diverge, supporting unknown-K selection as a decision layer that should be evaluated separately from representation learning.

1 Introduction

Generalized category discovery (GCD) studies a partially labeled open-world regime: labeled samples come from known classes, while unlabeled samples may belong to either known or novel classes [1]. This setting couples two problems that are often evaluated together but need not be solved by the same mechanism: representation learning transfers semantic structure from labeled seen classes, while model-order selection determines the final category count on a fixed feature bank.

*Corresponding authors: Qiang Zhou, Shifeng Chen.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

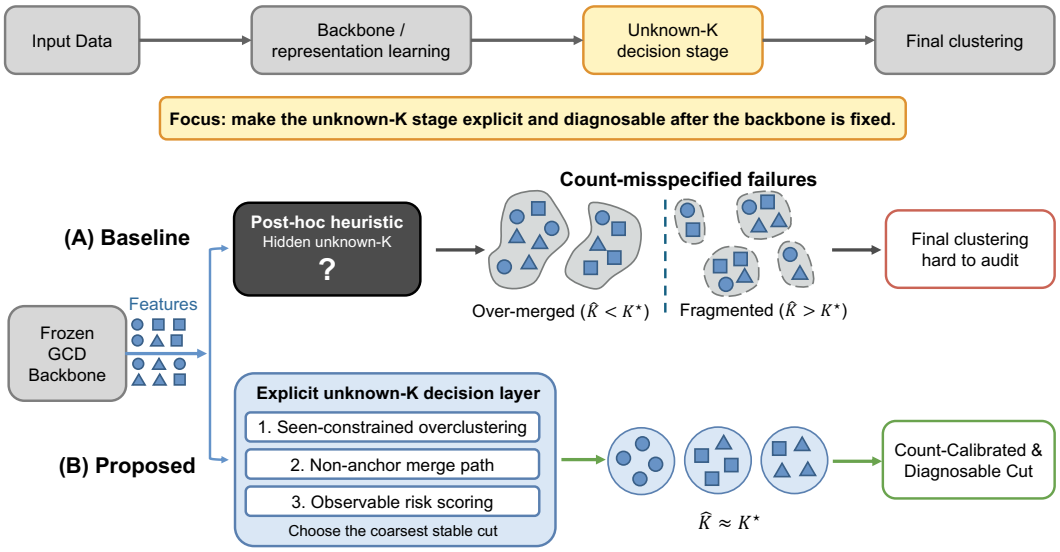

Figure 1: **Motivation: From implicit post-processing to explicit unknown-K model-order selection**. (A) **Baseline**: GCD systems often hide the class-count decision in method-specific post-processing, which can conflate representation quality with count error and lead to over-merging ($\widehat{K} < K^\star$) or over-fragmentation ($\widehat{K} > K^\star$). (B) **Proposed**: We formulate unknown-K as seen-constrained model-order selection: seen labels provide anchors, the merge path defines candidate orders, and the selector chooses the coarsest near-optimal cut under observable risks. Here $K^\star$ is the ground-truth total count used for evaluation and $\widehat{K}$ is the selected count.

Most recent GCD methods focus on the first problem, including strong parametric baselines such as SimGCD and specialized fine-grained systems such as SelEx [2, 3]. The second problem is usually left to method-specific post-processing. This is risky in fine-grained discovery: underestimating K over-merges nearby categories, whereas overestimating K fragments one semantic class into small clusters. Our paper therefore isolates the unknown-K stage and studies it as a seen-constrained model-order decision after feature extraction.

The key observation is that unknown-K selection is neither fully supervised nor fully unsupervised. Novel labels are unavailable at selection time, but labeled seen examples, or their class prototypes in the feature space, are available under the standard GCD setting. We call these *seen-category anchors*. They provide observable constraints on the selector without adding supervision. The target is therefore not an arbitrary internally high-scoring partition, but the *coarsest stable seen-constrained cut*: a small class count that preserves seen-category structure, remains stable across perturbations, fits the feature geometry, and avoids degenerate tiny fragments. Figure 1 illustrates this shift from implicit post-processing to an explicit model-order decision.

We instantiate this view with a fixed-feature selector that builds a seen-constrained candidate family, converts it into a nested non-anchor merge path, scores each cut with observable surrogate decision risks, and chooses the smallest near-optimal count. This design changes the evaluation target: class-count error, final clustering accuracy, same-path $K^\star$ behavior, and seed robustness are reported separately. These diagnostics show how cut selection affects partition quality on fixed merge paths and why count fidelity and clustering accuracy can diverge.

Contributions. Our contributions are fourfold. First, we formulate unknown-K GCD after feature extraction as *seen-constrained model-order selection*, where seen-category anchors make the decision observable but novel labels remain unavailable. Second, we introduce SEE-K (Seen-category Evidence for Estimating K), which instantiates the formulation with seen-constrained overclustering, a non-anchor merge path, and a surrogate decision-risk score combining seen consistency, cross-seed stability, unlabeled-side compactness, and degeneracy control. Third, we provide a fixed-feature evaluation protocol that separates count calibration, clustering quality, and robustness. Fourth, we report a matched four-dataset evaluation on CUB, SCars, Aircraft, and Herbarium-19 using SimGCD

and SelEx checkpoints; in the aligned SelEx unknown-K comparison, the selector reduces class-count error on all four datasets, including Herbarium-19 where the error drops from 342 to 15 classes. We further evaluate transfer to additional frozen feature sources without retuning and show that long-tailed discovery-pool sampling can increase All accuracy while reducing rare-class recovery.

Code is available at `https://github.com/MingfuYAN/SEE-K`.

2 Related Work

Recent GCD work has improved open-world clustering through stronger representation learning and category-transfer objectives. Before GCD, deep transfer clustering studied novel category discovery without the mixed labeled/unlabeled structure used in modern GCD [4]. The GCD formulation [1] established the partially labeled setting, SimGCD showed that a parametric classifier is a strong baseline [2], and later methods introduced information maximization, semi-supervised mixture models, prompt-based affinities, prototypical or hierarchical formulations, and fine-grained self-expertise mechanisms [3, 5–12].

Recent work also broadens the setting to domain shift, federated heterogeneity, consistency objectives, hyperbolic domain generalization, and semantic-aware hierarchy [13–17]. Public baselines such as AFGCD, RLCD, DebGCD, and ReLKD likewise focus on stronger features or debiased distillation-style objectives rather than making the unknown-K decision a primary object of study [18–21]. Thus representation design, clustering dynamics, and class-count choice are often evaluated jointly.

Our work is orthogonal to backbone-level improvements. We fix the feature bank and ask whether the class-count decision itself is accurate, stable, and interpretable. The contribution is therefore a decision-layer study: by isolating unknown-K selection from representation learning, we can diagnose count errors, merge-path gaps, and robustness failures that a single aggregate clustering score would otherwise obscure.

3 Problem Setting

Notation and task. A fixed GCD backbone produces a normalized feature bank; for simplicity, we write x_i for the extracted feature rather than the raw image. The feature bank is split into a labeled seen set $\mathcal{L} = \{(x_i, y_i) : y_i \in \mathcal{Y}_{\text{seen}}\}$ and an unlabeled set $\mathcal{U} = \{x_j\}$ whose samples may come from either $\mathcal{Y}_{\text{seen}}$ or unknown novel classes $\mathcal{Y}_{\text{novel}}$. Let $N = |\mathcal{L}| + |\mathcal{U}|$ denote the total feature-bank size. We write $K_{\text{seen}} = |\mathcal{Y}_{\text{seen}}|$, $K_{\text{novel}} = |\mathcal{Y}_{\text{novel}}|$, and $K^\star = K_{\text{seen}} + K_{\text{novel}}$ for the ground-truth total class count used only for evaluation. The selector estimates $\widehat{K}$ and returns a partition $\mathcal{C}_{\widehat{K}}$ of the full feature bank. In the algorithm, K_{init} is the initial overclustering count, R is the number of random seeds, and $\mathcal{C}_K^{(r)}$ denotes the candidate cut at count K on seed-specific path r.

Seen-category anchors. A seen-category anchor is either a labeled seen example or its class prototype in feature space. These anchors are not additional supervision: they are exactly the labeled seen data available in the standard GCD setting. Their role is to make the fixed-feature model-order decision partially observable. A valid unknown-K cut should preserve labeled seen-class structure while deciding how many non-anchor clusters are needed for the unlabeled side. Because novel labels are unavailable at selection time, the selector cannot directly evaluate all-class or novel-class clustering accuracy when choosing $\widehat{K}$; it instead relies on observable surrogate risks. Ground-truth labels and $K^\star$ are used only to evaluate the final partition, never to select $\widehat{K}$.

Evaluation references. We report clustering accuracy on all classes and on novel classes using the standard Hungarian matching protocol between predicted clusters and ground-truth labels [1]. We also report the class-count error $|\widehat{K} - K^\star|$. A *diagnostic known-K reference* fixes the clustering stage to the ground-truth total count $K^\star$ in the original backbone-specific evaluation protocol; it is a reference point, not an upper bound on our merge path. A *baseline unknown-K estimate* is the class count produced by a public baseline when $K^\star$ is not provided. SelEx exposes such an estimate, enabling a matched count-calibration comparison. SimGCD provides a known-K public evaluation protocol but not an additional aligned unknown-K estimate under our fixed-feature protocol, so its main comparison uses the diagnostic known-K reference.

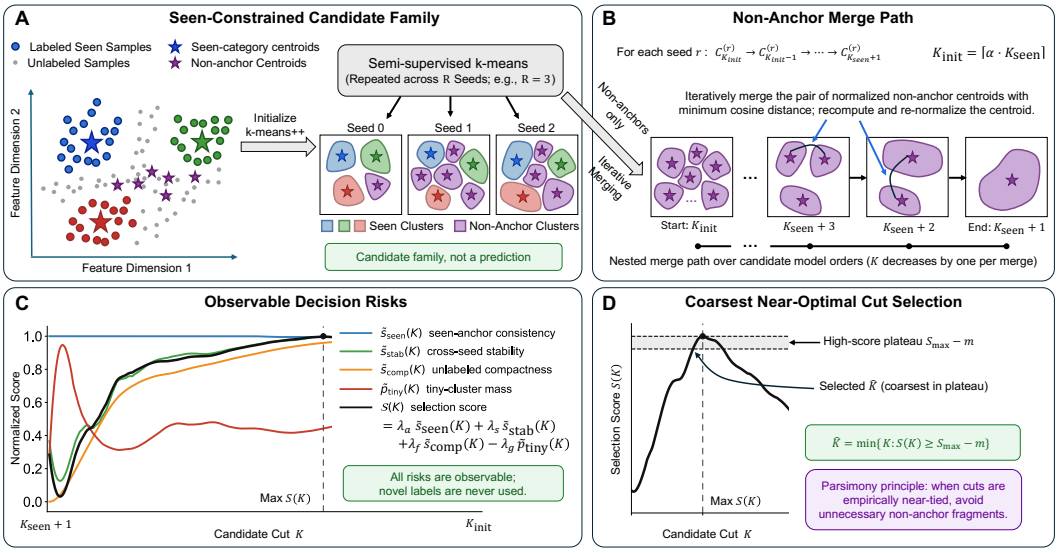

Figure 2: **Overview of SEE-K for seen-constrained model-order selection.** (A) **Seen-Constrained Candidate Family**: We initialize a conservative search space with one seen-category centroid per labeled class and non-anchor centroids initialized from unlabeled samples, repeating the semi-supervised k-means construction across seeds. (B) **Non-Anchor Merge Path**: For each seed, we construct a nested model-order path by iteratively merging the closest normalized non-anchor centroids under cosine distance and recomputing the merged centroid. (C) **Observable Decision Risks**: Candidate cuts are evaluated using normalized observable signals: seen-anchor consistency, cross-seed stability, unlabeled compactness, and tiny-cluster mass. Their combination yields the selection score $\mathcal{S}(K)$ without using novel labels. (D) **Coarsest Near-Optimal Cut Selection**: The final estimate $\widehat{K}$ is the smallest count in the high-score plateau, implementing the parsimony principle over empirically near-tied cuts.

4 Method

SEE-K operates after a GCD backbone has produced a fixed feature bank. It treats unknown-K inference as a model-order decision on this representation: it constructs a seen-constrained candidate family, turns this family into a nested sequence of candidate class counts, scores each cut using observable surrogate risks, and returns the coarsest near-optimal cut (Figure 2). The method is intentionally separated from representation learning so that class-count decisions can be audited without retraining the backbone.

4.1 Unknown-K as Seen-Constrained Model-Order Selection

We target the *coarsest stable seen-constrained cut*, expressed conceptually as

$$\min_{K} \; K \quad \text{s.t.} \quad R_{\text{anchor}}(K) \leq \epsilon_a,\; R_{\text{instab}}(K) \leq \epsilon_s,\; R_{\text{fit}}(K) \leq \epsilon_f,\; R_{\text{deg}}(K) \leq \epsilon_g.$$

Here R_{anchor} measures violation of seen-anchor consistency, R_{instab} measures cross-seed instability, R_{fit} measures geometric lack of fit, and R_{deg} measures fragmentation into tiny clusters. Since semantic feasibility thresholds are unavailable at selection time, we do not solve this constrained program directly. The selector instead combines these violations into a weighted surrogate-risk score and applies the plateau rule to select the coarsest near-optimal cut.

4.2 Seen-Constrained Candidate Family

The first step, shown in Figure 2A, is to construct a hypothesis class of candidate partitions rather than to predict K immediately. We start from a conservative overclustering so that unlabeled-side structure is unlikely to be prematurely collapsed. Seen-category anchors initialize the seen-class part

of the partition, while the remaining non-anchor centroids provide mergeable degrees of freedom for the unlabeled-side assignments. All features are ℓ_2-normalized and the initial cluster count is

$$K_{\text{init}} = \lceil \alpha\, K_{\text{seen}} \rceil, \qquad \alpha > 1,$$

where α controls the overclustering budget. This budget defines a coverage assumption: if K_{init} is below the semantic class count, the constructed path cannot recover missing model orders. We stress-test this assumption through K_{init} ablations.

We initialize one centroid per seen class using the mean labeled feature for that class; let $\mathcal{L}_c$ denote the labeled features of seen class c. The remaining $K_{\text{init}} - K_{\text{seen}}$ non-anchor centroids are initialized on the unlabeled set with k-means++ [22]. These non-anchor centroids are not assumed to be novel classes; they are unlabeled-side search variables and may later contain samples from either seen or novel categories. We then solve semi-supervised k-means:

$$\begin{aligned} \min_{\{\mu_k\}_{k=1}^{K_{\text{init}}},\, \{z_j\}} \quad & \sum_{c=1}^{K_{\text{seen}}} \sum_{x_i \in \mathcal{L}_c} \|x_i - \mu_c\|_2^2 + \sum_{x_j \in \mathcal{U}} \|x_j - \mu_{z_j}\|_2^2 \\ \text{s.t.} \quad & z_j \in \{1, \ldots, K_{\text{init}}\},\ \forall x_j \in \mathcal{U}, \quad \|\mu_k\|_2 = 1,\ \forall k = 1, \ldots, K_{\text{init}}. \end{aligned}$$

Labeled assignments remain constrained, but seen centroids are re-estimated with all assigned members, allowing unlabeled data to shape the feature-space partition while preserving seen identities. Centroids are re-normalized after each update, so nearest-centroid assignment under squared Euclidean distance is equivalent to maximal cosine similarity. We repeat this construction across R random seeds to expose perturbation sensitivity.

4.3 Non-Anchor Merge Path

The candidate family becomes the path in Figure 2B by converting each overclustered solution into a nested sequence of candidate class counts. We merge only *non-anchor* centroids, i.e., centroids not initialized from labeled seen categories. Seen-anchor centroids are kept unmerged because their identities are defined by labeled seen labels. This restriction prevents the model-order search from collapsing two seen categories into one cluster, while still allowing the unlabeled-side portion of the partition to be coarsened.

For seed r, let $\mathcal{C}_K^{(r)}$ denote the cut with K clusters. At each step, we merge the pair of normalized non-anchor centroids with minimum cosine distance, recompute the merged centroid from its member features, re-normalize, and record the new cut. Equivalently, because all centroids are normalized, this chooses the pair with minimum squared Euclidean distance on the unit sphere. Repeating this operation yields

$$\mathcal{C}_{K_{\text{init}}}^{(r)} \to \mathcal{C}_{K_{\text{init}}-1}^{(r)} \to \cdots \to \mathcal{C}_{K_{\text{seen}}+1}^{(r)},$$

while preserving the same seen-anchor identities across cuts. Thus unknown-K inference becomes selection along an explicit seen-constrained model-order path rather than an unconstrained search over unrelated partitions.

4.4 Observable Decision Risks

Along this path, the observable signals summarized in Figure 2C provide the evidence used for selection. Novel labels are unavailable at selection time, so each candidate cut is scored by surrogate risks corresponding to the failure modes in the constrained view: seen-structure violation, seed-specific instability, poor geometric fit, and degenerate fragmentation.

Seen consistency. $s_{\text{seen}}(K)$ releases the training constraint and reassigns each labeled seen sample to its nearest centroid; the reassignment is counted as correct only if that centroid corresponds to the sample's seen class. This is not the constrained training assignment, and therefore is not trivially equal to one. Instead, it diagnoses whether the unlabeled-side search has distorted the local geometry around labeled seen anchors.

Cross-seed stability. $s_{\text{stab}}(K)$ measures whether the same model order yields similar unlabeled assignments across perturbations of the overclustering stage. We compute the average pairwise normalized mutual information (NMI) between seed-specific cuts at the same K, $\frac{2}{R(R-1)} \sum_{r<r'} \text{NMI}(\mathcal{C}_K^{(r)}, \mathcal{C}_K^{(r')})$, which measures shared information between clusterings and ranges

from 0 for independent partitions to 1 for identical partitions. A high value indicates that the cut is not an artifact of a single initialization.

Unlabeled-side compactness and degeneracy. $s_{\mathrm{comp}}(K)$ is the negative mean squared distance of unlabeled samples to their assigned centroids, capturing geometric fit in the fixed feature space. $p_{\mathrm{tiny}}(K)$ is the fraction of non-anchor clusters with size below $s_{\min} = \max(5, \lceil 0.001N \rceil)$, discouraging cuts that explain noise through many small fragments.

After normalization over the merge path, these signals define risks

$$R_{\mathrm{anchor}}(K) = 1 - \tilde{s}_{\mathrm{seen}}(K), \quad R_{\mathrm{instab}}(K) = 1 - \tilde{s}_{\mathrm{stab}}(K),$$
$$R_{\mathrm{fit}}(K) = 1 - \tilde{s}_{\mathrm{comp}}(K), \quad R_{\mathrm{deg}}(K) = \tilde{p}_{\mathrm{tiny}}(K).$$

They are combined into the surrogate decision risk

$$\widehat{\mathcal{R}}(K) = \lambda_a R_{\mathrm{anchor}}(K) + \lambda_s R_{\mathrm{instab}}(K) + \lambda_f R_{\mathrm{fit}}(K) + \lambda_g R_{\mathrm{deg}}(K), \qquad \lambda_a, \lambda_s, \lambda_f, \lambda_g \geq 0.$$

Equivalently, we maximize

$$\mathcal{S}(K) = \lambda_a \tilde{s}_{\mathrm{seen}}(K) + \lambda_s \tilde{s}_{\mathrm{stab}}(K) + \lambda_f \tilde{s}_{\mathrm{comp}}(K) - \lambda_g \tilde{p}_{\mathrm{tiny}}(K),$$

which differs from minimizing $\widehat{\mathcal{R}}(K)$ by an additive constant. Compactness, stability, and tiny-cluster terms use min–max normalization:

$$\tilde{q}(K) = \frac{q(K) - \min_{k \in \mathcal{K}} q(k)}{\max_{k \in \mathcal{K}} q(k) - \min_{k \in \mathcal{K}} q(k)}, \qquad q \in \{s_{\mathrm{comp}}, s_{\mathrm{stab}}, p_{\mathrm{tiny}}\},$$

where $\mathcal{K}$ denotes the candidate class counts along the merge path. Seen consistency is often close to one for many cuts, so min–max scaling can over-amplify negligible differences. We instead use a close-to-best saturation rule,

$$\tilde{s}_{\mathrm{seen}}(K) = \max\left(0,\, 1 - \frac{\max(0,\, s_{\mathrm{seen}}^{\max} - s_{\mathrm{seen}}(K))}{\tau}\right), \qquad s_{\mathrm{seen}}^{\max} = \max_{K'} s_{\mathrm{seen}}(K').$$

The normalized seen term is therefore a tolerance around the best available seen consistency, not a new source of supervision. Default weights and thresholds are reported in the supplementary material.

4.5 Coarsest Near-Optimal Cut Selection

The score landscape in Figure 2D motivates the final selection rule: the direct maximizer of $\mathcal{S}(K)$ is not always the most appropriate decision, because several nearby granularities often have nearly indistinguishable observable risks. Larger K can improve compactness by fragmenting the unlabeled side, whereas smaller K can suppress noise but may under-separate semantic categories. We therefore apply a parsimony rule: among cuts whose scores are close to the best score along the path, choose the coarsest one.

Let $\mathcal{S}_{\max}$ be the best score along the path and define the high-score plateau

$$\{K : \mathcal{S}(K) \geq \mathcal{S}_{\max} - m\},$$

with margin m. The selected count is

$$\widehat{K} = \min\{K : \mathcal{S}(K) \geq \mathcal{S}_{\max} - m\}.$$

This rule implements the constrained objective's preference for the smallest feasible model order and avoids explaining initialization noise through unnecessary non-anchor fragments.

For seed-dependent diagnostics, seen consistency, compactness, and tiny-cluster mass are computed on each seed-specific cut and averaged; stability is computed by pairwise NMI across seeds. After selecting $\widehat{K}$, we return the seed-specific partition with highest seen consistency at $\widehat{K}$, breaking ties by higher compactness and then lower tiny-cluster mass. This deterministic rule uses only observable diagnostics, not novel labels, $K^\star$, or clustering accuracy; the same rule is used in same-path $K^\star$ diagnostics when the count is fixed for analysis.

Profile specification and calibration. The fixed default uses $(\lambda_a, \lambda_s, \lambda_f, \lambda_g) = (0.4, 0.3, 0.2, 0.1)$, $\tau = 0.05$, $m = 0.01$, $K_{\mathrm{init}} = 3K_{\mathrm{seen}}$, and seeds $\{0, 1, 2\}$, unchanged across target datasets and feature sources. We separately evaluate source-calibrated profiles through leave-one-dataset-out (LODO) selection: both feature sources of one dataset are held out, and one of 4,576 profiles is chosen using true class counts from the other three datasets. The profile is then frozen for the target; target $K^\star$ and novel labels are excluded from selection.

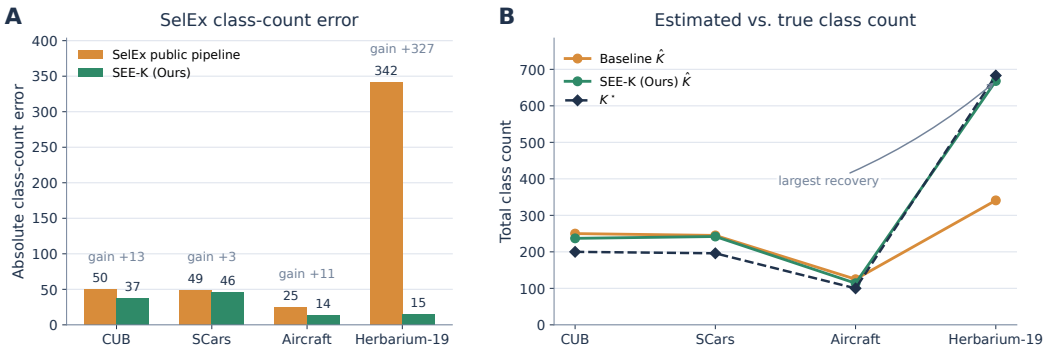

Figure 3: **Aligned SelEx unknown-K comparison with SEE-K.** The main decision-layer gain is count calibration, especially the correction of Herbarium-19 underestimation.

Table 1: **Class-count calibration in the aligned SelEx unknown-K comparison.** The primary decision-layer gain is improved class-count estimation. SEE-K reduces absolute class-count error on all four fine-grained datasets, with the largest correction on Herbarium-19.

Dataset	$K^\star$	SelEx $\widehat{K}$	SEE-K $\widehat{K}$	SelEx $\lvert e_K \rvert$	SEE-K $\lvert e_K \rvert$	Gain
CUB	200	250	237	50	37	+13
SCars	196	245	242	49	46	+3
Aircraft	100	125	114	25	14	+11
Herbarium-19	683	341	668	342	15	+327

5 Experiments

5.1 Experimental Setup

We evaluate the proposed selector as a fixed-feature decision layer, without retraining the representation backbone. The main fine-grained evaluation uses CUB [23], SCars [24], Aircraft [25], and Herbarium-19 [26]. We test two complementary backbone families: SimGCD [2], a canonical parametric GCD baseline, and SelEx [3], a recent fine-grained GCD system whose released implementation exposes a public unknown-K estimate. CIFAR100 [27] and ImageNet100 [28] are reserved for broader-domain failure-mode audits. We additionally test frozen features from AFGCD [18], RLCD [19], and DebGCD [20] on the same four fine-grained datasets, keeping the selector parameters and procedure unchanged.

The protocol isolates the class-count decision after representation learning. For each backbone–dataset pair, we reuse the fixed feature bank, construct our seen-constrained merge path, select $\widehat{K}$, and report all-class and novel-class clustering accuracy. We compare against the diagnostic known-K reference for both backbones and against the public unknown-K estimate where it is available, namely SelEx. The main tables report aligned fixed-feature point estimates; Appendix L provides supplementary multi-seed repeated runs, paired tests, and bootstrap intervals for representative settings. Supplementary sections also report exact defaults, fixed-K stress tests, policy comparisons, profile sensitivity, compute resources, and dataset-license details.

5.2 Main Results and Decision-Layer Comparisons

Class-count calibration. The cleanest aligned unknown-K comparison is SelEx, where the released implementation exposes a public class-count estimate. Table 1 shows that our selector reduces absolute class-count error on all four fine-grained datasets. The largest correction occurs on Herbarium-19: SelEx underestimates the total count by 342 classes, while our selected count is within 15 classes of $K^\star$. Figure 3 summarizes the same count-calibration result visually.

Comparison with unknown-K criteria. Appendix D.1 compares the fixed default with five external selection rules on the same eight frozen feature banks. The default has the lowest mean

Table 2: **Clustering quality after SEE-K selection.** "Known-K" is a diagnostic backbone-specific reference, not an upper bound. Mixed accuracy outcomes motivate evaluating count calibration and clustering quality separately.

		All		New		
Backbone	Dataset	Diagnostic known-K	SEE-K	Diagnostic known-K	SEE-K	SEE-K $\widehat{K}$
SimGCD	CUB	0.6181	0.6728	0.6046	0.5669	214
SimGCD	SCars	0.5182	0.6081	0.4202	0.4330	237
SimGCD	Aircraft	0.5233	0.5443	0.4861	0.3493	90
SimGCD	Herbarium-19	0.4398	0.5116	0.3656	0.3649	388
SelEx	CUB	0.7900	0.7302	0.7828	0.6523	237
SelEx	SCars	0.5923	0.6209	0.4955	0.4096	242
SelEx	Aircraft	0.6077	0.6222	0.5631	0.4822	114
SelEx	Herbarium-19	0.4615	0.5491	0.3782	0.3815	668

Table 3: **Same-path $K^\star$-cut diagnosis.** "$K^\star$-merge" fixes the cut to the ground-truth count on our merge path; "decision gap" is selected all-class accuracy minus $K^\star$-merge all-class accuracy.

Backbone	Dataset	$K^\star$	Selected $\widehat{K}$	$K^\star$-merge all	Selected all	Decision gap
SimGCD	CUB	200	214	0.6441	0.6728	$+0.0287$
SimGCD	SCars	196	237	0.5722	0.6081	$+0.0359$
SimGCD	Aircraft	100	90	0.5928	0.5443	-0.0484
SimGCD	Herbarium-19	683	388	0.5409	0.5116	-0.0293
SelEx	CUB	200	237	0.6538	0.7302	$+0.0764$
SelEx	SCars	196	242	0.5861	0.6209	$+0.0349$
SelEx	Aircraft	100	114	0.5530	0.6222	$+0.0691$
SelEx	Herbarium-19	683	668	0.5478	0.5491	$+0.0012$

relative count error in this comparison, while spherical GMM-BIC has the lowest mean absolute error. Appendix D.1 gives per-setting counts and separately evaluates the stability-only simplification.

Clustering quality after selection. Table 2 reports final clustering accuracy under the selected counts. These numbers should be read together with the count table rather than as a standalone leaderboard. Count quality and clustering quality are coupled but not identical: SelEx Herbarium-19 is a count-dominated case where correcting $\widehat{K}$ improves the final partition, while SelEx CUB shows that a better class-count decision does not remove all path or representation limitations. SimGCD Herbarium-19 improves all-class accuracy but still retains a substantial count gap.

Same-path $K^\star$ diagnosis. Table 3 compares selected and exact-count cuts on fixed feature banks and merge paths. The $K^\star$ cut improves All accuracy on SimGCD Aircraft and Herbarium-19, whereas the selected cut performs better on SelEx CUB and Aircraft; SelEx Herbarium-19 has nearly equal accuracy at both cuts. Appendix M extends this comparison to eight accuracy landscapes and CUB partition diagnostics.

Effect of model order on Herbarium-19. Figure 4 varies K on three fixed SelEx Herbarium-19 paths. Moving from the minimum feasible cut $K = 342$ to $\widehat{K} = 668$ raises All/New accuracy from 0.4303/0.1716 to 0.5454/0.3763 and reduces novel false merges, while increasing false splits. Further correction to $K^\star = 683$ changes accuracy little; $K = 750$ yields higher accuracy but greater false splitting and 9.8% count error. Thus, correcting severe underestimation improves assignments, while count fidelity and assignment accuracy remain distinct.

Partition changes from 200 to 237 clusters on CUB. On the SelEx CUB seed-2 path, changing K from 200 to 237 preserves Old accuracy at 0.8789 and raises New accuracy from 0.5098 to 0.6523, while overestimating the class count by 37. Only 23 of 200 parent clusters are subdivided, including 20 whose children have different dominant labels. Appendix M.2 reports the corresponding category-recovery statistics.

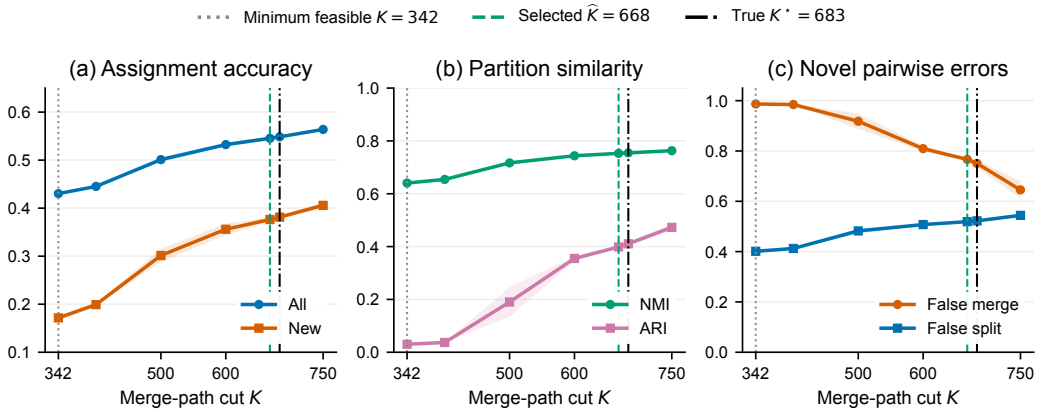

Figure 4: **Model order on fixed SelEx Herbarium-19 paths.** Means and one sample standard deviation across three paths. Vertical lines mark the minimum feasible count (342), selected count (668), and true count (683). Pairwise error definitions are given in Appendix M.1.

Table 4: **Risk-control ablations.** Components control different failure modes; removing one may improve pointwise accuracy on some settings while worsening calibration or robustness.

Variant	Removed control	Avg. Δ all	Avg. Δ new	Avg. $\lvert e_K \rvert$	Main failure mode
$K_{\text{init}} = 2K_{\text{seen}}$	Sufficient refinement	-0.0171	-0.0045	67.6	Under-refined unlabeled-side structure
$K_{\text{init}} = 4K_{\text{seen}}$	Controlled refinement	-0.0450	-0.0829	86.4	Fragmentation instability
No stability	Cross-seed reliability	-0.0527	-0.0944	110.1	Seed-specific partitions
No merge-path	Nested model-order path	+0.0133	+0.0095	147.0	High-K endpoint selection
No seen consistency	Seen-class constraint	+0.0175	+0.0285	54.5	Over-fragmentation / seen drift

Transfer across frozen feature sources. The unchanged selector applies to AFGCD, RLCD, and DebGCD features without retuning, with a mean relative count error of 0.2099 across all twelve settings. Herbarium-19 remains challenging: selected counts are 375, 496, and 421, respectively, against $K^\star = 683$. Appendix J reports the full results.

5.3 Ablation and Robustness Analysis

Table 4 and Figure 5(A,B) summarize the effects of decision-layer risk controls averaged across SimGCD and SelEx on CUB, SCars, Aircraft, and Herbarium-19. The K_{init} variants first test the coverage budget: $2K_{\text{seen}}$ is often under-refined, while $4K_{\text{seen}}$ increases fragmentation sensitivity. The remaining ablations remove risk controls. Removing stability produces the clearest average accuracy drop, while removing the merge path or seen consistency can improve average accuracy but worsens calibration or fragmentation behavior. The supplement expands these results with per-setting ablations, policy swaps, profile sweeps, repeated runs, and fixed-K stress tests.

Source-calibrated profile selection. Across the eight fine-grained settings, minimax LODO reduces mean relative count error from 0.1741 to 0.1377 (20.9%), while mean All/New accuracy decreases from 0.6074/0.4550 to 0.5885/0.4205. Calibration therefore improves held-out count estimation with an assignment-accuracy trade-off. Other main results retain the fixed default; Appendix E gives the calibration rule and per-fold results.

Long-tailed discovery pools. With features frozen after training on the original CIFAR100 data, we downsample only the unlabeled discovery pool and retain all labeled anchors. Across five feature sources and five sampling constructions, increasing nominal imbalance from 10 to 100 raises All accuracy while reducing rare-class recall, revealing rare-category loss hidden by sample-weighted accuracy. Appendix I details the protocol, realized frequency ratios, count errors, and recovery.

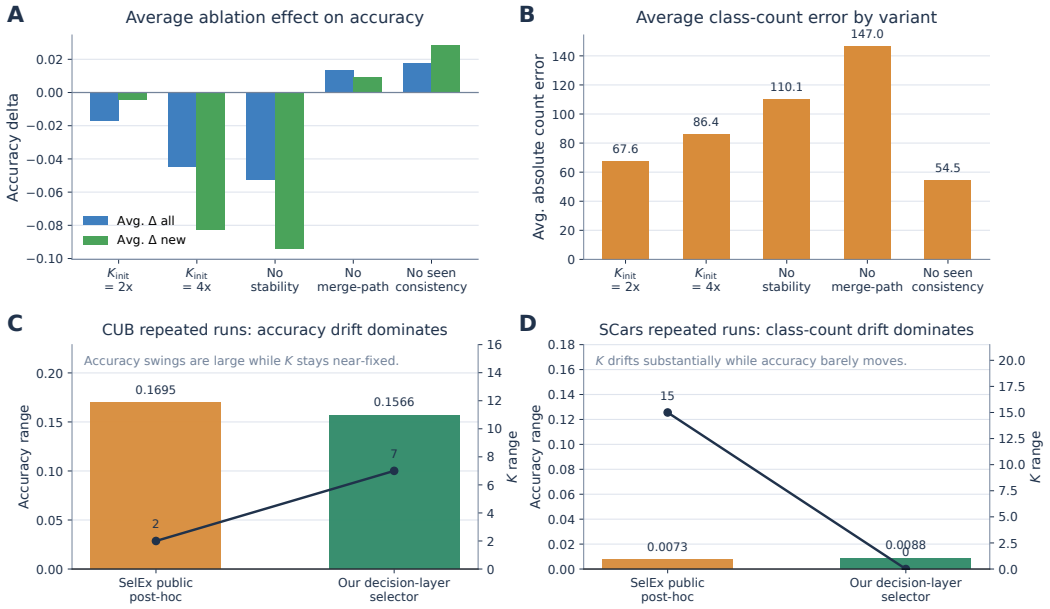

Figure 5: **SEE-K risk-control and robustness overview.** Ablations expose different decision risks, while repeated runs show that sensitivity can appear as either accuracy drift or class-count drift.

5.4 Scope Extension: Broader-Domain Audit

The broader-domain runs are diagnostic audits, not primary performance claims. With the fine-grained default profile, ImageNet100 exhibits a near-seen underestimation regime: SimGCD selects $\widehat{K} = 57$ and SelEx selects $\widehat{K} = 54$ for $K^\star = 100$, with novel-class accuracies of 0.0215 and 0.0301. CIFAR100 behaves differently: default cuts achieve substantially higher novel-class accuracy, and Appendix H shows that moving toward more accuracy-oriented cuts can sacrifice count calibration. The signal-triggered intervention is therefore not part of the default selector; it is used only to test whether a larger-K cut on the same landscape explains a failure. These audits show why unknown-K diagnosis requires more than final accuracy.

6 Conclusion

We presented SEE-K for unknown-K selection over frozen GCD features. SEE-K combines a seen-constrained candidate family, observable scoring criteria, and a coarsest near-optimal cut rule. Experiments with five GCD feature sources show that the selector can be applied without retuning, while its count errors remain dependent on the feature source and dataset. Fixed-path analyses demonstrate that count fidelity and assignment accuracy capture different aspects of partition quality. Source calibration improves held-out count estimation with an assignment-accuracy trade-off, and long-tailed discovery-pool sampling reveals declining rare-class recovery even as All accuracy rises. We advocate joint evaluation of count estimation, assignment quality, and category recovery.

Limitations and Broader Impact. The study has several boundaries. We evaluate five GCD feature sources under one fixed default selector profile, which is not calibrated for all domains or feature geometries. The selection risks are observable surrogates for seen consistency, stability, compactness, and degeneracy; they make the decision auditable but do not guarantee recovery of the semantic class count. The candidate family is also constrained by the initial overclustering and non-anchor merge path, so missing or poorly ordered path states cannot be recovered at selection time, and the broader-domain intervention is a diagnostic probe rather than a default calibration rule. These same diagnostics can improve transparency in open-world discovery by revealing over-merging, over-fragmentation, and seed sensitivity, but they should not be treated as safeguards: incorrect count decisions may still merge important categories, create misleading fragments, or propagate errors into retrieval and monitoring analyses.

Acknowledgments and Disclosure of Funding

Qiang Zhou was supported by the National Natural Science Foundation of China under Grant No. 62106045, the Jiangsu Provincial Double-Innovation Doctor Program under Grant No. JSSCBS20220185, and the Fundamental Research Funds for the Central Universities under Grant No. 2242026K30047. This work was supported by the Big Data Computing Center of Southeast University.

The authors declare no competing interests.

References

[1] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7492–7501, 2022.

[2] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16590–16600, 2023.

[3] Sarah Rastegar, Mohammadreza Salehi, Yuki M Asano, Hazel Doughty, and Cees GM Snoek. SelEx: Self-expertise in fine-grained generalized category discovery. In *European Conference on Computer Vision*, pages 440–458. Springer, 2024.

[4] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *IEEE/CVF International Conference on Computer Vision*, pages 8401–8409, 2019.

[5] Florent Chiaroni, Jose Dolz, Ziko Imtiaz Masud, Amar Mitiche, and Ismail Ben Ayed. Parametric information maximization for generalized category discovery. In *IEEE/CVF International Conference on Computer Vision*, pages 1729–1739, 2023.

[6] Bingchen Zhao, Xin Wen, and Kai Han. Learning semi-supervised gaussian mixture models for generalized category discovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16623–16633, 2023.

[7] Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Shahbaz Khan. PromptCAL: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3479–3488, 2023.

[8] Wenbin An, Feng Tian, Qinghua Zheng, Wei Ding, QianYing Wang, and Ping Chen. Generalized category discovery with decoupled prototypical network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 12527–12535, 2023.

[9] Sarah Rastegar, Hazel Doughty, and Cees G. M. Snoek. Learn to categorize or categorize to learn? self-coding for generalized category discovery. In *Advances in Neural Information Processing Systems*, 2023.

[10] Jona Otholt, Christoph Meinel, and Haojin Yang. Guided Cluster Aggregation: A hierarchical approach to generalized category discovery. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2618–2627, 2024.

[11] Hongjun Wang, Sagar Vaze, and Kai Han. SPTNet: An efficient alternative framework for generalized category discovery with spatial prompt tuning. In *International Conference on Learning Representations*, 2024.

[12] Mingfu Yan, Jiancheng Huang, Yifan Liu, and Shifeng Chen. Component adaptive clustering for generalized category discovery. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2025. doi: 10.1109/ICME59968.2025.11209408.

[13] Wei Feng and Zongyuan Ge. Generalized category discovery under domain shift: A frequency domain perspective. In *Advances in Neural Information Processing Systems*, 2025.

[14] Geeho Kim, Jinu Lee, and Bohyung Han. Fedlpa: Local prior alignment for heterogeneous federated generalized category discovery. In *Advances in Neural Information Processing Systems*, 2025.

[15] Jizhou Han, Shaokun Wang, Yuhang He, Chenhao Ding, Qiang Wang, Xinyuan Gao, SongLin Dong, and Yihong Gong. Consistent supervised-unsupervised alignment for generalized category discovery. In *Advances in Neural Information Processing Systems*, 2025.

[16] Vaibhav Rathore, Divyam Gupta, and Biplab Banerjee. HIDISC: A hyperbolic framework for domain generalization with generalized category discovery. In *Advances in Neural Information Processing Systems*, 2025.

[17] Zhenqi He, Yuanpei Liu, and Kai Han. SEAL: Semantic-aware hierarchical learning for generalized category discovery. In *Advances in Neural Information Processing Systems*, 2025.

[18] Qiyu Xu, Zhanxuan Hu, Yu Duan, Ercheng Pei, and Yonghang Tai. A hidden stumbling block in generalized category discovery: Distracted attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 405–414, 2025.

[19] Duo Liu, Zhiquan Tan, Linglan Zhao, Zhongqiang Zhang, Xiangzhong Fang, and Weiran Huang. Generalized category discovery via reciprocal learning and class-wise distribution regularization. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

[20] Yuanpei Liu and Kai Han. DebGCD: Debiased learning with distribution guidance for generalized category discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

[21] Fang Zhou, Zhiqiang Chen, Martin Pavlovski, and Yizhong Zhang. ReLKD: Inter-class relation learning with knowledge distillation for generalized category discovery. In *European Conference on Artificial Intelligence*, 2025.

[22] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

[23] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[25] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[26] Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.

[27] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[29] Haiyang Zheng, Nan Pu, Wenjing Li, Nicu Sebe, and Zhun Zhong. Generalized fine-grained category discovery with multi-granularity conceptual experts. *arXiv preprint arXiv:2509.26227*, 2025.

Algorithm 1 SEE-K: Seen-constrained unknown-K selection

Require: Feature bank X, labeled seen set $\mathcal{L}$, unlabeled set $\mathcal{U}$, seen-class count K_{seen}, initial count K_{init}, seeds $\mathcal{R}$, plateau margin m

Ensure: Estimated class count $\widehat{K}$ and final partition $\hat{\mathcal{C}}$

for seed $r \in \mathcal{R}$ **do**
    Initialize seen centroids from labeled seen-category prototypes.
    Initialize non-anchor centroids on $\mathcal{U}$ with k-means++.
    Run seen-constrained semi-supervised k-means.
    Construct the non-anchor merge path $\{\mathcal{C}_K^{(r)}\}$.
end for
for candidate count K shared by all seed paths **do**
    Compute seen consistency, compactness, and tiny-cluster penalty on each $\mathcal{C}_K^{(r)}$ and average them over seeds.
    Compute cross-seed stability as mean pairwise NMI on unlabeled assignments at K.
    Normalize diagnostics and compute the aggregated decision score $\mathcal{S}(K)$.
end for
Select $\widehat{K} = \min\{K : \mathcal{S}(K) \geq \max_{K'} \mathcal{S}(K') - m\}$.
Select seed $\hat{r}$ at $\widehat{K}$ by highest seen consistency, then higher compactness, then lower tiny-cluster penalty.
return $\widehat{K}$ and $\hat{\mathcal{C}} = \mathcal{C}_{\widehat{K}}^{(\hat{r})}$.

A Algorithmic Summary

Algorithm interpretation. Algorithm 1 gives an implementation-oriented summary of the selector described in the Method section of the main paper. It separates three operations that are easy to conflate in prose: seed-specific construction of candidate merge paths, cross-seed aggregation of observable decision risks for each candidate count, and deterministic selection of one final seed-specific partition after $\widehat{K}$ has been chosen. The returned partition $\hat{\mathcal{C}}$ is the concrete clustering used for all reported all-class and novel-class accuracy metrics; neither the plateau selection nor the final seed rule uses novel labels, $K^\star$, or clustering accuracy.

B Additional Experimental Details

Fixed quantities across runs. All main results reuse existing backbone checkpoints and feature banks generated before running our decision-layer selector. Our method does not retrain the backbone for the reported four-dataset comparison.

Default search hyperparameters. The default search uses three seeds $\{0, 1, 2\}$, $K_{\text{init}} = 3K_{\text{seen}}$, seen-consistency tolerance $\tau = 0.05$, plateau margin $m = 0.01$, and weights $(\lambda_a, \lambda_s, \lambda_f, \lambda_g) = (0.4, 0.3, 0.2, 0.1)$ for seen-anchor consistency, stability, compactness, and tiny-cluster penalty. These values define conservative default settings rather than theoretically optimal hyperparameters. The choice $K_{\text{init}} = 3K_{\text{seen}}$ reserves one seen-category centroid per known class and leaves an approximate $2\times$ non-anchor overclustering budget for candidate unlabeled-side structure under the roughly balanced standard benchmark splits; in the main-paper ablation, $2K_{\text{seen}}$ is typically too conservative and $4K_{\text{seen}}$ is less stable. The weight profile instantiates the surrogate decision-risk view in the main paper: seen consistency is the primary seen-structure term, stability is the main perturbation-reliability term, compactness is a secondary geometric-fidelity cue, and the tiny-cluster term acts as a lightweight degeneracy regularizer.

Seed aggregation and final partition. For seed-dependent quantities, we compute seen consistency, compactness, and the tiny-cluster penalty on each seed-specific cut $\mathcal{C}_K^{(r)}$ and average the resulting scalar values over seeds. Stability is the exception: $s_{\text{stab}}(K)$ is the mean pairwise NMI between unlabeled assignments from different seeds at the same K. Thus $\mathcal{S}(K)$ is a K-level score aggregated across seed-specific paths, not the score of a single seed path. After selecting $\widehat{K}$, we return one concrete seed-specific partition using a deterministic selector-side rule: among seed cuts at $\widehat{K}$, choose

the seed with the highest seen consistency, breaking ties by higher compactness and then by lower tiny-cluster penalty. This rule uses only diagnostics available under the standard GCD supervision; it does not use novel labels, $K^\star$, or all/new clustering accuracy. The same seed-selection rule is used for fixed-K and $K^\star$-merge diagnostics, with K fixed before selecting the representative seed.

Stress-test protocol. The $K^\star$-centered fixed-K stress test evaluates nearby class counts around $K^\star$ using relative offsets $\delta \in \{-0.3, -0.2, -0.1, +0.1, +0.2, +0.3\}$. Each candidate is rounded to the nearest valid class count and clipped not to fall below K_{seen}.

Compute note. The full workflow mixes GPU and CPU stages. Feature extraction follows the original backbone implementations and is GPU accelerated, while overclustering, merge-path search, and K-stress evaluation are primarily CPU-bound.

C Full Ablation Table

Table 5 expands the averaged ablation in the main paper into per-setting rows. The table should be read together with the discussion in the main text: several variants improve accuracy on a subset of datasets, but the same variants often degrade class-count calibration, especially on Herbarium-19.

Table 5: Full ablation results for the eight main fine-grained backbone-dataset settings used in the paper.

Backbone	Dataset	Variant	Best K	$K^\star$	Acc. all	Acc. new
SimGCD	CUB	no_merge_path	300	200	0.6823	0.5772
SimGCD	CUB	no_seen_consistency	214	200	0.6728	0.5669
SimGCD	CUB	no_stability	205	200	0.6608	0.5462
SimGCD	CUB	$K_{\text{init}} = 2\times$	163	200	0.6058	0.4985
SimGCD	CUB	$K_{\text{init}} = 4\times$	231	200	0.6538	0.5362
SimGCD	SCars	no_merge_path	294	196	0.5988	0.4123
SimGCD	SCars	no_seen_consistency	230	196	0.6012	0.4222
SimGCD	SCars	no_stability	222	196	0.5976	0.4183
SimGCD	SCars	$K_{\text{init}} = 2\times$	176	196	0.5866	0.3978
SimGCD	SCars	$K_{\text{init}} = 4\times$	245	196	0.5810	0.4111
SimGCD	Aircraft	no_merge_path	150	100	0.6066	0.4522
SimGCD	Aircraft	no_seen_consistency	116	100	0.6118	0.4717
SimGCD	Aircraft	no_stability	53	100	0.4105	0.1010
SimGCD	Aircraft	$K_{\text{init}} = 2\times$	87	100	0.6018	0.4465
SimGCD	Aircraft	$K_{\text{init}} = 4\times$	56	100	0.4068	0.1109
SelEx	CUB	no_merge_path	300	200	0.7536	0.6854
SelEx	CUB	no_seen_consistency	237	200	0.7302	0.6523
SelEx	CUB	no_stability	261	200	0.7624	0.7120
SelEx	CUB	$K_{\text{init}} = 2\times$	180	200	0.6828	0.6036
SelEx	CUB	$K_{\text{init}} = 4\times$	178	200	0.6034	0.3971
SelEx	SCars	no_merge_path	294	196	0.6559	0.4649
SelEx	SCars	no_seen_consistency	243	196	0.6221	0.4115
SelEx	SCars	no_stability	241	196	0.6209	0.4096
SelEx	SCars	$K_{\text{init}} = 2\times$	170	196	0.5922	0.4096
SelEx	SCars	$K_{\text{init}} = 4\times$	311	196	0.6197	0.3971
SelEx	Aircraft	no_merge_path	150	100	0.6558	0.5331
SelEx	Aircraft	no_seen_consistency	128	100	0.6538	0.5388
SelEx	Aircraft	no_stability	60	100	0.4707	0.2141
SelEx	Aircraft	$K_{\text{init}} = 2\times$	81	100	0.5805	0.4648
SelEx	Aircraft	$K_{\text{init}} = 4\times$	125	100	0.5889	0.3922
SimGCD	Herbarium-19	no_merge_path	1023	683	0.4993	0.2987
SimGCD	Herbarium-19	no_seen_consistency	617	683	0.5390	0.3917
SimGCD	Herbarium-19	no_stability	367	683	0.4856	0.3179
SimGCD	Herbarium-19	$K_{\text{init}} = 2\times$	388	683	0.5180	0.3757
SimGCD	Herbarium-19	$K_{\text{init}} = 4\times$	398	683	0.5000	0.3576
SelEx	Herbarium-19	no_merge_path	1023	683	0.5137	0.2917
SelEx	Herbarium-19	no_seen_consistency	877	683	0.5686	0.4127
SelEx	Herbarium-19	no_stability	342	683	0.4289	0.1656
SelEx	Herbarium-19	$K_{\text{init}} = 2\times$	572	683	0.5549	0.4070
SelEx	Herbarium-19	$K_{\text{init}} = 4\times$	803	683	0.5460	0.3745

D Selector Policy Comparison on the Same Merge Path

We compare cut-selection policies on fixed merge paths, followed by external unknown-K criteria on the same frozen feature banks.

Table 6: Selector-policy comparison on the same merge path for the eight main fine-grained backbone-dataset settings. "Default" denotes SEE-K with the fixed default profile used in the main paper. "Best-all" reports the policy with the highest all-class accuracy among the tested alternatives, while "Min-$|e_K|$" reports the policy with the smallest absolute class-count error.

Backbone	Dataset	Default $\widehat{K}$	Default all	Best-all policy	Best all	Min-$\lvert e_K\rvert$ policy	Min $\lvert e_K\rvert$
SimGCD	CUB	214	0.6728	compactness_only	0.6985	default	14
SimGCD	SCars	237	0.6081	argmax_default	0.6104	default	41
SimGCD	Aircraft	90	0.5443	compactness_only	0.6232	argmax_default	7
SimGCD	Herbarium-19	388	0.5116	stability_only	0.5409	stability_only	1
SelEx	CUB	237	0.7302	compactness_only	0.7654	stability_only	8
SelEx	SCars	242	0.6209	compactness_only	0.6488	default	46
SelEx	Aircraft	114	0.6222	compactness_only	0.6606	default	14
SelEx	Herbarium-19	668	0.5491	stability_only	0.5702	default	15

Interpretation. This table explains why some ablations or reduced policies can appear stronger when judged only by accuracy. On the main eight-setting fine-grained slice, `compactness_only` or `stability_only` yields the highest all-class accuracy in seven settings, whereas the default policy gives the smallest class-count error in five settings. The default selector is therefore a conservative model-order operating point balancing accuracy, calibration, and robustness, rather than a policy optimized solely for pointwise accuracy on each path.

D.1 Unknown-K Criteria and Stability-Only Selection

Table 7 summarizes count errors across the eight settings; Table 8 gives the individual counts.

All rules use the same frozen feature banks. Calinski–Harabasz (CH) and Davies–Bouldin (DB) score all available cuts using at most 4,096 sampled unlabeled points. Silhouette and spherical GMM-BIC use 1,024 sampled unlabeled points and at most 61 cuts. Semi-Infomap uses the public implementation accompanying Zheng et al. [29], with labeled seen samples selecting the neighborhood size; clusters of at most four samples are excluded from its count estimate. Target $K^\star$, novel labels, and clustering accuracy do not enter selection.

Stability-only simplification. Using cross-seed stability alone on the same merge paths yields lower mean absolute/relative count errors (44.25/0.1643) than the fixed four-term default (59.00/0.1741). Thus, the combined default does not improve aggregate count calibration over this simpler rule. Table 9 reports the per-setting results.

Profile-sensitivity interpretation. The same decision-risk logic does not have uniform sensitivity across datasets. SCars is relatively insensitive under both backbones: the all-class range is only 0.0128 on SimGCD and 0.0091 on SelEx across seven profile changes. Herbarium-19, Aircraft, and the broader-domain settings are more sensitive. Tight seen-consistency tolerances can push SimGCD Aircraft down to $\widehat{K} = 58$ and SelEx Herbarium-19 to $\widehat{K} = 342$, while SelEx ImageNet100 changes from 0.4680 to 0.7450 all-class accuracy under a stability-heavy profile. This motivates treating the default profile as a conservative operating point rather than as a claim that one fixed weight setting is uniformly optimal.

E Source-Calibrated Profile Selection

LODO holds out both SimGCD and SelEx settings of one dataset. The search recombines cached path diagnostics over 286 nonnegative weight tuples summing to one (step 0.1), $\tau \in \{0.01, 0.03, 0.05, 0.10\}$, and $m \in \{0, 0.005, 0.01, 0.02\}$: 4,576 profiles in total. Initialization, seeds, and merge paths stay fixed.

Table 7: Unknown-K count errors on frozen SimGCD/SelEx features. The public SelEx estimate covers only its four settings and is reported separately.

Criterion	Settings	Mean absolute error	Mean relative error
SEE-K (fixed default)	8	59.00	**0.1741**
Spherical GMM-BIC	8	**44.75**	0.1812
Semi-Infomap	8	79.62	0.1826
Calinski–Harabasz	8	74.75	0.2700
Silhouette	8	89.62	0.3020
Davies–Bouldin	8	105.50	0.3462
SelEx public estimate	4	116.50	0.3127

Table 8: Selected counts by setting. BIC denotes spherical GMM-BIC; SI denotes Semi-Infomap; Sil. denotes Silhouette. Public estimates are available only for SelEx.

Source	Dataset	$K^\star$	SEE-K	BIC	SI	CH	Sil.	DB	Public
SimGCD	CUB	200	214	227	204	145	246	224	–
SimGCD	SCars	196	237	242	152	241	245	99	–
SimGCD	Aircraft	100	90	128	81	124	126	51	–
SimGCD	Herbarium-19	683	388	682	403	535	818	342	–
SelEx	CUB	200	237	237	190	161	260	218	250
SelEx	SCars	196	242	193	167	120	281	135	245
SelEx	Aircraft	100	114	133	94	64	133	54	125
SelEx	Herbarium-19	683	668	500	438	508	966	891	341

For each source dataset, we average relative count error over its two feature sources. The minimax rule minimizes the largest of these dataset averages. Ties are broken by mean relative error, worst individual relative error, mean absolute error, then the summed absolute distance of the six profile parameters from the default. Any remaining tie uses ascending lexicographic grid order. All these comparisons use source data only. Tables 11 and 12 report the frozen profiles and held-out outcomes. Mean absolute count error falls from 59.00 to 54.50; accuracy decreases alongside the relative count-error improvement.

F Main Fine-Grained $K^\star$-Centered Fixed-K Stress Summary

The K-stress test is a $K^\star$-centered fixed-K probe, not a perturbation of the selected $\widehat{K}$ and not a merge-path neighborhood analysis. For each offset, we rerun semi-supervised k-means with the corresponding fixed K and measure how much the clustering accuracy changes. A small range means that fixed-K evaluations near $K^\star$ behave similarly; a large range means that the reported accuracy is sensitive even to class-count perturbations in the $K^\star$ neighborhood.

K-stress interpretation. The largest contrast is between SelEx Herbarium-19 and SelEx CUB. Herbarium-19 has a very small all-class accuracy range, suggesting that the $K^\star$ neighborhood is locally stable once the underestimation failure is corrected. CUB has the largest range, which is consistent with the repeated-execution analysis: on this dataset, even fixed-K perturbations around $K^\star$ can induce large performance shifts.

G Larger Seed-Budget Audit

This audit is separate from the main default profile, which uses three overclustering seeds. It compares two larger seed-budget profiles to test whether increasing the number of perturbation seeds changes the selected cut; the 'Seed5' rows are therefore not the main default point estimates.

The effect is dataset-dependent: SCars is unchanged, Herbarium-19 moves only slightly, Aircraft improves modestly, and SelEx CUB changes substantially.

Table 9: Stability-only selected counts on the same paths. SEE-K default counts appear in Table 8.

Source	CUB	SCars	Aircraft	Herbarium-19
SimGCD	221	241	124	684
SelEx	192	243	118	873

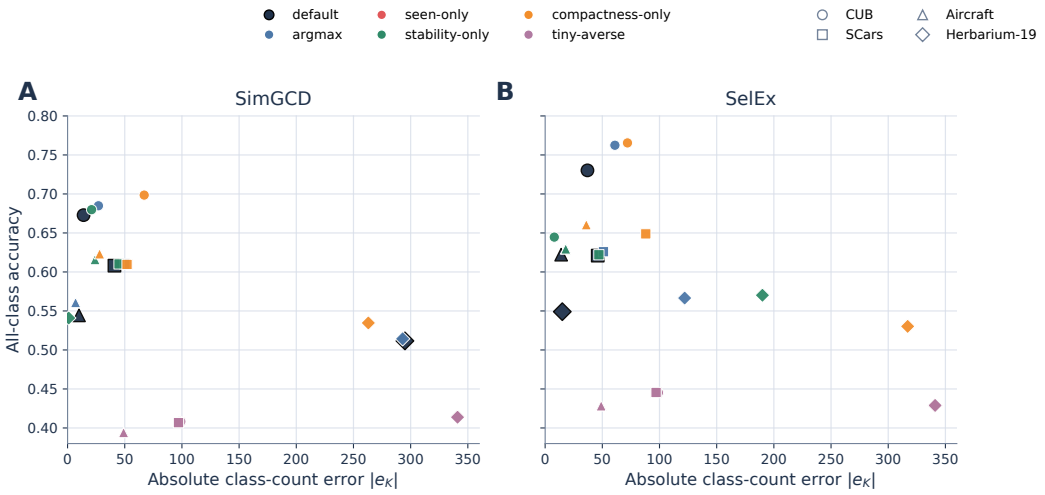

Figure 6: Policy-level trade-offs on the same merge path for the eight main fine-grained settings. Policies that maximize all-class accuracy on a fixed path often move farther away from the true class count, while the default selector stays closer to a conservative model-order operating point.

H Scope Extension Beyond Fine-Grained Datasets

Interpretation. The CIFAR100 rows provide limited evidence that the decision-layer framing is not confined to fine-grained bird, car, and aircraft benchmarks. At the same time, both ImageNet100 rows remain low on novel-class accuracy, with 0.0215 on SimGCD and 0.0301 on SelEx. We therefore treat the scope-extension results as diagnostic evidence rather than as a primary broad-domain generalization claim.

To determine whether this broader-domain degradation reflects path quality or a selector-side operating-point mismatch, we apply a signal-triggered diagnostic intervention. The intervention is indexed by observable signatures rather than dataset identity and is not part of the default selector. It tests whether selecting larger-K cuts along the same decision landscape can recover accuracy when a no-seen probe indicates that the default profile is too conservative for the broader-domain setting. In this diagnostic rule, this corresponds to $\widehat{K}_{\text{default}} \leq 1.8\, K_{\text{seen}}$ and $\widehat{K}_{\text{probe}} \geq \widehat{K}_{\text{default}} + 0.25\, K_{\text{seen}}$. Triggered cases first use a mild intervention with $K_{\text{init}} = 2K_{\text{seen}}$ and zero seen-consistency weight. We use a stronger intervention, with $K_{\text{init}} = 3K_{\text{seen}}$ and the same zero seen-weight, only when the probe indicates a larger shift toward higher K on the normalized K_{seen} scale. Because the broader-domain rows are seed-sensitive, we summarize the intervention behavior below with multi-seed repeated-run means rather than with only the single executions in Table 15.

Diagnostic-intervention interpretation. This table reports a diagnostic intervention profile rather than the default selector's repeated-run behavior. It asks whether moving along the same selector landscape can recover accuracy under an observable larger-K correction signature; the default repeated runs are reported separately in Table 23. Under that diagnostic interpretation, the intervention clarifies that the broader-domain degradation is not purely a representation failure. On ImageNet100, both backbones show a pronounced underestimation signature: the default selector stays too close to K_{seen}, and weakening the seen-consistency constraint moves the cut toward larger K values, recovering both all-class and novel-class accuracy while also reducing mean count error. CIFAR100 exhibits a different regime rather than another underestimation-dominated case. The same intervention

Table 10: Selector-profile sensitivity under seven conservative profile changes across twelve backbone-dataset settings. "All-class range" is the gap between the best and worst all-class accuracy among the tested profiles.

Backbone	Dataset	Default $\widehat{K}$	$[\widehat{K}_{\min}, \widehat{K}_{\max}]$	Distinct $\widehat{K}$	Default all	All-class range	Best profile	Worst profile
SimGCD	CUB	214	[203, 227]	5	0.6728	0.0274	no_plateau	wide_plateau
SimGCD	SCars	237	[219, 246]	5	0.6081	0.0128	no_plateau	wide_plateau
SimGCD	Aircraft	90	[58, 115]	6	0.5443	0.1732	loose_seen_tol	tight_seen_tol
SimGCD	Herbarium-19	388	[387, 617]	5	0.5116	0.0280	tight_seen_tol	wide_plateau
SimGCD	CIFAR100	130	[89, 162]	4	0.7595	0.1128	tight_seen_tol	wide_plateau
SimGCD	ImageNet100	57	[56, 58]	3	0.4628	0.0000	default	default
SelEx	CUB	237	[189, 261]	5	0.7302	0.1174	no_plateau	wide_plateau
SelEx	SCars	242	[236, 247]	5	0.6209	0.0091	no_plateau	wide_plateau
SelEx	Aircraft	114	[60, 118]	5	0.6222	0.1588	no_plateau	tight_seen_tol
SelEx	Herbarium-19	668	[342, 809]	7	0.5491	0.1377	loose_seen_tol	tight_seen_tol
SelEx	CIFAR100	178	[81, 188]	4	0.8359	0.1230	default	wide_plateau
SelEx	ImageNet100	54	[51, 117]	3	0.4680	0.2770	stability_heavy	default

Table 11: Profiles selected on the other three datasets and frozen for each held-out dataset.

Held-out dataset	λ_a	λ_s	λ_f	λ_g	τ	m
CUB	0.3	0.7	0	0	0.03	0.02
SCars	0.1	0.9	0	0	0.05	0.02
Aircraft	0.3	0.7	0	0	0.03	0.02
Herbarium-19	0.5	0.4	0	0.1	0.1	0.02

still improves all-class and novel-class accuracy on both backbones, but the mean absolute count error can increase, so we treat CIFAR100 as an accuracy-oriented operating-point intervention rather than as a general class-count calibration procedure. The more detailed paired significance and bootstrap summaries for these repeated runs are reported later in the supplementary repeated-run section.

I Long-Tailed Discovery-Pool Sampling

We use frozen SimGCD, SelEx, AFGCD, DebGCD, and RLCD features trained on the original CIFAR100 data. All labeled anchors are retained; only the unlabeled discovery pool is exponentially downsampled, with at least one sample per class. Nominal imbalance factors (IF) are 10, 50, and 100; the realized ratio is the maximum divided by the minimum retained unlabeled class frequency. Five construction seeds (2026–2030) vary the class ranks and retained samples. Within each construction, the fixed default selector uses its three overclustering seeds $\{0, 1, 2\}$. This evaluates sampling imbalance after representation learning, without long-tailed retraining.

Table 17 reports means and sample standard deviations over the five constructions. Rare-class recall is the fraction of the 25 least frequent retained classes whose best-matching cluster reaches F1 ≥ 0.5, evaluated on the unlabeled pool. From nominal IF 10 to 100, All accuracy rises for every feature source, while rare-class recall falls from 0.856–0.944 to 0.288–0.456. Count errors also vary across feature sources, so sample-weighted accuracy alone does not capture recovery under imbalance.

J Transfer across Frozen Feature Sources

We replace the frozen feature bank with AFGCD, RLCD, or DebGCD features while retaining the fixed default weights, τ, m, $K_{\text{init}} = 3K_{\text{seen}}$, seeds $\{0, 1, 2\}$, and merge procedure. Table 18 covers all four fine-grained datasets. Mean relative count error is 0.2099 over twelve settings; the three Herbarium-19 settings retain errors of 0.2738–0.4510.

K Additional Recent Open-Source Baselines

We include four recent open-source image-GCD baselines as supplementary breadth evidence: AFGCD [18], RLCD [19], DebGCD [20], and ReLKD [21]. We exclude these runs from the main table because the public repositories expose heterogeneous evaluation protocols, but they are

Table 12: Held-out results: each pair is fixed default / minimax LODO. Accuracies use the observable representative-seed rule.

Source	Dataset	$\widehat{K}$	Rel. count error	All	New
SimGCD	CUB	214 / 204	0.0700 / 0.0200	0.6728 / 0.6563	0.5669 / 0.5375
SimGCD	SCars	237 / 221	0.2092 / 0.1276	0.6081 / 0.5972	0.4330 / 0.4178
SimGCD	Aircraft	90 / 90	0.1000 / 0.1000	0.5443 / 0.5443	0.3493 / 0.3493
SimGCD	Herbarium-19	388 / 386	0.4319 / 0.4348	0.5116 / 0.5104	0.3649 / 0.3636
SelEx	CUB	237 / 186	0.1850 / 0.0700	0.7302 / 0.6406	0.6523 / 0.4838
SelEx	SCars	242 / 241	0.2347 / 0.2296	0.6209 / 0.6209	0.4096 / 0.4096
SelEx	Aircraft	114 / 107	0.1400 / 0.0700	0.6222 / 0.5788	0.4822 / 0.3982
SelEx	Herbarium-19	668 / 717	0.0220 / 0.0498	0.5491 / 0.5590	0.3815 / 0.4039
Mean		–	0.1741 / 0.1377	0.6074 / 0.5885	0.4550 / 0.4205

Table 13: Summary of fixed-K stress tests for the eight main fine-grained backbone-dataset settings. Each row aggregates 18 evaluations from six nearby class counts and three seeds.

Backbone	Dataset	$K_{\min}$	$K_{\max}$	Best K	Best acc. all	Worst K	Acc. range
SelEx	Aircraft	70	130	130	0.6468	70	0.0954
SelEx	CUB	140	260	260	0.7510	140	0.1559
SelEx	Herbarium-19	478	888	615	0.5329	888	0.0088
SelEx	SCars	137	255	255	0.6555	137	0.0956
SimGCD	Aircraft	70	130	110	0.6241	70	0.0863
SimGCD	CUB	140	260	260	0.6875	140	0.1122
SimGCD	Herbarium-19	478	888	478	0.5438	888	0.0429
SimGCD	SCars	137	255	235	0.6146	137	0.0555

still useful for checking whether broader-domain sensitivity remains visible beyond the two main backbones.

Interpretation. These reproduced recent baselines provide useful breadth evidence, but they are not sufficiently aligned to replace our main-table backbone pair. First, the public repositories do not align on protocol: DebGCD [20] and ReLKD [21] expose test-side numbers, while AFGCD [18] and RLCD [19] mainly report train-side SS-kmeans metrics. Second, even after retaining only interpretable outputs, several reproduced rows remain imperfectly aligned with our fixed-backbone decision-layer setting. Third, the broader-domain rows remain sensitive to protocol and seed choices. DebGCD and RLCD both degrade substantially on ImageNet100 under our reproduction, even though many fine-grained and CIFAR100 rows obtain substantially higher accuracy. We therefore use these baselines as supplementary context rather than as direct replacements for the fixed-backbone decision-layer analysis in the main text.

L Supplementary End-to-End Multi-Seed Repeated Runs

This section reports supplementary repeated-run evidence for representative settings. These runs are used only to characterize seed sensitivity and paired variability; they are not used to choose the main-table configurations or to update the point estimates reported in the main aligned comparison.

The controlled repeated runs serve three purposes. First, they check whether the Herbarium-19 result persists across independent training seeds. Second, they extend the repeated-run check to additional fine-grained settings under explicitly seeded execution. Third, they separate positive evidence from boundary evidence: Herbarium-19 remains the most reliable setting, while Aircraft and the broader-domain rows expose larger seed sensitivity.

Interpretation. The controlled repeated runs separate into four regimes. Herbarium-19 remains the most stable positive case, but it does so on two backbones with different profiles. For SelEx, all eight repeated runs improve both all-class and novel-class accuracy over the baseline, while keeping $\widehat{K}$ close to the true count compared with the original underestimation. The paired t tests remain

Table 14: Five-seed vs. seven-seed selector comparison on the eight main fine-grained backbone-dataset settings.

Backbone	Dataset	Seed5 $\widehat{K}$	Seed7 $\widehat{K}$	ΔK	Seed5 all	Seed7 all	Δ all	Seed5 new	Seed7 new
SimGCD	CUB	214	212	-2	0.6728	0.6690	-0.0038	0.5669	0.5596
SimGCD	SCars	228	228	0	0.6012	0.6012	0.0000	0.4222	0.4222
SimGCD	Aircraft	92	94	+2	0.5524	0.5692	+0.0168	0.3643	0.3976
SimGCD	Herbarium-19	385	387	+2	0.5005	0.5110	+0.0104	0.3521	0.3647
SelEx	CUB	206	229	+23	0.6577	0.7144	+0.0567	0.5155	0.6220
SelEx	SCars	240	240	0	0.6206	0.6206	0.0000	0.4091	0.4091
SelEx	Aircraft	113	112	-1	0.6135	0.6249	+0.0114	0.4651	0.4768
SelEx	Herbarium-19	688	692	+4	0.5493	0.5504	+0.0011	0.3819	0.3822

Table 15: Scope-extension diagnostics beyond the four fine-grained datasets.

Backbone	Dataset	K_{seen}	$K^\star$	$\widehat{K}$	Acc. all	Acc. new	NMI	ARI
SimGCD	CIFAR100	80	100	130	0.7595	0.6066	0.8354	0.6318
SelEx	CIFAR100	80	100	178	0.8359	0.6670	0.8748	0.7632
SimGCD	ImageNet100	50	100	57	0.4628	0.0215	0.6795	0.0478
SelEx	ImageNet100	50	100	54	0.4680	0.0301	0.6625	0.0475

highly significant after Holm correction; the exact Wilcoxon test is still significant before correction, but rises slightly above 0.05 after expanding the family-wise correction. For SimGCD, all eight controlled Herbarium-19 repeated runs improve all-class accuracy, with mean Δ all $= +0.0669$ and a Holm-corrected paired t test below 10^{-6}, but the novel-class mean is essentially flat at Δ new $= -0.0023$ and is not significant.

The additional fine-grained controlled repeated runs occupy an intermediate regime between the positive Herbarium-19 results and the CUB boundary case. Under SimGCD, both CUB and SCars show reproducible all-class gains across eight matched seeds: Δ all is $+0.0534$ on CUB and $+0.0733$ on SCars, and both effects remain significant after Holm correction. However, these remain mixed cases rather than uniform improvements. CUB loses novel-class accuracy on all eight runs (Δ new $= -0.0486$), and SCars is approximately flat on novel-class accuracy (Δ new $= -0.0084$). SelEx SCars shows the same profile over eight runs: all-class accuracy improves on every run, with mean Δ all $= +0.0605$ and Holm-corrected paired t $p = 0.000464$, but novel-class accuracy decreases on all eight runs (Δ new $= -0.0402$) and remains significant only under the paired t test family.

Aircraft is the most pronounced boundary regime in these repeated runs. SelEx Aircraft stays roughly neutral on all-class accuracy even after extending to eight runs (Δ all $= +0.0133$, bootstrap CI crossing zero), but loses novel-class accuracy on every run, with a mean Δ new of -0.1209 and a bootstrap interval that remains negative. SimGCD Aircraft is more challenging: the all-class mean is slightly negative and the bootstrap interval still crosses zero, while the novel-class bootstrap interval remains negative at [-0.2638, -0.1011]. The bootstrap table therefore reinforces the paired tests: some settings show reproducible all-class improvements without corresponding novel-class gains, and Aircraft remains a boundary case for uniform-improvement claims.

The broader-domain repeated runs indicate a fourth regime. On CIFAR100, mean all-class accuracy is roughly unchanged, but novel-class accuracy and $\widehat{K}$ vary substantially across seeds. On ImageNet100, the variance is larger: one run reproduces the low-accuracy outcome reported above at $\widehat{K} = 54$ and only 0.0301 novel-class accuracy, while later seeds recover substantially better cuts. SelEx CUB remains the fine-grained boundary case in this repeated-run analysis: over eight runs, all-class and novel-class accuracy both decrease on every run, with Holm-corrected paired t tests of 0.0076 and 0.0032, respectively. We therefore treat the broader-domain repeated runs as boundary evidence for seed sensitivity rather than as direct average-gain results.

Table 16: Signal-triggered diagnostic intervention on the broader-domain repeated runs. "$\Delta|e_K|$" is intervention minus baseline mean absolute count error, so negative values mean improved class-count estimation.

Backbone	Dataset	Runs	Baseline $\widehat{K}$	Intervention $\widehat{K}$	Baseline all / new	Intervention all / new	$\Delta\lvert e_K\rvert$
SimGCD	CIFAR100	4	100.2	138.8	0.7130 / 0.3208	0.8429 / 0.7582	+24.0
SelEx	CIFAR100	7	103.0	130.7	0.7719 / 0.3683	0.8425 / 0.6337	+15.4
SimGCD	ImageNet100	4	55.5	133.5	0.4656 / 0.0215	0.8306 / 0.7526	-11.0
SelEx	ImageNet100	7	87.0	107.3	0.6920 / 0.4375	0.8100 / 0.6696	-6.9

Table 17: CIFAR100 discovery-pool sampling with frozen features and $K^\star = 100$. Entries are mean $\pm$ sample standard deviation over five construction seeds. Rel. error is $|\widehat{K} - K^\star|/K^\star$; Rare is rare-class recall.

Source	Nominal IF	Realized ratio	$\widehat{K}$	Rel. error	All	Old	New	Rare
SimGCD	10	20.1 ± 0.9	125.8 ± 3.6	0.2580 ± 0.0363	0.8659 ± 0.0123	0.8977 ± 0.0052	0.6544 ± 0.0665	0.8560 ± 0.0669
SimGCD	50	115.7 ± 8.4	124.4 ± 3.9	0.2440 ± 0.0391	0.8984 ± 0.0067	0.9207 ± 0.0103	0.7017 ± 0.0905	0.4960 ± 0.0727
SimGCD	100	228.3 ± 19.3	124.4 ± 2.9	0.2440 ± 0.0288	0.9110 ± 0.0099	0.9301 ± 0.0098	0.7264 ± 0.1012	0.2880 ± 0.1035
SelEx	10	20.1 ± 0.9	125.4 ± 7.1	0.2540 ± 0.0709	0.8922 ± 0.0138	0.9401 ± 0.0038	0.6029 ± 0.0685	0.8880 ± 0.0179
SelEx	50	115.7 ± 8.4	130.0 ± 5.2	0.3000 ± 0.0520	0.9232 ± 0.0102	0.9525 ± 0.0081	0.6705 ± 0.0656	0.5840 ± 0.0607
SelEx	100	228.3 ± 19.3	132.4 ± 7.5	0.3240 ± 0.0754	0.9327 ± 0.0096	0.9578 ± 0.0052	0.7131 ± 0.0594	0.4560 ± 0.1513
AFGCD	10	20.1 ± 0.9	111.0 ± 6.2	0.1100 ± 0.0620	0.8577 ± 0.0107	0.9048 ± 0.0094	0.5456 ± 0.0643	0.9120 ± 0.0438
AFGCD	50	115.7 ± 8.4	107.4 ± 4.3	0.0740 ± 0.0428	0.8955 ± 0.0098	0.9244 ± 0.0107	0.6394 ± 0.1061	0.6000 ± 0.0849
AFGCD	100	228.3 ± 19.3	107.6 ± 3.9	0.0760 ± 0.0391	0.9085 ± 0.0082	0.9319 ± 0.0084	0.6849 ± 0.0718	0.4000 ± 0.1356
DebGCD	10	20.1 ± 0.9	93.2 ± 1.9	0.0680 ± 0.0192	0.8718 ± 0.0109	0.9133 ± 0.0030	0.5937 ± 0.0563	0.9280 ± 0.0522
DebGCD	50	115.7 ± 8.4	96.6 ± 5.2	0.0540 ± 0.0230	0.9116 ± 0.0068	0.9351 ± 0.0044	0.6987 ± 0.0501	0.6240 ± 0.0456
DebGCD	100	228.3 ± 19.3	93.0 ± 4.3	0.0700 ± 0.0430	0.9143 ± 0.0103	0.9399 ± 0.0051	0.6576 ± 0.0959	0.4320 ± 0.0867
RLCD	10	20.1 ± 0.9	151.0 ± 10.3	0.5100 ± 0.1030	0.9038 ± 0.0124	0.9304 ± 0.0064	0.7232 ± 0.0970	0.9440 ± 0.0358
RLCD	50	115.7 ± 8.4	163.8 ± 3.1	0.6380 ± 0.0311	0.9391 ± 0.0044	0.9514 ± 0.0068	0.8260 ± 0.0249	0.5600 ± 0.0566
RLCD	100	228.3 ± 19.3	158.8 ± 7.9	0.5880 ± 0.0795	0.9434 ± 0.0023	0.9569 ± 0.0049	0.8059 ± 0.0334	0.4400 ± 0.0849

M Model Order and Partition Quality on Fixed Merge Paths

Figure 7 summarizes the representative-partition comparison in the main text. The following diagnostics vary the cut on fixed feature banks and seed-specific merge paths.

M.1 Accuracy Landscapes across Eight Settings

Figure 8 covers all eight SimGCD/SelEx settings. We evaluate every common cut for CUB, SCars, and Aircraft, and a uniform grid augmented with critical counts for Herbarium-19 (103 cuts for SimGCD; 102 for SelEx). Curves show means and sample standard deviations across overclustering seeds $\{0, 1, 2\}$ on fixed features; the main tables instead report partitions chosen by the observable representative-seed rule. The public SelEx Herbarium-19 estimate $341 = K_{\text{seen}}$ lies below the path minimum, so Figure 4 starts at the nearest feasible cut, 342.

For novel samples, let P count unordered predicted-same pairs, T true-same pairs, and J pairs satisfying both conditions. False-merge and false-split rates are $(P - J)/P$ and $(T - J)/T$, respectively. Novel-class labels and these evaluation metrics do not enter cut selection.

M.2 CUB: Partition Structure and Category Recoverability

On the SelEx CUB seed-2 path, 177 of the 200 parent clusters remain unchanged; the other 23 have two, three, four, or five children at $K = 237$ (13, 7, 2, and 1 parents, respectively). Twenty split parents have children with different dominant labels. Cross-partition NMI and ARI are 0.9765 and 0.8119.

We measure category recoverability using one-to-one Hungarian matching on the full feature bank, maximizing matched sample count. A class G is recoverable if its matched cluster C reaches $F_1(C, G) = 2|C \cap G|/(|C| + |G|) \geq t$, for $t \in \{0.5, 0.8\}$. Table 27 reports these evaluation-only counts alongside accuracy; there are 100 seen and 100 novel classes.

Table 18: SEE-K with its fixed default profile across three additional GCD feature sources. All/Old/New use the selected cut. $K^\star$ All uses the same-path exact-count cut and is a diagnostic, not an accuracy upper bound. Both cuts use the observable representative-seed rule.

Source	Dataset	$K^\star$	$\widehat{K}$	Rel. error	All	Old	New	$K^\star$ All
AFGCD	CUB	200	218	0.0900	0.6788	0.8402	0.5649	0.6505
AFGCD	SCars	196	215	0.0969	0.6556	0.8848	0.4835	0.6175
AFGCD	Aircraft	100	114	0.1400	0.6514	0.7995	0.5073	0.6304
AFGCD	Herbarium-19	683	375	0.4510	0.4854	0.7240	0.3304	0.5365
RLCD	CUB	200	207	0.0350	0.7142	0.8872	0.5856	0.7065
RLCD	SCars	196	231	0.1786	0.6747	0.8343	0.5337	0.6185
RLCD	Aircraft	100	74	0.2600	0.5461	0.7500	0.3598	0.6178
RLCD	Herbarium-19	683	496	0.2738	0.5104	0.8105	0.3290	0.5472
DebGCD	CUB	200	225	0.1250	0.6742	0.8055	0.5606	0.6273
DebGCD	SCars	196	244	0.2449	0.6596	0.8300	0.5062	0.5948
DebGCD	Aircraft	100	124	0.2400	0.6424	0.7725	0.5322	0.5520
DebGCD	Herbarium-19	683	421	0.3836	0.5158	0.7397	0.3511	0.5364

Table 19: Recent open-source baselines whose public repositories expose test-side evaluation metrics. These rows are easier to interpret than the train-side SS-kmeans tables below, but they still follow each repository's default evaluation protocol rather than our fixed-feature selector setting.

Method	Dataset	Acc. all	Acc. old	Acc. new	Status
DebGCD	CUB	0.6540	0.7129	0.5955	public test-side metric
DebGCD	SCars	0.6356	0.7888	0.4879	public test-side metric
DebGCD	Aircraft	0.5665	0.6031	0.5297	public test-side metric
DebGCD	CIFAR100	0.8207	0.8264	0.7980	interpretable reproduced metric
DebGCD	ImageNet100	0.0100	0.0200	0.0000	near-zero reproduced metric
ReLKD	SCars	0.5487	0.7105	0.3926	public test-side metric
ReLKD	Aircraft	0.5329	0.5797	0.4859	public test-side metric
ReLKD	CIFAR100	0.8039	0.8105	0.7775	interpretable reproduced metric

N Extended Conclusions, Limitations, and Broader Impact

The main paper gives a conclusion-first summary. This appendix expands the discussion and limitations behind that conclusion: the proposed selector is best understood as a diagnosable model-order decision layer rather than as a universal accuracy maximizer.

Additional conclusion. Fixed-path diagnostics distinguish count fidelity from assignment quality: correcting severe underestimation improves assignments on SelEx Herbarium-19, while a finer-than-$K^\star$ cut improves novel assignments on CUB.

Scope limitations. The main evaluation uses SimGCD and SelEx on four fine-grained datasets, with transfer checks on AFGCD, RLCD, and DebGCD and broader-domain audits on CIFAR100 and ImageNet100. This scope is sufficient to demonstrate that unknown-K selection can be isolated and diagnosed, but it does not establish that one fixed selector profile is optimal across domains. The current risks are observable surrogates, not a guaranteed estimator of the semantic class count, and the candidate family can only select counts along the constructed overclustering-to-merge path.

Calibration and future work. LODO calibration improves held-out count estimation, with an accuracy trade-off (Appendix E). Future work includes stronger candidate families, consensus partitions across seeds, and representations better suited to model-order selection.

Broader impact. Explicit unknown-K diagnostics can improve transparency in open-world discovery systems by revealing when a model is over-merging, over-fragmenting, or behaving unstably across seeds. The same diagnostics also make clear that the selector is not a complete safety mechanism: wrong class-count decisions can still merge biologically or semantically distinct classes, split

Table 20: Recent open-source baselines whose public repositories primarily expose train-side SS-kmeans metrics. We report them as contextual evidence only, not as direct rows for the main comparison table.

Method	Dataset	Acc. all	Acc. old	Acc. new	Status
AFGCD	CUB	0.6479	0.7051	0.6193	public train-side metric
AFGCD	SCars	0.6226	0.7961	0.5387	public train-side metric
AFGCD	Aircraft	0.5585	0.6477	0.5139	public train-side metric
AFGCD	Herbarium-19	0.4348	0.5696	0.3624	public train-side metric
AFGCD	CIFAR100	0.8153	0.8577	0.7305	public train-side metric
AFGCD	ImageNet100	0.8326	0.9452	0.7760	train-side metric only
RLCD	CUB	0.6828	0.7785	0.6350	public train-side metric
RLCD	SCars	0.6113	0.7441	0.5472	public train-side metric
RLCD	Aircraft	0.5559	0.5840	0.5418	public train-side metric
RLCD	Herbarium-19	0.4561	0.6106	0.3730	public train-side metric
RLCD	CIFAR100	0.8282	0.8298	0.8251	interpretable reproduced metric
RLCD	ImageNet100	0.0217	0.0241	0.0205	substantial reproduced degradation

Table 21: SelEx end-to-end multi-seed repeated runs on Herbarium-19. We report mean $\pm$ standard deviation across eight runs, including the initial run and seven independently seeded repeated runs.

Dataset	Runs	Baseline all	Baseline new	SEE-K $\widehat{K}$ mean $\pm$ std	SEE-K all	SEE-K new
Herbarium-19	8	0.3792 ± 0.0060	0.3125 ± 0.0073	649.0 ± 43.1 [571, 711]	0.5450 ± 0.0094	0.3758 ± 0.0152

one class into many fragments, or distort downstream retrieval and monitoring analyses. We therefore recommend reporting diagnostic traces and stress tests alongside final clustering metrics.

N.1 Extended Experimental Setup

Why SimGCD and SelEx. We choose these two backbones to provide complementary fixed-feature test cases. SimGCD is a widely used parametric GCD baseline and provides a standard reference point for testing whether the decision layer applies beyond one specialized method. SelEx is a recent fine-grained GCD system with strong reported results and an explicit public unknown-K estimate, which makes class-count estimation directly auditable. Together they form a useful evaluation pair: SimGCD tests the selector on a canonical backbone, while SelEx tests it on a high-performing fine-grained system where the remaining failures are less likely to be explained by representation limitations alone. Their public codebases also let us generate compatible checkpoints and feature banks across the four main datasets, so the same decision-layer selector can be applied without changing the representation learner.

Compute. The main SimGCD/SelEx experiments used a Linux server with eight NVIDIA RTX 4090 GPUs (24 GB each), two AMD EPYC 7763 CPUs, and 503 GiB RAM. Training and feature extraction for the additional AFGCD, RLCD, and DebGCD sources used a separate RTX 3090 server (24 GB per GPU). After feature extraction, selector evaluation, LODO calibration, external-criterion comparisons, and long-tail sampling ran on CPUs using frozen feature banks, without backbone retraining. For the main SimGCD/SelEx runs, feature-bank construction and default selection took approximately 20–70 minutes per checkpoint on CUB, SCars, and Aircraft, and 13–21 hours on Herbarium-19. These timings include feature extraction and seed-specific statistics; cached-profile evaluation is substantially cheaper. They describe the main selector pipeline, not the total cost of all training and supplementary experiments.

Assets and licenses. Table 28 lists the external datasets and code assets used in the paper. We use all datasets only for academic benchmarking under their original access terms, and we do not redistribute raw datasets, third-party code, or checkpoints/feature banks derived from restricted datasets.

Table 22: Formal controlled SimGCD end-to-end multi-seed repeated runs on Herbarium-19. We report mean ± standard deviation across eight explicitly seeded repeated runs.

Dataset	Runs	Baseline all	Baseline new	SEE-K $\widehat{K}$ mean ± std	SEE-K all	SEE-K new
Herbarium-19	8	0.4364 ± 0.0057	0.3634 ± 0.0115	389.8 ± 5.2 [382, 399]	0.5033 ± 0.0046	0.3611 ± 0.0117

Table 23: SelEx end-to-end multi-seed repeated runs on the broader-domain scope-extension settings. The first seed reuses the previously reported broader-domain run, and the additional seeds use independently seeded repeated executions.

Dataset	Runs	Baseline all	Baseline new	SEE-K $\widehat{K}$ mean ± std	SEE-K all	SEE-K new
CIFAR100	4	0.7801 ± 0.0194	0.6621 ± 0.0320	116.8 ± 42.9 [81, 178]	0.7775 ± 0.0509	0.4009 ± 0.2335
ImageNet100	3	0.7822 ± 0.0128	0.7107 ± 0.0149	87.0 ± 29.6 [54, 111]	0.6692 ± 0.1775	0.4071 ± 0.3336

Table 24: Additional controlled multi-seed repeated runs on the fine-grained settings beyond the original Herbarium-19 and CUB audits. We report mean ± standard deviation across explicitly seeded runs.

Backbone	Dataset	Runs	Baseline all	Baseline new	SEE-K $\widehat{K}$ mean ± std	SEE-K all	SEE-K new
SimGCD	CUB	8	0.6126 ± 0.0085	0.5916 ± 0.0119	214.5 ± 12.1 [193, 231]	0.6660 ± 0.0202	0.5430 ± 0.0350
SimGCD	SCars	8	0.5368 ± 0.0367	0.4515 ± 0.0409	229.5 ± 4.6 [223, 235]	0.6100 ± 0.0292	0.4431 ± 0.0441
SimGCD	Aircraft	8	0.5350 ± 0.0235	0.5061 ± 0.0316	88.3 ± 21.1 [70, 121]	0.5271 ± 0.0711	0.3190 ± 0.1282
SelEx	SCars	8	0.5645 ± 0.0098	0.4656 ± 0.0128	235.8 ± 14.7 [214, 257]	0.6249 ± 0.0136	0.4254 ± 0.0189
SelEx	Aircraft	8	0.5854 ± 0.0151	0.5476 ± 0.0218	101.1 ± 11.3 [86, 117]	0.5987 ± 0.0362	0.4267 ± 0.0603

Table 25: Formal paired tests for supplementary multi-seed repeated runs. We report mean differences as ours minus baseline. Holm-adjusted p-values are computed over the metric-level tests in this table. The tests are still small-sample diagnostics, especially for exact non-parametric tests, not a complete significance analysis over all main settings.

Backbone	Dataset	Runs	Δ all	paired t p	Holm p	Wilcoxon p	Wilcoxon Holm	Δ new	paired t p	Holm p	Wilcoxon p	Wilcoxon Holm
SelEx	Herbarium-19	8	+0.1658	$< 10^{-6}$	$< 10^{-6}$	0.0078	0.1562	+0.0633	0.000026	0.000442	0.0078	0.1562
SelEx	SCars	8	+0.0605	0.000029	0.000464	0.0078	0.1562	-0.0402	0.002175	0.023925	0.0078	0.1562
SelEx	Aircraft	8	+0.0133	0.325255	1.0000	0.3125	1.0000	-0.1209	0.000801	0.009612	0.0078	0.1562
SelEx	CUB	8	-0.0852	0.000588	0.007644	0.0078	0.1562	-0.2117	0.000226	0.003164	0.0078	0.1562
SelEx	CIFAR100	4	-0.0026	0.932752	1.0000	0.8750	1.0000	-0.2612	0.120456	0.883592	0.1250	0.8750
SelEx	ImageNet100	3	-0.1130	0.408376	1.0000	0.5000	1.0000	-0.3036	0.267113	1.0000	0.2500	1.0000
SimGCD	Herbarium-19	8	+0.0669	$< 10^{-6}$	$< 10^{-6}$	0.0078	0.1562	-0.0023	0.616747	1.0000	0.6406	1.0000
SimGCD	CUB	8	+0.0534	0.000094	0.001410	0.0078	0.1562	-0.0486	0.003815	0.038150	0.0078	0.1562
SimGCD	SCars	8	+0.0733	$< 10^{-6}$	$< 10^{-6}$	0.0078	0.1562	-0.0084	0.110449	0.883592	0.1094	0.8750
SimGCD	Aircraft	8	-0.0080	0.754571	1.0000	0.8438	1.0000	-0.1871	0.004174	0.038150	0.0078	0.1562

Table 26: Bootstrap percentile intervals for the same multi-seed repeated runs. Each interval is computed from 10,000 bootstrap resamples of the run-level means.

Backbone	Dataset	Runs	Δ all 95% CI	Δ new 95% CI	$\widehat{K}$ 95% CI	SEE-K all 95% CI
SelEx	Herbarium-19	8	[0.1576, 0.1733]	[0.0514, 0.0749]	[619.8, 675.8]	[0.5387, 0.5509]
SelEx	SCars	8	[0.0493, 0.0725]	[-0.0554, -0.0242]	[226.5, 245.2]	[0.6165, 0.6341]
SelEx	Aircraft	8	[-0.0117, 0.0347]	[-0.1632, -0.0843]	[93.8, 108.5]	[0.5722, 0.6194]
SelEx	CUB	8	[-0.1116, -0.0586]	[-0.2694, -0.1570]	[197.0, 228.2]	[0.6582, 0.7110]
SelEx	CIFAR100	4	[-0.0583, 0.0404]	[-0.4965, -0.0656]	[88.0, 157.2]	[0.7351, 0.8180]
SelEx	ImageNet100	3	[-0.3280, 0.0240]	[-0.6958, -0.0460]	[54.0, 111.0]	[0.4680, 0.8037]
SimGCD	Herbarium-19	8	[0.0646, 0.0692]	[-0.0100, 0.0059]	[386.6, 393.1]	[0.5004, 0.5063]
SimGCD	CUB	8	[0.0409, 0.0648]	[-0.0695, -0.0282]	[206.5, 221.8]	[0.6533, 0.6786]
SimGCD	SCars	8	[0.0654, 0.0805]	[-0.0171, 0.0001]	[226.5, 232.4]	[0.5909, 0.6287]
SimGCD	Aircraft	8	[-0.0502, 0.0384]	[-0.2638, -0.1011]	[75.7, 102.8]	[0.4848, 0.5766]

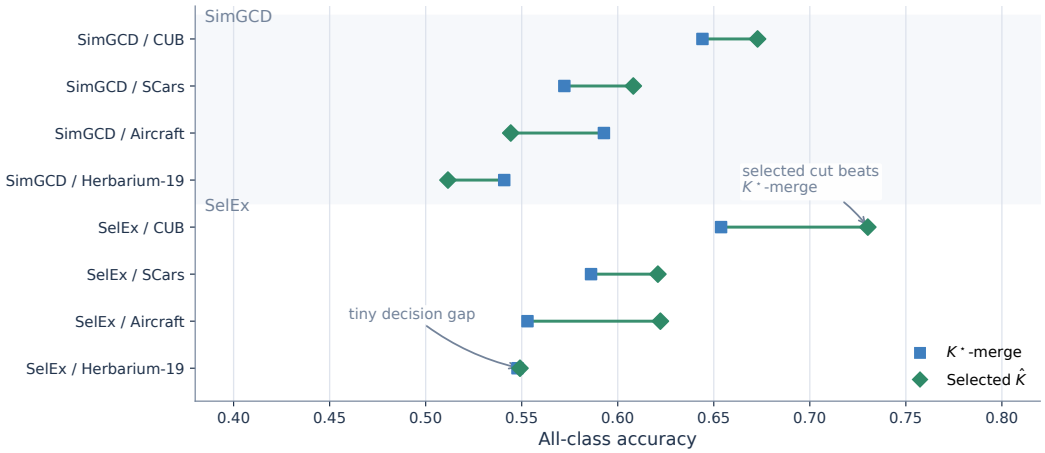

Figure 7: $K^\star$-cut bridge on the same merge path for the eight main fine-grained settings. Each row compares the $K^\star$-merge cut fixed to the ground-truth class count with the selected cut returned by our model-order decision layer. SelEx Herbarium-19 shows a near-zero accuracy difference between the two cuts, whereas SelEx CUB has higher accuracy at the selected cut than at $K^\star$.

Table 27: **SelEx CUB partition and recoverability diagnostics on seed 2.** Recoverability counts classes meeting the matched-F1 threshold.

Diagnostic	$K = 200$	$K = 237$
All accuracy	0.6538	0.7302
Old accuracy	0.8789	0.8789
New accuracy	0.5098	0.6523
Recoverable seen, $F_1 \geq 0.5$	96	96
Recoverable novel, $F_1 \geq 0.5$	39	62
Recoverable seen, $F_1 \geq 0.8$	79	79
Recoverable novel, $F_1 \geq 0.8$	21	35

Table 28: Existing assets, credits, and license or usage terms. "Not separately specified" means that our audit found no separate license name on the public source used for the experiments; in these cases we cite the original asset and follow the source's stated access conditions.

Asset	Source and credit	License or terms stated by source	Use in this paper
CUB-200-2011	Caltech-UCSD Birds-200-2011 [23]; official page	No separate SPDX-style license name. The official page states that Caltech does not own the image copyrights and that use is restricted to non-commercial research and educational purposes.	Fine-grained benchmark data only; no redistribution.
Stanford Cars / SCars	Stanford Cars [24]; download pointer from the SSB repository	Not separately specified on the public source available to us. We use the dataset as a standard academic benchmark and cite the original paper and SSB split source.	Fine-grained benchmark data only; no redistribution.
FGVC-Aircraft	FGVC-Aircraft [25]; official page	Images are made available for non-commercial research purposes only; original photographers retain copyright.	Fine-grained benchmark data only; no redistribution.
Herbarium-19	Herbarium Challenge 2019 [26]; Kaggle FGVC6 competition page	Kaggle competition data subject to the competition/data access terms.	Fine-grained benchmark data only; no redistribution.
CIFAR100	CIFAR-100 [27]; official page; accessed through torchvision.	No separate license name is stated on the official CIFAR page; users are asked to cite the technical report.	Broader-domain diagnostic benchmark only.
ImageNet100	ImageNet / ILSVRC [28]; access agreement	ImageNet access is restricted to non-commercial research and educational purposes under the ImageNet access agreement.	Broader-domain diagnostic benchmark subset only; no redistribution.
SimGCD and SelEx codebases	SimGCD [2], repository; SelEx [3], repository	MIT License in the public repositories.	Public backbone training/evaluation code; local checkpoints and feature banks generated for experiments.
Additional public GCD baselines	AFGCD [18], RLCD [19], DebGCD [20], and ReLKD [21] public repositories.	DebGCD includes an MIT license in our local clone. We did not find explicit license files in the local AFGCD, RLCD, or ReLKD clones, so those repositories are used only for local reproduction and are not redistributed.	Supplementary reproduction context only.
Generated checkpoints, feature banks, tables, and figures	Generated by this work from the assets above.	Derived experimental artifacts follow the upstream dataset/code access constraints; no third-party raw data or third-party code is redistributed with the paper.	Reported metrics and paper figures.

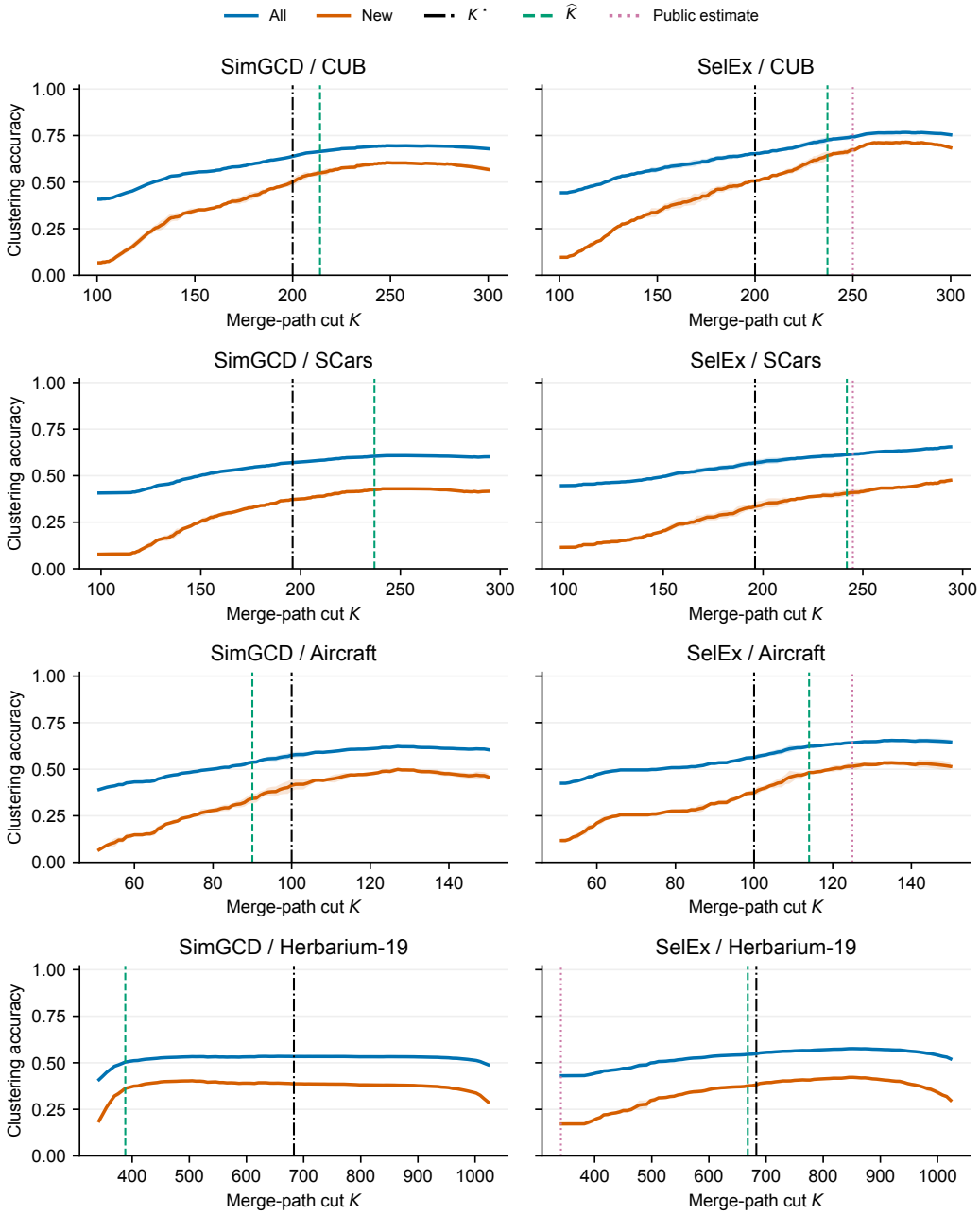

Figure 8: **Accuracy landscapes across eight fine-grained settings.** All/New accuracy: mean and one sample standard deviation across three paths. Vertical lines mark $K^\star$, $\widehat{K}$, and public estimates. Herbarium-19 uses sampled cuts; the other datasets include every common cut.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. The abstract and introduction explicitly state the paper's narrower claim: structured unknown-K selection and a decision-layer evaluation protocol rather than universal accuracy dominance; see the abstract, Section 4, and Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes. Section 6 discusses the limited empirical scope, the fixed default selector profile, the surrogate nature of the observable risks, the restriction to the constructed overclustering-to-merge path, and the diagnostic status of the broader-domain intervention.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: N/A. The paper is empirical and does not present new theorems, lemmas, or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes. The paper specifies the fixed-backbone evaluation protocol, the selector design, the evaluation metrics, the robustness protocol, and the experimental defaults in Sections 3–5.3 and in Appendices B, L, and M.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes. Code is publicly available at `https://github.com/MingfuYAN/SEE-K`, including environment specifications, data access and preparation instructions, and scripts for backbone training, feature extraction, and the main experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes. Sections 5.1, 5.2, and 5.3, together with Appendices B, L, and N.1, describe the datasets, backbone reuse protocol, search hyperparameters, stress-test settings, selector defaults, and the additional rerun statistics used for analysis.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: No. The appendix reports paired tests, Holm-corrected p-values, and bootstrap confidence intervals for the audited multi-seed reruns that support the robustness analysis; see Tables 25 and 26. However, the main comparison tables remain point-estimate summaries and do not provide uniform error bars or formal significance tests for every reported row.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes. The appendix section "Extended Experimental Setup" reports the hardware used for the main and additional feature sources, the GPU/CPU division of work, and approximate main-selector runtimes with their scope.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes. The work is an empirical study on existing datasets and public backbone families and does not involve human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes. Section 6 discusses both the positive impact of improved transparency in open-world discovery systems and the negative risk of wrong class-count decisions merging or fragmenting semantically meaningful classes.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: N/A. The paper does not release high-risk generative models, scraped datasets, or similar assets with elevated misuse risk; it studies a decision-layer clustering selector on standard vision benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes. Appendix Table 28 credits the reused datasets and public codebases, lists the license or stated access terms when available, and states that raw datasets, third-party code, and checkpoints or feature banks derived from restricted datasets are not redistributed.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Yes. The SEE-K repository (`https://github.com/MingfuYAN/SEE-K`) provides installation and usage instructions in its README and includes an MIT license.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: N/A. The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: N/A. The paper does not involve crowdsourcing or human-subject research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: N/A. The core method and experiments in this research do not use LLMs as an original or non-standard methodological component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.