Appendix A

1) Schematic: The schematic of the formulated OpenGCD is shown in Fig. 5, where (i) and (j) are flowcharts of the proposed solution and (a)-(h) are descriptions of each component.

2) Algorithm: Algorithm 1 provides the procedure for OpenGCD. Algorithms 2-4 provide the procedure of each component in OpenGCD.

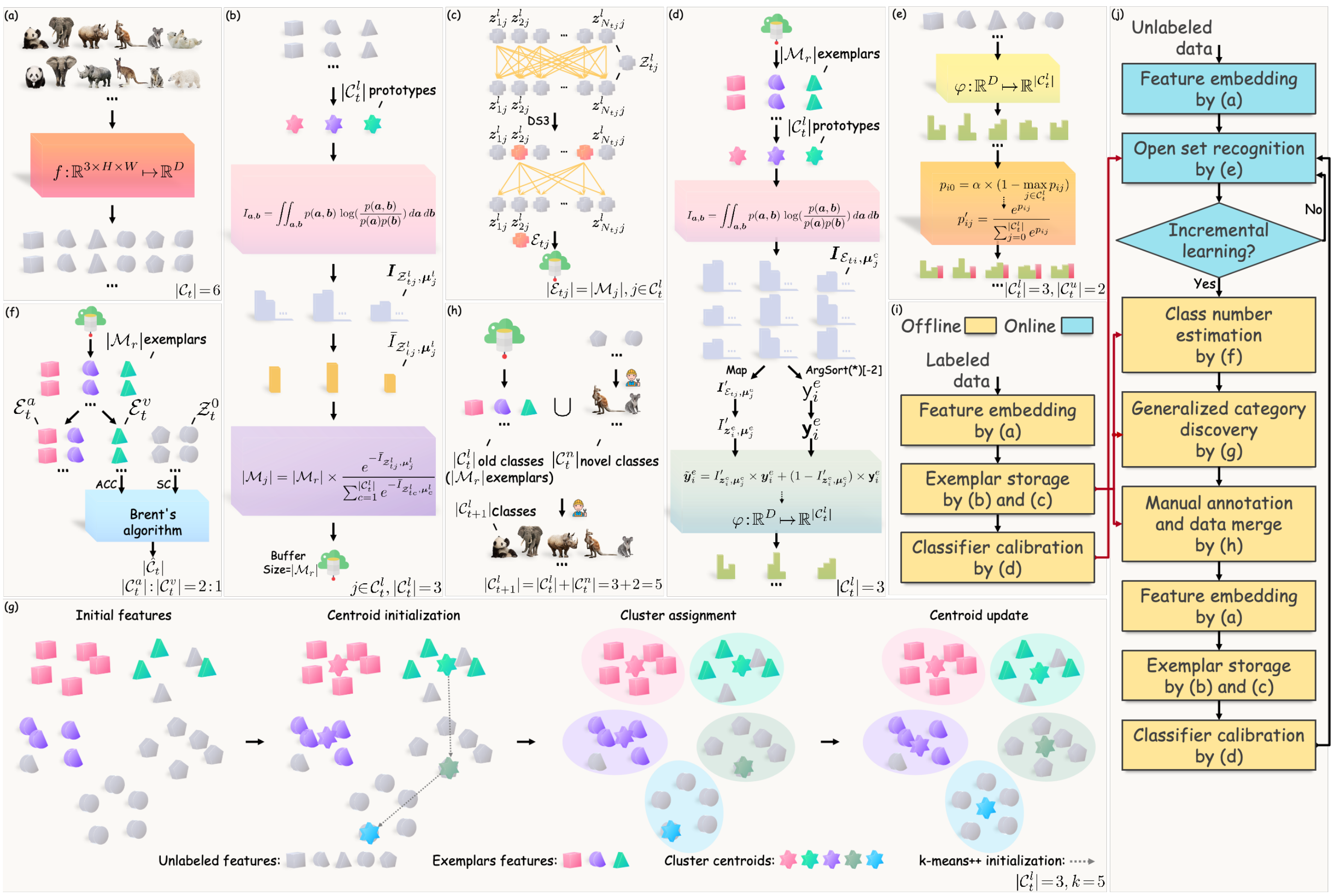

Fig. 5. Schematic of OpenGCD. (a) Feature embedding. No matter online or offline instance, its feature embedding should be obtained by the frozen feature extractor first (details c.f. Sec. III-B of the main paper). (b) MI-based memory allocation. Allocate a memory budget of size $|\mathcal{M}_j|$ for known class j ($j \in \mathcal{C}_t^l$) based on MI. (details c.f. Sec. III-C1). (c) Exemplar selection. For the known class j ($j \in \mathcal{C}_t^l$), $|\mathcal{M}_j|$ exemplars are selected by the DS3 algorithm to represent the original feature subset $\mathcal{Z}_{tj}^l$ (details c.f. Sec. III-C2). (d) Classifier calibration. The soft labels for (re)fitting the classifier are assigned to $|\mathcal{M}_r|$ exemplars based on MI (details c.f. Sec. III-D). (e) Open set recognition. For an online feature, the uncertainty of the closed set output of the classifier serves as an estimate for the unknown (details c.f. Sec. III-E). (f) Class number estimate. The clustering results of labeled $\mathcal{E}_t^v$ and unlabeled $\mathcal{Z}_t^0$ are evaluated with ACC and SC respectively and the optimal class number $|\hat{\mathcal{C}}_t|$ is determined by Brent's algorithm (details c.f. Sec. III-F3). (g) ss-k-means++ algorithm for $k=5$. The labeled exemplar set $\mathcal{E}_t$ and the rejected unlabeled feature set $\mathcal{Z}_t^0$ lie in the same feature space (Initial features). The centroids of $|\mathcal{C}_t^l|$ known classes are derived from labeled exemplars, and the centroids of the remaining $k-|\mathcal{C}_t^l|$ novel classes are initialized by k-means++ (Centroids initialization). For unlabeled features, clustering labels are assigned by identifying the nearest centroid (Cluster assignment). Centroids are updated by averaging the features in each cluster (Centroid update). Cluster assignment and Centroid update are then repeated until convergence, during which labeled exemplars are forced to follow their ground-truth labels (details c.f. Sec. III-F1). (h) Manual annotation and data merge. The labeler fetches the instance set $\hat{\mathcal{X}}_t^n$ and revises the label. Then, pick up the labeled instance set $\mathcal{X}_t^e$ and merge it with $\hat{\mathcal{X}}_t^n$ to get $\mathcal{X}_{t+1}^l$ (details c.f. Secs. III-F2). (i) Offline modeling procedure. Pre-stored exemplars and calibrated classifier are prepared for subsequent procedures. (j) OpenGCD procedure. Online instances can be input separately or in batches. Once a phase of OSL is completed, GCD and IL can be launched.

Algorithm 1: Procedure of OpenGCD.

Input: Labeled data: $\mathcal{X}_0^l = \{\boldsymbol{x}_i^l, y_i^l\}_{i=1}^{N_0}$; unlabeled data: $\left\{\mathcal{X}_t^u = \{\boldsymbol{x}_i^u\}_{i=1}^{M_t}\right\}_{t=0}^{T-1}$; buffer: $\mathcal{M}_r$; feature extractor: f; classifier: φ; regulatory factor: α; Maximum total number of classes: $|\mathcal{C}_t^{\max}|$.

for t *in range*(T) **do**
    Get labeled feature embeddings $\mathcal{Z}_t^l = \{\boldsymbol{z}_i^l, y_i^l\}_{i=1}^{N_t}$ via $f(\mathcal{X}_t^l)$
    $\mathcal{M}_r = \texttt{ExemplarStorage}(\mathcal{Z}_t^l, \mathcal{M}_r)$
    Get unlabeled feature embeddings $\mathcal{Z}_t^u = \{\boldsymbol{z}_i^u\}_{i=1}^{M_t}$ via $f(\mathcal{X}_t^u)$
    $\mathcal{Z}_t^0 = \texttt{ClassifierCalibrationOSR}(\mathcal{Z}_t^u, \mathcal{M}_r, \varphi, \alpha)$
    $\mathcal{X}_t^n = \texttt{AssistingManualAnnotationGCD}(\mathcal{Z}_t^0, \mathcal{M}_r, |\mathcal{C}_t^{\max}|)$
    Get labeled exemplars $\mathcal{E}_t = \{\boldsymbol{z}_i^e, y_i^e\}_{i=1}^{N_0}$ from $\mathcal{M}_r$
    Pick the instance set $\mathcal{X}_t^e$ corresponding to $\mathcal{E}_t$
    Get new labeled instance set $\mathcal{X}_{t+1}^l = \{\boldsymbol{x}_i^l, y_i^l\}_{i=1}^{N_{t+1}}$ by concatenating $\mathcal{X}_t^e$ with $\mathcal{X}_t^n$
end

Algorithm 2: Exemplar Storage Procedure in OpenGCD.

Input: Labeled feature embeddings: $\mathcal{Z}_t^l = \{\boldsymbol{z}_i^l, y_i^l\}_{i=1}^{N_t}$; buffer: $\mathcal{M}_r$.
Output: Buffer filled with examples: $\mathcal{M}_r$.

Function `ExemplarStorage` $(\mathcal{Z}_t^l, \mathcal{M}_r)$**:**
    Determine known classes $\mathcal{C}_t^l$ in $\mathcal{Z}_t^l$
    for j *in* $\mathcal{C}_t^l$ **do**
        Get the feature $\mathcal{Z}_{tj}^l$ of the j^{th} class from $\mathcal{Z}_t^l$;
        Calculate the prototype $\boldsymbol{\mu}_j^l$ of the j^{th} class via $\frac{1}{|\mathcal{Z}_{tj}^l|}\sum_{y_i^l=j} \boldsymbol{z}_i^l$
        Get the MI vector $\boldsymbol{I}_{\mathcal{Z}_{tj}^l, \boldsymbol{\mu}_j^l}$ between $\mathcal{Z}_{tj}^l$ and $\boldsymbol{\mu}_j^l$ via Eq. 1
        Calculate the mean $\bar{I}_{\mathcal{Z}_{tj}^l, \boldsymbol{\mu}_j^l}$ of $\boldsymbol{I}_{\mathcal{Z}_{tj}^l, \boldsymbol{\mu}_j^l}$
    end
    for j *in* $\mathcal{C}_t^l$ **do**
        Allocate memory $\mathcal{M}_j$ for the j^{th} class via Eq. 2
        $\mathcal{M}_j = \varnothing$
        Get the feature $\mathcal{Z}_{tj}^l$ of the j^{th} class from $\mathcal{Z}_t^l$
        Select $|\mathcal{M}_j|$ exemplars $\mathcal{E}_{tj}$ by `DS3` $(\mathcal{Z}_{tj}^l)$ for the j^{th} class
        Put $\mathcal{E}_{tj}$ into $\mathcal{M}_j$
    end
    $\mathcal{M}_r = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \cdots \cup \mathcal{M}_{|\mathcal{C}_t^l|}$
    return $\mathcal{M}_r$

Algorithm 3: Classifier Calibration and OSR Procedure in OpenGCD.

Input: Unlabeled feature embeddings: $\mathcal{Z}_t^u = \{\boldsymbol{z}_i^u\}_{i=1}^{M_t}$; buffer: $\mathcal{M}_r$; classifier: φ; regulatory factor: α.
Output: Rejected unlabeled feature embeddings: $\mathcal{Z}_t^0$.

Function `ClassifierCalibrationOSR` $(\mathcal{Z}_t^u, \mathcal{M}_r, \varphi, \alpha)$**:**
  Initialize classifier φ
  for j, $\mathcal{M}_j$ *in enumerate*$(\mathcal{M}_r)$ **do**
    Get the exemplar $\mathcal{E}_{tj}$ of the j^{th} class from $\mathcal{M}_j$
    Calculate the prototype $\boldsymbol{\mu}_j^e$ of the j^{th} class via $\frac{1}{|\mathcal{E}_{tj}|}\sum_{y_i^e=j}\boldsymbol{z}_i^e$
  end
  Get labeled exemplars $\mathcal{E}_t = \{\boldsymbol{z}_i^e, y_i^e\}_{i=1}^{N_0}$ from $\mathcal{M}_r$
  Get the MI matrix $\boldsymbol{I}_{\mathcal{E}_t}$ between $\mathcal{E}_t$ and $\boldsymbol{\mu}_j^e, j = 1, 2, \cdots, |\mathcal{C}_t^l|$ via Eq. 1
  Determine known classes $\mathcal{C}_t^l$ in $\mathcal{E}_t$
  for j *in* $\mathcal{C}_t^l$ **do**
    Get the MI submatrix $\boldsymbol{I}_{\mathcal{E}_{tj}}$ related to $\mathcal{E}_{tj}$ from $\boldsymbol{I}_{\mathcal{E}_t}$
    Get the MI vector $\boldsymbol{I}_{\mathcal{E}_{tj},\boldsymbol{\mu}_j^e}$ related to $\boldsymbol{\mu}_j^e$ from $\boldsymbol{I}_{\mathcal{E}_{tj}}$
    Form the mapped MI vector $\boldsymbol{I}'_{\mathcal{E}_{tj},\boldsymbol{\mu}_j^e}$ via mapping $\boldsymbol{I}_{\mathcal{E}_{tj},\boldsymbol{\mu}_j^e}$ to $(0.5, 1]$
    for $\boldsymbol{I}_{\boldsymbol{z}_i^e}$ *in* $\boldsymbol{I}_{\mathcal{E}_{tj}}$ **do**
      Determine the class y_i^e corresponding to the remaining maximum MI by masking the j^{th} column of $\boldsymbol{I}_{\boldsymbol{z}_i^e}$
      Get the labels $\boldsymbol{y}_i^e$ and $\mathbf{y}_i^e$ in one-hot form for the ground-truth class y_i^e and the nearest neighbor class y_i^e of $\boldsymbol{z}_i^e$
      Get the mapped MI $I'_{\boldsymbol{z}_i^e,\boldsymbol{\mu}_j^e}$ between $\boldsymbol{z}_i^e$ and $\boldsymbol{\mu}_j^e$ from $\boldsymbol{I}'_{\mathcal{E}_{tj},\boldsymbol{\mu}_j^e}$
      Get the softened label $\tilde{\boldsymbol{y}}_i^e$ of $\boldsymbol{z}_i^e$ via Eq. 3
    end
  end
  Feed the soft-labeled exemplars $\mathcal{E}_t = \{\boldsymbol{z}_i^e, \tilde{\boldsymbol{y}}_i^e\}_{i=1}^{N_0}$ to the classifier φ for training
  $\mathcal{Z}_t^0 = \{\}$
  for $\boldsymbol{z}_i^u$ *in* $\mathcal{Z}_t^u$ **do**
    Get the probability prediction $\boldsymbol{p}_i = \{p_{ij}\}_{j=1}^{|\mathcal{C}_t^l|}$ by feeding $\boldsymbol{z}_i^u$ into the fitted classifier φ
    Calculate the uncertainty u_i in $\boldsymbol{p}_i$ via Eq. 4
    Get the probability p_{i0} that $\boldsymbol{z}_i^u$ comes from the unknown via $\alpha \times u_i$
    Recognize $\boldsymbol{z}_i^u$ via $y_i^* = \arg\max_{j\in\{0,\mathcal{C}_t^l\}} p_{ij}$
    if $y_i^* == 0$ **then**
      Append $\boldsymbol{z}_i^u$ to $\mathcal{Z}_t^0$
    end
  end
  return $\mathcal{Z}_t^0$

Algorithm 4: Assisting Manual Annotation Procedure with GCD in OpenGCD.

Input: Rejected unlabeled feature embeddings: $\mathcal{Z}_t^0 = \{\boldsymbol{z}_i^0\}_{i=1}^{M_t^0}$; Buffer: $\mathcal{M}_r$; Maximum total number of classes: $|\mathcal{C}_t^{\max}|$.
Output: Labeled novel class instances: $\mathcal{X}_t^n$.

Function `AssistingManualAnnotationGCD` $(\mathcal{Z}_t^0, \mathcal{M}_r, |\mathcal{C}_t^{\max}|)$**:**
  Get labeled exemplars $\mathcal{E}_t = \{\boldsymbol{z}_i^e, y_i^e\}_{i=1}^{N_0}$ from $\mathcal{M}_r$
  Split $\mathcal{E}_t$ into $\mathcal{E}_t^a$ and $\mathcal{E}_t^v$ by $|\mathcal{C}_t^a| : |\mathcal{C}_t^v| = 2:1$
  Determine the optimal estimated number of classes $|\hat{\mathcal{C}}_t|$ by `Brent` $\left(\texttt{ss-k-means++}(k, \mathcal{E}_t^a, \mathcal{E}_t^v \cup \mathcal{Z}_t^0), |\mathcal{C}_t^{\max}|\right)$
  `// Bounded by` $(|\mathcal{C}_t^a|, |\mathcal{C}_t^{\max}|]$`, supervised by` $\mathcal{E}_t^a$`, aimed at maximizing ACC on` $\mathcal{E}_t^v$ `+ SC on` $\mathcal{Z}_t^0$
  Get $\hat{\mathcal{Z}}_t^n = \{\boldsymbol{z}_i^n, \hat{y}_i^n\}_{i=1}^{\hat{M}_t^n}$ by `ss-k-means++` $(|\hat{\mathcal{C}}_t|, \mathcal{E}_t, \mathcal{Z}_t^0)$ `// supervised by` $\mathcal{E}_t$`,` $k = |\hat{\mathcal{C}}_t|$
  Pick the instance set $\hat{\mathcal{X}}_t^n$ corresponding to $\hat{\mathcal{Z}}_t^n$
  Get $\mathcal{X}_t^n = \{\boldsymbol{x}_i^n, y_i^n\}_{i=1}^{M_t^n}$ by manually revision and label $\hat{\mathcal{X}}_t^n$
  return $\mathcal{X}_t^n$