Grow and Merge: A Unified Framework for Continuous Categories Discovery

Anonymous Author(s)
Affiliation
Address
email

Abstract

Although a number of studies are devoted to novel category discovery, most of them assume a static setting where both labeled and unlabeled data are given at once for finding new categories. In this work, we focus on the application scenarios where unlabeled data are continuously fed into the category discovery system. We refer to it as the **Continuous Category Discovery** (**CCD**) problem, which is significantly more challenging than the static setting. A common challenge faced by novel category discovery is that different sets of features are needed for classification and category discovery: class discriminative features are preferred for classification, while rich and diverse features are more suitable for new category mining. This challenge becomes more severe for dynamic setting as the system is asked to deliver good performance for known classes over time, and at the same time continuously discover new classes from unlabeled data. To address this challenge, we develop a framework of **Grow and Merge** (**GM**) that works by alternating between a growing phase and a merging phase: in the growing phase, it increases the diversity of features through a continuous self-supervised learning for effective category mining, and in the merging phase, it merges the grown model with a static one to ensure satisfying performance for known classes. Our extensive studies verify that the proposed GM framework is significantly more effective than the state-of-the-art approaches for continuous category discovery.

1 Introduction

Human beings are good at grouping objects into category through clustering, and the definition of categories are continuously expanding and updated over time. Recent developments of intelligent visual systems can not only distinguish pre-defined categories [1–5], but also discover new categories from unlabeled data, a task that is known as *novel category discovery* [6–8].

Existing works on novel category discovery are limited to the *static* setting, where both labeled data (by known classes) and unlabeled data (with potential unknown categories) are given at once. In contrast, for real-world applications, unlabeled data are continuously fed into the system for discovering new categories, making it a significantly more challenging problem. Besides, current studies for novel category discovery often assume that all the unlabeled data belong to the unknown new categories, which is generally not true in real applications. In this work, we examine the dynamic setup of novel category discovery where the system was initially given a set of data labeled by known classes, and unlabeled data are continuously streamed into the system for discovering new classes. The system is requested to consistently yield satisfying performance for known classes, and at the same time, dynamically discover new categories from the streaming unlabeled data. We refer to it as **Continuous Category Discovery**, or **CCD** for short.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

We illustrate the process of CCD in Figure 1. It is comprised of two main stages: the *initial stage* where a classification model is trained by a set of labeled examples, and the *continuous category discovery stage* where new categories are continuously discovered from a stream of unlabeled data belonging to both known and unknown classes. A intuitive approach to address the dynamic nature of CCD is to combine the existing methods for open-set recognition [9, 10], novel category discovery [11, 6, 12], and incremental learning [13, 14]. This is however insufficient because our learning system has to accomplish two tasks at the same time, *i.e.*, accurately classify instances into the known classes, and discover new categories from an unlabeled data stream. It turns out that these two task models usually produce different types of features: discriminative features on known classes are preferred by classification model, while rich and diverse features are critical for identifying new classes, as illustrated in Figure 2. A simple combination of novel category discovery and incremental learning will fail to address the trade-off consistently over time, which is further verified by our empirical studies.

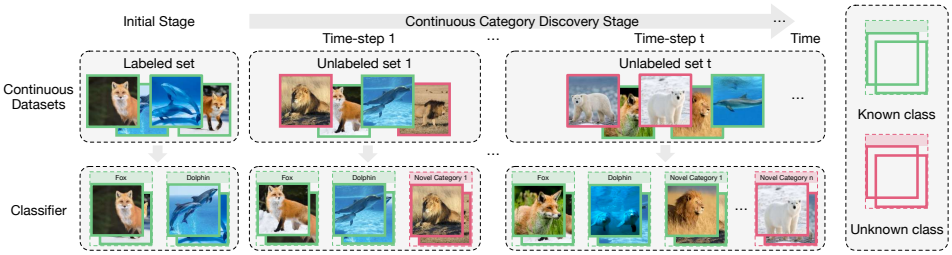

Figure 1: **Overview of the Continuous Category Discovery (CCD).** The continuous data stream is mixed with unlabeled samples from both known and novel categories. CCD requires to distinguish known categories, discover novel categories and merge the discovered categories into known set.

To address the challenge of continuous category discovery, we propose a framework of **Grow** and **Merge**, or **GM** for short. After pre-training a static model $\mathcal{A}$ over the labeled data, we will update model $\mathcal{A}$ with respect to unlabeled data stream by alternating between the growing phase and the merging phase: in the growing phase, we will increase the diversity of features by continuously training our model over received unlabeled data through a combination of supervised and self-supervised learning; in the merging phase, we will merge the grown model with the static one by taking a weighted combination of both models. By alternating between the growing and merging phases, we are able to maintain a good performance for known classes, and at same time, the power of discovering new categories. This is clearly visualized in Figure 2, where the first two panels show that existing approaches can do well on one of the two tasks but not both, and last panel shows that features learned by the proposed GM framework works well for both tasks.

Finally, one of the common issues with continuous training is catastrophic forgetting. To alleviating the forgetting effect as we are growing the number of categories over time, we maintain a small set of labeled samples from known categories and pseudo-labeled samples from novel categories. These selected examples are used in the growing phase to expand feature diversity for effective category discovery. Extensive experimental results show that our proposed method consistently shows satisfying performance under multiple practical scenarios compared with existing methods.

The main contributions of this paper are summarized as follows:

- We study a new problem named continuous category discovery, or CCD, which better reflects the challenge of category discovery in the wild. It needs to simultaneously maintain a good performance for known categories and the ability of discovering novel categories.
- We propose a framework of grow and merge, or GM, for CCD, that is able to resolve the conflicts between the classification task and the task of discovering new categories.
- We conduct experiments under four different settings to fully investigate the scenarios of novel category discovery in the wild. The proposed method shows less forgetting of known categories and better performance for category discovery compared to existing methods.

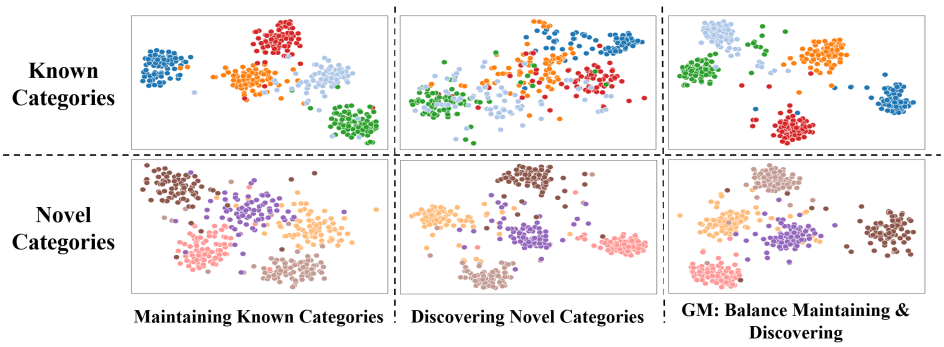

Figure 2: Features visualization of model A trained on known categories, model B trained for novel category discovery (based on model A) and the proposed GM model.

2 Problem Definition

In this section, we formulate the problem of Continuous Category Discovery (CCD). There are two main stages of the CCD problem, *i.e.*, **Initial Stage** and **Continuous Category Discovery Stage**. The settings of each stage are introduced, together with the evaluation metrics of CCD problem.

2.1 Setting of Continuous Category Discovery

During the initial stage, a labeled training dataset $\mathcal{D}^0_{train} = \{(\boldsymbol{x}^0_i, y^0_i)\}_{i=1}^{N^0}$ is provided to train the model on the initial known category set $\mathcal{C}^0 = \{1, 2, ..., K^0\}$, where $\boldsymbol{x}^0_i$ is the initial training data, and $y^0_i \in \mathcal{C}^0$ is the corresponding label. The model is expected to classify categories in $\mathcal{C}^0$, and learn meaningful representation from the labels' semantic information.

During the continuous category discovery stage, a serial of unlabeled training datasets $\{\mathcal{D}^t_{train}\}_{t=1}^{T}$ are sequentially provided, where $\mathcal{D}^t_{train} = \{\boldsymbol{x}^t_i\}_{i=1}^{N^t}$ indicates the dataset at time t. Though unlabeled, we denote the known or potential appeared categories until time t as $\mathcal{C}^t = \{1, 2, ..., K^t\}$. The model is expected to discover newly appeared unknown categories from $\mathcal{D}^t_{train}$, store the representations of the discovered novel categories and maintain the knowledge of the known categories.

2.2 Evaluation Metrics

At each time-step t, we evaluate the classification performance on the test dataset $\mathcal{D}^t_{test} = \{(\boldsymbol{x}^s_i, y^s_i) | s \leq t\}$, containing test samples from all the known or previously discovered categories. For the newly appeared unknown categories, we evaluate the novel category discovery performance. The **maximum forgetting metric** $\mathcal{M}_f$ and the **final discovery metric** $\mathcal{M}_d$ are designed for evaluation. To evaluate the performance of the clustering assignments, we follow the standard practise [6, 15] to adopt clustering accuracy on the known categories and the novel categories, denoted as $\text{ACC}^t_{\text{known}}$ and $\text{ACC}^t_{\text{novel}}$, respectively. The maximum forgetting $\mathcal{M}_f$ is defined as the maximum value of the differences between $\text{ACC}^0_{\text{known}}$ and $\text{ACC}^t_{\text{known}}$ for every t and the final discovery $\mathcal{M}_d$ is defined as the final cluster accuracy on novel categories, *i.e.*, $\mathcal{M}_d = \text{ACC}^T_{\text{novel}}$. See more details in the supplementary materials.

3 Method

3.1 An Overview of Grow and Merge Framework

The CCD problem requires both maintaining the performance on known categories and discovering novel categories under the continuous learning scheme. For the compatibility of the two tasks, we propose a **Grow and Merge (GM)** Framework, which is illustrated in Figure 3.

During the initial stage, the model $\phi^0(\cdot)$ is firstly pre-trained on the initial dataset with labeled data in both supervised and self-supervised manner. An *exemplars* set $P_k = \{\mathbf{p}_{k,i}\}$ is constructed as

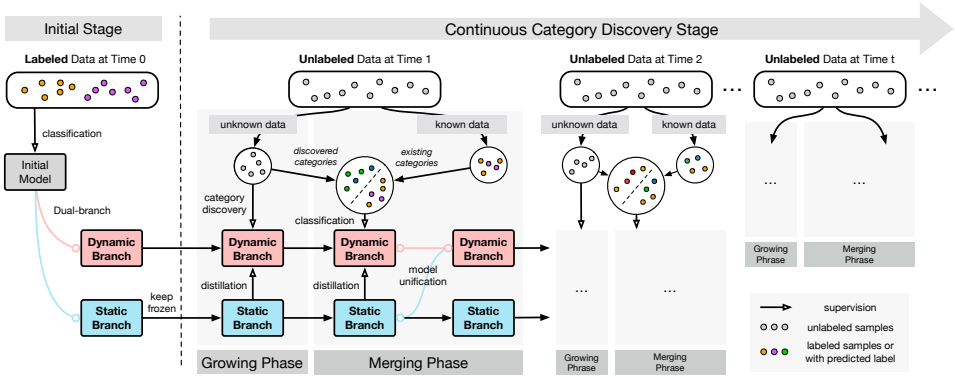

Figure 3: **The proposed Grow and Merge (GM) Framework.** There exist growing and merging phases for each time-step, where a dynamic branch is freely updated for discovering new classes in the growing phase, and then unified with a frozen static branch in the merging phase for maintaining classification on all appeared classes.

well as the *prototypes*. The exemplar set contains representative samples for each class while the prototype μ_k is the averaged feature vector of the exemplars from the k-th class. The exemplar set is constructed by iteratively picking several typical samples with the principle of maintaining the closest center to the prototype μ_k, *i.e.* $\mathbf{p}_{k,i} \leftarrow \text{argmin}_{\mathbf{x}} \parallel \mu_k - \frac{1}{i}[\phi^0(\mathbf{x}) + \Sigma_{j=1}^{i-1}\phi^0(\mathbf{p_{k,j}})] \parallel$. The definitions of prototype and exemplars are adapted from iCaRL [14], which enables us to use a non-parametric classifier that predicts label of a query sample from its nearest prototype's label.

During the continuous category discovery stage, a dual-branch architecture with a static branch $\phi_S(\cdot)$ and a dynamic branch $\phi_D(\cdot)$ is introduced for training, where both branches are initialized by the pre-trained model $\phi^0(\cdot)$. At each time-step, GM includes two phases: **Growing** and **Merging**. In the growing phase, the novel category data are filtered out from the continuous data and the dynamic branch is trained for novel category discovery. Then the merging phase fuses the newly discovered and known categories into a single model for unified classification. Knowledge from the frozen static branch is distillated into the dynamic one to avoid forgetting previous classes. It is noted that though GM maintains two branches during training, only one branch is needed during testing.

Experimental results show significant performance of the GM framework, where the dual-branch architecture and knowledge distillation succeed to balance the feature conflicts between continuous classification and novel categories discovery, and show better performance on both sub-tasks.

3.2 Growing: Enrich Features with Unlabeled Data

To achieve novel category discovery from the data stream with both labeled and unlabeled samples from known and unknown classes, we firstly filter out data of unknown classes with **novelty detection**, and then conduct **novel category discovery** based on both the new data and the previous model.

Novelty Detection aims to identify the samples belonging to the novel categories in the current training dataset $\mathcal{D}^t_{train}$, since there may also exist samples from previously known categories. With the maintained prototypes of known categories, the feature vectors of samples from known categories would lay closer to at least one prototype with a high probability. Then a novel distance d_{novel} is defined as the distance between the sample $\mathbf{x}^t_i$ and its nearest prototype, *i.e.* $d_{\text{novel}}(\boldsymbol{x}^t_i) = \min_{1\leq k\leq K^t} d(\mu_k, \phi^t_D(\boldsymbol{x}^t_i))$, where $d(\cdot,\cdot)$ is the distance function. Samples with d_{novel} larger than a threshold ϵ are filtered out as novel samples for the following process.

Novel Categories Discovery and Learning from Static Branch. We optimize the dynamic branch $\phi^t_D(\cdot)$ for novel category discovery, guided by both the novel samples and previously learned knowledge for rich feature representation. (i) For learning and discovering from the novel samples, we adopt an unsupervised representation learning method [6]. A cluster head $g(\cdot)$ is learned to assign samples $\hat{\boldsymbol{x}}^t_i$ to the clusters with novel category indices $\{K^{t-1}+1, ..., K^t\}$. Such an assignment is

supervised by the similarity s_{ij} between samples $\hat{\boldsymbol{x}}_i^t$ and $\hat{\boldsymbol{x}}_j^t$ with a binary cross entropy loss, *i.e.*

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{\hat{N}^t}\sum_{i=1}^{\hat{N}^t}\sum_{j=1}^{\hat{N}^t}\left(s_{ij}\log {\boldsymbol{c}_i^t}^\top \boldsymbol{c}_j^t + (1 - s_{ij}\log(1 - {\boldsymbol{c}_i^t}^\top \boldsymbol{c}_j^t))\right), \tag{1}$$

where $\hat{N}^t$ is the number of novel category samples at time t (specified or estimated). The similarity s_{ij} is measured by the Winner-Take-All hash (WTA), which equals one when samples share the same top-k channels of their feature vector, otherwise equals zero.

(ii) For learning from the static branch $\phi_D^t(\cdot)$ with previously learned knowledge, we introduce a *static-dynamic distillation loss* to transfer the representation learned from labeled data in the static branch into the dynamic one:

$$\mathcal{L}_{\text{SD}} = \frac{1}{N^t}\sum_{i}^{N_t} d\left(\boldsymbol{z}_i^t, \boldsymbol{z}_i'^t\right) = \frac{1}{N^t}\sum_{i}^{N_t} d\left(\phi_S(\boldsymbol{x}_i^t), \phi_D^t(\boldsymbol{x}_i^t)\right) \tag{2}$$

The joint loss $\mathcal{L}_{\text{BCE}}$ and $\mathcal{L}_{\text{SD}}$ enables the dynamic branch to discovering novel categories with richer feature representation, and the cluster head $g(\cdot)$ could assign new labels to the novel category data $\hat{\boldsymbol{x}}_i^t$.

3.3 Merging: Unification of Categories and Branches

After the Growing phase, both categories and models contain two parts for the known and novel categories, respectively. The Merging phase reunifies them together for all the appeared categories.

3.3.1 Category Unification: Continuous Learning for Newly Discovered Categories

To merge the novel categories discovered in Growing phase into the classifier, we firstly **sift samples** to construct the exemplar sets of the novel categories, then refine the model representation using a **pseudo label representation learning** for classification on both known and novel classes.

Sample Sifting aims to filter out incorrectly-assigned samples based on their cluster confidence. The assigned class indices by $g(\cdot)$ are denoted as the pseudo-labels of the novel samples, which may inevitably have errors and lead to less reliable prototypes of the novel categories. We sift out these potentially noisy samples via the local sample density, with the assumption that samples from the same class would lay tightly. Suppose $\mathcal{N}_j(\hat{\boldsymbol{z}}_i^t)$ indicates the j nearest neighbors of $\hat{\boldsymbol{z}}_i^t$ from the same class. The local sample density $G_j(\hat{\boldsymbol{z}}_i^t)$ is defined as the distance between $\hat{\boldsymbol{z}}_i^t$ and its j-th nearest neighbors, *i.e.* $G_j(\hat{\boldsymbol{z}}_i^t) = \max_{\boldsymbol{z}\in\mathcal{N}_j(\hat{\boldsymbol{z}}_i^t)} d(\hat{\boldsymbol{z}}_i^t, \boldsymbol{z})$. Samples with larger G_j are sifted out, then the left samples are denoted as $\tilde{\boldsymbol{x}}_i^t$ with pseudo labels $\tilde{c}_i^t$.

Pseudo Label Representation Learning. Based on the sifted novel samples of the k-th novel category, we firstly construct the exemplar set $P_k = \{\mathbf{p}_{k,i}\}$. Without knowing the prototype μ_k, *i.e.* the explicit center of the k-th category, we estimate it with the averaged vectors of sifted samples. For the purpose of continuous classification on all appeared classes, it is required to tighten the sparse representation distribution learned in the growing phase. Pseudo Label Learning (PLL) loss is introduced here. Compared with commonly-used Cross Entropy loss, PLL does not require an explicit classifier and thus does not modify the deployed model structure during the continuous phase. The loss function $\mathcal{L}_{\text{PLL}}$ aims to pull exemplars pull the exemplars $\boldsymbol{p}_k$ and the corresponding prototypes μ_k closer, and push $\boldsymbol{p}_k$ away from the other prototypes $\mu_j (j \neq k)$:

$$\mathcal{L}_{\text{PLL}} = -\frac{1}{K^t}\sum_{k=1}^{K^t}\frac{1}{|P_k|}\sum_{i=1}^{|P_k|}\log\frac{\exp(\phi_D^t(\boldsymbol{p}_{k,i})\cdot\mu_k/\tau)}{\sum_{j=1}^{K^t}\exp(\phi_D^t(\boldsymbol{p}_{k,i})\cdot\mu_j/\tau)}, \tag{3}$$

where τ is a temperature hyper-parameter. To avoid catastrophic forgetting, based on data in the latest exemplars, the static-dynamic distillation loss in Eq. 2 is also employed here.

3.3.2 Branch Unification: Merging Multiple Branches into One Unified Model

During the above process, the dynamic branch is optimized to classify all appeared categories. In order to reduce computation cost and further utilize the representation of the static branch trained

from labeled data, we merge the two branches into a single one. In experiments, we found such a branch merging can further improve the recognition performance. To fuse the representation of ϕ_S and ϕ_D^t, the exponential moving average strategy (EMA) is applied:

$$\theta_{\phi_D^t} \longleftarrow \alpha\theta_{\phi_S} + (1-\alpha)\theta_{\phi_D^t}, \tag{4}$$

where α is the momentum hyper-parameter to control the weight of fusion, generally valued close to 1 ($\alpha = 0.99$ in this paper).

4 Experiments

In this section, we design experiments in different scenarios to simulate the varying cases of real-world applications, which enables us to fully investigate the performance and flexibility of the proposed method. Two main evaluation metrics, namely, maximum forgetting $\mathcal{M}_f$ and final discovery $\mathcal{M}_d$ are employed to measure the performance. CIFAR-100 [16], CUB-200 [17] and ImageNet-100 [18, 19] representing the small, middle and large scale datasets are used in the experiments.

Experimental Scenarios.We formulate four different scenarios for experiments as follows. (1) *Class Incremental Scenario (CI)*: the data are only drawn from novel categories. (2) *Data Incremental Scenario (DI)*: the data are only drawn from the known categories. (3) *Mixed Incremental Scenario (MI)*: the data are drawn from both novel and known categories. (4) *Semi-supervised Mixed Incremental Scenario (SMI)*: the data are drawn from both novel and known categories, and a portion of the data are labeled, which is closed to the real-world application.

Table 1: The content of $\mathcal{D}_{train}^t$ under different scenarios DI, CI, MI and SMI.

	Known Categories	Novel Categories	Labeled data	Unlabeled data
CI	✗	✓	✗	✓
DI	✓	✗	✗	✓
MI	✓	✓	✗	✓
SMI	✓	✓	✓	✓

The settings of these scenarios are summarized in Table 1. We mainly focus on the standard scenario CI in this paper, while the experimental results on DI, MI, and SMI are reported to demonstrate the flexibility of the proposed method. Each scenario contains $T = 3$ time-steps of the continuous learning stage. Details of the scenarios could be found in supplementary materials.

Implementation Details. For all experiments, we train the model 100 epochs using $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{SD}}$ loss and 100 epochs using $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{SD}} + \mathcal{L}_{\text{PLL}}$ loss for each time-step t. The α for EMA is set to 0.99. The novelty detection threshold ϵ is set to 0.6. 15 neighbors are selected for each sample during sample sifting and the top 50% samples with larger local density are sifted out. The temperature τ used in $\mathcal{L}_{\text{PLL}}$ is set to 0.1. Mean Squared Error (MSE) Consistency Loss is also used during the continuous learning following AutoNovel [6]. More details could be found in supplementary materials.

4.1 Comparisons with SOTA methods on CI scenario

We first investigate the performance of the proposed method under CI scenario. The CIFAR-100, CUB-200, ImageNet-100, Stanford-Cars, and FGVC-Aircraft are used in this experiment. The maximum forgetting $\mathcal{M}_f$ and final discovery $\mathcal{M}_d$ are reported in Table 2. Since there are few works about continuous novel category discovery and recent works about incremental learning may not fit the unsupervised scenario, we compare the proposed method with the lower bound of $\mathcal{M}_f$ and the upper bound of $\mathcal{M}_d$, two recent methods for novel category discovery, *i.e.*, AutoNovel [6] and DRNCD [15], and the combinations of the novel category discovery methods and continuous learning methods.

It is noticed that there is not an exact lower/upper bound method under the CI scenario. Therefore, we conduct two methods under the **offline setting**, where all of the data containing both known and novel categories are available in a single stage. We present the results of K-means [20] as a reference to the lower bound of $\mathcal{M}_f$, and the original AutoNovel method as the upper bound of $\mathcal{M}_d$. It should be enhanced that the lower/upper bound results are for reference and could not exactly represents

Table 2: Experimental results under CI scenario.

Method	CIFAR-100		CUB-200		ImageNet-100		Stanford-Cars		FGVC-Aircraft	
	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$
lower-bound of $\mathcal{M}_f$ (offline K-Means) [20]	5.81	16.06	3.05	13.65	2.42	17.53	2.53	10.05	3.73	14.70
upper-bound of $\mathcal{M}_d$ (offline AutoNovel) [6]	4.17	47.87	8.37	16.19	6.48	35.13	11.27	19.46	4.85	26.90
AutoNovel (online) [6]	68.08	24.53	60.43	19.05	63.18	7.92	71.88	16.46	53.84	25.23
DRNCD (online) [15]	62.83	11.23	42.66	6.67	46.83	6.53	54.50	1.83	43.63	2.21
AutoNovel (online) + LwF [6, 13]	22.17	28.37	59.53	21.94	61.14	7.60	71.60	20.35	44.90	11.67
DRNCD (online) + LwF [15, 13]	27.03	2.87	44.83	6.31	31.13	5.47	27.86	4.60	33.09	9.03
iCaRL (fixed exemplars) + LwF [14, 13]	35.11	-	41.93	-	9.97	-	24.71	-	21.31	-
GM (Ours)	**9.79**	**35.77**	**18.52**	**26.07**	**8.00**	**26.80**	**18.00**	**25.16**	**8.41**	**30.82**

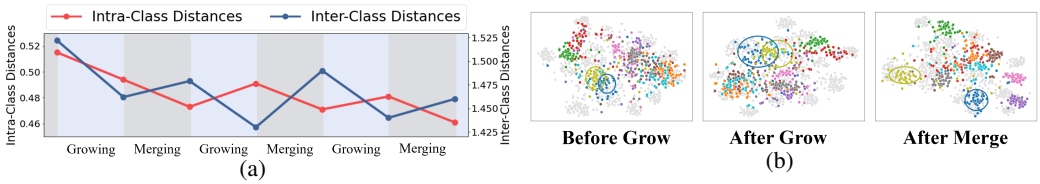

Figure 4: (a) The intra-/inter- class distance measured throughout the growing and merging phases. (b) Feature visualizations at different time of GM, where the colored dots denote previously appeared classes, and the gray dots denote the newly appeared classes without labels.

the best performance on both $\mathcal{M}_f$ and $\mathcal{M}_d$. Since recent novel category discovery and incremental learning methods usually contain multiple stages, we chose AutoNovel and LwF [13] and combine them for comparison based on the flexibility of implementation. It is observed that AutoNovel with LwF greatly improves the $\mathcal{M}_f$ and $\mathcal{M}_f$ compared to the online version of AutoNovel. However, such a simple combination not completely solves the feature conflict caused by the two different tasks. The proposed GM framework divides and conquers the conflict of features and achieves the best results among the compared methods (that conduct classification and discovery training in one single phase) on all datasets, indicating the effectiveness of the two phases of growing and merging.

Table 3: Experimental results under DI scenario.

Method	$\mathcal{M}_f \downarrow$
Original Model	0
Pseudo iCaRL	9.90
DeepClustering [21]	2.10
AutoNovel* [22, 23]	0.38
GM	**-23.87**

Table 4: Experimental results under MI and SMI scenario.

Method	MI		SMI	
	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$
AutoNovel [6]	63.11	5.92	62.44	17.33
DRNCD [15]	44.79	4.00	55.84	47.07
AutoNovel + LwF [6, 13]	62.51	3.34	64.25	30.87
DRNCE + LwF [15, 13]	37.01	3.20	63.35	**54.70**
GM w/o. novelty detection	**4.56**	8.53	**8.07**	30.93
GM	9.83	**32.41**	8.84	37.23

4.2 Extended Experiments on More Complicated Scenarios

In this part, we extend GM to more complicated scenarios: DI, MI, and SMI scenarios. The experiments are conducted on the CIFAR-100 dataset and the results are listed in Table 3 and Table 4. Due to there is no novel category in the incremental unlabeled data under DI, we remove the cluster head in the AutoNovel to form compared method AutoNovel*. Incremental learning method iCaRL [14] combined with pseudo label assignment and unsupervised learning method DeepClustering combined with finetuning on incremental data are also chosen for comparisons. As shown in Table 3, after finetuning on the incremental unlabeled data, all comparison methods show the catastrophic forgetting effect, and the performances are even worse than the original model. GM avoids catastrophic forgetting and effectively utilizes continuous unlabeled data, further improving the final performance. In MI and SMI scenarios, incremental data are mixed with known and novel categories, which makes label assignment for incremental data more difficult. Nonetheless, as shown in Table 4, GM can still effectively avoid catastrophic forgetting and achieve superior performance (especially on $\mathcal{M}_f$) with the help of two phases of growing and merging, compared to the comparison methods.

4.3 Ablation Studies

Previous experimental results indicate Grow and Merge framework can better accomplish the continuous learning and the novel category discovery in multiple scenarios. Here we provide intra- and inter- class distance among features and the visualizations in different time of GM in Figure 4. It can be clearly observed that with the alternation of the growing and merging phases, the intra-class distance first increases and then decreases. The classes are relatively compact but overlapped with others before growing phase, loose after growing phase, and compact and separated with others after merging phase. The results are consistent to our motivation that the GM relaxes and tightens the features alternatively to resolve the feature conflicts caused by CCD task.

We also provide detailed analysis for all proposed key components of GM. Ablation studies are conducted on the CIFAR-100 dataset under CI scenario, shown in Table 5 and Tabel 6.

Growing Phase. We investigate the performance of introducing additional methods for representation learning in the growing phase. Combining with cross entropy loss (CE) of the exemplars degrades $\mathcal{M}_d$, which validates that relaxing the constraint of memorizing previous category information can facilitate category discovery. Thus, we further introduce more relaxed self-supervised methods. Combing SimCLR [3] also degrades $\mathcal{M}_d$, which could be attributed that pushing the representation of negative samples away hurts the performance. An alternative method SimSiam [24] achieves comparable performance without pushing negative samples, which also validates the importance of a relaxed representation in this phase.

Merging Phase. In the merge phase, PLL is employed to improve the representation. We evaluate the performance of the models replacing PLL with other representation learning methods. *CE* represents cross entropy loss with the pseudo label given by the prototypes and *SSL* denotes standard InfoNCE loss [3, 4]. The experimental results demonstrate that, though these alternative methods could perform competitively on known classes, they sacrifice the performance of discovering novel classes.

Model Merging. We evaluate the performance of the model without EMA. $\mathcal{M}_f$ increases for 55% and $\mathcal{M}_d$ decreases for 14%, which demonstrates the necessity of EMA. We also consider an alternative fuse method which assigns $\phi_S \leftarrow \phi_D^t$ after the update of ϕ_D^t by Eq. 4. $\mathcal{M}_d$ is significantly affected with a decrease of 33%. Therefore, we use the fixed ϕ_S in this paper.

Static-Dynamic Branch. The multi-branch aims to maintain known category information and discover novel categories on different branches. We first evaluate the performance when directly replacing the multi-branch with a single one. The results show that $\mathcal{M}_f$ increases dramatically, which demonstrates the importance of the multi-branch to preserve already learned categories. We further investigate the components of the static-dynamic branch. Removing the initialization of branch ϕ_D^0 also degrades both $\mathcal{M}_f$ and $\mathcal{M}_d$. For the proposed loss $\mathcal{L}_{\text{SD}}$ aligning the feature space of ϕ_S and ϕ_D^t, results show that the model without $\mathcal{L}_{\text{SD}}$ performs worse on both $\mathcal{M}_f$ and $\mathcal{M}_d$ for the separation of the feature spaces leading to the failure of the prototype mechanism. The model replacing $\mathcal{L}_{\text{SD}}$ with DINO [25] increases $\mathcal{M}_f$ for 3% and decreases $\mathcal{M}_d$ for 6.3%.

Sifting in Merging. Sifting aims to sift out samples with low cluster confidence. Except for the criterion based on the adopted local geometry property, there are several potential criteria to sift samples. *Confidence Ranking (CR)* selects the portion of all samples with the highest confidence score given by the cluster head. *Conditional Confidence Ranking (CCR)* selects the portion of samples from each class with the highest confidence score. *Confidence Filtering (CF)* selects the samples with the confidence score higher than a given threshold. The results show that: (i) For $\mathcal{M}_f$, the model without sifting and CR performs worse, and CCR, CF and the local sample density based method performs at almost same level. (ii) For $\mathcal{M}_d$, the local sample density based method outperforms all of these compared methods.

Hyperparameter Analysis of Static-Dynamic Branch. We also evaluate the effect of α in Eq. 4, shown in Figure 5. The hyper-parameter α controls the importance of ϕ_S relative to ϕ_D^t. For the model with only EMA, both $\mathcal{M}_f$ and $\mathcal{M}_d$ decrease with the increase of α, since the model with higher α maintains more information from the known categories. The model with only $\mathcal{L}_{\text{SD}}$ (*i.e.*, the model w/o. EMA shown in Tabel 5) performs worse than the proposed full model, which demonstrates that only the constrain of the distance between the feature space of ϕ_S and ϕ_D^t is not enough to transfer the knowledge from ϕ_S to ϕ_D^t. The performances of the models with both EMA and $\mathcal{L}_{\text{SD}}$ varies smoother, compared with the models with only EMA. Therefore, the models could achieve higher performances and be more robust by combining EMA and $\mathcal{L}_{\text{SD}}$.

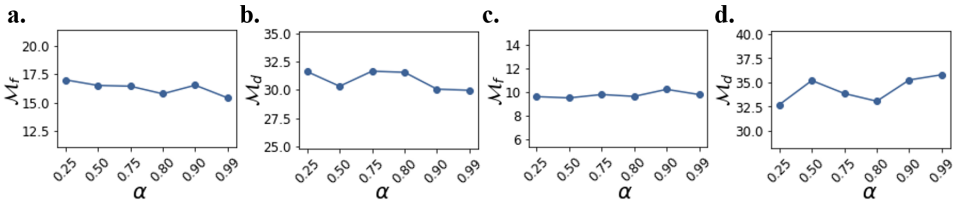

Figure 5: The variation comparison of the performance of the models by varying α. **a**: $\mathcal{M}_f$ of the model with only EMA; **b**: $\mathcal{M}_d$ of the model with only EMA; **c**: $\mathcal{M}_f$ of the model with both EMA and $\mathcal{L}_{\text{SD}}$; **d**: $\mathcal{M}_d$ of the model with both EMA and $\mathcal{L}_{\text{SD}}$.

Table 5: Results of ablation studies on GM phases.

		$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$
GM		9.79	35.77
Growing Phase	WTA+$\mathcal{L}_{\text{SD}}$+CE	9.51	31.50
	WTA+$\mathcal{L}_{\text{SD}}$+SimCLR	9.69	31.47
	WTA+$\mathcal{L}_{\text{SD}}$+SimSiam	9.51	34.07
Merging Phase	CE + PLL	8.89	33.23
	CE	7.41	31.03
	SSL + PLL	9.26	32.94
	SSL	7.23	29.03
Model Merging	w/o. EMA	15.27	30.87
	Update ϕ_S	10.67	24.13

Table 6: Results of ablation studies on Static-Dynamic Branch and Sifting in the Merging.

		$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$
GM		9.79	35.77
Static-Dynamic Branch	w/o. Multi-Branch	33.23	35.10
	w/o. Initialize ϕ_D^0	27.91	31.77
	w/o. $\mathcal{L}_{\text{SD}}$	15.41	29.97
	DINO	9.82	33.53
Sifting in the Merging	w/o. Sifting	9.83	31.44
	Sifting (CR)	9.93	32.46
	Sifting (CCR)	9.76	33.76
	Sifting (CF)	9.74	31.46

5 Related Work

Novel Class Discovery [6–8, 26] aims to discover unknown classes from unlabeled data by leveraging pairwise similarity [6, 27] or employing unsupervised clustering [12, 28]. Most of these methods assume that there is a pure unlabeled set that only contains samples from unknown categories, which means the class space of unlabeled data is completely disjoint with the labeled set. And at test time, samples are all from novel classes. These assumptions limit their applications in the real world. CCD aims to tackle novel class discovery under a more realistic situation in which the unlabeled data (i) are continuously confronted and (ii) contain both known and unknown classes.

Continual Learning needs the model to extend its ability to address new tasks when confronted with new data [29–37]. The main challenge of continual learning is to avoid catastrophic forgetting [38, 39], *i.e.*, the performance degradation on previously learned tasks when new ones are ingested. Existing methods can be categorized into replay methods [14, 40, 41], regularization-based methods [13, 42, 43] and parameter isolation methods [44–46]. Most of the methods assume there are available labeled data in each time-step. Nevertheless, the data in continuous streams are more likely to be unlabeled in reality, which means the category information is unknown and needs to be discovered.

Different from existing novel category discovery and continual learning, CCD requires the model to discover novel categories from the continuous unlabeled data while maintaining the classification performance on known and discovered classes.

6 Conclusion

In this paper, we study the continuous category discovery (CCD) problem where unlabeled data are continuously fed into the category discovery system. The main challenge of the CCD problem is that different sets of features are needed to classify known categories and discover novel categories. To address this challenge, we develop a framework called Grow and Merge that alternatively updates the model in the growing phase and the merging phase. Extensive experimental results demonstrate that GM outperforms the existing methods with the less forgetting effect of known categories and higher accuracy for category discovery.

References

[1] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285, 2020.

[2] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, 2020.

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9726–9735, 2020.

[5] Yutong Feng, Jianwen Jiang, Mingqian Tang, Rong Jin, and Yue Gao. Rethinking supervised pre-training for better downstream transferring. In *International Conference on Learning Representations*, 2022.

[6] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[7] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *IEEE/CVF International Conference on Computer Vision*, pages 9264–9272, 2021.

[8] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. *CoRR*, abs/2201.02609, 2022.

[9] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

[10] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10): 3614–3631, 2021.

[11] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *International Conference on Learning Representations*, 2020.

[12] 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.

[13] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pages 614–629, 2016.

[14] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5533–5542, 2017.

[15] Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 22982–22994, 2021.

[16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[17] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[18] 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.

[19] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, pages 776–794, 2020.

[20] Hans Hermann Bock. Clustering methods: a history of k-means algorithms. *Selected Contributions in Data Analysis and Classification*, pages 161–172, 2007.

[21] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, pages 139–156, 2018.

[22] Jay Yagnik, Dennis Strelow, David A. Ross, and Ruei-Sung Lin. The power of comparative reasoning. In *IEEE International Conference on Computer Vision*, pages 2431–2438, 2011.

[23] Xuhui Jia, Kai Han, Yukun Zhu, and Bradley Green. Joint representation learning and novel category discovery on single- and multi-modal data. In *IEEE/CVF International Conference on Computer Vision*, pages 590–599, 2021.

[24] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[25] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9630–9640, 2021.

[26] Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 22982–22994, 2021.

[27] Xuhui Jia, Kai Han, Yukun Zhu, and Bradley Green. Joint representation learning and novel category discovery on single- and multi-modal data. In *IEEE/CVF International Conference on Computer Vision*, pages 590–599. IEEE, 2021.

[28] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10867–10875, 2021.

[29] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[30] Quang Pham, Chenghao Liu, and Steven C. H. Hoi. Dualnet: Continual learning, fast and slow. In *Advances in Neural Information Processing Systems*, pages 16131–16144, 2021.

[31] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co^{2}L: Contrastive continual learning. In *IEEE/CVF International Conference on Computer Vision*, pages 9496–9505, 2021.

[32] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Chenglin Liu. Class-incremental learning via dual augmentation. In *Advances in Neural Information Processing Systems*, pages 14306–14318, 2021.

[33] Hao Liu and Huaping Liu. Continual learning with recursive gradient optimization. In *International Conference on Learning Representations*, 2022.

[34] Zichen Miao, Ze Wang, Wei Chen, and Qiang Qiu. Continual learning with filter atom swapping. In *International Conference on Learning Representations*, 2022.

[35] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2022.

[36] Yu-Ming Tang, Yi-Xing Peng, and Wei-Shi Zheng. Learning to imagine: Diversify memory for incremental learning using unlabeled data. *CoRR*, abs/2204.08932, 2022.

[37] MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. *CoRR*, abs/2203.13381, 2022.

[38] Anthony V. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[39] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *CoRR*, abs/1312.6211, 2013.

[40] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

[41] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30*, pages 6467–6476, 2017.

[42] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, pages 3987–3995, 2017.

[43] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662, 2017.

[44] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[45] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision*, pages 72–88, 2018.

[46] Ju Xu and Zhanxing Zhu. Reinforced continual learning. In *Advances in Neural Information Processing Systems*, pages 907–916, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See supplementary materials.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplementary materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] Theoretical results are not included.
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplementary materials.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 4 and supplementary materials.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary materials.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.

(b) Did you mention the license of the assets? [N/A] The license of the assets are not provided on their official website.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Crowdsourcing and human subjects are not used in this research.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]