Learning Separable Fine-Grained Representation via Dendrogram Construction from Coarse Labels for Fine-grained Visual Recognition

Supplementary Material

6. Theoretical Analysis

In this section, we delineate BuCSFR from the perspective of the Expectation Maximization (EM) algorithm. The pseudocode of our method is given in Algorithm 1.

6.1. E-step in BuCSFR

For simplicity, the theoretical analysis is conducted on samples in the c-th coarse class, $X_c = \{x_i\}_{i=1}^{N_c}$. As described in Eq. (1), our goal is to determine the optimal network parameter θ^* that maximizes the log-likelihood function of samples X_c, expressed as

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{N_c} \log p(\mathbf{x}_i; \theta). \tag{15}$$

As mentioned in the paper, the absence of fine-grained labels poses a challenge to the top-down learning paradigm outlined in Eq. (15). To address this issue, we posit the existence of a set of latent prototypes for clusters within the c-th coarse class, $O_c = \{o_k\}_{k=1}^{M_c}$, which represent the nodes at the same level of constructed dendrogram and can be regarded as the underlying fine-grained classes. Consequently, Eq. (15) can be reformulated as

$$\begin{aligned} \theta^* &= \arg\max_{\theta} \sum_{i=1}^{N_c} \log p(\mathbf{x}_i; \theta), \\ &= \arg\max_{\theta} \sum_{i=1}^{N_c} \log \sum_{o_k \in O_c} p(\mathbf{x}_i, o_k; \theta). \end{aligned} \tag{16}$$

However, directly optimizing Eq. (16) is challenging because there are two parameters, o and θ, to optimize.
Thus, we introduce a surrogate function $Q(\cdot)$ to derive its lower bound:

$$\begin{aligned} &\sum_{i=1}^{N_c} \log \sum_{o_k \in O_c} p(\mathbf{x}_i, o_k; \theta), \\ =&\sum_{i=1}^{N_c} \log \sum_{o_k \in O_c} Q(o_k) \frac{p(\mathbf{x}_i, o_k; \theta)}{Q(o_k)}, \\ \geq&\sum_{i=1}^{N_c} \sum_{o_k \in O_c} Q(o_k) \log \frac{p(\mathbf{x}_i, o_k; \theta)}{Q(o_k)}, \end{aligned} \tag{17}$$

where $Q(o_k)$ ($\sum_{o_k \in O_c} Q(o_k) = 1$) denotes a distribution associated with the prototype o_k, and the last step is derived from Jensen's inequality. According to the EM algorithm, only when $\frac{p(\mathbf{x}_i, o_k; \theta)}{Q(o_k)}$ is a constant b, the inequality holds with equality, i.e., the middle is equal to the bottom in Eq. (17). In this case, the lower bound of Eq. (16) is maximized. Since $\frac{p(\mathbf{x}_i, o_k; \theta)}{Q(o_k)} = b$, there is

Algorithm 1 Pseudo-code of BuCSFR

```
Input: Training dataset $X$, Coarse label $Y$, network $f_\theta$,
       momentum queue $\mathcal{Q}$, number of clusters $K$, hyper-
       parameter $\tau_s$ and $\tau$,
for $c \leftarrow 1$ to $C$ do
    // clustering $X_c$ into $K$ clusters
    $O_c \leftarrow$ K-means($f_\theta(X_c)$)
end
for $epoch \leftarrow 1$ to maxEpoch do
    for $iter \leftarrow 1$ to maxIter do
        sample a mini-batch $\mathcal{D}$ from $X$
        for $\mathbf{x}_i \in \mathcal{D}$ do
            // find nearest prototype
            $o(\mathbf{v}_i) \leftarrow \arg\max_{o \in O_c} sim(\mathbf{v}_i, o)$
            $\mathcal{B}_i^{pos} \leftarrow$ Eq. (10)  // select positives
            $\omega_{ij} \leftarrow$ Eq. (11)  // calculate weight
            $\mathcal{B}_i^{neg} \leftarrow$ Eq. (13)  // select negatives
            minimize $\mathcal{L} \leftarrow$ Eq. (14)  // network updating
        end
    end
    for $c \leftarrow 1$ to $C$ do
        // Merging two clusters into one group
        merging $X_c^i$ and $X_c^j \leftarrow$ Eq. (8)
        update $o_c^i$ and $o_c^j$
    end
end
```

$$\begin{array}{l} \displaystyle\sum_{o_k \in O_c} p(\mathbf{x}_i, o_k; \theta) = \sum_{o_k \in O_c} Q(o_k) b \\ \displaystyle\sum_{o_k \in O_c} p(\mathbf{x}_i, o_k; \theta) = b \end{array} . \tag{18}$$

Then, we can derive

$$Q(o_k) = \frac{p(\mathbf{x}_i, o_k; \theta)}{\sum_{o_k \in O_c} p(\mathbf{x}_i, o_k; \theta)} = \frac{p(\mathbf{x}_i, o_k; \theta)}{p(\mathbf{x}_i; \theta)} = p(o_k; \mathbf{x}_i, \theta), \tag{19}$$

where $Q(o_k)$ is the posterior prototype (fine-grained cluster) probabilitiy. In BuCSFR, for a query representation $\mathbf{v}_i$ of $\mathbf{x}_i$, its associated prototype is given by $o(\mathbf{x}_i) = \arg\max_{o_k \in O_c} sim(\mathbf{v}_i, o_k)$, where $sim(\cdot)$ is the cosine similarity function. Thus, Eq. (19) is calculated by

$p(o_k; \mathbf{x}_i, \theta) = \mathbb{I}(o(\mathbf{x}_i) = o_k)$. If $\mathbf{x}_i$ belongs to the cluster represented by the prototype o_k, then $\mathbb{I}(o(\mathbf{x}_i) = o_k) = 1$; otherwise, $\mathbb{I}(o(\mathbf{x}_i) = o_k) = 0$. The analysis in this subsection corresponds to the BuC module, in which the clusters are nodes of the constructed dendrogram. And the prototypes are updated by merging semantically similar clusters.

6.2. M-step in BuCSFR

In this step, o_k is fixed and can be treated as a constant. Ignoring the constant $-\sum_{i=1}^{N_c}\sum_{o_k \in O_c} Q(o_k)\log Q(o_k)$ in Eq. (17), the objective function can be reformulated to maximize the lower bound of Eq. (17):

$$\begin{aligned} &\max_\theta \sum_{i=1}^{N_c}\sum_{o_k \in O_c} Q(o_k)\log p(\mathbf{x}_i, o_k; \theta), \\ =&\max_\theta \sum_{i=1}^{N_c}\sum_{o_k \in O_c} p(o_k; \mathbf{x}_i, \theta)\log p(\mathbf{x}_i, o_k; \theta), \\ =&\max_\theta \sum_{i=1}^{N_c}\sum_{o_k \in O_c} \mathbb{I}(\mathbf{x}_i \in o_k)\log p(\mathbf{x}_i, o_k; \theta). \end{aligned} \tag{20}$$

Following that, since no prior information is provided, we assume a uniform prior over prototypes, $p(o_k; \theta) = 1/M_c$, then

$$p(\mathbf{x}_i, o_k; \theta) = p(o_k; \theta)p(\mathbf{x}_i; o_k, \theta) = \frac{1}{M_c}p(\mathbf{x}_i; o_k, \theta). \tag{21}$$

This assumption is reasonable because a uniform distribution makes the inter-prototype difference maximized, indicating a more separable feature distribution. Then, we assume that the feature distribution around each prototype is an isotropic Gaussian, leading to

$$p(\mathbf{x}_i; o_k, \theta) \approx \frac{\exp\left(-\frac{(\mathbf{v}_i - o_s)^2}{2\sigma_s^2}\right)}{\sum_{k=1}^{M_c}\exp\left(-\frac{(\mathbf{v}_i - o_k)^2}{2\sigma_k^2}\right)}, \tag{22}$$

where $\mathbf{x}_i \in o_s$. If ℓ_2-normalization is applied to $\mathbf{v}$ and o, then $(\mathbf{v} - o)^2 = 2 - 2\mathbf{v}\cdot o$. By integrating Eqs. (16), (17), (20), (21), and (22), the objective function can be reformulated as

$$\theta^* = \arg\max_\theta - \sum_{i=1}^{N_c}\log\frac{\exp(\mathbf{v}_i \cdot o_s/\phi_s)}{\sum_{k=1}^{M_c}\exp(\mathbf{v}_i \cdot o_k/\phi_k)}, \tag{23}$$

where $\phi \propto \sigma^2$. Considering that the prototype o_k is the average of each cluster $\{\mathbf{v}_i|\mathbf{v}_i \in o_k\}_{i=1}^{N_c^k}$, Eq. (23) effectively forces the samples within one cluster to be distributed densely while pushing different clusters apart. This conclusion can be extended to other coarse classes. The process in this subsection aligns with the SFR module in the paper, where SFR selects high-quality positives and negatives to optimize the objective function $\mathcal{L}$ in Eq. (14) and learn representations with both local alignment and inter-fine-grained-class separation, then achieve fine-grained visual recognition.

In summary, we derive the theoretical support for our method from the perspective of the EM algorithm. BuCSFR alternatively optimizes BuC and SFR during training, aligning with the E-Step and M-Step of the EM algorithm, respectively.

6.3. Convergence of Our Method

In this subsection, we prove the convergence of BuCSFR. Let

$$\begin{aligned} H(\theta) &= \sum_{i=1}^{N_c}\log p(\mathbf{x}_i;\theta) = \sum_{i=1}^{N_c}\log\sum_{o_k \in O_c} p(\mathbf{x}_i, o_k; \theta), \\ &= \sum_{i=1}^{N_c}\log\sum_{o_k \in O_c} Q(o_k)\frac{p(\mathbf{x}_i, o_k;\theta)}{Q(o_k)}, \\ &\geq \sum_{i=1}^{N_c}\sum_{o_k \in O_c} Q(o_k)\log\frac{p(\mathbf{x}_i, o_k;\theta)}{Q(o_k)}. \end{aligned} \tag{24}$$

As detailed in Eq. (17), the inequality holds with equality when $Q(o_k) = p(o_k; \mathbf{x}_i, \theta)$ (refer to Eq. (19)). Then, at the t-th E-step, $Q^t(o_k) = p(o_k; \mathbf{x}_i, \theta^t)$ can be estimated by

$$H(\theta^t) = \sum_{i=1}^{N_c}\sum_{o_k \in O_c} Q^t(o_k)\log\frac{p(\mathbf{x}_i, o_k;\theta)}{Q^t(o_k)}. \tag{25}$$

And at the t-th M-step, keeping $Q^t(o_k) = p(o_k; \mathbf{x}_i, \theta^t)$ constant, the network parameters θ is optimized by maximizing Eq. (25), thus

$$\begin{aligned} H(\theta^{t+1}) &\geq \sum_{i=1}^{N_c}\sum_{o_k \in O_c} Q^t(o_k)\log\frac{p(\mathbf{x}_i, o_k;\theta^{t+1})}{Q^t(o_k)}, \\ &\geq \sum_{i=1}^{N_c}\sum_{o_k \in O_c} Q^t(o_k)\log\frac{p(\mathbf{x}_i, o_k;\theta^{t})}{Q^t(o_k)}, \\ &= H(\theta^t). \end{aligned} \tag{26}$$

This equation indicates that with alternatively optimizing the BuC and SFR modules of BuCSFR, $H(\theta)$ increases monotonically. Therefore, the BuCSFR converges to an (local) optimal solution.

7. Dataset Details

ImageNet-1K contains 1,000 classes. For fairness, we follow the setup described in [7], utilizing a downsampled version of ImageNet-1K where each image has the resolution of 32×32. Due to the lack of official coarse labels, we

Table 7. The hierarchy of ImageNet-1K based on WordNet.

Level	**Classes**	**Number**
Coarse	Invertebrate, Domestic animal, Bird, Mammal, Reptile/Aquatic vertebrate, Device, Vehicle, Container, Instrument, Artifact, Clothing, Others	12
Fine-grained	stingray, jay, bulbul, vulture, jellyfish, flamingo, Yorkshire terrier, . . . , eft, terrapin, agama, ptarmigan, Blenheim spaniel, jacamar, isopod, fiddler crab	1000

Table 8. Classes of seven taxonomic levels in iNaturalist-2019.

Level	**Classes**	**Number**
Kingdom	Animalia, Fungi, Plantae	3
Phylum	Arthropoda, Basidiomycota, Chordata, Tracheophyta	4
Class	Agaricomycetes, Amphibia, Aves, Insecta, Liliopsida, Magnoliopsida, Pinopsida, Polypodiopsida, Reptilia	9
Order	Accipitriformes, Agaricales, Anura, Asparagales, Asterales, Brassicales, Caryophyllales, Charadriiformes, Coleoptera, Cornales, Dipsacales, Ericales, Fabales, Fagales, Gentianales, Geraniales, Hymenoptera, Lamiales, Lepidoptera, Liliales, Malpighiales, Myrtales, Odonata, Oxalidales, Passeriformes, Pinales, Poales, Polypodiales, Ranunculales, Rosales, Sapindales, Saxifragales, Solanales, Squamata	34
Family	Accipitridae, Amanitaceae, Apidae, Apocynaceae, . . . , Viburnaceae, Violaceae, Viperidae, Vireonidae	57
Genus	Acer, Amanita, Anemone, Argia, Artemisia, . . . , Veronica, Viburnum, Viola, Vireo, Yucca	72
Species	Acer campestre, Acer circinatum, Acer floridanum, Acer ginnala, . . . , Viola adunca, Viburnum tinus, Yucca pallida, Yucca rupicola, Yucca schidigera, Yucca treculeana	1010

construct them based on the WordNet [6] hierarchy, organizing the entire dataset into 12 coarse classes that align with the setting in [7]: 'Invertebrate', 'Domestic animal', 'Bird', 'Mammal', 'Reptile/Aquatic vertebrate', 'Device', 'Vehicle', 'Container', 'Instrument', 'Artifact', 'Clothing' and 'Others', as detailed in Tab. 7. It can be observed that the dataset consists of two semantic levels.

iNaturalist-2019 offers 7 granularity levels that follow the biological taxonomy: Kingdom (3 classes), Phylum (4 classes), Class (9 classes), Order (34 classes), Family (57 classes), Genus (72 classes) and Species (1,010 classes). The top taxonomic level, 'Kingdom', comprises three subclasses: 'Animalia', 'Fungi', and 'Plantae', which represent the most coarse-grained categories of the creatures in nature, with significant differences between them. At the next level, 'Phylum', all creatures are categorized into four subclasses: 'Arthropoda', 'Basidiomycota', 'Chordata', and 'Tracheophyta', with differences among them being less pronounced than those in 'Kingdom'. Similarly, at the 'Species' level, all samples are categorized into 1,010 categories with the finest granularity, such as 'Salvia dorrii', 'Salvia apiana', and 'Yucca elata', which convey very subtle inter-category differences. The aforementioned seven taxonomic levels and their corresponding biological categories form the hierarchical semantic structure of the iNaturalist-2019 dataset, as illustrated in Tab. 8.

Table 9. Results (%) of FALCON, DeepDPM and BuCSFR on four datasets. The best results are in bold.

Metrics	R@1	R@2	R@5	R@10	kNN@5	kNN@10
Dataset			CIFAR10toy			
FALCON [20]	46.81	64.28	83.96	92.98	53.56	55.46
DeepDPM [22]	31.50	39.54	52.36	65.78	31.42	30.88
Ours	**93.21**	**96.10**	**98.10**	**99.18**	**94.60**	**94.79**
Dataset			CIFAR100			
FALCON [20]	38.88	50.41	65.57	75.43	48.30	49.27
DeepDPM [22]	23.55	28.99	38.93	48.35	28.19	28.96
Ours	**73.83**	**81.59**	**88.93**	**93.14**	**78.84**	**79.17**
Dataset			FGVC-Aircraft			
FALCON [20]	35.61	46.35	62.26	73.69	40.71	41.07
DeepDPM [22]	13.99	16.24	21.28	28.15	12.67	13.39
Ours	**57.97**	**69.25**	**82.81**	**90.07**	**60.82**	**61.45**
Dataset			ImageNet-1K			
FALCON [20]	12.67	17.99	27.20	35.98	18.46	20.35
DeepDPM [22]	12.40	13.70	16.82	21.54	11.45	11.55
Ours	**23.38**	**32.09**	**44.87**	**54.95**	**33.37**	**36.37**

8. Additional Experiments and Results

8.1. Comparison with FALCON and DeepDPM

An emerging study, FALCON [20], addresses the fine-grained class discovery task by learning the mapping relationship between coarse and fine-grained classes. Another

Table 10. Results(%) of all methods based on ViT.

Dataset	CIFAR100				FGVC-Aircraft			
Backbone	ViT-small		ViT-base		ViT-small		ViT-base	
Metrics	R@1	K@5	R@1	K@5	R@1	K@5	R@1	K@5
CoIns [38]	72.46	78.78	77.47	82.82	43.53	49.74	44.43	51.07
Grafit [27]	74.46	79.45	78.20	83.19	42.54	48.33	42.63	48.60
MaskCon [7]	73.03	78.14	76.31	81.27	29.49	33.87	29.97	34.44
Δ_{SOTA}	6.30	4.95	5.49	3.29	2.88	0.58	6.10	1.83
Ours	**80.76**	**84.40**	**83.69**	**86.48**	**46.41**	**50.32**	**50.53**	**52.90**

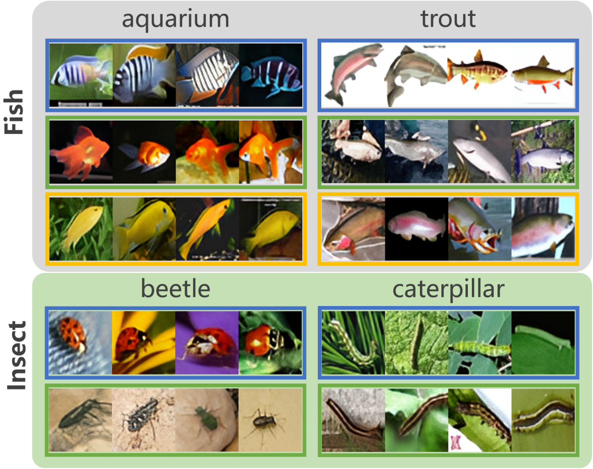


Figure 5. Finer-grained classes discovered within the coarse classes *Fish* and *Insect* in CIFAR100. Each colored box denotes a finer-grained class discovered by BuCSFR.

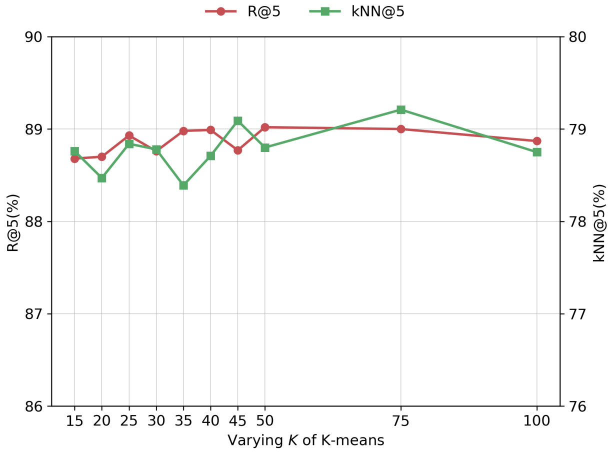


Figure 6. R@5 and kNN@5 $w.r.t$ varying K of K-means on CIFAR100.

new deep clustering approach, DeepDPM [22], does not predefine the number of clusters, and incorporate it with a coarse classification loss. Since their tasks share some similarities with ours, we also perform a comparative evaluation against them. The results are reported in Tab. 9.

Initially, we followed the experimental setup provided in the code of FALCON, where the network parameters were initialized from a pretrained model obtained through time-consuming self-supervised learning on the same dataset. The results we obtained align with those reported in their paper. However, since pretrained models on other datasets were not publicly available, we used a common model pretrained on ImageNet-1K for BuCSFR and all other competing methods. Under this condition, we observed a decline in FALCON's performance, rendering it less competitive. We attribute this to FALCON's reliance on the top-down learning paradigm, see Eq. (3), which makes its performance heavily dependent on the network parameter initialization. DeepDPM also shows a significant performance degradation, especially, on the FGVC-Aircraft dataset. We attribute its limited performance to the fact that the Dirichlet Process Gaussian Mixture Model-based clustering in DeepDPM is not suitable for learning fine-grained representations, thus struggling to distinguish between fine-grained classes.

8.2. Results across Different Architecture

To assess the architectural robustness of our method, we replaced the ResNet50 backbone with a Vision Transformer (ViT) and evaluated all competing methods. The MoCo v3 framework is adapted for the ViT implementation, and the results are reported in Tab. 10. Notably, since LCR inherently relies on convolutional layers, it is incompatible with ViT and thus excluded from comparison. It can be observed that BuCSFR outperforms all other methods, showcasing its strong superiority with the ViT architecture. And an interesting phenomenon emerges when compared with ResNet50 in the main paper. For all methods, performance improves on the CIFAR100 but degrades on the FGVC-Aircraft. We argue that this discrepancy stems from the well-known data-hungry nature of ViT. The FGVC-Aircraft dataset, with only 6,667 training images, is likely too small for ViT to be trained effectively, thus resulting in limited performances. In contrast, the larger scale of the CIFAR100 allows all methods to better leverage the representational power of the ViT.

8.3. Coarse-to-fine with Seven Taxonomic Levels

Table 11 showcases Recall@5 and kNN@5 of FALCON, CoIns, Grafit, MaskCon, LCR, and BuCSFR on iNaturalist-2019. It can be observed that our method achieves superior performance across all settings. Notably, MaskCon performs well when trained with coarser granularity labels but achieves suboptimal results when finer granularity labels are used, showing an opposite trend compared to CoIns, Grafit, and LCR. This difference can be attributed to the classification loss in the objective functions of CoIns, Grafit, and LCR. In contrast, our approach leverages the BuC module to construct a dendrogram for selecting representative positives and high-confidence negatives, ensuring robust performance regardless of the granularity level of the training data.

8.4. Fine-grained Class Discovery via Dendrogram

As mentioned in the third contribution in Introduction, BuCSFR has a capability of discovering novel latent classes. The reason is that each node of the constructed dendrogram represents a cluster of semantically similar samples, which enables BuCSFR to locate cluster centers and associated samples in feature space for fine-grained class discovery.

Figure 5 demonstrates that BuCSFR not only identifies the subclasses provided by CIFAR100 but also discovers meaningful finer-grained classes within them. For example, within the *aquarium* subclass, BuCSFR discovers different subspecies, and within *beetle*, it identifies distinct types such as *ladybug* and *aphid*, despite these finer-grained classes are not provided by the dataset. We believe this capability could substantially facilitate biological research. In contrast, other methods, which do not focus on the separable representations, fail on the task of fine-grained class discovery.

8.5. Ablation Study on K-means

Before training, we initialize the dendrogram by using K-means to reduce computing complexity. Figure 6 shows the results with varying K in K-means. The stable results across a wide range of K demonstrate the robustness of our method.

8.6. Visualization

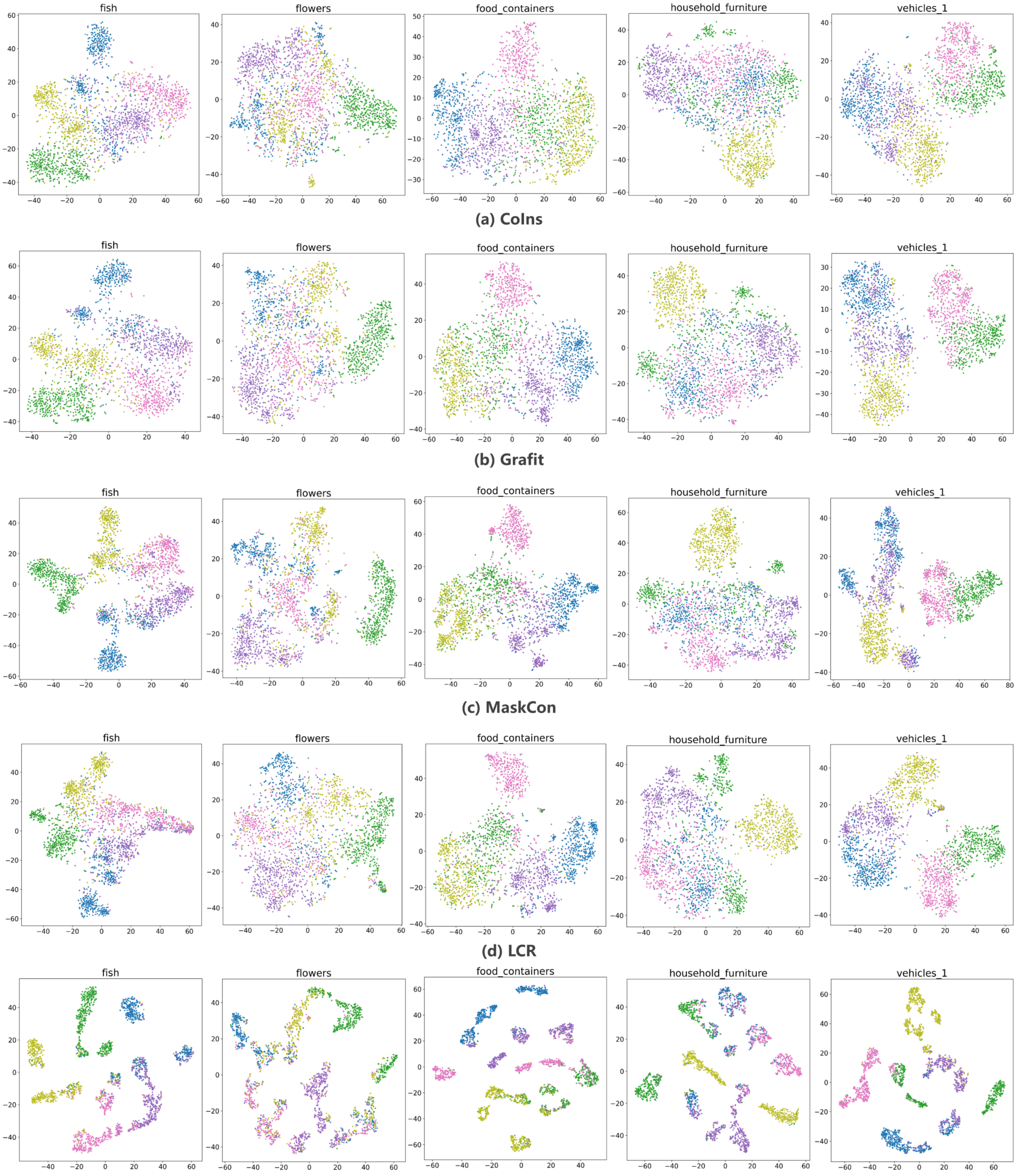
For a better view, Fig. 7 visualizes the learned features of five coarse classes in CIFAR100. One can observe BuCSFR produces more compact clusters with distinct boundaries, indicating it can learn more separable fine-grained representations. In addition, we illustrate the dendrogram inferred by BuCSFR in 'Plantae' of iNaturalist-2019, as shown in Fig. 8. Although the discovered multiple semantic levels do not exactly align with the taxonomy, it is meaningful by offering a unique perspective on the biological world.

9. Discussion

As discussed in the Conclusion of the main paper, there is not always a match between visual and semantic similarity. In fact, our method targets scenarios lacking fine-grained semantic labels, and leverages visual similarity to discover novel knowledge. [23] presents a tourist-recommendation scenario in which the algorithm must automatically identifies hotspots from a large collection of tourism photos that annotated simply with the destination, e.g., "Tower Bridge" (semantic label). Although sharing the common semantic "Tower Bridge", each hotspot exhibits unique temporal, locational, and view-dependent attractions of *Tower Bridge* to travers. In this scenario, coarse labels are readily available, whereas fine-grained semantics of these unique hotspots remain to be discovered, thus requiring visual similarity-driven fine grained representation learning. Similarly, in biodiversity recognition, emerging species often lack semantic labels and can only be distinguished visually. A case in Fig. 5 shows that previously unlabeled finer-grained classes can be identified via visual similarity, facilitating taxonomic research. In the future work, we will explore the integration of knowledge from Large Language Models to inject high-level semantics into representation learning. This would allow the model to not only discover visually knowledge but also to align them with the human cognition, leading to fine-grained representations that are both visual-driven and semantically rich.

Table 11. R@5 / kNN@5 (%) on iNaturalist-2019 dataset with seven semantic levels.

	Train(→)	kingdom	phylum	class	order	family	genus	species
	(↓)Test	3	4	9	34	57	72	1010
FALCON [20]	kingdom	96.76 / 91.30	-	-	-	-	-	-
	phylum	94.65 / 85.32	95.10 / 88.13	-	-	-	-	-
	class	88.08 / 75.04	89.21 / 77.74	90.29 / 80.48	-	-	-	-
	order	54.52 / 32.82	57.62 / 37.13	60.91 / 41.28	69.21 / 52.07	-	-	-
	family	45.40 / 23.88	49.71 / 28.32	53.37 / 32.30	62.70 / 43.11	63.62 / 45.54	-	-
	genus	40.91 / 21.02	45.20 / 24.92	49.43 / 28.98	59.29 / 39.59	60.44 / 42.03	61.54 / 43.14	-
	species	11.00 / 4.64	13.71 / 5.81	16.49 / 7.77	25.06 / 13.96	26.78 / 15.21	28.03 / 16.55	39.63 / 21.27
CoIns [38]	kingdom	98.74 / 97.30	-	-	-	-	-	-
	phylum	97.99 / 95.59	98.33 / 96.59	-	-	-	-	-
	class	95.65 / 90.16	96.09 / 91.18	97.30 / 94.62	-	-	-	-
	order	79.79 / 64.54	80.62 / 65.84	84.16 / 71.65	93.11 / 87.56	-	-	-
	family	75.13 / 57.79	76.11 / 59.04	80.21 / 64.82	90.36 / 81.00	92.09 / 86.09	-	-
	genus	72.63 / 54.56	73.66 / 55.70	77.87 / 61.57	88.78 / 77.62	90.87 / 82.74	91.65 / 85.58	-
	species	35.40 / 21.10	35.85 / 21.31	40.15 / 24.02	53.89 / 35.77	57.97 / 38.60	60.47 / 40.78	74.64 / 60.51
Grafit [27]	kingdom	98.93 / 97.26	-	-	-	-	-	-
	phylum	98.24 / 95.90	98.45 / 96.74	-	-	-	-	-
	class	96.17 / 91.03	96.53 / 92.05	97.36 / 94.74	-	-	-	-
	order	81.57 / 67.69	81.94 / 68.14	85.34 / 73.47	93.12 / 87.71	-	-	-
	family	77.23 / 61.05	77.52 / 61.43	81.60 / 67.07	90.52 / 81.61	91.86 / 85.75	-	-
	genus	74.82 / 57.80	75.25 / 58.21	79.40 / 63.73	89.05 / 78.76	90.56 / 82.66	91.20 / 84.86	-
	species	37.40 / 23.03	37.37 / 22.62	42.34 / 26.41	55.32 / 38.30	58.21 / 40.07	60.37 / 41.95	73.92 / 59.26
MaskCon [7]	kingdom	98.78 / 97.56	-	-	-	-	-	-
	phylum	98.29 / 96.87	98.38 / 97.08	-	-	-	-	-
	class	96.79 / 93.59	96.92 / 93.75	97.35 / 95.08	-	-	-	-
	order	86.67 / 76.28	87.11 / 76.89	88.36 / 79.24	92.97 / 87.98	-	-	-
	family	83.55 / 70.53	83.97 / 71.42	85.47 / 73.92	90.66 / 82.72	91.62 / 85.36	-	-
	genus	81.64 / 67.11	82.23 / 68.19	83.72 / 70.53	89.35 / 79.80	90.44 / 82.59	91.06 / 84.34	-
	species	45.04 / 27.60	45.71 / 28.47	47.53 / 29.65	56.53 / 38.96	58.51 / 39.84	60.13 / 41.46	69.33 / 52.90
LCR [24]	kingdom	98.96 / 97.92	-	-	-	-	-	-
	phylum	98.30 / 95.48	98.81 / 97.69	-	-	-	-	-
	class	95.65 / 89.42	96.54 / 91.89	98.04 / 95.86	-	-	-	-
	order	76.74 / 60.47	78.44 / 62.88	82.11 / 68.34	93.70 / 88.67	-	-	-
	family	70.91 / 52.30	73.17 / 54.52	77.30 / 60.10	90.63 / 81.08	92.42 / 87.10	-	-
	genus	67.72 / 48.65	69.99 / 50.90	74.26 / 56.36	88.73 / 77.27	91.04 / 83.27	91.64 / 85.77	-
	species	30.07 / 18.03	31.30 / 18.65	35.82 / 22.15	53.19 / 36.86	58.00 / 39.99	59.62 / 41.94	72.63 / 60.25
BuCSFR	kingdom	98.99 / 98.14	-	-	-	-	-	-
	phylum	98.56 / 97.38	98.88 / 97.92	-	-	-	-	-
	class	97.40 / 94.30	97.79 / 94.82	98.04 / 96.26	-	-	-	-
	order	87.91 / 78.74	88.53 / 79.21	89.40 / 81.35	94.15 / 89.91	-	-	-
	family	85.59 / 72.69	85.93 / 73.66	87.28 / 75.88	91.21 / 84.47	93.04 / 88.17	-	-
	genus	84.02 / 70.83	84.69 / 71.91	85.25 / 73.17	90.39 / 81.78	91.63 / 84.19	92.81 / 87.58	-
	species	47.57 / 30.97	48.24 / 31.34	50.08 / 33.10	59.46 / 41.13	61.73 / 42.51	63.47 / 44.60	76.84 / 62.92



(a) CoIns

(b) Grafit

(c) MaskCon

(d) LCR

(e) Ours

Figure 7. T-SNE visualization of learned representation on CIFAR100. Different colors represent the corresponding fine-grained classes.

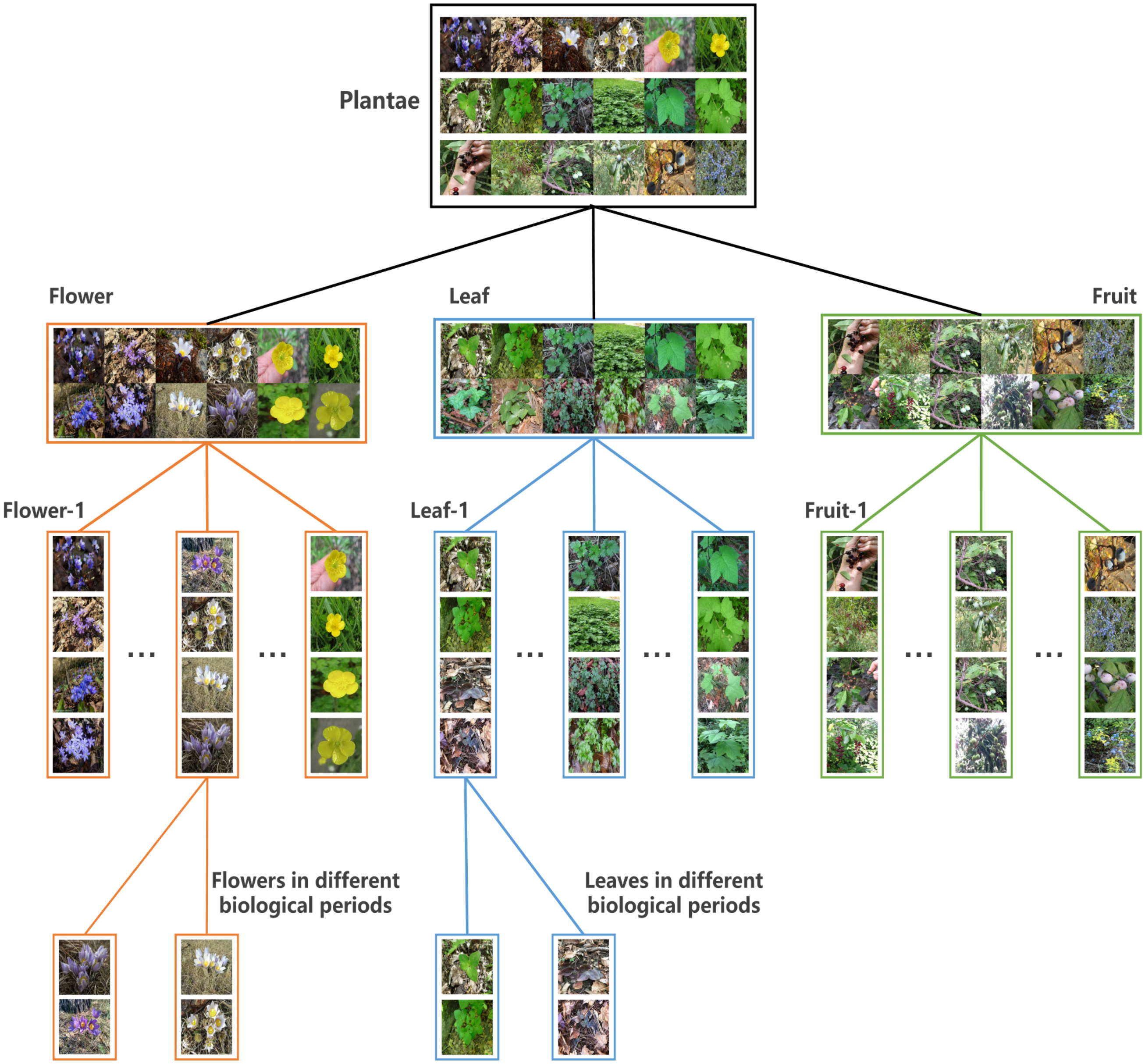

Plantae
Flower
Leaf
Fruit
Flower-1
Leaf-1
Fruit-1
...
...
...
...
...
...
Flowers in different biological periods
Leaves in different biological periods

Figure 8. The dendrogram inferred by BuCSFR in ‘Plantae’ of iNaturalist-2019.