A Appendix

A.1 Algorithm and implementation details

A.1.1 The expandable model

Algorithm 1 Expand Collective Model

1: **Input:** Dataset $\mathcal{D}^{col} = (\mathcal{D}_1, \ldots, \mathcal{D}_t)$, learning rate α, t moment of collective model parameters θ^t_{col} and network F^t_{col}, the newly expanded network module $\mathcal{M}$ of size $\mathbf{s}$, $\mathcal{L}$ comes from Eq. (1) in Sec. 4.1
2: **for** $t = 1, \ldots, T$ **do**
3: **if** $t = 1$ **then**
4: **while** $\mathcal{L}$ has not converged **do**
5: Update the network weights using Eq. (2) in Sec. 4.1
6: **end while**
7: **else**
8: $\mathcal{M} = \text{RANDOMDUPLICATE}(F^{t-1}_{col}, \mathbf{s})$
9: $F^t_{col} = \text{CONSTRUCTMODEL}(F^{t-1}_{col}, \mathcal{M})$
10: **while** $\mathcal{L}$ has not converged **do**
11: Update the network weights using Eq. (2) in Sec. 4.1
12: **end while**
13: **end if**
14: **end for**

The collective model handles sequential tasks in the open-world. Here we deliberate that each task is a multi-class classification. In the beginning, we train the parameters of the basic collective model on the first task (L3-L6 in Algo. 1). When deepening the network, it arbitrarily duplicates the network of the previous layer to the following layer. Similarly, when the network is widened, a replica of the current layer can be randomly picked and then spliced together (L8-L12 in Algo. 1).

A.1.2 Data division

As shown in Table 1 and 2, the collective model uses 60 classes as base classes, 16 classes as open-world classes, and the remaining 20 classes as novel classes, to ensure no data-interaction between the two models.

A.1.3 Model capacity

As shown in Figure 1, the collective model extends from the left to the network scale on the right. The individual model inherits the last three layers of network modules to reconstruct a lightweight model.

Table 1: Data division of cifar100.

Base classes	lobster, dinosaur, snake, plate, can, chair, oak_tree, turtle, tulip, pear, mushroom, lizard, kangaroo, castle, skunk, apple, rabbit, seal, pine_tree, spider, bowl, hamster, trout, orange, streetcar, clock, tiger, couch, bottle, palm_tree, train, bee, road, porcupine, caterpillar, sunflower, house, aquarium_fish, raccoon, keyboard, tank, forest, wolf, boy, mountain, ray, cup, bridge, lawn_mower, lion, dolphin, shrew, bus, elephant, orchid, bear, cockroach, cloud, girl, willow_tree, squirrel, skyscraper, mouse, possum, cattle
Open-world classes	beetle, beaver, butterfly, camel, cattle, crab, crocodile, flatfish, lamp, maple_tree, motorcycle, otter, sea, shark, television, tractor
Novel classes	baby, bed, bicycle, chimpanzee, fox, leopard, man, pickup_truck, plain, poppy, rocket, rose, snail, sweet_pepper, table, telephone, wardrobe, whale, woman, worm

Table 2: Data division of ImageNet100.

Base classes	'n02091831', 'n02747177', 'n03527444', 'n03337140', 'n07697537', 'n13133613', 'n04251144', 'n01532829', 'n04443257', 'n03908618', 'n01558993', 'n03047690', 'n02687172', 'n02795169', 'n04515003', 'n01770081', 'n04067472', 'n03924679', 'n02089867', 'n02108551', 'n02165456', 'n02108089', 'n13054560', 'n02823428', 'n04596742', 'n03062245', 'n03476684', 'n01704323', 'n04604644', 'n02105505', 'n02113712', 'n07747607', 'n06794110', 'n04243546', 'n04389033', 'n02606052', 'n02108915', 'n03017168', 'n04258138', 'n03854065', 'n02457408', 'n02120079', 'n03207743', 'n04275548', 'n04612504', 'n03347037', 'n01843383', 'n04296562', 'n04435653', 'n04509417', 'n03998194', 'n09246464', 'n01749939', 'n03888605', 'n07584110', 'n03400231', 'n02966193', 'n01910747', 'n03676483', 'n03838899', 'n02074367', 'n03220513', 'n02111277', 'n02101006'
Open-world classes	'n02971356', 'n03535780', 'n03770439', 'n03075370', 'n03417042', 'n03584254', 'n09256479', 'n02114548', 'n02091244', 'n01855672', 'n02950826', 'n03773504', 'n02174001', 'n02138441', 'n02981792', 'n03980874'
Novel classes	'n02110341', 'n01981276', 'n03146219', 'n04149813', 'n02219486', 'n02871525', 'n02110063', 'n04418357', 'n02116738', 'n03544143', 'n04522168', 'n02129165', 'n07613480', 'n03775546', 'n03127925', 'n03272010', 'n01930112', 'n02443484', 'n04146614', 'n02099601'

A.2 Derivation of Theorem 1

Lemma 1. *(Summarized from Sec.2 of [2]) For any $\delta > 0$ the following inequality holds with probability at least $1-\delta$ (over the training samples $\{D_1, \ldots, D_t\}$) for posterior distribution $\mathcal{Q}$*

$$er(\mathcal{Q}) - \widehat{er}(\mathcal{Q}) \leq \frac{1}{\sqrt{\bar{m}}}\left(\frac{1}{t}KL((\mathcal{Q}, Q^t)\|(\mathcal{P}, P^t) - \frac{1}{t}\log\frac{\delta}{2} + \frac{1}{8}\right) + O\left(\frac{1}{\sqrt{t}}\right) \quad (1)$$

where $(\mathcal{Q}, Q^n) = \mathcal{Q} \times \prod_{i=1}^{n} Q_i$ denotes the distribution in which we first sample P according to $\mathcal{Q}$ and then use it and the data D_t to produce a posterior Q_i for each task. $(\mathcal{P}, P^n) = \mathcal{P} \times \prod_{i=1}^{n} P$ denotes the distribution in which we sample P according to $\mathcal{P}$ and use it as a posterior for all tasks. $\bar{m} = \frac{1}{n}\sum_{i=1}^{n} m_i$ is the

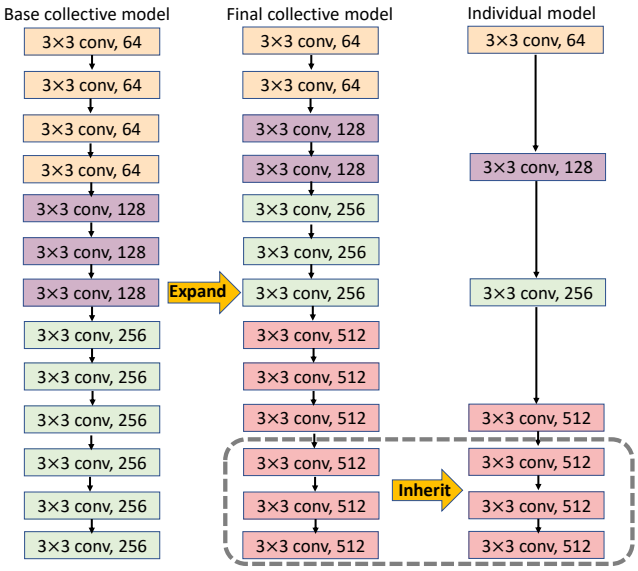


Figure 1: **Example network architectures for the collective-individual paradigm.** The collective model extends from the left to the network scale on the right. The individual model inherits the last three layers of network modules to reconstruct a lightweight model.

harmonic mean of the sample sizes. The convergence rate of the generalization error is $O(\frac{1}{\sqrt{\bar{m}}})$. Therefore, as $\bar{m} \to \infty$, the error bound approaches constant. Here, we present the theoretical analysis for learngene as a shared low-dimensional subspace and get the following tighter bound based on PAC-Bayesian [2]

Theorem 1. *For the prior P, we choose a Gaussian with zero mean and variance σI_k. The posterior Q, is a shifted Gaussian with variance σI_k and mean w_Q in the same subspace. The following bound holds for t tasks*

$$er(M) - \widehat{er}(M) < \frac{c_2}{2\sigma t\sqrt{\bar{m}}} + c_1 \tag{2}$$

where c_1 is a constant, c_2 is irrelevant with $\bar{m}$ and $D(I_k, M)$ is a special case of Langevin distribution.

Proof. When we use learngene to transfer representation, the following holds:

$$er(M) - \widehat{er}(M) \leq \frac{1}{2\sigma t\sqrt{\bar{m}}} \sum_{i=1}^{t} \mathop{\mathbb{E}}_{B\sim D(I_k,M)} \|w_i(B)\|^2 + c_1 \tag{3}$$

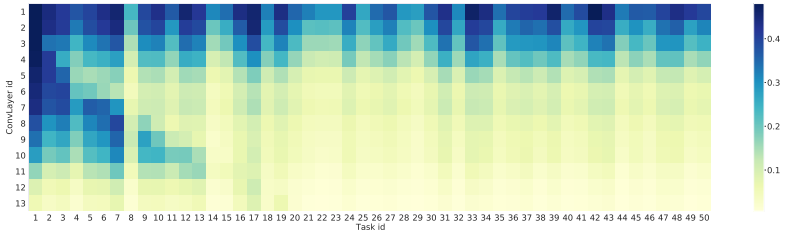
where B is the representation of the selected subspace. Futhermore, we can write

$$\begin{aligned} w_i(B) &= \frac{C}{m_i}(I_k + \frac{C}{m_i} B^\top X_i X_i^\top B)^{-1} B^\top X_i Y_i \\ &= C(m_i I_k + C B^\top X_i X_i^\top B)^{-1} B^\top X_i Y_i \\ &< C(I_k + C B^\top X_i X_i^\top B)^{-1} B^\top X_i Y_i \\ &= c_2' \end{aligned}$$

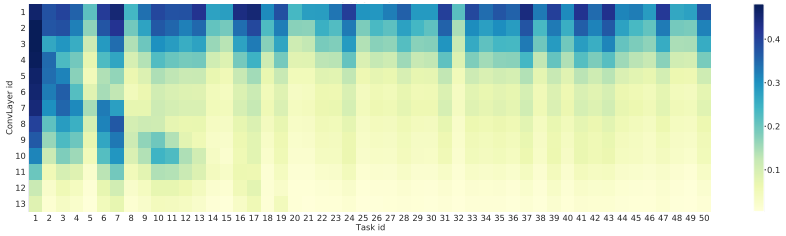
where c_2' is irrelevant with $\bar{m}$. In this way, $er(M) - \widehat{er}(M) < \frac{c_2}{2\sigma t\sqrt{\bar{m}}} + c_1$, which concludes the proof.

A.3 Experiments

A.3.1 Inheriting learngene that represents meta-knowledge



(a) Gradient trends on CUB-200



(b) Gradient trends on CIFAR-100

Figure 2: **Gradient trends on CUB-200 and CIFAR-100.** The expandable collective model handles 50 sequential tasks sampling from CUB-200 and CIFAR-100.

As shown by Figure A.3.1, the parameter changes of the deeper network layers tend to be stable with the task order on CUB-200 [1] and CIFAR-100, which are judged as the expected learngene. Furthermore, We hypothesize that, the top layer is more inclined to learn the semantic-level information among tasks, i.e., meta-knowledge, after training numerous tasks in time series.

References

[1] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge J. Belongie. Visual recognition with humans in the loop. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, volume 6314 of *Lecture Notes in Computer Science*, pages 438–451. Springer, 2010.

[2] Anastasia Pentina and Christoph H. Lampert. A pac-bayesian bound for lifelong learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 991–999. JMLR.org, 2014.