A. Supplementary Material

This supplementary material presents detailed optimization pipeline of the proposed HFC model in Section A.1, analysis of incremental tasks in Section A.2, and stable convergence analysis in Section A.3. The code is available at https://github.com/JiahuaDong/HFC.

A.1. Optimization Pipeline of Our HFC Model

The optimization of our HFC to learn new classes consecutively is summarized in **Algorithm** 1. Θ^t along with learnable $\mathbf{E}_0^t$ are optimized via $\mathcal{L}_{\rm CE}$ in Eq. (5) for the first task, and trained via $\mathcal{L}_{\rm obj}$ in Eq. (14) when $t \geq 2$. After optimizing Θ^t in the t-th task, we store Θ^t as the frozen old model Θ^{t-1} to perform the gradient-balanced relation distillation loss $\mathcal{L}_{\rm RD}$ in Eq. (13) for the next learning task. Meanwhile, the exemplar memory $\mathcal{M}$ is updated via following iCaRL [40] to replay only few samples (*i.e.*, $\frac{|\mathcal{M}|}{K^o+K^t}$) of each learned class for the next task, or following PODNet [17] to store 20 exemplars for each learned class.

Algorithm 1: Optimization pipeline of HFC.

Input: The consecutive tasks $\mathcal{T} = \{\mathcal{T}^t\}_{t=1}^T$, $\{\alpha_1, \alpha_2\}$;
for $t = 1, 2, \cdots, T$ **do**
  while not converged **do**
    if $t = 1$ **then**
      Obtain a mini-batch $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}_{i=1}^b \in \mathcal{T}^t$;
      Update Θ^t, $\mathbf{E}_0^t$ via optimizing $\mathcal{L}_{\rm CE}$ in Eq. (5);
    else
      Obtain a mini-batch $\{\mathbf{x}_i^t, \mathbf{y}_i^t\}_{i=1}^b \in \mathcal{T}^t \cup \mathcal{M}$;
      Update Θ^t, $\mathbf{E}_0^t$ via optimizing $\mathcal{L}_{\rm obj}$ in Eq. (14);
    end if
  end while
  Update memory $\mathcal{M}$ via following [40];
  Store Θ^t as Θ^{t-1}, and $\mathbf{E}_0^t$ for next initialization.
end for

A.2. Task-wise Performance Comparison

In order to comprehensively evaluate the effectiveness of our proposed HFC model, top-1 accuracy is employed to compare our model with other state-of-the-art CIL methods. As shown in Tabs. 8–9, our model achieves significant improvement over other methods by $1.4\% \sim 13.2\%$ in terms of top-1 average accuracy. Specially, compared to the previous methods, the performance gap increases as the number of stages grows. It reflects that our HFC model is more effective to deal with challenging tasks and achieve solid improvement for all incremental tasks. In addition, Our HFC model achieves superior performance compared to the baseline methods for most of the tasks, which shows that our method can effectively address the catastrophic forgetting of old classes from both representation and gradient aspects.

As shown in Tab. 10, we present comparison experiments in terms of top-1 accuracy between our HFC model and other CIL baselines on CIFAR-100 [33], when the number of tasks is 10. We apply two plug-and-play losses (*i.e.*, $\mathcal{L}_{\rm FC}$ and $\mathcal{L}_{\rm RD}$) to existing CIL methods. The experimental results show that the proposed plug-and-play losses help current CIL methods to overcome heterogeneous forgetting from a gradient perspective. More importantly, as shown in Tab. 10, our proposed losses help the existing state-of-the-art CIL methods significantly improve by $0.5\% \sim 6.7\%$ in terms of top-1 averaged accuracy on CIFAR-100 [33]. It verifies stable generalization of our HFC model to address heterogeneous catastrophic forgetting.

A.3. Qualitative Analysis of Convergence

As presented in Fig. 5, we introduce some qualitative convergence results in terms of top-1 accuracy for each incremental task on ImageNet-100 [10]. From these curves, we can observe that the accuracy in each increment task increases gradually until convergence as the training process. It shows that our proposed HFC model has robust convergence performance for each incremental task. More importantly, the convergence speed is fast via optimizing the proposed HFC model with only few training epochs. It also illustrates the effectiveness of our proposed model to address heterogeneous catastrophic forgetting on old classes.

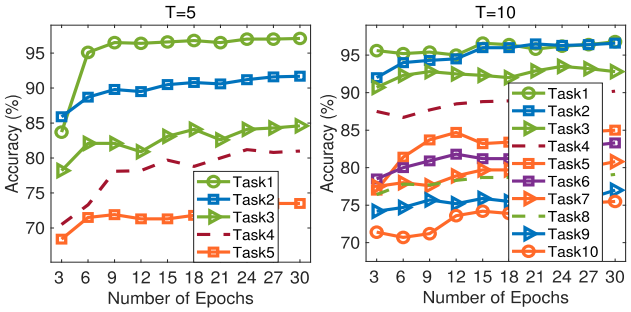

Figure 5. Convergence curves on ImageNet-100 [10], when we set $\mathcal{M} = 2000, \mathcal{B} = 0\%$ for $T = \{5, 10\}$.

Table 8. Performance of each incremental task on ImageNet-100 [10] in terms of top-1 accuracy, when $\mathcal{M} = 2000$, $T = 10$ and $\mathcal{B} = 0\%$.

Comparison Methods	Backbone	#Params	$\mathcal{B} = 0\%$										Avg.	Imp.
			10	20	30	40	50	60	70	80	90	100		
iCaRL [40] (CVPR'2017)	ViT-Base	85.10M	96.4	95.4	91.5	86.4	81.3	78.6	76.6	74.8	72.8	70.4	82.4	⇑3.0
BiC [50] (CVPR'2019)	ViT-Base	85.10M	**97.8**	96.1	89.5	86.8	79.0	76.8	73.5	72.7	67.3	62.9	80.2	⇑5.2
PODNet [17] (ECCV'2020)	ViT-Base	85.10M	**97.8**	96.1	89.5	86.9	79.1	76.9	73.6	71.9	66.4	63.1	80.1	⇑5.3
SS-IL [2] (ICCV'2021)	ViT-Base	85.10M	96.6	96.2	91.1	86.9	81.2	77.7	76.1	75.0	68.9	67.0	81.7	⇑3.7
PODNet [17] + CSCCT [4] (ECCV'2022)	ViT-Base	85.10M	96.8	94.0	90.3	85.6	79.0	75.6	75.1	74.8	70.4	68.1	81.0	⇑4.4
FOSTER [47] (ECCV'2022)	ViT-Base	85.10M	96.4	96.4	92.0	88.0	83.5	81.1	79.1	77.2	74.8	73.0	84.0	⇑1.4
AFC [32] (CVPR'2022)	ViT-Base	85.10M	96.8	96.7	89.5	86.3	79.7	77.5	75.5	74.7	70.0	65.6	81.2	⇑4.2
DyTox [18] (CVPR'2022)	ViT-Base	85.10M	96.4	95.7	92.2	88.2	83.0	79.5	77.0	75.3	70.4	66.8	83.4	⇑2.0
HFC (Ours)	ViT-Base	85.10M	96.8	**96.6**	**92.6**	**89.2**	**85.1**	**82.7**	**80.5**	**79.0**	**76.8**	**75.5**	**85.4**	–
Upper Bound	ViT-Base	85.10M	–	–	–	–	–	–	–	–	–	–	86.3	–

Table 9. Performance of each incremental task on ImageNet-1000 [10] in terms of top-1 accuracy, when $\mathcal{M} = 2000$, $T = 10$ and $\mathcal{B} = 0\%$.

Comparison Methods	Backbone	#Params	$\mathcal{B} = 0\%$										Avg.	Imp.
			10	20	30	40	50	60	70	80	90	100		
iCaRL [40] (CVPR'2017)	ViT-Base	85.10M	90.7	83.5	78.2	75.2	72.4	70.0	67.1	64.3	61.8	61.0	72.4	⇑4.0
BiC [50] (CVPR'2019)	ViT-Base	85.10M	90.7	84.4	76.4	71.9	66.6	61.1	57.4	53.5	50.8	47.4	66.0	⇑10.4
PODNet [17] (ECCV'2020)	ViT-Base	85.10M	84.4	79.1	75.8	72.7	71.4	69.0	67.1	65.8	64.7	63.5	71.3	⇑5.1
SS-IL [2] (ICCV'2021)	ViT-Base	85.10M	85.5	85.3	78.4	78.0	76.0	73.2	71.3	69.8	66.5	63.7	74.8	⇑1.6
PODNet [17] + CSCCT [4] (ECCV'2022)	ViT-Base	85.10M	90.3	67.6	62.7	62.2	60.9	60.0	58.8	57.7	56.4	55.7	63.2	⇑13.2
FOSTER [47] (ECCV'2022)	ViT-Base	85.10M	90.9	84.8	79.7	77.0	74.2	72.1	69.4	67.3	64.5	63.4	74.3	⇑2.1
AFC [32] (CVPR'2022)	ViT-Base	85.10M	90.5	86.3	80.1	76.3	72.2	67.8	64.6	61.6	58.3	56.0	71.4	⇑5.0
DyTox [18] (CVPR'2022)	ViT-Base	85.10M	**91.5**	**88.1**	**83.1**	**79.8**	76.3	72.2	68.6	65.7	62.4	59.4	74.7	⇑1.7
HFC (Ours)	ViT-Base	85.10M	90.7	85.9	81.6	79.3	**76.6**	**74.4**	**71.7**	**69.9**	**67.9**	**66.0**	**76.4**	–
Upper Bound	ViT-Base	85.10M	–	–	–	–	–	–	–	–	–	–	86.3	–

Table 10. Performance on CIFAR-100 [33] ($T = 10$), when we apply **Ours**‡ into existing distillation-based CIL methods and set $\mathcal{M} = 2000, \mathcal{B} = 0\%$. **Ours**‡ denotes the proposed plug-and-play losses $\mathcal{L}_{\mathrm{FC}}$ and $\mathcal{L}_{\mathrm{RD}}$.

Comparison Methods	Backbone	#Params	$\mathcal{B} = 0\%$										Avg.	Imp.
			10	20	30	40	50	60	70	80	90	100		
iCaRL [40] (CVPR'2017)	ResNet-32	0.46M	84.2	77.3	73.0	68.4	63.5	60.5	58.3	54.5	52.3	47.4	63.9	⇑1.8
iCaRL [40] + **Ours**‡	ResNet-32	0.46M	**84.2**	**78.6**	**74.9**	**69.8**	**65.6**	**62.5**	**60.3**	**55.4**	**54.2**	**51.4**	**65.7**	–
BiC [50] (CVPR'2019)	ResNet-32	0.46M	88.9	70.1	56.0	45.7	43.9	46.6	41.6	39.9	39.1	35.8	50.8	⇑3.0
BiC [50] + **Ours**‡	ResNet-32	0.46M	**88.9**	**72.1**	**58.1**	**49.0**	**47.7**	**49.4**	**44.8**	**44.0**	**43.5**	**40.5**	**53.8**	–
PODNet [17] (ECCV'2020)	ResNet-32	0.46M	85.7	73.0	63.8	56.4	53.7	48.3	45.2	42.0	39.0	38.1	54.5	⇑5.5
PODNet [17] + **Ours**‡	ResNet-32	0.46M	**89.5**	**77.8**	**69.6**	**62.6**	**57.2**	**55.6**	**51.1**	**49.6**	**45.9**	**41.2**	**60.0**	–
SS-IL [2] (ICCV'2021)	ResNet-32	0.46M	84.7	67.3	64.5	60.2	57.4	54.5	53.5	50.9	49.8	47.6	59.0	⇑5.9
SS-IL [2] + **Ours**‡	ResNet-32	0.46M	**86.0**	**78.2**	**74.2**	**68.9**	**64.8**	**61.7**	**59.6**	**54.3**	**52.9**	**48.0**	**64.9**	–
PODNet [17] + CSCCT [4] (ECCV'2022)	ResNet-32	0.46M	85.7	73.0	63.7	56.3	53.6	48.1	44.9	41.7	38.7	37.8	54.3	⇑4.2
PODNet [17] + CSCCT [4] + **Ours**‡	ResNet-32	0.46M	**85.7**	**75.0**	**67.7**	**60.9**	**57.9**	**53.2**	**50.4**	**46.9**	**44.0**	**43.5**	**58.5**	–
FOSTER [47] (ECCV'2022)	ResNet-32	0.46M	91.9	82.0	75.7	71.0	67.8	65.4	61.8	59.5	58.4	53.7	68.7	⇑1.2
FOSTER [47] + **Ours**‡	ResNet-32	0.46M	**91.9**	**82.2**	**77.2**	**71.5**	**68.4**	**67.4**	**63.3**	**61.6**	**59.5**	**56.0**	**69.9**	–
AFC [32] (CVPR'2022)	ResNet-32	0.46M	**90.9**	76.5	65.7	57.4	52.8	51.8	47.4	45.5	42.8	40.2	57.1	⇑6.7
AFC [32] + **Ours**‡	ResNet-32	0.46M	88.7	**81.2**	**72.3**	**65.9**	**62.6**	**60.2**	**56.2**	**53.4**	**50.2**	**46.8**	**63.8**	–
Upper Bound	ResNet-32	0.46M	–	–	–	–	–	–	–	–	–	–	76.6	–
DyTox [18] (CVPR'2022)	ViT-Tiny	10.71M	**91.9**	85.0	80.0	74.8	73.4	71.4	68.2	65.9	63.9	61.0	73.5	⇑0.5
DyTox [18] + **Ours**‡	ViT-Tiny	10.71M	91.7	**86.0**	**80.5**	**74.8**	**73.7**	**72.0**	**68.5**	**66.7**	**64.7**	**61.1**	**74.0**	–
Upper Bound	ViT-Tiny	10.71M	–	–	–	–	–	–	–	–	–	–	76.1	–