Appendix of "Learning with Fantasy: Semantic-Aware Virtual Contrastive Constraint for Few-Shot Class-Incremental Learning"

This Appendix first provides detailed experimental results on the three benchmarks in Sec. A. The performance with different fantasy methods is reported in Sec. B. Discussion on different contrastive methods and inference methods can be found in Sec. C and Sec. D, respectively. The training algorithm is elaborated in Sec. E. We finally discuss the limitations in Sec. F.

A. Detailed Results

A.1. Comparison with State of The Arts

In Sec. 4.2 Fig. 4, we provide the comparison with the state-of-the-art methods on three benchmarks in the form of line charts. Here we present the detailed numbers and compare our SAVC with more methods, including: naïve baseline that directly finetunes the model with limited data as 'finetune', classical CIL methods, *i.e.*, iCaRL [11], EEIL [1], Rebalancing [7], incremental-trainable FSCIL methods, *i.e.*, TOPIC [13], FSLL+SS [9], IDLVQ-C [2], and incremental-frozen FSCIL methods, *i.e.*, SPPR [21], F2M [12], CEC [16], MetaFSCIL [5], FACT [18], LIMIT [19]. We first give a detailed introduction about these methods, then show the detailed results in Tabs. 1 to 3.

Table 1. Comparison with SOTA methods on CIFAR100 dataset for few-shot incremental learning. $*$: Performances reported by [13]. Δ_{last}: Relative improvements of the last session compared to the Finetune baseline.

Method	Acc. in each session (%) ↑									Δ_{last}
	0	1	2	3	4	5	6	7	8	
Finetune* [13]	64.10	39.61	15.37	9.80	6.67	3.80	3.70	3.14	2.65	–
iCaRL* [11]	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	+11.08
EEIL* [1]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	+13.20
Rebalancing* [7]	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	+10.89
TOPIC* [13]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	+26.72
FSLL+SS [9]	66.76	55.52	52.20	49.17	46.23	44.64	43.07	41.20	39.57	+36.92
SPPR [21]	63.97	65.86	61.31	57.60	53.39	50.93	48.27	45.36	43.32	+40.67
F2M [12]	64.71	62.05	59.01	55.58	52.55	49.96	48.08	46.28	44.67	+42.02
CEC [16]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	+46.49
MetaFSCIL [5]	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	+47.32
FACT [18]	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	+49.45
LIMIT [19]	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	+48.58
SAVC (Ours)	**78.77**	**73.31**	**69.31**	**64.93**	**61.70**	**59.25**	**57.13**	**55.19**	**53.12**	**+50.47**

- **Finetune.** It directly finetunes the model with cross-entropy (CE) loss in incremental sessions, and cannot strike the stability and plasticity trade-off well.
- **iCaRL [11].** It maintains an "episodic memory" of the exemplars, which enables to incrementally learn new classes without forgetting old classes.
- **EEIL [1].** It considers an end-to-end framework, combining a distillation loss to retain old knowledge and a cross-entropy term to learn the new classes.

Table 2. Comparison with SOTA methods on *mini*ImageNet dataset for few-shot incremental learning. $*$: Performances reported by [13]. Δ_{last}: Relative improvements of the last session compared to the Finetune baseline.

Method	Acc. in each session (%) ↑									Δ_{last}
	0	1	2	3	4	5	6	7	8	
Finetune* [13]	61.31	27.22	16.37	6.08	2.54	1.56	1.93	2.60	1.40	–
iCaRL* [11]	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	+15.81
EEIL* [1]	61.31	46.58	44.00	37.29	33.14	27.12	24.1	21.57	19.58	+18.18
Rebalancing* [7]	61.31	47.80	39.31	31.91	25.68	21.35	18.67	17.24	14.17	+12.77
TOPIC* [13]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	+23.02
FSLL+SS [9]	68.85	63.14	59.24	55.23	52.24	49.65	47.74	45.23	43.92	+42.52
IDLVQ-C [2]	64.77	59.87	55.93	52.62	49.88	47.55	44.83	43.14	41.84	+40.44
SPPR [21]	61.45	63.80	59.53	55.53	52.50	49.60	46.69	43.79	41.92	+40.52
F2M [12]	67.28	63.80	60.38	57.06	54.08	51.39	48.82	46.58	44.65	+43.25
CEC [16]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	+46.23
MetaFSCIL [5]	72.04	67.94	63.77	60.29	57.58	55.16	52.90	50.79	49.19	+47.79
FACT [18]	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	+49.09
LIMIT [19]	72.32	68.47	64.30	60.78	57.95	55.07	52.70	50.72	49.19	+47.79
SAVC (Ours)	**81.12**	**76.14**	**72.43**	**68.92**	**66.48**	**62.95**	**59.92**	**58.39**	**57.11**	**+55.71**

Table 3. Comparison with SOTA methods on CUB200 dataset for few-shot incremental learning. $*$: Performances reported by [13]. Δ_{last}: Relative improvements of the last session compared to the Finetune baseline.

Method	Acc. in each session (%) ↑											Δ_{last}
	0	1	2	3	4	5	6	7	8	9	10	
Finetune* [13]	68.68	43.70	25.05	17.72	18.08	16.95	15.10	10.06	8.93	8.93	8.47	–
iCaRL* [11]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	+12.69
EEIL* [1]	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	+13.64
Rebalancing* [7]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	+11.40
TOPIC* [13]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	+17.81
FSLL+SS [9]	75.63	71.81	68.16	64.32	62.61	60.10	58.82	58.70	56.45	56.41	55.82	+47.35
IDLVQ-C [2]	77.37	74.72	70.28	67.13	65.34	63.52	62.10	61.54	59.04	58.68	57.81	+49.34
SPPR [21]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	+28.86
F2M [12]	81.07	78.16	75.57	72.89	70.86	68.17	67.01	65.26	63.36	61.76	60.26	+51.79
CEC [16]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	+43.81
MetaFSCIL [5]	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64	+44.17
FACT [18]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	+48.47
LIMIT [19]	75.89	73.55	71.99	68.14	67.42	63.61	62.40	61.35	59.91	58.66	57.41	+48.94
SAVC (Ours)	**81.85**	**77.92**	**74.95**	**70.21**	**69.96**	**67.02**	**66.16**	**65.30**	**63.84**	**63.15**	**62.50**	**+54.03**

- **Rebalancing [7].** It learns a unified classifier to address the imbalance problem, which consists of three components, *i.e.*, cosine normalization, less-forget constraint and inter-class separation.
- **TOPIC [13].** It mitigates forgetting by utilizing a Neural Gas (NG) network to preserve the topology of the feature manifold formed by different classes.
- **FSLL+SS [9].** It selects very few unimportant parameters to update for training every new set of classes, while explicitly maximizing old and new classes separation to prevent them from overlapping with each other.
- **IDLVQ-C [2].** It develops a unified incremental deep learning vector quantization framework and mitigates catastrophic forgetting by intra-class variance regularization, less forgetting constraints and calibration factors.

- **SPPR [21].** It employs a random episode selection strategy and a self-promoted prototype refinement mechanism, which equips the features with extensibility to incremental tasks.
- **F2M [12].** It searches the flat local minima of the base training objective function so that the model can be updated within the flat region on incremental tasks.
- **CEC [16].** It trains a graph model as a classifier adaptation module to propagate context information between old and new prototypes. The adaptation module is trained by sampling pseudo incremental learning tasks in the base session.
- **MetaFSCIL [5].** It proposes to sample sequences of incremental tasks and optimize a meta-objective guided by a bi-directional guided modulation, so that the model is capable of fast adapting to novel classes without forgetting.
- **FACT [18].** It pre-assigns multiple virtual prototypes and generates virtual instances via instance mixture in the embedding space, to reserve spaces for incoming new classes.
- **LIMIT [19].** It encourages the model to learn multi-phase incremental tasks synthesized in the base session. Besides, a transformer is used to calibrate the old and new prototypes into the same semantic scale.

A.2. Performance Measure of Base Classes and New Classes

In this work, we compare the Top 1 accuracy in the last session, *i.e.* $\mathcal{A}_{\mathrm{T}}$, to measure the final performance in all classes. [16] has defined a performance dropping rate (PD) to measure the absolute accuracy drop in the last session, *i.e.*, $\mathrm{PD} = \mathcal{A}_0 - \mathcal{A}_{\mathrm{T}}$. Here, we decompose the accuracy in the last session $\mathcal{A}_{\mathrm{T}}$ into the accuracy of base classes $\mathcal{A}_{\mathrm{TB}}$ and new classes $\mathcal{A}_{\mathrm{TN}}$, define $\mathrm{PD}_{\mathrm{B}} = \mathcal{A}_0 - \mathcal{A}_{\mathrm{TB}}$ to quantitatively measure the forgetting phenomena (because there are only base classes in the base session), and directly compare $\mathcal{A}_{\mathrm{TN}}$ to measure the adaptation to novel classes. The results are reported in Tabs. 4 to 6.

From the results, we observe that our SAVC significantly improves the accuracy of both base classes and novel classes, and boosts the overall performance more on novel classes compared with the CE baseline. On the one hand, Our SAVC outperforms other approaches on novel classes ($\mathcal{A}_{\mathrm{TN}}$) by a large margin, which verifies its effectiveness on novel class adaptation. On the other hand, SAVC acquires the best accuracy on base classes in the last session ($\mathcal{A}_{\mathrm{TB}}$), and comparable base performance dropping rate (PD_{B}) with other SOTA methods, which shows that our SAVC could maintain base class separation well and mitigate the catastrophic forgetting problem effectively.

Table 4. Performance measure of base and new classes on CIFAR100 dataset. †: Results from our implementation by the official published code. −: Results not reported in [18] or [19]. The improvement or degradation related to CE is shown in brackets.

Method	$\mathcal{A}_0$	$\mathcal{A}_{\mathrm{T}}$	$\mathcal{A}_{\mathrm{TB}}$	$\mathcal{A}_{\mathrm{TN}}$	$\mathrm{PD}_{\mathrm{B}} \downarrow$
CE	73.00	46.47	67.92	14.30	**5.08**
CEC† [16]	73.07 (+0.07%)	49.10 (+2.63%)	67.90 (-0.02%)	20.90 (+6.60%)	5.17 (+0.09%)
FACT [18]	74.60 (+1.60%)	52.10 (+5.63%)	–	–	–
LIMIT [19]	73.81 (+0.81%)	51.23 (+4.76%)	–	–	–
SAVC (Ours)	**78.77** (+5.77%)	**53.12** (+6.65%)	**73.07** (+5.15%)	**23.20** (+8.90%)	5.70 (+0.62%)

Table 5. Performance measure of base and new classes on *mini*ImageNet dataset. †: Results from our implementation by the officially published code. −: Results not reported in [18] or [19]. The improvement or degradation related to CE is shown in brackets.

Method	$\mathcal{A}_0$	$\mathcal{A}_{\mathrm{T}}$	$\mathcal{A}_{\mathrm{TB}}$	$\mathcal{A}_{\mathrm{TN}}$	$\mathrm{PD}_{\mathrm{B}} \downarrow$
CE	70.43	45.80	67.23	13.65	**3.20**
CEC† [16]	72.25 (+1.82%)	47.67 (+1.87%)	67.97 (+0.74%)	17.23 (+3.58%)	4.28 (+1.08%)
FACT [18]	72.56 (+2.13%)	50.49 (+4.69%)	–	–	–
LIMIT [19]	72.32 (+1.89%)	49.19 (+4.69%)	–	–	–
SAVC (Ours)	**81.12** (+10.69%)	**57.11** (+11.31%)	**74.67** (+7.44%)	**30.78** (+17.13%)	6.45 (+3.25%)

Table 6. Performance measure of base and new classes on CUB200 dataset. †: Results from our implementation by the officially published code. The improvement or degradation related to CE is shown in brackets.

Method	$\mathcal{A}_0$	$\mathcal{A}_{\mathrm{T}}$	$\mathcal{A}_{\mathrm{TB}}$	$\mathcal{A}_{\mathrm{TN}}$	$\mathrm{PD}_{\mathrm{B}} \downarrow$
CE	74.42	47.84	69.87	26.35	4.55
CEC† [16]	75.68 (+1.26%)	52.12 (+4.28%)	70.46 (+0.59%)	34.23 (+7.88%)	5.21 (+0.66%)
FACT [18]	75.90 (+1.48%)	56.94 (+9.10%)	73.90 (+4.03%)	40.50 (+14.15%)	**2.00** (-3.55%)
LIMIT [19]	75.89 (+1.47%)	57.41 (+9.57%)	73.60 (+3.73%)	41.80 (+15.45%)	2.29 (-2.26%)
SAVC (Ours)	**81.85** (+7.43%)	**62.50** (+14.66%)	**77.65** (+7.78%)	**47.68** (+21.33%)	4.20 (-0.35%)

B. Discussion on Fantasy Methods

B.1. Discussion on SSL transformations

In Sec. 4.1 Fig. 6a, we have compared three SSL transformation methods and found that the choice of transformation methods is crucial to the success of our algorithm. We adopt more transformation methods and conduct comprehensive experiments on *mini*ImageNet to study their impacts. Experimental results in Tab. 7 indicate that the performance would show little improvement with the limited number of transformations ($M = 2$) or without proper rotation transformations. Meanwhile, the performance can be degraded under too many transformations ($M = 24$).

Table 7. Comparison of different transformation methods on *mini*ImageNet dataset for few-shot incremental learning. Δ_{last}: Relative improvements of the last session compared to the CE baseline. 'ALL' in 'Rotation' denotes using all 0°, 90°, 180°, 270° degrees, and in 'Color permutation' denotes using all RGB, RBG, GRB, GBR, BRG, BGR permutations.

Method	Rotation	Color permutation	Acc. in each session (%) ↑									Δ_{last}
			0	1	2	3	4	5	6	7	8	
CE	–	–	70.27	65.17	61.13	57.80	54.80	51.84	49.14	47.03	45.54	–
3-permutations	0°	RGB, GBR, BRG	76.85	72.31	68.31	64.40	61.60	58.31	55.32	53.35	52.33	+6.79
6-permutations	0°	ALL	77.07	71.95	67.86	64.25	61.31	58.08	55.46	53.68	52.63	+7.09
2-fold rotations	0°, 180°	RGB	77.52	72.32	68.43	65.20	62.86	59.65	56.84	54.87	53.71	+8.17
6-augmentations	0°, 180°	RGB, GBR, BRG	79.42	74.43	70.19	66.71	64.01	60.25	57.51	55.56	54.64	+9.10
24-augmentations	ALL	ALL	80.75	75.37	71.57	67.55	64.96	61.34	58.12	56.31	54.96	+9.42
4-fold rotations	ALL	RGB	80.77	76.19	72.36	69.03	66.60	63.21	60.11	58.25	57.01	+11.47
12-augmentations	ALL	RGB, GBR, BRG	**81.12**	**76.14**	**72.43**	**68.92**	**66.48**	**62.95**	**59.92**	**58.39**	**57.11**	**+11.57**

B.2. Discussion on instance mixture methods

Apart from SSL transformation methods, there are other instance mixture ways to generate virtual classes, such as vanilla mixup [17] and manifold mixup [14] that previous works [18,20] have explored. In contrast to traditional mixup augmentation which has been considered unsuitable for incremental tasks [10], [20] randomly shuffles every mini-batches several times in the training dataset and combines the corresponding two images with the same index but different labels to generate new virtual classes. Hence the $|\mathcal{C}_0|$-class problem in the base task can be extended to a $|\mathcal{C}_0| + |\mathcal{C}_0|(|\mathcal{C}_0| - 1)/2$-class problem, while we generate $|\mathcal{C}_0|(|\mathcal{C}_0| - 1)/2$ virtual classes with fewer samples than the original classes. The mixture coefficient is bounded in the range of $[0.4, 0.6]$, which keeps the virtual samples away from the original samples in visual. At the end of the training of the base stage, these mixup nodes in the classifier would be removed.

We explore the influence of different instance mixture methods, including vanilla mixup [17], manifold mixup [14] and CutMix [15] on *mini*ImageNet dataset. The results are shown in Tab. 8. Obviously, our framework benefits more from SSL transformation than instance mixture, and we suppose the reasons lie in two folds: 1) SSL transformation enlarges the label space by M times, and every derived class has the same number of samples as the original classes. However, instance mixture generates extra $|\mathcal{C}_0|(|\mathcal{C}_0| - 1)/2$ classes with very less samples, which may cause overfitting on the virtual classes. 2) These virtual classes generated by SSL transformation have exact 'fine-grained' semantic meaning and enable a multi-semantic aggregated inference effect. But instance mixture generates virtual classes lacking 'semantic logic' which are unavailable for future inference or generalization.

Table 8. Comparison of different instance mixture methods on *mini*ImageNet dataset for few-shot incremental learning. Δ_{last}: Relative improvements of the last session compared to the CE baseline.

Method	Acc. in each session (%) ↑									Δ_{last}
	0	1	2	3	4	5	6	7	8	
CE	70.27	65.17	61.13	57.80	54.80	51.84	49.14	47.03	45.54	–
Vanilla	78.00	71.63	67.37	63.75	60.69	57.68	54.89	52.54	50.67	+5.13
Manifold	77.35	71.25	67.04	63.52	60.49	57.19	54.64	52.46	50.77	+5.23
CutMix	78.00	72.95	68.71	64.87	61.53	58.18	55.44	53.46	51.59	+6.05
12-augmentations	**81.12**	**76.14**	**72.43**	**68.92**	**66.48**	**62.95**	**59.92**	**58.39**	**57.11**	**+11.57**

C. Discussion on Contrastive Methods

In this section, we give an analysis of different contrastive methods from the gradient perspective. We mainly consider three forms of loss function, *i.e.*, unsupervised contrastive loss, supervised contrastive loss (Sec. 3.2 Eq. (3)) and our semantic-aware virtual contrastive loss (Sec. 3.3 Eq. (8)). They are all built on MoCo [3, 4, 6] framework.

- **Unsupervised contrastive loss.** Given a query embedding $\boldsymbol{q}$, it only regards its key embedding as positives and all in feature queue Q are negatives:

$$\mathcal{L}_{uncont}\left(g;\mathbf{x},\tau,A\right)=-\log\frac{\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_{+}/\tau\right)}{\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_{+}/\tau\right)+\sum\limits_{\boldsymbol{k}_{-}\in N(\mathbf{x})}\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_{-}/\tau\right)}, \tag{1}$$

  where $N(\mathbf{x})=Q$. Then we can obtain the gradient w.r.t. the query sample $\boldsymbol{q}$:

$$\frac{\partial\mathcal{L}_{uncont}}{\partial\boldsymbol{q}}=-\frac{1}{\tau}\left\{(1-p_{k_{+}})\boldsymbol{k}_{+}-\sum_{\boldsymbol{k}_{-}\in N(\mathbf{x})}p_{k_{-}}\boldsymbol{k}_{-}\right\}, \tag{2}$$

  where $p_{k_i}=\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_i/\tau\right)/\sum_{\boldsymbol{k}_i\in A(\mathbf{x})}\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_i/\tau\right)$, $A(\mathbf{x})=P(\mathbf{x})\cup N(\mathbf{x})$ is the embedding pool to select positives and negatives. It indicates that optimizing Eq. (1) would push the query embedding $\boldsymbol{q}$ towards the direction of its key embedding $\boldsymbol{k}$, and away from all embeddings in the feature queue, which would cause unexpected high intra-class entropy.

- **Supervised contrastive loss.** It regards the key embeddings in the feature queue which share the same label as the query sample as additional positives, and we only consider the form that summation is located outside the log [8]:

$$\mathcal{L}_{supcont}\left(g;\mathbf{x},\tau,A\right)=-\frac{1}{|P(\mathbf{x})|}\sum_{\boldsymbol{k}_{+}\in P(\mathbf{x})}\log\frac{\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_{+}/\tau\right)}{\sum\limits_{\boldsymbol{k}_{+}\in P(\mathbf{x})}\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_{+}/\tau\right)+\sum\limits_{\boldsymbol{k}_{-}\in N(\mathbf{x})}\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_{-}/\tau\right)}. \tag{3}$$

  Then we can obtain the gradient w.r.t. the query sample $\boldsymbol{q}$:

$$\frac{\partial\mathcal{L}_{supcont}}{\partial\boldsymbol{q}}=-\frac{1}{\tau}\left\{\left(\frac{1}{|P(\mathbf{x})|}-p_{k_{+}}\right)\boldsymbol{k}_{+}-\sum_{\boldsymbol{k}_{-}\in N(\mathbf{x})}p_{k_{-}}\boldsymbol{k}_{-}\right\}. \tag{4}$$

  Here, $N(\mathbf{x})$ only includes key embeddings belonging to different classes with the query sample $\boldsymbol{q}$ in the queue. Compared to Eq. (2), Eq. (4) corrects the optimization directions which these potential positives in the queue contribute to, that is, pushing $\boldsymbol{q}$ towards rather than away from them. When $|P(\mathbf{x})|=1$, Eq. (3) degrades into Eq. (1).

- **Semantic-aware virtual contrastive loss.** It generates many virtual classes by applying pre-defined SSL transforma-

tions, and enables contrast in finer semantic grains:

$$\mathcal{L}_{savcont}\left(g;\mathbf{x},\tau,A,\mathcal{F}\right)$$
$$=-\frac{1}{M}\sum_{m=1}^{M}\frac{1}{\left|P\left(\mathbf{x}_m\right)\right|}\sum_{\boldsymbol{k}_+\in P(\mathbf{x}_m)}\log\frac{\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_+/\tau\right)}{\sum\limits_{\boldsymbol{k}_+\in P(\mathbf{x}_m)}\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_+/\tau\right)+\sum\limits_{\boldsymbol{k}_-\in N(\mathbf{x}_m)}\exp\left(\boldsymbol{q}^{\mathsf{T}}\boldsymbol{k}_-/\tau\right)+\sum\limits_{\bar{\boldsymbol{k}}_-\in \bar{N}(\mathbf{x}_m)}\exp\left(\boldsymbol{q}^{\mathsf{T}}\bar{\boldsymbol{k}}_-/\tau\right)}. \tag{5}$$

Here, we split the negative set into $N(\mathbf{x}_m)$ which contains key embeddings from the queue sharing different original classes but the same transformation type with $\boldsymbol{q}$, and $\bar{N}(\mathbf{x}_m)$ which contains key embeddings from the queue sharing different transformation type with $\boldsymbol{q}$. Indeed, $\bar{N}(\mathbf{x}_m)$ can be regarded as a complementary hard negative set and plays an important role in the contrastive training process, especially those negatives which share the same original class but different transformation type with $\boldsymbol{q}$. Thus we can obtain the gradient w.r.t. the query sample $\boldsymbol{q}$:

$$\frac{\partial\mathcal{L}_{savccont}}{\partial\boldsymbol{q}}=-\frac{1}{M}\sum_{m=1}^{M}\frac{1}{\tau}\left\{\left(\frac{1}{|P(\mathbf{x}_m)|}-p_{k_+}\right)\boldsymbol{k}_+-\sum_{\boldsymbol{k}_-\in N(\mathbf{x}_m)}p_{k_-}\boldsymbol{k}_--\sum_{\bar{\boldsymbol{k}}_-\in\bar{N}(\mathbf{x}_m)}p_{\bar{k}_-}\bar{\boldsymbol{k}}_-\right\}. \tag{6}$$

Compared to Eq. (4), Eq. (6) adds additional gradients by virtual hard negatives and pushes $\boldsymbol{q}$ away from them. When $M=1$, Eq. (5) degrades into Eq. (3).

Algorithm 1 Semantic-Aware Class Fantasy Training

Input: Base train dataset $\mathcal{D}^0_{train}$, Fantasy set $\mathcal{F}$, Hyperparameters: coefficients α and β
Output: Classification model: ϕ, Query network: g, Key network: g_m, Feature queue and label queue

Randomly initialize ϕ, g, g_m, Feature queue and label queue
for $iter=1,2,...,$ **do**
  Get a mini-batch sample from $\mathcal{D}^0_{train}$: $B=\{(\mathbf{x}_i,\ y_i)\}_{i=1}^{N}$
  Generate virtual classes: $\mathcal{F}(B)=\{(\mathbf{x}_{im},\ y_{im})\}_{i=1,m=1}^{N,M}$
  Calculate classification loss:
  $\mathcal{L}_{cls}(\phi;\mathcal{F}(B))=\frac{1}{|\mathcal{F}(\mathcal{B})|}\sum_{(\mathbf{x}_{im},\ y_{im})\in\mathcal{F}(\mathcal{B})}\mathcal{L}_{ce}\left(\phi\left(\mathbf{x}_{im}\right),y_{im}\right)$
  Generate query and key embeddings:
  $B_q=\{\boldsymbol{q}_{im}\in g(\mathrm{Aug}_q(\mathbf{x}_{im}))|\mathbf{x}_{im}\in\mathcal{F}(B)\}$
  $B_k=\{\boldsymbol{k}_{im}\in g_m(\mathrm{Aug}_k(\mathbf{x}_{im}))|\mathbf{x}_{im}\in\mathcal{F}(B)\}$
  Calculate global and local contrastive loss:
  $\mathcal{L}_{cont}\left(g;B_q,\tau,A\right)=-\frac{1}{|B_q|}\sum_{\boldsymbol{q}_{im}\in B_q}\left\{\frac{1}{|P(\mathbf{x}_{im})|}\sum_{\boldsymbol{k}_+\in P(\mathbf{x}_{im})}\log\frac{\exp\left(\boldsymbol{q}_{im}^{\mathsf{T}}\boldsymbol{k}_+/\tau\right)}{\sum_{\boldsymbol{k}'\in A(\mathbf{x}_{im})}\exp\left(\boldsymbol{q}_{im}^{\mathsf{T}}\boldsymbol{k}'/\tau\right)}\right\}$
  Get the total loss:
  $\mathcal{L}=\mathcal{L}_{cls}+\alpha\mathcal{L}_{cont\_global}+\beta\mathcal{L}_{cont\_local}$
  Obtain derivative and update ϕ and g;
  Momentum update g_m by g
  Update feature and label queue
end
for $class\ c$ in $\mathcal{C}^0$ **do**
  Get all samples in $\mathcal{D}^0_{train}$ belonging to class c: $B_c=\{\mathbf{x}_{c,i}\}_{i=1}^{n_c^0}$
  Generate virtual classes: $\mathcal{F}(B_c)=\{\mathbf{x}_{c,im}\}_{i=1,m=1}^{n_c^0,M}$
  Replace the classifier with prototypes:
  $W_c^0=\left\{\boldsymbol{w}_{cm}^0\in\frac{1}{n_c^0}\sum_{i=1}^{n_c^0}f\left(\mathbf{x}_{c,im}\right)\middle|\mathbf{x}_{c,im}\in\mathcal{F}(B_c)\right\}_{m=1}^{M}$
end

Algorithm 2 Multi-Semantic Aggregated Inference

Input: Base test dataset $\mathcal{D}^0_{test}$, Fantasy set $\mathcal{F}$, Classification model: ϕ
Output: Inference results C

for $j = 1, 2, ...,$ **do**
  Get the mini-batch test sample in order from $\mathcal{D}^0_{test}$: $B^j = \{\mathbf{x}_i\}_{i=1}^N$
  Generate virtual classes: $\mathcal{F}(B^j) = \{\mathbf{x}_{im}\}_{i=1,m=1}^{N,M}$
  Initialize the prediction result $P^j = [\mathbf{0}] \in \mathbb{R}^{|\mathcal{C}^0| \times N}$
  for $m = 1, 2, ..., M$ **do**
    Form a normalized conditional sample and a normalized prototype subset:
    $\mathbf{X}_m = \left\{\mathbf{x}_{im} \in \mathcal{F}(B^j)\right\}_{i=1}^N, \ \widetilde{\mathbf{X}}_m = \mathrm{Norm}(\mathbf{X}_m, 0)$
    $W^0_m = \left\{\boldsymbol{w}^0_{cm} \in W^0\right\}_{c=1}^{|\mathcal{C}^0|}, \ \widetilde{W}^0_m = \mathrm{Norm}(W^0_m, 0)$
    Calculate the inference logits from m-th fantasy view:
    $P_m = \widetilde{W}^{0\,\top}_m f(\widetilde{\mathbf{X}}_m)$
  end
  Aggregate the inference logits and results:
  $P^j = \frac{1}{M}\sum_{m=1}^M P_m$
  $C^j = \arg\max(P^j, 0)$
end

D. Discussion on Inference Methods

As shown in Sec. 3.3 Eq. (9), we ensemble inference results from different views to boost performance. Owing to our specific multi-view learning scheme, the inference should also follow the same rules. When degrading our multi-semantic aggregated inference method to the original inference method, the last session accuracy changes from **57.11%** to **52.79%** on *mini*ImageNet dataset. It further demonstrates that virtual classes act as semantic knowledge providers which encourage extensive learning of different semantics for better generalization.

E. Pseudo-code of SAVC

We show the pseudo-code of our SAVC method of training and inference part respectively. The semantic-aware class fantasy training part is concluded in Algorithm 1, and the multi-semantic aggregated inference part is concluded in Algorithm 2.

F. Limitation

Although we have developed our SAVC based on the observation that base class separation facilitates novel class generalization, the underlying reason lies in it remains unknown. In addition, there may be other better ways to generate virtual classes by other newly-proposed advanced augmentation techniques. Besides, exploring the class fantasy in feature spaces is also a promising direction. We leave them for our future work.

References

[1] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 1, 2

[2] Kuilin Chen and Chi-Guhn Lee. Incremental few-shot learning via vector quantization in deep embedded space. In *ICLR*, 2020. 1, 2

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 5

[4] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 5

[5] Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metafscil: A meta-learning approach for few-shot class incremental learning. In *CVPR*, pages 14166–14175, 2022. 1, 2, 3

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 5

[7] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019. 1, 2

[8] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, pages 18661–18673, 2020. 5

[9] Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. In *AAAI*, volume 35, pages 2337–2345, 2021. 1, 2

[10] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for class incremental learning. In *CVPR*, pages 3513–3522, 2021. 4

[11] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 1, 2

[12] Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. In *NeurIPS*, pages 6747–6761, 2021. 1, 2, 3

[13] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, pages 12183–12192, 2020. 1, 2

[14] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447. PMLR, 2019. 4

[15] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 4

[16] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *CVPR*, pages 12455–12464, 2021. 1, 2, 3, 4

[17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4

[18] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *CVPR*, pages 9046–9056, 2022. 1, 2, 3, 4

[19] Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *TPAMI*, 2022. 1, 2, 3, 4

[20] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. In *NeurIPS*, pages 14306–14318, 2021. 4

[21] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *CVPR*, pages 6801–6810, 2021. 1, 2, 3