Neural Collapse-Driven, Uncertainty-Aware Framework for Few-Shot Class-Incremental Learning

Sungwon Woo,
Multimedia System Laboratory
Department of Artificial Intelligence

SOGANG UNIVERSITY

Table of Contents

Introduction

1.1 Background

- Class-Incremental Learning (CIL)
  - Deep neural networks have achieved remarkable success across a wide range of domains
  - However, when trained on streaming data, they face challenges such as ***catastrophic forgetting***
  - CIL aims to ***continuously learn new classes*** while preserving knowledge of previous classes [1]

- Few-Shot Class-Incremental Learning (FSCIL)
  - A more practical yet challenging subset of CIL, where ***new classes emerge with only a few samples*** [2]
  - Specifically, the FSCIL task consists of…

    a) A base session with sufficient training data

    b) Multiple incremental sessions with an extremely limited number of samples

1.1 Background: Research Trends in Few-Shot Class-Incremental Learning

- Stability-Focused (Forward compatible) FSCIL
  - Aims for better separation of *base* classes to **prepare for possible novel classes and future updates** during base session training
  - Adopt an ***incremental-frozen framework*** *and* ***a prototype-based classifier structure***
  - Most approaches such as FACT [3] and NC-FSCIL [4] primarily focus on forward compatibility

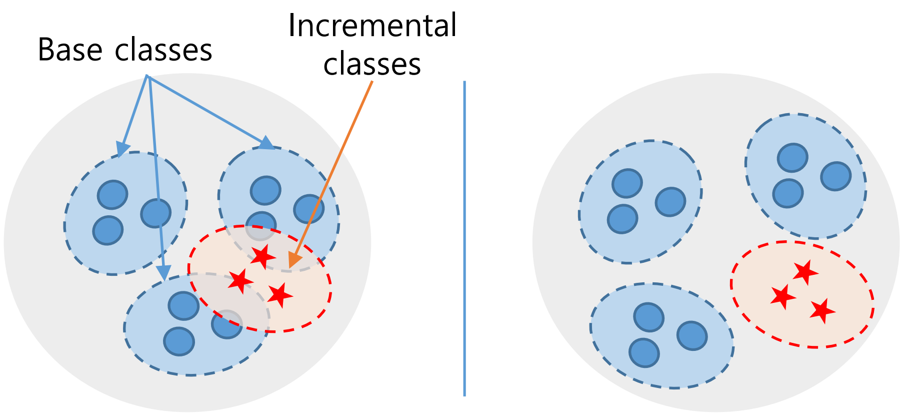


Figure 1. Diagram of Stability-Focused Approach

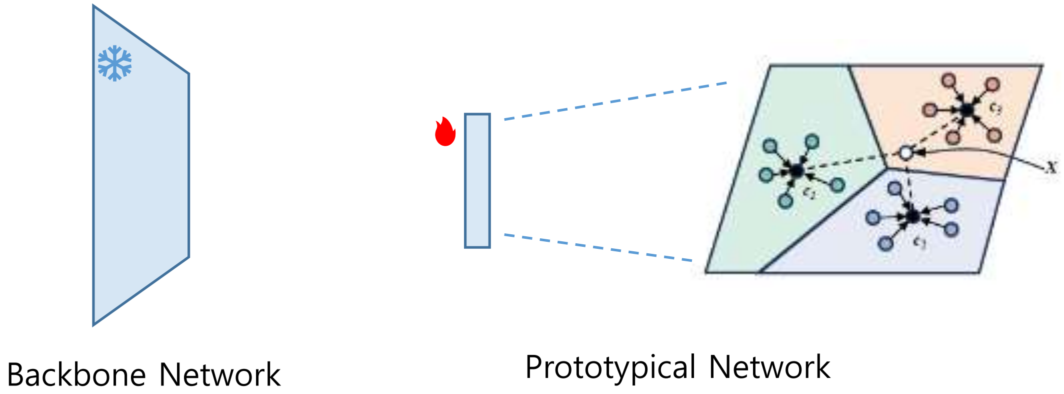


Figure 2. Diagram of Incremental-frozen, Prototype-based classifier framework

1.2 Motivation

- Performance Imbalance Problem
  - ***Accuracy is dominated by base classes***, whereas incremental classes show lower performance
- Asymmetrical misclassification problem
  - Incremental class samples are ***frequently misclassified as base classes, but not vice versa***

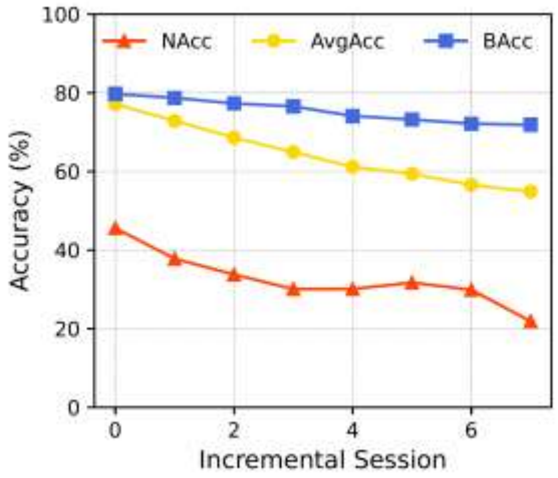


Figure 3. Accuracy gap across sessions

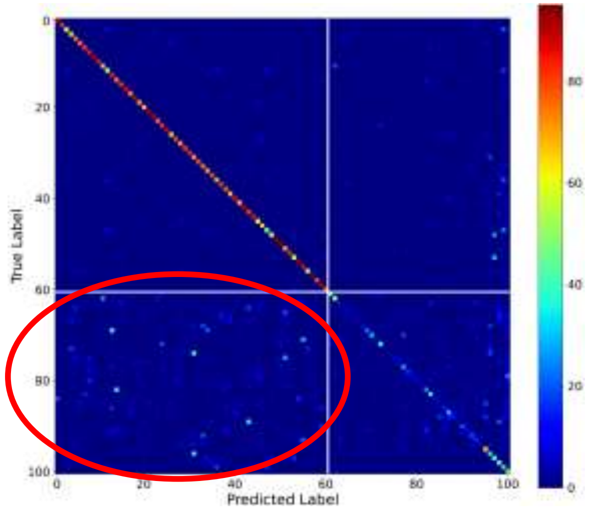


Figure 4. Confusion matrix at the final session

Introduction

1.2 Motivation: Need for Uncertainty Quantification

- Uncertainty Quantification for FSCIL
  - Incremental class samples represents near decision boundary
  - Base classes have abundant dataset, whereas incremental classes have only limited dataset
  - The model should be able to ***quantify its uncertainty for each class representation***

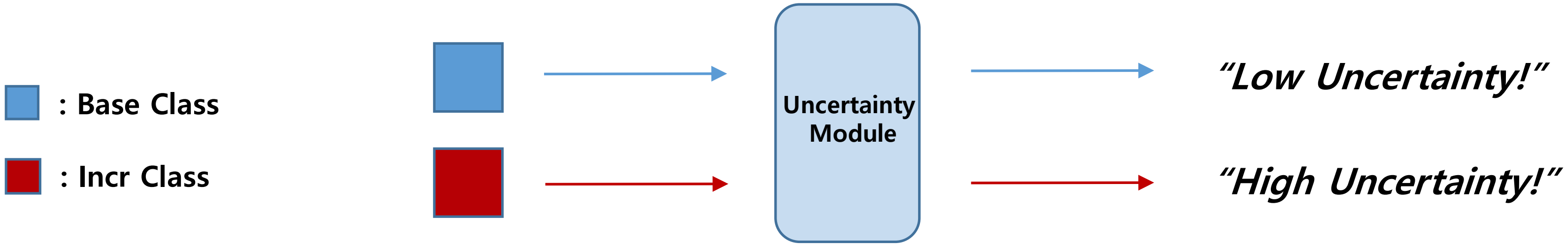


Figure 5. Diagram of uncertainty module

1.2 Motivation: Need for Uncertainty Quantification

- Prediction Calibration using Uncertainty Score
  - ***Adjust model predictions by incorporating uncertainty estimates*** for base and incremental class groups
  - Assign higher predictive weight to incremental classes in proportion to their uncertainty scores

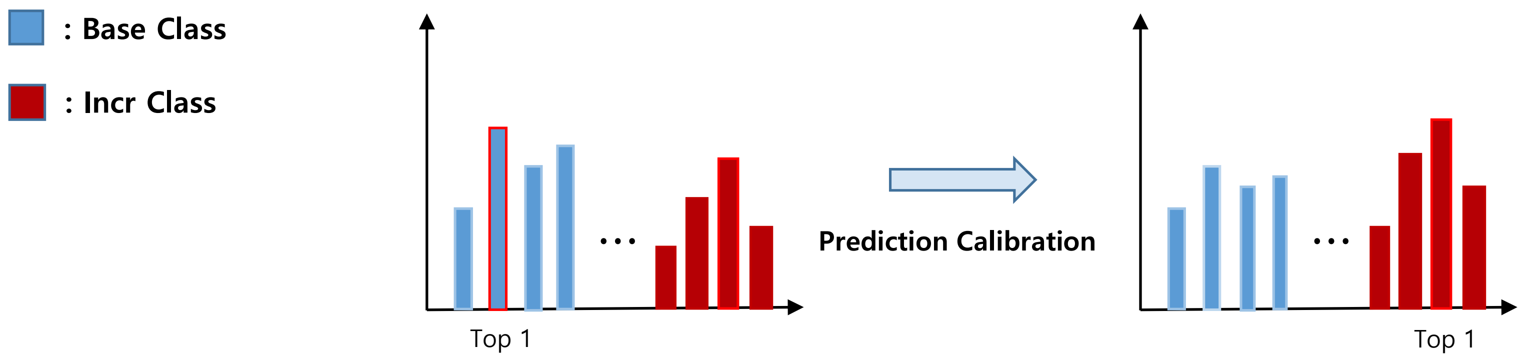


Figure 6. Diagram of prediction calibration

Related Works

2.1 Neural Collapse-Inspired Few-Shot Class Incremental Learning

- Neural collapse phenomenon
  - In a well-trained classification model, the last-layer features of each class collapse ***into their class mean***
  - For a K-class classification, these vectors form a K-dimensional ***simplex equiangular tight frame*** $(\boldsymbol{ETF})^{\dagger}$
  - The ETF is a geometric structure that ***maximizes the pair-wise angles*** between all vectors [5]

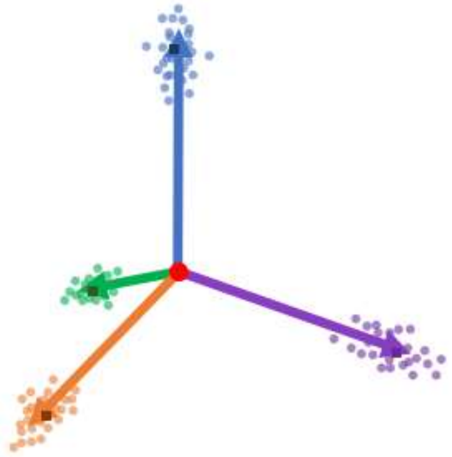


Figure 7. Visualization of a neural collapse phenomenon (figure from [6]).

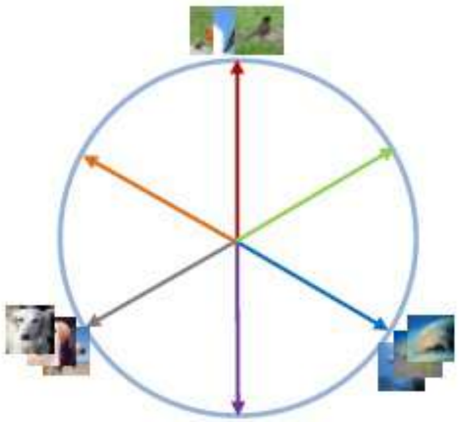


Figure 8. Visualization of a neural collapse in the hyperplane (figure from [4]).

† See Appendix A for detailed explanation.

2.1 Neural Collapse-Inspired Few-Shot Class Incremental Learning

- Neural collapse-Inspired FSCIL (NC-FSCIL)
  - Yang et al. (2023) [4] pre-assigns an ETF classifier and aligns the features to this optimal structure
  - Apply ***Dot-Regression(DR) Loss***† for alignment with ground-truth ETF vector

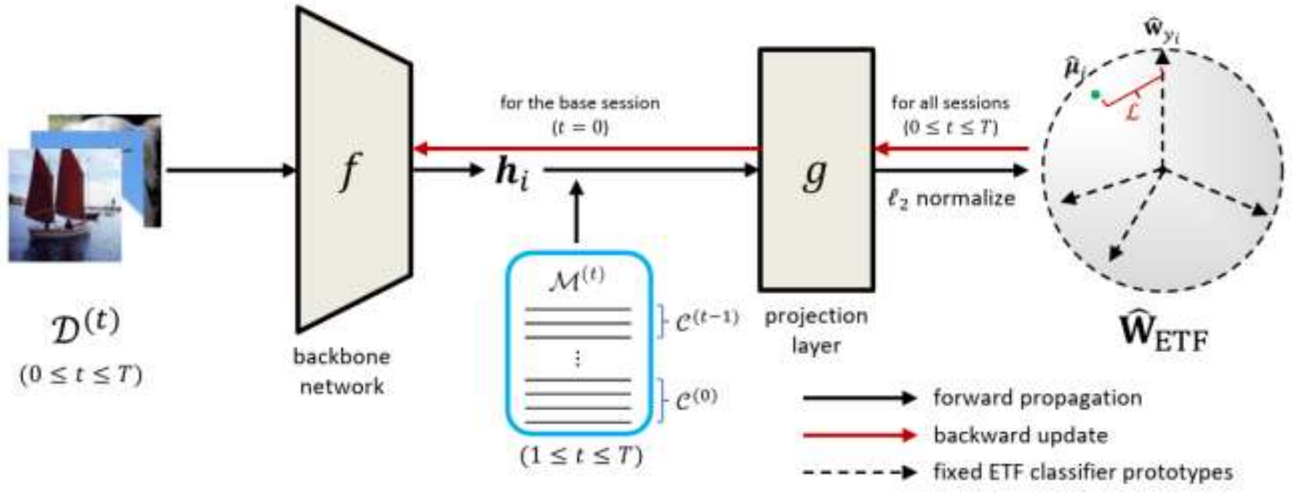


Figure 9. Neural Collapse-Inspired FSCIL Framework (figure from [4]).

† See Appendix B for detailed explanation.

2.2 Objective

- Neural Collapse-Driven Uncertainty Estimation
  - Based on the observation at *Figure 4*, we hypothesize the asymmetrical misclassification stems from ***the difference in alignment of base and incremental prototypes.***
  - ETF vectors are pre-defined vectors for all classes
    - We can mathematically quantify ***the alignment of a given feature with ETF vectors***
    - Base classes strongly aligns with ***base ETF vectors***
    - Incremental classes are influenced by ***both base, incremental ETF vectors***

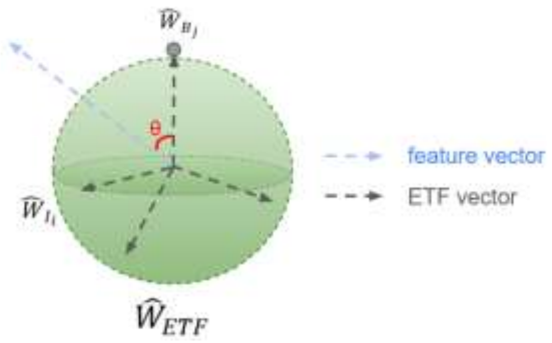


Figure 10. Diagram of Uncertainty Computation in the ETF Feature Space

Related Works

2.2 Objective

I. Propose a ***novel uncertainty estimation strategy tailored for FSCIL***, leveraging the ETF geometry to quantify how feature representations align with base and incremental class prototypes
   - Introduce an uncertainty estimation module on top of the NC-FSCIL [4] baseline
   - Allows the model to output both class predictions and uncertainty scores for each input sample

II. Design a ***post-hoc calibration method*** to improve classification performance
   - Leverage class-wise uncertainty distributions to re-weights predictions
   - Experiments demonstrate our method shows better performance across all incremental sessions

"We aim to fulfill these objectives through uncertainty estimation and calibration, thereby mitigating the asymmetric misclassification and prevalent bias towards base classes in FSCIL."

Methods

3.1 Overview

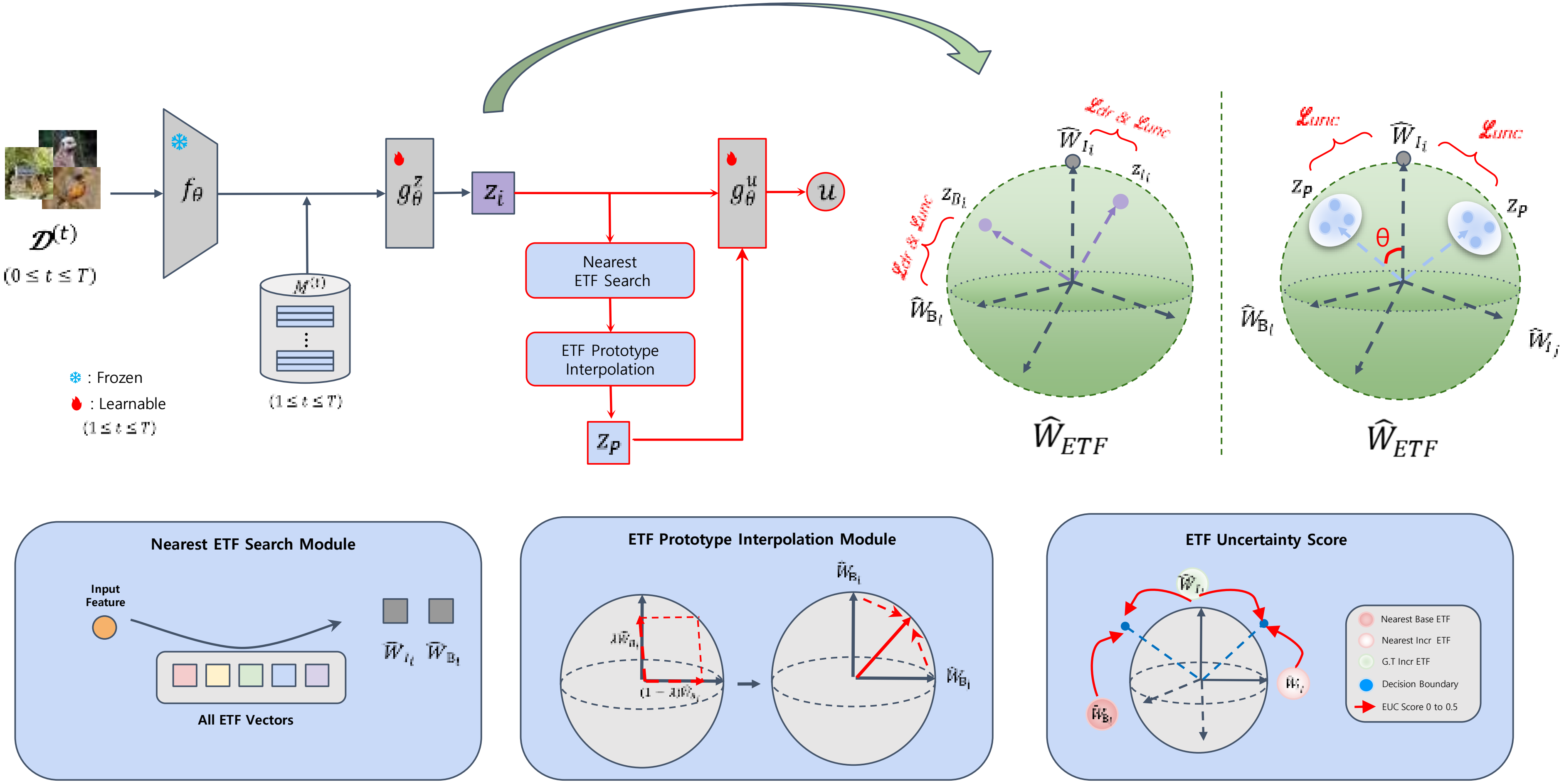


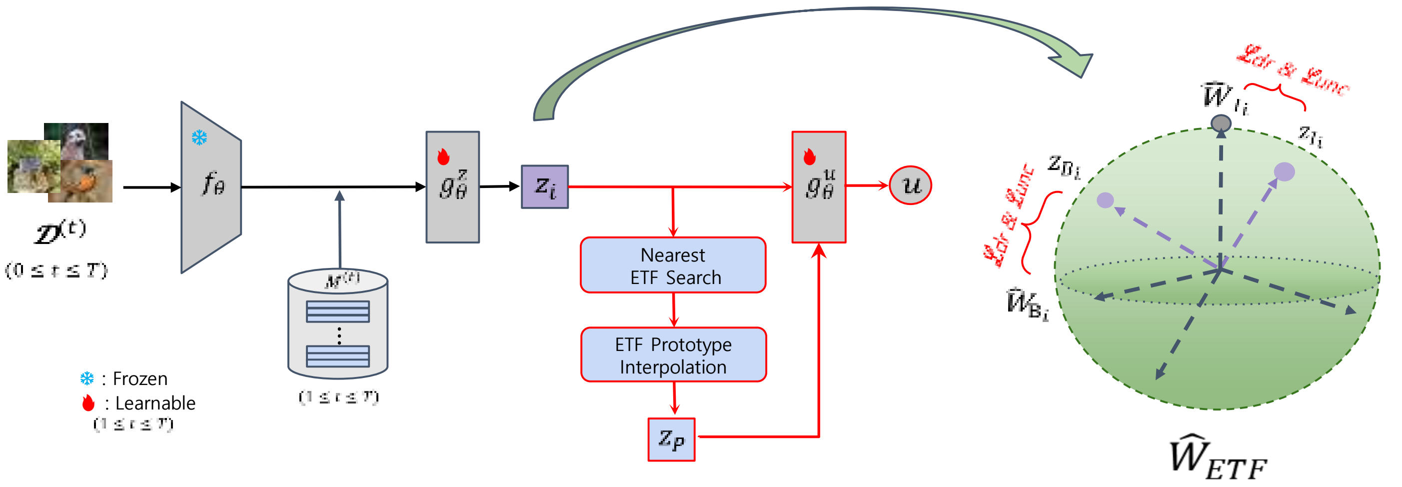
Figure 11. Overall framework of the proposed method. Red line indicate our proposed module integrated on top of the baseline.

Methods

3.2 Neural Collapse-Driven Uncertainty Estimation

- Uncertainty Estimation Module
  - Add additional projection head g^u for uncertainty score $u \in [0,1]$
  - The representation z_i is passed into g^u, which predicts the score u_i
  - The EUC Loss ($\mathcal{L}_{\mathrm{unc}}$) enforces the model to predict uncertainty score

$$\mathcal{L}_{total} = \mathcal{L}_{DR} + \lambda \cdot \mathcal{L}_{unc} \qquad \mathcal{L}_{unc} = MSE(g^u(\hat{\mu}_i), EUC\text{-}Score(\hat{\mu}_i))$$

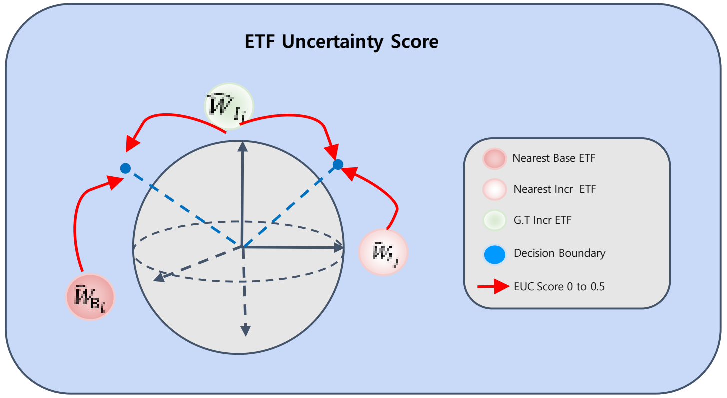


Methods

3.2 Neural Collapse-Driven Uncertainty Estimation

- ETF Uncertainty Score (EUC Score)
  - Supervision to the uncertainty projection module g^u
  - It is designed to ***quantify the alignment of both base and incremental ETF vectors***

$$EUC - Score = \frac{1 + [maxcos(z, W_{base}) - maxcos(z, W_{incr})]}{2}$$

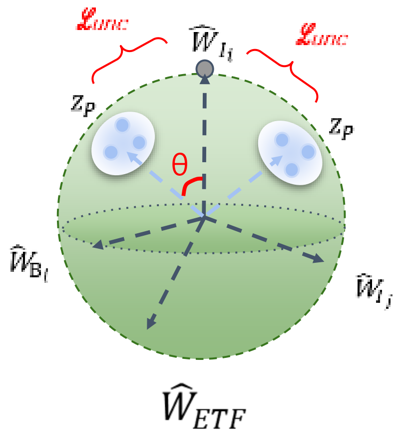


  - Lower score indicates strong alignment with the <u>*incremental ETF prototypes*</u>
  - Higher score indicates strong alignment with the <u>*base ETF prototypes*</u>
  - Middle score indicates representations <u>*near the decision boundary*</u>
  - For base class predictions, we interpret the score as <u>*1− score to ensure consistency with the uncertainty interpretation.*</u>

Methods

3.2 Neural Collapse-Driven Uncertainty Estimation

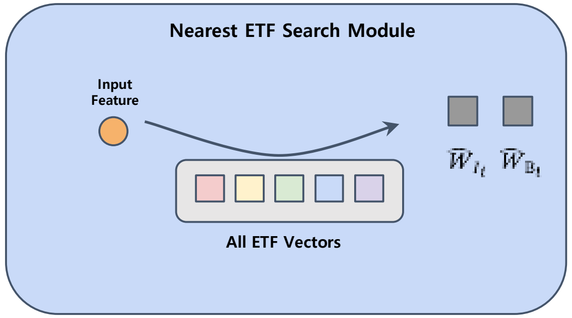
- Creating pseudo samples via Prototype Interpolation
  - Due to the strong alignment induced by $\mathcal{L}_{DR}$, all features are prone to tightly clustered near their respective ETF prototypes
  - This results in a behavior in g^u to ***collapse to predicting only zero or one uncertainty for all inputs***
  - We use two modules to create ***pseudo samples*** to provide more discriminative uncertainty estimates
    - Nearest ETF Search Module
    - ETF Prototype Interpolation Module



Methods

3.2 Neural Collapse-Driven Uncertainty Estimation

- Creating pseudo samples via Prototype Interpolation
  - Nearest ETF Search Module
    - Given an input feature, we retrieve the ETF vector with the high cosine similarity
    - For base session (t = 0), retrieve the ***most similar incremental ETF***
    - For incremental session (t > 0), retrieve the ***most similar base & incremental ETF***

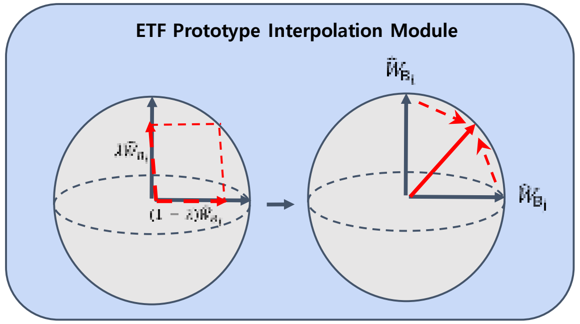


  - ETF Prototype Interpolation Module
    - Generate pseudo samples via interpolation

$$z_{pseudo} = \alpha \cdot w_{GT} + (1 - \alpha) \cdot w_{near}, \ \alpha \in (0, 0.5)$$

    - ***Inject Gaussian noise*** to promote diversity and generalization

$$z_{pseudo} = z_{pseudo} + \delta, \ \delta \sim N(0, \sigma^2 I)$$



Methods

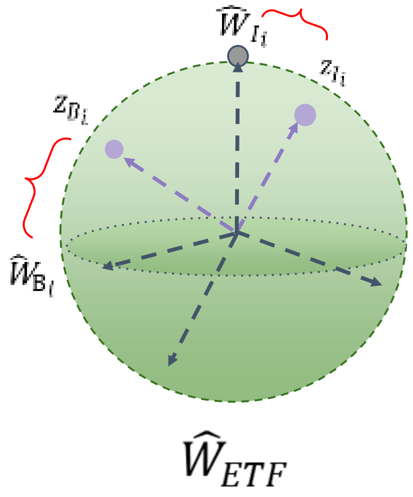
3.3 Uncertainty Score based Calibration

- Cosine Similarity-Based Prediction
  - Prediction is performed by computing following equation :

$$\hat{y} = \arg\max_{j} \cos(f, W_j)$$

- Likelihood-Based Scaling
  - During each incremental session (t > 0), ***estimate the Gaussian distribution of uncertainty scores*** for both base and incremental groups
  - We use ***prototype memory*** for base class, ***few-shot samples*** for incremental class

$$\lambda_{base} = N(1 - u(x); \mu_b, \sigma_b^2),\ \lambda_{incr} = N(u(x); \mu_i, \sigma_i^2),$$



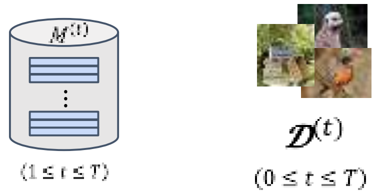


Figure 12. Prototype memory for base class, Few-shot sample for incremental class.

Methods

3.3 Uncertainty Score based Calibration

- Likelihood-Based Scaling
  - During each incremental session ($t > 0$), ***estimate the Gaussian distribution of uncertainty scores*** for both base and incremental groups
  - We use ***prototype memory*** for base class, ***few-shot samples*** for incremental class

$$\lambda_{base} = N(1 - u(x); \mu_b, \sigma_b^2),\ \lambda_{incr} = N(u(x); \mu_i, \sigma_i^2),$$

  - $\tilde{s}_j = cos\left(f, W_j\right) + \alpha \cdot w(x)$, ***where w(x) is normalized confidence weight*** $w(x) = \frac{\lambda_{base}}{\lambda_{base} + \lambda_{incr}}$
  - *Finally,* ***predict class label*** $\hat{\mathbf{y}}$, $where\ \hat{y} = \arg\max_j \tilde{s}_j$

Experiments

4.1 Experimental Settings

- Datasets and data splits
  - All experiments follow the data splits from [4] in a 5-shot setting (5 samples per incremental class)

Dataset	Number of class per session		Number of incremental sessions
	Base session	Incremental sessions	
CIFAR-100 [7]	60	5	8
*mini*ImageNet [8]	60	5	8
CUB-200 [9]	100	10	10

Table 1. Data splits in the FSCIL scenario for CIFAR-100 [7], *mini*ImageNet [8], and CUB-200 [9].

  - Evaluation metrics
    - Average accuracy ($aAcc$), base accuracy ($aAcc_b$), incremental accuracy ($aAcc_i$)
    - Generalized average accuracy ($gAcc$) [10] †

† See Appendix C for detailed explanation.

Experiments

4.1 Experimental Settings

- Implementation details
  - The backbone network and initial hyper-parameter settings are adopted from NC-FSCIL [4]
  - We adopt ***batch size of 32*** at incremental session, ***0.1 learning rate*** for uncertainty loss
  - We use ***fine-tune learning rate of 0.02*** at incremental session of CIFAR-100
  - *We re-implement the original baseline at our own reproduction* †

Dataset	Backbone	Batch size		Finetuning LR
		Base session	Incremental sessions	
CIFAR-100 [7]	ResNet-12	256	***8 -> 32***	***0.3 -> 0.02***
*mini*ImageNet [8]	ResNet-12	256	***8 -> 32***	0.05
CUB-200 [9]	ResNet-18	256	***8 -> 32***	0.01

Table 2. Implementation details and commonly used hyperparameters in the FSCIL research.

† See Appendix D for detailed explanation.

Experiments

4.2.1 CIFAR-100 Results

Methods	Accuracy in each session (%) ↑										
	0	1	2	3	4	5	6	7	8	Aacc	Gacc
NC-FSCIL (Baseline)	82.33	77.09	72.86	68.59	64.94	61.15	59.38	56.58	54.86	66.42	56.99
Base Class Acc	82.33	79.72	78.70	77.28	76.55	74.08	73.20	72.13	71.85	-	-
Incr Class Acc	-	45.60	37.80	33.80	30.10	30.12	31.73	29.91	29.37		
Base Class Acc (same hyperparameter)	82.33	81.88	80.60	78.93	78.08	76.35	77.03	77.23	76.03	-	-
Incr Class Acc (same hyperparameter)	-	7.00	17.50	21.33	23.15	22.56	19.07	17.98	20.13	-	-
Ours											
+ UC Module†	82.88	77.62	73.27	67.32	64.72	62.11	59.31	57.21	54.71	<u>66.57</u>	57.69
+ UC Calibration	82.88	**77.25**	**72.90**	66.41	64.74	**61.91**	59.21	**57.02**	54.47	66.24	**58.22**
Base Class Acc	82.88	81.02	79.37	74.77	78.53	74.72	74.42	74.92	72.73	-	-
Incr Class Acc	-	34.20	35.60	33.00	23.35	31.16	28.80	26.34	27.07	-	-

Table 3. Comparison with the baseline on CIFAR-100. We report base/incr/average acc(Aacc)/generalized acc(Gacc).

† See Appendix E for detailed explanation.

Experiments

4.2.2 miniImageNet Results

Methods	Accuracy in each session (%) ↑										
	0	1	2	3	4	5	6	7	8	Aacc	Gacc
NC-FSCIL (Baseline)	84.02	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	67.81	60.93
Base Class Acc	84.02	78.68	75.98	74.10	74.95	76.18	76.18	75.55	76.30	-	-
Incr Class Acc	-	54.20	48.10	42.73	40.55	34.88	32.00	32.09	31.33		
Ours											
+ UC Module†	84.33	76.35	72.71	68.23	67.11	64.41	62.67	60.39	59.01	<u>68.35</u>	61.77
+ UC Calibration	84.33	75.28	**72.23**	67.76	**66.47**	63.67	**62.54**	**60.45**	**59.18**	**67.99**	**61.96**
Base Class Acc	84.33	77.07	76.03	73.67	74.47	74.22	75.10	75.15	75.47	-	-
Incr Class Acc	-	53.80	49.40	44.13	42.50	38.36	37.43	35.26	34.75	-	-

Table 4. Comparison with the baseline on miniImageNet. We report base/incr/average acc(Aacc)/generalized acc(Gacc).

† See Appendix E for detailed explanation.

Experiments

4.2.3 CUB200 Results

Methods	Accuracy in each session (%) ↑												
	0	1	2	3	4	5	6	7	8	9	10	Aacc	Gacc
NC-FSCIL (Baseline)	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	61.56
Base Class Acc	80.45	76.89	77.62	78.63	77.23	77.13	76.85	76.29	76.61	75.94	76.19		
Incr Class Acc	-	66.67	45.41	42.59	45.88	41.72	44.11	44.99	41.16	42.66	43.07		
Ours													
+ UC Module †	80.87	76.01	72.97	69.93	67.83	65.44	64.87	63.71	61.47	60.36	59.67	<u>67.55</u>	<u>61.86</u>
+ UC Calibration	80.87	75.63	**72.97**	70.14	67.95	**65.44**	**65.04**	**63.73**	**61.22**	**60.36**	**60.06**	**67.30**	**62.55**
Base Class Acc	80.87	75.91	77.44	77.86	76.19	76.50	75.84	75.56	75.28	74.55	74.97		
Incr Class Acc	-	72.76	50.35	44.56	47.68	43.78	47.37	47.19	44.02	44.90	45.49		

Table 5. Comparison with the baseline on CUB200. We report base/incr/average acc(Aacc)/generalized acc(Gacc).

† See Appendix E for detailed explanation.

Experiments

4.2.4 Overall Results

- Analysis of Results
  - ***Consistent improvements in the generalized average accuracy (gAcc)*** across all benchmark datasets
  - Our method slightly ***sacrifices base class accuracy in favor of enhancing incremental class accuracy***
  - For CIFAR-100, difference in experimental settings such as number of GPU, resulting unfair comparison
    - *Under same experimental environment, our method outperforms the baseline*

Dataset	Final session	Average Base Class	Average Incremental Class		
				Aacc	Gacc
CIFAR-100 [7]	0.39% drop	1.21% drop	***11.35% gain***	0.18% drop	***1.23% gain***
*mini*ImageNet [8]	***0.87% gain***	0.71% drop	***2.46% gain***	***0.17% gain***	***1.03% gain***
CUB-200 [9]	***0.62% gain***	0.80% drop	***2.98% gain***	***0.02% gain***	***0.99% gain***

Table 6. Overall results of performance difference of three benchmarks.

Experiments

4.3 Discussion

- Comparison between Predicted and Post-hoc EUC Scores
  - Although the predicted uncertainty <u>does not perfectly match the post-hoc uncertainty score</u>, for ***CIFAR100*** and ***miniImageNet***, the model successfully captures the ***tendency of base class exhibiting lower uncertainty than incremental samples***
  - We conjecture ***CUB200*** mismatch is due to significantly smaller training size (1/6 size)
  - Possibility of over/under fitting -> <u>need extensive experiments</u>

	Model	**G.T**
Base	$N(0.286, 0.084)$	$N(0.23, 0.053)$
Incremental	$N(0.347, 0.083)$	$N(0.44, 0.05)$

(a) CIFAR100

	Model	**G.T**
Base	$N(0.227, 0.04)$	$N(0.306, 0.022)$
Incremental	$N(0.403, 0.04)$	$N(0.449, 0.019)$

(b) miniImageNet

	Model	**G.T**
Base	$N(0.33, 0.03)$	$N(0.312, 0.013)$
Incremental	$N(0.269, 0.026)$	$N(0.426, 0.008)$

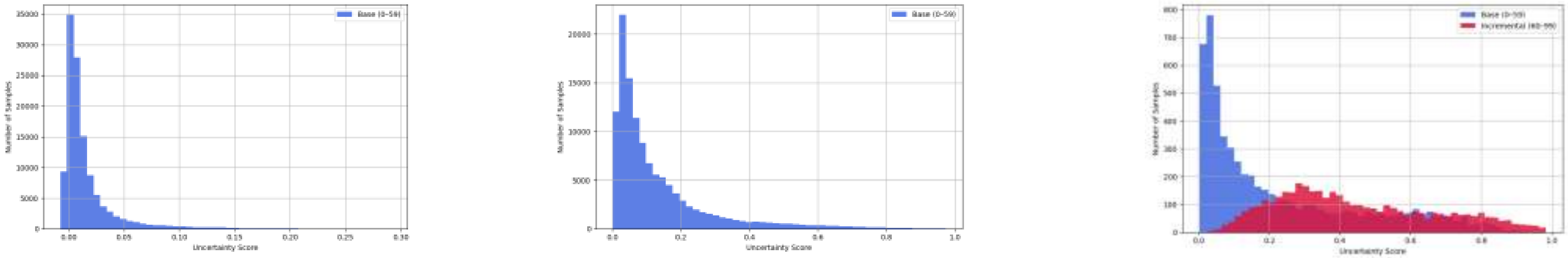
(c) CUB200

Table 7. Comparison of prediction UC score and Post-hoc UC score.

Experiments

4.3 Discussion

- Uncertainty Score Distribution of Base/Incremental Class
  - After the base session, uncertainty scores of base samples are highly concentrated near zero
  - As session progress, ***the distribution gradually shifts due to the influence of incremental classes (i.e., catastrophic forgetting)***
  - Incremental samples exhibit ***consistently higher uncertainty scores*** compared to base samples
  - This supports our hypothesis that incremental samples are more influenced by both base and incremental prototypes



(a) Session 0 **(b) Session 9** **(c) Session 9**

Figure 13. Comparison of base and incremental uncertainty distributions on CIFAR100. **(a)** and **(b)** shows train set distribution, while **(c)** shows the test set distribution.

4.3 Discussion

- Prediction Calibration for Base and Incremental samples
  - The mean uncertainty of incremental class samples remains consistently ***higher than that of base class samples*** across all sessions
  - This enables our calibration strategy to ***assign larger correction weights to incremental samples***, thereby refining the prediction

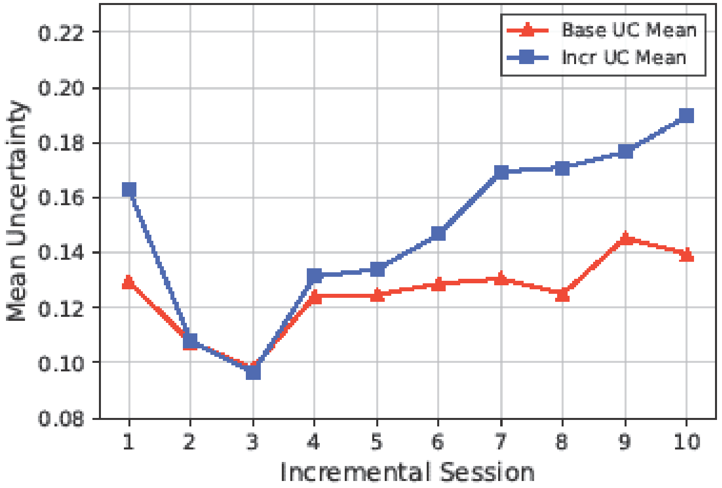


Figure 14. Mean uncertainty of the training set for each session on the CUB-200 dataset.

Conclusion

5.1 Limitation and Further Works

- Discrepancy between Train and Test Uncertainty EUC(ETF-Uncertainty) score Distribution
  - It is evident that the true distribution of uncertainty scores differs from that of the training data used for calibration
  - From *Figure 12*, we can only estimate the true uncertainty ***as a practical approximation under FSCIL constraints***
  - We can consider the actual test uncertainty distribution as an ***upper bound***
    - *Further extensive work to match the upper bound is required.*

- Unstable Uncertainty Module
  - More experiments with CIFAR100 required (fair comparison with baseline)
  - Additional tuning is required to better align with the post-hoc EUC scores

Conclusion

5.2 Conclusion

- Motivation
  - We aim to address the challenge of ***asymmetric misclassification*** in FSCIL, where incremental samples are often misclassified into base classes
- Method Summary
  - We propose an ***uncertainty-aware framework*** for Few-Shot Class-Incremental Learning (FSCIL)
  - A novel ***uncertainty score*** is introduced, base on alignment with class-specific ETF prototypes
  - We design a ***post-hoc calibrate scheme*** to adjust prediction confidence based on uncertainty score
- Experimental Results
  - We reduce the prevalent bias toward base classes, ***leading to more balanced predictions***
  - We enhance ***incremental class performance*** across all datasets, achieve notable improvements in ***generalized average accuracy*** ($gAcc$)

References

[1] Zhou, Da-Wei, et al. "Class-incremental learning: A survey." IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[2] Xiaoyu Tao, et al. “Few-shot class-incremental learning.” In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12183–12192, 2020.

[3] Zhou, Da-Wei, et al. "Forward compatible few-shot class-incremental learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[4] Yang, Yibo, et al. "Neural collapse inspired feature-classifier alignment for few-shot class incremental learning." arXiv preprint arXiv:2302.03004 (2023).

[5] Papyan, et al. "Prevalence of neural collapse during the terminal phase of deep learning training." Proceedings of the National Academy of Sciences 117.40 (2020): 24652-24663.

[6] Xie, Liang, et al. "Neural collapse inspired attraction–repulsion-balanced loss for imbalanced learning." Neurocomputing 527 (2023): 60-70.

[7] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

[8] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115 (2015): 211-252.

[9] Wah, Catherine, et al. "The caltech-ucsd birds-200-2011 dataset." (2011).

[10] Tang, Yu-Ming, et al. "Rethinking few-shot class-incremental learning: Learning from yourself." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024.

감사합니다.

Sungwon Woo,
Multimedia System Laboratory
Department of Artificial Intelligence

SOGANG UNIVERSITY

Appendix

A. Simplex Equiangular Tight Frame (ETF)

- A geometric structure formed at the terminal phase of training
  - A simplex ETF refers to a matrix that is composed of K vectors $E = [e_1, \dots, e_k] \in \mathbb{R}^{d \times K}$ that satisfies:

$$E = \sqrt{\frac{K}{K-1}} U \left(I_K - \frac{1}{K} 1_K 1_K^{\mathrm{T}} \right)$$

  - $U \in \mathbb{R}^{d \times K}$ is an orthogonal matrix that satisfies $U^T U = I_K$, and 1_K is an all-ones vector
  - All column vectors in E have the same ℓ_2 norm and any pair satisfies:

$$e_i^T e_j = \begin{cases} 1 & if\ i = j \\ -\dfrac{1}{K-1} & if\ i \neq j \end{cases}$$

Appendix

B. Dot-Regression (DR) Loss

- Elimination of the *push* term
  - The gradient of CE loss is composed of a *pull* term and a *push* term
    - *pull* term drives the feature into its classifier prototype of the same class
    - *push* term pushes the feature away from the prototypes of different classes
  - DR loss includes **only the *pull* gradients** assuming the optimal prototypes are given by the ETF:

$$\mathcal{L}(\hat{\mu}_i, \widehat{W}_{ETF}) = \frac{1}{2}\left(\widehat{w}_{y_i}^T \hat{\mu}_i - 1\right)^2$$

  - $\hat{\mu}_i$ is the normalized feature, $\widehat{W}_{ETF}$ is the ETF matrix, and $\widehat{w}_{y_i}^T$ is the prototype in E for class y_i

Appendix

C. Generalized Average Accuracy

- Complement metric for novel-class performance [10]
  - Existing $aAcc$, $lAcc$ is ***dominated*** by the base-class performance, having bias towards base-class regardless of the ***inferior performance on the novel class***
  - Existing $aAcc$ is written by:

$$aAcc_i = \frac{\frac{|y_i|}{|y_{novel}|} A_i^1 + \sum_{j=2}^{i} A_i^j}{\frac{|y_i|}{|y_{novel}|} + (i - 1)}$$

  - $|y_i|$ is the size of the labeling space of task T_i, A_i^j denotes the accuracy on the class set of task T_j, when the model has been trained up to the task T_i
  - $\frac{|y_i|}{|y_{novel}|}$ is too large in the FSCIL settings ($e.g.$ $\frac{|y_i|}{|y_{novel}|} = 12$ for CIFAR-100 and miniImageNet, 10 for CUB-200)

Appendix

C. Generalized Average Accuracy

- Complement metric for novel-class performance [10]
  - To overcome the limitations of existing $aAcc$, $lAcc$, we define the generalized accuracy ($gAcc$) as:

$$gAcc_i(\alpha) = \frac{\alpha \frac{|y_i|}{|y_{novel}|} A_i^1 + \sum_{j=2}^{i} A_i^j}{\alpha \frac{|y_i|}{|y_{novel}|} + (i - 1)}$$

  - $gAcc$ generalizes the α to any rational numbers in [0,1]
  - The area under the curve (AUC) of $gAcc_i(\alpha)$ is defined as:

$$gAcc_i = \int_0^1 gAcc_i(\alpha)\, d\alpha$$

  - For multiple tasks, we average the $gAcc_i$ for each task and get an overall metric $gAcc$:

$$gAcc = \frac{1}{n_t} \sum_{i}^{n_t} gAcc_i$$

Appendix

D. Training Implementation

- NC-FSCIL [4] Baseline
  - We follow the same hyperparameter settings except:
    - Fine-tuning learning rate (0.3 → 0.02) at CIFAR-100
    - Samples per GPU (8 → 32)
- Uncertainty Module
  - We adopt two-layer MLP with a hidden dimension of 256.
  - A ReLU activation function is used and a sigmoid activation at the final layer
  - We use Gaussian noise with a standard deviation of 0.05 to the interpolated features
  - We use uncertainty loss of 0.1 throughout all training stages.
- Prediction Calibration
  - We use value of 0.05 for α in the re-weighting process

Appendix

E. Effectiveness of Uncertainty Module

- Slight improvement in overall accuracy when UC module applied alone
  - We conjecture the ***additional UC loss applied*** in addition to the DR loss
  - DR loss primarily focuses on aligning with G.T ETF vectors
  - UC loss aims to ***model the relative alignment between Base and Incremental ETF***
  - Consider two feature vectors A,B
    - The DR loss is identical, whereas ***EUC scores differ***

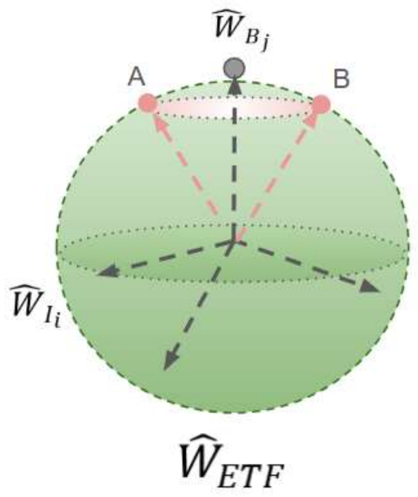


Figure 15. Diagram of difference between DR loss and EUC score in measuring alignment with ETF prototypes