NAPA-VQ: Neighborhood-Aware Prototype Augmentation with Vector Quantization for Continual Learning (Supplementary Materials)

Tamasha Malepathirana Damith Senanayake
Saman Halgamuge
Dept. of Mechanical Engineering
The University of Melbourne
{tamasha.malepathirana, damith.senanayake, saman.halgamuge}@unimelb.edu.au

1. Additional explanations

In this section, we provide additional details related to evaluation metrics and our implementation.

1.1. Evaluation metrics

- **Average Accuracy**: Similar to [10, 9], we calculate the average accuracy at the end of the final task T (A_T) as the mean of the accuracies across all the incremental tasks, including the initial task. The accuracy at task t (a_t) is the fraction of correctly classified samples from classes learned up to and during task t. A_T allows a fair comparison by summarizing a method's performance across all the incremental stages.

$$A_T = \frac{1}{T+1}\sum_{t=0}^{T} a_t \tag{1}$$

- **Average Forgetting**: In line with [10, 9], the forgetting of the classes learned in task j after the model has been trained on task t (f_j^t), is measured by the difference between the maximum accuracy for task j during the learning process and the accuracy for the same task at the end of training with task t.

$$f_j^t = (\max_{i\in\{j,..,t-1\}} a_{i,j}) - a_{t,j} \tag{2}$$

where $a_{i,j}$ is the accuracy of classes first learned in task j after the model has been trained on task i.

Consequently, the average forgetting at the end of the task t ($t > 0$) is defined as Eq. 3 below. We report the average forgetting at the end of the final task T (F_T).

$$F_t = \frac{1}{t}\sum_{j=0}^{t-1} f_j^t \tag{3}$$

1.2. Implementation details

For a fair comparison, we adapted the same backbone architecture, ResNet-18 [1] as [5, 9, 8, 10]. We use two optimizers, $Optim_\theta$ and $Optim_\phi$ to jointly train the feature extractor (F_θ) and the set of CVs (M_ϕ). During the training of the base task, we use the same optimizers and learning rate scheduling as [9] to optimize $Optim_\theta$ in CIFAR-100, TinyImageNet, ImageNet-Subset datasets. The detailed parameters used to train $Optim_\theta$ and $Optim_\phi$ are listed in Table 1. The concrete implementation can be found in our code attached to the supplementary materials.

Table 1. Detailed parameter values used to train $Optim_\theta$ and $Optim_\phi$

	CIFAR-100		TinyImageNet		ImageNet-Subset		ImageNet-1K	
# of epochs	100		50		100		50	
	$Optim_\theta$	$Optim_\phi$	$Optim_\theta$	$Optim_\phi$	$Optim_\theta$	$Optim_\phi$	$Optim_\theta$	$Optim_\phi$
Optimizer	Adam	SGD	Adam	SGD	SGD	SGD	SGD	SGD
Base task	initial learning rate of 0.001 decayed by 0.1 every 45 epochs	initial learning rate of 5 decayed by 0.1 every 20 epochs	initial learning rate of 0.001 decayed by 0.1 every 45 epochs	initial learning rate of 5 decayed by 0.1 every 20 epochs	initial learning rate of 0.1 decayed by 0.1 at steps 80, 120, and 150 (The base task is trained for 160 epochs similar to [9])	initial learning rate of 5 decayed by 0.1 every 20 epochs	initial learning rate of 0.01 decayed by 0.1 at steps 20, 40, and 60 (The base task is trained for 80 epochs)	initial learning rate of 5 decayed by 0.1 every 20 epochs
Incremental tasks	initial learning rate of 0.0001 decayed by 0.1 every 45 epochs	initial learning rate of 0.5 decayed by 0.1 every 20 epochs	initial learning rate of 0.0001 decayed by 0.1 every 45 epochs	initial learning rate of 0.5 decayed by 0.1 every 20 epochs	initial learning rate of 0.0001 decayed by 0.1 every 45 epochs	initial learning rate of 0.5 decayed by 0.1 every 20 epochs	initial learning rate of 0.001 decayed by 0.1 at steps 20, and 40	initial learning rate of 0.5 decayed by 0.1 every 20 epochs

Parameter values of $\tau = 10, K = 15, \epsilon = 0.9, e_{min} = 0.9^{10}, \alpha = 0.001$ were used for all four datasets.

1.3. Experiments on loss weights

The average accuracy and average forgetting values obtained for the ablation study conducted when introducing loss coefficients and their combinations are shown in Table 2. Due to the imbalance between the number of prototypes representing the old classes and the number of samples representing the new classes, we empirically determined that larger coefficients ($\lambda_1 = 10$ and $\lambda_2 = 10$) are crucial for L^t_{KD} (knowledge distillation) and $\hat{L}^t_{DCE}$ (prototype-based distance cross-entropy) to effectively mitigate the feature drift and the forgetting of old classes. This aligns with previous works (PASS [9], IL2A [8]) which also introduce larger coefficients for knowledge distillation and prototype-based cross-entropy loss.

Following the same reasoning, we introduced a larger coefficient ($\lambda_3 = 10$) for $\hat{L}^t_{NA}$ (Neighborhood-Adaptation loss for old class prototypes) and no scaling factor for L^t_{NA} (Neighborhood-Adaptation loss for new class samples). However, this setting (the last row in Table 2) only resulted in the stability either remaining the same or improving marginally, while the plasticity decreased. We recognize the importance of adequately learning new classes and, therefore, chose not to include a larger scaling factor for $\hat{L}^t_{NA}$. However, we agree that further investigations are needed to determine the optimal scaling values and consider this a subject for future explorations.

Table 2. Average Accuracy (%) and Average Forgetting (%) obtained for the ablation study conducted using CIFAR-100 to determine coefficients values used for $\hat{L}^t_{DCE}$ (λ_1), L^t_{KD} (λ_2) and $\hat{L}^t_{NA}$ (λ_3). The proposed setting is bolded.

λ_1	λ_2	λ_3	Average Accuracy ↑			Average Forgetting ↓		
			T=5	T=10	T=20	T=5	T=10	T=20
1	1	1	69.74	66.81	62.05	14.44	25.83	34.4
10	1	1	70.14	69.27	67.21	7.21	11.3	13.64
1	10	1	69.55	67.5	62.42	13.73	19.39	29.371
1	1	10	69.37	67.15	62.92	15.14	23.27	32.61
10	**10**	**1**	**70.44**	**69.04**	**67.42**	**6.9**	**9.65**	**9.08**
1	10	10	69.36	67.67	63.46	14.7	19.57	26.23
10	1	10	70.56	68.66	67.02	7.21	12.45	14.22
10	10	10	70.17	68.92	67.67	6.2	9.28	8.17

2. Additional analysis

2.1. Detailed values at the incremental tasks

We report the detailed accuracy values obtained by NAPA-VQ at each incremental stage under the three incremental scenarios ($T = 5$, $T = 10$, and $T = 20$) in Tables 3, 4, and 5.

Table 3. The accuracy at each incremental task under the setting of $T = 5$ scenario.

Dataset	Task						Average
	0	1	2	3	4	5	
CIFAR-100	81.15	75.28	71.47	67.70	64.75	62.30	**70.44**
TinyImageNet	61.2	56.84	54.38	51.43	48.42	44.38	**52.77**
ImageNet-Subset	80.57	75.34	70.46	66.62	62.70	59.19	**69.15**
ImageNet-1K	68.83	61.36	56.25	51.37	48.12	44.71	**55.11**

Table 4. The accuracy at each incremental task under the setting of $T = 10$ scenario.

Dataset	Task											Average
	0	1	2	3	4	5	6	7	8	9	10	
CIFAR-100	80.8	78.25	74.82	72.16	70.56	68.40	66.11	64.65	62.60	61.19	59.92	**69.04**
TinyImageNet	61.31	59.00	56.88	54.46	53.34	51.71	50.31	48.77	46.89	44.54	42.44	**51.78**
ImageNet-Subset	80.59	78.97	75.77	72.64	70.32	68.16	66.20	64.55	61.54	59.95	57.93	**68.83**
ImageNet-1K	68.89	63.53	60.24	57.2	54.72	51.97	49.47	47.1	45.30	43.19	41.86	**53.04**

2.2. New class accuracy values at the incremental tasks

We report the new class accuracy values obtained by NAPA-VQ at each incremental stage under the three incremental scenarios ($T = 5$, $T = 10$, and $T = 20$) in Tables 6, 7, and 8.

2.3. Comparison with SOTA on ImageNet-Subset

We plot the detailed accuracy curves obtained using NAPA-VQ and other compared methods for ImageNet-Subset in Fig. 1 and show that NAPA-VQ maintains higher accuracies over incremental tasks.

Table 5. The accuracy at each incremental task under the setting of $T = 20$ scenario.

Dataset	Task									
	0	1	2	3	4	5	6	7	8	9
CIFAR-100	82.16	79.87	78.75	77.06	75.72	73.95	72.09	70.06	68.40	67.08
TinyImageNet	60.63	58.93	58.42	56.88	55.53	54.11	52.86	52.18	51.19	50.49
ImageNet-Subset	81.48	78.93	75.68	74.26	72.40	70.88	70.03	67.09	65.51	64.11
ImageNet-1K	68.86	63.99	60.60	58.96	56.48	54.28	51.62	49.32	47.30	45.35

Dataset	Task											Average
	10	11	12	13	14	15	16	17	18	19	20	
CIFAR-100	66.35	64.79	64.06	62.86	61.80	60.87	59.95	58.59	57.78	57.14	56.53	**67.42**
TinyImageNet	49.36	48.54	47.57	46.59	45.56	44.40	43.50	42.41	41.09	40.19	39.27	**49.51**
ImageNet-Subset	62.09	60.49	59.08	57.53	56.26	55.51	53.28	51.60	50.50	50.03	48.21	**63.09**
ImageNet-1K	43.63	41.95	39.86	38.48	37	35.67	34.59	33.3	32.07	31.22	30.21	**45.46**

Table 6. The novel class accuracy at each incremental task under the setting of $T = 5$ scenario.

Dataset	Task					
	0	1	2	3	4	5
CIFAR-100	81.15	58.6	61.03	56.53	56.83	55.4
TinyImageNet	61.2	47.3	56.13	46.47	44.43	47.53
ImageNet-Subset	80.57	59.53	56	53.53	52.33	51
ImageNet-1K	68.83	53.22	51.08	47.6	49.38	44.75

Table 7. The novel class accuracy at each incremental task under the setting of $T = 10$ scenario.

Dataset	Task										
	0	1	2	3	4	5	6	7	8	9	10
CIFAR-100	80.8	65.27	51.27	58.4	67.47	60.53	57.27	65.53	54.93	61.93	60.4
TinyImageNet	61.31	55.2	48.87	48.6	61.87	50.67	48.33	50.2	42.6	52	45.4
ImageNet-Subset	80.59	75.73	55.07	54.94	62.93	61.87	57.2	58.8	59.47	67.07	47.6
ImageNet-1K	68.89	55.87	56.9	50.97	55.23	47.83	49.13	46.57	45.73	42.57	42.77

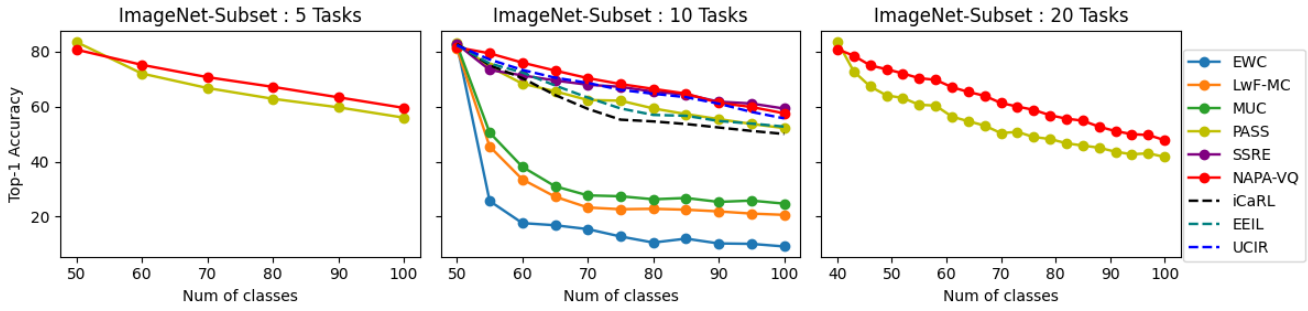

Figure 1. Detailed Accuracy curves showing the Top-1 Accuracy at each incremental step for ImageNet-Subset

Table 8. The novel class accuracy at each incremental task under the setting of $T = 20$ scenario.

Dataset	Task									
	0	1	2	3	4	5	6	7	8	9
CIFAR-100	82.16	67	77.33	68.33	68.55	61.67	56.44	45	50.44	65.44
TinyImageNet	60.63	54.67	61.2	49.47	46.27	47.33	48.27	61.33	60.8	48.93
ImageNet-Subset	81.48	62.89	37.78	66.67	60.89	75.11	68.67	38.89	66.67	65.56
ImageNet-1K	68.86	60	57.4	57.6	61.13	55.47	53.27	58.8	55.4	50.67

Dataset	Task										
	10	11	12	13	14	15	16	17	18	19	20
CIFAR-100	68.44	55.78	58.33	54	49.67	62.33	56.33	45.67	56.78	55.56	66.11
TinyImageNet	40	37.33	41.6	34.67	52.13	35.6	36.8	52.27	46.53	28.93	40.27
ImageNet-Subset	58.89	60.22	49.33	61.34	51.56	67.56	59.33	63.79	63.56	68.22	30.45
ImageNet-1K	50.33	49.47	52.93	49.07	52	47	42.8	43.4	40.6	41.8	42.07

2.4. Extended comparison of SOTA

This section includes the extended versions of Tables 1 and 2 from the main text (Tables 9 and 10) which compare the average accuracy and average forgetting values of NAPA-VQ with various other state-of-the-art (SOTA) methods including EWC [3], LwF_MC [5], MUC [4], SDC [7] and ABD [6]. Our method outperforms all the compared NECIL approaches and performs better than the compared exemplar-based approaches which store 20 exemplars per class for replay.

Table 9. Average Accuracy of NAPA-VQ compared to the existing methods on the three datasets. T represents the number of incremental tasks and E represents the number of exemplars used. Results for the methods with * were extracted from [10]. The improvement of NAPA-VQ compared to the best available SOTA is shown in red. NAPA-VQ obtains an average improvement of 5%, 2%, and 4% for CIFAR-100, TinyImageNet, and ImageNet-Subset respectively.

Methods		CIFAR-100			TinyImageNet			ImageNet-Subset		
		T=5	T=10	T=20	T=5	T=10	T=20	T=5	T=10	T=20
E=20	iCARL*	58.56	54.19	50.51	45.86	43.29	38.04	-	60.79	-
	EEIL*	60.37	56.05	52.34	47.12	45.01	40.50	-	63.34	-
	UCIR*	63.78	62.39	59.07	49.15	48.52	42.83	-	66.16	-
E=0	EWC*	24.48	21.20	15.89	18.80	15.77	12.39	-	20.40	-
	LwF_MC*	45.93	27.43	20.07	29.12	23.10	17.43	-	31.18	-
	MUC*	49.42	30.19	21.27	32.58	26.61	21.95	-	35.07	-
	SDC*	56.77	57.00	58.90	-	-	-	-	61.12	-
	PASS*	63.47	61.84	58.09	49.55	47.29	42.07	66.84	61.80	54.46
	IL2A	65.61	59.09	58.82	47.02	44.48	39.68	-	-	-
	ABD	58.38	53.49	47.73	-	-	-	-	-	-
	SSRE*	65.88	65.04	61.70	50.39	48.93	48.17	-	67.69	-
	NAPA-VQ	**70.44** **(+4.56)**	**69.04** **(+4)**	**67.42** **(+5.72)**	**52.77** **(+2.38)**	**51.78** **(+2.85)**	**49.51** **(+1.34)**	**69.15** **(+2.31)**	**68.83** **(+1.14)**	**63.09** **(+8.63)**
Average Improvement		5%			2%			4%		

Table 10. Average Forgetting of our methods compared to the other methods on the three datasets. T represents the number of incremental tasks and E represents the number of exemplars used. Results for the methods with * were reproduced in [10]. The improvement of our method compared to the best available SOTA is shown in red. NAPA-VQ exhibits a significant reduction in forgetting by an average of 10%, 3%, and 9% for CIFAR-100, TinyImageNet, and ImageNet-Subset respectively.

Methods		CIFAR-100			TinyImageNet			ImageNet-Subset		
		T=5	T=10	T=20	T=5	T=10	T=20	T=5	T=10	T=20
E=20	iCARL*	24.90	28.32	35.53	27.15	28.89	37.40	-	-	-
	EEIL*	23.36	26.65	32.40	25.56	25.91	35.04	-	-	-
	UCIR*	21.00	25.12	28.65	20.61	22.25	33.74	-	-	-
E=0	LwF_MC*	44.23	50.47	55.46	54.26	54.37	63.54	-	-	-
	MUC*	40.28	47.56	52.65	51.46	50.21	58.00	-	-	-
	PASS*	25.20	30.25	30.61	18.04	23.11	30.55	19.66	25.85	30.98
	IL2A	28.72	39.86	40.70	19.74	29.90	39.99	-	-	-
	ABD	21.80	23.92	32.76	-	-	-	-	-	-
	SSRE*	18.37	19.48	19.00	9.17	14.06	14.20	-	**8.30**	-
	NAPA-VQ	**6.90** **(-11.47)**	**9.65** **(-9.83)**	**9.08** **(-9.92)**	**9.08** **(-0.09)**	**10.81** **(-3.25)**	**9.31** **(-4.89)**	**7.17** **(-12.49)**	9.67 (+1.37)	**14.49** **(-16.49)**
Average Improvement		10%			3%			9%		

2.5. Detailed accuracy curves for the ablation study

In this section, we examine the effect of the proposed components in NA-VQ and NA-PA over the incremental steps by studying the detailed accuracy curves and forgetting curves obtained during our Ablation Study (Main text Sec 4.5). As shown in Fig. 2, the baseline model that utilizes both Categorical Cross Entropy Loss (CCE) and Knowledge Distillation Loss (KD) exhibits a rapid decline in accuracy and a rapid increase in forgetting as the incremental steps progress. While replacing CCE with Distance-based Cross Entropy Loss (DCE) initially leads to an accuracy improvement in all three incremental scenarios, a sudden drop in accuracy occurs in later increments. This drop results from the failure to retain the performance of the old classes (increased forgetting) and brings the performance close to that achieved by using CCE. This demonstrates that DCE alone is inadequate for maintaining good discrimination between classes throughout the incremental steps. The introduction of Neighborhood Adaptation Loss (NA) in addition to DCE eliminates the aforementioned sudden accuracy drop while significantly reducing forgetting due to its strong discriminatory properties.

The addition of old class representative prototypes leads to a significant improvement in performance over the incremental steps compared to previous experiments that did not use any old class representative information. The prototypes augmented with the NA-PA technique perform better than those augmented with simple Gaussian Noise in all incremental steps. This improvement can be attributed to the neighborhood awareness property in NA-PA which generates prototypes of the old classes using the feature representations of their neighboring classes, aiding to identify optimal decision boundaries and to reduce the misclassification rate. Over the incremental steps, this improvement was much more pronounced, demonstrating the effectiveness of our approach when dealing with a larger number of tasks.

2.6. Impact of the connectivity factor K

To assess the impact of the connectivity factor K on the algorithm's performance and efficiency, we experimented with different K-values ranging from 2 to 50 for the $T = 10$ incremental scenario in the CIFAR-100 dataset. The resulting average accuracy values and total training times of the models are plotted in Fig. 3. We observe that increasing the value of K leads to an increase in average incremental accuracy, which can be attributed to a wider neighborhood being considered to improve both decision boundary learning and prototype augmentation. However, beyond a certain point, the accuracy tends to fluctuate around the same level even with an increase in the K-value. This could be attributed to the inclusion of distant CVs as neighbors, who may not experience significant repulsion forces due to their distance, resulting in a similar overall performance to smaller values of K.

With respect to the efficiency of the algorithm, the running time of the model increases as the K-value is increased. This

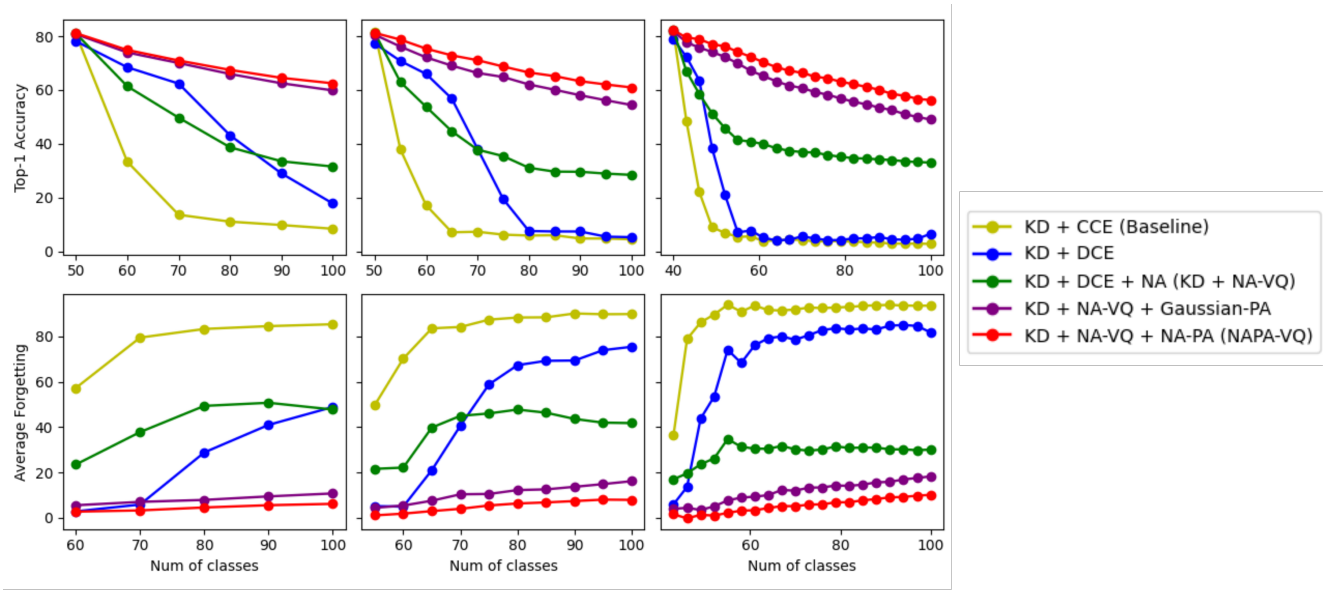

Figure 2. Detailed accuracy and forgetting curves for the ablation study

is due to the wider neighborhood being considered during topology approximation resulting in a larger number of updates during the CV adaptation. To maintain algorithm efficiency without compromising performance, $K = 15$ was determined to be desirable.

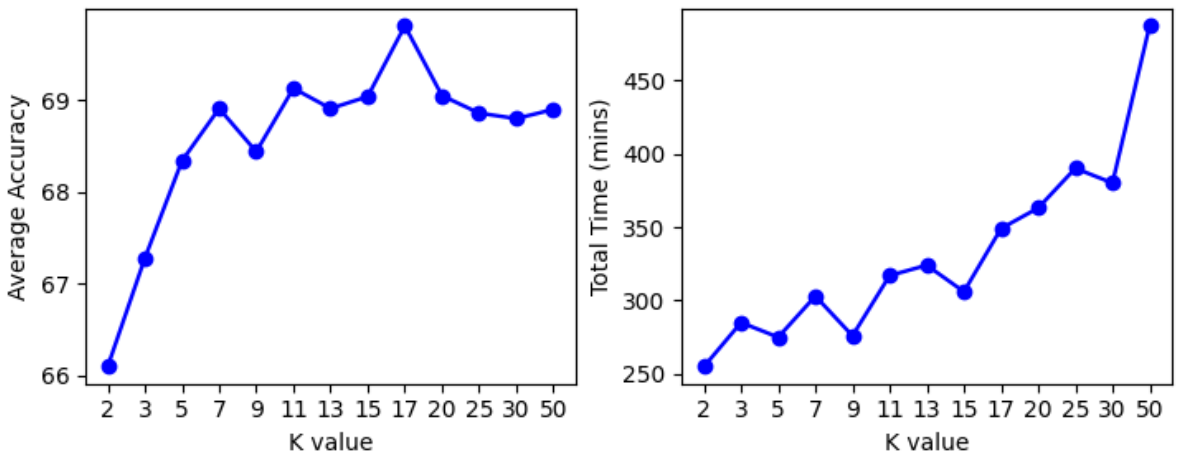

Figure 3. The variation of average accuracy and the total training time when the K-value is varied between 2 and 50.

2.7. Visualization of the NA-PA-generated prototypes

In order to demonstrate the effectiveness of the prototypes generated, we utilize t-SNE [2] to visualize the genuine feature representations of the old classes with their corresponding NA-PA-generated prototypes. It should be noted that the genuine feature representations of these old classes are not accessible in current or subsequent task training thus the prototypes generated for each old class should ideally approximate their true feature distributions. The obtained visualization is then compared with a set of prototypes generated using the Gaussian augmentation technique used in previous works [9]. As depicted in Fig. 4, the prototypes generated for each old class using Gaussian augmentation are clustered around the corresponding class center (mean prototype) and fail to account for the variations within the class. However, the prototypes generated using

NA-PA exhibit a more dispersed distribution within each class's feature space aiding in better representing the old classes. Therefore with NA-PA, we can generate a set of prototypes that captures the variations in the old classes without having to store the covariance matrix of each class [8].

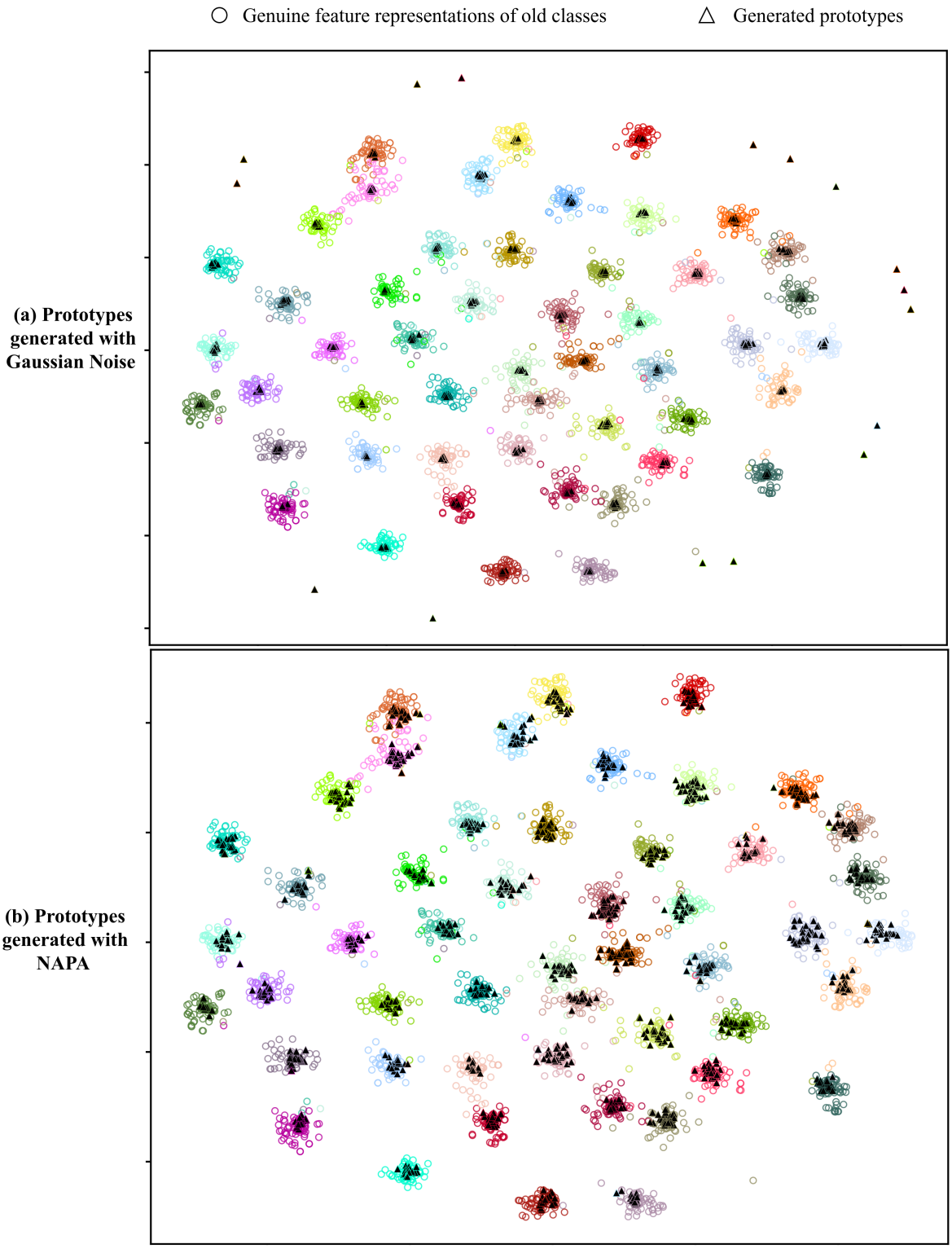

Figure 4. Visualization of the prototypes generated using Gaussian augmentation [9] and the proposed NA-PA technique. Each color represents a single class. (a) Prototypes generated using the Gaussian augmentation are located around the center of the class, failing to account for the variations within each class. (b) Prototypes generated using NA-PA are spread out in the feature space of each class producing high-impact prototypes that enables better discrimination between classes.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:770–778, 12 2016.

[2] Geoffrey E. Hinton and Sam Roweis. Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems*, 15, 2002.

[3] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526, 2017.

[4] Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More Classifiers, Less Forgetting: A Generic Multi-classifier Paradigm for Incremental Learning. In *Computer Vision–ECCV 2020: 16th European Conference*, pages 699–716, 2020.

[5] Sylvestre Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:5533–5542, 2017.

[6] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9374–9384, 2021.

[7] Lu Yu, Bartłomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6980–6989, 2020.

[8] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Class-Incremental Learning via Dual Augmentation. In *Advances in Neural Information Processing Systems*, number NeurIPS, pages 14306–14318, 2021.

[9] Fei Zhu, Xu-yao Zhang, Chuang Wang, Fei Yin, and Cheng-lin Liu. Prototype Augmentation and Self-Supervision for Incremental Learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 5267–5876, 2021.

[10] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022.