D^{3}FORMER: DEBIASED DUAL DISTILLED TRANSFORMER FOR INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In class incremental learning (CIL) setting, groups of classes are introduced to a model in each learning phase. The goal is to learn a unified model performant on all the classes observed so far. Given the recent popularity of Vision Transformers (ViTs) in conventional classification settings, an interesting question is to study their continual learning behaviour. In this work, we develop a Debiased Dual Distilled Transformer for CIL dubbed D^{3}Former. The proposed model leverages a hybrid nested ViT design to ensure data efficiency and scalability to small as well as large datasets. In contrast to a recent ViT based CIL approach, our D^{3}Former does not dynamically expand its architecture when new tasks are learned and remains suitable for a large number of incremental tasks. The improved CIL behaviour of D^{3}Former owes to two fundamental changes to the ViT design. First, we treat the incremental learning as a long-tail classification problem where the majority samples from new classes vastly outnumber the limited exemplars available for old classes. To avoid the bias against the minority old classes, we propose to dynamically adjust logits to emphasize on retaining the representations relevant to old tasks. Second, we propose to preserve the configuration of spatial attention maps as the learning progresses across tasks. This helps in reducing catastrophic forgetting by constraining the model to retain the attention on the most discriminative regions. D^{3}Former obtains favorable results on incremental versions of CIFAR-100, MNIST, SVHN, and ImageNet datasets.

1 INTRODUCTION

Real world data is ever evolving and new object categories appear over time. Therefore, it is desired to learn models that can incrementally update their knowledge when the new data arrives, without forgetting the past concepts. Existing deep learning models (LeCun et al., 2015; Schmidhuber, 2015) mostly consider a static world, where the learning happens once and if the model is trained on a new learning task, it catastrophically forgets the previously acquired knowledge (Kirkpatrick et al., 2017).

The goal of class incremental learning (CIL) is to continually learn new groups of classes (also referred to as tasks) without overwriting old task information (Joseph et al., 2022). The main challenge is to balance the stability-plasticity trade-off, i.e., the model should be able to adapt to new tasks (plastic but not to the point of forgetting) while retaining past knowledge (stable but not leading to intransigence) (Abraham & Robins, 2005). The previous works mostly concentrate on convolutional neural networks (CNNs) in incremental learning settings (Rebuffi et al., 2017; Hou et al., 2019; Liu et al., 2020; Yan et al., 2021). However, self-attention (Vaswani et al., 2017) based Vision Transformers (ViT) (Dosovitskiy et al., 2020) have been shown to outperform CNNs on conventional classification settings (Khan et al., 2021). Therefore, understanding the capabilities of ViTs for CIL is an interesting and open research question. In this work, our goal is to develop a ViT model tailored for incremental learning settings. While ViTs have excelled in large data regimes, their plain versions lack the necessary inductive biases, thereby perform poorly on small datasets as compared to CNNs. This problem intensifies in incremental learning, where the new task dataset is generally much smaller than a typical full training set. A recent approach DyTox (Douillard et al., 2022) proposes the first incremental learning transformer model, however it has a dynamically expandable architecture which grows as the new tasks are learned.

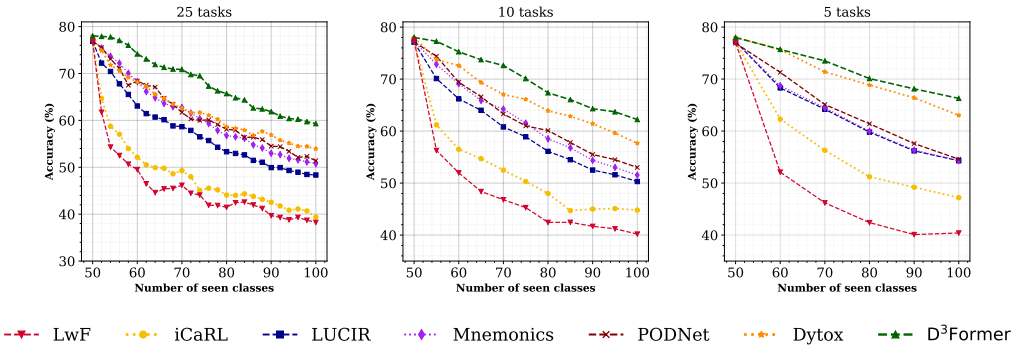

Figure 1: **D^{3}Former performance on small scale datasets:** Plots showing task wise accuracy for different number of incremental tasks for CIFAR-100. D^{3}Former achieves relatively high accuracy compared to other state-of-the-art methods when adding 2, 5 and 10 classes per task. We present ImageNet-1K results in Tab. 3, where we see a similar trend. Ours is the first transformer based incremental learning method, that scales well to small-scale and large-scale datasets alike.

We propose a hybrid ViT model for Incremental learning called D^{3}Former (Debiased Dual Distilled Transformer). D^{3}Former is data efficient and can be used equally well for both large and small-scale datasets (Fig. 1). The hybrid ViT designs (Liu et al., 2021b; Zhang et al., 2022b; Vaswani et al., 2021; Hassani et al., 2021) have proved to be more successful compared to pure self-attention based ViT designs at a lower computational cost. Specifically, our approach is based on a Nested Vision Transformer (Zhang et al., 2022b), that uses local self-attention within the patches and then hierarchically aggregates non-local information via convolution and pooling operations. The benefit manifests via improved data efficiency which is important for the incremental training where each task episode has a limited data belonging to a relatively small group of classes.

In order to render the ViT amenable to incremental setting, we propose two key modifications to minimize catastrophic forgetting. (a) *Debiasing via Logit Adjustment:* In the incremental phases, usually a small exemplar set of old task data is maintained due to memory constraints (Rebuffi et al., 2017). Since the classes in exemplar set are heavily imbalanced w.r.t the new task data, it bias the model against the previously observed classes. We propose a simple logit adjustment strategy to put appropriate emphasis on the previous task classes to avoid representational and classifier biasness. (b) *Dual Distillation:* In addition to the regular distillation loss applied on the logits / features (Rebuffi et al., 2017; Hou et al., 2019), we propose to maintain the attention cast on the input image by the teacher model and the student model to be consistent as the incremental learning progresses. To this end, we leverage the visual interpretability properties of Nested Transformer to obtain gradient based class activation maps (GradCAM) that are enforced to be consistent during incremental learning.

In summary, the main highlights of our approach are:

- We develop the first hybrid Transformer model for incremental settings, that can adaptively learn new task distributions. In comparison to state of the art methods (Yan et al., 2021; Douillard et al., 2022; Rajasegaran et al., 2019), our approach performs favorably well, as shown in Fig. 1, even without dynamically expanding its parameters as the number of tasks grow, making it scale easily.
- Owing to the inherent long tail distribution in CIL, our debiased loss formulation allocates high emphasis to the imbalanced data from old tasks, thereby minimizing loss of information relevant to previous tasks.
- We show that maintaining the attention on regions that are most crucial for predicting a particular class helps avoid overwriting the important features during incremental learning.
- Our extensive results on CIFAR-100, MNIST, SVHN and ImageNet datasets demonstrate considerable gains over the recent top performing incremental learning methods in terms of average and final task accuracies, as well as minimizing the forgetness measure.

2 RELATED WORK

2.1 INCREMENTAL LEARNING

We focus on class-incremental learning, where new classes are introduced to the model in distinct training phases. The methods are usually grouped into the following heads -

Regularization based: Knowledge distillation (Hinton et al., 2015) has been extensively used as a regularizer to minimise the changes to the decision boundaries of previous classes while learning incrementally. The model trained until the earlier phases of learning is treated as a teacher network, whose penultimate features or the logits are distilled into the incremental model. This was introduced in LwF (Li & Hoiem, 2016) and has been widely adopted by later methods. iCaRL (Rebuffi et al., 2017) uses KL Divergence loss for knowledge distillation. LUCIR (Hou et al., 2019) uses cosine similarity based loss for knowledge distillation and margin ranking loss for the hard examples. PODNet (Douillard et al., 2020) uses pooling as a means of restricting change. LwM (Dhar et al., 2019) and RRR (Ebrahimi et al., 2021) encourages the model to remember by making use of explanability techniques.

Replay based: In memory replay based methods, a small subset of data from the older classes are retained and replayed while learning the later incremental phases. This helps to alleviate the distributional shift caused while learning the new classes (Rebuffi et al., 2017; Liu et al., 2021a; Hou et al., 2019). Examples to be stored in the replay buffer may be randomly selected across all tasks (Riemer et al., 2019; Wu et al., 2019b), randomly selected per task (Joseph et al., 2021; Kj et al., 2021), by selecting an optimal coreset based on gradient statistics (Tiwari et al., 2022) or even by solving a submodular objective (Brahma & Othon, 2018). An alternative to storing exemplars would be to learn the distribution of the data using generative models and replaying the generated pseudo images (Shin et al., 2017). We refer reader to (Verwimp et al., 2021) for a more detailed treatment on replay-based continual learning methods. Replay based methods undesirably introduce bias towards new classes due to class imbalance. BiC (Wu et al., 2019a) learns an MLP explicitly to correct the bias, while SS-IL (Ahn et al., 2021) uses task-wise distillation along with separate heads for the current and previous tasks. (Jodelet et al., 2021) proposes balancing softmax outputs to reduce bias.

Structure based: Structure based methods usually allocate additional parameters for every new incremental phase. RPSNet (Rajasegaran et al., 2019), learns different paths for different tasks, ensuring weight sharing among tasks. DER (Yan et al., 2021) adds a new feature extractor for every task and uses pruning to reduce model size. A recent ViT based method DyTox (Douillard et al., 2022), proposes to use a dynamic task-token expansion based method to facilitate incremental learning.

2.2 VISION TRANSFORMERS

Self-attention based Transformer architecture (Vaswani et al., 2017) has revolutionized NLP. Vision Transformer (ViT) (Dosovitskiy et al., 2020) has helped to carry-over the successes from the NLP community to computer vision. Some of the notable ViT architecture include DeiT (Touvron et al., 2021) which uses knowledge distillation from a convolutional neural network through a distillation token, T2T ViT (Yuan et al., 2021) which tries to preserve local structure and reduce number of tokens by aggregating neighbouring tokens, XCiT (El-Nouby et al., 2021) which performs self-attention across feature channels to counter the quadratic complexity associated with self-attention between tokens. Recently, several hybrid ViTs – which use convolution layers along with self-attention – have been introduced. CvT (Wu et al., 2021) , CCT (Hassani et al., 2021), Swin (Liu et al., 2021b) and Nested Transformer (NesT) (Zhang et al., 2022b) are among the popular hybrid ViTs. To the best of our knowledge, ours is the first method that makes use of a hybrid ViT architecture for continual learning.

3 D^{3}FORMER: DEBIASED DUAL DISTILLED TRANSFORMER

Incrementally learning a classifier to expand its knowledge, without hampering its performance on the earlier set of classes is an arduous task for deep learning models. In our work, D^{3}Former, we

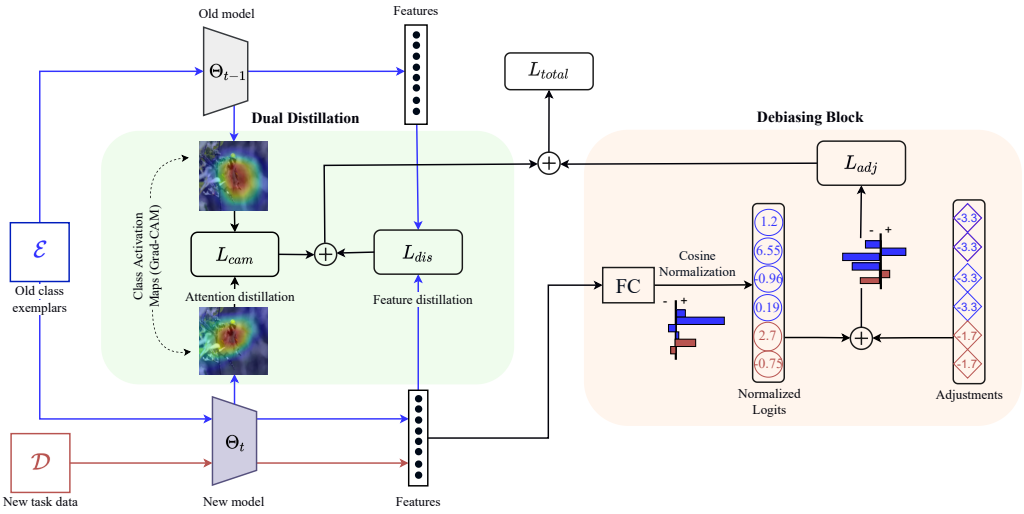

Figure 2: **(a)** Dual distillation: *(left)* In each learning phase t, the previous phase model $\boldsymbol{\Theta_{t-1}}$ is used to extract the features and Grad-CAMs of exemplars $\mathcal{E}$. Later, these attention maps are compared with the current model $\boldsymbol{\Theta_t}$ attention maps and L_{cam} loss is calculated between them. It discourages changes to the spatial attention response of $\boldsymbol{\Theta_t}$ w.r.t old classes. Knowledge distillation loss (L_{dis}) is computed as the cosine similarity between the features of $\mathcal{E}$ from $\boldsymbol{\Theta_{t-1}}$ and $\boldsymbol{\Theta_t}$. This maintains the orientation of the feature vectors for the old classes. **(b)** Debiasing block: (*right*) To compensate for bias towards new classes, in addition to cosine normalization of the logits, adjustments are added to the logits before applying cross-entropy. The adjusted logits result in stronger updates for the old (rare) classes to avoid their misclassification, thereby minimizing forgetting old task knowledge.

propose to make use of a hybrid model – that utilises the complementary advantages of transformer architectures and convolutional network – for class incremental learning. We detail about the model architecture in Sec. 3.1. Exemplar replay has emerged as a simple yet effective method to alleviate forgetting. Due to storage limitations, we store only few examples (close to 20 examples per task) in the exemplar memory. While learning a new task, we combine the data from exemplar memory with the incoming data. This skews the training data towards the latest task. We propose to address this imbalance by treating this setting as a long-tailed recognition problem, as explained in Sec. 3.2. Further, in Sec. 3.3, we propose to retain the spatial attention of exemplar images across tasks. This has a two fold effect: firstly, it improves the spatial awareness of the model; secondly, it helps to reduce forgetting by reminding the model on how it needs to attend to the more discriminative parts of the images during incremental learning. Concretely, let us consider learning a model $\boldsymbol{\Theta}$ across a total of $N+1$ training phases, where the first phase ($t=0$) involves learning a set of B base classes, followed by N incremental phases. Each phase ($1 \leq t \leq N$) involves learning a fixed number of C new classes. Consider the number of exemplars retained for each class in the previous tasks ($0 \ldots t-1$) is M, thereby forming a set $\mathcal{E} = \{\mathcal{E}_0, ..\mathcal{E}_{t-1}\}$. Thus, in the incremental phases, the model $\boldsymbol{\Theta}$ is trained using the replayed old class exemplars $\mathcal{E}$ and all new input classes data $\mathcal{D}$. Figure 2 illustrates the overall setup and the different loss functions used in D^{3}Former. L_{cam} and L_{dis} enforces the current model to not deviate much from the previous model, while a cross entropy loss on the adjusted logits (L_{adj}) helps to learn the new task. We explain more on these in the following sub-sections.

3.1 The Hybrid ViT Model

D^{3}Former builds upon the hybrid ViT NesT (Zhang et al., 2022b) which makes use of 2 basic operations - blockify and aggregation. The blockify operation combines spatially adjacent embeddings into a group. It captures intra-block information or local attention using several stacked transformer encoders. Each transformer encoder consists of Layer Normalization (LN) and Multi-head self-attention (MSA) followed by Feed-Forward network (FFN). On the other hand, the aggregation operation (AGG) combines neighboring blocks with the help of a simple convolution and pooling

layer. It captures inter-block relationships and helps gain global understanding of an image. The local and global processing steps help learn discriminative features.
The above operations are repeated alternately to eventually create the hierarchical structure of NesT (Fig. 3), where each hierarchy shares the same set of parameters. The final class prediction is performed through a global average pooling (GAP) layer followed by a fully connected (FC) layer. NesT is characterized by two parameters, patch size, S and number of block hierarchies, T_d. To render NesT suitable for CIL, we propose two principal modifications - Debiasing via Logit Adjustment and Dual Distillation.

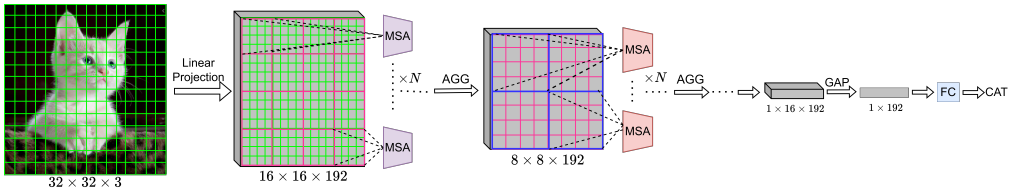

Figure 3: Nested Transformer (NesT) architecture illustrating blockify and aggregation operations.

3.2 Reducing the Bias in the Logits

In incremental phases, a small set of exemplars are usually stored for old tasks data due to memory constraints. However, current task samples outnumber old tasks exemplars in each phase leading to a strong bias towards new classes.
An intuitive approach to reduce bias involves placing more emphasis on rare classes during the learning process. This can be easily implemented using a simple logit adjustment strategy (Menon et al., 2021). Logit adjustment adds an appropriate offset to the output logits thereby increasing the margin between rare and frequent classes. The offset can be calculated as $\tau \log \pi_y$, where τ is a hyperparameter that controls the adjustment strength, π_y is the estimated prior for class y. Class priors are approximated as the frequency of each class in the dataset. However, in our case, since the number of exemplars from the old classes are equal and the number of samples from each new classes samples are also equal, there needs to be only two class priors $\{\pi_o, \pi_n\}$. The class priors for old and new classes are calculated as follows:

$$\pi_o = \frac{|\mathcal{E}_{c_o}|}{|\mathcal{E}| + |\mathcal{D}|}, \forall c_o \in \mathcal{E}, \;\; \pi_n = \frac{|\mathcal{D}_{c_n}|}{|\mathcal{E}| + |\mathcal{D}|}, \forall c_n \in \mathcal{D} \tag{1}$$

Where c_o are the old classes, and c_n are the new classes. Thus, the cross-entropy loss can be modified by including the logit adjustment offsets as:

$$L_{adj}(\boldsymbol{x}) = -\log \frac{e^{f_y(\boldsymbol{x}) + \tau \log \pi_y}}{\sum_{y' \in \mathcal{T}} e^{f_{y'}(\boldsymbol{x}) + \tau \log \pi_{y'}}}, \;\; s.t., \pi_y, \pi_{y'} \in \{\pi_o, \pi_n\}, \tag{2}$$

where $\mathcal{T}$ is the class labels set, and $f_y(\boldsymbol{x})$ is the cosine normalized logits for an input sample $\boldsymbol{x}$. Cosine normalization helps in further reducing bias towards new classes samples (Hou et al., 2019; Luo et al., 2018), and computed as:

$$f_y(\boldsymbol{x}) = \eta \langle \bar{\theta}(\boldsymbol{x}), \bar{w} \rangle, \tag{3}$$

where η is a learnable scaling parameter to control the peakness of the logits for softmax, as the values after normalization are between $[-1, 1]$, $\bar{\theta}(\boldsymbol{x})$ is the L_2-normalized extracted features, and $\bar{w}$ denotes the final layer L_2-normalized weights.

3.3 Dual-distillation Framework

Knowledge distillation was introduced to CIL as a means of reducing forgetting by transferring knowledge about old tasks from the teacher model $\boldsymbol{\Theta_{t-1}}$ to the student model $\boldsymbol{\Theta_t}$ (Li & Hoiem, 2016; Rebuffi et al., 2017). First, we incorporate knowledge distillation at feature-level (Hou et al., 2019) using a cosine similarity loss based on feature vectors computed as follows :

$$L_{dis} = 1 - \langle \bar{\theta}_{t-1}(\boldsymbol{x}), \bar{\theta}_t(\boldsymbol{x}) \rangle, \tag{4}$$

Method	*N=5*			*N=10*			*N=25*		
	Avg ↑	Last ↑	$\mathcal{F}$ ↓	Avg ↑	Last ↑	$\mathcal{F}$ ↓	Avg ↑	Last ↑	$\mathcal{F}$ ↓
LwF(Li & Hoiem, 2016)	49.59	40.40	43.36	46.98	40.19	43.58	45.51	38.25	41.66
BiC(Wu et al., 2019a)	59.36	-	31.42	54.20	-	32.50	50.00	-	34.60
iCaRL (Rebuffi et al., 2017)	57.12$_{\pm 0.50}$	47.20	31.88	52.66$_{\pm 0.89}$	44.80	34.10	48.22$_{\pm 0.76}$	39.39	36.48
LUCIR (Hou et al., 2019)	63.17$_{\pm 0.87}$	54.30	18.70	60.14$_{\pm 0.73}$	50.30	21.34	57.54$_{\pm 0.43}$	48.35	26.46
Mnemonics (Liu et al., 2020)	63.34$_{\pm 0.34}$	54.32	**10.91**	62.28$_{\pm 0.61}$	51.53	**13.38**	60.96$_{\pm 0.72}$	50.78	**19.80**
PODNet-CNN (Douillard et al., 2020)	64.83$_{\pm 1.11}$	54.60	-	63.19$_{\pm 1.31}$	53.00	-	60.72$_{\pm 1.54}$	51.40	-
DyTox* (Douillard et al., 2022)	70.28	63.02	24.54	66.72	59.62	29.86	62.83	53.95	33.72
D^{3}Former (ours)	**72.23**$_{\pm 0.08}$	**66.24**$_{\pm 0.1}$	12.09	**70.94**$_{\pm 0.43}$	**63.10**$_{\pm 0.54}$	16.12	**68.68**$_{\pm 0.4}$	**59.79**$_{\pm 0.44}$	21.23
D^{3}Former-NCM (ours)	71.38$_{\pm 0.32}$	64.26$_{\pm 0.47}$	16.52	69.35$_{\pm 0.47}$	61.46$_{\pm 0.58}$	19.36	67.03$_{\pm 0.59}$	58.12$_{\pm 0.80}$	22.84

Table 1: Results of **CIFAR-100** with Average accuracy (%), last phase accuracy (%) and forgetting rate $\mathcal{F}$(%) of different methods in 5,10 and 25 tasks settings. The top group of methods are based on CNN while the last three approaches (including ours) are based on transformer models. * indicates results reproduced by us using author's official codebase.

where $\bar{\theta}_{t-1}(\boldsymbol{x})$, $\bar{\theta}_t(\boldsymbol{x})$ denote the normalized feature vectors extracted from models $\boldsymbol{\Theta_{t-1}}$ and $\boldsymbol{\Theta_t}$, respectively.
In addition to Eq. 4 which preserves the orientation of feature vectors as incremental learning progresses, preserving the model response on regions that are critical for predicting a particular class can help further reduce catastrophic forgetting (Dhar et al., 2019; Ebrahimi et al., 2021). The enhanced visual interpretability of NesT (Zhang et al., 2022b) allows us to obtain these salient regions by using gradient based class activation maps (Grad-CAM) (Selvaraju et al., 2019). Grad-CAMs are essentially the heatmaps which localize the most discriminative regions for a particular class in a given image. We enforce that the attention response of $\boldsymbol{\Theta_t}$ on the old tasks must be maintained similar to that of $\boldsymbol{\Theta_{t-1}}$ through a Grad-CAM based L_1 distillation loss:

$$L_{cam}(x) = \| \, CAM(\boldsymbol{\Theta_t}, \boldsymbol{x}) - CAM(\boldsymbol{\Theta_{t-1}}, \boldsymbol{x}) \, \|_1 \tag{5}$$

We obtain Grad-CAMs from the feature maps of the final hierarchy in NesT, since it contains global information of the whole image. The total loss can thus be written as:

$$L_{total} = \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}} L_{adj}(\boldsymbol{x}) + \frac{\lambda}{|\mathcal{N}_o|} \sum_{y \in \mathcal{N}_o} L_{dis}(\boldsymbol{x}) + \frac{\gamma}{|\mathcal{N}_o|} \sum_{y \in \mathcal{N}_o} L_{cam}(\boldsymbol{x}), \tag{6}$$

where y is the class label of sample $\boldsymbol{x}$, $\mathcal{N}_o$ denotes the set of old classes, $\mathcal{T}$ is the set of all classes, λ is a scaling factor controlling cosine similarity based knowledge distillation and γ is a scaling factor controlling the magnitude of Grad-CAM based distillation.

4 Experiments

We analyze the performance of D^{3}Former on large scale datasets such as ImageNet-1K(Russakovsky et al., 2015), ImageNet Subset-100 and small scale datasets like MNIST(Deng, 2012), SVHN(Goodfellow et al., 2013) and CIFAR-100(Krizhevsky et al., 2009). MNIST contains 28×28 pixel grayscale images of handwritten single digits between 0 and 9, SVHN is a house numbers digit dataset with 32×32 images of 10 classes and CIFAR-100 has 32×32 images with 100 classes. We follow a setting where we initially train the model for half the number of classes(Rebuffi et al., 2017; Hou et al., 2019; Douillard et al., 2020) and then incrementally add 2, 5 and 10 classes in each task for ImageNet and CIFAR-100 experiments. A strict memory budget is considered where only 20 exemplars per class are stored. For MNIST and SVHN experiments, we always add 2 classes per task with a fixed exemplar memory of 4.4k as followed in (Rajasegaran et al., 2019).

4.1 Implementation Details

Small-scale Datasets: NesT-tiny architecture with a configuration of $S = 1$ is used for CIFAR-100 experiments, while for SVHN and MNIST we use $S = 2$. The embedding dimension is set to 192, the number of hierarchy levels is 3, the number of transformer encoder blocks per level is 4 and

the number of heads in each level is 6. Augmentations such as random erasing, cutmix, mixup and random augment are used as suggested in (Zhang et al., 2022b). However, mixup is not used in the incremental phases. The suitable choices of hyper-parameters found empirically are $\lambda = 7$, $\tau = 1$ and $\gamma = 0.1$. We use a batch size of 128 and observe that performing distillation only over memory samples is more favorable.

ImageNet: We use NesT-tiny architecture for ImageNet experiments too. We set $S = 4$, embedding dimensions is set to $(96, 192, 384)$, the number of hierarchy levels are 3, the number of transformer encoder blocks per level are $(3, 6, 12)$ and the number of heads per level are $(2, 2, 8)$. Augmentations such as random erasing, cutmix, mixup and random augment are used as suggested in (Zhang et al., 2022b). Mixup is also used in the incremental phases. Empirically, we find that the hyper-parameters when set to $\lambda = 4$, $\tau = 0.3$ and $\gamma = 0.05$ yield the best results. We observe that performing feature distillation over all samples provides more stability when training NesT on ImageNet. We use a batch size of 384 for ImageNet-100 and 1024 for ImageNet-1K.
For both small and large scale datasets, the model is trained for 250 epochs, 150 epochs in case of 2 classes per phase. Weighted Adam (Loshchilov & Hutter, 2019) is used as the optimizer. The learning rate starts from $2.5\text{e}-4$ and decays following cosine annealing scheduler. We make use of PyTorch implementation of NesT from timm library (Wightman, 2019) and train on an RTX A6000 GPU.

Method	*N=5*			*N=10*			*N=25*		
	Avg ↑	Last ↑	$\mathcal{F}$ ↓	Avg ↑	Last ↑	$\mathcal{F}$ ↓	Avg ↑	Last ↑	$\mathcal{F}$ ↓
DyTox Joint	-	79.82	-	-	79.82	-	-	79.82	-
D^{3}Former Joint	-	82.14	-	-	82.14	-	-	82.14	-
LwF (Li & Hoiem, 2016)	53.62	40.10	55.32	47.64	36.10	57.00	44.32	34.12	55.12
BiC (Wu et al., 2019a)	70.07	-	27.04	64.96	-	31.04	57.73	-	37.88
iCaRL (Rebuffi et al., 2017)	65.44$\pm$ 0.35	53.60	43.40	59.88$\pm$ 0.83	49.10	45.84	52.97$\pm$ 1.02	43.34	47.60
LUCIR (Hou et al., 2019)	70.84$\pm$ 0.69	60.00	31.88	68.32$\pm$ 0.81	57.10	33.48	61.44$\pm$ 0.91	49.26	35.40
Mnemonics (Liu et al., 2020)	75.54$\pm$ 0.85	61.36	**17.40**	74.33$\pm$ 0.56	59.56	**17.08**	68.31$\pm$ 0.39	59.22	**20.83**
PODNet-CNN (Douillard et al., 2020)	76.96$\pm$ 0.29	67.60	-	73.70$\pm$ 1.05	65.00	-	71.78$\pm$ 2.77	54.30	-
DyTox* (Douillard et al., 2022)	77.08	**70.24**	21.21	74.06	**65.44**	27.16	68.76	**61.54**	30.04
D^{3}Former (ours)	**77.31**$\pm$ 0.41	67.82$\pm$ 0.36	25.92	75.01$\pm$ 0.63	63.46$\pm$ 0.32	27.41	**72.43**$\pm$ 0.76	59.91$\pm$ 1.1	30.80
D^{3}Former-NCM (ours)	77.21$\pm$ 0.22	69.89$\pm$ 0.18	17.98	**75.26**$\pm$ 0.28	65.11$\pm$ 0.25	20.21	72.31$\pm$ 0.24	60.01$\pm$ 0.85	27.20

Table 2: Results of **ImageNet100** with Average accuracy (%), last phase accuracy (%) and forgetting rate $\mathcal{F}$ (%) of different methods in 5,10 and 25 task settings. * indicates results reproduced by us using author's official codebase.

Method	*N=5*			*N=10*		
	Avg ↑	Last ↑	$\mathcal{F}$ ↓	Avg ↑	Last ↑	$\mathcal{F}$ ↓
DyTox Joint	-	73.58	-	-	73.58	
D^{3}Former Joint	-	76.42	-	-	76.42	
LwF (Li & Hoiem, 2016)	44.35	34.20	48.70	38.90	30.10	47.94
BiC (Wu et al., 2019a)	62.65	-	25.06	58.72	-	28.34
iCaRL (Rebuffi et al., 2017)	51.50 $\pm$ 0.43	34.20	26.03	46.89 $\pm$ 0.35	38.91	33.76
LUCIR (Hou et al., 2019)	64.45 $\pm$ 0.32	56.60	24.08	61.57 $\pm$ 0.23	51.7	27.29
Mnemonics (Liu et al., 2020)	64.54 $\pm$ 0.49	56.85	**13.85**	63.01$\pm$ 0.57	54.99	**15.82**
PODNet-CNN (Douillard et al., 2020)	66.43	58.90	-	63.21	55.70	-
DyTox* (Douillard et al., 2022)	68.96	64.08	18.63	67.12	57.61	31.83
D^{3}Former (ours)	**72.73**$\pm$ 0.30	64.58$\pm$ 0.33	21.41	69.56$\pm$ 0.29	59.22$\pm$ 0.32	32.35
D^{3}Former-NCM (ours)	72.61 $\pm$ 0.32	**64.64** $\pm$ 0.29	17.03	**70.04**$\pm$ 0.34	**59.90**$\pm$ 0.31	27.87

Table 3: Results of **ImageNet-1K** with Average accuracy (%), last phase accuracy (%) and forgetting rate $\mathcal{F}$ (%) of different methods in 5 and 10 tasks setting. * indicates results reproduced by us using author's official codebase.

4.2 RESULTS

We conduct exhaustive experimental analysis to test the mettle of our approach. We use three metrics to quantify the performance: 1) average accuracy across all phases, 2) accuracy of the last phase and 3) forgetting rate $\mathcal{F}$ defined as the difference between accuracy of $\mathbf{\Theta_0}$ and $\mathbf{\Theta_N}$ on the same test data $\mathcal{D}_0^{test}$ following (Liu et al., 2020). Further, following (Liu et al., 2021a), we either use the softmax predictions from the final classifier or use a nearest class mean based classifier (Rebuffi et al., 2017) during inference. We refer to these as D^{3}Former and D^{3}Former-NCM respectively in the results.

CIFAR-100: Tab. 1 and Fig. 1 summarizes the results on CIFAR-100 dataset when we add incrementally add 10, 5 and 2 classes respectively to a model trained on the first 50 classes. We observe that as the number of phases increases, the gap between D^{3}Former and the compared methods progressively increases – thanks to our dual-distillation and logit-correction mechanisms. Specifically, for 25 task experiment, our method improves average accuracy from 62.83% to 68.68% (+5.8%).

ImageNet: We summarize the results of incrementally learning ImageNet Subset-100 dataset in Tab. 2. We consider 5, 10 and 25 task incremental setting. Our method achieves the best average accuracy of 77.5% in the 5 phases settings and 72.43% in 25 phases settings and is comparable to (Douillard et al., 2020; 2022) in 10 phases setting. Tab. 3 summarizes ImageNet-1K results in 5 and 10 phase setting. Unlike small scale datasets, ImageNet shows relatively better performance while using NCM. The aforementioned behaviour is not present in previous CNN based methods (Douillard et al., 2020). This can be attributed to two factors: first, transformers have better generalization compared to CNNs (Zhang et al., 2022a), which results in better class means, second, NesT uses higher embedding dimension for large scale datasets which can help in NCM based classification.

MNIST, SVHN: Thanks to the better inductive biases of our hybrid architecture, D^{3}Former can scale to small datasets like MNIST and SVHN too. This uniquely differentiates us to recent efforts (Douillard et al., 2022; Yu et al., 2021) in utilizing transformer architecture for incremental learning. Tab. 4 summarizes the average accuracy results on these datasets by adding two new classes in every incremental phase. Our method clearly surpasses other methods by more than 2% for MNIST and 5% for SVHN dataset.

Table 4: Average accuracy (%) for MNIST, SVHN in 5 tasks setting with 2 classes each with 4.4k fixed memory ($*$ *indicates use of exemplars*)

Method	MNIST	SVHN
EWC (Kirkpatrick et al., 2016)	19.80	18.21
LwF (Li & Hoiem, 2016)	24.17	-
GEM* (Lopez-Paz & Ranzato, 2017)	92.20	75.61
RPS-Net* (Rajasegaran et al., 2019)	96.16	90.83
D^{3}Former * (ours)	**98.85**	**95.81**

4.3 DISCUSSIONS AND ANALYSIS

4.3.1 Contribution from Each Loss Terms: We analyse the contribution of each component in our loss formulation in Fig. 4. We observe that with just cosine distillation, NesT is able to achieve almost comparable accuracy as the baselines (Wu et al., 2019a; Rebuffi et al., 2017; Hou et al., 2019). The addition of logit adjustment offset alone brings about 1.5% - 2% improvement over using cosine distillation loss. We observe that Grad-CAM loss alone is not strong enough to boost the accuracy. This is because of the model's inability to handle abrupt changes in model parameters caused due to class imbalance. However, when combined with other losses, we observe considerable improvement.

4.3.2 Sensitivity Analysis on τ, γ, λ: There is a trade off between forgetting and learning while doing logit adjustment. As shown in Tab. 6, a high value of τ effectively reduces the forgetting, but puts much emphasis on old classes that hinders new learning. In contrast, a small value of τ does not have enough impact on retaining old classes. Table 7 clearly shows the benefit of using L_{cam} and L_{dis} in improving accuracy. For 5 tasks CIFAR-100 setting, τ=1, λ=7 and γ=0.1 obtains the best results.

4.3.3 On Data Used for Distillation: We study the effect of distilling from exemplars verses all the data-points here. Applying distillation on all data combined with debiasing techniques such as logit adjustment, could impede learning of new tasks. Although it helps in reducing catastrophic forgetting, it adds a lot of constraints on the learning of new classes. This becomes more prominent in case of small datasets, due to less number of learnable parameters. Tab. 5 shows the positive effect of only applying distillation on exemplars, which is intuitive.

Mixup	Distillation	$S = 1$	$S = 2$
✓	all samples	62.10	60.80
✓	exemplars	66.71	64.71
	exemplars	72.21	67.07

Table 5: Effect of using Mixup and distillation in incremental phases. Impact of different patch size S is also shown. The average accuracy for 5 task CIFAR-100 is reported.

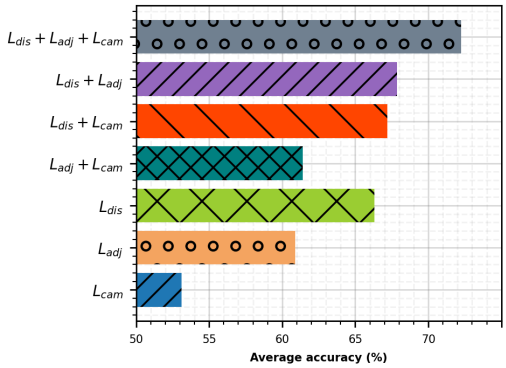

Figure 4: Ablation experiment on the contribution of each loss for CIFAR-100 (5 tasks).

τ	Avg ↑	Last ↑	$\mathcal{F}$ ↓
0	60.26	48.51	38.41
0.5	66.93	57.65	28.84
0.75	69.34	60.29	22.67
1	72.21	66.30	12.09
1.25	71.72	65.61	11.32
1.5	71.14	65.07	07.70

Table 6: Effect of varying τ in a 5 task CIFAR-100 setting.

γ	Avg ↑	Last ↑	λ	Avg ↑	Last ↑
0	67.85	60.16	0	57.34	48.67
0.05	71.97	66.39	5	71.78	66.21
0.10	72.21	66.24	7	72.35	66.36
0.15	72.03	66.25	9	72.17	66.57
0.20	71.81	66.16	12	71.95	66.46

Table 7: Effect of γ and λ in a 5 task CIFAR-100 setting.

4.3.4 Effect of Mixup: Our method uses mixup augmentation (Zhang et al., 2017) in the initial phase where half of the classes are learnt. However, we observe differences in performance when using mixup in incremental phases. For CIFAR-100, using mixup in incremental phases proves to be unfavorable. This is because distillation loss is indeed adding strong regularization for these small scale datasets. We see this trend in Tab. 5.

4.3.5 Generality of our Approach: We note that our proposed loss formulation (L_{dis}, L_{cam} and L_{adj}) is agnostic to the backbone network being used. To elucidate this, we swap the NesT backbone with a standard ResNet-18 backbone and report the result in Tab. 8 for 5 task Imagenet100 setting. We borrow the hyper-parameters for the ResNet backbone from AANet (Liu et al., 2021a) and use τ=0.3, λ=5 and γ=0.01. This shows that our proposed distillation and logit adjustments helps in reducing forgetting, however forgetting is much higher when compared to D^{3}Former.

Setting	Avg ↑	Last ↑	$\mathcal{F}$ ↓
ResNet + L_{dis}	68.52	55.83	34.81
ResNet + L_{dis}+L_{adj}	71.84	61.81	24.25
ResNet + L_{dis}+L_{adj}+L_{cam}	71.97	62.26	23.18

Table 8: Our proposed loss also shows improvement when applied to a ResNet-18 backbone

5 Conclusion

We propose D^{3}Former, a hybrid ViT based model that is tuned for class incremental learning. We propose two fundamental changes to effectively balance the stability and plasticity required for a continual learner: First, we view each incremental phase as a long tail distribution and show the effectiveness of a simple logit offset in reducing inherent bias towards new classes. Second, we show that preserving the spatial attention response of a model via distillation can help in improving the spatial awareness of the model and reduce catastrophic forgetting. D^{3}Former achieves superior performance gains over the state-of-the-art methods on MNIST, SVHN, CIFAR-100 and ImageNet. We hope our approach can serve as a simple baseline for incremental hybrid ViTs.

REFERENCES

Wickliffe C Abraham and Anthony Robins. Memory retention–the synaptic stability versus plasticity dilemma. *Trends in neurosciences*, 28(2):73–78, 2005.

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental. *In ICCV*, 2021.

Pratik Prabhanjan Brahma and Adrienne Othon. Subset replay based continual learning for scalable improvement of autonomous systems. In *In CVPR Workshops*, pp. 1179–11798. IEEE, 2018.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. *In CVPR*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *In ICLR*, 2020.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. *In ECCV*, 2020.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. *In CVPR*, 2022.

Sayna Ebrahimi, Suzanne Petryk, Akash Gokul, William Gan, Joseph E. Gonzalez, Marcus Rohrbach, and Trevor Darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. *In ICLR*, 2021.

Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jegou. Xcit: Cross-covariance image transformers. *In NeurIPS*, 2021.

Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv:2104.05704*, 2021.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *In CVPR*, June 2019.

Quentin Jodelet, Xin Liu, and Tsuyoshi Murata. Balanced softmax cross-entropy for incremental learning. In *International Conference on Artificial Neural Networks*, pp. 385–396. Springer, 2021.

K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *In CVPR*, 2021.

KJ Joseph, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwar, and Vineeth Balasubramanian. Energy-based latent aligner for incremental learning. *In CVPR*, 2022.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 2021.

James Kirkpatrick, Razvan Pascanu, Neil C. 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. *In CoRR*, abs/1612.00796, 2016.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Joseph Kj, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *In TPAMI*, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *In ECCV*, 2016.

Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *In CVPR*. IEEE, jun 2020. doi: 10.1109/cvpr42600.2020.01226. URL https://doi.org/10.1109%2Fcvpr42600.2020.01226.

Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. *In CVPR*, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *In ICCV*, 2021b.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continuum learning. *In CoRR*, abs/1706.08840, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *In ICLR*, 2019.

Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *International Conference on Artificial Neural Networks*, pp. 382–391. Springer, 2018.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *In ICLR*, 2021.

Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *In CVPR*, pp. 5822–5830, 2018.

Jathushan Rajasegaran, Munawar Hayat, Salman Khan, Fahad Shahbaz Khan, Ling Shao, and Ming-Hsuan Yang. An adaptive random path selection approach for incremental learning. *In NeurIPS*, 2019.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *In CVPR*, 2017.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *In ICLR*, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *In IJCV*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *In IJCV*, 128(2):336–359, oct 2019. doi: 10.1007/s11263-019-01228-7. URL https://doi.org/10.1007%2Fs11263-019-01228-7.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *In NeurIPS*, 30, 2017.

Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. *In CVPR*, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers &; distillation through attention. *In ICML*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *In NeurIPS*, 2017.

Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *In CVPR*, pp. 12894–12904, 2021.

Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. *arXiv:2104.07446*, 2021.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *In ICCV*, 2021.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. *In CVPR*, 2019a.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *In CVPR*, pp. 374–382, 2019b.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. *In CVPR*, 2021.

Pei Yu, Yinpeng Chen, Ying Jin, and Zicheng Liu. Improving vision transformers for incremental learning. *arXiv:2112.06103*, 2021.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *In ICCV*, 2021.

Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. *In CVPR*, 2022a.

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

Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan O. Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. *In AAAI*, 2022b.

A APPENDIX

A.1 TRAINING DETAILS

Zeroth phase: The zeroth phase training starts with 10 warm up epochs for small scale datasets, and 20 epochs for large scale datasets. The learning rate η starts with 2.5e-4 and increases as a factor of current step number and current epoch as follows:

$$\eta = \eta \times epoch \times step, \tag{7}$$

which reaches a maximum of 1.0 and then drops back to 2.5e-4. For the rest of the epochs, the learning rate decays following cosine annealing scheduler till it reaches zero on the last epoch. For data augmentation, mixup, random augmentation and random erasing are used for both large and small scale datasets.

Incremental phases: In each incremental phase, the new classes classifier weights are initialized following weight imprinting introduced in (Qi et al., 2018), old classes classifier weights are frozen. The learning rate starts from 2.5e-4 for the feature extractor and 2.5e-3 for the classifier. Both learning rates follow a cosine annealing scheduler that decays the weight till it reaches zero at the final epoch. The number of epochs for each phase is 250 in case of 10 classes per task and 5 classes per task, while for 2 classes per task the number of epochs is kept at 150.

Knowledge distillation factor λ is increased every phase as a factor of number of classes as follows:

$$\lambda_t = \lambda_{t-1} \times \sqrt{\frac{B+C}{C}}, \tag{8}$$

Where B is the number of base classes, and C is the number of new added classes every phase. The classes exemplars are chosen following the same herding method of (Rebuffi et al., 2017).

A.2 EFFECT OF AUGMENTATIONS

NesT uses augmentations such as Mixup, RandomErasing and RandAugment. These augmentations have been shown to be useful in stabilizing training and improve performance of hybrid ViTs (Liu et al., 2021b; Zhang et al., 2022b). The importance of these augmentations has also been discussed in the NesT paper. We show the effect of these augmentations when used in the incremental phases.

Table 9: Effect of augmentations when used in incremental phases of 5 tasks setting for CIFAR100

Augmentations	**Average accuracy**
With Mixup	71.89
With Randaug, RandomErasing	71.84
With all augmentations	72.33

A.3 QUALITATIVE ANALYSIS

Figure 5 shows some qualitative results in the form of Grad-CAMs with increasing number of incremental tasks. It is observed that the model does not forget much and makes use of the discriminatory regions in an image to make the correct prediction.

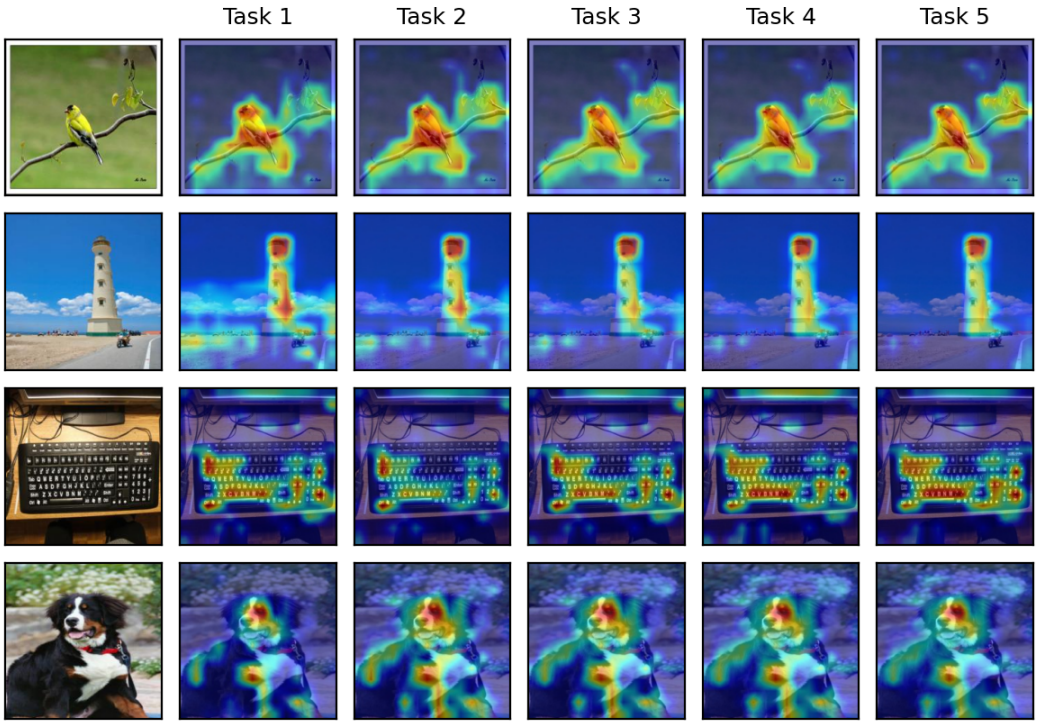

Figure 5: Grad-CAMs for images from ImageNet subset-100 as incremental learning progresses. This shows that Grad-CAM distillation helps D^3Former maintain attention on discriminative patches. (*figure best viewed with zoom-in*)