Doodle It Yourself: Class Incremental Learning by Drawing a Few Sketches

Ayan Kumar Bhunia[1] Viswanatha Reddy Gajjala* Subhadeep Koley[1,2] Rohit Kundu*
Aneeshan Sain[1,2] Tao Xiang[1,2] Yi-Zhe Song[1,2]
[1]SketchX, CVSSP, University of Surrey, United Kingdom.
[2]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.
{a.bhunia, s.koley, a.sain, t.xiang, y.song}@surrey.ac.uk; viswanathareddy998@gmail.com

Abstract

The human visual system is remarkable in learning new visual concepts from just a few examples. This is precisely the goal behind few-shot class incremental learning (FSCIL), where the emphasis is additionally placed on ensuring the model does not suffer from "forgetting". In this paper, we push the boundary further for FSCIL by addressing two key questions that bottleneck its ubiquitous application (i) can the model learn from diverse modalities other than just photo (as humans do), and (ii) what if photos are not readily accessible (due to ethical and privacy constraints). Our key innovation lies in advocating the use of sketches as a new modality for class support. The product is a "Doodle It Yourself" (DIY) FSCIL framework where the users can freely sketch a few examples of a novel class for the model to learn to recognise photos of that class. For that, we present a framework that infuses (i) gradient consensus for domain invariant learning, (ii) knowledge distillation for preserving old class information, and (iii) graph attention networks for message passing between old and novel classes. We experimentally show that sketches are better class support than text in the context of FSCIL, echoing findings elsewhere in the sketching literature.

1. Introduction

Fully supervised learning has served us great with performances on ImageNet already surpassing human-level [18]. In reality, however, such progress is primarily limited to a small number of object classes where labels were explicitly curated (1000 in ImageNet vs. possibly millions out there). Class Incremental Learning [29, 21, 23] is one of the popular fronts that attempt to extend model perception to novel classes while not "forgetting" about classes learned already. Amongst its many variants, the recent Few-Shot Class Incremental Learning (FSCIL) [54] is the most realistic where it also dictates the model to learn new classes with very few examples, the same as humans do.

*Interned with SketchX

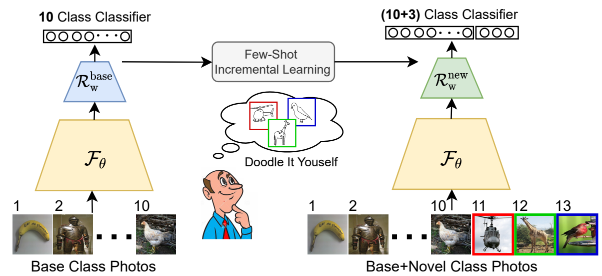

Figure 1. Illustration of our DIY-FSCIL framework. For instance, given *sketch* exemplars (1-shot here) from 3 novel classes as support-set, a 10-class classifier gets updated to $(10+3)$-class classifier that can classify photos from both base and novel classes.

As easy as providing a few samples might sound, questions start to emerge in practice as to (i) what data modality should the samples take? and (ii) how could these samples be obtained in practice. These questions, we argue, are key to the potentially ubiquitous application of FSCIL as (i) humans also learn from a broad range of data modalities that are not limited to just photo, and (ii) there are scenarios where photos are not necessarily always readily available due to privacy and ethical constraints (e.g., copyright).

In this paper, we set out to study the role of human sketches as a support modality for FSCIL. This results in a flexible FSCIL system that learns new classes just by observing a few sketches *doodled* by users themselves. Fig. 1 illustrates schematically our "Doodle It Yourself (DIY)" FSCIL scenario – "DIY-FSCIL". This importantly addresses the aforementioned problems in that (i) learning is no longer fixed to just photos but flexibly cross-modal with other data forms (just as humans do), and (ii) it works without asking the users to source photos which might have practical constraints attached (e.g., copyright, hazardous environments). There is of course also the added benefit of injecting creativity to the classifier by sketching something off the user's imagination [16], e.g., a "flying cow"?

The advocate of sketches is largely motivated by the line of work examining human-centric characteristics of sketches in many parallel applications – notably image retrieval [12], where the fine-grained nature of sketches is used to successfully conduct instance-level retrieval [2, 6, 46, 14, 11]. Sketches in context of FSCIL is closely reminiscent of its usage in fine-grained retrieval. While in retrieval they utilise the detailed nature of sketches to conduct sketch-photo matching, we use a few sketches collectively as faithful visual representatives (support) of novel classes for incremental learning. We show that sketches are better suited as class support in comparison to text, thanks to its inherent fine-grained nature, validated by findings in contemporary sketch literature [52, 11, 6].

Nonetheless, using sketches as class support in the FSCIL setting is non-trivial. *Sketch*, despite being visually representative, is just a coarse contour-like depiction of the visual world, that sit in an entirely different domain from photo [27]. Thus, off-the-shelf models naively pre-trained on photos commonly fail to generalise well on sketches [8]. Moreover, due to its highly abstract nature, the same object may be sketched in various ways under unique user-styles [51, 46], and with varied levels of detail [45]. We are also distinctly different to the parallel problem of SBIR – SBIR typically get exposed to *paired sketch-photo* data at training to learn a cross-modal embedding; we on the other hand need to work with *sketches only* at training (i.e., no photo information whatsoever), yet still aim to generate classification layer weights to classify *photos* from novel classes.

Three key design considerations for this cross-domain sketch-based FSCIL are: (i) how to make the model work cross-modal, (ii) how to preserve old class information, and (iii) how to leverage information from old classes to learn new ones. For the first issue, we design a gradient consensus based strategy that updates the model towards mutual agreement in the gradient space between sketch and photo domain, thus achieving a domain invariant feature extractor. For the second, we model an additional knowledge distillation loss to retain the acquired knowledge from old classes while incrementing the classifier to novel classes. Lastly, we devise a graph neural network to generate more discriminative decision boundaries for the incremented classifier via message passing between old and novel classes.

To summarise, our contributions are: (a) We extend incremental learning research even further towards practicality and human-likeness. (b) We achieve that by introducing sketches as class support for FSCIL, allowing the system to learn from modalities other than just photos and addressing issues around ethics and privacy while allowing user creativity. (c) We introduce the first cross-modal framework to tackle this novel DIY-FSCIL problem.

2. Related Work

Sketch Based Image Retrieval (SBIR): SBIR aims at retrieving paired photo given a query sketch, either at a *category-level* [11, 67, 12, 41, 65] or at a finer-grained *instance level* (FG-SBIR) [46, 10, 5, 6, 2]. For learning the joint embedding space, category-level SBIR typically employs either CNN [11, 12], RNN [65], or Transformer [41] based Siamese networks, accompanied by a triplet-ranking objective [68]. Contemporary research on this category is also directed towards zero-shot SBIR [12, 67, 44] and binary hash-code embedding [30, 49]. On the other hand, in FG-SBIR category, the seminal work by Yu *et al*. [68] first introduced deep triplet-ranking based Siamese networks for joint embedding space learning, which was further reinforced by attention [53], cross-domain translation [35], reinforcement learning based on-the-fly retrieval [6], semi-supervised retrieval [2], style-agnostic retrieval [46], etc.

Sketch for Vision Tasks: Hand-drawn sketches, by nature, are enriched with various human visual system-like understanding abilities and are quite close to the cognitive-subconscious of human intelligence [19]. Consequently, it has facilitated various visual understanding tasks in the past. Apart from the widely studied SBIR [11, 67, 12, 41, 46, 34, 53, 6, 2], sketch has also been employed in a variety of vision understanding tasks, including segmentation [22], video synthesis [28], representation learning [60, 3], object localisation [55], image-inpainting [62], 3D shape retrieval [31], 3D shape modelling [71], among others [64]. Some artistic application of sketch includes image editing [66], animation auto-completion [63], etc. Sketches have lately been used to create Pictionary-style competitive drawing games [4]. These establish the fact that hand-drawn sketches have enough representative ability to characterise a visual photo efficiently. Set upon this fact, in this paper, we aim to explore how *sketch* can act as a potential substitute to the conventional photos in class incremental learning.

Incremental Learning: Incremental Learning (IL) [38, 25] is a machine learning paradigm where a model adapts itself to learn new tasks sequentially while retaining the previously learnt knowledge. Although deep networks have demonstrated incredible achievements in a variety of tasks [48, 50], sequentially learning different tasks remains a key challenge. Consequently, IL continues to receive considerable research attention [21, 39, 7, 23, 1]. Majority of the present research either use memory-based [21, 39], distillation-based [9, 13], or regularisation-based [23] approaches to tackle the IL task. Based on the task at hand, IL can be categorised into (a) Incremental domain learning [42], which aims at performing incremental domain adaptation. (b) Incremental task learning [1], where each task consists of separate classification layers, and a task descriptor selects the appropriate layer during the testing phase. (c) Class incremental learning (CIL), the most challenging IL task that operates in a single-head setup with no available task descriptors. In CIL, the model needs to learn a unified classifier to fit all the new unseen classes incrementally. Distillation [29] and memory-based [21] methods are more effective than regularisation-based ones [23] in the CIL setting. This paper is mainly concerned with CIL setup, which is the most challenging task among its variants.

Few-Shot Class-Incremental Learning (FSCIL): Few shot learning (FSL) aims at adapting a trained model to learn patterns from novel classes (unseen during training) using only a *few labelled* samples [61]. Recently, it has experienced rapid proliferation [40, 50, 58] in the research community. There are three major swim lanes of the FSL problem: (a) recurrent-based [40, 48] (b) optimisation-based [43, 59], and (c) metric-based frameworks [17, 24]. Our work falls under metric-based methods in which similarity is drawn between the query sample and the novel support classes. Conventional CIL presumes that the incrementally provided novel classes have access to a substantial amount of labelled data. Although in the FSCIL paradigm [54], the initial dataset contains sufficient training data (base classes), the subsequently provided novel classes contain only a few labelled samples. Very few methods are present to tackle the FSCIL problem like, pseudo incremental learning [70], knowledge distillation [9, 13], neural-gas network [54]. While existing works intend to build a model to incrementally learn novel classes, we aim at building a model for a much harder and practically applicable sketch-based FSCIL setting that addresses user's privacy concerns.

Minimising Domain Discrepancy: Minimising sketch-photo domain discrepancy [12] is the key in our problem setup. In this context, the two most relevant branch of literature involves Domain Adaptation (DA) [15] and Domain Generalisation (DG) [27, 26]. While DA intends to adapt a model trained on a source domain to perform well on a new target domain using only unlabelled images, the aim of DG is to generalise a model from a set of *seen* domain samples to *unseen* domain samples without accessing the unseen domain instances. Our objective is more aligned with DG as we do not update the model parameters during inference. In this work, we take inspiration from the recent developments [69, 32] in DG to learn a domain-agnostic network, minimising the domain gap between sketch and photo.

3. Sketch for Incremental Learning

3.1. Problem Definition

Dataset: In few-shot class-incremental learning, we are given with K_b *base* classes and K_n *novel* classes respectively. From the set of base classes, we have *sufficient* access to labelled samples from *photo* $\mathcal{D}^P_{base} = \{(p_i, y^p_i)\}^{N^s_b}_{i=1}$ and *sketch* $\mathcal{D}^S_{base} = \{(s_i, y^s_i)\}^{N^s_b}_{i=1}$ domains, where $y_i \in \mathcal{C}_{base} = \{C^b_1, C^b_2, \cdots, C^b_{K_b}\}$. On the other side, for novel classes, we have *minimal* access to labelled samples from *only sketch* domain $\mathcal{D}^S_{novel} = \{(s_j, y_j)\}^{N^s_n}_{j=1}$ where number of samples for each novel category is limited, and $y_j \in \mathcal{C}_{novel} = \{C^n_1, C^n_2, \cdots, C^n_{K_n}\}$. Here, base and novel classes are completely disjoint, so that $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \Phi$.

Model: We have a neural network classifier, comprising of a feature extractor $\mathcal{F}_\theta$ followed by linear classifier $\mathcal{R}_w$, such that $y = \mathcal{R}_w(\mathcal{F}_\theta(x))$. $\mathcal{F}_\theta$ is employed using a convolutional neural network followed by global-average pooling, and given an input image $x \in \mathbb{R}^{h\times w\times 3}$, we get a feature representation as $f_d = \mathcal{F}_\theta(x) \in \mathbb{R}^d$. Following [17], for better generalisation $\mathcal{R}_w$ is devised as a *cosine similarity* function (unlike dot product based typical linear classifier), consisting a learnable W matrix whose size is of $\mathbb{R}^{|C|\times d}$, where $|C|$ is the number of classes. Thus, $\mathcal{R}_w : \mathbb{R}^d \rightarrow \mathbb{R}^{|C|}$ outputs a probability distribution over classes as $p(\bar{y}) = \texttt{softmax}(\hat{W} \cdot \frac{f_d}{\|f_d\|_2})$. $\hat{W}$ is obtained by l_2 normalising every d dimensional row-vector $w_k \in W$ that depicts weight-vector for k^{th} class, *i.e.* $\hat{w}_k = \frac{w_k}{\|w_k\|_2}$.

Learning Objective: The neural network classifier $\{\mathcal{F}_\theta, \mathcal{R}_w\}$ is trained from the abundant labelled samples of K_b base classes, and let the initial base classifier be $\mathcal{R}^{base}_w : \mathbb{R}^d \rightarrow \mathbb{R}^{K_b}$ whose weight matrix is $W_{base} \in \mathbb{R}^{K_b\times d}$. During inference under FSCIL [54], we do *not have* any access to labelled data of base classes, and given only k (small number) *sketch* samples for each of $\mathcal{C}_n$ novel categories, we intend to update the classifier $\mathcal{R}^{base}_w$ to $\mathcal{R}^{new}_w$ which can recognise *photos* from both $\mathcal{C}_{base} \cup \mathcal{C}_{novel}$ classes. To do so, we need to compute a new weight matrix $W_{new} \in \mathbb{R}^{(K_b+K_n)\times d}$ with respect to $\mathcal{R}^{new}_w : \mathbb{R}^d \rightarrow \mathbb{R}^{(K_b+K_n)}$ that can perform $(K_b + K_n)$-way class classification.

Therefore, our objective is to figure out a new W_{new} matrix for classifier $\mathcal{R}^{new}_w$ using the previous base classes' knowledge W_{base} and a few hand-drawn *sketch exemplars* from novel classes such that (i) the knowledge of base classes is not forgotten (preserved), as well as (ii) it quickly adapts to novel classes using few samples, (iii) thus, enabling it to perform well on real photos minimising the domain gap [26] with sketch samples from novel classes as support. Overall, our framework consists of *three* modules (i) a backbone *feature extractor* $\mathcal{F}_\theta$, (ii) a *classifier* $\mathcal{R}_w$ (iii) a *weight generator* G_ψ that will take previous base classifier weights W_{base} and sketch exemplars (support set) from novel classes as input, to generate a new weight matrix W_{new} for updated classifier $\mathcal{R}^{base}_w \rightarrow \mathcal{R}^{new}_w$ in order to classify real photos from *both* base and novel classes.

3.2. Cross Modal Pre-Training for Base Classes

Our framework follows a *two*-stage training. In the first stage, we train the model for base classes using standard cross-entropy loss, while in the second stage, we learn the *weight generator* via few-shot pseudo-incremental learning. Once trained, we freeze the weights of $\mathcal{F}_\theta$ in the next stage to (i) avoid over-fitting during the few-shot update and (ii) to alleviate catastrophic forgetting [23] of the base classes.

Unlike existing few-shot incremental learning, we need to handle the domain gap [27] between photos and sketches, so that the knowledge of incremental classes acquired

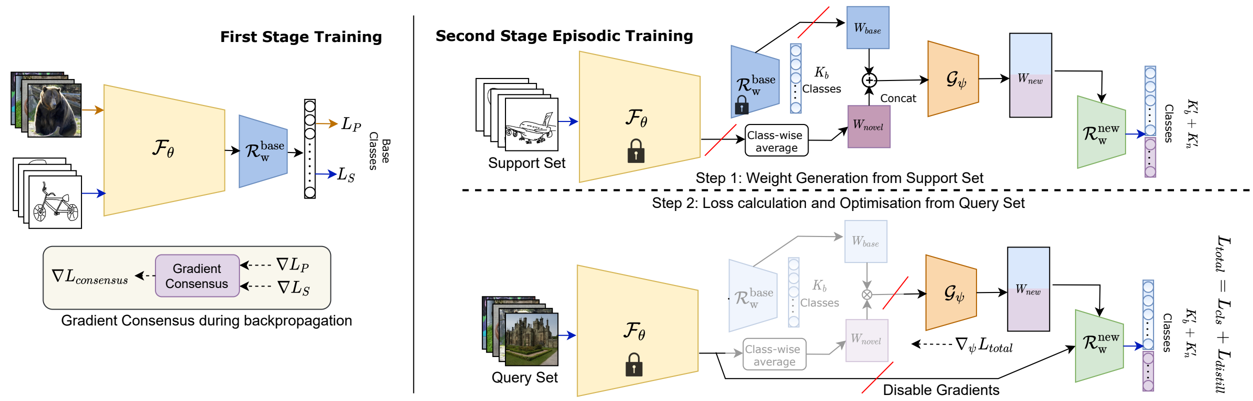

Figure 2. (a) Primarily, we aim to learn a domain-agnostic backbone feature extractor ($\mathcal{F}_\theta$) through gradient consensus. (b) In the second stage, we learn a weight generator (G_ψ) through episodic pseudo incremental learning involving two steps. Firstly, to obtain an updated [base+novel] classifier, a sketch support set is utilised to produce weight vectors for novel classes as well as to refine weight vectors for base classes. Secondly, for loss computation, the resulting weight vectors are evaluated against real photos from both [base+novel] classes.

through sketch exemplars can classify novel class images in real photo domain. As we have sufficient access to labelled training data from both photo and sketch domains for base classes, a very straightforward way to handle the domain gap is to train by combining labelled photos and sketches (spatially extended) with equal probability in every mini-batch – so that the model generalises equally well on both photos and sketches. Given an input x, let the model's output be $\bar{y} = \mathcal{R}_w^{base}(\mathcal{F}_\theta(x))$ where labelled data (x, y) come from either photo $(p, y^p) \sim \mathcal{D}_{base}^P$ or sketch $(s, y^s) \sim \mathcal{D}_{base}^S$ domain with $x \in \{p, s\}$ and y (y^s or y^p) being corresponding one-hot encoded class label. The cross-entropy loss $\mathcal{H}(\cdot, \cdot)$ can be calculated as $\mathcal{L} = \mathcal{H}(\bar{y}, y) = \sum_{i=1}^{K_b} y_i \log p(\bar{y}_i)$. Against a batch having b photos and b sketches, we can calculate the individual loss across photos and sketches as $\mathcal{L}_P$ and $\mathcal{L}_S$, respectively. Thereafter, we update the model by taking gradient $\nabla \mathcal{L}_{total}$ over total loss which is given as follows:

$$\mathcal{L}_{total} = \frac{1}{b} \sum_{(p,y)\sim\mathcal{D}_{base}^P} \mathcal{L}_P(p, y) + \frac{1}{b} \sum_{(s,y)\sim\mathcal{D}_{base}^S} \mathcal{L}_S(s, y) \tag{1}$$

However, naively training with two significantly different domains (photo vs sketch) gives rise to *conflicting gradients* within each batch, as information specific to the one domain might be irrelevant to the other, thereby suppressing the generalisation capability of the model. In other words, the information carried by $\nabla \mathcal{L}_P$ and $\nabla \mathcal{L}_S$ might not mutually agree, and adding them naively would lead to inhibiting [69] the training signal overall.

Gradient Consensus: Inspired from multi-task learning [69] and domain generalisation [32] literature, we aim to update the model in the direction where there is an agreement in the gradient space between two domains in order to learn a domain invariant representation. In particular, gradient vectors having the same sign will be retained, while those having conflicting signs will be set to *zero*, as shown in Eq. 2. Here, the $\text{sig}(\cdot)$ is a sign operator, and ∇L_P^n and ∇L_S^n denote the n-th component of the gradient associated to photo and sketch domain respectively. The gradient consensus function $\delta(\cdot, \cdot)$ checks element-wise if the signs of the gradient components match, and it returns 1 if all components have the same sign for a given n; otherwise 0.

$$\delta(\nabla L_P^n, \nabla L_S^n) = \begin{cases} 1, & \text{sig}(\nabla L_P^n) = \text{sig}(\nabla L_S^n) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\nabla L_{consensus}^n = \begin{cases} \nabla L_P^n + \nabla L_S^n, & \text{if} \quad \delta^n = 1 \\ 0, & \text{if} \quad \delta^n = 0 \end{cases} \tag{3}$$

This gradient agreement strategy helps to reduce the harmful cross-domain gradient interference while updating the model parameters using $\nabla L_{consensus}^n$. Thus, enabling us to adjust the model parameters in a direction that helps to improve generalisation across both sketch and photo.

3.3. Few-Shot Classifier Weight Generation

Overview: In order to classify photo from novel classes, we need to design a mechanism that can generate *additional* weight vectors for the novel classes. As we assume that only a few supporting hand-drawn sketch exemplars will be provided corresponding to every novel class, we design the weight generator G_ψ under a few-shot paradigm [61]. G_ψ produces weight vectors for novel classes and also regenerates (refines) weight vectors for base classes in order to get a better overall decision boundary in the presence of novel classes. Here, the two major objectives are (i) learn the knowledge of novel classes from *fewer sketch exemplars*, while *classifying photos* of novel classes through cross-modal generalisation (ii) *not to degrade* the performance of base classes while learning the novel ones.

We employ sketch exemplars as a *support set* to generate the *new* weight matrix following the episodic training [27] of few-shot learning. To determine the loss for updating the weight generating module, the quality of the generated weight matrix is assessed against a *query set* of photo samples. In particular, there are *two* steps [50] while training the weight generation module. (i) *Weight generation* using support set: sketch exemplars as support set are used together with W_{base} to generate the new weight matrix W_{new} (comprising both base and novel classes) (ii) *Loss calculation* on query set: W_{new} is used to classify query set *photos* in order to calculate loss, which is then utilised to optimise the weight generation module using gradient descent.

Weight Generation: G_{ψ} takes two things as input (i) W_{base} from $\mathcal{R}_w^{base}$ representing the knowledge of previous base classes (ii) class-wise representative features of novel classes from sketch exemplars. We assume to have access to k sketch samples for each of the K_n novel classes – the support set. A straightforward way to get class-wise representative vectors is to average feature representations of sketches for each individual classes. In particular, for j^{th} novel class, the representative vector can be calculated as:

$$w_j^{novel} = \frac{1}{k}\sum_{i=1}^{k}\mathcal{F}_{\theta}(s_i) \tag{4}$$

Thereafter, by applying l_2 norm on each w_j^{novel}, we can naively form the weight vectors of novel classes as $W_{novel} = \{w_1^{novel}, w_2^{novel}, \cdots, w_{K_n}^{novel}\} \in \mathbb{R}^{K_n \times d}$. The easiest way for incremental learning would be to use naive concatenation to get new weight matrix as $[W_{base}; W_{novel}] \in \mathbb{R}^{(K_b+K_n)\times d}$. However, it has *two* major limitations (i) W_{novel} remains unaware about the knowledge of bases classes (ii) W_{base} which was discriminative across the base classes might lose its representation-potential when we add additional weight vectors of novel classes without modelling a mutual agreement strategy for learning discriminative decision boundaries across all $K_b + K_n$ classes. Thus, to attain an optimal decision boundary for all classes under incremental setup, an *information passing* mechanism is critical for W_{new} generation.

Message Passing: For *information-propagation* among weight vectors of $K_b + K_n$ classes, we use Graph Attention Network (GAT) [57]. GAT is a good choice for information-propagation owing to its permutation-invariance to sequence of weight vectors as the novel classes may appear in any order. As the weights are shared across different nodes, it can also handle incoming variable number of novel classes effortlessly. The input to GAT is given as $W_I = \{w_1^{base}, \cdots, w_{K_b}^{base}, w_1^{novel}, \cdots, w_{K_n}^{novel}\}$ having $K_{total} = K_b + K_n$ weight vectors, where each $w_i \in \mathbf{R}^d$ denotes an input to a specific node to GAT. First it computes relation co-efficient between every pair of node by inner product operation as $e_{i,j} = \langle V_a w_i, V_b w_j\rangle$, with two learnable linear embedding weights V_a and V_b. $e_{i,j}$ is normalised by softmax function to get the attention weights with respect to node i as: $a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{K_{total}}\exp(e_{ik})}$. The update rule for i^{th} node gathering information from all other nodes becomes

$$w_i^{update} = w_i + \big(\sum_{j=1}^{K_{total}} a_{i,j}V_c w_i\big) \tag{5}$$

where, V_c is a learnable linear transformation. We repeatedly update the weight vectors at every node in the graph, and finally we obtain the generated weight vectors for both base and novel classes as W_{new}. In brief, $W_{new} = G_{\psi}(W_I) : \mathbb{R}^{(K_b+K_n)\times d} \rightarrow \mathbb{R}^{(K_b+K_n)\times d}$, where $W_I = [W_{base}; W_{novel}] \in \mathbb{R}^{(K_b+K_n)\times d}$, thus we generate the weight vectors for both base and novel classes during incremental learning.

Episodic Pseudo Incremental Training: Keeping the feature extractor $\mathcal{F}_{\theta}$ fixed, we train the few-shot weight generator G_{ψ} taking inspiration from few-shot learning literature [50, 40, 48]. As the training dataset is limited, we episodically construct pseudo incremental task based *only* on the base classes to mimic the real testing scenario.

In particular, following the first stage of training, we get classifier weight matrix of base classes as $W_{base} \in \mathbb{R}^{K_b \times d}$. In order to create each episode, we *synthetically drop* K_n' weight vectors from W_{base}, and we treat those corresponding classes as pseudo novel classes whose weights now need to be generated. That means, at a particular episode, the pseudo base class matrix becomes $W'_{base} \in \mathbb{R}^{K_b' \times d}$ where $K_b' = K_b - K_n'$. Thereafter, corresponding to those *dropped* base classes which now become pseudo novel classes, we use k sketch samples for each of the pseudo novel classes as the *support set* to first generate representative class-wise weight vectors $W'_{novel} \in \mathbb{R}^{K_n' \times d}$, which is again fed to GAT together with W'_{base} for relationship modelling to generate pseudo W'_{new}. In every episode, while support set ($\mathcal{S}$) is used to generate the classifier weights, another query set ($\mathcal{Q}$) involving real photos from both pseudo base and novel classes are fed through pre-trained backbone followed by classifier with newly generated weight matrix W'_{new} to compute loss for optimisation. Please refer to Fig. 2.

In contrast to earlier FSCIL works [54, 13], our episodic training is cross-modal in nature, where the support and query set consist of sketch and photo respectively. As training is done over base classes with pseudo-novel classes, we found mixing both sketch and photo in the support set with *gradient consensus* generalises better on real photos. However, sketch acts as the only exemplars during real inference.

Loss Functions: Contrary to fully supervised classification from abundant training data, few-shot learning [50] is more challenging as only a few samples are available for the new weight matrix generation. Given this rationale, we aim to design the pseudo incremental learning by dropping weights vectors from W_{base}, which is learned from base classes through standard supervised classification. We aim to see if the fully supervised knowledge learned in W_{base} could provide training signal [20] to learn the G_{ψ}.To do so, we additionally define a *distillation loss* along with standard *classification loss* calculated over the query set, which acts a consistency regularisation. This ensures that weight vectors predicted by the weight generator remain close to what has been learned through supervised classification from first stage. In particular, following few-shot weight generation we get an incrementally learned classifier $\mathcal{R}_w^{new}$ with generated weight matrix W_{new}. On the other side we already have $\mathcal{R}_w^{base}$ learned from first stage pre-training. Given a photo p from query set ($\mathcal{Q}$), for distillation loss we treat the *soft* prediction using $\mathcal{R}_w^{base}$ as a ground-truth to calculate the distillation loss. Thus, the total loss becomes $\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{distil}$ which is used to train G_ψ. If $\mathcal{H}(\cdot, \cdot)$ be cross-entropy loss, $\mathcal{L}_{cls}$ and $\mathcal{L}_{distil}$ are defined as:

$$\mathcal{L}_{cls} = \frac{1}{|\mathcal{Q}|} \sum_{(p,y)\sim\mathcal{Q}} \mathcal{H}(\mathcal{R}_w^{new}(\mathcal{F}_\theta(p)), y) \tag{6}$$

$$\mathcal{L}_{distil} = \frac{1}{|\mathcal{Q}|} \sum_{(p,y)\sim\mathcal{Q}} \mathcal{H}(\mathcal{R}_w^{new}(\mathcal{F}_\theta(p)), \mathcal{R}_w^{base}(\mathcal{F}_\theta(p))) \tag{7}$$

4. Experiments

Datasets: We evaluate our DIY-FSCIL framework on the popular Sketchy dataset [47] which is a large collection of photo-sketch pairs. As paired photo-sketch is *not essential* for our framework, we use the extended version of Sketchy with $60,502$ additional photos that Liu *et al.* [30] later introduced for category-level SBIR. In particular, Sketchy-extended comprises 125 categories with $75,471$ sketches and $73,002$ images in total. Existing zero-shot SBIR [12, 14] works split the dataset into $104/21$ disjoint classes for training/testing(unseen). We keep the same 21 classes for testing (novel classes), while for hyperparameter tuning, out of 104 classes, we consider 64 for training and the rest 40 classes for validation. In summary, we call them T_{train} (64 classes), T_{val} (40 classes), and T_{test} (21 classes) respectively. The train set (T_{train}) is often referred to as *base dataset* and is further split into three subsets ($T_{train}^{train} : T_{train}^{val} : T_{train}^{test}) = (60\% : 20\% : 20\%)$. The subset T_{train}^{test} is used to evaluate the overall performance on the base classes during incremental setup. The steps outlined above are followed for both sketches and photos. For every model evaluations, we follow the same settings, including the categories' division and incremental training samples.

Implementation Details: We have implemented the DIY-FSCIL framework using PyTorch [36] and conducted the experiments using one 11-GB NVIDIA RTX 2080-Ti GPU. We employ the standard ResNet18 model as the backbone feature extractor ($\mathcal{F}_\theta$). The features of the input image are derived from the final pooling layer of the $\mathcal{F}_\theta$ with a dimension of $d = 512$. We use a one-layer GAT to design our weight generator G_ψ. In the initial stage, the feature extractor ($\mathcal{F}_\theta$) is trained on the training set T_{train}^{train}. We train the $\mathcal{F}_\theta$ for 100 epochs, and during the second stage $\mathcal{F}_\theta$ is freezed and the weight generation module involving GAT is trained for 60 epochs. We use SGD optimiser with learning rate 0.01 and batch size of 8 for all experiments. In order to reduce the error caused by the random sampling of the incremental classes and its samples, we report the average results obtained by *five* different seeds.

4.1. Evaluation Protocol

Following the incremental step $\mathcal{R}_w^{base} \rightarrow \mathcal{R}_w^{new}$, we evaluate the performance of staged operations $\mathcal{F}_\theta \circ \mathcal{R}_w^{new}$ under three circumstances – (a) upon only novel classes, (b) upon only base classes, and (c) upon both base plus novel classes. While for only novel classes the class label space consists of $y_j \in \mathcal{C}_{novel} = \{C_1^n \cup C_2^n \cdots C_{K_n}^n\}$, the same for only base classes becomes $y_j \in \mathcal{C}_{base} = \{C_1^b \cup C_2^b \cdots C_{K_b}^b\}$. Furthermore, for evaluation under base plus novel classes, the label space spans across $y_j \in \mathcal{C}_{both} = \{\mathcal{C}_{base} \cup \mathcal{C}_{novel}\}$. These three evaluating situations answer – (a) how well the model adapts to novel classes from few (1 or 5) sketch examples, (b) how well the model is able to preserve the accuracy (mitigating catastrophic forgetting) of the base classes for which the training data is inaccessible during the incremental step, (c) how well the model performs overall for both base and novel classes. Following the two-stage training using T_{train}^{train}, i.e., pre-training on base-classes followed by learning few-shot weight generator, we obtain $\mathcal{F}_\theta$ and G_ψ, which are used for inference under incremental setup.

Evaluation of novel classes (Acc@novel): Test set (T_{test}) is used to create few shot tasks similar to episodic training. These few shot tasks are formed by sampling $K_{novel} = 5$ categories. Then, we sample one (1-shot) or five (5-shot) exemplars per category (sketches) and 15 query samples per category (photos). Here, the query samples will be from the same novel categories but we make sure that they do not overlap with the exemplars under a particular episode. G_ψ uses exemplar embeddings obtained via $\mathcal{F}_\theta$, along with base weights, to generate incremented classifier's weight W_{new}, which is then evaluated on the query set. Apart from helping to understand the model's capability to learn novel classes in a few-shot setting excluding the base classes, this metric also helps assessing the *model's generalisation capability* on *cross-domain* data. Following the existing FSCIL literature [17], we create 600 few-shot tasks and report the average results from them.

Evaluation of base classes (Acc@base): To verify the potential of mitigating the catastrophic forgetting issue, we evaluate the recognition performance on base categories using the T_{train}^{test} subset on the incremented classifier $\mathcal{R}_w^{new}$. Here, we create few shot tasks by randomly sampling $K_b = 5$ categories from the base classes without replacement, followed by evaluating with 15 query photos for each category.

Evaluation of all the classes (Acc@both): Here the label space spans across all the classes ($y_j \in \mathcal{C}_{both}$). In each episode, we sample from all K_b base and K_n novel classes. Then, we sample one (1-shot) or five (5-shot) exemplars per novel category (sketches), 15 query samples for each of the base and novel categories (photos) to evaluate the perfor-

Table 1. Average classification accuracy of DIY-FSCIL framework using our self-designed baselines and adopted SOTA FSCIL [17] (not specifically designed for cross-modal). For every experiment, we create 600 episodes each with 5 random classes from both novel and base categories separately. Each episode contains a total 15×5 (15 samples from each of the 5 classes) and 15×5 query photos from from both novel and base categories respectively. $B5^*$ is an upper bound.

Methods		5-Shot Learning			1-Shot Learning		
		Acc@both	Acc@base	Acc@novel	Acc@both	Acc@base	Acc@novel
Baselines	$B1$	36.29 %	73.94%	38.92%	31.52%	73.98%	34.68%
	$B2$	25.86%	32.85%	70.58%	28.81%	40.91%	50.24%
	$B3$	58.92%	73.81%	72.34%	53.35%	73.75%	59.93%
	$B4$	54.5%	71.68%	71.81%	51.41%	71.68%	51.44%
	$B5^*$	71.52%	75.72%	85.46%	63.47%	75.83%	73.90%
SOTA FSCIL	[17]	50.45%	74.35%	65.81%	44.71%	73.98%	64.21%
	[50]	45.25%	74.10%	63.46%	41.97%	74.60%	61.85%
	[54]	51.54%	73.21%	66.82%	45.81%	73.58%	63.95%
Ours	DIY-FSCIL	60.54%	74.38%	75.84%	54.97 %	74.06%	64.10%

mance. This metric helps to determine how the base classes' knowledge affects novel classes and vice-versa.

4.2. Competitors

As there exists no prior work dealing with sketch-based FSCIL, we implement the following set of baselines and their adaptions in order to assess the contribution of our proposed framework. • **B1:** We use a combination of old-base and new-novel classes to *retrain* the complete model. Besides requiring a lot of computational power, this suffers from a severe class imbalance problem between sufficiently available base classes and few exemplars from novel classes. Nevertheless, this can not be realised in a real scenario. • **B2:** We only fine-tune the model using the novel classes. It acts as a naive baseline, and is limited due to the issue of catastrophic forgetting. • **B3:** We freeze backbone feature extractor F_θ, and use the class-wise average feature of sketch exemplars as the representative weight-vectors of novel classes along with the pre-trained base-classifier. In other words, we remove the GAT module from our proposed framework. • **B4:** We further examine the performance of our framework by training both the $\mathcal{F}_\theta$ along with the $\mathcal{G}_\psi$. This is used to analyse the importance of freezing the feature extractor $\mathcal{F}_\theta$. • **B5:** During testing, we utilise real images as the support set. As images are more detailed than sketches, this model serves as our upper boundary. However, it fails to address our main concern of violating the data privacy norm. For a fair comparison, we utilise the same settings for all the models as our framework. Though existing FSCIL methods [17, 54, 50] are not specifically designed to deal with cross-modal sketch exemplars, we naively adopt those under our sketch-based FSCIL setup.

4.3. Performance Analysis

In Table 1, we report the comparative results using the standard *one*-shot and *five*-shot sketch-based FSCIL setting on Sketchy dataset. We make the following observations: (i) Despite using abundant memory and computational resources **B1** performs poorly on the novel classes, due to the absence of any mechanism to handle few shot classes (i.e., severe class imbalance). This suggests that few shot paradigm is essential to perform reasonably well on novel classes. (ii) **B2** adapts fine-tuning on the novel classes without heavy computational overhead. However, doing so declines the model's performance on the base classes due to catastrophic interference. (iii) While **B3** outperforms baselines **B1** and **B2**, it fails to model mutual agreement between base and novel classes for learning discriminative decision boundaries under incremental setup, revealing the importance of our weight refining strategy through GAT module. (iv) Low performance of **B4** signifies the necessity of freezing the weights of $\mathcal{F}_\theta$ during the second stage of training in order to reduce the catastrophic forgetting problem, and also to generalise notably better on *unseen* categories. (v) **B5** (upper bound) achieves the best numbers, as the support set comes directly from photos, and this is unlike ours where we have a critical challenge due to the domain gap between sketch exemplars and query photos. (vi) Moreover, the performance of SOTA FSCIL methods is limited by a margin of 9.09% under DIY-FSCIL setup.

To summarise, our framework helps in solving the challenging DIY-FSCIL problem by both alleviating the catastrophic forgetting of the old classes and enhancing the learning of the new classes under a cross-modal sketch-based few shot setting. Moreover, the proposed framework effectively enables the users to build their own novel classes with the support of their imaginative drawings.

4.4. Further Analysis and Insights

Ablation Study: We further dive deeper to figure out the contribution of individual design components in Table 2. (i) *GAT:* To access the importance of weight refinement, we remove the GAT module and adapt the framework accordingly. Consequently Acc@novel significantly drops to 62.34% with a decrease of 3.5% for 5-shot case, and is more pronounced for 1-shot context, where we perceive larger a drop of 4.17%. This observation further strengthens our initial assumption that GAT models an effective mutual agreement strategy for learning discriminative decision boundaries across all the $K_b + K_n$ classes. (ii) *Gradient Consensus (GC):* The use of GC improves the model's perfor-

Table 2. Ablative study: GAT (Graph Attention Network), GC (Gradient Consensus), KD (Knowledge Distillation Loss), CMT (Cross-Modal Training)

GAT	GC	KD	CMT		Metrics		
					Acc@both	Acc@base	Acc@novel
✔	✔	✔	✔	5−shot	60.54%	74.38%	75.84%
				1−shot	54.97%	74.06%	64.10%
✗	✔	✔	✔	5−shot	58.92%	73.81%	72.34%
				1−shot	53.35%	73.75%	59.93%
✗	✗	✔	✔	5−shot	58.47%	73.96%	71.67%
				1−shot	53.22%	73.67%	59.46%
✗	✗	✗	✔	5−shot	57.47%	70.96%	69.67%
				1−shot	51.22%	71.67%	57.46%
✗	✗	✗	✗	5−shot	35.19%	62.98%	40.52%
				1−shot	27.67%	61.72%	32.83%

Table 3. Performance with varying n-way/k-shot evaluation

		Metrics		
		Acc@both	Acc@base	Acc@novel
5− way	1−shot	54.97%	74.06%	64.10%
	5−shot	60.54%	74.38%	75.84%
	10−shot	61.61%	74.14%	76.95%
	15−shot	62.08%	73.95%	77.48%
	20−shot	62.35%	74.83%	78.35%
10− way	1−shot	43.62%	73.24%	47.31%
	5−shot	51.82%	73.37%	59.97%
	10−shot	53.75%	73.54%	61.21%
	15−shot	55.46%	73.38%	62.74%
	20−shot	57.58%	73.23%	64.37%

mance substantially, and this is particularly apparent in the initial stages. During the initial stage training, GC improves the model accuracy by 2.72% via effective handling of the harmful cross-domain gradient interference while updating the model parameters. (iii) *Knowledge Distillation (KD):* Knowledge distillation-based regularisation seeks to provide stability and enforces weight generation module learning. Getting rid of it reduces the Acc@both by a significant $3.75\%(3.07\%)$ for $1(5)$-shot setting, thus illustrating its need. (iv) *Cross Modal Training (CMT):* While we use only sketch exemplars as the support set during real inference, during episodic training we mix both sketch and photos along with gradient consensus strategy to bridge the domain gap in weight generation process. Removing this cross-modal training drops the Acc@both by $27.3\%(25.35\%)$ for $1(5)$-shot setting. In summary, all of the components work in unison to produce the best overall performance.

Effect of the number of sketch-exemplars: Next, to investigate how the number of classes and samples affect the overall model performance, we evaluate the framework by varying the number of shots from $\{1, 5, 10, 15, 20\}$ and the number of ways only from $\{5, 10\}$. We depict the corresponding results in Table 3. We infer that larger way hurts the performance because of the ambiguity created by the new classes, while the model's performance increases when training with more number of samples. This portrays the potency of our proposed framework for other CIL variants.

Comparison with text as support-set: In order to compare our approach with text-based support set, we use the word embeddings from Word2Vec [33] and GloVe [37] to generate class representations. We delineated the results in Table 4. The fine-grained nature of sketches helped surpass text results by a wide margin, showing its efficacy as the class support and a possible substitute to photos.

Table 4. Comparative study between *sketch* vs *text* for support set

	One-shot learning		
	Acc@both	Acc@base	Acc@novel
Text (Word2Vec)	22.85%	73.98%	26.15%
Text (GloVe)	22.80%	74.04%	26.85%
Sketch (Ours)	54.97%	74.06%	64.10%

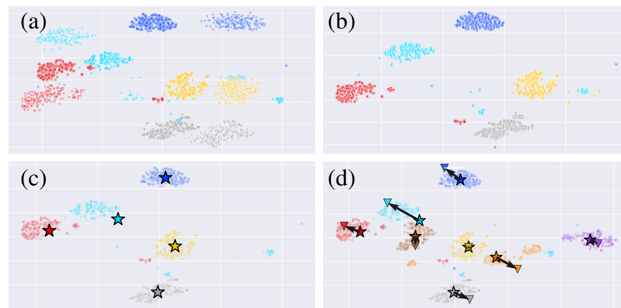

Figure 3. t-SNE Plots: (a) Photo (○) and Sketch (∇) on the common embedding space of $\mathcal{F}_\theta$ using naive baseline. (b) Photo (○) and Sketch (∇) on the shared embedding space of $\mathcal{F}_\theta$ obtained *using* our framework. This can be inferred that our method aligns the both modalities better by minimising the domain gap. (c) Base Classes (d) DIY-FSCIL. Here, deep-colour points are class prototypes, light-colour ones show the distribution of real data, star represents class representations during 1st stage, delta (bold) represents the refined vectors during incremental stage, and black arrow indicates the change in weights. Our DIY-FSCIL pushes the classifier weights away from the uncertain areas, resulting in better decision boundaries. Zoom in for better view.

Visualisation of GAT refined features: With t-SNE [56], we visualise class representation weight vectors and classifier weights in a low-dimension space. We exhibit the results for the two configurations – (i) with GAT, and (ii) without GAT. For this study, five classes are chosen randomly as the base classes, and five additional classes are added as incremental classes. As evident from Fig. 3, during incremental setup, the GAT module refines weights efficiently to push the classifier weights away from the uncertain areas, resulting in better decision boundary.

5. Conclusion

In this paper, we have introduced a novel framework for few shot class incremental learning without violating the data privacy and ethical norms. This method also empowers the users to construct novel categories just by providing a few imaginative sketches doodled by themselves. The proposed framework unifies Knowledge Distillation, Gradient Consensus, and Graph Attention Networks to handle this newly proposed DIY-FSCIL paradigm. The effectiveness of the framework is validated by various experiments on the Sketchy dataset. Our framework is also extendable to other IL methods beyond the CIL used in this study.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018. 2

[2] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. More photos are all you need: Semi-supervised learning for fine-grained sketch based image retrieval. In *CVPR*, 2021. 2

[3] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Vectorization and rasterization: Self-supervised learning for sketch and handwriting. In *CVPR*, 2021. 2

[4] Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Pixelor: A competitive sketching ai agent. so you think you can beat me? In *ACM TOG*, 2020. 2

[5] Ayan Kumar Bhunia, Subhadeep Koley, Abdullah Faiz Ur Rahman Khilji, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketching without worrying: Noise-tolerant sketch-based image retrieval. In *CVPR*, 2022. 2

[6] Ayan Kumar Bhunia, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *CVPR*, 2020. 2

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *ICLR*, 2018. 2

[8] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 2

[9] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *CVPR*, 2021. 2, 3

[10] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Viswanatha Reddy Gajjala, Aneeshan Sain, Tao Xiang, and Yi-Zhe Song. Partially does it: Towards scene-level fg-sbir with partial input. In *CVPR*, 2022. 2

[11] John Collomosse, Tu Bui, and Hailin Jin. Livesketch: Query perturbations for guided sketch-based visual search. In *CVPR*, 2019. 2

[12] Sounak Dey, Pau Riba, Anjan Dutta, Josep Llados, and Yi-Zhe Song. Doodle to search: Practical zero-shot sketch-based image retrieval. In *CVPR*, 2019. 2, 3, 6

[13] Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-shot class-incremental learning via relation knowledge distillation. In *AAAI*, 2021. 2, 3, 5

[14] Anjan Dutta and Zeynep Akata. Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval. In *CVPR*, 2019. 2, 6

[15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 3

[16] Songwei Ge, Vedanuj Goswami, C Lawrence Zitnick, and Devi Parikh. Creative sketch generation. In *ICLR*, 2021. 1

[17] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3, 6, 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 1

[19] Aaron Hertzmann. Why do line drawings work? a realism hypothesis. *Perception*, 2020. 2

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning Workshop*, 2014. 5

[21] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. In *NeurIPS Continual Learning Workshop*, 2018. 1, 2

[22] Conghui Hu, Da Li, Yongxin Yang, Timothy M Hospedales, and Yi-Zhe Song. Sketch-a-segmenter: Sketch-based photo segmenter generation. *IEEE TIP*, 2020. 2

[23] 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. *Proc. of the NAS*, 2017. 1, 2, 3

[24] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML*, 2015. 3

[25] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From n to n+ 1: Multiclass transfer incremental learning. In *CVPR*, 2013. 2

[26] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 3

[27] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *ICCV*, 2019. 2, 3, 4

[28] Xiaoyu Li, Bo Zhang, Jing Liao, and Pedro Sander. Deep sketch-guided cartoon video inbetweening. *IEEE TVCG*, 2021. 2

[29] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE TPAMI*, 2017. 1, 2

[30] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*, 2017. 2, 6

[31] Ling Luo, Yulia Gryaditskaya, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Towards 3d vr-sketch to 3d shape retrieval. In *3DV*, 2020. 2

[32] Lucas Mansilla, Rodrigo Echeveste, Diego H Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *ICCV*, 2021. 3, 4

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 8

[34] Kaiyue Pang, Ke Li, Yongxin Yang, Honggang Zhang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Generalising fine-grained sketch-based image retrieval. In *CVPR*, 2019. 2

[35] Kaiyue Pang, Yi-Zhe Song, Tony Xiang, and Timothy M Hospedales. Cross-domain generative learning for fine-grained sketch-based image retrieval. In *BMVC*, 2017. 2

[36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*, 2017. 6

[37] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 8

[38] Robi Polikar, Lalita Upda, Satish S Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE TSMC:C*, 2001. 2

[39] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 2

[40] Danilo Rezende, Ivo Danihelka, Karol Gregor, Daan Wierstra, et al. One-shot generalization in deep generative models. In *ICML*, 2016. 3, 5

[41] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *CVPR*, 2020. 2

[42] Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *IEEE TPAMI*, 2018. 2

[43] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 3

[44] Aneeshan Sain, Ayan Kumar Bhunia, Vaishnav Potlapalli, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketch3t: Test-time training for zero-shot sbir. In *CVPR*, 2022. 2

[45] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Cross-modal hierarchical modelling for fine-grained sketch based image retrieval. In *BMVC*, 2020. 2

[46] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Stylemeup: Towards style-agnostic sketch-based image retrieval. In *CVPR*, 2021. 2

[47] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM TOG*, 2016. 6

[48] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 2, 3, 5

[49] Yuming Shen, Li Liu, Fumin Shen, and Ling Shao. Zero-shot sketch-image hashing. In *CVPR*, 2018. 2

[50] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2, 3, 4, 5, 7

[51] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to sketch with shortcut cycle consistency. In *CVPR*, 2018. 2

[52] Jifei Song, Yi-Zhe Song, Tony Xiang, and Timothy M Hospedales. Fine-grained image retrieval: the text/sketch input dilemma. In *BMVC*, 2017. 2

[53] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *CVPR*, 2017. 2

[54] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, 2020. 1, 3, 5, 7

[55] Aditay Tripathi, Rajath R Dani, Anand Mishra, and Anirban Chakraborty. Sketch-guided object localization in natural images. In *ECCV*, 2020. 2

[56] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 8

[57] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. 2018. 5

[58] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NeurIPS*, 2016. 3

[59] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *NeurIPS*, 2019. 3

[60] Alexander Wang, Mengye Ren, and Richard Zemel. Sketchembednet: Learning novel concepts by imitating drawings. In *ICML*, 2021. 2

[61] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM CSUR*, 2020. 3, 4

[62] Minshan Xie, Menghan Xia, and Tien-Tsin Wong. Exploiting aliasing for manga restoration. In *CVPR*, 2021. 2

[63] Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. Autocomplete hand-drawn animations. *ACM TOG*, 2015. 2

[64] Peng Xu, Timothy Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep learning for free-hand sketch: A survey. *arXiv preprint arXiv:2001.02600*, 2020. 2

[65] Peng Xu, Yongye Huang, Tongtong Yuan, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, Timothy M. Hospedales, Zhanyu Ma, and Jun Guo. Sketchmate: Deep hashing for million-scale human sketch retrieval. In *CVPR*, 2018. 2

[66] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In *ECCV*, 2020. 2

[67] Sasi Kiran Yelamarthi, Shiva Krishna Reddy, Ashish Mishra, and Anurag Mittal. A zero-shot framework for sketch based image retrieval. In *ECCV*, 2018. 2

[68] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *CVPR*, 2016. 2

[69] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. 3, 4

[70] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *CVPR*, 2021. 3

[71] Song-Hai Zhang, Yuan-Chen Guo, and Qing-Wen Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *CVPR*, 2021. 2