Hyper-LifelongGAN: Scalable Lifelong Learning for Image Conditioned Generation

Mengyao Zhai Lei Chen Greg Mori
Simon Fraser University
{mzhai, chenleic}@sfu.ca mori@cs.sfu.ca

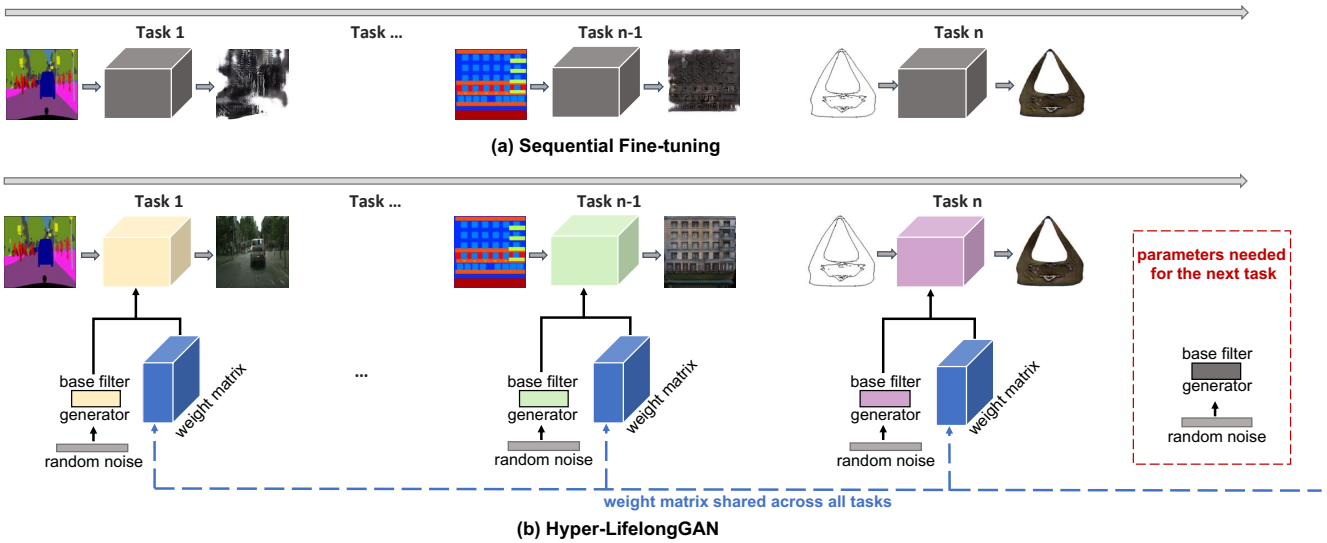

Figure 1: **Lifelong learning of image-conditioned generation.** Encountering a new task, traditional training methods forget how to perform previous tasks (Figure 1 (a)). Hyper-LifelongGAN is a scalable framework addressing catastrophic forgetting. It can adapt to the new task with few additional parameters, while preserving the knowledge of previous tasks(Figure 1 (b)).

Abstract

Deep neural networks are susceptible to catastrophic forgetting: when encountering a new task, they can only remember the new task and fail to preserve its ability to accomplish previously learned tasks. In this paper, we study the problem of lifelong learning for generative models and propose a novel and generic continual learning framework Hyper-LifelongGAN which is more scalable compared with state-of-the-art approaches. Given a sequence of tasks, the conventional convolutional filters are factorized into the dynamic base filters which are generated using task specific filter generators, and deterministic weight matrix which linearly combines the base filters and is shared across different tasks. Moreover, the shared weight matrix is multiplied by task specific coefficients to introduce more flexibility in combining task specific base filters differently for different tasks. Attributed to the novel architecture, the proposed method can preserve or even improve the generation quality at a low cost of parameters. We validate Hyper-LifelongGAN on diverse image-conditioned generation tasks, extensive ablation studies and comparisons with state-of-the-art models are carried out to show that the proposed approach can address catastrophic forgetting effectively.

1. Introduction

The continuous learning ability is one of the hallmarks of human intelligence. Humans are lifelong learners, we acquire and accumulate knowledge throughout our lives. The accumulation of knowledge in turn makes us more and more knowledgeable, and better and better at learning when encountering new problems. In contrast to human learning, modern deep neural networks are susceptible to catastrophic

forgetting [26]: when adapted to perform new tasks, they often fail to generalize and cannot maintain their ability to accomplish previously learned tasks (see Figure 1 (a)). Recent approaches [35, 41, 40] have been proposed for lifelong learning for generative models, and how generative models can continually learn a sequence of tasks was explored in these methods. Though progress has been made towards lifelong learning for generative models, it remains a challenging area.

The pioneer work addressing catastrophic forgetting in the generative setting is *memory replay* [35], namely generating data of previous tasks using a trained model and treating these generated data as part of the training examples in the new tasks. Although alleviating catastrophic forgetting by taking advantage of the generative setting, memory replay is limited to label-conditioned generation scenarios: when training data for only the current task is accessible, no conditional image can be accessed and as a result no images could be generated for replay. More generic continual learning frameworks [41, 40] have been proposed enabling lifelong learning of image-conditioned generation tasks. LifelongGAN [41] continually adapts a single trained model to later tasks, thus the whole model is shared across all tasks. However, due to the intrinsic differences among tasks, it is hard to adapt all parameters of a trained model to a new task. As a result, LifelongGAN is not able to preserve the generation quality of previous tasks while learning the new task well. This performance degradation makes it not scalable in general. PiggybackGAN [40] addresses the performance degradation problem by sacrificing memory storage. Though it is more parameter efficient compared with training separate models for each task, the unconstrained filters bring millions of additional parameters for each new task. This storage requirement limits its scalability. Therefore, a more scalable continual learning framework that can preserve the generation quality with no or little sacrifice of storage is valuable.

In this paper, we introduce a generic continual learning framework *Hyper-LifelongGAN* (see Figure 1 (b)) that is more scalable compared with state-of-the-art approaches. Hypernetwork [13] and knowledge distillation [15] are employed to address catastrophic forgetting for generative tasks. First, all the conventional convolutional and deconvolutional filters in the generator are factorized into a set of base filters and a weight matrix that linearly combines the base filters. And instead of learning deterministic base filters, we learn to generate dynamic base filters from random noises using hypernetworks. Given a sequence of tasks, different hypernetworks are trained to generate base filters for different tasks (referred to as *task specific filter generators*); while the *weight matrix* is deterministic, and shared across all the tasks. Moreover, the shared weight matrix is multiplied by task specific *coefficients* to introduce more flexibility in combining task specific filters differently for different tasks. The memory requirement is low since the base filters in each layer can be generated with just few thousand parameters, and the weight matrix is shared across all tasks. To keep the memory of previous tasks, knowledge is extracted from a previously trained model and distilled to the model trained for the new task, encouraging the new model to generate the same output as the previous model.

To summarize, our contributions are as follows. *First*, we propose a novel and generic continual learning framework *Hyper-LifelongGAN* that is more scalable. *Second*, we propose to factorize conventional convolutional filters into dynamic task specific base filters and deterministic task independent weight matrix. This design enables the proposed model to preserve or even improve the generation quality of a sequence of tasks at a low cost of parameters. *Third*, extensive ablation studies and comparisons with state-of-the-art models are carried out across diverse data domains, qualitative and quantitative results are provided to illustrate the capability of our framework to learn new generation tasks without the catastrophic forgetting of previous tasks.

2. Related Work

Lifelong Learning. For discriminative tasks e.g. classification, recent efforts [29, 8, 3, 4] have achieved great success towards continual learning of a sequence of tasks. Regularization-based approaches were proposed addressing catastrophic forgetting by regularizing the network parameters when learning new tasks [21, 39, 7] or regularizing the discrepancy between the output of the old and new network using a distillation loss [22, 30, 29, 6]. Modular compositional approaches [4, 11, 12] continually learn multiple tasks by combining different submodules, and each task is solved by a corresponding submodule. Memory buffer based approaches [25, 8] store a subset of training examples of previous tasks, thus requiring extra memory at training time.

For generative tasks, on the other hand, relatively less work is proposed addressing the problem of catastrophic forgetting and lifelong learning remains an under-explored area. Memory replay based approaches [35] form a joint training set by combining images generated from a model trained on previous tasks with the training images for the current task. However, memory replay is limited to label-conditioned image generation and is not applicable for image-conditioned generation scenarios since without previous conditional images, no images could be generated for replay. LifelongGAN [41] is a generic generative lifelong learning method regularizing the outputs of the model using knowledge distillation. However, the proposed auxiliary data generation techniques cannot fully address the conflicts caused by sharing the whole model across all tasks, resulting in degraded performance of either previous tasks or the

new task. PiggybackGAN [40] constructs filters of the new task by making use of filters from previously trained model, which remain freezed during the learning of the new task. To allow for more flexibility, unconstrained filters are also introduced for each new task, which largely increased the memory requirement. These prior work are not scalable due to either degraded performance or high storage requirement.

Hypernetworks. There has been increasing interest in generating parameters of neural networks using hypernetworks [13, 10, 5, 23, 37]. This idea has been applied to applications in different research fields such as few shot learning [27, 33], image segmentation [2, 36] and generative models [28, 18, 16], which is the focus of our paper. Producing the entire set of weights of a target generative model through hypernetworks would be computation and memory extensive. Therefore, most approaches would only predict the filters of certain layers. For instance, for U-Net generator [17], only the decoder would be dynamic, parameterized as hypernetworks while the encoder remains deterministic [24]; for Resnet generator [19, 42], there would be a fixed sub-model while only last few layers are parameterized as hypernetworks [34]. Oswald, Henning and Sacramento et al. [32] extend hypernetworks to the setting of lifelong learning. However, their approach is not applicable to image-conditioned generation and has the following drawbacks. First, their approach stores the previous task embeddings to generate different sets of parameters for different tasks. Second, their approach generates all parameters in a layer by using chunk embedding and network partitioning. As a result, compared with Hyper-LifelongGAN, the output size of their approach increases a hundredfold. Most importantly, their approach is not applicable to image-conditioned generation as memory replay is adopted to continually learn a sequence of generation tasks.

Hyper-LifelongGAN is a generic and scalable generative lifelong learning framework, enabling various generation tasks across different data domains. The architecture designs of the task specific base filter generators, shared weight matrix and task specific coefficients contribute to the high generation quality and low memory requirement, which make a clear difference from prior works.

3. Method

The goal of lifelong learning is to learn a model performing a sequence of generation tasks while assuming that the model is restricted to the training data for only the current task. We proposed Hyper-LifelongGAN addressing catastrophic forgetting for generative models. The overall architecture is illustrated in Figure 2. Given a sequence of tasks, Hyper-LifelongGAN decomposes the conventional convolutional and deconvolutional filters into dynamic base filters, which are generated by task specific filter generators, and deterministic weight matrix, which is shared across different tasks and linearly combines the generated base filters. To allow for more flexibility, the shared weight matrix is multiplied by task specific coefficients to combine the task specific filters in different ways for different tasks. The proposed Hyper-LifelongGAN is trained using knowledge distillation: knowledge is extracted from a previously trained model and distilled to the model trained for the new task, encouraging the new model to generate the same output as the previous model.

3.1. Hyper-LifelongGAN

When the t^{th} task T_t comes, the goal is to train a model M_t that could perform all tasks from task T_1 till task T_t while model M_t is restricted to the training data of the current task T_t. A naive approach to continually learn a sequence of tasks would be training a separate model for each task. However this approach is not scalable in general since the memory requirement increases drastically as new tasks are added. LifelongGAN [41], on the other hand, learns a sequence of tasks by sharing the whole model across all tasks. Due to the intrinsic differences among tasks, it is hard to adapt all parameters of a trained model to a new task, resulting in degraded performance in either previous tasks or the new task. Therefore, we propose to factorize the conventional convolutional and deconvolutional filters into a set of base filters and a weight matrix that linearly combines the base filters. And as new task comes, new set of base filters are learned while the weight matrix is not task conditional and is shared across all tasks. In this way, certain flexibility is granted to each task and at the meantime, extra parameters introduced for each task are largely reduced. Now we introduce the details of the filter factorization.

Convolutional filter factorization. Let the generator and discriminator be G_t and D in the model M_t. Assume the generator G_t consists of L layers of filters $\{F_t^\ell \in \mathbb{R}^{s_w^\ell \times s_h^\ell \times c_{in}^\ell \times c_{out}^\ell}\}_{\ell=1}^L$ where ℓ denotes the index of layers, s_w^ℓ is the the kernel width, s_h^ℓ is the the kernel height, c_{in}^ℓ is the number of input channels, and c_{out}^ℓ is the number of output channels. For simplicity, notation ℓ is dropped, we denote the filters using notation F_t. Then F_t is factorized into a set of base filters $\mathcal{B}_t \in \mathbb{R}^{(s_w \times s_h) \times K}$ and a weight matrix $\mathcal{W}_t \in \mathbb{R}^{K \times (c_{in} \times c_{out})}$. As a result,

$$F_t = \mathcal{R}(\mathcal{B}_t * \mathcal{W}_t), \tag{1}$$

where $\mathcal{R}$ is the reshaping operation that reshapes the output to 4D tensor.

To allow for greater flexibility in learning each task, M_t maintains different sets of base filters $\{\mathcal{B}_t^i\}_{i=1}^t$ for tasks from task T_1 till task T_t. And to make the model parameter efficient, weight matrix $\mathcal{W}_t$ is shared across all tasks from task T_1 till task T_t. By multiplying base filters by the

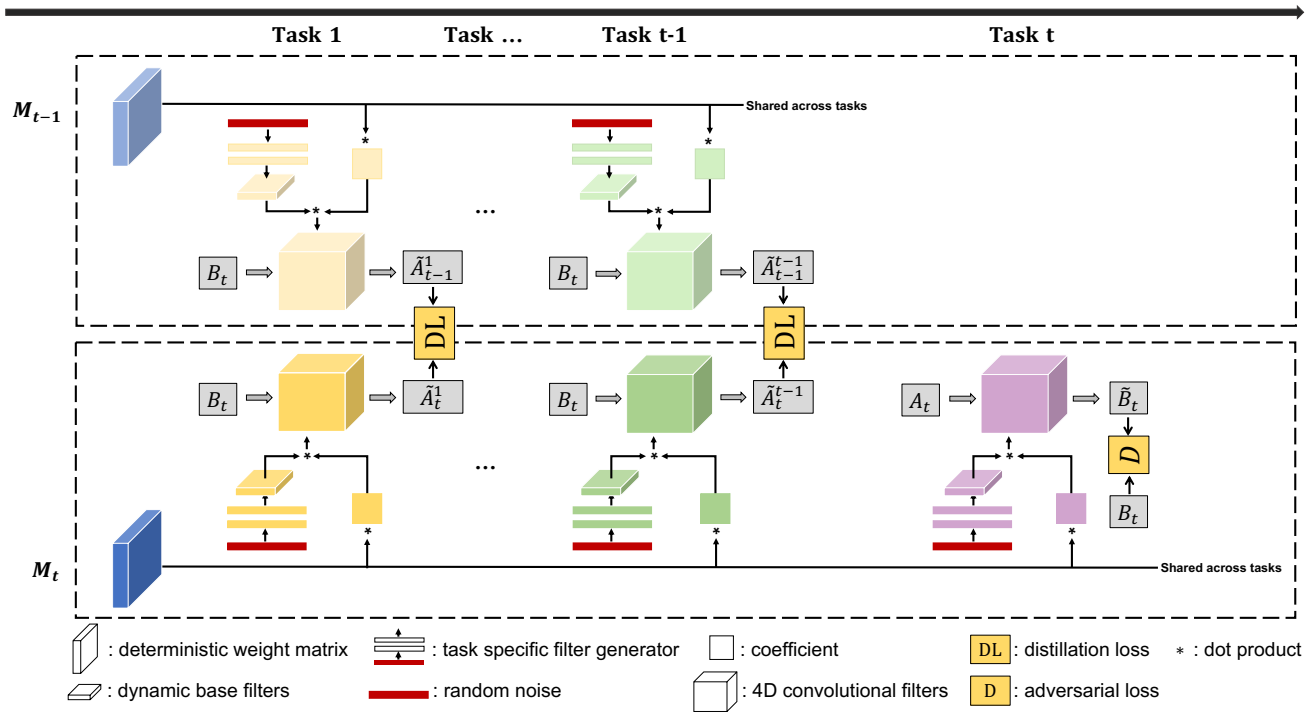

Figure 2: **Hyper-LifelongGAN.** Our method factorizes conventional convolutional filters into dynamic base filters, which are generated by task specific filter generators, and deterministic weight matrix, which is shared across all tasks. Moreover, task specific coefficients are adopted to introduce more flexibility in combining base filters differently for each task. To prevent the model from catastrophic forgetting, knowledge distillation is adopted to encourage the two networks to produce similar outputs.

shared weight matrix, M_t would have different sets of filters $\{F_t^i\}_{i=1}^t$ for different tasks, namely

$$F_t^i = \mathcal{R}(\mathcal{B}_t^i * \mathcal{W}_t). \tag{2}$$

Task specific filter generator. The above mentioned filter factorization though grants greater flexibility in learning different tasks, largely reduces the number filters learned without constraints (K filters and K is often very small to reduce the number of parameters). To address this problem, we propose to generate base filters $\{\mathcal{B}_t^i\}_{i=1}^t$ from random noise $\mathbf{z}$ using task specific filter generators $\{H_t^i\}_{i=1}^t$ for all tasks from T_1 till task T_t, and H_t^i is the base filter generator of model M_t for the task T_i. Specifically,

$$\mathcal{B}_t^i = H_t^i(\mathbf{z}). \tag{3}$$

By using filter generators, the number of base filters are no longer fixed and confined to K. Many more sets of base filters could be sampled from the vast parameter space by sampling different $\mathbf{z}$ from some pre-defined distribution, e.g. $\mathcal{N}(0, 1)$.

Since the weight matrix $\mathcal{W}_t$ is shared across all tasks, different base filters of different tasks are combined in the same way. It is more desirable to combine the base filters of each task differently. However, different from base filters, having a separate weight matrix for each task would be memory extensive. The reason is that the base filters in each layer could be generated with just a few thousand of parameters, while extra millions of parameters are need for each task to maintain task specific weight matrices. Therefore, we introduce the deterministic task specific coefficients $\{\mathcal{C}_t^i \in \mathbb{R}^{K\times K}\}_{i=1}^t$, multiplying with the shared weight matrix to allow for different combinations of base filters for different tasks. Specifically,

$$\begin{aligned} \mathcal{W}_t^i &= \mathcal{C}_t^i * \mathcal{W}_t \\ F_t^i &= \mathcal{R}(\mathcal{B}_t^i * \mathcal{W}_t^i). \end{aligned} \tag{4}$$

As a result, G_t can be viewed as a generator consisting of t sub-generators $\{G_t^i\}_{i=1}^t$, where the sub-generator G_t^i generates image for task T_i at the t^{th} time step.

3.2. Learning Hyper-LifelongGAN

In this paper, we explore two conditional generation scenarios: (1) *Paired image generation*, in which the training set for task T_t is $\mathbb{S}_t = \{(\mathbf{A}_{i,t}, \mathbf{B}_{i,t}) | \mathbf{A}_{i,t} \in \mathbb{A}_t, \mathbf{B}_{i,t} \in \mathbb{B}_t\}_{i=1}^{N_t}$ where N_t is the number of training instances in the

training set, and $\mathbb{A}_t$ and $\mathbb{B}_t$ denote the domain of conditional images and ground truth images respectively. For each conditional image $\mathbf{A}_{i,t}$, its corresponding ground-truth image $\mathbf{B}_{i,t}$ is provided. (2) *Unpaired image generation*, in which the training set for task T_t is $\mathbb{S}_t = \{(\{\mathbf{A}_{i,t}\}_{i=1}^{N_t^a}, \{\mathbf{B}_{i,t}\}_{i=1}^{N_t^b}) | \mathbf{A}_{i,t} \in \mathbb{A}_t, \mathbf{B}_{i,t} \in \mathbb{B}_t\}$. Different from paired image generation, the correspondence between $\mathbf{A}_{i,t}$ and $\mathbf{B}_{i,t}$ does not exist. For simplicity, notations $\mathbf{A}_t, \mathbf{B}_t$ are used referring to an instance from the respective domain.

Let M_{t-1} be the model trained for task T_{t-1}. Given the new task T_t, to prevent the current model M_t from forgetting previous tasks, the data of current task $\mathbb{S}_t$ is inputted to both M_t and M_{t-1}, and knowledge distillation loss is adopted to distill knowledge from M_{t-1} to M_t, encouraging the outputs of M_{t-1} and M_t to be the same. First, the outputs of the sub-generators of model M_{t-1} are computed as:

$$\begin{aligned} \widetilde{\mathbf{B}}_{t-1}^1 = G_{t-1}^1(\mathbf{A}_t, \mathbf{z}), ..., \widetilde{\mathbf{B}}_{t-1}^i = G_{t-1}^i(\mathbf{A}_t, \mathbf{z}), \\ ..., \widetilde{\mathbf{B}}_{t-1}^{t-1} = G_{t-1}^{t-1}(\mathbf{A}_t, \mathbf{z}). \end{aligned} \tag{5}$$

And the corresponding $t-1$ outputs of the sub-generators of model M_t are computed as:

$$\begin{aligned} \widetilde{\mathbf{B}}_t^1 = G_t^1(\mathbf{A}_t, \mathbf{z}), ..., \widetilde{\mathbf{B}}_t^i = G_t^i(\mathbf{A}_t, \mathbf{z}), \\ ..., \widetilde{\mathbf{B}}_t^{t-1} = G_t^{t-1}(\mathbf{A}_t, \mathbf{z}). \end{aligned} \tag{6}$$

Given these outputs, the knowledge distillation loss is defined as:

$$\mathcal{L}_{\text{distill}}^t = \sum_{i=1}^{t-1} ||\widetilde{\mathbf{B}}_{t-1}^i - \widetilde{\mathbf{B}}_t^i||_1. \tag{7}$$

Moreover, $\mathbb{S}_t$ is also inputted to M_t to minimize the loss $\mathcal{L}_{\text{task}}$ [1] related with current task T_t, namely

$$\begin{aligned} \widetilde{\mathbf{B}}_t^t &= G_t^t(\mathbf{A}_t, \mathbf{z}), \\ \mathcal{L}_{\text{task}}^t &= \mathcal{L}_{\text{task}}(\mathbf{A}_t, \mathbf{B}_t^t, \widetilde{\mathbf{B}}_t^t). \end{aligned} \tag{8}$$

And the total loss of Hyper-LifelongGAN at the t^{th} time step is defined as:

$$\mathcal{L}_{\text{total}}^t = \mathcal{L}_{\text{task}}^t + \beta \mathcal{L}_{\text{distill}}^t, \tag{9}$$

where β is the loss weight for knowledge distillation.

Data for Knowledge Distillation. There are conflicts in Equation 9: given the same input $(\mathbf{A}_t, \mathbf{z})$, there are two training goals $\mathcal{L}_{\text{task}}^t$ and $\mathcal{L}_{\text{distill}}^t$. $\mathcal{L}_{\text{task}}^t$ would encourage the model to produce an output belonging to domain $\mathbb{B}_t$, while $\mathcal{L}_{\text{distill}}^t$ encourages the model to produce an output belonging to previous domains, e.g. $\mathbb{B}_{t-1}$. Though the task specific base filters could alleviate the conflicts, it still would be beneficial to use different inputs for the two losses. Therefore, we propose to use real image $\mathbf{B}_t$ as input for knowledge distillation $\mathcal{L}_{\text{distill}}^t$, while the input remains $\mathbf{A}_t$ for learning the new task $\mathcal{L}_{\text{task}}^t$. In other words, the conditional and real images are *swapped* for the two training losses (see Figure 2):

$$\begin{aligned} \mathcal{L}_{\text{total}}^t = \mathcal{L}_{\text{task}}(\mathbf{A}_t, \mathbf{B}_t^t, G_t^t(\mathbf{A}_t, \mathbf{z})) \\ + \beta \sum_{i=1}^{t-1} ||G_{t-1}^i(\mathbf{B}_t, \mathbf{z}) - G_t^i(\mathbf{B}_t, \mathbf{z})||_1. \end{aligned} \tag{10}$$

4. Experiments

We evaluate Hyper-LifelongGAN under two settings: (1) paired image generation, and (2) unpaired image generation. First, ablation studies on the number of base filters K, different types of input data for knowledge distillation and model components are conducted. Then we compare our model with 5 baselines, including the state-of-the-art approaches LifelongGAN [41] and PiggybackGAN [40].

Training Details. All the generative models are trained on images of size 128×128. We use the Tensorflow [1] framework with Adam optimizer [20]. The loss weight for knowledge distillation β is set to 100.0 for all experiments. For Hyper-LifelongGAN and all baseline methods, we use the Resnet generator [19, 42] with 6 residual blocks. The length of random noise $\mathbf{z}$ is set to 64, the task specific filter generator is a MLP with hidden layer of size 64 in all experiments.

Baseline Models. We compare Hyper-LifelongGAN to the following baseline models: (a) *Hyper-Full*: The model is trained on single task, the generator is decomposed into the base filter generator and the weight matrix. (b) *Full*: The model is trained on single task, and the generator consists of conventional convolutional filters. (c) *Sequential Fine-tuning (SFT)*: The model is fine-tuned in a sequential manner, with parameters initialized from the model trained/fine-tuned on the previous task. (d) *PiggybackGAN*: We trained PiggybackGAN [40] with $\lambda = 0.5$ in all experiments. (d) LifelongGAN++: We propose an improved LifelongGAN [41] baseline, using task-conditional instance normalization as Hyper-LifelongGAN for more fair comparisons.

Quantitative Metrics. Two metrics *Acc* and *Frchet Inception Distance (FID) [14]* are used to evaluate the generation quality. *Acc* is the classification accuracy of the classifier trained on real images and evaluated on generated images (higher Acc indicates better generation results). *FID* is an extensively used metric to compare the statistics of gener-

[1] For example, if the t^{th} task is conditional image generation based on Pix2Pix [17], $\mathcal{L}_{\text{task}}^t = \mathcal{L}_{\text{cGAN}}(G_t^t, D) + \alpha \mathcal{L}_{\text{L1}}(G_t^t)$, which is the exact loss used in Pix2Pix.

ated images to the ground-truth images (lower FID indicates higher generation quality).

4.1. Paired Image-conditioned Generation

We first demonstrate the effectiveness of Hyper-LifelongGAN on a sequence of 4 paired image generation tasks on challenging datasets with large variations across different modalities [9, 17, 31, 38]. The first task is *segmentations* $\rightarrow$ *street photos*, the second task is *maps* $\rightarrow$ *aerial photos*, the third task is *semantic labels* $\rightarrow$ *facades*, and the fourth task is *edges* $\rightarrow$ *handbag photos*.

Ablation study on the base filter size K. First we conduct an ablation study on the choice of different values of K, which determines the number of additional parameters needed for each subsequent task. As observed from the quantitative result in Table 1, the model performs best when $K = 7$. Therefore, K is set to 7 in all later experiments.

	K=3	K=5	K=7
Acc	66.00	76.60	75.40
FID	72.96	72.25	57.58

Table 1: Ablation study on K. Different models are trained and evaluated on the initial tasks *cityscapes*, and corresponding Acc and FID are reported.

Ablation study on the data for knowledge distillation. We explored three types of inputs for computing the knowledge distillation loss $\mathcal{L}^t_{\text{distill}}$, which are listed below.

(1) Unswap. Conditional image $\mathbf{A}_t$ is inputted to the model for computing both the distillation loss $\mathcal{L}^t_{\text{distill}}$ and the current task loss $\mathcal{L}^t_{\text{task}}$.

(2) Swap. When computing the losses $\mathcal{L}^t_{\text{distill}}$ and $\mathcal{L}^t_{\text{task}}$, conditional image and real image are swapped. Namely real image $\mathbf{B}_t$ is used as input for computing the distillation loss $\mathcal{L}^t_{\text{distill}}$ and conditional image $\mathbf{A}_t$ is used as input for computing the current task loss $\mathcal{L}^t_{\text{task}}$.

(3) Random noise. Random noise is sampled and inputted to the model for computing the distillation loss $\mathcal{L}^t_{\text{distill}}$, and conditional image $\mathbf{A}_t$ is used as input for computing the current task loss $\mathcal{L}^t_{\text{task}}$.

It is observed from Table 2 that *swapping* conditional image and real image for computing the two losses $\mathcal{L}^t_{\text{distill}}$ and $\mathcal{L}^t_{\text{task}}$ provides best results as it avoids the conflicting training objectives. And unlike *random noise*, it provides inputs with variations across different modalities, which could be beneficial for lifelong learning. In all later experiments, we adopt the *swapping* strategy.

Ablation study on the model components. We also conduct an ablation study on the model components as shown in Table 3, to test whether each component of our model is necessary. *Task specific base filters* denotes whether there is a separate set of base filters for each task. *Dynamic base filters* denotes whether the base filters are dynamic (generated by filter generators). *Coeff* denotes whether task specific coefficients are used. For instance, the first row in Table 3 refers to the baseline that the base filters are generated using hypernetworks and are shared across all tasks, and coefficients are not adopted.

	random noise	unswap ($\mathbf{A}_t$)	swap ($\mathbf{B}_t$)
Acc	91.92	91.27	92.55
FID	110.84	118.98	101.43

Table 2: Ablation study on the data for knowledge distillation. Different models are trained and evaluated on all tasks, average Acc and FID score over all 4 tasks are reported.

dynamic base filters	task specific base filters	coeff	Acc	FID
✗	✗	✗	89.58	137.99
✓	✗	✗	89.60	114.73
✓	✓	✗	91.77	105.42
✓	✓	✓	92.55	101.43

Table 3: Ablation study on model components. Different models are trained and evaluated on all tasks, average Acc and FID score over all 4 tasks are reported.

Comparison with SOTA methods and baselines. We compare Hyper-LifelongGAN with two most recent state-of-the-art approaches PiggybackGAN [40] and an improved version of LifelongGAN [41], and three baselines Hyper-Full, Full, and Sequential Fine-tuning (SFT). Baseline Full is provided since it serves as the "upper bound" approach for LifelongGAN++ and PiggybackGAN, both of which are built on the conventional convolutional filters as in Pix2Pix [17] as the Full model.

The visualization of images generated from all approaches are shown in Figure 3 and the quantitative evaluations of all approaches are summarized in Table 4. It is observed that the sequentially fine-tuned model suffers catastrophic forgetting: after the final task is learned, it completely forgets all previous tasks and can only generate images with edges2handbags-like patterns. Both LifelongGAN++ and PiggybackGAN can remember previous tasks while learning the new task. However, LifelongGAN++ cannot learn the new task well while preserving the performance of previous tasks, and the performance of previous tasks may degrade while adapting the model to the new task. Though PiggybackGAN achieves a performance on par with the Full model, it introduces millions of additional parameters for each new task. While Hyper-LifelongGAN

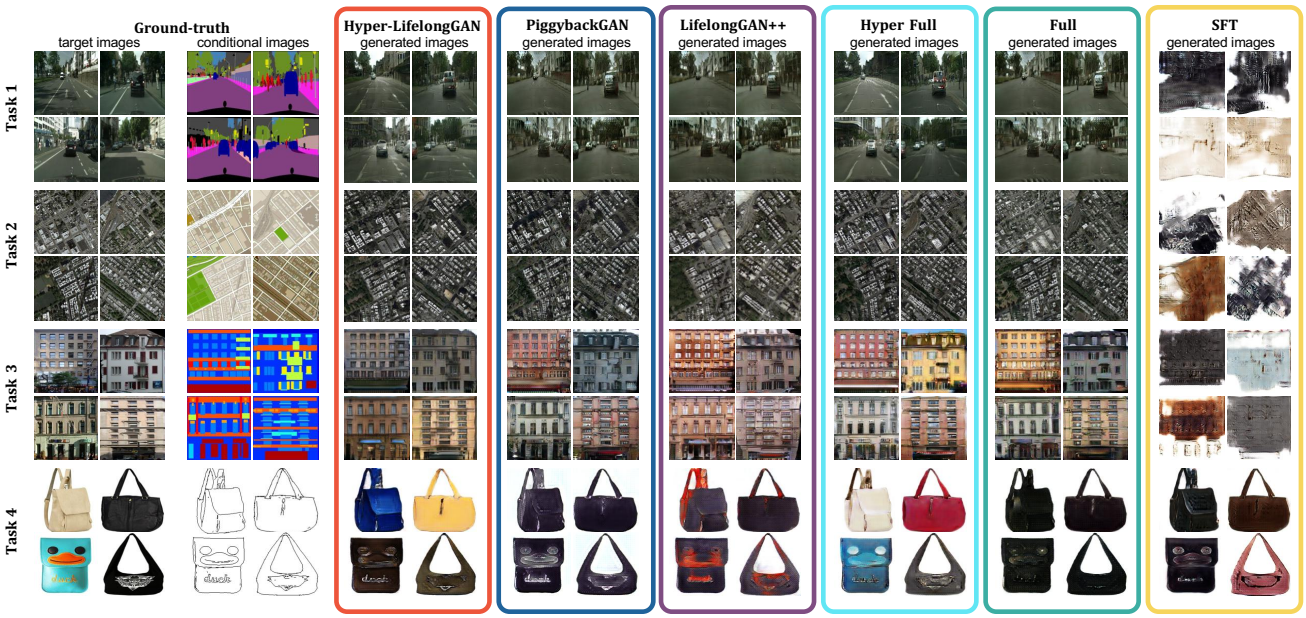

Figure 3: Visualizations of images generated from different approaches for paired image-conditioned generation. Sequential Fine-tuning suffers from catastrophic forgetting: when we add new tasks, the network forgets how to perform previous tasks. Hyper-LifelongGAN generates high quality images for both previous tasks and the new task. It can well preserve the knowledge from previous tasks while learning the current task well at a low memory requirement.

	Hyper Full	Hyper LifelongGAN	SFT
Acc	90.30	92.55	25.46
FID	100.53	101.43	250.37
	Full	LifelongGAN^{++}	PiggybackGAN
Acc	89.42	89.58	88.53
FID	126.01	137.99	128.23

Table 4: Quantitative evaluation among different approaches for continual learning of paired image-conditioned generation tasks. Different models are trained and evaluated on all tasks, average Acc and FID score over all 4 tasks are reported.

can better preserve or even increase the generation quality of given tasks (e.g. cityscapes and maps) at a low memory requirement: for most layers in Hyper-LifelongGAN (besides the first and the last layer), ~8k parameters are introduced for each new task.

4.2. Unpaired Image-conditioned Generation

We also apply Hyper-LifelongGAN to another challenging scenario: unpaired image-conditioned generation, translating images from domain $\mathbb{A}$ to domain $\mathbb{B}$ and the correspondence between domain $\mathbb{A}$ and domain $\mathbb{B}$ does not exist.

We explored a special situation where two tasks in a given sequence share the same input domain but have different output domains, e.g. T_1 is *Photo* $\rightarrow$ *Monet Paintings* and T_2 is *Photo* $\rightarrow$ *Ukiyo-e Paintings*. The training goals of the two tasks completely conflicts each other: the inputs are exactly the same while output are different. The goal is to verify whether Hyper-LifelongGAN is generic and powerful enough to handle this special case well.

Comparison with SOTA methods and baselines. Same as paired image-conditioned generation, we compare Hyper-LifelongGAN with two most recent state-of-the-art approaches PiggybackGAN [40] and an improved version of LifelongGAN [41], and three baselines Hyper-Full, Full, and Sequential Fine-tuning (SFT). Baseline Full is provided since it serves as the "upper bound" approach for LifelongGAN++ and PiggybackGAN, both of which are built on the conventional convolutional filters as in CycleGAN [42] as the Full model.

The visualization of images generated from all approaches are shown in Figure 4 and the quantitative evaluations of all approaches are summarized in Table 6. Since the two tasks share the same input space, though SFT can generate realistic images depicting the correct contents, it can only generate images with *Ukiyoe* style and completely forgets the *Monet* style learned in the initial task. Both LifelongGAN++ and PiggybackGAN can generating images with *Monet* and *Ukiyoe* styles. However, the smaller gap between Hyper-LifelongGAN and Hyper-Full indicates that Hyper-LifelongGAN can better preserve knowledge ac-

	Hyper-Full	Hyper-LifelongGAN	Full	Lifelong-GAN++	Piggyback-GAN	SFT
Task 1	6.27M	6.27M	7.84M	7.84M	7.84M	6.27M
Task 2	12.54M	6.46M	15.68M	7.85M	12.22M	6.27M
Additional	6.27M	0.19M	7.84M	0.01M	4.38M	0M

Table 5: **The number of parameters of each model.** This table shows the size of the generator for unpaired generation. Additional parameters needed for each subsequent task of each model are shown in the last row.

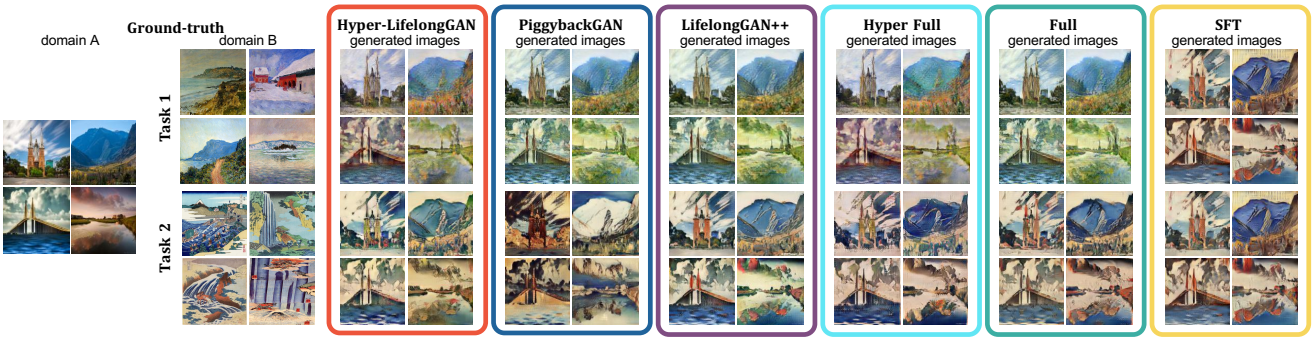

Figure 4: Visualizations of images generated from different approaches for unpaired image-conditioned generation. Sequential Fine-tuning suffers from catastrophic forgetting: when learning the *Ukiyoe* style, it forgets the *Monet* style. Hyper-LifelongGAN generates high quality images depicting correct styles for both tasks. It can well preserve the knowledge from previous tasks while learning the current task well at a low memory requirement.

	Hyper Full	Hyper LifelongGAN	SFT
Acc	72.90	73.30	50.00
FID	98.95	96.79	139.03
	Full	LifelongGAN^{++}	PiggybackGAN
Acc	72.30	70.17	70.51
FID	102.95	110.60	104.51

Table 6: Quantitative evaluation among different approaches for continual learning of unpaired image-conditioned generation tasks. Different models are trained and evaluated on all tasks, average Acc and FID score over all 2 tasks are reported.

quired from previous tasks and learn the current task well. With a freezed weights of previous task, PiggybackGAN is able to maintain the exact performance for previous tasks. However, as observed from Table 6, it is possible for Hyper-LifelongGAN to improve the generation quality compared with separate models trained for each task.

Parameter efficiency. The parameter efficiency of different models for unpaired generation are shown in Table 5. Hyper-LifelongGAN requires additional 0.19M parameters for each new task. When computing distillation loss, restoring previous model also requires additional 0.19M parameters. However, once the model is learned, previous model can be discarded. Hyper-LifelongGAN is more scalable since it can best maintain the generation quality at a low cost of parameters.

5. Conclusion

A generic and scalable lifelong learning algorithm Hyper-LifelongGAN for generative models is proposed in this paper. It decomposes the conventional convolutional filters into the dynamic task specific base filters and a deterministic generic weight matrix. Attributed to the novel architecture, fine details in each task can be well captured and learned, and information in previous tasks can be well preserved. Moreover, since the weight matrix is shared across all tasks and dynamic base filters in each layer can be generated with just few thousand parameters, the memory requirement of Hyper-LifelongGAN is low. As a result, compared with previous state-of-the-art approaches, Hyper-LifelongGAN is more scalable as it can generate high quality images for all tasks at a low cost of parameters. The proposed approach is validated on various image-conditioned generation tasks across different domains, and the qualitative and quantitative results are provided to show that Hyper-LifelongGAN addresses catastrophic forgetting effectively and efficiently.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *Symposium on Operating Systems Design and Implementation (OSDI)*, 2016. 5

[2] Iasonas Kokkinos Adam W Harley, Konstantinos G. Derpanis. Segmentation-aware convolutional networks using local attention masks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 3

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

[4] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[5] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip H. S. Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 3

[6] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[7] Arslan Chaudhry, Puneet K Dokania Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[10] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 3

[11] C. Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David R Ha, Andrei A. Rusu, A. Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017. 2

[12] Leslie Pack Kaelbling Ferran Alet, Tomas Lozano-Perez. Modular meta-learning. In *Conference on Robot Learning (CoRL)*, 2018. 2

[13] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *International Conference on Learning Representations (ICLR)*, 2017. 2, 3

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 5

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS workshop on Deep Learning and Representation Learning*, 2015. 2

[16] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 5, 6

[18] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 3, 5

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 2017. 2

[22] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2

[23] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees G. M. Snoek, and Arnold W. M. Smeulders. Tracking by natural language specification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[24] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[25] David Lopez-Paz and MarcAurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[26] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*. 1989. 2

[27] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[28] Falong Shen, Shuicheng Yan, and Gang Zeng. Neural style transfer via meta networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[29] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[30] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[31] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition (GCPR)*, 2013. 6

[32] Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F. Grewe. Continual learning with hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2020. 3

[33] Xin Wang, Fisher Yu, Ruth Wang, Trevor Darrell, and Joseph E Gonzalez. Tafe-net: Task-aware feature embeddings for low shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[34] Ze Wang, Xiuyuan Cheng, Guillermo Sapiro, and Qiang Qiu. Stochastic conditional generative networks with basis decomposition. In *International Conference on Learning Representations (ICLR)*, 2020. 3

[35] Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. Memory replay gans: learning to generate images from new categories without forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2

[36] Jialin Wu, Dai Li, Yu Yang, Chandrajit Bajaj, and Xiangyang Ji. Dynamic filtering with large sampling field for convnets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3

[37] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016. 3

[38] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6

[39] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, 2017. 2

[40] Mengyao Zhai, Lei Chen, Jiawei He, Megha Nawhal, Frederick Tung, and Greg Mori. Piggyback gan: Efficient lifelong learning for image conditioned generation. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 5, 6, 7

[41] Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong gan: Continual learning for conditional image generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 5, 6, 7

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 3, 5, 7