This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

SDDGR: Stable Diffusion-based Deep Generative Replay for Class Incremental Object Detection

Junsu Kim[1] Hoseong Cho [1,2†] Jihyeon Kim[1,3†] Yihalem Yimolal Tiruneh[1] Seungryul Baek[1]

[1]UNIST [2]LG Electronics [3]KETI

Abstract

In the field of class incremental learning (CIL), generative replay has become increasingly prominent as a method to mitigate the catastrophic forgetting, alongside the continuous improvements in generative models. However, its application in class incremental object detection (CIOD) has been significantly limited, primarily due to the complexities of scenes involving multiple labels. In this paper, we propose a novel approach called stable diffusion deep generative replay (SDDGR) for CIOD. Our method utilizes a diffusion-based generative model with pre-trained text-to-image diffusion networks to generate realistic and diverse synthetic images. SDDGR incorporates an iterative refinement strategy to produce high-quality images encompassing old classes. Additionally, we adopt an L2 knowledge distillation technique to improve the retention of prior knowledge in synthetic images. Furthermore, our approach includes pseudo-labeling for old objects within new task images, preventing misclassification as background elements. Extensive experiments on the COCO 2017 dataset demonstrate that SDDGR significantly outperforms existing algorithms, achieving a new state-of-the-art in various CIOD scenarios.

1. Introduction

The key challenge in artificial intelligence is the development of models capable of continuous learning, similar to human knowledge accumulation over a lifetime. This challenge has sparked the field of class incremental learning (CIL), the continual learning in the classification task. The CIL focuses on developing techniques that enable models to learn new classes without compromising previously acquired knowledge.

To address the challenge, researchers have primarily focused on mainly two strategies: knowledge distillation [16, 33, 38, 47, 61], replay [5–7, 12, 17, 47, 57, 58, 66, 67]. Among these, replay has been employed as a prominent solution in addressing the challenge of forgetting. Replay can be classified into two categories: partial experience replay [5, 7, 17, 47, 57] and deep generative replay [6, 12, 58, 66, 67]. The partial experience replay needs to store actual data samples from old tasks, acting as a reservoir of previous knowledge for the model. On the other hand, generative replay employs generative models to mimic the distribution of old task's data, enabling the current model to re-experience the previous knowledge.

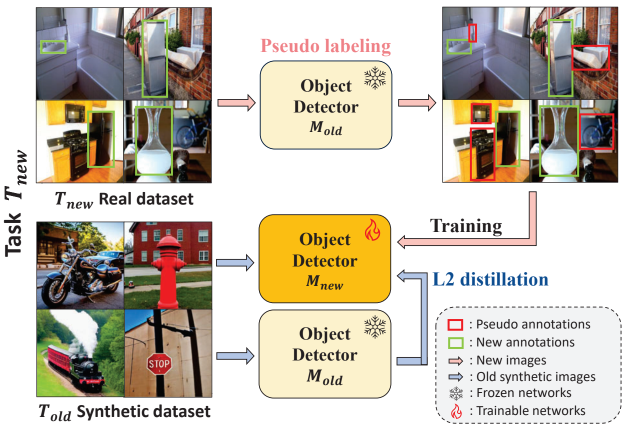

Figure 1. We utilize a pre-trained text-to-image diffusion model [51] to generate realistic images that include objects from the old task. These images are then filtered out via iterative refinement and filtered synthetic images are integrated into the training process of the new task. During training, we employ L2 distillation to a synthetic dataset. Additionally, when training an image for the new task, we employ a pseudo-labeling that finds the old task objects from the new task images. The series of methods enable us to effectively mitigate the issue of catastrophic forgetting.

These methods have made significant progress in the field of image classification when there is only a single object present in an image. Yet, there has been a pressing need for techniques that can handle more complex and realistic scenes including multi-labels in a scene based on the object detection algorithms. Consequently, class incremental object detection (CIOD) has emerged, with the goal of improving models to detect multiple labels in a scene while still preserving the ability to recognize previously learned object classes.

Initial researches [1, 11, 35, 60] in class incremental object detection (CIOD) extended image classification methods

This research was conducted when Hoseong Cho and Jihyeon Kim were graduate students (Master candidates) at UNIST†.

to CIOD, showing encouraging results. Furthermore, as the Transformer-based architectures [3, 29, 71] are introduced as the alternative to the CNN-based approaches [31, 48], the CIOD for Transformer-based object detector is also proposed. Specifically, Gupta *et al*. [15] and Liu *et al*. [37], which utilize deformable-DETR [71], have introduced distinctive characteristics into Transformer-based object detection, also incorporating partial experience replay in their methodologies. Despite significant advancements, they still heavily rely on the direct use of real data.

In parallel to the advancements in incremental learning, generative models have seen noticeable advancements. Moving away from traditional generative models like generative adversarial networks (GANs) [13, 40] and Variational autoencoders (VAEs) [26], recent image generation has focused on more sophisticated and realistic techniques, such as diffusion models [9, 20, 62, 63]. Notably, the stable diffusion (SD) method [51], which has been trained on a vast amount of online knowledge [53, 54], has gained significant attention for its impressive performance. This has led to various studies [32, 65, 69, 70] to utilize the model's capabilities with its original weights fixed. Motivated by the research trends, we proposed to utilize the pre-trained SD network for high-quality image generation to prevent the catastrophic forgetting. While generative models like the SD have shown proficiency in reproducing knowledge from the prompt, their effectiveness in multi-label scenarios, such as CIOD, remains constrained by the complexity of the scenario. However, we observed that the naïve application of SD is not suitable for successful CIOD. Thus, we proposed to improve the SD to control it based on grounding inputs such as classes and bounding boxes, via GLIGEN [32]. Furthermore, we proposed series of methods to secure the generated image quality.

In this study, we introduce the stable diffusion-based deep generative replay (SDDGR) strategy, a novel method for utilizing a pre-trained generative model for mitigating the catastrophic forgetting in CIOD. The SDDGR generates images by using grounding inputs and prompts that explain complex scenes, which include previously learned objects. However, we observed that the pre-trained SD weights are sub-optimal for the CIOD. To relieve the issue, we further proposed to refine the image fidelity through iterative refinement via a trained detector. Additionally, we trained a model using the L2 distillation to facilitate effective knowledge transfer from these synthetic images to the updated model, rather than the direct training. Simultaneously, we perform the pseudo-labeling for the old task's objects which exist in the new task's training images, to prevent it from being detected as the background. Using series of proposed methods, the SDDGR demonstrates excellent performance on the COCO dataset, achieving state-of-the-art accuracy. The overall training process of our method is shown in Figure 1. Our contributions are summarized as follows:

- As far as we are aware, we, for the first time, proposed to apply the diffusion-based generative model in CIOD problem.
- Naïvely applying the diffusion model to CIOD can decrease the overall accuracy. To make it properly work, we introduced series of methods to improve the generated image quality, to prevent the overfitting or mis-led information during training.
- The extended experiments demonstrate state-of-the-art performance on the COCO dataset, substantiating its efficacy in various CIOD scenarios.

2. Related works

2.1. Continual learning

Class-incremental learning (CIL) is a subset of continual learning, aiming to seamlessly integrate new classes into a model while maintaining the ability to recognize existing ones. Most influential CIL studies focused on classification, where one image represents a single class. In our paper, unless otherwise specified, CIL refers to the classification task. On the other hand, class incremental object detection (CIOD) presents a challenging task due to the presence of multiple instances belonging to various classes within images. When instances are trained under different tasks, they cannot be trained simultaneously. This can cause the model to classify these instances as background, which in turn degrades detection performance sequentially. Despite the clear challenges, CIOD has received relatively less research attention compared to CIL due to its complex nature.

Class incremental learning. In CIL, we can cluster the main methods into knowledge distillation [16, 33, 38, 61], and replay [5–7, 12, 17, 22, 47, 50, 57, 58, 66, 67] in general. Among these, Replay methods are most frequently utilized for their simple yet powerful effects and can be categorized into two types: partial experience replay (ER) [2, 4, 14, 27, 44, 47] and generative replay (GR) [6, 12, 23, 58, 66, 67]. The former involves reusing a subset of the original data repeatedly, while The latter employs a generative model to recreate the data distribution of previous tasks, effectively mitigating the forgetting [49]. In the GR, DGR [58] is an initial attempt of the GR method to prevent the loss of incremental classes using GAN [13]. MRGAN [66] and ILCAN [67], which evolved from DGR. Furthermore, DDGR [12] leverages advanced generative models, particularly diffusion-based techniques, to enhance the fidelity and variability of generated data. However, these methods have mainly been used in simpler scenarios for CIL because they require significant resources for training a generative. In our research, we shift the focus to applying advanced generative models within CIOD.

Class incremental object detection. CIOD has progressed from primarily employing CNN-based methods [1, 11, 28, 36, 43, 60] to also incorporating Transformer-based approaches [10, 15, 24, 25, 37]. In this trend, ILOD [60], a pioneering work in CIOD, implemented the LWF [33] method to handle forgetting. Besides, Feng *et al*. [11] focuses on maximizing the utility of heads in the distillation. More recent developments like CL-DETR [37] and OW-DETR [15] have adopted the Deformable DETR [71] (D-DETR) as a baseline. CL-DETR employs knowledge distillation at the level of the labels, utilizing an old

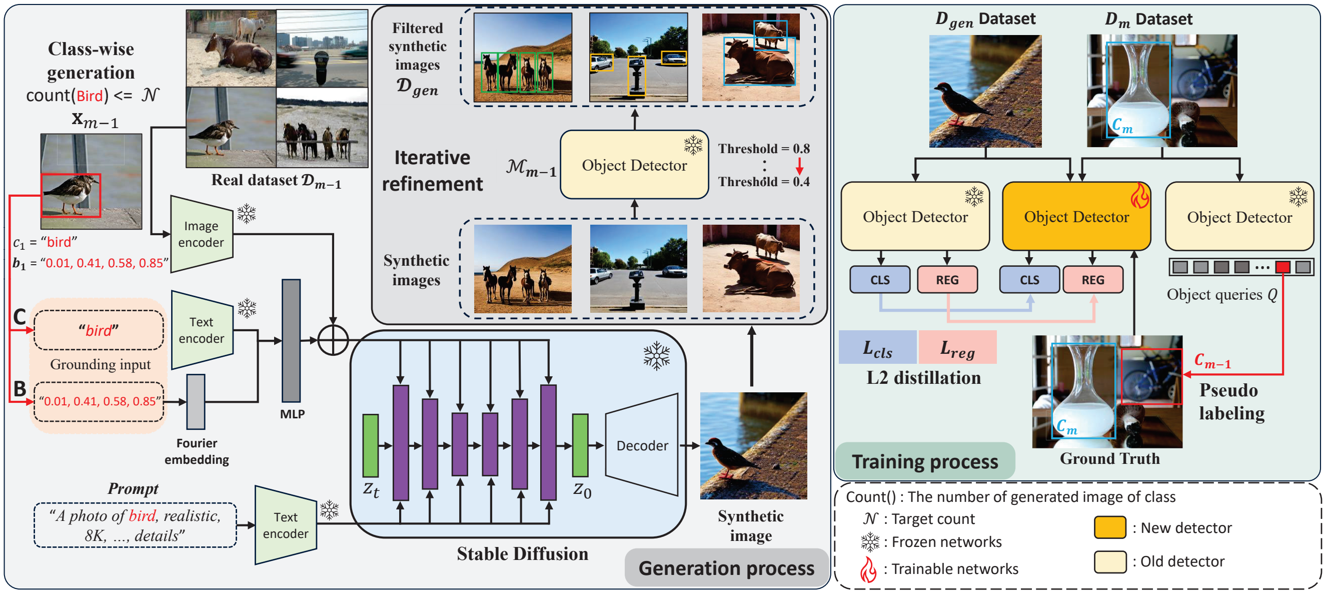

Figure 2. *Schematic of Our SDDGR Framework:* In the 'Generation process', our method individually generates each image based on class labels $\mathbf{C}_{\text{label}}$, specific bounding box locations $\mathbf{B}_{\text{location}}$, and old real images $\mathbf{x}_{m-1}$ in the old dataset $\mathcal{D}_{m-1}$. An 'Iterative refinement', employing the trained model $\mathcal{M}_{m-1}$, is applied to these synthetic images. In this algorithm, images with object scores below a dynamically adjusted threshold (ranging from 0.8 to 0.4 in our study) are systematically excluded. This cycle of generation and dynamic refinement continues until each class reaches the pre-defined target number of instances $\mathcal{N}$, or the lower threshold limit is met. In the 'Training process', the synthetic dataset is utilized for the continual learning via L2 distillation loss. Furthermore, real images undergo pseudo-labeling before being incorporated into the 'Training process'.

model to perform this process. OW-DETR introduce attention-driven pseudo-labeling, helping to identify unrecognized labels. In this study, we use D-DETR as a base detector to exploit the advantages of DETR [3] and compare its performance.

2.2. Diffusion models

Diffusion models have been largely researched due to their powerful generation capability. [20, 62, 63] proposed a basic framework for training through U-Net [52]. [9, 19] have demonstrated superior results compared to GAN [13] and VAE [26] based methods. However, since these models typically operate directly in pixel space, substantial computational resources are required. To solve this problem, Rombach *et al.* [51] proposed latent diffusion model(LDM), which performs the diffusion steps in latent space. They leveraged large-scale datasets such as LAION [53] and the pre-trained BERT [8] for text-to-image synthesis. This approach enables the incorporation of conditions during the image generation process, leading to the generation of desired images. Building on this foundation, Stability AI advanced the field further by developing stable diffusion (SD). SD utilizes an even larger dataset [54] and incorporates pre-trained CLIP [46]. Recent research has focused on using pre-trained SD as a foundation to effectively leverage the extensive knowledge. [21, 32, 41, 65, 70] have gained popularity for controlled generation by incorporating additional conditions. [59, 68] have demonstrated performance enhancements by generating additional data for training. In line with these advancements, our study employs pre-trained SD as a form of generative replay model to prevent the forgetting of previous knowledge.

3. Preliminaries

3.1. Stable diffusion

Stable diffusion (SD) [51] includes an VAE [26] structure for first extracting the latent vector $\mathbf{z} \in \mathbb{R}^{64\times64}$ from the image $\mathbf{x} \in \mathbb{R}^{512\times512}$ and gaining the same dimensional reconstructed images $\hat{\mathbf{x}}$ from the latent vector $\mathbf{z}$. It uses also a U-Net [52] architecture to add Gaussian noise to the latent vector and to remove the noise during the backward process, which is called the diffusion process [20, 62, 63]. By leveraging the text embedding of CLIP [45] and cross-attention [64], SD efficiently generates images based on the text prompt $\mathbf{T}$.

The core function is $f_\theta(\mathbf{z}_t, t, \mathbf{T})$, where the trained U-Net is used for f_θ, t denotes the time embedding and $\mathbf{z}_t$ represents the latent representation at the t-th diffusion time step. Although SD is adept at generating images from assigned prompts $\mathbf{T}$, it lacks the capability to utilize additional grounding inputs that would guide the generation process in terms of specific locations and categories of objects, thus limiting the elaboration of images.

3.2. Controllable image generation

To exploit the SD in the context of CIOD, we need to involve additional conditions such as bounding boxes and classes particularly when generating images with scenes containing multiple objects. However, as pointed out before, the SD lacks such a capability. To address this limitation, we extended the SD to incorporate the additional guiding inputs, following GLIGEN [32] (Unless otherwise noted, subsequent references to SD in this paper denote the SD whose grounding capability is enhanced by the use of GLIGEN [32].) This approach is able to leverage the pre-trained knowledge in the SD; while using the grounding inputs, such as classes and bounding boxes (bbox) additionally to the original text prompt $\mathbf{T}$. Grounding inputs are represented as classes and bounding boxes for N objects in an image as follows:

$$\mathbf{C}_{\text{label}} = [c_1, ..., c_N], \tag{1}$$

$$\mathbf{B}_{\text{location}} = [\mathbf{b}_1, ..., \mathbf{b}_N], \tag{2}$$

where each c_i represents a specific class within the set of trained classes C, and $\mathbf{b}_i$ denote the corresponding bounding box's normalized coordinate values $[x_{min}, y_{min}, x_{max}, y_{max}]$ for the i-th instance, respectively. Now, the SD becomes to combine the text prompt $\mathbf{T}$ with grounding tokens $\mathbf{C}_{\text{label}}$ and $\mathbf{B}_{\text{location}}$ using a gated self-attention mechanism, to generate accurate images. The diffusion function f_θ is then modified to incorporate grounding inputs:

$$f_\theta(\mathbf{z}_t, t, \mathbf{T}, \mathbf{C}_{\text{label}}, \mathbf{B}_{\text{location}}). \tag{3}$$

Furthermore, a hyper-parameter $\beta \in [0,1]$ is used to handle the influence of grounding inputs over the diffusion process.

4. Methods

The objective of class incremental object detection (CIOD) is to progressively assimilate new classes without compromising the knowledge of previously learned classes. This paradigm is characterized by a sequence of tasks, each represented as $\mathcal{T}_m$, where $m \in [1, M]$ and M denote the cumulative number of tasks. Each task contains specific data, represented as $\mathcal{D}_m$. Specifically, the dataset $\mathcal{D}_m$ consists of a set of input images $\mathcal{X}_m = \{\mathbf{x}^1_m, \ldots, \mathbf{x}^D_m\}$ and a set of corresponding annotations $\mathcal{Y}_m = \{\mathbf{y}^1_m, ..., \mathbf{y}^D_m\}$, where D is the data length. It is important to note that in object detection, individual annotations $\mathbf{y}^i_m$ consist of multiple object instances. We also follow the conventional CIOD configuration [11, 60], which implies that some images can be shared across different tasks.

Our approach, called SDDGR, consists of four key modules: 1) A method to generate images that include previous class objects (Section 4.1), 2) A technique for filtering more expressive images (Section 4.2), 3) A method for implementing pseudo labeling of the DETR framework (Section 4.3), and 4) A training protocol for using the synthetic images (Section 4.4). Figure 2 provides a comprehensive overview of these components.

4.1. Image generation

Text prompt Design. To generate images that accurately reflect the object categories of previous tasks, we carefully design text prompts $\mathbf{T}$ that encapsulate object classes of $\mathcal{Y}_{m-1}$ from the previous dataset $\mathcal{D}_{m-1}$. Initially, we identify the object labels, as multiple objects may appear in a single image for CIOD. Subsequently, we count the occurrence of each object and express the number using words (e.g., one, two, etc.). As a result, we frame our prompts to reflect both the object category and the number of occurrences: "A photo of{count}{object A},{count} {object B}, and{count}{object C},{scene environments}". The term {scene environments} is included at the end of the prompt to describe the overall style and aesthetic quality of the image (e.g. 4K, 8K, realistic, etc). However, when generating images in SD with prompts, not all prompts are accurately reflected, so it is still challenging to precisely place the objects.

Control strategy for stable diffusion. To generate images $\mathcal{X}_{\text{gen}}$ that are consistent with both spatial and object categories from the previous dataset $\mathcal{D}_{m-1}$, we employ grounding input in conjunction with the text prompt $\mathbf{T}$. Specifically, for each annotation $\mathbf{y}^i_{m-1}$, we extract the category labels $\mathbf{C}_{\text{label}}$ and their corresponding object locations $\mathbf{B}_{\text{location}}$ for all entities. The grounding input is defined as a set of label and location pairs as follows:

$$\underbrace{\{\mathbf{C}_{\text{label}}, \mathbf{B}_{\text{location}}\}}_{\textit{grounding input}} = \{\underbrace{(c_1, \mathbf{b}_1), (c_2, \mathbf{b}_2), ..., (c_N, \mathbf{b}_N)}_{\textit{entities}}\}. \tag{4}$$

Next, we use the text encoder in CLIP [45] to convert the labels $\mathbf{C}_{\text{label}}$ into text-to-image matching embeddings, the same as we apply to the prompts for the SD. Concurrently, the location $\mathbf{B}_{\text{location}}$ is transformed into the Fourier embeddings as suggested by Mildenhall *et al.* [39] for high-dimension representation. These embeddings are then fused across the feature dimension by the MLP layer, serving as a condition for the SD. The fused grounding embeddings are incorporated into the generation process using a gated self-attention fusion strategy [32] with the text prompt $\mathbf{T}$. We employ them throughout the entire denoising process using $\beta = 1$. Furthermore, we leverage CLIP's image encoder to extract image embedding from the corresponding image $\mathbf{x}^i_{m-1}$ associated with each annotation $\mathbf{y}^i_{m-1}$. The image embedding is replicated N times, corresponding to the number of objects in $\mathbf{y}^i_{m-1}$. The image embeddings are then concatenated with the grounding embeddings across the feature dimension. This process, aligned with the text prompt T, is employed in the generation process. It closely reproduces the realistic style and quality of the original images from the previous dataset $\mathcal{D}_{m-1}$. The image qualities that are varied with additional inputs are shown in Figure 3.

4.2. Iterative class-wise refiner

Despite our efforts to closely mimic the characteristics of real images, training the model with images $\mathcal{X}_{gen}$ generated using all of the previous annotations $\mathcal{Y}_{m-1}$ resulted in only limited performance improvements, while also leading to extended generation times. To address these issues, we employ a class-wise generation limit, denoting the maximum number of generated images

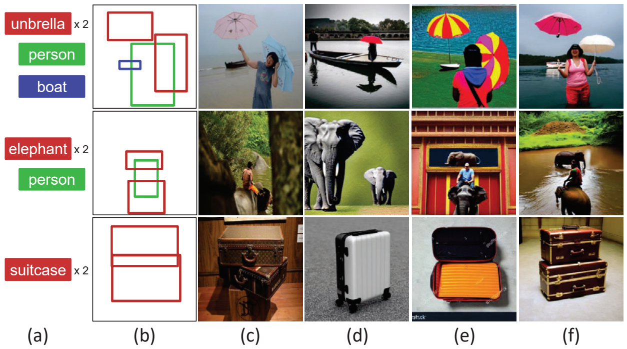

Figure 3. Differences in image generation based on input types. Each row represents different examples used for image synthesis. The first row uses prompts like "A photo of two umbrellas, person and boat, realistic, ... details". The second row uses prompts like "A photo of two elephants and person, ...". The last row uses prompts like "A photo of two suitcases, ...". (a) and (b) show the grounding input. (c) shows COCO real images. (d) depicts the prompt-only synthetic images. (e) depicts combined the grounding input and the prompt. (f) shows used the prompt, grounding input, and CLIP image embedding for image synthesis.

for each class as $\mathcal{N}$. This constraint not only ensures a more efficient, but also a balanced generation process. We further employ a refinement process using the model $\mathcal{M}_{m-1}$ to ensure the quality and fidelity of the generated images. This refinement process is conducted through pseudo-labeling (described in 4.3) where images containing objects with a probability lower than p_{refine} are discarded in $\mathcal{D}_{gen}$. This approach presents a trade-off: a higher threshold p_{refine} results in higher-quality images but often fails to meet the class-wise quantity $\mathcal{N}$, while a lower threshold achieves the quantity goal but compromises image fidelity. Therefore, we adopt an iterative process where the threshold p_{refine} is gradually decreased by 0.05 in each generation cycle. This process continues until the generated images for all classes meet the pre-defined class-wise quantity $\mathcal{N}$ or until the lower-bound threshold p_{refine}, which is 0.4 in our case, is reached.

However, if after the completion of the iterative refinement process, the generated image count for any class still does not meet the standard $\mathcal{N}$, we introduce a class-specific generation strategy. This additional step involves creating an additional generation process with specialized prompts and phrases tailored for the classes that have not generated enough images. To focus the synthesis on the target object, we strategically position the bounding box at the center of normalized coordinates [0.3, 0.3, 0.6, 0.6]. In this manner, we also apply the previously described refinement process, utilizing $\mathcal{M}_{m-1}$ with only lower-bound threshold. Figure 4 shows an example of this class-specific image generation technique.

4.3. Pseudo labeling

Based on the prediction mechanism introduced in [15, 37], D-DETR utilizes learned object queries in the decoder. Each query is processed through multiple decoder layers and then fed into the classification and regression heads to predict classes and bounding boxes, respectively. The classification head computes a matrix $F_{\text{cls}} \in \mathbb{R}^{Q\times(\text{cls}+1)}$, where Q denotes the number of object queries and cls+1 represents the total number of learned classes with an additional one for the background prediction. Each entry in F_{cls} denotes a logit score, signifying the probability of a query being a specific class. For pseudo-labeling, after processing through the final decoder layer, we examine the logits in F_{cls} for each query, identifying the logit with the highest score. If this highest score exceeds the pre-defined threshold p_{pseudo}, the query is then pseudo-labeled with the corresponding class. Concurrently, the regression head outputs a matrix $F_{\text{reg}} \in \mathbb{R}^{Q\times 4}$, where each column contains the normalized coordinates of the bbox for each query. The queries exceeding the threshold in F_{cls} are aligned with the bbox predictions in F_{reg} for the pseudo ground truth. It plays a crucial role in mitigating the forgetting of previously learned objects during the current training phase $\mathcal{T}_m$, particularly by reducing the misclassification of these objects as background, especially in scenarios where previous annotations $\mathcal{Y}_{n-1}$ are not available. This strategy is also used in Section 4.2 to refine the synthetic images.

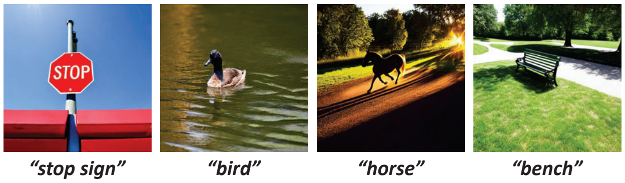

Figure 4. Class-specific image generation. This process utilizes a single class label as a prompt and grounding input with a fixed location. For example, in the first column, we have $\mathbf{T}$ = "stop sign", $\{\mathbf{C}_{\text{label}}, \mathbf{B}_{\text{location}}\}$ ={("stop sign", [0.3, 0.3, 0.6, 0.6])}.

4.4. Training with generated image

While employing synthetic images $\mathcal{X}_{gen}$, we observed that despite our attempts to preserve the previous knowledge, the performance improvement when training directly is insufficient compared to the state-of-the-art. To improve this, instead of using the synthetic images $\mathcal{X}_{gen}$ as direct inputs for training, we enforce the new model $\mathcal{M}_m$ to acquire previous knowledge indirectly from the previous model $\mathcal{M}_{m-1}$. Inspired by [11], we apply an L2 distillation loss to both the classification and regression outputs. The formulation of the distillation function in terms of the object queries Q and at a given task index m is defined as follows:

$$\mathcal{L}_{\text{cls}} = \frac{1}{Q\times C}\sum_{i=1}^{Q}\sum_{j=1}^{C}\left(F^{ij}_{\text{cls},m} - F^{ij}_{\text{cls},(m-1)}\right)^2, \tag{5}$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{Q\times 4}\sum_{i=1}^{Q}\sum_{k=1}^{4}\left(F^{ik}_{\text{reg},m} - F^{ik}_{\text{reg},(m-1)}\right)^2, \tag{6}$$

where $F^{ij}_{\text{cls},m}$ and $F^{ik}_{\text{reg},m}$ represent the predicted scores for the i-th query's j-th class index and k-th bbox coordinate by the new model $\mathcal{M}_m$, and $F^{ij}_{\text{cls},(m-1)}$ and $F^{ik}_{\text{reg},(m-1)}$ are those predicted by the old model $\mathcal{M}_{m-1}$. This approach retains the predictive consistency of $\mathcal{M}_{m-1}$, thereby mitigating the forgetting. Furthermore, since the decoder in D-DETR extracts predictions over 6 layers, we extend the application of L2 distillation loss across all these layers to facilitate distillation. In the training, we use the same loss formulation for D-DETR, denoted as $\mathcal{L}_{DETR}$. To effectively integrate and balance the distillation loss with the inherent D-DETR loss, we introduce a weight λ. The final loss function, reflecting a blend of the standard losses with the additional distillation components, is formalized as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{DETR} + \lambda(\alpha\mathcal{L}_{\text{cls}} + \beta\mathcal{L}_{\text{reg}}). \tag{7}$$

Here, α and β are the weights for the classification and regression loss terms, respectively, adapted from the original D-DETR configuration, where α is 2 and β is 5.

5. Experiments

5.1. Dataset and metrics

Our research utilizes the MS COCO 2017 [34], which consists of 80 diverse classes across 118,000 images for training and 5,000 images for evaluation. These classes are strategically divided based on our experiment scenario. For evaluation, we employ standard COCO metrics, including mean average precision (mAP, %) at different intersection over union (IoU) thresholds and object sizes: AP, $AP_{.5}$, $AP_{.75}$, AP_S, AP_M, and AP_L. Here, AP refers to the mAP calculated over IOU thresholds ranging from 0.5 to 0.95. In our ablation study, we introduce the *forgetting percentage points* (FPP) as proposed by CL-DETR [37], as a metric to evaluate the degree of forgetting for trained categories.

5.2. Implementation and experiments

Implementation details. Our method is based on deformable-DETR [71], which leverages pre-trained ResNet-50 [18] as a multi-scale backbone. We set the number of object queries Q to 300 while keeping all other settings consistent with our baseline [71]. All experiments are performed using NVIDIA A100 GPUs with a batch size of 8. In our generation process, we utilize stable diffusion version 1.4. For incorporating grounding input, we employ pre-trained GLIGEN's gated self-attention weights that have been trained on various datasets including GoldG [30], O365 [55], SBU [42], and CC3M [56]. Additionally, we set the classifier-free guidance scale to 7.5.

Scenario setup. In our experiment, we focus on two scenarios: the *two-phase setting* and the *multiple-phase setting*. In the *two-phase setting*, we train a model first task on $\mathcal{T}_1$ and then on a different task $\mathcal{T}_2$, evaluating on a combined total of $\mathcal{T}_1+\mathcal{T}_2$ classes, such as 40+40 or 70+10. In the *multiple-phase setting*, we begin by training on 40 classes as a $\mathcal{T}_1$, then sequentially add new classes in $\mathcal{T}_n$ phases (e.g., 40+20+20 or 40+10+10+10+10), with evaluation conducted on all classes $\mathcal{T}_{1:n}$ learned up to each phase.

5.3. Results

Two-phase setting. Tab. 1 shows that our method outperforms previous approaches such as LWF [33], RILOD [28], SID [43], and ERD [11] using GFLv1 [31], including CL-DETR [37]. Importantly, we achieved a 0.5% increase in AP for the 70+10 scenario and 1.0% for the 40+40 scenario. Moreover, we observed even higher gains of 1.5% and 2.0% in $AP_{.5}$, respectively. It is particularly noteworthy that while CL-DETR relies on a 10% replay buffer comprising real data from previous tasks, our method stands out by achieving remarkable performance improvements without any reliance on real previous data.

Multi-phase setting. Tab. 2 shows that our method, which utilizes synthetic image-based training, surprisingly outperforms other approaches significantly in multi-phase scenarios. Despite using different baselines like [11, 28, 43], it is evident that our method maintains consistent performance. We achieve 8.7% and 5.8% gains in AP for the 40+10+10+10+10 and 40+20+20 scenarios, respectively, compared to CL-DETR. This highlights that our approach, which uniformly employs synthetic data through knowledge distillation from the old model across all phases, effectively trains on new task data while maintaining high performance by alleviating catastrophic forgetting.

5.4. Ablations

Main components. In Tab. 3, we present an ablation study of our method's components in the 70+10 scenario. For the 'Fine-tuning' component, we do not apply specific CIOD strategies. The results show a significant improvement when employing the pseudo-labeling strategy, which notably reduces FPP by 39.1% in AP. This highlights its crucial role in minimizing the misclassification of previously trained objects as background. Following the introduction of a synthetic dataset to mitigate forgetting, we noticed a modest increase in AP by 1.2% in all categories and a reduction in FPP by 1.5%. Although this indicates that synthetic data contributes to knowledge retention, its impact on reaching state-of-the-art performance is somewhat insufficient. However, the results take a significant turn when we integrate distillation with the synthetic dataset training, marking a substantial improvement. As a result, we achieve an AP of 40.9% in all categories and 41.5% in old categories, along with an FPP of 1.9%. These indicate a significant advancement in the effectiveness of our method.

Pseudo-labeling. Tab. 4 illustrates the impact of varying confidence score p_{pseudo} thresholds on the selection of optimal queries for pseudo ground-truth labeling. The data reveals a marked improvement in performance when predictions are labeled using a query score threshold above 0.3. However, it also shows a gradual decline in performance as the threshold is increased beyond this point.

Table 1. CIOD results (%) on COCO 2017 in *two-phase setting*. The results of related research [11, 28, 33, 43] extract from CL-DETR paper. The order of data follows the [11]. The best performance is highlighted in **bold**, and a red upward arrow ↑ signifies an improvement in performance relative to the state-of-the-art.

Scenarios	Method	Baseline	AP	$AP_{.5}$	$AP_{.75}$	AP_S	AP_M	AP_L
40 + 40	LWF [33]	GFLv1	17.2	25.4	18.6	7.9	18.4	24.3
	RILOD [28]	GFLv1	29.9	45.0	32.0	15.8	33.0	40.5
	SID [43]	GFLv1	34.0	51.4	36.3	18.4	38.4	44.9
	ERD [11]	GFLv1	36.9	54.5	39.6	21.3	40.4	47.5
	CL-DETR [37]	Deformable DETR	42.0	60.1	45.9	24.0	45.3	55.6
	Ours	Deformable DETR	**43.0** ↑1.0	**62.1** ↑2.0	**47.1** ↑1.2	**24.9** ↑0.9	**46.9** ↑1.6	**57.0** ↑1.4
70 + 10	LWF [33]	GFLv1	7.1	12.4	7.0	4.8	9.5	10.0
	RILOD [28]	GFLv1	24.5	37.9	25.7	14.2	27.4	33.5
	SID [43]	GFLv1	32.8	49.0	35.0	17.1	36.9	44.5
	ERD [11]	GFLv1	34.9	51.9	37.4	18.7	38.8	45.5
	CL-DETR [37]	Deformable DETR	40.4	58.0	43.9	23.8	43.6	53.5
	Ours	Deformable DETR	**40.9** ↑0.5	**59.5** ↑1.5	**44.8** ↑0.9	**23.9** ↑0.1	**44.7** ↑1.1	**54.0** ↑0.5

Table 2. CIOD results (AP/$AP_{.5}$, %) on COCO 2017 in *multi-phase setting*. The tasks are divided into two scenarios: 40+10+10+10+10 and 40+20+20. Ours and CL-DETR are based on deformable DETR, while ERD, RILOD, and SID are based on GFLv1. A red upward arrow ↑ indicates a performance improvement compared to the state-of-the-art CL-DETR. The "-" symbol indicates a missing value, as reported in paper [37].

Method	$\mathcal{T}_1$ (1-40)	40+10+10+10+10				40+20+20	
		$\mathcal{T}_2$ (40-50)	$\mathcal{T}_3$ (50-60)	$\mathcal{T}_4$ (60-70)	$\mathcal{T}_5$ (70-80)	$\mathcal{T}_2$ (40-60)	$\mathcal{T}_3$ (60-80)
Ours	46.5 / 68.6	42.3 / 62.8	40.6 / 60.2	40.0 / 59.0	**36.8** ↑8.7 / 54.7	42.5 / 62.2	**41.1**↑5.8 / 59.5
CL-DETR [37]		-	-	-	28.1 / -	-	35.3 / -
ERD [11]	45.7 / 66.3	36.4 / 53.9	30.8 / 46.7	26.2 / 39.9	20.7 / 31.8	36.7 / 54.6	32.4 / 48.6
RILOD [28]		25.4 / 38.9	11.2 / 17.3	10.5 / 15.6	8.4 / 12.5	27.8 / 42.8	15.8 / 4.0
SID [43]		34.6 / 52.1	24.1 / 38.0	14.6 / 23.0	12.6 / 23.3	34.0 / 51.8	23.8 / 36.5

Refiner. Tab. 5 presents our findings on how varying the number of generated images per class ($\mathcal{N}$) and the refinement threshold (p_{refine}) influences the performance of our iterative refinement method (Section 4.2).

When examining different $\mathcal{N}$ values (50, 100, and 200), we observed comparable performances, particularly with a fixed threshold range (e.g., from 0.8 to 0.4). This suggests that our method is robust to variations in class-wise image count regulation. However, when we set $\mathcal{N}$ to "no limit" (∞), generating images based on all old annotations without class-wise restrictions, there is a performance drop of 1% (39.9%) in AP compared to our best (40.9%). This outcome highlights the necessity of regulating image generation for each class. Furthermore, by limiting the production quantity per class using $\mathcal{N}$, we significantly reduce the time required for generation. This efficiency gain is further discussed in the supplementary Tab. 2.

Regarding the refinement threshold (p_{refine}), our best result (40.9%) is obtained with a dynamic range between 0.4 and 0.8. Setting p_{refine} to a fixed value, either at the low end (0.4) or high end (0.8), led to diminished performance. This indicates the importance of a dynamic threshold range in optimizing the generation and refinement process. In conclusion, these results demonstrate that our iterative refinement strategy effectively refines the synthetic images while balancing the quality and quantity of the synthetic images.

Weight parameter λ. In Tab. 6, we examine the effect of different weight parameter λ. We found that a weight of 2 achieved the best performance with an AP of 40.9%. Importantly, all tested weights performed better than the current state-of-the-art performance of 40.4%, demonstrating the effectiveness of our knowledge distillation approach using synthetic images.

CLIP image embedding. In Tab 7, we conducted an ablation experiment to evaluate the effect of incorporating CLIP's image embedding in the generation process (Sec. 4.1). The result indicates that incorporating CLIP's image embedding led to a performance improvement of 1.2% in AP. This highlights the impact of CLIP's image embedding in enhancing the realism of synthetic images, which in turn positively impacts the detector performance. On the other hand, without the CLIP's image embedding, it results in an inferior performance that is similar to the baseline using pseudo-labeling alone (38.6% AP in Tab. 3). This result implies that image quality and realism are important

Table 3. Ablation study of main contribution components on COCO 2017 (*two-phase setting*, 70+10). The metrics assess performance after completing training across all phases, measuring results across all categories (higher is better) and specifically in old categories (higher is better). The *forgetting percentage point* (FPP, lower is better) specifically reflects the performance change in the initial 70 categories, as measured by the difference in AP between the first phase and the last phase. The best performance is represented in **bold**, with the final row indicating our method's results.

Method	All categories ↑			Old categories ↑			FPP ↓		
	AP	$AP_{.5}$	$AP_{.75}$	AP	$AP_{.5}$	$AP_{.75}$	AP	$AP_{.5}$	$AP_{.75}$
Fine-tuning	14.8	23.6	15.6	0.0	0.0	0.0	43.4	62.8	47.2
+ Pseudo labeling	38.6	56.2	42.1	39.1	57.3	42.7	4.3	5.5	4.5
++ Deep generative replay	39.8	57.7	43.4	40.6	59.2	44.0	2.8	3.6	3.2
+++ Knowledge distillation	**40.9**	**59.5**	**44.8**	**41.5**	**60.6**	**45.4**	**1.9**	**2.2**	**1.8**

Table 4. Ablation study of the range of confidence scores in the pseudo-labeling strategy on COCO 2017 (70+10). The best performance is highlighted in **bold**.

Setting	AP	$AP_{.5}$	$AP_{.75}$	AP_S	AP_M	AP_L
$p_{\text{pseudo}} \geq 0.2$	29.5	44.9	32.0	17.4	33.4	38.8
$p_{\text{pseudo}} \geq 0.3$	**38.6**	**56.2**	**42.1**	**22.3**	**42.1**	**50.6**
$p_{\text{pseudo}} \geq 0.4$	37.5	54.2	40.8	22.0	41.3	48.5
$p_{\text{pseudo}} \geq 0.5$	35.2	51.1	38.8	20.3	38.7	46.1

Table 5. Ablation study on image generation regulation and refinement confidence score thresholds on COCO 2017 (70+10). The best result is highlighted in **bold** among each ablation.

Setting	AP	$AP_{.5}$	$AP_{.75}$	AP_S	AP_M	AP_L
$\mathcal{N}=50$	**40.9**	**59.5**	**44.8**	**24.0**	**44.7**	54.0
$\mathcal{N}=100$	40.7	59.4	44.6	**24.0**	44.4	53.9
$\mathcal{N}=200$	40.8	**59.5**	44.6	**24.0**	44.2	**54.5**
$\mathcal{N}=\infty$	39.9	58.6	43.4	23.0	43.6	53.1
$p_{\text{refine}}=0.4$	39.8	58.8	43.4	23.1	43.2	52.6
$p_{\text{refine}}=0.8$	38.5	57.2	42.2	22.9	41.8	51.0
$p_{\text{refine}} \in [0.4,0.8]$	**40.9**	**59.5**	**44.8**	23.9	**44.7**	**54.0**
$p_{\text{refine}} \in [0.5,0.8]$	40.7	59.3	44.5	**24.1**	44.0	53.6
$p_{\text{refine}} \in [0.6,0.8]$	40.5	59.2	44.3	23.4	44.2	53.6
$p_{\text{refine}} \in [0.7,0.8]$	39.6	56.8	43.3	22.5	42.6	51.8

Table 6. Ablation study of knowledge distillation weight on COCO 2017 (70+10). The best result is highlighted in **bold**.

Weight	AP	$AP_{.5}$	$AP_{.75}$	AP_S	AP_M	AP_L
$\lambda=1$	40.6	59.2	44.2	23. 7	44.1	53.8
$\lambda=2$	**40.9**	**59.5**	**44.8**	**23.9**	**44.7**	**54.0**
$\lambda=3$	40.5	59.3	44.4	24.0	44.1	53.0

in CIOD with the deep generative model to effectively prevent the catastrophic forgetting.

Table 7. Ablation study of CLIP's image embedding on COCO 2017 (70+10). The experiment was conducted excluding L2 knowledge distillation to evaluate the effect of synthetic data. The best result is highlighted in **bold**. A red upward arrow ↑ indicates the performance improvement.

Setting	AP	$AP_{.5}$	$AP_{.75}$
w/o Image embedding	38.6	56.9	42.1
w/ Image embedding	**39.8** 1.2↑	**57.7** 0.8↑	**43.4** 1.3↑

6. Conclusions

In this paper, we introduced the SDDGR strategy, a novel diffusion-based deep generative replay approach for class incremental object detection. The proposed SDDGR includes a method for generating synthetic images that encompass objects from previously trained classes, with the goal of enhancing their quality and high fidelity while maintaining computational efficiency. To achieve this, we suggested a rigorous refinement technique and class-wise regulation of quantity. Additionally, the synthetic images are effectively used to mitigate forgetting through the application of L2 knowledge distillation. Finally, SDDGR utilizes an effective pseudo-labeling technique that substantially reduces the misclassification of objects as background. The combination of these proposed methods enables our SDDGR to achieve state-of-the-art performance in class incremental object detection.

Acknowledgements. This work was supported by IITP grants (No. 2020-0-01336 Artificial intelligence graduate school program (UNIST) 10%; No. 2021-0-02068 AI innovation hub 10%; No. 2022-0-00264 Comprehensive video understanding and generation with knowledge-based deep logic neural network 10%) and the NRF grant (No. RS-2023-00252630 10%), all funded by the Korean government (MSIT). This work was also supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by Ministry of Oceans and Fisheries (RS-2022-KS221674) 20% and received support from LG Electronics (20%) and AI Center, CJ Corporation (20%).

References

[1] Manoj Acharya, Tyler L Hayes, and Christopher Kanan. Rodeo: Replay for online object detection. *BMVC*, 2020. 1, 2

[2] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, 2021. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 3

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018. 2

[5] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. *arXiv*, 2019. 1, 2

[6] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. Gan memory with no forgetting. *NeurIPS*, 2020. 1, 2

[7] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *ICCV*, 2021. 1, 2

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018. 3

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 2021. 2, 3

[10] Na Dong, Yongqiang Zhang, Mingli Ding, and Gim Hee Lee. Incremental-detr: Incremental few-shot object detection via self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 543–551, 2023. 2

[11] Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *CVPR*, 2022. 1, 2, 4, 5, 6, 7

[12] Rui Gao and Weiwei Liu. Ddgr: Continual learning with deep diffusion-based generative replay. *ICML*, 2023. 1, 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 2, 3

[14] Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Improved schemes for episodic memory-based lifelong learning. In *NIPS*, 2020. 2

[15] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *CVPR*, 2022. 2, 5

[16] Yu Hao, Yanwei Fu, Yu-Gang Jiang, and Qi Tian. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *ICME*, 2019. 1, 2

[17] Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *BMVC*, 2018. 1, 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv*, 2022. 3

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2, 3

[21] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023. 3

[22] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *AAAI*, 2018. 2

[23] Quentin Jodelet, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. Class-incremental learning using diffusion model for distillation and replay. In *ICCVW*, 2023. 2

[24] Junsu Kim, Sumin Hong, Chanwoo Kim, Jihyeon Kim, Yihalem Yimolal Tiruneh, Jeongwan On, Jihyun Song, Sunhwa Choi, and Seungryul Baek. Class-wise buffer management for incremental object detection: An effective buffer training strategy. In *ICASSP*, 2024. 2

[25] Junsu Kim, Yunhoe Ku, Jihyeon Kim, Junuk Cha, and Seungryul Baek. Vlm-pl: Advanced pseudo labeling approach class incremental object detection with vision-language model. *arXiv*, 2024. 2

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2013. 2, 3

[27] Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. *arXiv*, 2021. 2

[28] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. Rilod: Near real-time incremental learning for object detection at the edge. In *SEC*, 2019. 2, 6, 7

[29] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022. 2

[30] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 6

[31] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *NIPS*, 2020. 2, 6

[32] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *CVPR*, 2023. 2, 3, 4

[33] Zhizhong Li and Derek Hoiem. Learning without forgetting. *ECCV*, 2016. 1, 2, 6, 7

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[35] Liyang Liu, Zhanghui Kuang, Yimin Chen, Jing-Hao Xue, Wenming Yang, and Wayne Zhang. Incdet: In defense of elastic weight consolidation for incremental object detection. *TNNLS*, 2020. 1

[36] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Multi-task incremental learning for object detection. *arXiv*, 2020. 2

[37] Yaoyao Liu, Bernt Schiele, Andrea Vedaldi, and Christian Rupprecht. Continual detection transformer for incremental object detection. In *CVPR*, 2023. 2, 5, 6, 7

[38] Yichen Lu, Mei Wang, and Weihong Deng. Augmented geometric distillation for data-free incremental person reid. In *CVPR*, 2022. 1, 2

[39] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2021. 4

[40] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv*, 2014. 2

[41] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 3

[42] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *NIPS*, 2011. 6

[43] Can Peng, Kun Zhao, Sam Maksoud, Meng Li, and Brian C Lovell. Sid: Incremental learning for anchor-free object detection via selective and inter-related distillation. *CVIU*, 2021. 2, 6, 7

[44] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 2

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 4

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[47] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 2

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[49] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 1995. 2

[50] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *NIPS*, 2019. 2

[51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 3

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3

[53] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv*, 2021. 2, 3

[54] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 2022. 2, 3

[55] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019. 6

[56] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018. 6

[57] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NIPS*, 2017. 1, 2

[58] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *NIPS*, 2017. 1, 2

[59] Jordan Shipard, Arnold Wiliem, Kien Nguyen Thanh, Wei Xiang, and Clinton Fookes. Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 769–778, 2023. 3

[60] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, 2017. 1, 2, 4

[61] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *CVPR*, 2021. 1, 2

[62] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv*, 2020. 2, 3

[63] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv*, 2020. 2, 3

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[65] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. In *SIGGRAPH*, 2023. 2, 3

[66] Chenshen Wu, Luis Herranz, Xialei Liu, Joost Van De Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *NIPS*, 31, 2018. 1, 2

[67] Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. Incremental learning using conditional adversarial networks. In *NeurIPS*, 2019. 1, 2

[68] Jie Yang, Bingliang Li, Fengyu Yang, Ailing Zeng, Lei Zhang, and Ruimao Zhang. Boosting human-object interaction detection with text-to-image diffusion model. *arXiv preprint arXiv:2305.12252*, 2023. 3

[69] Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, et al. Reco: Region-controlled text-to-image generation. In *CVPR*, 2023. 2

[70] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 3

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2020. 2, 6