As Pseudo-label Free As Possible: Leveraging Adaptive Feature Generation for Sparsely Annotated Object Detection

Shuilian Yao [1], Yu Liu [1], Qi Jia [1*], Sihong Chen [2], Wei Zhuo [3]

[1]School of Software Technology, DUT-RU International School of Information Science & Engineering, Dalian University of Technology, Dalian, China
[2]AILab, Tencent, Shenzhen, China
[3]National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, China
shuilian_yao@mail.dlut.edu.cn, {liuyu8824,jiaqi}@dlut.edu.cn, whalechen@tencent.com, weizhuo@szu.edu.cn

Abstract

Compared to fully supervised object detection, training with sparse annotations typically leads to a decline in performance due to insufficient feature diversity. Existing sparsely annotated object detection (SAOD) methods often rely on pseudo-labeling strategies, but these pseudo-labels tend to introduce noise under extreme sparsity. To simultaneously avoid the impact of pseudo-label noise and enhance feature diversity, we propose a novel Adaptive Feature Generation (AdaptFG) model that generates features based on class names. This model integrates a pre-trained CLIP into a VAE-based feature generator, with its core innovation being an Adaptor that adaptively maps CLIP semantic embeddings to the object detector domain. Additionally, we introduce inter-class relationship reasoning in detector, which effectively mitigates misclassifications stemming from similar features. Extensive experimental results demonstrate that AdaptFG consistently outperforms state-of-the-art SAOD methods on the PASCAL VOC and MS COCO benchmarks.

Code — https://github.com/YAOSL98/AdaptFG

Introduction

In contrast to traditional object detection approaches (Lin et al. 2017; Ren et al. 2015; Tan, Pang, and Le 2020; Wang, Bochkovskiy, and Liao 2023), which generally require a substantial number of labeled instances for each class to achieve high performance, sparsely annotated object detection (SAOD) offers a remedy that makes a detector learn with a limited number of labeled samples. Researching SAOD is particularly significant because acquiring fully annotated samples in practical applications is not only costly but also labor-intensive. However, training such a detector becomes challenging, as limited class labels and bounding boxes hinder the network from acquiring sufficient target features for training, as shown in Figure1(a).

To address this challenge, several seminal works, such as Unbiased Teacher (Niitani et al. 2019) and Co-mining (Wang et al. 2021), typically employ pseudo-labeling to explore objects misclassified as background. However, the generated pseudo-labels often plagued by substantial noise, evident from the red corsses in Figure1(b).

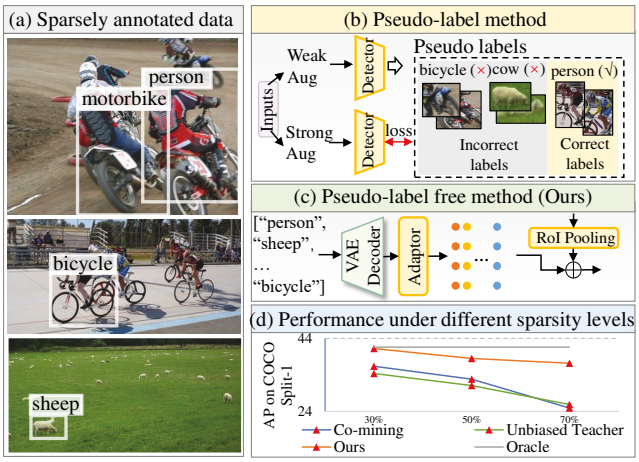

Figure 1: Comparison of the pseudo-label method with our approach. (a) Three figures exhibit typical examples of sparsely annotated data, in which only a few objects are labeled. (b) The pseudo-label method mines more proposals as foreground, but it is prone to producing incorrect predictions under extreme sparsity. (c) Our pseudo-label free method innovatively integrates a CLIP-infused VAE and an Adapter, alongside leveraging target feature correlations, to address the challenges posed by limited annotations. (d) AP performance on COCO under different sparsity levels. The horizontal axis represents the sparsity, i.e., the percentage of annotations removed. Our method demonstrates greater robustness compared to other methods, with performance close to that of the fully annotated scenario (Oracle).

Figure 1 demonstrates a 70% reduction in the number of labels (indicating increased sparsity along the horizontal axis) significantly degrades performance, underscoring the unresolved issue of sparsity sensitivity.

The sparsity sensitivity primarily arises from the detector's exclusive dependence on the visual information throughout the learning phase. Firstly, the scarcity of labeled data inherently restricts the diversity of features, inadequately fulfilling the detector's learning prerequisites. While pseudo-labeling techniques can augment the number of foreground proposals, enhancing feature diversity, they

*Corresponding author.

inherently risk introducing noise in scenarios of extreme sparsity. Secondly, the detector's exclusive reliance on visual information overlooks the potential benefits of leveraging inter-class relationships. For example, if inter-class relationships reveal that 'sheep' and 'cow' share similarities, yet they belong to distinct categories, this prior knowledge can significantly enhance the learning process for sparsely annotated 'sheep' examples. By leveraging this information, we can effectively minimize misclassifications of sparsely annotated features, ensuring more accurate recognition. Therefore, a critical question arises: *can we utilize a generator to adaptively synthesize diverse features within the feature space of annotated proposals while incorporating inter-class relationship reasoning?*

To address this issue, we propose a pseudo-label free method, namely Adaptive Feature Generation (AdaptFG), for sparsely annotated object detection, as shown in Figure 1(c). Through our feature adaptation process, it facilitates the generation of diverse features, aligns them with the corresponding real features of the detector in the feature space, and participates in detector training and feature reasoning. Specifically, AdaptFG utilizes a CLIP-integrated Variational Autoencoder (VAE) framework to generate visual features tailored to any given class names. Moreover, we design an adaptor to harmoniously align these synthetic features with the Region of Interest (RoI) feature space from labeled instances. Additionally, recognizing the significance of inter-class relationships within these synthetic features, we integrate class-relation reasoning to bolster the precision of classification predictions. Extensive experiments on MS COCO (Lin et al. 2014) and PASCAL VOC (Everingham et al. 2015) validate the effectiveness of AdaptFG.

Our main contributions can be summarized as follows:

- Our work is the first to address SAOD via a feature generation model that incorporates a novel CLIP-infused VAE Generator and an RoI Feature Adaptor, significantly augmenting feature diversity and enhances overall performance via a pseudo-label free fashion.
- We propose a class-relation reasoning module, inferring inter-class relationships as prior to mitigate the label-scarce challenge in SAOD.
- Our AdaptFG approach demonstrates substantial enhancements over the baseline, and consistently achieves state-of-the-art performance across two benchmarks.

Related Work

Sparsely Annotated Object Detection (SAOD) aims to train detectors in more realistic scenarios where each training image may contain unlabeled instances. One of the pioneering works, (Wu et al. 2018) suggests a re-weighting approach that down-weights negative samples based on their Intersection over Union (IoU) with existing labels to mitigate the impact of incorrect annotations. (Niitani et al. 2019) introduces part-aware sampling and a pseudo-label-guided sampling strategy to improve detection performance. Another approach, (Yoon, Hong, and Choi 2021) proposes a semi-supervised learning framework that leverages tracking algorithms to generate pseudo-labels for unlabeled data, thereby enhancing object detection performance. In a similar vein, (Zhang et al. 2020) introduces a background recalibration loss strategy for single-stage detectors, which treats unlabeled regions as easy positives to avoid generating large error signals. Co-mining (Wang et al. 2021) employs a Siamese network to predict pseudo-label sets for each other, while SparseDet (Suri et al. 2023) utilizes a dual-stream network for self-supervised learning for unlabeled regions. However, when annotations are highly sparse, these operations on pseudo-labels or background regions result in substantial noise, complicating the learning process.

Feature Generation has become a commonly employed technique for various low-shot learning tasks (Xu and Le 2022; Guirguis et al. 2023; Xu, Le, and Samaras 2023; Zhu et al. 2021), with the overarching aim of generating diverse and reliable features. (Xu and Le 2022) proposes generating representative samples using a variational autoencoder (VAE) model conditioned on the semantic embedding of each class. (Zhang and Wang 2021) suggests increasing data variance for novel classes by transferring shared within-class variation from base classes. (Guirguis et al. 2023) leverages the statistics of RoI features from the base model to create base instance-level features without accessing base images. Norm-VAE (Xu, Le, and Samaras 2023) develops the first CLIP-based VAE model, capable of generating features with increased diversity related to cropping. (Tang et al. 2024) aligns the virtual image features from CLIP with textual features and trains a conditional generative model to produce new class features, resulting in improved classification performance. However, **none** of these works explore or provide a general solution for addressing synthetic feature shifts in downstream tasks. Moreover, feature generation methods have yet to be explored in the context of SAOD. Therefore, we propose a novel VAE-based adaptive feature generation method for SAOD, incorporating relational reasoning to ensure that synthetic features are more effectively aligned with the target distribution.

Approach

Overview. Figure 2 summarizes the overall scheme of the proposed AdaptFG. The left column details the process starting with training a CLIP-infused VAE feature generator to produce synthetic features, followed by training an Adaptor to map these features into the detector space. The right column presents the generation of class-balanced synthetic features using the trained VAE generator and Adaptor based on class names c, integrated into the detector's training with real RoI features, and enhanced by prototypical metric learning and class-relation reasoning to optimize feature representations. Additionally, we employ a pseudo-label free baseline, namely SparseDet (Suri et al. 2023), for discovering unlabeled regions, where proposals generated by the RPN with objectness scores exceeding a threshold τ_{obj} and an IoU below τ_{fg} relative to any ground truth are identified as unlabeled regions. A self-supervised constraint is then applied using L_{SSL} to ensure that the high-confidence proposals extracted by the detector for both the original and augmented images are similar.

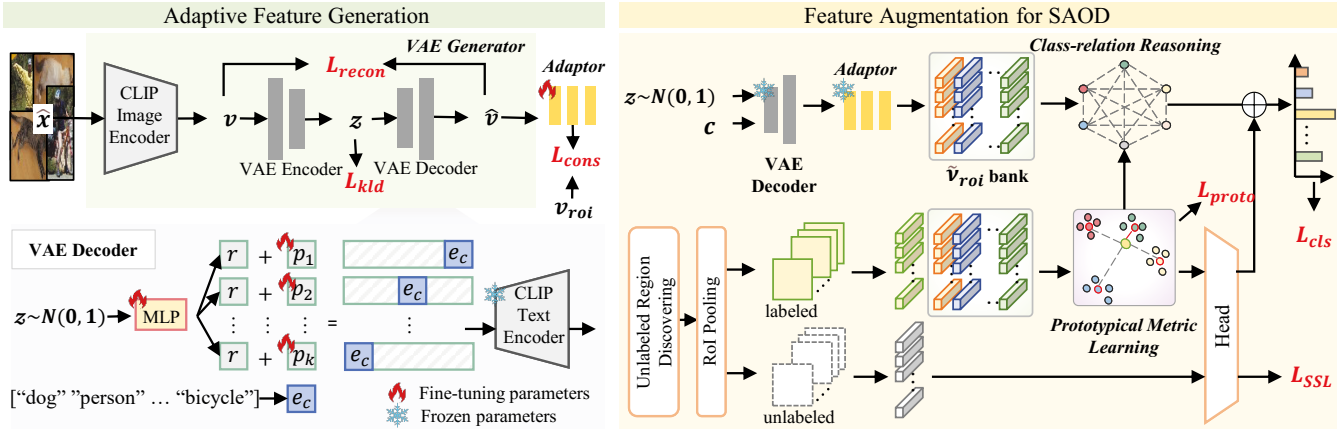

Figure 2: Overview of our AdaptFG for sparsely annotated object detection. *Left:* VAE generator and Adaptor are trained on the pre-trained CLIP to map features into the detector space. *Right:* Class-balanced synthetic features $\tilde{v}_{roi}$ bank is generated through class names c, and the output from the class-relation reasoning module is integrated with the classification layer outputs from the detector head.

Adaptive Feature Generation

Since a pre-trained CLIP (Radford et al. 2021) has aligned visual and semantic embeddings in a latent space, it can be used to guide feature generation in low-shot learning tasks effectively. However, this unmodified guidance overlooks the feature representation in the target domain, leading to significant distribution shifts between real and synthetic features. To address this, we design a VAE Generator and an Adaptor, enabling the CLIP semantic embeddings to better adapt to the feature distributions of the detector.

CLIP-infused VAE Generator First of all, we employ the labeled samples for learning a CLIP-infused VAE Generator. As shown on the left of Fig. 2, the VAE Generator $Gen(\cdot)$ consists of a CLIP image encoder $\mathcal{I}(\cdot)$, a VAE encoder $Enc(\cdot)$ and a VAE decoder $Dec(\cdot,\cdot)$. To eliminate background interference and obtain accurate region for each instance $(x, y) \in D$, where D denotes the labeled training set, we crop n object instances from each image x, based on their ground-truth annotations $y = \{(b_i, c_i)\}_{i=1}^{n}$, with b_i representing the bounding box annotation and $c_i \in C$ representing the ground-truth class label, instead of training with the whole images. The resulting objects are denoted with $\hat{x} = \{(\hat{x}_i, c_i)\}_{i=1}^{n}$, and the following omits the index i for simplicity. For each object instance $\hat{x}$, we first derive its visual feature $v = \mathcal{I}(\hat{x})$ from the CLIP image encoder. Afterward, $Enc(v)$ encodes the CLIP visual feature v into a latent code z, and $Dec(z, c)$ then reconstructs v using the latent code z and the corresponding class name c.

Furthermore, we plug the CLIP text encoder $\mathcal{T}(\cdot)$ into the VAE decoder, so as to obtain the feature $\hat{v}$. As it is impractical to fine-tune the parameters in the CLIP text encoder, the challenge becomes how to minimize the reconstruction loss between v and $\hat{v}$. Drawing inspiration from recent work (Zhou et al. 2022; Wang et al. 2023), we construct learnable prompts conditioned on the latent code z, to achieve the aim of adjusting the feature $\hat{v}$ achieved by the CLIP text encoder. Specifically, given the latent code z, we generate a local bias r by $r = h(z)$, where $h(\cdot)$ is a fully-connected layer projecting z into the class token embedding space. Next, we add r to the prompts as follows

$$p(z) = [p_1 + r, p_2 + r, ..., p_k + r], \tag{1}$$

where $\{p_i, i = 1, 2, ..., k\}$ is a set of learnable prompts which are randomly initialized. Then, we extract the class token embedding e_c according to the class name c and concatenate it behind the prompts $p(z)$, resulting in the final prompt description $t = \{p(z), e_c\}$. Finally, we obtain the $\hat{v} = \mathcal{T}(t)$, where $\mathcal{T}(\cdot)$ is the fixed CLIP text encoder. At last, the loss function of optimizing Generator is written by

$$\begin{aligned} \mathcal{L}_{vae} &= \mathcal{L}_{recon} + \mathcal{L}_{kld} \\ &= \mathbb{E}[-\log Dec(z, c)] + \mathrm{KL}(Enc(v)||p(z|c)), \end{aligned} \tag{2}$$

where $\mathcal{L}_{recon}$ is reconstruction loss, $\mathcal{L}_{kld}$ represents the Kullback-Leibler divergence, and $p(z|c)$ is a prior distribution that is assumed to be $N(0, 1)$.

RoI Feature Adaptor After CLIP-infused VAE generator, we further synthesize RoI features suitable for the object detector. To make it, we build an Adaptor $Adt(\cdot)$ following VAE generator. The Adaptor contains a three-layer MLP, aiming to embed the feature $\hat{v}$ into the RoI feature space by $\tilde{v} = Adt(\hat{v})$, where $\tilde{v}$ denotes the final synthetic feature. Note that, it is feasible to gain the real RoI feature v_{roi} via feeding each object instance into the trained detector $\Phi_{Det}(\cdot)$. Therefore, we optimize a feature consistency loss between the synthetic and real RoI features via

$$\arg\min_{Adt} \mathcal{L}_{cons} = \mathrm{MSE}(\tilde{v}, v_{roi}), \tag{3}$$

where MSE estimates the mean square error.

VAE generator and Adaptor can be optimized jointly, so the total loss cost is formulated by

$$\arg\min_{Enc(\cdot),h,\{p_i\}_{i=1}^{k},Adt(\cdot)} \mathcal{L}_{total} = \mathcal{L}_{vae} + \mathcal{L}_{cons}. \tag{4}$$

Feature Augmentation for SAOD

Upon completing the training of the CLIP-infused VAE Generator and Adaptor, the fixed VAE decoder and Adaptor generate synthetic features for the learning and optimization of the SAOD detector.

Sythetic Feature Integration Generating additional synthetic features is essential to alleviate the feature scarcity during SAOD.

Given a class name c and a noise $z \sim N(0,1)$ sampled from the prior distribution, the VAE decoder $Dec(c,z)$ and Adaptor $Adt(\cdot)$ are sequentially utilized to generate a new RoI feature which is denoted by

$$\tilde{v}_{roi} = Adt(Dec(c,z)). \tag{5}$$

When varying the noise z, each class can produce a number of different RoI features. The resulting set of class-balanced synthetic features $\tilde{v}_{roi}$ bank is then merged with the set of real RoI features v_{roi}, to continue training the detector $\Phi_{Det}(\cdot)$. Apart from the real RoI features, it is necessary to calculate a classification loss for the synthetic features. Hence, we rewrite the loss $\mathcal{L}_{cls}$ during training with

$$\mathcal{L}_{cls} = -\sum c \cdot \sigma(\Phi_{cls}(v_{roi})) - \sum c \cdot \sigma(\Phi_{cls}(\tilde{v}_{roi})), \tag{6}$$

where σ is Softmax function.

Prototypical Metric Learning The standard classification loss in object detector exhibits limitations in enhancing inter-class separability due to sparsely annotations, as proposed in (Wen et al. 2016). A natural approach to increasing the inter-class feature distance and clustering samples of the same classes is to introduce prototypical metric learning. For labeled proposals, a prototype vector p^i is maintained for each class $i \in C$. Let v^c_{roi} represent either v^c_{roi} or $\tilde{v}^c_{roi}$ for an object of class c. The prototypical distance loss $\mathcal{L}_{proto}$ is defined as follows

$$\mathcal{L}_{proto} = \sum_{i=0}^{C} \ell(\mathrm{v}^c_{roi}, p^i), \tag{7}$$

where the function $\ell(\mathrm{v}^c_{roi}, p^i)$ is given by

$$\ell(\mathrm{v}^c_{roi}, p^i) = \begin{cases} Dist(\mathrm{v}^c_{roi}, p^i) & i = c \\ max\{0, \epsilon - Dist(\mathrm{v}^c_{roi}, p^i)\} & otherwise. \end{cases} \tag{8}$$

Here, $Dist(\cdot,\cdot)$ represents the distance which calculates the pairwise distances between feature vectors. The margin ϵ defines how close similar items should be compared to dissimilar ones. To establish the set of class prototypes $\mathcal{P} = p^0, p^1, ..., p^C$ for our feature generation method, we compute the mean of the feature vectors corresponding to each class, including both real RoI features v_{roi} and synthetic features $\tilde{v}_{roi}$. These prototypes are designed to evolve gradually as the associated feature vectors change during training. The prototypical distance loss calculation is deferred until a burn-in period of I_{begin} iterations is completed, allowing feature embeddings to stabilize and accurately encode class-specific information. Additionally, every I_{per} iterations, a new set of prototypes $\mathcal{P}_{new}$ is derived. The current prototypes $\mathcal{P}$ are then updated by combining them with $\mathcal{P}_{new}$ using a momentum parameter η

$$\mathcal{P} = \eta\mathcal{P} + (1-\eta)\mathcal{P}_{new}. \tag{9}$$

This gradual updating of class prototypes ensures that they retain historical context while adapting to new features.

Class-relation Reasoning After generating the synthetic feature bank $\tilde{v}_{roi}$ for each class c using the VAE Generator and Adaptor, guidance on category relationships is missing. To address this, we compute the class centers of the $\tilde{v}_{roi}$ to obtain a set of class feature embeddings $\mathcal{T}$. To further enhance inter-class relational reasoning, we construct the relation graph using self-attention mechanisms inspired by transformers (Vaswani et al. 2017). Concretely, we project $\mathcal{T}$ into graph node representations $\mathcal{V}_f, \mathcal{V}_g, \mathcal{V}_h$ by three linear layers, respectively. The self-attention matrix $\mathbb{A}^{inter}_{f,g}$ is calculated by a matrix multiplications as

$$\mathbb{A}^{inter}_{f,g} = \sigma(\mathcal{V}^T_f \times \mathcal{V}_g), \tag{10}$$

where σ represents the Softmax function. Consequently, we obtain the augmented class feature embeddings $\mathcal{T}^{'}$ by

$$\mathcal{T}^{'} = \mathbb{A}^{inter}_{f,g} \times \mathcal{V}^T_h. \tag{11}$$

Then the v_{roi} serves as the input feature, the augmented class feature embedding $\mathcal{T}^{'}$ provides the associative information, and the adjacency matrix $\mathcal{R} \in \mathbb{R}^{C\times C}$ of $\mathcal{T}^{'}$ acts as the relational weight. The relational reasoning function $\psi(\cdot)$ is defined by

$$\psi(\mathcal{T}, \mathrm{v}_{roi}, R) = \sigma(\mathcal{T}^{'T} \times \mathrm{v}_{roi}) \times \mathcal{R}. \tag{12}$$

Then the training loss $\mathcal{L}_{cls}$ is rewrited as

$$\begin{aligned} \mathcal{L}_{cls} = &-\sum c \cdot \sigma(\Phi_{cls}(v_{roi}) + \psi(\mathcal{T}, \mathrm{v}_{roi}, R)) \\ &-\sum c \cdot \sigma(\Phi_{cls}(\tilde{v}_{roi}) + \psi(\mathcal{T}, \mathrm{v}_{roi}, R)). \end{aligned} \tag{13}$$

Collaborative Training

Collaborative training dynamically adjusts the training of the Adaptor and SAOD detector, progressively introducing synthetic features, updating class prototypes, and performing class-relation reasoning to better capture and adapt to the evolving feature representations. During the detector training process, the iterations are denoted as $Iter$. In the initial burn-in phase, which lasts until iteration I_{begin}, the detector's classifier $\Phi_{cls}(\cdot)$ is trained using the standard classification loss $\mathcal{L}_{cls} = -\sum c \cdot \sigma(\Phi_{cls}(v_{roi}))$ to ensure the stability of the initial feature embeddings and the accurate encoding of class information.

Once the burn-in phase is complete, every I_{per} iteration, RoI features v_{roi} are extracted from the detector $\Phi_{Det}(\cdot)$. The VAE encoder $Enc(\cdot)$, VAE decoder $Dec(\cdot,\cdot)$ and Adaptor $Adt(\cdot)$ are then trained according to the total loss function specified in Eq.(4). Synthetic features $\tilde{v}_{roi}$ are generated by inputting class name c and noise z into $Dec(\cdot,\cdot)$ and $Adt(\cdot)$, as described in Eq.(5). Simultaneously, class prototypes $\mathcal{P}$ are computed or updated using Eq.(9) to capture the evolving class representations. During iterations before

Algorithm 1: Adaptive Feature Generation for SAOD

Input: labeled training set D, object instances $\hat{x}$, class name $c \in C$, $I_{begin} = I_{per}$

Parameter: SAOD detector $\Phi_{Det}(\cdot)$, VAE encoder $Enc(\cdot)$, VAE decoder $Dec(\cdot,\cdot)$, Adaptor $Adt(\cdot)$, and Class-relation reasoning $\psi(\cdot)$

Output: The trained $\Phi_{Det}(\cdot)$

while $Iter \geq 0$ **do**
  if $Iter \leq I_{begin}$ **then**
    $\Phi_{Det}(\cdot) \leftarrow$ Train $\Phi_{cls}(\cdot)$ by $\mathcal{L}_{cls} = -\sum c \cdot \sigma(\Phi_{cls}(v_{roi}))$
  else
    if $Iter\%I_{per} = 0$ **then**
      Extract RoI features v_{roi} from $\Phi_{Det}(\cdot)$
      Train $Enc(\cdot)$, $Dec(\cdot,\cdot)$ and $Adt(\cdot)$ by Eq.(4)
      $\tilde{v}_{roi} \leftarrow$ Input c and z into $Dec(\cdot,\cdot)$ and $Adt(\cdot)$ by Eq.(5)
      Compute or update prototypes $\mathcal{P}$ by Eq.(9)
    else
      Prototypical metric learning by Eqs. (7–8)
    end if
    $\psi(\cdot) \leftarrow$ Class-relation reasoning by Eqs. (10–12)
    $\Phi_{Det}(\cdot) \leftarrow$ Train $\Phi_{cls}(\cdot)$ by Eq.(13)
  end if
end while
return $\Phi_{Det}(\cdot)$

reaching the next I_{per}, the model performs prototypical metric learning by Eqs.(7–8) to further refine the feature space.

For iterations where $Iter > I_{begin}$ (with I_{begin} set to I_{per} for convenience), the model engages in class-relation reasoning as outlined in Eqs. (10–12), and updates the classifier $\Phi_{cls}(\cdot)$ using the modified classification loss in Eq.(13). This iterative process continues until the maximum number of iterations is reached, after which the trained detector $\Phi_{Det}(\cdot)$ is returned. Algorithm 1 details the pseudo-code for AdaptFG.

Experimental Results

Data and Metrics

We conduct experiments on the COCO (Lin et al. 2014) and PASCAL-VOC (2007+2012) (Everingham et al. 2015) datasets. For COCO, we use the standard COCO-style Average Precision (AP), and for PASCAL-VOC, we use the AP50 metric (AP at IoU=0.5). Following prior work (Suri et al. 2023; Yang, Liang, and Carin 2020; Wang et al. 2021; Zhang et al. 2020; Bourdev and Brandt 2005), we evaluate our method with five data splits: the first three from COCO and the last two from PASCAL.

Split-1: randomly deletes $p\%$ of annotations for each class c in the training set, with p values of $\{30, 50, 70\}$.

Split-2: for each class c, images containing c are selected. All annotations of class c are deleted with a probability of $p\%$, where p is $\{30, 50, 70\}$, ensuring each image has at least one annotation.

Split-3: on the COCO train set, removes $p\%$ of annotations from each image in a class-agnostic manner, retaining at least one annotation per image. p values are $\{30, 50, 70\}$.

Split-4: evaluates models on the PASCAL-VOC 2007+12 trainval set under three settings (easy, hard, extreme). 'Easy' removes one random annotation per image, 'hard' removes half, and 'extreme' retains only one annotation per image. Results are reported on the PASCAL-VOC 2007 test set.

Split-5: uses the PASCAL-VOC 2007 dataset and removes $p\%$ of annotations per class, with p values of $\{30, 40, 50\}$. Randomly selected categories are fully annotated, while others have no annotations. This split ensures at least one annotation per image.

Implementation Details

For all experiments, we utilize a Faster R-CNN (Ren et al. 2015) architecture based on ResNet-101 (He et al. 2016). It is implemented using the Detectron2 (Yuxin et al. 2019) framework. Our model is trained with a batch size of 8 for 270,000 and 18,000 iterations on the COCO and PASCAL-VOC datasets, respectively, with an initial learning rate of 0.01. The learning rate is reduced by a factor of ten twice: at 210,000 and 250,000 iterations for COCO, and at 12,000 and 15,000 iterations for PASCAL-VOC. A warmup strategy is employed, with 1000 and 100 iterations for COCO and PASCAL-VOC, respectively. Following (Suri et al. 2023), we set τ_{obj} and τ_{fg} to 0.5 and 0.4. Additionally, I_{begin}, I_{per}, ϵ and η are set to 5000, 5000, 10, 0.9. All experiments are conducted on a NVIDIA A800-SXM4-80GB GPU.

Comparison with State-of-the-arts

In this section, we compare our approach with state-of-the-art methods. All methods are evaluated using Faster R-CNN with a ResNet-101 backbone to ensure a fair comparison. SparseDet, which exhibits the best performance among current methods, serves as the baseline for our experiments. The 'Oracle' results reflect the performance achieved with full annotations, providing a theoretical upper bound.

Results on COCO Table 1 presents the AP performance on COCO Splits 1, 2, and 3. Our method outperforms all other methods across all metrics. Notably, in the 70% sparsity setting for each split, our method shows substantial advantages, emphasizing the effectiveness of augmenting visual information with feature generation. For example, in the 70% sparsity setting of Split 1, our method surpasses SparseDet by 3.9%. Although the improvements are less pronounced at 30% sparsity, our method still achieves results close to the oracle level, reflecting robust performance even with significant annotation sparsity.

Results on Pascal VOC Table 2 presents the AP50 results for our method on PASCAL VOC Splits 4 and 5. Consistent with the performance trend on COCO, we surpass SparseDet across all settings, with particularly pronounced advantages in high levels of sparsity. Specifically, when label sparsity is Extreme or 50%, our method achieves performance improvements of 3.1% and 2.8% over SparseDet, further demonstrating its effectiveness.

Method	Pub'Year	Split-1			Split-2			Split-3			100%
		30%	50%	70%	30%	50%	70%	30%	50%	70%	
Oracle	-										41.6
Pseudo Label (Niitani et al. 2019)	CVPR'19	-	27.5	-	-	-	-	-	-	-	-
BRL (Zhang et al. 2020)	ICASSP'20	-	32.7	-	-	-	-	-	-	-	-
Unbiased Teacher (Liu et al. 2021)	ArXiv'21	32.0	31.1	27.9	36.4	32.9	31.4	36.0	32.1	30.1	-
Co-mining (Wang et al. 2021)	AAAI'21	36.3	32.8	24.9	36.7	33.0	24.8	36.8	32.5	24.9	-
SparseDet (Suri et al. 2023)	ICCV'23	38.5	36.0	32.5	39.5	36.5	35.1	39.8	37.4	36.4	-
AdaptFG (Ours)	AAAI'25	**41.2**	**38.5**	**36.4**	**41.2**	**38.6**	**37.3**	**41.1**	**39.8**	**38.2**	-

Table 1: Comparison with recent SAOD methods on three splits of COCO dataset. 'Oracle' corresponds to training models using full annotations.

Method	Split-4			Split-5		
	Easy	Hard	Extreme	30%	40%	50%
Oracle	83.2			77.6		
BRL	73.50	71.7	66.2	-	-	-
Co-mining	79.6	78.4	69.6	74.4	73.3	69.9
SparseDet	81.6	80.9	75.0	76.4	75.3	73.6
AdaptFG (Ours)	**82.9**	**82.5**	**78.1**	**77.2**	**77.0**	**76.4**

Table 2: Comparison with recent SAOD methods on two splits of PASCAL-VOC dataset.

	Components				Split-4		
	PL	FI	CR	Adp	Easy	Hard	Extreme
SparseDet					81.6	80.9	75.0
	✓				81.8	81.2	75.7
	✓	✓			82.1	81.5	76.3
	✓	✓		✓	82.4	82.8	77.2
	✓	✓	✓		82.3	81.7	76.8
Ours	✓	✓	✓	✓	82.9	82.5	78.1

Table 3: Ablative performance on Split-4 by gradually applying the proposed components to the baseline SparseDet. **PL**: Prototypical Metric Learning. **FI**: Synthetic Feature Integration. **CR**: Class-relation Reasoning. **Adp**: Adaptor.

Ablation Study

We conduct ablation studies to validate our key components tailored specifically for AdaptFG, including the CLIP-infused VAE generator, RoI Feature Adaptor, prototypical metric learning and class-relation reasoning.

Effectiveness of key components Table 3 presents an ablation study on the SparseDet baseline's performance (measured by AP50) on the Split-4 dataset across three difficulty levels: Easy, Hard, and Extreme. Starting with the SparseDet baseline, adding prototypical metric learning (PL) improves performance across all levels, setting a new state-of-the-art. Incorporating synthetic feature integration (FI) further boosts results, especially at the Extreme level (AP50 rises from 75.0 to 76.3). The Adaptor (Adp) combined with PL and FI leads to substantial gains, particularly in Hard and Extreme cases (AP50 increases to 82.8 and 77.2). Class-relation reasoning (CR) also improves performance, but the full model (Ours) with all components achieves the highest AP50 scores—82.9, 82.5, and 78.1 for Easy, Hard, and Extreme levels, respectively. This demonstrates the synergistic effect of all components, with the Adaptor playing a key role in enhancing detection across different difficulty levels.

Split-4	20	50	100	200	300
Easy	81.9	82.3	**82.9**	82.6	82.7
Hard	81.5	82.2	**82.5**	82.3	82.1
Extreme	76.5	77.3	**78.1**	77.8	77.6

Table 4: Impact of the number of the synthetic features.

Method	Novel Split 1			Novel Split 3		
	1	2	3	1	2	3
Efficient-FSOD	56.5	59.6	62.0	50.8	56.2	56.5
Ours	62.3	65.0	66.8	58.5	60.7	62.7

Table 5: Performance improvement in Few-Shot Object Detection (FSOD). Using Efficient-FSOD (Yang et al. 2022) as the baseline.

Number of synthetic feature samples Table 4 shows the impact of varying the number of synthetic feature samples on AP50 across three difficulty levels (Easy, Hard, Extreme) in the Split-4 dataset. Performance consistently improves as samples increase from 20 to 100, with peak AP50 scores of 82.9, 82.5, and 78.1, respectively. Beyond 100 samples, the performance plateaus or slightly decreases, suggesting diminishing returns with additional samples.

Generalization on Few-shot Setting

We can view few-shot object detection (FSOD) task as a manifestation of SAOD under extremely sparse conditions. Tab. 5 presents the improvements achieved by our method in FSOD across various shots and data splits on the PASCAL VOC dataset, with settings indicating 1, 2, or 3 annotated instances per class. Using Efficient-FSOD (Yang et al. 2022) as the baseline, our approach consistently outperforms the baseline, with particularly significant gains observed in the 1-shot scenario. Specifically, performance increases of 5.8% and 7.7% are noted across the two data splits. This substantial enhancement underscores the effectiveness of our method in leveraging minimal annotated instances.

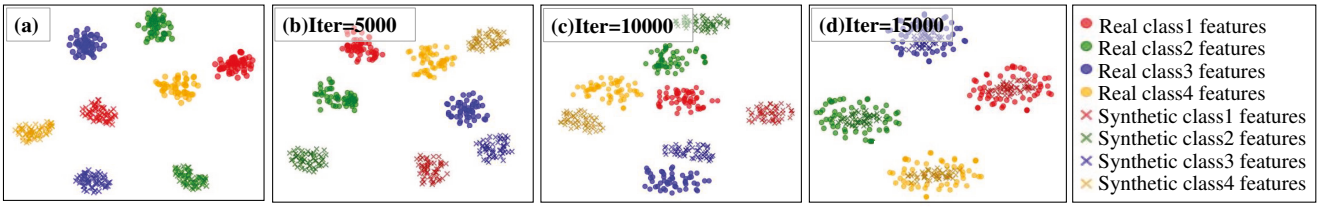

Figure 3: *t*-SNE visualizations (Van der Maaten and Hinton 2008) of the progressive alignment of synthetic (the '×' sign) and real (the dots) RoI features on Split-4 across different training iterations.

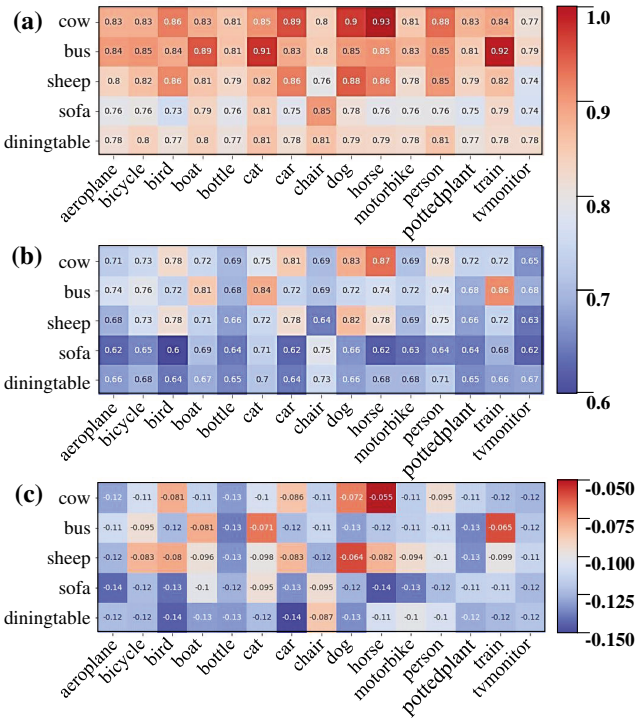

Figure 4: Correlation of the class feature embeddings before and after the class-relation reasoning between classes.

Qualitative Results

Figure 3 illustrates feature distributions on Split-4 using *t*-SNE (Van der Maaten and Hinton 2008), showing how synthetic features $\tilde{v}_{roi}$ align with real features v_{roi} across iterations. In plot (a), where synthetic features are generated without the Adaptor, there is a noticeable distribution shift from real features. However, as shown in plots (b), (c), and (d) for 5000, 10000, and 15000 iterations respectively, the alignment improves progressively, highlighting the Adaptor's effectiveness in minimizing distribution shifts.

Figure 4 illustrates changes in inter-class relationships before and after class-relation reasoning and reveals how reasoning influences inter-class similarity. We analyzed five less-annotated categories from Pascal VOC (cow, bus, sheep, sofa, dining table) against the other fifteen categories. Before reasoning (plot (a)), category similarities were influenced by data distribution, causing blurred boundaries. After reasoning (plot (b)), semantic information clarified these boundaries, reducing overall similarity and misclassification risk. The difference matrix (plot (c)) shows that reasoning had a smaller impact on already similar categories and a more significant effect on less similar ones.

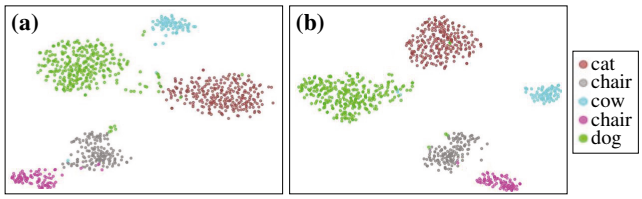

Figure 5: *t*-SNE visualization of clustering for five classes. (a) Shows initial clustering with some overlap. (b) Displays improved clustering with tighter, more distinct classes.

Prototypical metric learning significantly improves feature separation. Figure 5 shows the v_{roi} distribution for five classes using *t*-SNE. Plot (a) reveals distinguishable clusters, though some confusion remains, particularly between the brown 'cat' and cyan 'dog' clusters. After applying prototypical metric learning, as seen in plot (b), clusters become more compact, and separation improves markedly. The distinction between 'cat' and 'dog' clusters, previously prone to confusion, is significantly enhanced. This demonstrates how prototypical metric learning sharpens class boundaries, improving feature separation and representations for class-relation reasoning and classification.

Conclusion

In this work, we introduce the Adaptive Feature Generation (AdaptFG) model, designed to simultaneously avoid the impact of pseudo-label noise and enhance feature diversity in SAOD. Our approach leverages a CLIP-infused VAE Generator and an Adaptor to align the distribution shift between synthetic and real RoI features. Additionally, by incorporating inter-class relationship reasoning, we further enhance the detector's performance. Experimental results demonstrate that AdaptFG significantly improves SAOD performance on both the PASCAL VOC and COCO benchmarks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 62272083, 62472066 and 62306183, Guangdong Natural Science Foundation under

Grant 2024A1515010194, Shenzhen Natural Science Foundation under Grant JCYJ 20240813141807010.

References

Bourdev, L.; and Brandt, J. 2005. Robust object detection via soft cascade. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, 236–243. IEEE.

Everingham, M.; Eslami, S. A.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111: 98–136.

Guirguis, K.; Meier, J.; Eskandar, G.; Kayser, M.; Yang, B.; and Beyerer, J. 2023. NIFF: Alleviating Forgetting in Generalized Few-Shot Object Detection via Neural Instance Feature Forging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24193–24202.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, Y.-C.; Ma, C.-Y.; He, Z.; Kuo, C.-W.; Chen, K.; Zhang, P.; Wu, B.; Kira, Z.; and Vajda, P. 2021. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*.

Niitani, Y.; Akiba, T.; Kerola, T.; Ogawa, T.; Sano, S.; and Suzuki, S. 2019. Sampling techniques for large-scale object detection from sparsely annotated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6510–6518.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Suri, S.; Rambhatla, S.; Chellappa, R.; and Shrivastava, A. 2023. SparseDet: Improving Sparsely Annotated Object Detection with Pseudo-positive Mining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6770–6781.

Tan, M.; Pang, R.; and Le, Q. V. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790.

Tang, B.; Zhang, J.; Yan, L.; Yu, Q.; Sheng, L.; and Xu, D. 2024. Data-Free Generalized Zero-Shot Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5108–5117.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475.

Wang, T.; Yang, T.; Cao, J.; and Zhang, X. 2021. Co-mining: Self-supervised learning for sparsely annotated object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 2800–2808.

Wang, Z.; Liang, J.; He, R.; Xu, N.; Wang, Z.; and Tan, T. 2023. Improving zero-shot generalization for clip with synthesized prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3032–3042.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11–14, 2016, proceedings, part VII 14*, 499–515. Springer.

Wu, Z.; Bodla, N.; Singh, B.; Najibi, M.; Chellappa, R.; and Davis, L. S. 2018. Soft sampling for robust object detection. *arXiv preprint arXiv:1806.06986*.

Xu, J.; and Le, H. 2022. Generating representative samples for few-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9003–9013.

Xu, J.; Le, H.; and Samaras, D. 2023. Generating Features with Increased Crop-related Diversity for Few-Shot Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19713–19722.

Yang, Y.; Liang, K. J.; and Carin, L. 2020. Object detection as a positive-unlabeled problem. *arXiv preprint arXiv:2002.04672*.

Yang, Z.; Zhang, C.; Li, R.; Xu, Y.; and Lin, G. 2022. Efficient few-shot object detection via knowledge inheritance. *IEEE Transactions on Image Processing*, 32: 321–334.

Yoon, J.; Hong, S.; and Choi, M.-K. 2021. Semi-supervised object detection with sparsely annotated dataset. In *2021 IEEE International Conference on Image Processing (ICIP)*, 719–723. IEEE.

Yuxin, W.; Alexander, K.; Francisco, M.; WanYen, L.; and Ross, G. 2019. Detectron2. In *https://github.com/facebookresearch/detectron2*.

Zhang, H.; Chen, F.; Shen, Z.; Hao, Q.; Zhu, C.; and Savvides, M. 2020. Solving missing-annotation object detection with background recalibration loss. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1888–1892. IEEE.

Zhang, W.; and Wang, Y.-X. 2021. Hallucination improves few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13008–13017.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Conditional Prompt Learning for Vision-Language Models. In *CVPR*, 16795–16804.

Zhu, C.; Chen, F.; Ahmed, U.; Shen, Z.; and Savvides, M. 2021. Semantic relation reasoning for shot-stable few-shot object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8782–8791.