Deep Feature Deblurring Diffusion for Detecting Out-of-Distribution Objects

Aming Wu[1] Da Chen[2] Cheng Deng[1*]
[1]School of Electronic Engineering, Xidian University, Xi'an, China
[2]University of Bath
amwu@xidian.edu.cn, da.chen@bath.edu, chdeng@mail.xidian.edu.cn

Abstract

To promote the safe application of detectors, a task of unsupervised out-of-distribution object detection (OOD-OD) is recently proposed, whose goal is to detect unseen OOD objects without accessing any auxiliary OOD data. For this task, the challenge mainly lies in how to only leverage the known in-distribution (ID) data to detect OOD objects accurately without affecting the detection of ID objects, which can be framed as the diffusion problem for deep feature synthesis. Accordingly, such challenge could be addressed by the forward and reverse processes in the diffusion model. In this paper, we propose a new approach of Deep Feature Deblurring Diffusion (DFDD), consisting of forward blurring and reverse deblurring processes. Specifically, the forward process gradually performs Gaussian Blur on the extracted features, which is instrumental in retaining sufficient input-relevant information. By this way, the forward process could synthesize virtual OOD features that are close to the classification boundary between ID and OOD objects, which improves the performance of detecting OOD objects. During the reverse process, based on the blurred features, a dedicated deblurring model is designed to continually recover the lost details in the forward process. Both the deblurred features and original features are taken as the input for training, strengthening the discrimination ability. In the experiments, our method is evaluated on OOD-OD, open-set object detection, and incremental object detection. The significant performance gains over baselines demonstrate the superiorities of our method. The source code will be made available at: `https://github.com/AmingWu/DFDD-OOD`.

1. Introduction

Discriminating known from unknown objects is indispensable for building reliable detection systems. Currently, most object detection models [38, 14, 53, 2] usually follow a close-set assumption, i.e., the training and testing stages share the same category space. However, the real scenario is open and full of unknown objects, presenting significant challenges for close-set assumption based detectors. To promote the safe deployment of detectors, a task of unsupervised out-of-distribution object detection (OOD-OD) [6] is proposed, aiming to detect OOD objects never-seen-before during training without using any auxiliary OOD data.

For unsupervised OOD-OD [6], there exist two essential challenges: one is to only leverage the given in-distribution (ID) data to improve the ability of discriminating OOD objects. Another is to avoid the performance degradation of ID object detection. One feasible solution [6, 37] is to synthesize a series of virtual OOD features based on the ID data, which promotes the detector to learn a clear boundary between ID and OOD objects. The work [6] first utilizes ID data to estimate class-conditional distribution for each

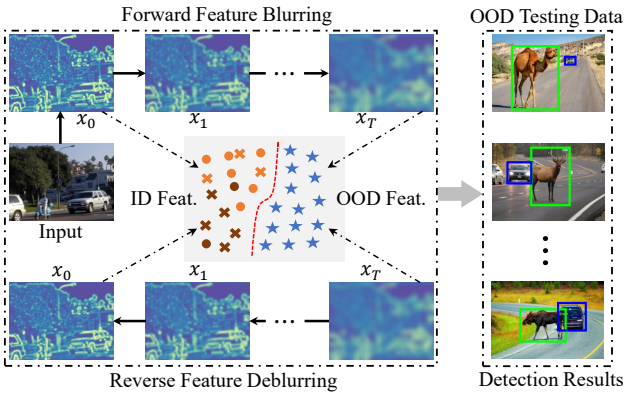


Figure 1. Deep Feature Deblurring Diffusion (DFDD) for detecting OOD objects. The forward process aims to synthesize virtual OOD features (as shown in the blue stars) by gradually performing Gaussian Blur on the extracted feature maps, which reduces the impact of lacking OOD data and improves the ability of detecting OOD objects. Meanwhile, the reverse stage is to recover the deblurred results (as shown in dark orange) of the forward output. Finally, both the deblurred results and original feature maps (as shown in orange) are taken as the input for training, which enhances the discrimination of the object classifier.

*Corresponding author

category. Then, virtual OOD features are sampled from the region that slightly deviates from the estimated distribution. However, to estimate the distributions accurately, it is critical to use a large number of objects for each category, which limits its application in the case of few samples.

To reduce the impact of lacking OOD data, in this paper, we focus on synthesizing virtual OOD features that are close to the classification boundary of ID and OOD objects. Here, we assume that the data points located on the boundary should own an important attribution, i.e., the boundary data is related to the ID data but can not be classified into ID categories. One intuitive idea is to utilize GANs [10] for feature synthesis, which is difficult to optimize [6] and could not resolve the two challenges of OOD-OD effectively. To this end, as shown in Fig. 1, we explore handling the two key challenges from the diffusion perspective [18, 39, 20]. In general, diffusion models [32, 24] contain a forward diffusion by adding noise and a reverse denoising process. For unsupervised OOD-OD, the forward process could be utilized to synthesize OOD features near the boundary, improving the capability of distinguishing OOD objects. The reverse process is exploited to recover the features involving rich object-related information, used to enhance the discrimination of the object classifier.

Unfortunately, experimental results show that using the denoising diffusion [18, 44] for feature synthesis could not boost the performance of discriminating OOD objects. The reason may be that adding much noise destroys the semantic structure of the features, resulting in the synthesized features being far away from the classification boundary of ID and OOD objects and then attenuating the discrimination of the detector. To this end, we pay attention to exploiting Gaussian Blur to replace adding noise and design a dedicated deblurring mechanism to generate expected features for addressing the two challenges of OOD-OD.

Specifically, as shown in Fig. 1, an approach of Deep Feature Deblurring Diffusion (DFDD) is proposed, which consists of forward blurring and reverse deblurring. During the forward process, we gradually perform Gaussian Blur [9] on the feature maps extracted by a backbone network. Since Gaussian Blur is a weighted average of neighboring elements, compared with adding noise roughly, this operation continually removes detail content and could retain plentiful input-related information. For example, in Fig. 1, we can see that after T iterations, compared with the original feature x_0, x_T contains much less detail information but still involves rich content related to x_0. And based on x_T, it is difficult to recognize the categories of ID objects. Therefore, x_T is the expected virtual OOD feature that is close to the classification boundary of ID and OOD objects, which is instrumental in improving the performance of distinguishing OOD objects. Next, in the reverse stage, a U-Net model [40] is designed to recover the deblurred features containing plentiful object-related information. Finally, both the deblurred features and original features are taken as the input for training, which ameliorates the discrimination ability. In the experiments, our method is evaluated on three different tasks. Extensive experimental results on multiple datasets demonstrate the superiorities of our method.

The contributions are summarized as follows:

(1) Though diffusion models have achieved impressive performance in image generation, it is under-explored for feature generation. In this paper, we convert the challenges of unsupervised OOD-OD to the diffusion problem for feature synthesis and present a new solution to strengthen the ability of discriminating OOD objects.

(2) We propose an approach of Deep Feature Deblurring Diffusion (DFDD) for unsupervised OOD-OD. Particularly, the forward process is to synthesize virtual OOD features by gradually performing Gaussian Blur. Meanwhile, a dedicated deblurring mechanism is designed to enhance the discrimination of the object classifier.

(3) In the experiments, our method is evaluated on OOD-OD [6], open-set object detection [25, 43], and incremental object detection [13]. Particularly, based on MS-COCO [30], compared with the baseline method [6], our method significantly reduces FPR95 by around **13.56%**.

2. Related Work

OOD Detection. To promote the reliable deployment of models in real scenarios, OOD detection [16, 29, 55, 59] has recently attracted much attention, whose goal is to discriminate OOD data from ID data. Most existing methods [8, 21, 23, 33, 34] focus on OOD image classification and attempt to design an effective regularization-based method. For example, the model is regularized to produce lower confidence [17] or higher energy [31] on the OOD data. Besides, Bendale et al. [1] design the OpenMax score based on the extreme value theory, which promotes the development of score-based methods [49, 45]. Though these methods have been shown to be effective, since object detection [52, 50] involves object localization and classification, these methods could not be directly applied to OOD-OD.

Recently, a task of OOD-OD [6, 51] is proposed, which aims to detect objects never-seen-before during training without degrading the detection performance for ID objects. Du et al. [6] explore estimating class-conditional distribution for each category. And virtual OOD features are sampled from the region that slightly deviates from the distribution. However, this work [6] may require a large number of objects to estimate the distributions accurately, which limits its application in the case of few samples. Besides, Du et al. [5] further propose to learn unknown-aware knowledge from auxiliary videos, which does not match the setting of unsupervised OOD-OD. The work [4] tries to utilize a distance-based mechanism to shape the learned represen-

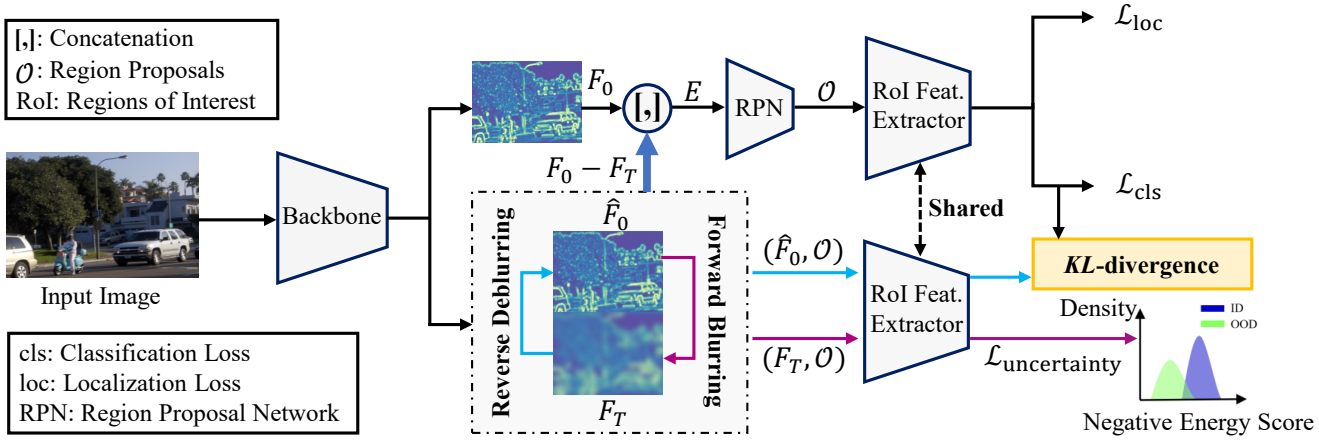


Figure 2. Deep Feature Deblurring Diffusion for unsupervised OOD-OD. We explore leveraging the idea of diffusion models to address the two challenges. Particularly, to reduce the impact of lacking OOD data for training, the forward process aims to gradually perform Gaussian Blur on the extracted feature map F_0, which outputs the virtual OOD map F_T (as shown in the red arrow). Meanwhile, the reverse deblurring process is to recover the feature map $\hat{F}_0$ that is taken as the augmentation of F_0. Finally, $\hat{F}_0$ (as shown in the blue arrow) and F_0 are taken as the input for training, which enhances the discrimination ability of the object classifier.

tations. Different from the above methods, we convert the challenges of unsupervised OOD-OD to the diffusion problem [18, 32] for feature synthesis and present a new solution to improve the ability of detecting OOD objects.

Diffusion Models. In general, diffusion models [39, 56] include forward diffusion for adding noise and a reverse process to recover the denoised data. Particularly, Ho et al. [18] first propose Denoising Diffusion Probabilistic Models, which accelerates the popularity of diffusion models. Based on this work, some methods [39, 11, 20] explore introducing the attention mechanism [48] and Variational AutoEncoder (VAE) [47] into existing diffusion models, which produce stable diffusion models and generate high-quality images. However, diffusion models are rarely used for specific feature generation. In this paper, we propose a method of Deep Feature Deblurring Diffusion, which contains a forward blurring process to synthesize virtual OOD features and a reverse deblurring stage to recover the original features. Extensive experiments on multiple datasets demonstrate the superiorities of our method.

3. Denoising Diffusion for Feature Synthesis

Diffusion models [18, 44] are a class of latent variable models, which define a Markov chain of forward diffusion process by gradually adding noise to data samples. The forward noise process is defined as follows:

$$q(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathbf{I}), \quad (1)$$

which transforms data sample x_0 to a latent noisy sample x_t by adding noise to x_0. $t \in \{1, ..., T\}$. $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s = \prod_{s=1}^{t}(1-\beta_s)$ and β_s represents the noise variance schedule [18]. During training, a denoising network ϵ_θ is trained to minimize the following loss:

$$\mathcal{L} = \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)||^2]. \quad (2)$$

In other words, at each timestep t, the denoising network ϵ_θ is tasked with correctly removing the noise ϵ. Although diffusion models could generate high-quality images, it is under-explored for feature generation. To this end, we study whether the denoising diffusion model [18] could be used for synthesizing virtual OOD features. Specifically, given the feature map x_0 extracted by a backbone network, we first utilize the forward diffusion to obtain the noisy map x_T defined as the virtual OOD map. Next, a U-Net model [40] is designed to remove the added noise and recover the original feature x_0. Finally, both the recovered feature and original feature are used for training. The overall processes are the same as our method (as shown in Algorithm 1).

In the experiments, we follow the settings of the baseline work [6]. PASCAL VOC [7] and MS-COCO [30] are separately taken as ID data for training and OOD data for evaluation. Compared with the baseline [6], FPR95 performance is increased by around **7.26%**. The reason may be that directly adding noise into the feature map x_0 is prone to destroy the semantic structure of x_0, which results in the synthesized virtual OOD feature x_T being far away from the classification boundary of ID and OOD objects. At this time, the feature x_T could not be used to effectively enhance the ability of discriminating OOD objects.

4. Deep Feature Deblurring Diffusion

Based on the above analysis, adding noise may hinder the application of diffusion models for feature generation. To this end, we explore replacing adding noise with Gaussian Blur and propose a new diffusion model, i.e., Deep

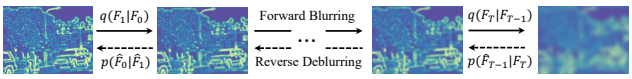


Figure 3. Details of Deep Feature Deblurring Diffusion for Feature Synthesis. During the forward process, we gradually perform Gaussian Blur on the extracted features to obtain the virtual OOD map F_T. Meanwhile, it is worth noting that there are no learnable parameters in the forward diffusion process. For the reverse process, a network is designed to deblur continually and recover the feature map $\hat{F}_0$. Finally, for the forward and reverse processes, the size of feature maps is kept unchanged.

Feature Deblurring Diffusion, for unsupervised OOD-OD. Concretely, we follow the settings [6] and only utilize the ID data that own a fixed number of categories for training. During inference, the object detector should distinguish ID objects from OOD objects accurately.

4.1. Forward Blurring Diffusion

Concretely, as shown in Fig. 2 and 3, we follow the baseline work [6] and exploit the widely used object detector, i.e., Faster R-CNN [38, 14], as the basic detection model. Given an input image, a backbone network, e.g., ResNet [15], is utilized to extract the corresponding feature map $F_0 \in \mathbb{R}^{w\times h\times c}$, where w, h, and c separately denote width, height, and the number of channels.

To alleviate the impact of lacking OOD data, we leverage the forward process to synthesize virtual OOD features. Specifically, the diffusion process is fixed to a Markov chain that gradually performs Gaussian Blur on the extracted feature map according to a variance schedule σ_1, ..., σ_T:

$$F_t = F_{t-1} * \mathcal{G}(\sigma_t),\ \mathcal{G}(\sigma_t) = \frac{1}{2\pi\sigma_t^2} e^{-(i^2+j^2)/2\sigma_t^2}, \quad (3)$$

where $\mathcal{G}(\sigma_t)$ represents Gaussian Kernel with the variance σ_t. (i, j) indicates the position in the kernel. $t = 1, ..., T$. $F_t \in \mathbb{R}^{w\times h\times c}$ is the convolutional output. Besides, to ensure the blur effect, the variance is increased linearly. Here, we ignore the fact that the kernel size could be changed and instead fix the size to 5×5. Meanwhile, it is worth noting that the forward process has no learnable parameters.

Finally, the forward process owns a notable property that it admits blurring F_t at an arbitrary timestep t:

$$\begin{aligned} F_t &= F_{t-1} * \mathcal{G}(\sigma_t) = (F_{t-2} * \mathcal{G}(\sigma_{t-1})) * \mathcal{G}(\sigma_t) \\ &= F_0 * \mathcal{G}(\sigma_1) * \mathcal{G}(\sigma_2) * \cdots * \mathcal{G}(\sigma_t) \\ &= F_0 * \mathcal{G}(\sigma), \end{aligned} \quad (4)$$

where $\sigma = \sum_{k=1}^{t} \sigma_k$. Using this property is beneficial for reducing the computational time of the forward process.

Since Gaussian Blur is a weighted average of neighboring elements, compared with directly adding noise [18], this operation reduces the damage to the semantic structure of the input feature F_0 and retains rich input-related information. Furthermore, after multiple Gaussian Blur operations, it is difficult to recognize the category of the blurred feature $F_T \in \mathbb{R}^{w\times h\times c}$. Thus, F_T is the expected virtual OOD map that is close to the classification boundary of ID and OOD objects, which is instrumental in improving the ability of discriminating OOD objects.

4.2. Reverse Deblurring Process

The reverse process aims to recover the original features from the blurred features, which is instrumental in improving the discrimination of the classifier for ID objects. Concretely, as shown in Fig. 3, taking the output F_T of the forward process as the input, a U-Net model [40] is designed to predict detail feature $D_t \in \mathbb{R}^{w\times h\times c}$. Then, we combine the predicted detail feature with the blurred feature as the input of the model. The processes are shown as follows:

$$D_t = \epsilon_\theta(\hat{F}_t, t), \quad \hat{F}_{t-1} = \hat{F}_t + D_t, \quad (5)$$

where $\epsilon_\theta(\cdot,\cdot)$ represents the learned U-Net model. And $t = T, ..., 1$. $\hat{F}_T = F_T$. $\hat{F}_{t-1} \in \mathbb{R}^{w\times h\times c}$ is the output at the timestep t. Next, taking $\hat{F}_{t-1}$ as the input, we continually perform the above operations to obtain the recovered deblur feature map $\hat{F}_0 \in \mathbb{R}^{w\times h\times c}$.

During training, a loss function $\mathcal{L}_{DFDD}$ is proposed to promote the designed U-Net model to possess the capability of detail prediction:

$$\mathcal{L}_{DFDD} = \mathbb{E}_t[||\epsilon_t - \epsilon_\theta(\hat{F}_t, t)||^2],\ \epsilon_t = F_{t-1} - F_t. \quad (6)$$

Since blur is continually strengthened, ϵ_t describes the lost detail from the timestep $t-1$ to t. And $\epsilon_\theta(\hat{F}_t, t)$ is to recover the lost detail, which is beneficial for promoting the deblurred $\hat{F}_0$ to contain plentiful object-related information.

4.3. DFDD-Driven OOD Object Detection

In general, object detection includes object localization and classification. Thus, strengthening object-related information in the feature F_0 is beneficial for detecting objects accurately. As shown in Fig. 2, we first perform a residual operation between F_0 and the blurred feature F_T, whose output involves rich information of object structure. Then, the residual output is concatenated with F_0 to obtain the enhanced result $E \in \mathbb{R}^{w\times h\times c}$, i.e., $E = \Psi([F_0, F_0 - F_T])$, where $\Psi(\cdot) \in \mathbb{R}^{1\times1\times2c\times c}$ represents one-layer convolution to transform the number of channels.

Next, E is taken as the input of the RPN module [38, 14] to output a set of object proposals $\mathcal{O}$. Meanwhile, based on $\mathcal{O}$, RoI-Alignment followed by RoI-Feature extraction [14] is separately performed on E and the recovered map $\hat{F}_0$ to obtain $P_{\text{in}} \in \mathbb{R}^{m\times n}$ and $\hat{P}_{\text{in}} \in \mathbb{R}^{m\times n}$, where m and n denote the number of proposals and channels respectively. Then, P_{in} is taken as the input of the object classifier and

regressor to calculate the classification loss $\mathcal{L}_{\text{cls}}$ and localization loss $\mathcal{L}_{\text{loc}}$. The joint objective is shown as follows:

$$\mathcal{L}_{\text{in}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}} + \beta \cdot \text{KL}[p(P_{\text{in}}), p(\hat{P}_{\text{in}})], \quad (7)$$

where β is a hyper-parameter and is set to 0.001. The KL-divergence loss is to promote the prediction consistency between P_{in} and $\hat{P}_{\text{in}}$, which is conducive to ameliorating the ability of discriminating ID objects.

Finally, to achieve the goal of discriminating OOD objects from ID objects, based on $\mathcal{O}$, RoI-Alignment followed by RoI-Feature extraction is performed on F_T to extract OOD features $P_{\text{ood}} \in \mathbb{R}^{m \times n}$. P_{ood} and P_{in} are used to calculate an uncertainty loss [6], which regularizes the detector to produce a low OOD score for the ID object features, and a high OOD score for the synthesized OOD features:

$$\begin{aligned}\mathcal{L}_{\text{uncertainty}} = \mathbb{E}_{u \backsim P_{\text{in}}}[-\log \frac{\exp^{-\text{E}(u)}}{1+\exp^{-\text{E}(u)}}] + \\ \mathbb{E}_{v \backsim P_{\text{ood}}}[-\log \frac{1}{1+\exp^{-\text{E}(v)}}],\end{aligned} \quad (8)$$

where $\text{E}(\cdot)$ is the object-level energy score [6, 31]. During training, **the overall objective** is shown as follows:

$$\mathcal{L} = \mathcal{L}_{\text{in}} + \lambda \cdot \mathcal{L}_{DFDD} + \tau \cdot \mathcal{L}_{\text{uncertainty}}, \quad (9)$$

where λ and τ are two hyper-parameters, which are set to 0.001 and 0.1 in the experiments.

4.4. Inference for OOD Object Detection

During inference, we only leverage the forward diffusion process to synthesize the blur feature used to enhance object-related information. Meanwhile, we only calculate the uncertainty loss for OOD object detection [6]. Specifically, for a predicted bounding box **b**, the processes of distinguishing OOD objects are shown as follows:

$$\mathcal{S} = \frac{\exp^{-\text{E}(\mathbf{b})}}{1+\exp^{-\text{E}(\mathbf{b})}}, \quad \mathcal{C}(\mathbf{b}) = \begin{cases} 0 & \text{if } \mathcal{S} < \gamma, \\ 1 & \text{if } \mathcal{S} \geq \gamma. \end{cases} \quad (10)$$

For the output of the classifier $\mathcal{C}(\cdot)$, we use the threshold mechanism [6] to distinguish ID objects (the result is 1) from OOD objects (the result is 0). The threshold γ is commonly set to 0.95 so that a high fraction of ID data is correctly classified. Finally, Algorithm 1 shows the training and evaluation details of our method.

5. Further Discussion

In this section, we further discuss the advantages of the forward blurring diffusion.

Currently, most diffusion methods [18, 39, 24, 20] gradually add Gaussian noise to simulate the physical diffusion process. Though these methods have achieved impressive generative performance, they are rarely applied to feature-level generation for OOD-OD. Through experiments, we observe that adding noise to the features reduces the OOD-OD performance significantly. For this case, there may exist two reasons: one is that adding noise is a global operation, which is prone to destroy the semantic information in the original features. Another is that it is difficult to determine the appropriate range of the added noise. If the noise value is large, the semantic information in the features may be damaged rapidly.

Compared with globally adding noise, Gaussian Blur is a local operation, which is beneficial for preserving certain important information of the input and outputting expected OOD features. Meanwhile, like traditional diffusion [18], Gaussian Blur also owns a notable property (Eq. (4)) for reducing computational time. Extensive experimental results demonstrate the effectiveness of our method.

Algorithm 1 DFDD for Unsupervised OOD-OD

Input: ID data $\{X, Y\}$, randomly initialized detector with parameter φ, randomly initialized U-Net with parameter θ, weight β for the KL-loss, weight λ for the loss $\mathcal{L}_{DFDD}$, weight τ for the uncertainty loss $\mathcal{L}_{\text{uncertainty}}$.
Output: Detector φ^*, U-Net θ^*, and OOD classifier $\mathcal{C}$.
while *train* **do**
  Sample images from the ID dataset $\{X, Y\}$.
  Synthesize the virtual OOD map F_T of the forward diffusion using Eq. (3) and (4).
  for *t = T, ..., 1* **do**
    $D_t = \epsilon_\theta(\hat{F}_t, t)$ and $\hat{F}_{t-1} = \hat{F}_t + D_t$.
  end
  Calculate the overall training objective $\mathcal{L}$ using Eq. (6), (7), (8), and (9).
  Update the parameters φ and θ based on Eq. (9).
end
while *eval* **do**
  Calculate the OOD uncertainty score using Eq. (10).
  Perform thresholding comparison using Eq. (10).
end

6. Experiments

In the experiments, for unsupervised OOD-OD, we first evaluate our method on two different benchmarks [6]. Then, to further demonstrate the effectiveness of our method, we evaluate our method on class-incremental object detection (IOD) [25] and open-set object detection (OSOD) [13].

6.1. Experimental Setup

Implementation Details. We utilize Faster R-CNN [38] with RoI-Alignment layer [14] as the basic detection model. ResNet-50 [15] is taken as the backbone. The weights pre-trained on ImageNet [41] are used for initialization. For the

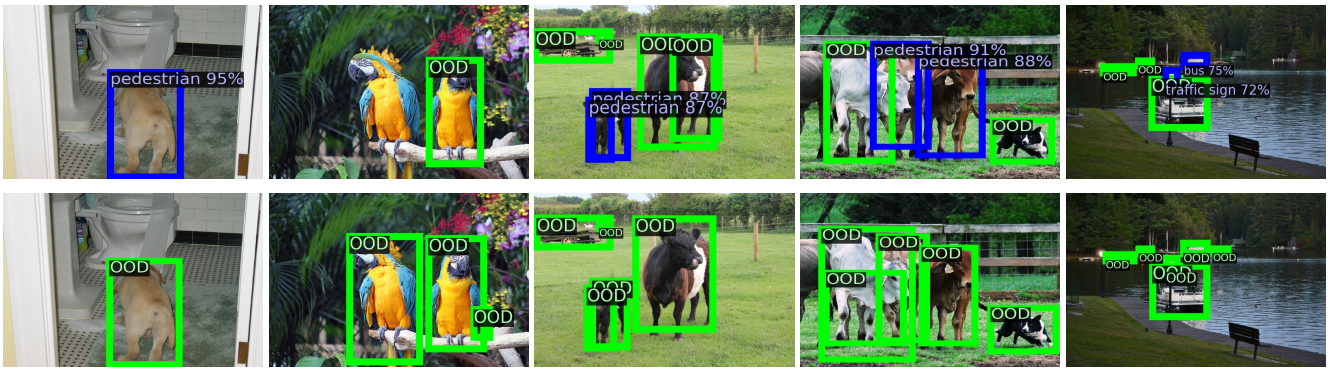


Figure 4. Detection results on the OOD images from MS-COCO. The first and second rows respectively indicate results based on VOS [6] and our method. The in-distribution dataset is BDD-100k. Blue boxes represent objects detected and classified as one of the ID categories. Green boxes indicate OOD objects. We can see that our method accurately determines OOD objects.

Method (VOC)	**FPR95** ↓	**AUROC** ↑	**mAP (ID)**↑
	OOD: MS-COCO / OpenImages		
MSP [16]	70.99 / 73.13	83.45 / 81.91	48.7
ODIN [29]	59.82 / 63.14	82.20 / 82.59	48.7
Mahalanobis [28]	67.73 / 65.41	81.45 / 81.48	48.7
Gram matrices [42]	62.75 / 67.42	79.88 / 77.62	48.7
Energy score [31]	56.89 / 58.69	83.69 / 82.98	48.7
Generalized ODIN [19]	59.57 / 70.28	83.12 / 79.23	48.1
CSI [46]	59.91 / 57.41	81.83 / 82.95	48.1
GAN-synthesis [27]	60.93 / 59.97	83.67 / 82.67	48.5
SIREN-vMF [4]	64.68 / 68.53	85.36 / 82.78	-
SIREN-KNN [4]	47.45 / 50.38	89.67 / **88.80**	-
VOS (Baseline) [6]	47.53 / 51.33	88.70 / 85.23	48.9
VOS + Blur-Aug	46.26 / 49.68	89.11 / 85.76	49.0
DDPM [18]	54.79 / 58.21	86.53 / 83.91	48.8
DFDD	**41.34 / 44.52**	**90.79** / 88.65	**49.2**
Method (BDD)	**FPR95** ↓	**AUROC** ↑	**mAP (ID)**↑
	OOD: MS-COCO / OpenImages		
MSP [16]	80.94 / 79.04	75.87 / 77.38	31.2
ODIN [29]	62.85 / 58.92	74.44 / 76.61	31.2
Mahalanobis [28]	55.74 / 47.69	85.71 / 88.05	31.2
Gram matrices [42]	60.93 / 77.55	74.93 / 59.38	31.2
Energy score [31]	60.06 / 54.97	77.48 / 79.60	31.2
Generalized ODIN [19]	57.27 / 50.17	85.22 / 87.18	**31.8**
CSI [46]	47.10 / 37.06	84.09 / 87.99	30.6
GAN-synthesis [27]	57.03 / 50.61	78.82 / 81.25	31.4
VOS (Baseline) [6]	44.27 / 35.54	86.87 / 88.52	31.3
VOS + Blur-Aug	42.85 / 34.23	86.91 / 88.67	31.3
DDPM [18]	50.63 / 41.12	83.79 / 85.63	31.2
DFDD	**30.71 / 22.67**	**90.74 / 92.48**	31.4

Table 1. Performance (%) of unsupervised OOD-OD. All methods are trained based on ID data and do not use any auxiliary data. ↑ denotes larger values are better and ↓ denotes smaller values are better. "VOS + Blur-Aug" indicates that the training images are performed Gaussian Blur as data augmentation and employs VOS [6] for detection. "DDPM" represents that our method is replaced with DDPM [18]. The training processes are kept unchanged.

forward process, we perform five Gaussian Blur operations. σ_1 and σ_5 in Eq. (3) are separately set to 0.1 and 0.8. For the reverse stage, the encoder and decoder in the U-Net model ϵ_θ all consist of three convolutional layers. All the experiments are trained using the standard SGD optimizer with a learning rate of 0.02.

Datasets. For unsupervised OOD-OD, PASCAL VOC [7] and Berkeley DeepDrive (BDD-100k) [57] datasets are taken as the ID data for training. Meanwhile, we adopt MS-COCO [30] and OpenImages [26] as the OOD datasets to evaluate the trained model. And the OOD datasets are manually examined to guarantee they do not contain ID categories. PASCAL-VOC [7] includes the following 20 categories: Person, Car, Bicycle, Boat, Bus, Motorbike, Train, Airplane, Chair, Bottle, Dining Table, Potted Plant, TV, Sofa, Bird, Cat, Cow, Dog, Horse, Sheep. And the corresponding number of ID training and validation images is 16,551 and 4,952. BDD-100k [57] contains the following 10 classes: Pedestrian, Rider, Car, Truck, Bus, Train, Motorcycle, Bicycle, Traffic light, Traffic sign. And the corresponding number of ID training and validation images is 69,853 and 10,000.

Besides, for IOD, we follow the standard evaluation protocol [25] and evaluate our method on PASCAL VOC [7]. We initially learn 10, 15, or 19 base classes, and then introduce 10, 5, or 1 new classes as the second task. Finally, for OSOD, we follow the work [13] and utilize 20 VOC classes and 60 non-VOC classes in COCO to evaluate our method under different open-set conditions. To effectively exploit the synthesized blurred features, we train a binarized classifier, i.e., the output of the known category is 1, and the output of the blurred features is 0. By minimizing the cross-entropy loss, the discrimination ability of the object classifier could be well enhanced.

Metrics. For unsupervised OOD-OD, we report: (1) the false positive rate (FPR95) of OOD objects when the true positive rate of ID objects is at 95%; (2) the area under the receiver operating characteristic curve (AUROC); (3) mean average precision (mAP). For OSOD, we use Wilderness Impact (WI) [3] to measure the degree of unknown objects misclassified to known classes. And we also use Absolute Open-Set Error (AOSE) [35] to count the number of mis-

10 + 10 setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	**mAP**
Faster ILOD (50) [36]	72.8	75.7	71.2	60.5	61.7	70.4	83.3	76.6	53.1	72.3	36.7	70.9	66.8	67.6	66.1	24.7	63.1	48.1	57.1	43.6	62.2
ORE (50) [22]	63.5	70.9	58.9	42.9	34.1	76.2	80.7	76.3	34.1	66.1	56.1	70.4	80.2	72.3	81.8	42.7	71.6	68.1	77	67.7	64.6
OW-DETR (50) [12]	61.8	69.1	67.8	45.8	47.3	78.3	78.4	78.6	36.2	71.5	57.5	75.3	76.2	77.4	79.5	40.1	66.8	66.3	75.6	64.1	65.7
ROSETTA (50) [54]	74.2	76.2	64.9	54.4	57.4	76.1	84.4	68.8	52.4	67.0	62.9	63.3	79.8	72.8	78.1	40.1	62.3	61.2	72.4	66.8	66.8
iOD (50) [25]	76.0	74.6	67.5	55.9	57.6	75.1	85.4	77.0	43.7	70.8	60.1	66.4	76.0	72.6	74.6	39.7	64.0	60.2	68.5	60.5	66.3
iOD + Ours (50)	77.4	73.9	71.2	55.2	58.2	80.7	85.5	80.0	46.2	74.3	57.3	76.2	81.3	76.4	79.7	45.8	67.7	66.1	70.9	67.9	**69.6**
iOD (75) [25]	39.0	36.5	28.4	19.4	24.2	47.2	56.7	41.0	19.1	48.0	21.1	32.1	43.0	36.3	40.0	14.8	40.1	36.5	37.3	45.3	35.3
iOD + Ours (75)	42.9	39.6	31.3	19.9	26.2	54.5	62.9	39.2	18.1	40.8	24.1	34.8	41.3	42.9	36.8	16.8	41.7	33.9	39.0	48.0	**36.7**
15 + 5 setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	**mAP**
Faster ILOD (50) [36]	66.5	78.1	71.8	54.6	61.4	68.4	82.6	82.7	52.1	74.3	63.1	78.6	80.5	78.4	80.4	36.7	61.7	59.3	67.9	59.1	67.9
ORE (50) [22]	75.4	81.0	67.1	51.9	55.7	77.2	85.6	81.7	46.1	76.2	55.4	76.7	86.2	78.5	82.1	32.8	63.6	54.7	77.7	64.6	68.5
OW-DETR (50) [12]	77.1	76.5	69.2	51.3	61.3	79.8	84.2	81.0	49.7	79.6	58.1	79.0	83.1	67.8	85.4	33.2	65.1	62.0	73.9	65.0	69.4
ROSETTA (50) [54]	76.5	77.5	65.1	56.0	60.0	78.3	85.5	78.7	49.5	68.2	67.4	71.2	83.9	75.7	82.0	43.0	60.6	64.1	72.8	67.4	69.2
iOD (50) [25]	78.4	79.7	66.9	54.8	56.2	77.7	84.6	79.1	47.7	75.0	61.8	74.7	81.6	77.5	80.2	37.8	58.0	54.6	73.0	56.1	67.8
iOD + Ours (50)	77.4	79.4	71.6	57.1	62.0	74.3	85.7	80.5	50.1	79.4	65.5	81.7	84.7	76.5	77.5	41.5	63.1	58.5	72.5	67.2	**70.3**
iOD (75) [25]	40.7	40.9	28.7	19.1	23.8	61.6	56.1	38.8	23.6	47.5	18.7	40.1	40.2	41.5	39.8	9.1	40.6	32.4	41.9	47.6	36.6
iOD + Ours (75)	46.2	43.6	31.1	29.3	33.5	45.2	61.0	40.0	23.7	51.3	24.2	38.1	44.8	42.9	41.8	11.1	42.6	33.0	41.7	48.3	**38.7**
19 + 1 setting	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv	**mAP**
Faster ILOD (50) [36]	64.2	74.7	73.2	55.5	53.7	70.8	82.9	82.6	51.6	79.7	58.7	78.8	81.8	75.3	77.4	43.1	73.8	61.7	69.8	61.1	68.6
ORE (50) [22]	67.3	76.8	60.0	48.4	58.8	81.1	86.5	75.8	41.5	79.6	54.6	72.8	85.9	81.7	82.4	44.8	75.8	68.2	75.7	60.1	68.9
OW-DETR (50) [12]	70.5	77.2	73.8	54.0	55.6	79.0	80.8	80.6	43.2	80.4	53.5	77.5	89.5	82.0	74.7	43.3	71.9	66.6	79.4	62.0	70.2
ROSETTA (50) [54]	75.3	77.9	65.3	56.2	55.3	79.6	84.6	72.9	49.2	73.7	68.3	71.0	78.9	77.7	80.7	44.0	69.6	68.5	76.1	68.3	69.6
iOD (50) [25]	78.2	77.5	69.4	55.0	56.0	78.4	84.2	79.2	46.6	79.0	63.2	78.5	82.7	79.1	79.9	44.1	73.2	66.3	76.4	57.6	70.2
iOD + Ours (50)	77.2	77.8	72.3	56.3	61.0	81.6	84.8	81.2	50.7	82.1	63.1	81.6	83.3	76.9	77.5	44.8	73.0	69.5	75.1	60.1	**71.5**
iOD (75) [25]	35.9	44.7	31.6	22.4	26.9	52.0	56.5	38.7	21.6	48.4	21.2	35.9	37.9	30.7	38.7	17.2	38.5	34.2	40.7	46.6	36.0
iOD + Ours (75)	42.5	45.3	32.4	22.1	30.0	59.9	62.0	42.9	24.2	48.2	24.4	39.0	42.1	38.5	42.1	18.6	46.7	37.1	45.1	47.2	**39.5**

Table 2. Performance (%) analysis of class-incremental object detection. 'iOD + Ours' indicates that our method is plugged into iOD [25]. Here, '50' and '75' separately represent that the mAP metric is calculated when the IOU threshold is set to 0.5 and 0.75.

Method	VOC	VOC-COCO-20				VOC-COCO-40				VOC-COCO-60			
	$\text{mAP}_{\mathcal{K}\uparrow}$	$\text{WI}_{\downarrow}$	$\text{AOSE}_{\downarrow}$	$\text{mAP}_{\mathcal{K}\uparrow}$	$\text{AP}_{\mathcal{U}\uparrow}$	$\text{WI}_{\downarrow}$	$\text{AOSE}_{\downarrow}$	$\text{mAP}_{\mathcal{K}\uparrow}$	$\text{AP}_{\mathcal{U}\uparrow}$	$\text{WI}_{\downarrow}$	$\text{AOSE}_{\downarrow}$	$\text{mAP}_{\mathcal{K}\uparrow}$	$\text{AP}_{\mathcal{U}\uparrow}$
FR-CNN [38]	80.10	18.39	15118	58.45	0	22.74	23391	55.26	0	18.49	25472	55.83	0
FR-CNN† [38]	80.01	18.83	11941	57.91	0	23.24	18257	54.77	0	18.72	19566	55.34	0
PROSER [58]	79.68	19.16	13035	57.66	10.92	24.15	19831	54.66	7.62	19.64	21322	55.20	3.25
ORE [22]	79.80	18.18	12811	58.25	2.60	22.40	19752	55.30	1.70	18.35	21415	55.47	0.53
DS [35]	80.04	16.98	12868	58.35	5.13	20.86	19775	55.31	3.39	17.22	21921	55.77	1.25
OpenDet [13]	80.02	14.95	11286	58.75	14.93	18.23	16800	55.83	10.58	14.24	18250	56.37	4.36
OpenDet + Ours	**80.26**	**12.73**	**10727**	**60.21**	**16.22**	**14.92**	**13638**	**57.98**	**12.09**	**12.57**	**16668**	**57.51**	**5.12**

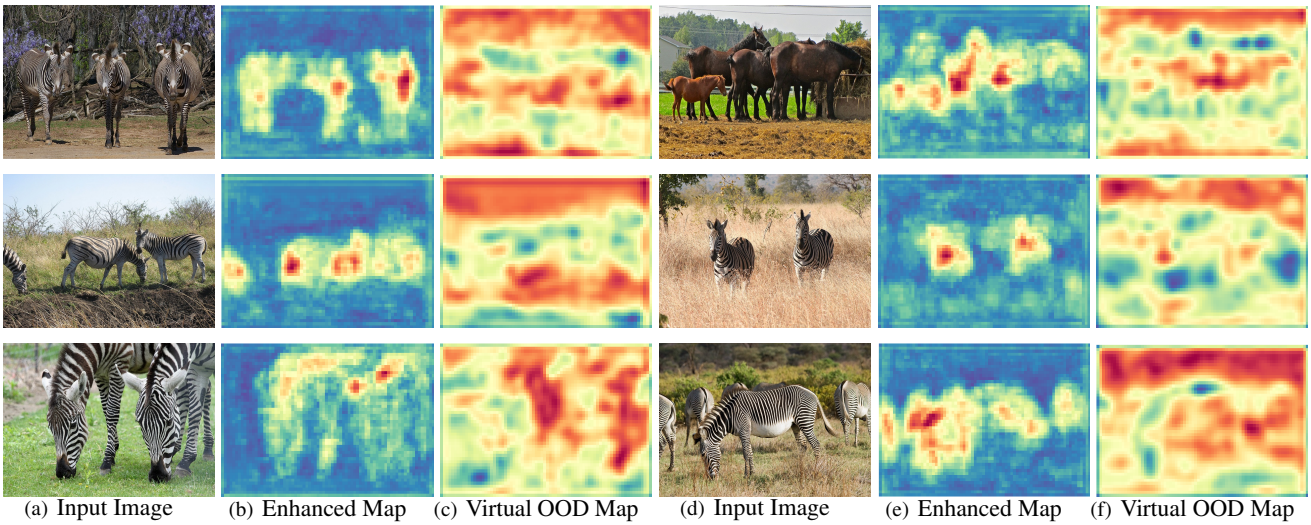
Table 3. Performance analysis of OSOD. We report close-set performance ($\text{mAP}_{\mathcal{K}}$) on VOC, and both close-set ($\text{mAP}_{\mathcal{K}}$) and open-set (WI, AOSE, $\text{AP}_{\mathcal{U}}$) performance of different methods on VOC-COCO-{20, 40, 60}. Here, '20', '40', and '60' indicate that the testing COCO images separately contain 20, 40, and 60 non-VOC classes. '†' means a higher score threshold (i.e., 0.1) for testing.

classified unknown objects.

6.2. OOD-OD Performance Analysis

In Table 1, we show the performance of unsupervised OOD-OD. Although different methods own similar performance for ID objects, the detection performance for OOD objects differs significantly. This shows that existing detectors are easily affected by OOD objects. Besides, we can see that our method achieves the best performance. Particularly, based on BDD-100k [57], compared with VOS [6], our method separately reduces FPR95 by **13.56%** and **12.87%**, which demonstrates the effectiveness of our method. Meanwhile, directly adding noise into the extracted features degrades the performance. The reason may be that this operation destroys the semantic structure, which results in the synthesized OOD features being far away from the classification boundary of ID and OOD objects. This further indicates replacing adding noise with Gaussian Blur is effective for synthesizing OOD features, which alleviates the impact of lacking OOD data for training and improves the ability of discriminating OOD objects from ID objects.

Fig. 4 shows several OOD detection examples. Com-



(a) Input Image (b) Enhanced Map (c) Virtual OOD Map (d) Input Image (e) Enhanced Map (f) Virtual OOD Map

Figure 5. Visualization of the Enhanced map E (i.e., $E = \Psi([F_0, F_0 - F_T])$) and Virtual OOD map F_T (Eq. (3)) based on the OOD data (MS-COCO). For each feature map, the channels corresponding to the maximum value are selected for visualization.

Blur	Enhance	Deblur	**FPR95** ↓	**AUROC** ↑	**mAP**↑
✓			40.04%	89.06%	30.8%
✓	✓		36.76%	90.12%	31.1%
✓		✓	35.82%	90.38%	31.2%
✓	✓	✓	**30.71%**	**90.74%**	**31.4%**

Table 4. Ablation analysis of DFDD for unsupervised OOD-OD. 'Blur' indicates we only perform the forward blur process and do not calculate $\mathcal{L}_{DFDD}$. 'Enhance' represents $E = \Psi([F_0, F_0 - F_T])$. 'Deblur' denotes we utilize the deblurred maps to compute the KL-divergence loss (Eq. (7)).

Iteration Number T	**FPR95** ↓	**AUROC** ↑	**mAP**↑
1	38.45%	88.76%	30.8%
3	36.94%	89.12%	31.1%
5	**30.71%**	**90.74%**	**31.4%**

Table 5. Analysis of the iteration number T in the reverse stage.

pared with VOS [6], our method detects OOD objects accurately. Taking the first image as an example, VOS [6] misclassifies the dog into the pedestrian category. Whereas, our method could localize and recognize OOD objects effectively, which further demonstrates that our method is indeed instrumental in improving the discrimination.

6.3. Performance Analysis of IOD and OSOD

To further demonstrate the effectiveness of our method, we verify our method on two different tasks, i.e., IOD [25] and OSOD [13]. We directly plug our method into the two state-of-the-art methods [25, 13]. Meanwhile, we do not utilize the uncertainty loss. The training and testing processes are the same as the two baselines [25, 13]. Table 2 shows the IOD performance. We can see that when the IOU threshold is separately set to 0.5 and 0.75, plugging our method improves the performance of the baseline method [25] effectively. For example, for the "19 + 1" setting, when the IOU threshold is set to 0.75, our method outperforms iOD [25] by 3.5%. Besides, in Table 3, we show the performance of OSOD. Compared with unsupervised OOD-OD, during testing, OSOD usually requires to indicate the number of unknown categories [13]. Plugging our method significantly improves the baseline's performance based on all the metrics. These results all demonstrate that the proposed diffusion method could synthesize expected virtual features, which is beneficial for enhancing the discrimination ability of the object detector.

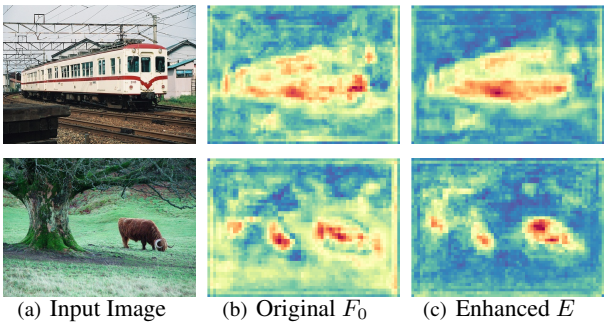
Variance σ_T	**FPR95** ↓	**AUROC** ↑	**mAP**↑
0.4	36.65%	89.67%	31.3%
0.6	34.02%	90.48%	31.2%
0.8	**30.71%**	**90.74%**	**31.4%**
1.0	32.43%	90.56%	**31.4%**

Table 6. Analysis of the variance σ_T in Gaussian Blur.

6.4. Ablation and Visualization Analysis

In this section, we utilize BDD-100k as the ID data for training and MS-COCO as the OOD data to perform an ablation analysis of our method.

Analysis of DFDD. Our method mainly includes the forward blurring process and the reverse deblurring process. In Table 4, we make an ablation experiment of our method. We can see that only performing the blur operation on the extracted features could improve the ability of detecting OOD objects. This shows that blurring current features is indeed beneficial for promoting virtual OOD features to be close to the classification boundary of ID and OOD objects. Next, we observe that using F_T to make feature enhancement and leveraging the deblurring process to perform feature augmentation all improve the detection performance. This fur-



(a) Input Image (b) Original F_0 (c) Enhanced E

Figure 6. Visualization of the feature F_0 and enhanced feature E.

ther demonstrates that our method could indeed synthesize expected virtual features, which enhances the discrimination of the object detector.

The size of Gaussian kernel. In general, the size of a Gaussian Kernel can be set up to $\lfloor(6\sigma+1)\times(6\sigma+1)\rfloor$. Thus, based on our σ setting, when the kernel size is set larger than 5×5, the performance is minimally affected.

The iteration number T in the reverse stage. To promote the recovered features to contain rich object-related information, we employ the same model to perform the deblurring operation continually (as shown in Algorithm 1). Here, we make an ablation analysis of the iteration number. We do not change our method and training details. Table 5 shows the results. Since the output of the forward process is very blurry, it is difficult to use a small number of iterations to recover the deblurred results, which could not strengthen the discrimination. This indicates that using more iterations is beneficial for obtaining the augmented features containing plentiful object-related information, which improves the detection performance of the model.

Analysis of the variance σ_T. To reduce the impact of lacking OOD data, the forward process gradually performs Gaussian Blur to obtain expected virtual OOD maps. Here, we make an analysis of the variance σ_T (Eq. (3)). In Table 6, we can see that setting a small value for σ_T weakens the blur strength of the synthesized features, which could not be used to well enhance the discrimination ability. Instead, setting a large value is prone to promote the synthesized features to be far away from the classification boundary of ID and OOD objects. For our method, when σ_T is set to 0.8, the performance is the best.

Analysis of hyper-parameters. For our method, we utilize the hyper-parameter β for the KL-divergence loss (Eq. (7)), the hyper-parameter λ for the loss $\mathcal{L}_{DFDD}$ (Eq. (9)), and the hyper-parameter τ for the loss $\mathcal{L}_{\text{uncertainty}}$ (Eq. (9)). Since the uncertainty loss $\mathcal{L}_{\text{uncertainty}}$ is directly related to the current task, the value of τ should be set larger than β and λ. Meanwhile, if β and λ are set to a small value, the role of the two corresponding losses will be weakened in optimization. Here, we make an ablation analysis of these hyper-parameters. And we only change these hyper-parameters and keep other modules unchanged.

(1) The hyper-parameter β in Eq. (7) is to balance the detection loss and the loss that aims to minimize the KL-divergence between the prediction probabilities from P_{in} and $\hat{P}_{\text{in}}$. In the experiments, we observe that when β is set to 0.01, 0.001, and 0.0001, the performance of FPR95 is 32.92%, 30.71%, and 31.86%.

(2) The goal of the hyper-parameter λ in Eq. (9) is to weigh the importance of the module of DFDD. In the experiments, we find that when λ is set to 0.01, 0.001, and 0.0001, the corresponding FPR95 performance is 33.15%, 30.71%, and 32.28%.

(3) The hyper-parameter τ in Eq. (9) is to constrain the uncertainty loss $\mathcal{L}_{\text{uncertainty}}$. In the experiments, we observe that when τ is set to 0.5, 0.1, and 0.01, the FPR95 performance is 33.95%, 30.71%, and 32.53%.

Visualization analysis. Since there is no OOD-related information available, we explore making the synthesized virtual OOD map to be different from the original feature map while retaining certain input-related information. In Fig. 5, we can see that the enhanced map E contains plentiful object-related information and less object-irrelevant information. Moreover, after the blurring operation, the synthesized virtual OOD map F_T still involves input-related information. Meanwhile, it is hard to discriminate the corresponding categories. This indicates that our method could synthesize expected virtual OOD features, which improves the performance of detecting OOD objects.

Fig. 6 further shows some ID examples. Compared with F_0, E contains much stronger object-related information. After Gaussian Blur operations, the object-related information in F_T is blurred, facilitating $F_0 - F_T$ to weaken the object-irrelevant information. By concatenation operation and optimization, E is promoted to reduce attention to the object-irrelevant information.

7. Conclusion

In this paper, we convert the challenges of unsupervised OOD-OD to the diffusion problem for feature synthesis and propose a new method, i.e., Deep Feature Deblurring Diffusion. Specifically, the forward process is to gradually perform Gaussian Blur to synthesize expected OOD maps utilized to enhance the ability of detecting OOD objects. The reverse process is to recover the deblurred features continually, which improves the discrimination of the object classifier. Extensive experimental results on three different tasks demonstrate the effectiveness of our method.

Acknowledgement. Our work was supported by Joint Fund of Ministry of Education of China (8091B022149), Key Research and Development Program of Shaanxi (2021ZDLGY01-03), National Natural Science Foundation of China (62102293, 62132016, 62171343, and 62071361), and Fundamental Research Funds for the Central Universities (ZDRC2102).

References

[1] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016.

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

[3] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *WACV*, pages 1021–1030, 2020.

[4] Xuefeng Du, Gabriel Gozum, Yifei Ming, and Yixuan Li. Siren: Shaping representations for detecting out-of-distribution objects. In *NeurIPS*, 2022.

[5] Xuefeng Du, Xin Wang, Gabriel Gozum, and Yixuan Li. Unknown-aware object detection: Learning what you don't know from videos in the wild. In *CVPR*, pages 13678–13688, 2022.

[6] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *ICLR*, 2022.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[8] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *ICML*, pages 2151–2159, 2019.

[9] Pascal Getreuer. A survey of gaussian convolution algorithms. *Image Processing On Line*, 2013:286–310, 2013.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[11] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, pages 10696–10706, 2022.

[12] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *CVPR*, pages 9235–9244, 2022.

[13] Jiaming Han, Yuqiang Ren, Jian Ding, Xingjia Pan, Ke Yan, and Gui-Song Xia. Expanding low-density latent regions for open-set object detection. In *CVPR*, pages 9591–9600, 2022.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

[17] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2019.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

[19] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *CVPR*, pages 10951–10960, 2020.

[20] Minghui Hu, Yujie Wang, Tat-Jen Cham, Jianfei Yang, and Ponnuthurai N Suganthan. Global context with discrete diffusion in vector quantised modelling for image generation. In *CVPR*, pages 11502–11511, 2022.

[21] Taewon Jeong and Heeyoung Kim. Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification. *NeurIPS*, 33:3907–3916, 2020.

[22] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, pages 5830–5840, 2021.

[23] Julian Katz-Samuels, Julia B Nakhleh, Robert Nowak, and Yixuan Li. Training ood detectors in their natural habitats. In *ICML*, pages 10848–10865, 2022.

[24] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 34:21696–21707, 2021.

[25] Joseph Kj, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[26] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International journal of computer vision*, 128(7):1956–1981, 2020.

[27] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *ICLR*, 2018.

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*, 31, 2018.

[29] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *ICLR*, 2017.

[30] 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*, pages 740–755. Springer, 2014.

[31] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *NeurIPS*, 33:21464–21475, 2020.

[32] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

[33] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *NeurIPS*, 31, 2018.

[34] Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. *ICLR*, 2020.

[35] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *ICRA*, pages 3243–3249, 2018.

[36] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern Recognition Letters*, 140:109–115, 2020.

[37] Tal Reiss, Niv Cohen, Eliahu Horwitz, Ron Abutbul, and Yedid Hoshen. Anomaly detection requires better representations. *arXiv preprint arXiv:2210.10773*, 2022.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[42] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *ICML*, pages 8491–8501. PMLR, 2020.

[43] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, pages 3400–3409, 2017.

[44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015.

[45] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *ICML*, pages 20827–20840, 2022.

[46] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 33:11839–11852, 2020.

[47] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[49] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? *NeurIPS*, 34:29074–29087, 2021.

[50] Aming Wu and Cheng Deng. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *CVPR*, pages 847–856, 2022.

[51] Aming Wu and Cheng Deng. Discriminating known from unknown objects via structure-enhanced recurrent variational autoencoder. In *CVPR*, pages 23956–23965, 2023.

[52] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4178–4193, 2021.

[53] Aming Wu, Rui Liu, Yahong Han, Linchao Zhu, and Yi Yang. Vector-decomposed disentanglement for domain-invariant object detection. In *ICCV*, pages 9342–9351, 2021.

[54] Binbin Yang, Xinchi Deng, Han Shi, Changlin Li, Gengwei Zhang, Hang Xu, Shen Zhao, Liang Lin, and Xiaodan Liang. Continual object detection via prototypical task correlation guided gating mechanism. In *CVPR*, pages 9255–9264, 2022.

[55] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[56] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

[57] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2636–2645, 2020.

[58] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *CVPR*, pages 4401–4410, 2021.

[59] Yibo Zhou. Rethinking reconstruction autoencoder-based out-of-distribution detection. *CVPR*, 2022.