MaskDiff: Modeling Mask Distribution with Diffusion Probabilistic Model for Few-Shot Instance Segmentation

Minh-Quan Le[1, 2, 3], **Tam V. Nguyen**[4], **Trung-Nghia Le**[1, 2], **Thanh-Toan Do**[5], **Minh N. Do**[6], **Minh-Triet Tran**[1, 2*]

[1] University of Science, VNU-HCM, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] Stony Brook University, United States
[4] University of Dayton, United States
[5] Monash University, Australia
[6] University of Illinois at Urbana-Champaign, United States
lmquan@selab.hcmus.edu.vn, tamnguyen@udayton.edu, ltnghia@fit.hcmus.edu.vn, toan.do@monash.edu, minhdo@illinois.edu, tmtriet@fit.hcmus.edu.vn

Abstract

Few-shot instance segmentation extends the few-shot learning paradigm to the instance segmentation task, which tries to segment instance objects from a query image with a few annotated examples of novel categories. Conventional approaches have attempted to address the task via prototype learning, known as point estimation. However, this mechanism depends on prototypes (*e.g.* mean of $K-$shot) for prediction, leading to performance instability. To overcome the disadvantage of the point estimation mechanism, we propose a novel approach, dubbed MaskDiff, which models the underlying conditional distribution of a binary mask, which is conditioned on an object region and $K-$shot information. Inspired by augmentation approaches that perturb data with Gaussian noise for populating low data density regions, we model the mask distribution with a diffusion probabilistic model. We also propose to utilize classifier-free guided mask sampling to integrate category information into the binary mask generation process. Without bells and whistles, our proposed method consistently outperforms state-of-the-art methods on both base and novel classes of the COCO dataset while simultaneously being more stable than existing methods. The source code is available at: https://github.com/minhquanlecs/MaskDiff.

Introduction

To achieve outstanding performances, instance segmentation models (He et al. 2017; Le et al. 2022) require to be trained on substantial pixel-level annotated images, which is labor-intensive and costly. Moreover, their ability to generalize from a few examples is still far from acceptable compared to the human visual system, which constrains their feasibility in practical applications. Inspired by the miraculous ability of the human visual system to recognize novel objects with limited data, few-shot learning (FSL) aims to learn new concepts on a handful of training data (K examples) by generalizing models trained on base classes to adapt novel classes.

*Corresponding author

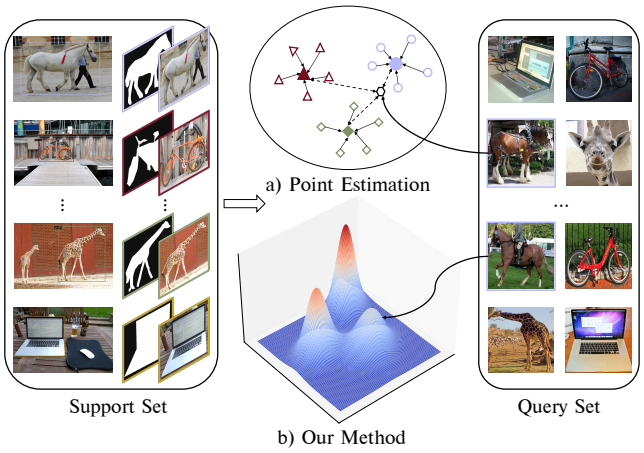

Figure 1: (a) Prior work estimates a specific point as representative of each class (Fan et al. 2020). (b) Our MaskDiff method models the conditional distribution of masks given object regions and $K-$shot samples.

Conventional few-shot instance segmentation (FSIS) methods tried to address learning on a few examples via prototype learning (Fan et al. 2020; Ganea et al. 2021). This matching mechanism searches the nearest prototype and its corresponding support class from support images as guidance for segmenting the query image (Fig.1a). Unlike existing FSIS methods, we introduce a novel probabilistic model to estimate the distribution of a binary mask given a detected object region and K samples (Fig.1b). Concretely, object region, object category, and $K-$shot are treated as conditional information for generating the binary mask representation of each object. Mathematically, we model the conditional distribution of a binary mask of an object instance conditioned on an object region and $K-$shot of a specific i-th class $p_\theta\left(\mathbf{y}_{\text{mask}}|\mathbf{x}_{\text{region}}, K-\text{shot}^i\right)$. To the best of our knowledge, our work is the first method to model mask distribution for FSIS.

Derived from the insight that perturbs data with Gaussian noise for populating low data density regions (Song and Ermon 2019) and motivated by diffusion-based models (Sohl-Dickstein et al. 2015), we define a Markov chain of diffusion steps to slowly add Gaussian noise to an object mask and then learn to reverse the diffusion process conditioned on an object region and $K-$shot samples to reconstruct the desired mask representation corresponding to the objects from the noise. We also design a method using classifier-free guidance to guide the diffusion model during sampling. Hence, the diffusion model is aware of categories of binary masks in the sampling procedure.

Our proposed approach includes several advantages. First, our method is more stable and robust to different $K-$shot samples than point estimation methods since we capture the underlying conditional distribution of mask representation rather than depending on prototypes for prediction (see subsection stability analysis). We hypothesize that the binary mask representation of each object is sampled from a high-dimensional conditional distribution conditioned on RGB image and $K-$shot. Thus, a single prototypical vector is insufficient to comprehensively capture diverse levels of semantic information, including object boundaries, poses, or categories. In addition, we argue that the mask representation relies on not only the instance object region but also its category. Therefore, we follow class-specific mask predictor and integrate category knowledge into the mask generation process. Last but not least, our class-specific mask predictor alleviates spatial information losses from pooling operators of existing FSIS methods (Ganea et al. 2021; Nguyen et al. 2022). When exploiting detected bounding box locations as input for diffusion models, we do not leverage any pooled features such as Mask-RCNN (He et al. 2017) to generate masks but directly use image channels, thus preserving detailed spatial information. Our contributions lie in four-fold:

- We introduce a novel mask distribution modeling approach, dubbed **MaskDiff**, for FSIS. Conceptually, unlike point estimation, MaskDiff tries to model the conditional distribution of a binary mask conditioned on the detected object region and $K-$shot samples.
- Our work is the first to adapt conditional diffusion probabilistic models for modeling instance binary mask distribution in FSIS.
- We propose to utilize guidance from the available classification head of the object detector to integrate category information into the mask sampling procedure. Consequently, in the reverse process, the diffusion model is aware of the categories of mask representation. Our class-specific mask predictor outperforms class-agnostic ones (Ganea et al. 2021; Nguyen et al. 2022). Moreover, when we integrate category information into the mask-generation process, the sampling procedure becomes more stable and produces more organized content (see Supplementary Materials for qualitative results).
- Extensive experimental results on the standard COCO dataset show that our MaskDiff not only outperforms state-of-the-art FSIS methods on both base and novel classes but also is more stable than existing methods.

Related Work

Few-shot instance segmentation (FSIS). The majority of prior works for FSIS try to provide guidance to specific components of Mask-RCNN (He et al. 2017) to guarantee that the networks better understand novel classes or both base and novel classes as well. Most early approaches adapted meta-learning to episodic training (Yan et al. 2019; Fan et al. 2020). On the other hand, Wang *et al*. (2020) proposed a two-stage fine-tuning approach (TFA) which first trains Faster-RCNN (Ren et al. 2015) on the base classes and then fine-tunes the box predictor on a balanced subset of base and novel classes. Modern FSIS methods (Wang et al. 2020; Ganea et al. 2021; Nguyen et al. 2022) were then developed in this fine-tuning direction by fine-tuning the last layers of certain heads on novel classes. Our proposed MaskDiff also follows the latter training strategy. In addition, current techniques (Ganea et al. 2021; Nguyen et al. 2022) mainly concentrate on class-agnostic mask prediction heads, meaning each object's mask representation does not depend on its category. On the contrary, we propose to leverage object classes to generate semantic and stable masks.

Diffusion probabilistic model (DPM). Diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) belong to a class of likelihood-based generative models that comprise the forward process and the reverse process. Later on, Nichol and Dhariwal (2021) design an additional classifier which is trained on noisy images and utilizes gradients to guide the sampling process based on the conditioning information. In contrast, Ho and Salimans (2021) argued that a pure generative model could provide guidance without needing a classifier. Inspired by their work, we employ classifier-free guided mask sampling, using the off-the-shelf cosine classifier from the box-classification head instead of training additional classifiers. Recent works have studied the adaptation of conditional DPMs to downstream tasks including super-resolution (Rombach et al. 2022), text-to-image synthesis (Gu et al. 2022), and image segmentation (Baranchuk et al. 2022). To our best knowledge, our proposed MaskDiff is the first work adapting conditional DPM to FSIS.

Methodology

Problem Formulation

In few-shot learning, we have a disjoint set of base classes $\mathbf{C}_{\text{base}}$ which contains a large quantity of training data and a set of novel classes $\mathbf{C}_{\text{novel}}$ with a limited number of annotated data $\mathbf{C}_{\text{base}} \cap \mathbf{C}_{\text{novel}} = \emptyset$. The main objective is to train a model that performs well on the novel classes $\mathbf{C}_{\text{test}} = \mathbf{C}_{\text{novel}}$ or on both base and novel classes together $\mathbf{C}_{\text{test}} = \mathbf{C}_{\text{base}} \cup \mathbf{C}_{\text{novel}}$. Following the steps of few-shot classification, prior works (Kang et al. 2019; Yan et al. 2019) simulate the *episodic-training* methodology which randomly samples a series of episodes $\mathcal{E} = \left\{\left(\mathbf{I}_j^q, \mathcal{S}_j\right)\right\}_{j=1}^{|\mathcal{E}|}$ where the j-th episode is formulated by sampling a support set $\mathcal{S}_j$ including N classes from $\mathbf{C}_{\text{train}} = \mathbf{C}_{\text{base}} \cup \mathbf{C}_{\text{novel}}$ in tandem with K samples per class ($N-$way, $K-$shot) and a query image $\mathbf{I}_j^q$. The goal of FSIS is not only to localize all the object instances from any query image and classify those objects but also to determine their segmentation

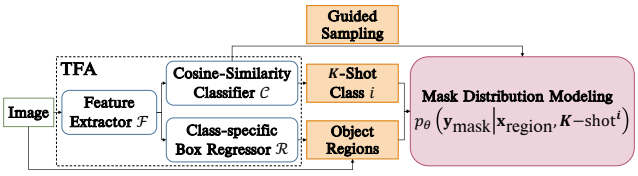

Figure 2: Our MaskDiff is built upon TFA (Wang et al. 2020) by integrating a mask distribution modeling head and adapting guided sampling to integrate category information into mask generation procedure.

masks. For all objects in a query image $\mathbf{I}^q$ that belong to $\mathbf{C}_{\text{test}}$, FSIS generates the corresponding labels, bounding boxes, and segmentation masks.

Proposed MaskDiff Method

Overview. Figure 2 illustrates the architecture of our MaskDiff, adapted from TFA (Wang et al. 2020), which is a two-stage object detection architecture. We build a mask distribution modeling head on top of TFA. Unlike class-agnostic mask predictor (Ganea et al. 2021) or prototype-based mask head (Fan et al. 2020), we propose to model the conditional distribution of a binary mask conditioned on an instance region and $K-$shot samples via conditional DPMs. With regard to denoising architecture of the mask distribution modeling branch, we adapt UNet (Ronneberger, Fischer, and Brox 2015) architecture with attention heads (Vaswani et al. 2017) from Guided Diffusion (Dhariwal and Nichol 2021). In particular, the conditioning module consists of concatenating a detected object region and $K-$shot. We empirically found that the permutation of $K-$shot does not affect the model's performance since the encoder-decoder architecture leverages information of each shot independently. The detailed denoising architecture is illustrated in Supplementary Material (Supp.).

Diffusion-based two-stage training. Figure 3 visualizes our diffusion-based two-stage training for FSIS. In the first stage, *i.e.*, the base training stage, the network is trained only on the base classes $\mathbf{C}_{\text{base}}$. We separately train few-shot object detector heads and estimate the mask distribution. Specifically, the RoI cosine-similarity classifier $\mathcal{C}$ and the box regressor $\mathcal{R}$ follow the standard training process while the mask distribution modeling head $p_\theta\left(\mathbf{y}_{\text{mask}}|\mathbf{x}_{\text{region}}, K\text{−shot}^i\right)$ is proposed to train based on conditional DPM. The reason is that in this early stage, the box regressor $\mathcal{R}$ is not stable enough, and it is complicated for probabilistic models to estimate the mask distribution efficiently. Obviously, incorrectly localizing objects leads to extremely unsatisfactory segmentation results.

The second stage, *i.e.*, the few-shot fine-tuning stage, involves freezing the entire feature extractor $\mathcal{F}$ and jointly fine-tuning prediction heads on a balanced dataset of K shots for both $\mathbf{C}_{\text{base}}$ and $\mathbf{C}_{\text{novel}}$ classes. Likewise, all the prediction heads $\mathcal{C}$, $\mathcal{R}$, and p_θ are fine-tuned.

Instance Mask Modeling with Conditional DPM

Given an FSIS dataset, we extract a dataset of input-output pairs denoted $\mathcal{D} = \left\{\left(\mathbf{x}^d_{\text{region}}, K\text{−shot}; \mathbf{y}^d_{\text{mask}}\right)\right\}_{d=1}^{|\mathcal{D}|}$ for

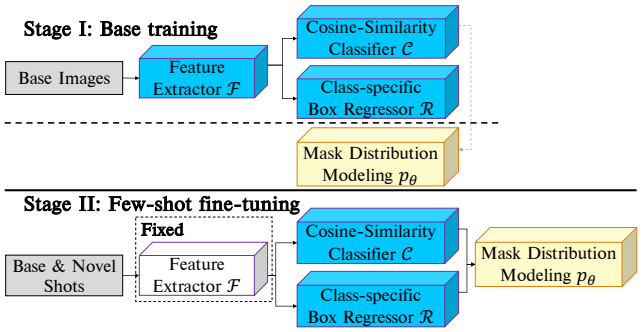

Figure 3: Pipeline of diffusion-based two-stage training. In the first stage, the entire object detector (feature extractor, classifier, and box regressor) and the mask distribution modeling head are trained separately on the base classes. In the second stage, the feature extractor is frozen while the cosine classifier, the box regressor, and the mask distribution modeling head are jointly fine-tuned on both base and novel classes. Our mask distribution modeling head is trained based on diffusion (yellow), while the remaining two heads follow the standard object detection training (blue).

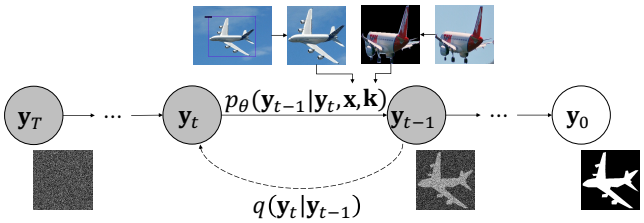

Figure 4: Conditional diffusion model for few-shot binary mask segmentation. The directed graphical model transforms the noise from standard Gaussian distribution to the mask representation of objects through an iterative denoising process. Solid arrows indicate the reverse process, while the dashed arrow implies the forward step. $\mathbf{y}, \mathbf{x}, \mathbf{k}$ denote binary masks, object regions and removed background version of $K-$shot samples, respectively.

modeling the binary mask distribution given object regions and $K-$shot information. $\mathbf{x}^d_{\text{region}}$'s are collected by cropping object regions from the RGB images by bounding box locations in the annotations. $\mathbf{y}^d_{\text{mask}}$'s are converted from polygons to binary mask representations at the same location as $\mathbf{x}^d_{\text{region}}$'s. Finally, $K-$shot samples of each class are the background removed version of the object region. For simplicity, we denote $\mathbf{y}, \mathbf{x}, \mathbf{k}$ to represent binary masks, object regions, and $K-$shot guidance, respectively. In this work, we aim to learn the underlying conditional distribution from which a data point representing a binary mask is sampled $\mathbf{y}_0 \sim q(\mathbf{y}|\mathbf{x}, \mathbf{k})$. However, we are unable to determine exactly the true distribution. Via diffusion models, we maximize the likelihood $p_\theta\left(\mathbf{y}_0|\mathbf{x}, \mathbf{k}\right) = \int p_\theta\left(\mathbf{y}_{0:T}|\mathbf{x}, \mathbf{k}\right) d\mathbf{y}_{1:T}$ to approximate the true distribution. Two processes are defined in the conditional diffusion probabilistic models, including the forward and reverse processes (see Fig. 4).

Algorithm 1: Training Procedure

repeat
$\mathbf{x} \sim q(\mathbf{x})$, $\mathbf{x}$ belongs to class i, $\mathbf{k} = K$ shots of class i
$\mathbf{y}_0 \sim q(\mathbf{y}_0|\mathbf{x}, \mathbf{k})$
$t \sim \text{Uniform}(1, \ldots, T)$
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\mathbf{y}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \mathbf{x}, \mathbf{k}, t \right) \right\|^2$$

until converged

Forward diffusion process. The forward diffusion process is defined in which we gradually add a small amount of Gaussian noise to the sample in T steps, producing a sequence of noisy samples $\mathbf{y}_1, \ldots, \mathbf{y}_T$, formulated as follows:

$$q\left(\mathbf{y}_{1:T}|\mathbf{y}_0\right) = \prod_{t=1}^{T} q\left(\mathbf{y}_t|\mathbf{y}_{t-1}\right). \tag{1}$$

Reverse diffusion process. The reverse diffusion process $p_\theta\left(\mathbf{y}_{0:T}|\mathbf{x}, \mathbf{k}\right)$ is defined as a Markov chain with learned Gaussian transitions beginning with $p(\mathbf{y}_T) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is formulated as follows:

$$p_\theta\left(\mathbf{y}_{0:T}|\mathbf{x}, \mathbf{k}\right) = p\left(\mathbf{y}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x}, \mathbf{k}\right). \tag{2}$$

Diffusion loss. The conditional diffusion probabilistic model is trained to minimize the cross entropy as the learning objective, which is equivalent to minimizing variational upper bound L_{VUB} (see Supp. for detailed derivation of loss functions). The loss L_{VUB} can be rewritten to be a combination of several KL-divergence and entropy terms:

$$L_{\text{VUB}} = \mathbb{E}_q \Bigg[\underbrace{D_{\text{KL}}(q(\mathbf{y}_T|\mathbf{y}_0) \parallel p_\theta(\mathbf{y}_T))}_{L_T} \underbrace{- \log p_\theta(\mathbf{y}_0|\mathbf{y}_1, \mathbf{x}, \mathbf{k})}_{L_0} + \sum_{t=1}^{T-1} \underbrace{D_{\text{KL}}(q(\mathbf{y}_t|\mathbf{y}_{t+1}, \mathbf{y}_0) \parallel p_\theta(\mathbf{y}_t|\mathbf{y}_{t+1}, \mathbf{x}, \mathbf{k}))}_{L_t} \Bigg]. \tag{3}$$

Following the standard training process of DPM (Ho, Jain, and Abbeel 2020), the loss term L_t is parameterized to achieve better training, resulting in a simplified loss:

$$L_t = \mathbb{E}_{\mathbf{y}_0, \boldsymbol{\epsilon}} \Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{y}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \mathbf{x}, \mathbf{k}, t)\|^2 \Big], \tag{4}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Training procedure is shown in Alg. 1.

Classifier-free guided mask sampling. In contrast to class-agnostic mask segmentation (Ganea et al. 2021), we integrate category knowledge into binary mask generation. It is known that mask representation of each object depends upon not only its boundary but also its category. The mask distribution modeling head can be aware of the discriminative properties of objects. In other words, our diffusion model explicitly controls binary masks via object labels.

Inspired by the derivation of score-based models (Song and Ermon 2019), the distribution $p(\mathbf{y}_t|\mathbf{x}, \mathbf{k}, \mathbf{c})$, where $\mathbf{c}$ is a one-hot vector indicating the object category, has the score:

$$\begin{aligned} \nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}, \mathbf{c}\right) &= \nabla \log \left(\frac{p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}\right) p\left(\mathbf{c}|\mathbf{y}_t, \mathbf{x}, \mathbf{k}\right)}{p(\mathbf{c}|\mathbf{x}, \mathbf{k})} \right) \\ &= \underbrace{\nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}\right)}_{\text{guidance-agnostic score}} + \underbrace{\nabla \log p\left(\mathbf{c}|\mathbf{y}_t, \mathbf{x}, \mathbf{k}\right)}_{\text{adversarial gradient}}. \end{aligned} \tag{5}$$

Classifier guidance (Dhariwal and Nichol 2021) adjusts the adversarial gradient of the noisy classifier by a ω hyperparameter term to introduce fine-grained control to either encourage or dissuade the model from accepting the conditioning information:

$$\nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}, \mathbf{c}\right) = \nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}\right) + \omega \nabla \log p\left(\mathbf{c}|\mathbf{y}_t, \mathbf{x}, \mathbf{k}\right). \tag{6}$$

Inspired by classifier-free guidance (Ho and Salimans 2021), we substitute the adversarial gradient term in Eq. 5 into Eq. 6, resulting:

$$\nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}, \mathbf{c}\right) = \underbrace{\omega \nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}, \mathbf{c}\right)}_{\text{guidance-specific score}} + \underbrace{(1-\omega) \nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}\right)}_{\text{guidance-agnostic score}}. \tag{7}$$

As $\nabla \log p\left(\mathbf{y}_t|\mathbf{x}, \mathbf{k}, \mathbf{c}\right) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{y}_t, \mathbf{x}, \mathbf{k}, \mathbf{c}, t)$ (see Supp. for more details), we have the equivalent form:

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{y}_t, \mathbf{x}, \mathbf{k}, \mathbf{c}, t) = \underbrace{\omega \boldsymbol{\epsilon}_\theta(\mathbf{y}_t, \mathbf{x}, \mathbf{k}, \mathbf{c}, t)}_{\text{guidance-specific score}} + \underbrace{(1-\omega) \boldsymbol{\epsilon}_\theta(\mathbf{y}_t, \mathbf{x}, \mathbf{k}, t)}_{\text{guidance-agnostic score}}. \tag{8}$$

The inference procedure with guided sampling is shown in Alg. 2. When $\omega > 1$, the diffusion model not only favors the guidance-specific score function over the guidance-agnostic one but also moves away from it. Along with producing samples that accurately match the conditioning information, this also reduces sample variety and makes generated masks more stable. Following Ho *et al*. (2021), we set $\omega = 5$ to obtain the most individual sample fidelity.

Algorithm 2: Inference Procedure with Guided Sampling

$\mathbf{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, otherwise $\boldsymbol{\epsilon} \leftarrow \mathbf{0}$
$\hat{\boldsymbol{\epsilon}}_\theta \leftarrow \omega \boldsymbol{\epsilon}_\theta(\mathbf{y}_t, \mathbf{x}, \mathbf{k}, \mathbf{c}, t) + (1-\omega) \boldsymbol{\epsilon}_\theta(\mathbf{y}_t, \mathbf{x}, \mathbf{k}, t)$
$\mathbf{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{y}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\boldsymbol{\epsilon}}_\theta \right) + \sigma_t \boldsymbol{\epsilon}$
end for
return $\mathbf{y}_0$

Experiments

Implementation Details

Our MaskDiff was implemented using the Detectron2 library. The used backbone architecture was a ResNet50 (He et al. 2016), and we utilized FPN (Lin et al. 2017) for the path aggregation block. The mask distribution modeling head was implemented using Guided Diffusion (Dhariwal and Nichol 2021) with $1,000$ reverse steps. In the base training stage, similar to TFA (Wang et al. 2020), object detector heads were trained using SGD with a learning rate of 0.02, momentum of 0.9, and weight decay of 10^{-4}. The mask distribution modeling head was trained using AdamW

Shots	Method	Published	Object Detection						Instance Segmentation					
			All Classes		Base Classes		Novel Classes		All Classes		Base Classes		Novel Classes	
			AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}
1	MRCN+ft-full (2017)	ICCV 2017	10.21	21.58	17.63	26.32	0.74	2.33	9.88	19.25	15.57	24.18	0.64	2.14
	MTFA (2021)	CVPR 2021	24.32	39.64	31.73	51.49	2.10	4.07	22.98	37.48	29.85	48.64	2.34	3.99
	iMTFA (2021)	CVPR 2021	21.67	31.55	27.81	40.11	3.23	5.89	20.13	30.64	25.9	39.28	2.81	4.72
	iFS-RCNN (2022)	CVPR 2022	31.19	52.83	40.08	71.14	4.54	10.29	28.45	46.72	36.35	63.11	3.95	7.89
	MaskDiff (Ours)	-	**32.59**	**54.61**	**41.23**	**72.85**	**5.26**	**11.35**	**29.42**	**48.37**	**37.59**	**64.26**	**4.85**	**8.73**
5	MRCN+ft-full (2017)	ICCV 2017	12.31	23.69	19.43	28.12	1.15	2.09	11.28	22.36	17.89	26.78	1.17	2.58
	MTFA (2021)	CVPR 2021	26.39	41.52	33.11	51.49	6.22	11.63	25.07	39.95	31.29	49.55	6.38	11.14
	iMTFA (2021)	CVPR 2021	19.62	28.06	24.13	33.69	6.07	11.15	18.22	27.10	22.56	33.25	5.19	8.65
	iFS-RCNN (2022)	CVPR 2022	32.52	54.30	40.06	71.19	9.91	19.24	29.89	48.22	36.33	62.81	8.80	15.73
	MaskDiff (Ours)	-	**33.45**	**56.19**	**41.51**	**72.28**	**10.49**	**21.07**	**30.48**	**50.35**	**38.12**	**64.36**	**9.43**	**17.12**
10	MRCN+ft-full (2017)	ICCV 2017	12.44	24.39	20.57	29.72	2.33	5.64	12.15	23.29	18.08	27.53	1.86	4.25
	MTFA (2021)	CVPR 2021	27.44	42.84	33.83	52.04	8.28	15.25	25.97	41.28	31.84	50.17	8.36	14.58
	iMTFA (2021)	CVPR 2021	19.26	27.49	23.36	32.41	6.97	12.72	17.87	26.46	21.87	32.01	5.88	9.81
	iFS-RCNN (2022)	CVPR 2022	33.02	56.15	40.05	69.84	12.55	25.97	30.41	49.54	36.32	63.29	10.06	19.72
	MaskDiff (Ours)	-	**35.21**	**59.80**	**42.17**	**72.36**	**14.04**	**28.33**	**31.89**	**52.15**	**38.55**	**66.48**	**11.84**	**21.27**

Table 1: FSOD and FSIS performance on COCO dataset for both base and novel classes (COCO-All). MaskDiff outperforms state-of-the-art methods. The best performance is marked in boldface.

Shots	Method	Detection		Segmentation	
		AP	AP_{50}	AP	AP_{50}
10	MRCN+ft-full (2017)	2.52	5.78	1.93	4.68
	Meta-RCNN (2019)	5.60	14.20	4.40	10.60
	MTFA (2021)	8.52	15.53	8.40	14.62
	iMTFA (2021)	7.14	12.91	5.94	9.96
	iFS-RCNN (2022)	11.27	22.15	10.22	20.61
	MaskDiff (Ours)	**12.18**	**23.61**	**11.01**	**21.58**

Table 2: FSOD and FSIS performance on only COCO novel classes (COCO-Novel). MaskDiff outperforms state-of-the-art methods. Results on other $K-$shot are shown in Supp.

Shots	Method	Detection		Segmentation	
		AP	AP_{50}	AP	AP_{50}
1	FGN (2020)	-	30.80	-	16.20
	MTFA (2021)	9.99	21.68	9.51	19.28
	iMTFA (2021)	11.47	22.41	8.57	16.32
	MaskDiff (Ours)	**24.15**	**38.57**	**22.73**	**37.59**

Table 3: FSIS performance on COCO2VOC. MaskDiff outperforms state-of-the-art methods. The authors of FGN (Fan et al. 2020) reported only AP50 results of one-shot.

with a learning rate of 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Object regions, binary masks, and $K-$shot guidance were all resized to 128×128 for training the mask distribution modeling head. In the few-shot fine-tuning stage, the entire network, including object detector heads and the mask distribution modeling head, was jointly trained with the same configuration as training the mask distribution modeling head in the first stage. In the final step of the denoising process, we set the threshold of noisy binary masks to 0.5 to separate 0 and 1 values. All variants of MaskDiff were trained with a batch size of 8 on a single NVIDIA RTX 3090 GPU.

Experimental Settings

Following the standard FSIS evaluation (Wang et al. 2020; Ganea et al. 2021), we evaluated MaskDiff on COCO (Lin et al. 2014), VOC2007 and VOC2012 (Everingham et al. 2010) datasets. 80 classes of COCO were split as suggested by Kang *et al*. (2019). 20 classes that intersect with VOC were used as novel classes, whereas the remaining 60 classes were used as base classes. The union of COCO training set ($80k$ images) and COCO validation set ($35k$ images) was utilized for training, while the remaining $5k$ images served as the test set. We also combined validation sets of VOC2007 and VOC2012 only for testing.

We compared the performance of $K = 1, 5, 10$ shots per novel class. We repeated all tests 10 times with K random samples per class to limit the influence of outliers caused by the random selection of K shots and report the mean result. Our FSIS assessment approach is identical to that of few-shot object detection (FSOD) (Wang et al. 2020). Crucially, segmentation results of different runs are not unique since the masks sampled in the inference depend on random factors ($\mathbf{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$). To provide a stable and reliable evaluation procedure of generative-based models, we propose a simple yet effective evaluation inspired by TFA (Wang et al. 2020). With a model trained on specific K shots, we performed inference for multiple runs on different random seeds to obtain averages and confidence intervals. We observed that 10 runs are comparatively reliable and stable for evaluation reports. To sum up, we trained models for 10 runs on different samples of training shots, and with each trained model, we tested on 10 different random seeds.

Comparison with State-of-the-art Methods

Results on both COCO base and novel classes (COCO-All). In this experiment, we aim to predict all 80 COCO classes and report the standard evaluation metrics, including mAP and AP50. Following the newly introduced evaluation process of Ganea *et al*. (2021), we report the performance of the base and novel classes independently as well as all classes. We compared our MaskDiff against state-of-the-art methods, such as MTFA, iMTFA (Ganea et al. 2021), iFS-RCNN (Nguyen et al. 2022), and a fully-converged Mask

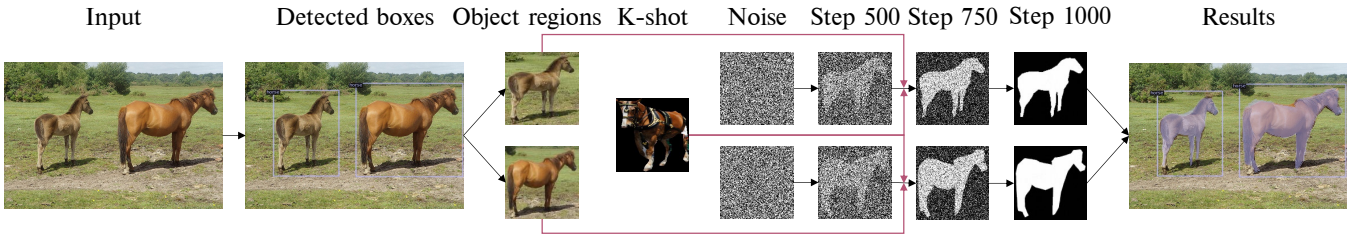

Figure 5: Inference example of our MaskDiff when training and inference on one-shot setting for the COCO novel classes.

Method	Training (Hours)		Inference (FPS)
	Base	Fine-tuning	
MRCN+ft-full	24.6	12.8	10.38
MTFA	24.8	12.7	10.32
iMTFA	**24.2**	**12.4**	**11.75**
iFS-RCNN	28.5	15.6	10.28
MaskDiff	32.3	16.1	8.27

Table 4: Runtime of FSIS ($1-$shot) on COCO dataset.

R-CNN model fine-tuned on the novel classes (MRCN+ft-full (He et al. 2017)). Results in Tab. 1 show MaskDiff consistently outperforms state-of-the-art methods on both FSOD and FSIS tasks for all numbers of provided shots on both base and novel classes. This implies MaskDiff can adapt to novel classes while embracing performance in base classes. Especially, MaskDiff outperforms MTFA and iMTFA with significant margins on the COCO dataset. Compared with MTFA on 10 shots, our performance of FSIS gains is about $+3.5$ on new classes and $+7$ on base classes. Likewise, our MaskDiff also dramatically outperforms iMTFA with gains of $+6$ on novel classes and $+17$ on base classes. Modeling mask distribution via DPM is more effective than conventional methods and gains large performance in learning existing and new categories on limited data. Regarding FSOD, our MaskDiff is built upon TFA, a multi-task learning design, i.e., detection and segmentation. The improvement in one task can benefit another one. As our segmentation head generates a better binary mask for each region. This information can be used to improve the accuracy of object detection by providing a more precise location and shape of the object.

Results on only COCO novel classes (COCO-Novel). Table 2 reports our results on only COCO novel classes, in which we compared MaskDiff against SOTA FSIS methods, *e.g.*, Meta-RCNN (Yan et al. 2019), MTFA (Ganea et al. 2021), iMTFA (Ganea et al. 2021), iFS-RCNN (Nguyen et al. 2022), and MRCN+ft-full (He et al. 2017). Experimental results demonstrate the superiority of MaskDiff on both detection and segmentation tasks consistently on all $K-$shots. MaskDiff surpasses the recent iFS-RCNN and drastically outperforms other methods by a large margin in terms of AP. This indicates that our proposal is better than previous strategies including episodic-training (Meta-RCNN), class-specific mask predictor (MTFA), and class-agnostic one (iMTFA, iFS-RCNN).

Shots	MaskDiff		Detection		Segmentation	
	Guided	Two-Stage	AP	AP_{50}	AP	AP_{50}
10	✗	✗	10.59	20.96	9.68	18.83
	✓	✗	10.86	21.55	10.02	19.48
	✗	✓	12.05	23.32	10.76	20.63
	✓	✓	12.18	23.61	11.01	21.58

Table 5: Our ablation study on COCO-Novel dataset. MaskDiff with classifier-free guided mask sampling and diffusion-based two-stage training strategy performs best. Results on other $K-$shot are shown in Supp.

Cross-dataset COCO2VOC evaluation. To investigate the ability to learn new categories, we compared MaskDiff with some SOTA FSIS methods (*e.g.*, FGN (Fan et al. 2020), MTFA (Ganea et al. 2021), and iMTFA (Ganea et al. 2021)) on cross-dataset COCO2VOC setting. Methods were trained on COCO dataset and evaluated on VOC dataset. We reported public results provided by authors, which are only in one-shot setting ($K = 1$). Table 3 shows that our MaskDiff transcends cutting-edge methods on cross-dataset evaluation. More importantly, our mask distribution modeling approach outperforms point estimation one (Fan et al. 2020).

Qualitative evaluation. Figure 5 demonstrates an inference example of our MaskDiff when training and inference on the one-shot setting for the COCO novel classes. More successful and failed cases can be found in Supp.

Runtime evaluation. Regarding the training and inference time, Table 4 shows that other methods can be trained and can infer faster than ours. However, we remark that for few-shot learning tasks, effectiveness (AP) should be prioritized rather than efficiency (processing time).

Ablation Study

Effectiveness of two-stage training strategy. We aim to evaluate the effectiveness of the proposed two-stage training strategy consisting of base training and few-shot fine-tuning. In base training, we first train the object detector and then $K-$shot mask modeling on base images. In the fine-tuning stage, we jointly train both object detector heads and our mask distribution modeling head on balanced base and novel classes. We compared our approach with the common approach using only a few-shot fine-tuning stage. Particularly, we exclude the first stage training on base images (*i.e.*, base training) and utilize only the second stage. In early steps, the object detector cannot precisely localize and classify the objects. Given unsatisfactory localized object regions, the

Shots	MaskDiff	Segmentation	
		AP	AP_{50}
1	w/o guided sampling	6.04 ± 0.09	12.29 ± 0.12
	w/ guided sampling	$6.23 \pm$ **0.07**	$12.47 \pm$ **0.08**

Table 6: We train our MaskDiff on a specific set of training samples and perform inference on 10 different random seeds. Then, we compute the mean and standard deviation (std) AP of those runs. MaskDiff with guided sampling not only outperforms the one without guidance but also is more stable (less std). Qualitative results of classifier-free guided mask sampling and other $K-$shot can be seen in Supp.

MaskDiff	All Classes	Base Classes	Novel Classes
w/ basic UNet	24.16 ± 0.22	31.49 ± 0.33	3.32 ± 0.35
w/ diffusion	**29.42** $\pm$ **0.09**	**37.59** $\pm$ **0.0093**	**4.85** $\pm$ **0.24**

Table 7: Stability and effectiveness of our MaskDiff with and without diffusion on COCO-All dataset ($1-$shot).

mask distribution modeling head is much more difficult to train, leading to generating not good segmentation masks. Table 5 shows a considerable collapse in results.

Significance of classifier-free guided mask sampling. We studied the effectiveness as well as stability of classifier-free guided mask sampling by comparing the performance MaskDiff with and without classifier-free guided mask sampling. From Table 5, we conclude that classifier-free guided mask sampling boosts the performance of FSIS. Furthermore, we evaluated the standard deviation of testing results on 10 different random seeds when running on a specific sample of training shots. In Table 6, we can see that the standard deviation of MaskDiff with guidance over 10 different random seeds is less than that of without guidance. In other words, the classifier-free guided mask sampling makes the mask distribution modeling head more stable.

Effectiveness of diffusion probabilistic model. We replace our diffusion with the same UNet (without noise) to verify the effectiveness of our diffusion head. Table 7 shows that replacing our diffusion with the basic UNet downgrades the effectiveness as well as stability of our method significantly, consolidating the contribution of our mask distribution modeling approach via DPM.

Stability analysis. The performance of various methods has been generally evaluated based on the evaluation protocol proposed by Wang *et al*. (2020). In particular, a model is trained on a different set of training samples and is evaluated on the same test set to report the mean result. We further computed the standard deviation of different runs and compared the stability with state-of-the-art methods. Table 8 shows that MaskDiff achieves the highest performance and the lowest standard deviation in terms of AP, indicating MaskDiff is the most stable and robust to different $K-$shot.

Spatial information preservation. Most conventional FSIS methods leverage pooled object features and feed them into FCN (Long, Shelhamer, and Darrell 2015) to generate binary masks. However, passing through several pooling layers leads to a severe collapse in spatial information, especially the detailed one. By directly utilizing object regions

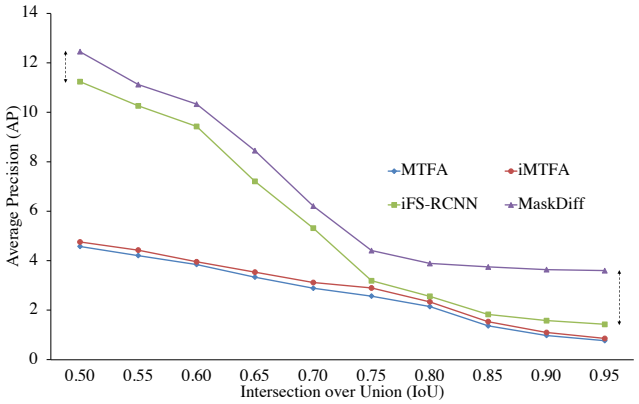

Figure 6: MaskDiff preserves spatial information, especially at detailed levels. We compare the performance of MaskDiff with state-of-the-art methods in one-shot instance segmentation at different IoU thresholds. MaskDiff outperforms other methods with large margins at high IoU thresholds, which indicates its ability to segment objects more precisely.

$1-$**Shot**	All Classes	Base Classes	Novel Classes
MRCN+ft-full	9.88 ± 0.31	15.57 ± 0.30	0.64 ± 0.26
MTFA	22.98 ± 0.24	29.85 ± 0.35	2.34 ± 0.31
iMTFA	20.13 ± 0.28	25.90 ± 0.32	2.81 ± 0.37
iFS-RCNN	28.45 ± 0.12	36.35 ± 0.01	3.95 ± 0.48
MaskDiff	$29.42 \pm$ **0.09**	$37.59 \pm$ **0.0093**	$4.85 \pm$ **0.24**

Table 8: Stability of FSIS on COCO-All dataset. MaskDiff is the most stable method with the smallest standard deviation of AP. The best performance is marked in boldface. Results on other $K-$shot are described in Supp.

from input images to generate binary masks, our MaskDiff can embrace comprehensive spatial details and induce precise delineation. In Fig. 6, we visualize the line graph representing AP from 0.50 to 0.95 with step size 0.05 to illustrate the changes in the performance of FSIS methods when the IoU requirement rises. Our MaskDiff surpasses cutting-edge methods, especially at high IoU thresholds.

Conclusion

We presented the first mask distribution modeling approach for FSIS and designed a novel method dubbed MaskDiff. We adapted conditional DPM to model binary mask distribution conditioned on RGB object regions and $K-$shot samples. Furthermore, we leveraged the off-the-shelf classifier branch to guide the mask sampling procedure with the category information. Experimental results demonstrated that our proposed method achieves state-of-the-art results while being more stable than existing methods.

Acknowledgments

This research was funded by Vingroup and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2019.DA19. The project was also supported by National Science Foundation Grant (NSF#2025234).

References

Baranchuk, D.; Voynov, A.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2022. Label-Efficient Semantic Segmentation with Diffusion Models. In *ICLR*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. In *NeurIPS*, volume 34, 8780–8794.

Everingham, M.; Gool, L. V.; Williams, C. K. I.; Winn, J. M.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88: 303–338.

Fan, Z.; Yu, J.-G.; Liang, Z.; Ou, J.; Gao, C.; Xia, G.-S.; and Li, Y. 2020. FGN: Fully Guided Network for Few-Shot Instance Segmentation. In *CVPR*.

Ganea, D. A.; Boom, B.; Poppe, R.; and Abc, X. 2021. Incremental Few-Shot Instance Segmentation. In *CVPR*, 1185–1194.

Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B. 2022. Vector Quantized Diffusion Model for Text-to-Image Synthesis. In *CVPR*.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask R-CNN. In *ICCV (ICCV)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *NeurIPS*, volume 33, 6840–6851.

Ho, J.; and Salimans, T. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS Workshops*.

Kang, B.; Liu, Z.; Wang, X.; Yu, F.; Feng, J.; and Darrell, T. 2019. Few-Shot Object Detection via Feature Reweighting. In *ICCV*.

Le, T.-N.; Cao, Y.; Nguyen, T.-C.; Le, M.-Q.; Nguyen, K.-D.; Do, T.-T.; Tran, M.-T.; and Nguyen, T. V. 2022. Camouflaged Instance Segmentation In-The-Wild: Dataset, Method, and Benchmark Suite. *IEEE Transactions on Image Processing*, 31: 287–300.

Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature Pyramid Networks for Object Detection. In *CVPR*.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

Nguyen, K.; Todorovic, S.; Abc, X.; and Abc, X. 2022. iFS-RCNN: An Incremental Few-Shot Instance Segmenter. In *CVPR*, 7010–7019.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved Denoising Diffusion Probabilistic Models. In *ICML*, volume 139, 8162–8171.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, volume 28.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *CVPR*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 234–241.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *ICML*, volume 37, 2256–2265.

Song, Y.; and Ermon, S. 2019. Generative Modeling by Estimating Gradients of the Data Distribution. In *NeurIPS*, volume 32.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*, volume 30. Curran Associates, Inc.

Wang, X.; Huang, T.; Gonzalez, J.; Darrell, T.; and Yu, F. 2020. Frustratingly Simple Few-Shot Object Detection. In *ICML*, volume 119, 9919–9928.

Yan, X.; Chen, Z.; Xu, A.; Wang, X.; Liang, X.; and Lin, L. 2019. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *ICCV*, 9577–9586.