Concurrent Misclassification and Out-of-Distribution Detection for Semantic Segmentation via Energy-Based Normalizing Flow

Denis Gudovskiy[1] **Tomoyuki Okuno**[2] **Yohei Nakata**[2]

[1]Panasonic AI Lab, Mountain View, CA, USA
[2]Panasonic Holdings Corporation, Osaka, Japan

Abstract

Recent semantic segmentation models accurately classify test-time examples that are similar to a training dataset distribution. However, their discriminative closed-set approach is not robust in practical data setups with distributional shifts and out-of-distribution (OOD) classes. As a result, the predicted probabilities can be very imprecise when used as confidence scores at test time. To address this, we propose a generative model for concurrent in-distribution misclassification (IDM) and OOD detection that relies on a normalizing flow framework. The proposed flow-based detector with an energy-based inputs (FlowEneDet) can extend previously deployed segmentation models without their time-consuming retraining. Our FlowEneDet results in a low-complexity architecture with marginal increase in the memory footprint. FlowEneDet achieves promising results on Cityscapes, Cityscapes-C, FishyScapes and SegmentMeIfYouCan benchmarks in IDM/OOD detection when applied to pretrained DeepLabV3+ and SegFormer semantic segmentation models.

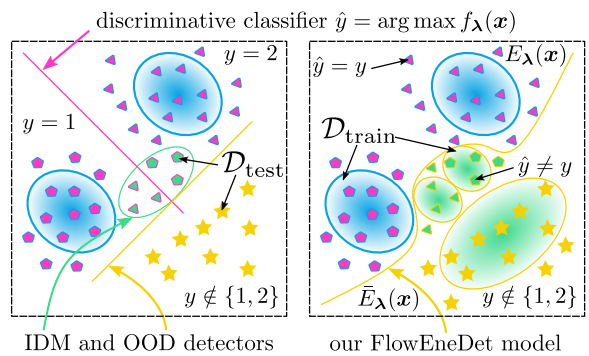

Figure 1: A discriminative model $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ is trained to predict segmentation classes $\hat{y}$ for images $\boldsymbol{x}$ using an empirical dataset $\mathcal{D}_{\text{train}}$ (blue ovals) with a closed-set labels $y \in \{1, 2\}$. However, an open-world data $\mathcal{D}_{\text{test}}$ (stars, triangles etc.) can contain out-of-distribution (OOD) classes ($y \notin \{1, 2\}$) and in-distribution misclassified (IDM) predictions ($\hat{y} \neq y$). Conventional approaches (left) aim either IDM or OOD detection. Our FlowEneDet (right) is a generative normalizing flow model that estimates likelihoods of correctly classified in-distribution data (purple positives) as well as IDM (green negatives) and OOD (yellow negatives) samples. We achieve this by modeling distributions of a scalar free energy score $E_{\boldsymbol{\lambda}}(\boldsymbol{x})$ for positives and an opposite $\bar{E}_{\boldsymbol{\lambda}}(\boldsymbol{x})$ for negatives using $\mathcal{D}_{\text{train}}$ (green ovals).

1 INTRODUCTION

Test-time robustness is one of the most wanted yet missing properties in current machine learning (ML) models when they are applied to decision-critical computer vision applications [Hendrycks et al., 2021]. Typically, ML-based models achieve high average accuracy metrics only for test-time data that are similar to a labeled training dataset distribution with a predefined set of categories. However, a test-train distributional shift and a novel open-set categories can significantly decrease accuracy [Croce et al., 2021].

We sketch this scenario with a toy example in Figure 1. Here, a discriminative task model $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ misclassifies test examples (green triangles and pentagons), and assigns wrong closed-set class predictions to novel categories (yellow stars) due to lack of coverage in a training dataset (blue ovals). In-distribution misclassification (IDM) and out-of-distribution (OOD) detection are test-time approaches for the above problem. Conventional IDM and OOD detectors estimate confidence scores for the classifier predictions as shown in Figure 1 (left). OOD detector separates a distribution of unknown categories ($y \notin \{1, 2\}$) from a distribution of known categories ($y \in \{1, 2\}$) using a threshold [Morteza and Li, 2022]. IDM detection aims to identify correctly ($\hat{y} = y$ positives) and incorrectly ($\hat{y} \neq y$ negatives) classified in-distribution data [Ramalho and Miranda, 2020].

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Existing detectors experiment with either IDM or, more often, OOD detection. Our analysis shows that IDM and OOD detection objectives have common root causes and, hence, can be addressed concurrently. We approach both objectives by explicitly modeling distributions of free energy function for positives and IDM/OOD negatives as shown in Figure 1 (right). Inspired by Djurisic et al. [2023], we explicitly learn *what a trained discriminative model knows and what it doesn't know* from the empirical training dataset. To accomplish this, we propose a low-complexity generative normalizing flow model (FlowEneDet) for concurrent IDM/OOD detection, which is trained on top of a fixed (task-pretrained) discriminative semantic segmentation model. In summary, our contributions are as follows:

- We derive a low-complexity flow-based model to estimate exact likelihoods of free energy both for positives and negatives from the training dataset.
- We tailor it for semantic segmentation application as a compact and stable 2D Glow-like [Kingma and Dhariwal, 2018] architecture that employs both the logit- and latent-space spatial context information.
- FlowEneDet achieves promising results on IDM/OOD benchmarks [Michaelis et al., 2020, Blum et al., 2019, Chan et al., 2021a] for the task-pretrained setup[1].

2 RELATED WORK

IDM and OOD detection is an active area of research for many ML-centric applications. We survey and compare a line of research that estimates categorical classifier's confidence scores for semantic segmentation in Table 1.

Several popular methods estimate confidence scores at the output of a task classifier. These include a maximum of softmax probabilities (MSP) [Hendrycks and Gimpel, 2017] or unnormalized logits (MLG) [Hendrycks et al., 2022], standardized logits (SML) [Jung et al., 2021], an energy-based detection (ENE) [Liu et al., 2020], and ODIN [Liang et al., 2018]. The latter has higher complexity due to test-time gradient perturbations. In-distribution scores in such methods are often accurate in the proximity of train data distribution due to the task's Kullback-Leibler (KL) divergence objective, but less accurate for OOD data [Kull et al., 2019].

Mukhoti and Gal [2018] propose an uncertainty-based detector that relies on approximate Bayesian inference (MCD). A notion of uncertainty can be viewed as an alternative way to define low confidence. MCD is implemented using forward passes at test-time for a task model with dropout layers and a scoring function. Unfortunately, its complexity scales linearly with the number of passes without approximation methods [Postels et al., 2019], and the dropout layer's configuration is sensitive to heuristic hyperparameters.

[1]Our code is available at github.com/gudovskiy/flowenedet

Lee et al. [2018] model data distributions using Gaussian discriminant analysis in the task's latent-space, and employ Mahalanobis distance as a confidence score. The above SML improves OOD accuracy using a similar approach, but operates in low-dimensional logit-space. Their main drawback is the assumption of Gaussian prior, which can be inaccurate in multi-label classification [Kamoi and Kobayashi, 2020].

Variational autoencoders [Baur et al., 2019] and generative adversarial networks (GANs) can be used to implement reconstruction-based detectors by training a dedicated generative model at the expense of higher complexity (Image Resynthesis by Lis et al. [2019] and SynthCP by Xia et al. [2020]). Then, a test-time difference between an input image and a generated image is a proxy of the confidence score. Unlike normalizing flows [Rezende and Mohamed, 2015], such models cannot estimate the exact data likelihoods and can be unreliable due to the tendency of capturing semantically-irrelevant low-level correlations [Nalisnick et al., 2019b]. SynBoost [Di Biase et al., 2021] addresses the latter by combining GAN sampling with other non-parametric methods.

Besnier et al. [2021] propose a dedicated observer (ObsNet) that exactly mirrors the task model architecture. It is trained to predict misclassifications using binary cross-entropy loss and adversarial attacks. Therefore, ObsNet is an improved discriminative model similar to a simple OOD detection head in [Bevandić et al., 2019]. Unlike it, our FlowEneDet is a theoretically more robust generative model that processes low-complexity scalar free energy scores.

Blum et al. [2019] introduce a relatively high complexity latent-space flow-based density estimator (flow emb. density) trained using marginal likelihood objective with the pretrained task model. Unlike it, FlowEneDet has significantly lower complexity, and, importantly, a more advanced distributional model that supports joint likelihood estimation for positives and negatives. Though not implemented, this density estimator and SynBoost [Di Biase et al., 2021] without task retraining can be used for IDM detection.

Lastly, we contrast the above IDM/OOD detectors from OOD-only methods at the bottom of Table 1. The latter retrain all task model parameters (NFlowJS [Grcić et al., 2021], Meta-OOD [Chan et al., 2021b], DenseHybrid [Grcić et al., 2022], GMMSeg [Liang et al., 2022]) or its subset (PEBAL [Tian et al., 2022]). GMMSeg does not rely on an outlier exposure [Wang et al., 2023], while NFlowJS is trained with the sampled negatives. Others emulate OOD distribution by a proxy data such as COCO [Lin et al., 2014] or ADE20K [Zhou et al., 2017] with augmentations [Li et al., 2021]. Though such methods currently achieve state-of-the-art results in OOD-only detection, they bear several major limitations such as: lack of IDM detection, inability to extend already deployed task models, and a certain degradation in tasks' in-domain segmentation accuracy. We compare FlowEneDet to these baselines on OOD-only benchmarks.

Table 1: A landscape of IDM/OOD detectors for semantic segmentation. Symbols indicate: ✓ for "yes", ✗ for "no", and † for a possible extension. We categorize methods by: discriminative or generative type, intact task mIoU accuracy (no retraining setup), IDM detection, extra network for detection, inference speed (time for detection is lower than the segmentation), source of negatives such as in-domain data (void class, misclassified pixels) or proxy dataset to emulate OOD distribution.

Method	Type	Intact mIoU, no retraining	IDM detection	Extra det. network	Fast inference	In-domain negative data	Extra OOD negative data
MSP, MLG, ENE, SML	disc.	✓	✓	✗	✓	✗	✗
ODIN, MCD	disc.	✓	✓	✗	✗	✗	✗
Mahalanobis distance	disc.	✓	✓	✗	✓	✗	✗
SynthCP, Image Resynthesis	gen.	✓	✓	✓	✗	✗	✗
SynBoost	gen.	✓	✗†	✓	✗	✓	✗
ObsNet	disc.	✓	✓	✓	✗	✓	✗
Flow emb. density	gen.	✓	✗†	✓	✗	✗	✗
FlowEneDet (ours)	gen.	✓	✓	✓	✓	✓	✗
NFlowJS	disc.	✗	✗	✗	✓	✓ (sampled)	✗
Meta-OOD	disc.	✗	✗	✗	✓	✓	✓ (COCO)
PEBAL	disc.	✗	✗	✗	✓	✗	✓ (COCO)
DenseHybrid	disc.	✗	✗	✓	✓	✗	✓ (ADE20K)
GMMSeg	gen.	✗	✗	✗	✓	✗	✗

3 THEORETICAL BACKGROUND

3.1 LIMITATIONS OF CONVENTIONAL CLOSED-SET DISCRIMINATIVE MODELS

Let $(\boldsymbol{x}, y)$ be an input-label pair where a vector $\boldsymbol{x}$ is an input image and a closed-set scalar label $y \in \{1, \dots, C\}$ has C classes. Then, a conventional discriminative model $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ from Figure 1 is optimized using a supervised training dataset $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i \in \mathbb{N}}$ of size N with an empirical risk minimization objective expressed by

$$\mathcal{L}(\boldsymbol{\lambda}) = \frac{1}{N} \sum_{i \in \mathbb{N}} L(y_i, \text{softmax} f_{\boldsymbol{\lambda}}(\boldsymbol{x}_i)), \tag{1}$$

where $L(\cdot)$ is a loss function, $\boldsymbol{\lambda}$ is the vector of parameters. The classifier's test-time prediction $\hat{y} = \arg\max \hat{\boldsymbol{y}}$, where the vector of unnormalized logits $\hat{\boldsymbol{y}} = f_{\boldsymbol{\lambda}}(\boldsymbol{x}) \in \mathbb{R}^C$.

Typically, discriminative models minimize KL divergence $D_{KL}\left[p(\boldsymbol{x}, y) \| p_{\boldsymbol{\lambda}}(\boldsymbol{x}, y)\right]$ between, correspondingly, the joint data and model probability density functions in the (1) loss. However, the underlying $p(\boldsymbol{x}, y)$ is a-priori unknown for test data and it is *approximated by the empirical training set* $\mathcal{D}_{\text{train}}$ with $\hat{p}(\boldsymbol{x}, y) = \hat{p}(y|\boldsymbol{x})\hat{p}(\boldsymbol{x})$ density function. As shown in [Gudovskiy et al., 2020], the KL divergence for (1) with one-hot labels y i.e. the cross-entropy loss can be equivalently derived with these notations as

$$\mathbb{E}_{\boldsymbol{x} \sim \hat{p}(\boldsymbol{x})} D_{KL}\left[\hat{p}(y|\boldsymbol{x}) \| p_{\boldsymbol{\lambda}}(y|\boldsymbol{x})\right] = -\frac{1}{N} \sum_{i \in \mathbb{N}} \log p_{\boldsymbol{\lambda}}(y_i|\boldsymbol{x}_i). \tag{2}$$

Hence, the discriminative approach is limited to modeling conditional density $p_{\boldsymbol{\lambda}}(y|\boldsymbol{x})$, where inputs are sampled as $\boldsymbol{x} \sim \hat{p}(\boldsymbol{x})$ and labels y are from the closed set.

3.2 MOTIVATION AND PROBLEM STATEMENT FOR CONCURRENT IDM/OOD DETECTION

Conventional OOD detection data setup assumes an in-distribution $p(\boldsymbol{x})$ and an out-of-distribution $p_{\text{OOD}}(\boldsymbol{x})$ at test-time, where the latter can have an arbitrary number of classes and is not accessible during training. Then, an OOD detector typically implements a $(C+1)$ classifier using the task's $p_{\boldsymbol{\lambda}}(y|\boldsymbol{x})$ with or without outlier exposure to separate $p(\boldsymbol{x})$ and $p_{\text{OOD}}(\boldsymbol{x})$ using an additional OOD class.

However, this conventional formulation does not account for assumptions in (2). If the empirical $\mathcal{D}_{\text{train}}$ with $\hat{p}(\boldsymbol{x})$ density does not approximate true test-time $p(\boldsymbol{x})$, the learned predictions $p_{\boldsymbol{\lambda}}(y|\boldsymbol{x})$ cannot be reliable due to a distributional shift. Then, test-time misclassifications are caused by a mismatch between $\hat{p}(\boldsymbol{x})$ and a-priori inaccessible $p(\boldsymbol{x})$. Similarly, $p_{\text{OOD}}(\boldsymbol{x})$ is a result of unavailable at train-time open-world data distribution. This is sketched in Figure 1 bottom right corner: the $p(\boldsymbol{x})$ tail is misclassified and, concurrently, there are novel OOD classes from $p_{\text{OOD}}(\boldsymbol{x})$. Lastly, the statistical objective (2) typically cannot be fully achieved even for available $\mathcal{D}_{\text{train}}$ due to model underfitting ($\mathcal{L}_{\text{train}}(\boldsymbol{\lambda}) > 0$).

This analysis motivates us to *narrow down a definition of in-distribution data* in the realistic data setup to a distribution of correctly classified examples only. Then, the detector's objective is to assign high confidence scores only for a distribution of positives in the proximity of $\hat{p}(\boldsymbol{x})$. In opposite, the detector has to assign low confidence scores both for the OOD density $p_{\text{OOD}}(\boldsymbol{x})$ and the misclassified data distribution ($\hat{y} \neq y$). While considering a single type of negatives is widely used in prior literature, our problem statement advocates to incorporate both types of negatives during training and revisit the conventional evaluation setup.

3.3 NORMALIZING FLOW FRAMEWORK

Unlike other generative models, normalizing flows introduced by Rezende and Mohamed [2015] can estimate the *exact data likelihoods*, which makes them an ideal candidate for IDM/OOD detection. These models use a change-of-variable formula to transform an arbitrary probability density function $p(\boldsymbol{z})$ into a base distribution with $p(\boldsymbol{u})$ density using a bijective invertible mapping $g : \mathbb{R}^D \rightarrow \mathbb{R}^D$. Usually, the mapping g is a sequence of basic composable transformations. Then, the $\log$-likelihood of a D-dimensional input vector $\boldsymbol{z} \sim p(\boldsymbol{z})$ can be estimated as

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \log p(\boldsymbol{u}) + \sum_{l=1}^{L} \log |\det \boldsymbol{J}_l| \,, \tag{3}$$

where a base random variable vector $\boldsymbol{u} \in \mathbb{R}^D$ is from the standard Gaussian distribution $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and the Jacobian matrices $\boldsymbol{J}_l^{D\times D} = \nabla_{\boldsymbol{z}^{l-1}} g_{\boldsymbol{\theta}_l}(\boldsymbol{z}^l)$ can be sequentially calculated for the lth block of a model $g(\boldsymbol{\theta})$ with L blocks.

3.4 THE PROPOSED FLOWENEDET MODEL

The conventional flow framework in Section 3.3 can estimate only the marginal likelihood as $\prod_{d=1}^{D} p_{\boldsymbol{\theta}}(\boldsymbol{z}_d)$. In result, the previous flow-based density estimator in [Blum et al., 2019] is limited to likelihood estimates from positives only and bears a significant computational complexity by processing high-dimensional latent-space embedding vectors. First, we address the latter limitation by processing a low-dimensional free energy vectors $\boldsymbol{z} \in \mathbb{R}^{D=2}$ that are related to the $\hat{p}(\boldsymbol{x})$ of interest in Section 3.4.1. Second, we use an autoregressive interpretation of flows from Section 3.4.2 and resolve the former limitation by introducing a distributional model for data positives and negatives in Section 3.4.3.

3.4.1 Energy-Based Approach for Flows

Grathwohl et al. [2020] and Liu et al. [2020] show that the scalar *free energy score* $E_{\boldsymbol{\lambda}}(\boldsymbol{x})$ can be derived from a pretrained classifier $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ and it is theoretically aligned with the density of input $\hat{p}(\boldsymbol{x})$ as

$$\hat{p}(\boldsymbol{x}) \approx p_{\boldsymbol{\lambda}}(\boldsymbol{x}) = e^{-E_{\boldsymbol{\lambda}}(\boldsymbol{x})}/Z(\boldsymbol{\lambda}), \tag{4}$$

where the free energy $E_{\boldsymbol{\lambda}}(\boldsymbol{x}) = -\log \sum_{y=1}^{C} e^{f_{\boldsymbol{\lambda}}(\boldsymbol{x})[y]}$ and $Z(\boldsymbol{\lambda})$ is the normalizing constant (partition function).

The energy-based framework [Lecun et al., 2006] in (4) is a key to relate the in-domain $\hat{p}(\boldsymbol{x})$ from Section 3.2 with the trained classifier's density $p_{\boldsymbol{\lambda}}(\boldsymbol{x})$. We use this result in our flow-based detector by assigning its (3) input vectors $\boldsymbol{z}$ to the scalar energy of positives $E_{\boldsymbol{\lambda}}(\boldsymbol{x})$ and the scalar energy of IDM/OOD negatives $\bar{E}_{\boldsymbol{\lambda}}(\boldsymbol{x})$ as

$$\boldsymbol{z} = [-E_{\boldsymbol{\lambda}}(\boldsymbol{x}); \bar{E}_{\boldsymbol{\lambda}}(\boldsymbol{x})] = [-E_{\boldsymbol{\lambda}}(\boldsymbol{x}); \log(1 - e^{-E_{\boldsymbol{\lambda}}(\boldsymbol{x})})]. \tag{5}$$

3.4.2 Autoregressive Interpretation of Flows

The real-valued non-volume preserving (RNVP) architecture [Dinh et al., 2017] is a sequence coupling blocks. Each lth block represents an invertible transformation $g : \mathbb{R}^D \rightarrow \mathbb{R}^{D-d}$ for the first $d < D$ elements of vector $\boldsymbol{z}$ as

$$\boldsymbol{z}_{1:d}^l = \boldsymbol{z}_{1:d}^{l-1} \,,\; \boldsymbol{z}_{d:D}^l = \boldsymbol{z}_{d:D}^{l-1} \odot e^{s\left(\boldsymbol{z}_{1:d}^{l-1}\right)} + t\left(\boldsymbol{z}_{1:d}^{l-1}\right), \tag{6}$$

where $s(\cdot)$ and $t(\cdot)$ are scale and translation operations that are implemented as two feedforward neural networks with $\boldsymbol{\theta}$ parameters, and $\odot$ is the Hadamard (element-wise) product.

The Jacobian of such transformation is a triangular matrix with a tractable $\log$-determinant in (3). Importantly, Papamakarios et al. [2017] show that the RNVP coupling implements *a special case of autoregressive transformation*. The autoregressive characterization of the coupling block using a single Gaussian is given by m^{th} conditional likelihoods

$$p_{\boldsymbol{\theta}}(z_m^l | \boldsymbol{z}_{1:m-1}^l) = \mathcal{N}(z_m^l | t_m, e^{2s_m}), \tag{7}$$

where $t_m = s_m = 0$ for $\forall m \leq d$ and depend on $\boldsymbol{z}_{1:d}^{l-1}$ only.

Hence, our FlowEneDet model with the $\boldsymbol{z} \in \mathbb{R}^{D=2}$ input (5) can estimate conditional likelihoods of positive's and negative's free energy scores from the output of $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ using the autoregressive interpretation of RNVP at every coupling (6). This results in a very *low-complexity architecture*, because the coupling compute is $\mathcal{O}(D^2)$. In contrast, the complexity of Blum et al. [2019] with latent-space vectors $\boldsymbol{e} \in \mathbb{R}^V$ is significantly higher since $V \gg 2$ as in Figure 2.

3.4.3 Distributional Model with Full Covariance

The conventional choice for a base distribution in (3) is not suitable for modeling joint probability density (7) of positive and negative energy scores (5). Therefore, we replace the base univariate Gaussian in (3) by a

$$p(\boldsymbol{u}) = \boldsymbol{\beta} \odot \mathcal{N}(\boldsymbol{u} | \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{8}$$

where $\boldsymbol{\beta} \in \mathbb{R}^D$ is a vector of probabilities to model data imbalances between positives and negatives. A mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ parameterize multivariate Gaussian distribution.

Then, the conditional $\log$-likelihoods $\log p_{\boldsymbol{\theta}}(\boldsymbol{z}|m)$ define whether an input is from positive or negative category. They can be derived by substituting (8) to (3) and conditioning each term in (7) by the category m using the chain rule for autoregressive output [Papamakarios et al., 2017] as

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}|m) = \sum\nolimits_{d=1}^{D} \log p_{\boldsymbol{\theta}}(z_d | \boldsymbol{z}_{1:d-1}, m) = \\ \log \boldsymbol{\beta} + \log \mathcal{N}(\boldsymbol{u} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \sum\nolimits_{l=1}^{L} \log |\det \boldsymbol{J}_l| \,, \end{aligned} \tag{9}$$

where the compute-intensive distributional and Jacobian terms are calculated only once for the whole model.

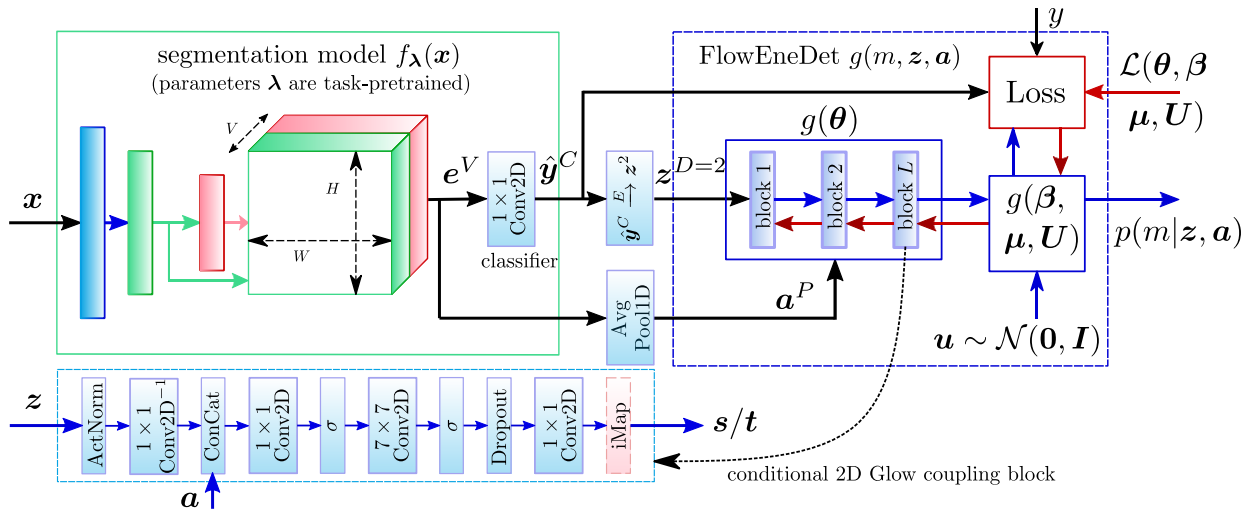

Figure 2: A pretrained segmentation model $f_{\boldsymbol{\lambda}}(\boldsymbol{x})$ (top left) is a multi-scale network with a linear classifier and fixed parameters $\boldsymbol{\lambda}$. Its outputs are latent-space vectors $\boldsymbol{e}$ and unnormalized logits $\hat{\boldsymbol{y}}$. Our FlowEneDet (right) derives an energy-based input vector $\boldsymbol{z}$ from $\hat{\boldsymbol{y}}$ and a condition vector $\boldsymbol{a}$ from $\boldsymbol{e}$, and processes them by a 2D Glow-like [Kingma and Dhariwal, 2018] architecture with L blocks and a distributional part. Then, FlowEneDet estimates conditional likelihoods $p(m|\boldsymbol{z}, \boldsymbol{a})$, where the mth category defines a likelihood of image $\boldsymbol{x}$ being either a positive or negative (IDM/OOD).

We model full covariance matrix $\boldsymbol{\Sigma}$ of the multivariate Gaussian distribution by an upper triangular matrix $\boldsymbol{U}$ using the Cholesky decomposition similarly to [Kruse, 2020]. Then, the distributional term in (9) is given by

$$\log \mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum\nolimits_{d=1}^{D} \operatorname{diag}(\boldsymbol{U})_d - \frac{1}{2}\|\boldsymbol{U}(\boldsymbol{z} - \boldsymbol{\mu})\|_2^2. \tag{10}$$

Using the Bayes rule for (9), confidence scores of interest can be estimated as conditional likelihoods

$$p_{\boldsymbol{\theta}}(m|\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|m)p(m) / \sum\nolimits_{d=1}^{D} p_{\boldsymbol{\theta}}(\boldsymbol{z}|m = d). \tag{11}$$

Unlike the discriminative model (2) that learns only conditionals $p_{\boldsymbol{\lambda}}(y|\boldsymbol{x})$, our generative FlowEneDet models the joint density $p_{\boldsymbol{\theta}}(m, \boldsymbol{z})$. Hence, it exactly estimates $p_{\boldsymbol{\theta}}(m|\boldsymbol{z})$ and approximates $\hat{p}(\boldsymbol{x})$ [Nalisnick et al., 2019a]. The joint modeling can be practically used to generate hard cases by the virtual outlier synthesis [Du et al., 2022].

4 FLOWENEDET FOR SEMANTIC SEGMENTATION

In this section, we present FlowEneDet architecture adopted for semantic segmentation. It contains two high-level parts: a sequence of L coupling blocks $g(\boldsymbol{\theta})$ and a distributional part $g(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{U})$ as shown Figure 2 (right). Next, we explain key modifications to the theoretical model from Section 3.4.

4.1 THE PROPOSED ARCHITECTURE

First, we extend the conventional RNVP coupling by 2-dimensional processing as shown in Figure 2 (bottom left). This captures information encoded along spatial dimensions for image segmentation. It is achieved by a sequence of Conv2D layers with kernels of size $1 \times 1 \to \sigma \to 7 \times 7 \to \sigma \to 1 \times 1$, where σ is the sigmoid activation function.

Second, we extend the RNVP coupling by the activation normalization (ActNorm) and invertible 1×1 convolution (Conv2D^{-1}), which, effectively, results in a 2D Glow [Kingma and Dhariwal, 2018] coupling block. Empirical experiments show that such layers significantly speed up convergence time and training stability. Dropout with 20% probability is applied before the last 1×1 Conv2D layer to decrease overfitting. Optionally, we add an invertible map-based attention layer (iMap) from [Sukthanker et al., 2022]. In particular, we apply it only to the SegFormer [Xie et al., 2021] backbone. We empirically find that this improves training stability and decreases variance in results.

Third, we recognize that the logit-space energy score alone can limit the expressiveness of our density estimator. Therefore, we augment (condition) each coupling block by the low-dimensional embedding vector $\boldsymbol{a}^P$. A mapping from the embedding $\boldsymbol{e}^V$ ($\boldsymbol{e}^V \to \boldsymbol{a}^P$) is accomplished using 1D average pooling. Then, we follow [Ardizzone et al., 2019] and concatenate $\boldsymbol{z}$ intermediate results with the pooled projection $\boldsymbol{a}$ in Figure 2 (bottom left). We compare FlowEneDet (FED) that is configured with conditional vector $\boldsymbol{a}$ (FED-C) as well as unconditional FED-U model in our experiments.

Fourth, we improve experimental results by reparameteriz-

ing the scale operation $s(\boldsymbol{z}_{1:d})$ in (6) and the corresponding Jacobian. Particularly, we define the scale as $1 - \text{sigmoid}(\boldsymbol{z}_{1:d})$ and $\log|\det \boldsymbol{J}| = -\text{softplus}(\boldsymbol{z}_{1:d})$, which limits their range to $(0 : 1)$ and $(-\infty : 0)$, respectively. At the same time, we follow the conventional channel-wise input masking [Dinh et al., 2017] and exchange the first and second halves of the input $\boldsymbol{z}$ after every coupling block.

4.2 OPTIMIZATION OBJECTIVE

We are interested in modeling and estimating likelihoods of energy scores for positive and negative (IDM/OOD) examples as defined in (5). Energy score calculation is implemented using numerically-stable *logsumexp* operation. Therefore, the proposed FlowEneDet explicitly estimates conditional likelihoods $p_{\boldsymbol{\theta}}(m|\boldsymbol{z}, \boldsymbol{a})$, where $m \in \{1, 2\}$, $\boldsymbol{z} \in \mathbb{R}^2$ and $\boldsymbol{a} \in \mathbb{R}^P$. At test-time, we always output likelihood of the second negative category ($m = 2$) as an uncertainty estimate.

In total, FlowEneDet contains $\boldsymbol{\theta}_{\text{FED}} = [\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{U}]$ parameters, where $\boldsymbol{\theta}$ are the coupling parameters and the rest of parameters describe the distributional model. All parameters are jointly optimized using an objective that maximizes (11) log-likelihoods with conditional and marginal (denominator) terms. This can be simplified by a numerically-stable log-softmax operation as

$$\mathcal{L}(\boldsymbol{\theta}_{\text{FED}}) = -\sum_{i \in \mathbb{N}} \log \text{softmax} \ \log p_{\boldsymbol{\theta}_{\text{FED}}}(\boldsymbol{z}_i, \boldsymbol{a}_i | m_i)/N, \tag{12}$$

where this objective is equivalent to the cross-entropy loss.

FlowEneDet labels m are binary (positive or negative examples) in the (12) loss function. We derive binary labels from the task ground-truth y (including the void class) and task classifier predictions $\hat{y}$ such that $m_i = (y_i \neq \hat{y}_i)$. In order to increase training stability, we optimize distributional parameters $\boldsymbol{\beta}$ and $\text{diag}(\boldsymbol{U})$ using the same sigmoid/softplus reparameterization as for $s(\cdot)$ operation in Section 4.1.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Task models. We experiment with SegFormer-B2 (SF-B2) [Xie et al., 2021] and DeepLabV3+ [Chen et al., 2018] with ResNet-101 backbone (DL-R101) semantic segmentation models. We use their public checkpoints pretrained on Cityscapes [Cordts et al., 2016], and our code extends open-source MMSegmentation [Contributors, 2020] library.

Benchmarks. Cityscapes (CS) contains 19 labeled classes and the unlabeled void class (background). The pretrained DL-R101 and SF-B2 models achieve, correspondingly, 81.0% and 81.1% mean intersection over union (mIoU) metric for CS validation split. In our IDM/OOD experiments, we use 19 in-domain (ID) classes for IDM detection and the void class for OOD detection. We evaluate detection robustness by adding test-time image corruptions to CS. We follow recent robustness benchmarks [Croce et al., 2021, Michaelis et al., 2020] and experiment with a synthetically-corrupted CS-C validation dataset. In particular, we apply motion blur, brightness and snow types of image corruptions with severity range from 1 (low) to 4 (high) and average corresponding results. Lastly, we use Fishyscapes [Blum et al., 2019] (FS) and SegmentMeIfYouCan [Chan et al., 2021a] (SMIYC) benchmarks designed for OOD-only evaluations with the binary ID/OOD labels.

Table 2: FED SF-B2 ablation study on FS L&F **validation split**, %. The **best** result is highlighted. Design space is defined as follows: covariance matrix $\boldsymbol{U}$ is full or diagonal, kernel size K for the flow's Conv2D layer is 3×3 or 7×7. Our default configuration: full-covariance $\boldsymbol{U}$, $K = 7 \times 7$, $L = 8$, and $P = 32$ for FED-C or $P = 0$ for FED-U.

Method	$\boldsymbol{U}$	K	P	FS L&F	
				AP↑	FPR_{95} ↓
FED-U	full	7×7	-	39.90	18.66
FED-C	full	7×7	32	41.15	11.1
FED-U (TTA)	full	7×7	-	41.75	10.05
FED-C (TTA)	full	7×7	32	**56.11**	**3.87**
FED-U (TTA)	full	3×3	-	42.28	9.94
FED-C (TTA)	full	3×3	32	51.98	6.88
FED-U (TTA)	diag	7×7	-	41.71	9.99
FED-C (TTA)	diag	7×7	32	51.62	4.04

Baselines. We reimplement MSP, MLG, SML, ENE, and MCD baselines from Table 1. In our MCD implementation, we apply dropout only to the last linear layer to avoid high complexity. We exclude ODIN because it requires test-time gradients and underperforms compared to ENE method. We report results for other relevant baselines from Table 1 using their best publicly available benchmark results. Due to differences in architectures and implementations, we present several FlowEneDet configurations that are comparable in terms of complexity to the above methods.

FlowEneDet. We experiment with the following configurations: unconditional FED-U and conditional FED-C with $P = 32$ latent vectors. We train each configuration four times and report evaluation's mean (μ) and standard deviation ($\pm\sigma$) for every benchmark with the exception of private test splits. Reimplemented baselines have been evaluated once. Each detector with DL-R101 and SF-B2 backbone has $L = 4$ and $L = 8$ coupling blocks, respectively. In addition, we can apply a test-time augmentation (TTA) with $1/4\times$, $1/2\times$ and $1\times$ image resizing to SF-B2 backbone for confidence score averaging. Our TTA configuration is chosen to match inference speed of a popular WideResNet-38 (WRN-38) backbone in FS and SMIYC leaderboards.

Table 3: OOD results for Fishyscapes **validation split**, %. The **best** and the second best results are highlighted.

Method	Task backbone	CS mIoU↑	L&F AuROC↑	L&F AP↑	L&F FPR_{95} ↓	Static AuROC↑	Static AP↑	Static FPR_{95} ↓
MCD	DL-R101	80.3	88.94	10.85	37.79	93.14	25.59	27.24
MSP	DL-R101	80.3	86.99	6.02	45.63	88.94	14.24	34.10
MLG	DL-R101	80.3	92.00	18.77	38.13	92.80	27.99	28.50
ENE	DL-R101	80.3	93.50	25.79	32.26	91.28	31.66	37.32
SML	DL-R101	80.3	96.88	36.55	14.53	96.69	48.67	**16.75**
SynthCP	DL-R101	80.3	88.34	6.54	45.95	89.90	23.22	34.02
Synboost	DL-R101	80.3	94.89	40.99	34.47	92.03	48.44	47.71
FED-U	DL-R101	81.0	97.65$\pm$0.2	37.05$\pm$0.6	11.35$\pm$0.5	95.96$\pm$0.2	46.32$\pm$0.4	20.15$\pm$1.3
FED-C	DL-R101	81.0	96.34$\pm$0.2	28.71$\pm$2.5	18.48$\pm$1.5	92.89$\pm$0.4	25.34$\pm$4.7	32.69$\pm$1.4
FED-U	SF-B2	81.1	96.72$\pm$0.2	39.90$\pm$0.7	18.66$\pm$1.5	**96.84**$\pm$0.1	**55.93**$\pm$0.7	17.15$\pm$0.9
FED-C	SF-B2	81.1	**98.28**$\pm$0.1	**42.15**$\pm$0.4	**11.10**$\pm$0.1	93.31$\pm$0.8	47.56$\pm$2.5	37.53$\pm$3.1
SML	DL-WRN38	81.4	94.97	22.74	33.49	97.25	**66.72**	12.14
SynBoost	DL-WRN38	81.4	96.21	**60.58**	31.02	95.87	66.44	25.59
FED-U (TTA)	SF-B2	81.1	97.83$\pm$0.1	41.75$\pm$0.3	10.05$\pm$0.2	**98.30**$\pm$0.1	66.60$\pm$0.2	**8.94**$\pm$0.1
FED-C (TTA)	SF-B2	81.1	**99.11**$\pm$0.1	56.11$\pm$4.4	**3.87**$\pm$0.2	96.88$\pm$0.2	52.61$\pm$1.4	14.91$\pm$1.2

Metrics. We use standardized metrics for FS and SMIYC benchmarks: area under the receiver operating characteristic curve (AuROC), average precision (AP) [Hendrycks and Gimpel, 2017], and false positive rate when the true positive rate is 95% (FPR_{95}) [Liang et al., 2018]. The latter metric is considered the most important in practice. We use an open-mIoU metric for concurrent IDM/OOD detection evaluations. First, we compute a detection threshold using F_1-score [Lipton et al., 2014]. Then, this threshold is used to predict a binary (positive or IDM/OOD negative) decision. Next, the predicted negatives are added as an extra void class to IoU computation as proposed by Grcić et al. [2022]. Finally, we calculate the open-mIoU metric for $(C+1)$ IoUs with averaging by C classes to conform with the open-world setup. Unlike it, the conventional mIoU rejects all OOD (unlabeled void) pixels using the ground truth mask, which leads to an unrealistic closet-set recognition setup.

5.2 QUANTITATIVE RESULTS

Ablation study. Table 2 shows an ablation study for FED variants with SF-B2 backbone on FS L&F validation dataset. We find that TTA significantly increases performance metrics both for FED-U and FED-C. Next, we verify that the full covariance matrix $\boldsymbol{U} \in \mathbb{R}^{2\times 2}$ from Section 3.4.3 outperforms the univariate $\text{diag}(\boldsymbol{U}) \in \mathbb{R}^2$ approach. Finally, a 7×7 kernel size with larger receptive field is superior to a 3×3 Conv2D layer for a more advanced FED-C configuration. We use the selected configurations as default in further experiments. Appendix contains an extended ablation study.

OOD-only detection. Tables 3-5 present comprehensive OOD evaluations when applied to FS public validation split as well as FS and SMIYC private test splits, respectively. Our conditional FED-C configuration exceeds or is on par with the state-of-the-art in majority of metrics for the setup without task retraining, and even outperforms the best methods with retraining (NFlowJS, DenseHybrid, PEBAL) in Table 4 on FS L&F test split (50.15% AP, 5.2% FPR_{95}).

The only outlier is FS Static dataset in Tables 3-4, where unconditional FED-U is consistently superior than the more advanced FED-C variant. Particularly, FED-U has the second best test split results using AP metric (67.80% AP for FED-U vs. 72.59% AP for SynBoost), but it underperforms in FPR_{95} (21.58% FPR_{95} for FED-U vs. 17.43% FPR_{95} for flow embedded density method [Blum et al., 2019]). A possible reason why FED-C achieves lower performance metrics on FS Static than the FED-U is the distribution of latent-space features in OOD objects that cannot be properly captured by our naïve average pooling. Hence, a more robust feature pooling can be a topic for future research.

OOD detection results cannot be considered separately from the task's semantic segmentation accuracy metric itself. For example, DenseHybrid and PEBAL sacrifice, correspondingly, 0.4% and 0.7% Cityscapes closed-set mIoU accuracy due to a setup with retraining. A benchmark-agnostic task model with the corresponding detector that can be universally applied in all experiments using the same parameters is another important factor when analyzing Table 4-5 results. For instance, DenseHybrid uses convolutional DL-WRN38 backbone for FS, but LDN-121 backbone for SMIYC. On the other hand, PEBAL trains several detector models with different hyperparameters and applies each checkpoint depending on the selected benchmark. In our empirical studies, we find that the transformer-based SegFormer-B2 is more universally-applicable segmentation backbone. As a result, we apply the same backbone and detector parameters using a single checkpoint file to all our OOD evaluations without additional hyperparameter tuning.

Table 4: OOD results for Fishyscapes **test split**, %. The **best** and the second best results are highlighted.

Method	Intact mIoU, no retraining	Task backbone	CS	L&F		Static	
			mIoU↑	AP↑	FPR_{95} ↓	AP↑	FPR_{95} ↓
MSP	✓	DL-R101	80.3	1.77	44.85	12.88	39.83
Emb. density	✓	DL-R101	80.3	4.25	47.15	62.14	**17.43**
SML	✓	DL-R101	80.3	31.05	21.52	53.11	19.64
Image Resynthesis	✓	PSP-R101	79.9	5.70	48.05	29.60	27.13
SynBoost	✓	DL-WRN38	81.4	43.22	15.79	**72.59**	18.75
FED-U TTA	✓	SF-B2	81.1	20.45	11.38	67.80	21.58
FED-C TTA	✓	SF-B2	81.1	**50.15**	**5.20**	61.06	31.97
NFlowJS	✗	LDN-121	77.4	43.66	8.61	54.68	10.00
DenseHybrid	✗	DL-WRN38	81.0	43.90	6.18	72.27	5.51
PEBAL	✗	DL-WRN38	80.7	44.17	7.58	92.38	1.73
GMMSeg	✗	DL-R101	81.1	55.63	6.61	76.02	15.96

Table 5: OOD results for SMIYC **test split**, %. The **best** and the second best results are highlighted.

Method	Intact mIoU, no retraining	Task backbone	CS	Obstacle		L&F	
			mIoU↑	AP↑	FPR_{95} ↓	AP↑	FPR_{95} ↓
MSP	✓	DL-WRN38	81.4	15.7	16.6	30.1	33.2
MCD	✓	DL-R101	80.3	4.9	50.3	36.8	35.6
Emb. density	✓	DL-R101	80.3	0.8	46.4	61.7	10.4
Void Classifier	✓	DL-R101	80.3	10.4	41.5	4.8	47.0
Image Resynthesis	✓	PSP-R101	79.9	37.7	4.7	57.1	8.8
Mah. distance	✓	DL-WRN38	81.4	20.9	13.1	55.0	12.9
SynBoost	✓	DL-WRN38	81.4	71.3	3.2	**81.7**	4.6
FED-C (TTA)	✓	SF-B2	81.1	**73.7**	**1.0**	79.8	**2.9**
NFlowJS	✗	LDN-121	77.4	85.6	0.4	89.3	0.7
DenseHybrid	✗	LDN-121	N/A	81.7	0.2	78.7	2.1

Concurrent IDM/OOD detection. Table 6 demonstrates the utility of concurrent IDM/OOD detection on CS, CS-C and snow-only corruption using the open-mIoU metric. Our FED-U marginally outperforms SML on the uncorrupted CS because OOD pixels represent the majority of negatives in this case, while FED-C with TTA averaging and $P = 128$ achieves 5.2% higher result. However, an amount of IDM negatives increases in case of CS-C and, especially, snow-type corruption. Then, OOD-focused SML is inferior even when compared to ENE [Liu et al., 2020]. Our FED-U surpasses others by a larger margin (2.5-6% open-mIoU) on CS-C and snow-only case relatively to the uncorrupted CS (up to 0.2% improvement), while a more complex FED-C with TTA shows an additional 3-4% gain.

Interestingly, transformer-based SF-B2 is significantly more robust (10-30% higher open-mIoU) to corruptions than the convolutional DL-R101. Lastly, Table 6 shows that image distortions present a significant threat to task's accuracy and not all IDM/OOD detectors are accurate enough to surpass a simple no-detector baseline using the open-mIoU metric. Therefore, it is important to use robust task's backbone e.g. transformer-based SegFormer [Zhou et al., 2022] and avoid operating in an extreme environment when detector predicts broadly low-confident segmentation predictions.

5.3 QUALITATIVE RESULTS

Figure 3 compares qualitative results when different FED configurations are applied to the FS validation data. FED-U detector with the convolutional DL-R101 backbone outputs significantly less accurate confidence scores when compared to the transformer-based SF-B2 backbone. In particular, convolutional backbone produces noisy predictions for certain in-domain areas such as road patterns or a clutter of small objects in the background. We believe, this is related to a very local receptive field for convolutional backbones. FED configurations with the SF-B2 backbone output more consistent confidence scores due to global transformer receptive field. The FED configuration with TTA improves predictions by capturing very fine details in OOD object shapes. This is related to the convolutional architecture of our flow network itself, and TTA's multi-scale detection allows to partially overcome this limitation. Also, though not visible in these examples, TTA likely suppresses spurious false predictions because it smooths the estimated scores. Appendix contains additional qualitative visualizations.

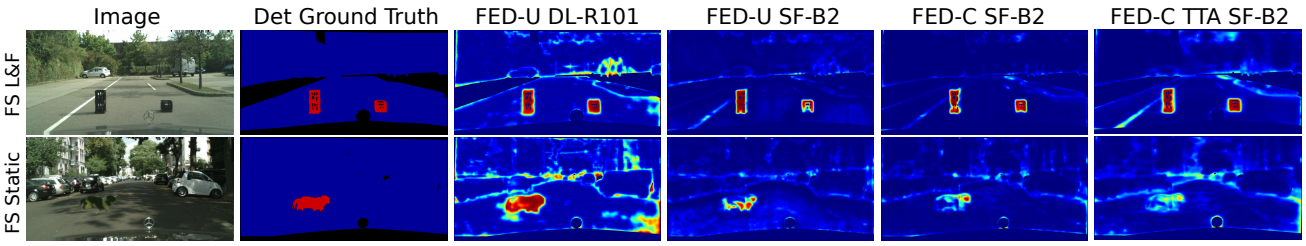

Figure 3: This figure presents: images from FS L&F and Static validation datasets, OOD ground truth, and predictions for FED variants. FED-U predictions with DL-R101 backbone are less precise than the ones with SF-B2 transformer. Test-time augmentation (TTA) allows to further refine the exact shape of OOD objects by averaging multi-scale confidence scores.

Table 6: Concurrent IDM/OOD detection for Cityscapes (CS), corrupted CS-C and snow-only CS-C, open-mIoU %.

Method	Backbone	CS↑	CS-C↑	Snow↑
None	DL-R101	81.0	53.8	15.4
MCD	DL-R101	54.6	39.3	15.3
MSP	DL-R101	61.7	44.3	16.5
ENE	DL-R101	52.8	38.7	17.1
SML	DL-R101	84.4	57.4	12.6
FED-U	DL-R101	**84.6**$\pm$0.6	**59.9**$\pm$0.8	**18.9**$\pm$1.5
None	SF-B2	81.1	62.21	35.83
MCD	SF-B2	58.1	47.77	30.33
MSP	SF-B2	63.6	52.22	32.82
ENE	SF-B2	68.9	59.91	45.63
SML	SF-B2	81.4	66.26	40.30
FED-U	SF-B2	81.4$\pm$0.8	70.1$\pm$0.7	51.5$\pm$0.8
FED-C (TTA)	SF-B2	**86.6**$\pm$0.6	**74.5**$\pm$0.8	**54.3**$\pm$0.6

Table 7: Inference speed: frames per second (fps) on A6000 GPU and total model size (MB) for 1024×2048 images.

Method	Backbone	Speed, fps↑	Size, MB↓
MSP, ENE, SML	DL-R101	4.46	230.44
MCD	DL-R101	3.79	230.44
FED-U	DL-R101	4.39	230.53
FED-C	DL-R101	4.25	236.01
SynBoost	DL-WRN38	0.9	2,286.80
MSP, ENE, SML	SF-B2	5.2	94.47
FED-U (TTA)	SF-B2	4.1 (2.2)	94.62
FED-C (TTA)	SF-B2	3.6 (0.9)	101.69

5.4 COMPLEXITY EVALUATIONS

Table 7 reports complexity estimates for the evaluated DL-R101 and SF-B2 task models with detectors using frames per second (fps) metric with a size-1 mini-batch on A6000 GPU and the size of all floating-point parameters. Also, we include SynBoost with WRN-38. The first row shows complexity metrics for the task model and computation-free detectors (MSP, ENE, SML). The reimplemented MCD with 32 forward passes has a dropout layer applied only to the classifier layer to avoid high complexity.

FED detector variants contain 4 and 8 coupling blocks for DL-R101 and SF-B2, respectively. Then, its model size is marginally larger (up to 8% for FED-C) than the task model itself. In comparison, reconstruction-based SynBoost is more than 20× larger than our FED-C with SF-B2. Inference speed without TTA is 5% to 44% lower depending on the backbone and architecture. The enabled TTA nearly linearly decreases inference speed in the current off-the-shelf implementation. This can be improved if exclude task's processing from TTA and apply it only to the FED detector.

6 CONCLUSIONS

In this paper, we analyzed a practical data setup with distributional shifts and out-of-distribution classes, which can result in critically-incorrect predictions produced by ML-based semantic segmentation models. To improve task model robustness, we proposed to incorporate a concurrent IDM/OOD detector to predict in-distribution misclassified data points and out-of-distribution classes. While IDM/OOD detection is challenging for certain types of corruptions, we significantly improved detection results using the proposed normalizing flow-based FlowEneDet model.

FlowEneDet with 2D architecture explicitly modeled likelihoods for semantic segmentation's positive (correctly classified) and negative (IDM/OOD) pixels using low-complexity energy-based inputs. We achieved promising results in IDM and/or OOD detection without task's retraining on Cityscapes, Cityscapes-C, Fishyscapes and SegmentMeIfYouCan benchmarks. This setup can extend already deployed segmentation models, keep their original mIoU accuracy intact, and improve practical open-mIoU metric. Moreover, we showed that FlowEneDet has relatively low complexity and memory overhead when applied to DeepLabV3+ and a more empirically robust SegFormer backbone.

References

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv:1907.02392*, 2019.

Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, 2019.

Victor Besnier, Andrei Bursuc, David Picard, and Alexandre Briot. Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation. In *ICCV*, 2021.

Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous semantic segmentation and outlier detection in presence of domain shift. In *Proceedings of the German Conference on Pattern Recognition*, 2019.

Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *ICCVW*, 2019.

Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. SegmentMeIfYouCan: A benchmark for anomaly segmentation. In *Proceedings of the NeurIPS Track on Datasets and Benchmarks*, 2021a.

Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *ICCV*, 2021b.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

MMSegmentation Contributors. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. RobustBench: a standardized adversarial robustness benchmark. In *Proceedings of the NeurIPS Track on Datasets and Benchmarks*, 2021.

Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes. In *CVPR*, 2021.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2017.

Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *ICLR*, 2023.

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

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020.

Matej Grcić, Petra Bevandić, Zoran Kalafatić, and Siniša Šegvić. Dense anomaly detection by robust learning on synthetic negative data. *arXiv:2112.12833*, 2021.

Matej Grcić, Petra Bevandić, and Siniša Šegvić. DenseHybrid: Hybrid anomaly detection for dense open-set recognition. In *ECCV*, 2022.

Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. Deep active learning for biased datasets via fisher kernel self-supervision. In *CVPR*, 2020.

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

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ML safety. *arXiv:2109.13916*, 2021.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *ICML*, 2022.

Sanghun Jung, Jungsoo Lee, Daehoon Gwak, Sungha Choi, and Jaegul Choo. Standardized Max Logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *ICCV*, 2021.

Ryo Kamoi and Kei Kobayashi. Why is the Mahalanobis distance effective for anomaly detection? *arXiv:2003.00402*, 2020.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

Jakob Kruse. Technical report: Training mixture density networks with full covariance matrices. *arXiv:2003.05739*, 2020.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *NeurIPS*, 2019.

Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. *A tutorial on energy-based learning*. MIT Press, 2006.

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

Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, 2021.

Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. GMMSeg: Gaussian mixture based generative semantic segmentation models. In *NeurIPS*, 2022.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, 2014.

Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *ICCV*, 2019.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv:1907.07484*, 2020.

Peyman Morteza and Yixuan Li. Provable guarantees for understanding out-of-distribution detection. In *AAAI*, 2022.

Jishnu Mukhoti and Yarin Gal. Evaluating Bayesian deep learning methods for semantic segmentation. *arXiv:1811.12709*, 2018.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. In *ICML*, 2019a.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *ICLR*, 2019b.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *NeurIPS*, 2017.

Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *ICCV*, 2019.

Tiago Ramalho and Miguel Miranda. Density estimation in representation space to predict model uncertainty. In *International Workshop on Engineering Dependable and Secure Machine Learning Systems*, 2020.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

Rhea Sanjay Sukthanker, Zhiwu Huang, Suryansh Kumar, Radu Timofte, and Luc Van Gool. Generative flows with invertible attentions. In *CVPR*, 2022.

Yu Tian, Yuyuan Liu, Guansong Pang, Fengbei Liu, Yuanhong Chen, and Gustavo Carneiro. Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes. In *ECCV*, 2022.

Qizhou Wang, Junjie Ye, Feng Liu, Quanyu Dai, Marcus Kalander, Tongliang Liu, Jianye HAO, and Bo Han. Out-of-distribution detection with implicit outlier transformation. In *ICLR*, 2023.

Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In *ECCV*, 2020.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.

Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M. Alvarez. Understanding the robustness in vision transformers. In *ICML*, 2022.