A Metacognitive Approach to Out-of-Distribution Detection for Segmentation

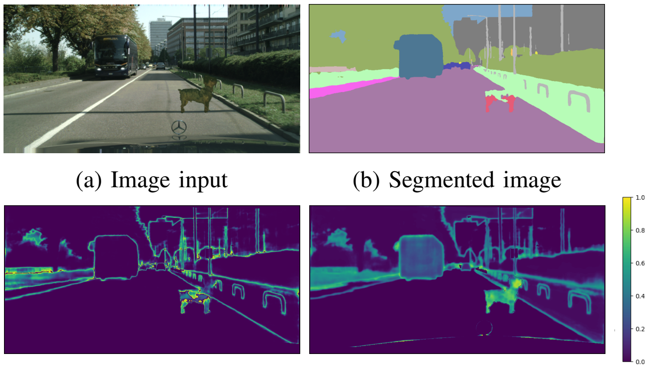
Meghna Gummadi, Cassandra Kent, Karl Schmeckpeper, and Eric Eaton
University of Pennsylvania, Philadelphia, PA, USA
{meghnag, dekent, karls, eeaton}@seas.upenn.edu

***Abstract*— Despite outstanding semantic scene segmentation in closed-worlds, deep neural networks segment novel instances poorly, which is required for autonomous agents acting in an open world. To improve out-of-distribution (OOD) detection for segmentation, we introduce a metacognitive approach in the form of a lightweight module that leverages entropy measures, segmentation predictions, and spatial context to characterize the segmentation model's uncertainty and detect pixel-wise OOD data in real-time. Additionally, our approach incorporates a novel method of generating synthetic OOD data in context with in-distribution data, which we use to fine-tune existing segmentation models with maximum entropy training. This further improves the metacognitive module's performance without requiring access to OOD data while enabling compatibility with established pre-trained models. Our resulting approach can reliably detect OOD instances in a scene, as shown by state-of-the-art performance on OOD detection for semantic segmentation benchmarks.**

I. INTRODUCTION

Current deep neural networks (DNNs) achieve near-perfect performance in semantic segmentation, but only in the closed-world paradigm, where test data is drawn from the same distribution as training data [1], [2], [3], [4]. However, most real-world agents must operate in highly dynamic open-world settings, including autonomous driving [5], [6], assistive robotics [7], [8], [9], and social robotics [10], [11], where encountering objects beyond the fixed training distribution is the norm. These deployed agents must behave reasonably on out-of-distribution (OOD) instances encountered in the open world. Training agents on large amounts of annotated data fails to generalize to unseen classes [12], [13], [14], but a practical alternative is to recognize novel or OOD instances as they are encountered and then accommodate the new data [15], [16]. We focus on the first step: enabling semantic segmentation to identify OOD instances.

An intuitive way of tackling this challenge is to look for low-confidence segmentations, based on the assumption that predictions will be less certain for OOD instances [17], [18]. Reasoning over uncertainty metrics, including prediction dispersion [19] and model consensus [20], [21], endows *metacognitive* systems [22], [23] to monitor their own performance and identify errors. A common approach is to make decisions using a simple threshold over the uncertainty quantification; while often effective, simple thresholding is non-adaptive and may overly generalize. We propose to provide segmentation models with a lightweight decision making module that considers context along with proven uncertainty measures, enabling segmentation models to perform metacognitive reasoning about their predictions.



(a) Image input

(b) Segmented image

(c) Entropy map

(d) OOD estimate

Fig. 1: Example input-output for MEMOS from Fishyscapes dataset. Input image (a) passes through a segmentation net, yielding segmentation output (b) and prediction entropy (c). Images (b) and (c) are combined channel-wise as input to the metacognitive net, which outputs (d): per-pixel predictions of OOD vs ID. Low entropy is blue, high entropy is yellow.

Such metacognitive reasoning heavily relies on uncertainty estimates, however large DNNs are known to be overconfident, which may hinder detection of OOD instances [24]. Network calibration using an appropriate validation set [24] is a widely accepted way to reduce this overconfidence, but this raises the question: what validation data is appropriate to use? Top-performing OOD detectors assume access to sample OOD validation sets for network calibration [19], hyperparameter tuning [18], and additional training [25]. While these methods show that using OOD data during training is effective, the choice of such data can bias a model toward whatever a data engineer considers most likely to be encountered. Additionally, actual OOD data is not typically available at design time, and can not fully represent OOD data encountered after deployment.

We present a framework that detects OOD instances without relying on any OOD data during training, by leveraging measures of dispersion of well-calibrated segmentation networks in a metacognitive network module. We make further use of the available in-distribution (ID) data by transforming

*This research was partially supported by DARPA SAIL-ON contract HR001120C0040 and the Army Research Office MURI W911NF20-1-0080.

portions of ID images into unrecognizable classes, creating synthetic OOD data situated in an ID context. The combined ID and synthetic OOD data is used to improve the entropy calibration of segmentation models through maximum entropy training [26] by 1) tightening the boundary around low-entropy predictions, and 2) reducing the network's confidence on identifying synthetic OOD data, which primes the network to make high-entropy predictions more frequently when encountering actual OOD data. Our metacognitive network then uses these predictions to make final context-supported decisions on which pixels belong to OOD class instances (Figure 1). We refer to our framework as Maximum-Entropy Metacognitive OOD Segmentation (MEMOS).

Our key contributions include:

- We develop a metacognitive network module that leverages the predicted class, entropy, and spatial context to generate pixel-wise OOD detection for segmentation.
- We propose a method for generating synthetic OOD data from ID data that improves the entropy calibration of segmentation models through maximum entropy training.
- The full MEMOS framework reliably identifies novel instances in real-time (30-40 Hz), and achieves up to a 75% increase in performance over comparable methods.

II. Background and Related Work

OOD Detection for Segmentation The most common approach to enable segmentation models to detect OOD inputs is to provide sample OOD data to the training process and optimize explicitly to detect these OOD samples [27], [19], [28], [29]—an approach that can bias the model, as discussed previously. Other methods reason about the uncertainty of the prediction by using handcrafted metrics derived from the segmentation model's output confidences [30], [19], [31], by combining these confidences with other approaches [32], by using Bayesian Neural Networks [20], or by using ensembles [33]. Reconstruction-based approaches attempt to reconstruct the input image from intermediate representations or the final semantic segmentation, intuiting that regions that are difficult to reconstruct are most likely OOD [34], [35], [36], [37], [38], [32]. Our approach is most similar to Di Biase, et al. [32], in that we use a network to reason about the segmentation network's confidence, but we do not require an expensive secondary reconstruction-based pipeline. Other approaches reason about temporal information [39], in contrast to our method that only requires a single frame. A survey of different anomaly detection approaches is available from Bogdoll et al. [40].

Maximum Entropy Many methods attempt to exploit the entropy of neural network outputs for OOD detection. Prior work has regularized the network by penalizing low entropy scores [26] or training to maximize the entropy on known OOD images [19], [41]. We take a similar maximum entropy approach to increase the entropy of the base segmentation net on OOD data, and thus refine the inputs to our metacognitive network, although we leverage synthetic generation to remove the dependency on known OOD training samples.

Synthetic OOD Data Generation Throughout this paper, we distinguish between three types of data: ID, OOD, and synthetic OOD. We define *ID data* as all data available to an agent for its segmentation task at training, *OOD data* as any data the agent will encounter after deployment beyond the ID classes, and *synthetic OOD data* as generated data whose use in training improves OOD detection. We make these distinctions to motivate our means of refining models over a limited ID training set, which we discuss below and detail in Section III-C.

Generating synthetic OOD data is a common approach for training OOD detection when OOD data is inaccessible. It is typically used for image classification, but is underexplored for segmentation. Several approaches generate synthetic OOD data for image classification by interpolating between examples in the ID dataset [42], [41], [43], or performing image transformations to corrupt ID data [44]. These techniques allow for easily generating large quantities of data likely to be outside of the training data distribution, but the generated images may not be visually realistic. Further, it is unclear how to apply the interpolation techniques to segmentation, where instances of different classes are completely different shapes. Such an approach would have to address how to select instance pairs for interpolation, and how to realistically compose full images from corrupted class instances together with ID data. Generative models have also been used to produce synthetic OOD data for image classification [45] by training a generator model capable of producing fully-artificial OOD images, which can be time- and resource-intensive. Further, adapting generative approaches for classification to semantic segmentation problems is non-trivial.

In contrast to the above methods, our approach generates synthetic OOD data by heavily blurring a random subset of ID class instances into something unrecognizable as the ID data (similar to Hebbalaguppe et al. [44]), creating mixed synthetic OOD and ID training images. This does not require training a resource-intensive generator, and yields synthetic OOD data that fit directly among ID data as context.

III. Approach

Our MEMOS framework is comprised of two components: 1) a standard base segmentation network fine-tuned with maximum entropy training (detailed in Section III-B) over ID and generated synthetic OOD data (see Section III-C), and 2) a metacognitive network module (Section III-A) that reasons about the base model's predictions in order to identify OOD instances. The full framework is shown in Figure 2. The core of the approach is the novel metacognitive network, which we design as a modular component that can re-use training data from the base model. This module sits on top of the fine-tuned base model, using its class predictions, their dispersion, and their spatial relationships to each other to generate a binary mask that grades the quality and correctness of the base segmentation net's predictions.

Our key insight is that semantic segmentation networks are originally trained only for segmentation, not OOD detection. By adding a lightweight metacognitive network that makes

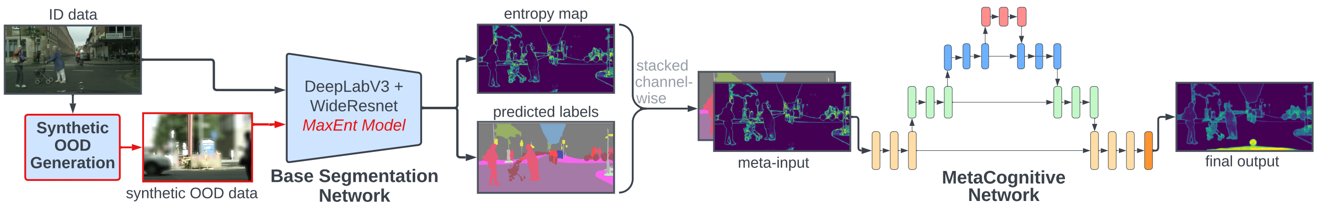


Fig. 2: An overview of our MEMOS framework. Synthetic OOD data is generated and used to fine-tune the base segmentation network via maximum entropy training. Predicted labels and the entropy map are stacked as channels for input into the metacognitive network, which generates a final uncertainty mask for OOD detection.

improved quality judgements, we can use these quality judgements directly for OOD detection. Prior work has established that the entropy of network predictions is highly indicative of uncertainty and can be relied upon for OOD detection [19], [41], [43], and our metacognitive network offers improved uncertainty judgements. Additionally, the metacognitive network relies on entropy as an input, and thus improves further as the entropy of the segmentation network's outputs are better calibrated on both ID and OOD data. To achieve this improvement in entropy estimates, we fine-tune the base segmentation network using maximum entropy training over a dataset composed of ID data and our generated synthetic OOD data. We detail the fine-tuning process and the resulting MaxEnt segmentation network in Section III-B, and the supporting synthetic OOD data generation in Section III-C.

A. The Metacognitive Network

Most approaches make OOD detections by simply thresholding uncertainty measures [17], [18] output either by a single network or an ensemble. However, such simple thresholding fails to take contextual cues into account. While an end-to-end segmentation model trained for OOD detection has the necessary context to produce accurate uncertainty measures, this does not always happen in practice. For example, the OOD dog in Figure 1(c) is difficult to discern from simple pixel-wise thresholding alone. However, all information needed to make this decision is present in the segmentation network's output (Figures 1(b) and (c)), suggesting that this is a failure of it considering its own segmentation confidence, and suggesting a metacognitive approach would enable more effective OOD detection (Figure 1(d)).

The metacognitive network module, which appends to any base segmentation network, computes a quantitative uncertainty estimate for each prediction, which can then be used for OOD detection. We train the metacognitive network using the binary cross entropy loss function $\mathcal{L}_{\text{bce}}$ as follows:

$$\arg\min_{\phi} \sum_{x \in D_{\text{id}}} \mathcal{L}_{\text{bce}}(\phi, g(x)) \ . \tag{1}$$

We generate the metacognitive input $g(x)$ by stacking the pixel-wise entropy and predicted class channel-wise, keeping the original image structure to maintain spatial context:

$$g(x) = \arg\max_{y \in Classes} f_{seg}(\theta, x)^\frown \Omega_{ent}(\theta, x) \ , \tag{2}$$

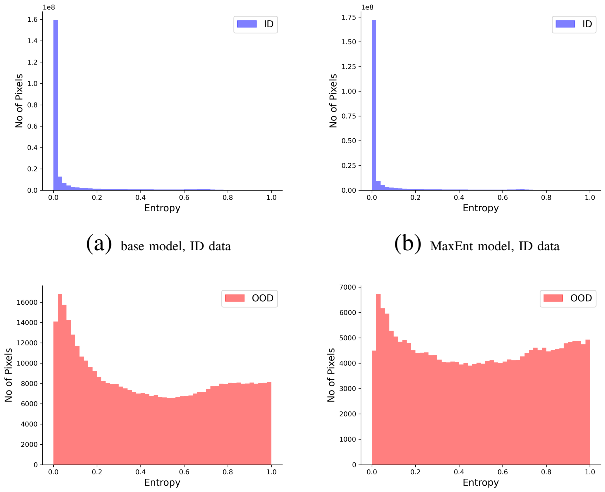
where f_{seg} is a base segmentation network with parameters θ, ϕ are the parameters for the metacognitive network, D_{id} is the ID training data, and $\Omega_{\text{ent}}(\theta, x)$ is the entropy of the conditional distribution $p_\theta(\mathbf{y}|\mathbf{x})$ produced by a network with parameters θ over classes $\mathbf{y}$ for the input $\mathbf{x}$, defined as

$$\Omega_{\text{ent}}(\theta, x) = -\sum_{i \in \text{Classes}} p_\theta(\mathbf{y}_i|\mathbf{x}) \log p_\theta(\mathbf{y}_i|\mathbf{x}) \ . \tag{3}$$

The intuition behind this input structure is that reasoning about the distribution of certainty in the model's predictions is a strong indicator of novelty [19], and including the predicted classes allows the metacognitive network to condition its reasoning based on different commonly observed arrangements of classes. For example, we would expect higher entropy on the border of similar correctly classified objects, such as grass bordering a bush, but would expect very low entropy on the borders of easily distinguished classes, such as grass bordering a building. Additionally, we do not include any of the original image data to prevent overfitting while also keeping the metacognitive network space and time efficient. As it computes its input from class predictions that essentially come from a black box, the metacognitive network does not depend on the architecture of the segmentation model and can be used with any compatible segmentation pipeline.

We use a U-Net architecture [46] for our metacognitive network (Figure 2) to enable the network to reason about both local features and the larger scale image context. We use a subset of the training data and its corresponding predictions from the segmentation network to train the metacognitive network. We compute a target label as a binary mask indicating correct vs. incorrect segmentation network predictions, where 0 indicates a correct prediction and 1 indicates an incorrect prediction. The metacognitive network predicts a soft mask (Figure 1(d)), consisting of values in $[0, 1]$ for each pixel in the corresponding input image. An estimate closer to 1 indicates a poor, highly uncertain prediction; as the value moves closer to 0, the metacognitive network is more certain about the segmentation prediction. The network is trained using a binary cross-entropy loss to learn a function of entropy, correlated with each predicted class, aided by the context of neighbouring pixels, to determine the quality of the segmentation prediction. Pixels with high uncertainty can then be identified as OOD data.

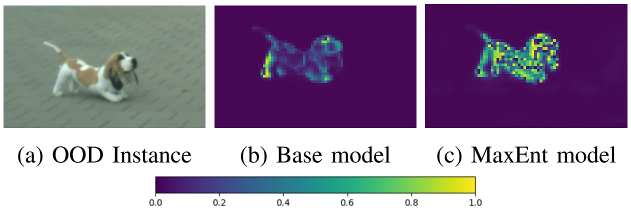
OOD detection could likely be improved further by filter-



(a) base model, ID data (b) MaxEnt model, ID data

(c) base model, OOD data (d) MaxEnt model, OOD data

Fig. 3: Histograms of entropy distribution across ID and OOD data for the base segmentation model and our MaxEnt segmentation model.



(a) OOD Instance (b) Base model (c) MaxEnt model

Fig. 4: Entropy heatmap of (a) an OOD instance for (b) the standard segmentation model and (c) the segmentation model with maximum entropy fine-tuning. Yellow pixels indicate high entropy and uncertainty.

ing out contiguous detected regions under a size threshold—a post-processing step. We do not evaluate post-processing approaches as shown in other work [19] to reduce the dependent variables evaluated in the scope of this paper, and leave it to future work. However, we do expect many post-processing approaches could benefit this framework, as the metacognitive network's inclusion of context from neighboring pixel entropy and class labels enables it to detect more contiguous regions, as seen in Figures 1(c) and (d). This can be both a benefit and a hindrance: the now-contiguous dog is correctly labeled OOD and the contiguous bus mirror is incorrectly labeled OOD. However, our ablation studies in Section IV show this is more beneficial overall.

B. Maximum Entropy Segmentation Model

Large DNNs are known to be overconfident about their predictions [24], [47], resulting in unreliable uncertainty estimates for downstream components. This is a concern for OOD detection through our metacognitive network, since the base segmentation network typically underestimates the entropy for OOD data due to overconfidence. The result is that the base network is skewed towards low-entropy predictions, even on OOD data, as shown in Figure 3(c). To alleviate this issue, we calibrate the base segmentation model to generate high-entropy predictions for OOD instances and low-entropy predictions for ID instances by fine tuning it using synthetic OOD data; we refer to this as the MaxEnt segmentation model. Figure 3(c) and (d) show that our MaxEnt model does produce a broader entropy distribution over OOD data from the same samples, including more high-entropy predictions, when compared to the original segmentation network. Additionally, we see improvement in the frequency of low-entropy predictions for ID samples in comparison to the base segmentation model (Figures 3(a) and (b)). Figure 4 shows a qualitative example of the difference between the base and calibrated MaxEnt models, the latter of which has higher entropy predictions for the OOD puppy.

To learn the MaxEnt segmentation model, we add a maximum entropy regularizer [26] to the standard cross entropy loss $\mathcal{L}_{\text{ce}}$. The resulting optimization is given by

$$\arg\min_{\theta} \sum_{x \in D_{\text{id}}} \mathcal{L}_{\text{ce}}(\theta, x) - \underbrace{\lambda \sum_{x \in D_{\text{ood}}} \Omega_{\text{ent}}(\theta, x)}_{\text{MaxEnt regularizer}} \;, \tag{4}$$

where D_{id} represents the ID training data, D_{ood} represents the synthetic OOD training data described in Section III-C, λ is the regularization parameter, and Ω_{ent} is the entropy as defined in Equation 3. The regularization term encourages the segmentation network to maximise entropy for the predictions when input $\mathbf{x}$ is synthetic OOD data. Training the segmentation by optimizing Equation 4 calibrates the network to make low-entropy predictions for ID samples and high-entropy predictions for synthetic OOD samples, which should translate to actual OOD samples at deployment.

C. Synthetic OOD Data Generation

The MaxEnt approach raises an important question, namely how to access OOD training data for the *MaxEnt regularizer* to significantly effect the model's performance? Some approaches train using a predetermined set of known OOD data [19], but we treat the separation of ID and OOD data as inviolable, as we define OOD data as unknown pre-deployment. One could divide the training data to create a subset of OOD classes [48], but this would reduce the amount of ID training data and impact ID performance. We circumvent this issue by using a data augmentation approach based on transforming ID images for classification [42], [41], [43], [44]. To our knowledge, this is the first approach that generates synthetic OOD data from ID data for segmentation. Specifically, we create a randomly sampled subset of the training data $D_{\text{sub}} \subset D_{\text{id}}$, and a subset of the ID classes, C_{sub}. For every image in D_{sub}, we apply a Gaussian blur on all pixels belonging to classes not in C_{sub}. D_{sub} is added to the train set as additional data and the Gaussian blurred pixels in every image serve as synthetically generated OOD data, D_{ood}.

This data augmentation approach has a few advantages. First, it ensures our synthetic OOD data is directly based on natural images, and produces objects of a similar size, resolution, and intensity. Second, it keeps some ID data belonging to C_{sub} in each training image to provide context for the synthetic OOD instances. Last, it also provides training data for ID classes that are in the presence of synthetic OOD instances, simulating a condition we expect to occur after the model is trained and deployed. We show that this data augmentation approach successfully increases entropy for OOD samples in Figure 3, an example of which is shown in Figure 4, and we evaluate its effect on OOD performance via ablation studies in Section IV.

IV. EXPERIMENTS AND RESULTS

Evaluating the performance of various OOD detection methods requires both ID and OOD datasets. We use Cityscapes [49] as our ID data, which consists of street scenes from 50 different cities; we train both our MaxEnt and metacognitive networks using its 2,975 image trainset. We evaluate OOD detection on the following established OOD detection for semantic segmentation benchmarks from the Lost and Found [50] and Fishyscapes [51] benchmarks, which comprise of road and street scenes with anomalies not present in Cityscapes: the test split of Lost and Found dataset, the Fishyscapes Static dataset, and the Fishyscapes (FS) Lost and Found dataset. Wherever possible, we use the same base segmentation model for comparison across experiment conditions. Specifically, we use the state-of-the-art DeepLabV3+ with a WideResnet38 backbone using pretrained weights [1]. For further details, experiment code, models, and hyperparameters, see our repository[1] .

A. MaxEnt Model and Metacognitive Training Details

We begin by training the MaxEnt model (Section III-B) using the Cityscapes trainset as D_{id} along with a synthetic OOD dataset D_{ood}. To generate D_{ood}, we randomly sample 500 images from the Cityscapes trainset, and select 12 classes ($|C_{\text{sub}}| = 12$) to remain in-distribution. The classes were selected randomly to avoid bias. We found that selecting approximately half of the classes to generate synthetic OOD data gave the best performance, and hypothesize this is effective as it provides an even balance of contextualized synthetic OOD data in the individual images.

Next, we train the metacognitive network, as described in Section III-A, constructing a trainset by randomly sampling 500 images and their corresponding ground truth labels from D_{id}. We compute the final training labels from their corresponding ground truth labels and the output of the previously trained MaxEnt network.

During evaluation, we pass each test sample from the OOD datasets through the MaxEnt segmentation model, construct the input $g(x)$, and pass $g(x)$ through the metacognitive network to predict the final detection mask, as shown in Figure 2.

B. Evaluation Procedure

We evaluate our method against the following baselines: directly thresholding the entropy of the base model's prediction (*Entropy*), softmax thresholding (*Softmax*) [17], ensemble consensus over three networks *(Ensemble)*, *Learned Density* [51] proposed by the authors of the Fishyscapes benchmark, ODIN [18], and Entropy Maximization using Coco dataset samples (*EM-Coco*) [19]. Note that we do not include generative baselines that require training additional large models, such as Synboost [32], as their inference time is orders of magnitude greater than our method and baselines, making them impractical for use with robots that require real-time perception loops. See Table II for more details.

All approaches use the same base model, with the following modifications: *Ensemble* trains three models over three random seeds, MaxEnt finetunes the base model on our synthetic OOD data, ODIN uses the base model with additional hyperparameters finetuned on an OOD validation dataset (reported in [18]), and *EM-Coco* uses its author-provided pre-trained weights. Additionally, we use the metric values reported by [51] for *Learned Density*, since we were unable to find an implementation, and results were reported using the same base model as in this work. The metrics reported for all baselines are averaged over three random seeds[2]. All methods are evaluated on an input image size of 1024 x 2048.

We evaluate OOD detection performance using AUPRC, which gives greater importance to OOD detection by accounting for class imbalance. We treat OOD labels as the positive class. Additionally, we report the mean intersection-over-union (mIoU) for the ID cityscapes validation set, and the false positive rate for a 0.95 true positive rate (FPR-95) for OOD detection. These metrics serve as sanity checks to verify that the models can still perform effective semantic segmentation for ID classes while performing OOD detection. We also report inference time in milliseconds.

C. Results

Over all of the benchmarks, our MEMOS framework outperforms all baselines that use no OOD data at training time, as shown in the top half of Table I. As MEMOS is additive to the simple *Entropy* baseline, significantly outperforming *Entropy* indicates that our MaxEnt model and/or metacognitive network are beneficial for OOD detection, which we evaluate further in the ablation studies below. Additionally, when some OOD data *is* available at training time, shown in the bottom half of Table I, adding our metacognitive network to *EM-Coco* improves performance over all other methods, showing both our modules's effectiveness for OOD detection and also validating its compatibility with other segmentation methods. Note that MEMOS without access to any OOD data also outperforms ODIN, and approaches the performance of *EM-Coco* in the Lost and Found benchmark[3].

ID detection on the Cityscapes validation, as shown by the mIoU column in Table I demonstrates that our framework

[1] `https://github.com/meghna30/metacognitive_segmentation/tree/main`

[2] Learned Density and *EM-Coco* and *EM-Coco&Meta* are reported over one random seed, as they use author-provided weights/results.

[3] For Fishyscapes benchmark, we attribute *EM-Coco*'s high performance to data overlap with their additional OOD validation set, which contains classes similar to the OOD instances in the Fishyscapes datasets.

TABLE I: Performance of OOD detection methods across Lost and Found and Fishyscapes benchmarks. Methods below the horizontal line require access to OOD data during training.

OOD Detection Method	OOD Data?	Val mIoU	Lost and Found		FS Lost and Found		Fishyscapes - Static	
			AUPRC	FPR-95	AUPRC	FPR-95	AUPRC	FPR-95
Softmax		0.89±.008	0.26±.002	0.17±.023	0.05±.012	0.36±.055	0.18±.045	0.19±.016
Entropy		0.89±.008	0.44±.001	0.22±.102	0.13±.031	0.33±.061	0.35±.034	0.18±.018
Ensemble		0.88±.000	0.07±.005	0.26±.087	0.02±.302	0.29±.017	0.32±.021	0.16±.006
Learned Density		0.80	—	—	0.04	0.47	0.62	0.17
MEMOS (Ours)		0.87±.000	**0.70±.012**	0.12±.037	**0.23±.005**	0.46±.172	**0.65±.045**	0.35±.126
ODIN	✓	0.89±.008	0.56±.008	0.12±.009	0.15±.014	0.27±.11	0.13±.03	0.49±.016
EM-Coco	✓	0.89	0.76	0.095	0.41	0.37	0.81	0.094
EM-Coco&Meta	✓	0.89	**0.79**	0.009	**0.43**	0.43	**0.84**	0.11

TABLE II: Inference time

Method	Inf. Time (ms)
Synboost	$1,055.5$
Add'l Conv.	816.9
DeepLabV3	24.5
Softmax	24.5
Entropy	24.5
Ensemble	24.5×3
ODIN	$24.5 + 595.7$
EM-COCO	24.5
MEMOS (Ours)	$24.5 + 6.4$

TABLE III: Ablation studies across Lost and Found and Fishyscapes benchmarks

OOD Detection	Val mIoU	Lost and Found		FS Lost and Found		Fishyscapes - Static	
		AUPRC	FPR-95	AUPRC	FPR-95	AUPRC	FPR-95
Entropy	0.89±.008	0.44±.001	0.22±.102	0.13±.031	0.33±.061	0.35±.034	0.18±.018
Add'l Conv.	0.80±.008	0.45±0.021	0.30±0.016	0.12±0.063	0.41±0.057	0.22±0.034	0.25±0.092
Metacognitive-Only	0.85±.009	0.48±.014	0.15±.031	0.13±.033	0.55±.1	0.39±.043	0.26±.062
MaxEnt	0.90±.000	0.64±.0.005	0.28±.066	0.22±.045	0.24±.013	0.61±.060	0.147±.010
MEMOS (Ours)	0.87±.000	**0.70±.012**	0.12±.037	**0.23±.005**	0.46±.172	**0.65±.045**	0.35±.126

does not hinder ID semantic segmentation performance; there is no tradeoff when improving OOD detection performance. FPR-95 varies considerably across all methods and datasets, but when considered with the high mIoU performance on ID data, it shows that all of the methods we evaluated are able to perform OOD detection without sacrificing performance on ID data.

When considering open-world robotics applications [5], [6], [7], [8], [9], [10], [11], short inference times are critical. We evaluate inference times for all methods on an NVIDIA RTX 3090. Table II shows that many of the methods we evaluated, including our MEMOS framework, can be deployed in a real-time perception loop of 30-40 Hz. Notable exceptions are *Ensemble* methods, which multiply the inference time by the ensemble size, and the significant computational burden of *ODIN*, generative methods represented by *Synboost*, and simply learning a larger end-to-end base network (*Add'l Conv*, discussed further in the ablation studies below), which are all significantly less practical running closer to 1 Hz. Note, *Add'l Conv*, was run on an NVIDIA RTX A6000 as the computational load was too high for a 3090.

Ablation Studies: We show contributions of MEMOS' individual components via ablation studies summarized in Table III. Both the metacognitive network (*Metacognitive-Only*) and MaxEnt components individually improve the performance of *Entropy*. Combining both components improves performance further, showing that the two components are complementary. We take this as evidence that the entropy calibration of the MaxEnt model improves the detection ability of the metacognitive module, as our design intended.

We also create an additional baseline *Add'l Conv.* by appending additional convolution layers to the base network, with as many parameters as the base plus metacognitive network (*Metacognitive-Only*), and train it using the Cityscpaes dataset. While this baseline does marginally better than *Entropy* due to its larger network size for the Lost and Found dataset, it performs worse than the comparably-sized *Metacognitive-Only* across all datasets. This shows that the structure imposed by our metacognitive approach is beneficial beyond simply increasing the number of network parameters. Further, designing a metacognitive module as a separate network component reduces computational burden as shown in Table II.

V. LIMITATIONS

Our framework assumes that training on synthetic OOD data will generalize sufficiently to actual OOD samples. Our main failure mode is sensitivity to well-calibrated networks—any base model with a poorly calibrated entropy will likely limit the metacognitive network's efficacy. We examined only Gaussian blurring to generate the synthetic OOD images, and suggest evaluating additional image transformations in future work. While we demonstrate significant increase in performance over comparable baselines on standard benchmarks, this work would benefit from real-world OOD detection experiments situated on physical robots to overcome the limitations of such constructed benchmarks.

VI. CONCLUSIONS

We presented the MEMOS framework, consisting of a novel metacognitive network module that can improve OOD detection for state-of-the-art semantic segmentation models by leveraging uncertainty measures and spatial information. We also demonstrate that the performance of the metacognitive module improves considerably when fine-tuning the base segmentation model using maximum entropy training over synthetic OOD data generated in context with ID data, outperforming state-of-the-art OOD detection for segmentation baselines trained with equivalent access to ID data and realistic restrictions on available OOD data. Finally, we show that our framework has a low inference time that is suitable for real-time perception.

References

[1] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8856–8865.

[2] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] X. Li, X. Li, L. Zhang, G. Cheng, J. Shi, Z. Lin, S. Tan, and Y. Tong, "Improving semantic segmentation via decoupled body and edge supervision," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer, 2020, pp. 435–452.

[4] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[5] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, "Identifying unknown instances for autonomous driving," in *Conference on Robot Learning*. PMLR, 2020, pp. 384–393.

[6] L. Balasubramanian, F. Kruber, M. Botsch, and K. Deng, "Open-set recognition based on the combination of deep learning and ensemble method for detecting unknown traffic scenarios," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 674–681.

[7] A. Boteanu, D. Kent, A. Mohseni-Kabir, C. Rich, and S. Chernova, "Towards robot adaptability in new situations," in *2015 AAAI fall symposium series*, 2015.

[8] F. Feng, R. H. Chan, X. Shi, Y. Zhang, and Q. She, "Challenges in task incremental learning for assistive robotics," *IEEE Access*, vol. 8, pp. 3434–3441, 2019.

[9] D. Kim, T.-Y. Lin, A. Angelova, I. S. Kweon, and W. Kuo, "Learning open-world object proposals without learning to classify," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5453–5460, 2022.

[10] M. Gunther, S. Cruz, E. M. Rudd, and T. E. Boult, "Toward open-set face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 71–80.

[11] S. Dey and G. Saha, "Addressing the semi-open set dialect recognition problem under resource-efficient considerations," *Speech Communication*, vol. 152, p. 102957, 2023.

[12] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.

[13] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] A. Bendale and T. Boult, "Towards open world recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[15] M. B. Ganapini, M. Campbell, F. Fabiano, L. Horesh, J. Lenchner, A. Loreggia, N. Mattei, F. Rossi, B. Srivastava, and K. B. Venable, "Thinking fast and slow in ai: The role of metacognition," in *Machine Learning, Optimization, and Data Science: 8th International Conference, LOD 2022, Certosa di Pontignano, Italy, September 18–22, 2022, Revised Selected Papers, Part II*. Springer, 2023, pp. 502–509.

[16] M. Gummadi, D. Kent, J. A. Mendez, and E. Eaton, "Shels: Exclusive feature sets for novelty detection and continual learning without class boundaries," in *Conference on Lifelong Learning Agents*. PMLR, 2022, pp. 1065–1085.

[17] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[18] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[19] R. Chan, M. Rottmann, and H. Gottschalk, "Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation," in *Proceedings of the ieee/cvf international conference on computer vision*, 2021, pp. 5128–5137.

[20] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[21] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*. Springer, 2000, pp. 1–15.

[22] B. Johnson, "Metacognition for artificial intelligence system safety – an approach to safe and desired behavior," *Safety Science*, vol. 151, p. 105743, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753522000832

[23] M. D. Schmill, M. L. Anderson, S. Fults, D. Josyula, T. Oates, D. Perlis, H. Shahri, S. Wilson, and D. Wright, "12 the metacognitive loop and reasoning about anomalies," *Metareasoning: Thinking about thinking*, p. 183, 2011.

[24] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[25] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," *arXiv preprint arXiv:1812.04606*, 2018.

[26] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *arXiv preprint arXiv:1701.06548*, 2017.

[27] P. Bevandić, I. Krešo, M. Oršić, and S. Šegvić, "Simultaneous semantic segmentation and outlier detection in presence of domain shift," in *Pattern Recognition: 41st DAGM German Conference, DAGM GCPR 2019, Dortmund, Germany, September 10–13, 2019, Proceedings 41*. Springer, 2019, pp. 33–47.

[28] ——, "Dense outlier detection and open-set recognition based on training with noisy negative images," *arXiv preprint arXiv:2101.09193*, 2021.

[29] X. Du, Z. Wang, M. Cai, and Y. Li, "Vos: Learning what you don't know by virtual outlier synthesis," *arXiv preprint arXiv:2202.01197*, 2022.

[30] S. Jung, J. Lee, D. Gwak, S. Choi, and J. Choo, "Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 425–15 434.

[31] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song, "Scaling out-of-distribution detection for real-world settings," in *International Conference on Machine Learning*. PMLR, 2022, pp. 8759–8773.

[32] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena, "Pixel-wise anomaly detection in complex driving scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 918–16 927.

[33] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[34] C. Creusot and A. Munawar, "Real-time small obstacle detection on highways using compressive rbm road reconstruction," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 162–167.

[35] K. Lis, K. Nakka, P. Fua, and M. Salzmann, "Detecting the unexpected via image resynthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2152–2161.

[36] K. Lis, S. Honari, P. Fua, and M. Salzmann, "Detecting road obstacles by erasing them," *arXiv preprint arXiv:2012.13633*, 2020.

[37] T. Ohgushi, K. Horiguchi, and M. Yamanaka, "Road obstacle detection method based on an autoencoder with semantic segmentation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[38] T. Vojir, T. Šipka, R. Aljundi, N. Chumerin, D. O. Reino, and J. Matas, "Road anomaly detection by partial image reconstruction with segmentation coupling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 651–15 660.

[39] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun, "Efficient uncertainty estimation for semantic segmentation in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 520–535.

[40] D. Bogdoll, M. Nitsche, and J. M. Zöllner, "Anomaly detection in autonomous driving: A survey," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4488–4499.

[41] F. Pinto, H. Yang, S.-N. Lim, P. H. Torr, and P. K. Dokania, "Mixmaxent: improving accuracy and uncertainty estimates of deterministic neural networks," 2021.

[42] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[43] F. Pinto, H. Yang, S.-N. Lim, P. H. Torr, and P. K. Dokania, "Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness," *arXiv preprint arXiv:2206.14502*, 2022.

[44] R. Hebbalaguppe, S. S. Ghosal, J. Prakash, H. Khadilkar, and C. Arora, “A novel data augmentation technique for out-of-distribution sample detection using compounded corruptions,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part III*. Springer, 2023, pp. 529–545.

[45] K. Lee, H. Lee, K. Lee, and J. Shin, “Training confidence-calibrated classifiers for detecting out-of-distribution samples,” *arXiv preprint arXiv:1711.09325*, 2017.

[46] O. Ronneberger, P.Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a

[47] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. Torr, and P. Dokania, “Calibrating deep neural networks using focal loss,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 288–15 299, 2020.

[48] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, “Out-of-distribution detection using an ensemble of self supervised leave-out classifiers,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 550–564.

[49] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[50] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester, “Lost and found: detecting small road hazards for self-driving vehicles,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1099–1106.

[51] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, “The fishyscapes benchmark: Measuring blind spots in semantic segmentation,” *International Journal of Computer Vision*, vol. 129, pp. 3119–3135, 2021.