2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Masked γ-SSL: Learning Uncertainty Estimation via Masked Image Modeling

David S. W. Williams, Matthew Gadd, Paul Newman and Daniele De Martini
Oxford Robotics Institute, Dept. of Engineering Science, University of Oxford, UK.
{dw,mattgadd,pnewman,daniele}@robots.ox.ac.uk

***Abstract*— This work proposes a semantic segmentation network that produces high-quality uncertainty estimates in a single forward pass. We exploit general representations from foundation models and unlabelled datasets through a Masked Image Modeling (MIM) approach, which is robust to augmentation hyper-parameters and simpler than previous techniques. For neural networks used in safety-critical applications, bias in the training data can lead to errors; therefore it is crucial to understand a network's limitations at run time and act accordingly. To this end, we test our proposed method on a number of test domains including the SAX Segmentation benchmark, which includes labelled test data from dense urban, rural and off-road driving domains. The proposed method consistently outperforms uncertainty estimation and Out-of-Distribution (OoD) techniques on this difficult benchmark.**

***Index Terms*— Segmentation, Scene Understanding, Introspection, Performance Assessment, Deep Learning, Autonomous Vehicles**

I. Introduction

Semantic segmentation gives robots a useful semantic understanding of their surroundings. Segmentation networks are typically trained with data annotated with the relevant semantic concepts; however, when presented with instances of classes with a different appearance or instances of a different class, they are prone to making mistakes. In the safety-critical context of perception in robotics, this is often unacceptable, and thus, methods must be developed to help neural networks understand their limitations. This work presents a method that produces high-quality uncertainty estimates for a semantic segmentation task, focusing on the benefits of leveraging general representations from foundation models. In addition, we exploit easy-to-collect, unlabelled domain-specific datasets. Specifically, given a labelled training dataset in one domain (a.k.a. the *source domain*), we are interested in mitigating the segmentation error rate on a distributionally-shifted unlabelled domain (a.k.a. the *target domain*).

Recently, it has become effective to perform self-supervised training on diverse image datasets which try to approximate the set of all natural images. The intent is to learn a general semantic representation of the provided unlabelled images. A primary benefit is that these models can be fine-tuned to solve specific image-based perception tasks, giving a particularly large boost in performance to tasks with small amounts of data. If we view distributional shifts as a problem of insufficient labelled data, this suggests that they can be ameliorated by leveraging these network representations.

This work was supported by EPSRC Programme Grant "From Sensing to Collaboration" (EP/V000748/1).

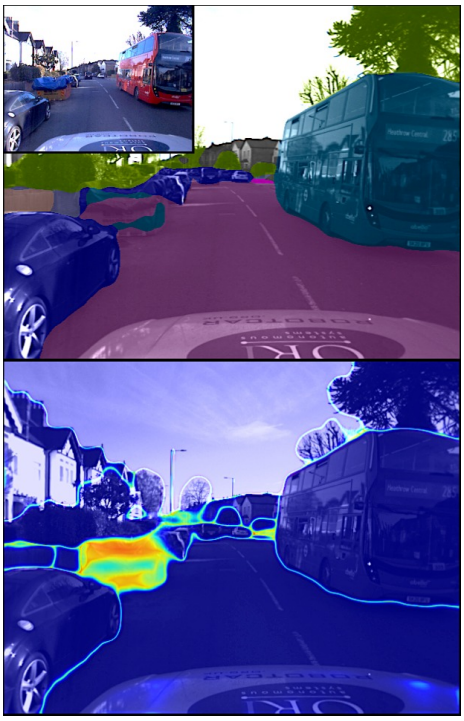

Fig. 1. Our proposed method jointly performs high-quality semantic segmentation (top) *and* pixel-wise uncertainty estimation (bottom) on an image (top, top left) from the SAX London test dataset. Note a dumpster (an undefined semantic class) is *inaccurately* segmented, however our network is correspondingly uncertain (bottom, yellow and red), while the rest of the segmentation is accurate and certain (bottom, blue).

In this work, we define our task of interest as the semantic segmentation of images from a source domain for which labelled training images are provided. The encoder of a segmentation network is initialised with a general representation – namely DINOv2 [1] – and then the entire network is specialised to solve the task at hand. A key question is then: as a result of this necessary model specialisation, how much does the quality of uncertainty estimation degrade on images from different target domains?

This work investigates this question and presents a method that uses Masked Image Modeling (MIM) on unlabelled images to investigate a given segmentation network's representation of distributionally-shifted data *during training*, in order to learn high-quality uncertainty estimation.

MIM has recently been used to learn representations from images in a self-supervised manner by removing patches of the input image and having the network predict the

information content of these masked regions. This task requires a detailed understanding of the unmasked patches, and the semantic interactions between masked and unmasked patches, and is thus also predictive of network robustness.

When presented with a masked, unfamiliar image, if the network cannot extract sufficient salient information from the unmasked patches to accurately segment the masked patches, the segmentation is likely to differ from that of the same image, but unmasked. Without the requirement of labelled images, this gives us the signal required to discover the limits of the network, and how image appearance relates to segmentation quality.

This work is based on our previous work [2], beyond which this work's contributions are as follows:

- A MIM-based method for learning uncertainty estimation from unlabelled data, which is simpler, produces higher quality uncertainty estimates and with less sensitivity to augmentation hyperparameters;
- An empirical investigation of the effects of distributional shift on networks with general image representations.

II. Related Work

A. Epistemic Uncertainty Estimation

As epistemic uncertainty is defined as inherent to the model, and not the data, this set of literature concerns itself with modeling the distribution of model parameters. A full Bayesian treatment is intractable, however a number of possible approximations are made in [3]–[5], and notably in Monte Carlo Dropout networks [6]. Uncertainty estimation is performed by sampling sets of parameters, segmenting with each, and then evaluating the segmentation consistency. An alternative that produces high-quality uncertainty estimates is to model this model distribution with an ensemble [7].

The major drawback to these methods for robotics is that they require multiple forward passes of a network at runtime, greatly increasing the latency, which is often unacceptable in safety-critical contexts. Our proposed method is therefore designed to use only a single forward pass at runtime.

B. Aleatoric Uncertainty Estimation

For aleatoric uncertainty estimation, the uncertainty is inherent to the data, and not the model. The focus is therefore to learn the appearance of regions that are likely to result in error. [8], [9] parameterise the network output with a Gaussian distribution, such that the network is softly-constrained to either estimate the quantity of interest, or to output a large variance. [10] parameterises the features from each layer as a distribution, using assumed density filtering.

The above uses labelled images, such that ground-truth error is known during training. However [11] uses unlabelled images to solve both a geometric matching problem and uncertainty estimation by parameterising the output as an exponential distribution. This work, along with [2], also leverages unlabelled images to learn uncertainty estimation, but for a semantic segmentation task.

C. Single-Pass Uncertainty Methods

Deterministic uncertainty methods (DUMs) [12]–[14] estimate uncertainty in a single forward pass with the use of spectral normalisation layers [15], which restrict the Lipschitz constant of the model with the hope that large semantic differences in the input lead to large distances in feature space. Alternatively, [16] injects noise during model training, and calculates the approximate feature covariances as a measure of epistemic uncertainty. [17] trains a set of orthonormal linear layers on top of a feature extractor, such that each feature from the training dataset is mapped to zero - thus epistemic uncertainty is represented by the average output of linear layers. [18] shows that regularisation from the Deep Variational Bottleneck method [19] allows for improved single-pass uncertainty estimation.

Much like the cited DUMs and [18], this work turns uncertainty estimation into a representation learning problem, which is solved by refining a general feature representation using MIM uncertainty training and unlabelled Out-of-Distribution (OoD) data.

D. Out-of-Distribution Detection

OoD detection methods find images that are distinct from the distribution defined by a given dataset. A common procedure is to train a network on this dataset, then freeze it and develop an inference method that leverages the learned representation. [20] calculates the max softmax score, [21] uses a Gaussian Mixture model and [22] calculates a score from both features and logits.

Another form of OoD detection explicitly trains a network to separate labelled training and OoD data, by training on a dataset of curated OoD images, [23], [24]. However, [25] and [26], show that the greater the difference between the source and OoD datasets, the worse the OoD detection. We take inspiration from this, and use a dataset which is composed of images with in-distribution, near-distribution and out-of-distribution instances all within the same images.

E. Masked Image Modelling

Recently, MIM has been utilised in order to learn a representation from diverse image data in a self-supervised manner, in conjunction with the Transformer architecture [27]. [28] trains a vision transformer (ViT) to reconstruct the RGB values for masked patches, while [1], [29] maximise the consistency between extracted masked and unmasked features. [28] masks randomly, while [30] masks semantic parts, and [31] learns a masking policy. These methods are ultimately judged on the network's accuracy on diverse tasks, for which the network is fine-tuned. Instead of model pre-training, this work uses MIM to fine-tune a model trained with the above methods, focusing purely on producing high-quality uncertainty estimates.

III. Preliminaries and Notation

A. Semantic Segmentation and Uncertainty Estimation

Semantic segmentation requires a model, $\mathtt{f}$, to assign a class to each pixel of an RGB image, $I \in \mathbb{R}^{3\times H\times W}$,

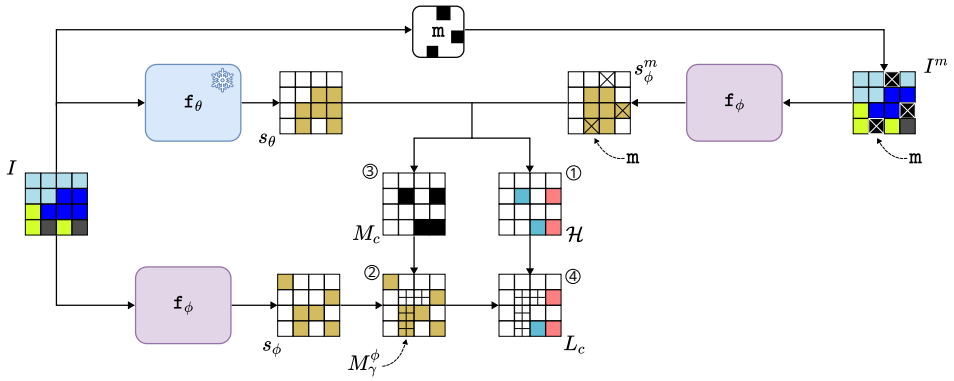

Fig. 2. *Overview of our uncertainty training framework.* It involves frozen network $\mathtt{f}_\theta$ (blue) and the network being trained $\mathtt{f}_\phi$ (purple). $\mathtt{f}_\theta$ has been trained to perform semantic segmentation with images from a labelled domain. Using unlabelled images in a different domain, regions of likely segmentation error are found by comparing masked segmentation s_ϕ^m (masking denoted by $\boxtimes$) and s_θ. Loss L_c refines the uncertainty estimates of $\mathtt{f}_\phi$ by minimising the soft consistency $\mathcal{H}$ ①, but only for regions where $\mathtt{f}_\phi$ is certain, implemented via an uncertainty masking procedure by binary mask M_γ^ϕ ② (denoted by $\boxplus$). The threshold for M_γ^ϕ is calculated using a hard consistency mask ③, giving the final loss ④.

resulting in a segmentation mask, $s \in \mathbb{R}^{K\times H\times W}$, where $K = |\mathcal{K}|$, and $\mathcal{K}$ is the set of defined semantic classes. The model, $\mathtt{f} = \mathtt{D} \circ \mathtt{E}$, is composed of a feature encoder, $\mathtt{E}$, and a segmentation decoder, $\mathtt{D}$.

Uncertainty estimation for this task requires a model to also produce a pixel-wise score, representing the likelihood of a pixel being assigned to the wrong class. Therefore, $\mathtt{f}$ also produces a confidence score mask $u \in \mathbb{R}^{H\times W}$.

One method for calculating this score from a segmentation network is to consider the categorical distribution for each pixel (by applying the $\mathrm{softmax}$ function to each pixel location as the final layer) and to calculate the maximum over the probabilities in order to assign a class to each pixel. This value gives us the network confidence, calculated as $u = \max \circ \mathrm{softmax} \circ \mathtt{f}(I)$ and, for completeness, $s = \mathrm{argmax}\ \mathtt{f}(I)$. Given a threshold γ, we can then calculate a binary confidence mask $M_\gamma \in \{0,1\}^{H\times W}$ which is defined such that for the pixel i:

$$M_\gamma^{(i)} = u^{(i)} > \gamma \tag{1}$$

Pixels which are $\mathtt{1}$ in this mask have confident class predictions, while $\mathtt{0}$ have uncertain class predictions.

Unless otherwise stated, uncertainty is estimated in this way, and this is the quantity that we seek to improve with uncertainty training in this work.

B. Overview of the Proposed Approach

The method in this work is part of a three-step training process: pre-training, task learning, and uncertainty training; these are described in more detail in the following. This work proposes a method for the final of the three steps.

1) Pretraining: The first step trains the encoder to produce a general feature representation of natural images. This relates to the pre-training of the encoder $\mathtt{E}$ component of $\mathtt{f}_\theta$ and $\mathtt{f}_\phi$ to build a good initialisation for the further specialisation of $\mathtt{f}_\theta$ and $\mathtt{f}_\phi$ during the next steps.

The benefits of this are as follows: (1) this step is only done once, allowing any subsequent task learning to be achieved more quickly and easily (2) if only a relatively small labelled dataset is available, the segmentation quality is likely to be better in the source domain and (3) also in the target domains; (4) the uncertainty estimation will be better in both the source and target domains (see Sec. VI-B).

2) Task Learning: The second step specialises the model to perform the task of interest. In this work, labelled source domain images are used to maximise semantic segmentation quality in a supervised manner. The result is the segmentation network $\mathtt{f}_\theta$ in Fig. 2, whose encoder $\mathtt{E}_\theta$ has been initialised from the result of the first step, and then its weights are frozen after this step (indicated by ❄ in Fig. 2).

3) Uncertainty Training: The objective of this step is to maximise the quality of the model's uncertainty estimates for target domain images. For this reason, this step uses many unlabelled images from target domains, i.e. domains distinct from the source domain. This ensures that even if the segmentation quality degrades due to distributional shift, a model is trained to describe the uncertainty quantitatively so that the system as a whole can act appropriately. The result is the segmentation network $\mathtt{f}_\phi$ in Fig. 2, which shares the same architecture of $\mathtt{f}_\theta$, but is parameterised differently. Prior to training, the encoder $\mathtt{E}_\phi$ has been initialised from the result of the first step, and the decoder $\mathtt{D}_\phi$ is randomly initialised.

This final step is our core contribution, and is presented in the next section.

IV. Uncertainty Training with MIM

During the second step, a network with general weights has been fine-tuned to perform semantic segmentation on the labelled source dataset. Although performant on the source domain, this network, $\mathtt{f}_\theta$, has lost some of the generality of the pre-trained model; consequently, both the segmentation and uncertainty estimation quality has decreased on OoD data (see Sec. VI-B).

Now, we want to train a new network that can segment the source domain to the same quality as the trained $\mathtt{f}_\theta$ but produce higher-quality uncertainty estimates on target domains. We thus initialise $\mathtt{f}_\phi$ with the encoder from the first step and a random decoder. This allows us to start with a general model, which can be specialised to segment the source domain well but remain general enough to perform

uncertainty estimation on the target domains. We hypothesise that generality is important for uncertainty estimation because, although it does not require the recognition of all semantic objects, it does require their detection in order to represent them distinctly from the in-distribution classes.

A. Learning Uncertainty Estimation

This work uses a method based on [2], in which the core of this method are more thoroughly presented. Without labels, we use the assumption that segmentation consistency over image augmentation approximates ground-truth accuracy – i.e. if we have two networks that assign two corresponding pixels from two views of the same image to the same class, this is likely to be accurate; otherwise, it is not.

This assumption draws from works in self-supervised learning works [1], [29], [32], where networks are trained under the assumption that *maximising* their consistency over both crop-and-resize (C&R) and masking augmentations, leads to *maximising* task performance. Here, the objective is instead to learn to *detect* inconsistency, in order to *detect* when the frozen network exhibits bad task performance. Unlike [2], this work defines a masking policy $\mathtt{m}$, which produces masked image $I^m = \mathtt{m}(I)$ (see Sec. IV-B for details).

Firstly, $\mathtt{f}_\phi$ segments I and I^m to produce s_ϕ and s_ϕ^m respectively, while $\mathtt{f}_\theta$ segments I to produce s_θ. Our aim is to train $\mathtt{f}_\phi$ such that either: (1) $\mathtt{f}_\phi$ produces a masked segmentation s_ϕ^m that is consistent with the unmasked frozen model's segmentation s_θ, and thus s_θ was likely to have been accurate or (2) s_ϕ^m and s_θ are inconsistent, and therefore one or both segmentations are likely incorrect, *however* $\mathtt{f}_\phi$ expresses uncertainty with s_ϕ.

Initially, the soft consistency loss is computed between s_ϕ^m and s_θ, using the cross-entropy function $\mathcal{H}$, seen as ① in Fig. 2. In order to achieve our stated aim, we allow $\mathtt{f}_\phi$ to reduce the loss by masking out pixels which it estimates to be uncertain via low values of $\max[s_\phi]$ (masking depicted as ⊞ in Fig. 2).

We choose to mask the loss in a hard manner requiring a threshold γ to use Eq. (1) to calculate M_γ^ϕ (see ②). The calculation of γ (see [2] for more detail), is such that the mean value of M_γ^ϕ is equal to the mean value of hard consistency mask M_c (see ③), where M_c is calculated as:

$$M_c^{(i)} = \begin{cases} 1 & \text{if } \operatorname{argmax}[s_\theta^{(i)}] = \operatorname{argmax}[s_\phi^{m(i)}] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We calculate γ in this way because the method seeks to find pixels where s_θ is inaccurate, therefore the proportion of certain pixels in M_γ^ϕ should be equal to the proportion of accurate pixels, therefore we use consistency as an approximation. Finally using both ② and ④, we define the following objective:

$$L_{\mathrm{c}} = \frac{\sum_i^{NHW} M_\gamma^{\phi(i)}\ \mathcal{H}[\rho_{\mathrm{T}}(s_\theta^{(i)}), s_\phi^{m(i)}]}{\sum_j^{NHW} M_\gamma^{\phi(j)}} \tag{3}$$

Where $\rho_{\mathrm{T}}(\cdot)$ is a sharpening function [33], where $\mathrm{T} = 0.5$. The latter's inclusion means not only that L_c maximises consistency for certain pixels, but also the increases their certainty, which helps to increase the separation between certain and uncertain pixels.

This uncertainty training is done in conjunction with supervised training on the source domain, such that the total loss is the sum of L_c and the supervised losses found in [34].

B. Masking Procedure

The masking policy, $\mathtt{m}$, needs to be designed such that segmentations are consistent between $\mathtt{f}_\theta$ and $\mathtt{f}_\phi \circ \mathtt{m}$ when the mutually assigned class is accurate. For this reason, masked and unmasked predictions are compared in semantic segmentation space, as opposed to RGB pixel space [28], or an abstract feature space [1], [29].

Defining the masking policy in this work is a significant challenge, due to the nature of the images used. In any given driving image, there are a large number of different semantic objects and the range in the scale of objects is vast, e.g. the difference in scale between a traffic light in the distance and a car a few metres from the camera. This is in contrast to datasets typically used in MIM such as ImageNet [35], which contain one semantic entity of interest per image, with a smaller variation in scale.

The challenge posed by driving images is two-fold: (1) if an entire small object is masked out, the masked image contains no information from which the model can predict its existence (2) if only small regions are masked out, then the task is too easy, as interpolation can be used to maximise consistency, therefore deteriorating our assumption about consistency approximating accuracy.

Our solution is to use a masking policy that chooses mask elements $M^{(i)}$ independently to produce mask $M \in \{0,1\}^{\hat{H}\times\hat{W}}$, where:

$$M^{(i)} \sim \mathrm{Bernoulli}(p_{\mathrm{mask}}) \tag{4}$$

where $p_{\mathrm{mask}} = 0.5$ unless otherwise stated. Choosing elements independently reduces the chances of masking out entire semantic objects, and empirically a range of p_{mask} lead to an appropriate task difficulty (see Sec. VI-C). Additionally, there were no benefits to more complex masking policies in our experiments, such as learning a masking policy that maximised the inconsistency between unmasked and masked segmentations, or masking only uncertain regions. Masks are applied as $\mathtt{m}(I) = M \cdot [\mathtt{E}]_{0:1}(I)$, where $[\mathtt{E}]_{0:1}$ is the patch embedding of encoder $\mathtt{E}$, thus the dimensions of the mask are $\hat{H}, \hat{W} = \frac{H}{P}, \frac{W}{P}$, where P is the patch size of $\mathtt{E}$. Due to the masking scheme's simplicity, our method is both easy to implement and inherently general, as it is not biased to the specific scale or location of the semantic instances in any given dataset.

V. Experimental Setup

A. Network Architecture

The network architecture for this work follows that of Mask2Former [34], comprising an encoder $\mathtt{E}$, a transformer decoder, and a convolutional decoder, grouped in the variable $\mathtt{D}$ in the above discussion. As in DINOv1 [36] and DINOv2 [1], the encoder is a DeiT [37] transformer.

B. Data

In this work, we use three types of data. Firstly, we use labelled images from the source domain. Secondly, we use unlabelled images from domains distinct from the source domain, called the target domains. Lastly, we use labelled images from the target domains for testing.

The labelled source domain dataset is Cityscapes [38] (CS). As described, the target domains ideally require a large corpus of unlabelled training images and a smaller number of labelled test images. The SAX Semantic Segmentation benchmark [2] provides labels for three target domains: London (LDN), New Forest (NF) , and Scotland (SCOT), where this ordering describes an increasing distributional shift w.r.t. Cityscapes. Additionally, we use the BerkeleyDeepDrive [39] (BDD) dataset as a fourth target domain for training and testing. WildDash [40] (WD), a very diverse set of labelled driving images, is used as a test dataset to measure network generalisation, in addition to the validation set of images in Cityscapes (CS).

C. Model Variants

Data: Using the MIM task, we train models on each unlabelled target domain, to evaluate how a model can be trained to be detect error in target, unlabeled domains and how it generalises to unseen target domains. The target domain used can be found in the model name.

Initialisation: We initialise models with either DINOv1 or DINOv2 to determine how the feature representation affects uncertainty estimation, under the assumption that DINOv2 is more general than DINOv1 – as evidenced by its better transfer learning performance to a wider variety of tasks. Denoted as `d2` and `d1` in the model names.

Input Augmentation: We also compare the models trained with a MIM task to models trained with C&R augmentation [2], allowing us to investigate the differences between MIM or C&R in terms of performance and hyperparameter sensitivity.

Freezing: (a) For the C&R task, we investigate the effect of *not* freezing $\mathtt{f}_\theta$, see $\mathtt{f}^+_\theta$, (b) For the supervised baselines, `E` is frozen. In contrast to this work, [2] did not use weights from general pretraining and so the representation of the target domain needed to be substantially improved during training. Freezing `E` is denoted by $\mathtt{E}^*$, and contrasts with networks fully fine-tuned in a supervised manner.

D. Baselines

We compare our method to baselines from uncertainty estimation and OoD detection literature. Firstly, we consider epistemic-uncertainty estimation methods, Monte Carlo Dropout [6] and [7], both initialised with DINOv2, and trained on Cityscapes. These methods are significantly slower to run than both ours and OoD detection methods, and thus cannot be the solution for many mobile robotics applications, however represent a gold-standard for uncertainty estimation.

The chosen OoD methods take a model trained in a supervised manner on the source domain and leverage the learned feature representation by designing an inference procedure that produces an OoD score. The inference procedures chosen involve: (1) a max-softmax score [20] (`MaxS`), and the Mahalanobis distance between mean features from the source dataset and extracted feature from a target domain [21] (`GMM`). In addition to supervised training, these networks are initialised with DINOv1 and DINOv2.

E. Metrics

The quality of uncertainty estimation is evaluated through a misclassification detection task. This is a binary classification problem, whereby the positive and negative label states are *accurate* and *inaccurate* respectively, and the positive and negative prediction states are *certain* and *uncertain* respectively. The ideal model classifies all accurate pixels as certain, and inaccurate pixels as uncertain.

This allows us then to use metrics typically used for binary classification, namely area under PR curves (AUPR) and maximum F_β scores, which we report with the proportion of pixels that are both accurate and certain, $\mathrm{p}(\mathrm{a}, \mathrm{c})$. Due to the above definitions, a model that prioritises precision over recall puts a higher negative cost on pixels that are (*certain*, *inaccurate*). In the context of safety-critical robotics, this is key, resulting in our decision to measure the $F_{0.5}$ score, i.e. a score that prioritises precision over recall.

VI. Results

A. MIM for different target domains

The model with the highest quality uncertainty estimates is our `Mask-d2` networks trained with unlabelled data from same domain as testing. This is evidenced in Fig. 3, Tab. I and Tab. II, in which the former two consider the point at which uncertain pixels are optimally rejected, and the latter considers the full sweep of uncertainty thresholds.

On top of this, the results show that the `Mask-d2` models also generalise effectively to unseen target domains. The `Mask-BDD-d2` model performed best on the diverse unseen WildDash benchmark, with other `Mask-d2` models similar in performance to the epistemic approaches. For every target domain, the `Mask-d2` models outperform the baselines based on a fully fine-tuned segmentation network (`MaxS-d2`,`GMM-d2`), demonstrating how the uncertainty training maintains generality for improved uncertainty estimation.

B. MIM with different weight initialisations

Each `Mask-d2` model outperforms the trained `Mask-d1` models, even when the `Mask-d1` model is trained on the same domain as testing. Additionally, when the encoder was not fine-tuned in `Mask-`$\mathtt{E}^*$, this also relates to better uncertainty estimation.

The higher scores for `Mask-d2` over `MaxS-d2-`$\mathtt{E}^*$, and `Mask-d1` over `MaxS-d1-`$\mathtt{E}^*$ demonstrate that our uncertainty training has effectively improved the representation beyond that of general pretraining, by including task-specific information, and without the over-specialisation seen in `MaxS-d2`.

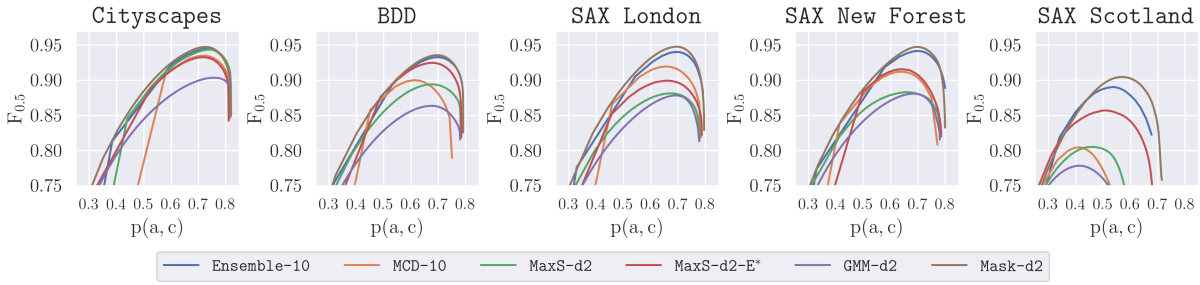

Fig. 3. In these plots, we measure misclassification detection performance using $F_{0.5}$ scores plotted against the proportion of pixels that are certain and accurate $p(a, c)$. The baselines are trained only with labelled Cityscapes data, while our proposed model, `Mask-d2`, leverages unlabelled images from the domain in which testing is occurring. All models are able to perform uncertainty estimation similarly well for Cityscapes, however when tested on the distributionally-shifted target domains, `Mask-d2`'s performance exceeds that of the baselines. The gap in $\text{MaxF}_{0.5}$ score between `Mask-d2` and `MaxS-d2`, `MaxS-d2-E*` is descriptive of the benefit of our proposed uncertainty training.

		$\text{MaxF}_{0.5}$ @ $p(a, c)$					
	Method	CS	LDN	NF	SCOT	BDD	WD
Baselines	Ensemble-5-d2	0.944 @ 0.721	0.939 @ 0.699	0.94 @ 0.697	0.894 @ 0.552	0.932 @ 0.695	0.912 @ 0.624
	Ensemble-10-d2	0.945 @ 0.727	0.94 @ 0.689	0.942 @ 0.701	0.891 @ 0.537	0.933 @ 0.7	0.913 @ 0.626
	MCD-5-d2	0.935 @ 0.726	0.919 @ 0.656	0.912 @ 0.64	0.803 @ 0.416	0.926 @ 0.679	0.903 @ 0.599
	MCD-10-d2	0.935 @ 0.726	0.92 @ 0.656	0.912 @ 0.639	0.805 @ 0.415	0.926 @ 0.679	0.903 @ 0.599
	MaxS-d2-E*	0.933 @ 0.714	0.899 @ 0.658	0.916 @ 0.641	0.857 @ 0.507	0.925 @ 0.678	0.908 @ 0.609
	MaxS-d2	0.944 @ 0.739	0.881 @ 0.674	0.883 @ 0.659	0.805 @ 0.458	0.894 @ 0.674	0.87 @ 0.603
	GMM-d2	0.904 @ 0.757	0.878 @ 0.692	0.881 @ 0.683	0.778 @ 0.415	0.864 @ 0.674	0.805 @ 0.567
	MaxS-d1-E*	0.912 @ 0.67	0.859 @ 0.579	0.899 @ 0.612	0.82 @ 0.359	0.886 @ 0.586	0.86 @ 0.483
	MaxS-d1	0.936 @ 0.716	0.858 @ 0.61	0.887 @ 0.628	0.79 @ 0.343	0.896 @ 0.622	0.853 @ 0.497
	GMM-d1	0.894 @ 0.751	0.789 @ 0.639	0.823 @ 0.61	0.687 @ 0.281	0.827 @ 0.656	0.743 @ 0.572
	C&R-NF-d2	0.928 @ 0.703	0.911 @ 0.641	0.925 @ 0.662	0.869 @ 0.518	0.912 @ 0.669	0.891 @ 0.61
	C&R-NF-d2-f_θ^+	0.931 @ 0.69	0.894 @ 0.658	0.902 @ 0.66	0.863 @ 0.524	0.923 @ 0.676	0.908 @ 0.618
	C&R-NF-d1	0.917 @ 0.666	0.867 @ 0.567	0.901 @ 0.607	0.825 @ 0.364	0.898 @ 0.618	0.874 @ 0.522
	C&R-NF-d1-f_θ^+	0.919 @ 0.67	0.894 @ 0.599	0.907 @ 0.612	0.752 @ 0.338	0.891 @ 0.622	0.852 @ 0.5
Ours	Mask-LDN-d2	0.945 @ 0.735	**0.948** @ 0.693	**0.948** @ 0.697	0.889 @ 0.481	0.934 @ 0.694	0.913 @ 0.605
	Mask-NF-d2	**0.947** @ 0.724	0.941 @ 0.679	**0.948** @ 0.696	0.903 @ 0.517	0.927 @ 0.677	0.902 @ 0.582
	Mask-SCOT-d2	0.934 @ 0.706	0.924 @ 0.645	0.939 @ 0.693	**0.905** @ 0.568	0.92 @ 0.662	0.899 @ 0.579
	Mask-BDD-d2	0.938 @ 0.725	0.931 @ 0.675	0.942 @ 0.698	0.891 @ 0.501	**0.936** @ 0.696	**0.918** @ 0.631
	Mask-LDN-d1	0.931 @ 0.693	0.9 @ 0.623	0.919 @ 0.641	0.848 @ 0.361	0.91 @ 0.63	0.879 @ 0.508
	Mask-SCOT-d1	0.926 @ 0.679	0.881 @ 0.599	0.919 @ 0.636	0.851 @ 0.397	0.907 @ 0.624	0.87 @ 0.502

TABLE I. Misclassification Detection performance described by $\text{MaxF}_{0.5}$ @ $p(a, c)$ for a range of test domains.

		AUPR					
	Method	CS	LDN	NF	SCOT	BDD	WD
Baselines	Ensemble-5-d2	0.98	0.976	0.979	0.933	0.967	0.948
	Ensemble-10-d2	0.981	0.976	0.978	0.932	0.967	0.946
	MCD-5-d2	0.977	0.971	0.963	0.778	0.971	0.959
	MCD-10-d2	0.977	0.971	0.963	0.786	0.971	0.959
	MaxS-d2-E*	0.975	0.958	0.968	0.931	0.97	0.961
	MaxS-d2	0.982	0.933	0.942	0.861	0.94	0.919
	GMM-d2	0.936	0.894	0.914	0.838	0.896	0.845
	MaxS-d1-E*	0.966	0.92	0.953	0.889	0.942	0.925
	MaxS-d1	0.98	0.913	0.942	0.859	0.949	0.918
	GMM-d1	0.921	0.781	0.866	0.741	0.835	0.739
	C&R-NF-d2	0.972	0.967	0.975	0.936	0.954	0.937
	C&R-NF-d2-f_θ^+	0.978	0.927	0.932	0.92	0.965	0.96
	C&R-NF-d1	0.972	0.929	0.954	0.883	0.951	0.935
	C&R-NF-d1-f_θ^+	0.972	0.949	0.957	0.794	0.941	0.915
Ours	Mask-LDN-d2	0.985	**0.987**	**0.985**	0.947	**0.973**	0.966
	Mask-NF-d2	**0.988**	0.984	**0.985**	0.958	0.964	0.958
	Mask-SCOT-d2	0.977	0.975	0.98	**0.96**	0.965	0.957
	Mask-BDD-d2	0.98	0.977	0.981	0.952	0.972	**0.967**
	Mask-LDN-d1	0.979	0.956	0.969	0.91	0.957	0.937
	Mask-SCOT-d1	0.977	0.936	0.966	0.92	0.959	0.933

TABLE II. Misclassification Detection performance summarised over all possible thresholds described by AUPR for a range of test domains.

C. MIM versus C&R

In our tests in the NF domain, `Mask` models are generally superior to the `C&R` models, while also exhibiting additional benefits. We trained two `Mask` and `C&R` variants with different augmentation hyperparameters. We use $p_{\text{mask}} = [0.25, 0.75]$, one variation of `C&R` with less colour-space augmentation, and the other cropping using different scale parameters. This experiment demonstrated that the $\text{MaxF}_{0.5}$ @ $p(a, c)$ = [0.926@0.655, 0.891@0.648] for the `C&R` in the order presented, and similarly for `Mask` models [0.945@0.705, 0.951@0.713] on the NF dataset. The scores for `C&R` vary considerably more than for `Mask`, thereby making masking more convenient.

D. Freezing f_θ

For the `C&R` task and training on NF, when DINOv1 is used and f_θ is not frozen, we see a boost in uncertainty estimation performance in NF over the equivalent with frozen f_θ. This however does not necessarily hold for other domains and DINOv2 – this shows that f_θ can be frozen when using sufficiently general representations, removing the possibility of feature collapse seen in [2].

VII. Conclusion

This work presents a method that leverages MIM with unlabelled images and general feature representations in order to train a network to jointly perform high-quality semantic segmentation and uncertainty estimation. We show that this method outperforms a number of baselines on a number of different test datasets, alongside investigations into the effect of initial representations, augmentations and training datasets.

REFERENCES

[1] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," *arXiv:2304.07193*, 2023.

[2] D. Williams, D. De Martini, M. Gadd, and P. Newman, "Mitigating distributional shift in semantic segmentation via uncertainty estimation from unlabelled data," in *IEEE Transactions on Robotics (T-RO)*, 2024.

[3] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*, 2015.

[4] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, 2011.

[5] C. Louizos and M. Welling, "Multiplicative normalizing flows for variational bayesian neural networks," in *International Conference on Machine Learning*. PMLR, 2017.

[6] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016.

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

[8] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, 2017.

[9] R. Weston, S. H. Cen, P. Newman, and I. Posner, "Probably unknown: Deep inverse sensor modelling radar," *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[10] J. Gast and S. Roth, "Lightweight probabilistic deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[11] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self-supervised learning of geometrically stable features through probabilistic introspection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal, "Deterministic neural networks with inductive biases capture epistemic and aleatoric uncertainty," *arXiv preprint arXiv:2102.11582*, 2021.

[13] J. R. van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, "Simple and scalable epistemic uncertainty estimation using a single deep deterministic neural network," in *ICML*, 2020.

[14] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan, "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness," in *Advances in Neural Information Processing Systems*, 2020.

[15] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[16] J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari, "Sampling-free epistemic uncertainty estimation using approximated variance propagation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[17] N. Tagasovska and D. Lopez-Paz, "Single-model uncertainties for deep learning," *Advances in Neural Information Processing Systems*, 2019.

[18] A. A. Alemi, I. Fischer, and J. V. Dillon, "Uncertainty in the variational information bottleneck," *arXiv preprint arXiv:1807.00906*, 2018.

[19] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *International Conference on Learning Representations*, 2017.

[20] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proceedings of International Conference on Learning Representations*, 2017.

[21] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018.

[22] H. Wang, Z. Li, L. Feng, and W. Zhang, "Vim: Out-of-distribution with virtual-logit matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[23] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," *Advances in neural information processing systems*, 2018.

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

[25] D. S. W. Williams, M. Gadd, D. D. Martini, and P. Newman, "Fool me once: Robust selective segmentation via out-of-distribution detection with contrastive learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[26] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *International Conference on Learning Representations*, 2018.

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

[28] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv:2111.06377*, 2021.

[29] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "ibot: Image bert pre-training with online tokenizer," *International Conference on Learning Representations (ICLR)*, 2022.

[30] G. Li, H. Zheng, D. Liu, C. Wang, B. Su, and C. Zheng, "Semmae: Semantic-guided masking for learning masked autoencoders," *Advances in Neural Information Processing Systems*, 2022.

[31] Y. Shi, N. Siddharth, P. Torr, and A. R. Kosiorek, "Adversarial masking for self-supervised learning," in *International Conference on Machine Learning*, 2022.

[32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020.

[33] M. Assran, M. Caron, I. Misra, P. Bojanowski, A. Joulin, N. Ballas, and M. Rabbat, "Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[34] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, 2015.

[36] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers &; distillation through attention," in *International Conference on Machine Learning*, 2021.

[38] 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.

[39] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G. F. Dominguez, "Wilddash - creating hazard-aware benchmarks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.