CALIBRATED ADVERSARIAL REFINEMENT FOR STOCHASTIC SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Ambiguities in images or unsystematic annotation can lead to multiple valid solutions in semantic segmentation. To learn a distribution over predictions, recent work has explored the use of probabilistic networks. However, these do not necessarily capture the empirical distribution accurately. In this work, we aim to learn a calibrated multimodal predictive distribution, where the empirical frequency of the sampled predictions closely reflects that of the corresponding labels in the training set. To this end, we propose a novel two-stage, cascaded strategy for calibrated adversarial refinement. In the first stage, we explicitly model the data with a categorical likelihood. In the second, we train an adversarial network to sample from it an arbitrary number of coherent predictions. The model can be used independently or integrated into any black-box segmentation framework to facilitate learning of calibrated stochastic mappings. We demonstrate the utility and versatility of the approach by attaining state-of-the-art results on the multigrader LIDC dataset and a modified Cityscapes dataset. In addition, we use a toy regression dataset to show that our framework is not confined to semantic segmentation, and the core design can be adapted to other tasks requiring learning a calibrated predictive distribution.

1 INTRODUCTION

Even though ambiguities are often unavoidable in many real-world datasets, the majority of the research encompassing semantic segmentation focuses on optimising models that assign a single prediction to each input image (Ronneberger et al., 2015; Jégou et al., 2017; Takikawa et al., 2019; Chen et al., 2017a;b; 2016a;b; 2015). These are often incapable of capturing the entire empirical distribution of outputs as they optimise for a one-fits-all solution. Consequently, conflicts between labels corresponding to a given input image can compromise the quality of the final predictions. To resolve such ambiguities, ideally, one would sample multiple consistent hypotheses, and use uncertainty information to identify potential errors in each one. Such a system would be particularly useful for safety-critical applications, such as in medical diagnostics, or for guiding user interactions for refinement in semi-automated settings.

Several approaches have been proposed to capture label multimodality in image-to-image translation tasks (Huang et al., 2018; Lee et al., 2018; Zhu et al., 2017a; Bao et al., 2017; Zhang, 2018), with only a few of them applied on stochastic semantic segmentation (Kohl et al., 2018; 2019; Baumgartner et al., 2019; Hu et al., 2019; Kamnitsas et al., 2017; Rupprecht et al., 2017). These methods have the capacity to learn a diverse set of labels for each input, however, they are typically limited to a fixed number of samples (Kamnitsas et al., 2017; Rupprecht et al., 2017), return uncalibrated predictions, or do not account for uncertainty. First, restricting to a fixed number of predictions can hinder a model from learning all of the modes of the ground truth distribution. Second, an uncalibrated predictive distribution is one that does not accurately reflect the occurrence frequencies of individual modes in the training set (Guo et al., 2017; Kull et al., 2019). Finally, uncertainty information is highly desirable, as it can be used for distinguishing between predictions that can be trusted, and those that should be reassessed (Kendall and Gal, 2017a; Ovadia et al., 2019).

In this work, we tackle all three challenges by introducing a two-stage cascaded strategy for adversarial refinement that facilitates the learning of a calibrated multimodal predictive distribution, and allows the extraction of reliable data uncertainty estimates. The key contributions are as follows [1]:

[1] Code is publicly available at <URL OMITTED FOR ANONYMITY>

- We propose a novel cascaded architecture that constructively combines explicit likelihood modelling with adversarial refinement to sample an arbitrary number of confident, and self-consistent predictions given an input image.
- We introduce a novel loss term that facilitates learning of calibrated stochastic mappings when using adversarial neural networks. To our knowledge this is the first work to do so.
- The proposed model can be trained independently or used to augment any pretrained black-box semantic segmentation model, endowing it with a multimodal predictive distribution.

2 Related work

Straightforward strategies towards learning multiple predictions include ensembling (Lakshminarayanan et al., 2017; Kamnitsas et al., 2017) or using multiple prediction heads (Rupprecht et al., 2017). Even though these approaches can capture a diverse set of sampled predictions, they are limited to only a fixed number of samples. Alternatively, a probability distribution over the outputs can be induced by activating dropout during test time (Gal and Ghahramani, 2016). This method does offer useful uncertainty estimates over the pixel-space, however, Isola et al. (2017) have demonstrated that it introduces only minor stochasticity in the output and returns inconsistent samples.

To address this, Kohl et al. (2018) take an orthogonal approach in combining a U-Net (Ronneberger et al., 2015) with a conditional variational autoencoder (cVAE) (Kingma and Welling, 2013) to learn a distribution over semantic labels. In Kohl et al. (2019) and Baumgartner et al. (2019) the authors build on Kohl et al. (2018) to improve the diversity of the samples by modelling the data on several scales of the image resolution. Nonetheless, these methods do not explicitly calibrate the predictive distribution in the pixel-space, and consequently do not provide reliable aleatoric uncertainty estimates (Kendall and Gal, 2017b; Choi et al., 2018; Gustafsson et al., 2019). Hu et al. (2019) address this shortcoming by using the intergrader variability as additional supervision. A major limitation of this approach is the requirement of a-priori knowledge of all the modalities of the data distribution. For many real-world datasets, however, this information is not readily available.

In the more general domain of image-to-image translation, alternative methods employ hybrid models that use adversarially trained cVAEs (Zhu et al., 2017a; Bao et al., 2017) to learn a distribution over a latent code, capturing multimodality, in order to sample diverse and coherent predictions. A common hurdle in conditional generative adversarial network (cGAN) approaches is that simply incorporating a noise vector as an additional input often results in mode collapse. This occurs due to the lack of regularisation between the noise input and generator output, allowing the generator to learn to ignore the noise vector (Isola et al., 2017). This issue is commonly resolved by using supplementary cycle-consistency losses (Huang et al., 2018; Lee et al., 2018; Zhu et al., 2017a; Bao et al., 2017), as proposed by Zhu et al. (2017b). However, none of these methods address the challenge of calibrating the predictive distribution.

3 Preliminaries

Given an input image $x \in \mathbb{R}^{H \times W \times C}$, semantic segmentation refers to the task of predicting a pixel-wise class label $y \in \{1, \dots, K\}^{H \times W}$. For a dataset of N image and label pairs, $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, the conditional distribution $p_{\mathcal{D}}(y \mid x)$ can be explicitly modelled through a likelihood $q_\theta(y \mid x)$, parameterised by a convolutional neural network F with weights θ, and activated by a softmax function (Ronneberger et al., 2015; Jégou et al., 2017). One simple, yet effective way to learn the class probabilities is to express $y \in \{0, 1\}^{H \times W \times K}$ as a one-hot encoded label, and set q_θ as a pixelwise factorised categorical distribution, given by:

$$q_\theta(y \mid x) = \prod_{i}^{H} \prod_{j}^{W} \prod_{k}^{K} F_\theta(x)_{i,j,k}^{y_{i,j,k}}. \tag{1}$$

The parameters θ are then optimised by minimising the cross entropy between $p_{\mathcal{D}}$ and q_θ, defined as:

$$\mathcal{L}_{\text{ce}}(\mathcal{D}, \theta) = -\mathbb{E}_{p_{\mathcal{D}}(x,y)}[\log q_\theta(y \mid x)]. \tag{2}$$

When trained with Eq. (2), F_θ learns an approximation of $\mathbb{E}_{p_{\mathcal{D}}}[y \mid x]$ (Bishop, 2006) that captures the pixelwise class probabilities over the label corresponding to a given input. This accommodates

the quantification of the amount of noise inherent to the data, also referred to as aleatoric uncertainty, which can be obtained by computing the entropy of the output of F_θ, $\mathbb{H}(F_\theta(x))$ (Kendall and Gal, 2017a). Further, the final segmentation map is typically obtained by applying the argmax function along the class dimension of the predicted probabilities. However, due to its deterministic nature, this approach is unable to produce multiple alternative predictions for the same input image. On the other hand, direct sampling from q_θ yields incoherent semantic maps, as for noisy datasets, maximising the factorised likelihood results in unconfident predictions in regions of inter-label inconsistencies, e. g. fuzzy object boundaries, exemplified later in the model overview diagram in Fig. 1.

The aforementioned limitations can be partially addressed by adapting the framework of generative adversarial networks (GANs) (Goodfellow et al., 2014) to the context of conditional semantic segmentation, as proposed by Luc et al. (2016). Formally, this involves training a binary discriminator network D to optimally distinguish between ground truth and predictions, while concurrently training a generative network G to maximise the probability that prediction samples $G(x)$ are perceived as real by D. Importantly, in contrast to explicit pixelwise likelihood maximisation, the adversarial setup learns an implicit sampler through G, capable of modelling the joint pixel configuration of the synthesised labels, and capturing both local and global consistencies present in the ground truth.

In practice, the generator loss is often complemented with the pixelwise loss from Eq. (2) to improve training stability and prediction quality (Luc et al., 2016; Ghafoorian et al., 2018; Samson et al., 2019). However, we argue that the two objective functions are not well aligned in the presence of noisy data. While the categorical cross entropy optimises for a single solution for each input x, thus encouraging high entropy in $q_\theta(y \mid x)$ within noisy regions of the data and calibrating the predictive distribution, the adversarial term allows multiple solutions while optimising for low entropy, label-like output. Therefore combining these losses in an additive manner, and enforcing them on the same set of parameters can be suboptimal. This issue can be mitigated to some extent by a scheduled downscaling of $\mathcal{L}_{\text{ce}}$, however, the residual conflict between the two losses adversely affects the optimisation process.

4 METHOD

4.1 CALIBRATED ADVERSARIAL REFINEMENT

In this work, we propose to avert potential conflict between the cross entropy and adversarial losses by decoupling them in a two-stage cascaded architecture consisting of a *calibration network* F_θ optimised with $\mathcal{L}_{\text{ce}}$ from Eq. (2), the output of which is fed to a *refinement network* G_ϕ. G_ϕ is then optimised with an adversarial loss term, parameterised by an auxiliary discriminator D_ψ trained with a binary cross entropy loss. To account for the multimodality in the labels, we additionally condition the refinement network on an extraneous noise variable $\epsilon \sim \mathcal{N}(0, 1)$. In practice, we also condition the refinement network and the discriminator on input x, however, we do not show this explicitly for notational convenience. More formally, using the non-saturated version of the adversarial loss (Goodfellow et al., 2014), the objectives for the refinement and discriminator networks respectively are given by:

$$\mathcal{L}_{\text{adv}}(\mathcal{D}, \theta, \phi) = -\mathbb{E}_{p_{\mathcal{D}}(x,y),p(\epsilon)}[\log D_\psi(G_\phi(F_\theta(x), \epsilon))], \tag{3}$$

$$\mathcal{L}_{\text{D}}(\mathcal{D}, \theta, \phi, \psi) = -\mathbb{E}_{p_{\mathcal{D}}(x,y),p(\epsilon)}[\log\left(1 - D_\psi(G_\phi(F_\theta(x), \epsilon))\right) + \log D_\psi(y)]. \tag{4}$$

To calibrate the predictive distribution, we impose diversity regularisation on G_ϕ by introducing a novel loss term that encourages the sample average $\overline{G}_\phi(F_\theta(x)) \coloneqq \mathbb{E}_{p(\epsilon)}[G_\phi(F_\theta(x), \epsilon)]$ to match the class probabilities predicted by $F_\theta(x)$. Here, $\overline{G}_\phi(F_\theta(x))$ serves as an approximation of the implicit predictive distribution of the refinement network. To this end, we define an auxiliary fully-factorised categorical likelihood q_ϕ as:

$$q_\phi(y \mid F_\theta(x)) = \prod_i^H \prod_j^W \prod_k^K \overline{G}_\phi(F_\theta(x))_{i,j,k}^{y_{i,j,k}}, \tag{5}$$

where ϕ is optimised by minimising the Kullback-Leibler divergence $\mathrm{KL}(q_\phi \mid\mid q_\theta)$.[2] Since both q_ϕ and q_θ are categorical distributions, the divergence can be computed exactly. We coin this loss term

[2]The choice of divergence is heuristically motivated and can be changed to fit different use-case requirements. We delegate theoretical and experimental analysis of other divergences to future work.

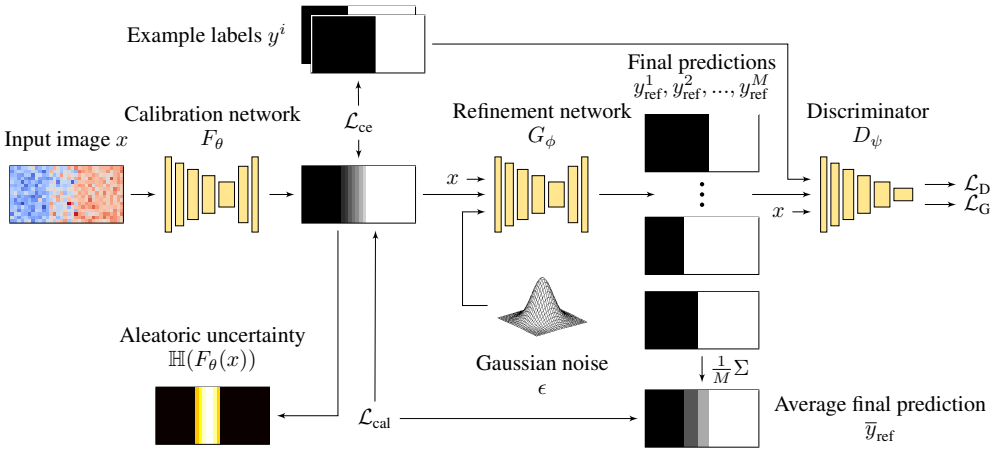


Figure 1: Model overview with an illustrative example where red and blue pixels are vertically segmented. A fuzzy boundary in the input image x allows for multiple valid ground truth labels y^i. First, the calibration network maps the input to a calibrated pixelwise distribution over the labels. This is then fed into the refinement network which samples an arbitrary number of diverse, crisp label proposals $y^1_{\text{ref}}, \ldots, y^M_{\text{ref}}$. To ensure calibration, the average of the final predictions is matched with the calibration target from the first stage through the $\mathcal{L}_{\text{cal}}$ loss. Additionally, the aleatoric uncertainty can be readily extracted from the calibration target, e. g. by computing the entropy $\mathbb{H}(F_\theta(x))$.

as the *calibration loss*, defined as:

$$\mathcal{L}_{\text{cal}}(\mathcal{D}, \theta, \phi) = \mathbb{E}_{p_{\mathcal{D}}}\left[\mathbb{E}_{q_\phi}[\log q_\phi(y \mid F_\theta(x)) - \log q_\theta(y \mid x)]\right]. \tag{6}$$

Since $\mathcal{L}_{\text{cal}}$ optimises through $\overline{G}_\phi(F_\theta(x))$ rather than a single sampled prediction, the model is not restricted to learning a single mode-averaging solution for each input x, and is therefore more compatible with $\mathcal{L}_{\text{adv}}$. The total loss for the refinement network then becomes:

$$\mathcal{L}_{\text{G}}(\mathcal{D}, \theta, \phi) = \mathcal{L}_{\text{adv}}(\mathcal{D}, \theta, \phi) + \lambda \mathcal{L}_{\text{cal}}(\mathcal{D}, \theta, \phi), \tag{7}$$

where $\lambda \geq 0$ is an adjustable hyperparameter. Fig. 1 shows the interplay of F_θ, G_ϕ and D_ψ and the corresponding loss terms.

Intuitively, the calibration network F_θ serves three main purposes. It accommodates the extraction of sample-free aleatoric uncertainty maps. It provides G_ϕ with an augmented representation of x enclosing probabilistic information about y. Finally, it sets a calibration target used by $\mathcal{L}_{\text{cal}}$ to regularise the predictive distribution of G_ϕ, in a cycle-consistent manner (Zhu et al., 2017b). The refinement network can then be interpreted as a stochastic sampler, modelling the inter-pixel dependencies to draw self-consistent samples from the explicit likelihood provided by the calibration network. Thus both the pixelwise class probability and object coherency are preserved. This approach leads to improved mode coverage and training stability, and increased convergence speed, as demonstrated in Section 5.

4.2 Practical Considerations

An important consequence of the loss decomposition is that the weights of F_θ can be kept fixed, while the adversarial pair G_ϕ and D_ψ are being trained. This allows F_θ to be pretrained in isolation, consequently lowering the overall peak computational burden and improving training stability. Further, computing $\mathcal{L}_{\text{cal}}$ requires a Monte Carlo estimation of $\overline{G}_\phi(F_\theta(x))$, where the quality of the loss feedback increases with the sample count. However, modern deep learning frameworks allow for the samples to be subsumed in the batch dimension, and can therefore be efficiently computed on GPUs. We also note that on noisy data points, the trained model does not require to be provided with all solutions for a given input. Instead, during training a random input-label pair is sampled, and the model automatically learns to identify cases where the input is ambiguous. Finally, our method can augment any existing black-box model B for semantic segmentation, furnishing it with a calibrated multimodal predictive distribution. This can be done by conditioning F_θ on the output of B, which we demonstrate in Section 5.2.2.

5 EXPERIMENTS

5.1 1D BIMODAL REGRESSION

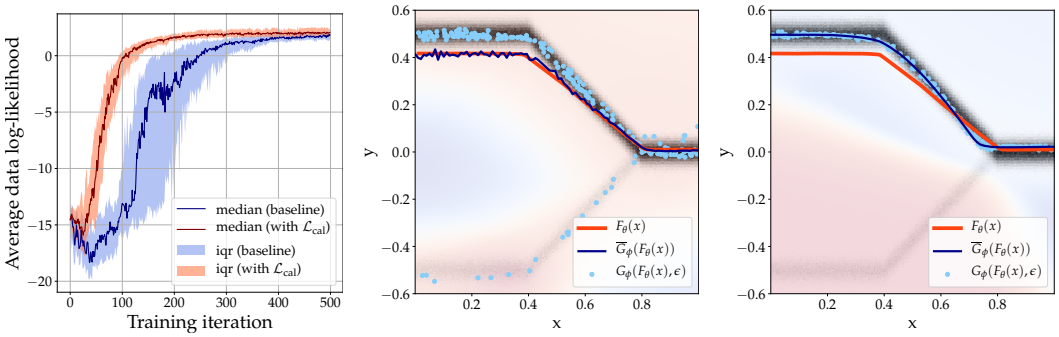
We give intuitive insight into the mechanics of the proposed calibration loss by designing and experimenting on a simple one-dimensional regression task. To create the dataset, an input $x \in [0, 1]$ is mapped to $y \in \mathbb{R}$ as follows:

$$y = \begin{cases} 0.5 - b + \epsilon, & x \in [0, 0.4) \\ (-1)^b(-1.25x + 1) + \epsilon, & x \in [0.4, 0.8) \\ \epsilon, & x \in [0.8, 1] \end{cases} \tag{8}$$

where $b \sim \text{Bernoulli}(\pi)$ and $\epsilon \sim \mathcal{N}(0, \sigma)$. We generate 9 different scenarios by varying the degree of mode selection probability $\pi \in \{0.5, 0.6, 0.9\}$ and the mode noise $\sigma \in \{0.01, 0.02, 0.03\}$.

For every data configuration, we use a 4-layer MLP for each of F_θ, G_ϕ and D_ψ, and train with and without calibration loss by setting the coefficient λ in Eq. (7) to 1 or 0, respectively. All runs are trained with a learning rate of $1\text{e}{-4}$, and each experiment is repeated five times. Note that unlike the categorical likelihood used in semantic segmentation tasks, we employ a Gaussian likelihood with fixed scale of 1. This changes the formulation of both Eqs. (2) and (6) to mean squared error losses between ground truth labels y and individual final predictions $\hat{y}$ for $\mathcal{L}_{\text{ce}}$, and between the output of the calibration network $F_\theta(x)$ and the average of multiple final predictions $\overline{G}_\phi(F_\theta(x))$ for $\mathcal{L}_{\text{cal}}$.

The results, depicted in Fig. 2, show that when using calibration loss, the optimisation process shows improved stability, converges faster and results in better calibrated predictions, in comparison to the non-regularised baseline. The effect is more pronounced in data configurations with higher bias. Further plots of the individual experiments are presented in Appendix B.1.



(a) (b) (c)

Figure 2: **(a)** Median and interquartile range (iqr) over the data log-likelihood, averaged over all 9×5×2 experiments. **(b)** High bias and noise configuration ($\pi = 0.9$, $\sigma = 0.03$) with calibration loss. The ground truth target is shown as black dots and the predicted samples as light blue dots. The predictions average in dark blue matches the calibration target in red. The discriminator output is shown in the background in shades of red (real) and blue (fake). **(c)** The same experiment configuration but without the proposed calibration loss, resulting in a mode collapse.

5.2 STOCHASTIC SEMANTIC SEGMENTATION

In this section we examine the capacity of our model to learn shape and class multimodality in real-world segmentation datasets. We begin by sharing essential implementation details below.

Network architectures For the calibration network F_θ, we use the encoder-decoder architecture from SegNet (Badrinarayanan et al., 2017). For G_ϕ, we designed a U-Net-style (Ronneberger et al., 2015) architecture with 4 down- and upsampling blocks, each consisting of a convolutional layer, followed by a batch normalisation layer (Ioffe and Szegedy, 2015), a leaky ReLU activation, and a dropout layer (Srivastava et al., 2014) with 0.1 dropout probability. We use a base number of 32

channels, doubled or halved at every down- and upsampling transition. To propagate the sampled noise vector to the output, we inject it into every upsampling block of the network in an affine manner. To do so, we project the noise vector using two fully connected layers into scale and residual matrices, with the same number of channels as the feature maps at the points of injection, and use these matrices to adjust the channel-wise mean and variance of the activations. This is similar to the mechanism used for adaptive instance normalisation (Huang and Belongie, 2017). We base the architecture for D_ψ on that used in DC-GAN (Radford et al., 2015) except that we remove batch normalisation. Any deviations from this setup are described in the corresponding sections.

Training details We utilise the Adam optimiser (Kingma and Ba, 2014) with an initial learning rate of $2\mathrm{e}{-4}$ for F_θ and G_ϕ, and $1\mathrm{e}{-5}$ for D_ψ. The learning rates are linearly decayed over time and we perform scheduled updates to train the networks. Additionally, the discriminator loss is regularised by using the R_1 zero-centered gradient penalty term (Mescheder et al., 2018). For a detailed list of hyperparameter values, see Appendix A.1.

Metrics Following Kohl et al. (2018; 2019), Huang et al. (2018) and Baumgartner et al. (2019), we use the Generalised Energy Distance (GED) (Székely and Rizzo, 2013) metric:

$$D^2_{\mathrm{GED}}(p_{\mathcal{D}}, q_\phi) = 2\mathbb{E}_{s\sim q_\phi, y\sim p_{\mathcal{D}}}[d(s,y)] - \mathbb{E}_{s,s'\sim q_\phi}[d(s,s')] - \mathbb{E}_{y,y'\sim p_{\mathcal{D}}}[d(y,y')], \tag{9}$$

where $d(s,y) = 1 - \mathrm{IoU}(s,y)$. As an additional metric, we follow Kohl et al. (2019) in using the Hungarian-matched IoU (HM-IoU) (Kuhn, 2005). All experiments are performed in triplicate and we report results as mean and standard deviation. Further details regarding the exact implementation of the GED and HM-IoU metrics for each experiment can be found in Appendix A.2.

5.2.1 Learning shape diversity on the LIDC dataset

The Lung Image Database Consortium (LIDC) (Armato III et al., 2011) dataset consists of 1018 thoracic CT scans from 1010 lung cancer patients, graded independently by four expert annotators. We use the 180×180 crops from the preprocessed version of the LIDC dataset used and described in Kohl et al. (2018). The dataset is split in 8882, 1996 and 1992 images in the training, validation and test sets respectively. All models are trained on lesion-centered 128×128 crops where at least one of the four annotations indicates a lesion. The final evaluation is performed on the provided test set.

To assess the accuracy and diversity of samples generated by our model we examine how well the learnt conditional distribution captures the shape diversity of lesion segmentations in the dataset. We pretrain $F_\theta(x)$ with $\mathcal{L}_{\mathrm{ce}}$ in isolation, fix its weights, and then train G_ϕ with $\mathcal{L}_{\mathrm{G}}$, where we estimate $\mathcal{L}_{\mathrm{cal}}$ with 20 samples from G_ϕ. Further experiments with varying sample size are disclosed in Table 3 in Appendix B.2. As a control experiment, we train using the same architecture but replace $\mathcal{L}_{\mathrm{cal}}$ in the refinement network loss function with a cross entropy loss $\mathcal{L}_{\mathrm{ce}}$, as used in Luc et al. (2016).

Table 1: Mean GED and HM-IoU scores on LIDC. Top section: approaches using the original data splits defined by Kohl et al. (2018), which we also adhere to; middle: approaches using random data splits; bottom: the $\mathcal{L}_{\mathrm{ce}}$-regularised baseline and our $\mathcal{L}_{\mathrm{cal}}$-regularised cGAN. The three central columns show the GED score computed with 16, 50 and 100 samples, respectively. The rightmost column shows the HM-IoU score, computed with 16 samples. The arrows ↑ and ↓ indicate if higher or lower score is better.

Method	GED ↓ (16)	GED ↓ (50)	GED ↓ (100)	HM-IoU ↑ (16)
Kohl et al. (2018)	0.320 ± 0.030	—	$0.252 \pm$ N/A[1]	0.500 ± 0.030
Kohl et al. (2019)	0.270 ± 0.010	—	—	0.530 ± 0.010
Hu et al. (2019)	—	0.267 ± 0.012	—	—
Baumgartner et al. (2019)	—	—	$\mathbf{0.224}\pm$**N/A**[2]	—
cGAN+$\mathcal{L}_{\mathrm{ce}}$	0.639 ± 0.002	—	—	0.477 ± 0.004
cGAN+$\mathcal{L}_{\mathrm{cal}}$	$\mathbf{0.264 \pm 0.002}$	$\mathbf{0.248 \pm 0.004}$	0.243 ± 0.004[2]	$\mathbf{0.592 \pm 0.005}$

[1] This score is taken from Baumgartner et al. (2019).
[2] Note that when following the data split methodology used in Baumgartner et al. (2019) and computing the GED (100) metric, we achieve a score of 0.217 instead of 0.243 (see Appendix A.2 for more details).

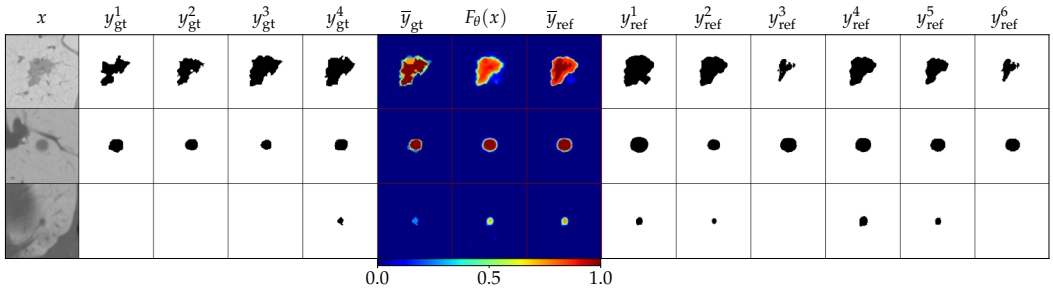


Figure 3: LIDC validation samples. From left to right: an input image x, followed by the four ground truth annotations $y^1_{\text{gt}} \dots y^4_{\text{gt}}$, the mean of the labels $\overline{y}_{\text{gt}}$, the output of the calibration network $F_\theta(x)$, the mean of the six refinement network samples $\overline{y}_{\text{ref}}$, shown in columns $y^1_{\text{ref}} \dots y^6_{\text{ref}}$.

The $\mathcal{L}_{\text{cal}}$-regularised model performs on par with other state-of-the-art methods w. r. t. the GED score, and outperforms them on the HM-IoU score (only available for Kohl et al. (2018; 2019)). Numerical results are summarised in Table 1. The diversity and fidelity of sampled predictions are illustrated in Fig. 3. In contrast, the $\mathcal{L}_{\text{ce}}$-regularised baseline collapses the predictive distribution, showing no perceptible diversity in the samples (see Fig. 9b in Appendix B.2.3), which results in a stark increase in the mean GED score, and decrease in the HM-IoU score, as shown in the bottom section of Table 1.

5.2.2 Learning a calibrated distribution on a multimodal Cityscapes dataset

The Cityscapes dataset contains 1024×2048 RGB images of urban scenes, and their corresponding segmentation maps. It consists of 2975 training, 500 validation and 1525 test images. Following Kohl et al. (2018), we use the version of the Cityscapes dataset with 19 semantic classes and downsampled images and segmentation maps at a spatial resolution of 256×512. They establish controlled multi-modality by augmenting the dataset with 5 new classes: *sidewalk2*, *person2*, *car2*, *vegetation2* and *road2*, introduced by flipping their original counterparts with probabilities $^8/_{17}$, $^7/_{17}$, $^6/_{17}$, $^5/_{17}$ and $^4/_{17}$, respectively (see Fig. 4a). Following Kohl et al. (2018), we report results on the validation set.

Since we have full knowledge over the probability of each mode in the ground truth distribution, we use this dataset to assess the calibration of the tested models. Further, to demonstrate that our approach can be easily integrated on top of any existing black-box segmentation model B, we employ the network from TensorFlow DeepLab Model Zoo (2020), finetuned on the official Cityscapes dataset, which achieves a mIoU of 0.79 on the test set. We utilise its predictions as input to our calibration network, F_θ, which consists of 5 convolutional blocks, each composed of a 3×3 convolutional layer, followed by a batch normalisation layer, a leaky ReLU activation, and a dropout layer with 0.1 dropout rate. We pretrain the calibration network in isolation, and subsequently apply it in inference

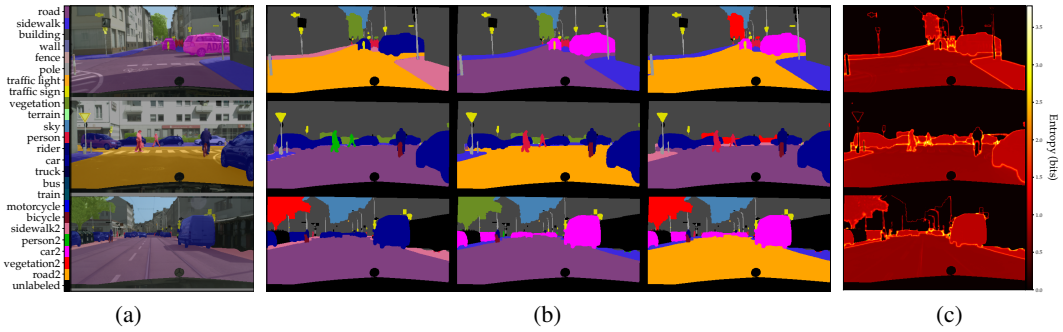


Figure 4: **(a)** Input images overlaid with the corresponding labels. **(b)** Samples obtained from the refinement network. **(c)** Aleatoric uncertainty computed as the entropy of the calibration output.

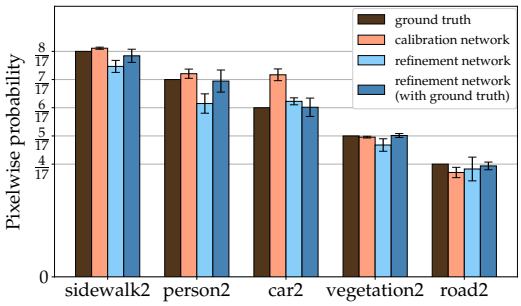


Figure 5: Calibration of the pixelwise probabilities of the five stochastic classes. Note that the calibration network (in orange) is conditioned on black-box predictions.

Table 2: Mean GED scores on the modified Cityscapes. Top section: competing model; middle: $\mathcal{L}_{\text{cal}}$-regularised cGAN and $\mathcal{L}_{\text{ce}}$-regularised baseline; bottom: ground truth calibrated cGAN. The GED scores are computed using 16 samples.

Method	GED
Kohl et al. (2018)	$0.202 \pm$ N/A
cGAN+$\mathcal{L}_{\text{cal}}$	$\mathbf{0.164 \pm 0.01}$
cGAN+$\mathcal{L}_{\text{ce}}$	0.632 ± 0.07
cGAN+$\mathcal{L}_{\text{cal}}$ (gt)	0.038 ± 0.00

mode while adversarially training the refinement network. We use a batch size of 16, and train with $\mathcal{L}_{\text{G}}$, estimating $\mathcal{L}_{\text{cal}}$ with 7 samples from G_ϕ. The same baseline as in the LIDC experiment is employed, where we replace $\mathcal{L}_{\text{cal}}$ with $\mathcal{L}_{\text{ce}}$. As a second control experiment we completely omit the calibration network and instead condition the refinement network on the known ground truth pixelwise categorical distribution over the label. This allows us to directly evaluate the quality of sampling administered from the refinement network.

When training the refinement network with a cross entropy loss instead of the calibration loss $\mathcal{L}_{\text{cal}}$, the predictive distribution collapses, making the output deterministic. Conversely, when we train our refinement network with $\mathcal{L}_{\text{cal}}$, the learnt predictive distribution is well adjusted, with high diversity and reconstruction quality, significantly outperforming the current state-of-the-art. Fig. 4b shows representative sampled predictions from our model for three input images, and Fig. 4c illustrates the corresponding aleatoric uncertainty maps extracted from $F_\theta(x)$. The learnt multimodality and noise in the dataset is reflected by regions of high uncertainty, where objects belonging to the different stochastic classes consistently display distinct shades of red, corresponding to their respective flip probabilities. Finally, we show that when using the ground truth pixelwise distribution as the input to the refinement network, we attain an almost perfect GED score. Thus, we demonstrate that G_ϕ learns calibrated refinement of the predictions from F_θ, where the quality of the final predictive distribution depends on the quality of $F_\theta(x)$. The GED scores are reported in Table 2 and the calibration on the stochastic classes, evaluated over the entire dataset, is shown in Fig. 5.

An important outstanding issue in our approach is that the calibration network does not perfectly capture the class-probabilities, e. g. for the *car2* category in Fig. 5. The limitations of modern neural networks w. r. t. calibration are well documented (Guo et al., 2017; Kull et al., 2019; Zhang et al., 2020). This has been attributed to several factors, such as long-tailed data distributions, out-of-distribution inputs, specific network architectural elements or optimising procedures. Therefore, an important direction for future work is improving the overall calibration of deep neural networks.

6 Conclusion

In this work, we developed a novel framework for semantic segmentation capable of learning a calibrated multimodal predictive distribution, closely matching the ground truth distribution of labels. We attained improved results on a modified Cityscapes dataset and competitive scores on the LIDC dataset, indicating the utility of our approach on real-world datasets. We also showed that our approach can be easily integrated into an off-the-shelf, deterministic, black-box semantic segmentation model, enabling sampling an arbitrary number of plausible segmentation maps. By providing multiple valid label proposals and highlighting regions of high data uncertainty, our approach can be used to identify and resolve ambiguities, diminishing risk in safety-critical systems. Therefore, we expect our approach to be particularly beneficial for applications such as map making for autonomous driving or computer-assisted medical diagnostics. Finally, even though the primary focus of this work is semantic segmentation, we demonstrated its versatility through an illustrative toy regression problem, alluding to a broader applicability beyond semantic image segmentation.

REFERENCES

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19, 2017.

Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5229–5238, 2019.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017a.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017b.

Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016a.

Liang-Chieh Chen, Jonathan T Barron, George Papandreou, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *CVPR*, 2016b.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in neural information processing systems*, pages 465–476, 2017a.

Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2745–2754, 2017.

Yongqi Zhang. Xogan: One-to-many unsupervised image-to-image translation. *arXiv preprint arXiv:1805.07277*, 2018.

Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *Advances in Neural Information Processing Systems*, pages 6965–6975, 2018.

Simon AA Kohl, Bernardino Romera-Paredes, Klaus H Maier-Hein, Danilo Jimenez Rezende, SM Eslami, Pushmeet Kohli, Andrew Zisserman, and Olaf Ronneberger. A hierarchical probabilistic u-net for modeling multi-scale ambiguities. *arXiv preprint arXiv:1905.13077*, 2019.

Christian F Baumgartner, Kerem C Tezcan, Krishna Chaitanya, Andreas M Hötker, Urs J Muehlematter, Khoschy Schawkat, Anton S Becker, Olivio Donati, and Ender Konukoglu. Phiseg: Capturing uncertainty in medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 119–127. Springer, 2019.

Shi Hu, Daniel Worrall, Stefan Knegt, Bas Veeling, Henkjan Huisman, and Max Welling. Supervised uncertainty quantification for segmentation with multiple annotations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 137–145. Springer, 2019.

Konstantinos Kamnitsas, Wenjia Bai, Enzo Ferrante, Steven McDonagh, Matthew Sinclair, Nick Pawlowski, Martin Rajchl, Matthew Lee, Bernhard Kainz, Daniel Rueckert, et al. Ensembles of multiple models and architectures for robust brain tumour segmentation. In *International MICCAI Brainlesion Workshop*, pages 450–462. Springer, 2017.

Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3591–3600, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

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 *Advances in Neural Information Processing Systems*, pages 12295–12305, 2019.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017a.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017b.

Sungjoon Choi, Kyungjae Lee, Sungbin Lim, and Songhwai Oh. Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6915–6922. IEEE, 2018.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schön. Evaluating scalable bayesian deep learning methods for robust computer vision. *arXiv preprint arXiv:1906.01620*, 2019.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017b.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

Laurens Samson, Nanne van Noord, Olaf Booij, Michael Hofmann, Efstratios Gavves, and Mohsen Ghafoorian. I bet you are wrong: Gambling adversarial networks for structured semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12): 2481–2495, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? *arXiv preprint arXiv:1801.04406*, 2018.

Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143(8):1249–1272, 2013.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52(1): 7–21, 2005.

Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

TensorFlow DeepLab Model Zoo. Pretrained model: xception71_dpc_cityscapes_trainfine, 2020. URL download.tensorflow.org/models/deeplab_cityscapes_xception71_trainfine_2018_09_08.tar.gz.

Jize Zhang, Bhavya Kailkhura, and T Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. *arXiv preprint arXiv:2003.07329*, 2020.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Iclr*, 2(5):6, 2017.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

Danijar Hafner, Dustin Tran, Timothy Lillicrap, Alex Irpan, and James Davidson. Noise contrastive priors for functional uncertainty. *arXiv preprint arXiv:1807.09289*, 2018.

Christos Louizos, Xiahan Shi, Klamer Schutte, and Max Welling. The functional neural process. *arXiv preprint arXiv:1906.08324*, 2019.

Appendices

A IMPLEMENTATION DETAILS

In this section we describe the overall training procedure and delve into the training and evaluation details for the stochastic segmentation experiments on the LIDC dataset and the modified Cityscapes dataset.

A.1 TRAINING PROCEDURE

Algorithm 1 outlines the practical procedure used to pretrain the calibration network, and the subsequent training of the refinement network. Even though the two networks can be trained end-to-end at once, in our experiments we use the two-step training procedure to stabilise the training and reduce the memory consumption on the GPU. This way we are able to fit larger batches and/or more samples for the estimate of $\mathcal{L}_{\text{cal}}$. Algorithm 2 shows the inference procedure for obtaining M output samples.

Algorithm 1 Model training with calibration network pretraining

require: training data $\mathcal{D}$, number of samples M, learning rate η, calibration loss scale λ;
procedure TRAINING($\mathcal{D}, M, \eta, \lambda$)
  while not converged **do** ▷ Pretraining of F_θ
    Sample batch $\{x, y\}_t \in \mathcal{D}$
    Update θ_{t+1} with $-\eta \nabla_{\theta_t} \mathcal{L}_{\text{ce}}(\{x, y\}_t, \theta_t)$
  end while
  while not converged **do** ▷ Adversarial training of G_ϕ and D_ψ
    Sample batch $\{x, y\}_t \in \mathcal{D}$
    for $i = 1, 2, \ldots, M$ **do**
      Sample $y_{\text{ref}}^{i,t} = G_{\phi_t}(F_{\theta^*}(x_t), \epsilon_i)$ where $\epsilon_i \sim \mathcal{N}(0, 1)$
    end for
    Compute $\overline{G}_\phi(F_\theta(x))^t = \frac{1}{M} \sum_{i=1}^{M} y_{\text{ref}}^{i,t}$
    Compute $\mathcal{L}_{\text{cal}}(x_t, \theta^*, \phi_t) = \sum_{i,j,k} \left(\overline{G}_\phi(F_\theta(x))^t \left(\log \overline{G}_\phi(F_\theta(x))^t - \log F_{\theta^*}(x_t)\right)\right)_{i,j,k}$
    Update ϕ_{t+1} with $-\eta \nabla_{\phi_t} \left(\mathcal{L}_{\text{G}}(\theta^*, \phi_t, \{x, y\}_t) + \lambda \mathcal{L}_{\text{cal}}(x_t, \theta^*, \phi_t)\right)$
    Update ψ_{t+1} with $-\eta \nabla_{\psi_t} \mathcal{L}_{\text{D}}(\theta^*, \phi_t, \psi_t, \{x, y\}_t)$
  end while
end procedure

Algorithm 2 Inference procedure

require: test data point x, number of samples M;
procedure INFERENCE(x, M) ▷ Using θ^* and ϕ^* from Algorithm 1
  for $i = 1, 2, \ldots, M$ **do**
    Sample $y_{\text{ref}}^{i} = G_{\phi^*}(F_{\theta^*}(x), \epsilon_i)$ where $\epsilon_i \sim \mathcal{N}(0, 1)$
  end for
end procedure

Notice that any off-the-shelf optimisation algorithm can be used to update the parameters θ, ϕ and ψ. For the segmentation experiments, we utilise the Adam optimiser (Kingma and Ba, 2014) with $\beta_1 = 0.5$, $\beta_2 = 0.99$ and weight decay of $5\text{e}{-4}$. F_θ is trained with a learning rate of $2\text{e}{-4}$ which is then lowered to $1\text{e}{-4}$ after 30 epochs. G_ϕ and D_ψ are updated according to a schedule, where G_ϕ is updated at every iteration, and D_ψ is trained in cycles of 50 iterations of weight updating, followed by 200 iterations with fixed weights. The refinement network is trained with an initial learning rate of $2\text{e}{-4}$, lowered to $1\text{e}{-4}$ after 30 epochs, whereas the discriminator has an initial learning rate of $1\text{e}{-5}$, lowered to $5\text{e}{-6}$ after 30 epochs. Additionally, we utilise the R_1 zero-centered gradient penalty term (Mescheder et al., 2018), to regularise the discriminator gradient on real data with a weight of 10. Other hyperparameter specifics such as the batch-size and whether we inject stochasticity via random noise samples or latent code samples, depend on the experiment and are disclosed in the respective sections below or in the main text.

A.2 LIDC

Architectures For the calibration network, F_θ, we use the encoder-decoder architecture from SegNet (Badrinarayanan et al., 2017), with a softmax activation on the output layer.

Training During training, we draw random 180×180 image-annotation pairs, and we apply random horizontal flips and crop the data to produce 128×128 lesion-centered image tiles. All of our models were implemented in PyTorch and trained for 80k iterations on a single 32GB Tesla V100 GPU.

We train all our models for the LIDC experiments using 8-dimensional noise vectors in the cGAN experiments, or latent codes in the cVAE-GAN experiments. This value was empirically found to perform well, sufficiently capturing the shape diversity in the dataset. Additionally, in the refinement networks loss, we set the weighting parameter λ in the total generator loss, defined in Eq. (7) in the main text, so as to establish a ratio of $\mathcal{L}_{\mathrm{G}} : \mathcal{L}_{\mathrm{cal}} = 1 : 0.5$, where $\mathcal{L}_{\mathrm{G}}$ is the adversarial component of the loss, and $\mathcal{L}_{\mathrm{cal}}$ is the calibration loss component. In practice, the actual weights used are 10 for $\mathcal{L}_{\mathrm{G}}$, and 5 for $\mathcal{L}_{\mathrm{cal}}$.

Evaluation Following Kohl et al. (2018), Kohl et al. (2019), Huang et al. (2018), and Baumgartner et al. (2019) we use the Generalised Energy Distance (GED) (Székely and Rizzo, 2013) metric, given as:

$$D^2_{\mathrm{GED}}(p_{\mathcal{D}}, q_\phi) = 2\mathbb{E}_{s\sim q_\phi, y\sim p_{\mathcal{D}}}\left[d(s,y)\right] - \mathbb{E}_{s,s'\sim q_\phi}\left[d(s,s')\right] - \mathbb{E}_{y,y'\sim p_{\mathcal{D}}}\left[d(y,y')\right], \tag{10}$$

where $d(s,y) = 1 - \mathrm{IoU}(s,y)$. Intuitively, the first term of Eq. (10) quantifies the disparity between sampled predictions and the ground truth labels, the second term—the diversity between the predictions, and the third term—the diversity between the ground truth labels. It is important to note that the GED is a sample-based metric, and therefore the quality of the score scales with the number of samples. We approximate the expectations with all 4 ground truth labels ($y \sim p_{\mathcal{D}}$) and 16, 50 or 100 samples from the model ($s \sim q_\phi$) for each input image x.

As Kohl et al. (2019) have pointed out, even though the GED metric is a good indicator for how well the learnt predictive distribution fits a multimodal ground truth distribution, it can reward high diversity in sampled predictions even if the individual samples do not show high fidelity to the ground truth distribution. As an alternative metric that is less sensitive to such degenerate cases, Kohl et al. (2019) propose to use the Hungarian-matched IoU (HM-IoU), which finds the optimal 1:1 IoU matching between ground truth samples and sampled predictions. Following Kohl et al. (2019), we duplicate the set of ground truth labels so that the number of both ground truth and predicted samples are 16, and we report the HM-IoU as the average of the best matched pairs for each input image.

In the main text we show the evaluated performance with both GED and HM-IoU metrics over the entire test set and compute the IoU on only the foreground of the sampled labels and predictions. In the case where both the matched up label and prediction do not show a lesion, the IoU is set to 1, so that a correct prediction of the absence of a lesion is rewarded.

Note that the methods of Baumgartner et al. (2019) and Hu et al. (2019), whose GED scores we report in Table 1, use test sets that differ from the original splits defined in Kohl et al. (2018), which are used in Kohl et al. (2018; 2019) and our work. Baumgartner et al. (2019) use a random 60:20:20 split for the training, testing and validation sets, and 100 samples to compute the GED score, whereas Hu et al. (2019) use a random 70:15:15 split and 50 samples to compute the GED score. Due to the lack of reproducibility, we do not consider this the conventional way of benchmarking. Therefore, in Table 1 we only report the scores for our models evaluated on the original splits. Nevertheless, we also trained and tested our cGAN+$\mathcal{L}_{\mathrm{cal}}$ model on the split methodology of Baumgartner et al. (2019) to enable a fairer comparison. This improved our GED score from 0.243 ± 0.004 to 0.217, while the HM-IoU, evaluated using 16 samples, remained similar at 0.598 (in the original splits we achieved an HM-IoU score of 0.592 ± 0.005). This shows that different random splits can significantly affect the final performance w. r. t. GED score, while HM-IoU appears to be a more robust metric.

A.3 CITYSCAPES

Architectures For the calibration network F_θ, we design a small neural network with 5 convolutional blocks, each comprised of a 3×3 convolutional layer, followed by a batchnorm layer and a leaky ReLU activation. The network is activated with a softmax function.

Training During training, we apply random horizontal flips, scaling and crops of size 128×128 on the image-label pairs. All of our models were implemented in PyTorch and trained for 120k training iterations on a single 16GB Tesla V100 GPU.

We train all our models for the modified Cityscapes experiments using 32-dimensional noise vectors. Similarly to the LIDC experiments, this value was empirically found to perform well, however, it can be further tuned. As

commonly practiced, we use the ignore-masks provided by the Cityscapes dataset to filter out the cross entropy, calibration and adversarial losses during training on the unlabelled pixels. Similarly to our LIDC experiment, we use a weight of 10 for $\mathcal{L}_{\mathrm{G}}$, and 5 for $\mathcal{L}_{\mathrm{cal}}$ in the refinement networks loss.

Evaluation The GED metric for Cityscapes is implemented as described in the appendix of Kohl et al. (2018) and evaluated across the entire validation set. In this dataset we have full knowledge of the ground truth class distribution and therefore we compute the GED metric by using the probabilities of each mode directly, as follows:

$$D^2_{\mathrm{GED}}(p_{\mathcal{D}}, q_\phi) = 2\mathbb{E}_{s\sim q_\phi, y\sim p_{\mathcal{D}}}[d(s,y)w(y)] - \mathbb{E}_{s,s'\sim q_\phi}\big[d(s,s')\big] - \mathbb{E}_{y,y'\sim p_{\mathcal{D}}}\big[d(y,y')w(y)w(y')\big], \quad (11)$$

where $w(\cdot)$ is a function mapping the mode of a given label y to its corresponding probability mass. The distance $d(s,y)$ is computed using the average IoU of the 10 switchable classes only, as done in Kohl et al. (2018). In the cases where none of the switchable classes are present in both the ground truth label and the prediction paired up in $d(s,y)$, the distance score is not considered in the expectation. We use 16 samples to compute the GED score.

For the calibration results presented in Fig. 5, Section 5.2.2 in the main text, we compute the calibration network class-probabilities using the raw predictions of $F_\theta(x)$. We obtain class masks by computing the overlap between the ground truth labels and the black-box predictions for each class. Using these masks we then compute the average class-wise probabilities. The probabilities for the refinement network G_ϕ were computed as the average over 16 samples. Here the class masks are obtained by finding the pixels that are specified as the class of interest in the ground truth labels.

B ADDITIONAL EXPERIMENT RESULTS

To reinforce the results reported in Section 5 we present supplementary results for the bimodal regression experiment and the LIDC and Cityscapes segmentation experiments.

B.1 1D BIMODAL REGRESSION

Fig. 6 shows the data log-likelihoods for the 9 data configurations for varying mode bias $\pi \in \{0.5, 0.6, 0.9\}$ and mode noise $\sigma \in \{0.01, 0.02, 0.03\}$ trained with and without the calibration loss $\mathcal{L}_{\text{cal}}$. Each experiment is repeated 5 times and the individual likelihood curves are plotted in Fig. 6b and Fig. 6d respectively. The results show that high bias is harder to learn, reflected by a slowed down convergence, however, the $\mathcal{L}_{\text{cal}}$-regularised model shows greater robustness to weight initialisation. In contrast the non-regularised GAN exhibits mode oscillation expressed as a fluctuation of higher likelihood (one mode is covered) and lower one (between modes).

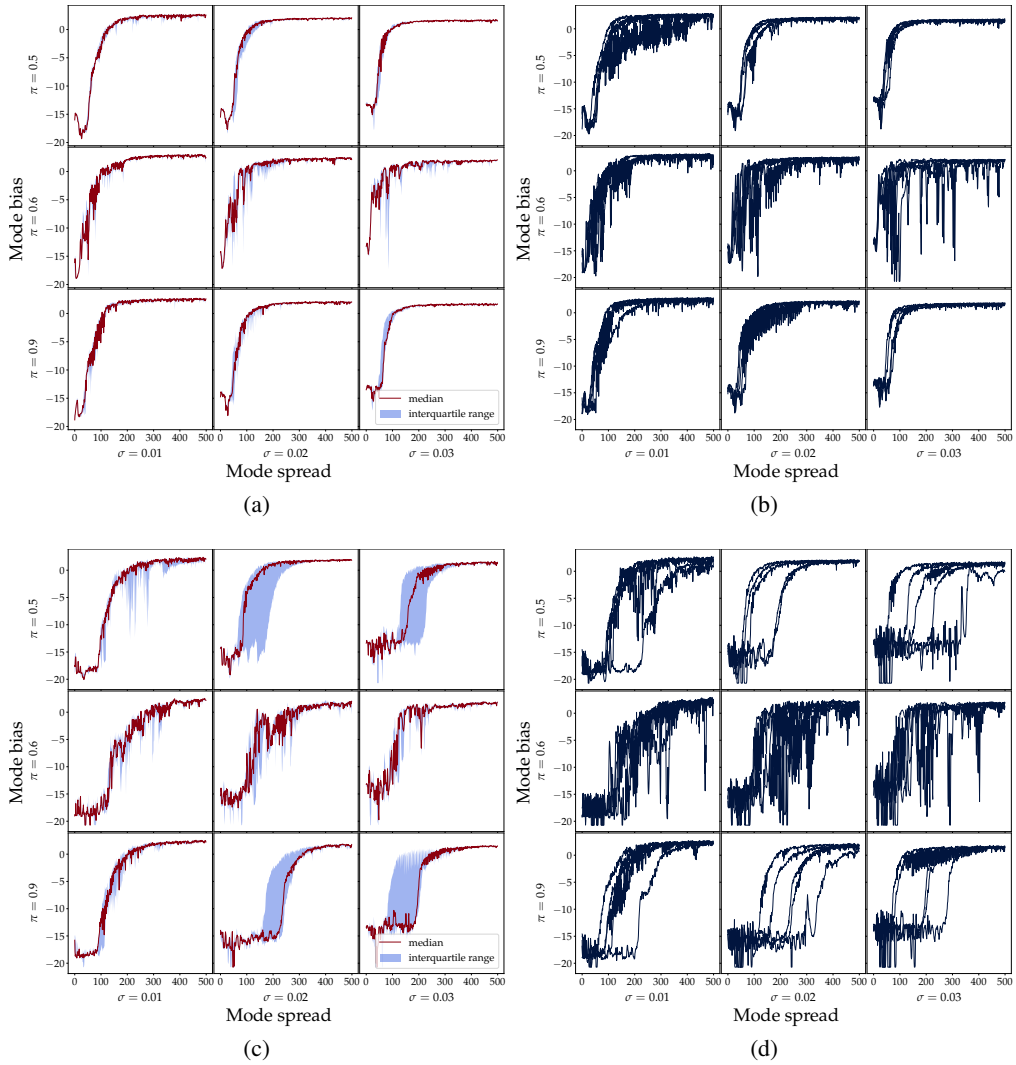


Figure 6: Log-likelihood curves for 5 runs on each of the 9 data configurations. **(a)** No calibration loss ($\lambda = 0$), averaged. **(b)** No calibration loss, individual runs. **(c)** With calibration loss ($\lambda = 1$), averaged. **(d)** With calibration loss, individual runs.

B.2 LIDC

B.2.1 QUALITATIVE ANALYSIS

To further examine our $\mathcal{L}_{\text{cal}}$-regularised cGAN model trained on the LIDC dataset, we illustrate representative qualitative results in Fig. 7 and Fig. 8. For every input image x, we show the ground truth labels $y^1_{\text{gt}}, \ldots, y^4_{\text{gt}}$ provided by the different expert annotators, overlaying the input image, in the first four columns, and 6 randomly sampled predictions $y^1_{\text{ref}}, \ldots, y^6_{\text{ref}}$ in the last six columns. From left to right, the three columns with the dark blue background in the center of the figures show the average ground truth predictions $\bar{y}_{\text{gt}}$, the output of the calibration network $F_\theta(x)$ and the average of 16 sampled predictions from the refinement network $\bar{y}_{\text{ref}}$. Our results show that even though there is a significant variability between the refinement network samples for a given input image, $\bar{y}_{\text{ref}}$ is almost identical to the calibration target $F_\theta(x)$, due to the diversity regularisation enforced by the calibration loss $\mathcal{L}_{\text{cal}}$.

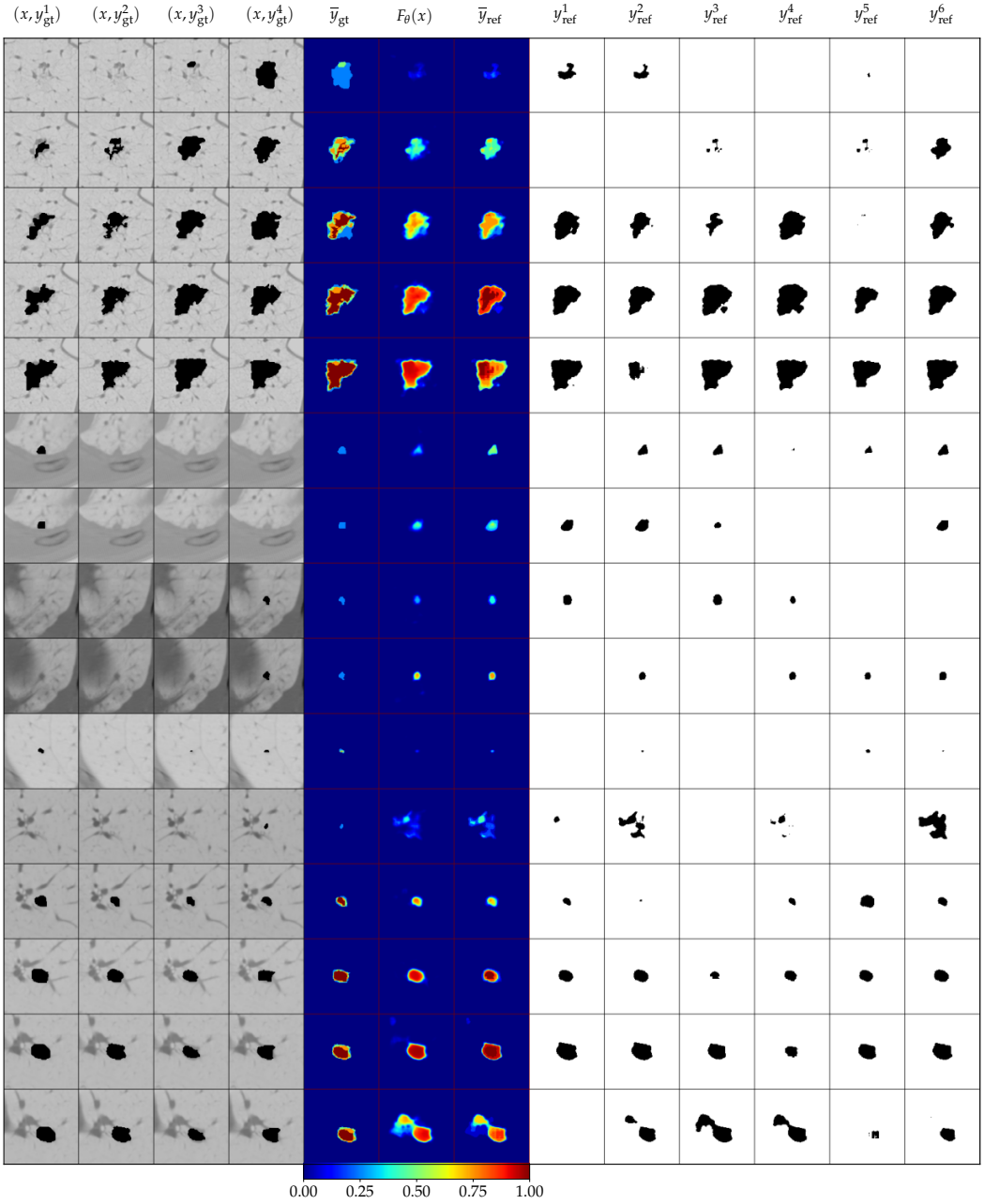


Figure 7: Qualitative results on LIDC samples for the $\mathcal{L}_{\text{cal}}$-regularised cGAN model.

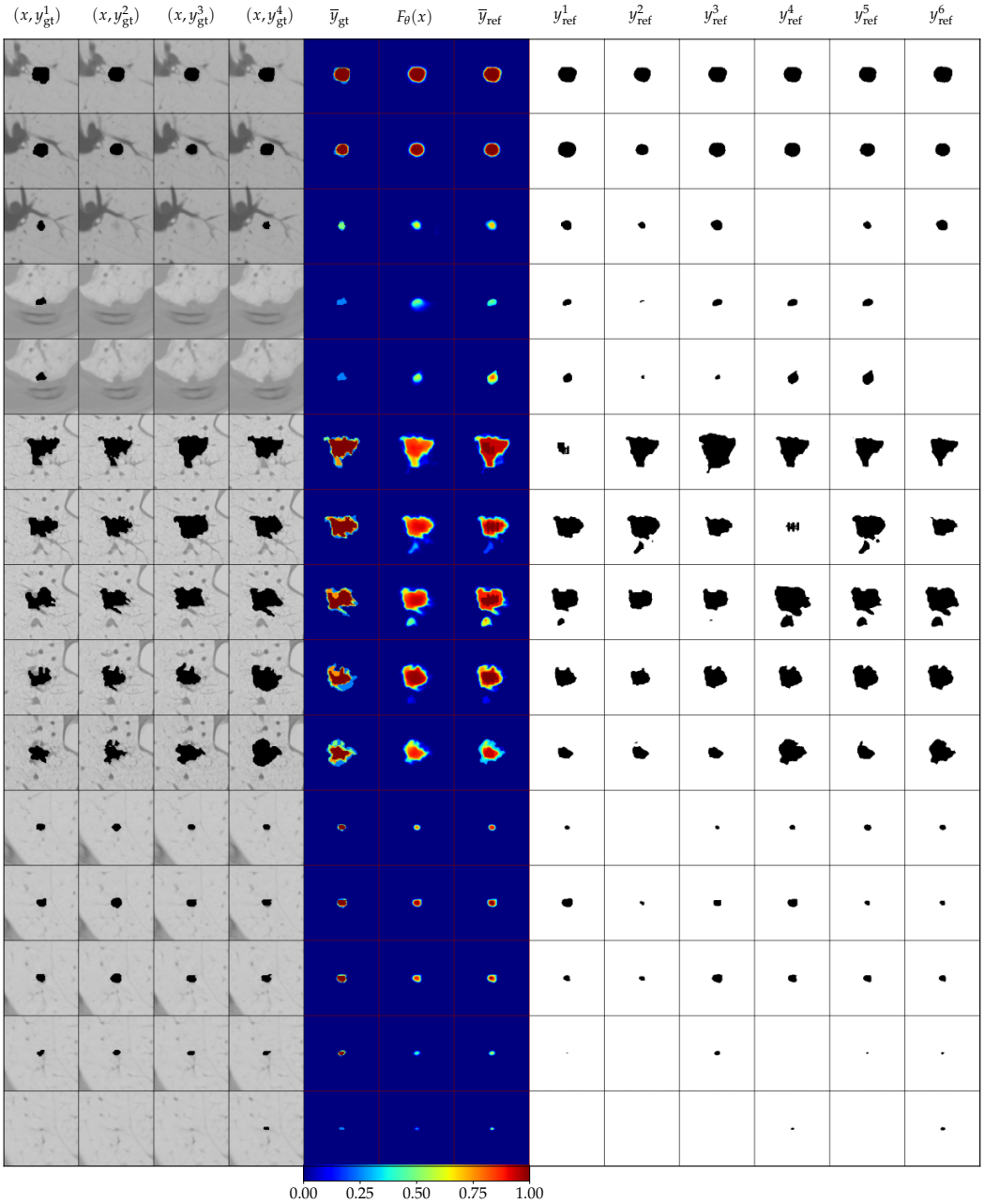


Figure 8: Qualitative results on LIDC samples for the $\mathcal{L}_{\text{cal}}$-regularised cGAN model.

From the qualitative results in Fig. 7 and Fig. 8, it can be seen that the calibration target $F_\theta(x)$ does not always capture well the average of the ground truth distribution $\hat{y}_{\text{gt}}$, affecting the fidelity of the predictive distribution of the refinement network G_ϕ. This further highlights the importance of future work on improving the calibration of F_θ, e. g. implementing the approaches of Guo et al. (2017); Kull et al. (2019).

B.2.2 TUNING THE NUMBER OF REFINEMENT NETWORK SAMPLES

Table 3: Mean GED and HM-IoU scores on LIDC for the $\mathcal{L}_{\text{cal}}$-regularised cGAN with 1, 5, 10, 15 and 20 samples. The number of samples used to compute the GED score is denoted in the parentheses in the header of each column. The arrows ↑ and ↓ denote if higher or lower score is better.

Method	GED ↓ (16)	GED ↓ (50)	GED ↓ (100)	HM-IoU ↑ (16)
cGAN+$\mathcal{L}_{\text{cal}}$ (1)	0.644 ± 0.033	0.643 ± 0.033	0.643 ± 0.033	0.494 ± 0.013
cGAN+$\mathcal{L}_{\text{cal}}$ (5)	0.278 ± 0.000	0.257 ± 0.002	0.252 ± 0.001	0.585 ± 0.003
cGAN+$\mathcal{L}_{\text{cal}}$ (10)	0.277 ± 0.003	0.257 ± 0.003	0.250 ± 0.003	0.589 ± 0.007
cGAN+$\mathcal{L}_{\text{cal}}$ (15)	0.271 ± 0.002	0.250 ± 0.001	0.245 ± 0.003	$\mathbf{0.593 \pm 0.002}$
cGAN+$\mathcal{L}_{\text{cal}}$ (20)	$\mathbf{0.264 \pm 0.002}$	$\mathbf{0.248 \pm 0.004}$	$\mathbf{0.243 \pm 0.004}$	$\mathbf{0.592 \pm 0.005}$

To investigate the effect of the number of samples used to compute $\mathcal{L}_{\text{cal}}$ on the learnt predictive distribution, we experimented on the $\mathcal{L}_{\text{cal}}$-regularised cGAN model using 5, 10, 15 or 20 samples from the refinement network G_ϕ during training. As a control experiment, we also train the same model using one sample. Our results, reported in Table 3, show that increasing the number of samples improves the quality of the predictive distribution, whereas using only one sample collapses it. This is expected because increasing the number of samples reduces the variance of the sample mean $\overline{G}_\phi$ and refines the approximation q_ϕ of the implicit predictive distribution realised by $G_\phi(x, \epsilon)$. Since in our implementation we reuse the samples from G_ϕ in the adversarial component $\mathcal{L}_{\text{G}}$ of the total refinement network loss $\mathcal{L}_{\text{G}}$, the discriminator D_ψ interacts with a larger set of diverse fake samples during each training iteration, thus also improving the quality of $\mathcal{L}_{\text{G}}$.

It is important to note that the benefit of increasing the sample size on the quality of $\mathcal{L}_{\text{cal}}$ highly depends on the intrinsic multimodality in the data. In theory, if the number of samples used matches or exceeds the number of ground truth modes for a given input, it is sufficient to induce a calibrated predictive distribution. However, we usually do not have a priori access to this information. Conversely, if the sample size is too small, the $\mathcal{L}_{\text{cal}}$ loss may introduce bias in the predictive distribution. This could lead to mode coupling or mode collapse, as exemplified in our control experiment with one sample.

In the LIDC dataset, even though we have access to four labels per input image, we argue that the dataset exhibits distributed multimodality, where a given pattern in the input space, e. g. in a patch of pixels, can be associated to many different local labels throughout the dataset. As a result, an input image may correspond to more solutions than the four annotations provided. Therefore increasing the number of samples to more than four shows further improvement in performance. This however can come at the cost of decreased training speed which can be regulated by tuning the sample count parameter while considering the system requirements.

B.2.3 INDUCING MULTIMODALITY IN LATENT VARIABLE MODELS ON THE LIDC DATASET

To examine whether conditioning the source of stochasticity in our model on the input is beneficial, we adapt our framework in order to learn a distribution over a latent code z using variational inference (Graves, 2011). Following most of the existing work on stochastic semantic segmentation (Kohl et al., 2018; 2019; Hu et al., 2019; Baumgartner et al., 2019), we maximise a modified lower bound on the conditional marginal log-likelihood, through a variational distribution $q(z \mid x, y)$. This is realised by minimising the loss function in Higgins et al. (2017), given by:

$$\mathcal{L}_{\text{ELBO}}(x, y) = -\mathbb{E}_{q(z|x,y)}[\log p(y \mid x, z)] + \beta \, \text{KL}(q(z \mid x, y) \,||\, p(z)) \geq -\log p(y \mid x), \tag{12}$$

where β controls the amount of regularisation from a prior distribution $p(z)$ on the approximate posterior $q(z \mid x, y)$. Both $q(z \mid x, y)$ and $p(z)$ are commonly taken as factorised Gaussian distributions.

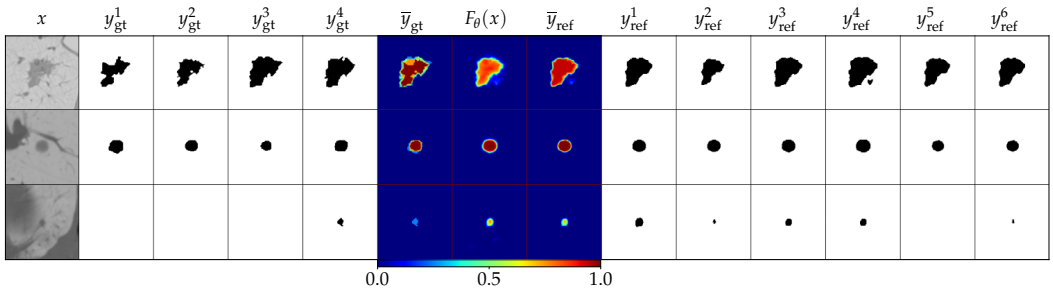
To this end, we compare our cGAN model to two baselines where the refinement network G_ϕ is given as a cVAE-GAN (Larsen et al., 2015). In the first one, we train G_ϕ by complementing the adversarial loss $\mathcal{L}_{\text{G}}$ with Eq. (12), using $\beta \in \{0.1, 1, 10\}$ and a fixed standard normal prior. In the second, we introduce a calibration network and train G_ϕ using Eq. (7). This does not necessitate specifying a prior.

For a fair comparison, we use the same core models for all of our experiments, introducing only minor modifications to the refinement network to convert the deterministic encoder into a probabilistic one with Gaussian output. This is achieved by splitting the output head of the encoder so as to predict the mean and standard deviation of the encoded distribution (Kingma and Welling, 2013; Kendall and Gal, 2017a). Instead of using random noise sampled from a standard Gaussian as our source of stochasticity, the decoder of the refinement network is now injected with latent codes sampled from the Gaussian distribution encoded for each input image. To train the model, we pretrain the F_θ in isolation, and subsequently apply it in inference mode while training G_ϕ. We use a batch size of 32, an 8-dimensional latent code, and use 20 samples to compute $\mathcal{L}_{\text{cal}}$.

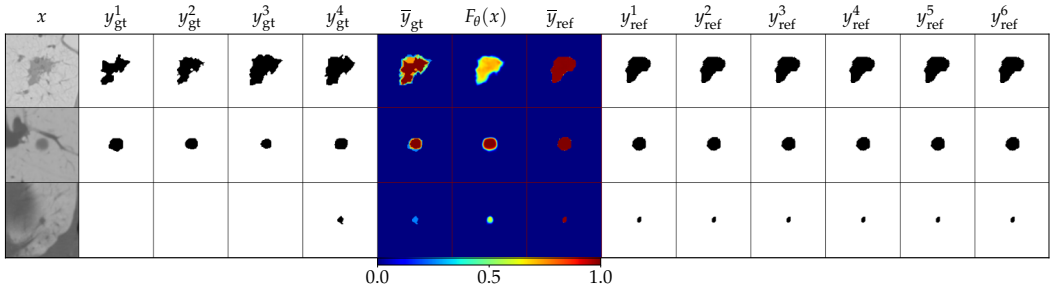
Table 4: GED and H-IoU scores on LIDC. The top section shows the $\mathcal{L}_{\text{ce}}$-regularised baseline and the $\mathcal{L}_{\text{cal}}$-regularised cGAN; the bottom section shows baseline and $\mathcal{L}_{\text{cal}}$-regularised cVAE-GANs. All $\mathcal{L}_{\text{cal}}$-regularised models are trained using 20 samples. The three central columns show the GED score computed with 16, 50 and 100 samples, respectively. The last column shows the HM-IoU score, computed with 16 samples. The arrows $\uparrow$ and $\downarrow$ indicate whether higher or lower score is better.

Method	GED $\downarrow$ (16)	GED $\downarrow$ (50)	GED $\downarrow$ (100)	HM-IoU $\uparrow$ (16)
cGAN+$\mathcal{L}_{\text{ce}}$	0.639 ± 0.002	—	—	0.477 ± 0.004
cGAN+$\mathcal{L}_{\text{cal}}$	$\mathbf{0.264 \pm 0.002}$	$\mathbf{0.248 \pm 0.004}$	0.243 ± 0.004	$\mathbf{0.592 \pm 0.005}$
cVAE-GAN (β=0.1)	0.577 ± 0.095	—	—	0.484 ± 0.006
cVAE-GAN (β=1)	0.596 ± 0.078	—	—	0.474 ± 0.005
cVAE-GAN (β=10)	0.609 ± 0.061	—	—	0.482 ± 0.010
cVAE-GAN+$\mathcal{L}_{\text{cal}}$ (β=0)	0.272 ± 0.006	0.252 ± 0.006	0.246 ± 0.006	$\mathbf{0.593 \pm 0.003}$

We show that the cVAE-GAN model trained with $\mathcal{L}_{\text{cal}}$ instead of the traditional complexity loss term, $\text{KL}(q(z \mid x, y) \mid\mid p(z))$ from Eq. (12) is able to learn a distribution over segmentation maps, and performs similarly to our cGAN+$\mathcal{L}_{\text{cal}}$ model. This is important because it abrogates the need for specifying a latent-space prior, which is often selected for computational convenience, rather than task relevance (Hafner et al., 2018; Louizos et al., 2019). On the other hand, our cVAE-GAN models trained using the KL-divergence complexity term showed limited diversity, even for large β values. The results, shown quantitatively in the bottom part of Table 4, and qualitatively in Fig. 9a, demonstrate that the interaction between $\mathcal{L}_{\text{G}}$ and $\mathcal{L}_{\text{cal}}$ can sufficiently induce a multimodal predictive distribution in latent variable models, and indicate that for the purpose of stochastic semantic segmentation the use of a probabilistic encoder is not strictly required.



(a)



(b)

Figure 9: LIDC validation samples for the **(a)** cVAE-GAN and **(b)** cGAN+$\mathcal{L}_{\text{ce}}$ baseline model.

B.3 CITYSCAPES

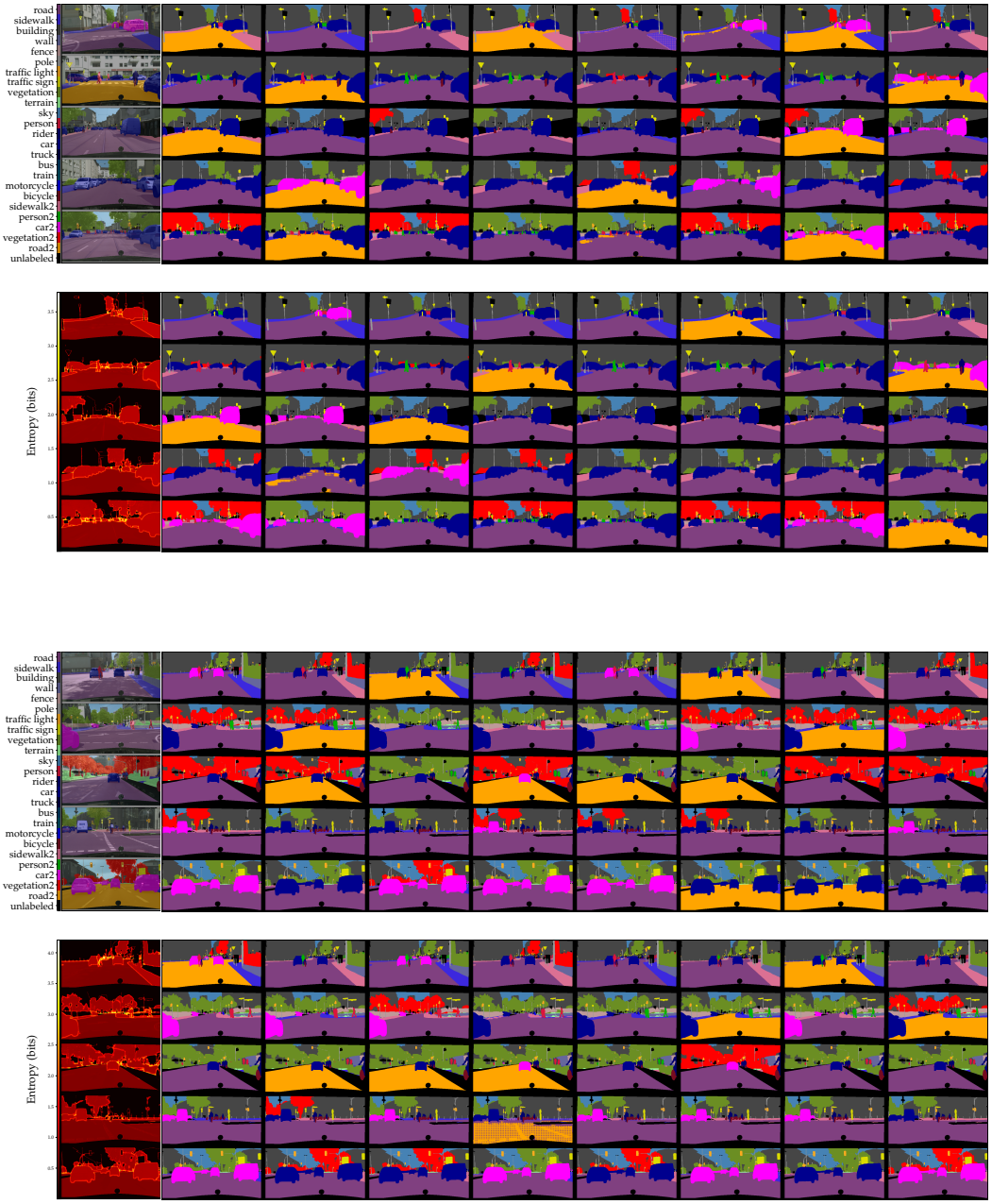


Figure 10: 10 input images, the corresponding aleatoric maps from the calibration network and 16 samples from the refinement network. For visualisation purposes, the samples are split into 8 per row.

B.3.1 QUALITATIVE ANALYSIS

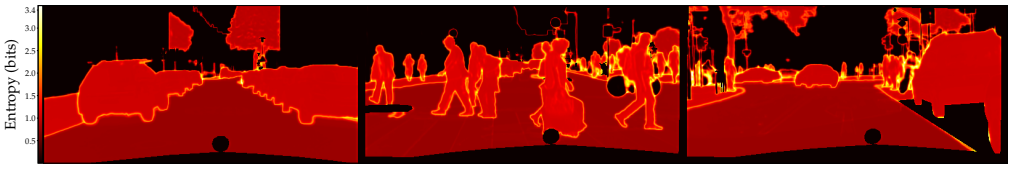
In this section we provide additional qualitative results for the $\mathcal{L}_{\text{cal}}$-regularised cGAN model trained on the modified Cityscapes dataset (Kohl et al., 2018). In Fig. 10, we show 16 randomly sampled predictions for representative input images x, and their corresponding aleatoric uncertainty maps, obtained by computing the entropy of the output of the calibration network, $\mathbb{H}(F_\theta(x))$, as done in Kendall and Gal (2017a). The predicted samples are of high quality, evident by object coherence and crisp outlines, and high diversity, where all classes are well represented. Our model effectively learns the entropy of the ground truth distribution in the stochastic classes (*sidewalk*, *person*, *car*, *vegetation* and *road*), as their distinct entropy levels are captured as different

shades of red in the entropy maps, corresponding to the different flip probabilities ($^8/_{17}$, $^7/_{17}$, $^6/_{17}$, $^5/_{17}$ and $^4/_{17}$ respectively). Additionally, it can be seen that edges or object boundaries are also highlighted in the aleatoric uncertainty maps, which reflects inconsistency during manual annotation, which often occurs on input pixels that are difficult to segment.

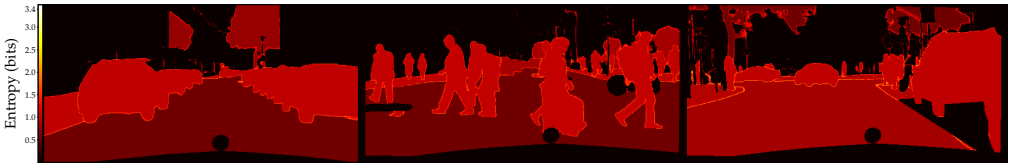
Fig. 11c shows the entropy of the predictive distribution of the refinement network G_ϕ, $\mathbb{H}\big(\overline{G}_\phi(F_\theta(x))\big)$, where $\overline{G}_\phi(F_\theta(x))$ is computed as the average of 16 samples from G_ϕ. Our results demonstrate that $\mathbb{H}\big(\overline{G}_\phi(F_\theta(x))\big)$ is similar to $\mathbb{H}(F_\theta(x))$, depicted in Fig. 11b, as encouraged by the $\mathcal{L}_{\text{cal}}$ regularisation. Notice that object boundaries are also highlighted in $\mathbb{H}\big(\overline{G}_\phi(F_\theta(x))\big)$, indicating that our model captures shape ambiguity as well as class ambiguity. However, some uncertainty information from $\mathbb{H}(F_\theta(x))$ is not present in $\mathbb{H}\big(\overline{G}_\phi(F_\theta(x))\big)$, e. g. the entropy of the different stochastic classes are not always consistent across images, as evident from the different shades of red seen for the road class in Fig. 11c. We expect that increasing the number of samples from the refinement network will improve the aleatoric uncertainty estimates. Nevertheless, the sample-free estimate extracted from $F_\theta(x)$ is cheaper to obtain and more reliable than the sample-based average from G_ϕ, highlighting an important benefit of our cascaded approach.

Finally, we illustrate in Fig. 11d the high confidence of the predictions from the refinement network $G_\phi(F_\theta(x))$, reflected by their low entropy, $\mathbb{H}(G_\phi(F_\theta(x)))$. This is attributed to the adversarial component in the refinement loss function, which encourages the predictions to assume a one-hot representation, matching the ground truth annotations. Even though each prediction of the refinement network is highly confident, the average of the predictions $\overline{G}_\phi(F_\theta(x))$ is calibrated, as shown in Fig. 11c. This is a clear illustration of the advantage of complementing the adversarial loss term $\mathcal{L}_{\text{G}}$ with the calibration loss term $\mathcal{L}_{\text{cal}}$ in the training objective for the refinement network.

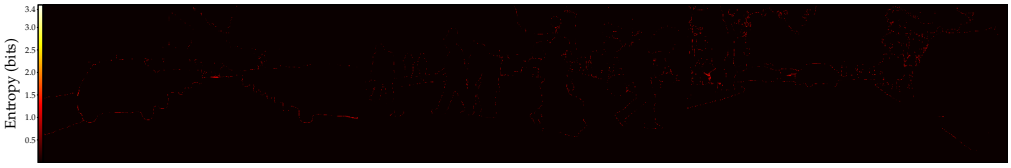
(a)



(b)



(c)



(d)

Figure 11: **(a)** Three input images overlaid with the corresponding labels; **(b)** The aleatoric maps from the calibration network; **(c)** Aleatoric maps computed as the entropy of the average of 16 predictions of the refinement network; **(d)** The entropy of one sample of the refinement network output for each input image.