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 $y_{\text{ref}}^{i,t} = G_{\phi_t}(F_{\theta^*}(x_t), \epsilon_i) \quad \text{where } \epsilon_i \sim \mathcal{N}(0, 1)$
    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 $y_{\text{ref}}^{i} = G_{\phi^*}(F_{\theta^*}(x), \epsilon_i) \quad \text{where } \epsilon_i \sim \mathcal{N}(0, 1)$
  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 [29] 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 [43], 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. 1D bimodal regression

In the following we derive the mean squared error form of the cross entropy and calibration losses used in experiment Section 4.1 under the assumption that the likelihood model for q_θ and q_ϕ is a univariate Gaussian distribution with a fixed unit variance. Using the setup

from Eq. (2) it then follows that:

$$\mathcal{L}_{\text{ce}}(\mathcal{D},\theta) = -\mathbb{E}_{p_{\mathcal{D}}(x,y)}[\log \mathcal{N}(y \mid F_\theta(x),\, 1)] \tag{10}$$

$$= \frac{1}{2}\mathbb{E}_{p_{\mathcal{D}}(x,y)}\left[(y - F_\theta(x))^2\right] + \text{const.} \tag{11}$$

Based on the definition of the calibration loss in Eq. (6) we show that:

$$\mathcal{L}_{\text{cal}}(\mathcal{D},\theta,\phi) = \mathbb{E}_{p_{\mathcal{D}}}\left[\text{KL}\big(\mathcal{N}\big(y \mid \overline{G}_\phi(F_\theta(x)),\, 1\big) \;||\; \mathcal{N}(y \mid F_\theta(x),\, 1)\big)\right] \tag{12}$$

$$= \mathbb{E}_{p_{\mathcal{D}}}\left[\mathbb{E}_{\mathcal{N}\left(y|\overline{G}_\phi(F_\theta(x)),\,1\right)}\left[\log \mathcal{N}\big(y \mid \overline{G}_\phi(F_\theta(x)),\, 1\big) - \log \mathcal{N}(y \mid F_\theta(x),\, 1)\right]\right] \tag{13}$$

$$= \mathbb{E}_{p_{\mathcal{D}}}\left[\mathbb{E}_{\mathcal{N}\left(y|\overline{G}_\phi(F_\theta(x)),\,1\right)}\left[-\frac{1}{2}\left(y - \overline{G}_\phi(F_\theta(x))\right)^2 + \frac{1}{2}\left(y - F_\theta(x)\right)^2\right]\right] + \text{const} \tag{14}$$

$$= \mathbb{E}_{p_{\mathcal{D}}}\left[\mathbb{E}_{\mathcal{N}\left(y|\overline{G}_\phi(F_\theta(x)),\,1\right)}\left[y\overline{G}_\phi(F_\theta(x)) - \frac{1}{2}\overline{G}_\phi(F_\theta(x))^2 - yF_\theta(x) + \frac{1}{2}F_\theta(x)^2\right]\right] + \text{const} \tag{15}$$

$$= \mathbb{E}_{p_{\mathcal{D}}}\left[\overline{G}_\phi(F_\theta(x))^2 - \frac{1}{2}\overline{G}_\phi(F_\theta(x))^2 - \overline{G}_\phi(F_\theta(x))F_\theta(x) + \frac{1}{2}F_\theta(x)^2\right] + \text{const} \tag{16}$$

$$= \frac{1}{2}\mathbb{E}_{p_{\mathcal{D}}}\left[\left(\overline{G}_\phi(F_\theta(x)) - F_\theta(x)\right)^2\right] + \text{const.} \tag{17}$$

A.3. LIDC

Architectures For the calibration network, F_θ, we use the encoder-decoder architecture from SegNet [2], 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 $\mathcal{L}_{\text{G}} : \mathcal{L}_{\text{cal}} = 1 : 0.5$, where $\mathcal{L}_{\text{G}}$ is the adversarial component of the loss, and $\mathcal{L}_{\text{cal}}$ is the calibration loss component. In practice, the actual weights used are 10 for $\mathcal{L}_{\text{G}}$, and 5 for $\mathcal{L}_{\text{cal}}$.

Evaluation Following [31], [32], [23], and [4] we use the Generalised Energy Distance (GED) [52] metric, given as:

$$D^2_{\text{GED}}(p_{\mathcal{D}}, q_\phi) = 2\mathbb{E}_{s\sim q_\phi, y\sim p_{\mathcal{D}}}[d(s,y)] - \mathbb{E}_{s,s'\sim q_\phi}\left[d(s,s')\right] - \mathbb{E}_{y,y'\sim p_{\mathcal{D}}}\left[d(y,y')\right], \tag{18}$$

where $d(s,y) = 1 - \text{IoU}(s,y)$. Intuitively, the first term of Eq. (18) 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 [32] 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, [32] 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 [32], 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 [4], [44] and [21], whose GED scores we report in Table 1, use test sets that differ from the original splits defined in [31], which are used in [31, 32] and our work. [4] and [44] uses a random 60:20:20 split for the training, testing and validation sets, and 100 samples to compute the GED score, whereas [21] 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 CAR model on the split methodology defined by [4], and also used by [44], to enable a fairer comparison. This improved our GED score from 0.243 ± 0.004 to 0.228 ± 0.009, while the HM-IoU, evaluated using 16 samples, remained similar at 0.590 ± 0.007 (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.4. 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 [31] 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:

$$D^2_{\mathrm{GED}}(p_{\mathcal{D}}, q_\phi) = 2\mathbb{E}_{s\sim q_\phi, y\sim p_{\mathcal{D}}}[d(s,y)w(y)] - \mathbb{E}_{s,s'\sim q_\phi}\left[d(s,s')\right] - \mathbb{E}_{y,y'\sim p_{\mathcal{D}}}\left[d(y,y')w(y)w(y')\right], \tag{19}$$

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 [31]. 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. 6, Section 4.2.2 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 4 we present supplementary results for the bimodal regression experiment and the LIDC and Cityscapes segmentation experiments.

B.1. 1D bimodal regression

Fig. 8 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. 8b and Fig. 8d respectively. The results show that high bias is harder to learn, reflected by a slowed down convergence, however, the CAR 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).

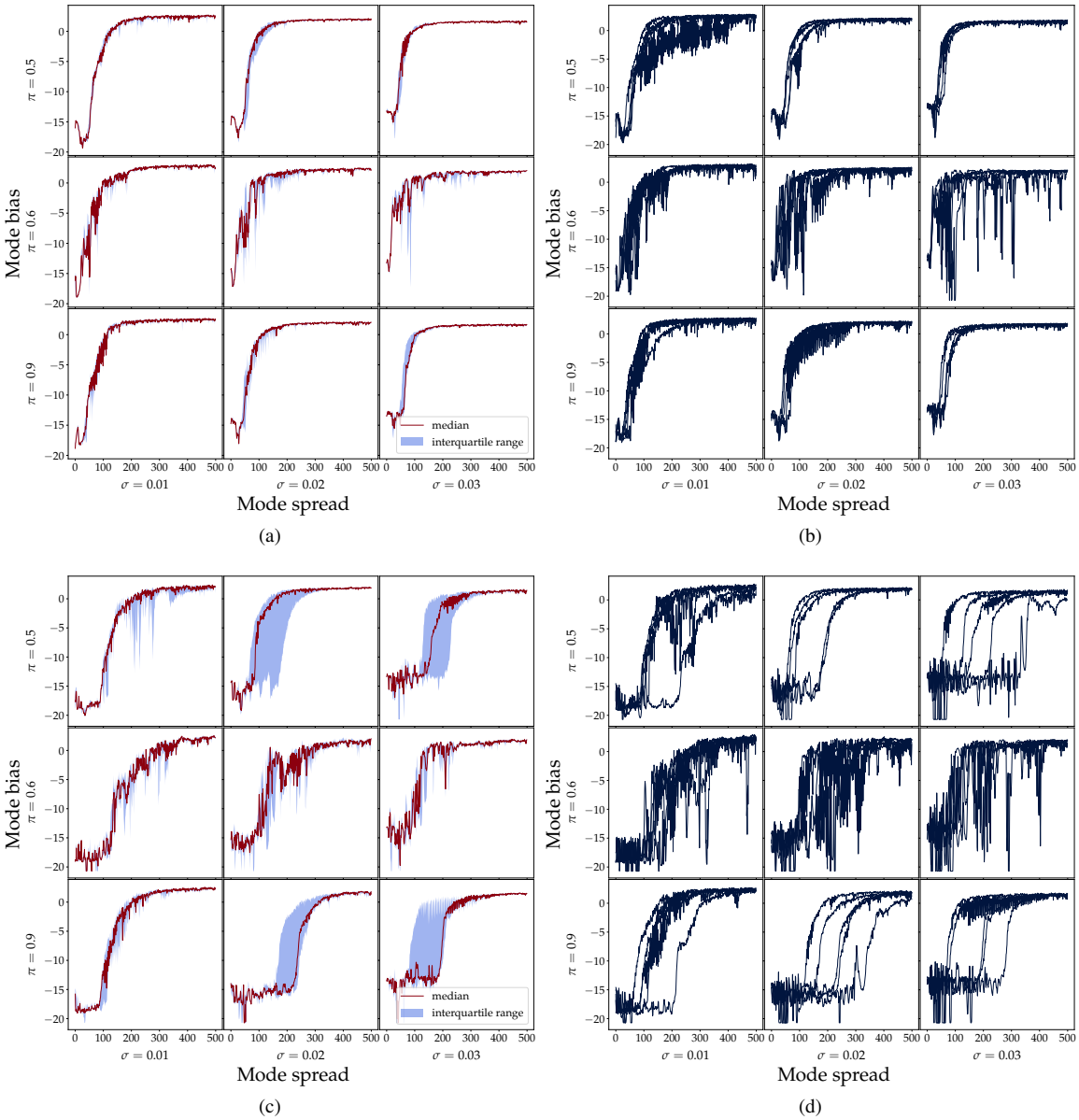


Figure 8: Log-likelihood curves for 5 runs on each of the 9 data configurations. **(a)** No calibration loss ($\lambda = 0$), averaged. **(b)** No calibration loss, individual runs. **(c)** With calibration loss ($\lambda = 1$), averaged. **(d)** With calibration loss, individual runs.

B.2. LIDC

B.2.1 Qualitative Analysis

To further examine the CAR model trained on the LIDC dataset, we illustrate representative qualitative results in Fig. 9 and Fig. 10. 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 $\mathcal{L}_{\text{cal}}$.

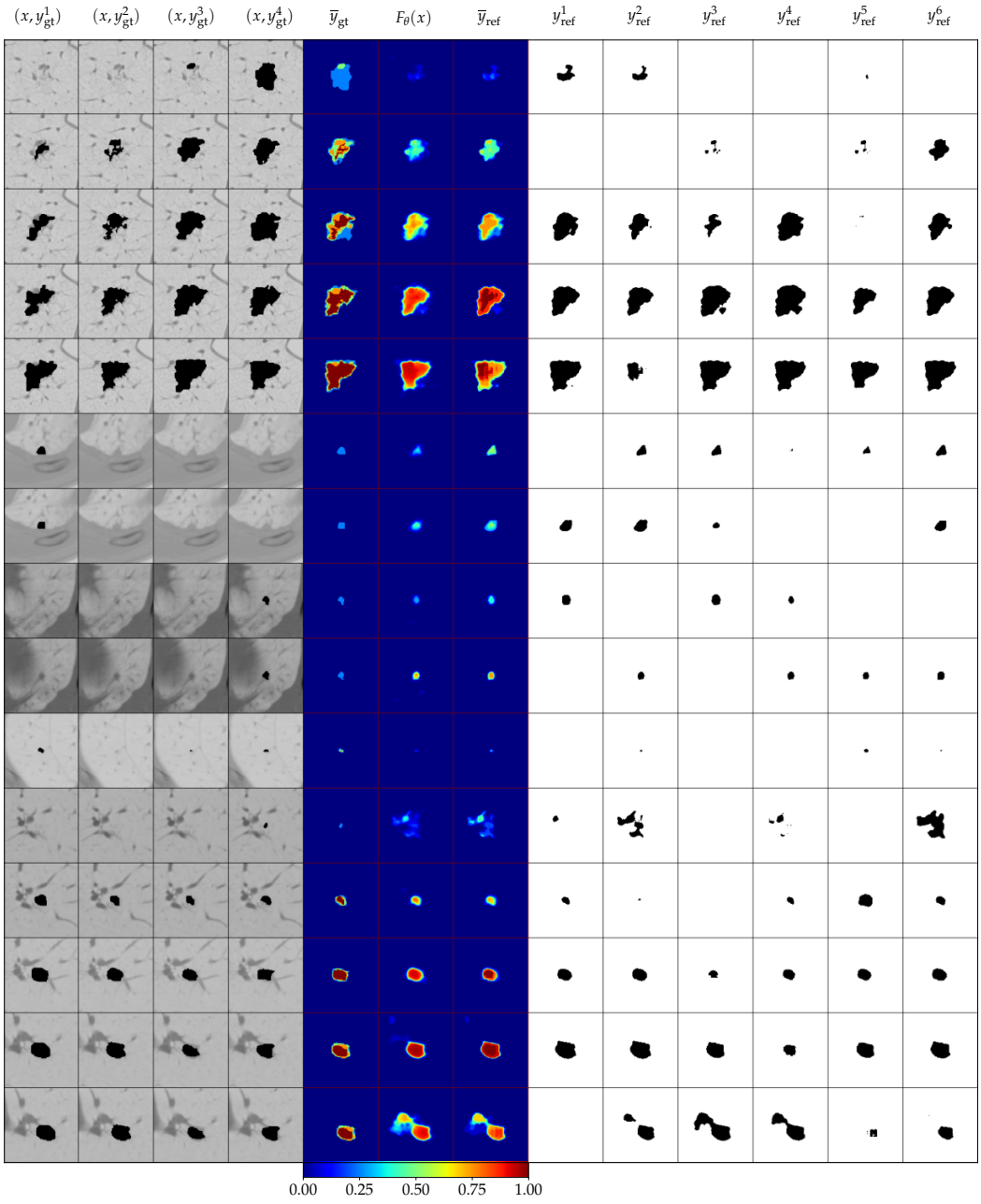


Figure 9: Qualitative results on LIDC samples for the CAR model.

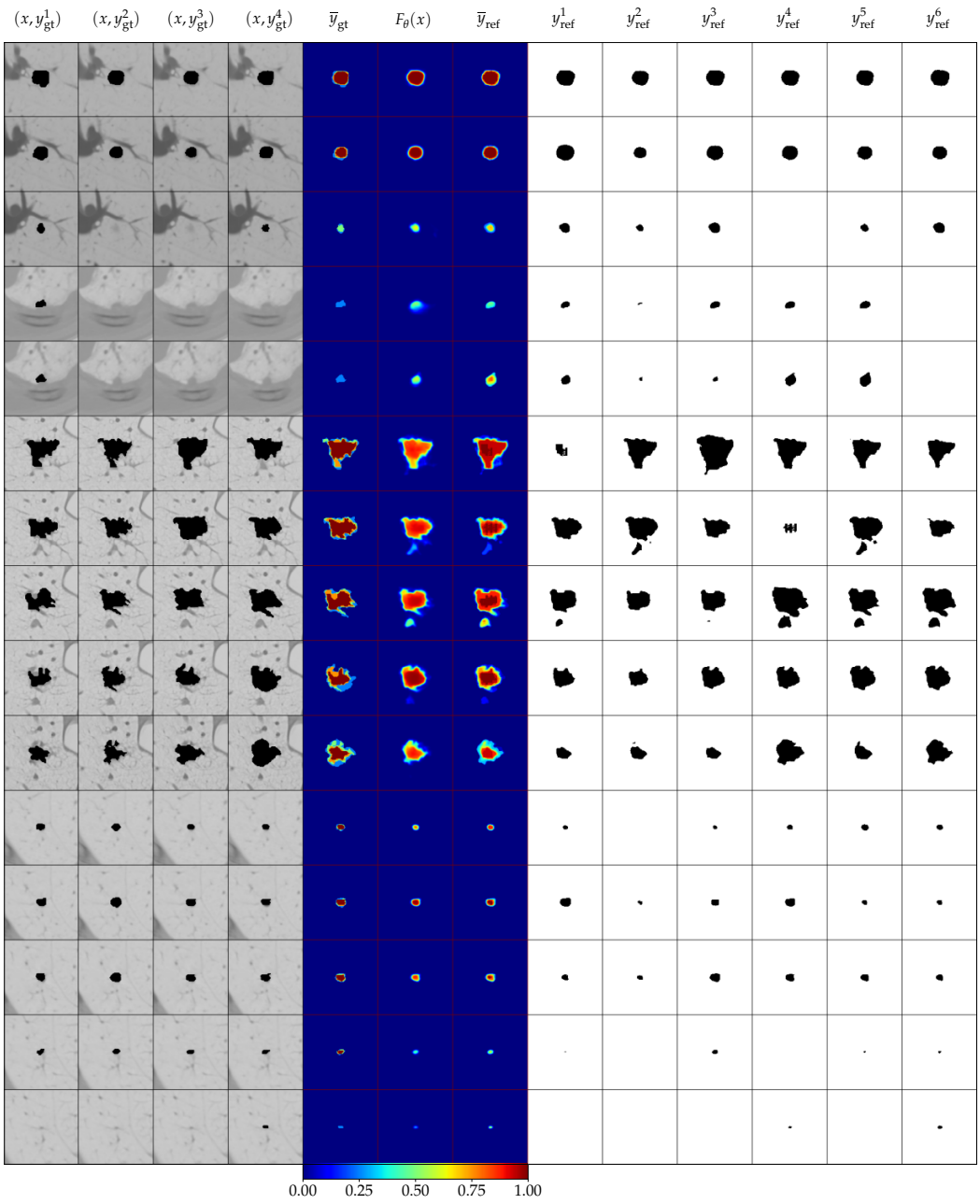


Figure 10: Qualitative results on LIDC samples for the CAR model.

From the qualitative results in Fig. 9 and Fig. 10, it can be seen that the calibration target $F_\theta(x)$ does not always precisely capture 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 indicates the importance of future work on improving the calibration of F_θ, e. g. implementing the approaches of [17, 34].

B.2.2 Tuning the number of refinement network samples

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)	$\mathbf{0.264 \pm 0.002}$	$\mathbf{0.248 \pm 0.004}$	$\mathbf{0.243 \pm 0.004}$	$\mathbf{0.592 \pm 0.005}$

Table 3: Mean GED and HM-IoU scores on LIDC for the CAR model ($\mathcal{L}_{\text{cal}}$-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.

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 CAR 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, instead of taking samples from a fixed noise distribution $\epsilon \sim \mathcal{N}(0, 1)$, using variational inference [16]. Following most of the existing work on stochastic semantic segmentation [31, 32, 21, 4], 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 [19], given by:

$$\mathcal{L}_{\text{ELBO}}(x, y) = -\mathbb{E}_{q(z|x,y)}[\log p(y \mid x, z)] + \beta \, \text{KL}(q(z \mid x, y) \mid\mid p(z)) \geq -\log p(y \mid x), \tag{20}$$

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 CAR model to two baselines where the refinement network G_ϕ is given as a cVAE-GAN [36]. In the first one, we train G_ϕ by complementing the adversarial loss $\mathcal{L}_{\text{adv}}$ with Eq. (20), 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). Note that this does not necessitate specifying a prior.

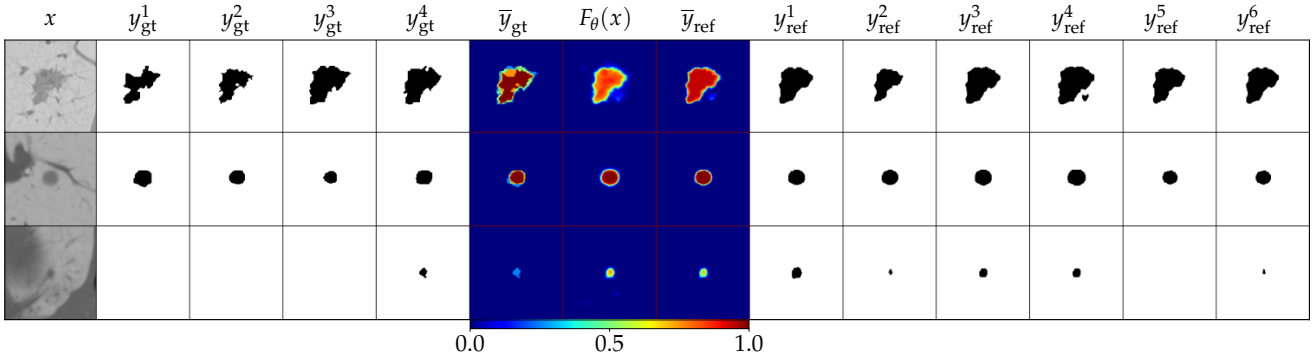
For a fair comparison, we use the same core models for all of our experiments, introducing only minor modifications to the refinement network to convert the deterministic encoder into a probabilistic one with Gaussian output. This is achieved by splitting the output head of the encoder so as to predict the mean and standard deviation of the encoded distribution [30, 28]. 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 $\mathcal{L}_{\text{cal}}$.

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. (20) is able to learn a distribution over segmentation maps, and performs similarly to the CAR 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 [18, 40]. 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. 11a, demonstrate that the

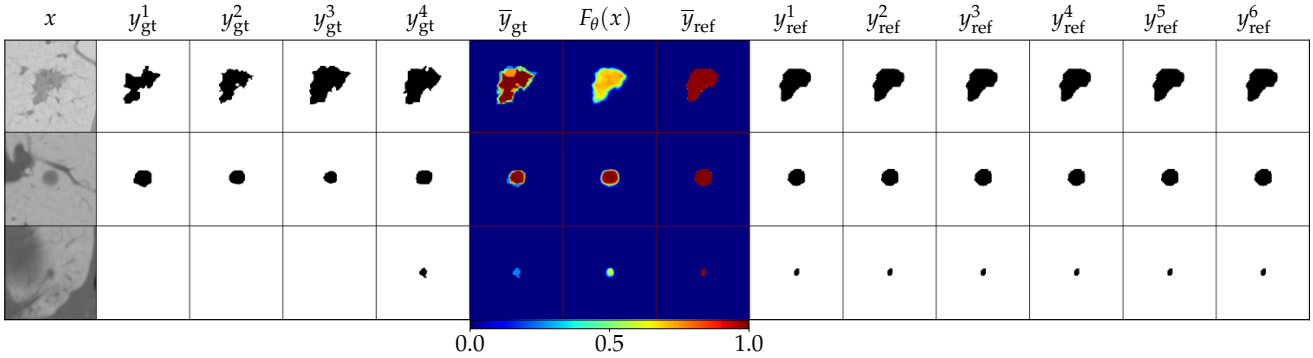
Method	GED ↓ (16)	GED ↓ (50)	GED ↓ (100)	HM-IoU ↑ (16)
cGAN+$\mathcal{L}_{\text{ce}}$	0.639 ± 0.002	—	—	0.477 ± 0.004
CAR (ours)	$\mathbf{0.264 \pm 0.002}$	$\mathbf{0.248 \pm 0.004}$	0.243 ± 0.004	$\mathbf{0.592 \pm 0.005}$
cVAE-GAN (β=0.1)	0.577 ± 0.095	—	—	0.484 ± 0.006
cVAE-GAN (β=1)	0.596 ± 0.078	—	—	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	$\mathbf{0.593 \pm 0.003}$

Table 4: GED and HM-IoU scores on LIDC. The top section shows the $\mathcal{L}_{\text{ce}}$-regularised baseline and the CAR model; 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 ↑ and ↓ indicate whether higher or lower score is better.

interaction between $\mathcal{L}_{\text{adv}}$ 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 11: LIDC validation samples for the **(a)** cVAE-GAN and **(b)** cGAN+$\mathcal{L}_{\text{ce}}$ baseline model.

B.3. Cityscapes

B.3.1 Qualitative Analysis

In this section we provide additional qualitative results for the CAR model trained on the modified Cityscapes dataset [31]. In Fig. 12, 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 [28]. 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, reflecting inconsistency during manual annotation, which often occurs on input pixels that are difficult to segment.

Fig. 13d 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. 13c, 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

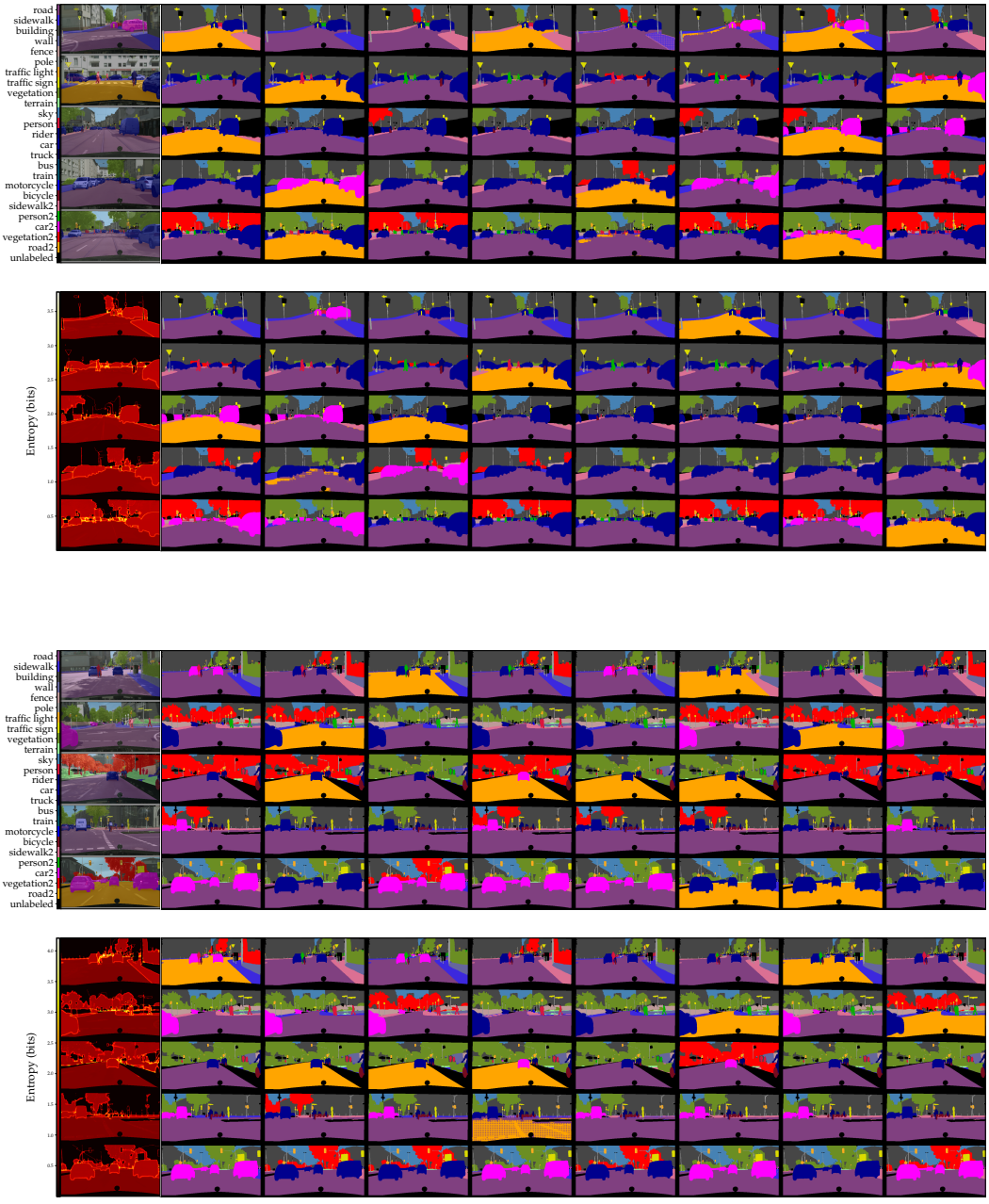


Figure 12: 10 input images, the corresponding aleatoric maps from the calibration network and 16 samples from the refinement network. For visualisation purposes, the samples are split into 8 per row.

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. 13d. 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. 13e 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 confident predictions to mimic the one-hot representation of 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. 13d. 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.

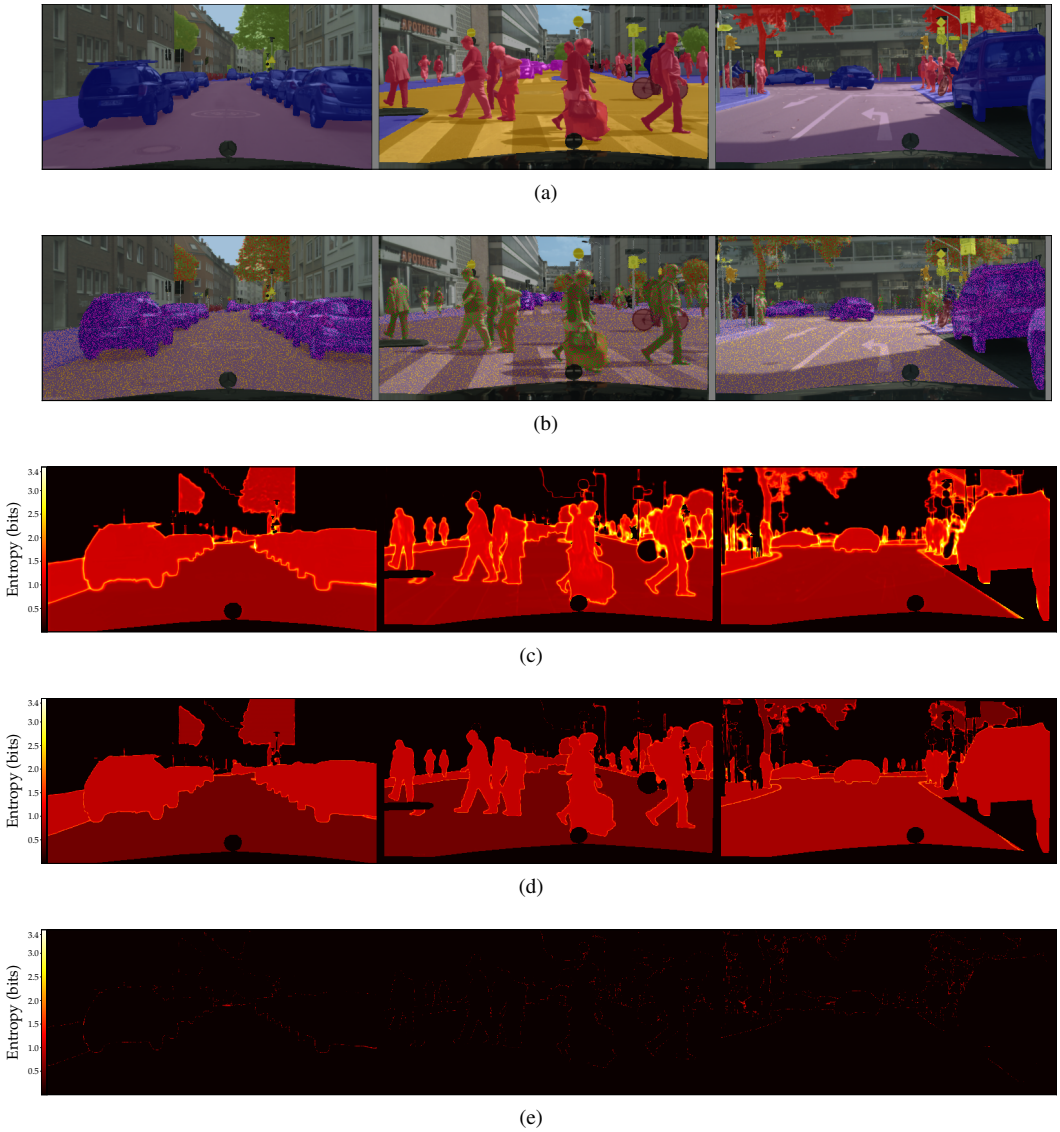


Figure 13: **(a)** Three input images overlaid with the corresponding labels; **(b)** Incoherent samples from the predictive distribution of the calibration network; **(c)** The aleatoric maps from the calibration network; **(d)** Aleatoric maps computed as the entropy of the average of 16 predictions of the refinement network; **(e)** The entropy of one sample of the refinement network output for each input image.