Appendix

A. Proofs of Direct Uncertainty Prediction Results

We first prove Theorem 1.

Proof. To show unbiasedness of h_{dup}, we need to show that $\mathbb{E}[h_{dup}] = \mathbb{E}[U(\mathbf{Y})]$. But from the tower law (law of total expectation):

$$\mathbb{E}[h_{dup}] = \mathbb{E}\Big[\mathbb{E}\big[U(\mathbb{E}[\mathbf{Y}|O])\big|g(O)\big]\Big] = \mathbb{E}[U(\mathbb{E}[\mathbf{Y}|O])]$$

To prove the biasedness of h_{uvc}, first note that

$$h_{uvc} = U(\mathbb{E}[\mathbf{Y}|g(O)]) = U(\mathbb{E}[\mathbb{E}[\mathbf{Y}|O]|g(O)])$$

by the conditional independence of $Y, g(O)$ given O. Next, by the fact that $U(\cdot)$ is concave and Jensen's inequality,

$$h_{uvc} = U(\mathbb{E}[\mathbb{E}[\mathbf{Y}|O]|g(O)]) \geq \mathbb{E}[U(\mathbb{E}[\mathbf{Y}|O]|g(O)] = h_{dup}$$

This is a strict inequality whenever the distribution of posteriors induced by conditioning on $g(O)$ is not a point-mass. Therefore we have that h_{uvc} overestimates the the true uncertainty $U(\mathbf{Y})$. $\square$

For specific $U(\cdot)$, we can compute the bias term by first computing $h_{uvc} - h_{dup}$ and then taking an expectation. For $U_{disagree}$, we have

$$h_{uvc} = U(\mathbb{E}[\mathbf{Y}|g(O)]) = 1 - \sum_l \mathbb{E}[\mathbb{E}[Y_l|O]|g(O)]^2$$

and

$$h_{dup} = \mathbb{E}[U(\mathbb{E}[\mathbf{Y}|O])|g(O)] = 1 - \sum_l \mathbb{E}[\mathbb{E}[Y_l|O]^2|g(O)]$$

And so,

$$h_{uvc} - h_{dup} = \sum_l \mathbb{E}[\mathbb{E}[Y_l|O]^2|g(O)] - \sum_l \mathbb{E}[\mathbb{E}[Y_l|O]|g(O)]^2$$

But this is just

$$Var\left(\sum_l \mathbb{E}[\mathbb{E}[Y_l|O]|g(O)]\right)$$

Taking expectations over values of $g(O)$ gives the bias, i.e.

$$\mathbb{E}\left[Var\left(\sum_l \mathbb{E}[\mathbb{E}[Y_l|O]|g(O)]\right)\right]$$

For U_{var}, we have

$$h_{uvc} = \sum_l l^2\mathbb{E}[\mathbb{E}[Y_l|O]|g(O)] - \left(\sum_l l\mathbb{E}[\mathbb{E}[Y_l|O]|g(O)]\right)^2$$

and

$$h_{dup} = \mathbb{E}\left[\sum_l l^2\mathbb{E}[Y_l|O] - \left(\sum_l l\mathbb{E}[Y_l|O]\right)^2\middle| g(O)\right]$$

And so $h_{uvc} - h_{dup}$ becomes

$$\mathbb{E}\left[\left(\sum_l l\mathbb{E}[Y_l|O]\right)^2\middle| g(O)\right] - \left(\sum_l l\mathbb{E}[\mathbb{E}[Y_l|O]|g(O)]\right)^2$$

Which is just

$$Var\left(\sum_l l\cdot\mathbb{E}[Y_l|O]\middle| g(O)\right)$$

Taking expectations over values of $g(O)$ like before gives the result.

B. Mixture of Gaussians Setting

We train a DUP and UVC (Figure 1) on a synthetic task where data is generated from a mixture of Gaussians. All of our settings have uniform mixtures of Gaussians, with the Gaussian mean vectors being drawn from $\mathcal{N}(0, 1/d)$ (d corresponding to the dimension) so that in expectation, each mean vector has norm 1. The variance is set to be the identity. Like before, we set $x = g(o) = |o|$. We draw five labels for each x from the posterior distribution over Gaussian centers given x, and apply $U_{disagree}()$ to the empirical histogram. As with the medical imaging application, we threshold these uncertainty scores (with threshold 0.5) to give a binary low uncertainty/high uncertainty label, which we use to train our DUPs and UVCs. Results are given in percentage AUC to account for some settings having unbalanced classes.

We train fully connected networks with two hidden layers of width 300 on this task, using the SGD with momentum optimizer and an initial learning rate of 0.01.

C. SVHN and CIFAR-10 Setting

In Section 2.2, we train DUP and UVC models to predict label disagreement on a synthetic task on SVHN and CIFAR-10. The task setup is as follows: for each image in SVHN/CIFAR-10, we decide on a variance $(0, 1, 2, 3)$ for a Gaussian filter that is applied to the image. Three labels are then drawn for the image from a noisy distribution over labels, with the label noise distribution depending on the variance of the Gaussian filter. Specifically, for a Gaussian filter with variance 0, the noise distribution is just a point mass on the true label. For a Gaussian filter with variance 1, the three labels are drawn from a distribution with 0.02 mass on four incorrect labels, and the remaining 0.92 mass on the correct label. For variance equal to 2, the labels are drawn from a distribution with 0.08 mass on four different labels, and remaining mass on the true label. For variance 3, this mass is now 0.12 on the incorrect labels.

A simple conv network, with 3x3 kernels and channels $64 - -128 - -256$, followed by fully connected layers of width 1000 and 200 (each with batch normalization) is trained on this dataset, with the UVC model trained on the empirical histogram, and the DUP model trained on a binary agree/disagree target. (Disagreement threshold is if at least one label disagrees.) We find that DUP outperforms UVC on both SVHN and CIFAR-10.

Learned Features Interestingly, we also observe that the features learned by the DUP and UVC models are different to each other. We apply saliency maps, specifically SmoothGrad (Smilkov et al., 2017) and IntGrad (Sundararajan et al., 2017) to study the features that DUP and UVC pay attention to in the input image.

D. Details of DUP in the Medical Domain

As described in Section 4, to train DUP models, we threshold the scores given by applying $U_{disagree}, U_{var}$ to the data $(x_i, \hat{\mathbf{p}}_i)$. Preliminary experiments in trying to directly regress onto the raw scores using mean-squared error performed poorly.

We threshold the scores as follows. For U_{var} we thresholded at approximately $2/9$, the variance when three doctors have more than an 'off by one' disagreement: more than a single disagreement, or a single grade disagreement.

For $U_{disagree}$, where only the number of disagreements counts, we thresholded at 0.3, to prioritize being sensitive enough to disagreement cases and having more than 20% of the data marked as high disagreement. We also experimented with using soft targets for disagreement classification, but the results (Table 6) showed that this was less effective than than having the binary $0/1$ scores, likely because this makes the classification problem more like a regression.

Our model consists of an Inception-v3 base, with the ImageNet head removed and a small (2 hidden layer, 300 hidden units) fully connected neural network using Inception-v3 PreLogits to perform DUP. The full Inception-v3 network is trained with a batch size of 8 and learning rate 0.001 with the Adam optimizer. For training only the small neural network, we use the SGD with momentum optimizer, a batch size of 32 and learning rate of 0.01.

Prelogits, Calibration and Regularization Our training data for DUP models, $T_{train}^{(var)}, T_{train}^{(disagree)}$, only consists of x_i with more than one label, and is too small to effectively train an Inception sized model end to end. Therefore, we use the prelogit embeddings of x_i from a pretrained DR classification model (Histogram-E2E), and training smaller models on top of these embeddings. We do this both for the baseline, getting the Histogram-PC model, as well as the DUP models, Variance-PRC and Disagree-PC.

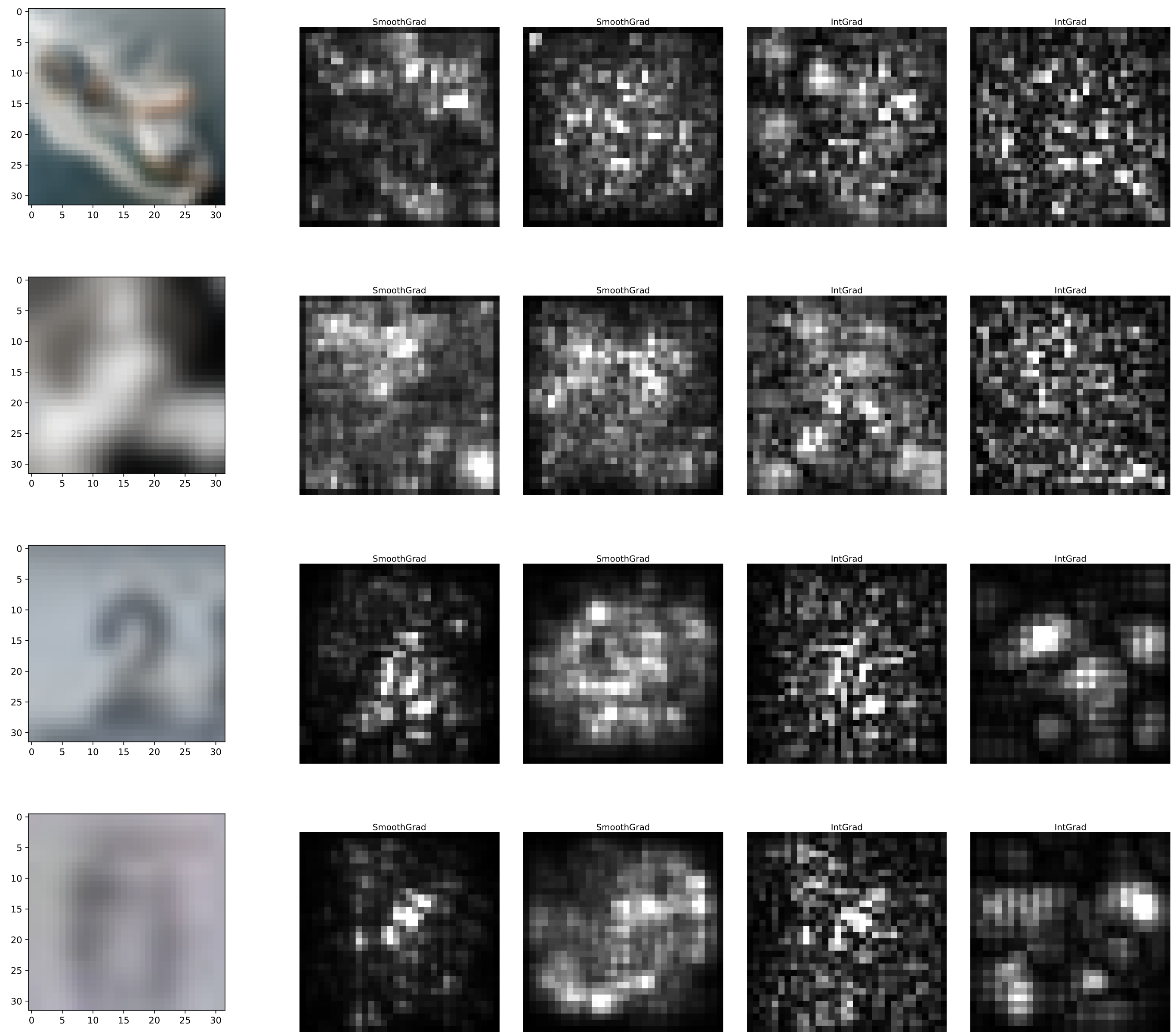

Figure 4. **Saliency maps for DUP and UVC models on the SVHN/CIFAR-10 disagreement task.** The plot shows two images from the blurred CIFAR-10 dataset and two images from the blurred SVHN dataset. The second column is SmoothGrad applied to the UVC model, and the third SmoothGrad applied to the DUP model. The third and fourth columns show IntGrad applied to the DUP and UVC models. We observe that the DUP and UVC models appear to be paying attention to different features of the dataset.

Model Type	T_{test} **AUC**	**Majority**	**Median**	**Majority**$= 1$	**Median**$= 1$	**Referable**
Disagree Soft Targets	76.3%	79.0%	78.7%	81.6%	79.0%	84.7%
Disagree-P	**78.1%**	**81.0%**	80.8%	**84.6%**	**81.9%**	**86.2%**
Disagree-PC	**78.1%**	80.9%	**80.9%**	84.5%	81.8%	**86.2%**

Table 6. Using soft targets for disagreement prediction does not help in performance (AUC). Holdout AUC column corresponds to Disagreement Prediction Performance in Table 3, other columns refer to Table 4 in main text.

Task	Model Type	Performance (AUC)
Variance Prediction	Variance-E2E-2H	72.7%
Variance Prediction	Variance-LR	72.4%
Disagreement Prediction	Disagree-LR	75.9%

Table 7. Additional results from table 3.

Task		Model Type	Performance (AUC)
Entropy Prediction	UVC	Histogram-PC	75.5%
Entropy Prediction	DUP	Disagree-P	**77.2**%

Table 8. DUP and UVC models trained with entropy as a target function. Again, we see that the DUP model outperforms the UVC model.

The *C* suffix of all of these models corresponds to calibration on the logits. Following the findings of (Guo et al., 2017), we apply temperature scaling on the logits: we set the predictions of the model to be $f(\mathcal{F}/T)$ where f is the softmax function, applied pointwise, and $\mathcal{F}$ are the logits. We initialize T to 1, and then split e.g. $T_{train}^{(disagree)}$ into a $T_{train}^{'(disagree)}$ and a $T_{valid}^{'(disagree)}$, with 10% of the data in the validation set. We train as normal on $T_{train}^{'(disagree)}$, with T fixed at 1, and then train on $T_{valid}^{'(disagree)}$, by *only* varying the temperature T, and holding all other parameters fixed.

The use of Prelogit embeddings and Calibration gives the strongest performing baseline UVC and DUPs: Histogram-PC, Variance-PRC and Disagree-PC. For the Variance DUP, an additional regularization term is added to the loss by having a separate regressing on the raw variance value.

Additional Model: Variance-E2E We tried a variant of Variance-E2E, Variance-E2E-2H, which has one head for predicting variance and the other for classifying, to enable usage of all the data. We then evaluate the variance head on T_{test}, but in fact noticed a small drop in performance, Table 7.

Do we need the Prelogit embeddings? We tried seeing if we could match performance by training on pretrained classifier logits instead of the prelogit embeddings. Despite controlling for parameter difference by experimenting with more hidden layers, we found we were unable to match performance from the prelogit layer, Table 7, compare to Table 3. This demonstrates that some information is lost between the prelogit and logit layers.

E. Additional Results: Entropy, Finite Sample Behavior and Convergence Analysis

We performed additional experiments to further understand the properties of DUP and UVC models. For these experiments, we compare a representative DUP model, *Disagree-P*, to a representative UVC model, *Histogram-PC*.

Theorem 1 states that DUP offers benefits over UVC for *concave* target uncertainty functions. This is a natural property for measures of spread, simply stating that the measure of spread increases with averaging (probability distributions). In the main text, we concentrate on two such specific uncertainty functions, U_{var} and $U_{disagree}$, which are particularly suited to the domain. However, other standard uncertainty functions, such as entropy, are also concave. We test the performance of DUP (*Disagree-P*) and UVC (*Histogram-PC*) with $U_{entropy}$ as the target function.

The results are shown in Table 8, where we again see that DUP outperforms UVC.

We also study how model performance is impacted by different training set sizes (similar to the analysis in (Chen et al., 2018)). We subsample different amounts of the original training set $T_{train}^{(disagree)}$, and train DUP and UVC models on this subset. The results over 5 repeats of different subsamples and optimization runs are shown in Figure 5.

We see that the performance gap between DUP and UVC is robust to train data size differences. Additionally, when $\geq 30\%$ of the training data is used, DUP and UVC performance remains relatively constant. This supports carrying over the results of Theorem 1) and the full joint distribution $f(o, \mathbf{y})$ to the finite data setting.

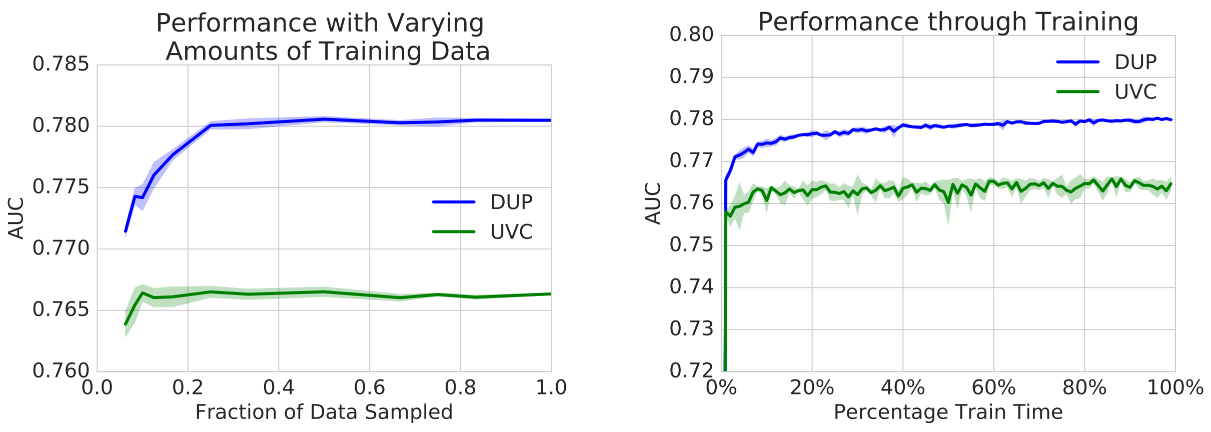

Figure 5. **DUP and UVC performance during training and when varying train data size.** We study DUP (Disagree-P) and UVC (Histogram-PC) performance for varying amounts of training data. We find that the gap in performance is robust to variations in dataset size. For more than 30% of the data, performance of DUP and UVC remains relatively constant, supporting the applicability of Theorem 1 in the finite data setting. The right plot looks at performance through training, with the gap appearing rapidly early in training, and slowly widening.

We also study convergence of DUP and UVC models. We find that the performance gap between DUP and UVC manifests very early in training (Figure 5, right plot), and continues to gradually widen through training.

F. Background on the Wasserstein Distance

Given two probability distributions f, g, and letting $\Pi(f, g)$ be all product probability distributions with marginals f, g, the Wasserstein distance between p, q is

$$||f - g||_w = \inf_{\pi \in \Pi(f,g)} \mathbb{E}_{(r,t)\sim\pi} [d(r, t)]$$

where $d(,)$ is some metric. This distance has connections to optimal transport, and corresponds to the minimum cost (with respect to $d(,)$) of moving the mass in distribution f so that it is matches the mass in distribution g. We can represent the amount of mass to move from r to t with $\pi(r, t)$. To be consistent with the mass at the start, $f(r)$, and the mass at the end $g(t)$ we must have that $\int_{t'} \pi(r, t') = f(r)$ and $\int_{r'} \pi(r', t) = g(t)$.

The result in the main text follows from the following theorem:

Theorem 3. *If f, g are (discrete) probability distributions and g is a point mass distribution at t_0, then $\pi \in \Pi(f, g)$ is uniquely defined as:*

$$\pi(r, t) = \begin{cases} 0 & \text{if } t \neq t_0 \\ f(r) & \text{if } t = t_0 \end{cases}$$

Proof. The proof is direct: for $t \neq t_0$, we must have $\int_{r'} \pi(r', t) = g(t) = 0$, and so $\int_{t'} \pi(r, t') = \pi(r, t_0) = f(r)$. □

We consider three different distances $d(,)$:

1 *Absolute Value* $d(r, t) = |r - t|$. This follows an interpretation in which the grades are equally spaced, so that all successive grade differences have the same weight.

2 *2-Wasserstein Distance* $d(r, t) = (r - t)^2$, and, to make into a metric

$$||f - g||_w = \left(\mathbb{E}_{(r,t)\sim\pi} [d(r, t)]\right)^{1/2}$$

This adds a higher penalty for larger grade differences.

3 *Binary Disagreement* We set $d(r, t) = 0$ if $r = t$ and 1 otherwise.