UNCERTAINTY ESTIMATION FOR DEEP MEDICAL IMAGE SEGMENTATION

DATA INNOVATION LAB

AUTHORS: Mohammad Alamleh, Guillem Brasó, Yu Chen, Sven Elflein, Martin Hermann

MENTORS: Dr. Ghazal Ghazaei, Dr. Alexander Urich (CARL ZEISS AG)

CO-MENTOR: Dr. Tobias Köppl (DEPARTMENT OF MATHEMATICS)

PROJECT LEAD: Dr. Ricardo Acevedo Cabra (DEPARTMENT OF MATHEMATICS)

SUPERVISOR: Prof. Dr. Massimo Fornasier (DEPARTMENT OF MATHEMATICS)

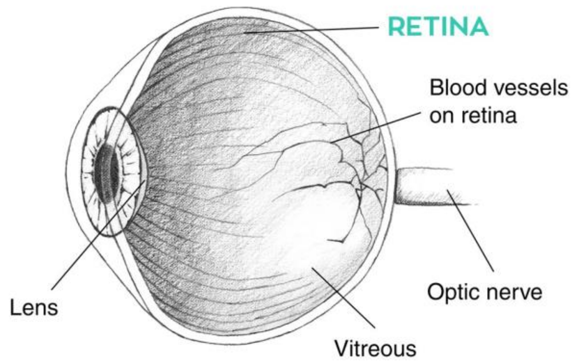
OUTLINE

Introduction
Semantic Segmentation
Uncertainty Estimation
Conclusion

INTRODUCTION

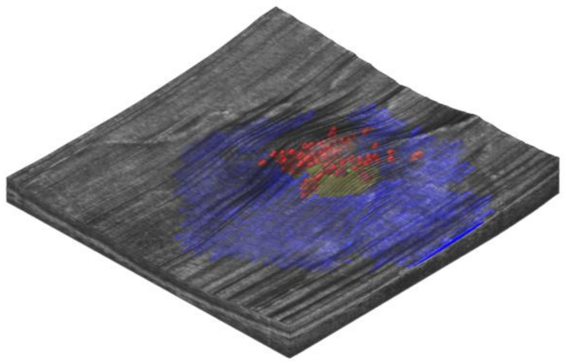
Background Information
Dataset - RETOUCH
Semantic Segmentation
Uncertainty Estimation

Background Information



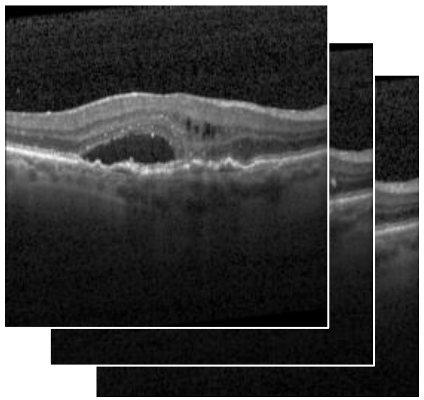
Anatomy of the Eye

OCT Imaging



3D RETINA with fluids:
IRF, SRF, PED

Gives



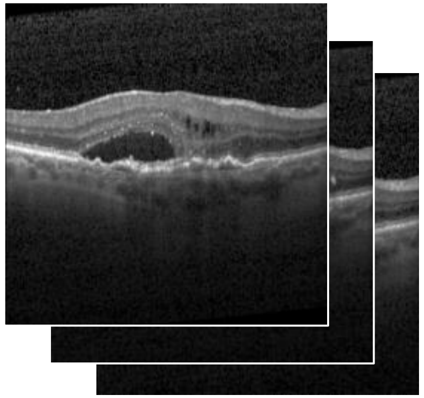
A volume of B-scans

- How do doctors **detect the fluids** from B-scans?

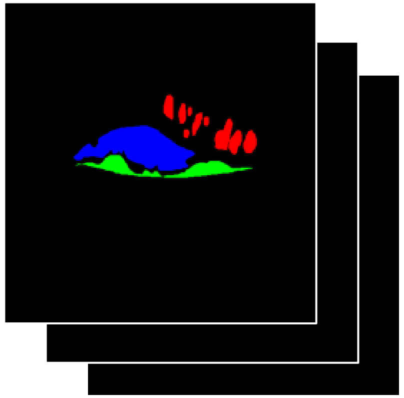
Dataset - RETOUCH

Dataset

RETOUCH:
- 70 Volumes
- 3 Devices
- 1% Fluid



Manual Annotation



Intraretinal Fluid (IRF)

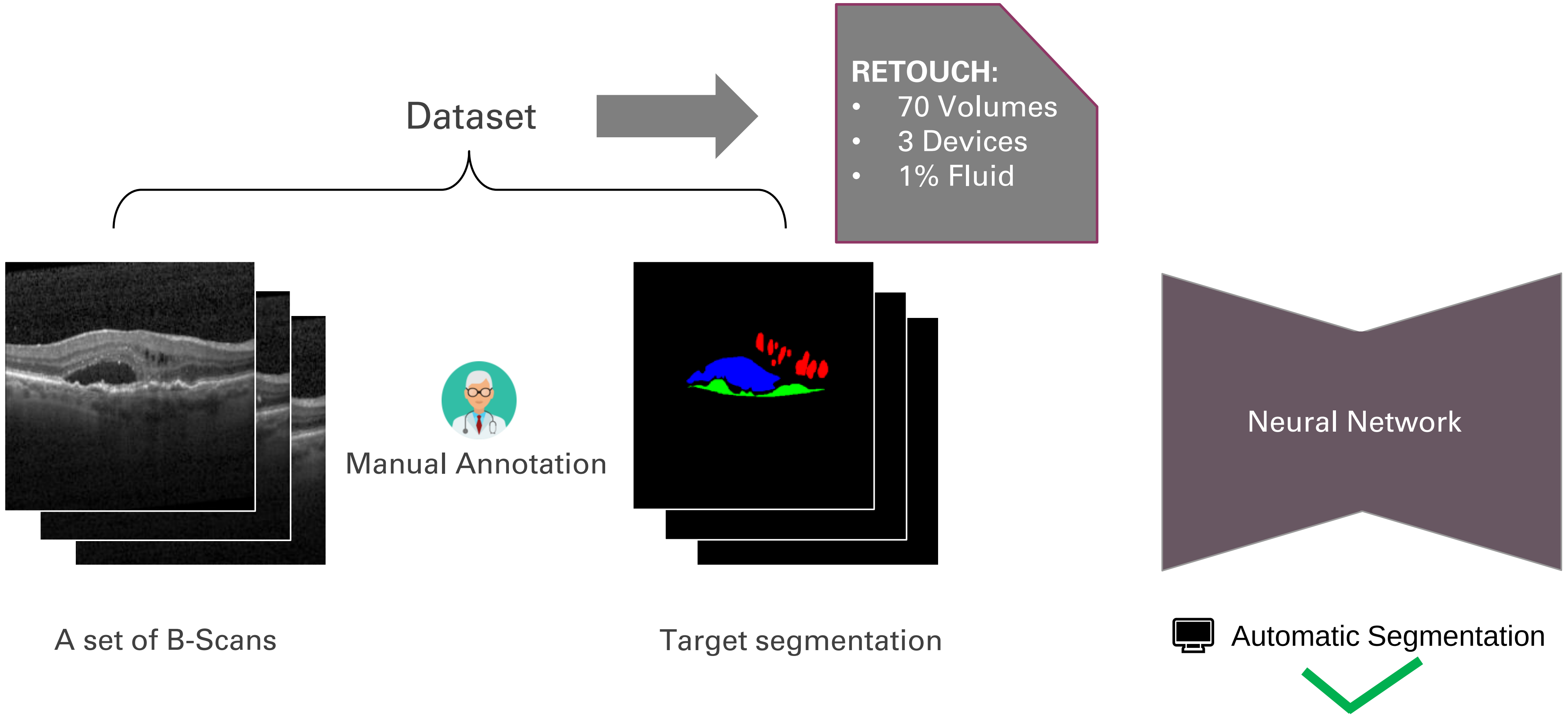
Subretinal Fluid (SRF)

Pigment Epithelium Detachments (PED)

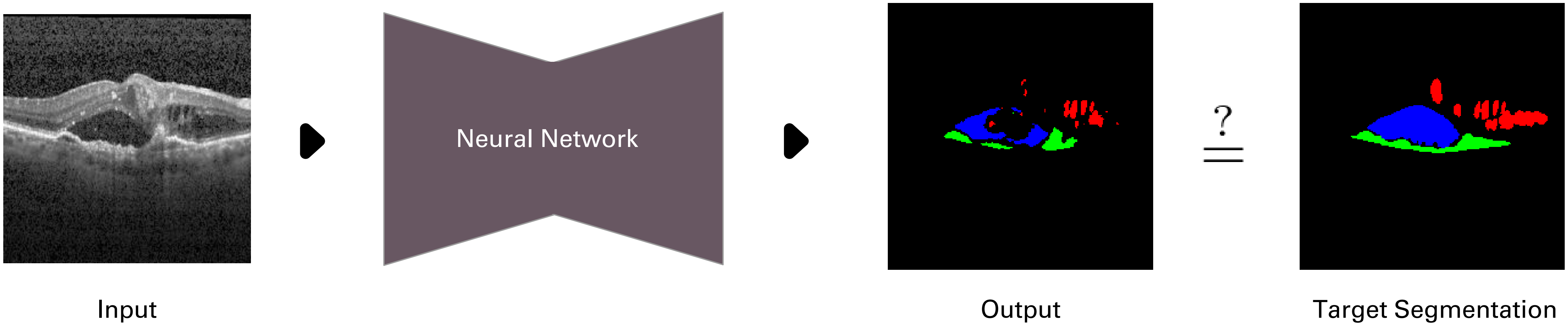
A set of B-Scans

Target segmentation

Dataset - RETOUCH



Segmentation using Deep Learning



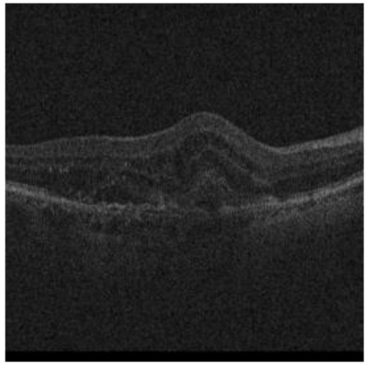
- **Training** = Update the weights of NN to **minimize the difference** between **Output** and **Target Segmentation**

Prediction using Trained Neural Network

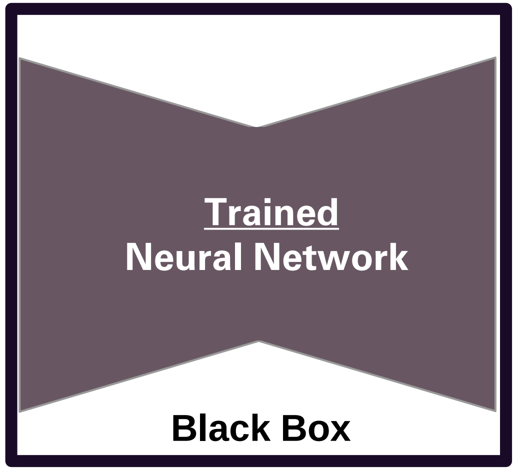
New Input
(NOT in Training set)

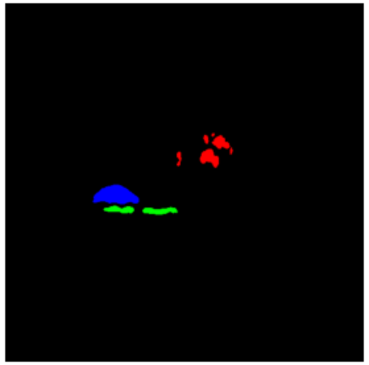
Prediction

Can You Trust the Prediction of a NN?



New Input
(NOT in Training set)



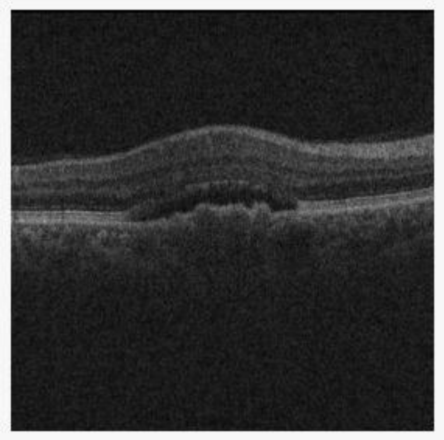


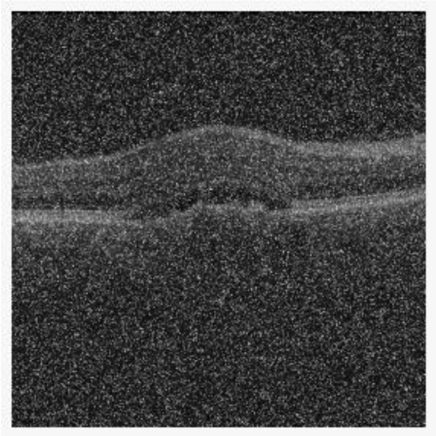
Prediction

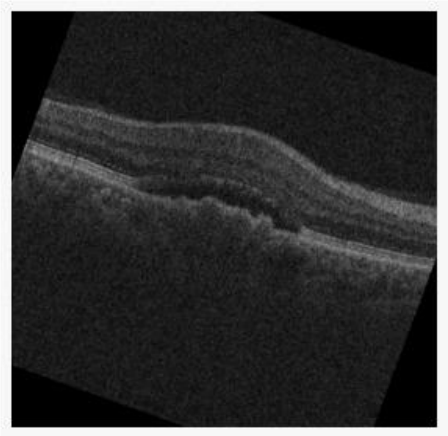
Problem: How certain/uncertain the **Trained NN** is about its prediction?

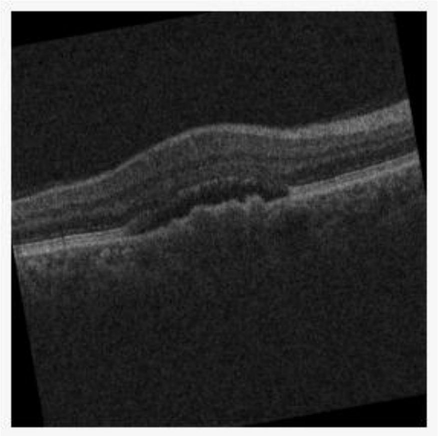
- Need **Uncertainty Estimation**!

Where does Uncertainty come from (1)?









B-Scan

Noise

Geometric Transformation

- **Aleatoric Uncertainty** – from data generation process (Noise and Geometric Transformation)

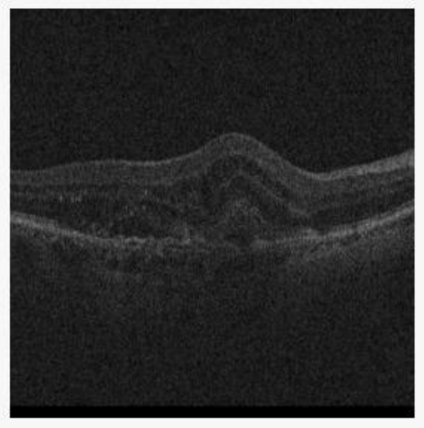
Where does Uncertainty come from (2)?

"Unseen" Input

Prediction

- **Epistemic Uncertainty** – from the trained NN that **has not seen** all samples during training

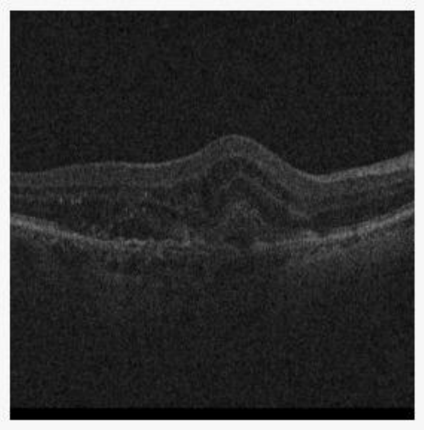
Pipeline



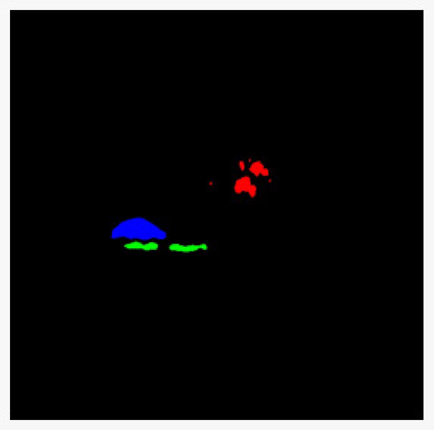
B-Scans

- RETOUCH Dataset

Pipeline



NN
Loss



IRF

SRF

PED

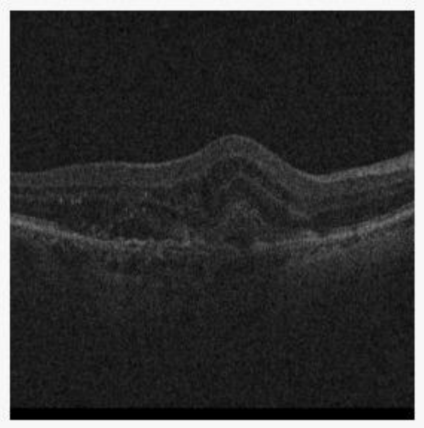
B-Scans

- RETOUCH Dataset

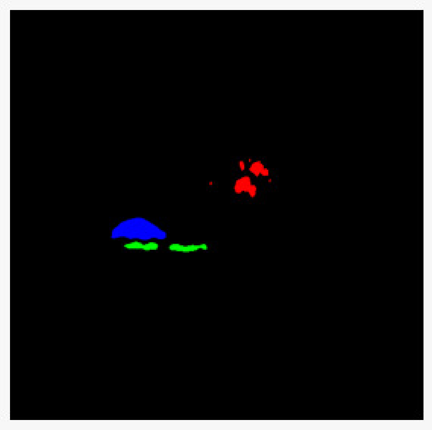
Segmentation

Evaluated by
- Dice Score

Pipeline

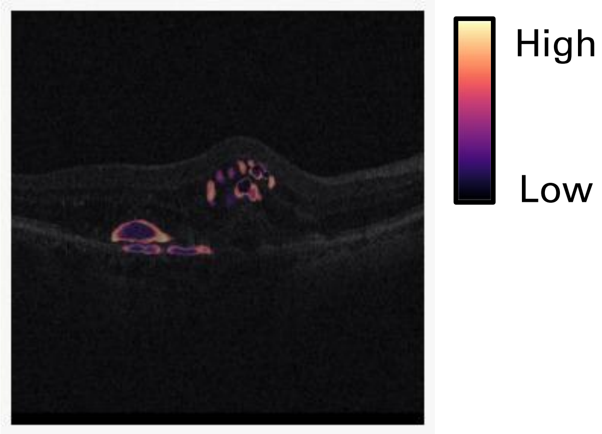


NN
Loss



IRF
SRF
PED

Test-time Augmentation
MC Dropout
Ensemble
MC Dropout + Ensemble
Loss Attenuation
Direct Error Prediction



B-Scans

- RETOUCH Dataset

Segmentation

Evaluated by
- Dice Score

Uncertainty Estimation

- Described by
  - Pixelwise Level
  - Imagewise Level
- Evaluated by
  - Correlation
  - Calibration

SEMANTIC SEGMENTATION

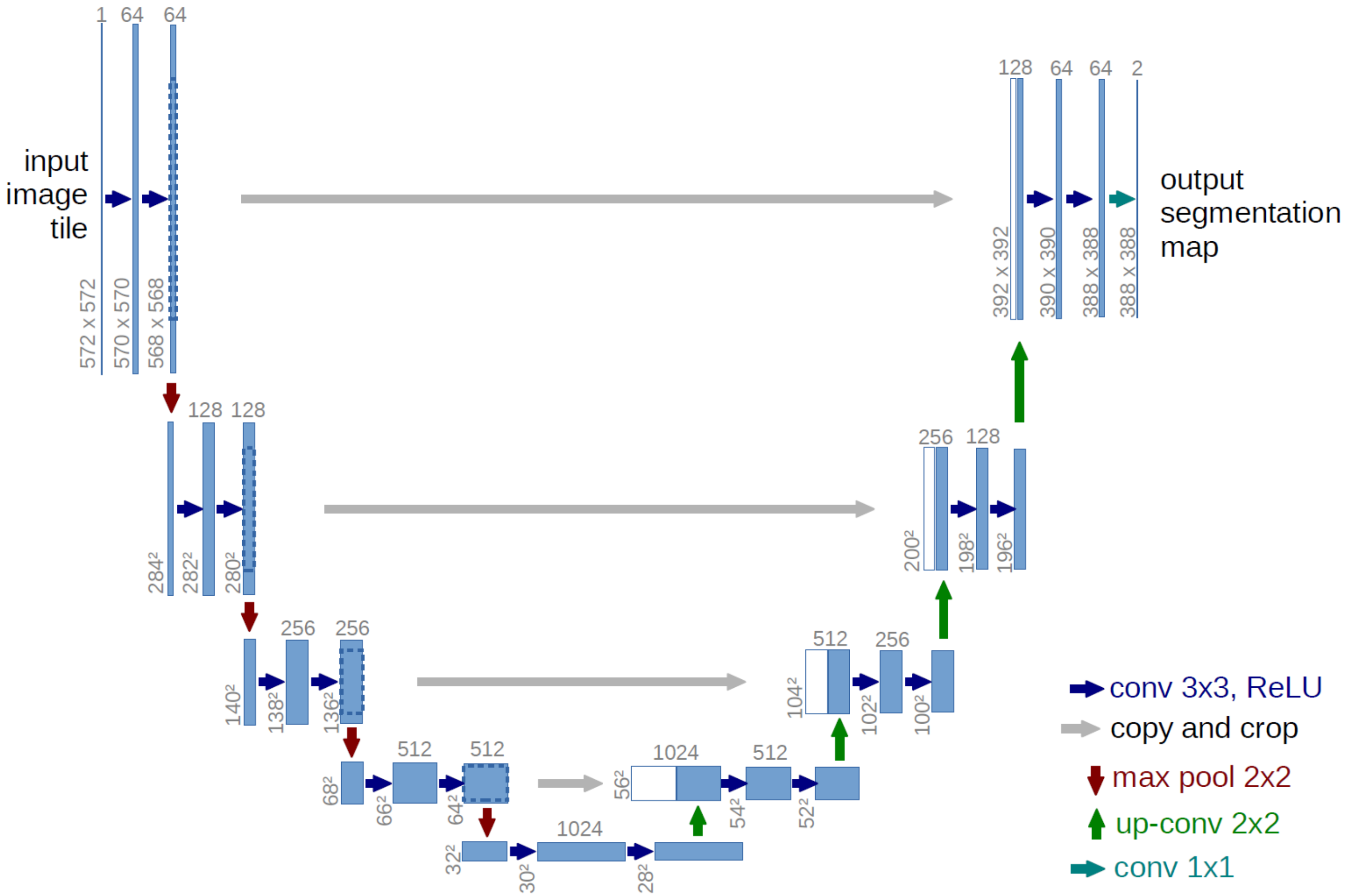
U-Net Architecture

Loss Functions

Evaluation Metrics

Results

U-Net Architecture



- Standard baseline for biomedical segmentation tasks [1]
- Fully convolutional
- Outputs 4 scores per input pixel at input resolution

Modifications:

- Batch-Normalization layers
- Dropout after convolutional blocks

Loss Functions

- **Cross Entropy Loss:**

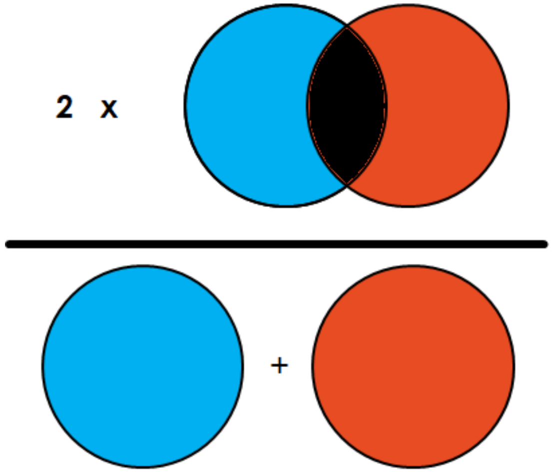
$$\underbrace{-\frac{1}{d \times d} \sum_{i=1}^{d} \sum_{j=1}^{d}}_{\text{Average over all pixels}} \underbrace{\sum_{c=1}^{C}}_{\text{Sum over classes}} \underbrace{w_c}_{\text{Per-class weight}} \underbrace{y_{i,j,c}}_{\text{True prob. that pixel (i, j) belongs to class c}} \log \Big(\underbrace{\sigma(f_{i,j,c})}_{\text{Predicted prob. that pixel (i, j) belongs to class c}} \Big)$$

The Cross-Entropy loss penalizes predicted pixelwise probability distributions that deviate from the ground truth

Loss Functions

- **Dice Loss:**

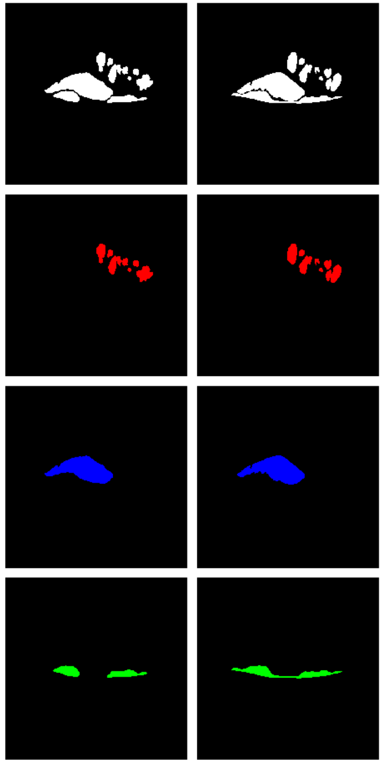
$$1 - \frac{2\sum_i \sum_j (A_{i,j} * B_{i,j})}{\sum_i \sum_j (A_{i,j} + B_{i,j}) + \epsilon}$$



The Dice Loss is a continuous relaxation of the Dice Score, which measures the *similarity* of two sets

Loss Functions

- **Dice Loss:**



Output Target

Dice Loss Background.

Dice Loss IRF

Dice Loss SRF

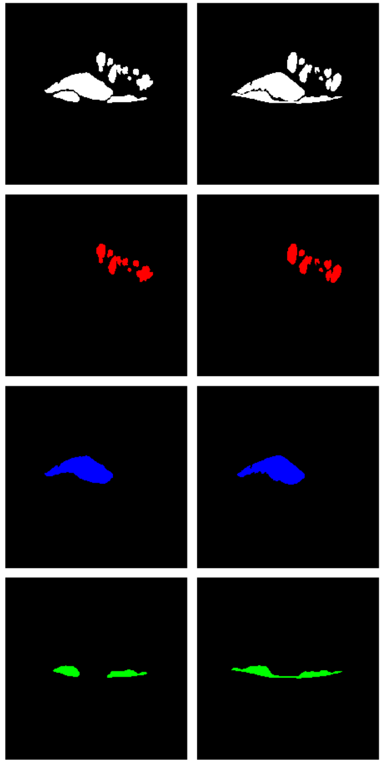
Dice Loss PED

$$AvgDiceLoss = 1/4 \left[\text{Dice Loss Background} + \text{Dice Loss IRF} + \text{Dice Loss SRF} + \text{Dice Loss PED} \right]$$

We compute the average dice loss by averaging the dice loss over each fluid's output mask

Loss Functions

- **Dice Loss:**



Output Target

~~Dice Loss Background.~~

Dice Loss IRF

Dice Loss SRF

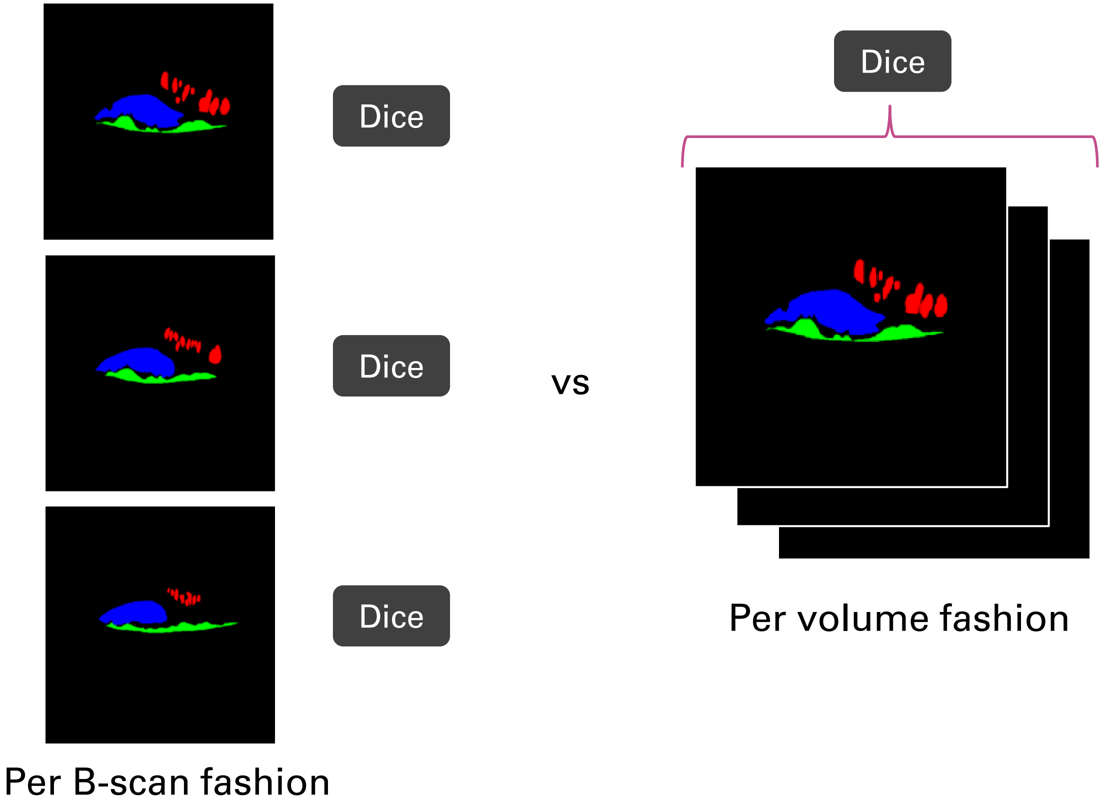
Dice Loss PED

$$AvgDiceLoss = 1/3 \left[\cancel{\text{Dice Loss Background}} + \text{Dice Loss IRF} + \text{Dice Loss SRF} + \text{Dice Loss PED} \right]$$

We experiment with not computing the loss over the background class

Evaluation Metrics

- The *Dice Score* is computed as 1 – Dice Loss over discretized output probabilities
- Scores are computed over entire 3D OCT volumes (instead of individual scans)
- Volumes with no fluid are skipped for consistency with the RETOUCH evaluation protocol



Results

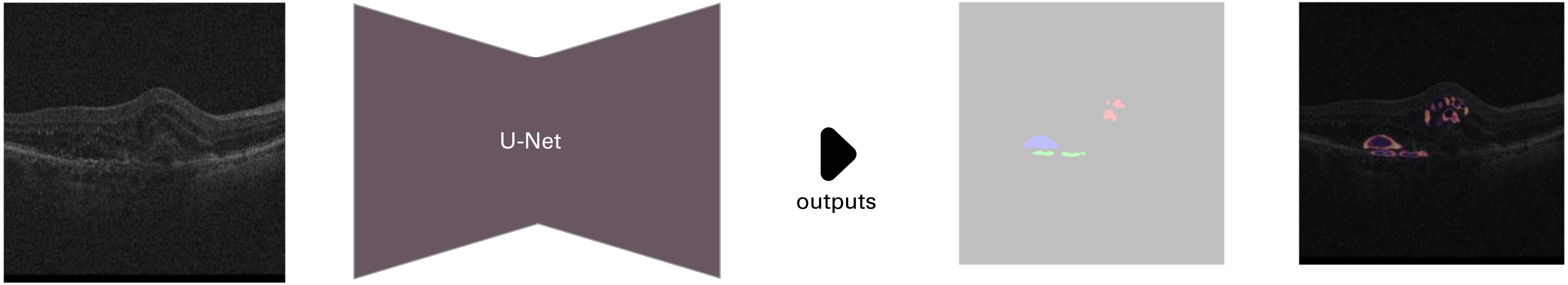
Setting \ Metrics		Dice Score			
		Mean	PED	IRF	SRF
CE Loss	$w_c = 1\ \forall c$	0.580	0.603	0.537	0.600
	Weighted by Inv. Frequencies	0.524	0.523	0.494	0.556
Dice Loss	$w_c = 1\ \forall c$	0.644	0.652	0.556	**0.725**
	$w_{background} = 0$	0.644	0.635	0.640	0.658
Dice + CE Loss	$w_c = 1\ \forall c$	0.646	0.604	**0.653**	0.680
	$w_{background} = 0$	0.660	0.639	0.648	0.695
Helios team	U-Net + heavy engineering	**0.680**	**0.730**	0.610	0.700

- Dice Loss clearly outperforms Cross-Entropy, and a combination of both works best
- Not computing the dice loss over background pixels improves performance
- Our model performs comparably to the Helios Team, without heavy engineering

UNCERTAINTY ESTIMATION

- Sample-based Methods
  - Imagewise Uncertainty
- Loss Attenuation
- Direct Error Prediction
- Results

Uncertainty Estimation



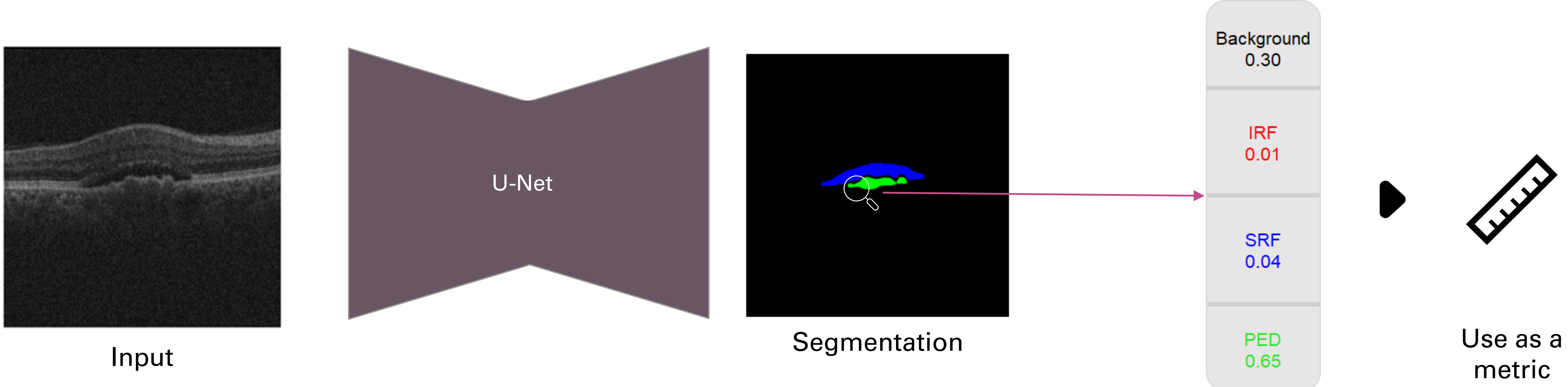
Input

(1) Segmentation

(2) Uncertainty

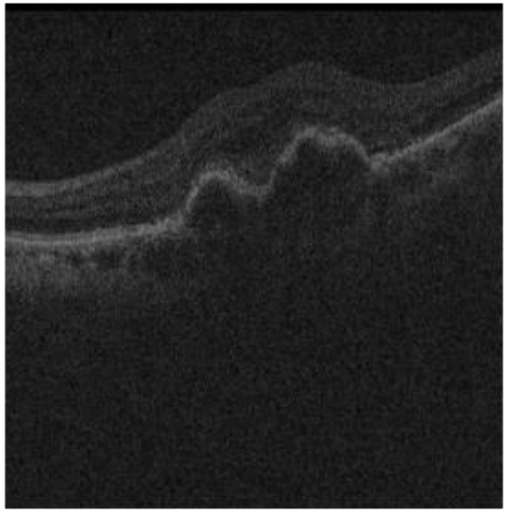
Baseline

- From the segmentation, we already have a (softmaxed) distribution among the classes
- It is tempting to interpret this as a form of uncertainty

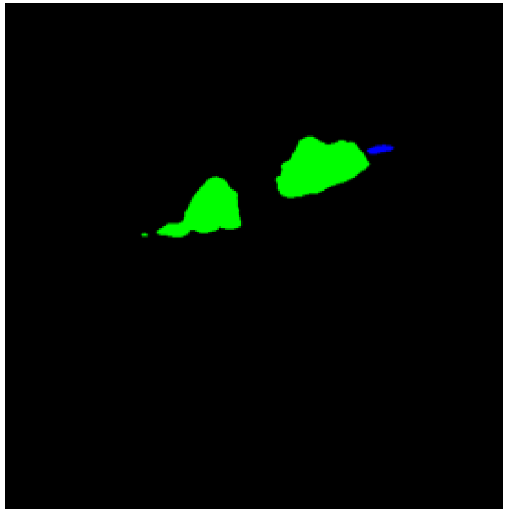


Baseline

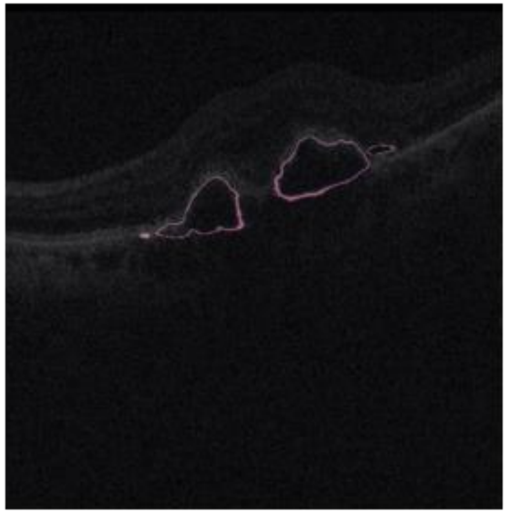
- **<u>However</u>**, it turns out that this does not work very well
- In most cases the "uncertainty" from this method just traces the edges of the prediction



Input



Segmentation

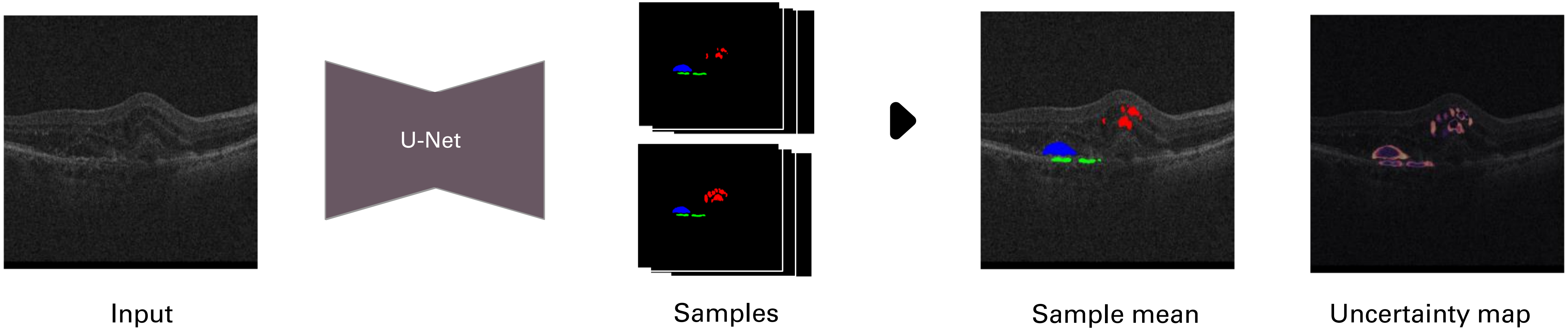


Uncertainty

Sample-based Methods

- Usually, uncertainty is interpreted as some property of some probability distribution
  - Epistemic: Distribution over the network weights
  - Aleatoric: Distribution over the data

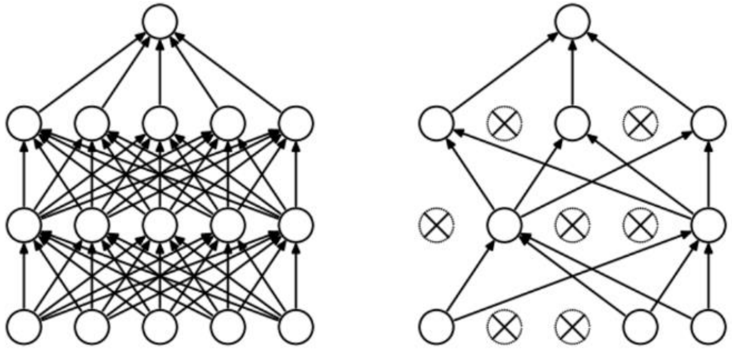
- Standard approach for this sort of problem: Monte Carlo Integration



Sample-based Methods

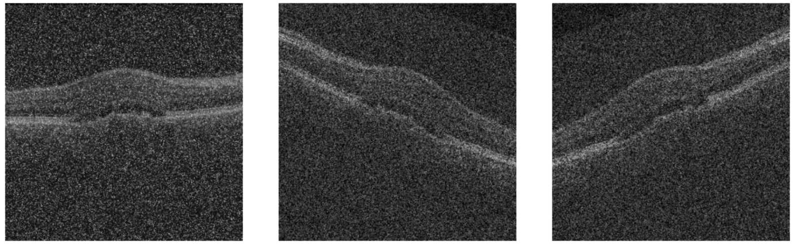
Different ways to generate samples:

- Monte Carlo Dropout [2]:
  - For every forward pass, disable some neurons at random
  - Approximates a probability distribution over the learned weights



(a) Standard Neural Net (b) After applying dropout.

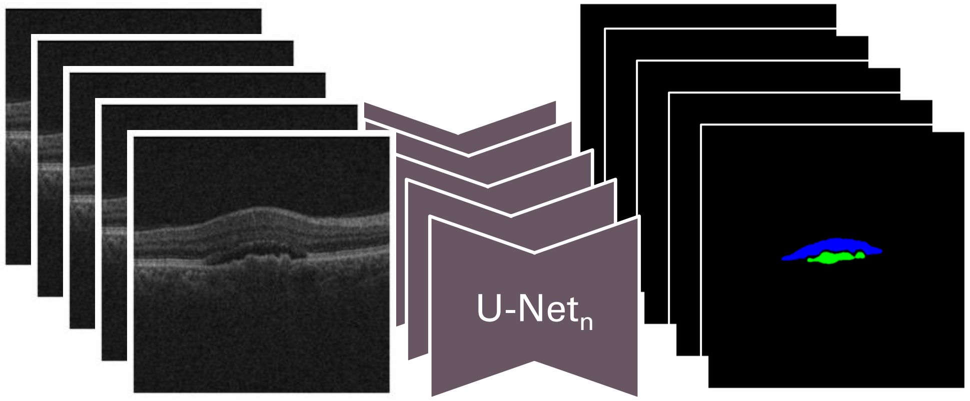
- Test Time Augmentation [3]:
  - Apply different rotations and noise to the image before classifying
  - Tries to mirror the variation found in the dataset



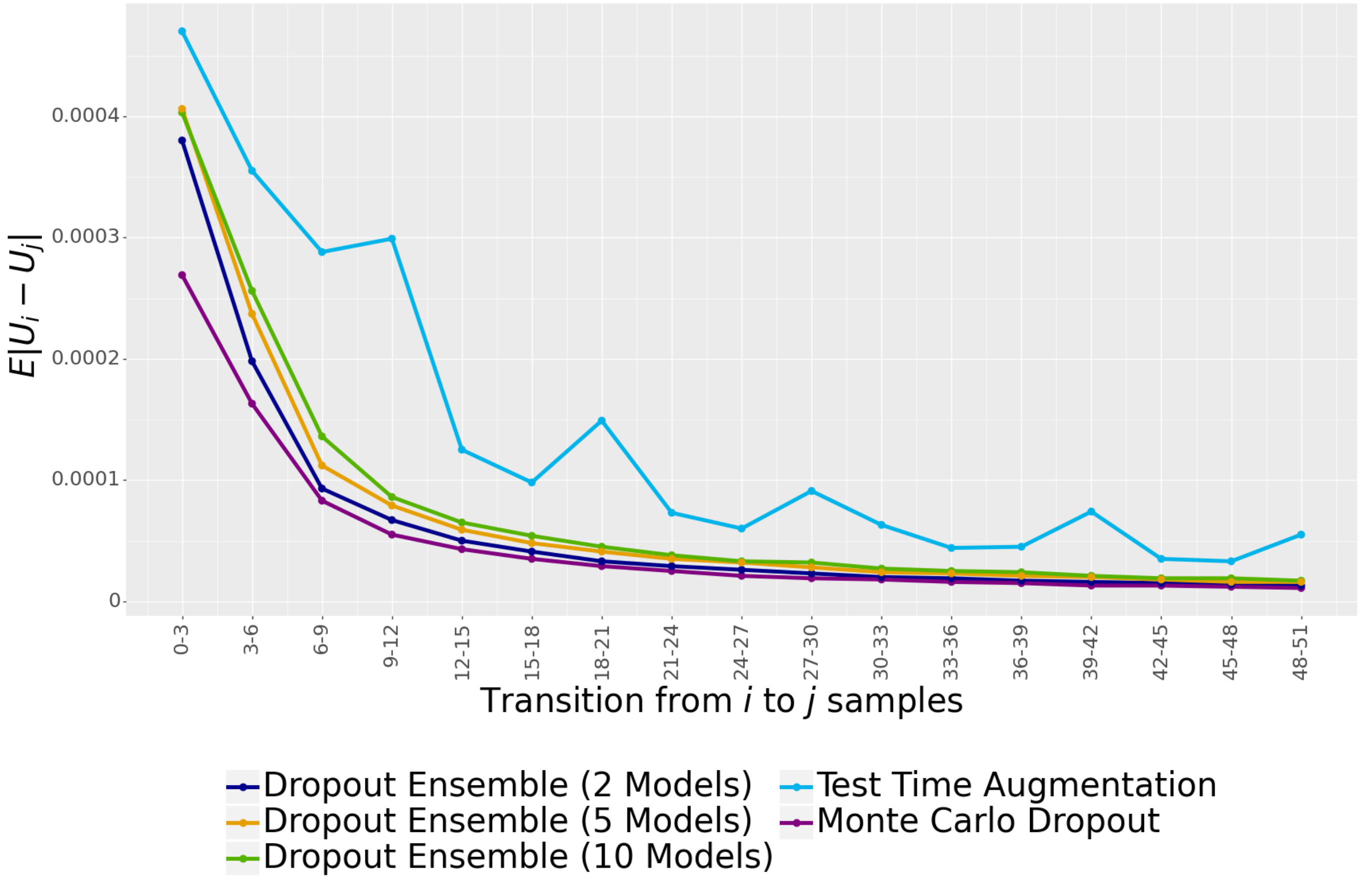
Sample-based Methods

Different ways to generate samples:

- Deep Ensembles [4]:
  - Train multiple networks, each with different initializations
  - For each sample, use the output of a different network
- Dropout Ensembles [5]:
  - Combine Ensembles and Dropout
  - The different methods approximate different uncertainties



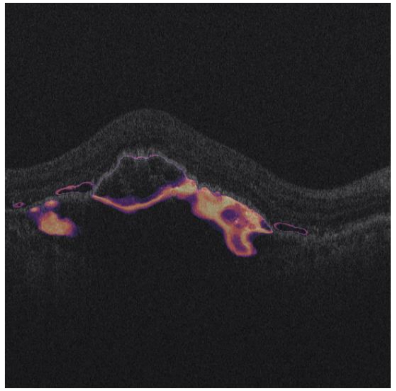
How many samples to choose?



- After ~30 samples, not much additional information from adding additional ones
- We make a cutoff there, as model evaluation runtime is linear
- Less stable behavior for Test Time Augmentation, but similar in principle

Imagewise Uncertainty

- Aggregate uncertainty maps into single number
- Helps with decision-making in practical situations
- Refer images with high uncertainty to human medical experts
- Two measures, both rely on agreement between samples:
  - iDais: $1 - \frac{2}{N(N-1)} \sum_{i \neq j}^{N} Dice\left(s_i, s_j\right)$
  - iIoU: $1 - \frac{1}{C} \sum_{c=1}^{C} \frac{|(S_1=c)\cap(S_2=c)\cap\cdots\cap(S_N=c)|}{|(S_1=c)\cup(S_2=c)\cup\cdots\cup(S_N=c)|}$

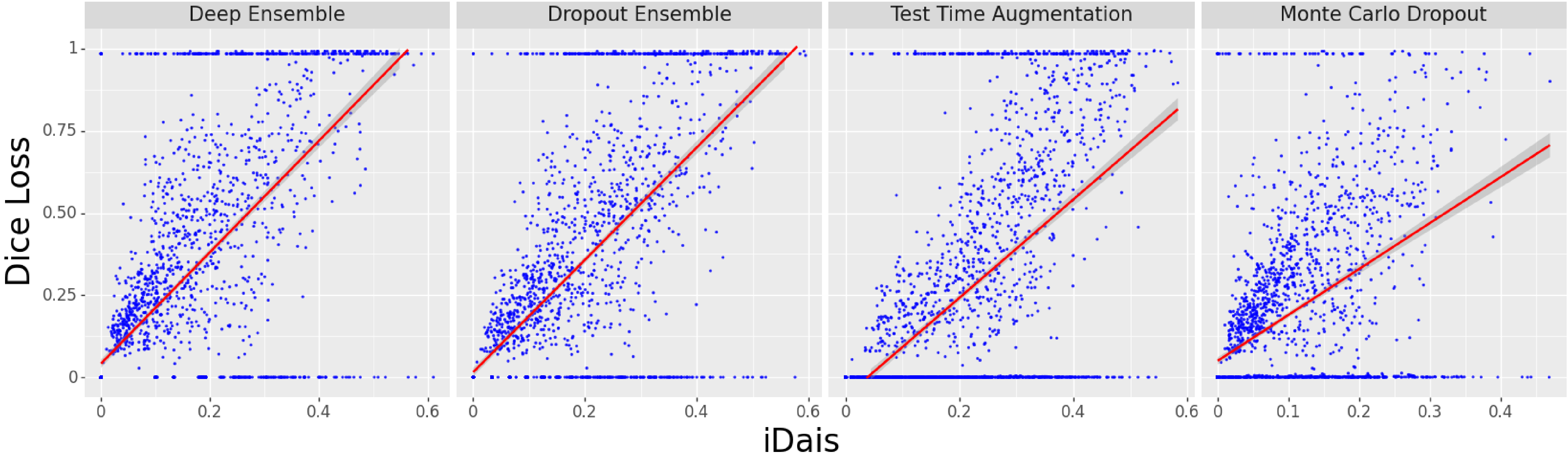


Map to

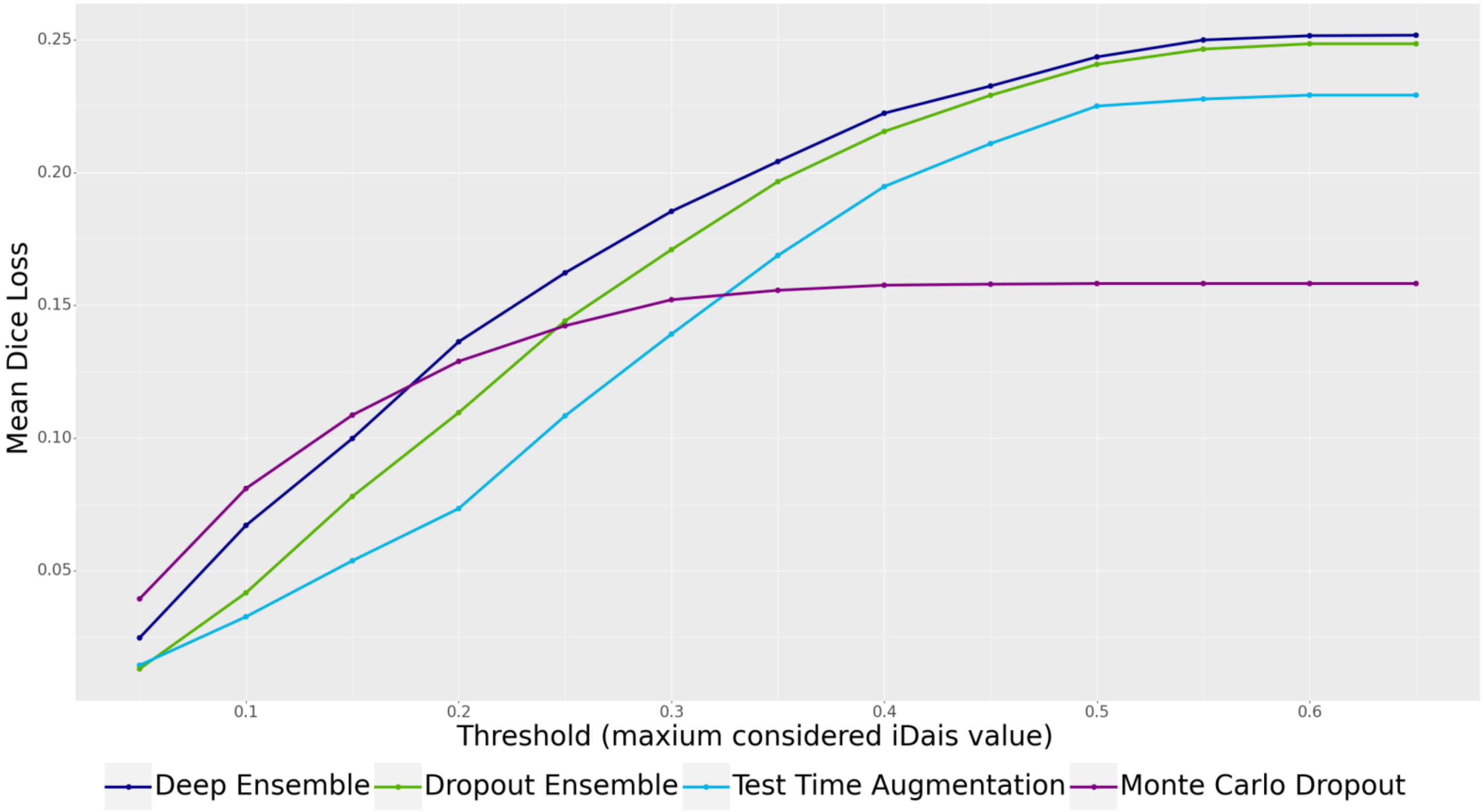
Single number

Imagewise Uncertainty: Performance (iDais)

- Plotting Error (Dice loss) versus Uncertainty (iDais) reveals a linear relationship
- Outliers at top and bottom caused by properties of Dice loss



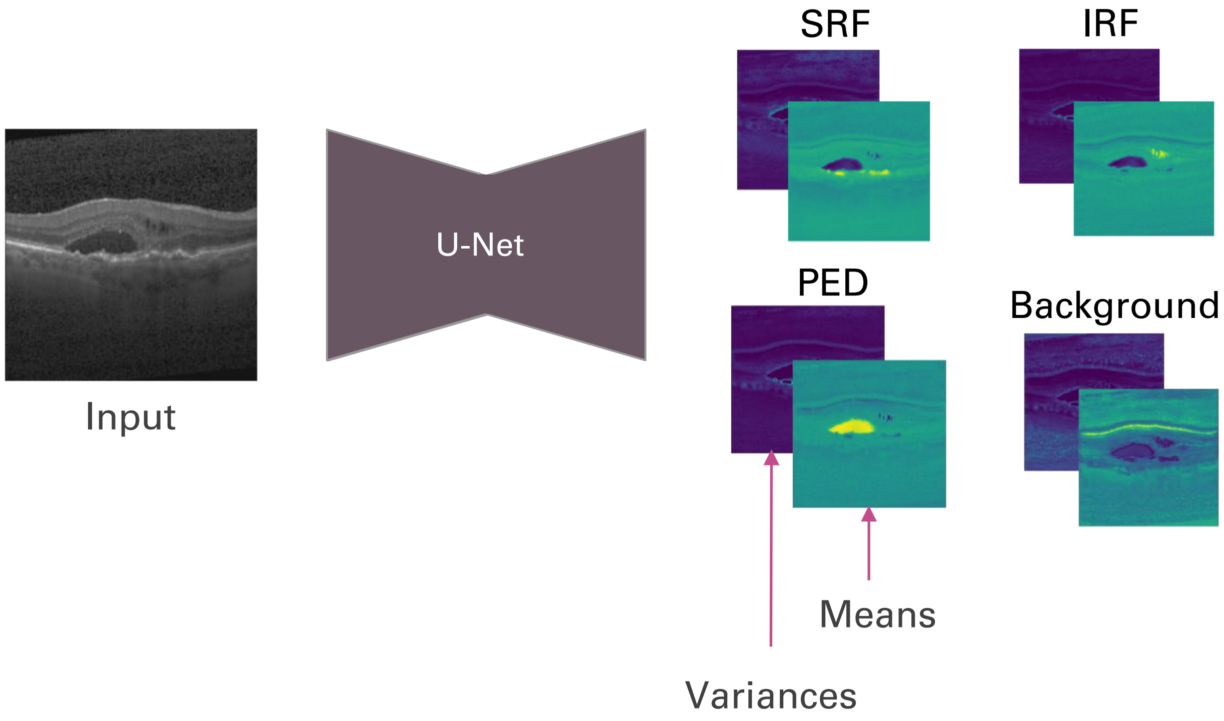
Imagewise Uncertainty: Performance (iDais)



→ Similar behavior for all methods, except Monte Carlo Dropout

Loss Attenuation

- **Sampling-free** aleatoric uncertainty estimation method [6]
- Replace **every** output per pixel and class with two outputs
  - Variance σ^2
  - Mean μ
- Parametererize Normal distribution $N(\mu, \sigma^2)$ per pixel and class (**logit**)

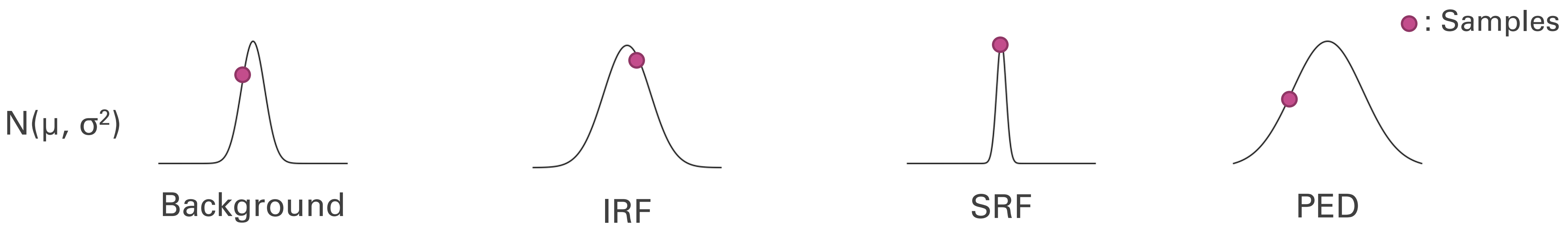


Loss Attenuation

- **Testing**: Single forward pass to obtain logit prediction (means) and uncertainty (variances)

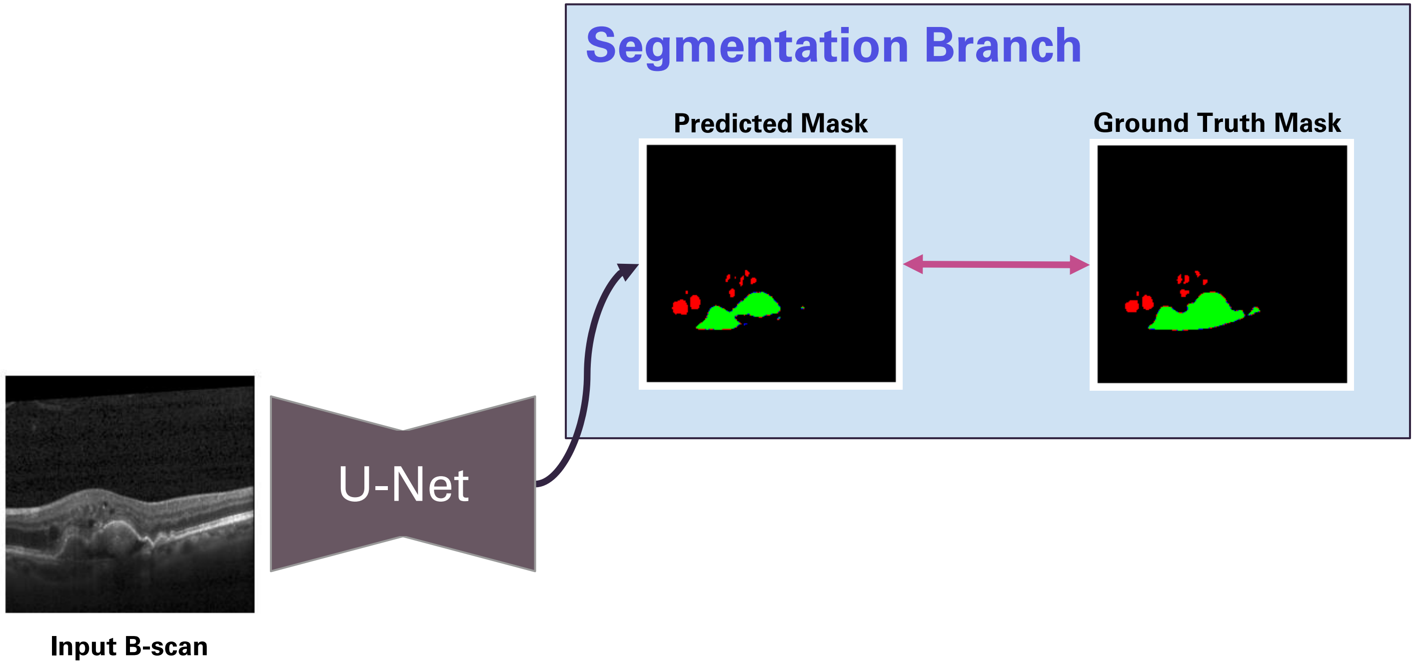
- **Training** Combine Ensembles and Dropout
  - Forward pass to get mean and variance of normal per logit
  - Monte Carlo approximation of the loss by sampling logits from the distributions

→ For **every** pixel we get four Normal distributions over the **class logits**



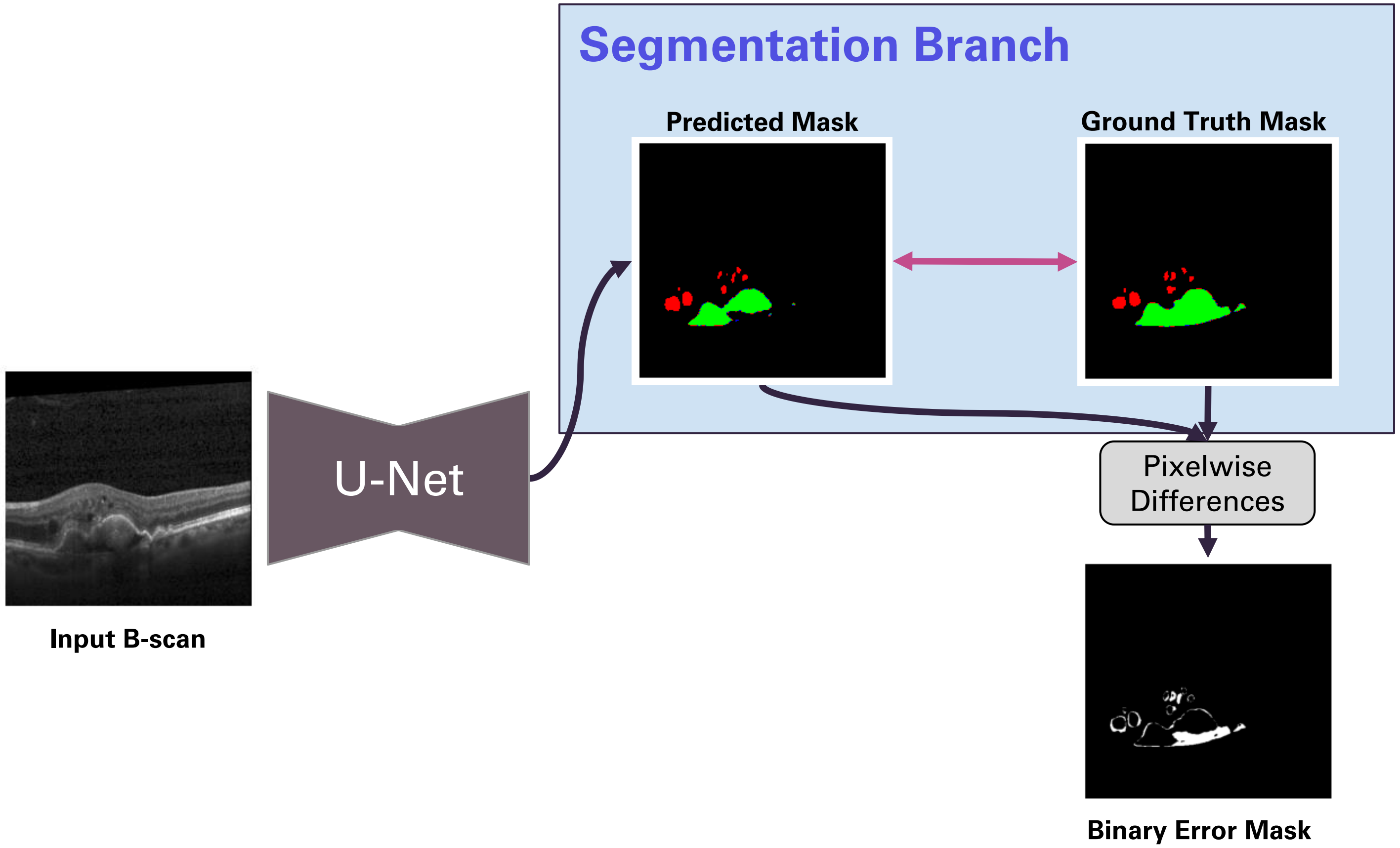
Direct Error Prediction

- **Idea**: Directly predict probability of network being wrong
- Realized by additional output branch trained jointly with the rest of the network
- 'Ground Truth' for the new branch is generated on-the-fly based on segmentation branch



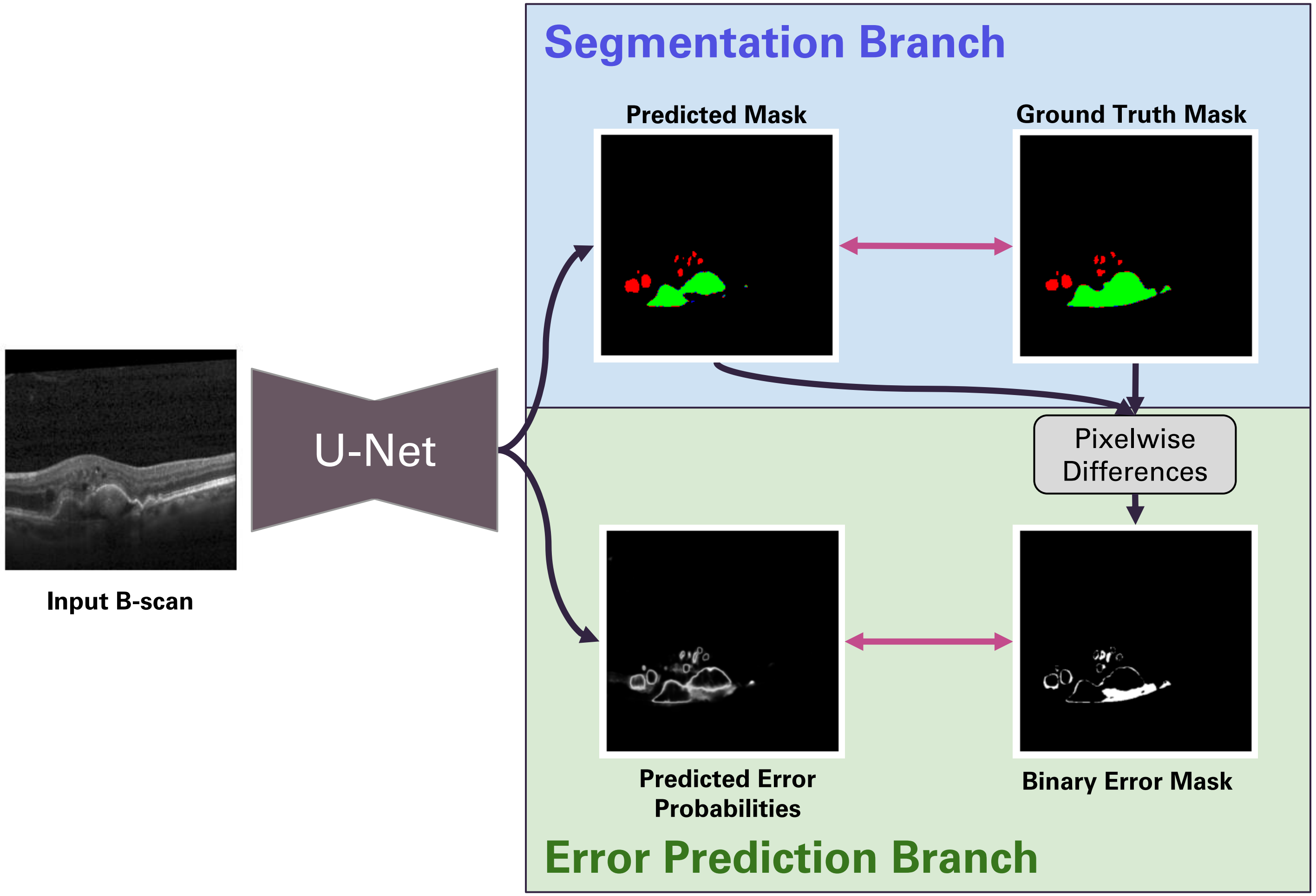
Direct Error Prediction

- **Idea**: Directly predict probability of network being wrong
- Realized by additional output branch trained jointly with the rest of the network
- 'Ground Truth' for the new branch is generated on-the-fly based on segmentation branch

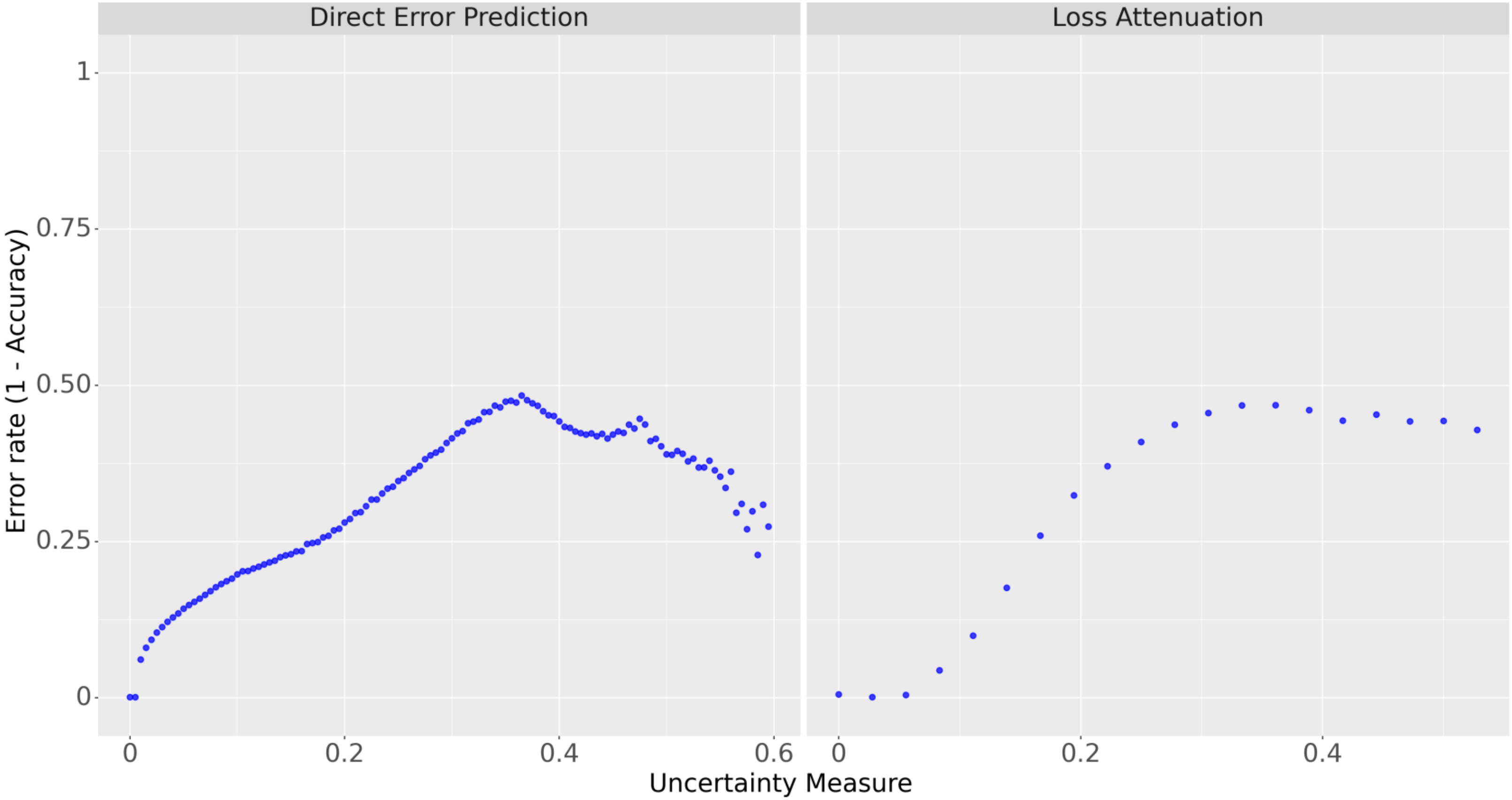


Direct Error Prediction

- **Idea**: Directly predict probability of network being wrong
- Realized by additional output branch trained jointly with the rest of the network
- 'Ground Truth' for the new branch is generated on-the-fly based on segmentation branch



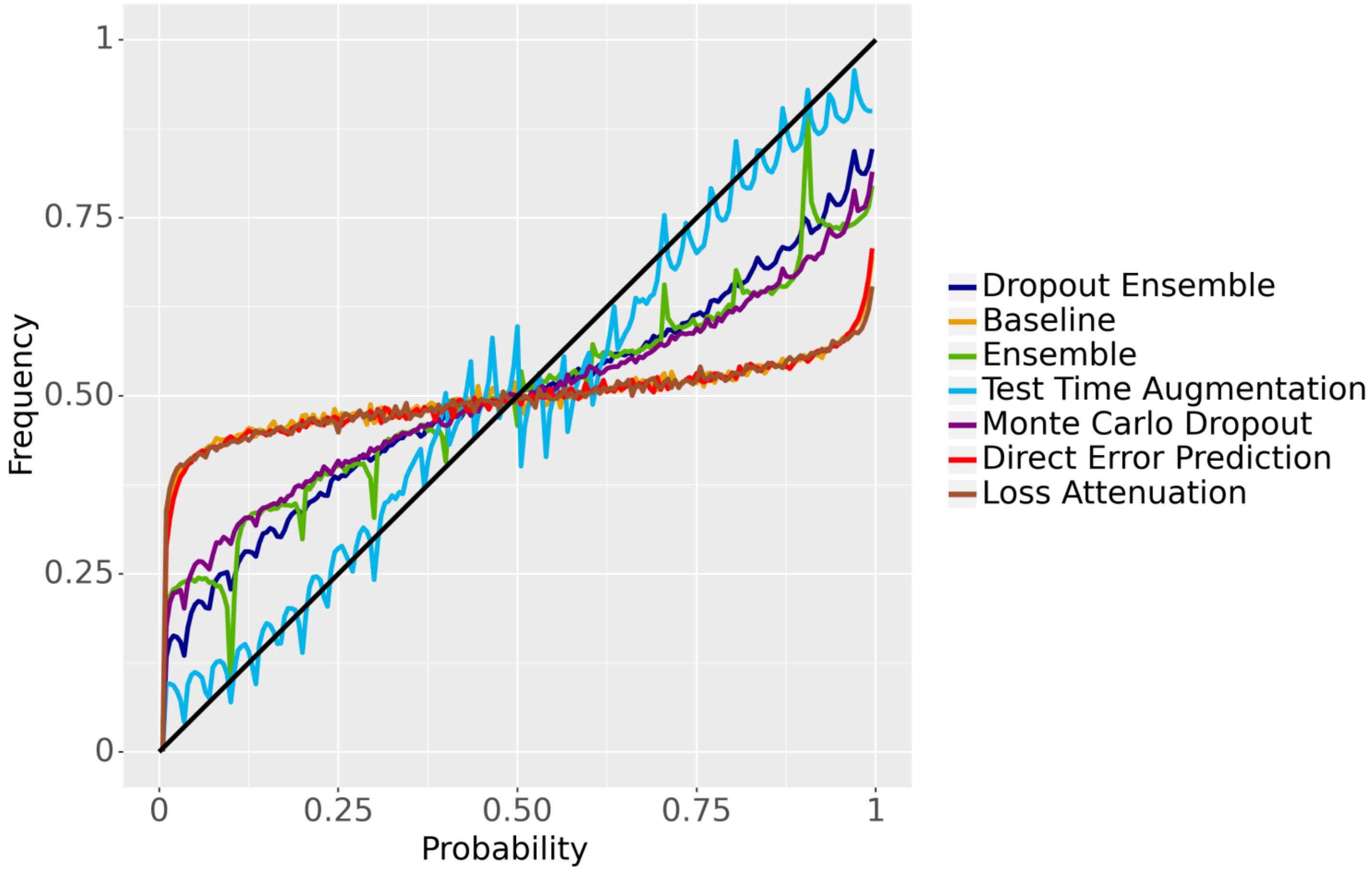
Correlation – Non sampling-based methods



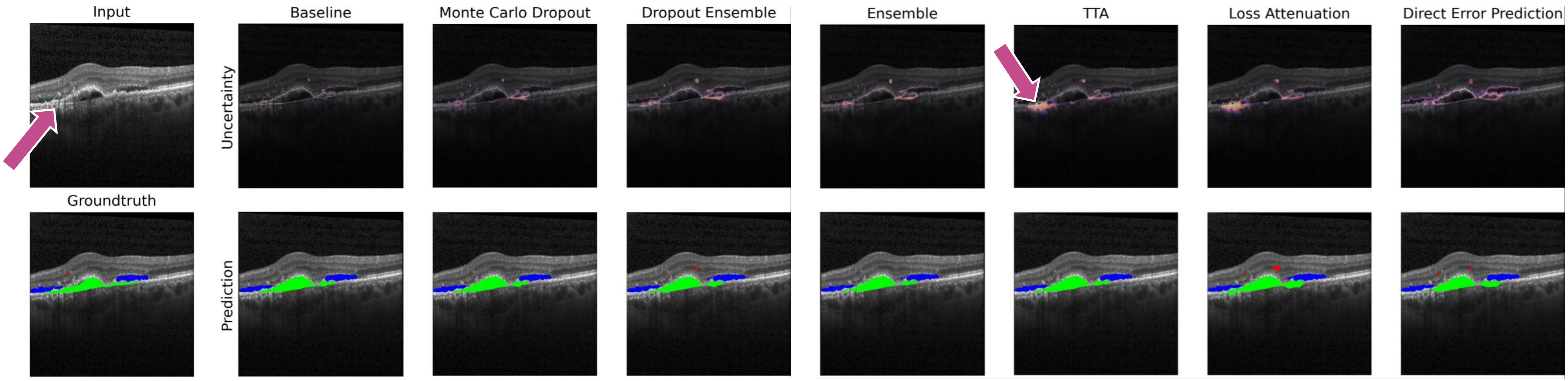
→ Uncertainty heads produce reasonable uncertainty estimates

Calibration of the Softmax distribution

- **Idea**: Confidence of a model should match the frequency the model making a correct prediction
- **Remark**: Considers the Softmax output of the methods only
- Sampling improves calibration of the output distribution
- Behavior of baseline and sample-free methods similar



Visual Comparison – RETOUCH Dataset



CONCLUSION

Conclusion

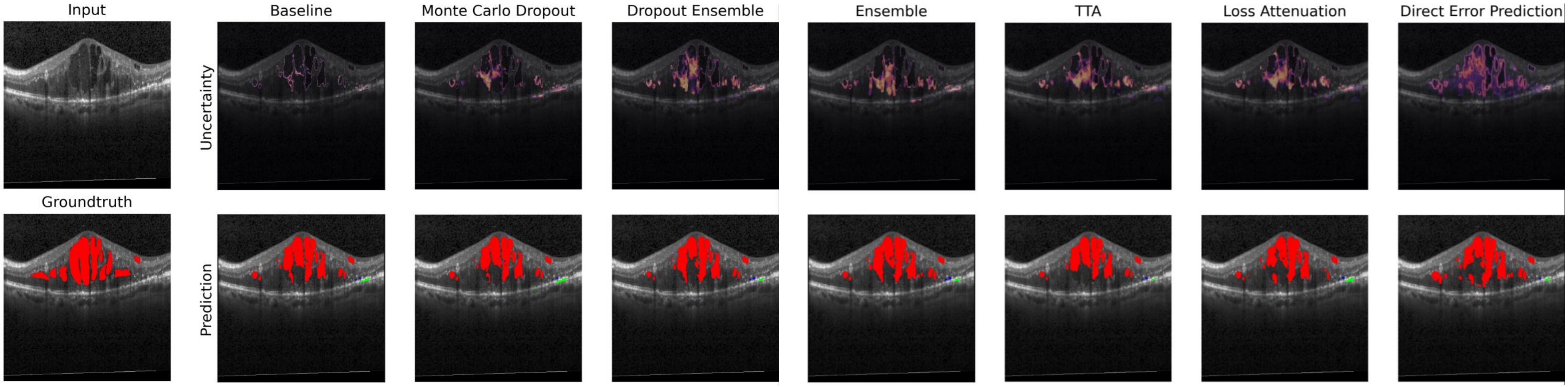
- Segmentation performance close to participants of the RETOUCH challenge without additional engineering
- All methods improve upon the baseline uncertainty estimate considerably
- Adding dropout to ensembles helps with uncertainty estimation quality
- Incorporating uncertainty during training does not boost calibration/meaningfulness of the softmax outputs, but aggregating samples does
- Similar performance on imagewise uncertainty measures except for Monte Carlo Dropout

Future Work

- Evaluation of methods independent of how uncertainty estimates are obtained
- Evaluation of uncertainty under dataset shift (DUKE Dataset) to better reflect epistemic uncertainty

Future Work

- Evaluation of methods independent of how uncertainty estimates are obtained
- Evaluation of uncertainty under dataset shift (DUKE Dataset) to better reflect epistemic uncertainty



References

[1] O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: International Conference on Medical image computing and computer-assisted intervention. Springer. 2015, pp. 234–241

[2] Y. Gal and Z. Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: international conference on machine learning. 2016

[3] G. Wang et al. “Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks”. In: Neurocomputing 338 (2019)

References

[4] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: Advances in neural information processing systems. 2017

[5] S. Bachstein. “Uncertainty Quantification in Deep Learning”. MA thesis. 2019.

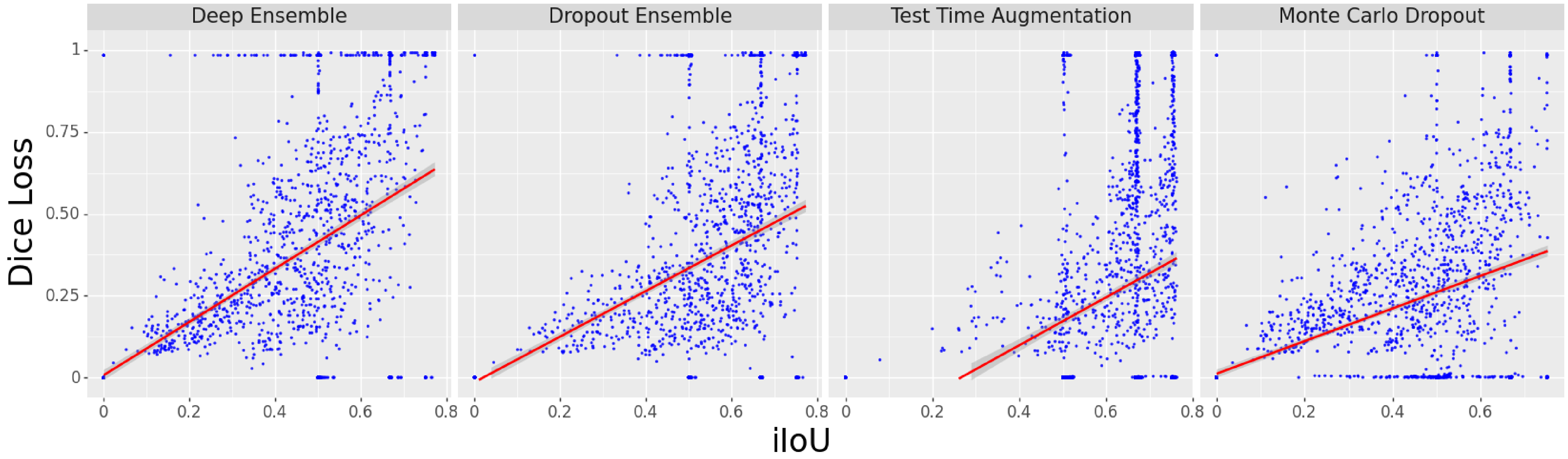
[6] A. Kendall and Y. Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: Advances in neural information processing systems. 2017

Sources for Figures

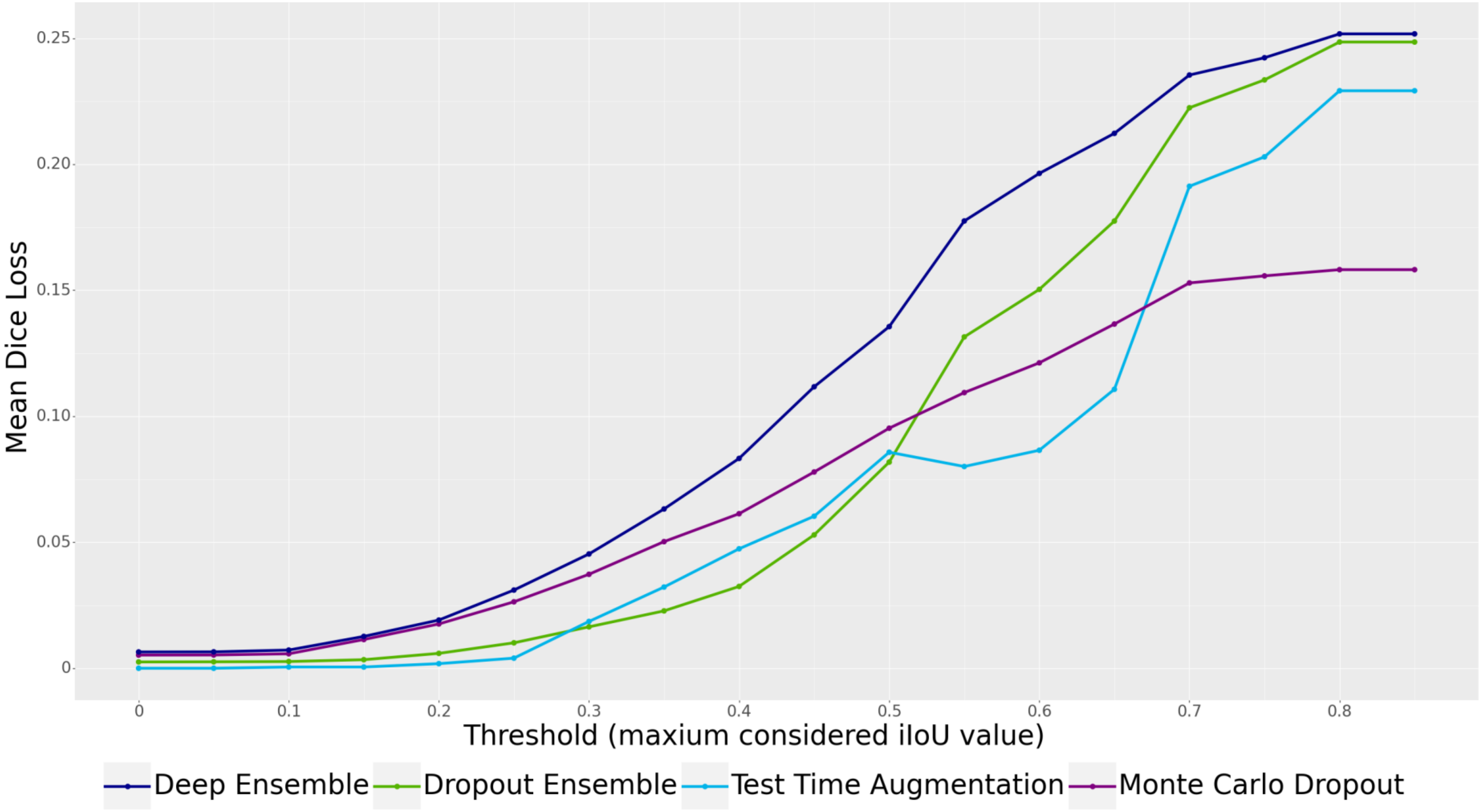
- https://engedal.it/saadan-optimerer-du-retina-billeder/
- https://www.zeiss.com/meditec/int/product-portfolio/optical-coherence-tomography-devices.html
- https://retouch.grand-challenge.org/Background/
- https://www.flaticon.com/free-icon/doctor_194915
- https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2
- Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting", JMLR 2014
- https://medium.com/konvergen/understanding-dropout-ddb60c9f98aa

Supplementary Material

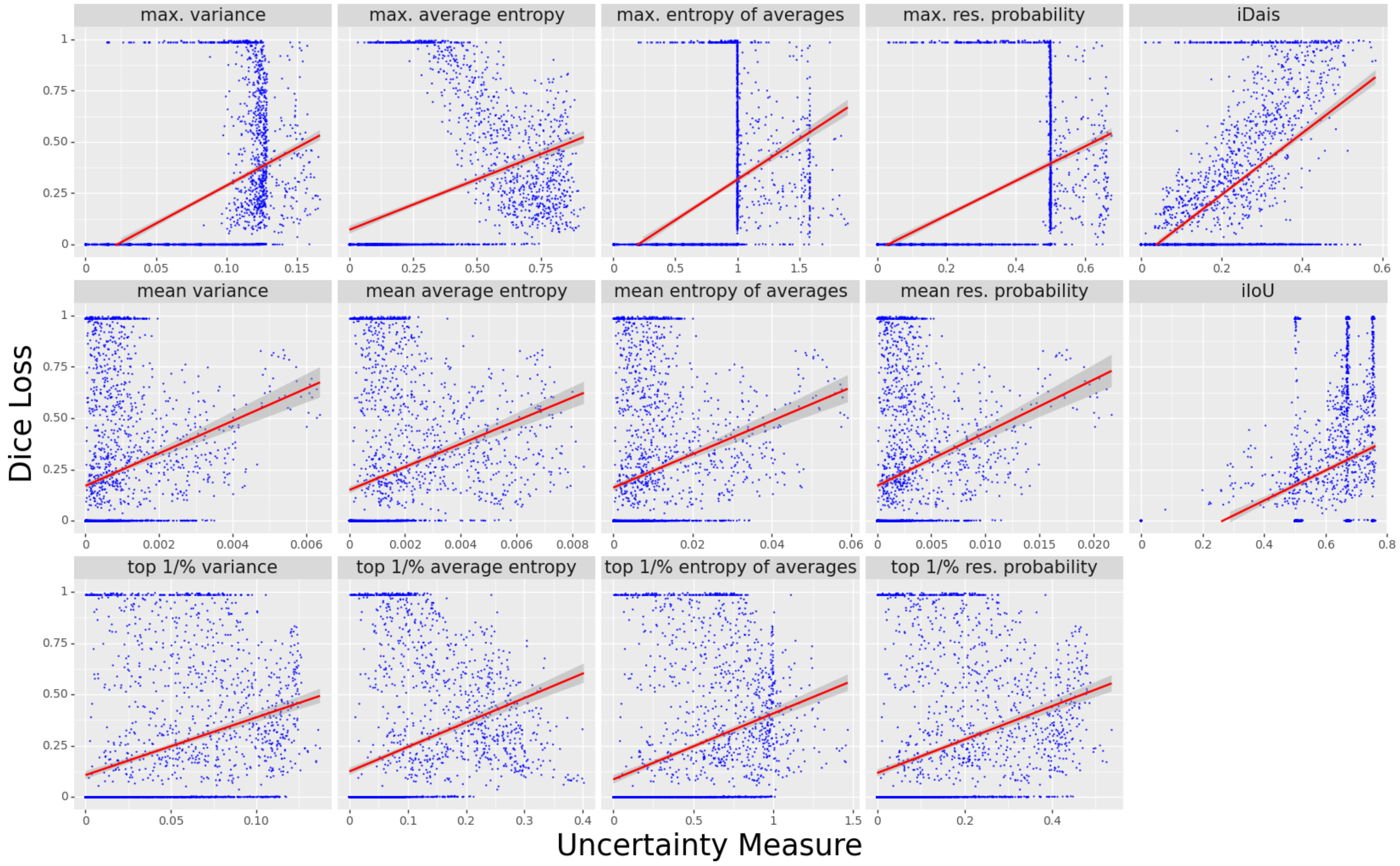
Supplementary Material: Performance (iIoU)



Supplementary Material: Performance (iIoU)



Supplementary Material: Imagewise Scores



Supplementary Material: Segmentation Quality

	Dice score				
Method	PED	SRF	IRF	Mean (Per Volume)	Mean (Per B-Scan)
Baseline	0.646	0.680	0.659	0.662	0.722
Monte Carlo Dropout	0.646	0.680	0.659	0.662	0.841
Ensemble	0.636	**0.694**	0.670	0.666	0.748
Dropout Ensemble	0.632	0.692	**0.671**	0.665	0.751
Test Time Augmentation	0.519	0.563	0.446	0.509	0.770
Loss Attenuation	0.651	0.663	0.635	0.650	0.722
Direct Error Prediction	**0.677**	0.685	0.656	**0.672**	**0.912**