Confusion Matrix - noisy labels

January 20, 2016

1 White noise

Labels distribution

	Test	*Train*$^{(s)}$	N1	N2	N3	N4	N5	N6
size	4661	4152	..	..	..	..	..	..
1	246	214	220	195	207	614	757	1015
0	4415	3938	3932	3957	3945	3538	3395	3137
(1)%	5.3%	5%	5.3%	4.7%	5%	14.8%	18.2%	24.4%

True confusion matrices

$$Q_1 = \begin{pmatrix} .99 & .01 \\ .19 & .81 \end{pmatrix} \; Q_2 = \begin{pmatrix} .98 & .02 \\ .38 & .62 \end{pmatrix} \; Q_3 = \begin{pmatrix} .97 & .03 \\ .57 & .43 \end{pmatrix}$$
$$Q_4 = \begin{pmatrix} .9 & .1 \\ .01 & .99 \end{pmatrix} \; Q_5 = \begin{pmatrix} .85 & .015 \\ .15 & .985 \end{pmatrix} \; Q_6 = \begin{pmatrix} .8 & .02 \\ .2 & .98 \end{pmatrix}$$

Learning

Training for 10000 iters ($\approx$ 150 epochs)- Architecture: $3 \times \{Conv, MaxPool, ReLU\}$ (the baseline) $\rightarrow$ InfoGain loss ($H := Q$).

$$\mathcal{L} = -\frac{1}{N} \sum_i \left(H_{l^{(n)},0}. \log \hat{p}_0^{(n)} + H_{l^{(n)},1}. \log \hat{p}_1^{(n)} \right)$$

Where $l^{(n)}$ and $\hat{p}^{(n)}$ are the label and class probabilities of sample n

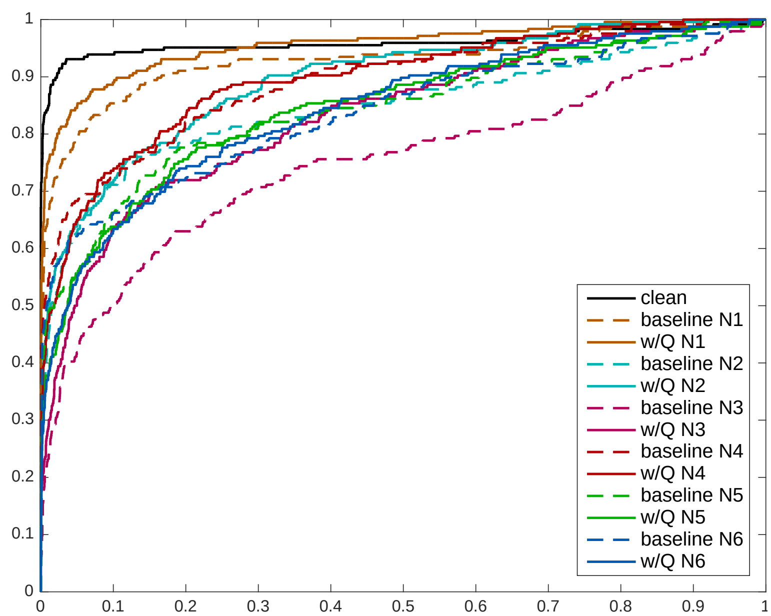

Figure 1: Roc curves: variant noise levels - baseline vs. Q

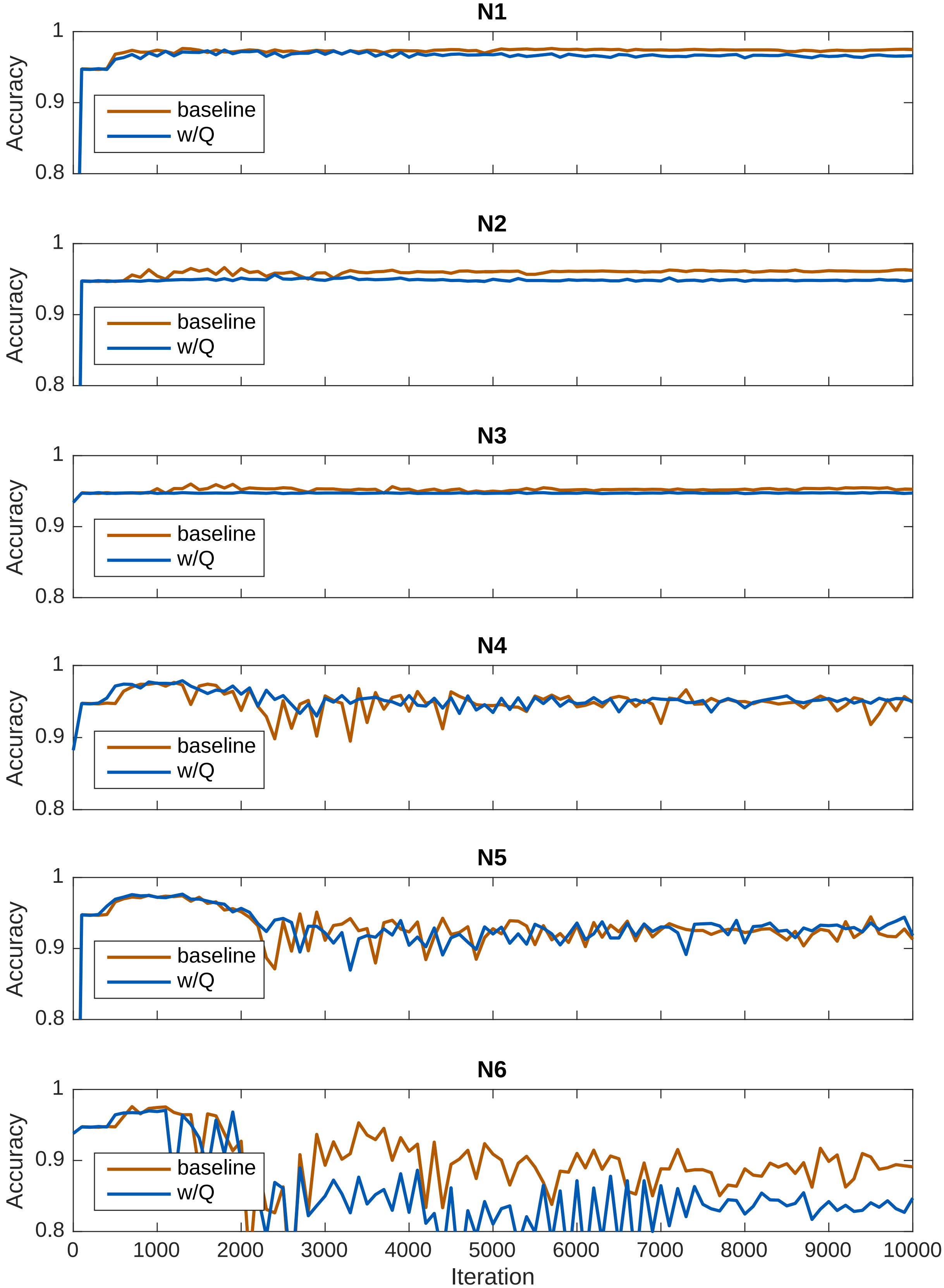

Figure 2: Accuracies: variant noise levels - baseline vs. Q

2 Image dependent noise

Thresholding the old classifier scores : 0 if score < th1 , 1 if score > th2

N1: th1 = 10, th2 = 30

N2: th1 = th2 = 20

N3: th1 = th2 = 30

N3: th1 = th2 = 40

	$Test$	$Train^{(s)}$	$Train^{(f)}$	N1	N2	N3	N4
size	4661	4152	381949	249683	381916	381942	381947
1	246	214	-	15862	32132	15827	10079
0	4415	3938	-	233821	349784	366115	371868
(1)%	5.3%	5%	-	6.35%	8.41%	4.14%	2.64%

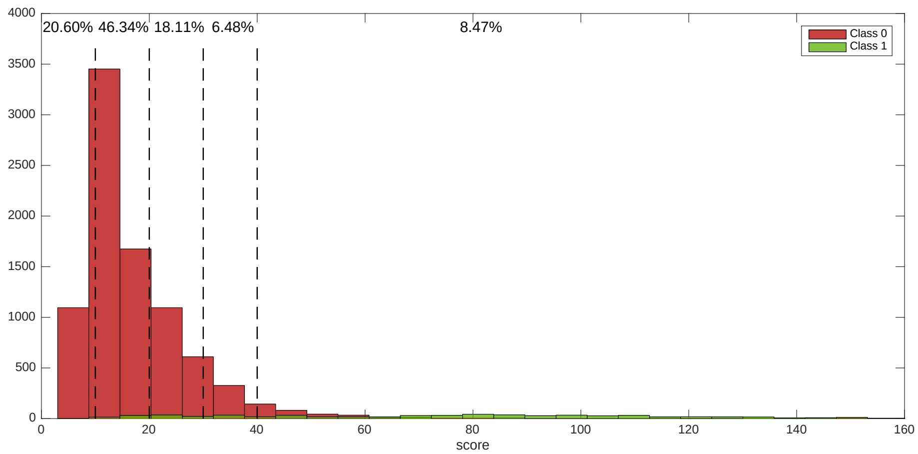

Figure 3: Benchamrk dataset

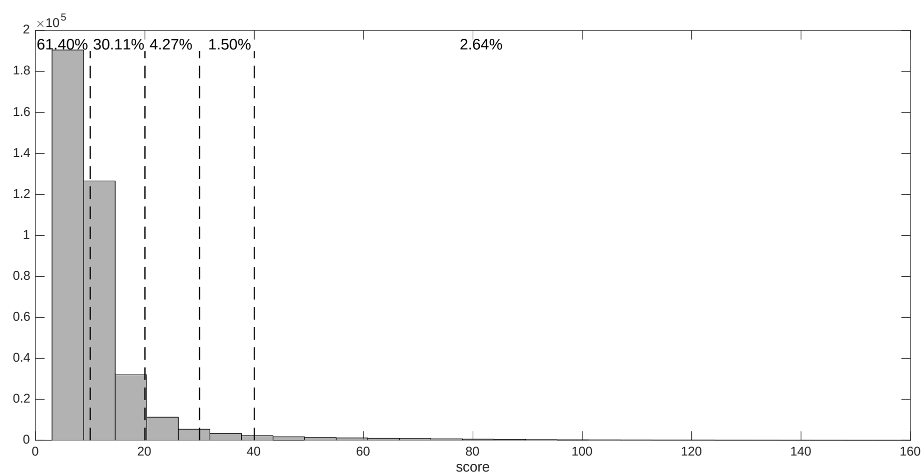

Figure 4: Training dataset

We estimate Q on a hand-labeled subset $Train^{(s)} \subset Train^{(f)}$

$$Q_1 = \begin{pmatrix} .979 & .021 \\ .08 & .92 \end{pmatrix} \; Q_2 = \begin{pmatrix} .9502 & .0498 \\ .2103 & .7897 \end{pmatrix} \; Q_3 = \begin{pmatrix} .9963 & .0037 \\ .3925 & .6075 \end{pmatrix} \; Q_4 = \begin{pmatrix} .9952 & .0048 \\ .5447 & .4553 \end{pmatrix}$$

Training for 10000 iters (≈ 2 epochs):

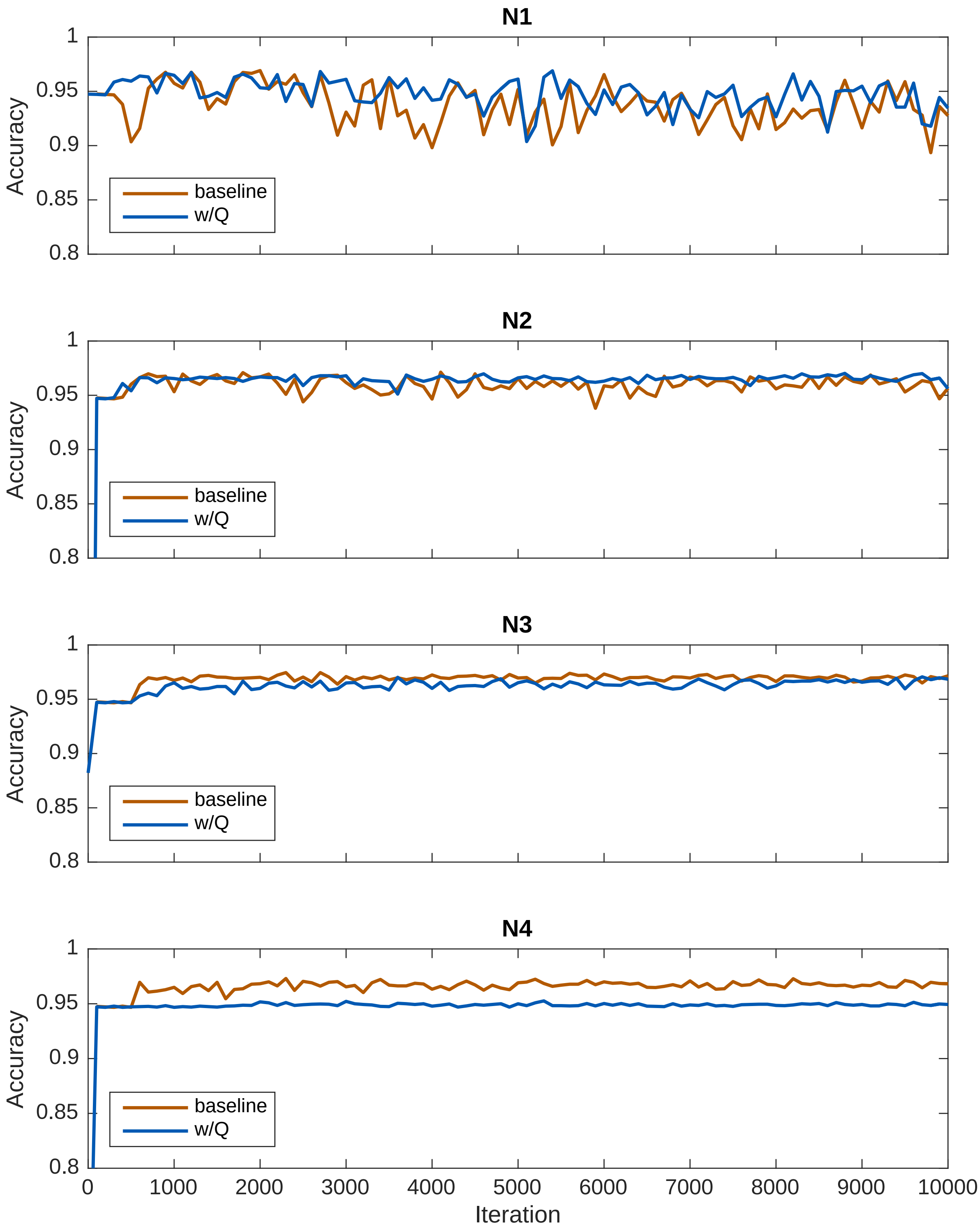

Figure 5: Accuracies: baseline vs. Q

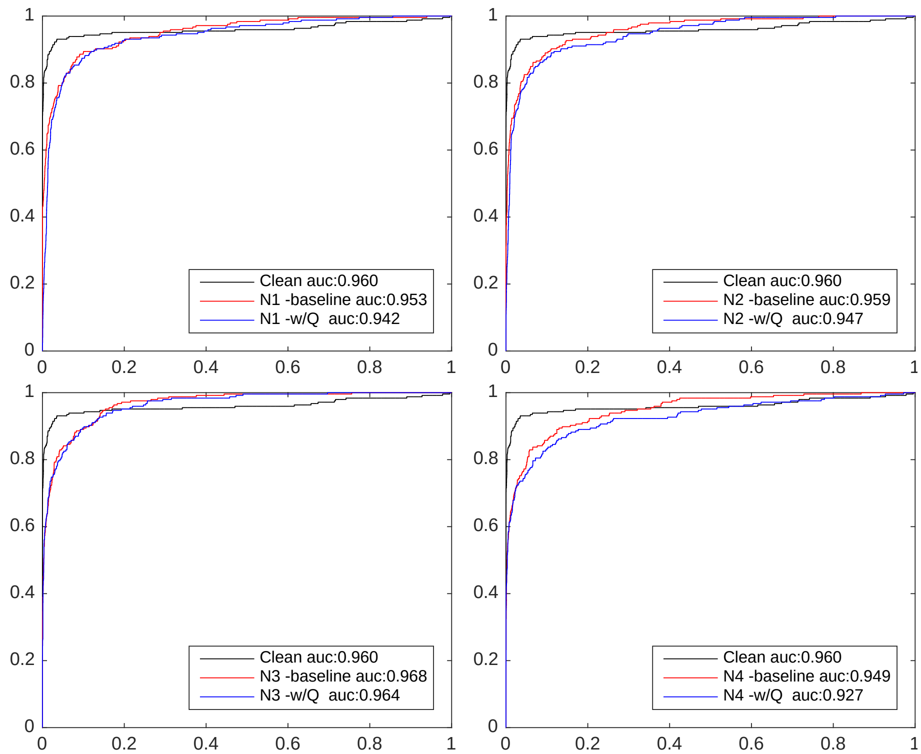

Figure 6: Roc curves: baseline vs. Q

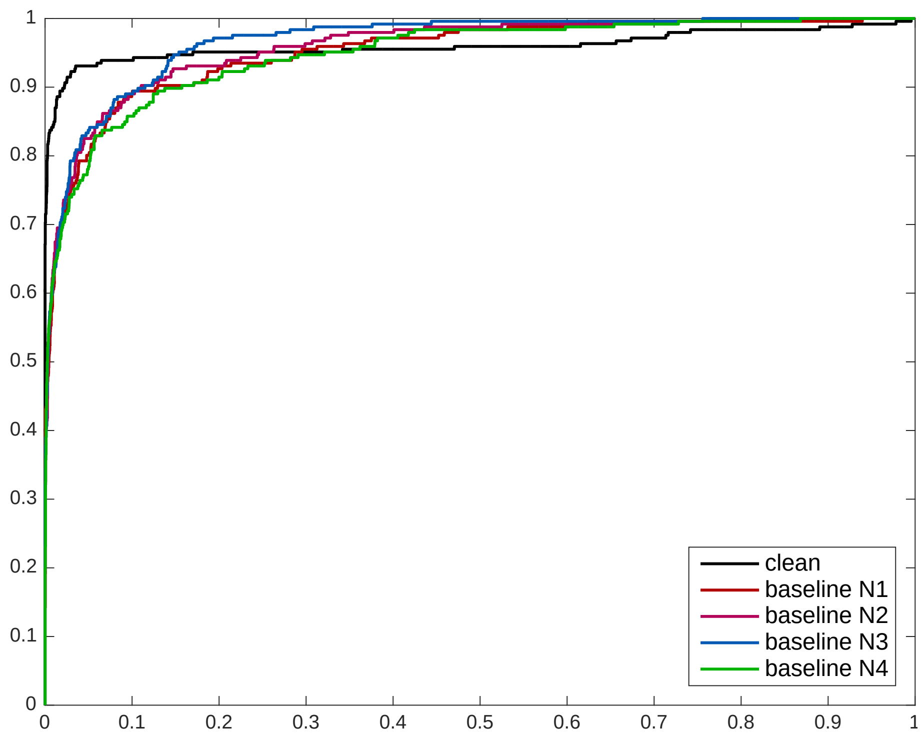

Figure 7: Roc curves: baseline N1 -N2 -N3

Training for 10000 iters ($\approx$ 2 epochs) - Then tuning w/Q for 10000 iters:

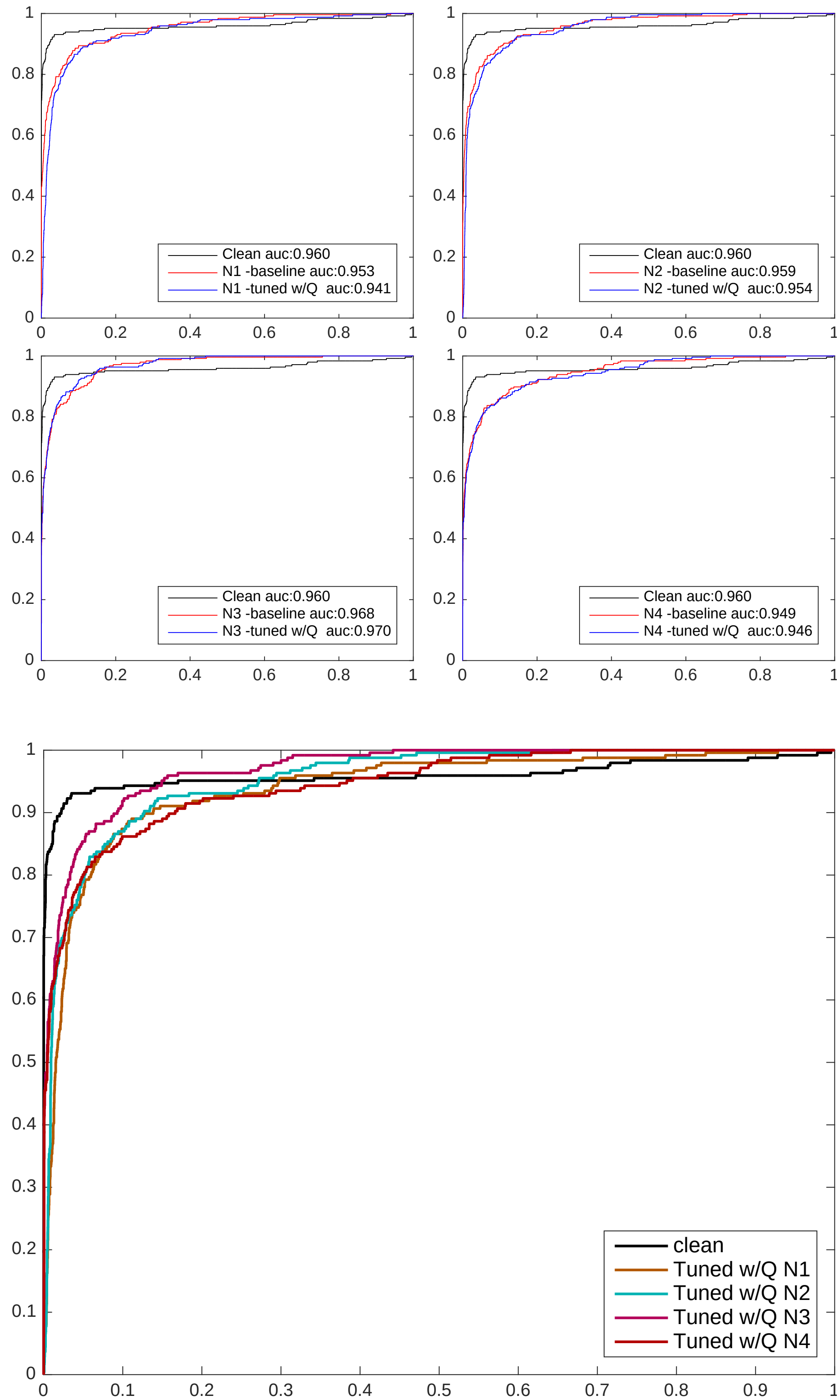

Figure 8: Roc curves: tuning N1 - N2 - N3

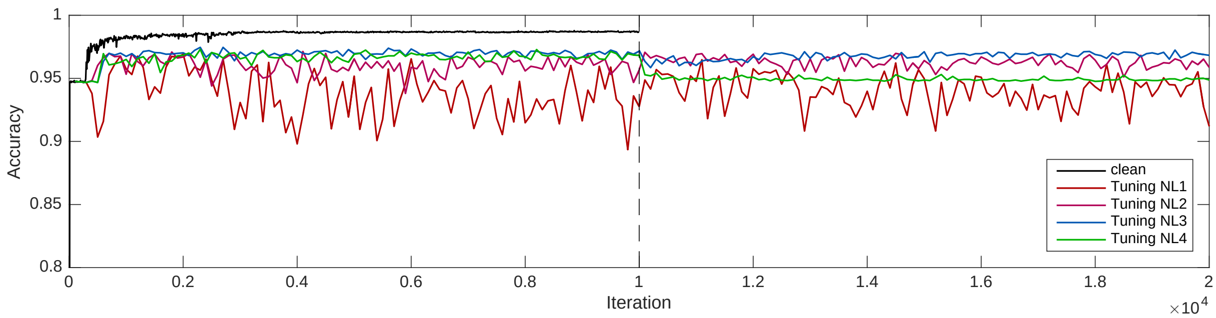

Figure 9: Accuracies - varying the thresholds + tuning w/Q

3 Clean datasets

C1: Multimedia Lab @Hong Kong
C2: $Train^{(s)}$

	C1	C2: $Train^{(s)}$
Size	4151	4152
0	3362	3838
1	789	214
(1)%	19%	5%

Test information loss with $Q_1 = \begin{pmatrix} .99 & .01 \\ .19 & .81 \end{pmatrix}$

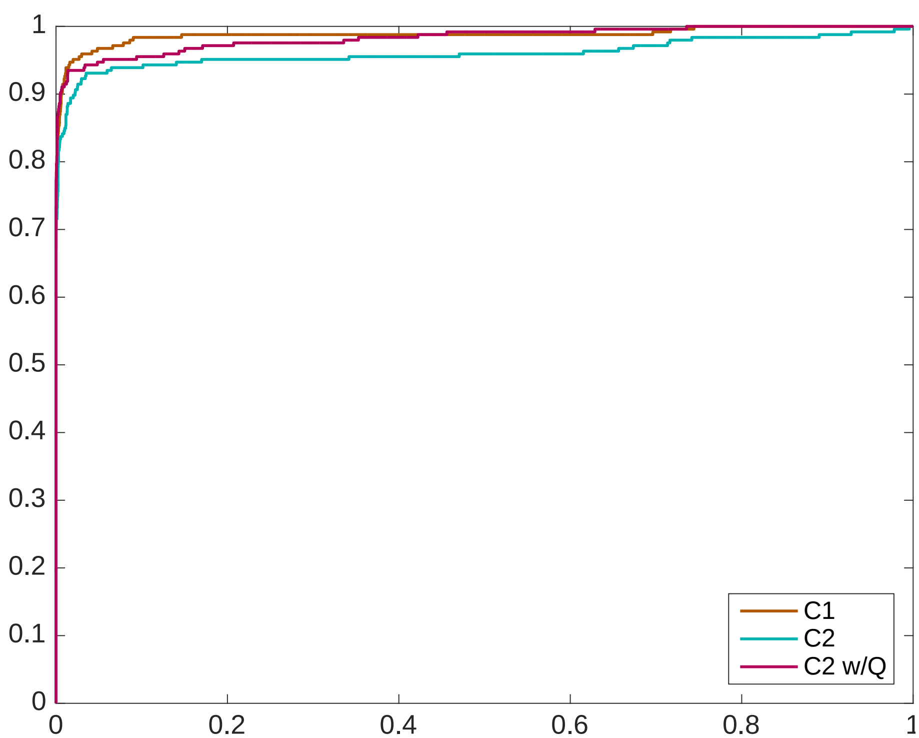

Figure 10: Roc curves - Clean training sets

4 Balancing the datasets

For N1, N2 & N3, we randomly choose 4× number of 1s from the 0' samples.

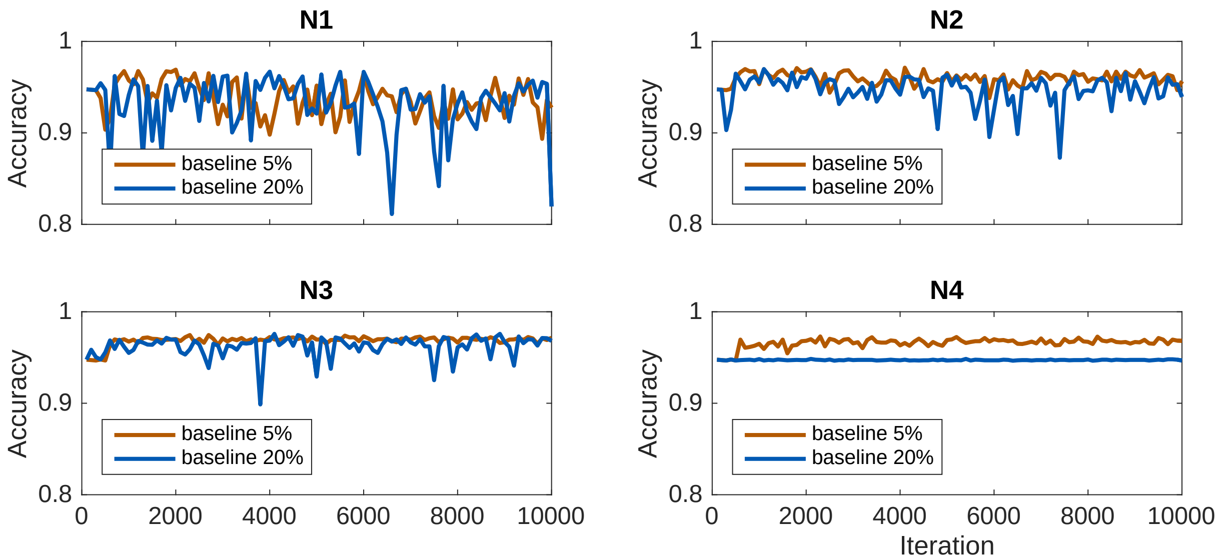

Figure 11: Accuracies

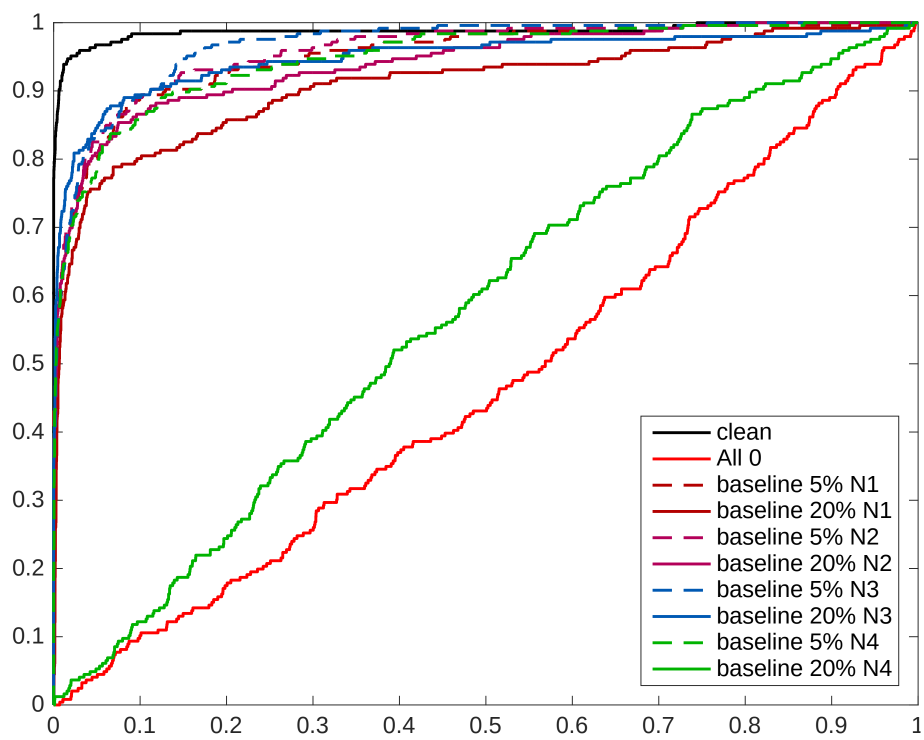

Figure 12: Roc curves - baseline model 5% vs 20%

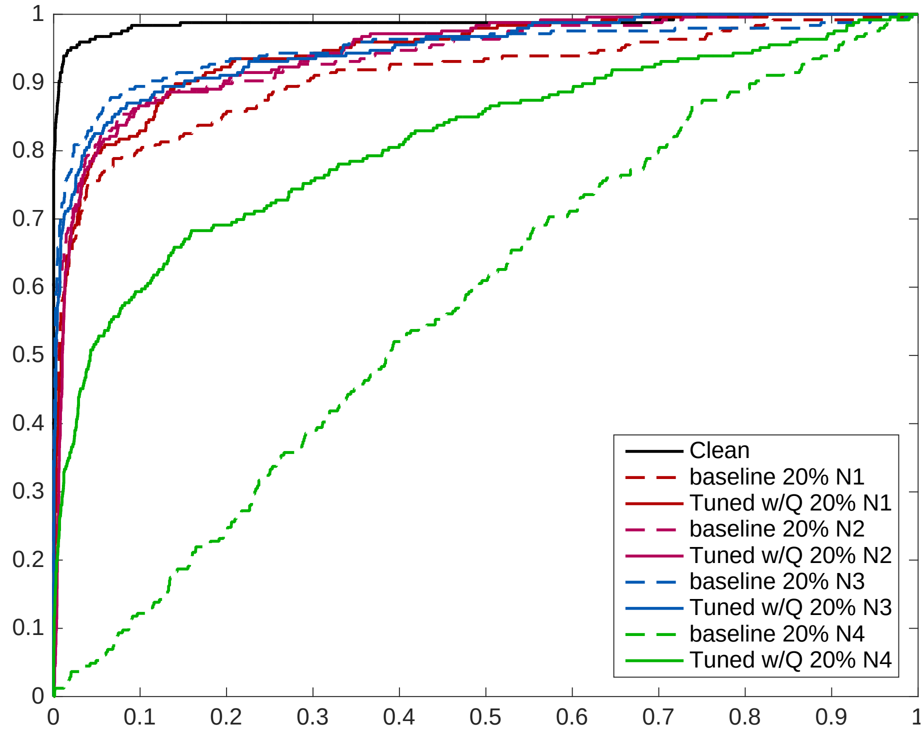

Figure 13: Roc curves - tuning w/ Q

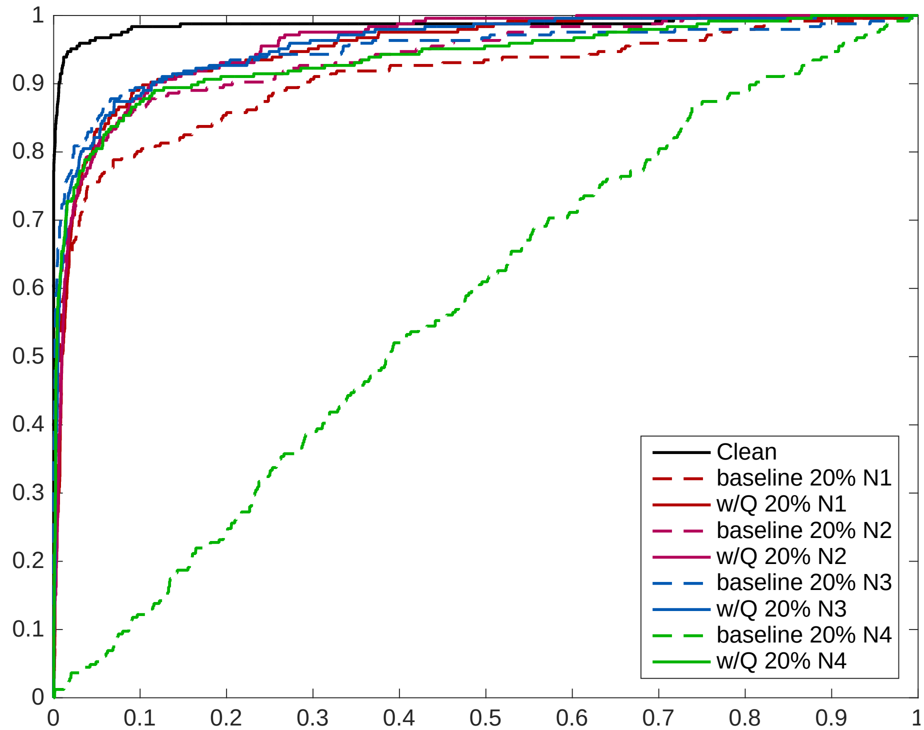

Figure 14: Roc curves - Learning w/ Q

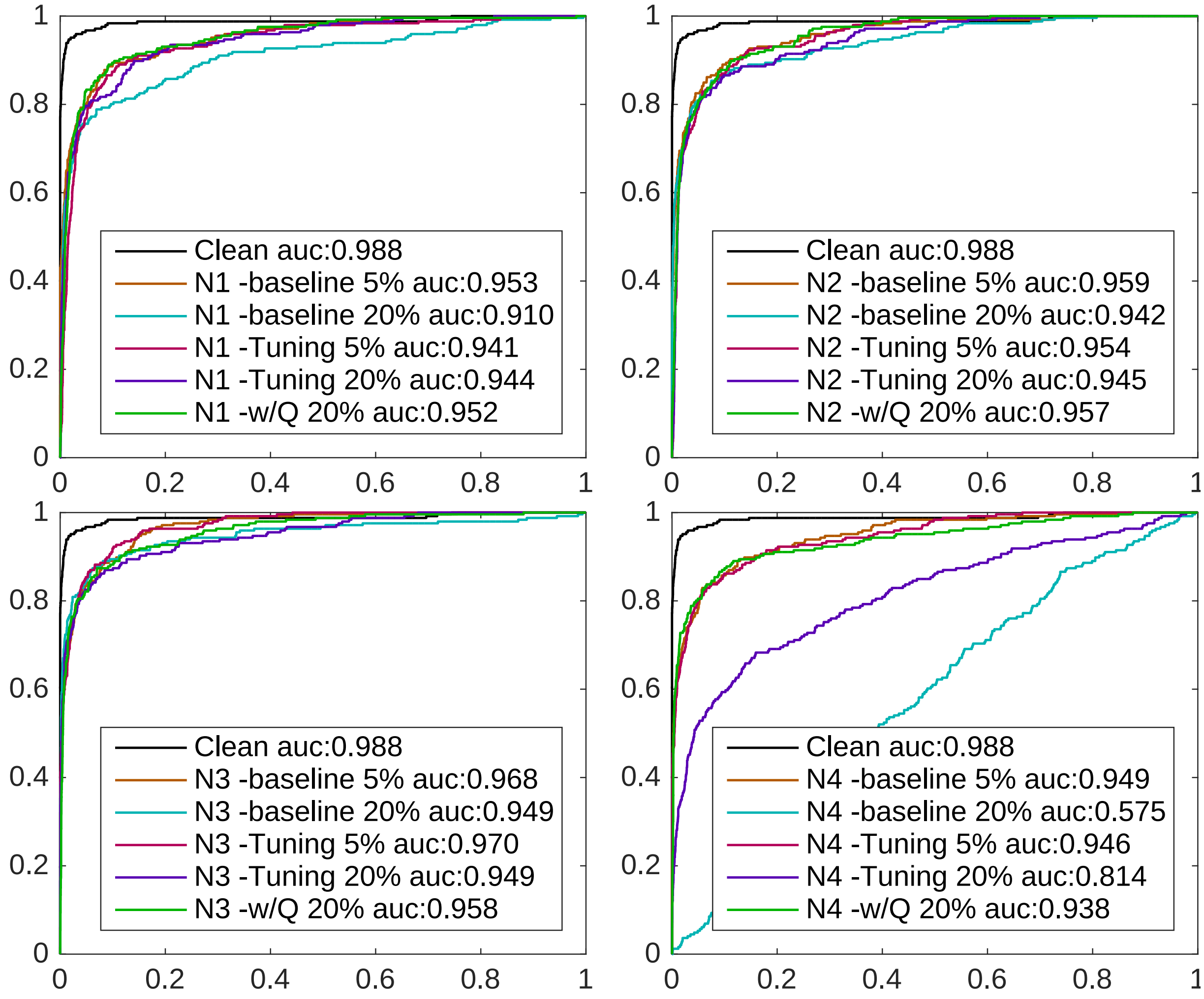

Figure 15: Roc curves - comparison

For N1, N2 & N3, we randomly choose $1\times$ number of 1s from the 0' samples.

Baseline: batch loss $+\infty$

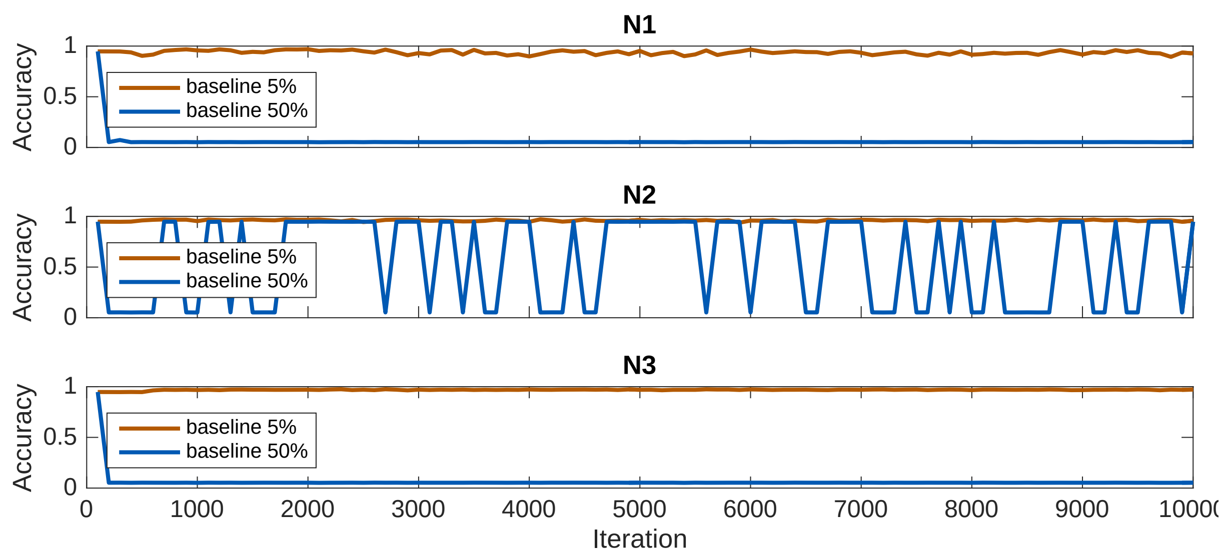

Figure 16: Accuracies

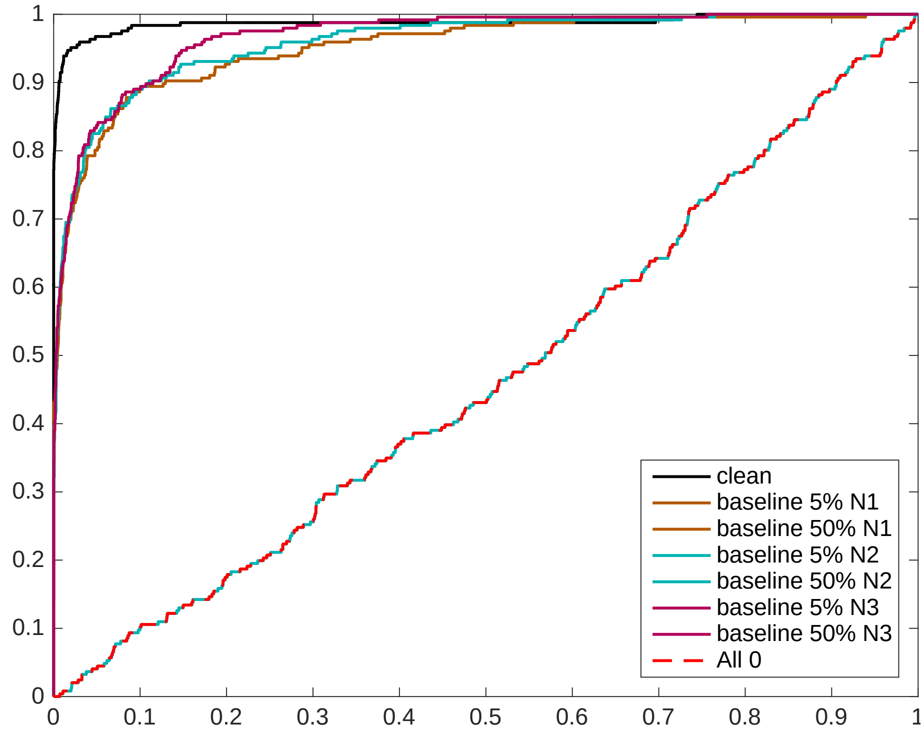

Figure 17: Roc curves - baseline model 5% vs 50% - baseline roc≡ 0

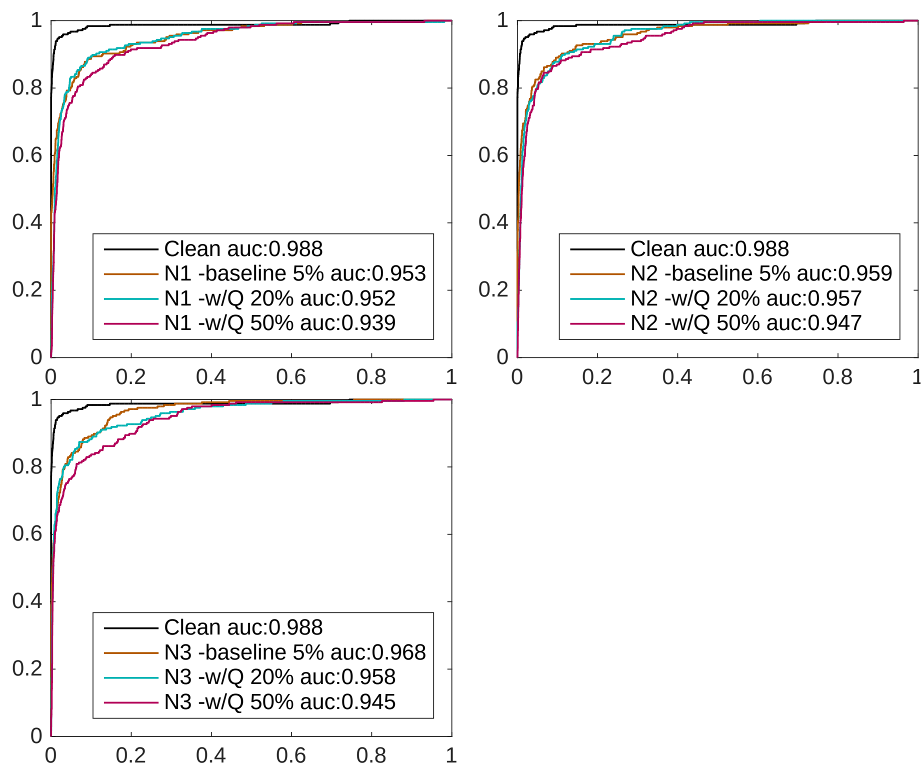

Figure 18: Roc curves - 50% Learning w/ Q