Supplementary Online Content

Burlina P, Paul W, Mathew P, Joshi N, Pacheco KD, Bressler NM. Low-shot deep learning of diabetic retinopathy with potential applications to address artificial intelligence bias in retinal diagnostics and rare ophthalmic diseases. *JAMA Ophthalmol.* Published online September 3, 2020. doi:10.1001/jamaophthalmol.2020.3269

This supplementary material has been provided by the authors to give readers additional information about their work.

eTable 1. Performance of Algorithms: Accuracy

Number of Samples	10	20	40	79	160	320	639	1280	2560	5120
RES_FT	51.47% [49.14%, 53.80%]	49.21% [46.88%, 51.54%]	54.93% [52.61%, 57.25%]	56.85% [54.54%, 59.16%]	59.85% [57.56%, 62.14%]	62.74% [60.48%, 65.00%]	67.44% [65.25%, 69.63%]	72.65% [70.57%, 74.73%]	73.33% [71.27%, 75.39%]	74.29% [72.25%, 76.33%]
RES_KNN	50.23% [47.90%, 52.56%]	51.25% [48.92%, 53.58%]	51.47% [49.14%, 53.80%]	56.34% [54.03%, 58.65%]	58.61% [56.31%, 60.91%]	59.80% [57.51%, 62.09%]	61.44% [59.17%, 63.71%]	60.65% [58.37%, 62.93%]	61.10% [58.83%, 63.37%]	61.04% [58.77%, 63.31%]
RES_SVM	50.45% [48.12%, 52.78%]	53.34% [51.01%, 55.67%]	55.04% [52.72%, 57.36%]	54.93% [52.61%, 57.25%]	63.19% [60.94%, 65.44%]	65.35% [63.13%, 67.57%]	68.52% [66.35%, 70.69%]	69.37% [67.22%, 71.52%]	71.74% [69.64%, 73.84%]	73.16% [71.09%, 75.23%]
RES_RF	50.91% [48.58%, 53.24%]	54.42% [52.10%, 56.74%]	57.36% [55.05%, 59.67%]	58.10% [55.80%, 60.40%]	63.08% [60.83%, 65.33%]	62.85% [60.60%, 65.10%]	66.65% [64.45%, 68.85%]	68.18% [66.01%, 70.35%]	68.97% [66.81%, 71.13%]	69.20% [67.05%, 71.35%]
DIM	**55.95% [53.63%, 58.27%]**	56.12% [53.81%, 58.43%]	**62.74% [60.48%, 65.00%]**	64.04% [61.80%, 66.28%]	65.23% [63.01%, 67.45%]	66.99% [64.80%, 69.18%]	72.14% [70.05%, 74.23%]	69.31% [67.16%, 71.46%]	71.69% [69.59%, 73.79%]	75.71% [73.71%, 77.71%]
DIM_KNN	52.10% [49.77%, 54.43%]	52.15% [49.82%, 54.48%]	54.59% [52.27%, 56.91%]	57.30% [54.99%, 59.61%]	59.34% [57.05%, 61.63%]	60.25% [57.97%, 62.53%]	63.82% [61.58%, 66.06%]	61.10% [58.83%, 63.37%]	63.02% [60.77%, 65.27%]	64.33% [62.10%, 66.56%]
DIM_SVM	54.53% [52.21%, 56.85%]	57.36% [55.05%, 59.67%]	62.06% [59.80%, 64.32%]	**64.27% [62.03%, 66.51%]**	**67.04% [64.85%, 69.23%]**	**70.72% [68.60%, 72.84%]**	**73.16% [71.09%, 75.23%]**	**74.29% [72.25%, 76.33%]**	**75.65% [73.65%, 77.65%]**	**76.39% [74.41%, 78.37%]**
DIM_RF	55.72% [53.40%, 58.04%]	**57.81% [55.51%, 60.11%]**	59.12% [56.83%, 61.41%]	62.80% [60.55%, 65.05%]	63.14% [60.89%, 65.39%]	68.01% [65.83%, 70.19%]	69.59% [67.44%, 71.74%]	69.37% [67.22%, 71.52%]	70.84% [68.72%, 72.96%]	71.06% [68.94%, 73.18%]

Accuracy shown in %, along with 95% CI (brackets). Best results are bold-faced. Rows compare various algorithms including: (top) a traditional fine-tuned ResNet (denoted as RES_FT), which is compared to other-low shot deep learning (LSDL) algorithms, shown in the bottom half of the table. These LSDL algorithms include: ResNet encoding fed into a random forest or SVM classifier (denoted as RES_RF and RES_SVM). Augmented

Multiscale Deep InfoMax (AMDIM) encoding [Bachman2019] yielding local and global features, fed to a classifier, consisting of either ResNet (using only local features, and denoted as DIM), and three other classifiers using the global features of DIM and either K Nearest Neighbors (DIM-KNN), Support Vector Machine (DIM_SVM), or Random Forest (DIM_RF). We show performance for values of N (numbers of samples per class) ranging from a minimum of N=10 to a maximum of N=5120. As seen in the table, the low-shot deep learning methods using DIM outperform the traditional fine-tuned ResNet method.

eTable 2. Performance of Algorithms: ROC AUC

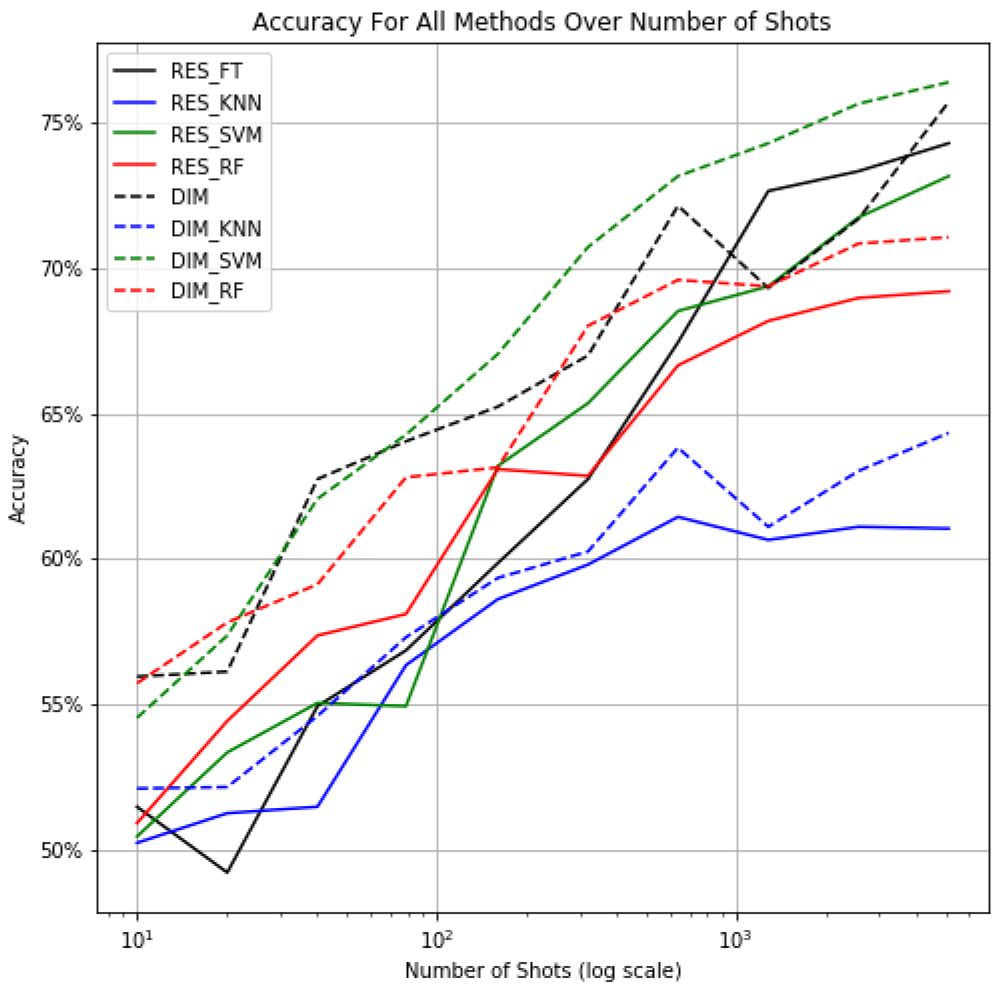
Number of Samples	10	20	40	79	160	320	639	1280	2560	5120
RES_FT	0.5178 [0.4909, 0.5447]	0.4799 [0.4530, 0.5068]	0.5671 [0.5404, 0.5938]	0.5859 [0.5594, 0.6124]	0.6585 [0.6332, 0.6838]	0.6624 [0.6372, 0.6876]	0.7441 [0.7212, 0.7670]	0.8028 [0.7823, 0.8233]	0.8089 [0.7887, 0.8291]	0.8330 [0.8140, 0.8520]
RES_KNN	0.5076 [0.4807, 0.5345]	0.5234 [0.4965, 0.5503]	0.5221 [0.4952, 0.5490]	0.5778 [0.5512, 0.6044]	0.6148 [0.5887, 0.6409]	0.6327 [0.6069, 0.6585]	0.6516 [0.6262, 0.6770]	0.6401 [0.6145, 0.6657]	0.6419 [0.6163, 0.6675]	0.6549 [0.6296, 0.6802]
RES_SVM	0.4992 [0.4722, 0.5262]	0.5657 [0.5390, 0.5924]	0.5912 [0.5648, 0.6176]	0.5944 [0.5680, 0.6208]	0.6787 [0.6539, 0.7035]	0.7089 [0.6849, 0.7329]	0.7595 [0.7372, 0.7818]	0.7782 [0.7566, 0.7998]	0.7971 [0.7763, 0.8179]	0.8078 [0.7875, 0.8281]
RES_RF	0.5055 [0.4786, 0.5324]	0.5639 [0.5372, 0.5906]	0.5940 [0.5676, 0.6204]	0.6238 [0.5979, 0.6497]	0.6742 [0.6493, 0.6991]	0.6900 [0.6655, 0.7145]	0.7235 [0.6999, 0.7471]	0.7451 [0.7223, 0.7679]	0.7483 [0.7256, 0.7710]	0.7564 [0.7340, 0.7788]
DIM	**0.5778 [0.5512, 0.6044]**	**0.6427 [0.6171, 0.6683]**	**0.6760 [0.6511, 0.7009]**	0.6746 [0.6497, 0.6995]	**0.7467 [0.7239, 0.7695]**	0.7351 [0.7119, 0.7583]	0.7794 [0.7579, 0.8009]	0.7559 [0.7335, 0.7783]	0.7846 [0.7633, 0.8059]	0.8348 [0.8159, 0.8537]
DIM_KNN	0.5248 [0.4979, 0.5517]	0.5267 [0.4998, 0.5536]	0.5625 [0.5358, 0.5892]	0.5898 [0.5634, 0.6162]	0.6134 [0.5873, 0.6395]	0.6481 [0.6226, 0.6736]	0.6770 [0.6522, 0.7018]	0.6527 [0.6273, 0.6781]	0.6690 [0.6440, 0.6940]	0.6884 [0.6638, 0.7130]
DIM_SVM	0.5440 [0.5172, 0.5708]	0.6027 [0.5764, 0.6290]	0.6525 [0.6271, 0.6779]	**0.7040 [0.6799, 0.7281]**	0.7455 [0.7227, 0.7683]	**0.7903 [0.7692, 0.8114]**	**0.8114 [0.7913, 0.8315]**	**0.8276 [0.8083, 0.8469]**	**0.8479 [0.8297, 0.8661]**	**0.8581 [0.8405, 0.8757]**
DIM_RF	0.5706 [0.5440, 0.5972]	0.6061 [0.5799, 0.6323]	0.6234 [0.5975, 0.6493]	0.6751 [0.6502, 0.7000]	0.7039 [0.6798, 0.7280]	0.7495 [0.7268, 0.7722]	0.7729 [0.7511, 0.7947]	0.7748 [0.7531, 0.7965]	0.7769 [0.7553, 0.7985]	0.7985 [0.7778, 0.8192]

ROC AUC, along with 95% CI (brackets). Best results are bold-faced. Rows compare various algorithms including: (top) a traditional fine-tuned ResNet (denoted as RES_FT), which is compared to other low-shot deep learning (LSDL) algorithms, shown in the bottom half of the table. These LSDL algorithms include: ResNet encoding fed into a random forest or SVM classifier (denoted as RES_RF and RES_SVM). Augmented Multiscale Deep InfoMax (AMDIM) encoding [Bachman2019] yielding local and global features, fed to a classifier, consisting of either ResNet (using only local features, and denoted as DIM), and three other classifiers using the global features of DIM and either K Nearest Neighbors (DIM-KNN), Support Vector Machine (DIM_SVM), or Random Forest (DIM_RF). We show performance for values of N (numbers of samples per class) ranging from a minimum of N=10 to a maximum of N=5120. As seen in the table, the low-shot deep learning methods using DIM outperform the traditional fine-tuned ResNet method.

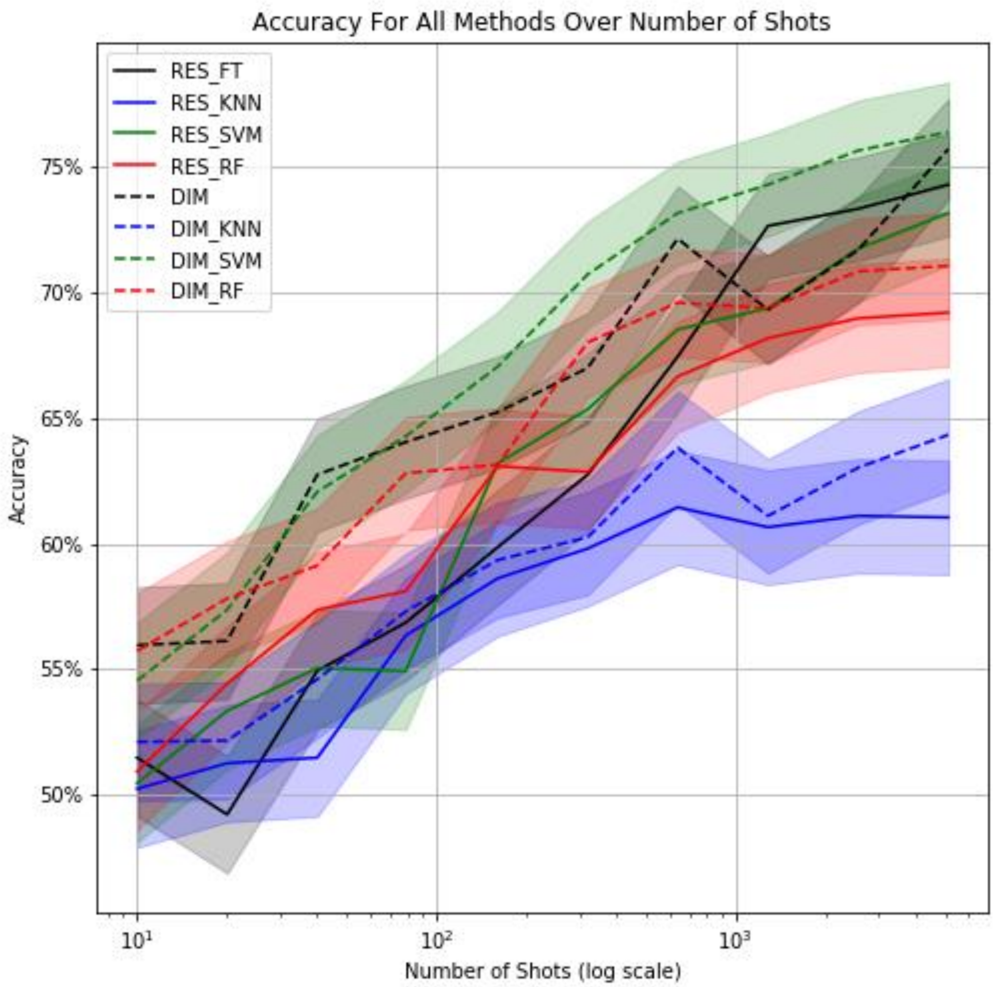
eTable 3. Performance of Algorithms: F1 Score

Number of Samples	10	20	40	79	160	320	639	1280	2560	5120
RES_FT	0.5648	0.5221	0.4865	0.4844	0.4925	0.5760	0.6991	0.7085	0.7201	0.7291
RES_KNN	0.5536	0.5176	0.5263	0.5830	0.5863	0.5984	0.6025	0.5962	0.5803	0.5870
RES_SVM	0.6067	0.5517	0.5340	0.5646	0.5812	0.6136	0.6323	0.6530	0.6799	0.7011
RES_RF	0.5893	0.5725	0.5962	0.5969	0.6414	0.6372	0.6525	0.6698	0.6780	0.6739
DIM	0.6381	**0.6599**	0.5846	0.5984	**0.7022**	0.6791	0.7113	0.6095	0.7076	0.7360
DIM_KNN	0.5447	0.5503	0.5674	0.5769	0.5864	0.5970	0.6120	0.5844	0.5926	0.6082
DIM_SVM	**0.6513**	0.5558	0.6086	**0.6363**	0.6600	**0.6899**	**0.7145**	**0.7265**	**0.7346**	**0.7446**
DIM_RF	0.5660	0.5666	**0.6316**	0.6319	0.6498	0.6866	0.6951	0.6871	0.7014	0.6985

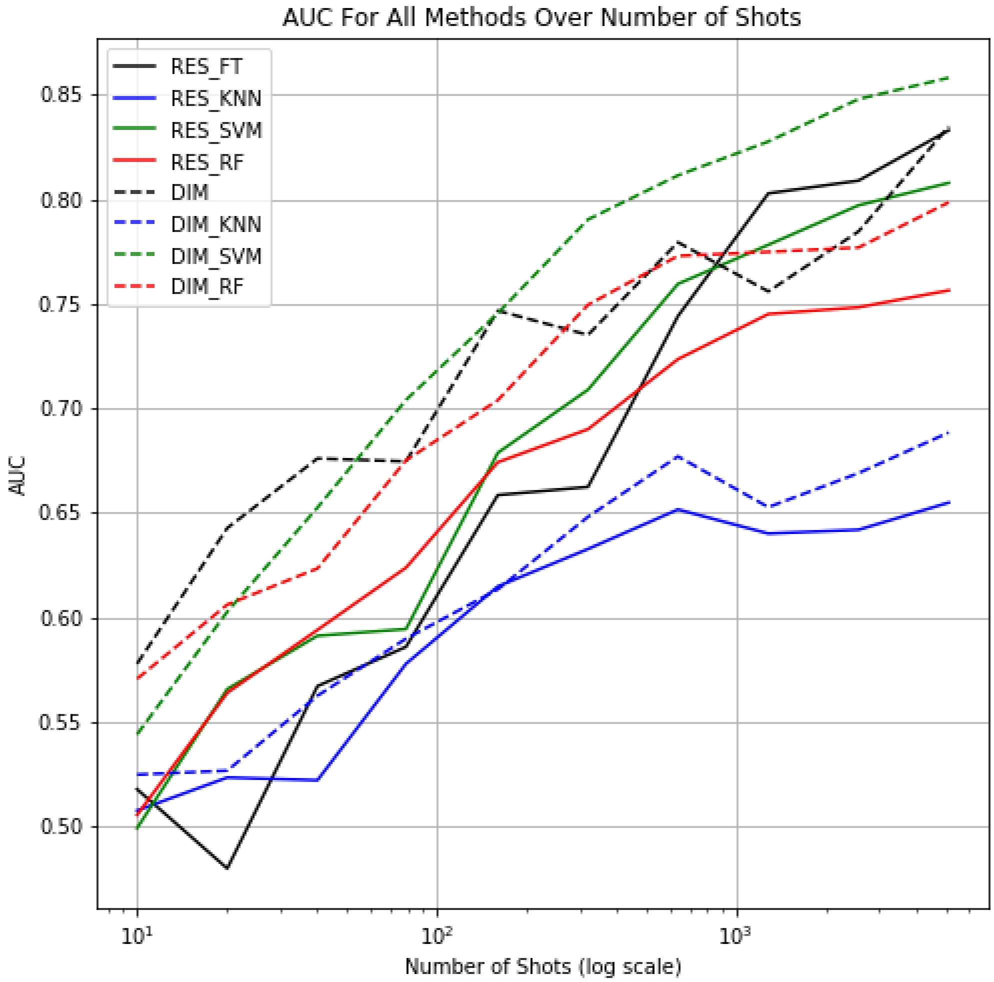
F1 score. Best results are bold-faced. Rows compare various algorithms including: (top) a traditional fine-tuned ResNet (denoted as RES_FT), which is compared to other low-shot deep learning (LSDL) algorithms, shown in the bottom half of the table. These LSDL algorithms include: ResNet encoding fed into a random forest or SVM classifier (denoted as RES_RF and RES_SVM). Augmented Multiscale Deep InfoMax (AMDIM) encoding [Bachman2019] yielding local and global features, fed to a classifier, consisting of either ResNet (using only local features, and denoted as DIM), and three other classifiers using the global features of DIM and either K Nearest Neighbors (DIM-KNN), Support Vector Machine (DIM_SVM), or Random Forest (DIM_RF). We show performance for values of N (numbers of samples per class) ranging from a minimum of N=10 to a maximum of N=5120. As seen in the table, the low-shot deep learning methods using DIM outperform the traditional fine-tuned ResNet method.



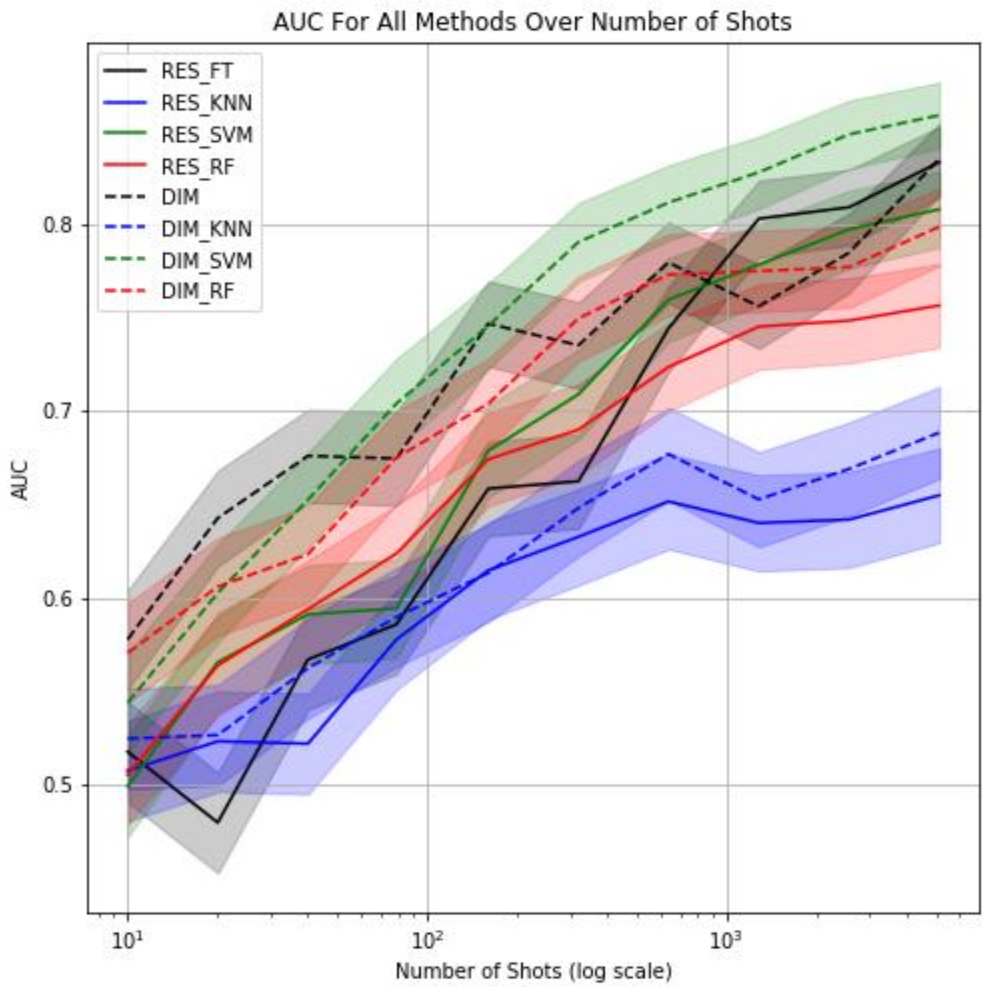
eFigure 1. Accuracy for All Methods and Number of Shots



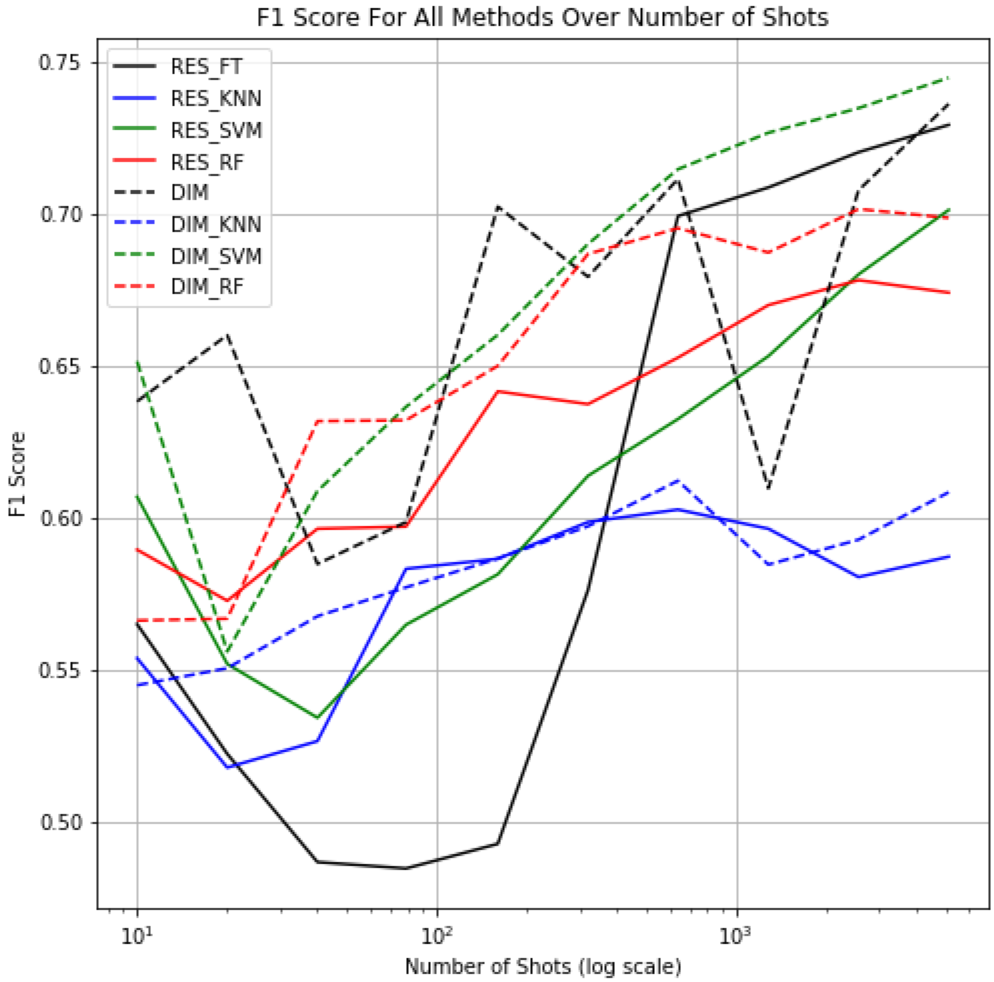
eFigure 2. Accuracy and Confidence Intervals for All Shots



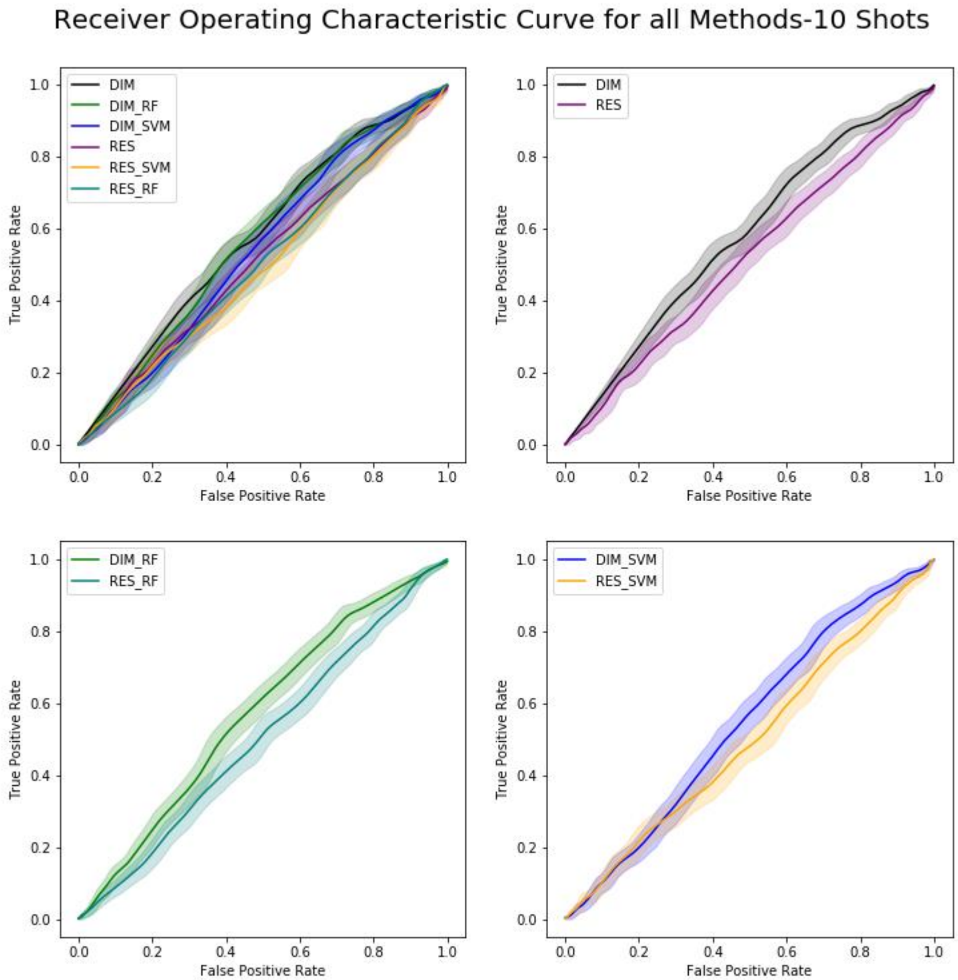
eFigure 3. ROC AUC for All Shots



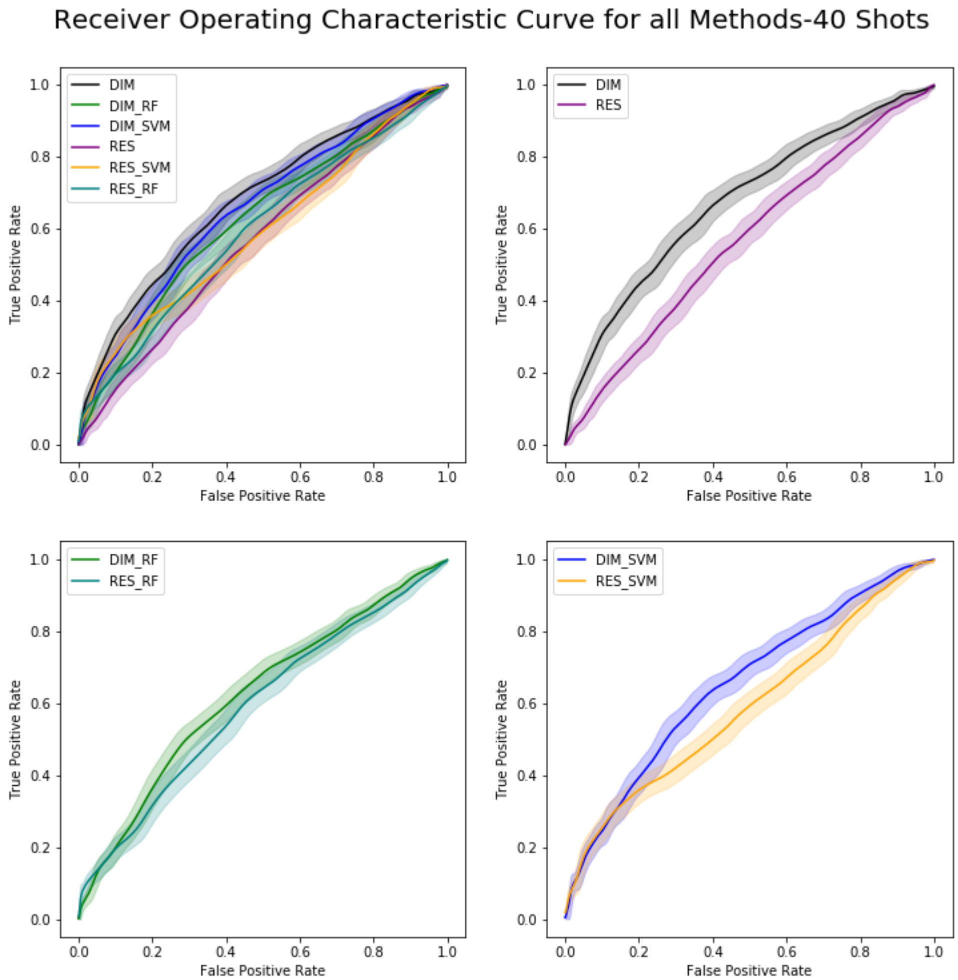
eFigure 4. ROC AUC for All Shots



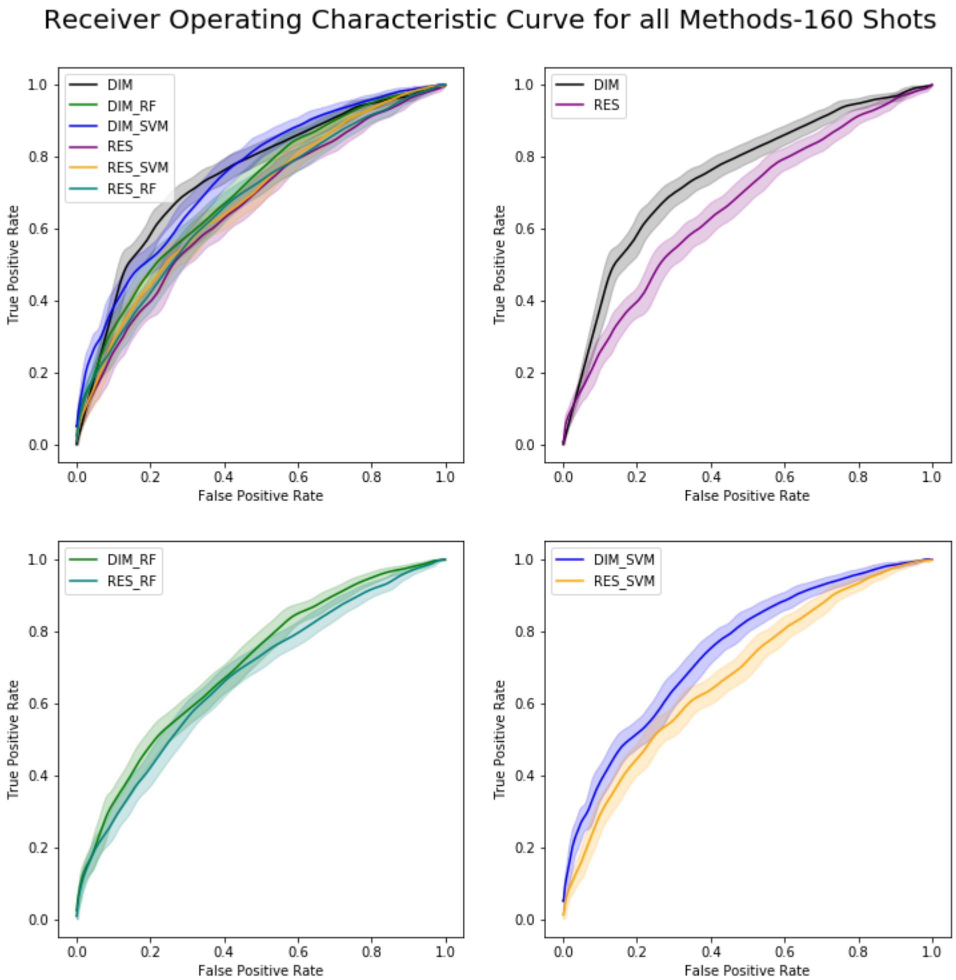
eFigure 5. F1 Score for All Shots



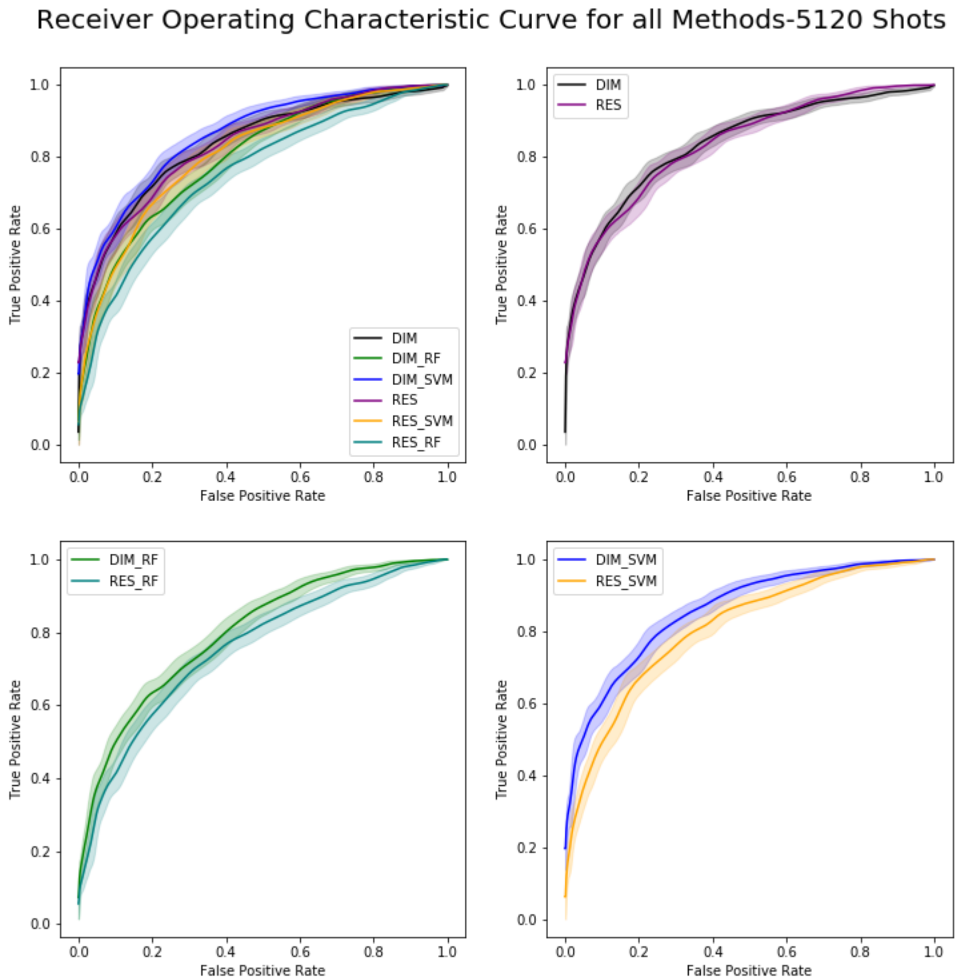
eFigure 6. N=10 Shots Results: ROCs and Confidence Intervals, (upper left) All, (rest) Two-Curve Comparisons of Methods



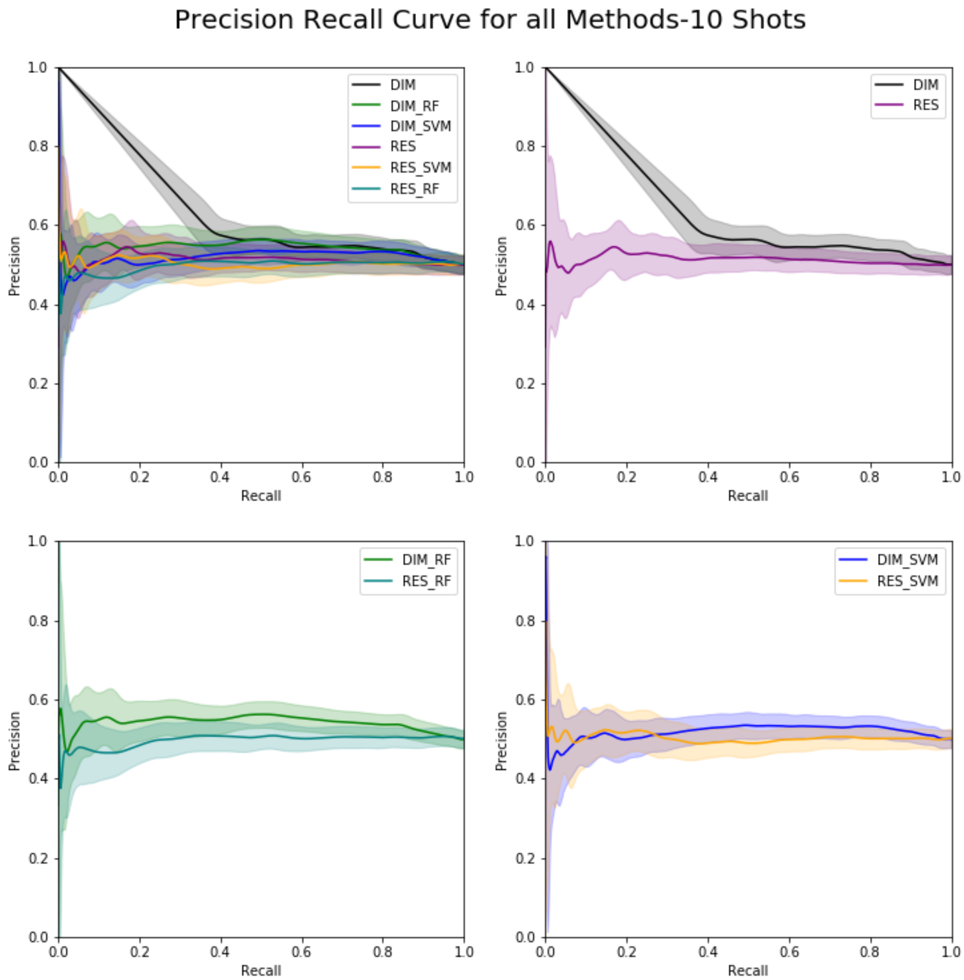
eFigure 7. N=40 Shots Results: ROCs and Confidence Intervals, (upper left) All Methods, (rest) Two by Two Comparison of Methods



eFigure 8. N=160 Shots Results: ROCs and Confidence Intervals, (upper left) All Methods, (rest) Two by Two Comparison of Methods

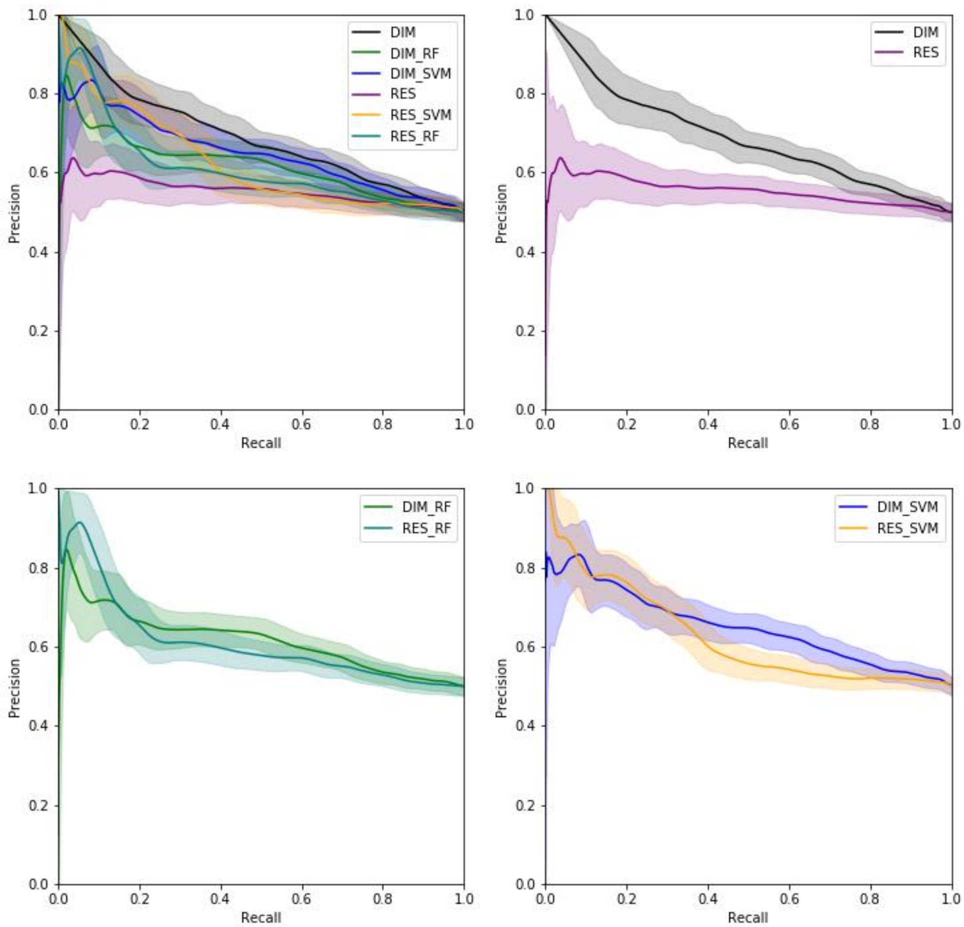


eFigure 9. N=5120 Shots Results: ROCs and Confidence Intervals, (upper left) All Methods, (rest) Two by Two Comparison of Methods

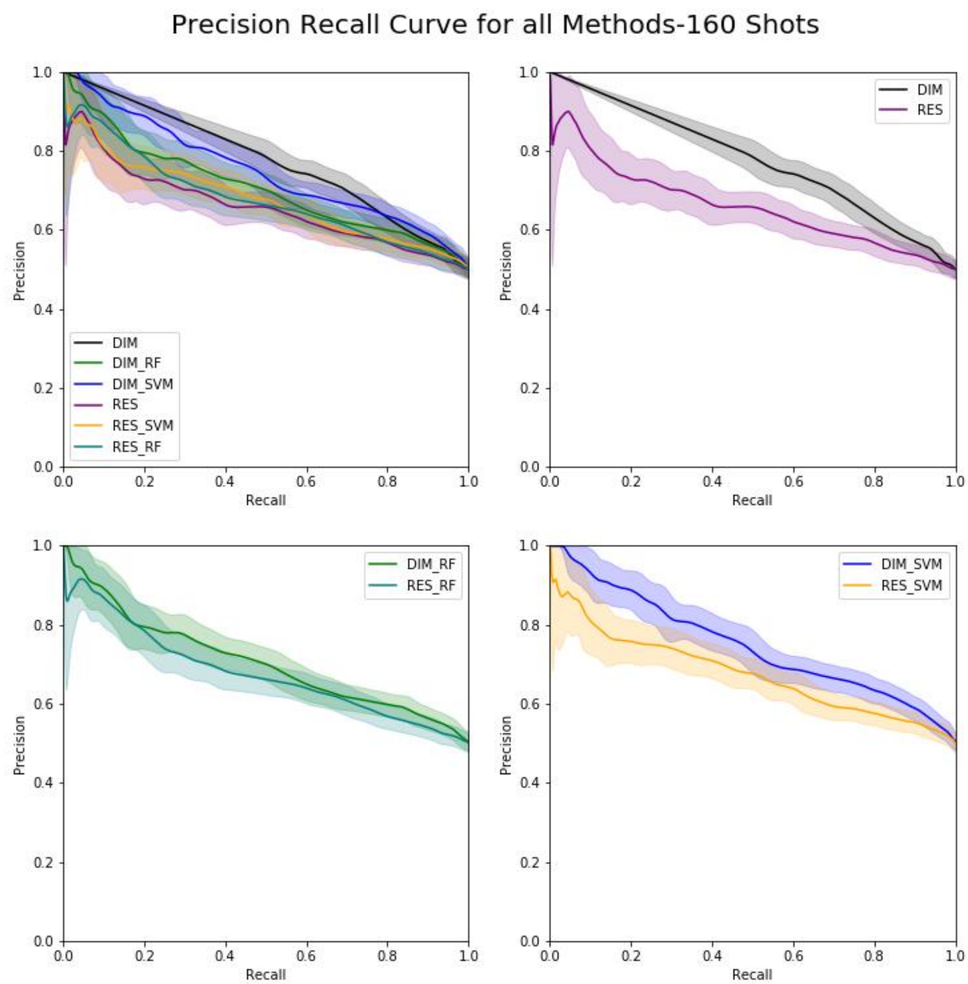


eFigure 10. N=10 Shots Results: PR Curves and Confidence Intervals, (upper left) All Methods, (rest) Two by Two Comparison of Methods

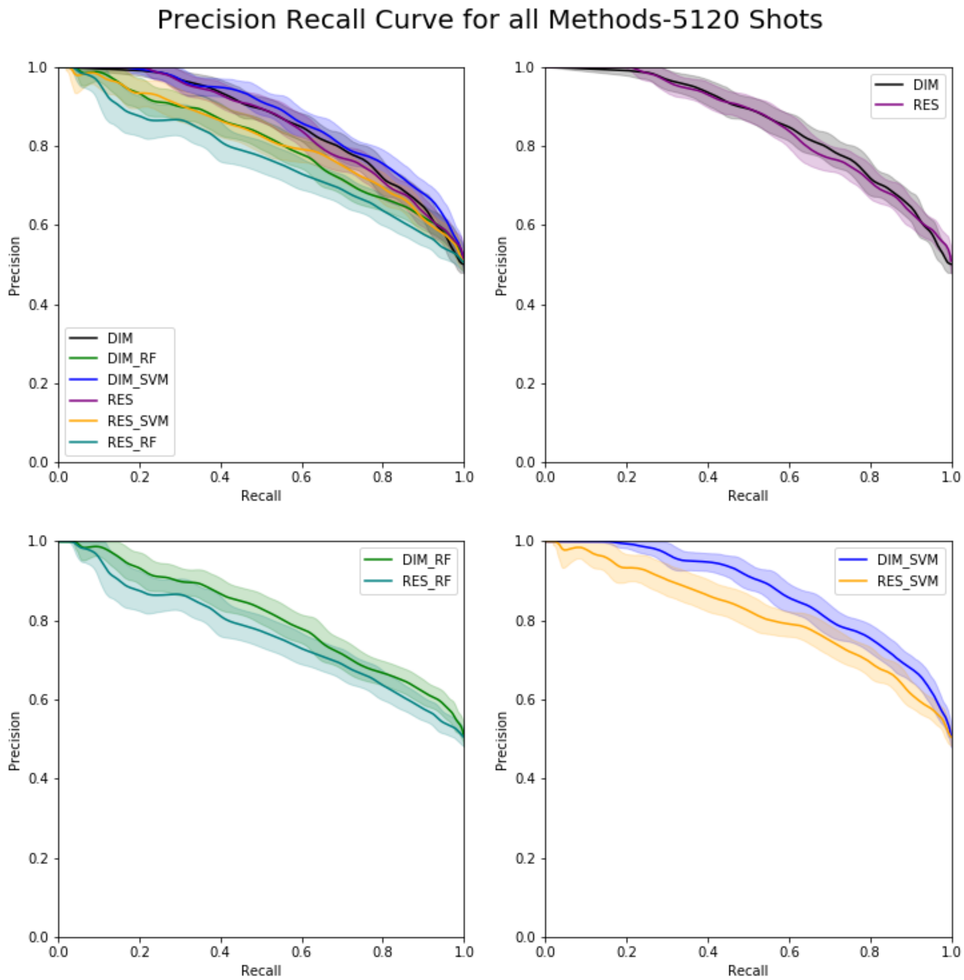
Precision Recall Curve for all Methods-40 Shots



eFigure 11. N=40 Shots Results: PR Curves and Confidence Intervals, (upper left) All Methods, (rest) Two by Two Comparison of Methods



eFigure 12. N=160 Shots Results: PR Curves and Confidence Intervals, (upper left) All Methods, (rest) Two by Two Comparison of Methods



eFigure 13. N=5120 Shots Results: PR Curves and Confidence Intervals, (upper left) All Methods, (rest) Two by Two Comparison of Methods