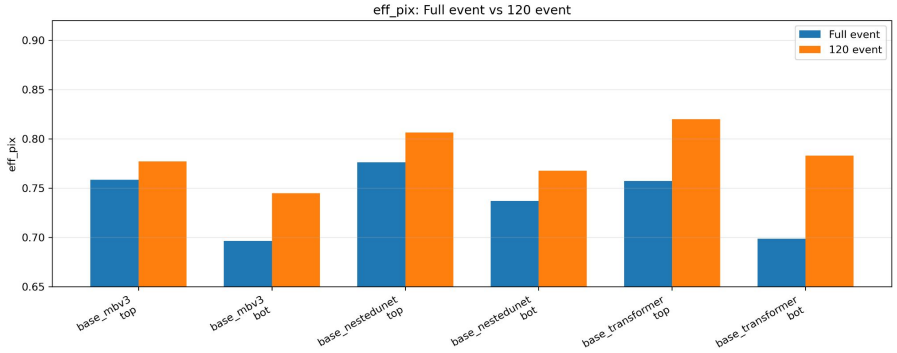
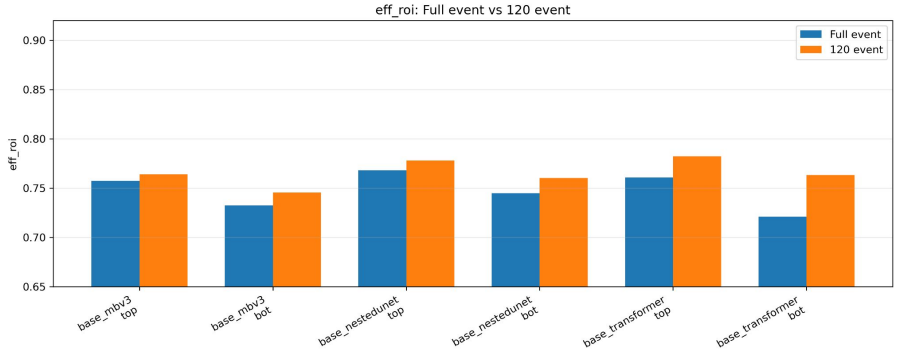
Status report on

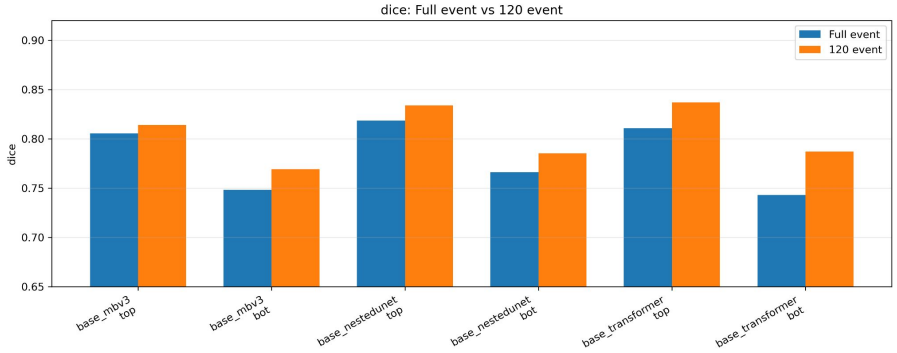
DNNROI sigproc

Hokyeong Nam
Chung-Ang University

PDVD - Full Dataset vs. Small Dataset

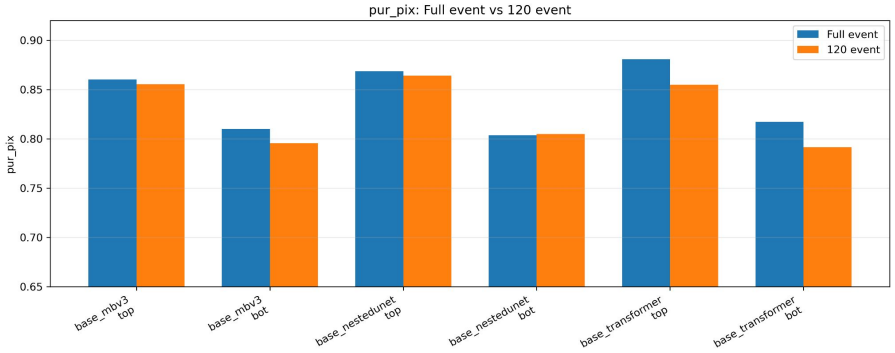


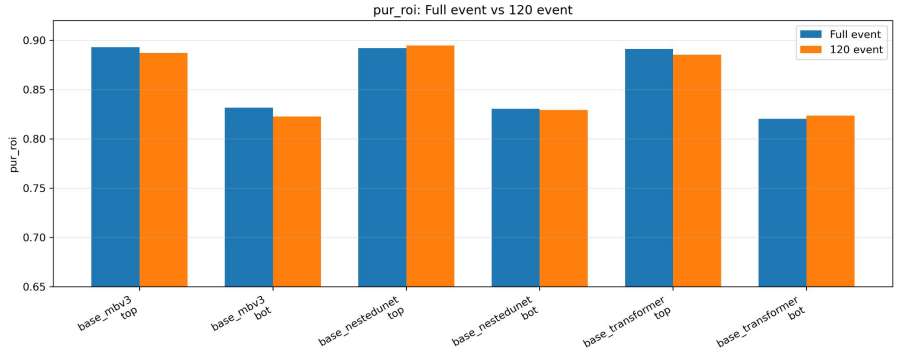




- All the results are using 6ch inputs
- Small datasets (120 events) tend to have better performance in validation dice, pixel efficiency, roi efficiency
- This shows better consistency with Xin's unified vs split study

PDVD - Full Dataset vs. Small Dataset





- For the both pixel and roi purities, full dataset results showed similar or slightly improved performance than small dataset results

model	test_set	Δeff_pix	Δpur_pix	Δeff_roi	Δpur_roi	Δdice
base_mbv3	top	-0.0185	+0.0047	-0.0066	+0.0058	-0.0085
base_mbv3	bot	-0.0486	+0.0144	-0.0131	+0.0088	-0.0210
base_nestedunet	top	-0.0301	+0.0048	-0.0099	-0.0025	-0.0154
base_nestedunet	bot	-0.0307	-0.0013	-0.0152	+0.0013	-0.0192
base_transformer	top	-0.0628	+0.0258	-0.0214	+0.0059	-0.0262
base_transformer	bot	-0.0845	+0.0258	-0.0422	-0.0033	-0.0441

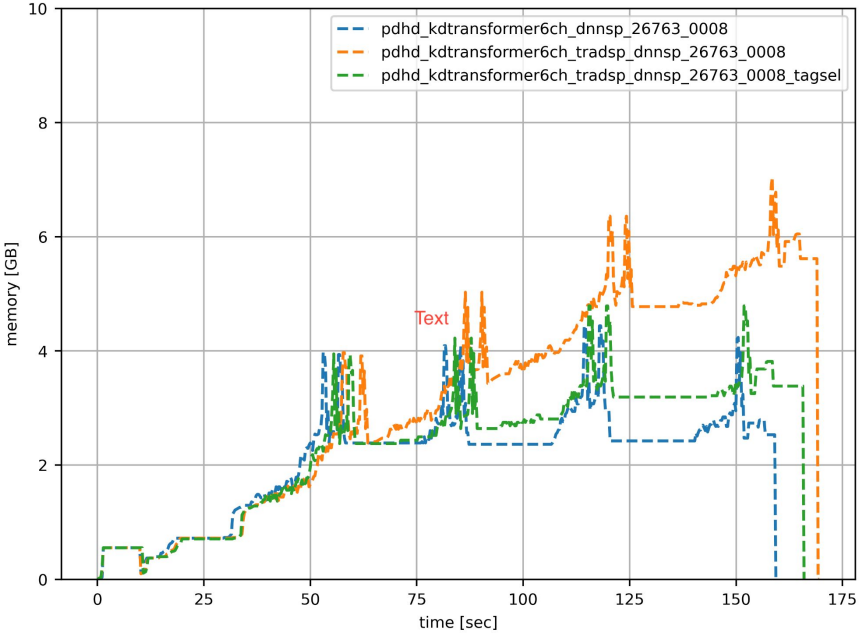
PDVD - Knowledge Distillation Validation

Run	precision	Dice	eff_roi	pur_roi
pdvd_mobilenetv3_all_6ch	FP32	0.7538	0.7135	0.8594
pdvd_teacher_nestedunet_6ch	FP32	0.7818	0.7429	0.8570
pdvd_teacher_transformer_6ch	FP32	0.7733	0.7385	0.8485
pdvd_distill_nestedunet_6ch	FP32	0.7816	0.7520	0.8537
pdvd_distill_transformer_6ch	FP32	0.7680	0.7437	0.8506

- Baseline mobileunet model I obtained (Bottom) has better roi efficiency than Xin's model (Top), still lower than the teacher model

Run	precision	Dice	eff_roi	pur_roi
pdvd_base_mbv3	FP32	0.7762	0.7505	0.8521
pdvd_base_mbv3_trial2	FP32	0.7778	0.7552	0.8444
pdvd_base_nestedunet	FP32	0.7953	0.7634	0.8537
pdvd_base_transformer	FP32	0.7858	0.7551	0.8512
pdvd_kd_mbv3_from_nestedunet	FP32	0.7662	0.7462	0.8504
pdvd_kd_mbv3_from_transformer	FP32	0.7709	0.7487	0.8494

PDHD Memory Usage



- Tested within the real data processing workflow
- Two options
  - keep DNN + Traditional ROI result with/without tagselector
  - keep Traditional ROI result only
- Peak memory usage is about 5 GB, higher than 3ch model's memory consumption, but acceptable according to Jake

Back Up

Workflow Validation

Cross-evaluation matrix (held-out test sets)

model	test set	n	eff_pix	pur_pix	eff_roi	pur_roi	dice
M1	top	120	0.7631	0.8618	0.7543	0.8967	0.8089
M1	bot	120	0.7281	0.8033	0.7395	0.8291	0.7636
M2	top	120	0.7141	0.8629	0.7372	0.8911	0.7793
M2	bot	120	0.6453	0.8137	0.7034	0.8285	0.7116

model	test set	n	eff_pix	pur_pix	eff_roi	pur_roi	dice
M1	top	120	0.7622	0.8569	0.7546	0.8923	0.8059
M1	bot	120	0.7188	0.8046	0.7342	0.8289	0.7587
M2	top	120	0.6813	0.8764	0.7358	0.8991	0.7656
M2	bot	120	0.6422	0.8172	0.7136	0.8343	0.7178

- Used existing M1 and M2 datasets in Xin's directory
- Created held-out test sets for this time
- The performance is close to Xin's results
- Iteration showed some fluctuations

3ch Results

model	test set	eff_pix	pur_pix	eff_roi	pur_roi	dice
base_mbv3_3ch	top	0.7424	0.8512	0.7498	0.8779	0.7881
base_mbv3_3ch	bot	0.6856	0.7946	0.7193	0.8215	0.7279
base_nestedunet_3ch	top	0.7804	0.8640	0.7699	0.8814	0.8178
base_nestedunet_3ch	bot	0.7456	0.7969	0.7462	0.8183	0.7677
base_transformer_3ch	top	0.7213	0.8803	0.7359	0.8948	0.7804
base_transformer_3ch	bot	0.6907	0.8171	0.7258	0.8277	0.7382
kd_mbv3_from_nestedunet_3ch	top	0.7106	0.8661	0.7429	0.8867	0.7756
kd_mbv3_from_nestedunet_3ch	bot	0.6613	0.8047	0.7164	0.8235	0.7166
kd_mbv3_from_transformer_3ch	top	0.6973	0.8687	0.7310	0.8886	0.7626
kd_mbv3_from_transformer_3ch	bot	0.6504	0.8087	0.7073	0.8253	0.7079

- Overall, the top test set consistently shows better performance than bottom
- No clear benefit from KD implementation

6ch Results

model	test set	eff_pix	pur_pix	eff_roi	pur_roi	dice
base_mbv3_6ch	top	0.7586	0.8602	0.7576	0.8931	0.8056
base_mbv3_6ch	bot	0.6964	0.8100	0.7326	0.8315	0.7483
base_nestedunet_6ch	top	0.7763	0.8688	0.7682	0.8920	0.8184
base_nestedunet_6ch	bot	0.7371	0.8035	0.7450	0.8305	0.7663
base_transformer_6ch	top	0.7573	0.8808	0.7609	0.8913	0.8107
base_transformer_6ch	bot	0.6987	0.8173	0.7212	0.8203	0.7431
kd_mbv3_from_nestedunet_6ch	top	0.7262	0.8656	0.7482	0.8912	0.7884
kd_mbv3_from_nestedunet_6ch	bot	0.6872	0.8061	0.7279	0.8305	0.7388
kd_mbv3_from_transformer_6ch	top	0.7370	0.8636	0.7556	0.8899	0.7935
kd_mbv3_from_transformer_6ch	bot	0.6958	0.8040	0.7273	0.8275	0.7428

- The 6ch results show modest but consistent improvements over the 3ch results
- No clear benefit from KD implementation