Week 14

- Figured out to run everything on the cluster and in the same environment
- Re-run the selector and re-generated the performance data
- Now, I normalized the data
- Noticed that BFGS does not stay within bound which results in a few problems
- So far, I tried clipping the values and also already run the selector for that case (poor performance :/)
- First, without clipping on the next slide

Data normalization

- I normalized the ELA features and CMA internal state features (including the standard deviations) per budget, feature-wise
- I also normalized the algorithm performances within (1e-12, 1) per budget and per instance across all algorithms

Performance with BFGS out of bounds (so as before)

- Selector still best
- However: No statistical significance
- (p-value of 0.17 compared to B48)
- Parameters for switching models not tuned

Method	Ratio
selector_precision	0.443708076240659
static_B80	0.4027017132661988
static_B48	0.3981219612897716
static_B64	0.39127223183729676
static_B96	0.3904859778817766
static_B72	0.39012462726286945
static_B100	0.38256529192637545
static_B88	0.37146583768630725
static_B56	0.35865374724761967
static_B150	0.3366742471042941
static_B500	0.31917538467783985
static_B200	0.31890591369106447

BFGS clipping

- Without clipping, BFGS always reaches target precision for fids 10 and 12
- With clipping, BFGS sometimes gets stuck at some border and is really really far away from the target precision (like 4000 or even more). This changes the performances drastically, as the sum of precisions is now dominated by these runs because the algorithm selectors usually choose BFGS on fid 10 and 12

Results

See fid_specific_switching/results/precision_ration_all_with_algos_clipped.pdf

- Performance is really poor
- However, this is largely due to one single bad prediction for fid 10 which alone has a precision of more than 4000! (Normally we aim for 2000 across the entire test set, the sbs has 3300).
- Its also interesting that now, switching later is generally better. This is because these really bad BFGS runs are only encountered if you switch early
- This should theoretically make our selector outperform the static ones even more because now its really important to also switch at later switching points for some functions
- The one bad prediction also happens when we switch earlier than the optimal switching point for that function
- Currently, I run parameter tuning for the switching models and I hope that this tuning will result in parameters that really make us avoid switching too early on fids 10 and 12

Questions

- In parameter tuning, I use the true precisions (so not the normalized ones) for the target function. Is that right? I really want to emphasize that we avoid really high precisions
- Is clipping for BFGS ideal? I also thought of trying mirroring but the improvements won’t be big. Which bound correction was used in the per-run paper?