Before (w/o nof\_batches\_normalization, w/step\_down, w/Kaimig)

Results: 40% Urban 22% Rural ~~~

> Try normalizing loss before backprop. Results: 42% Urban 23% Rural ~~ Nice

> **Try using polynomial update**. Results:

> Try Xavier instead of Kaimig. Results:

> Try tweaking **learning rates** (**mainly discriminator** but also main one)

> Try tweaking LAMBDA\_SEG: 0.1 seems to be better but idk, they’re really similar

> Try tweaking **LAMBDA\_REV**:

> Try tweaking LAMBDA\_DISC: 0.1 seems to be better than 0.5

> Try using Adam on both or SGD on both

> Try backpropagating only twice instead of thrice (loss\_pidnet and loss\_adv together)

> optimizer.zero\_grad() and optimizer.step() at each epoch instead of each batch: Nice

> TODO: Test the Discriminator model too (implement: Save, Validate, and Test (test is 50% done))

> Understand if using

output = outputs[1].detach()

output\_t = output\_t.detach()

and

for param in model.parameters():

param.requires\_grad = False

Is the same and what should we use (cause our results were actually different in the two configs. The first yielded 30% and 19%, the second 27% and 20%)