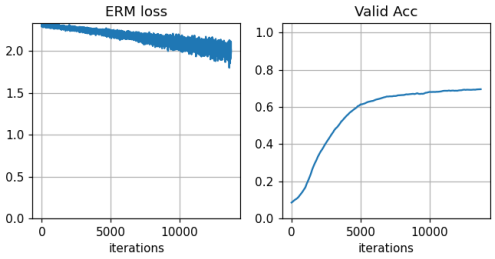
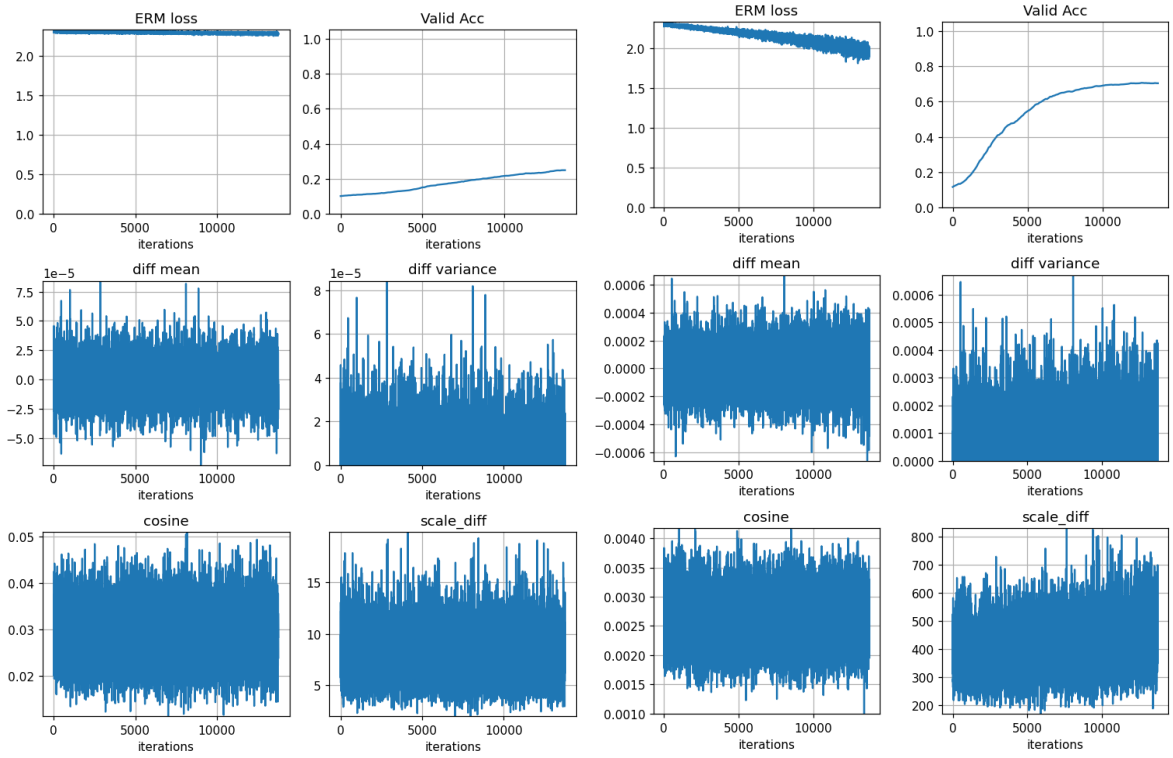
Using inputs to generate random vectors

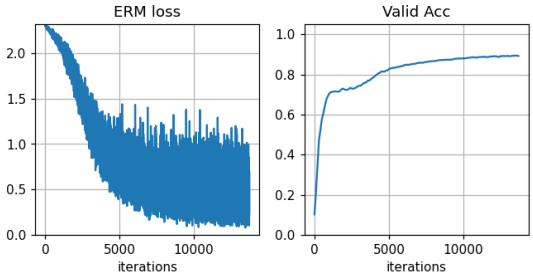
Batch_size = num_dirs = 20, 5 epochs

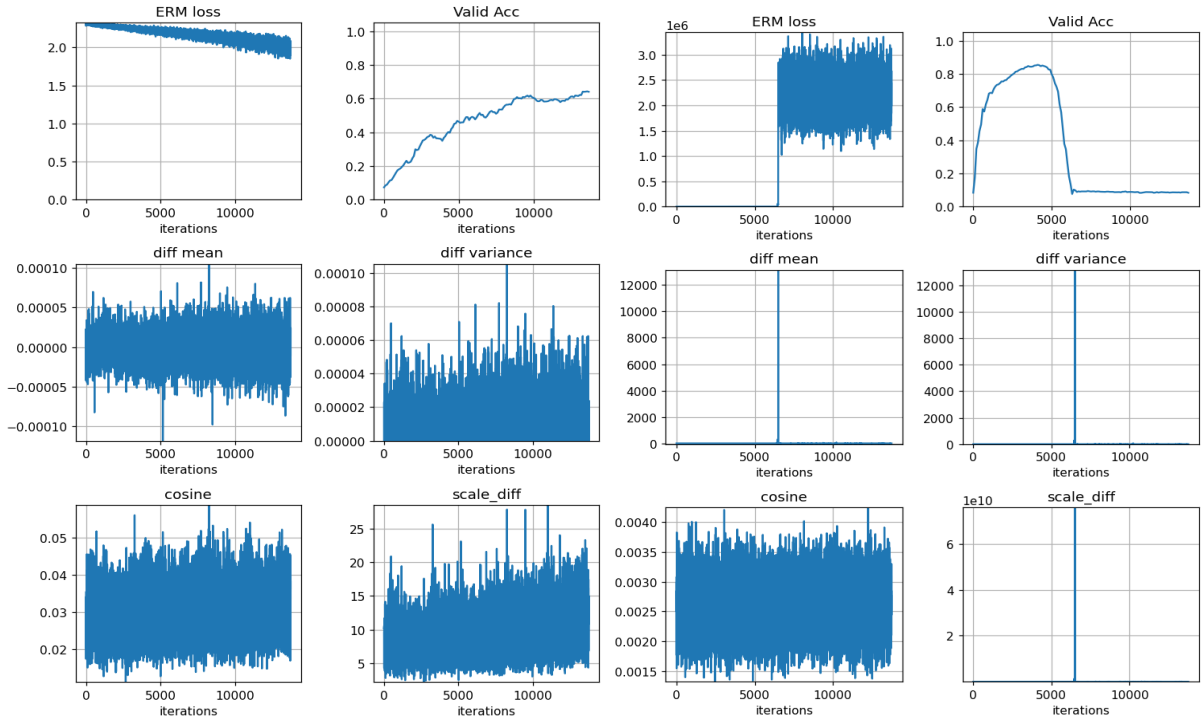
Lr = 1e-4, backprop/using_input/naïve_forward:





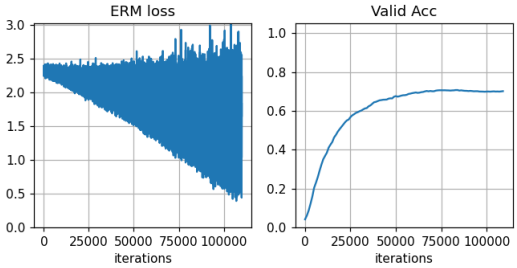
Lr = 1e-3, backprop/using_input/naïve_forward:

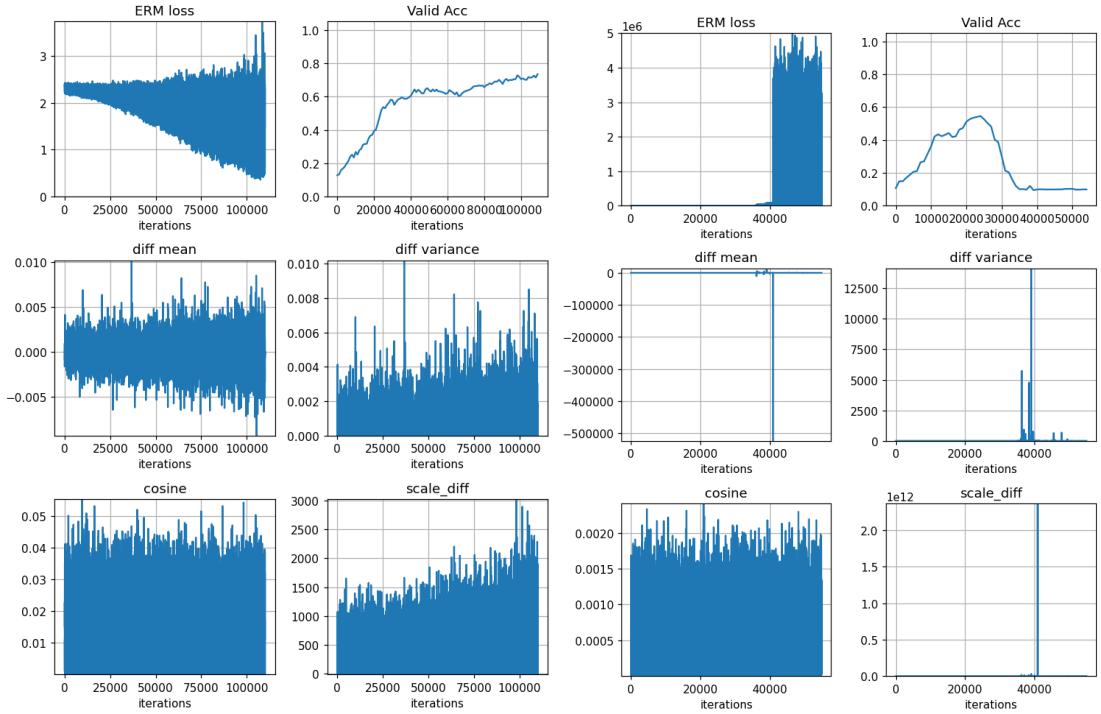




Batch_size = num_dirs = 1, 2 epochs (1 for naïve forward because it diverges)

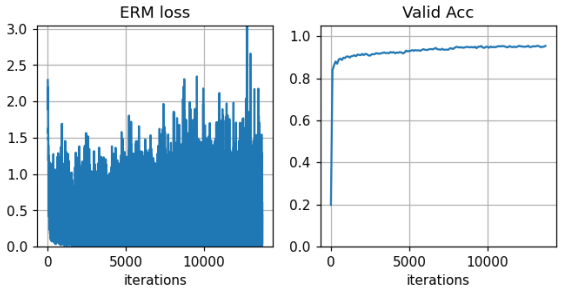
Lr = 2e-5, backprop/using_input/naïve_forward:

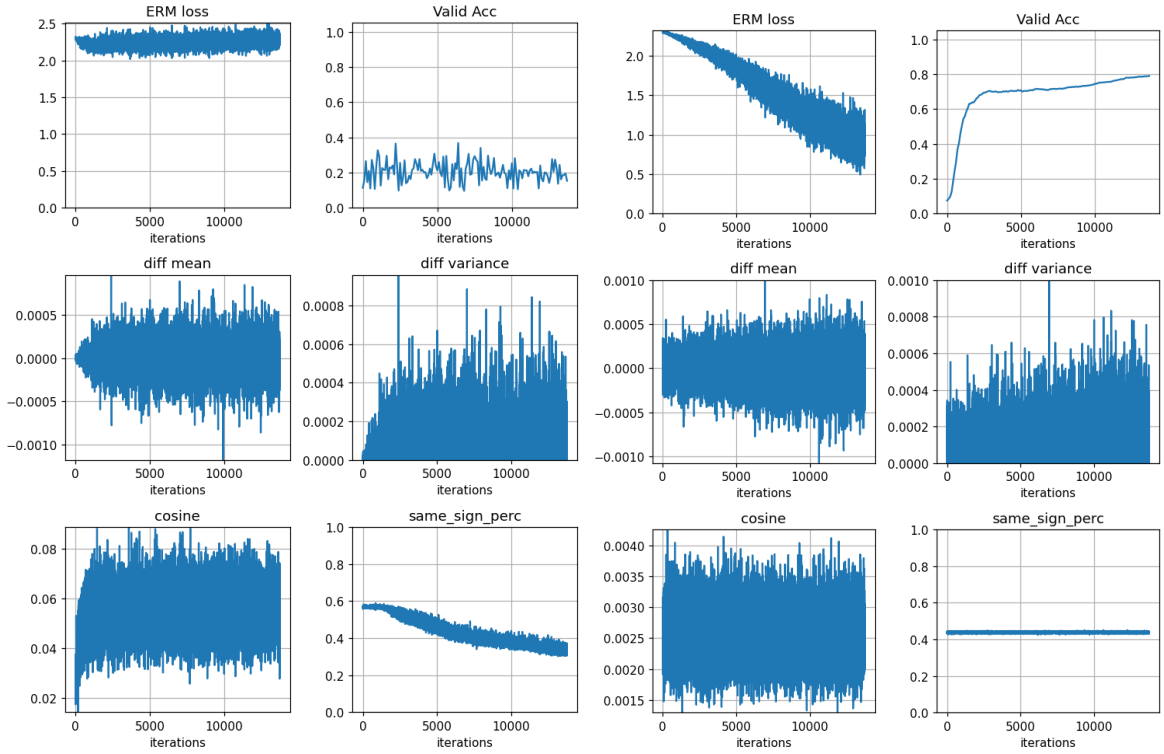




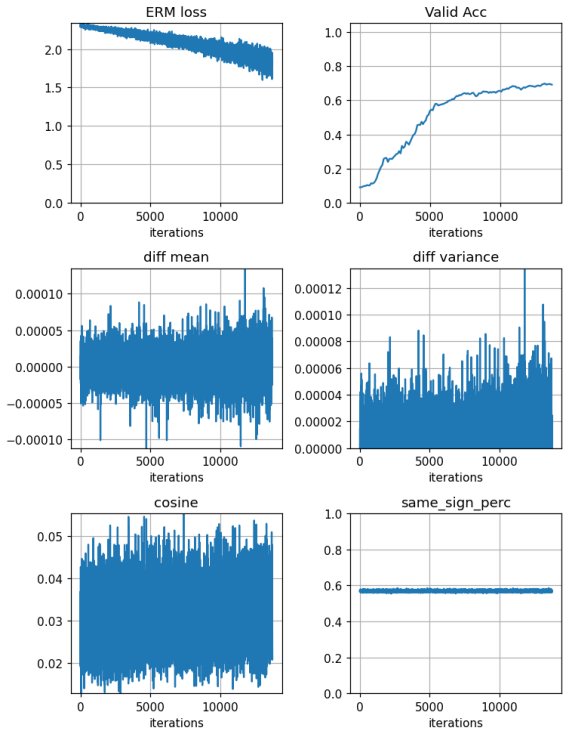
Batch_size = num_dirs = 20, 5 epochs

Lr = 1e-4, backprop/using_input/naïve_forward, using sign_sgd:

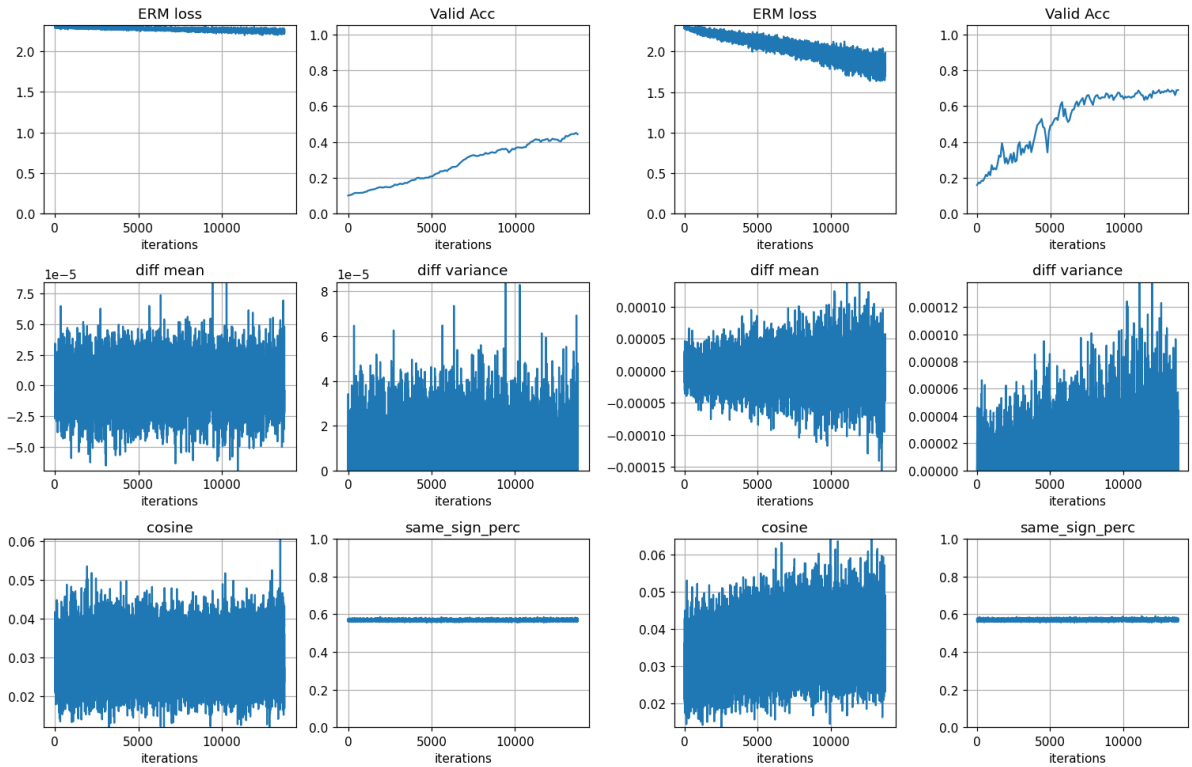


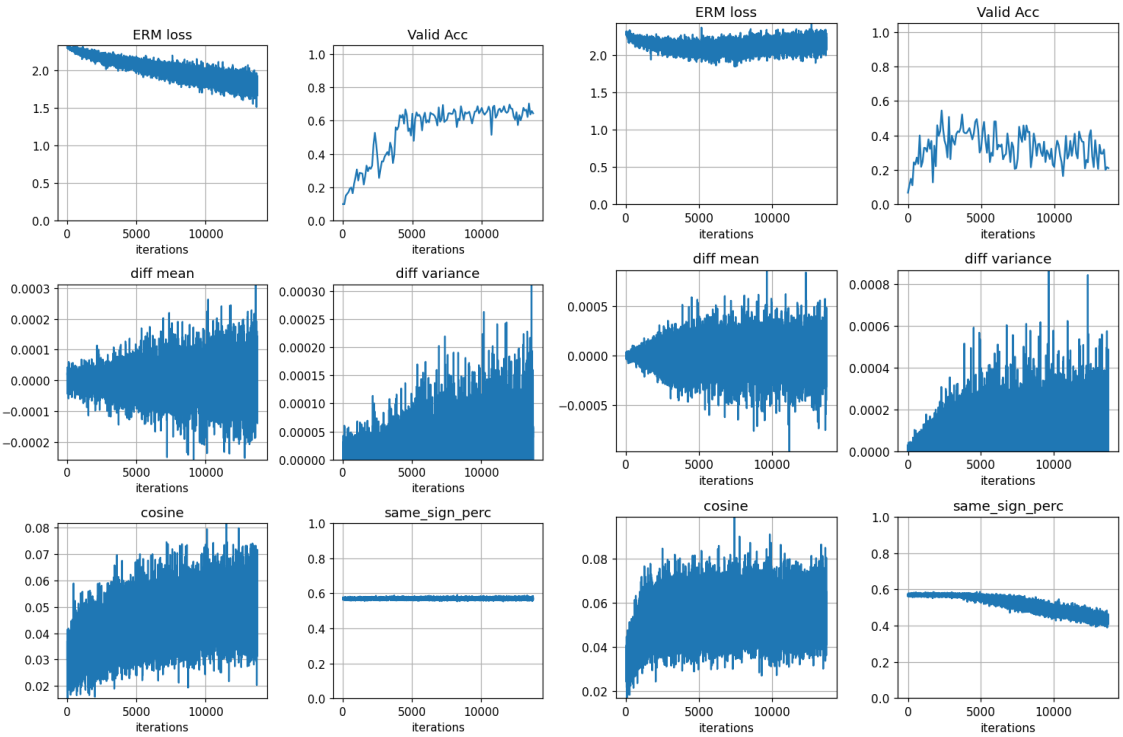


Decrease the lr for using_input method to 1e-5



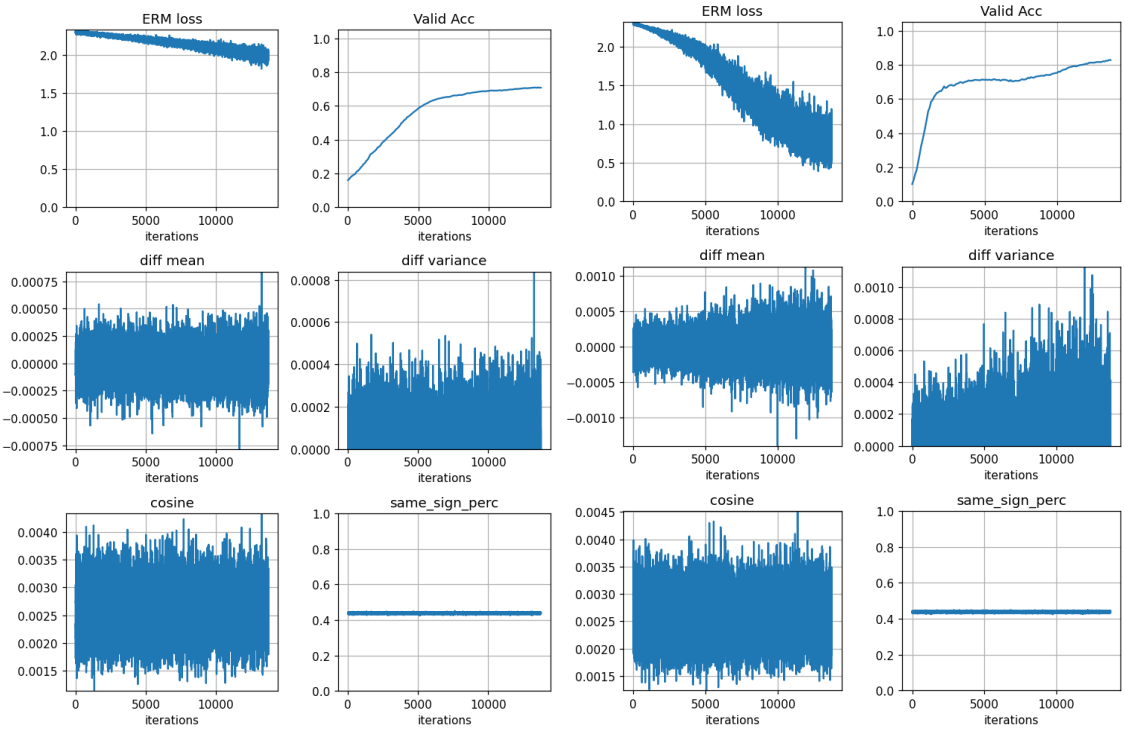
If we still use 1e-4 as learning rate and suppress the small values in estimation: stronger/weaker suppression (1.5, 3, 5, 10)





Let’s forger this method for a sec! What if we suppress the small noise and enlarge the large dimension in naïve form of forward mode?

Lr = 1e-4



VGG on Cifar10:

Lr = 1e-4, naïve forward-mode / suppress&&enlarge: