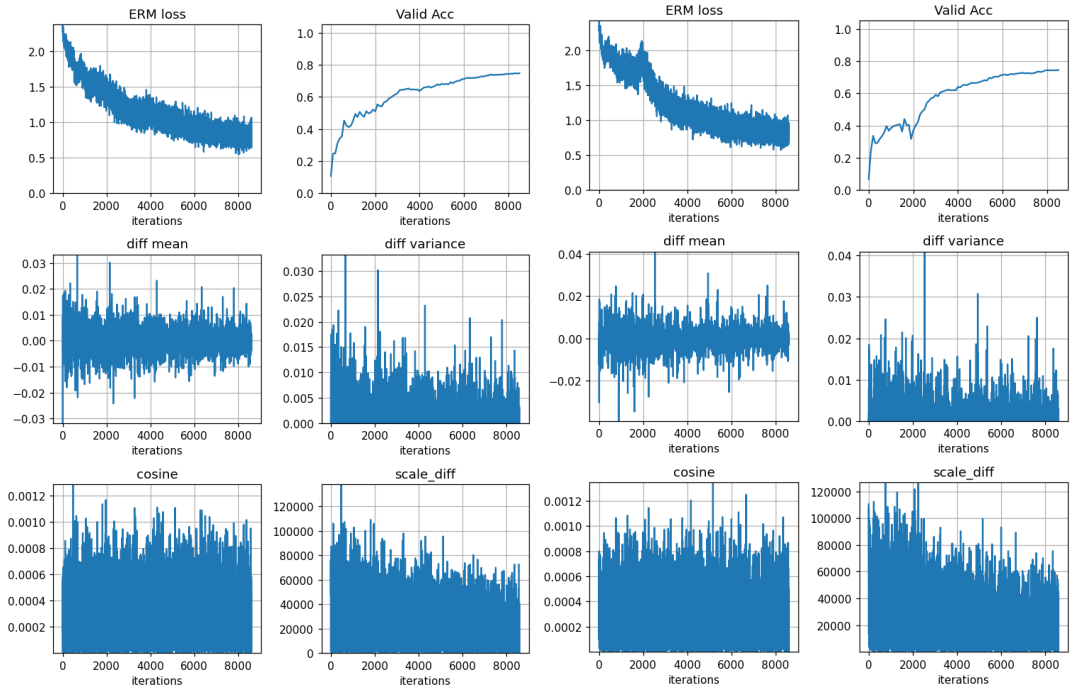
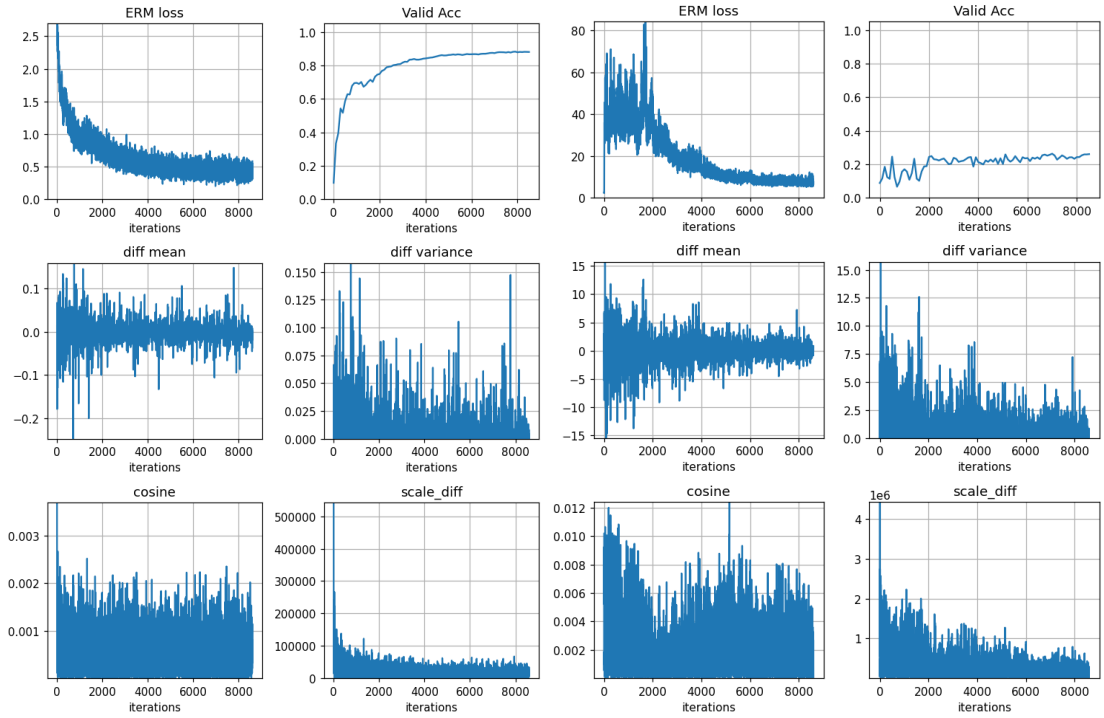
B = 128

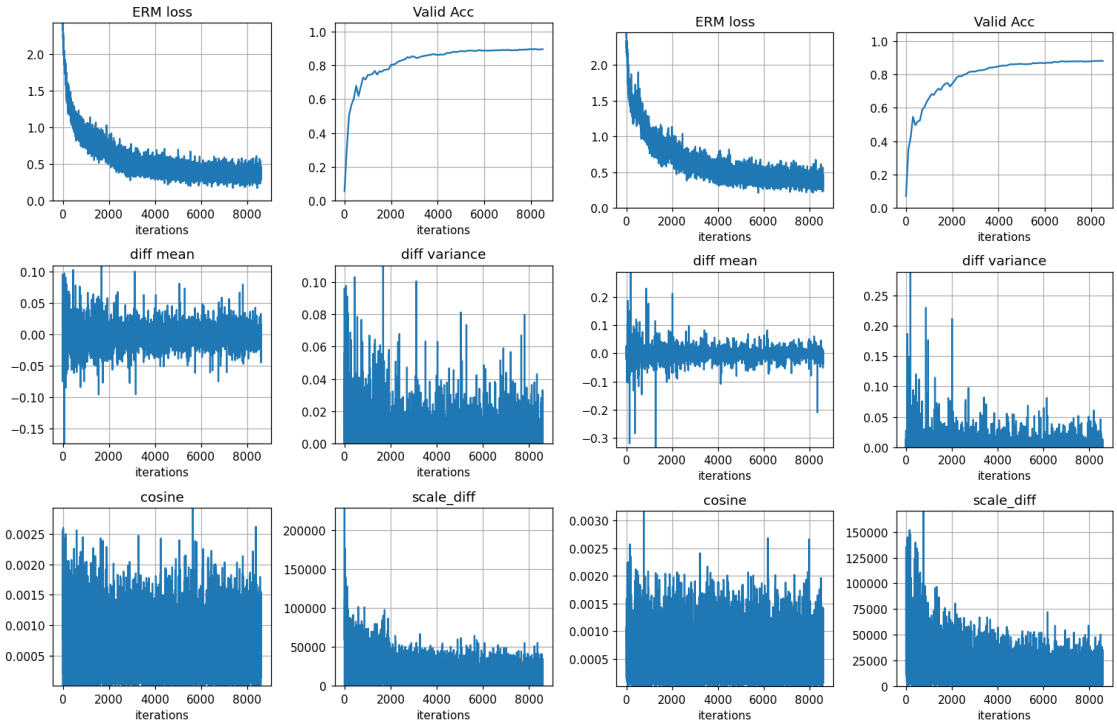
Naïve: 0.000228; version 0 : 0.000231



Version 1: 0.000424;   version2: 0.00158

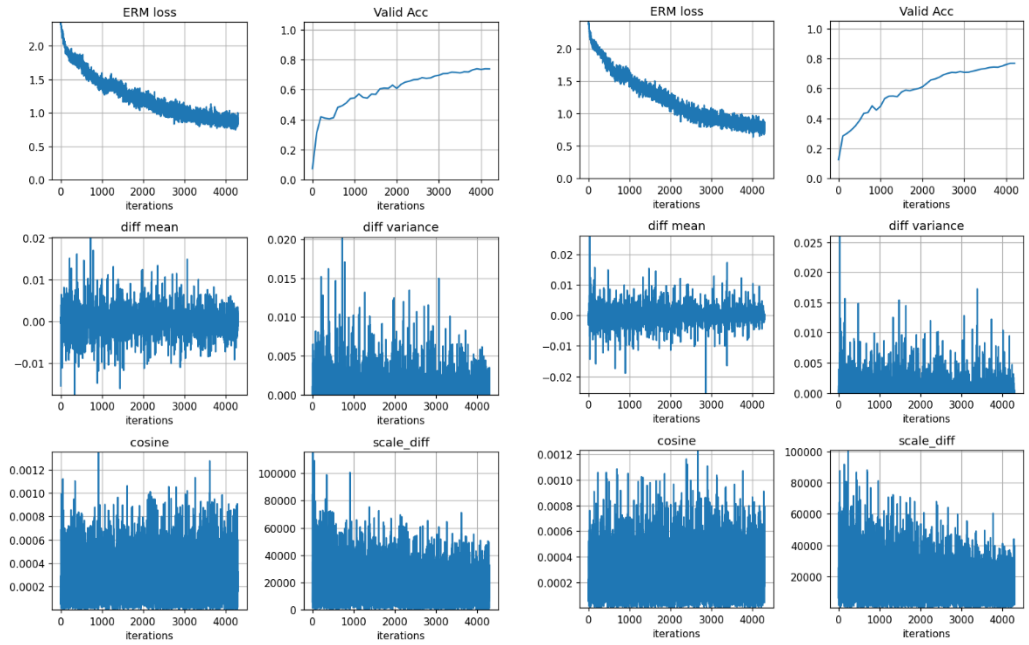


Version 3: 0.000469; Version 4: 0.000438

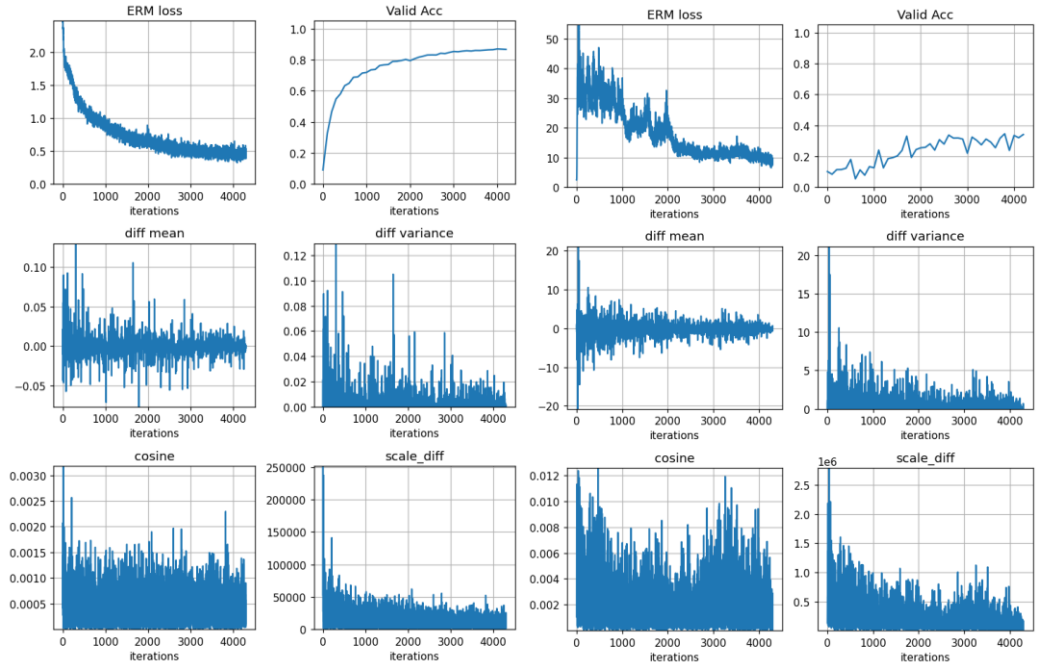


B = 256

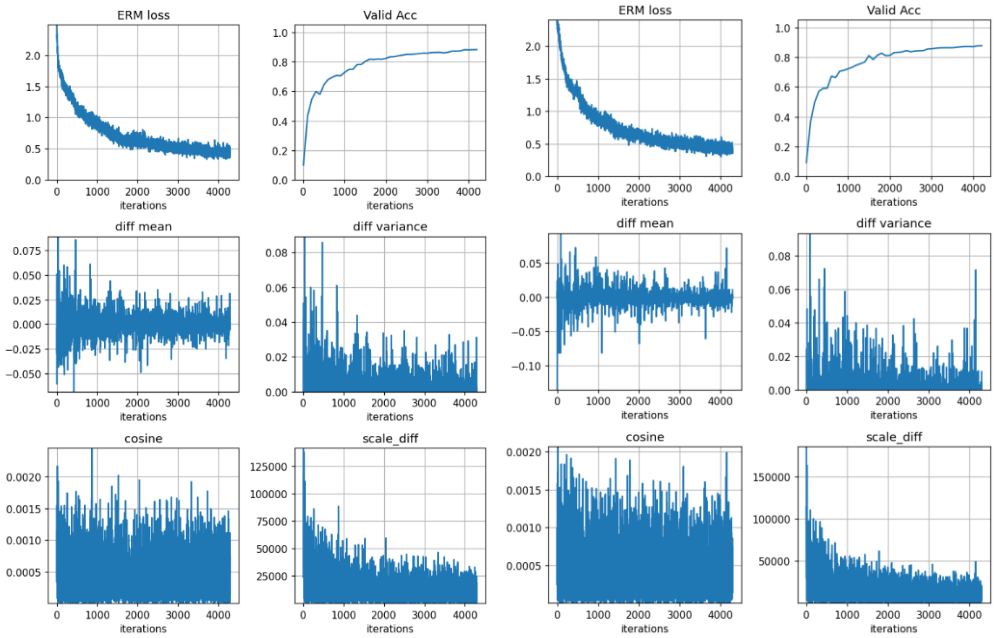
Naïve: 0.000222; version 0 : 0.000230



Version 1: 0.000360;   version2: 0.00181:

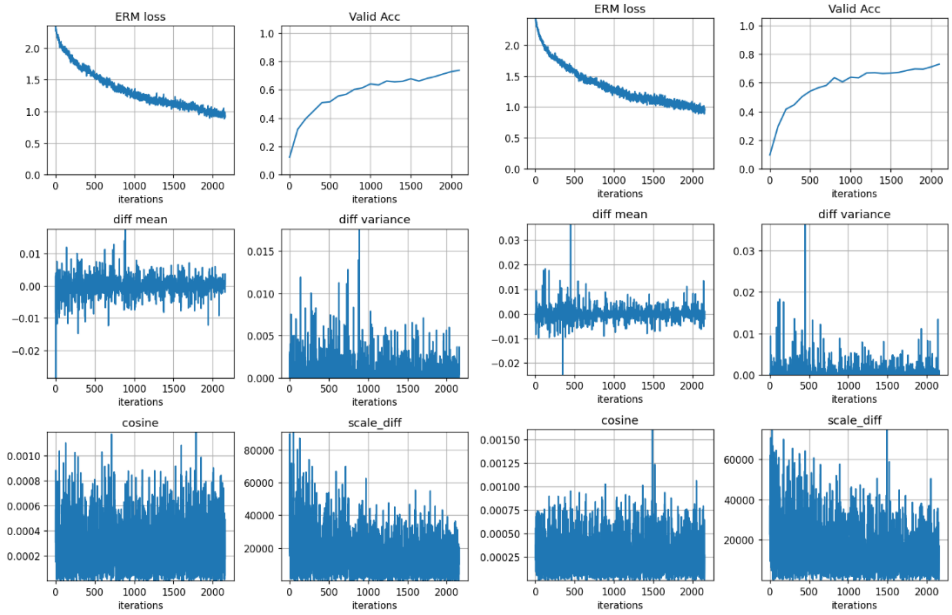


Version 3: 0.000379; Version 4: 0.000365

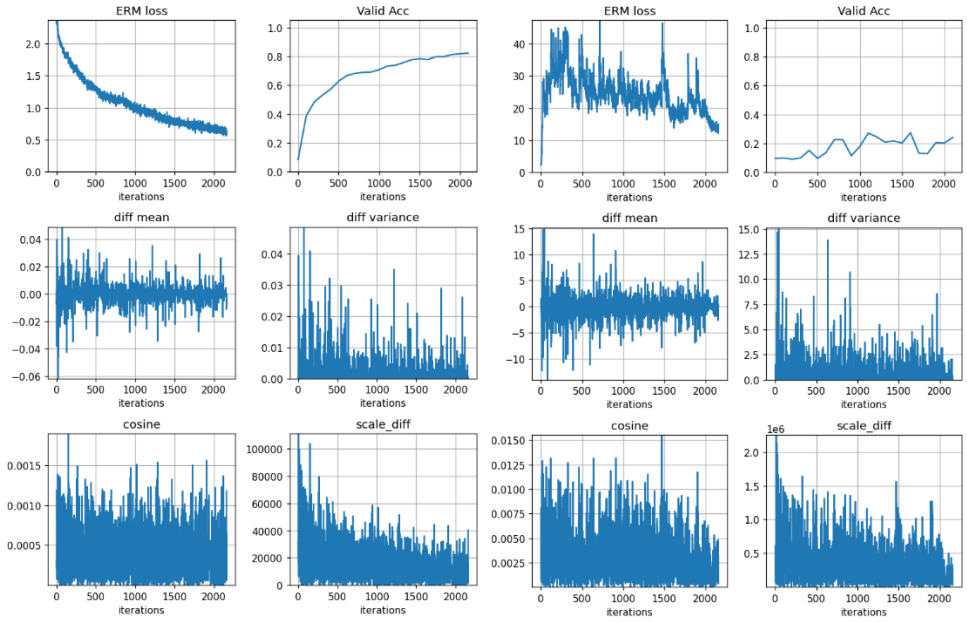


B = 512

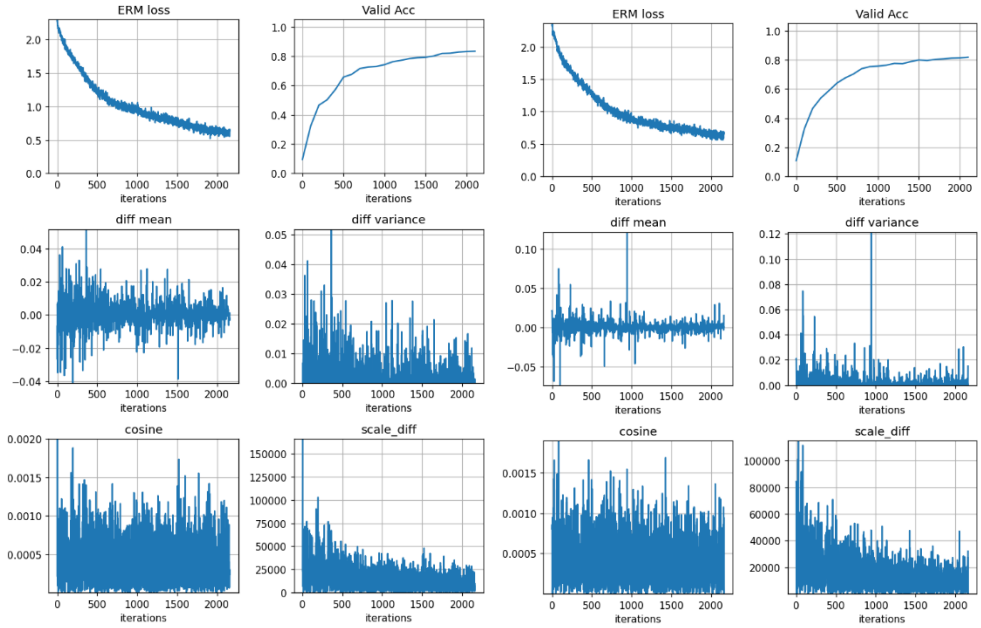
Naïve: 0.000221; version 0 : 0.000228:



Version 1: 0.000312; version2: 0.00249:

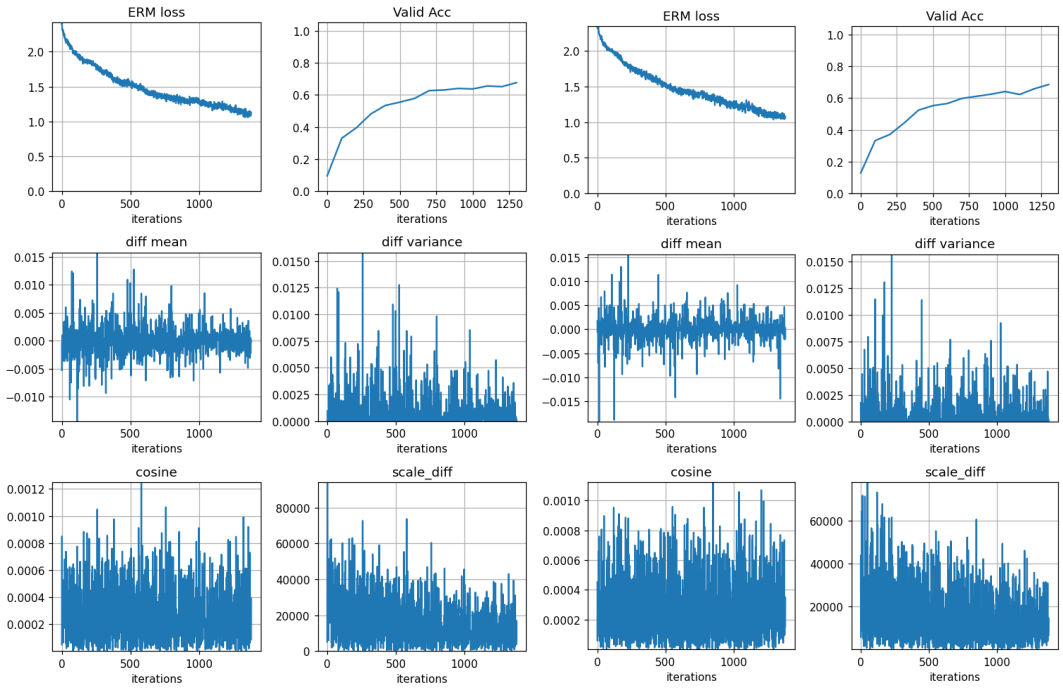


Version 3: 0.000342; Version 4: 0.000334

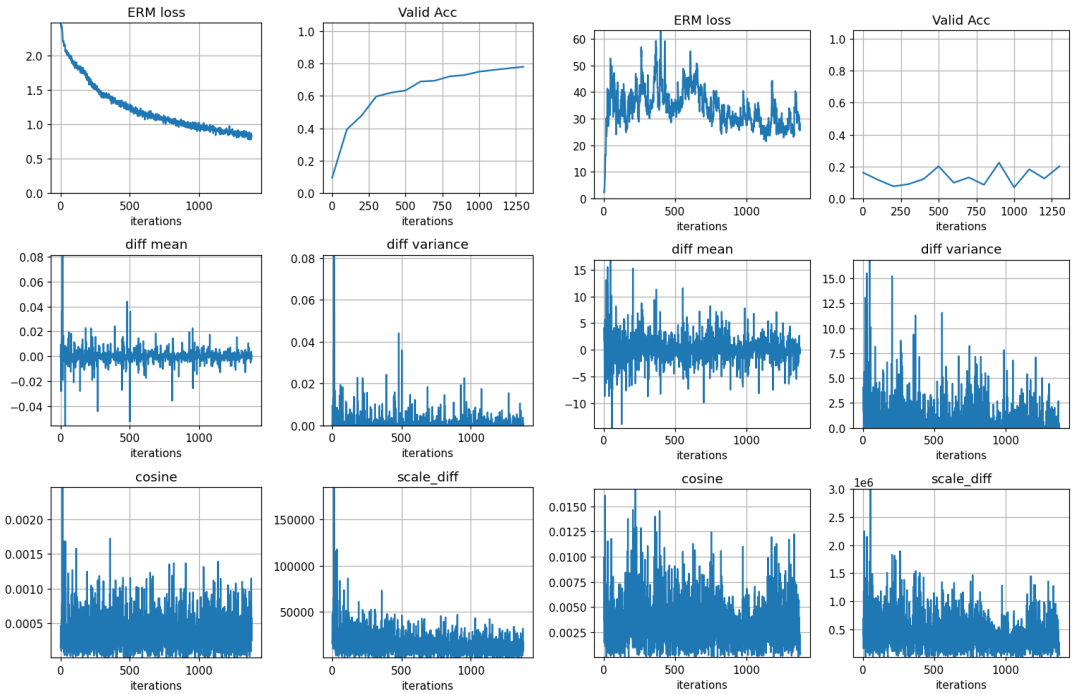


B = 800

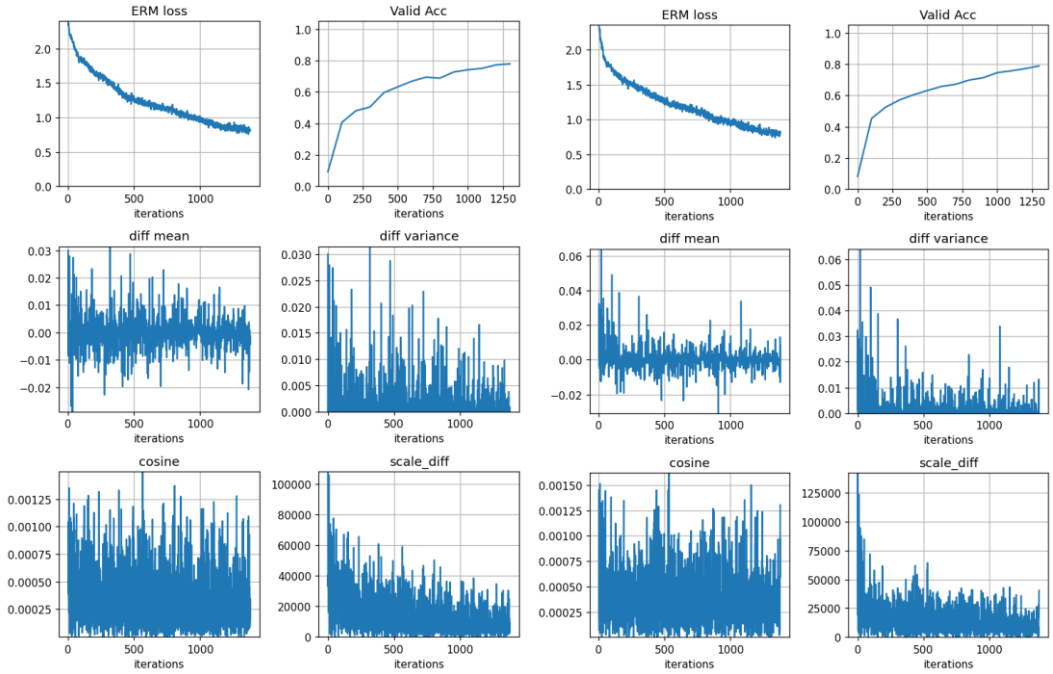
Naïve: 0.000235; version 0 : 0.000235:



Version 1: 0.000300;   version2: 0.00291:

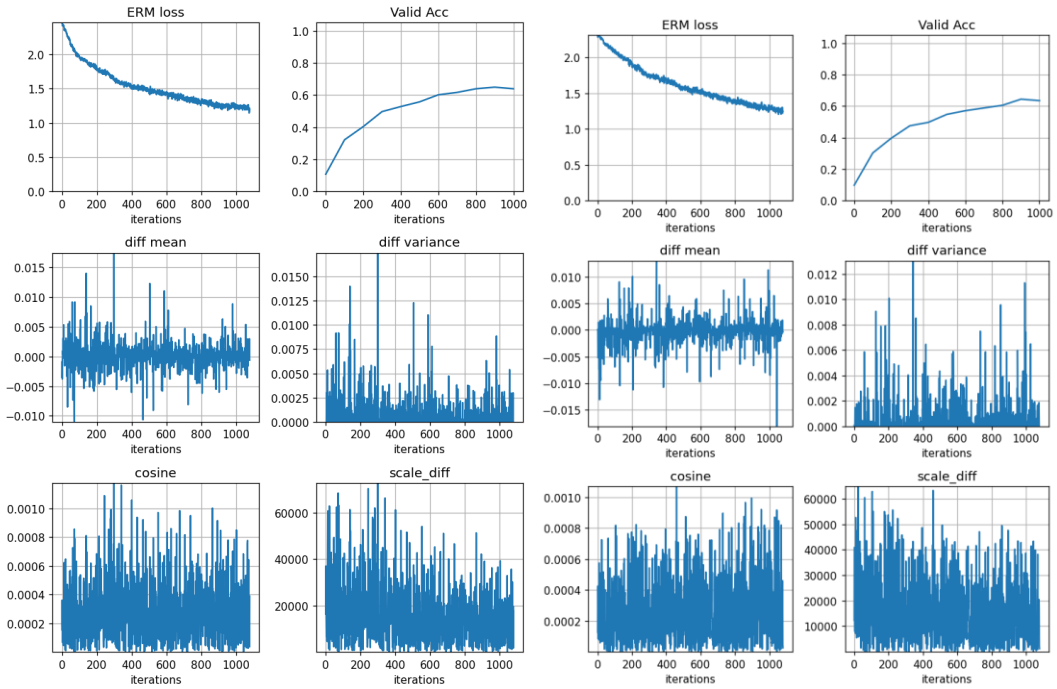


Version 3: 0.000309; Version 4: 0.000303

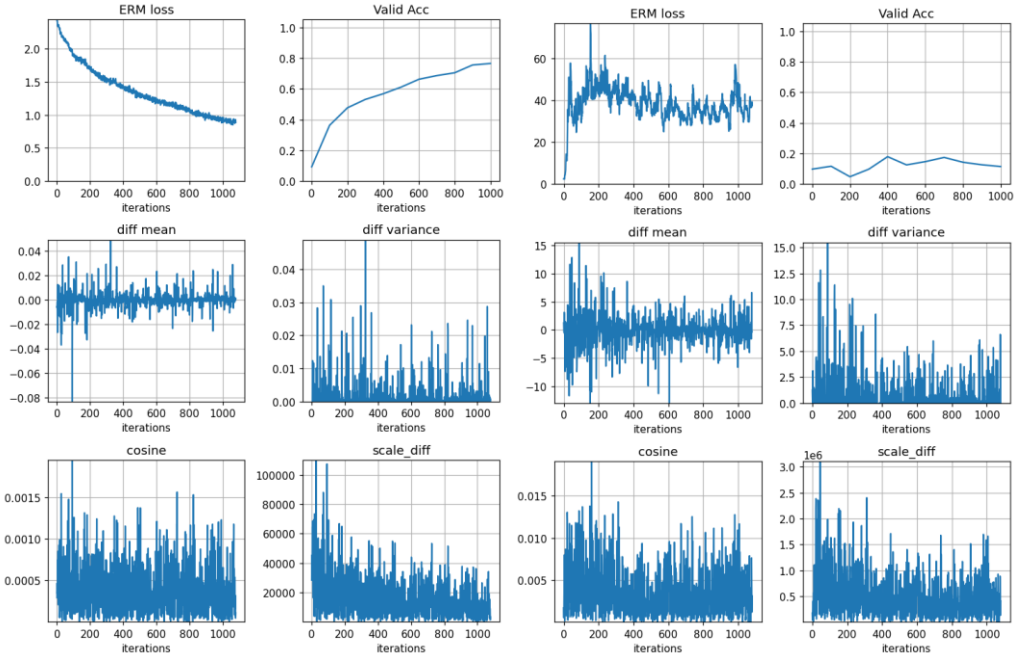


B = 1024

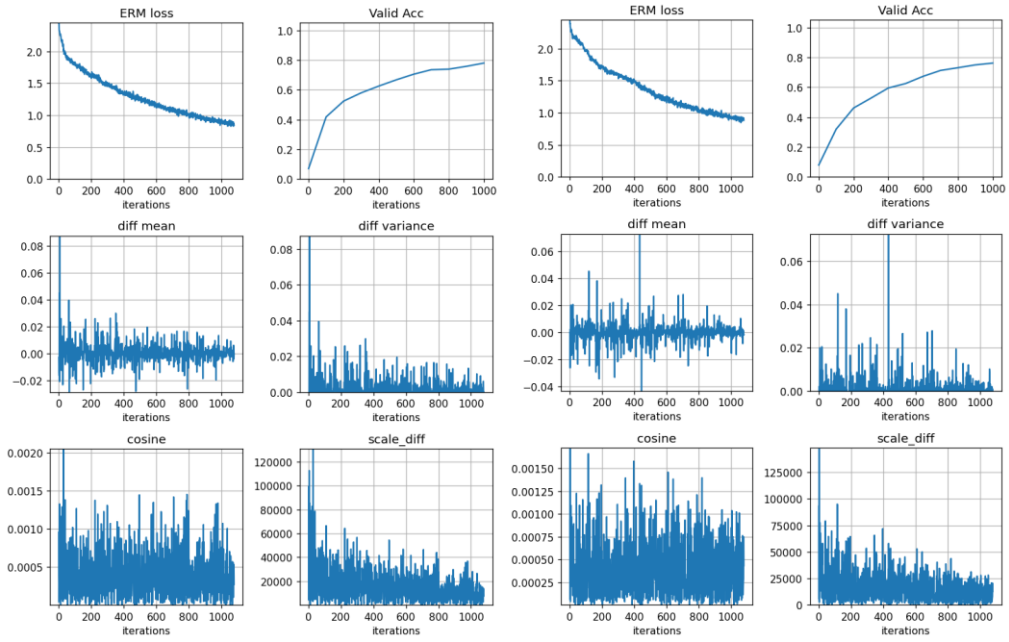
Naïve: 0.000229; version 0 : 0.000208:



Version 1: 0.000292; version2: 0.00284:



Version 3: 0.000299; Version 4: 0.000281



	naive	V0	V1	V2	V3	V4
128	0.000228	0.000231	0.000424	0.00158	0.000469	0.000438
256	0.000222	0.000230	0.000360	0.00181	0.000379	0.000365
512	0.000221	0.00228	0.000312	0.00249	0.000342	0.000334
800	0.000235	0.000235	0.000300	0.00291	0.000309	0.000303
1024	0.000229	0.000208	0.000292	0.00284	0.000299	0.000281

Performance:

V2< naïve = V0 < V1 = V4 < V3

V1, V3, V4 all decrease with B increase,

V2 very special, very large cosine similarity but very bad performance. Also cosine similarity increase with B increase

For now, V3 is the best.

V1 and V2's cosine similarity trend change:

For V2: A bug, sorry

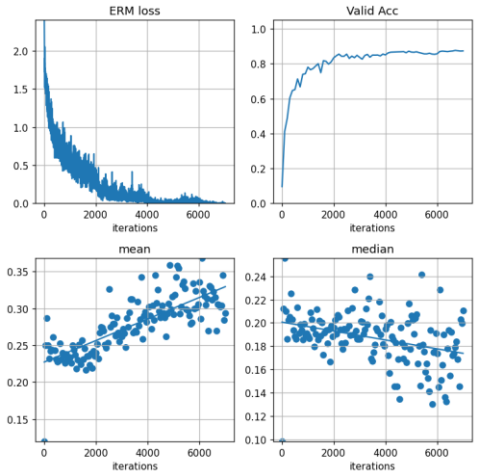
For V1: Just a guess. Change of scale (which shouldn't effect the cosine similarity which is the direction), but unlike other method, convergence performance effect the direction of this method

Some random exploration:

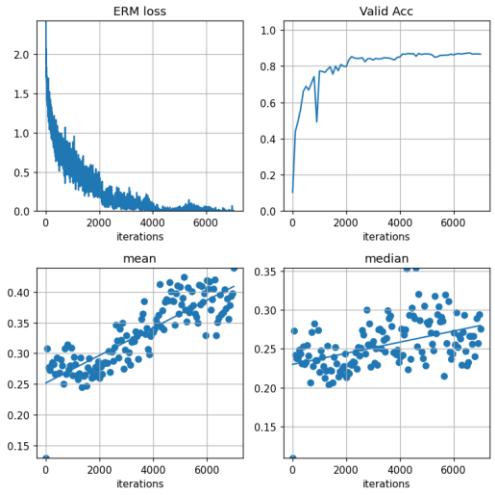
Row-wise correlation of the gradient matrix

Cifar-10:

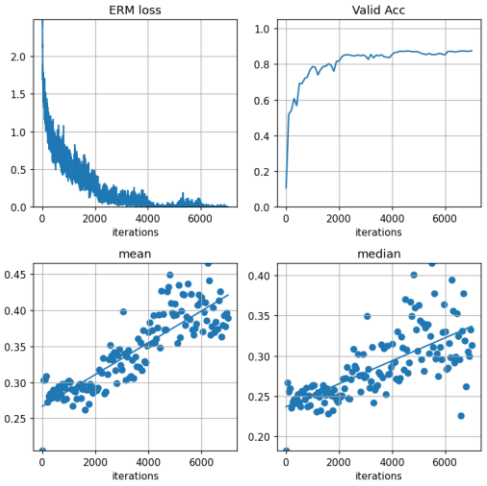
Layer 0:



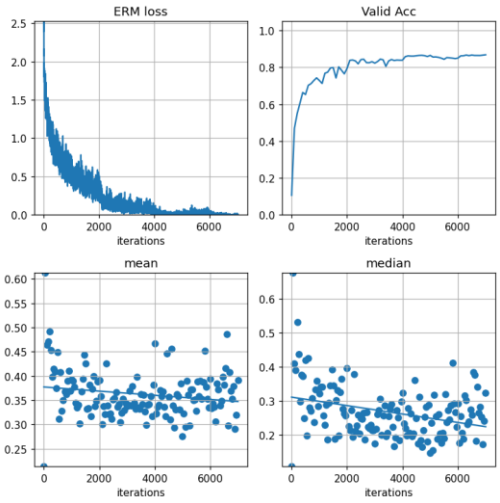
Layer 1:



Layer 2:

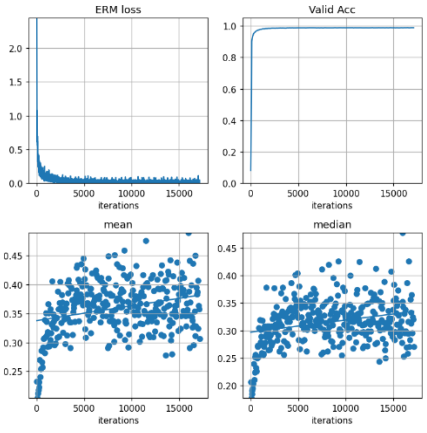


Layer 3:

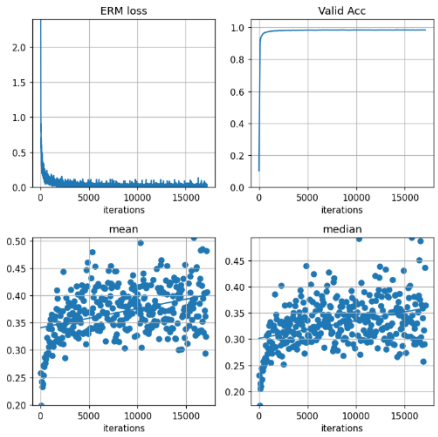


Mnist:

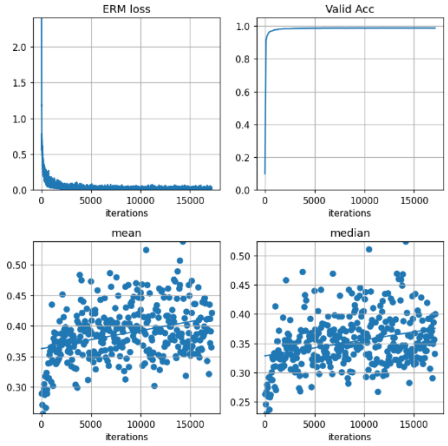
Layer 0:



Layer 1:



Layer 2:



Layer 3: