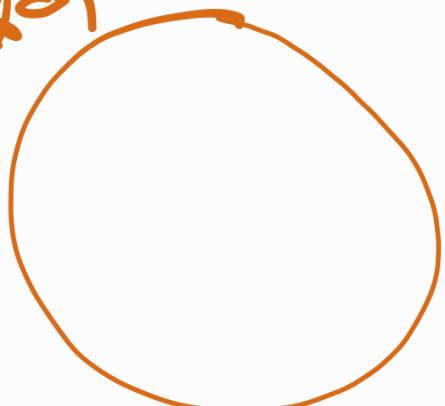
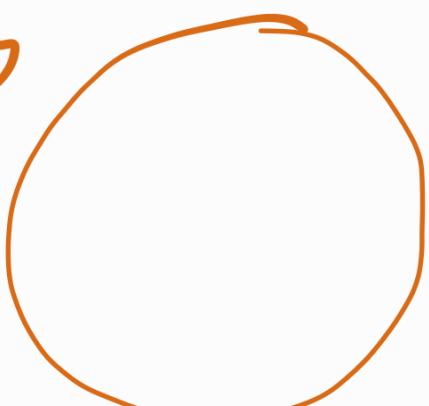


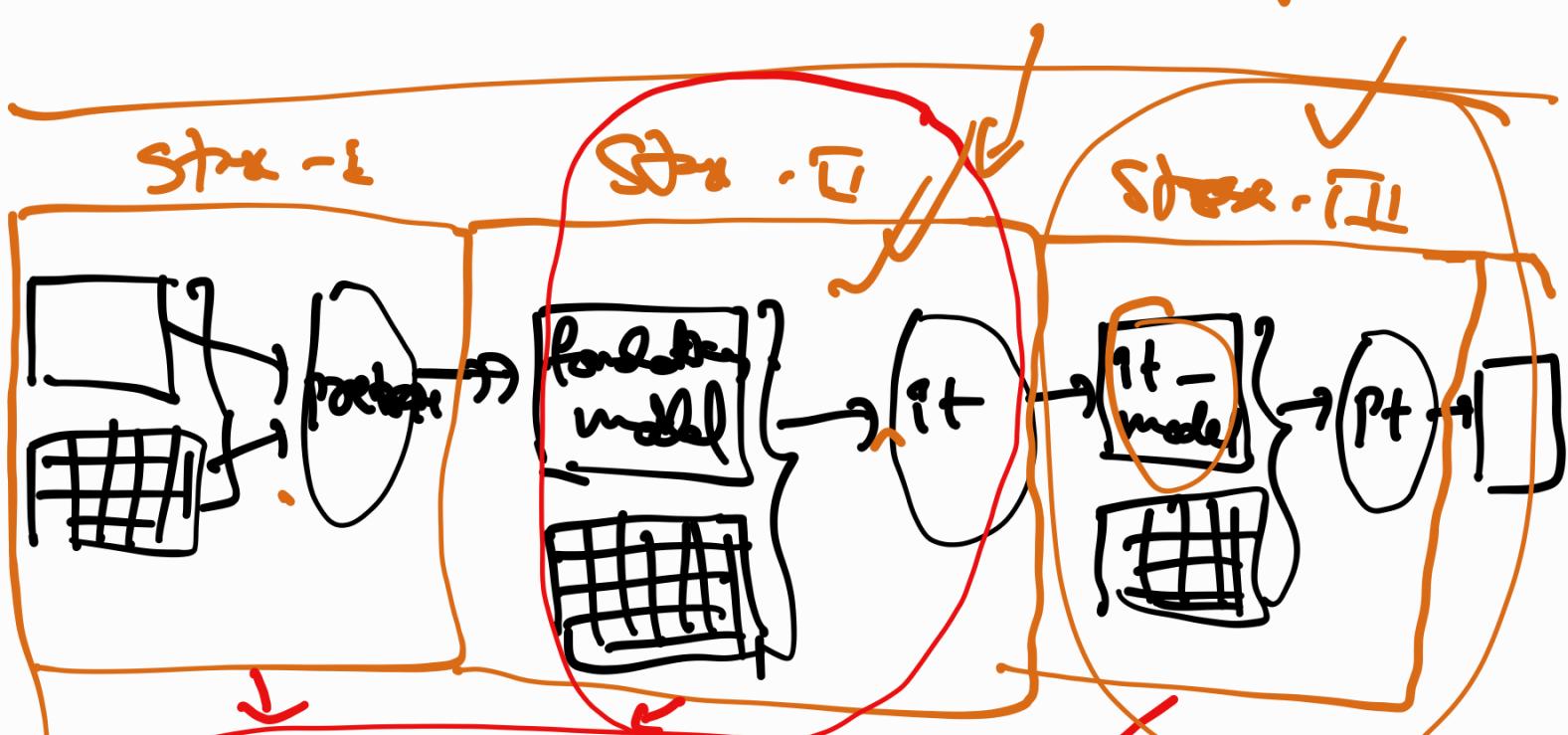
application



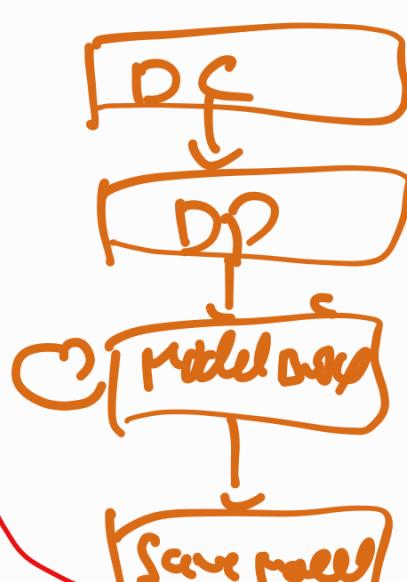
building
Bustine



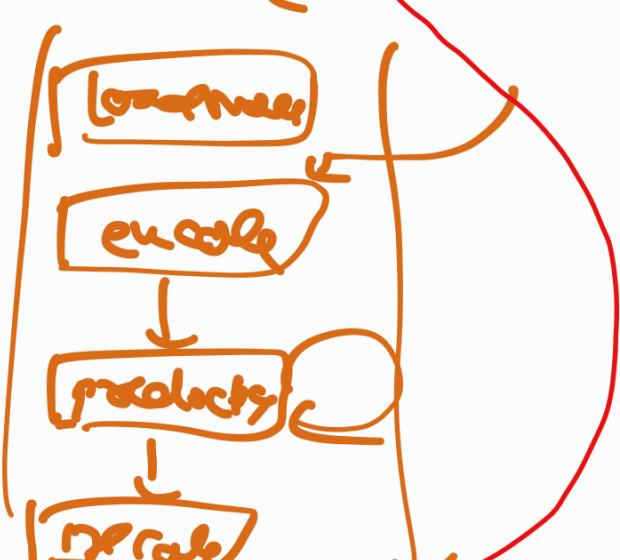
Centr.
Sectkt



tokens



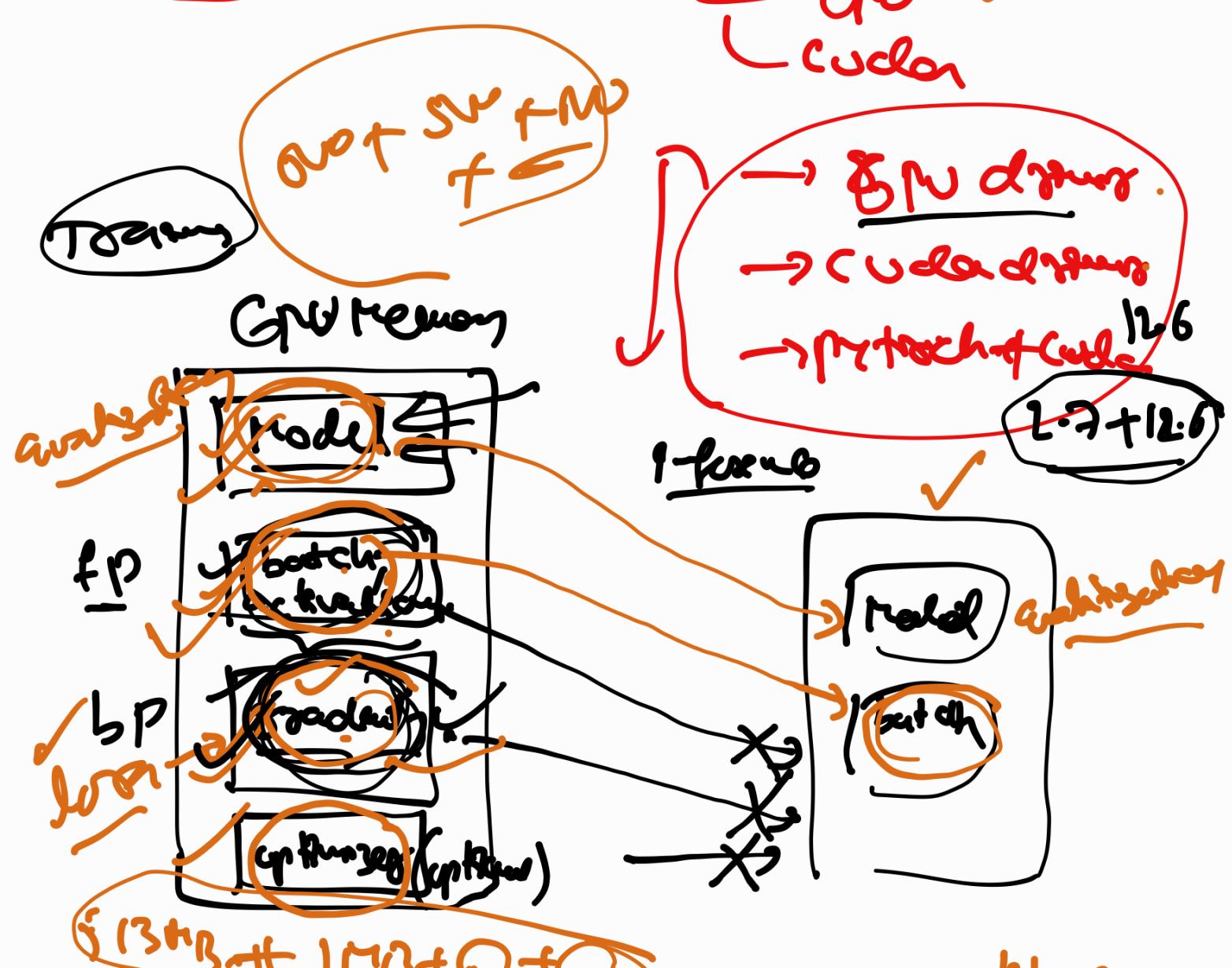
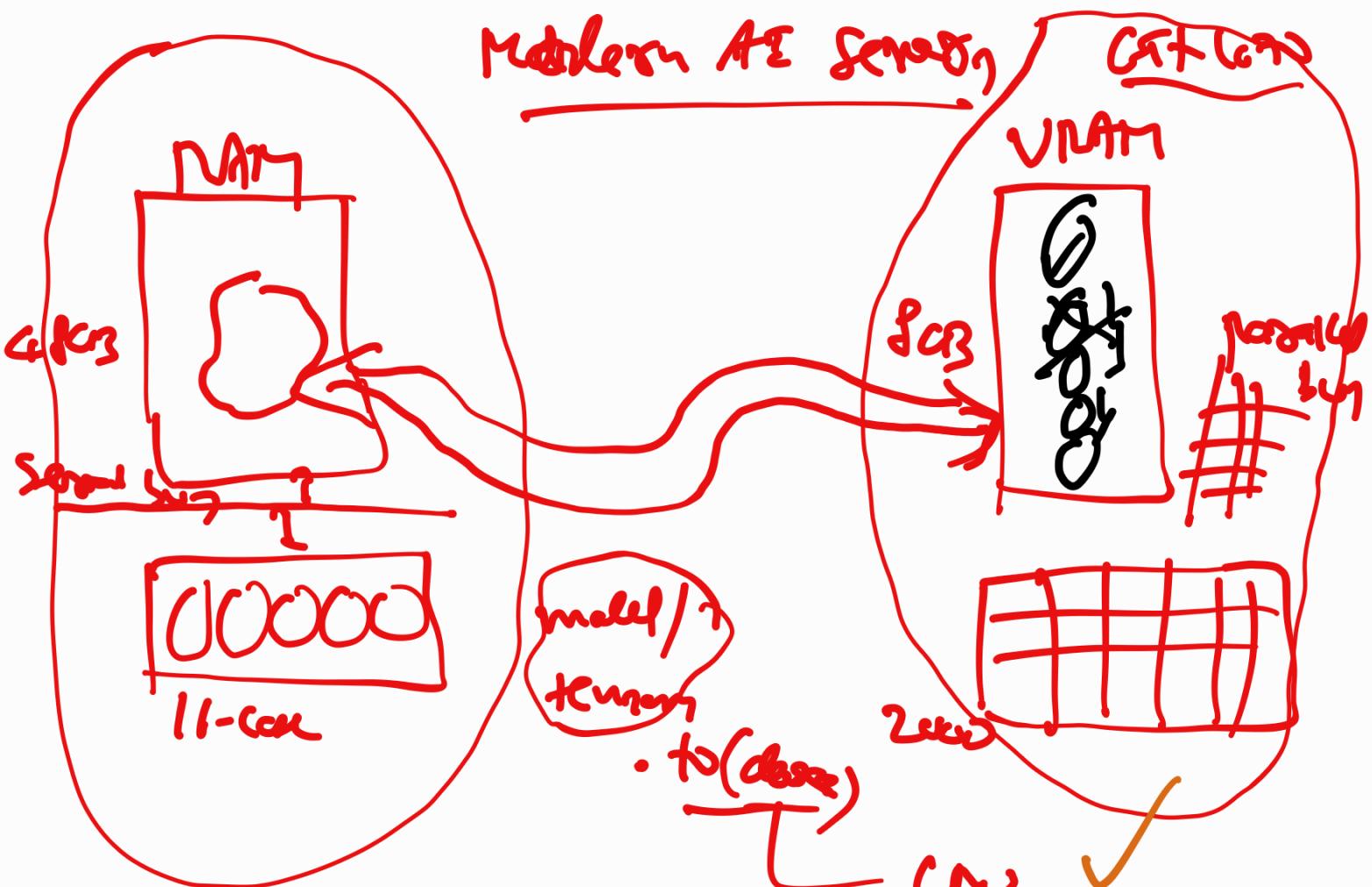
Inference



-> pytorch
-> OHF

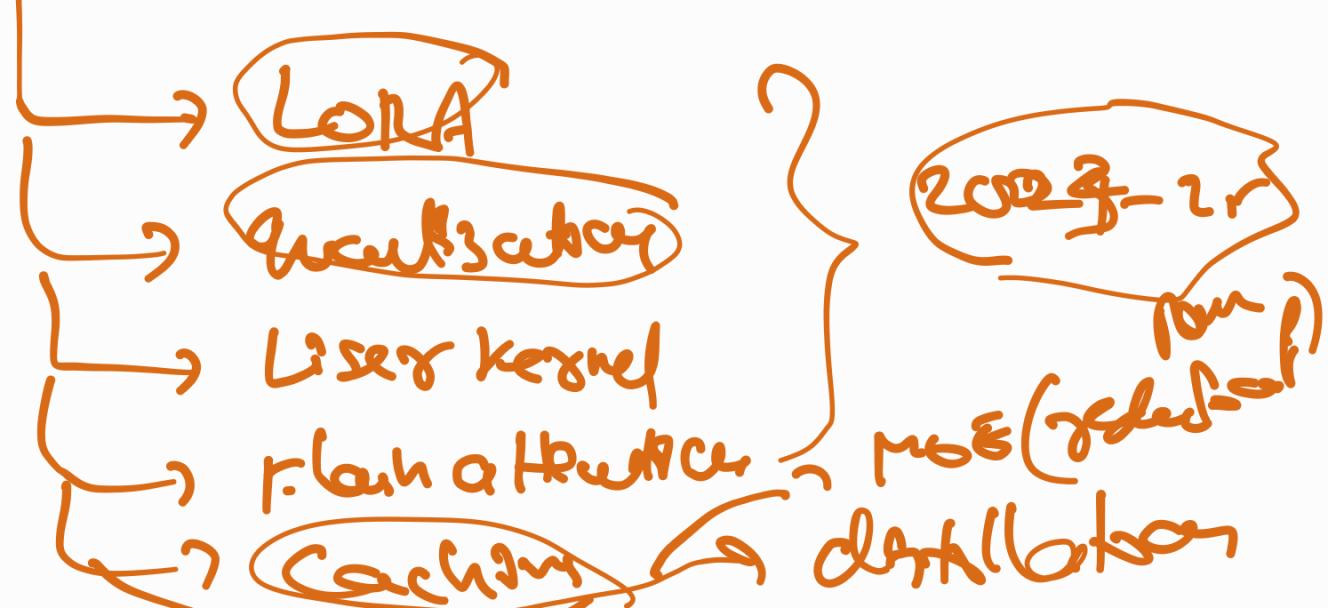
-> =

64GB
multi GPU → 1000
algo



Efficiency techniques to optimize

GPU memory effectively



| <gradient> | w1

$$2 \times 10 \times 4$$

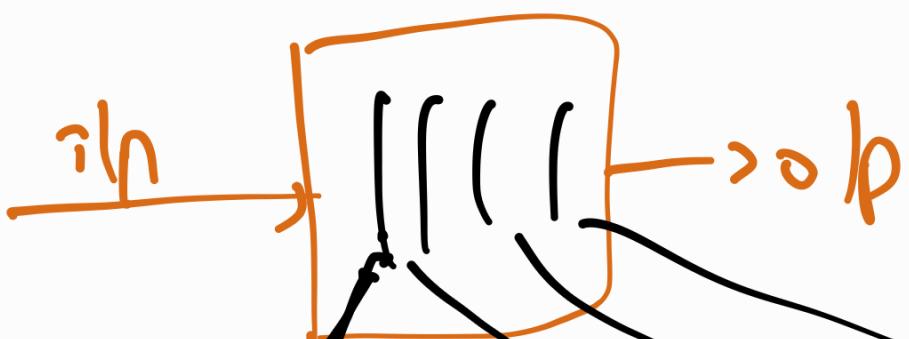
$$8 \times 6 \beta = 8 \text{GB}$$

$$2 \times 10 \times 4 \beta$$

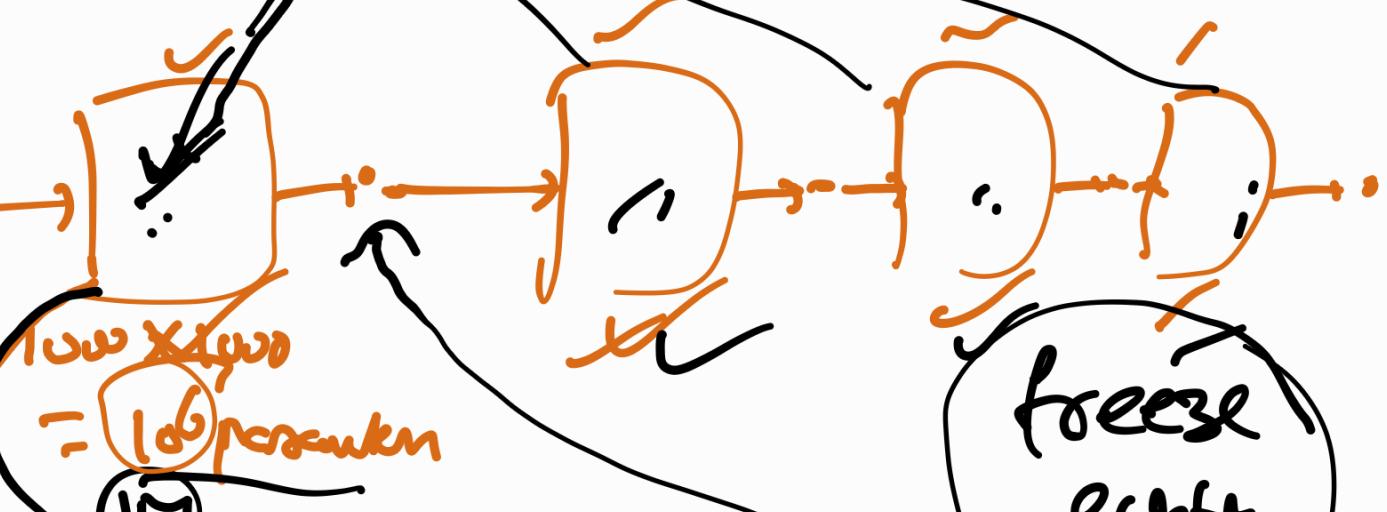
$$8 \text{GB}$$

↓ Cost 3-4x

Low Rank Adapter (LRA)

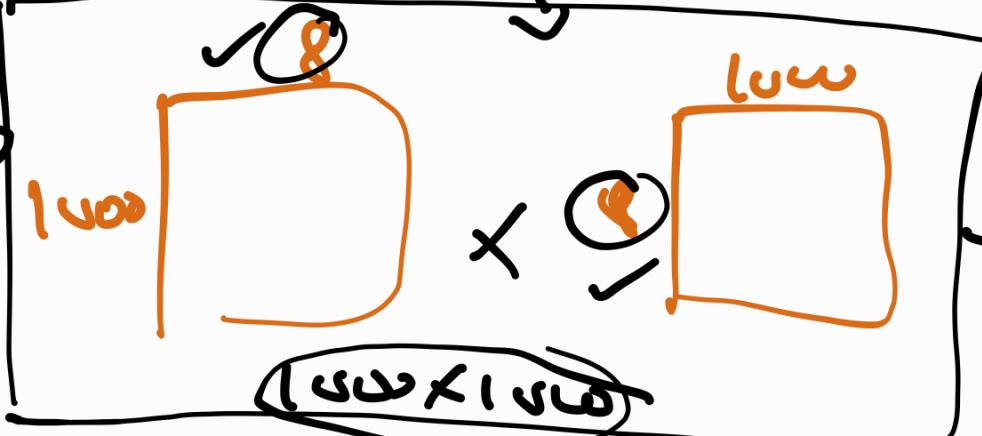


Learnable
naturally
avoids + blocks



Two X 1000
= 16 parameters

freeze
embed
layer

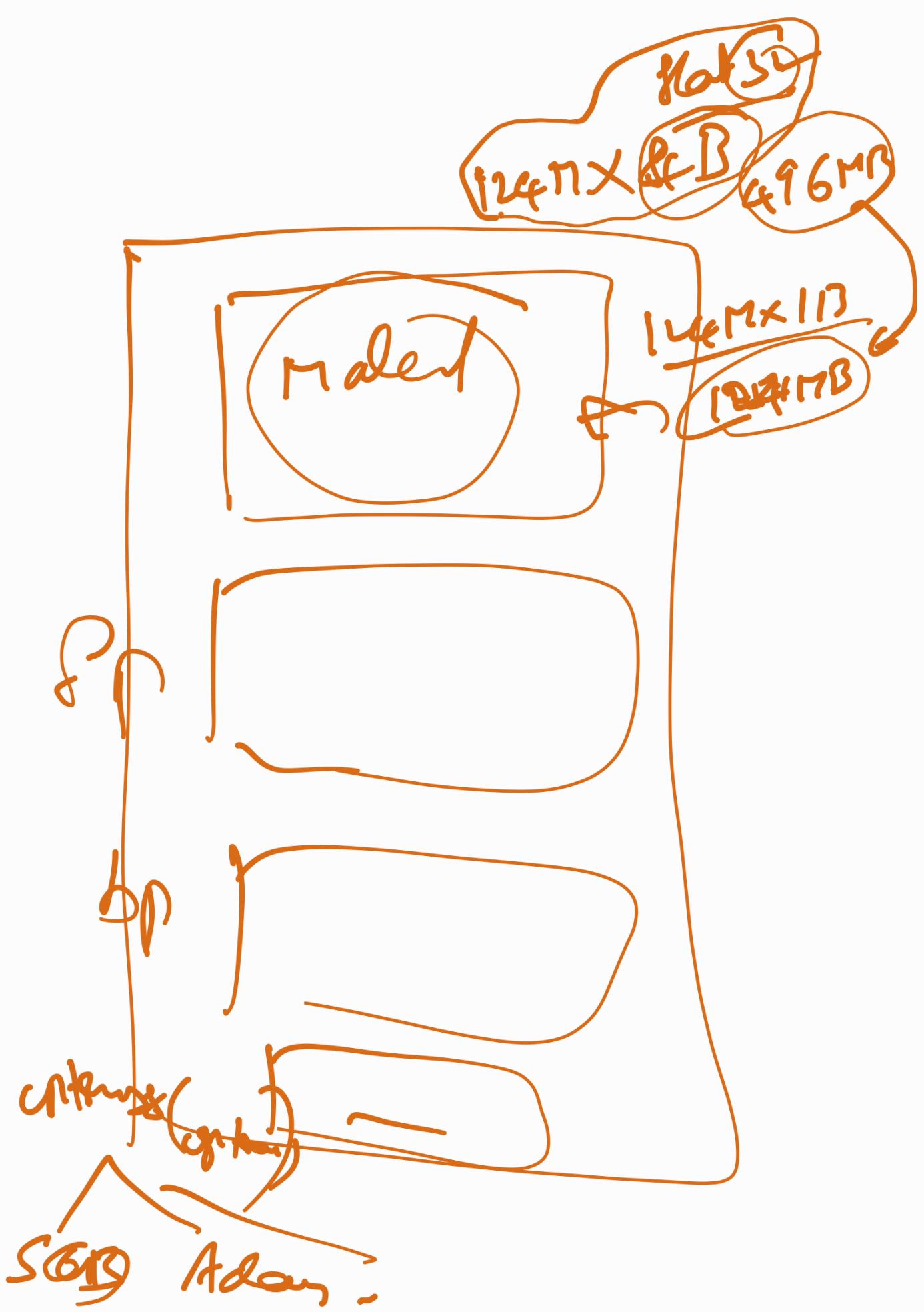
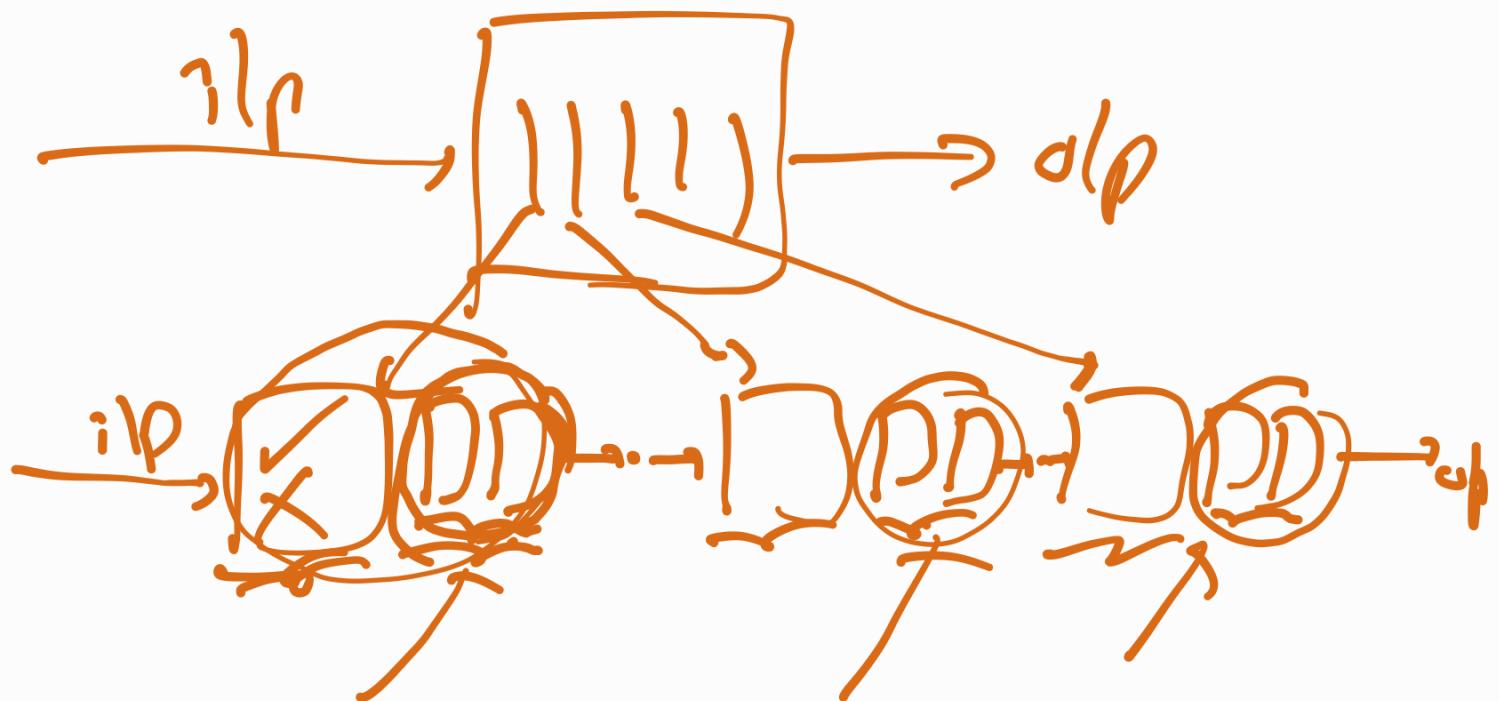


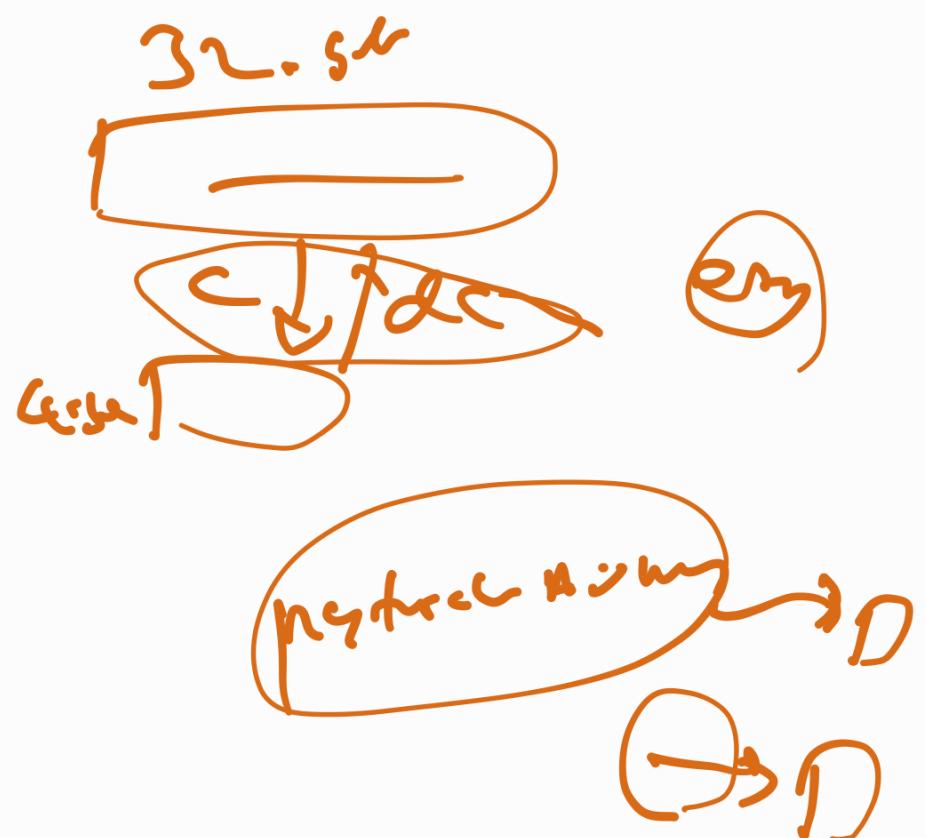
$$800 + 800 = 16 \text{ thousand}$$

16k

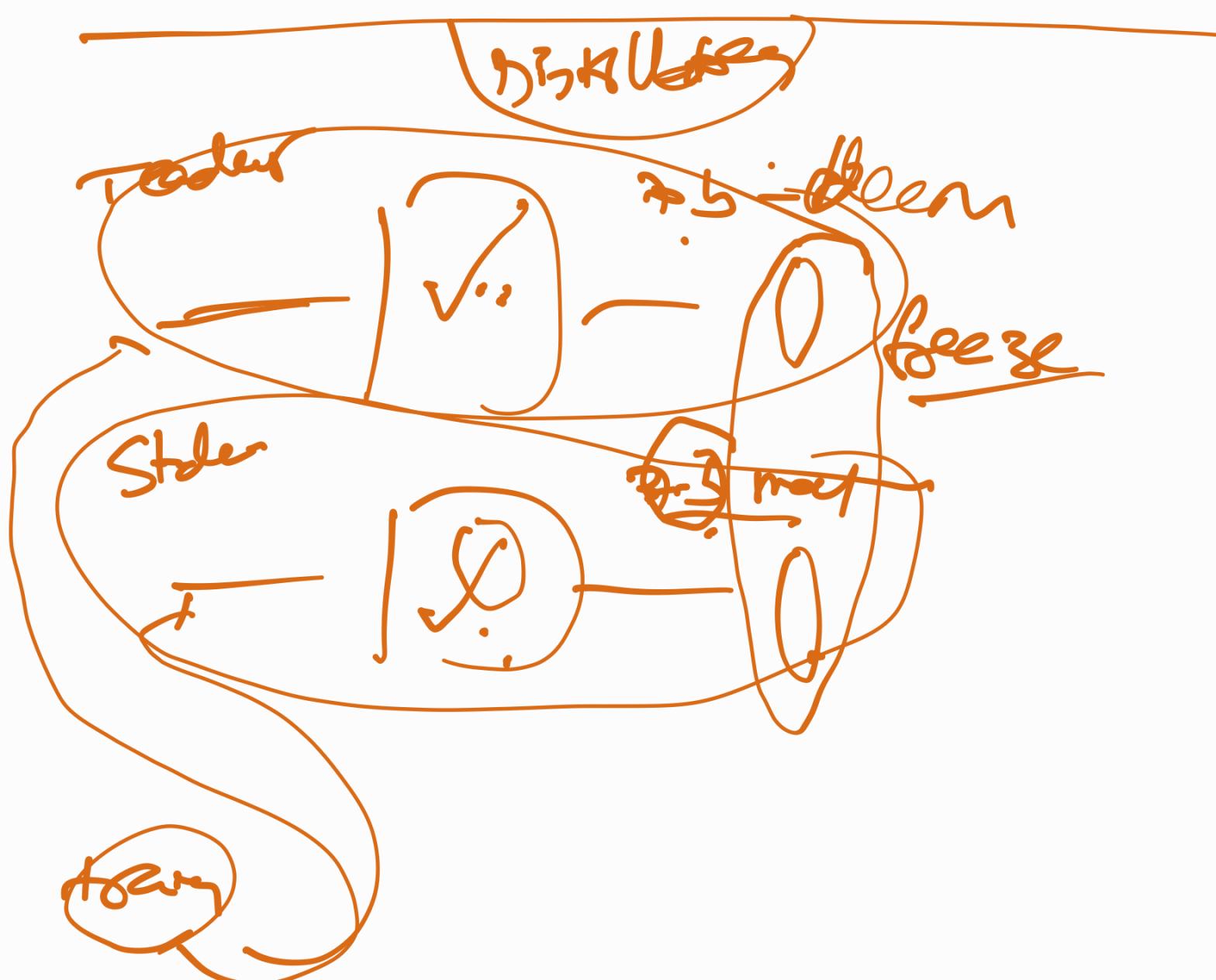
act-linear.
layer.

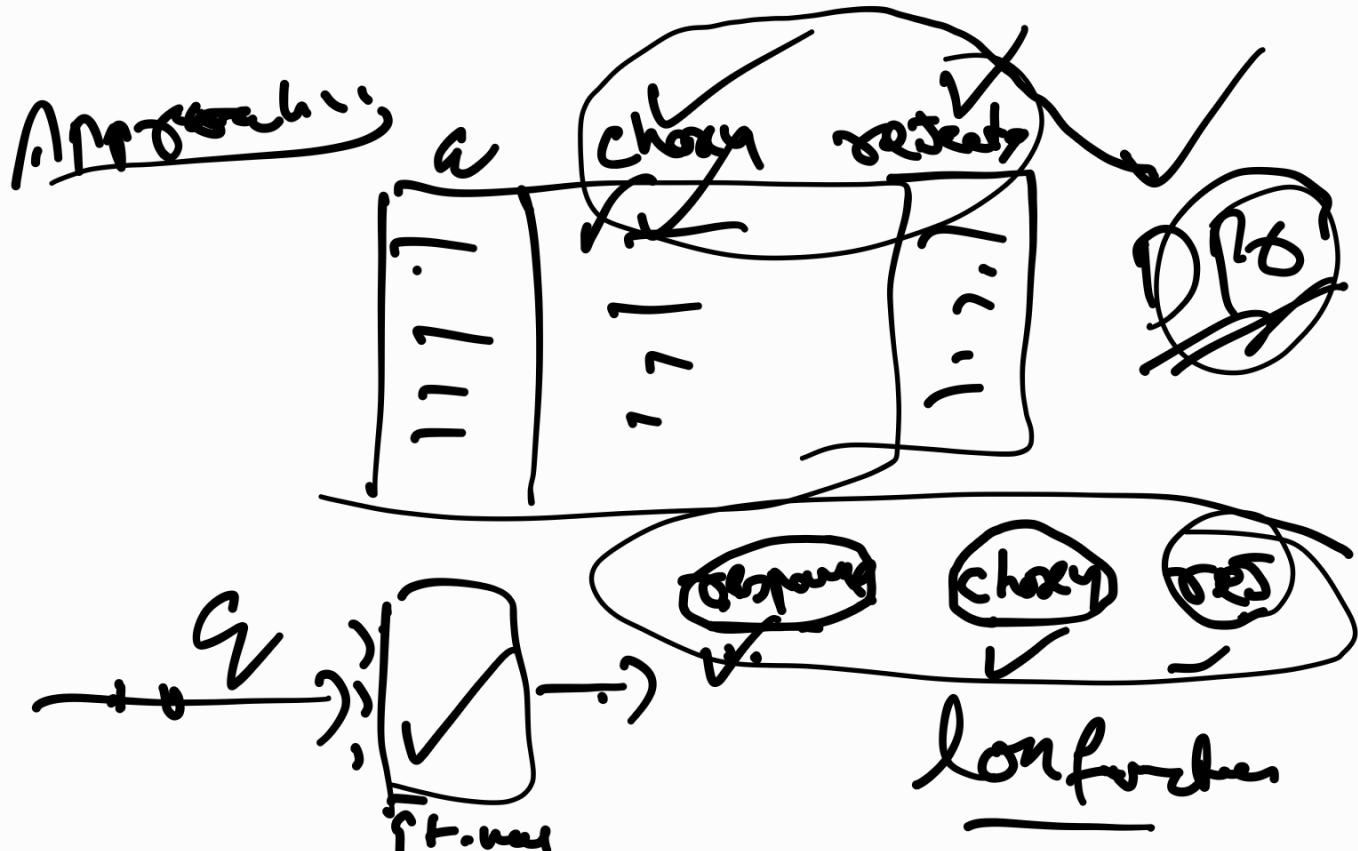
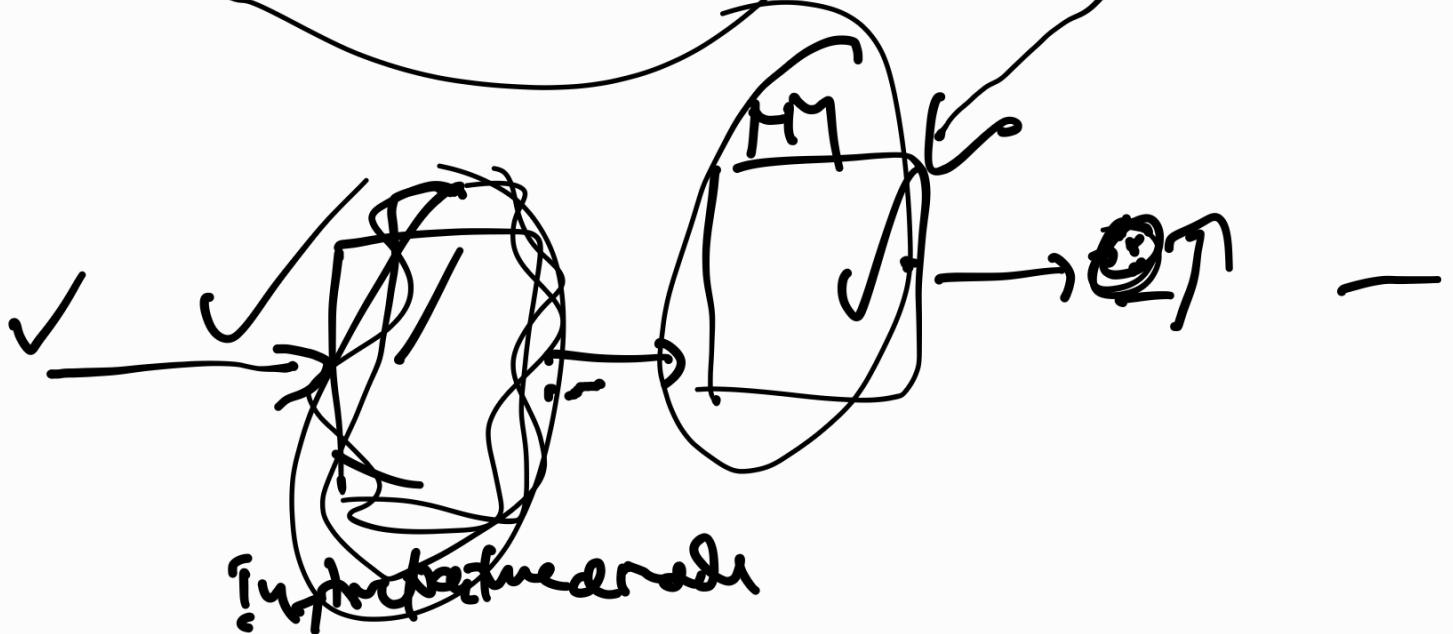
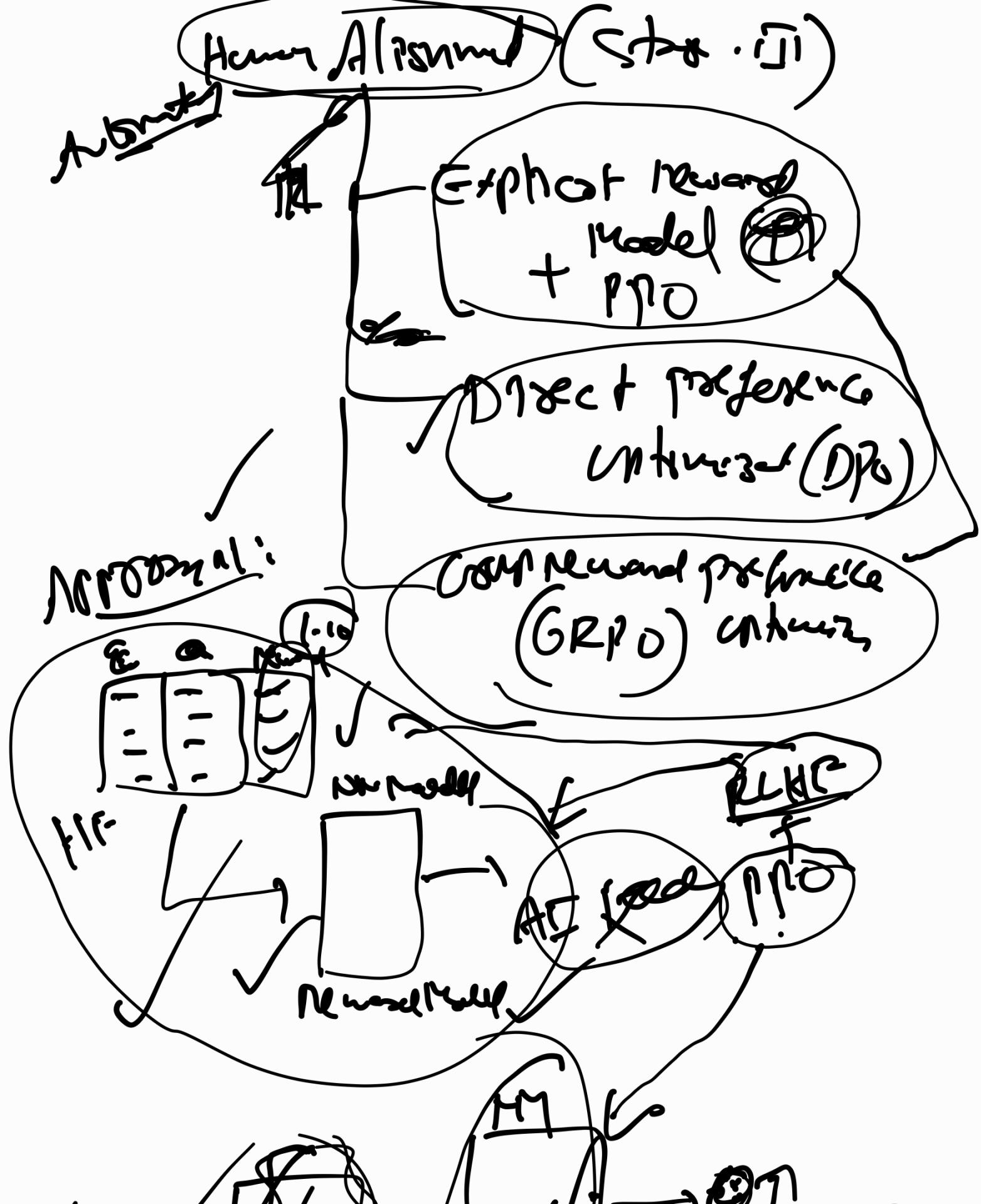
- faster to train
- less memory footprint



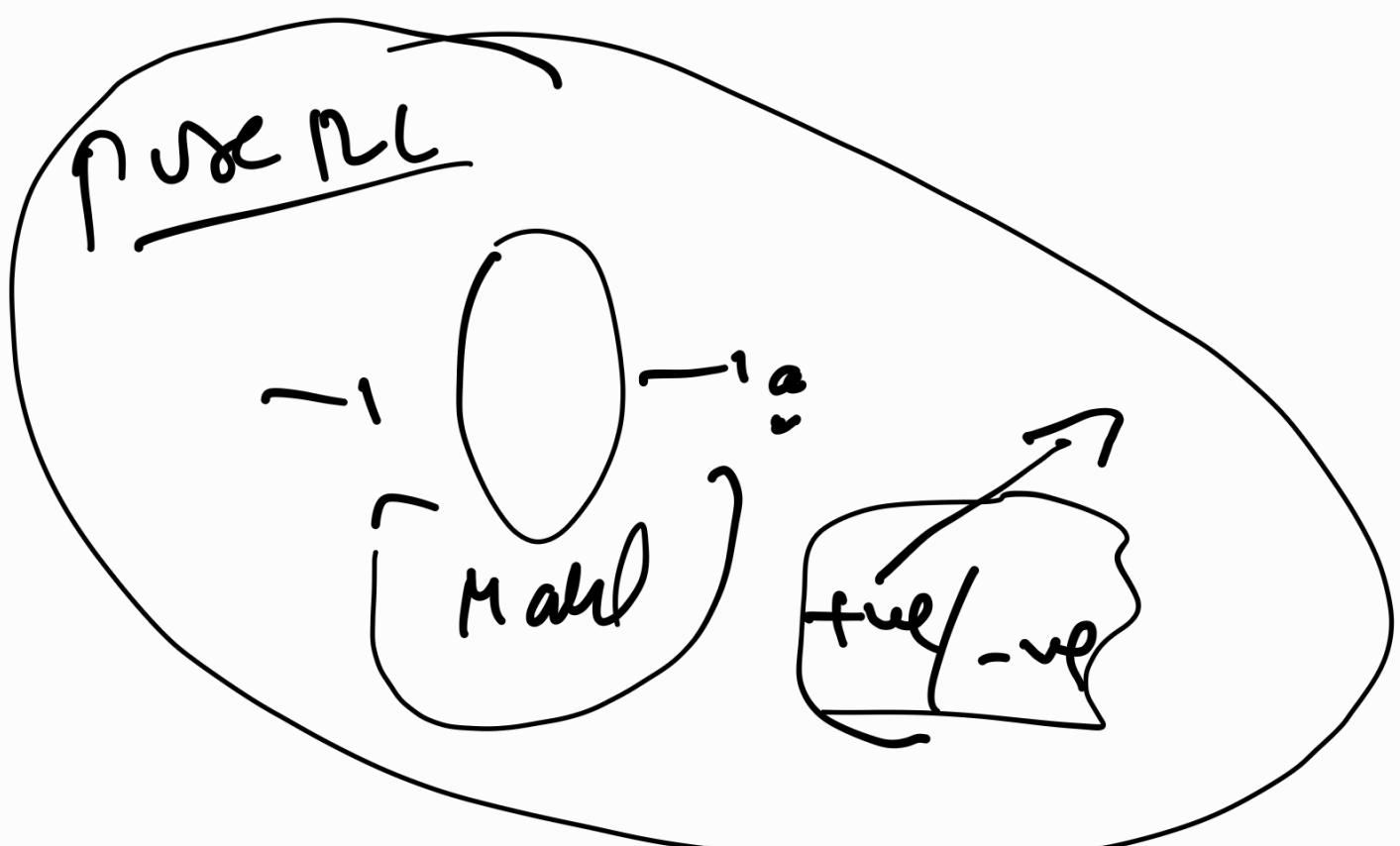
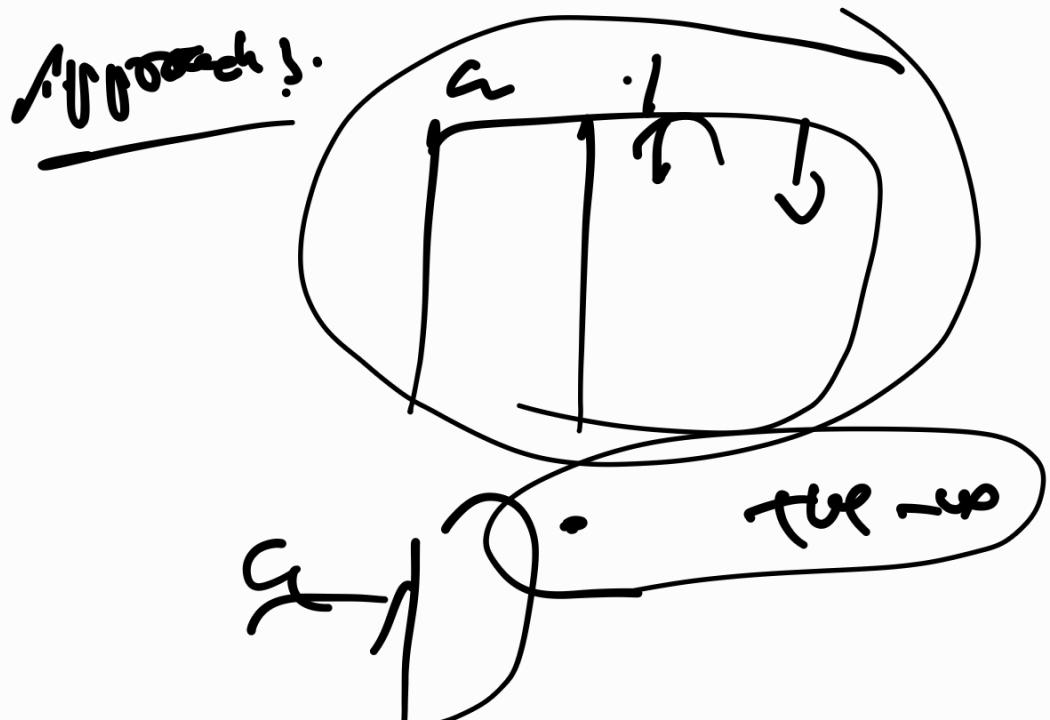


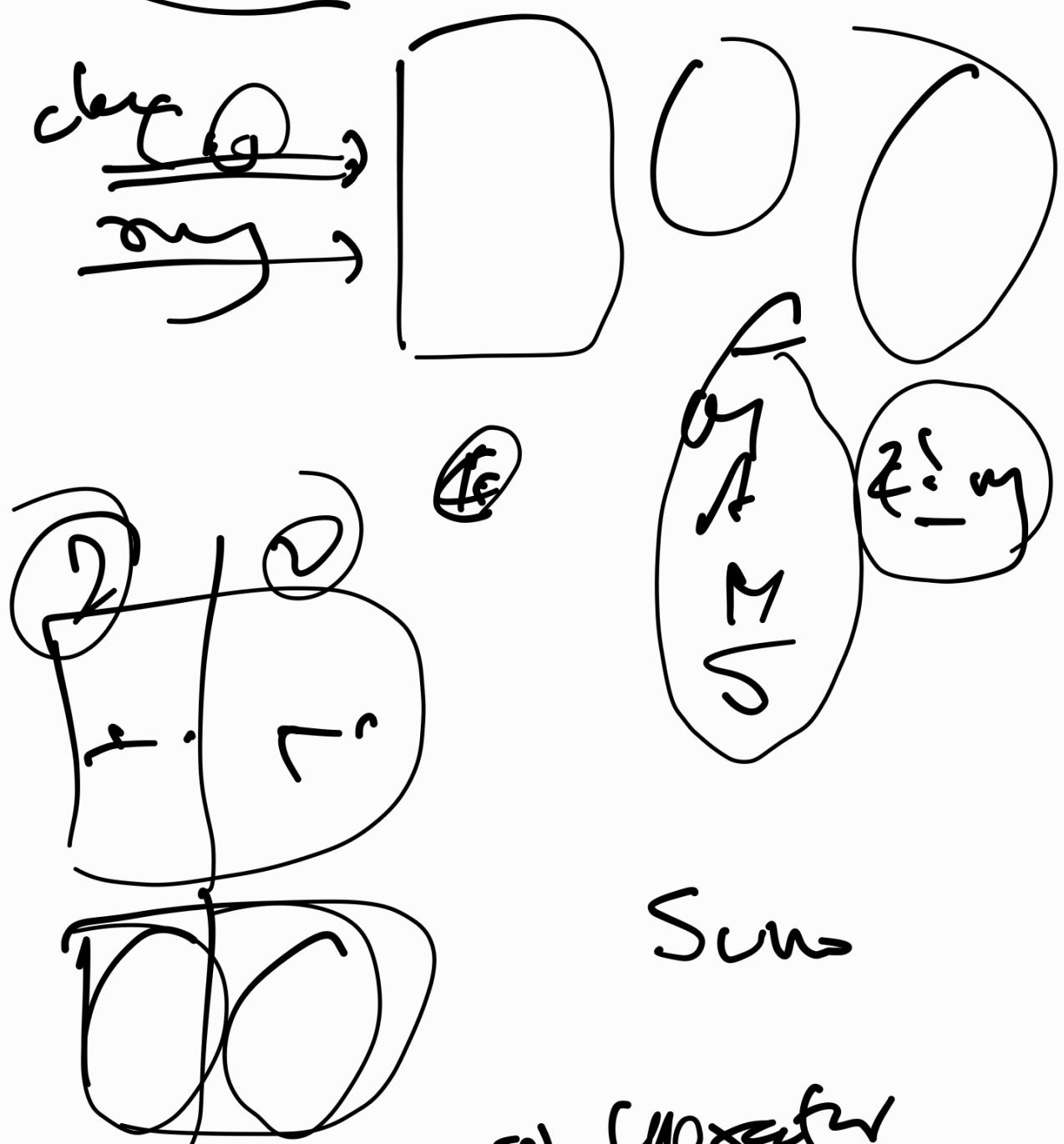
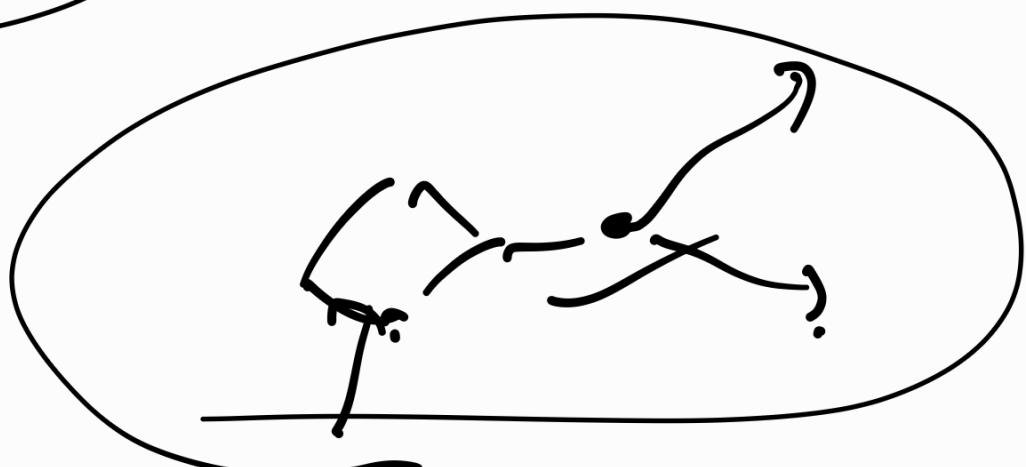
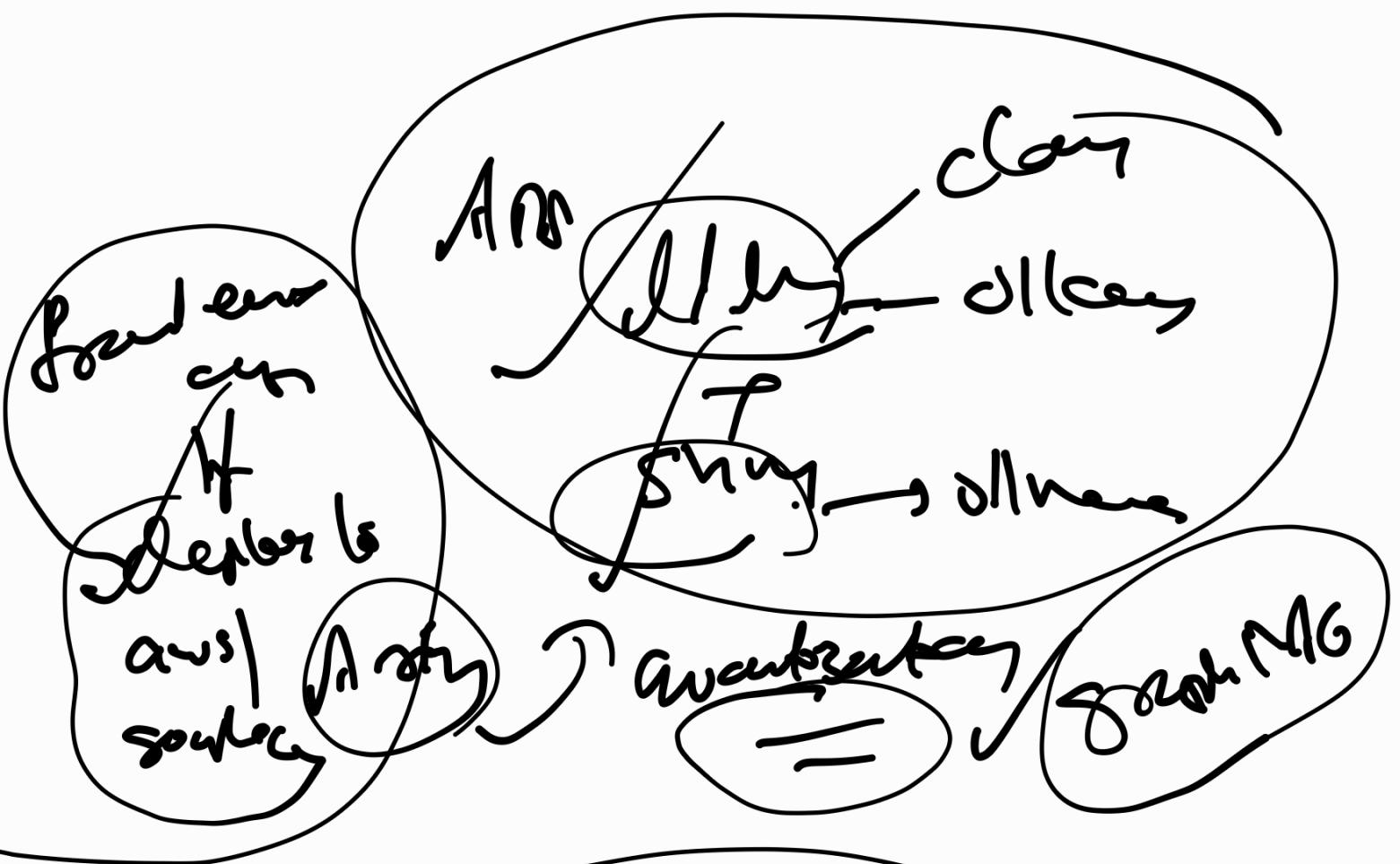
mother - k - 2





Approach 3:





→ Gash MG
 → Sun
 → Sun