

2024.12.18

reusing a layer expression in different models does not share weights, in same model shares weights

model inputs can be a list, outputs can be a dictionary

for cyclic or recursive) computations: subclass Model

can mix-and-match Sequential, layer expressions and a subclassing -- via composing)

(sub)models

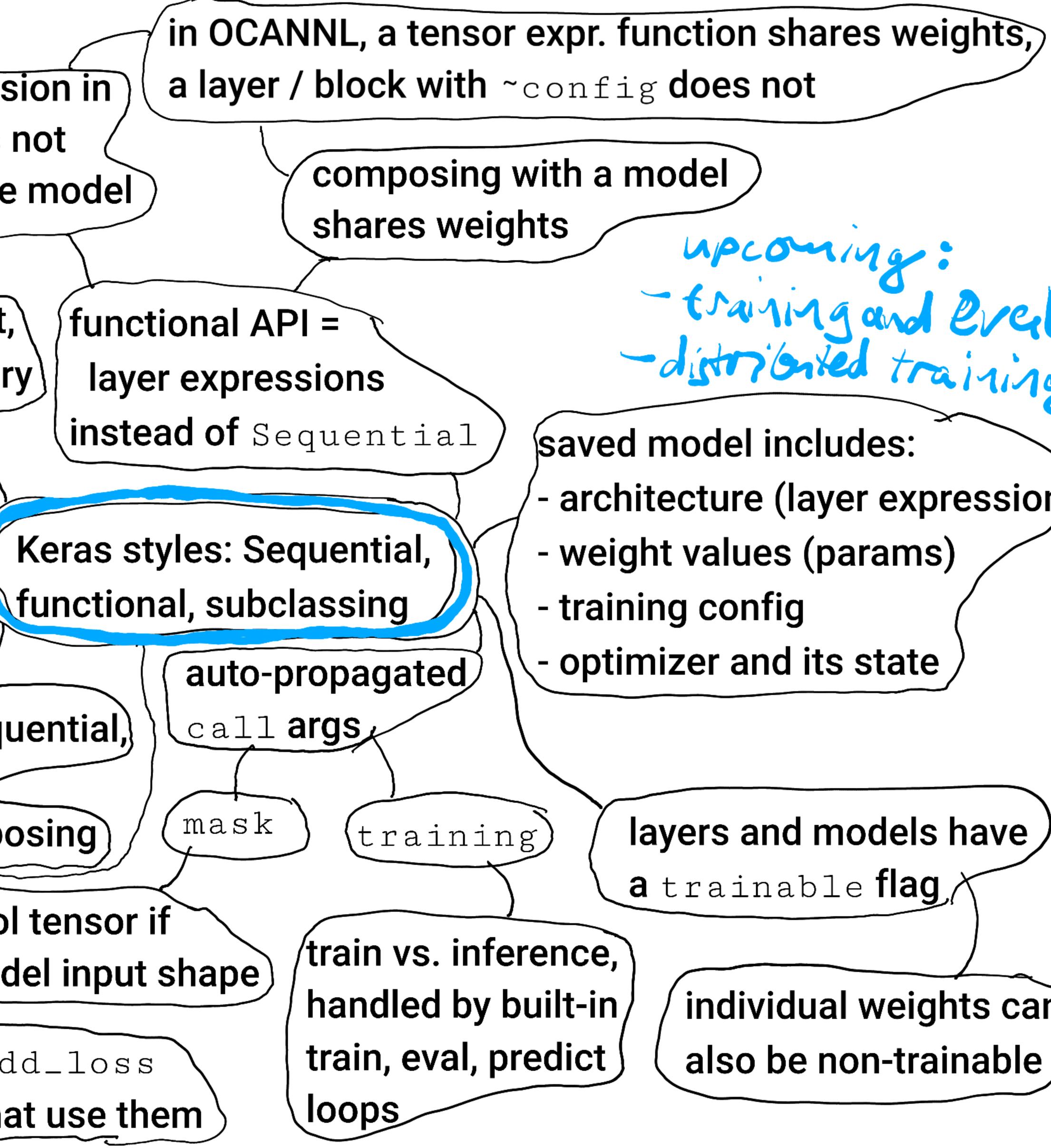
regenerated

per-call

bool tensor if model input shape

layers can add_loss

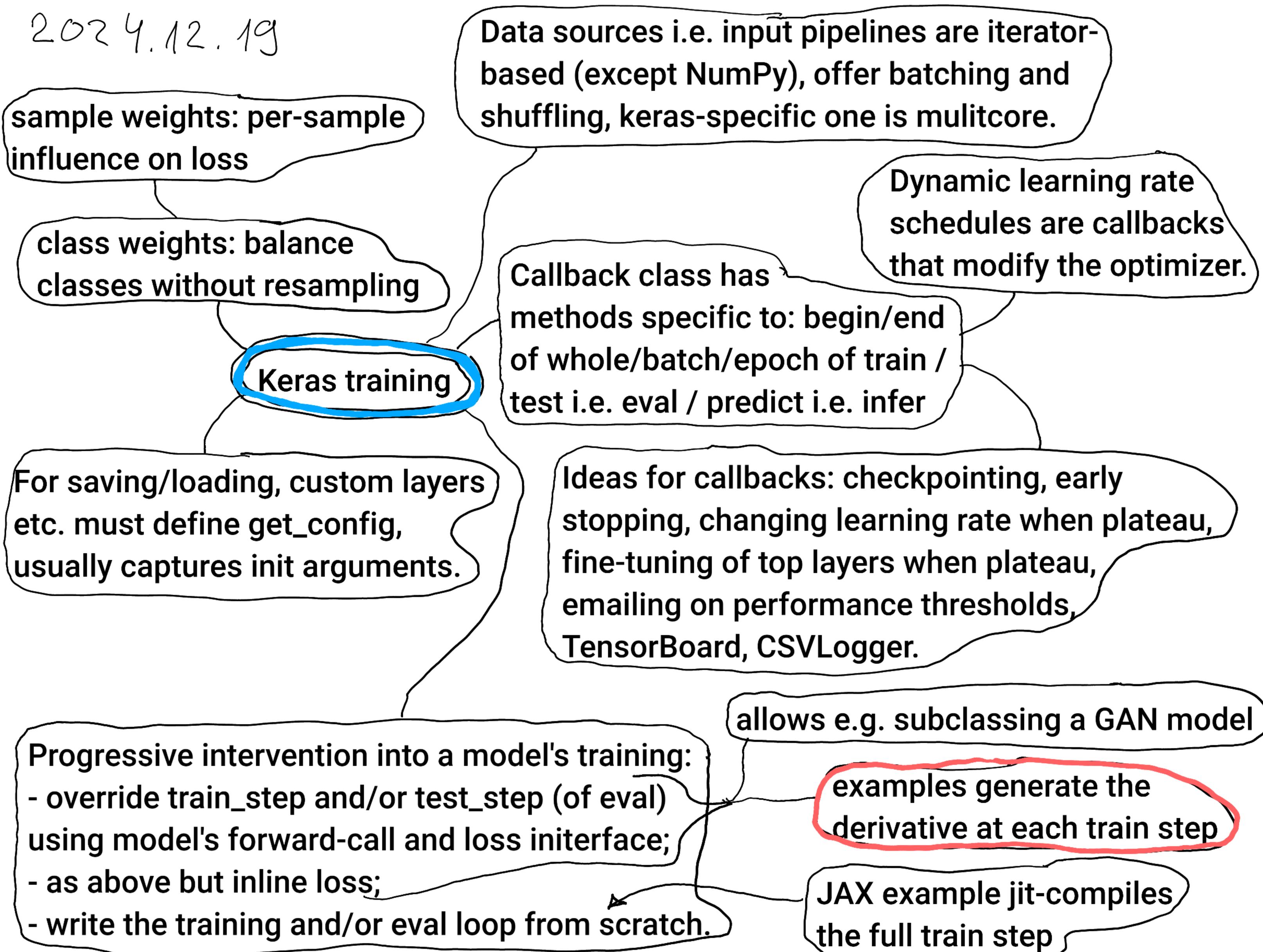
to models that use them

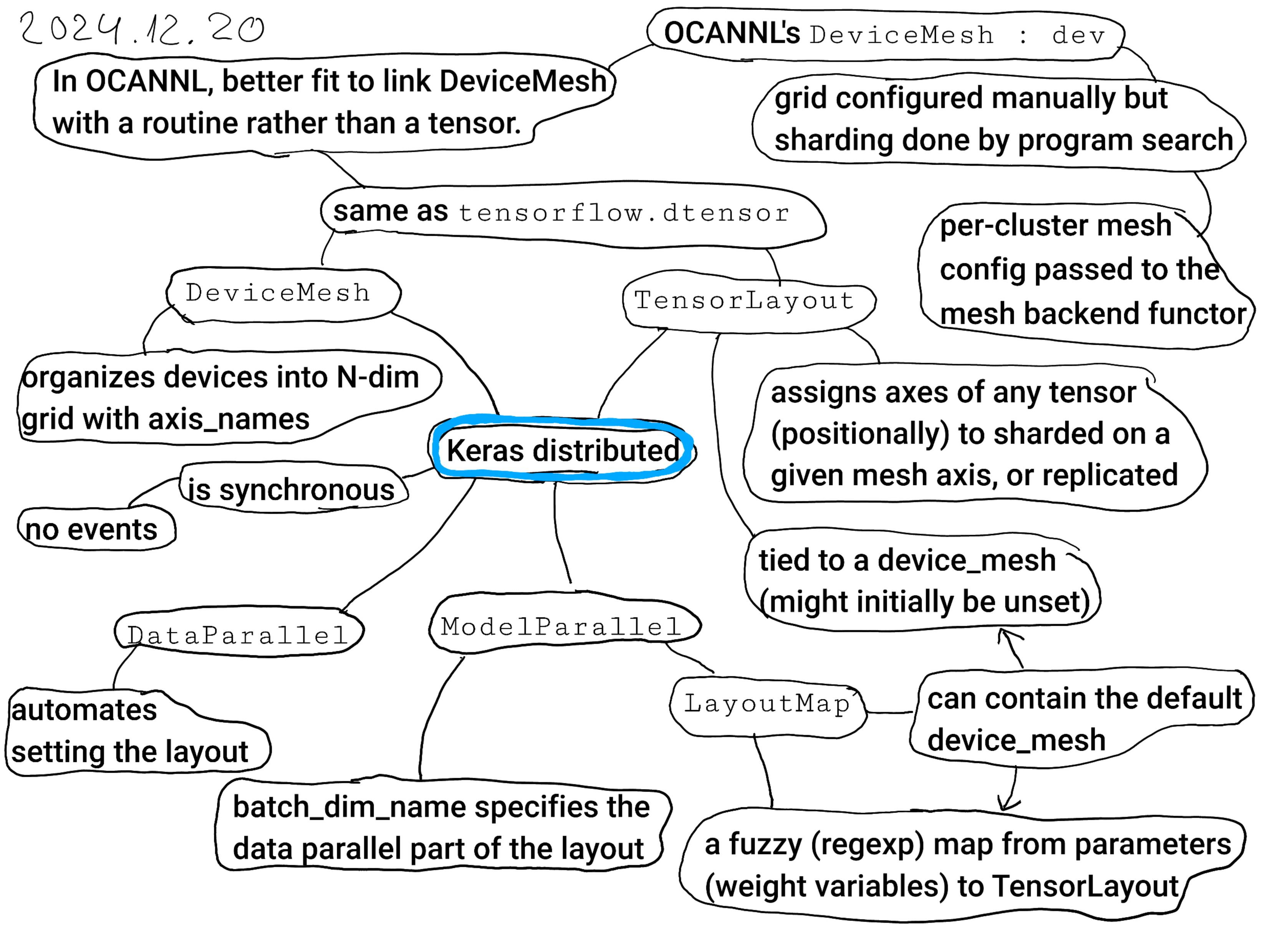


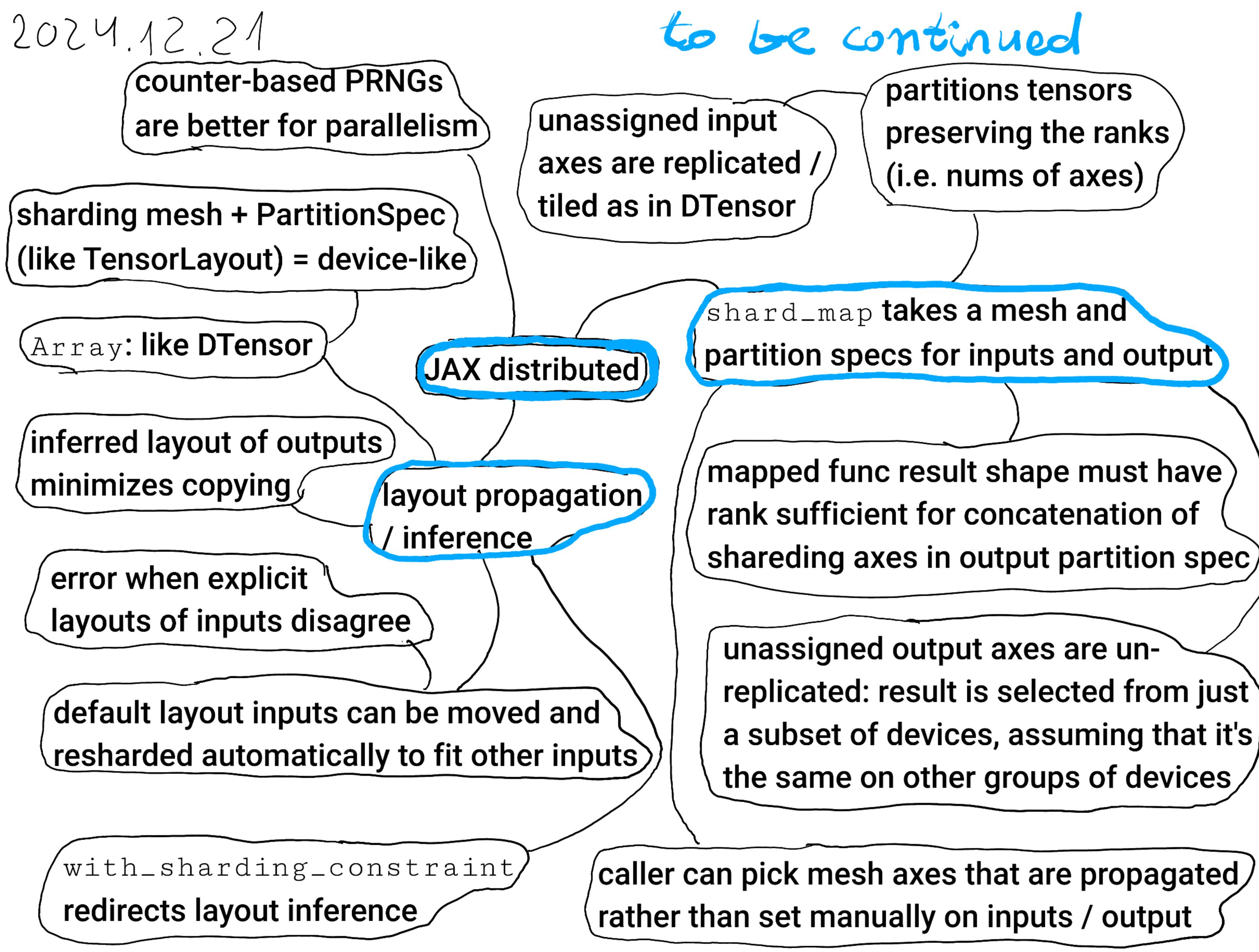
upconing: Esaising and evel distributed training

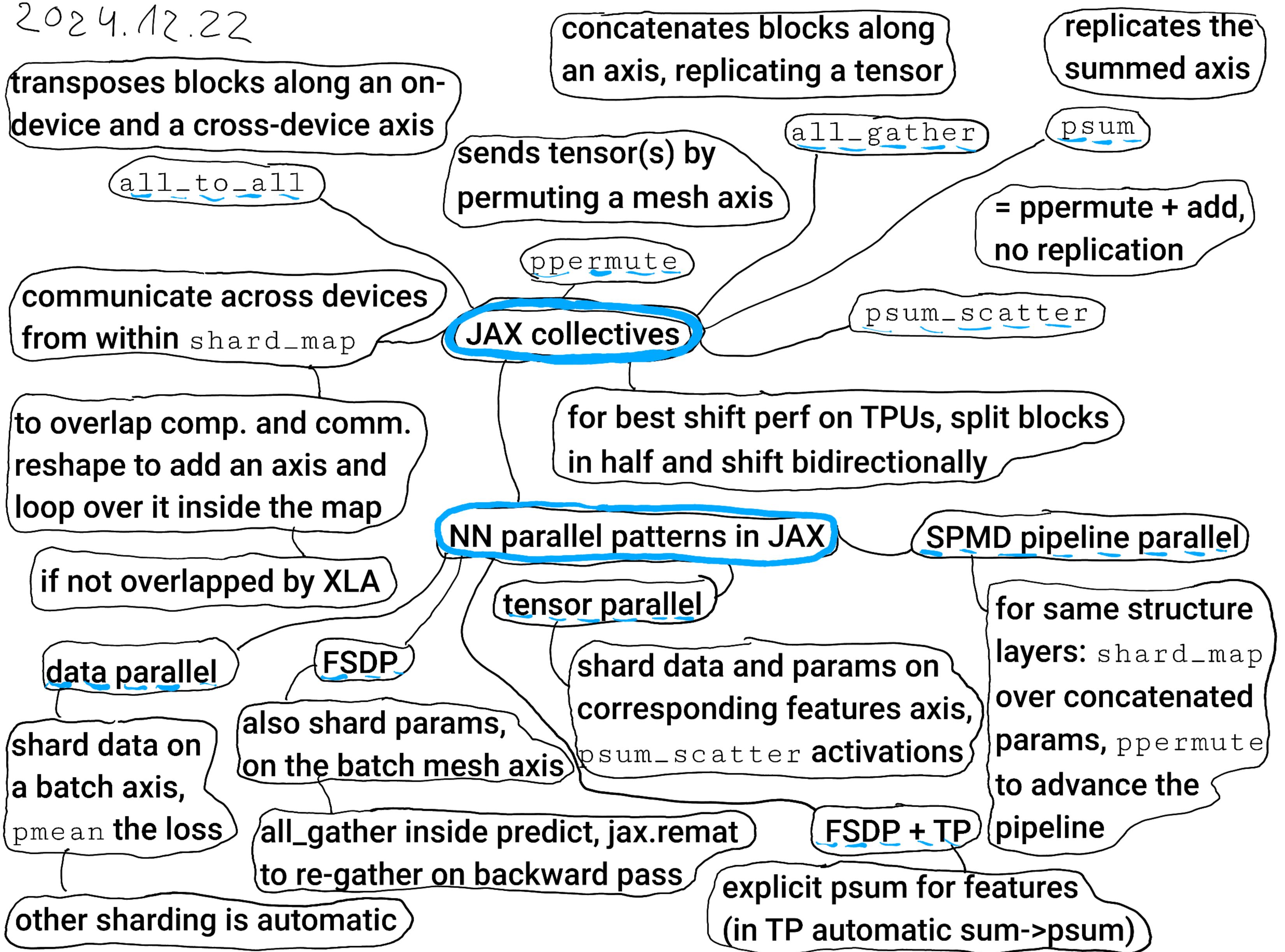
saved model includes: architecture (layer expression)) - weight values (params) - optimizer and its state

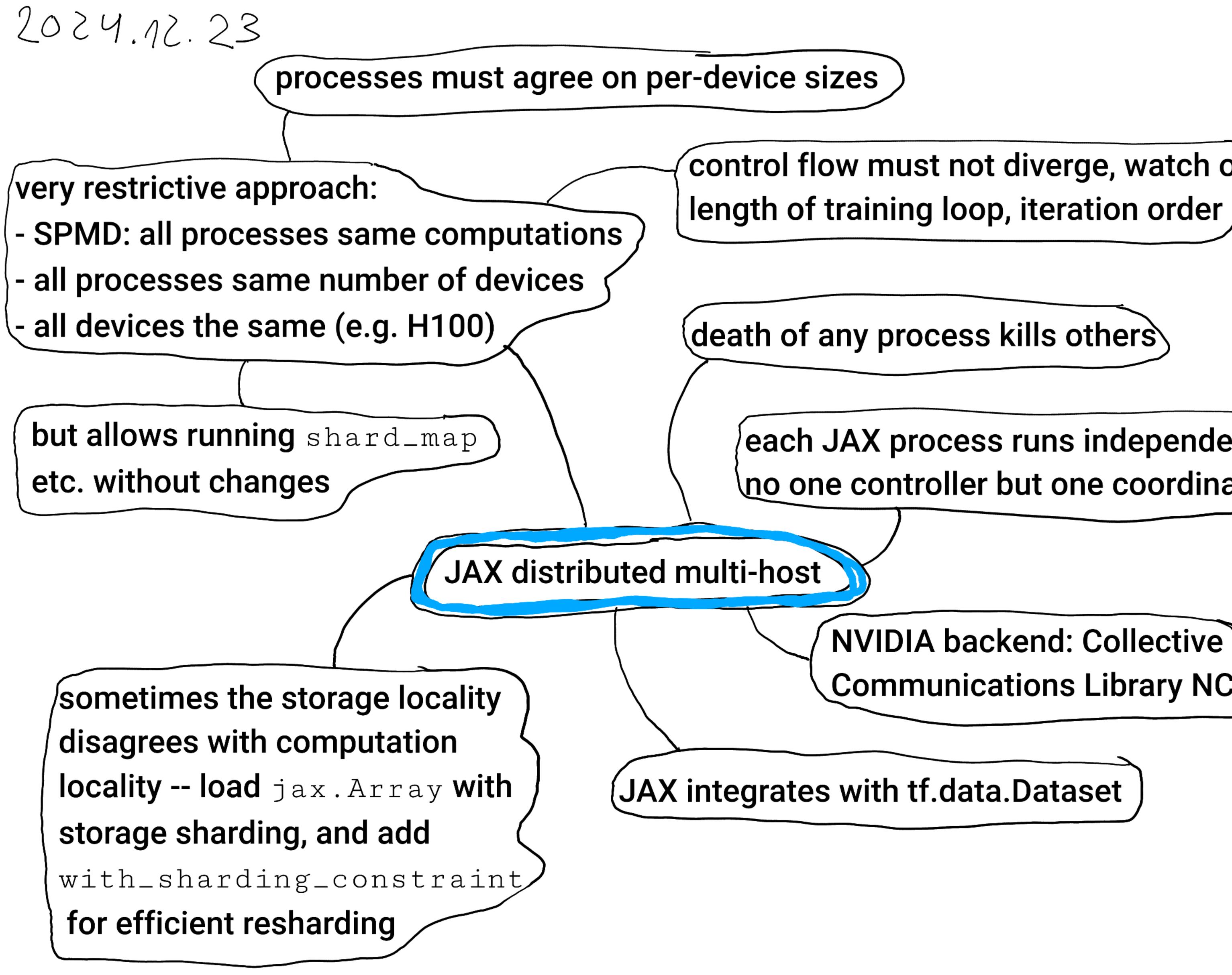
> layers and models have **a**trainable **flag** individual weights can also be non-trainable









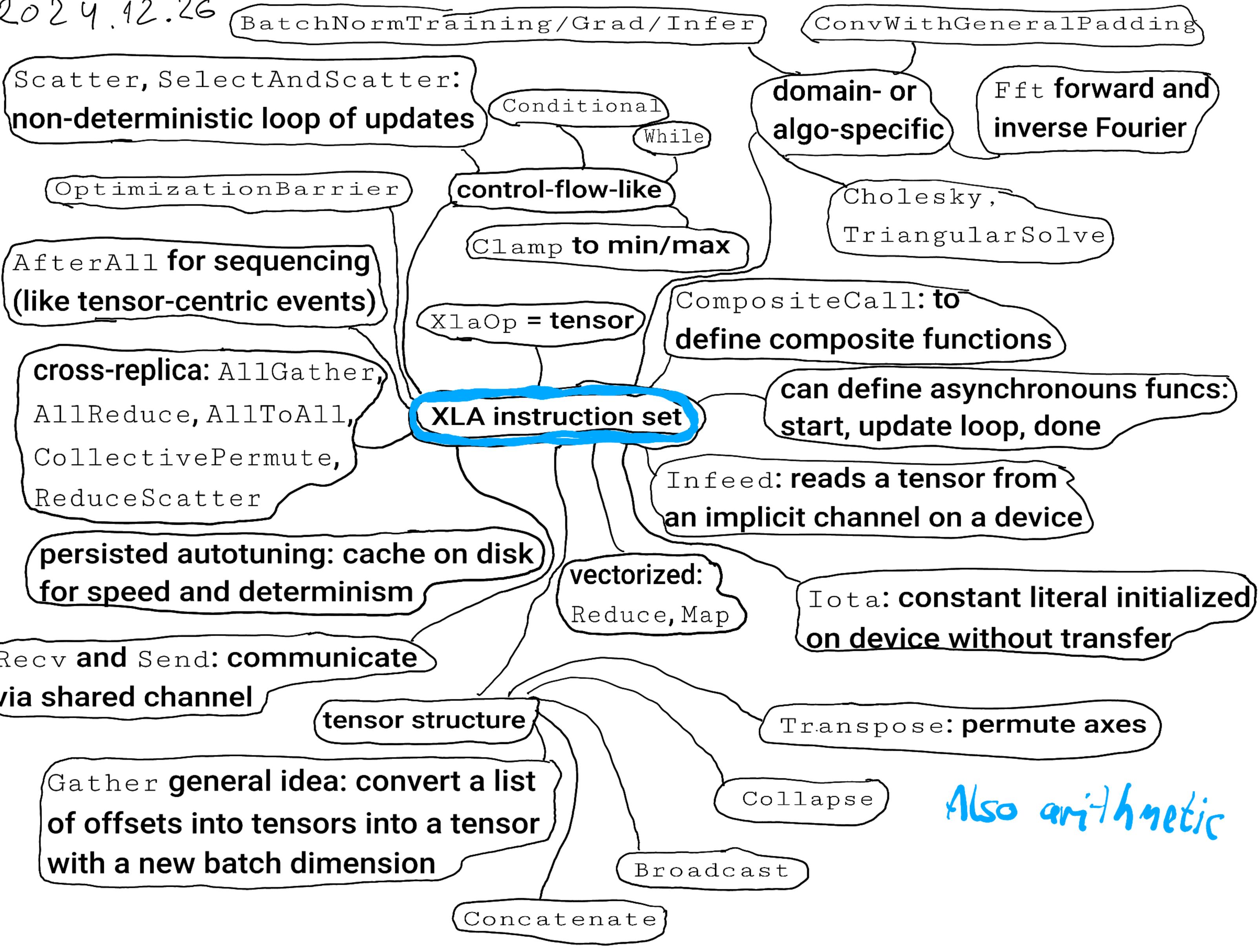


control flow must not diverge, watch out?

each JAX process runs independently, \no one controller but one coordinator

NVIDIA backend: Collective **Communications Library NCCL**

2024 12.26



Recv and Send: communicate

via shared channel

2024.12.27

3 compilation routes: libraries like cuBLAS /& cuDNN; tiling followed by Triton; Emitters

Partitioning: tensors are emitted in a single function when they interact pointwise without duplication.

Subkernel function inputs: "inflow" tensors and indices of "outflow" tensors; outputs: "outflow" values at the indices." Kernel function: takes both "inflow" and "outflow" tensor args."

(Only single-call functions are inlined.)

loop traversals linear in output tensors for coalesced writes, with boundary checks inside

tensors flattened to 1D as in memory

