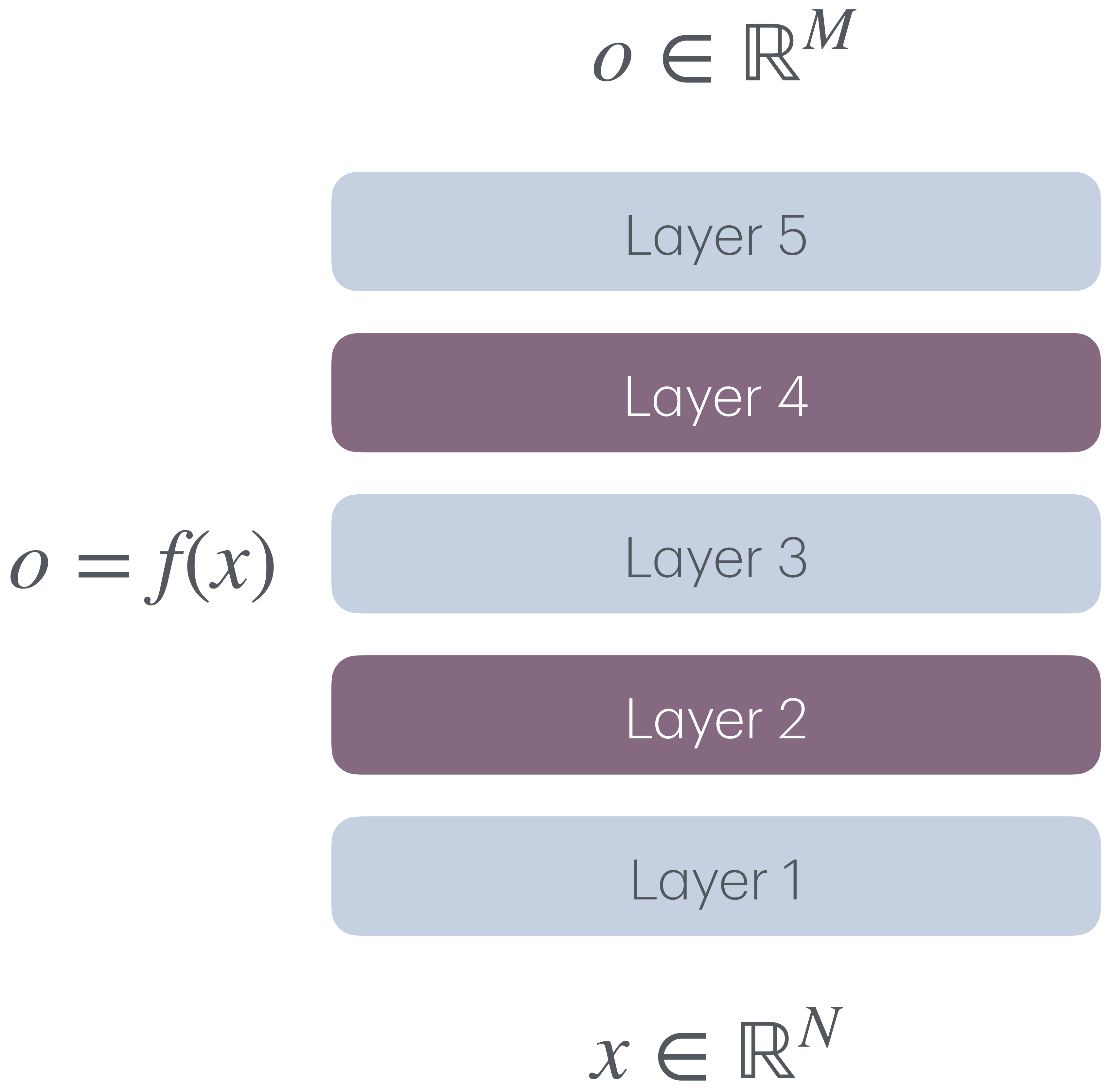
# Background: Training Deep Networks

Philipp Krähenbühl, UT Austin

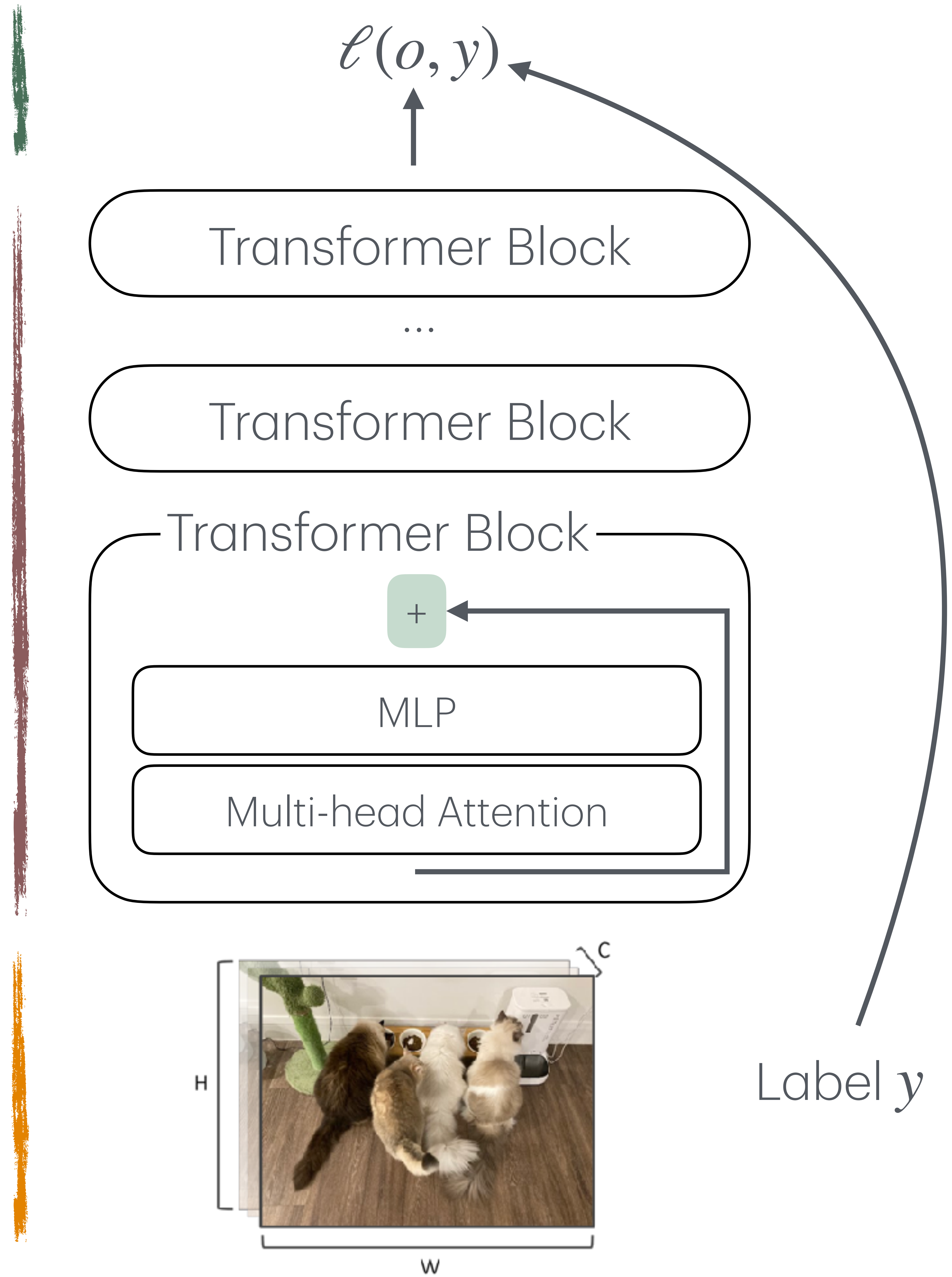
# What is a Deep Network?

- A really BIG differentiable function
  - Stacks layers of “simple” functions
    - Computation Graph
  - Trained with gradient descent and automated differentiation (backpropagation)



# Training a Deep Network

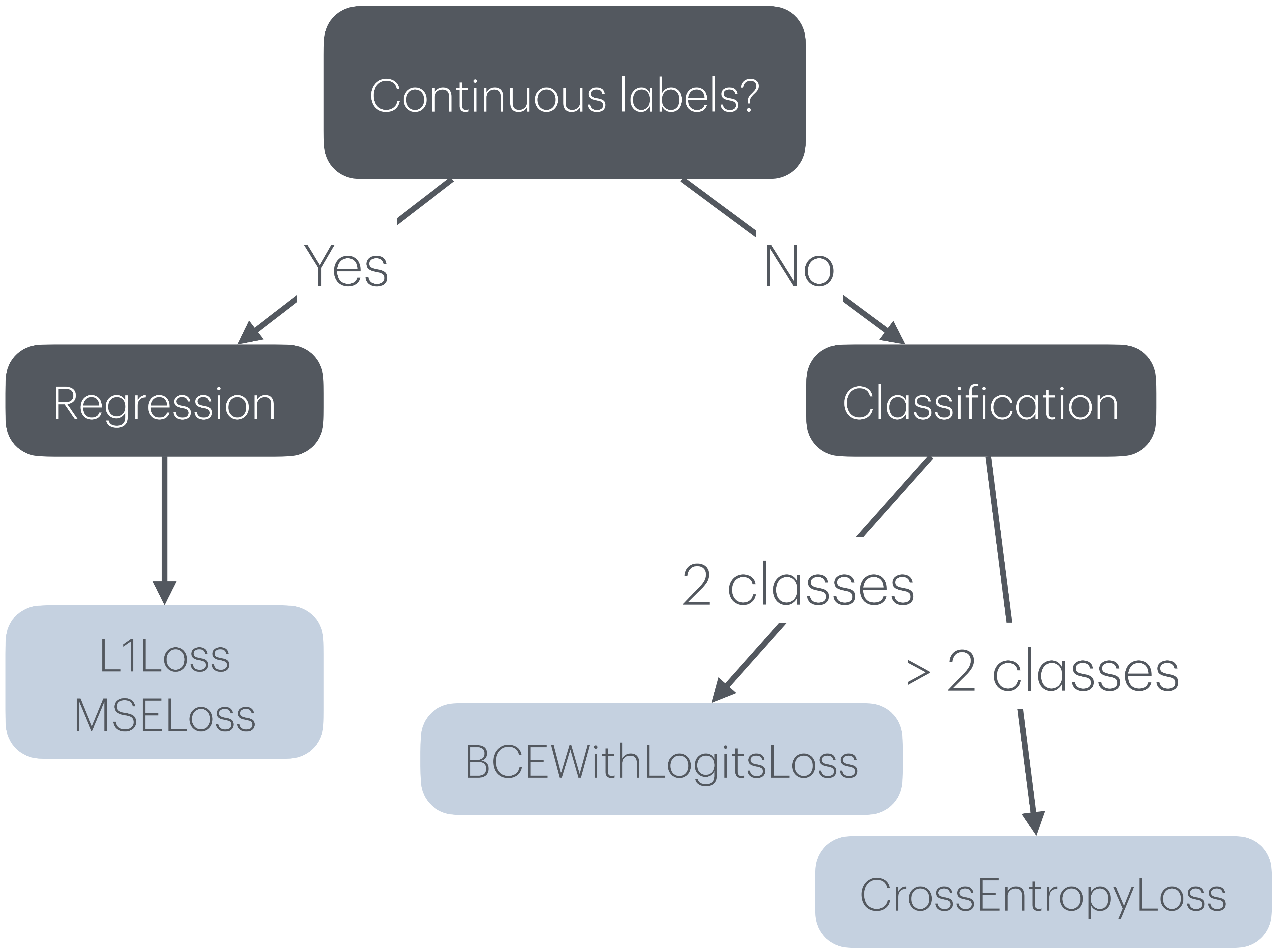
- Optimizer + Training objective (Loss)
- Architecture
- Dataset



# Loss Function

- Regression
  - L1 or L2 (MSE) loss
- Classification
  - (Binary)CrossEntropy
- Image/word-embeddings
  - CrossEntropy or specialized losses

$\ell(o, y)$

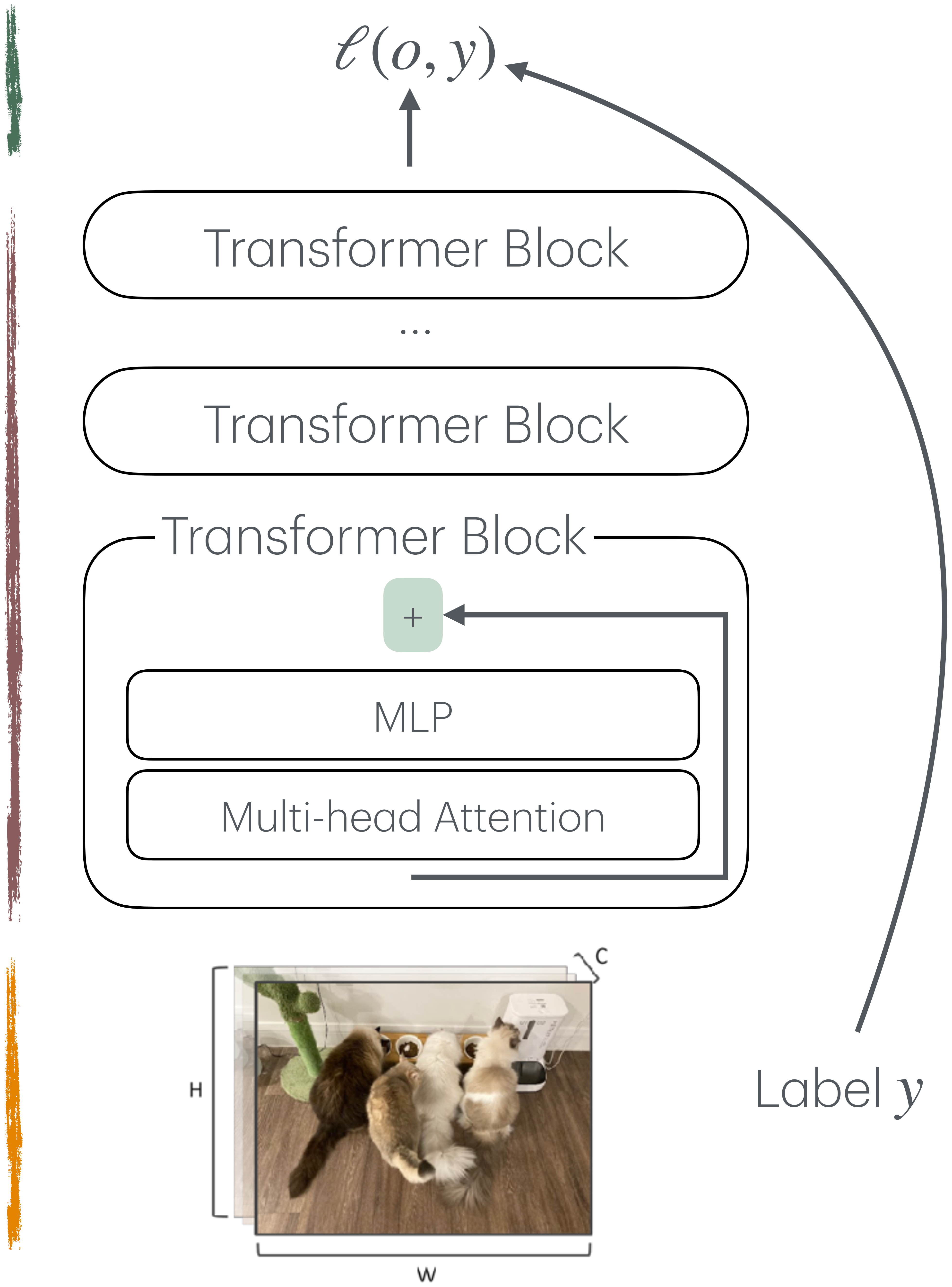


# Training a Deep Network

- Optimizer + Training objective (Loss)
- Architecture
- Dataset

Training objective

$$L(\theta) = E_{x,y\sim D}\left[\underbrace{\ell(f_\theta(x), y)}_{l(\theta|x,y)}\right]$$



# Training a Deep Network

- Minimize $L(\theta) = E_{x,y \sim D}\underbrace{\left[\ell(f_\theta(x), y)\right]}_{l(\theta|x,y)}$
- Use stochastic gradient descent
  - Specifically Adam / AdamW

```
m, v, t = 0, 0, 1
for epoch in range(n):
   for (x, y) in dataset:
      J = ∇l(θ|x,y)
      m = (1−β_1) ∗ J + β_1 ∗ m
      v = β_2 ∗ v + (1−β_2) ∗ J.square()
      m = m / (1 − β_1^t)
      v = v / (1 − β_2^t)
      θ = θ − ε ∗ (m / v.sqrt() + decay ∗ θ)
      t += 1
```

# Training a Deep Network

- Optimizer + Training objective (Loss)
- Architecture
- Dataset