**Why is the loss so high? (e.g., 100,000 vs 0.004)**

**Answer:**

We're predicting **x, y positions in pixels**, where:

* X ranges up to 1920
* Y ranges up to 1080

So even small prediction errors (like 10 pixels) lead to **high squared error values** (e.g., 10² + 10² = 200).

Multiply that by:

* 150 objects
* 5 future frames
* Per sample  
  → and it's normal to get losses in the **thousands** or more.
* Even a ~3-pixel error per coordinate →  
  150 objects × 5 frames × (3² + 3²) = ~13,500 loss

**What this *actually* computes:**

* It computes **per-coordinate squared error**, masked for only valid targets.
* Then it **sums** over all predictions, and divides by the number of valid entries (mask.sum()), which gives you:

**Mean squared error over all valid (x, y) coordinates.**