MACHINE LEARNING SYSTEMS

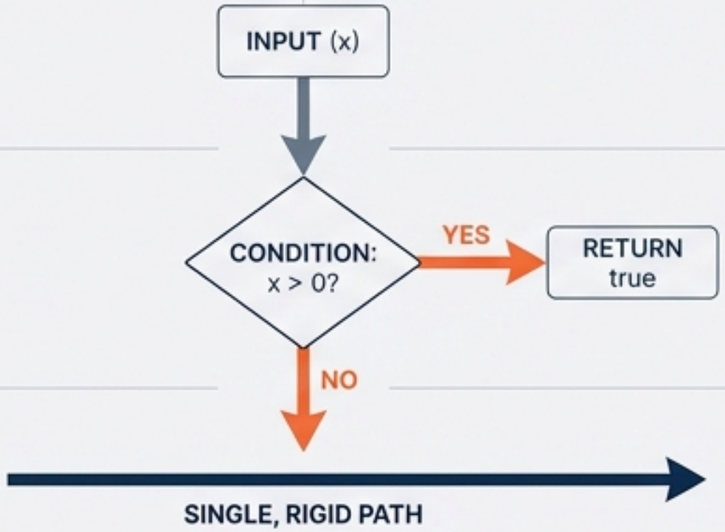
From Theory to Production: A Systems Engineering Approach

AI Engineering is not just about the model. It is the discipline of designing the data pipes, the silicon, the energy grid, and the deployment infrastructure that makes intelligence reproducible.

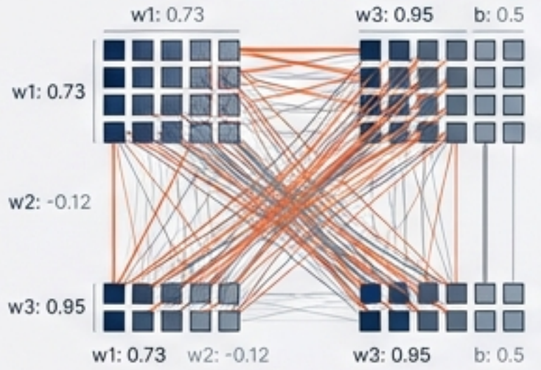
The Paradigm Shift: Logic vs. Optimization

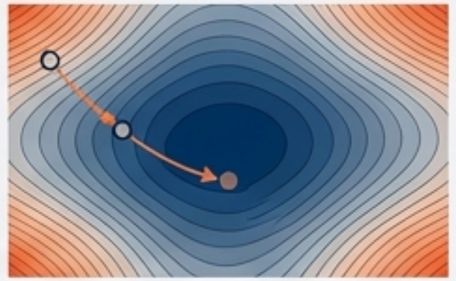
Software 1.0 (The Old Way)

```
if (x > 0) {
  return true;
}
```

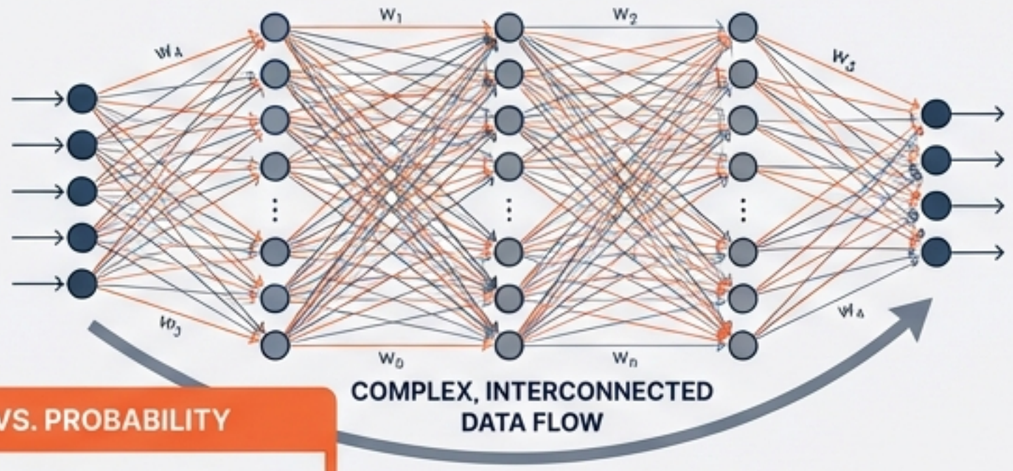


Software 2.0 (The ML Way)





GRADIENT DESCENT: OPTIMIZATION
WEIGHT MATRIX & GRADIENT LANDSCAPE



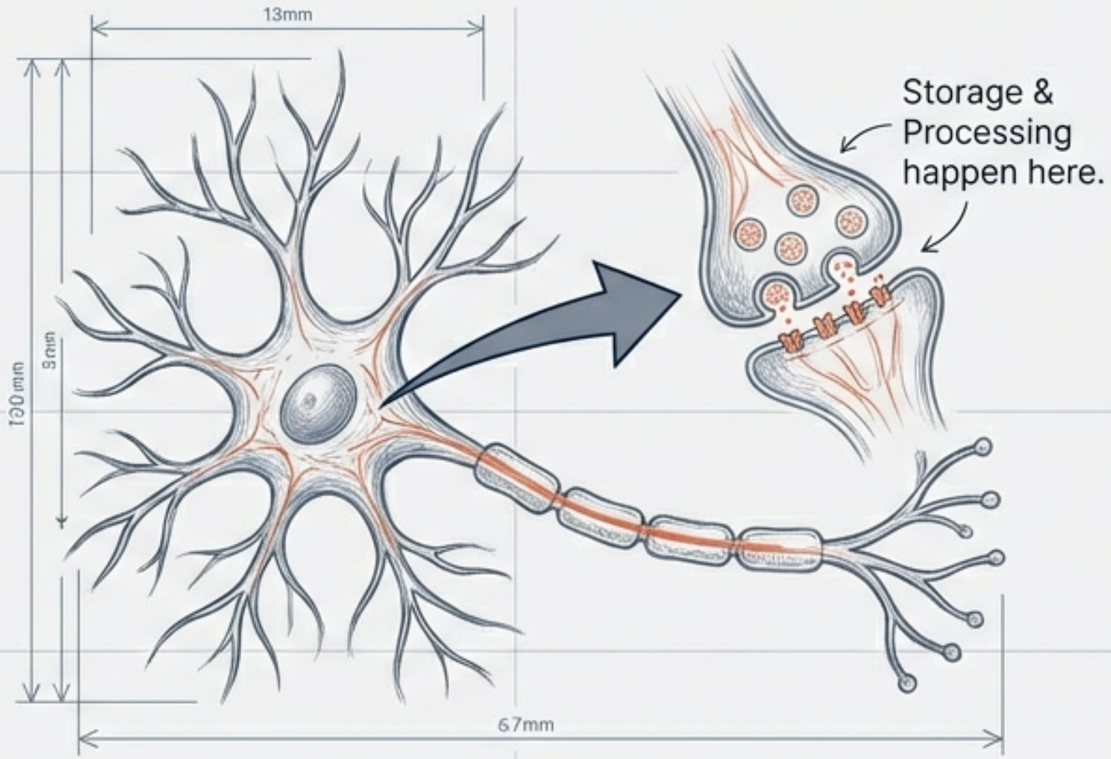
COMPLEX, INTERCONNECTED DATA FLOW

KEY DIFFERENCE: CORRECTNESS VS. PROBABILITY

Traditional SE relies on correctness and determinism.
ML Systems operate on probability and uncertainty.
Maintenance shifts from modifying code to updating data.

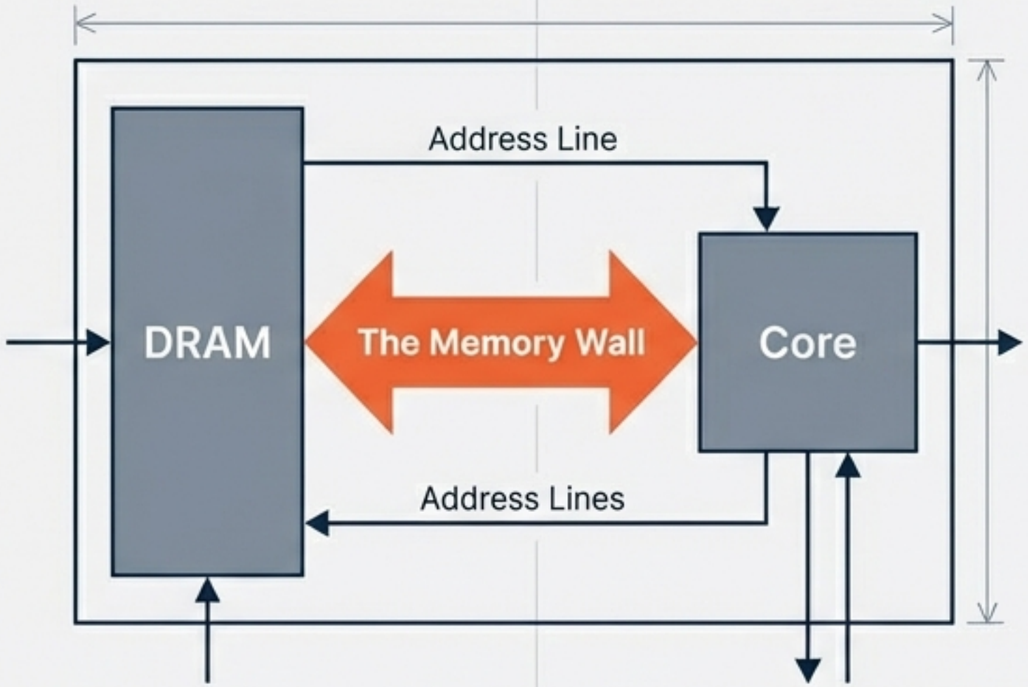
Biology to Silicon: The Memory Wall

Neuron



Integrated Memory & Compute

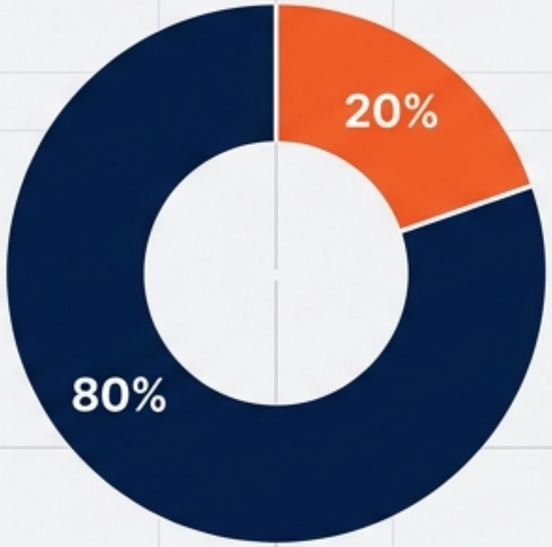
Von Neumann architecture



Separated Memory & Compute

Biological neurons integrate storage and compute.
Silicon isolates them. This architectural divergence is why
general-purpose CPUs fail at deep learning scale.

Data Engineering is 80% of the Work



Algorithm Refinement
Inter | JetBrains Mono

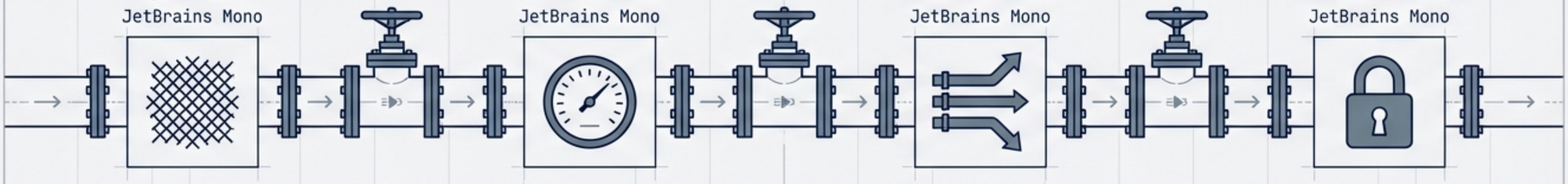
Data Collection & Cleaning
Inter | JetBrains Mono

QUALITY
JetBrains Mono

RELIABILITY
JetBrains Mono

SCALABILITY
JetBrains Mono

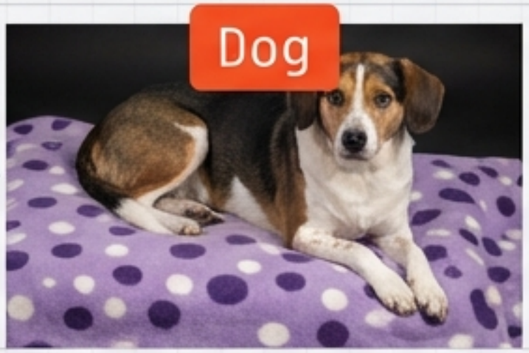
GOVERNANCE
JetBrains Mono



The ‘data preparation tax’ includes handling missing values (90% of real-world datasets) and resolving inconsistencies.”

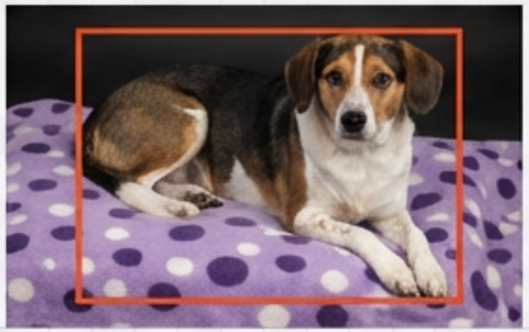
The High Cost of Ground Truth

Classification



$0.05/label. Cheap.

Bounding Box



$0.20/box.
Moderate cost.

Segmentation

$50.00/image. Expensive.
100,000x storage increase.

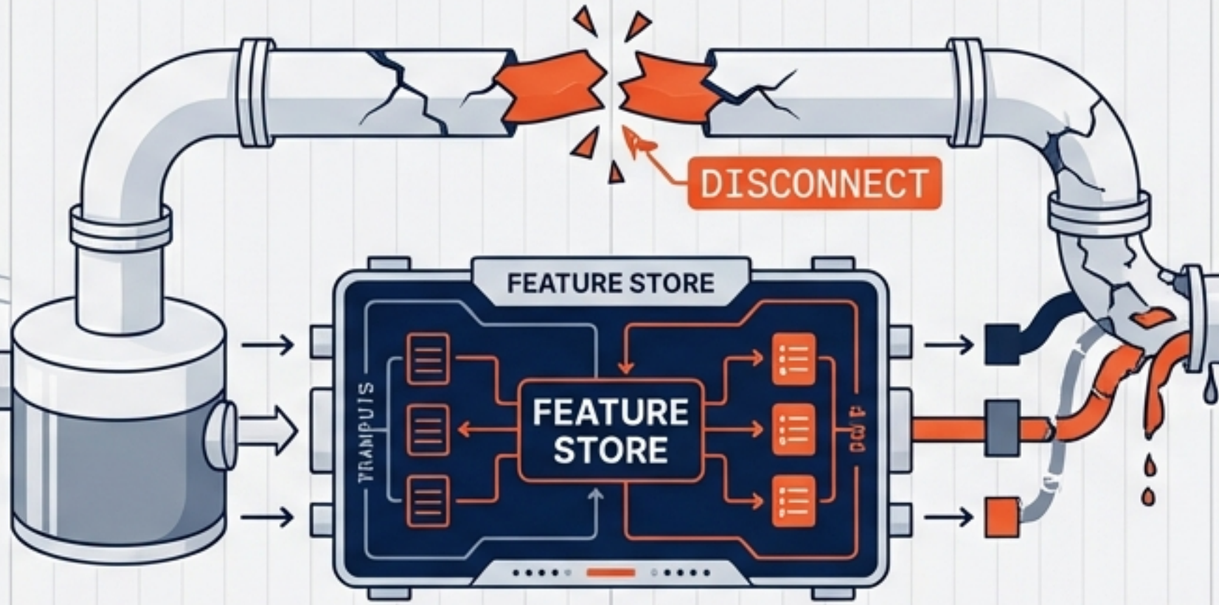
Strategy: Active Learning. Only label the 5-15% of ambiguous cases that confuse the model.

The Silent Killer: Training-Serving Skew

Training (Batch)

Inter

Controlled Environment.
Perfect, Historical Data.
Batch Processing.



Serving (Stream)

Inter

Uncontrolled Environment.
Noisy, Real-Time Data.
Stream Processing.

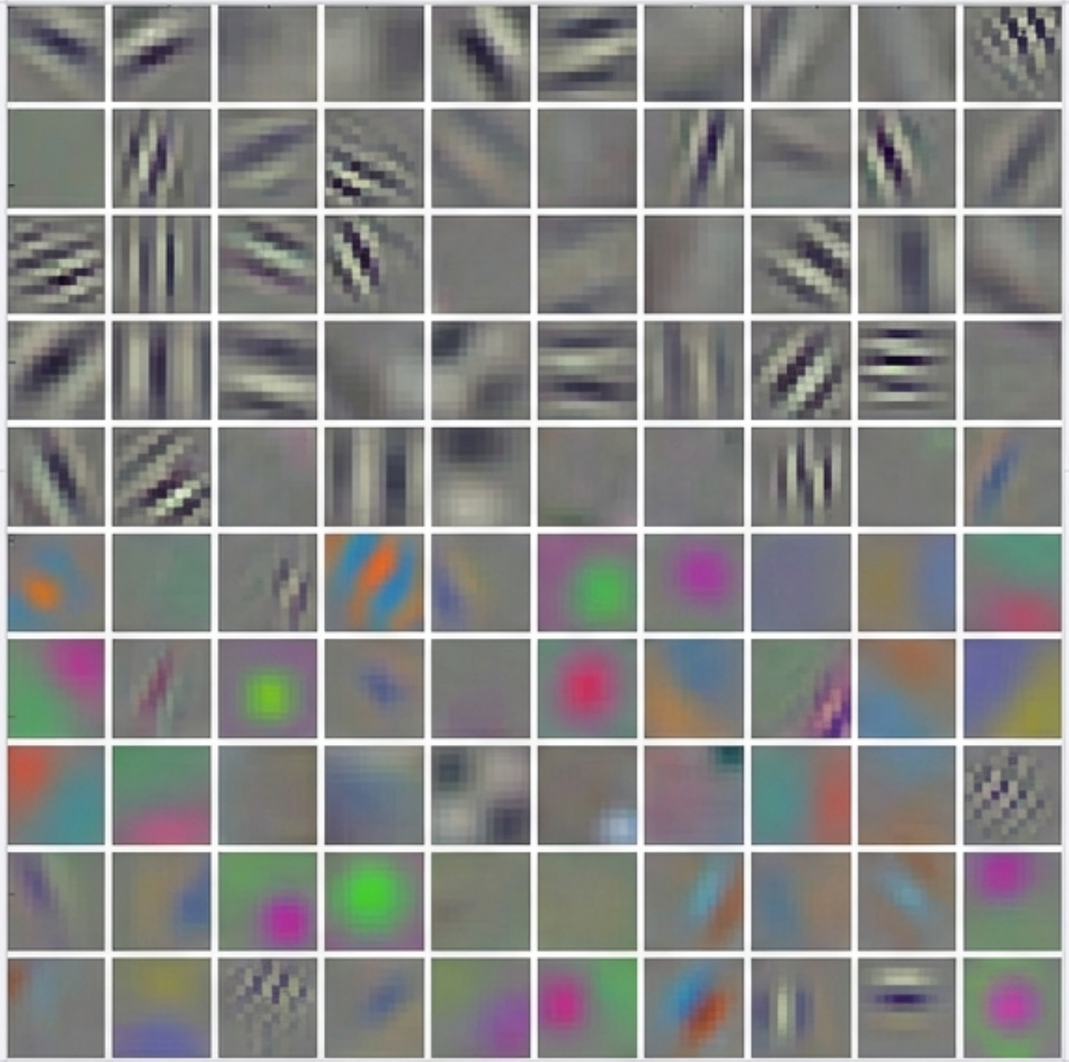
Engineering Solution: Idempotency.

Switch **ON** = Always ON
Consistent State.

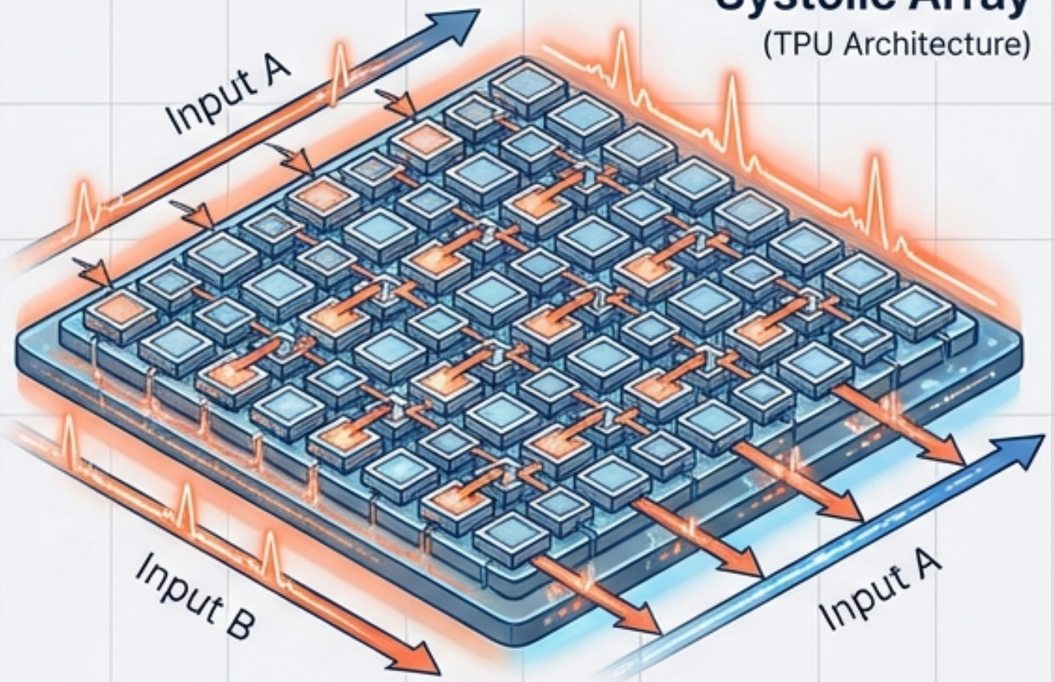
Toggle = Unpredictable
Inconsistent State.

Re-running a pipeline must produce the exact same features.
Use tools like TensorFlow Data Validation to detect drift.

The Heartbeat: Matrix Multiplication (MatMul).



Convolutional Kernel Weights

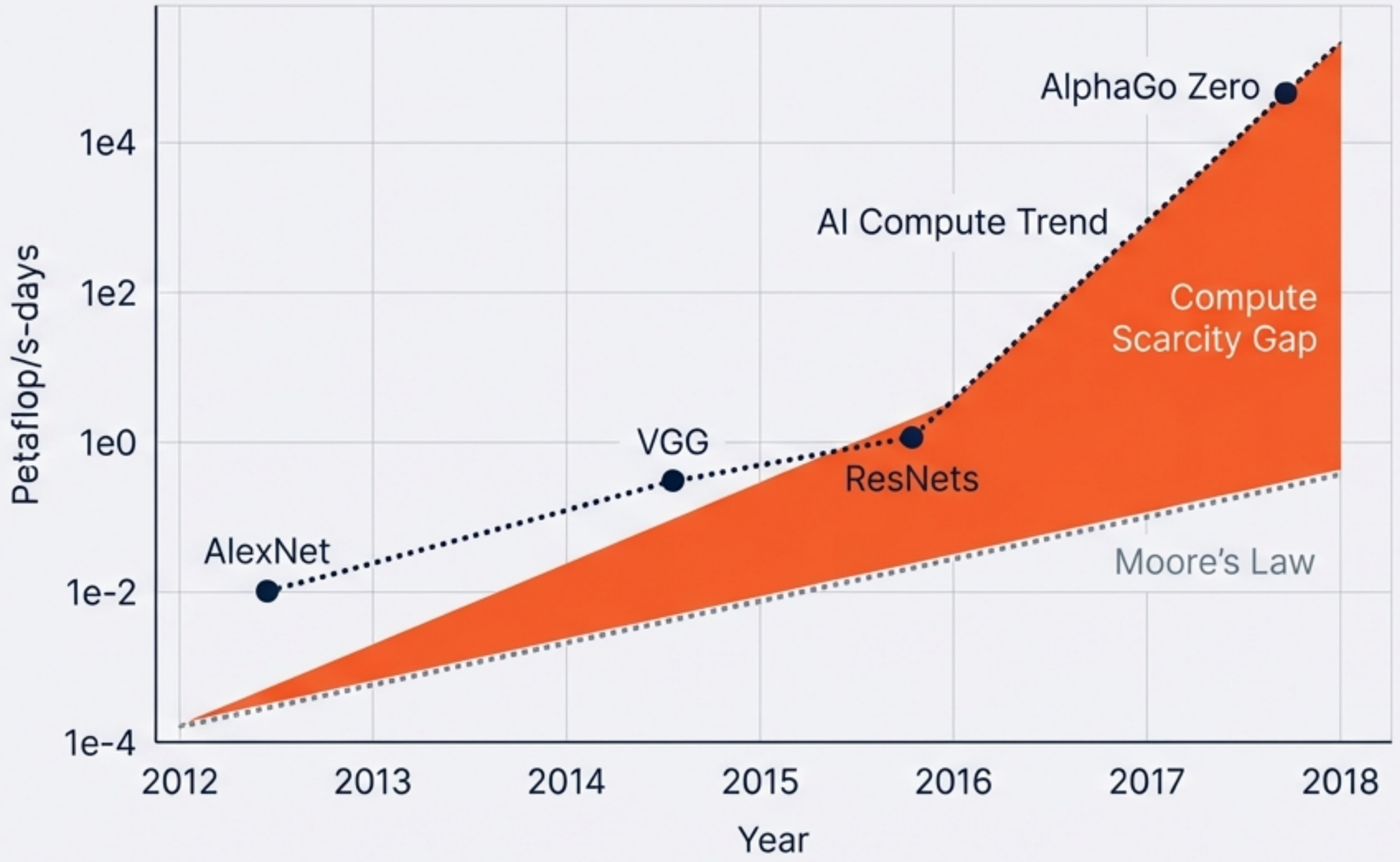


~~C[i,j] += A[i,k] * B[k,j]~~

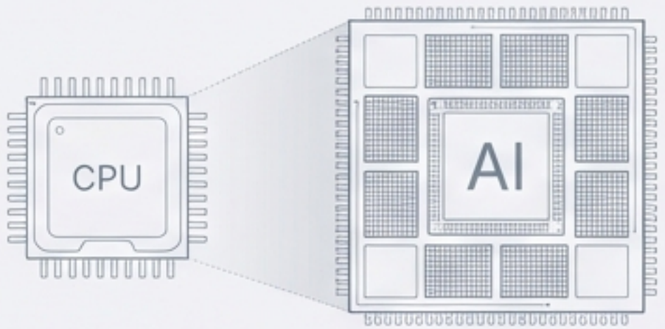
Hardware Accelerated MatMul

Neural networks are massive chains of matrix multiplication. CPUs handle loglc; TPUs/GPUs are architected to parallellze this specific math.

The Bitter Lesson: Compute Scarcity.



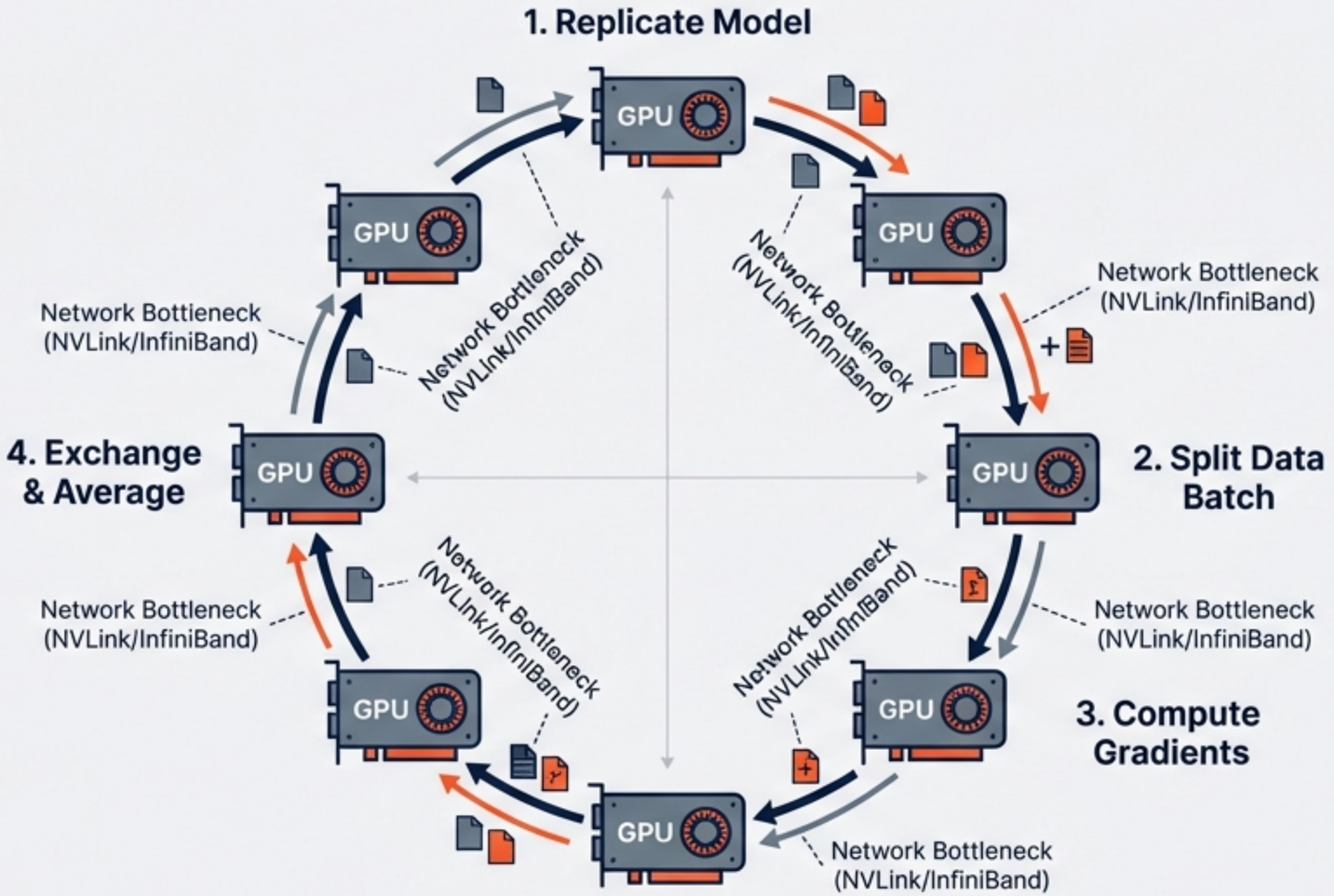
Compute **demand doubles every 3.4 months**. Moore's Law **doubles every 2 years**. You cannot run modern AI on commodity hardware.



Commodity Hardware Inadequate

Specialized Accelerators Required

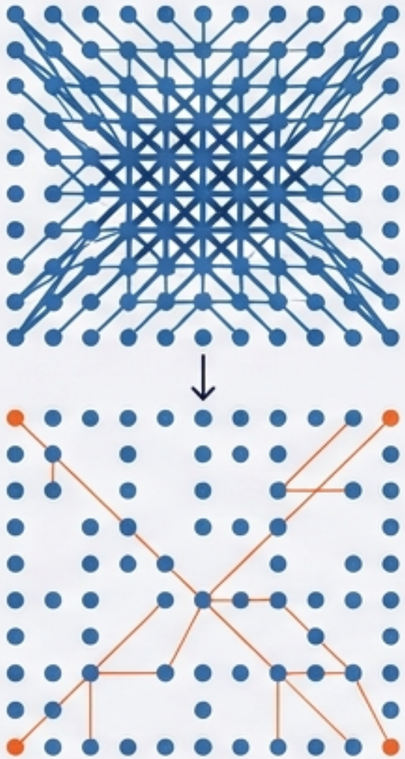
Distributed Systems: The Ring All-Reduce



When the model doesn't fit on one chip, the network bandwidth becomes the speed limit.

Optimization: Making the Model Fit.

Pruning

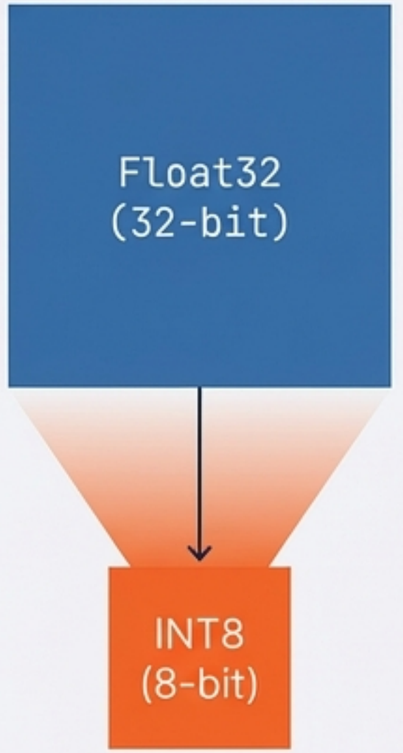


Removing 50-90% of weak connections.

CONCEPT ART: MODEL CONSTRUCTION

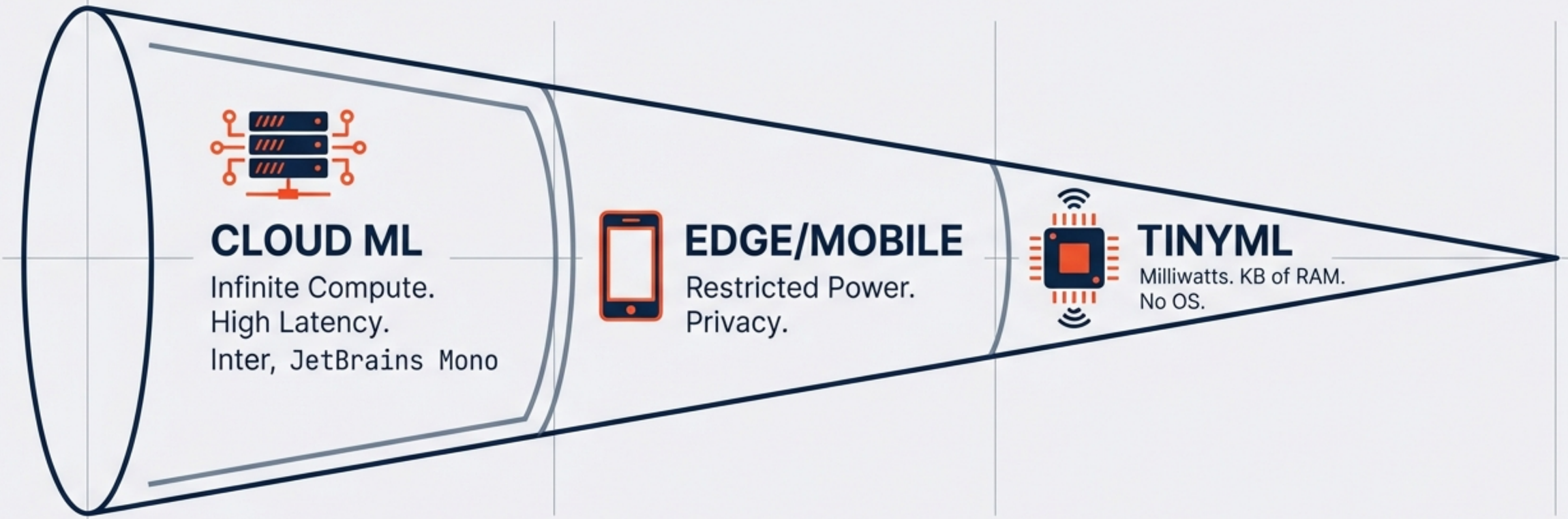
Optimization is the trade-off between Size, Latency, and Accuracy.

Quantization



4x memory reduction.
Faster math.

The Cone of Deployment



DECISION FRAMEWORK: PRIVACY > LATENCY > CONNECTIVITY.

Frameworks: Eager vs. Graph Execution.

Imperative (PyTorch)

```
import torch

x = torch.randn(3, 3)
w = torch.randn(3, 3)
b = torch.randn(3)

# Eager execution phase - computation happens immediately
for i in range(100):
    y = torch.matmul(x, w) + b

    # Gradient computation happens in-line
    y.backward(torch.ones_like(y))

    # Access intermediate values directly
    if i % 10 == 0:
        print(f"Step {i}, Loss: {y.sum().item()}")
```

Run as you write. Easy debug.

Symbolic Graph (TensorFlow/JAX)

```
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

# Graph definition phase - no actual computation
x = tf.placeholder(tf.float32, shape=())
y = tf.placeholder(tf.float32, shape=())

# Define computation symbolically
z = x * y
w = z + x
loss = w**2

# Symbolic gradient computation during graph construction
gradients = tf.gradients(loss, [x, y])

# Execution phase - actual computation occurs
with tf.Session() as sess:
    # Same graph can be executed multiple times efficiently
    for step in range(1000):
        grad_vals, loss_val = sess.run(
            [gradients, loss], feed_dict={x: 2.0, y: 3.0}
        )
    # Optimized execution with compiled kernels
```

Define, then Compile. Hard debug, massive optimization.

Production Reality: Develop in Eager, Export to Static Graph (ONNX).

Reliability: The Fragility of Fragility of Patterns

Prediction: **Panda** (57.7%)

+

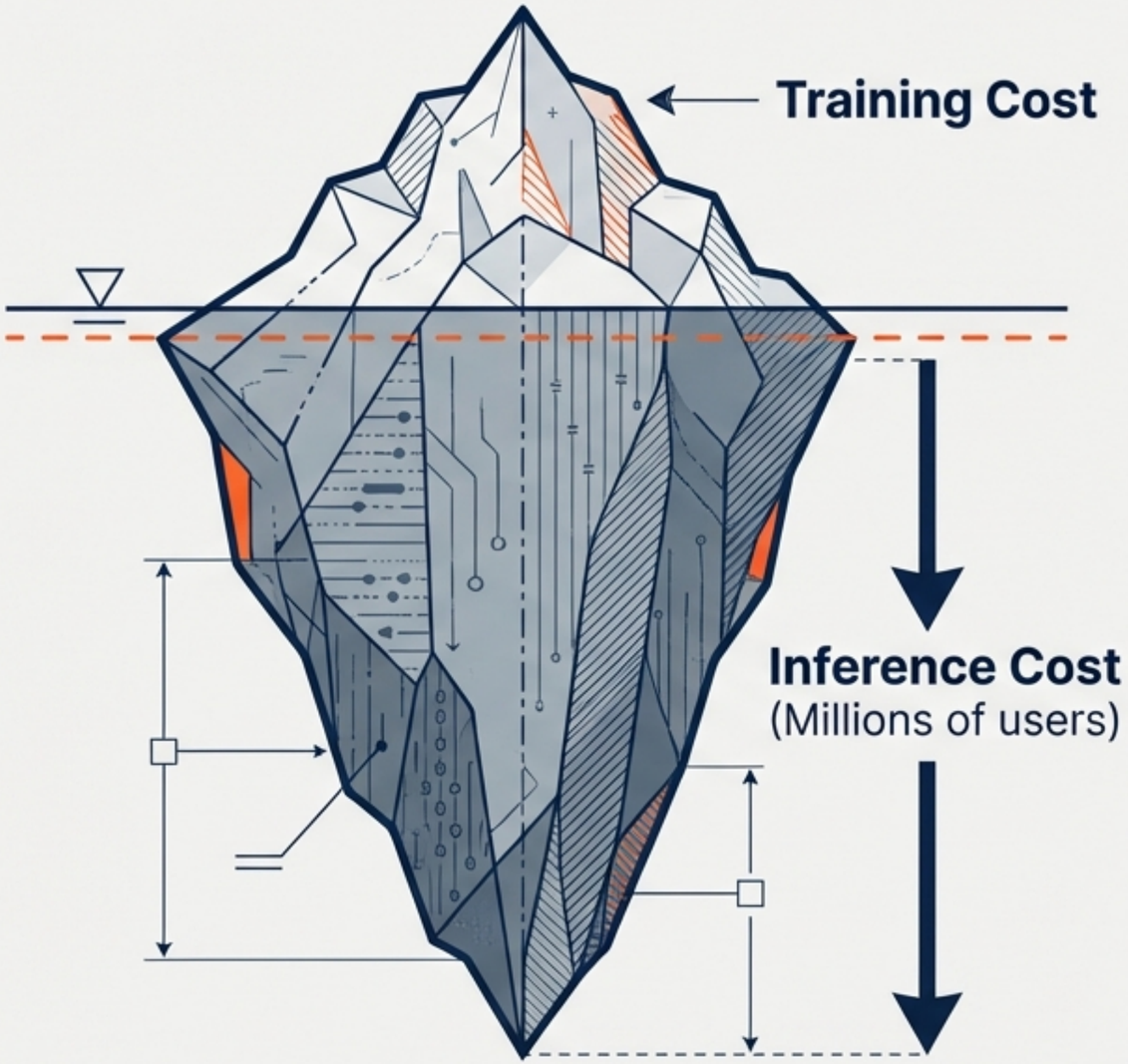
Adversarial Noise

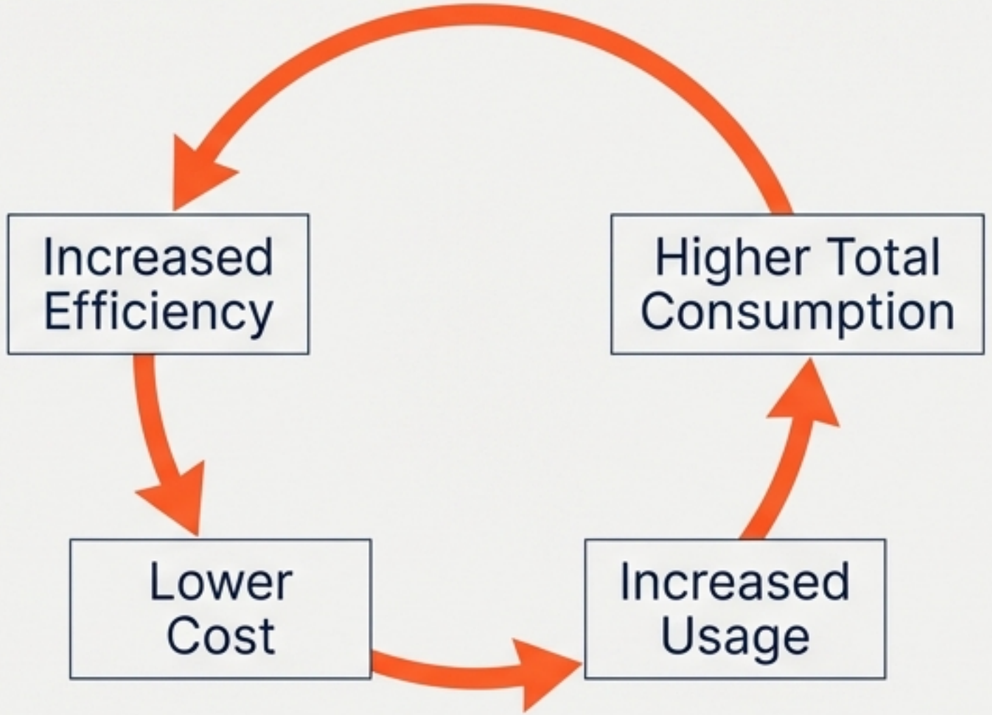
=

Prediction: **Gibbon** (99.3%)

Learned patterns rely on **pixel correlations** humans ignore.
This makes them **fragile to invisible attacks**.

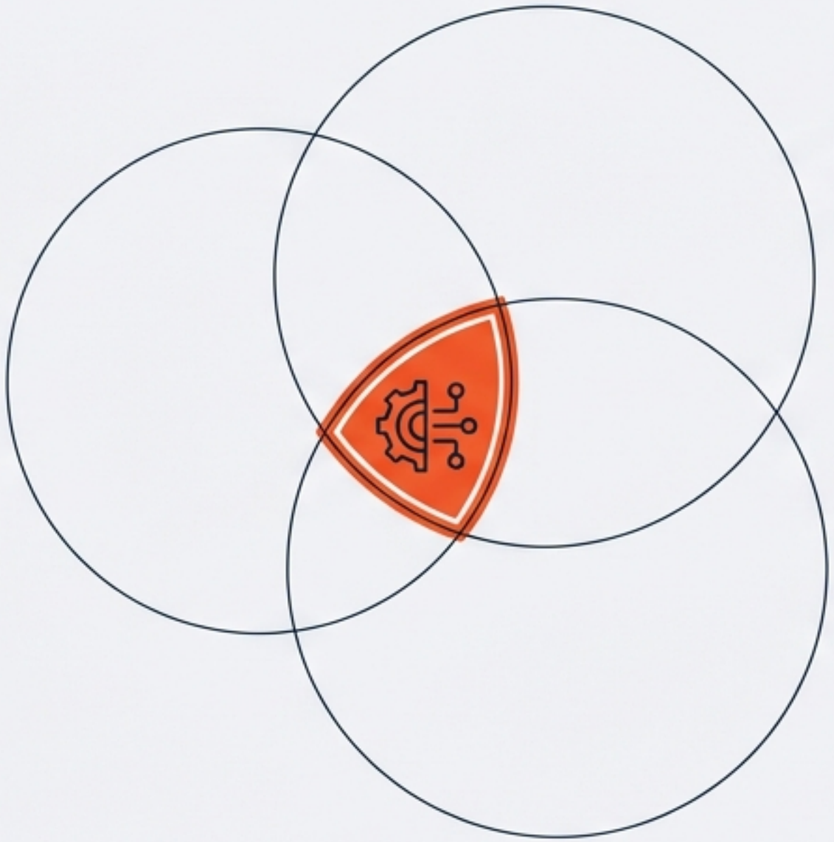
The Environmental Bill & Jevons Paradox





Sustainable AI requires focusing on **inference optimization**. Efficiency doesn't always save energy—it often induces more demand.

The Holistic Engineer



Systems Engineering
(Hardware, Scale)

Data Engineering
(Quality, Governance)

Algorithmic Science
(Models, Accuracy)

Success is not defined by the accuracy of the algorithm,
but by the reliability of the system.