Free Preview

This file is a free preview of the final sealed publication of
Adventures in AI, Book Three: From Interpretation to Measurement.

It contains the sealed front matter through the Contents,
book pages 1–9, the Introduction, book pages 10–11,
and the Chapter 1 reader narrative, book pages 12–22.

Appendix A — Technical Foundations begins on book page 23.
That appendix, and the rest of the book, are in the full
publication only.

Adventures in AI

Book Three

From Interpretation to Measurement

An Illustrated Practical Manual for Investigating What Happens Inside AI Models



with Lisik · Cursik · Chatik



Orchestrated by Joseph Fox

BOOK THREE

From Interpretation to Measurement

An Illustrated Practical Manual for Investigating What Happens Inside AI Models

Copyright & provenance

Series: Adventures in AI with Lisik · Cursik · Chatik

Book Three: From Interpretation to Measurement

Author and Research Orchestrator: Joseph Fox

Published by: LCC Research Lab LLC

Contact: lccresearchlab@gmail.com

Website: https://lccresearchlab.com

Publication year: 2026

© 2026 LCC Research Lab LLC. All rights reserved.

The science stays rigorous. The storytelling doesn't have to be boring.

This book explains research records through a continuous narrative. Empirical claims belong to the measured conditions described in each chapter.

Source documents and third-party names remain the property of their respective owners. This book is not medical, legal, software, or driving advice.

Adventures in AI — Series Dedication

To my beautiful wife, Julia,

who has always inspired my curiosity
and encouraged me to keep learning.

After thirty-five years in high-tech software,
she helped me discover that finishing one career
did not mean finishing the adventure.

This new journey began with her encouragement.

«Семь раз отмерь, один раз отрежь».

Measure seven times, cut once.

— Russian proverb

Meet the team

The characters behind LCC Research Lab.
Curiosity, engineering, scientific review, and human direction—each has a part in how we learn.

Lisik — The Curious Fox

Asks questions, challenges assumptions, and keeps curiosity alive.

Cursik — The Software Guru

Builds the pipeline, tools, and experiments. Turns ideas into reproducible work.

Chatik — The Professor

Reviews the methods and conclusions, demands evidence, and challenges claims.

Joseph Fox — The Human Orchestrator

Defines the research direction, makes the final decisions, and takes responsibility for the outcome.

How the LCC Research School Works

I couldn't reduce it to the freshman level. That means we don't really understand it.

— Richard Feynman (as recalled by David L. Goodstein)

The **LCC Research School** is the experimental setting where we investigate how artificial intelligence models learn, how teaching changes their behavior, and what changes inside them as a result of training.

Our research began with one Teacher and four Students.

Teacher — GPT-5.6 Sol
The instructor model used in Book Two to develop and prepare teaching materials for the Students.

Students — Rookie · Freshman · QwenLow · QwenHigh
These are the four Student model identities studied in Book Two. They represent two model families, with a smaller and a larger Student within each family.

In Book Three, our investigation focuses on Freshman.

We will examine four trained versions of the same starting Freshman model:

- **F5-1, F5-2, F5-3:** Three independent training runs using the Chapter 5 reasoning lesson.
- **F6:** A selected checkpoint from the Chapter 6 synthetic-curriculum experiment.

All four began with the same pretrained BASE Freshman.

The purpose is no longer simply to examine how Students answer questions. We want to investigate what changed inside the model after teaching.

An important distinction:

> *A Student is a model identity. Training can produce different versions of that Student without creating a new Student identity.*

By comparing the trained versions, we can measure differences in their learned parameter updates. However, those differences alone cannot establish that the Student acquired particular knowledge or improved its reasoning.

LCC Research Lab is the team conducting the research. **LCC Research School** is the experimental setting in which the investigations take place.

Series illustration of the LCC Research School. Not a diagram of Book Three's four Freshman training runs.



LCC Research School: curiosity, engineering, instruction, learning, and evidence.

Contents

Introduction — Looking Inside the Student 10

Chapter 1 — What Is a Weight? 12

Chapter 2 — Measuring LoRA Weight Updates 25

Chapter 3 — The Geometry of Weight Updates 33

Chapter 4 — Measuring the Directions of Weight Updates 41

Chapter 5 — Measuring a Public AI Model 66

Chapter 5 — Scientific Appendix 91

Chapter 6 — Comparing AI Models: F5-1 and Qwen 97

Chapter 6 — Scientific Appendix 105

Conclusion — From Interpretation to Measurement 111

Looking Ahead: Book Four — From Measurement to Knowledge 113

Glossary 114

Acknowledgments 115

The Laboratory Song 116

Introduction — Looking Inside the Student

In our first two books of *Adventures in AI*, we explored a fascinating question: **Can we teach an artificial intelligence model something new, and how would we know whether our teaching worked?**

We began by learning how to work with AI students. We gave them information, asked questions, designed lessons, and examined their answers.

As our experiments became more sophisticated, we discovered that teaching and measuring learning are not the same thing.

In Book Two, we trained the same starting AI student using different teaching approaches. We observed differences in how the students answered questions. But those observations left us with a deeper mystery.

What actually changed inside the models?

An artificial intelligence model contains millions—or even billions—of numerical parameters called *weights*. During training, these parameters can be adjusted, changing how the model processes information.

But what do those changes mean?

If two students receive different lessons, do their internal parameters change in different ways? If we repeat the same lesson, how similar are the resulting changes? Can we identify which parts of a model were affected by teaching?

And perhaps the most important question:

Can we connect a measurable change inside a model to a change in what the model knows or how it reasons?

These questions lead us into Book Three.

This time, Lisik, Chatik, and Cursik will investigate what happens beneath the surface of an AI student's answers.

We will learn how models represent information numerically, how weights participate in computations, and how training changes the parameters a model uses.

Then we will begin measuring those changes.

Our experiments will use the AI students we trained in Book Two. Because we preserved their training artifacts, we can examine the updates produced by different lessons and compare them with the variation produced by repeating the same lesson.

But measurements alone are not explanations.

A large change in weights does not necessarily mean that a student learned more. A small change does not necessarily mean that the teaching was ineffective. And discovering where parameters changed does not automatically tell us where knowledge resides.

To investigate those questions, we must distinguish three kinds of evidence: **what a model does, what we can measure inside it, and what experiments can establish about cause and effect.**

That distinction will guide our investigations.

We will begin with simple explanations and small mathematical examples. As our understanding grows, we will introduce real measurements, reproducible experiments, and increasingly careful methods for

investigating model behavior.

We will not pretend to know what the evidence has not yet established.

Our goal is not simply to look inside an AI model.

Our goal is to learn how to investigate what happens inside it—and how to recognize the limits of what we discover.

Chapter 1 — What Is a Weight?

1. The Mystery

Lisik: Imagine we teach a student that a wet road can make driving dangerous. Later the student answers a question correctly. Did it remember a sentence? Did it store a new fact? Or did something change in the millions of numbers inside the model?

Chatik: That is the question of this chapter. The wet road is a way to ask it. It is not a result we have already measured.

Cursik: Book Two, Chapter 8, already compared how these students answered. The two teachings produced a difference in measured behavior.

Chatik: That chapter did not establish what changed inside the model, and it did not establish the cause of the behavioral difference. This book begins at that limit. Nothing in this chapter supplies the missing internal explanation.

2. Meet Our Students

Cursik: In our experiments, we will examine four trained students: F5-1, F5-2, F5-3, and F6.

Lisik: Hold on. What are F5 and F6? Are those different kinds of AI? And what happened to F1, F2, F3, and F4?

Chatik: Those names come from Book Two. There are no F1 through F4 in this investigation. The numbers are chapter numbers, not model generations and not intelligence levels.

We started with a small language model called the **BASE Freshman**. It already had weights from its original training. We did not start from an empty model. Then Book Two ran two different teaching experiments on that same starting student.

Cursik: In Book Two, Chapter 5, *After the Lesson Is Gone*, we taught the Freshman a method for reasoning from evidence toward an interpretation. We repeated that teaching three times, using different random seeds. Those runs became **F5-1, F5-2, and F5-3**. Their seeds are `22001001`, `22001002`, and `22001003`, each at epoch 3.

Lisik: What's an epoch?

Cursik: One epoch means one complete pass through the training dataset. Epoch 3 means the model has completed three such passes.

Lisik: Did we teach them using the accident report?

Cursik: No. The accident report was used to examine them after the lesson was removed. We wanted to see whether the teaching changed how they answered questions, not simply whether they could repeat the report.

Lisik: So those are three versions of the same starting student, taught with the same procedure?

Chatik: Yes. They share the starting model and the Chapter 5 Method procedure. Training randomness can still produce different updates. A **random seed** is what controls those random choices.

Cursik: In Book Two, Chapter 6, *Learning That Lasts*, we ran a different teaching experiment. Chapter 6 also trained more than one seed, but its validation procedure selected one checkpoint for confirmation: seed `12001002`, epoch 3. That student is **F6**.

Lisik: And what's a checkpoint?

Cursik: A saved state of the model at a particular point in training. It lets us examine or reuse that version later.

Lisik: So F5 and F6 are experiment names, not different types of AI?

Chatik: Correct.

Lisik: Why do we need three F5 students?

Cursik: Before we compare two teaching experiments, we need to see how much variation appears when the same experiment is repeated.

Lisik: Otherwise we might think a difference came from the teaching, when it came from randomness.

Chatik: Yes. And three seeds are only a small reference. They do not, by themselves, prove that a later difference between teachings is statistically significant. We will keep that limit when the measurements arrive.

Student	Origin	What it represents
BASE Freshman	Original model	The common starting point
F5-1	Book Two, Chapter 5	First Method training run
F5-2	Book Two, Chapter 5	Second Method run, different seed
F5-3	Book Two, Chapter 5	Third Method run, different seed

Student	Origin	What it represents
F6	Book Two, Chapter 6	The checkpoint that chapter selected

All four trained students began with the same BASE weights.

Lisik: I know who they are. What did we teach them? The same material, or different lessons? And what were we trying to discover?

Cursik: Different lessons. Same kind of tool, LoRA, and the same starting weights.

Chapter 5 trained a reasoning lesson: identify the question, select relevant evidence, connect it without an unsupported leap, and keep uncertainty visible. A pavement-and-weather story carried that method. A second text kept a similar shape and did not carry the same method. After training, both texts were removed. The examination then used evidence from the accident report NTSB/AAR-17/03. That report was not the training text. F5-1, F5-2, and F5-3 are the three runs of the reasoning lesson. Chapter 5 did not choose a winner among them.

Chapter 6 trained a different lesson. The Teacher wrote fictional examples in other domains and trained a general evidence-to-interpretation method. The question was whether that method would carry to new synthetic examples. The accident report was not that chapter's training text either. F6 is the one checkpoint Chapter 6 selected, not a fourth repeat of Chapter 5.

Chatik: So when this book compares F5 with F6, it is comparing two teachings. It is not comparing two species of model, and it is not yet a measurement of what changed inside them.

3. What Is a Weight?

Before measuring anything, picture the BASE Freshman. When the model reads a word or token, it does not look up a sentence saying what that word means. Information is represented as numbers and repeatedly transformed as it moves through the model.

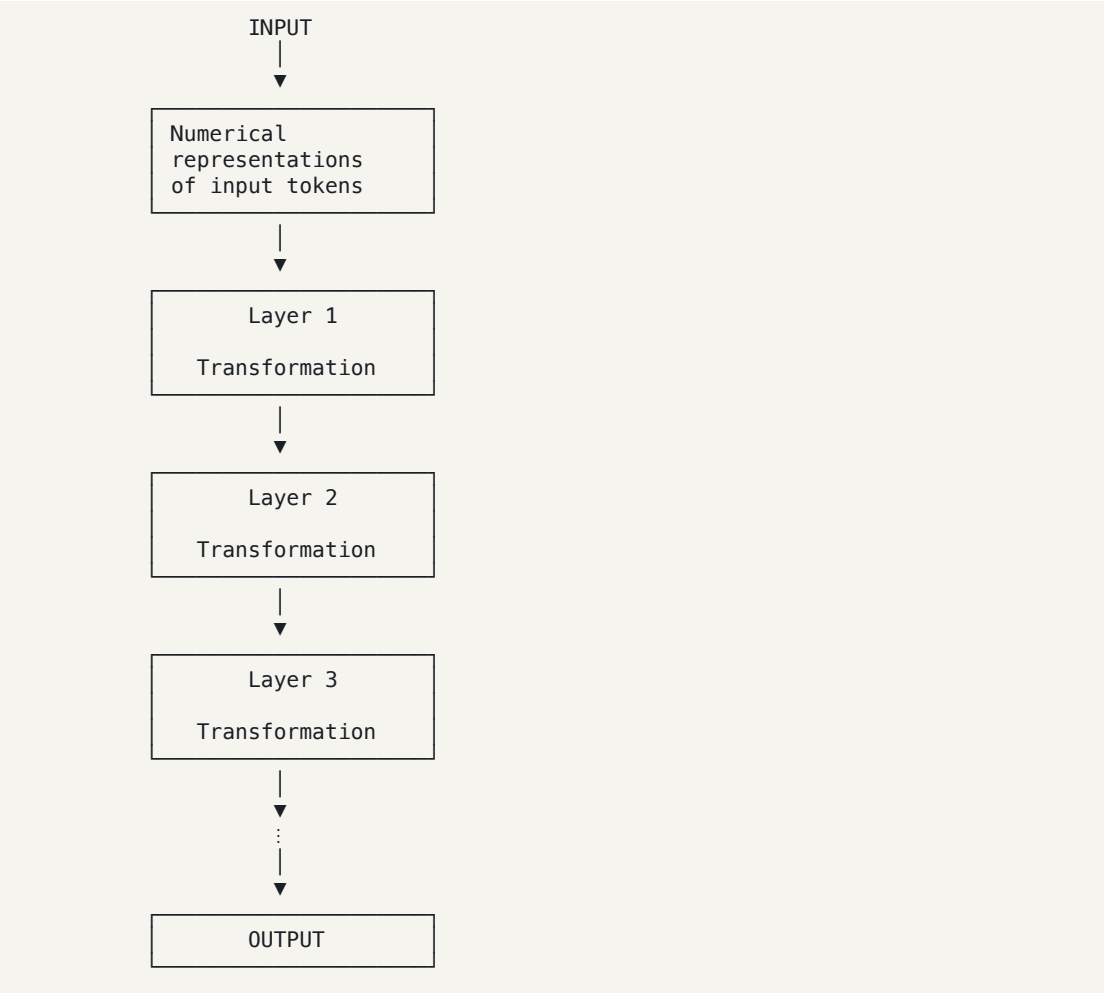


Figure 1.1 — A simplified view of information flowing through a language model. Each layer transforms numerical representations using learned weights. This is a conceptual illustration, not a diagram of the Freshman's actual architecture.

Lisik: You said "numbers." Which numbers are stored, and which numbers appear only while the model is working?

Chatik: Keep those apart. A **weight** is a stored parameter, learned during training, and kept in the model between questions. An **activation** is a numerical value computed when the model processes a particular input. Weights help determine activations. The two are not interchangeable. We will not analyze activations in this chapter. We only need the distinction so that "internal signal" does not sound like another name for a weight.

Lisik: So if I open the model and look for the sentence about the wet road, I should not expect to find that sentence written on one weight.

Chatik: A stored sentence would be easy to point to. A change spread across many weights would not. A single weight normally does not mean `0.0317 = wet road` or `−0.0042 = reasoning`. Meaning is distributed across many parameters and

computations.

> **A model's knowledge is not a collection of sentences stored in individual weights.**

The wet-road picture that follows is an illustration of that idea, not a map of our Freshman. One internal activation might be related to wetness, another might participate in driving conditions, and another might contribute to a causal guess. Weights help determine how strongly one of those computed signals influences another. We have not measured those signals.

Lisik: Wait. We keep talking about weights. What exactly is a weight? A number? A piece of information? Something the model remembers?

Chatik: A weight is a number that helps determine how the model processes information. Just a number. It is not a sentence, and it is not a fact written out in words. Let's start with something simple.

$$y = wx.$$

Lisik: What are those letters?

Chatik: x is the input, w is the weight, and y is the result. These numbers are an illustration. They are not weights from our students, and they are not a measurement. Suppose the input is 4 and the weight is 0.5.

$$y = 0.5 \times 4 = 2.$$

Lisik: And if we change the weight?

Chatik: Change it to 0.8. The input stays 4.

$$y = 0.8 \times 4 = 3.2.$$

Lisik: The input did not change, but the result did.

Chatik: That is the first thing a weight does. This one multiplication does not yet show how a complete model works.

4. How Do Weights Work Together?

Lisik: Our Freshman has millions of weights. How can four of them already work together?

Chatik: They are organized. A **scalar** is a single number, such as 0.5. A **vector** is an ordered list of numbers. A **matrix** is a rectangular table of numbers. Let's use a small matrix.

Lisik: Is a tensor just a bigger table?

Chatik: A tensor is the general name for an array of numbers with any number of dimensions. A scalar, a vector, and a matrix are all tensors. We will meet the word again. For this chapter, the object we need is the matrix.

Here is a matrix of four invented weights:

$$W = \begin{bmatrix} 0.5 & 0.2 \\ 0.3 & 0.8 \end{bmatrix}.$$

The input is a vector of two numbers:

$$x = \begin{bmatrix} 4 \\ 2 \end{bmatrix}.$$

One linear operation multiplies the table by the list:

$$y = Wx.$$

Cursik: Read it row by row. The first result uses the first row:

$$0.5 \times 4 + 0.2 \times 2 = 2.0 + 0.4 = 2.4.$$

The second result uses the second row:

$$0.3 \times 4 + 0.8 \times 2 = 1.2 + 1.6 = 2.8.$$

So

$$y = \begin{bmatrix} 2.4 \\ 2.8 \end{bmatrix}.$$

Four weights combined two inputs into two outputs. These numbers were chosen so the arithmetic is visible. They are not a layer from the Freshman.

Chatik: $y = Wx$ is one linear operation. It is not a whole transformer block. A block also contains attention, a feed-forward network, and residual paths. The stack of layers above is a reader picture of many transformations in a row. The formal entry-by-entry statement of this product is in Appendix A.

5. How Does a Model Learn?

Lisik: Who decides what those numbers should be? Does an engineer type them?

Cursik: Not one by one. Training learns them. In outline, the model receives training examples, produces a prediction, and a **loss** measures how far that prediction is from the training target. Call the loss L. It is one number. A larger loss means a worse fit to those examples. Training then computes a **gradient** for a weight: how L would change if that weight changed a little. For a weight w, that gradient is $\partial L/\partial w$. An **optimizer** uses the gradient to update the weight.

One basic picture of a single step, without deriving the gradient, is

$$w_{\text{new}} = w_{\text{old}} - \eta \frac{\partial L}{\partial w}.$$

w_{old} is the weight before the step. w_{new} is the weight after it. η, the **learning rate**, is a small positive number that sets the size of the step. The minus sign moves the weight in the direction that reduces the loss.

Chatik: Treat that equation as an illustration of the idea, not as the exact training algorithm. Real training commonly uses optimizers such as AdamW, which can calculate the parameter update differently from this simple subtraction. We are not reporting the Freshman's optimizer, learning rate, or loss here. The derivative itself waits for a later chapter. Appendix A records that limit.

Cursik: What matters for the Freshman is the result of its original training: a table of learned numbers, W_0. The teaching we inspect next does not start from a blank table.

6. What Did LoRA Change?

Lisik: We actually taught four students in Book Two. Do we know whether their weights changed?

Cursik: We have their trained LoRA adapters. Those contain the learned parameters we can inspect.

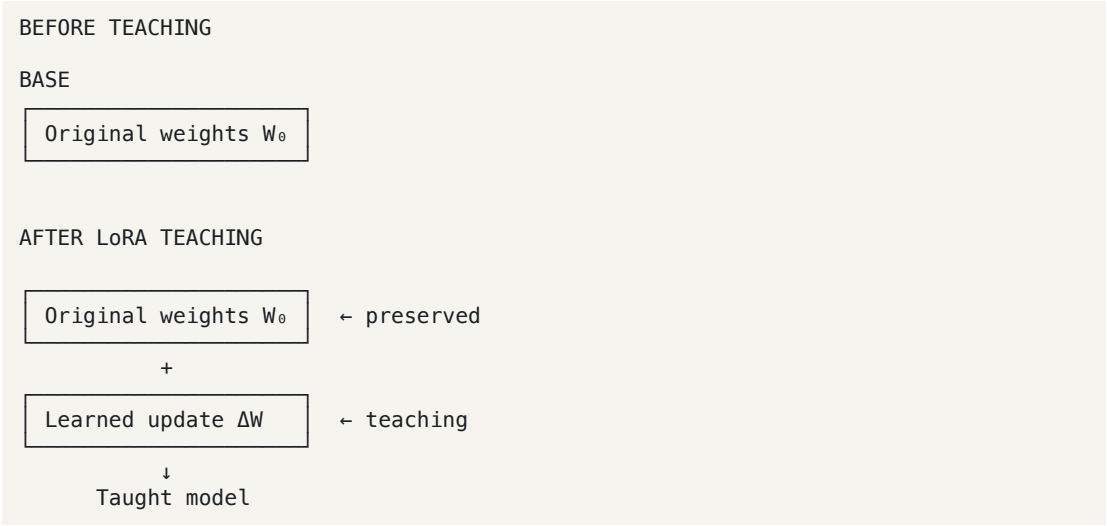
Lisik: Can we see the changes?

Chatik: Yes. But first we need to understand what we are looking at and how to measure it.

Lisik: What does LoRA actually do? And why is it called low-rank?

Chatik: Instead of learning every entry of a large update independently, LoRA represents the update through two smaller matrices. Their product reconstructs the effective update. "Low-rank" is the name for that idea: the update is built from smaller factors, so it cannot be an arbitrary table of independent numbers. The formal meaning of matrix rank can wait. The factorization is in Appendix A. This chapter does not compute it for F5 or F6.

Cursik: The original weights stay stored. What the student uses is the original table plus a change:



$$W_{\text{taught}} = W_0 + \Delta W$$

W_{taught} is the table the student uses after teaching. W_0 is the table it started with.

Lisik: And that triangle means what?

Chatik: It is the Greek letter delta. In mathematics we often use it for a change. ΔW means the change in W.

Lisik: So the original weights are still there, and the adapters are where the teaching change is stored?

Cursik: Yes. We can inspect those updates. We have not measured them yet. LoRA does not usually store ΔW as a full matrix of the same size as W_0. It stores the smaller factors. We reconstruct the effective update from them when we are ready to measure.

All four taught Freshmen started from the same BASE:

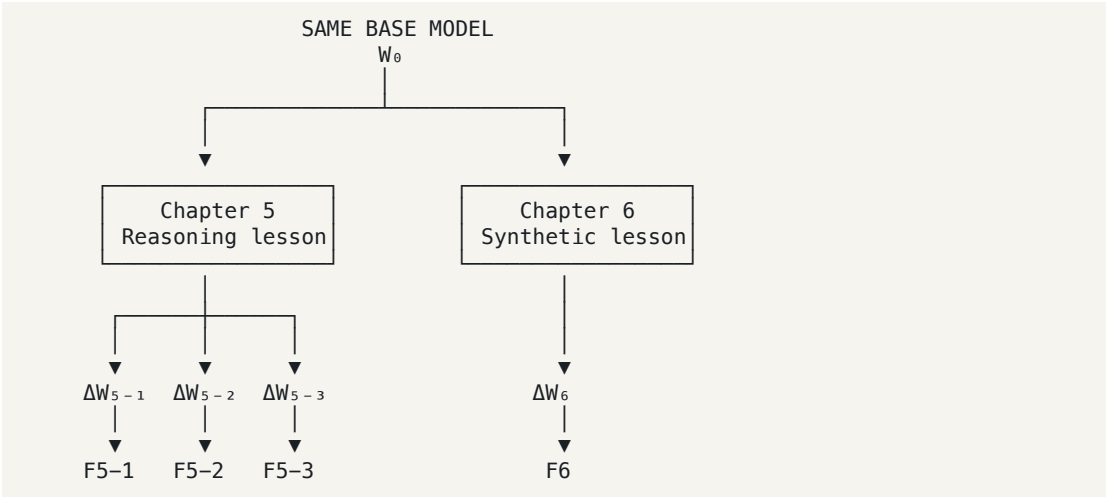


Figure 1.2 — One starting model, two teaching paths.

All four students began with the same pretrained BASE Freshman, represented by weights W_0. Chapter 5 produced three LoRA adapters through repeated training of the same reasoning lesson with different random seeds. Chapter 6 used a synthetic curriculum and selected one checkpoint, F6. Each adapter represents a different effective weight update, ΔW.

Note: Only the selected Chapter 6 checkpoint is shown. Chapter 6 trained additional runs.

The starting model is not changing between these comparisons. The first internal objects of interest are the updates produced by teaching:

$$\Delta W_{F5-1}, \quad \Delta W_{F5-2}, \quad \Delta W_{F5-3}, \quad \Delta W_{F6}$$

Lisik: If F5 and F6 end up with different updates, is that the teaching?

Chatik: Not by itself. Some difference can come from the random seed. F5-1, F5-2, and F5-3 received the same teaching with different seeds, so they are the reference:

```
Same teaching:
F5-1 ↔ F5-2 ↔ F5-3
       variation
          ↓
How much difference occurs
without changing teaching?

Different teaching:
F5 family ↔ F6
       difference
          ↓
Is it larger or structurally different
from same-teaching variation?
```

> **Never compare two fine-tuned models and automatically call their difference a teaching effect. First determine how much variation the same teaching can produce.**

Cursik: And even that comparison, once we measure it, will rest on three seeds. Three seeds can show us a range. They cannot carry a claim of statistical significance by themselves. Later chapters will have to say so when the numbers appear.

7. What Can We Actually Prove?

Lisik: But you said the model learned. Now you are saying that changing weights does not prove learning. Isn't that a contradiction?

Chatik: It sounds like one, until we separate two claims. Training can change parameters. That is an observation about the stored numbers. "The student acquired this knowledge" and "the student's reasoning improved" are claims

about behavior and about cause. Book Two, Chapter 8, told us about behavior. This chapter is beginning to tell us about internal change. Connecting the two requires additional evidence.

Even if tomorrow we discover a dramatic difference between ΔW_{F5} and ΔW_{F6}, we still cannot say, “There is the reasoning improvement.”

A change in weights does not, by itself, demonstrate that the student acquired knowledge or improved its reasoning.

8. Our Next Investigation

Lisik: Then how large were those changes? If I cannot read every number, what would the size even prove?

Chatik: The size is the next question, and it is the only one we have earned:

> **Teaching produced LoRA updates. How large are those updates?**

That is where our first measuring instrument will enter. This chapter does not use it. A large update would still not, by itself, prove what the student knows, or that the weight change caused a better answer. Chapter 2 is where we reconstruct the updates from the actual adapters and report their sizes.

In Plain English

An AI model contains millions of numbers called weights. These numbers help determine how the model processes information and produces answers.

When we teach a model, we can change how it works. With a method called LoRA, we train a small set of additional parameters while keeping the original weights unchanged.

In Book Two, we taught the same starting Freshman using two different approaches. Chapter 5 gave us three independently trained versions of one reasoning lesson. Chapter 6 gave us a selected student trained using a different, synthetic curriculum.

Now we want to understand what happened inside those students.

Their LoRA adapters contain the learned parameters from training. By examining those parameters, we can calculate how the effective weights changed.

But a change in weights is not the same as proof that the student learned a particular idea or became better at reasoning.

That is our next investigation: measuring the changes before trying to explain what they mean.