NAIRR WORKSHOP · MEDICAL AI

Teaching AI to Read Medical X-Rays

A model trained on half a million X-rays learns to detect elbow abnormalities — in minutes, with limited data and a free GPU.

EVA-02
Pre-trained AI model

Fine-Tuning
Teaching a new task

MURA Dataset
Elbow X-rays

Tennessee Tech University
Dr. Jesse Roberts

OVERVIEW

What We'll Cover Today

1 **Start With the Data**

A few hundred X-rays and a real clinical question

2 **The Three Paths Forward**

Why two approaches fail and one succeeds

3 **Train Smarter, Not Harder**

The insight that makes this possible

4 **Why GPUs Matter**

The hardware behind fast training — and NAIRR

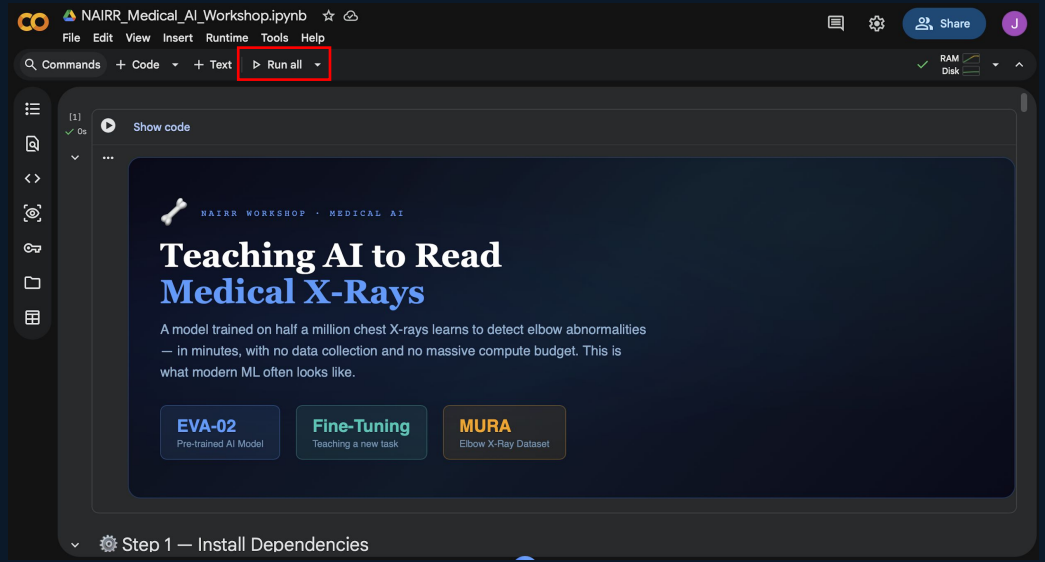
5 **Live Demos**

Fine-tune a real model on real X-rays
Look inside the randomness of an LLM

Let’s start the Notebook

The code takes a few minutes to run.

We’ll let it run while we talk. Open the link and select run all!



tinyurl.com/TnTech-LoRA-Xray

We’ll come back to the workbook in a moment.

THE PROBLEM

ML begins With the Data

Scenario: You are a public health researcher. You have collected a few hundred elbow X-rays from a local clinic — some normal, some showing fractures or arthritis. A radiologist labeled them. It took months. This is everything you have.

MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs

~400
Labeled X-rays you have collected

Binary
Normal or Abnormal — labeled by radiologists

MURA
Dataset of many different x-rays. Broken into subsets for finger, elbow, etc.

Potential goals

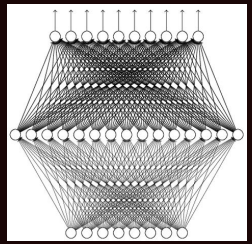
Build a tool that helps a doctor catch something they might have missed.
A second-opinion system that a patient in a rural area can access when no radiologist is available.
Something that democratizes access to expertise that has historically been locked inside large academic medical centers.

DEAD END 1

Could we train a powerful model from scratch?

You need a model complex enough to detect fractures, arthritis, and subtle abnormalities in X-rays.
So, you reach for something powerful like a large multi-layer neural network.

The problem



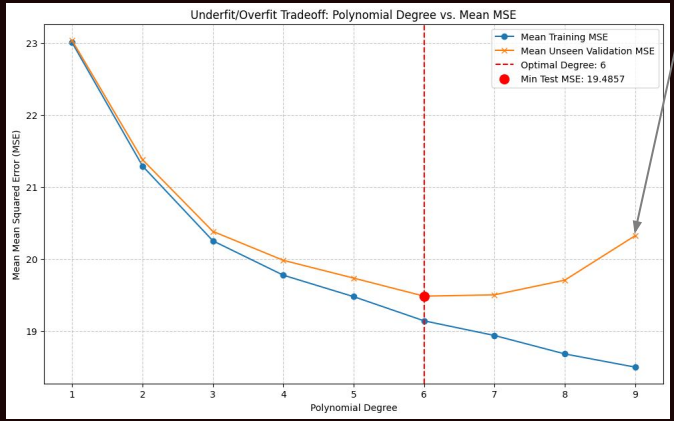
A powerful model has millions of parameters.

With only a few hundred training examples, it memorize the specific images rather than the underlying pattern.

It fits the noise — not the signal.

What happens

Memorization

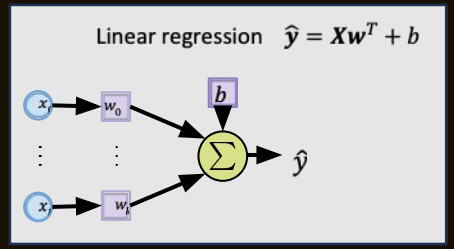


DEAD END — OVERFITTING · The model memorized your data. It hasn't learned anything generalizable about X-ray pathology.

What if we use a simpler model?

If a powerful model overfits on small data, the obvious fix is to use a smaller, simpler model. Fewer parameters — less ability to memorize.

The problem



Fewer parameters does limit memorization

But it also limits what the model can learn at all.

Detecting pathology in X-rays is genuinely hard — it requires real representational power.

What happens

Accuracy ceiling too low to be useful

Model can't distinguish subtle findings

No amount of training data fixes this

You've solved overfitting by creating a model too weak for the task

DEAD END — UNDERFITTING You traded one problem for another. The model generalizes — but it can't learn the task.

CORE CONCEPT

Alternative option: Train Smarter, Not Harder

Foundation models are large models trained on a tremendous amount of related data.
An example for language is chat-GPT

Chest x-rays are very different from elbow x-rays. But many of the skills of xray eval may transfer!

OVERFITS

Large model + small dataset

The model is powerful enough to memorize your few hundred examples rather than learn the underlying pattern. It performs perfectly on data it has seen and fails on every new X-ray you show it. More training makes this worse, not better.

UNDERFITS

Small model + small dataset

Reducing model size limits memorization — but it also limits what the model can learn at all. It simply doesn't have the capacity to recognize complex pathology no matter how long or hard you train it. You've traded one problem for another.

JUST RIGHT

Fine-tune a pre-trained foundation + small dataset

Only ~4% of parameters update — not enough room to memorize your limited data. The frozen 96% already understands X-ray structure from 520,000 images of pre-training. Small updates on top of deep knowledge. Train smarter, not harder.

THE MODEL

Meet EVA-02

A Vision Transformer pre-trained on 520,000 X-ray images of chests.

520K+
X-ray images
seen during pre-training

86M
Internal values
that encode knowledge

Article | Open access | Published: 17 November 2025

EVA-X: a foundation model for general chest x-ray analysis with self-supervised learning

Jingfeng Yao, Xinggang Wang ✉, Yuehao Song, Huangxuan Zhao, Jun Ma, Yajie Chen, Wenyu Liu & Bo Wang ✉

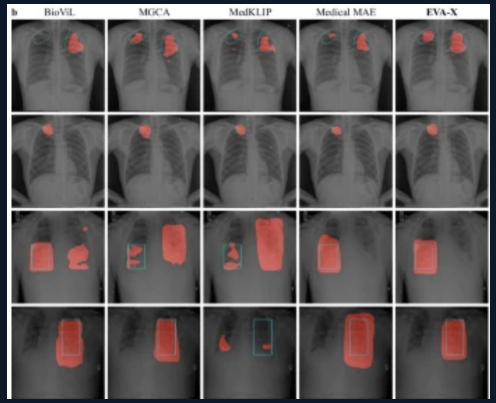


Image from paper

THRIVE Tennessee Health Research and Innovation using Verified AI Ecosystems

How can EVA-02 help with elbows?

A Vision Transformer pre-trained on 520,000 X-ray images of chests.
We can adjust this model by updating only 0.4% through LoRA fine-tuning. A low power adjustment to a big model.

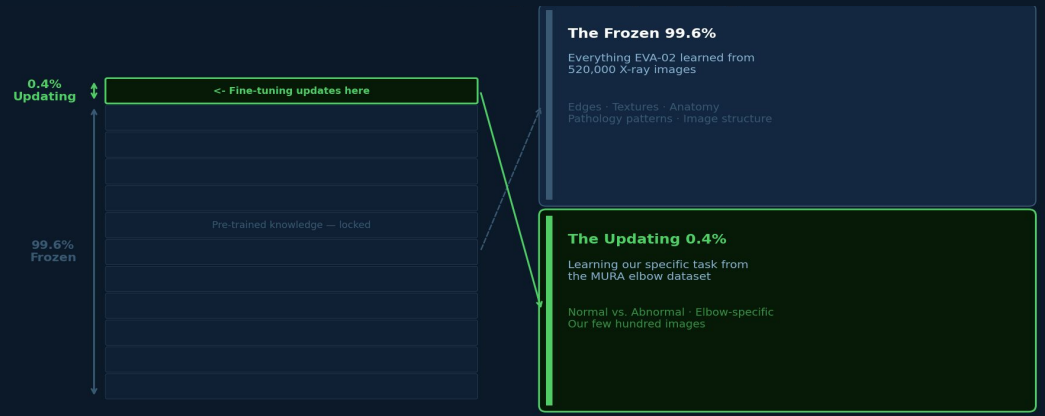
520K+
X-ray images
seen during pre-training

86M
Internal values
that encode knowledge

0.4%
Of the model updated
during fine-tuning for elbow xrays

Potential clickbait titles for this slide

1. Big Tech Doesn't Want You To Know This One Weird Fine-Tuning Trick
2. I Trained A Model With 0.1% Of The Parameters And You Won't Believe What Happened
3. GPUs HATE Him! Local Man Fine-Tunes LLM With This One Simple Trick
4. She Fine-Tuned A 70B Model On A Laptop. Researchers Are FURIOUS.
5. The Low-Rank Trick Big AI Labs Are Desperately Trying To Hide
6. I Replaced My Entire Dataset With 500 Examples And The Results SHOCKED Me
7. OpenAI Engineers HATE This Guy For Fine-Tuning On A Budget
8. What NVIDIA Doesn't Want You To Know About Training Your Own Model
9. Doctors, Lawyers, And AI Researchers All Agree: Stop Wasting VRAM
10. I Injected Tiny Matrices Into A Giant Model. My Landlord Is FURIOUS.



LIVE DEMO

Back to the Notebook

Take a look at the notebook. Here are the sections to focus on.

Workbook:
tinyurl.com/TnTech-LoRA-Xray

Step 4

Explore the data — look at real elbow X-rays

Step 5

Understand why fine-tuning — not building from scratch — is the right approach

Step 6

Load EVA-02 and see what 0.4% of parameters actually means

Step 8

Time one batch on CPU, then GPU — see the speedup live

Step 9

Start fine-tuning — watch loss drop and accuracy climb across 3 epochs

Step 10

Evaluate on the full validation set — confusion matrix and accuracy

Step 11

Run live inference — predictions on individual X-rays with confidence scores

Step 12

Modify the code to work for finger x-rays instead.

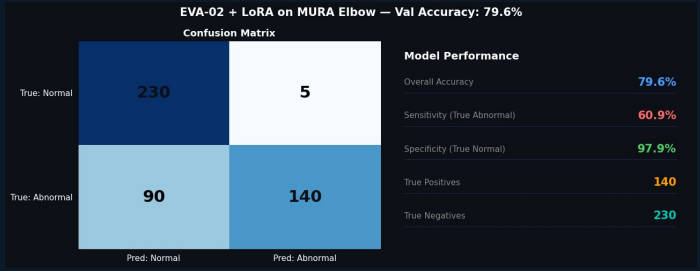
RESULTS

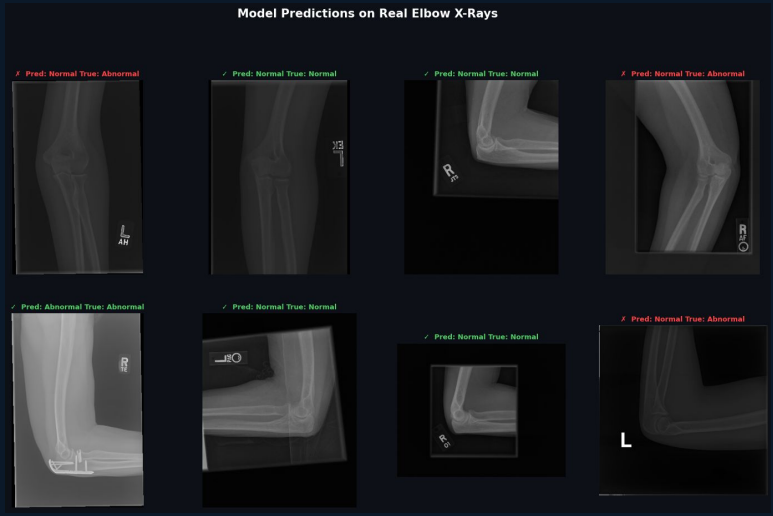
What the model achieved

~80%

Validation Accuracy

3 epochs · ~13 minutes on a free T4 GPU · EVA-02 + 0.4% fine-tuning





When Fine-Tuning Isn't Enough

Fine-tuning democratizes access to good models.
But hard problems still need serious resources — that is what NAIRR is built for.

Rare conditions

When you have hundreds of cases, not thousands — fine-tuning can can miss rare, under-represented patterns entirely

Gaps in available models

Fine-tuning works because there's already a model that knows something related to solving your problem (chest xray vs elbow).

If there's no good base model, fine-tuning doesn't work.

Lots of complicated data

If you have a large dataset that you hope to model, LoRA fine-tuning is not the right tool for the job. Full training is needed.

Use the right tool for the right job

LoRA fine-tuning a large model shines with limited data.

Full training works best when you have millions of datapoints.

The next presenter will show you how NAIRR compute makes these harder problems tractable.

YOUR TURN

Hands on session coming up! Fine-Tune on Finger X-Rays

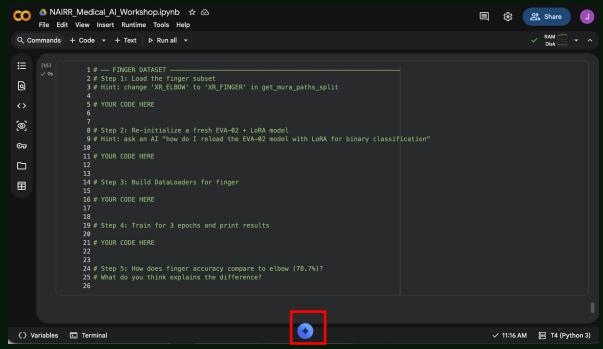
The model just learned elbows. Now teach it fingers — using AI assistance to write the code.

1 Change the dataset

Ask the LLM built into google colab to
add a new code section that trains the model for 3 epochs on 80% of the finger data in place of the elbow subset.

2 Evaluate

Ask the LLM built into google colab to
Evaluate the trained model on the held out finger data.



FRAMEWORK

Which Approach Fits Your Problem?

A reusable framework for any ML problem you bring back to your own research.

Lots of Data

Train a Large Model

NAIRR scale

Lots of labeled data + a complex task = the conditions where large models trained from scratch genuinely shine

Deep neural networks, large vision transformers, foundation models built for your domain

Weeks of training on GPU clusters — NAIRR resources make this accessible to researchers

Simple / Moderate Problem

Complex Problem

Little Data

Gradient Boosting / Ensembles

XGBoost territory

XGBoost, random forests, gradient boosting

Handles moderate complexity with limited data

Strong regularization limits overfitting

Interpretable enough for most public health contexts

Fine-Tune with PEFT

YOU ARE HERE

Pre-trained foundation + small targeted update

Only ~4% of parameters change — can't overfit

Already understands images and X-ray structure

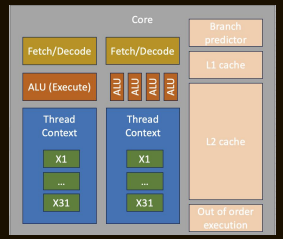
Train smarter, not harder ← Today's approach

THRIVE Tennessee Health Research and Innovation using Verified AI Ecosystems

Why GPUs Matter

CPU

Central Processing Unit



— A handful of very flexible cores

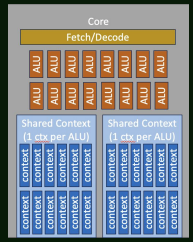
— Designed for sequential/multi-instruction tasks

— The generalist, good at everything (great at nothing)

— Slow at the parallel math neural nets need

GPU

Graphics Processing Unit



— Thousands of smaller cores

— Designed for multi-data computation

— The specialist, great at big data math (bad at all else)

— Neural training is matrix multiplication —
exactly what GPUs are built for

This is why NAIRR exists

NAIRR provides pools of GPU and CPU compute at national scale — purpose-built for exactly this kind of workload.

A free google T4 is often enough to get started. NAIRR is where you go when you are ready to scale up!

NAIRR Resources

What resources are available through NAIRR?

THRIVE Tennessee Health Research and Innovation using Verified AI Ecosystems

tinyurl.com/NAIRR-Resources

CPU

Best for parallel modeling (like fluid simulations)

- SDSC CPU
- Purdue Anvil CPU
- PSC Bridges
- SDSC Voyager
- DOE Argonne National Lab
- Cerebras AI Accelerator

GPU

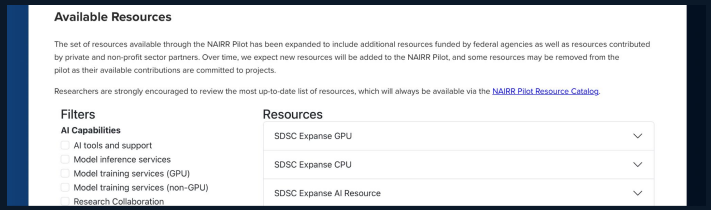
Best for machine learning

- SDSC GPU
- SDSC Expanse AI (BIG GPUs)
- Purdue Anvil GPU
- Purdue Anvil AI (BIG GPUs)
- PSC Bridges (BIG GPUs)
- NCSA Delta (BIG GPUs)
- Indiana Jetstream (GPUs on Demand – great for class)
- TAME ACEs
- PSC Neocortex 2 (designed for LLM research)
- Voltage Park
- TACC Vista
- NVIDIA DGX

Misc: Weights and biases (for tracking model learning), Samba cloud (model access), OpenAI (tokens), Hugging face (LLM research), Groq Model (LLM access), google cloud (similar to colab), Antrhopic (tokens), Databricks (data management), cloud bank (serving models),

tinyurl.com/NAIRR-Resources

How to request resources



1. Travel to the resources page
2. Evaluate which resource is most appropriate
3. Request resources via proposal submission

NAIRR Resources

THRIVE Tennessee Health Research and Innovation using Verified AI Ecosystems

NAIRR Submission Types

tinyurl.com/NAIRR-Resources

NAIRR Start-Up

Submission Begin Date: 2025-06-16

Important Submission Notes:

The Start-Up project call is a result of an effort to ensure that researchers are fully prepared to efficiently make use of NAIRR Pilot resources. A Start-Up proposal is designed to be a smaller, lighter-weight request, making it an ideal entry point for researchers who are new to NAIRR Pilot resources or unfamiliar with the process of requesting resource allocations. A Start-Up project provides researchers with the opportunity to gain hands-on experience, helping them build the foundation needed to prepare and submit a successful full NAIRR Pilot proposal.

NAIRR Pilot Start-Up projects are intended to be completed within three (3) months. Projects with longer-term timelines should explore the NAIRR Pilot Research Opportunity at https://nairrpilot.org/opportunities/allocations. NAIRR Pilot Research projects are awarded for twelve (12) months.

Start a New NAIRR Start-Up Submission

NAIRR Pilot

Submission Begin Date: 2026-03-16 **Submission End Date:** 2026-04-15

Important Submission Notes:

In the NAIRR Pilot, the US National Science Foundation (NSF), the US Department of Energy (DOE), and numerous private and non-profit sector partners are providing an opportunity for the research community to request access to a set of computing, model, platform and educational resources for projects related to advancing AI research and education. Proposals will be reviewed on an ongoing monthly cycle. Due to the volume of requests being received, requests submitted by the 15th of the month will be reviewed and have their outcome decided within three months. A new submission opportunity will open on the 16th of the month. Projects will be awarded for twelve (12) months duration.

Start a New NAIRR Pilot Submission

Three Things to Take Home

01

Start with what exists

Pre-trained models carry years of learning. For many public health imaging tasks, fine-tuning is faster, cheaper, and often better than building from scratch.

02

Hardware shapes what is possible. NAIRR provides hardware

A single GPU is often enough to test an idea.
NAIRR provides large scale compute for making scaling up accessible to researchers who need it for hard problems.

03

AI can help you write the code

You don't need to memorize every API or learn every programming language.
Knowing the right questions to ask — what model, what data, what task — is the real skill. The code follows.

How does an LLM work?

I just recommended using an LLM to help write the code

Many mean LLMs when they say “AI”. LLMs are a specific form of AI.
They are very practical and powerful, but understanding how they work helps reason about the mistakes they may make.

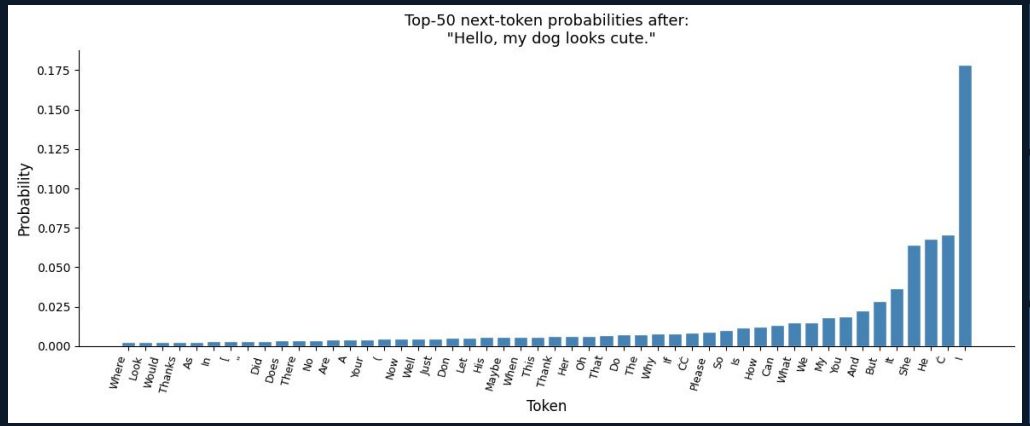
LLM

1. (deterministic) Estimate how probable every word is in context
2. (deterministic) Throw away all the least probable options
3. (random) Randomly choose from the remaining top – p words. This prevents repetitions and other pathological responses.

Takeaway: LLMs use a random process to respond and are therefore always going to make some mistakes!

tinyurl.com/Tntech-LLM-Example

Never blindly trust an LLM



LLMs are deterministic

Given an input, the output probabilities won't change

LLMs are randomly sampled

Options are randomly sampled to improve response quality.

Never trust a random process!

```
Strategy : Greedy (deterministic)
Output   : Hello, my dog looks cute. I'm not sure if she's a dog or not. I'm not sure if she's a dog or not.


Strategy : Random sampling
Output   : Hello, my dog looks cute. Every time they saw it happy or scared, was it shy or unfamiliar? I feel I must go take a sip from my

Strategy : Top-p (p=0.4)
Output   : Hello, my dog looks cute. She's very shy and quiet. She's a little like me, but she's very friendly. I love her, but
```

MNIST with a Foundation Model

https://tinyurl.com/TnTech-MNIST-LoRA