

• OPPORTUNITY OPEN SOURCE CONFERENCE 4.0 • IIIT ALLAHABAD

Is Git for Research?

Git's Linearity vs. The Researcher's Need for Diversity

By **Bibekananda Hati**

AI Researcher and Founder of **ExperQuick Research Infra Pvt. Ltd.**





Linus Torvalds

CREATOR OF GIT

*I'm an egotistical bastard, and I name all my projects after myself. First **Linux**, now **git**.*

OFFICIAL README DEFINITION:

"Git - the stupid content tracker"

What does it mean to be a **content tracker**?
It records a strict, linear progression of changes.



And hence, Git is a

Version Control System

How does this fit **Software Development**?

We branch out to work in parallel, but always **converge** back to a single linear history.



But in Research...

Research has a messy flow. It's a massive, diverging search space.



The Mess of Experimentation

AT THE CORE OF RESEARCH

TYPICAL WORKSPACE REALITY

- 0_jan_feb_output
- 0_small
- 10_tf_idfid
- 10_tfidf_NB
- 7_tfidf_svm
- 8_tfidf_1500_LG
- 4_tfidf
- 6_tfidf_10000
- 3_tfidf_output_1

ONLY LOGS ARE TRACKED

FOLDER NAME	ACCURACY	LOSS
0_small	0.829478	0.3566
0_jan_feb_output	0.839356	0.5895
0_jan_feb_output	0.805344	0.2903
0_jan_feb_output	0.556634	0.3467

Experiments are hidden in folders

Each folder has a README of high-level architectural info.

PROBLEMS

- ✗ Only manual, high-level config stored per run
- ✗ In-depth info needs painful file-to-file traversal
- ✗ Code repeated across shared-component experiments
- ✗ Data generated — but never knowledge

Is Git suitable for experiment exploration?

REALITY CHECK



```
git log --graph --all
```

EXPERIMENT TRACKING VIA COMMITS?

```
commit -m "try lr 0.001"
```

```
commit -m "try another architecture"
```

```
commit -m "change dataset split"
```

WHICH BRANCH HAD THE GOOD MODEL?

```
commit -m "final experiment REAL"
```

Let's argue

GIT GIVES HISTORY OF CHANGES, NOT THE KNOWLEDGE OF IMPACTS

- ? Which component variant contributes most to overall performance?
- ? What explicitly changed in the data or architecture between these two runs?
- ? Have I already run this exact configuration before?
- ? Can I easily find the branch where this specific combination actually worked?

Git is for software.

RESEARCH NEEDS SOMETHING ELSE.

Meet PyLabFlow

THE MISSING LAYER OF RESEARCH SPACE



OPEN-SOURCE

PYTHON

EXPERIMENT FRAMEWORK

Structure

Organize experiments as components and configurations.

Remember

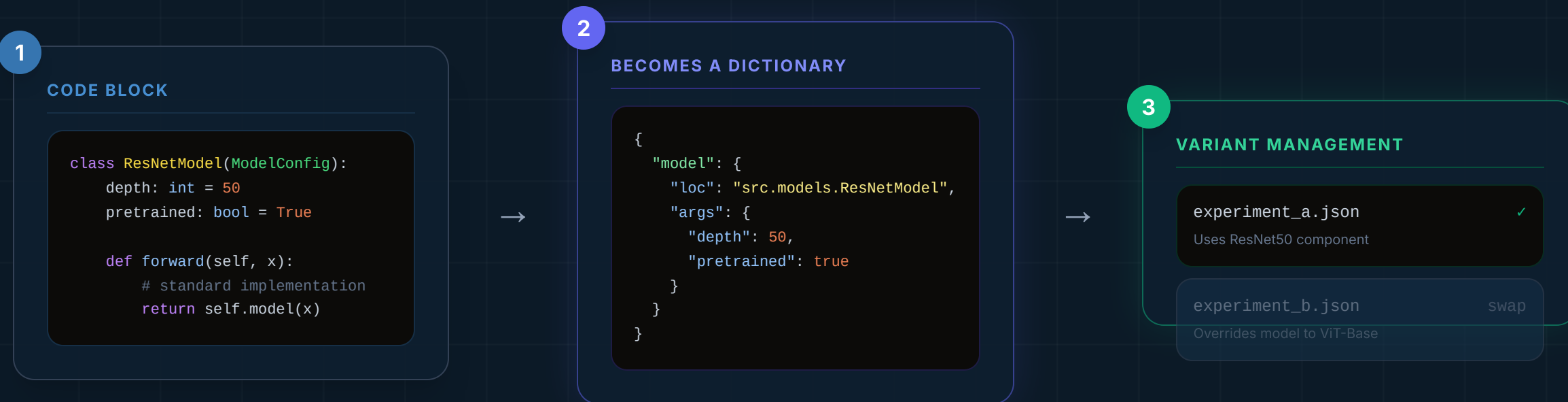
Keep experiment states and runs perfectly structured.

Query

Instantly find and reproduce what you already tried.

Component-Centric Architecture

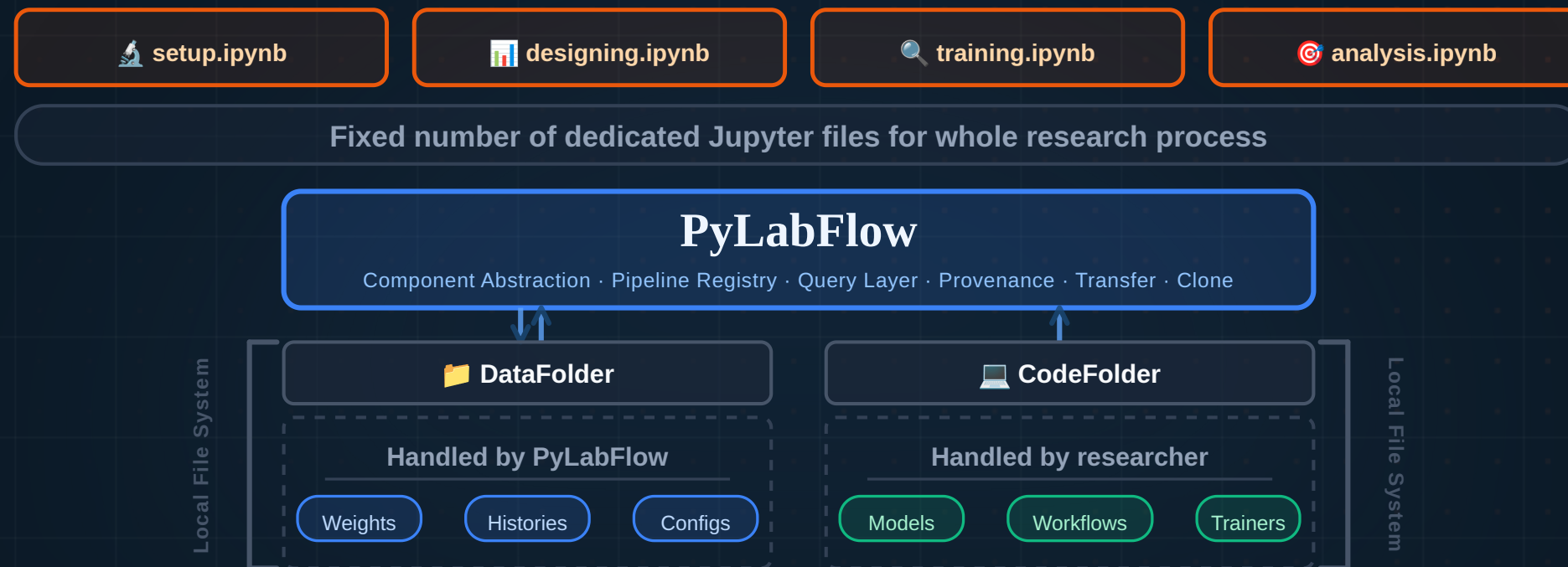
IT DEFINES A WAY TO WRITE CODE BLOCKS



Code is **static**. Experiments are just **data structures**.

How PyLabFlow Interacts

LOCAL INTERNAL MANAGEMENT & QUERYABLE INTERFACE



The Structural Advantage

WHAT HAPPENS WHEN EXPERIMENTS BECOME QUERYABLE OBJECTS?



Learn from every attempt

Every experiment — including failed ones — is a permanent, hashed record. Dead ends become permanent evidence, not lost time.



Guaranteed Reproducibility

The experiment definition *is* the reproduction recipe. Exact reproduction becomes structurally guaranteed — not manually maintained.



Transferable Assets

Knowledge is institutionalised. A full decision trail can be shared across teams or licensed as a monetisable asset.

The Journey

HOW IT STARTED & WHERE IT'S GOING

ORIGIN STORY



PyTorchLabFlow

The original prototype, built specifically for PyTorch workflows.



PyLabFlow

Generalized open-source experiment management framework for Python.



ExperQuick Research Infra Pvt. Ltd.

The company founded to scale and support this vision for research teams.



ExperLab IRE BETA

The first-ever Integrated Research Environment (IRE) by ExperQuick.

RESOURCES



Original Prototype

github.com/BBEK-Anand/PyTorchLabFlow



Published Research

doi.org/10.53555/jab.v11si6.1940



PyLabFlow Framework

github.com/ExperQuick/PyLabFlow



ExperQuick

experquick.org

Thank You.

Time for the **Live Demo**

UPCOMING WORKSHOP

11:35 – 13:05

Hands-On PyLabFlow: Building Reproducible ML & Deep Learning Pipelines from Scratch

Want to build this yourself? I'm running a hands-on workshop right after this talk where we'll set this up from scratch. Catch me after the demo if you want to jump in!