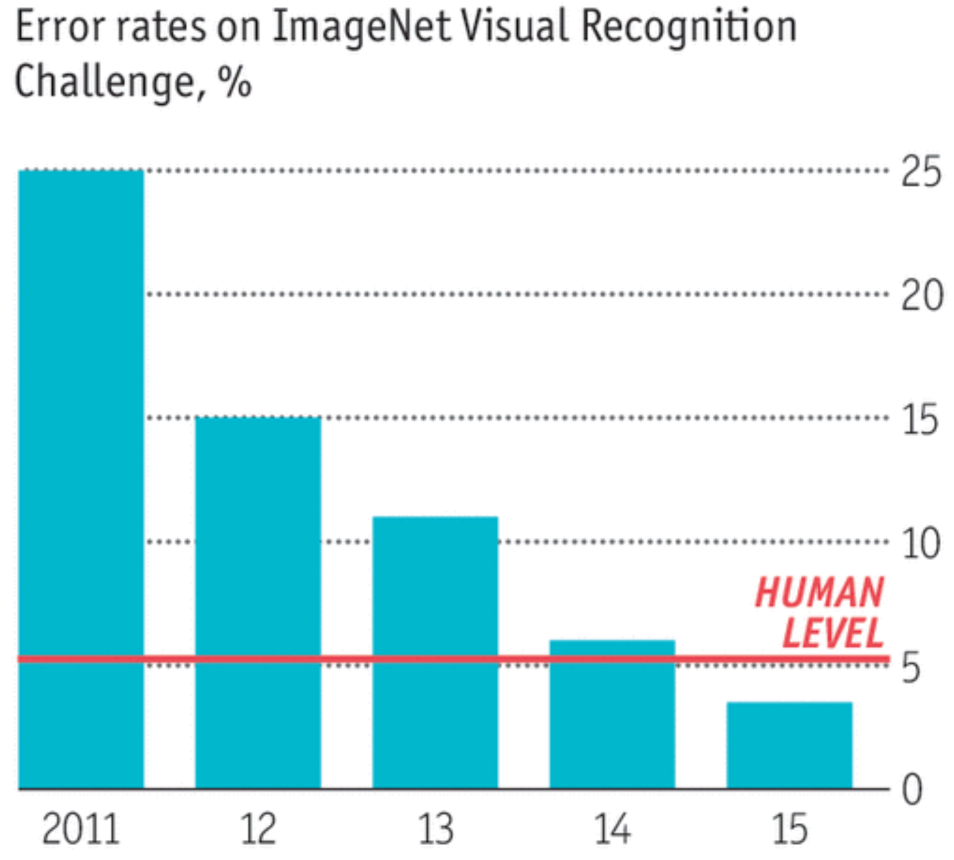
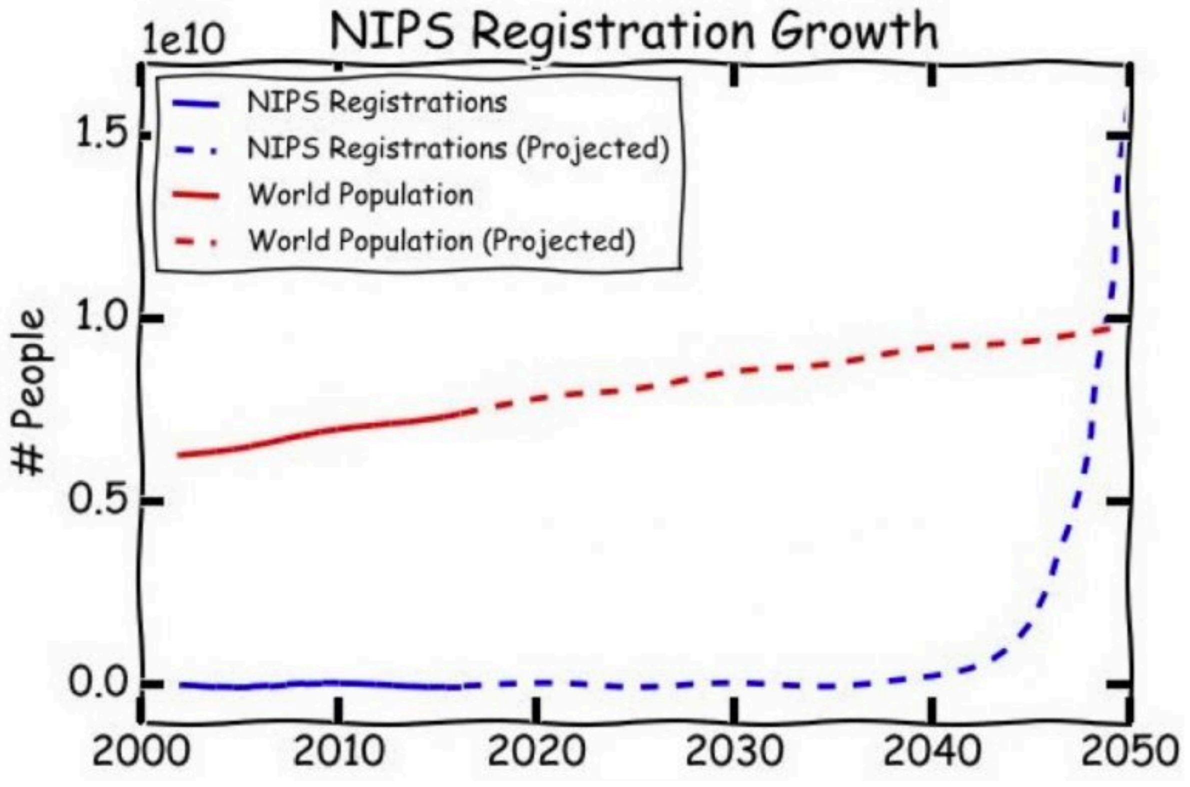
Determined AI

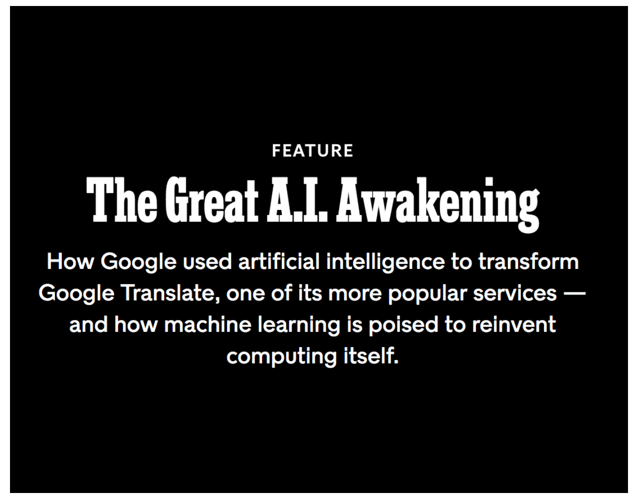
Golden Age for AI, Dark Ages for AI Infrastructure

Neil Conway
Co-Founder and CTO, Determined AI
April 28, 2019

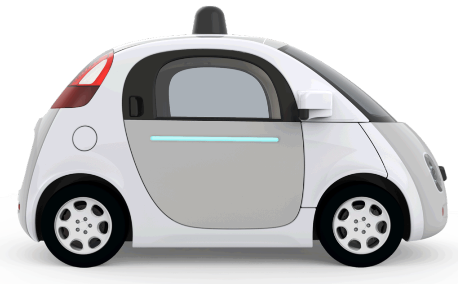
The Golden Age of AI






FEATURE
The Great A.I. Awakening
How Google used artificial intelligence to transform Google Translate, one of its more popular services — and how machine learning is poised to reinvent computing itself.


Tech Giants Are Paying Huge Salaries for Scarce A.I. Talent
Nearly all big tech companies have an artificial intelligence project, and they are willing to pay experts millions of dollars to help get it done.



Technology

Coming This Fall to Carnegie Mellon: America's First AI Degree

TECHNOLOGY

Stanford's Top Major Is Now Computer Science

Keras4Kindergartners

AI ready for widespread adoption?

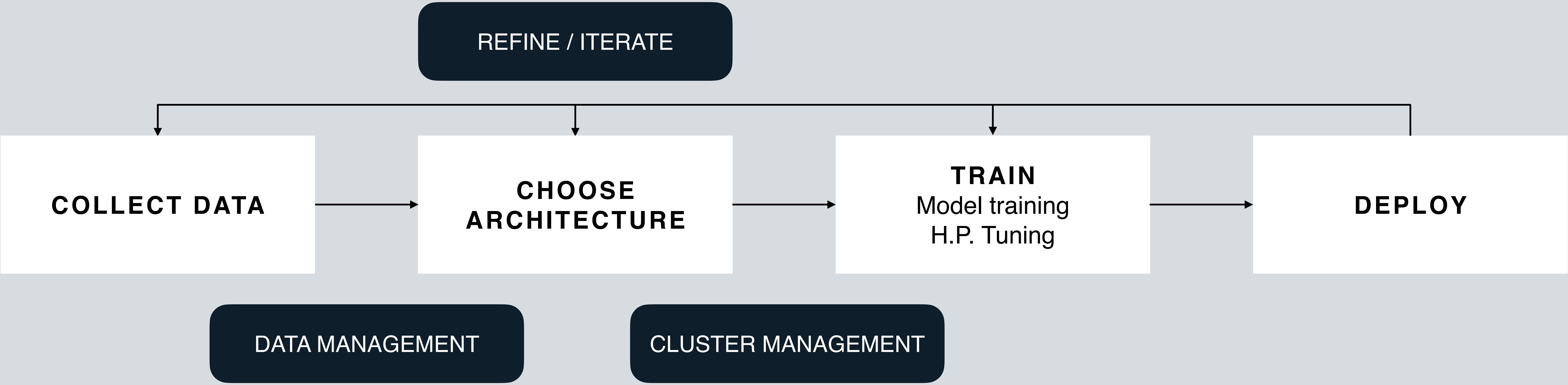
The Dark Age of AI Infrastructure

Forcing users to wait for **days** to recover from faults.

Reproducing existing models is **death by a thousand cuts:** data ordering, software versions, hyperparmeters, random seeds, model weights.

Hand-implemented, **impossibly slow** methods to find good models.

Deep Learning Today (For Everyone Else)



Deep Learning Today (For Everyone Else)

Existing Tools (e.g., TensorFlow):

- Mostly focused on 1 researcher, training 1 model, on 1 GPU

Limited Support For:

- Teams of researchers, clusters of GPUs, many models
- Deployment, ops, and collaboration
- Data management or cluster management

Most existing tools fall
into one of two buckets:

Too Generic
(e.g., Spark, Sun Grid Engine)

Technical Point Solutions
(e.g., TensorFlow, Horovod)

We need AI
Infrastructure that is:

Specialized for Deep Learning

DL is both <u>different</u> and <u>extremely important</u>

Holistic & Integrated

Orders of magnitude wins in performance and usability!

The Dark Age of AI Infrastructure

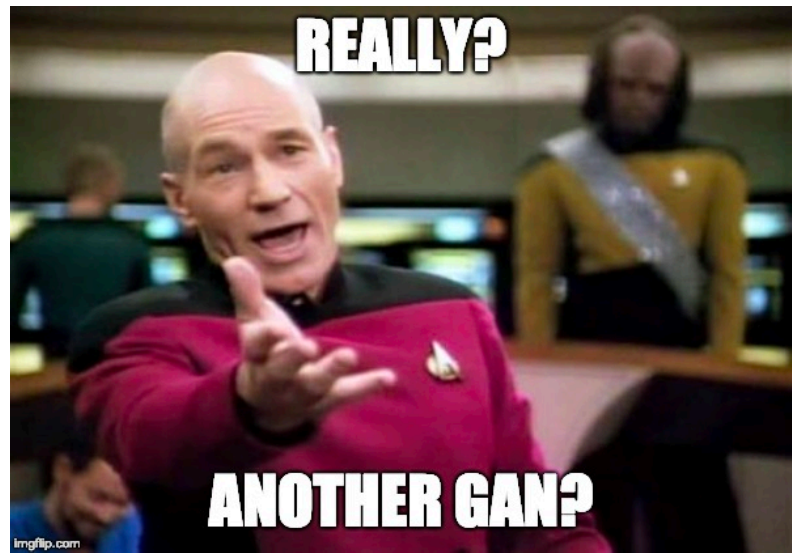
Forcing users to wait for **days** to recover from faults.

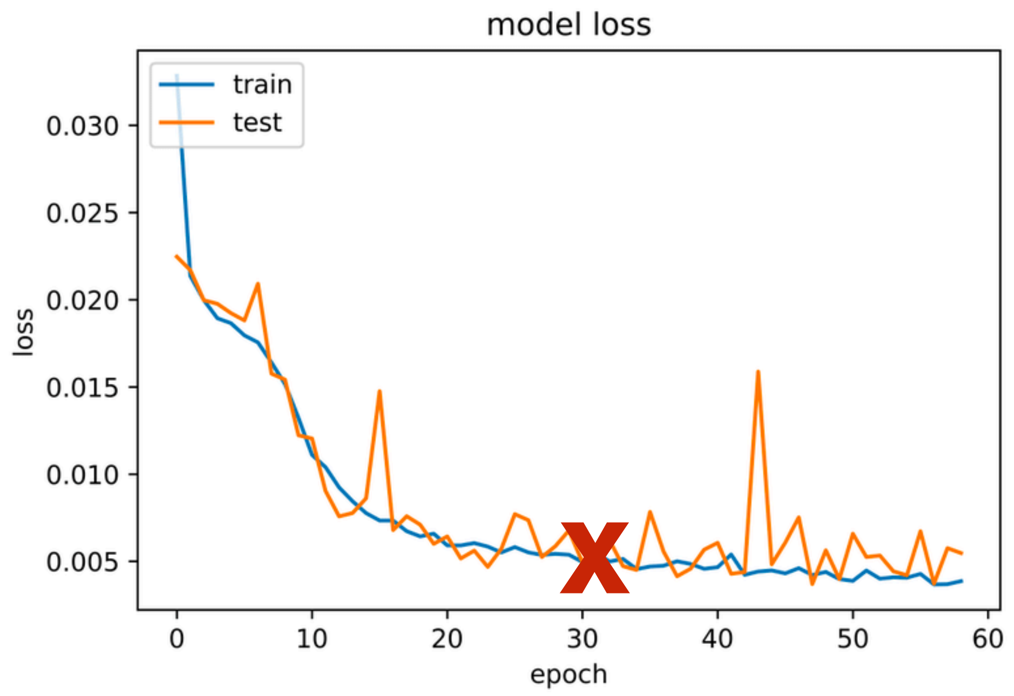
Reproducing existing models is **death by a thousand cuts:** data ordering, software versions, hyperparameters, random seeds, model weights.

Hand-implemented, **impossibly slow** methods to find good models.

Dave's got a problem.

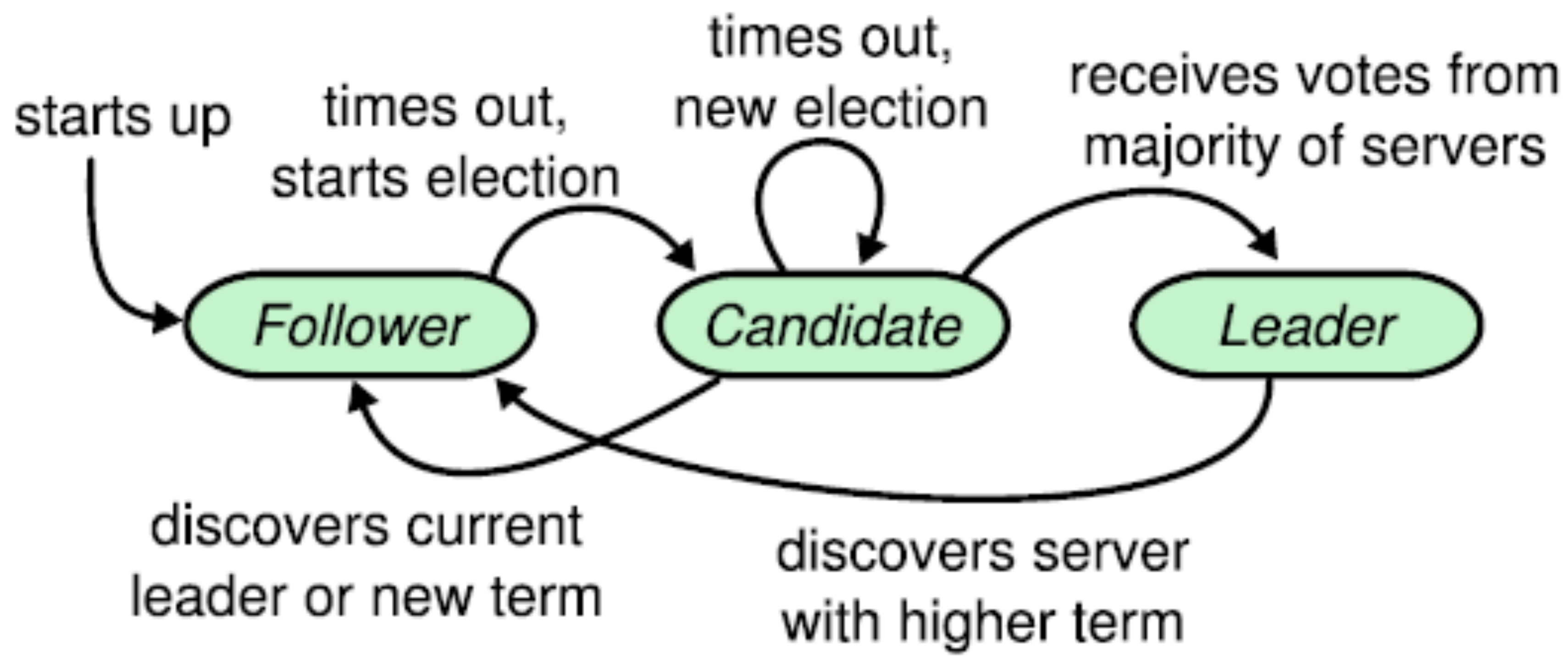
- Dave's a super smart DL engineer.
- He's got a brilliant model for style transfer that automatically makes every picture a dank meme.
- It takes two days for his model to converge on a couple of DGX-1s.
- Every time his model crashes he loses (on average) a day of work and 400 GPU-hours of compute time.
- This makes Dave sad.





Dave’s got a solution!

- Dave wants to make sure he doesn’t lose work.
- In general, this is a “hard problem.”
- In Deep Learning - this isn’t so bad.
  Enter `tf.saved_model.simple_save`.
- So, Dave instruments his code, and the next time it crashes he loads his model using `tf.saved_model.loader.load` and keeps on training.



Only, he doesn't.

- TensorFlow only saves:
  - Weights, optimizer state.
- Dave also needs
  - Input read position, random seeds, model definition, dependencies.
- Eventually, **Dave writes a pile of code to save all this stuff.**

And Dave's life <u>still sucks</u>

- Learns the hard way that checkpoints are really big and runs out of disk space.
- Teaches himself PagerDuty so that he can find out when his models crash and ssh back into the cluster to kick the models off.
- Loses his place in the queue.
- **Dave writes a pile of cron jobs to make sure his work is being done.**

What if Dave had <u>holistic</u> but <u>specialized</u> AI infrastructure?

- Fault tolerance would be taken care of (the right way) out of the box.
- The **infrastructure** would automatically take checkpoints.
- The **infrastructure** would monitor and retry failed jobs from latest checkpoint automatically.
- The **infrastructure** would manage its own checkpoint storage according to sane rules (“keep checkpoints with the best n validation errors”).
- The **infrastructure** could leverage checkpoints in other, surprising ways: to enable reproducibility, as a unit of scheduling/job migration, and to enable distributed training.
- All of this would be **transparent** to Dave.

The Dark Age of AI Infrastructure

Forcing users to wait for **days** to recover from faults.

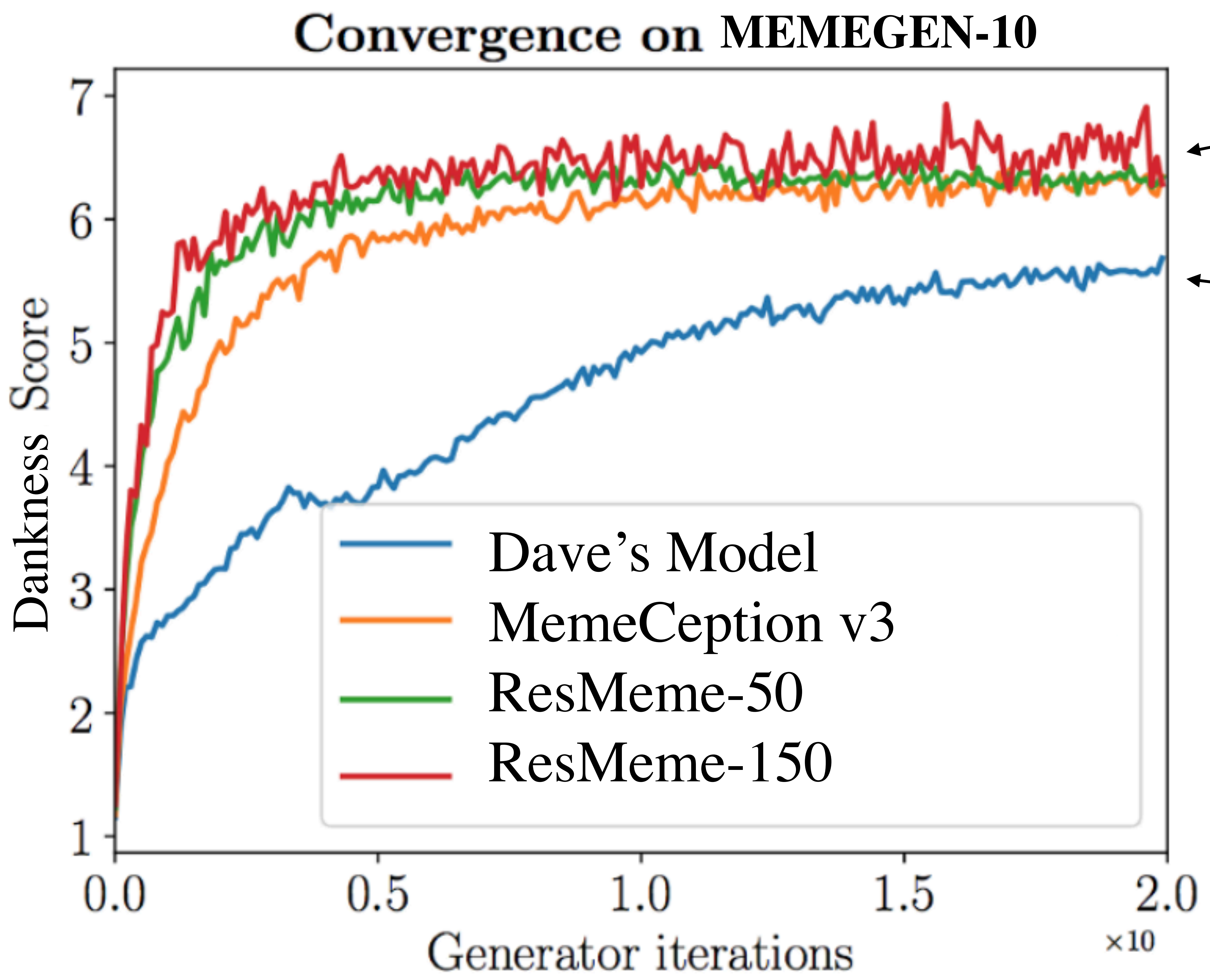
Reproducing existing models is **death by a thousand cuts:** data ordering, software versions, hyperparameters, random seeds, model weights.

Hand-implemented, **impossibly slow** methods to find good models.

Dave trains his model

```
freshpond:DLRox sparks$ python train_script.py --learning_rate=0.1 --dropout=0.5 > logs/result-0.1-0.5.log
freshpond:DLRox sparks$ ls logs
result-0.1-0.5.log
```

Dave’s got a quality problem



State of the art meme generation.

Dave’s memes

Dave’s memes aren’t dank enough.

So Dave starts tuning hyperparameters.

Dave Discovers Grid Search

```
freshpond:DLRox sparks$ for lr in -0.001 -0.01 -0.1 1.0 10.0 100.0 1000.0
> do
>   for dropout in 0.0 0.1 0.2 0.3 0.4 0.5
>   do
>     python train_script.py --learning_rate=$lr --dropout=$dropout > logs/results-$lr-$dropout.log
>   done
> done
```

Nested `for` loops FTW

Dave Discovers Grid Search

```
freshpond:DLRox sparks$ ls logs
result-0.1-0.5.log        results--0.01-0.4.log   results-1.0-0.3.log     results-100.0-0.2.log
results--0.001-.log       results--0.01-0.5.log   results-1.0-0.4.log     results-100.0-0.3.log
results--0.001-0.0.log    results--0.1-.log       results-1.0-0.5.log     results-100.0-0.4.log
results--0.001-0.1.log    results--0.1-0.0.log    results-10.0-.log       results-100.0-0.5.log
results--0.001-0.2.log    results--0.1-0.1.log    results-10.0-0.0.log    results-1000.0-.log
results--0.001-0.3.log    results--0.1-0.2.log    results-10.0-0.1.log    results-1000.0-0.0.log
results--0.001-0.4.log    results--0.1-0.3.log    results-10.0-0.2.log    results-1000.0-0.1.log
results--0.001-0.5.log    results--0.1-0.4.log    results-10.0-0.3.log    results-1000.0-0.2.log
results--0.01-.log        results--0.1-0.5.log    results-10.0-0.4.log    results-1000.0-0.3.log
results--0.01-0.0.log     results-1.0-.log        results-10.0-0.5.log    results-1000.0-0.4.log
results--0.01-0.1.log     results-1.0-0.0.log     results-100.0-.log      results-1000.0-0.5.log
results--0.01-0.2.log     results-1.0-0.1.log     results-100.0-0.0.log
results--0.01-0.3.log     results-1.0-0.2.log     results-100.0-0.1.log
```

The results are in.. (kinda)

Dave Discovers Grid Searc

That's slow, let's u RCE_MANAGER

```
freshpond:DLRox sparks$ for lr in -0.0000 1.0 10.0 100.0 1000.0 10000.0 100000.0 ;
do   for dropout in 0.0 0.05 0.1 0.15 0.2 ;   do     qsub python train_script.py --l
earning_rate=$lr --dropout=$dropout > logs ;   done; done
```

F !

```
freshpond:DLRox sparks$ ls logs
result-0.1-0.5.log
results--0.00001-0.0.log
results--0.00001-0.05.log
results--0.00001-0.1.log
results--0.00001-0.15.log
results--0.00001-0.2.log
results--0.00001-0.25.log
results--0.00001-0.3.log
results--0.00001-0.35.log
results--0.00001-0.4.log
results--0.00001-0.45.log
results--0.00001-0.5.log
results--0.0001-0.0.log
results--0.0001-0.05.log
results--0.0001-0.1.log
results--0.0001-0.15.log
results--0.0001-0.2.log
results--0.0001-0.25.log
results--0.0001-0.3.log
results--0.0001-0.35.log
results--0.0001-0.4.log
results--0.0001-0.45.log
results--0.0001-0.5.log
results--0.001-.log
results--0.001-0.0.log
results--0.001-0.05.log
results--0.001-0.1.log
results--0.001-0.15.log
results--0.001-0.2.log
results--0.001-0.25.log
results--0.001-0.3.log
results--0.001-0.35.log
results--0.001-0.4.log
results--0.001-0.45.log
results--0.001-0.5.log
results--0.01-.log
results--0.01-0.0.log
results--0.01-0.05.log
results--0.01-0.1.log
results--0.01-0.15.log
results--0.01-0.2.log
results--0.01-0.25.log
results--0.01-0.3.log
results--0.01-0.35.log
results--0.01-0.4.log
results--0.01-0.45.log
results--0.01-0.5.log
results--0.1-.log
results--0.1-0.0.log
results--0.1-0.05.log
results--0.1-0.1.log
results--0.1-0.15.log
results--0.1-0.2.log
results--0.1-0.25.log
results--0.1-0.3.log
results--0.1-0.35.log
results--0.1-0.4.log
results--0.1-0.45.log
results--0.1-0.5.log
results-1.0-.log
results-1.0-0.0.log
results-1.0-0.05.log
results-1.0-0.1.log
results-1.0-0.15.log
results-1.0-0.2.log
results-1.0-0.25.log
results-1.0-0.3.log
results-1.0-0.35.log
results-1.0-0.4.log
results-1.0-0.45.log
results-1.0-0.5.log
results-10.0-.log
results-10.0-0.0.log
results-10.0-0.05.log
results-10.0-0.1.log
results-10.0-0.15.log
results-10.0-0.2.log
results-10.0-0.25.log
results-10.0-0.3.log
results-10.0-0.35.log
results-10.0-0.4.log
results-10.0-0.45.log
results-10.0-0.5.log
results-100.0-0.0.log
results-100.0-0.05.log
results-100.0-0.1.log
results-100.0-0.15.log
results-100.0-0.2.log
results-100.0-0.25.log
results-100.0-0.3.log
results-100.0-0.35.log
results-100.0-0.4.log
results-100.0-0.45.log
results-100.0-0.5.log
results-1000.0-.log
results-1000.0-0.0.log
results-1000.0-0.05.log
results-1000.0-0.1.log
results-1000.0-0.15.log
results-1000.0-0.2.log
results-1000.0-0.25.log
results-1000.0-0.3.log
results-1000.0-0.35.log
results-1000.0-0.4.log
results-1000.0-0.45.log
results-1000.0-0.5.log
results-10000.0-0.0.log
results-10000.0-0.05.log
results-10000.0-0.1.log
results-10000.0-0.15.log
results-10000.0-0.2.log
results-10000.0-0.25.log
results-10000.0-0.3.log
results-10000.0-0.35.log
results-10000.0-0.4.log
results-10000.0-0.45.log
results-10000.0-0.5.log
results-100000.0-0.0.log
results-100000.0-0.05.log
results-100000.0-0.1.log
results-100000.0-0.15.log
results-100000.0-0.2.log
results-100000.0-0.25.log
results-100000.0-0.3.log
results-100000.0-0.35.log
results-100000.0-0.4.log
results-100000.0-0.45.log
results-100000.0-0.5.log
results2--0.00001-0.0.log
results2--0.00001-0.05.log
results2--0.00001-0.1.log
results2--0.00001-0.15.log
results2--0.00001-0.2.log
results2--0.00001-0.25.log
results2--0.00001-0.3.log
results2--0.00001-0.35.log
results2--0.00001-0.4.log
results2--0.00001-0.45.log
results2--0.00001-0.5.log
results2--0.0001-0.0.log
results2--0.0001-0.05.log
results2--0.0001-0.1.log
results2--0.0001-0.15.log
results2--0.0001-0.2.log
results2--0.0001-0.25.log
results2--0.0001-0.3.log
results2--0.0001-0.35.log
results2--0.0001-0.4.log
results2--0.0001-0.45.log
results2--0.0001-0.5.log
results2--0.001-0.0.log
results2--0.001-0.05.log
results2--0.001-0.1.log
results2--0.001-0.15.log
results2--0.001-0.2.log
results2--0.001-0.25.log
results2--0.001-0.3.log
results2--0.001-0.35.log
results2--0.001-0.4.log
results2--0.001-0.45.log
results2--0.001-0.5.log
results2--0.01-0.0.log
results2--0.01-0.05.log
results2--0.01-0.1.log
results2--0.01-0.15.log
results2--0.01-0.2.log
results2--0.01-0.3.log
results2--0.01-0.35.log
results2--0.01-0.4.log
results2--0.01-0.45.log
results2--0.01-0.5.log
results2--0.1-0.0.log
results2--0.1-0.05.log
results2--0.1-0.1.log
results2--0.1-0.15.log
results2--0.1-0.2.log
results2--0.1-0.25.log
results2--0.1-0.3.log
results2--0.1-0.35.log
results2--0.1-0.4.log
results2--0.1-0.45.log
results2--0.1-0.5.log
results2-1.0-0.0.log
results2-1.0-0.05.log
results2-1.0-0.1.log
results2-1.0-0.15.log
results2-1.0-0.2.log
results2-1.0-0.25.log
results2-1.0-0.3.log
results2-1.0-0.35.log
results2-1.0-0.4.log
results2-1.0-0.45.log
results2-1.0-0.5.log
results2-10.0-0.0.log
results2-10.0-0.05.log
results2-10.0-0.1.log
results2-10.0-0.15.log
results2-10.0-0.2.log
results2-10.0-0.25.log
results2-10.0-0.3.log
results2-10.0-0.35.log
results2-10.0-0.4.log
results2-10.0-0.45.log
results2-10.0-0.5.log
results2-100.0-0.0.log
results2-100.0-0.05.log
results2-100.0-0.1.log
results2-100.0-0.15.log
results2-100.0-0.2.log
results2-100.0-0.25.log
results2-100.0-0.3.log
results2-100.0-0.35.log
results2-100.0-0.4.log
results2-100.0-0.45.log
results2-100.0-0.5.log
results2-1000.0-0.0.log
results2-1000.0-0.05.log
results2-1000.0-0.1.log
results2-1000.0-0.15.log
results2-1000.0-0.2.log
results2-1000.0-0.25.log
results2-1000.0-0.3.log
results2-1000.0-0.35.log
results2-1000.0-0.4.log
results2-1000.0-0.45.log
results2-1000.0-0.5.log
results2-10000.0-0.0.log
results2-10000.0-0.05.log
results2-10000.0-0.1.log
results2-10000.0-0.15.log
results2-10000.0-0.2.log
results2-10000.0-0.25.log
results2-10000.0-0.3.log
results2-10000.0-0.35.log
results2-10000.0-0.4.log
results2-10000.0-0.45.log
results2-10000.0-0.5.log
results2-100000.0-0.0.log
results2-100000.0-0.05.log
results2-100000.0-0.1.log
results2-100000.0-0.15.log
results2-100000.0-0.2.log
results2-100000.0-0.25.log
results2-100000.0-0.3.log
results2-100000.0-0.35.log
results2-100000.0-0.4.log
results2-100000.0-0.45.log
results2-100000.0-0.5.log
```

Now Dave Has Two Problems

(1) Poor Infrastructure Support

- No fault tolerance
- No experiment tracking
- No metadata storage
- Missed optimization opportunities

(2) Dumb Search Strategy

Dave is wasting > 99% of his time!

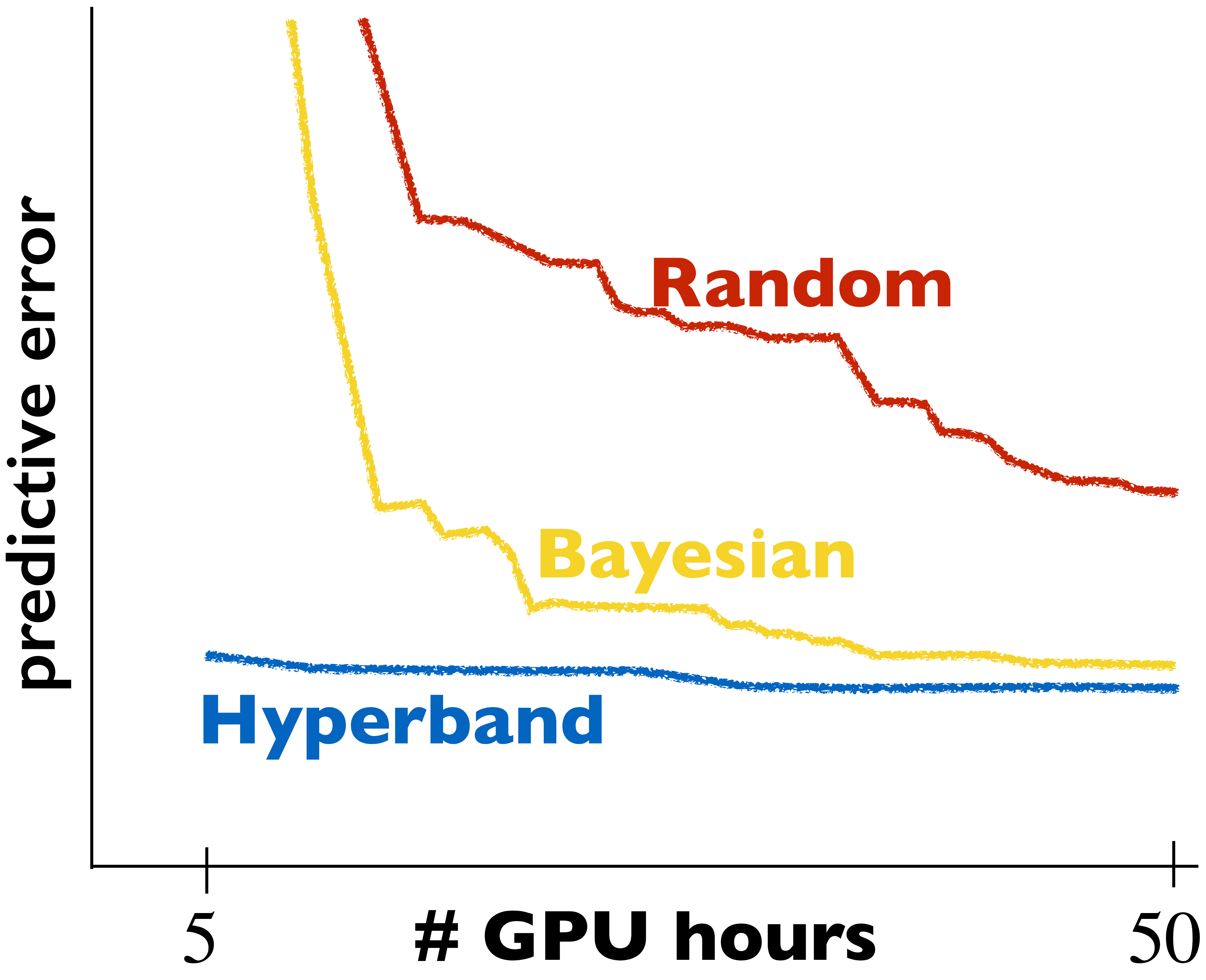
Hyperparameter Optimization

4 layer CNN

8 Hyperparameters

Image recognition

CIFAR10



Hyperband: Massively Parallel HPO [ICLR 2017]

Intuition:

- Examine **many** hyperparameter configurations at once
- **Prune** the configurations that are doing poorly (“early stopping”)
- **Adaptively** allocate more training resources to the configurations that are doing well

Speedups

>50x over Random

10x over Bayesian

✔ **Lower final error**

✔ **Lower variance**

Unfortunately, Dave can’t Hyperband

Dave’s Infrastructure Dilemma

Cluster Manager:

Doesn’t understand the semantics of deep learning workloads

DL Frameworks:

Built to train a **single model** for a **single user** on a **single machine**

What’s missing is **holistic** but **specialized** infrastructure to provide the glue between these two

The Dark Age of AI Infrastructure

Forcing users to wait for **days** to recover from faults.

Reproducing existing models is **death by a thousand cuts:** data ordering, software versions, hyperparameters, random seeds, model weights.

Hand-implemented, **impossibly slow** methods to find good models.

Dave is taking over for Leslie

Dave is assigned to a new project. A former colleague, Leslie, trained a production model six months ago. Dave wants to explore new modeling techniques to see if he can improve the model's performance.

He re-runs Leslie's training script but get **drastically higher error**

Time to debug…

What does Dave discover?

Training data: New samples recently added to Leslie's directory

Hyperparameters: Leslie didn't use default values, and instead specified batch size and learning rate at runtime

	Validation Error	Difference from Baseline
Baseline	30.3%	0.0%
Test1 (w/o fixes)	52.8%	22.5%
Test2 (includes fixes)	37.3%	**7%**

Ugh…Debug…

Randomness is an intrinsic part of training

- e.g., weight initialization, shuffling and augmentation of datasets, noisy hidden layers (e.g. dropout)

Fix random seeds!

- There are lots of them!
- ML framework dependent
- Must be recorded for reuse

Ugh…Debug…

Variation across specialized software

- Within versions and across ML frameworks (TF, Keras, PyTorch)
- Underlying libraries (NumPy, cuDNN, CUDA, MKL)

Leverage the power of containerization!

Requires non-trivial engineering infrastructure

Ugh…Debug…

	Validation Error	Difference from Baseline
Baseline	30.3%	0.0%
Test1: No changes	52.8%	22.5%
Test2: Fix dataset + hyperparameters	37.3%	7.0%
Test3: Fix for libraries + random seeds	29.2%	**-1.1%**

UGH!!!

Inherent System/Hardware Level Randomness

- non-deterministic GPU operations
- CPU multi-threading

Fixing this, at last, we have perfect reproducibility

But it requires **CPU-only** training with multi-threading disabled…**SLOW**!

What would an <u>holistic</u> but <u>specialized</u> DL reproducibility solution include?

Feature	Purpose
Version control for model definitions	Track changes in model architecture, optimization algorithm, data preprocessing pipeline
Metadata capture and storage	Record training + validation metrics, training logs, model hyperparameters
Dependency management	Ensure ML framework and all dependencies are consistent between runs
Experiment seed management	Generate the same pseudo-random values every run
Hardware resource flexibility	Allow users to disable multi-threading and GPU usage, if desired

Conclusion

1. Dave's life sucks because today's DL Infrastructure tools are bad.

2. Existing tools: overly **generic** or **narrow technical point solutions**

3. What we need: tools that are **specialized** for DL and support DL workflows in a **holistic**, end-to-end way.

Determined AI

Our platform gives AI teams the tools they need to train and deploy DL models dramatically more quickly.

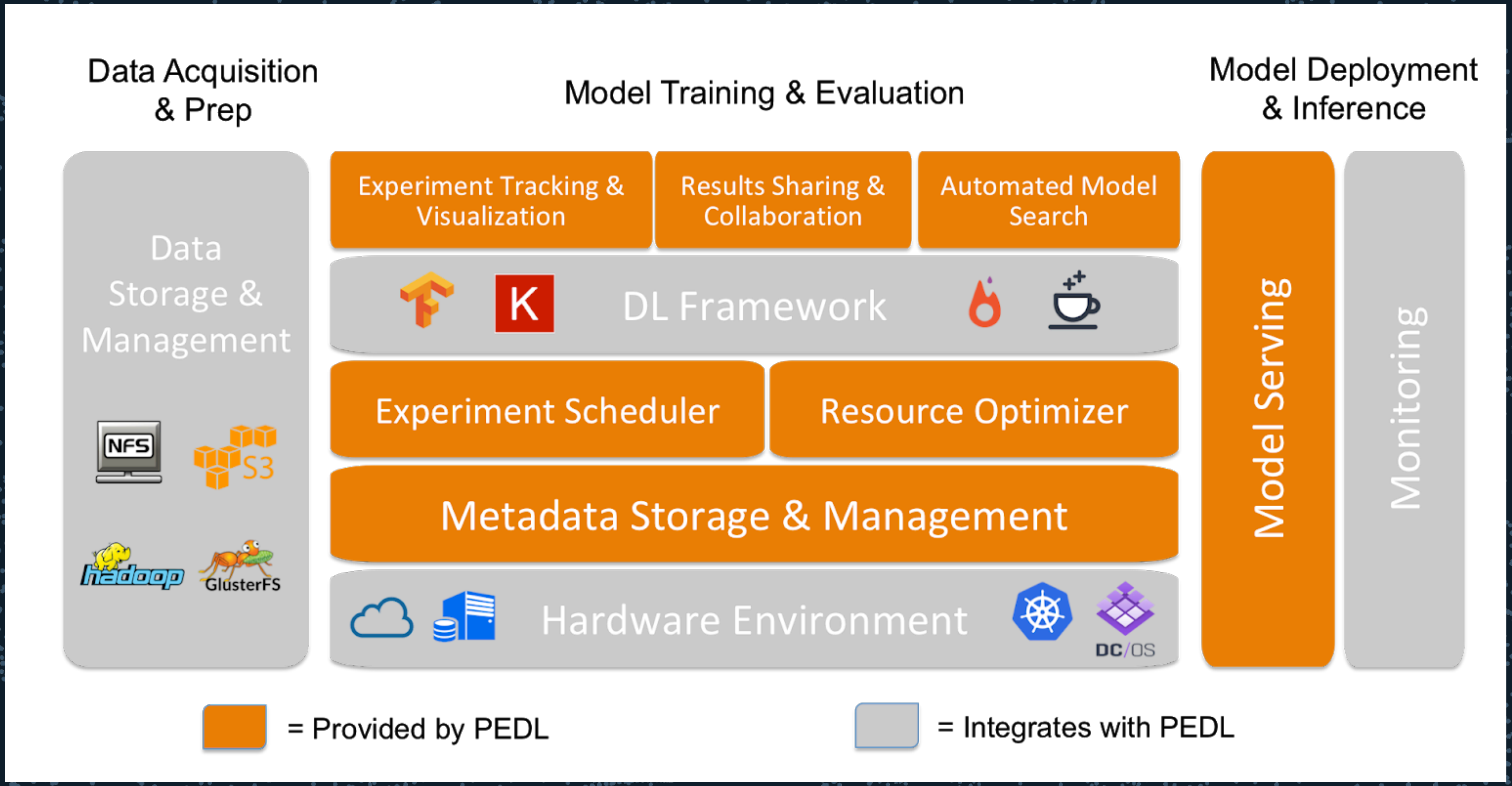
Best-in-class AutoML capabilities

Automated GPU resource optimization

Reproducibility and experiment tracking

Supports cloud, on-premise, hybrid usage

Works with TensorFlow, PyTorch, and Keras



Thank you!

neil@determined.ai

https://determined.ai