Mobile ML Benchmarking at Facebook and Beyond

Fei Sun

Software Engineer, AI platform, Facebook

Mobile ML Deployment at Facebook

ML Deployment at Facebook

- Machine learning workload used at Facebook
  - Neural style transfer
  - Person segmentation
  - Object detection
  - Classification
  - etc.

Neural Style Transfer

Person Segmentation

- Mermaid effect in Instagram

Object Detection

Facebook AI Performance Evaluation Platform

Challenges of ML Benchmarking

- What ML models matter?
- How to enhance ML model performance?
- How to evaluate system performance?
- How to guide future HW design?

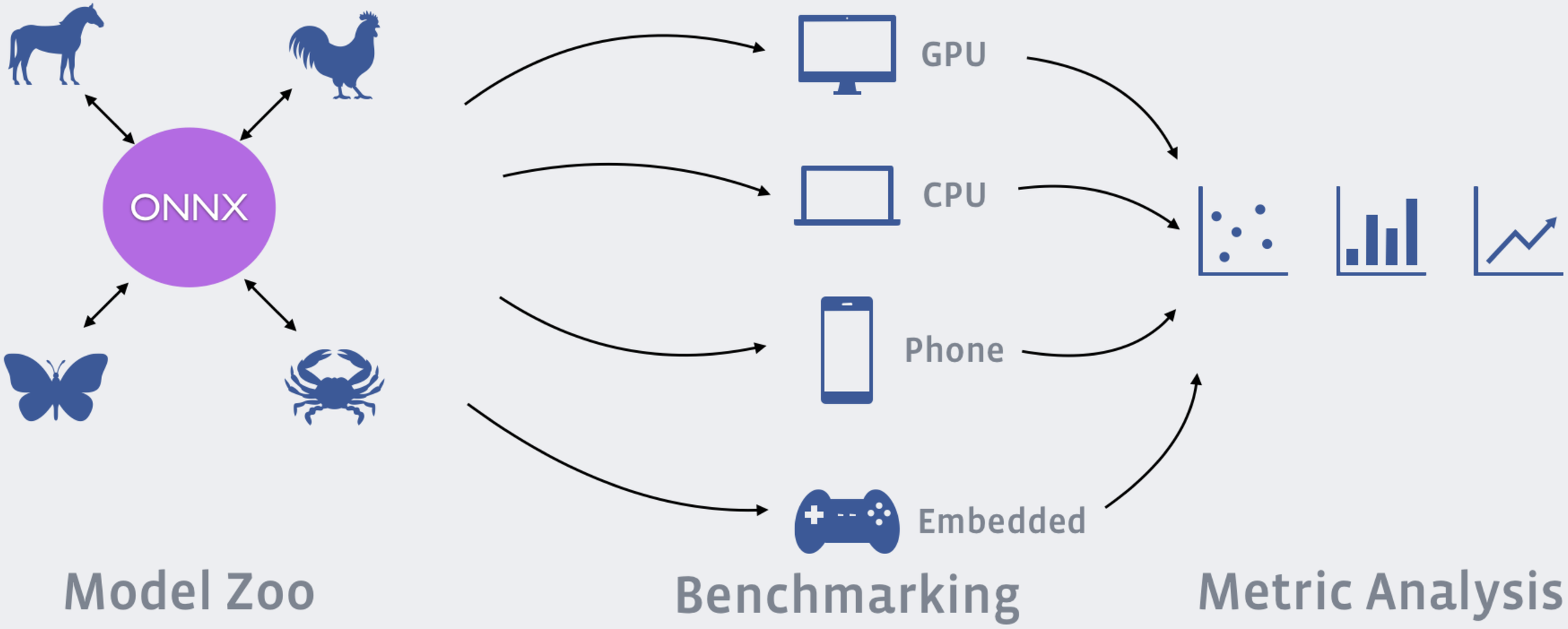
Goals of AI Performance Evaluation Platform

- Provide a model zoo on important models
- Normalize the benchmarking metrics and conditions
- Automate the benchmarking process
- Honest measurement on performance

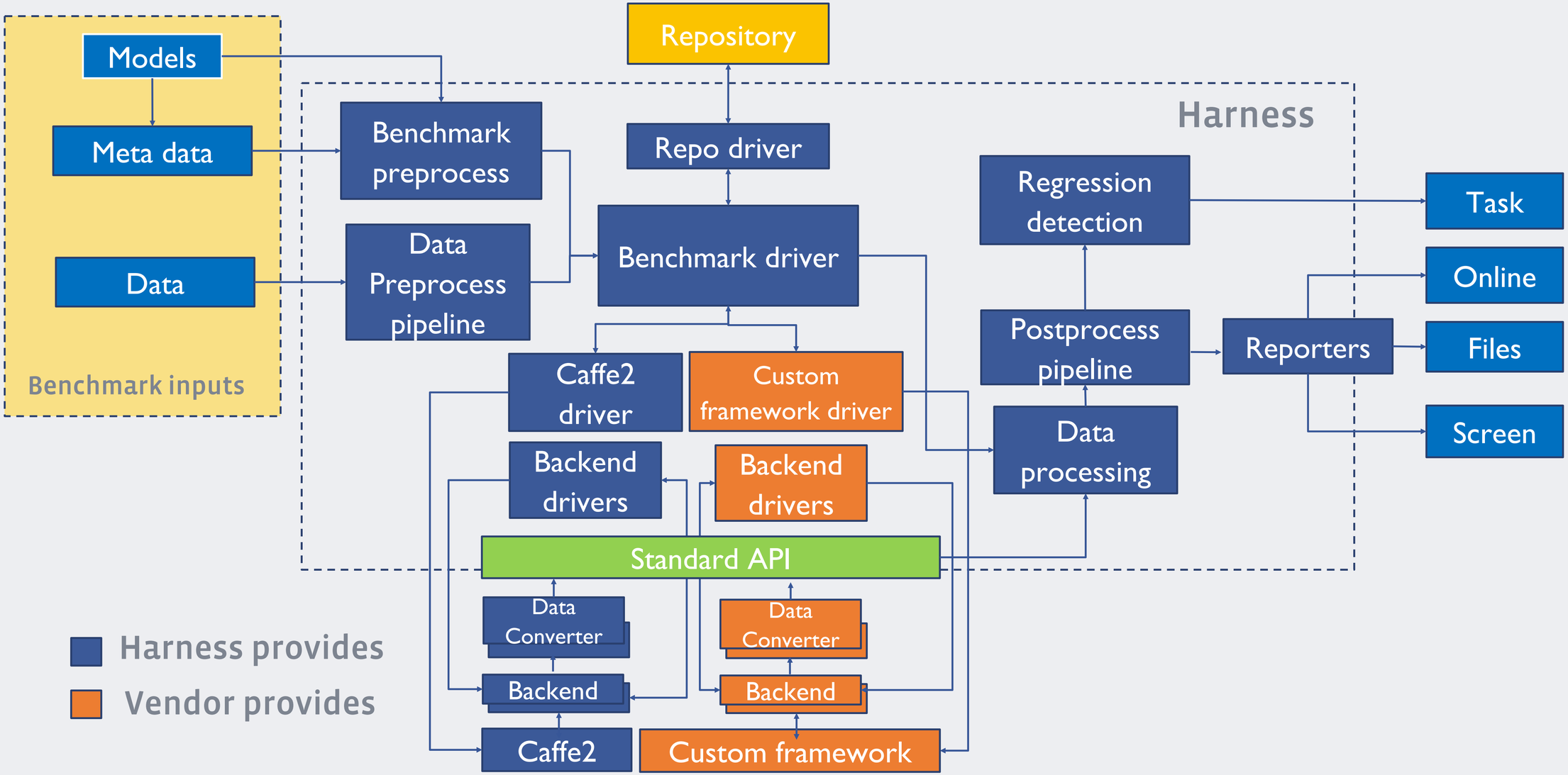
Backend & Framework Agnostic Benchmarking Platform

- Same code base deployed to different backends
- Vendor needs to plugin their backend to the platform
  - The build/run environment is unique
  - The benchmark metrics are displayed in different formats
- Good faith approach
  - Easily reproducing the result

Benchmarking Flow



Benchmark Harness Diagram



Current Features

- Support android devices, windows, and linux like host systems.
- Support Pytorch/Caffe2, TFLite etc. frameworks.
- Support latency, accuracy, FLOPs, power, energy metrics.
- Save results locally, remotely, or display on screen.
- Deal with git, hg version controls.
- Cache models, binaries locally.
- Continuously run benchmarks on new commits.
- A/B testing on regression detection.

Open Source

- Facebook AI Performance Evaluation Platform
- https://github.com/facebook/FAI-PEP

- Encourage community to contribute

Pitfalls of Mobile ML Benchmarking

Hardware Variations

- CPU
- GPU
- DSP
- NPU

Software Variations

- ML frameworks (e.g. Pytorch/Caffe2, TFLite)
- Kernel libraries (e.g. NNPACK, EIGEN)
- Compute libraries (e.g. Open CL, Open GL ES)
- Proprietary libraries
- Different build flags

OEM Influence

- Some performance related knobs are not tunable at user domain.
  - Require rooted phone or special built phone.
- But the OEM/Chipset vendors can change them for different scenarios.
- Some algorithms (e.g. scheduling) are hard-coded by OEM/Chipset vendors, but they influence performance a lot.

Run to Run Variation

- Run-to-run variation on mobile is very obvious
  - Difference between different iterations of the same run
  - Difference between different runs
- How to correlate the metrics collected from the benchmark runs to the production runs?
- How to specify the benchmark condition?

How to Avoid Benchmark Engineering

- Is it useful if
  - the hardware only runs models in benchmark suite efficiently
  - the software is only bug free running models in the benchmark suite
  - a different binary is used to benchmark each model
  - the code is bloated if the binary includes different algorithms for different models

Key Components of Mobile ML Benchmarking

- Relevant mobile ML Models
- Representative mobile ML workload
- *Unified and standardized methodology of evaluating the metrics in different scenarios*

Questions