CANDLE Project Overview

Legion Retreat, Stanford, CA
December 2022

Jamaludin Mohd Yusof
Future Architectures and Applications
Los Alamos National Laboratory
jamal@lanl.gov

EXASCALE COMPUTING PROJECT

Talk Outline

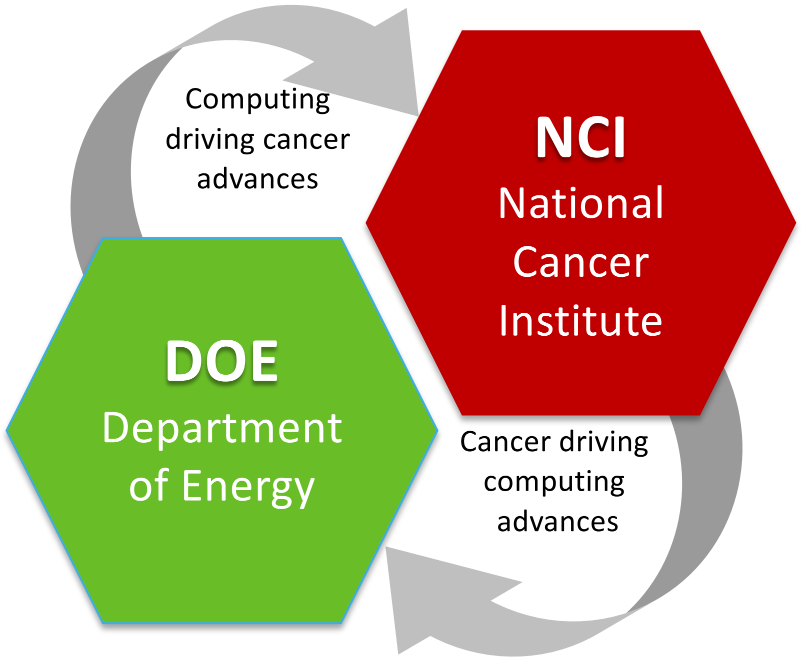
- Background
  - Three Pilots
  - Project Goals
- CANDLE Environment
  - Candle-lib
  - Benchmarks
  - Supervisor
- Current Status
- Path Forward

LA-UR-22-32709

CANDLE Background

- DOE-NCI Partnership (beginning~2016)
  - Joint Design of Advanced Computing Solutions for Cancer (JDACS4C)
  - Cancer Moonshot & National Strategic Computing Initiative
  - ANL, LLNL, LANL, ORNL, and Fredrick National Lab for Cancer Research

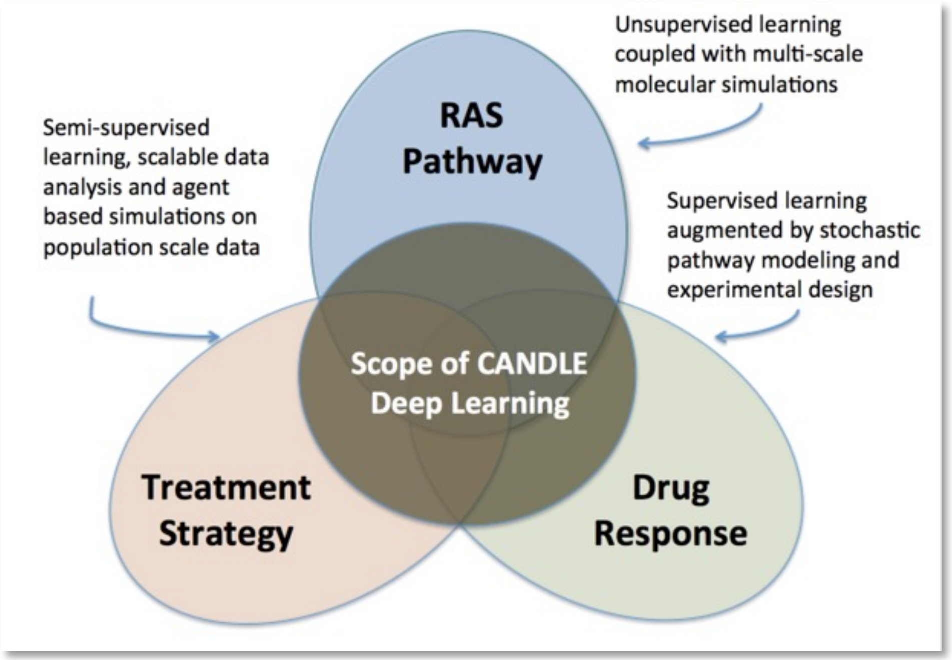
- Accelerate precision oncology capabilities
  - 3 application-focused Pilots
  - 2 cross-cutting initiatives
    - Uncertainty quantification
    - Scalable deep learning (CANDLE)



LA-UR-22-32709

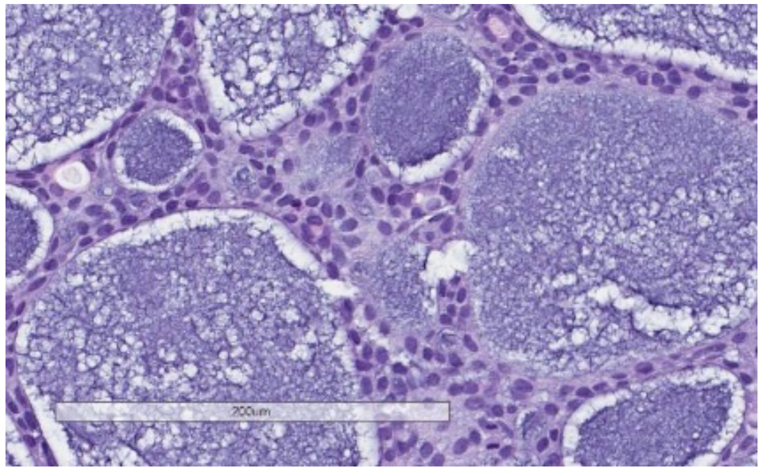
JDACS4C Pilots

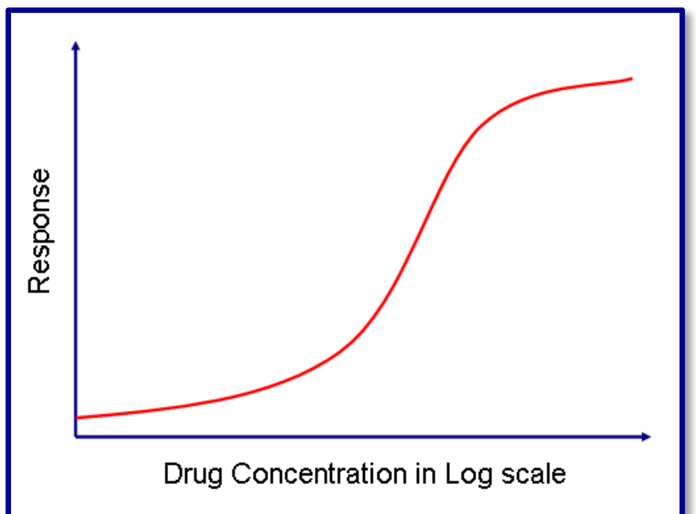
- Pilot 1: Pre-clinical models
  - Predictive patient drug response models
  - PI: Rick Stevens (ANL)
- Pilot 2: Biological models
  - Multi-scale computational biological models
  - PI: Fred Streitz (LLNL)
- Pilot 3: Cancer Surveillance
  - Insight into factors impacting clinical response
  - PI: Gina Tourassi (ORNL)



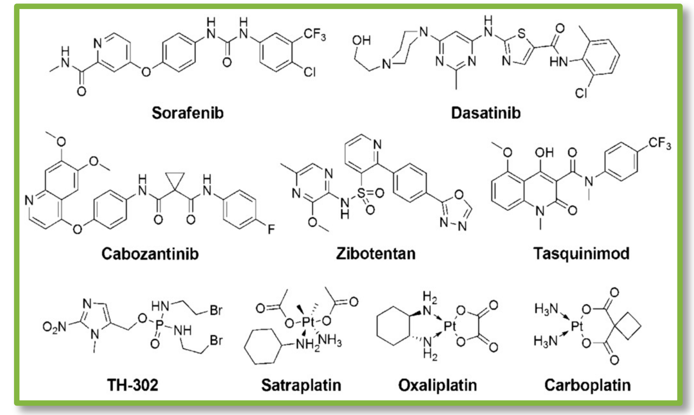
LA-UR-22-32709

Pilot 1 : Modeling Cancer Drug Response

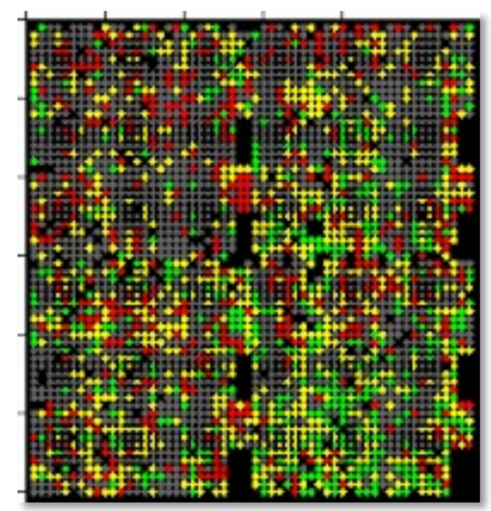




Drug (s)
- descriptors
- fingerprints
- structures
- SMILES
- dose



$$R = f(T, D, [P]^*)$$



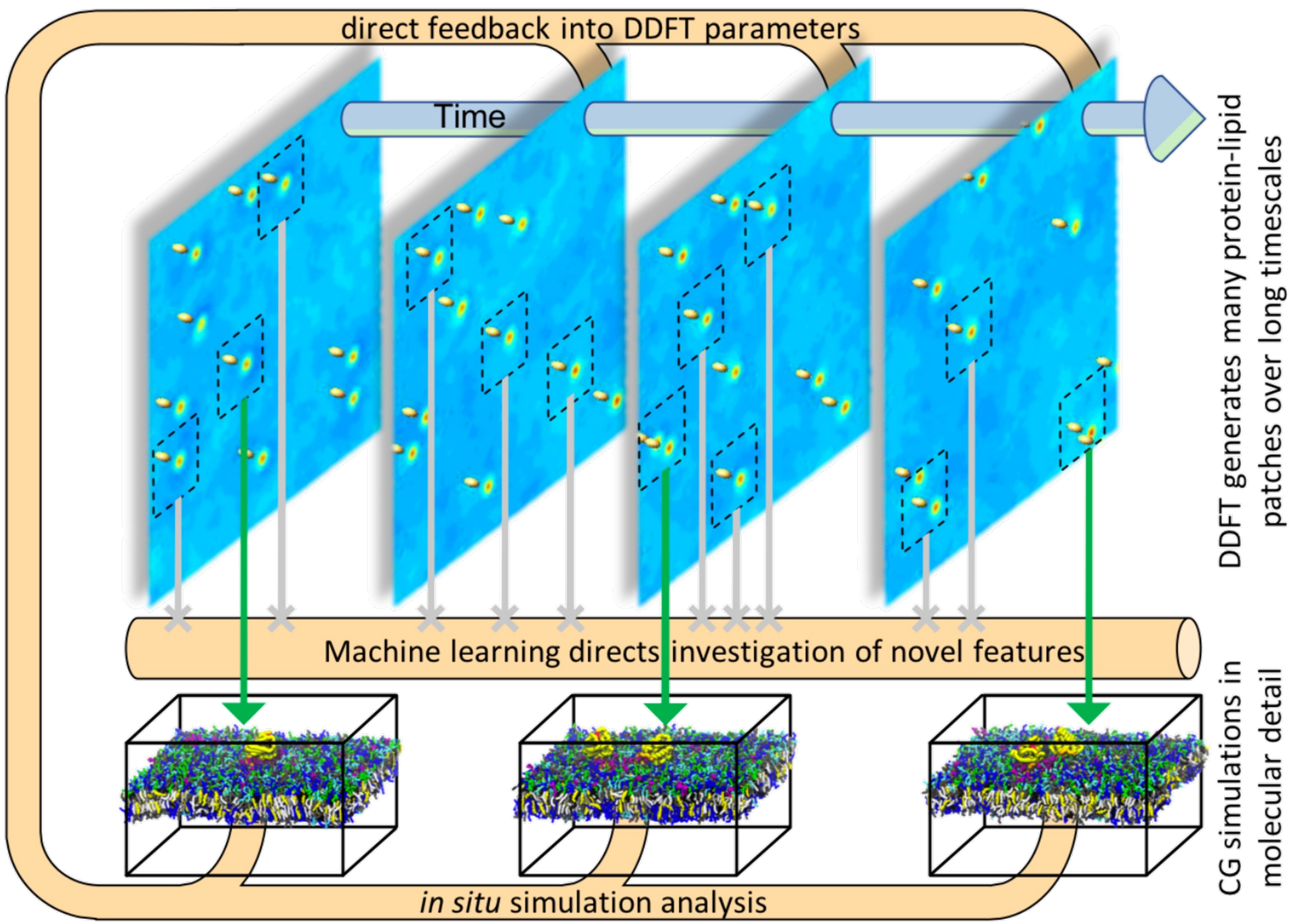
Response
- IC50
- AUC
- GI50
- % growth
- Z-score

Tumor
- gene expression levels
- SNPs
- protein abundance
- microRNA
- methylation

[P] (patient /treatment history, etc.)*

LA-UR-22-32709

Pilot 2 : Detecting interesting RAS / RAF binding

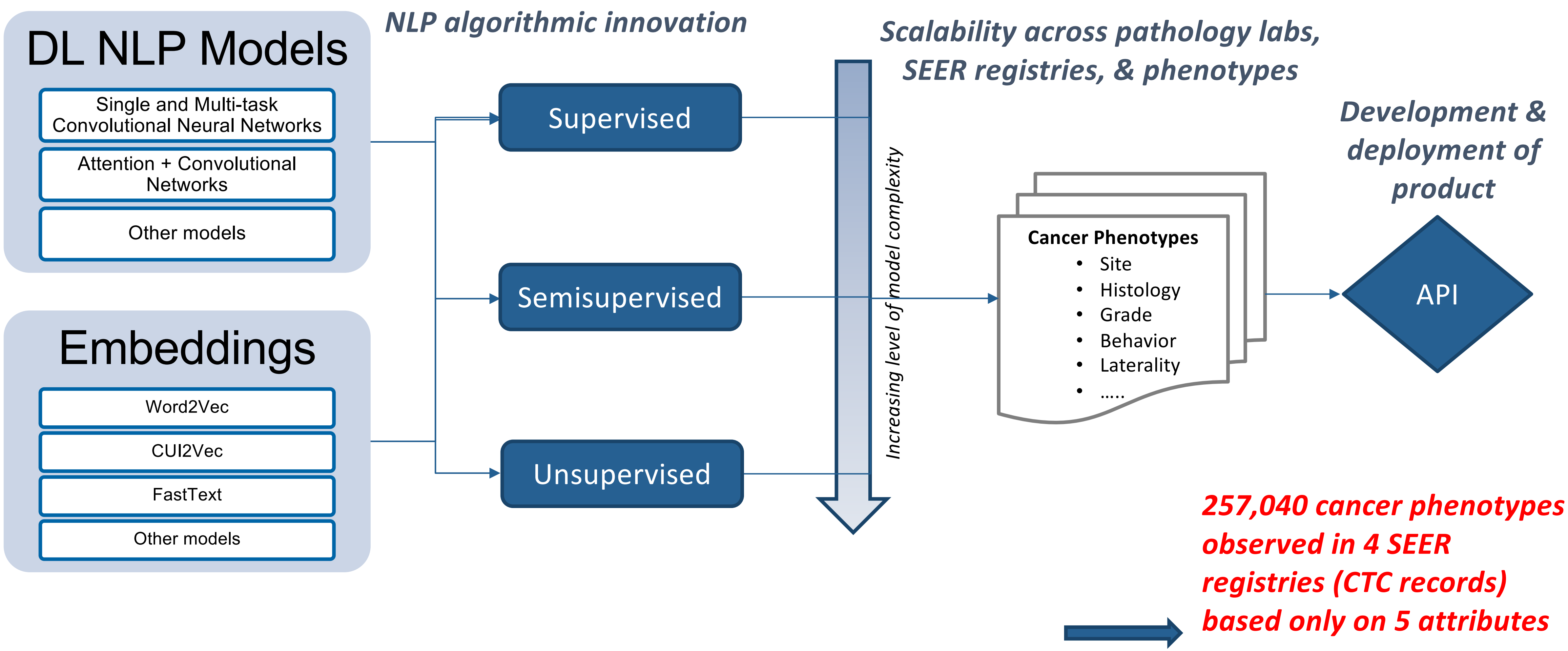


- Identify when the binding event is occurring within the simulation
- Classify the state of the binding as normal or anomalous
- Allow researchers to watch the evolution of the binding to find new methods for intervention

LA-UR-22-32709

EXASCALE COMPUTING PROJECT

Pilot 3: Treatment Strategy Problem



- 7 SEER cancer registries, 1M unique cancer patients, 3.5M pathology reports
- 70 cancer sites (306 subsites); 515 histologies; 9 grades; 7 lateralities; 4 behaviors

LA-UR-22-32709

CANDLE Challenge Problem Workflow Structure

Each challenge problem has three compute intensive phases and two data processing phases

1. Prepare input datasets, including normalization, outlier removal, joins and merges, etc.

2. Broad search of model space (AutoML+HPO) to find good performing model structures and hyper-parameter settings ⟹ running $O(10^4)$ – $O(10^6)$ model instances (HPO model selection ⟹ small number of model templates < 10)

3. Training a large-scale ensemble of the best model types on relevant training data for a “factorial study” ⟹ $O(10^6)$ models on $O(10^4)$ datasets (5x cross validation and optional bootstrap UQ) (CV model selection ⟹ small number of models < 10 per Source-Target pair)

4. Inferencing with selected models with UQ on new samples ($O(10^7)$ in (and out) of each Source-Target Pair distributions (UQ implies sampling model space $O(10^2)$ times)

5. Post-process inference output for actionable decisions and to capture performance, confidence and scoring (for validation)

LA-UR-22-32709

Project Goals

- Provide API for deep learning on DOE supercomputers
  - CANDLE Library (now candle_lib)
  - Facilitate dependencies, IO, UQ, visualization, profiling
- Support general set of deep learning workflows
  - CANDLE Supervisor
  - Portable, scalable across DOE computing platforms
- Produce example proxy application models for vendors and users
  - CANDLE Benchmarks
  - Pilot-based deep learning models

LA-UR-22-32709

The (original) CANDLE Environment

Hyperparameter Sweeps, Data Management (e.g. DIGITS, Swift, etc.)	Workflow
Network description, Execution scripting API (e.g. Keras, Mocha)	Scripting
Tensor/Graph Execution Engine (e.g. Theano, TensorFlow, LBANN-LL, etc.)	Engine
Architecture Specific Optimization Layer (e.g. cuDNN, MKL-DNN, etc.)	Optimization

LA-UR-22-32709

Team

- Argonne
  - Tom Brettin
  - Nick Collier
  - Rajeev Jain
  - Jonathan Ozik
  - Arvind Ramanathan
  - Rick Stevens
  - Richard Turgeon
  - Justin Wozniak
  - Fangfang Xia
  - Harry Yoo
  - …

- Fredrick
  - Andrew Weisman
  - George Zaki
  - …

- Livermore
  - Brian van Essen
  - Sam Ade Jacobs
  - Adam Moody
  - …

- Los Alamos
  - Christina Garcia-Cardona
  - Jamal Mohd-Yusof
  - …

- Oak Ridge
  - Shang Gao
  - John Gounley
  - Hong-Jun Yoon
  - Todd Young
  - …

LA-UR-22-32709

CANDLE Library (candle_lib)

- Purpose
  - To streamline the writing of CANDLE-compliant codes
    - Provides various utility packages that promote reuse and streamline code development
    - I/O, UQ, checkpointing, visualization, profiling etc
  - Allow rapid prototyping and exploration of hyperparameters
    - Command line interface to modify hyperparameters
    - Integrate with the Supervisor framework
- Historical perspective
  - Consolidation of frequently used functionality from the Benchmark codes
  - Evolving to incorporate new functionality as needed
    - Improved usability over time

LA-UR-22-32709

Benefits provided by CANDLE

- Consistent
  - Standardized network specification with a "`default_model_file`"
  - Standardized command line intercept protocol
  - Standardized default values across frameworks
  - Ideal for testing the same problems with consistency on new DOE hardware
- Convenient
  - Pass arguments via command line
    - Standard keywords parsed automatically, user can add new ones
  - Modify the default file
    - Provide a new default model specification '`--config_file new_default_model.txt`'

LA-UR-22-32709

Deep learning on supercomputers

- Steep learning curve with myriad technologies
  - Workflow manager (Swift/T, EMEWS) ; Scheduler ; Scripting
  - Deep learning (Keras, TensorFlow, Horovod)
  - Optimization algorithms (R, Python)
  - MPI implementation (MVAPICH, Open MPI)

etc. …

LA-UR-22-32709

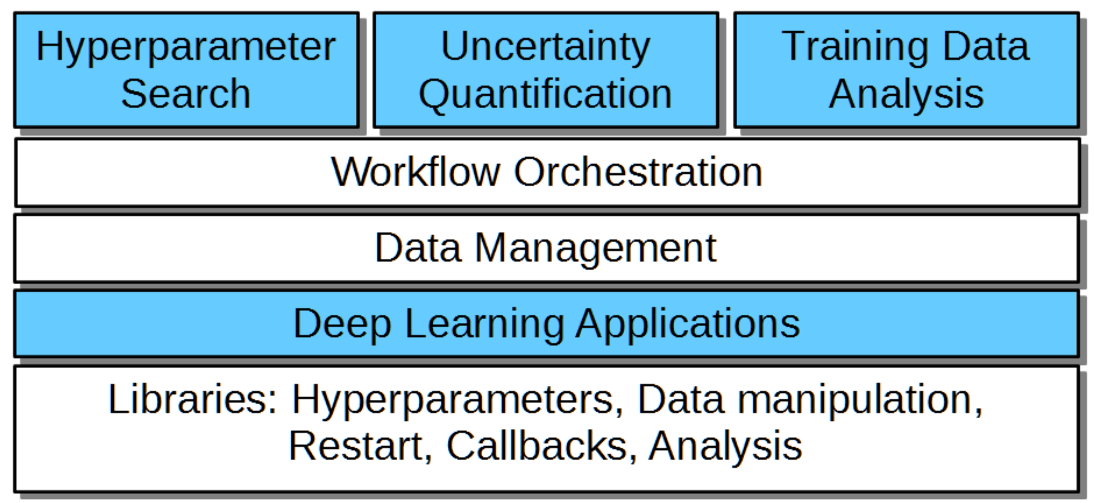
CANDLE/Supervisor Goals

- Develop an exascale deep learning environment for cancer
- Building on open source deep learning frameworks
- Optimization for CORAL and future exascale platforms
- Support all three Pilot project needs for deep learning
- Collaborate with DOE computing centers, HPC vendors and ECP co-design and software technology projects
- Mission statement: Enable the most challenging deep learning problems in cancer research to run on the most capable supercomputers in the DOE
- Provide a path forward for machine learning applications at the largest scale…

LA-UR-22-32709

CANDLE/Supervisor overview

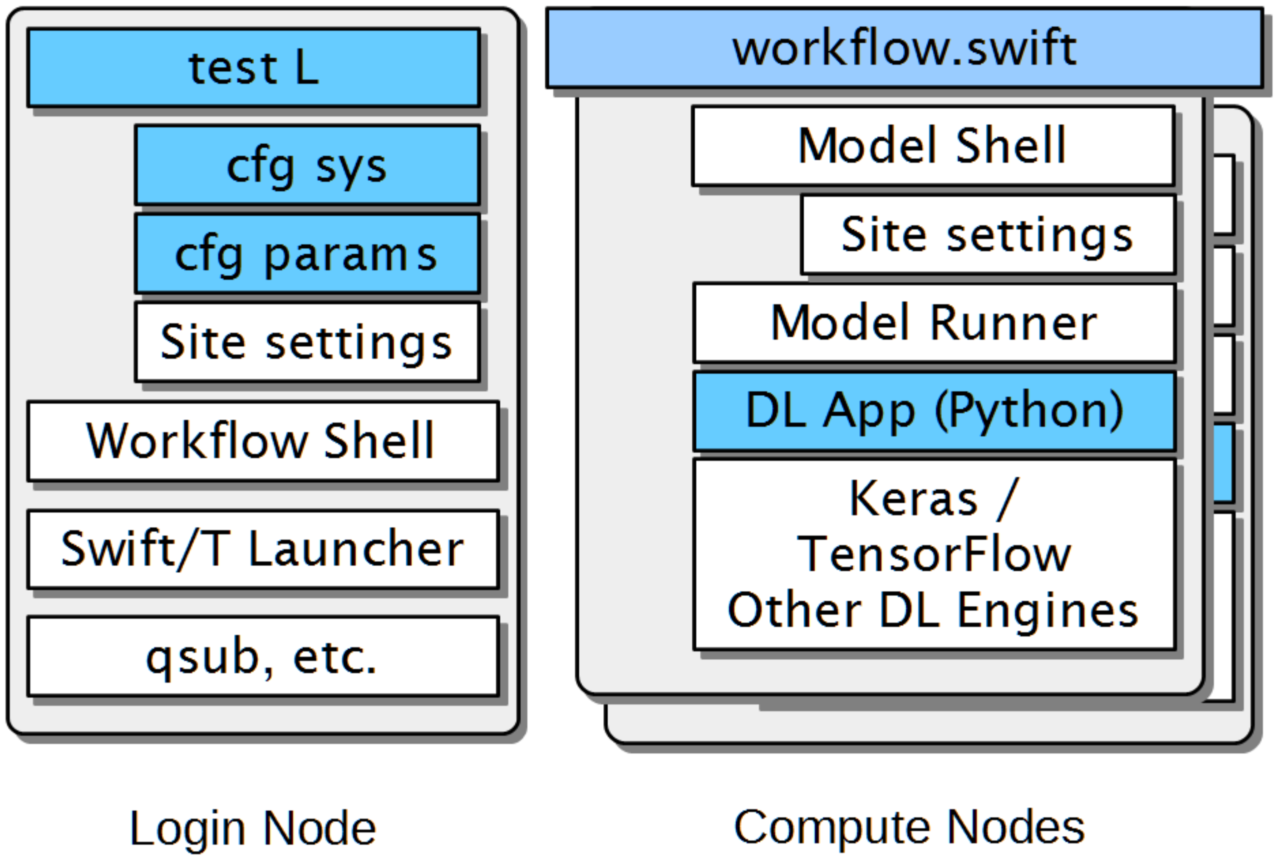
- CANDLE/Supervisor consists of several high-level workflows:
  - Capable of modifying/controlling application parameters dynamically as the workflow progresses and training runs complete
  - Distribute work across large computing infrastructure, manage progress
- Underlying applications are Python programs that use Keras/TensorFlow
- “User code” shown in blue
- “Provided tools” shown in white
- New studies would be developed by modifying the blue sections



LA-UR-22-32709

CANDLE/Supervisor Implementation

- Runs start with a test script
- CFG scripts contain settings for a system or parameters for a given study (e.g., search space)
- Reusable site settings
- The workflow shell script sets up the run
- Swift/T launches and manages the workflow
- Reusable Model scripts set up each app run
- The DL app uses Keras/TF plus CANDLE Python libraries



LA-UR-22-32709

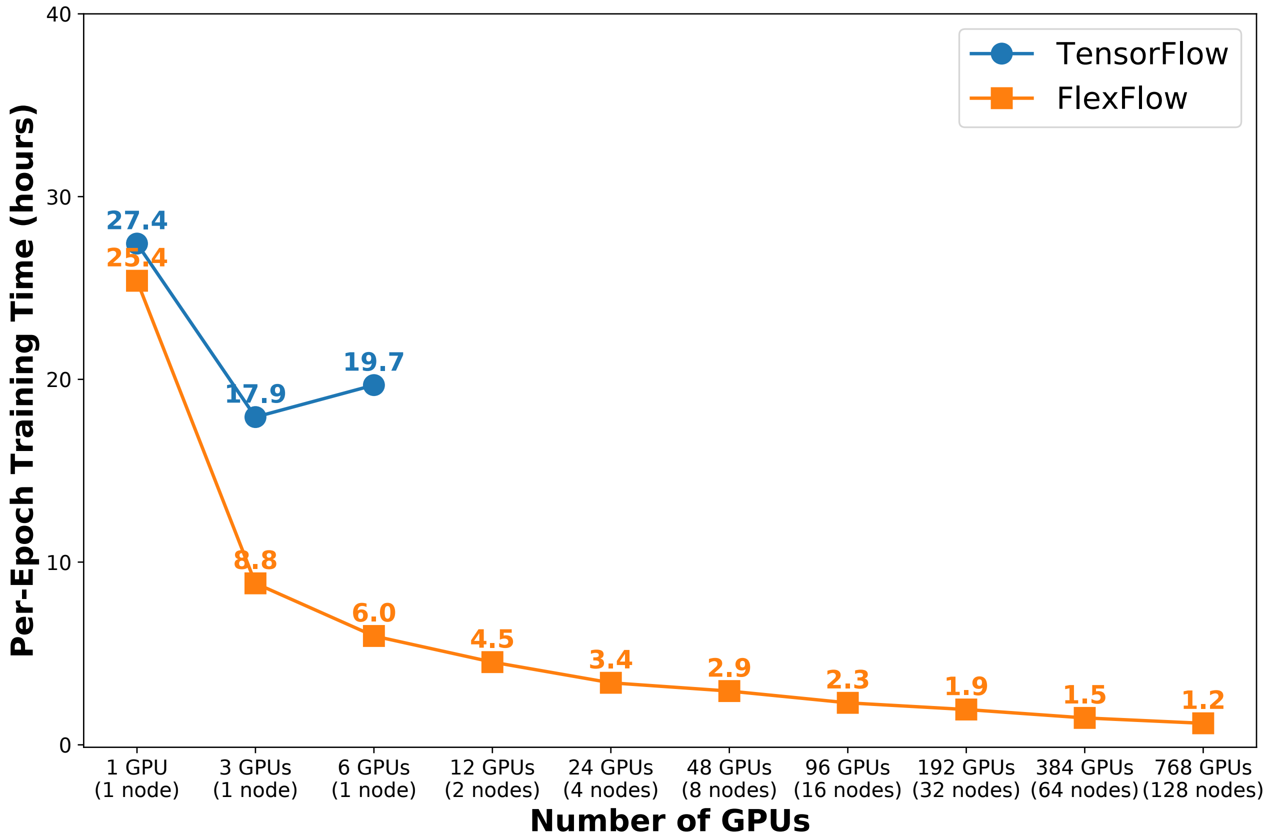
Project Evolution

- The composition of the landscape has changed significantly
  - Workflows
    - Accommodate more flexible scheduling, larger models and multiple coupled codes
  - Scripting:
    - Essentially Keras/Tensorflow and PyTorch.
    - Proliferation of external models
    - Larger models
  - Engine
    - Tensorflow and PyTorch
      - LBANN and FlexFlow are alternatives specifically designed for parallelism performance
  - Optimization
    - Proliferation of alternate hardware, including AMD and Intel GPUs, specialized hardware (Cerebras, SambaNova)

LA-UR-22-32709

Current Status

- Benchmark
  - FlexFlow support for Keras and PyTorch
    - Relatively straightforward porting
  - Example
    - CANDLE P1 Uno Benchmark ported to (old) FlexFlow
    - Tested on Summit at ORNL
      - IBM Power9 + Nvidia V100
  - P3B9 port in progress



LA-UR-22-32709

Current Status

- Supervisor
  - Supervisor workflows run inside a single big MPI job - allows for efficient communication and resource reuse
  - For single-node training runs, integrating with a FF application should be straightforward. Can be called via shell command or library interface
  - For multi-node training runs, Supervisor has the capability to invoke parallel engines like Horovod by creating a subcommunicator and invoking the DL library on those resources
    - Need to demonstrate this with the Legion runtime and eventually a full FF example

LA-UR-22-32709

Path Forward

- All three Pilots have spawned follow-on projects
  - Pilot 1-> IMPROVE: Innovative Methodologies and New Data for Predictive Oncology Model Evaluation
- Builds on Pilot 1 and uses a new engagement model based on extensive collaboration with the cancer research community
  - Uses candle_lib to unify the interface, integrate with Supervisor
- Two related goals aimed at IMPROVING deep learning models for predicting Drug Response in Tumors:
  - **Aim 1: IMPROVE Models :** Development of semi-automatic protocols for comparing deep learning model and identifying model attributes that contribute to prediction performance with the goal of IMPROVING predictive models of drug response
  - **Aim 2: IMPROVE Data:** Development of protocols for specifying drug screening experiments and to generate new data explicitly aimed at IMPROVING predictive models of drug response

LA-UR-22-32709

Path Forward

- Pilot 3-> MOSSAIC
  - Modeling Outcomes using Surveillance data and Scalable Artificial Intelligence for Cancer
- Evolution of models from multi-task CNN to HiSAN to Transformers
- Issues
  - Clinical reports are longer than typical 512 tokens (up to ~ 4k)
    - Resource requirement is quadratic
  - Pretrained models do not perform well
    - Require training from scratch on private datasets
  - Currently running ~100M trainable parameters, millions of samples
    - Require custom sparse attention implementation to make training viable
      - Due to memory limits, compute performance
  - Target is ~1B parameter model
    - FlexFlow can help here

LA-UR-22-32709

Path Forward

- Future CANDLE/FlexFlow integration needs
  - Workflows
    - Supervisor/FlexFlow interoperability
  - Scripting/Engine
    - Better candle_lib support for FlexFlow
    - Streamline FF porting from Keras/Tensorflow/PyTorch
      - IMPROVE will generate a stream of models
    - Transformer model support (all 3 Pilots)
      - Training from scratch needed for accuracy
  - Optimization
    - Legion/FlexFlow support on AMD, Intel accelerators, others (?)

LA-UR-22-32709

Notable Achievements

- 2017 HPCWire Editors Choice Award
- Pilot 1
  - Pivot from cancer drug discovery to COVID-19 during the pandemic, discovery of ~60 candidate molecules
  - Development of novel data transformation methods to improve learning
- Pilot 3
  - Development of novel abstaining classifier to satisfy accuracy requirements to replace human annotators
  - Initial deployment/testing of the multi-task classifier to select SEER registries

LA-UR-22-32709

Repository

- Links
  - Codebase: https://github.com/ECP-CANDLE
  - Documentation: https://ecp-candle.github.io/Candle

- Repos
  - candle_lib
  - Benchmarks
  - Supervisor

LA-UR-22-32709

Acknowledgments

This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations - the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, to support the nation's exascale computing imperative.

EXASCALE COMPUTING PROJECT