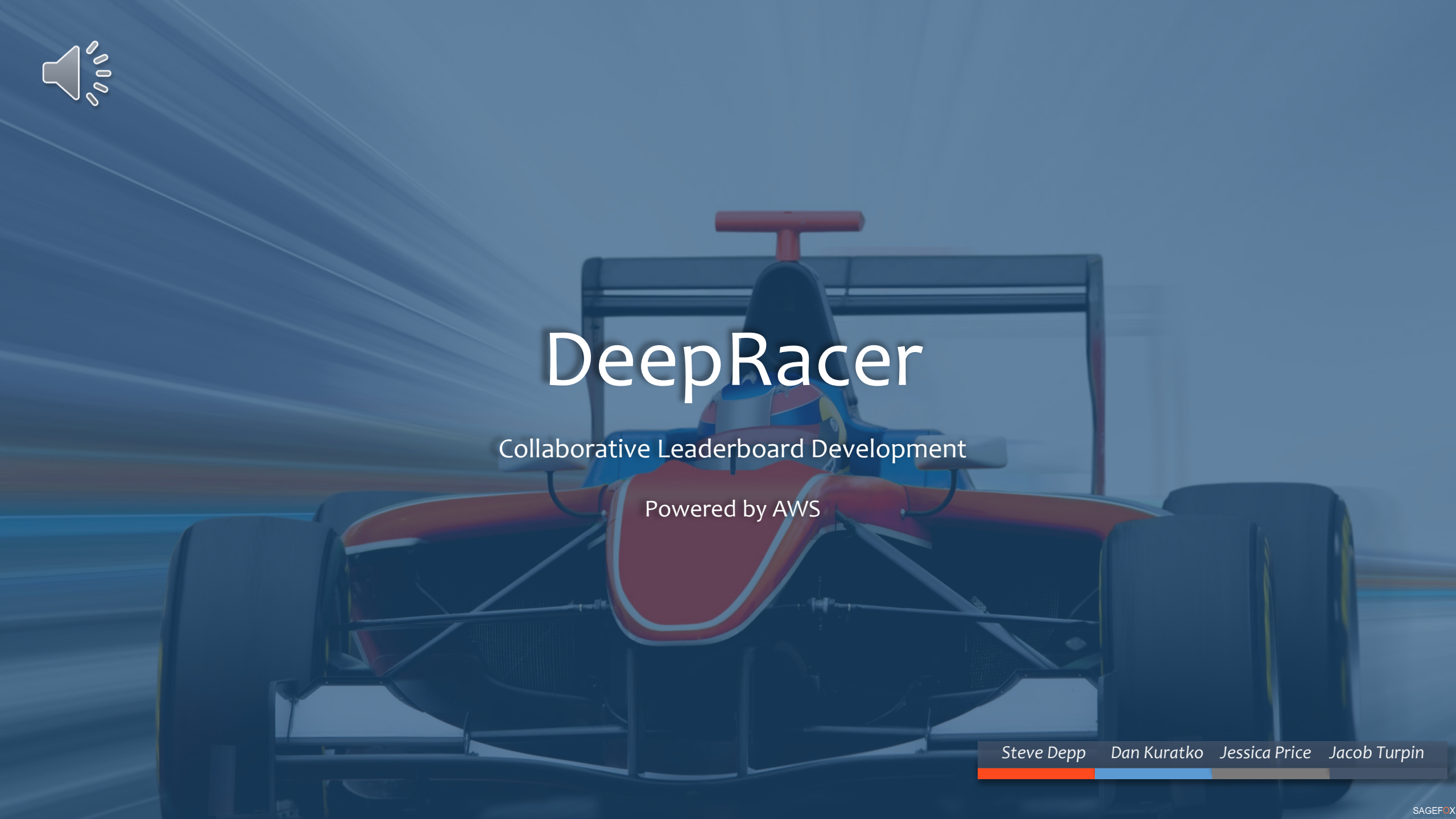


DeepRacer

Collaborative Leaderboard Development

Powered by AWS

Steve Depp *Dan Kuratko* *Jessica Price* *Jacob Turpin*

DeepRacer

CLI Driven Leaderboard

Concept

Build, train and test an AWS DeepRacer virtual race car

Submit design and performance data to the cloud

View a leaderboard composed of many DeepRacer submissions

01

AWS DeepRacer Model

02

Submit Model Data

03

View a DeepRacer Leaderboard

BUILD A RACECAR

Players create an AWS DeepRacer model, tune their car by modifying hyperparameters, and train their models on the practice track

SUBMIT YOUR CAR

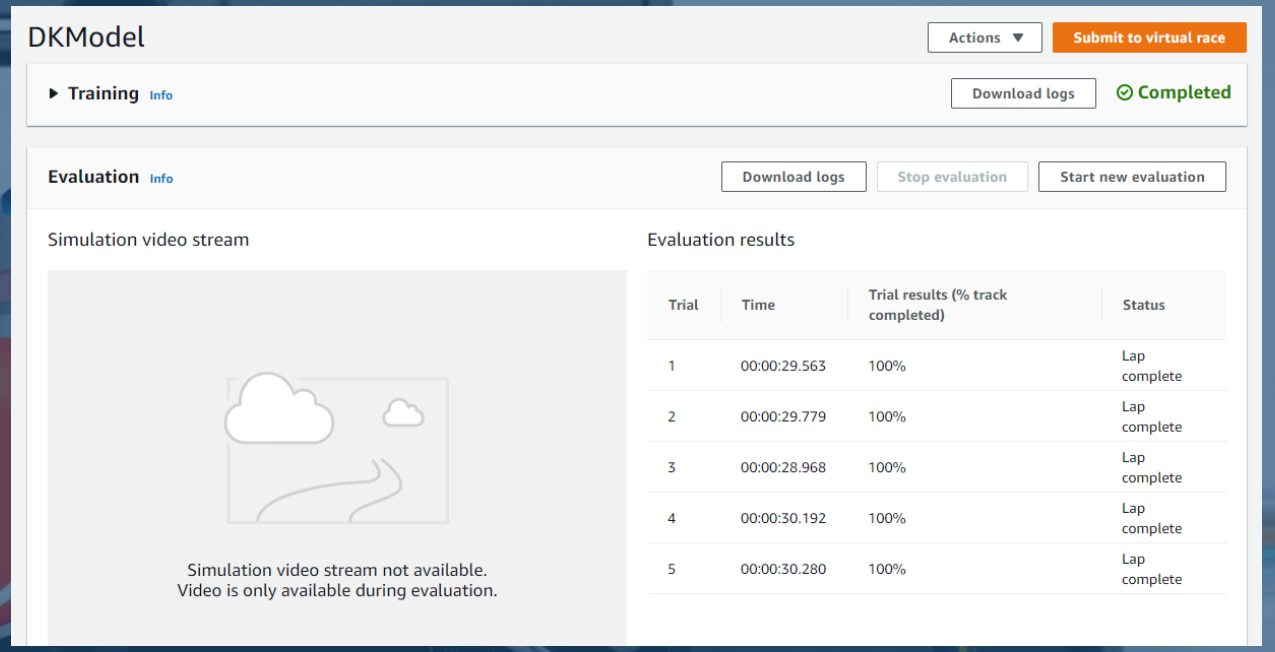
Players submit their model specs and performance into a DynamoDB table through a CLI that pulls from DeepRacer log files

COMPARE RACERS

Players access a collaborative leaderboard to see how their model stacks up against the competition

Constructing a (Virtual) Race Car

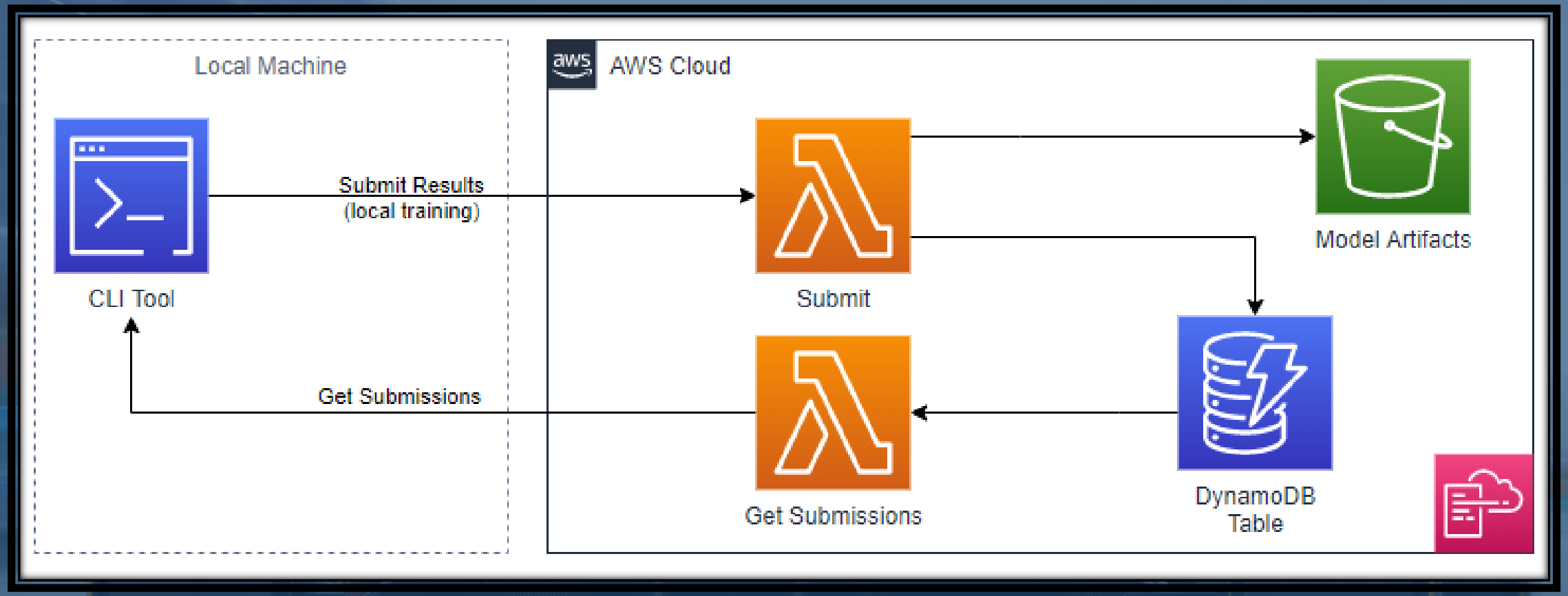
AWS DeepRacer



Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.999
Loss type	Huber
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Architecture

AWS Cloud Formation



Infrastructure

AWS Cloud Formation

CLI Tool

Racers submit their data and gain access to the leaderboard through a CLT

Lambda Functions

CLI initiates “put” and “pull” lambda functions to move the data

DynamoDB Table

Racer data captured in one central table

Assembler

Racer log files collected from user machines

S3 Bucket

DynamoDB table stored in the cloud

Command Line Interface

AWS Command Line Interface

The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI v2 offers several new features including improved installers, new configuration options such as AWS Single Sign-On (SSO), and various interactive features.

Getting Started » CLI Reference » GitHub Project » Community Forum »

Screen Shot of CLI

Assembler & DynamoDB

- **Assembler:** Collects data from downloaded SpeedRacer files
- **LeaderboardTable: S**tores data from new assembler.py file into DynamoDB table
  - Sort and query functionality
  - Source for get_submissions Lambda function

ASSEMBLER

Assembles data from 5 files

- ✓ Model_data.json
- ✓ Hyperparameters.json
- ✓ Training_params.yaml
- ✓ Training.json
- ✓ Evaluation.json

LEADERBOARDTABLE

Two key variables defined

- Track
- Person

All other variables are called through the Lambda function

Lambda Functions

Push & Pull

Basic Setup

- Author From Scratch
- Fill in Basic Information
- Set Permissions

Push Function

- Pulls specific data from .json and .yaml files
- Sends data into DynamoDB table

Pull Function

- Queries DynamoDB table
- Develops a table for user to view and/or save to local device

Lambda Code

Push Lambda

Pull Lambda

LAMBDA CODE

LAMBDA CODE

RESULTS

Output Example

ADD OUTPUT EXAMPLE

Discussion

- ✓ List successes
- ✓ List successes
- ✓ List successes
- ✓ List successes

SUSCESSES

LIMITATIONS

- ✓ List limitations
- ✓ List limitations
- ✓ List limitations
- ✓ List limitations

- ✓ Educational tool
- ✓ List Use Cases
- ✓ List Use Cases
- ✓ List Use Cases

USE CASES

FUTURE DEVELOPMENT

- ✓ List Future Development
- ✓ List Future Development
- ✓ List Future Development
- ✓ List Future Development