

Использования моделей машинного обучения в программных стеках экспериментов



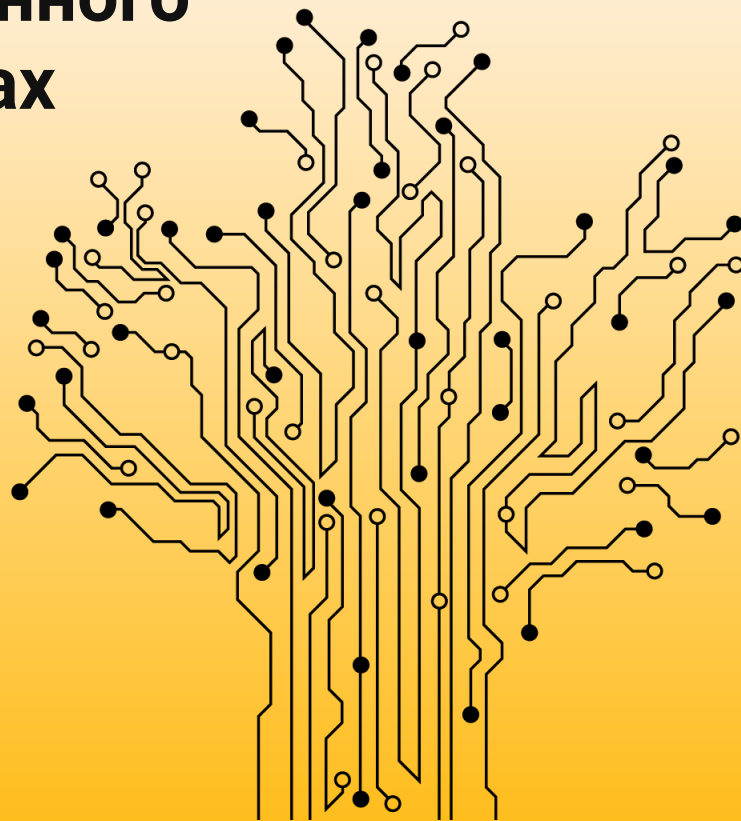
LAMBDA · HSE

Фёдор Ратников

Научно-учебная лаборатория методов анализа больших данных

Институт искусственного интеллекта и цифровых наук

Факультет компьютерных наук НИУ "Высшая школа экономики"



Контекст

Различные модели машинного обучения (ML) широко используются на различных этапах цепочки обработки данных:

➤ Реконструкция, отборы, анализ, генерация, ...

Модели машинного обучения мощны, но непосредственно зависят от данных, на которых они тренировались

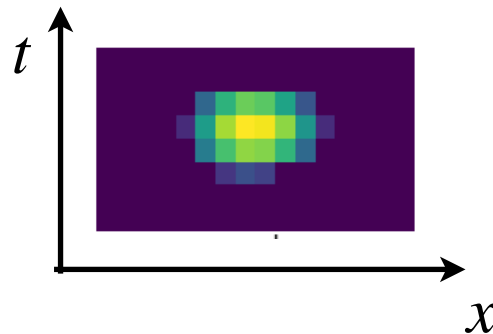
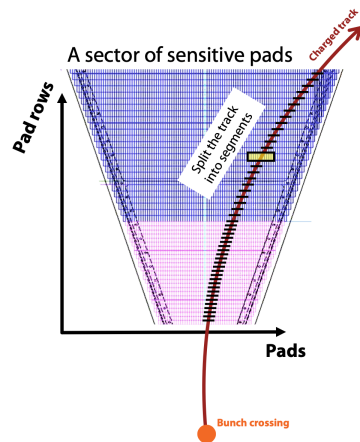
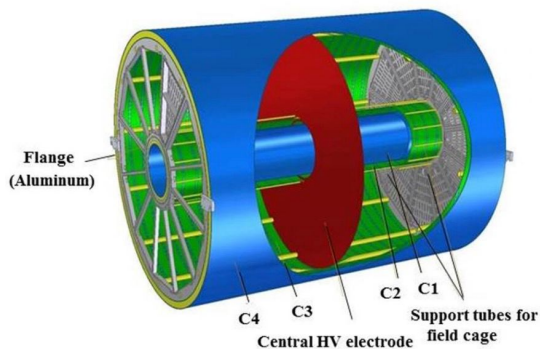
➤ Данные и их условия меняются во времени, поэтому “ML модель” - это не одна стабильная модель, которую достаточно загрузить в программный стек эксперимента, а библиотека различных моделей, подгружаемых в зависимости от конкретных условий данной задачи

■ Подходящей модели под конкретные условия может не существовать вовсе

■ Тогда модель придется строить на лету

Подход был разработан в рамках задачи создания модели быстрой симуляции отклика TPC для детектора MPD

Эксперимент MPD: быстрая симуляция ТРС



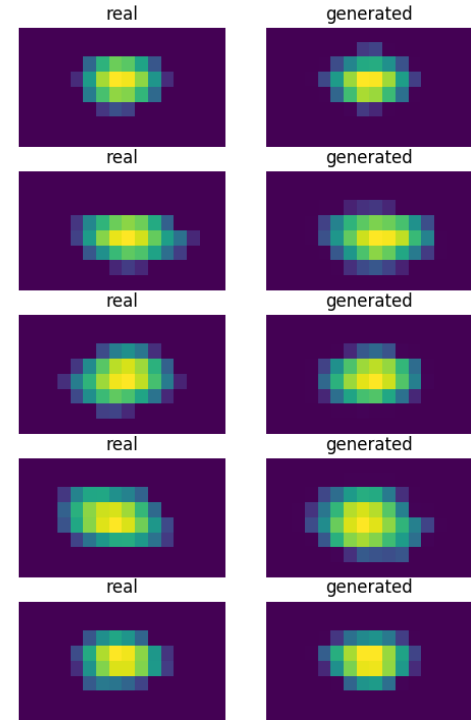
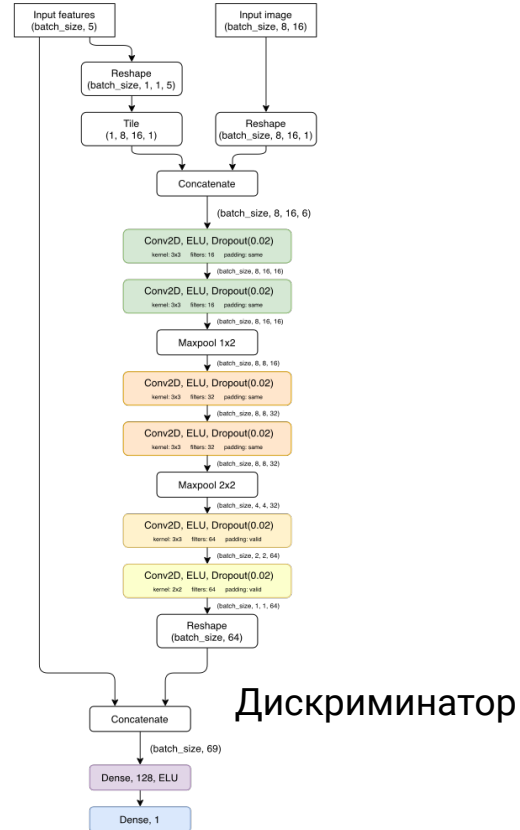
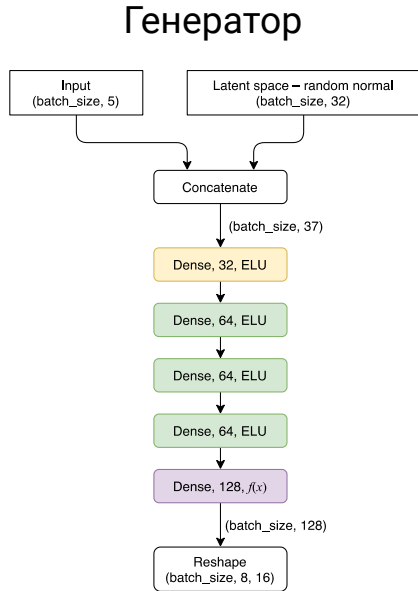
Суррогатная генеративная модель

➤ Вход: 6 параметров трека

➤ Выход: стохастический сигнал

■ 8 (пады) × 16 (время) отклик для каждого ряда паров

Быстрая симуляция ТРС: генеративные-состязательные сети



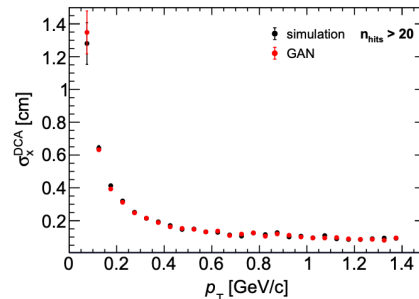
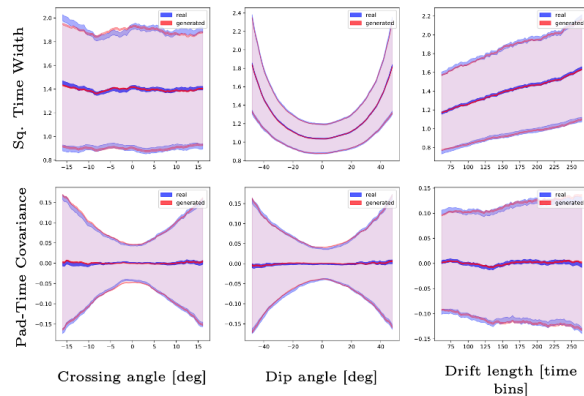
Быстрая симуляция TRC: валидация

Низкий уровень

- вычисляем первые и вторые моменты сигнального кластера
- оцениваем разницу между этими моментами для оригинальных и сгенерированных сигналов
- для разных значений параметров

Высокий уровень

- интегрируем модель в программный стек симуляции детектора
- сравниваем качества реконструкции



(a) Distance of closest approach resolution along x

Обучение модели

- ▶ Требуется соответствующий обучающий датасет, созданный с использованием базового стимулятора (GEANT)

Валидация модели

- ▶ Требуются обучающий датасет, а также новый, созданный с использованием вновь обученной модели

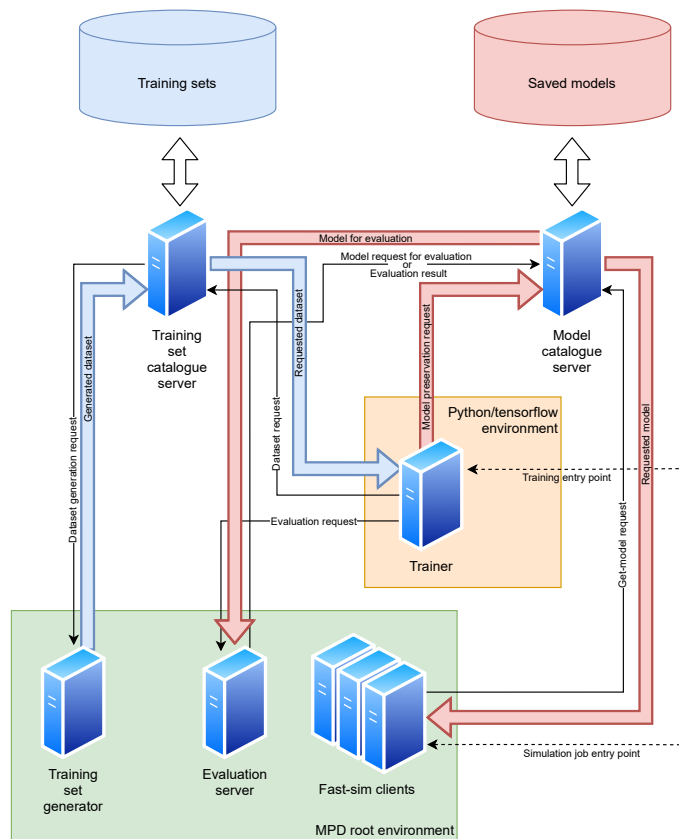
Каталогизация и сохранение модели

- ▶ Требуется уникальный ключ (хэш) модели, отражающий условия и конфигурации программного стека при создании и тренировке модели

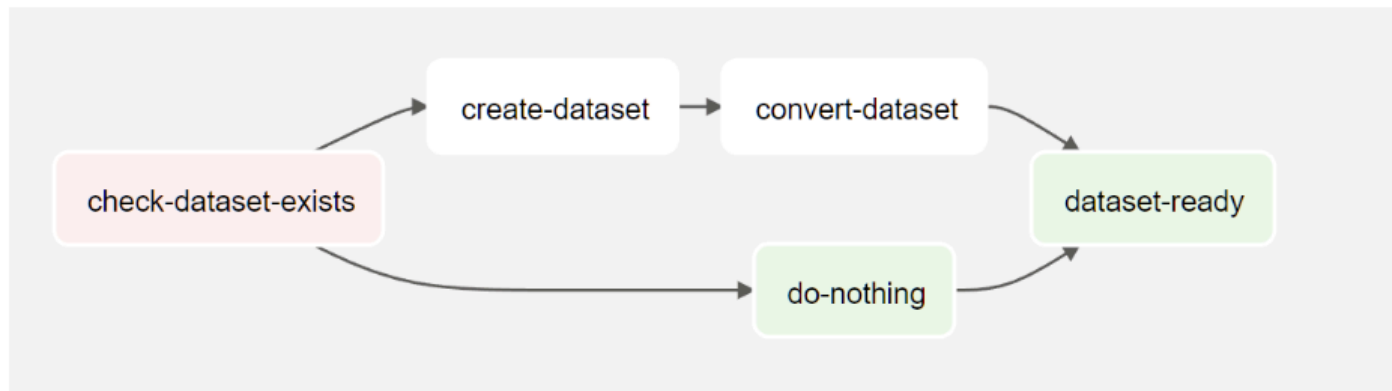
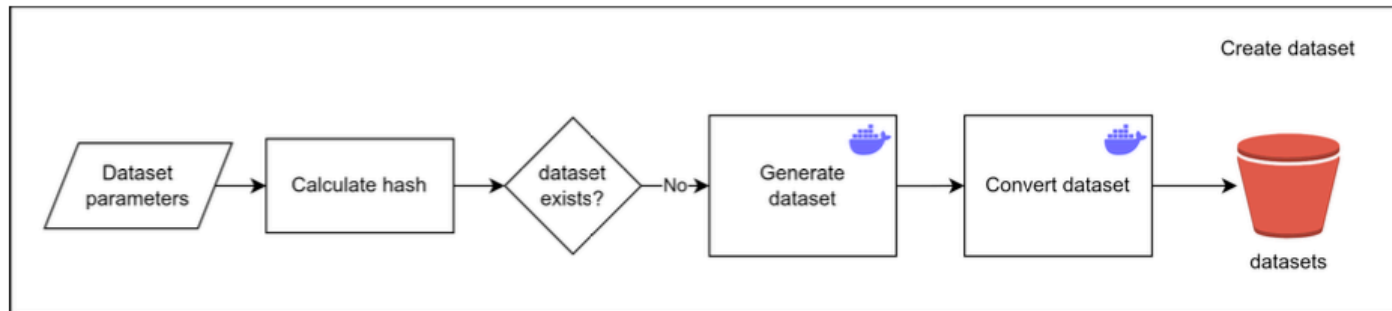
Использование модели для быстрой симуляции

- ▶ Требуется найти нужную модель в хранилище и загрузить ее в программный стек для использования

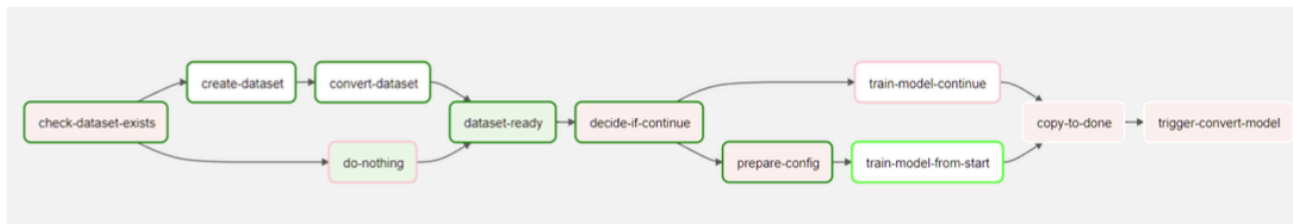
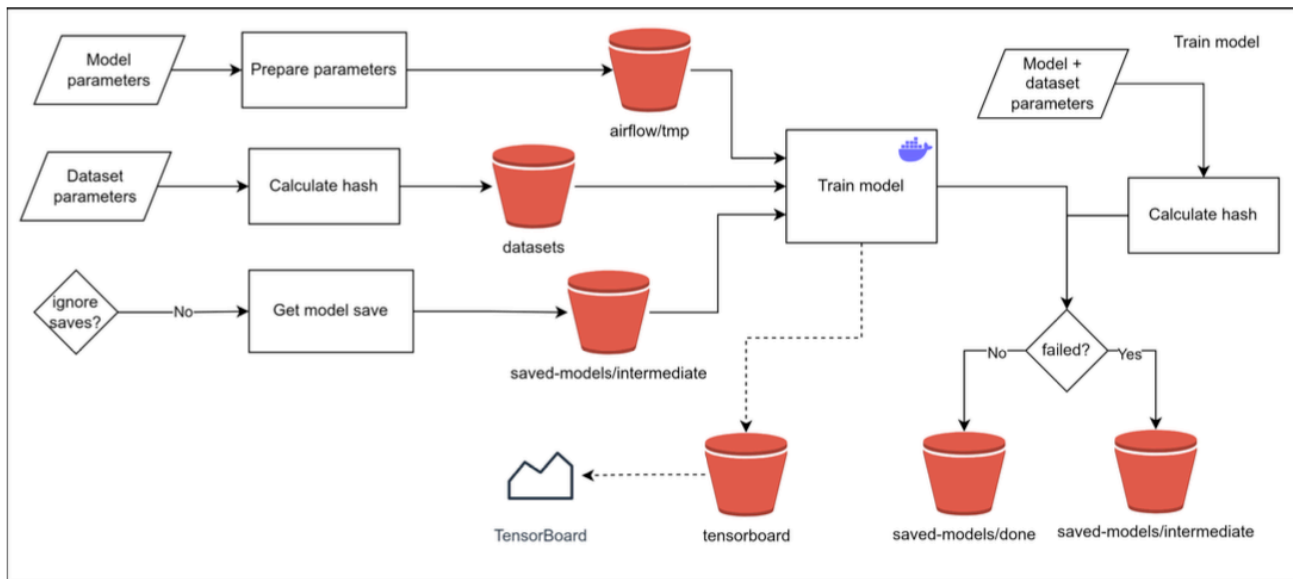
Взаимодействие компонент



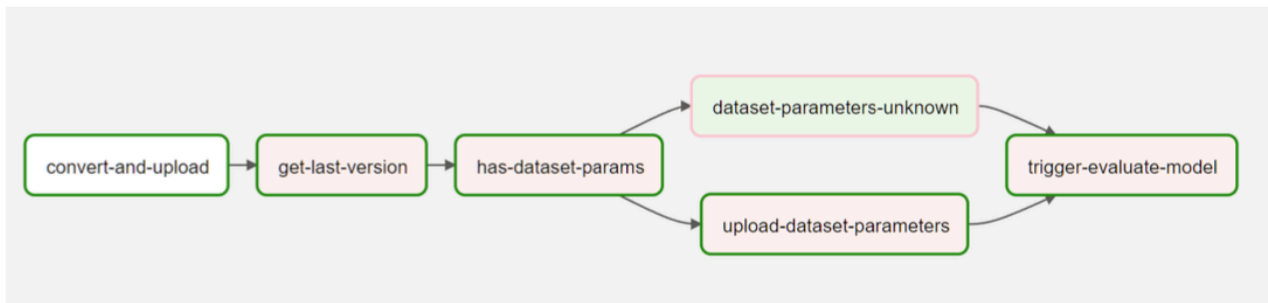
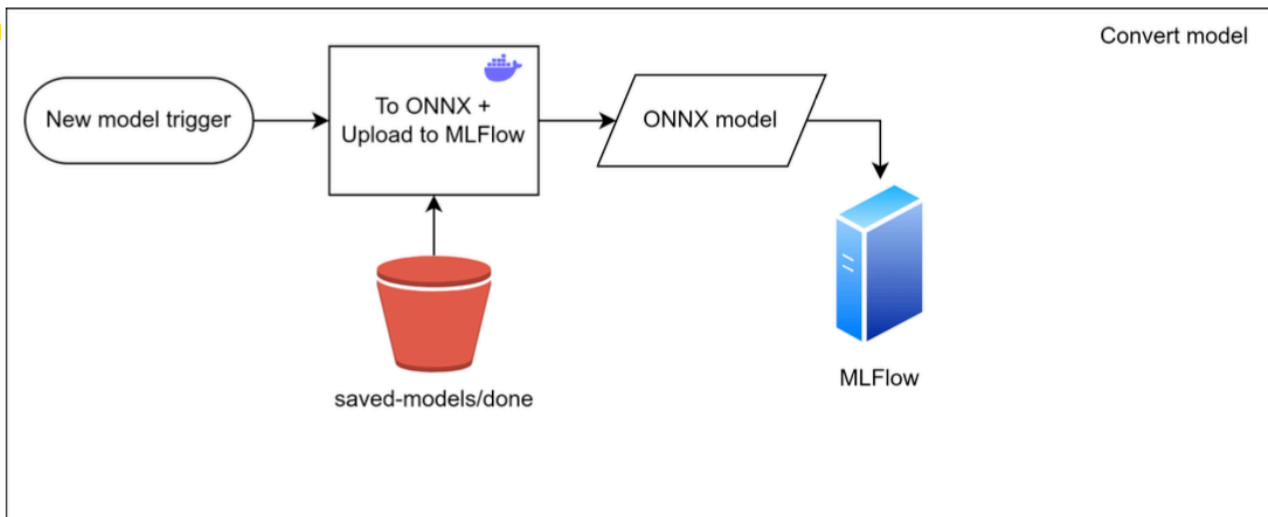
Создание обучающего датасета



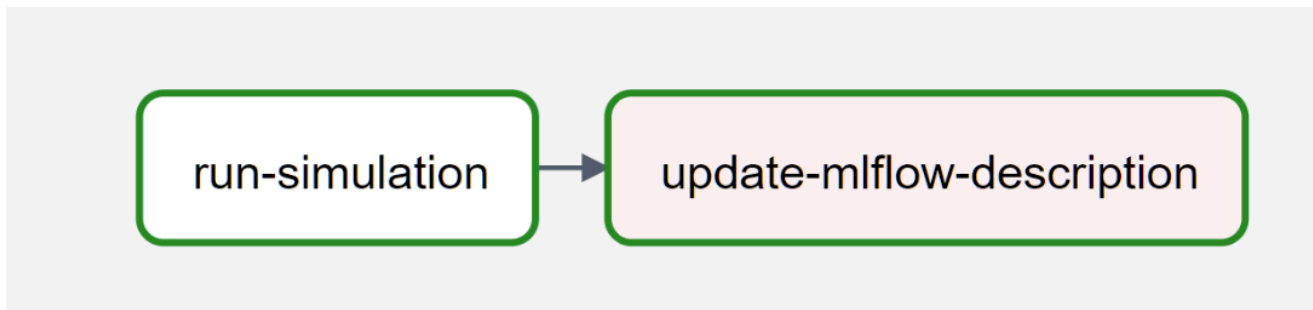
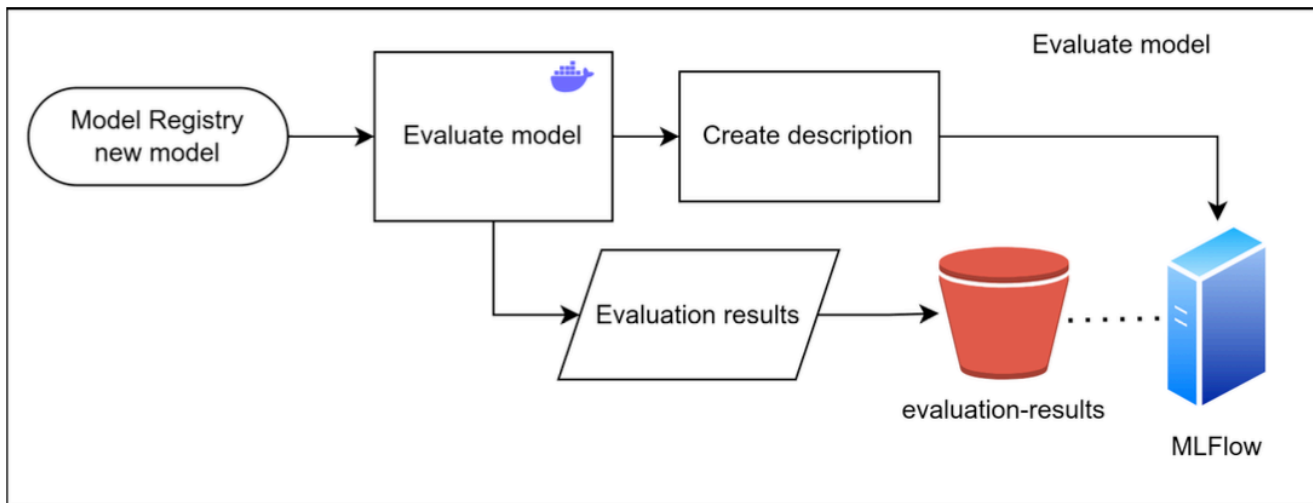
Обучение новой модели



Экспорт новой модели на сервер каталогизации



Валидация новой модели

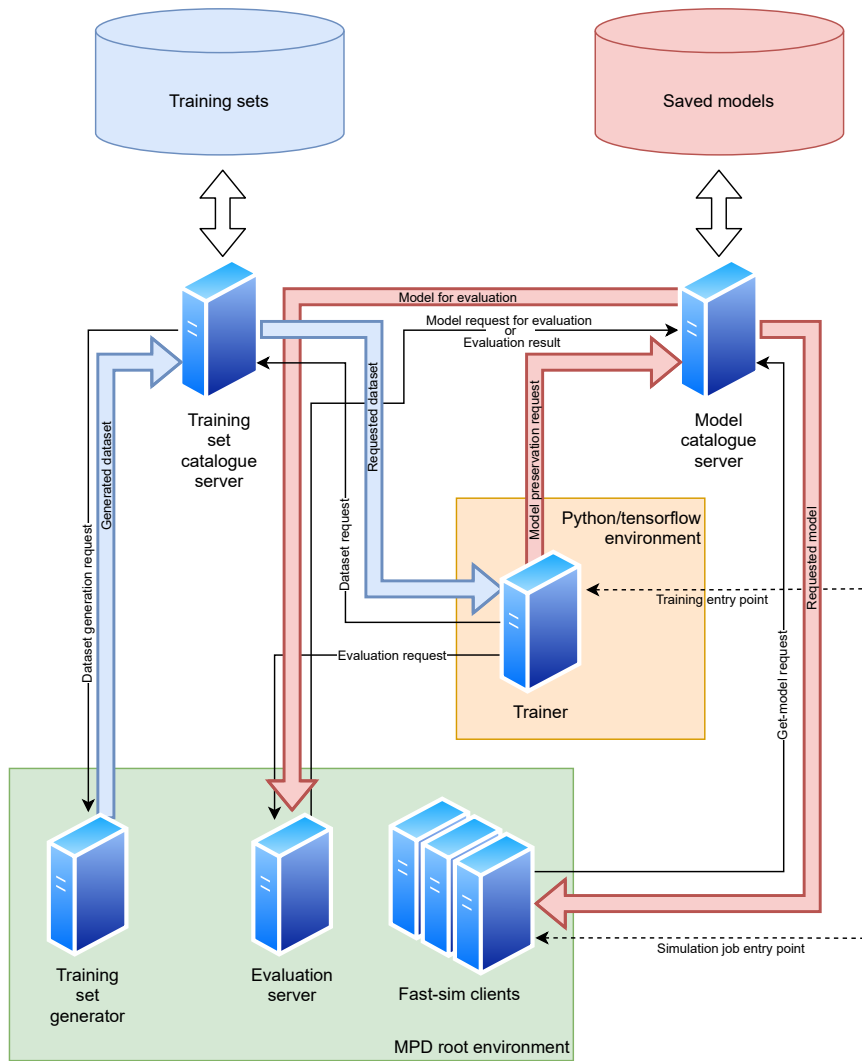


Использование модели в программном стеке

Модель тренируется на специальном оптимизированном сервере

Возможные способы загрузки:

- Модель компилируется в динамическую библиотеку
 - ✓ наиболее эффективный локальный код
 - Тяжелый файл, зависит от системы (GLib)
- Модель конвертируется в специальный формат (например ONNX) ▶ наш выбор
 - ✓ Универсальный загрузчик моделей
 - Дополнительная консервация, не всякая модель конвертируется, некоторый оверхед
- Модель обрабатывается на специальном сервере
 - ✓ наиболее эффективное исполнение модели на оптимизированном железе
 - Требуется глубокой параллелизации программного стека для эффективной работы в режиме больших матчей



Заключение

Модели машинного обучения эффективно решают многие задачи в рамках обработки и генерации данных физических экспериментов

Однако, модели машинного обучения зависят от условий и данных, использованных при обучении

Для эффективного использования недостаточно просто обучить модель; модели надо

- ▶ Каталогизировать и сложить в хранилище
- ▶ Валидировать
- ▶ (До)обучать, если в хранилище не нашлось нужной модели
- ▶ Переносить из хранилища в программный стек

Функциональность, общая для всех экспериментов

- ▶ Общее решение может быть эффективным