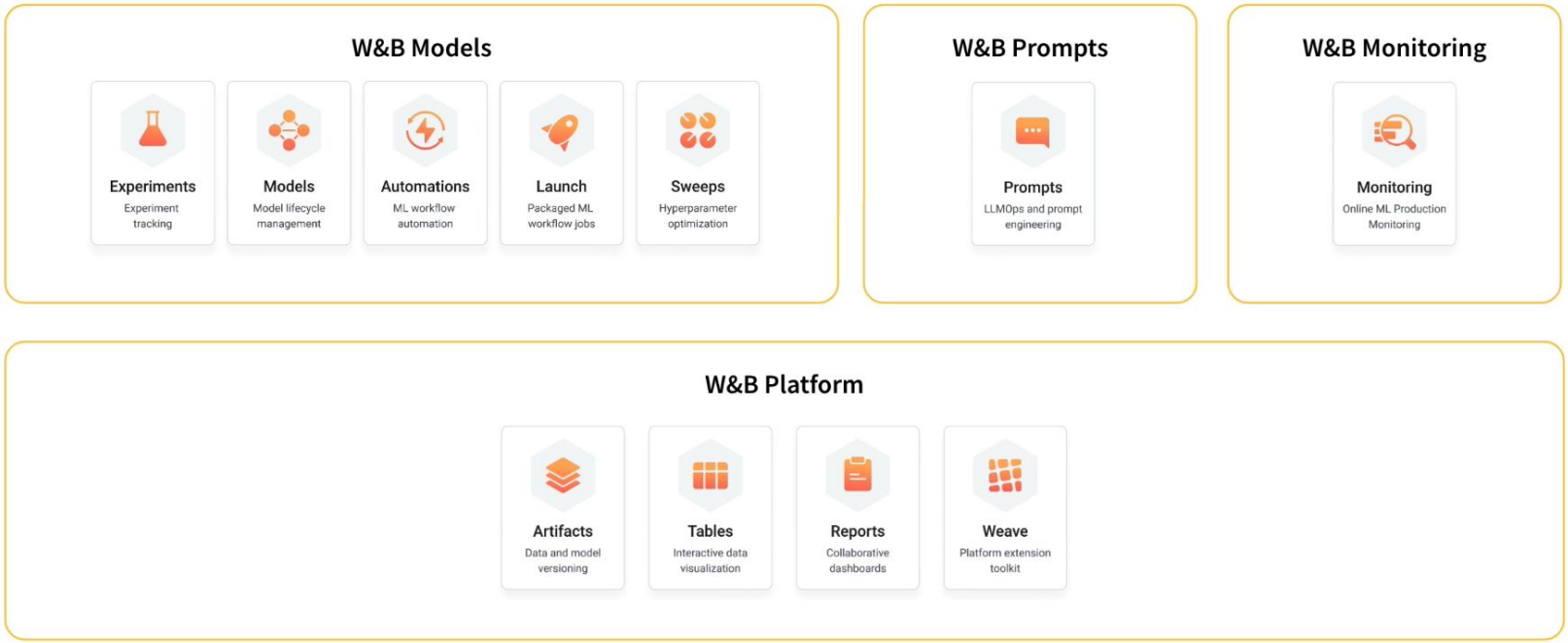
Evaluating and Debugging Generative AI using Weights and Biases

Weights and Biases MLOps Portfolio

Tools for Machine Learning Practitioner



Weights and Biases MLOps Platform

Integrated into every popular ML framework

Runs on every cloud or in your own infra

Why use Weights and Biases

Debugging and evaluating Generative AI

- Integrate quickly, track and version automatically
- Visualize your data and uncover critical insights
- Improve performance so you can evaluate and deploy with confidence

Instrumenting Weights and Biases

Integrate with any Python script

Installation & Import:

Install the `wandb` library using `pip` and import it into your script.

Hyperparameter Organization:

Define your hyperparameters in a `config` dictionary, such as `config = {'learning_rate': 0.001}`.

Start W&B Run:

Initialize a W&B run using `wandb.init()`, specifying a `project` name and passing your `config`.

Log Metrics:

During model training, log metrics like `loss` over time using `wandb.log({"loss": loss})` to visualize performance.

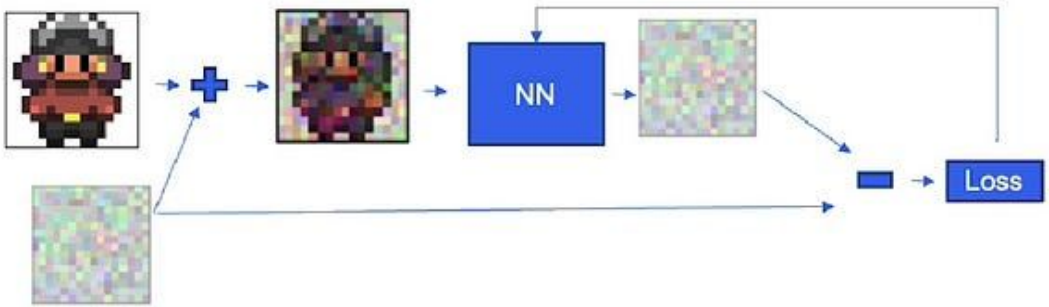
Finish Run (Notebooks):

If working in a Jupyter Notebook or similar environment, call `wandb.finish()` to explicitly end the W&B run.

Training a Diffusion Model

Tracking progress with Weights and Biases

- Neural network learns to predict noise–really learns the distribution of what is not noise
- Sample random timestep (noise level) per image to train more stably.

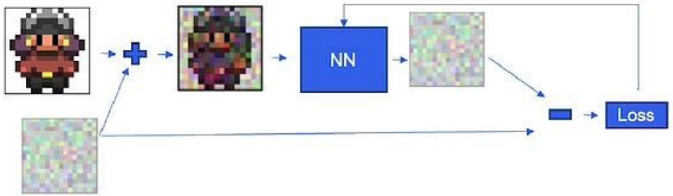


- A diffusion model learns how to iteratively remove small amounts of noise from an image

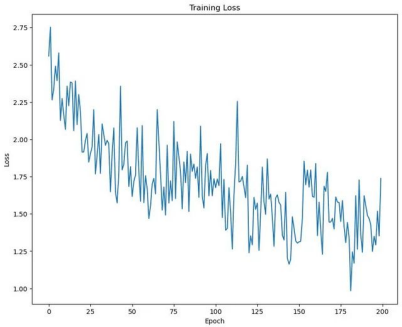
Training a Diffusion Model

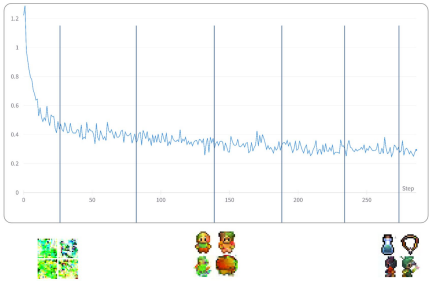
Tracking progress with Weights and Biases

- Neural network learns to predict noise–really learns the distribution of what is not noise
- Sample random timestep (noise level) per image to train more stably.



- A diffusion model learns how to iteratively remove small amounts of noise from an image
- Telemetry is very important when it comes to training generative models
- For the diffusion models, we can:
  - Keep track of the loss and relevant metrics
  - Sample images from the model during training
  - Safely store and version model checkpoints





Comparing Model Outputs

Managing Models

- Model registry: a central system of record of your models
  - Publish production-ready models
  - Move model versions through the lifecycle from staging to production
  - Collaborate on models across teams
  - Audit model lineage across training, evaluation, and production
  - Automate downstream actions

W&B Tables

- Log, query, and analyze tabular data including rich media: images, videos, molecules, etc
- Compare changes precisely across models

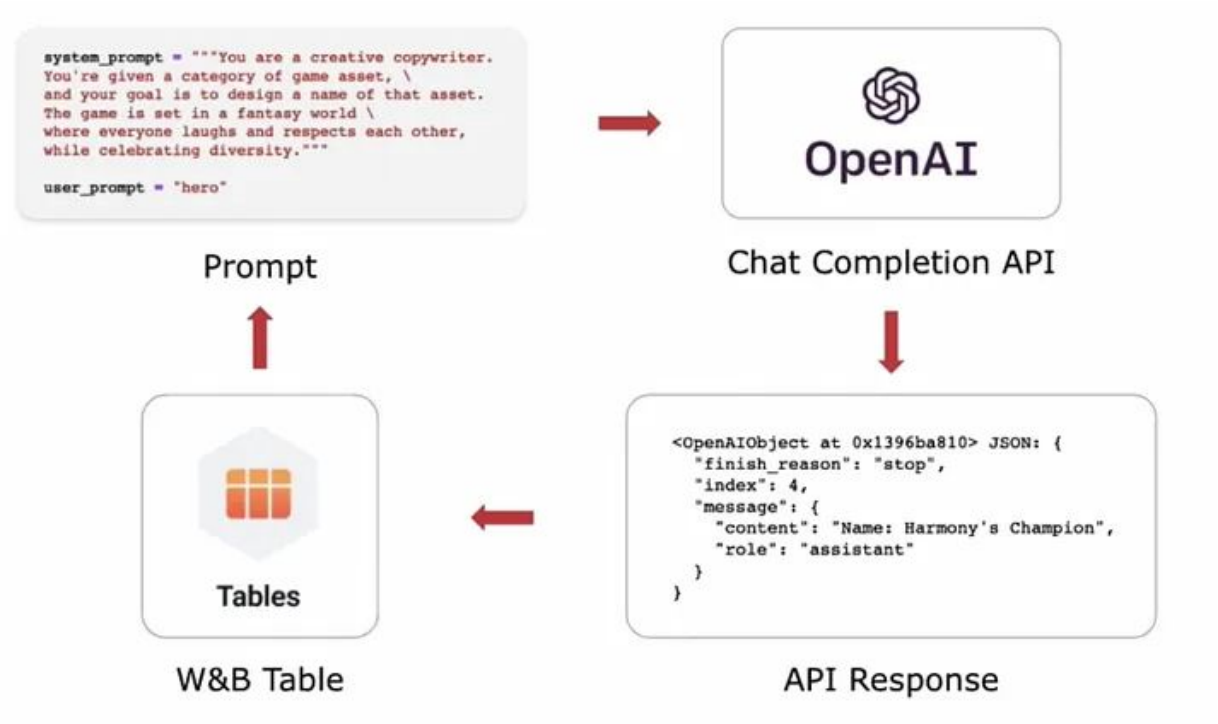
Evaluating LLMs

Using Weights and Biases Prompts

- Using APIs with Tables
- Tracking LLM chain spans with Tracer
- Tracking Langchain Agents

Evaluating LLMs

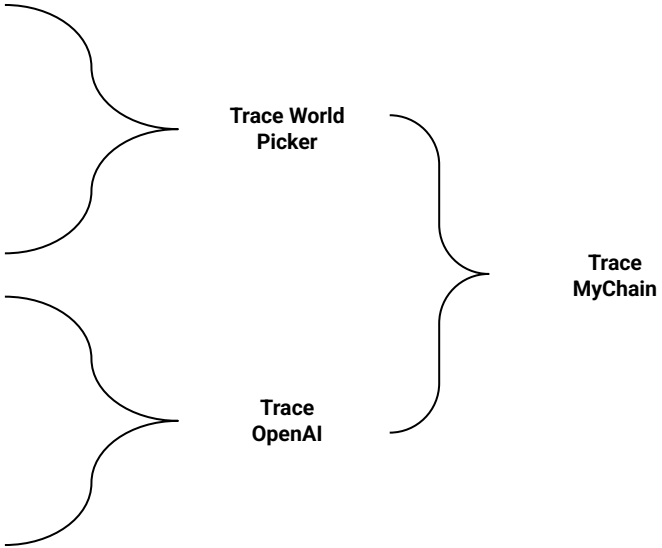
Calling OpenAI APIs



Evaluating LLMs

Tracking LLM chain spans with Tracer

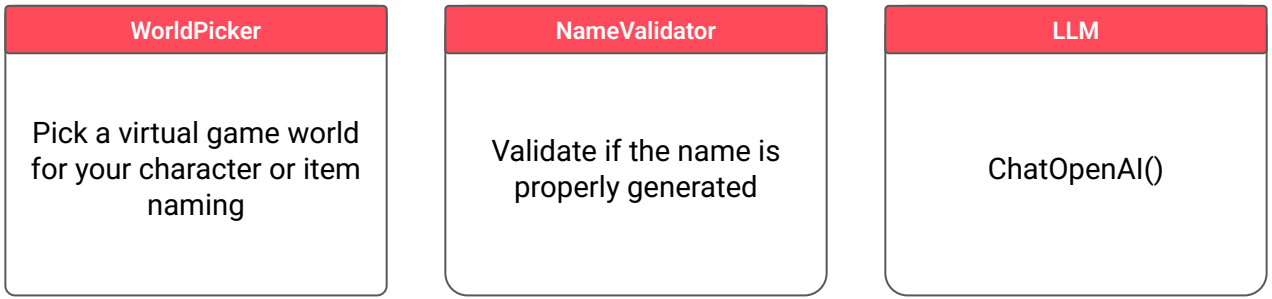
1. Pick a virtual world
    - input
    - output
    - start time
    - end time result
    - status
2. Generate description
    - input
    - output
    - start time
    - end time result
    - status



Evaluating LLMs

Tracking Langchain Agent

- ReAct Agent:
  - looping through reasoning (what should I do),
  - Actions (using tools),
  - Observations (what have I learned)



Training and Finetuning LLMs

Using Weights and Biases

- Training from scratch
  - Long and expensive training runs
  - Expensive and difficult evaluations
  - Monitoring is critical
  - Ability to restore training from a checkpoint
- Fine-tuning
  - Efficient methods being developed
  - Expensive and difficult evaluations

THANK YOU