Doing Good Research and Experiments!

Dual-Degree orientation for CVIT Group

https://tinyurl.com/2023CVIT-DDWorkshop

May 27th, 2023

Slides by Vladlen Kotlun (Intel)

Doing (Good) Research

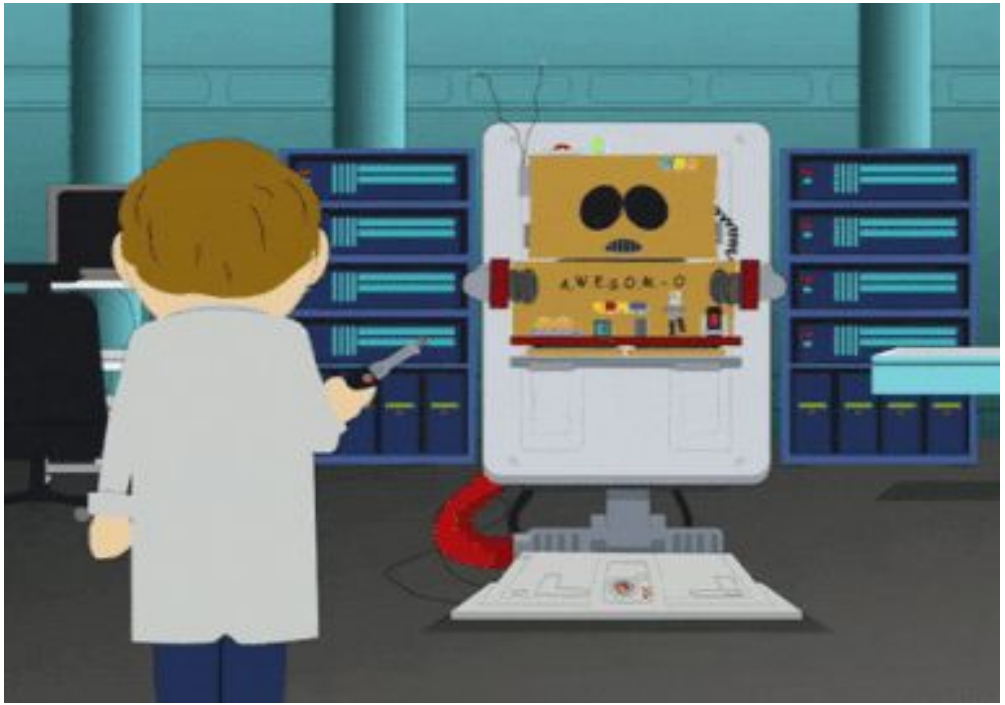
- Welcome to the CV community! Good Citizen workshop @ CVPR 2018
- What is Research?
- Ways to contribute
- Picking an area (work with your advisors on this)
- Publications
- On high standards
- Work ethic
- Ethics

Best Practices in ML

Link to the document

- Research Attitude
  - This is not coursework
  - Creating new information is not easy
  - Celebrate the small wins
  - Adopt the scientific method
- Never forget ML 101 (even when you are experienced)
  - When working with a new dataset
  - When working with a new model
  - Loss functions and plots
  - Closing the loop
- [check out the doc]

Experiments



Wonderful Twitter threads by Jia-Bin Huang

Doing research with your advisor / mentor / collaborators

- How to work with your advisor(s)?
- How to meet with your advisors/mentors?
- How to share your progress with your mentors/collaborators?
- How to do research with my mentors effectively?

Writing well

- How to write clear and concise sentences?
- How to write a paper that looks like a good one?
- How to create a good table?

How to Read a Paper

How to Read a Paper

- The first pass: Quick scan
- The second pass: Figures and Tables
- The third pass: Re-implement or understand link between code and math

In general

- Read critically, understand strengths and weaknesses (unstated assumptions)
- Think of what could be improved
- Learn not just the content, but also

Tracking experiments with WandB

B.Tech@IIIT-H orientation for CVIT Group

May 27th, 2023

Agenda

Introduce the audience to WandB and its key features, to explain the benefits of experiment tracking in machine learning, and to show you how to use WandB to improve your workflow.

- Introduction- experiment tracking
- What is WandB?
- Let's see some **code**
- - - - - - BREAK - - - - - - -
- Overview of WandB sweep
- Overview, applications and functionality in WandB website

Introduction

Experiment tracking and **WandB**.

What do we mean by experiment tracking.

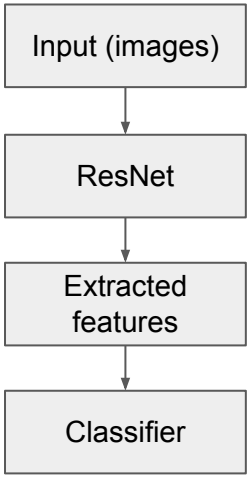
Improve your machine learning workflow

- Experiment tracking:
  Experiment tracking refers to the process of systematically logging and organizing experiments in order to better understand the outcomes of different machine learning models and algorithms.

- Importance of experiment tracking in machine learning
  Experiment tracking is critical for machine learning practitioners as it enables them to keep track of their work and reproduce their results. This is particularly important in complex projects where multiple people are working on different parts of the codebase.

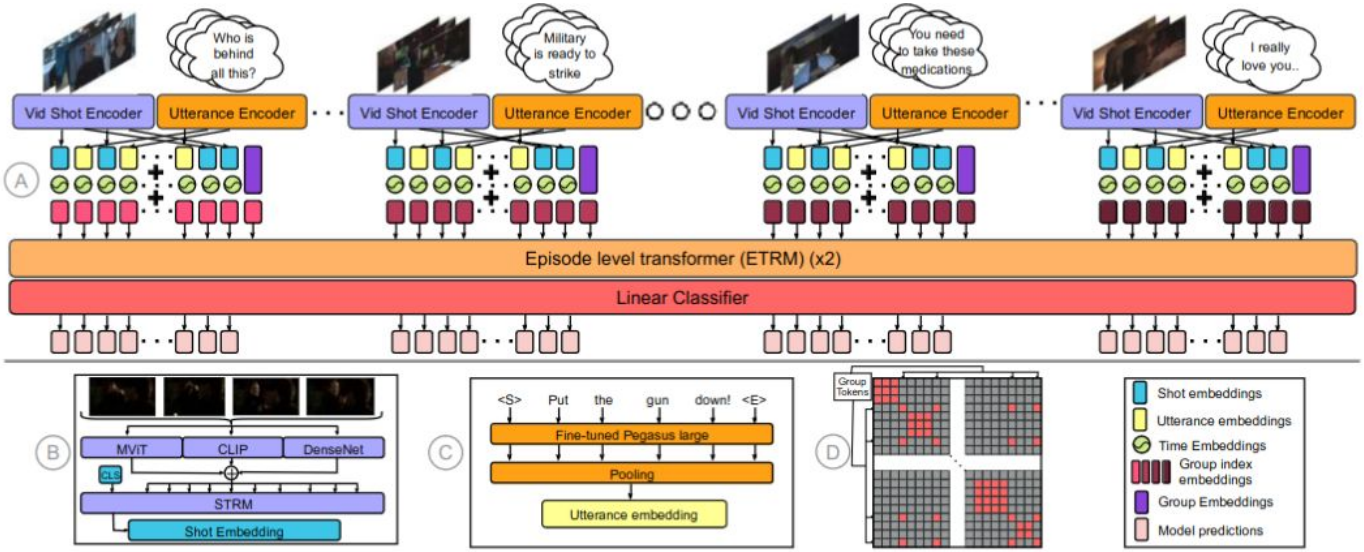
What do we mean by experiment tracking.

Projects grow and get complex over time.



What do we mean by experiment tracking.

Projects grow and get complex over time.



What do we mean by experiment tracking.

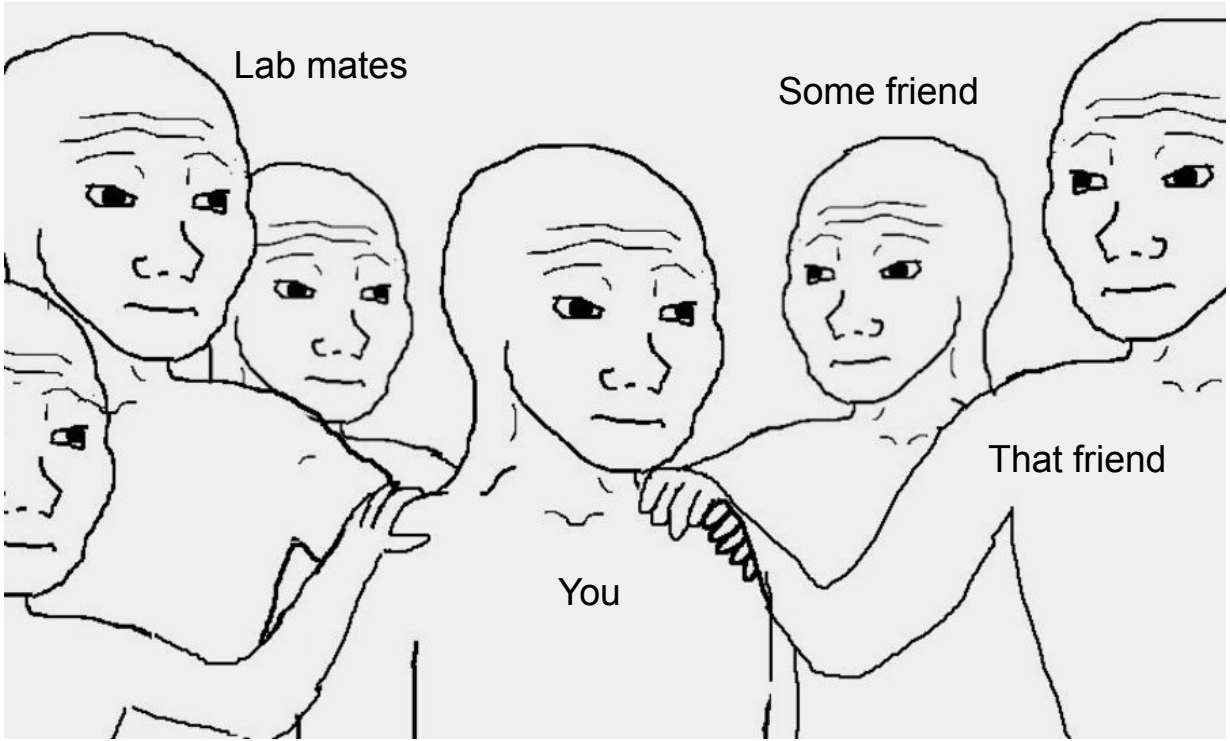
You may put a lot of efforts, those efforts should not be useless!

USEFUL

USELESS

What do we mean by experiment tracking.

Research work needs support



What do we mean by experiment tracking.

We will share one more friend with you today!

What do we mean by experiment tracking.

Improve your machine learning workflow

There is no fun in analysing this!

- {'epoch': 1, 'train_loss': 1.1452948740124702, 'eval_loss': 0.7713702845573426, 'train_ap_score': 0.6731415009678289, 'eval_ap_score': 0.8116030823185824, 'lr': 0.0001}
- {'epoch': 2, 'train_loss': 0.5890168227255345, 'eval_loss': 0.6724603283405304, 'train_ap_score': 0.8755422621356335, 'eval_ap_score': 0.8489012873912365, 'lr': 0.0001}
- {'epoch': 3, 'train_loss': 0.3640854485332966, 'eval_loss': 0.6902201867103577, 'train_ap_score': 0.9443318746126932, 'eval_ap_score': 0.8558312675174203, 'lr': 0.0001}
- {'epoch': 4, 'train_loss': 0.20835940316319465, 'eval_loss': 0.733428498506546, 'train_ap_score': 0.9796843642719427, 'eval_ap_score': 0.8559268354265821, 'lr': 0.0001}
- {'epoch': 5, 'train_loss': 0.1133101735264063, 'eval_loss': 0.8106619369983673, 'train_ap_score': 0.9937608794513503, 'eval_ap_score': 0.8549025780413813, 'lr': 0.0001}
- {'epoch': 6, 'train_loss': 0.06395852622576058, 'eval_loss': 0.872140085697174, 'train_ap_score': 0.9981744515435429, 'eval_ap_score': 0.856293925063724, 'lr': 0.0001}

What do we mean by experiment tracking.

Improve your machine learning workflow

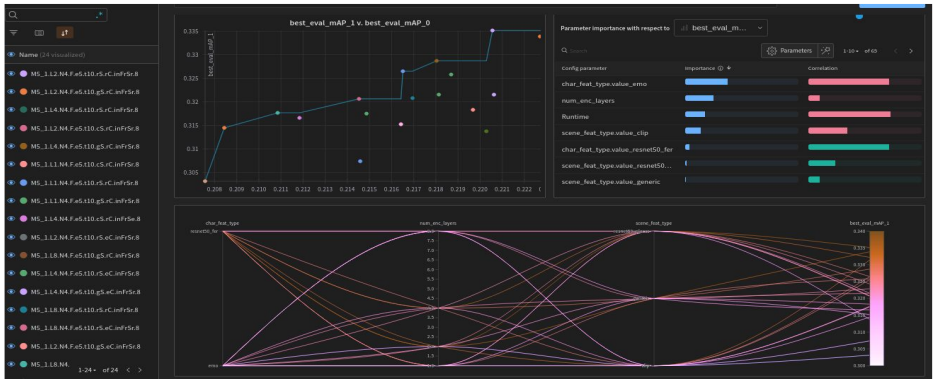
Without visualization/tracking, it is tough to answer queries like-

- How long does it take to run your experiments?
- Around what epoch does it start to overfit?
- When scheduler updated the learning rate, how much did that affect the metrics? Did it even trigger?!
- How to compare X different runs that only have change in one parameter? Which one to choose?
- Many more.....

Introduction to WandB

One place for all you experiments

- WandB is a powerful experiment tracking tool that helps machine learning practitioners to keep track of their models, datasets, and experiments. WandB offers a range of features including real-time visualization, hyperparameter tuning, and experiment comparison.
- Code → github | Exp. → WandB
- Easy to configure and use!



Lets jump to **code**

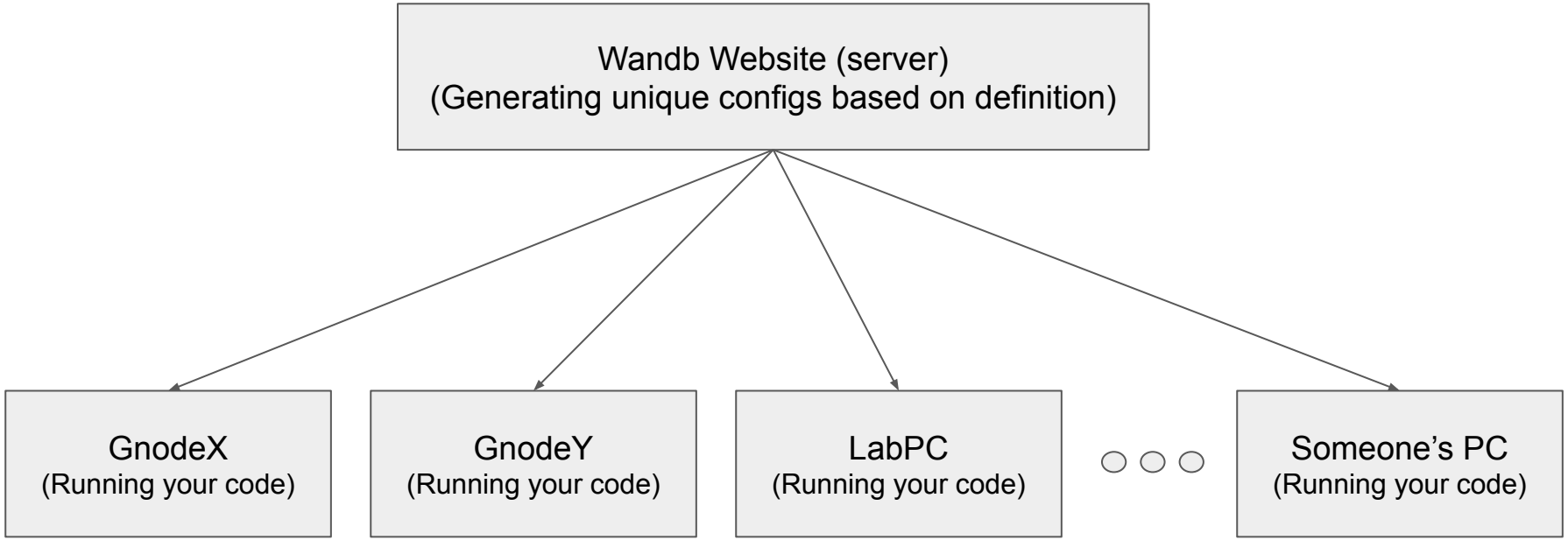
Log some configs and experiment data to **WandB**.

Head over to

https://tinyurl.com/KathaAiWandB

WandB sweeps

Overview



Break

Application of Sweeps

Generating configs automatically

- Efficient ablation management
- Grid Search
- Visualizing and Comparing multiple runs (of our choice) through parallel coordinates plot.
- We can engage as many gnodes (agents/workers) we want just by mentioning sweep-id.

WandB Functionality

UI elements from **COARSE** to **FINE** and how to effectively use them

Sections

- Charts
- Sweep (If sweeping is done)
- System
- Hidden Panels
- Miscellaneous
  (Smoothing, Outlier-handling, Layout Thingy)

Adding a custom panel:

- Line/Bar/Scatter/Parallel Plots
- Run Comparer
- Parameter Importance
- Code
- Markdown
- Custom chart
- Weave (to check/verify project essentials)

WandB RUNs

Chart View

- Edit panel (Data, Grouping, Chart, Legend, Expression)
- Miscellaneous (Export, Duplicate, Download)

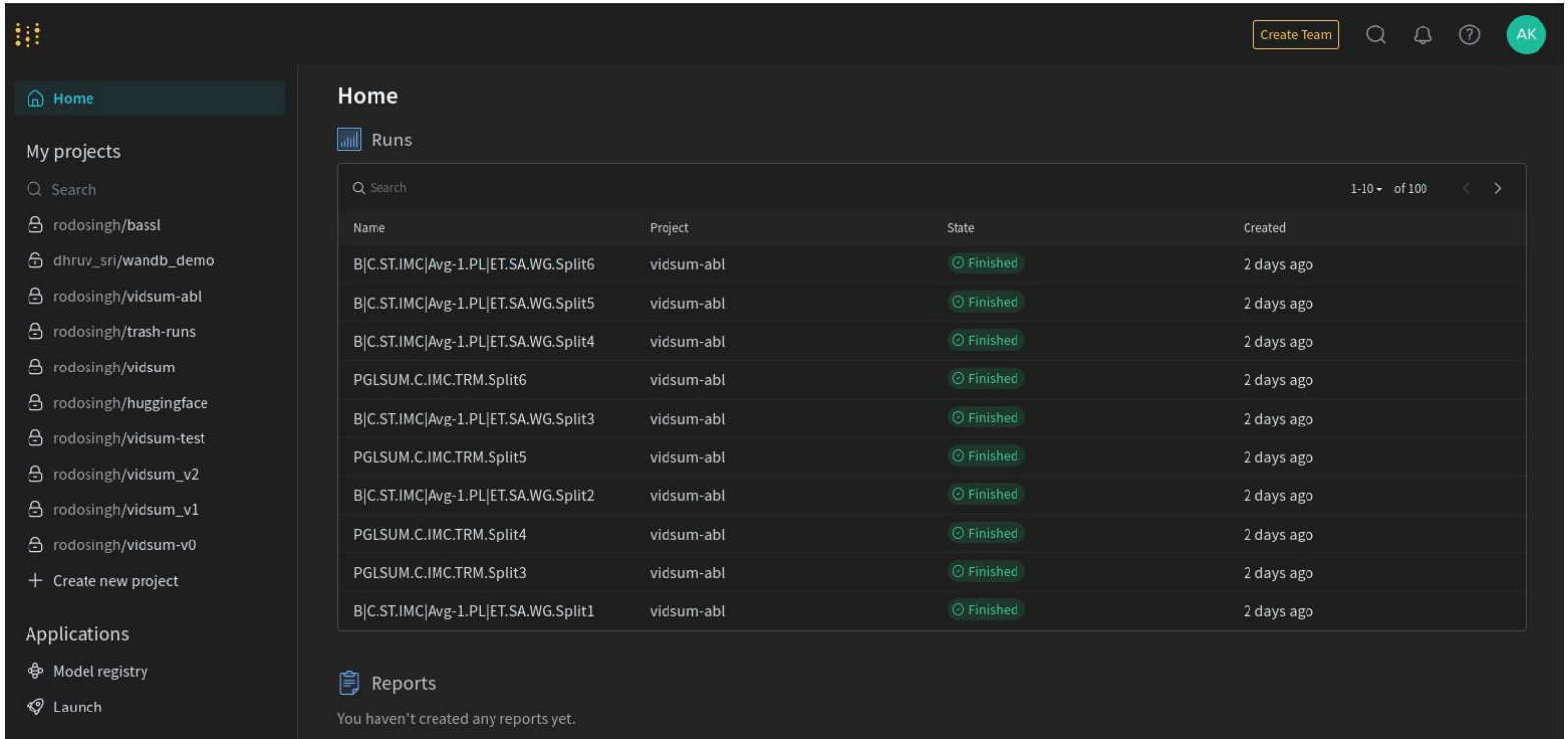
Table view

- Search / Filter / Group
- Sort
- Tag
- Move / Delete
- Magic wand (feature selection)
- Columns

WandB Functionality

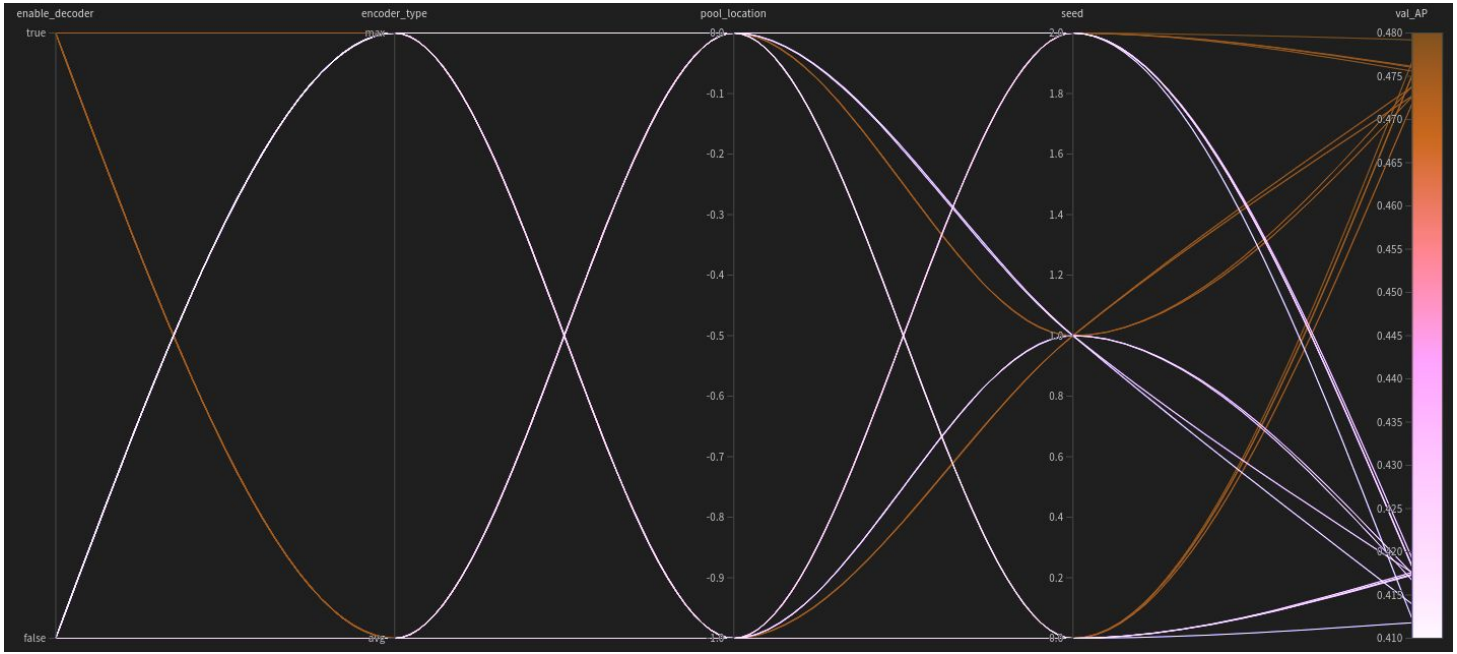
UI elements from **COARSE** to **FINE** and how to effectively use them

WandB Home - the GLOBAL view of all your experiments...



WandB Functionality

Parallel plots - **OUR SAVIOR**



WandB Reports - [Optional]

Parallel plots - **OUR SAVIOR**

Application of Sweeps (Revisited)

- Efficient ablation management: Hassle free auto execution of every combination of varying parameters.
- Grid Search: Can estimate feature importance (can be selected manually) based on metric objective.
- Visualizing and Comparing multiple runs (of our choice) through parallel coordinates plot.
- We can engage as many `gnodes` (agents/workers) we want just by mentioning `sweep-id` while submitting batch job for each `gnode`. This parallely completes all the runs expected in that `sweep`. No need to submit usual batch job (with all pre-requirements of data and code satisfied for that gnode).

- So it does have a lot of perks, Agree? 🤦 🤦 🤦

ACHIVE - 6 Perks of WandB

Advanced Features

Experiment Reproducibility

Hyperparameter Optimization

Visualization and Analysis

Integration and Compatibility

Collaboration and Sharing

Multithreading vs Multiprocessing

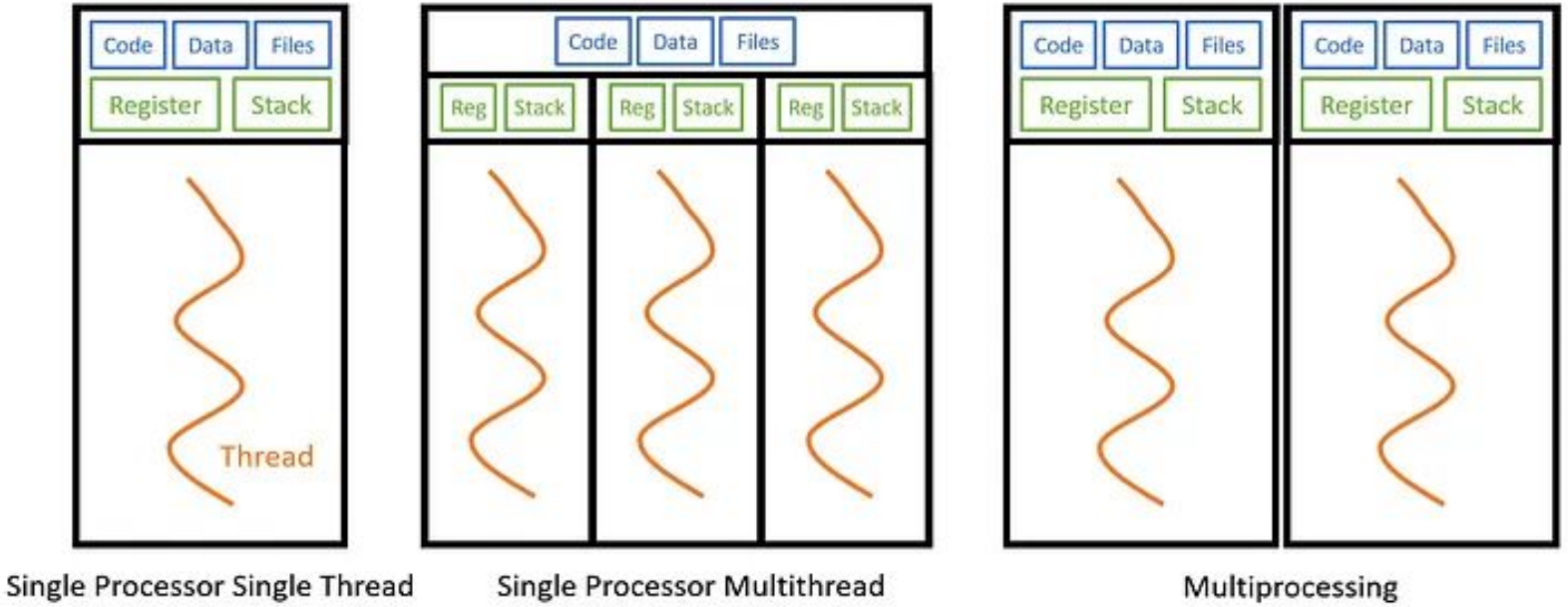


Image Credits: Silberschatz et al., "Operating System Concepts", Chapter 4

What are Vision Language models (VLMs)?

- VLMs are a type of neural network models that combines computer vision and natural language processing techniques to perform a variety of different multimodal tasks such as image captioning, visual question answering, and image retrieval.

- VLMs (like CLIP, Flamingo, BLIP, etc.) have been highly successful in recent years, with state-of-the-art results on various benchmark datasets for the above mentioned tasks.

What tasks can VLMs perform?

- Image Captioning
- Visual Question Answering (VQA)
- Visual Grounding
- Image-Text Matching
- Text to Image Retrieval
- Visual Scene Understanding

Image Captioning

A politician receives a gift from politician.

A collage of different colored ties on a white background.

Silhouette of a woman practicing yoga on the beach at sunset.

Aerial view of a road in autumn.

Image Credits: https://arxiv.org/abs/2111.09734 (ClipCap)

Transformer

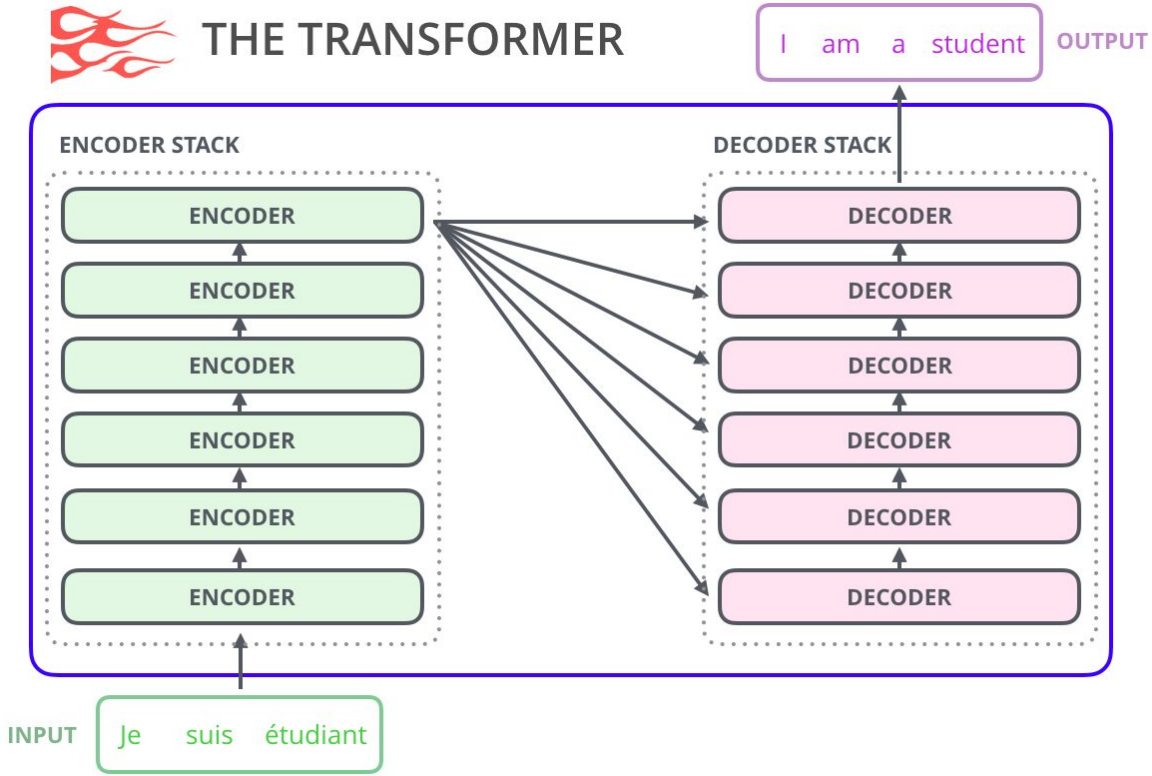


Image Credits: https://jalammar.github.io/illustrated-gpt2/

CLIP (Contrastive Language–Image Pre-training)

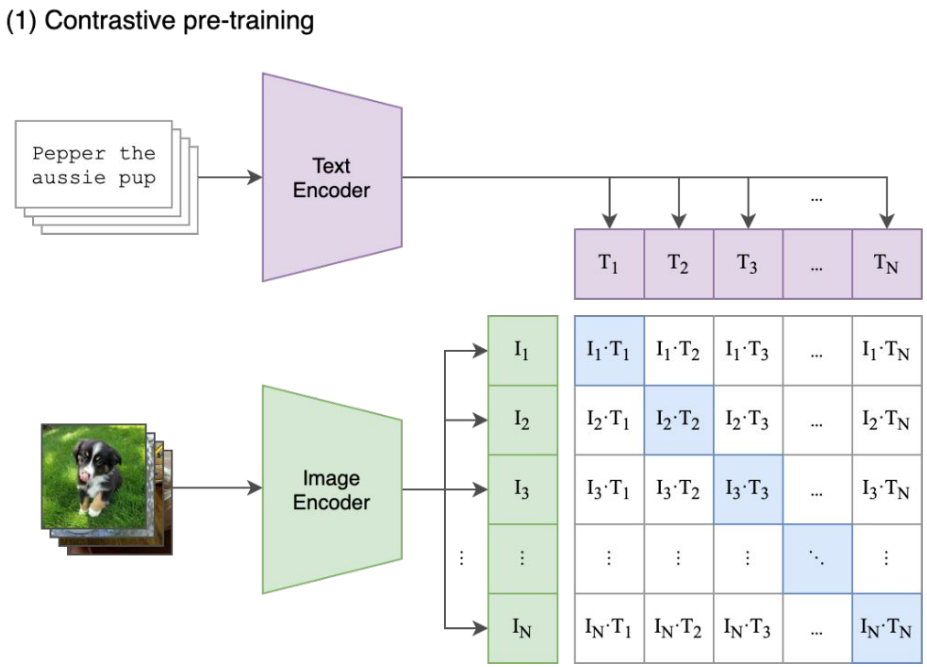


Image Credits: https://github.com/openai/CLIP

GPT-2

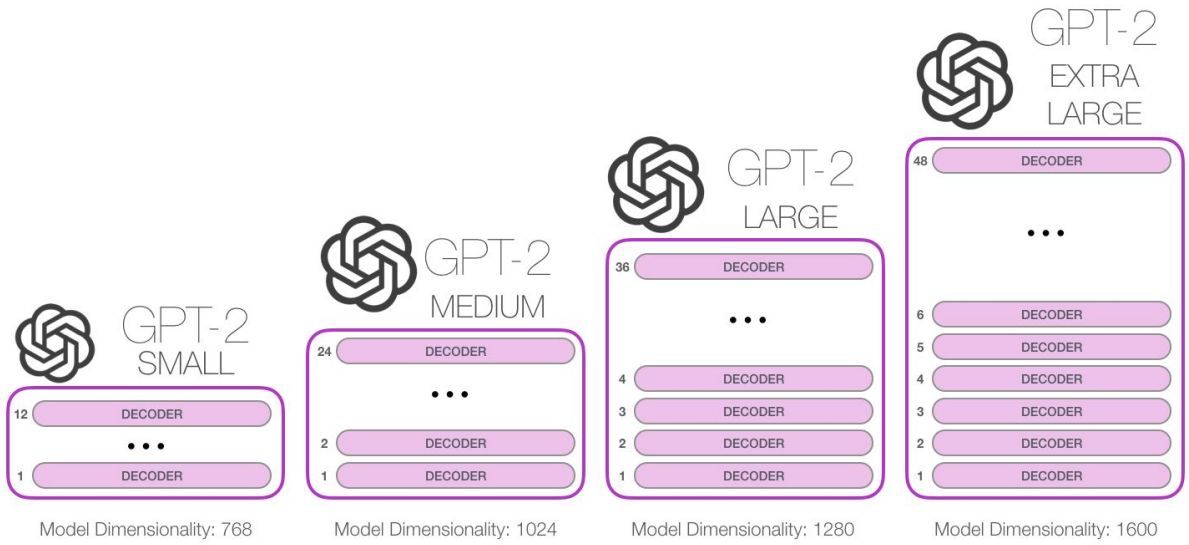
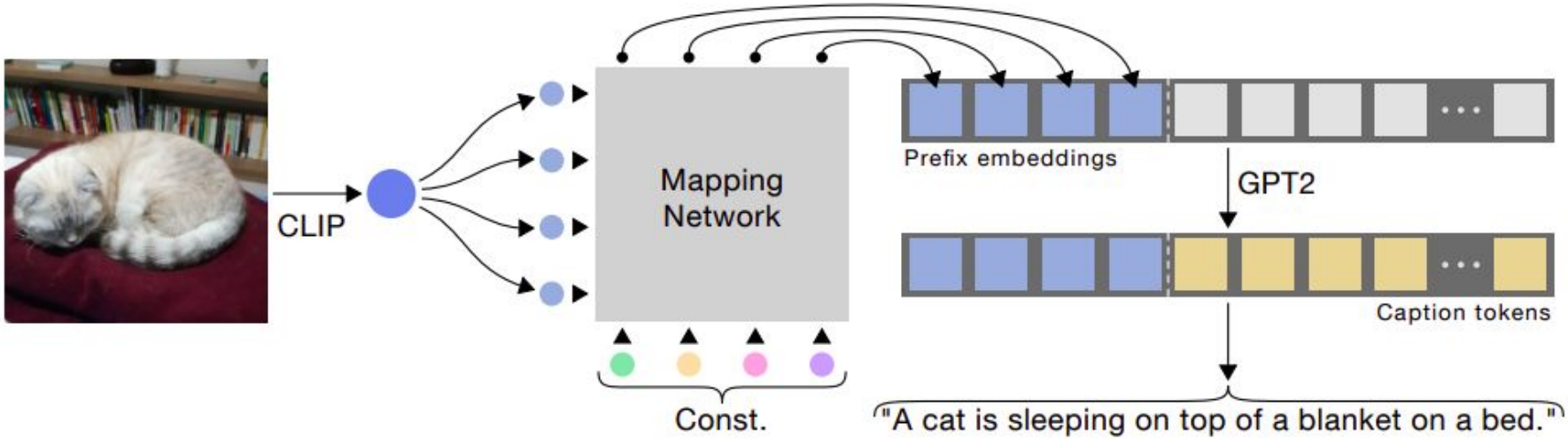


Image Credits: https://jalammar.github.io/illustrated-gpt2/

ClipCap: CLIP Prefix for Image Captioning[1]



[1] https://arxiv.org/abs/2111.09734

ClipCap: Objective Function[1]

Dataset: images and captions $\{x^i, c^i\}_{i=1}^N$

$$c^i = c_1^i, \ldots, c_\ell^i$$

$$p_1^i, \ldots, p_k^i = F(\text{CLIP}(x^i))$$

$$\mathcal{L}_X = -\sum_{i=1}^{N}\sum_{j=1}^{\ell} \log p_\theta(c_j^i | p_1^i, \ldots, p_k^i, c_1^i, \ldots, c_{j-1}^i)$$

[1] https://arxiv.org/abs/2111.09734

ClipCap[1]

Colab Notebook:

https://colab.research.google.com/drive/1IigZ0ZUABgcFWHKPmTgK6Sa_SVXB7hu3?usp=sharing

[1] https://arxiv.org/abs/2111.09734

Thank You

Python Debugger (pdb) Functionality

Debugging without print statements

- Debugging in Python is facilitated by pdb module (python debugger) which comes built-in to the Python standard library.
- Major advantage: Runs purely in the command line → great for debugging code on remote servers when we don't have the privilege of a GUI-based debugger.
- Features:
  - Setting breakpoints
  - Stepping through code
  - Source code listing
  - Viewing stack traces

Python Debugger (pdb) Functionality

Debugging without print statements

- To start debugging within the program just insert
  - `import pdb`
  - `pdb.set_trace()` (`breakpoint()`) commands at required.
  - Essential commands: [all you need is (`n, s, c, l, ll, q`)].
- `ipdb` - The interactive version of debugging where one can have following benefits:
  - `TAB` completion
  - Multiline code execution (e.g., for loop) with manual context-length control.
  - Syntax highlighting
  - Better tracebacks and introspection.

Credits: https://realpython.com/python-debugging-pdb/

Debugging with VSCode Remotely

VSCode Utilities

1. **Connecting to Remote Server**
2. **VSCode Python Debugger – debugpy**

Source:

1. https://code.visualstudio.com/docs/python/debugging
2. https://www.youtube.com/watch?v=R3smFr6W8jI