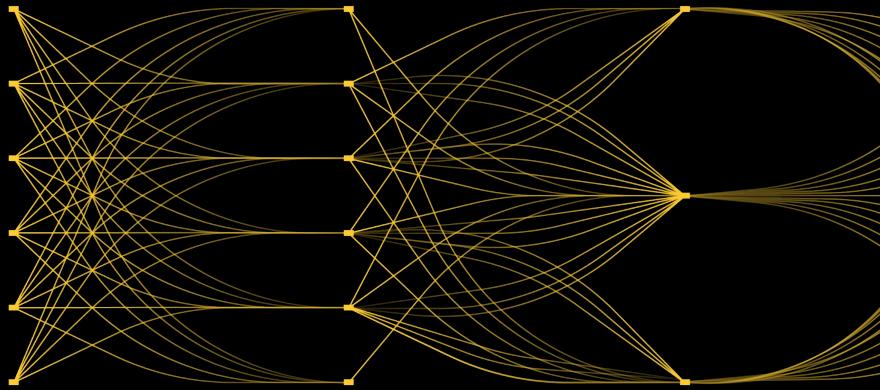




Illuminating ML Workflows

Thomas Capelle
Machine Learning Engineer, Weights & Biases





Principles of a **luminous** ML workflow



Observable

Surface metrics and predictions in real time within an accessible, visual interface



Reproducible

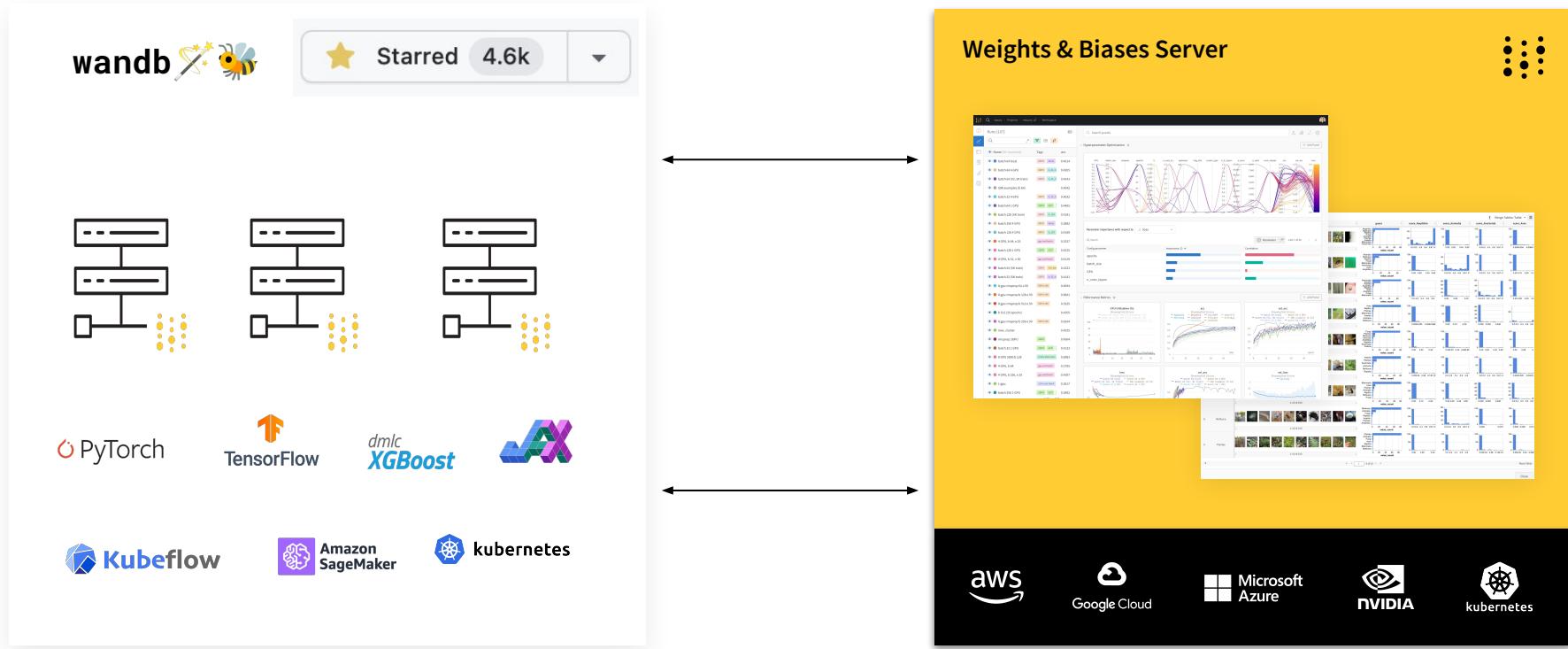
Capture the inputs, outputs, and intermediate data. Takes models to production responsibly and reduce key-person dependencies



Efficient Collaboration

Accelerate organizational knowledge transfer and reduce duplicative work

Weights & Biases





Get started in 3 steps

```
$ pip install wandb
```

→ Install wandb 

```
1 import wandb  
2 wandb.init()
```

→ Track with wandb 

```
3 for epoch in range(num_epochs):  
4     for idx, batch in enumerate(dataloader):  
5         # Model training...  
6         wandb.log(...)
```

→ Stream training metrics to
Weights & Biases

```
1 import wandb
2
2 wandb.init()
```



→ Track with wandb 

expert-sweep-14

What makes this run special? 

Benchmark 

Privacy 

Tags 

Author  bcanfieldsherman

State finished

Start time July 15th, 2022 at 11:21:50 am

Sweep 8bfumt2b

Duration 1h 58m 8s

Run path wandb-smle/vertex-super-resolution/kg50ne3e

Hostname cmle-training-3366323309258293058

OS Linux-5.10.107+x86_64-with-debian-bullseye-sid

Python version 3.7.12

Python executable /opt/conda/bin/python

Git repository git clone https://github.com/bcshermer/superres.git

Git state git checkout -b "expert-sweep-14" 08b2cc3e2a2e48e85c92bc8cf24664b841e2455f

Command train.py div2k:4x --log_grad=200 --lr=5e-05 --nblocks=16 --nlayers=4 --train_batch_size=32

System Hardware CPU count 4

GPU count 1

GPU type Tesla T4

W&B CLI Version 0.12.9

Job Type train

Config

Config parameters describe your model's inputs. [Learn more](#)

Key	Value
dataset	"div2k:4x"
dl_workers	4
growth_rate	64
log_grad	200
lr	0.00005
max_epochs	200
nblocks	16
nchannels	3
nfeatures	64
nlayers	4
patch_size	32
precision	32
save_code	false
scale_factor	4
train_batch_size	32
val_batch_size	4
val_im_size	128

GPU Power Usage (W)

— stoic-sweep-15



GPU Power Usage (%)

— stoic-sweep-15



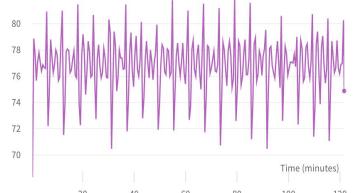
GPU Time Spent Accessing Memory (%)

— stoic-sweep-15



GPU Temp (°C)

— stoic-sweep-15



```
wandb.log(...)
```

metrics to Weights & Biases

Classes

`class Audio` : Wandb class for audio clips.

`class BoundingBoxes2D` : Format images with 2D bounding box overlays for logging to W&B.

`class Graph` : Wandb class for graphs

`class Histogram` : wandb class for histograms.

`class Html` : Wandb class for arbitrary html

`class Image` : Format images for logging to W&B.

`class ImageMask` : Format image masks or overlays for logging to W&B.

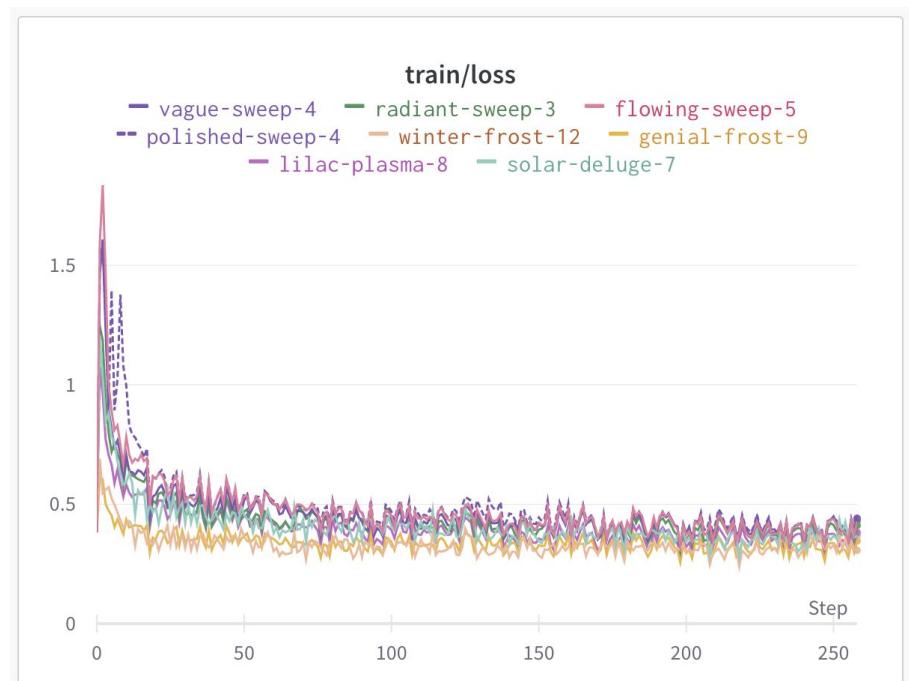
`class Molecule` : Wandb class for 3D Molecular data

`class Object3D` : Wandb class for 3D point clouds.

`class Plotly` : Wandb class for plotly plots.

`class Table` : The Table class is used to display and analyze tabular data.

`class Video` : Format a video for logging to W&B.



wandb.log(...)

audio to Weights & Biases

Classes

class `Audio`: Wandb class for audio clips.

```
class BoundingBoxes2D : Format images with 2D bounding box overlays for logging to W&B
```

class Graph : Wandb class for graphs

`class Histogram`: wandb class for histograms.

`class Html`: Wandb class for arbitrary html

`class Image` : Format images for logging to W&B.

class TImageMask : Format image masks or overlays for logging to

`class Molecule` : Wandb class for 3D Molecular data

`class Object3D`: Wandb class for 3D point clouds

class `Plotly`: Wandb class for plotly plots

`class Table`: The Table class is used to display and analyze tabular data.

[class_Video](#) : Format a video for logging to W&R

runs.summary["hifigan_predictions"]

B.   

	Text	Real validation audio	Audio Speaker 42
1	is yes, then you really do have a machine learning problem. And hopefully this got you	 00:03:00:06	 00:00:00:04
2	excited enough about all the applications of machine learning that you want to watch further	 00:03:00:05	 00:00:00:04
3	videos that explain actually how to build these models and how to deploy these models. And	 00:00:00:05	 00:02:00:04
4	we're going to keep creating these videos so you should probably subscribe so that you're the	 00:00:00:05	 00:00:00:03
5	first to know when a new video comes out.	 00:00:00:03	 00:00:00:02

    - 5 of 5 

wandb.log(...)

video to Weights & Biases

Classes

`class Audio`: Wandb class for audio clips.

`class BoundingBoxes2D`: Format images with 2D bounding box overlays for logging to W&B.

`class Graph`: Wandb class for graphs

`class Histogram`: wandb class for histograms.

`class Html`: Wandb class for arbitrary html

`class Image`: Format images for logging to W&B.

`class ImageMask`: Format image masks or overlays for logging to W&B.

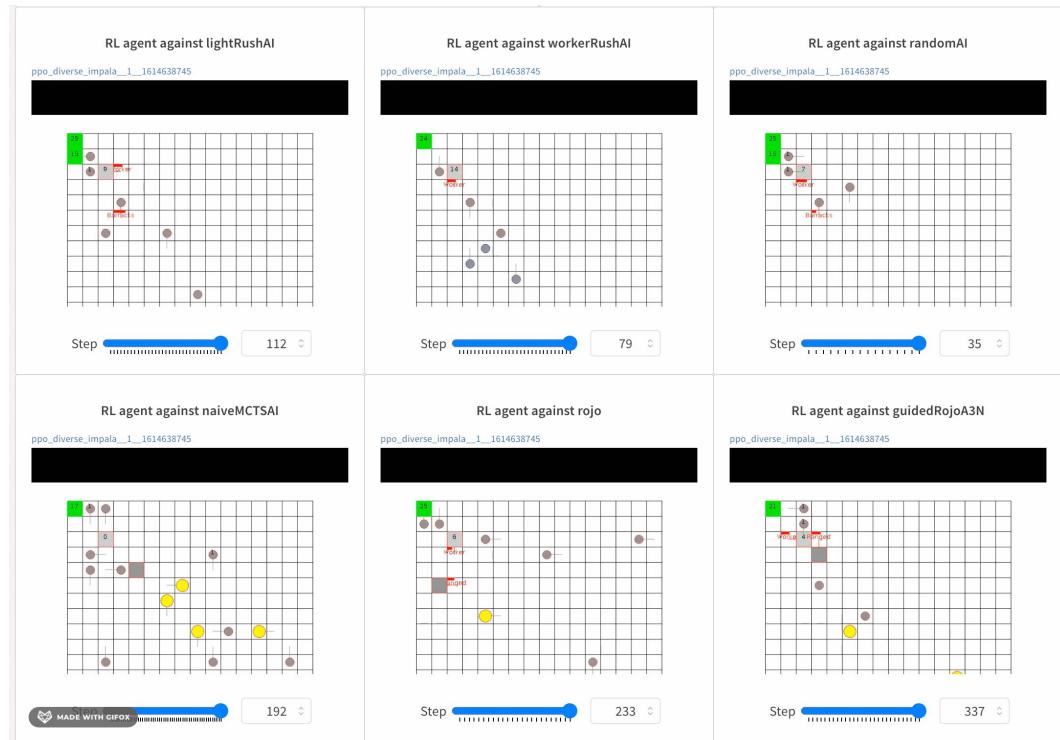
`class Molecule`: Wandb class for 3D Molecular data

`class Object3D`: Wandb class for 3D point clouds.

`class Plotly`: Wandb class for plotly plots.

`class Table`: The Table class is used to display and analyze tabular data.

`class Video`: Format a video for logging to W&B.



```
wandb.log(...)
```

annotated images to Weights & Biases

Classes

`class Audio`: Wandb class for audio clips.

`class BoundingBoxes2D`: Format images with 2D bounding box overlays for logging to W&B.

`class Graph`: Wandb class for graphs

`class Histogram`: wandb class for histograms.

`class Html`: Wandb class for arbitrary html

`class Image`: Format images for logging to W&B.

`class ImageMask`: Format image masks or overlays for logging to W&B.

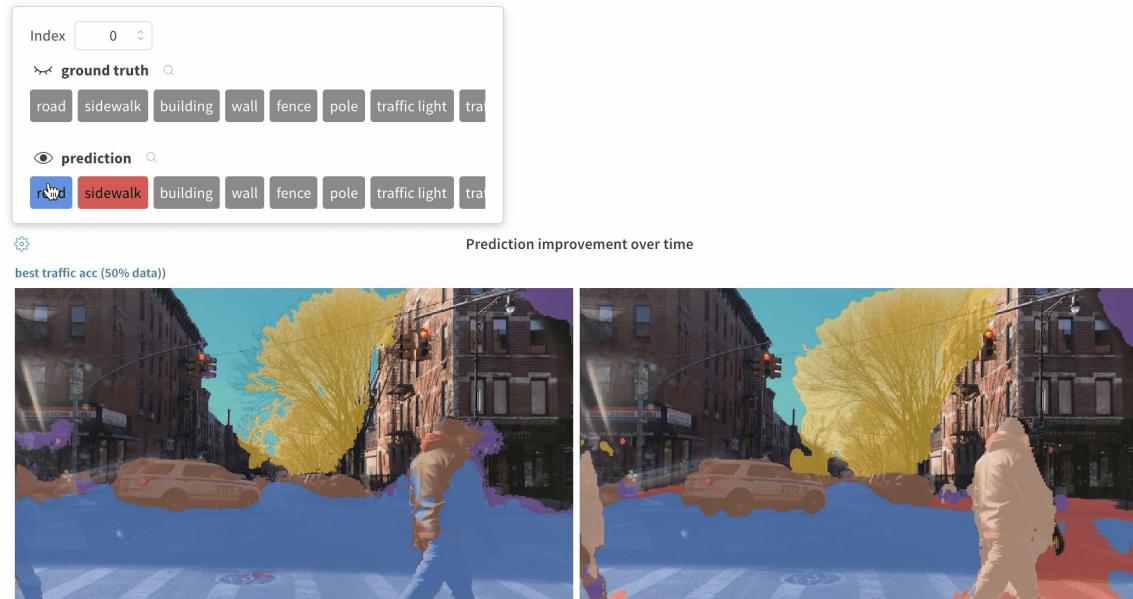
`class Molecule`: Wandb class for 3D Molecular data

`class Object3D`: Wandb class for 3D point clouds.

`class Plotly`: Wandb class for plotly plots.

`class Table`: The Table class is used to display and analyze tabular data.

`class Video`: Format a video for logging to W&B.



wandb.log(...)

custom visualizations to Weights & Biases

Classes

`class Audio`: Wandb class for audio clips.

`class BoundingBoxes2D`: Format images with 2D bounding box overlays for logging to W&B.

`class Graph`: Wandb class for graphs

`class Histogram`: wandb class for histograms.

`class Html`: Wandb class for arbitrary html

`class Image`: Format images for logging to W&B.

`class ImageMask`: Format image masks or overlays for logging to W&B.

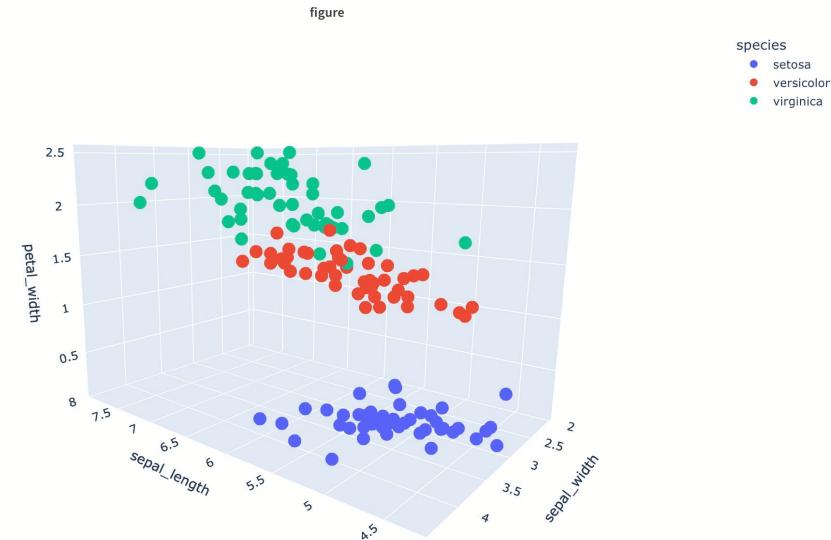
`class Molecule`: Wandb class for 3D Molecular data

`class Object3D`: Wandb class for 3D point clouds.

`class Plotly`: Wandb class for plotly plots.

`class Table`: The Table class is used to display and analyze tabular data.

`class Video`: Format a video for logging to W&B.



wandb.log(...)

tables to Weights & Biases

Classes

`class Audio`: Wandb class for audio clips.

`class BoundingBoxes2D`: Format images with 2D bounding box overlays for logging to W&B.

`class Graph`: Wandb class for graphs

`class Histogram`: wandb class for histograms.

`class Html`: Wandb class for arbitrary html

`class Image`: Format images for logging to W&B.

`class ImageMask`: Format image masks or overlays for logging to W&B.

`class Molecule`: Wandb class for 3D Molecular data

`class Object3D`: Wandb class for 3D point clouds.

`class Plotly`: Wandb class for plotly plots.

`class Table`: The Table class is used to display and analyze tabular data.

`class Video`: Format a video for logging to W&B.

runs.summary["predictions"]											
	image	label	prediction	score_0	score_1	score_2	score_3	score_4	score_5	score_6	score
8		9	5	0.00005464	6.304e-8	0.00003019	0.08346	0.07822	0.6602	1.582e-8	0.00
9		5	6	1.579e-12	1.915e-12	1.378e-10	5.420e-17	6.871e-11	0.4641	0.5359	1.46
19		3	5	0.0005906	7.744e-15	4.142e-12	0.06027	0.00001118	0.9234	9.153e-7	3.21
96		4	9	0.0001233	1.037e-9	0.000117	0.00007159	0.3716	0.0002828	0.000004016	0.00
120		2	8	0.001332	0.04431	0.2517	0.02707	0.0004833	0.02105	0.00279	0.00
125		7	4	0	9.932e-18	9.808e-31	1.338e-30	0.6915	6.017e-31	0	0.31
150		2	5	0.0007864	0.0003291	0.3655	7.815e-7	4.728e-9	0.6316	0.00008617	0.00
211		4	7	7.492e-23	1.320e-32	0.000001329	9.893e-22	0.1411	0.0002506	7.616e-33	0.01



Weights & Biases



NVIDIA/**NeMo**

NeMo: a toolkit for conversational AI



193

Contributors



31

Issues



300

Discussions



5k

Stars



1k

Forks





Nemo is already integrated with W&B! 😎

Built on top of PyTorch and pytorch_lightning weights and biases is already in there...

```
1 exp_manager.create_wandb_logger=True  
2 exp_manager.wandb_logger_kwargs={"project":"tts_project", "log_model":True}
```

→ Create the logger 

That's it!



If you want to learn more

- wandb.me/course
- wandb.me/discord

Explore the Weights & Biases platform



Experiments

Experiment tracking



Reports

Collaborative dashboards



Artifacts

Dataset and model versioning



Tables

Interactive data visualization



Sweeps

Hyperparameter optimization

Trusted by 200,000+ ML practitioners



OpenAI



NVIDIA



lyft



BMW
GROUP



FUJIFILM
SonoSite



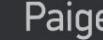
Truebill



ClearCapital



NURO



Paige



iRobot