Montreal, July 8–12

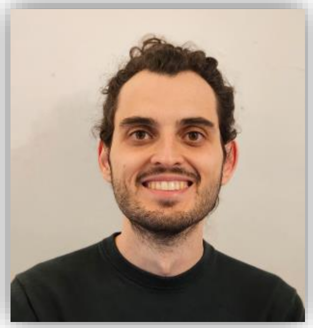
DLMI2024

Deep learning for medical imaging school

Hands-on session

Foundation Models

Julio
Silva-Rodríguez



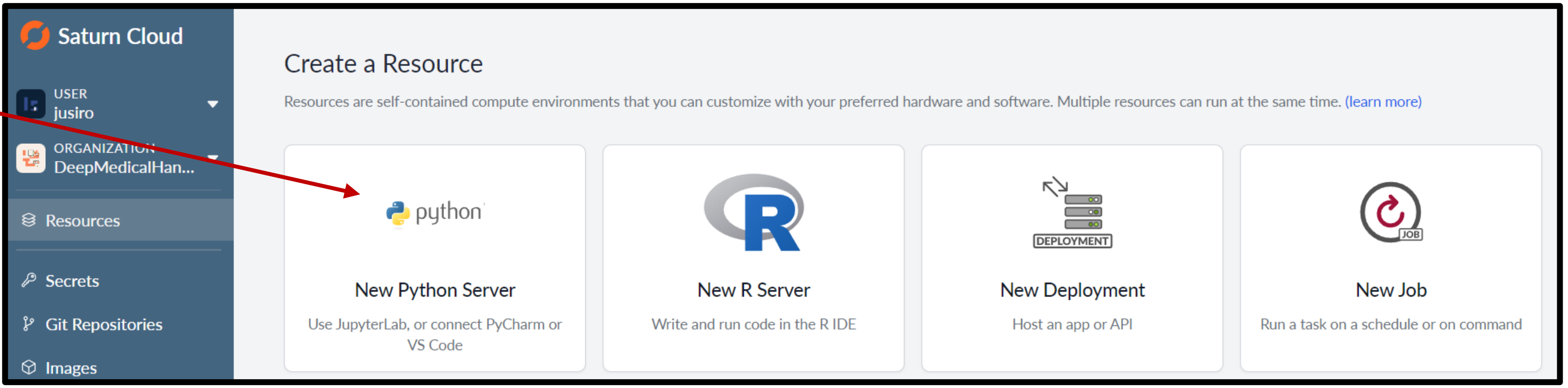
Jose Dolz

Setup your

SaturnCloud

enviroment

1. Click to “New Python Server”



Saturn Cloud

USER
jusiro

ORGANIZATION
DeepMedicalHan...

Resources

Secrets

Git Repositories

Images

Create a Resource

Resources are self-contained compute environments that you can customize with your preferred hardware and software. Multiple resources can run at the same time. (learn more)

python

New Python Server

Use JupyterLab, or connect PyCharm or VS Code

New R Server

Write and run code in the R IDE

DEPLOYMENT

New Deployment

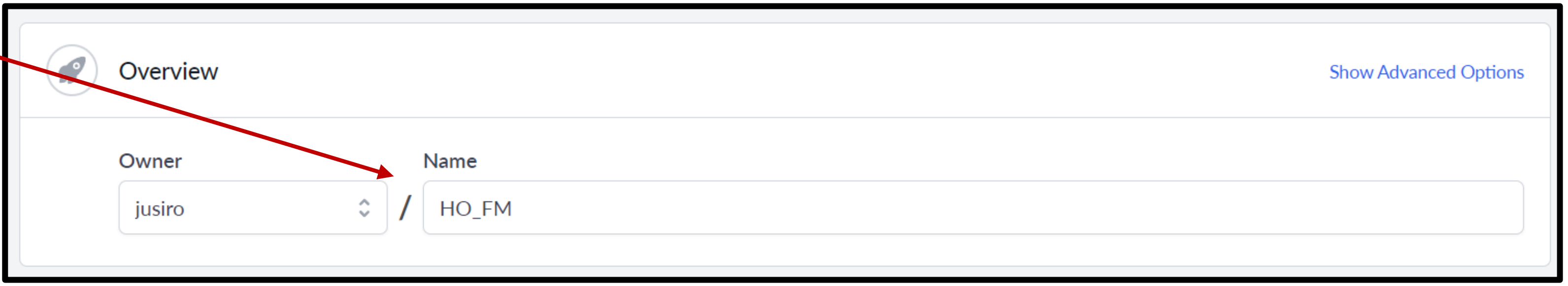
Host an app or API

JOB

New Job

Run a task on a schedule or on command

2. Add Name



Overview

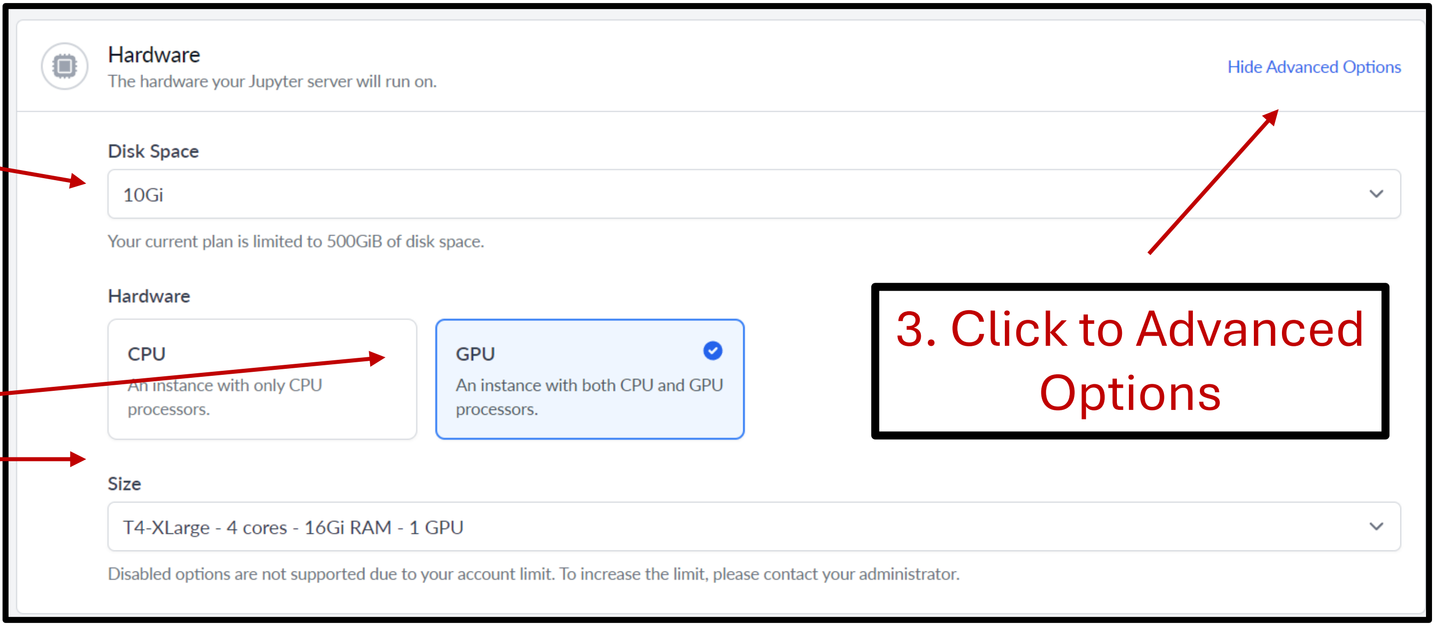
Show Advanced Options

Owner: jusiro

Name: HO_FM

4. 10Gb of Disk Space

5. Select GPU Hardware



Hardware

The hardware your Jupyter server will run on.

Hide Advanced Options

Disk Space: 10Gi

Your current plan is limited to 500GiB of disk space.

Hardware

○ CPU
An instance with only CPU processors.

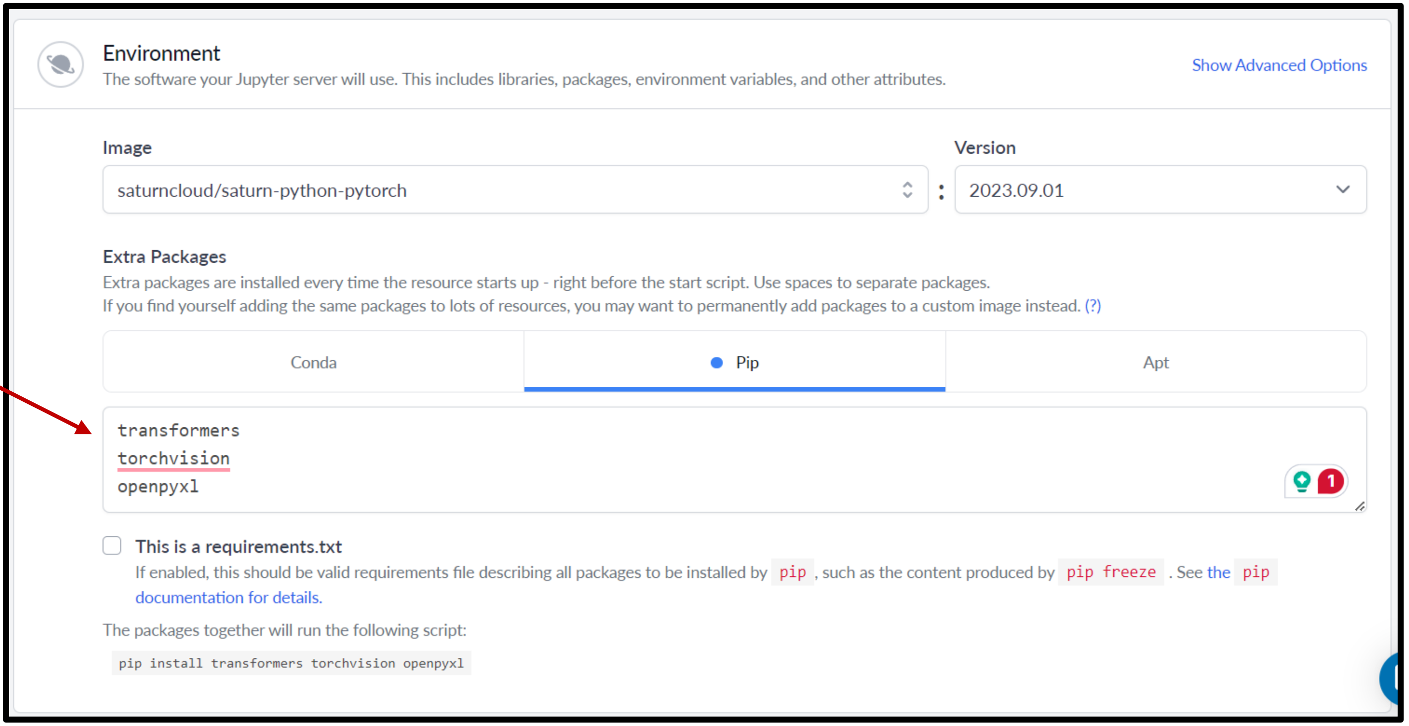
● GPU
An instance with both CPU and GPU processors.

Size: T4-XLarge - 4 cores - 16Gi RAM - 1 GPU

Disabled options are not supported due to your account limit. To increase the limit, please contact your administrator.

3. Click to Advanced Options

6. Add libraries using pip



Environment

The software your Jupyter server will use. This includes libraries, packages, environment variables, and other attributes.

Image: saturncloud/saturn-python-pytorch

Version: 2023.09.01

Extra Packages

Extra packages are installed every time the resource starts up - right before the start script. Use spaces to separate packages.
If you find yourself adding the same packages to lots of resources, you may want to permanently add packages to a custom image instead. (?)

```
transformers
torchvision
openpyxl
```

□ This is a requirements.txt

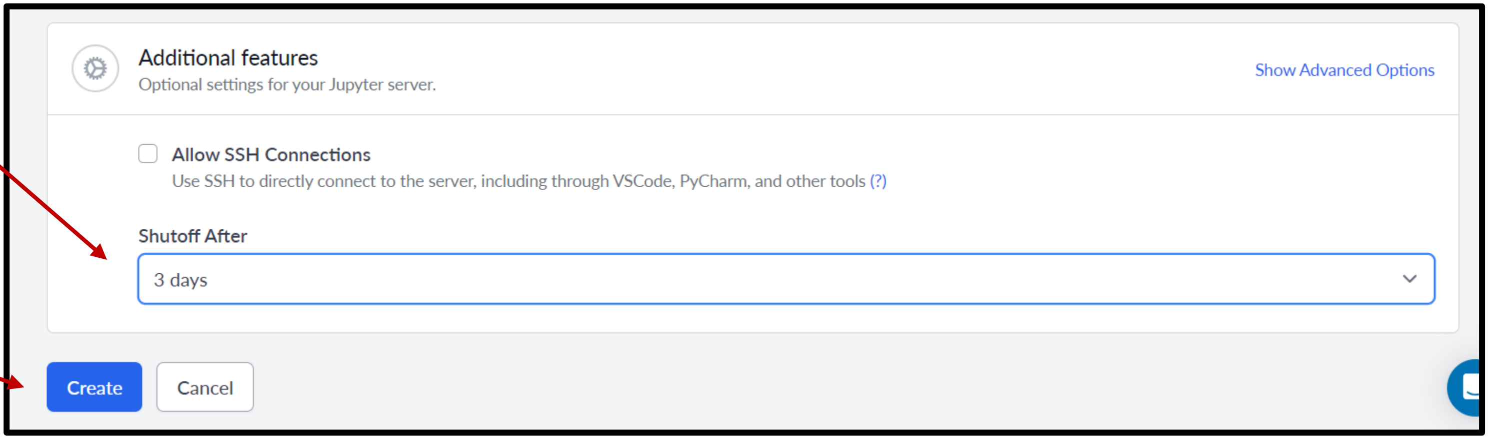
If enabled, this should be valid requirements file describing all packages to be installed by `pip`, such as the content produced by `pip freeze`. See the `pip` documentation for details.

The packages together will run the following script:

```
pip install transformers torchvision openpyxl
```

7. Select duration of project

8. Create



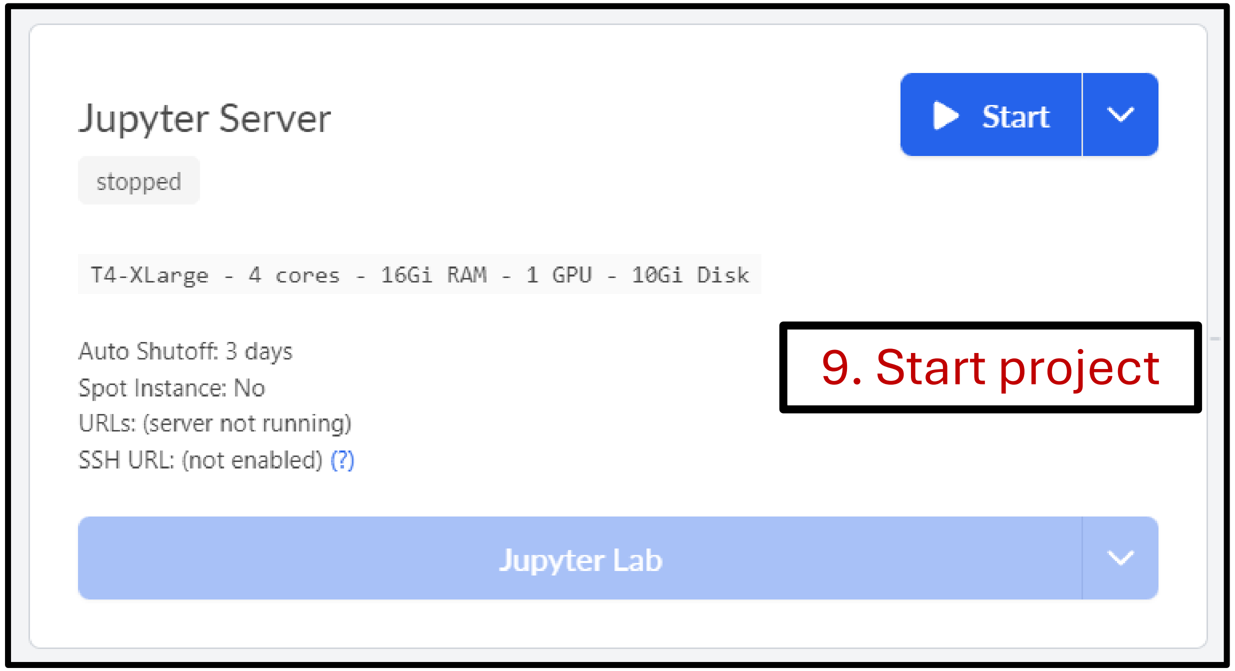
Additional features

Optional settings for your Jupyter server.

□ Allow SSH Connections

Use SSH to directly connect to the server, including through VSCode, PyCharm, and other tools (?)

Shutoff After: 3 days



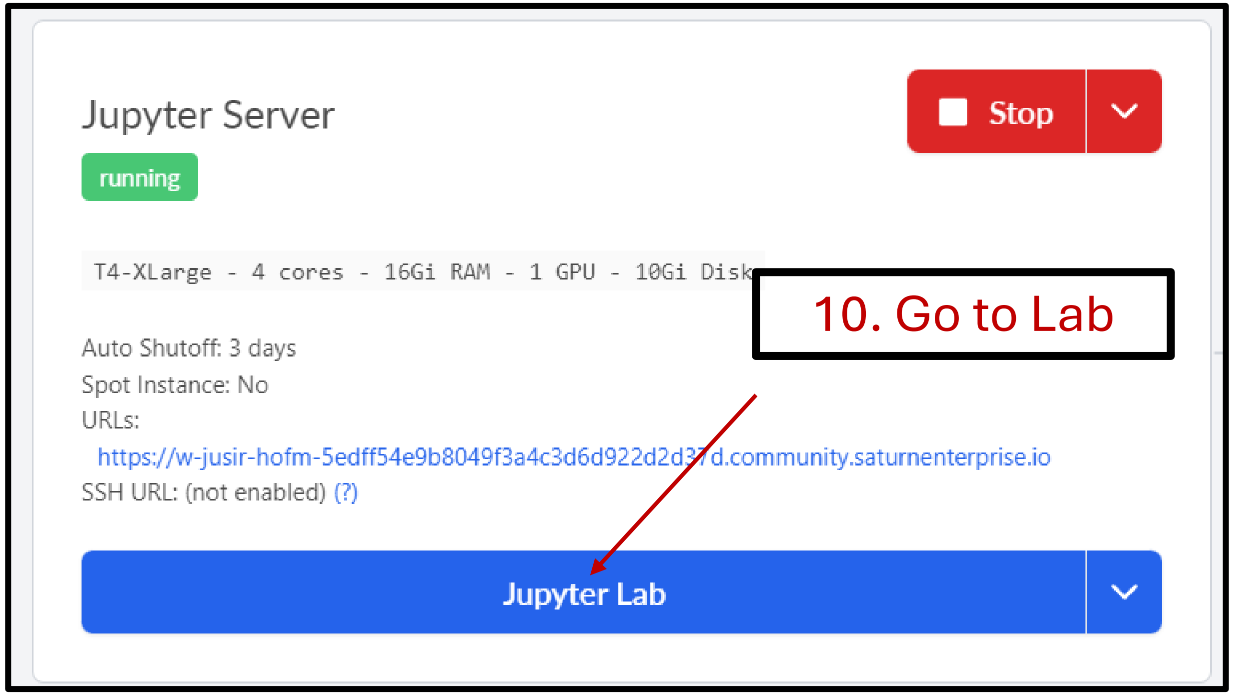
Jupyter Server

stopped

```
T4-XLarge - 4 cores - 16Gi RAM - 1 GPU - 10Gi Disk
```

Auto Shutoff: 3 days
Spot Instance: No
URLs: (server not running)
SSH URL: (not enabled) (?)

9. Start project



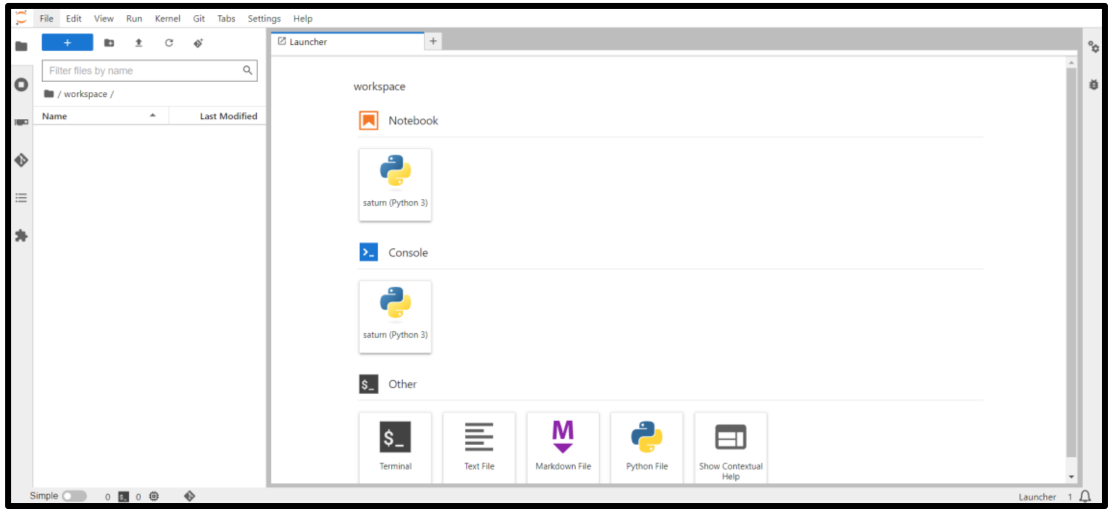
Jupyter Server

running

```
T4-XLarge - 4 cores - 16Gi RAM - 1 GPU - 10Gi Disk
```

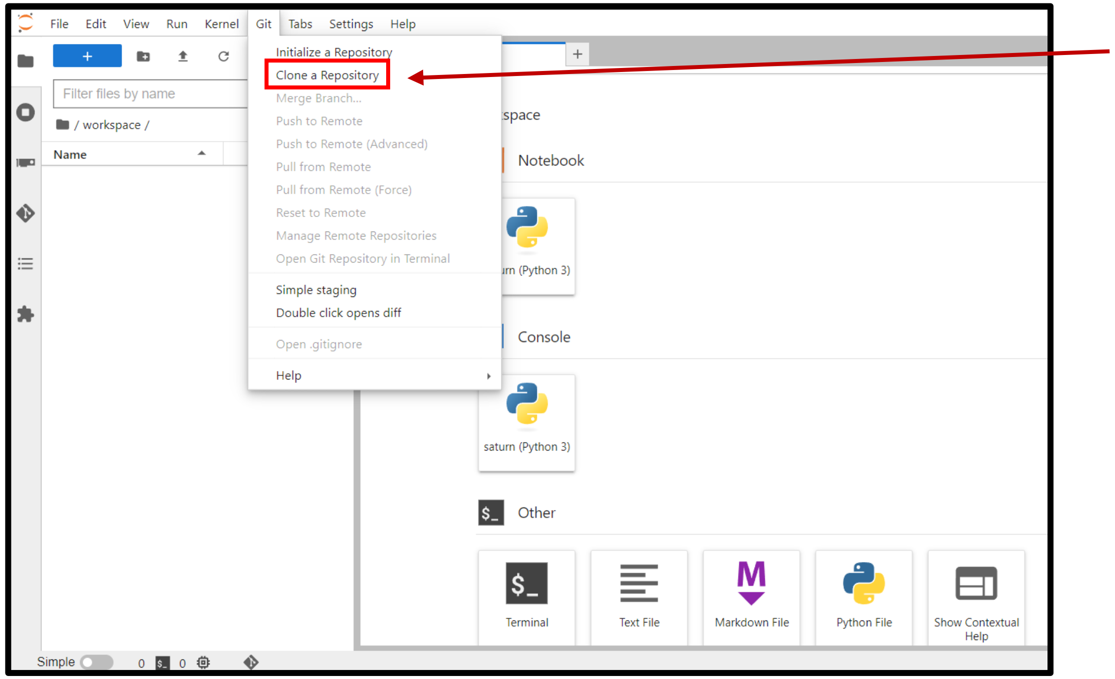
Auto Shutoff: 3 days
Spot Instance: No
URLs:
https://w-jusir-hofm-5edff54e9b8049f3a4c3d6d922d2d37d.community.saturnenterprise.io
SSH URL: (not enabled) (?)

10. Go to Lab

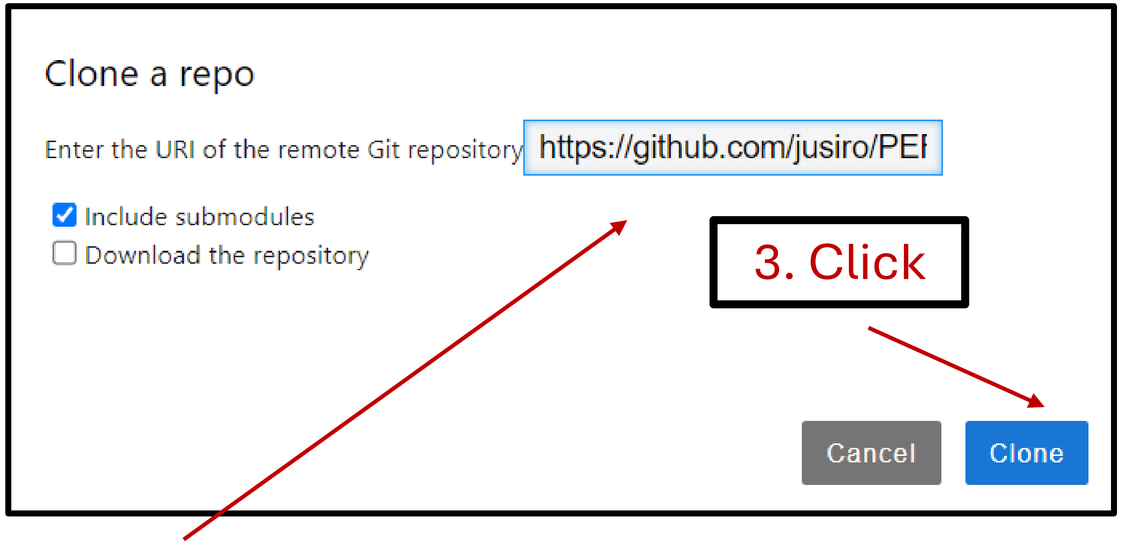


Ready to start!

Code Download



1. Git - Clone



Clone a repo

Enter the URI of the remote Git repository https://github.com/jusiro/PEI

☑ Include submodules

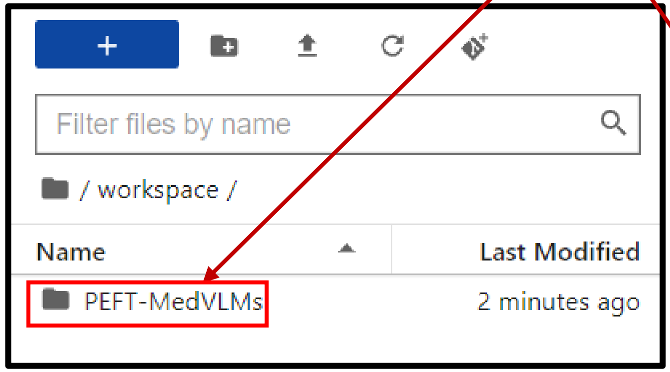
□ Download the repository

3. Click

2. Add URL

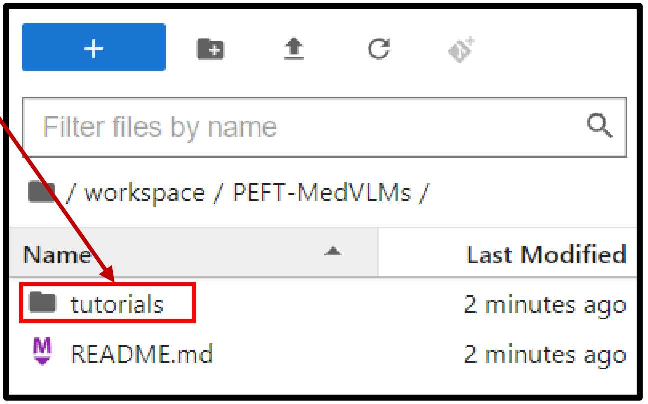
https://github.com/jusiro/PEFT-MedVLMs

4. Click



/ workspace /

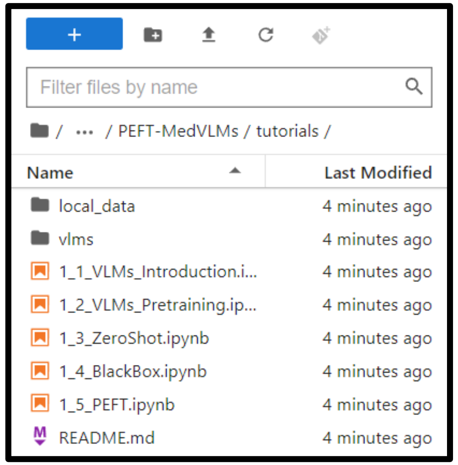
Name	Last Modified
PEFT-MedVLMs	2 minutes ago



/ workspace / PEFT-MedVLMs /

Name	Last Modified
tutorials	2 minutes ago
README.md	2 minutes ago

Ready to start!



/ … / PEFT-MedVLMs / tutorials /

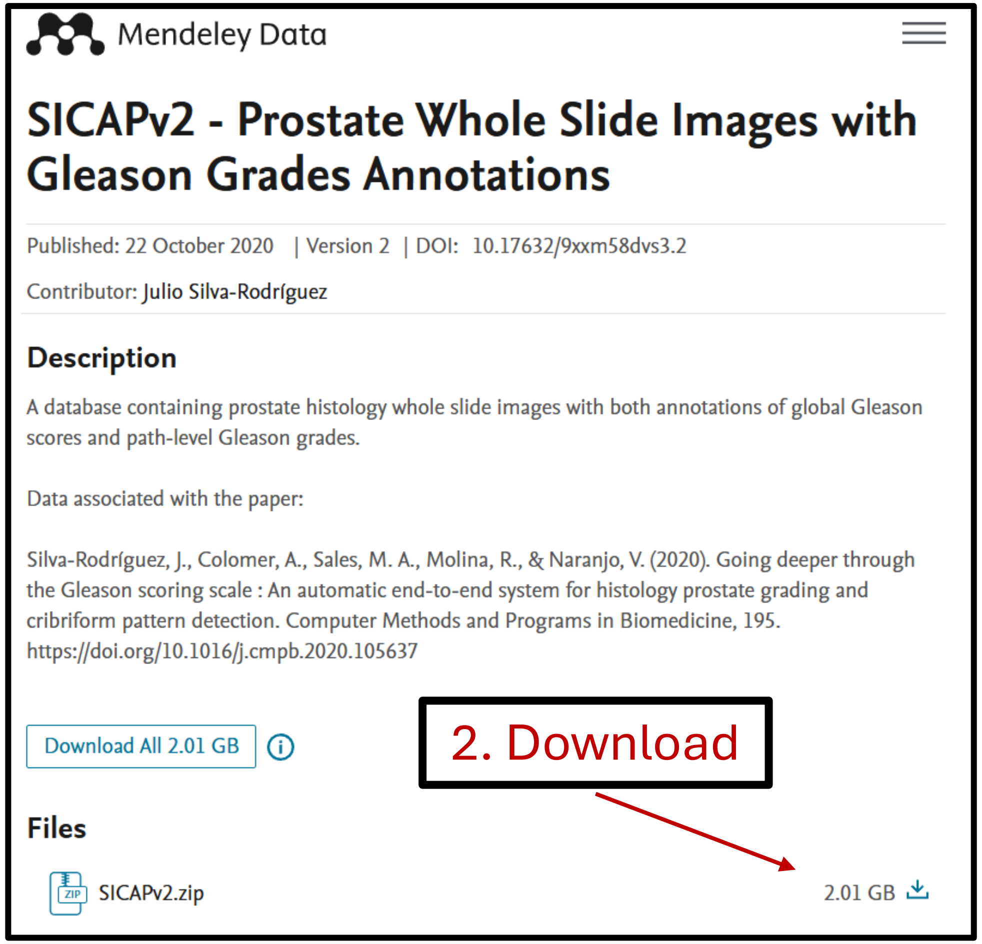
Name	Last Modified
local_data	4 minutes ago
vlms	4 minutes ago
1_1_VLMs_Introduction.i...	4 minutes ago
1_2_VLMs_Pretraining.ip...	4 minutes ago
1_3_ZeroShot.ipynb	4 minutes ago
1_4_BlackBox.ipynb	4 minutes ago
1_5_PEFT.ipynb	4 minutes ago
README.md	4 minutes ago

Datasets

1. Go to the link

⌛5 minutes

https://data.mendeley.com/datasets/9xxm58dvs3/2



Mendeley Data

SICAPv2 - Prostate Whole Slide Images with Gleason Grades Annotations

Published: 22 October 2020 | Version 2 | DOI: 10.17632/9xxm58dvs3.2

Contributor: Julio Silva-Rodríguez

Description

A database containing prostate histology whole slide images with both annotations of global Gleason scores and path-level Gleason grades.

Data associated with the paper:

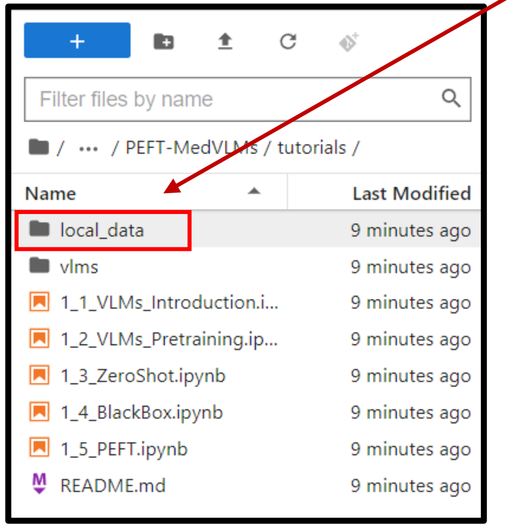
Silva-Rodríguez, J., Colomer, A., Sales, M. A., Molina, R., & Naranjo, V. (2020). Going deeper through the Gleason scoring scale : An automatic end-to-end system for histology prostate grading and cribriform pattern detection. Computer Methods and Programs in Biomedicine, 195. https://doi.org/10.1016/j.cmpb.2020.105637

Files

SICAPv2.zip	2.01 GB

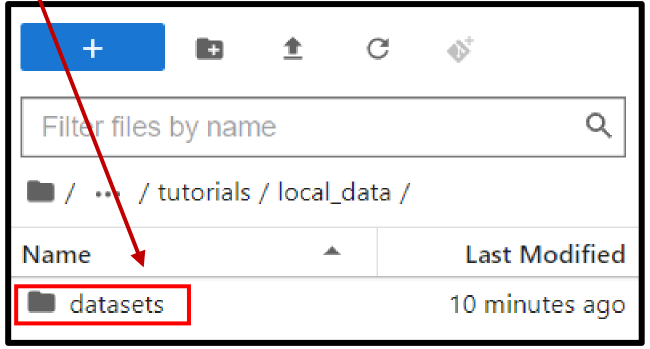
2. Download

3. Go to dataset folder



/ … / PEFT-MedVLMs / tutorials /

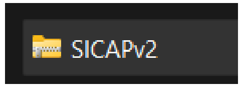
Name	Last Modified
local_data	9 minutes ago
vlms	9 minutes ago
1_1_VLMs_Introduction.i...	9 minutes ago
1_2_VLMs_Pretraining.ip...	9 minutes ago
1_3_ZeroShot.ipynb	9 minutes ago
1_4_BlackBox.ipynb	9 minutes ago
1_5_PEFT.ipynb	9 minutes ago
README.md	9 minutes ago

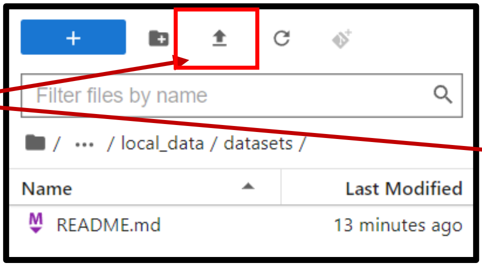


/ … / tutorials / local_data /

Name	Last Modified
datasets	10 minutes ago

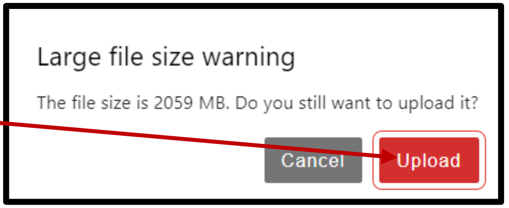
4. Upload





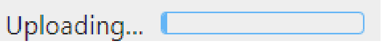
/ … / local_data / datasets /

Name	Last Modified
README.md	13 minutes ago



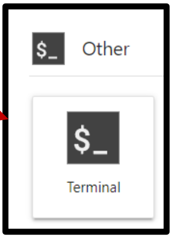
Large file size warning

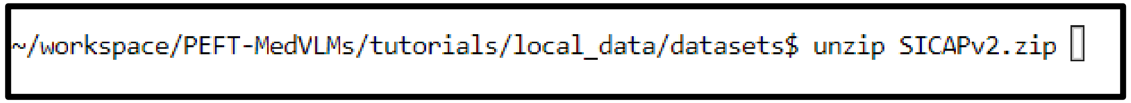
The file size is 2059 MB. Do you still want to upload it?



⌛5 minutes

5. Unzip



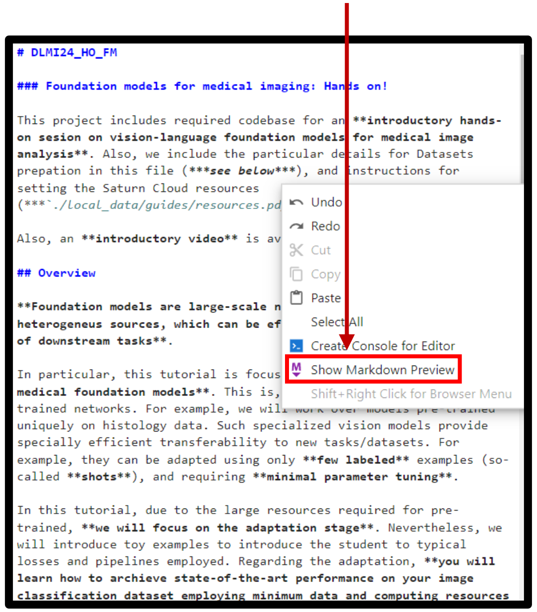


```
~/workspace/PEFT-MedVLMs/tutorials/local_data/datasets$ unzip SICAPv2.zip
```

HANDS-ON

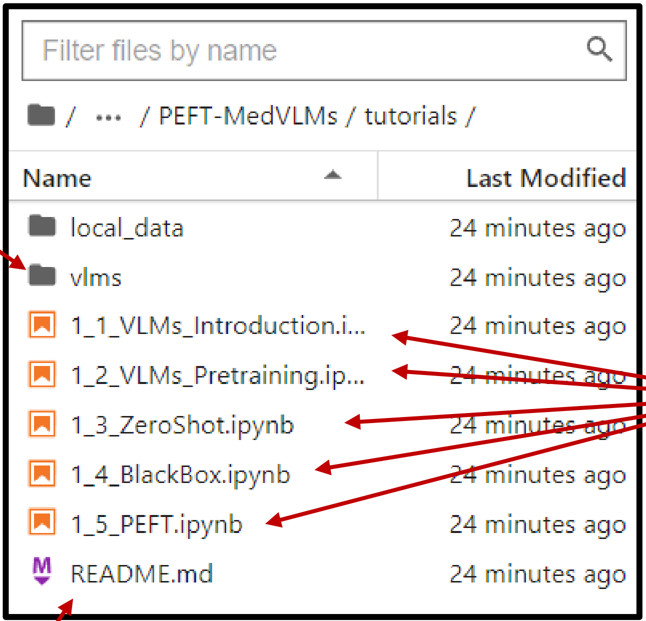
3. Auxiliary functions

Right click in the README + “Show...”



```
# DLMI24_HO_FM

### Foundation models for medical imaging: Hands on!

This project includes required codebase for an **introductory hands-
on sesion on vision-language foundation models for medical image
analysis**. Also, we include the particular details for Datasets
prepation in this file (***see below***), and instructions for
setting the Saturn Cloud resources
(***`./local_data/guides/resources.pd

Also, an **introductory video** is av

## Overview

**Foundation models are large-scale n
heterogeneus sources, which can be ef
of downstream tasks**.

In particular, this tutorial is focus
medical foundation models**. This is,
trained networks. For example, we will work over models pre-trained
uniquely on histology data. Such specialized vision models provide
specially efficient transferability to new tasks/datasets. For
example, they can be adapted using only **few labeled** examples (so-
called **shots**), and requiring **minimal parameter tuning**.

In this tutorial, due to the large resources required for pre-
trained, **we will focus on the adaptation stage**. Nevertheless, we
will introduce toy examples to introduce the student to typical
losses and pipelines employed. Regarding the adaptation, **you will
learn how to archieve state-of-the-art performance on your image
classification dataset employing minimum data and computing resources
```

Undo
Redo
Cut
Copy
Paste
Select All
Create Console for Editor
Show Markdown Preview
Shift+Right Click for Browser Menu



Filter files by name
/ ··· / PEFT-MedVLMs / tutorials /

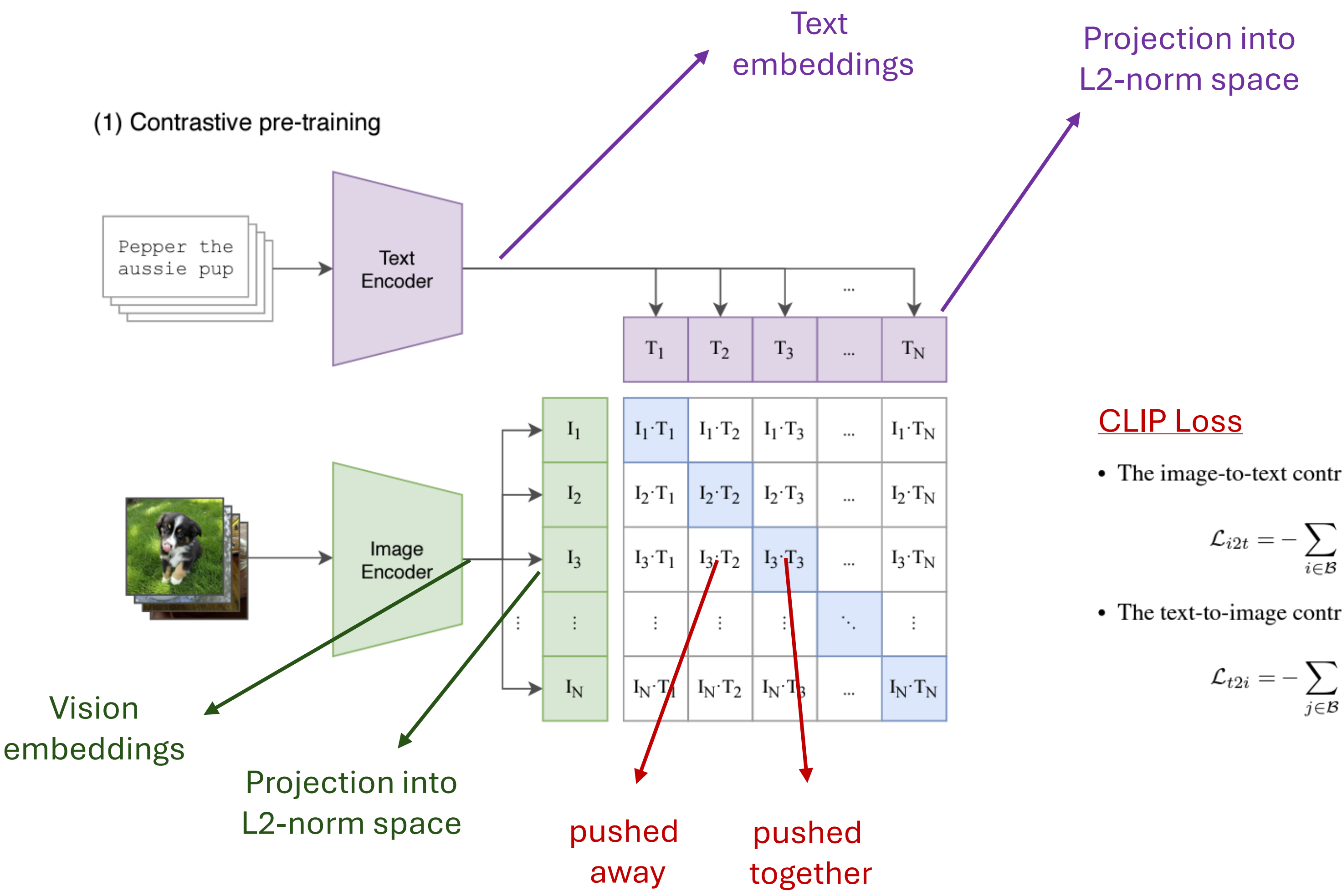
Name	Last Modified
local_data	24 minutes ago
vlms	24 minutes ago
1_1_VLMs_Introduction.i...	24 minutes ago
1_2_VLMs_Pretraining.ip...	24 minutes ago
1_3_ZeroShot.ipynb	24 minutes ago
1_4_BlackBox.ipynb	24 minutes ago
1_5_PEFT.ipynb	24 minutes ago
README.md	24 minutes ago

2. Tutorials

1. README

Introduction to Vision-Language Models and PEFT

CONTRASTIVE VISION-LANGUAE PRE-TRAINING (CLIP)



	T$_1$	T$_2$	T$_3$	...	T$_N$
I$_1$	I$_1$·T$_1$	I$_1$·T$_2$	I$_1$·T$_3$	...	I$_1$·T$_N$
I$_2$	I$_2$·T$_1$	I$_2$·T$_2$	I$_2$·T$_3$	...	I$_2$·T$_N$
I$_3$	I$_3$·T$_1$	I$_3$·T$_2$	I$_3$·T$_3$	...	I$_3$·T$_N$
⋮	⋮	⋮	⋮	⋱	⋮
I$_N$	I$_N$·T$_1$	I$_N$·T$_2$	I$_N$·T$_3$	...	I$_N$·T$_N$

Text embeddings

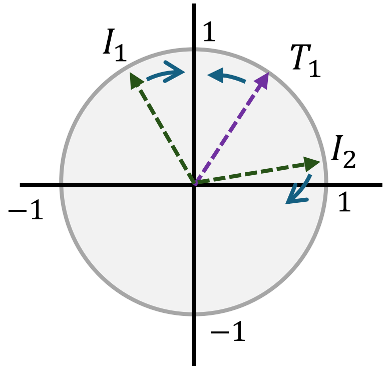
Projection into L2-norm space

Vision embeddings

Projection into L2-norm space

pushed away

pushed together



Dot product between l2-norm vectors is cosine similarity (angle between vectors)

Softmax + Cross-Entropy (over rows)

I$_1$·T$_1$	I$_1$·T$_2$	I$_1$·T$_3$	...	I$_1$·T$_N$

CLIP Loss

- The image-to-text contrastive loss

$$\mathcal{L}_{i2t} = -\sum_{i \in \mathcal{B}} \log \frac{\exp(\tau \boldsymbol{u}_i \boldsymbol{v}_i)}{\sum_{j \in \mathcal{B}} \exp(\tau \boldsymbol{u}_i \boldsymbol{v}_j)} \tag{5}$$

- The text-to-image contrastive loss

$$\mathcal{L}_{t2i} = -\sum_{j \in \mathcal{B}} \log \frac{\exp(\tau \boldsymbol{u}_j \boldsymbol{v}_j)}{\sum_{i \in \mathcal{B}} \exp(\tau \boldsymbol{u}_i \boldsymbol{v}_j)} \tag{6}$$

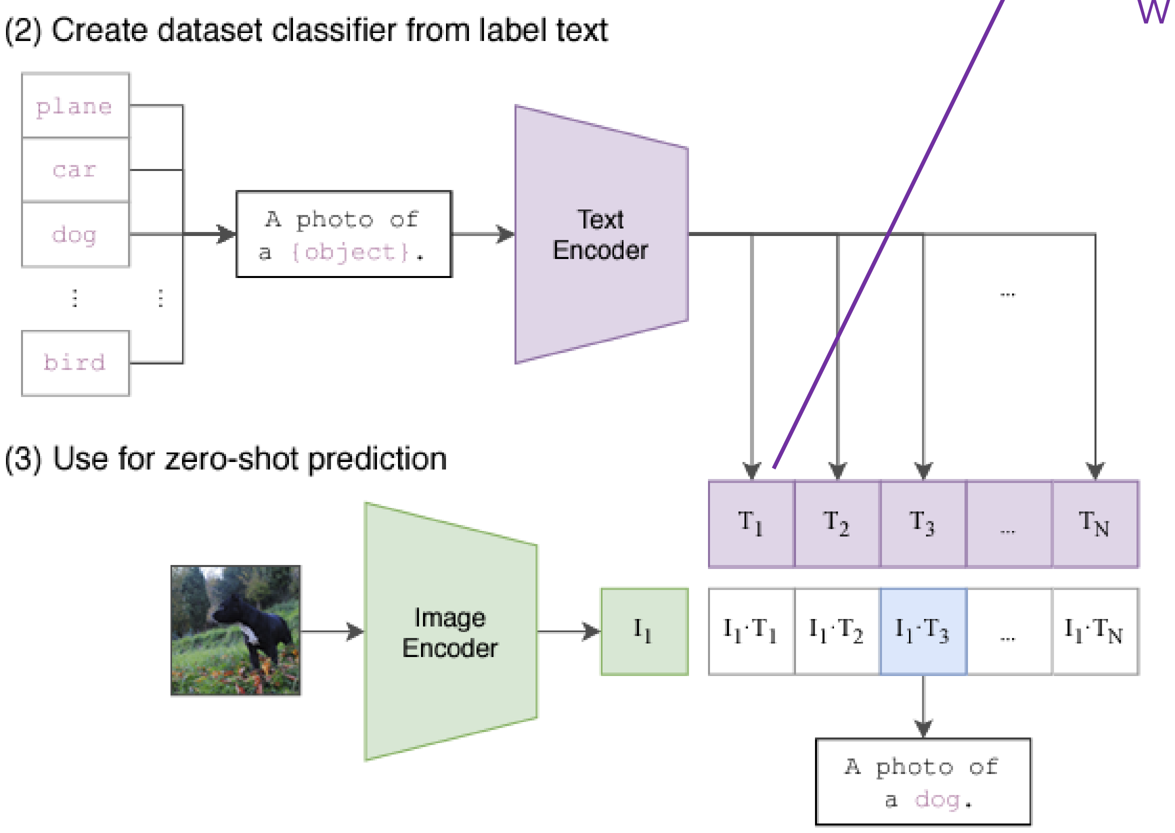
I$_1$·T$_1$
I$_2$·T$_1$
I$_3$·T$_1$
⋮
I$_N$·T$_1$

Softmax + Cross-Entropy (over columns)

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. International Conference on Machine Learning.

ZERO-SHOT PREDICTIONS

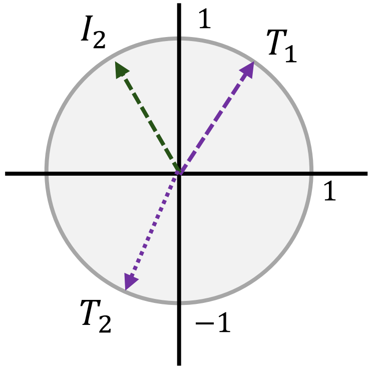
Note: Text embeddings for target categories are also called **class prototypes**, or **zero-shot prototypes**.
They do not require image samples to compute this reference embedding, but only text, and that is why they are called "zero-shot". Images with similar representations will be more likely to belong to this category.
Note that they are equivalent to a Linear output layer!
W (classes, features).



argmax over image-to-text similarities

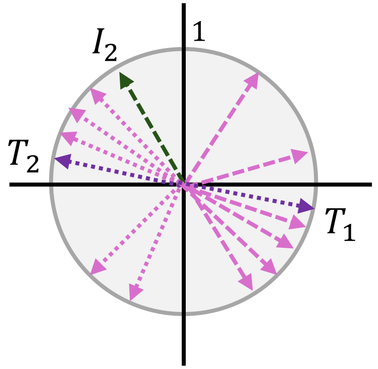
A. Single Prompt

"a photo of [CLS]"



B.Prompt Ensemble

"a photo of [CLS]"
"a sketch of [CLS]"
"small animal with black hair"
"has four legs"
"two animals playing"
…

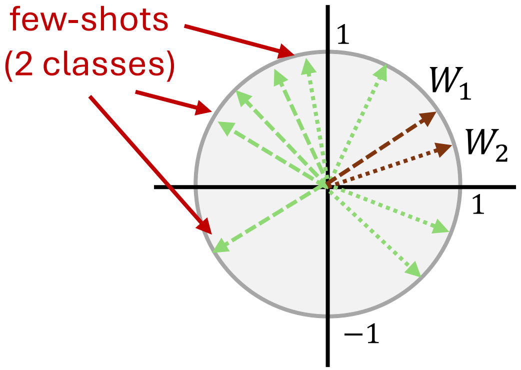


Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. International Conference on Machine Learning.

BLACK-BOX ADAPTERS

- Work over pre-computed vision features - They are backbone-agnostic.
- May profit zero-shot prototypes for the target tasks.
- They are backbone-agnostic.
- Very efficient, do not even require GPU.
- Potentially, they do not require access to pre-trained weights (similar to ChatGPT).

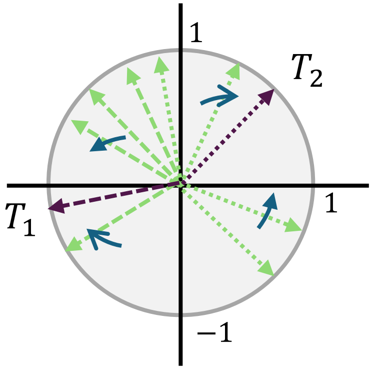
A. Linear Probing



few-shots
(2 classes)

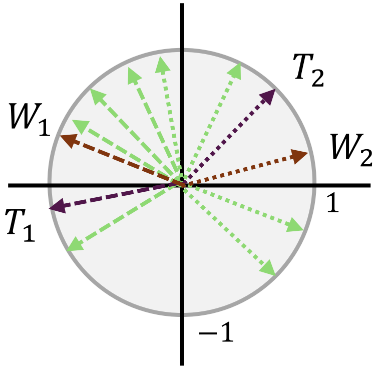
Train W1,W2 to
minimize cross-entropy
We train class prototypes

B. CLIP-Adapter



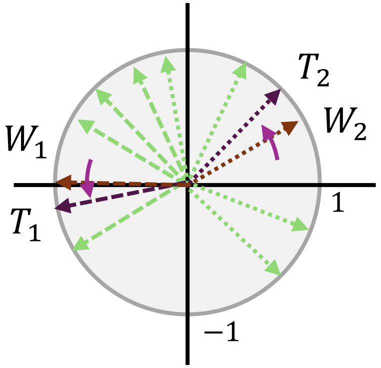
Train an mlp to modify vision features
keep class prototypes as zero-shot

C. ZS-Linear Probe



Train W1,W2 to
minimize cross-entropy
Initialize then to zero-shot

D. Class-Adaptive Linear Probe (CLAP)



Train W1,W2 to
minimize cross-entropy
Constraint them to remail close to zero-shot

Gao, P., Geng, S., Zhang, R. et al. (2024). CLIP-Adapter: Better Vision-Language Models with Feature Adapters. Int J Comput Vis.
Silva-Rodriguez, J., Hajimiri, S., Ben Ayed, I., Dolz, J. (2024). A Closer Look at the Few-Shot Adaptation of Large Vision-Language Models. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

PARAMETER-EFFICIENT FINE-TUNING

- Tran a subset of parameters to modify deep features.
- Two types: selective, and additive.
- More efficient than full-finetuning, and more flexible than black-box Adapters.
- If carefully designed, they can avoid catastrophic forgetting.

A. Affine-Layer Norm

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

We only tune these from the whole encoder

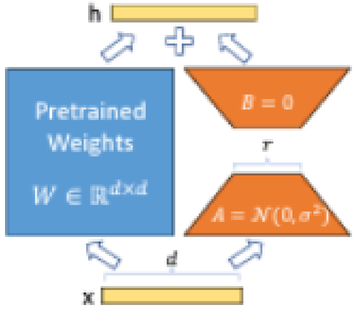
B. Bias Tuning

$$y = xA^T + b.$$

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

C. Low-Rank Adapters

Usually applied in ViTs to k,q,v layers of MultiHeadAttention


h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Add and tune a residual connection with low-rank weights.
Important!
Note B=0 when t=0

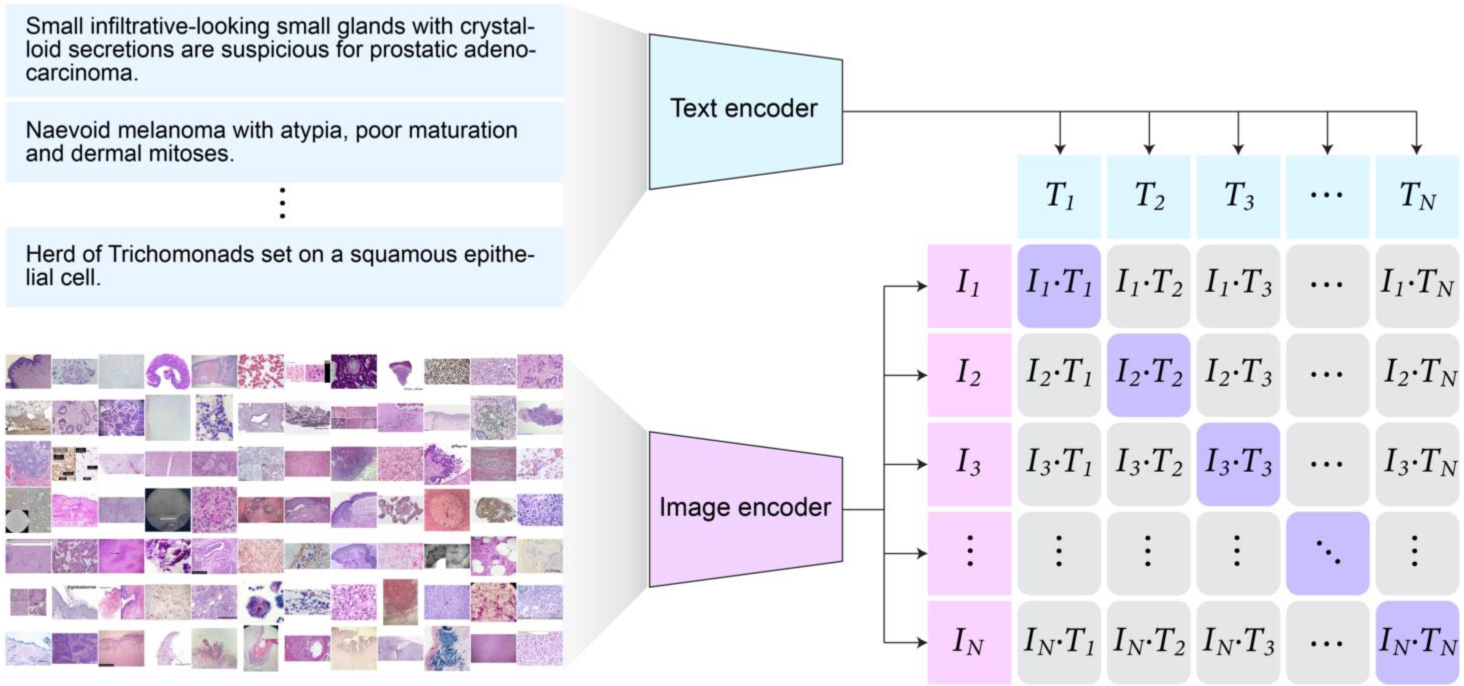
Frankle, J., Schwab, D. J., Morcos, A. S. (2021). Training batchnorm and only batchnorm: On the expressive power of random features in cnns. International Conference on Learning Representations (ICLR).

Ben-Zaken, E., Ravfogel, S., Goldberg, Y. (2021). Bitfit: Simple parameter efficient fine-tuning for transformer-based masked language-models. Association for Computational Linguistics.

Hu, E. J., et al., (2022). LoRA: Low-rank adaptation of large language models. International Conference on Learning Representations (ICLR).

MEDICAL VLMs - PLIP

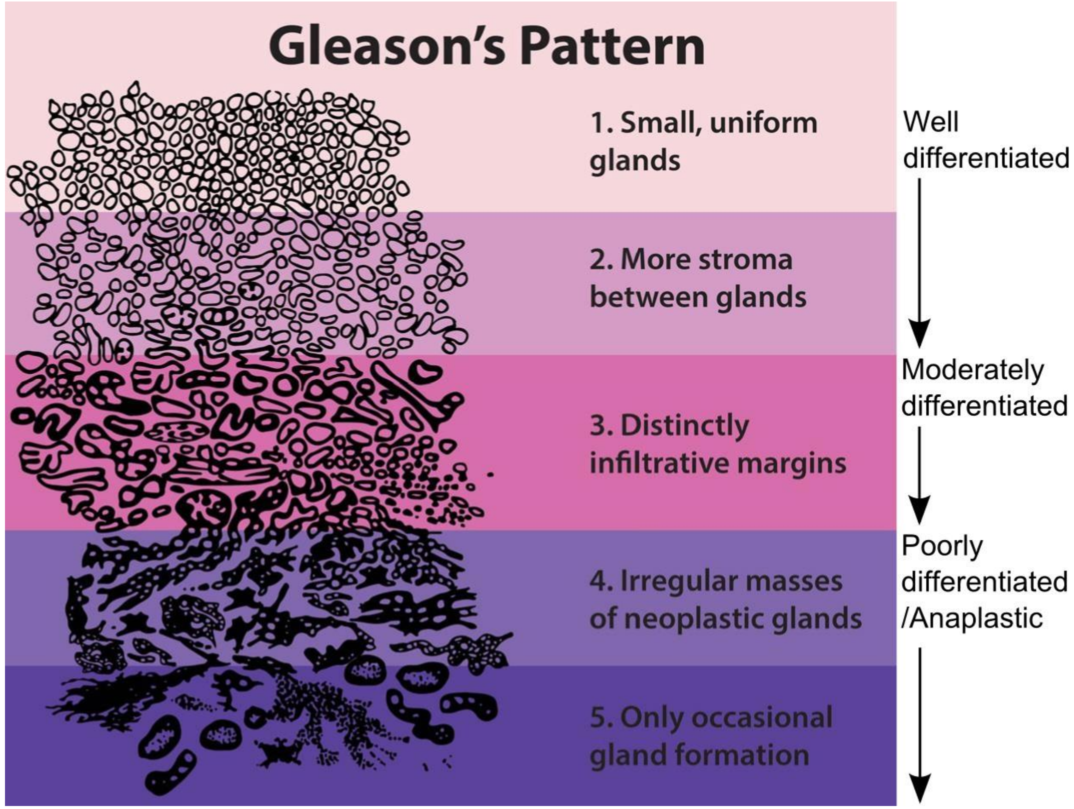
Twitter data!


Small infiltrative-looking small glands with crystalloid secretions are suspicious for prostatic adenocarcinoma.
Naevoid melanoma with atypia, poor maturation and dermal mitoses.
⋮
Herd of Trichomonads set on a squamous epithelial cell.
Text encoder
Image encoder
T_1 T_2 T_3 ⋯ T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ⋯ $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ⋯ $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ⋯ $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ⋯ $I_N \cdot T_N$

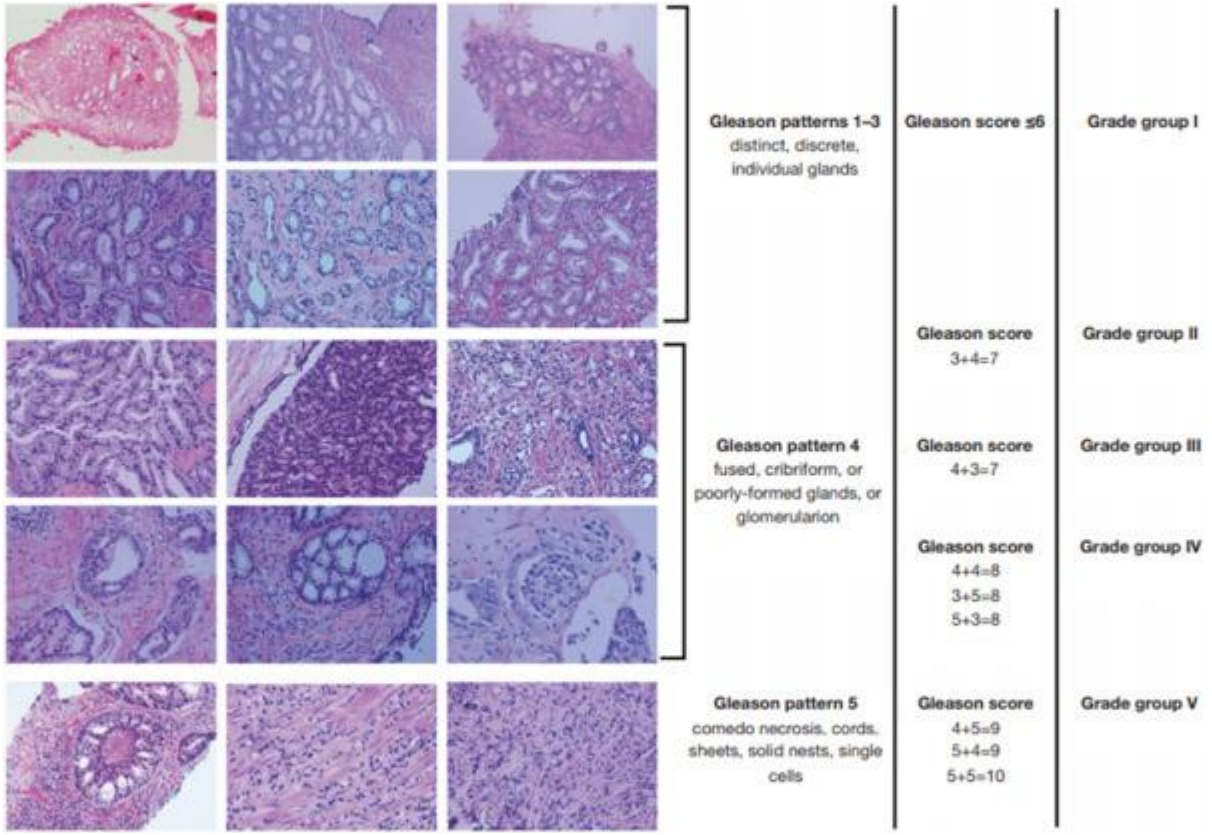
Specialized on histology images

Huang Z, Bianchi F, Yuksekgonul M, Montine TJ, Zou J. (2023). A visual-language foundation model for pathology image analysis using medical Twitter. Nature Medicine.

APPLICATION: GLEASON GRADING


Gleason's Pattern
1. Small, uniform glands
2. More stroma between glands
3. Distinctly infiltrative margins
4. Irregular masses of neoplastic glands
5. Only occasional gland formation
Well differentiated
Moderately differentiated
Poorly differentiated /Anaplastic

Gleason grading system - Wikipedia


Gleason patterns 1–3
distinct, discrete, individual glands
Gleason score ≤6
Grade group I
Gleason score 3+4=7
Grade group II
Gleason pattern 4
fused, cribriform, or poorly-formed glands, or glomerularion
Gleason score 4+3=7
Grade group III
Gleason score 4+4=8 3+5=8 5+3=8
Grade group IV
Gleason pattern 5
comedo necrosis, cords, sheets, solid nests, single cells
Gleason score 4+5=9 5+4=9 5+5=10
Grade group V

Deep learning in prostate cancer diagnosis and Gleason grading in histopathology images: An extensive study. Informatics in Medicine Unlocked.

Thanks!