

## START HERE: Instructions

### Submitting your work:

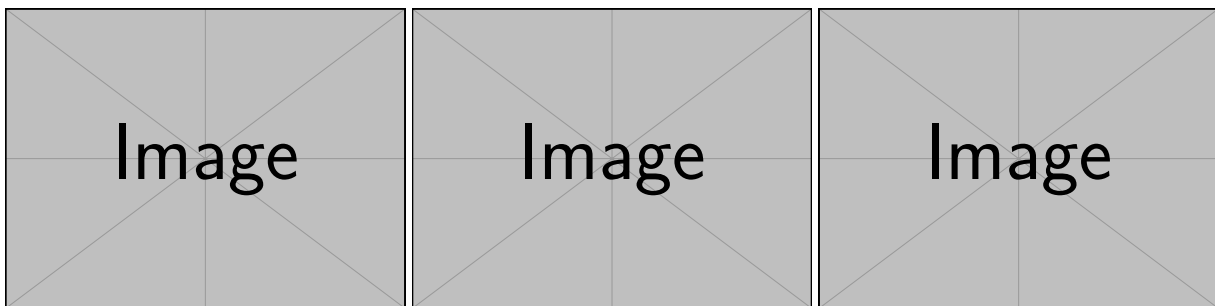
- **Deliverables:** Please submit all the `.py` files. Add all relevant plots and text answers in the boxes provided in this file. TO include plots you can simply modify the already provided latex code. Submit the compiled `.pdf` report as well.

*NOTE: Partial points will be given for implementing parts of the homework even if you don't get the mentioned numbers as long as you include partial results in this pdf.*

## 1 Image Captioning with Transformers (70 points)

We will be implementing the different pieces of a Transformer decoder ([Transformers](#)), and train it for image captioning on a subset of the [COCO dataset](#).

- **Setup:** Run the following command to extract COCO data, in the `transformer_captioning/datasets` folder: `./get_coco_captioning.sh`
  - **Question:** Follow the instructions in the `README.md` file in the `transformer_captioning` folder to complete the implementation of the transformer decoder.
  - **Deliverables:** After implementing all parts, use `run.py` for training the full model. The code will log plots to `plots`. Extract plots and paste them into the appropriate section below.
  - **Expected results:** These are expected training losses after 100 epochs. Do not change the seed in `run.py`.
    - 2-heads, 2-layers, lr 1e-4: Final loss  $\leq 1$
    - 4-heads, 6-layers, lr 1e-4: Final loss  $\leq 0.3$
    - 4-heads, 6-layers, lr 1e-3: Final loss  $\leq 0.05$
1. Paste training loss plots for each of the three hyper-param configs  
2-heads-2-layers-lr-1e-4: **TODO: fill in final train loss here.**  
4-heads-6-layers-lr-1e-4: **TODO: fill in final train loss here.**  
4-heads-6-layers-lr-1e-3: **TODO: fill in final train loss here.**

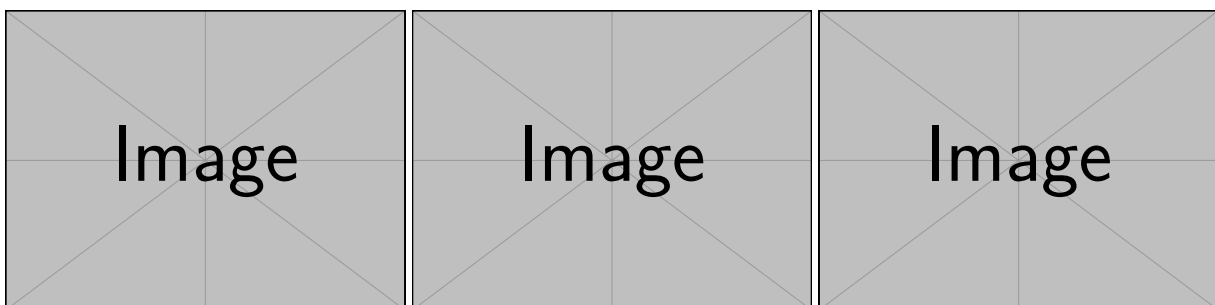


(a) 2-heads-2-layers-lr-4

(b) 4-heads-6-layers-lr-4

(c) 4-heads-6-layers-lr-3

2. Paste any three generated captioning samples from the training set with the three different settings. The provided code creates these plots at the end of training.



(a) Sample1: 2-heads-2-layers-lr-4

(b) Sample2: 4-heads-6-layers-lr-4

(c) Sample3: 4-heads-6-layers-lr-3

## Homework: Transformers

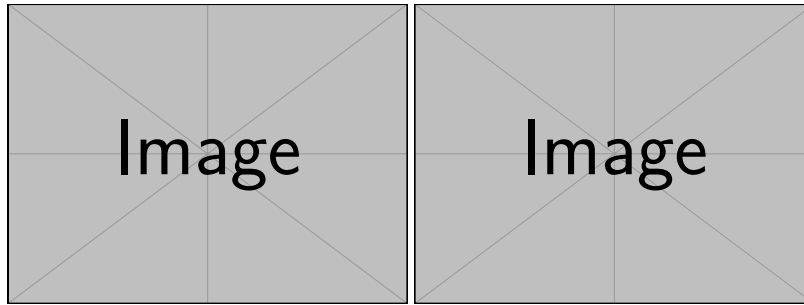
3. Based on the observations of the three different settings, What would you change in the training procedure to get better validation performance? Why tweaking these hyper-parameters will lead to better performances?

**Solution:**

## 2 Classification with Vision Transformers (30 points)

We will use the transformer you implemented in the previous part to implement a Vision Transformer (ViT), for classification on CIFAR10.

- **Question:** Follow the instructions in the `README.md` file in the `vit_classification` folder. You are encouraged to reuse code from the previous question.
- **Deliverables:** Run training using `run.py` for training the full model. The code will log plots `acc_out.png` (train and test accuracy) and `loss_out.png` (train loss).
- **Expected Results:** After 100 epochs, test accuracy should be  $\geq 65\%$ , train accuracy should be  $\approx 100\%$ , and training loss  $\leq 0.3$ .



(a) Train/test accuracy

(b) Training loss

**Collaboration Survey** Please answer the following:

1. Did you receive any help whatsoever from anyone in solving this assignment?

☐ Yes

☐ No

- If you answered 'Yes', give full details:
- (e.g. "Jane Doe explained to me what is asked in Question 3.4")

2. Did you give any help whatsoever to anyone in solving this assignment?

☐ Yes

☐ No

- If you answered 'Yes', give full details:
- (e.g. "I pointed Joe Smith to section 2.3 since he didn't know how to proceed with Question 2")

3. Note that copying code or writeup even from a collaborator or anywhere on the internet violates the [Academic Integrity Code of Conduct](#).