

# How to write a good course project report

CSC413/2516 Tutorial 5

Micaela Consens 2023

With inspiration from Keiran Paster (2021) and Marta Skreta (2022)

# Final Project

1

Original research  
experience

2

Technical writing  
experience



# Final Project

1

Original research  
experience

- Expand your understanding of topics in the course
  - What interests you?
  - What did you want to learn more about?
- Find gaps in experimental knowledge
  - What hasn't been tried yet?
  - What is missing from the picture?

2

Technical writing  
experience

- Summarize the current state of the field
  - What are seminal papers in the area?
  - What has the progress been?
- Explain how you are addressing gaps in knowledge
  - How is your work a **contribution** to the field?

# Final Project

1

## Original research experience

- Expand your understanding of topics in the course
  - What interests you?
  - What did you want to learn more about?
- Find gaps in experimental knowledge
  - What hasn't been tried yet?
  - What is missing from the picture?

2

## Technical writing experience

- Summarize the current state of the field
  - What are seminal papers in the area?
  - What has the progress been?
- Explain how you are addressing gaps in knowledge
  - How is your work a **contribution** to the field?

# Where to Begin

## Literature Review

- You're going to have to read some papers
  - This is a skill in and of itself
- You will have to actually read them
- The good news is, this can help with ideation



Reorganizing papers that he has no serious plans to read, the CSC413/2516 student engages in barely productive procrastination.

# How to Read a Paper

- Do a **first-pass** read of the paper
  - Read the introduction first
  - Read the conclusion next to put the experiments and results into a context
  - Read the rest of the paper in order
- Then do a **second-pass**
  - Read it start to finish and take notes
  - Highlight words/topics you don't understand and write down their definitions/understanding
- Give **figures and tables extra** time

# Literature Review



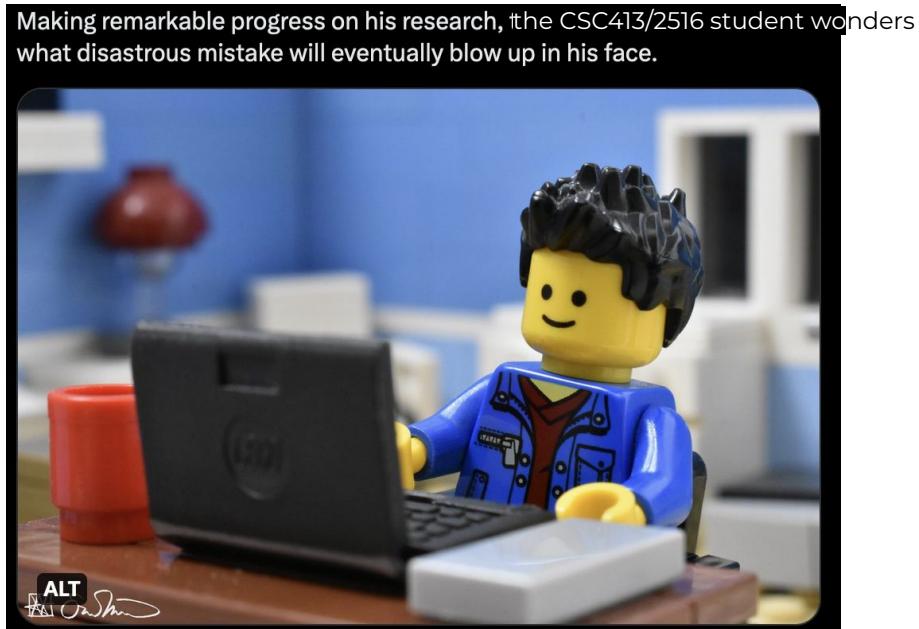
Rummaging through non-circulating books crucial to his research,  
the CSC413/2516 student can almost hear the clinking of his shackles.

- Think about what you've found most interesting in the course so far
- Or what area you want experience in
- Look at recent papers that have come out in that area
- What did they do that was cool?

# What's next?

## Project Idea

- You can pivot ideas, so don't feel too much pressure
- Pick a general field within deep learning that interests you, and give yourself room to change ideas within it
- You **will** end up modifying many aspects of your project plan and that is to be expected



# Once you have an idea...

- You can start your project outline
- You don't have to wait until you've done a lot of research to begin
- Several sections can be started and worked on while you are experimenting



What sections?

# Report Organization

*Typical sections*

- Introduction
- Related Work / Background
- Method
- Experiments
- Results
- Conclusion

# Report Organization

*Typical sections*

- Introduction
- Related Work / Background ←
- Method
- Experiments
- Results
- Conclusion

# Related Work / Background

Since at this point you've already done an extensive literature review...

- Summarize key findings from papers that you used to get inspiration for your idea
- Describe any papers from which you are using their methods (including for baseline evaluation!)
- Key items:
  - Provide references to important prior works
  - Explain the history of solutions to the problem
  - Place your paper's contributions in context

# Related Work / Background Tips

Include:

- Key components a reader should know before reading your paper
- A common theme to each paragraph in this section
- Be extremely generous in including related work. Be intellectually honest.

# Related Work / Background → EXAMPLE

## Attention Is All You Need

- Paragraphs are organized by high-level topic
- Many citations of prior work
- Ends placing current paper in context, stating contribution

### 2 Background

The goal of reducing sequential computation also forms the foundation of the Extended Neural GPU [16], ByteNet [18] and ConvS2S [9], all of which use convolutional neural networks as basic building block, computing hidden representations in parallel for all input and output positions. In these models, the number of operations required to relate signals from two arbitrary input or output positions grows in the distance between positions, linearly for ConvS2S and logarithmically for ByteNet. This makes it more difficult to learn dependencies between distant positions [12]. In the Transformer this is reduced to a constant number of operations, albeit at the cost of reduced effective resolution due to averaging attention-weighted positions, an effect we counteract with Multi-Head Attention as described in section 3.2.

Self-attention sometimes called intra-attention is an attention mechanism relating different positions or a single sequence in order to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [4, 27, 28, 22].

End-to-end memory networks are based on a recurrent attention mechanism instead of sequence-aligned recurrence and have been shown to perform well on simple-language question answering and language modeling tasks [34].

To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution. In the following sections, we will describe the Transformer, motivate self-attention and discuss its advantages over models such as [17, 18] and [9].

# Report Organization

*Typical sections*

- Introduction
- Related Work / Background
- Method
- Experiments
- Results
- Conclusion



# Experiments

The main goal here is **reproducibility**, write down what you're doing as you experiment (KEEP A RECORD)

- Your report should provide enough information to reproduce results.
  - Architecture, loss, optimizer details, seed, hyperparameters (and how you found those hyperparameters)
  - What tricks were useful to make your approach work?
  - What were the most important parameters to tune?
  - How much compute was used?
- This section does not need to be extremely long
- Methods is related to this section

# Experiments Tips

As you code/test things out:

- Keep a log of everything you're trying and the parameters you're using
- Fill out bullet points in this section as you experiment
- Clean it up (paragraph format & good writing style) when you're done experimenting

# Experiments → EXAMPLE

## Attention Is All You Need

- Paragraphs are organized by different settings under which experiment was run
- All settings are included to recreate their results

This section describes the training regime for our models.

### 5.1 Training Data and Batching

We trained on the standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs. Sentences were encoded using byte-pair encoding [3], which has a shared source-target vocabulary of about 37000 tokens. For English-French, we used the significantly larger WMT 2014 English-French dataset consisting of 36M sentences and split tokens into a 32000 word-piece vocabulary [38]. Sentence pairs were batched together by approximate sequence length. Each training batch contained a set of sentence pairs containing approximately 25000 source tokens and 25000 target tokens.

### 5.2 Hardware and Schedule

We trained our models on one machine with 8 NVIDIA P100 GPUs. For our base models using the hyperparameters described throughout the paper, each training step took about 0.4 seconds. We trained the base models for a total of 100,000 steps or 12 hours. For our big models,(described on the bottom line of table [3]), step time was 1.0 seconds. The big models were trained for 300,000 steps (3.5 days).

### 5.3 Optimizer

We used the Adam optimizer [20] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$  and  $\epsilon = 10^{-9}$ . We varied the learning rate over the course of training, according to the formula:

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5}) \quad (3)$$

This corresponds to increasing the learning rate linearly for the first  $warmup\_steps$  training steps, and decreasing it thereafter proportionally to the inverse square root of the step number. We used  $warmup\_steps = 4000$ .

# Report Organization

*Typical sections*

- Introduction
- Related Work / Background
- Method
- Experiments
- Results
- Conclusion



# Method

Here you present the main non-experimental results

- Show in detail what the method is and why it is (theoretically) justified
- This can include description of model architecture, equations/theorems that are used etc.

# Method Tips

Include:

- Algorithm box, equations describing your model, theorems or formally stated conjectures
- Image of the model architecture or algorithm pipeline
- Write this section along with Experiments

# Method → EXAMPLE

## Attention Is All You Need

- Model architecture shown

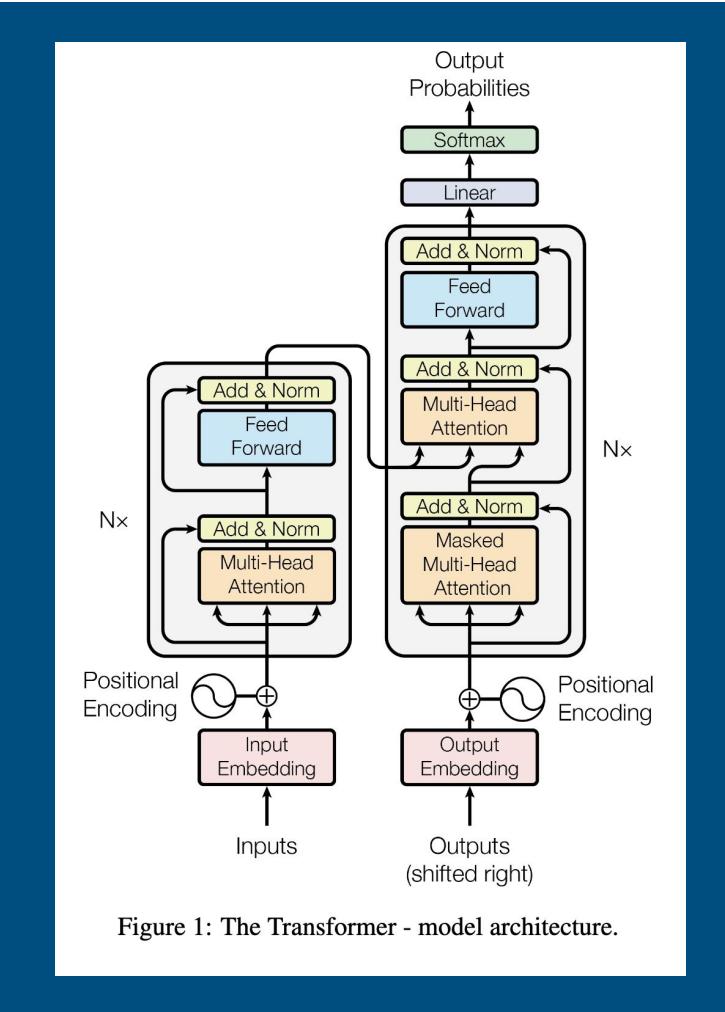


Figure 1: The Transformer - model architecture.

# Method → EXAMPLE

## Attention Is All You Need

- Justification of the theoretical validity of the method

### 4 Why Self-Attention

In this section we compare various aspects of self-attention layers to the recurrent and convolutional layers commonly used for mapping one variable-length sequence of symbol representations  $(x_1, \dots, x_n)$  to another sequence of equal length  $(z_1, \dots, z_n)$ , with  $x_i, z_i \in \mathbb{R}^d$ , such as a hidden layer in a typical sequence transduction encoder or decoder. Motivating our use of self-attention we consider three desiderata.

One is the total computational complexity per layer. Another is the amount of computation that can be parallelized, as measured by the minimum number of sequential operations required.

The third is the path length between long-range dependencies in the network. Learning long-range dependencies is a key challenge in many sequence transduction tasks. One key factor affecting the ability to learn such dependencies is the length of the paths forward and backward signals have to traverse in the network. The shorter these paths between any combination of positions in the input and output sequences, the easier it is to learn long-range dependencies [12]. Hence we also compare the maximum path length between any two input and output positions in networks composed of the different layer types.

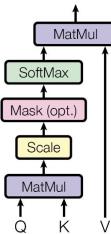
As noted in Table 1, a self-attention layer connects all positions with a constant number of sequentially executed operations, whereas a recurrent layer requires  $O(n)$  sequential operations. In terms of computational complexity, self-attention layers are faster than recurrent layers when the sequence length  $n$  is smaller than the representation dimensionality  $d$ , which is most often the case with sentence representations used by state-of-the-art models in machine translations, such as word-piece [38] and byte-pair [31] representations. To improve computational performance for tasks involving very long sequences, self-attention could be restricted to considering only a neighborhood of size  $r$  in

# Method → EXAMPLE

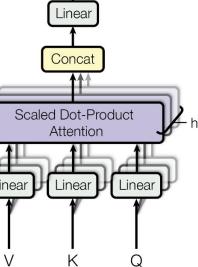
## Attention Is All You Need

- Attention mechanism explained and equation shown

Scaled Dot-Product Attention



Multi-Head Attention



### 3.2.1 Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension  $d_k$ , and values of dimension  $d_v$ . We compute the dot products of the query with all keys, divide each by  $\sqrt{d_k}$ , and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix  $Q$ . The keys and values are also packed together into matrices  $K$  and  $V$ . We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

The two most commonly used attention functions are additive attention [2], and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of  $\frac{1}{\sqrt{d_k}}$ . Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

While for small values of  $d_k$  the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of  $d_k$  [3]. We suspect that for large values of  $d_k$ , the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients [4]. To counteract this effect, we scale the dot products by  $\frac{1}{\sqrt{d_k}}$ .

# Report Organization

*Typical sections*

- Introduction
- Related Work / Background
- Method
- Experiments
- Results
- Conclusion



# Results

This section is focused on showing the results of the experiment and methodology

- Metrics
  - Describe the metrics you are using to define success.
  - What metrics make sense given your context?
  - Is your data imbalanced? Accuracy value (# correct / total) may be uninformative
  - How bad are false positives, false negatives?

# Results

This section is focused on showing the results of the experiment and methodology

- Metrics
  - Some useful metrics to think about: AUROC, AUPRC, F-score, macro/micro accuracy, KL divergence, loss
  - Can you get a confidence bound or a p-value\*?
  - \*Different subfields have different trends for reporting error estimates and significance. Take a look at what previous papers did.
- As a starting point, look at what metrics previous works used

# Results Tips

- Make sure you show the performance of your work against an established baseline
  - You cannot simply say your work is better without stating *how* and *what* it is better than
- It is generally good to include a table showing the performance of your algorithm/model

# Results → EXAMPLE

## Attention Is All You Need

- Table is included which shows how their method outperformed current SOTA

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	<b>41.29</b>	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	<b><math>3.3 \cdot 10^{18}</math></b>	
Transformer (big)	<b>28.4</b>	<b>41.8</b>	$2.3 \cdot 10^{19}$	

# Results → EXAMPLE

## Attention Is All You Need

- When presenting results they summarize the findings of the experiments and place them in context of previous works

### 6.1 Machine Translation

On the WMT 2014 English-to-German translation task, the big transformer model (Transformer (big) in Table 2) outperforms the best previously reported models (including ensembles) by more than 2.0 BLEU, establishing a new state-of-the-art BLEU score of 28.4. The configuration of this model is listed in the bottom line of Table 3. Training took 3.5 days on 8 P100 GPUs. Even our base model surpasses all previously published models and ensembles, at a fraction of the training cost of any of the competitive models.

On the WMT 2014 English-to-French translation task, our big model achieves a BLEU score of 41.0 outperforming all of the previously published single models, at less than 1/4 the training cost of the previous state-of-the-art model. The Transformer (big) model trained for English-to-French used dropout rate  $P_{drop} = 0.1$ , instead of 0.3.

For the base models, we used a single model obtained by averaging the last 5 checkpoints, which were written at 10-minute intervals. For the big models, we averaged the last 20 checkpoints. We used beam search with a beam size of 4 and length penalty  $\alpha = 0.6$  [38]. These hyperparameters were chosen after experimentation on the development set. We set the maximum output length during inference to input length + 50, but terminate early when possible [38].

Table 2 summarizes our results and compares our translation quality and training costs to other model architectures from the literature. We estimate the number of floating point operations used to train a model by multiplying the training time, the number of GPUs used, and an estimate of the sustained single-precision floating-point capacity of each GPU<sup>5</sup>.

# Report Organization

*Typical sections*

- Introduction
- Related Work / Background
- Method
- Experiments
- Results
- Conclusion



# Introduction

The introduction and conclusion should relate to one another, it's best to write them both last

- Objective:
  - Help the reader understand the problem and why it is important.
- Show that other solutions to the problem are unsatisfactory.
- Show the new solution to the problem and say why it's better

# Introduction Tips

General flow:

- Introduce the field you are contributing to
- Explain the knowledge gap you are addressing and provide motivation for why it is important
- Make a statement about the contribution of your work

# Introduction → EXAMPLE

## Attention Is All You Need

- Introduces the field which the paper contributes to
- States a gap in knowledge in the field & motivates finding a solution
- Introduces the contribution of their method/model

### 1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [17] neural networks, in particular, have been firmly established as state of the art approaches in sequence modeling and

transduction problems such as language modeling and machine translation [35, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [38, 24, 15].

Recurrent models typically factor computation along the symbol positions of the input and output sequences. Aligning the positions to steps in computation time, they generate a sequence of hidden states  $h_t$ , as a function of the previous hidden state  $h_{t-1}$  and the input for position  $t$ . This inherently sequential nature precludes parallelization within training examples, which becomes critical at longer sequence lengths, as memory constraints limit batching across examples. Recent work has achieved significant improvements in computational efficiency through factorization tricks [21] and conditional computation [32], while also improving model performance in case of the latter. The fundamental constraint of sequential computation, however, remains.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [2, 19]. In all but a few cases [27], however, such attention mechanisms are used in conjunction with a recurrent network.

In this work we propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs.

# Report Organization

*Typical sections*

- Introduction
- Related Work / Background
- Method
- Experiments
- Results
- Conclusion



# Conclusion

Here you present a concise version of the findings, with a focus on why your method/approach is useful for the field

- State hypotheses and motivation for each experiment.
- Think about whether you have clear evidence to verify or nullify the hypothesis.
- Don't cherry pick results!
- Including negative results is fine, especially for this project report.
- If you have negative results, would be good to write about why you think it turned out that way

# Conclusion

Don't forget to include limitations/future directions

- Write about limitations of your work.
  - If you weren't able to run all the experiments you planned to, why not?
  - Was anything holding you back? E.g. compute, time
  - Are there underlying problems with your dataset, model that you couldn't fix?
- Mention some future work (experiments to extend your work)

# Conclusion Tips

- Provide a high level summary of what the experiments showed, and how they support your models contribution
- Middle section on limitations
- End with future directions

# Conclusion → EXAMPLE

## Attention Is All You Need

- Summarize the contribution and some results
- They didn't explicitly mention limitations but also it's the transformer paper so they get off free on that
- End with future work

### 7 Conclusion

In this work, we presented the Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention.

For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers. On both WMT 2014 English-to-German and WMT 2014 English-to-French translation tasks we achieve a new state of the art. In the former task our best model outperforms even all previously reported ensembles.

We are excited about the future of attention-based models and plan to apply them to other tasks. We plan to extend the Transformer to problems involving input and output modalities other than text and to investigate local, restricted attention mechanisms to efficiently handle large inputs and outputs such as images, audio and video. Making generation less sequential is another research goals of ours.

The code we used to train and evaluate our models is available at <https://github.com/tensorflow/tensor2tensor>.

# On Figures...

- Key details in the paper should not be hidden within figures
- A reader should be able to understand your paper only looking at the figures
- Figures stand out and readers should be able to get a general idea about the paper just from looking at them.
- When writing captions, keep in mind that some people may not have read the text that refers to the figure. Include enough details to explain the main idea of the figure.

# References

- Use [dblp.org](#) instead of arxiv to get bibtex entries.
- Cite generously
  - Make sure to cite sources wherever you use them (not just the first time; not just in the related works section)
  - If in doubt, cite (better to over-cite than under-cite!)
- Use `\citet{}` for a textual citation, `\citep{}` for parenthetical.

While forward dynamics models map a state and action to the next state, an inverse dynamics model maps two subsequent states to an action. Inverse dynamics models have been used in various ways in sequential decision making. In exploration, inverse dynamics serves as a way to learn representations of the controllable aspects of the state (Pathak et al., 2017). In imitation learning, inverse dynamics models can be used to map a sequence of states to the actions needed to imitate the trajectory (Pavse et al., 2019). Christiano et al. (2016) use inverse dynamics models to translate actions taken in a simulated environment to the real world.

# Final Step

*Stop editing at  
some point and  
submit the  
report*

Editing a paper draft to an obsessive degree, the CSC413/2516 student cherishes a rare moment of having control.



# Rubric

- **Quality** [35%] (do good technical work)
- **Clarity** [25%] (follow the writing guidelines in this tutorial)
- **Originality** [20%] (is what you are doing novel?)
- **Significance** [5%] (are the results important? important can mean many things!)
- **Participation** [15%] (write reviews of other students' projects)

# How to Review another student

*From the ICLR 2021 Reviewer Guide*

- Read the paper carefully, considering the following:
  - What is the goal of the paper?
  - What did the authors do well?
  - What are the weaknesses of the paper?
  - Keep in mind that even if a paper isn't interesting to you, it may be interesting to others. (AKA be nice)

# How to Review another student

*From the ICLR 2021 Reviewer Guide*

- Answer these key questions:
  - What is specifically being tackled by the paper?
  - Is the approach well motivated?
  - Does the paper support the claims it makes?

# The Review

- Summarize what the paper contributes.
- List strong and weak points.
- State the score you are giving the paper.
- Provide support for your score.
- Write what would be needed to improve the score.

# Resources

*Much of this tutorial is inspired by the following resources:*

- <https://billf.mit.edu/sites/default/files/documents/cvprPapers.pdf>
- <http://www.ai.mit.edu/courses/6.899/papers/ted.htm>
- <http://approximatelycorrect.com/2018/01/29/heuristics-technical-scientific-writing-machine-learning-perspective/>

# GOOD LUCK!

Enjoying his work, the CSC413/2516 student/ ponders whether he has fallen victim to Stockholm syndrome.

