Lab - 1: NanoGPT

Mihir Gohel (117284926)*
Stony Brook University
CS519, Data Science fundamentals

I. FREQUENCY DISTRIBUTION OF CHARACTERS IN THE CORPORA

Frequency distribution for math, wikipedia, and shakespeare are Fig. 1, Fig. 2, and Fig. 3 respectively.

II. ANALYSIS MODEL ARCHITECTURE DIAGRAM

Analysis model Architecture Diagram for the math, wikipedia, and shakespeare dataset are as: Fig: 4, Fig: 5, and Fig: 6 respectively. Here, n_layer=2, n_head=2, n_embd=2, block_size=64, vocab_size_math=110, vocab_size_wikipedia=292, vocab_size_shakespeare=65.

III. ANALYSIS: TRACK MODEL OUTPUT OVER TRAINING

Math:
At lower iterations there was a lot of misuse of mathematical symbols. Iterations above 4000 had lesser spelling mistakes and lower usage of mathematical symbols. At higher iterations there are repetitive patterns, like using 'time-constant' after the term 'time' and using some frequent mathematical symbols like ';', '=', at places where it's not required.

Wikipedia:
After 2500 iterations the text it generated had less and less spelling mistakes. Although at 4000-5000 iterations there are still a lot of grammatical and contextual mistakes, but is still better in terms of structure. At higher iterations there are repetitive characters like joining 's' after 'time', even if it's grammatically incorrect.

Shakespeare:
At lower iterations it tends to make more spelling mistakes and higher usage of old english terms. At higher iterations this gets better and there are less of a spelling errors. In terms of repetition, there's high usuage of ':', '.', ';'.

In conclusion, math dataset will try to use as many mathematical equations as possible, wikipedia will try to generate essay type sentences and shakespeare will try to generate sentences in old-english tone.

Files for the same: Link

	wikipedia	math	shakespeare
wikipedia	1.424	2.618	2.472
math	2.719	1.219	3.52
shakespeare	2.369	3.988	1.407

TABLE I: Cross domain loss matrix

IV. ANALYSIS: VARIABILITY OF THE FINAL MODEL

Math:
There's not much of a variation between the 5 outputs. It's trying to use numbers, symbols, more and more often.

Wikipedia:
There's not much of a variation between the 5 outputs. They all are different, but does sound like wikipedia text. I'm not fully satisfied because there's a lot of grammatical errors and there's a lack of coherence.

Shakespeare:
Again there's not much of a variation between the 5 outputs. There's a lot of usage of capital letters and grammatical errors.

In conclusion, these models needs more training and cross-training for better output.

Files for the same: Link

V. ANALYSIS: PLOT TRAINING LOSS FROM CHECKPOINTS

Fig. 7, Fig. 8, and Fig. 8 shows the loss curves for the increasing iterations.
newline
Initially in all the graphs there's a huge drop in the loss value, which is not sufficient for generating meaningful text. It's only after 4000-5000 iterations the loss value gets significantly lower for it to generate some meaningful text.

*Electronic address: mgohel@cs.stonybrook.edu

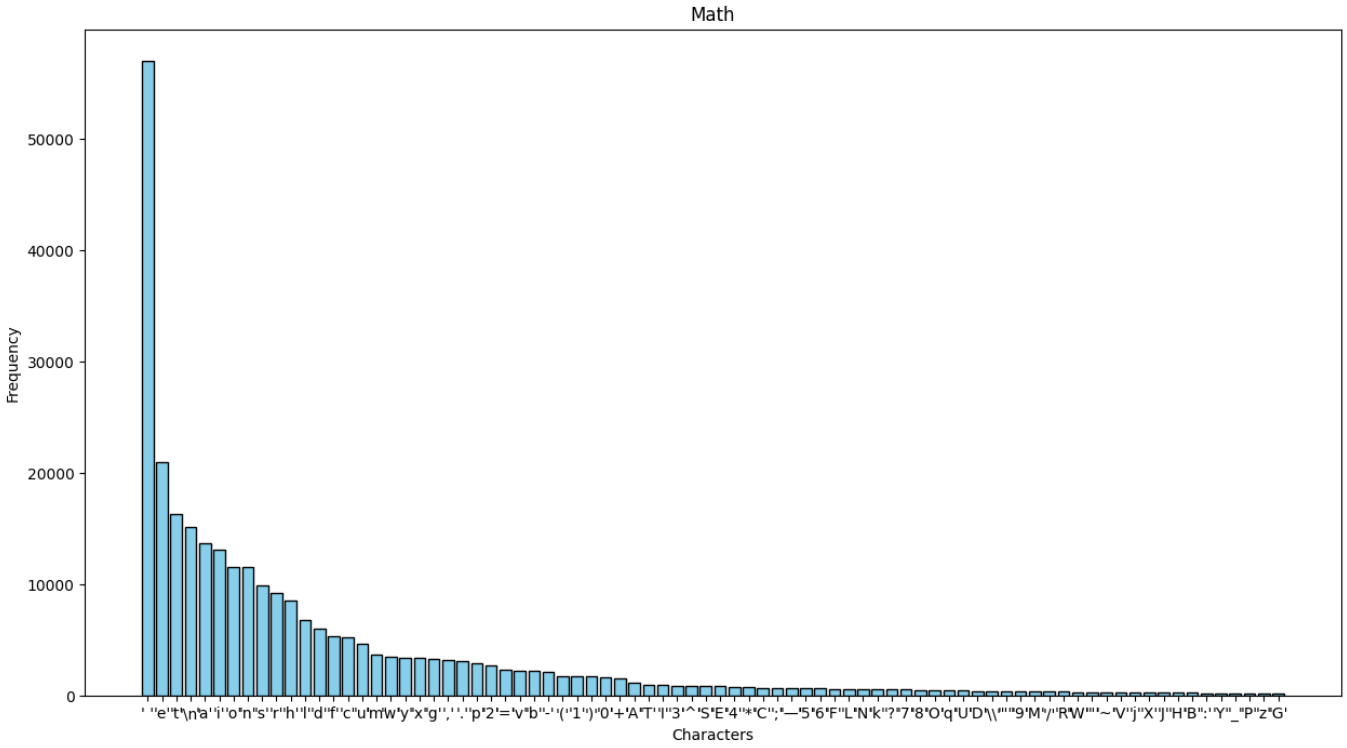

FIG. 1: Frequency distribution of math dataset (top 80)

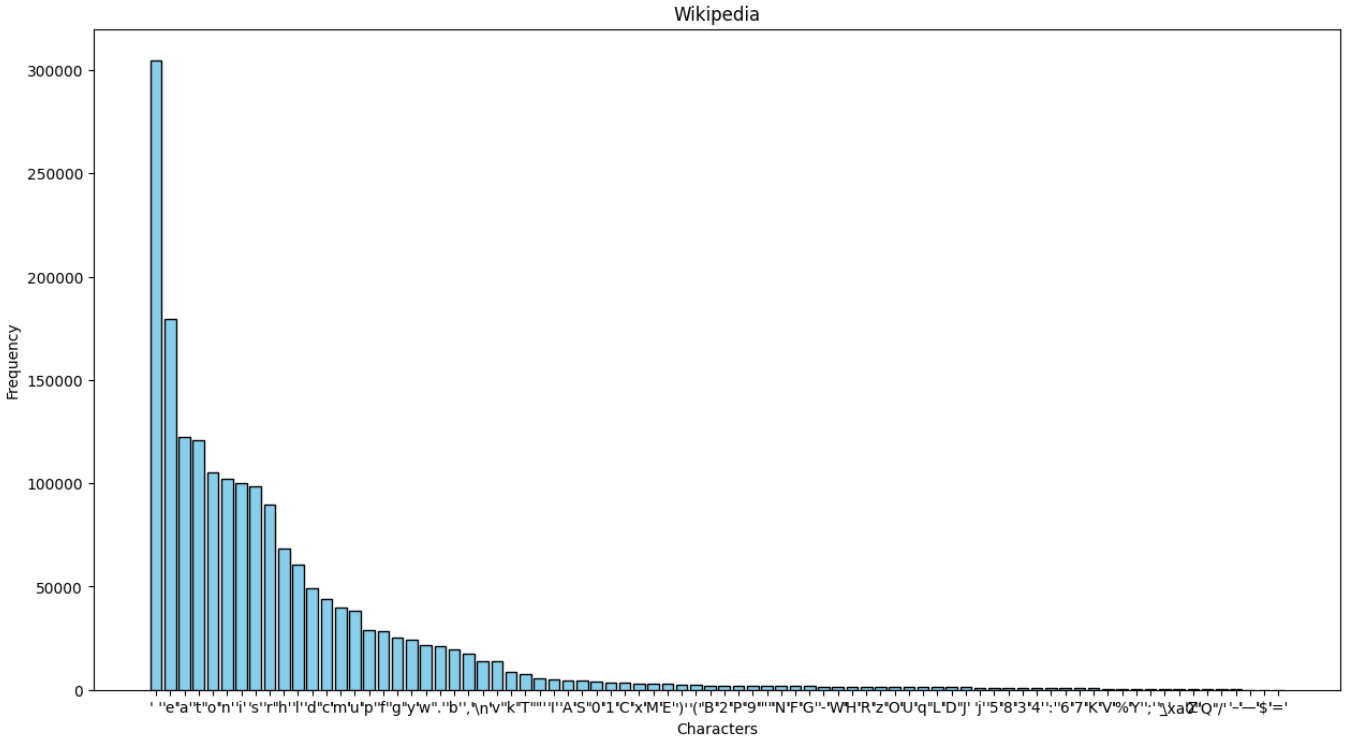

FIG. 2: Frequency distribution of wikipedia dataset (top 80)

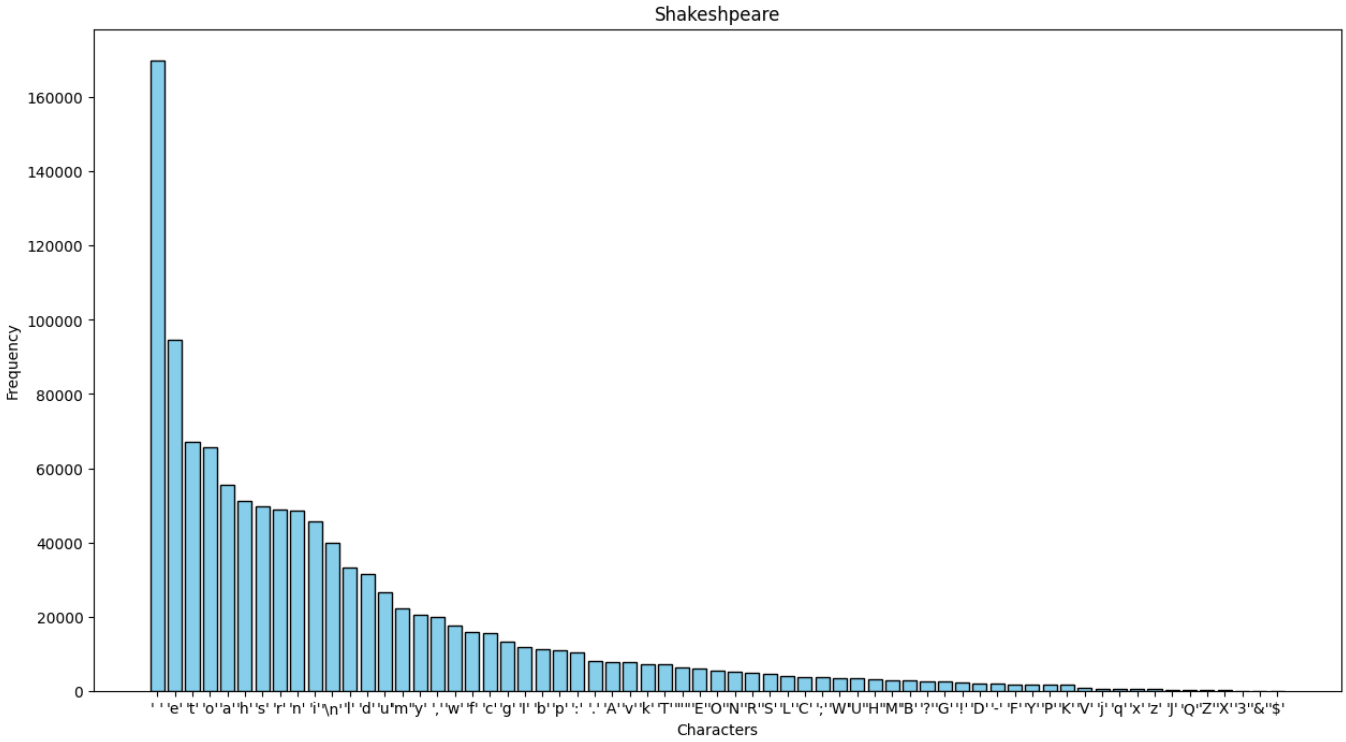

FIG. 3: Frequency distribution of shakespeare dataset (top 80)

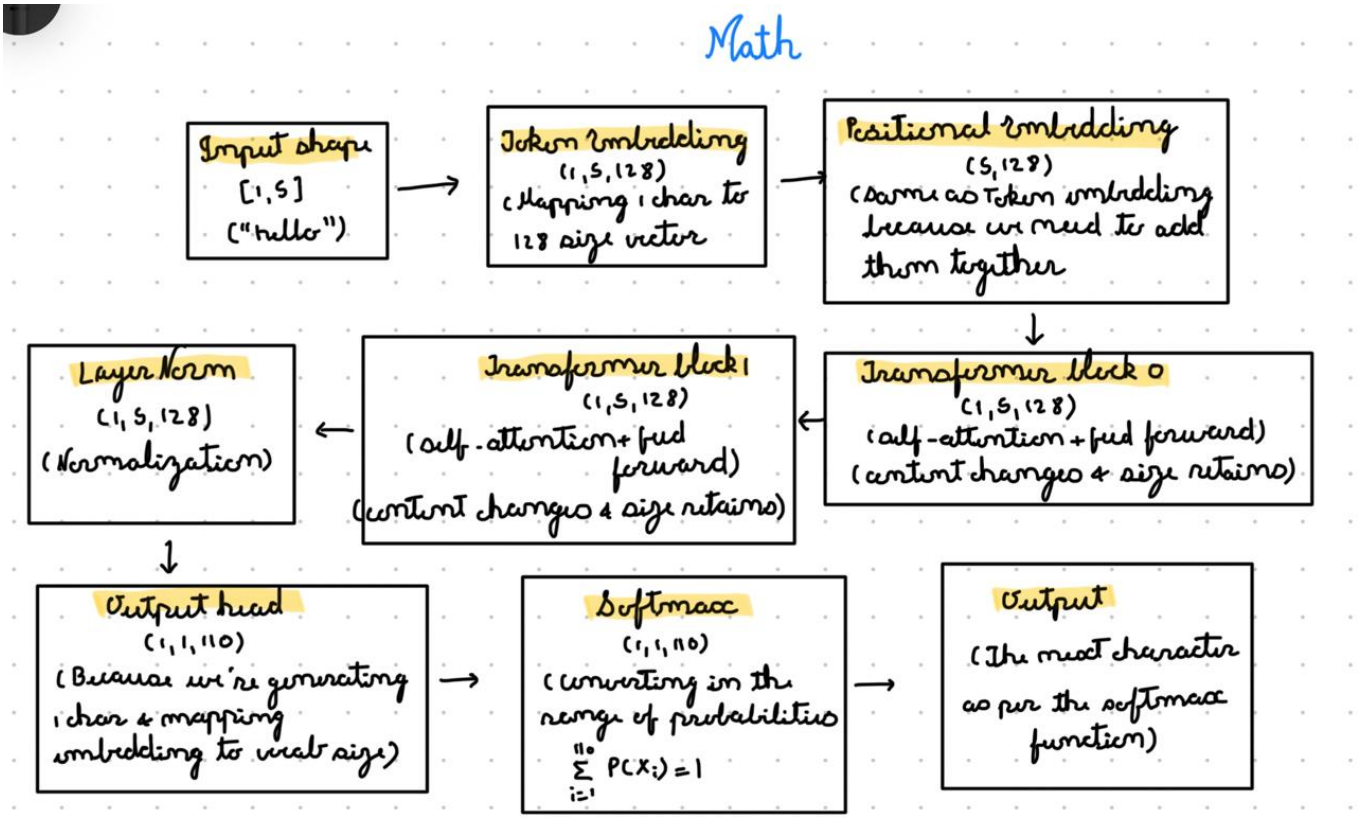

FIG. 4: Flow Chart of the model for math dataset

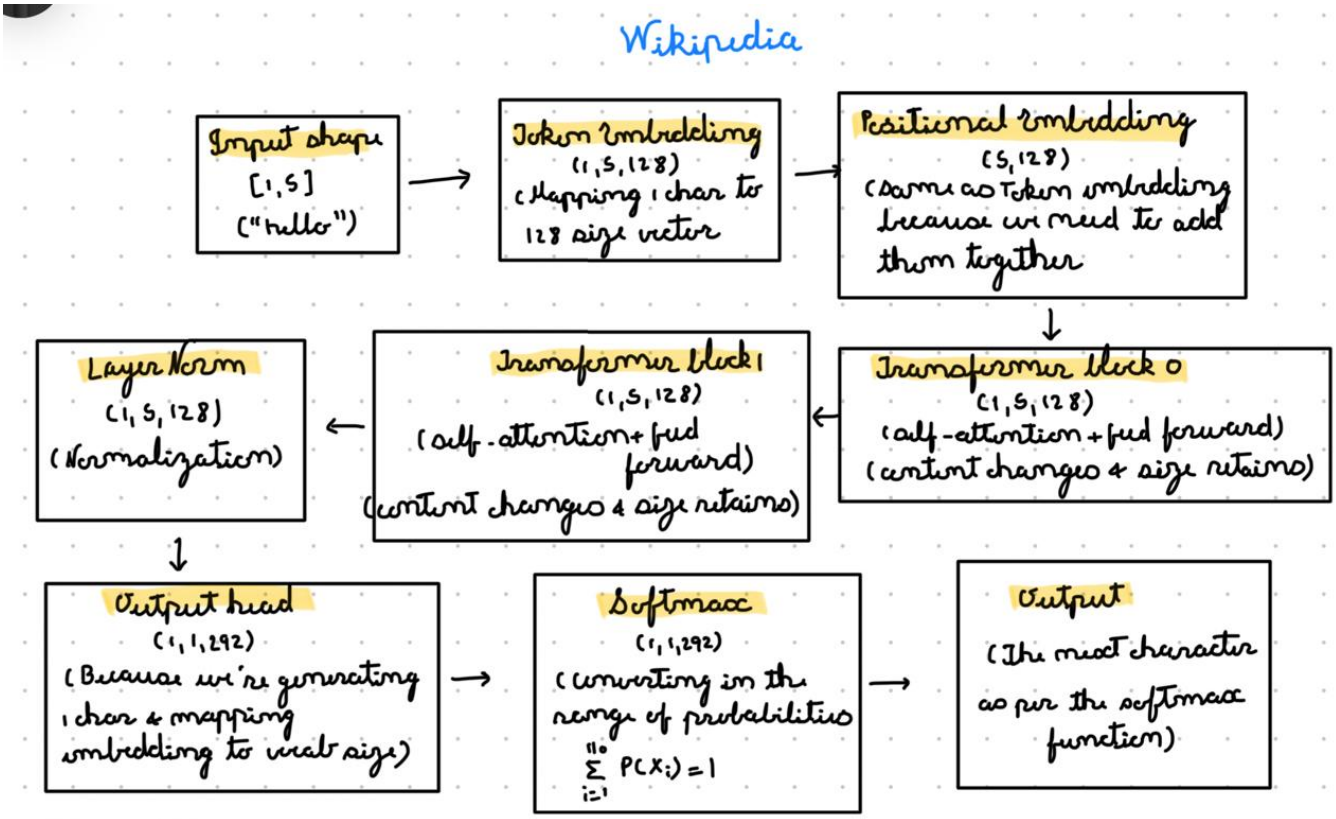

FIG. 5: Flow Chart of the model for wikipedia dataset

VI. ZERO-SHOT EVALUATION

Table I shows the $3x3$ cross domain loss matrix, where the diagonal elements are the individual domain loss value after 5000 iterations.

VII. FEW-SHOT EVALUATION

Fig. 10, 11, 12, 13, 14, 15 shows the loss curve when model A is trained on few instances of model B. Every curve shows a slight decrease in the value upon being trained from different dataset, which is a good thing, but the drop in the loss value is not enough to produce something really meaninglful. Fig. 16 shows the execution cell.

VIII. FEW-SHOT FROM EARLIER CHECKPOINTS

Fig. 17, 18, 19, 20, 21, 22 shows the loss curve when model A is trained on few instances of model B. This time, instead of taking 5000th iter of model A, we take 4000th iter of model A and then train it on model B for 200 iter. As per the observation, for this lab, this technique is not giving a better results than the norma few-shot technique.

IX. INSPECTING THE SOFTMAX OUTPUT LAYER

Here first we are choosing a prompt: biology, math, natural. After choosing a prompt, we can generate a character for which we get a softmax function. That softmax function gives us the value of entropy for the given character and iteration. Fig. 23, 24, 25, 26, 27 shows the softmax distribution for 5 different iterations and for 3 sequential character generation. Here we are displaying the top 7 softmax values. Fig. 28 shows the average entropy across all 5000 iter.

Likewise we can feed biology and natural prompt to math model. We can repeat the same process for wikipedia and shakespeare model. All the files are available at: Link.

Observation: Here we can see that as the number of iter increases, the softmax distribution becomes sharper and sharper. That means the level of uncertainty decreases as iter increases. And as the softmax distribution becomes sharper then value of entropy also decreases, which again shows the increasing confidence of the model over iter.

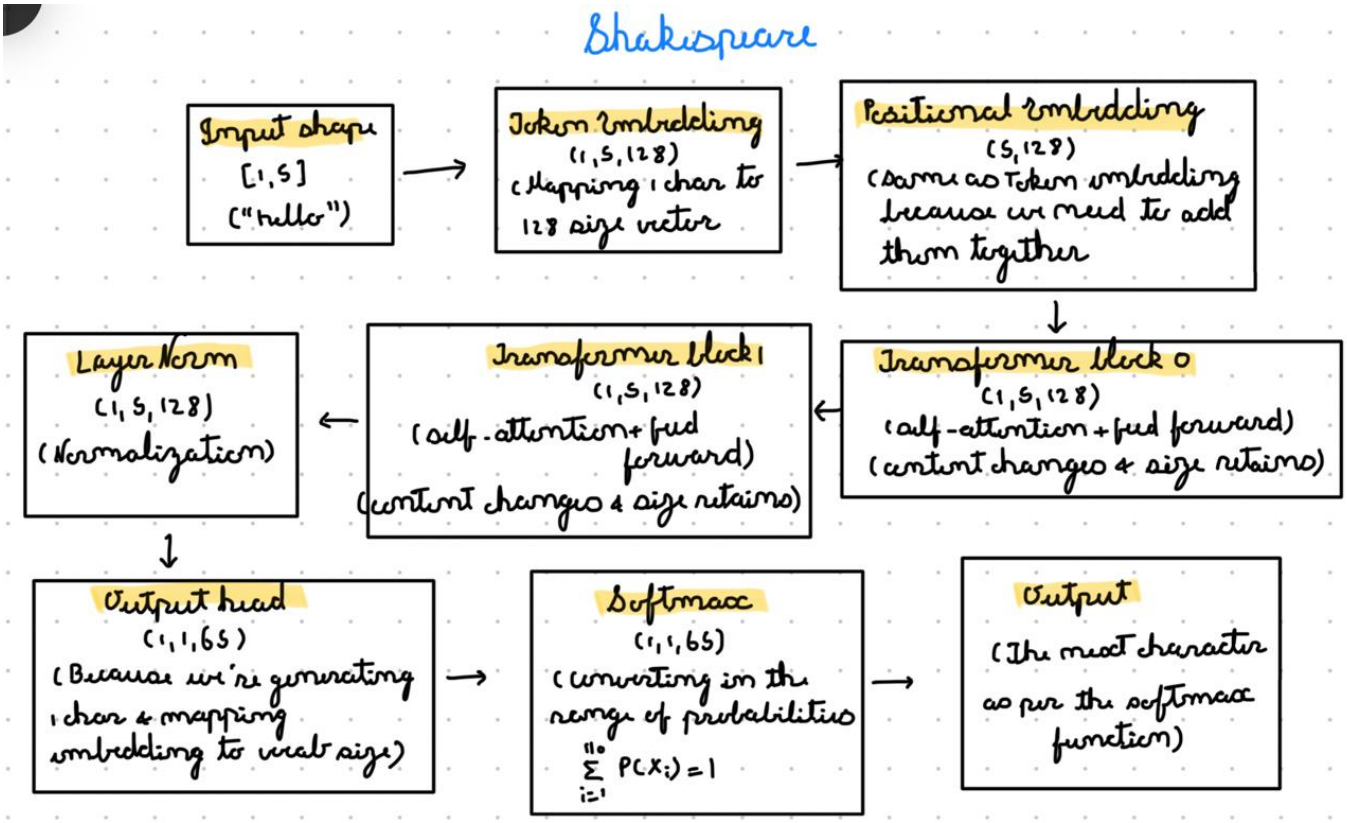

FIG. 6: Flow Chart of the model for shakespeare dataset

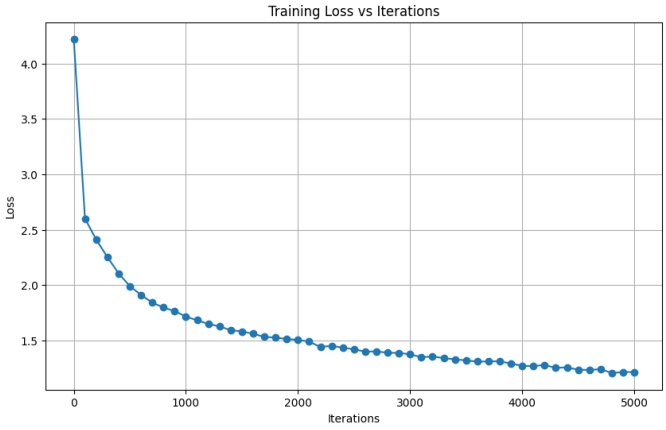

FIG. 7: Loss of the model for math dataset

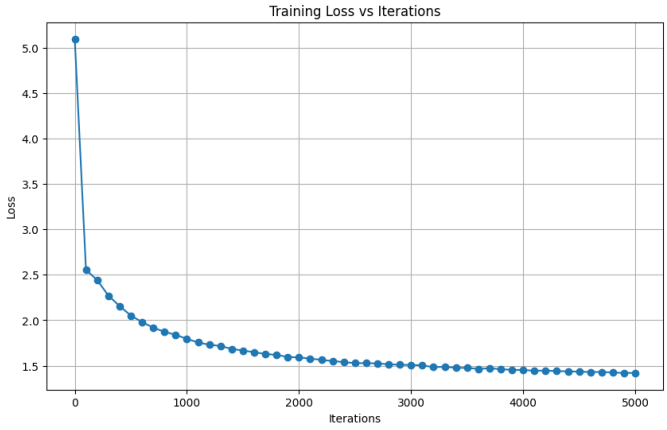

FIG. 8: Loss of the model for wikipedia dataset

X. GRAD-CAM FOR LANGUAGE MODELS

Here we can feed a prompt: math, biology, natural to any of the 3 datasets. Then we can give a target character, for which the function can give us a graph which shows the specific part of the prompt, that played a significant role for that target character to occur. Fig. 29 is an example of this.

My prompt examples throughout the lab:
prompt_math = "The integral of x^2 is"
prompt_biology = "In biology, mitochondria are"
prompt_natural = "Once upon a time"
target_token_math = "x"
target_token_biology = "p"
target_token_natural = "a"

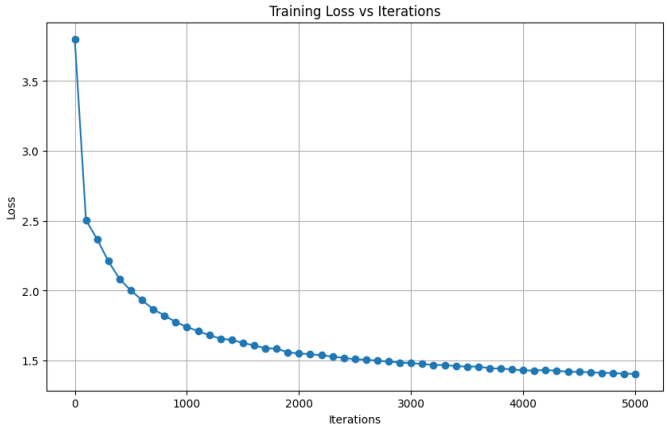

FIG. 9: Loss of the model for shakespeare dataset

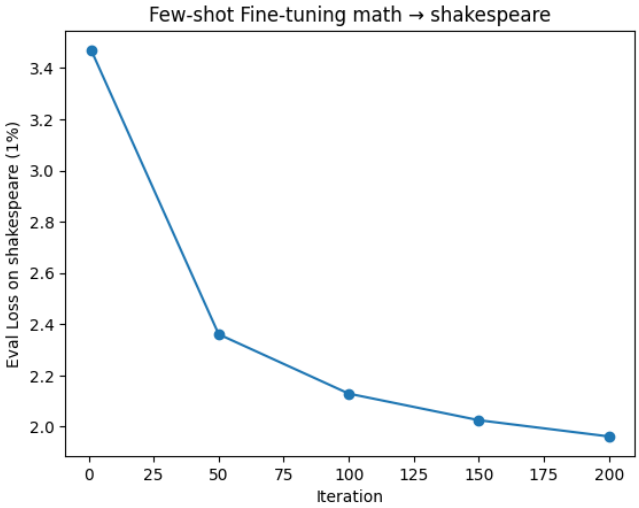

FIG. 10: Loss of the math model trained on Shakespeare dataset for 200 iterations.

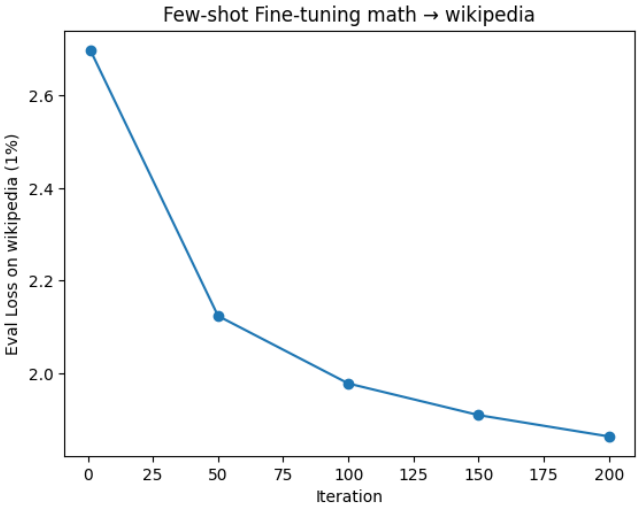

FIG. 11: Loss of the math model trained on Wikipedia dataset for 200 iterations.

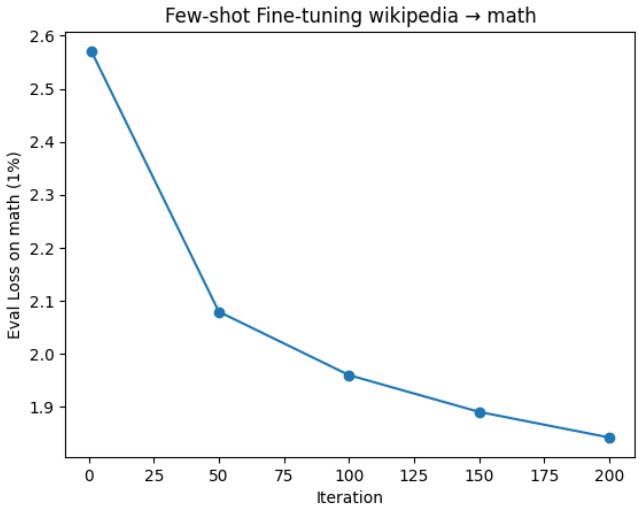

FIG. 12: Loss of the Wikipedia model trained on Math dataset for 200 iterations.

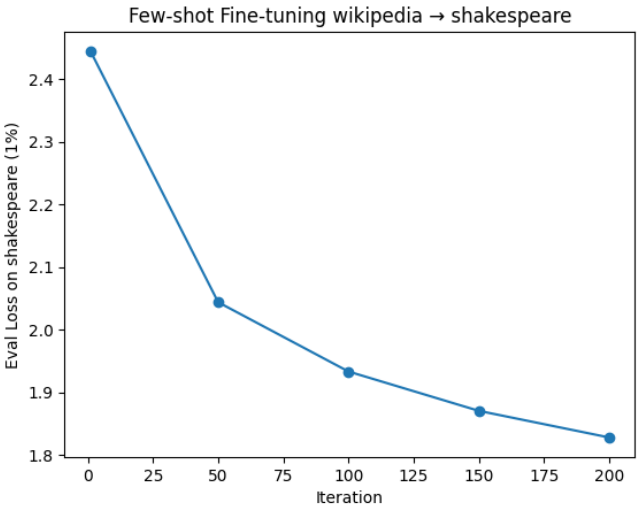

FIG. 13: Loss of the Wikipedia model trained on Shakespeare dataset for 200 iterations.

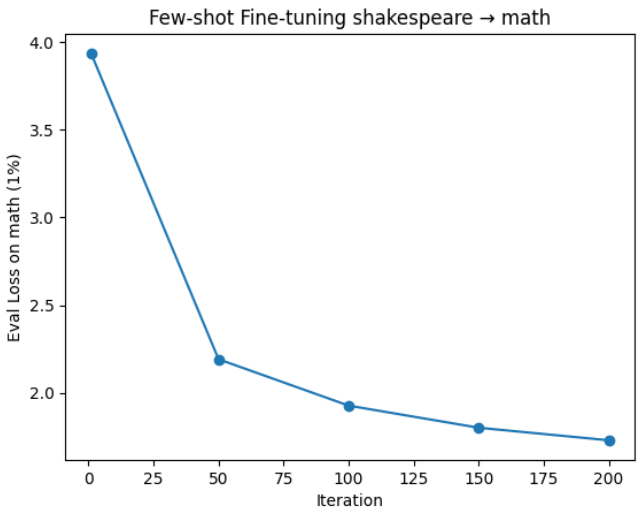

FIG. 14: Loss of the Shakespeare model trained on Math dataset for 200 iterations.

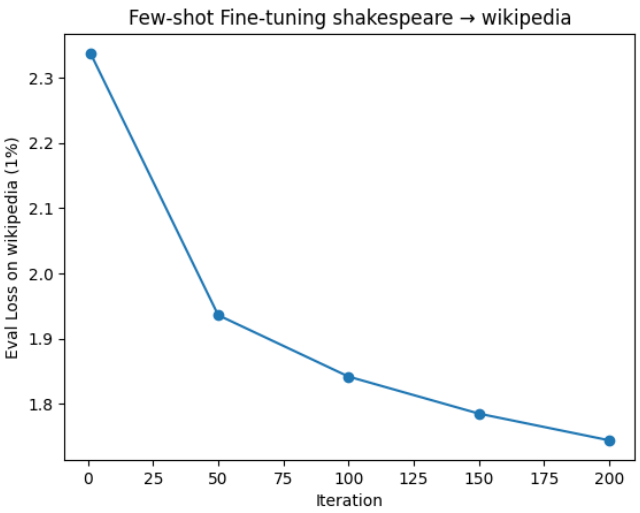

FIG. 15: Loss of the Shakespeare model trained on Wikipedia dataset for 200 iterations.

```
number of parameters: 0.41M
  0%|          | 0/200 [00:00<?, ?it/s]Step 1: eval loss = 2.3374
  1%|          | 2/200 [00:27<37:26, 11.35s/it]  Sample @ step 1:
The theorem statesman: this she love, you come, I thank you are loved;
I am art ell accompanied himselves to heaven wi

 24%|██        | 49/200 [00:33<00:23,  6.50it/s]Step 50: eval loss = 1.9361
 26%|██        | 51/200 [01:00<14:02,  5.65s/it]Sample @ step 50:
The theorem states and deserve a fortuned that words the compossitinged sends the enemy be well as the sister the brai

 50%|█████     | 99/200 [01:07<00:12,  7.78it/s]Step 100: eval loss = 1.8420
 50%|█████     | 101/200 [01:34<09:26,  5.72s/it]Sample @ step 100:
The theorem states and it carled the mat-called falls and general if the gentured Grance.
'Country, which live is doi

 74%|███████   | 149/200 [01:40<00:06,  7.66it/s]Step 150: eval loss = 1.7852
 76%|███████   | 151/200 [02:07<04:40,  5.73s/it]Sample @ step 150:
The theorem states are rice or genereting rims in such or more oreuple fora salined to a sorther in sabil.

Force wher

100%|██████████| 199/200 [02:14<00:00,  7.66it/s]Step 200: eval loss = 1.7443
100%|██████████| 200/200 [02:41<00:00,  1.24it/s]Sample @ step 200:
The theorem states and conject it
forse mean a ni
 proclorian to negly ascall it. At the soveraphes that the bagges th
```

FIG. 16: Example of how the output of few-shot training looks like

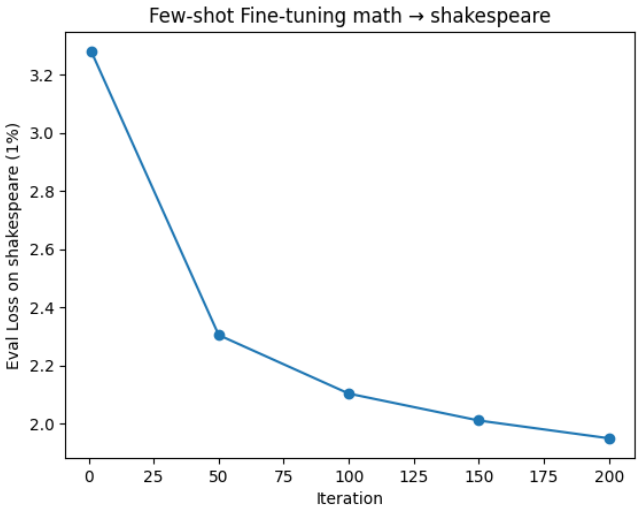

FIG. 17: Loss of the Math model (from 4000th iteration) trained on Shakespeare dataset for 200 iterations.

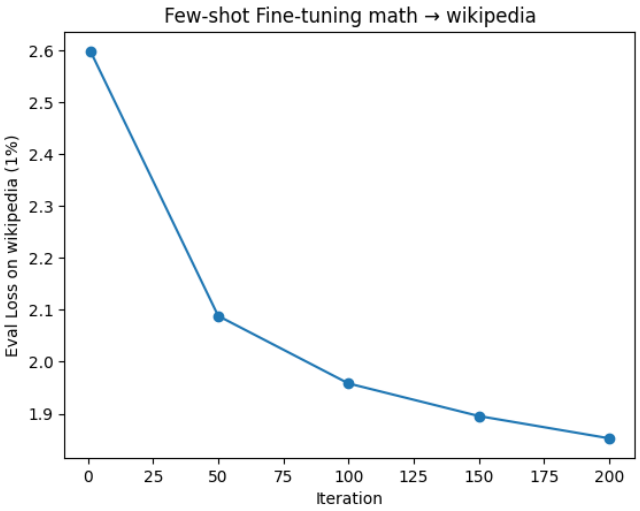

FIG. 18: Loss of the Math model (from 4000th iteration) trained on Wikipedia dataset for 200 iterations.

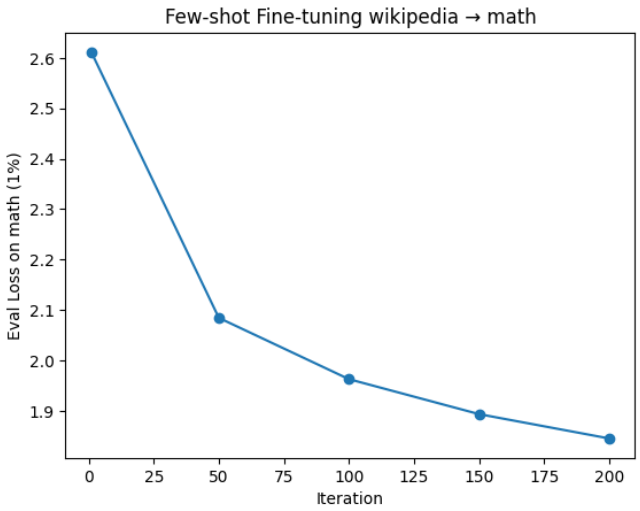

FIG. 19: Loss of the Wikipedia model (from 4000th iteration) trained on Math dataset for 200 iterations.

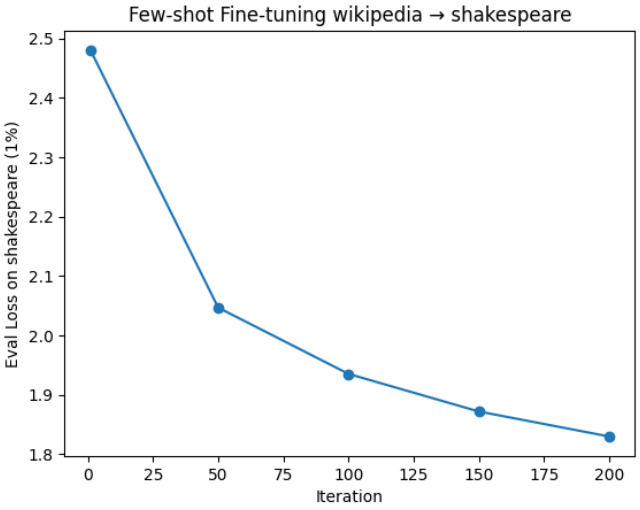

FIG. 20: Loss of the Wikipedia model (from 4000th iteration) trained on Shakespeare dataset for 200 iterations.

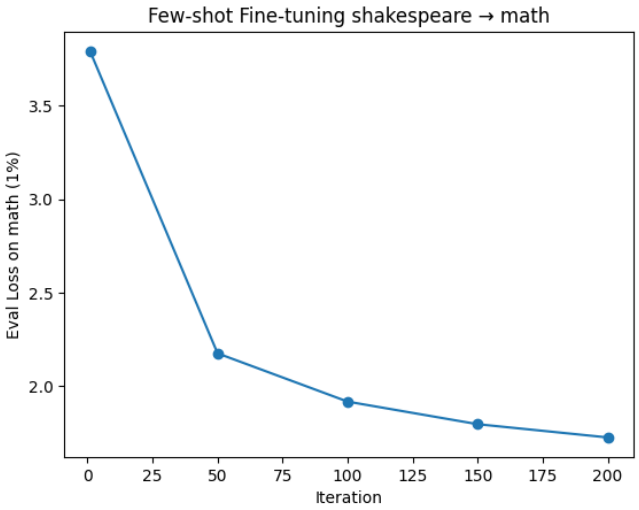

FIG. 21: Loss of the Shakespeare model (from 4000th iteration) trained on Math dataset for 200 iterations.

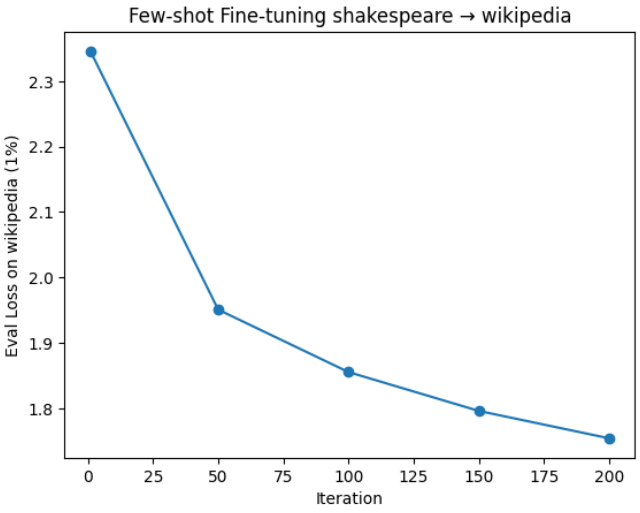

FIG. 22: Loss of the Shakespeare model (from 4000th iteration) trained on Wikipedia dataset for 200 iterations.

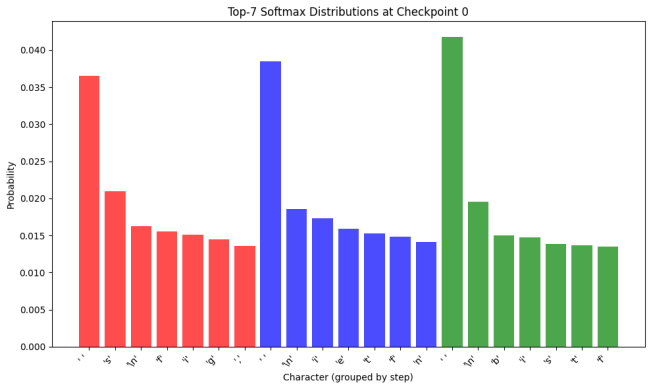

FIG. 23: Math dataset - Math prompt.

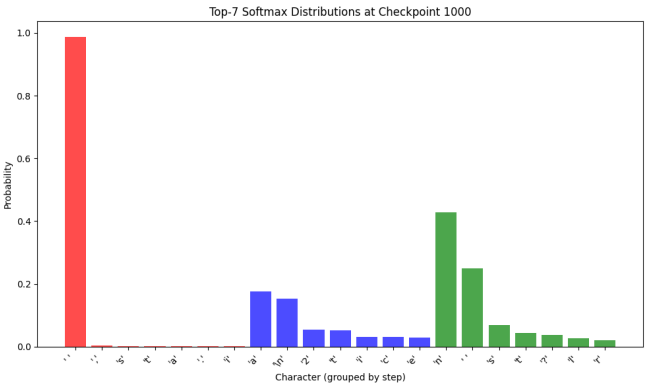

FIG. 24: Math dataset - Math prompt.

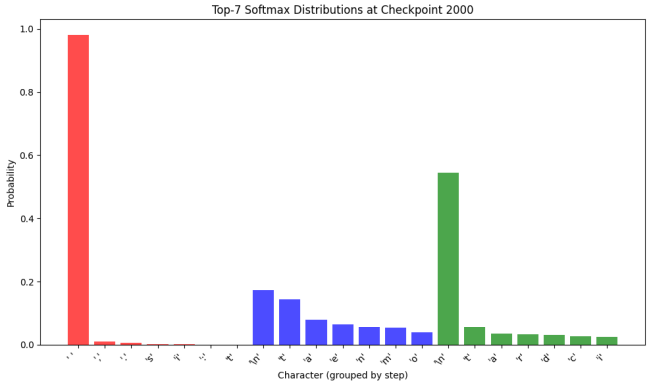

FIG. 25: Math dataset - Math prompt.

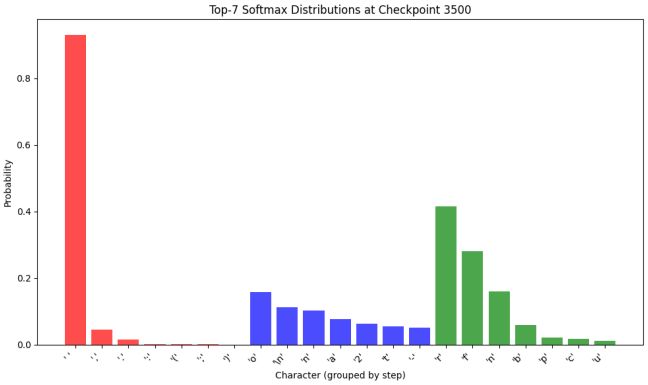

FIG. 26: Math dataset - Math prompt.

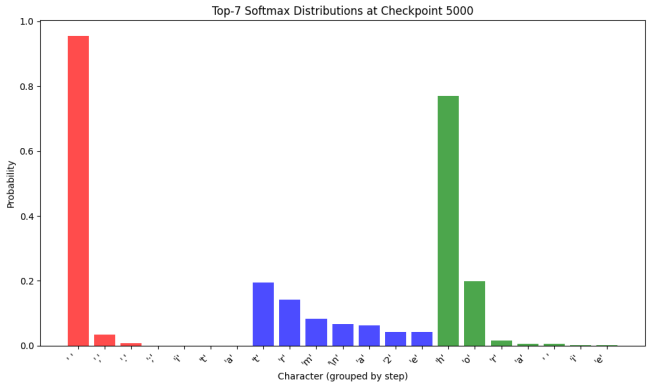

FIG. 27: Math dataset - Math prompt.

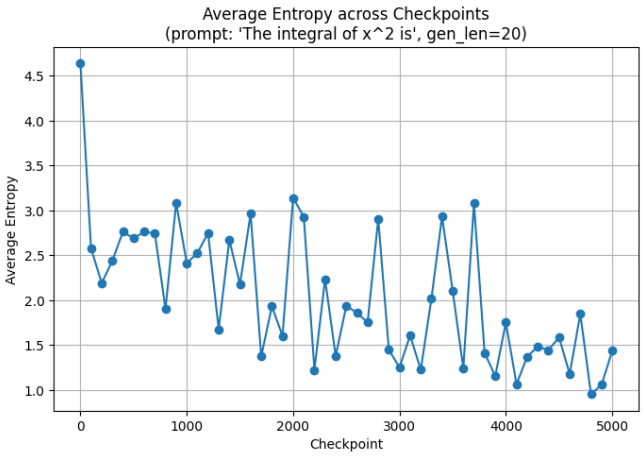

FIG. 28: Math dataset - Math prompt.

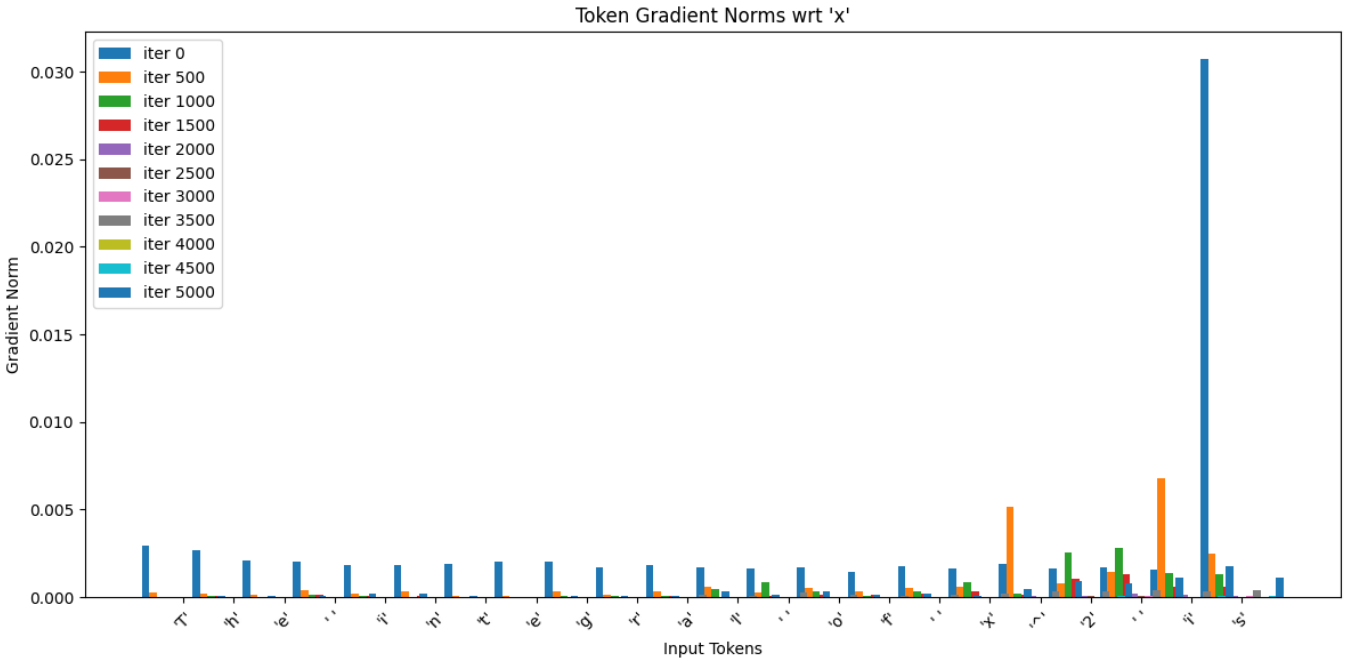

FIG. 29: math dataset - math prompt