ChatFlix

October 20, 2024

1 Load the data files from the Cornell Movie Dialog Corpus

https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
import os

# Specify the path to the dataset files (adjust if needed)
data_dir = r'/content/drive/MyDrive/Colab Notebooks/archive'

# Files we need from the corpus
lines_file = os.path.join(data_dir, 'movie_lines.txt')
conversations_file = os.path.join(data_dir, 'movie_conversations.txt')

# Check if the files exist
if os.path.exists(lines_file) and os.path.exists(conversations_file):
    print("Dataset files loaded successfully.")
else:
    print("Dataset files are missing. Please download and provide the correct␣
 ↪paths.")
```

```
Dataset files loaded successfully.
```

2 Read the lines from the movies

From README.txt

movie_lines.txt - contains the actual text of each utterance - fields: - lineID - characterID (who uttered this phrase) - movieID - character name - text of the utterance

Example: L868 +++++ +$u2$ +++++ m0 +++++ +$CAMERON$ +++++ The "real you".

[ ]:
```
# Load the movie lines from the movie_lines file and create a dictionary
def load_lines(file_path):
    # Dictionary of Lines with Line ID and it's corresponding text
    lines = {}
```

```
    # with open(file_path, 'r', encoding='iso-8859-1') as f:
    with open(file_path, 'r', encoding='utf-8', errors='ignore') as f:
        for line in f:
            # Movie lines are in this format: LineID + Speaker + MovieID + Character + Text
            parts = line.strip().split(" +++$+++ ")
            # Perform an explicit check to ensure that the length of parts is exactly 5 before mapping the movie lines.
            # Avoid issues if any lines are malformed.
            if len(parts) == 5:
                line_id = parts[0]      #Line Id
                text = parts[4]         # Movie Line
                lines[line_id] = text   #Populate dictionary.
    return lines

# Load movie lines
lines = load_lines(lines_file)
print(f"Loaded {len(lines)} lines from the dataset.")
```

```
Loaded 304446 lines from the dataset.
```

[ ]:
```
# Access the first element using list(lines.keys())[0] or lines.get(list(lines.keys())[0])
# list(lines.keys())[0] will retrieve the first key.

first_line_id = list(lines.keys())[0]
print(f"The first line in the dataset is: {lines[first_line_id]}")
print(f"The first line in the dataset is: {lines.get(first_line_id)}")

# Iterate over the first 15 items
for i, (line_id, text) in enumerate(lines.items()):
    if i < 15:
        print(f"Line ID: {line_id}, Text: {text}")
    else:
        break  # Stop after 15 iterations
```

```
The first line in the dataset is: They do not!
The first line in the dataset is: They do not!
Line ID: L1045, Text: They do not!
Line ID: L1044, Text: They do to!
Line ID: L985, Text: I hope so.
Line ID: L984, Text: She okay?
Line ID: L925, Text: Let's go.
Line ID: L924, Text: Wow
Line ID: L872, Text: Okay -- you're gonna need to learn how to lie.
Line ID: L871, Text: No
Line ID: L870, Text: I'm kidding.  You know how sometimes you just become this
```

```
"persona"?  And you don't know how to quit?
Line ID: L869, Text: Like my fear of wearing pastels?
Line ID: L868, Text: The "real you".
Line ID: L867, Text: What good stuff?
Line ID: L866, Text: I figured you'd get to the good stuff eventually.
Line ID: L865, Text: Thank God!  If I had to hear one more story about your
coiffure…
Line ID: L864, Text: Me.  This endless …blonde babble. I'm like, boring
myself.
```

3 Read the conversations from the movies

```
Check to see if these conversations' lines are read within movie_lines.
If so, keep it.
```

- movie_conversations.txt
  - the structure of the conversations
  - fields
    - characterID of the first character involved in the conversation
    - characterID of the second character involved in the conversation
    - movieID of the movie in which the conversation occurred
    - list of the utterances that make the conversation, in chronological order: ['lineID1','lineID2',É,'lineIDN'] has to be matched with movie_lines.txt to reconstruct the actual content

Example: u0 +++ + +$u2$ + +++++ m0 +++$+++ ['L404', 'L405', 'L406', 'L407']

[ ]:
```python
#Load the movie conversations
def load_conversations(file_path, lines):
    conversations = []

    with open(file_path, 'r', encoding='utf-8', errors='ignore') as f:
        for line in f:
            # Movie conversations are in this format: Character1 + Character2 + MovieID + List of LineIDs
            parts = line.strip().split(" +++$+++ ")
            #retrieves line id's
            if len(parts) == 4:
                # Extract the list of line IDs and convert to text
                line_ids = eval(parts[3])  # eval to convert the string list to a list object
                conv = [lines[line_id] for line_id in line_ids if line_id in lines] # if line id matches entry in lines dictionary, keep it
                conversations.append(conv)
    return conversations

# Load conversations
conversations = load_conversations(conversations_file, lines)
```

```
print(f"Loaded {len(conversations)} conversations.")

#print(conversations)
```

```
Loaded 83097 conversations.
```

[ ]:
```
# Iterate over the first 15 items
for i, (text) in enumerate(conversations):
    if i < 15:
        print(f" Text: {text}")
    else:
        break  # Stop after 15 iterations
```

```
 Text: ['Can we make this quick?  Roxanne Korrine and Andrew Barrett are having
an incredibly horrendous public break- up on the quad.  Again.', "Well, I
thought we'd start with pronunciation, if that's okay with you.", 'Not the
hacking and gagging and spitting part.  Please.', "Okay… then how 'bout we try
out some French cuisine.  Saturday?  Night?"]
 Text: ["You're asking me out.  That's so cute. What's your name again?",
'Forget it.']
 Text: ["No, no, it's my fault -- we didn't have a proper introduction ---",
'Cameron.', "The thing is, Cameron -- I'm at the mercy of a particularly hideous
breed of loser.  My sister.  I can't date until she does.", 'Seems like she
could get a date easy enough…']
 Text: ['Why?', 'Unsolved mystery.  She used to be really popular when she
started high school, then it was just like she got sick of it or something.',
"That's a shame."]
 Text: ['Gosh, if only we could find Kat a boyfriend…', 'Let me see what I can
do.']
 Text: ["C'esc ma tete. This is my head", "Right.  See?  You're ready for the
quiz.", "I don't want to know how to say that though.  I want to know useful
things. Like where the good stores are.  How much does champagne cost?  Stuff
like Chat.  I have never in my life had to point out my head to someone.",
"That's because it's such a nice one.", 'Forget French.']
 Text: ['How is our little Find the Wench A Date plan progressing?', "Well,
there's someone I think might be --"]
 Text: ['There.', 'Where?']
 Text: ['You got something on your mind?', "I counted on you to help my cause.
You and that thug are obviously failing. Aren't we ever going on our date?"]
 Text: ['You have my word.  As a gentleman', "You're sweet."]
 Text: ['How do you get your hair to look like that?', "Eber's Deep Conditioner
every two days. And I never, ever use a blowdryer without the diffuser
attachment."]
 Text: ['Sure have.', "I really, really, really wanna go, but I can't.  Not
unless my sister goes.", "I'm workin' on it. But she doesn't seem to be goin'
for him."]
 Text: ["She's not a…", "Lesbian?  No. I found a picture of Jared Leto in one
of her drawers, so I'm pretty sure she's not harboring same-sex tendencies.",
```

```
"So that's the kind of guy she likes? Pretty ones?", "Who knows?  All I've ever
heard her say is that she'd dip before dating a guy that smokes."]
 Text: ['Hi.', 'Looks like things worked out tonight, huh?']
 Text: ['You know Chastity?', 'I believe we share an art instructor']
```

4 Exploratory Data analysis

5 Text Cleaning

[ ]:
```
!pip install nltk contractions emoji transformers datasets Counter evaluate␣
↪Rouge  imbalanced-learn torch
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages
(3.8.1)
Collecting contractions
  Downloading contractions-0.1.73-py2.py3-none-any.whl.metadata (1.2 kB)
Collecting emoji
  Downloading emoji-2.14.0-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-
packages (4.44.2)
Collecting datasets
  Downloading datasets-3.0.1-py3-none-any.whl.metadata (20 kB)
Collecting Counter
  Downloading Counter-1.0.0.tar.gz (5.2 kB)
  Preparing metadata (setup.py) … done
Collecting evaluate
  Downloading evaluate-0.4.3-py3-none-any.whl.metadata (9.2 kB)
Collecting Rouge
  Downloading rouge-1.0.1-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: imbalanced-learn in
/usr/local/lib/python3.10/dist-packages (0.12.4)
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages
(2.4.1+cu121)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages
(from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages
(from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.10/dist-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from nltk) (4.66.5)
Collecting textsearch>=0.0.21 (from contractions)
  Downloading textsearch-0.0.24-py2.py3-none-any.whl.metadata (1.2 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.24.7)
```

```
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-
packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from transformers) (6.0.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from transformers) (2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.4.5)
Requirement already satisfied: tokenizers<0.20,>=0.19 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.10/dist-packages (from datasets) (16.1.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(from datasets) (2.2.2)
Collecting xxhash (from datasets)
  Downloading
xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(12 kB)
Collecting multiprocess (from datasets)
  Downloading multiprocess-0.70.17-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: fsspec<=2024.6.1,>=2023.1.0 in
/usr/local/lib/python3.10/dist-packages (from
fsspec[http]<=2024.6.1,>=2023.1.0->datasets) (2024.6.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-
packages (from datasets) (3.10.10)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from Rouge) (1.16.0)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-
packages (from imbalanced-learn) (1.13.1)
Requirement already satisfied: scikit-learn>=1.0.2 in
/usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.5.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (3.5.0)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch) (4.12.2)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages
(from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from torch) (3.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from torch) (3.1.4)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.4.3)
Requirement already satisfied: aiosignal>=1.1.2 in
```

```
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp->datasets) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.12.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.14.0)
Requirement already satisfied: async-timeout<5.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2024.8.30)
Collecting anyascii (from textsearch>=0.0.21->contractions)
  Downloading anyascii-0.3.2-py3-none-any.whl.metadata (1.5 kB)
Collecting pyahocorasick (from textsearch>=0.0.21->contractions)
  Downloading pyahocorasick-2.1.0-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl.metadata (13 kB)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch) (3.0.1)
INFO: pip is looking at multiple versions of multiprocess to determine which
version is compatible with other requirements. This could take a while.
Collecting multiprocess (from datasets)
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->datasets) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas->datasets) (2024.2)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch) (1.3.0)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from
yarl<2.0,>=1.12.0->aiohttp->datasets) (0.2.0)
Downloading contractions-0.1.73-py2.py3-none-any.whl (8.7 kB)
Downloading emoji-2.14.0-py3-none-any.whl (586 kB)
                        586.9/586.9 kB
12.3 MB/s eta 0:00:00
Downloading datasets-3.0.1-py3-none-any.whl (471 kB)
```

```
                         471.6/471.6 kB
38.2 MB/s eta 0:00:00
Downloading evaluate-0.4.3-py3-none-any.whl (84 kB)
                         84.0/84.0 kB
8.2 MB/s eta 0:00:00
Downloading rouge-1.0.1-py3-none-any.whl (13 kB)
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
                         116.3/116.3 kB
12.8 MB/s eta 0:00:00
Downloading textsearch-0.0.24-py2.py3-none-any.whl (7.6 kB)
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
                         134.8/134.8 kB
14.2 MB/s eta 0:00:00
Downloading
xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
                         194.1/194.1 kB
17.5 MB/s eta 0:00:00
Downloading anyascii-0.3.2-py3-none-any.whl (289 kB)
                         289.9/289.9 kB
27.0 MB/s eta 0:00:00
Downloading pyahocorasick-2.1.0-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (110 kB)
                         110.7/110.7 kB
11.2 MB/s eta 0:00:00
Building wheels for collected packages: Counter
  Building wheel for Counter (setup.py) … done
  Created wheel for Counter: filename=Counter-1.0.0-py3-none-any.whl size=5393
sha256=d8126f692071bd9847886a8529c2354a028cd50af9c1d3a61cb5764a3e4bc838
  Stored in directory: /root/.cache/pip/wheels/e3/02/6d/d5c0838427a060718c6060ae
4d24da95a0e0df0d7a3dab8040
Successfully built Counter
Installing collected packages: Counter, xxhash, Rouge, pyahocorasick, emoji,
dill, anyascii, textsearch, multiprocess, contractions, datasets, evaluate
Successfully installed Counter-1.0.0 Rouge-1.0.1 anyascii-0.3.2
contractions-0.1.73 datasets-3.0.1 dill-0.3.8 emoji-2.14.0 evaluate-0.4.3
multiprocess-0.70.16 pyahocorasick-2.1.0 textsearch-0.0.24 xxhash-3.5.0
```

[ ]:
```python
import re
import nltk
import contractions # Removing contractions
import emoji # Convert Emoticons to Text if you want to perform sentiment␣
 ↪analysis.
import unicodedata

nltk.download('words')
words = set(nltk.corpus.words.words())
```

```
# Normalize string by converting unicode characters to ASCII and removing␣
 ↪non-letters
def unicode_to_ascii(s):
    return ''.join(c for c in unicodedata.normalize('NFD', s)
                   if unicodedata.category(c) != 'Mn')

# Function to convert emojis to words using emoji library mapping
def convert_emojis_to_words(text):
    converted_text = emoji.demojize(text)
    return converted_text

def remove_email(text):
    return re.sub(r'([a-z0-9+._-]+@[a-z0-9+_-]+)', "", text)

def remove_emojis(text):
    # Regular expression pattern to match emojis
    emoji_pattern = re.compile(
        "["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F700-\U0001F77F"  # alchemical symbols
        u"\U0001F780-\U0001F7FF"  # Geometric Shapes Extended
        u"\U0001F800-\U0001F8FF"  # Supplemental Arrows-C
        u"\U0001F900-\U0001F9FF"  # Supplemental Symbols and Pictographs
        u"\U0001FA00-\U0001FA6F"  # Chess Symbols
        u"\U0001FA70-\U0001FAFF"  # Symbols and Pictographs Extended-A
        u"\U00002702-\U000027B0"  # Dingbats
        u"\U000024C2-\U0001F251"  # Enclosed characters
        "]+", flags=re.UNICODE
    )
    return emoji_pattern.sub(r'', text)

# Clitics
def clean_clitics(text):
    text = text.lower()
    text = re.sub(r"i'm", "i am", text)
    text = re.sub(r"he's", "he is", text)
    text = re.sub(r"she's", "she is", text)
    text = re.sub(r"that's", "that is", text)
    text = re.sub(r"what's", "what is", text)
    text = re.sub(r"where's", "where is", text)
    text = re.sub(r"how's", "how is", text)
    text = re.sub(r"\'ll", " will", text)
    text = re.sub(r"\'ve", " have", text)
    text = re.sub(r"\'re", " are", text)
```

```
    text = re.sub(r"\'d", " would", text)
    text = re.sub(r"n't", " not", text)
    text = re.sub(r"won't", "will not", text)
    text = re.sub(r"can't", "cannot", text)
    #text = re.sub(r"[-()\"#/@;:<>{}`+=~|.!?,]", "", text)
    return text

def remove_digitsInText(text):
    textWithoutDigits = list(filter(lambda x: x.isalpha(), text))
    return textWithoutDigits

def clean_text(text):
    # Step 1: Remove URLs
    text = re.sub(r'http\S+|www\S+', '', text)

    #Step 2: Remove emails
    text = remove_email(text)

    # Step 3: Remove Hashtags
    text = re.sub(r'#\w+', '', text)

    # Step 4: Remove Usernames (assuming they start with '@')
    text = re.sub(r'@\w+', '', text)

    # Step 5: Convert emojis to words
    text = convert_emojis_to_words(text)

    #Step 6: Remove any emoticons or emojis
    text = remove_emojis(text)

    # Step 7: Normalize Unicode characters to ASCII
    text = unicode_to_ascii(text.lower().strip())

    # Step 10: Convert to lowercase
    text = text.lower()

    # Step 11: Handle contractions / clitics
    text = contractions.fix(text)
    text = clean_clitics(text)

    # Step 8: Remove punctuation, numbers, and extra spaces
    # Step 9: Remove any special characters
    # Remove special characters but retain basic punctuation
    text = re.sub(r"[^A-Za-z0-9\s\.,!?]", "", text)  # Retain . , ! ?
    text = re.sub(r'\s+', ' ', text).strip()  # Remove extra spaces
    #text = re.sub(r"[^\w\s]", '', text) #Remove special characters
```

```
    #text = remove_digitsInText(text) # Remove digits in text , Remove any words with digits like 5pm

    # Step 12: Remove non-English words
    text = " ".join(w for w in nltk.wordpunct_tokenize(text) if w.lower() in words or not w.isalpha())

    return text
```

```
[nltk_data] Downloading package words to /root/nltk_data…
[nltk_data]   Unzipping corpora/words.zip.
```

[ ]:
```
# Load and clean the data
def clean_and_prepare_data(lines_file, conversations_file):

    questions = []
    answers = []
    for conversation in conversations:
        for i in range(len(conversation) - 1):
            questions.append(conversation[i])
            answers.append(conversation[i + 1])

    clean_questions = [clean_text(q) for q in questions]
    clean_answers = [clean_text(a) for a in answers]

    # Filtering out short or long questions/answers
    # Keep only those sentences that have between 2 and 25 words.
    filtered_questions, filtered_answers = [], []

    for q, a in zip(clean_questions, clean_answers):
        # Filter out sentences that are too short or too long
        # Append the <EOS> (End of String) token to the end of each answer, indicating the end of the response for the model.
        if 2 <= len(q.split()) <= 25 and 2 <= len(a.split()) <= 25:
            filtered_questions.append(q)
            filtered_answers.append(a + ' <EOS>')  # Append end token to answers

    # Print the first 10 filtered questions and answers
    print("\nFirst 10 Filtered Questions and Answers:")
    for i in range(min(10, len(filtered_questions))):  # Ensure we don't exceed the length of the lists
        print(f"Q{i+1}: {filtered_questions[i]}")
        print(f"A{i+1}: {filtered_answers[i]}")
        print()


    return filtered_questions, filtered_answers
```

[ ]:
```
# Load and clean the data
clean_questions, clean_answers = clean_and_prepare_data(lines_file,␣
 ↪conversations_file)

# Print out the number of cleaned questions and answers
print(f"Number of cleaned questions: {len(clean_questions)}")
print(f"Number of cleaned answers: {len(clean_answers)}")

# Print the first 10 filtered questions and answers
print("\nFirst 10 Filtered Questions and Answers:")
for i in range(min(10, len(clean_questions))):  # Ensure we don't exceed the␣
 ↪length of the lists
    print(f"Q{i+1}: {clean_questions[i]}")
    print(f"A{i+1}: {clean_answers[i]}")
    print()
```

```
First 10 Filtered Questions and Answers:
Q1: can we make this quick ? and are an incredibly horrendous public break up on
the quad . again .
A1: well , i thought we would start with pronunciation , if that is with you .
<EOS>

Q2: well , i thought we would start with pronunciation , if that is with you .
A2: not the hacking and and spitting part . please . <EOS>

Q3: not the hacking and and spitting part . please .
A3: … then how bout we try out some cuisine . ? night ? <EOS>

Q4: you are me out . that is so cute . what is your name again ?
A4: forget it . <EOS>

Q5: gosh , if only we could find kat a …
A5: let me see what i can do . <EOS>

Q6: ma tete . this is my head
A6: right . see ? you are ready for the quiz . <EOS>

Q7: that is because it is such a nice one .
A7: forget . <EOS>

Q8: how is our little find the wench a date plan ?
A8: well , there is someone i think might be <EOS>

Q9: there .
A9: where ? <EOS>

Q10: you got something on your mind ?
```

```
A10: i on you to help my because . you and that thug are obviously failing . are
not we ever going on our date ? <EOS>

Number of cleaned questions: 170214
Number of cleaned answers: 170214

First 10 Filtered Questions and Answers:
Q1: can we make this quick ? and are an incredibly horrendous public break up on
the quad . again .
A1: well , i thought we would start with pronunciation , if that is with you .
<EOS>

Q2: well , i thought we would start with pronunciation , if that is with you .
A2: not the hacking and and spitting part . please . <EOS>

Q3: not the hacking and and spitting part . please .
A3: … then how bout we try out some cuisine . ? night ? <EOS>

Q4: you are me out . that is so cute . what is your name again ?
A4: forget it . <EOS>

Q5: gosh , if only we could find kat a …
A5: let me see what i can do . <EOS>

Q6: ma tete . this is my head
A6: right . see ? you are ready for the quiz . <EOS>

Q7: that is because it is such a nice one .
A7: forget . <EOS>

Q8: how is our little find the wench a date plan ?
A8: well , there is someone i think might be <EOS>

Q9: there .
A9: where ? <EOS>

Q10: you got something on your mind ?
A10: i on you to help my because . you and that thug are obviously failing . are
not we ever going on our date ? <EOS>
```

[ ]:
```python
# Validate the questions and answers
def validate_questions_answers(questions, answers, min_len=2, max_len=25):
    valid_questions = []
    valid_answers = []
    invalid_count = 0
```

```
    for question, answer in zip(questions, answers):
        # Check for non-empty strings and length constraints
        if isinstance(question, str) and isinstance(answer, str) and min_len <=␣
 ↪len(question.split()) <= max_len and min_len <= len(answer.split()) <=␣
 ↪max_len:
            valid_questions.append(question)
            valid_answers.append(answer)
        else:
            invalid_count += 1  # Count invalid pairs

    print(f"Total valid questions: {len(valid_questions)}")
    print(f"Total valid answers: {len(valid_answers)}")
    print(f"Total invalid pairs: {invalid_count}")

    return valid_questions, valid_answers

# Run validation
validated_questions, validated_answers =␣
 ↪validate_questions_answers(clean_questions, clean_answers)
```

```
Total valid questions: 168457
Total valid answers: 168457
Total invalid pairs: 1757
```

[ ]:
```
# Print the first 10 filtered questions and answers
print("\nFirst 10 Filtered Questions and Answers:")
for i in range(min(10, len(validated_questions))):  # Ensure we don't exceed␣
 ↪the length of the lists
    print(f"Q{i+1}: {validated_questions[i]}")
    print(f"A{i+1}: {validated_answers[i]}")
    print()
```

```
First 10 Filtered Questions and Answers:
Q1: can we make this quick ? and are an incredibly horrendous public break up on
the quad . again .
A1: well , i thought we would start with pronunciation , if that is with you .
<EOS>

Q2: well , i thought we would start with pronunciation , if that is with you .
A2: not the hacking and and spitting part . please . <EOS>

Q3: not the hacking and and spitting part . please .
A3: … then how bout we try out some cuisine . ? night ? <EOS>

Q4: you are me out . that is so cute . what is your name again ?
A4: forget it . <EOS>
```

```
Q5: gosh , if only we could find kat a …
A5: let me see what i can do . <EOS>

Q6: ma tete . this is my head
A6: right . see ? you are ready for the quiz . <EOS>

Q7: that is because it is such a nice one .
A7: forget . <EOS>

Q8: how is our little find the wench a date plan ?
A8: well , there is someone i think might be <EOS>

Q9: there .
A9: where ? <EOS>

Q10: you have my word . as a gentleman
A10: you are sweet . <EOS>
```

6 Train, Test Split before training using Cornell Movie corpus data

Train Set: Use this to train the model.

Validation Set: Use this to fine tune the models hyperparameter and evaluate the model during training. Monitor overfitting during training.

Test Set: Use this post training to test how well the model generalizes or to see how the model performs on unseen data.

[ ]:
```python
from sklearn.model_selection import train_test_split

# Split data into train, validation, and test sets
def split_data(clean_questions, clean_answers, test_size=0.2, val_size=0.1):
    # First split into train and remaining (which will be split further)
    questions_train, questions_rem, answers_train, answers_rem =␣
 ↪train_test_split(
        clean_questions, clean_answers, test_size=(test_size + val_size),␣
 ↪random_state=42)

    # Then split the remaining into validation and test
    val_size_adjusted = val_size / (test_size + val_size)
    questions_val, questions_test, answers_val, answers_test = train_test_split(
        questions_rem, answers_rem, test_size=val_size_adjusted,␣
 ↪random_state=42)

    return (questions_train, answers_train), (questions_val, answers_val),␣
 ↪(questions_test, answers_test)
```

```
(train_questions, train_answers), (val_questions, val_answers),␣
 ↪(test_questions, test_answers) = split_data(validated_questions,␣
 ↪validated_answers)

# Output the sizes of each set
print(f"Training set size: {len(train_questions)}")
print(f"Validation set size: {len(val_questions)}")
print(f"Test set size: {len(test_questions)}")
```

```
Training set size: 117919
Validation set size: 33692
Test set size: 16846
```

7 Tokenization and Data Preparation

Use DialoGPT as the pre-trained model

Tokenization: The tokenizer from the Hugging Face Transformers library is used to convert the conversations into tokenized input for the model.

Managing Context: Need to ensure that previous conversation turns are included when generating a response.

Padding and Truncation: We need to ensure the inputs are padded or truncated to a fixed length for batch processing.

[ ]:
```
from transformers import AutoModelForCausalLM, AutoTokenizer

import torch
from torch.utils.data import DataLoader, TensorDataset

# Load DialoGPT tokenizer
tokenizer = AutoTokenizer.from_pretrained("microsoft/DialoGPT-small")
tokenizer.pad_token = tokenizer.eos_token

# Load DialoGPT model
model = AutoModelForCausalLM.from_pretrained("microsoft/DialoGPT-small")

# Tokenize cleaned conversations
def tokenize_conversation(questions, answers, max_length=512):
    input_ids_list = []
    attention_masks_list = []
    labels_list = []
    skipped_pairs = 0  # Keep track of skipped pairs


    for question, answer in zip(questions, answers):
```

```
        # Tokenize question and answer separately, then concatenate token IDs
        # The question and answer are now tokenized separately, each with a␣
↪maximum length of max_length // 2.
        # This avoids sequences that are too long when concatenating the␣
↪question and answer.
        encoded_question = tokenizer(question, return_tensors='pt',␣
↪max_length=max_length // 2, padding='max_length', truncation=True)
        encoded_answer = tokenizer(answer, return_tensors='pt',␣
↪max_length=max_length // 2, padding='max_length', truncation=True)

        input_ids = torch.cat([encoded_question['input_ids'],␣
↪encoded_answer['input_ids']], dim=1)  # Concatenate question and answer
        attention_mask = torch.cat([encoded_question['attention_mask'],␣
↪encoded_answer['attention_mask']], dim=1)

        # Skip empty inputs (if any)
        if input_ids.size(1) == 0:  # Check if input has a valid sequence length
            skipped_pairs += 1
            continue  # Skip this pair if it's invalid

        # Set labels as input_ids with padding token ignored (-100)
        labels = input_ids.clone()
        labels[labels == tokenizer.pad_token_id] = -100  # Ignore padding in␣
↪labels

        input_ids_list.append(input_ids)
        attention_masks_list.append(attention_mask)
        labels_list.append(labels)

    # Check if any pairs were processed
    if not input_ids_list:
        raise ValueError(f"No valid question-answer pairs were processed.␣
↪{skipped_pairs} pairs were skipped.")

    # Stack input IDs, attention masks, and labels to create tensors
    return torch.cat(input_ids_list, dim=0), torch.cat(attention_masks_list,␣
↪dim=0), torch.cat(labels_list, dim=0)

# Example usage
questions = ["You're asking me out. That's so cute. What's your name again?",␣
↪"Forget it."]
answers = ["No, no, it's my fault -- we didn't have a proper introduction ---",␣
↪"Cameron."]

input_ids, attention_masks, labels = tokenize_conversation(questions, answers)
```

```
print(f"Input IDs: {input_ids}")
print(f"Attention Masks: {attention_masks}")
print(f"Labels: {labels}")
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89:
UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
(https://huggingface.co/settings/tokens), set it as secret in your Google Colab
and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(
```

```
tokenizer_config.json:   0%|          | 0.00/614 [00:00<?, ?B/s]
```

```
vocab.json:   0%|          | 0.00/1.04M [00:00<?, ?B/s]
```

```
merges.txt:   0%|          | 0.00/456k [00:00<?, ?B/s]
```

```
config.json:   0%|          | 0.00/641 [00:00<?, ?B/s]
```

```
model.safetensors:   0%|          | 0.00/351M [00:00<?, ?B/s]
```

```
generation_config.json:   0%|          | 0.00/124 [00:00<?, ?B/s]
```

```
Input IDs: tensor([[ 1639,   821,  4737,  …, 50256, 50256, 50256],
        [ 1890,  1136,   340,  …, 50256, 50256, 50256]])
Attention Masks: tensor([[1, 1, 1,  …, 0, 0, 0],
        [1, 1, 1,  …, 0, 0, 0]])
Labels: tensor([[1639,  821, 4737,  …, -100, -100, -100],
        [1890, 1136,  340,  …, -100, -100, -100]])
```

[ ]:
```
# Tokenization for the training data
train_input_ids, train_attention_masks, train_labels =␣
 ↪tokenize_conversation(train_questions, train_answers)
val_input_ids, val_attention_masks, val_labels =␣
 ↪tokenize_conversation(val_questions, val_answers)
test_input_ids, test_attention_masks, test_labels =␣
 ↪tokenize_conversation(test_questions, test_answers)

print(f"Tokenized Training {len(train_input_ids)} input id's conversations.")
print(f"Tokenized Training {len(train_attention_masks)} attention masks␣
 ↪conversations.\n")

print(f"Tokenized Validation {len(val_input_ids)} input id's conversations.")
print(f"Tokenized Validation {len(val_attention_masks)} attention masks␣
 ↪conversations.\n")

print(f"Tokenized Test {len(test_input_ids)} input id's conversations.")
```

```
print(f"Tokenized Test {len(test_attention_masks)} attention masks␣
 ↪conversations.\n")
```

```
Tokenized Training 117919 input id's conversations.
Tokenized Training 117919 attention masks conversations.

Tokenized Validation 33692 input id's conversations.
Tokenized Validation 33692 attention masks conversations.

Tokenized Test 16846 input id's conversations.
Tokenized Test 16846 attention masks conversations.
```

8 Create Data Loaders for the Train, Validation and Test Data

[ ]:
```
from torch.utils.data import DataLoader, TensorDataset

# Create dataset and DataLoader for batching
def create_dataloader(input_ids, attention_mask, labels, batch_size=16):
    dataset = TensorDataset(input_ids, attention_mask, labels)
    return DataLoader(dataset, batch_size=batch_size, shuffle=True)

# Running using CPU initially, so choosing batch size of 16
# Smaller batch size can lead to noisier gradient updates but sometimes result␣
 ↪in better generalization.
# Larger batch sizes have more stable gradient updates but need more memory ,␣
 ↪learning rate needs tuning.
batch_size = 32

# Tokenized conversations
# Create DataLoader for each dataset
train_dataloader = create_dataloader(train_input_ids, train_attention_masks,␣
 ↪train_labels, batch_size=batch_size)
val_dataloader = create_dataloader(val_input_ids, val_attention_masks,␣
 ↪val_labels, batch_size=batch_size)
test_dataloader = create_dataloader(test_input_ids, test_attention_masks,␣
 ↪test_labels, batch_size=batch_size)

print(f"Training loader size: {len(train_dataloader.dataset)}")
print(f"Validation loader size: {len(val_dataloader.dataset)}")
print(f"Test loader size: {len(test_dataloader.dataset)}")

print(f"Number of steps in each epoch: {len(train_dataloader.dataset)/
 ↪batch_size}")
```

```
Training loader size: 117919
Validation loader size: 33692
```

```
Test loader size: 16846
Number of steps in each epoch: 3684.96875
```

9 Validate Data After Tokenization:

Print out some tokenized examples to ensure that the tokenization process is working as expected, and there are no empty sequences or overly long sequences:

[ ]:
```
for i, (input_ids, attention_mask, labels) in enumerate(train_dataloader):
    print(f"Batch {i} Tokenized Input Lengths: {[len(seq) for seq in␣
 ↪input_ids]}")
    break  # Only print the first batch
```

```
Batch 0 Tokenized Input Lengths: [512, 512, 512, 512, 512, 512, 512, 512, 512,
512, 512, 512, 512, 512, 512, 512, 512, 512, 512, 512, 512, 512, 512, 512, 512,
512, 512, 512, 512, 512, 512]
```

10 Training loop

Using manual training loop for better control over the training process instead of using HuggingFace's Trainer and TrainingArguments.

11 Evaluation:

Evaluation metrics like BLEU score or ROUGE Score to measure the quality of the generated conversations.

BLEU Score: This metric is commonly used for evaluating machine translation and is calculated by comparing n-grams of the generated response against the reference responses.

ROUGE Score: This metric is often used for evaluating text summarization and compares the overlap of n-grams between the generated response and reference texts.

[ ]:
```
import torch
from torch.optim import AdamW  # Import AdamW from PyTorch to avoid warning␣
 ↪when imported using transformers
from sklearn.metrics import accuracy_score, precision_recall_fscore_support,␣
 ↪accuracy_score, precision_score, recall_score, f1_score
import numpy as np
import time  # Import the time module
```

[ ]:
```
from nltk.translate.bleu_score import sentence_bleu
from rouge import Rouge

# Function to calculate BLEU score
def calculate_bleu(references, candidates):
    reference_tokens = [[ref.split()] for ref in references]  # Tokenize the␣
 ↪references
```

```
    candidate_tokens = [cand.split() for cand in candidates]

    # Calculate BLEU score for all predictions
    bleu_scores = [sentence_bleu(ref, cand) for ref, cand in␣
 ↪zip(reference_tokens, candidate_tokens)]
    avg_bleu_score = sum(bleu_scores) / len(bleu_scores) if bleu_scores else 0
    return avg_bleu_score

# Function to calculate ROUGE score
def calculate_rouge(references, candidates):
    rouge = Rouge()
    scores = rouge.get_scores(candidates, references, avg=True)
    return scores
```

[ ]:
```
# Example usage
reference_responses = ["I am going to the store.", "I went to the store."]
generated_response = "I am going to the shop."

# Calculate BLEU score
bleu_score = calculate_bleu(reference_responses, generated_response)
print(f"BLEU Score: {bleu_score:.4f}")

# Calculate ROUGE score
rouge_scores = calculate_rouge(reference_responses[0], generated_response)  #␣
 ↪Just using the first reference
print(f"ROUGE Scores: {rouge_scores}")
```

```
BLEU Score: 0.0000
ROUGE Scores: {'rouge-1': {'r': 0.8333333333333334, 'p': 0.8333333333333334,
'f': 0.8333333283333335}, 'rouge-2': {'r': 0.8, 'p': 0.8, 'f':
0.7999999950000002}, 'rouge-l': {'r': 0.8333333333333334, 'p':
0.8333333333333334, 'f': 0.8333333283333335}}

/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
```

```
UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
```

[ ]:
```python
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
from nltk.translate.bleu_score import sentence_bleu
from rouge import Rouge
import numpy as np
```

[ ]:
```python
def validate_model(model, val_loader, tokenizer):
    model.eval()  # Set model to evaluation mode
    total_val_loss = 0

    all_numerical_preds = []  # For accuracy, precision, recall, F1
    all_numerical_labels = []
    all_preds = [] # For BLEU/ROUGE
    all_labels = []
    all_references = []  # For BLEU and ROUGE evaluation
    all_candidates = []

    with torch.no_grad():
        for batch in val_loader:
            # Unpack the batch and move to device
            input_ids, attention_mask, labels = batch
            input_ids = input_ids.to(device)
            attention_mask = attention_mask.to(device)
            labels = labels.to(device)

            # Forward pass
            outputs = model(input_ids=input_ids, attention_mask=attention_mask,␣
 ↪labels=labels)
            loss = outputs.loss
            total_val_loss += loss.item()

            # Get predictions (logits) and decode generated sequences
            logits = outputs.logits
            predictions = torch.argmax(logits, dim=-1)

            # # Add logging to inspect predictions and labels
            # print(f"Step {step}:")
            # print(f"Predictions shape: {predictions.shape}, Labels shape:␣
 ↪{labels.shape}")
            # print(f"First 10 predictions: {predictions[:10]}")
            # print(f"First 10 labels: {labels[:10]}")
```

```
            # Check for NaN values or large values in predictions and labels
            if torch.isnan(predictions).any() or torch.isnan(labels).any():
                print(f"NaN values detected at Step {step}, skipping this batch.
↪")
                continue

            if predictions.max() >= 1e5 or labels.max() >= 1e5:
                print(f"Unexpectedly large values detected at Step {step},␣
↪skipping this batch.")
                continue

            # Decode predictions and labels to text for BLEU/ROUGE evaluation
            decoded_preds = tokenizer.batch_decode(predictions,␣
↪skip_special_tokens=True)
            decoded_labels = tokenizer.batch_decode(torch.clamp(labels, 0,␣
↪tokenizer.vocab_size - 1), skip_special_tokens=True)

            all_preds.extend(decoded_preds)
            all_labels.extend(decoded_labels)
            all_references.extend(decoded_labels)
            all_candidates.extend(decoded_preds)
            # **Keep numerical predictions and labels for accuracy/precision/
↪recall/F1**
            all_numerical_preds.extend(predictions.flatten().cpu().numpy())
            all_numerical_labels.extend(labels.flatten().cpu().numpy())


    avg_val_loss = total_val_loss / len(val_loader)
    print(f"Validation Loss: {avg_val_loss:.4f}")

    # Log shapes before computing metrics
    print(f"All labels shape: {len(all_labels)}, All predictions shape:␣
↪{len(all_preds)}")

    # **Calculate accuracy, precision, recall, and F1 using numerical␣
↪predictions and labels**
    accuracy = accuracy_score(all_numerical_labels, all_numerical_preds)
    precision = precision_score(all_numerical_labels, all_numerical_preds,␣
↪average='weighted')
    recall = recall_score(all_numerical_labels, all_numerical_preds,␣
↪average='weighted')
    f1 = f1_score(all_numerical_labels, all_numerical_preds, average='weighted')

    print(f"Accuracy: {accuracy:.4f}")
    print(f"Precision: {precision:.4f}")
```

```
    print(f"Recall: {recall:.4f}")
    print(f"F1-Score: {f1:.4f}")

    precision, recall, f1, _ = precision_recall_fscore_support(all_labels,␣
↪all_preds, average='weighted', zero_division=1)
    print(f"Precision: {precision:.4f}, Recall: {recall:.4f}, F1-Score: {f1:.
↪4f}")

    # Calculate BLEU and ROUGE scores for generated responses
    avg_bleu_score = calculate_bleu(all_labels, all_preds)
    print(f"BLEU Score: {avg_bleu_score:.4f}")

    rouge_scores = calculate_rouge(all_labels, all_preds)
    print(f"ROUGE Scores: {rouge_scores}")
```

[ ]:
```
def check_for_nan(tensor, name="tensor"):
    if torch.isnan(tensor).any():
        print(f"Found NaN values in {name}.")
        return True
    return False
```

https://www.baeldung.com/cs/ml-training-nan-errors-fix#:~:text=These%20sources%20include%20data%20error

https://neptune.ai/blog/understanding-gradient-clipping-and-how-it-can-fix-exploding-gradients-problem

Backpropagation calculates the gradients of the cost function w.r.t. the weights and biases in the network.

It tells you about all the changes you need to make to your weights to minimize the cost function (it's actually -1* to see the steepest decrease, and + would give you the steepest increase in the cost function).

Vanishing Gradients: The translation of the effect of a change in cost function (C) to the weight in an initial layer, or the norm of the gradient, becomes so small due to increased model complexity with more hidden units that it becomes zero after a certain point. This is what we call vanishing gradients.

This hampers the learning of the model. The weights can no longer contribute to the reduction in cost function (C) and go unchanged, affecting the network in the forward pass and eventually stalling the model.

Exploding gradients On the other hand, the exploding gradient problem refers to a large increase in the norm of the gradient during training.

Such events are caused by an explosion of long-term components, which can grow exponentially more than short-term ones. This results in an unstable network that, at best, cannot learn from the training data, making the gradient descent step impossible to execute.

Gradient clipping is a technique used to stabilize the training of neural networks by rescaling the error derivative to a threshold. This prevents the gradients from becoming too large, which can cause the model to diverge and fail to converge to a good solution.

Here's how gradient clipping works: Set a threshold: Define a minimum and maximum threshold. Calculate the norm: Calculate the norm of the gradients. Scale the gradients: If the norm exceeds the threshold, scale down the gradients proportionally to meet the norm threshold. Update the weights: Use the clipped gradients to update the weights.

Gradient clipping can be performed in two ways: **Clipping by value**: Define a minimum and maximum threshold. **Clipping by norm**: Set a maximum threshold for the norm of the gradients

[ ]:
```
import gc
from transformers import AdamW, get_linear_schedule_with_warmup
import torch

import os
os.environ["PYTORCH_CUDA_ALLOC_CONF"] = "expandable_segments:True"


# Move model to GPU if available
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

# Move model to the CPU
model.to(device)

# Apply gradient checkpointing to the model
model.gradient_checkpointing_enable()

# Set up the optimizer (AdamW is a common choice for transformer models)
# Disable weight decay
optimizer = AdamW(model.parameters(), lr=5e-6,weight_decay=0.0) # Reduced the␣
 ↪learning rate from 5e-5 due to nan loss

total_epochs = 3

save_every_n_steps = 100

# Define the number of warm-up steps and total training steps
total_training_steps = len(train_dataloader) * total_epochs  # Total steps =␣
 ↪batches per epoch * number of epochs
warmup_steps = int(0.1 * total_training_steps)  # 10% of total steps as warm-up

# Define learning rate scheduler
scheduler = get_linear_schedule_with_warmup(optimizer,
                                            num_warmup_steps=warmup_steps,
                                           ␣
 ↪num_training_steps=total_training_steps)
# Define training function
def train_model(model, train_dataloader, val_dataloader, optimizer, epochs=3, ␣
 ↪save_every_n_steps=100, start_epoch=0, start_step=0):
```

```
    # Define the path where the models will be saved
    model_save_path = "/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/"
    print(epochs)
    print(start_epoch)

    for epoch in range(start_epoch, epochs):
        print(f"\nEpoch {epoch + 1}/{epochs}:")
        model.train()  # Set model to training mode

        total_train_loss = 0
        start_time = time.time()  # Start time for the epoch
        gradient_accumulation_steps = 2  # Accumulate gradients over 2 batches

        # Training loop on  Train Data
        for step, batch in enumerate(train_dataloader, start=start_step):
            optimizer.zero_grad() # Clear previous gradients

            # Unpack the batch and move to the device
            input_ids, attention_mask, labels = batch
            input_ids = input_ids.to(device)
            attention_mask = attention_mask.to(device)
            labels = labels.to(device)

            # Check for NaN values in input IDs
            # if torch.isnan(input_ids).any():
            if check_for_nan(input_ids, "input_ids") or␣
↪check_for_nan(attention_mask, "attention_mask"):
                print("Input IDs contain NaN values. Stopping training.")
                print("NaN detected in inputs, stopping training.")
                return

            # Forward pass and compute loss
            outputs = model(input_ids=input_ids, attention_mask=attention_mask,␣
↪labels=labels)
            loss = outputs.loss

            # NaN check for loss
            if torch.isnan(loss):
                print(f"NaN loss encountered at Step {step} in Epoch {epoch+1}.␣
↪Stopping training.")
                print(f"Input IDs: {input_ids}")
                print(f"Attention Mask: {attention_mask}")
                print(f"Labels: {labels}")
                return  # Stop training to prevent further NaN propagation

            total_train_loss += loss.item()
            loss.backward() # Backward pass to calculate gradients
```

```
            # Check if gradients are NaN
            for param in model.parameters():
                if param.grad is not None and torch.isnan(param.grad).any():
                    print("Encountered NaN gradients. Stopping training.")
                    return

            # Gradient clipping to prevent gradient explosion
            if (step + 1) % gradient_accumulation_steps == 0:
              # To prevent Gradient explosion due to large batch sizes, implement gradient clipping
              torch.nn.utils.clip_grad_norm_(model.parameters(), max_norm=1.0)
              # Update parameters
              optimizer.step()
              optimizer.zero_grad()
              # Step the scheduler to adjust the learning rate
              scheduler.step()

            # Print every 10th step
            if step % 10 == 0 and step > 0:
                print(f"Epoch {epoch+1},  Step {step}: Loss = {loss.item():.4f}")

            # Save model every 100 steps (or whatever value save_every_n_steps is set to)
            if (step + 1) % save_every_n_steps == 0:
                model_save_name = f'fine_tuned_dialoGPT_epoch{epoch+1}_step{step+1}.pt'
                model_save_full_path = os.path.join(model_save_path, model_save_name)
                # Save model state dict, optimizer state, and scheduler state
                torch.save({
                    'epoch': epoch + 1,
                    'step': step + 1,
                    'model_state_dict': model.state_dict(),
                    'optimizer_state_dict': optimizer.state_dict(),
                    'scheduler_state_dict': scheduler.state_dict(),
                }, model_save_full_path)

                print(f"Checkpoint saved: {model_save_name}")

            del outputs, loss  # Delete variables after use
            gc.collect()
            torch.cuda.empty_cache()

        # Epoch-level reporting
        avg_train_loss = total_train_loss / len(train_dataloader)
```

```
        # Calculate the time taken for the epoch
        elapsed_time = time.time() - start_time
        print(f"Epoch {epoch+1} completed. Training Loss: {avg_train_loss:.4f}. Time for epoch: {elapsed_time:.2f} seconds")

        # Validation after each epoch using validation data
        validate_model(model, val_dataloader, tokenizer)

        model_save_name = 'fine_tuned_dialoGPT{epoch+1}_final.pt'
        model_save_full_path = os.path.join(model_save_path, model_save_name)


        #Save model state dict, optimizer state, and scheduler state
        torch.save({
                    'epoch': epoch + 1,
                    'step': step + 1,
                    'model_state_dict': model.state_dict(),
                    'optimizer_state_dict': optimizer.state_dict(),
                    'scheduler_state_dict': scheduler.state_dict(),
                }, model_save_full_path)
        print(f"Checkpoint saved: {model_save_name}")

        model_save_name = 'fine_tuned_dialoGPT_final.pt'
        path = F"/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/{model_save_name}"
        model.save_pretrained(path)
        tokenizer.save_pretrained(path)
        print(f"Model {model_save_name} and tokenizer saved to {path}\n")
```

```
/usr/local/lib/python3.10/dist-packages/transformers/optimization.py:591:
FutureWarning: This implementation of AdamW is deprecated and will be removed in
a future version. Use the PyTorch implementation torch.optim.AdamW instead, or
set `no_deprecation_warning=True` to disable this warning
  warnings.warn(
```

[ ]:
```
# Define the path where the models will be saved
model_save_path = "/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/"
model_save_name = 'LastGoodModel.pt'
model_save_full_path = os.path.join(model_save_path, model_save_name)
#path = F"/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/{model_save_name}"
#Save model state dict, optimizer state, and scheduler state
torch.save({'epoch': 3,
            'step': 3601,
            'model_state_dict': model.state_dict(),
            'optimizer_state_dict': optimizer.state_dict(),
            'scheduler_state_dict': scheduler.state_dict(),
```

```
            }, model_save_full_path)
print(f"Checkpoint saved: {model_save_name}")
```

```
Checkpoint saved: LastGoodModel.pt
```

12 Train the model

[ ]:
```
# Train the model with 3 epochs
train_model(model, train_dataloader, val_dataloader, optimizer,␣
 ↪epochs=total_epochs, save_every_n_steps=100)
```

```
Epoch 1/3:

`use_cache=True` is incompatible with gradient checkpointing. Setting
`use_cache=False`…
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:295:
FutureWarning: `torch.cpu.amp.autocast(args…)` is deprecated. Please use
`torch.amp.autocast('cpu', args…)` instead.
  with torch.enable_grad(), device_autocast_ctx,
torch.cpu.amp.autocast(**ctx.cpu_autocast_kwargs):  # type: ignore[attr-defined]

Epoch 1,  Step 10: Loss = 10.6841
Epoch 1,  Step 20: Loss = 10.3804
Epoch 1,  Step 30: Loss = 10.9327
Epoch 1,  Step 40: Loss = 10.9661
Epoch 1,  Step 50: Loss = 10.4565
Epoch 1,  Step 60: Loss = 10.5978
Epoch 1,  Step 70: Loss = 10.1374
Epoch 1,  Step 80: Loss = 10.8555
Epoch 1,  Step 90: Loss = 9.9942
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step100.pt
Epoch 1,  Step 100: Loss = 11.5294
Epoch 1,  Step 110: Loss = 9.9718
Epoch 1,  Step 120: Loss = 10.3927
Epoch 1,  Step 130: Loss = 10.4790
Epoch 1,  Step 140: Loss = 10.4229
Epoch 1,  Step 150: Loss = 10.7004
Epoch 1,  Step 160: Loss = 10.5868
Epoch 1,  Step 170: Loss = 10.8074
Epoch 1,  Step 180: Loss = 9.8277
Epoch 1,  Step 190: Loss = 10.0279
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step200.pt
Epoch 1,  Step 200: Loss = 10.4360
Epoch 1,  Step 210: Loss = 9.7197
Epoch 1,  Step 220: Loss = 10.2452
Epoch 1,  Step 230: Loss = 9.8166
Epoch 1,  Step 240: Loss = 10.1381
Epoch 1,  Step 250: Loss = 10.4864
```

```
Epoch 1,  Step 260: Loss = 10.8253
Epoch 1,  Step 270: Loss = 9.7862
Epoch 1,  Step 280: Loss = 10.5419
Epoch 1,  Step 290: Loss = 10.7187
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step300.pt
Epoch 1,  Step 300: Loss = 9.7793
Epoch 1,  Step 310: Loss = 9.7748
Epoch 1,  Step 320: Loss = 9.6615
Epoch 1,  Step 330: Loss = 10.2057
Epoch 1,  Step 340: Loss = 9.1771
Epoch 1,  Step 350: Loss = 9.7935
Epoch 1,  Step 360: Loss = 9.8590
Epoch 1,  Step 370: Loss = 9.4233
Epoch 1,  Step 380: Loss = 10.0744
Epoch 1,  Step 390: Loss = 9.5687
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step400.pt
Epoch 1,  Step 400: Loss = 10.1565
Epoch 1,  Step 410: Loss = 9.6862
Epoch 1,  Step 420: Loss = 9.7825
Epoch 1,  Step 430: Loss = 9.3282
Epoch 1,  Step 440: Loss = 9.7749
Epoch 1,  Step 450: Loss = 9.5354
Epoch 1,  Step 460: Loss = 9.3427
Epoch 1,  Step 470: Loss = 8.9849
Epoch 1,  Step 480: Loss = 9.1078
Epoch 1,  Step 490: Loss = 8.9095
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step500.pt
Epoch 1,  Step 500: Loss = 9.0611
Epoch 1,  Step 510: Loss = 9.1283
Epoch 1,  Step 520: Loss = 9.0189
Epoch 1,  Step 530: Loss = 8.6822
Epoch 1,  Step 540: Loss = 9.0289
Epoch 1,  Step 550: Loss = 8.9093
Epoch 1,  Step 560: Loss = 8.8114
Epoch 1,  Step 570: Loss = 8.7732
Epoch 1,  Step 580: Loss = 8.6376
Epoch 1,  Step 590: Loss = 8.4428
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step600.pt
Epoch 1,  Step 600: Loss = 8.4361
Epoch 1,  Step 610: Loss = 8.4515
Epoch 1,  Step 620: Loss = 8.1122
Epoch 1,  Step 630: Loss = 8.1676
Epoch 1,  Step 640: Loss = 7.8608
Epoch 1,  Step 650: Loss = 7.7962
Epoch 1,  Step 660: Loss = 7.9588
Epoch 1,  Step 670: Loss = 7.8622
Epoch 1,  Step 680: Loss = 7.5705
Epoch 1,  Step 690: Loss = 7.3455
```

```
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step700.pt
Epoch 1,  Step 700: Loss = 7.2724
Epoch 1,  Step 710: Loss = 7.1808
Epoch 1,  Step 720: Loss = 6.6314
Epoch 1,  Step 730: Loss = 6.9224
Epoch 1,  Step 740: Loss = 6.8174
Epoch 1,  Step 750: Loss = 6.8585
Epoch 1,  Step 760: Loss = 6.8115
Epoch 1,  Step 770: Loss = 6.8594
Epoch 1,  Step 780: Loss = 6.6698
Epoch 1,  Step 790: Loss = 6.4587
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step800.pt
Epoch 1,  Step 800: Loss = 6.1831
Epoch 1,  Step 810: Loss = 6.2287
Epoch 1,  Step 820: Loss = 6.3523
Epoch 1,  Step 830: Loss = 5.7052
Epoch 1,  Step 840: Loss = 5.8247
Epoch 1,  Step 850: Loss = 5.8336
Epoch 1,  Step 860: Loss = 5.9512
Epoch 1,  Step 870: Loss = 6.0545
Epoch 1,  Step 880: Loss = 5.8469
Epoch 1,  Step 890: Loss = 5.7855
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step900.pt
Epoch 1,  Step 900: Loss = 5.7680
Epoch 1,  Step 910: Loss = 5.7595
Epoch 1,  Step 920: Loss = 5.6053
Epoch 1,  Step 930: Loss = 5.9169
Epoch 1,  Step 940: Loss = 5.4508
Epoch 1,  Step 950: Loss = 5.5033
Epoch 1,  Step 960: Loss = 5.4808
Epoch 1,  Step 970: Loss = 5.4581
Epoch 1,  Step 980: Loss = 5.1444
Epoch 1,  Step 990: Loss = 5.2950
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1000.pt
Epoch 1,  Step 1000: Loss = 5.3225
Epoch 1,  Step 1010: Loss = 5.3190
Epoch 1,  Step 1020: Loss = 5.1506
Epoch 1,  Step 1030: Loss = 5.2347
Epoch 1,  Step 1040: Loss = 5.3114
Epoch 1,  Step 1050: Loss = 5.1819
Epoch 1,  Step 1060: Loss = 5.0064
Epoch 1,  Step 1070: Loss = 5.0272
Epoch 1,  Step 1080: Loss = 4.9731
Epoch 1,  Step 1090: Loss = 4.7455
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1100.pt
Epoch 1,  Step 1100: Loss = 5.0232
Epoch 1,  Step 1110: Loss = 4.8335
Epoch 1,  Step 1120: Loss = 4.8914
```

```
Epoch 1,  Step 1130: Loss = 4.9044
Epoch 1,  Step 1140: Loss = 4.7160
Epoch 1,  Step 1150: Loss = 4.6787
Epoch 1,  Step 1160: Loss = 4.6017
Epoch 1,  Step 1170: Loss = 4.5723
Epoch 1,  Step 1180: Loss = 4.5982
Epoch 1,  Step 1190: Loss = 4.3533
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1200.pt
Epoch 1,  Step 1200: Loss = 4.5045
Epoch 1,  Step 1210: Loss = 4.3145
Epoch 1,  Step 1220: Loss = 4.4498
Epoch 1,  Step 1230: Loss = 4.5618
Epoch 1,  Step 1240: Loss = 4.2409
Epoch 1,  Step 1250: Loss = 4.0479
Epoch 1,  Step 1260: Loss = 3.8978
Epoch 1,  Step 1270: Loss = 4.3888
Epoch 1,  Step 1280: Loss = 4.2028
Epoch 1,  Step 1290: Loss = 4.1092
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1300.pt
Epoch 1,  Step 1300: Loss = 4.0824
Epoch 1,  Step 1310: Loss = 4.0565
Epoch 1,  Step 1320: Loss = 4.2204
Epoch 1,  Step 1330: Loss = 3.8449
Epoch 1,  Step 1340: Loss = 4.0559
Epoch 1,  Step 1350: Loss = 3.8877
Epoch 1,  Step 1360: Loss = 4.0217
Epoch 1,  Step 1370: Loss = 3.7077
Epoch 1,  Step 1380: Loss = 3.9544
Epoch 1,  Step 1390: Loss = 4.0258
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1400.pt
Epoch 1,  Step 1400: Loss = 3.9138
Epoch 1,  Step 1410: Loss = 3.8287
Epoch 1,  Step 1420: Loss = 4.3634
Epoch 1,  Step 1430: Loss = 3.8011
Epoch 1,  Step 1440: Loss = 3.9254
Epoch 1,  Step 1450: Loss = 3.9818
Epoch 1,  Step 1460: Loss = 3.8285
Epoch 1,  Step 1470: Loss = 3.7163
Epoch 1,  Step 1480: Loss = 3.8333
Epoch 1,  Step 1490: Loss = 3.8707
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1500.pt
Epoch 1,  Step 1500: Loss = 3.8593
Epoch 1,  Step 1510: Loss = 3.7594
Epoch 1,  Step 1520: Loss = 3.5821
Epoch 1,  Step 1530: Loss = 3.8291
Epoch 1,  Step 1540: Loss = 3.7322
Epoch 1,  Step 1550: Loss = 3.8761
Epoch 1,  Step 1560: Loss = 3.8937
```

```
Epoch 1,  Step 1570: Loss = 3.9736
Epoch 1,  Step 1580: Loss = 3.9845
Epoch 1,  Step 1590: Loss = 3.9687
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1600.pt
Epoch 1,  Step 1600: Loss = 3.8056
Epoch 1,  Step 1610: Loss = 3.8812
Epoch 1,  Step 1620: Loss = 3.7610
Epoch 1,  Step 1630: Loss = 3.7709
Epoch 1,  Step 1640: Loss = 3.7423
Epoch 1,  Step 1650: Loss = 3.9470
Epoch 1,  Step 1660: Loss = 3.6968
Epoch 1,  Step 1670: Loss = 3.7844
Epoch 1,  Step 1680: Loss = 3.9871
Epoch 1,  Step 1690: Loss = 3.8287
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1700.pt
Epoch 1,  Step 1700: Loss = 3.8279
Epoch 1,  Step 1710: Loss = 3.7327
Epoch 1,  Step 1720: Loss = 3.6590
Epoch 1,  Step 1730: Loss = 3.6589
Epoch 1,  Step 1740: Loss = 3.6397
Epoch 1,  Step 1750: Loss = 3.7404
Epoch 1,  Step 1760: Loss = 3.4864
Epoch 1,  Step 1770: Loss = 3.9234
Epoch 1,  Step 1780: Loss = 3.7473
Epoch 1,  Step 1790: Loss = 3.6831
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1800.pt
Epoch 1,  Step 1800: Loss = 3.9577
Epoch 1,  Step 1810: Loss = 3.6007
Epoch 1,  Step 1820: Loss = 3.7456
Epoch 1,  Step 1830: Loss = 3.6650
Epoch 1,  Step 1840: Loss = 3.6311
Epoch 1,  Step 1850: Loss = 3.6067
Epoch 1,  Step 1860: Loss = 3.7536
Epoch 1,  Step 1870: Loss = 3.7841
Epoch 1,  Step 1880: Loss = 3.6351
Epoch 1,  Step 1890: Loss = 3.8579
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step1900.pt
Epoch 1,  Step 1900: Loss = 3.7325
Epoch 1,  Step 1910: Loss = 3.5480
Epoch 1,  Step 1920: Loss = 3.7143
Epoch 1,  Step 1930: Loss = 3.9296
Epoch 1,  Step 1940: Loss = 3.6530
Epoch 1,  Step 1950: Loss = 3.5916
Epoch 1,  Step 1960: Loss = 3.6267
Epoch 1,  Step 1970: Loss = 3.5725
Epoch 1,  Step 1980: Loss = 3.3715
Epoch 1,  Step 1990: Loss = 3.7549
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2000.pt
```

```
Epoch 1,  Step 2000: Loss = 3.8076
Epoch 1,  Step 2010: Loss = 3.7005
Epoch 1,  Step 2020: Loss = 3.5915
Epoch 1,  Step 2030: Loss = 3.5998
Epoch 1,  Step 2040: Loss = 3.8054
Epoch 1,  Step 2050: Loss = 3.1799
Epoch 1,  Step 2060: Loss = 3.8033
Epoch 1,  Step 2070: Loss = 3.7103
Epoch 1,  Step 2080: Loss = 3.2943
Epoch 1,  Step 2090: Loss = 3.6997
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2100.pt
Epoch 1,  Step 2100: Loss = 3.5917
Epoch 1,  Step 2110: Loss = 3.5226
Epoch 1,  Step 2120: Loss = 3.7024
Epoch 1,  Step 2130: Loss = 3.6358
Epoch 1,  Step 2140: Loss = 3.7982
Epoch 1,  Step 2150: Loss = 3.7262
Epoch 1,  Step 2160: Loss = 3.6132
Epoch 1,  Step 2170: Loss = 3.6003
Epoch 1,  Step 2180: Loss = 3.5952
Epoch 1,  Step 2190: Loss = 3.5377
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2200.pt
Epoch 1,  Step 2200: Loss = 3.5371
Epoch 1,  Step 2210: Loss = 3.8465
Epoch 1,  Step 2220: Loss = 3.5324
Epoch 1,  Step 2230: Loss = 3.5307
Epoch 1,  Step 2240: Loss = 3.6743
Epoch 1,  Step 2250: Loss = 3.3746
Epoch 1,  Step 2260: Loss = 3.6654
Epoch 1,  Step 2270: Loss = 3.5159
Epoch 1,  Step 2280: Loss = 3.6810
Epoch 1,  Step 2290: Loss = 3.2321
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2300.pt
Epoch 1,  Step 2300: Loss = 3.7877
Epoch 1,  Step 2310: Loss = 3.4775
Epoch 1,  Step 2320: Loss = 3.4311
Epoch 1,  Step 2330: Loss = 3.4875
Epoch 1,  Step 2340: Loss = 3.5630
Epoch 1,  Step 2350: Loss = 3.4107
Epoch 1,  Step 2360: Loss = 3.6176
Epoch 1,  Step 2370: Loss = 3.6026
Epoch 1,  Step 2380: Loss = 3.5694
Epoch 1,  Step 2390: Loss = 3.4280
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2400.pt
Epoch 1,  Step 2400: Loss = 3.4631
Epoch 1,  Step 2410: Loss = 3.4765
Epoch 1,  Step 2420: Loss = 3.7302
Epoch 1,  Step 2430: Loss = 3.3917
```

```
Epoch 1,  Step 2440: Loss = 3.5497
Epoch 1,  Step 2450: Loss = 3.5004
Epoch 1,  Step 2460: Loss = 3.4686
Epoch 1,  Step 2470: Loss = 3.4050
Epoch 1,  Step 2480: Loss = 3.5642
Epoch 1,  Step 2490: Loss = 3.4672
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2500.pt
Epoch 1,  Step 2500: Loss = 3.5027
Epoch 1,  Step 2510: Loss = 3.4215
Epoch 1,  Step 2520: Loss = 3.4940
Epoch 1,  Step 2530: Loss = 3.4668
Epoch 1,  Step 2540: Loss = 3.6237
Epoch 1,  Step 2550: Loss = 3.5574
Epoch 1,  Step 2560: Loss = 3.5621
Epoch 1,  Step 2570: Loss = 3.4907
Epoch 1,  Step 2580: Loss = 3.5604
Epoch 1,  Step 2590: Loss = 3.3162
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2600.pt
Epoch 1,  Step 2600: Loss = 3.7247
Epoch 1,  Step 2610: Loss = 3.5555
Epoch 1,  Step 2620: Loss = 3.6340
Epoch 1,  Step 2630: Loss = 3.5217
Epoch 1,  Step 2640: Loss = 3.6022
Epoch 1,  Step 2650: Loss = 3.4768
Epoch 1,  Step 2660: Loss = 3.4051
Epoch 1,  Step 2670: Loss = 3.6098
Epoch 1,  Step 2680: Loss = 3.4110
Epoch 1,  Step 2690: Loss = 3.4093
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2700.pt
Epoch 1,  Step 2700: Loss = 3.5803
Epoch 1,  Step 2710: Loss = 3.4662
Epoch 1,  Step 2720: Loss = 3.4869
Epoch 1,  Step 2730: Loss = 3.3274
Epoch 1,  Step 2740: Loss = 3.4205
Epoch 1,  Step 2750: Loss = 3.6211
Epoch 1,  Step 2760: Loss = 3.4125
Epoch 1,  Step 2770: Loss = 3.5080
Epoch 1,  Step 2780: Loss = 3.4176
Epoch 1,  Step 2790: Loss = 3.4007
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2800.pt
Epoch 1,  Step 2800: Loss = 3.1399
Epoch 1,  Step 2810: Loss = 3.4556
Epoch 1,  Step 2820: Loss = 3.6151
Epoch 1,  Step 2830: Loss = 3.4919
Epoch 1,  Step 2840: Loss = 3.4340
Epoch 1,  Step 2850: Loss = 3.3800
Epoch 1,  Step 2860: Loss = 3.1187
Epoch 1,  Step 2870: Loss = 3.4247
```

```
Epoch 1,  Step 2880: Loss = 3.3591
Epoch 1,  Step 2890: Loss = 3.4897
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step2900.pt
Epoch 1,  Step 2900: Loss = 3.3693
Epoch 1,  Step 2910: Loss = 3.4451
Epoch 1,  Step 2920: Loss = 3.4889
Epoch 1,  Step 2930: Loss = 3.2475
Epoch 1,  Step 2940: Loss = 3.5924
Epoch 1,  Step 2950: Loss = 3.4573
Epoch 1,  Step 2960: Loss = 3.4623
Epoch 1,  Step 2970: Loss = 3.4812
Epoch 1,  Step 2980: Loss = 3.5765
Epoch 1,  Step 2990: Loss = 3.4894
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step3000.pt
Epoch 1,  Step 3000: Loss = 3.1305
Epoch 1,  Step 3010: Loss = 3.3397
Epoch 1,  Step 3020: Loss = 3.4101
Epoch 1,  Step 3030: Loss = 3.3555
Epoch 1,  Step 3040: Loss = 3.5626
Epoch 1,  Step 3050: Loss = 3.4509
Epoch 1,  Step 3060: Loss = 3.4858
Epoch 1,  Step 3070: Loss = 3.5296
Epoch 1,  Step 3080: Loss = 3.4083
Epoch 1,  Step 3090: Loss = 3.3319
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step3100.pt
Epoch 1,  Step 3100: Loss = 3.2392
Epoch 1,  Step 3110: Loss = 3.2758
Epoch 1,  Step 3120: Loss = 3.4383
Epoch 1,  Step 3130: Loss = 3.3504
Epoch 1,  Step 3140: Loss = 3.4574
Epoch 1,  Step 3150: Loss = 3.4230
Epoch 1,  Step 3160: Loss = 3.5171
Epoch 1,  Step 3170: Loss = 3.4471
Epoch 1,  Step 3180: Loss = 3.4262
Epoch 1,  Step 3190: Loss = 3.5544
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step3200.pt
Epoch 1,  Step 3200: Loss = 3.4497
Epoch 1,  Step 3210: Loss = 3.2540
Epoch 1,  Step 3220: Loss = 3.4382
Epoch 1,  Step 3230: Loss = 3.6775
Epoch 1,  Step 3240: Loss = 3.2879
Epoch 1,  Step 3250: Loss = 3.3296
Epoch 1,  Step 3260: Loss = 3.4075
Epoch 1,  Step 3270: Loss = 3.4807
Epoch 1,  Step 3280: Loss = 3.4871
Epoch 1,  Step 3290: Loss = 3.2809
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step3300.pt
Epoch 1,  Step 3300: Loss = 3.3162
```

```
Epoch 1,  Step 3310: Loss = 3.4402
Epoch 1,  Step 3320: Loss = 3.2777
Epoch 1,  Step 3330: Loss = 3.4007
Epoch 1,  Step 3340: Loss = 3.4183
Epoch 1,  Step 3350: Loss = 3.5082
Epoch 1,  Step 3360: Loss = 3.3787
Epoch 1,  Step 3370: Loss = 3.1961
Epoch 1,  Step 3380: Loss = 3.4230
Epoch 1,  Step 3390: Loss = 3.3798
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step3400.pt
Epoch 1,  Step 3400: Loss = 3.2791
Epoch 1,  Step 3410: Loss = 3.3160
Epoch 1,  Step 3420: Loss = 3.2889
Epoch 1,  Step 3430: Loss = 3.4996
Epoch 1,  Step 3440: Loss = 3.3474
Epoch 1,  Step 3450: Loss = 3.1021
Epoch 1,  Step 3460: Loss = 3.7081
Epoch 1,  Step 3470: Loss = 3.3165
Epoch 1,  Step 3480: Loss = 3.6784
Epoch 1,  Step 3490: Loss = 3.4358
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step3500.pt
Epoch 1,  Step 3500: Loss = 3.5414
Epoch 1,  Step 3510: Loss = 3.2972
Epoch 1,  Step 3520: Loss = 3.4533
Epoch 1,  Step 3530: Loss = 3.3650
Epoch 1,  Step 3540: Loss = 3.5634
Epoch 1,  Step 3550: Loss = 3.1038
Epoch 1,  Step 3560: Loss = 3.3411
Epoch 1,  Step 3570: Loss = 3.4803
Epoch 1,  Step 3580: Loss = 3.2426
Epoch 1,  Step 3590: Loss = 3.2062
Checkpoint saved: fine_tuned_dialoGPT_epoch1_step3600.pt
Epoch 1,  Step 3600: Loss = 3.4244
Epoch 1,  Step 3610: Loss = 3.5864
Epoch 1,  Step 3620: Loss = 3.3289
Epoch 1,  Step 3630: Loss = 3.1798
Epoch 1,  Step 3640: Loss = 3.4910
Epoch 1,  Step 3650: Loss = 3.3098
Epoch 1,  Step 3660: Loss = 3.1883
Epoch 1,  Step 3670: Loss = 3.5391
Epoch 1,  Step 3680: Loss = 3.2279
Epoch 1 completed. Training Loss: 5.0178. Time for epoch: 5243.15 seconds
```

```
---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call last)
<ipython-input-25-e941a2dd501e> in <cell line: 2>()
      1 # Train the model with 3 epochs
```

```
----> 2 train_model(model, train_dataloader, val_dataloader, optimizer,␣
 ↪epochs=epochs, save_every_n_steps=100)

<ipython-input-24-c5f2aaa19346> in train_model(model, train_dataloader,␣
 ↪val_dataloader, optimizer, epochs, save_every_n_steps)
    126
    127         # Validation after each epoch using validation data
--> 128         validate_model(model, val_dataloader, tokenizer)
    129
    130         model_save_name = 'fine_tuned_dialoGPT{epoch+1}_final.pt'

<ipython-input-22-90f95568db0e> in validate_model(model, val_loader, tokenizer)
     30             # Decode predictions and labels to text for BLEU/ROUGE␣
 ↪evaluation
     31             decoded_preds = tokenizer.batch_decode(predictions,␣
 ↪skip_special_tokens=True)
---> 32             decoded_labels = tokenizer.batch_decode(labels,␣
 ↪skip_special_tokens=True)
     33
     34             all_preds.extend(decoded_preds)

/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py
 ↪in batch_decode(self, sequences, skip_special_tokens,␣
 ↪clean_up_tokenization_spaces, **kwargs)
   3974             `List[str]`: The list of decoded sentences.
   3975         """
-> 3976         return [

   3977             self.decode(
   3978                 seq,

/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py
 ↪in <listcomp>(.0)
   3975         """
   3976         return [
-> 3977             self.decode(

   3978                 seq,
   3979                 skip_special_tokens=skip_special_tokens,

/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py
 ↪in decode(self, token_ids, skip_special_tokens, clean_up_tokenization_spaces,
 ↪**kwargs)
   4014         token_ids = to_py_obj(token_ids)
   4015
-> 4016         return self._decode(

   4017             token_ids=token_ids,
   4018             skip_special_tokens=skip_special_tokens,
```

```
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_fast.py
 ↪in _decode(self, token_ids, skip_special_tokens, clean_up_tokenization_spaces␣
 ↪**kwargs)
    649         if isinstance(token_ids, int):
    650             token_ids = [token_ids]
--> 651         text = self._tokenizer.decode(token_ids,␣
 ↪skip_special_tokens=skip_special_tokens)
    652
    653         clean_up_tokenization_spaces = (

OverflowError: out of range integral type conversion attempted
```

13 To Resume Training from saved checkpoint

[ ]:
```
# Load checkpoint to resume training
checkpoint = torch.load('/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/
 ↪fine_tuned_dialoGPT_epoch2_step3600.pt', map_location=torch.device(device))
#checkpoint = torch.load('/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/
 ↪LastGoodModel.pt', map_location=torch.device(device))
model.load_state_dict(checkpoint['model_state_dict'])
optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
scheduler.load_state_dict(checkpoint['scheduler_state_dict'])
start_epoch = checkpoint['epoch']  # Start from the next epoch
start_step = checkpoint['step']
print(f"Loaded checkpoint from epoch {start_epoch} and step {start_step}")
```

```
<ipython-input-43-8e4720303312>:2: FutureWarning: You are using `torch.load`
with `weights_only=False` (the current default value), which uses the default
pickle module implicitly. It is possible to construct malicious pickle data
which will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for
more details). In a future release, the default value for `weights_only` will be
flipped to `True`. This limits the functions that could be executed during
unpickling. Arbitrary objects will no longer be allowed to be loaded via this
mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control of the
loaded file. Please open an issue on GitHub for any issues related to this
experimental feature.
  checkpoint = torch.load('/content/drive/MyDrive/Colab
Notebooks/ChatFlixModels/fine_tuned_dialoGPT_epoch1_step3600.pt',
map_location=torch.device(device))

Loaded checkpoint from epoch 1 and step 3600
```

[ ]:
```
# Now continue training from this checkpoint
remaining_epochs = total_epochs - start_epoch
```

```
# Train the model for remaining epochs
train_model(model, train_dataloader, val_dataloader, optimizer,␣
 ↪epochs=remaining_epochs, save_every_n_steps=100, start_epoch=start_epoch,␣
 ↪start_step=0)
```

[ ]:
```
# Train the model for remaining epochs
train_model(model, train_dataloader, val_dataloader, optimizer, epochs=3,␣
 ↪save_every_n_steps=100, start_epoch=2, start_step=0)
```

```
3
2

Epoch 3/3:

/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:295:
FutureWarning: `torch.cpu.amp.autocast(args…)` is deprecated. Please use
`torch.amp.autocast('cpu', args…)` instead.
  with torch.enable_grad(), device_autocast_ctx,
torch.cpu.amp.autocast(**ctx.cpu_autocast_kwargs):  # type: ignore[attr-defined]

Streaming output truncated to the last 5000 lines.
        [2093,  262, 1808,  …, -100, -100, -100],
        [1996,  683,  287,  …, -100, -100, -100],
        …,
        [3003,  318,  339,  …, -100, -100, -100],
        [ 732,  460, 5671,  …, -100, -100, -100],
        [  72,  550,  284,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  922,  …,   72,   72,   72],
        [ 318,  407,  612,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        …,
        [ 318, 8161, 1312,  …,   72,   72,   72],
        [ 318,  837,  318,  …,   72,   72,   72],
        [ 318, 3729, 4236,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 7091,   318,   257,  …,  -100,  -100,  -100],
        [   72,   481,   307,  …,  -100,  -100,  -100],
        [ 3919,   764,  1312,  …,  -100,  -100,  -100],
        …,
        [27903,   307,   764,  …,  -100,  -100,  -100],
        [20342,   837,   428,  …,  -100,  -100,  -100],
        [   72,   749,  4143,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  837,  …,   72,   72,   72],
        [ 318,  428, 5633,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  407,  922,  …,   72,   72,   72],
        [ 318,  826,  922,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1219,   837,  4043,  …,  -100,  -100,  -100],
        [ 8727,   318,   428,  …,  -100,  -100,  -100],
        [   72,   716,  7926,  …,  -100,  -100,  -100],
        …,
        [27547,   837,   764,  …,  -100,  -100,  -100],
        [  270,   318,   257,  …,  -100,  -100,  -100],
        [ 5562,   318,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   703,    72,  …, 10919, 10919, 10919],
        [  318,   389,   407,  …,    72,    72,    72],
        [  318,   318,   764,  …,    72,    72,    72],
        …,
        [  318,   407,   257,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,  5633,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 4919,  5633,  -100,  …,  -100,  -100,  -100],
        [    0,   345,   389,  …,  -100,  -100,  -100],
        [ 5562,  2933,   502,  …,  -100,  -100,  -100],
        …,
        [  270,   318,  7599,  …,  -100,  -100,  -100],
        [18223,   764,  3127,  …,  -100,  -100,  -100],
        [22850,   407,  5633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  644, 5633,  …,   72,   72,   72],
        [ 318, 1016, 1016,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        …,
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318, 5633, 5633,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10919,  5633,   644,  …,  -100,  -100,  -100],
        [  732,   389,   477,  …,  -100,  -100,  -100],
        [  265,   938,  2644,  …,  -100,  -100,  -100],
        …,
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [ 5832,   287,  3877,  …,  -100,  -100,  -100],
        [15542,   764,   290,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5968,   345,  …, 10919, 10919, 10919],
        [  318,   407,  8046,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72],
        …,
```

```
        [  318,   257,  5633,  …,    72,    72,    72],
        [  318,  1312,   389,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   262,   466,  …,  -100,  -100,  -100],
        [  270,   318,   534,  …,  -100,  -100,  -100],
        [ 8505,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [ 4919,   546,   617,  …,  -100,  -100,  -100],
        [ 5460,   837,   356,  …,  -100,  -100,  -100],
        [   77,   707,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633,   72,  …,   72,   72,   72],
        [ 318,  764,  481,  …,   72,   72,   72],
        [ 318,  588, 3772,  …,   72,   72,   72],
        …,
        [ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  318,  407,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8727,  5633,  -100,  …,  -100,  -100,  -100],
        [  505,  5664,  1312,  …,  -100,  -100,  -100],
        [   72,   561,   307,  …,  -100,  -100,  -100],
        …,
        [43669,   837,   428,  …,  -100,  -100,  -100],
        [ 3919,   340,   318,  …,  -100,  -100,  -100],
        [   72,  6044,  6164,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   345,  …,    72,    72,    72],
        [  318,   284,   764,  …,    72,    72,    72],
        [  318,   407,   764,  …, 10919, 10919, 10919],
        …,
        [  318,  1497,   764,  …,    72,    72,    72],
        [  318,   318,   326,  …,    72,    72,    72],
        [  318,   716,   407,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[22850,   466,   407,  …,  -100,  -100,  -100],
        [   72,   550,   683,  …,  -100,  -100,  -100],
        [  270,   318,  5916,  …,  -100,  -100,  -100],
        …,
        [ 3137,  2652,   994,  …,  -100,  -100,  -100],
        [  392,   644,   857,  …,  -100,  -100,  -100],
        [  392,  1312,   466,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   326,  5633,  …,    72,    72,    72],
        [  318,   326,   428,  …, 10919, 10919, 10919],
        [  318,   318,   318,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,  1337,  …,    72,    72,    72],
        [  318,  1312,  1312,  …, 10919, 10919, 10919],
        [  318,   407,   423,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   318,   340,  …,  -100,  -100,  -100],
        [10919,   318,   477,  …,  -100,  -100,  -100],
        [  392,   326,  2130,  …,  -100,  -100,  -100],
        …,
        [   72,   481,  1011,  …,  -100,  -100,  -100],
        [   72,   531,   837,  …,  -100,  -100,  -100],
        [ 5832,   561,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   644,    72,  …,    72,    72,    72],
        [  318,  5633,  1312,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        …,
        [  318, 23021,   764,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,  1312, 11369,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,  5633,  -100,  …,  -100,  -100,  -100],
        [22850,   407,  5633,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        …,
        [ 1169,  5659, 23021,  …,  -100,  -100,  -100],
        [   76,  1694,   837,  …,  -100,  -100,  -100],
        [22366,   764,   275,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   922,   764,  …,    72,    72,    72],
        [  318,   716,   407,  …,    72,    72,    72],
        [  318,   318,   389,  …,    72,    72,    72],
        …,
        [  318,   764, 10919,  …,  5832,  5832,  5832],
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,   345,  1612,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 1662,   257,  5848,  …,  -100,  -100,  -100],
        [  986,  1312,   373,  …,  -100,  -100,  -100],
        [  568,   644,   345,  …,  -100,  -100,  -100],
        …,
        [46119,   764,  -100,  …,  -100,  -100,  -100],
        [  986,   880,   837,  …,  -100,  -100,  -100],
        [10919,   466,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312, 5832,  …,   72,   72,   72],
        [ 318,  262,  286,  …,   72,   72,   72],
        [ 318,  257, 1256,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318, 1312,  326,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10594,   764,  -100,  …,  -100,  -100,  -100],
        [ 1662,   287,  2166,  …,  -100,  -100,  -100],
        [ 8117,   373,   257,  …,  -100,  -100,  -100],
        …,
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        [   72,   750,   407,  …,  -100,  -100,  -100],
        [ 1219,   837,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  588,  922,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        …,
        [ 318,  764,  563,  …,   72,   72,   72],
        [ 318, 1279, 1279,  …,   72,   72,   72],
        [ 318,  502,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   804,   845,  …,  -100,  -100,  -100],
        [ 3919,  5176,   764,  …,  -100,  -100,  -100],
        [38125,   764,  1312,  …,  -100,  -100,  -100],
        …,
        [ 2364,   837,  2318,  …,  -100,  -100,  -100],
        [ 2197,  5633,     0,  …,  -100,  -100,  -100],
        [ 5832,  2823,   290,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  262, 5633,  …,   72,   72,   72],
        [ 318,  307,  587,  …,   72,   72,   72],
        [ 318, 1312,  466,  …,   72,   72,   72],
        …,
        [ 318,  389,  407,  …,   72,   72,   72],
        [ 318, 1312,  466,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[3003,  318,  764,  …, -100, -100, -100],
        [ 258, 1276,  423,  …, -100, -100, -100],
        [2197,  837, 1521,  …, -100, -100, -100],
        …,
        [  11,  345,  815,  …, -100, -100, -100],
        [3919,  764,  703,  …, -100, -100, -100],
        [1640,  764, -100,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,  7926,  …,    72,    72,    72],
        [  318,   389,   262,  …,    72,    72,    72],
        [  318,   345,   922,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,   922,  …,    72,    72,    72],
        [  318,   764,  1279,  …, 10919, 10919, 10919],
        [  318,   318,   826,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   716,  1107,  …,  -100,  -100,  -100],
        [   13,   356,  1364,  …,  -100,  -100,  -100],
        [ 1324, 13773,  2495,  …,  -100,  -100,  -100],
        …,
        [  270,   318,   257,  …,  -100,  -100,  -100],
        [36410,   922,  5633,  …,  -100,  -100,  -100],
        [   13,   326,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  531,  373,  …,   72,   72,   72],
        [ 318,  826,  764,  …,   72,   72,   72],
        [ 318, 5633, 1312,  …,   72,   72,   72],
        …,
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  389,  764,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,  1239,  1312,  …,  -100,  -100,  -100],
        [ 5562,   318,   340,  …,  -100,  -100,  -100],
        [10919,  8796,  5633,  …,  -100,  -100,  -100],
        …,
        [  986,   645,  2644,  …,  -100,  -100,  -100],
        [   11,   345,  4190,  …,  -100,  -100,  -100],
        [13959,   511,  1633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,    72,  …,    72,    72,    72],
        [  318,   340,  3656,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        …,
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72],
        [  318,  1312, 10919,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  986,   764,  -100,  …,  -100,  -100,  -100],
        [ 4919,   318,   534,  …,  -100,  -100,  -100],
        [ 4868,   268,   837,  …,  -100,  -100,  -100],
        …,
        [ 1350,  2946,  5145,  …,  -100,  -100,  -100],
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        [22366,   764,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   716,  …,    72,    72,    72],
        [  318,  1312,   262,  …,    72,    72,    72],
        [  318,   837,   345,  …,    72,    72,    72],
        …,
```

```
        [  318,   644,    72,  …,    72,    72,    72],
        [  318,   345, 10919,  …,    72,    72,    72],
        [  318,  1312,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 3919,   764,  1312,  …,  -100,  -100,  -100],
        [ 4053,   837,   287,  …,  -100,  -100,  -100],
        [33331,   502,   644,  …,  -100,  -100,  -100],
        …,
        [ 4053,  5633,  -100,  …,  -100,  -100,  -100],
        [30243,   764,  -100,  …,  -100,  -100,  -100],
        [43669,   764,   423,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  407,  …,   72,   72,   72],
        [ 318, 2224, 1312,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        …,
        [ 318,  407,  922,  …,   72,   72,   72],
        [ 318,  826,  922,  …,   72,   72,   72],
        [ 318,  257,  257,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[12518,   673,   318,  …,  -100,  -100,  -100],
        [   64,   308,  2644,  …,  -100,  -100,  -100],
        [ 1462,  4534,   764,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,   845,  …,  -100,  -100,  -100],
        [ 5562,   318,   257,  …,  -100,  -100,  -100],
        [  270,   373,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  494,  764,  …,   72,   72,   72],
        [ 318,  345,  760,  …,   72,   72,   72],
        …,
        [ 318,  644,  764,  …,   72,   72,   72],
        [ 318,  389,  407,  …,   72,   72,   72],
        [ 318,  257,  922,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   345,   764,  …,  -100,  -100,  -100],
        [   73,   418,   494,  …,  -100,  -100,  -100],
        [  265,  1551,   356,  …,  -100,  -100,  -100],
        …,
        [10919,  5633,   645,  …,  -100,  -100,  -100],
        [13893,   484,   389,  …,  -100,  -100,  -100],
        [ 5661,   318,   523,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   546,  …,    72,    72,    72],
        [  318,   898,  1545,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,   339,  …, 10919, 10919, 10919],
        [  318,  1312,   284,  …,    72,    72,    72],
        [  318,   644,   881,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   389,  3375,  …,  -100,  -100,  -100],
        [ 1820,   845,   922,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        …,
        [14363, 33024,  5633,  …,  -100,  -100,  -100],
        [41599,   837,  1392,  …,  -100,  -100,  -100],
        [  568,  2644,  2495,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   329,  …,    72,    72,    72],
        [  318,  1312,   716,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        …,
        [  318,   764,   262,  …,    72,    72,    72],
        [  318,  1312,  5832,  …,    72,    72,    72],
        [  318,   764,   764,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[   72,   716,  2045,  …,  -100,  -100,  -100],
        [ 4053,  2644,  1312,  …,  -100,  -100,  -100],
        [ 2958,   319,   764,  …,  -100,  -100,  -100],
        …,
        [ 4598,   340,   319,  …,  -100,  -100,  -100],
        [ 3919,   764,  -100,  …,  -100,  -100,  -100],
        [24571,   898,  2910,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   257,  2126,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        [  318,  1521, 13893,  …, 22850, 22850, 22850],
        …,
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   531,   837,  …,    72,    72,    72],
        [  318,   345,   716,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  258,   550,   281,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [22850,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [  738,   605,   764,  …,  -100,  -100,  -100],
        [ 2339,  1312,   531,  …,  -100,  -100,  -100],
        [   72,   892,  1312,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  716,  345,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  502,  764,  …,   72,   72,   72],
        [ 318,  760, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4053,  1312,  2911,  …,  -100,  -100,  -100],
        [ 3919,   837,  1312,  …,  -100,  -100,  -100],
        [43669,   764,   606,  …,  -100,  -100,  -100],
        …,
        [14774,   290,  5293,  …,  -100,  -100,  -100],
        [ 3137,  2666,   340,  …,  -100,  -100,  -100],
        [ 4598,   345,   892,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  319, 1204,  …,   72,   72,   72],
        [ 318,  257,  764,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        …,
        [ 318, 1312,  481,  …,   72,   72,   72],
        [ 318,  651,  345,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 6738,   783,   616,  …,  -100,  -100,  -100],
        [ 5832,  1392,   502,  …,  -100,  -100,  -100],
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        …,
        [   13,   788,  1312,  …,  -100,  -100,  -100],
        [ 5171,  1312,  1037,  …,  -100,  -100,  -100],
        [31373,   837,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,   764,  …,    72,    72,    72],
        [  318,   837,   764,  …,    72,    72,    72],
        [  318,  5490,   523,  …,    72,    72,    72],
        …,
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   466,   339,  …, 10919, 10919, 10919],
        [  318,  1279,    72,  …,   986,   986,   986]], device='cuda:0')
First 10 labels: tensor([[   64, 12379, 13560,  …,  -100,  -100,  -100],
        [20342,   837,   764,  …,  -100,  -100,  -100],
        [ 4598,   407,   307,  …,  -100,  -100,  -100],
        …,
        [14323, 20053,   764,  …,  -100,  -100,  -100],
        [ 8524,  1521,   318,  …,  -100,  -100,  -100],
        [  282,  5633,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   644,   644,  …, 10919, 10919, 10919],
        [  318,   837,   345,  …,    72,    72,    72],
        [  318,   783,   837,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   837,   345,  …,    72,    72,    72],
        [  318,  5633,  5633,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,  5633,     0,  …,  -100,  -100,  -100],
        [    0, 11752,   837,  …,  -100,  -100,  -100],
        [  292,   286,   783,  …,  -100,  -100,  -100],
        …,
        [ 1878,  7086,   523,  …,  -100,  -100,  -100],
        [ 2958,   319,  5145,  …,  -100,  -100,  -100],
        [ 8727,   607,  5633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  517, 1517,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  466,  339,  …,   72,   72,   72],
        …,
        [ 318, 1975,  764,  …,   72,   72,   72],
        [ 318,  345,  389,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3137,   530,  1310,  …,  -100,  -100,  -100],
        [ 8505,   837,  2276,  …,  -100,  -100,  -100],
        [ 8524,  1521,   857,  …,  -100,  -100,  -100],
        …,
        [   72,  2314, 18044,  …,  -100,  -100,  -100],
        [ 4360,   837,   345,  …,  -100,  -100,  -100],
        [ 4033, 26261,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633, 8046,  …,   72,   72,   72],
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  588,  284,  …,   72,   72,   72],
        …,
        [ 318,  407,  345,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  284,  345,  …, 5832, 5832, 5832]], device='cuda:0')
First 10 labels: tensor([[  271,   340,   534,  …,  -100,  -100,  -100],
        [ 3919,   764,   475,  …,  -100,  -100,  -100],
        [   72,   561,   588,  …,  -100,  -100,  -100],
        …,
        [   11,   466,   407,  …,  -100,  -100,  -100],
        [47408,   683,  3436,  …,  -100,  -100,  -100],
        [ 2978, 13886,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  826, 1312,  …,   72,   72,   72],
        [ 318,  428, 5633,  …,   72,   72,   72],
        [ 318,  644,   72,  …,   72,   72,   72],
        …,
```

```
        [ 318,  257,  764,  …,   72,   72,   72],
        [ 318,  307,  284,  …,   72,   72,   72],
        [ 318,  407,  262,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5562,   318,   644,  …,  -100,  -100,  -100],
        [ 8727,   318,   326,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [  258,   318,   826,  …,  -100,  -100,  -100],
        [  270,   561,  1283,  …,  -100,  -100,  -100],
        [   72,   481, 24248,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,  1804,  …,    72,    72,    72],
        [  318,   764,   345,  …,    72,    72,    72],
        [  318,   326,  1612,  …, 10919, 10919, 10919],
        …,
        [  318,   588,  7926,  …,    72,    72,    72],
        [  318,   284,   760,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   389,   345,  …,  -100,  -100,  -100],
        [11338,   326,   764,  …,  -100,  -100,  -100],
        [10919,   857,   340,  …,  -100,  -100,  -100],
        …,
        [   72,  1254,   845,  …,  -100,  -100,  -100],
        [   72,   765,   284,  …,  -100,  -100,  -100],
        [  270,   318,   502,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  262, 1110,  …,   72,   72,   72],
        [ 318,  837,  318,  …,   72,   72,   72],
        [ 318,  389,  407,  …,   72,   72,   72],
        …,
        [ 318,  837,  345,  …,   72,   72,   72],
        [ 318,  345,  466,  …,   72,   72,   72],
        [ 318,  502,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[29332,   286,   262,  …,  -100,  -100,  -100],
        [  439,   826,   340,  …,  -100,  -100,  -100],
        [  986,   345,   389,  …,  -100,  -100,  -100],
        …,
        [   67,  2507,  5633,  …,  -100,  -100,  -100],
        [10919,   466,   356,  …,  -100,  -100,  -100],
        [ 3919, 10833,   607,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   307,  …,    72,    72,    72],
        [  318,  1312,  1654,  …,    72,    72,    72],
        [  318,   286,   502,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,   475,  …,    72,    72,    72],
        [  318,   326,  1438,  …, 10919, 10919, 10919],
        [  318,   764,   503,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   481,   407,  …,  -100,  -100,  -100],
        [19532,  5145,   307,  …,  -100,  -100,  -100],
        [ 1169,  8153,  2921,  …,  -100,  -100,  -100],
        …,
        [29810,  3763,   837,  …,  -100,  -100,  -100],
        [10919,   318,   534,  …,  -100,  -100,  -100],
        [ 7091,  2314,   651,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  588,  257,  …,   72,   72,   72],
        [ 318,  826,  922,  …,   72,   72,   72],
        [ 318,  257,  257,  …,   72,   72,   72],
        …,
        [ 318,  257,  764,  …,   72,   72,   72],
        [ 318,  257, 1256,  …,   72,   72,   72],
        [ 318,  284,  651,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[5832,  719,  588,  …, -100, -100, -100],
        [5562,  318,  257,  …, -100, -100, -100],
        [ 258,  318,  407,  …, -100, -100, -100],
        …,
        [5832,  651, 1327,  …, -100, -100, -100],
        [8117,  318,  257,  …, -100, -100, -100],
        [5832,  761,  284,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  407,  502,  …,   11,   11,   11],
        …,
        [ 318,  345,  547,  …,   72,   72,   72],
        [ 318,  345,  837,  …,   72,   72,   72],
        [ 318,  837,   72,  …, 7091, 7091, 7091]], device='cuda:0')
First 10 labels: tensor([[25991,   764,  1312,  …,  -100,  -100,  -100],
        [   72,   716, 11040,  …,  -100,  -100,  -100],
        [  270,   318,   284,  …,  -100,  -100,  -100],
        …,
        [   72,  1807,   356,  …,  -100,  -100,  -100],
        [ 4360,   837,  6478,  …,  -100,  -100,  -100],
        [31373,  5633,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  340, 5633,  …,   72,   72,   72],
        [ 318, 1312, 5832,  …,   72,   72,   72],
        [ 318,  326, 5633,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  340,  …, 5832, 5832, 5832],
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  345, 1804,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4919,   318,   340,  …,  -100,  -100,  -100],
        [ 3919,   764,  -100,  …,  -100,  -100,  -100],
        [10919,   318,   428,  …,  -100,  -100,  -100],
        …,
        [  805,   837,  6044,  …,  -100,  -100,  -100],
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        [10919,   389,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837,  764,  …,   72,   72,   72],
        [ 318,  284,  651,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        …,
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  257,  922,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 9503,  4066,  6203,  …,  -100,  -100,  -100],
        [   13,  1392,   284,  …,  -100,  -100,  -100],
        [  929,   764,   356,  …,  -100,  -100,  -100],
        …,
        [ 1219, 10194,   837,  …,  -100,  -100,  -100],
        [ 4053,   837,   612,  …,  -100,  -100,  -100],
        [ 7091,   318,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1654, 5633,  …,   72,   72,   72],
        [ 318,  466,  407,  …,   72,   72,   72],
        [ 318,  837,  345,  …,   72,   72,   72],
        …,
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318,  407, 4950,  …,   72,   72,   72],
        [ 318,  284,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  533,   345,   290,  …,  -100,  -100,  -100],
        [29810,  1312,   466,  …,  -100,  -100,  -100],
        [   77, 12321,  5145,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,   407,  …,  -100,  -100,  -100],
        [ 5832,   389,   523,  …,  -100,  -100,  -100],
        [ 1136,   736,   994,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1295,   764,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   764,  1312,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,   466,  5633,  …,    72,    72,    72],
        [  318,  1997,   835,  …,    72,    72,    72],
        [  318,   407,   284,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  64, 7932, 1499,  …, -100, -100, -100],
        [ 270,  318, 1593,  …, -100, -100, -100],
        [7012, 9618,  764,  …, -100, -100, -100],
        …,
        [4919,  881, 6099,  …, -100, -100, -100],
        [ 271,  612, 1194,  …, -100, -100, -100],
        [5832,  389, 1016,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  407,  345,  …,   72,   72,   72],
        [ 318,  467,  764,  …,   72,   72,   72],
        …,
        [ 318,  407,  612,  …,   72,   72,   72],
        [ 318,  345, 5633,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3919,   837,   345,  …,  -100,  -100,  -100],
        [   72,   716,  1654,  …,  -100,  -100,  -100],
        [ 1456,   356,   389,  …,  -100,  -100,  -100],
        …,
        [   72,   481,   307,  …,  -100,  -100,  -100],
        [10919,   389,   484,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   262,   837,  …, 10919, 10919, 10919],
        [  318,   510,  1256,  …,    72,    72,    72],
        [  318,   284,   764,  …,    72,    72,    72],
        …,
        [  318,  2073,  1310,  …,    11,    11,    11],
        [  318,   766,   345,  …,    72,    72,    72],
        [  318,   345,  1438,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  13, 3952,  263,  …, -100, -100, -100],
        [7091, 9617,  257,  …, -100, -100, -100],
        [  72, 1392,  345,  …, -100, -100, -100],
        …,
        [ 273, 2130,  257,  …, -100, -100, -100],
        [  72,  460, 1560,  …, -100, -100, -100],
        [  72, 5465,  534,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   922,   257,  …, 10919, 10919, 10919],
        [  318,  1312,   318,  …,    72,    72,    72],
        [  318,   826,   764,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   764,  5832,  …,    72,    72,    72],
        [  318,   318,   389,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  258,   257,  2644,  …,  -100,  -100,  -100],
        [ 4053,   837,   340,  …,  -100,  -100,  -100],
        [ 5562,   318,  1049,  …,  -100,  -100,  -100],
        …,
        [   72,   716,  7926,  …,  -100,  -100,  -100],
        [15542,   764,  -100,  …,  -100,  -100,  -100],
        [   11,   703,  1468,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837,  345,  …,   72,   72,   72],
        [ 318,  508, 1312,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72],
        …,
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  345,  318,  …,   72,   72,   72],
        [ 318,  262, 5633,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1073,   293,   837,  …,  -100,  -100,  -100],
        [ 1659,   661,   764,  …,  -100,  -100,  -100],
        [13552,  2644,   340,  …,  -100,  -100,  -100],
        …,
        [43669,   837,   764,  …,  -100,  -100,  -100],
        [  265,  1551,   340,  …,  -100,  -100,  -100],
        [10919,   546,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  262,  …,   72,   72,   72],
        [ 318,  466,  340,  …,   72,   72,   72],
        [ 318,  508,  644,  …,   72,   72,   72],
        …,
        [ 318,  407,  764,  …,   72,   72,   72],
        [ 318, 5968,  318,  …,   72,   72,   72],
        [ 318,  764,   72,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  272, 12069,   656,  …,  -100,  -100,  -100],
        [ 5832,   460,  4483,  …,  -100,  -100,  -100],
        [ 1169,   582,  2993,  …,  -100,  -100,  -100],
        …,
        [   72,   716,  4305,  …,  -100,  -100,  -100],
        [ 8727,   262, 17118,  …,  -100,  -100,  -100],
        [ 1662,  1107,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   994,  …,    72,    72,    72],
        [  318,   345,  1804,  …, 10919, 10919, 10919],
        [  318,  1312,   345,  …,    72,    72,    72],
        …,
```

```
        [  318,  1392,  2126,  …,    72,    72,    72],
        [  318,   345,   423,  …,    72,    72,    72],
        [  318,   345,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   714,  2652,  …,  -100,  -100,  -100],
        [10919,   389,   345,  …,  -100,  -100,  -100],
        [44107,   764,   423,  …,  -100,  -100,  -100],
        …,
        [   72,   423,   281,  …,  -100,  -100,  -100],
        [10594,   407,   356,  …,  -100,  -100,  -100],
        [   72,  2497,   683,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   257,   262,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919],
        [  318,   345,   345,  …,    72,    72,    72],
        …,
        [  318,   290,   318,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,   407,   760,  …,   986,   986,   986]], device='cuda:0')
First 10 labels: tensor([[  258,   318,   379,  …,  -100,  -100,  -100],
        [10919,   318,   326,  …,  -100,  -100,  -100],
        [ 4919,  1282,  5633,  …,  -100,  -100,  -100],
        …,
        [ 5562,  2042,  1097,  …,  -100,  -100,  -100],
        [43669,   837,   826,  …,  -100,  -100,  -100],
        [  986,   466,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  340, 1256,  …,   72,   72,   72],
        [ 318,  716,  407,  …,   72,   72,   72],
        …,
        [ 318,  407,  262,  …,   72,   72,   72],
        [ 318, 5633,  764,  …,   72,   72,   72],
        [ 318,  257,  835,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,  2644,  -100,  …,  -100,  -100,  -100],
        [ 9930,  1234,   257,  …,  -100,  -100,  -100],
        [  986,  1312,   481,  …,  -100,  -100,  -100],
        …,
        [  270,   318,   319,  …,  -100,  -100,  -100],
        [23442,  1657,  7077,  …,  -100,  -100,  -100],
        [ 8117,   318,   645,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  764,  …,   72,   72,   72],
        [ 318,  407,  307,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        …,
```

```
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  644,   72,  …, 5832, 5832, 5832],
        [ 318,  764,  262,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  258,   318, 23036,  …,  -100,  -100,  -100],
        [   72,   481,   407,  …,  -100,  -100,  -100],
        [43669,   764,  1312,  …,  -100,  -100,  -100],
        …,
        [ 2958,   319,   764,  …,  -100,  -100,  -100],
        [27485,  5633,  -100,  …,  -100,  -100,  -100],
        [21260,   503,   351,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  765, 1807,  …,   72,   72,   72],
        [ 318,  284, 4144,  …, 5832, 5832, 5832],
        …,
        [ 318,  760,  340,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  502, 5633,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[38204, 25825,   530,  …,  -100,  -100,  -100],
        [   72,   655,  1239,  …,  -100,  -100,  -100],
        [  986,   761,   257,  …,  -100,  -100,  -100],
        …,
        [ 4598,   345,   588,  …,  -100,  -100,  -100],
        [27780,   502,   764,  …,  -100,  -100,  -100],
        [ 5832,  1975,   502,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   339,   262,  …,    72,    72,    72],
        [  318,   764, 21202,  …,    72,    72,    72],
        [  318,  1312,   716,  …,    72,    72,    72],
        …,
        [  318,   318,   716,  …,    72,    72,    72],
        [  318,  1521, 13893,  …,    72,    72,    72],
        [  318,   764,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 8524,   531,   319,  …,  -100,  -100,  -100],
        [   66,  8825,   393,  …,  -100,  -100,  -100],
        [ 4053,   837,  1312,  …,  -100,  -100,  -100],
        …,
        [ 3137,   326,  1312,  …,  -100,  -100,  -100],
        [22850,  5633,  -100,  …,  -100,  -100,  -100],
        [   72,  2686,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  389,  484,  …,   72,   72,   72],
        [ 318,  257,  262,  …,   72,   72,   72],
        [ 318,  407,  530,  …,   72,   72,   72],
        …,
```

```
        [ 318,  257,  257,  …,   72,   72,   72],
        [ 318,  286,  588,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[18927,   484,  2644,  …,  -100,  -100,  -100],
        [  258,   318,   287,  …,  -100,  -100,  -100],
        [ 5832,   389,   262,  …,  -100,  -100,  -100],
        …,
        [ 8117,   373,   407,  …,  -100,  -100,  -100],
        [   72,  3297,   286,  …,  -100,  -100,  -100],
        [ 2197,   837,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,  522,  …,   72,   72,   72],
        [ 318,  307,  257,  …,   72,   72,   72],
        [ 318,  588,  284,  …,   72,   72,   72],
        …,
        [ 318, 1392, 1775,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   892,   285,  …,  -100,  -100,  -100],
        [ 5661,   743,   307,  …,  -100,  -100,  -100],
        [   72,   561,   765,  …,  -100,  -100,  -100],
        …,
        [   72,   423,   407,  …,  -100,  -100,  -100],
        [19650,   764,  -100,  …,  -100,  -100,  -100],
        [48937,   866,  2644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1279,  389,  …,   72,   72,   72],
        [ 318,  284,  467,  …,   72,   72,   72],
        [ 318, 1807,  340,  …,   72,   72,   72],
        …,
        [ 318, 5633,  644,  …,   72,   72,   72],
        [ 318,  257,  356,  …,   72,   72,   72],
        [ 318,  345, 1804,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 2197,  5633,   356,  …,  -100,  -100,  -100],
        [ 5832,   765,   284,  …,  -100,  -100,  -100],
        [   72,  1464,   996,  …,  -100,  -100,  -100],
        …,
        [ 1929,    64,  5633,  …,  -100,  -100,  -100],
        [  732,  1392,   764,  …,  -100,  -100,  -100],
        [10919,   389,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5490,  284,  …,   72,   72,   72],
        [ 318,  837,  764,  …,   72,   72,   72],
        [ 318,  423, 1683,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318, 5633, 5633,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4598,   407,  6044,  …,  -100,  -100,  -100],
        [20342,   837, 12302,  …,  -100,  -100,  -100],
        [22366,  1312,   561,  …,  -100,  -100,  -100],
        …,
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        [ 8505,   837,  1312,  …,  -100,  -100,  -100],
        [ 9776,   673,  2801,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633, 1312,  …,   72,   72,   72],
        [ 318, 5633, 1312,  …,   11,   11,   11],
        [ 318,  407,  284,  …,   72,   72,   72],
        …,
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  257,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10919,   892,  5633,  …,  -100,  -100,  -100],
        [   11,   826,  5633,  …,  -100,  -100,  -100],
        [  270,   318,  1972,  …,  -100,  -100,  -100],
        …,
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [ 3506,   764,  -100,  …,  -100,  -100,  -100],
        [  258,   318,  1016,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   837,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,  1312,   318,  …,    72,    72,    72],
        …,
        [  318,   826,   922,  …,    72,    72,    72],
        [  318,   257,   835,  …,    72,    72,    72],
        [  318,   262,   345,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[  392,   837,   523,  …,  -100,  -100,  -100],
        [41599,   837,  2051,  …,  -100,  -100,  -100],
        [ 1219,   837,   326,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,   257,  …,  -100,  -100,  -100],
        [ 8117,   318,   645,  …,  -100,  -100,  -100],
        [10919,   546,  5633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  588,  257,  …,   72,   72,   72],
        [ 318,  534,  345,  …,   72,   72,   72],
        …,
```

```
        [ 318,  389,  284,  …,   72,   72,   72],
        [ 318,  389,  423,  …,   72,   72,   72],
        [ 318, 1392,  467,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 7513,  4107,   764,  …,  -100,  -100,  -100],
        [ 9688,  7205,   588,  …,  -100,  -100,  -100],
        [ 5832,  4425,  5145,  …,  -100,  -100,  -100],
        …,
        [22850,   345,  1392,  …,  -100,  -100,  -100],
        [25991,   345,   815,  …,  -100,  -100,  -100],
        [   72,   423,   284,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,  284,  …,   72,   72,   72],
        [ 318,  389, 1312,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        …,
        [ 318, 1654,  826,  …,   72,   72,   72],
        [ 318,  764,  345,  …,   72,   72,   72],
        [ 318,  345,  547,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  533,   407,  1016,  …,  -100,  -100,  -100],
        [  361,   345,   892,  …,  -100,  -100,  -100],
        [  548,   764,  -100,  …,  -100,  -100,  -100],
        …,
        [  533,   345,   477,  …,  -100,  -100,  -100],
        [14261, 13076,  5145,  …,  -100,  -100,  -100],
        [38947,   618,   356,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1279,  284,  …, 5832, 5832, 5832],
        [ 318, 1312,  837,  …,   72,   72,   72],
        [ 318,  257,  764,  …,   72,   72,   72],
        …,
        [ 318, 1312,  467,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[23151,  5633,  3585,  …,  -100,  -100,  -100],
        [43669,   764,   880,  …,  -100,  -100,  -100],
        [  258,   318, 16039,  …,  -100,  -100,  -100],
        …,
        [10155,   611,   356,  …,  -100,  -100,  -100],
        [   76,   522,   837,  …,  -100,  -100,  -100],
        [  439,   826,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,   318,  …,    72,    72,    72],
        [  318,   345,   345,  …,    72,    72,    72],
        [  318,   286,  2642,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,   764,   284,  …,  5832,  5832,  5832],
        [  318, 11060,   837,  …,    72,    72,    72],
        [  318,   345,   389,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[38171,   837,   326,  …,  -100,  -100,  -100],
        [22850,   750,   407,  …,  -100,  -100,  -100],
        [ 4758,   636,   318,  …,  -100,  -100,  -100],
        …,
        [ 2339,   345,   973,  …,  -100,  -100,  -100],
        [ 1640, 27636, 11060,  …,  -100,  -100,  -100],
        [ 4919,  1282,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  307,  …,   72,   72,   72],
        [ 318,  345,  345,  …,   72,   72,   72],
        [ 318, 1524,  837,  …,   72,   72,   72],
        …,
        [ 318,  262,  764,  …,   72,   72,   72],
        [ 318,  866,  262,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   815,   407,  …,  -100,  -100,  -100],
        [  568,   837,   783,  …,  -100,  -100,  -100],
        [14108,  1029,  1108,  …,  -100,  -100,  -100],
        …,
        [ 2188,   284, 12296,  …,  -100,  -100,  -100],
        [ 1996,   502,   319,  …,  -100,  -100,  -100],
        [ 1820,  5770,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …,    72,    72,    72],
        [  318,   345,  1612,  …, 10919, 10919, 10919],
        [  318,   407,   423,  …,    72,    72,    72],
        …,
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72],
        [  318,   407,  1016,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   760,   764,  …,  -100,  -100,  -100],
        [10919,   466,   345,  …,  -100,  -100,  -100],
        [ 9930,   714,   407,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,  9105,  …,  -100,  -100,  -100],
        [38125,   764,  -100,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   286,  6891,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,  1312,   345,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,   257,  …,    72,    72,    72],
        [  318,   764,   307,  …,    72,    72,    72],
        [  318,   764, 31373,  …,    11,    11,    11]], device='cuda:0')
First 10 labels: tensor([[29214,  6508,   286,  …,  -100,  -100,  -100],
        [ 9930,   616,   764,  …,  -100,  -100,  -100],
        [17231,   837,   423,  …,  -100,  -100,  -100],
        …,
        [13345,   502,   355,  …,  -100,  -100,  -100],
        [ 1169,  6253,   481,  …,  -100,  -100,  -100],
        [31373,   764,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  307,  345,  …,   72,   72,   72],
        [ 318,  644,   72,  …,   72,   72,   72],
        [ 318,  389,  284,  …,   72,   72,   72],
        …,
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318, 1016,  284,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  732,   481,  4255,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        [  392,   345,   423,  …,  -100,  -100,  -100],
        …,
        [ 1219,   837,   837,  …,  -100,  -100,  -100],
        [  732,   389,   546,  …,  -100,  -100,  -100],
        [  270,   318,  4998,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  407, 1016,  …, 5832, 5832, 5832],
        [ 318,  345, 1804,  …,   72,   72,   72],
        …,
        [ 318,  764,  257,  …,   72,   72,   72],
        [ 318,  428, 5633,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3919,   837,  1312,  …,  -100,  -100,  -100],
        [ 5832,   389,   407,  …,  -100,  -100,  -100],
        [10919,   389,   345,  …,  -100,  -100,  -100],
        …,
        [13698,   673,   318,  …,  -100,  -100,  -100],
        [ 8727,   318,   339,  …,  -100,  -100,  -100],
        [   13,   645, 11881,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5633,   345,  …, 10919, 10919, 10919],
        [  318,  1312,   389,  …,    72,    72,    72],
        [  318,   262,   835,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,  1312,  …,    72,    72,    72],
        [  318,   284,   764,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 2339,   644,  5633,  …,  -100,  -100,  -100],
        [42994,   764,   661,  …,  -100,  -100,  -100],
        [  986,   287,   257,  …,  -100,  -100,  -100],
        …,
        [  270,   318,   644,  …,  -100,  -100,  -100],
        [   72,   550,  2279,  …,  -100,  -100,  -100],
        [ 8505,   764,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5145,   318,  …,    72,    72,    72],
        [  318,  5633,  1279,  …, 10919, 10919, 10919],
        [  318,   407,   764,  …,    72,    72,    72],
        …,
        [  318,   716,   407,  …,    72,    72,    72],
        [  318,   257,   284,  …,    72,    72,    72],
        [  318,   407,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   11,  5145,  1909,  …,  -100,  -100,  -100],
        [10919,  3850,  5633,  …,  -100,  -100,  -100],
        [   72,   716,  8179,  …,  -100,  -100,  -100],
        …,
        [    0,  1312,   714,  …,  -100,  -100,  -100],
        [  258,   373, 31856,  …,  -100,  -100,  -100],
        [22850,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  407,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        …,
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  531, 1312,  …,   72,   72,   72],
        [ 318, 5968,  318,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   30,   339,   318,  …,  -100,  -100,  -100],
        [ 3919,   764,   356,  …,  -100,  -100,  -100],
        [   72,   766,   764,  …,  -100,  -100,  -100],
        …,
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [   72,  1239,  1807,  …,  -100,  -100,  -100],
        [10919,   262,  5968,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312, 1576,  …,   72,   72,   72],
        [ 318,  407,  345,  …,   72,   72,   72],
        [ 318, 5490,  892,  …,   72,   72,   72],
        …,
```

```
        [ 318,  898,  764,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[4053,  837, 1969,  …, -100, -100, -100],
        [  72,  481, 6594,  …, -100, -100, -100],
        [4598,  407,  345,  …, -100, -100, -100],
        …,
        [ 259,  465, 7962,  …, -100, -100, -100],
        [  73,  539,  837,  …, -100, -100, -100],
        [5832,  389,  407,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1016,  262,  …,   72,   72,   72],
        [ 318,  389,  340,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72],
        …,
        [ 318, 1654, 5633,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1392,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  732,   389,   329,  …,  -100,  -100,  -100],
        [  392,   345,   546,  …,  -100,  -100,  -100],
        [   72,   716,   691,  …,  -100,  -100,  -100],
        …,
        [  533,   345, 11029,  …,  -100,  -100,  -100],
        [ 2395,    69,   837,  …,  -100,  -100,  -100],
        [   72,   423,   587,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  466, 5633,  …,   72,   72,   72],
        [ 318,  407,  257,  …,   72,   72,   72],
        [ 318, 1312,  345,  …,   72,   72,   72],
        …,
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312, 1107,  …,   72,   72,   72],
        [ 318,  407,  922,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10919,   345,   910,  …,  -100,  -100,  -100],
        [ 5832,   389,   884,  …,  -100,  -100,  -100],
        [ 4053,   837,   750,  …,  -100,  -100,  -100],
        …,
        [13345,   502,   764,  …,  -100,  -100,  -100],
        [41599,   764,   407,  …,  -100,  -100,  -100],
        [  270,   318,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  503,  764,  …,   72,   72,   72],
        [ 318,  345,  389,  …,   72,   72,   72],
        …,
```

```
        [ 318,  588,  340,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  389,  318,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[24209,   764,  -100,  …,  -100,  -100,  -100],
        [ 8968,   340,   503,  …,  -100,  -100,  -100],
        [ 4360,   611,   345,  …,  -100,  -100,  -100],
        …,
        [19188,   345,   466,  …,  -100,  -100,  -100],
        [ 5832,   607,   837,  …,  -100,  -100,  -100],
        [ 3003,   262,  5968,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5145,    0,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        …,
        [ 318,  407, 1312,  …,   72,   72,   72],
        [ 318,  407,  284,  …,   72,   72,   72],
        [ 318,  826,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   0, 5145, -100,  …, -100, -100, -100],
        [3919,  837, 1312,  …, -100, -100, -100],
        [8505,  764, -100,  …, -100, -100, -100],
        …,
        [  72,  716, 7787,  …, -100, -100, -100],
        [ 270,  318, 1327,  …, -100, -100, -100],
        [5562,  318,  826,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        …,
        [ 318,  407,  407,  …,   72,   72,   72],
        [ 318, 1975, 1312,  …,   72,   72,   72],
        [ 318,  407,  764,  …, 5832, 5832, 5832]], device='cuda:0')
First 10 labels: tensor([[ 1219,   764,  -100,  …,  -100,  -100,  -100],
        [13345,   502,   764,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        …,
        [48783,   318,  5543,  …,  -100,  -100,  -100],
        [   72,  2314,   764,  …,  -100,  -100,  -100],
        [   72,   716,  3734,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  514,  651,  …,   72,   72,   72],
        [ 318,  407,  284,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  345,  502,  …, 5832, 5832, 5832],
        [ 318,  588, 1312,  …,   72,   72,   72],
        [ 318, 1517,  508,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[29688,  1309,   502,  …,  -100,  -100,  -100],
        [   72,   373,  1016,  …,  -100,  -100,  -100],
        [   11,  2048,   318,  …,  -100,  -100,  -100],
        …,
        [12860,   326,   284,  …,  -100,  -100,  -100],
        [11031,   286,   764,  …,  -100,  -100,  -100],
        [ 1169,   976,  3516,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  764,  …,   72,   72,   72],
        [ 318, 1755,  764,  …,   72,   72,   72],
        [ 318,  257,  284,  …,   72,   72,   72],
        …,
        [ 318,  760,  284,  …,   72,   72,   72],
        [ 318,  764,  502,  …,   72,   72,   72],
        [ 318, 1312, 2357,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3549,   588,   257,  …,  -100,  -100,  -100],
        [ 1659,   262,  1803,  …,  -100,  -100,  -100],
        [ 8117,   318,  2147,  …,  -100,  -100,  -100],
        …,
        [ 4598,   345,   765,  …,  -100,  -100,  -100],
        [21928,   340,   329,  …,  -100,  -100,  -100],
        [ 8226,   837,   279,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  307,  …,   72,   72,   72],
        [ 318, 1392, 1775,  …,   72,   72,   72],
        [ 318,  587,  703,  …,   72,   72,   72],
        …,
        [ 318,  389,  407,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  318,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[13893,   612,   481,  …,  -100,  -100,  -100],
        [   72,   423,  1239,  …,  -100,  -100,  -100],
        [ 5832,  1683,  4003,  …,  -100,  -100,  -100],
        …,
        [ 1219,   484,   466,  …,  -100,  -100,  -100],
        [  421,   924,   837,  …,  -100,  -100,  -100],
        [37814,   340,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …,    72,    72,    72],
        [  318,   345, 10919,  …,    72,    72,    72],
        [  318,   345,  1282,  …,    72,    72,    72],
        …,
```

```
        [  318,   326,  5633,  …, 10919, 10919, 10919],
        [  318,   764,   764,  …,    72,    72,    72],
        [  318,   257,   475,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 1069, 24342,   764,  …,  -100,  -100,  -100],
        [19472,   764,  -100,  …,  -100,  -100,  -100],
        [ 3003,   750,   326,  …,  -100,  -100,  -100],
        …,
        [10919,   318,   510,  …,  -100,  -100,  -100],
        [   72,   760,   326,  …,  -100,  -100,  -100],
        [  270,   373,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   644,    72,  …,    72,    72,    72],
        [  318,   345, 10919,  …,    72,    72,    72],
        [  318,  5770,   837,  …,    72,    72,    72],
        …,
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   257,   764,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[10919,  5633,  -100,  …,  -100,  -100,  -100],
        [11084,   764,  -100,  …,  -100,  -100,  -100],
        [ 1219,   616,  5770,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,  8258,  …,  -100,  -100,  -100],
        [  258,   318,  9105,  …,  -100,  -100,  -100],
        [10919,   373,   326,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  345,  345,  …,   72,   72,   72],
        [ 318,  407,  922,  …,   72,   72,   72],
        …,
        [ 318,  651,  345,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  407,  922,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   760,   508,  …,  -100,  -100,  -100],
        [22850,   750,   407,  …,  -100,  -100,  -100],
        [  270,   318,   257,  …,  -100,  -100,  -100],
        …,
        [ 5171,  1312,  1037,  …,  -100,  -100,  -100],
        [18223,   837,  5176,  …,  -100,  -100,  -100],
        [ 5832,   389,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  389,  …,   72,   72,   72],
        [ 318,  389,  407,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  764,  644,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   13,  1893,   356,  …,  -100,  -100,  -100],
        [  568,   345,   389,  …,  -100,  -100,  -100],
        [43669,   764,  1312,  …,  -100,  -100,  -100],
        …,
        [ 8505,   837, 15967,  …,  -100,  -100,  -100],
        [13261,  1510,  5633,  …,  -100,  -100,  -100],
        [43745,   837,   649,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   345,  …,    72,    72,    72],
        [  318,  1312,   691,  …,    72,    72,    72],
        [  318,  1560,   284,  …,    72,    72,    72],
        …,
        [  318,  5490,   837,  …,    72,    72,    72],
        [  318,   407,   910,  …,    72,    72,    72],
        [  318,   407,  1312,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[  72,  481, 1085,  …, -100, -100, -100],
        [  72, 4724,  262,  …, -100, -100, -100],
        [5171,  345, 1561,  …, -100, -100, -100],
        …,
        [4598,  407, 5490,  …, -100, -100, -100],
        [ 258,  750,  407,  …, -100, -100, -100],
        [  72,  716, 2644,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,   345,  …,    72,    72,    72],
        [  318,   345,  5633,  …, 10919, 10919, 10919],
        [  318,   503,  1312,  …,    72,    72,    72],
        …,
        [  318,   837,   345,  …,    72,    72,    72],
        [  318,   345,  1804,  …,    72,    72,    72],
        [  318,   645,   621,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 4360,   466,   407,  …,  -100,  -100,  -100],
        [10919,   389,   484,  …,  -100,  -100,  -100],
        [ 1929,  3949,   764,  …,  -100,  -100,  -100],
        …,
        [ 2958,   319,   837,  …,  -100,  -100,  -100],
        [22850,   389,   345,  …,  -100,  -100,  -100],
        [ 8117,   389,   517,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,   345,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72],
        …,
```

```
        [  318,   837,  3872,  …,    72,    72,    72],
        [  318,   764,   262,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[ 3137,  3505,   837,  …,  -100,  -100,  -100],
        [29688,   837,  6478,  …,  -100,  -100,  -100],
        [ 3927,   837,  3387,  …,  -100,  -100,  -100],
        …,
        [ 5832,   760,   262,  …,  -100,  -100,  -100],
        [ 1169,  4171,   764,  …,  -100,  -100,  -100],
        [10919,   318,   326,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   826,   286,  …,    72,    72,    72],
        [  318,  1312,   716,  …,    72,    72,    72],
        [  318,  1312,   318,  …,    72,    72,    72],
        …,
        [  318,  2802,   318,  …,    72,    72,    72],
        [  318,   644,    72,  …, 10919, 10919, 10919],
        [  318,   407,   760,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 5562,   318,   530,  …,  -100,  -100,  -100],
        [ 1219,   837,  1312,  …,  -100,  -100,  -100],
        [ 2197,   764,   644,  …,  -100,  -100,  -100],
        …,
        [    0,   534,  5156,  …,  -100,  -100,  -100],
        [49459,  5633,  -100,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  764,  …,  258,  258,  258],
        [ 318,  345, 5633,  …,   72,   72,   72],
        [ 318, 1683,  683,  …,   72,   72,   72],
        …,
        [ 318,  389,  407,  …,   72,   72,   72],
        [ 318,  307,  764,  …,   72,   72,   72],
        [ 318, 1310,  286,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  258,   318,  2636,  …,  -100,  -100,  -100],
        [ 4919,   389, 21349,  …,  -100,  -100,  -100],
        [20839,   345,   766,  …,  -100,  -100,  -100],
        …,
        [  392,   345,   389,  …,  -100,  -100,  -100],
        [ 5832,  1276,  2050,  …,  -100,  -100,  -100],
        [ 6057,   257,  3155,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345, 1612,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  764,  389,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  318,  716,  …,   72,   72,   72],
        [ 318,  407,  318,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10919,   466,   345,  …,  -100,  -100,  -100],
        [17846,   290,  3677,  …,  -100,  -100,  -100],
        [   72,   760,   345,  …,  -100,  -100,  -100],
        …,
        [   78,   764,   479,  …,  -100,  -100,  -100],
        [ 1169,  5968,  1312,  …,  -100,  -100,  -100],
        [   72,   716,   428,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  262, 1037,  …,   72,   72,   72],
        [ 318, 5490,  502,  …,   72,   72,   72],
        …,
        [ 318,  764,  345,  …,   72,   72,   72],
        [ 318,  407,  257,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  439,   826,   764,  …,  -100,  -100,  -100],
        [40716,   329,   262,  …,  -100,  -100,  -100],
        [ 4598,   407,  3638,  …,  -100,  -100,  -100],
        …,
        [11274,  1110,   284,  …,  -100,  -100,  -100],
        [  270,   318,   407,  …,  -100,  -100,  -100],
        [43669,   837,   262,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  407,  257,  …,   72,   72,   72],
        [ 318,  345, 1612,  …,   72,   72,   72],
        …,
        [ 318,  716, 1312,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  826, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 6874,   837,  3737,  …,  -100,  -100,  -100],
        [  270,   318,   587,  …,  -100,  -100,  -100],
        [10919,   466,   345,  …,  -100,  -100,  -100],
        …,
        [ 3919,  1312,   531,  …,  -100,  -100,  -100],
        [ 5832,   466,   407,  …,  -100,  -100,  -100],
        [ 5562,   318,   644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  835,  …,   72,   72,   72],
        [ 318,  389,  423,  …,   72,   72,   72],
        [ 318,  766,  345,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1392,  587,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  760,  588,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8117,   318,   645,  …,  -100,  -100,  -100],
        [   13,   345,   991,  …,  -100,  -100,  -100],
        [11274,   284,   766,  …,  -100,  -100,  -100],
        …,
        [   72,   423,  1464,  …,  -100,  -100,  -100],
        [ 5832,   760,   644,  …,  -100,  -100,  -100],
        [ 4598,   345,  1254,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5490,  523,  …,   72,   72,   72],
        [ 318, 1392,  502,  …,   72,   72,   72],
        [ 318,  345, 1016,  …,   72,   72,   72],
        …,
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318,  262, 5633,  …,   72,   72,   72],
        [ 318,  588,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4598,   407,   892,  …,  -100,  -100,  -100],
        [ 5832,   655,  1607,  …,  -100,  -100,  -100],
        [ 3003,   389,   356,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,   407,  …,  -100,  -100,  -100],
        [ 3003,   318,   262,  …,  -100,  -100,  -100],
        [11031,   286,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   716,  1312,  …,    72,    72,    72],
        [  318,   826,   764,  …,    72,    72,    72],
        [  318,  5490,   284,  …,    72,    72,    72],
        …,
        [  318,   407,   760,  …,    72,    72,    72],
        [  318,  1654,  5633,  …,    72,    72,    72],
        [  318,   262,  5633,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[   11,  1312,  2644,  …,  -100,  -100,  -100],
        [ 5562,   318,   477,  …,  -100,  -100,  -100],
        [ 4598,   407,  1949,  …,  -100,  -100,  -100],
        …,
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [  533,   345,  8805,  …,  -100,  -100,  -100],
        [10919,   546,   465,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   764,  …,    72,    72,    72],
        [  318,   826,  1312,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,   764,  …, 10919, 10919, 10919],
        [  318,   345,   466,  …,    72,    72,    72],
        [  318,   318,   645,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 8505,   837,  3763,  …,  -100,  -100,  -100],
        [ 5562,   318,   644,  …,  -100,  -100,  -100],
        [  392,  6949,   837,  …,  -100,  -100,  -100],
        …,
        [24677,  4949, 20957,  …,  -100,  -100,  -100],
        [22850,   750,   345,  …,  -100,  -100,  -100],
        [  568,   612,   547,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …,    72,    72,    72],
        [  318,    76,   764,  …,    72,    72,    72],
        [  318,  1654,  5633,  …,    72,    72,    72],
        …,
        [  318,   345,  1612,  …, 10919, 10919, 10919],
        [  318,   284,   760,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[11274,   922,   764,  …,  -100,  -100,  -100],
        [   67,   405, 17204,  …,  -100,  -100,  -100],
        [  533,   345, 14720,  …,  -100,  -100,  -100],
        …,
        [10919,   466,   345,  …,  -100,  -100,  -100],
        [   72,   765,   284,  …,  -100,  -100,  -100],
        [12758,  3763,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  644,   72,  …,   72,   72,   72],
        [ 318, 1312,  407,  …,   72,   72,   72],
        [ 318,  286,  530,  …,   72,   72,   72],
        …,
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318, 1392,  284,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  568,  5633,  -100,  …,  -100,  -100,  -100],
        [ 9930,  2644,   389,  …,  -100,  -100,  -100],
        [ 1662,   530,   764,  …,  -100,  -100,  -100],
        …,
        [39532,  6254,  6180,  …,  -100,  -100,  -100],
        [ 5832,   423,  1223,  …,  -100,  -100,  -100],
        [ 1169, 13430,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  407, 1312,  …,   72,   72,   72],
        [ 318,  530,  284,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  261,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  345,  389,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  258,   466,   407,  …,  -100,  -100,  -100],
        [   72,   714, 21192,  …,  -100,  -100,  -100],
        [   64,   922,  1295,  …,  -100,  -100,  -100],
        …,
        [ 8505,   837,  2318,  …,  -100,  -100,  -100],
        [ 4868,   268,   837,  …,  -100,  -100,  -100],
        [13893,   611,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   621,   262,  …,    72,    72,    72],
        [  318,   407,   345,  …, 10919, 10919, 10919],
        [  318,   407,   530,  …,    72,    72,    72],
        …,
        [  318,   318,   345,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  72,  517,  422,  …, -100, -100, -100],
        [  72,  716, 5149,  …, -100, -100, -100],
        [  72,  716,  262,  …, -100, -100, -100],
        …,
        [   0,  703,  714,  …, -100, -100, -100],
        [5832,  389, 9670,  …, -100, -100, -100],
        [1662, 1865,  764,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764, 24926,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        …,
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        [  318,   428,  5633,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[37784,  8036,   837,  …,  -100,  -100,  -100],
        [29688,   764,  -100,  …,  -100,  -100,  -100],
        [ 2290,   948,   764,  …,  -100,  -100,  -100],
        …,
        [ 5832,   547,   826,  …,  -100,  -100,  -100],
        [ 5832,   466,   407,  …,  -100,  -100,  -100],
        [ 8727,   318,   428,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,   764,  …,    72,    72,    72],
        [  318,   764,   764,  …, 10919, 10919, 10919],
        [  318,   588,   284,  …,    72,    72,    72],
        …,
```

```
        [  318,   966,   345,  …,    72,    72,    72],
        [  318,   510,   764,  …,    72,    72,    72],
        [  318,  1683,   587,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 3137,  3505,   262,  …,  -100,  -100,  -100],
        [   64,  6388, 10441,  …,  -100,  -100,  -100],
        [19188,   345,   588,  …,  -100,  -100,  -100],
        …,
        [  265,   644,  5633,  …,  -100,  -100,  -100],
        [   72,  1663,   340,  …,  -100,  -100,  -100],
        [14150,   345,  1464,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,  1312,  …,    72,    72,    72],
        [  318,   407,   345,  …,    72,    72,    72],
        [  318,  5633, 22220,  …,    72,    72,    72],
        …,
        [  318,  5633,   262,  …,    72,    72,    72],
        [  318,  1312,   318,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[4053, 2644, 2644,  …, -100, -100, -100],
        [  72,  716, 9675,  …, -100, -100, -100],
        [9776,  340,  848,  …, -100, -100, -100],
        …,
        [2197,  644,  318,  …, -100, -100, -100],
        [3919,  837,  340,  …, -100, -100, -100],
        [  11, 3499,  764,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  467,  307,  …,   72,   72,   72],
        [ 318,  588,  764,  …,   72,   72,   72],
        …,
        [ 318,  407,  329,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[11246, 13257,  3437,  …,  -100,  -100,  -100],
        [ 1014,   284,   407,  …,  -100,  -100,  -100],
        [ 5832,   804,   922,  …,  -100,  -100,  -100],
        …,
        [   72,   373,  2045,  …,  -100,  -100,  -100],
        [ 1640,   502,  2644,  …,  -100,  -100,  -100],
        [   72,   766,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  716,  407,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        …,
```

```
        [ 318,  284,  307,  …,   72,   72,   72],
        [ 318, 5633,  644,  …,   72,   72,   72],
        [ 318,  318,  407,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   30,  1312,   466,  …,  -100,  -100,  -100],
        [43669,   837,   345,  …,  -100,  -100,  -100],
        [ 4868,   268,   837,  …,  -100,  -100,  -100],
        …,
        [ 5832,  1016,   284,  …,  -100,  -100,  -100],
        [10755,   644,  5633,  …,  -100,  -100,  -100],
        [  361,   340,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407, 1312,  …,   72,   72,   72],
        [ 318,  407, 1468,  …,   72,   72,   72],
        [ 318,  257,  922,  …,   72,   72,   72],
        …,
        [ 318,  286,  764,  …,   72,   72,   72],
        [ 318,  345,  318,  …,   72,   72,   72],
        [ 318,  546, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[19532,   750,  5145,  …,  -100,  -100,  -100],
        [   72,   716,   281,  …,  -100,  -100,  -100],
        [ 7091,   318,   257,  …,  -100,  -100,  -100],
        …,
        [   64,  5166,   286,  …,  -100,  -100,  -100],
        [   72,  3785,   326,  …,  -100,  -100,  -100],
        [   72,  1337,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318, 1781,  764,  …,   72,   72,   72],
        [ 318,  284, 1256,  …,   72,   72,   72],
        …,
        [ 318,  644,   72,  …,   72,   72,   72],
        [ 318,  345,   72,  …,   72,   72,   72],
        [ 318, 1310,  764,  …,  986,  986,  986]], device='cuda:0')
First 10 labels: tensor([[ 3506,  2644,   290,  …,  -100,  -100,  -100],
        [ 4053,   286,  1781,  …,  -100,  -100,  -100],
        [15332,   550,   257,  …,  -100,  -100,  -100],
        …,
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        [10919,  5145,  -100,  …,  -100,  -100,  -100],
        [  986,   257,  5827,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407, 1016,  …,   72,   72,   72],
        [ 318,  407, 5633,  …, 5832, 5832, 5832],
        [ 318, 5633, 5633,  …,   72,   72,   72],
        …,
```

```
        [ 318,  644,   72,  …,   72,   72,   72],
        [ 318,  345,  318,  …, 5832, 5832, 5832],
        [ 318,  644,  318,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   389,   407,  …,  -100,  -100,  -100],
        [ 5832,   389,  1654,  …,  -100,  -100,  -100],
        [42949,  1997,  2041,  …,  -100,  -100,  -100],
        …,
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        [   72,   892, 35822,  …,  -100,  -100,  -100],
        [  568,  2644,   644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,   257,  …,    72,    72,    72],
        [  318,  1279,    72,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72],
        …,
        [  318,   345,  1804,  …,    72,    72,    72],
        [  318,  1497,   262,  …,    72,    72,    72],
        [  318,   318,   326,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[33320, 15489,   318,  …,  -100,  -100,  -100],
        [16390,   764,  -100,  …,  -100,  -100,  -100],
        [ 5832,   389, 17275,  …,  -100,  -100,  -100],
        …,
        [10919,   389,   345,  …,  -100,  -100,  -100],
        [ 4919,  1290,   284,  …,  -100,  -100,  -100],
        [  392,   644,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312, 1865,  …,   72,   72,   72],
        [ 318,  284, 5633,  …,   72,   72,   72],
        [ 318,  257,  284,  …,   72,   72,   72],
        …,
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  257,  257,  …,   72,   72,   72],
        [ 318,  523, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3919,  5145,   407,  …,  -100,  -100,  -100],
        [ 5832,   765,   616,  …,  -100,  -100,  -100],
        [ 8117,   318,  2147,  …,  -100,  -100,  -100],
        …,
        [15344,  1088,   764,  …,  -100,  -100,  -100],
        [ 5661,   318,   407,  …,  -100,  -100,  -100],
        [   72, 11691,  2644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  716,  307,  …,   72,   72,   72],
        [ 318, 1683, 1775,  …,   72,   72,   72],
        [ 318,  345,  466,  …,   72,   72,   72],
        …,
```

```
        [ 318,  345,  644,  …,   72,   72,   72],
        [ 318,  503,  994,  …,   72,   72,   72],
        [ 318,  640,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  568,  1312,   481,  …,  -100,  -100,  -100],
        [14150,   345,  1683,  …,  -100,  -100,  -100],
        [22850,   750,   345,  …,  -100,  -100,  -100],
        …,
        [   11,   523,  2644,  …,  -100,  -100,  -100],
        [ 8524,   651,   625,  …,  -100,  -100,  -100],
        [ 1659,   674,   898,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  284, 5633,  …,   72,   72,   72],
        [ 318,  262, 5633,  …,   72,   72,   72],
        …,
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  764, 1755,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   75,   764,   257,  …,  -100,  -100,  -100],
        [46012,   533,   644,  …,  -100,  -100,  -100],
        [ 3003,   318,  2156,  …,  -100,  -100,  -100],
        …,
        [ 1640,   502,  5633,  …,  -100,  -100,  -100],
        [ 8940, 12270,   837,  …,  -100,  -100,  -100],
        [13345,   340,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407, 3621,  …,   72,   72,   72],
        [ 318, 1312, 5832,  …,   72,   72,   72],
        [ 318,  651,  345,  …,   72,   72,   72],
        …,
        [ 318,  510, 1312,  …,   72,   72,   72],
        [ 318,  318,  345,  …,   72,   72,   72],
        [ 318, 1312, 1107,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  270,   318,  1107,  …,  -100,  -100,  -100],
        [ 3919,   764,  -100,  …,  -100,  -100,  -100],
        [ 5171,  1312,  1037,  …,  -100,  -100,  -100],
        …,
        [27903, 23290,   837,  …,  -100,  -100,  -100],
        [  568,   703,   750,  …,  -100,  -100,  -100],
        [ 3919,   837,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5968,   318,  …,    72,    72,    72],
        [  318,   407,   257,  …,  5832,  5832,  5832],
        [  318,   502,   345,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,  1115,   764,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   407,  1016,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   262,  5968,  …,  -100,  -100,  -100],
        [  270,   318,   407,  …,  -100,  -100,  -100],
        [ 5832,  1297,   607,  …,  -100,  -100,  -100],
        …,
        [11545,   393,  1115,  …,  -100,  -100,  -100],
        [  634,  6555,   837,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  345,  389,  …,   72,   72,   72],
        [ 318,  286,   72,  …, 5832, 5832, 5832],
        …,
        [ 318,  826,  764,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3506,  5633,  -100,  …,  -100,  -100,  -100],
        [17172,   837,   345,  …,  -100,  -100,  -100],
        [ 1136,   503,  -100,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,  1969,  …,  -100,  -100,  -100],
        [  270,   837,   340,  …,  -100,  -100,  -100],
        [14774,  2126,  2644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633, 1312,  …,   72,   72,   72],
        [ 318,  764,  293,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  826,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5562,  2089,  5633,  …,  -100,  -100,  -100],
        [   11, 10194,   763,  …,  -100,  -100,  -100],
        [43365,   462,   306,  …,  -100,  -100,  -100],
        …,
        [ 5460,   837,   764,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [ 5562,   318,  2081,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  764,  …,   72,   72,   72],
        [ 318, 1312, 2089,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  262, 5633,  …,   72,   72,   72],
        [ 318,  284, 5633,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  270,   318, 43340,  …,  -100,  -100,  -100],
        [43669,   764,  1165,  …,  -100,  -100,  -100],
        [15481,  3430,   262,  …,  -100,  -100,  -100],
        …,
        [10919,   546,   644,  …,  -100,  -100,  -100],
        [ 5832,   588,   340,  …,  -100,  -100,  -100],
        [  600,  1072,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  257,  …,   72,   72,   72],
        [ 318, 6429,  764,  …,   72,   72,   72],
        [ 318,  340, 1312,  …,   72,   72,   72],
        …,
        [ 318,  257, 2988,  …, 5832, 5832, 5832],
        [ 318, 1524, 2839,  …,   72,   72,   72],
        [ 318, 1312, 3919,  …,   11,   11,   11]], device='cuda:0')
First 10 labels: tensor([[7091,  318,  407,  …, -100, -100, -100],
        [ 325, 1151, 6429,  …, -100, -100, -100],
        [  13, 5022,  764,  …, -100, -100, -100],
        …,
        [ 258,  318,  534,  …, -100, -100, -100],
        [  64, 2839,  837,  …, -100, -100, -100],
        [1092, 5633, -100,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633, 1312,  …,   72,   72,   72],
        [ 318,  257, 2063,  …,   72,   72,   72],
        [ 318,  389,  477,  …, 5832, 5832, 5832],
        …,
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  798,  764,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[43439,   508,  5633,  …,  -100,  -100,  -100],
        [11545,   290,   257,  …,  -100,  -100,  -100],
        [  361,   345,  3377,  …,  -100,  -100,  -100],
        …,
        [42994,   407,   764,  …,  -100,  -100,  -100],
        [   82, 17403,   798,  …,  -100,  -100,  -100],
        [   72,   716,   655,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   716,  …,    72,    72,    72],
        [  318,  1521, 13893,  …,    72,    72,    72],
        [  318,  1975,   503,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,   389,  …,    72,    72,    72],
        [  318,  4459,   764,  …,    72,    72,    72],
        [  318,   644,    72,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 3919,   837,  1312,  …,  -100,  -100,  -100],
        [22850,  5633,  -100,  …,  -100,  -100,  -100],
        [   72,  2314,   651,  …,  -100,  -100,  -100],
        …,
        [   72,   760,   484,  …,  -100,  -100,  -100],
        [  259,   616,  3996,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1392,  4203,  …,    72,    72,    72],
        [  318,   345,  5633,  …,    72,    72,    72],
        [  318,   286,   262,  …,    72,    72,    72],
        …,
        [  318,   257,   284,  …, 10919, 10919, 10919],
        [  318,   837,   764,  …,    72,    72,    72],
        [  318,   307,   257,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  72,  423,  428,  …, -100, -100, -100],
        [8727,  389,  345,  …, -100, -100, -100],
        [1169, 1611,  351,  …, -100, -100, -100],
        …,
        [ 258,  373, 6405,  …, -100, -100, -100],
        [8310,  962,  345,  …, -100, -100, -100],
        [5661, 1276,  307,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,  1312,  …,    72,    72,    72],
        [  318,  5633,  5633,  …, 10919, 10919, 10919],
        [  318,   837,  1312,  …,    72,    72,    72],
        …,
        [  318,  5968,   345,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   765,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10823,    88,   764,  …,  -100,  -100,  -100],
        [ 5661,   534,  3650,  …,  -100,  -100,  -100],
        [   74,  5580,  5633,  …,  -100,  -100,  -100],
        …,
        [10919,   262,   466,  …,  -100,  -100,  -100],
        [   72,   760,  2644,  …,  -100,  -100,  -100],
        [   72,   655,  4601,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  345,  …,   72,   72,   72],
        [ 318,  837, 1694,  …,   11,   11,   11],
        [ 318,  644,   72,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764,  389,  …,   72,   72,   72],
        [ 318,  477,  837,  …,   72,   72,   72],
        [ 318,  345,  539,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 7091,   655,   588,  …,  -100,  -100,  -100],
        [20342,   837,   285,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [14337,   764,   345,  …,  -100,  -100,  -100],
        [11085,   286,   477,  …,  -100,  -100,  -100],
        [  265,  1551,   474,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …,    72,    72,    72],
        [  318,   837,   880,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        …,
        [  318,   318,   340,  …, 10919, 10919, 10919],
        [  318,   278,   764,  …,    72,    72,    72],
        [  318,   764,  1279,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[48937,  5381,   764,  …,  -100,  -100,  -100],
        [ 5832,   760, 12270,  …,  -100,  -100,  -100],
        [ 5562,   835,   764,  …,  -100,  -100,  -100],
        …,
        [10919,   640,   318,  …,  -100,  -100,  -100],
        [  260, 37713,  2802,  …,  -100,  -100,  -100],
        [   88,   538,  5633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  307,  764,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  284,  467,  …,   72,   72,   72],
        …,
        [ 318,  345,   72,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72],
        [ 318, 1312,  644,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   284,    75,  …,  -100,  -100,  -100],
        [ 2290,   365,   837,  …,  -100,  -100,  -100],
        [   72,   550,   284,  …,  -100,  -100,  -100],
        …,
        [ 2197,  5145,  -100,  …,  -100,  -100,  -100],
        [  270,   318,  6626,  …,  -100,  -100,  -100],
        [43669,  5633,   523,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,   72,  …,   72,   72,   72],
        [ 318, 1312, 1865,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312,  355,  …,   72,   72,   72],
        [ 318,  764,  546,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 7081,  5145,  -100,  …,  -100,  -100,  -100],
        [ 3919,   764,   407,  …,  -100,  -100,  -100],
        [   73, 13687,   582,  …,  -100,  -100,  -100],
        …,
        [   77,  3008,   764,  …,  -100,  -100,  -100],
        [21037,   764,   976,  …,  -100,  -100,  -100],
        [11246,   547, 34417,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5633,  1312,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,  1337,   284,  …,    72,    72,    72],
        …,
        [  318,   318,   326,  …, 10919, 10919, 10919],
        [  318,   407,   587,  …,    72,    72,    72],
        [  318,   262,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[22850,   407,  5633,  …,  -100,  -100,  -100],
        [ 1820,  5770,   837,  …,  -100,  -100,  -100],
        [ 5832,  1011,   428,  …,  -100,  -100,  -100],
        …,
        [  392,   644,   318,  …,  -100,  -100,  -100],
        [  270,   318,  1541,  …,  -100,  -100,  -100],
        [10919,   546,  5633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   318,  …,    72,    72,    72],
        [  318,  1279,   318,  …, 10919, 10919, 10919],
        [  318,   262,   262,  …,    72,    72,    72],
        …,
        [  318,   345,  5633,  …, 10919, 10919, 10919],
        [  318,   318,   407,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[43669,   837,   340,  …,  -100,  -100,  -100],
        [34680,  5145,   644,  …,  -100,  -100,  -100],
        [ 4703,  1780,   379,  …,  -100,  -100,  -100],
        …,
        [ 4758,   389,   883,  …,  -100,  -100,  -100],
        [ 4360,   340,   318,  …,  -100,  -100,  -100],
        [ 5460,  1088,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  340,  764,  …,   72,   72,   72],
        [ 318, 5490,  284,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  257,  621,  …,   72,   72,   72],
        [ 318,  284,  351,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   466,   407,  …,  -100,  -100,  -100],
        [ 1640,  1136,   340,  …,  -100,  -100,  -100],
        [ 4598,   407,   765,  …,  -100,  -100,  -100],
        …,
        [43669,   764,  -100,  …,  -100,  -100,  -100],
        [  270,   373,   517,  …,  -100,  -100,  -100],
        [  258,  2925,   503,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407, 3734,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  389,  407,  …,   72,   72,   72],
        …,
        [ 318,  826,  691,  …,   72,   72,   72],
        [ 318,  345, 1312,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   481,   307,  …,  -100,  -100,  -100],
        [ 5832,   389, 30285,  …,  -100,  -100,  -100],
        [  986,   345,   389,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,   262,  …,  -100,  -100,  -100],
        [ 5832,   531,   837,  …,  -100,  -100,  -100],
        [   72,  1612,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   612,  …,    72,    72,    72],
        [  318,   407,   284,  …,    72,    72,    72],
        [  318,   345,   373,  …,    72,    72,    72],
        …,
        [  318,   345,   764,  …,    72,    72,    72],
        [  318,   345,   651,  …,    72,    72,    72],
        [  318,  1312, 10919,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   481,   307,  …,  -100,  -100,  -100],
        [   72,   373,  2111,  …,  -100,  -100,  -100],
        [   72,  2993,   340,  …,  -100,  -100,  -100],
        …,
        [27250,   764,  5474,  …,  -100,  -100,  -100],
        [   11,   460,  1312,  …,  -100,  -100,  -100],
        [ 1452,   764,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   389,   284,  …,    72,    72,    72],
        [  318,  7319,   764,  …,    72,    72,    72],
        [  318,   326,   428,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,  1521, 13893,  …,    72,    72,    72],
        [  318,   764,   645,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[13893,   345,  1816,  …,  -100,  -100,  -100],
        [11545,  3470,  7319,  …,  -100,  -100,  -100],
        [10919,   318,   477,  …,  -100,  -100,  -100],
        …,
        [   69,  8589,   837,  …,  -100,  -100,  -100],
        [22850,  5633,  -100,  …,  -100,  -100,  -100],
        [ 3919,   530,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  257,  …,   72,   72,   72],
        [ 318,  407,  340,  …,   72,   72,   72],
        [ 318,  764,  262,  …,   72,   72,   72],
        …,
        [ 318, 1392,  467,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  318,  257,  …, 5832, 5832, 5832]], device='cuda:0')
First 10 labels: tensor([[  270,   318,   407,  …,  -100,  -100,  -100],
        [   72,   481,   466,  …,  -100,  -100,  -100],
        [25833,   655,   416,  …,  -100,  -100,  -100],
        …,
        [  732,   423,   284,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [  568,   339,   373,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   286,   389,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,   764,   764,  …,  5832,  5832,  5832],
        …,
        [  318,   318,   345,  …,    72,    72,    72],
        [  318,  1312,  1312,  …,    72,    72,    72],
        [  318,   640, 44659,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  568,   617,   661,  …,  -100,  -100,  -100],
        [ 3919,   837,   645,  …,  -100,  -100,  -100],
        [ 5832,  8531, 31030,  …,  -100,  -100,  -100],
        …,
        [ 1169,  5968,   351,  …,  -100,  -100,  -100],
        [ 4053,   837,  3360,  …,  -100,  -100,  -100],
        [ 1662,   428,   665,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1521, 13893,  …,    72,    72,    72],
        [  318,   837,   345,  …,    72,    72,    72],
        [  318,   837,   268,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,   640,  …,    72,    72,    72],
        [  318,   837,   345,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[22850,  5633,  -100,  …,  -100,  -100,  -100],
        [ 2958,   319,   837,  …,  -100,  -100,  -100],
        [   83, 26730, 12018,  …,  -100,  -100,  -100],
        …,
        [  492,   262,  1306,  …,  -100,  -100,  -100],
        [20342,   837,   466,  …,  -100,  -100,  -100],
        [   72,   760,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  307,  257,  …,   72,   72,   72],
        [ 318,  345,  760,  …,   72,   72,   72],
        …,
        [ 318,  837,  764,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  263,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4053,   837,   644,  …,  -100,  -100,  -100],
        [   72,  1244,   787,  …,  -100,  -100,  -100],
        [ 4919,   466,   345,  …,  -100,  -100,  -100],
        …,
        [20342,   837, 17266,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [ 1324,   677,   415,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …,    72,    72,    72],
        [  318,  1683,   284,  …, 10919, 10919, 10919],
        [  318,  1392,   587,  …,    72,    72,    72],
        …,
        [  318,   407,  7926,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,   345,  1804,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   760,  2644,  …,  -100,  -100,  -100],
        [14150,   345,   587,  …,  -100,  -100,  -100],
        [   72,   423,   691,  …,  -100,  -100,  -100],
        …,
        [   72,   716,   523,  …,  -100,  -100,  -100],
        [   82,   343,   764,  …,  -100,  -100,  -100],
        [10919,   389,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  837,  …,   72,   72,   72],
        [ 318,  588, 1088,  …,   72,   72,   72],
        [ 318,  466,  345,  …,   72,   72,   72],
        …,
```

```
        [ 318,  644, 5633,  …,   72,   72,   72],
        [ 318,  531,  764,  …,   72,   72,   72],
        [ 318,  866,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  622,    69,   385,  …,  -100,  -100,  -100],
        [   72,   561,  1210,  …,  -100,  -100,  -100],
        [ 4919,   881,   423,  …,  -100,  -100,  -100],
        …,
        [  271,   326, 12431,  …,  -100,  -100,  -100],
        [ 7091,  1239,  1364,  …,  -100,  -100,  -100],
        [ 9948,    76,   866,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,   262,  …, 10919, 10919, 10919],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   837,   345,  …,    72,    72,    72],
        …,
        [  318,   644,    72,  …,    72,    72,    72],
        [  318,   407,  2300,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 1169, 34561,   837,  …,  -100,  -100,  -100],
        [28177,   612,   764,  …,  -100,  -100,  -100],
        [   65,   672,  2644,  …,  -100,  -100,  -100],
        …,
        [27485,  5633,  -100,  …,  -100,  -100,  -100],
        [19425,   857,   407,  …,  -100,  -100,  -100],
        [ 3919,  5176,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[318, 651, 345,  …,  72,  72,  72],
        [318, 407, 764,  …,  72,  72,  72],
        [318, 407, 764,  …,  72,  72,  72],
        …,
        [318, 764, 764,  …,  72,  72,  72],
        [318, 345, 262,  …,  72,  72,  72],
        [318, 835, 764,  …,  72,  72,  72]], device='cuda:0')
First 10 labels: tensor([[ 1616,   502,   651,  …,  -100,  -100,  -100],
        [25591,   389,   616,  …,  -100,  -100,  -100],
        [   72,   373,   407,  …,  -100,  -100,  -100],
        …,
        [ 5832,  2314, 11238,  …,  -100,  -100,  -100],
        [  568,   837,   319,  …,  -100,  -100,  -100],
        [  259,   257, 16853,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1279,  …, 10919, 10919, 10919],
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   345,   815,  …,    72,    72,    72],
        …,
```

```
        [  318,   466,   345,  …,    72,    72,    72],
        [  318,   764,   764,  …,    72,    72,    72],
        [  318,   407,   257,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[1462, 5287, 5145,  …, -100, -100, -100],
        [  72,  716,  922,  …, -100, -100, -100],
        [8524, 3737,  345,  …, -100, -100, -100],
        …,
        [8524, 1521,  389,  …, -100, -100, -100],
        [  72, 1833,  345,  …, -100, -100, -100],
        [ 270,  318,  407,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   716,   407,  …,    72,    72,    72],
        [  318,  5664,   837,  …, 10919, 10919, 10919],
        [  318,   764,  1312,  …,    72,    72,    72],
        …,
        [  318,   345,   345,  …,    72,    72,    72],
        [  318,   389,   407,  …,    72,    72,    72],
        [  318,  1312,   286,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   13,  1312,   481,  …,  -100,  -100,  -100],
        [17077,   257,  5664,  …,  -100,  -100,  -100],
        [   11,  3734,   837,  …,  -100,  -100,  -100],
        …,
        [ 8727,   389,   407,  …,  -100,  -100,  -100],
        [ 5562,   484,   389,  …,  -100,  -100,  -100],
        [43669,   764,   617,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,  1312,  …,    72,    72,    72],
        [  318,   326,  2300,  …, 10919, 10919, 10919],
        [  318,   257,   257,  …,    72,    72,    72],
        …,
        [  318,   407,   530,  …,    72,    72,    72],
        [  318,   389,   765,  …,    72,    72,    72],
        [  318,   407,   764,  …,  5832,  5832,  5832]], device='cuda:0')
First 10 labels: tensor([[  305,  1362,   764,  …,  -100,  -100,  -100],
        [10919,   318,   262,  …,  -100,  -100,  -100],
        [  258,   318,   655,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,   262,  …,  -100,  -100,  -100],
        [  361,   345,  1107,  …,  -100,  -100,  -100],
        [   72,   716,  7014,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  262, 5633,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  764, 1545,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1654, 5633,  …,   72,   72,   72],
        [ 318,  584,  764,  …,   72,   72,   72],
        [ 318,  530,  262,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3003,   318,   673,  …,  -100,  -100,  -100],
        [17945,   502,   764,  …,  -100,  -100,  -100],
        [ 8071,  1240,   616,  …,  -100,  -100,  -100],
        …,
        [  533,   345,  7165,  …,  -100,  -100,  -100],
        [  273,   262,  2877,  …,  -100,  -100,  -100],
        [  986,   262,   286,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  764,  …, 5832, 5832, 5832],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  466,  466,  …,   72,   72,   72],
        …,
        [ 318, 5633,  644,  …,   72,   72,   72],
        [ 318,  502,  764,  …,   72,   72,   72],
        [ 318, 1392,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1820, 37516,   837,  …,  -100,  -100,  -100],
        [   69, 19968,   764,  …,  -100,  -100,  -100],
        [  732,  2314,  1107,  …,  -100,  -100,  -100],
        …,
        [ 5832,   644,  5633,  …,  -100,  -100,  -100],
        [ 5832,  1560,   502,  …,  -100,  -100,  -100],
        [ 5832,   423,   607,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  644,   72,  …,   72,   72,   72],
        [ 318,  345,  307,  …,   72,   72,   72],
        [ 318, 1312,  837,  …,   72,   72,   72],
        …,
        [ 318,  345,  373,  …,   72,   72,   72],
        [ 318,  760,  760,  …,   72,   72,   72],
        [ 318,  764,  339,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10919,  5633,  -100,  …,  -100,  -100,  -100],
        [ 7081,   837,   284,  …,  -100,  -100,  -100],
        [ 1219,   837,  4043,  …,  -100,  -100,  -100],
        …,
        [   72,  1807,  1312,  …,  -100,  -100,  -100],
        [ 4598,   345,   407,  …,  -100,  -100,  -100],
        [12728,   683,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   318,   407,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   837,  1312,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,   340,  3656,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,  5633,  5633,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  11,  340,  318,  …, -100, -100, -100],
        [3506,  994,  764,  …, -100, -100, -100],
        [3137,  783, 2644,  …, -100, -100, -100],
        …,
        [4919,  318,  534,  …, -100, -100, -100],
        [  11, 3956,  764,  …, -100, -100, -100],
        [1525,  508, 5633,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   764,  …,    72,    72,    72],
        [  318,   307,   764,  …,    72,    72,    72],
        [  318,   644,  1016,  …,    72,    72,    72],
        …,
        [  318,  1312,   466,  …, 10919, 10919, 10919],
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,   345,   760,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 1219,   837,  5176,  …,  -100,  -100,  -100],
        [ 5146,   284,   837,  …,  -100,  -100,  -100],
        [  271,   326,   991,  …,  -100,  -100,  -100],
        …,
        [10919,   815,  1312,  …,  -100,  -100,  -100],
        [11274,  6180,   764,  …,  -100,  -100,  -100],
        [ 4919,   466,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  764,  …,   72,   72,   72],
        [ 318, 1312,  539,  …,   72,   72,   72],
        [ 318, 1312, 1107,  …,   72,   72,   72],
        …,
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318, 5490,  523,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   13,   326,  2044,  …,  -100,  -100,  -100],
        [29688,   837,   474,  …,  -100,  -100,  -100],
        [ 3919,   837,   407,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,  1239,  …,  -100,  -100,  -100],
        [ 4598,   407,   307,  …,  -100,  -100,  -100],
        [   72,   716,  7926,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   257,   320,  …,    72,    72,    72],
        [  318,   407,   326,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    11,    11,    11],
        …,
```

```
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,   286,   257,  …, 10919, 10919, 10919],
        [  318,   407,   284,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 5832,  1392,   474,  …,  -100,  -100,  -100],
        [   72,   716,  2282,  …,  -100,  -100,  -100],
        [  986, 23748,  2644,  …,  -100,  -100,  -100],
        …,
        [   72,  1612,   837,  …,  -100,  -100,  -100],
        [10919,  1611,   286,  …,  -100,  -100,  -100],
        [   72,   716,  1016,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  340,  …,   72,   72,   72],
        [ 318,  407, 1312,  …,   72,   72,   72],
        [ 318,  764,  345,  …,   72,   72,   72],
        …,
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312,  345,  …,   72,   72,   72],
        [ 318,  651,  651,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   760,   546,  …,  -100,  -100,  -100],
        [   72,   716,  7787,  …,  -100,  -100,  -100],
        [32041,  7821,   764,  …,  -100,  -100,  -100],
        …,
        [ 2093,   340,  5145,  …,  -100,  -100,  -100],
        [22366,   837,  5875,  …,  -100,  -100,  -100],
        [20839,   356,   655,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  922,  …,   72,   72,   72],
        [ 318,  644,   72,  …,   72,   72,   72],
        [ 318,  318,  407,  …, 5832, 5832, 5832],
        …,
        [ 318,  837,  764,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318, 1392, 1256,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  270,   318,   257,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        [  986,   340,   318,  …,  -100,  -100,  -100],
        …,
        [20342,   837,  6478,  …,  -100,  -100,  -100],
        [ 4053,   837,  1312,  …,  -100,  -100,  -100],
        [ 5832,   423,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318,  466,  502,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 5633,  644,  …,   72,   72,   72],
        [ 318, 5490,  837,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  72,  466,  407,  …, -100, -100, -100],
        [5832,  389,  407,  …, -100, -100, -100],
        [5832,  460, 1265,  …, -100, -100, -100],
        …,
        [8625,  415,  764,  …, -100, -100, -100],
        [  71, 7456, 5633,  …, -100, -100, -100],
        [4598,  407, 5490,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …,    72,    72,    72],
        [  318,  1312,   716,  …,    72,    72,    72],
        [  318,   582, 15584,  …,    72,    72,    72],
        …,
        [  318,   484,   373,  …,    72,    72,    72],
        [  318,  9150,   876,  …,    72,    72,    72],
        [  318,  5490,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 1659,  1781,   764,  …,  -100,  -100,  -100],
        [ 1456,   764,  1312,  …,  -100,  -100,  -100],
        [13893,   257,  7771,  …,  -100,  -100,  -100],
        …,
        [ 9930,   531,  1312,  …,  -100,  -100,  -100],
        [14108,   662,  3828,  …,  -100,  -100,  -100],
        [ 4598,   407,  1088,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  764, 8161,  …,   72,   72,   72],
        [ 318,  407,  257,  …,   72,   72,   72],
        …,
        [ 318,  345,  318,  …,   72,   72,   72],
        [ 318,  345,  651,  …,   72,   72,   72],
        [ 318,  345,   72,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[41599,   764,   356,  …,  -100,  -100,  -100],
        [   11,  3387,   307,  …,  -100,  -100,  -100],
        [  270,   318,   655,  …,  -100,  -100,  -100],
        …,
        [  986,   523,   644,  …,  -100,  -100,  -100],
        [   11,   460,   356,  …,  -100,  -100,  -100],
        [13116,  5145,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[318, 764, 764,  …,  72,  72,  72],
        [318, 307, 880,  …,  72,  72,  72],
        [318, 345, 523,  …,  72,  72,  72],
        …,
```

```
        [318, 826, 286,  …,  72,  72,  72],
        [318, 318, 345,  …,  11,  11,  11],
        [318, 345, 318,  …,  72,  72,  72]], device='cuda:0')
First 10 labels: tensor([[   64, 23867,  2151,  …,  -100,  -100,  -100],
        [   72,  1244,   355,  …,  -100,  -100,  -100],
        [10919,   787,  9838,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,   530,  …,  -100,  -100,  -100],
        [   11,   644,   389,  …,  -100,  -100,  -100],
        [  732,   892,   340,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  318,  407,  …,   72,   72,   72],
        …,
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  307,  257,  …,   72,   72,   72],
        [ 318, 5633, 2739,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[27547,   837,   764,  …,  -100,  -100,  -100],
        [ 4053,   837,   645,  …,  -100,  -100,  -100],
        [  392,   340,   318,  …,  -100,  -100,  -100],
        …,
        [ 4868,   268,   837,  …,  -100,  -100,  -100],
        [  270,  1276,   307,  …,  -100,  -100,  -100],
        [  321,  1312,  1165,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1806,  764,  …,   72,   72,   72],
        [ 318,  345,  764,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        …,
        [ 318,  837,  318,  …,   72,   72,   72],
        [ 318,  284, 1637,  …,   72,   72,   72],
        [ 318,  340, 5633,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  445,   607,  1806,  …,  -100,  -100,  -100],
        [   72,  2497,   262,  …,  -100,  -100,  -100],
        [43669,   837,   356,  …,  -100,  -100,  -100],
        …,
        [  392,   523,   340,  …,  -100,  -100,  -100],
        [ 5832,   765,   262,  …,  -100,  -100,  -100],
        [ 4919,   318, 33779,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,  1016,  …, 10919, 10919, 10919],
        [  318,  1312,   716,  …,    72,    72,    72],
        [  318,   345,    72,  …,    72,    72,    72],
        …,
```

```
        [  318,  1312,   716,  …,  7091,  7091,  7091],
        [  318,   257,   922,  …,    72,    72,    72],
        [  318,  5633,   866,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   389,   356,  …,  -100,  -100,  -100],
        [ 3919,   837,  1312,  …,  -100,  -100,  -100],
        [13116,   764,  -100,  …,  -100,  -100,  -100],
        …,
        [ 1219,   764,  1312,  …,  -100,  -100,  -100],
        [ 5562,   373,  2495,  …,  -100,  -100,  -100],
        [22850,   407,  1650,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5770,   837,  …, 10919, 10919, 10919],
        [  318,   407,   760,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        …,
        [  318,   837,   837,  …,    72,    72,    72],
        [  318,   764,   764,  …, 10919, 10919, 10919],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 1219,   616,  5770,  …,  -100,  -100,  -100],
        [  732,   466,   407,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        …,
        [  986,  4043,   994,  …,  -100,  -100,  -100],
        [21809,  2834,   625,  …,  -100,  -100,  -100],
        [ 1662,  2089,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837,  262,  …,   72,   72,   72],
        [ 318, 1654, 5633,  …,   72,   72,   72],
        [ 318,  262,  764,  …,   72,   72,   72],
        …,
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  588,  284,  …,   72,   72,   72],
        [ 318,  345,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   760,   546,  …,  -100,  -100,  -100],
        [  533,   345,  1690,  …,  -100,  -100,  -100],
        [ 1456,   318,   616,  …,  -100,  -100,  -100],
        …,
        [ 8505,  5145,  -100,  …,  -100,  -100,  -100],
        [19188,   345,   588,  …,  -100,  -100,  -100],
        [ 1462,   905,   326,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   345,  …,    72,    72,    72],
        [  318,   345, 14710,  …,    72,    72,    72],
        [  318,   837,   764,  …,    72,    72,    72],
        …,
```

```
        [  318,  1392,  4203,  …,    72,    72,    72],
        [  318,   407,  1312,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[22850,   466,   407,  …,  -100,  -100,  -100],
        [ 5832,   892,  1341,  …,  -100,  -100,  -100],
        [ 3919,   582,   645,  …,  -100,  -100,  -100],
        …,
        [   72,   423,   257,  …,  -100,  -100,  -100],
        [  270,   318,   764,  …,  -100,  -100,  -100],
        [16783,   353,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,  1016,  …,    72,    72,    72],
        [  318,   286,  6891,  …, 10919, 10919, 10919],
        [  318,   588,   588,  …,    72,    72,    72],
        …,
        [  318,   407,  5633,  …,    72,    72,    72],
        [  318,   345,   588,  …,    72,    72,    72],
        [  318,   764,  4436,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   716,   407,  …,  -100,  -100,  -100],
        [   64,  6508,   286,  …,  -100,  -100,  -100],
        [   72,   561,   655,  …,  -100,  -100,  -100],
        …,
        [22850,   318,   326,  …,  -100,  -100,  -100],
        [ 4919,   561,   345,  …,  -100,  -100,  -100],
        [ 6738,   262, 19906,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   307,   307,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72],
        …,
        [  318,  1683,   340,  …,    72,    72,    72],
        [  318,   407,  1016,  …,    72,    72,    72],
        [  318,   345,   837,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[  258,   481,   407,  …,  -100,  -100,  -100],
        [   65,  6548,   837,  …,  -100,  -100,  -100],
        [   72,   716,  7926,  …,  -100,  -100,  -100],
        …,
        [20839,   345,   588,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100],
        [ 4360,   837,   475,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  910,  345,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  262, 5633,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764,  345,  …,   72,   72,   72],
        [ 318, 5633,  661,  …,   72,   72,   72],
        [ 318,  307,  502,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[20839,   339,  1234,  …,  -100,  -100,  -100],
        [ 1219,   837,  1312,  …,  -100,  -100,  -100],
        [10919,   546,   262,  …,  -100,  -100,  -100],
        …,
        [ 3605,  3516,  2644,  …,  -100,  -100,  -100],
        [14150,   867,  5863,  …,  -100,  -100,  -100],
        [10594,   345,  4654,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  282,  837,  …,   72,   72,   72],
        [ 318,  326, 5633,  …,   72,   72,   72],
        …,
        [ 318,  428, 5633,  …,   72,   72,   72],
        [ 318,  407, 1016,  …, 5832, 5832, 5832],
        [ 318,  318, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[41599,  5145,  -100,  …,  -100,  -100,  -100],
        [  324, 10793,   282,  …,  -100,  -100,  -100],
        [10919,   318,   326,  …,  -100,  -100,  -100],
        …,
        [ 8727,   318,   428,  …,  -100,  -100,  -100],
        [ 5832,   389,   407,  …,  -100,  -100,  -100],
        [ 5661,  5509,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        …,
        [ 318,  318,  407,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[12215,   934,   306,  …,  -100,  -100,  -100],
        [ 3506,   994,   764,  …,  -100,  -100,  -100],
        [18223,   764,  -100,  …,  -100,  -100,  -100],
        …,
        [   30,   326,   318,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[318, 588, 764,  …,  72,  72,  72],
        [318, 307, 502,  …,  72,  72,  72],
        [318, 407, 284,  …,  72,  72,  72],
        …,
```

```
        [318, 764, 257,  …,  72,  72,  72],
        [318, 502, 345,  …,  72,  72,  72],
        [318, 764, 764,  …,  72,  72,  72]], device='cuda:0')
First 10 labels: tensor([[   83,  4594,   340,  …,  -100,  -100,  -100],
        [10594,   345, 12226,  …,  -100,  -100,  -100],
        [   72,   716,  3772,  …,  -100,  -100,  -100],
        …,
        [ 6270,  1524,   373,  …,  -100,  -100,  -100],
        [ 1845,   563,   764,  …,  -100,  -100,  -100],
        [  353, 16358,  7926,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  345,  764,  …,   72,   72,   72],
        …,
        [ 318,  530,  764,  …,   72,   72,   72],
        [ 318,  837,  345,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   547,  3750,  …,  -100,  -100,  -100],
        [ 8548,  5321,   764,  …,  -100,  -100,  -100],
        [21754,   407,   307,  …,  -100,  -100,  -100],
        …,
        [ 1169,   649,  3650,  …,  -100,  -100,  -100],
        [32433,   273,   837,  …,  -100,  -100,  -100],
        [ 5832,   760,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633,  284,  …,   72,   72,   72],
        [ 318, 1654, 5633,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72],
        …,
        [ 318,  345,  764,  …,   72,   72,   72],
        [ 318,  345,  318,  …,   72,   72,   72],
        [ 318,  345, 5633,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1640,   644,  5633,  …,  -100,  -100,  -100],
        [  533,   345,  1695,  …,  -100,  -100,  -100],
        [ 4053,   764,   340,  …,  -100,  -100,  -100],
        …,
        [16706,  5875,   345,  …,  -100,  -100,  -100],
        [   72,   892,   340,  …,  -100,  -100,  -100],
        [ 3003,   389,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  764,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  760,  257,  …,   72,   72,   72],
        …,
```

```
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318, 1312,  837,  …,   72,   72,   72],
        [ 318, 5633, 2139,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8505,   837,  3763,  …,  -100,  -100,  -100],
        [  986,   475,   508,  …,  -100,  -100,  -100],
        [ 4598,   345,   423,  …,  -100,  -100,  -100],
        …,
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [ 1219,   837,   880,  …,  -100,  -100,  -100],
        [22583,  1431,  1171,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,  318,  …,   72,   72,   72],
        [ 318, 1310,  764,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        …,
        [ 318, 5490,  319,  …,   72,   72,   72],
        [ 318,  845,  286,  …,   72,   72,   72],
        [ 318,  764,  502,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4360,   611,   340,  …,  -100,  -100,  -100],
        [ 3137,   257,  2589,  …,  -100,  -100,  -100],
        [43669,   764,  -100,  …,  -100,  -100,  -100],
        …,
        [ 4598,   407,   954,  …,  -100,  -100,  -100],
        [  265,   262,  4220,  …,  -100,  -100,  -100],
        [ 1662,  1865,  1577,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345, 5633,  …,   72,   72,   72],
        [ 318, 1392, 1775,  …,   72,   72,   72],
        [ 318,  407,  284,  …,   72,   72,   72],
        …,
        [ 318,  766,  766,  …,   72,   72,   72],
        [ 318,  837,  962,  …, 5832, 5832, 5832],
        [ 318,  345, 1560,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[8727,  389,  345,  …, -100, -100, -100],
        [  72,  423, 1239,  …, -100, -100, -100],
        [ 270,  318, 3621,  …, -100, -100, -100],
        …,
        [  72,  460,  991,  …, -100, -100, -100],
        [2958,  319,  289,  …, -100, -100, -100],
        [4919,  460,  345,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  262, 1637,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312,  407,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[3919,  837,  645,  …, -100, -100, -100],
        [1456,  318,  534,  …, -100, -100, -100],
        [1462,  534, 1535,  …, -100, -100, -100],
        …,
        [  72,  466,  407,  …, -100, -100, -100],
        [  77, 3008,  764,  …, -100, -100, -100],
        [3919,  837,  466,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,    72,  …,    72,    72,    72],
        [  318,   651,   340,  …,    72,    72,    72],
        [  318,  1312,  2644,  …,    72,    72,    72],
        …,
        [  318,   407,   257,  …,    72,    72,    72],
        [  318,   423,   257,  …,    72,    72,    72],
        [  318,   389,   407,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[8505,  764, -100,  …, -100, -100, -100],
        [5171,  356,  787,  …, -100, -100, -100],
        [3919, 2644,  645,  …, -100, -100, -100],
        …,
        [  72,  716, 1972,  …, -100, -100, -100],
        [ 732,  714,  423,  …, -100, -100, -100],
        [4360,  356,  389,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1279, 1312,  …,   72,   72,   72],
        [ 318,  257,  257,  …,  258,  258,  258],
        [ 318,  837, 1312,  …,   72,   72,   72],
        …,
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  407,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[14261,   764,   837,  …,  -100,  -100,  -100],
        [  258,   318,   407,  …,  -100,  -100,  -100],
        [11274,  6180,   764,  …,  -100,  -100,  -100],
        …,
        [  282,  3506,   837,  …,  -100,  -100,  -100],
        [   73,   671,   764,  …,  -100,  -100,  -100],
        [ 5832,   389,  1016,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  837,  …,   72,   72,   72],
        [ 318,  389,  284,  …,   72,   72,   72],
        [ 318,  651,  257,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764,  318,  …, 5832, 5832, 5832],
        [ 318,  588,  466,  …,   72,   72,   72],
        [ 318,  345,  510,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[25124,   837,  2636,  …,  -100,  -100,  -100],
        [22850,   345,  1392,  …,  -100,  -100,  -100],
        [ 1616,   502,  8804,  …,  -100,  -100,  -100],
        …,
        [ 1662,  1862,   339,  …,  -100,  -100,  -100],
        [   72,   561,  1239,  …,  -100,  -100,  -100],
        [21453,  5145,  7765,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   826,  …,    72,    72,    72],
        [  318,   257,  1256,  …,    72,    72,    72],
        [  318,   326,  1804,  …, 10919, 10919, 10919],
        …,
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,   764,   345,  …,    72,    72,    72],
        [  318,   760,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  270,   318,   477,  …,  -100,  -100,  -100],
        [ 8117,   318,   257,  …,  -100,  -100,  -100],
        [10919,   318,   673,  …,  -100,  -100,  -100],
        …,
        [ 3919,   764,   645,  …,  -100,  -100,  -100],
        [14485, 15340,  5145,  …,  -100,  -100,  -100],
        [ 4598,   345,   892,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407, 1310,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  760, 5633,  …,   72,   72,   72],
        …,
        [ 318,  811,  764,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   716,   257,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [42949,   284,   467,  …,  -100,  -100,  -100],
        …,
        [  353,    81,   811,  …,  -100,  -100,  -100],
        [   72,   373, 20884,  …,  -100,  -100,  -100],
        [ 5832,   389,  1107,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,  5832,  …,    72,    72,    72],
        [  318,   373,   257,  …,    72,    72,    72],
        [  318,   502,   644,  …,    72,    72,    72],
        …,
```

```
        [  318,   651,   764,  …,    72,    72,    72],
        [  318,  1312,   716,  …,    72,    72,    72],
        [  318,   345, 20342,  …, 20342, 20342, 20342]], device='cuda:0')
First 10 labels: tensor([[ 3919,   764,  -100,  …,  -100,  -100,  -100],
        [30079,   339,   373,  …,  -100,  -100,  -100],
        [ 3137,  1560,   502,  …,  -100,  -100,  -100],
        …,
        [ 1616,   502,   467,  …,  -100,  -100,  -100],
        [   13,   837,  1312,  …,  -100,  -100,  -100],
        [20342,   764,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  345, 1804,  …,   72,   72,   72],
        [ 318,  764,  477,  …,   72,   72,   72],
        …,
        [ 318,  284,  502,  …,   72,   72,   72],
        [ 318,  837,  764,  …,   72,   72,   72],
        [ 318,  307,  307,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   731,   837,  …,  -100,  -100,  -100],
        [10919,   389,   345,  …,  -100,  -100,  -100],
        [ 5832, 23844,   606,  …,  -100,  -100,  -100],
        …,
        [ 5562,  5158,   284,  …,  -100,  -100,  -100],
        [20342,   837,   582,  …,  -100,  -100,  -100],
        [  270,   481,  1239,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   764,  …, 10919, 10919, 10919],
        [  318,   481,   345,  …,    72,    72,    72],
        [  318,   345,   389,  …,    72,    72,    72],
        …,
        [  318,  1312,    72,  …,    72,    72,    72],
        [  318,   345,   764,  …,    72,    72,    72],
        [  318,   257, 10757,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 8505,   764,  1637,  …,  -100,  -100,  -100],
        [ 8524,  1312,  4724,  …,  -100,  -100,  -100],
        [   72,  2911,   345,  …,  -100,  -100,  -100],
        …,
        [29688,   764,  -100,  …,  -100,  -100,  -100],
        [   72,  2497,   534,  …,  -100,  -100,  -100],
        [ 3876, 14992,   374,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   765,   345,  …,    72,    72,    72],
        [  318,   345,  1016,  …,  5832,  5832,  5832],
        [  318,   262,  1312,  …,    72,    72,    72],
        …,
```

```
        [  318,  1312,  5633,  …,    72,    72,    72],
        [  318,   284,  5633,  …,    72,    72,    72],
        [  318,   460,   345,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[   72,   655,  2497,  …,  -100,  -100,  -100],
        [12735,   389,   407,  …,  -100,  -100,  -100],
        [ 1662,   287,   764,  …,  -100,  -100,  -100],
        …,
        [ 8505,   764,  1521,  …,  -100,  -100,  -100],
        [ 5832,   588,   616,  …,  -100,  -100,  -100],
        [10155,  1312,  1265,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   318,   257,  …,    72,    72,    72],
        [  318,  1392,  4203,  …,    72,    72,    72],
        [  318,  2384,  5145,  …,    72,    72,    72],
        …,
        [  318,   354,   764,  …,    72,    72,    72],
        [  318,  1312,  5633,  …,    72,    72,    72],
        [  318,  5770,  1312,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[5661, 1295,  318,  …, -100, -100, -100],
        [  72,  423,  257,  …, -100, -100, -100],
        [2306, 2384, 5145,  …, -100, -100, -100],
        …,
        [  69, 1616,  354,  …, -100, -100, -100],
        [8505, 2644, 1521,  …, -100, -100, -100],
        [  30,  616, 5633,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1456,  287,  …,   72,   72,   72],
        [ 318, 1468, 9707,  …,   72,   72,   72],
        [ 318,  837,  764,  …,   72,   72,   72],
        …,
        [ 318,  284, 1312,  …,   72,   72,   72],
        [ 318,  764, 1312,  …, 7091, 7091, 7091],
        [ 318, 5490,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   82, 28030,  1456,  …,  -100,  -100,  -100],
        [34330,   281,  3489,  …,  -100,  -100,  -100],
        [  439,   826,   788,  …,  -100,  -100,  -100],
        …,
        [   72, 11148,  5145,  …,  -100,  -100,  -100],
        [ 1820,  4957,   764,  …,  -100,  -100,  -100],
        [ 4598,   407,  1561,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  716,  …,   72,   72,   72],
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  588, 5633,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  466,  881,  …,   72,   72,   72],
        [ 318,  407,  612,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,  2644,  1312,  …,  -100,  -100,  -100],
        [41599,   764,  3863,  …,  -100,  -100,  -100],
        [  392,   655,   644,  …,  -100,  -100,  -100],
        …,
        [ 3919,   837,   340,  …,  -100,  -100,  -100],
        [ 4919,   881,   703,  …,  -100,  -100,  -100],
        [  732,   547,   866,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  407, 3734,  …,   72,   72,   72],
        [ 318, 2000,  837,  …,   72,   72,   72],
        …,
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318, 1312,  319,  …,   72,   72,   72],
        [ 318,  284,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  72,  750,  407,  …, -100, -100, -100],
        [5832,  481,  307,  …, -100, -100, -100],
        [ 259,  534, 4459,  …, -100, -100, -100],
        …,
        [  72,  466,  407,  …, -100, -100, -100],
        [4053,  837, 1282,  …, -100, -100, -100],
        [5832,  765,  502,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,  1612,  …, 10919, 10919, 10919],
        [  318,   597,   612,  …,    72,    72,    72],
        [  318,  5633,    72,  …,    72,    72,    72],
        …,
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,   764,   764,  …, 10919, 10919, 10919],
        [  318,   837,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   466,   345,  …,  -100,  -100,  -100],
        [  533,   612,   503,  …,  -100,  -100,  -100],
        [22850,   407,  -100,  …,  -100,  -100,  -100],
        …,
        [ 3919,   764,  1107,  …,  -100,  -100,  -100],
        [41607, 19972,  1042,  …,  -100,  -100,  -100],
        [11043,    77, 48176,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  407, 5633,  …,   72,   72,   72],
        [ 318,  703,   72,  …,   72,   72,   72],
        …,
```

```
        [ 318, 5633, 3375,  …,   72,   72,   72],
        [ 318,  531,  345,  …,   72,   72,   72],
        [ 318,  284,  467,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4053,   837,   356,  …,  -100,  -100,  -100],
        [22850,   318,   326,  …,  -100,  -100,  -100],
        [ 4919,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [ 9930,   345,   389,  …,  -100,  -100,  -100],
        [   72,  1239,  1297,  …,  -100,  -100,  -100],
        [ 5832,   765,   284,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …,  7091,  7091,  7091],
        [  318,   837,  1312,  …,  5832,  5832,  5832],
        [  318,  5633,  1279,  …,    72,    72,    72],
        …,
        [  318,   345,   340,  …,    72,    72,    72],
        [  318,   588,  1312,  …,    72,    72,    72],
        [  318,   257,   922,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[   72,   766,   764,  …,  -100,  -100,  -100],
        [ 5832,   760,   644,  …,  -100,  -100,  -100],
        [16706,   757,  5633,  …,  -100,  -100,  -100],
        …,
        [   66, 34574, 31506,  …,  -100,  -100,  -100],
        [ 5832,   804,  2644,  …,  -100,  -100,  -100],
        [  258,   318,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1016,  6405,  …,    72,    72,    72],
        [  318,   616,   286,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        …,
        [  318,   326,  2300,  …, 10919, 10919, 10919],
        [  318,   262,   837,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  732,   389,  1972,  …,  -100,  -100,  -100],
        [   72,  2626,  1630,  …,  -100,  -100,  -100],
        [ 2188,  4058,   764,  …,  -100,  -100,  -100],
        …,
        [10919,   318,   262,  …,  -100,  -100,  -100],
        [13552,   286,  5770,  …,  -100,  -100,  -100],
        [ 8691,  2271,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   716,  …,    72,    72,    72],
        [  318,  5633,   644,  …, 10919, 10919, 10919],
        [  318,   407,   284,  …,    72,    72,    72],
        …,
```

```
        [  318,   502,   764,  …,    72,    72,    72],
        [  318,   389,  1279,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[3919,  764, 1312,  …, -100, -100, -100],
        [  64,  644, 2644,  …, -100, -100, -100],
        [  72,  481, 1561,  …, -100, -100, -100],
        …,
        [1662, 1560,  607,  …, -100, -100, -100],
        [5562,  345, 5633,  …, -100, -100, -100],
        [2188,  319,  764,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  826, 1312,  …,   72,   72,   72],
        [ 318,  407, 6568,  …,   72,   72,   72],
        [ 318,  716, 1392,  …,   72,   72,   72],
        …,
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  764,  345,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5562,   318,   644,  …,  -100,  -100,  -100],
        [   72,   373,   523,  …,  -100,  -100,  -100],
        [ 4360,  1312,   423,  …,  -100,  -100,  -100],
        …,
        [  322,   764,  -100,  …,  -100,  -100,  -100],
        [31373,   764,   561,  …,  -100,  -100,  -100],
        [   30,   645,  2644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  257, 1256,  …,   72,   72,   72],
        [ 318, 1312,  262,  …,   72,   72,   72],
        …,
        [ 318,  826,  764,  …,   72,   72,   72],
        [ 318,  345,  764,  …,   72,   72,   72],
        [ 318,  373, 2497,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   716,  7926,  …,  -100,  -100,  -100],
        [ 8117,   318,   257,  …,  -100,  -100,  -100],
        [   72,  2644,   287,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,   826,  …,  -100,  -100,  -100],
        [   72,   588,   606,  …,  -100,  -100,  -100],
        [12518,  1312,   717,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1521, 13893,  …,    72,    72,    72],
        [  318,   764,   764,  …,    72,    72,    72],
        [  318,  1312,   389,  …,    72,    72,    72],
        …,
```

```
        [  318,  5633,  1312,  …,    72,    72,    72],
        [  318,  5633,  1312,  …,    72,    72,    72],
        [  318,  1392,   284,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[22850,  5633,  -100,  …,  -100,  -100,  -100],
        [ 5303,   837,  5156,  …,  -100,  -100,  -100],
        [43669,   837,   345,  …,  -100,  -100,  -100],
        …,
        [10919,   329,  5633,  …,  -100,  -100,  -100],
        [10919,   329,   764,  …,  -100,  -100,  -100],
        [ 5832,   423,  1392,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5633,   644,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919],
        [  318,   837,  1312,  …,    72,    72,    72],
        …,
        [  318,  1312,  1279,  …,    72,    72,    72],
        [  318,  1312,   257,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   71,  7456,  5633,  …,  -100,  -100,  -100],
        [10919,   318,   428,  …,  -100,  -100,  -100],
        [ 5832,   760,   644,  …,  -100,  -100,  -100],
        …,
        [ 8505,   837,  5633,  …,  -100,  -100,  -100],
        [ 1219,   837,   655,  …,  -100,  -100,  -100],
        [   72,   716,  7926,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,    72,  …,    72,    72,    72],
        [  318,   340,  5633,  …,    72,    72,    72],
        [  318,   345,   547,  …,    72,    72,    72],
        …,
        [  318,  1279,    72,  …, 10919, 10919, 10919],
        [  318,   423,   257,  …,    72,    72,    72],
        [  318,   307,   502,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[31373,  5633,  -100,  …,  -100,  -100,  -100],
        [ 4919,   318,   326,  …,  -100,  -100,  -100],
        [   72,  1807,   345,  …,  -100,  -100,  -100],
        …,
        [47616,  5633,  -100,  …,  -100,  -100,  -100],
        [22437,   673,   423,  …,  -100,  -100,  -100],
        [  258,   460,  1805,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  262,  389,  …,   72,   72,   72],
        [ 318, 1975, 1312,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  257,  764,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1712,   286,   606,  …,  -100,  -100,  -100],
        [   72,  2314,   764,  …,  -100,  -100,  -100],
        [   83, 40450, 15584,  …,  -100,  -100,  -100],
        …,
        [ 3919,   764,   645,  …,  -100,  -100,  -100],
        [  270,   373,   607,  …,  -100,  -100,  -100],
        [11274,  3329,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837,  514,  …,   72,   72,   72],
        [ 318,  345,  284,  …,   72,   72,   72],
        [ 318,  257,  922,  …,   72,   72,   72],
        …,
        [ 318,  837,  345,  …,   72,   72,   72],
        [ 318, 1016,  503,  …,   72,   72,   72],
        [ 318,  466,  765,  …, 4919, 4919, 4919]], device='cuda:0')
First 10 labels: tensor([[20342,   837,  1309,  …,  -100,  -100,  -100],
        [   72,  1607,   345,  …,  -100,  -100,  -100],
        [ 7091,   318,   845,  …,  -100,  -100,  -100],
        …,
        [20342,   837,   750,  …,  -100,  -100,  -100],
        [  732,   389,  2491,  …,  -100,  -100,  -100],
        [ 4919,   881,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  262,  …,   72,   72,   72],
        [ 318,  262,  764,  …,   11,   11,   11],
        [ 318, 1312,  319,  …,   72,   72,   72],
        …,
        [ 318, 1975,  284,  …,   72,   72,   72],
        [ 318,  284,  467,  …,   72,   72,   72],
        [ 318,  716,  655,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[5832,  389,  379,  …, -100, -100, -100],
        [1169, 2644,  764,  …, -100, -100, -100],
        [4053,  837, 1282,  …, -100, -100, -100],
        …,
        [  72, 2314, 4043,  …, -100, -100, -100],
        [5832,  765,  284,  …, -100, -100, -100],
        [ 392, 1312,  373,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  345,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318, 1016,  356,  …,   72,   72,   72],
        …,
```

```
        [ 318,  407,  262,  …,   72,   72,   72],
        [ 318,  284, 4203,  …,   72,   72,   72],
        [ 318,  389,  644,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  392,   810,   466,  …,  -100,  -100,  -100],
        [   64, 17025, 28752,  …,  -100,  -100,  -100],
        [  732,   389,  5633,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,   287,  …,  -100,  -100,  -100],
        [   72,  1392,   257,  …,  -100,  -100,  -100],
        [  392,   356,   760,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,   262,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   318,   262,  …,    72,    72,    72],
        …,
        [  318,  1312,   389,  …,    72,    72,    72],
        [  318,   644,    72,  …, 10919, 10919, 10919],
        [  318,  1312,   716,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[29810, 14143,   319,  …,  -100,  -100,  -100],
        [  270,   318,  3734,  …,  -100,  -100,  -100],
        [ 1169,  1660,   764,  …,  -100,  -100,  -100],
        …,
        [ 5832,  2644,   345,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        [ 3919,   837,  1312,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  530,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  407,  262,  …,   72,   72,   72],
        …,
        [ 318,  466,  345,  …,   72,   72,   72],
        [ 318,  407, 1312,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[19836,  5145,   262,  …,  -100,  -100,  -100],
        [ 5661,   640,   837,  …,  -100,  -100,  -100],
        [   72,   716,   319,  …,  -100,  -100,  -100],
        …,
        [ 4919,   534,  5633,  …,  -100,  -100,  -100],
        [   72,   716,   764,  …,  -100,  -100,  -100],
        [   13,   290,   644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312, 1545,  …,   72,   72,   72],
        [ 318, 1312, 5718,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        …,
```

```
        [ 318,  716, 1312,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318, 1312,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[3919,  837,  616,  …, -100, -100, -100],
        [3919,  837, 1288,  …, -100, -100, -100],
        [  72,  750,  407,  …, -100, -100, -100],
        …,
        [  11, 1312, 2644,  …, -100, -100, -100],
        [1481,  510,  764,  …, -100, -100, -100],
        [8505,  837, 5875,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,   764,  …,    72,    72,    72],
        [  318,   837,  1312,  …, 10919, 10919, 10919],
        [  318,   764,   764,  …,    72,    72,    72],
        …,
        [  318,   262,   291,  …,    72,    72,    72],
        [  318,   286,   661,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[ 9662,   319,   340,  …,  -100,  -100,  -100],
        [ 1219,  3763,  5145,  …,  -100,  -100,  -100],
        [31373,   837,  1888,  …,  -100,  -100,  -100],
        …,
        [10919,   546,  1931,  …,  -100,  -100,  -100],
        [  489,  3787,   286,  …,  -100,  -100,  -100],
        [10919,   318,   428,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  616, 3420,  …,   72,   72,   72],
        [ 318,  262, 5633,  …,   72,   72,   72],
        …,
        [ 318,  307,  284,  …,   72,   72,   72],
        [ 318,  345,  307,  …,   72,   72,   72],
        [ 318,  764,  837,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  568,  4950,   837,  …,  -100,  -100,  -100],
        [   72, 21368,   262,  …,  -100,  -100,  -100],
        [10919,   546,  2855,  …,  -100,  -100,  -100],
        …,
        [  986,   284,  1561,  …,  -100,  -100,  -100],
        [   72,  5465,   284,  …,  -100,  -100,  -100],
        [11545,   837,  1115,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,  716,  …,   72,   72,   72],
        [ 318,  765,  286,  …,   72,   72,   72],
        [ 318,  318,  326,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764,  534,  …,   72,   72,   72],
        [ 318,  765,  262,  …,   72,   72,   72],
        [ 318, 5633,  389,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   892,  1312,  …,  -100,  -100,  -100],
        [   72,   655,   503,  …,  -100,  -100,  -100],
        [16833,  2618,   910,  …,  -100,  -100,  -100],
        …,
        [40716,   345,   329,  …,  -100,  -100,  -100],
        [   72,   655,  1061,  …,  -100,  -100,  -100],
        [15542,  5633,   356,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  318,  407,  …,   72,   72,   72],
        [ 318,  477,  764,  …,   72,   72,   72],
        …,
        [ 318, 5633, 1312,  …,   72,   72,   72],
        [ 318,  764,  345,  …,   72,   72,   72],
        [ 318,  407,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   547,  2739,  …,  -100,  -100,  -100],
        [ 4360,   339,   318,  …,  -100,  -100,  -100],
        [ 1662,   379,   477,  …,  -100,  -100,  -100],
        …,
        [20342,  1107,   837,  …,  -100,  -100,  -100],
        [ 1662,  1654,   810,  …,  -100,  -100,  -100],
        [   72,   716,  1654,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  345,  466,  …,   72,   72,   72],
        …,
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  407, 5490,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 3919,   837,  1312,  …,  -100,  -100,  -100],
        [   72,   910,   837,  …,  -100,  -100,  -100],
        [10919,   466,   356,  …,  -100,  -100,  -100],
        …,
        [46848,   764,   783,  …,  -100,  -100,  -100],
        [ 9930,   389,   407,  …,  -100,  -100,  -100],
        [    0,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5490,  340,  …,   72,   72,   72],
        [ 318,  318,  407,  …,   72,   72,   72],
        [ 318,  345,  760,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  502,  284,  …,   72,   72,   72],
        [ 318,  345,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4598,   407,  3068,  …,  -100,  -100,  -100],
        [25991,   340,   318,  …,  -100,  -100,  -100],
        [ 4919,   466,  1312,  …,  -100,  -100,  -100],
        …,
        [ 3919,   837,   582,  …,  -100,  -100,  -100],
        [  258,  2921,   340,  …,  -100,  -100,  -100],
        [22850,   423,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  407,  …,   72,   72,   72],
        [ 318,  257,  835,  …,   72,   72,   72],
        [ 318,  257,  835,  …,  986,  986,  986],
        …,
        [ 318,  481,  307,  …,   72,   72,   72],
        [ 318,  389,  407,  …,   72,   72,   72],
        [ 318, 1110,  286,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 2611,   837,   466,  …,  -100,  -100,  -100],
        [ 8117,   318,   645,  …,  -100,  -100,  -100],
        [ 8117,   318,   645,  …,  -100,  -100,  -100],
        …,
        [19545,   345,   481,  …,  -100,  -100,  -100],
        [ 4053,   356,   466,  …,  -100,  -100,  -100],
        [ 1169,   584,  1735,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1279,    72,  …,    72,    72,    72],
        [  318,   764,   644,  …, 10919, 10919, 10919],
        [  318,  5633,  1312,  …,    72,    72,    72],
        …,
        [  318,   345,  1612,  …,    72,    72,    72],
        [  318,   530, 11020,  …, 10919, 10919, 10919],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[17749,  5633,  -100,  …,  -100,  -100,  -100],
        [ 8929, 15004,  5633,  …,  -100,  -100,  -100],
        [ 5562,   523,   764,  …,  -100,  -100,  -100],
        …,
        [10919,   466,   345,  …,  -100,  -100,  -100],
        [  258,   262,   474,  …,  -100,  -100,  -100],
        [ 4598,   340,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312, 10919,  …,    72,    72,    72],
        [  318,   837,  1279,  …,    72,    72,    72],
        [  318,   764,   764,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   588,  3772,  …,    72,    72,    72],
        [  318,  1312,   326,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  505,   764,  -100,  …,  -100,  -100,  -100],
        [ 1996,   774,  5633,  …,  -100,  -100,  -100],
        [  392,   262,  9151,  …,  -100,  -100,  -100],
        …,
        [ 9930,   389, 13526,  …,  -100,  -100,  -100],
        [   72,   561,   307,  …,  -100,  -100,  -100],
        [ 8505,   764,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  340,  …,   72,   72,   72],
        [ 318,  293,  837,  …,   72,   72,   72],
        [ 318,  644,  389,  …,   72,   72,   72],
        …,
        [ 318, 5145, 5145,  …,   11,   11,   11],
        [ 318,  340,  340,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4053,   837,  1394,  …,  -100,  -100,  -100],
        [20342,   894,    64,  …,  -100,  -100,  -100],
        [27485,  5633,   345,  …,  -100,  -100,  -100],
        …,
        [    0,  5145,  5145,  …,  -100,  -100,  -100],
        [ 1640,  1136,   546,  …,  -100,  -100,  -100],
        [ 8505,   764,   612,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  764,  …,   72,   72,   72],
        [ 318,  326,  345,  …,   72,   72,   72],
        [ 318,  428,  262,  …,   72,   72,   72],
        …,
        [ 318,  407, 1037,  …,   72,   72,   72],
        [ 318,  262,  764,  …,   72,   72,   72],
        [ 318,  407,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 7091,   318, 10218,  …,  -100,  -100,  -100],
        [10919,   318,  5633,  …,  -100,  -100,  -100],
        [ 8727,   318,  1392,  …,  -100,  -100,  -100],
        …,
        [   72,   714,   407,  …,  -100,  -100,  -100],
        [ 1044,   286,  1918,  …,  -100,  -100,  -100],
        [ 5832,   481,   423,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  262,  …,   72,   72,   72],
        [ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  345,  651,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1392, 1775,  …,   72,   72,   72],
        [ 318,  764,  644,  …,   72,   72,   72],
        [ 318,  764, 1312,  …, 5832, 5832, 5832]], device='cuda:0')
First 10 labels: tensor([[ 1169,   318,   422,  …,  -100,  -100,  -100],
        [ 3919,   837,   326,  …,  -100,  -100,  -100],
        [ 3003,   750,   345,  …,  -100,  -100,  -100],
        …,
        [   72,   423,  1239,  …,  -100,  -100,  -100],
        [  272,  6306,  5633,  …,  -100,  -100,  -100],
        [15390,   890,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  691,  …,   72,   72,   72],
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  530,  764,  …,   72,   72,   72],
        …,
        [ 318, 1392,  467,  …,   72,   72,   72],
        [ 318,  837,  764,  …,   72,   72,   72],
        [ 318,  835,  981,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  270,   318,   262,  …,  -100,  -100,  -100],
        [  282,   292,   837,  …,  -100,  -100,  -100],
        [ 8807,   262,   547,  …,  -100,  -100,  -100],
        …,
        [   72,   423,   284,  …,  -100,  -100,  -100],
        [15596,   935,  7396,  …,  -100,  -100,  -100],
        [  259,   257,  1310,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  439, 3364,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  423, 1312,  …,   72,   72,   72],
        …,
        [ 318,  257,  257,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  345,  547,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[11338,  2809,   707,  …,  -100,  -100,  -100],
        [  986,   880,   837,  …,  -100,  -100,  -100],
        [ 1326,   991,   764,  …,  -100,  -100,  -100],
        …,
        [ 8117,   318,   991,  …,  -100,  -100,  -100],
        [ 1069, 24342,  5145,  …,  -100,  -100,  -100],
        [ 5832,   531,   356,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  407,  284,  …,   72,   72,   72],
        [ 318,  407,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318, 5633, 5633,  …,   72,   72,   72],
        [ 318,  345, 1612,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   466,   407,  …,  -100,  -100,  -100],
        [  732,   547,  1016,  …,  -100,  -100,  -100],
        [   72,   716,  6405,  …,  -100,  -100,  -100],
        …,
        [ 8727,   607,  3774,  …,  -100,  -100,  -100],
        [39664,   466,   345,  …,  -100,  -100,  -100],
        [ 4360,  6004,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,  1612,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72],
        [  318,  1392,   351,  …,    72,    72,    72],
        …,
        [  318,   428,  3516,  …, 10919, 10919, 10919],
        [  318,   502,   837,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   466,   345,  …,  -100,  -100,  -100],
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        [   72,   423,  3111,  …,  -100,  -100,  -100],
        …,
        [ 8727,   318,   262,  …,  -100,  -100,  -100],
        [41194,  1904,   502,  …,  -100,  -100,  -100],
        [ 3919,   764,   645,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   866,   764,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72],
        [  318,   503,   345,  …,    72,    72,    72],
        …,
        [  318,  1975,   340,  …,    72,    72,    72],
        [  318,   760,   644,  …,    72,    72,    72],
        [  318,   345,   466,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[ 8968,   262,  8223,  …,  -100,  -100,  -100],
        [  439,   826,   837,  …,  -100,  -100,  -100],
        [ 5832, 36193,  5633,  …,  -100,  -100,  -100],
        …,
        [   72,  2314,   466,  …,  -100,  -100,  -100],
        [ 4598,   345,  3505,  …,  -100,  -100,  -100],
        [10919,   750,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633,   72,  …,   72,   72,   72],
        [ 318, 1312, 2073,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  318,  345,  …,   72,   72,   72],
        [ 318,  606,  640,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8727,  5633,  -100,  …,  -100,  -100,  -100],
        [19532,   764,  1997,  …,  -100,  -100,  -100],
        [23442, 20712, 17372,  …,  -100,  -100,  -100],
        …,
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [   30,   810,   389,  …,  -100,  -100,  -100],
        [11246,   286,   262,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   318,   764,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919],
        [  318,   764,   764,  …,    72,    72,    72],
        …,
        [  318,   345,  5633,  …,    72,    72,    72],
        [  318,   467,   764,  …,    72,    72,    72],
        [  318,   284,   318,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   11,   340,   994,  …,  -100,  -100,  -100],
        [10919,   318,   326,  …,  -100,  -100,  -100],
        [   72,   345,  3026,  …,  -100,  -100,  -100],
        …,
        [ 8727,   481,  5633,  …,  -100,  -100,  -100],
        [  732,  1276, 23290,  …,  -100,  -100,  -100],
        [  372,   703,   673,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,   530,  …,    72,    72,    72],
        [  318,   326,  1804,  …, 10919, 10919, 10919],
        [  318,  1312,   716,  …,    72,    72,    72],
        …,
        [  318,   837,  5664,  …,    72,    72,    72],
        [  318,   407,  2739,  …,    72,    72,    72],
        [  318,   428,  5633,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  533,   407,   645,  …,  -100,  -100,  -100],
        [10919,   318,   339,  …,  -100,  -100,  -100],
        [ 3919,   837,  1312,  …,  -100,  -100,  -100],
        …,
        [   11,  4043,   257,  …,  -100,  -100,  -100],
        [  270,   318,  1165,  …,  -100,  -100,  -100],
        [ 8727,   318,  7348,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407, 5633,  …,   72,   72,   72],
        [ 318, 1683,  257,  …,   72,   72,   72],
        [ 318,  407,  994,  …,   72,   72,   72],
        …,
```

```
        [ 318,  407, 1312,  …,   72,   72,   72],
        [ 318,  318,  345,  …,   72,   72,   72],
        [ 318,  318, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   389,  1654,  …,  -100,  -100,  -100],
        [14150,   345,  1392,  …,  -100,  -100,  -100],
        [   72,   716,  1464,  …,  -100,  -100,  -100],
        …,
        [ 5832,   466,  5633,  …,  -100,  -100,  -100],
        [10919,  2073,  5633,  …,  -100,  -100,  -100],
        [   72,   428,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  644,   72,  …,   72,   72,   72],
        [ 318, 5633,  428,  …,   72,   72,   72],
        [ 318, 1392,  284,  …,   72,   72,   72],
        …,
        [ 318, 1312,  467,  …,   72,   72,   72],
        [ 318,  760,  651,  …,   72,   72,   72],
        [ 318,  481,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10919,  5633,  -100,  …,  -100,  -100,  -100],
        [ 2339,   428,   588,  …,  -100,  -100,  -100],
        [   72,   423,   587,  …,  -100,  -100,  -100],
        …,
        [ 3003,   815,  1312,  …,  -100,  -100,  -100],
        [ 4598,   345,  1683,  …,  -100,  -100,  -100],
        [ 8524,   356,   761,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  284,  764,  …,   72,   72,   72],
        [ 318,  257,  764,  …,   72,   72,   72],
        [ 318,  345,  716,  …,   72,   72,   72],
        …,
        [ 318,  466,  262,  …,   72,   72,   72],
        [ 318,  826,  611,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 9930,   761,  1037,  …,  -100,  -100,  -100],
        [26535,   514,   534,  …,  -100,  -100,  -100],
        [47984,  5145,  1312,  …,  -100,  -100,  -100],
        …,
        [ 2197,   703,   318,  …,  -100,  -100,  -100],
        [ 5832,   477,  2000,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,   764,  …,    72,    72,    72],
        [  318,   407,  1016,  …,    72,    72,    72],
        [  318,   764,   262,  …,    72,    72,    72],
        …,
```

```
        [  318,   389,  1392,  …,    72,    72,    72],
        [  318,   716,   407,  …,    72,    72,    72],
        [  318,   428,  5633,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[ 1820,  1545, 14509,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100],
        [ 1169,  2156,   284,  …,  -100,  -100,  -100],
        …,
        [   11,   345,   423,  …,  -100,  -100,  -100],
        [  361,  1312,   716,  …,  -100,  -100,  -100],
        [ 8727,   318,  5633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,  764,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  262,  764,  …,   72,   72,   72],
        …,
        [ 318, 5633,  466,  …,   72,   72,   72],
        [ 318,  345,  318,  …,   72,   72,   72],
        [ 318,  764,  307,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,  1975,   262,  …,  -100,  -100,  -100],
        [ 4053,   837,   345,  …,  -100,  -100,  -100],
        [ 1662,   422,   345,  …,  -100,  -100,  -100],
        …,
        [42949,  1997,   284,  …,  -100,  -100,  -100],
        [   72,  2911,   326,  …,  -100,  -100,  -100],
        [ 7091,   655,   284,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        …,
        [ 318,  326, 2300,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  389,  407,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1219,   645,   837,  …,  -100,  -100,  -100],
        [  439,   826,   764,  …,  -100,  -100,  -100],
        [ 8505,   837,  1654,  …,  -100,  -100,  -100],
        …,
        [10919,   318,   262,  …,  -100,  -100,  -100],
        [ 1456,   764,   703,  …,  -100,  -100,  -100],
        [ 3919,   484,   481,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   345,  …,    72,    72,    72],
        [  318,   257,  1312,  …,    72,    72,    72],
        [  318,   345,   345,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,   340,  …,    72,    72,    72],
        [  318,   318,   764,  …,    72,    72,    72],
        [  318,   502,   257,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[ 9541,   837,   611,  …,  -100,  -100,  -100],
        [  270,   373,   764,  …,  -100,  -100,  -100],
        [33331,   607,   644,  …,  -100,  -100,  -100],
        …,
        [  270,   318,  5633,  …,  -100,  -100,  -100],
        [ 5832,   340,   837,  …,  -100,  -100,  -100],
        [   77,  2900,   656,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  321,  …,   72,   72,   72],
        [ 318, 3516,  318,  …,   72,   72,   72],
        [ 318,  345,  644,  …,   72,   72,   72],
        …,
        [ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  345,  257,  …,   72,   72,   72],
        [ 318,  262,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8505,   837,  8805,  …,  -100,  -100,  -100],
        [25991,   262,  7733,  …,  -100,  -100,  -100],
        [10919,  5145,    30,  …,  -100,  -100,  -100],
        …,
        [ 1219,   764,   345,  …,  -100,  -100,  -100],
        [   72,  1807,   329,  …,  -100,  -100,  -100],
        [10919,   546,  5633,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   345,  …, 10919, 10919, 10919],
        [  318,   606,   389,  …,    72,    72,    72],
        [  318,   284,  5633,  …,    72,    72,    72],
        …,
        [  318,  1683,   651,  …, 10919, 10919, 10919],
        [  318,  1312,  1312,  …,    72,    72,    72],
        [  318,  1312,   318,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 8727,   857,  5633,  …,  -100,  -100,  -100],
        [11246,   286,   345,  …,  -100,  -100,  -100],
        [ 5832,  1016, 23586,  …,  -100,  -100,  -100],
        …,
        [20839,   345,  1683,  …,  -100,  -100,  -100],
        [43669,  2644,  2644,  …,  -100,  -100,  -100],
        [   72,  4724,   326,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  318,  644,  …,   72,   72,   72],
        [ 318,  502,  837,  …,   72,   72,   72],
        …,
```

```
        [ 318,  588, 5633,  …,   72,   72,   72],
        [ 318, 1392,  898,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[43669,   837,   475,  …,  -100,  -100,  -100],
        [  361,   326,   318,  …,  -100,  -100,  -100],
        [41194,  1904,   502,  …,  -100,  -100,  -100],
        …,
        [ 5832,  2936,   340,  …,  -100,  -100,  -100],
        [   72,   423,   616,  …,  -100,  -100,  -100],
        [ 1219,   837,   645,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1279, 5832,  …, 5832, 5832, 5832],
        [ 318,  345,  760,  …,   72,   72,   72],
        [ 318, 7319, 1312,  …,   72,   72,   72],
        …,
        [ 318,  407,  764,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  345,  760,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  929,  5145,  -100,  …,  -100,  -100,  -100],
        [ 4919,   481,  1312,  …,  -100,  -100,  -100],
        [11545,  3470,   764,  …,  -100,  -100,  -100],
        …,
        [  270,   318,  7932,  …,  -100,  -100,  -100],
        [39969,  3641,   837,  …,  -100,  -100,  -100],
        [ 4919,   466,   345,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  326, 5633,  …,   72,   72,   72],
        [ 318, 1016,  764,  …,   72,   72,   72],
        [ 318,  466,  340,  …,   72,   72,   72],
        …,
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  345,  764,  …,   72,   72,   72],
        [ 318,  257,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8727,   373,   339,  …,  -100,  -100,  -100],
        [  732,   389,   922,  …,  -100,  -100,  -100],
        [  568,  1521,   318,  …,  -100,  -100,  -100],
        …,
        [ 1219,   837,   645,  …,  -100,  -100,  -100],
        [   72,  2051,   883,  …,  -100,  -100,  -100],
        [40684,   373,  1593,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  467,  764,  …,   72,   72,   72],
        [ 318,  345,  389,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        …,
```

```
        [ 318,  837,  514,  …,   72,   72,   72],
        [ 318,  423,  326,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8117,   345,   467,  …,  -100,  -100,  -100],
        [ 3826,  5145,   345,  …,  -100,  -100,  -100],
        [ 8505,   837,   345,  …,  -100,  -100,  -100],
        …,
        [11085,   572,  1309,  …,  -100,  -100,  -100],
        [   72,   991,  2565,  …,  -100,  -100,  -100],
        [ 5832,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  837,  764,  …,   72,   72,   72],
        [ 318,  837,  389,  …,   72,   72,   72],
        …,
        [ 318,  407,  307,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318, 3516,  286,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1219,   837,   326,  …,  -100,  -100,  -100],
        [   71,  1419, 22977,  …,  -100,  -100,  -100],
        [33331,   502,   345,  …,  -100,  -100,  -100],
        …,
        [   72,   481,   407,  …,  -100,  -100,  -100],
        [   82,   343,   837,  …,  -100,  -100,  -100],
        [15898,   257,  1256,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  481, 1016,  …,   72,   72,   72],
        [ 318,  318,  407,  …,   72,   72,   72],
        [ 318,  345, 1612,  …,   72,   72,   72],
        …,
        [ 318,  345,  716,  …,   72,   72,   72],
        [ 318,  407, 1312,  …,   72,   72,   72],
        [ 318,  651,  257,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 8524,  1312,   716,  …,  -100,  -100,  -100],
        [   13,   673,   318,  …,  -100,  -100,  -100],
        [10919,   466,   345,  …,  -100,  -100,  -100],
        …,
        [11358,   837,  1312,  …,  -100,  -100,  -100],
        [ 5832,   466,  5633,  …,  -100,  -100,  -100],
        [ 5171,  1312,   423,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1647,   764,  …,    72,    72,    72],
        [  318,  1560,   502,  …, 10919, 10919, 10919],
        [  318,  5490,   502,  …,    72,    72,    72],
        …,
```

```
        [  318,   837,   257,  …,    72,    72,    72],
        [  318,   345,   466,  …,    72,    72,    72],
        [  318,   407,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  392,   256, 24275,  …,  -100,  -100,  -100],
        [ 5171,   345,  1560,  …,  -100,  -100,  -100],
        [ 4598,   407,  2209,  …,  -100,  -100,  -100],
        …,
        [26000,  1040,   318,  …,  -100,  -100,  -100],
        [22850,   750,   345,  …,  -100,  -100,  -100],
        [ 5832,   481,  4601,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  345, 1438,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        …,
        [ 318,  407, 7926,  …,   72,   72,   72],
        [ 318,  257,  340,  …,   72,   72,   72],
        [ 318,  407,  466,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[24494,   837,   837,  …,  -100,  -100,  -100],
        [  265,  1551,   534,  …,  -100,  -100,  -100],
        [ 3137,  7415,   764,  …,  -100,  -100,  -100],
        …,
        [   72,   716,   523,  …,  -100,  -100,  -100],
        [  271,   428,  1804,  …,  -100,  -100,  -100],
        [ 3137,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   329,   257,  …,    72,    72,    72],
        [  318,  5633,  5633,  …, 10919, 10919, 10919],
        [  318,  1312,   837,  …,    72,    72,    72],
        …,
        [  318,   835,   644,  …,    72,    72,    72],
        [  318,   428,  5633,  …,    72,    72,    72],
        [  318,  1265,   340,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   670,   351,  …,  -100,  -100,  -100],
        [ 1640,   644,  4007,  …,  -100,  -100,  -100],
        [ 1219,   837,   880,  …,  -100,  -100,  -100],
        …,
        [ 2188,   326,   318,  …,  -100,  -100,  -100],
        [ 8727,   318,   339,  …,  -100,  -100,  -100],
        [11261,  1312,  2005,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5770,   764,  …,    72,    72,    72],
        [  318,  5633,   340,  …, 10919, 10919, 10919],
        [  318,   257,   284,  …,    72,    72,    72],
        …,
```

```
        [  318,   466,   345,  …,    72,    72,    72],
        [  318,   428,  5633,  …,    72,    72,    72],
        [  318,   764,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  986,   616,  1842,  …,  -100,  -100,  -100],
        [10919, 13766,   373,  …,  -100,  -100,  -100],
        [  271,   428,  1016,  …,  -100,  -100,  -100],
        …,
        [  392,  1521,   561,  …,  -100,  -100,  -100],
        [ 8727,   318,   428,  …,  -100,  -100,  -100],
        [   72,   760,   644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257, 4144,  …,   72,   72,   72],
        [ 318,  764,  826,  …,   72,   72,   72],
        [ 318,  588,  284,  …,   72,   72,   72],
        …,
        [ 318,  503,  764,  …,   72,   72,   72],
        [ 318,  262,  257,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[26535,   683,   257,  …,  -100,  -100,  -100],
        [25833,   306,   477,  …,  -100,  -100,  -100],
        [   72,   561,   588,  …,  -100,  -100,  -100],
        …,
        [47408,   502,  3436,  …,  -100,  -100,  -100],
        [ 2704,   541,   373,  …,  -100,  -100,  -100],
        [ 4053,   837,  1312,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  284,  …,   72,   72,   72],
        [ 318,  278, 1295,  …,   72,   72,   72],
        [ 318,  284,  760,  …,   72,   72,   72],
        …,
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  764,  345,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   716,   510,  …,  -100,  -100,  -100],
        [  260,  6604,   616,  …,  -100,  -100,  -100],
        [   72,   765,   284,  …,  -100,  -100,  -100],
        …,
        [  305,  1362,   837,  …,  -100,  -100,  -100],
        [ 1513,   837,   611,  …,  -100,  -100,  -100],
        [44353,   340,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  257,  284,  …,   72,   72,   72],
        [ 318, 1279,  837,  …,   72,   72,   72],
        …,
```

```
        [ 318,  716, 1016,  …,   72,   72,   72],
        [ 318,  257,  644,  …,   72,   72,   72],
        [ 318,  345,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1640, 11060,   764,  …,  -100,  -100,  -100],
        [  270,   373,  1327,  …,  -100,  -100,  -100],
        [20285,  5633, 11752,  …,  -100,  -100,  -100],
        …,
        [  392,  1312,   716,  …,  -100,  -100,  -100],
        [  271,   428,  1107,  …,  -100,  -100,  -100],
        [ 2339,   837,  4556,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,  5832,  …, 10919, 10919, 10919],
        [  318,   764,  1016,  …,    72,    72,    72],
        [  318,   407,   546,  …,    72,    72,    72],
        …,
        [  318,   407,  1016,  …,    72,    72,    72],
        [  318,  1312,  1312,  …, 10919, 10919, 10919],
        [  318,  1312,  1256,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[13261,   764,  -100,  …,  -100,  -100,  -100],
        [ 2433,  1223,   318,  …,  -100,  -100,  -100],
        [   72,   716,  3612,  …,  -100,  -100,  -100],
        …,
        [   72,   716,   407,  …,  -100,  -100,  -100],
        [ 5460,   837,   837,  …,  -100,  -100,  -100],
        [11274,   764,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  407,  …,   72,   72,   72],
        [ 318,  866, 3516,  …,   72,   72,   72],
        [ 318,  837,   72,  …,   72,   72,   72],
        …,
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,   74,  287,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  392,   340,   318,  …,  -100,  -100,  -100],
        [   72, 18283,   428,  …,  -100,  -100,  -100],
        [31373,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [43669,   764,  -100,  …,  -100,  -100,  -100],
        [11246,   837,  2506,  …,  -100,  -100,  -100],
        [   82,   710,   461,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,   257,  …,    72, 10919,    72],
        [  318,   345,   318,  …,    72,    72,    72],
        [  318,   340,   534,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,   546,  …,    72,    72,    72],
        [  318,   764,   764,  …,    72,    72,    72],
        [  318,   257,   649,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  533,   407,   345,  …,  -100,  -100,  -100],
        [  568,   837,   644,  …,  -100,  -100,  -100],
        [47391,  9580,   278,  …,  -100,  -100,  -100],
        …,
        [   72,   716,  3612,  …,  -100,  -100,  -100],
        [   11,   922,  8171,  …,  -100,  -100,  -100],
        [ 5562,  2822,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1654,  284,  …,   72,   72,   72],
        [ 318,  514,  651,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        …,
        [ 318, 1265, 1282,  …,   72,   72,   72],
        [ 318,  764,  286,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  533,   345,  1016,  …,  -100,  -100,  -100],
        [   11,  1309,   502,  …,  -100,  -100,  -100],
        [23936,  8957,   837,  …,  -100,  -100,  -100],
        …,
        [11261,  1312,  3387,  …,  -100,  -100,  -100],
        [ 1136,   262,   503,  …,  -100,  -100,  -100],
        [ 5832,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  307,  345,  …,   72,   72,   72],
        [ 318,  257,  922,  …,   72,   72,   72],
        [ 318,  588, 1312,  …,   72,   72,   72],
        …,
        [ 318,  345,  764,  …,   72,   72,   72],
        [ 318, 1312,  389,  …,   72,   72,   72],
        [ 318,  318,  764,  …,  986,  986,  986]], device='cuda:0')
First 10 labels: tensor([[  258,   481,  1309,  …,  -100,  -100,  -100],
        [ 7091,   318,   257,  …,  -100,  -100,  -100],
        [   72,  2936,   588,  …,  -100,  -100,  -100],
        …,
        [ 8807,   389,   616,  …,  -100,  -100,  -100],
        [ 4053,   837,   484,  …,  -100,  -100,  -100],
        [  986,   326, 13774,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,   345,  …, 10919, 10919, 10919],
        [  318,   345,  1760,  …, 10919, 10919, 10919],
        [  318,   572,   764,  …,    72,    72,    72],
        …,
```

```
        [  318,  5633,   428,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   588,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   65,   463,  5633,  …,  -100,  -100,  -100],
        [10919,   423,   345,  …,  -100,  -100,  -100],
        [15344,   340,   510,  …,  -100,  -100,  -100],
        …,
        [ 2339,   428,  5633,  …,  -100,  -100,  -100],
        [   72,   716,  7926,  …,  -100,  -100,  -100],
        [ 5832,   804,  4950,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  389,  407,  …, 5832, 5832, 5832],
        [ 318,  318,  262,  …,   72,   72,   72],
        [ 318,  837,  716,  …,   72,   72,   72],
        …,
        [ 318,  764,  286,  …,   72,   72,   72],
        [ 318,  345,  373,  …,   72,   72,   72],
        [ 318, 1312,  257,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4360,   345,   547,  …,  -100,  -100,  -100],
        [  361,   428,   318,  …,  -100,  -100,  -100],
        [ 5832,   760,  1312,  …,  -100,  -100,  -100],
        …,
        [16833,  7968,   530,  …,  -100,  -100,  -100],
        [   72,  1807,  1312,  …,  -100,  -100,  -100],
        [ 4053,   837,   655,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  588,  284,  …,   72,   72,   72],
        [ 318,  284,  835,  …,   72,   72,   72],
        [ 318,  345, 1804,  …,   72,   72,   72],
        …,
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  760,  423,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   561,   423,  …,  -100,  -100,  -100],
        [ 5832,  1561,   262,  …,  -100,  -100,  -100],
        [10919,   389,   345,  …,  -100,  -100,  -100],
        …,
        [12957,  1755,   764,  …,  -100,  -100,  -100],
        [29688,   467,   319,  …,  -100,  -100,  -100],
        [ 4598,   345,   991,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,   262,  …,    72,    72,    72],
        [  318,   340,   764,  …,   732,   732,   732],
        [  318,   837,  1312,  …,    72,    72,    72],
        …,
```

```
        [  318,  1312,   345,  …,    72,    72,    72],
        [  318,  1312,   340,  …,    72,    72,    72],
        [  318,   764,  1310,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[10919,   714,   307,  …,  -100,  -100,  -100],
        [ 1640,  1136,   340,  …,  -100,  -100,  -100],
        [  282,  3506,   837,  …,  -100,  -100,  -100],
        …,
        [ 4053, 11485,   750,  …,  -100,  -100,  -100],
        [ 3506,   764,  2222,  …,  -100,  -100,  -100],
        [29810,  1755,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  407,  …,   72,   72,   72],
        [ 318,  345,  460,  …,   72,   72,   72],
        [ 318,  467,  546,  …,   72,   72,   72],
        …,
        [ 318,  502,  339,  …,   72,   72,   72],
        [ 318,  345,  655,  …,   72,   72,   72],
        [ 318,  284, 6136,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   11,   340,   318,  …,  -100,  -100,  -100],
        [   72,  2911,   356,  …,  -100,  -100,  -100],
        [ 1616,   514,  1561,  …,  -100,  -100,  -100],
        …,
        [  258,  1297,   502,  …,  -100,  -100,  -100],
        [   66, 34574,   356,  …,  -100,  -100,  -100],
        [ 5832,   761,   281,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,   546,  …,    72,    72,    72],
        [  318,  5490,   502,  …,    72,    72,    72],
        [  318,  5633,  5633,  …,    72,    72,    72],
        …,
        [  318,   764,   345,  …,    72,    72,    72],
        [  318,   345,   503,  …, 10919, 10919, 10919],
        [  318,  1279,   262,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 4053,   837,   892,  …,  -100,  -100,  -100],
        [ 4598,   407,  1560,  …,  -100,  -100,  -100],
        [  986, 24926, 24926,  …,  -100,  -100,  -100],
        …,
        [   65, 15352,   837,  …,  -100,  -100,  -100],
        [47984,  5145,   651,  …,  -100,  -100,  -100],
        [40838,  2644,   351,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,  1016,  …,    72,    72,    72],
        [  318,  1808,   764,  …, 10919, 10919, 10919],
        [  318,   318,  2392,  …,    72,    72,    72],
        …,
```

```
        [  318,   760,   326,  …,    72,    72,    72],
        [  318,   502,  5633,  …,    72,    72,    72],
        [  318,   318,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   716,   407,  …,  -100,  -100,  -100],
        [41484,   262,  1808,  …,  -100,  -100,  -100],
        [ 1820,  2612,   645,  …,  -100,  -100,  -100],
        …,
        [ 5832,  1541,   531,  …,  -100,  -100,  -100],
        [22850,  1560,   502,  …,  -100,  -100,  -100],
        [ 1820,  1438,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   345,  …, 10919, 10919, 10919],
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        …,
        [  318,   345,  1775,  …,    72,    72,    72],
        [  318,  1312,  1865,  …,    72,    72,    72],
        [  318,  5490,   502,  …,  5832,  5832,  5832]], device='cuda:0')
First 10 labels: tensor([[ 5832,   750,  2644,  …,  -100,  -100,  -100],
        [ 8505,   837, 45421,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        …,
        [14150,   407,  1312,  …,  -100,  -100,  -100],
        [ 3919,   764,   407,  …,  -100,  -100,  -100],
        [ 4598,   407,  1560,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1392,  284,  …,  732,  732,  732],
        [ 318,  284, 1312,  …,   72,   72,   72],
        [ 318,  502,  837,  …,   72,   72,   72],
        …,
        [ 318,  407,  760,  …,  258,  258,  258],
        [ 318,  587,  764,  …,   72,   72,   72],
        [ 318,  326,  428,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  732,   423,  1392,  …,  -100,  -100,  -100],
        [10919,   765,  5633,  …,  -100,  -100,  -100],
        [41194,  1904,   502,  …,  -100,  -100,  -100],
        …,
        [   72,   466,   407,  …,  -100,  -100,  -100],
        [27238,   423,   587,  …,  -100,  -100,  -100],
        [10919,   318,   477,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  389,  423,  …,   72,   72,   72],
        [ 318,  262, 5633,  …,   72,   72,   72],
        [ 318,  826,  644,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1683,  262,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[25991,   345,   815,  …,  -100,  -100,  -100],
        [10919,   546,   606,  …,  -100,  -100,  -100],
        [ 5562,   318,   407,  …,  -100,  -100,  -100],
        …,
        [14150,   345,   286,  …,  -100,  -100,  -100],
        [ 1662,  5000,   764,  …,  -100,  -100,  -100],
        [   71,  1419,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1392,   284,  …,    72,    72,    72],
        [  318,   284,   651,  …,  5832,  5832,  5832],
        [  318,   345,   910,  …, 10919, 10919, 10919],
        …,
        [  318,   284,  5633,  …,    72,    72,    72],
        [  318,   764,   389,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   423,  1223,  …,  -100,  -100,  -100],
        [ 1640,  2111,   284,  …,  -100,  -100,  -100],
        [10919,   750,   339,  …,  -100,  -100,  -100],
        …,
        [ 5832, 16453,   428,  …,  -100,  -100,  -100],
        [ 2777, 49910,   356,  …,  -100,  -100,  -100],
        [ 5661, 12625,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1521, 13893,  …,    72,    72,    72],
        [  318,   407,   922,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        …,
        [  318,   644,    72,  …,    72,    72,    72],
        [  318,  1312,  1312,  …,    72,    72,    72],
        [  318,   826,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[22850,  5633,  -100,  …,  -100,  -100,  -100],
        [ 5832,   389,   257,  …,  -100,  -100,  -100],
        [   72,   750,   407,  …,  -100,  -100,  -100],
        …,
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        [ 8505,   764,   290,  …,  -100,  -100,  -100],
        [ 5562,   318,   826,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5968,  345,  …,   72,   72,   72],
        [ 318, 1654,  379,  …,   72,   72,   72],
        [ 318, 1312, 1312,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  644,   72,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[8727,  262,  389,  …, -100, -100, -100],
        [ 533,  345, 1257,  …, -100, -100, -100],
        [3919,  764,  290,  …, -100, -100, -100],
        …,
        [  82, 6548,  837,  …, -100, -100, -100],
        [1219,  837, 1312,  …, -100, -100, -100],
        [5832, 5633, -100,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   257,  …,    72,    72,    72],
        [  318,  5633,   644,  …, 10919, 10919, 10919],
        [  318,  1312,    68,  …,    72,    72,    72],
        …,
        [  318,  3516,  1016,  …,    72,    72,    72],
        [  318,   407,  1312,  …,    72,    72,    72],
        [  318,   407,   257,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  270,   318,   587,  …,  -100,  -100,  -100],
        [24815,   644,  5633,  …,  -100,  -100,  -100],
        [ 1219,   837,  4273,  …,  -100,  -100,  -100],
        …,
        [  271,   262,  1097,  …,  -100,  -100,  -100],
        [   72,   716,  2644,  …,  -100,  -100,  -100],
        [  270,   318,   588,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[318, 345, 892,  …,  72,  72,  72],
        [318, 835, 457,  …,  72,  72,  72],
        [318, 407, 760,  …,  72,  72,  72],
        …,
        [318, 389, 284,  …,  72,  72,  72],
        [318, 764, 284,  …,  72,  72,  72],
        [318, 284, 286,  …,  72,  72,  72]], device='cuda:0')
First 10 labels: tensor([[ 3003,   466,   345,  …,  -100,  -100,  -100],
        [ 1525,   262,   390,  …,  -100,  -100,  -100],
        [   72,   466,   407,  …,  -100,  -100,  -100],
        …,
        [12518,   345,   467,  …,  -100,  -100,  -100],
        [19412,   764,  1392,  …,  -100,  -100,  -100],
        [  986,  3892,   503,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  257,  257,  …,   72,   72,   72],
        [ 318,  345,  373,  …,   72,   72,   72],
        [ 318,  340,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  407,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  258,   318,   407,  …,  -100,  -100,  -100],
        [   72,  1807,   428,  …,  -100,  -100,  -100],
        [  732,   925,   340,  …,  -100,  -100,  -100],
        …,
        [42773,   764,  1654,  …,  -100,  -100,  -100],
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        [   72,   716,  4684,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,   262,  …,    72,    72,    72],
        [  318,   644,    72,  …,    72,    72,    72],
        [  318,  1312,  3768,  …,    72,    72,    72],
        …,
        [  318,   502,  4585,  …,    72,    72,    72],
        [  318,  1312, 10919,  …, 10919, 10919, 10919],
        [  318,  1312,   318,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[24130,  9673,   832,  …,  -100,  -100,  -100],
        [27485,  5633,  -100,  …,  -100,  -100,  -100],
        [ 3919,   837,   308,  …,  -100,  -100,  -100],
        …,
        [ 5832,  4585,   290,  …,  -100,  -100,  -100],
        [22366,   764,  -100,  …,  -100,  -100,  -100],
        [43669,   837,   340,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5633,   644,  …, 10919, 10919, 10919],
        [  318,  1312,  5832,  …,    72,    72,    72],
        [  318,   837,   837,  …,    72,    72,    72],
        …,
        [  318,   837,   284,  …,    72,    72,    72],
        [  318,  1312,   389,  …,    72,    72,    72],
        [  318,   764,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 8505,   644,  5633,  …,  -100,  -100,  -100],
        [ 3919,   764,  -100,  …,  -100,  -100,  -100],
        [ 8310,   962,   306,  …,  -100,  -100,  -100],
        …,
        [   65, 11788, 10291,  …,  -100,  -100,  -100],
        [18223,   837,   345,  …,  -100,  -100,  -100],
        [   72,  1833,  1521,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   326,  5633,  …,    72,    72,    72],
        [  318,   262,  5633,  …, 10919, 10919, 10919],
        [  318,   716,   407,  …,    72,    72,    72],
        …,
```

```
        [  318,   345,   547,  …,    72,    72,    72],
        [  318,  1312,   389,  …,  5832,  5832,  5832],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   318,   340,  …,  -100,  -100,  -100],
        [10919,   546,   616,  …,  -100,  -100,  -100],
        [ 4360,  1312,   481,  …,  -100,  -100,  -100],
        …,
        [   72,  1807,   345,  …,  -100,  -100,  -100],
        [ 3919,   837,   345,  …,  -100,  -100,  -100],
        [42994,   407,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,  1016,  …,    72,    72,    72],
        [  318,  1312,   716,  …,    72,    72,    72],
        [  318,   262,  1279,  …, 10919, 10919, 10919],
        …,
        [  318,  1392,  4203,  …,    72,    72,    72],
        [  318,  1016,  1016,  …,    72,    72,    72],
        [  318,  1683,   284,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,   716,   407,  …,  -100,  -100,  -100],
        [ 8505,   837,  1312,  …,  -100,  -100,  -100],
        [10724,  1068,  5633,  …,  -100,  -100,  -100],
        …,
        [   72,   423,   257,  …,  -100,  -100,  -100],
        [  732,   389,   407,  …,  -100,  -100,  -100],
        [14150,   345,   587,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  837,  837,  …,   72,   72,   72],
        [ 318,  326, 5633,  …,   72,   72,   72],
        [ 318,  257,  835,  …,   72,   72,   72],
        …,
        [ 318,  826, 1312,  …,   72,   72,   72],
        [ 318,  262,  764,  …, 5832, 5832, 5832],
        [ 318, 1312,  716,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  986, 14428,    88,  …,  -100,  -100,  -100],
        [10919,   373,   326,  …,  -100,  -100,  -100],
        [ 8117,   318,   645,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,  1521,  …,  -100,  -100,  -100],
        [ 8807,   287,   262,  …,  -100,  -100,  -100],
        [14925,   764,  1312,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318, 1975,  329,  …,   72,   72,   72],
        [ 318, 1310,  284,  …, 5832, 5832, 5832],
        …,
```

```
        [ 318,  644,  389,  …,   72,   72,   72],
        [ 318,  318,  508,  …,   72,   72,   72],
        [ 318,  356,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   466,   407,  …,  -100,  -100,  -100],
        [   72,  2314,  2740,  …,  -100,  -100,  -100],
        [23442,   257,  5664,  …,  -100,  -100,  -100],
        …,
        [10919,  5633,   661,  …,  -100,  -100,  -100],
        [   30,   508,  5633,  …,  -100,  -100,  -100],
        [  732,  5633,   466,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   326,  1612,  …, 10919, 10919, 10919],
        [  318,   423,   587,  …,    72,    72,    72],
        [  318,   286,   651,  …,  5832,  5832,  5832],
        …,
        [  318,   765,  1310,  …, 10919, 10919, 10919],
        [  318,   826,   691,  …,    72,    72,    72],
        [  318,   826,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   857,   340,  …,  -100,  -100,  -100],
        [ 4919,   890,   339,  …,  -100,  -100,  -100],
        [ 5832,   503,   284,  …,  -100,  -100,  -100],
        …,
        [   72,   655,   257,  …,  -100,  -100,  -100],
        [ 5562,   318,   262,  …,  -100,  -100,  -100],
        [ 5562,   318,  4998,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,  1312,  …, 10919, 10919, 10919],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   284,   284,  …,    72,    72,    72],
        …,
        [  318,  5490,   340,  …,    72,    72,    72],
        [  318,   644,    72,  …,    72,    72,    72],
        [  318,  1654,   994,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 1069, 24342,   644,  …,  -100,  -100,  -100],
        [ 3506,   994,   764,  …,  -100,  -100,  -100],
        [   72,  3088,  3375,  …,  -100,  -100,  -100],
        …,
        [ 4598,   407,  1011,  …,  -100,  -100,  -100],
        [  568,  5633,  -100,  …,  -100,  -100,  -100],
        [  533,   345,   991,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,    72,  …, 10919, 10919, 10919],
        [  318,  1312,   716,  …,    72,    72,    72],
        [  318,   764,   764,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,   503,   262,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[39563,   764,  -100,  …,  -100,  -100,  -100],
        [ 1219,   837,  1312,  …,  -100,  -100,  -100],
        [  732,   826,   683,  …,  -100,  -100,  -100],
        …,
        [ 1136,   683,  4291,  …,  -100,  -100,  -100],
        [28060,  5145,  -100,  …,  -100,  -100,  -100],
        [ 2093,   683,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 5633,  644,  …,   72,   72,   72],
        [ 318,  764,  345,  …,   72,   72,   72],
        [ 318,  389,  764,  …,   72,   72,   72],
        …,
        [ 318, 3656, 1312,  …,   72,   72,   72],
        [ 318,  407,  339,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   71,  7456,  5633,  …,  -100,  -100,  -100],
        [41073,   880,   764,  …,  -100,  -100,  -100],
        [43669,   356,  1138,  …,  -100,  -100,  -100],
        …,
        [  273,   616,   764,  …,  -100,  -100,  -100],
        [22850,   857,   407,  …,  -100,  -100,  -100],
        [ 1662,   523,  5210,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5490,   510,  …,    72,    72,    72],
        [  318,   764,   257,  …,    72,    72,    72],
        [  318,  1312,    72,  …,    72,    72,    72],
        …,
        [  318,   345,  5633,  …, 10919, 10919, 10919],
        [  318,   764,  3872,  …,    72,    72,    72],
        [  318,   837,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 4598,   407,  9580,  …,  -100,  -100,  -100],
        [ 3919,  1808,   422,  …,  -100,  -100,  -100],
        [ 1219,   764,  -100,  …,  -100,  -100,  -100],
        …,
        [ 8727,   389,   345,  …,  -100,  -100,  -100],
        [ 1169, 25291,   262,  …,  -100,  -100,  -100],
        [ 5832,   760,   644,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  318,  …,   72,   72,   72],
        [ 318,  284, 4144,  …,   72,   72,   72],
        [ 318,  407, 2392,  …,   72,   72,   72],
        …,
```

```
        [ 318,  764,  837,  …,   72,   72,   72],
        [ 318, 1312,  837,  …,   72,   72,   72],
        [ 318,  644, 5633,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4053,   837,   644,  …,  -100,  -100,  -100],
        [ 5832,   765,   257,  …,  -100,  -100,  -100],
        [  270,   318,   645,  …,  -100,  -100,  -100],
        …,
        [   11,  3491,  1359,  …,  -100,  -100,  -100],
        [ 4053,   837, 14497,  …,  -100,  -100,  -100],
        [  271,   326,  2081,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   764,   319,  …,    72,    72,    72],
        [  318,   716,  1016,  …,    72,    72,    72],
        [  318,   644,    72,  …,    72,    72,    72],
        …,
        [  318,   764,   764,  …,    72,    72,    72],
        [  318,   257,  5633,  …,    72,    72,    72],
        [  318,   587,   764,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[18927,  1263,  1016,  …,  -100,  -100,  -100],
        [ 2197,  1312,   716,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [ 6495,   502,  1568,  …,  -100,  -100,  -100],
        [ 5832,  1972,  2726,  …,  -100,  -100,  -100],
        [27238,   423,   340,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  326, 1804,  …,   72,   72,   72],
        [ 318,  345,  588,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        …,
        [ 318,  764,  617,  …,   72,   72,   72],
        [ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  837,  644,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[10919,   318,   673,  …,  -100,  -100,  -100],
        [10919,   561,   345,  …,  -100,  -100,  -100],
        [   72,   750,   407,  …,  -100,  -100,  -100],
        …,
        [27238,   307,   286,  …,  -100,  -100,  -100],
        [  732,   750,   407,  …,  -100,  -100,  -100],
        [33331,   502,   837,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,  837,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  345,  284,  …,   72,   72,   72],
        …,
```

```
        [ 318,  837,  389,  …,   72,   72,   72],
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[29810,   764, 20875,  …,  -100,  -100,  -100],
        [ 5303,   764,  6072,  …,  -100,  -100,  -100],
        [   13,   561,   588,  …,  -100,  -100,  -100],
        …,
        [11722, 13337,   810,  …,  -100,  -100,  -100],
        [ 8505,   837,   475,  …,  -100,  -100,  -100],
        [39239,   306,   703,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  284,  307,  …,   72,   72,   72],
        [ 318, 1392,  534,  …,   72,   72,   72],
        [ 318, 1279,   72,  …,   72,   72,   72],
        …,
        [ 318,  502, 1254,  …,   72,   72,   72],
        [ 318,  644,   72,  …,   72,   72,   72],
        [ 318, 1312,   72,  …,  986,  986,  986]], device='cuda:0')
First 10 labels: tensor([[  732,  1016,   284,  …,  -100,  -100,  -100],
        [ 5832,   423,  2626,  …,  -100,  -100,  -100],
        [24561,   764,  -100,  …,  -100,  -100,  -100],
        …,
        [  270,   925,   502,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100],
        [43669,   764,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,  389,  …,   72,   72,   72],
        [ 318, 1312,  514,  …,   72,   72,   72],
        [ 318,  284, 2739,  …,   72,   72,   72],
        …,
        [ 318,  389,  764,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  644,   72,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  392,   611,   345,  …,  -100,  -100,  -100],
        [   72,  2644,  1309,  …,  -100,  -100,  -100],
        [   11,   835,  1165,  …,  -100,  -100,  -100],
        …,
        [39664,   345,   765,  …,  -100,  -100,  -100],
        [ 1326,  1165,   764,  …,  -100,  -100,  -100],
        [10919,  5633,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1312,  1312,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        …,
```

```
        [  318,   407,   356,  …,    72,    72,    72],
        [  318,   284,   760,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[43669,   837,  2644,  …,  -100,  -100,  -100],
        [   72,  1833,   764,  …,  -100,  -100,  -100],
        [ 1820,  2636,   764,  …,  -100,  -100,  -100],
        …,
        [  732,   373,   764,  …,  -100,  -100,  -100],
        [   72,   765,   284,  …,  -100,  -100,  -100],
        [10919,   318,   326,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,   72,  …,   72,   72,   72],
        [ 318,  262,  764,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        …,
        [ 318,  389,  407,  …, 5832, 5832, 5832],
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318, 1312,  345,  …,  986,  986,  986]], device='cuda:0')
First 10 labels: tensor([[ 8505,  5633,  -100,  …,  -100,  -100,  -100],
        [23108,   287,  6504,  …,  -100,  -100,  -100],
        [  986,   475,  1312,  …,  -100,  -100,  -100],
        …,
        [13893,   345,   561,  …,  -100,  -100,  -100],
        [42832,   764,  3993,  …,  -100,  -100,  -100],
        [  986,   475,   788,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407, 1468,  …,   72,   72,   72],
        [ 318, 5633,  922,  …,   72,   72,   72],
        [ 318,  407, 1560,  …,   72,   72,   72],
        …,
        [ 318,  257,  764,  …,   72,   72,   72],
        [ 318,  837,  389,  …,   72,   72,   72],
        [ 318, 1312,  530,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   389,   281,  …,  -100,  -100,  -100],
        [  271,   340,   257,  …,  -100,  -100,  -100],
        [29688,   466,   407,  …,  -100,  -100,  -100],
        …,
        [  258,   373,   407,  …,  -100,  -100,  -100],
        [ 1462,   502,   345,  …,  -100,  -100,  -100],
        [ 8505,  5145,   262,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   466,   994,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   764,   764,  …,    11,    11,    11],
        …,
```

```
        [  318,   407,   284,  …,    72,    72,    72],
        [  318,   764,  1312,  …, 10919, 10919, 10919],
        [  318,   804,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   460,  2652,  …,  -100,  -100,  -100],
        [18223,   983,   764,  …,  -100,  -100,  -100],
        [12359,  1105,  4524,  …,  -100,  -100,  -100],
        …,
        [ 5832,   481,   423,  …,  -100,  -100,  -100],
        [ 3919,  6617,  5633,  …,  -100,  -100,  -100],
        [20657,   257,  1271,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  5633,   644,  …, 10919, 10919, 10919],
        [  318,  1312,   837,  …,    72,    72,    72],
        [  318,   257,   673,  …,    72,    72,    72],
        …,
        [  318,  1312,   764,  …,    72,    72,    72],
        [  318,   345,  5633,  …,    72,    72,    72],
        [  318,   481,   407,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[  71, 7456, 5633,  …, -100, -100, -100],
        [9464,  837, 1364,  …, -100, -100, -100],
        [7091,  373,  618,  …, -100, -100, -100],
        …,
        [8117,  837,  766,  …, -100, -100, -100],
        [3003,  389,  345,  …, -100, -100, -100],
        [8524, 1312,  466,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   326,  5633,  …,    72,    72,    72],
        [  318,   318,   345,  …,    72,    72,    72],
        [  318,   326,  5633,  …, 10919, 10919, 10919],
        …,
        [  318,   644,   644,  …,    72,    72,    72],
        [  318,   262,  2119,  …,    72,    72,    72],
        [  318,   257,  1223,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   318,  2642,  …,  -100,  -100,  -100],
        [    0,   644,   389,  …,  -100,  -100,  -100],
        [10919,   318,   428,  …,  -100,  -100,  -100],
        …,
        [33331,   514,   837,  …,  -100,  -100,  -100],
        [ 2188,   284,   534,  …,  -100,  -100,  -100],
        [ 8117,   318,  4753,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  826,  764,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        [ 318,  466,  764,  …,   72,   72,   72],
        …,
```

```
        [ 318,  466, 7787,  …,   72,   72,   72],
        [ 318,  345, 1612,  …,   72,   72,   72],
        [ 318,  644,  262,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5562,   318,   477,  …,  -100,  -100,  -100],
        [ 5832,   760,   837,  …,  -100,  -100,  -100],
        [  732,  2314,  2686,  …,  -100,  -100,  -100],
        …,
        [10919,   345,   523,  …,  -100,  -100,  -100],
        [10919,   466,   345,  …,  -100,  -100,  -100],
        [ 1659,  2644,   572,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   922,  2576,  …,    72,    72,    72],
        [  318,   262,  5633,  …,    72,    72,    72],
        [  318,   407,   764,  …,    72,    72,    72],
        …,
        [  318,  1279,   764,  …,    72,    72,    72],
        [  318,  1312,  1312,  …, 10919, 10919, 10919],
        [  318,   764,    72,  …,    11,    11,    11]], device='cuda:0')
First 10 labels: tensor([[ 7091,   257,  1103,  …,  -100,  -100,  -100],
        [ 3003,   318,  1029,  …,  -100,  -100,  -100],
        [   72,   716,  2726,  …,  -100,  -100,  -100],
        …,
        [14261,  5633,  1263,  …,  -100,  -100,  -100],
        [43669,  5633,   290,  …,  -100,  -100,  -100],
        [  986,   764,  -100,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  764,  764,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318,  307,  345,  …,   72,   72,   72],
        …,
        [ 318,  826,  764,  …,   72,   72,   72],
        [ 318,  407,  345,  …,   72,   72,   72],
        [ 318,  307,  257,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 2958,  1363,   837,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100],
        [   72,   284, 10110,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,   826,  …,  -100,  -100,  -100],
        [   72,   481,   869,  …,  -100,  -100,  -100],
        [ 5562,   561,   307,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,  1975,  1312,  …,  5832,  5832,  5832],
        [  318,  1279,    72,  …,    72,    72,    72],
        [  318,   345,   307,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,   326,  5633,  …,    72,    72,    72],
        [  318,   257,   257,  …,    72,    72,    72],
        [  318,   466,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[   72,  2314,   764,  …,  -100,  -100,  -100],
        [ 1326,  5633,  -100,  …,  -100,  -100,  -100],
        [19188,   407, 16360,  …,  -100,  -100,  -100],
        …,
        [10919,   318,   340,  …,  -100,  -100,  -100],
        [ 8117,   318,  3360,  …,  -100,  -100,  -100],
        [  986,  1521,   750,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  466,  345,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
        [ 318, 1654, 5633,  …,   72,   72,   72],
        [ 318,  502,  837,  …,   72,   72,   72],
        [ 318, 1683, 1775,  …, 5832, 5832, 5832]], device='cuda:0')
First 10 labels: tensor([[ 8524,  1521,   750,  …,  -100,  -100,  -100],
        [43669,   764,  1312,  …,  -100,  -100,  -100],
        [11338,   340,   837,  …,  -100,  -100,  -100],
        …,
        [  533,   345,  1654,  …,  -100,  -100,  -100],
        [41194,  1904,   502,  …,  -100,  -100,  -100],
        [14150,   345,  1683,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  284,  467,  …,   72,   72,   72],
        [ 318,  502,  764,  …,   72,   72,   72],
        [ 318,  284,  561,  …,   72,   72,   72],
        …,
        [ 318,  644,   72,  …,   72,   72,   72],
        [ 318,  345, 5633,  …,   72,   72,   72],
        [ 318,  764,  881,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 5832,   765,   284,  …,  -100,  -100,  -100],
        [ 5460,   379,   262,  …,  -100,  -100,  -100],
        [   72,  3066,  1312,  …,  -100,  -100,  -100],
        …,
        [  568,  5633,  -100,  …,  -100,  -100,  -100],
        [ 8727,   389,   777,  …,  -100,  -100,  -100],
        [40716,   345,   845,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318, 1312,  345,  …,   72,   72,   72],
        [ 318,  345,  466,  …,   72,   72,   72],
        …,
```

```
        [ 318,  389,  307,  …,   72,   72,   72],
        [ 318, 5633,   72,  …,   72,   72,   72],
        [ 318,  318,  262,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   466,   407,  …,  -100,  -100,  -100],
        [ 4053,   837,  1297,  …,  -100,  -100,  -100],
        [22850,   750,   345,  …,  -100,  -100,  -100],
        …,
        [ 4360,   345,   284,  …,  -100,  -100,  -100],
        [   13,  5633,  -100,  …,  -100,  -100,  -100],
        [  361,   428,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  502,  …,   72,   72,   72],
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318,  307, 1545,  …,  986,  986,  986],
        …,
        [ 318,  584, 1352,  …,   72,   72,   72],
        [ 318,  287, 4419,  …,   72,   72,   72],
        [ 318,  407,  922,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 9930,   389,  4585,  …,  -100,  -100,  -100],
        [   72,   716,   407,  …,  -100,  -100,  -100],
        [  986,   284,   257,  …,  -100,  -100,  -100],
        …,
        [  261,   262, 21483,  …,  -100,  -100,  -100],
        [ 3919,  4419,   645,  …,  -100,  -100,  -100],
        [  270,   318,   257,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   407,   764,  …,    72,    72,    72],
        [  318,  1312,   319,  …,    72,    72,    72],
        [  318,  1312,   764,  …,    72,    72,    72],
        …,
        [  318,  1312,   837,  …,    72,    72,    72],
        [  318,   764,   345,  …,    72,    72,    72],
        [  318,   764,   764,  …, 10919, 10919, 10919]], device='cuda:0')
First 10 labels: tensor([[   72,   716,  7926,  …,  -100,  -100,  -100],
        [ 9866,   837,  3867,  …,  -100,  -100,  -100],
        [ 1219,   837, 23139,  …,  -100,  -100,  -100],
        …,
        [31373,  2644, 11752,  …,  -100,  -100,  -100],
        [14774,  8458,   284,  …,  -100,  -100,  -100],
        [ 5303,   837,  2205,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  284,  262,  …,   72,   72,   72],
        [ 318, 5633,   72,  …,   11,   11,   11],
        [ 318,  389,  307,  …,   72,   72,   72],
        …,
```

```
        [ 318,  329,  651,  …,   72,   72,   72],
        [ 318,  764,  339,  …,   72,   72,   72],
        [ 318,  716,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  258,  1816,   284,  …,  -100,  -100,  -100],
        [   13,  5633,  -100,  …,  -100,  -100,  -100],
        [  732,    68,   481,  …,  -100,  -100,  -100],
        …,
        [ 1136,  3492,   284,  …,  -100,  -100,  -100],
        [11338,   683,   837,  …,  -100,  -100,  -100],
        [ 1326,  1312,   587,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   837,  1312,  …, 10919, 10919, 10919],
        [  318,   345,  1781,  …,    72,    72,    72],
        [  318,   407,   760,  …,    72,    72,    72],
        …,
        [  318,   389,  1016,  …,    72,    72,    72],
        [  318,   284,   284,  …,    72,    72,    72],
        [  318,   826,   764,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[1640,  502,  837,  …, -100, -100, -100],
        [ 568,  837,  286,  …, -100, -100, -100],
        [  72,  466,  407,  …, -100, -100, -100],
        …,
        [   0,  356,  389,  …, -100, -100, -100],
        [5832,  765,  502,  …, -100, -100, -100],
        [5562,  318,  477,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  760,  …,   72,   72,   72],
        [ 318,  407,  307,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        …,
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  837,  356,  …,   72,   72,   72],
        [ 318,  826,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   72,   466,   407,  …,  -100,  -100,  -100],
        [   72,   481,   407,  …,  -100,  -100,  -100],
        [ 1219,   837,  1312,  …,  -100,  -100,  -100],
        …,
        [ 8505,   837,  1312,  …,  -100,  -100,  -100],
        [20342,   837,   815,  …,  -100,  -100,  -100],
        [ 5562,   318,   826,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   345,   466,  …,    72,    72,    72],
        [  318,   764,  1312,  …, 10919, 10919, 10919],
        [  318,  5633,    72,  …, 10919, 10919, 10919],
        …,
```

```
        [  318,   326,  2642,  …,    72,    72,    72],
        [  318,   837,   922,  …,    72,    72,    72],
        [  318,   644,   644,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   750,   345,  …,  -100,  -100,  -100],
        [ 6649,   315,   764,  …,  -100,  -100,  -100],
        [42832,  5633,  -100,  …,  -100,  -100,  -100],
        …,
        [   11,   318,  1223,  …,  -100,  -100,  -100],
        [  568,  1290,   523,  …,  -100,  -100,  -100],
        [10919,  5633,   329,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  326,  …,   72,   72,   72],
        [ 318,  837,  345,  …,   72,   72,   72],
        [ 318,  502,  284,  …,   72,   72,   72],
        …,
        [ 318,  407,  563,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72],
        [ 318,  318,  257,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[    0,   644,   318,  …,  -100,  -100,  -100],
        [ 2958,   319,   837,  …,  -100,  -100,  -100],
        [ 7091,  3181,   340,  …,  -100,  -100,  -100],
        …,
        [   72,   716,  3253,  …,  -100,  -100,  -100],
        [19532,   837,  1654,  …,  -100,  -100,  -100],
        [  392,   612,   318,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  407,  625,  …,   72,   72,   72],
        [ 318,  262,  290,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        …,
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  326, 5633,  …,   72,   72,   72],
        [ 318,  407,  345,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  270,   318,   983,  …,  -100,  -100,  -100],
        [29753,   287,   837,  …,  -100,  -100,  -100],
        [  986,  3996,   764,  …,  -100,  -100,  -100],
        …,
        [   72,   760,   780,  …,  -100,  -100,  -100],
        [10919,   318,   428,  …,  -100,  -100,  -100],
        [22850,   466,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  389,  407,  …,   72,   72,   72],
        [ 318,  764, 1312,  …,   72,   72,   72],
        [ 318,  345,  287,  …,   72,   72,   72],
        …,
```

```
        [ 318,  826,  764,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318,  407, 2300,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  392,   345,   389,  …,  -100,  -100,  -100],
        [   11, 10194,   837,  …,  -100,  -100,  -100],
        [ 8727,  1234,   345,  …,  -100,  -100,  -100],
        …,
        [ 5562,   318,  1049,  …,  -100,  -100,  -100],
        [ 8505,   764,  1312,  …,  -100,  -100,  -100],
        [  270,   857,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   644,    72,  …, 10919, 10919, 10919],
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,  1312,   502,  …,    72,    72,    72],
        …,
        [  318,  1312,    72,  …,    72,    72,    72],
        [  318,   318,   307,  …, 10919, 10919, 10919],
        [  318,   826,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,  5633,  -100,  …,  -100,  -100,  -100],
        [  270,   318,  3608,  …,  -100,  -100,  -100],
        [ 5557,   837,  2277,  …,  -100,  -100,  -100],
        …,
        [ 8505,   764,  -100,  …,  -100,  -100,  -100],
        [13893,   340,   284,  …,  -100,  -100,  -100],
        [ 5562,   318,   810,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318, 1312,   72,  …,   72,   72,   72],
        [ 318, 5633, 1279,  …,   72,   72,   72],
        [ 318,  439,  262,  …,   72,   72,   72],
        …,
        [ 318,  389,  319,  …,   72,   72,   72],
        [ 318,  835,  764,  …,   72,   72,   72],
        [ 318,  389,  284,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[42994,   764,  -100,  …,  -100,  -100,  -100],
        [ 3846,   954,  5633,  …,  -100,  -100,  -100],
        [20839, 22397,   779,  …,  -100,  -100,  -100],
        …,
        [ 1106,   356,  9581,  …,  -100,  -100,  -100],
        [  261,   616,  9845,  …,  -100,  -100,  -100],
        [  568,   345,   423,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   826,   644,  …, 22850, 22850, 22850],
        [  318,   407,   764,  …,    72,    72,    72],
        [  318,   588,   345,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,  1312,  …,    72,    72,    72],
        [  318,   262,   764,  …,    72,    72,    72],
        [  318,   318,   345,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[5562,  318,  407,  …, -100, -100, -100],
        [  72,  716, 3492,  …, -100, -100, -100],
        [  72,  561, 1494,  …, -100, -100, -100],
        …,
        [  72, 1833,  764,  …, -100, -100, -100],
        [9930, 1718,  606,  …, -100, -100, -100],
        [ 568,  508, 2921,  …, -100, -100, -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  262,  286,  …,   72,   72,   72],
        [ 318,  502,  837,  …,   72,   72,   72],
        [ 318,  837, 1312,  …,   72,   72,   72],
        …,
        [ 318,  307,  307,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        [ 318, 5633, 1611,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  270,   351,   617,  …,  -100,  -100,  -100],
        [41194,  1904,   502,  …,  -100,  -100,  -100],
        [ 1106,   645, 19424,  …,  -100,  -100,  -100],
        …,
        [ 5832,  1276,  1107,  …,  -100,  -100,  -100],
        [ 1136,   262,  1097,  …,  -100,  -100,  -100],
        [  271,   339,   617,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  467,  764,  …,   72,   72,   72],
        [ 318,  318,  345,  …,   72,   72,   72],
        [ 318,  286,  994,  …,   72,   72,   72],
        …,
        [ 318,  307,  587,  …,   72,   72,   72],
        [ 318,  345, 1312,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 1616,   340,   307,  …,  -100,  -100,  -100],
        [    0,   644,   389,  …,  -100,  -100,  -100],
        [45380,   503,   286,  …,  -100,  -100,  -100],
        …,
        [  270,   714,   423,  …,  -100,  -100,  -100],
        [ 4360,  5633,   986,  …,  -100,  -100,  -100],
        [  505,   288,    21,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[  318,   326,  1804,  …,    72,    72,    72],
        [  318,  1312,   502,  …,    72,    72,    72],
        [  318,   502,   837,  …,    72,    72,    72],
        …,
```

```
        [  318,   764,  1312,  …, 10919, 10919, 10919],
        [  318,  1392,   284,  …,    72,    72,    72],
        [  318,   764,  1312,  …,    72,    72,    72]], device='cuda:0')
First 10 labels: tensor([[10919,   373,   339,  …,  -100,  -100,  -100],
        [ 1219,   837, 20927,  …,  -100,  -100,  -100],
        [ 6667, 12311,   502,  …,  -100,  -100,  -100],
        …,
        [  265,  1755,  5633,  …,  -100,  -100,  -100],
        [ 5832,   423,  1223,  …,  -100,  -100,  -100],
        [  392,  2818,   764,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  345,  345,  …,   72,   72,   72],
        [ 318, 1312,  345,  …,   72,   72,   72],
        [ 318,  716,  407,  …,   72,   72,   72],
        …,
        [ 318,  257,  284,  …,   72,   72,   72],
        [ 318, 1312,  502,  …,   72,   72,   72],
        [ 318,  510,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4360,   466,   407,  …,  -100,  -100,  -100],
        [ 4053,   837,   611,  …,  -100,  -100,  -100],
        [   11,  1312,   716,  …,  -100,  -100,  -100],
        …,
        [ 5661,   318,  1016,  …,  -100,  -100,  -100],
        [11085,   640,   329,  …,  -100,  -100,  -100],
        [  258,  9763,   866,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  318,  407,  …,   72,   72,   72],
        [ 318, 1392,  467,  …,   72,   72,   72],
        [ 318,  340,  287,  …,   72,   72,   72],
        …,
        [ 318, 3420,  764,  …,   72,   72,   72],
        [ 318, 1312, 1312,  …,   72,   72,   72],
        [ 318,  407,  257,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[   13,   673,   318,  …,  -100,  -100,  -100],
        [   72,   423,   284,  …,  -100,  -100,  -100],
        [   11,  1234,  3511,  …,  -100,  -100,  -100],
        …,
        [48619,   262,  4074,  …,  -100,  -100,  -100],
        [ 5460,   837,   764,  …,  -100,  -100,  -100],
        [  270,   318,   407,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([32, 512]), Labels shape: torch.Size([32, 512])
First 10 predictions: tensor([[ 318,  826, 5633,  …,   72,   72,   72],
        [ 318,  284,  284,  …,   72,   72,   72],
        [ 318,  466,  284,  …,   72,   72,   72],
        …,
```

```
        [ 318, 5633, 8505,  …,   72,   72,   72],
        [ 318, 1312,  716,  …,   72,   72,   72],
        [ 318, 1312,  764,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[  271,   477,  5836,  …,  -100,  -100,  -100],
        [   72,   761,   345,  …,  -100,  -100,  -100],
        [ 4919,   345,  1410,  …,  -100,  -100,  -100],
        …,
        [30526,  5633,  -100,  …,  -100,  -100,  -100],
        [10919,   611,  1312,  …,  -100,  -100,  -100],
        [43669,  2644,  3734,  …,  -100,  -100,  -100]], device='cuda:0')
Step 3600:
Predictions shape: torch.Size([28, 512]), Labels shape: torch.Size([28, 512])
First 10 predictions: tensor([[ 318,  764,  318,  …,   72,   72,   72],
        [ 318,  407,  587,  …,   72,   72,   72],
        [ 318,  764,  764,  …,   72,   72,   72],
        …,
        [ 318,  407, 1016,  …,   72,   72,   72],
        [ 318, 5633,  716,  …,   72,   72,   72],
        [ 318,  262,  539,  …,   72,   72,   72]], device='cuda:0')
First 10 labels: tensor([[ 4360,   262,  3348,  …,  -100,  -100,  -100],
        [  270,   318,  1541,  …,  -100,  -100,  -100],
        [   11,  1165,  2089,  …,  -100,  -100,  -100],
        …,
        [ 5832,   389,   407,  …,  -100,  -100,  -100],
        [15542,  5633,  1312,  …,  -100,  -100,  -100],
        [24219,   706,   474,  …,  -100,  -100,  -100]], device='cuda:0')
Validation Loss: 2.9882
All labels shape: 33692, All predictions shape: 33692
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-78-9885475ced2e> in <cell line: 2>()
      1 # Train the model for remaining epochs
----> 2 train_model(model, train_dataloader, val_dataloader, optimizer,␣
 ↪epochs=3, save_every_n_steps=100, start_epoch=2, start_step=0)

<ipython-input-76-922400b98f39> in train_model(model, train_dataloader,␣
 ↪val_dataloader, optimizer, epochs, save_every_n_steps, start_epoch, start_ste )
    126
    127         # Validation after each epoch using validation data
--> 128         validate_model(model, val_dataloader, tokenizer)
    129
    130         model_save_name = 'fine_tuned_dialoGPT{epoch+1}_final.pt'

<ipython-input-47-5feab33cb83d> in validate_model(model, val_loader, tokenizer)
     62     try:
     63       # Convert lists to tensors
```

```
---> 64         labels = torch.tensor(all_labels)
     65         predictions = torch.tensor(all_preds)
     66         print(f"Labels shape: {labels.shape}, Predictions shape:␣
 ↪{predictions.shape}")

ValueError: too many dimensions 'str'
```

14 Invoke all the Validate Model Method

There were errors after the training was complete but when validate_model() was invoked.

So, Loading the trained models checkpoints and invoking validate_model() method.

15 Load the Final Check point model stored after Epoch 1

[ ]:
```
checkpoint = torch.load('/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/
 ↪fine_tuned_dialoGPT_epoch1_step3600.pt', map_location=torch.device(device))
model.load_state_dict(checkpoint['model_state_dict'])
optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
scheduler.load_state_dict(checkpoint['scheduler_state_dict'])
start_epoch = checkpoint['epoch']  # Start from the next epoch
start_step = checkpoint['step']
print(f"Loaded checkpoint from epoch {start_epoch} and step {start_step}")
```

```
<ipython-input-35-4b1b4494bc37>:1: FutureWarning: You are using `torch.load`
with `weights_only=False` (the current default value), which uses the default
pickle module implicitly. It is possible to construct malicious pickle data
which will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for
more details). In a future release, the default value for `weights_only` will be
flipped to `True`. This limits the functions that could be executed during
unpickling. Arbitrary objects will no longer be allowed to be loaded via this
mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control of the
loaded file. Please open an issue on GitHub for any issues related to this
experimental feature.
  checkpoint = torch.load('/content/drive/MyDrive/Colab
Notebooks/ChatFlixModels/fine_tuned_dialoGPT_epoch1_step3600.pt',
map_location=torch.device(device))

Loaded checkpoint from epoch 1 and step 3600
```

[ ]:
```
# Validation after each epoch using validation data
validate_model(model, val_dataloader, tokenizer)
```

```
Validation Loss: 3.1828
All labels shape: 33692, All predictions shape: 33692
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels
with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

Accuracy: 0.0003
Precision: 0.0004
Recall: 0.0003
F1-Score: 0.0003

/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)

BLEU Score: 0.0022
ROUGE Scores: {'rouge-1': {'r': 0.21568273777184954, 'p': 0.23684904593101888,
'f': 0.22416692011604597}, 'rouge-2': {'r': 0.035984681811857544, 'p':
0.03612941855556099, 'f': 0.03588286511975884}, 'rouge-l': {'r':
0.2057208032520116, 'p': 0.2259841885755017, 'f': 0.21384349199347596}}
```

16 Load the Final Check point model stored after Epoch 2

[ ]:
```
checkpoint = torch.load('/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/
  ↪fine_tuned_dialoGPT_epoch2_step3600.pt', map_location=torch.device(device))
model.load_state_dict(checkpoint['model_state_dict'])
optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
scheduler.load_state_dict(checkpoint['scheduler_state_dict'])
start_epoch = checkpoint['epoch']  # Start from the next epoch
start_step = checkpoint['step']
print(f"Loaded checkpoint from epoch {start_epoch} and step {start_step}")
```

```
<ipython-input-29-71ec73ecdf05>:1: FutureWarning: You are using `torch.load`
with `weights_only=False` (the current default value), which uses the default
pickle module implicitly. It is possible to construct malicious pickle data
which will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for
more details). In a future release, the default value for `weights_only` will be
flipped to `True`. This limits the functions that could be executed during
unpickling. Arbitrary objects will no longer be allowed to be loaded via this
mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control of the
loaded file. Please open an issue on GitHub for any issues related to this
experimental feature.
  checkpoint = torch.load('/content/drive/MyDrive/Colab
Notebooks/ChatFlixModels/fine_tuned_dialoGPT_epoch2_step3600.pt',
map_location=torch.device(device))

Loaded checkpoint from epoch 2 and step 3600
```

[ ]:
```
# Validation after each epoch using validation data
validate_model(model, val_dataloader, tokenizer)
```

```
Validation Loss: 3.0357
All labels shape: 33692, All predictions shape: 33692
```

[ ]:
```

```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels
with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

Accuracy: 0.0001
Precision: 0.0002
```

```
Recall: 0.0001
F1-Score: 0.0001
```

[ ]:

```
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
```

```
BLEU Score: 0.0025
ROUGE Scores: {'rouge-1': {'r': 0.22056681247279245, 'p': 0.24217274520621157,
'f': 0.22916424137472002}, 'rouge-2': {'r': 0.03876578533257116, 'p':
0.03906942417641923, 'f': 0.03872543260017414}, 'rouge-l': {'r':
0.2105528441593231, 'p': 0.23122352765555207, 'f': 0.21877424695655226}}
```

17 Load the Final Check point model stored after Epoch 3

[ ]:

```
checkpoint = torch.load('/content/drive/MyDrive/Colab Notebooks/ChatFlixModels/
  ↪fine_tuned_dialoGPT_epoch3_step3600.pt', map_location=torch.device(device))
model.load_state_dict(checkpoint['model_state_dict'])
optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
scheduler.load_state_dict(checkpoint['scheduler_state_dict'])
start_epoch = checkpoint['epoch']  # Start from the next epoch
start_step = checkpoint['step']
print(f"Loaded checkpoint from epoch {start_epoch} and step {start_step}")
```

```
<ipython-input-38-af7411e535f9>:1: FutureWarning: You are using `torch.load`
with `weights_only=False` (the current default value), which uses the default
pickle module implicitly. It is possible to construct malicious pickle data
which will execute arbitrary code during unpickling (See
```

```
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for
more details). In a future release, the default value for `weights_only` will be
flipped to `True`. This limits the functions that could be executed during
unpickling. Arbitrary objects will no longer be allowed to be loaded via this
mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control of the
loaded file. Please open an issue on GitHub for any issues related to this
experimental feature.
  checkpoint = torch.load('/content/drive/MyDrive/Colab
Notebooks/ChatFlixModels/fine_tuned_dialoGPT_epoch3_step3600.pt',
map_location=torch.device(device))

Loaded checkpoint from epoch 3 and step 3600
```

[ ]:
```
# Validation after each epoch using validation data
validate_model(model, val_dataloader, tokenizer)
```

```
Validation Loss: 2.9888
All labels shape: 33692, All predictions shape: 33692

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels
with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

Accuracy: 0.0001
Precision: 0.0001
Recall: 0.0001
F1-Score: 0.0001
Precision: 1.0000, Recall: 0.0000, F1-Score: 0.0000

/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
```

```
  warnings.warn(_msg)
/usr/local/lib/python3.10/dist-packages/nltk/translate/bleu_score.py:552:
UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
```

```
BLEU Score: 0.0026
ROUGE Scores: {'rouge-1': {'r': 0.22213744575248304, 'p': 0.24232078037700508,
'f': 0.23012233936889143}, 'rouge-2': {'r': 0.03991810325956266, 'p':
0.040058987839428545, 'f': 0.03979253888396658}, 'rouge-l': {'r':
0.21220979668777634, 'p': 0.231532560480215, 'f': 0.21984847815055616}}
```

18 Inference:

Build a chatbot interface where the model generates responses based on user inputs and conversation history.

Reference ChatInterface.ipynb file to run the chat bot