Alexnet_Verification_with_Imgnet_drop_momentum_training

April 19, 2024

[1]:

```
import numpy as np
from functools import partial
from typing import Any, Optional

import os
import cv2
import time

import pandas as pd
import torch.nn.init as init

import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
from PIL import Image
from torchvision import transforms
# from torch.transforms._presets import ImageClassification
# from torch.utils import _log_api_usage_once
# from ._api import register_model, Weights, WeightsEnum
# from ._meta import _IMAGENET_CATEGORIES
# from ._utils import _ovewrite_named_param, handle_legacy_interface

import matplotlib.pyplot as plt


model_alex_given = torch.hub.load('pytorch/vision:v0.10.0', 'alexnet',␣
  ↪pretrained=True)
model_alex_given.eval()
# Device configuration
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

```
Using cache found in C:\Users\Limit/.cache\torch\hub\pytorch_vision_v0.10.0
C:\Apps installed by Lim\anaconda3\Lib\site-
packages\torchvision\models\_utils.py:208: UserWarning: The parameter
'pretrained' is deprecated since 0.13 and may be removed in the future, please
use 'weights' instead.
  warnings.warn(
C:\Apps installed by Lim\anaconda3\Lib\site-
packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a
weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed
in the future. The current behavior is equivalent to passing
`weights=AlexNet_Weights.IMAGENET1K_V1`. You can also use
`weights=AlexNet_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
```

[2]:

```python
class AlexNet(nn.Module):
    def __init__(self, num_classes: int = 1000, dropout: float = 0.5) -> None:
        super().__init__()
#         _log_api_usage_once(self)
        self.features = nn.Sequential(
            nn.Conv2d(3, 96, kernel_size=11, stride=4, padding=2),
            nn.BatchNorm2d(96),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(96, 256, kernel_size=5, padding=2),
            nn.BatchNorm2d(256),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(256, 384, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(384, 384, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(384, 256, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
        )
        self.avgpool = nn.AdaptiveAvgPool2d((6, 6))
        self.classifier = nn.Sequential(
            nn.Dropout(p=dropout),
            nn.Linear(256 * 6 * 6, 4096),
            nn.ReLU(inplace=True),
            nn.Dropout(p=dropout),
            nn.Linear(4096, 4096),
            nn.ReLU(inplace=True),
            nn.Linear(4096, num_classes),
        )

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        x = self.features(x)
        x = self.avgpool(x)
        x = torch.flatten(x, 1)
        x = self.classifier(x)
        return x
    def initialize_weights(self):
        for m in self.modules():
            if isinstance(m, nn.Conv2d) or isinstance(m, nn.Linear):
                # Initialize weights for convolutional and linear layers
                init.xavier_uniform_(m.weight)
                if m.bias is not None:
                    # Initialize biases if they exist
                    init.constant_(m.bias, 0)
```

[3]:

```
class ConvertToRGB(object):
    def __call__(self, img):
        if img.shape[0] == 1:  # Check if the image has only one channel
            img = torch.stack([img[0]] * 3, dim=0)  # Convert single channel to␣
 ↪RGB
        return img


preprocess = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    ConvertToRGB(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
    ])
```

[4]:

```
image_path = "/Users/Limit/imagenet-object-localization-challenge_100"

filenames_image_path = []
label_train = []
root_image = []
counter = 0
current_label = 0
for root, _, filenames in os.walk(image_path):
    current_root = root
    for i in filenames:
#         print(i)
        counter += 1
        if ((counter) %1300 == 0):
            current_label += 1
            print(counter)
        label_train.append(current_label)


#         print(current_root)
#         print(i)
        temp = current_root + "\\" + i
#         print(temp)
```

```python
        filenames_image_path.append(temp)
true_label = 0
correct_labels = 0
start_time = time.time()

counter_1=0
x_train = []
for i in range(26000):

#     print(i)
#     print(counter)
    image_name = filenames_image_path[i]
    # black and white
#     image_name = "/Users/Limit/imagenet-object-localization-challenge/n01440764/n01440764_15560.JPEG"
    input_image = Image.open(image_name)



    input_tensor = preprocess(input_image)
#     input_batch = input_tensor.unsqueeze(0) # create a mini-batch as expected by the model
    input_batch = input_tensor
    # move the input and model to GPU for speed if available
    if torch.cuda.is_available():
        input_batch = input_batch.to('cuda')
        model_alex_given.to('cuda')
    x_train.append(input_batch)
    counter_1 += 1
    if ((counter_1+1) %1300 == 0):
            counter_1 += 1
            print(counter_1)
```

```
1300
2600
3900
5200
6500
7800
9100
10400
11700
13000
14300
15600
16900
```

18200
19500
20800
22100
23400
24700
26000
27300
28600
29900
31200
32500
33800
35100
36400
37700
39000
40300
41600
42900
44200
45500
46800
48100
49400
50700
52000
53300
54600
55900
57200
58500
59800
61100
62400
63700
65000
66300
67600
68900
70200
71500
72800
74100
75400
76700
78000
79300

```
80600
81900
83200
84500
85800
87100
88400
89700
91000
92300
93600
94900
96200
97500
98800
100100
101400
102700
104000
105300
106600
107900
109200
110500
111800
113100
114400
115700
117000
118300
119600
120900
122200
123500
124800
126100
127400
128700
1300
2600
3900
5200
6500
7800
9100
10400
11700
13000
```

```
14300
15600
16900
18200
19500
20800
22100
23400
24700
26000
```

[ ]:

```

```

[5]:

```
y_train = label_train[:26000]
```

[6]:

```
len(x_train)
```

[6]:

```
26000
```

[7]:

```
# # Add channel to greyscale images so that it has 3 channels required by␣
 ↪Alexnet.
# # Add 1 more dimension to tensor to represent channel;
# labels = []

# # Function to load images from a folder directory with multiple sub-folders
# def load_images_from_folder(folder):
#     images = []
#     for root, _, filenames in os.walk(folder):
#         for filename in filenames:
#             # here filename has the label information
#             label_temp = filename.rsplit('_', 1)[0]
#             labels.append(label_temp)
#             img = cv2.imread(os.path.join(root, filename))
#             if img is not None:
#                 images.append(img)
#     return images

# image_path = "/Users/Limit/imagenet-object-localization-challenge/n01440764/"
# images = load_images_from_folder(image_path)


# # Ensure grayscale images have 3 channels
# for idx, image in enumerate(images):
#     if len(image.shape) == 2:  # If grayscale image
#         image = cv2.cvtColor(image, cv2.COLOR_GRAY2RGB)  # Convert to␣
 ↪3-channel image
#         images[idx] = image
# # print the number of unique labels:
# # unique_elements = list(set(labels))
# # print(len(unique_elements))
```

[8]:

```
len(x_train)
```

[8]: 26000

[9]:

```
# Define your dataset class
class CustomDataset(Dataset):
    def __init__(self, data, labels):
        self.data = data
        self.labels = labels

    def __len__(self):
        return len(self.data)

    def __getitem__(self, idx):
        return self.data[idx], torch.tensor(self.labels[idx])
```

[10]:

```
# unique_labels = labels.copy()
# unique_labels = list(set(unique_labels))
# unique_labels.sort()
# dict_labels = {}
# for i in range(len(unique_labels)):
#     dict_labels[unique_labels[i]] = i
```

[11]:

```
# labels_numerical = []
# for i in labels:
#     labels_numerical.append(dict_labels[i])
```

[ ]:

```

```

[12]:

```
# Define your loss function
criterion = nn.CrossEntropyLoss()

# Define your optimizer
model = AlexNet().to(device)
model.initialize_weights()

optimizer = optim.Adam(model.parameters(), lr=1e-10, weight_decay=5e-4)

# Prepare your data
train_data = x_train  # List of input tensors
train_labels = y_train  # List of corresponding labels
```

```python
dataset = CustomDataset(train_data, train_labels)
dataloader = DataLoader(dataset, batch_size=128, shuffle=True)
losses = []
grad_magnitudes = []


# Train your models
num_epochs = 10
for epoch in range(num_epochs):
    current_loss = 0.0
    for inputs, labels in dataloader:

        optimizer.zero_grad()
        outputs = model(inputs)
        Before = list(model.parameters())[0].clone()

        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

        current_loss += loss.item()
        After = list(model.parameters())[0].clone()

        # Log loss
        losses.append(loss.item())
        total_grad_norm = 0.0
        # Log gradient magnitudes
        for param in model.parameters():
            if param.grad is not None:
                total_grad_norm += param.grad.norm().item() ** 2
        total_grad_norm = total_grad_norm ** 0.5  # Taking the square root to get the norm
        grad_magnitudes.append(total_grad_norm)

    for param_group in optimizer.param_groups:
        print("Learning rate:", param_group['lr'])
    print()
    print('another way to print learning rate:')
    for group in optimizer.param_groups:
        for p in group['params']:
            print(p.grad)  # Print gradients
    print('end of p.grad')
    print()
    # Verify whether gradient is computed successfully
    print(torch.equal(Before.data, After.data))
    print(f'Epoch {epoch+1} finished')
    epoch_loss = current_loss / len(dataset)
    print(f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')
    print('**************************************************************')
    print()
```

```
Learning rate: 1e-10

another way to print learning rate:
tensor([[[[-3.0159e-02, -4.8719e-02, -5.8743e-02,  …, -4.1884e-02,
           -1.2790e-02,  3.1551e-02],
          [-7.2237e-02, -8.0503e-02, -6.9788e-02,  …, -8.2153e-03,
            2.8003e-02,  3.8666e-02],
          [-1.0637e-01, -1.0923e-01, -9.4848e-02,  …,  3.7936e-02,
            5.0295e-02,  2.9648e-02],
          …,
          [-6.8481e-02, -5.8184e-02, -6.9721e-02,  …, -2.1899e-02,
           -9.0854e-03,  8.7588e-03],
          [-6.0504e-02, -3.9292e-02, -2.4440e-02,  …, -2.3642e-02,
            1.9699e-03,  2.5438e-02],
          [-3.1837e-02, -9.1242e-03, -1.9869e-03,  …,  1.4541e-02,
            2.4168e-02,  7.2153e-03]],

         [[-5.8092e-02, -8.7460e-02, -9.3202e-02,  …, -7.9699e-02,
           -4.3930e-02,  2.1810e-03],
          [-9.8094e-02, -1.1629e-01, -1.0281e-01,  …, -2.1990e-02,
            1.0147e-02,  1.9777e-02],
          [-1.3823e-01, -1.4567e-01, -1.2916e-01,  …,  1.8947e-02,
            4.1200e-02,  1.5309e-02],
          …,
          [-7.7942e-02, -6.9136e-02, -7.4424e-02,  …, -3.4075e-02,
           -2.9086e-02, -1.0317e-02],
          [-8.1097e-02, -6.5735e-02, -5.3328e-02,  …, -3.1015e-02,
           -2.4525e-02,  2.6067e-03],
          [-6.5261e-02, -3.4665e-02, -3.4842e-02,  …, -2.6396e-03,
            8.0046e-03, -5.9222e-03]],

         [[-1.6048e-02, -5.0465e-02, -6.3550e-02,  …, -3.3603e-02,
           -1.2857e-02,  3.3391e-02],
          [-5.7602e-02, -8.8228e-02, -7.8969e-02,  …,  2.1566e-02,
            3.9859e-02,  5.5243e-02],
          [-9.7053e-02, -1.1603e-01, -9.6202e-02,  …,  4.6900e-02,
            7.0721e-02,  6.3375e-02],
          …,
          [-5.2242e-02, -4.2838e-02, -4.3695e-02,  …,  1.1772e-02,
            1.7679e-02,  4.6093e-02],
          [-5.3082e-02, -3.8671e-02, -1.8900e-02,  …,  3.1983e-02,
            4.7884e-02,  6.8819e-02],
          [-3.0146e-02, -1.6927e-03,  5.2370e-04,  …,  6.1042e-02,
            6.0355e-02,  3.9329e-02]]],
[[[ 4.7539e-02,  4.0955e-02,  2.3659e-02,  …,  2.0530e-02,
    2.0222e-02,  2.0274e-02],
  [ 3.3435e-02,  2.2307e-02,  3.5571e-02,  …,  1.4244e-02,
    1.8260e-02,  1.2575e-02],
  [ 1.5188e-02,  2.1676e-02,  3.2917e-02,  …,  1.9094e-02,
    4.2994e-03,  7.1473e-03],
  …,
  [-1.2489e-02, -1.6863e-02, -3.6635e-03,  …,  3.5663e-02,
    3.6320e-02,  3.0849e-02],
  [-3.3703e-03, -3.9119e-03, -4.7277e-03,  …,  4.3580e-02,
    5.7339e-02,  4.5600e-02],
  [ 1.0592e-02,  1.2757e-02,  1.2690e-02,  …,  4.1900e-02,
    4.2554e-02,  4.1161e-02]],

 [[ 4.9029e-02,  4.8100e-02,  3.8446e-02,  …,  1.9991e-02,
    1.9122e-02,  1.5439e-02],
  [ 4.2792e-02,  3.7055e-02,  4.7883e-02,  …,  1.6218e-02,
    2.2276e-02,  1.1348e-02],
  [ 2.5461e-02,  2.9090e-02,  3.7706e-02,  …,  2.2477e-02,
    1.6797e-02,  1.0710e-02],
  …,
  [ 1.0731e-02,  4.9013e-03,  2.2180e-02,  …,  3.9948e-02,
    4.3245e-02,  4.8329e-02],
  [ 2.1008e-02,  2.2391e-02,  3.0093e-02,  …,  5.9028e-02,
    7.0103e-02,  6.5604e-02],
  [ 2.6874e-02,  3.5325e-02,  4.6285e-02,  …,  6.5383e-02,
    6.5485e-02,  6.1233e-02]],

 [[ 1.7274e-02,  1.5775e-02, -3.2499e-05,  …, -1.3254e-02,
   -8.1420e-03, -9.9685e-03],
  [ 4.0424e-03, -7.0270e-04,  7.4734e-03,  …, -1.1751e-02,
   -1.6800e-03, -1.2605e-02],
  [-1.4957e-02, -5.2099e-03,  6.6323e-03,  …, -8.2979e-03,
   -8.9556e-03, -1.4350e-02],
  …,
  [-2.0090e-02, -2.6623e-02, -1.1342e-02,  …,  3.0080e-02,
    3.3893e-02,  3.7036e-02],
  [-9.3228e-03, -5.0674e-03,  6.6509e-03,  …,  4.0629e-02,
    5.6718e-02,  5.7045e-02],
  [-1.5796e-03,  5.7210e-03,  2.6240e-02,  …,  4.3121e-02,
    4.8054e-02,  5.3477e-02]]],


[[[-2.5008e-02, -3.4062e-02, -6.1137e-03,  …,  2.5757e-02,
    1.8915e-02,  2.9775e-02],
  [-4.0948e-02, -5.2592e-02, -2.5647e-02,  …,  5.2573e-02,
    4.5372e-02,  2.6190e-02],
  [-2.6357e-02, -5.9442e-02, -3.3594e-02,  …,  7.1840e-02,
    4.9266e-02,  2.7912e-02],
  …,
  [-9.6346e-03, -7.9153e-03,  1.5682e-03,  …,  9.8430e-02,
    8.9909e-02,  1.1264e-01],
  [ 2.4512e-02,  4.5314e-02,  3.7559e-02,  …,  5.3415e-02,
    4.4329e-02,  7.4874e-02],
  [ 7.2047e-02,  7.5855e-02,  6.2331e-02,  …,  9.8641e-02,
    6.9040e-02,  1.0853e-01]],

 [[ 2.5417e-02,  3.7784e-02,  6.7762e-02,  …,  7.9835e-02,
    7.5739e-02,  8.6039e-02],
  [ 1.7621e-02, -3.7582e-03,  5.1002e-02,  …,  1.1813e-01,
    1.1647e-01,  9.1842e-02],
  [ 2.7551e-02, -7.6670e-03,  4.2798e-02,  …,  1.3741e-01,
    1.1232e-01,  7.3011e-02],
  …,
  [ 3.8980e-02,  4.0320e-02,  7.0831e-02,  …,  1.4204e-01,
    1.4735e-01,  1.7534e-01],
  [ 6.4989e-02,  8.9589e-02,  8.0837e-02,  …,  6.6837e-02,
    8.2698e-02,  1.2578e-01],
  [ 9.4905e-02,  9.6906e-02,  7.7613e-02,  …,  1.0602e-01,
    8.9419e-02,  1.4043e-01]],

 [[ 1.0142e-01,  9.5231e-02,  1.1184e-01,  …,  1.1988e-01,
    9.0567e-02,  8.7658e-02],
  [ 9.1480e-02,  7.2439e-02,  1.0994e-01,  …,  1.5526e-01,
    1.2077e-01,  7.8078e-02],
  [ 9.5941e-02,  6.5512e-02,  1.1954e-01,  …,  1.4576e-01,
    1.1491e-01,  6.4193e-02],
  …,
  [ 1.2564e-01,  9.6640e-02,  7.7345e-02,  …,  1.2691e-01,
    1.1896e-01,  1.5278e-01],
  [ 1.4128e-01,  1.2485e-01,  7.8251e-02,  …,  5.3519e-02,
    6.4762e-02,  9.1408e-02],
  [ 1.4622e-01,  1.0927e-01,  1.0372e-01,  …,  8.6752e-02,
    6.7850e-02,  1.0328e-01]]],


…,


[[[ 3.3217e-01,  3.1281e-01,  3.5439e-01,  …,  3.6818e-01,
    2.9912e-01,  2.8320e-01],
  [ 2.7255e-01,  3.1591e-01,  3.7873e-01,  …,  3.6386e-01,
    3.2232e-01,  2.7624e-01],
  [ 2.7277e-01,  2.9888e-01,  3.6610e-01,  …,  2.2720e-01,
   1.7653e-01,  2.1413e-01],
  …,
  [ 1.0667e-01,  1.3845e-01,  1.9604e-01,  …,  2.3609e-01,
    2.7008e-01,  1.8879e-01],
  [ 1.7156e-01,  2.1725e-01,  2.4790e-01,  …,  2.4100e-01,
    2.0989e-01,  1.8699e-01],
  [ 2.4650e-01,  2.1210e-01,  1.9487e-01,  …,  2.1112e-01,
    2.0108e-01,  1.8661e-01]],

 [[ 1.8472e-01,  1.7465e-01,  2.1590e-01,  …,  2.1088e-01,
    1.4218e-01,  1.3880e-01],
  [ 1.2749e-01,  1.8193e-01,  2.4566e-01,  …,  2.4711e-01,
    2.1189e-01,  1.3416e-01],
  [ 1.5790e-01,  1.7269e-01,  2.0309e-01,  …,  1.0425e-01,
    4.1365e-02,  4.8731e-02],
  …,
  [-3.6360e-02, -2.1096e-02,  2.7576e-02,  …,  8.6377e-02,
    1.3500e-01,  4.8272e-02],
  [ 1.1458e-02,  2.4789e-02,  4.7940e-02,  …,  7.4695e-02,
    7.1854e-02,  5.3613e-03],
  [ 9.2438e-02,  3.2668e-02, -4.3059e-03,  …,  2.4974e-02,
    5.3580e-02, -1.6389e-02]],

 [[ 2.0259e-01,  1.9637e-01,  2.1576e-01,  …,  2.0133e-01,
    1.2200e-01,  1.4521e-01],
  [ 1.1863e-01,  1.9612e-01,  2.3526e-01,  …,  2.4358e-01,
    2.1026e-01,  1.4514e-01],
  [ 1.5094e-01,  2.0128e-01,  2.0562e-01,  …,  1.1657e-01,
    6.1049e-02,  4.2142e-02],
  …,
  [ 4.1123e-02,  5.7446e-02,  1.0486e-01,  …,  9.2102e-02,
    1.4825e-01,  5.6373e-02],
  [ 9.7486e-02,  1.2025e-01,  1.2008e-01,  …,  7.2071e-02,
    9.0882e-02,  5.2637e-02],
  [ 1.3550e-01,  1.1189e-01,  6.2081e-02,  …,  4.9823e-02,
    6.4387e-02,  1.5015e-02]]],


[[[ 9.6641e-02,  1.3439e-01,  1.0592e-01,  …,  1.3435e-03,
    3.4535e-02,  1.8786e-02],
  [ 1.5462e-01,  1.7999e-01,  1.5122e-01,  …, -9.0365e-03,
    4.5557e-02,  5.2208e-02],
  [ 9.5431e-02,  1.0809e-01,  1.0218e-01,  …, -3.1457e-02,
    1.2140e-02,  5.8313e-02],
  …,
  [ 3.7890e-02,  4.8855e-02,  2.5143e-02,  …,  1.3620e-02,
    2.3278e-02,  1.9076e-03],
  [ 6.7198e-02,  5.4085e-02,  3.1468e-02,  …, -4.6385e-02,
   -2.8698e-02,  1.1506e-02],
  [ 1.2980e-01,  8.9567e-02,  5.5089e-02,  …, -4.3651e-02,
   -3.4772e-02, -3.1252e-02]],

 [[ 5.2626e-02,  1.1292e-01,  1.0592e-01,  …,  5.0558e-02,
    6.9881e-02,  8.7385e-02],
  [ 1.3813e-01,  1.6178e-01,  1.6851e-01,  …,  3.7441e-02,
    8.6025e-02,  1.1511e-01],
  [ 1.1472e-01,  1.1689e-01,  1.0799e-01,  …,  2.0391e-02,
    5.7352e-02,  1.1294e-01],
  …,
  [ 2.9146e-02,  5.7511e-02,  5.0362e-02,  …, -7.3186e-03,
    1.2245e-02,  3.3462e-02],
  [ 4.0531e-02,  4.8153e-02,  2.7483e-02,  …, -7.8772e-02,
   -3.5630e-02,  3.3870e-02],
  [ 1.1428e-01,  8.1022e-02,  3.4465e-02,  …, -7.2838e-02,
   -3.4121e-02, -1.0549e-02]],

 [[ 7.6667e-02,  1.2305e-01,  1.3374e-01,  …,  3.8857e-02,
    2.3052e-02,  4.1789e-02],
  [ 1.4418e-01,  1.5211e-01,  1.4846e-01,  …,  2.7547e-03,
    9.3826e-03,  1.9562e-02],
  [ 1.1034e-01,  1.0408e-01,  7.9192e-02,  …, -2.1993e-02,
   -3.1186e-02,  1.1442e-03],
  …,
  [ 3.0246e-02,  4.6485e-02,  3.7242e-02,  …, -4.1042e-02,
    2.8492e-03,  5.5580e-03],
  [ 1.8579e-02,  1.7822e-02, -7.1864e-03,  …, -1.2410e-01,
   -6.6223e-02,  3.2811e-03],
  [ 7.1008e-02,  3.2846e-02, -1.5795e-02,  …, -9.2098e-02,
   -7.4342e-02, -3.4704e-02]]],


[[[-9.7282e-04,  7.5125e-03,  1.1862e-02,  …, -3.4619e-02,
   -3.0268e-02, -2.2832e-02],
  [ 6.5790e-03,  1.3131e-02,  6.8466e-03,  …, -3.0510e-02,
   -3.2872e-02, -2.3249e-02],
  [ 8.3701e-03,  7.5984e-03, -1.0860e-02,  …, -3.7481e-02,
   -3.8355e-02, -3.2677e-02],
  …,
  [ 1.3118e-02,  5.7313e-03,  1.5081e-02,  …, -1.0112e-02,
   -1.6327e-02, -4.2776e-03],
  [-1.3297e-02, -1.0686e-02,  1.7493e-02,  …, -2.2124e-02,
   -3.0113e-02, -2.7171e-02],
  [-1.2517e-02, -1.0856e-02,  3.6446e-03,  …, -2.5224e-02,
   -4.0523e-02, -3.4867e-02]],

 [[-3.3327e-02, -2.6597e-02, -2.9782e-02,  …, -5.8481e-02,
           -6.3636e-02, -6.0962e-02],
          [-2.8862e-02, -2.2449e-02, -3.4102e-02,  …, -7.1712e-02,
           -7.4477e-02, -6.6827e-02],
          [-3.4475e-02, -3.1935e-02, -4.9550e-02,  …, -7.6437e-02,
           -8.3160e-02, -7.3951e-02],
          …,
          [-2.6900e-02, -3.7282e-02, -2.2923e-02,  …, -5.6694e-02,
           -6.0094e-02, -4.3057e-02],
          [-4.2940e-02, -5.3616e-02, -2.3714e-02,  …, -6.3472e-02,
           -6.9066e-02, -6.5664e-02],
          [-4.1140e-02, -4.8326e-02, -2.8874e-02,  …, -6.6648e-02,
           -7.7513e-02, -6.9412e-02]],

         [[-2.3500e-02, -1.5441e-02, -2.0581e-02,  …, -6.3744e-02,
           -7.5516e-02, -6.5724e-02],
          [-2.3807e-02, -2.0226e-02, -3.1078e-02,  …, -8.0872e-02,
           -8.5865e-02, -7.2593e-02],
          [-2.2953e-02, -2.9310e-02, -4.6663e-02,  …, -8.0961e-02,
           -9.2810e-02, -8.3039e-02],
          …,
          [-2.0316e-02, -2.3970e-02, -1.0517e-02,  …, -3.0540e-02,
           -4.0005e-02, -2.9248e-02],
          [-4.0092e-02, -3.8163e-02, -9.3266e-03,  …, -3.4188e-02,
           -5.0157e-02, -4.7641e-02],
          [-3.1939e-02, -3.2332e-02, -1.5980e-02,  …, -4.2863e-02,
           -5.6758e-02, -4.3981e-02]]]])
tensor([ 4.6566e-08,  1.1362e-07, -1.7066e-07, -1.0384e-07,  1.0245e-07,
        -4.8429e-08, -4.2841e-08, -1.4901e-08,  1.6764e-07,  3.2037e-07,
         1.6391e-07,  5.6345e-08,  6.9849e-09, -5.2154e-08,  3.0920e-07,
         5.5879e-09, -1.3504e-07, -1.2480e-07, -7.4506e-09, -1.0058e-07,
         3.3528e-08, -1.8906e-07,  6.5193e-08, -1.6391e-07,  1.0245e-08,
         5.3085e-08, -2.8312e-07,  1.8720e-07,  9.4995e-08, -1.9465e-07,
        -1.1781e-07,  6.5193e-09, -1.1548e-07, -4.0978e-08, -3.5157e-07,
        -2.0023e-08, -3.8091e-07,  2.1979e-07, -3.7998e-07,  3.2131e-08,
         9.3132e-08, -1.3970e-07,  5.7655e-08, -2.0862e-07,  1.7788e-07,
        -1.7881e-07,  9.4995e-08, -8.3819e-09,  3.1665e-08, -4.0978e-08,
        -1.1735e-07, -8.5682e-08, -6.0908e-07,  1.6764e-08,  4.0978e-08,
        -2.9802e-08, -9.6858e-08, -2.6450e-07,  1.2759e-07, -2.7008e-08,
        -2.2724e-07,  1.0431e-07, -2.1420e-08,  8.3819e-08, -3.1665e-08,
         7.6368e-08, -1.0151e-07,  1.8626e-09, -1.4529e-07,  3.7253e-08,
         6.9849e-08,  1.6764e-08, -9.1270e-08, -2.9802e-08, -5.6811e-08,
         2.4680e-08, -4.8080e-08, -4.8429e-08,  7.6368e-08,  9.3132e-10,
         2.9802e-08,  1.3411e-07,  1.0477e-07, -3.8370e-07, -7.4506e-09,
        -4.8429e-08,  4.4797e-07,  3.2783e-07,  3.7253e-08, -2.1583e-07,
         2.5146e-08,  3.4692e-08, -7.4506e-08,  0.0000e+00,  2.1840e-07,
        -7.4506e-08])
tensor([ 2.0541e-03,  1.4512e-02, -1.1767e-02,  6.9167e-04, -4.4476e-03,
         1.1506e-02,  1.2286e-02, -1.8468e-02, -1.4194e-03, -1.3325e-03,
         1.7757e-02, -2.3889e-02,  7.9818e-03,  6.2344e-03, -4.5144e-03,
         2.0428e-02, -1.6079e-02, -2.3937e-03, -2.1206e-02, -1.2263e-02,
         7.7377e-03, -8.0644e-03, -1.5723e-02, -8.9586e-03,  2.1454e-02,
         1.5115e-02,  3.7269e-03,  1.6321e-02, -1.6918e-05, -2.7927e-03,
        -1.1528e-02, -3.4059e-04,  1.3901e-02, -3.6040e-02, -1.7774e-02,
         3.0059e-03, -4.3260e-04,  2.5990e-03, -4.8310e-04,  8.9795e-03,
         2.6002e-02, -4.0295e-04, -1.8400e-02, -5.3652e-03, -1.4653e-02,
        -1.7363e-03,  3.1212e-03, -1.2560e-03, -2.7913e-02,  1.9851e-02,
        -6.3312e-03,  1.3024e-02,  4.1876e-03,  5.2338e-03,  1.2694e-03,
         2.2001e-02,  2.1366e-02, -1.9478e-02,  1.9081e-02, -1.8369e-02,
        -4.3131e-03,  1.8858e-02, -1.3812e-02, -2.0537e-02,  3.4483e-03,
         1.0702e-02,  1.9816e-02,  1.3845e-02, -7.2672e-03, -9.5895e-03,
        -1.3458e-02, -1.1306e-02,  9.1540e-04, -1.2615e-02,  1.1405e-02,
        -1.0941e-03,  3.7044e-02,  1.6525e-02,  1.1517e-02, -1.0028e-02,
        -9.9907e-03, -1.0217e-02,  3.7832e-04, -1.1336e-02, -1.3181e-02,
         2.9550e-02,  9.8688e-03,  4.5496e-03, -2.1341e-03,  2.2173e-02,
        -5.9322e-03, -2.1243e-02,  1.0205e-02, -1.4460e-02, -7.4728e-03,
        -8.3791e-03])
tensor([-1.6223e-02,  2.0171e-02, -9.7963e-03,  8.0583e-03, -2.4743e-03,
         5.4195e-03, -3.5872e-03, -2.2135e-03,  6.0922e-04, -2.6366e-03,
         7.0972e-03, -7.7128e-03,  8.3073e-03,  6.7350e-03, -2.0667e-03,
        -4.1742e-04,  6.5410e-03,  3.2594e-03, -1.7536e-02, -4.5403e-04,
         4.7219e-03, -4.4390e-03, -1.5906e-02,  1.3029e-02,  5.4899e-03,
         6.0558e-03, -8.6200e-03,  2.3151e-04, -1.6651e-03, -4.5878e-03,
        -6.7987e-03, -5.8862e-03,  1.1323e-02, -1.1372e-02, -3.4017e-04,
         4.4765e-03, -2.8653e-03,  3.1757e-03, -1.5756e-02,  1.3644e-02,
         7.2340e-03, -8.0847e-04,  1.9445e-03,  4.9017e-03, -1.2701e-02,
        -1.1221e-02, -3.5123e-03,  5.8876e-04, -1.2786e-02,  3.6473e-03,
        -5.0962e-03,  1.2726e-02,  2.0549e-03,  1.5476e-02,  4.0940e-03,
        -2.5339e-03,  6.8851e-03, -1.7494e-03,  6.3378e-03, -2.3623e-03,
        -6.1455e-03,  1.2669e-02, -1.6162e-02, -1.2588e-02, -6.6873e-03,
         2.2560e-03,  2.1092e-02,  9.0382e-03, -2.6722e-03,  1.7768e-03,
        -3.1692e-03, -6.5067e-03, -4.8213e-03, -1.8851e-02,  7.5182e-03,
        -1.1407e-02,  8.7622e-03,  1.1401e-02,  1.1577e-03, -1.0273e-02,
         3.1752e-03,  7.5624e-03,  2.1106e-03, -1.3474e-02, -1.0435e-02,
         1.3561e-02,  5.1881e-04,  6.9866e-05, -2.5974e-03, -5.2835e-04,
        -1.1898e-02, -7.5803e-03,  9.5796e-03, -1.6832e-03, -5.0406e-03,
         3.2416e-03])
tensor([[[[ 2.0058e-03, -6.5564e-04, -3.7950e-03, -6.9621e-03, -3.2072e-03],
          [ 4.7174e-03,  6.7349e-03, -6.2681e-04, -2.5701e-03, -6.6206e-04],
          [ 5.3983e-03,  9.0043e-03, -1.5660e-03, -4.1413e-03, -3.4767e-03],
          [ 6.0484e-03,  4.7308e-03,  3.7974e-04,  7.6692e-04, -5.1296e-03],
          [ 1.0139e-03,  5.2073e-03,  2.0140e-03,  2.3957e-04, -5.9104e-03]],

         [[-3.0769e-03, -2.7512e-03, -1.2244e-03, -6.0970e-03, -7.7372e-03],
          [ 1.1303e-03,  3.6722e-03,  1.7292e-03,  1.8145e-03, -1.6448e-03],
          [ 2.1397e-03,  2.7610e-03, -1.4394e-04, -1.4084e-03, -9.0419e-04],
          [ 8.7293e-04,  3.9669e-03, -1.6211e-03, -2.5620e-03, -2.8736e-03],
          [-6.2333e-03, -5.2568e-04, -1.4564e-03, -3.0920e-03, -1.1171e-03]],

         [[ 5.3108e-03,  6.7371e-03,  9.0261e-03,  3.4385e-03,  1.4638e-02],
          [ 5.9625e-03,  4.7659e-03,  2.3948e-03,  7.4350e-03,  1.3368e-02],
          [ 1.2383e-02,  5.1568e-03,  7.6728e-03,  2.6751e-03,  7.5066e-03],
          [ 1.1231e-02,  1.3406e-02,  1.0818e-02,  7.4454e-03,  2.0742e-03],
          [ 8.0503e-03, -1.1669e-04,  2.5665e-05,  4.8702e-03,  4.0656e-03]],

         …,

         [[-4.2837e-03, -3.3894e-03,  7.7769e-03,  1.0673e-02,  1.4277e-02],
          [-4.8823e-03, -8.4854e-04,  1.1923e-02,  1.5434e-02,  1.7793e-02],
          [-1.9072e-03,  2.6261e-03, -7.3343e-03,  1.3860e-02,  1.7704e-02],
          [ 1.0274e-03,  2.6958e-03,  3.8650e-03,  8.4750e-03,  8.6592e-03],
          [ 2.7530e-03, -4.3276e-03,  4.9585e-03,  1.6616e-02,  1.0869e-02]],

         [[-1.7460e-03, -4.5906e-03, -3.6635e-03, -3.1055e-03, -3.8425e-03],
          [-5.1926e-03, -5.9172e-03, -4.9572e-04,  2.0265e-03,  4.2858e-03],
          [ 7.3700e-03,  2.0483e-03,  1.8172e-03, -3.9649e-03,  1.8688e-03],
          [ 8.6676e-03,  5.5425e-03,  3.8405e-04, -6.1040e-04, -2.6390e-03],
          [ 7.2196e-03,  3.1377e-03,  3.1229e-06, -4.6612e-03, -5.4480e-03]],

         [[-7.4788e-03, -7.5811e-03, -8.3217e-04, -1.1359e-03, -6.0531e-05],
          [-6.8183e-03, -8.2166e-03, -4.3652e-03, -6.0187e-03, -5.6071e-05],
          [-6.8989e-03, -7.0044e-03, -6.1263e-03, -1.5851e-02, -3.9472e-03],
          [-4.5014e-03, -1.0907e-02, -1.2081e-02, -1.3256e-02, -5.9401e-03],
          [-6.1993e-03, -8.7451e-03, -4.1970e-03, -5.5310e-03, -3.5333e-03]]],


        [[[-1.4790e-02, -2.7802e-03,  8.5407e-03,  2.7035e-03,  1.2335e-02],
          [-1.6839e-02, -5.0358e-03,  2.9096e-03,  1.1531e-02,  1.5766e-02],
          [-1.7342e-02, -1.2684e-02, -1.3175e-02, -7.4427e-03, -6.4396e-03],
          [-8.1649e-03, -1.7238e-02, -1.7226e-02, -1.2290e-02, -3.9044e-03],
          [-1.5334e-02, -1.6512e-02, -2.1557e-02, -2.2732e-02, -1.5672e-02]],

         [[-6.3581e-03,  2.0844e-05, -6.7617e-04, -3.4406e-03,  1.9066e-03],
          [-1.4101e-02, -2.8403e-03,  3.8507e-04,  4.5312e-03,  1.0961e-02],
          [-2.5642e-03,  1.4246e-03,  1.9149e-03,  8.0705e-03,  9.6279e-03],
          [ 1.3368e-03, -1.5815e-03, -8.7264e-03, -5.0552e-03,  6.3950e-03],
          [-2.2586e-03, -3.9236e-03,  1.0579e-03, -2.3511e-03,  3.6996e-03]],

         [[ 1.5646e-02,  2.4099e-02,  2.2887e-02,  2.5270e-02,  1.0633e-02],
          [ 1.4738e-02,  2.4225e-02,  1.7567e-02, -2.8204e-03, -3.6638e-03],
          [ 1.4019e-02,  1.5080e-02,  2.4622e-03, -6.3468e-03, -3.1744e-03],
          [ 5.4185e-03,  1.3505e-02,  8.4553e-04, -7.2992e-03,  4.0095e-03],
          [ 1.1926e-02,  6.9909e-03, -5.1316e-04, -2.1389e-04,  7.7516e-03]],

         …,
         [[-1.3709e-03,  3.2944e-03, -3.9233e-03,  5.0670e-03, -3.9622e-03],
          [-1.9269e-03,  8.4031e-03,  1.2570e-02,  9.4566e-03, -2.6414e-03],
          [ 3.6188e-03,  6.9044e-03,  7.6113e-04, -2.3835e-03, -1.3678e-03],
          [ 9.6408e-03, -5.3544e-03, -6.4404e-03, -2.2888e-03,  4.9251e-03],
          [ 4.9734e-04, -7.4998e-03, -1.3433e-02, -5.5280e-03, -3.2386e-03]],

         [[-1.0670e-02, -2.1436e-04,  5.9772e-03,  1.5762e-02,  9.3504e-03],
          [-1.5544e-02, -6.6827e-03,  5.2917e-03,  9.0572e-03,  1.8085e-02],
          [-2.4067e-02, -1.2140e-02, -4.3218e-03, -1.7148e-03,  4.2589e-03],
          [-7.7544e-03, -9.5399e-03, -1.5129e-02, -1.2840e-02, -2.5368e-03],
          [-1.1307e-02, -4.9662e-03, -1.5374e-02, -1.8655e-02, -1.6070e-02]],

         [[ 5.2852e-03,  1.0083e-02,  4.6112e-03,  3.7622e-03, -8.3252e-03],
          [ 6.1855e-03,  1.2092e-02,  6.5274e-03,  7.6725e-03, -3.2253e-03],
          [-8.6542e-03, -6.4156e-03, -4.6913e-03, -2.2123e-03, -6.0980e-03],
          [-5.3708e-03, -4.9179e-04, -2.9134e-03, -5.9065e-03, -3.9609e-03],
          [-5.7301e-03,  1.2984e-03, -6.8769e-04,  3.3212e-03,  5.4205e-03]]],


        [[[ 6.6754e-03, -6.7096e-03, -1.9619e-02, -2.2911e-02, -2.0664e-02],
          [ 8.0931e-03, -3.4890e-03, -1.0651e-02, -1.9082e-02, -2.3438e-02],
          [ 1.0163e-02,  4.3188e-03,  1.9909e-03, -8.1005e-03, -1.0498e-02],
          [-3.8546e-03, -4.7426e-03,  5.4839e-03, -3.0090e-03, -5.8473e-03],
          [-1.9206e-02, -1.7683e-02, -4.6523e-03, -7.6514e-03, -9.1111e-03]],

         [[ 6.5801e-03,  3.2731e-03, -5.2575e-03, -7.3999e-03, -2.2923e-03],
          [-9.4817e-04, -1.6169e-03, -1.0098e-02, -8.2229e-03, -6.6729e-03],
          [ 2.0274e-03,  3.6333e-03,  2.2258e-04, -1.2857e-03, -4.4863e-03],
          [ 4.5984e-03,  7.7235e-03,  9.6896e-04, -3.0382e-04, -5.8294e-03],
          [-3.2665e-03,  9.1025e-04,  7.6758e-04,  4.2241e-03,  3.3179e-03]],

         [[-4.1402e-03, -1.3397e-03,  3.5438e-03,  6.8064e-03,  4.8889e-03],
          [-1.5026e-03,  1.1196e-02,  2.3950e-02,  1.9713e-02,  2.1794e-02],
          [ 3.6958e-03,  1.3873e-02,  2.1081e-02,  1.1972e-02,  2.6538e-02],
          [-4.2401e-04, -1.7342e-03, -2.7799e-03,  4.0782e-04,  5.6515e-03],
          [-6.1249e-03, -4.6638e-03, -5.0475e-03, -1.2014e-02, -3.4218e-03]],

         …,

         [[-1.2278e-02,  2.5168e-03,  1.2779e-02,  1.3504e-02,  1.6879e-02],
          [-1.1019e-02, -1.5956e-03,  5.9497e-03,  1.6614e-02,  1.8226e-02],
          [ 6.2682e-03, -8.8121e-04,  3.5691e-03,  1.6981e-02,  1.2899e-02],
          [ 2.2204e-05,  4.8361e-03,  3.0125e-03, -1.0182e-03,  2.0349e-03],
          [ 7.2866e-03,  9.5289e-03,  1.2636e-02,  9.5765e-03,  1.2312e-02]],

         [[ 1.6133e-03, -7.8591e-03, -1.4644e-02, -2.5489e-02, -2.5837e-02],
          [ 4.6976e-03, -2.7266e-04, -1.1282e-02, -1.6308e-02, -2.3998e-02],
          [ 4.8484e-03,  9.2770e-04, -1.4288e-04, -6.4292e-03, -2.1444e-02],
          [-8.3412e-03, -4.0712e-03, -6.5197e-03, -7.0663e-03, -1.4734e-02],
          [-1.4614e-02, -8.9031e-03, -2.9249e-03, -4.3256e-03, -1.3753e-02]],

         [[-9.4902e-04,  1.8315e-03,  5.5685e-03,  7.5260e-03,  2.6746e-03],
          [-4.5584e-03, -2.2962e-03, -1.4185e-03,  3.4834e-03,  8.4527e-04],
          [-8.7975e-03, -9.8946e-03, -8.2930e-03, -2.8846e-03, -4.9714e-03],
          [-4.1307e-03, -4.1418e-03, -9.1994e-03, -8.6596e-03, -1.3085e-02],
          [-1.1915e-02, -1.4230e-02, -1.6627e-02, -1.4630e-02, -1.1530e-02]]],


        …,


        [[[ 9.9773e-03,  1.0629e-02,  6.6467e-03,  2.1241e-03, -5.7904e-04],
          [ 1.5154e-02,  1.3024e-02,  6.6563e-03,  9.7523e-03,  3.4658e-03],
          [ 1.2695e-02,  1.1915e-02,  1.7592e-02,  9.2090e-03, -9.2610e-04],
          [ 2.9407e-03,  5.0807e-03,  5.1753e-03,  2.5964e-03, -4.7299e-04],
          [-4.3129e-03, -6.3849e-03, -1.0213e-02, -3.0393e-03,  4.3821e-04]],

         [[-8.4633e-03, -9.3070e-03, -7.4944e-03, -1.0250e-02, -7.8662e-03],
          [-6.9130e-03, -7.0252e-03, -9.1420e-03, -4.5456e-03, -7.9239e-03],
          [-4.3569e-03, -2.5661e-03,  1.2696e-03,  8.4595e-03,  3.4620e-03],
          [-2.9104e-03,  4.4624e-03,  3.2378e-03,  9.1985e-03,  1.3066e-03],
          [-5.1532e-03, -1.7879e-03, -2.2787e-03, -2.2101e-03, -6.0149e-03]],

         [[ 1.3249e-02,  1.1578e-02,  1.2025e-02,  7.1640e-03, -1.5462e-03],
          [ 1.5619e-02,  1.6808e-02,  8.8803e-03,  8.0274e-03, -4.5034e-03],
          [ 2.1194e-02,  8.1972e-03,  4.6982e-04,  5.1258e-03, -2.3513e-03],
          [ 1.9045e-02,  1.1870e-02,  5.8135e-03,  1.5899e-03,  3.5395e-03],
          [ 1.8167e-02,  1.3915e-02,  1.3747e-02,  8.3496e-03, -4.8289e-04]],

         …,

         [[-1.7870e-03,  3.5371e-03,  7.0085e-03,  4.1332e-03, -1.5084e-02],
          [ 3.5361e-03, -3.3991e-03, -5.1887e-03, -4.9789e-03, -1.6020e-02],
          [ 4.2643e-03, -1.3710e-02,  8.4589e-04,  1.6790e-03, -1.0849e-02],
          [-5.4341e-03, -1.6063e-02,  1.4060e-03, -6.7847e-03, -1.8251e-02],
          [ 7.2541e-03, -1.3870e-03, -1.2810e-02, -1.3764e-02, -1.0263e-02]],

         [[ 1.8017e-02,  1.4600e-02,  1.0858e-02,  5.7017e-03,  7.5065e-03],
          [ 1.2436e-02,  9.5299e-03,  6.4237e-03,  7.6076e-03,  7.2875e-03],
          [ 4.8285e-03,  1.0655e-02,  5.1026e-03,  2.7475e-03, -3.3329e-03],
          [ 3.0275e-03, -3.1901e-03,  7.6140e-04,  3.7375e-03, -1.9126e-03],
          [ 3.6890e-04, -9.6159e-03, -9.6552e-03, -1.1279e-02, -2.8789e-03]],

         [[ 1.6369e-02,  1.2864e-02,  8.3374e-03,  1.2737e-02,  1.3485e-03],
          [ 1.6656e-02,  5.2374e-03,  4.2246e-03,  4.9820e-03, -8.9224e-04],
          [ 1.2104e-02,  4.1845e-03,  1.0819e-03,  3.6079e-03,  1.6427e-03],
          [ 1.8312e-02,  8.4768e-03,  7.6439e-03,  9.7281e-03,  6.2262e-03],
          [ 1.8200e-02,  1.3566e-02,  1.2566e-02,  1.3003e-02,  1.0406e-02]]],


        [[[-5.6381e-03, -5.3488e-03, -5.8741e-03, -3.7298e-04,  4.2223e-04],
          [-3.2036e-03, -5.1792e-03, -7.8237e-03,  3.9969e-03, -2.1372e-03],
          [-1.0690e-02, -1.2507e-02, -1.4426e-02, -1.0864e-02, -1.1189e-02],
          [-1.1828e-02, -1.3707e-02, -1.4992e-02, -1.4448e-02, -3.2337e-03],
          [-6.8869e-03, -3.2330e-03, -1.1991e-03, -8.3659e-03,  1.1253e-03]],

         [[-1.0653e-02, -3.0598e-03,  6.2503e-05, -2.6545e-03,  1.7361e-03],
          [ 3.6165e-03,  3.0632e-03,  1.0987e-02,  6.2500e-03,  7.7110e-03],
          [ 2.8369e-03,  8.5106e-03,  1.0065e-02,  5.8833e-03,  1.0707e-02],
          [-3.3037e-04,  3.2971e-03,  3.0284e-03, -4.9911e-03, -4.6776e-03],
          [ 2.2562e-04,  5.7891e-03,  5.3123e-03, -4.2626e-03, -2.4602e-03]],

         [[ 1.7319e-02,  2.4166e-02,  1.7767e-02,  8.5393e-03,  1.1834e-02],
          [ 2.2009e-02,  1.8559e-02,  1.1000e-02,  1.0327e-02,  1.0600e-02],
          [ 9.1503e-03,  7.9247e-03,  4.0583e-03,  1.3998e-02,  1.9045e-02],
          [ 1.2694e-02,  6.0455e-03,  7.0234e-03,  1.0442e-02,  1.9767e-02],
          [ 1.4287e-02,  1.3705e-02,  1.9415e-02,  1.5529e-02,  1.4962e-02]],

         …,

         [[ 2.1213e-02,  1.6753e-02,  2.7164e-02,  2.3371e-02,  1.5007e-02],
          [ 2.4364e-02,  2.8938e-02,  1.7230e-02,  1.4399e-02,  2.2645e-02],
          [ 5.4530e-03,  1.5792e-02,  2.6139e-02,  2.9134e-02,  2.7644e-02],
          [ 1.1752e-03,  1.4719e-02,  3.0143e-02,  2.2333e-02,  9.4477e-03],
          [ 4.3982e-03,  1.0380e-02,  2.1689e-02,  1.5474e-02,  1.4310e-02]],

         [[ 9.1032e-04,  6.9182e-04,  3.4400e-03,  7.6135e-03,  7.2777e-03],
          [-1.4592e-03, -9.2191e-04,  1.3936e-03,  4.8104e-03,  5.8803e-03],
          [-1.3749e-02, -7.8512e-03, -1.2565e-02, -3.5061e-03, -1.1985e-02],
          [-1.4385e-02, -5.8967e-03, -6.4890e-03, -1.2381e-02, -5.7649e-03],
          [-9.3691e-03, -2.7956e-03, -5.4081e-03, -7.2994e-03, -1.7724e-03]],

         [[ 1.0657e-02,  4.7694e-03,  4.0942e-03,  6.5397e-03,  2.1401e-03],
          [ 1.1807e-03, -1.3223e-03,  3.5946e-03,  5.3105e-03,  1.1015e-03],
          [-9.4442e-03, -9.6988e-03,  1.3523e-03,  5.4949e-03,  5.4043e-03],
          [-6.7778e-03, -1.2676e-02, -3.2108e-03,  6.4626e-03,  5.9734e-03],
          [ 4.0680e-03,  2.3465e-04, -1.0291e-03,  7.0237e-03,  4.8954e-03]]],


        [[[-2.1916e-03,  8.0363e-03,  1.1123e-02,  4.9621e-03,  5.6647e-04],
          [-2.8751e-03,  2.8288e-03,  5.7155e-03,  6.1606e-03, -5.0865e-03],
          [-1.9510e-03,  4.6636e-03,  9.1149e-03,  2.5353e-03, -5.5730e-03],
          [ 1.2993e-03,  5.2654e-03,  2.6766e-03,  1.2246e-04, -1.0878e-03],
          [ 1.3051e-02,  1.1327e-02,  2.0516e-03, -1.8444e-03, -2.6735e-03]],

         [[-1.7870e-03,  2.4335e-03,  6.1623e-03,  6.5151e-03, -1.8367e-03],
          [-1.6604e-03,  3.5973e-03,  4.6007e-03,  7.1659e-03,  3.5832e-03],
          [ 6.1626e-04,  1.1453e-03,  3.2843e-03,  1.1421e-02,  2.7539e-03],
          [ 4.3056e-03,  1.1918e-03,  1.4204e-03,  6.2045e-03, -7.9823e-04],
          [ 5.8076e-03,  9.4965e-03,  3.5239e-03, -5.2896e-03, -1.8404e-03]],

         [[ 2.3987e-02,  1.9995e-02,  1.6823e-02,  1.6353e-02,  2.4400e-02],
          [ 7.1649e-03,  3.0770e-03,  7.4557e-03,  9.5928e-03,  1.7895e-02],
          [ 1.0411e-02, -8.2642e-03, -5.1873e-03,  1.1190e-02,  1.3282e-02],
          [-8.1763e-04, -1.4730e-02, -7.8218e-03,  4.9250e-03,  1.5879e-02],
          [ 1.3739e-02, -1.9384e-03, -1.0728e-02,  5.1707e-04,  1.8307e-02]],

         …,

         [[ 2.7757e-02,  1.6456e-02,  2.6324e-02,  1.5503e-02,  1.4005e-02],
          [ 1.3932e-02,  1.4778e-02,  2.3971e-02,  6.1661e-03,  1.7107e-02],
          [ 2.6459e-02,  5.2664e-03,  5.6127e-03,  1.9953e-03,  1.4305e-02],
          [ 2.6226e-02,  2.3121e-02,  1.2547e-02,  1.6391e-02,  7.9325e-03],
          [ 2.2355e-02,  2.5045e-02,  1.2191e-02,  4.8426e-03,  5.7412e-03]],

         [[-3.7065e-03,  9.6596e-03,  8.4394e-03,  4.9697e-03,  1.0260e-03],
          [-3.0559e-03,  1.2797e-02,  3.7818e-03,  1.0404e-02,  1.4032e-04],
          [ 2.9618e-03,  7.0981e-03,  4.3437e-03,  6.8065e-03,  9.5220e-04],
          [ 9.2871e-04,  1.4641e-03, -1.7899e-03, -6.3156e-03,  1.4966e-03],
          [ 1.3869e-02,  1.2818e-02,  2.7918e-03, -1.9487e-03,  4.9543e-04]],

         [[ 1.1341e-02,  9.0960e-03,  1.0901e-02,  1.3080e-02,  1.4791e-02],
          [ 6.4606e-04, -2.2381e-03,  1.1729e-03,  3.4349e-03,  8.4204e-03],
          [-2.7214e-03, -6.1144e-03, -4.2256e-03,  5.6305e-03,  7.7486e-03],
          [-9.3823e-03, -8.6136e-03, -5.7700e-03,  1.5884e-04,  4.8813e-03],
          [-1.6114e-03, -1.7696e-03, -5.7364e-03, -4.1465e-03, -5.9053e-03]]]])
tensor([ 1.2360e-09, -1.2922e-09, -3.5288e-08, -2.0807e-08, -2.7207e-08,
         1.2574e-09, -3.1653e-09, -6.6509e-09,  1.8936e-08,  6.9366e-10,
        -5.2204e-08, -3.2048e-08, -1.6665e-09, -8.9080e-09,  1.5297e-08,
         1.8686e-08,  6.7761e-08,  5.0750e-09, -2.9443e-09,  2.8877e-09,
        -1.0286e-08, -1.7887e-09, -6.6722e-10, -2.1755e-08,  8.0845e-09,
        -3.6175e-08,  5.7429e-10,  9.4386e-10, -9.2992e-09, -9.9530e-09,
        -7.6288e-09,  1.1049e-08, -3.9551e-08, -9.8189e-09,  5.0513e-09,
        -5.7667e-09, -3.5929e-09, -1.5092e-08,  2.1501e-08, -1.1336e-08,
         1.9079e-08, -2.1857e-08,  5.1100e-09, -1.0820e-08, -9.3160e-09,
         2.0893e-08,  1.0200e-08, -4.1623e-09,  1.3824e-08,  9.0769e-09,
         1.1819e-08, -2.7964e-09,  2.3772e-08,  1.1767e-08,  1.2240e-09,
         4.8888e-08,  1.3002e-09, -1.5197e-08,  1.4272e-08,  7.6466e-09,
        -1.1565e-08,  3.3889e-08,  4.5675e-09,  1.1301e-08, -1.6867e-09,
        -3.4970e-09, -3.1673e-09,  5.7446e-10, -2.0284e-08,  4.3771e-09,
         5.9814e-09, -2.1935e-08,  1.1861e-08,  2.6495e-08,  7.3287e-09,
        -1.0440e-08,  1.7569e-09,  1.8494e-08, -5.4995e-09,  5.5709e-09,
         2.3917e-09,  1.5140e-08,  1.2610e-08, -8.8495e-09, -2.1897e-08,
         6.7957e-09,  7.2932e-08,  4.0330e-08, -9.2750e-09, -4.8136e-09,
         1.1891e-08,  6.5147e-09, -2.1580e-08, -1.1693e-09,  2.8597e-08,
        -2.1315e-08,  1.6585e-08,  1.8534e-08,  8.9109e-09,  6.1531e-09,
        -3.0817e-09,  3.6423e-08,  1.4508e-08,  7.0554e-09,  2.0246e-08,
        -2.6976e-09,  6.4254e-08,  9.3387e-09, -8.1982e-09,  2.5119e-08,
        -6.4923e-09,  4.6834e-09,  1.0039e-08, -2.2460e-09, -7.2028e-09,
        -2.7283e-09,  9.5997e-09, -2.3635e-09,  5.8940e-09,  1.2960e-08,
        -6.2231e-09, -2.9744e-08, -1.0417e-08,  1.3550e-08, -3.5935e-08,
         5.1156e-08,  9.8498e-10,  2.7649e-10, -8.8982e-09,  2.0287e-08,
        -3.8951e-08, -1.1639e-09, -1.4686e-08,  1.8387e-08,  3.2257e-08,
         5.1045e-11,  4.4114e-08, -4.5695e-09,  2.2076e-09, -1.4926e-09,
        -3.4926e-09,  1.4047e-09, -5.8288e-09,  8.6823e-09,  5.6496e-09,
         3.7554e-09,  2.5263e-09, -1.2419e-08, -3.5797e-09,  1.0060e-08,
         7.7939e-09,  8.6795e-09, -7.7447e-09, -1.2072e-08,  3.4600e-08,
         1.1113e-09,  9.5891e-09,  3.3606e-09, -1.4317e-08, -7.7035e-09,
        -2.3145e-08,  8.8398e-09,  1.0106e-08, -2.0350e-10, -1.1627e-08,
        -2.4307e-08, -2.9327e-08, -1.5325e-08, -2.9669e-08,  1.8156e-08,
         1.9095e-08,  3.7279e-09, -1.7972e-09,  1.2683e-08, -4.0331e-09,
         9.0268e-09,  3.1107e-09,  1.3508e-08,  1.5825e-08,  5.4357e-09,
         7.2296e-09, -4.4762e-09, -1.3853e-08,  1.0438e-08,  1.0069e-08,
        -2.1600e-08, -1.1063e-08,  1.9124e-09, -2.6219e-08,  2.2352e-08,
         2.5696e-08,  3.3725e-09, -1.1375e-08, -4.5022e-09,  4.0171e-08,
         1.2994e-09,  7.3232e-09,  8.0486e-09, -4.4146e-09, -4.6097e-09,
         6.7221e-09, -2.0011e-09,  4.6958e-09, -2.0589e-08,  1.7933e-08,
        -5.6454e-10,  8.6010e-10,  9.2607e-09, -3.0258e-09, -2.7394e-08,
        -7.3335e-09, -3.0026e-09,  2.8235e-09,  1.6531e-08,  7.0838e-09,
         2.5308e-09,  8.8330e-09,  1.9613e-08, -1.1372e-08, -4.8469e-09,
        -1.4762e-08,  9.0501e-09, -1.8935e-08, -7.2497e-09,  3.1804e-09,
        -1.9807e-08,  5.9129e-09, -1.3428e-08,  1.9860e-08,  2.8343e-08,
         9.5425e-09, -1.7702e-08, -3.1664e-09, -4.8226e-10,  1.3885e-08,
         1.4402e-08,  1.7648e-08, -2.8847e-08, -4.5011e-09, -6.3155e-09,
        -1.9718e-08,  1.1926e-09, -8.0804e-09, -7.9308e-09, -9.1510e-09,
         2.7182e-08,  2.0023e-08,  5.9831e-09, -1.2271e-08, -3.8333e-08,
         1.9072e-09,  1.2963e-08, -7.8762e-09, -8.3601e-09,  2.2362e-08,
         1.0259e-09])
tensor([-0.0190,  0.0114, -0.0101, -0.0053,  0.0069,  0.0116, -0.0153, -0.0006,
         0.0092,  0.0148,  0.0094,  0.0090, -0.0039,  0.0050, -0.0015,  0.0182,
         0.0173,  0.0009, -0.0168,  0.0163, -0.0011,  0.0100,  0.0125, -0.0167,
         0.0053, -0.0069, -0.0109,  0.0026,  0.0055,  0.0245, -0.0177, -0.0004,
         0.0019,  0.0061,  0.0158, -0.0020, -0.0094,  0.0101,  0.0101,  0.0037,
        -0.0201,  0.0134, -0.0067, -0.0051, -0.0069,  0.0078,  0.0156, -0.0096,
         0.0091,  0.0018, -0.0089, -0.0111,  0.0108, -0.0106,  0.0055, -0.0006,
        -0.0050, -0.0052,  0.0020,  0.0194, -0.0059, -0.0145, -0.0054, -0.0028,
        -0.0061,  0.0110,  0.0072, -0.0013, -0.0107,  0.0073,  0.0032, -0.0117,
        -0.0055,  0.0228,  0.0208, -0.0053,  0.0188,  0.0182, -0.0061, -0.0141,
         0.0088,  0.0033, -0.0114,  0.0249,  0.0110,  0.0140, -0.0081,  0.0092,
         0.0103, -0.0002, -0.0113, -0.0016,  0.0205, -0.0006, -0.0022,  0.0081,
        -0.0116, -0.0193, -0.0091,  0.0026, -0.0340, -0.0020, -0.0121,  0.0087,
        -0.0075,  0.0152, -0.0042, -0.0096,  0.0209,  0.0151, -0.0092, -0.0054,
         0.0041,  0.0102,  0.0142,  0.0104,  0.0073,  0.0094,  0.0103,  0.0055,
         0.0081, -0.0138,  0.0026,  0.0125,  0.0326, -0.0070, -0.0150, -0.0105,
         0.0039, -0.0056, -0.0198, -0.0076, -0.0137,  0.0002,  0.0073,  0.0004,
        -0.0060,  0.0013,  0.0102,  0.0057,  0.0139, -0.0200, -0.0130, -0.0011,
         0.0094, -0.0029,  0.0102,  0.0070,  0.0086,  0.0058,  0.0088,  0.0086,
        -0.0118,  0.0090, -0.0096,  0.0068, -0.0025,  0.0091,  0.0093,  0.0036,
         0.0012, -0.0018,  0.0022,  0.0154,  0.0156,  0.0235,  0.0109,  0.0206,
         0.0063, -0.0177, -0.0081, -0.0112,  0.0065,  0.0014,  0.0059,  0.0027,
         0.0157, -0.0019,  0.0166,  0.0117, -0.0122, -0.0016, -0.0099,  0.0086,
         0.0071,  0.0156,  0.0108, -0.0095,  0.0019,  0.0030, -0.0152, -0.0124,
         0.0030, -0.0045, -0.0046,  0.0058,  0.0160,  0.0110, -0.0001, -0.0066,
         0.0213, -0.0135, -0.0060,  0.0066, -0.0093, -0.0008,  0.0046,  0.0085,
         0.0211,  0.0025,  0.0038,  0.0044, -0.0120,  0.0182, -0.0047, -0.0006,
         0.0144,  0.0004, -0.0035, -0.0140,  0.0058,  0.0056,  0.0053, -0.0026,
        -0.0013,  0.0038, -0.0161,  0.0090,  0.0207,  0.0011,  0.0053, -0.0068,
        -0.0161,  0.0048,  0.0135, -0.0006, -0.0168,  0.0138, -0.0136, -0.0117,
         0.0130, -0.0006, -0.0105,  0.0123,  0.0136,  0.0078, -0.0065, -0.0025,
         0.0076,  0.0313,  0.0216, -0.0093,  0.0091,  0.0134, -0.0028,  0.0201])
tensor([-0.0133,  0.0089, -0.0070, -0.0066,  0.0048,  0.0107, -0.0102, -0.0046,
        -0.0033,  0.0029,  0.0015,  0.0079, -0.0067,  0.0023, -0.0021,  0.0194,
        -0.0035,  0.0135, -0.0188,  0.0105,  0.0027,  0.0079,  0.0050, -0.0128,
        -0.0013, -0.0098, -0.0079,  0.0033,  0.0054,  0.0166,  0.0017,  0.0042,
         0.0068,  0.0066,  0.0071,  0.0017,  0.0013,  0.0081,  0.0074,  0.0037,
        -0.0169,  0.0146, -0.0073, -0.0055, -0.0011,  0.0037,  0.0183, -0.0077,
         0.0121,  0.0075, -0.0111, -0.0085,  0.0100, -0.0083,  0.0034,  0.0062,
        -0.0052, -0.0023, -0.0013,  0.0127, -0.0038, -0.0026, -0.0066, -0.0032,
        -0.0073,  0.0062,  0.0124, -0.0051, -0.0110,  0.0044, -0.0039, -0.0057,
         0.0068,  0.0090,  0.0060, -0.0040,  0.0108,  0.0101, -0.0041, -0.0013,
        -0.0012, -0.0026, -0.0124,  0.0051,  0.0018,  0.0113, -0.0012,  0.0142,
        -0.0015,  0.0065, -0.0047, -0.0148,  0.0135, -0.0019, -0.0066,  0.0033,
        -0.0086, -0.0140, -0.0073, -0.0057, -0.0162, -0.0004, -0.0025,  0.0028,
        -0.0108,  0.0113, -0.0128, -0.0107,  0.0104,  0.0121, -0.0062, -0.0093,
         0.0052,  0.0055,  0.0081,  0.0020, -0.0050,  0.0087,  0.0025, -0.0016,
         0.0017, -0.0113,  0.0004,  0.0154,  0.0113, -0.0007, -0.0089, -0.0038,
        -0.0036, -0.0061, -0.0128, -0.0110, -0.0061, -0.0003,  0.0027,  0.0049,
         0.0021,  0.0079, -0.0042, -0.0004,  0.0082, -0.0211, -0.0063,  0.0008,
         0.0105,  0.0020,  0.0117,  0.0048,  0.0068,  0.0044,  0.0073,  0.0081,
        -0.0046,  0.0079, -0.0136,  0.0050, -0.0028,  0.0111,  0.0134,  0.0076,
        -0.0025,  0.0003,  0.0017,  0.0086,  0.0083,  0.0215,  0.0012,  0.0185,
         0.0025, -0.0036, -0.0065, -0.0080, -0.0004,  0.0021,  0.0056,  0.0062,
         0.0103, -0.0093,  0.0116,  0.0020, -0.0065,  0.0071, -0.0115,  0.0019,
         0.0144,  0.0109,  0.0096, -0.0036, -0.0055,  0.0014, -0.0092, -0.0076,
         0.0089,  0.0006,  0.0025,  0.0132,  0.0095,  0.0067,  0.0089,  0.0035,
         0.0030,  0.0015,  0.0004,  0.0032, -0.0144, -0.0047,  0.0049,  0.0011,
         0.0153, -0.0008,  0.0053,  0.0076, -0.0121,  0.0160,  0.0024,  0.0010,
         0.0067,  0.0015,  0.0014, -0.0050,  0.0035,  0.0027,  0.0015,  0.0011,
         0.0009,  0.0044, -0.0028,  0.0051,  0.0151,  0.0077,  0.0094,  0.0019,
        -0.0193, -0.0032,  0.0087,  0.0032, -0.0117,  0.0076, -0.0020, -0.0072,
         0.0111, -0.0068, -0.0130,  0.0114,  0.0044,  0.0003,  0.0014, -0.0127,
         0.0022,  0.0215,  0.0131, -0.0052,  0.0092,  0.0123,  0.0011,  0.0167])
tensor([[[[ 4.2626e-03,  8.1350e-03,  1.2894e-03],
          [ 4.8499e-03,  2.3401e-03,  1.5324e-03],
          [ 8.9318e-03,  1.0457e-02,  9.4479e-03]],

         [[-2.2377e-04,  1.1226e-02,  1.0609e-02],
          [-1.2600e-03,  1.1278e-02,  1.0585e-02],
          [ 1.0317e-02,  1.0275e-02,  2.3686e-03]],

         [[ 1.2431e-02,  7.4749e-03,  1.0008e-03],
          [-5.4056e-03, -1.1676e-03,  1.2097e-03],
          [-3.3397e-03,  8.8642e-03,  6.1937e-03]],

         …,

         [[ 6.3725e-03,  1.6197e-02,  1.2309e-02],
          [ 1.5443e-03,  6.0672e-03,  3.5455e-03],
          [ 3.9215e-03,  4.7084e-03,  5.9576e-03]],

         [[-8.1538e-05,  1.2697e-03,  5.2084e-03],
          [-3.7320e-03, -3.6461e-03, -5.1699e-04],
          [-3.4634e-03, -6.5887e-03, -4.1284e-03]],

         [[ 3.3329e-03,  9.6644e-03,  2.8297e-03],
          [ 1.6620e-03,  4.2839e-03,  4.1762e-03],
          [ 1.0770e-02,  1.5732e-02,  5.8993e-03]]],


        [[[ 3.6254e-03,  4.1576e-03,  5.7313e-03],
          [ 4.9221e-03,  6.0289e-03,  8.5548e-03],
          [ 5.1389e-03,  4.9678e-03,  7.3005e-03]],

         [[ 5.0911e-03,  6.8132e-03,  8.3485e-03],
          [ 6.8878e-03,  1.2052e-02,  1.1970e-02],
          [ 5.1140e-03,  1.1049e-02,  9.9857e-03]],

         [[ 1.0049e-02,  1.1683e-02,  1.0736e-02],
          [ 1.4469e-02,  1.4205e-02,  1.2310e-02],
          [ 1.3384e-02,  1.4187e-02,  1.4866e-02]],

         …,

         [[ 1.3417e-02,  1.1662e-02,  8.2283e-03],
          [ 1.3408e-02,  1.2623e-02,  1.0965e-02],
          [ 1.5207e-02,  1.3657e-02,  9.3852e-03]],

         [[ 1.3861e-02,  1.2713e-02,  1.3968e-02],
          [ 1.8759e-02,  1.9093e-02,  1.7677e-02],
          [ 1.1913e-02,  1.4422e-02,  1.5460e-02]],

         [[ 4.6063e-03,  5.9907e-03,  6.9679e-03],
          [ 6.6983e-03,  5.7922e-03,  5.6996e-03],
          [ 5.6230e-03,  9.6892e-03,  8.7195e-03]]],


        [[[ 1.5764e-02,  1.6362e-02,  1.3764e-02],
          [ 1.4726e-02,  1.5362e-02,  1.2149e-02],
          [ 1.4699e-02,  1.3765e-02,  1.1625e-02]],

         [[ 9.2744e-03,  4.0829e-03,  4.6620e-03],
          [ 1.0720e-02,  8.3013e-03,  5.4411e-03],
          [ 1.5757e-02,  1.3719e-02,  7.9940e-03]],

         [[ 1.1418e-03, -4.4418e-03, -2.5652e-03],
          [-1.9954e-03, -1.9073e-03, -3.4291e-03],
          [ 6.3938e-03,  6.2277e-03,  4.0505e-03]],

         …,

         [[ 2.3095e-03, -1.8453e-03, -4.0888e-03],
          [-1.5095e-03, -1.8065e-03, -1.8633e-03],
          [-3.0248e-03, -2.4441e-03, -2.8127e-03]],

         [[ 1.1358e-02,  1.1196e-02,  1.1159e-02],
          [ 8.5663e-03,  8.8751e-03,  4.4243e-03],
          [ 1.0764e-02,  9.6223e-03,  6.5381e-03]],

         [[ 2.3729e-02,  1.4858e-02,  9.6373e-03],
          [ 1.7086e-02,  1.0344e-02,  6.8078e-03],
          [ 1.5811e-02,  1.5824e-02,  1.2729e-02]]],


        …,


        [[[ 8.4689e-04, -1.0711e-03, -9.6081e-04],
          [ 2.5669e-03,  9.4944e-04, -5.4394e-04],
          [ 2.1182e-03,  1.2244e-03, -1.3002e-03]],

         [[-2.7328e-03, -8.3261e-04, -9.9777e-04],
          [-1.2441e-03, -6.8845e-04, -5.0568e-04],
          [-2.5058e-03, -1.0565e-03, -1.2564e-03]],
         [[-1.7823e-03, -5.2654e-04, -1.1133e-03],
          [-3.6750e-03, -5.5973e-04, -2.1898e-03],
          [-3.5264e-03, -7.3315e-05, -2.2657e-03]],

         …,

         [[ 1.4076e-03,  1.1631e-03,  1.1313e-03],
          [ 1.2183e-03,  3.6505e-04,  1.5832e-03],
          [-4.8117e-04,  7.1279e-04,  2.5129e-03]],

         [[-1.4271e-03, -8.2415e-04, -2.5448e-03],
          [-2.5592e-03, -3.8943e-03, -2.3574e-03],
          [-3.3093e-03, -3.7739e-03, -1.9077e-03]],

         [[-5.2817e-04, -5.9117e-04,  3.5778e-04],
          [ 1.4405e-03,  1.3527e-03,  1.0360e-03],
          [ 1.5900e-03,  3.8093e-03,  2.5845e-03]]],


        [[[-8.6868e-04,  5.2230e-05, -1.3214e-03],
          [-2.4136e-03, -1.8149e-03, -5.9361e-03],
          [ 2.8348e-04,  3.5228e-04, -4.5905e-03]],

         [[ 1.2584e-02,  1.3271e-02,  9.1390e-03],
          [ 1.2026e-02,  8.3111e-03,  6.7404e-03],
          [ 5.6441e-03,  9.7629e-03,  1.1940e-02]],

         [[ 2.2107e-02,  1.9752e-02,  2.2178e-02],
          [ 1.7087e-02,  1.6388e-02,  2.0485e-02],
          [ 1.6186e-02,  1.0783e-02,  1.4033e-02]],

         …,

         [[ 1.5904e-02,  1.5786e-02,  1.7359e-02],
          [ 6.8627e-03,  1.0150e-02,  1.4625e-02],
          [ 1.2125e-02,  1.1386e-02,  8.9812e-03]],

         [[ 4.9863e-03,  4.6592e-03,  3.3607e-03],
          [-2.2572e-03, -2.9273e-03, -4.5522e-03],
          [ 4.4798e-03,  4.1452e-03,  3.4073e-03]],

         [[-3.6065e-03,  6.7462e-03,  3.1259e-03],
          [-8.8774e-03, -1.8808e-04, -3.1363e-03],
          [ 1.8349e-03,  7.7920e-03,  3.9206e-03]]],


        [[[ 4.9613e-03,  3.5232e-03, -1.4635e-03],
                  [ 3.5836e-04, -6.3585e-05, -1.1771e-04],
                  [ 5.2669e-03,  2.1539e-04,  2.8601e-03]],

                 [[-2.8228e-03, -2.0255e-03, -7.0547e-03],
                  [ 3.0262e-03, -4.5433e-04, -2.8221e-03],
                  [ 2.2609e-03, -4.8003e-03, -1.4958e-03]],

                 [[ 2.1611e-03, -1.8246e-03, -7.7178e-03],
                  [ 2.0715e-03, -6.0743e-04, -1.1830e-02],
                  [ 2.7284e-03,  9.1241e-03,  7.8921e-04]],

                 …,

                 [[ 1.0350e-02,  7.0573e-03, -4.4555e-03],
                  [ 3.9216e-03, -1.8173e-03, -9.8242e-03],
                  [ 1.5843e-02,  7.0260e-03, -6.5851e-04]],

                 [[ 7.3434e-03,  1.1636e-02,  7.2861e-03],
                  [ 1.0185e-02,  1.4729e-02,  9.5557e-03],
                  [ 7.4613e-03,  1.2924e-02,  1.4436e-02]],

                 [[ 9.2278e-03,  1.1587e-03, -1.4049e-03],
                  [ 2.1351e-03, -8.9114e-03,  2.1633e-03],
                  [ 9.4915e-03, -4.0400e-04, -1.9359e-03]]]])
        tensor([ 4.5518e-04,  1.3439e-02,  8.3259e-03, -3.9493e-03, -3.0504e-04,
                -1.1685e-02,  1.1734e-04, -4.9925e-04, -1.0259e-02,  8.4847e-03,
                -2.9505e-03,  8.6464e-03,  9.1817e-03, -6.2803e-03, -4.7856e-04,
                -3.2516e-03, -5.4575e-03,  3.6809e-02, -1.5129e-02, -2.0234e-02,
                -9.4614e-03, -7.0499e-03,  8.0370e-03,  3.2038e-03, -1.1738e-02,
                 1.3518e-02, -5.3540e-04,  2.8394e-02,  1.6771e-03, -1.7810e-02,
                -2.4675e-03, -8.0715e-04, -2.6143e-03, -2.6201e-03, -3.7850e-04,
                 7.6388e-03, -1.6222e-02,  5.4625e-03,  1.3557e-02, -1.0999e-02,
                 1.0111e-02,  4.1691e-03, -1.2280e-02, -1.0530e-02,  2.5821e-03,
                 1.0162e-02, -8.9376e-03, -4.9507e-04, -2.6895e-03,  8.7789e-03,
                -1.2172e-02, -4.3845e-03, -9.6719e-03,  6.3926e-03,  6.7187e-03,
                 1.4597e-03, -2.7475e-03,  9.3370e-03, -1.3161e-03,  3.6881e-03,
                -1.1035e-02,  9.5060e-04,  1.5460e-03,  2.0945e-03, -3.3391e-03,
                -3.5299e-04, -4.5063e-03,  1.0433e-02, -5.2061e-03, -3.1765e-03,
                -4.8179e-03,  2.6855e-02, -9.7616e-04, -9.0868e-03,  7.0675e-03,
                 3.1480e-03,  5.1231e-03, -1.7118e-04, -1.2565e-03,  3.7247e-03,
                -3.9096e-03, -1.1693e-02,  4.4805e-03,  5.8028e-03,  2.4084e-03,
                -4.0658e-03,  1.3540e-02, -6.2449e-03, -9.5139e-04,  1.9977e-03,
                 2.1986e-02, -7.5013e-04, -3.9986e-04, -1.2762e-02,  1.2813e-02,
                 1.9634e-03, -3.1343e-03,  1.1899e-03, -9.0923e-03, -9.4593e-03,
                 2.1518e-03, -9.3969e-03,  7.8191e-03,  1.2309e-03, -8.2094e-03,
                 2.3064e-02,  5.1150e-04, -5.4265e-03,  5.6735e-03,  7.5143e-04,
                 1.8137e-02, -1.2133e-02,  1.6761e-02,  3.4347e-03,  6.8234e-04,
                -6.7668e-03,  1.8915e-03,  3.6734e-03,  1.4897e-02,  1.3944e-03,
        -2.0481e-02, -4.0626e-03, -1.1831e-02,  1.0927e-02,  1.3643e-03,
        -5.5097e-03,  4.3687e-03,  4.5861e-03,  1.1739e-03,  1.1837e-02,
        -2.3755e-03, -1.2805e-02,  1.0695e-02, -2.0495e-03, -3.2240e-03,
        -1.2456e-02, -9.3316e-05, -1.0082e-02, -4.4722e-03,  9.4349e-04,
        -1.8397e-02,  1.1965e-02, -3.2680e-04, -8.3976e-03, -4.2294e-04,
         4.4003e-03, -7.1846e-03,  5.2618e-03, -2.2504e-03, -6.4854e-03,
        -3.0356e-03,  4.0144e-03,  6.0647e-03, -3.9367e-03,  2.8592e-02,
        -2.0930e-03, -9.5628e-03,  1.6697e-02,  5.2567e-03,  1.0512e-02,
         1.8791e-02,  3.6042e-04,  3.7513e-03,  1.3882e-03, -8.7285e-03,
         1.2396e-03, -1.1415e-02,  1.4309e-02,  1.3690e-03,  3.7715e-04,
        -2.5321e-03,  2.3527e-03,  1.7667e-03,  6.0755e-03, -8.6348e-04,
         1.7887e-03, -1.2613e-03,  6.3473e-03,  5.1307e-03,  6.6613e-03,
         1.7700e-03,  1.1412e-02, -3.0097e-03, -6.2732e-04,  1.0679e-02,
         5.8217e-03,  1.0311e-02,  6.6345e-03, -3.7653e-03,  8.8641e-03,
        -5.4448e-03,  2.9904e-03, -3.0913e-03, -9.4521e-04,  7.7538e-03,
        -1.0741e-03, -1.1071e-03, -2.8729e-03,  7.1803e-03,  1.1507e-02,
        -3.6047e-03,  8.0566e-03, -1.2579e-03,  3.3712e-02,  4.0247e-03,
         4.7954e-03,  1.5863e-02,  2.9873e-02, -2.3016e-03,  4.7208e-03,
        -6.1129e-03,  2.5925e-02,  7.0407e-04,  6.1254e-03,  1.7166e-02,
         7.5784e-03, -2.2434e-03,  8.5898e-04,  7.5907e-03,  3.0702e-03,
        -1.6618e-03, -1.3529e-02,  8.1663e-05, -1.5757e-03, -2.0333e-02,
         3.9388e-04, -1.4837e-02, -3.0916e-03, -1.9576e-03,  6.4340e-03,
        -8.0607e-04, -7.9028e-03,  1.4747e-02, -7.2805e-03,  4.7471e-03,
         1.2581e-02,  7.0661e-03, -5.3474e-03,  9.3304e-03, -1.4677e-03,
        -8.7609e-04,  1.0058e-02, -2.4193e-04, -1.5941e-03, -9.0572e-03,
         6.8763e-04, -2.3692e-03,  7.6516e-03, -5.7157e-03, -6.9091e-04,
         4.4771e-03,  7.7333e-04,  2.6588e-03,  1.9209e-03, -1.0535e-02,
        -1.5519e-02,  1.4926e-02, -5.0328e-03,  6.0476e-03, -4.0335e-03,
        -3.7063e-03,  2.2098e-02,  3.7092e-04,  1.3539e-02, -4.5603e-03,
        -7.6920e-03, -1.3647e-02,  2.4828e-03,  1.8629e-03, -9.6365e-03,
         7.4769e-03,  2.0788e-02, -1.5007e-05, -4.3136e-03, -9.7447e-03,
        -3.2316e-03,  2.1179e-03,  1.7506e-02, -1.2528e-02,  3.9252e-03,
         2.5495e-04,  3.2040e-03,  1.2051e-02,  2.1735e-03,  8.3258e-03,
         1.7512e-04, -2.3094e-03, -1.3645e-03, -6.0382e-03, -2.6322e-03,
         1.9893e-02,  3.3634e-03, -2.5263e-04, -4.5998e-03, -1.9901e-03,
         7.6305e-03,  0.0000e+00, -1.3542e-03, -3.5543e-03, -1.1621e-03,
         4.6903e-03, -5.3576e-04,  2.0494e-02,  1.7455e-03,  2.5489e-03,
         2.3520e-03, -1.9614e-03,  5.3561e-03, -4.8947e-04,  3.6677e-03,
        -8.9811e-03, -7.8044e-03, -5.4345e-03, -8.9941e-03, -4.2252e-04,
        -1.6903e-02,  4.8683e-04,  1.1371e-02, -4.1355e-03,  2.0278e-03,
         2.4489e-03,  6.4368e-03, -4.3404e-03, -5.4723e-03, -1.8901e-03,
         1.3530e-02,  9.5312e-04, -1.1008e-02, -5.3508e-03,  1.3627e-02,
         1.0328e-02, -1.2387e-02,  6.8107e-03,  2.4210e-02, -1.2363e-02,
        -1.1071e-03,  2.5760e-03,  8.8223e-05,  9.0715e-03, -1.7807e-03,
        -6.1560e-03, -2.3593e-03,  2.3320e-03, -1.8924e-03,  4.1894e-03,
        -3.4348e-02,  5.7162e-04,  1.3944e-03,  2.3492e-02, -6.7284e-03,
        -6.4771e-03,  3.1608e-03,  1.6298e-03, -9.0739e-03,  3.5723e-03,
        -3.8310e-04,  1.2706e-02,  5.3783e-03, -4.6351e-03, -1.2051e-04,
                 2.3620e-02, -9.1216e-04, -2.1395e-03, -1.5139e-03,  5.4582e-03,
                -4.8138e-03,  3.7196e-03,  3.2084e-03, -3.7760e-04, -1.4788e-02,
                 4.7989e-03,  1.5431e-03, -2.8689e-03, -2.9166e-02,  5.8768e-03,
                 6.4472e-03,  1.0247e-02,  7.6035e-04, -7.8461e-03,  5.3034e-03,
                 1.5665e-02, -5.3150e-04,  7.8126e-03,  3.5273e-03])
        tensor([[[[ 1.6274e-02,  2.5139e-02,  2.3986e-02],
                  [ 1.6844e-02,  2.6643e-02,  2.2549e-02],
                  [ 1.2542e-02,  2.0755e-02,  1.8794e-02]],

                 [[ 3.0062e-03,  1.3317e-03, -1.8148e-04],
                  [ 2.8355e-03,  7.6308e-04, -4.0032e-03],
                  [ 2.0234e-03, -1.5507e-03, -1.5210e-03]],

                 [[ 9.4195e-03,  4.9848e-03,  1.9842e-04],
                  [ 6.4259e-03,  5.7351e-04,  8.5521e-04],
                  [ 1.3805e-03,  9.0772e-04,  2.5906e-03]],

                 …,

                 [[ 2.8204e-03,  6.2496e-04,  3.7753e-03],
                  [ 1.7048e-03, -2.7913e-03,  1.1061e-03],
                  [ 1.2650e-03,  1.0574e-03, -2.4633e-03]],

                 [[ 6.6981e-03,  8.3499e-03,  7.3757e-03],
                  [ 7.0942e-03,  1.0574e-02,  1.1388e-02],
                  [ 6.5394e-03,  6.4885e-03,  5.6581e-03]],

                 [[ 3.0620e-02,  1.8910e-02,  1.7718e-02],
                  [ 3.3769e-02,  1.2396e-02,  1.2529e-02],
                  [ 2.2474e-02,  8.1725e-03,  1.6568e-02]]],


                [[[ 1.0351e-02,  6.2822e-03,  1.1061e-02],
                  [-1.7322e-03,  1.5426e-03,  1.2089e-02],
                  [ 6.9611e-03,  7.0031e-03,  1.6799e-02]],

                 [[ 4.5011e-04,  2.9194e-03,  7.3309e-03],
                  [ 1.9906e-03,  7.8338e-04,  4.4009e-03],
                  [ 1.3673e-03, -2.4692e-03,  1.9898e-03]],

                 [[ 3.7050e-03,  2.0563e-03,  4.6682e-03],
                  [ 5.1989e-03,  5.3582e-03,  1.1044e-02],
                  [ 5.0782e-03,  7.9865e-03,  7.0409e-03]],

                 …,

                 [[-8.5384e-04, -1.3073e-03,  8.8558e-04],
                  [-1.1774e-03,  1.3940e-03,  8.9070e-05],
          [ 1.8737e-03,  3.1426e-03,  4.6718e-04]],

         [[ 5.4268e-03,  6.2239e-03,  6.0902e-03],
          [ 1.0099e-02,  4.1879e-03, -1.1613e-03],
          [ 3.6402e-03,  2.8776e-03,  2.7131e-03]],

         [[ 1.1156e-02, -1.2311e-03,  1.4143e-02],
          [ 3.1346e-03,  1.7606e-03,  2.0594e-02],
          [-7.0847e-03, -2.2953e-03,  1.7591e-02]]],


        [[[-8.0769e-03, -1.5535e-02, -2.2248e-02],
          [-7.8105e-03, -8.4002e-03, -5.3068e-03],
          [-1.3427e-03, -5.8848e-03, -8.5538e-03]],

         [[ 1.4281e-03,  4.7736e-03,  5.3078e-03],
          [-2.0499e-04, -1.2098e-03,  4.6515e-03],
          [ 1.6755e-03, -1.4425e-03, -1.7029e-03]],

         [[-7.7088e-05,  3.7549e-03, -9.2815e-04],
          [ 1.2769e-03,  2.4020e-03,  4.3576e-03],
          [ 6.4381e-03,  2.6909e-03,  5.8762e-03]],

         …,

         [[-8.9366e-04, -5.2820e-04,  3.2931e-04],
          [-2.9557e-03,  4.6716e-04,  1.8521e-03],
          [-3.3634e-03,  1.6262e-04,  7.3525e-04]],

         [[ 9.1657e-03,  8.7615e-03,  9.4234e-03],
          [ 1.0279e-02,  9.7673e-03,  8.3837e-03],
          [ 2.3588e-03,  7.7759e-03,  2.0641e-03]],

         [[-2.4561e-03, -7.1384e-03, -1.6908e-03],
          [ 1.5049e-03,  8.7883e-03,  2.1211e-03],
          [ 6.0672e-03, -1.8987e-04, -4.7907e-03]]],


        …,


        [[[ 1.0827e-02,  8.3649e-03,  1.2303e-02],
          [ 8.3960e-03, -3.8603e-03,  9.0274e-03],
          [ 9.2727e-03,  8.3025e-03,  1.3901e-02]],

         [[-9.8473e-04, -9.4255e-05, -1.2092e-03],
          [ 6.9143e-04,  1.1139e-03,  2.7654e-03],
          [ 5.8692e-04,  2.1285e-03,  2.7927e-03]],
         [[-6.2874e-03, -6.3691e-03, -1.4887e-03],
          [-5.4063e-03, -4.5764e-03, -1.1297e-03],
          [-4.8059e-03, -5.3316e-03,  1.5650e-03]],

         …,

         [[ 3.9581e-04,  3.1536e-04,  6.9468e-04],
          [-3.9541e-04, -6.2982e-04, -1.9519e-04],
          [ 5.7714e-04,  1.6529e-03,  2.3005e-03]],

         [[-1.9895e-02, -1.5339e-02, -1.0308e-02],
          [-6.1204e-03, -2.4767e-03, -3.8483e-03],
          [-6.5966e-03, -1.0369e-02, -1.2606e-02]],

         [[ 2.2728e-03,  1.1535e-02,  1.0672e-02],
          [-2.7928e-03,  1.9107e-02,  1.9657e-02],
          [ 5.3596e-04,  2.1297e-02,  2.5332e-02]]],


        [[[ 1.8749e-02,  1.0904e-02,  1.5156e-02],
          [ 1.3721e-02,  2.2499e-02,  2.5031e-02],
          [ 7.8219e-03,  2.2812e-02,  2.6603e-02]],

         [[ 1.1454e-03,  1.9587e-03,  1.6844e-03],
          [ 2.0429e-03,  9.4989e-04,  1.8370e-03],
          [ 5.3351e-04, -2.4865e-03, -9.2414e-05]],

         [[ 2.1406e-04,  6.1702e-03,  2.6520e-03],
          [ 5.2192e-03,  4.6386e-03,  6.6447e-03],
          [ 8.8167e-03,  3.2095e-03,  6.1108e-03]],

         …,

         [[ 1.8333e-03,  7.7552e-05,  1.8345e-04],
          [ 2.6971e-04,  6.4813e-04,  1.6720e-03],
          [ 1.5391e-03,  8.4835e-05,  1.8435e-04]],

         [[-2.7691e-03, -1.3872e-03,  6.6111e-03],
          [-3.2684e-04, -1.8789e-03,  2.2384e-03],
          [-1.1466e-03,  2.4341e-03,  8.6149e-03]],

         [[ 1.1018e-02,  2.0556e-02,  1.6278e-02],
          [ 7.6084e-03,  2.2317e-02,  2.5330e-02],
          [ 1.8752e-02,  2.5268e-02,  1.9514e-02]]],


        [[[-1.9086e-04, -7.8569e-04, -1.6424e-03],
                  [-1.0172e-03, -9.2244e-04, -1.3073e-03],
                  [-3.6537e-04, -9.0746e-05, -3.6970e-04]],

                 [[-2.5341e-04, -2.5767e-05, -1.3888e-04],
                  [-3.6624e-04,  9.4824e-05, -1.4752e-05],
                  [-2.7152e-04, -2.8887e-04, -1.0159e-04]],

                 [[-4.4299e-04, -9.7693e-07,  1.6300e-04],
                  [-1.1676e-04,  6.3089e-05,  8.5326e-05],
                  [-2.3957e-04, -2.3544e-04, -1.0299e-04]],

                 …,

                 [[-1.6565e-04,  8.7075e-05, -8.4717e-05],
                  [-5.5734e-05,  1.6033e-05, -2.9626e-04],
                  [-2.0297e-04,  2.2661e-05, -1.7152e-04]],

                 [[-2.6847e-04, -4.8197e-04, -5.5337e-04],
                  [-4.0280e-05, -7.1809e-05,  6.9868e-05],
                  [ 1.5874e-05,  4.9702e-05,  9.1762e-07]],

                 [[-2.0148e-03,  4.4531e-04, -1.2853e-03],
                  [-1.7529e-03, -6.5677e-04, -5.1332e-04],
                  [ 2.3486e-05, -5.1334e-04, -7.2126e-05]]]]])
        tensor([ 1.9748e-02,  2.4086e-02, -8.1642e-03, -1.0753e-03, -1.4867e-02,
                -1.2356e-03,  3.1048e-03, -6.6378e-03,  9.0560e-03, -1.9042e-02,
                 2.1869e-03, -1.4240e-03,  1.1802e-02, -1.4870e-02,  1.1456e-02,
                -1.1676e-02,  7.7850e-03,  7.4915e-03, -4.9983e-03,  9.4720e-03,
                -8.5167e-03,  1.1649e-02,  1.7327e-03,  2.2564e-02, -5.6727e-03,
                 2.1090e-02, -1.1107e-02,  1.6424e-03, -1.9744e-02, -1.3112e-02,
                -7.0562e-03, -2.3277e-02,  1.2383e-03,  3.1637e-03, -1.1340e-03,
                -1.2782e-02,  4.0066e-03,  1.0987e-03,  1.8099e-02, -4.9916e-04,
                -8.0169e-03, -7.3432e-03,  4.5773e-03, -3.8786e-02, -4.9721e-04,
                -5.3294e-04, -1.6759e-03, -1.3006e-02, -6.5041e-03,  7.2949e-03,
                -2.9964e-03,  1.4207e-02,  8.5112e-04, -1.9208e-02, -3.1910e-03,
                -7.8185e-03,  1.6681e-03, -2.1252e-04, -1.0575e-03, -9.9504e-03,
                -1.5652e-02,  4.0859e-04, -5.7352e-04, -7.6983e-03, -8.6024e-04,
                -1.6849e-03,  8.3878e-04, -1.6129e-02,  5.5886e-03,  7.1546e-03,
                 3.2140e-02,  3.1544e-02, -3.3169e-02, -5.8800e-03,  1.6805e-03,
                -8.5414e-03,  2.7124e-02, -1.2460e-02, -1.6504e-03, -9.2734e-03,
                -1.4207e-02, -2.8867e-03, -1.2964e-02,  4.8541e-03,  1.0758e-02,
                 7.3559e-04, -8.9534e-03, -8.5328e-03, -2.5097e-04, -4.4786e-03,
                 2.1587e-02, -4.1605e-03,  3.5859e-03, -4.4087e-03,  1.5686e-03,
                 7.9819e-03, -9.6926e-04, -1.8995e-03, -1.3508e-02, -6.8646e-03,
                 1.1912e-02, -5.9098e-03, -1.7701e-04,  4.0927e-03, -2.7646e-02,
                -7.7708e-03,  1.1855e-02,  6.4186e-04,  2.2174e-04,  2.3067e-03,
                -2.5196e-03, -2.1760e-05, -4.7059e-03, -3.6034e-02, -2.2865e-04,
                 8.1596e-04, -5.0671e-03, -9.5696e-04,  1.1705e-02,  2.7447e-02,
        -5.5285e-03, -1.4753e-02,  2.2725e-03, -8.2763e-03, -7.3155e-03,
         1.4016e-02, -2.5631e-03,  5.7785e-03, -7.4946e-03, -4.4582e-03,
        -9.0698e-03,  3.0930e-03, -4.6994e-03,  4.7947e-03, -2.7052e-03,
         1.1853e-02,  3.5102e-02,  8.6231e-04, -1.3696e-02, -9.5323e-03,
         6.2765e-03, -7.8749e-04, -2.3031e-02, -8.3894e-03,  2.3832e-02,
        -1.6692e-03, -8.8883e-03,  2.3506e-03, -8.6770e-03,  3.7269e-03,
        -1.2636e-02,  1.4815e-02, -1.8156e-02, -2.6646e-03, -1.8174e-02,
         7.6034e-03, -5.8013e-04, -7.2688e-03,  2.0964e-03,  6.1797e-04,
         4.0955e-03,  4.2162e-03,  2.2452e-03, -6.3273e-05, -4.8890e-03,
         9.8512e-03, -6.8655e-04,  5.4413e-03, -5.4886e-03,  3.1326e-02,
         1.7440e-02,  8.1106e-03,  2.8439e-02, -9.2858e-03, -3.9252e-03,
        -3.9602e-03,  9.2441e-03, -8.4750e-04, -1.8622e-04, -2.3024e-02,
         4.3806e-03,  2.2073e-05, -1.2719e-03,  2.8502e-02,  1.3469e-02,
        -5.4493e-03, -1.0878e-02,  1.9181e-02,  1.1818e-02, -1.0247e-02,
         3.4986e-03, -1.2875e-03,  1.9344e-02, -3.5427e-03,  1.0868e-02,
         3.6193e-03,  1.5844e-03, -1.3521e-02, -1.0768e-02, -4.5277e-03,
        -2.5967e-02,  1.3638e-02, -7.9841e-04, -3.9801e-03,  1.9486e-03,
         4.7071e-03, -6.9070e-04,  1.8497e-02,  1.2443e-02,  1.8332e-03,
         5.0427e-02, -1.3482e-03,  7.6370e-03,  1.7844e-02, -1.7708e-03,
         1.2953e-02,  2.7693e-03, -8.8900e-03, -1.6981e-03,  3.5460e-02,
        -3.7194e-03, -3.8518e-03,  1.8761e-02,  8.2577e-03, -9.4853e-03,
         5.6529e-03,  2.6027e-03, -1.9941e-03,  2.0187e-02, -7.3219e-03,
         1.2810e-02,  4.3460e-02,  1.3627e-02,  4.1688e-03,  4.6284e-03,
         2.1625e-03, -1.4322e-03,  2.2584e-03, -1.3880e-03,  1.2602e-03,
        -2.9931e-03, -1.4323e-02, -1.8654e-02, -2.1297e-03, -8.1187e-03,
        -1.2261e-02,  6.4060e-05, -8.1940e-03,  3.4232e-03, -3.4814e-03,
        -1.5011e-02,  2.0663e-02, -2.0527e-02, -4.4575e-03,  1.4267e-02,
        -7.4471e-04,  2.3463e-02, -4.6608e-03,  4.6586e-03,  3.8301e-03,
         9.7145e-03,  1.5281e-03,  1.8864e-02,  4.7574e-03,  2.4551e-03,
         2.8675e-03,  1.5469e-03, -3.5300e-03,  1.1795e-03,  2.1692e-02,
         2.5792e-03,  7.9610e-03, -5.4471e-04,  1.0216e-02, -8.4722e-03,
         3.9347e-02, -9.1447e-03,  7.3303e-04,  5.3610e-02,  1.3956e-02,
         1.4548e-02,  6.0914e-03,  2.3280e-03, -1.0750e-02, -3.4579e-03,
        -1.2369e-02,  3.2749e-02, -7.0464e-03,  4.7454e-03,  2.3082e-02,
         1.3592e-02, -1.1551e-02, -6.1379e-03,  2.9437e-03, -2.5722e-05,
        -1.3870e-02,  3.6687e-04, -1.7258e-03,  2.1114e-02, -4.1842e-03,
         1.4071e-03,  7.2994e-03, -8.4964e-03,  4.3764e-02,  4.7220e-03,
         7.4755e-03,  6.1190e-03,  4.2294e-03,  9.4921e-03, -1.2751e-03,
         7.7613e-03,  1.5296e-04, -1.5781e-02,  1.1586e-02, -1.3101e-02,
         7.2827e-03, -5.4304e-03, -1.3643e-03, -2.1989e-02, -1.8064e-03,
        -4.3804e-04,  2.2830e-02, -1.2769e-02, -2.4463e-03, -1.7535e-02,
        -1.1660e-02,  2.4502e-02,  6.8699e-03, -5.2979e-03,  8.4100e-03,
        -5.6499e-03,  7.1430e-03,  1.4991e-02, -5.4805e-04,  1.8452e-03,
        -2.1653e-02,  3.4004e-03,  1.7589e-02,  1.4940e-02,  1.6004e-03,
         7.4650e-03,  2.8273e-02,  1.3044e-02,  9.1585e-04,  2.1514e-02,
         7.3205e-03, -4.3850e-03,  9.3162e-04,  2.1255e-03,  8.6208e-03,
         1.2580e-02,  5.2927e-03,  3.8233e-04, -1.3556e-02,  2.1479e-02,
         7.6984e-04,  1.5290e-03,  7.0703e-03, -2.1461e-02,  1.5121e-02,
                 1.1192e-02,  4.2355e-03,  1.9020e-03,  1.6043e-02,  5.0862e-03,
                -1.8949e-03,  7.6114e-04, -8.4212e-03,  6.4817e-04, -3.0797e-03,
                 1.0594e-02, -1.3265e-04, -1.2823e-03, -1.2753e-03, -2.1049e-03,
                -9.0010e-03,  2.1198e-03,  9.4539e-03,  5.7928e-04,  1.6127e-02,
                 4.7800e-03,  2.9743e-03,  1.6364e-02, -1.4396e-03])
        tensor([[[[ 1.9207e-03,  2.3834e-03, -1.0954e-03],
                  [ 3.7300e-03,  1.2971e-03, -1.6397e-03],
                  [-8.9260e-03, -1.4671e-02, -8.5108e-03]],

                 [[-3.1789e-02, -3.1056e-02, -1.4768e-02],
                  [-3.8555e-02, -2.8712e-02, -1.5523e-02],
                  [-3.8648e-02, -3.2050e-02, -2.0969e-02]],

                 [[-8.8128e-03, -7.7889e-03, -8.5585e-04],
                  [-1.6058e-02, -1.3751e-02, -6.7816e-03],
                  [-1.3857e-02, -6.0714e-03, -8.2680e-03]],

                 …,

                 [[-5.7967e-03, -9.7712e-03,  2.0513e-03],
                  [-5.0367e-03, -1.3967e-03,  8.8508e-03],
                  [-1.7454e-03, -3.1994e-03,  6.6608e-03]],

                 [[ 3.2556e-03,  6.4755e-04, -1.2056e-03],
                  [ 7.3317e-03,  1.7022e-02,  7.0620e-03],
                  [ 3.2440e-03,  2.5754e-03,  5.7816e-03]],

                 [[-4.9330e-04, -4.1088e-04, -7.6116e-04],
                  [-3.8603e-04, -6.4244e-06, -9.9028e-04],
                  [-1.3972e-05,  0.0000e+00,  6.8397e-08]]],


                [[[ 3.5472e-03,  1.7846e-03, -6.7616e-05],
                  [ 4.3224e-03,  5.0718e-03, -6.1944e-04],
                  [-2.0499e-03, -1.9210e-03, -9.0033e-03]],

                 [[ 2.2302e-02,  3.2033e-02,  2.3775e-02],
                  [ 1.3046e-02,  2.2941e-02,  2.8771e-02],
                  [ 1.3149e-02,  1.5303e-02,  1.5249e-02]],

                 [[ 4.4483e-03,  3.6751e-03, -6.3293e-03],
                  [ 1.4932e-03,  9.0464e-03,  5.0887e-04],
                  [ 7.7518e-03, -4.5527e-04,  7.4594e-04]],

                 …,

                 [[ 6.3509e-03,  6.6529e-03, -5.2192e-04],
                  [-9.3736e-04,  1.7312e-03,  2.9595e-03],
          [-1.3828e-03, -3.8834e-03,  1.3510e-03]],

         [[ 6.4726e-03,  6.8532e-03, -1.0027e-03],
          [ 5.9697e-03,  5.9749e-03,  4.2616e-03],
          [ 6.1364e-03,  1.1444e-02,  4.7696e-03]],

         [[ 1.7053e-04,  4.6816e-04,  0.0000e+00],
          [-2.7435e-05,  0.0000e+00, -1.1499e-04],
          [-1.1227e-05,  4.4476e-04, -2.9729e-05]]],


        [[[-1.8311e-03, -8.2472e-03, -8.2197e-03],
          [-3.7106e-03, -4.8054e-03, -6.0645e-03],
          [-8.4416e-03, -9.2115e-03, -2.7671e-03]],

         [[-1.4647e-02, -6.2610e-03, -4.4206e-03],
          [-1.9976e-02, -1.0569e-02, -5.7485e-03],
          [-1.7452e-02, -1.1789e-02, -8.3658e-03]],

         [[-1.7869e-03, -7.5967e-03, -5.4236e-03],
          [-2.1033e-03, -6.4340e-03, -3.4944e-03],
          [-4.8561e-03, -2.1560e-03, -2.8573e-03]],

         …,

         [[-5.1423e-03, -2.8965e-03, -6.6030e-03],
          [-6.5891e-03, -1.9138e-03, -5.0303e-03],
          [-5.1099e-03,  1.8488e-03, -2.5194e-03]],

         [[-3.3467e-03, -5.6455e-03, -4.9538e-03],
          [-3.5338e-03, -5.9367e-03, -3.8308e-03],
          [-2.1187e-03, -7.5593e-04, -2.7160e-03]],

         [[ 0.0000e+00,  0.0000e+00,  0.0000e+00],
          [ 0.0000e+00, -2.1027e-04, -8.9535e-05],
          [ 0.0000e+00,  0.0000e+00,  0.0000e+00]]],


        …,


        [[[ 1.9330e-03, -2.9532e-03, -5.5919e-03],
          [ 2.7911e-03, -1.0817e-03, -2.3040e-03],
          [ 3.8575e-03,  3.6618e-03, -1.9421e-03]],

         [[ 6.7850e-03, -5.1319e-03, -3.7051e-03],
          [ 7.0849e-03, -8.6668e-04, -5.8699e-03],
          [ 1.8738e-03, -4.1276e-03, -1.2978e-02]],
         [[-4.9013e-03, -6.0337e-03, -2.0487e-03],
          [-7.9683e-03, -5.2531e-03,  1.9608e-04],
          [-5.4796e-03,  7.5357e-04, -4.5330e-03]],

         …,

         [[-4.6785e-03, -2.8939e-03,  1.0207e-03],
          [-4.6450e-04, -1.6807e-03, -1.0606e-03],
          [-1.2679e-03, -7.8500e-05,  5.0333e-03]],

         [[-2.4173e-03, -6.3663e-03, -5.9968e-03],
          [-6.7828e-03, -2.2760e-03, -1.9684e-03],
          [-5.0216e-03,  1.5243e-03,  1.8207e-05]],

         [[ 0.0000e+00, -3.6953e-04,  1.8483e-04],
          [-3.6159e-04, -5.9800e-05, -1.8002e-04],
          [ 5.9198e-05, -4.8060e-04, -9.0545e-05]]],


        [[[ 2.5975e-03,  9.8703e-03, -8.7856e-04],
          [ 1.7498e-03,  2.0032e-03,  1.0192e-03],
          [ 5.9209e-03, -3.4650e-03,  2.1552e-03]],

         [[ 2.3900e-02,  2.7617e-02,  2.9860e-02],
          [ 1.2392e-02,  2.7320e-02,  2.6796e-02],
          [ 7.0897e-03,  2.1732e-02,  1.9361e-02]],

         [[ 1.0195e-02,  1.3101e-02,  1.5383e-03],
          [ 1.1293e-02,  1.1078e-06,  4.9914e-03],
          [ 2.8364e-03, -6.8010e-04,  4.6351e-03]],

         …,

         [[-1.8213e-03,  3.2706e-03, -1.3942e-03],
          [-5.8717e-03, -4.0537e-03, -4.2349e-03],
          [ 5.3427e-04,  3.4216e-03, -8.4542e-03]],

         [[-1.4580e-03, -8.9591e-03, -5.0759e-03],
          [ 1.0136e-02,  1.7400e-04, -5.0188e-03],
          [ 1.4877e-02,  1.4218e-02,  9.8916e-03]],

         [[ 0.0000e+00,  0.0000e+00, -6.0712e-04],
          [-2.6171e-04,  3.4966e-05,  0.0000e+00],
          [ 1.2189e-04,  0.0000e+00, -9.4597e-05]]],


        [[[-1.2504e-03, -8.3550e-04, -6.6743e-03],
          [ 5.3835e-03,  8.1654e-03,  3.8759e-03],
          [-8.3910e-03,  4.9120e-03, -2.7753e-04]],

         [[-2.4887e-02, -1.2432e-02,  2.1913e-03],
          [-1.5423e-02, -1.4222e-02, -2.0511e-02],
          [-2.0302e-02, -1.6486e-02, -1.7563e-02]],

         [[ 1.7530e-03,  1.8691e-03, -8.8638e-04],
          [ 8.0579e-04,  4.2054e-03, -7.8361e-03],
          [-4.6694e-03,  2.6675e-03, -6.1190e-03]],

         …,

         [[-2.3405e-03,  4.0764e-03,  7.2441e-03],
          [-7.9998e-03, -5.7551e-03,  6.9389e-04],
          [-9.8938e-03, -2.1008e-03, -4.2331e-03]],

         [[-5.8848e-04,  8.5743e-04,  4.5877e-03],
          [ 9.2807e-03,  2.5294e-03,  9.4195e-03],
          [ 3.2689e-03, -3.5621e-03,  2.1687e-03]],

         [[-2.1424e-04,  6.0449e-05, -2.1172e-04],
          [ 0.0000e+00,  7.1449e-05,  9.2306e-05],
          [ 1.5816e-04,  1.1947e-04,  2.8884e-04]]]])
tensor([-0.0300,  0.0142, -0.0162,  0.0181,  0.0225,  0.0216,  0.0107,  0.0274,
        -0.0071, -0.0092, -0.0003,  0.0495, -0.0266,  0.0011,  0.0209,  0.0093,
         0.0168,  0.0469, -0.0135, -0.0243, -0.0051,  0.0081,  0.0555,  0.0317,
         0.0031,  0.0810,  0.0222, -0.0084,  0.0114, -0.0156,  0.0415, -0.0018,
        -0.0172, -0.0235,  0.0018,  0.0184, -0.0084,  0.0224,  0.0290,  0.0011,
         0.0187, -0.0362, -0.0259,  0.0201,  0.0069,  0.0083, -0.0077,  0.0111,
        -0.0062, -0.0082, -0.0139,  0.0101,  0.0150, -0.0287, -0.0319, -0.0134,
         0.0336, -0.0077,  0.0404,  0.0090,  0.0091,  0.0102, -0.0057, -0.0336,
         0.0013, -0.0031,  0.0066,  0.0020,  0.0119,  0.0173,  0.0179,  0.0011,
        -0.0031,  0.0261, -0.0074, -0.0246, -0.0194,  0.0300, -0.0080,  0.0261,
         0.0012, -0.0118,  0.0082, -0.0118, -0.0125,  0.0259,  0.0041, -0.0072,
        -0.0194,  0.0013,  0.0237, -0.0128,  0.0134,  0.0200, -0.0324, -0.0002,
        -0.0275,  0.0071,  0.0152, -0.0101, -0.0004, -0.0147,  0.0263,  0.0085,
        -0.0038, -0.0262, -0.0389,  0.0214,  0.0054, -0.0305, -0.0033,  0.0236,
         0.0273,  0.0014,  0.0473,  0.0248,  0.0226, -0.0131,  0.0044,  0.0336,
        -0.0077,  0.0134, -0.0145,  0.0163, -0.0155,  0.0151,  0.0371,  0.0152,
        -0.0168, -0.0088,  0.0014,  0.0109,  0.0125,  0.0245, -0.0074,  0.0267,
        -0.0067, -0.0002, -0.0018,  0.0130, -0.0231,  0.0063, -0.0106, -0.0006,
         0.0012,  0.0082,  0.0105, -0.0051,  0.0129, -0.0450, -0.0097,  0.0191,
         0.0140, -0.0023,  0.0016, -0.0109,  0.0305, -0.0172, -0.0103, -0.0442,
         0.0023,  0.0117, -0.0067, -0.0055, -0.0256,  0.0018,  0.0040, -0.0085,
         0.0128,  0.0269,  0.0070, -0.0053,  0.0531,  0.0254,  0.0122, -0.0207,
        -0.0054,  0.0059,  0.0195,  0.0032,  0.0068,  0.0051, -0.0063,  0.0057,
        -0.0033,  0.0422, -0.0069, -0.0163, -0.0117, -0.0017, -0.0004,  0.0435,
         0.0259, -0.0188,  0.0018, -0.0293, -0.0119, -0.0025, -0.0247,  0.0131,
        -0.0092,  0.0034, -0.0110, -0.0135, -0.0196,  0.0177,  0.0077,  0.0095,
        -0.0111, -0.0192, -0.0393, -0.0179,  0.0227, -0.0248, -0.0401, -0.0014,
         0.0012, -0.0145, -0.0139, -0.0161, -0.0294, -0.0275, -0.0033, -0.0173,
        -0.0282,  0.0238,  0.0079, -0.0022,  0.0030,  0.0058,  0.0040, -0.0215,
        -0.0362,  0.0343, -0.0631,  0.0505,  0.0013, -0.0024, -0.0063, -0.0092,
         0.0237,  0.0161,  0.0184,  0.0140, -0.0080, -0.0089, -0.0314,  0.0192,
         0.0032,  0.0043,  0.0392, -0.0106,  0.0167, -0.0062,  0.0106, -0.0072])
tensor([[-4.9161e-06,  1.2127e-05,  2.7000e-04,  …, -1.0834e-03,
         -1.3799e-04, -3.0782e-05],
        [-3.9491e-04, -4.6077e-04, -1.1291e-03,  …, -6.1800e-05,
          5.4905e-04,  3.7408e-04],
        [-1.1095e-03,  1.3967e-03, -7.2145e-04,  …, -1.7862e-03,
          1.6512e-04,  2.5351e-03],
        …,
        [-4.6327e-03, -3.0094e-03, -2.7497e-04,  …, -1.1056e-03,
         -2.1648e-04, -1.7803e-03],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  …,  0.0000e+00,
          0.0000e+00,  0.0000e+00],
        [ 0.0000e+00,  0.0000e+00,  0.0000e+00,  …,  0.0000e+00,
          1.4439e-04,  4.1247e-05]])
tensor([-0.0023, -0.0006,  0.0012,  …, -0.0042,  0.0000,  0.0002])
tensor([[ 0.0024, -0.0002,  0.0013,  …, -0.0083,  0.0000, -0.0032],
        [-0.0111,  0.0066,  0.0014,  …, -0.0135,  0.0000,  0.0000],
        [-0.0032, -0.0063, -0.0117,  …, -0.0044,  0.0000,  0.0036],
        …,
        [-0.0058,  0.0085, -0.0056,  …,  0.0036,  0.0000, -0.0024],
        [-0.0021, -0.0120, -0.0027,  …, -0.0070,  0.0000, -0.0016],
        [ 0.0046,  0.0171,  0.0168,  …,  0.0150,  0.0000, -0.0038]])
tensor([-0.0025, -0.0039, -0.0020,  …, -0.0010, -0.0048,  0.0042])
tensor([[ 2.9031e-04,  5.4359e-04,  5.9013e-04,  …,  5.7797e-04,
          1.6916e-04,  4.9334e-04],
        [-3.0680e-03, -1.5783e-01, -2.2812e-02,  …, -2.6560e-01,
         -7.8460e-02, -4.9600e-02],
        [ 2.8365e-04,  5.2318e-04,  3.5235e-04,  …,  6.1531e-04,
          9.4084e-05,  4.4814e-04],
        …,
        [ 1.1301e-04,  4.8400e-04,  5.2026e-04,  …,  4.7308e-04,
          1.3642e-04,  4.1856e-04],
        [ 5.3086e-04,  1.5612e-03,  6.8125e-04,  …,  1.7094e-03,
          4.0488e-04,  1.4689e-03],
        [ 1.8768e-04,  4.8557e-04,  3.2950e-04,  …,  5.7537e-04,
          9.7106e-05,  4.3173e-04]])
tensor([ 6.2808e-04, -1.2370e-01,  5.8144e-04, -1.2483e-01,  6.7637e-04,
         1.5640e-03, -6.2235e-02, -1.8500e-01, -6.1918e-02,  6.5839e-04,
         5.9716e-04, -6.2172e-02,  1.0832e-04, -1.2107e-01,  1.7957e-03,
         1.1336e-03,  8.2513e-04, -6.1847e-02, -1.2419e-01, -6.0613e-02,
         1.4873e-03,  4.0976e-04,  4.7895e-03,  8.3320e-04,  2.4039e-03,
         8.6407e-04,  5.0556e-03,  3.1659e-04,  8.7714e-04,  1.9899e-04,
         5.8151e-04,  6.3622e-04,  2.3896e-04,  2.3771e-03,  8.6264e-04,
         5.6905e-04,  5.8998e-04,  6.1742e-04,  3.1889e-04,  4.6703e-04,
         8.1365e-04,  5.1004e-04,  1.1401e-03,  1.3973e-03,  1.5031e-03,
         1.6780e-04,  1.7503e-03,  3.1233e-04,  1.6161e-03,  6.9218e-04,
         1.7708e-03,  2.7385e-03,  1.0871e-03,  1.3488e-03,  1.6830e-03,
         7.0401e-04,  1.3189e-03,  4.2013e-04,  1.0463e-03,  4.8066e-04,
         6.2077e-04,  1.7425e-03,  1.1336e-03,  1.4162e-03,  8.1295e-04,
         1.3375e-03,  2.6595e-04,  6.4343e-04,  1.1703e-03,  9.7791e-04,
         6.0881e-04,  9.0893e-04,  1.0109e-03,  4.2205e-04,  6.3752e-04,
         1.0898e-03,  3.0180e-04,  1.1363e-03,  5.5339e-04,  7.6050e-04,
         3.3882e-04,  2.5814e-04,  1.1921e-03,  1.5884e-03,  1.7123e-03,
         8.3696e-04,  8.4844e-03,  9.2717e-04,  6.2297e-04,  2.2924e-04,
         2.1236e-04,  8.7702e-04,  8.2450e-04,  5.8969e-04,  1.3373e-03,
         3.8467e-04,  5.8803e-04,  7.0844e-04,  1.8091e-03,  3.7375e-04,
         1.0910e-03,  3.8144e-04,  8.4027e-04,  8.4538e-04,  6.8363e-04,
         6.8229e-04,  1.0810e-03,  1.5914e-03,  2.7894e-04,  5.5466e-04,
         5.4724e-04,  1.0681e-03,  7.5155e-04,  1.1043e-03,  2.8810e-03,
         1.0948e-03,  2.5072e-03,  2.4415e-04,  7.2363e-04,  7.2011e-04,
         8.9645e-04,  4.2486e-04,  4.4244e-04,  1.6512e-04,  9.6098e-04,
         3.6072e-04,  5.9807e-04,  5.6353e-04,  1.8363e-03,  2.0292e-03,
         1.5049e-03,  1.5343e-04,  1.2407e-03,  2.1743e-04,  3.0330e-03,
         6.8529e-04,  1.0661e-03,  4.3713e-04,  1.8777e-04,  3.1457e-04,
         1.2972e-03,  1.3265e-03,  8.1128e-04,  1.0541e-03,  1.2507e-03,
         8.7517e-04,  8.7249e-04,  5.6201e-04,  4.3000e-04,  2.5211e-03,
         4.3106e-04,  1.5620e-03,  5.3617e-04,  1.0942e-03,  9.5446e-04,
         1.4450e-03,  1.5102e-03,  3.5504e-04,  1.3896e-04,  6.0465e-04,
         8.6992e-04,  2.9140e-04,  7.0728e-04,  3.8607e-04,  9.8951e-04,
         3.9661e-04,  2.8989e-04,  9.9532e-04,  1.5522e-03,  8.9892e-04,
         6.6208e-04,  1.9837e-03,  8.4599e-04,  2.5734e-04,  2.6641e-03,
         8.8886e-04,  9.7113e-04,  1.6205e-03,  1.0215e-03,  6.5314e-04,
         6.4715e-04,  3.1179e-04,  3.6896e-04,  3.9552e-04,  2.6102e-04,
         2.6939e-03,  5.7872e-04,  1.8722e-03,  1.2140e-03,  4.8855e-04,
         4.7794e-04,  3.2711e-03,  3.3153e-04,  2.4101e-03,  7.7206e-04,
         6.3158e-04,  1.8118e-03,  9.6247e-04,  5.7527e-03,  1.0049e-03,
         3.2362e-03,  6.0857e-04,  3.5539e-04,  4.6623e-04,  8.0343e-04,
         1.4903e-03,  1.8810e-04,  6.1178e-04,  2.9847e-03,  6.8765e-04,
         3.3748e-04,  1.0433e-03,  1.2034e-04,  6.1181e-04,  1.4038e-03,
         7.9719e-04,  1.9955e-03,  8.3477e-04,  9.1761e-04,  1.1444e-03,
         1.7803e-03,  1.8038e-03,  1.3773e-03,  1.8295e-03,  7.0950e-04,
         5.1974e-04,  1.1415e-03,  9.4804e-04,  4.2877e-04,  2.7284e-03,
         1.4935e-03,  6.1090e-04,  5.9133e-04,  8.4015e-04,  5.5870e-04,
         1.6869e-03,  1.9639e-04,  5.8961e-04,  4.8949e-04,  2.4568e-04,
         6.3351e-04,  3.4745e-04,  6.4821e-04,  1.2262e-03,  4.1256e-04,
         9.4856e-04,  1.9402e-03,  6.5892e-04,  1.8067e-03,  2.6591e-04,
         8.0299e-04,  2.6645e-04,  1.4361e-03,  1.3245e-03,  1.6355e-03,
         1.8412e-03,  1.7798e-03,  4.2842e-04,  2.3938e-03,  1.4417e-03,
         8.1000e-04,  1.4684e-03,  3.7436e-04,  2.3294e-03,  4.9255e-04,
         3.2407e-04,  9.8282e-04,  9.7176e-04,  5.9254e-04,  8.8250e-04,
         2.6257e-04,  9.6558e-04,  4.0223e-04,  3.4196e-04,  2.2902e-04,
         9.1276e-04,  2.0610e-03,  2.3329e-04,  3.5452e-04,  3.4936e-04,
         5.2871e-03,  7.0412e-04,  3.5560e-04,  1.3141e-03,  7.3007e-04,
         6.0298e-04,  2.3909e-04,  6.4319e-04,  1.5918e-03,  2.5575e-04,
         3.3484e-04,  2.6883e-04,  1.1998e-03,  2.1489e-03,  7.7006e-04,
         7.5866e-04,  1.1636e-03,  3.6739e-04,  5.7578e-04,  2.6285e-03,
         4.8051e-04,  2.1227e-04,  5.9000e-04,  4.6777e-04,  4.8063e-04,
         1.3221e-03,  5.2516e-03,  6.8373e-04,  1.0670e-03,  3.6616e-04,
         2.9835e-04,  1.4876e-04,  5.4881e-04,  9.6996e-04,  2.4769e-03,
         8.2581e-04,  2.9064e-04,  5.7153e-04,  6.1077e-04,  3.4168e-04,
         5.0436e-04,  3.0360e-04,  1.1743e-03,  2.4356e-04,  1.3954e-03,
         1.0832e-03,  1.3218e-03,  2.3897e-03,  8.5921e-04,  2.5939e-03,
         1.2989e-03,  7.6442e-04,  9.5841e-04,  2.0884e-04,  2.5108e-04,
         2.5996e-03,  2.2976e-04,  1.0843e-03,  5.8399e-04,  8.6641e-04,
         7.7804e-04,  1.0202e-03,  1.1197e-03,  1.5832e-03,  2.8464e-04,
         7.7201e-04,  6.2819e-04,  2.2477e-03,  1.2628e-03,  9.8795e-04,
         2.9966e-04,  5.0742e-04,  1.3799e-03,  8.8207e-04,  5.9498e-04,
         1.7365e-04,  2.0420e-03,  5.0059e-04,  5.1354e-03,  1.0138e-03,
         1.0005e-03,  2.3807e-03,  3.9287e-04,  4.7880e-04,  1.3765e-03,
         1.3622e-03,  3.0588e-04,  9.7838e-04,  3.2082e-04,  9.0527e-04,
         1.8959e-03,  4.3093e-03,  2.9390e-04,  9.5946e-04,  9.0660e-04,
         1.6312e-03,  5.2874e-04,  3.6262e-04,  2.8550e-03,  7.2147e-04,
         4.3935e-04,  1.1717e-03,  1.3307e-03,  4.0984e-04,  7.1452e-04,
         8.5463e-04,  2.3426e-04,  5.0433e-04,  8.2653e-04,  8.9171e-04,
         1.2075e-03,  2.0337e-04,  2.3007e-03,  1.8149e-04,  7.9112e-04,
         3.1713e-04,  1.6185e-03,  1.5609e-04,  2.4920e-04,  1.4732e-03,
         3.3405e-04,  3.4608e-04,  2.7670e-03,  9.3166e-04,  3.7762e-04,
         8.6814e-04,  4.5219e-04,  2.5261e-03,  3.9962e-04,  3.9975e-04,
         2.5631e-04,  5.2159e-04,  1.3573e-03,  2.6060e-04,  9.0227e-04,
         3.3197e-04,  9.0662e-04,  1.6461e-04,  1.1604e-03,  3.7855e-04,
         4.0211e-04,  4.8214e-04,  7.7519e-04,  4.6769e-04,  6.2913e-04,
         5.3643e-04,  1.6242e-03,  5.0028e-04,  8.1578e-04,  8.7193e-04,
         1.2792e-03,  3.2606e-04,  6.8047e-04,  3.0256e-03,  2.0714e-04,
         1.3907e-03,  1.2190e-03,  9.7578e-04,  1.1830e-03,  1.0673e-03,
         4.7041e-04,  1.5385e-03,  1.6048e-03,  1.2078e-03,  7.4793e-04,
         7.0391e-04,  5.0785e-04,  5.9376e-04,  5.1717e-04,  7.6377e-04,
         4.9437e-04,  4.8253e-04,  5.5349e-04,  6.9169e-04,  8.4291e-04,
         4.2549e-04,  3.2520e-03,  1.4577e-03,  7.4606e-04,  3.4326e-04,
         3.1852e-03,  7.7020e-04,  1.2130e-03,  1.5808e-03,  1.3542e-03,
         1.9986e-04,  5.3089e-04,  3.8118e-03,  5.8662e-04,  2.7139e-04,
         4.7103e-04,  1.4651e-03,  4.9464e-04,  7.4713e-04,  2.6870e-04,
         4.0238e-04,  6.5865e-04,  1.5116e-04,  1.4138e-03,  1.4022e-03,
         7.1995e-04,  1.9606e-04,  8.4813e-04,  1.6061e-04,  3.2283e-04,
         1.6162e-03,  4.5153e-04,  1.2100e-03,  1.1554e-03,  1.1843e-03,
         1.3349e-03,  7.6254e-04,  1.0787e-03,  6.0308e-04,  4.5272e-04,
         1.6574e-03,  1.0609e-03,  9.8932e-04,  3.6124e-04,  6.6474e-04,
         4.6314e-04,  6.1141e-03,  7.2823e-04,  4.4819e-04,  7.5577e-04,
         4.0603e-04,  2.9197e-04,  1.8709e-03,  1.2279e-03,  6.7410e-04,
         5.8403e-04,  4.3323e-03,  5.0231e-04,  8.0997e-04,  4.5601e-04,
         1.4951e-03,  2.7376e-04,  2.4572e-04,  1.1317e-03,  6.6020e-04,
         3.9990e-04,  6.3069e-04,  2.2032e-03,  1.5147e-03,  7.6362e-04,
         3.2955e-03,  6.3879e-04,  1.4815e-03,  1.9007e-03,  1.1967e-03,
         8.2497e-04,  2.6085e-03,  3.1667e-04,  6.4876e-04,  1.5058e-03,
         8.0815e-04,  3.8085e-04,  1.2203e-03,  7.2405e-04,  9.8791e-04,
         1.5427e-03,  6.2055e-04,  3.5046e-03,  4.7905e-04,  2.9893e-04,
         9.7819e-04,  8.9174e-04,  1.0068e-03,  2.0843e-04,  5.4186e-04,
         1.6387e-03,  8.3952e-04,  9.6065e-04,  4.9415e-04,  5.5538e-04,
         3.7910e-04,  2.0631e-03,  5.3112e-04,  3.5054e-04,  3.9495e-04,
         8.7847e-04,  8.3199e-04,  2.3017e-03,  1.5683e-03,  1.2518e-03,
         5.7265e-04,  3.2472e-04,  1.1482e-03,  1.0727e-03,  3.4314e-03,
         8.1570e-04,  2.3877e-03,  4.3796e-04,  1.0139e-03,  1.0553e-03,
         3.9616e-04,  1.0683e-03,  1.4066e-03,  1.5399e-04,  2.8133e-04,
         8.9603e-04,  7.7193e-04,  8.8645e-04,  3.8920e-04,  8.4505e-04,
         9.9061e-04,  9.4459e-04,  1.7954e-03,  9.9639e-04,  9.2096e-04,
         6.3648e-04,  6.4652e-04,  6.1068e-04,  6.2793e-04,  1.1737e-03,
         1.2138e-03,  2.3395e-04,  3.6302e-03,  1.9281e-04,  3.6128e-04,
         4.0320e-04,  1.0911e-03,  2.2535e-03,  1.2874e-03,  1.5104e-03,
         1.1560e-03,  1.2931e-03,  2.3216e-04,  5.9814e-04,  5.6071e-04,
         6.8805e-04,  2.9388e-04,  8.0498e-04,  1.6778e-03,  5.0581e-04,
         5.0814e-04,  6.1910e-04,  9.6094e-04,  8.7882e-04,  5.3286e-04,
         5.2086e-03,  8.9286e-04,  6.9072e-04,  9.9792e-04,  3.6108e-04,
         3.0921e-04,  2.6886e-03,  1.8231e-03,  9.0845e-04,  4.3651e-04,
         1.0429e-03,  5.9476e-04,  6.1594e-04,  1.4148e-03,  1.6773e-03,
         1.0035e-03,  1.3550e-03,  3.6171e-04,  2.4137e-04,  5.5536e-04,
         6.6162e-04,  7.4198e-04,  1.6224e-03,  1.8891e-03,  4.0234e-04,
         3.8175e-04,  7.4437e-04,  6.6010e-04,  2.8905e-03,  5.9831e-04,
         6.5237e-04,  2.3589e-04,  2.4360e-03,  7.2506e-04,  4.9351e-04,
         5.5636e-04,  2.9992e-04,  2.5168e-04,  7.5286e-04,  1.9772e-04,
         3.7144e-04,  4.8625e-04,  2.7131e-04,  9.7736e-04,  3.2002e-04,
         3.9736e-04,  1.5073e-04,  2.3888e-03,  3.2704e-04,  9.2391e-04,
         2.3393e-03,  1.0046e-03,  1.1888e-03,  2.5653e-03,  1.2569e-03,
         9.7879e-04,  1.2961e-03,  8.1196e-04,  3.3616e-04,  7.8645e-04,
         8.0159e-04,  1.2667e-03,  5.7445e-04,  8.7498e-04,  3.5027e-03,
         5.3998e-04,  4.7220e-04,  5.1535e-04,  1.0021e-03,  2.2831e-03,
         4.9470e-04,  1.1851e-03,  3.0774e-04,  2.8344e-04,  4.3135e-03,
         7.1514e-04,  8.9134e-04,  1.1455e-03,  5.2361e-04,  4.8078e-04,
         6.2271e-04,  1.5871e-03,  1.5234e-03,  9.0189e-04,  1.4291e-03,
         5.2062e-04,  7.2476e-04,  7.0335e-04,  1.3956e-03,  1.7249e-03,
         3.9126e-04,  6.7501e-04,  1.1058e-03,  1.3647e-04,  5.7257e-04,
         8.7383e-04,  4.5980e-04,  2.2872e-04,  3.5070e-04,  1.4704e-03,
         6.2461e-04,  2.5426e-04,  5.0601e-04,  1.0808e-03,  2.9238e-03,
         3.7406e-04,  7.6552e-04,  7.9050e-04,  5.0680e-04,  9.1444e-04,
         1.5055e-03,  9.9882e-04,  4.9932e-04,  8.5060e-04,  1.1078e-03,
         5.4648e-04,  1.3369e-03,  3.4398e-04,  4.4872e-04,  6.4464e-04,
         5.2131e-04,  3.5592e-04,  8.9601e-04,  9.4685e-04,  6.9117e-04,
         6.1475e-04,  4.5789e-04,  3.0895e-04,  4.2892e-04,  4.4948e-04,
         1.5251e-03,  1.0041e-03,  6.9514e-04,  8.7107e-04,  5.9901e-04,
         1.2998e-03,  1.7968e-04,  2.1038e-04,  8.1374e-04,  6.5002e-04,
         3.2356e-04,  3.3453e-04,  7.4257e-04,  2.9631e-04,  1.1797e-03,
         1.3494e-03,  3.7498e-04,  1.2263e-04,  1.4943e-03,  9.8839e-04,
         5.2167e-04,  3.9190e-04,  2.1812e-04,  6.2187e-04,  1.8017e-03,
         5.1248e-04,  3.2089e-04,  3.0013e-04,  7.2742e-04,  4.1205e-04,
         1.8873e-03,  1.4838e-03,  1.3111e-03,  1.0818e-03,  5.2070e-04,
         1.6254e-03,  7.4639e-04,  7.9942e-04,  9.5873e-04,  1.2497e-03,
         8.2992e-05,  8.6849e-04,  2.3509e-04,  8.2134e-04,  2.9887e-03,
         2.3837e-03,  4.7169e-04,  1.6932e-04,  3.8406e-04,  1.8987e-03,
         5.2310e-04,  1.0699e-03,  2.9837e-04,  6.1013e-04,  2.3067e-04,
         3.6834e-04,  1.5889e-03,  5.3714e-04,  7.6610e-04,  1.6425e-04,
         9.1353e-04,  1.5993e-03,  1.0456e-03,  5.9859e-04,  5.6500e-04,
         6.8133e-04,  8.3906e-04,  6.4239e-04,  3.1528e-03,  3.0232e-04,
         4.4304e-04,  6.2174e-04,  5.5175e-04,  4.3439e-04,  1.3655e-03,
         2.0773e-03,  6.6552e-04,  3.6184e-04,  1.0140e-03,  4.8142e-04,
         1.1299e-03,  5.5363e-04,  4.2753e-04,  9.0768e-04,  1.7015e-03,
         8.5027e-04,  4.6057e-04,  8.4938e-04,  3.6773e-04,  1.1050e-03,
         1.8920e-03,  8.7556e-04,  7.5328e-04,  2.1328e-03,  1.8737e-03,
         1.7448e-03,  2.1523e-03,  1.0530e-03,  7.4138e-04,  5.6896e-04,
         4.7729e-04,  5.4536e-04,  1.8914e-03,  2.6022e-04,  7.9050e-04,
         3.5844e-04,  1.7855e-03,  5.3980e-04,  1.2103e-04,  1.4021e-03,
         7.9479e-04,  2.1178e-04,  1.8770e-03,  7.3505e-04,  4.7775e-03,
         2.5689e-04,  1.0876e-03,  2.1071e-04,  8.5528e-04,  1.3903e-03,
         1.1739e-03,  2.2280e-03,  1.2326e-03,  1.8111e-04,  4.1502e-04,
         3.7826e-04,  1.6822e-04,  7.6320e-04,  8.4979e-04,  5.9273e-04,
         3.2191e-04,  4.9450e-04,  1.3579e-03,  6.0118e-04,  4.7245e-04,
         8.7121e-04,  9.4072e-04,  5.4796e-04,  7.8756e-04,  1.6755e-04,
         3.0750e-03,  1.1253e-03,  2.9434e-03,  8.7150e-04,  1.7802e-03,
         7.0338e-04,  6.1657e-04,  4.2093e-04,  1.5193e-03,  1.1593e-03,
         6.0905e-04,  5.2436e-04,  2.9305e-04,  4.5757e-04,  8.2649e-04,
         7.2828e-04,  2.8958e-04,  1.1794e-03,  5.9505e-04,  9.8521e-04,
         2.5082e-04,  6.7129e-04,  1.7328e-03,  4.5162e-03,  2.5766e-03,
         4.4252e-04,  8.1609e-04,  3.1964e-03,  5.0038e-04,  3.8979e-04,
         4.4875e-04,  9.0313e-04,  6.1620e-04,  7.2215e-04,  6.3170e-04,
         3.4160e-04,  4.3995e-04,  8.4801e-04,  3.7187e-04,  1.7067e-03,
         7.4184e-04,  6.1266e-04,  8.7126e-04,  1.7655e-03,  7.0142e-04,
         3.1092e-04,  1.3067e-03,  1.0552e-03,  9.3681e-04,  1.2916e-03,
         7.5779e-04,  5.1611e-04,  6.6998e-04,  6.0556e-04,  9.7016e-04,
         1.9007e-03,  2.4535e-03,  1.5733e-04,  5.1272e-04,  4.3365e-04,
         1.3749e-03,  1.0981e-03,  5.7933e-04,  1.0881e-03,  2.1924e-04,
         1.9047e-03,  7.7182e-04,  6.8598e-04,  1.0495e-03,  2.3208e-03,
         1.0183e-03,  5.6721e-04,  3.0849e-04,  6.8286e-04,  1.2439e-04,
         1.4320e-03,  4.0944e-03,  6.5377e-04,  1.6620e-04,  8.5559e-04,
         1.5269e-04,  1.0664e-03,  9.6491e-04,  1.8869e-03,  5.5884e-04,
         4.1387e-04,  1.2024e-03,  2.3907e-03,  2.3719e-03,  2.0631e-04,
         4.4545e-03,  2.9034e-04,  2.8386e-03,  1.5761e-03,  3.3299e-04,
                  1.1910e-03,  8.5258e-04,  1.5932e-03,  6.1843e-04,  1.5049e-03,
                  5.5548e-04,  2.6360e-04,  2.8434e-03,  6.9971e-04,  1.4907e-03,
                  5.1436e-03,  2.8670e-03,  5.7873e-04,  1.6789e-03,  5.3531e-04])
         end of p.grad

         False
         Epoch 1 finished
         Epoch [1/10], Loss: 6.8876
         **************************************************************
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[12], line 26
     23 for inputs, labels in dataloader:
     25     optimizer.zero_grad()
---> 26     outputs = model(inputs)
     27     Before = list(model.parameters())[0].clone()
     29     loss = criterion(outputs, labels)

File C:\Apps installed by␣
 ↪Lim\anaconda3\Lib\site-packages\torch\nn\modules\module.py:1511, in Module.
 ↪_wrapped_call_impl(self, *args, **kwargs)
   1509     return self._compiled_call_impl(*args, **kwargs)  # type:␣
 ↪ignore[misc]
   1510 else:
-> 1511     return self._call_impl(*args, **kwargs)

File C:\Apps installed by␣
 ↪Lim\anaconda3\Lib\site-packages\torch\nn\modules\module.py:1520, in Module.
 ↪_call_impl(self, *args, **kwargs)
   1515 # If we don't have any hooks, we want to skip the rest of the logic in
   1516 # this function, and just call forward.
   1517 if not (self._backward_hooks or self._backward_pre_hooks or self.
 ↪_forward_hooks or self._forward_pre_hooks
   1518         or _global_backward_pre_hooks or _global_backward_hooks
   1519         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1520     return forward_call(*args, **kwargs)
   1522 try:
   1523     result = None

Cell In[2], line 34, in AlexNet.forward(self, x)
     33 def forward(self, x: torch.Tensor) -> torch.Tensor:
---> 34     x = self.features(x)
     35     x = self.avgpool(x)
     36     x = torch.flatten(x, 1)
```

```
File C:\Apps installed by␣
  ↪Lim\anaconda3\Lib\site-packages\torch\nn\modules\module.py:1511, in Module.
  ↪_wrapped_call_impl(self, *args, **kwargs)
   1509     return self._compiled_call_impl(*args, **kwargs)  # type:␣
  ↪ignore[misc]
   1510 else:
-> 1511     return self._call_impl(*args, **kwargs)

File C:\Apps installed by␣
  ↪Lim\anaconda3\Lib\site-packages\torch\nn\modules\module.py:1520, in Module.
  ↪_call_impl(self, *args, **kwargs)
   1515 # If we don't have any hooks, we want to skip the rest of the logic in
   1516 # this function, and just call forward.
   1517 if not (self._backward_hooks or self._backward_pre_hooks or self.
  ↪_forward_hooks or self._forward_pre_hooks
   1518         or _global_backward_pre_hooks or _global_backward_hooks
   1519         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1520     return forward_call(*args, **kwargs)
   1522 try:
   1523     result = None

File C:\Apps installed by␣
  ↪Lim\anaconda3\Lib\site-packages\torch\nn\modules\container.py:217, in␣
  ↪Sequential.forward(self, input)
    215 def forward(self, input):
    216     for module in self:
--> 217         input = module(input)
    218     return input

File C:\Apps installed by␣
  ↪Lim\anaconda3\Lib\site-packages\torch\nn\modules\module.py:1511, in Module.
  ↪_wrapped_call_impl(self, *args, **kwargs)
   1509     return self._compiled_call_impl(*args, **kwargs)  # type:␣
  ↪ignore[misc]
   1510 else:
-> 1511     return self._call_impl(*args, **kwargs)

File C:\Apps installed by␣
  ↪Lim\anaconda3\Lib\site-packages\torch\nn\modules\module.py:1520, in Module.
  ↪_call_impl(self, *args, **kwargs)
   1515 # If we don't have any hooks, we want to skip the rest of the logic in
   1516 # this function, and just call forward.
   1517 if not (self._backward_hooks or self._backward_pre_hooks or self.
  ↪_forward_hooks or self._forward_pre_hooks
   1518         or _global_backward_pre_hooks or _global_backward_hooks
   1519         or _global_forward_hooks or _global_forward_pre_hooks):
-> 1520     return forward_call(*args, **kwargs)
   1522 try:
   1523     result = None
```

```
File C:\Apps installed by Lim\anaconda3\Lib\site-packages\torch\nn\modules\conv
 ↪py:460, in Conv2d.forward(self, input)
    459 def forward(self, input: Tensor) -> Tensor:
--> 460     return self._conv_forward(input, self.weight, self.bias)

File C:\Apps installed by Lim\anaconda3\Lib\site-packages\torch\nn\modules\conv
 ↪py:456, in Conv2d._conv_forward(self, input, weight, bias)
    452 if self.padding_mode != 'zeros':
    453     return F.conv2d(F.pad(input, self._reversed_padding_repeated_twice,
 ↪mode=self.padding_mode),
    454                     weight, bias, self.stride,
    455                     _pair(0), self.dilation, self.groups)
--> 456 return F.conv2d(input, weight, bias, self.stride,
    457                 self.padding, self.dilation, self.groups)

KeyboardInterrupt:
```

[ ]:

```python
# Plot loss and gradient magnitudes
plt.plot(losses, label='Loss')
plt.xlabel('Epoch')
plt.ylabel('Value')
plt.legend()
plt.show()
```

[ ]:

```python
labels_path = '/Users/Limit/imagenet_annot/validation_set_labels.csv'

labels_df = pd.read_csv(labels_path)

labels_df_leq_20 = labels_df[labels_df['label'] <= 20]

labels_validation_images = labels_df_leq_20['label'].tolist()
```

[ ]:

```python
len(labels_df_leq_20['ImageId'].tolist())
model.eval()
```

[ ]:

```python
image_path = "/Users/Limit/imagenet-object-localization-challenge_validation/
 ↪val"
filenames_image_path = []
for root, _, filenames in os.walk(image_path):
    for i in filenames:
#         print(i)
        if (i.split('.')[0] in labels_df_leq_20['ImageId'].tolist()):
            filenames_image_path.append(i)
true_label = 0
counter = 0
correct_labels = 0
```

```
start_time = time.time()
grab_980_max_val = []
for i in range(len(filenames_image_path)):
    counter +=1
#     print(i)
#     print(counter)
    image_name = image_path + '/' + filenames_image_path[i]
    # black and white
#     image_name = "/Users/Limit/imagenet-object-localization-challenge/
 ↪n01440764/n01440764_15560.JPEG"
    input_image = Image.open(image_name)



    input_tensor = preprocess(input_image)
    input_batch = input_tensor.unsqueeze(0) # create a mini-batch as expected␣
 ↪by the model

    # move the input and model to GPU for speed if available
    if torch.cuda.is_available():
        input_batch = input_batch.to('cuda')
        model.to('cuda')

    if (counter%100 == 0):
        print("currently at", counter, 'current time is', time.time() -␣
 ↪start_time)
    with torch.no_grad():
        output = model(input_batch)
    # Tensor of shape 1000, with confidence scores over ImageNet's 1000 classes
#     if (torch.argmax(output[0]).item() == true_label):
#         correct_labels += 1
   ␣
 ↪print('********************************************************************************')
    print('Predicting Test Sample', counter, ':   Prediction is Correct?')
    if (torch.argmax(output[0]).item() == labels_validation_images[i]):
        print('Yes')
        correct_labels += 1
    else:
        print('No')
    prob_softmax = torch.softmax(output[0], dim = 0)
    print("first 20 classes probability:", prob_softmax[:20])
    print()
    print('max probability is', torch.max(prob_softmax, dim = 0))
    print()
    print('the max probability of the rest of 980 dim is')
    print(torch.topk(prob_softmax[20:], k=4))
```

```
  ␣
↪print('End*****************************************************************************')
    print()

    grab_980_max_val.append(torch.topk(prob_softmax[20:], k=4)[1][1:])
    # The output has unnormalized scores. To get probabilities, you can run a␣
↪softmax on it.
    probabilities = torch.nn.functional.softmax(output[0], dim=0)
    # print(probabilities)

print('the overall testing error is')
print(correct_labels/counter)
```

```
[ ]: sum(grab_980_max_val)/counter
```

```
[ ]: counter
```

```
[ ]: # OLD code




# model_alex_given.eval()

# for i in range(40):
#     input_batch = input_tensor[i].unsqueeze(0) # create a mini-batch as␣
↪expected by the model

#     prediction = model_alex_given(input_batch)
#     print('prediction:')
#     print(torch.argmax(prediction))
#     print("answer:")
# #     print(labels_numerical[i])
#     print()
```

```
[ ]: torch.save(model.state_dict(), 'model_weights_alexnet_temp_temp_temp.pth')
```

```
[ ]: # filenames_image_path
```