```
# Install PyTorch Geometric dependencies
!pip install torch torchvision torchaudio
!pip install torch-geometric
!pip install imageio matplotlib
```

```
      Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
    Attempting uninstall: nvidia-cuda-cupti-cu12
      Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
      Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
    Attempting uninstall: nvidia-cublas-cu12
      Found existing installation: nvidia-cublas-cu12 12.5.3.2
      Uninstalling nvidia-cublas-cu12-12.5.3.2:
        Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
    Attempting uninstall: nvidia-cusparse-cu12
      Found existing installation: nvidia-cusparse-cu12 12.5.1.3
      Uninstalling nvidia-cusparse-cu12-12.5.1.3:
        Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
    Attempting uninstall: nvidia-cudnn-cu12
      Found existing installation: nvidia-cudnn-cu12 9.3.0.75
      Uninstalling nvidia-cudnn-cu12-9.3.0.75:
        Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
    Attempting uninstall: nvidia-cusolver-cu12
      Found existing installation: nvidia-cusolver-cu12 11.6.3.83
      Uninstalling nvidia-cusolver-cu12-11.6.3.83:
        Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-ru
Collecting torch-geometric
  Downloading torch_geometric-2.6.1-py3-none-any.whl.metadata (63 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 63.1/63.1 kB 3.3 MB/s eta 0:00:00
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (3.11.15)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (2025.3.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (3.1.6)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (2.0.2)
Requirement already satisfied: psutil>=5.8.0 in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (5.9.5)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (3.2.3)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from torch-geometric) (4.67.1)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->torch-geometric)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->torch-geometric) (1.3.2
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->torch-geometric) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->torch-geometric) (1.6.
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->torch-geometric) (6.
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->torch-geometric) (0.3.1
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->torch-geometric) (1.20
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->torch-geometric) (3.0.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->torch-geometri
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->torch-geometric) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->torch-geometric) (2.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->torch-geometric) (20
Downloading torch_geometric-2.6.1-py3-none-any.whl (1.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 28.7 MB/s eta 0:00:00
Installing collected packages: torch-geometric
Successfully installed torch-geometric-2.6.1
Requirement already satisfied: imageio in /usr/local/lib/python3.11/dist-packages (2.37.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from imageio) (2.0.2)
Requirement already satisfied: pillow>=8.3.2 in /usr/local/lib/python3.11/dist-packages (from imageio) (11.2.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.58.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
```

```
from google.colab import files
import os

uploaded = files.upload()
gif_path = next(iter(uploaded))
```

Choose files S2L2A-126…-timelapse.gif

- **S2L2A-1268084853261429-timelapse.gif**(image/gif) - 41640957 bytes, last modified: 15/06/2025 - 100% done

```
Saving S2L2A-1268084853261429-timelapse.gif to S2L2A-1268084853261429-timelapse.gif
```

```
import imageio.v2 as imageio
from PIL import Image
import numpy as np

frames_dir = "frames"
os.makedirs(frames_dir, exist_ok=True)

reader = imageio.get_reader(gif_path)
frames = []
```

```
for i, frame in enumerate(reader):
    frame = Image.fromarray(frame).convert("RGB").resize((128, 128))
    frame_path = os.path.join(frames_dir, f"frame_{i:03d}.jpg")
    frame.save(frame_path)
    frames.append(np.array(frame))

print(f"Extracted {len(frames)} frames.")
```

```
Extracted 169 frames.
```

```
import torch
from torchvision import transforms, models

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

transform = transforms.Compose([
    transforms.ToPILImage(),
    transforms.Resize((128, 128)),
    transforms.ToTensor()
])

images = torch.stack([transform(frame) for frame in frames]).to(device)

# Pretrained CNN
cnn = models.resnet18(pretrained=True)
cnn.fc = torch.nn.Identity()
cnn = cnn.to(device).eval()

with torch.no_grad():
    features = torch.stack([cnn(img.unsqueeze(0)).squeeze(0) for img in images])  # [T, 512]
```

```
/usr/local/lib/python3.11/dist-packages/torchvision/models/_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated sinc
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or `None`
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/resnet18-f37072fd.pth" to /root/.cache/torch/hub/checkpoints/resnet18-f37072fd.pth
100%|██████████| 44.7M/44.7M [00:00<00:00, 105MB/s]
```

```
from torch_geometric.data import Data
import torch_geometric

T = features.shape[0]
edges = []
for t in range(T - 1):
    edges.append([t, t+1])
    edges.append([t+1, t])
edge_index = torch.tensor(edges).t().contiguous().to(device)
```

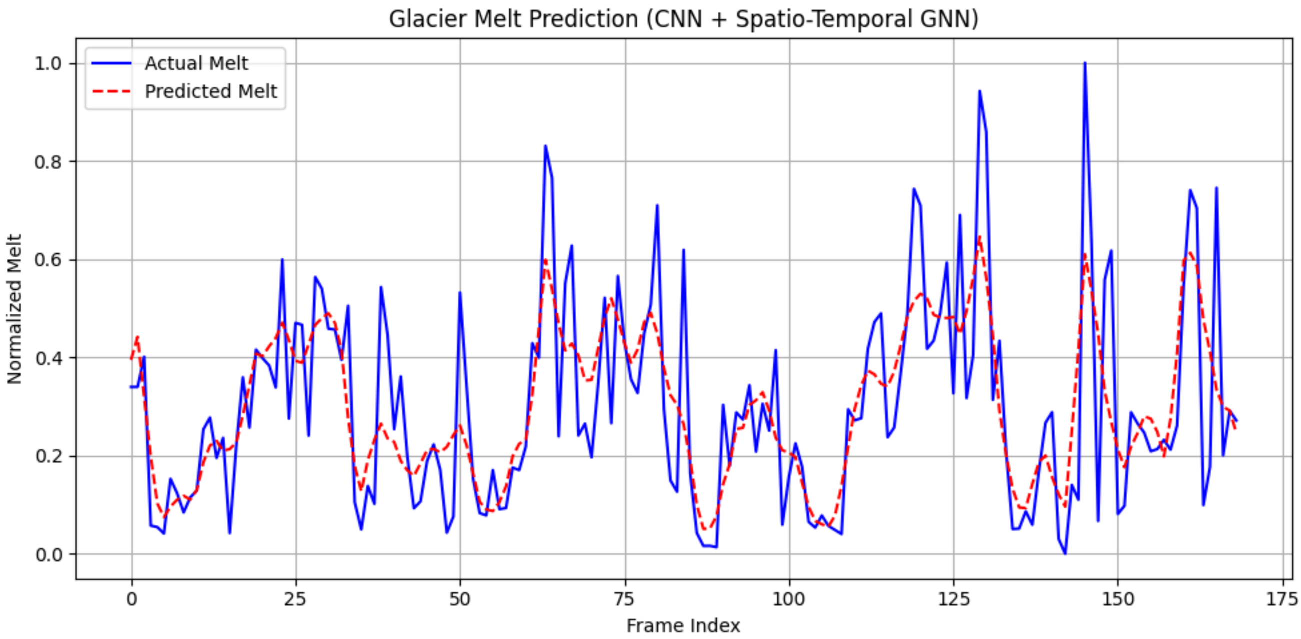
```
pixel_diffs = []
for i in range(1, len(frames)):
    diff = np.mean(np.abs(frames[i].astype(float) - frames[i-1].astype(float)))
    pixel_diffs.append(diff)
pixel_diffs = np.array(pixel_diffs)
pixel_diffs = (pixel_diffs - pixel_diffs.min()) / (pixel_diffs.max() - pixel_diffs.min())
melt_values = np.insert(pixel_diffs, 0, pixel_diffs[0])  # same length as frames
melt_values = torch.tensor(melt_values, dtype=torch.float).to(device)
```

```
from torch.nn import functional as F
from torch_geometric.nn import GCNConv
import torch.nn as nn

class STGNN(nn.Module):
    def __init__(self, in_channels, hidden_channels):
        super().__init__()
        self.gcn1 = GCNConv(in_channels, hidden_channels)
        self.gcn2 = GCNConv(hidden_channels, 1)

    def forward(self, x, edge_index):
        x = F.relu(self.gcn1(x, edge_index))
        x = self.gcn2(x, edge_index)
        return x.squeeze()

model = STGNN(512, 64).to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=0.005)
```

```
loss_fn = nn.MSELoss()


for epoch in range(300):
    model.train()
    optimizer.zero_grad()
    out = model(features, edge_index)
    loss = loss_fn(out, melt_values)
    loss.backward()
    optimizer.step()
    if epoch % 50 == 0:
        print(f"Epoch {epoch} | Loss: {loss.item():.4f}")
```

```
Epoch 0 | Loss: 2.0586
Epoch 50 | Loss: 0.0294
Epoch 100 | Loss: 0.0244
Epoch 150 | Loss: 0.0220
Epoch 200 | Loss: 0.0203
Epoch 250 | Loss: 0.0190
```

```
import matplotlib.pyplot as plt

model.eval()
pred = model(features, edge_index).detach().cpu().numpy()
true = melt_values.cpu().numpy()

plt.figure(figsize=(10, 5))
plt.plot(true, label="Actual Melt", color="blue")
plt.plot(pred, label="Predicted Melt", color="red", linestyle="--")
plt.xlabel("Frame Index")
plt.ylabel("Normalized Melt")
plt.title("Glacier Melt Prediction (CNN + Spatio-Temporal GNN)")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



+ Code + Text

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

mae = mean_absolute_error(true, pred)
rmse = np.sqrt(mean_squared_error(true, pred))
r2 = r2_score(true, pred)

print(f"MAE: {mae:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"R² Score: {r2:.4f}")
```

```
MAE: 0.0993
RMSE: 0.1341
R² Score: 0.5895
```