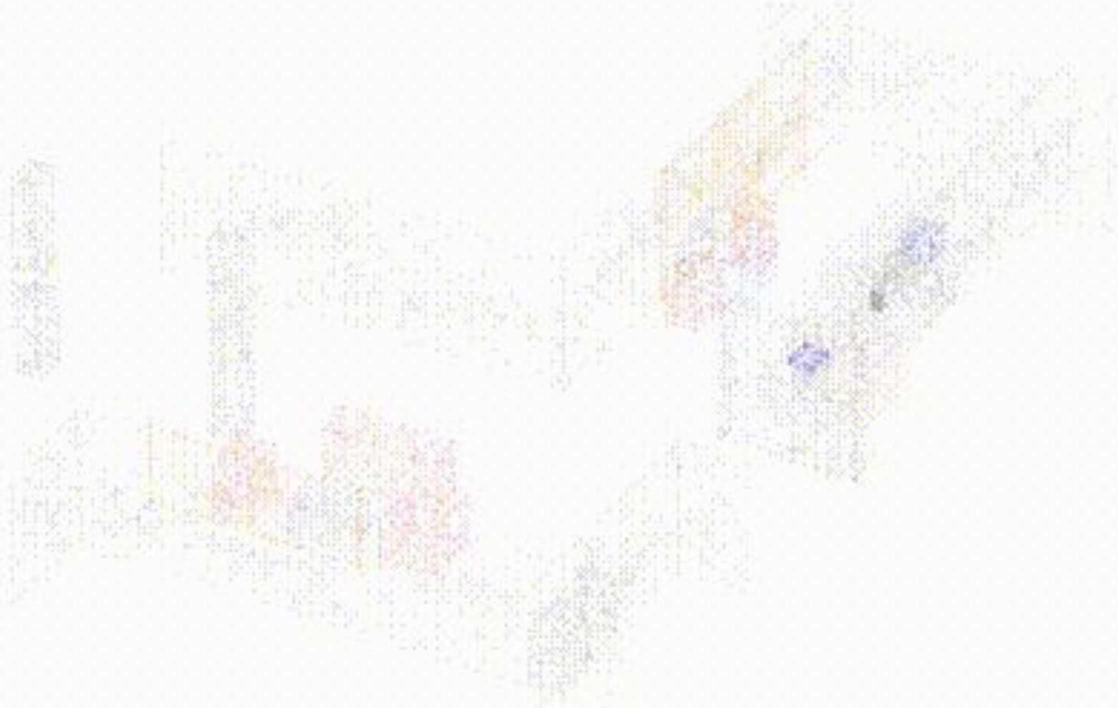
Human Pose Estimation

Davide Mattioli, Pablo Jahnen, Kaisar Dauletbek

Introduction

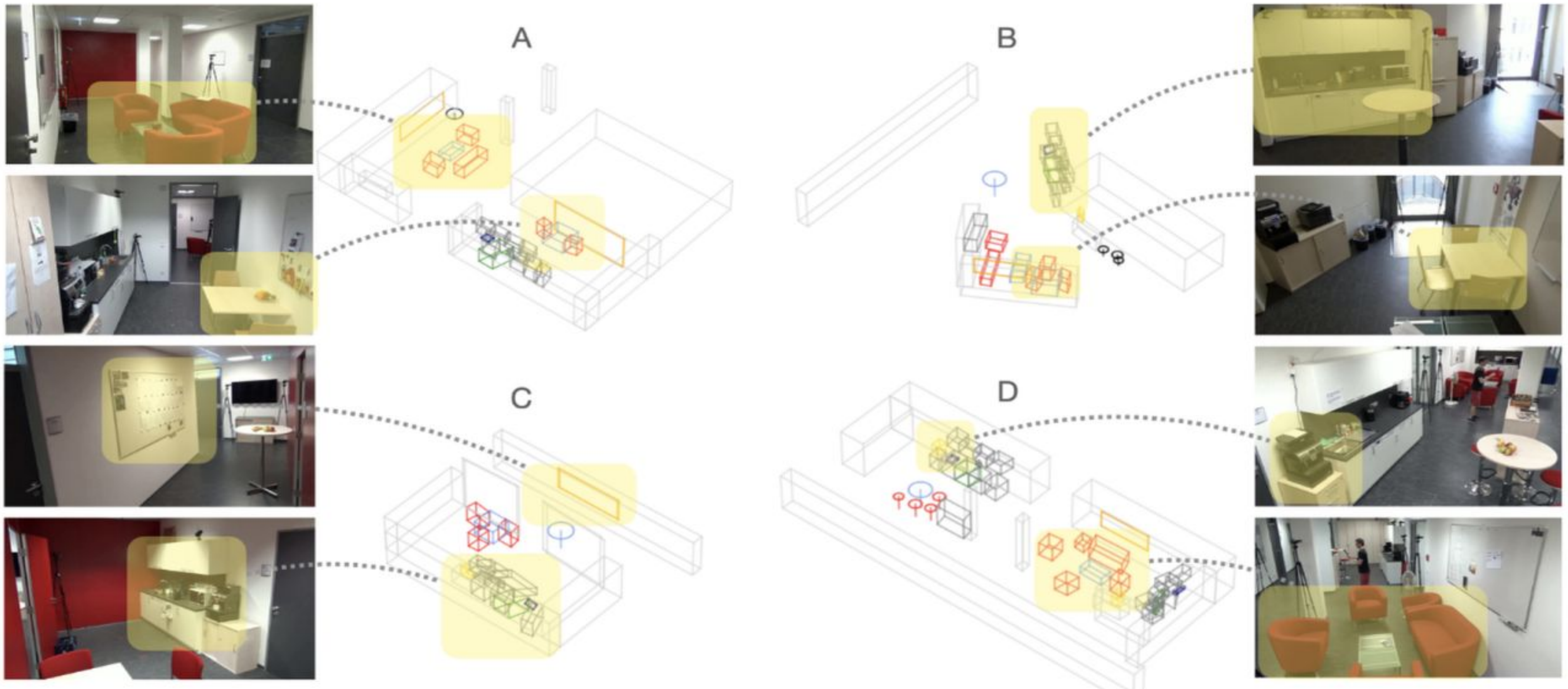
- Objective: Prediction of Poses through graph classification
- Dataset: Humans in kitchens - Realistic behaviour of multiple interacting persons. Data collected in four real office kitchens for a duration between 1.5h and 1.9h
- Graph building: Using 29 keypoint locations that map to the body (nodes), and connecting spatial and temporal neighbors (edges)
- Target: Based on a X frame sequence, we want to predict the activity label of the center frame



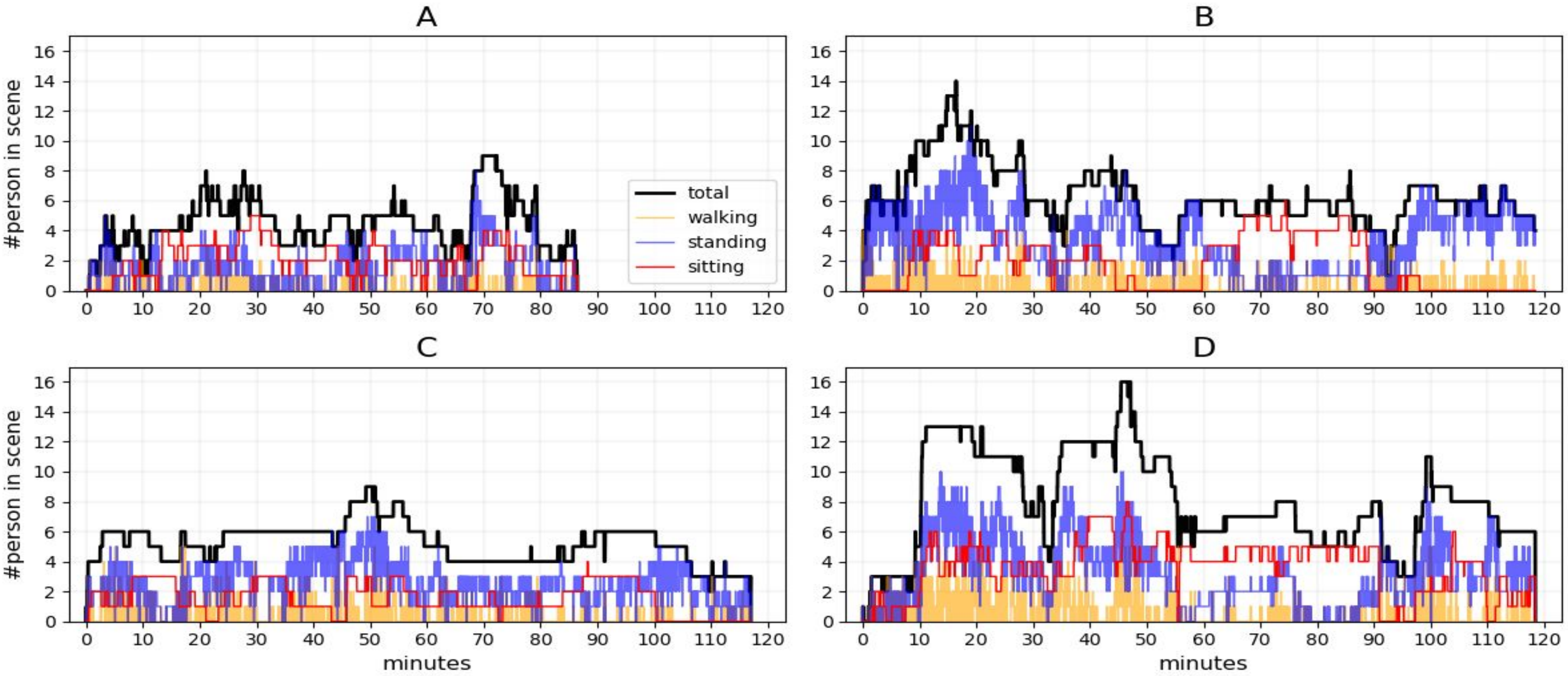
The Dataset

- 4M Annotated Poses of 90 Individuals
- 4 Dataset of different parts of the kitchen

Dataset Regions

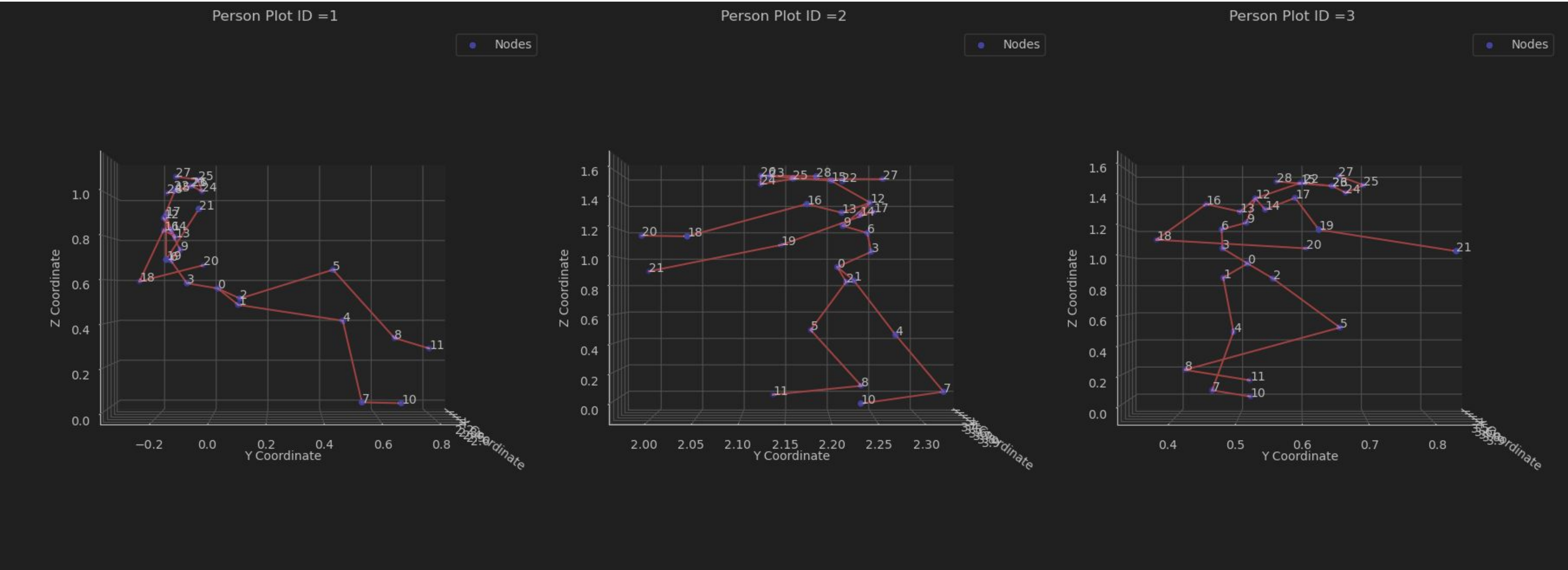


Different Region leads to Different Distributions

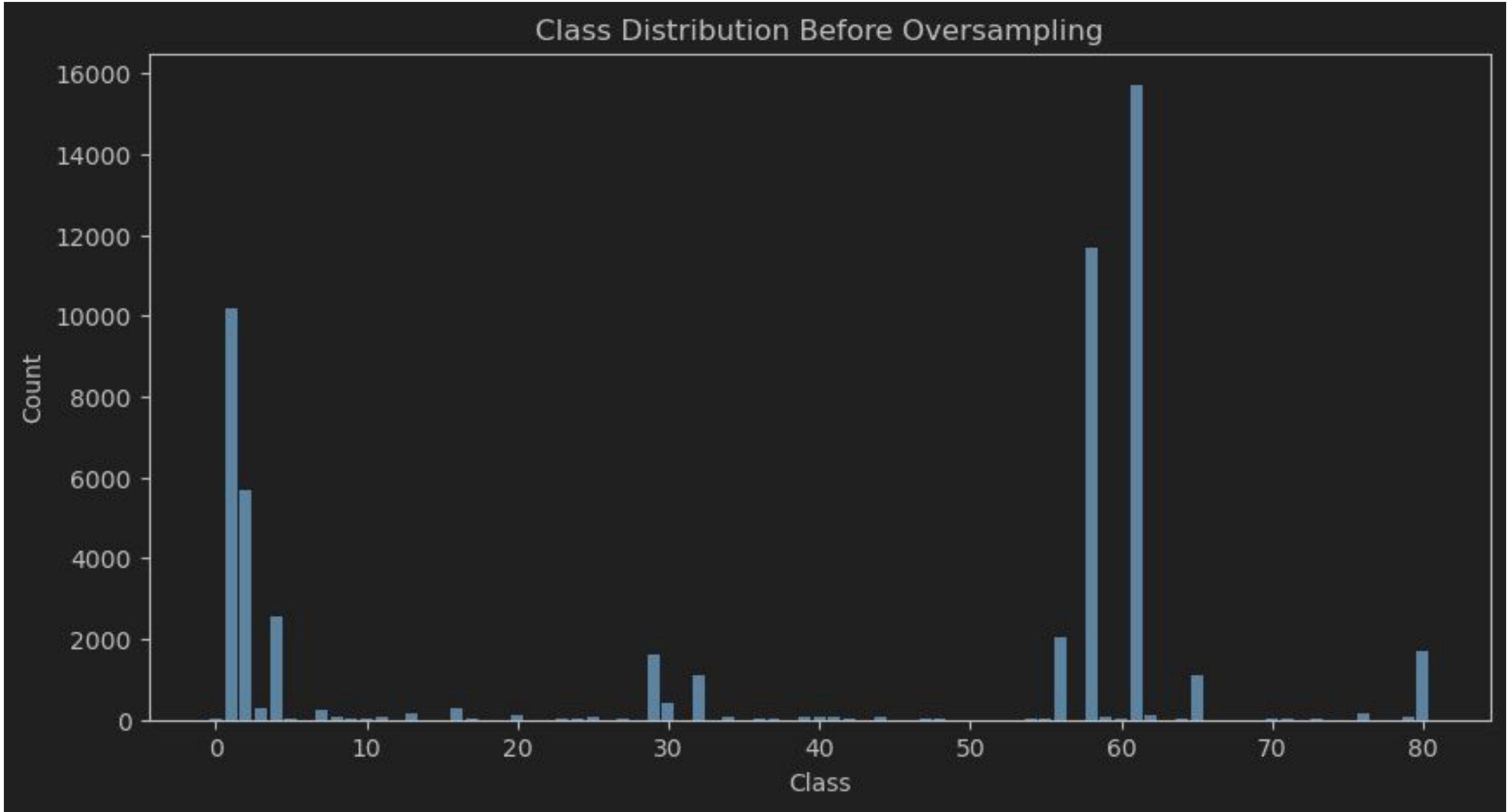


Poses Examples

Each node has (X,Y,Z) coordinates



Class Distribution



Architectures

GCM

GINE

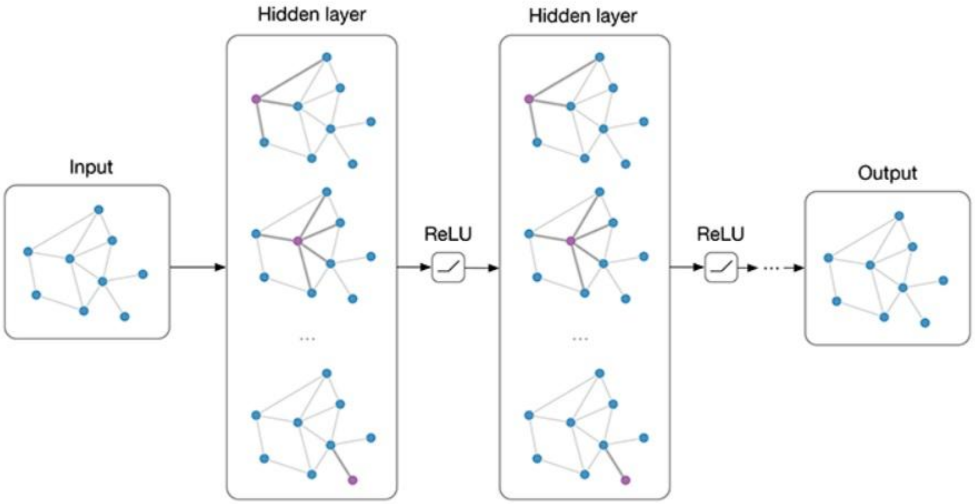
GINE with edge features

Graph Transformer

GCNConv Layer

Message passing: Combines node features and normalizes by degree

Node update: sparse matrix multiplication to aggregate messages without additional scaling mechanisms.



Gine Layer

Message passing: the GINELayer combines source node features and transformed edge features in message computation.

Node update: Uses a learnable trainable epsilon to scale the contribution of node features during updates.

```
src, dst = edge_index

# updating edge features with mlp, bringing them to the same dimension as node features
edge_attr = self.edge_mlp(edge_attr.float())
message = x[src] + edge_attr

# aggregating messages
aggr_out = scatter_mean(message, dst, dim=0, dim_size=x.size(0))

# node update
out = self.node_mlp((1 + self.eps) * x + aggr_out)
```

2-Step- Gine

Uses an additional pre-trained sit-stand model to compute extra input features for the GINE model.

```
sit_stand_model = GINE(
    num_features=3,
    num_classes=3,
    hidden_channels=256,
    num_layers=8,
    use_batch_norm=True
).to(device)
```

```
model = GINE(
    num_features=config["input_dim"],
    num_classes=output_dim,
    hidden_channels=128,
    num_layers=config["num_layers"],
    use_batch_norm=True
).to(device)
```

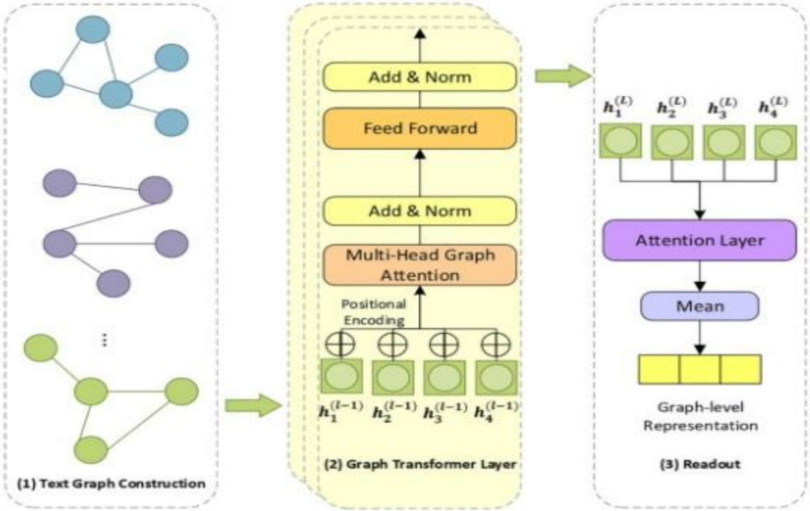
The pre-trained model predicts the sit-stand state, which is added as an additional feature column to the input node features

```
sit_stand_pred = torch.argmax(sit_stand_out, dim=1)
sit_stand_pred = torch.repeat_interleave(sit_stand_pred, 145, dim=0).unsqueeze(0).T
batch.x = torch.cat((batch.x, sit_stand_pred), 1)
```

Graph Transformer

Message passing: Each `TransformerConv` layer performs message passing based on self-attention

Self-attention: dot-product attention mechanism but applied locally in the graph neighborhood

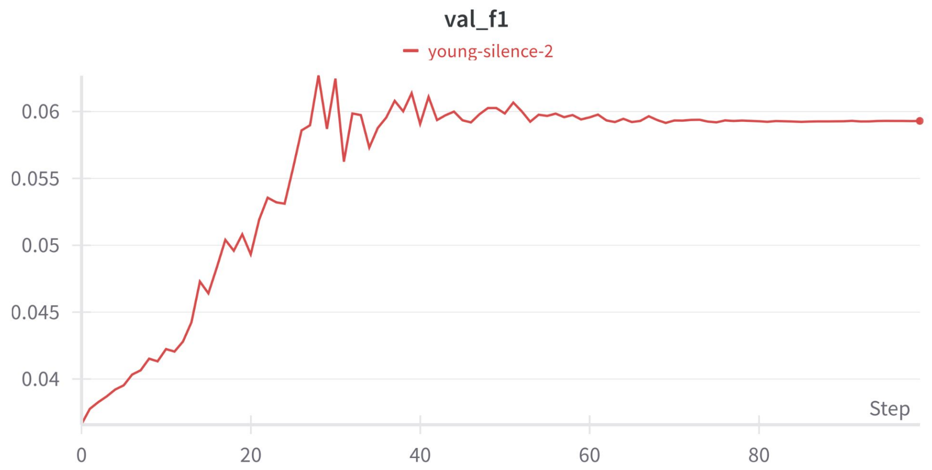


Experimental results – GCN Baseline

256 hidden size, 2 layers.

Test Scores:

- F1**: 0.0513**
- Multiclass accuracy: **0.6542**



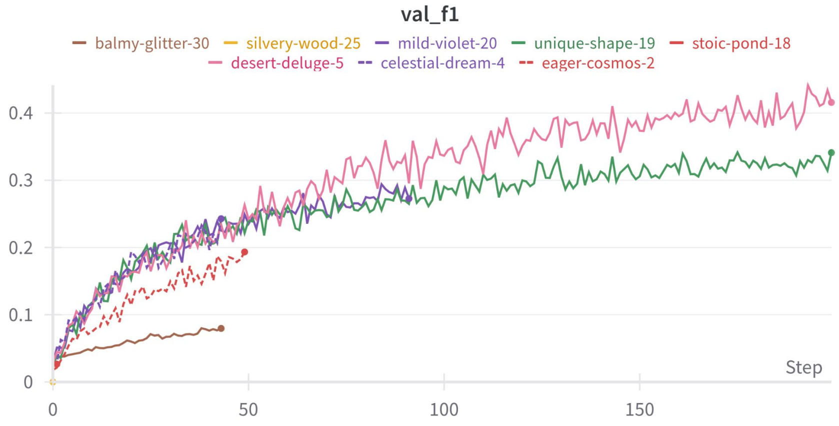
Experimental Results - Just a Big Transformer

200 epochs, 256 hidden size, 16 heads.

Tweaks: batch normalization across features

Test F1 Scores:

- Without learning rate scheduler: **0.3775**.
- With cosine LR scheduler: **0.3737**.
- Adding feature normalization across batch: (Result pending completion).

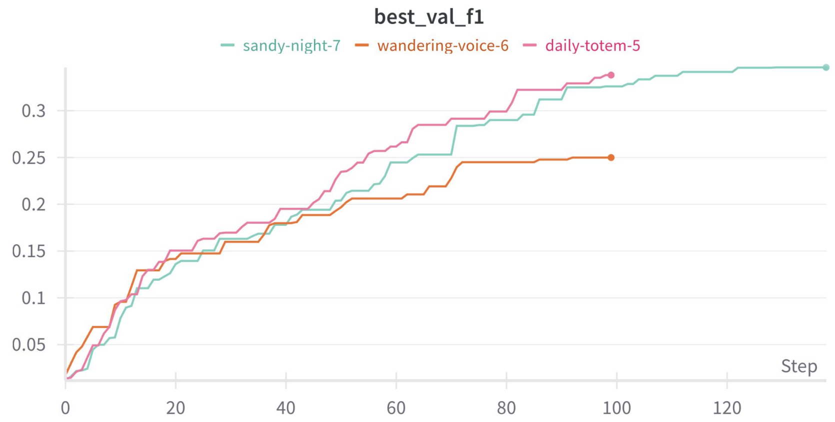


Experimental Results - GINE

Graph Isomorphism Network with Edge Features (GINE)

Tweaks: +plateu scheduler, + edge features, + undersampling

- **Configuration 1:**
  - 64 hidden size, 4 layers
  - **Test F1 Score: 0.2983**.
- **Configuration 2:**
  - 512 hidden size, 12 layers, same epochs and batch size.
  - Similar performance to Configuration 1.



Experimental Results - 2-Step GINE

Two-Step Classification

First, we trained a small GINE to classify sitting, standing, and none of that (trained on 20% of the data):

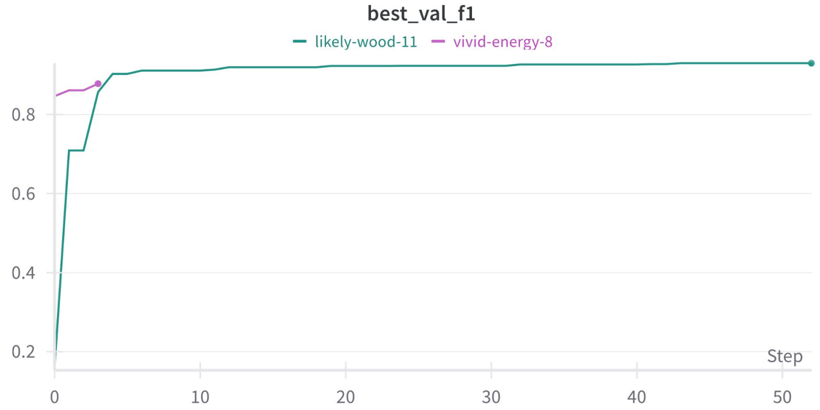
- Additional feature dimension for `sit_stand` label.
- Configuration: 256 hidden size, 4 layers, batch size, early stopping.
- **Test Scores:**
  - F1 Score: **0.9302**

Then, we trained a larger model for which the features were augmented by the prediction of the sit/stand model.

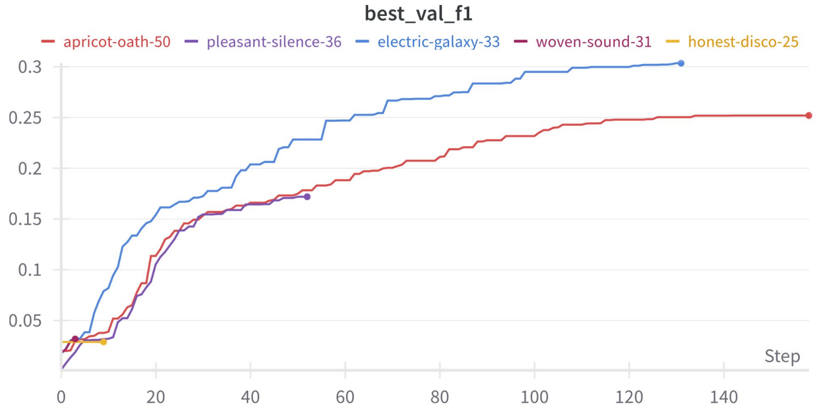
- Additional feature dimension for `sit_stand` label. (Did it in 2 ways)
- Configuration: 512 hidden size, 8 layers, 1028 batch size, early stopping.
- **Test Scores:**
  - F1 Score: **0.2489**.
  - Multiclass Accuracy: **0.6012**.

Notes: GINE is small, quick to train

Sit/stand:



Main Model:



Experimental Results - Back to Transformer

Lastly, we trained a smaller transformer model, but added the following tweaks:

Tweaks: + class weights to the loss function

First, we trained a small GINE to classify sitting, standing, and none of that:

- Additional feature dimension for `sit_stand` label.
- Configuration: 256 hidden size, 2 layers, 4 attention heads.
- **Test Scores:**
    - F1 Score: **0.0934**
    - Multiclass Accuracy: 0**.8032**

Discussion

What we could have done also:

Experiment more with spatio-temporal features

Positional encodings (RW in space, RW in time)

Add weights to the metrics