Generative AI Project Overview

General

1. Text Encoder →
    a. pretrained models to extract sentence embeddings (BERT or CLIP –maybe both)
    b. Optional: extract object-relationship triples from text using NLP parser (spaCy,OpenIE)

2. Scene Graph Construction →
    a. Nodes: objects (“lamp”,”bed” etc.)
    b. Edges: spatial or semantic relations (“left”,”on”,”near”)
3. Graph Transformer →
    a. Transformer over the scene graph (GAT and Graphomer/SceneGraphFormer)
    b. Predict layout parameters for each node (x,y,w,h)
4. Layout Rendering →
    a. Visualize layout predictions (matplotlib)
5. Evaluation →
    a. IoU: Overlap between predicted and GT bounding boxes
    b. Scene Graph Accuracy: Consistency of relative positions
    c. CLIP Similarity: CLIP score between text & rendered layout
    d. Diversity: Are different layouts generated for similar prompts
    e. Graph Edit Distance(GED) between predicted and ground truth graphs

f. Node/Edge Classification Accuracy (only for supervised graph construction)

6. Datasets →
   a. Visual Genome (native scene graphs)
   b. GQA (Graph-based questions and annotations)
   c. CLEVR (synthetic graph -layout supervision)
   d. SUN RGB-D

Person A: GAT Model

- **Dataset Preprocessing:** Prepare Visual Genome : extract scene graphs and layout annotations
- **Text Encoder:** Integrate pretrained encoder BERT for textual feature extraction
- **Text → Scene Graph:** Implement NLP parser (spaCy + OpenIE)
- **Model Implementation:** Implement GAT model to predict layout from scene graph
- **Training:** Build training and validation loop for GAT
- **Loss Function:** L1/L2
- **Scene Graph Evaluation:** Measure graph consistency: node/edge accuracy, edit distance (for GAT pipeline)
- **Evaluation metrics:** as stated above (here mode 1 vs model 2)
- **Visualization:** output visualization
- **Study:** Try different neighborhood radii or attention heads in GAT

Person B: Graphormer Model

- **Dataset Preprocessing:** Prepare SUN RGB-D: convert layouts to scene graphs and normalize object labels

- **Text Encoder:** Integrate pretrained encoder CLIP for textual feature extraction
- **Text → Scene Graph:** Implement SUN RGB-D parser (spaCy + OpenIE)
- **Model Implementation:** Implement Graphormer model to predict layout from scene graph
- **Training:** Build training and validation loop for Graphormer
- **Loss Function:** L1/L2 (same as other model)—maybe relational loss
- **Scene Graph Evaluation:** Measure graph consistency: node/edge accuracy, edit distance (for Graphormer pipeline)
- **Evaluation metrics:** as stated above
- **Visualization:** output visualization
- **Study:** Try different positional encodings or number of layers in Graphormer