Proposal for Advancing Geometric Set Analysis via Point-wise Inference

Prepared by:

Ahmed Mohammed EL Sherbeny - ID: 120220271

Hatem Feckry Sultan - ID: 120220264

Seif eldin Farghaly - ID: 120220252

Abdulrahman Ashraf - ID: 120220292

Proposal for Advancing Geometric Set Analysis via Point-wise Inference

1. Building on Week 13's Foundation

Week 13's work provided a crucial transformation of the Happy Ending problem into a classification task. Through the use of YOLO on rasterized point sets, you demonstrated impressive accuracy in determining whether a point set contained a convex polygon. However, this global binary decision can be expanded into richer inference. Instead of just detecting the existence of convexity, our goal now is to identify and *understand* the geometric factors contributing to convex or empty configurations. This requires transitioning from classification to segmentation.

2. Theory and Justification for Next Steps

From a theoretical standpoint, segmentation-where each point is labeled as critical or not-requires the model to learn not just presence, but structure. This aligns well with the geometric essence of the Erdos-Szekeres conjecture. We propose applying *Graph Neural Networks (GNNs)* and *contrastive learning* in addition to Point Transformer V3. Graph models like DGCNN and contrastive frameworks can explicitly model relationships between points, emphasizing the latent geometric features that define emptiness in a set.
This shift from classification to relation-based learning transforms the model's understanding into something explainable and potentially generalizable across polygon sizes. Using spectral graph theory, for example, eigenvalue distributions of the adjacency matrix could indicate convexity structures. Embedding this insight into training loss functions or graph-based attention modules can yield interpretable, mathematical representations.

3. Integrated Plan Based on Week 13 Findings

Step 1: Use the previously labeled dataset (Empty vs Non-Empty) to initialize feature extraction layers with pretrained CNN weights.

Step 2: Move to raw point inputs and define graphs (kNN-based) for DGCNN to use as input. Alternatively, use Point Transformer V3 if dependencies can be resolved via Docker/AWS as previously planned.

Step 3: Label the segmentation masks using heuristics from convex hull algorithms-e.g., Graham scan-to define which points form convex polygons.

Step 4: Introduce contrastive loss that separates convex-forming vs non-convex-forming point embeddings in feature space.

Step 5: Visualize attention or edge weights to gain insight into which point connections matter for classification.

4. Deployment Strategy

To overcome technical bottlenecks, a Docker-based container with CUDA 11.7 and PyTorch 2.0 will be configured. This will be hosted on AWS g4dn.xlarge or equivalent GPU instances. An alternative local solution includes WSL2 with NVIDIA GPU passthrough if AWS is inaccessible.

A gradual curriculum-learning approach can help stabilize training:

- Start with synthetic sets of 8 points
- Gradually increase to 16, 24, 32 points
- For each, assess both classification accuracy and segmentation precision

5. Expected Impact and Scientific Contribution

This refined approach bridges classic combinatorics and modern AI. By labeling not just sets but points, we can construct a benchmark dataset of minimal empty sets for various polygon sizes, aiding further research on the Erdos-Szekeres problem.

We also expect to uncover geometric 'motifs'-frequently occurring local patterns that inhibit convexity. These patterns, if captured through attention heads or graph kernels, can be cataloged and linked to theoretical bounds.

In summary, this phase transforms the model from a detector into a geometric analyst-one capable of explaining *why* a set is empty.