Proposal for Advancing Geometric Set Analysis via Point-wise Inference

Ahmed Mohammed EL Sherbeny - ID: 120220271
Hatem Feckry Sultan - ID: 120220264
Seif Eldin Farghaly - ID: 120220252
Abdulrahman Ashraf - ID: 120220292

May 16, 2025

1. Building on Week 13's Foundation

Week 13's work marked a pivotal step in reframing the Happy Ending problem from a purely theoretical question to a computational classification task. Through the conversion of point sets into rasterized images and the training of a YOLO-based CNN, we demonstrated high accuracy in distinguishing between empty and non-empty sets—those which do and do not contain convex polygons, respectively. This approach validated that convolutional architectures, when trained with sufficient data, can learn to recognize even subtle geometric cues.

However, this binary classification merely scratches the surface of what can be learned. Rather than stopping at "yes" or "no" decisions, the next stage aims to delve deeper into *why* a set exhibits or lacks convexity. This motivates the transition from image-level classification to point-wise segmentation, where the model predicts, for each point, its role in the emptiness or convexity of the set.

2. Theoretical Basis and Justification

From a mathematical perspective, the Happy Ending problem concerns structural properties within combinatorial geometry, particularly relating to the Erdős–Szekeres conjecture. Our objective shifts from classifying entire configurations to *inferring internal geometric dependencies*. To do so, we propose incorporating advanced geometric learning techniques:

2.1 Graph Neural Networks (GNNs)

GNNs, especially DGCNNs, are suitable for modeling the relationships between points in a set. Each point is considered a node, and the edges are determined using spatial proximity (e.g., k-NN graphs). Unlike CNNs, GNNs inherently encode relational structure, enabling them to identify subtle motifs such as nested quadrilaterals or spirals that block convexity.

2.2 Spectral Graph Theory

The use of eigenvalues from graph Laplacians can reflect the structural complexity of the point cloud. For instance, low spectral gaps often correspond to symmetric, possibly convex configurations, while higher gaps indicate disorder. Embedding spectral characteristics in the loss function may improve model interpretability.

2.3 Contrastive Learning

To enforce separation in learned feature space between critical and non-critical points, contrastive losses can be applied. These losses push embeddings of points that contribute to convexity or emptiness closer to their respective prototypes, enhancing feature discrimination and robustness.

3. Detailed Integration Plan

Building on the classification foundation, the next phase will be executed as follows:

Step 1: **Pretraining:** Utilize the labeled dataset from Week 13 to initialize CNN backbones, retaining features that generalize well to geometric distinctions.

Step 2: **Graph Construction:** Transition to raw (x, y) coordinates. Use k-NN to build input graphs for DGCNN or serialize point clouds for Point Transformer V3 input.

Step 3: **Ground Truth Generation:** For each point set, use convex hull algorithms (e.g., Graham scan) to identify which points form the convex structure and label them accordingly.

Step 4: **Contrastive Embedding:** Augment loss functions with contrastive terms to enforce separation between convex-contributing and passive points in embedding space.

Step 5: **Attention Visualization:** Visualize learned attention weights or edge importance from GNNs or transformers to interpret which parts of the structure are influential in prediction.

4. Deployment and Technical Roadmap

4.1 Dockerization and Remote Training

The previously noted installation issues (CUDA compatibility, dependency mismatches) will be resolved using Docker containers with pre-built images for PyTorch ≥ 2.0 and CUDA 11.7+. AWS EC2 (g4dn.xlarge) will be used to run these containers with GPU acceleration.

4.2 Curriculum Learning Strategy

Training will follow a staged complexity model:

- Begin with synthetic 8-point sets.
- Progress to 16, 24, and 32-point sets, adjusting model capacity and attention depth.
- Evaluate segmentation precision and visual interpretability at each stage.

4.3 Alternative Local Solution

For offline work, WSL2 with NVIDIA passthrough is being configured to allow native Linux execution with GPU access for local experiments.

5. Expected Contributions and Research Value

By moving beyond binary classification, this phase aims to build a *semantic geometry engine*—capable of explaining structural voids in point sets. Contributions include:

- A segmented dataset with per-point annotations for geometric roles.
- A model that can identify, explain, and predict critical point structures in geometric sets.
- Insights into latent geometric motifs through learned attention or spectral embeddings.

This work contributes to both machine learning and discrete geometry communities by proposing a computational method to approach long-standing conjectures with scalable tools. It turns deep learning models into interpretable assistants for mathematical discovery.

6. Hypothetical Results and Contingency Insights

While our planned strategy is theoretically sound, it is crucial to consider the possibility of imperfect outcomes. If the model underperforms—but not catastrophically—it may still yield insights of significant value:

6.1 Partial Success in Segmentation

Even if segmentation accuracy is modest (e.g., 70–80% IoU on critical point identification), the model can still act as a filter or preprocessor. Human inspection of model outputs may reveal patterns the model frequently captures (e.g., radial symmetries, clustered gaps) and where it fails (e.g., high-density point overlaps).

6.2 Insights Through Failure Modes

Failure cases can guide the development of new geometric hypotheses. For instance:

- If the model systematically mislabels certain motifs, this could hint at undiscovered geometric invariants or exceptions to existing heuristics.
- If attention weights concentrate on misleading points, we may refine the ground truth definition (e.g., ambiguity in edge cases of convexity).

6.3 Useful Artifacts Despite Lower Accuracy

Artifacts from early-stage models—such as attention maps, learned embeddings, or proximity graphs—can still be used to:

1. Initialize more interpretable models (e.g., rule-based or hybrid models).
2. Train smaller classifiers on specific geometric motifs (e.g., triangles nested in quadrilaterals).
3. Quantify uncertainty in geometric configurations, forming probabilistic descriptors of emptiness.

6.4 Adaptive Model Tuning

Sub-optimal outcomes would direct us to:

- Adjust the granularity of segmentation labels (e.g., ranking point importance rather than binary labels).
- Explore hybrid CNN-GNN architectures to combine spatial and topological learning.
- Conduct model distillation from transformer outputs to lighter and more tunable architectures.

In essence, even in non-ideal cases, the project will produce:

- A rare annotated dataset for geometric segmentation.
- An understanding of what geometric features neural models struggle to learn.
- A foundation for iterative refinement based on error analysis.