

3. Geometry-Aware Operator Transformer (GAOT) (50 points + 20 points)

In this project, you will work with the state-of-the-art Geometry-Aware Operator Transformer (GAOT). The standard implementation primarily utilizes **Strategy I (Structured Stencil Grid)** for tokenization (see **S.M. B.1**), effectively treating the latent physics space as an image processed by a Vision Transformer (ViT).

However, real-world engineering problems often involve highly irregular geometries where structured grids are inefficient. Your goal is to extend GAOT to support **Strategy II: Random Sampling Tokenization**. This involves designing a dynamic radius for information aggregation and rethinking positional encodings (PE) for continuous coordinates.

Tasks

Task 1: Establishing a Baseline (10 points)

Before modifying the architecture, you must establish a performance baseline using the official GAOT implementation.

1. Download the **Elasticity** dataset ([Link](#)) as described in the GAOT paper.
2. Train the default GAOT model (using Strategy I: Stencil Grid) on this dataset.

Record the Test Relative L^1 Error, the number of tokens used, and the total training time. This will serve as your baseline for subsequent comparisons.

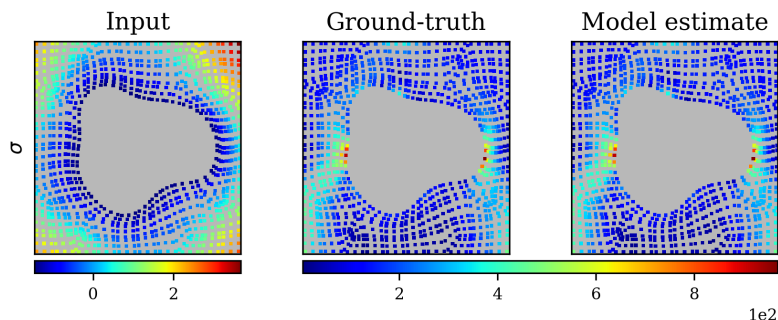


Figure 3: Model input, ground-truth solution, and model estimate of a test sample of the Elasticity dataset.

Task 2: Random Sampling & Dynamic Radius Strategy (40 points)

- **(20 points)** Modify the tokenization logic. Instead of a fixed meshgrid, you will randomly sample N points (e.g., $N = 128$ or 256) from the domain \mathcal{D} to serve as latent tokens $\{y_i\}_{i=1}^N$. You may choose to use a *fixed set* of sampled tokens (support precompute graph structures) or *resample* every epoch (improves resolution invariance but requires rebuilding graphs).

- **(20 points)** With random points, a fixed radius r may give rise to "holes" (too small) or "large overlap" (too large) in the domain. You may implement a **Dynamic Radius** strategy inspired by *RIGNO*. Compute a local radius r_l for each token y_l :

$$r_l = \alpha \cdot d_k(y_l) \quad (2)$$

where $d_k(y_l)$ is the distance determined by KNN or Delaunay triangulation, and α is a scaling factor. Your Goal is to ensure the graph connectivity covers the entire physical domain despite the randomness of tokens.

Task 3: Re-thinking Positional Encoding (BONUS - 20 points)

Transformers are permutation invariant and require explicit position information. In Strategy II, grid indices no longer exist, so you must implement:

- **Absolute PE:** Create an MLP that maps coordinate vectors $\mathbf{x} \in \mathbb{R}^2$ to the hidden dimension D .
- **Continuous Relative Bias:** Standard RoPE relies on integer gaps. Investigate applying RoPE using continuous coordinates or replacing it with a relative bias term based on Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ in the Attention matrix.

Since we are not using patching, the sequence length N can be large. Implement a **Cross-Attention** layer (similar to PerceiverIO) to project N random tokens into a smaller set of "seed" tokens, or propose a method to patch and merge irregular tokens (see *SpiderSolver*).

References

- Codebase: <https://github.com/camlab-ethz/GAOT>
- **GAOT:** Wen et al. (2025). *Geometry aware operator transformer as an efficient and accurate neural surrogate for PDEs on arbitrary domains*.
- **RIGNO:** Mousavi et al. (2025). *A Graph-based framework for robust and accurate operator learning for PDEs on arbitrary domains*.
- **SpiderSolver:** Qi et al. *A Geometry-Aware Transformer for Solving PDEs on Complex Geometries*. (NeurIPS).