Question 1: Determining What Data an RTSTRUCT Stores (Boundary Points vs. Interior Points)

1. Objective

Before further processing anatomical contours in medical images (e.g., contour simplification, similarity evaluation, point density analysis), we must first clarify:

> **Does an RTSTRUCT store all points inside a region, or only the boundary (contour) points?**

This decision directly affects whether we need preprocessing steps such as explicit "boundary extraction".

2. Background

The study object is a DICOM RTSTRUCT file. RTSTRUCT is the standard format used in radiotherapy planning to store organ contours delineated manually or semi-automatically.

3. Method

We read the RTSTRUCT using Python with `pydicom` and inspect key fields that define contours, including:

- `ContourGeometricType`
- `NumberOfContourPoints`
- the structure and length of `ContourData`

4. Implementation Code

```python
from pathlib import Path
import pydicom

# Find an RTSTRUCT file
rtstruct_path = next(Path(r'd:\point_plotting_reserch').rglob('*.RTSTRUCT*.dcm'))

# Read DICOM
ds = pydicom.dcmread(str(rtstruct_path))

# Print ROI list
print("ROIs in this RTSTRUCT:")
for r in ds.StructureSetROISequence:
    print(f"  ROINumber={r.ROINumber}, ROIName={r.ROIName}")

# Read the first contour of the first ROI
contour = ds.ROIContourSequence[0].ContourSequence[0]

# Print key fields
print("\nContour information:")
```

```
print("ContourGeometricType:", contour.ContourGeometricType)
print("NumberOfContourPoints:", contour.NumberOfContourPoints)
print("len(ContourData):", len(contour.ContourData))
```

5. Output

```
(.venv) PS D:\point_plotting_reserch\src> python check_edge_points.py
ROIs in this RTSTRUCT:
  ROINumber=1, ROIName=Prostate
  ROINumber=2, ROIName=Lesion 1

Contour information:
  ContourGeometricType: CLOSED_PLANAR
  NumberOfContourPoints: 308
  len(ContourData): 924
```

Because `len(ContourData)` is exactly three times `NumberOfContourPoints` (each point has x, y, z), this confirms the file stores only contour (boundary) points.

Question 2: Determining the Minimal Number of Boundary Points

---

(Contour Simplification and Similarity Evaluation)

---

1. Objective

After confirming that the RTSTRUCT stores organ **boundary points**, the next question is:

> **How many boundary points are minimally required to keep the simplified contour highly consistent with the original contour geometry?**

Identifying the minimal effective point count helps:

- reduce contour data size
- improve efficiency of geometric computations and downstream analysis
- provide a standardized contour representation for multi-case statistics

---

2. Method Overview

2.1 Original Contour

- **Source**: DICOM RTSTRUCT
- **ROI**: Prostate
- **Original number of points**: 308

- **Contour type**: `CLOSED_PLANAR` (closed planar contour)

2.2 Simplification Strategy

We treat the original contour as a **closed polyline** and perform **uniform resampling along arc length** to generate simplified contours with different target point counts.

The tested target point counts are:

```
16
32
48
64
128
256
```

2.3 Similarity Metrics

To quantitatively compare simplified contours with the original, we use two complementary metrics:

(1) Area Relative Error

$\text{Area Error} = \frac{|\text{Area}_{simplified} - \text{Area}_{original}|}{\text{Area}_{original}}$

This measures how well the simplified contour preserves the **global shape and scale**.

---

(2) Hausdorff Distance

Hausdorff distance measures the **maximum local deviation** between two curves:

$d_H(A, B) = \max \left(\sup_{a \in A} \inf_{b \in B} |a - b|, \sup_{b \in B} \inf_{a \in A} |b - a| \right)$

This metric is highly sensitive to local geometric error, with units of **millimeters (mm)**.

2.4 Acceptable Error Thresholds

Based on practical geometric accuracy needs in medical imaging, we use the following thresholds:

Metric	Threshold
Area relative error	≤ 1.0 %
Hausdorff distance	≤ 0.5 mm

If a simplified contour satisfies both thresholds, we consider it geometrically equivalent to the original.

---

3. Experimental Results

Program output:

```
pts | area_err(%) | hausdorff(mm)
----------------------------------
 16 |    2.417 |        1.199
 32 |    1.314 |        0.991
 48 |    0.767 |        0.464
 64 |    0.214 |        0.367
128 |    0.234 |        0.256
256 |    0.022 |        0.129
```

Automatic decision:

```
Minimal acceptable points = 48
(area_err ≤ 1.0%, Hausdorff ≤ 0.5 mm)
```

4. Analysis

When the number of points is small (16, 32)

- Both the area error and the Hausdorff distance exceed the thresholds
- Significant distortion appears globally and locally

When the number of points reaches 48

- Area relative error first drops below **1%**
- Hausdorff distance first drops below **0.5 mm**

This indicates strong agreement with the original contour in both global and local geometry.
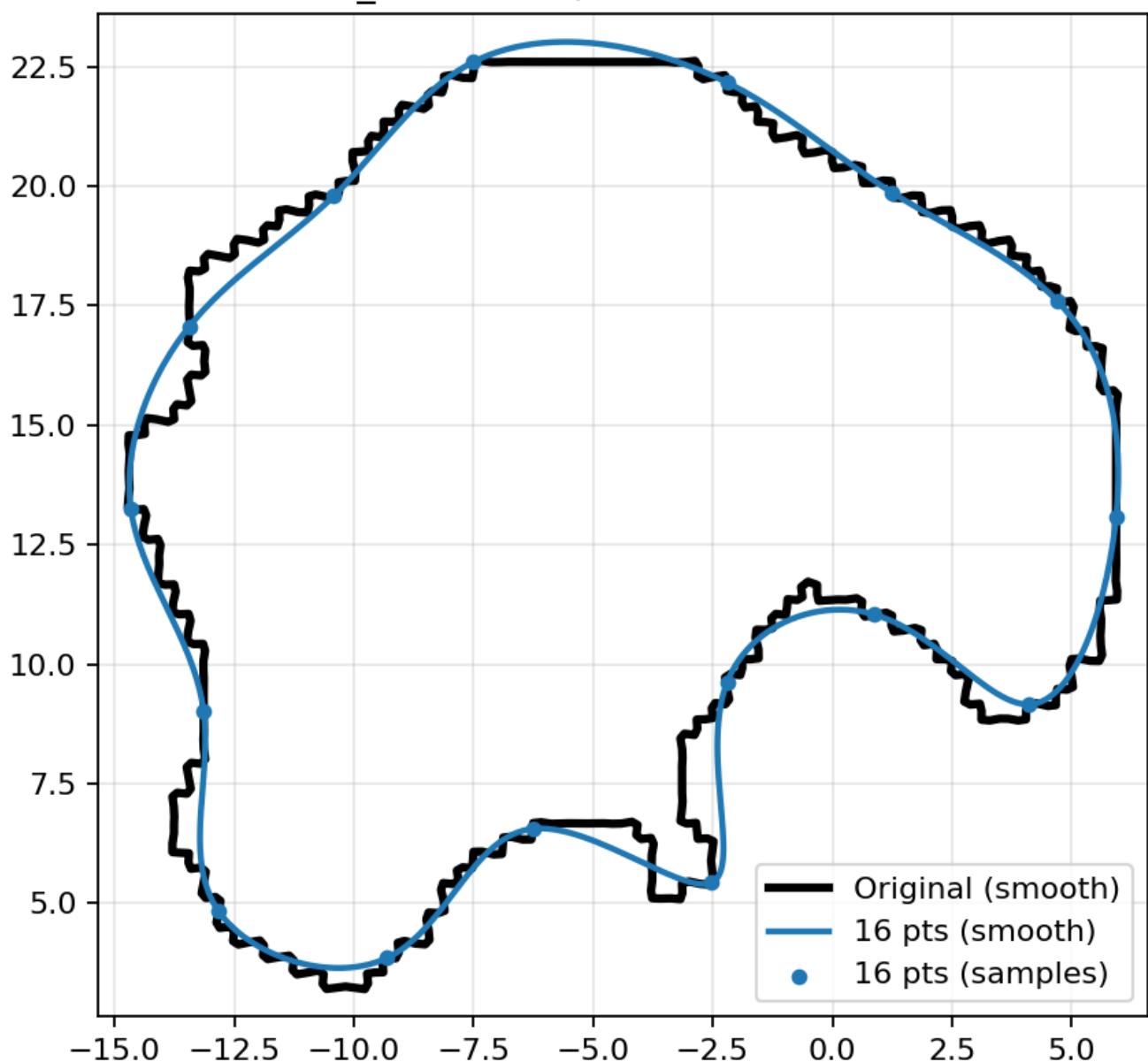
When increasing points further (64, 128, 256)

- Geometric error continues to decrease
- The reduction rate becomes smaller
- Clear diminishing returns appear
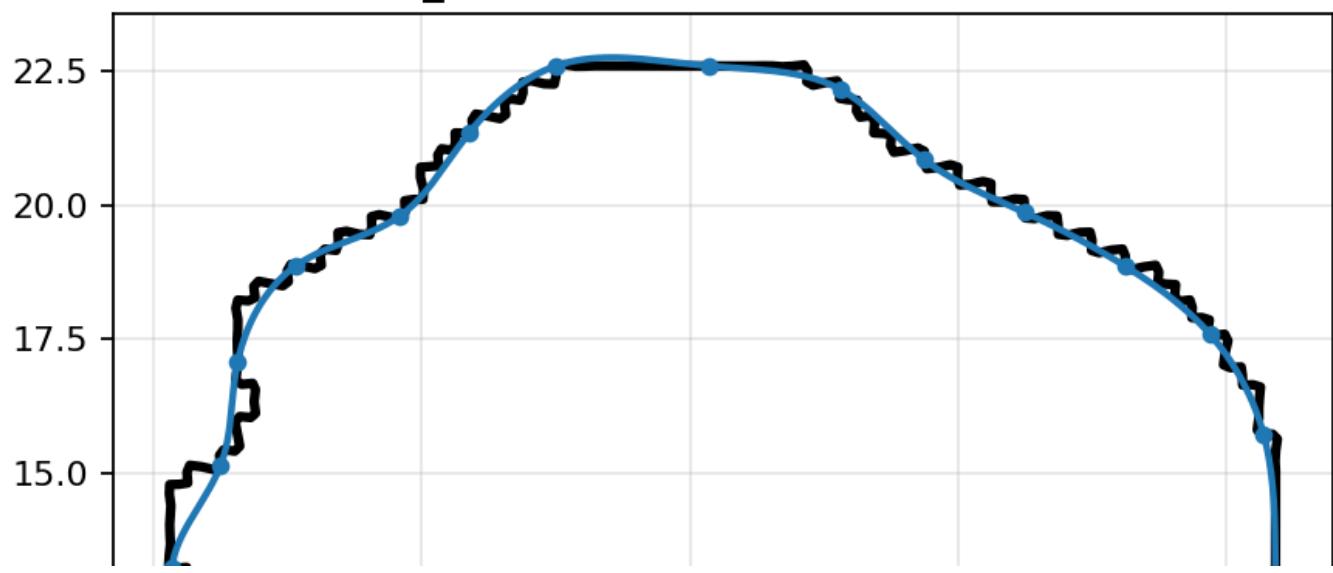
5. Visualization Results
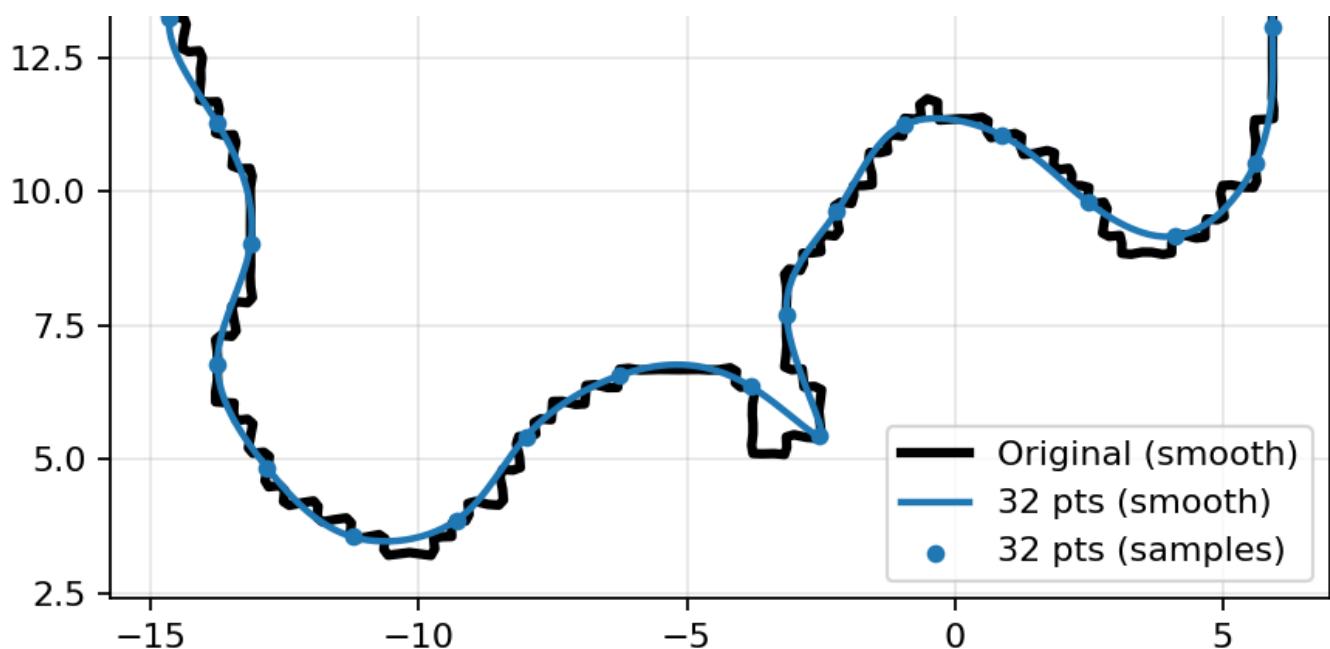
The following figures compare the original contour to simplified contours with different point counts:

Prostate – 16 pts
## area_err=2.42%, Hausdorff=1.20 mm

Legend:
- Original (smooth)
- 16 pts (smooth)
- 16 pts (samples)
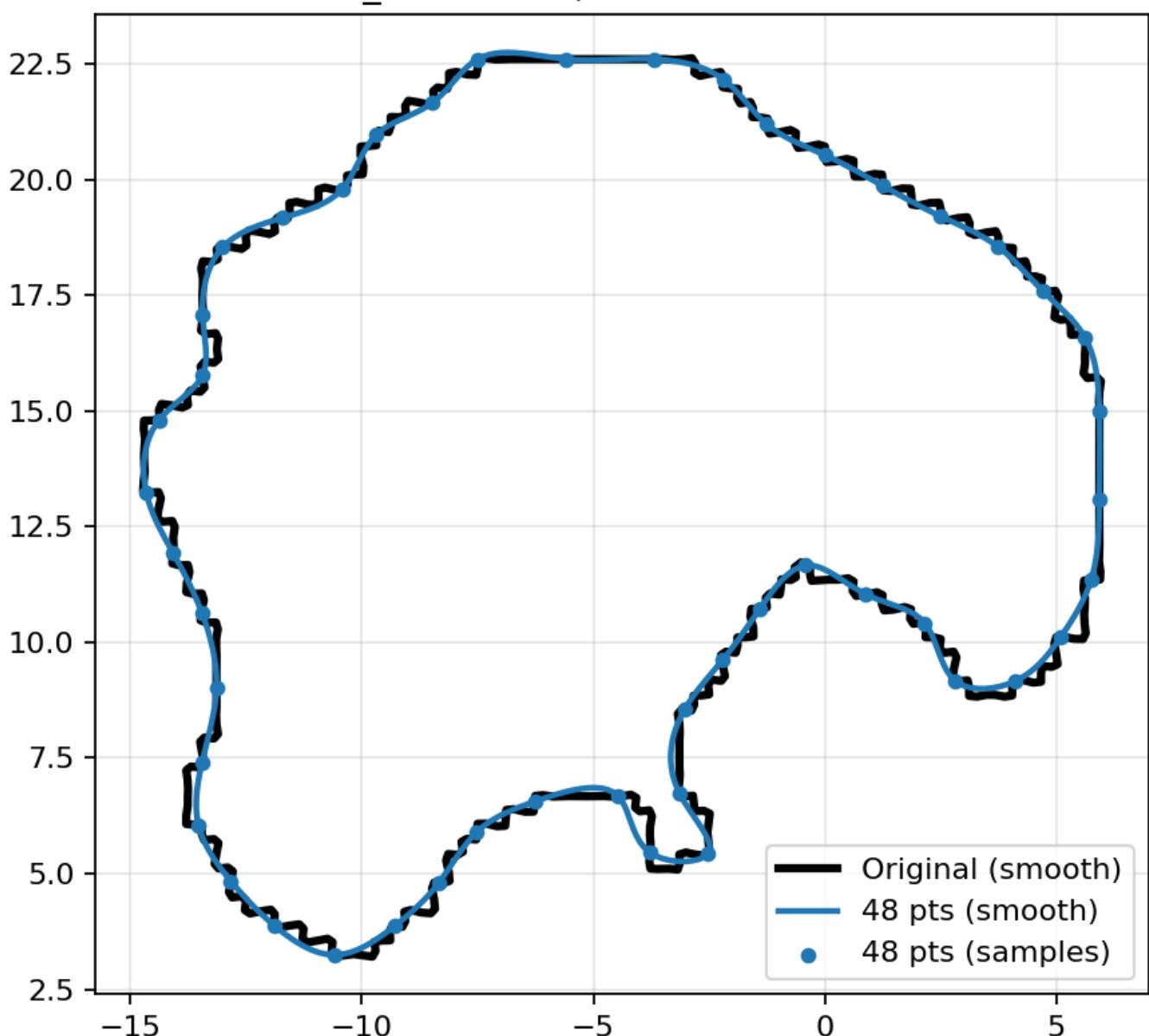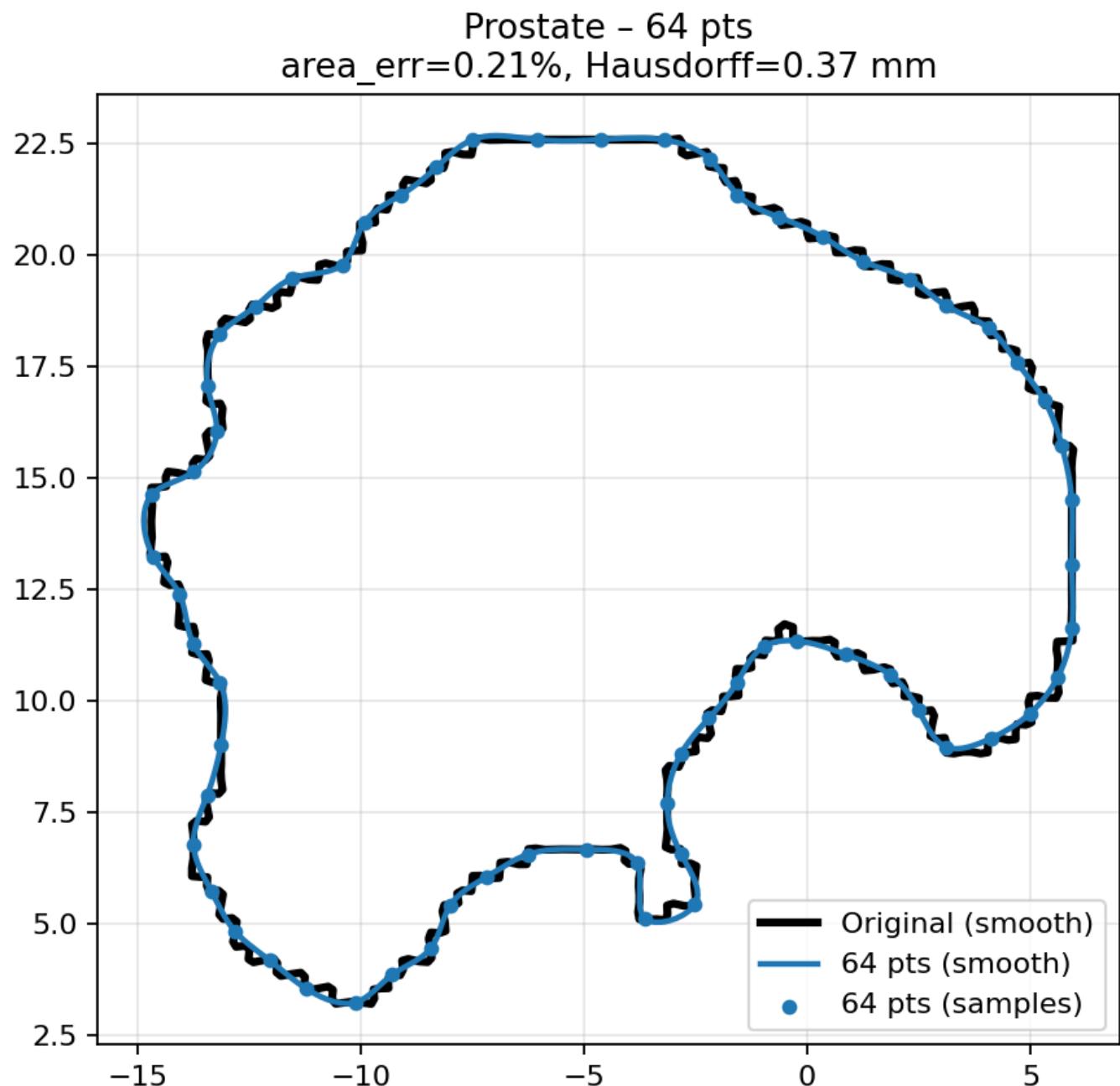
Prostate – 32 pts
area_err=1.31%, Hausdorff=0.99 mm
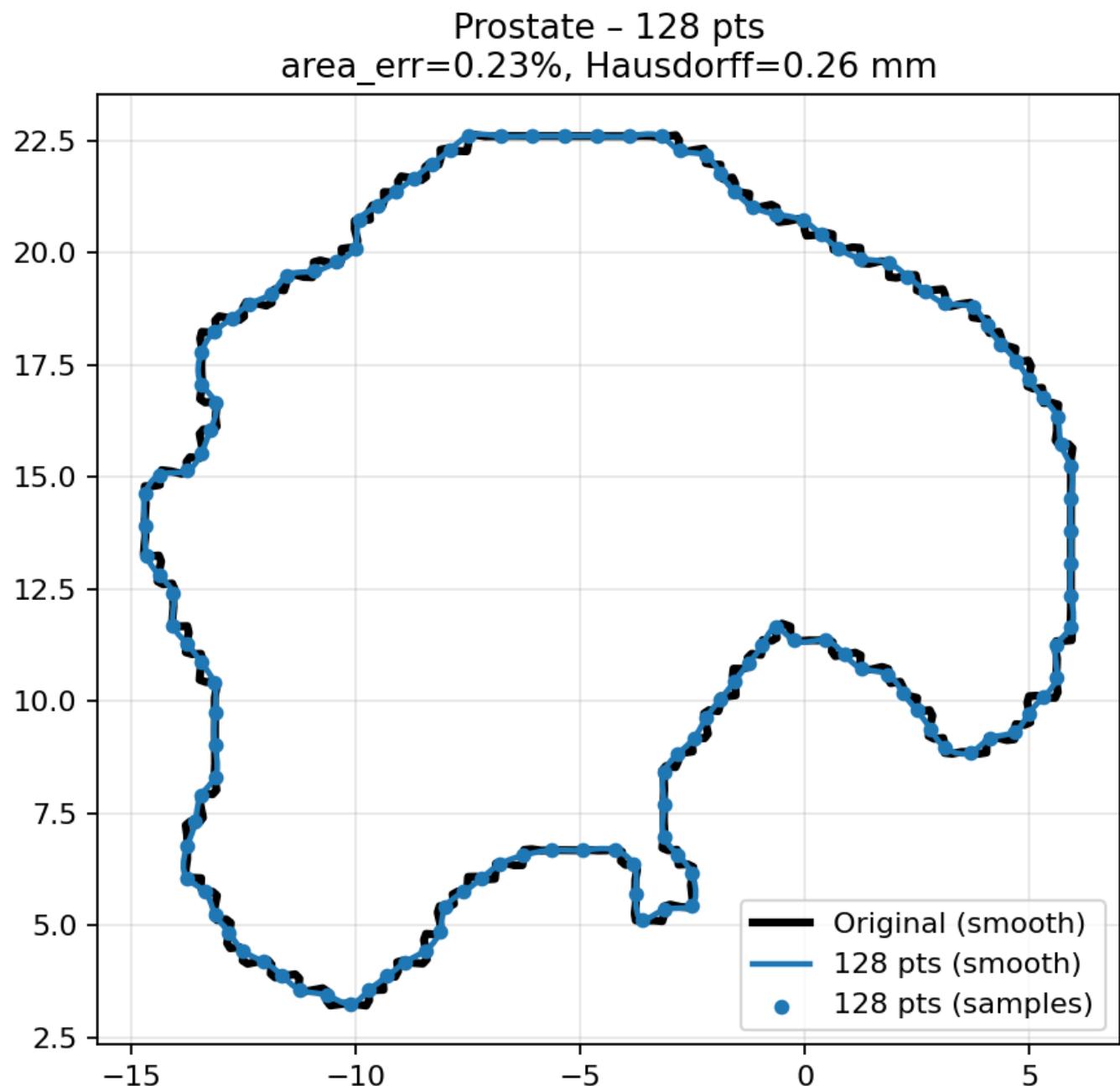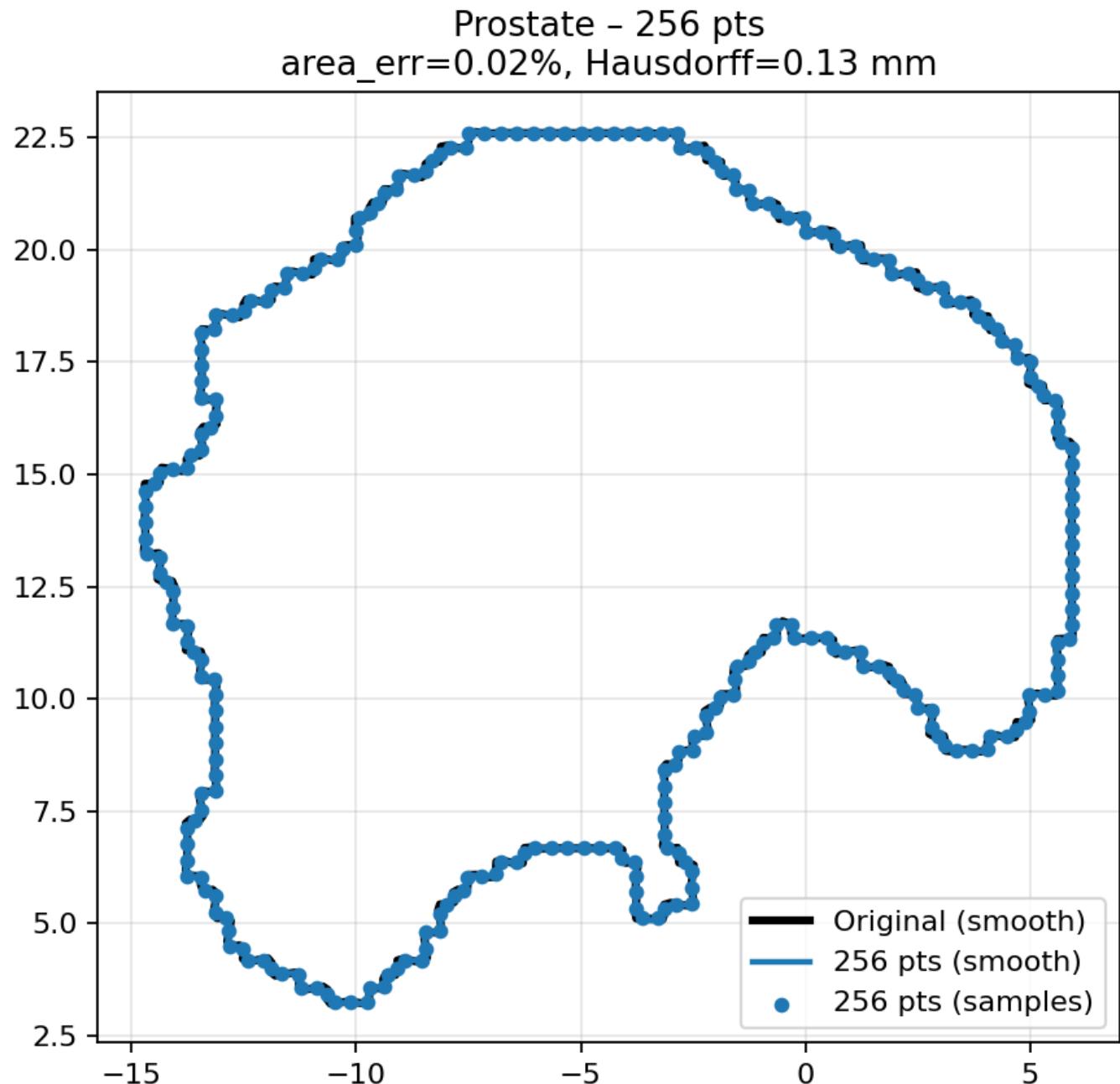
Prostate – 48 pts
area_err=0.77%, Hausdorff=0.46 mm

Prostate – 64 pts
area_err=0.21%, Hausdorff=0.37 mm

Legend:
- Original (smooth)
- 64 pts (smooth)
- 64 pts (samples)

Prostate – 128 pts
area_err=0.23%, Hausdorff=0.26 mm

Prostate – 256 pts
area_err=0.02%, Hausdorff=0.13 mm

Notes:

- All geometric errors are computed on the true polyline points
- The smoothed curve is only for visualization and is not used in error computation

---

6. Conclusion

- Geometric error decreases monotonically as point count increases
- Under the thresholds defined above:

> **48 boundary points are the minimal acceptable number to preserve geometric similarity**

Question 3: Adaptive Sampling Based on Boundary Geometric Complexity

(Analysis of Boundary Point Density and Contour Reconstruction Quality)

---

1. Motivation

The previous experiments confirmed:

- The RTSTRUCT stores only organ **boundary points**
- For the prostate contour, **48 uniformly resampled boundary points** can be geometrically equivalent to the original contour in terms of **area error** and **Hausdorff distance**

However, **uniform arc-length resampling** makes a key assumption:

> **Geometric complexity is uniform everywhere along the contour**

This assumption is not true for real anatomical shapes. Organ contours typically contain both:

- **high-curvature regions**: sharp turns
- **low-curvature regions**: nearly straight or slowly varying segments

Uniformly distributing points can cause:

- wasted points in smooth regions
- insufficient points in complex regions, causing local shape distortion

Therefore, the core question of this section is:

> **Can a non-uniform boundary point distribution reconstruct the contour better under the same total point budget?**

---

2. Method Overview

2.1 Key Idea

We treat the original contour as a **closed boundary curve** and allocate sampling points adaptively according to **local geometric complexity** (e.g., turning strength / curvature):

- higher complexity → **denser points**
- lower complexity → **sparser points**

With a fixed total number of points, the goals are:

- preserve local geometric features as much as possible
- reduce the maximum boundary deviation (Hausdorff distance)

2.2 Adaptive Sampling Strategy

Implementation steps:

1. represent the original boundary with a high-resolution parameterization
2. estimate local geometric variation (e.g., turning strength)
3. build a non-uniform cumulative distribution function along the curve
4. sample uniformly in that distribution to obtain the target number of points

In essence:

$\text{point density} \propto \text{local geometric complexity}$

---

3. Notes on Smoothing for Visualization (Technical Details)

To avoid jagged edges or unnatural straight-line connections after simplification, we apply smoothing on the adaptively sampled points.

Two smoothing approaches were tried:

- global spline smoothing
- local Chaikin subdivision smoothing

Important note:

> **Smoothing is not the core focus here; it is only used to visualize the effect of point distribution on reconstruction quality.**

---

4. Experimental Results

With the same adaptive sampling point counts, we compute:

- area relative error
- Hausdorff distance

4.1 Results with Global Spline Smoothing

pts	area_err (%)	Hausdorff (mm)
16	4.020	4.063
32	0.912	3.112
48	0.501	2.534
64	0.372	2.500
128	0.026	2.202
256	0.027	2.154

4.2 Results with Local Chaikin Smoothing

pts	area_err (%)	Hausdorff (mm)
16	3.546	3.797
32	1.163	2.825
48	0.268	2.401
64	0.531	2.193
128	0.141	1.508
256	0.043	0.987

5. Analysis: Point Distribution Matters More Than Curve Form

5.1 Main Observations

Despite using different smoothing methods, the overall trend is consistent:

- at the same point count,
- adaptive sampling achieves:
    - consistently low area error
    - steadily decreasing Hausdorff distance as the point count increases

5.2 Why Hausdorff Improves

Hausdorff distance measures:

> the **maximum local deviation** between two contours

With uniform arc-length sampling:

- low-curvature regions are oversampled
- high-curvature regions are undersampled
- causing local boundary expansion/contraction

With adaptive sampling:
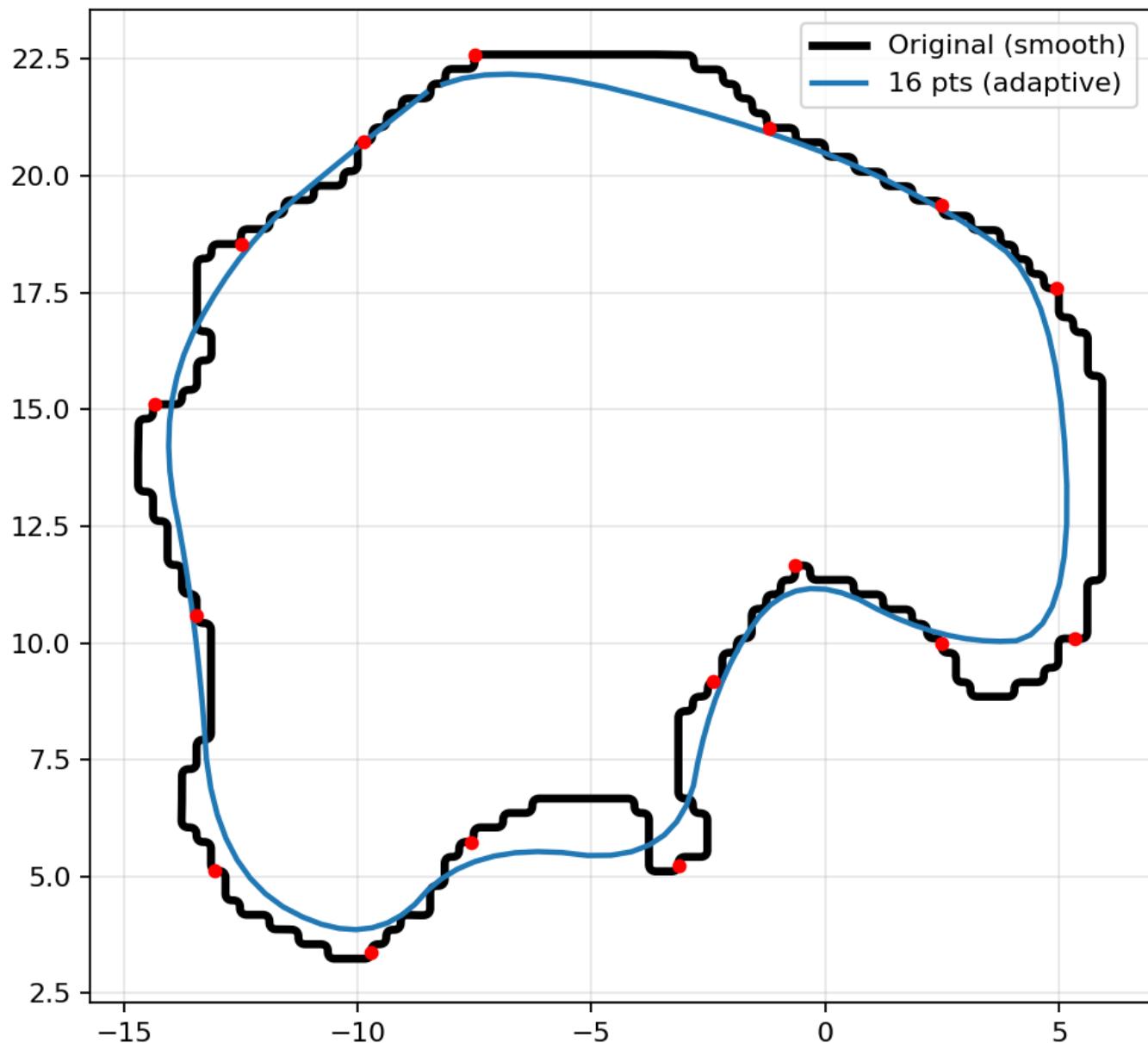
- high-curvature regions are sampled densely
- the maximum boundary deviation is better constrained
- leading to a lower Hausdorff distance

6. Visualization
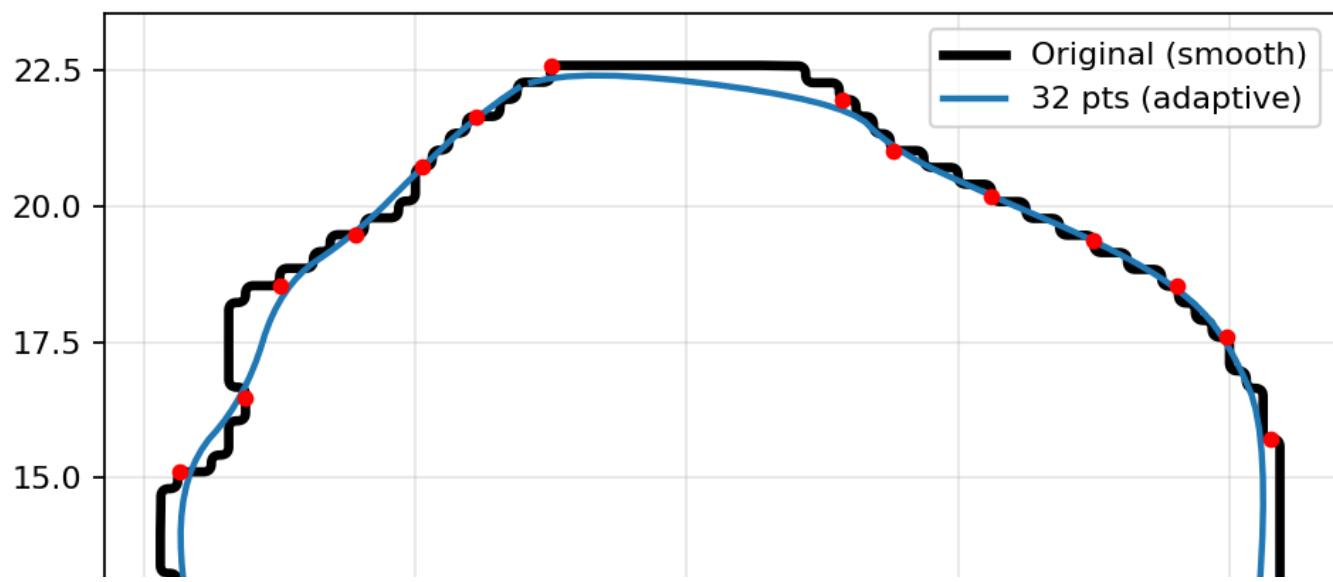
The following figures show point distribution under adaptive sampling at different point counts:
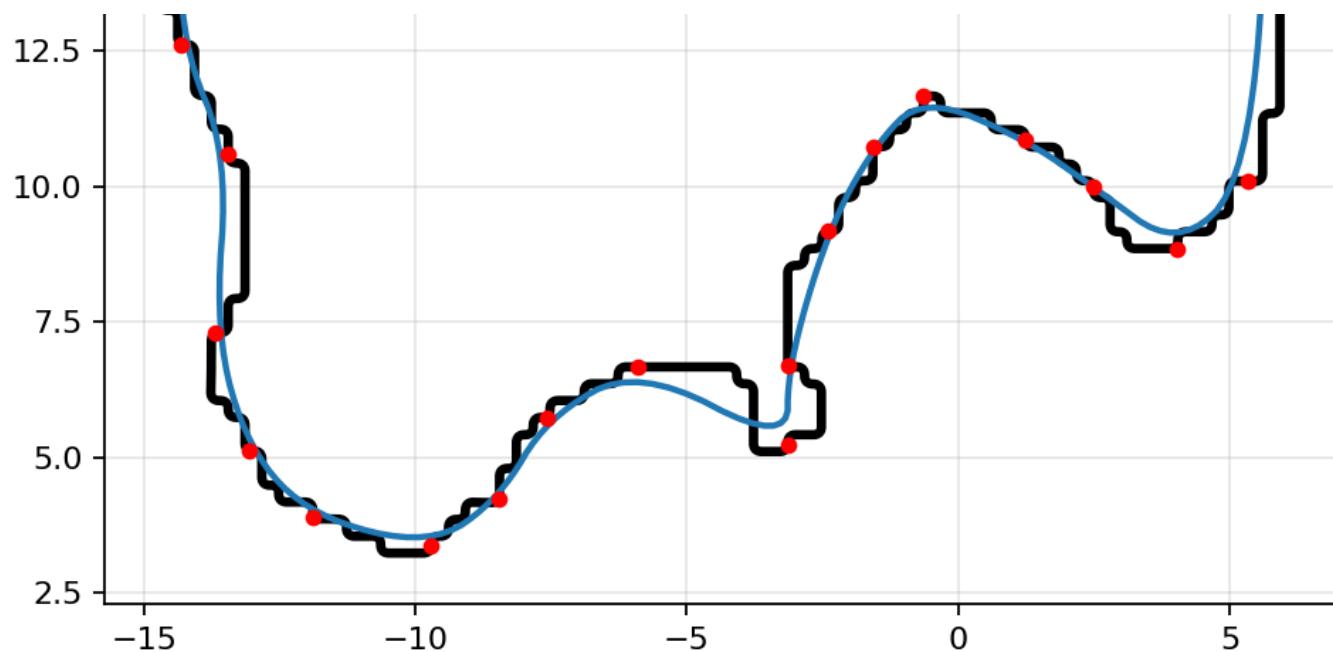
- high-turn regions: visibly denser points
- straight regions: visibly sparser points
- overall contour shape: highly consistent with the original boundary

16 pts
Area err=3.55%, Hausdorff=3.80 mm
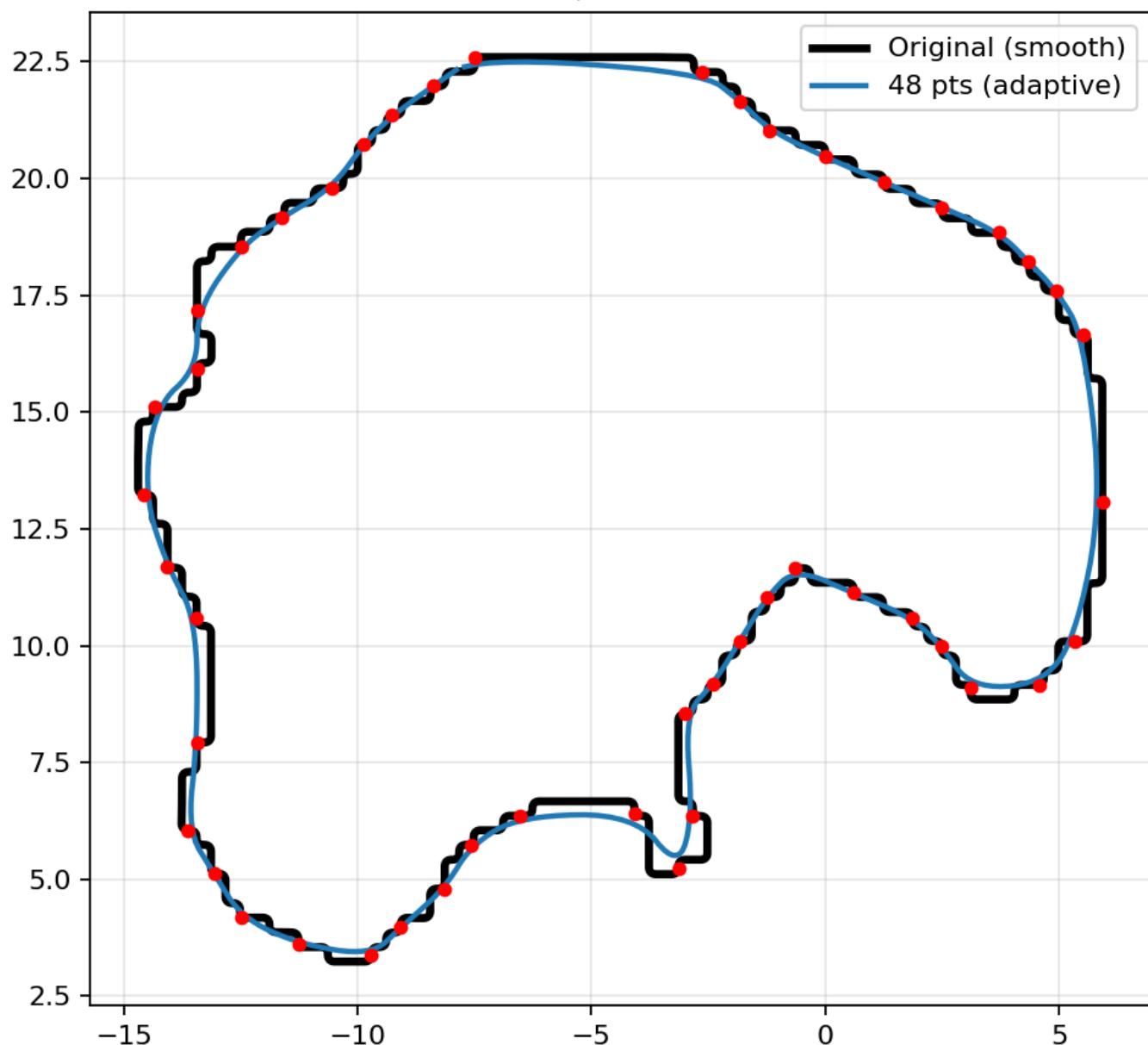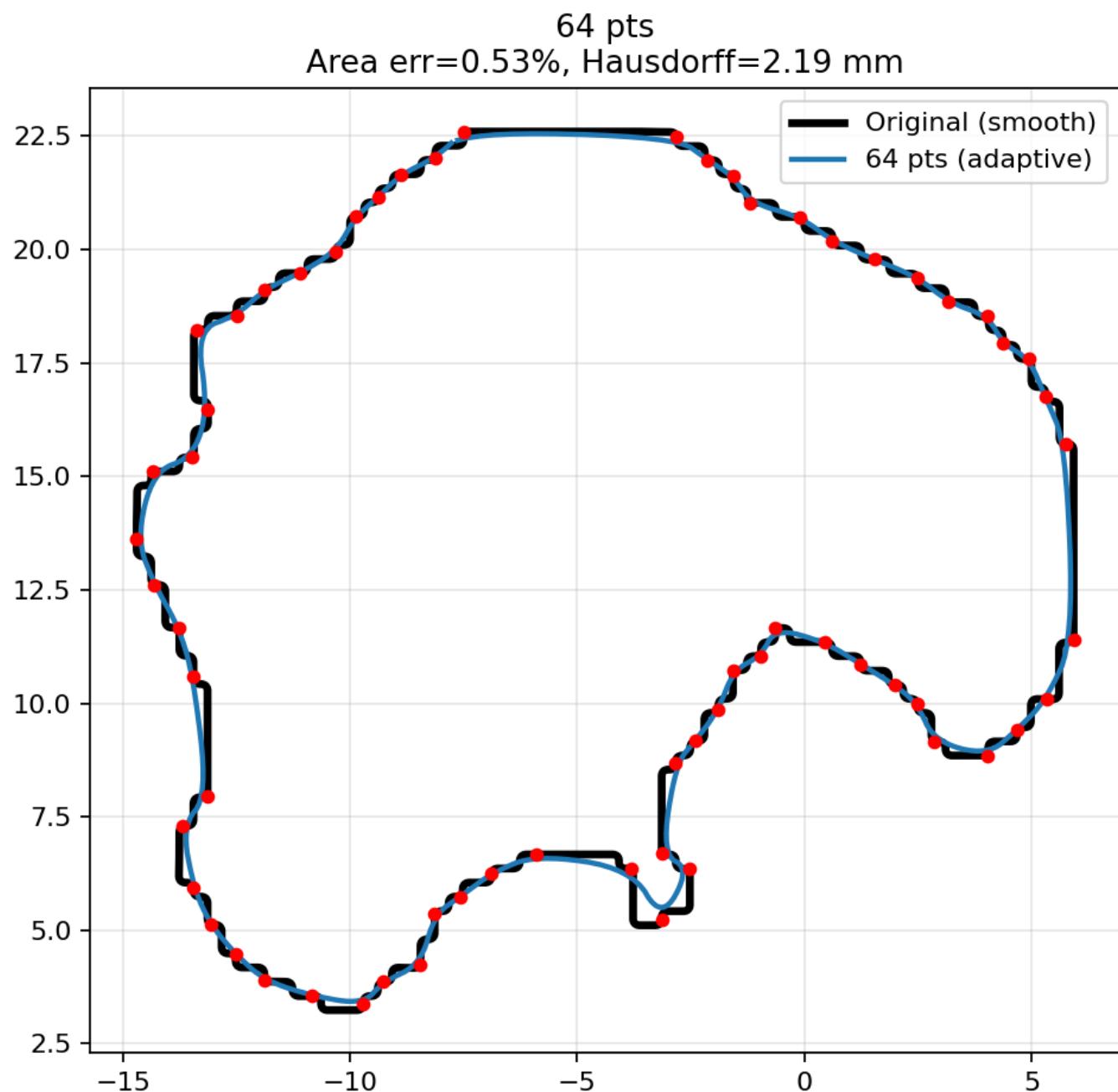
32 pts
Area err=1.16%, Hausdorff=2.83 mm
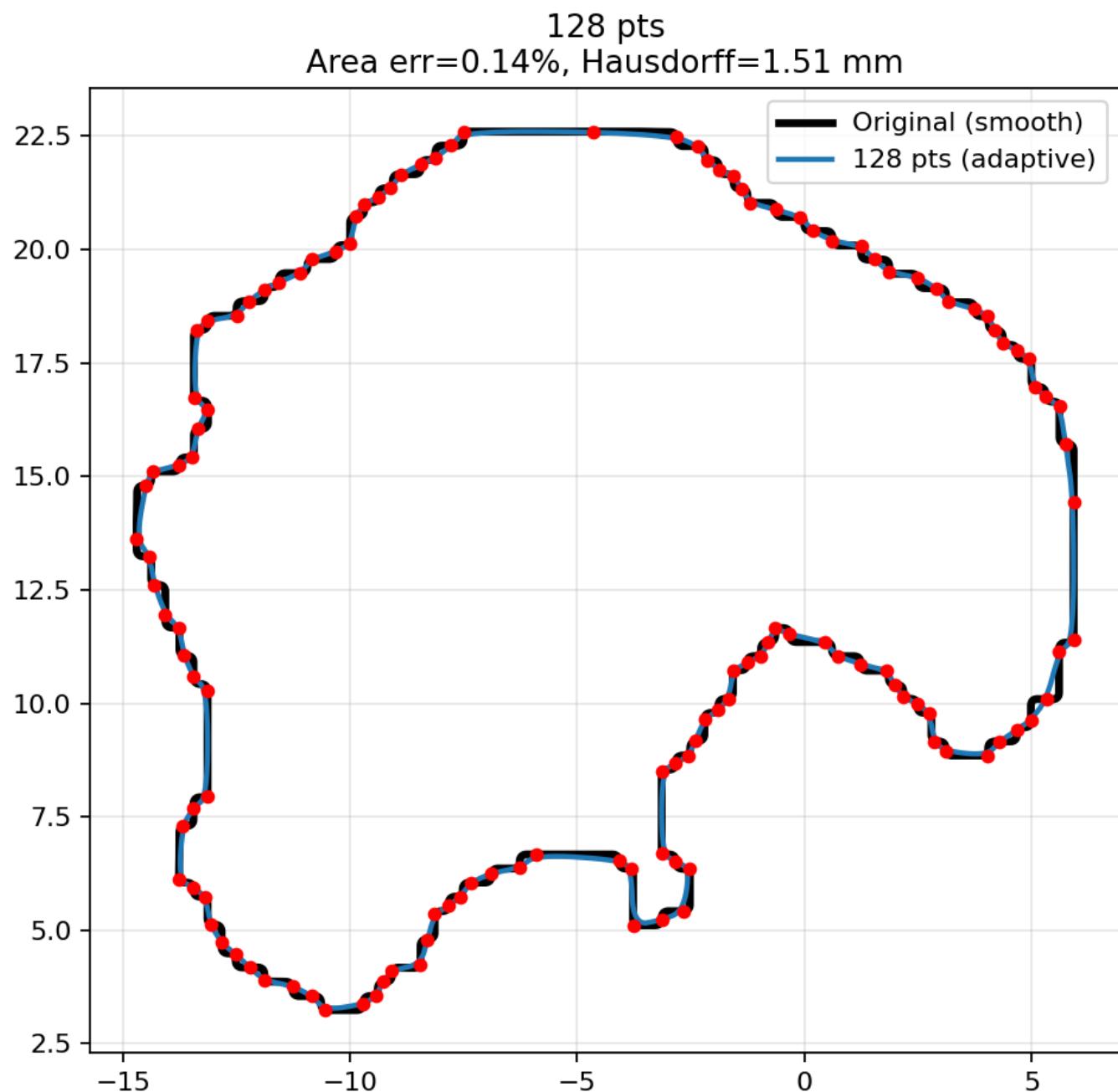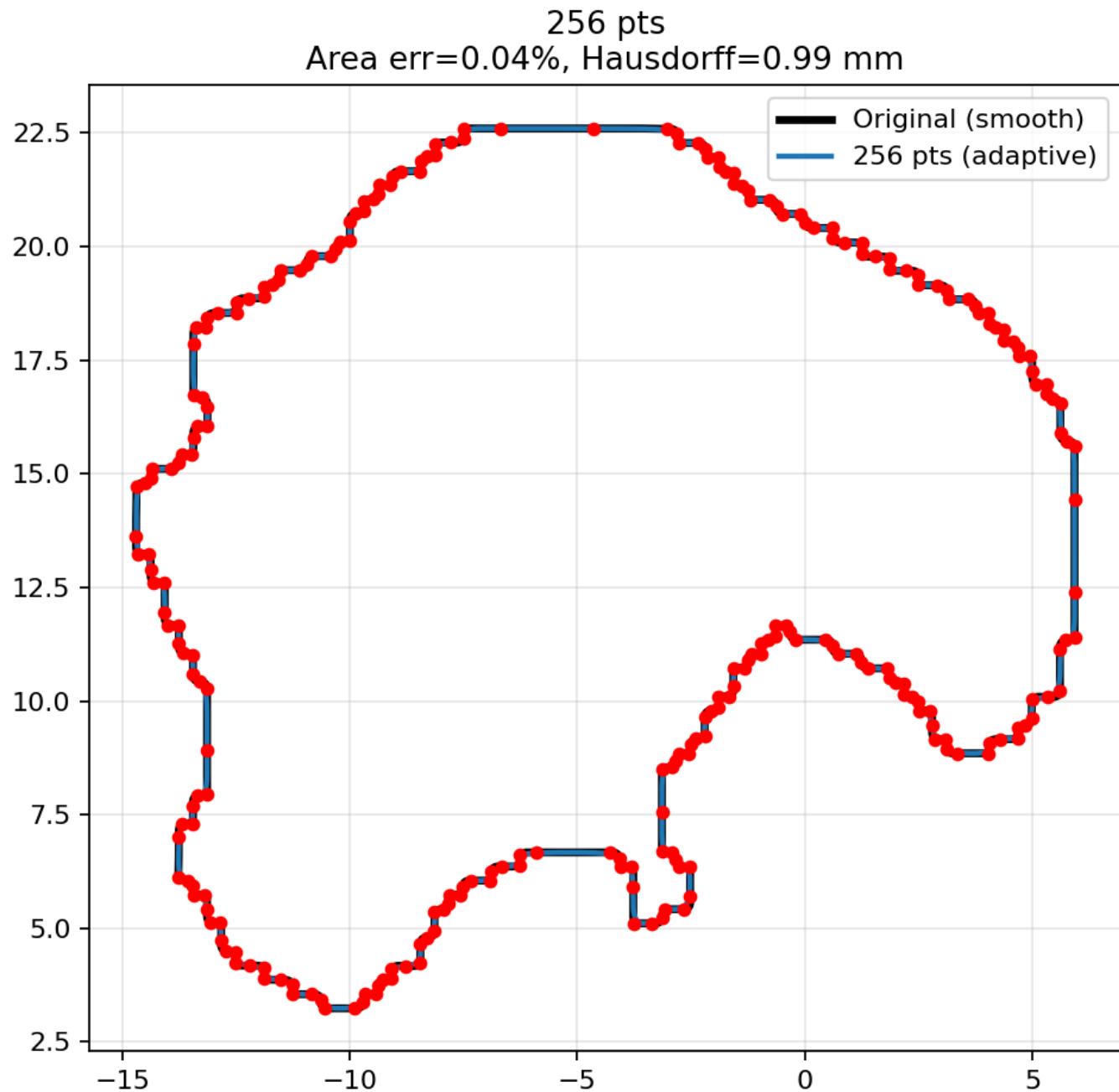
48 pts
Area err=0.27%, Hausdorff=2.40 mm

Legend:
- Original (smooth)
- 48 pts (adaptive)

64 pts
Area err=0.53%, Hausdorff=2.19 mm

128 pts
Area err=0.14%, Hausdorff=1.51 mm

256 pts
Area err=0.04%, Hausdorff=0.99 mm

7. Conclusion

- RTSTRUCT boundary points show clear non-uniform geometric complexity
- Adaptive sampling based on geometric complexity:
    - outperforms uniform sampling at the same point count
    - reduces the maximum local geometric error (Hausdorff distance)
- Smoothing primarily affects visualization, not the core geometric consistency

Question 4: Comparison Across Slices With Different Areas (small / medium / large)

This section complements the question: whether simplification behaves consistently across slices with different contour areas. Because the prostate cross-sectional area varies significantly across z-slices (small near the apex/base and larger in the middle), the same point budget may yield different error behavior.

1. Slice Selection (Consistent With Code)

Using `select_examples_by_area()` in src/area_diff.py:

- compute contour area for each slice and sort slices by `area`
- pick representative slices by area quantiles:
    - small: 10% quantile (q=0.1)
    - medium: 60% quantile (q=0.6)
    - large: 80% quantile (q=0.8)

This avoids bias from selecting only the maximum-area slice and better reflects the true distribution.

2. Metrics (Consistent With Code)

- Area relative error (%):

$$\text{AreaErr}(\%) = \frac{|A_{simp} - A_{orig}|}{A_{orig}} \times 100$$

- Hausdorff distance (mm): we use a **point-to-segment** Hausdorff distance to avoid inflated results caused by different sampling densities.

Note: the black/colored curves use Chaikin smoothing only for visualization; error metrics are computed on the simplified polyline points.

3. Results (N11780398, ROI=Prostate)

The script reports errors for `POINT_LIST = [16, 32, 48, 64, 128, 256]` for each representative slice.
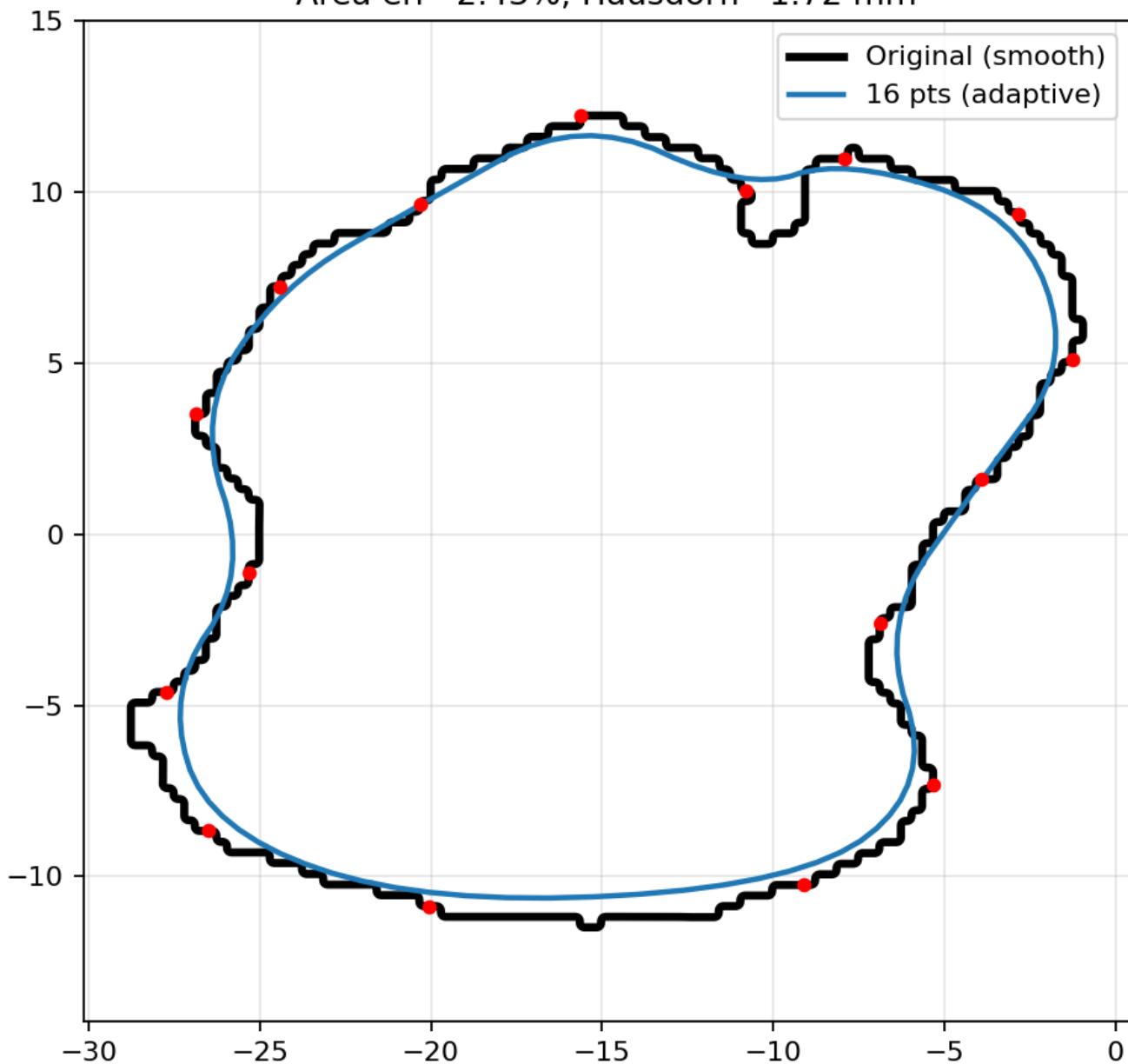
3.1 small example

Representative small slice: `area ≈ 453.42`, `z ≈ 0.66`
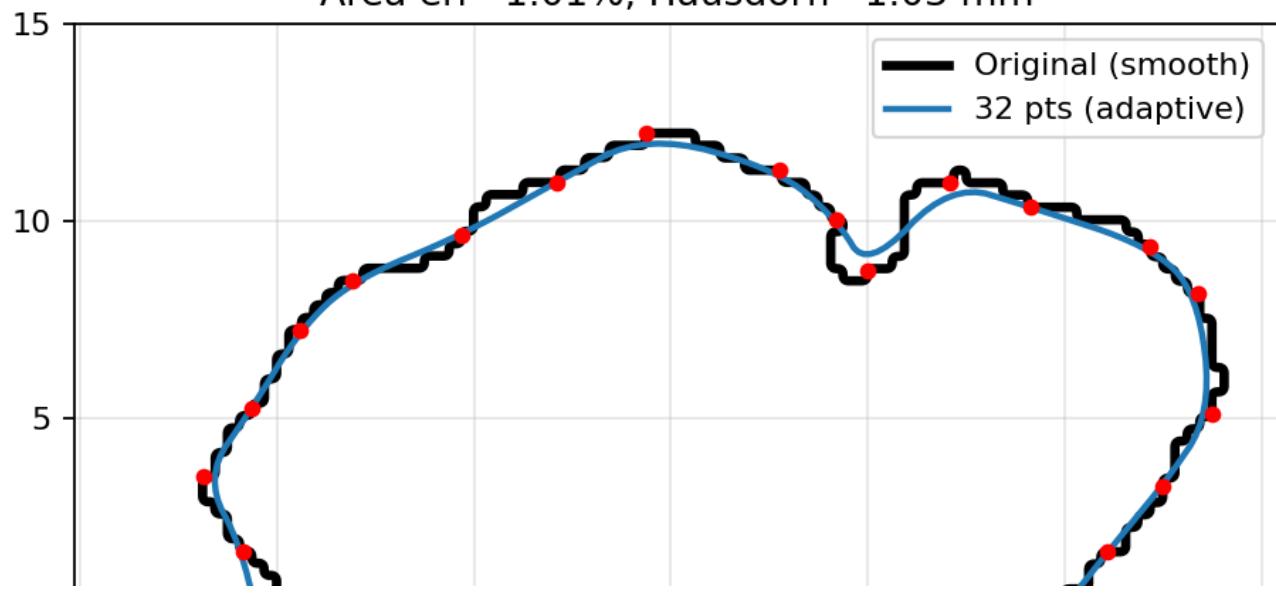
| pts	area_err (%)	Hausdorff (mm)
16	2.428	1.724
32	1.011	1.027
48	0.581	0.902
64	0.376	0.691
128	0.038	0.449
256	0.011	0.191
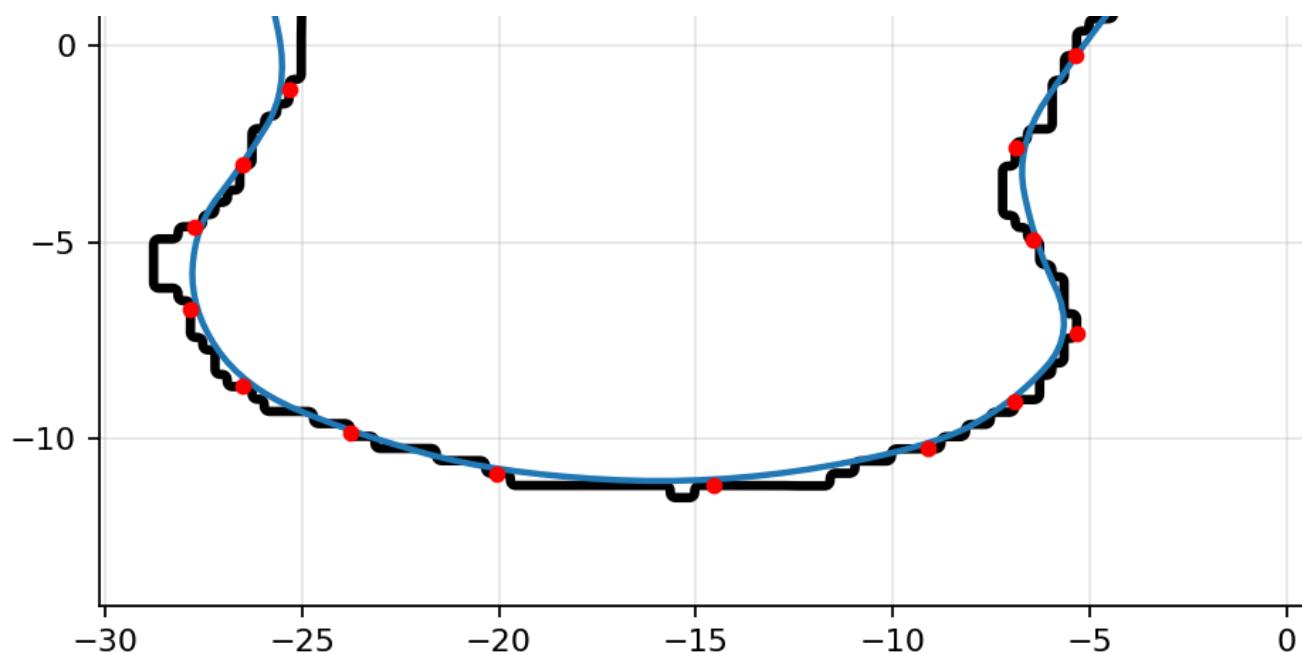
Visualization:
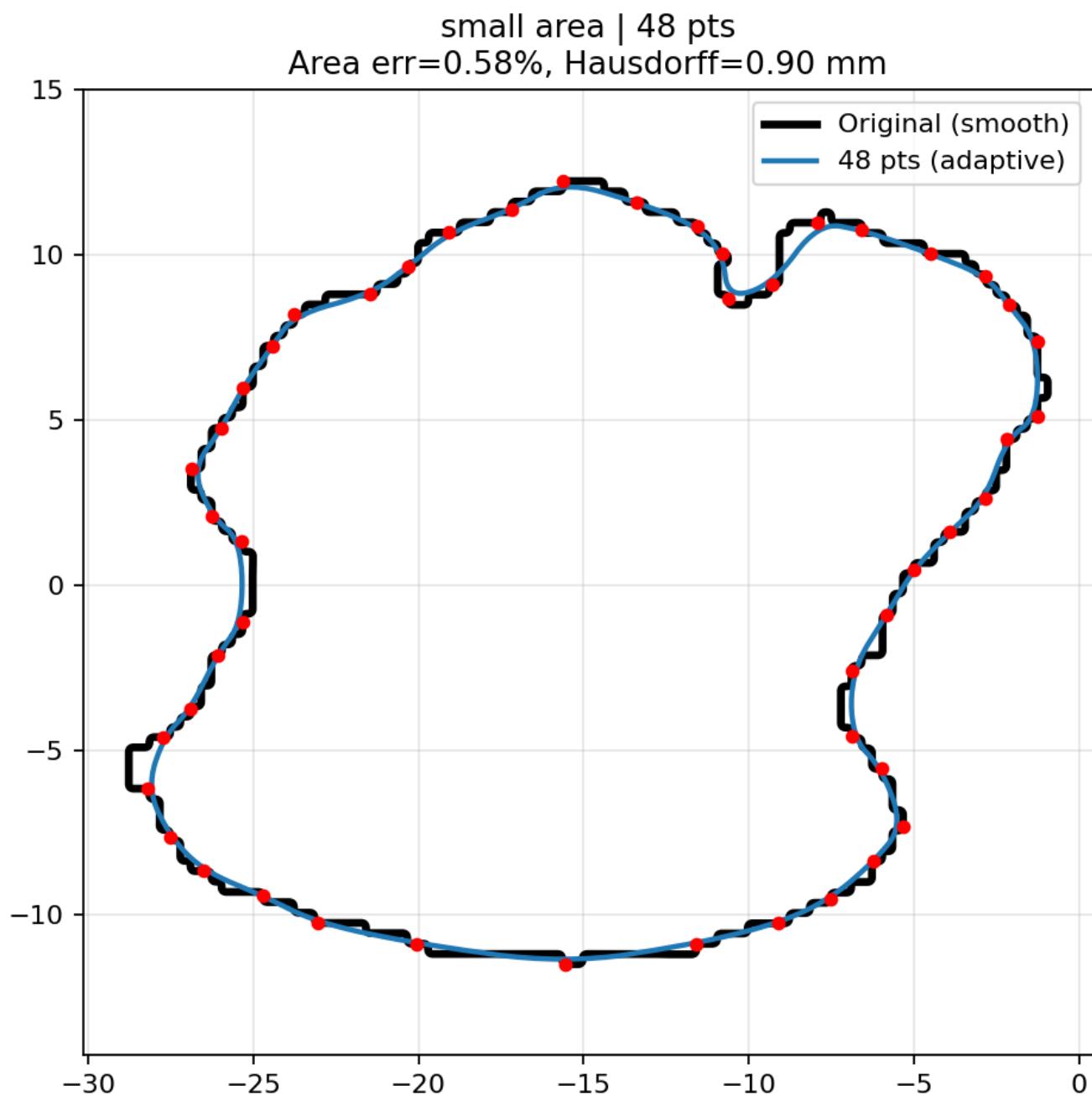
small area | 16 pts
Area err=2.43%, Hausdorff=1.72 mm
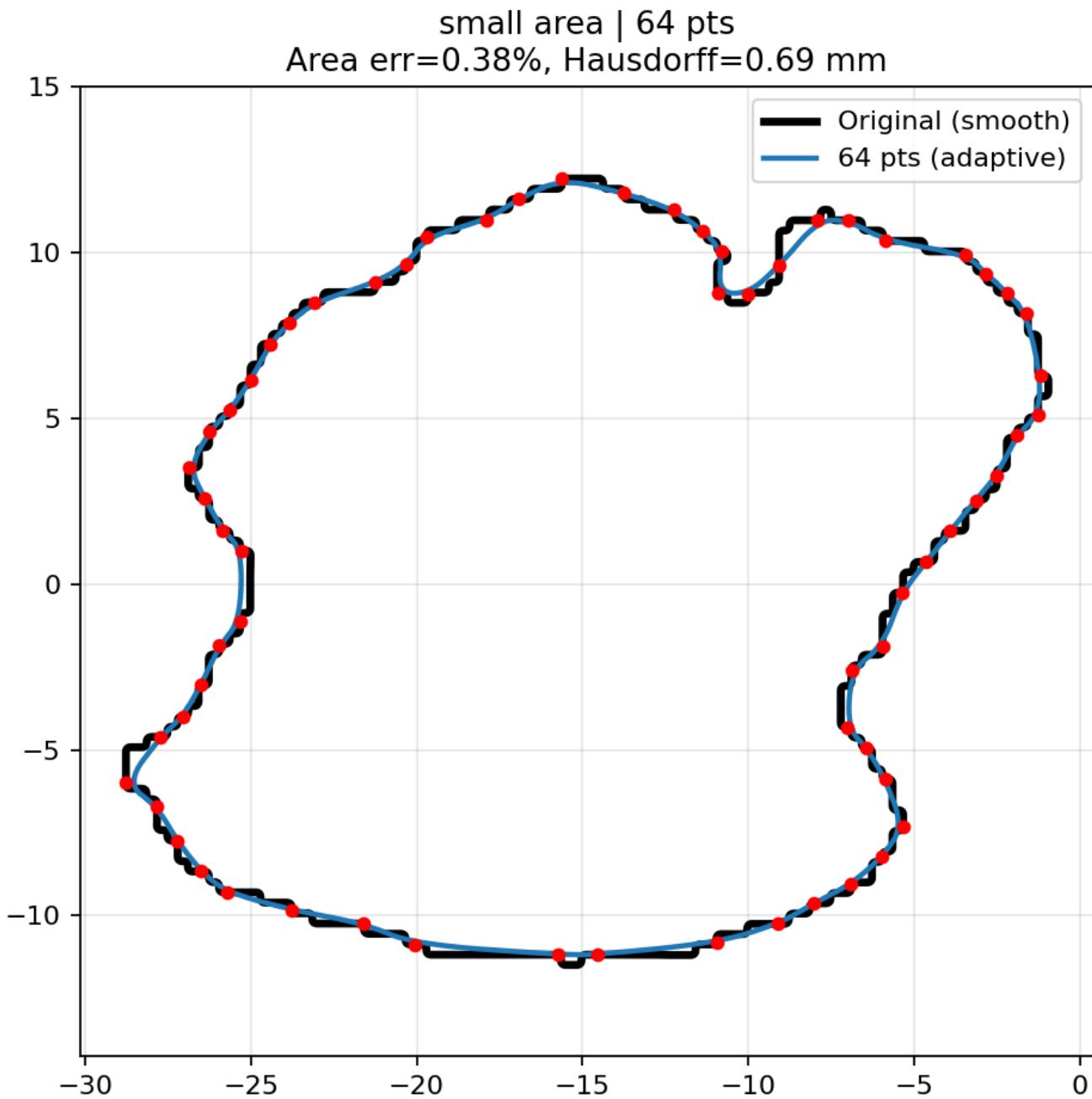
small area | 32 pts
Area err=1.01%, Hausdorff=1.03 mm

small area | 48 pts
Area err=0.58%, Hausdorff=0.90 mm

small area | 64 pts
Area err=0.38%, Hausdorff=0.69 mm
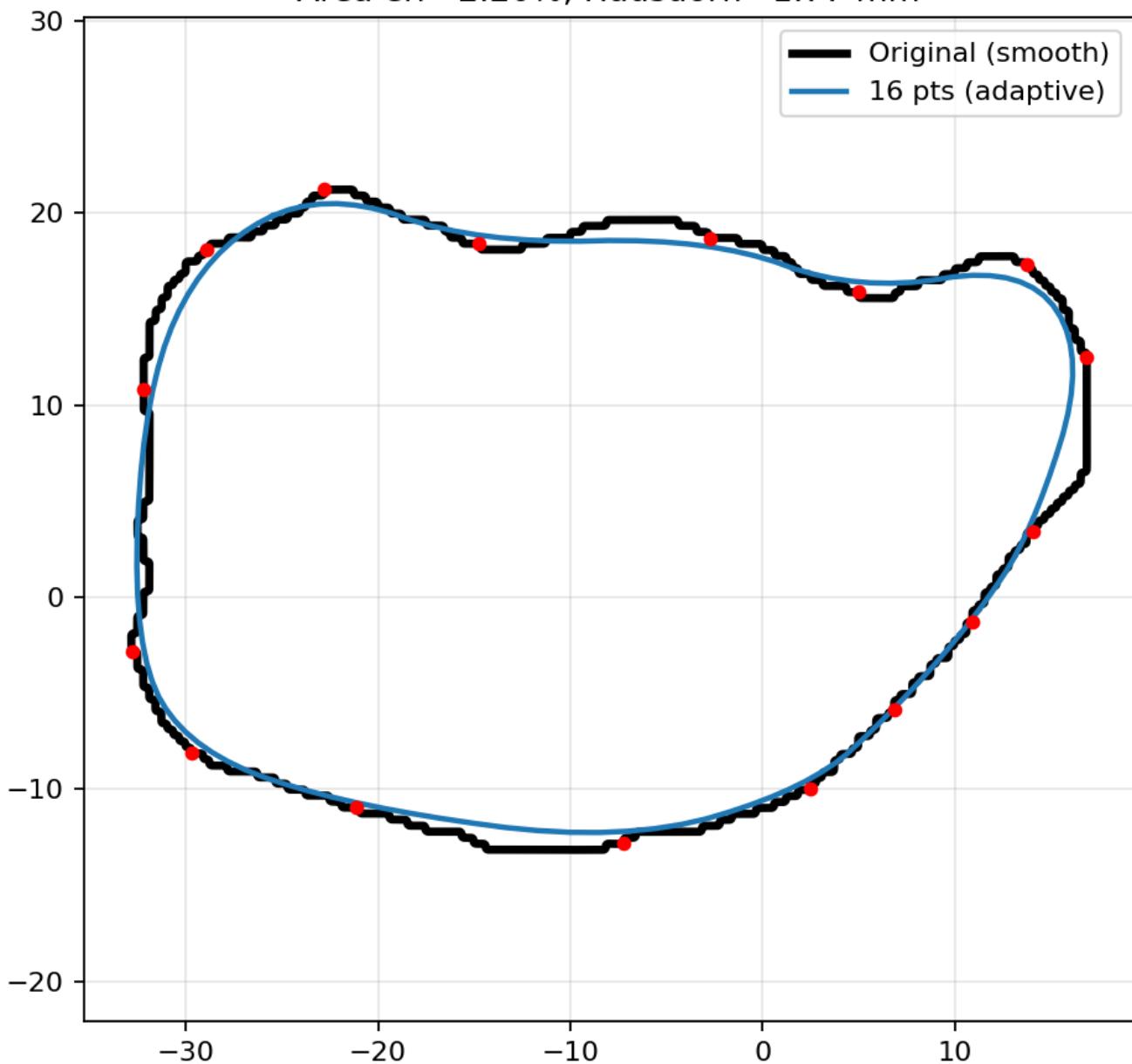
3.2 medium example

Representative medium slice: area ≈ 1303.78, z ≈ -5.02
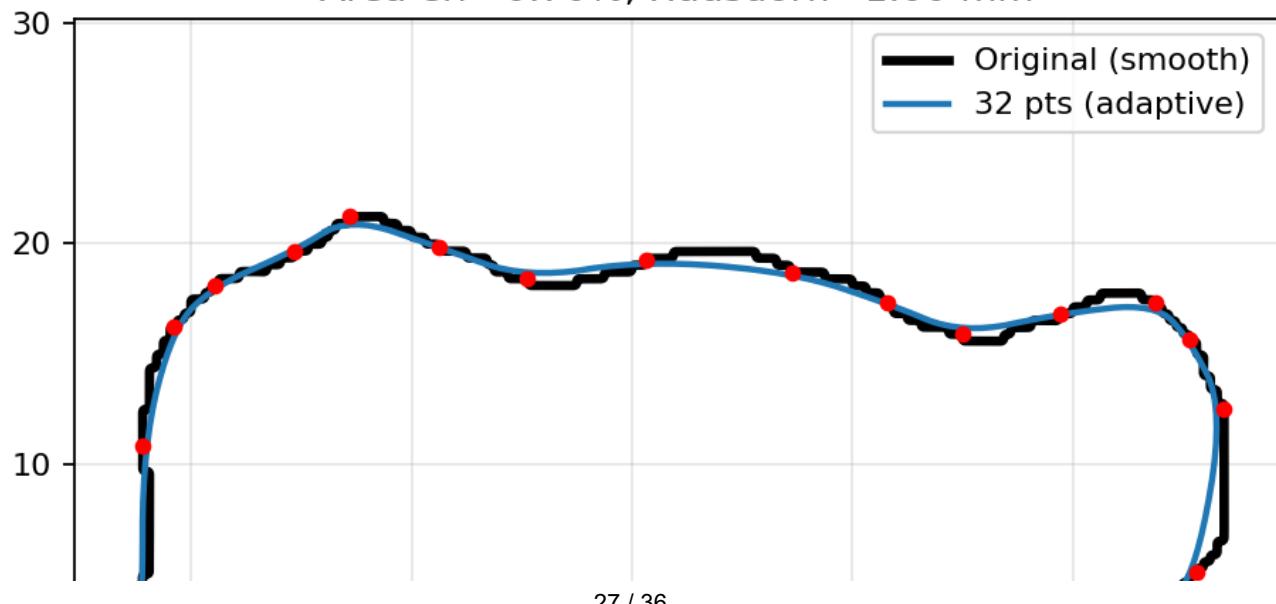
pts	area_err (%)	Hausdorff (mm)
16	2.203	1.766
32	0.695	0.996
48	0.431	0.566
64	0.196	0.497
128	0.068	0.398
256	0.009	0.254
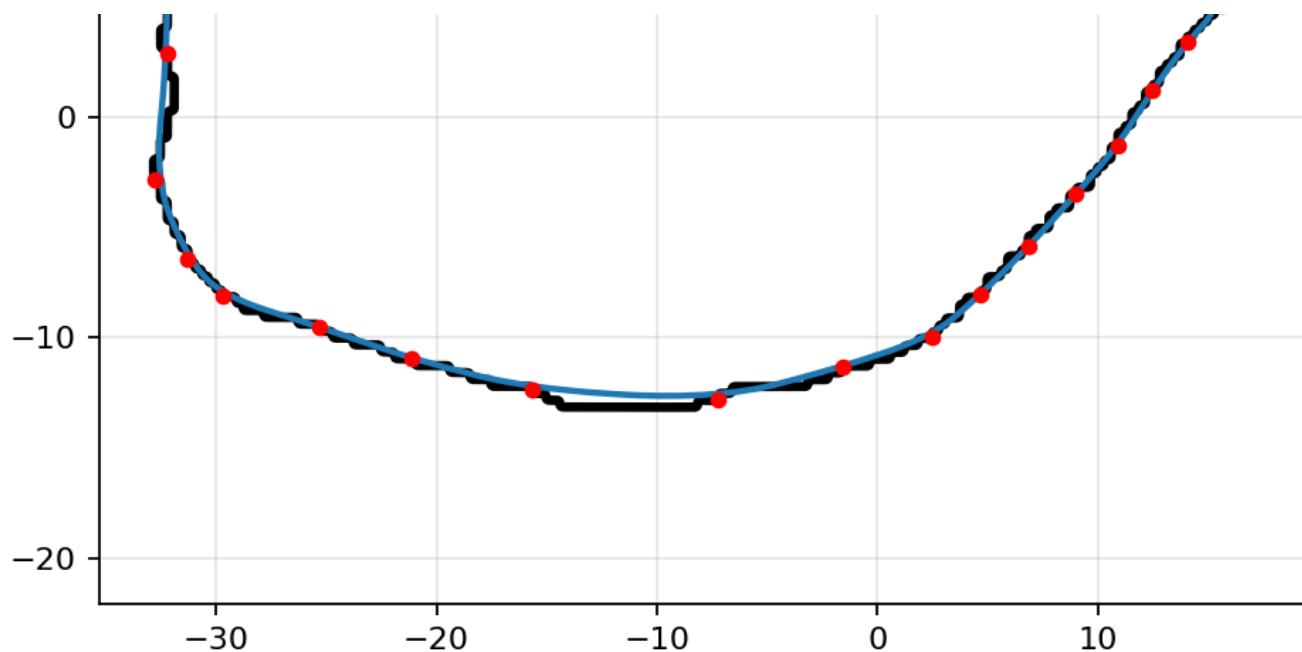
Visualization:
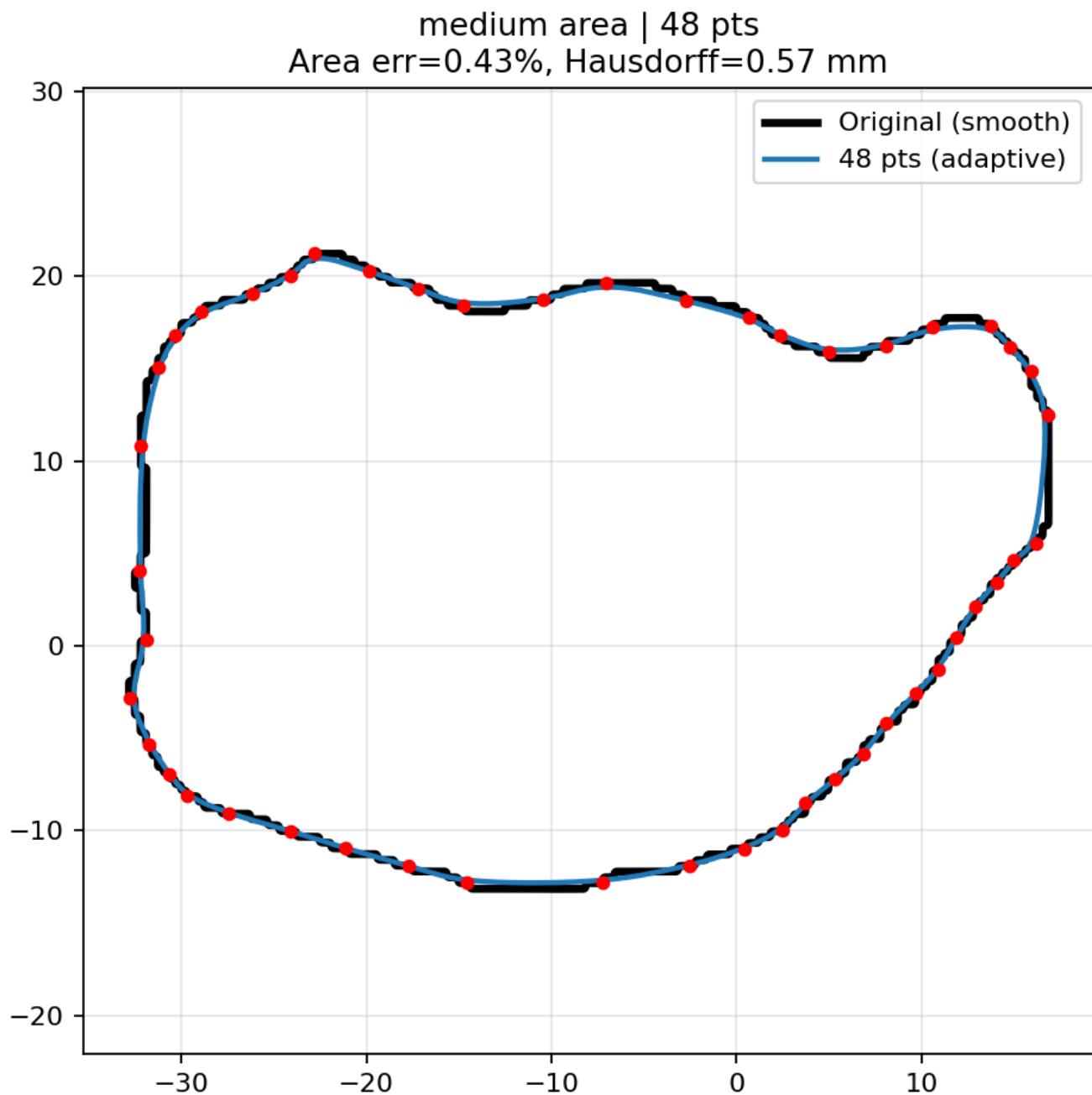
medium area | 16 pts
Area err=2.20%, Hausdorff=1.77 mm
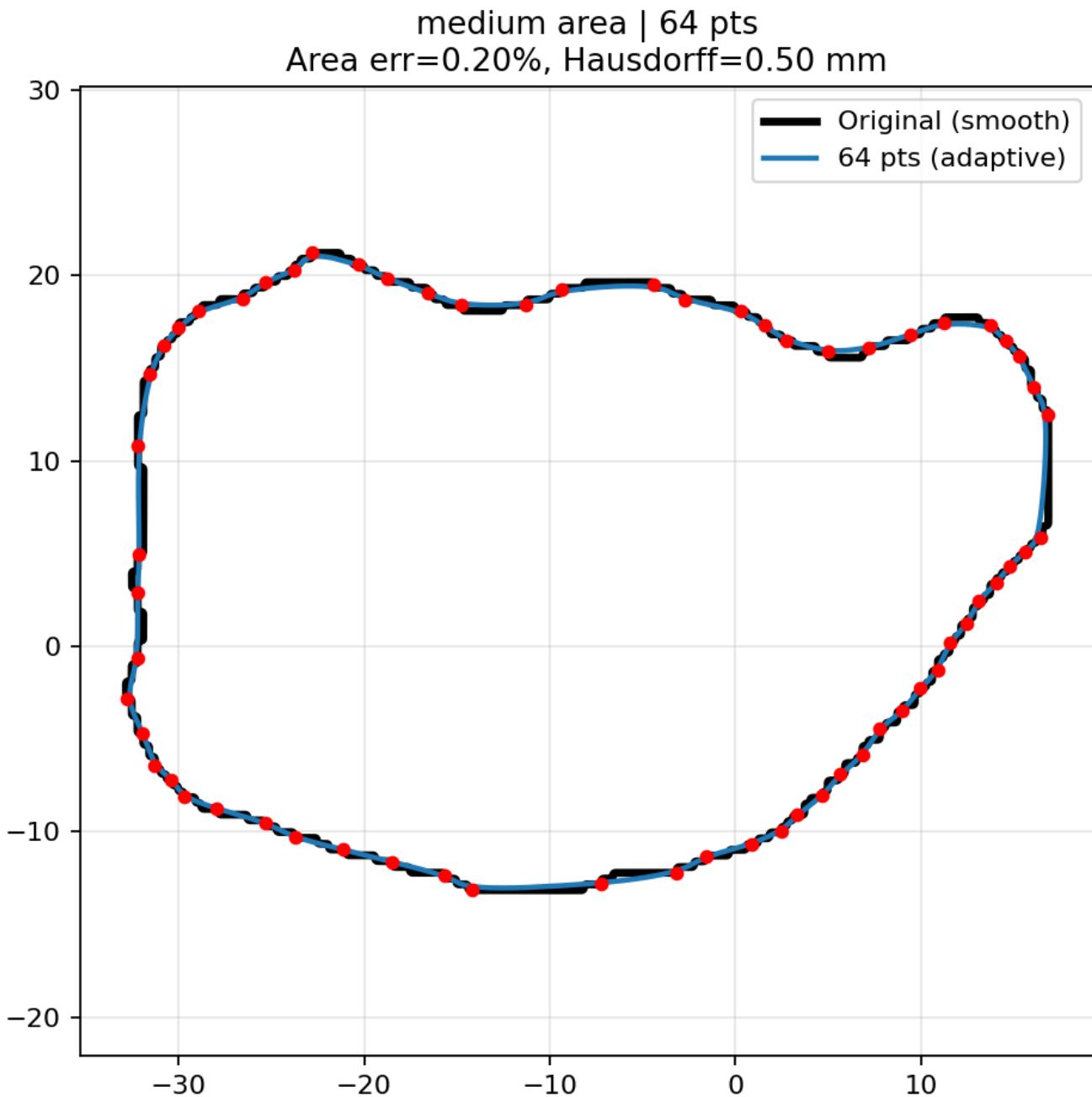
medium area | 32 pts
Area err=0.70%, Hausdorff=1.00 mm

medium area | 48 pts
Area err=0.43%, Hausdorff=0.57 mm

medium area | 64 pts
Area err=0.20%, Hausdorff=0.50 mm
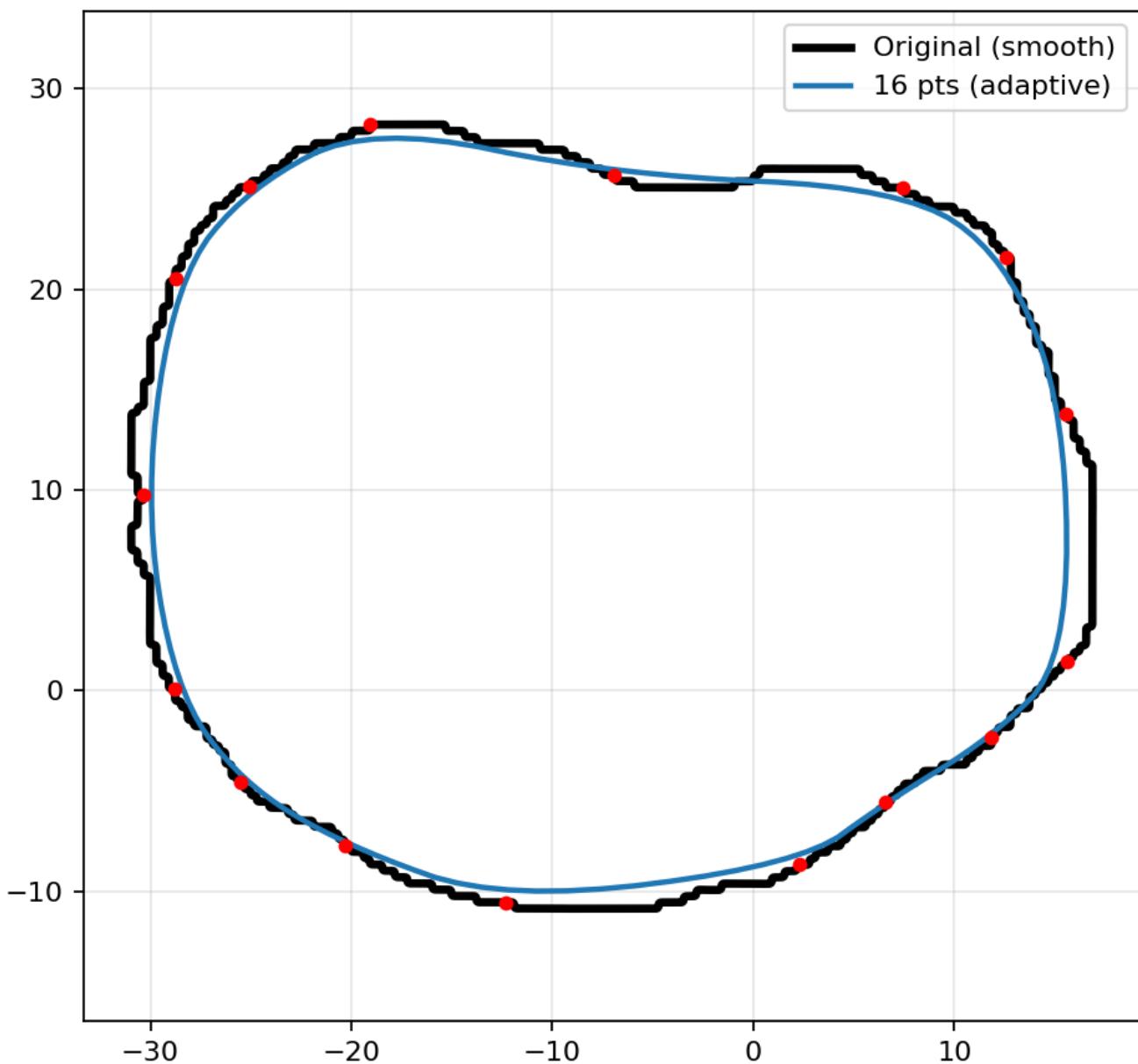
3.3 large example

Representative large slice: area ≈ 1483.18, z ≈ -16.77
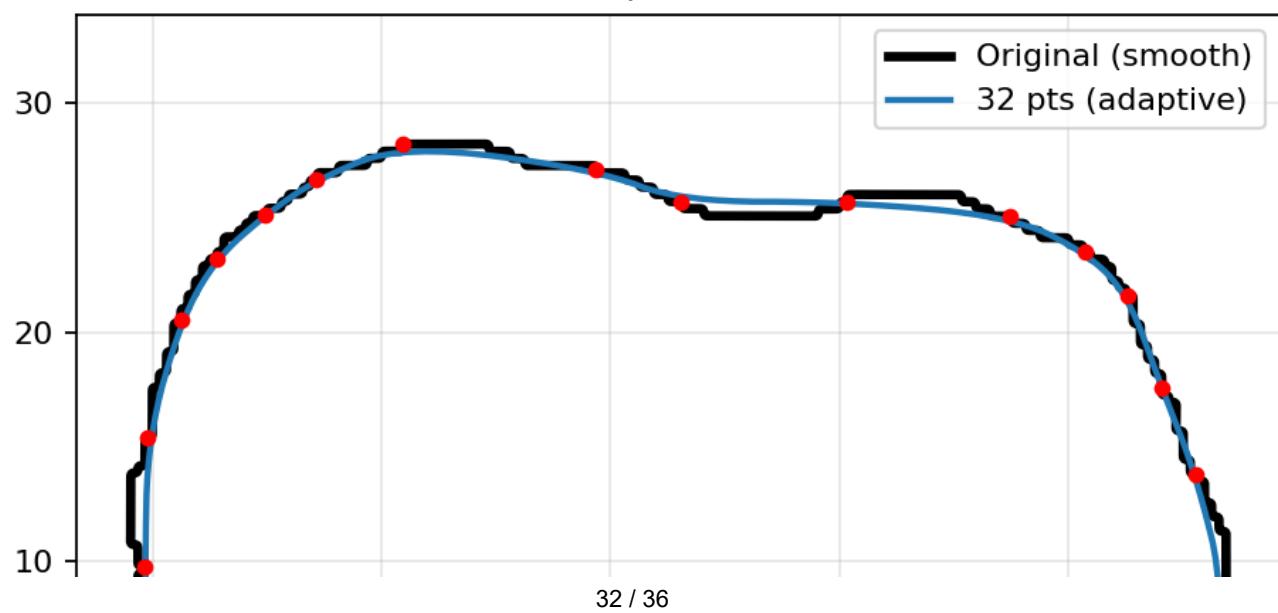
| pts	area_err (%)	Hausdorff (mm)
16	3.551	1.310
32	1.401	0.927
48	0.464	0.556
64	0.428	0.482
128	0.133	0.368
256	0.029	0.252
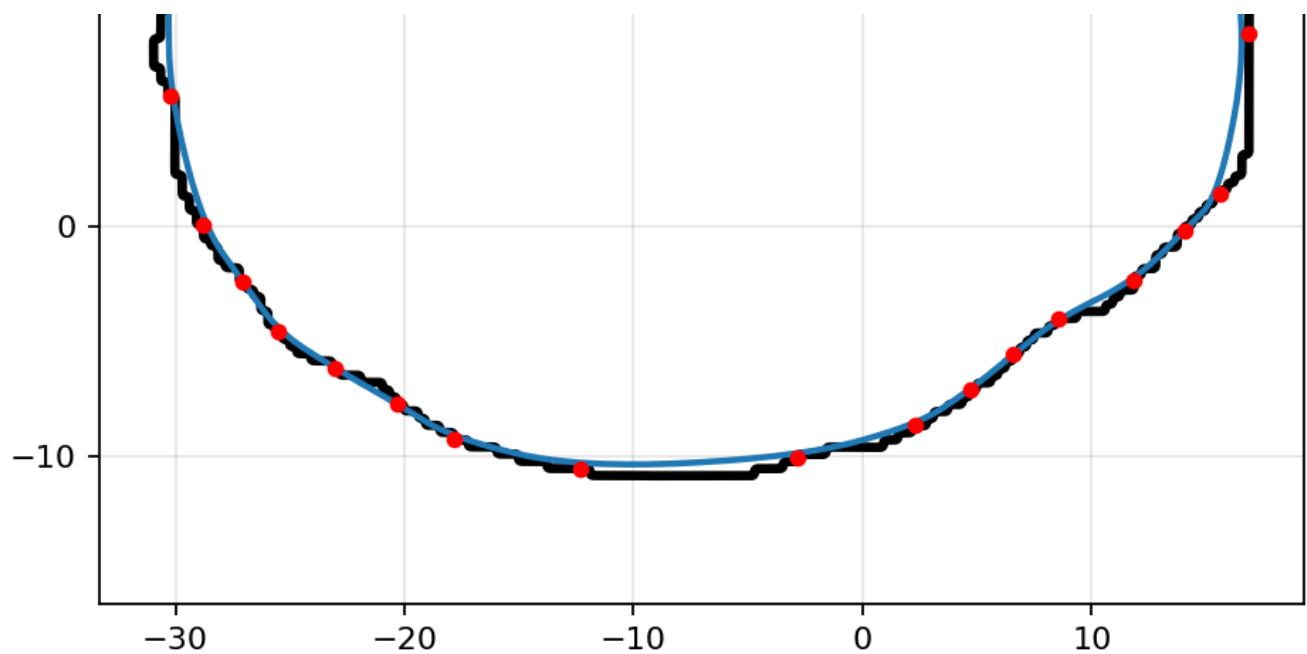
Visualization:

large area | 16 pts
Area err=3.55%, Hausdorff=1.31 mm
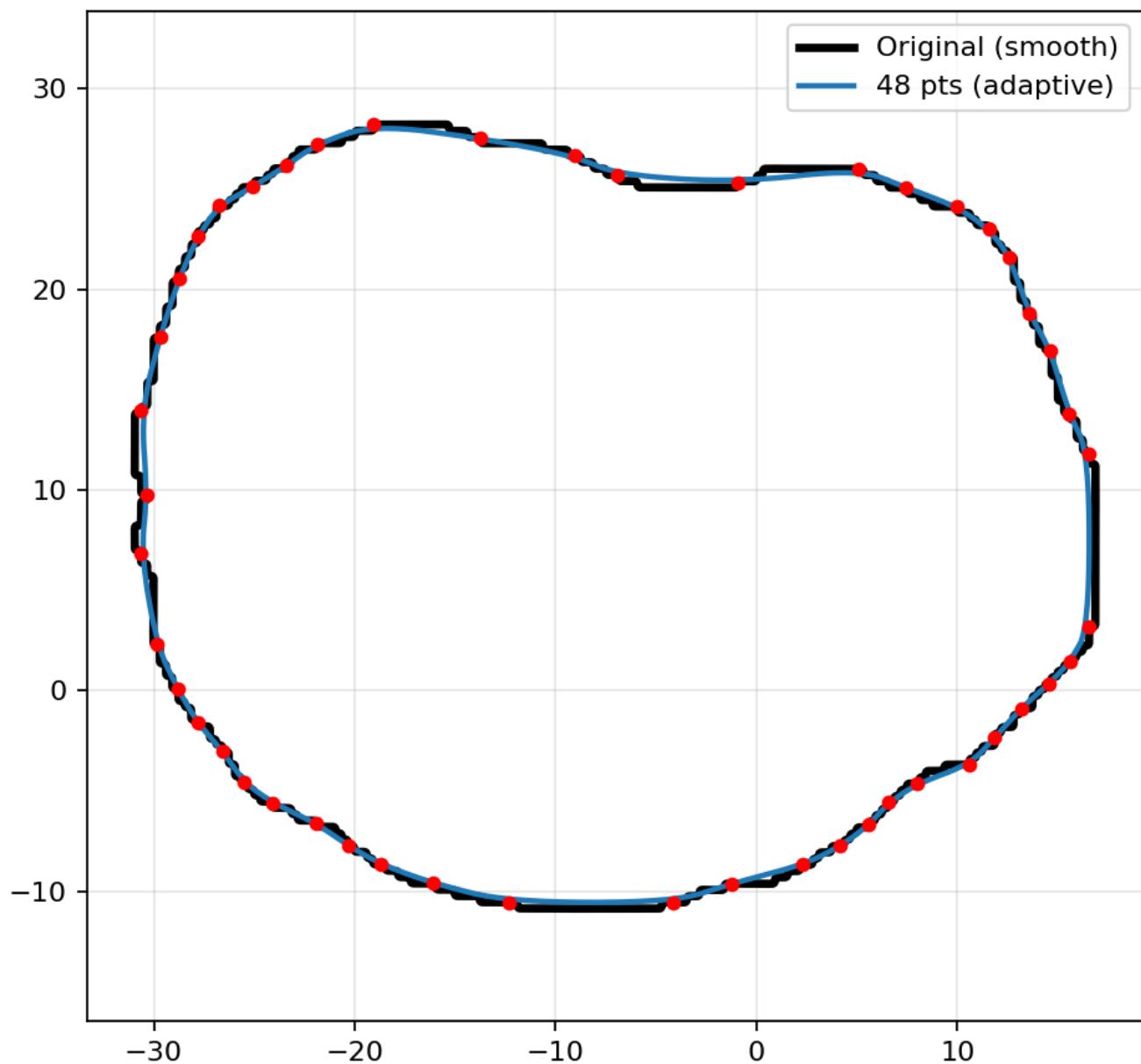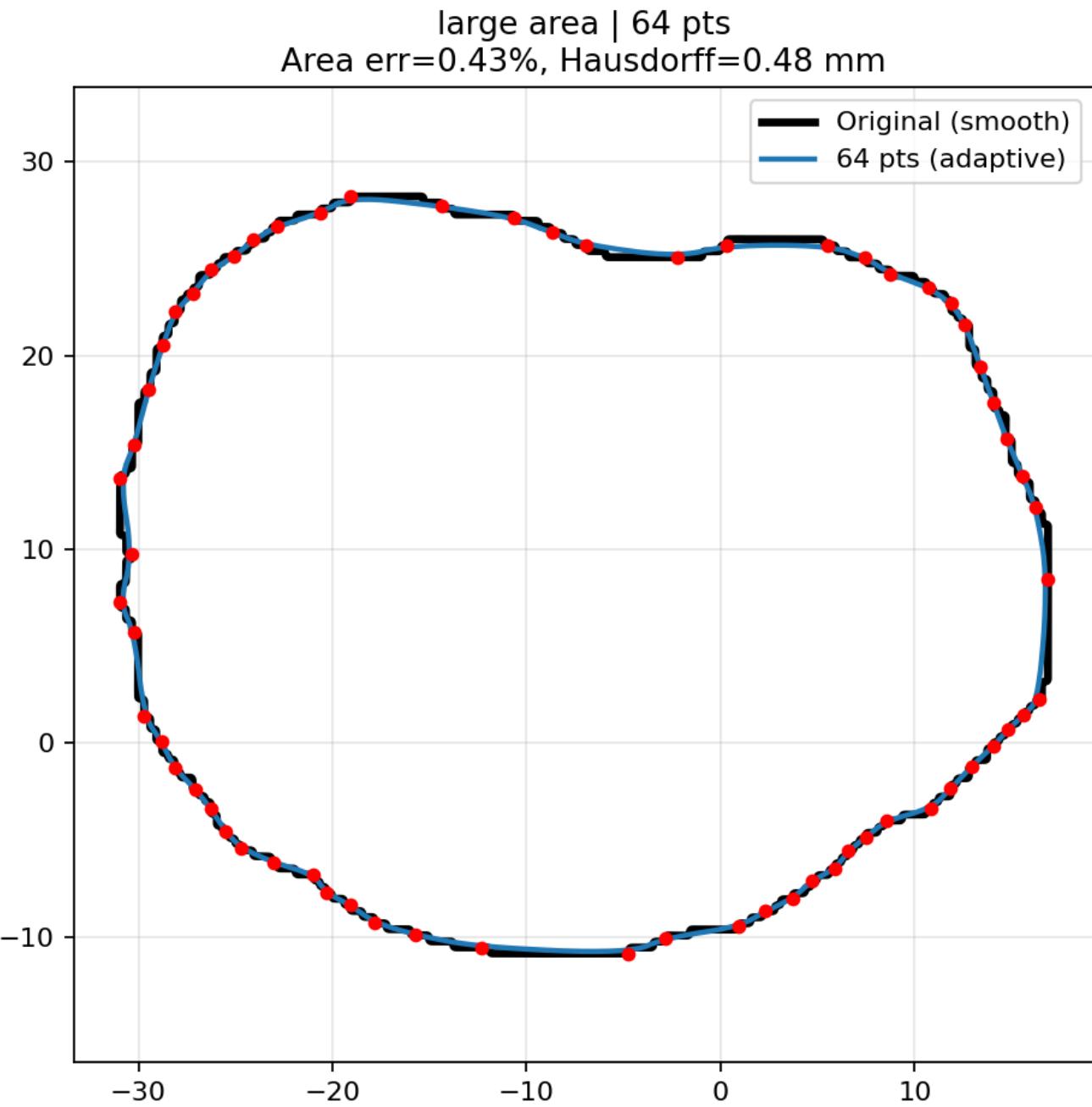
large area | 32 pts
### Area err=1.40%, Hausdorff=0.93 mm

large area | 48 pts
Area err=0.46%, Hausdorff=0.56 mm

large area | 64 pts
Area err=0.43%, Hausdorff=0.48 mm

4. Conclusion: Error Differences Across Slice Areas

Two takeaways from comparing small / medium / large:

1. **Area error does not uniquely determine shape error**: for example, in the small slice at 64 points, the area error is only **0.376%**, but the Hausdorff distance is still **0.691 mm**. This means the global area is already close, while some local boundary regions can still deviate noticeably.
2. **The minimal acceptable point count depends on thresholds and slice type**:
   - With Hausdorff ≤ 0.5 mm, medium/large are largely acceptable at 64 points, while the small slice often needs a higher point count (e.g., 128) to reliably fall below 0.5 mm.

Therefore, if a single unified point count is required, it should be chosen based on the study goal:

- for global quantities (area/volume) stability: a smaller point count may suffice (e.g., 48–64)
- for sensitivity to maximum boundary deviation (dosimetry / geometry-sensitive use cases): a more conservative point count (e.g., ≥128), or an adaptive policy that allocates more points to small slices