

PolyTopoBench: A Benchmark for Complex Vector Polygon Generation from Remote Sensing Imagery

Zeping Liu¹ Ni Lao¹ Weiwei Sun² Gil Wolff²
Yiqun Xie³ Liang Zhao⁴ Junfeng Jiao¹ Gengchen Mai^{1,*}

¹The University of Texas at Austin ²Amazon ³University of Maryland ⁴Emory University

{zeping.liu, nlao}@utexas.edu, weiwei.sun3@gmail.com, wolffg@amazon.com,
xie@umd.edu, liang.zhao@emory.edu, {jjiao, gengchen.mai}@austin.utexas.edu

Abstract

Vector polygon generation converts visual inputs, e.g., remote sensing (RS) images, into vectorized polygonal geometries, supporting applications such as autonomous driving, vector map construction, and remote sensing. Early pipelines predict raster masks and post-process them into polygons, which prevents end-to-end optimization and may miss small objects or introduce inaccurate vertices. Recent methods directly generate vector polygons, but most focus on simple exterior contours, while they either cannot represent complex polygons with holes or fail to preserve their topology. In this paper, we propose **PolyTopoBench**, a unified evaluation framework for vector polygon generation from RS images with explicit emphasis on complex polygons. PolyTopoBench evaluates both exterior and interior rings, and benchmarks 11 representative methods, including segmentation-based polygonization pipelines, vision foundation model baselines, and specialized vector polygon generators, on two RS-image datasets covering buildings, roads, vegetation, and unvegetated regions. Experiments show that existing methods often recover simple exterior boundaries but degrade substantially on polygons with holes or multiple rings. These results reveal complex polygon generation as an unresolved challenge and motivate topology-aware benchmarks and model designs. Code and data are available at <https://github.com/seai-lab/PolyTopoBench>.

1 Introduction

Vector polygon generation from remote sensing images converts visual observations into editable geometric vectors, and has many applications such as urban planning [25, 21], environmental monitoring [53, 3], and disaster response [34, 46]. A useful polygon prediction must therefore capture not only the visible extent of an object, but also the structure that makes the geometry valid and interpretable. In this paper, we distinguish between *simple polygons*, comprising a single exterior ring, and *complex polygons*, each of which consists of one exterior ring and one or more interior rings (holes). This distinction is especially important because complex polygons encode essential topological and semantic information that cannot be recovered merely from the exterior boundary alone. Interior rings indicate regions that are explicitly excluded from the current object, and their presence often corresponds to meaningful real-world spatial structure rather than noise. For example, buildings may contain courtyards or enclosed voids, roads may encircle traffic islands or land parcels, and land parcels may contain holes induced by other classes. Missing these hole structures fundamentally misrepresents the spatial extent and function of the object, and can introduce systematic errors in downstream tasks such as area and shape descriptor computation [47], spatial relation prediction [31, 15], and geographic question answering [48, 5, 30].

Despite the importance of this multi-ring representation, most existing polygon-generation methods [10, 54, 17] and evaluation benchmarks [29, 14, 39] still **treat polygonization mainly as an**

*Corresponding author.

exterior-boundary problem. Classical pipelines predict raster masks and then simplify their contours into polygons [51, 55] or use simplified bounding boxes [44], which often results in heavy artifacts, such as irregular and inaccurate boundaries due to the lack of explicit geometric modeling. To address these, recent specialized polygon-generation methods learn vertex [45, 17, 16], contour [49, 43], frame-field [10], graph [54], or token-sequence [2] representations to directly generate vector outputs. These models achieve strong results on common benchmarks, especially for building-footprint extraction. However, these standard evaluation settings remain limited in three ways. The targeted geospatial objects are often dominated by buildings, whose instances are comparatively compact and frequently simple. Many benchmarks [29, 14, 28, 39] contain few complex polygons with interior rings, making it difficult to evaluate whether methods can handle multi-ring topology. Furthermore, many existing evaluations [23, 7, 4, 10] often reduce polygon geometry to a single closed contour, without explicitly distinguishing exterior from interior rings, or simply omitting interior rings altogether. As a result, a method can appear successful under common scores while producing geometries that are incomplete or topologically incorrect for downstream geospatial applications.

These limitations call for benchmarks that explicitly evaluates models’ capabilities to generate complete multi-ring vector geometries, rather than simple exterior contours. To this end, we introduce **PolyTopoBench**, a benchmark and evaluation framework for complex vector polygon generation from remote-sensing imagery. To the best of our knowledge, PolyTopoBench is the first benchmark in remote-sensing polygon generation that places complex polygons with interior rings, rather than only simple polygons or exterior boundaries, at the center of evaluation. It goes beyond the common building and crop field-only setting [29, 14, 39, 19] by covering four object categories: buildings, roads, vegetation, and unvegetated regions. Based on Inria buildings [29] and Deventer land-cover polygons [17], we incorporate OpenStreetMap [35] annotations and manually correct them to align with the imagery and recover missing interior rings. The resulting benchmark contains nearly 300K polygon instances, more than 10K complex polygons, and over 42K interior rings.

PolyTopoBench further includes 11 diverse and publicly available vector polygon generation models, spanning segmentation-then-polygonization pipelines [38, 11, 55], vision foundation model-based polygonization [33], representation-based polygon models [45, 17, 10, 49, 43], and direct polygon generation methods [2, 54, 16]. Finally, PolyTopoBench introduces hole-aware evaluation metrics that separately assess overall polygon geometries, exterior rings, and interior rings for more robust topology evaluation.

In summary, our contributions are as follows:

- We introduce PolyTopoBench, a unified remote-sensing benchmark that places *complex* vector polygon generation, rather than simple exterior-boundary extraction.
- We construct a large-scale and multi-category benchmark with nearly 300K polygon instances, over 10K complex polygons, more than 42K interior rings, and four classes of geospatial objects.
- We provide a comprehensive and reproducible comparison of 11 diverse, publicly available vector polygon generation models, and propose a hole-aware evaluation protocol that separately assesses exterior and interior rings through ring-aware boundary metrics.
- We reveal a shared limitation of current polygon-generation methods – most existing evaluation metrics for polygon generation models focus mainly on region overlap or exterior-boundary quality, which are insufficient since they do not guarantee correct interior rings; and no existing method reliably solves complex polygon generation problem based on our evaluation.

2 Related Work

2.1 Vector Polygon Generation

Vector polygon generation has been studied broadly in computer vision, evolving from raster post-processing to structured vector prediction. Early pipelines typically follow a segmentation-then-polygonization paradigm, where semantic segmentation [41] or instance segmentation models [11] first produce raster masks, and post-processing algorithms then extract, simplify, or regularize contours into vector polygons [51, 55]. Beyond mask-based pipelines, later methods model object boundaries more explicitly, including contour-deformation methods that deform initialized contours toward object boundaries [18, 36, 22, 27], RNN-based methods that sequentially predict ordered vertices [6, 1], graph-based methods that represent object boundaries as graphs and iteratively refine polygon vertices and edges [24]. These general-purpose vision methods provide important foundations for editable vector output, but they are not specifically optimized for geospatial objects, whose boundaries can be long, irregular, multi-scale, and topologically complex [52, 9].

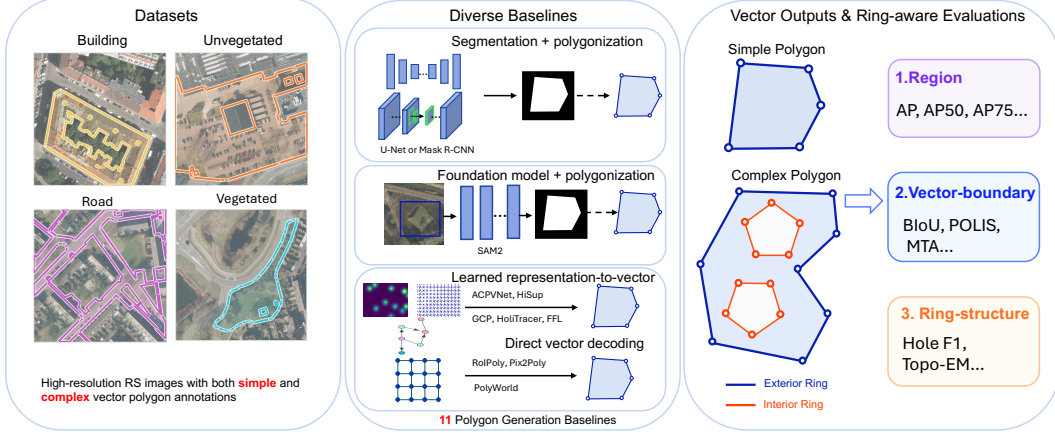


Figure 1: **Overview of PolyTopoBench.** PolyTopoBench evaluates complex vector polygon generation from remote sensing imagery. It consists of diverse remote sensing imagery datasets with simple and complex polygon annotations on different geographic features, including *buildings*, *roads*, *unvegetated regions*, and *vegetated regions*. It supports eleven diverse polygon generation baselines. A unified evaluator is developed, which jointly considers *polygonal region agreement*, *vector-boundary quality*, and *ring-structure correctness*.

Recent remote-sensing methods introduce domain-specific geometric representations for polygon extraction. Frame-field methods predict local orientation fields to guide polygonization and improve boundary regularity [10]. Graph-based methods represent vector structures as connected vertices and edges [54], and vertex- or representative-point-based methods detect corners, control points, or boundary keypoints before assembling polygons [13, 26, 45, 12, 16, 17]. Recently, LLM-based methods have also been explored for vector polygon generation [50], showing promising performance. These representations have improved geometric accuracy and regularity for remote-sensing objects, but they still provide limited support for complex polygons with interior rings, and often fail to explicitly model multi-ring topology.

2.2 Vector Polygon Generation Benchmarks and Evaluation

Existing geospatial polygon benchmarks are mainly built around buildings and crop fields. Representative building-footprint datasets include Inria [29], CrowdAI [32], WHU [14], and SpaceNet [42], while crop-field polygon datasets include AI4SmallFarms [37], Fields of The World (FTW) [19], and AI4Boundaries [8]. Other category-specific datasets have also been introduced for water bodies, such as GLH-Water [20], and roads, such as VHR-Road [43]. More recently, Deventer-512 [17] extends polygon vectorization to multiple land-cover categories, including buildings, roads, vegetation, water, and unvegetated areas.

Despite these efforts, existing benchmarks provide limited support for evaluating complex polygon generation. Many datasets focus on category-level region extraction rather than multi-ring vector geometry topology, and common metrics such as IoU, AP, BioU, POLIS, and MTA [23, 7, 4, 10, 33] mainly measure region overlap or aggregated boundary quality. Since they usually do not explicitly distinguish exterior rings from interior rings, methods can achieve strong benchmark scores while still wrongly representing complex polygons.

3 PolyTopoBench

3.1 Task and Polygon Representation

PolyTopoBench aims to systematically evaluate diverse vector polygon generation models from remote sensing imagery. Unlike most existing polygon generation methods, which focus on simple polygon generation, we systematically evaluate 11 polygon generation models on both simple polygon generation and complex polygon generation (i.e., polygons with one or multiple holes). A unified evaluator is developed to evaluate and compare diverse methods, where novel evaluation metrics have been proposed to evaluate polygon generations from three distinct perspectives: region agreement, vector-boundary quality, and ring-structure correctness. Figure 1 summarizes the benchmark design. Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, each method is converted to a set of scored polygon instances

$$\hat{\mathcal{P}}(\mathbf{I}) = \{(\hat{c}_i, \hat{s}_i, \hat{G}_i)\}_{i=1}^N, \quad (1)$$

Table 1: Dataset complexity statistics for the four single-class PolyTopoBench tasks. *Instances* denotes the number of total polygon annotations in each dataset/class. *Simple* and *Complex* denote instances with $H(G) = 0$ and $H(G) \geq 1$. *Holes* counts all interior rings in complex instances. *Holes / complex* is the average number of holes per complex instance. *Avg. verts* is the average number of vertices per instance, including exterior and interior rings and excluding the repeated closing point.

Dataset	Class	Instances	Simple	Complex	Holes	Holes / complex	Avg. verts
Inria	Building	254,133	248,621	5,512	19,854	3.60	10.27
Deventer	Road	4,533	3,467	1,066	9,992	9.37	62.57
Deventer	Vegetation	23,517	22,658	859	1,477	1.72	11.98
Deventer	Unvegetated	16,552	13,610	2,942	10,749	3.65	21.53

Table 2: Comparing polygon generation baselines. *Ring mode* is explicit (denoted as *Exp.*) when holes are natively modeled and implicit (denoted as *Imp.*) otherwise.

Method	Type and Description	Ring
Seg: Segmentation-then-polygonization		
U-Net + Poly. [38, 55]	Predict a semantic mask and convert its contours into polygons with rule-based simplification.	Imp.
Mask R-CNN + Poly. [11]	Predict instance masks and convert each mask contour into a simplified polygon.	Imp.
FM: Foundation-model-assisted polygonization		
SAM2 + Poly. [33]	Generate masks with SAM2 prompts or priors and polygonize the resulting mask contours.	Imp.
Rep.: Learned representation-to-vector		
HiSup [45]	Learn masks, vertices, and attraction fields, then extract mask contours to vertices.	Imp.
ACPV-Net [17]	Learn semantic masks and vertex heatmaps, then reconstruct a shared-boundary planar graph.	Exp.
FFL [10]	Learn masks and frame fields, then optimize contours along the learned directions.	Exp.
GCP [49]	Refine mask contours with a transformer and simplify them with global collinearity.	Exp.
HoliTracer [43]	Segment large images, reform mask contours, and trace vertices with a polygon sequence model.	Imp.
Direct: Direct vector decoding		
Pix2Poly [2]	Decode polygon vertices as a transformer token sequence.	Imp.
PolyWorld [54]	Predict vertices and connect them with a graph model to form polygons.	Imp.
RoPoly [16]	Decode ordered vertices from each proposal region using vertex and logit queries.	Imp.

where $\hat{c}_i$ is the predicted category, $\hat{s}_i$ is a confidence score, and $\hat{G}_i$ is a single polygonal instance geometry with exactly one exterior ring and zero or more interior rings. The corresponding ground-truth set is

$$\mathcal{P}(\mathbf{I}) = \{(c_j, G_j)\}_{j=1}^M. \quad (2)$$

Here, c_j and G_j denote the ground-truth category and polygonal geometry.

For a polygon instance G , we denote its exterior ring by $e(G)$ and its set of interior rings by $\mathcal{I}(G)$. Let

$$H(G) = |\mathcal{I}(G)|, \quad R(G) = 1 + H(G), \quad (3)$$

where $H(G)$ is the number of holes and $R(G)$ is the total number of rings. We call G *simple* when $H(G) = 0$ and *complex* when $H(G) \geq 1$. Thus, unlike raster segmentation benchmarks that mainly evaluate the occupied image region, PolyTopoBench evaluates the full vector polygon structure. Both exterior and interior rings are part of the target geometry, so a prediction with a correct outer boundary but missing holes is still treated as an incomplete polygon.

3.2 Datasets and Benchmark Tasks

PolyTopoBench uses two aerial-image datasets that have different geometric objects with diverse classes. The Inria building dataset [29] evaluates building-footprint generation. Buildings are often compact and piecewise regular, but courtyards, inner voids, and clipped structures create multi-ring instances that cannot be represented by a simple polygon with one exterior boundary alone. Since the original Inria release provides binary masks, we align OpenStreetMap [35] vector annotations to the image tiles, correct them manually, and use the resulting polygons as vector polygon ground truth, which contains both simple and complex polygonal geometries. The Deventer [17] data provide multi-class land-cover vector polygons on 512×512 aerial patches. In this paper, we focus on the road, vegetation, and unvegetated classes because they contain richer complex-polygon structures.

Table 1 summarizes the statistics of hole-bearing polygons that appear in all four tasks. These statistics motivate a systematic evaluation that separates polygonal geometry exterior-boundary accuracy from interior-ring recovery.

3.3 Polygon Ring-Aware Evaluation Metrics

Prior polygon benchmarks treat a model prediction either as an occupied image region with evaluation metrics such as IoU and AP [23, 42] or as a single polygon boundary contour with evaluation metrics

such as BIoU, POLIS, and MTA [4, 26]. Both scores conflate errors on the exterior ring with errors on interior rings, and neither records whether the predicted polygon has the correct hole count or a valid topology. A method that returns only the exterior of a hole-bearing polygon can therefore obtain a near-perfect AP50. However, polygon holes sometimes carry an important meaning (e.g., farmland ownership), and failing to capture these holes can lead to significant issues in downstream applications [31, 48].

Our PolyTopoBench closes this gap by evaluating a polygon as a set of role-specific rings rather than as an image region or a single contour. We organize metrics into three complementary groups and highlight our novel contributions in each: **region agreement** metrics reuse standard AP50 for comparability; **vector-boundary quality** metrics keep BIoU, POLIS, and MTA but report them under three novel *ring-aware scopes* – Overall, Exterior, and Hole – that localize error to the exterior ring or the holes; and **ring-structure correctness** metrics include two novel topology-oriented metrics, *Hole-F1* and *Topo-EM*, built on a *role-specific ring matching* protocol. Full details are in Appendix A.1; below we summarize metrics and focus on novel metrics proposed in PolyTopoBench.

Region agreement metrics. For comparability with prior polygon-generation benchmarks, we report *AP50*. AP50 measures instance-level area overlap and is ring-agnostic by construction: a prediction that recovers the exterior boundary but misses every hole still receives a near-perfect score, which motivates the ring-localized metrics below.

Vector-boundary quality metrics. We report three standard contour metrics—*BIoU* [7], *POLIS* [4], and *MTA* [10]—which measure boundary-buffer overlap, vertex-to-boundary distance, and tangent-angle deviation, respectively (BIoU is higher-is-better, while POLIS and MTA are lower-is-better). Our contribution is to evaluate each of them under three *ring-aware scopes* that restrict which rings of a matched instance pair enter the computation: *Overall* (O) uses all rings of the matched pair (exterior and interior), *Ext.* (E) uses only the paired exterior rings, and *Hole* (H) uses interior-ring pairs produced by role-specific matching (below). The three scopes split each boundary score into contributions from the outer boundary versus recovered holes, in which a single polygon-level number silently merges.

Ring-structure correctness metrics. Ring-structure metrics evaluate whether interior holes and the complete polygon topology are recovered. They require a correspondence between predicted and ground-truth rings, which we obtain by *role-specific ring matching*: for an IoU-matched instance pair $(\hat{G}, G)$, the two exterior rings are paired directly, while interior rings in $\mathcal{I}(\hat{G})$ and $\mathcal{I}(G)$ are matched one-to-one via Hungarian assignment with per-ring BIoU as the cost, accepting a hole pair only when its BIoU exceeds a fixed threshold. This prevents a predicted hole from being counted against a ground-truth hole in a different part of the polygon. Please see Definition 12 in Appendix A.1 for detailed descriptions. Based on this, we introduce two metrics:

Definition 1 (Hole-F1). *Hole-F1* treats each interior ring as a detection object. Let TP_h be the number of accepted hole matches, FP_h the number of unmatched predicted holes, and FN_h the number of unmatched ground-truth holes. Hole-F1 is defined as

$$\text{Hole-F1} = \frac{2TP_h}{2TP_h + FP_h + FN_h}. \quad (4)$$

Unlike IoU or exterior BIoU, Hole-F1 collapses to zero whenever a method never emits interior rings, directly exposing the single-exterior-contour failure mode.

Definition 2 (Topo-EM). *Topo-EM* (topology exact match) is a stricter instance-level check on the entire ring structure. For a matched pair $(\hat{G}, G)$, we set $T(\hat{G}, G) = 1$ only if all the following requirements are met: 1) $\hat{G}$ is a valid polygon; 2) its exterior ring is matched to $e(G)$; 3) the predicted and ground-truth hole counts are equal; and 4) every ground-truth hole is matched one-to-one with a predicted hole. Otherwise, $T(\hat{G}, G) = 0$. Let $\mathcal{M}_{\text{inst}}$ be the matched instance pairs and $\mathcal{U}_{\text{gt}}, \mathcal{U}_{\text{pred}}$ the unmatched ground-truth and predicted instances. Then

$$\text{Topo-EM} = \frac{1}{K} \sum_{(\hat{G}, G) \in \mathcal{M}_{\text{inst}}} T(\hat{G}, G), \quad K = |\mathcal{M}_{\text{inst}}| + |\mathcal{U}_{\text{gt}}| + |\mathcal{U}_{\text{pred}}|. \quad (5)$$

Topo-EM simultaneously punishes missing holes, hallucinated holes, invalid geometry, and mismatched exterior rings. It is therefore used as a strict topology exactness diagnostic rather than as the only leaderboard criterion. In our default evaluation, BIoU uses a one-pixel boundary buffer and ring matches are accepted at a threshold 0.1. Appendix A.2 studies the sensitivity of Hole-F1 and Topo-EM to stricter ring matching and hole-size filtering.

3.4 Baselines and Evaluation Protocol

As summarized in Table 2, PolyTopoBench evaluates eleven baselines and groups them into four types: (1) *Seg.* methods are segmentation-then-polygonization pipelines, including U-Net [38] and Mask R-CNN [11], followed by a deterministic raster-to-vector polygonization [55]. (2) *FM* methods are vision-foundation-model-assisted polygonization pipelines. We prompt SAM2 with Mask R-CNN bounding boxes and polygonize its masks [51, 33]. We also include specialized vector generation models, which are split into *Rep.* and *Direct.* (3) *Rep.* methods include HiSup [45], ACPV-Net [17], Frame Field Learning (FFL) [10], GCP [49], and HoliTracer [43]. Each method first predicts an intermediate representation and then decodes it into vector polygons using a representation-specific step (vertex attraction in HiSup, planar-graph assembly in ACPV-Net, frame-field optimization in FFL, transformer contour refinement in GCP, or polygon-sequence tracing in HoliTracer). (4) *Direct* methods include Pix2Poly [2], PolyWorld [54], and RoIPoly [16]. These methods decode vector primitives or polygon connectivity more directly at the instance level.

Among all baselines, we further distinguish whether interior rings are modeled implicitly or explicitly. *Implicit* hole-aware methods do not natively predict holes as separate polygon rings. Instead, holes are recovered indirectly from masks [45], contours [43], ring containment [2, 54], or post-processing [55]. *Explicit* hole-aware methods natively model interior rings or shared boundary structures, so holes can be represented as part of the predicted vector topology. In our taxonomy, ACPV-Net, FFL, and GCP are *explicit* hole-aware methods, while the others are *implicit*.

These baselines are intentionally heterogeneous. To compare them fairly, PolyTopoBench normalizes every prediction to the same instance-level geometry before evaluation, i.e., a scored polygon with one exterior ring and zero or more interior rings. Native confidence scores are used when available. Score-free methods are assigned a constant score and are primarily compared using non-ranked boundary, ring, and topology metrics.

For *Seg.* methods, we convert raster masks to polygons using contour extraction and Douglas–Peucker simplification. For *FM* methods, SAM2 generates raster masks using Mask R-CNN bbox prior, followed by the same polygonization procedure as *Seg.*. We ablate SAM2 priors under different settings (see Section 4.4). In the *Direct* method group, for methods that output independent closed rings [2, 16], we infer polygon structure by ring containment: outer rings become exterior boundaries, while enclosed rings are assigned as interior rings. PolyTopoBench does not add holes, repair missing rings, or complete invalid topology before evaluation, so missing interior rings are penalized by Hole-F1 and Topo-EM. Appendix A.3 provides the detailed settings for each baseline.

4 Experiment

4.1 Main Benchmark Results

Table 3 reports the full PolyTopoBench leaderboard across the four single-class tasks. We discuss the Inria Building task and the three Deventer tasks in turn.

Results on the Inria Building dataset. The two *Seg.* methods obtain the highest region agreement, with Mask R-CNN + Poly. at $AP50 = 0.720$ and U-Net + Poly. at 0.670 , above every *Rep.* and *Direct* baseline. Among *Rep.* methods, GCP (0.631), ACPV-Net (0.575), and HiSup (0.496) are closest, while the three *Direct* methods collapse on $AP50$ (Pix2Poly 0.102 , PolyWorld 0.016 , RoIPoly 0.001). HoliTracer (*Rep.*), designed for holistic large-tile inference, transfers poorly to the 512×512 patches used in PolyTopoBench (0.034). A clearer ordering emerges in ring-structure metrics. Hole-F1 is led by ACPV-Net (0.527), U-Net + Poly. (0.514), and HiSup (0.473), while the remaining baselines drop sharply (Mask R-CNN + Poly. 0.167 , GCP 0.015 , FFL 0.000 , and all *Direct* methods below 0.05). The distribution does not cleanly split along *Exp./Imp.* lines. GCP and FFL are *Exp.* yet recover almost no holes, while the *Imp.* U-Net + Poly. and HiSup rank in the top three. Topo-EM flips the ranking. U-Net + Poly. is best (0.531), Mask R-CNN + Poly. follows (0.519), and GCP scores only 0.065 despite strong $AP50$ and E-BIoU (0.312), because it rarely recovers the complete ring structure. Together, these numbers show that region agreement and ring-structure correctness are largely uncorrelated on Inria dataset.

Results on the Deventer multi-class dataset. The Deventer road, vegetation, and unvegetated tasks are substantially harder. The best $AP50$ drops to 0.520 (U-Net + Poly., Road), 0.382 (U-Net + Poly., Vegetation), and 0.247 (U-Net + Poly., Unvegetated). The relative ordering across method types is similar, with *Seg.* leading on $AP50$, *Rep.* methods (HiSup, ACPV-Net) close behind on Road and Vegetation, and *Direct* methods again near zero. Deventer Road dataset has, on average, 9.37 holes

Table 3: Main results across the four single-class polygon generation tasks. *Building* uses the Inria dataset [29]; *Road*, *Veg.* (vegetation), and *Unveg.* (unvegetated) use the Deventer dataset [17]. O, E, and H denote the Overall, Ext., and Hole ring-aware scopes. AP50, BIoU, Hole-F1, and Topo-EM are higher-is-better; POLIS and MTA are lower-is-better. “–” marks scores that are undefined because no hole match is accepted on that task. Best value per task and per column is **bold**.

Method	Task	Region	Vector-boundary quality									Ring structure	
		AP50	BIoU			POLIS			MTA			Hole-F1	Topo-EM
		↑	O↑	E↑	H↑	O↓	E↓	H↓	O↓	E↓	H↓	↑	↑
U-Net + Poly.	Building	0.670	0.294	0.308	0.234	2.62	2.99	2.79	47.52	47.02	48.23	0.514	0.531
	Road	0.520	0.328	0.311	0.337	5.16	8.44	2.66	52.66	52.58	46.93	0.466	0.144
	Veg.	0.382	0.340	0.313	0.208	4.41	6.12	5.17	50.52	49.48	48.83	0.022	0.222
	Unveg.	0.247	0.222	0.206	0.253	5.39	8.24	3.15	56.70	55.43	50.68	0.428	0.088
Mask R-CNN + Poly.	Building	0.720	0.267	0.292	0.155	2.61	2.72	3.78	43.10	42.49	53.40	0.167	0.519
	Road	0.113	0.234	0.273	0.184	7.75	8.90	4.28	52.06	49.33	48.81	0.010	0.081
	Veg.	0.357	0.321	0.320	0.244	4.50	4.72	2.59	46.63	45.94	42.75	0.017	0.273
	Unveg.	0.181	0.220	0.234	0.186	5.23	6.21	3.26	53.70	51.55	53.29	0.254	0.099
SAM2 + Poly.	Building	0.431	0.242	0.269	0.154	3.05	3.17	3.61	46.43	45.63	48.85	0.017	0.377
	Road	0.023	0.299	0.301	0.177	6.31	10.40	4.55	52.55	52.30	52.36	0.014	0.012
	Veg.	0.079	0.309	0.298	0.243	6.06	6.43	2.73	50.19	49.54	50.46	0.003	0.099
	Unveg.	0.005	0.239	0.230	0.165	5.57	7.44	4.62	51.89	50.27	44.06	0.009	0.020
HiSup	Building	0.496	0.282	0.289	0.224	2.84	3.79	3.02	39.82	39.66	41.43	0.473	0.387
	Road	0.389	0.267	0.257	0.256	6.49	9.40	3.97	49.84	48.74	43.82	0.303	0.137
	Veg.	0.264	0.295	0.270	0.176	6.86	8.88	4.07	45.40	44.46	49.65	0.016	0.181
	Unveg.	0.110	0.181	0.180	0.207	7.66	12.31	3.85	54.24	51.16	46.72	0.350	0.044
ACPV-Net	Building	0.575	0.259	0.272	0.234	2.76	3.33	2.64	45.76	45.37	45.84	0.527	0.440
	Road	0.361	0.302	0.281	0.283	6.35	9.70	2.52	49.00	48.89	45.98	0.394	0.137
	Veg.	0.274	0.332	0.306	0.153	6.24	8.27	16.81	47.03	45.70	38.23	0.006	0.154
	Unveg.	0.082	0.157	0.169	0.173	6.94	10.90	4.07	55.84	53.31	50.77	0.265	0.026
FFL	Building	0.400	0.238	0.255	–	3.72	4.70	–	45.44	44.74	–	0.000	0.222
	Road	0.163	0.225	0.216	–	10.06	15.14	–	56.23	56.04	–	0.000	0.027
	Veg.	0.254	0.303	0.271	–	7.00	9.18	–	48.89	48.24	–	0.000	0.132
	Unveg.	0.011	0.092	0.166	–	21.09	10.03	–	60.33	53.41	–	0.000	0.002
GCP	Building	0.631	0.288	0.312	0.152	2.80	3.28	5.61	34.94	34.83	44.16	0.015	0.065
	Road	0.281	0.246	0.254	0.203	7.95	12.18	4.94	49.87	47.63	45.64	0.015	0.004
	Veg.	0.203	0.313	0.299	0.184	5.72	7.78	3.75	47.31	46.28	45.76	0.001	0.024
	Unveg.	0.068	0.134	0.179	0.151	10.64	11.78	5.56	54.85	52.66	47.60	0.009	0.008
HoliTracer	Building	0.034	0.146	0.178	0.148	5.36	6.73	4.08	46.83	45.26	44.41	0.008	0.056
	Road	0.008	0.102	0.160	0.168	12.84	10.37	4.11	56.63	48.73	49.75	0.024	0.010
	Veg.	0.097	0.256	0.248	0.218	7.66	10.31	2.49	48.21	46.61	37.42	0.009	0.104
	Unveg.	0.011	0.094	0.148	0.142	11.87	10.70	4.23	57.71	53.47	47.36	0.065	0.007
Pix2Poly	Building	0.102	0.125	0.183	0.158	5.90	7.76	4.75	33.04	32.56	36.71	0.047	0.084
	Road	0.001	0.232	0.205	–	7.84	15.61	–	41.68	41.30	–	0.000	0.003
	Veg.	0.028	0.414	0.345	–	41.76	63.63	–	38.11	37.12	–	0.000	0.023
	Unveg.	0.000	0.171	0.156	–	28.60	39.12	–	59.53	52.22	–	0.000	0.000
PolyWorld	Building	0.016	0.151	0.180	0.129	3.96	7.79	5.16	41.39	40.91	–	0.002	0.021
	Road	0.000	–	0.106	–	–	7.81	–	–	–	–	0.000	0.000
	Veg.	0.004	0.113	0.146	–	53.56	72.95	–	54.06	53.34	–	0.000	0.004
	Unveg.	0.000	0.114	0.142	–	57.77	94.82	–	57.38	52.57	–	0.000	0.000
RoIPoly	Building	0.001	0.062	0.135	0.121	9.83	10.84	7.18	52.59	47.99	44.80	0.002	0.001
	Road	0.002	0.186	0.221	0.261	9.27	6.81	2.40	47.27	43.37	49.58	0.192	0.009
	Veg.	0.032	0.196	0.217	0.210	8.67	6.46	2.22	50.12	46.02	58.45	0.009	0.060
	Unveg.	0.005	0.104	0.175	0.230	10.69	6.71	3.57	55.25	49.37	45.46	0.127	0.008

per complex instance (Table 1), the highest hole density in the benchmark, and therefore provides the most informative signal on hole recovery. U-Net + Poly. reaches Hole-F1 = 0.466, ACPV-Net 0.394, and HiSup 0.303, while the rest of the baselines stay below 0.03. Topo-EM on Road remains below 0.15 for every method, indicating that recovering some holes is very different from recovering the complete ring structure. On Vegetation, Pix2Poly attains the highest O-BIoU (0.414) and E-BIoU (0.345) of any method, but POLIS = 41.76 and Topo-EM = 0.023 reveal that its matched polygons are crisp in the boundary buffer yet poorly localized at the vertex level. This error pattern is hidden by exterior-only evaluation. Hole metrics marked “–” for FFL, Pix2Poly, and PolyWorld indicate that, although their architectures can represent interior rings and we trained each on the same splits, the trained models rarely produce an accepted hole match.

Overall, three findings can be concluded. First, *Seg.* pipelines lead the PolyTopoBench because their region-first training signal transfers robustly across complexity regimes, not because their vector outputs are higher quality. Second, *Direct* decoders that excel on simple-building benchmarks

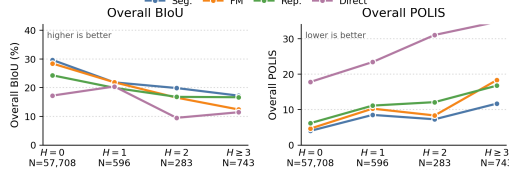


Figure 2: Simple-to-complex generalization results. Ground-truth instances are grouped by hole count H with sample size N . Each curve reports the average of one method type (*Seg.*, *FM*, *Rep.*, *Direct*) over its constituent baselines.

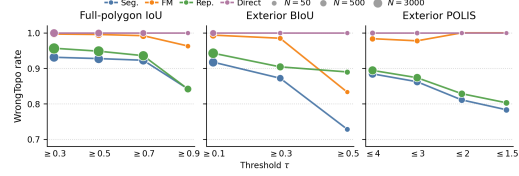


Figure 3: WrongTopo rate under different geometry filters on complex polygons. From left to right, we filter matched pairs by full-polygon IoU, exterior BioU, and exterior POLIS; marker size is the number of ground-truth-prediction pairs (N).

collapse once holes and long, irregular contours dominate the dataset. Third, Topo-EM does not saturate even when AP50 and exterior-boundary scores are strong, so ring-structure correctness is a distinct aspect that current methods do not address.

4.2 Hole-Aware Evaluation

Simple-to-complex generalization. We group ground-truth polygon annotations by hole count ($H = 0$ to $H \geq 3$, where H is a short name for $H(G)$) and report the average Overall BioU and Overall POLIS of each method type in Figure 2. Overall BioU falls, and Overall POLIS grows monotonically with H for every method type. *Seg.* drops from Overall BioU 0.296 at $H = 0$ to 0.172 at $H \geq 3$, with Overall POLIS rising from 4.0 to 11.7. *FM* degrades more sharply, from 0.284 to 0.124 on BioU and from 4.6 to 18.3 on POLIS. The same trend is observed in specialized vector polygon generation methods (both *Rep.* and *Direct.*). In a nutshell, when the complexity increases, the model performance drops significantly.

Good exterior geometry still hides topology errors. Next, we ask whether a prediction has the correct topology given that its exterior geometry is already good. To make this question quantitative, we define WrongTopo. Let $\mathcal{M}_{\text{complex}}$ be the set of IoU-matched pairs $(\hat{G}, G)$ whose ground-truth polygon has at least one hole, and let $T(\hat{G}, G) \in \{0, 1\}$ be the per-pair topology indicator defined for Topo-EM in Section 3.3. Let q denote a geometry filter on matched pairs, chosen from $\{\text{IoU} \geq \tau, E\text{-BioU} \geq \tau, E\text{-POLIS} \leq \tau\}$, where the prefix E restricts the score to the exterior ring. $\mathcal{S}_{q,\tau} \subseteq \mathcal{M}_{\text{complex}}$ is the subset of pairs that satisfy q at threshold τ , WrongTopo averages $1 - T$ over this subset:

$$\text{WrongTopo}(q, \tau) = \frac{1}{|\mathcal{S}_{q,\tau}|} \sum_{(\hat{G}, G) \in \mathcal{S}_{q,\tau}} (1 - T(\hat{G}, G)). \quad (6)$$

IoU filters on full-polygon overlap, E-BioU, on exterior-boundary overlap, and E-POLIS on exterior-boundary distance. A lower WrongTopo score indicates a better topology preservation ability.

Figure 3 shows that no geometry filter removes topology errors. Even at $\text{IoU} \geq 0.9$, WrongTopo stays between 0.84 (*Seg.* and *Rep.*) and 1.00 (*Direct*), with *FM* at 0.96. At $E\text{-BioU} \geq 0.3$, WrongTopo is 0.87 for *Seg.*, 0.98 for *FM*, 0.90 for *Rep.*, and 1.00 for *Direct*. At $E\text{-POLIS} \leq 2$, the rates are 0.81, 1.00, 0.83, and 1.00, respectively. Good exterior geometry therefore does not tell us whether the interior rings are correct.

4.3 Qualitative Results

Figure 4 shows that visual quality is often limited by ring topology rather than exterior geometry. Across all four categories, the predicted exterior rings usually follow the target object, but the interior rings are wrongly represented. The building and road examples contain many ground-truth holes, yet different methods recover very different hole counts; the vegetation and unvegetated examples further show that even plausible outer boundaries can have wrong or incomplete holes. These patterns match the quantitative results: *Seg.* and *FM* pipelines often turn mask fragments into false holes, while *Rep.* and *Direct* methods produce smoother contours but still fail to recover reliable interior-ring structure.

4.4 Why Do Baselines Fail?

We further conduct baseline-setting ablations in Appendix A.5. The results show that the eleven methods fail for different reasons, and none of the tested settings produces reliable interior rings. *Seg.* and *FM* pipelines rely on binary foreground masks without explicit interior-ring supervision, so holes lost in the mask cannot be recovered by polygonization; for example, SAM2 with ground-truth boxes reaches $\text{AP50} = 0.824$ but Hole-F1 remains 0.000. *Rep.* methods improve exterior geometry

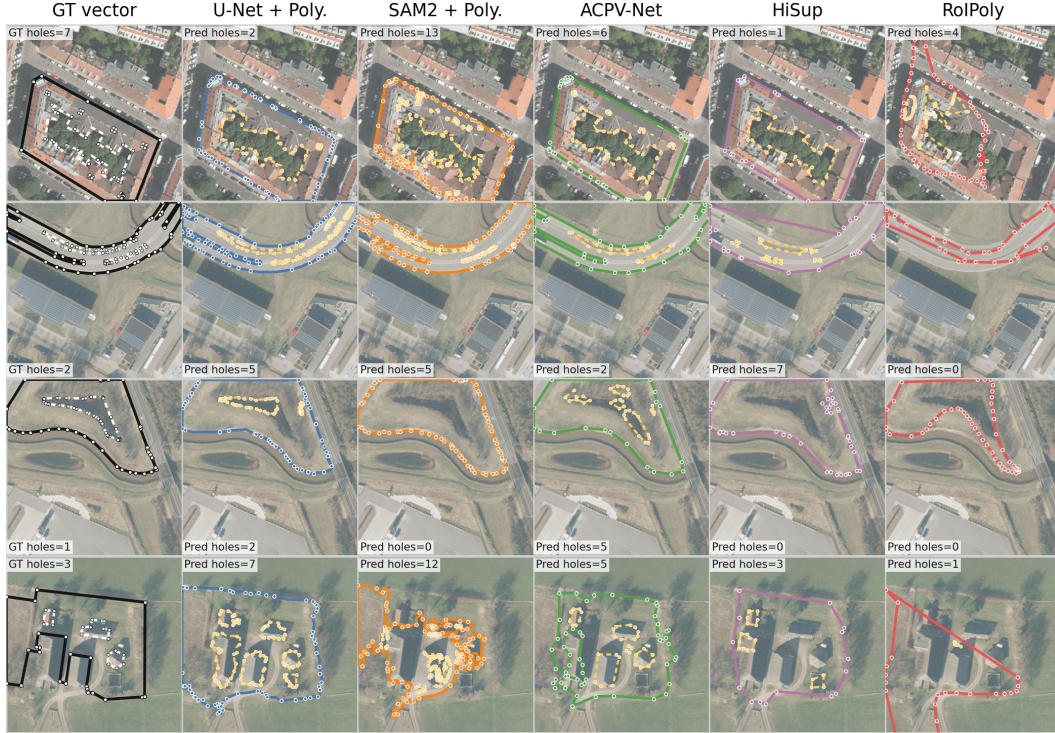


Figure 4: Qualitative vector prediction results. Rows from top to bottom show building, road, vegetation, and unvegetated examples. Columns show five methods, one per type, which are U-Net + Poly. (*Seg.*), SAM2 + Poly. (*FM*), ACPV-Net [17] (*Rep.*, *Exp.*), HiSup [45] (*Rep.*, *Imp.*), and RoIPoly [16] (*Direct*, *Imp.*). Each patch is annotated with its ground-truth or predicted hole count. Ground-truth exterior rings are black, interior is white, predicted exterior rings use per-method colors, holes are dashed yellow. Vertices are marked with small circles. See Appendix A.4 for more.

but rarely preserve topology: GCP obtains the highest E-BIoU = 0.312 on Inria, while Topo-EM stays at 0.065. *Direct* methods are limited by upstream proposal or vertex modules, which prevent the decoder from forming valid ring structures; RoIPoly’s Hole-F1 drops from 0.310 to 0.002 when ground-truth boxes are replaced by Sparse R-CNN proposals [40], and Pix2Poly improves from AP50 = 0.000 to 0.102 when the maximum vertex length increases from 192 to 224, but Hole-F1 only reaches 0.047. HoliTracer and PolyWorld also remain weak on interior-ring recovery, with accepted hole matches appearing only rarely and not translating into reliable topology exactness. Overall, no configuration achieves Topo-EM above 0.377, and most remain below 0.1, indicating that existing polygon generators are still optimized mainly for regions or exterior boundaries. Future models should directly supervise interior rings and their topological roles, enforce polygon validity and ring containment during decoding, and jointly train polygon decoders with their proposal modules rather than relying on separately trained upstream components.

5 Limitation and Broader Impact

PolyTopoBench is limited to four single-class tasks from two aerial-image datasets, and its annotations may inherit OSM incompleteness, raster-label errors, and upstream Deventer label noise. We only include baselines with public code; therefore LLM-based methods such as VectorLLM [50] are not covered because their code was unavailable at submission time. The benchmark supports multiple geospatial applications such as urban planning and disaster response, but could also be misused for privacy-sensitive mapping and surveillance.

6 Conclusion

We presented PolyTopoBench, a benchmark and evaluation framework for complex vector polygon generation from remote-sensing imagery. PolyTopoBench provides four single-class tasks, dense complex-polygon annotations, 11 public baselines, and ring-aware metrics for evaluating exterior shape and interior topology. Results show that strong region or exterior-boundary scores do not guarantee correct interior rings, and no existing model reliably preserves topology.

References

- [1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018.
- [2] Yeshwanth Kumar Adimoolam, Charalambos Poullis, and Melinos Averkiou. Pix2poly: A sequence prediction method for end-to-end polygonal building footprint extraction from remote sensing imagery. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8484–8493. IEEE, 2025.
- [3] Zurui Ao, Xiaomei Hu, Shengli Tao, Xie Hu, Guoquan Wang, Mingjia Li, Fang Wang, Litang Hu, Xiuyu Liang, Jingfeng Xiao, et al. A national-scale assessment of land subsidence in china’s major cities. *Science*, 384(6693):301–306, 2024.
- [4] Janja Avbelj, Rupert Müller, and Richard Bamler. A metric for polygon comparison and building extraction evaluation. *IEEE Geoscience and Remote Sensing Letters*, 12(1):170–174, 2014.
- [5] Riyang Bao, Cheng Yang, Dazhou Yu, Zhexiang Tang, Gengchen Mai, and Liang Zhao. Spatial-agent: Agentic geo-spatial reasoning with scientific core concepts. *arXiv preprint arXiv:2601.16965*, 2026.
- [6] Lluís Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5230–5238, 2017.
- [7] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15334–15342, 2021.
- [8] Raphaël d’Andrimont, Martin Claverie, Pieter Kempeneers, Davide Muraro, Momchil Yordanov, Devis Peressutti, Matej Batič, and François Waldner. Ai4boundaries: an open ai-ready dataset to map field boundaries with sentinel-2 and aerial photography. *Earth System Science Data*, 15(1):317–329, 2023.
- [9] Vivien Sainte Fare Garnot and Loic Landrieu. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4872–4881, 2021.
- [10] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. Polygonal building extraction by frame field learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5891–5900, 2021.
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [12] Yuan Hu, Zhibin Wang, Zhou Huang, and Yu Liu. Polybuilding: Polygon transformer for building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 199:15–27, 2023.
- [13] Wei Huang, Hong Tang, and Penglei Xu. Oec-rnn: Object-oriented delineation of rooftops with edges and corners using the recurrent neural network from the aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2021.
- [14] Shunping Ji, Shiqing Wei, and Meng Lu. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Transactions on geoscience and remote sensing*, 57(1):574–586, 2018.
- [15] Yuhan Ji, Song Gao, Ying Nie, Ivan Majić, and Krzysztof Janowicz. Foundation models for geospatial reasoning: assessing the capabilities of large language models in understanding geometries and topological spatial relations. *International Journal of Geographical Information Science*, 39(9):1866–1903, 2025.
- [16] Weiqin Jiao, Hao Cheng, George Vosselman, and Claudio Persello. Roipoly: Vectorized building outline extraction using vertex and logit embeddings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 224:317–328, 2025.

- [17] Weiqin Jiao, Hao Cheng, George Vosselman, and Claudio Persello. Acpv-net: All-class polygonal vectorization for seamless vector map generation from aerial imagery. *arXiv preprint arXiv:2603.16616*, 2026.
- [18] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- [19] Hannah Kerner, Snehal Chaudhari, Aninda Ghosh, Caleb Robinson, Adeel Ahmad, Eddie Choi, Nathan Jacobs, Chris Holmes, Matthias Mohr, Rahul Dodhia, et al. Fields of the world: A machine learning benchmark dataset for global agricultural field boundary segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28151–28159, 2025.
- [20] Yansheng Li, Bo Dang, Wanchun Li, and Yongjun Zhang. Glh-water: A large-scale dataset for global surface water detection in large-size very-high-resolution satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22213–22221, 2024.
- [21] Zhuohong Li, Linxin Li, Ting Hu, Mofan Cheng, Wei He, Tong Qiu, Liangpei Zhang, and Hongyan Zhang. Satellite mapping of every building’s function in urban china reveals deep built environment disparities. *Nature Communications*, 2026.
- [22] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9131–9140, 2020.
- [23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [24] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5257–5266, 2019.
- [25] Zeping Liu, Hong Tang, Lin Feng, and Siqing Lyu. China building rooftop area: the first multi-annual (2016–2021) and high-resolution (2.5 m) building rooftop area dataset in china derived with super-resolution segmentation from sentinel-2 imagery. *Earth System Science Data*, 15(8):3547–3572, 2023.
- [26] Zeping Liu, Hong Tang, and Wei Huang. Building outline delineation from vhr remote sensing images using the convolutional recurrent neural network embedded with line segment information. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.
- [27] Zichen Liu, Jun Hao Liew, Xiangyu Chen, and Jiashi Feng. Dance: A deep attentive contour model for efficient instance segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 345–354, 2021.
- [28] Muying Luo, Shunping Ji, and Shiqing Wei. A diverse large-scale building dataset and a novel plug-and-play domain generalization method for building extraction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:4122–4138, 2023.
- [29] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International geoscience and remote sensing symposium (IGARSS)*, pages 3226–3229. IEEE, 2017.
- [30] Gengchen Mai, Krzysztof Janowicz, Cheng He, Sumang Liu, and Ni Lao. Poireviewqa: A semantically enriched poi retrieval and question answering dataset. In *Proceedings of the 12th Workshop on Geographic Information Retrieval*, pages 1–2, 2018.
- [31] Gengchen Mai, Chiyu Jiang, Weiwei Sun, Rui Zhu, Yao Xuan, Ling Cai, Krzysztof Janowicz, Stefano Ermon, and Ni Lao. Towards general-purpose representation learning of polygonal geometries. *GeoInformatica*, 27(2):289–340, 2023.

- [32] Sharada Prasanna Mohanty, Jakub Czakon, Kamil A Kaczmarek, Andrzej Pyskir, Piotr Tarasiewicz, Saket Kunwar, Janick Rohrbach, Dave Luo, Manjunath Prasad, Sascha Fleer, et al. Deep learning for understanding satellite imagery: An experimental survey. *Frontiers in Artificial Intelligence*, 3, 2020.
- [33] Gedeon Muhawenayo, Caleb Robinson, Subash Khanal, Zhanpei Fang, Isaac Corley, Alexander Wollam, Tianyi Gao, Leonard Strnad, Ryan Avery, Lyndon Estes, et al. Prue: A practical recipe for field boundary segmentation at scale. *arXiv preprint arXiv:2603.27101*, 2026.
- [34] Husain Najafi, Pallav Kumar Shrestha, Oldrich Rakovec, Heiko Apel, Sergiy Vorogushyn, Rohini Kumar, Stephan Thober, Bruno Merz, and Luis Samaniego. High-resolution impact-based early warning system for riverine flooding. *Nature communications*, 15(1):3726, 2024.
- [35] OpenStreetMap contributors. Planet dump retrieved from <https://planet.osm.org> . <https://www.openstreetmap.org>, 2017.
- [36] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8533–8542, 2020.
- [37] Claudio Persello, Jeroen Grift, Xinyan Fan, Claudia Paris, Ronny Hänsch, Mila Koeva, and Andrew Nelson. Ai4smallfarms: A dataset for crop field delineation in southeast asian smallholder farms. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5, 2023.
- [38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [39] Raphael Sulzer, Liuyun Duan, Nicolas Girard, and Florent Lafarge. The p³ dataset: Pixels, points and polygons for multimodal building vectorization. *arXiv preprint arXiv:2505.15379*, 2025.
- [40] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021.
- [41] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.
- [42] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018.
- [43] Yu Wang, Bo Dang, Wanchun Li, Wei Chen, and Yansheng Li. Holitracer: Holistic vectorization of geographic objects from large-size remote sensing imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8482–8491, 2025.
- [44] Yiqun Xie, Jiannan Cai, Rahul Bhojwani, Shashi Shekhar, and Joseph Knight. A locally-constrained yolo framework for detecting small and densely-distributed building footprints. *International Journal of Geographical Information Science*, 34(4):777–801, 2020.
- [45] Bowen Xu, Jiakun Xu, Nan Xue, and Gui-Song Xia. Hisup: Accurate polygonal mapping of buildings in satellite imagery with hierarchical supervision. *ISPRS Journal of Photogrammetry and Remote Sensing*, 198:284–296, 2023.
- [46] Susu Xu, Joshua Dimasaka, David J Wald, and Hae Young Noh. Seismic multi-hazard and impact estimation via causal inference from satellite imagery. *Nature Communications*, 13(1):7793, 2022.
- [47] Xiongfeng Yan, Tinghua Ai, Min Yang, and Hongmei Yin. A graph convolutional neural network for classification of building patterns using spatial vector data. *ISPRS journal of photogrammetry and remote sensing*, 150:259–273, 2019.

- [48] Dazhou Yu, Riyang Bao, Ruiyu Ning, Jinghong Peng, Gengchen Mai, and Liang Zhao. Spatial-rag: Spatial retrieval augmented generation for real-world geospatial reasoning questions. *arXiv preprint arXiv:2502.18470*, 2025.
- [49] Fahong Zhang, Yilei Shi, and Xiao Xiang Zhu. Global collinearity-aware polygonizer for polygonal building mapping in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2025.
- [50] Tao Zhang, Shiqing Wei, Shihao Chen, Wenling Yu, Muying Luo, and Shunping Ji. Vectorllm: Human-like extraction of structured building contours via multimodal llms. *ISPRS Journal of Photogrammetry and Remote Sensing*, 233:55–68, 2026.
- [51] Kang Zhao, Jungwon Kang, Jaewook Jung, and Gunho Sohn. Building extraction from satellite images using mask r-cnn with building boundary regularization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 247–251, 2018.
- [52] Zhuo Zheng, Yanfei Zhong, Junjue Wang, and Ailong Ma. Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4096–4105, 2020.
- [53] Zhe Zhu, Shi Qiu, and Su Ye. Remote sensing of land change: A multifaceted perspective. *Remote Sensing of Environment*, 282:113266, 2022.
- [54] Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, and Friedrich Fraundorfer. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1848–1857, 2022.
- [55] Stefano Zorzi, Ksenia Bittner, and Friedrich Fraundorfer. Machine-learned regularization and polygonization of building segmentation masks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3098–3105. IEEE, 2021.

A Appendix

A.1 Evaluation Metric Details

This appendix provides the full definitions of the metrics used in PolyTopoBench, organized into the three groups: region agreement, vector-boundary quality, and ring-structure correctness. Each normalized polygon instance has one exterior ring and zero or more interior rings. For a polygon G , we write $e(G)$ for its exterior ring and $\mathcal{I}(G)$ for its set of interior rings.

Definition 3 (Instance Matching). The predicted polygons $\hat{\mathcal{P}}(\mathbf{I}) = \{(\hat{c}_i, \hat{s}_i, \hat{G}_i)\}_{i=1}^N$ are first matched to ground-truth instances $\mathcal{P}(\mathbf{I}) = \{(c_j, G_j)\}_{j=1}^M$ using the full polygon IoU. Let $\Omega(G)$ denote the occupied region of polygon G , with interior rings subtracted from the exterior-ring region. The instance IoU is

$$\text{IoU}(\hat{G}, G) = \frac{|\Omega(\hat{G}) \cap \Omega(G)|}{|\Omega(\hat{G}) \cup \Omega(G)|}. \quad (7)$$

If $\text{IoU}(\hat{G}, G) \geq \beta$, then we can say the predicted polygon $\hat{G}$ *matches* the ground-truth polygon G , where β denotes the IoU matching threshold. Note that, the predicted polygons $\hat{\mathcal{P}}(\mathbf{I})$ are sorted by the model’s confidence scores $\{\hat{s}_i\}_{i=1}^N$, and each ground-truth instance (c_j, G_j) is matched at most once. We denote the resulting set of matched prediction–ground-truth instance pairs by $\mathcal{M}_{\text{inst}}$, and the unmatched ground-truth and prediction sets by $\mathcal{U}_{\text{gt}}$ and $\mathcal{U}_{\text{pred}}$.

A.1.1 Region agreement metrics

Definition 4 (AP50). AP50 is the average precision computed from full polygon IoU with matching threshold $\beta = 0.50$ (see Definition 3).

AP50 is reported once per benchmark task. Because AP50 depends only on occupied area, it does not distinguish between errors on the exterior ring, errors on interior rings, and topology failures.

A.1.2 Vector-boundary quality metrics

Definition 5 (Metric Scope). BIoU, POLIS, and MTA are reported under three ring-aware scopes that restrict which rings of a matched instance pair are used:

$$\mathcal{R}^{\text{Overall}}(G) = \{e(G)\} \cup \mathcal{I}(G), \quad \mathcal{R}^{\text{Ext}}(G) = \{e(G)\}, \quad \mathcal{R}^{\text{Hole}}(G) = \mathcal{I}(G). \quad (8)$$

The *Overall* metric uses all rings of a matched pair, the *Ext.* metric uses the paired exterior rings only, and the *Hole* metric uses interior-ring pairs produced by the role-specific ring matching defined in the next group. Hole-scope metrics therefore measure the *shape quality* of recovered holes.

Missed and hallucinated holes are counted separately by Hole-F1 and Topo-EM, which will be described later.

Definition 6 (Boundary IoU (BIoU)). Given a set of rings $\mathcal{R}$, let $B_r(\mathcal{R})$ be the union of r -pixel boundary buffers around all rings in $\mathcal{R}$. Boundary IoU is

$$\text{BIoU}_r(\mathcal{R}_1, \mathcal{R}_2) = \frac{|B_r(\mathcal{R}_1) \cap B_r(\mathcal{R}_2)|}{|B_r(\mathcal{R}_1) \cup B_r(\mathcal{R}_2)|}. \quad (9)$$

A higher BIoU indicates a better model performance.

Definition 7 (Overall-BIoU, Ext-BIoU, and Hole-BIoU). Overall-BIoU and Ext-BIoU apply BIoU_r to $\mathcal{R}^{\text{Overall}}$ and $\mathcal{R}^{\text{Ext}}$ of each matched instance pair. Hole-BIoU is the mean over accepted hole matches:

$$\text{Hole-BIoU} = \frac{1}{|\mathcal{M}_{\text{hole}}|} \sum_{(\hat{h}, h) \in \mathcal{M}_{\text{hole}}} \text{BIoU}_r(\{\hat{h}\}, \{h\}), \quad (10)$$

where $\mathcal{M}_{\text{hole}}$ is defined in Definition 13 below. If a method produces no accepted hole matches on a task, Hole-BIoU is undefined and reported as “–”. In that case, Hole-F1 (Definition 1) reflects the failure to recover holes.

Definition 8 (POLIS). POLIS measures the average bidirectional distance from polygon vertices to the opposite boundary. For two ring sets $\mathcal{R}_1, \mathcal{R}_2$ with vertex sets V_1, V_2 and boundaries $\partial\mathcal{R}_1, \partial\mathcal{R}_2$,

$$\text{POLIS}(\mathcal{R}_1, \mathcal{R}_2) = \frac{1}{2} \left(\frac{1}{|V_1|} \sum_{v \in V_1} d(v, \partial\mathcal{R}_2) + \frac{1}{|V_2|} \sum_{v \in V_2} d(v, \partial\mathcal{R}_1) \right). \quad (11)$$

Here, $\mathcal{R}_1, \mathcal{R}_2$ can denote either exterior polygon rings or interior hole rings. A lower POLIS indicates a better model performance.

Definition 9 (Overall-POLIS, Ext-POLIS, and Hole-POLIS). Overall-POLIS and Ext-POLIS apply POLIS to $\mathcal{R}^{\text{Overall}}$ and $\mathcal{R}^{\text{Ext}}$. Hole-POLIS averages per-pair POLIS over $\mathcal{M}_{\text{hole}}$:

$$\text{Hole-POLIS} = \frac{1}{|\mathcal{M}_{\text{hole}}|} \sum_{(\hat{h}, h) \in \mathcal{M}_{\text{hole}}} \text{POLIS}(\{\hat{h}\}, \{h\}). \quad (12)$$

Hole-POLIS should be read together with Hole-F1 (Definition 1) because it characterizes the geometric quality of recovered holes, not the completeness of hole recovery.

Definition 10 (MTA). MTA measures the tangent-angle discrepancy between predicted and ground-truth polygon contours. We uniformly sample polygon exterior and interior contour segments from the predicted ring set and project each sample to the nearest ground-truth boundary segment. For a sampled predicted segment Δp_ℓ and its projected ground-truth segment Δq_ℓ , the orientation-invariant tangent-angle error is

$$\alpha_\ell = \arccos \left(\frac{|\langle \Delta p_\ell, \Delta q_\ell \rangle|}{\|\Delta p_\ell\|_2 \|\Delta q_\ell\|_2} \right). \quad (13)$$

Here, $\langle \cdot, \cdot \rangle$ denotes a dot product between two vectors and $\|\cdot\|_2$ denotes the vector L2 norm. Let $\mathcal{V}$ be the set of valid projected segments after the precision and stretch filters. Then MTA is computed as

$$\text{MTA} = \max_{\ell \in \mathcal{V}} \alpha_\ell. \quad (14)$$

A lower MTA indicates a better model performance.

Definition 11 (Overall-MTA, Ext-MTA, and Hole-MTA). Overall-MTA and Ext-MTA apply MTA to $\mathcal{R}^{\text{Overall}}$ and $\mathcal{R}^{\text{Ext}}$. Hole-MTA averages over $\mathcal{M}_{\text{hole}}$:

$$\text{Hole-MTA} = \frac{1}{|\mathcal{M}_{\text{hole}}|} \sum_{(\hat{h}, h) \in \mathcal{M}_{\text{hole}}} \text{MTA}(\{\hat{h}\}, \{h\}). \quad (15)$$

Like Hole-BIoU and Hole-POLIS, Hole-MTA is a conditional shape-quality metric for recovered holes and should be interpreted together with Hole-F1.

A.1.3 Ring-structure correctness

Ring-scoped and topology metrics require a correspondence between predicted and ground-truth rings. Thus, we give the definition of role-specific ring matching below.

Definition 12 (Role-specific Ring Matching). For each IoU-matched instance pair $(\hat{G}, G) \in \mathcal{M}_{\text{inst}}$, the exterior rings $e(\hat{G})$ and $e(G)$ are paired directly because each normalized polygon has exactly one exterior ring, and the exterior pair is considered matched when $\text{BiOU}_r(\{e(\hat{G})\}, \{e(G)\})$ exceeds the exterior matching threshold γ . Interior rings $\mathcal{I}(\hat{G})$ and $\mathcal{I}(G)$ are matched one-to-one via Hungarian assignment using per-ring BiOU, and a predicted hole and a ground-truth hole form an *accepted hole match* only when their BiOU exceeds the hole matching threshold γ .

Definition 13 (Matched Hole Set $\mathcal{M}_{\text{hole}}$). Based on Definition 12, we denote the set of all accepted hole matches across the whole evaluation set by $\mathcal{M}_{\text{hole}}$.

Unless otherwise stated, the main benchmark uses $r = 1$ pixel and $\gamma = 0.1$.

Definition 14 (Hole-F1). Hole-F1 treats interior rings as detection objects. Let $\text{TP}_h = |\mathcal{M}_{\text{hole}}|$, let FP_h be the number of unmatched predicted holes (including holes from unmatched predicted instances), and let FN_h be the number of unmatched ground-truth holes (including holes from unmatched ground-truth instances). Then

$$\text{Hole-F1} = \frac{2\text{TP}_h}{2\text{TP}_h + \text{FP}_h + \text{FN}_h}. \quad (16)$$

A method that emits only exterior rings, therefore receives zero hole recall on hole-bearing instances.

Definition 15 (Valid Polygon Geometry). For topology evaluation, we say that a predicted polygon $\hat{G}$ is *valid* if it satisfies the following five requirements:

1. Its exterior ring $e(\hat{G})$ is simple;

2. Each interior ring $\hat{h} \in \mathcal{I}(\hat{G})$ is simple;
3. All interior rings $\mathcal{I}(\hat{G})$ lie inside the exterior ring $e(\hat{G})$;
4. Interior rings do not overlap or cross one another;
5. The occupied region is non-empty.

Invalid polygons are counted as topology failures.

Definition 16 (Topo-EM). Topo-EM is computed over all IoU-matched instance pairs, with unmatched ground-truth instances and unmatched predictions counted as failures. A matched pair receives $T(\hat{G}, G) = 1$ only when all of the following hold:

1. $\hat{G}$ is a valid polygon according to Definition 15;
2. $e(\hat{G})$ is matched to $e(G)$;
3. The predicted and ground-truth hole counts are equal, i.e., $H(\hat{G}) = H(G)$;
4. Every ground-truth hole is matched one-to-one with a predicted hole.

Otherwise $T(\hat{G}, G) = 0$. We compute the Topo-EM as follows:

$$\text{Topo-EM} = \frac{1}{K} \sum_{(\hat{G}, G) \in \mathcal{M}_{\text{inst}}} T(\hat{G}, G), \quad K = |\mathcal{M}_{\text{inst}}| + |\mathcal{U}_{\text{gt}}| + |\mathcal{U}_{\text{pred}}|. \quad (17)$$

Topo-EM is stricter than AP50 and the exterior-scope boundary metrics – it penalizes missing holes, hallucinated holes, invalid polygons, and incorrect complete ring structures that can be hidden by region-level overlap.

A.1.4 How the Metrics Separate Error Types

The three metric groups respond to different error types, so the severity of an error is reflected by which metrics it affects. *Instance-level errors*, such as merging neighboring buildings into one polygon, break instance matching (Definition 3) and therefore reduce AP50 and all downstream metrics. *Ring-level errors*, i.e., missing or extra interior rings, preserve the instance match but reduce Hole-F1 and Topo-EM. *Geometric errors* in recovered rings are measured separately by BIoU, POLIS, and MTA under the Overall, Ext., and Hole scopes. Reading the three groups together therefore distinguishes a severe instance error from a topology error on an otherwise correct instance, and both from a small boundary deviation.

A.2 Sensitivity Analysis on Evaluation Metrics

In this section, we analyze whether the proposed metrics are sensitive to two evaluator choices, i.e., the minimum hole area $A_{\min}$ used to filter small interior rings, and the ring matching threshold γ used to determine whether two rings are matched. The goal is to verify that our main conclusion is not driven by a single threshold setting.

A.2.1 Metric Sensitivity to Minimum Hole Area $A_{\min}$

We first evaluate whether the topology metrics are dominated by many tiny holes. Figure 5 sweeps the minimum hole area $A_{\min}$ from 0 to 64 pixels (denoted by "px") on the Inria Building dataset, since the Inria dataset has the most complex buildings with diverse polygon complexity. At each threshold, holes smaller than the threshold $A_{\min}$ are removed from both ground truth and predictions before evaluation. We evaluate all 11 models on the Inria building dataset under different thresholds $A_{\min}$ by using four metrics – such as H -F1, H -BoI, Topo-Em, and AP50.

The main trend is stable across different thresholds. Filtering small holes increases Hole-F1 for the baseline, especially U-Net + Poly., ACPV-Net, and HiSup, because the remaining holes are larger and easier to match. However, the relative performance ordering does not change. Methods that fail to recover holes at the default threshold still remain weak after small holes are removed. Topo-EM is less affected than Hole-F1. This is expected because Topo-EM requires the whole polygon topology to be correct, including exterior matching, the exact number of holes, and one-to-one hole matching. Removing small holes reduces some difficult cases, but it does not fix missing holes, extra holes, or invalid ring structures. AP50 is almost unchanged, which again shows that region overlap is insensitive to interior-ring topology.

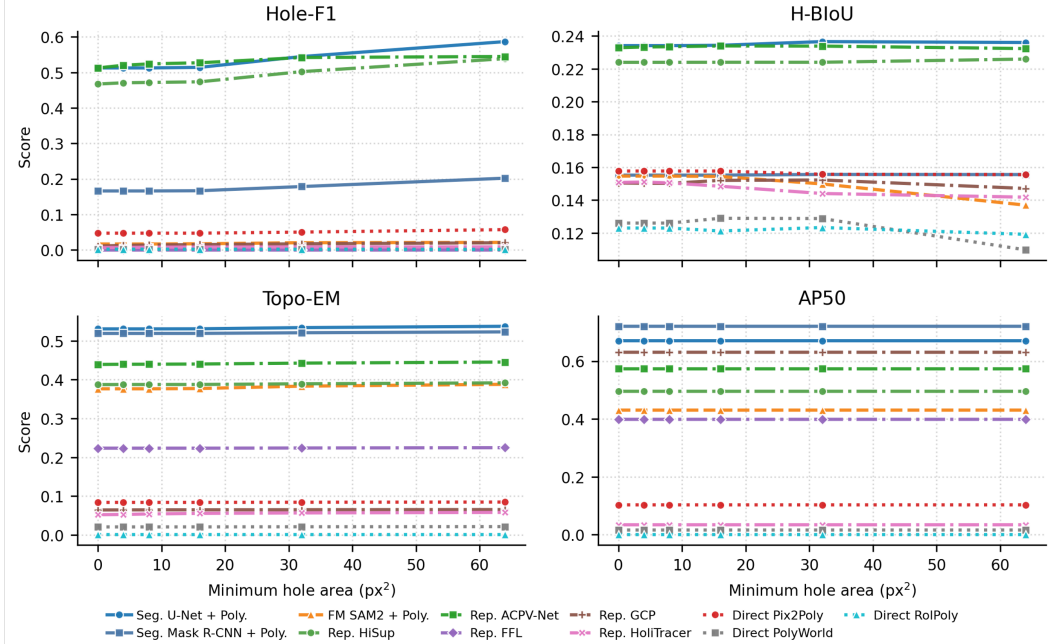


Figure 5: Metric sensitivity to the minimum hole area on the Inria Building dataset. We vary the minimum area used to keep ground-truth and predicted interior rings while fixing the ring matching threshold to $\gamma = 0.1$. Hole-F1 and H-BIoU are more sensitive to this filtering because they directly evaluate interior rings, while AP50 and Topo-EM remain relatively stable. This shows that the proposed hole-aware metrics are not dominated by a single choice of minimum hole size.

Overall, this analysis shows that our conclusion is not driven only by tiny holes: current methods still struggle with topology even when small holes are filtered out.

A.2.2 Metric Sensitivity to Ring Matching Threshold γ

Figure 6 studies how the proposed ring-structure metrics change when the ring matching threshold γ becomes stricter. We report macro-averaged Hole-F1 and Topo-EM over the four tasks and group methods by their taxonomy. As expected, both metrics of all models decrease as γ increases because a predicted interior ring must align more closely with the ground-truth ring to be counted as a match.

The main ranking pattern is stable under this sweep. For Hole-F1, U-Net + Poly., ACPV-Net, and HiSup remain the strongest methods across thresholds, while most FM and Direct methods stay close to zero. For Topo-EM, U-Net + Poly. and Mask R-CNN + Poly. remain the strongest Seg. baselines, and HiSup remains the strongest Rep. method. This shows that the low topology scores are not caused by one arbitrary threshold choice of γ . Stricter matching lowers the absolute scores, but it does not change the conclusion that current baselines still fail to recover the complete interior-ring topology.

A.2.3 Metric Robustness to Vertex Density

The same polygon shape can be represented with different numbers of vertices, e.g., a straight edge can be stored as two vertices or with additional collinear points. Most PolyTopoBench metrics are invariant to this choice by construction because they measure the shape rather than its vertex list. AP50 depends only on the occupied area, and the same shape always has the same area. BIoU buffers each boundary into a band of width r (Definition 12) and compares the two bands, so two vertex lists that trace the same boundary produce the same band and the same score. Hole-F1 and Topo-EM match rings with per-ring BIoU and inherit the same invariance. MTA samples contours at a fixed spacing of 2 pixels, so vertex density enters only through the positions of the samples along each edge. POLIS is the only metric whose definition depends directly on vertex density, because it averages distances from each vertex to the other boundary, so vertex density acts as a weight. We keep POLIS unchanged because it is a standard metric and preserving its original definition allows direct comparison with prior work.

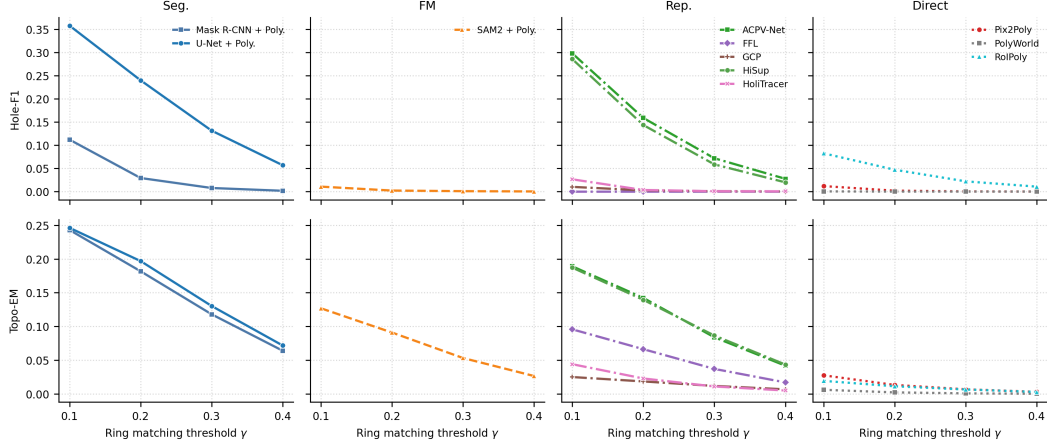


Figure 6: Metric sensitivity to the ring matching threshold γ . Scores are macro-averaged over the four tasks and grouped by method type. Increasing γ makes ring matching stricter, which lowers both Hole-F1 and Topo-EM, but the leading methods remain largely stable across thresholds.

Table 4: Metric robustness to vertex density on Deventer Road. Collinear midpoints are inserted into every predicted ring to obtain $2\times$ and $4\times$ more vertices without changing the shape. AP50, E-BIoU, Hole-F1, and Topo-EM are identical for all three densities; POLIS (Overall scope) is reported for $1\times / 2\times / 4\times$ vertices.

Method	AP50	E-BIoU	Hole-F1	Topo-EM	POLIS ($1\times / 2\times / 4\times$)
U-Net + Poly.	0.520	0.311	0.466	0.144	5.16 / 5.13 / 5.12
HiSup	0.389	0.257	0.303	0.137	6.49 / 6.51 / 6.50
ACPV-Net	0.361	0.281	0.394	0.137	6.35 / 6.31 / 6.29
RoIPoly	0.002	0.221	0.192	0.009	9.27 / 9.32 / 9.32

We verify this property empirically. We take the predictions of four methods on Deventer Road, insert collinear midpoints into every ring to produce $2\times$ and $4\times$ more vertices while preserving exactly the same shapes, and re-run the evaluator. Table 4 shows that AP50, E-BIoU, Hole-F1, and Topo-EM are identical to four decimal places across all three vertex densities, while POLIS changes by less than 1%. MTA changes by at most 4.1% (U-Net + Poly., O-MTA $52.66 \rightarrow 54.83$ at $4\times$ vertices), which does not alter the method ordering.

A.3 Baseline Implementation Details

This section discusses the training and inference settings that produced the experimental results shown in Table 3. All baselines are trained per PolyTopoBench task under a 512×512 crop using their native loss and optimizer, and consume the same train/val split defined by PolyTopoBench. Unless otherwise stated, we run each training job on a single GPU and launch different baselines or tasks as independent jobs. The main benchmark runs were executed on a node with four NVIDIA RTX A6000 GPUs, each with 48GB memory, using eight CPU dataloader workers per job. We fix the random seed for all baselines to ensure a fair comparison. All 11 baselines are trained with the same multi-ring ground truth, converted into each method’s native supervision format, so interior-ring annotations are available to every method during training. Below, we describe the implementation details of each baseline. For every method, we start from its released repository and follow the default implementation unless otherwise stated.

Selection of training settings. The baselines have substantially different pipelines, and our goal is to let each method reach its best achievable performance on PolyTopoBench. We therefore follow a two-step procedure. For every baseline, we first use the official implementation and the training configuration recommended by its authors; this is always our priority. We then examine the resulting performance and run additional ablations that deviate from the official setting only when the results are clearly abnormal, particularly with respect to interior-ring accuracy (Appendix A.5). The main table reports the best setting identified for each baseline. Initialization follows the same principle. We explicitly ablate initialization for GCP (training from scratch versus transfer learning, Table 5) and PolyWorld (training from scratch, fine-tuning, and zero-shot evaluation, Table 9). For the remaining methods, we do not sweep alternative initialization strategies, because their official implementations

already perform as expected and their authors do not recommend alternative initialization settings; for example, the released Pix2Poly implementation is designed to train its ViT encoder from scratch. Given our limited compute budget, we focus additional ablations on the methods and settings where they are most likely to affect performance, especially interior-ring recovery.

U-Net + Poly. [38, 55] We train U-Net with an EfficientNet-B3 encoder (ImageNet pretraining) for 100 epochs at batch size 32, learning rate 10^{-4} , and a combined Dice and cross-entropy loss. Masks are thresholded at 0.5 and polygonized by Douglas–Peucker simplification at tolerance 1.0 px.

Mask R-CNN + Poly. [11, 55] We fine-tune Mask R-CNN with a ResNet-50 FPN backbone from COCO-pretrained weights for 100 epochs at batch size 6 and learning rate 5×10^{-3} . Predicted masks with detector score ≥ 0.05 are polygonized with the same procedure as U-Net + Poly.

SAM2 + Poly. [33] We use SAM2 without fine-tuning. For each image, we run the Mask R-CNN detector trained above, keep predicted boxes with score ≥ 0.5 , and prompt SAM2 with each box. The returned masks are polygonized with the same tolerance and area thresholds as U-Net + Poly. We conduct an ablation study on different prompts used for SAM2. Table 10 shows that Mask R-CNN box prompts are the selected realistic setting for the main benchmark; the ground-truth box prompt is reported only as an upper-bound diagnostic.

HiSup. [45] We train HiSup with an HRNet-48 backbone on 512×512 image crops at batch size 8 and base learning rate 10^{-4} . Attraction-field, vertex, and mask loss weights follow the released configuration.

ACPV-Net. [17] We train ACPV-Net with an HRNet-32 backbone for 4×10^5 iterations at batch size 8 and base learning rate 6×10^{-5} . Other settings follow the released configuration.

FFL. [10] We train Frame Field Learning with its default U-Net backbone for a total budget of 100 epochs at batch size 13, polygonization learning rate 10^{-2} , and learning-rate decay $\gamma = 0.99$, and report the checkpoint with the best validation performance, which is reached after 5 epochs. This checkpoint occurs early because FFL overfits quickly on our data, especially on hole-bearing buildings. The original FFL paper uses early stopping at 15–25 epochs [10]; our 100-epoch budget with best-validation checkpoint selection is therefore more generous than the official recipe.

GCP. [49] GCP is trained in two stages. Stage 1 trains the segmentation and corner prediction backbone for 24 epochs at batch size 24 and learning rate 10^{-4} . Stage 2 fine-tunes the Transformer contour-regression module for another 24 epochs under the same batch size and learning rate, initialized from the stage 1 checkpoint. The global collinearity polygonizer keeps the tolerance and vertex length from its released implementation. Table 5 compares training from scratch with transfer from the WHU-Mix checkpoint; training from scratch gives the stronger AP50, Hole-F1, and Topo-EM in this setting, so the main benchmark uses the scratch-trained GCP model.

HoliTracer. [43] HoliTracer is trained in two stages with a Swin-L backbone. The Context Attention segmentation network is trained for 10 epochs at batch size 7 and learning rate 10^{-5} . The Polygon Sequence Tracer is then trained for 10 epochs at batch size 8 and learning rate 10^{-3} , using the stage-1 checkpoint as its backbone. Table 6 sweeps the corner confidence threshold. The sweep selects corner threshold 0.10 for the reported HoliTracer setting, but the low Hole-F1 and Topo-EM show that changing this threshold alone does not solve interior-ring recovery.

Pix2Poly. [2] We train Pix2Poly end-to-end for 200 epochs at batch size 28 and learning rate 2×10^{-4} , using the released ViT-based encoder without ImageNet pretraining. Table 8 ablates the maximum vertex length N_{vert} and affine-rotation augmentation. The main benchmark reports $N_{\text{vert}} = 224$ without affine rotation, the strongest configuration under these tested settings.

PolyWorld. [54] We train PolyWorld from scratch with its R2U-Net backbone for 100 epochs at batch size 16 and learning rate 10^{-4} . The Sinkhorn-style optimal matching and vertex-detection hyperparameters follow the released configuration. The training from scratch setting, the zero-shot from the pretrained checkpoint setting, and the fine-tuning variants are compared in Table 9.

RoIPoly. [16] RoIPoly uses a ResNet-50 backbone with 160 learnable proposals per image and 64 corner queries per polygon. We train the polygon head for 1.35×10^5 iterations at batch size 8 and learning rate 2.5×10^{-5} . During model inference, proposals come from a separately trained Sparse R-CNN [40] detector (ResNet-50 backbone with ImageNet pretraining) rather than ground-truth boxes. The impact of this proposal source on Hole-F1 is characterized in Table 7. The main table

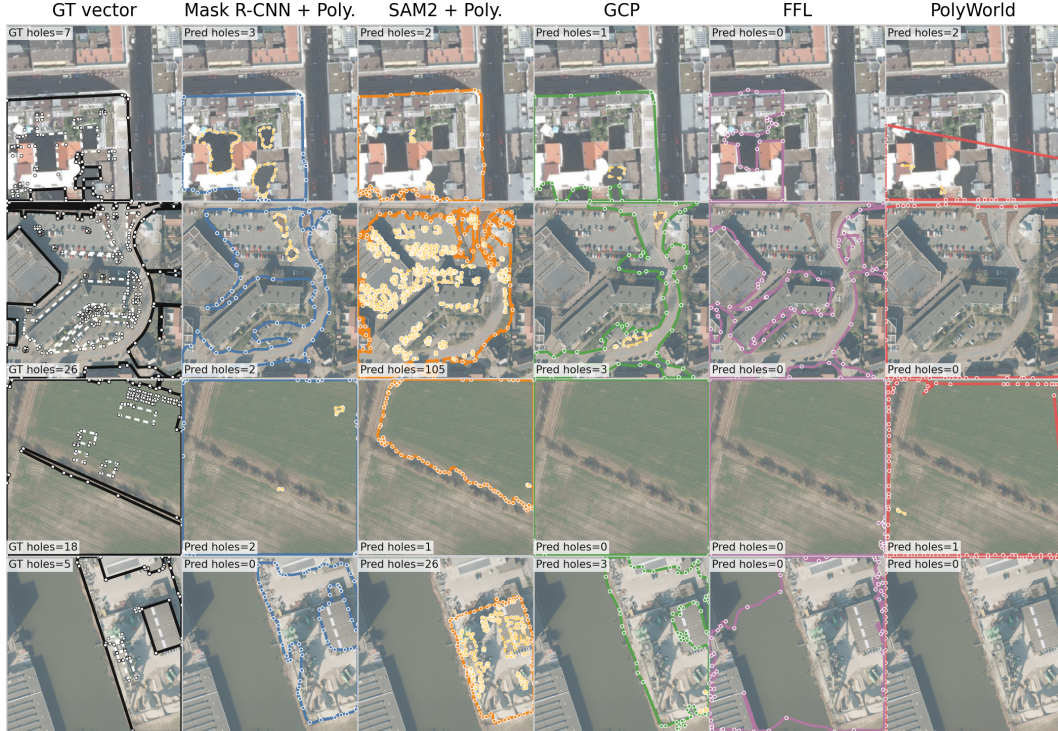


Figure 7: Additional qualitative vector predictions. Rows show building, road, vegetation, and unvegetated examples. Columns show the ground-truth vector and five representative methods: Mask R-CNN + Poly. (*Seg.*), SAM2 + Poly. (*FM*), GCP (*Rep.*), FFL (*Rep.*), and PolyWorld (*Direct*). The first column uses black exterior rings for ground truth. Prediction columns show only the predicted vector output: colored solid lines are exterior rings, dashed yellow lines are interior rings, and small circles are vertices. Patch labels report the ground-truth or predicted hole count.

Table 5: GCP setting ablation on Inria Building. Since GCP releases its checkpoint on WHU-Mix [28], we ablate the transfer learning setting and the training from scratch setting.

Setting	AP50	E-BIoU	Hole-F1	Topo-EM
Train from the released WHU-Mix checkpoint	0.190	0.588	0.000	0.004
Train from scratch	0.631	0.312	0.015	0.065

(Table 3) uses the Sparse R-CNN proposals. The selection of Sparse R-CNN follows the original paper [16].

A.4 Additional Qualitative Results

This section provides additional qualitative examples, as shown in Figure 7– 11. We can observe the same trend as we describe in the main paper.

A.5 Method Setting Ablation

This section reports method-level setting ablations used in the discussion. We ablate each baseline under its relevant training or inference settings. All ablation tables in this section use the Inria Building dataset, so the rows compare variants within each method on the same dataset. All values are rounded to three decimals. As shown in Tables 5–10, the final setting selected for the main benchmark is highlighted in bold.

A.5.1 Failure Analysis by Method Family

All baselines are trained with the same multi-ring ground truth (Appendix A.3), so their low interior-ring scores arise from their polygon representations rather than from missing ring annotations. Figure 12 compares all 11 methods on the same examples, and we analyze the failure modes of each method family below.



Figure 8: Additional qualitative vector predictions with segmentation and representation-to-vector baselines. Rows show the four object categories in the same order as Figure 7. Columns show the ground-truth vector, U-Net + Poly. and Mask R-CNN + Poly. (Seg.), HiSup, HoliTracer, and ACPV-Net (Rep.). Prediction columns contain no ground-truth overlay, so the figure directly shows each method’s predicted exterior ring, interior rings, and vertices.

Table 6: HoliTracer setting ablation on the Inria Building dataset. We sweep the corner confidence threshold used by the vector tracing stage.

Setting	AP50	E-BoU	Hole-F1	Topo-EM
Corner threshold 0.00	0.034	0.558	0.000	0.000
Corner threshold 0.10	0.034	0.558	0.000	0.000
Corner threshold 0.50	0.000	—	0.000	0.000

Segmentation-based methods. A hole is preserved only if the predicted mask retains the enclosed background. The smoothness bias of segmentation models often fills small holes, which cannot be recovered by the subsequent polygonization.

Representation-based methods. These methods predict attraction fields, frame fields, or contours and then polygonize them. Both stages primarily emphasize object boundaries, so exterior rings are recovered more reliably than interior rings. For example, GCP’s collinearity-based simplification yields a Hole-F1 of only 0.015 on Inria (Table 5).

Direct vector decoders. On PolyTopoBench, the direct methods fail at a more basic level than hole recovery: they do not produce enough complete instances, so their scores collapse already at the region level (AP50), before the ring-structure metrics apply. This differs from their original benchmarks, where each image contains a few simple, compact buildings. AP50 requires complete polygons at $\text{IoU} \geq 0.5$ together with a usable confidence ranking, and each direct method breaks one of these requirements.

- *Pix2Poly* generates all polygons of a patch as one token sequence with a fixed maximum vertex budget. In the original paper, this budget only needs to cover exterior corner points of a few compact buildings in small crops. Each 512×512 PolyTopoBench patch contains many instances, and the model must also emit vertices on interior rings, which the original task never required. Increasing the vertex budget raises AP50 only from 0.000 to 0.102 (Table 8).

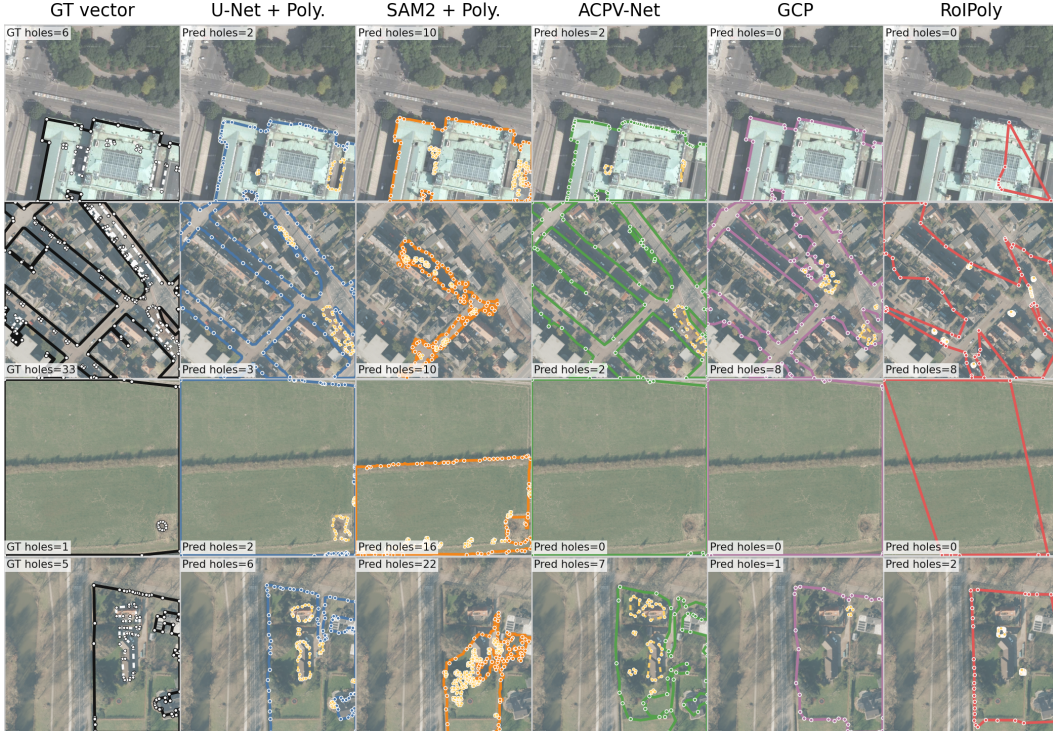


Figure 9: Additional qualitative vector predictions comparing all four method types. Columns show the ground-truth vector, U-Net + Poly. (*Seg.*), SAM2 + Poly. (*FM*), ACPV-Net and GCP (*Rep.*), and RoIPoly (*Direct*). Across the four object categories, exterior geometry can remain visually plausible while the number and placement of interior rings differ strongly from the ground truth.

Table 7: RoIPoly setting ablation on the Inria Building dataset. We compare the polygon head under ground truth boxes with a setting where a separately trained Sparse R-CNN [40] detector provides the bounding boxes.

Proposal source	Setting	AP50	E-BoU	Hole-F1	Topo-EM
Ground truth boxes (upper bound)	Evaluate the RoIPoly polygon head with ground truth object boxes.	0.158	0.680	0.310	0.082
Sparse R-CNN boxes	Train a separate Sparse R-CNN detector on Inria and use its predicted boxes as RoIPoly proposals.	0.001	0.135	0.002	0.001

- *PolyWorld* detects corner points and links them into closed rings through vertex-permutation cycles. Although this can in principle represent holes, the model has no explicit notion of ring role or hierarchy, and it must detect small, low-contrast interior vertices and connect them into separate cycles. On our complex polygons it misses too many corners, so rings do not close properly, the output becomes fragmented, and AP cannot rank good polygons above fragments (Section 3.4).
- *RoIPoly* is a two-stage method that first proposes bounding boxes with an object detector and then decodes one ring per proposal. With ground-truth boxes it reaches $AP50 = 0.158$ and $Hole-F1 = 0.310$, but with detector proposals $AP50$ drops to 0.001 and $Hole-F1$ to 0.002 (Table 7). The polygon head is functional, and the proposal stage is the main bottleneck.

In summary, the $AP50$ collapse of direct methods is caused by instance discovery rather than hole recovery. All three direct methods were designed for scenes containing a few simple buildings; on dense patches with complex polygons they fail to find and complete the instances in the first place. This is a limitation of how these models represent their outputs rather than of how they are trained, and hyperparameter tuning alone cannot resolve it. These results motivate explicitly supervising interior rings and enforcing ring hierarchy and validity during decoding (Section 4.4).

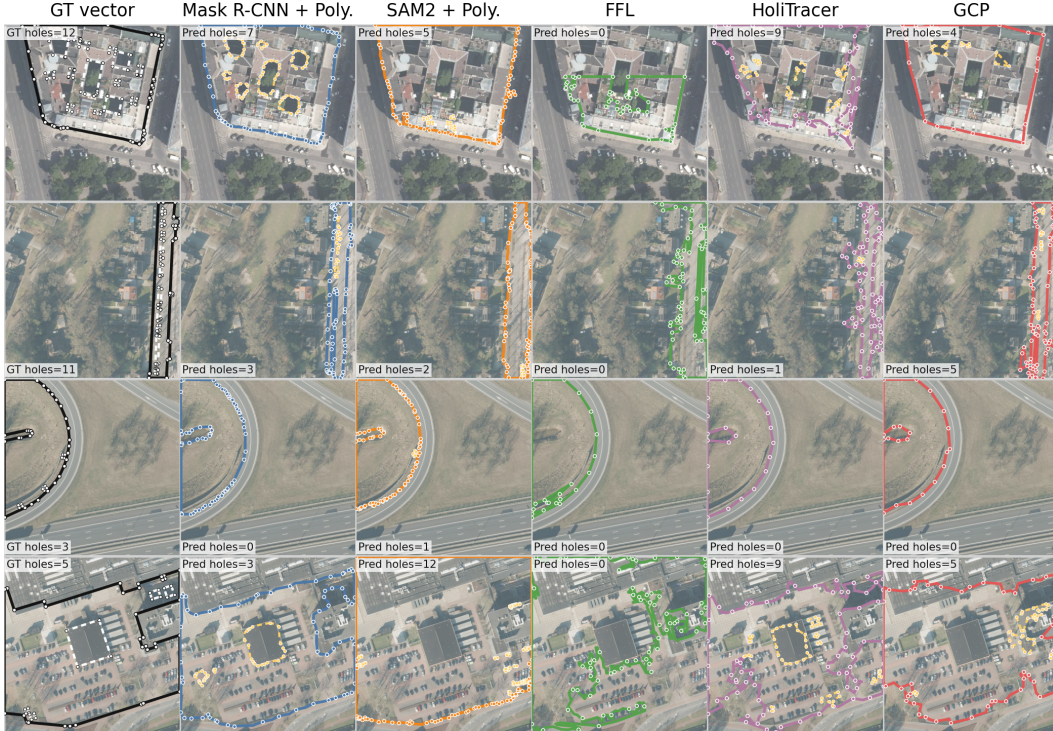


Figure 10: Additional qualitative vector predictions emphasizing representation-to-vector methods. Columns show the ground-truth vector, Mask R-CNN + Poly. (*Seg.*), SAM2 + Poly. (*FM*), FFL, HoliTracer, and GCP (*Rep.*). The examples illustrate common topology errors: missing holes, extra holes from fragmented evidence, and exterior rings that trace nearby structures rather than the target instance.

Table 8: Pix2Poly setting ablation on the Inria Building dataset. Pix2Poly predicts vertex tokens with an image-to-sequence Transformer and recovers connectivity with an optimal matching network. We mainly ablate the maximum vertex-sequence length and image data augmentation strategy.

Variant	AP50	E-BIoU	Hole-F1	Topo-EM
512 input with affine rotation, 192 maximum vertices	0.000	0.592	0.000	0.000
512 input without affine rotation, 192 maximum vertices	0.000	0.608	0.000	0.000
512 input without affine rotation, 224 maximum vertices	0.102	0.183	0.047	0.084

A.6 Annotation Protocol and Quality Control

Inria Building. The Inria Aerial Image Labeling release [29] provides 180 tiles of 5000×5000 pixels across five cities with binary foreground raster masks but no vector annotation. We take OpenStreetMap [35] building footprints as the vector source because they provide topologically clean multi-ring polygons with rich metadata, and correct two known issues before release: (1) 2–8 pixel geo-registration offsets against the imagery, and frequent omission of inner courtyards. To fix the offsets, for every OSM polygon we search a translation within ± 16 pixels that maximises IoU with the overlapping raster connected component, accept the translation when the resulting IoU exceeds 0.5, and drop polygons without a majority-overlap component. The authors then review the aligned polygons tile by tile and manually correct residual mismatches. To recover missing courtyards, we vectorise the raster mask by tracing connected components and extracting nested rings, then transfer every nested ring that falls inside an aligned OSM exterior as a candidate interior ring; this step produces most of the hole-bearing annotations in Table 1. We finally apply automated filters to every polygon. Firstly, self-intersections via `shapely.buffer(0)`, and reassigning interior rings that are not strictly contained in their exterior via `shapely.contains`. Tiles are sliced into non-overlapping 512×512 patches and split at the tile level (145/35 for train/val), stratified by city and hole-bearing fraction, and the split is shared by all baselines.



Figure 11: Additional qualitative vector predictions including direct vector decoders. Columns show the ground-truth vector, U-Net + Poly. (*Seg.*), ACPV-Net and HiSup (*Rep.*), Pix2Poly and RoIPoly (*Direct*). Direct decoders often produce simple or distorted exterior rings and miss many interior rings, while representation-to-vector methods recover more local structure but still fail to match the complete ring topology.

Table 9: PolyWorld setting ablation on the Inria Building dataset. PolyWorld detects vertex peaks, extracts visual descriptors, and predicts vertex connectivity with a GNN and Sinkhorn-style optimal matching. The variants test whether better vertex supervision or longer fine-tuning fixes topology errors.

Variant	AP50	E-BIoU	Hole-F1	Topo-EM
Zero-shot using pretrained checkpoint	0.004	0.600	0.000	0.001
Fine-tuning	0.012	0.544	0.000	0.000
Train from scratch	0.016	0.180	0.002	0.021

Deventer land cover. The Deventer tasks reuse the *Deventer-512* vector annotations of Jiao et al. [17]; we select three of their six classes (*Road*, *Vegetation*, *Unvegetated*) because they contain the most multi-ring structure (Table 1) and we inherit the released train/val split. We do not re-annotate and only apply the automated filters applied in Inria, and review the annotations by our authors. We refer the reader to [17] for the upstream annotation procedure.

Limitations. Our Inria annotation inherits two error sources. First, OSM itself can be incomplete or temporally mismatched with the imagery, so buildings that appear in only one source are dropped rather than re-annotated; we do not cross-validate against a second independent vector source due to the limited publicly available vector data. Second, the hole-restoration step relies on the Inria raster mask and therefore inherits any omissions or false positives in the official raster label. For the Deventer dataset, we rely on the upstream annotation of [17]. Though our authors examine the annotations, some wrong labels may still exist.

A.7 Bootstrap Confidence Intervals

We additionally report image-level bootstrap confidence intervals to estimate the statistical stability of the main benchmark metrics. We first compute the evaluation statistics separately for each image, using the saved predictions and ground truth annotations. For one bootstrap trial on a task with N evaluation images, we sample N image IDs from the original image list. The sampling is with

Table 10: SAM2-assisted polygonization setting ablation on the Inria Building dataset. Each variant changes only the SAM2 prompt source: ground truth boxes, Mask R-CNN predicted boxes, U-Net predicted boxes, U-Net masks with boxes, iterative SAM2 mask refinement, U-Net mask-only prompts, or U-Net box-only prompts. All SAM2 masks are converted to polygons using the same contour extraction and simplification procedure.

Prompting variant	Prompt construction	AP50	O-BIoU	Hole-F1	Topo-EM
Ground truth box prompt (upper bound)	Use each ground truth instance box as the SAM2 box prompt.	0.824	0.266	0.000	0.050
Mask R-CNN box prompt	Use Mask R-CNN predicted boxes as SAM2 box prompts.	0.431	0.242	0.017	0.377
U-Net mask + box	Use a U-Net predicted mask together with its bounding box.	0.506	0.251	0.000	0.029
Iterative mask refinement	Run SAM2 with U-Net box and mask prompts, then feed the predicted mask back for a second refinement pass.	0.473	0.242	0.000	0.027
U-Net box only	Use the bounding box of each U-Net predicted instance mask.	0.402	0.243	0.000	0.024
U-Net mask only	Use the U-Net predicted mask as the only SAM2 prompt.	0.035	0.203	0.000	0.001



Figure 12: Qualitative comparison of all 11 methods on the same patches. Rows show building, road, vegetation, and unvegetated examples; columns show the ground truth followed by every benchmark method. All methods share one color scheme: black lines are ground-truth exterior rings, red lines are predicted exterior rings, dashed lines are interior rings, and white dots are vertices. Each panel reports its ground-truth or predicted hole count, and “no pred” means the method produced no output overlapping the patch. *Seg.* and *Rep.* methods such as U-Net + Poly., HiSup, and ACPV-Net generally recover exterior boundaries and some interior rings, but the predicted hole counts and locations often differ from the ground truth. SAM2 + Poly. tends to hallucinate holes (e.g., 44 predicted holes on the road example with 5 ground-truth holes). The direct methods frequently produce fragmented or incomplete instances and almost no interior rings, consistent with their low AP50.

replacement, so the same image ID can be selected more than once and some image IDs may not be selected. We then aggregate the per-image statistics over the sampled image IDs, counting repeated images repeatedly, and compute AP50, E-BIoU, Hole-F1, and Topo-EM for that sampled task. We repeat this process 500 times and report the macro-average confidence intervals over the four tasks. The results are shown in Table 11. This analysis measures uncertainty from the finite evaluation set. It is not a multi-seed training study, so it does not capture variation from retraining the same model under different random seeds.

A.8 Effect of the Simple–Complex Polygon Imbalance

Table 1 shows that simple polygons are much more frequent than complex polygons at the instance level. The imbalance is milder at the patch level, which is the actual training unit: 15.2% of the Inria training patches contain at least one complex polygon, and the corresponding proportions on Deventer are 44.0% for Road, 30.7% for Vegetation, and 55.7% for Unvegetated. Models therefore receive interior-ring supervision approximately every second to sixth patch.

Table 11: Macro-average benchmark scores with 95% image-level bootstrap confidence intervals over 500 resamples.

Method	Type	AP50	E-BIoU	Hole-F1	Topo-EM
U-Net + Poly.	Seg.	0.455 [0.437, 0.478]	0.294 [0.288, 0.299]	0.357 [0.335, 0.379]	0.246 [0.235, 0.257]
Mask R-CNN + Poly.	Seg.	0.343 [0.329, 0.361]	0.260 [0.252, 0.268]	0.112 [0.100, 0.127]	0.243 [0.233, 0.254]
SAM2 + Poly.	FM	0.134 [0.128, 0.145]	0.274 [0.264, 0.285]	0.011 [0.008, 0.013]	0.127 [0.121, 0.133]
HiSup	Rep.	0.315 [0.298, 0.338]	0.256 [0.250, 0.262]	0.286 [0.263, 0.309]	0.187 [0.177, 0.198]
ACPV-Net	Rep.	0.323 [0.306, 0.341]	0.263 [0.256, 0.270]	0.298 [0.277, 0.318]	0.189 [0.180, 0.199]
FFL	Rep.	0.207 [0.194, 0.221]	0.217 [0.207, 0.229]	0.000 [0.000, 0.000]	0.096 [0.092, 0.100]
GCP	Rep.	0.296 [0.283, 0.313]	0.249 [0.242, 0.256]	0.010 [0.008, 0.012]	0.025 [0.024, 0.026]
HoliTracer	Rep.	0.043 [0.038, 0.052]	0.202 [0.194, 0.210]	0.062 [0.052, 0.072]	0.053 [0.049, 0.059]
Pix2Poly	Direct	0.033 [0.032, 0.037]	0.252 [0.205, 0.352]	0.012 [0.010, 0.014]	0.028 [0.025, 0.030]
PolyWorld	Direct	0.005 [0.002, 0.005]	0.146 [0.132, 0.162]	0.000 [0.000, 0.001]	0.006 [0.006, 0.007]
RoIPoly	Direct	0.010 [0.008, 0.013]	0.138 [0.116, 0.162]	0.082 [0.069, 0.096]	0.019 [0.017, 0.023]

Table 12: Effect of hole-bearing patch sampling on U-Net + Poly. (Inria Building). Only the sampling probability of patches containing at least one complex polygon is changed; data, gradient steps, and seed are fixed. The natural distribution corresponds to the main benchmark setting.

Hole-bearing patches in the training stream	AP50	E-BIoU	Hole-F1	Topo-EM
Down-sampled (4.3%)	0.678	0.304	0.455	0.529
Natural (15.2%)	0.670	0.308	0.514	0.531
Up-sampled (41.8%)	0.679	0.304	0.559	0.533

The observed failure pattern does not indicate that complex polygons are treated as noise. If models simply ignored these less frequent instances, complex polygons would be missed entirely or localized poorly at the instance level. Instead, many complex instances are successfully matched to their ground truth with $\text{IoU} \geq 0.5$, showing that the models recognize the objects and recover their overall extent. Yet 84%–100% of the matched complex polygons still have incorrect ring structures (Figure 3), most commonly because one or more interior rings are missing. The models have thus learned the presence and exterior shape of complex polygons but struggle to represent their multi-ring topology.

To measure the effect of the imbalance directly, we retrain U-Net + Poly. on Inria Building while keeping the data, the number of gradient steps, and the random seed fixed, and vary only the sampling probability of hole-bearing patches with weighted random sampling. Table 12 shows that a $16\times$ increase in the sampling weight of hole-bearing patches (from 4.3% to 41.8% of the training stream) raises Hole-F1 from 0.455 to 0.559, so oversampling helps detect individual holes. Topo-EM, however, remains nearly unchanged, because it requires the complete ring structure to be correct. The imbalance therefore moderately affects hole detection but is not the main bottleneck for topology correctness. PolyTopoBench intentionally preserves the natural class distribution because it reflects how these objects occur in the real world; rebalancing would make the benchmark easier but less faithful.