Aerial2Poly: Aerial Imagery to Polygon Vectorization

Anonymous CVPR submission

Paper ID 4

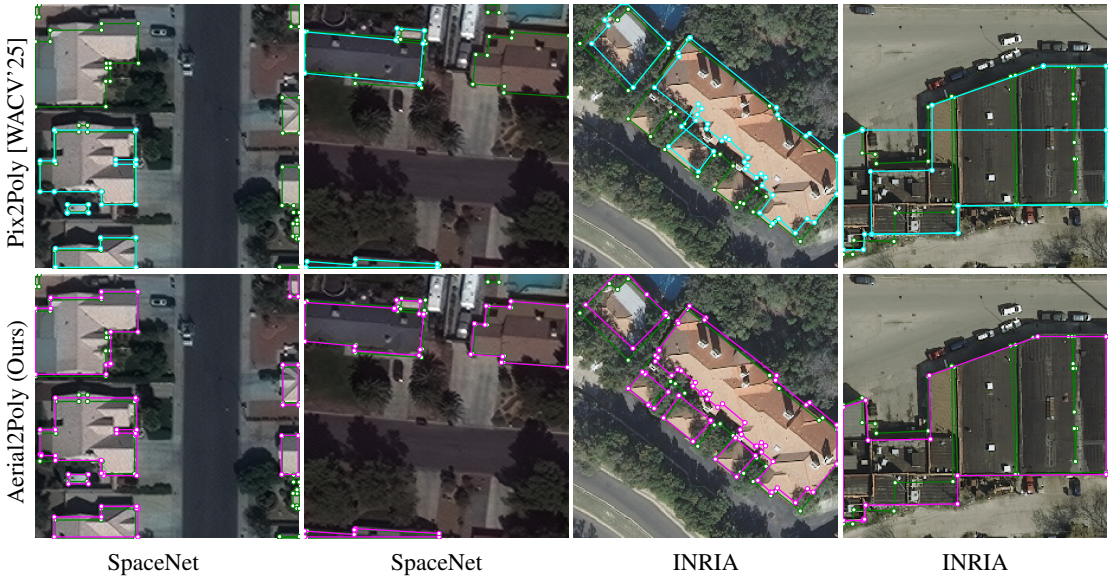


Figure 1. Aerial2Poly extends autoregressive polygon sequencing to aerial building footprint extraction and introduces Gaussian-biased multi-trial decoding to recover complete, geometrically faithful boundaries. Without task-specific modules or post-hoc vectorization, Aerial2Poly sets new state-of-the-art results on the challenging SpaceNet [6], INRIA [17], and WHU Buildings [11] benchmarks. We visualize our predicted polygons (magenta) against those from the strong baseline Pix2Poly [1] (cyan) on exemplar SpaceNet and INRIA validation images and show that our model's predictions align more closely to the ground truth (green).

Abstract

Building footprint extraction from aerial imagery is often treated as a mask-first problem, followed by contour extraction, simplification, or geometric post-processing to recover polygons. We instead formulate it as scene-level autoregressive polygon generation. We present Aerial2Poly, an encoder-decoder model that predicts the polygon set of an image directly as a sequence of coordinate, separator, and class tokens. Aerial2Poly encodes connectivity in traversal order without relying on object prompts or auxiliary connectivity modules. To make this formulation effective and scalable in practice, we introduce Gaussian-biased decoding, a recall-oriented anti-`EOS`* objective with noise polygons, and a vector-space trial consolidation procedure. Aerial2Poly sets new state-of-the-art results on the challenging SpaceNet, INRIA, and WHU Buildings benchmarks, including a 10.36% relative C-IoU gain on WHU Buildings. These results show that a simple scene-level sequence interface is a strong foundation for direct polygon extraction from aerial imagery.*

1. Introduction

Building footprint extraction from aerial imagery is a core problem in remote sensing, with applications in mapping, urban analysis, disaster response, and GIS pipelines. The desired output is vector geometry rather than raster masks: polygons preserve boundaries, vertices, and topology, and they integrate directly with editors, spatial databases, and

cadastral records. Yet many current systems remain mask-first, predicting dense segmentations and only later converting them into polygons through contour extraction, simplification, or geometric post-processing [7, 13, 18, 22]. This conversion is lossy: small mask errors surface as shifted vertices, irregular jagged outlines, and missing corners. The final polygon quality is bounded as much by the post-processing chain as by the recognition model itself.

Recent work moves toward direct polygon prediction via connectivity graphs of building corners [1, 19, 23], but important gaps remain. Decoupling recognition from connectivity introduces a failure mode in which every corner is well localized yet the polygon is topologically broken. The result is the crossed edges, missing rings, and merged neighbors visible in Figure 1. Even modern autoregressive approaches [4] decode one instance at a time conditioned on a previously detected box, rather than describing the full scene in one pass. Aerial tiles vary widely in object count and polygon complexity, so an ideal model should produce the full polygon output of a tile in a single pass.

We address these gaps with Aerial2Poly, a scene-level autoregressive model for aerial polygon extraction. Aerial2Poly generates the polygon set of the entire scene as a single sequence of coordinate, separator, and class tokens. Connectivity is encoded by the traversal order itself, so there is no need for an auxiliary matching module, contour extractor, or per-instance prompt. Moreover, our approach efficiently stops the polygon decoding procedure once the scene is described rather than running to a fixed maximum length [3]. The design of Aerial2Poly comprises three simple yet effective components: a Gaussian coordinate bias for locally coherent decoding, a recall-oriented anti-`EOS` objective trained with synthetic noise polygons, and a multi-trial consolidation step that operates entirely in polygon space.

Main Contributions. (1) We present Aerial2Poly, a scalable, end-to-end, scene-level autoregressive model for aerial polygon extraction. (2) Our framework adapts the autoregressive sequence formulation to efficiently work on dense aerial scenes with the introduction of Gaussian-biased coordinate decoding, an anti-`EOS` recall objective paired with a noisy-polygon training procedure, and a vector-space multi-trial consolidation scheme. (3) Aerial2Poly demonstrates excellent performance and scalability by establishing state-of-the-art results, with compelling margins, on the challenging INRIA, SpaceNet, and WHU Buildings benchmarks.

2. Related Work

Polygons from Segmentation Masks. Early work on polygonal building extraction relied on pixel-perfect segmentation maps learned by a convolutional neural network (CNN) to produce high-quality vector shapes via a separate post-processing step. Zhao et al. [21] proposed a boundary regularization method to refine the building instances predicted from Mask-RCNN [9]. Mask2Poly [22] trained two separate CNNs to refine raster segmentation masks into vector polygons by using a combination of adversarial, reconstruction, and regularized losses. FFL [7] imposed frame fields on top of raster segmentation maps via a multi-task objective, which were then used to generate building polygons. HiSup [18] strengthened the multi-stage polygonization pipeline with line-segment attraction fields and hierarchical supervision over masks, but polygon recovery still occurs through sub-optimal geometric post-processing.

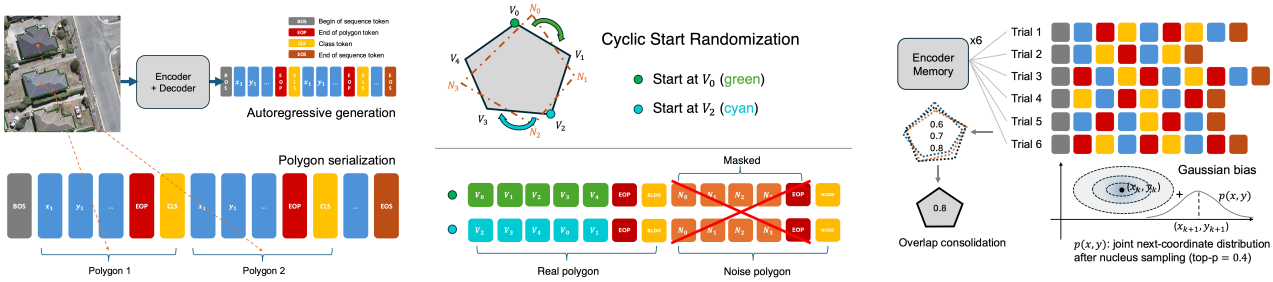
Polygons as Connectivity Graphs. A promising class of end-to-end approaches formulates the polygonization task as predicting vertices in a connectivity graph of edges [14–16]. PolyWorld [23] and TopDig [19] proposed to first detect building corner candidates from encoded image features followed by a matching network to connect polygon vertices. More recently, Pix2Poly [1] extends this graph-based framework by adopting an image-to-sequence transformer network to detect building corners as a discrete sequence prediction task [3]. Although Pix2Poly achieves strong polygonal results, topological errors and misalignment can still arise from the auxiliary graph connectivity module.

Autoregressive Polygon Sequencing. An emerging line of work treats visual geometry predictions as a series of sequence generations. Pix2Seq [3] casts object detection as autoregressive prediction over quantized coordinates and class tokens with a generic encoder-decoder transformer network. Pix2Seq v2 [4] extends the same interface to multiple tasks, including polygon-based instance segmentation. These approaches establish a shared token vocabulary for geometric outputs, but they do not study scene-level polygon decoding for remote sensing applications such as building extraction, map generation, and aerial object detection.

Recent HiT [20] and GeoFormer [12] approaches have adapted autoregressive polygon sequencing to aerial imagery, but not without limitations. HiT requires an auxiliary region proposal network for building detection together with polygon serialization. GeoFormer requires multiple forward passes to extract all buildings and is not scalable in dense urban scenes. By contrast, our method encodes connectivity directly in autoregressive traversal order and efficiently decodes the polygon set of the full image in a single pass without task-specific auxiliary modules or object-level prompts.

3. The Aerial2Poly Framework

We cast building footprint extraction as scene-level pixel-to-sequence prediction. Given an aerial image, the decoder emits the polygon set directly as coordinate, separator, and class tokens. The remainder of this section describes sequence construction, the model architecture and training



(a) Polygon serialization (b) Cyclic start & noise augmentation (c) Gaussian-biased multi-trial merging

Figure 2. **Overview of Aerial2Poly.** (a) Building footprint polygons are serialized into a flat token sequence of quantized coordinates, polygon separators (EOP), class tokens, and a shared BOS/EOS framing. A ViT encoder and a Transformer decoder predict this sequence autoregressively. (b) During training, cyclic start randomization lets any vertex begin a polygon's token subsequence (top), and synthetic noise polygons are appended with a dedicated noise class and masked coordinate losses to improve recall (bottom). (c) At inference, multiple stochastic trials are decoded with nucleus sampling; a Gaussian spatial bias on coordinate logits encourages locally coherent boundaries, and the resulting polygon sets are merged in vector space via confidence filtering and overlap consolidation.

objective, and inference, with the main serialization and decoding steps illustrated in Figure 2.

Sequence Construction. Let I denote the input aerial image. We serialize its annotation as a single token sequence. The ith polygon ring of I is encoded as

$$P_i = [x_1^i, y_1^i, \ldots, x_{n_i}^i, y_{n_i}^i, \texttt{EOP}, c_i], \tag{1}$$

where coordinates are uniformly quantized into B bins, c_i is a class token, and in all experiments B equals the input image resolution. Rings are concatenated between BOS (begin-of-sequence) and EOS (end-of-sequence) in a shared vocabulary over coordinate, class, and special tokens. Because the polygon set is unordered and each ring is cyclic, we randomize ring order and circularly shift each ring's start vertex during training [4]. For annotations with holes, exterior rings keep the object class and interior rings use a dedicated interior token.

Architecture and Objective. The image encoder maps the pixels of I to a sequence of visual features and can be instantiated as a pre-trained CNN (*e.g.*, ResNet [8]) or Vision Transformer [5]. A Transformer decoder predicts the polygon sequence $\boldsymbol{y} = (y_1, \ldots, y_T)$ autoregressively. Task structure is encoded by the sequence itself rather than specialized prediction heads. Conditioned on the encoded image features, the decoder models

$$P(\boldsymbol{y} \mid I) = \prod_{t=1}^{T} P(y_t \mid \boldsymbol{y}_{<t}, I), \tag{2}$$

and is trained with teacher forcing under the weighted cross-entropy objective (optionally with label smoothing):

$$\mathcal{L} = \frac{\sum_{t=1}^{T} \boldsymbol{w}_t \, \mathrm{CE}_t}{\sum_{t=1}^{T} \boldsymbol{w}_t} + \lambda_{\text{ae}} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} P(\texttt{EOS} \mid \boldsymbol{y}_{<t}, I), \tag{3}$$

where $\mathrm{CE}_t = -\log P(y_t \mid \boldsymbol{y}_{<t}, I)$, $\mathcal{T}$ denotes all non-padding, non-EOS positions, and $\boldsymbol{w}_t$ is a token-specific weight. We set the weights of coordinate, EOP, and class tokens to 1.0, the weight of EOS to 0.5, and $\lambda_{\text{ae}} = 0.1$. The lower EOS weight and anti-EOS term reduce premature termination by penalizing excessive EOS probability.

We also append synthetic noise polygons during training and assign them a dedicated noise class token. Each noise polygon is derived from a randomly selected ground-truth ring by jittering and randomly dropping vertices. Their coordinate and EOP labels are masked while the noise class token remains supervised, as illustrated in Figure 2(b).

Inference. Prior autoregressive approaches either decode to a fixed maximum length [3] or condition polygon generation on per-instance prompts [4]. We instead decode at the scene level: one trial produces a polygon set for the full image, and the final prediction repeats this prompt-free decoding K times with nucleus sampling [10], as shown in Figure 2(c). Each trial is variable-length and stops at EOS, optionally with a small EOS-logit penalty to reduce premature termination. The decoding budget is therefore scene-adaptive, which suits aerial tiles that range from a few sparse buildings to hundreds in dense urban scenes, and the number of trials is independent of the number of objects in the tile. All experiments use $K = 6$ trials, top-p sampling with $p = 0.4$, and the same decode-time EOS setting across datasets. Raw trial sequences are converted to polygons without per-trial NMS and then passed to a shared merge-and-filter pipeline.

We add a Gaussian spatial bias to coordinate logits during decoding. For a coordinate bin k, the biased logit is

$$\tilde{\ell}_k = \ell_k + \alpha \exp\left(-\frac{(k - \hat{k})^2}{2\sigma^2}\right), \tag{4}$$

INRIA (155) val	Method	Encoder	Decoder	GPU Hrs	IoU (%) ↑	C-IoU (%) ↑	MTA (°) ↓	PoLiS ↓	IoU^{topo}(%) ↑	F1^{topo}(%) ↑	PA^{topo}(%) ↑
	FFL [CVPR'21] [7]	–	–	–	68.30	49.80	35.62	2.865	43.38	58.78	89.67
	HiSup [ISPRS'23] [18]	–	–	–	74.90	66.10	43.86	2.438	53.51	67.94	93.20
	Pix2Poly [WACV'25] [1]	ViT-S/8	6L-256d	–	79.46	71.73	**34.31**	**1.914**	61.08	74.29	94.37
	Aerial2Poly (Ours)	ViT-S/8	6L-256d	12.93	**85.02**	**78.55**	34.74	1.944	**72.75**	**81.22**	**96.32**

SpaceNet Vegas val	Method	Encoder	Decoder	GPU Hrs	IoU (%) ↑	C-IoU (%) ↑	MTA (°) ↓	PoLiS ↓	IoU^{topo}(%) ↑	F1^{topo}(%) ↑	PA^{topo}(%) ↑
	FFL [CVPR'21] [7]	–	–	–	76.00	57.60	36.29	2.398	49.46	65.00	91.10
	HiSup [ISPRS'23] [18]	–	–	–	82.10	75.20	33.89	1.722	59.56	73.43	93.80
	Pix2Poly [WACV'25] [1]	ViT-S/8	6L-256d	–	81.81	75.05	33.40	**1.717**	60.31	74.20	93.80
	Aerial2Poly (Ours)	ViT-S/8	6L-256d	13.75	**84.32**	**77.56**	**33.10**	1.751	**67.40**	**77.81**	**95.63**

WHU Buildings test	Method	Encoder	Decoder	GPU Hrs	IoU (%) ↑	C-IoU (%) ↑	MTA (°) ↓	PoLiS ↓	IoU^{topo}(%) ↑	F1^{topo}(%) ↑	PA^{topo}(%) ↑
	FFL [CVPR'21] [7]	–	–	–	77.64	54.52	35.79	1.747	56.56	70.01	94.02
	HiSup [ISPRS'23] [18]	–	–	–	87.12	79.62	34.75	1.158	72.11	82.47	96.80
	Pix2Poly [WACV'25] [1]	ViT-S/8	6L-256d	–	89.15	81.63	31.64	1.082	75.38	84.96	97.14
	Aerial2Poly (Ours)	ViT-S/8	6L-256d	12.85	**93.79**	**90.09**	**31.58**	**0.993**	**87.17**	**91.68**	**98.65**

Table 1. Main results on building footprint prediction. GPU hours report total training compute to 240k iterations on 8 NVIDIA L4 GPUs as a proxy for training cost. Results for methods not proposed in this work are quoted from Pix2Poly.

INRIA (155) test	Method	Encoder	Decoder	IoU (%) ↑	Acc (%) ↑
	Mask2Poly [22]	–	–	74.40	96.10
	FFL [7]	–	–	74.80	95.96
	HiSup [18]	–	–	75.53	96.27
	Pix2Poly [1]	ViT-S/8	6L-256d	75.87	96.37
	Aerial2Poly (Ours)	ViT-S/8	6L-256d	**76.88**	**96.53**

Table 2. Official INRIA test-set results from the online evaluator. Decoder is reported as depth-hidden size. Results for methods not proposed in this work are quoted from Pix2Poly.

where $\hat{k}$ is the most recently decoded coordinate bin on the same axis within the current polygon: the previous x-bin when the next token is x, or the previous y-bin when the next token is y. The bias is inactive for the first coordinate on each axis and reset after every `EOP`. In all experiments, we set the spatial bias strength to $\alpha = 1.5$ and $\sigma = 40$ coordinate bins, equivalent to pixels at our resolution. Applied only to coordinate tokens, the bias encourages local boundary coherence without altering separator or class prediction.

After discarding polygons predicted as `noise` along with very small or low-confidence ones, we consolidate the K trial outputs per class in vector space. The trials are split into two groups, an IoU-medoid is selected per overlap cluster within each group, and the survivors are merged across groups using a shared polygon-IoU threshold of 0.45. Interior polygons are consolidated as a separate class and subtracted from exterior polygons only when fully contained.

4. Experiments

Experimental Setup. We evaluate Aerial2Poly under the shared protocol for polygonal extraction following previous work [1, 2, 18, 23]. For a direct comparison with the Pix2Poly baseline, we use the same ViT-S/8 image encoder at 224×224 resolution and a Transformer decoder of matching scale with 6 layers and hidden size 256. We train on the INRIA, SpaceNet Vegas, and WHU Buildings benchmarks following the same data splits and patchification procedure as HiSup and Pix2Poly. Unless noted otherwise, our model and training hyperparameters are fixed across datasets. We also fix the inference recipe across datasets: $K = 6$ trials, top-p sampling with $p = 0.4$, Gaussian bias with spatial bias strength $\alpha = 1.5$ and $\sigma = 40$ coordinate bins, raw-trial polygon extraction without per-trial NMS, and vector-space trial consolidation at polygon-IoU threshold 0.45, followed by the same final cross-class and area filters.

Metrics. Following the established evaluation protocol above, we report polygon quality using IoU, C-IoU (which penalizes vertex-count mismatch in addition to overlap), MTA, PoLiS, and the topology-aware IoU/F1/PA boundary metrics. For INRIA, we also report the official online test-set IoU and Accuracy metrics from the public evaluator.

4.1. Main Results

Table 1 shows that under the matched setting, Aerial2Poly outperforms Pix2Poly and competing methods on IoU, C-IoU, and the topology-aware IoU/F1/PA metrics across all three datasets. On WHU Buildings, C-IoU improves to 90.09%, yielding a $+8.46$ point absolute gain. This remarkable improvement suggests that our model recovers more geometrically aligned polygons rather than locally accurate but topologically broken outlines.

Official INRIA Test Set. Table 2 reports the pixel-level test scores returned by the INRIA online evaluator with undisclosed ground truth. Following the standard INRIA test-time protocol, we apply the model in a sliding-window fashion over each whole test image and combine the resulting pixel-level predictions into a test submission. Under this official evaluation, Aerial2Poly attains 76.88% IoU and 96.53% Accuracy, outperforming the best test results of Pix2Poly and

other alternative methods.

5. Next Steps

This work introduced Aerial2Poly, a simple and scalable formulation for direct building footprint vectorization from aerial imagery. By predicting the full-tile polygon set directly in vector space, Aerial2Poly avoids mask-to-polygon recovery, object prompts, and auxiliary connectivity heads. Coupled with Gaussian-biased decoding, recall-oriented training, and vector-space trial consolidation, our approach establishes new state-of-the-art results on the INRIA, SpaceNet Vegas, and WHU Buildings benchmarks.

While this workshop paper focuses on building footprint extraction, the broader takeaway is that full-tile polygon generation is a viable interface for aerial perception. We hope these results encourage future work on richer label spaces, larger aerial benchmarks, and more general vector-native prediction from imagery.

References

[1] 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 *WACV*, 2025. 1, 2, 4

[2] Janja Avbelj, Rupert Müller, and Richard Bamler. A Metric for Polygon Comparison and Building Extraction Evaluation. *IEEE Geoscience and Remote Sensing Letters*, 2015. 4

[3] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2Seq: A Language Modeling Framework for Object Detection. In *ICLR*, 2022. 2, 3

[4] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J. Fleet, and Geoffrey E. Hinton. A Unified Sequence Interface for Vision Tasks. In *NeurIPS*, 2022. 2, 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 3

[6] Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. SpaceNet: A Remote Sensing Dataset and Challenge Series. https://arxiv.org/abs/1807.01232, 2018. 1

[7] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. Polygonal Building Segmentation by Frame Field Learning. In *CVPR*, 2021. 2, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 3

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2

[10] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *ICLR*, 2020. 3

[11] 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*, 2019. 1

[12] Maxim Khomiakov, Michael Riis Andersen, and Jes Frellsen. GeoFormer: A Multi-Polygon Segmentation Transformer. In *BMVC*, 2024. 2

[13] Muxingzi Li, Florent Lafarge, and Renaud Marlet. Approximating Shapes in Images with Low-Complexity Polygons. In *CVPR*, 2020. 2

[14] Weijia Li, Wenqian Zhao, Huaping Zhong, Conghui He, and Dahua Lin. Joint Semantic-Geometric Learning for Polygonal Building Segmentation. In *AAAI*, 2021. 2

[15] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi. Topological Map Extraction from Overhead Images. In *ICCV*, 2019.

[16] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast Interactive Object Annotation with Curve-GCN. In *CVPR*, 2019. 2

[17] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017. 1

[18] 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*, 2023. 2, 4

[19] Bingnan Yang, Mi Zhang, Zhan Zhang, Zhili Zhang, and Xiangyun Hu. TopDiG: Class-Agnostic Topological Directional Graph Extraction from Remote Sensing Images. In *CVPR*, 2023. 2

[20] Mingming Zhang, Qingjie Liu, and Yunhong Wang. HiT: Building Mapping With Hierarchical Transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 2

[21] Kang Zhao, Jungwon Kang, Jaewook Jung, and Gunho Sohn. Building Extraction from Satellite Images Using Mask R-CNN with Building Boundary Regularization. In *CVPR Workshops*, 2018. 2

[22] Stefano Zorzi, Ksenia Bittner, and Friedrich Fraundorfer. Machine-Learned Regularization and Polygonization of Building Segmentation Masks. In *ICPR*, 2020. 2, 4

[23] Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, and Friedrich Fraundorfer. PolyWorld: Polygonal Building Extraction with Graph Neural Networks in Satellite Images. In *CVPR*, 2022. 2, 4