GAST:
Geometry-Aware Structure Transformer

Maxim Khomiakov[1,2], Michael Riis Andersen[1] & Jes Frellsen[1]
[1]Department of Applied Mathematics and Computer Science, Technical University of Denmark
[2]Otovo AS
{maxk,miri,jefr}@dtu.dk

Abstract

We present GAST, a novel model for realistic building delineation, trained using noisy, imperfect ortho imagery and designed for real-life applications. While most popular methods today rely on some form of semantic segmentation, the core interest is not the building's interior points but rather the sequence of points surrounding the outer hull, i.e., the most sparse set of points encapsulating the geometry of the building. Our method works end-to-end, removing the need for post-processing, while demonstrating generalization across large geographical differences. We compare our method to state-of-the-art, complementary works and demonstrate that our model outperforms the baselines in a variety of circumstances and across all metrics relating to polygon fidelity. We release our dataset and model checkpoints at www.huggingface.co/datasets/pihalf/ERBD

1. Introduction

Various geospatial applications rely on the analysis of imagery, often stemming from a remotely sensed medium such as satellite or aerial imagery. In analyses of urban environments, buildings are frequently of interest. Although there have been successful attempts to use remote sensing imagery to capture the geometric attributes of buildings, several issues remain. The canonical method of extracting building polygons have typically initially conducted semantic segmentation of building objects [7, 11, 22], identifying pixels that belong to the building including pixels within the building interior. The resulting semantically segmented map then serves as the basis for post-processing techniques, which may involve rule-based or learned methods [7, 22, 25], to create the final representation: a minimal set of vertices that capture the outline of the building. While working well under many circumstances, these methods seem to worsen under imperfect image conditions.

Figure 1. The aim of our study is to predict the most compact set of vertices that produce a geometrically accurate building delineation from data acquired under real-world conditions. From left to right: Ground truth, HiSup [22], GAST (ours).

Our study introduces an alternative approach to this problem, seeking to determine the extent to which we can learn the geometric polygons of building footprints directly end-to-end from noisy aerial imagery obtained in real-world conditions.

To solve this specific challenge, we present GAST, a novel model for building delineation from aerial imagery. Unlike prior studies, we trained and tested on public data sources utilising orthorectified imagery containing a common flaw: A parallax leaning effect for buildings not exactly under the camera, creating translation artifacts of annotated polygons [8]. Furthermore, we test generalization across multiple countries and conduct an extensive comparison with respect to recent state-of-the-art work [22]. Finally, we provide our data and model checkpoints to facilitate further research. In summary, the main contributions of our paper are:

- A novel model for single object building delineation
- Extensive evaluations across geospatial datasets
- Qualitative and quantitative comparison against recent works
- A curated dataset of single buildings spanning multiple countries with varying complexity

2. Background

The problem we seek to solve is learning a mapping from an image to a set of vertices that give the most sparse and geometrically accurate polygon surrounding the building. Prior studies have sought to solve this problem using semantic segmentation [4, 7, 11, 12, 22, 25] to classify each pixel in the image as to whether it is a building or not. Such techniques present some challenges when applied towards geospatial planning for buildings due to a number of reasons: The methods will often become uncertain around the edges of the building, forming 'blob'-like shapes instead of right-angled structures [23]. Less common textures that overlap the building roof (such as trees or leaves) may generally have a higher likelihood of not being assigned to the building class. Finally, and perhaps most importantly, the segmentation map is but an intermediary step towards vectorization of the building, which is among the most useful data mediums for working with geospatial data due to its scale-invariant properties, as opposed to raster graphics. There are a number of studies that aim to mitigate the issues mentioned above. Several studies have sought to reclassify pixels of higher uncertainty using methods such as conditional random fields or the Potts model [25]. Similarly, there have been attempts to use frame-field learning (FFL) [7], where the predicted segmentation map is refined according to geometric constraints. In PolyWorld [24], the segmentation map predicts the corners of the building, which are then refined by a graph neural network to form the closed graph encapsulating each building. These methods share the trait of applying semantic segmentation to the image of interest (A U-Net style architecture) following either heuristics or learned post-processing methods.

Another set of methods relies on the prediction of a sequence of points directly. This skips the rasterization towards vectorization process otherwise required by the prior methods above. For this sets of models, popular works include PolyMapper [13] and Polygon-RNN++ [1], the study by Zhao et. al [23] as well as a more recent studies known as Polygonizer [10] and PolyFormer [15]. Common to these methods is the autoregressive formulation of the polygon sequence prediction. Most of these studies utilize the same formulation first devised by Accuna et al. [1], where a single object vectorization is being modeled as a set of conditional probabilities $p(x_t|I, x_0, x_{t-1}, x_{t-2})$, where x_t represents the t'th polygon coordinates in the form of spatial tokens. Each spatial token is auto-regressively inferred with a convolutional LSTM [18], depending on the input image I, an initial starting point x_0 and the two prior values x_{t-1}, x_{t-2}. Another recent work introduces the Polygonizer [10], where the authors demonstrate the ability to solve a vectorization problem depending only on the prior predicted tokens and an LSTM with attention for decoding. This involves learning a conditional distribution of $p(x_t|I, x_{<t})$ where $(x_t)_{t\in\mathbb{N}_0}$ forms the sequence of the building polygon and I represents the image. Finally, former works exist predicting the polygon directly, but without the use of probabilistic auto-regressive decoding. These include PolyBuilding [9], which extends the Deform-DETR object detection model to predict both the building polygon and corner coordinates. A complementary study of the HEAT transformer model [5], which also resembles our work, albeit with two major differences: We assume the building polygon can be drawn with one sequence (i.e., we do not allow for inner polygons); similarly, our model is auto-regressive, which voids the necessity to fix the length of the polygon a priori. Similarly PolyFormer [15] proposes to solve general segmentation problems using a transformer decoder which regresses over 2D-coordinate pairs auto-regressively. PolyFormer has multiple output heads of which one predicts the object class, while another predicts the polygon coordinate pairs. While this work has come to our attention upon the writing this paper, we note that the PolyFormer is a multi-modal model relying on a text embedding for each image, which provides additional information to the decoder than is otherwise the case in our setting. In summary, our work takes its queues from the Polygonizer [10] as well as the HEAT transformer [5], with a particular emphasis on residential geospatial inference from 'in-the-wild'-sourced data.

3. Methods

Our model consists of two components, an image-based encoder $\mathcal{F}_{\text{enc}}(I)$ which transforms the input image I into a set of patches, which are then flattened and parsed in a sequence. The output of $\mathcal{F}_{\text{enc}}(I)$, the intermediary feature representation $\mathcal{I}$, is then forwarded to a geometric decoder $\mathcal{F}_{\text{dec}}(\mathbf{x}, \mathcal{I})$, which learns to associate the sequence spatial tokens $\mathbf{x}$ to the respective patch on the image I, as shown in Figure 2. Formally we combine a Vision Transformer [6] with an auto-regressive decoder transformer [21], which learns to predict the sequence of points on the image in an auto-regressive manner. Thus, we assume that we can parameterize our problem as a function of conditional probabilities of the polygon coordinates $\mathbf{x}$ given an image I:

$$p(\mathbf{x}|I) = p(x_0|I)p(x_1|I, x_0)p(x_2|I, x_1, x_0)\dots \quad (1)$$

$$= \prod_{t=0}^{N_x} p(x_t|I, x_{<t}) \quad (2)$$

Here $\mathbf{x}$ consists of a flattened sequence of tokens containing the image indices of the polygon in counterclockwise order, i.e., $\mathbf{x} = (\text{S}, x_1, y_1, x_2, y_2, ..., x_n, y_n, \neg\text{S})$ and N_x is the number of tokens in the sequence. S and $\neg$S are two special tokens, the first representing the start of a sequence, and the latter representing the end of the sequence. Our likelihood of the individual tokens, $p(x_t|I, x_{<t})$, is a categorical distribution over the indices of the input image plus

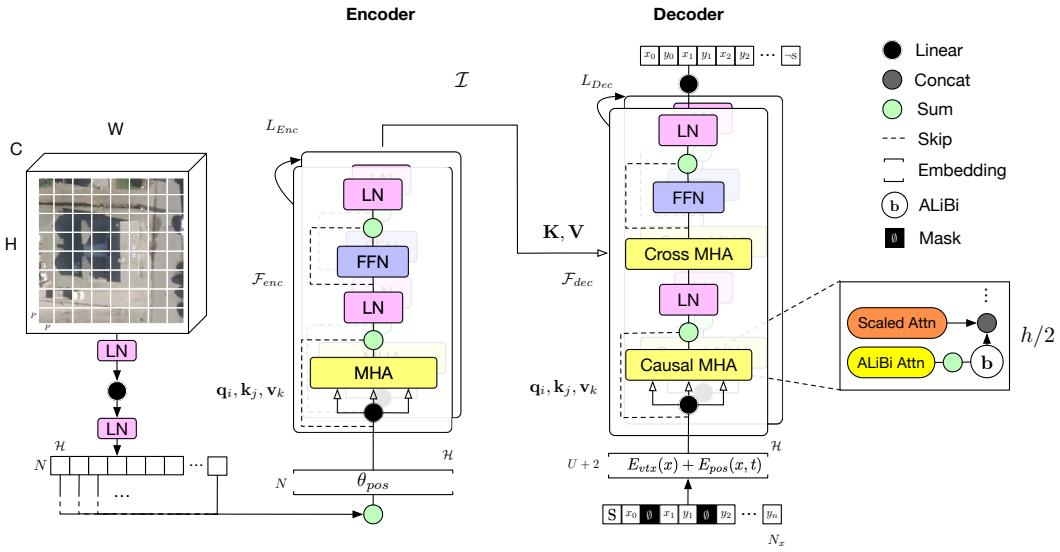

Figure 2. GAST Model architecture. From the left hand side, we produce an image embedding which acts as context for the auto-regressive decoder in the Cross Multi-Head Attention block. For each of the attention blocks in the Decoder we dedicate half of the attention heads to short-range tokens (ALiBi) and the other half using canonical scaled dot-product attention. During training we mask the tokens with a random proportion between [0;.35]

the two special tokens coding for the start and the end of the sequence. We optimize the parameters of our model θ by minimizing the negative log-likelihood of the model parameters, which can be expressed using the loss function $\mathcal{L}$:

$$\mathcal{L} = -\sum_{t=0}^{N_x} \log p_\theta(x_t = x_t^* | x_{<t}, \mathcal{I}), \tag{3}$$

where x_t^* is the true value at timestep t, N_x is the number of tokens in the sequence and $p(x_t|x_{<t}, \mathcal{I})$ are the predicted parameters of a categorical distribution defined over the image indices plus start and end tokens. Note that all our coordinates in the study are discretized.

Encoder The ViT encoder [6] takes an image of $I \in \mathbb{R}^{H\times W\times C}$ and splits it into patches, $I_p \in \mathbb{R}^{N\times(P^2\times C)}$ ordered from top left to bottom right, where $N = HW/P^2$ is the number of patches, each with dimensions $P \times P \times C$ of which C is the number of channels, and P is the size of each patch while H and W are the height and width dimensions of the input image. It then performs an image embedding for each of the N image patches of dimension $P \times P \times C$, by means of a learnable linear projection to match the hidden dimension $\mathcal{H}$ of the multiheaded self-attention layer (MHA) of the transformer encoder, effectively making N the sequence length of each patch to the transformer encoder. Prior and post projection of the image, we apply the LayerNorm (LN) elementwise [3], for positional encoding we learn a vector $\theta_{\text{pos}} \in \mathbb{R}^{N\times\mathcal{H}}$, which is added to each embedded image before input into the encoder block layers $\mathcal{F}_{\text{enc}}^{(l)}$, where $\mathcal{H}$ is the embedding dimension. Following this, a standard set of transformer encoder blocks entail [21].

Following L_{enc} layers of transformer encoder blocks, the image context is formed, producing a tensor of dimensions $\mathbf{B} \times N \times \mathcal{H}$ where N is the number of the image patches, $\mathbf{B}$ is the batch dimension and $\mathcal{H}$ is the hidden dimensions of the encoder $\mathcal{F}_{\text{enc}}$. These form the keys and values for the multi-head cross-attention in the Decoder.

Decoder The decoder $\mathcal{F}_{\text{dec}}$ is a transformer decoder that follows the original [21] for the most part, although with some key variations that we found to be beneficial in our environment. We apply two sets of learned embeddings to our input sequence $\mathbf{x}$. $E_{\text{pos}}(\mathbf{x})$ which applies a positional embedding defined in $\mathbb{R}^{x_{\max}\times\mathcal{H}}$ where $x_{\max}$ is the maximum sequence length, and $E_{vtx}(\mathbf{x}) \in \mathbb{R}^{U\times E_h}$ which applies a token-based embedding for U unique tokens defined over the image indices plus two special tokens. These embeddings are added to form a tensor of dimensions $\mathbf{B}\times N_x\times\mathcal{H}$, where N_x is the sequence length of the input sequence, $\mathbf{B}$ is the batch dimension and $\mathcal{H}$ are the hidden dimensions of the transformer decoder $\mathcal{F}_{\text{dec}}$.

When designing the transformer decoder, we wanted to introduce inductive biases for the model to attend nearby geometric points, as well as making sure we also attend back towards the very first part of the sequence to make sure we learn to close the polygons. For this, we have included ALiBi [17] which biases

queries q_i and keys k_j so that values in close proximity will receive a higher attention score than tokens further apart. We follow the original implementation, where we add a different static value to each attention head as $\text{softmax}\left(\mathbf{q}_i\mathbf{K}^\top + m \cdot [-(i-1), \ldots, -2, -1, 0]\right)$ where m is a scalar slope defined for each attention head, $\mathbf{q_i}$ is a query $\in \mathbb{R}^{1\times d}$ for the timestep i in the sequence of length $L \in (1 \leq i \leq L)$, and $\mathbf{K}^\top \in \mathbb{R}^{i\times d}$ are the set of keys from the first to the ith timestep, while d is the embedding dimension. The geometric sequence for n attention heads is defined as starting in $2^{\frac{-8}{n}}$, using that same value as its ratio. As previously mentioned, we do not only want our model to focus on nearby tokens, but also on more distant parts of the sequence. Therefore we dedicate half of our model $\mathcal{F}_{\text{dec}}$ attention heads towards ALiBi [17] biased attention heads, whilst the rest may attend as normal in a canonical causal attention way.

In addition to the ALiBi positional embeddings, which is a relative embedding that enforces local dependencies, we also use rotary positional embeddings [19] (RoPE). The rotary positional embeddings are applied by multiplication prior the attention operations in the queries and keys, where the sequences of the embedded tokens $\tilde{\mathbf{x}} = E_{\text{pos}}(\mathbf{x}) + E_{vtx}(\mathbf{x})$ are each rotated at an angle using a rotation matrix, relative to the position of the token in the sequence. We have found that the incorporation of RoPE embeddings improve the model performance in our setting, while not necessarily a conclusive explanation, we argue it is related to the inductive bias of performing rotations of a representation inherent to the spatial tokens that exist in two dimensions, further biasing the model to learn the angles present between the spatial tokens themselves.

Following the embeddings, our transformer decoder follows the traditional transformer decoder blocks, where we first perform causal multi-head attention (Causal MHA) using queries, keys, and values from our target sequence and a causal lower triangular mask. Following, a second multi-head attention block performs self-attention into the image context (Cross MHA), where the queries represent the interim feature representation of the target sequence, and the keys and values are the outputted image context representation from the transformer encoder $\mathcal{F}_{\text{enc}}$. At the final layer of the decoder, we perform a standard linear projection to the categorical distribution dimensions, forming the logits of our outputted categorical distribution over the spatial tokens.

4. Experimental setup

The images in our datasets are scaled to 224x224x3. For all our subsequent results, our model has the following hyperparameters: each patch has the size of 32x32x3, the hidden dimension of the encoder attention is 512 with 8 attention heads and 6 layers, the feedforward layer has a size of 2048, and for the activation function of the encoder, we use the GELU. The Decoder consists of 12 layers with 16 attention heads in each, the hidden dimension of the attention is 512 and the feedforward layer is 2048. We set a maximum sequence length of 400 and project our tokens using 227 unique learned embeddings with an embedding layer dropout of 0.1. The hidden dimension of the attention is 512 with 16 attention heads, of which 8 are using the ALiBi (added linear bias) [17] and 8 without. We apply a dropout of 0.1 following the attention layer and a 0.2 dropout on the fully-connected layer in each of the decoder transformer blocks. For the decoder we employ token masking, which means that we mask at random up to 35% of the tokens in a sequence at each training step. Our model in this configuration consists of 115 million trained parameters. We use the AdamW optimizer for training the model with a learning rate $3 \cdot 10^{-4}$, $\beta = (0.9, 0.999)$ and a weight decay of $1 \cdot 10^{-2}$. For all training purposes we employ early stopping, in which a consistent rise in the validation loss for more than two epochs, results in the conclusion of the training. For inference we utilize the trained model with the lowest validation loss.

C	N	$\mathbf{I}_\Delta$	$\mathbf{T}_\mu$	$\mathbf{T}_\sigma$	$\mathbf{T}_{\text{min}}$	$\mathbf{T}_{\text{max}}$
NL	89,362	.25m	12	6	5	153
DK	119,956	.1m	10	4	4	61

Table 1. Summary statistics of the European Residential Building Dataset (ERBD) sampled from the Netherlands and Denmark. N represents the number of unique instances, I_Δ is the spatial resolution of the image, and $T_{(\mu,\sigma,\text{min,max})}$ represents summary statistics of the polygon lengths in the dataset being the mean, standard deviation, minima and maxima.

4.1. The European Residential Building Dataset (ERBD)

Part of our contributions to this study relate to the collection and distribution of our own curated dataset dubbed the European Residential Building Dataset (ERBD). To motivate the utility of our dataset, we would like to argue the limited number of alternative datasets which are 1) providing annotated polygons (not raster masks), 2) is focused on single detached residential homes, and 3) span multiple countries.

We collected datasets from two different countries, with a particular emphasis on acquiring detached houses. This is motivated by two reasons. First being detached houses, it increases the chance that there is but one object in the image, thus focusing our efforts on the finer details of the building in question. Second, in a practical application scenario, it is possible to request the user for a bounding box, or extrapolate a bounding box from the click of a user in-

teraction. All datasets share the properties of begin very high-resolution orthorectified imagery, with a ground sampling distance ranging between 10-25cm per pixel. The annotation methods across the data vary, as some objects are represented with the minimal representation of points per object, and some are annotated with multiple points along the linear segments of the building. The dataset and the annotation also vary in quality, representing noise, which is bound to be encountered when working with realistic geospatial data. These artifacts include: translation error, parallax, reflections, and occlusion effects. We believe the ability to demonstrate models working reliably across large geographical distances and in spite of common faced artifacts is important to advance the field. We present a general overview and summary statistics in Table 1 and refer to the supplementary material for additional information.

Denmark We collected our data based on a cross section of multiple sets of dataset from the danish national cadastral registers. For this study, we have sampled 119,956 images from Denmark (uniformly), with a particular emphasis on buildings classified as detached homes. The dataset is split into an 68/12/20% train,validation, and test split consecutively.

Netherlands dataset was chosen as the second country for the dataset. The Dutch dataset is split similarly to the Danish with 65/15/25% train, validation, and test splits consecutively. For additional details on how the examples from each country were collected, we refer to the supplementary material.

5. Evaluation metrics

We evaluate our model applying the standard MSCOCO detection metrics as commonly used in our setting [9, 10, 24], in addition to four metrics which have become increasingly popular in building delineation problems. Complexity-aware IoU introduced by Zorzi et. al [24] which is defined as the Intersection-over-Union weighted by the number of vertices in each polygon, the maximum tangent angle error (MTA) introduced by Girard et. al [7] and PoLiS the polygon distance measure introduced by Avbelj et. al. [2]. For the MSCOCO-metrics [14] we measure the performance on the segmentation aspect of our predictions w.r.t. ground truth, this implies a conversion from the predicted polygon of our method towards a binary mask, and computing IoU-scores. The most important metrics that we measure are: mAP, AP_{50}, AP_{75}, mAR, AR_{50}, AP_{75}. Mean average precision is given as the average precision for all objects where IoU is at `[0.5,0.55,0.60,...,1]` levels of overlap. Similarly for AP_{50}, a prediction counts as a true positive in all instances where IoU> 0.5, while for AP_{75} we count observations with IoU > 0.75. The mean average recall mAR is computed with threshold levels evaluated at `[0.0,0.01,0.02,...,1]`, while AR_{50} and AR_{75} represent recall at IoU thresholds of .50 and .75 respectively. We do not take into account precision or recall relative to object size, as our dataset is designed with a fairly similar sized objects. All COCO-measures are related to the segmentation aspect of our result, which are based on a raster of our vectorized output, while the additional metrics chosen are more strongly linked to the actual model output and intended usecase.

5.1. The principal polygon orientation difference

In addition to these measures, we would like to introduce an additional metric that would present an additional measure of the performance of the inference conducted. It is a fairly simple measure, which allows us to gauge to what extent the predicted polygon orientation matches the orientation of the ground truth. Calculating the metric is straightforward: For any polygon $P \in \mathbb{R}^{Nx2}$ compute the eigen vectors e_P of Cov(P) and select the one with the highest eigen value λ_P, then we can compute the principal orientation taking the $\arctan2(e_y, e_x)$. The orientation difference is then $\Delta_\theta = (\theta_{GT} - \theta_P + 180°) \mod 360° - 180°$

6. Results

The results are presented with inference on the *test*-set examples in each of the provided datasets. We initialize with S-tokens and continue inference until the stop tokens $\neg$S or the maximum length of 400 tokens is reached. The outputted sequence is then parsed according to the evaluation metrics described above.

6.1. Performance on the ERBD dataset

6.2. Qualitative evaluation

In Figure 3 we observe the qualitative visualization of our method compared to the HiSup [22] model. From the columns left to right, we present the ground truth image, the image with annotations, the HiSup [22], and our model predictions. On the row levels we present two rows by dataset, the Danish, Dutch, and combined test splits. All models seem to capture the object in the image to a fair extent, with some notable differences: we notice upon higher model uncertainty, it seems that the HiSup [22] model tends to output many more points than both the ground truth and our method. Meanwhile, on occasions with high visibility and less noisy environments, both models perform quite similarly (see the first two rows). Meanwhile, however, in instances of off-nadir/nadir imagery, GAST performs better than HiSup [22], and generally leads to more visually appeasing results. This is also what seems to be generally occurring across the different OOD dataset inferences; we refer to the supplementary material for additional examples of this.

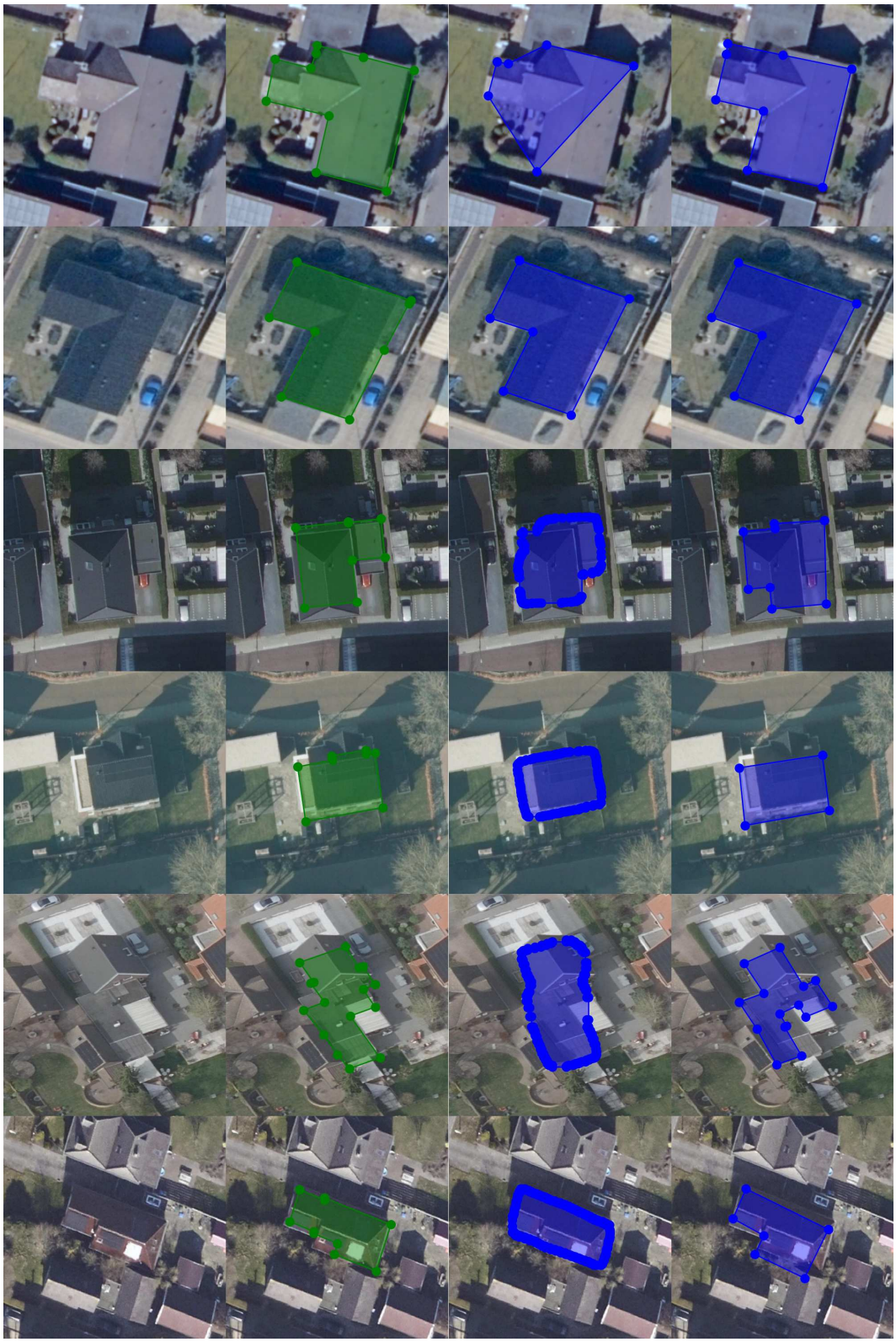

Figure 3. Qualitative results from left to right: Ground truth, ground truth with annotation, HiSup prediction, GAST (ours) prediction. From the top row down, for each two rows the figures correspond to models trained and performing inference on the Danish, Dutch and Danish and Dutch datasets combined.

	AP↑	AP_{50}↑	AP_{75}↑	AR↑	AR_{50}↑	AP_{75}↑	IoU↑	C-IoU↑	MTA↓	PoLiS↓	$\Delta\theta$	N-ratio	Θ_{Model}	Dataset
HiSup	**68.39**	89.28	**75.27**	**71.94**	91.72	78.73	**82.4**	52.29	57.87	13.32	26.45	2.84	DK	DK
	44.83	73.79	**47.3**	**53.87**	**84.85**	**58.44**	**70.97**	4.33	58.44	39.42	-12.85	42.78	NL	NL
	59.62	83.91	**67.59**	67.97	91.64	77.51	**80.1**	4.31	59.01	42.16	-18.44	69.97	NL+DK	NL+DK
	6.97	35.94	0.95	12.33	53.54	1.76	50.96	2.06	57.3	54.6	-18.74	37.98	DK	NL
	18.7	**49.21**	**10.25**	**31.33**	**68.72**	**23.96**	**58.48**	7.86	58.44	25.75	-12.83	15.49	NL	DK
	70.87	90.73	**78.77**	**73.95**	92.98	81.12	**83.5**	55.24	57.87	12.45	25.85	2.04	NL+DK	DK
	55.06	**87.07**	**62.76**	**61.49**	**92.74**	**71.6**	**76.85**	3.94	59.01	35.39	-16.1	50.95	NL+DK	NL
GAST	58.16	**96.45**	65.12	69.77	**99.19**	**81.93**	81.72	**58.53**	**55.92**	**12.32**	**-6.24**	**0.81**	DK	DK
	34.59	**82.11**	16.78	24.23	78.72	7.00	65.69	**58.76**	**32.41**	**13.67**	**-7.18**	**2.32**	NL	NL
	45.15	**89.29**	38.00	**69.45**	**99.47**	**81.42**	76.98	**58.38**	**47.78**	**12.44**	**-9.64**	**1.55**	NL+DK	NL+DK
	12.50	**52.60**	**1.49**	**52.94**	**97.96**	**50.44**	**52.53**	**59.17**	**20.29**	**23.07**	**-7.61**	**0.80**	DK	NL
	11.30	45.54	1.37	22.50	67.66	9.93	50.83	**46.72**	**19.99**	**24.09**	**-2.49**	**2.40**	NL	DK
	58.34	**95.17**	65.31	70.32	**99.15**	**82.48**	81.74	**58.18**	**55.45**	**12.21**	**-10.83**	**0.88**	NL+DK	DK
	37.25	84.97	20.73	40.87	89.21	31.20	67.58	**58.78**	**34.30**	**13.26**	**-10.40**	**1.35**	NL+DK	NL

Table 2. Computed benchmark results across all ERBD dataset splits for both the HiSup [22] model and GAST (ours).

6.3. Quantitative metrics

In Table 2 we present the main results of our paper and performance metrics in a variety of training and inference instances, compared to the HiSup [22] model. The metrics computed correspond to the evaluation metrics described in Section 5 with a few minor additions: N-ratio measures cardinality of the predicted set versus the cardinality of ground truth, we want this to be as close to 1 as possible. Θ_{Model} represents the split of the data set on which the model was trained, while the column of the data set represents the test set for inference. In the final column, we measure the inference speed per example (in seconds).

6.3.1 The Danish ERBD set

In summary, both methods perform very well on the Danish ERBD split, in particular we note that HiSup outperforms on the AP, AP_{75} and the AR to some extent. While also marginally performing better on the IoU score, the average precision is where the HiSup model beats our method in this instance. Meanwhile, however, the average recall on the higher thresholds of AP_{50} and AP_{75} our model performs better. Additionally, on the metrics that are more closely related to the building coordinates, we exceed the performance for the Danish dataset to a minor extent.

6.3.2 The Dutch ERBD set

Somewhat expectedly, we notice a substantially lower performance when training and evaluating the Dutch dataset. Similarly across the models, we notice larger differences than what was the case in the prior scenario. The models perform similarly at the .50 threshold for both the precision and recall, but at the higher levels we see a clear lead towards the HiSup model. However, when we start comparing geometry specific attributes, we are seeing a similar pattern as before - GAST shows substantial improvement on the C-IoU,MTA,PoLiS, and POD measures.

6.3.3 Combining the ERBD dataset

In Table 2 we present results for situation in which either GAST or HiSup was trained on a combined set of the ERBD dataset and performing inference on either a combine test set of the two countries, or each of the respective countries one by one. For this experiment, we wanted to assess whether there was any indication of model learning which would transfer across the datasets. The HiSup model sees an improvement when performing inference across different countries. In fact, the combined model, perform better when doing inference on the NL test set, than when solely trained on the NL itself, indicating the model learned transferable features. The same holds true for GAST, while slightly worse in general for the NL+DK versus DK split, the NL+DK versus the NL dataset performs much better than if trained on NL alone. When comparing the two models against each other, we are contiuing to see a clear benefit towards using GAST if interested in vector-based inference, while for the segmentation metrics HiSup is performing better.

6.4. Country to country

A good model should ideally generalize and, in our setting, this is certainly paramount for any deployment in a practical setting. Therefore, we present results for inference performed on a different test-set than what the model was trained on. We refer to these as Out-of-distribution inference (OOD). Starting with the model trained on DK, perform inference on NL. In this situation, we are performing inference on a more complicated dataset with a model trained on a more simple one. Therefore, the drop in performance across both methods is to be expected. However, in this instance, we see a clear indication of better perfor-

RoPE	Mask	ALiBi	mAP	IoU	PoLiS
	✓		14.59	37.8	22.01
			22.62	45.12	19.76
✓			25.44	47.51	18.58
✓	✓		27.37	52.46	18.64
		✓	31.22	53.89	17.61
	✓	✓	33.52	56.44	16.56
✓		✓	35.17	60.55	15.10
✓	✓	✓	35.73	59.29	15.35

Table 3. Results produced when trained on a subset of the NL training set, performed doing inference on a subset of the NL test set.

mance by GAST relative to the [22] model, which is not only on the geometry-based metrics but also on the raster segmentation measures. In comparison, when going from NL to DK, the HiSup model again shows its strengths when it comes to metrics that are directly linked to the semantic segmentation raster.

7. Model ablation studies

To assess whether our chosen components in our model are contributing towards a better, or worse performance, we train and compute a number of models on a set of metrics of relevance, focusing on mAP, IoU and the PoLiS distance measure. In Table 3 we present our findings from training GAST on a smaller subset of the NL training dataset using 10.000 random samples, while doing early stopping as a function of the full validation set. The metrics below are given by test-set inference on the NL dataset for 210 samples. We find that ALiBi is a significant component towards the stable training of the model, while RoPE and Masking does indeed contribute towards a better model, relying solely on RoPE and ALiBi in addition to our existing learned embeddings, gives us the best results.

8. Discussion

GAST is a deep generative model with limited geometric constraints, which in our setup means we attempt to directly learn the patterns of the data through the model itself. We argue the benefit of this approach is the improved ability for generalization. Additionally, modeling probabilities auto-regressively allows for more diverse inference samples, while providing uncertainties in a deployment scenario, thus improving the applicability of the model in a production environment. The weakness of this approach however, it is computationally more expensive and requires larger amounts of data to achieve good performance compared to alternative models such as [5, 22]. We are, however, more confident in our thesis that methods relying on semantic segmentation in a geospatial context are generally more prone to artifacts stemming from this representation, as opposed to directly modeling the geometric tokens, as is the case for [5, 9, 10].

8.1. Limitations & Future work

Our model is unable to process more than one object in a scene, as it relies on the assumption that only one sequence exists. This includes objects with holes or multiple parts. Similarly, due to the auto-regressive nature of our model, we require multiple forward passes at inference time to produce a prediction. This is certainly a constraint, but as we argue above, it is also a possibility towards a more informed inference sample. In this work, we have only presented *greedy*-inference, while improvements such as beam-seach or nucleus sampling could improve the inferred results.

For future work, we would like to look into parameterisations that can handle multiple objects in a scene, while also considering model improvements that will allow for stronger inductive biases in the geometric representations of the model.

9. Conclusion

In this paper we have presented GAST, a novel model towards the learning of building footprint polygons end-to-end of residential homes. We show that our model learns to produce vectorized building polygons when applied to realistic real-world data, while still having ability to improve. We conduct an extensive comparison towards state-of-the-art model HiSup [22], and demonstrate superior performance in all metrics relating to polygon fidelity.

10. Acknowledgements

This work is supported by Otovo AS. We acknowledge EuroHPC Joint Undertaking for awarding us access to Karolina at IT4Innovations, Czech Republic.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018. 2

[2] 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

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[4] Ksenia Bittner, Fathalrahman Adam, Shiyong Cui, Marco Körner, and Peter Reinartz. Building Footprint Extraction

From VHR Remote Sensing Images Combined With Normalized DSMs Using Fused Fully Convolutional Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(8):2615–2629, 2018. 2

[5] Jiacheng Chen, Yiming Qian, and Yasutaka Furukawa. Heat: Holistic edge attention transformer for structured reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3866–3875, 2022. 2, 8

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[7] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. Polygonal Building Segmentation by Frame Field Learning. pages 1–30, 2020. 1, 2, 5

[8] Yong Hu, David Stanley, and Yubin Xin. True ortho generation of urban area using high resolution aerial photos. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:3–10, 2016. 1, 10

[9] Yuan Hu, Zhibin Wang, Zhou Huang, and Yu Liu. Polybuilding: Polygon transformer for end-to-end building extraction. *arXiv preprint arXiv:2211.01589*, 2022. 2, 5, 8

[10] Maxim Khomiakov, Michael Riis Andersen, and Jes Frellsen. Polygonizer: An auto-regressive building delineator. *arXiv preprint arXiv:2304.04048*, 2023. 2, 5, 8

[11] Weijia Li, Wenqian Zhao, Jinhua Yu, Juepeng Zheng, Conghui He, Haohuan Fu, and Dahua Lin. Joint semantic–geometric learning for polygonal building segmentation from high-resolution remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 201:26–37, 2023. 1, 2

[12] Weijia Li, Wenqian Zhao, Huaping Zhong, Conghui He, and Dahua Lin. Joint semantic-geometric learning for polygonal building segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1958–1965, 2021. 2

[13] Zuoyue Li, Jan DIrk Wegner, and Aurelien Lucchi. Topological map extraction from overhead images. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:1715–1724, 2019. 2

[14] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[15] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. Polyformer: Referring image segmentation as sequential polygon generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18653–18663, 2023. 2

[16] 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. 10

[17] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021. 3, 4

[18] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 2

[19] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 4

[20] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018. 10

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[22] 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. 1, 2, 5, 7, 8

[23] Wufan Zhao, Claudio Persello, and Alfred Stein. Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework. *ISPRS journal of photogrammetry and remote sensing*, 175:119–131, 2021. 2

[24] 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. 2, 5

[25] Stefano Zorzi, Ksenia Bittner, and Friedrich Fraundorfer. Machine-learned Regularization and Polygonization of Building Segmentation Masks. 2020. 1, 2