OpenCity3D: What do Vision-Language Models know about Urban Environments?

Valentin Bieri[1], Marco Zamboni[1], Nicolas S. Blumer[1,2], Qingxuan Chen[1,2], and Francis Engelmann[1,3]

[1]ETH Zürich, [2]University of Zurich, [3]Stanford University

https://opencity3d.github.io

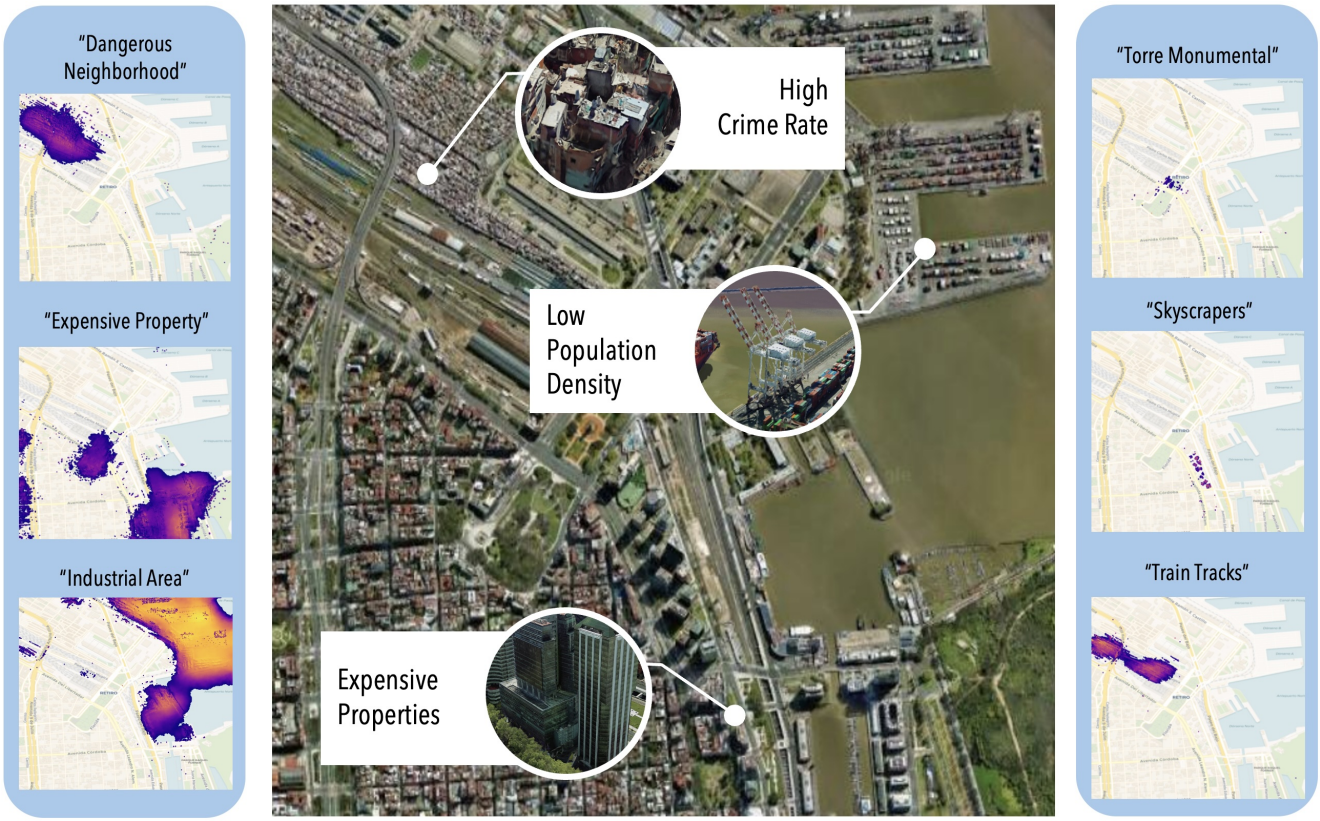


Figure 1. OpenCity3D enables zero-shot urban 3D scene understanding including higher-level understanding such as crime rate, population density, housing prices or local landmarks. For each text prompt, we visualize the response heat map from higher *(yellow)* to lower *(blue)*.

Abstract

The rise of 2D vision-language models (VLMs) has enabled new possibilities for language-driven 3D scene understanding tasks. Existing works focus on indoor scenes, or autonomous driving scenarios and typically validate against a pre-defined set of semantic object classes. In this work, we analyze the capabilities of vision-language models for large-scale urban 3D scene understanding, and propose new applications of VLMs that directly operate on aerial 3D reconstructions of cities. In particular, we address higher-level 3D scene understanding tasks such as population density, building age, property prices, crime rate, and noise pollution. Our analysis reveals surprising zero-shot and few-shot performance of VLMs in urban environments.

1. Introduction

Recent developments in 3D scene representation, including Vision-Language Models (VLM) like CLIP [22] and SIGLIP [28], Neural Radiance Fields (NERF) [15], and Gaussian Splatting (GS) [12], have significantly advanced open-set inference capabilities. Impressive results are obtained with methods such as OpenScene [18], OpenMask3D [27], LangSplat [21], and LERF [13]. These methods have predominately been evaluated in indoor spaces or autonomous driving scenarios. In this work, we analyze, for the first time, their applicability to large-scale urban 3D scenes at city scale.

The city scale introduces unique challenges due to its scale and diverse nature, that render existing methods less effective. Many dense reconstruction techniques are too ex-

pensive to run on such a scale. Understanding urban variables – ranging from the age of buildings to population density and crime rates – is crucial for urban planning and development. Despite their cost, these methods have potential to offer valuable insights into urban scene understanding, providing a foundation for improving urban living conditions and sustainability.

In this study, we extend these methodologies to operate effectively at the city scale. We introduce OpenCity as our approach that involves utilizing mesh data and generating a language-features enriched point cloud using rendered RGB-D images, inspired by the feature extraction procedures of LangSplat [21]. By leveraging language encoders, we query this language-features enriched point cloud to analyze the information content of features related to tangible urban objects such as buildings and multifaceted urban properties like population density and crime rates.

Our findings indicate promising results concerning the understanding of urban inventory, particularly for identifying building ages, housing prices, and population density. While initial findings for crime rate and noise emission prediction are less robust, our methodology demonstrates the potential for comprehensive urban analysis and planning.

This work details our methodology, findings, and implications for advancing 3D scene understanding on a city scale, offering insights into leveraging advanced computational methods for urban research and development.

2. Related Work

Large-scale 3D Reconstructions. Point clouds, Meshes, Occupancy ImpliCity [26] polish

Language-augmented 3D Scenes. Several recent studies have explored advanced techniques in 3D scene understanding and instance segmentation. Peng *et al.* [18] introduced a method that assigns per-point features to point clouds, followed by a multi-view feature fusion using CLIP [22] features. Their approach, OpenScene, supports open-vocabulary queries but faces challenges in achieving sharp segmentation masks.

OpenMask3D [27] is designed specifically for open-vocabulary 3D *instance* segmentation. It leverages CLIP embeddings to extend Mask3D [24], a model for 3D semantic instance segmentation, on an open vocabulary. To do so, it uses SAM [14] masks from posed RGB-D images of the scene to obtain CLIP embeddings that are then assigned to Mask3D masks in 3D space, embeddings that can be then compared to the ones from open vocabulary queries. A significant strength of OpenMask3D is the fact that it reasons at the mask level (instead of point-wise) which significantly improves efficiency and storage usage, both important factors for scalability. This would make it a perfectly suited candidate for large urban scene representations. However, it relies on the Mask3D segmentation model, which is trained on indoor scenes, and thus does not generalize to large city scenes. Alternative segmentation models such as Segment3D [9] did not remedy the situation, which is likely due to the indoor training data of both methods.

Kerr *et al.* [13] propose LERF, Language Embedded Radiance Fields, which builds on NeRF [15]. LERF integrates language features by learning a language field from 2D CLIP features analogously to how NeRFs learn color fields. This representation enables querying and rendering arbitrary user queries.

LangSplat [21], on the other hand, combines 3D Gaussian Splatting with language features extracted using Segment Anything Model (SAM) [14] techniques. This hybrid approach hierarchically crops parts of images and feeds them into CLIP, compressing resulting VLM features with an auto-encoder. The efficacy of LangSplat lies in its optimization of language features through rendered comparisons with CLIP features. Scaling Gaussian Splatting to large urban scenes is still an active field of research. Additionally, LangSplat relies on VLM feature compression due to memory constraints, which can lead to reduced open-vocabulary capabilities. In this work, we do not require such constraints. [26]

In our approach, we adapt LangSplat's hierarchical feature extraction with OpenScene's point cloud-based scene representation. Using a sparse point cloud instead of Gaussian Splatting enables us to analyze the full, uncompressed VLM features at the cost of less accurate geometry which is less relevant for higher-level urban queries as described below. We furthermore experiment with SigLIP [28] as a replacement for CLIP as a VLM backbone. SigLIP is a modification of CLIP, utilizing a Sigmoid loss instead of a softmax for pairwise language-image pre-training.

3. Method

3.1. City Scene Pre-processing

An illustration of the pipeline is shown in Fig. 2. The goal is to produce a point cloud with VLM features for each vertex. Input is a 3D mesh obtained from geospatial data [1]. To that end, we first render RGB color and depth images of the 3D city mesh from multiple viewpoints, ranging from satellite-like images to street-level images.

For each rendered image, we apply SAM [14] to extract segments at 4 hierarchy levels (see Fig. 3). To obtain VLM features of each segment, we then crop the image around each segment and we highlight the segmented area (see Fig. 4). Highlighting is performed by reducing the opacity of the non-segmented area and marking the border of the segment with a red line. In experiments (Tab. 2), we found that this approach improves over existing methods

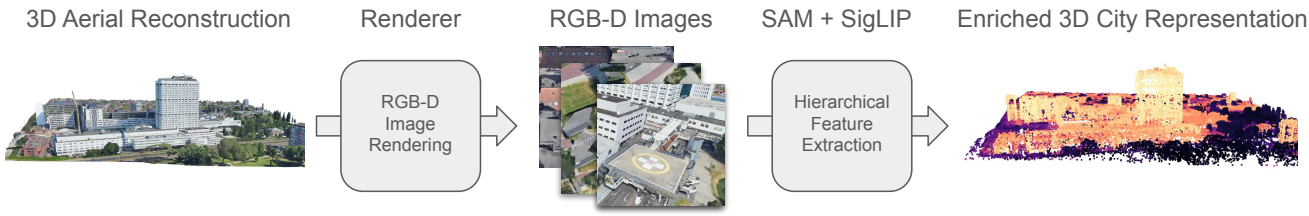
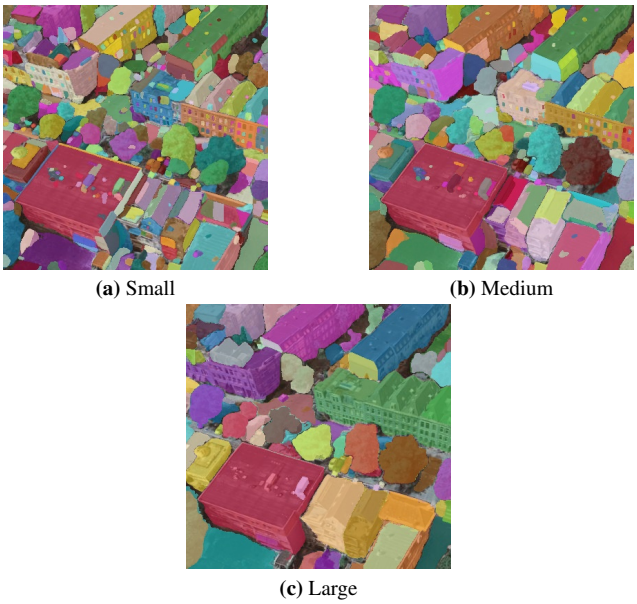


Figure 2. Illustration of the OpenCity pipeline. We first render multi-perspective images from aereal 3D reconstructions, then compute pixel-wise hierarchical visual-language features, which are projected back to the 3D mesh for searching the scene with language.



(a) Small **(b)** Medium **(c)** Large

Figure 3. Examples SAM hierarchies. In 3a segments are small and separate finer details; in 3c segments are big and merge the details.how many hierarchies do you use? 2 or 3?

such as [21] which completely remove the background and thus ignore relevant background context. Finally, we run each highlighted segment image through SigLIP [28] to obtain VLM features per mask. We also add the VLM feature of the entire image as a 5-th hierarchy level.

Figure 4. Example of a highlighted street segment. Methods that remove the background make the street almost unrecognizable.

The resulting pipeline takes about 48h on an NVIDIA 4090 graphic card for a scene with 10k images. Fortunately, the structure allows the hierarchies to be processed separately. We trade off speed against accuracy throughout this work by at times only processing certain levels.

Next, to project the 2D features to 3D points, we average the per-pixel features of all segments in which the relevant point was observed. This results in a point cloud where each point has a SigLIP feature attached, which we use for prompt-based interaction.

3.2. Language-guided Zero-Shot Scene Interaction

Based on this enriched point cloud of a city, we construct a prompt-based estimator for prediction tasks such as building age or property value.

Given a text prompt, we use the SigLIP text encoder to produce the embedding ϕ_{query}. Then we compare it to the point features ϕ^l_{point} to infer point-wise similarity scores. That is, we follow [13, 21] in considering the maximal cosine similarity across the levels between the point and the prompt embedding, normalized with a set of one or more negative queries:

$$\overline{\text{sim}}_{\text{query, point}} = \max_{l \in \text{Levels}} \exp(\phi^T_{\text{query}} \phi^l_{\text{point}}) \tag{1}$$

$$\text{sim}_{\text{query, point}} = \frac{\overline{\text{sim}}_{\text{query, point}}}{\overline{\text{sim}}_{\text{query, point}} + \sum_{n \in \text{Negatives}} \overline{\text{sim}}_{\text{n, point}}} \tag{2}$$

Similarity scores can be visualized as a 3D heatmap of values between zero and one. We either interpret them as an indicator (for regression), or a probability (classification). To evaluate against 2D ground truth map data in Sec. 4 we project the heatmap to 2D.

3.3. Improved Prediction with Few-Shot Learning

The similarity score only provides relative estimates between zero and one; when we want to predict actual values like the building ages in Sec. 4.1.1, we can use VLM features as inputs to a model that operates on a point-to-point

basis. To do so, we split the ground truth values into q quantiles and train a classifier. At inference time, we multiply the predicted class probabilities with the bin centers to obtain continuous values. Experiments indicate that this discretization improves robustness over direct regression. We experiment with both K-Nearest Neighbours (KNN) Classifiers and Light Gradient Boosting Machines (LGBMs [11]). Unless otherwise stated, we use $q = 5$ with 30% of the data as training set and report the average of five random draws.

3.4. GPT-4o Integration

We also propose a GPT-4o-based [16] version of OpenCity. For each image, we prompt it to provide a score based on the given query. The result is again fused into the point cloud (details in supplementary). Note that this method comes at the cost of significant processing time per query, as it requires re-processing every image upon change.

4. Experiments

We choose three datasets to evaluate our method. Based on cadastre records we estimate building footprints and construction years in seven cities in the Netherlands (Sect. 4.1). In Sec. 4.2 we predict property values, evaluating against Zillow [2] data in seven cities in North America. Lastly, we estimate population density, crime rate, and noise pollution in Buenos Aires (Sect 4.3) based on official records.

4.1. Dataset 1: The Netherlands

We create a dataset of **building footprints** and associated **construction years** using the BAG API [19], which provides cadastre data for the entire Netherlands. With its granularity and size, this database represents a unique, high-quality data collection and thus an interesting opportunity to evaluate the capabilities of modern VLMs. The resulting dataset comprises of 19349 annotated buildings.

Along with the cadastre data, we extract corresponding meshes from Google Earth [1] and process them as described above. The result is an 'enriched' point cloud, in which each point has five hierarchies of VLM features attached to it. We use the features to infer building footprints and construction years in a prompt-driven zero-shot setting.

4.1.1 Building Footprints

Using the point cloud and per-point features, we segment the area into the classes *building* and *background* as given by the BAG cadastre data in a zero-shot setting.

To this end, we query the enriched point cloud with the positive query `'building'` and a set of canonical queries representing common urban objects such as `'tree'` `'road'`, or `'car'` (listed in the appendix) as background classes. The resulting similarity score is computed according to Eq. 2 and interpreted as a probability score. The scores are then projected onto a 2D plane and interpolated linearly to a regular grid to avoid edge artifacts. We then assign each point its ground-truth label based on the presence of a building in the BAG dataset.

We find that this classifier attains a Receiver Operating Characteristic Area Under the Curve [4] (ROC-AUC) score between 86.0% and 94.6%, accompanied by accuracies in the range of 83.2% and 89.8 % given an appropriately chosen threshold. The ROC-AUC score indicates how clearly a classifier distinguishes positive from negative classes. This is a significant improvement compared to LangSplat-style features projected to the same point cloud, which achieve only 79.8 % accuracy with a ROC-AUC of 86.2 % on the Rotterdam scene. Furthermore, our method strongly benefits from projecting the features to a 3D point cloud. When using a flat 2D point grid instead, scores degrade significantly (see appendix).

4.1.2 Building Age

Given a point cloud and per-point features, we predict the same buildings' construction year in a zero-shot setting. We predict age scores by comparing the positive prompt `'modern building'` to the negative `'old building'`. The ratio (Eq. 2) between the similarity of the two is our indicator for the building age. Then we again project the points to two dimensions and re-sample them on a regular grid. Each point within a building is assigned a ground truth construction year, all other points are ignored.

The results are displayed in Tab. 1. With Spearman correlations above 50% for both zero-shot approaches in four out of seven scenes, our model provides a robust first zero-shot baseline for vision-based building age prediction.

In the few-shot setting, we train an LGBM Classifier [11], which predicts an actual construction year instead of an indicator. When trained *within* scenes (Tab. 1), this consistently results in higher correlations, combined with robust F1 scores between 0.42 and 0.64. Yet, outliers such as medieval churches lead to varying Mean Absolute Error (MAE) scores - particularly in the historic Amsterdam scene. Similar results of experiments *across* scenes are displayed in the supplementary material.

Fig. 5 further shows how the method is able to distinguish entire districts with more modern architecture from more traditional areas. Modern houses that are built back-to-back to older houses as seen in Fig. 8 are harder to differentiate, as these are often built to match the style of the existing neighborhood. We furthermore find that OpenCity outperforms LangSplat-style features, which achieve lower performance on the task (see Tab. 2).

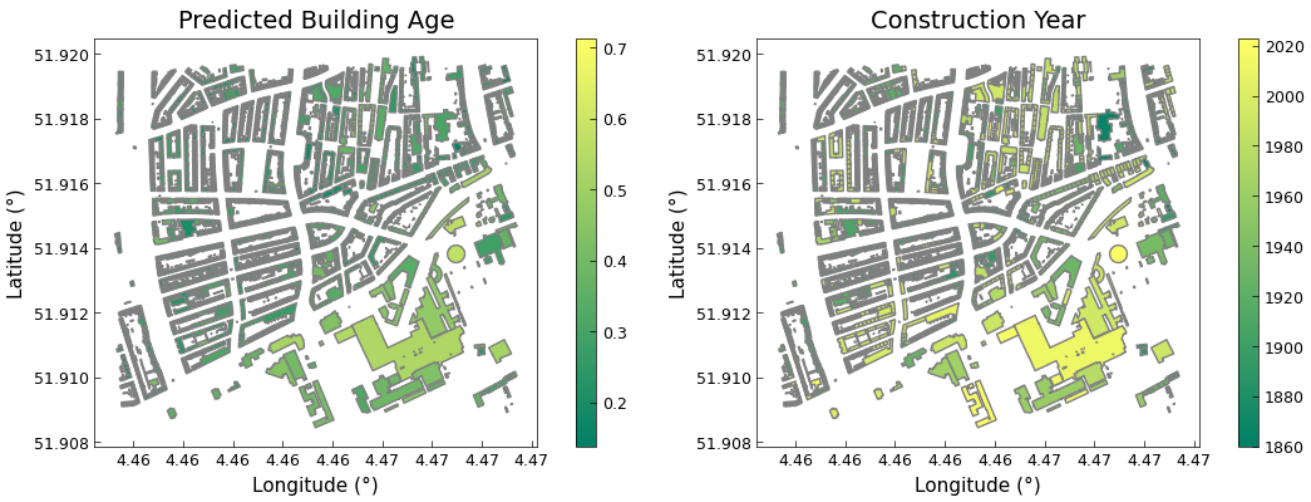


Figure 5. Zero-shot predicted age (left) vs. ground truth construction years (right) in Rotterdam.

Scene	Method	Building Age			Building Segmentation		
		Correlation	F1 Score	MAE [y]	Max Accuracy	ROC-AUC [4]	F1 Score
Rotterdam	OpenCity (prompt)	0.556	*0.317**	*35.7**	**87.7%**	**0.927**	**0.796**
	OpenCity (LGBM)	**0.769**	**0.639**	**20.9**	84.6%	0.906	0.702
	OpenCity (GPT-4o)	0.565	*0.566**	*34.5**	–	–	–
Amsterdam	OpenCity (prompt)	0.507	*0.343**	*240.22**	**85.3%**	**0.860**	**0.722**
	OpenCity (LGBM)	**0.577**	**0.419**	**97.7**	76.5%	0.853	0.642
	OpenCity (GPT-4o)	0.293	*0.300**	*251.22**	–	–	–
The Hague	OpenCity (prompt)	0.533	*0.321**	*151.34**	83.8%	0.925	**0.791**
	OpenCity (LGBM)	**0.689**	**0.457**	**50.0**	**86.4%**	**0.928**	0.761
	OpenCity (GPT-4o)	0.498	0.313	146.80	–	–	–
Utrecht	OpenCity (prompt)	0.364	*0.280**	*170.76**	**83.2%**	**0.866**	**0.752**
	OpenCity (LGBM)	**0.516**	**0.482**	**58.3**	74.1%	0.813	0.703
	OpenCity (GPT-4o)	0.502	*0.355**	*156.76**	–	–	–
Eindhoven	OpenCity (prompt)	0.430	*0.270**	*27.63**	87.2%	**0.931**	**0.798**
	OpenCity (LGBM)	**0.753**	**0.501**	**12.82**	**87.7%**	0.899	0.626
	OpenCity (GPT-4o)	0.636	*0.380**	*21.60**	–	–	–
Groningen	OpenCity (prompt)	0.636	*0.329**	*81.52**	**89.8%**	**0.946**	**0.813**
	OpenCity (LGBM)	**0.744**	**0.503**	**19.28**	84.4%	0.901	0.665
	OpenCity (GPT-4o)	0.716	*0.392**	*74.9**	–	–	–
Maastricht	OpenCity (prompt)	0.473	*0.332**	*223.81**	**84.5%**	**0.901**	**0.760**
	OpenCity (LGBM)	**0.717**	**0.542**	**57.3**	81.8%	0.889	0.722
	OpenCity (GPT-4o)	0.527	*0.341**	*210.50**	–	–	–

Table 1. Overview of OpenCity results for building age prediction *within* various cities in the Netherlands. The asterisk (*) indicates scores estimated by matching the score with the ground truth distribution based on quantiles, which is described in the supplementary material.

4.2. Dataset 2: North American Property Prices

We use Zillow [2] to create a dataset of **housing prices**. Zillow is a commercial analytics tool for the US real estate market that combines property data from public records with property listings from various sources. Our dataset consists of 1260 homes sold between 2020 and 2024 with corresponding sales prices and locations for seven cities. Then, we generate the meshes using Google Earth [1] and process them according to Sec. 3.1 to obtain a point cloud

Feature Type	Age Correlation	Building seg. max accuracy
LangSplat		
+ CLIP + prompt	0.394	79.8
+ CLIP + KNN	0.544	80.7
+ SigLIP + prompt	0.186	81.3
+ SigLIP + KNN	0.577	80.9
Highlighting-Based (ours)		
+ CLIP + prompt	0.520	76.1
+ CLIP + KNN	0.681	79.1
+ SigLIP + prompt	0.556	**87.7**
+ SigLIP + KNN	**0.728**	83.0

Table 2. Comparison of various feature extraction methods evaluated on the Rotterdam scene. Mask highlighting, done by OpenCity, outperforms the white-background LangSplat approach. For LangSplat, the uncompressed, point-projected features were evaluated.

with associated point features at the coarsest feature level.

4.2.1 Housing Prices

Taking the point cloud with per-point features as input, we estimate the sales price of the listed homes.

To this end, we construct an indicator analogous to the previous section, using `'expensive property'` as positive and `'cheap property'` as negative prompt. The result is interpreted as a score for expensiveness, projected to 2D and linearly interpolated to the known coordinates of the sold properties in the Zillow dataset.

The resulting score has correlations between 0.28 and 0.67 with the ground-truth sales prices. Training a LGBM Classifier *across* scenes improves upon this (Tab. 3) and reaches a MAE of 0.25M$, which is significantly better than chance ($0.52M$ MAE).

These results indicate that VLMs understand some of the mechanics that determine urban property value. Their features may be a valuable addition to larger parametric models such as Zillow's *Zestimate* [7].

4.3. Dataset 3: Buenos Aires

We collect official statistics from the Autonomous City of Buenos Aires (CABA) of **population count** [3], **crime records** [5], and urban **noise emissions** [6]. Along with them, we process one larger mesh sourced from Google Earth [1] following Sec. 3.1 to obtain a point cloud with point-wise VLM features, using the coarsest feature level.

4.3.1 Population Density

Given the point cloud and features, we use prompts to estimate population density as given by the CABA data.

The population density is given at the granularity of neighborhoods and computed by dividing the number of residents between 2015 and 2018 by the area (see Fig. 7 a). We build an indicator using the positive prompts `'densely populated area'`, and `'strongly populated district'`. As negatives, we choose `'loosely populated area'`, and `'unpopulated area'`. Once again, we project the points to two dimensions, resample them to a regular grid, and assign them the ground truth value taken from the CABA records.

We find that the indicator yields a Spearman correlation of 0.625. The model correctly identifies the population cluster in the north-western section (see Fig. 7 a). However, it erroneously assigns high scores to the city center south of the train station. With the two additional negatives `'nature'` and `'industrial area'`, the correlation is boosted to 0.753.

We also evaluate the features in a few-shot setting, using 28 training and 94 validation neighborhoods to train a KNN regressor. This results in a similar correlation of 0.61 (see Tab. 4).

These comparably strong results do not come unexpectedly. The population density is in a direct relationship with the number and size of visible residential buildings.

4.3.2 Crime Rate

Given the same features we predict Buenos Aires crime rates and validate the result against the CABA records.

CABA provides locations and descriptions of all recorded crimes between 2016 and 2022 [5]. We remove any crimes that do not involve a weapon to exclude incidents that are not necessarily tied to a location, such as tax evasion or fraud. This leaves us with a dataset of 2146 crimes within the scene. To avoid artifacts at region boundaries and attenuate sparsity effects, we consider each crime a 2D Gaussian distribution ($\sigma = 50$m), from which we sample to compute the ground truth expected number of annual armed crimes per km^2 and neighborhood.

As an estimator we invoke Eq. 2, using positive query `'dangerous neighborhood'` and the negative `'safe neighborhood'`. The resulting indicator obtains a relatively low Spearman correlation of 0.30. As visualized in Fig. 7 b, the task mainly consists of identifying the port-facing side of the north-western district as a dangerous area. The model however assigns high danger scores to the port as well as the park to the southeast.

We can once again include prior knowledge to increase the correlation to 0.42. In this case, however, this prior is less easily justified, as large city parks do not universally induce lower crime rates - though the mere absence of people may indicate such a tendency.

		Detroit	Miami	San Juan	Boston	San Fran.	Seattle	Los Angeles	**Overall**
Spearman	OpenCity (prompt)	0.528	**0.492**	0.348	0.278	0.674	**0.419**	0.504	0.402
	OpenCity (LGBM)	0.506	0.338	**0.432**	**0.433**	0.568	0.414	**0.728**	**0.739**
	OpenCity (GPT-4o)	**0.600**	0.487	0.260	0.194	**0.710**	0.366	0.192	0.339
F1 Score	OpenCity (prompt)	0.298	0.278	0.337	0.254	0.381	0.300	0.294	0.308
	OpenCity (LGBM)	**0.489**	**0.398**	**0.340**	**0.397**	**0.594**	**0.485**	**0.541**	**0.491**
	OpenCity (GPT-4o)	0.362	0.318	0.337	0.223	0.413	0.309	0.212	0.309
MAE [M$]	OpenCity (prompt)	0.201	**0.354**	0.477	0.173	0.179	0.365	0.276	0.360
	OpenCity (LGBM)	**0.174**	0.698	**0.389**	**0.160**	**0.163**	**0.349**	**0.174**	**0.251**
	OpenCity (GPT-4o)	0.195	0.364	0.498	0.189	0.171	0.419	0.350	0.373

Table 3. Result overview for housing price prediction *across scenes* in North America for zero- and few-shot setting. Zero-shot estimates of F1 and MAE are again computed by quantile-based distribution matching as described in the supplementary material.

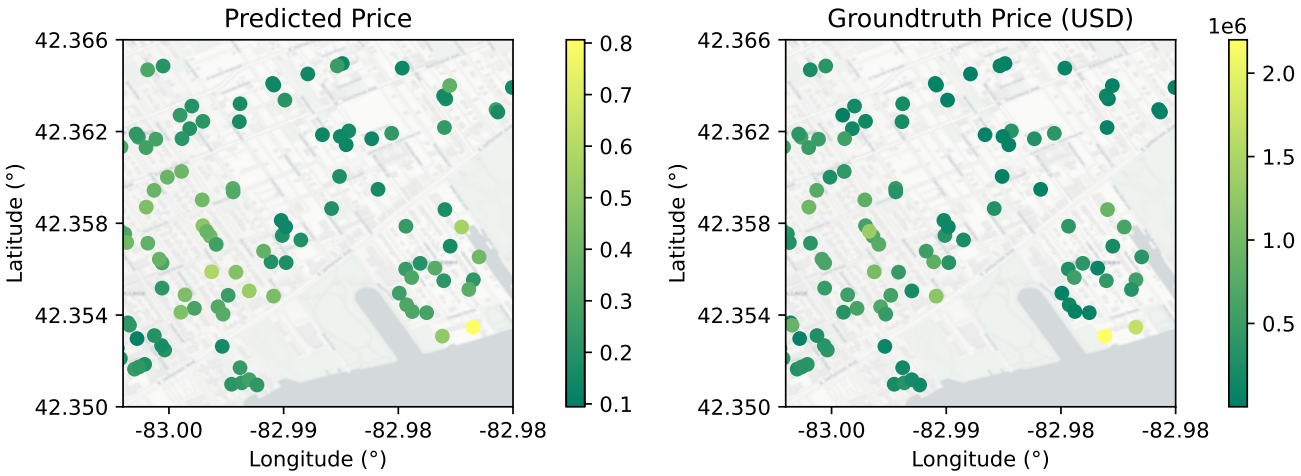


Figure 6. Zero-shot predicted (left) vs. ground truth (right) prices of sold homes in the Detroit scene. Basemaps are from CartoDB [8]

When evaluated in the aforementioned few-shot setting, KNN classification on the averaged neighborhood embeddings results in an improved correlation of 0.67.

In summary, predicting crime rates presents itself as a complex task where many influential factors may not be immediately visible. Having reference values, like in the KNN version, greatly increased the quality of the results. This finding indicates the need for a more nuanced approach, potentially incorporating a broader range of data types.

4.3.3 Noise Pollution

We follow the same procedure to estimate urban noise levels, comparing the results to official CABA measurements.

The relevant CABA noise emission dataset [6] provides a map of estimated average daytime noise in decibels along major city roads (see Fig. 7 c). To build an estimator, we again prompt the features using Eq. 2 with `'noisy urban area'` as positive prompt, contrasted with `'quiet area'` as a negative. This gives us a weak Spearman correlation of 0.19.

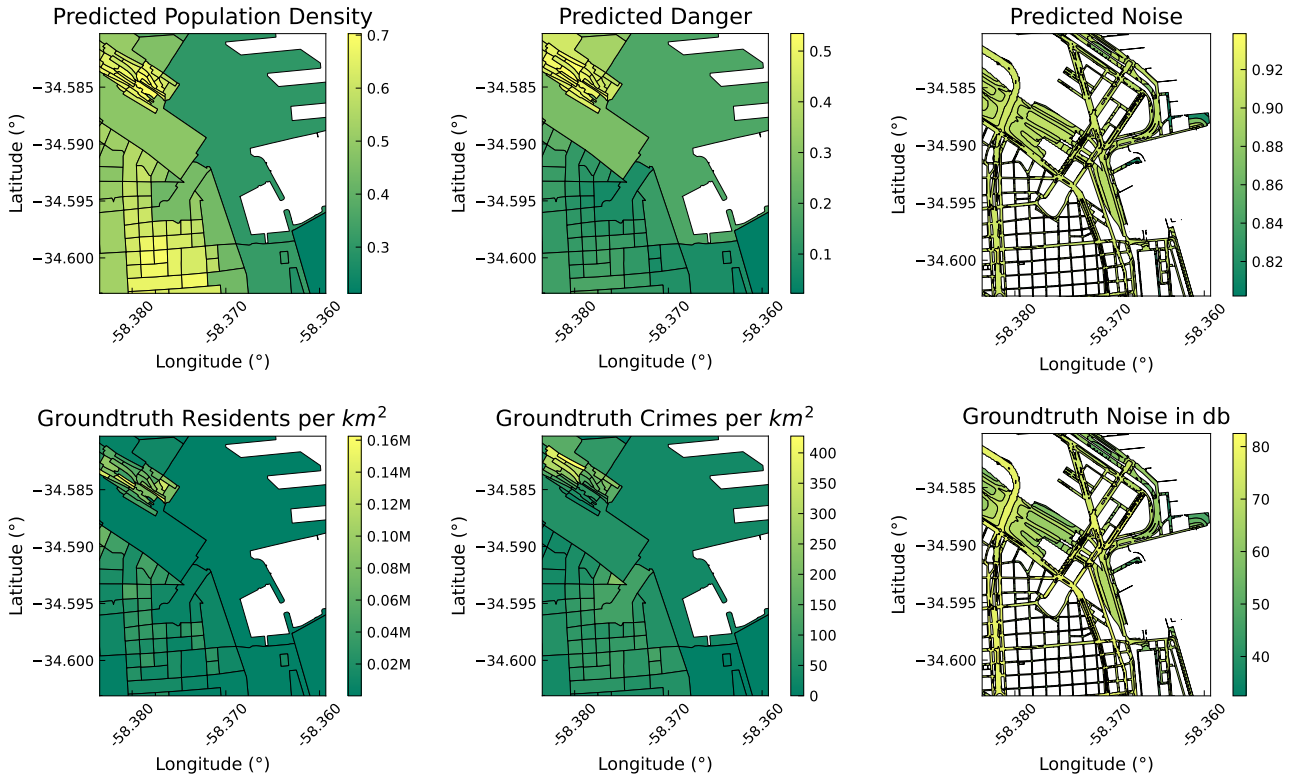
In the few-shot setting, we train a KNN regressor and obtain a moderate correlation of 0.71. Similar to the prediction of crime rates, noise level estimations remain difficult for VLMs, in particular in a zero-shot setting.

Model	Population Density	Crime Rate	Noise Level
Prompt	**0.625**	0.422	0.198
KNN	0.609	**0.673**	**0.716**
GPT-4o	0.451	0.544	0.286

Table 4. Spearman correlations for predictions population density, crime rate, and noise levels on the Buenos Aires dataset.

5. Limitations

A current limitation of large-scale urban 3D scene understanding is the lack of established datasets and test benchmarks; in this work, we provided a first step towards that



(a) Population Density. **(b)** Crimes per km^2 and year. **(c)** Measured noise emissions.

Figure 7. Zero-shot prediction *(top)* and ground truth *(bottom)* on Buenos Aires scenes.

Figure 8. Showcase of the difficulty of determining the age of houses in the Rotterdam mesh. The buildings were built back-to-back; the left one in 1907 and the right one in 1997.

direction by presenting baselines for two sizeable data collections using the BAG building dataset [19] and the Zillow housing dataset [2]. Yet our choice of dataset is limited to locations where public data is available, which can bias our findings towards more developed places that collect such data. Further, the scale of large cities remains a technical limitation of our approach. Unlike existing methods such as LangSplat [21], we don't compress the VLM feature space to three dimensions, which preserves better open-vocabulary properties at the cost of higher memory consumption. In either case, large cities are processed in rectangular chunks, which can lead to artifacts at the borders of neighboring chunks if the overlap is insufficient. Another technical limitation is that images rendered from relatively low-quality meshes (such as ours) may be less well understood by VLMs. To avoid biases, we recommend only comparing predictions of meshes of the same quality and origin. Furthermore, the underlying imagery is in some cases more recent than the ground-truth data (see supplementary material). Further, OpenCity can manifest social and cultural biases inherited from the visual language models. Those biases stem from the under- or over-representation of certain demographic groups in the training datasets. In particular, for tasks like crime rate prediction (Sec. 4.3.2), our method can perpetrate and reinforce stereotypes and systemic discrimination with the limited availability of diverse test data as revealed by Pouget *et al.* [20]. Mechanisms such as those proposed by [25] may help to mitigate these effects.

6. Conclusion

In this work, we investigated foundation models, in particular large visual-language models, and their understanding of urban properties. With our method OpenCity, we explore the capabilities of VLMs at city scale, focusing on inferring higher-level characteristics such as population density, building age, property value, crime rate, and noise levels. Our findings suggest that VLMs exhibit significant potential in urban scene analysis. Although predictions for crime rates and noise levels remain less robust and may contain substantial bias, our experiments on population density, building age, and property value demonstrate considerable promise for advancing urban research. Overall, our experiments raise hopes that VLMs can contribute significantly to city-scale urban scene understanding, and we hope that our work encourages further research into this direction.

Acknowledgments. This project is partially supported by an ETH AI Center Postdoctoral Fellowship and an SNF PostDoc mobility fellowship.

References

[1] Google 3d tiles. https://www.google.com/3dtiles/. Accessed: 2024-06-15. 2, 4, 5, 6, 15

[2] Zillow group, inc. https://www.zillow.com/homes/. Accessed: 2024-07-13. 4, 5, 8

[3] Bits and Bricks. Buenos aires population density. https://bitsandbricks.github.io/data/CABA_rc.geojson, 2024. Accessed: 2024-06-15. 6

[4] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. 4, 5, 14

[5] Buenos Aires City Government. Buenos aires government open data portal, 2021. Accessed: 2024-05-18. 6

[6] Buenos Aires City Government. Buenos aires government open data portal, 2021. Accessed: 2024-05-18. 6, 7

[7] Angela Burden. Imputing data for the zestimate. https://www.zillow.com/tech/imputing-data-for-the-zestimate/. Accessed: 2024-06-15. 6

[8] CARTO. Carto basemap styles. Accessed: 2024-07-16. 7, 17, 18, 19, 20

[9] Meida Chen, Qingyong Hu, Thomas Hugues, Andrew Feng, Yu Hou, Kyle McCullough, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset, 2022. 2, 14

[10] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. *arXiv*, 2023. 14

[11] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017. 4, 11

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 1

[13] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[15] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[16] OpenAI. Open ai. hello gpt-4o. https://openai.com/index/hello-gpt-4o. Accessed: 2024-09-9. 4

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 11

[18] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, 2023. 1, 2

[19] Ravi Peters, Balázs Dukai, Stelios Vitalis, Jordi van Liempt, and Jantien Stoter. Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands, 2022. 4, 8

[20] Angéline Pouget, Lucas Beyer, Emanuele Bugliarello, Xiao Wang, Andreas Peter Steiner, Xiaohua Zhai, and Ibrahim Alabdulmohsin. No filter: Cultural and socioeconomic diversityin contrastive vision-language models. *arXiv preprint arXiv:2405.13777*, 2024. 8

[21] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *arXiv preprint arXiv:2312.16084*, 2023. 1, 2, 3, 8

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[23] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 14

[24] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023. 2, 14

[25] Ashish Seth, Mayur Hemani, and Chirag Agarwal. Dear: Debiasing vision-language models with additive residuals, 2023. 8

[26] Corinne Stucker, Bingxin Ke, Yuanwen Yue, Shengyu Huang, Iro Armeni, and Konrad Schindler. ImpliCity: City Modeling from Satellite Images with Deep Implicit Occupancy Fields. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2022. 2

[27] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 11, 14

[28] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. 1, 2, 3

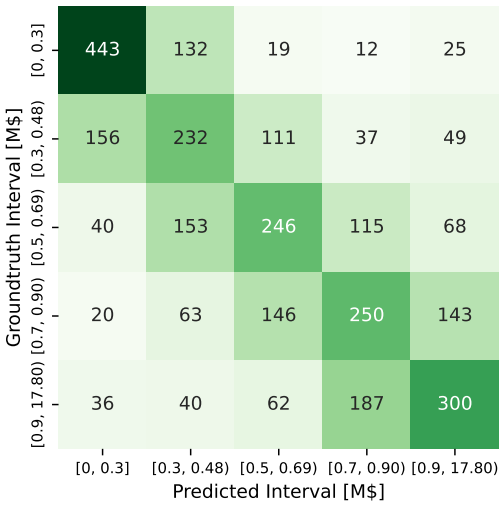


Figure 9. Confusion matrix of property price classification with LGBM [11] *across* scenes.

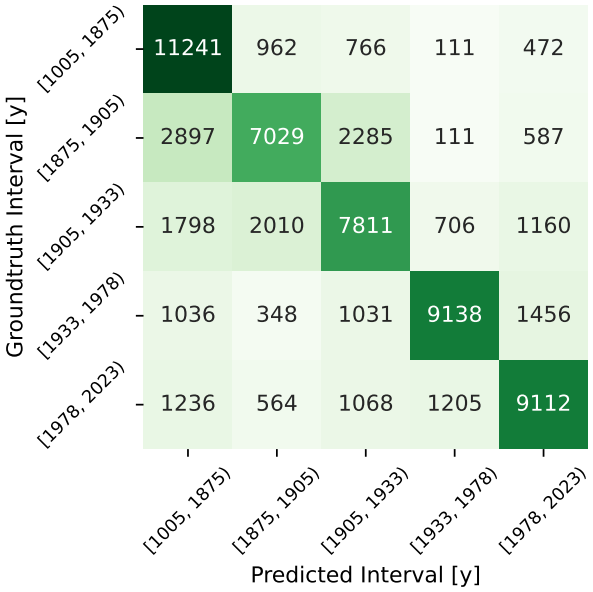


Figure 10. Confusion matrix of building age classification with LGBM [11] *across* scenes.

A. Additional Results

A.1. Evaluation Across and Within Scenes

In the main paper, we presented results for building age prediction *within* cities and property price estimation *across* scenes. The complementary results for building age *across* cities and property price *within* scenes are presented in Tables 1 and 6, featuring additional metrics. Furthermore, confusion matrices are visualized in Figures 9 and 10.

A.2. Evaluation with more Training Data

We find that the results *across* scenes can be significantly boosted when training with more than 30% of the dataset. Figures 11 and 12 visualize this effect.

A.3. Ablation: 3D Point Cloud vs. Flat Grid

Although only evaluating on a 2D grid we find that the usage of a 3D point cloud is beneficial for feature fusing. In table 7 we demonstrate that performance degrades significantly if projecting to a flat 2D point grid instead. We believe that this is caused by the imprecise attribution of points to masks.

B. Implementation Details

B.1. Dataset Creation

We sample positions based on a 2D grid, adding random offsets on all axes. The angle to the z-xis is sampled between 0 and 90 degrees to avoid sky-facing perspectives. The other angles are sampled uniformly at random. RGB-D images with depth closer than 50m and images with infinite depth in more than 20% of the pixels are discarded. See Table 9 for details on the scenes.

B.2. Projection to Point Cloud

The point cloud is first downsampled to 1M points (0.5M if only the coarsest level was processed) to reduce memory consumption. Following OpenMask3D [27], point visibility is determined based on depth.

However, we filter the masks before projection. As most segments only cover a handful of pixels, we retain only those that cover at least 0.25% of the image. This leads to the removal of roughly 60% of all segments and speeds up the overall processing time by 40%.

B.3. Prompting the Point Embeddings

As mentioned in the main paper, we prompt the model with *positive* and *negative* queries. We find that the choice of negatives can have a strong impact on performance. For building segmentation, the full set of negatives was: `'tree'`, `'road'`, `'park'`, `'river'`, `'car'`, `'sea / lake / canal'`, `'parking lot'`, `'urban scene'`, and `'city'`.

B.4. Estimation

We use scikit-learn [17] to build unweighted KNN regressors and classifiers ($k = 5$). Each point and feature level provides a data point. As for LightGBM [11], we use the official package with default settings. We find that classifiers on building age, crime rate, noise levels, and popula-

F1 Score	Overall	Amsterdam-	The Hague-	Eindhoven-	Groningen-	Maastricht-	Rotterdam-	Utrecht-
lgbm	**0.67**	**0.54**	**0.47**	**0.81**	**0.75**	**0.60**	**0.76**	**0.59**
linear	0.61	0.52	0.38	0.76	0.66	0.56	0.55	0.53
knn	0.61	0.51	0.43	0.78	0.70	0.54	0.70	0.49
dummy	0.20	0.23	0.21	0.28	0.24	0.21	0.23	0.21
Spearman Correlation								
lgbm	**0.73**	**0.32**	**0.56**	**0.40**	**0.84**	**0.65**	**0.76**	**0.68**
linear	0.67	0.29	0.46	0.32	0.70	0.61	0.57	0.60
knn	0.67	0.25	0.46	0.33	0.77	0.56	0.67	0.52
dummy	0.00	-0.01	0.01	-0.02	0.03	-0.00	-0.01	0.01
MAE [y]								
lgbm	**50.85**	**122.23**	**57.99**	**12.64**	**18.26**	**63.50**	**15.65**	**60.62**
linear	62.84	137.46	88.79	13.09	25.09	82.48	22.31	68.57
knn	55.62	125.30	62.59	14.46	24.12	67.76	18.67	72.12
dummy	102.95	166.55	93.28	77.03	88.49	106.14	75.75	109.80
MAPE [%]								
lgbm	**3.03**	**8.28**	**3.11**	**0.64**	**0.94**	**3.43**	**0.81**	**3.72**
linear	3.63	8.94	4.71	0.66	1.29	4.40	1.15	4.10
knn	3.30	8.53	3.36	0.73	1.23	3.67	0.96	4.31
dummy	5.85	11.10	5.03	3.94	4.51	5.82	3.93	6.33

Table 5. OpenCity few-shot results for construction year prediction trained *across* various cities in the Netherlands.

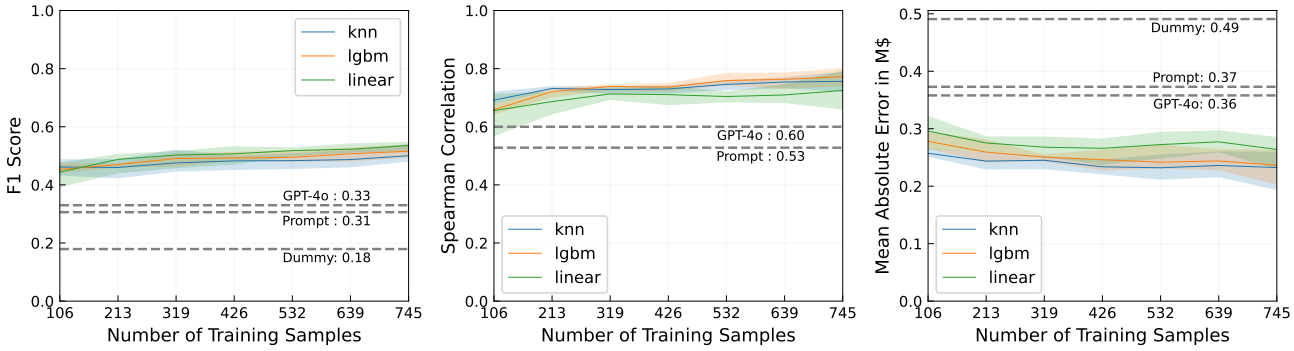


Figure 11. Property price estimation results against dataset size for experiment *across* scenes. Zero-shot MAE baselines were obtained from scores by matching quantiles.

tion density benefit strongly from reducing noise by averaging the embeddings of the relevant area before training and inference.

B.5. Projection of Scores to Ground Truth Scale

We experiment with methods to convert the scores into estimates matching the scale of the ground truth distribution. To that end, we compute the q quantiles of the predicted and the ground truth distribution. Then we assign a prediction in the i-th quantile of the score distribution the mean of the values in the i-th quantile of the true distribution. We implement this strategy with $q = 5$

B.6. GPT-4o Integration

We use GPT-4o to produce one score per prompt and image. The obtained score is then fused into the point cloud analogously to the embeddings. Due to cost and time constraints, we only process full images (coarsest level) and

F1 Score	Mean	Detroit	Miami	San Juan	Boston	San Fran.	Seattle	Los Angeles
lgbm	**0.34**	0.33	**0.25**	**0.38**	**0.34**	0.34	**0.33**	0.40
linear	0.28	0.30	0.19	0.33	0.29	0.24	0.22	0.38
knn	0.32	**0.34**	0.19	0.36	0.31	**0.35**	0.26	**0.45**
dummy	0.20	0.20	0.20	0.19	0.22	0.21	0.17	0.18
Spearman Correlation								
lgbm	0.49	0.55	0.24	**0.45**	**0.49**	0.57	0.39	0.75
linear	**0.51**	0.55	**0.30**	0.38	0.44	**0.68**	0.43	**0.79**
knn	**0.51**	**0.59**	0.29	0.39	0.41	0.63	**0.46**	0.77
dummy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MAE [M$]								
lgbm	0.34	0.19	1.03	0.39	**0.14**	0.17	0.32	0.14
linear	0.37	0.21	1.10	0.45	0.16	0.16	0.35	0.13
knn	**0.32**	**0.17**	**0.97**	**0.37**	**0.14**	**0.14**	**0.30**	**0.11**
dummy	0.52	0.28	1.29	0.55	0.20	0.39	0.51	0.39
RMSE [M$]								
lgbm	0.58	0.28	2.20	0.56	**0.17**	0.24	0.42	0.19
linear	0.60	0.31	2.23	0.64	0.21	0.22	0.44	0.17
knn	**0.55**	**0.26**	**2.15**	**0.54**	**0.17**	**0.19**	**0.39**	**0.15**
dummy	0.80	0.38	2.60	0.74	0.25	0.48	0.64	0.50

Table 6. OpenCity few-shot results for property price prediction trained *within* various cities in the US. This experiment was conducted using 50% of the samples as training data. The small training set size (down 30 samples) can otherwise lead to overfitting.

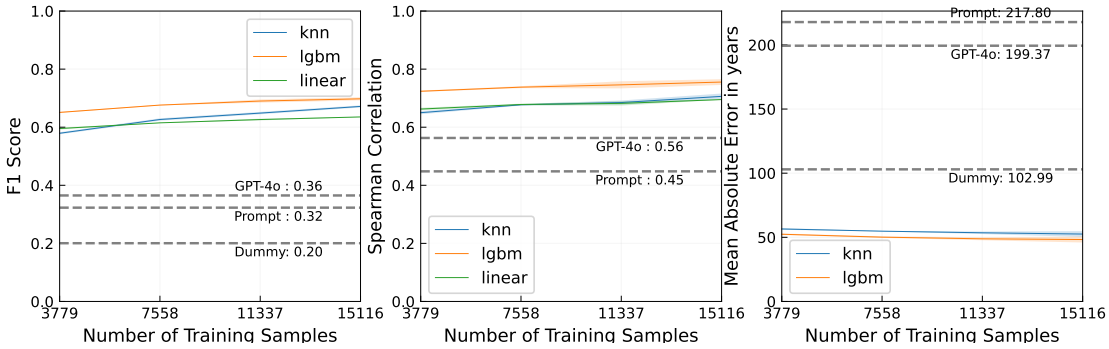


Figure 12. Building age estimation results against dataset size for experiment *across* scenes. Note how quantile matching fails to produce meaningful zero-shot baselines, producing MAE significantly worse than chance.

no individual masks. Table 8 shows the used prompts for the GPT experiments (GPT4o). For the property price and building age experiments, the rating has been grounded by providing reference values for ratings 3, 6, and 9. These reference values are obtained by binning the ground truth data into 10 bins. Despite this grounding, the resulting scores only match the ground truth distribution to a limited extent. We therefore evaluate them analogously to the similarity scores. The induced prompting cost scales with the number and quality of images as well as the length of the response. Our experiments with 7k to 10k images per scene cost 10-20$ per query. At the time of creation (September 2024), the inference time was roughly at 4-8h per scene.

Geometry Type	ROC-AUC [4]	F1 Score
3D Point Cloud		
+ prompt	**0.946**	**0.813**
+ KNN	0.828	0.625
Flat Geometry		
+ prompt	0.904	0.724
+ KNN	0.789	0.591

Table 7. Comparison of building segmentation performance in Groningen with a 3D point cloud vs. using a flat point grid.

B.7. Evaluation

Unless stated otherwise, the 3D point cloud is projected to 2D and then interpolated linearly to a regular grid. Correlation is computed on the points (not the districts/buildings). The validation set of the KNN estimators is uniformly randomly downsampled to 20k points per scene to reduce inference time. Preliminary experiments showed that this has no significant effect on the results.

C. Experiment: OpenMask3D for Urban Point Clouds

One of the key characteristics of OpenMask3D [27] is that it segments the input point cloud and then stores one feature per 3D segment. This greatly boosts storage and memory efficiency, making it well-suited for city-scale input.

Unfortunately, Mask3D [24], the 3D segmentation model used by OpenMask3D, failed to generate meaningful segments for our 3D city scenes. Neither OpenMask3D's Scannet200 [23] and STPLS3D [9] checkpoints, nor the more recent Segment3D [10] - a model claimed to have superior generalization performances compared to Mask3D - remedied the situation (see Fig. 13).

In particular, we find that the models display high sensitivity to the density and scale of the point clouds.

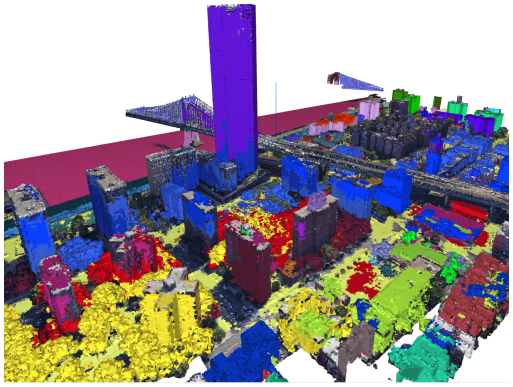


Figure 13. An example segmentation of a city area using Segment3D

D. Additional Visualizations

We provide qualitative results for open-set segmentation in Fig. 14. Figures 15 and 16 visualize the complete results for property price prediction, whereas figures 17 and 18 display the ones for building age prediction.

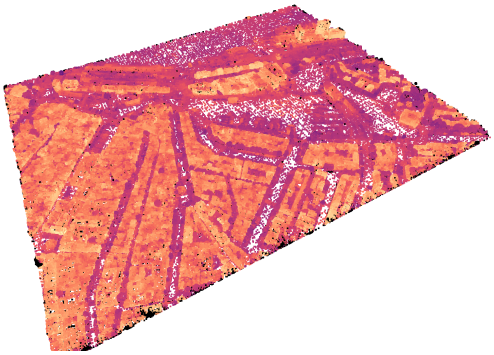
Experiment	**Prompt**
Noise Levels, Population Density and Dangerous Neighborhoods	Estimate the noise level, population density and how dangerous the neighborhood might be of the area shown in this image from 0 to 10. return the result without explanation
Property Prices	Estimate the average property value of the area in the US from a scale from 0 to 10: 3 meaning around 250k$ 6 meaning around 600k$ 9 meaning around 1.5m$ return the result without explanation
Building Age	Estimate the average building age of the area from a scale from 0 to 10: 3 meaning around 1739 6 meaning around 1883 9 meaning around 1987 return the result without explanation

Table 8. GPT4-o experiments and their corresponding prompts.

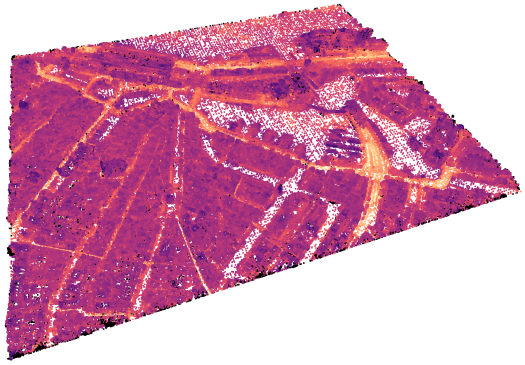
Scene	Area (km^2)	Latitude Bounds	Longitude Bounds	Sampling Year	Rendered Images
Buenos Aires (Argentina)	5.20	[-58.3801, -58.3593]	[-34.6041, -34.5803]	2021 - 2023	14261
Rotterdam (Netherlands)	1.68	[51.9088, 51.9194]	[4.4542, 4.4741]	2019 - 2023	5704
Amsterdam (Netherlands)	1.99	[52.3698, 52.3809]	[4.8937, 4.9174]	2021 - 2023	6597
The Hague (Netherlands)	1.70	[52.0782, 52.0887]	[4.3073, 4.3285]	2020 - 2023	6520
Utrecht (Netherlands)	1.78	[52.0818, 52.0929]	[5.0987, 5.1197]	2017 - 2019	6527
Eindhoven (Netherlands)	1.35	[5.42727, 5.44250]	[51.43233, 51.44241]	2015 - 2023	8946
Groningen (Netherlands)	1.10	[6.57495, 6.59036]	[53.21107, 53.21964]	2024	7310
Maastricht (Netherlands)	2.20	[5.68648, 5.70744]	[50.8425, 50.8525]	2011 - 2023	12390
San Juan (Puerto Rico)	3.45	[-66.0883, -66.0707]	[18.4475, 18.4642]	2016	9369
Detroit (USA)	4.12	[-83.0038, -82.9789]	[42.3467, 42.3648]	2019 - 2023	9649
Miami Beach (USA)	3.18	[-80.1444, -80.1272]	[25.7664, 25.7831]	2018 - 2022	9377
Seattle (USA)	2.10	[-122.39508, -122.36096]	[47.49694, 47.51248]	2018 - 2023	12834
Boston (USA)	3.83	[-70.99674, -70.96593]	[42.36831, 42.39076]	2018 - 2021	14800
San Francisco (USA)	1.98	[-122.16672, -122.15059]	[37.67978, 37.69241]	2022 - 2023	9822
Los Angeles	2.67	[-117.71718, -117.69846]	[33.61083, 33.62591]	2017 - 2024	7610

Table 9. Scene information. Sampling year indicates the time underlying footage for the reconstruction was taken according to Google Earth [1].

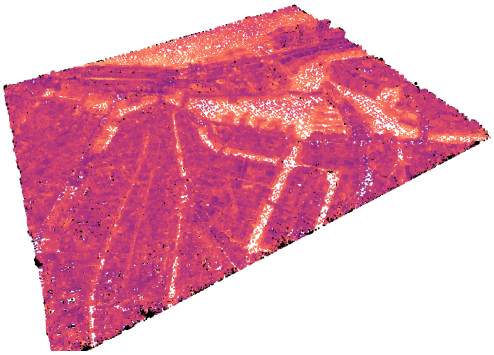
(a) Rendered mesh



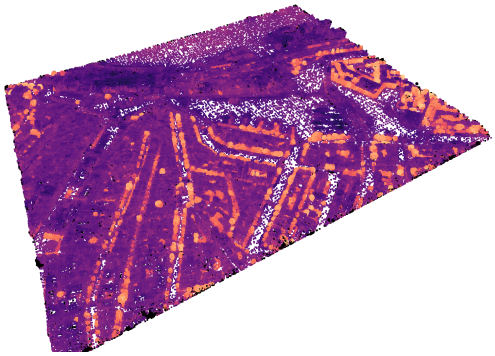
(b) Prompt ”building”



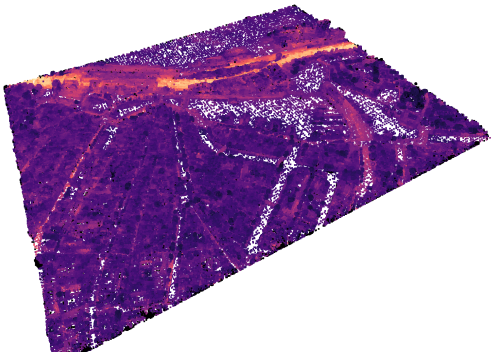
(c) Prompt ”road”



(d) Prompt ”water”



(e) Prompt ”tree”



(f) Prompt ”train tracks”

Figure 14. Qualitative results for open-set segmentation in Amsterdam. We can see that buildings 14b, trees 14e and train tracks 14f are recognized with high precision, but the model has difficulties for water 14d and roads 14c

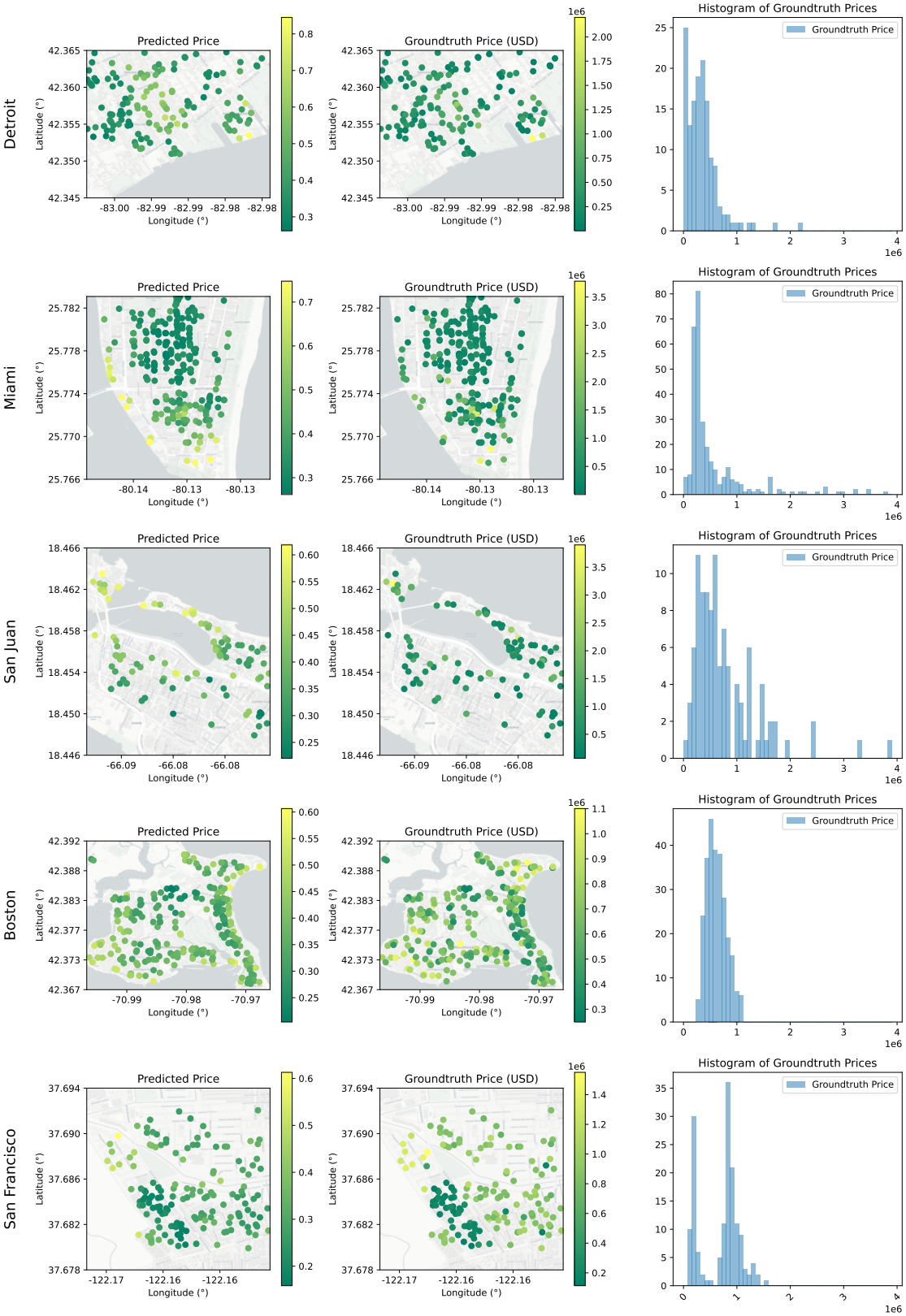


Figure 15. Visualization of zero-shot property price predictions (left) vs ground truth (right) by OpenCity. Basemaps are from CartoDB [8].

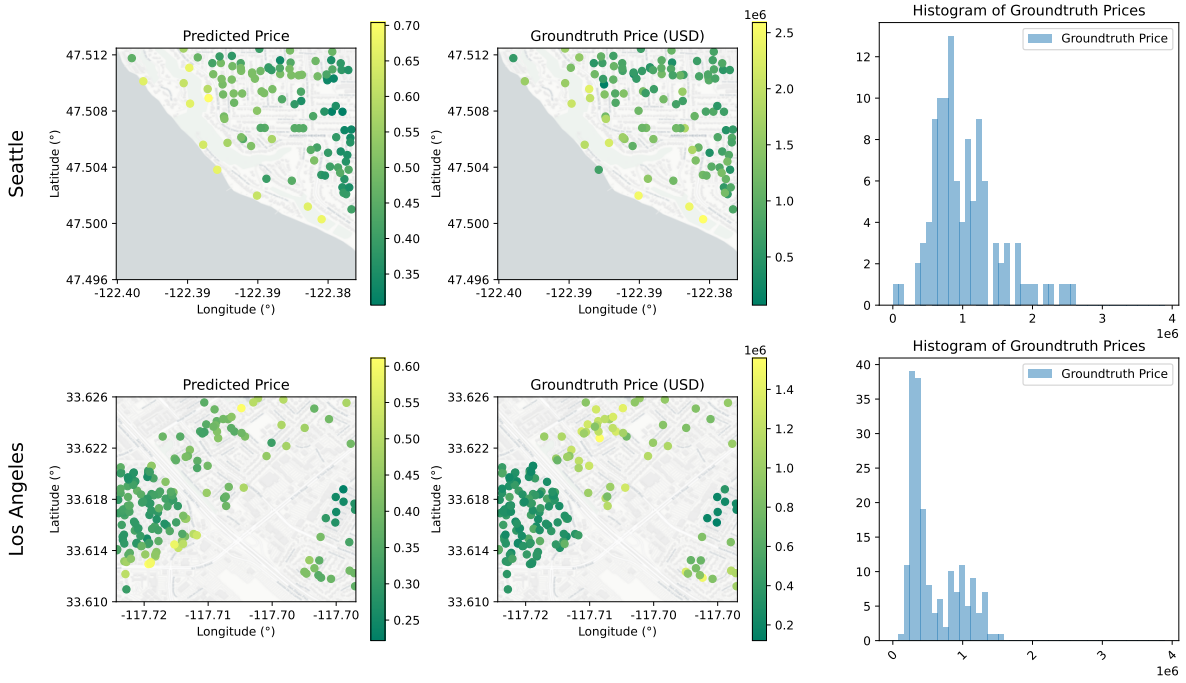


Figure 16. Visualization of zero-shot property price predictions (left) vs ground truth (right) by OpenCity. Basemaps are from CartoDB [8].

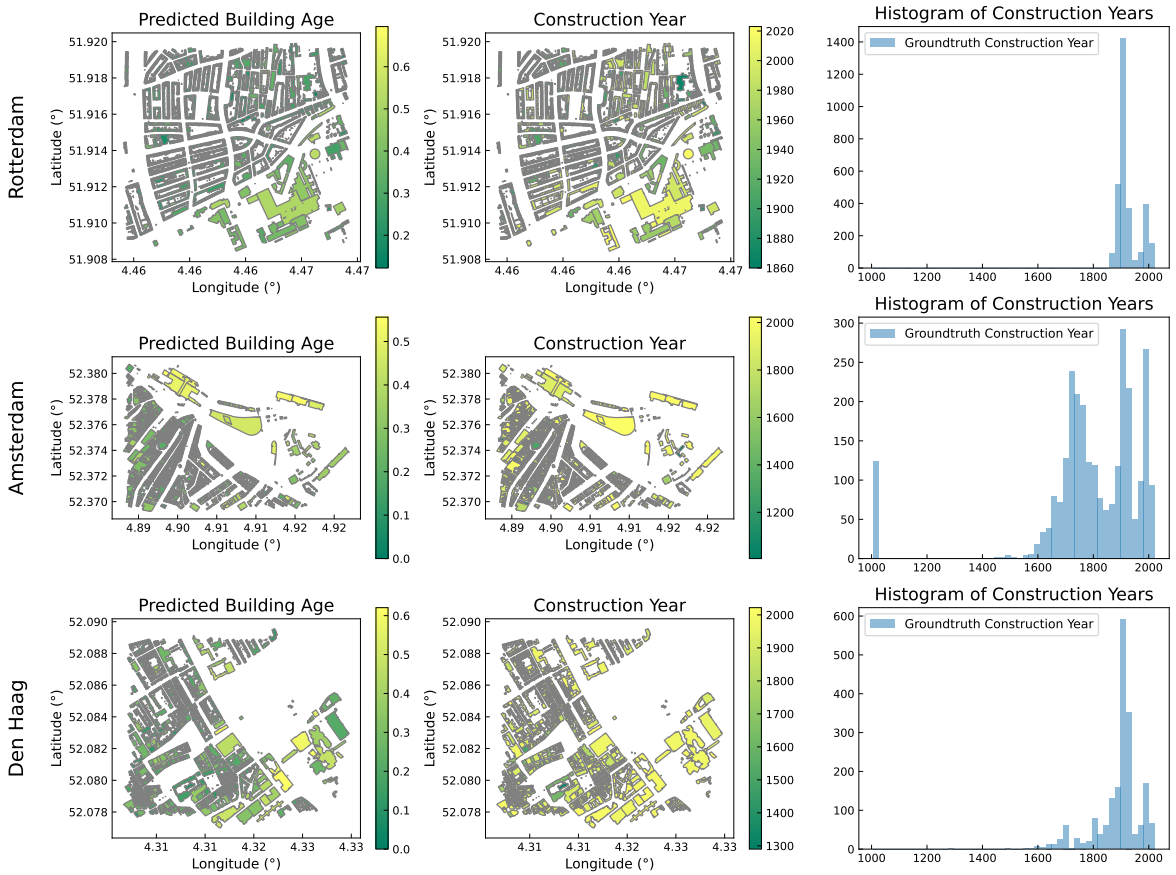


Figure 17. Visualization of zero-shot building age predictions (left) vs ground truth (right) by OpenCity. Basemaps are from CartoDB [8].

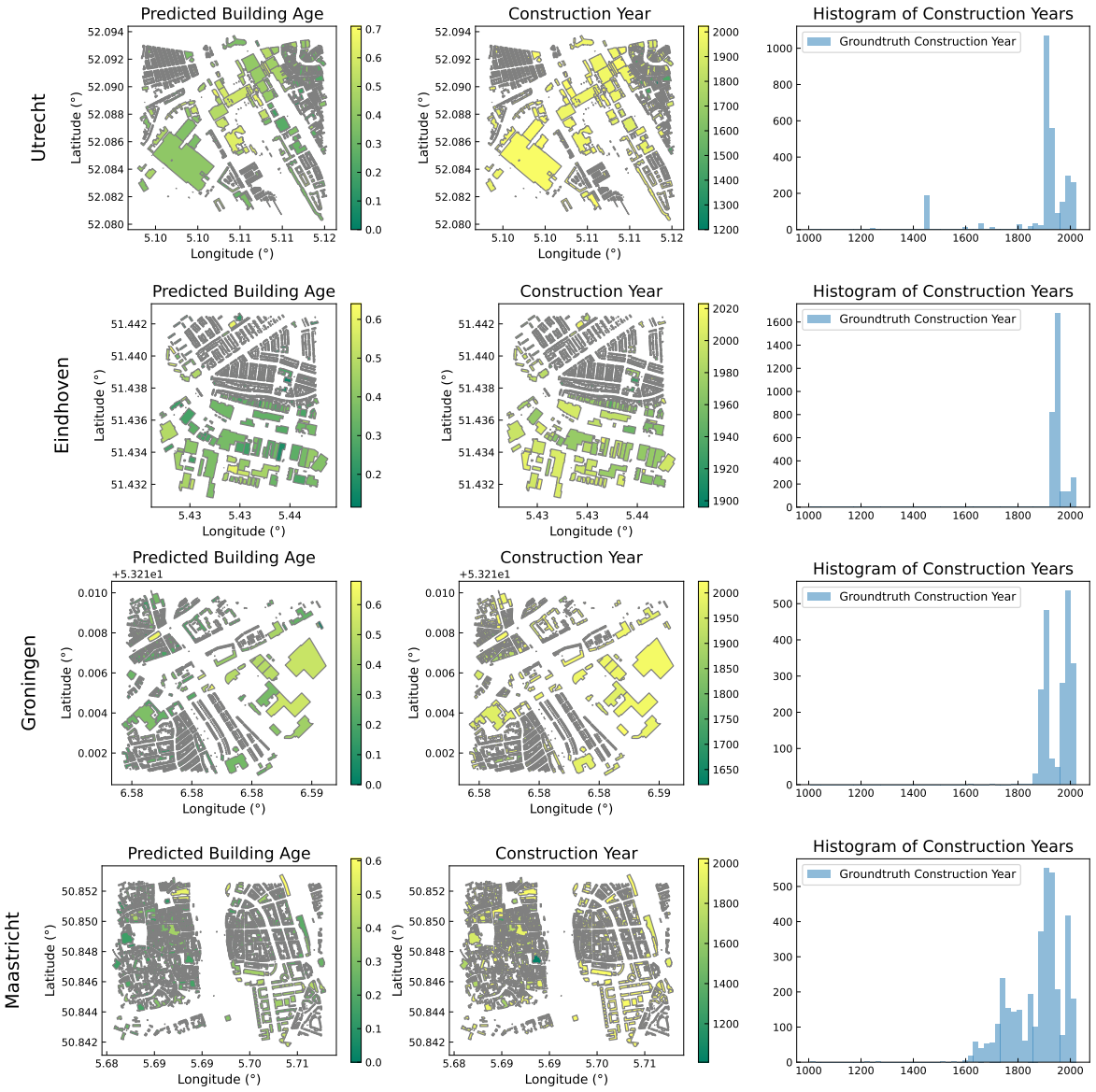


Figure 18. Visualization of zero-shot building age predictions (left) vs ground truth (right) by OpenCity. Basemaps are from CartoDB [8].

References

[1] Google 3d tiles. https://www.google.com/3dtiles/. Accessed: 2024-06-15. 2, 4, 5, 6, 15

[2] Zillow group, inc. https://www.zillow.com/homes/. Accessed: 2024-07-13. 4, 5, 8

[3] Bits and Bricks. Buenos aires population density. https://bitsandbricks.github.io/data/CABA_rc.geojson, 2024. Accessed: 2024-06-15. 6

[4] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. 4, 5, 14

[5] Buenos Aires City Government. Buenos aires government open data portal, 2021. Accessed: 2024-05-18. 6

[6] Buenos Aires City Government. Buenos aires government open data portal, 2021. Accessed: 2024-05-18. 6, 7

[7] Angela Burden. Imputing data for the zestimate. https://www.zillow.com/tech/imputing-data-for-the-zestimate/. Accessed: 2024-06-15. 6

[8] CARTO. Carto basemap styles. Accessed: 2024-07-16. 7, 17, 18, 19, 20

[9] Meida Chen, Qingyong Hu, Thomas Hugues, Andrew Feng, Yu Hou, Kyle McCullough, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset, 2022. 2, 14

[10] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. *arXiv*, 2023. 14

[11] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017. 4, 11

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 1

[13] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[15] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[16] OpenAI. Open ai. hello gpt-4o. https://openai.com/index/hello-gpt-4o. Accessed: 2024-09-9. 4

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 11

[18] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, 2023. 1, 2

[19] Ravi Peters, Balázs Dukai, Stelios Vitalis, Jordi van Liempt, and Jantien Stoter. Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands, 2022. 4, 8

[20] Angéline Pouget, Lucas Beyer, Emanuele Bugliarello, Xiao Wang, Andreas Peter Steiner, Xiaohua Zhai, and Ibrahim Alabdulmohsin. No filter: Cultural and socioeconomic diversityin contrastive vision-language models. *arXiv preprint arXiv:2405.13777*, 2024. 8

[21] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *arXiv preprint arXiv:2312.16084*, 2023. 1, 2, 3, 8

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[23] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 14

[24] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023. 2, 14

[25] Ashish Seth, Mayur Hemani, and Chirag Agarwal. Dear: Debiasing vision-language models with additive residuals, 2023. 8

[26] Corinne Stucker, Bingxin Ke, Yuanwen Yue, Shengyu Huang, Iro Armeni, and Konrad Schindler. ImpliCity: City Modeling from Satellite Images with Deep Implicit Occupancy Fields. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2022. 2

[27] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 11, 14

[28] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. 1, 2, 3