# OpenCity: What do Vision-Language Models know about Urban Dynamics?

Valentin Bieri
ETH Zurich
bieriv@ethz.ch

Marco Zamboni
ETH Zurich
mzamboni@ethz.ch

Nicolas S. Blumer
ETH Zurich, UZH
nblume@ethz.ch

Qingxuan Chen
ETH Zurich, UZH
qingxuan.chen@uzh.ch

## Abstract

*The rise of 2D vision-language models (VLMs) has enabled a new level of language-driven 3D scene interaction, setting new standards for prompt-based zero-shot inference on various tasks [8] [14] [6] [10]. However, most experiments are conducted on controlled indoor scenes and validated against concrete, tangible ground-truth classes.*

*In this work, we adapt existing methods to work on large, city-scale datasets generated using Google Earth [1] - with the goal of detecting not only urban inventory but also urban dynamics such as population density, building age, crime rate, and noise pollution.*

*Our analysis in zero-shot and few-shot settings indicates that VLMs have the potential to solve urban classification and localization tasks. Simultaneously, they have a surprisingly good understanding of some abstract phenomena but completely fail to identify others.*

## 1. Introduction

Recent developments in 3D scene representation, including Vision-Language Models (VLM), Neural Radiance Fields (NERF), and Gaussian Splatting, have significantly advanced open-set inference capabilities. However, these methods have predominately been evaluated in closed indoor environments, such as OpenScene [8], OpenMask3D [14], LangSplat [10], and LERF [6], which limits their applicability to the complexities of urban landscapes.

The city scale introduces unique challenges due to its scale and dynamic nature, that render existing methods less effective. Many dense reconstruction techniques are too expensive to run on such a scale. Understanding urban dynamics - ranging from the age of buildings to population density and crime rates - is crucial for urban planning and development. Despite their limitations, these methods offer valuable insights into urban dynamics, 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 Google Earth mesh data and generating an embedded point cloud using rendered RGB-D images, inspired by the feature extraction procedures of LangSplat. By leveraging language encoders, we query this embedded point cloud to analyze the information content of embeddings related to tangible urban objects such as buildings and dynamic urban phenomena like population density and crime rates.

Our findings indicate promising results in urban inventory localization segmentation, particularly for identifying building ages and population density, improving upon the capabilities of LangSplats features. 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 report 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

Several recent studies have explored advanced techniques in 3D scene understanding and instance segmentation. Peng et al. [8] introduced a method that assigns per-point features to point clouds, followed by a multi-view feature fusion using CLIP features [11]. Their approach, OpenScence, supports open-vocabulary queries but faces challenges in achieving sharp segmentation.

Another notable advancement is OpenMask3D [14], designed specifically for open-vocabulary 3D instance segmentation. OpenMask3D is a method that leverages CLIP embeddings to extend Mask3D [13], a model for 3D semantic instance segmentation, on an open vocabulary. To do so it uses SAM [7] 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 massively improves efficiency and storage usage, both important factors for scalability. This would make it a perfectly suited candidate for large urban scene representations.

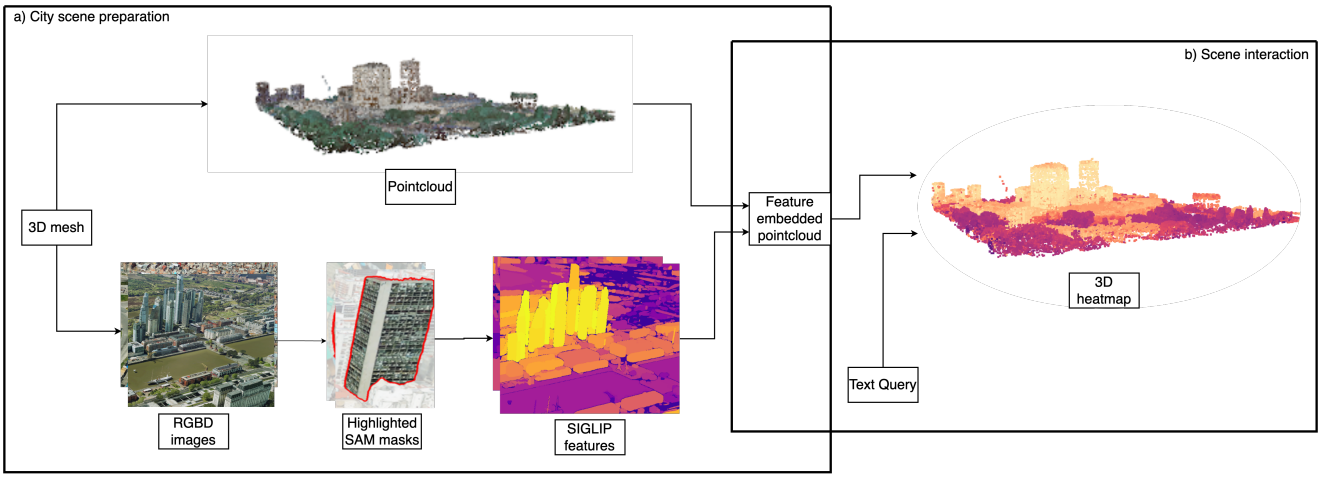


Figure 1: **OpenCity zero shot pipeline.** a] To prepare a city scene the point-cloud and RGB-D renderings are generated from the source mesh; highlighted masks are then extracted from the image using SAM [7] and are used to get visual language features using SigLIP [15] that can be combined with the point cloud. b] For the zero-shot approach the feature-embedded point cloud is compared to open vocabulary text embeddings and the similarity scores can be displayed on the point-cloud as an heat-map

However, preliminary experiments reveal that its segmentation model (Mask3D) is unable to generalize to our city scenes, which we believe to be out-of-distribution regarding Mask3D's training data. Alternative segmentation models such as Segment3D [5] did not remedy the situation. An example of Mask3D segmentation on an urban scene is displayed in Figure 2, and a more thorough report of our experiments is provided in the appendix.

Kerr et al. [6] proposed the Language-Enhanced Rendering Function (LERF), a 3D Neural Radiance Field (NERF) model. LERF integrates language features by learning a language field from 2D CLIP features analogously to how NERFs learn color fields, enabling real-time querying and rendering capabilities.

LangSplat [10], on the other hand, combines 3D Gaussian Splatting with language features extracted using Segment Anything Model (SAM) techniques. This hybrid approach hierarchically crops parts of images and feels them into CLIP, compressing resulting features with an autoencoder. The efficacy of LangSplat lies in its optimization of language features through rendered comparisons with CLIP features.
However, scaling Gaussian Splatting to large urban scenes is an active research area in its own right. Also, LangSplat requires feature compression to seven dimensions or less out of memory constraints. This is undesirable for the nuanced analysis of social dynamics we want to perform.

In our approach, we adapt LangSplat's hierarchical feature extraction with OpenScene's point cloud-based scene representation. Using a sparse scene model instead of Gaussian Splatting enables us to analyze the full, uncompressed features at the cost of accurate geometries.

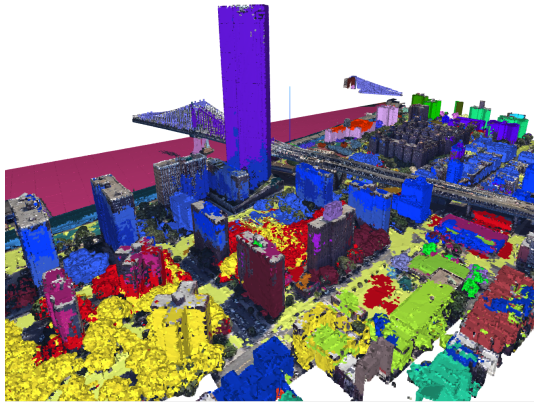

Figure 2: Unsatisfying segmentation result of Mask3D on an urban scene.

We furthermore experiment with SigLIP [15] 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 preparation

As data source we have 3D meshes obtained from google earth [1] and we want to obtain a corresponding point cloud with language features attached. To do so we first generate RGB and depth images of the city by rendering the mesh from multiple perspectives scanning the whole mesh.

In particular we generate 3 classes of images: satellite-like were the camera is set up in the air looking vertical, aerial where the camera is set a little above the buildings looking between 30 and 60 degrees and street view where the camera is a couple of meters over the points looking horizontal.

We then used SAM [7] on the RGB images using 4 hierarchies since it has excellent performance in segmenting 2D images. For each obtained segment we cut out a corresponding image patch, in which we highlight the segmented area. Highlighting is performed by partially whitening out the non-segmented area and marking the border of the segment with a red line. Refer to Figure 3 for a visualization. Since most segments only cover a handful of pixels we only retain the ones covering at least 0.25% of the image. This leads to the removal of roughly 60% of all segments. We also add the embedding of the entire image as a 5-th hierarchy.

Finally, we run each of the highlighted segment images through SigLIP to get one embedding per segment.

Note that our feature extraction pipeline is identical to LangSplat's with three major differences. Firstly, we do not completely cut out each segment but instead present it highlighted in its original context, as we found that shapeless urban structures are otherwise hard to identify even for humans. Secondly, we use SigLIP instead of CLIP as an embedding model. Thirdly, we have one more hierarchy: the coarsest one.

To project the 2D features to 3D, we average the embeddings of all segments in which the relevant point was observed. This results in a point cloud where each point has a SigLIP embedding attached, which we can use for prompt-based interaction.

An illustration of the pipeline can be seen in Figure 1.a.

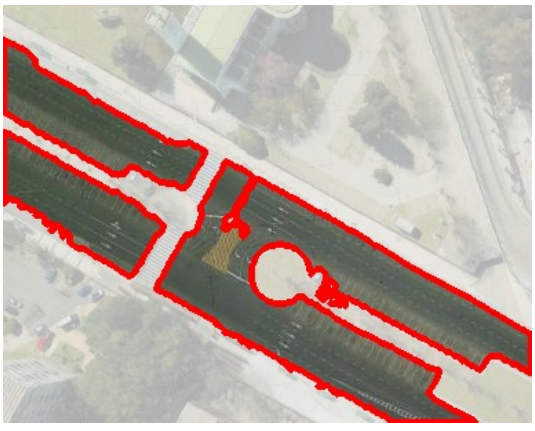

Figure 3: Example of segmented RGB image with highlight

### 3.2. Language-guided zero-shot Scene Interaction

Based on this enriched point cloud of a city scene it is possible to infer point-wise similarity scores to a given prompt by using the text embedding $\phi_{query}$ obtained from the SigLIP text encoder. We follow LangSplat in this regard and compute the dot product between the point and the prompt embedding, optionally contrasted with negative queries:

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

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

Similarity scores can be rendered as a heat map for visualization, as illustrated in Figure 1.b, or projected to 2D to be correlated with ground truth map data. However, they only provide relative estimates between zero and one. To perform actual regression, we use nearest-neighbor classifiers as described below.

### 3.3. Few-shot learning by KNN Regression

Alternatively, the point embeddings can be used as features for few-shot learning. To that end, we assume we have ground truth data for a part of the scene and construct a K-Nearest-Neighbors (KNN) regressor from it, which we then use to perform inference on points from unseen parts.

Notably, we perform the train/test split at the coarse granularity of regions (houses, neighborhoods, roads, etc.) but otherwise operate on the finer granularity of single points and their embeddings.

The number of neighbors was set to 50, each equally weighted and selected with respect to cosine similarity.

## 4. Datasets

Three scenes were extracted from locations chosen based on the availability of ground-truth data and 3D meshes.

### 4.1. Rotterdam and Amsterdam

We extract Google Earth [1] meshes for the Dutch cities Rotterdam and Amsterdam, each covering 1-2km$^2$. They feature the Rotterdam University and the Amsterdam Central train station area, respectively.

Corresponding ground-truth data was taken from the 3D BAG API [9], which provides among other things 2D building footprints and associated construction years as given by the cadastre for the entire Netherlands. To the best of our knowledge, this database is unique in its granularity and size, providing valuable ground truth data for an interesting prediction task.

### 4.2. Buenos Aires

We extract one larger scene of roughly 4km$^2$ covering the Buenos Aires city center. Along with it, we use official records from the Autonomous City of Buenos Aires (CABA) of population count [2], crime records [3], and urban noise emissions. [4]

**Population density**: We directly compute the density by dividing the number of residents by the area at the granularity of neighborhoods. The data is from the years 2015-2018.

**Crime rate**: CABA provides locations and descriptions of all recorded crimes between 2016 and 2022 [3]. 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, and lighter offenses such as traffic incidents. This leaves us with a dataset of 2146 out of 32609 crimes within the selected area. To avoid artifacts at region boundaries and attenuate sparsity effects, we consider each crime a 2D Gaussian distribution with a standard deviation of 50m. Then we sample from it to compute the expected number of annual armed crimes per square kilometer and neighborhood.

We point out that this does not correspond directly to the actual crime rate, but rather represents an indicator of danger, which we believe to be a more sensible quantity to predict.

**Noise Pollution**: The CABA noise emission dataset [4] provides a map of estimated average daytime noise along major city roads. It divides the roads into twelve linearly spaced noise intervals based on estimated noise in decibels. To compute meaningful correlations, we take the mean of each interval as the ground truth value.

A visualization of the extracted area and its core characteristics can be found in the appendix. Also note that only the coarsest feature level of this scene is processed, as we are most interested in information whose geometrical scale matches the scale of the available ground-truth data.

## 5. Experiments

### 5.1. Building segmentation

Given the point cloud with associated features, we perform zero-shot classification of the points into the classes *building* and *background*. We use *building* as a positive query and a set of canonical queries representing common urban objects (listed in the appendix). The resulting similarity score is interpreted as a probability. 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 3D BAG dataset.

We find that this classifier attains an ROC-AUC score of 0.927 in Rotterdam, accompanied by an accuracy of 87.5% given an appropriately chosen threshold. This is a significant improvement compared to LangSplat-style features projected to the same point cloud. The latter achieves only up to 79.8% accuracy with a ROC-AUC of 0.862, as visualized in Figure 4.

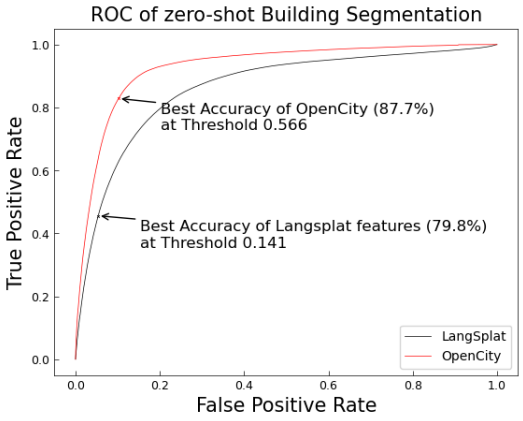


Figure 4: Zero-shot building segmentation results based on OpenCity features (black) and LangSplat features (red).

### 5.2. Building Age

Given the point cloud and point embeddings, we predict the construction year of Dutch buildings in a zero-shot setting. We predict age scores by feeding the positive prompt *modern building* contrasted with the negative *old building*. Then we again project the points to two dimensions and resample them to a regular grid. Each point within a building is then assigned a ground truth construction year, all other points are omitted.

The Spearman correlation between the age scores and the construction year is 0.507 for the Amsterdam scene, and 0.556 for the Rotterdam one.

To emulate a few-shot setting, we furthermore split the buildings into 30% training and 70% validation samples to train a KNN classifier for each city as described above on the embeddings. This results in a higher correlation of 0.73 and an R2 score of 0.51 for the Rotterdam scene. For Amsterdam, the correlation increases to 0.53 but the R2 score is very low at 0.05.

We furthermore find that OpenCity outperforms LangSplat-style features, which achieve lower correlation on the task. Refer to table 1 for a summary.

### 5.3. Population Density

Given the embedded point cloud, we aim to predict the population density in Buenos Aires in a zero-shot setting.

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,

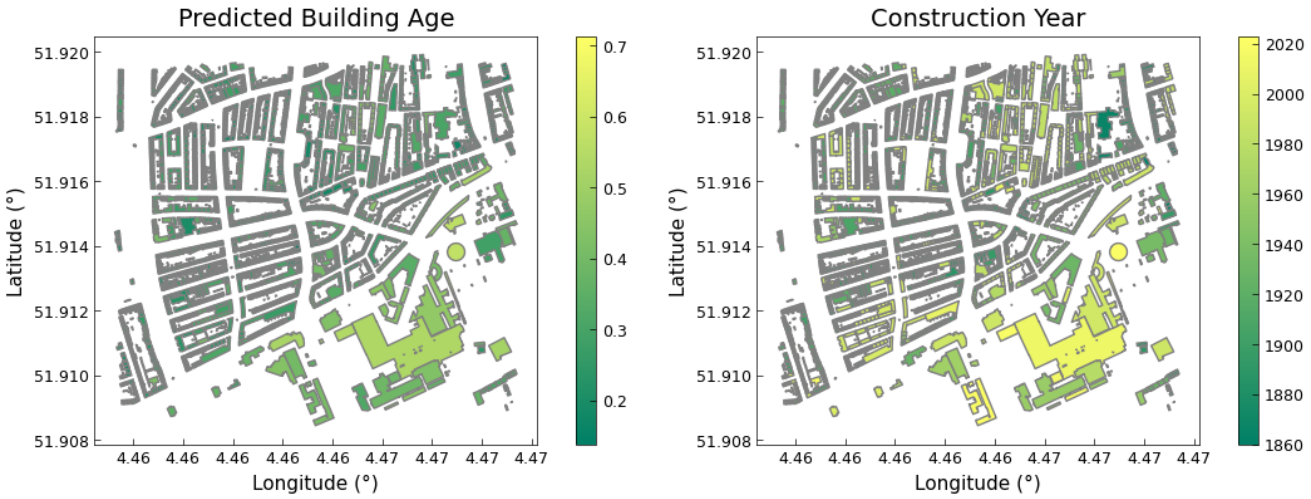


Figure 5: Result of the building age prediction of OpenCity. Prediction on the left, ground truth of Rotterdam on the right.

Table 1: Comparison of various feature extraction methods evaluated on the Rotterdam Scene. For LangSplat, the uncompressed, point-projected features were evaluated. The best results are highlighted.

| Method | Age Correlation | Building seg. max accuracy |
|---|---|---|
| LangSplat | 0.394 | 79.8 |
| OpenCity (prompt) | 0.556 | **87.7** |
| OpenCity (KNN) | **0.733** | N/A |

Table 2: OpenCity result summary for various prediction tasks on the Buenos Aires scene.

| Task | Prompted (Spearman) | KNN (Spearman) | KNN (R2 score) |
|---|---|---|---|
| Population Density | 0.63 | 0.61 | 0.29 |
| Crime Rate | 0.42 | 0.61 | 0.18 |
| Noise Level | 0.19 | 0.50 | 0.11 |

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 appendix for visualization). However, it erroneously assigns comparably high scores to the city center south of the train station. It also over-estimates the population in the industrial port area to the northeast.

The latter can be improved by adding prior information to the model. With the two additional negatives *nature* and *industrial area*, the correlation is boosted to 0.753. We hypothesize that this effect is due to the ambiguity of the term *dense population* in the context of an image of a natural shoreline or a container ship.

We again evaluate the features in a few-shot setting by using 28 training and 94 validation neighborhoods to train a KNN classifier. This results in a similar correlation of 0.61 and an R2 score of 0.29.

### 5.4. Crime Rate

We use our embeddings to predict the expected number of crimes per square meter in the city of Buenos Aires.

The used prompt consists of the positive query *dangerous neighborhood* and the negative *safe neighborhood*. Using this indicator, we obtain a Spearman correlation of 0.301.

The task mainly consists of identifying the port-facing side of the north-western district as a dangerous area. For a visualization refer to the appendix. Notably, the model again 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 0.422 by adding *nature* as a negative prompt. In this case, however, this prior is less easily justified, as large city parks do not generally come with lower crime rates - though the mere absence of people may indicate such a tendency.

When evaluated in the aforementioned few-shot setting, KNN classification results in a correlation of 0.61 and an R2 score of 0.18, indicating that the information is only represented in the features to a limited extent.

### 5.5. Noise Emissions

We follow the same approach to predict urban noise levels as given by the City of Buenos Aires, using *noisy ur-*

Figure 6: Showcase of difficulty of determining the age of houses using low-resolution images. The building on the left is from 1907 and the right one is from 1997. They are from the scene of Rotterdam.

*ban area* as positive, contrasted with *quiet area* as negative. This gives us a weak Spearman correlation of 0.19. Visual results are again presented in the appendix.

In the few-shot setting, we use roughly a third of the areas (207 out of 691) as training data for the KNN classifier. This yields a moderate correlation of 0.500 and a low R2 score of 0.11.

## 6. Discussion

The OpenCity framework has demonstrated significant potential in advancing the understanding of complex urban environments through 3D scene understanding combined with language embeddings. It provides a proof-of-concept for image-based understanding of urban dynamics at a larger scale, which we believe to be an interesting future research topic.

### 6.1. Building Age and Segmentation

The OpenCity framework has demonstrated significant potential in advancing the understanding of complex urban environments through 3D scene understanding. The new features outclassed the LangSplat features by a large margin. The method is able to differentiate regions with more modern architecture from more traditional areas as can be seen in Figure 5. The method has a harder time differentiating newer houses from older ones if they are in a dense cluster together. This is not necessarily an issue of the method, but can also be a problem with the data. Often newer houses are built to match the style of the preexisting neighborhood, as seen in the example shown in Figure 6.

### 6.2. Population Density Estimation

Our framework has shown reasonable results in estimating population densities. This outcome is expected given the nature of the corresponding urban features—where the number and size of residential buildings often have a very clear influence on the population density.

### 6.3. Crime Rate and Noise Level Prediction

The difficulty in predicting crime rates and noise levels from 3D data illustrates the complexity of urban dynamics, where many influential factors are not immediately visible or quantifiable through spatial analysis. This finding indicates the need for a more nuanced approach that incorporates a broader range of data types. Having reference values, like in the KNN version, greatly increased the quality of the results, especially for noise levels.

## 7. Limitations

An overarching issue in the area of large-scale outdoor open vocabulary methods is the lack of benchmarks and datasets. Most of the field focuses on smaller areas, mostly inside buildings. As a consequence of that, one is forced to find problems to solve for which data is available, making the generalizability of the results an open question. We hope in the future, more benchmarks will be published.

Due to limited time and computational resources, the experiments are not as extensive as they could have been. In particular, our experiments have not made clear how much of the gain over LangSplat is obtained by using SigLIP instead of CLIP.

Also note that this approach was developed as a simple, exact-as-possible method to analyze the information content of VLM features. The scalability of the method is limited by the memory and storage taken up by storing the full point features, as we - unlike other methods - do not use compression.

## 8. Conclusion

We adapt existing methodologies to analyze city-scale datasets, focusing on detecting urban characteristics such as population density, building age, crime rate, and noise pollution. Our findings suggest that VLMs exhibit significant potential in urban scene understanding. By extending these techniques on a city scale, we introduce OpenCity, which utilizes Google Earth mesh data to generate an embedded point cloud via rendered RGB-D images. This allows for a comprehensive analysis of urban dynamics, enhancing our understanding of the city. Although predictions for crime rates and noise emissions remain less robust, our approach demonstrates considerable promise for advancing urban analysis efforts.

## References


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

[2] Bits and Bricks. Buenos aires population density. https://bitsandbricks.github.io/data/CABA_rc.geojson, 2024. Accessed: 2024-06-15. 4

[3] Buenos Aires City Government. Buenos aires government open data portal, 2021. Accessed: 2024-05-18. 4

[4] Buenos Aires City Government. Buenos aires government open data portal, 2021. Accessed: 2024-05-18. 4

[5] 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, 8

[6] 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

[7] 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. 1, 2, 3, 8

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

[9] 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. 3

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

[11] 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, 8

[12] 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. 8

[13] 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. 1, 8

[14] 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, 8

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

# Appendix

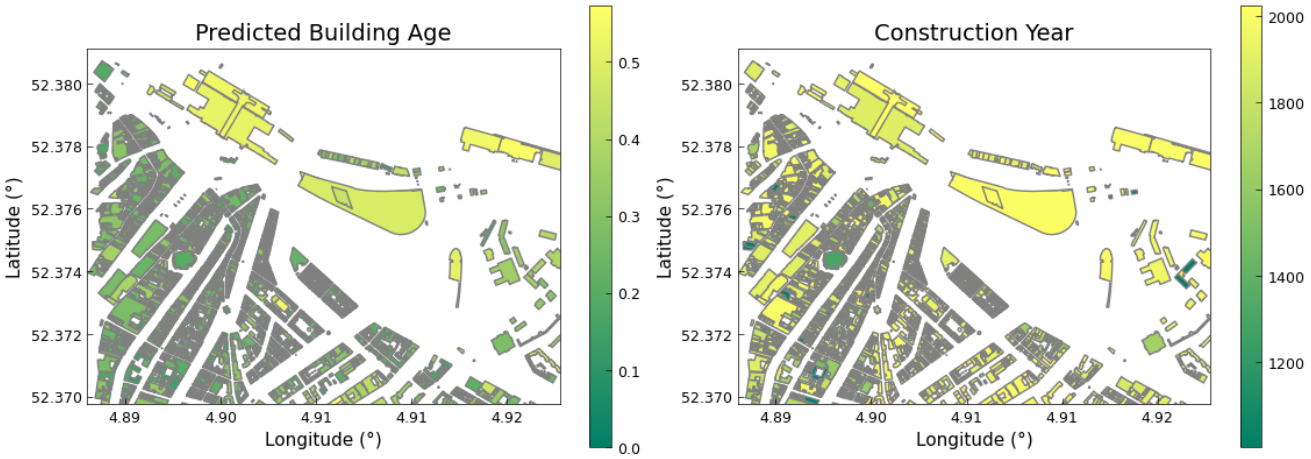


Figure 7: Result of the building age prediction of OpenCity. Prediction on the left, ground truth of Amsterdam on the right.

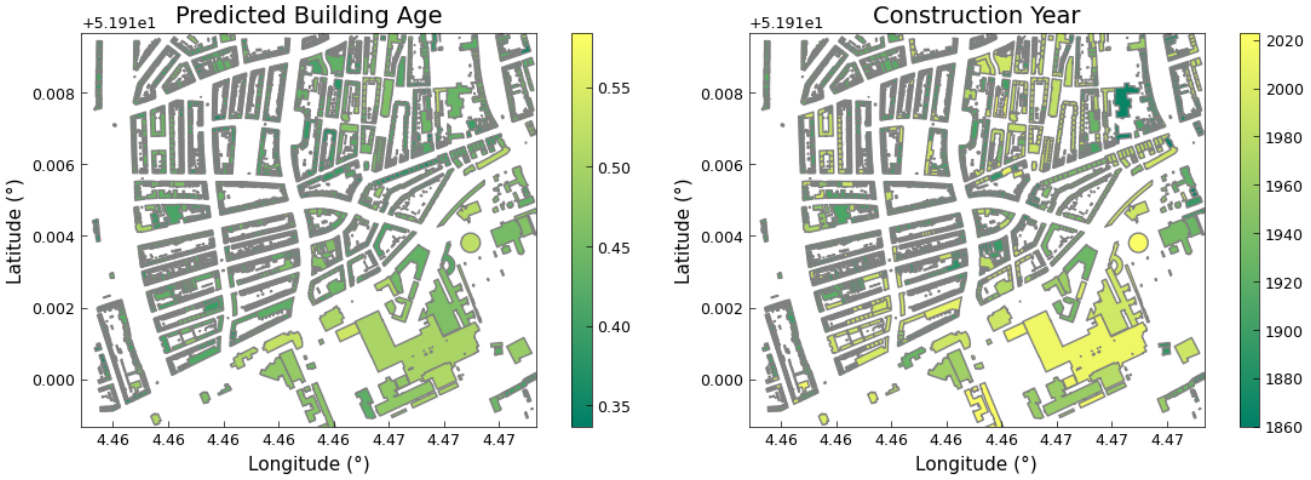


Figure 8: Prediction result (left) vs. ground truth (right) of the building age prediction of Rotterdam using LangSplat.

**Analysis of OpenMask3D for urban point clouds**

OpenMask3D [14] is a method that leverages CLIP [11] embeddings to extend Mask3D [13], a model for 3D semantic instance segmentation, on an open vocabulary. To do so it uses SAM [7] 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 massively improves efficiency and storage usage, both important factors for scalability.

While we did try to use this approach on a city scale it unfortunately fails because one of the backbones: Mask3D, isn't able to generalize to city data. Mask3D is pre-trained on ScanNet200 [12] which is highly detailed and diverse for indoor scenes thus making it a good fit for OpenMask3D experiments, but on city scale data it fails at generating any meaningful masks. We also try to use a Mask3D model trained on STPLS3D [5], a synthetic urban dataset, and use Segment3D [5], an alternative to Mask3D, but with no good results.

**Negative Prompts used for Building Segmentation**

We use the positive prompt "building" against the empirically chosen negatives "tree", "road", "park", "river", "car", "sea / lake / canal", "urban scene", "parking lot", and "city".

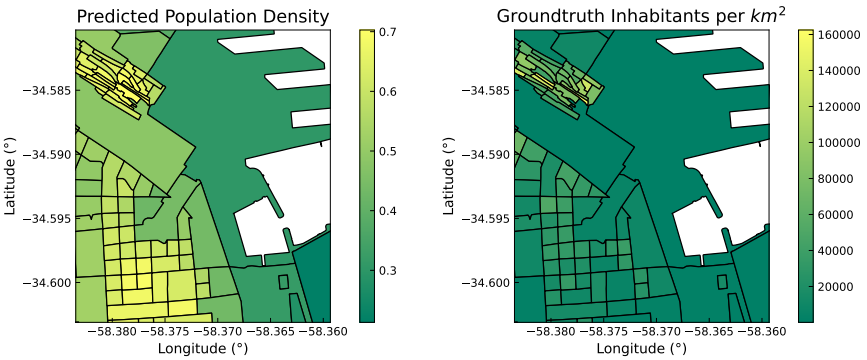


Figure 9: Prediction (left) vs. ground truth (right) population density in Buenos Aires.

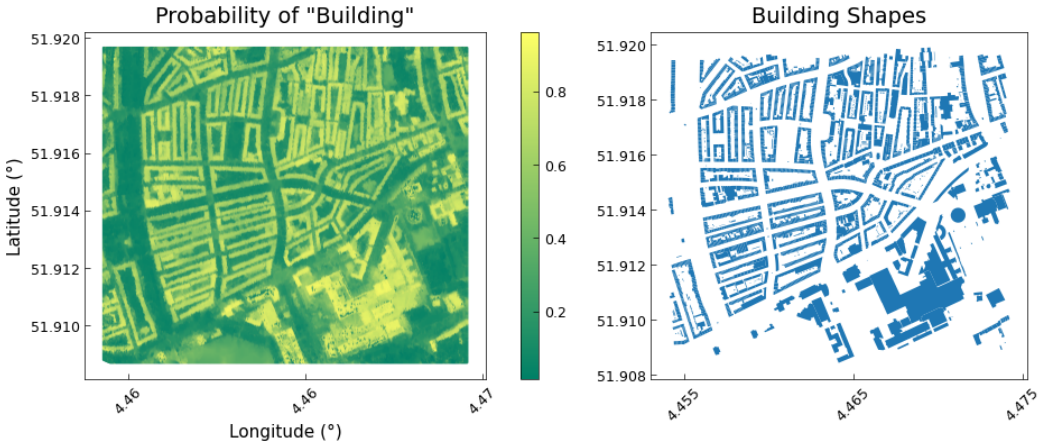


Figure 10: Predicted building probabilities of OpenCity (left) vs. ground truth (right) building outlines of the Rotterdam scene

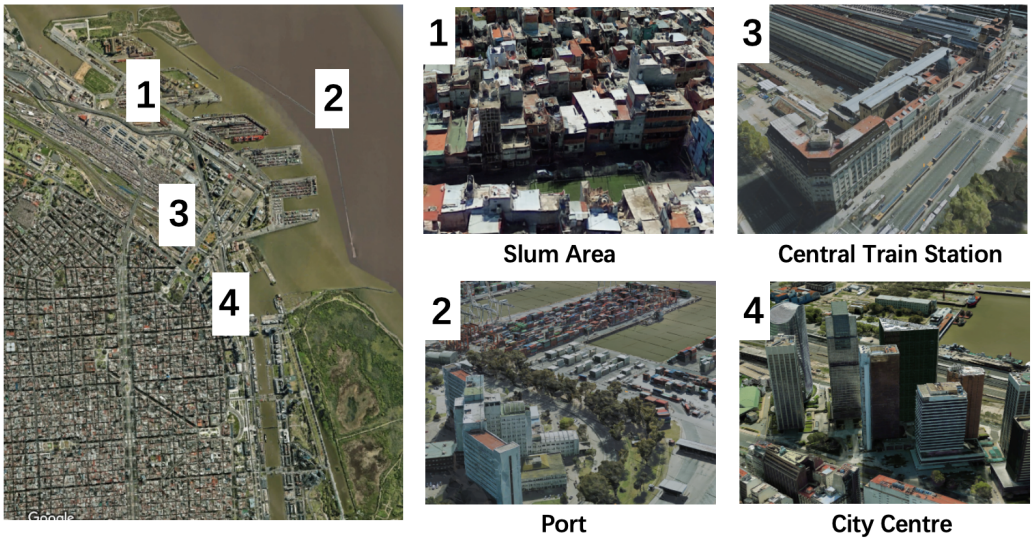


Figure 11: Illustration of the Buenos Aires extract. Most notably, we have the *Barrio 31*, a *villa miseria* (slum) in the northwest. East of it is a large industrial port. In the center is the train station with tracks going north-west. South of them is what is commonly considered the city center.

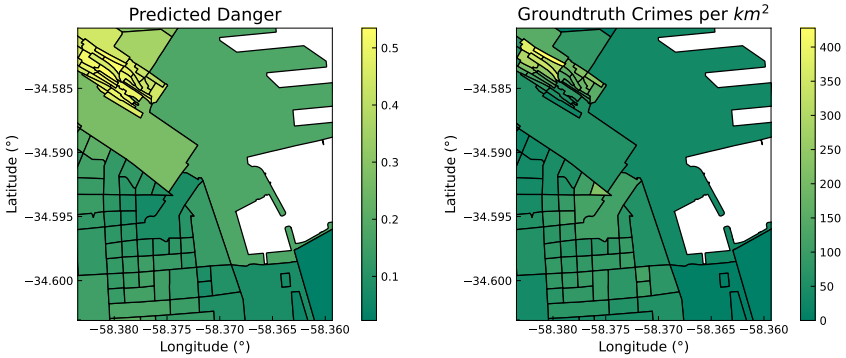


Figure 12: Prediction (left) vs. ground truth (right) crime occurrence in Buenos Aires.

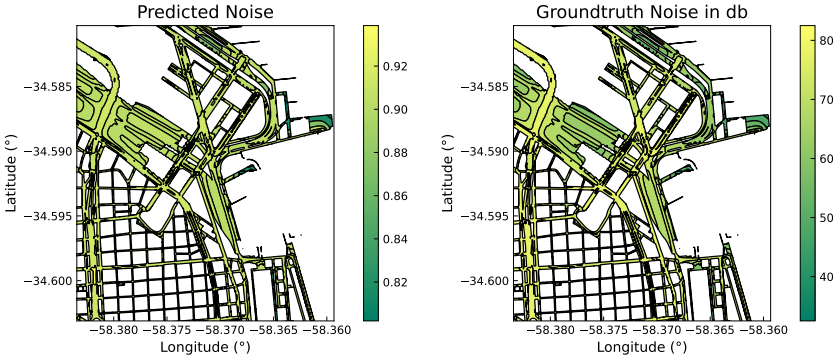


Figure 13: Prediction (left) vs. ground truth (right) urban noise level in Buenos Aires.