npj | urban sustainability

Article

Published in partnership with RMIT University

https://doi.org/10.1038/s42949-026-00369-2

AI-based urban layout generation model

Check for updates

Liu He[1], Harsh G. Kamath[2], Songlin Fei[3], Dev Niyogi[2] & Daniel Aliaga[1]

Accurate urban geometric layouts are critical for urban planning, simulation, and sustainability management; however, no ideal solution currently exists for large-scale urban geometric layout creation. We propose a generative AI-based urban layout model capable of encoding arbitrary 3D city blocks to a unified latent representation and generating urban layouts for the 330 cities in North America that have over 100,000 inhabitants. Given only a few percentage of the city blocks, our approach is able to generate an entire realistic 3D city, to support street-scale physical simulations, to predict social-economic metrics, and to enable "what-if" scenarios for policy-making. Compared to data-driven approaches, ours provides a unified representation suitable for large-scale urban simulation and policy-making as part of a digital twin framework, which would otherwise require a time- and resource-consuming process of obtaining the position, geometry, and height of all buildings individually.

Cities are important, highly populated, and complex structures of the modern era. In fact, the current urban population exceeds 4.6 billion people (57% of the global population) and is predicted to reach 7.1 billion by 2050. In the US alone, the urban population is spread amongst 20,000 incorporated towns and cities. Organizing the growth and development of these complex and often large places is of critical importance, but often overwhelming. Although a daunting process, creating a digital representation of the layout of a city (i.e., roads, city blocks, trees, and buildings) is fundamental to urban simulation, modeling, and prediction.

State-of-the-art approaches to create a digital twin and representation of cities are based either on manual modeling (e.g., crowd sourcing and manual modeling efforts such as OpenStreetMap (OSM)[1]), remote sensing (e.g., satellite/aerial/ground imagery and LIDAR scanning[2,3]), or procedural/generative modeling. This latter approach has a long history. The seminal paper of Parish et al.[4] laid the foundation for urban procedural modeling, including the generation of building masses[5], urban layouts[6–8], and parcel generation (e.g.,[9], Esri CityEngine). In addition, urban-related layout approaches have also been developed for building interiors[10–17], building structures[18], and houses[19]. However, existing research disproportionately targets a small set of prominent large cities (e.g., New York, Chicago, Paris, Beijing, and Tokyo), leading to significant geographical and representational bias in available urban datasets. Most medium and smaller cities are often missing geometry data in city layout due to the lack of airborne/spaceborne and incomplete OSM data (e.g., OSM provides less than 20% completeness for half of the urban population[20], and global building height dataset are typically a grid-based coarse mapping which fails to capture fine-grained urban structures[3]); even for cities with geometric layouts, they are mostly incomplete or hard to use. Moreover, repetitively re-creating urban layouts for simulating urban futures and what-if scenario generation, as part of a digital twin framework, for all US cities, let alone the world, is extremely time-consuming. Currently, *there is no ideal solution for large-scale urban geometric layout creation, and in all cases, producing complete, detailed, and accurate layouts is invariably tedious.*

To create and modify urban layout geometric data, we developed a novel AI-based urban layout generation model for current or future urban layouts requiring only a very small fraction of a priori data (e.g., less than 5%). Our model not only changes the methodology for bootstrapping urban planning and simulation from capture to generation, but also facilitates three important tasks:

- Generation without capture: creation of realistic urban layouts without explicitly requiring detailed capture, and thus suitable for previously unseen areas.
- From geometry to indicators: similar to the fact that humans can observe a neighborhood and infer properties about it, our approach can infer social-economic properties of an urban layout solely by inspecting the latent space vector of an urban layout.
- Indicator-oriented layout editing: our method provides intuitive ways to edit an urban layout to support what-if urban scenario configurations.

Our self-supervised learning method uses a graph-based variational autoencoder (Fig. 1a) without the need for manual labels. Our model is trained and applied to all North American cities with a population of more than 100,000 people (i.e., 330 cities). The available building footprints (e.g., OSM, Microsoft) and building-level height information from the UT-GLOBUS system[21] are automatically collected as our training dataset, totaling 17 million building footprints that span 5 million city blocks. The 3D building geometry, position, and shape features of each city block are represented as a sparse vector-based graph, which greatly reduces computing and memory resources during training and generation. Training for our model was performed in parallel on a cluster of 100 NVIDIA A100

[1]Department of Computer Science, Purdue University, West Lafayette, IN, USA. [2]Department of Earth & Planetary Sciences, Maseeh Department of Civil, Architectural, and Environmental Engineering, and Oden Institute of Computational Science & Engineering, University of Texas at Austin, Austin, TX, USA. [3]Department of Forestry and Natural Resources, Purdue University, West Lafayette, IN, USA. e-mail: aliaga@cs.purdue.edu



Fig. 1 | Urban layout generation. a The processing pipeline of our generative framework. **b** A sample of generation results in Pittsburgh and Miami – the left side of slanted line in each image is ground truth and the other part is automatically generated by our method. **c** Additional generated examples for neighborhoods in Tampa, Denver, Baltimore, and Phoenix.

GPUs. Afterward, quick generation and/or modification of an entire large city as well as smaller cities can be performed in 3–5 min by a single desktop (or laptop) GPU (e.g., Fig. 1b, c) with an average reconstruction error of near 1% in terms of building position, size, and count.

Results

We successfully show that a pre-trained deep network is able to condense knowledge of nationwide 3D urban footprints into a distribution of a 512-dimensional latent space. Thus, augmenting or even in the absence of remote sensing data, urban layout can be generated automatically, given only a road network and an extremely small portion (e.g., 5%) of a priori city layout. While several prior works generate pixel-based outputs representing large urban areas (e.g.,[22–24]), they do not generate the geometry of individual buildings and their heights. Further, we demonstrate that our generative approach provides a unified gateway to generate urban geometric descriptions without capture suitable for weather planning (e.g., urban meteorological simulation), for social-economic indicator prediction, and for intuitive indicator-based editing, which is provided by prior methods.

Generate without capture

In order to overcome the time and expense of explicitly requiring the footprint, height, and position of each building in a city, our approach enables a deep urban extrapolation to produce the entire layout of a city from only a small fraction. This is possible because urban spaces are not a random collection of structures but rather have an organized, though complex, set of rules that govern urban spaces. These rules are what urban planners historically use and develop in their design process. However, we show that the layout rules can be extracted from a small but broad set of samples. In particular, our deep urban extrapolation is able to infer this information, after training, also using only a small fraction of a previously unprocessed urban location. This alternative machine learning based methodology to obtain the urban morphology greatly facilitates and accelerates urban planning and simulation processes.

We performed a deep extrapolation for the 330 North American cities with a population over 100,000. Figure 2a, b shows the result of several experiments that vary the amount of data used during training and inferencing. The first graph in Fig. 2a shows the reconstruction errors when comparing generated results to the training data at different training set

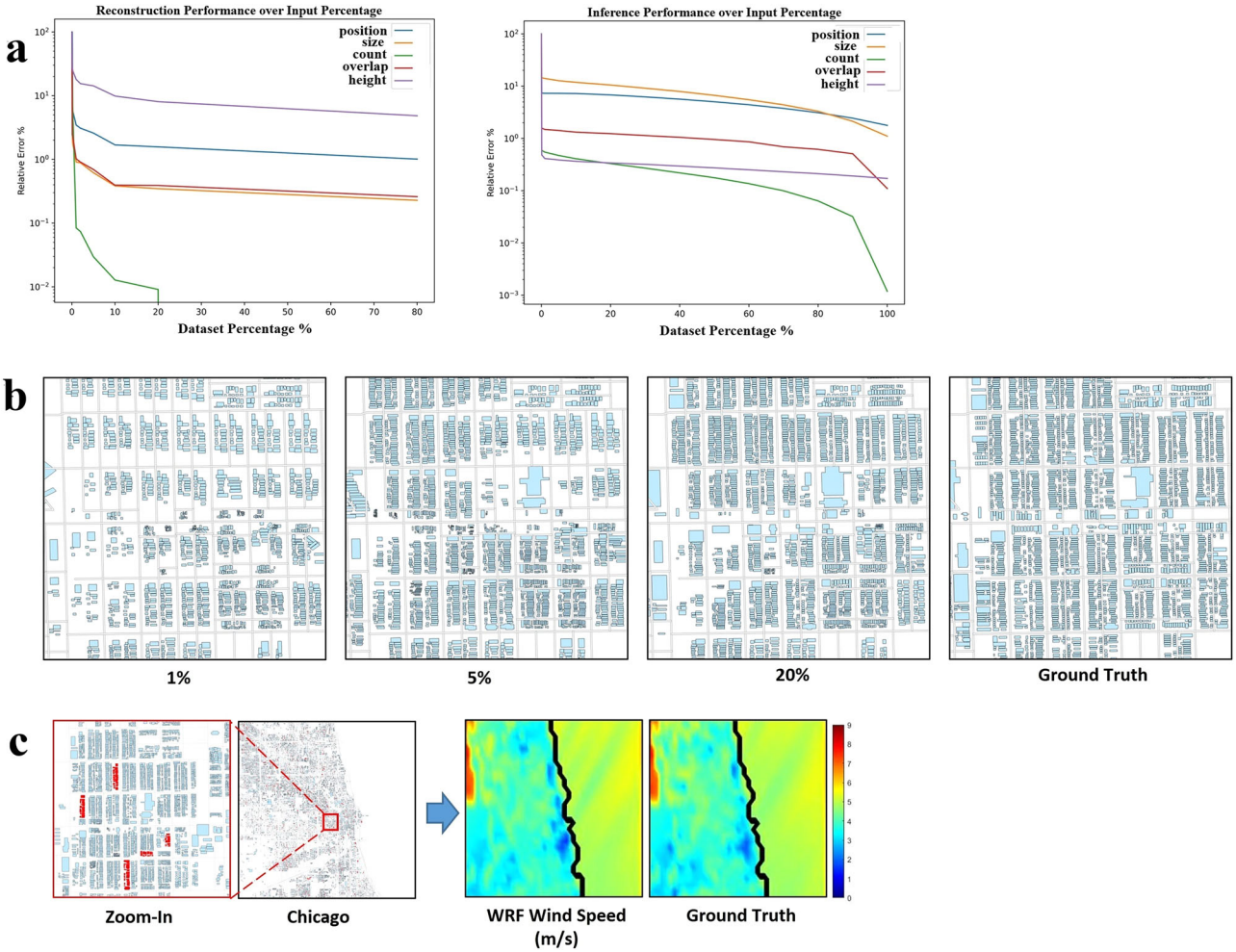

Fig. 2 | From generation to capture. a The reconstruction accuracy over the input dataset percentage. Given 5% dataset or more, we obtain a reasonable reconstruction accuracy. **b** The inference accuracy (i.e., ability to infer missing buildings) over the percentage of given city blocks for inference also works given only 5% or more of the dataset. **c** The visualization sample of deep extrapolation in Chicago (red blocks are given, blue blocks are generated) and a Weather Research and Forecasting (WRF) simulation comparison.

percentages—in summary, building position, size, and count have about 1% error when using as little as 5% of the dataset and further reduce for higher dataset percentages. At the same 5% of the dataset, building height error is near 10% and it reduces to 3% at higher dataset percentages. In a similar fashion, the inference ability (i.e., the ability to recover omitted buildings) of building height and count is near 1% starting at 3% of the dataset and further reduces at higher dataset percentages. Building position and size inference is near 10% for 3% dataset size and reduces at higher dataset percentages to 2%. As Fig. 2b shows, we exploit that the 512-D representations of all blocks in a city is smoothly distributed in a high-dimensional manifold. This smooth distribution feature enables a small portion of randomly sampled city blocks to represent the distribution of the entire city. Given that, we may extrapolate the distribution of missing city blocks in a city by a simple weighted distance function. For example, at around 5% dataset size, the reconstruction and inference ability are quite good.

Meteorological modeling application

Figure 2c demonstrates using the layout generation ability to drive a Weather Regional Forecasting (WRF) simulation for wind speed simulation. We show an example comparison of such WRF simulations for a portion of Chicago. Thus, we can use our generative AI system to produce an urban layout suitable for urban weather simulation. Further improvements could be accomplished by linking in sunlight and shadow estimators as well, for example[25].

From layouts to sustainability/hazard indicators

In the quest towards creating city digital twins, social, economic, and environmental indicators (e.g., population, income, air quality concerns) play a pivotal role in the assessment of urban sustainability and hazards. Obtaining such indicators relies on a lengthy household survey process or statistical extrapolation from outdated census data. Remarkably, our innovative deep modeling approach demonstrates the ability to derive urban indicators based solely on the geometry of buildings and city blocks in a census tract. This holds substantial benefits, as it offers a proxy to the need for lengthy and costly survey campaigns when seeking quantitative indicator values for new urban regions in real-world scenarios or what-if simulations. Notably, much like humans do upon physical inspection, our approach also unveils the profound implication that specific urban layout patterns are indicative of corresponding social and economic conditions.

Our approach supports predicting high (or low) metric values for a set of diverse 100 urban metrics derived from census tract data. First, we calculate for each city block the values of these 100 metrics. Then, for each metric, we extract the city blocks in the lower 1st percentile and the city blocks in the upper 99th percentile. Empirically, we observe that for each metric, there are 10,000 or more city blocks. We use the extracted city blocks to train a per-metric classifier to determine if a block is in the high-value or low-value group. The classifier takes in the 512-D latent vector of a city block and yields a binary classification (for the corresponding metric). We employ conventional classification algorithms, including random forest, naive

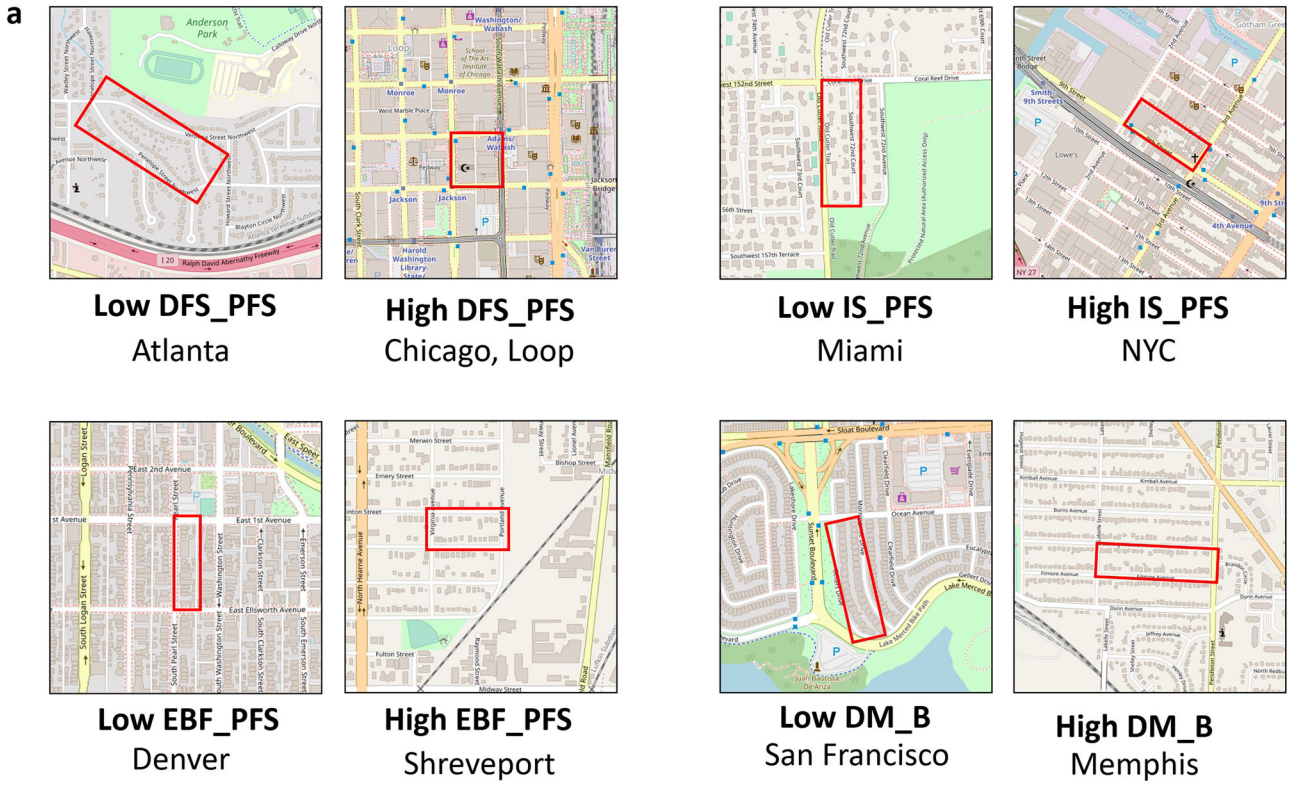

Fig. 3 | OSM layout fragments with predicted indicator values. Example of fragments of cities and the (correctly) predicted indicator value (high or low) as per our method. The relevant city is referenced below each image.

Table 1 | Layout to Indicator Prediction

Abbrv.	Metric full name	Acc. % (SVM)	Acc. % (XGBoost)
DSF_PFS	Diesel particulate matter exposure (percentile)	89.76	**91.04**
IS_PFS	Share of the tract's land area that is covered by impervious surface or cropland as a percent (percentile)	88.16	**89.42**
TSDF_PFS	Proximity to hazardous waste sites	84.36	**85.29**
EBF_PFS	Energy burden (percentile)	**84.26**	82.08
LMI_PFS	Low median household income as a percent of area median income (percentile)	**83.62**	80.23
LLEF_PFS	Low life expectancy (percentile)	**80.30**	78.55
EBLR_PFS	Expected building loss rate (Natural Hazards Risk Index) (percentile)	**80.13**	75.57
LPF_PFS	Percent pre-1960s housing (lead paint indicator) (percentile)	**79.83**	78.23
DF_PFS	Diagnosed diabetes among adults aged greater-than or equal-to 18 years (percentile)	**77.98**	75.80
HBF_PFS	Housing burden (percent) (percentile)	**76.82**	72.46
UST_PFS	Leaky underground storage tanks (percentile)	**75.91**	73.82
EPLR_PFS	Expected population loss rate (Natural Hazards Risk Index) (percentile)	**75.82**	74.04
P100_PFS	Percent of individuals <100% Federal Poverty Line (percentile)	**75.46**	71.67
FLD_PFS	Share of properties at risk of flood in 30 years (percentile)	**75.46**	72.97
TF_PFS	Traffic proximity and volume (percentile)	**75.15**	73.48

Using our latent-space representation and a classifier, our method can categorize a city block as low-valued or high-valued for a large number of diverse economic, social, and environmental indicators. We show the top performing indicators with accuracies in the 90 to 75% range.
Bold values means the value is that column is the greatest of the two columns (i.e., the best value).

Bayesian, support vector machine (SVM) and XGBoost to do binary classification for each metric, with SVM performing best. Remarkably, the use of our 512-D latent space and the above simple classification scheme yields a powerful mechanism to predict urban metric values based solely on the urban layout geometry (Fig. 3).

In Table 1, we present 18 metrics with 90% to 75% classification accuracy in descending order. Notably, the latent representation of block 3D geometry adeptly captures and predicts metrics related to critical domains such as air quality, household income, vulnerability to weather hazards, and land use, for instance. This predictive capability is a testament to the inherent value encoded within the geometric essence of urban blocks. SVM delineates an explicit linear boundary within the high-dimensional manifold to distinguish between the two groups. This compelling result paves the way for an additional layer of exploration, where we leverage these boundaries to uncover distinctive clusters within the latent space. Subsequent analysis of city blocks residing within these clusters indicates intriguing pattern variations of urban metrics.

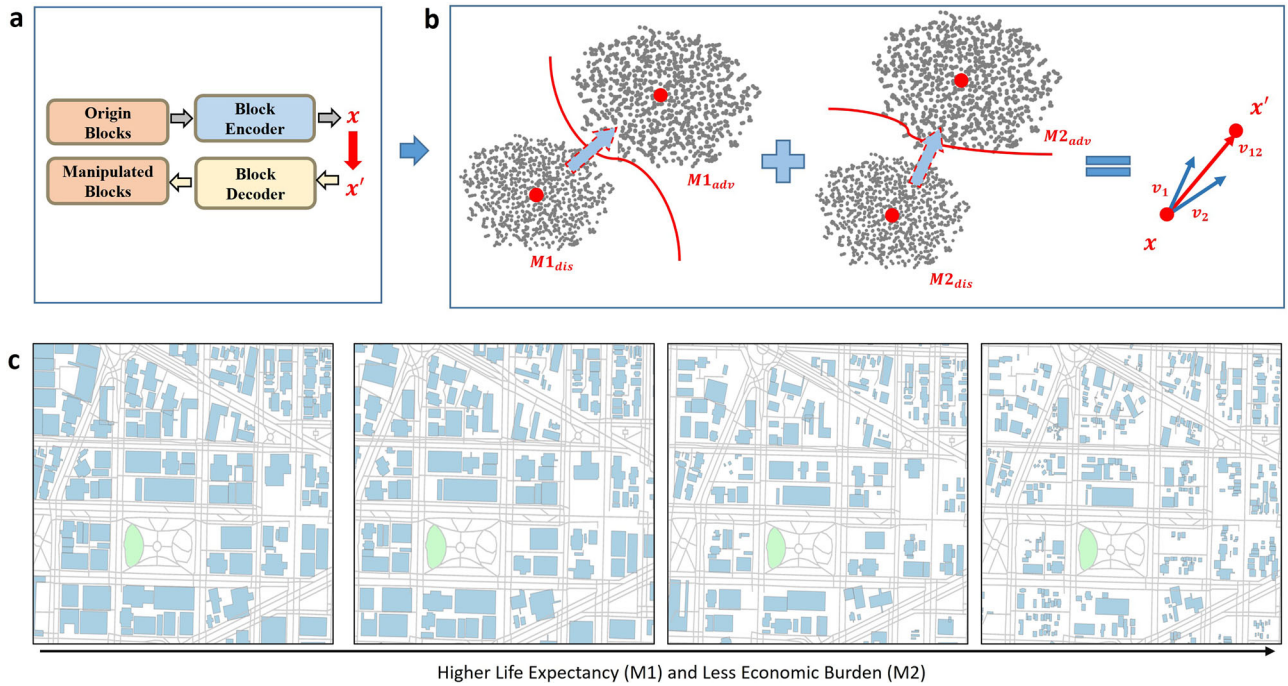

Fig. 4 | Indicator-oriented urban modeling. a The indicator-oriented manipulation converts encoded latent vector x to x' following the direction of a manipulation vector. Then, urban blocks corresponding to x' are generated by our block decoder. **b** For example metrics $M1$ and $M2$, the binary classification described in the text separates city blocks into low and high-value indicator clusters in the high-dimensional latent space. Our indicator-oriented manipulation vectors (v_1, v_2) guide an interpolation from low to high values, or vice versa. The manipulation vectors for the two metrics seem to be independent, thus they can be combined and still preserve their respective layout styles. **c** An example showcasing the usage of indicator-oriented manipulation to alter the initial layout to one of higher life expectancy and less economic burden.

Indicator-Oriented Urban Modeling

Urban design and modeling have long followed the guidance of established policies, budgets, and planning principles. However, the ultimate social, economic, and environmental outcomes remain uncertain. Our generative method introduces a novel approach to urban modeling that enables fast and responsive indicator-oriented urban layout editing. Regardless of the initial urban context, our framework empowers decision-makers with the potential to chart more favorable configurations and better achieve targeted outcomes for their cities.

Our indicator-oriented urban modeling relies on vector manipulation in latent space. As Fig. 4a shows, an input block will be encoded, manipulated, and decoded to produce a modified urban layout. The latent vector manipulation is illustrated by Fig. 4b. Using the binary classifications and 512-D latent vectors detailed in the prior section, we can formulate a high-dimensional vector direction between low and high value groups representing an embedding shifting in latent space. Gradually walking along this embedding shift (e.g., interpolating the latent vector from the value in the low group to the value in the high group, or vice versa) corresponds to a smooth style transformation of the urban layout. Further, we empirically observe that performing the interpolation for one metric does not seem to significantly affect other metrics for the same layout. Thus, the manipulation vectors across different metrics can be coalesced to a direction that interpolates two or more metrics simultaneously.

In Fig. 4c, we present an example of a community in Washington D.C., which is identified as a group with low life expectancy and high economic burden in housing and energy consumption. Our indicator-oriented modeling can manipulate the latent vector of several blocks and generate new latent vectors corresponding to higher indicator values. There is no guarantee of the metric behavior of the intermediate layout configurations, but our experiments do show the space to be generally smooth and predictable, thus facilitating intuitive editing abilities.

Discussion

Cities are important, populated, and complex. While a first step is capturing and reconstructing their current state, even more critical is being able to design and plan their future. Key to this is being able to generate current and future potential city layouts. Even more so in today's age of digital twins, where more than just creating a twin of the city today, we need the ability to generate variations of the city of tomorrow, and to correlate behaviors of today so as to plan for tomorrow.

Our proposed generative AI methodology provides a compelling approach to large-scale urban layout generation. By training on a diverse dataset of 330 cities, we demonstrate the efficacy of our approach in generating large-scale urban layouts with realism and semantic consistency, with example uses in urban weather, urban indicator prediction, layout editing, as well as quantitative analysis for reconstruction and inference.

Our generation approach, however, does have some limitations. First, our approach cannot represent city block contours with interior boundaries (e.g.,., interior parks). Second, we do not support all possible building shapes and oddly shaped city blocks. Third, only up to a fixed maximum number of buildings are supported in a block. Fourth, while our method can handle common dead-end roads, such as cul-de-sacs, it cannot handle all cases of dead-end roads. For example, concave and complex block contours may result in an incoherent main axis during block encoding; thus, some building footprints will overlap the dead-end roads in the final output maps. Fifth, road-network to contextual structure relations are not explicitly considered. Finally, we do not explicitly model uncommon structures such as bridges, industrial areas, and some layout-to-indicator predictions are not possible solely from geometric inspection.

Moving forward, our approach paves the way for more accurate and efficient urban simulation, digital twin creation, and game/content design. As future work, we intend to incorporate our method for large-scale photo-realistic multi-view scene synthesis, city-scale 3D modeling and simulation (e.g., for addressing concerns of the World Climate Research Programme

(WCRP)), and explicitly include social-economic features within the city-block features. Those potentials will contribute to synthetic data generation in autonomous driving and world model urban training, and more.

Methods

Building upon the backbone network of GlobalMapper[26] and its extension to context sensitivity[27], we enable realistic 3D city generation given a small percentage of block priors (e.g., building layouts for 5% of the city blocks in a random city). We provide a more extended method overview as supplementary information, including layout graph representation and training (Fig. S1), ablation studies (Tables S1 and S2), comparisons (Tables S3 and S4), generalization and robustness demonstrations (Fig. S2 and S3) and performance (Tables S5). Succinctly, given arbitrary road networks, a city graph is created where each city block is defined as a simple closed road cycle and corresponds to a node. Within each city block, a block graph is created where each node represents a single building, and edges represent connections between spatially adjacent nodes/buildings. Each node stores its 2D building layouts and corresponding building height values. Proposed network structure utilizes stacked graph attention layers[28] as the backbone of the encoder and decoder. Graph attention layers perform weighted multi-layer message passing between connected nodes while progressively extracting features for each node across layers. In the real-world context, messages of 3D geometries are passed between buildings next-to and in-back-of other buildings. The multi-layer message-passing aggregates node features to the 512-D latent space. Subsequently, the latent vector is decoded back to the generated graph representing urban layouts. Autoencoders are a popular tool with many applications[29]; in our case, we exploit variational autoencoders (VAE)[30]. Generated graphs are regularized by a group of reconstruction losses. The weighted sum of each loss component is back-propagated for weight updating. We will make our source code available via GitHub, upon publication.

Datasets

We collected and will make available datasets from the central metropolitan subset of 330 cities across North America, each having at least 100,000 people. The total number of city blocks is 833,473, containing 17,663,607 buildings. We obtained the 2D building layouts and road networks from OpenStreetMap (OSM)[1]. The corresponding building height information is derived from UT-GLOBUS dataset[21]. Minimal clean-up was done to simplify parallel multi-lane roads to single edges and to remove self-looping edges.

Training details

We performed an end-to-end training of our pipeline. We trained our network by a cluster of 100 A100 GPUs. Training typically converges in 10^5 iterations, costing about 9 h. The network itself consists of 130M float parameters, and the average inference time is 5.81 ms per city block by a single A5000 GPU.

Data availability

We will make available via Github the main code defining our generation method. We will make available the data representing our pretrained neural network and our training dataset.

Code availability

We will make available the main code that defines our generation method.

Received: 29 August 2025; Accepted: 23 February 2026;
Published online: 04 April 2026

References

1. OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org (2017).
2. MSFT. Microsoft. Microsoft building footprints. https://github.com/microsoft/GlobalMLBuildingFootprints (2023).
3. Esch, T. et al. World settlement footprint 3d-a first three-dimensional survey of the global building stock. *Remote Sens. Environ.* **270**, 112877 (2022).
4. Parish, Y. I. H. & Müller, P. Procedural modeling of cities. In *Proc. 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 301–308, https://doi.org/10.1145/383259.383292 (Association for Computing Machinery, 2001).
5. Müller, P., Wonka, P., Haegler, S., Ulmer, A. & Van Gool, L. Procedural modeling of buildings. *ACM Trans.Graph*. **25**, 614–623 (2006).
6. Aliaga, D. G., Vanegas, C. A. & Benes, B. Interactive example-based urban layout synthesis. *ACM Trans. Graph.* **27**, 1–10 (2008).
7. Groenewegen, S. A., Smelik, R. M., de Kraker, K. J. & Bidarra, R. Procedural city layout generation based on urban land use models. In *Proc. Short Paper of Eurographics* (2009).
8. Lipp, M., Scherzer, D., Wonka, P. & Wimmer, M. Interactive modeling of city layouts using layers of procedural content. *Comput. Graph. Forum* **30**, 345–354 (2011).
9. Vanegas, C. A. et al. Procedural generation of parcels in urban modeling. In *Computer Graphics Forum*, Vol 31, 681–690 (Wiley Online Library, 2012).
10. Merrell, P., Schkufza, E. & Koltun, V. Computer-generated residential building layouts. In *ACM SIGGRAPH Asia 2010 Papers*, SIGGRAPH ASIA '10. https://doi.org/10.1145/1866158.1866203. (Association for Computing Machinery, 2010).
11. Yu, L.-F. et al. Make it home: automatic optimization of furniture arrangement. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11. https://doi.org/10.1145/1964921.1964981 (Association for Computing Machinery, 2011).
12. Feng, T., Yu, L.-F., Yeung, S.-K., Yin, K. & Zhou, K. Crowd-driven mid-scale layout design. *ACM Trans. Graph*. **35**, https://doi.org/10.1145/2897824.2925894 (2016).
13. Ritchie, D., Wang, K. & Lin, Y.-A. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6175–6183 (IEEE, 2019).
14. Wang, K., Savva, M., Chang, A. X. & Ritchie, D. Deep convolutional priors for indoor scene synthesis. *ACM Trans. Graph*. **37**, https://doi.org/10.1145/3197517.3201362 (2018).
15. Wang, K. et al. Planit: planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Trans. Graph*. **38**, https://doi.org/10.1145/3306346.3322941 (2019).
16. Wu, W. et al. Data-driven interior plan generation for residential buildings. *ACM Trans. Graph*. **38**, https://doi.org/10.1145/3355089.3356556 (2019).
17. Para, W., Guerrero, P., Kelly, T., Guibas, L. J. & Wonka, P. Generative layout modeling using constraint graphs. In *Proc. IEEE/CVF international conference on computer vision*, 6690–6700 (IEEE, 2021).
18. Chang, K.-H. et al. Building-gan: graph-conditioned architectural volumetric design generation. In *Proc. IEEE/CVF International Conference on Computer Vision*, 11956–11965 (IEEE, 2021).
19. Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G. & Furukawa, Y. House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In (eds Vedaldi, A., Bischof, H., Brox, T. & Frahm, J.-M.) *Computer Vision – ECCV 2020*, 162–177 (Springer International Publishing, 2020).
20. Herfort, B., Lautenbach, S., Porto de Albuquerque, J., Anderson, J. & Zipf, A. A spatio-temporal analysis investigating completeness and inequalities of global urban building data in openstreetmap. *Nat. Commun.* **14**, 3985 (2023).
21. Kamath, H. G. et al. Global building heights for urban studies (ut-globus) for city- and street- scale urban simulations: development and first applications. *Scientific Data* **11**, https://www.osti.gov/biblio/2429739 (2024).

22. Zhang, W., Ma, Y., Zhu, D., Dong, L. & Liu, Y. Metrogan: simulating urban morphology with generative adversarial network. In *Proc. 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 2482–2492. https://doi.org/10.1145/3534678.3539239 (Association for Computing Machinery, 2022).
23. Wu, A. N. & Biljecki, F. Ganmapper: geographical data translation. *Int. J. Geogr. Inf. Sci.* **36**, 1394–1422 (2022).
24. Wang Q., Liang Y., Zheng Y., Xu K., Zhao J. & Wang S. Generative AI for urban planning: Synthesizing satellite imagery via diffusion models. In *Computers, Environment and Urban Systems.* Vol. 122, 2025, https://doi.org/10.1016/j.compenvurbsys.2025.102339 (2025).
25. Omar, K. S. et al. Deep umbra: a generative approach for sunlight access computation in urban spaces. *IEEE Trans. Big Data* **11**, 388–401 (2025).
26. He, L. & Aliaga, D. Globalmapper: arbitrary-shaped urban layout generation. In *Proc. IEEE/CVF International Conference on Computer Vision* (2023).
27. He, L. & Aliaga, D. Coho: context-sensitive city-scale hierarchical urban layout generation. In (eds Leonardis, A. et al.) *Computer Vision – ECCV 2024*, 1–18 (Springer, 2025).
28. Velickovic, P. et al. Graph attention networks. In *Proc*. *International Conference on Learning Representations* (ICLR, 2018).
29. Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y. & Xu, Y. Autoencoders and their applications in machine learning: a survey. *Artif. Intell. Rev.* **57**, 28 (2024).
30. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. In *Proc*. *International Conference on Learning Representations* (ICLR, 2014).

Acknowledgements

This work benefitted in parts from NSF Grants 2411273, 2324744 and 2107096, NIST 60NANB24D235, NASA 80NSSC25K7417 and 80NSSC26K0190, and Farish Endowment and UNESCO Chair: UNESCO-UT International Initiative (U2I2; S. Kumar, and R. Bashyam gift) at UT Austin .

Author contributions

L.H. and D.A. wrote the initial draft of the manuscript. H.K., D.N., and S.F. helped edit the manuscript and perform the analysis. All authors have read and approved this manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42949-026-00369-2.

Correspondence and requests for materials should be addressed to Daniel Aliaga.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026