

# A hypergraph model shows the carbon reduction potential of effective space use in housing

Received: 16 April 2024

Accepted: 11 September 2024

Published online: 27 September 2024

 Check for updates

Ramon Elias Weber  , Caitlin Mueller  & Christoph Reinhart 

Humans spend over 90% of their time in buildings, which account for 40% of anthropogenic greenhouse gas emissions and are a leading driver of climate change. Incentivizing more sustainable construction, building codes are used to enforce indoor comfort standards and minimum energy efficiency requirements. However, they currently only reward measures such as equipment or envelope upgrades and disregard the actual spatial configuration and usage. Using a new hypergraph model that encodes building floorplan organization and facilitates automatic geometry creation, we demonstrate that space efficiency outperforms envelope upgrades in terms of operational carbon emissions in 72%, 61% and 33% of surveyed buildings in Zurich, New York, and Singapore. Using automatically generated floorplans in a case study in Zurich further increased access to daylight by up to 24%, revealing that auto-generated floorplans have the potential to improve the quality of residential spaces in terms of environmental performance and access to daylight.

Current estimates predict that the global built area may grow by 250 billion square meters by 2050 to house a growing population<sup>1,2</sup>. Such estimates are necessarily extrapolations from current building practices. While many decades of building science research and practice have enabled design teams across the world to precisely predict carbon reduction savings that can be attained through any number of upgrades for building operation<sup>3–6</sup> and materials<sup>7,8</sup>, surprisingly little attention has been paid to space evaluation methods. The ubiquitous energy use intensity (EUI) metric, defined as energy use per conditioned floor area, has become the de facto benchmark for high-performance buildings, leading to sometimes absurd situations where over-sized single-family homes with rooftop photovoltaics are hailed as beacons of sustainability despite their significant material and space use per occupant.

Given that energy use roughly scales with building size, reducing the floor area per apartment unit while maintaining good indoor environmental conditions, offers a complementary path toward a net zero building stock. Traditional architectural design workflows are unsuitable for this type of exploration since they rely on a human manually drawing interior walls while considering a plethora of

architectural, safety, and Americans with Disabilities Act (ADA) requirements<sup>9</sup>. In residential construction, the position of these interior partitions obviously impacts access to daylight, thermal comfort, and views of the outside. Many design decisions are intuitively made by architects based on prior experience or reference projects<sup>10,11</sup> but without assessing their impact on building performance<sup>12</sup> due to the time, effort, and technical sophistication required to conduct this type of analysis. However, coupling methods of design with quantitative simulation feedback in an early design stage has the potential to significantly improve design outcomes<sup>13</sup>.

While the construction industry has long shied away from quantitatively evaluating space use, the urgency of the climate crisis, along with a shortage of architects to meet the global housing demand, has led to some, mostly developer-driven and proprietary attempts to automatically generate floor plans<sup>14</sup>. Most current implementations are linked to financial cost models, evaluating multiple ways to divide a building footprint into a desired number of apartment units<sup>15–17</sup>. Current approaches for within-unit room divisions are an active area of computer graphics research but are not presently used in the architecture field due to various limitations, including only being able to

Building Technology Program, Department of Architecture, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, USA.

 e-mail: [reweber@mit.edu](mailto:reweber@mit.edu)

represent rectangular<sup>18</sup> or orthogonal boundary conditions<sup>19</sup>, or solely responding to either topological or spatial or boundary constraints<sup>20–24</sup>. On a technical level, machine learning (ML) based models create neural networks that relate the geometric graph structures from room walls to an adjacency graph (vector<sup>25,26</sup> or pixel-based<sup>27</sup>) or use reinforcement learning to subdivide a space<sup>28</sup>. This results in a linear, one-sided generation process, where a room adjacency graph is converted into a visually real and geometrically valid floor plan. Inherently these statistical processes do not allow for exact specifications of room sizes, boundary conditions, or further geometric manipulation of parts of the final output, as are needed in architectural design. Furthermore, implicit geometric relationships are difficult to train, and floor plan training data is sparse, scarce, and unvetted; thus, such approaches can neither guarantee architectural quality nor environmental performance<sup>14</sup>.

In this work, we present the hypergraph, a generalizable shape generator and descriptor for floor plans. The hypergraph represents key characteristics of the shape divisions of any given floor plan layout, enabling both the mapping and benchmarking of suitable, high-performing floor plans, as well as their automatic generation. A hypergraph is created from existing building floor plans and can be applied to new conditions. This allows for translating cultural conventions and practices into new designs, and a fully transparent source attribution. We introduce a spatial analysis workflow to minimize “excess space” while retaining the same spatial functionality of a given floor plan. The concept of excess space is based on the notion that a room with a certain program, for example, a bedroom, has minimum functional requirements in terms of furniture (bed, dresser, cabinet) and space around that furniture that supports its proper use. Areas beyond those functional requirements are then defined as excess. Furthermore, an automatic integration of environmental analysis methods, assessing energy use and daylight, allows us to benchmark high-performance designs and maximize occupant comfort conditions.

## Results

### The hypergraph framework is a graph-based representation of an architectural floor plan

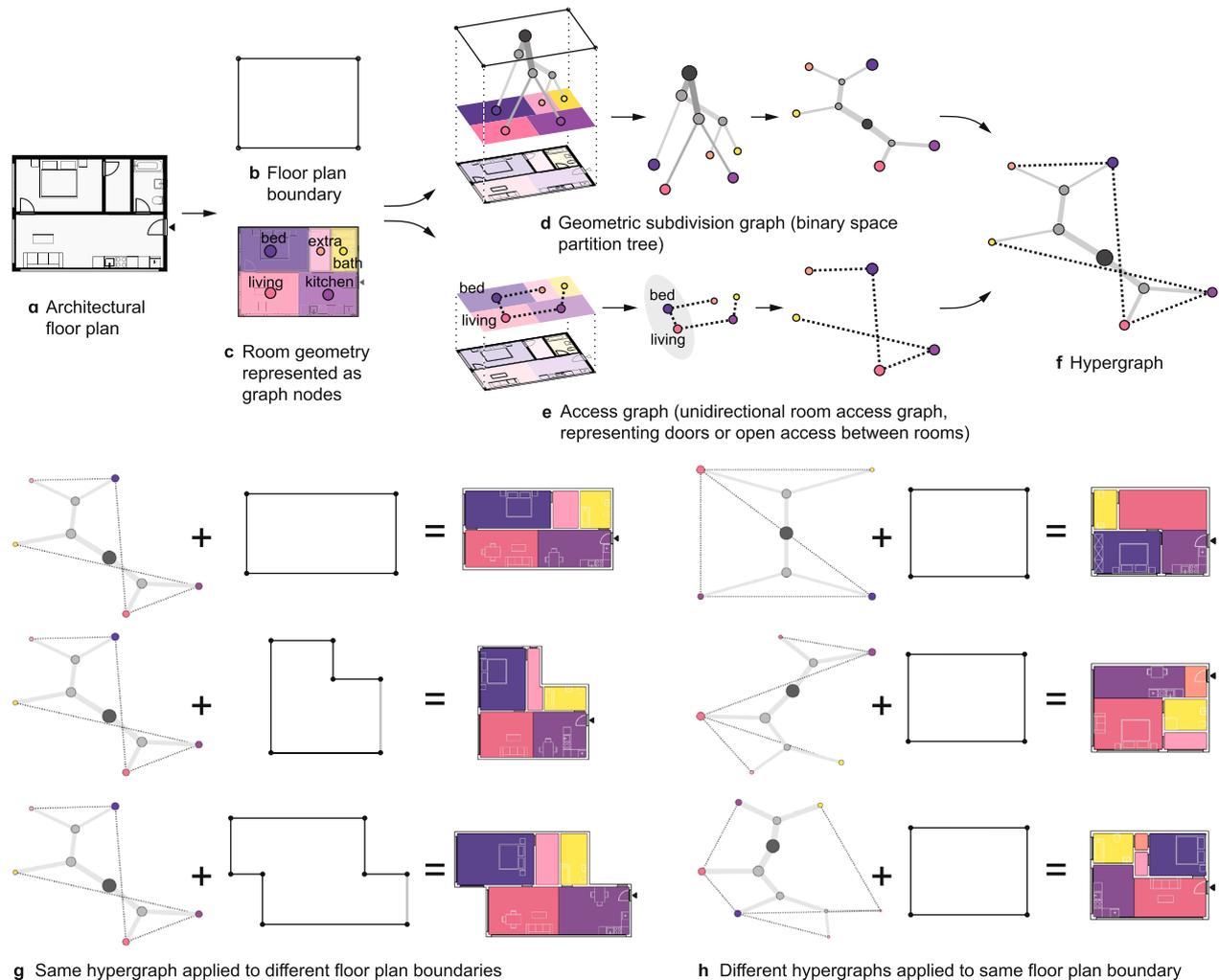
The architectural design of residential buildings, the fitting of apartment units, as well as the internal subdivision within units to create a floor plan, remain processes that are typically performed manually by an architect. The hypergraph aims to computationally execute this two-dimensional design process by providing a unique mapping that either divides a building outline into apartments or an apartment into rooms. This mapping can be applied to any building outline and stored as a graph-based representation. In this paper, we will apply the hypergraphs to subdivide residential apartment units into rooms. To generate a hypergraph, key components of a typical architectural representation of a floor plan (Fig. 1a) are analyzed to extract the floor plan boundary (Fig. 1b) and the different rooms with their specified program (Fig. 1c). In the hypergraph, nodes represent rooms and store information such as room size or program, while edges represent geometric relationships in terms of configuration and adjacency. A binary space partition (BSP) tree<sup>29</sup>, is constructed to represent the geometric subdivision of the boundary into different rooms, where the outermost nodes are the actual rooms of the apartment, color-coded by the program (Fig. 1d). An undirected access graph represents the connectivity between rooms and through that defines the spatial organization of the floor plan (Fig. 1e). The resulting hypergraph (Fig. 1f) is a combination of the two subgraphs, the BSP tree, and the access graph, with nodes representing rooms and edges representing spatial subdivision or access (see “Methods”, Creation of hypergraphs). The BSP tree subdivision can be applied to all convex and simple concave polygonal boundaries, which allowed the encoding of all real-world floor plans we encountered. Given the same boundary

condition, the hypergraph constitutes a bijective mapping that results in the same floor plan and vice versa. The same hypergraph can be applied to a variety of boundary polygons that will result in a unique floor plan for each boundary condition (Fig. 1g), while different hypergraphs applied to the same boundary condition result in different internal subdivisions (Fig. 1h).

### Hypergraphs can be used to generate and evaluate floor plans

To describe a whole building, we can apply the hypergraphs to an apartment boundary, generating detailed floor plans for each apartment unit. A fitting procedure is shown in detail in Fig. 2, in which an apartment boundary polygon (Fig. 2a) is subdivided by a library of different hypergraphs (Fig. 2b) to create different internal apartment configurations (Fig. 2c). We then use the apartment boundary polygon and its orientation toward the building circulation to filter floor plans with similar orientations and façade to adiabatic wall ratios. The hypergraph method removes the need for manual drawing of floor plans and preparation of geometry for different environmental simulation procedures. It allows the complex structure of a floor plan to be described as a graph, a quantifiable and searchable data structure that encodes key parameters of a design. To filter geometrically valid but spatially inadequate outputs, a series of heuristics filter and rank feasible results (see “Methods”, Apartment validity heuristic). With this, we can generate architecturally feasible floor plans where rooms have an aspect ratio and size that makes them usable for their specified use, have access to a façade, and are configured in a way that allows access within the apartment and to the building’s circulation. For assessing the spatial validity of a floor plan, we propose an automatic version of the spatial scoring system developed by the City of Berlin’s public housing provider<sup>30</sup>. Using automatic placement of furniture blocks, we can assess if rooms are large enough to result in livable spaces and compare the overall area to reference floor plans with the same occupancy (Fig. 2d) (see “Methods”, Furniture placement).

To estimate daylight and energy performance, the selected floor plans are automatically converted into a simple 3D model, with walls and windows, that creates a building energy model of the apartment. Focusing on the spatial layout, the sizing of windows was kept constant (at a window-to-wall ratio of 0.6) and placed automatically on walls of rooms requiring daylight. This 3D model can be used to create a building energy model (BEM), a heat-transfer and mass-flow simulation that is industry standard for energy use predictions<sup>31</sup>. However, current BEM practices do not break up apartments (or floors) into detailed sub-spaces, creating a lack of data and standards on detailed room models and how they relate to occupancy. Because of this, we create a simplified apartment BEM that does not differentiate energy use by room program or occupancy and uses the same temperature setpoint across all zones. While this use is justified in the context of the United States, where apartments typically have unified setpoints by a single thermostat, this does not hold in Zurich or Singapore. However, in a sample apartment simulation in different cities, we found only a modest impact of differentiating BEM zones (see Supplementary Note 1.10). Furthermore, we calculate daylight access in the apartments by assessing the spatial daylight autonomy (sDA), a metric for interior spaces that, through a yearly illuminance simulation with physics-based raytracing and local weather data<sup>32</sup>, predicts the percentage of hours per year when a minimum light level of 300 lux can be achieved with daylight (Fig. 2e). While whole building energy models typically do not have the geometric resolution of single rooms, the models generated with the hypergraph method will allow more detailed energy performance analysis that can capture effects of air-flow and natural ventilation for more accurate predictions. Detailed room geometry further allows for more accurate daylight predictions than simple shoebox models or whole building massings, as the internal configuration of a floor plan will determine how light is obstructed inside an apartment.



**Fig. 1 | Generation of Hypergraphs.** The hypergraph is generated from an architectural floorplan (a) that is converted into a boundary (b) and programmatic zones (c) that are translated into the graph nodes. The geometric subdivision of the boundary into the rooms is computed with a binary space partition (BSP) tree (d). The access between rooms is represented as a unidirectional room access graph (e)

– combined they result in a hypergraph (f). A hypergraph can be applied to different floor plan boundaries (g) to create floor plans with the same topology. Different hypergraphs can be applied to the same floor plan boundary to create floor plans with different internal configurations (h).

## Hypergraphs allow the characterization and comparison of floor plans

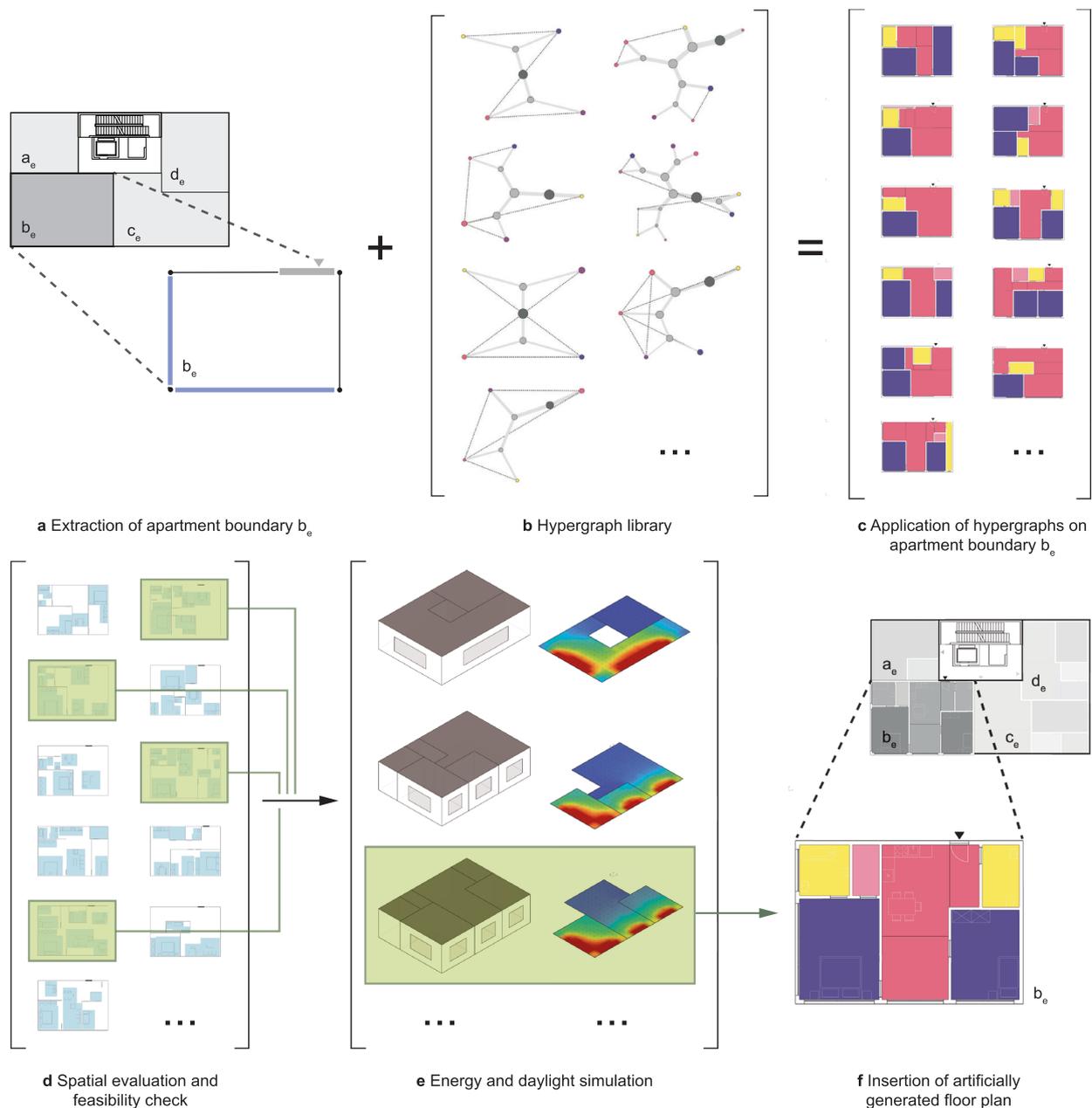
To show the spatial analysis potential of the hypergraph framework, we created a dataset of residential floor plans from around the globe (see “Methods”, Residential building floor plan repository). In order to characterize differences between cities, we compared a representative subset of floor plans from three different cities: Zurich, New York, and Singapore. Contrary to explicit representations with Euclidean geometry or pixel-based representations, the hypergraph encodes relative spatial relationships in addition to geometric properties. This allows the mapping of spatial and typological similarities between floor plans that have different boundary conditions. Based on the number of rooms, subdivision graphs have a variety of sizes which require comparison functions to work with matrices of different dimensions. For a comparison of different hypergraphs, we compare the subdivision matrix (derived from the spatial subdivision graph) separately from the access matrix (derived from the room access graph).

We can describe the overall configuration and complexity of a hypergraph through the number and degree of access and subdivision edges, normalized by the number of rooms. Using a principal

component analysis (PCA) of key attributes (see Supplementary Fig. 9), we can demonstrate that the hypergraph method can be used to distinguish and group similar floor plans according to size and occupancy, as well as compactness (Fig. 3a). Hypergraphs with lower graph complexity, corresponding to smaller size and occupancy, are grouped in the left around the x-axis, while more complex configurations are grouped towards the right. This creates opportunities to quantify spatial differences of apartments across cities, such as simpler spatial properties of apartments in New York, when compared to small-scale, more complex floor plans with higher hypergraph subdivisions in Singapore. The hypergraph allows us to show and encapsulate architectural differences and investigate spatial configurations that are encoded in local architectural practices, prevalent construction techniques, building codes, and climate.

## Hypergraph models allow for new insights into building performance

An automated spatial and environmental analysis allows us to capture differences in daylighting across apartments in the three cities. The results of our simulations support qualitative architectural observations, including that residential apartment in Zurich and Singapore

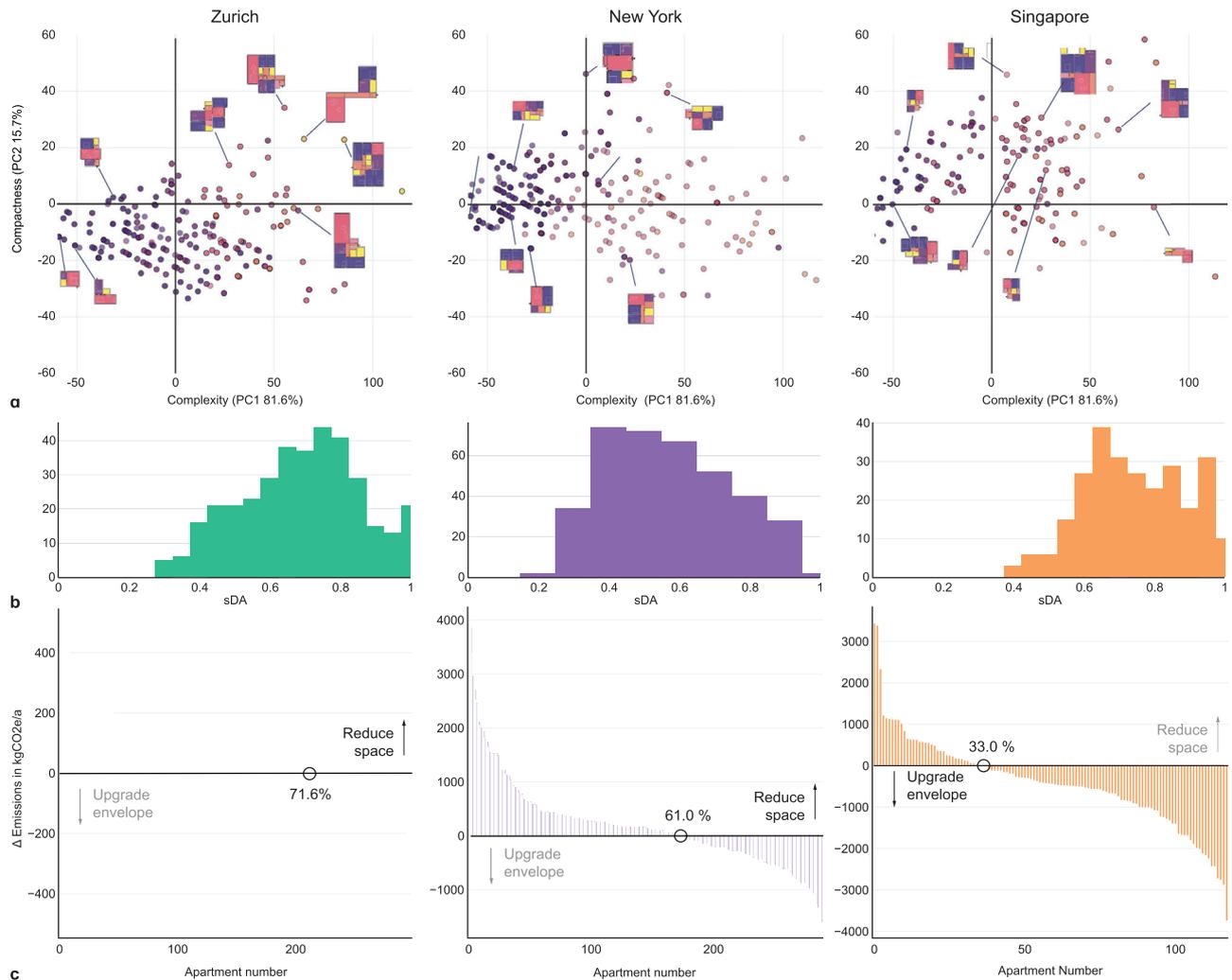


**Fig. 2 | Steps for fitting an apartment using the hypergraph method.** An apartment boundary is extracted from a building (a) and combined with a library of hypergraphs (b). The applied hypergraphs generate different internal subdivisions for the apartment boundary (c). A spatial evaluation using placement of furniture,

accessibility, and room geometry is performed to filter feasible solutions (d). Energy and daylight analysis are performed to further evaluate the resulting floor plan (e), and a chosen plan is inserted into the building (f).

have more access to daylight and are mostly daylit from different sides, while apartments in New York in larger buildings have less daylight access (Fig. 3b). To study the energy efficiency of different apartment geometries, we derive two building energy models for each apartment, with a standard and a high-performance building envelope (see “Methods”, Environmental evaluation workflow). The difference in energy use between the two models shows the energy savings from upgrading the building envelope. We conduct an automated spatial analysis to assess if a floor plan is usable and how its area compares to the minimum size requirements for its occupancy. With this, we can quantify the unused space of a floor plan and with it, the excess emissions associated with heating or cooling. Floor plans that are too large in area or have large ‘unusable’ circulation areas are penalized. A

comparison shows how excess emissions from unused space can be significantly higher than savings from building energy upgrades, assuming that the size of the apartment could be reduced until no excess space remains. We find unused space to be more impactful than envelope upgrades, especially in the more temperate climate in Zurich (71.6%), while the opposite is found in hot and humid Singapore (33.0%), where floor plans are already compact, and envelope performance is crucial due to the climate. In New York, a balance from both measures yields the best results (61.0%) (Fig. 3c). This means that in the case of new construction, apartments that are closer to the minimal spatial requirements with less excess space will have significantly lower energy use, even when constructed with less performative envelope standards.



**Fig. 3 | Comparison of floor plans from Zurich, New York, and Singapore.** A principal component analysis (PCA) of the hypergraphs of all floor plans (see “Methods” Residential floor plan dataset) shows a unique graph structure of each city (a). The calibrated dataset is analyzed to evaluate spatial daylight autonomy

(sDA) (b). Carbon saving scenarios through excess area reduction and envelope upgrades are compared and show that in Zurich, 71.6% of apartments would save more carbon through reducing excess area, compared with 61.0% in New York and 33.0% in Singapore (c) (see “Methods”, Excess area and emissions).

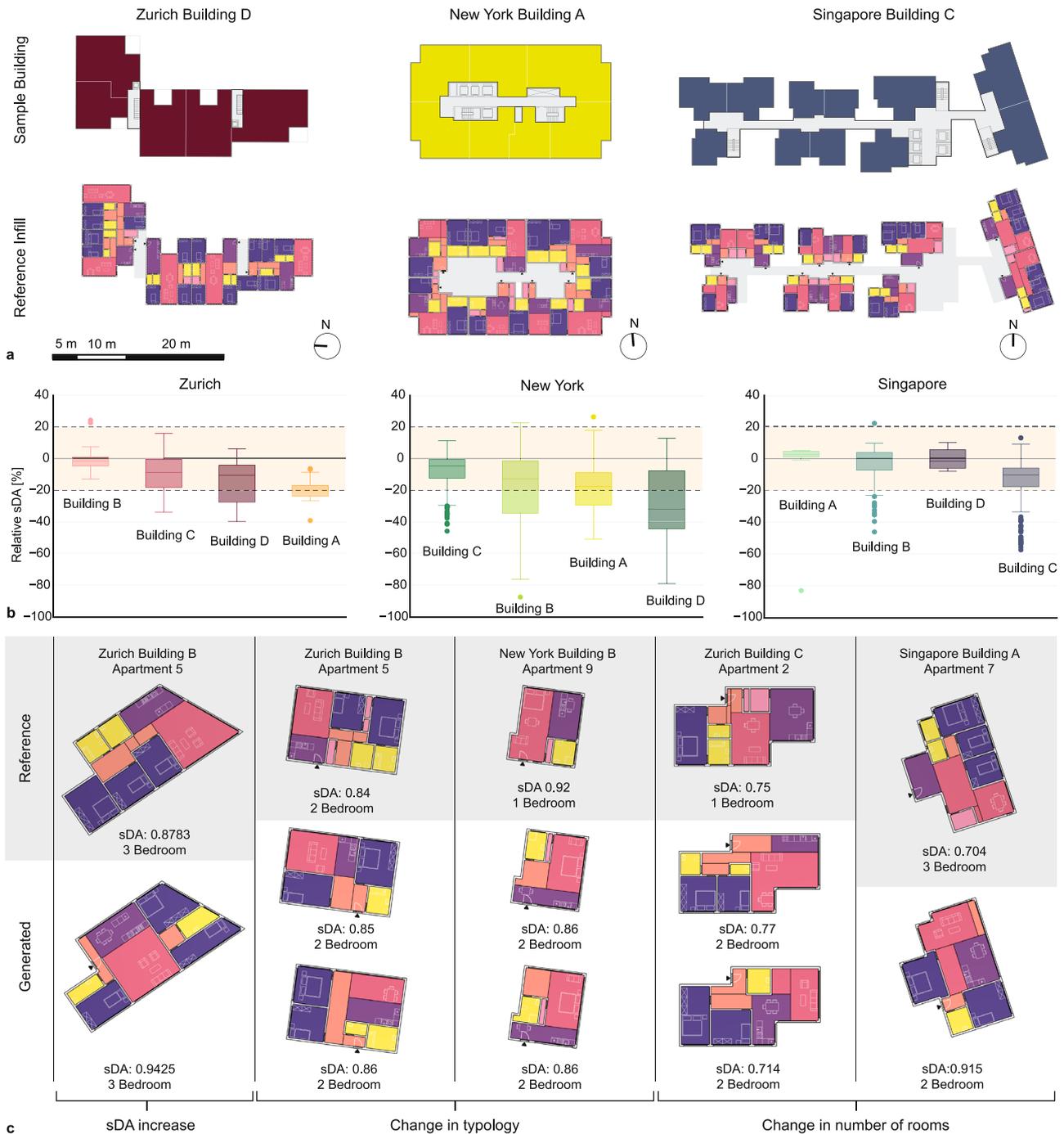
### Hypergraph-generated layouts can outperform real floor plans

Apart from analyzing existing floorplans, hypergraphs can also be used to generate new floorplans. We used the hypergraphs of all collected floor plans and test fit them automatically into real-world boundary geometries of residential buildings from Zurich, New York, and Singapore. A sample building for each city that was fitted is shown in Fig. 4a (for all reference buildings, see Supplementary Figs. 10–12). Depending on the apartment boundary, there are different numbers of valid apartment subdivisions possible, and the hypergraph fitting method was able to propose alternative apartment layouts inside real-world buildings. The apartments created through the hypergraph fitting method were then assessed for daylight to compare their sDA. Even though not all of the artificially generated floorplans would be spatially desirable, the aggregated results of the simulation could be used to predict the daylight performance of a building (Fig. 4b). When comparing the sDA performance of the artificially created apartments, the third quartile of results is within a 20% range or better than the real-world floor plans for sDA and in 5% of cases (Zurich) outperformed the reference floor plan by up to 24% (5.8% of cases in New York by up to 16%, and 0.4% of cases in Singapore by up to 10%). Furthermore, a more detailed qualitative analysis of example floor plans that performed in the upper percentiles of the performance ranges reveals significant

opportunities for the design of new buildings: different spatial configurations that substantially increase daylight, alternative spatial configurations that retain daylight performance, opportunities for adding additional rooms (which indicates that the chosen number of rooms might be too small) or reducing the number of rooms (which indicates that a floor plan might be too tightly fitted) (Fig. 4c).

### Discussion

In summary, we have demonstrated how the hypergraph framework, as a bijective mapping procedure for creating and representing apartment floor plans, can be used to describe spaces across the world. To our knowledge, the hypergraph is the first method that can generally describe apartment layouts and can translate architectural geometry into graph-based representations and vice versa. We show how the method can be used for mapping and comparing different apartments and propose alternative solutions for existing buildings. These assessments will impact retrofit decisions and regulations on a policy and city planning level, allowing us to better understand and shape dense urban environments. Secondly, the automated testing and generation of multiple design options will create opportunities for better design of new buildings, from providing design ideas to creating quality controls that can predict achievable daylight levels and energy



**Fig. 4 | Architectural opportunities for automated floor plan design.** Three sample buildings with reference floor plans are being replaced by hypergraph-generated floorplans (a) (all buildings are defined in Supplemental Figs. 10–12). A

plot showing the relative sDA performance of all successful floor plans with equal or more rooms (b). Comparing reference floor plans with artificially generated floor plans within the same boundary highlights different architectural opportunities (c).

performance for a given context. It could allow for new types of software that would allow self-building and design for communities that cannot afford professional architects while ensuring that the automatically generated buildings have architectural precedents that promote healthy and sustainable spaces.

Currently, emissions from buildings vary greatly<sup>33</sup>, and our method shows new pathways for helping architects align the energy performance and spatial requirements on an urban level with the comfort and needs of a building's inhabitants. The automated nature of the procedure lowers the barrier for environmental simulations of all buildings, which is key in enabling sustainable building design

across the globe. We can show how, in the design of a building, the spatial configuration is more important than building envelope specifications when it comes to building energy usage. Both in the surveyed reference floor plans, as well as in our artificially generated ones, it would have been more favorable in terms of total carbon emissions to build with less space instead of higher performance envelopes. With this we demonstrate that space sufficiency can become a highly impactful carbon mitigation strategy, informing future building energy policy, and should guide the standards and building codes of cities in the future. To address the climate crisis, an overhaul<sup>34</sup> of current environmental certifications – such as the cost balance

method in ASHRAE 90.1<sup>35</sup> or LEED<sup>36</sup> standards that currently do not award spatial efficiency and compactness – is needed. Given the prevalence of EUI in such certifications, smaller spaces are penalized due to higher “equipment” per floor area ratios. Contrary to current energy codes that specify performance requirements, our results show great potential in savings through spatial efficiency measures and thoughtful planning and design of buildings – and with it, the possibility to include spatial metrics for designing buildings with greater energy sufficiency<sup>37</sup>. Evaluating apartment efficiency through standardized furniture blocks per number of expected occupants and the resulting excess space, instead of dwelling size in square foot, could lead to more equitable benchmarks. Even though EUI is the industry standard for evaluating building energy use, occupancy should be actively considered as a key metric to determine emissions, enabling key insights into the carbon footprint of the inhabitants by predicting how the space can be used and programmed (Supplementary Note 1.8).

While the paper shows substantial promise for using automated floor plans to lower building energy use, the authors acknowledge that there are important questions of ownership when using an automated procedure that is based on precedent designs. If clearly vetted floorplans are in the public domain, or are generated in-house by an architecture firm, reproducing geometric configurations will be highly beneficial to increase the speed of design workflows. Questions of intellectual ownership will arise that will have to be addressed by legislators, which ties into the existing debate around large language models<sup>38</sup> and generative AI<sup>39</sup>. However, a key difference with the hypergraph approach is a clear source attribution of each graph and the possibility to explicitly map differences and similarities to existing designs. When deployed responsibly, this could enable validated, quality-controlled design databases.

The hypergraph method specifically targets the spatial generation of residential building layouts. Currently, the scope of the research focuses on single apartments and excludes overall building layout, structural systems, and interfacing with building-level mechanical, electrical, and plumbing (MEP) systems. Future research should address the influence of overall building form on interior layouts, both in terms of spatial efficiency and building performance. Automated spatial evaluation of interior layouts, by predicting use and occupancy, will allow architects to calibrate the overall design of a building to its intended use – by choosing appropriate low-carbon spanning systems that work with the interior configuration, tailoring the building form to allow for more daylight, and enabling building layouts where the rearrangement of interior walls can facilitate different use scenarios for residents<sup>40</sup>. Furthermore, the detailed building energy models that can be created through the hypergraph representation allow for the automatic generation of airflow zoning models that can be used to simulate natural ventilation and airflow between rooms, replacing current practices of manual modeling or simplified assumptions<sup>41</sup>.

The fully automated method can be used to create architecturally valid apartments that can be combined with environmental performance analysis to evaluate and automatically generate culturally relevant and high-performing buildings. By utilizing architecturally vetted reference designs and heuristic procedures that respond to local requirements, the hypergraph method can produce high-quality spaces from virtually any boundary condition. While the method yields geometrically valid options, these designs may not always be of sufficient architectural quality as they depend on the quality of the underlying floorplans in the reference database. However, using only a minimal dataset, we managed to generate artificial solutions that are on par – and up to 24% better in daylight performance – than the real-world built references. This reveals that our method has great potential to lastingly improve the performance of new construction worldwide. We further see opportunities to apply the method to automated benchmarking of building retrofits, including the conversions<sup>42</sup> of

some of the currently 20% empty office buildings<sup>43</sup> in the United States to residential units<sup>44</sup>.

## Methods

### Residential building floor plan repository

We assembled a reference library of ~1444 real-world floor plans, combining award-winning residential floor plans from North American and European contexts from the literature<sup>45–47</sup> and online databases<sup>48</sup> with residential developer plans. The library contains unique real-world floor plans (and their mirrored geometry). Using publicly available data from real-estate brokers and public housing providers, the dataset represents a small subset of a city’s actual apartments. However, we curated the library to encompass a variety of different designs and to represent the prevalent apartment layouts of each city, from studio apartments to large multi-room apartment units and across price ranges from public housing to high-end apartments. From the reference library, dataset floorplans are sampled to map the distribution of a number of bedrooms of the real-world data surveyed in Singapore<sup>49</sup>, New York<sup>50</sup>, and Zurich<sup>51</sup>. As almost 80% of residents in Singapore live in government-provided housing<sup>52</sup> that is built in a standardized fashion, there is less variety in the building stock. This is reflected in a smaller dataset than from Zurich or New York. We compare the real-world data with our dataset in Fig. 5.

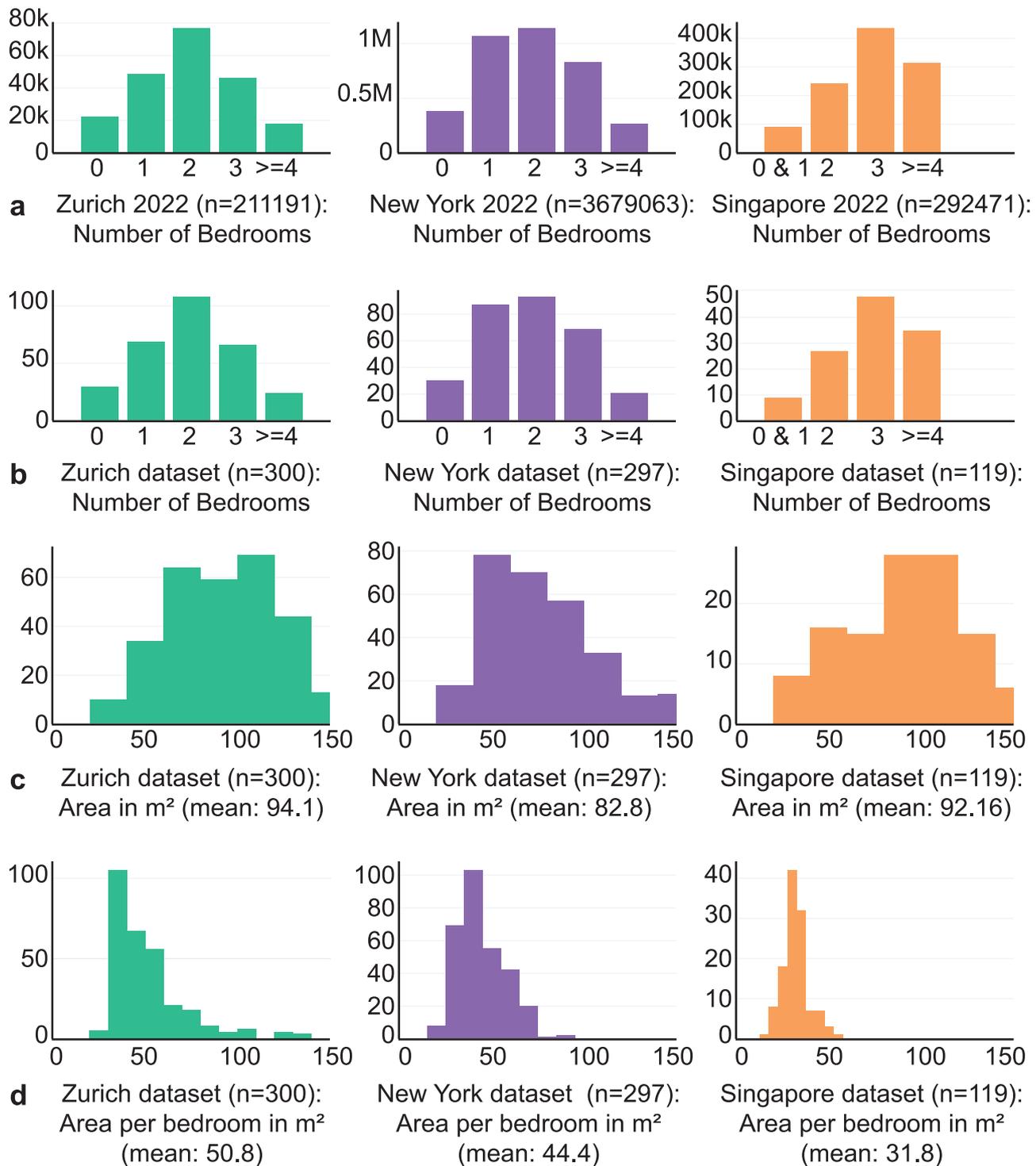
### Reference apartment buildings for artificial floor plan insertion

To test the artificial generation of floor plans and the application of reference floor plans to new boundary conditions, we gathered reference buildings from three different cities. Four buildings from Singapore, Zurich, and New York were chosen to qualitatively reflect the contemporary residential housing architecture of their city. For reasons of data protection for the residents, as well as the architects, we have anonymized the buildings and refer to them as Buildings A, B, C, and D from their respective cities. The building boundaries, as well as the floor plans from Zurich, New York, and Singapore, that we used to benchmark the artificially generated floor plans, can be accessed via the supplemental material (Supplementary Figs. 8–10).

### Creation of hypergraphs

We introduce the hypergraph as a shape descriptor for building floor plans. Graph-based data structures have been applied successfully to represent and generate structured data in biology<sup>53</sup>, chemistry<sup>54</sup>, robotics<sup>55</sup>, building structures<sup>56</sup>, computer games<sup>57</sup>, and urban planning<sup>58</sup>. For the design of building floor plans, graph-based data structures have been deployed to represent wall lines and adjacency graphs<sup>14</sup>. The presented hypergraphs are a combination between an access graph and a subdivision graph. While previous work used the explicit geometric structure of, for example, a molecule, wall segment, or street intersection, as a part of a graph, the hypergraph here is a combination of explicit geometry through adjacency of specific rooms and implicit geometric representation through the subdivision graph. Both graphs can be accessed and analyzed independently via edge and node type specifications in our custom data format.

The BSP<sup>29</sup> tree of the subdivision graph simultaneously represents the final geometry, as well as its step-by-step construction. Each node corresponds to an area (or ratio) and a subdivision angle  $\alpha$ , with (directed) edges connecting the child nodes to the parent node that was subdivided (Fig. 6). Our BSP implementation allows for subdivision of polygons with 3 or more boundary vertices and includes convex and (most) concave polygons. In the BSP tree, the root node specifies the overall area of the subdivision graph. Subsequent children (of type “subdivision”) always have degree 2 and assigned areas, as well as a subdivision angle  $\alpha$ . Leaf nodes of the subdivision graph have a degree 0 on the subdivision graph, and the area is assigned a programmatic type of either (living, bedroom, kitchen, bath, extra, foyer) and a unique ID. The access graph is defined by lists

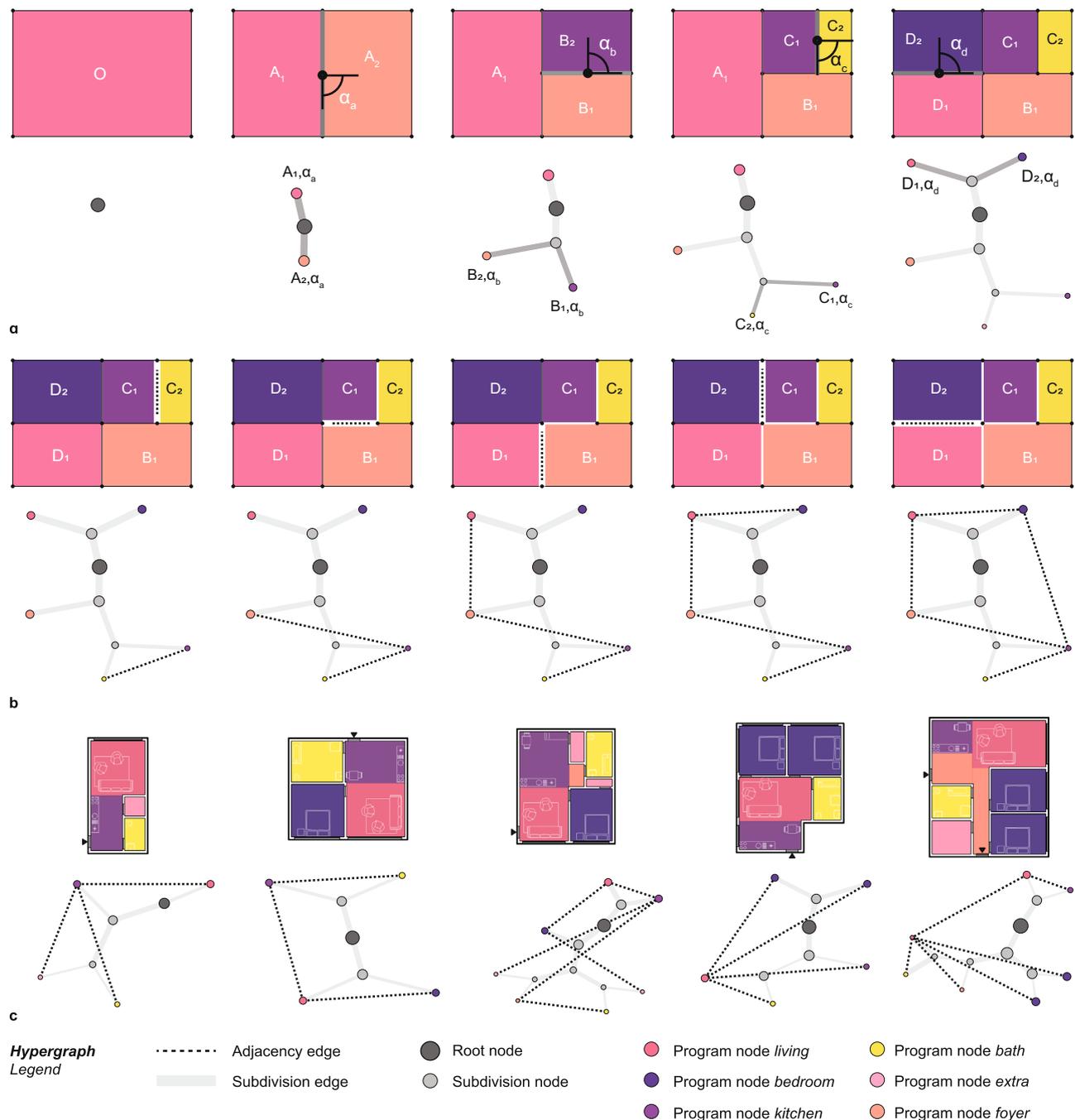


**Fig. 5 | Floor plan dataset.** Distribution of the number of rooms in the cities of Zurich, New York, and Singapore in 2022 by number of bedrooms (a), compared with our reference dataset of analyzed floorplans (b). Histogram plot of apartment area in each city (c) and the area normalized per number of bedrooms in an apartment (d).

of unique IDs in the leaf nodes. Different hypergraphs are illustrated with annotated edges in Fig. 6 (and additionally in Supplementary Fig. 8). Compared to existing methods, the purely geometric nature of the hypergraph creates a direct relationship between graph and spatial form. It is an explicit and not an iterative or optimization-based process that can be computed in real-time.

The outermost child nodes, therefore, represent the rooms in the final floor plan, while inner nodes correspond to the intermediate parent areas in the subdivision process. Even though the

room adjacencies are defined geometrically through the subdivision, the access graph represents the spatial adjacency by (undirected) edges that connect the room nodes (e.g., through a door or an open wall). This dual representation of the internal organization can be captured from any given floor plan boundary. Furthermore, a mapping of both graph nodes of the subdivision and adjacency graphs to the resulting rooms allows for the recording of secondary information, such as room type. The procedure is fully reversible, meaning that a spatial floor plan layout can be encoded in a graph,



**Fig. 6 | Elements of the hypergraph representation.** Step-by-step generation of the subdivision graph (a) from area O, represented by the gray point (graph root). It is subsequently divided into areas  $A_1$  and  $A_2$  with the angle  $\alpha_a$ ,  $A_2$  is divided into  $B_1$  and  $B_2$  with angle  $\alpha$ ,  $B_2$  is divided into  $C_1$  and  $C_2$  with angle  $\alpha_c$ , and  $A_1$  is divided into

$D_1$  and  $D_2$  with angle  $\alpha_d$ . The access between rooms is converted into a bidirectional graph (b) where edges connect the nodes of the rooms that are accessible between one another. Different layout subdivisions and their corresponding hypergraphs are shown in (c).

and the same floor plan layout will emerge given a graph and the original boundary polygon.

### Preprocessing and data preparation of floor plans

The apartment floor plans were sourced as raster images. They were input into the computer-aided design (CAD) software Rhino, where the images were traced and rooms annotated with their respective program, circulation access, façade with lists of lines, and room access (doors). We deploy the inverse of the subdivision algorithm to find the corresponding subdivision graph, and the points in the door locations

to determine access via the access graph. Both graphs are combined into a hypergraph and stored together with façade, circulation, and boundary lines in a JSON database.

### Implementation and visualization

We implement the geometric floor plan placement process in the commercial architectural CAD software Rhino via a custom geometry library in C# through the scripting platform Grasshopper<sup>59</sup>. For this, we utilize and extend the functionalities of the open-source linear algebra library Math.NET<sup>60</sup> and the 2D polygon clipping and offsetting library

**Table 1 | Energy simulation settings for Climate Studio and Energy Plus**

City	New York	Singapore	Zurich
Energy Zone Template	Ashrae 90.1 – Climate Zone 4	Ashrae 90.1 – Climate Zone 1	SIA 2024
Weather File	USA_NY_New.York-LaGuardia.AP.725030_TMYx.2004-2018.epw	SGP_SG_Changi.Intl.AP.486980_TMYx.2004-2018.epw	CHE_ZH_Zurich.Fluntern.066600_TMYx.2004-2018.epw
Grid Carbon Intensity (kg/kWh)	0.55 <sup>51</sup>	0.4057 <sup>52</sup>	0.128 <sup>53</sup>
Energy Template	Ashrae 90.1 – Climate Zone 4	Ashrae 90.1 – Climate Zone 1	SIA 2024
Standard Envelope			
U value (W/m <sup>2</sup> K)	0.3		
Window to Wall Ratio (WWR)	0.6		
Window	DoublePaneClr		
HVAC	Standard Electric HP COP[3,3]		
High-Performance Envelope			
U value (W/m <sup>2</sup> K)	0.1		
Window to Wall Ratio (WWR)	0.6		
Window	Triple Pane LoE		
HVAC	Standard Electric HP COP[3,3]		

Clipper2<sup>61</sup>. For the visualization and representation of the hypergraphs, we convert the graph data structure to a NetworkX<sup>62</sup> graph and visualize it with the force-based Kamada-Kawai algorithm<sup>63</sup> that is applied to the nodes of the subdivision graph. The full code of the implementations, as well as sample files showcasing hypergraph creation and analysis, can be accessed via the supplementary software submission.

### Limitations of the BSP subdivision graph representation

In the current BSP tree implementation, we can represent almost any geometric polygon and subdivision. Even though we were able to represent the studied buildings, there are certain limitations for apartment geometry and configuration that currently cannot be captured in the data format. Failure cases of the subdivision algorithm include highly complex non-convex boundary geometries, as well as polygonal boundaries with holes. While convex boundary geometries are guaranteed to produce a feasible result, highly concave boundary conditions do not. Typical apartment layouts, and those that we observed in our database, fall into the former category, however, this is not guaranteed, especially for synthetic datasets. On an architectural level, we limited the scope of the current implementation to single-story floor plans of multi-unit residential buildings while excluding duplex apartments and single-family homes.

### Apartment validity heuristic

Even though the subdivision algorithm produces a geometrically feasible floor plan, the resulting geometry might not be spatially valid. Different failure cases exist where apartment boundaries are subdivided and produce architecturally infeasible rooms that are inaccessible or do not have access to daylight (Supplementary Fig. 2). For creating artificial floor plans that would be further used in a design context, a visual inspection of the results together with placed furniture items that visualize the scale of rooms, proved to be useful. However, for analysis of large-scale datasets automatic procedures are needed to identify feasible results. Since all hypergraphs are created from a geometrically feasible reference floor plan, we can compare the room geometry of the artificially created floor plan with the original reference floor plan. For this comparison to be computationally efficient, we utilize a scoring method that is computed from the perimeter of the room polygons. The perimeter difference score (Eq. 1) can be applied to single-room polygons (Supplementary Fig. 3), as well as whole apartment floor plans (Supplementary Fig. 4) to determine geometric changes between target and reference. It is a computationally efficient indicator of fit. For more accurate control, a more

computationally intensive pathway and geometry analysis could be envisioned<sup>48</sup>. Furthermore, we can use the furniture placement algorithm to verify if an apartment is feasible by comparing the minimum required furniture to the placed furniture (Supplementary Fig. 7).

$$\delta_p = \left| 1 - \frac{L_{SA}L_B}{L_AL_{SB}} \right| \quad (1)$$

Equation 1: Perimeter difference score  $\delta_p$ , where  $L_A$  is the perimeter of polygon A,  $L_{SA}$  is the perimeter of the square polygon with the same area as A,  $L_B$  is the perimeter of polygon B, and  $L_{SB}$  is the perimeter of the square polygon with the same area as B.

### Environmental evaluation workflow

The automated workflow was implemented in the commercial architectural CAD software Rhino and its integrated scripting platform Grasshopper<sup>59</sup> where the generated floor plan geometry can be automatically converted to be used by the energy simulation software EnergyPlus<sup>64</sup> and the lighting simulation tool Radiance through the Climate Studio package<sup>65</sup>. The simulations were conducted on a Windows computer with the following specifications: 64 GB Ram, Nvidia GeForce GTX 1080 Graphics card, Intel(R) Core (TM) i7-6700 K @ 4.0 GHz Processor. The full daylight and energy simulation required >10 s of calculation time per apartment. Settings for the energy simulation of each city and settings for high and standard-performing building envelopes are listed in Table 1. For each apartment we calculated the EUI in kWh/m<sup>2</sup>/yr for both a standard and high-performance building envelope. To only compare building geometry related factors, we kept the HVAC system the same, even though in a standard building energy retrofit, a more efficient HVAC system could be installed. In the BEM, all rooms were modeled as zones with equal loads according to the ASHRAE and SIA energy templates (Table 1). In typical BEM approaches for early-stage design, zoning, and set point schedules are not differentiated by rooms in a dwelling because of the uncertainty of MEP systems and operational decisions. However, in some cities and regions, it is common to have individualized systems and controls that can be operated independently in different spaces within an apartment (e.g., bedroom cooled at night, living room cooled during the day). In these cases, the layout of the floorplan and its occupancy could impact energy use because of interactions between building physics and systems operations. While our main results do not model this scenario, we tested the impact on a few example cases to show how differentiated zoning could be modeled with our approach. As shown in Supplementary Fig. 15, the impact is

relatively modest for the cases we considered. To calculate the sDA (indicating the fraction of space with more than 300 lux of daylight on average), we only looked at specific rooms in an apartment that require daylight, excluding bathrooms and extra (storage) space (Supplementary Fig. 5). Furthermore, we created a sDA score of each apartment by weighing the area of each room (Eq. 2).

$$d_{tot} = \frac{\sum_{i=1}^n (d_i * A_i)}{\sum_{i=1}^n (A_i)} \quad (2)$$

**Equation 2:** To get the apartments overall daylight score  $d_{tot}$  we multiply the area of each daylit space with its sDA value from our radiance simulation and divide it by the sum of the area of all daylit spaces.

### Furniture placement

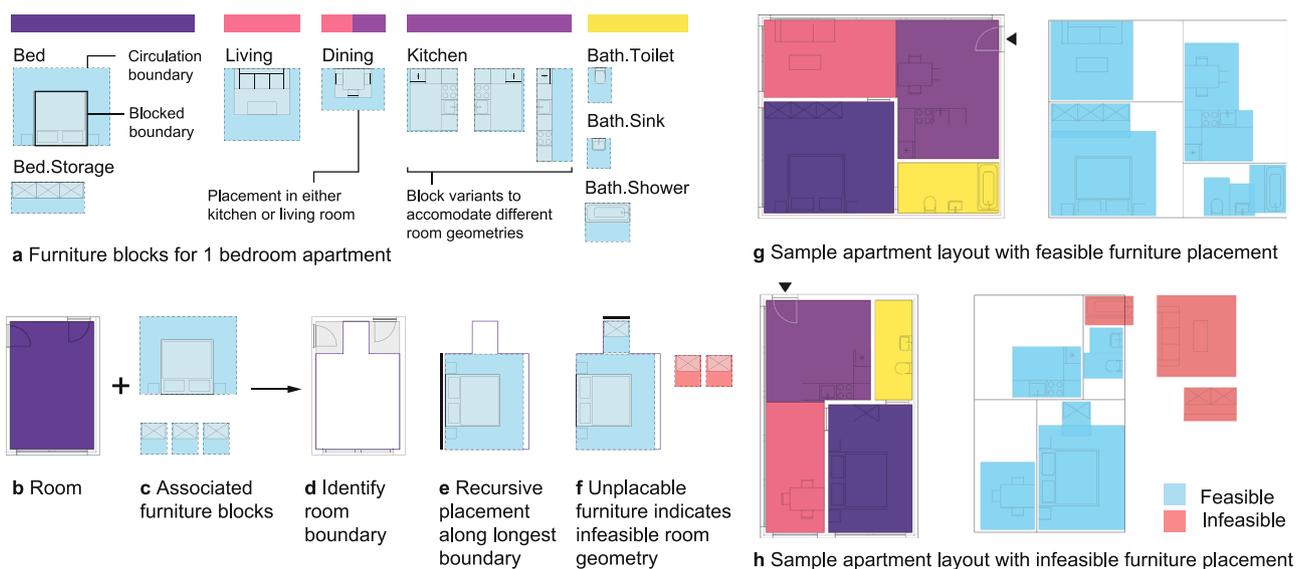
To spatially evaluate a floor plan, we test fit of the layout with furniture. In the computer graphics discipline, furniture placement algorithms have been widely explored using machine learning and procedural techniques<sup>20,66–68</sup>. The use of furniture blocks to test spatial feasibility has been used in the architectural discipline and building codes in defining minimum planning standards in different countries, especially when it comes to affordable housing<sup>69</sup>. A room is deemed feasible if it fits a certain number of predefined furniture blocks. However, the planning standards are only visual guides meant for manual placement of furniture blocks by architecture professionals and are not automated digital procedures. Inspired by the spatial scoring system developed by the City of Berlin's public housing provider<sup>30</sup> and the City of London's planning standard<sup>70</sup>, we translate the manual workflow to an automated digital approach and procedurally place furniture blocks (Fig. 7a) into a floor plan, where furniture blocks are placed recursively along the boundary geometry of each room (Fig. 7b). By grouping furniture items inside a program, we can provide different simple configurations using a faster, less computationally intensive, procedural method.

Each apartment has a minimal number of furniture items that need to fit to be a valid floor plan (Supplementary Fig. 7). In the case of bedrooms and bathrooms, we distinguish between a primary room, such as a bathroom with a bathtub, and a secondary bathroom, with toilet and sink only, in the larger apartments. We used the same minimal furniture to assess floor plans from Zurich, Singapore, and New York. The workflow is very flexible and could be adjusted to include more nuanced cultural requirements. An example floor plan subdivision is valid if all required furniture can be placed (Fig. 7g). If the furniture placement is infeasible (Fig. 7h), that is an indication that the hypergraph subdivision did not create a feasible layout.

### Excess area and emission delta

To show the carbon impact of excess area, and to compare it to the potential energy savings of building envelope upgrades, we compute an emission delta for each floor plan. To calculate the excess carbon from excess area, we use a floor plan furnished with a minimum furniture area. After placement of the furniture, we sum up the total furniture area and compare it to the minimum furniture area of the corresponding apartment size (Supplementary Fig. 7). We derive the total excess area from a subdivision of the furniture area with the total apartment area and the carbon emissions from excess space by multiplying the excess area with the local grid carbon content and EUI (Eqs. 3–6). This value indicates how much carbon could have been saved if the apartment was built in a more compact size with the same number of bedrooms. The emission delta of excess area and envelope upgrade ( $\Delta_e$ ) is derived from Equation 7, using the EUI results of the environmental simulation. If the emission delta  $\Delta_e$  is positive, the emissions from excess space exceed the emissions that could have been saved through a high-performance building envelope. In this research, we derive excess areas by comparing apartments to their standard-sized equivalent with the same number of rooms. In addition, an occupant-centric approach can be used to derive an emission delta ( $\Delta_e$ ) for different numbers of inhabitants in an apartment (Supplementary Note 1.8).

$$F_{tot} = \sum_{n=1}^n (F_n) \quad (3)$$



**Fig. 7 | Spatial evaluation of floor plan.** Furniture blocks that have to be placed inside an apartment to create a valid 1-bedroom unit are shown in (a). A sample placement of furniture for a bedroom is shown in (b–f), where a bed and storage

are placed along a room boundary, and two storage units remain unplaceable, rendering the layout infeasible. A sample layout with feasible furniture placement is shown in (g) and one with infeasible placement in (h).

Equation 3. **Furniture area.**  $F_{tot}$  is the total furniture area (in  $m^2$ ) sum of all furniture areas  $F_n$  of all furniture objects inside the apartment (extra rooms count as furniture, foyer rooms do not). If the  $F_{tot}$  is smaller than the minimum furniture area (Supplementary Fig. 7). If the furnishing was unsuccessful and  $F_{tot}$  is clamped at the minimum furniture area.

$$A_e = A_{apt} - (F_{tot} * M) \quad (4)$$

Equation 4. **Excess area.**  $A_e$  is the excess area (in  $m^2$ ) derived from subtracting the sum of all furniture areas  $F_{tot}$  from the total apartment area  $A_{apt}$  with a multiplier buffer. A positive  $A_e$  indicates an excess area (an apartment exceeding sufficiency), a value close to 0 indicates a good fit, and a value of less than 0 indicates no excess area and a compact apartment. The multiplier (M) can be adjusted to cultural contexts. We use  $M=1.6$  to create apartments with target areas according to the German public housing standard<sup>30</sup>: Studio 34  $m^2$ , 1 Bed 53.6  $m^2$ , 2 Bed 72.6  $m^2$ , 3 Bed 93.1  $m^2$ , 4 Bed 105.9  $m^2$ , 5 Bed 118.7  $m^2$ .

$$C_e = A_e * EUI_s * g_{cc} \quad (5)$$

Equation 5. **Carbon emissions from excess area.**  $C_e$  is the excess carbon emitted from an apartment per annum (kgCO<sub>2e</sub>/a), where  $A_e$  is the excess area ( $m^2$ ) (Eq. 4), the energy use intensity (EUI) (kWh/ $m^2/a$ ) derived from the energy simulation of the apartment with the standard building envelope, and  $g_{cc}$  the local grid carbon content (kgCO<sub>2e</sub>/kWh).

$$\Delta_e = C_e - (A * EUI_s * g_{cc} - A * EUI_{hp} * g_{cc}) \quad (6)$$

Equation 6. **Emission delta.**  $\Delta_e$  is the difference between the carbon emitted from an apartment from excess space  $C_e$  (Equation 5), and the excess carbon emitted from not having an envelope upgrade, where  $A$  is the apartment area ( $m^2$ ),  $EUI_s$  the EUI in (kWh/ $m^2/a$ ) of the apartment with a standard envelope and  $EUI_{hp}$  the EUI (kWh/ $m^2/a$ ) of the apartment with high-performance building envelope, and  $g_{cc}$  the local grid carbon content (kgCO<sub>2e</sub>/kWh).

### Integration of the hypergraph method with automated design workflows

Buildings are highly complex assemblies with both qualitative and quantitative attributes. Their design requires professional expertise from various disciplines, including architecture, structural engineering, and building physics, and has to balance conflicting requirements without a single, optimal solution. Spatial layout generation via the hypergraph is a first step towards automated exploration of diverse geometric layouts that can give insights into a building's performance and artificially generate design solutions. In this research, apartments were studied in isolation from the whole building and evaluated only from a spatial, energy, and daylighting perspective. For whole building simulations, mechanical, electrical, and plumbing (MEP) and structural constraints could further influence the design and the selection of appropriate layouts.

In Supplementary Fig. 14, we outline how the hypergraph method can fit into larger architectural workflows. In the current scope of the work on multi-story residential buildings, it is assumed that apartments with the same floor plan are stacked on top of each other, similar to what we find in most real-world designs. This implies that any movement of vertical supply shafts or structural walls for different apartment layouts would apply across multiple floors and, therefore, would not be a geometric constraint that applies to an individual apartment unit. The BEM created by the hypergraph can be used to dimension and size heating, ventilation, and cooling (HVAC) systems

and test different zone configurations to select configurations with favorable stacking of bathroom and kitchen zones.

Further research should address non-standard, multi-story construction and building retrofits where the stacking of layouts and structure is often not given. On a structural level, the current hypergraph implementation is limited by not differentiating the load-bearing capacity of walls inside a building layout. The structure can interface with the spatial geometric layout of an apartment on different levels. Depending on the construction system, partial sub-graphs inside the hypergraph could be used to represent non-load-bearing internal walls that divide the space or represent shear walls inside a floor plan. For this, the hypergraph could be extended, where nodes could store information beyond the area and program, such as window-to-wall ratio or material. Additional types of edges between nodes could describe physical links between room nodes, such as the structural properties of dividing wall elements.

### Data availability

Source Data related to Figs. 3, 4, 5, and Supplementary Figs. 15 and 16 is available as a Source Data file. Further data, such as building floor plans, used to support the findings of the study are outlined in the Supplementary Data. Additional data and sample hypergraphs are available via the the Github and Zenodo repository (see Code Availability). Source data are provided in this paper.

### Code availability

The source code of the hypergraph implementation is available via Github [<https://github.com/ramonweber/hypergraph>] and Zenodo [<https://doi.org/10.5281/zenodo.13353497>].

### References

1. IEA, *The Critical Role of Buildings, Perspectives for the Clean Energy Transition*, International Energy Agency, Paris, <https://www.iea.org/reports/the-critical-role-of-buildings> (2019).
2. IEA, *Net Zero by 2050 - A Roadmap for the Global Energy Sector*, International Energy Agency, <https://www.iea.org/reports/net-zero-by-2050> (2021).
3. Reyna, J. L. & Chester, M. V. Energy efficiency to reduce residential electricity and natural gas use under climate change. *Nat. Commun.* **8**, 14916 (2017).
4. Zhong, X. et al. Global greenhouse gas emissions from residential and commercial building materials and mitigation strategies to 2060. *Nat. Commun.* **12**, 6126 (2021).
5. Ang, Y. Q., Berzolla, Z. M., Letellier-Duchesne, S. & Reinhart, C. F. Carbon reduction technology pathways for existing buildings in eight cities. *Nat. Commun.* **14**, 1689 (2023).
6. Baniassadi, A., Heusinger, J., Gonzalez, P. I., Weber, S. & Samuelson, H. W. Co-benefits of energy efficiency in residential buildings. *Energy* **238**, 121768 (2022).
7. Simonen, K., Rodriguez, B. X. & De Wolf, C. Benchmarking the Embodied Carbon of Buildings. *Technol. Des* **1**, 208–218 (2017).
8. Röck, M. et al. Embodied GHG emissions of buildings – The hidden challenge for effective climate change mitigation. *Appl. Energy* **258**, 114107 (2020).
9. Civil Rights Division, Disability Rights Section, *2010 ADA Standards for Accessible Design*. <https://www.ada.gov/assets/pdfs/2010-design-standards.pdf> (2010).
10. T. Jocher and S. Loch, *Raumplilot Grundlagen*. kraemerverlag, (2012).
11. O. Heckmann, F. Schneider, and E. Zapel, *Floor Plan Manual Housing*, Fifth, Rev. Birkhäuser, <https://doi.org/10.1515/9783035611496> (2018).
12. Çavuşoğlu Ö. H. & Çağdaş, G. Why do we need building information modeling (BIM) in conceptual design phase? In *Computer-Aided Architectural Design. Future Trajectories*, G. Çağdaş, M. Özkar, L. F.

- Gül, and E. Güner, Eds., Singapore: Springer Singapore, pp. 121–135 (2017).
13. Burnell, E., Stern, M., Flooks, A. & Yang, M. C. Integrating design and optimization tools: A designer centered study, In *Volume 7: 29th International Conference on Design Theory and Methodology*, Cleveland, Ohio, USA: American Society of Mechanical Engineers, (2017).
  14. Weber, R. E., Mueller, C. & Reinhart, C. Automated floorplan generation in architectural design: A review of methods and applications. *Autom. Constr.* **140**, 104385 (2022).
  15. Sidewalk Labs - Part of Google, Delve. New York, 2024. <https://www.sidewalklabs.com/products/delve> (2024).
  16. Archistar. Sydney, <https://www.archistar.ai/> (2024).
  17. Test Fit. Dallas <https://www.testfit.io/> (2024).
  18. Ślusarczyk, G., Strug, B., Paszyńska, A., Grabska, E. & Palacz, W. Semantic-driven graph transformations in floor plan design. *Comput. Aided Des.* **158**, 103480 (2023).
  19. Bisht, S. et al. Transforming an adjacency graph into dimensioned floorplan layouts. *Comput. Graph. Forum* **41**, 5–22 (2022).
  20. Para, W., Guerrero, P., Kelly, T., Guibas, L. & Wonka, P. Generative Layout Modeling using Constraint Graphs. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, 6670–6680 (2021).
  21. Wu, W., Fan, L., Liu, L., Wonka, P. MIQP-based layout design for building interiors, *Comput. Graph. Forum* **37**, 511–521 (2018).
  22. Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G. & Furukawa, Y. *House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation*, presented at the Computer Vision – ECCV 162–177. [https://doi.org/10.1007/978-3-030-58452-8\\_10](https://doi.org/10.1007/978-3-030-58452-8_10) (2020).
  23. Hu, R. et al., Graph2Plan: Learning floorplan generation from layout graphs. *ACM Trans. Graph.* **39**, 118 (2020).
  24. Sun, J. et al. WallPlan: synthesizing floorplans by learning to generate wall graphs. *ACM Trans. Graph.* **41**, 1–14 (2022).
  25. Shabani, M. A., Hosseini, S., & Furukawa, Y. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, 5466–5475 (2023).
  26. Tang, H. et al., Graph transformer GANs for graph-constrained house generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, 2173–2182 (2023).
  27. Carrera, L. et al. The impact of architecturally qualified data in deep learning methods for the automatic generation of social housing layouts. *Autom. Constr.* **158**, 105238 (2024).
  28. Kakooee R. & Dillenburger, B. Reimagining space layout design through deep reinforcement learning, *J. Comput. Des. Eng.* **11**, 43–55 (2024).
  29. De Berg, M., Cheong, O., Van Kreveld, M. & Overmars, M. *Computational Geometry: Algorithms and Applications*. Berlin, Heidelberg: (Springer Berlin Heidelberg, 2008).
  30. Howoge, *Grundrissbewertungssystem*. <https://www.howoge.de/wohnungsbau/expertise-bau/grundrissbewertungssystem.html> (2023).
  31. Polly, B., Kruijs, N. & Roberts, D. *Assessing and Improving the Accuracy of Energy Analysis for Residential Buildings*. National Renewable Energy Lab. (NREL), Golden, CO (United States, 2011).
  32. Illuminating Engineering Society of North America. *ANSI Standard: IES Spatial Daylight Autonomy (sDA) and Annual Sunlight Exposure (ASE)*. [https://webstore.ansi.org/preview-pages/IESNA/preview\\_IES+LM-83-12.pdf](https://webstore.ansi.org/preview-pages/IESNA/preview_IES+LM-83-12.pdf) (2024).
  33. Goldstein, B., Gounaridis, D. & Newell, J. P. The carbon footprint of household energy use in the United States. *Proc. Natl. Acad. Sci. USA* **117**, 19122–19130 (2020).
  34. Asensio, O. I. & Delmas, M. A. The effectiveness of US energy efficiency building labels. *Nat. Energy* **2**, 17033 (2017).
  35. ASHRAE 90.1-2022 (I-P). [https://www.techstreet.com/ashrae/standards/ashrae-90-1-2022-i-p?product\\_id=2522082](https://www.techstreet.com/ashrae/standards/ashrae-90-1-2022-i-p?product_id=2522082) (2023).
  36. USGBC, LEED Certification, U.S. Green Building Council. <https://www.usgbc.org/> (2023).
  37. Hu, S. et al. A systematic review of building energy sufficiency towards energy and climate targets. *Renew. Sustain. Energy Rev.* **181**, 113316 (2023).
  38. Grynbaum, M. & Mac, R. The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work. *New York Times*, New York, **28**, (2023).
  39. Epstein, Z. et al. Art and the science of generative AI. *Science* **380**, 1110–1111 (2023).
  40. Schneider, T. & Till, J. Flexible housing: opportunities and limits. *Archit. Res. Q.* **9**, 157–166 (2005).
  41. Tarkhan, N., Mokhtar, S., Weber, R. E. & Reinhart, C. Natural ventilation in a warming climate: An evaluation of computational simulation methods and metrics. In *2022 Annual Modeling and Simulation Conference (ANNSIM)*, San Diego, CA, USA: IEEE, 694–705 (2022).
  42. Poley, D. *The Next Crisis Will Start With Empty Office Buildings*. The Atlantic. <https://www.theatlantic.com/ideas/archive/2023/06/commercial-real-estate-crisis-empty-offices/674310/> (2023).
  43. Rowden, Jacob U. S. Office Outlook, Cyclical challenges persist but tentative green shoots emerging in second quarter. *JLL*, <https://www.us.jll.com/content/dam/jll-com/documents/pdf/research/americas/us/jll-us-office-outlook-q2-2023.pdf> (2023).
  44. Loh, T. Hadden, Terplan, E. & Rowlands, D. *Myths about converting offices into housing—and what can really revitalize downtowns*. (The Brookings Institution, Washington, 2023).
  45. Stamm-Teske, W., Fisher, K. & Haag, T. *Raumpilot: Wohnen. Krämer*, (2010).
  46. Zapel, E. *Birkhäuser. Floor plan manual housing*, (2017).
  47. Badger, E. & Buchanan, L. *Here's How to Solve a 25-Story Rubik's Cube*, New York Times, New York, NY, <https://www.nytimes.com/interactive/2023/03/11/upshot/office-conversions.html> (2023).
  48. *Architectural Database*. Birkhauser, <https://bdt.degruyter.com/> (2023).
  49. Wee Kim, W. *Census of population, 2020: statistical release 2: households, geographic distribution, transport, and difficulty in basic activities*. Singapore: Department of Statistics, Ministry of Trade and Industry, Republic of Singapore, (2021).
  50. Gaumer, E. *The 2021 New York City Housing and Vacancy Survey: Selected Initial Findings*, NY: Department of Housing Preservation & Development, New York, <https://www.nyc.gov/assets/hpd/downloads/pdfs/services/2021-nychvs-selected-initial-findings.pdf> (2022).
  51. Statistik Stadt Zürich, Präsidialdepartement, *Haushalte in Neubau-Wohnungen nach Bauperiode, Zimmerzahl, Eigentumsart, Haushalttyp und Stadtquartier, seit 2006*. [https://data.stadt-zuerich.ch/dataset/bau\\_neubau\\_whg\\_hh\\_wnhf\\_bp\\_zizahl\\_eart\\_hhtyp\\_quart\\_od6991](https://data.stadt-zuerich.ch/dataset/bau_neubau_whg_hh_wnhf_bp_zizahl_eart_hhtyp_quart_od6991) (2023).
  52. Department of Statistics Singapore, *Singapore Department of Statistics | Population Trends 2021*, <https://www.singstat.gov.sg/-/media/files/publications/population/population2021.pdf> (2021).
  53. Guo, M. et al. *Data-Efficient Graph Grammar Learning for Molecular Generation*. International Conference on Learning Representations (ICLR). <https://doi.org/10.48550/arXiv.2203.08031> (2022).
  54. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **1**, 045024 (2020).
  55. Zhao, A. et al. RoboGrammar: graph grammar for terrain-optimized robot design. *ACM Trans. Graph.* **39**, 1–16 (2020).

56. Whalen, E. & Mueller, C. Toward reusable surrogate models: Graph-based transfer learning on trusses. *J. Mech. Des.* **144**, 021704 (2022).
57. Merrell, P. Example-based procedural modeling using graph Grammars. *ACM Trans. Graph.* **42**, 1–16 (2023).
58. Fiser, M. et al. Learning geometric graph grammars. In *Proceedings of the 32nd Spring Conference on Computer Graphics*, Slovenia Slovakia: ACM, 7–15 (2016).
59. Robert McNeel & Associates, *Rhino3D® and Grasshopper3D®*. <https://www.rhino3d.com> (2022).
60. C. Ruegg, J. Van Gael, J. Kalias, and J. Larsson, *Math.Net open source library*. <https://www.mathdotnet.com/> (2023).
61. A. Johnson, *Clipper2*. <https://github.com/AngusJohnson/Clipper2> (2023).
62. A. Hagberg, D. Schult, and P. Swart, *NetworkX*. <https://github.com/networkx/networkx> (2023).
63. Kamada, T. & Kawai, S. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* **31**, 7–15 (1989).
64. Crawley, D. B., Pedersen, C. O., Lawrie, L. K., Winkelmann, F. C. & Juve, G. Energy plus: Energy simulation program'. *ASHRAE J* **42**, 49–56 (2000).
65. Solemma, *Climate Studio*. Solemma LLC., <https://www.solemma.com/> (2023).
66. Deitke, M. et al. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. *Advances in Neural Information Processing Systems*, **35**, 5982–5994. [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/27c546ab1e4f1d7d638e6a8dfbad9a07-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/27c546ab1e4f1d7d638e6a8dfbad9a07-Paper-Conference.pdf) (2022).
67. Fisher, M., Ritchie, D., Savva, M., Funkhouser, T. & Hanrahan, P. Example-based synthesis of 3D object arrangements. *ACM Trans. Graph.* **31**, 1–11 (2012).
68. Yu, L.-F. et al. Make it home: automatic optimization of furniture arrangement. *ACM Trans. Graph.* **30**, 1–12 (2011).
69. Jacoby, S., Arancibia, A. & Alonso, L. Space standards and housing design: typological experimentation in England and Chile. *J. Archit.* **27**, 94–126 (2022).
70. Design for London and Mayor of London, *London Housing Design Guide*, London Development Agency, London, [http://www.designforlondon.gov.uk/uploads/media/Interim\\_London\\_Housing\\_Design\\_Guide.pdf](http://www.designforlondon.gov.uk/uploads/media/Interim_London_Housing_Design_Guide.pdf) (2010).

## Acknowledgements

This research was supported by an MIT Presidential Graduate Fellowship.

## Author contributions

R.W. conceived and designed the analysis with inputs from C.R. and C.M. R.W. collected the data, created the analysis tools, and performed the computational study and analysis. R.W. wrote the manuscript in consultation with C.R. and C.M.

## 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/s41467-024-52506-z>.

**Correspondence** and requests for materials should be addressed to Ramon Elias Weber.

**Peer review information** *Nature Communications* thanks Guohao Dai, Lars O Grobe, and the other anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

**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) 2024