Urban Areal Spatial Data Generation Using Auto-Encoding Tree Supplementary Material

Anonymous CVPR 2021 submission

Paper ID 11414

I. Overview

In the supplementary, we first introduce the distance metrics used for hierarchical clustering. Then the different ways of data representation are discussed. Next, we present the details about our encoding and decoding modules with different basic functions(i.e. MLP and LSTM). Finally, more reconstruction and generation results are shown.

II. Discussion about relative representation

As described in Section 3.1 of the main paper, we preprocess all nodes' parameters relative to their father nodes. To verify the effectiveness of this relative representation, we conduct a comparison experiment between different ways of representation. In Figure I, we plot the reconstruction results of models with relative and absolute representation on NYC dataset. It can be found that relative representation keeps the spatial relations among objects better than absolute representation, which is thus employed in our tree-based structure data.

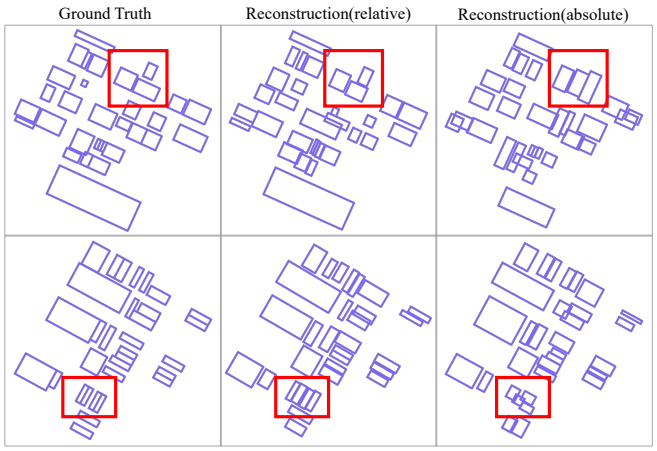


Figure I. Reconstruction results of models with the relative and absolute representation of spatial data. The red boxes highlight the most obvious difference between results.

III. Distance metric for hierarchical clustering

We detail the distance metrics used for hierarchical clustering below,

$$
\begin{aligned}
D(i,j) &= \lambda_1 D_{\text{center}}(i,j) + \lambda_2 D_{\text{area}}(i,j) + \lambda_3 D_{\text{shape}}(i,j) + \\
&\quad \lambda_4 D_{\text{angle}}(i,j) + \lambda_5 D_{\text{merge}}(i,j), \\
D_{\text{center}}(i,j) &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \\
D_{\text{area}}(i,j) &= \left| l_i w_i - l_j w_j \right|, \\
D_{\text{shape}}(i,j) &= \left| (l_i / w_i) - (l_j / w_j) \right|, \\
D_{\text{angle}}(i,j) &= \left| (a_i + a_j)/2 - a_{\texttt{MBR}(i,j)} \right|, \\
D_{\text{merge}}(i,j) &= \left| l_i w_i + l_j w_j - l_{\texttt{MBR}(i,j)} w_{\texttt{MBR}(i,j)} \right|,
\end{aligned}
$$

where $D(i,j)$ represents the distance between rectangle i and j, $\texttt{MBR}(i,j)$ represents the minimum bounding rectangle (MBR) of rectangle i and j and λ represents the weight of each distance. Specifically, D_{center} measures the Euclidean distance between the center points of two rectangles; D_{area}, D_{shape} and D_{angle} separately measure the difference between the area, the aspect ratio, and the orientation of two rectangles; and D_{merge} measures the difference between the sum of the two rectangles area and their MBR area. The values of λ are determined empirically, detailed in Section 4.5 of the main paper,.

IV. Encode and decode modules

As introduced in Section 3.2 of the main paper, we employ both MLP and LSTM cell as the f_e function to hierarchically encode features from bottom to top. The details of two different encoding modules can be found in Figure II. For the encoding module of AETree(MLP) (Figure II left), we adopted a shared MLP to learn features of the left and right nodes. And then the features of the father nodes ($\mathcal{F}'_f$) are obtained by using a summation aggregation function on learned features. Moreover, for encoding module of AE-Tree(LSTM) (Figure II right), we first split the features of children nodes to hidden states(h) and cell states(c), which are input into a LSTM Cell along with the parameters of

nodes($\mathcal{P}$). The output of the LSTM Cell are concatenated at each node and then summarized to obtain the features of the father nodes($\mathcal{F}'_f$).

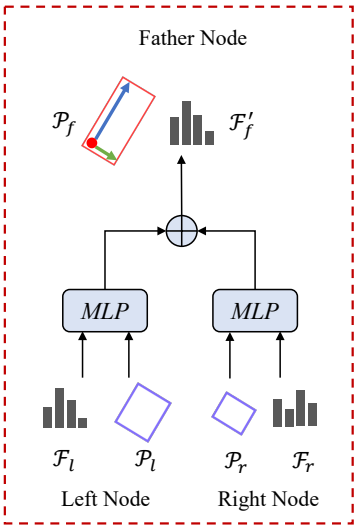
Similarly, we illustrate the decoding module of AETree(MLP) and AETree(LSTM) in Figure III. For AETree(MLP) model, the learned features and parameters of nodes are input to its decoding module. The inputs first pass through a MLP layer to enlarge the feature dimensions and then go through another MLP layer to acquire the parameters, features, and indicators of children nodes. And for AETree(LSTM)'s decoding module, we also implement a linear layer to enlarge the feature dimensions and then split features to hidden states(h) and cell states(c). By inputting hidden states, cell states and the parameters to a LSTM Cell, we divide the output of hidden states and cell states into two parts, where one part of hidden states(h'_l) and cell states(c'_l) are concatenated as the intermediate features of the left nodes, and the other part of these two states are concatenated as the intermediate features of the right node. The parameters, features, and the indicator of each node are finally acquired by employing the MLP on the corresponding intermediate features.

V. More reconstruction results

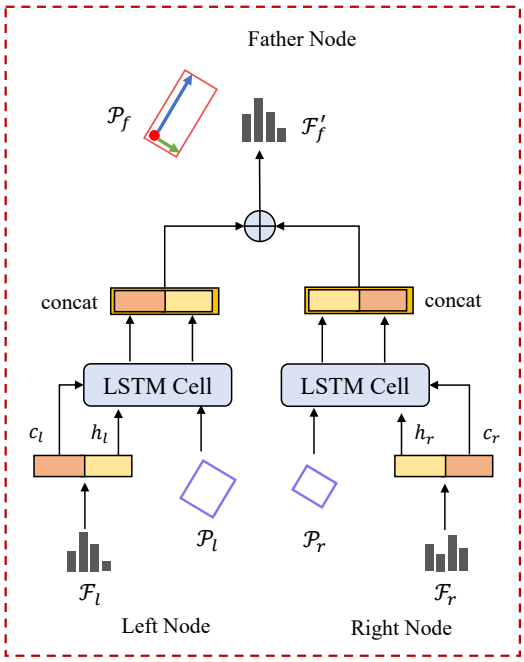
Figure IV and VI, and Figure V and VII show the 3D and 2D reconstruction results on the NYC and Zurich dataset, respectively.

VI. More generation results

Figure VIII and Figure IX illustrate the unconditional 2D generation results on the NYC and Zurich dataset, separately.

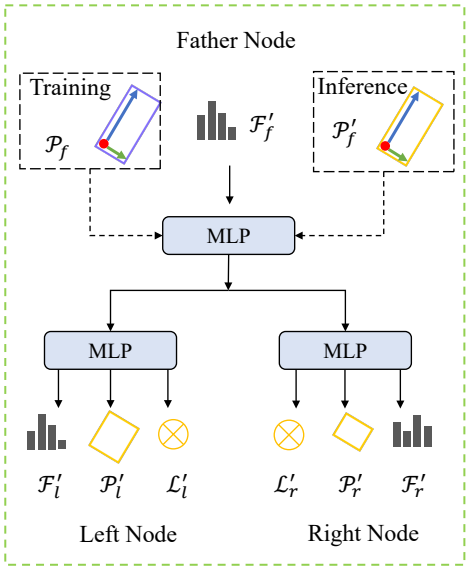


Encoding module of AETree (MLP)

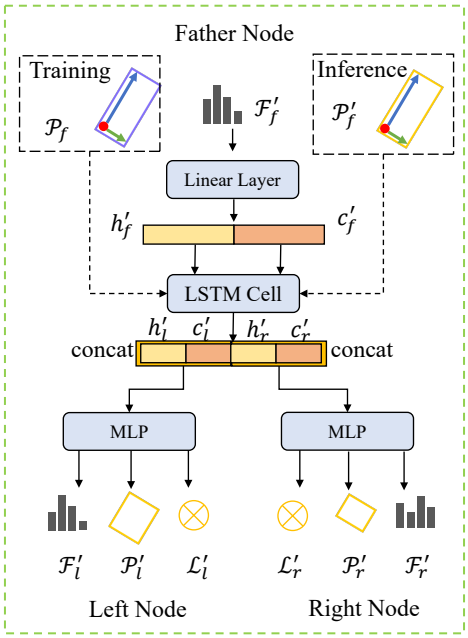


Encoding module of AETree (LSTM)

Figure II. Illustration of the encoding module of AETree(MLP) and AETree(LSTM).



Decoding module of AETree (MLP)



Decoding module of AETree (LSTM)

Figure III. Illustration of the decoding module of AETree(MLP) and AETree(LSTM).

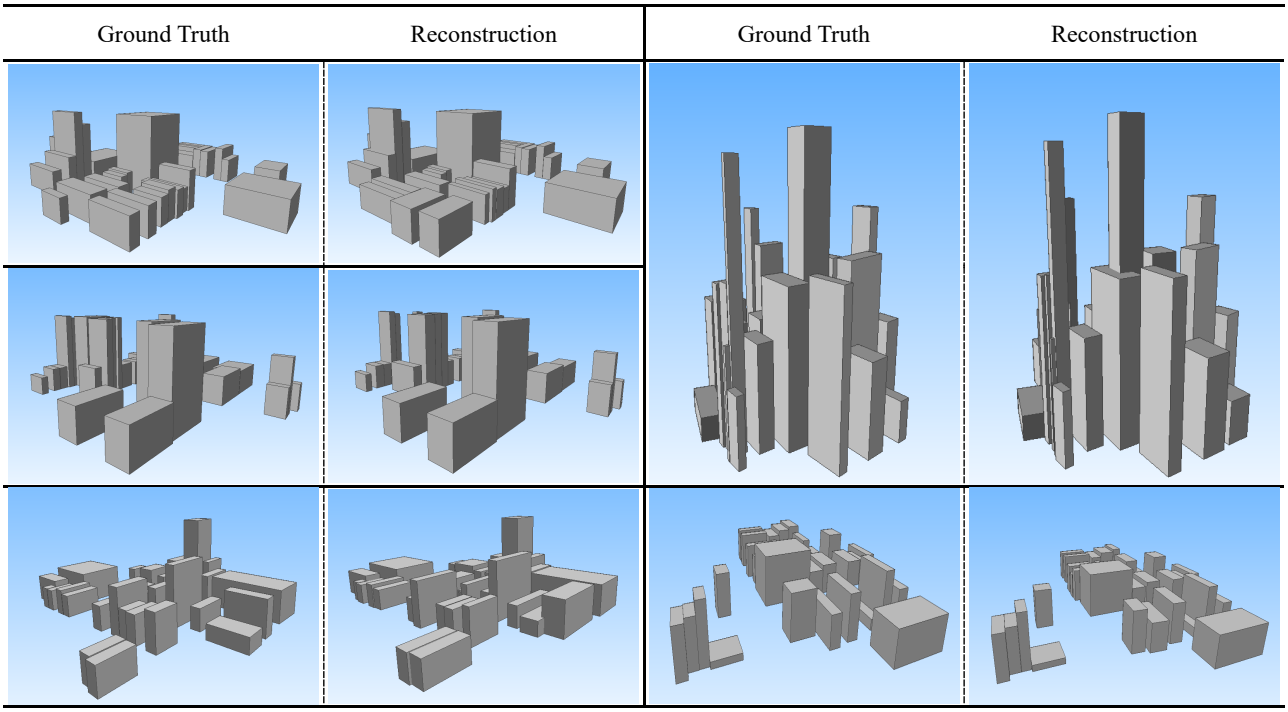


Figure IV. 3D reconstruction results of AETree trained on the NYC Dataset

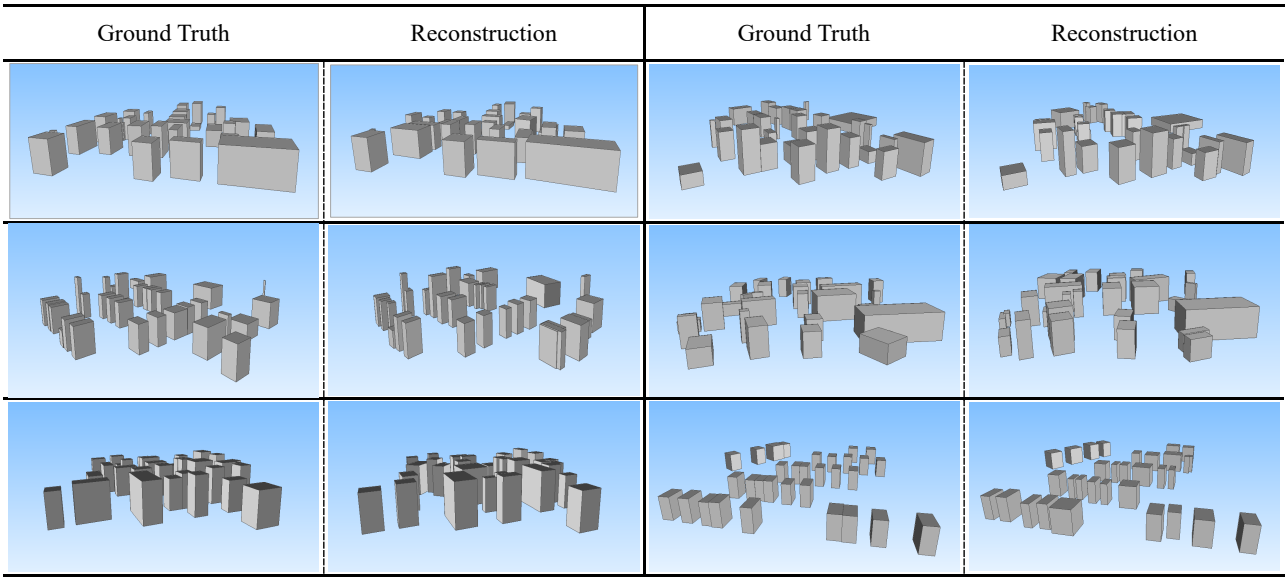


Figure V. 3D reconstruction results of AETree trained on the Zurich Dataset

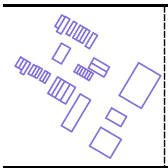
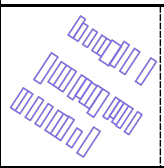
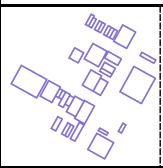
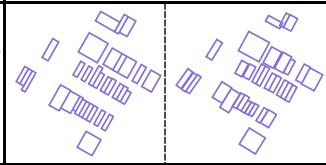
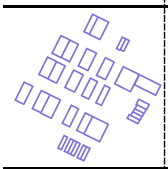
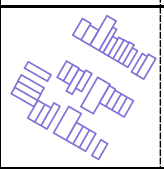
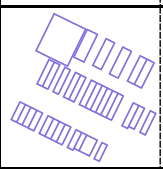
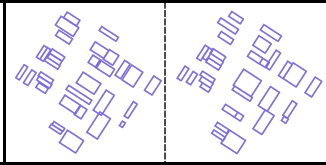
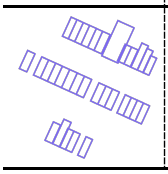
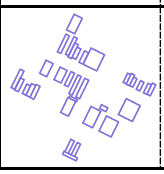
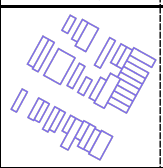
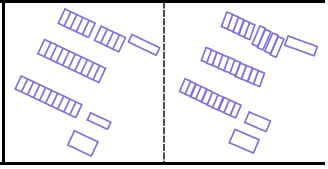
Ground Truth	Reconstruction	Ground Truth	Reconstruction	Ground Truth	Reconstruction	Ground Truth	Reconstruction
							
							
							

Figure VI. 2D reconstruction results of AETree trained on the NYC Dataset

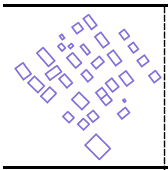
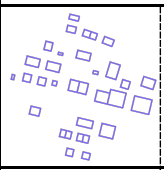
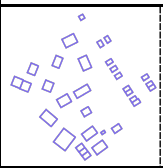
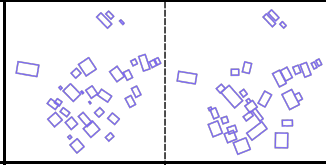
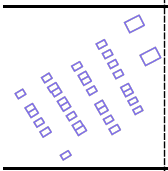
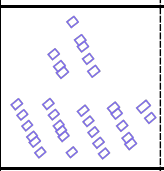
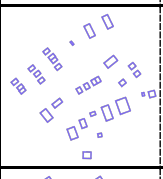
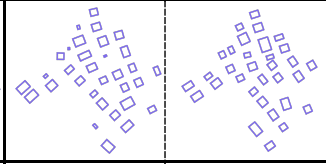
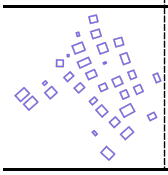
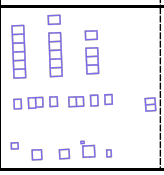
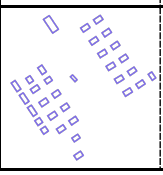
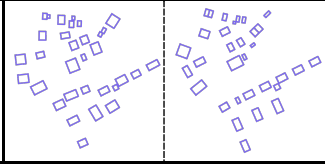
Ground Truth	Reconstruction	Ground Truth	Reconstruction	Ground Truth	Reconstruction	Ground Truth	Reconstruction
							
							
							

Figure VII. 2D reconstruction results of AETree trained on the Zurich Dataset

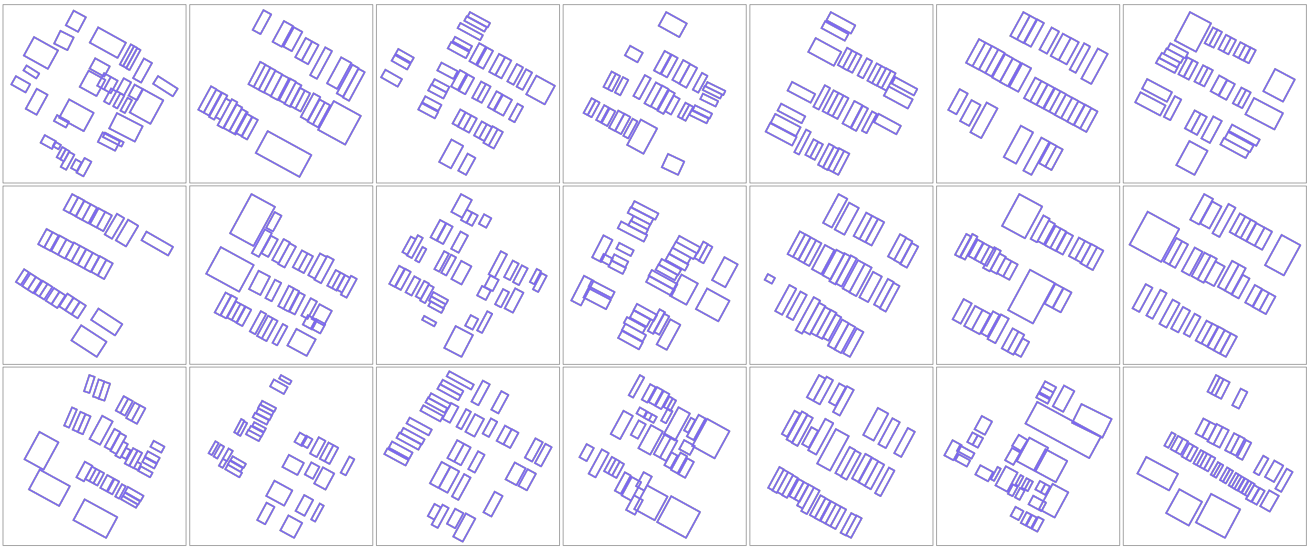


Figure VIII. 2D generation results of AETree trained on the NYC Dataset

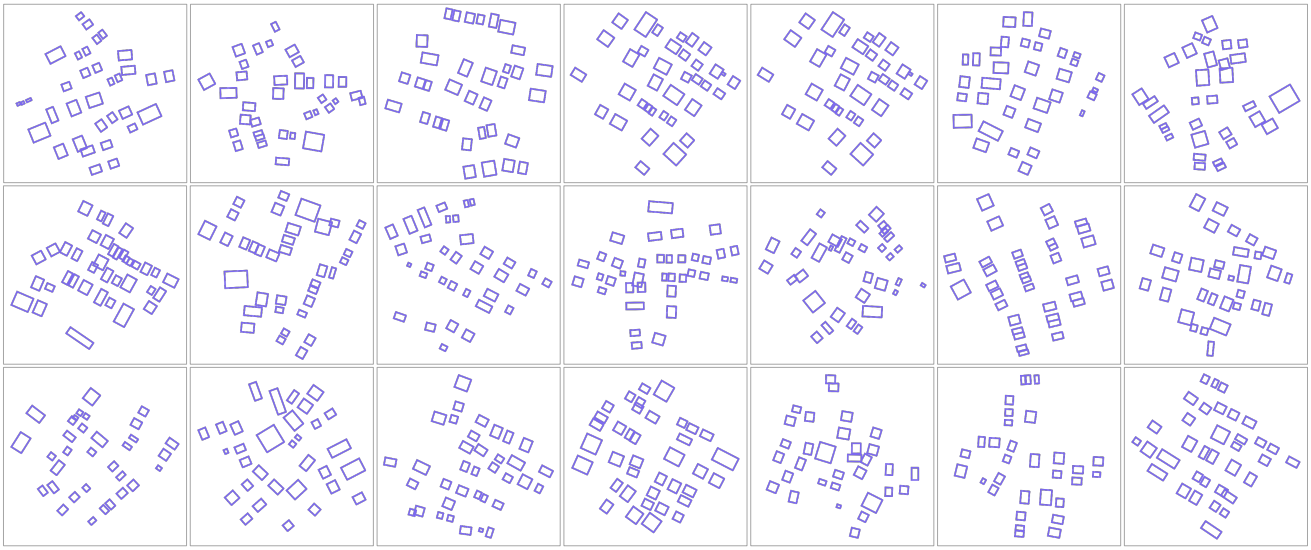


Figure IX. 2D generation results of AETree trained on the Zurich Dataset