# HDMapGen: A Hierarchical Graph Generative Model of High Definition Maps

Supplementary Materials

In the supplementary materials, we describe the statistics of HD map datasets in Section 1; more qualitative results of generated maps in Section 2; and details of HDMapGen models in Section 3.

## 1. Dataset Statistics

In this section, we introduce the statistics of the HD map datasets in Table 1. In the plain graph setting, we have a maximum of 250 nodes (control points) for the Argoverse dataset and 498 nodes for our In-house dataset. After representing the map data as hierarchical graphs, we convert 30% of the original control points into global graph nodes and 70% of which into local graph nodes. For the local graph generation, as we allow a variable number of local nodes in each lane, we set a maximum length of $W$ with a node validity mask. $W$ is set to 8 for the Argoverse dataset and 20 for the In-house dataset.

## 2. Qualitative Results

### 2.1. Global Graph Diversity

In Figure 1, we demonstrate more generated global graph results as the temperature changes during the diversity control. We show that more novel HD map patterns (three intersections or more parallel lanes) are generated when a large temperature $\tau$ is applied, yet the quality-diversity trade-off still exists.

### 2.2. HDMapGen Generation Quality

We show more results from HDMapGen, generated maps with a FoV of $200m\times200m$ in Figure 2, and generated maps with a FoV of $400m\times400m$ in Figure 3. The average number of global nodes in the smaller maps is 43, and the average number of global edges is 50, while for the larger maps, the average number of global nodes is 136, and the average number of global edges is 175, which means more than four times of nodes. While our proposed HDMapGen still achieves promising results for large graph generation, which demonstrates its large scalability.

Notice that an issue with our current results is the unsmoothness of the local graph in some generated samples. However, all of our demonstrated results are not post-processed or applied with any heuristic thresholding or filtering during generation. One can expect a better version with additional steps as described above.

## 3. Model Details

In this section, we introduce more details of baseline and two variants of HDMapGen models.

### 3.1. Global Graph Generation: Two Variants

In Figure 4, we show the neural network structures of the other two variants of HDMapGen using *topology-first* or *independent* for global graph generation.

***Topology-first***: At the step $t$, the model firstly generated the topology $L_t$ to generate the connections of node $t$ with previously generated nodes. And then taking the new topology information $L_t$ as additional inputs to the GRAN model to generate the spatial coordinates $C_t$ of node $t$.

***Independent***: At the step $t$, the model is trained to generate $C_t$ and $L_t$ simultaneously, while not consider any dependence between these two variables.

### 3.2. Baselines

In this work, we implement two baselines SketchRNN [2] and PlainGen. For SketchRNN which uses a sequence generative model, we use "layer norm" model as our encoder and decoder model. We also apply different temperatures to fine-tune a better output. For a plain graph generative model, we use PlainGen, a model derived from our global graph generation step of HDMapGen. Notice that NTG [1] which is designed for a more coarse road layout generation, also uses a plain graph generative model. However, since the model has not been open-sourced yet, we are not able to perform a comparison with NTG on this high-definition map generation task.

## References


[1] Hang Chu, Daiqing Li, David Acuna, Amlan Kar, Maria Shugrina, Xinkai Wei, Ming-Yu Liu, Antonio Torralba, and Sanja Fidler. Neural turtle graphics for modeling city road layouts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4522–4530, 2019.

[2] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

| Graph Type | | Plain Graph | | | | | Global Graph | | | | | Local Graph |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Component | | #Nodes | | #Edges | | | #Nodes | | #Edges | | | #Nodes |
| Dataset | City | Max | Mean | Max | Mean | No edge/Edge | Max | Mean | Max | Mean | No edge/Edge | Max |
| Argoverse | MIA | 250 | 147 | 267 | 151 | 164 | 112 | 43 | 138 | 50 | 40 | 8 |
| Argoverse | PIT | 250 | 178 | 265 | 185 | 187 | 111 | 51 | 134 | 64 | 44 | 8 |
| In-house | SF | 498 | 164 | 537 | 166 | 188 | 100 | 47 | 135 | 52 | 48 | 20 |

Table 1: Graph statistic for Argoverse Dataset (Miami and Pittsburg) and in-house Dataset (San Francisco): number of nodes, number of edges, the ratio of no edges v.s. edges (sparsity level) in a plain graph or a hierarchical graph (including both global graph and local graph).

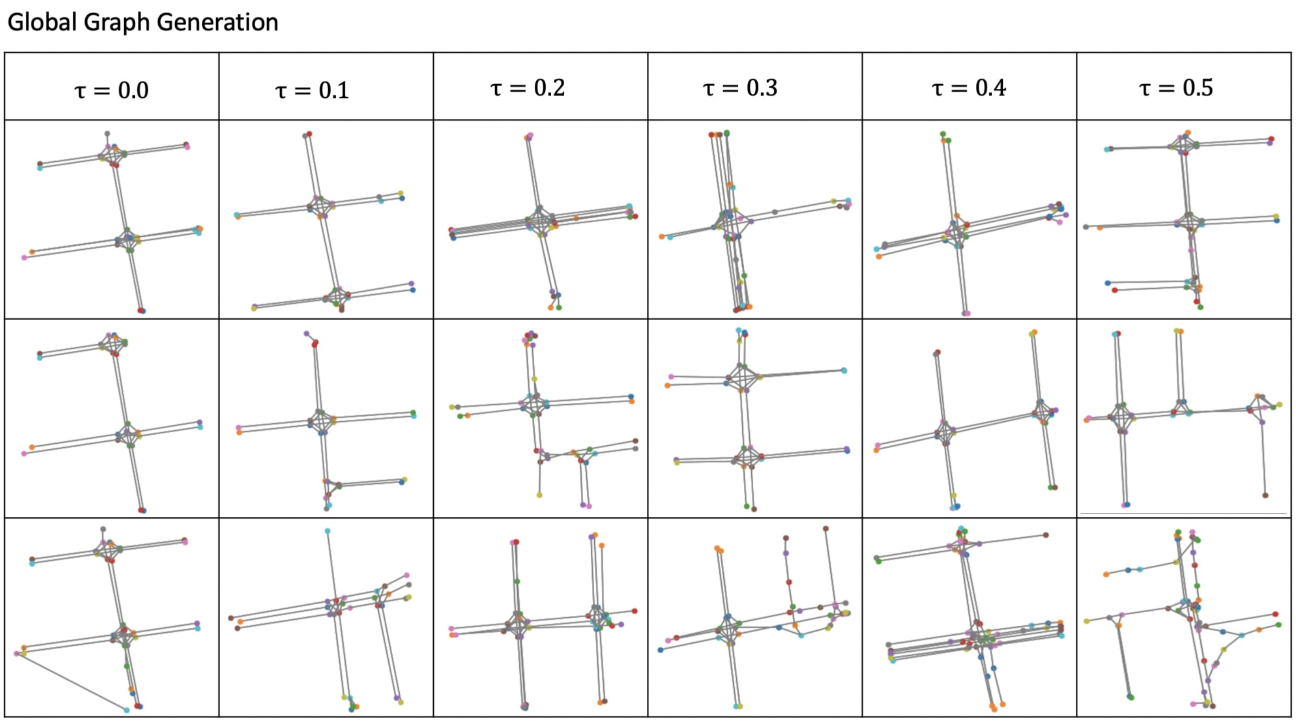


Figure 1: Diversity control for global graph generated from HDMapGen. Outputs generated with different temperatures $\tau$.

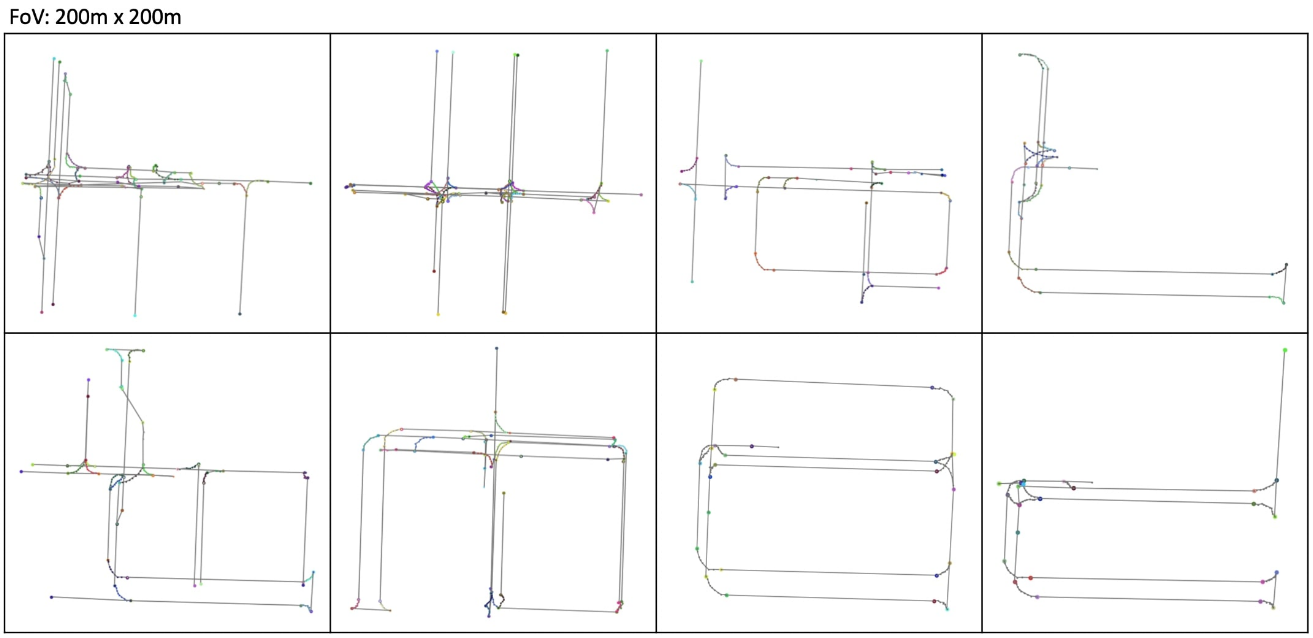


Figure 2: Hierarchical graph with a FoV of $200m \times 200m$ generated from HDMapGen.

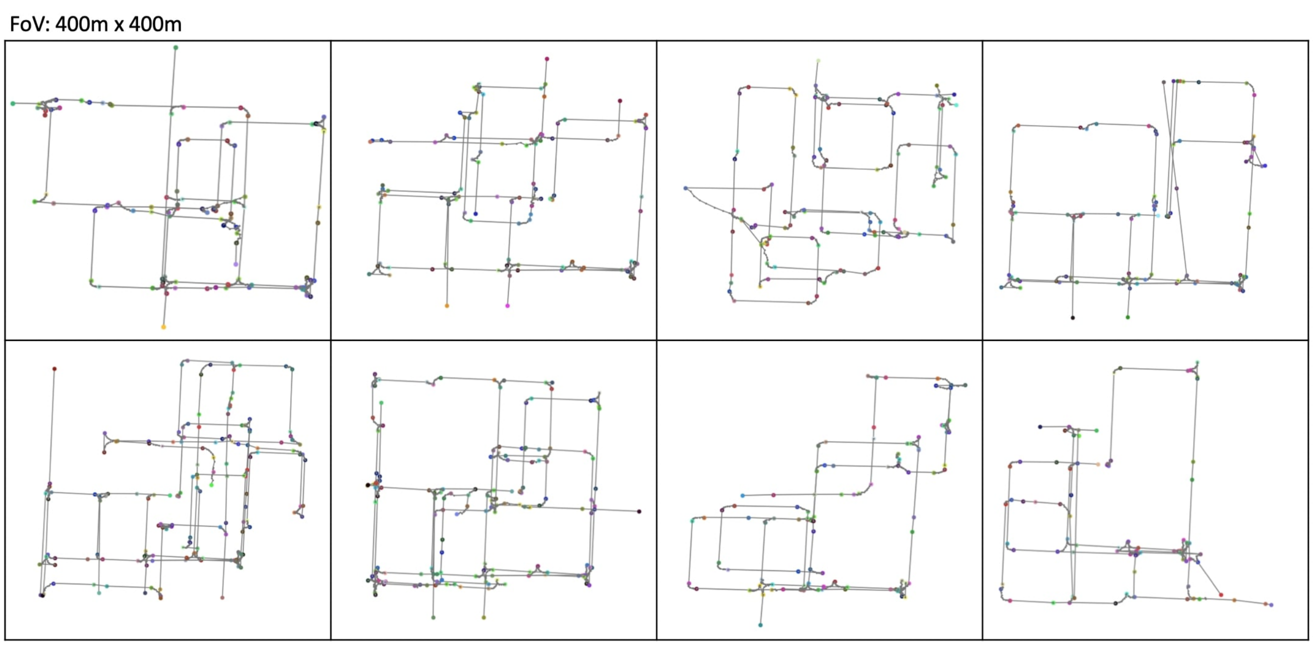


Figure 3: Hierarchical graph with a FoV of $400m \times 400m$ generated from HDMapGen.

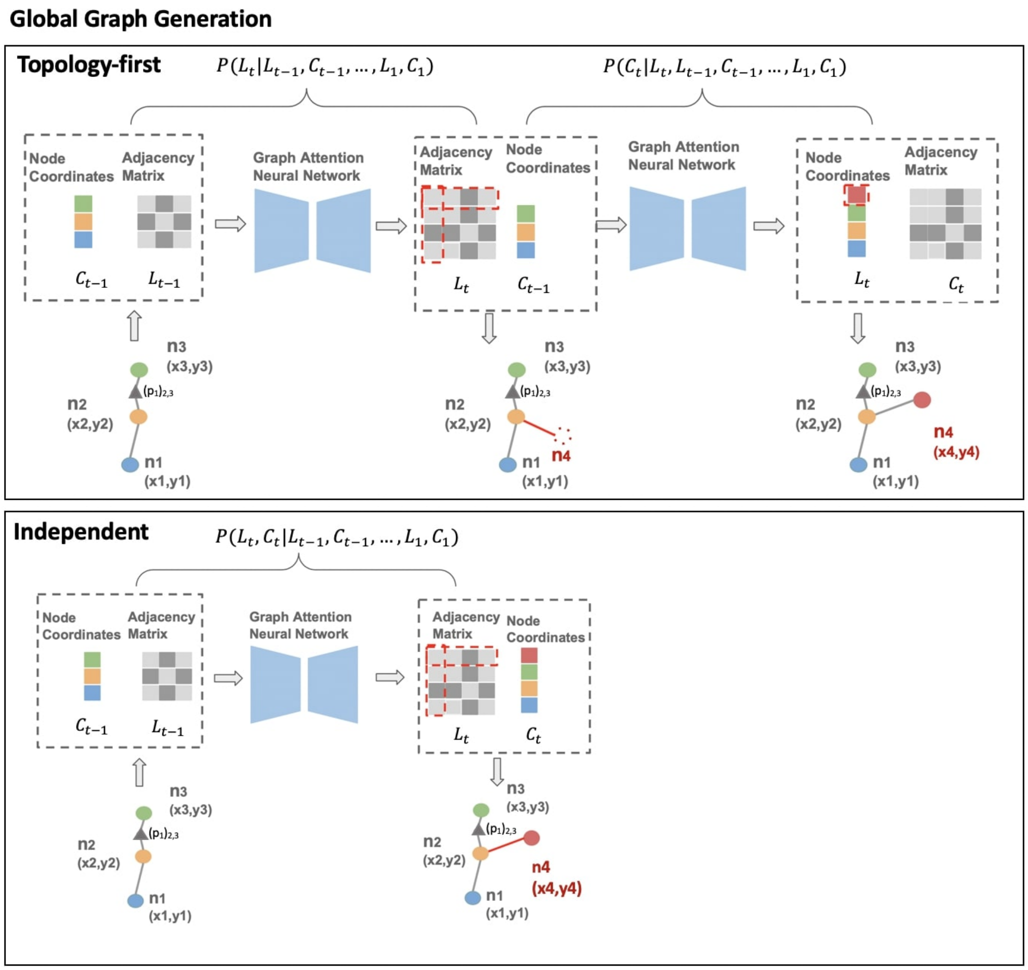


Figure 4: Two variants of HDMapGen: *topology-first* and *independent*. The models consider different dependence and priority to generate coordinates and topology at each step.