2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

FloorplanNet: Learning Topometric Floorplan Matching for Robot Localization

Delin Feng[1*], Zhenpeng He[2*], Jiawei Hou[1], Sören Schwertfeger[1], Liangjun Zhang[2]

***Abstract*— Given a building floorplan, humans can localize themselves by matching the observation of the environment with the floorplan using geometric, semantic, and topological clues. Inspired by this insight, this paper proposes a learning-based topometric robot localization method $FloorplanNet$, which implements a match between a metric robot map and the potentially inaccurate building floorplan in nonuniform scales and different shapes by semantic information. The method uses a novel Graph Neural Network to learn descriptors of nodes from topometric graphs generated from the input maps. We demonstrate that our method can match the 3D point cloud sub-map generated by the robot during the SLAM process with the 2D map. Furthermore, we apply our map-matching algorithm for real-world robot localization. We evaluate our method on several publicly available real-world datasets. Even though our network is solely trained using simulation data, our method demonstrates high robustness and effectiveness in real-world indoor environments and outperforms the existing SOTA map-matching algorithms. We further develop a simulator that automatically creates and annotates the required training data to train our neural networks. The method and simulator are released at: https://github.com/fengdelin/FloorplanNet.git**

I. INTRODUCTION

Robot localization is critical for a robot to navigate an indoor environment and achieve a specific goal [1], [2]. Floorplans of indoor buildings contain useful structure information about building interiors for localization. Furthermore, floorplans are often easy to acquire as most buildings have floorplans. A human often uses floorplans to locate themselves. For instance, a human finds out where he or she is by walking around, comparing the observed scenery with the floorplan, leveraging high-level semantic cues rather than precise depth information.

By making use of the floorplan or sketch map, the robot localization problem is reduced to the map matching between the floorplan and robot map constructed by the sensors mounted on a robot. Through the matching, the robot can obtain its position from the pre-known floorplan map with semantic cues.

In general, the problem of robot localization using the floorplan, however, is challenging. The robot map and the floorplan have nonuniform scales and different shapes; moreover, the robot map could be incomplete. Without topological clues, it is challenging for a geometric-based map-matching algorithm to deal with those mentioned situations.

Many approaches have been proposed for robot localization based on map matching. However, most of the previous works rely on assumptions that maps have similar modalities, the same scale, or an initial guess for the alignment [3], [4].

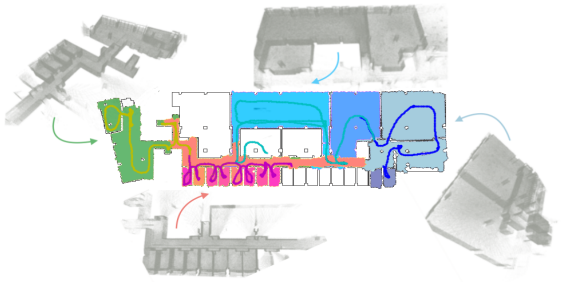

Fig. 1. Robot coarse localization under inaccurate floorplans. The gray point cloud maps are sub-maps obtained by selecting different starting points to explore in the real-world environment. After the point cloud maps are segmented, the matching and relative transformation of the robot maps and the floorplan are generated by the proposed method. As shown, the 2D robot maps (colored blocks) and motion trajectories (colored curves) are correctly projected onto the floorplan (black contours) by our algorithm.

In this paper, we present a new way of thinking about multimodal map matching – a Graph Neural Network (GNN) based learning method $FloorplanNet$ for robot localization, which is inspired by Superglue [5]. As outlined in Fig. 2, it works by first generating topological graphs from the segmentations of the input maps. The floorplan global graph is generated by segmenting the input 2D grid map using a morphology method [6], while the input 3D point cloud of the robot local map is segmented using our previous approach for topometric map representation [7]. The core of our paper is graph matching, which is done using a Graph Attention Network (GAT) [8] trained to provide distinctive descriptors for the nodes of the two input graphs. We extend the baseline GNN by adding an edge mapping module, topological constraints and spatial location constraints. Finally, the Sinkhorn algorithm [9] is used to solve the problem of finding the best pairing of vectors between the two graphs.

Robots can generate local point cloud maps by using Simultaneous Localization and Mapping (SLAM) [10], [11]. With map matching, our method can obtain the relationship between a robot map during the SLAM process and a floorplan (or global map), and estimate the correct scale, translation and rotation between the maps, as shown in the experiment depicted in Fig. 1. The powerful representation capability of neural networks allows us to complete the matching quickly, effectively, and robustly in complex environments, leveraging abundant information such as semantics and topology. Additionally, the graph matching method makes it possible to relate the 3D robot map to the 2D floorplan in nonuniform scales and different shapes, which is difficult to do in other work.

[1] School of Information Science and Technology, ShanghaiTech University, Shanghai, China. {fengdl, houjw, soerensch}@shanghaitech.edu.cn
[2] Robotics and Auto-Driving Laboratory (RAL), Baidu Research.
* Work done during an internship at RAL.

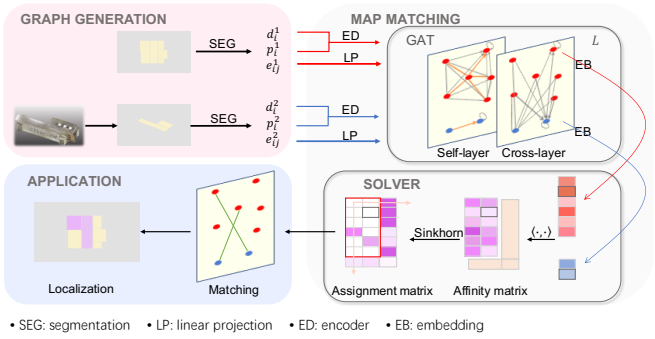

Fig. 2. Architecture of our $FloorplanNet$ algorithm for robot localization.

The key contributions of this paper are:

- We propose a novel GNN-based learning architecture, which can effectively encode spatial position, topology and geometry features to match partial robot maps with complete prior maps of the environment. Our method can handle partial robot maps with errors due to sensor measurements and floorplan maps with errors in scale.
- We formulate a robust graph generation method to transfer the robot map and floorplan to topological semantic representations. We demonstrate the map matching algorithm application for robot localization in the real world and show the effectiveness of our method.
- We further contribute a simulator for creating diverse and automatically labeled training data for supervised learning algorithms working on 2D floorplan matching.

II. RELATED WORK

A. Map Matching

Map matching aims to align two maps that are possibly in different modalities, e.g. robot maps and CAD models.

Considering that a 2D map can be treated as an image, feature-based image matching methods, for example, image-based matching with SIFT [12] were used initially. This performs well on high-textured images, but building structures in an occupancy grid map are mostly just lines in some main directions [13] and often depict quite self-similar environments (e.g. rooms, corridors). Region segmentation is conducive to matching maps that have different types of noise and there is only partial overlap between them. Saeedi *et al*. [4] segment the map and generate a bounding box for each region for searching the correct transformation. This method deals with maps in different scales and modalities well because the bounding boxes hide the detailed features of the regions and can change the region scale easily. We will compare our approach with theirs in our experiments.

The hypothesis clustering method in the work of Hou *et al*. [14], [15] is an intermediate approach of feature-based matching and region matching. They cluster the transformations computed from the matched regions to align the maps. However, this algorithm cannot deal with the maps in different scales due to the reason that it uses region size as a feature for matching.

B. Graph matching

One can abstract grid maps as graphs and thus treat the map matching task as a graph matching problem, which needs to consider the similarity between nodes and edges simultaneously. It is an NP-hard quadratic assignment problem. Recently, many approximation algorithms have been proposed to solve this problem efficiently and accurately.

The Hungarian algorithm [16] is based on the idea of sufficiency proof in Hall's theorem. This classic algorithm's core is to use augmenting paths to find the maximum matching of bipartite graphs. In [17] topology graphs of 2D occupancy grid maps were matched by assigning certain features to edges and nodes and then propagating similarities of possible matches over their neighbors.

Random walk is an essential algorithm in graph theory, which can calculate weights for nodes in a graph. [18] transforms the matching of two graphs into a single graph, converting the correspondences and the similarity information into the association graph. [19] proves that the random walk algorithm based on the association graph is equivalent to the spectral decomposition algorithm.

Andrei *et al*. [20] first combine end-to-end deep learning with the graph matching problem, extracting image features through a CNN to construct an affinity matrix and solving this matrix by a spectral method [19]. The method is based on the spatial distances between corresponding pairs rather than actual correspondences. Li *et al*. [21] propose a GNN-based graph matching network (GMN) with a cross-attention matching mechanism for computing similarity scores between a pair of graphs. This method proves that GNN can generate graph embeddings for similarity learning. The work most similar to ours is the Superglue network proposed by Sarlin *et al*. [5], which matches two sets of local features by jointly finding correspondences and rejecting non-matchable points. This method utilizes a GNN for graph embedding, aggregates attention to nodes, and reduces this task into an optimal transport problem. Compared with those GNN methods, we make full use of the characteristics of the edges in the graph.

C. Floorplan Localization

For global localization in a pre-known (floorplan) map, the Monte Carlo method is commonly used [22], [2], [23], [24]. There are two problems here. First, in most cases, the method requires the map and the current sensors' observation to be on the same scale. Secondly, inconsistencies between the pre-known map and the observation can lead to positioning errors. For example, often there is no furniture in the pre-known map, but there is furniture in the observation, or the pre-known map is inaccurate.

Global localization leverages interesting features in maps to register the local map with the pre-known global map to calculate the relative transformation and obtain the current position of the robot [25], [26], [1]. Since the floorplan is a 2D map that contains only simple features like rooms' contour and no texture information, the localization under inaccurate floorplans is usually based on regions and corners.

III. GRAPH GENERATION

The floorplan and the 3D robot point cloud map are segmented as shown in Fig. 2. In order to ensure the consistency and robustness of the representation of the multimodal map, each region represents a node. On this basis, we add shape features, spatial location, and semantic features with scale, rotation, and translation invariance to nodes, and generate edge relationships and features at the same time.

A. Floorplan Map Segmentation

We use the morphology method [6] for region segmentation. In this method, first, a map is represented as a binary image of the traversable and non-traversable areas. Second, we divide the traversable areas by repeating a morphological erosion operation [27] and assigning labels. Third, the representative points of regions are found from the points of erosion. Finally, the boundaries are found by backtracking the erosion process.

Naturally, we derive crucial semantic labels ("room" and "door") that will be assigned to subsequent graph generation.

Fig. 3. Region segmentation of floorplan.

B. Robot Map Segmentation

We use point clouds as input for region segmentation. In our previous work [7], we can cluster the space through the height information to get the region candidates using the priors that doors connect rooms and the height of the top of doors is lower than the ceiling height of rooms. Due to the occlusion and clutter in the environment, usually, a room is divided into multiple regions. We cluster these regions with noise leveraging prior knowledge. The result is shown in Fig. 4.

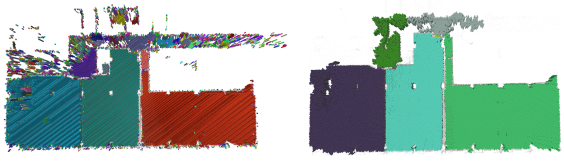

Fig. 4. **Left**: All the region candidates of the robot map before clustering. **Right**: All the regions of the robot map after clustering.

For robots, whether different regions are neighbors is not only determined by the proximity of spatial locations but also depends on whether there are paths between regions. According to the assumption that rooms connect with other regions by doors, our goal of region clustering is to find the "room" - "connection" - "room" pattern in the map and cluster those candidates as one region. We chose those candidate vertices that exceed the size threshold a_{th} as "region" seeds. We use a breadth-first search that starts from each seed and stops when it meets other seeds to generate regions.

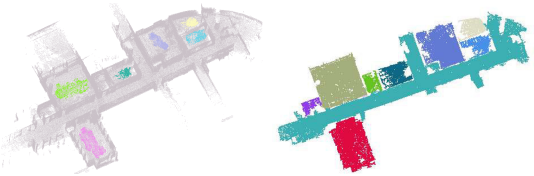

Fig. 5. **Left**: The robot map where furniture is annotated in color. **Right**: The robot map after segmentation.

C. Graph Generation

1) Formulation: We adopt the segmented regions as the vertices of the graph. Edges represent the connections between a vertex labeled "door" and its adjacent vertices labeled "room". We generate two undirected graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ which represent global map and local map, respectively: $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, where $\mathcal{V}_1 = \{1, 2, \ldots, N\}$, $N \geq 2$ is the set of nodes in $\mathcal{G}_1$, $\mathcal{E}_1 \subseteq \{(i,j) : \ i,j \in \mathcal{V}_1\}$ is the set of undirected connectivity links in $\mathcal{G}_1$, $\mathcal{V}_2 = \{1, 2, \ldots, M\}$, $M \geq 1$ is the set of nodes in $\mathcal{G}_2$, $\mathcal{E}_2 \subseteq \{(i,j) : \ i,j \in \mathcal{V}_2\}$ is the set of undirected connectivity links in $\mathcal{G}_2$. Each graph denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ associates a node feature vector x_i with each node . Each edge $(i,j) \in \mathcal{E}$ is associated by an edge feature vector e_{ij}.

2) Region Vertex Descriptors: Detecting key points and generating suitable features for each vertex is the first and the crucial step of our map matching. The the final vertex descriptor contains 7-dimensional Hu-moments [28], a 15-dimensional Fourier-descriptor, 2-dimensional vertex position, 1-dimensional area size, 2-dimensional vertex label (0 for "door" and 1 for "region") and 1-dimensional confidence. Therefore, in our paper node feature vector is $d_i \in \mathbb{R}^{28}$.

3) Edge Features: The features of the initially generated edges use 0 and 1 to represent the connectivity between nodes ($e_{ij} \in \mathbb{R}^1$).

IV. GRAPH MATCHING

In an environment consisting of regions with similar geometric structures, the global-local maps matching task remains challenging. In the process of robot exploration, the connectivity between regions can enrich the local map's information and provide support for subsequent applications such as path planning. We present a graph matching network that takes the graph's topological relationship and geometric context into account, as Fig. 2 shows.

Problem Formulation: Given two undirected graphs which represent the global map and local map, $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, our goal is to establish one-to-one node correspondences between two graphs.

We use an indicator matrix $X \in \{0,1\}^{N \times M}$, where $X_{ij} = 1$ implies that node $i \in \mathcal{V}_1$ is matched to node $j \in \mathcal{V}_2$, otherwise there is no matching between them.

Using the random walk or GNN idea, the edge information is aggregated into the nodes to solve the node assignment problem as follows:

$$\begin{aligned} \max_{X} \quad & \sum_{i,j} S_{ij} X_{ij} \\ \text{subject to} \quad & \forall j \ \sum_{i} X_{ij} = 1, \ \forall i \ \sum_{j} X_{ij} \leq 1, \end{aligned} \tag{1}$$

where $S \in \mathbb{R}^{N \times M}$ is the score matrix.

Graph Matching Network: From the perspective of the characteristics of maps, we design robust and suitable node features and propose a matching network based on GAT [8] with an edge mapping module, topological constraints and spatial location constraints, as shown in Fig. 2. Superglue is a learning feature matching method with GNN and learns priors over geometric transformations and regularities of the 3D world through end-to-end training from image pairs. Thus, we adopt Superglue as our network backbone to enrich the spatial representation ability.

1) ***Graph encoder:*** We use the MLP layers to encode the nodes' position p_i and features d_i mentioned above. This encoder maps nodes' position into high dimensions, and it can make the receptive field wider. h_i^t denotes the t-th hidden layer respect to the i-th node.

$$h_i^{(0)} = MLP(p_i) + d_i, \ \forall i \in \mathcal{V} \tag{2}$$

2) ***Propagation layer:*** The propagation layer maps nodes of the current layers to a new representation through aggregating neighbor nodes information by edges weight, as follows:

$$h_i^{(t+1)} = f_{update}(h_i^t, m_{\mathcal{E}\to i}), \ \forall i \in \mathcal{V} \tag{3}$$

where f_{update} is an MLP neural network. It could also be other recurrent neural network cores such as RNN, and LSTM [21].

$\mathcal{E} \in \{\mathcal{E}_{self}, \mathcal{E}_{cross}\}$, $\mathcal{E}_{self}$ is the set of edges in a graph, and $\mathcal{E}_{cross}$ is the set of connected edges between two graphs. Recent research [5] and [21] have shown that establishing cross propagation layers between two graphs can produce robust graph matching.

3) ***Attentional Aggregator:*** We also import the attention mechanism [8] to aggregate neighbor nodes and edges. The advantages of attention are as follows: when applying multi-head attention, the computation can be completely parallelized. It implicitly assigns different weights to the same neighbor of a node, thereby expanding the expressiveness of the model and improving its interpretability of the model. The attention mechanism is applied to all edges of the graph in a shared way, which benefits that it can be directly applied to inductive learning. We can get the message propagated between layers:

$$\begin{aligned} m_{\mathcal{E}_{self}\to i} &= f_{message}(h_i^{(t)}, h_j^{(t)}, e_{ij}), \ \forall (i,j) \in \mathcal{E}_{self} \\ m_{\mathcal{E}_{cross}\to i} &= f_{message}(h_i^{(t)}, h_j^{(t)}), \ \forall (i,j) \in \mathcal{E}_{cross} \end{aligned} \tag{4}$$

Let H denote the embedding layer as the input of the attention layer. Each layer has different parameters. We learn classic embedding methods used in tasks such as term matching and database retrieval, and embed the inputs as vectors, as follows:

$$Q_i = H^{(t)}(q_i), K_i = H^{(t)}(k_i), V_i = H^{(t)}(v_i), \tag{5}$$

where Q_i is a query for node i, the constituent elements in the query Q_i are composed of a series of key and value data pairs denoted by K_i and V_i. Here q_i is the output descriptor of graph encoder layer $h_i^{(0)}$, k_i and v_i are the output descriptors need to be retrieved $h_i^{(0)}$ or $h_j^{(0)}$. Given an element in the target Q_j, we obtain the weight coefficient of corresponding value V_i by calculating the similarity or correlation between Q_j and each key K_i. Then the weighted summation of the value is then obtained, that is, the final attention:

$$\alpha_{ij} = Softmax(Q_j^T K_i), \tag{6}$$

where α_{ij} is the attention weight. We can easily aggregate all nodes in the two graphs with attention weights. However, for graphs, nodes are usually connected by edges. Therefore, known topological relationships are crucial.

$$\begin{aligned} m_{\mathcal{E}_{self}\to i} &= \sum_j \left(\alpha_{ij} \odot \phi(e_{ij})\right) V_j, \ \forall (i,j) \in \mathcal{E}_{self}, \\ m_{\mathcal{E}_{cross}\to i} &= \sum_j \alpha_{ij} V_j, \ \forall (i,j) \in \mathcal{E}_{cross}, \end{aligned} \tag{7}$$

where $\odot$ denotes element-wise product, and $\Phi(e_{ij})$ is a function defined by connectivity between nodes and relative spatial positions jointly. In the experiment, we use two different functions to implement. Admittedly, this function also can be a simple projection network to learn how to express edges [29].

Thus, all nodes in the same graph update in the same way. We aggregate the nodes of the entire graph $\mathcal{V}_1$ to obtain matching descriptors f_i^1, as follows:

$$f_i^1 = W_H h_i^1 + b_H, \forall i \in \mathcal{V}_1, \tag{8}$$

and similarly for $\mathcal{G}_2$.

4) ***Solver:*** We calculate the affinity matrix S of two graphs as Eq. 1.

$$S_{ij} = \langle f_i^1, f_j^2 \rangle, \ \forall (i,j) \in \mathcal{V}_1 \times \mathcal{V}_2, \tag{9}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product which indicates similarity, $\times$ between sets is the Cartesian product.

Here we adopt the Sinkhorn algorithm [9] as our solver. It performs row normalization and column normalization on the score matrix alternately, and finally converges to a doubly stochastic matrix X.

5) ***Constraints:*** We use the same dustbins technology as superglue to avoid node mismatch [5]. Pairs that meet the threshold are valid pairs that are not thrown into the dustbins, and their number is denoted as $\bar{N}$. Since robot exploration is an incremental process, we grade the thresholds.

6) ***Loss:*** Our method utilizes a one-to-one node ground truth match $\mathcal{M}$ for supervised learning while adding edge topological error and the spatial position error between matching nodes as penalty terms.

$$\begin{aligned} Loss_{match} &= -\sum_{(i,j)\in\mathcal{M}} \log X_{ij}, \\ Loss_{topo} &= 1 - \frac{\sum_{(i,j)\in\mathcal{V}_1\times\mathcal{V}_2} |e_{ij} - e\_gt_{ij}|}{\bar{N}^2}, \\ Loss_{spatial} &= -\frac{\sum_{(i,j)\in\mathcal{V}_1\times\mathcal{V}_2} \|p_i - p\_gt_j\|}{\bar{N}}, \\ Loss &= a \cdot Loss_{match} + b \cdot Loss_{topo} + c \cdot Loss_{spatial}, \end{aligned} \tag{10}$$

where a, b, c denotes the weights for different losses, e_gt and p_gt are the ground truth of edge feature and position

feature. Since the $Loss_{match}$ cannot be calculated correctly when the element in the allocation matrix X_{ij} is 0, the actual experiment $Loss_{match}$ is the negative log likelihood of X_{ij} plus a small value (ϵ).

V. EXPERIMENTS

A. Datasets

1) Data Generator: One of the key questions is whether networks can be trained on synthetic data which is augmented so that they can still achieve robust performance on real-world data. It is especially important in our task when real-world data is not sufficiently available.

We, therefore, develop a data simulator that contains 1) a grid map generator and 2) a matching map generator. The grid map generator on the backbone of [30] is used to create various 2D floorplans composed of labeled rooms, doors and various corridors. The matching pair generator can sample partial regions of the floorplan and produce the matching region pairs. Moreover, our matching map generator can produce matching region pairs between the floorplan and its partial samples with various noise and clips to augment the data. All transformation parameters are randomly chosen inside intervals whose bounds can either be specified or are constrained by spatial relations. Our data simulator is open-source and can be used for semantic supervised learning methods working on occupancy grids. Examples are shown in Fig. 6.

2) Comparison Data: We train and evaluate on three different datasets: synthetic data, real data, and challenging data. In the training data of the network, we use the grid map generator to create 500 synthetic floorplans. For the experiment with real data additionally use 50 real-world floorplans. Furthermore, we use the matching map generator to sample 50 matching maps (simulated robot maps) for each floorplan. The ground truth is the affine matrix between two maps given by the matching map generator.

The real-world floorplans are selected from Bormann's [6] and Carpin's [13] dataset. The dataset from Bormann is used as different modalities map matching, and the one from Carpin offers partial overlap cases, which includes the Fort AP Hill dataset collected with four robots mapping the same environment [31] and the Radish dataset collected with two robots mapping the same building.

In the challenging data, the synthetic floorplans are additionally cut into different small areas to make them more similar to the robot maps collected in real time.

3) Validation Data: In the validation, we explore several indoor buildings with a mobile LiDAR sensor. The 3D point cloud maps are generated from a SLAM process during the exploration.

Setup: In the network training phase, we use $L = 9$ layers of alternating multi-head self-attention and cross-attention with 2 heads each, and perform $T = 20$ Sinkhorn iterations. If not specifically mentioned, the default loss parameters are $a = 1$, $b = 0.25$, $c = 0.5$.

Evaluation metrics: The matching performance of our method is evaluated using one metric from correspondence ground truth (MS) and two (AUC, TS) from affine transformation ground truth: i) Matching score (MS) reports correct correspondences as a proportion of all output vertex correspondences. ii) AUC [5]. iii) Transformation score (TS) illustrates the average correctness of the computed transformations. The two maps are matched correctly when the errors between the estimated transformation against the ground truth transformation are below the thresholds (translation threshold is 0.35 rad, rotation threshold is 0.08 rad and scale threshold is 0.07).

B. Homography Estimation

We perform homography transformation estimation on synthetic, real and challenging data, respectively. We get the graph matching results through the network and calculate the homography transformation through the RANSAC algorithm. The qualitative matching effect is shown in Fig. 6. It can be seen that our network can match the local map and the nodes in the floorplan well.

We also compare our method with others, including some widely-used methods, e.g., nearest neighbors add RANSAC, a learning method based on GNN, our backbone Superglue [5], and a recent work based on geometry called Shahbandi [4]. The quantitative results are shown in Tables I and II . Since the node selection of Shahbandi is quite different from ours, its matching score is NAN. Our matching method outperforms the baselines on all three datasets, e.g. for transformation score (TS) by 20% - 35%.

Notably, our model on synthetic datasets still has excellent performance when transferred to real datasets directly. After training on a complex real-world dataset, the performance on the real dataset is MS=60.534%, TS=72.000%. For supervised learning work, it can thus significantly save data annotation costs and reflect the robustness and generality of the graph model.

C. Ablation Studies

In Section IV, we add a penalty constraint of spatial position and topological relationship to the loss. To verify its effectiveness, we conduct ablation experiments, mainly focusing on the configuration of different losses. The experiments are carried out on a synthetic dataset, and the results are shown in Table III.

Since our design on loss is adding constraints on the assignment result, we use the $loss_{match}$ as the benchmark. The constraints on the spatial position of nodes $loss_{spatial}$ can correct some pairs that are far apart and improve the matching score. The addition of topological constraints $loss_{topo}$ will make the backpropagation process difficult, and in the case of a small spatial position error, the matching of neighbors can be affected by topological constraints. When the base matching effect is better, the joint constraints $loss_{all}$ can play a greater role.

In Section IV, we explore the influence of edge information mapping. This ablation study also illustrates that our edge encoder block is useful and fruitful as shown in Table III.

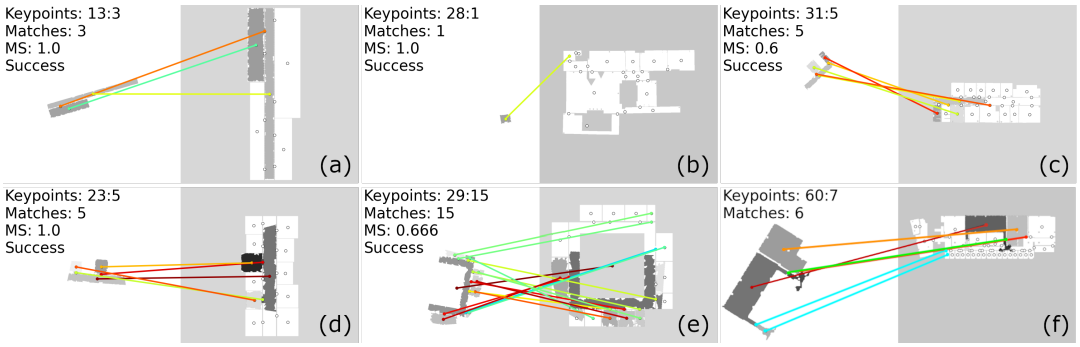

Fig. 6. Qualitative graph matches. (a) is test results on synthetic data, (b) and (c) are test results on real data, (d) and (e) are test results on challenging data, and (f) is test results on a real-world robot map (validation data). The small circles in the figure represent the geometric positions of the nodes in the image, the lines between the images are the predicted node-to-node matching results, and different colors represent different confidence levels. The upper left corner of each sub-figure shows the evaluation results, which include the number of nodes, the matching score and the transformation score. The right half of the figure is composed of the local map to be matched and the global map after an affine transformation. It can be visually judged whether the transformation calculated by the matching is correct.

TABLE I

COMPARISON WITH OTHER METHODS ON SYNTHETIC AND REAL DATA.

Matcher	Synthetic data					Real data				
	Pose estimation AUC			MS	TS	Pose estimation AUC			MS	TS
	@1°	@10°	@20°			@1°	@10°	@20°		
Nearest Neighbor	7.300%	12.449%	13.542%	23.691%	15.200%	4.156%	7.781%	8.391%	16.002%	10.000%
Shahbandi	17.756%	70.154%	76.944%	NAN	84.000%	10.264%	49.082%	60.535%	NAN	56.000%
Superglue	67.758%	77.438%	78.272%	70.071%	81.000%	45.907%	52.301%	53.253%	53.692%	55.000%
Ours	**74.734%**	**83.339%**	**84.067%**	**85.900%**	**90.000%**	**63.034%**	**69.303%**	**69.642%**	**57.393%**	**76.000%**

TABLE II

COMPARISON WITH OTHER METHODS ON CHALLENGING DATA

Matcher	challenging data				
	Pose estimation AUC			MS	TS
	@1°	@10°	@20°		
Nearest Neighbor	1.7%	4.5%	5.5%	9.1%	5.0%
Shahbandi	14.1%	18.2%	26.4%	NAN	22.5%
Superglue	20.9%	34.1%	37.6%	60.5%	47.8%
Ours	**26.2%**	**40.7%**	**44.7%**	**71.2%**	**54.6%**

TABLE III

ABLATION STUDIES ABOUT LOSSES AND EDGE ENCODERS

Matcher	MS	TS	Matcher	MS	TS
$loss_{match}$	80%	87%	w/o edge-connectivity	70%	79%
$loss_{spatial}$	81%	86%	w/o edge-distance	71%	83%
$loss_{topo}$	79%	86%	w/o threshold grading	84%	**90%**
$loss_{all}$	**85%**	**90%**	Full	**85%**	**90%**

D. Localization

It's widely acknowledged that localization under a building sketch or a floorplan is arduous due to the fact that floorplans are often simple and structurally similar. In addition, when the robot explores the indoor environment, it is difficult to judge whether a certain area is completely scanned, so inevitably, there will be many incomplete areas that are disparate from the building floorplan.

We apply our proposed map matching method to the task of global localization under floorplans on a real-world scenario (validation data) collected by a ZEB horizon 3d scanner. Then we input the map and floorplan into our GNN-based map matching network to obtain the matching relationship between nodes. With this matching, we adopt the RANSAC method to calculate the affine transformation of the robot map to the floorplan. Through this transformation, we project the floorplan onto the robot 2d map. Finally, combined with the trajectory of the robot in the real-world coordinate system, we get the localization performance under the floorplan as shown in Fig. 1. It shows that our $FloorplanNet$ framework can complete the global localization task on the building floorplan. The video accompanying the paper also shows an experiment using the real map shown in Fig. 5 where our algorithm performs well in a furnished environment.

It is worth mentioning that the model we use for localization in this real-world environment is the model trained on simulated data. Of course, our simulator also provides the function of data synthesis for a specific floorplan, which can help improve the localization accuracy.

VI. CONCLUSIONS

This paper presents the $FloorplanNet$, a pipeline to relate the 3D robot map to a 2D floorplan with the goal of robot localization. We further contribute a simulator that automatically creates and annotates the required training data to train our neural networks. Even though our network is solely trained using simulation data, our method demonstrates high robustness and effectiveness in real-world indoor environments and is better than the existing SOTA map-matching algorithms, as shown in the experiments.

One of the limitations is that our algorithm needs a good 3D robot map for segmentation. But this could be solved by an offline global registration method such as Quatro [32]. In future work, we will use a learning-based descriptor for graph nodes to improve the matching result of the GNN network. We also believe that a learning-based approach could benefit semantic segmentation in floorplans.

REFERENCES

[1] Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Lost shopping! monocular localization in large indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2695–2703, 2015.

[2] Federico Boniardi, Tim Caselitz, Rainer Kümmerle, and Wolfram Burgard. Robust lidar-based localization in architectural floor plans. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3318–3324. IEEE, 2017.

[3] Daisuke Kakuma, Satoki Tsuichihara, Gustavo Alfonso Garcia Ricardez, Jun Takamatsu, and Tsukasa Ogasawara. Alignment of occupancy grid and floor maps using graph matching. In *2017 IEEE 11th international conference on semantic computing (ICSC)*, pages 57–60. IEEE, 2017.

[4] Saeed Gholami Shahbandi and Martin Magnusson. 2d map alignment with region decomposition. *Autonomous Robots*, 43(5):1117–1136, 2019.

[5] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4937–4946, 2020.

[6] Richard Bormann, Florian Jordan, Wenzhe Li, Joshua Hampp, and Martin Hägele. Room segmentation: Survey, implementation, and analysis. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1019–1026. IEEE, 2016.

[7] Zhenpeng He, Hao Sun, Jiawei Hou, Yajun Ha, and Sören Schwertfeger. Hierarchical topometric representation of 3d robotic maps. *Autonomous Robots*, 45(5):755–771, 2021.

[8] Petar Velikovi, G. Cucurull, A. Casanova, A. Romero, P Liò, and Y. Bengio. Graph attention networks. 2017.

[9] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2), 1967.

[10] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 2022.

[11] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020.

[12] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[13] Stefano Carpin. Fast and accurate map merging for multi-robot systems. *Autonomous Robots*, 25(3):305–316, 2008.

[14] Jiawei Hou, Haofei Kuang, and Sören Schwertfeger. Fast 2d map matching based on area graphs. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, IEEE, 2019.

[15] Jiawei Hou, Yijun Yuan, Zhenpeng He, and Sören Schwertfeger. Matching maps based on the area graph. *Intelligent Service Robotics*, pages 1–26, 2022.

[16] J. Munkres. Algorithms for the assignment and transportation problems. *SIAM. J*, 10, 1962.

[17] Sören Schwertfeger and Andreas Birk. Map evaluation using matched topology graphs. *Autonomous Robots*, 40(5):761–787, 2016.

[18] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *European Conference on Computer Vision*, 2010.

[19] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision*, 2005.

[20] A. Zanfir and C. Sminchisescu. Deep learning of graph matching. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3835–3845. PMLR, 09–15 Jun 2019.

[22] Bahram Behzadian, Pratik Agarwal, Wolfram Burgard, and Gian Diego Tipaldi. Monte carlo localization in hand-drawn maps. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4291–4296. IEEE, 2015.

[23] Wera Winterhalter, Freya Fleckenstein, Bastian Steder, Luciano Spinello, and Wolfram Burgard. Accurate indoor localization for rgb-d smartphones and tablets given 2d floor plans. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3138–3143. IEEE, 2015.

[24] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. Sedar-semantic detection and ranging: Humans can localise without lidar, can robots? In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6053–6060. IEEE, 2018.

[25] Sheng Guo, Hanjiang Xiong, Xianwei Zheng, and Yan Zhou. Activity recognition and semantic description for indoor mobile localization. *Sensors*, 17(3):649, 2017.

[26] Hang Chu, Dong Ki Kim, and Tsuhan Chen. You are here: Mimicking the human thinking process in reading floor-plans. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2210–2218, 2015.

[27] Pierre Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[28] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.

[29] C. Ying, T. Cai, S. Luo, S. Zheng, and T. Y. Liu. Do transformers really perform bad for graph representation? 2021.

[30] Markus Hiller, Chen Qiu, Florian Particke, Christian Hofmann, and Jörn Thielecke. Learning topometric semantic maps from occupancy grids. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4190–4197. IEEE, 2019.

[31] Andrew Howard, Lynne E Parker, and Gaurav S Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *The International Journal of Robotics Research*, 25(5-6):431–447, 2006.

[32] Hyungtae Lim, Suyong Yeon, Soohyun Ryu, Yonghan Lee, Youngji Kim, Jaeseong Yun, Euigon Jung, Donghwan Lee, and Hyun Myung. A single correspondence is enough: Robust global registration to avoid degeneracy in urban environments. *arXiv preprint arXiv:2203.06612*, 2022.