Layout Representation Learning with Spatial and Structural Hierarchies

Yue Bai1, Dipu Manandhar3, Zhaowen Wang4, John Collomosse4, Yun Fu1,2

1Department of Electrical and Computer Engineering, Northeastern University
2Khoury College of Computer Science, Northeastern University
3Centre for Vision, Speech and Signal Processing, University of Surrey
4Adobe Research

Abstract

We present a novel hierarchical modeling method for layout representation learning, the core of design documents (e.g., user interface, poster, template). Existing works on layout representation often ignore element hierarchies, which is an important facet of layouts, and mainly rely on the spatial bounding boxes for feature extraction. This paper proposes a Spatial-Structural Hierarchical Auto-Encoder (SSH-AE) that learns hierarchical representation by treating a hierarchically annotated layout as a tree format. On the one side, we model SSH-AE from both spatial (semantic views) and structural (organization and relationships) perspectives, which are two complementary aspects to represent a layout. On the other side, the semantic/geometric properties are associated at multiple resolutions/granularities, naturally handling complex layouts. Our learned representations are used for effective layout search from both spatial and structural similarity perspectives. We also newly involve the tree-edit distance (TED) as an evaluation metric to construct a comprehensive evaluation protocol for layout similarity assessment, which benefits a systematic and customized layout search. We further present a new dataset of POSTER layouts which we believe will be useful for future layout research. We show that our proposed SSH-AE outperforms the existing methods achieving state-of-the-art performance on two benchmark datasets. Code is available at github.com/yueb17/SSH-AE.

Introduction

Layout design is widely used in user interface (UI), graphics templates, architecture plan, etc. Given the increasing number of these creative products with diverse layout designs available to users, it is important to have scalable approaches to represent the layout in a customized fashion which benefits downstream tasks such as searching, and recommendation. There are recent works (Deka et al. 2017; Patil et al. 2021; Liu et al. 2018; Manandhar, Ruta, and Collomosse 2020; Patil et al. 2020; Li et al. 2019) on layout representation learning which aim to represent a layout sample as a latent vector to support various tasks.

Different from typical visual data, layout is a special data type featuring both visual characteristics and topological relationships of contained elements. In our work, we refer to visual characteristics as term “spatial” and relational patterns among different elements as “structural” and learn layout representation from these two aspects. Both the spatial and structural aspects are critical in practice. For example, when users want to explore design variations with similar content, spatial aspect is more important as it provides cues on how the designs are perceived visually (Bylinskii et al. 2017; O’Donovan, Agarwala, and Hertzmann 2014). On the other hand, when users want to customize designs arrangement, structural properties such as groupings and alignments are critical aspects as these relationships and structure define the blueprint of layout (Yang et al. 2017).

The spatial and structural aspects are seen as an orthogonal pair as any content can be designed into any layout formats. However, both of them exhibit a hierarchical characteristic, decomposed into a multi-level format. Concretely, for spatial aspect, the elements in a layout typically have varying spatial sizes and attain the most visual salience when being viewed at a certain spatial resolution: larger elements (e.g., toolbar and teaser image) serve as an outline forming the top of a hierarchy; smaller elements contain detailed contents relevant to larger elements forming the bottom levels of a hierarchy. Similarly, for structural aspect, a sample hierarchy can be constructed based grouping geometric relation-
Our key technical contributions as follows:

Motivated by these insights, we introduce a novel hierarchical approach that jointly considers both the spatial and structural nature of layout. Our Spatial-Structural Hierarchical Auto-Encoder (SSH-AE) is a self-supervised representation learning framework: 1) a layout hierarchy is decomposed into multiple levels; 2) level-wise features are recursively aggregated capturing layout attributes at different granularities; 3) a two-pathway training strategy orthogonally maintains the trade-off between spatial and structural layout information. Our work differs from the existing methods mainly in two ways: 1) the works in (Deka et al. 2017; Liu et al. 2018) treat layout as images without encoding explicit geometric hierarchy. Although the recent works (Manandhar, Ruta, and Collomosse 2020; Patil et al. 2021) use GNNs to employ geometric features, but they form dense graphs which negates the hierarchical information. In contrast, SSH-AE utilizes rich layout hierarchy information to naturally handle complex layouts and obtain discriminative embeddings; 2) most of the layout representations (Deka et al. 2017; Liu et al. 2018; Manandhar, Ruta, and Collomosse 2020; Patil et al. 2021) are designed only from spatial perspective by training to decode semantic maps. We propose to model the layout with dual perspective capturing both spatial and structural properties. Moreover, in addition to the existing intersection-of-union (IoU) and human evaluation, we also present to use a Tree-Edit Distance (TED) to measure the layout structural similarity. We believe this comprehensive evaluation protocol will help future research to systematically evaluate layout retrieval. We summarize our key technical contributions as follows:

- A hierarchical layout representation learning approach is proposed that recursively extracts coarse-to-fine-grained representation. It enables learning the layout representation at different granularity. Most importantly, it naturally handles the complex layouts with a huge number of components by organizing them into a tree structure.
- We are the first to learn the layout representation by considering both spatial (semantic map) and structural (element organization) perspectives. The SSH-AE handles the dual aspects by training the model with the reconstruction of semantic map and a newly proposed adjacent matrix which defines the structure in the layout.
- A comprehensive evaluation protocol is proposed to systematically measure layout similarity by newly involving TED metric to supplement structural aspect evaluation. We argue that structural similarity is also a necessary aspect compared with the spatial measurement. Our quantitative evaluation shows improved consistency with human subjective evaluation, and enables tuning models for trade-off between spatial and structural similarities.
- We achieve state-of-the-art performance on both RICO and POSTER. The new evaluation protocol and POSTER dataset (to be released upon paper acceptance) are expected to benefit further layout researches.

Related Work

Layout Analysis. Pioneering works (Hurst, Li, and Marriott 2009; Breuel 2003; O’Gorman and Kasturi 1995; Simon, Pret, and Johnson 1997) involve prior knowledge to study document layout structure. Exploring layout from aesthetic angle is a distinctive direction compared with classic vision analysis (Harrington et al. 2004). In addition, numerically analyzing layout needs defining appropriate distance metrics (Ritchie, Kejriwal, and Klemmer 2011; Geigel and Loui 2000), and extract object elements to represent the whole sample layout based on detection techniques (Yang et al. 2017; Swearngin et al. 2018). Several new layout datasets are collected such as RICO (Deka et al. 2017), Floorplan (Wessel, Blümel, and Klein 2008), ICDAR2015 (Antonacopoulos et al. 2015), and PubLayNet (Zhong, Tang, and Yepes 2019). An MLP-based auto-encoder (Deka et al. 2017) is proposed to obtain hidden layout representation used for downstream retrieval. Similarly, a convolutional auto-encoder (Liu et al. 2018) is designed for a better layout retrieval. A GCN-CNN auto-encoder framework (Manandhar, Ruta, and Collomosse 2020) is also developed to extract layout structural patterns using GCN and further improve the retrieval performance. A graph matching based retrieval framework LayoutGMN (Patil et al. 2021) processes a pair of layout as graphs then deploy graph matching algorithm to obtain layout similarity. Our work is related to learning layout representation for search embeddings and closely aligned with (Deka et al. 2017; Liu et al. 2018; Manandhar, Ruta, and Collomosse 2020; Patil et al. 2021).

Graph/Hierarchical Modeling GNNs have been popular recently as they are suitable for modeling topologically structured data (Zhang, Cui, and Zhu 2020). The works in (Manandhar, Ruta, and Collomosse 2020; Patil et al. 2021) have used graph encoding to obtain layout representations. Graph learning algorithms effectively handle non-euclidean but still lose the specificity for data with high hierarchies. Hierarchical modeling further considers the fine-grained structural patterns to learn more discriminative features. The differentiable pooling technique (Ying et al. 2018) is developed for general graph representation learning. The higher-order structural information is extracted to preserve graph hierarchies (Chen et al. 2018). It can be widely used to enhance graph mining methods. Practically, modeling hierarchies benefits visual-related tasks which involve highly architectural data. For instance, StructureNet (Mo et al. 2019) uses a hierarchical graph network to achieve 3D shape generation. Point cloud 3D object detection is realized by utilizing a hierarchical graph module (Chen et al. 2020). Document layout analysis is studied by leveraging the sample hierarchy for different tasks (e.g., generation (Patil et al. 2020) and classification (Simon, Pret, and Johnson 1997)). Similarly, natural language contains even more complex hierarchies. Several hierarchy-based models are proposed for different language tasks (e.g., document summarization (Zhang, Wei, and Zhou 2019; Liu and Lapata 2019) and text classification (Pappagari et al. 2019)). Our work, for the first time, employs the hierarchical modeling to learn layout representation for designs (e.g., UI, posters, and templates).
Figure 2: The illustration of Spatial-Structural Hierarchical Auto-Encoder (SSH-AE). Given an input layout containing a set of elements with hierarchical annotations, we separate elements and construct a tree hierarchy. Then we obtain level-wise layout features based on multi-level encoding and progressive level fusion. The multi-level features are then decoded as semantic segmentation maps and structural adjacency matrices in each level to capture layout information from both spatial and structural aspects. The highest level representation is recursively aggregated from lower levels and serves for downstream retrieval.

Method

We propose Spatial-Structural Hierarchical Auto-Encoder (SSH-AE) to learn discriminative representations for layouts in a self-supervised fashion (see Fig. 2). Our framework jointly considers the given layout from two ways: spatial-structural and multi-level hierarchical aspects. In this way, we represent layouts using hierarchy annotations which are divided into several hierarchy levels based on different spatial resolutions and structural granularities. The encoder is realized by a graph-based network that recursively encodes layouts from coarse to fine-grained levels by conducting a fusion operation. The obtained features are decoded to semantic maps/structural adjacent matrix as supervision of spatial/structural aspects for training (see Fig. 3).

Hierarchy Construction

Given a layout containing design components with corresponding classes (e.g., background, button, and slider) and bounding box coordinates, we organize all the components into a tree hierarchy \( T \) with different depth \( d \). Specifically, the root node (depth \( d = 1 \)) is the background covering the entire design. The leaf nodes with no children are basic design elements in layouts. Several leaf nodes are grouped by geometric alignments and contained by intermediate nodes with edge between them to represent the containment relationship. Such hierarchies are readily available in many design layouts or can be extracted by components geometric alignment (see supplementary for hierarchy extraction used on layouts without original hierarchy annotations). In this way, a layout sample can be represented as a hierarchical tree data format \( T \). We separate \( T \) with overall \( D \) depth into \( L \) levels so that layout information of different scales and granularities can be encoded and aggregated appropriately. As an example, if we have a layout \( T \) with 6-depth \( (D = 6) \), we may separate it into 3 levels \( (L = 3) \), where depth 1/2, 3/4, and 5/6 are grouped into level 1, 2, and 3, respectively. Jointly, we separate the layout from spatial or structural aspects. Each element is denoted as a node \( i \) in tree \( T \). We rank all the nodes according to either their elements’ spatial area \( a_i \) (spatial) or their levels in the hierarchy i.e. depth \( d_i \) (structural), and evenly divide them into \( L \) levels (root node belongs to level 1 and leaf nodes with the largest depth belong to level \( L \)). For either spatial or structural aspect, the complete tree is represented as \( T = T_1 \cup T_2 \cup \ldots \cup T_L \) with nodes \( V = V^1 \cup V^2 \cup \ldots \cup V^L \) and edges \( C = C^1 \cup C^2 \cup \ldots \cup C^L \), where \( C^d \) means all the edges connected from a node in \( V^d \).

Hierarchical Auto-Encoder

Given the layout spatial/structural hierarchy, we utilize an auto-encoder to learn representation in multi-level format. It consists of a multi-level encoding and decoding architecture (see Fig. 2). A multi-level encoder first processes each level individually, and then progressively aggregates the level-wise feature from low to high level by a feature fusion operation. In this way, we obtain an integrated multi-level layout representation. The decoding also adopts a multi-level reconstruction strategy in accordance with the encoding.

Level-wise Encoding Each level \( l \) has a subset of the whole tree \( T^l = \{V^l, C^l\} \) and the encoder is given by

\[
f^l = E(V^l, C^l),
\]

where \( E \) takes the attributes of nodes and edges in \( T^l \) as inputs, and generates the level-wise feature \( f^l \). We use a weight-shared encoder \( E \) for any level \( l \) to encode common layout patterns across all the levels.

Level Fusion All level-wise features \( \{f^1, f^2, ..., f^L\} \) are recursively fused to obtain the entire layout representation from low to high level. In this way, the feature \( f^l \) of each level \( l \) is progressively constructed with more detailed information from lower levels:

\[
f^l = f^1, f^l = U^{l-1}(f^{l-1} \oplus f^l), l \geq 2,
\]

where \( U^{l-1} \) is an MLP that aligns the feature from level \( l-1 \) to \( l \), and \( \oplus \) is the fusion operation which is implemented as summation operation. In this way, each level has a feature that contains integrated information from itself and all levels below it. The multi-level feature set \( F = \{f^1, f^2, ..., f^L\} \) is passed to the decoder during training, and used for downstream retrieval.
Spatial Feature – Level

Structural Encoding. We define the geometric feature for edge \( c_{i,j} \) as

\[
g_{i,j}^c = \left[ \phi_{i,j}, \theta_{i,j}, \frac{\Delta x}{A_i}, \frac{\Delta y}{A_i}, \frac{w_j}{h_j}, \frac{1}{D} \sqrt{\Delta x^2 + \Delta y^2} \right],
\]

where \( \Delta x = x_j - x_i, \Delta y = y_j - y_i, \) and \( D = \sqrt{w^2 + h^2}. \) The orientation \( \theta = \arctan2(\frac{\Delta y}{\Delta x}) \in [-\pi, \pi]. \) \( \phi_{i,j} \) serves as the Intersection over Union (IoU) between node \( v_i \) and \( v_j \) given by \( \phi = \frac{M(v_i \cap M(v_j))}{M(v_i) + M(v_j) - M(v_i \cap M(v_j))} \), where \( M(\cdot) \) represents the single element mask. Since the edge features are calculated based on two corresponding nodes, there is no symmetrical relation and \( g_{i,j}^c \neq g_{j,i}^c. \) Based on this definition, we project each edge feature \( g_{i,j}^e \) for edge \((i, j) \in C\) together with the paired node features with an MLP \( E^c. \) Finally, all the edge features are aggregated with a node-keyed attention module which are given by

\[
f_{i,j}^c = E^c([f_i^v, g_{i,j}^c, f_j^v]), f^c = \sum_{ij} \alpha_{i,j}^c f_{i,j}^c,
\]

where \( \alpha_{i,j}^c \propto \exp(w_c^T [f_i^v, f_j^v]) \) are attention weights with learnable parameter \( w_c. \)

Spatial and Structural Supervision

The feature \( f^l \) for each level \( l \) contains information from level \( l \) itself and other lower levels. To achieve the self-supervised training, we use reconstruction loss for multi-level decoding from both spatial and structural aspects. We refer \( O^l \) as the reconstruction supervision for \( f^l. \) Basically, we use L2 loss \( \|O^l - f^l\|^2\) as the optimization objective to train the decoder of all the levels. Training SSH-AE with spatial and structural hierarchy are achieved by different \( O^l \) implementations. They are illustrated in Fig. 3 and elaborated as below. Please note the level \( l \) in decoding represents the subset containing both current level \( l \) and its lower levels of a given layout sample.

Spatial Pathway: Semantic Map. For each component in level \( l \) of a given layout sample, it has bounding box \((x_j, y_j, w_j, h_j)\) and semantic label \( s_i. \) We construct a multi-channel binary image \( O^l \in R^{h_l \times w_l \times M} \) as semantic segmentation ground truth for the current level. \( M \) is the total number of semantic classes. Each channel \( m \in M \) is 2D binary mask for the components belonging to class \( m. \) \( (w_l, h_l) \) is the spatial size of \( O^l \) which matches the decoder output resolution at level \( l. \) Larger resolutions are used for higher level as more detailed components are included. This spatial pathway focuses more on modeling visual patterns by reconstructing semantic label maps.

Structural Pathway: Adjacent Matrix. Spatial supervision is naturally constructed by the typical decoding recon-
Table 1: Retrieval performance on RICO dataset based on MIoU, TED, and NDCG evaluation metrics. Best results are boldfaced and best ideal values are underlined. The ideal NDCG values are 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>MIoU@1</th>
<th>MIoU@5</th>
<th>MIoU@10</th>
<th>TED@1</th>
<th>TED@5</th>
<th>TED@10</th>
<th>NDCG@1</th>
<th>NDCG@5</th>
<th>NDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE (Deka et al. 2017)</td>
<td>0.430</td>
<td>0.362</td>
<td>0.312</td>
<td>15.960</td>
<td>19.132</td>
<td>19.930</td>
<td>0.325</td>
<td>0.318</td>
<td>0.324</td>
</tr>
<tr>
<td>CAE (Liu et al. 2018)</td>
<td>0.595</td>
<td>0.471</td>
<td>0.440</td>
<td>14.100</td>
<td>16.124</td>
<td>17.976</td>
<td>0.482</td>
<td>0.434</td>
<td>0.416</td>
</tr>
<tr>
<td>SN (Mo et al. 2019)</td>
<td>0.407</td>
<td>0.379</td>
<td>0.360</td>
<td>17.540</td>
<td>19.268</td>
<td>18.994</td>
<td>0.428</td>
<td>0.448</td>
<td>0.466</td>
</tr>
<tr>
<td>GCN-CNN (Manandhar, Ruta, and Collomosse 2020)</td>
<td>0.600</td>
<td>0.514</td>
<td>0.482</td>
<td>15.360</td>
<td>17.644</td>
<td>19.398</td>
<td>0.576</td>
<td>0.564</td>
<td>0.574</td>
</tr>
<tr>
<td>GCN-CNN+Tri</td>
<td>0.617</td>
<td>0.541</td>
<td>0.513</td>
<td>13.820</td>
<td>16.748</td>
<td>17.696</td>
<td>0.601</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LayoutGMN (Patil et al. 2021)</td>
<td>0.446</td>
<td>0.384</td>
<td>0.342</td>
<td>15.174</td>
<td>17.139</td>
<td>18.241</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In every screenshot, every layout is annotated with bounding boxes for design elements. There are totally 25 classes for elements such as “text”, “button”, and “icon”. We follow (Manandhar, Ruta, and Collomosse 2020) to assign 53K samples as training, 13K samples as gallery, and 50 samples as query set. RICO originally only has flat structure, thus, we extract rich hierarchy annotations for its layouts using geometric properties of their elements (see supplementary).

## Experiments

### Datasets

**RICO** (Deka et al. 2017) is largest publicly available dataset of UI layout. It contains 66K samples from mobile apps screenshots. Every screenshots are annotated with bounding boxes for design elements. There are totally 25 classes for elements such as “text”, “button”, and “icon”. We follow (Manandhar, Ruta, and Collomosse 2020) to assign 53K samples as training, 13K samples as gallery, and 50 samples as query set. RICO originally only has flat structure, thus, we extract rich hierarchy annotations for its layouts using geometric properties of their elements (see supplementary).

**POSTER** is a new dataset we collected from Adobe Spark website. It contains 35K poster templates created by design professionals. There are 4 element classes including “background”, “text”, “image” and “vector”. We split the POSTER into 28K training set, 7K gallery set, and 50 query set. We plan to release this data conditioned on internal approval. POSTER dataset originally contains the hierarchy annotations, thus, we directly employ our model on it.

Please note, since we target at hierarchical layout, we demonstrate our model advantages on RICO and POSTER, while not using other datasets with relatively flat or sequential structure, such as floorplans (Wu et al. 2019), ICDAR2015 (Antonacopoulos et al. 2015), and PubLayNet (Zhong, Tang, and Yepes 2019).

### Evaluation Protocol

Existing works mainly rely on the IoU values to measure layout similarity. Besides, human subjective evaluation is also a necessary measurement for layout searching. In our work, we newly propose to use tree-edit distance (TED) to evaluate layout similarity from structural perspective. Our evaluation protocol allows a comprehensive evaluation for layouts from both spatial (IoU) and structural (TED) aspects accompanied with human subjective evaluation.

### Mean Intersection Over Union (MIoU)

Existing works (Manandhar, Ruta, and Collomosse 2020; Deka et al. 2017) mainly focus on spatial similarity computed as overlapping element area, which is measured by MIoU:

$$\text{MIoU}(Q, G) = \frac{1}{Q} \sum_{Q_i \in Q} \sum_{j=1}^{M} \frac{A_j(Q_i) \cap A_j(G_i)}{A_j(Q_i) \cup A_j(G_i)}$$

where $A_j(\cdot)$ is the area class $j$ elements in this sample. $Q$, $G$ are query and gallery sets. We use top@k, $k = \{1, 5, 10\}$ retrievals for MIoU calculation.

### Tree-Edit Distance (TED)

We propose to measure layout structural similarity by involving TED metric. TED is originally defined to measure the distance between two trees. It calculates minimum cost to transform one tree to another by three basic operations: 1) insert a node, $I(\cdot)$; 2) delete a node, $D(\cdot)$; 3) revise the label of a node, $R(\cdot)$ (Sidorov et al. 2017).
et al. 2015). Each operation has a cost value given by $\mathcal{F}()$. The edit distance $\delta(\mathcal{T}_1, \mathcal{T}_2)$ is the sum of cost for a editing sequence to transfer $\mathcal{T}_1$ to $\mathcal{T}_2$:

$$\delta(\mathcal{T}_1, \mathcal{T}_2) = \sum_{j=1,...,J} \mathcal{F}(S_j(\mathcal{T}_1)),$$

(10)

where $S_j \in \{I, D, R\}$ and $\mathcal{T}_2 = S_J(S_{J-1}(...S_1(\mathcal{T}_1)))$. $J$ is the length of the operation sequence to transfer $\mathcal{T}_1$ to $\mathcal{T}_2$. In our work, we employ the Zhang-Shasha algorithm (Zhang and Shasha 1989) to implement our TED measurement for 2D layout data. We report TED at top@$k$. In Tab. 2, we observe GCN-CNN serves as strong baseline, and can be further improved when trained with auxiliary triplet supervision. Our SSH-AE-(L3 SP) achieves the best performance in terms of MIoU for spatial similarity, and SSH-AE-(L3 ST) is the best in terms of TED for structural similarity. Note that our models are trained without triplet loss, and the combined setting SSH-AE-(L3 SP+L3 ST) outperforms GCN-CNN without triplet on all the metrics.

The second block shows the ablation study of different SSH-AE variations. It clearly shows our models trained with spatial (SP) and structural supervision (ST) are good for MIoU and TED metric, respectively. We find adding more hierarchical levels leads to improvement for both supervisions on all the metrics. These results demonstrate the effectiveness of our multi-level modeling for layout hierarchy and necessity of considering layout from spatial and structural perspectives. Compared with the second block, we find our SSH-AE-(L3 SP) (0.694) is very close to the MIoU upper bound (0.715), but there is still a relatively large gap between the perfect TED OPT (6.520) and SSH-AE-(L3 ST) (10.520). It indicates there is room for further improvement. Also from the second block, we note MIoU and TED are indeed two facets of the layout matching. The optimal results for MIoU are not competitive for TED, and vice versa. This justifies the need to examine the two metrics jointly.

The NDCG is based on a user study in the similar way as (Manandhar, Ruta, and Collomosse 2020). We can see the best NDCG performance is achieved by methods keeping a good trade-off between spatial and structural losses (L3 SP+L3 ST). Such observation offers strong support to our method with joint spatial and structural layout modeling.

### Experimental Analysis

#### RICO Retrieval

Tab. 1 shows the RICO retrieval results. The first block contains baseline methods. Since the retrieval results of LayoutGMN paper (Patil et al. 2021) is based on a non-standard query set, we cannot report its NDCG results with our human evaluation data. The second block contains the optimal retrieval performance with respect to MIoU (MIoU OPT) and TED (TED OPT) on the given test set. We also combine the ranking scores of MIoU and TED as a trade-off between the two metrics (MIoU+TED OPT). They serves as the upper bound for MIoU and TED. The third block contains our SSH-AE with different settings. The last block contains the combined ranking scores of two level 3 settings (L3 SP+L3 ST) as an ensemble result to provide a trade-off between the spatial and structural aspects.

In Tab. 1, we observe GCN-CNN serves as strong baseline, and can be further improved when trained with auxiliary triplet supervision. Our SSH-AE-(L3 SP) achieves the best performance in terms of MIoU for spatial similarity, and SSH-AE-(L3 ST) is the best in terms of TED for structural similarity. Note that our models are trained without triplet loss, and the combined setting SSH-AE-(L3 SP+L3 ST) outperforms GCN-CNN without triplet on all the metrics.

The third block shows the ablation study of different SSH-AE variations. It clearly shows our models trained with spatial (SP) and structural supervision (ST) are good for MIoU and TED metric, respectively. We find adding more hierarchical levels leads to improvement for both supervisions on all the metrics. These results demonstrate the effectiveness of our multi-level modeling for layout hierarchy and necessity of considering layout from spatial and structural perspectives. Compared with the second block, we find our SSH-AE-(L3 SP) (0.694) is very close to the MIoU upper bound (0.715), but there is still a relatively large gap between the perfect TED OPT (6.520) and SSH-AE-(L3 ST) (10.520). It indicates there is room for further improvement.

### Baseline Methods

A MLP-based auto-encoder (AE) is proposed in (Deka et al. 2017) which reconstructs rasterized images. A convolutional auto-encoder (CAE) (Liu et al. 2018) is designed to improve the layout feature capacity. Representative hierarchy-based generation model StructureNet (SN) (Mo et al. 2019) uses the hierarchical characteristics to generate 3D objects. We implement SN on layout as a baseline for hierarchical modeling. LayoutGMN (Patil et al. 2021) employs a graph matching approach for the layout retrieval. The state-of-the-art layout retrieval model GCN-CNN (Manandhar, Ruta, and Collomosse 2020) is also included for comparisons.

### Table 2: Retrieval performance on the POSTER dataset based on MIoU, TED, and NDCG evaluation metrics. Best results are boldfaced and best ideal values are underlined. The ideal NDCG values are 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>MIoU OPT</th>
<th>MIoU OPT</th>
<th>MIoU OPT</th>
<th>TED OPT</th>
<th>TED OPT</th>
<th>NDCG OP@1</th>
<th>NDCG OP@5</th>
<th>NDCG OP@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH-AE-(L3 SP)</td>
<td>0.624</td>
<td>0.579</td>
<td>0.565</td>
<td>18.900</td>
<td>20.592</td>
<td>20.486</td>
<td>0.702</td>
<td>0.774</td>
</tr>
<tr>
<td>SSH-AE-(L2 SP)</td>
<td>0.647</td>
<td>0.604</td>
<td>0.587</td>
<td>14.980</td>
<td>20.776</td>
<td>20.570</td>
<td>0.748</td>
<td>0.798</td>
</tr>
<tr>
<td>SSH-AE-(L3 ST)</td>
<td>0.654</td>
<td>0.612</td>
<td>0.596</td>
<td>16.060</td>
<td>21.084</td>
<td>20.608</td>
<td>0.733</td>
<td>0.794</td>
</tr>
<tr>
<td>SSH-AE-(L1 ST)</td>
<td>0.378</td>
<td>0.347</td>
<td>0.344</td>
<td>8.140</td>
<td>10.048</td>
<td>11.940</td>
<td>0.622</td>
<td>0.719</td>
</tr>
<tr>
<td>SSH-AE-(L2 ST)</td>
<td>0.396</td>
<td>0.354</td>
<td>0.348</td>
<td>7.020</td>
<td>9.276</td>
<td>11.090</td>
<td>0.629</td>
<td>0.715</td>
</tr>
<tr>
<td>SSH-AE-(L3 ST)</td>
<td>0.405</td>
<td>0.359</td>
<td>0.353</td>
<td>6.420</td>
<td>7.952</td>
<td>8.922</td>
<td>0.653</td>
<td>0.737</td>
</tr>
</tbody>
</table>

SSH-AE-(L3 SP+L3 ST)   | 0.653    | 0.611    | 0.593    | 16.740  | 20.924  | 20.512    | 0.744     | 0.801       |

Table 2 shows the retrieval results on the POSTER dataset based on MIoU, TED, and NDCG evaluation metrics. Best results are boldfaced and best ideal values are underlined. The ideal NDCG values are 1.
Figure 4: Visualization of different methods’ operating curves and points for spatial/structural trade-off on RICO (top) and POSTER (bottom). The operating curve for each method exhibits inverse relationship between MIoU@10 and TED@10. SSH-AE has better results than others with the reference of the optimal metric (OPT) curve. The big dot on each curve indicates the point with best NDCG value.

Trade-off Analysis. The ensemble of SSH-AE-(L3 SP+L3 ST) is obtained by a particular weighted ranking combination of SSH-AE-(L3 SP) and SSH-AE-(L3 ST). It serves as a trade-off to balance the spatial and structural aspects. More generally, we can tune the combination weight within [0, 1] to obtain an operating curve that connects the two operating points representing two base methods. They are plotted in the 2D space of MIoU and TED metrics (see Fig. 4). We find the curves of SSH-AE are closer to the optimal curves (OPT) than all the baselines. SSH-AE performs better when it is configured with more levels.

The operating curves are similar as ROC curves for detection. It provides a comprehensive evaluation and enable us to compare two methods with different spatial/structural emphasis. In Fig. 4, the point on each curve with the best NDCG is highlighted as a big dot. For RICO, it is interesting to note the best NDCG operating points almost have the best trade-off between MIoU and TED, indicating a high consistency between the subjective evaluation and our new objective evaluation. However, the NDCG points on different curves are not always consistent with their locations, as the best point on OPT curve has lower NDCG result than the best point on SSH-AE curve. For POSTER, the subjective evaluation prefers the pure SP setting, and the synergy between SP and ST settings is not strong. This can be explained by the fact the curves for SSH-AE looks like “concave” in top right region, and therefore the linear combination of two operating points may become worse than each of them. We believe this new evaluation protocol opens a direction to explore better objective evaluation for layout matching. Overall, SSH-AE outperforms baseline methods under a wide range of spatial-structural trade-off.

Qualitative Retrieval Results. We provide RICO representative retrieval visualization in Fig. 5. We show the query (RGB and semantic map) and top1 retrieval of GCN-CNN (triplet), our method (L3 SP+L3 ST ensemble), and MIoU/TED OPT. Compared with most competitive GCN-CNN, ours captures more fine-grained details and retrieves better results based on both the query sample and the optimal retrieval of MIoU and TED. More qualitative results with discussions are in supplementary material.

Conclusion

We learn layout representation from a novel way – considering both spatial and structural perspectives in a multi-level hierarchical fashion. Our Spatial-Structural Auto-Encoder (SSH-AE) is built to handle layout data with hierarchical annotations based on its elements. A hierarchical autoencoder is used to extract and fuse layout features with different spatial and structural significance. Orthogonally, a two-pathway optimization and inference design is used to enforce layout information from spatial and structural aspects. Accordingly, we also introduce a new evaluation protocol with a newly involved tree-edit distance (TED) metric for a comprehensive layout similarity measurement which better aligns with human judgement. Experiments on both RICO and POSTER datasets demonstrate the superiority of SSH-AE in layout retrieval. The new collected POSTER dataset and our SSH-AE with new evaluation protocol are expected to benefit future researches in layout area.
References


