Layout Representation Learning with Spatial and Structural Hierarchies

Yue Bai[1*], **Dipu Manandhar**[3], **Zhaowen Wang**[4], **John Collomosse**[4], **Yun Fu**[1,2]

[1]Department of Electrical and Computer Engineering, Northeastern University
[2]Khoury College of Computer Science, Northeastern University
[3]Centre for Vision, Speech and Signal Processing, University of Surrey
[4]Adobe 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 will be released.

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 another 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).

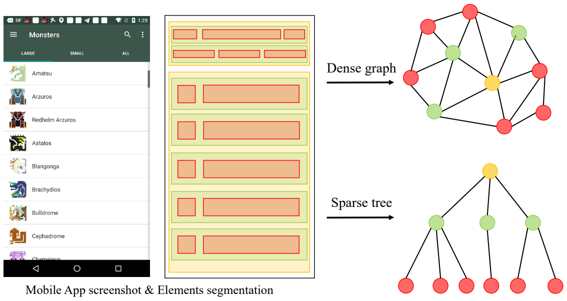

Figure 1: Layouts contain several elements with nested containment and alignments as highlighted by bounding boxes. They are modelled as graphs: elements as nodes and their relationships as edges. Previous works treat all nodes equivalently as a dense graph ignoring the structural characteristic. We propose to utilize rich hierarchy information and construct a sparse tree leading to a discriminative representation. Different colors mean different element levels.

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 hierar-

[*] Corresponding author: bai.yue@northeastern.edu; Work done during the author's internship at Adobe Research.

chy can be constructed based grouping geometric relationships of design elements. These structural relationships are intrinsically encoded in the digital layout documents. Fig. 1 illustrates a layout example with multiple elements which are structurally aligned by geometric relationships.

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) process 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).

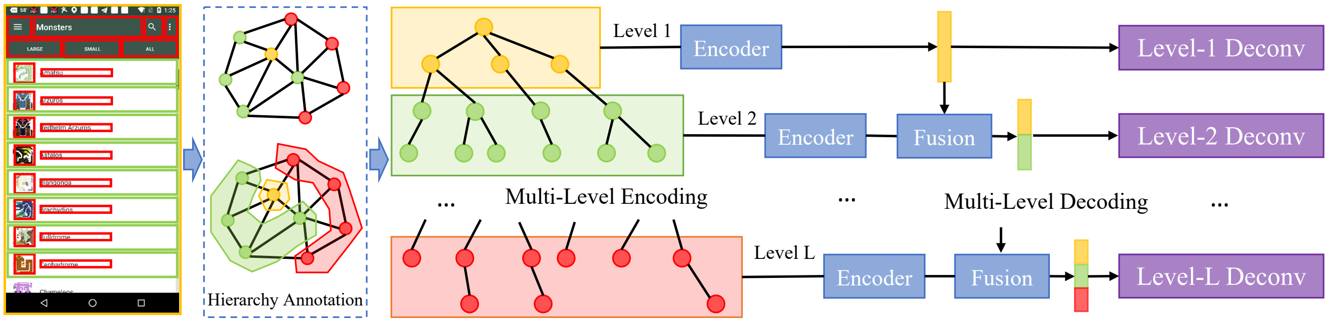

Figure 2: The illustration of Spatial-Structural Hierarchical Auto-Encoder (SSH-AE). Given an input layout containing a set of elements with hierarchical annotations, we separate elements and construct a tree hierarchy. Then we obtain level-wise layout features based on multi-level encoding and progressive 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 $T{=}T^1 \cup T^2 \cup, ..., \cup T^L$ with nodes $V{=}V^1 \cup V^2 \cup, ..., \cup V^L$ and edges $C{=}C^1 \cup C^2 \cup, ..., \cup C^L$, where C^l means all the edges connected from a node in V^l.

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

$$f^l = E(V^l, C^l), \tag{1}$$

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:

$$\mathit{f}^1 = f^1, \mathit{f}^l = U^{l-1}(\mathit{f}^{l-1}) \oplus f^l, l \geq 2, \tag{2}$$

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 $F{=}\{\mathit{f}^1, \mathit{f}^2, ..., \mathit{f}^L\}$ is passed to the decoder during training, and used for downstream retrieval.

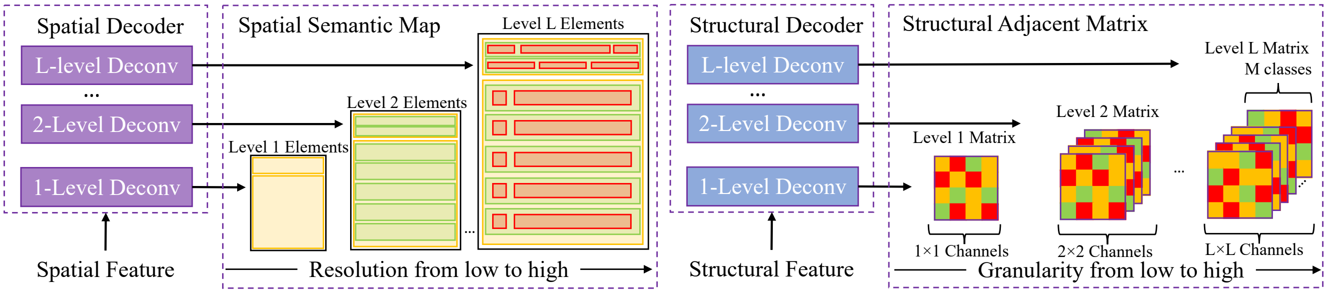

Figure 3: The spatial perspective (left): decoders of different levels reconstruct spatial semantic maps in different resolutions. The structural perspective (right): decoders of different levels reconstruct structural adjacent matrices in different granularities.

Level-wise Decoding Each level feature f^l is fed into level-specific decoder D^l to generate an output I^l:

$$I^l = D^l(f^l), \tag{3}$$

where D^l is implemented as deconvolution blocks containing several layers of strided convolution that upsample the output to match the reconstruction supervision tensor.

Spatial and Structural Encoding

We design two variations of layout encoder from spatial and structural perspectives. Either of these can be deployed in the hierarchical auto-encoder in a multi-level fashion. For simplicity, we omit the superscript l for each level and introduce the realization for spatial and structural aspects.

Spatial Encoding. A layout hierarchy is represented as $T{=}\{V, C\}$ with nodes $V{=}\{v_1, ..., v_i, ..., v_n\}$ denoting elements and edges $C{=}\{c_{1,2}, ..., c_{i,j}, ..., c_{n-1,n}\}$ denoting tree-like architecture. Each node v_i has a semantic label s_i and a geometric feature g_i^v (Manandhar, Ruta, and Collomosse 2020). We encode the s_i as one-hot vector representing the element class. The geometric feature g_i^v is given by

$$g_i^v = \left[\frac{x_i}{w}, \frac{y_i}{h}, \frac{w_i}{w}, \frac{h_i}{h}, \frac{A_i}{wh}, d_i\right], \tag{4}$$

where x_i, y_i, w_i, h_i, d_i, and A_i are the centroid coordinates, width, height, node depth, and the square root of the area for v_i. w and h are the width and height of the entire layout. We first concatenate the one-hot semantic label s_i and node geometric feature g_i^v for each node $i{\in}V$. Then, we project it as a complete node feature f_i^v for spatial content learning:

$$f_i^v = E^v([s_i, g_i^v]), \tag{5}$$

where E^v is implemented as a MLP. Then all the node features are combined with a semantic-keyed attention module:

$$f^v = \sum_i \alpha_i^v f_i^v, \tag{6}$$

where $\alpha_i^v \propto \exp(w_v^\top s_i)$ are attention weights with learnable parameter w_v.

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

$$g_{i,j}^c = \left[\phi_{i,j}, \theta_{i,j}, \frac{\Delta x}{A_i}, \frac{\Delta y}{A_i}, \frac{w_j}{w_i}, \frac{h_j}{h_i}, \frac{1}{D}\sqrt{\Delta x^2 + \Delta y^2}\right], \tag{7}$$

where $\Delta x = x_j - x_i$, $\Delta y = y_j - y_i$, and $D = \sqrt{w^2 + h^2}$. The orientation $\theta = atan2(\frac{\Delta y}{\Delta x}) \in [-\pi, \pi]$. $\phi_{i,j}$ serves as the Intersection over Union (IoU) between node v_i and v_j given by $\phi = \frac{M(v_i) \cap M(v_j)}{M(v_i) \cup M(v_j)}$, 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 $g_{i,j}^c$ for edge $(i,j){\in}C$ together with the paired node features with an MLP E^c. Finally, all the edge features are aggreagated with a node-keyed attention module which are given by

$$f_{i,j}^c = E^c([f_i^v, g_{i,j}^c, f_j^c]), f^c = \sum_{ij} \alpha_{ij}^c f_{i,j}^c, \tag{8}$$

where $\alpha_{i,j}^c \propto \exp(w_c^\top [f_i^v, f_j^v])$ 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 $\|O^l - I^l\|$ as the optimization objective to train the encoder and 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_i, y_i, w_i, h_i) 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.

Table 1: Retrieval performance on RICO dataset based on MIoU, TED, and NDCG evaluation metrics. Best results are **bold-faced** and best ideal values are underlined. The ideal NDCG values are 1.

Method	MIoU			TED			NDCG		
	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10
AE (Deka et al. 2017)	0.430	0.362	0.312	15.960	19.132	19.930	0.325	0.318	0.324
CAE (Liu et al. 2018)	0.595	0.471	0.440	14.100	16.124	17.976	0.482	0.434	0.416
SN (Mo et al. 2019)	0.407	0.379	0.360	17.540	19.268	18.994	0.428	0.448	0.466
GCN-CNN (Manandhar, Ruta, and Collomosse 2020)	0.600	0.514	0.482	15.360	17.644	19.398	0.576	0.564	0.574
GCN-CNN+Triplet (Manandhar, Ruta, and Collomosse 2020)	0.617	0.541	0.513	13.820	16.748	17.696	0.601	**0.576**	**0.588**
LayoutGMN (Patil et al. 2021)	0.446	0.384	0.342	15.174	17.139	18.241	-	-	-
MIoU OPT	<u>0.715</u>	<u>0.634</u>	<u>0.607</u>	14.720	19.840	21.260	0.579	0.525	0.547
TED OPT	0.468	0.372	0.343	<u>6.520</u>	<u>8.560</u>	<u>9.552</u>	0.572	0.530	0.542
MIoU+TED OPT	0.546	0.452	0.430	6.580	8.704	9.710	0.607	0.550	0.561
SSH-AE-(L1 SP)	0.624	0.549	0.536	19.480	21.704	23.260	0.552	0.525	0.547
SSH-AE-(L2 SP)	0.678	0.601	0.574	17.640	20.184	21.512	0.555	0.546	0.553
SSH-AE-(L3 SP)	**0.694**	**0.610**	**0.580**	15.540	19.220	21.880	0.585	0.536	0.546
SSH-AE-(L1 ST)	0.498	0.399	0.371	13.000	15.700	16.612	0.553	0.502	0.526
SSH-AE-(L2 ST)	0.476	0.386	0.367	12.300	14.512	15.618	0.539	0.487	0.510
SSH-AE-(L3 ST)	0.493	0.390	0.362	**10.520**	**14.144**	**15.264**	0.597	0.510	0.520
SSH-AE-(L3 SP+ L3 ST)	0.684	0.582	0.554	13.380	15.352	15.830	**0.656**	0.565	0.562

Structural Pathway: Adjacent Matrix. Spatial supervision is naturally constructed by the typical decoding reconstruction, but this is not straightforward for structural pathway. Inspired by the concept of adjacency matrix in graph learning, we integrate the structural information into a multi-channel adjacent matrix which follows the same format of spatial aspect. Given the hierarchy at level l, we construct an adjacent matrix $O^l \in R^{M \times M \times (l \times l)}$. It contains $l \times l$ channels to represent all the combinations between different levels. Each channel has a $M \times M$ matrix, where the value of each element denotes how many edges are connected between the corresponding two classes. For example, each element of the tensor $O[c_i, c_j, l_m, l_n]$ denotes the number of edges between component class i to class j from level m to level n. Ignoring class dimension for simplicity, $O[:, :, 2, 2]$ represents edges between nodes at level 2, $O[:, :, 2, 3]$ denotes edges from level 2 to level 3 forming parent-child relations. In this way, we obtain a tensor $O^l \in R^{M \times M \times (l \times l)}$ with $l \times l$ channels, where each channel is a $M \times M$ adjacency matrix representing the class-wise connection relationship and overall O^l provides supervision of the structural patterns from both class-wise and level-wise perspectives.

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 https://www.adobe.com/express/trial/create-together*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:

$$\mathrm{MIoU}(\mathcal{Q}, \mathcal{G}) = \frac{1}{\mathcal{Q}} \sum_{\mathcal{Q}_\mathrm{i} \in \mathcal{Q}} \sum_{\mathrm{j}=1}^{\mathrm{M}} \frac{\mathrm{A_j}(\mathcal{Q}_\mathrm{i}) \cap \mathrm{A_j}(\mathcal{G}_\mathrm{i})}{\mathrm{A_j}(\mathcal{Q}_\mathrm{i}) \cup \mathrm{A_j}(\mathcal{G}_\mathrm{i})}, \tag{9}$$

where $A_j(\cdot)$ is the area class j elements in this sample. $\mathcal{Q}$, $\mathcal{G}$ 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, $\mathcal{I}(\cdot)$; 2) delete a node, $\mathcal{D}(\cdot)$; 3) revise the label of a node, $\mathcal{R}(\cdot)$ (Sidorov et al. 2015). Each operation has a cost value given by $\mathcal{F}(\cdot)$. The edit distance $\delta(\mathcal{T}_1, \mathcal{T}_2)$ is the sum of cost for a editing

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.

Method	MIoU			TED			NDCG		
	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10
AE (Deka et al. 2017)	0.484	0.408	0.395	16.300	19.764	20.624	0.728	0.798	0.838
CAE (Liu et al. 2018)	0.460	0.404	0.376	14.980	18.160	19.508	**0.757**	0.800	0.838
GCN-CNN (Manandhar, Ruta, and Collomosse 2020)	0.586	0.547	0.538	19.420	20.604	20.800	0.677	0.766	0.816
SN (Mo et al. 2019)	0.322	0.325	0.319	18.149	16.494	15.940	0.652	0.757	0.811
MIoU OPT	0.666	0.626	0.609	17.160	19.812	20.738	0.726	0.796	0.843
TED OPT	0.359	0.331	0.321	2.740	3.860	4.386	0.637	0.744	0.795
MIoU+TED OPT	0.648	0.607	0.591	7.700	9.608	10.308	0.769	0.827	0.855
SSH-AE-(L1 SP)	0.624	0.579	0.565	18.900	20.592	20.486	0.702	0.774	0.831
SSH-AE-(L2 SP)	0.647	0.604	0.587	14.980	20.776	20.570	**0.748**	0.798	**0.845**
SSH-AE-(L3 SP)	**0.654**	**0.612**	**0.596**	16.060	21.084	20.608	0.733	0.794	0.841
SSH-AE-(L1 ST)	0.378	0.347	0.344	8.140	10.048	11.940	0.622	0.719	0.782
SSH-AE-(L2 ST)	0.396	0.354	0.348	7.020	9.276	11.090	0.629	0.715	0.782
SSH-AE-(L3 ST)	0.405	0.359	0.353	**6.420**	**7.952**	**8.922**	0.653	0.737	0.787
SSH-AE-(L3 SP+L3 ST)	0.653	0.611	0.593	16.740	20.924	20.512	0.744	**0.801**	**0.845**

sequence to transfer $\mathcal{T}_1$ to $\mathcal{T}_2$:

$$\delta(\mathcal{T}_1, \mathcal{T}_2) = \sum_{j=1,\ldots,J} \mathcal{F}(\mathcal{S}_j(\mathcal{T}_1)), \qquad (10)$$

where $\mathcal{S}_j \in \{\mathcal{I}, \mathcal{D}, \mathcal{R}\}$ and $\mathcal{T}_2 = \mathcal{S}_J(\mathcal{S}_{J-1}(\ldots\mathcal{S}_1(\mathcal{T}_1)))$. 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, $k = \{1, 5, 10\}$.

Normalized Discounted Cumulative Gain (NDCG). We also report NDCG for layout retrieval based on subjective user study (see supplementary for more details).

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.

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. 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.

POSTER Retrieval. Tab. 2 shows the retrieval results on POSTER. We report results for GCN-CNN, optimal metric rankings, six variations of our SSH-AE, and the L3 ensemble. We observe similar trend as what has been seen from the RICO. Note the SSH-AE-(L3 SP) model alone achieves better results than GCN-CNN. We attribute this to our hierarchical modeling strategy. The ensemble trade-off model also achieves the most promising results for subjective NDCG.

Trade-off Analysis. The ensemble of SSH-AE-(L3 SP+L3 ST) is obtained by a particular weighted ranking combina-

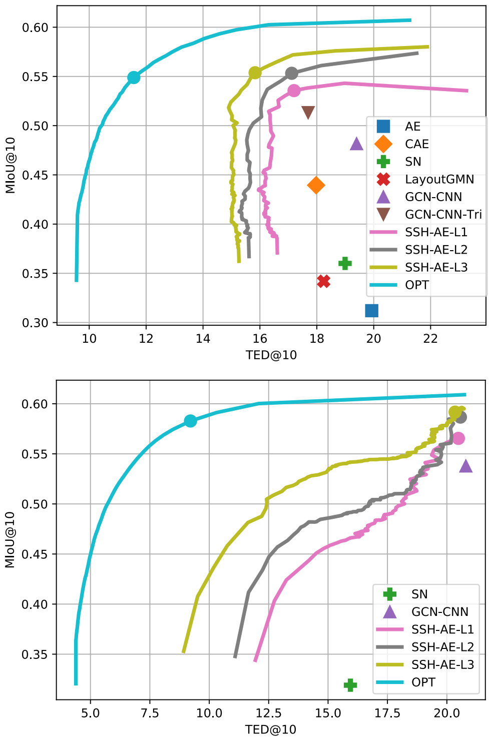

Figure 4: Visualization of different methods' operating curves and points for spatial/structural trade-off on RICO (**top**) and POSTER (**bottom**). The operating curve for each method exhibits inverse relationship between MIoU@10 and TED@10. SSH-AE has better results than others with the reference of the optimal metric (OPT) curve. The big dot on each curve indicates the point with best NDCG value.

tion 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.

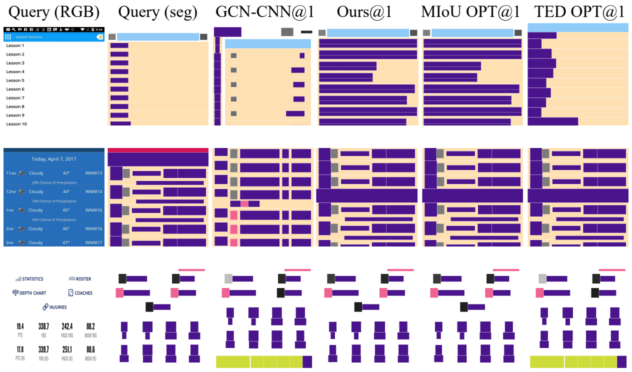

Figure 5: RICO retrieval visualization with query (RGB/semantic map) and top1 retrieval of GCN-CNN (tri)/ours (L3 SP+L3 ST ensemble)/MIoU OPT/TED OPT. For the first example, ours obtains more organized retrieval than GCN-CNN compared with the query. For the second/third examples, both ours and GCN-CNN retrieve good results. However, ours captures better **horizontal element arrangement** (second) and **detailed design element in the middle** (third). Compared with the query and MIoU/TED optimal results, our method generally captures more layout details and obtains better results than the strong baseline GCN-CNN.

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 retrieve 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 auto-encoder 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

Antonacopoulos, A.; Clausner, C.; Papadopoulos, C.; and Pletschacher, S. 2015. ICDAR2015 competition on recognition of documents with complex layouts-RDCL2015. In *2015 13th international conference on document analysis and recognition (ICDAR)*, 1151–1155. IEEE.

Breuel, T. M. 2003. High performance document layout analysis. In *Proceedings of the Symposium on Document Image Understanding Technology*, 209–218.

Bylinskii, Z.; Kim, N. W.; O'Donovan, P.; Alsheikh, S.; Madan, S.; Pfister, H.; Durand, F.; Russell, B.; and Hertzmann, A. 2017. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the 30th Annual ACM symposium on user interface software and technology*, 57–69.

Chen, H.; Perozzi, B.; Hu, Y.; and Skiena, S. 2018. Harp: Hierarchical representation learning for networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chen, J.; Lei, B.; Song, Q.; Ying, H.; Chen, D. Z.; and Wu, J. 2020. A hierarchical graph network for 3D object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 392–401.

Deka, B.; Huang, Z.; Franzen, C.; Hibschman, J.; Afergan, D.; Li, Y.; Nichols, J.; and Kumar, R. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*, UIST '17.

Geigel, J.; and Loui, A. C. 2000. Automatic page layout using genetic algorithms for electronic albuming. In *Internet Imaging II*, volume 4311, 79–90. International Society for Optics and Photonics.

Harrington, S. J.; Naveda, J. F.; Jones, R. P.; Roetling, P.; and Thakkar, N. 2004. Aesthetic measures for automated document layout. In *Proceedings of the 2004 ACM symposium on Document engineering*, 109–111.

Hurst, N.; Li, W.; and Marriott, K. 2009. Review of automatic document formatting. In *Proceedings of the 9th ACM symposium on Document engineering*, 99–108.

Li, J.; Yang, J.; Hertzmann, A.; Zhang, J.; and Xu, T. 2019. Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767*.

Liu, T. F.; Craft, M.; Situ, J.; Yumer, E.; Mech, R.; and Kumar, R. 2018. Learning Design Semantics for Mobile Apps. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, 569–579. New York, NY, USA: ACM. ISBN 978-1-4503-5948-1.

Liu, Y.; and Lapata, M. 2019. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*.

Manandhar, D.; Ruta, D.; and Collomosse, J. 2020. Learning Structural Similarity of User Interface Layouts using Graph Networks. In *Proc. ECCV*.

Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N.; and Guibas, L. 2019. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38(6): Article 242.

O'Gorman, L.; and Kasturi, R. 1995. *Document image analysis*, volume 39. Citeseer.

O'Donovan, P.; Agarwala, A.; and Hertzmann, A. 2014. Learning layouts for single-pagegraphic designs. *IEEE transactions on visualization and computer graphics*, 20(8): 1200–1213.

Pappagari, R.; Zelasko, P.; Villalba, J.; Carmiel, Y.; and Dehak, N. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 838–844. IEEE.

Patil, A. G.; Ben-Eliezer, O.; Perel, O.; and Averbuch-Elor, H. 2020. Read: Recursive autoencoders for document layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 544–545.

Patil, A. G.; Li, M.; Fisher, M.; Savva, M.; and Zhang, H. 2021. LayoutGMN: Neural Graph Matching for Structural Layout Similarity. In *CVPR*, 11048–11057.

Ritchie, D.; Kejriwal, A. A.; and Klemmer, S. R. 2011. d. tour: Style-based exploration of design example galleries. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 165–174.

Sidorov, G.; Gómez-Adorno, H.; Markov, I.; Pinto, D.; and Loya, N. 2015. Computing text similarity using tree edit distance. In *2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC)*, 1–4. IEEE.

Simon, A.; Pret, J.-C.; and Johnson, A. P. 1997. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3): 273–277.

Swearngin, A.; Dontcheva, M.; Li, W.; Brandt, J.; Dixon, M.; and Ko, A. J. 2018. Rewire: Interface design assistance from examples. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12.

Wessel, R.; Blümel, I.; and Klein, R. 2008. The room connectivity graph: Shape retrieval in the architectural domain.

Wu, W.; Fu, X.-M.; Tang, R.; Wang, Y.; Qi, Y.-H.; and Liu, L. 2019. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6): 1–12.

Yang, X.; Yumer, E.; Asente, P.; Kraley, M.; Kifer, D.; and Lee Giles, C. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5315–5324.

Ying, R.; You, J.; Morris, C.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804*.

Zhang, K.; and Shasha, D. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6): 1245–1262.

Zhang, X.; Wei, F.; and Zhou, M. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*.

Zhang, Z.; Cui, P.; and Zhu, W. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Zhong, X.; Tang, J.; and Yepes, A. J. 2019. PubLayNet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1015–1022. IEEE.