



Layout Generation for Various Scenarios in Mobile Shopping Applications

Qianzhi Jing
Zhejiang University
Hangzhou, China
jingqz@zju.edu.cn

Liuqing Chen*
Zhejiang University
Hangzhou, China
chenlq@zju.edu.cn

Tingting Zhou
Alibaba Group
Hangzhou, China
miaojing@taobao.com

Lingyun Sun
Zhejiang University
Hangzhou, China
sunly@zju.edu.cn

Yixin Tsang
Zhejiang University
Hangzhou, China
22221357@zju.edu.cn

Yankun Zhen
Alibaba Group
Hangzhou, China
zhenyankun.zyk@alibaba-inc.com

Yichun Du
Alibaba Group
Hangzhou, China
yichun.dyc@alibaba-inc.com

ABSTRACT

Layout is essential for the product listing pages (PLPs) in mobile shopping applications. To clearly convey the information that consumers require and to achieve specific functions, PLPs layouts often have many variations driven by scenarios. In this work, we study the PLPs layout design for different scenarios and propose a design space to guide the large-scale creation of PLPs. We propose LayoutVQ-VAE, a novel model specialized in generating layouts with internal and external constraints. LayoutVQ-VAE differs from previous methods as it learns a discrete latent representation of layout and can model the relationship between layout representation and scenarios without applying heuristics. Experiments on publicly available benchmarks for different layout types validate that our method performs comparably or favorably against the state-of-the-art methods. Case studies show that the proposed approach including the design space and model is effective in producing large-scale high-quality PLPs layouts for mobile shopping platforms.

CCS CONCEPTS

- Human-centered computing → User interface design.

KEYWORDS

layout generation, deep generative model, mobile application user interface, product listing pages

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '23, April 23–28, 2023, Hamburg, Germany

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-9421-5/23/04...\$15.00
<https://doi.org/10.1145/3544548.3581446>

ACM Reference Format:

Qianzhi Jing, Tingting Zhou, Yixin Tsang, Liuqing Chen, Lingyun Sun, Yankun Zhen, and Yichun Du. 2023. Layout Generation for Various Scenarios in Mobile Shopping Applications. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3544548.3581446>

1 INTRODUCTION

The product listing pages (PLPs) on mobile shopping platforms, where a number of products are displayed and recommended to consumers for marketing purposes [28], are usually composed of well-arranged product images and attribute information (product title, price, discount, sales, etc.)[13]. In order to clearly demonstrate the characteristics of products and quickly engage consumers, PLPs layouts are created based on both the external constraints - shopping scenarios and the internal constraints - UI elements to be presented. Moreover, even if the elements are the same, the layouts may still vary because different information needs to be emphasized in different scenarios [36]. Figure 1 shows examples of real-world PLPs in different shopping scenarios and corresponding layouts. It can be seen that the PLPs in the recommendation (first row) mainly show images of a variety of products, while the PLPs in the search result (second row) mainly display the attribute information of a smaller number of products belonging to the same category. In addition, each PLP layout can be divided into two parts: mall layout and product card layout, where mall layout segments the entire PLP into several regions, and each region is filled with a product card layout for displaying a product.

Massive PLPs layouts need to be created when developing and running a large mobile shopping platform. However, designing a high-quality layout is an empirical and time-consuming process for designers [4, 32]. It is prohibitively expensive and slow for e-commerce companies to complete this extensive work manually. Traditional solutions are mainly based on generic templates created by designers and produce a layout by selecting a template that best fits the given UI elements [41]. In practical work in industry,

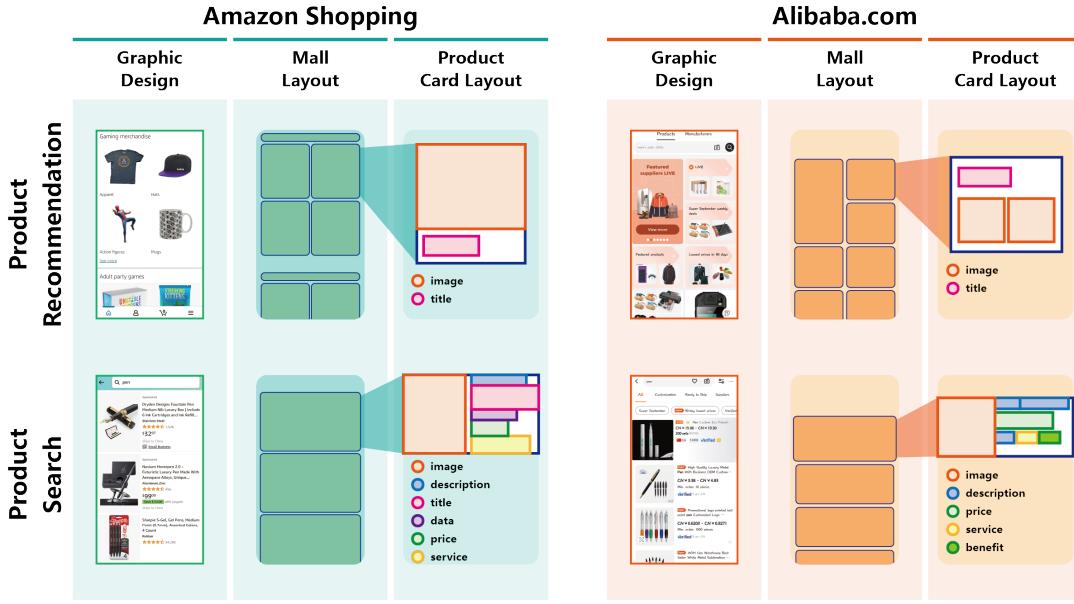


Figure 1: Industrial multi-product display interfaces from Amazon Shopping (first column) and Alibaba.com (second column). The two examples in the first row are obtained from the homepage recommendation section of the two mobile shopping apps, and the ones in the second row are the results of searching for the product "pen".

designers first complete the mall layout design and UI element selection. Then product card layouts are created by the template-based generative method. Although template-based method can reduce the workload of designers, such a predefined and limited set of templates cannot adequately reflect the rich layout variations, nor can perform well in scenario-based and customized layout design which requires multiple distinct layouts for different scenarios.

Recently, since deep generative models have shown great promise in learning a real data distribution and rapidly generating large-scale distinct samples, the field of data-driven layout generation has received extensive attention, and various layout generation methods have been proposed [2, 10, 19, 25]. Most methods represent the layout with a list of bounding boxes of UI elements and define the layout generation task as predicting the attributes of bounding boxes. Some of them [2, 10, 25] only focus on unconditional layout generation where the attributes of UI elements in the generated layout are random and uncontrollable, others only implement conditional layout generation based on internal constraints such as element labels [19, 23] or content [41]. To the best of our knowledge, no prior published study considers the impact of external scenarios on layout design. Thus, no prior method can be applied to the layout design of PLPs which is heavily influenced by shopping scenarios.

In this work, we take a step toward investigating how shopping scenarios affect PLPs layouts and assisting designers in efficiently creating large-scale PLPs layouts with constraints of external scenarios and internal element labels. We firstly report the interviews about PLPs layout design with four designers, and summarize three popular product display scenarios in mobile shopping apps and the workflow of creating PLPs layouts. Then, in order to study explicit features of layouts, we introduce a dataset PDCard, which contains

product card layouts annotated with shopping scenarios from a highly popular mobile shopping app, and analyze the explicit feature distribution of the layouts for different scenarios. Next, based on the above interviews and data analysis, we propose a design space of PLPs layout design for different scenarios. In this design space, we discuss the three popular scenarios from three perspectives including consumer, product, and layout pattern, and conclude the principles of mall layout design and UI elements selection. Finally and most importantly, to address the challenge on complex and time-consuming arrangement of elements in the product card layouts, we introduce a novel generative model, denoted LayoutVQ-VAE, that is capable of synthesizing multiple layouts based on the internal element labels and external scenarios within seconds. In the model, we innovatively propose the discrete latent representation of layouts by training a novel VQ-VAE [37] and model the relationship between the layout representation and constraints through a unidirectional Transformer [38]. The discrete latent vector can avoid “posterior collapse” which is often encountered in the training of VAE models [2].

We summarize our contributions in the following:

- (1) **Design Space:** We propose a design space to guide the creation and evaluation of PLPs layouts for different scenarios in mobile shopping applications. In particular, to quantitatively analyze the explicit features of the layouts and provide a basis for the proposal of the design space, we construct a dataset, called PDCard, by collecting 2,575 annotated product card layouts from a mobile shopping App. The dataset can also be utilized in other related studies.
- (2) **LayoutVQ-VAE:** We propose a novel generative model specialized in generating layouts satisfying both internal and

external constraints (element labels and scenarios). Discrete latent representation of layouts allows us to improve the quality of generated layouts and better model the relationship between layout and discrete constraints.

- (3) **Evaluation:** The qualitative and quantitative experiments on three public layout datasets and our proposed dataset validate that our model outperforms the state-of-the-art models in terms of realism, aesthetics, and scenario relevance. The case studies on the PDCard dataset demonstrate that our proposed model can generate layouts conforming to the scenario constraints, and our approach can significantly reduce the time to produce large-scale and diverse PLPs layouts, with the promise of greatly improving UI development efficiency in industry.

2 RELATED WORK

2.1 Graphic Layout Generation

Automatic layout generation is a classic research topic. With the great success of deep generative models in the field of computer vision, such as Generative Adversarial Networks (GANs) [8] and Variational Autoencoders (VAEs) [22], researchers applied them to graphic layout generation. LayoutGAN [25] is the first work to utilize a GAN framework for this task, which also proposes a wireframe renderer to evaluate the layout at the pixel level. READ [31] trains a VAE which is based on Recursive Neural Networks (RvNNs) [7] to model the layout distribution. NDN [24] models a design layout as a graph and applies Graph Convolution Networks (GCNs) [34] to capture the dependency among elements. However, NDN and READ both use heuristics to label the relationship between elements, which are limited to datasets with similar labels and unable to model element relationships uniformly, and comprehensively. Gupta et al. [10] and VTN [2] address this problem by exploiting the multi-head attention mechanism in Transformers [38], and employ an autoregressive decoder to model the distribution of layout. A recent work [16] considers segmenting a layout into several regions and decomposing the generation process into two stages, which effectively improves the quality of generated layouts with lots of elements. However, few of the above studies consider the layout generation with conditions (e.g., type and number of elements), which does not satisfy the real-world needs of generating layouts with some known elements and scenarios.

To achieve conditional layout generation, LayoutVAE [17] applies a VAE framework to generate scene layouts according to a given label set. Zheng et al. [41] propose a content-aware layout generation framework that can synthesize layouts based on visual and textual semantics of designer inputs. Guo et al. [9] propose Vinci that can automatically select design materials and templates based on product images and text descriptions input by designers, and finally synthesize an advertising poster. BLT [23] proposes a bidirectional layout transformer that first generates a draft layout based on the designer inputs and then refines the layout iteratively. Kikuchi et al. [19] present LayoutGAN++ and CLG-LO which satisfy the constraints via optimizing the layout latent. The latest study CanvasVAE [39] defines vector graphic documents by a multi-modal set of attributes about canvas and elements and trains a VAE to learn the representation of document layout. Nevertheless, there

are two problems with these studies. Firstly, most of them only consider element and canvas labels and do not meet the requirements of scenarios (e.g., for PLPs, different scenarios require different information types and arrangements). Although Zheng et al. [41] take the magazine category as a conditional input to the model, the output of its generator is a noisy image, which cannot respect the aesthetic rules and requires post-processing work such as element recognition and correction. Secondly, LayoutVAE, LayoutGAN++, CanvasVAE, and [41] use continuous latent to represent layouts, making it difficult to model the relationship between layouts and discrete conditions. To advance these problems, our work introduces scenario constraints and the discrete latent representations of layouts.

2.2 Product Listing Page Design in Online Shopping Applications

UI design in shopping platforms is of enduring interest to researchers as online shopping tends to be a mainstream shopping way. Of special interest is the design of PLPs, which has been shown to have a great influence on the traffic and sales volume on a website in early research [28]. In PLPs, two major information types, visual and textual information, govern the modalities of consumers in acquiring product-related information [13]. Prior studies have debated which of both two information types can enhance the shopping performance of consumers, wherein some studies suggest that visual product information is superior to textual information, especially in terms of search time, recall of brand names and product images, and shopping experience [13, 15]. On the contrary, other scholars have indicated that textual product information is more able to facilitate shopping performance in terms of product attribute recall and perception [20]. Recently, some studies have further explored the effects of dynamic visual forms such as video and virtual reality on consumers' shopping performance [11, 21, 40]. In this paper, we unified these dynamic visual forms as image information for simplicity. In addition, some scholars suggest that the impact of PLPs on consumers' online shopping performance depends on the amount of information conveyed through online product presentations [29]. For instance, Sicilia and Ruiz [35] reported that both the absence and the excess of information result in a lack of attention to the product. Li et al. [27] suggested that the visual-based online product representation has a greater impact on high information load, while the text-based has greater performance advantages under the condition of low information load. Unlike previous studies, this paper discusses the effects of the two major information types and the balance of information load on consumers' online shopping performance in different shopping scenarios, and accordingly proposes suggestions for PLP layout design.

3 INTERVIEWS AND DESIGN SPACE

We aim to propose an approach to solve the mass production problem of PLP layouts in industry, which usually change with different shopping scenarios. To gain insight into the motivations, rules, and workflow of scenario-based layout design, we conducted in-depth interviews with four experts. In addition, we built a *Product Card layout* dataset, named PDCard, to analyze the explicit features of

the layouts. Based on the interviews and the analysis of the dataset, we propose a design space for layout design in different scenarios.

3.1 Expert Interviews

We conducted an interview with four experts individually. Two of them (E1 and E2) are senior UI designers from a large e-commerce company with more than eight years of design experience, and the remaining two (E3 and E4) are UX designers from the same company with more than three years of design experience. The average interview time is about 40 minutes. The questions in the interview mainly involved the following three topics: (1) why the shopping scenario is considered when designing the layouts of shopping apps; (2) the classification of scenarios in shopping apps; (3) the workflow of designing PLPs layouts. The feedback of the experts is summarized in the following:

Importance of the application scenario All designers agreed that rich PLP layout variations in shopping apps are largely driven by the shopping scenarios [14]. One of the reasons is that entering a certain scenario is a reflection of consumer intent which determines what information consumers require and should be displayed. For example, E1 said that “*The consumers without clear goals usually browse the recommendation section on the home page, so we need to show a variety of products and attempt to attract them; for those with clear targets, they often search for the target product directly and want to obtain more sales information from different stores.*” E4 commented that, “*images and videos are more attractive to consumers when they are browsing, but they will pay more attention to quality-related information of the product before making a purchase decision.*”. Another main reason is that shopping scenarios affect the style of layout as different styles can influence consumers’ shopping performance from different aspects, E2 explained that “*irregular layouts can attract the attention of consumers with vague interests, such as asymmetrical layouts, instead regular layout is more conducive for consumers to compare the products.*” Based on the above feedback, we are inspired to describe the features of different scenarios in terms of consumer, product information, and layout pattern, and further extend them as three dimensions of the design space in section 3.3.

Classification of the scenarios Each designer classified the shopping scenarios from different perspectives. E1 classified them according to the categories of displayed products: scenarios for electronics, apparel, foods, and so on. E2 classified them based on the functionality of scenarios, including the recommendation page, the category page, and the search page. E3 divided them into three categories according to the mentality of consumers: scenarios for consumers without a target, with a vague target, and with a clear target. And the classification result of E4 is similar to that of E2. Despite the designers adopt different classifications, there are strong correlations among them. For example, the pages with different functionalities proposed by E2 can meet the needs of various consumers proposed by E3, and the presentation of different products proposed by E1 can also be incorporated into the category page proposed by E2. Finally, to comprehensively cover different shopping scenarios, we adopt the classification based on the scenario

functionality, that is, the pages for product recommendation, product by category, and product search. For the features of consumers and product categories, we will discuss them in section 3.

Layout Design workflow Given a particular scenario, designers usually design the PLP layouts through the following three steps: Firstly, they need to design the mall layout which divides the whole PLP into several product cards. E2 commented that “*The single-column mall layout is more suitable for displaying the results of searching for a certain product, yet the double-column layout is better for the recommendation interface of homepage and category page.*” Secondly, designers select how many and which UI elements to display in each card, E1 explained that “*usually an product card includes a main image of product, a title and several attribute elements, such as price, benefits information, etc.*” Finally, the product card layout is designed based on the given scenario and selected elements, including the size of elements and the appropriate position. For example, E1 shared his design experience that “*For consumers who do not have clear goals and strong interests, we design a layout by placing images and videos with a large area, because visual information can be understood faster than textual information, and it is easier to stimulate consumer interest. On the contrary, for consumers with clear goals, we place discounts and sales information in a prominent position to increase their purchase intention.*” According to the workflow and experience provided by the experts, we can infer the explicit design rules from the first two steps and summarize them in the design space. However, due to complex element relationships and various arrangements, the product card layout design in the third step is difficult to formalize with heuristic rules. To this end, we propose a deep generative model for constrained layout generation.

3.2 Dataset

Generating layouts with a machine learning model requires a large product card layout dataset with ground-truth layout annotations. Although there are some publicly available UI design datasets such as Rico [5] and VINS [3], they only contain geometric information of each element in layout and are not labeled with particular scenarios. The layouts in the Magazine dataset [41] are labeled with six magazine categories, which is similar to our shopping scenarios, but not applicable to online shopping apps. To address this challenge, we created PDCard dataset by collecting 2,575 product card layouts from Taobao, the most popular online shopping platform in China. All layouts were initially collected in the form of design drafts produced by Sketch [1], and finally annotated with the three shopping scenarios and bounding boxes describing the boundaries of UI elements. The scenarios where the design drafts are used in the real world provided a preliminary definition for the annotations of the shopping scenarios (e.g., Popular products recommendation, Beauty products, ...), then two interviewed designers were invited to check the initial annotations and the group them into the three popular scenarios we defined and they receive \$12 per hour as remuneration. The bounding boxes were automatically extracted by a program from the design drafts, which contain detailed geometric parameters of each bounding box. The aspect ratio of layouts in the dataset is not exactly the same, there are seven aspect ratios: 3:7, 1:2, 3:5, 3:4, 5:4, 2:1, and 5:2. The numbers of layouts for the seven aspect ratios are 266, 445, 794, 242, 214, 416, and 198. According

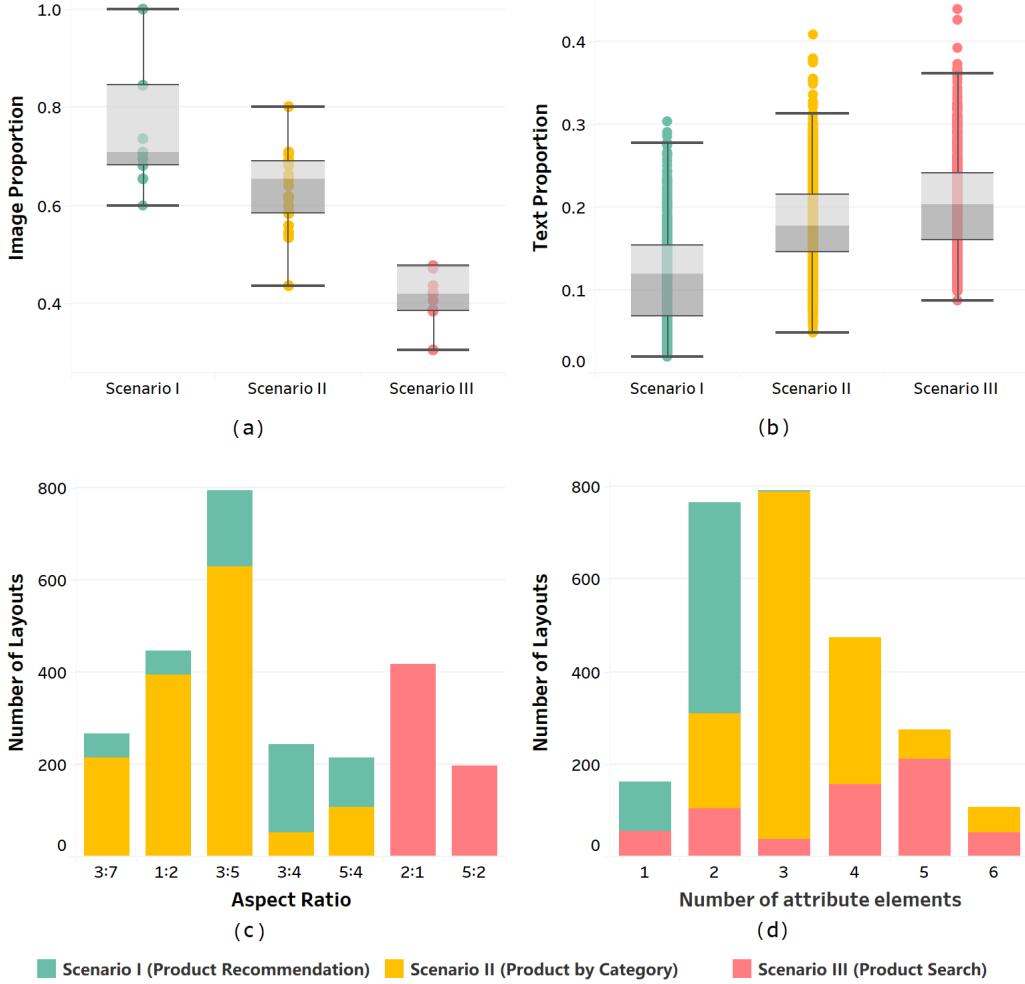


Figure 2: Quantitative analysis of layouts in PDCard dataset. (a) the distribution of the image proportion of layouts for three scenarios; (b) the distribution of text proportion of layouts for three scenarios; (c) the distribution of layouts for three scenarios under different aspect ratios; (d) the distribution of layouts for three scenarios under different numbers of attribute elements.

to the UI design specifications of Taobao, the elements in layouts are divided into eight categories: image, title, price, description, benefit, data, service, and action point, where the last six categories describing product attributes are uniformly represented as attribute elements in the design space. Examples of product card layouts in PDCard are given in the supplementary materials.

To explore the explicit characteristics of layouts suitable for different scenarios, we calculated the proportion of two main information types in PLPs layouts: image proportion P_{img} and text proportion P_{txt} [13]. We also counted the frequency distribution of the layout aspect ratio R and the number of layout attribute elements N_{attr} . Note that the text proportion is the percentage of area occupied by the title and attribute elements. Figure 2 is a visualization of the above calculation results. It can be seen from 2(a) that the layouts for scenario I (product recommendation) have the highest P_{img} , followed by scenario II (product by category) and

the lowest in the scenario III (product search), but the distribution of P_{txt} (seen in 2(b)) is completely opposite, which means scenario I focuses on visual information, while scenario III emphasizes textual information. The statistical results of aspect ratio (seen in 2(c)) demonstrate that the layouts for scenario I and II are mainly vertical layouts ($R > 1$), rather the layouts for scenario III are horizontal layouts ($R < 1$). From 2(d) we can see that most layout for scenario II have more attribute elements than layouts for scenario I, and the number of attribute elements in layouts for scenario III varies widely, with 4 or 5 being the majority.

3.3 Design Space Overview

This section introduces a design space for PLPs layouts applied to different scenarios. Figure 3 shows the overview of the design space. Based on the feedback of the interviews, we classify the scenarios for PLPs in mobile shopping into 3 categories: the scenarios for

Scenarios		Product Recommendation	Product by Category	Product Search
Consumer	Mentality	No target	Convergent target	Clear target
	Behavior	Aimlessly browse	Actively explore	Inspect details
Product Information	Categories	 Wide varieties	 Focused on a certain category e.g., electronics...	 Focused on a specific product e.g., smartphone...
	Information Dimensions	 Visual information dominates	 Visual information & a little textual information	 Visual information & rich textual information
Layout Pattern	Mall Layout	2-3 columns & small product cards 	2 columns & middle product cards 	1 column & large product cards 
	Product Card Layout	Sample  	Sample  	Sample  
		R: 4:3, 5:3, ... Pi ∈ [0.68, 0.84] Pt ∈ [0.06, 0.15]	R: 5:3, 2:1, 7:3, ... Pi ∈ [0.58, 0.69] Pt ∈ [0.14, 0.21]	R: 1:2, 2:5, ... Pi ∈ [0.39, 0.48] Pt ∈ [0.16, 0.24]
	Element	1-3 image 0-1 title 1-2 attribute elements	1 image 1 title 2-4 attribute elements	1 image 1 title 4-6 attribute elements

Note

R: aspect ratio of the canvas;

P_i, P_t: image and text proportion of the layout, the value range is given according to the upper and lower quartiles of the box plot in Figures 2a,b.

Figure 3: Design space for creating PLPs layouts applied to different scenarios in mobile shopping apps.

product recommendation, product by category, and product search. For each scenario, we introduce the corresponding mental and behavioral characteristics of consumers. According to the mentality and behavior of consumers, we analyze what product information needs to be presented by PLPs and how to present it. Finally, we propose some explicit layout design rules and metrics based on the designers' insights in section 3.1 and the analysis results of the PDCard dataset in section 3.2.

Product Recommendation This is a scenario showing products recommended by a recommendation system, usually existing in the homepage of shopping malls on mobile shopping platforms.

Consumers entering this scenario often have no clear shopping targets, and just aimlessly browse product cards. If these cards do not arouse any interest of consumers, they will have a high probability of leaving, so PLPs in this scenario need to present as various products as possible. Furthermore, due to the superiority of visual over textual information in terms of recall and recognition [13, 15], product information is mainly presented in the form of images. Externalized to the layout pattern, the mall layout in this scenario usually adopts many small product cards and arranges them in two or three columns. The design of card layout often chooses a moderate aspect ratio (e.g., 5:3, 4:3, 4:5, ...) and very

few attribute elements, but the image area accounts for a large proportion, usually between 0.68 and 0.84.

Product by Category This is a scenario showing products by category, usually located in the page “All categories”. Consumers entering this scene are usually more interested in a certain category of products (e.g., Electronics, Beauty, Apparel, …). In terms of behavior, consumers are more active and try to find more attractive products. In this scenario, PLPs should display diverse products within the scope of the major category and outstanding features of products to convert consumers’ interests into purchase desires [20]. The mall layout is dominated by two columns and misaligned medium-sized product cards. The card layout typically applies large or medium aspect ratios (e.g., 7:3, 2:1, 5:3, …) and two to four attribute elements that contain the main features of the product. Its image and text proportion are also moderate compared to other scenarios.

Product Search This is a scenario showing the products consumers are searching for. In this scenario, consumers already have a clear purchase target (e.g., smartphone, lipstick, shirt, …), and try to compare the price, quality, and other details of different products to assist their decision-making. Textual information, as a superior information presentation type [33], has a more significant impact on consumers’ purchase intention and decision-making than visual information [20], and a regular layout is more conducive for consumers to compare and inspect products. Therefore, it is recommended that PLPs present a small number of products on the screen, and provide more textual information about products. To be specific, the mall layout in this scenario is better to leverage single-column and large product cards. The card layout accordingly adopts a horizontal layout where attribute elements dominate.

4 LAYOUT GENERATION MODEL

Given a particular scenario, we can determine the mall layout (single column or two columns) of PLPs and select the elements for each product card based on the design space in section 3.3. For the complex and massive product card layout design, we introduce a generative model, denoted LayoutVQ-VAE, to synthesize layouts constrained by scenarios and element labels (number and category). In this section, we discuss the problem formulation and how we combine VQ-VAE [37] and Transformers [38] to learn the discrete latent representation of layouts. Pairing the representation with corresponding constraints, we can generate high-quality layouts that well match the constraints.

4.1 Problem Formulation

A graphic layout consists of a list of design elements. Our goal is to predict the size and coordinates of these elements based on given constraints, including external constraint - scenario of a layout, and internal constraints - number and categories of elements. Specifically, a graphic layout $x \in X$ can be defined as $x = [s, g_1, g_2, \dots, g_n]$, where s is its scenario, g_i is the i^{th} element and n represents the number of elements. For each element, we represent it by category and bounding box, i.e., $g_i = [c_i, b_i]$, where c_i is the element category, $b_i = [b_i^x, b_i^y, b_i^w, b_i^h]$ represents the center coordinates and size of the bounding box. The definitions of scenario s and element category c_i depend on the dataset (e.g., s

represent the three shopping scenarios and c_i represent the eight element categories in PDCard). In practice, we concatenate the geometric parameters of all bounding boxes into a flattened sequence as $B = [b_1^x, b_1^y, b_1^w, b_1^h, \dots, b_n^x, b_n^y, b_n^w, b_n^h]$, and discretize the float values using 7-bit uniform quantization [10]. Layout constraints(i.e., scenario and element category) can also be discrete vectors, so we project them and discretized geometric parameters into a same learned d -dimensional space, which is equivalent to project one-hot encoded category vectors to the latent space. To correspond in position to the sequence of bounding boxes B , the d -dimensional feature vectors representing constraints are repeated and concatenated into sequences C and S with the same length as B . For brevity, we use $b_i^t, c_i^t, s_i^t, i \in (1, \dots, n), t \in (x, y, w, h)$ to represent each feature vector in B , C , and S .

To model the probability distribution of layouts X , prior works [2, 16, 39] leverage a VAE framework and train it with continuous latent variables. However, there are two issues in learning representations of layouts with continuous features. Firstly, it is difficult and infeasible to reason the relationship between constraints and layouts, since the constraints are usually discrete (i.e., scenarios and element categories). Secondly, a well-known problem “posterior collapse” is typically observed in the VAEs with continuous latent variables, where the latent variables are often neglected when they have a powerful decoder such as LSTMs [12] and Transformers. In this work, we propose a novel method that synthesizes layouts by learning the discrete latent representations of layouts. In this way, the relationship between layout and discrete constraints is interpretable and easily modeled by a powerful autoregressive network, which enables the model to generate higher quality layouts that match the constraints and even allows the model to avoid “posterior collapse” in VAEs [37].

4.2 LayoutVQ-VAE

To learn the discrete latent representations of layouts X , we train a conditional VQ-VAE where the attention layers in Transformers are the backbone of the encoder and decoder. The model takes bounding boxes B , conditional element categories C and scenarios S as input, that are passed through an encoder producing a multi-head representation of layout $Z_e(B, C, S) = [z_e^1(B, C, S), \dots, z_e^{n_h}(B, C, S)]$, where $z_e^j \in R^D$ and n_h is the number of layout heads. For each vector z_e^j , we have a shared embedding space e that maps the vector to a discrete latent vector z_q^j by a nearest neighbor lookup. With the decoder input being $Z_q = [z_q^1, \dots, z_q^{n_h}]$ and the conditional sequences C, S , the bounding boxes is reconstructed as \hat{B} . After training, we use a unidirectional Transformer to model the prior distribution over the discrete latent Z_q , so that we can sample the discrete latent representations of layouts according to input constraints and generate layouts with the decoder. Our overall architecture is shown in Figure 4.

Encoder Our encoder $q_\phi(Z_e|B, C, S)$ firstly uses a multi-layer perceptron to project each input item to a d -dimensional space and adds up with the position embedding [6] to obtain the hidden input of the Transformer block. Next, to map a layout into a fixed-size representation, the encoder prepends several learnable embeddings to the hidden inputs and uses the Transformer encoder to produce

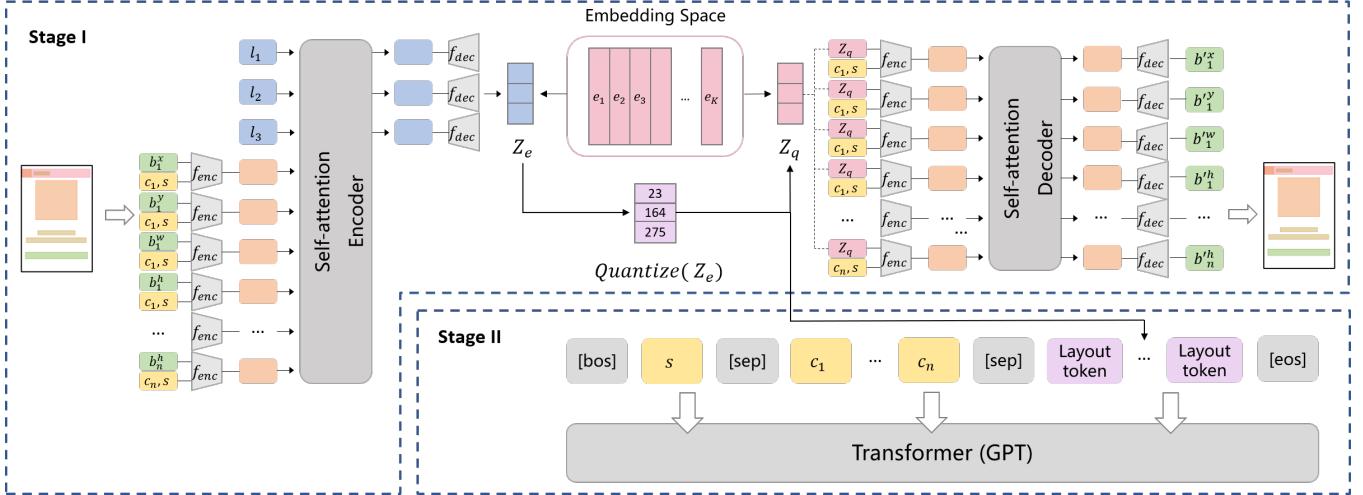


Figure 4: The overview of LayoutVQ-VAE.

vectors. The attention mechanism in Transformers has been proved to be effective at exploiting the complicated relationships between layout elements in prior works [2, 10]. Finally, the encoder output is limited to the final hidden vectors corresponding to the learnable embeddings which are considered to contain the characteristic information of the layout. The strategy we use to obtain the multi-head layout representation is inspired by the sentence classification strategy in BERT [6]. We formulate our encoder in the following:

$$h_i^t = f_{\text{enc}}(b_i^t, c_i^t, s_i^t; \phi) + p_i^t, \quad (1)$$

$$\{l_j'\} = \text{Transformer}(\{l_j\}, \{h_i^t\}; \phi), \quad (2)$$

$$z_e^j = f_{\text{dec}}(l_j'; \phi), \quad (3)$$

where f_{enc} , f_{dec} are multi-layer perceptrons, h_i^t is the hidden representation of each input item, p_i^t is the learnable position embedding, $l_i, j \in (1, \dots, n_h)$ is the j^{th} learnable embedding, l_j' is the final hidden vector of l_j , hyperparameter n_h is the number of the learnable embeddings, ϕ is the parameters of the encoder.

Discrete Latent variables Following VQ-VAE [37], we define a discrete latent embedding space $e = [e_1, e_2, \dots, e_K] \in R^{K \times D}$ to quantize each layout head vector z_e^j , where K is the size of the space, and D is the dimension of e_i . We map z_e^j to the nearest element of embedding e as equation 4, so that we can obtain the discrete latent representation of layout:

$$\begin{aligned} z_q^j(B, C, S) &= \text{Quantize}(z_e^j(B, C, S)) = e_k, \\ k &= \operatorname{argmin}_i \|z_e^j(B, C, S) - e_i\|_2. \end{aligned} \quad (4)$$

Decoder Our decoder $p_\theta(\hat{B}|Z_q, C, S)$ takes the quantized layout representation Z_q and conditions as input, passes the input through a Transformer encoder and finally reconstructs the bounding boxes \hat{B} .

$$h_i^t = f_{\text{enc}}(Z_q, c_i^t, s_i^t; \theta) + p_i^t, \quad (5)$$

$$\{h_i^t\} = \text{Transformer}(\{h_i^t\}; \theta), \quad (6)$$

$$\hat{b}_i^t = f_{\text{dec}}(h_i^t; \theta), \quad (7)$$

where \hat{b}_i^t is a reconstructed geometric parameter of bounding boxes, and θ is the parameters of the decoder. We employ a non-autoregressive decoder instead of an autoregressive one, because we find that the former can better understand the relationship between before and after elements and reconstruct a higher quality layout.

Training The training objective of our model is to minimize:

$$L(B, \hat{B}; \phi, \theta) = L_r(B, \hat{B}) + \beta \|Z_e(B, C, S) - sg(e_k)\|_2^2, \quad (8)$$

where L_r is the reconstruction loss (i.e., the cross entropy loss), the second term is the commitment loss, β is the weight coefficient and we use $\beta = 0.25$ in all experiments, $sg(.)$ represents the stop gradient operator that is defined as the following:

$$sg(x) = \begin{cases} x & \text{forward pass} \\ 0 & \text{backward pass} \end{cases}. \quad (9)$$

Thus, the decoder is optimized by the reconstruction loss only, the encoder is optimized by the reconstruction and commitment loss, the embedding space is optimized via exponentially moving averages (EMA), as details in [18].

Prior The prior distribution over the discrete latents $p(Z_q)$ is a categorical distribution, thus we use a unidirectional Transformer (GPT) to autoregressively predict the discrete latent representations of layouts after training LayoutVQ-VAE. Furthermore, we pair each layout latent representation $[z_q^1, \dots, z_q^{n_q}]$ with its corresponding constraints $[s, c_1, \dots, c_n]$ by projecting each vector to the same dimension and concatenating them into a sequence, so that the self-attention mechanism in the GPT can be used to learn the relationship between the constraints and layouts. Three separator tokens, [bos] (beginning of the sequence), [sep] (separator between the conditional tokens and the discrete latent tokens), and [eos] (end of the sequence) are also added to each sequence. During training the GPT, we only optimize the predictions for the discrete layout

representation and ignore the corresponding outputs of the constraints. After training, we can employ the GPT to autoregressively sample the discrete latent representation of layout that matches the constraints, and then input the representation and constraints into the decoder to generate an appropriate layout.

Implement details We implement our LayoutVQ-VAE with PyTorch [30]. For Transformer blocks in the encoder and decoder, we stack 12 layers with an input/output size of 512 and a feed-forward representation size of 256, use 8 multi-attention heads and employ 3 layout heads to represent the layout. For Transformer blocks in the GPT for prior, we stack 4 layers with an input/output size of 1024 and a feed-forward representation size of 1024, and use 2 multi-attention heads. In the embedding space, we use $K = 512$ and $D = 20$. We train the model using the Adam optimizer with a learning rate of $1e-5$ on a GPU of NVIDIA GeForce RTX 3090 Founders Edition.

5 EXPERIMENTS

In this section, we discuss the quantitative and qualitative performance of the proposed model in extensive experiments. We first describe the three public datasets and the evaluation metrics used in our experiments. Then for each experiment, we introduce its experimental setup and discuss the results. Note that in the generation and construction experiments, we don't consider the effect of scenarios on layouts. The reason is that we hope for a fair comparison with other methods. Besides, the public layout datasets lack labels for layout scenarios. While in the generation experiments of product card layout and magazine layout, we take the scenario as an external constraint.

5.1 Dataset

We evaluate our model on PDCard dataset and the following three publicly available datasets which are widely used in graphic layout generation tasks.

Publaynet [42] This is a dataset for document layout analysis, which contains 360k+ document layouts with 5 element types. In our experiment, we exclude layouts with more than 9 elements and use 160k+ layouts of the official training split for training, the rest 8k+ layouts for validation, and the official validation split (i.e., 4k+ layouts) for testing.

Rico [5] This dataset provides UI layouts with 27 element types collected from Android apps. Since some element types in the dataset appear less frequently, which may affect the performance of the model on these elements, we follow [19, 25] and only preserve the 13 most frequent elements in the dataset. Similar to Publaynet, we also exclude layouts with more than 9 elements. Since there is no official split, we use 17k+ layouts for training, 1k+ layouts for validation and 2k+ layouts for testing.

Magazine [41] This dataset contains 3,919 magazine layouts covering 6 common categories, including fashion, food, news, science, travel, and wedding. Each page is annotated with 6 different semantic elements. In our experiment, we exclude layouts with more than 10 elements and obtain 2.5k+ layouts for training, 100+ layouts for validation, and 300+ layouts for testing validation. We also input magazine categories as constraints into the model.

5.2 Evaluation Metrics

To measure the generation performance, we use four metrics representing different aspects of perceptual quality: Fréchet Inception Distance (FID), Maximum Intersection over Union (MaxIoU), Alignment, and Overlap. For the reconstruction experiments, we use FID, MaxIoU AND W_{bbox} to evaluate the similarity between the reconstructed layouts and real layouts.

FID This metric describes the distribution difference between real and generated layouts. Following [19], we obtain the layout embedding by training a neural network to discriminate ground-truth layouts from noise-added layouts on Publaynet and Rico, and calculate the distance between the features of real layouts and the generated layouts for FID.

MaxIoU Maximum IoU is defined to calculate the similarity between two collections of generated layouts and real layouts. Following [19], we first calculate the similarity of two layouts $B = \{b_i\}_{i=1}^N$ and $B' = \{b'_i\}_{i=1}^N$ under the optimal matching of elements as:

$$s_{IoU}(B, B', C) = \max_{\pi \in P_N} \frac{1}{N} \sum_{i=1}^N \text{IoU}(b_i, b'_{\pi(i)}), \quad (10)$$

where P_n represents all possible matching methods of two sets of elements with length n , $\text{IoU}(\cdot)$ calculates the Intersection over Union of the two bounding boxes. Note that the matching element categories must be the same. Then we calculate the MaxIoU between the two layout collections $\mathcal{B} = \{B_m\}_{m=1}^M$ and $\mathcal{B}' = \{B'_m\}_{m=1}^M$ as:

$$\text{Max IoU}(\mathcal{B}, \mathcal{B}', \mathbb{C}) = \max_{\pi \in P_M} \frac{1}{M} \sum_{m=1}^M s_{IoU}(\mathcal{B}, \mathcal{B}'_{\pi(m)}), \quad (11)$$

where the matching layouts must have the same set of categories.

W bbox Wasserstein distance also describe the distance between the real and learned data distributions. Unlike FID, which calculates the similarity at the feature level, we can use W_{bbox} [2] to accurately evaluate the distance between bounding boxes of real and generated layout.

Alignment and Overlap These two metrics are employed to measure the quality of layout in terms of aesthetics. Following [26], we measure six possible alignment types (i.e., Left, X-center, Right, Top, Y-center and Bottom aligned) among adjacent elements and take the smallest of them:

$$s_{\text{Alignment}}(B) = \frac{1}{N} \sum_{i=1}^N \min_{\forall j \neq i} l \left(\min_{t \in T} \Delta b_i^t - \Delta b_j^t \right), \quad (12)$$

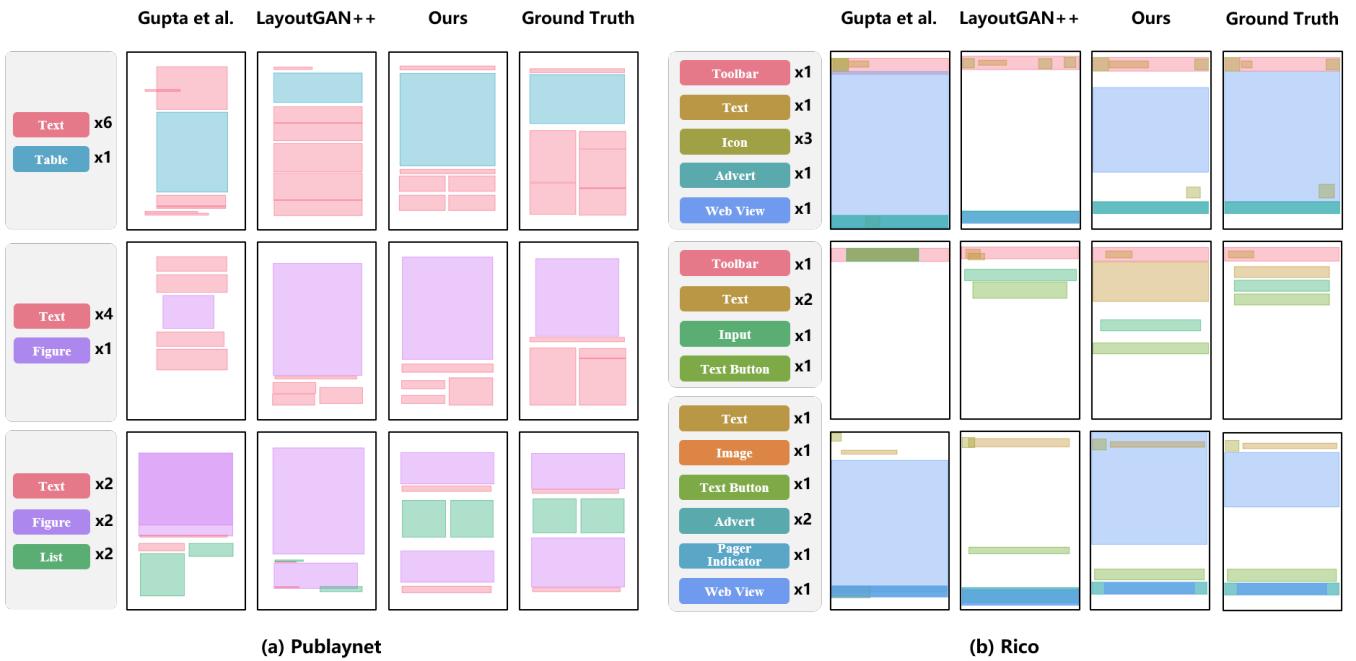
where $l(x) = -\log(1-x)$, T represents the set of six alignment types. For overlap, we calculate the average intersection area of any two elements in the layout:

$$s_{\text{Overlap}}(B) = \frac{1}{2N} \sum_{i=1}^N \sum_{\forall j \neq i} \frac{a_i \cap a_j}{a_i}, \quad (13)$$

where $a_i \cap a_j$ represents the intersection area of element i and j .

Table 1: Quantitative comparisons of element labels constrained layout generation.

	Rico				Publaynet			
	FID↓	MaxIoU↑	Alignment↓	Overlap↓	FID↓	MaxIoU↑	Alignment↓	Overlap↓
LayoutGAN++	13.65±0.29	0.36±0.01	0.58±0.03	66.04±0.64	24.35±0.39	0.34±0.01	0.19±0.01	15.99±0.14
Gupta et al.	10.35±0.01	0.36±0.01	0.36±0.01	64.32±0.01	20.61±0.01	0.33±0.01	0.13±0.01	10.65±0.01
Ours	6.48±0.19	0.47±0.01	0.28±0.05	85.84±0.65	20.49±0.17	0.39±0.01	0.14±0.01	22.65±0.34
Real data	4.47	0.65	0.26	50.58	9.54	0.53	0.04	0.22

**Figure 5: Qualitative comparisons of element labels constrained layout generation.**

5.3 Layout Generation with Internal Constraints

Settings In this experiment, we use **Gupta et al.** [10] and **LayoutGAN++** [19] as our baselines and evaluate the performance of our model in layout generation constrained with element labels. Both of the baselines are implemented with Official codes. The model in Gupta et al. first leverages self-attention mechanism to learn contextual relationships between layout elements and achieves the state-of-the-art performance regarding unconditional layout generation. It represents a layout as $[<\text{bos}>, c_1, b_1^x, b_1^y, b_1^h, b_1^w, \dots, c_n, b_n^x, b_n^y, b_n^h,$

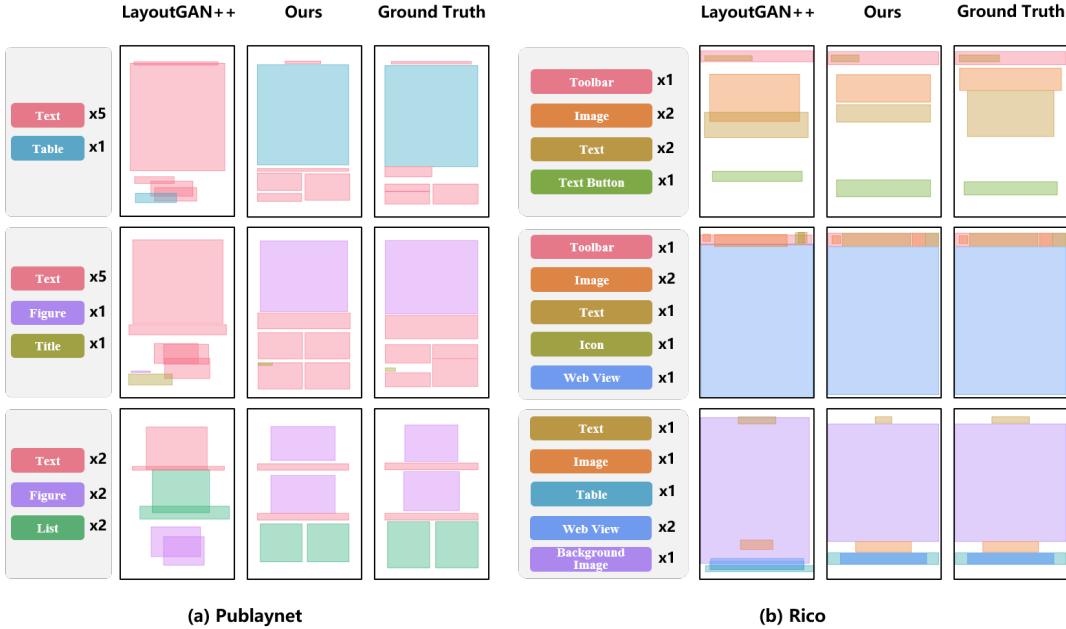
$b_n^w, <\text{eos}>]$, starts with the $<\text{bos}>$ token, predicts the category and geometric parameters of each element one by one, and ends when the $<\text{eos}>$ token is predicted. In practice, to satisfy the requirement of controllable element labels in the experiment, we adapted its code: ignore the element categories and $<\text{eos}>$ token predicted by the model, and replace it with the given ones, so that the model only needs to predict the geometric parameters according to the input constraints. LayoutGAN++ is a recent work for conditional layout generation which employs a GAN framework

to generate layout and uses optimization algorithms to adjust the aesthetic quality of layout. To focus on the performance of the generative model, we omit its optimization algorithm in this experiment. Qualitative and quantitative experimental results are based on the layouts generated by each trained model on the test-sets of Publaynet and Rico. Especially in quantitative experiments, we performed five sampling generations for each sample in test-set to eliminate the bias caused by random sampling in the process of model generation, and calculated the mean and standard deviation of each metric.

Results Table 1 shows the quantitative comparisons of generation performance. Following [19], the FID and MaxIoU of real data are computed between the validation and test data, and the Alignment and Overlap are computed with the test data for reference. On both Publaynet and Rico, our method achieves the best performance in the evaluation of FID and MaxIoU, thanks to the discrete layout representation. This illustrates that the layout distribution learned by our model is the closest to the ground truth. In terms of aesthetics, our model achieves the best or comparable alignment score which benefits from the discrete representation of geometric parameters of elements. Compared with Rico dataset, the layouts

Table 2: Quantitative comparisons of layout construction.

	Rico			Publaynet		
	FID↓	MaxIoU↑	$W_{bbox}\downarrow$	FID↓	MaxIoU↑	$W_{bbox}\downarrow$
LayoutGAN++	23.21	0.39	0.084	109.3	0.26	0.133
Ours	6.23	0.55	0.053	17.79	0.49	0.044

**Figure 6: Qualitative comparisons of layout construction.**

generated by our model on Publaynet dataset has a larger gap with the performance of real layouts on several metrics, which is caused by the unique document layouts in Publaynet. In more detail, the layouts in this dataset are obtained from PDF articles that are rigorously typeset and publicly available on PubMed Central [42]. The elements in them must be strictly aligned, closely arranged but avoid overlapping, and elements with the same category may appear in different positions of the same layout (e.g., there are usually multiple closely arranged text elements in a document layout), which increases the difficulty for the model to learn the relationship between elements. On the contrary, the categories of elements in UI layouts have certain functional semantics and their positions are relatively fixed (e.g., toolbar elements are usually at the top of layouts), which is easy to be modeled. Figure 5 provides the qualitative comparisons. It can be seen that our method produces the highest-quality layouts, especially with great alignment and rational arrangement of different types of elements. The results of LayoutGAN++ are close to ours, but some layouts have the problem of uneven size distribution and misalignment of elements. Due to the use of unidirectional Transformer, Gupta et al. can not obtain the type and number of elements that appear later [23] and fail to arrange all elements from a global perspective, so that produces poor results in conditional generation. On the contrary, our model

generates only latent representation of layout with the unidirectional Transformer, and then feeds the latent representation into a decoder whose backbone is a bidirectional transformer to predict all bounding boxes simultaneously, effectively addressing the above issues.

5.4 Layout Reconstruction

Settings To further illustrate the advantages of our proposed discrete latent representation of layout, we evaluate the performance of our VQ-VAE model in layout reconstruction. In this experiment, we employ the neural network used for layout reconstruction in **LayoutGAN++** as a baseline (i.e., the discriminator and auxiliary decoder in LayoutGAN++), which leverages the similar encoder and decoder as ours but represents layout with a continuous feature vector. In addition, in order to measure the similarity between the input real layout and the reconstructed layout, we only perform the optimal matching of the elements in Equation 10 when calculating MaxIoU.

Results Table2 shows the quantitative comparisons. Our model achieves the better performances on all metrics, which indicates that the layout generated by our model is closer to the real layout in both feature distribution (illustrated by FID) and bounding box distribution (illustrated by MaxIoU and W_{bbox}). Figure6 shows the qualitative comparisons. It can be found that our model can not only

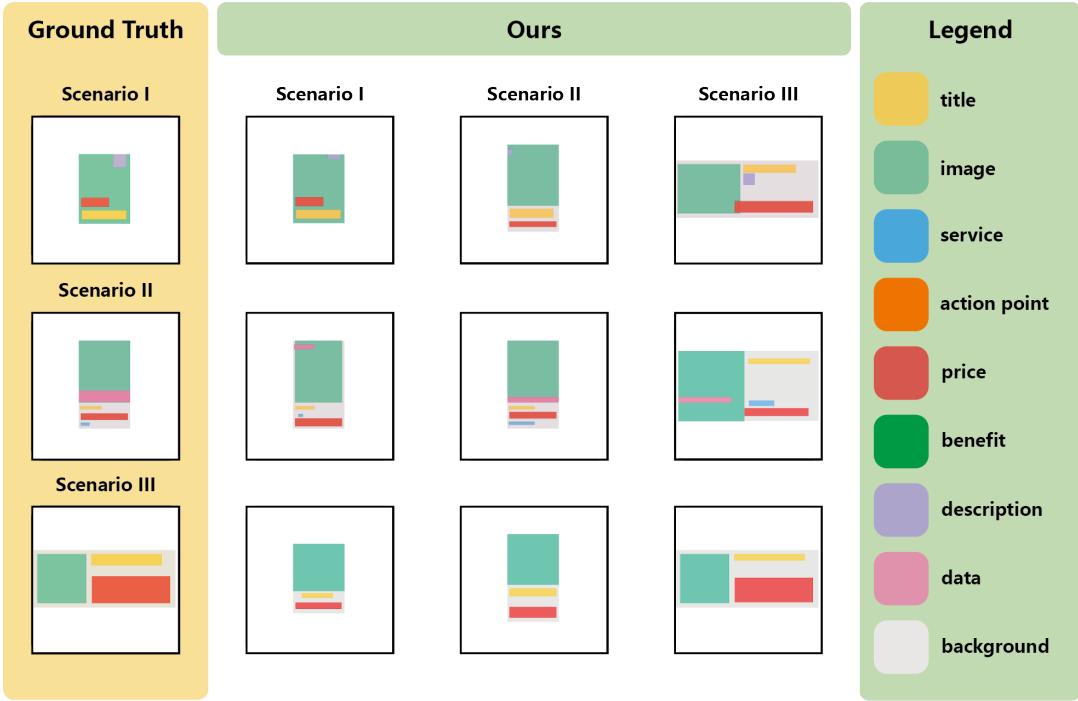


Figure 7: Results of our model on PDCard dataset. In each line, the real layout (Ground Truth) and its scenario are shown on the left, the layouts generated with the same element labels and 3 different scenarios by our model are shown on the left, where Scenario I represents Product Recommendation, Scenario II represents Product by Category and Scenario III represents Product Search.

reconstruct the structure of the ground truth, but also accurately restore the position and size of elements in detail. LayoutGAN++ can roughly capture the layout structure, but is not precise enough while predicting the bounding boxes, and suffers from misalignment and overlapping problems. In general, both qualitative and quantitative experimental results show that discrete latent vectors are better at reconstructing the global structure and local details of the layout, and are more conducive to representing layout features.

5.5 Layout Generation with External Constraints

Settings Since no other model can generate layouts based on scenarios, and the impact of scenarios on layouts is difficult to quantify, we qualitatively evaluate the performance of our model by comparing different generated results obtained from the same element labels but different scenario constraints on PDCard dataset. For the sake of uniform representation of layouts with different aspect ratios in PDCard, similar to [39], we use an element with "background" type to represent the original canvas of the layout, and define a larger canvas that can accommodate all layouts as a coordinate reference. Therefore, the model has to predict not only the geometric parameters of the elements in the layout, but also the width, height and position of the "background" element, where the width and height (i.e., the aspect ratio of product card layout) is also affected by the constraints of scenario.

In addition, to demonstrate the generality of our scenario constrained generative model in other types of graphic layouts, we also performed an experiment similar to the above on the Magazine dataset. Especially, the concept of "magazine category" makes similar sense to magazine layout as the "scenario" to product card layout, so we can define it as the external constraint and generate layouts with the same element labels but different magazine categories to evaluate the performance of our model.

Results Figure 7 shows the generated results on PDCard. By comparing the layouts in the same row, it can be found that even with the same element labels, our model can generate distinctly styled layouts based on different shopping scenarios. For example, the image element occupies a larger area in the product recommendation layout (Scenario I), while that in the product search layout (Scenario III) is smaller, which is similar to the distribution of image proportions in the real data (see in Figure 2(a)). Besides, the size and aspect ratio of the generated layout also changes with the scenario. The layouts for product recommendation and product by category are both a vertical layouts, but those for product search are horizontal layouts. The size of layouts for the three scenarios is gradually increasing. These results are all consistent with the explicit rules proposed in our design space and prove that our model can capture the relationship between layouts and scenarios.

In Figure 8, the resulting layout also has a rich variety of magazine categories. For instance, the layouts for science and news

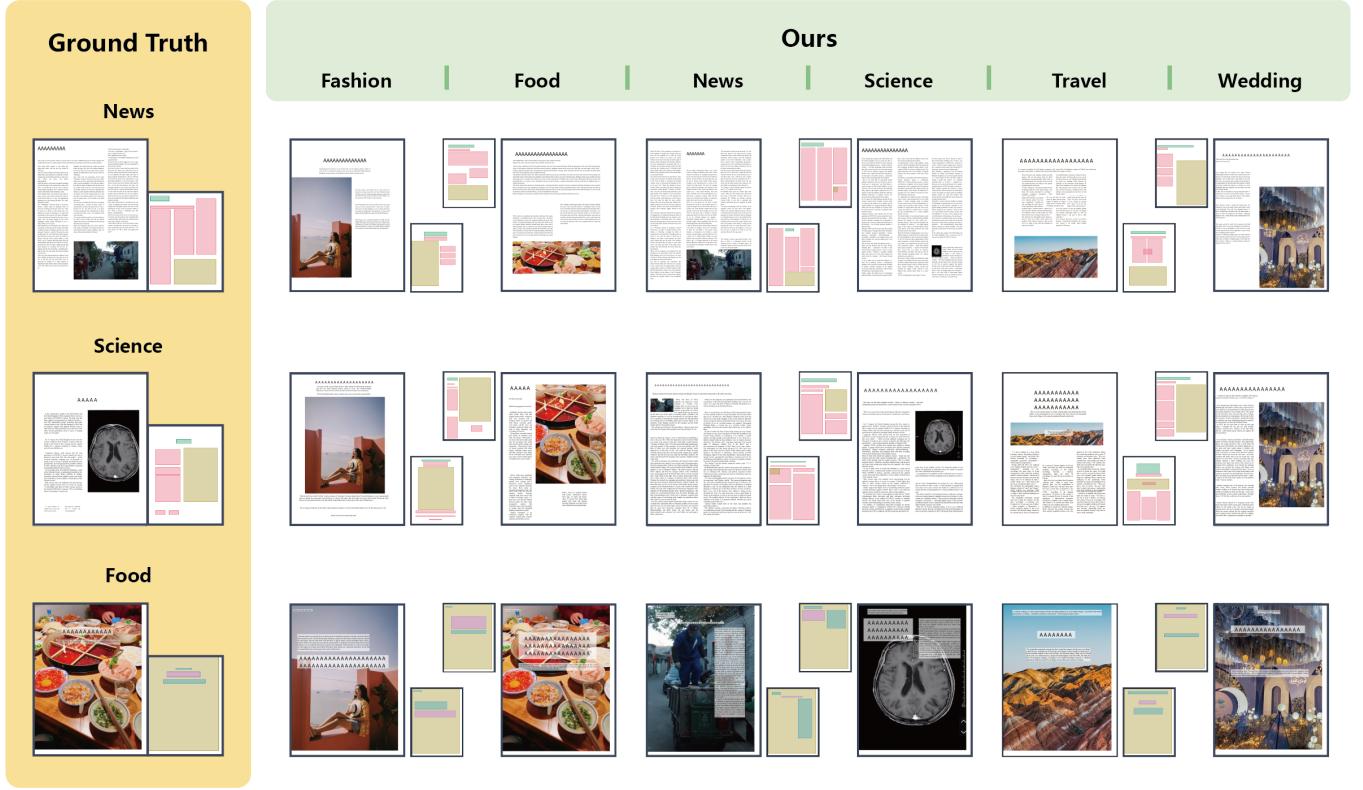


Figure 8: Results of our model on Magazine dataset. In each line, the real layout (Ground Truth) and its magazine category are shown on the left, the layouts generated with the same element labels and 6 different categories by our model are shown on the left. The rendering images of the layouts are also shown in the figure.

magazines are usually structured and aligned, but the ones for others are more casual and varied, especially for fashion. These results accord to our common sense that science and news magazines are more serious and require rigorous layout, while fashion and food magazines are prepared for entertainment and leisure and require more innovative and unconventional layout. Therefore, we can say that our model is capable in the generation with external constraints for other layout types.

6 CASE STUDY

In this section, we conducted two case studies. Study I was utilized to evaluate the applicability of product card layouts generated by our model, and Study II was used to evaluate whether our proposed approach, including the design space and generative model, could improve the efficiency of PLP layout design while ensuring the applicability and diversity of the layout in specific scenarios.

6.1 Study I: Evaluating Layout Applicability

Method To evaluate the applicability of the generated layout, Study I asks people to choose the most suitable one among two generated layouts with and without scenario constraints. Based on the PDCard test-set, we used LayoutVQ-VAE to generate 257 groups of layouts, each of which has two layouts with the same element labels, but one with the scenario constraints, denoted scenario-based layout,

and one without, denoted common layout. After generation, we randomly selected 12 groups of layouts (4 groups per scenario) from the results as stimulus in the study, and invited a designer to render the layouts as realistic product cards for reference. The participants are 60 college students who were publicly recruited from a social networking site. We paid each participant \$10 per hour, and the average time to complete the study was about 9 minutes.

The study was conducted through an online questionnaire. We first introduced three online shopping scenarios described in our design space to the participants, and presented two examples of real page screenshots from TaoBao for each scenario. To test whether the participants obtained the above information accurately, they were asked to answer several questions related to the above content. After completing the preliminary test, we presented each participant with 6 groups of layouts randomly selected from the above 12 groups. The orders of the groups and the two layouts in each group are completely shuffled to avoid bring bias. For each group, participants were asked to choose the layout with higher aesthetic quality, better presentation of the two main information types (visual and textual information), and better applicability. To improve the granularity of samples, the first three indicators about quality, visual and textual performance are measured with a 5-point Likert scale (i.e., 1: layout I performs much better, 2: layout I performs slightly better, 3: equal performance, 4: layout II performs slightly better, 5: layout

Table 3: The results of Study I showing average ratings of the participants in the aspect of aesthetic quality, image presentation and text presentation.

Scenario	Aesthetic Quality	Image Presentation	Text Presentation
Product Recommendation	4.09±1.28	4.13±1.26	3.75±1.47
Product by Category	4.13±1.39	3.98±1.36	3.91±1.46
Product Search	3.49±1.55	3.28±1.43	3.44±1.55

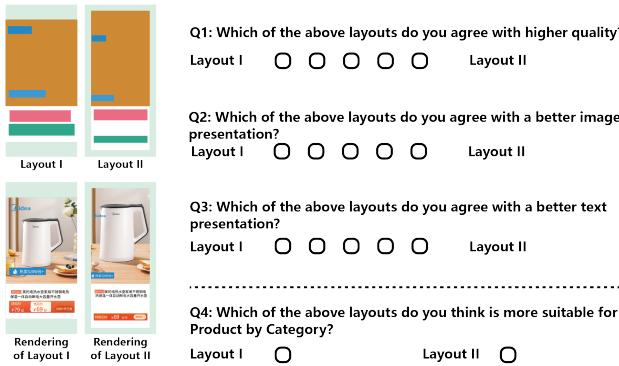


Figure 9: Examples for a group of layouts and related problem settings in Study I.

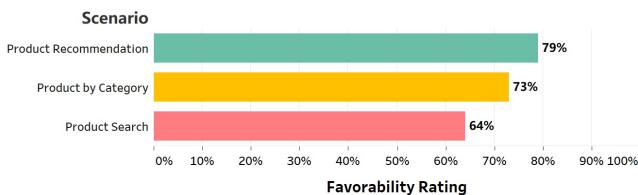


Figure 10: The results of Study I showing the favorability ratings of the product card layouts with scenario constraints generated by LayoutVQ-VAE.

II performs much better), while the final question on applicability takes a dichotomous format to allow participants to make a clear choice. Figure 9 shows a group of layouts and problem settings in the questionnaire as an example. Afterward, We first filtrated the questionnaires according to the results of the preliminary test and obtained 51 valid questionnaires, then revised the order of layouts and the value of answers in the groups which are disrupted, in detail, all common layouts were recorded as layout I, corresponding to rating 1 in the 5-point answer, and all scenario-based layouts were recorded as layout II, corresponding to rating 5. Finally, we processed the filtered and revised results.

Results Table 3 shows the evaluation results of the aesthetic quality, image presentation, and text presentation of the two kinds of layouts in the three scenarios. In theory, the model without scenario constraints learns the distribution of all layouts in the dataset and generates layouts with no regard to each scenario, while the model with scenario constraints can classify layouts according to

the input scenario labels, and model the layout distribution corresponding to each scenario. Thus, we can regard the common layout as the baseline and vertically compare the metric results of scenario-based layout. The value of 3 means that the two kinds of layouts has the same performance, and the value closer to 5, the scenario-based layout performs better. It can be found that the scenario-based layout is superior to the common layout on all indicators for each scenario. Besides, the image presentation of the scenario-based layouts in the three scenarios is gradually declining, inversely the text presentation of product by category is better than that of product recommendation. However, the text presentation of product search is abnormal, even the results of each indicator in the search scenario are only slightly greater than 3. The main reason is that product card layouts usually consist of one image, one title and several attribute elements, and the scenario of layouts with a large number of attribute elements in the dataset are mostly labelled product search, as shown in Figure 2(d). Therefore, even if the scenario constraints are not input into the model, it can still learn the layout features corresponding to the search scenario based only on the input element labels.

Figure 10 presents the support rate of the scenario-based layout in the current scenario (results of the last question), from which it can be seen that scenario-based layout is far favorable than common layout in the first two scenarios. Although the value in product search is slightly lower, it is also greater than 60%. Overall, results of the study show that comparing with the common layout, the scenario-based layout achieves higher aesthetic quality, the change trend of its visualization and text information performance in the three scenarios is consistent with the real data, and it is more suitable for the target application.

6.2 Study II: Applying Design Space and LayoutVQ-VAE to PLP Design

Method The purpose of this study is to verify whether our approach, including the design space and generative model, could improve the efficiency of creating high-quality scenario-based PLPs layouts. To this end, three groups of participants were invited to conduct a PLP design challenge, and sixty participants were invited for the evaluation.

Each group in the PLP design challenge has three designers with at least two years of design experience, and each designer was asked to design a PLP for the three shopping scenarios respectively. The design process was observed and timed by the coordinator. One group (the *Designer* group) was asked to use general graphic design tools (e.g., Sketch) to create PLPs, another group (the *Template* group) was required to use 30 product card templates which are commonly used and provided by an online shopping platform, and the last group (the *Intelligent Generation* group) was introduced

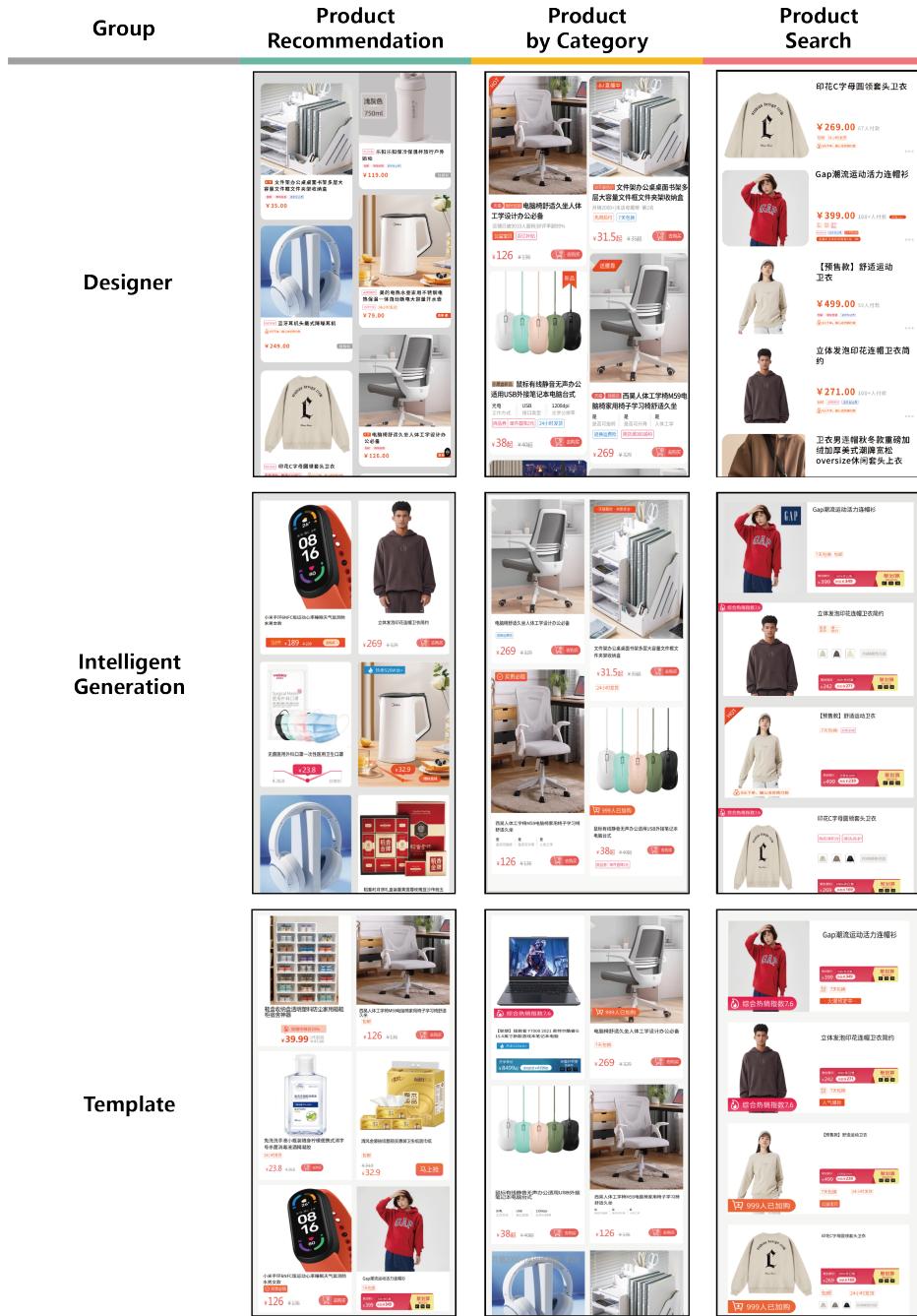


Figure 11: Examples for the scenario-based PLPs designed by Designer Group, Template Group and Intelligent Generation Group in Study II.

to use our approach, including the design space and generative model, LayoutVQ-VAE. We provided the three groups with the same materials (i.e., UI elements in SVG format) and introductions to the three shopping scenarios. For the Intelligent Generation group, we also introduced the workflow of PLPs design using our approach as the following: Given a shopping scenario, the designers can

segment the entire page and create the mall layout of it according to the hints of the design space, and then select the UI elements to be displayed for each product card, including product's image, title, and attribute elements. After this, they can input the target scenario and element labels into our model LayoutVQ-VAE, which will generate ten layouts to choose at a time. Finally, the designers can insert the

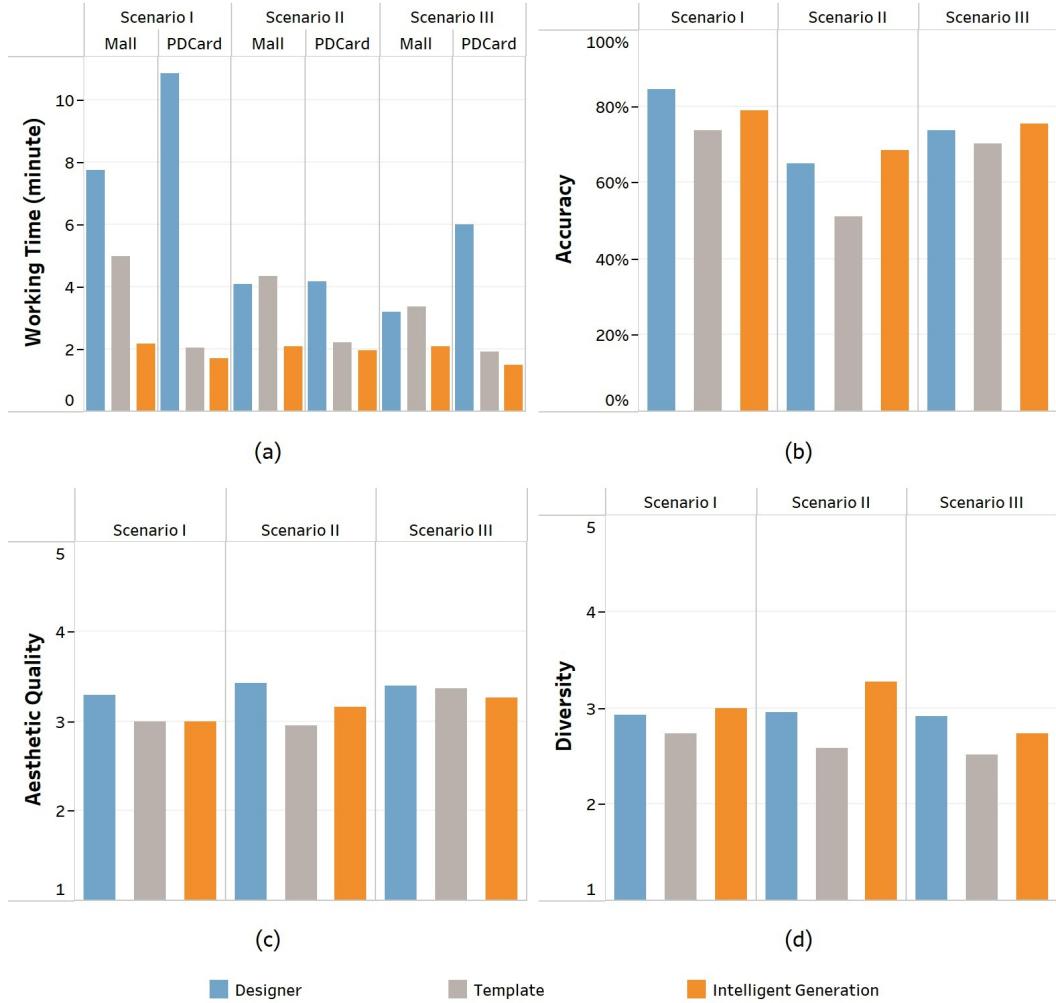


Figure 12: The results of Study II. (a) Average time to create a mall layout and a product card layout; (b) Average matching accuracy between the target scenario of the PLPs and the scenario judged by the participants; (c) Ratings on aesthetic quality of the PLPs by participants; (d) Ratings on diversity of the PLPs by participants. Scenario I represents Product Recommendation, Scenario II represents Product by Category, and Scenario III represents Product Search.

selected product card layouts into the mall layout and complete the PLP design. To test whether these three groups can produce suitable layout according to given constraints, participants were required to select the UI elements to be displayed before designing the layout. Specifically, the Template group needed to select the template best fitting the element labels and scenario, the Intelligent Generation group needed to choose from the layouts generated by the model, and both of them were not allowed to optimize the size and position of an element other than its aspect ratio. In the end, twenty-seven PLPs were produced by nine designers from the three groups (some of them are shown in Figure 11).

In the evaluation, we recruited sixty college students on a social networking site and conducted the study through an online questionnaire. Each participant received a scenario introduction and a pre-test at the beginning of the questionnaire. In the formal test,

we randomly selected 9 of the above 27 PLPs and showed them to the participants one by one. For each PLP, the participants were asked to choose in which shopping scenario the PLP layout would satisfy their needs, and rate the aesthetic quality and the diversity of product card layouts using a five-point Likert scale (1: very poor, 2: poor, 3: medium performance, 4: good, 5: very good). Each participant was paid \$10 per hour, and the average time to complete the questionnaire was about 8 minutes. After filtering results based on the pre-test, 57 valid questionnaires are finally processed.

Results Figure 12(a) shows the average time spent by the three groups of designers on designing one mall layout and one product card layout for each scenario. It can be seen that both our proposed approach and template-based approach can significantly reduce the working time of designers. Especially in the stage of creating product card layouts, the Intelligent Generation group

spent 84.47% (Scenario I, $AVG_D = 10.840$, $SD_D = 1.432$, $AVG_{IG} = 1.683$, $SD_{IG} = 0.062$), 53.28% (Scenario II, $AVG_D = 4.167$, $SD_D = 0.261$, $AVG_{IG} = 1.947$, $SD_{IG} = 0.144$), and 75.27% (Scenario III, $AVG_D = 6.003$, $SD_D = 1.304$, $AVG_{IG} = 1.487$, $SD_{IG} = 0.325$) less time respectively on designing a layout for the three scenarios than the Designer group. In this stage, thanks to the guidance of the design space and the capability of LayoutVQ-VAE to quickly generate scenario-based layouts (0.138s per layout on average), the designers can create a product card layout within two minutes on average, including UI element selection, layout generation, element aspect ratio optimization and layout export. In the stage of creating mall layout, the working time was also reduced compared with the other two groups, which benefits from the mall layout pattern recommended by the design space. Another finding observed from the figure is that the time spent by the Designer group on designing product card layouts for the three scenarios is gradually decreasing, while the time taken by the other two groups using automatic generation tools is relatively consistent. This is caused by the creation order and habits of participants. According to our observation, the participants in the Designer group usually start from Product Recommendation, and for the first PLP, they need to create the entire layout from a blank design draft, which will take up much more time. But in the following design for other scenarios, participants usually copy the previous design draft and adjust the layout according to the scenario constraints and UI elements, so that they can improve the work efficiency.

Figure 12(b) shows the matching accuracy between the target scenario of PLP and the scenario judged by the participants. The higher matching accuracy means the PLPs meet the requirements of consumers in the target shopping scenarios better. The figure demonstrates that the Intelligent Generation group can obtain the results comparable with the Designer group, but there is a certain gap between the performance of the Template group and the other two. This is caused by the limited variation and pre-designed nature of the template set. It is difficult for designers to choose a layout that perfectly fits the scenario requirements and element labels in the limited set of templates, so a relatively low matching accuracy rate is obtained. This shortcoming also leads to another problem shown in Figure 12(d): the lack of variation between individual product cards in the PLPs created by the Template group. Different from the Template group, LayoutVQ-VAE used by the Intelligent Generation group modeled the relationship between constraints and layout data through training, which is beneficial for it to generate a layout that matches the constraints input by designers. Moreover, since the layout generated by the model is obtained by decoding the sampled layout latent variables, there is a certain randomness in the sampling process, so various layouts can be generated through multiple sampling. This explains why the PLPs produced by the Intelligent Generation group achieved higher scores in the diversity evaluation, which was similar to that of the Designer group. In the aesthetic quality evaluation shown in Figure 12(c), the Intelligent Generation group and the Template group perform almost equally, while the Designer group obtains higher scores. In general, using our approach to create PLPs can not only reduce the working time significantly, but also ensure the quality of PLPs in terms of scenario constraint satisfaction, aesthetics, and diversity.

7 CONCLUSION

In this paper, we propose a design space and a conditional generative model, LayoutVQ-VAE, to improve the performance of large-scale high-quality PLP layout creation within various scenarios for online shopping platforms. In the design space, we classify common shopping scenarios into three categories: product recommendation, product by category and product search. For each scenario, we analyze it in terms of consumer characteristics, displayed product information, and layout patterns. Especially for the layout patterns, we conclude some explicit rules by summarizing the experience of senior designers and analyzing real data. In LayoutVQ-VAE, the discrete latent representation of layout is learnt by training a novel VQ-VAE and we model the relationship between the discrete representation and constraints through a unidirectional Transformer. The above innovations enable our model to obtain comparable or better results than state-of-the-art methods in layout generation tasks as well as in layout reconstruction tasks. Extensive experiments on multiple datasets with different layout types also illustrate the generality of our model in graphic layout design constrained with scenarios. Through two case studies, we also prove that our design space can assist designers to make reasonable decisions about mall layout design and UI elements selection, and our generative model can synthesize multiple appropriate product card layouts within seconds, which are both beneficial to quickly create a large number of high-quality and diverse PLPs satisfying the requirements of shopping scenarios. In the future, we plan to further expand our research, such as exploring the impact of product type and consumer characteristics on layout, and develop an intelligent PLP design system based on our generative model to produce large-scale personalized and scenario-based PLPs for different target consumers in real time, which is the ultimate goal of intelligent UI for mobile shopping platforms.

ACKNOWLEDGMENTS

We thank all the participants for their time and the anonymous reviewers for their valuable comments. This research is supported by Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies and National Natural Science Foundation of China (62207023).

REFERENCES

- [1] [n. d.]. Sketch. <https://www.sketch.com/>
- [2] Diego Martin Arroyo, Janis Postels, and Federico Tombari. 2021. Variational Transformer Networks for Layout Generation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Nashville, TN, USA, 13637–13647. <https://doi.org/10.1109/CVPR46437.2021.01343>
- [3] Sara Bunian, Kai Li, Chaima Jemmali, Casper Harteveld, Yun Fu, and Magy Seif Seif El-Nasr. 2021. VINS: Visual Search for Mobile User Interface Design. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–14. <https://doi.org/10.1145/3411764.3445762>
- [4] Niraj Ramesh Dayama, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. 2020. GRIDS: Interactive Layout Design with Integer Programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376553>
- [5] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hirschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) (UIST '17). Association for Computing Machinery, New York, NY, USA, 845–854. <https://doi.org/10.1145/3126594.3126651>

- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [7] Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, Vol. 1. IEEE, 347–352.
- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.
- [9] Shuhan Guo, Zhuochen Jin, Fuling Sun, Jingwen Li, Zhaorui Li, Yang Shi, and Nan Cao. 2021. Vinci: An Intelligent Graphic Design System for Generating Advertising Posters. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–17. <https://doi.org/10.1145/3411764.3445117>
- [10] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry Davis, Vijay Mahadevan, and Abhinav Shrivastava. 2021. LayoutTransformer: Layout Generation and Completion with Self-attention. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Montreal, QC, Canada, 984–994. <https://doi.org/10.1109/ICCV48922.2021.00104>
- [11] Ruohong Hao, Bingjia Shao, and Rong Ma. 2019. Impacts of Video Display on Purchase Intention for Digital and Home Appliance Products—Empirical Study from China. *Future Internet* 11, 11 (Nov. 2019), 224. <https://doi.org/10.3390/fi1110224> Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.
- [12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [13] Weiyan Hong, James YL Thong, and Kar Yan Tam. 2004. Designing product listing pages on e-commerce websites: an examination of presentation mode and information format. *International Journal of Human-Computer Studies* 61, 4 (2004), 481–503. Number: 4 Publisher: Elsevier.
- [14] WEIYIN HONG, JAMES Y.L. THONG, and KAR YAN TAM. 2004. The Effects of Information Format and Shopping Task on Consumers' Online Shopping Behavior: A Cognitive Fit Perspective. *Journal of Management Information Systems* 21, 3 (2004), 149–184. <https://doi.org/10.1080/07421222.2004.11045812> arXiv:<https://doi.org/10.1080/07421222.2004.11045812>
- [15] Zhenhui Jiang and Izak Benbasat. 2007. Research note—investigating the influence of the functional mechanisms of online product presentations. *Information Systems Research* 18, 4 (2007), 454–470. <https://doi.org/10.1287/isre.1070.0124>
- [16] Zhaoyun Jiang, Shizhao Sun, Jihua Zhu, Jian-Guang Lou, and Dongmei Zhang. 2022. Coarse-to-Fine Generative Modeling for Graphic Layouts. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 1 (June 2022), 1096–1103. <https://doi.org/10.1609/aaai.v36i1.19994>
- [17] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. 2019. LayoutVAE: Stochastic Scene Layout Generation From a Label Set. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- [18] Lukasz Kaiser, Aurko Roy, Ashish Vaswani, Niki Parmar, Samy Bengio, Jakob Uszkoreit, and Noam Shazeer. [n. d.]. Fast Decoding in Sequence Models Using Discrete Latent Variables. ([n. d.]), 10.
- [19] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. 2021. Constrained Graphic Layout Generation via Latent Optimization. In *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, Virtual Event China, 88–96. <https://doi.org/10.1145/3474085.3475497>
- [20] Minjeong Kim and Sharon Lennon. 2008. The effects of visual and verbal information on attitudes and purchase intentions in internet shopping. *Psychology & Marketing* 25, 2 (Feb. 2008), 146–178. <https://doi.org/10.1002/mar.20204> Publisher: John Wiley & Sons, Ltd.
- [21] Seeun Kim, Tae Hyun Baek, and Sukki Yoon. 2020. The effect of 360-degree rotatable product images on purchase intention. *Journal of Retailing and Consumer Services* 55 (July 2020), 102062. <https://doi.org/10.1016/j.jretconser.2020.102062>
- [22] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [23] Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. 2022. BLT: Bidirectional Layout Transformer for Controllable Layout Generation. <http://arxiv.org/abs/2112.05112> arXiv:2112.05112 [cs].
- [24] Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B. Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. 2020. Neural Design Networks: Graphic Layout Generation with Constraints. In *Computer Vision – ECCV 2020 (Lecture Notes in Computer Science)*. Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 491–506. https://doi.org/10.1007/978-3-030-58580-8_29
- [25] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. 2019. LayoutGAN: Generating Graphic Layouts with Wireframe Discriminators. <https://doi.org/10.48550/arXiv.1901.06767> arXiv:1901.06767 [cs].
- [26] Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. 2021. Attribute-Conditioned Layout GAN for Automatic Graphic Design. *IEEE Transactions on Visualization and Computer Graphics* 27, 10 (Oct. 2021), 4039–4048. <https://doi.org/10.1109/TVCG.2020.2999335>
- [27] Mengxiang Li, Kwok-Kei Wei, Giri Kumar Tayi, and Chuan-Hoo Tan. 2016. The moderating role of information load on online product presentation. *Information & Management* 53, 4 (June 2016), 467–480. <https://doi.org/10.1016/j.im.2015.11.002>
- [28] Gerald L Lohse and Peter Spiller. 1998. Electronic shopping. *Commun. ACM* 41, 7 (1998), 81–87.
- [29] Naresh K. Malhotra. 1982. Information Load and Consumer Decision Making. *Journal of Consumer Research* 8, 4 (03 1982), 419–430. <https://doi.org/10.1086/208882> arXiv:<https://academic.oup.com/jcr/article-pdf/8/4/419/5068452/8-4-419.pdf>
- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2019/file/bdbca288ee/f92f2bfa9f701272740-Paper.pdf>
- [31] Akshay Gadi Patil, Omri Ben-Eliezer, Or Perel, and Hadar Averbuch-Elor. 2020. READ: Recursive Autoencoders for Document Layout Generation. 544–545. https://openaccess.thecvf.com/content_CVPRW_2020/html/w34/Patil_READ_Recursive_Autoencoders_for_Document_Layout_Generation_CVPRW_2020_paper.html
- [32] Steven F. Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. 1994. Interactive graphic design using automatic presentation knowledge. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 112–117.
- [33] Reijo Savolainen. 2009. Information use and information processing: Comparison of conceptualizations. *Journal of Documentation* 65, 2 (Jan. 2009), 187–207. <https://doi.org/10.1108/00220410910937570> Publisher: Emerald Group Publishing Limited.
- [34] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80. <https://doi.org/10.1109/TNN.2008.2005605>
- [35] Maria Sicilia and Salvador Ruiz. 2010. The effects of the amount of information on cognitive responses in online purchasing tasks. *Electronic Commerce Research and Applications* 9, 2 (March 2010), 183–191. <https://doi.org/10.1016/j.eleralp.2009.03.004>
- [36] Mary Stribley. 10. Rules of Composition All Designers Live By. *Retrieved May 23 (10), 2016.*
- [37] Aaron Van Den Oord and Oriol Vinyals. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fb0d053c1c4a845aa-Abstract.html>
- [39] Kota Yamaguchi. 2021. CanvasVAE: Learning To Generate Vector Graphic Documents. 5481–5489. https://openaccess.thecvf.com/content/ICCV2021/html/Yamaguchi_CanvasVAE_Learning_To_Generate_Vector_Graphic_Documents_ICCV_2021_paper.html
- [40] Wanxian Zeng and Alex Richardson. 2016. Adding Dimension to Content: Immersive Virtual Reality for e-Commerce. *ACIS 2016 Proceedings* (Jan. 2016). <https://aisel.aisnet.org/acis2016/24>
- [41] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson W. H. Lau. 2019. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics* 38, 4 (Aug. 2019), 1–15. <https://doi.org/10.1145/3306346.3322971>
- [42] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. PubLayoutNet: Largest Dataset Ever for Document Layout Analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1015–1022. <https://doi.org/10.1109/ICDAR.2019.00166>