CLASS: Conditional Latent Architecture for Search and Synthesis of Design Layouts

Dipu Manandhar[1] Paul Guerrero[2] Zhaowen Wang[2] John Collomosse[1,2]
[1]CVSSP, University of Surrey [2]Adobe
dips4717@gmail.com guerrero@adobe.com zhawang@adobe.com j.collomosse@surrey.ac.uk

Abstract

We propose CLASS; a novel unified model for the synthesis and search for design layouts, two tasks that are often handled separately by prior works. We propose to learn a compact and coherent latent feature of a layout supporting joint search and synthesis. This allows various operations such style-conditioned layout generation, latent space manipulation and provides seamless integration of search and synthesis for an effective design workflow. We train CLASS with a dual decoder: a new transformer-based layout-conditioned decoder and a CNN-based raster decoder. The latent-conditioned decoder explicitly conditions upon a latent vector while generating a layout in an auto-regressive fashion. We train CLASS under variational framework which in conjunction with a raster-decoder enhances the latent representation improving both generation and retrieval performances. We show the effectiveness of CLASS on the RICO and PubLayNet benchmarks, and demonstrate that CLASS is capable of high-quality synthesis from scratch, as well as performing self-completion, interpolation, project between design layouts, whilst achieving close to or better than state-of-the-art search performance.

1. Introduction

Design layouts form the core of visual media and interactive applications where design components are organized for effective communication while providing a better user experience [5, 36]. Designing a layout from scratch is daunting and time-consuming. A preferred layout design workflow is to start from a reference layout matching a user's general requirement, based on which a user can further customize some key components with the remaining components being automatically adjusted to maintain the overall design e.g. retrieve-then-adapt [37]. At any point during the design process, assistive tools or designers can generate multiple conditioned layouts or search repositories

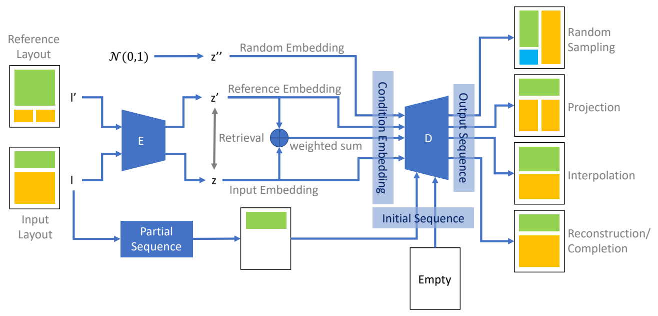


Figure 1. Our proposed CLASS framework supports various layout search and synthesis tasks. The framework conditions upon latent embedding for generative tasks such as conditional generation, self-completion, projection, interpolation, unconditional generation. The latent vectors are also used as search embeddings which achieve a competitive layout search performance.

with the current layout. In this way, new layouts can be efficiently created with a personalized experience. Such a workflow can be achieved with seamless integration of large repository layout search and user-enforced layout synthesis, which we explore in this paper.

Recently, strong interests in design layout modeling have emerged in research problems such as layout synthesis to generate realistic and diverse layouts, and search and retrieval tasks for automatic search on large layout collections. Although both are being explored actively, the generative and retrieval tasks have always been studied under different hoods in the literature and modeled with separate architectures, leaving a gap for applying these techniques together in real applications. Existing methods for layout search are based on convolution and graph-based autoencoders [28, 29, 35]. For layout synthesis, researchers have explored various architectures based on as VAEs [1, 18, 19, 21] and GANs [9, 20, 26]. The recent state-of-the-art methods use Transformer-based models [1, 11, 20, 38] and diffusion model [14, 16, 39]. While these achieve significantly higher generation quality than VAEs and GANs, they cannot be directly applied for layout search, since they do not produce a compact layout representation.

In this work, we propose to unify layout synthesis and search in an elegant architecture. For this purpose, we introduce a novel conditional latent architecture for search and synthesis (CLASS) of design layouts. As shown in Fig.1, our CLASS framework is based on an encoder-decoder network architecture. Our method learns the encoding of layouts as a compact latent feature representation, rather than a sequence. This latent representation can serve as search embedding to perform retrieval, as well as enabling additional and more practical generative tasks that are not supported by the existing methods, like interpolation between layouts and projection to the layout.

We found that a good distribution of the feature representations in latent space is important for both retrieval and interpolation tasks. To this end, we employ two strategies: i) similar to VAEs, we regularize the distribution of layout vectors in latent space to approximate a standard normal distribution; and ii) we add a CNN-based raster image decoder, in addition to the Transformer decoder, which allows us to apply an image-based loss as supervision. The image-based loss encourages neighboring layouts in latent space to be visually similar, while the sequence-based loss of the Transformer decoder encourages latent space neighbors to be structurally similar. We found that using both structural and visual similarity improves retrieval performance.

In contrast to the existing methods that tackle the search and synthesis tasks separately [1, 11, 20, 28, 29, 35], our unified CLASS framework seamlessly handles both tasks while achieving good performance for both search and synthesis – a key advantage of the proposed method. Different from the existing works on conditional generation, which often condition the generation using a set of labels [20], or a set of labels and geometric constraints [24], we for the first time propose to directly operate on a latent representation for conditional layout generation. With this novel design, our method is capable of several generative tasks such as self-completion, projection of partial layouts towards a given reference layout, interpolating between layouts as well as the synthesis of novel layouts by sampling latent vectors from a known prior distribution as shown in Fig. 1.

We evaluate our method on two benchmark datasets: RICO [7] and PubLayNet [40]. We show that our method produces high-quality layouts that are perceptually plausible and diverse while supporting several generative subtasks. We compare with various baselines using standard metrics and show the effectiveness of the proposed method. Our method also achieves on-par or even better performance than state-of-the-art methods tailored to layout search tasks.

2. Related Work

2.1. Layouts Analysis

Early works on layouts have been studied for automated design and document formatting that mostly relied on templates [6, 8, 15] and exemplars [23]. These methods often require expert knowledge as they rely on predefined templates and heuristic rules, and hence do not capture complex layout distribution. The authors in [31, 32] proposed optimization techniques leveraging learnable metrics from exemplars for an interactive interface that assists layout design by providing automatic suggestions. Similarly, gaze mechanism [33] has been employed to assist designers in easily directing users' attention to desired locations in web layouts. In terms of design metrics, the aesthetics of document layout have been studied by [12] based on alignment, regularity, uniform separation and balance properties of the design components.

2.2. Layout Synthesis

Layout synthesis has recently gained great research interests and has been explored using various neural architectures in the field. LayoutGAN [26] is the first to study layout representation learning for synthesis with a differentiable wireframe renderer over input graphic elements and their geometric parameters. While capable of generating simple layout formats, it does not handle complex design layouts. READ [34] proposed a recursive neural network (RvNN) based autoencoder with heuristics to determine relationships between elements for document layout generation. Neural Design Networks (NDN) [24] proposed a graph neural network (GNN) based method that models the relationships of the components given partial user specifications. READ and NDN methods are both based on heuristics and do not comprehensively learn component relationships. Layout VAE [19] proposed to train two separate networks using conditional VAE to generate the counts and bounding boxes to the given label set. Most of these methods fail to capture long-range relationships between the design components, especially in a complex design scenario with a large number of components.

Transformer architecture has been recently studied for layout generations [1, 11, 18, 20]. The use of transformer networks for layouts has two main benefits. First, it naturally handles layouts with varying number of components. Second, the transformer attention modules inherently capture the relationships between the components, resulting in robust layout representation. LayoutTransformer [11] proposed an auto-regressive transformer decoder network with causal attention modules. Similarly, VTN [1] combined the standard transformer network with a variational module to sample from learned distribution for layout generation. LayoutGAN++ proposed by [20] used a transformer-based encoder-decoder network with a GAN objective together with an auxiliary decoder for reconstruction. Recently, a coarse-to-fine transformer model is proposed by [18] that generates a layout in two stages, decoding the regions first and then filling the regions with components. LayoutFormer++ [17] uses serialized input constraints and additionally impose restrictions to avoid sequence violation during decoding.

Concurrent to our work, recent research has explored the application of diffusion models for the task of layout generation such as LayoutDiffusion [39] , LDGM [14], LayoutDM [16]. These methods are based on discrete state diffusion [3] and are inspired from VQDiffusion [10] to isolate diffusion process for different layout semantics. Nevertheless, these methods only emphasize on the synthesis task. In contrast, our proposed approach with layout latent representation enables a spectrum of generative as well as search tasks. One of the methods that also relies on a compact latent representation is StructureNet [30], a hierarchical graph encoder-decoder network for 3D shape generations. We compare and contrast with a 2D version of StructureNet for layouts, and show that our proposed CLASS framework excels at both search and synthesis.

2.3. Layout Search

Layout search methods give an opportunity to discover, refer, re-use layouts and hence their source code from large existing repositories. Once a relevant layout is retrieved, further generative operations including projection, interpolation, completion, can be seamlessly conducted within the retrieve-and-adapt paradigm [37] . In layout search literature, [7] proposed an MLP-based auto-encoder to learn the search embeddings. Similarly, a convolutional auto-encoder has been proposed by [28]. GCN-CNN proposed by [29] used a graph network-based encoder to learn the semantic and geometric properties of the components while capturing their mutual relationships. LayoutGMN [35] used a graph-matching network to obtain the similarity between two layouts. We learn our search embedding from transformers within the proposed CLASS framework and show that it provides competitive or better layout retrieval performance than the prior arts.

3. Proposed Method

3.1. Overview of Method

The overall architecture of the proposed CLASS framework is depicted in Figure 2. The encoder q_ϕ learns the latent representation $\mathbf{z}$ which serves a dual purpose: i) search embedding for layout retrieval, and ii) conditioning vector for the decoder during layout generation. The encoder is trained using two decoders to learn robust and discriminative latent representations: 1. a new latent-conditioned

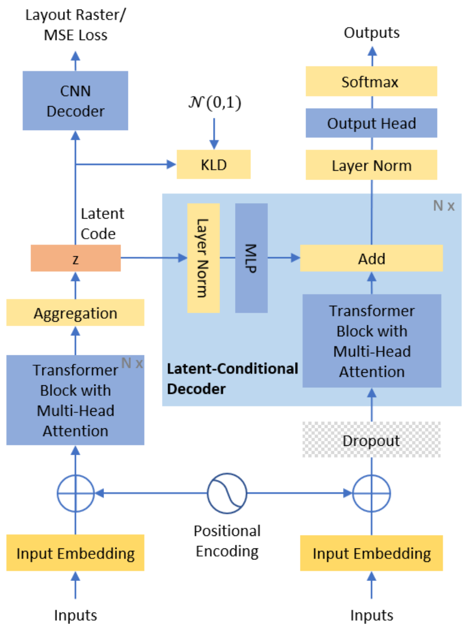


Figure 2. Proposed CLASS framework. The encoder produces a latent code that is used for both layout search and synthesis, and trained using dual CNN raster decoder, and a latent-conditioned transformer decoder.

transformer decoder f_θ and 2. an auxiliary raster decoder r_ω. The proposed latent-conditioned decoder explicitly conditions upon the latent codes while predicting the next token based on visible tokens. The auxiliary decoder injects visual information in the latent space as we show later this improves both search and synthesis qualities. The latent space is further regularized using a variational module with KL-Divergence. The following sections describe each module in detail and present our training objectives.

3.2. Layout as Sequence

A layout can be described with semantic and geometric information of its N_e elements: $\{s_i\}_{i=1}^{N_e}$ and $\{x_i, y_i, w_i, h_i\}_{i=1}^{N_e}$ where s_i indicates the element type and tuple (x, y, w, h) its upper-left coordinates, width and height. Following [11], we discretize the spatial information using 8-bin uniform quantization into categorical data. This implicitly introduces alignment constraints, an important property for design layouts such as documents and app wireframes. A layout is hence represented as a sequence $\mathbf{l} = \{\langle\text{bos}\rangle, s_1, x_1, y_1, w_1, h_1, \cdots s_N, x_N, y_N, w_N, h_N, \langle\text{eos}\rangle\}$; $\langle\text{bos}\rangle/\langle\text{eos}\rangle$ indicates the beginning/end of a sequence.

3.3. Architecture and Training Objectives

3.3.1 Encoder

Our encoder is a stack of transformer blocks [38] with multi-head attention which learns semantic and geometric properties of layout components while encoding their mutual relationships. It takes an input sequence $\mathbf{l}=\{l_i\}_{i=1}^N$ of any length N and produces a set of hidden vectors which are then aggregated to produce the latent representation:

$$\mathbf{z} = (\Sigma_i^N h_i)/N;\ \{h_i\}_{i=1}^N = q_\phi(\mathbf{l}). \tag{1}$$

We observe that the above Average Pooling operation performs the best. We explored various other aggregation techniques. A detailed ablation study is presented in the supplementary materials.

To regularize the latent space, we train the latent representation within a Variational Autoencoder (VAE) framework [21]. We parameterize the posterior distribution $q_\phi(\mathbf{z}|\mathbf{l})$ as a multivariate standard normal distribution $\mathcal{N}(0, I)$ where I is a unit diagonal matrix. We use the reparameterization trick [21] to learn the parameters μ and σ of the Gaussian distribution, *i.e.* $\mathbf{z} = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. The vectors μ and σ are outputs learned using two multilayer perceptron (MLP)-based networks on the aggregated vector within the VAE module. Overall, this helps to learn a smooth latent space that facilitates various synthesis tasks such as self-completion, interpolation, and unconditional generation while boosting search relevance.

3.3.2 Latent-Conditioned Decoder

Different from existing methods that use a naive autoregressive decoder [11] or a label sets conditioned decoder [20], we propose a novel latent conditioned transformer decoder $f_\theta(l_i|\mathbf{z}, l_{1:i-1})$ that explicitly relies upon the latent vector while predicting the next token in the sequence based on the previously visible tokens. The conditional decoder $f_\theta(l_i|\mathbf{z}, l_{1:i-1})$ is implemented with masked multihead attention (MMHA) where the information in the latent representation is infused into the feed-forward network of the decoder block. In particular, the latent conditional block is realized as follows using MMHA [38], LayerNorm [4] and residual connections.

$$\hat{u}_i = \text{LayerNorm}\left(l_i + \text{MMHA}\left(l_i, l_{i-1}, \cdots, l_0\right)\right) \tag{2}$$
$$u_i = \text{LayerNorm}\left(\hat{u}_i + \text{MLP}_{\text{FF}}(\hat{u}_i)\right) \tag{3}$$
$$v_i = \text{LayerNorm}\left(u_i + \text{MLP}_{\text{Cond}}(\text{LayerNorm}(\mathbf{z}))\right) \tag{4}$$

Here both MLP_{FF} and MLP_{Cond} are implemented as a two-layered feed-forward network with GeLU activation [13]. The final block is fed into an MLP network head that produces logit o_i over the category distribution and trained with teacher-forcing strategy against the groud-truth.

3.3.3 Auxiliary Raster Decoder

Learning a discriminative latent space is most crucial in our framework as all the search and synthesis tasks directly rely upon it. To learn rich representation, we propose to use an auxiliary decoder on the latent vector. This serves two purposes. First, it enhances the quality of search embeddings and hence improves the retrieval performance. Second, it produces a perceptually meaningful and informative latent representation that the latent conditional decoder can reliably condition upon instead of neglecting the latent vector during the generation process.

The auxiliary decoder r_ω is implemented as a CNN based decoder that outputs a raster representation of the layout from the latent vector, *i.e.* $\hat{R}^{H\times W\times C} = r_\omega(\mathbf{z})$, where H and W are the height and width of the raster, and C represents the number of component classes in the dataset. The CNN decoder is implemented as a stack of 4 strided transposed convolution layers that gradually increase the spatial resolution to match the required output dimensions.

3.3.4 Training Objectives

Our training objective comprises three losses for the latent conditional decoder, the auxiliary raster decoder, and the VAE module. The following equations describe the overall loss.

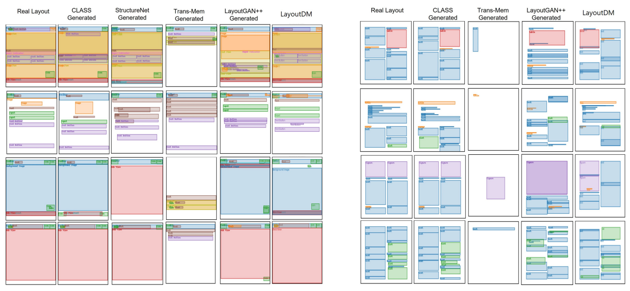
$$L_{\text{total}} = L_{\text{recon}} + \alpha L_{\text{vae}} + \beta L_{\text{raster}} \tag{5}$$
$$L_{\text{recon}} = L_{\text{CE}}(o_i, l_i) \tag{6}$$
$$L_{\text{vae}} = \text{KL}\left(q_\phi(\mathbf{z}|\mathbf{l})\,||\,\mathcal{N}(0, I)\right) \tag{7}$$
$$L_{\text{raster}} = \text{MSE}(R, \hat{R}) \tag{8}$$

Here, the reconstruction loss is the cross-entropy (CE) loss on the predicted logits o_i and groudtruth class l_i. The VAE loss is KL-Divergence loss between the posterior $q_\phi(\mathbf{z}|\mathbf{l})$ and Gaussian Normal distribution. Finally, the raster loss is implemented as mean squared error (MSE) between the predicted raster $\hat{R}$ and its groudtruth semantic map R as defined in [29]. The α and β are weighting parameters for the losses.

4. Experiments

4.1. Datasets

RICO [7] is a large collection of UX designs curated by crowd-sourcing and mining 9.3K free mobile apps. It has 25 types of components such as text, icon, image etc. with annotated Andriod view hierarchies. Following [29], we divide the dataset into 53K training, 13K test samples, and additional 50 samples as the query set to perform retrieval.
PubLayNet [40] contains 360K+ scientific document layouts crawled from the internet and includes 5 element types: text, title, figure, list, table. A set of 200 layouts are held out from validation as the query to perform retrieval. For both datasets, we exclude layout samples with over 100 elements due to GPU memory constraints.



(a) RICO (b) PubLayNet

Figure 3. Conditional generation for the proposed CLASS framework, Trans-Mem, LayoutGAN++ and LayoutDM. We encourage readers to explore the HTML visualization in the supplementary for more qualitative comparisons.

4.2. Settings

We use d=512 as the dimension of the transformer model and latent embeddings, and set the number of transformer blocks to 6, each with n_{head}=8 for MMHA. We train the network with Adam optimizer with default betas of $(0.9, 0.999)$ with an initial learning rate of $1e^{-3}$. For generation, we start from scratch with $\langle\text{bos}\rangle$ unless otherwise specified and use top-k sampling while decoding. The parameters α and β are empirically set to $1e^{-5}$ and 5 respectively.

4.3. Evaluation Metrics

4.3.1 Conditional Generation Metrics

We first measure the latent-conditioning capability of the proposed CLASS framework. To do so, we use two metrics: **1. Class IoU** which measures the average intersection over union (IoU) for each element class in the groundtruth layout over generated ones from latent vectors. **2. Edit Distance (ED)** measures the number of operations (add/remove/swap) required from the generated layout to reach the same set of component classes in the groundtruth. Note that though we reconstruct from latent space, the proposed decoder is latent-conditioned while being autoregressive, and hence CLASS is able to generate multiple plausible conditioned layouts from a single latent code.

4.3.2 Generation Metrics

We use a set of standard metrics to comprehensively evaluate the generated layouts from both perceptual and diversity perspectives following the definitions in [20].
1. Alignment measures how well-aligned the components are in the layouts. **2. Overlap** measures the unusual overlapping between components. It is observed that the existing works apply different pre-processing and experimental settings, and use different train-val splits due to unavailability of an official split. This lack of uniformity introduces a considerable degree of variability in the reported metrics in the published papers, particularly concerning alignment and overlap assessments – an issue also raised by [16]. In view of this, we advocate towards normalizing the deviations of the metrics from real layouts for alignment and overlap measures *i.e.* $|r-o|/r$, where r is the corresponding value real data and o is the observed value. This proposition aims to mitigate the inherent bias and discrepancies arising from variations in metric implementation and experimental setups.

3. Maximum Intersection over Union (maxIoU) measures the similarity between the generated layouts and real layouts which is based on IoU of the component bounding boxes. We adapt the implementation in [20] to compute MaxIoU between different sets of component classes.
4. Frechet Inception Distance (FID) provides a difference in the distribution of real and generated layouts. We follow [20] and train a network to differentiate between real and bounding box perturbed layouts, and then use it to extract layout features to compute the FID score. We provide more details on these metrics in the supplementary materials. Following the standard practice, we compute these metrics on 1000 layouts generated for all methods.

4.3.3 Retrieval Metrics

Following [29], we use **MeanIoU** between query and retrieval layouts and compute the average for top-k retrievals. We also report **Mean Edit Distance (MED)** @k indicating the edit distance between the query and retrieved layouts. We report both scores at $k = [1, 5, 10]$.

4.4. Latent Conditional Generation Analysis

We compare latent-conditioning capabilities of our CLASS framework with existing works [18, 20, 30]. LayoutGAN++ [20] generates layouts for a given set of component labels by predicting the bounding box for each component. Recently, [18] used an average-pooled vector as a latent representation but as memory input into the standard transformer decoder [38]. We dub this baseline as 'Trans-Mem' hereafter. We also compare with StructureNet [30] which uses hierarchical graph networks and relies on latent representation for 3D generative tasks. We implement StructureNet for 2D layouts for RICO dataset; PubLayNet lacks the hierarchical data required by StructureNet. We generate 1000 layouts conditioned upon layouts from the test set as groundtruth. We report the Class-IoU and ED metrics in Table 1. The columns VAE and Raster indicate whether or not the KLD loss Eq.(7) and auxiliary decoder loss Eq.(8) are used.

From Table 1, the proposed CLASS outperforms other methods on both datasets. On RICO, our CLASS frame-

Table 1. Conditional generation: Trans-Mem [18], LayoutGAN++ [20] , StructureNet [30], LayoutDM [16] and CLASS.

			RICO		PubLayNet	
Method	VAE	Raster	**Class-IoU**	**ED**	**Class-IoU**	**ED**
Trans-Mem	✓		0.1287	23.103	0.0768	11.05
LayoutGAN++			0.3114	0.0	0.2623	0.0
StructureNet	✓		0.3046	8.04	-	-
LayoutDM			0.4805	0.0	0.4547	0.0
CLASS			0.4551	4.25	0.5925	**0.298**
CLASS	✓		0.4896	4.03	0.6269	0.589
CLASS		✓	**0.5389**	3.91	**0.7161**	0.360
CLASS	✓	✓	0.5259	**3.86**	0.6649	0.532

work achieves a Class-IoU of 0.455, CLASS with VAE module achieves 0.489, and CLASS with raster achieves the best Class-IoU of 0.538. The observations show that the conditional generation indeed benefits from the proposed VAE and the auxiliary decoder modules in the framework. This holds for the PubLayNet dataset as well. Next, we see that only using the raster decoder performs better than CLASS with both VAE and raster. The reason behind this observation can be explained by the tension between reconstruction and KL-divergence losses [2]. Note that LayoutGAN++ and LayoutDM have null values for ED metrics as it directly uses the groudtruth class labels during generation. Overall, our proposed CLASS achieves 0.538 IoU and 3.86 ED on RICO, and the best scores of 0.716 IoU and 0.29 ED on PubLayNet outperforming existing the graph-based [30], GAN-based [20] and transformer-based [18] architectures.

Figure 3 shows conditional generation for various methods. We noticed that Trans-Mem often generates layouts with limited variations and are barely conditioned upon the given layout. LayoutGAN++ generates layouts with the required types of components but accompanied by undesired overlapping and misalignment. StructureNet could not handle the complex layouts well. Compared to these, our method conditions upon the latent embedding of the given layout and generates different while perceptually plausible layout variations, demonstrating its effectiveness. LayoutDM sets a strong baseline but CLASS has provide more aligned and less overlapped layouts (as also observed in Tab. 2). We provide an easy visualization for more qualitative results to give a sense of how different methods compare with one another - please refer to supplementary for more conditional and multiple generations from the same latent embedding.

4.5. Generation Quality Analysis

We compare our methods with several baselines including the non-sequential methods [25, 30], the transformer-based methods [1,11,17,18,20,22], and the recent diffusion-based models [14, 16, 39]. Table 2 summarizes results for the set of metrics in presented in Sec 4.3.2. For a fair comparison, we compare both unconditional and conditional generation, and report them on the top and bottom of the table respectively.

We highlight that the goal here is not to achieve the lowest/highest value on each metric but to obtain balanced scores for all metrics which imply that the generations are both good from both perceptual and diversity perspectives. For both unconditional and conditional generation, our proposed method is frequently ranked on the top 2 while obtaining a balanced set of scores for all the metrics While comparing individually, some of the metric may perform well than our approach. For example, under conditional generation, StructureNet achieves the highest score of maxIoU 0.590 while our CLASS-raster and LDGM achieve slightly lower maxIoUs of 0.573 and 0.580. However, we note that StructureNet comparatively obtains larger misalignments and overlaps errors, and has less diversity in generation.

Further, our method performs the best according to negative log-likelihood (NLL) and we note that NLL improves with the VAE module and auxiliary decoder. We encourage readers to refer to qualitative visualisation results in the supplementary. In a nutshell, our method achieves a good balance between perceptual quality of the layouts (alignment and overlap) while maintaining the diversity in the generation with the lower FID scores. Most importantly, our approach stand out with seamlessly integration of joint search and generation, setting it apart from all other generative methods in the table.

4.6. Generative Sub-tasks

Besides generation shown in Fig. 3, we further explore advantages of the proposed architecture and show our CLASS framework is capable of self-completion, projection of layouts, interpolation, and unconditional generation as represented in Fig. 1. In the following, we briefly discuss them and present respective qualitative results.

Self-Completion Given a partial layout with only a few initial components, our method can complete the layout by decoding a latent vector sampled from Normal Gaussian distribution. Fig. 4 shows examples of self-completions. This can be a useful application where users want to complete their initial designs into plausible layouts. CLASS indeed provides multiple versions of completions from a single partial design allowing the user to choose from the generated corpus (See Supplementary for such examples).

Projection As CLASS operates on latent conditioning, it can project partial layouts to a reference layout by taking the reference latent embedding as conditioning. Fig. 5 shows sample projections where the initial partial designs are completed based on another reference. This can serve users to complete the partial initial layout designs with inspiration from the reference layouts they desire/require.

* and ‡ indicate values referenced from [39] and the corresponding papers respectively.

Table 2. Quantitative evaluation of generated layouts for RICO and PubLayNet datasets using Alignment, Overlap, maxIoU, FID. We include the negative log-likelihood (NLL) loss on the test set for applicable methods. For reference, the FID score for real layouts is computed between the train and test set. The best values are bold and the runner-ups are underlined.

	Method	VAE	Raster	**RICO**					**PubLayNet**				
				Align ment↓	**Over lap↓**	**max IoU↑**	**FID ↓**	**NLL ↓**	**Align ment↓**	**Over lap↓**	**max IoU↑**	**FID ↓**	**NLL ↓**
Unconditional	LayoutTransformer [11]			**0.075**	0.087	0.549	28.05	1.12	**0.023**	0.182	**0.499**	18.75	1.32
	VTN* [1]			4.129	0.204	0.336	88.12	-	8.409	70.290	0.312	105.91	-
	Coarse2Fine* [18]			0.376	0.451	0.360	46.48	-	9.045	44.806	0.361	50.85	-
	LayoutFormer++* [17]			0.452	0.172	$\underline{0.634}$	20.2	-	0.273	0.677	0.401	47.08	-
	Diffusion-LM* [27]			0.978	0.354	**0.662**	11.45	-	2.455	3.032	0.439	$\underline{11.9}$	-
	LayoutDM [16]			0.828	$\underline{0.056}$	0.3852	12.03		7.893	41.22	0.470	13.73	
	LDGM‡ [14]			0.385	0.130	0.620	26.06		30.25	2.828	0.460	25.94	
	LayoutDiffusion* [39]			0.258	0.077	0.620	**2.49**	-	1.955	**0.032**	0.417	**8.63**	-
	CLASS (Ours)			$\underline{0.085}$	**0.010**	0.568	26.19	1.12	$\underline{0.230}$	$\underline{0.166}$	$\underline{0.493}$	18.53	1.29
Conditional	LayoutGAN++ [20]			0.288	0.235	0.371	4.05	-	10.254	106.743	0.388	16.64	-
	Trans.-Mem [18]	✓		0.090	0.121	0.527	47.26	1.47	11.451	529.864	0.211	125.84	1.64
	StructureNet [30]	✓		0.169	0.206	**0.590**	64.37	-	-	-	-	-	-
	NDN-none* [24]			5.022	0.180	0.350	13.76	-	14.909	53.839	0.310	35.67	-
	BLT* [22]			0.613	1.109	0.216	25.63	-	0.636	62.226	0.140	38.68	-
	LayoutFormer++* [17]			0.333	0.152	0.377	2.48	-	**0.136**	$\underline{1.903}$	0.333	10.15	-
	Diffusion-LM* [27]			1.14	0.232	0.324	6.53	-	1.091	7.387	0.316	7.4	-
	LDGM‡ [14]			0.500	0.121	$\underline{0.580}$	16.64		17.750	2.259	0.440	20.69	
	LayoutDiffusion* [39]			0.333	0.054	0.345	**1.56**	-	$\underline{0.318}$	**0.613**	0.343	3.73	-
	LayoutDM [16]			0.262	0.032	0.461	12.71		3.67	56.55	0.490	4.56	
	CLASS			0.037	0.047	0.571	3.80	0.94	0.948	8.234	0.604	3.89	0.39
	CLASS	✓		$\underline{0.026}$	**0.004**	0.566	2.57	0.85	1.455	5.541	0.640	$\underline{3.65}$	0.37
	CLASS		✓	**0.014**	0.018	0.573	2.52	**0.78**	1.751	5.548	**0.691**	3.87	**0.29**
	CLASS	✓	✓	0.035	$\underline{0.016}$	0.565	$\underline{2.28}$	$\underline{0.83}$	1.441	4.435	$\underline{0.654}$	**3.56**	$\underline{0.35}$

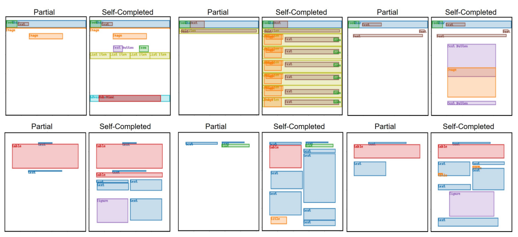


Figure 4. Self-Completion. Given a partial layout (left), CLASS self-completes them into visually plausible layouts (right). Top row is for RICO and the bottom for PubLayNet.

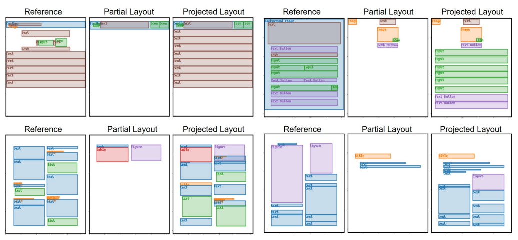


Figure 5. Projection. For each example, given initial partial layouts (middle), our CLASS framework completes them into plausible projected layouts (right) based on the inspiration/style of the reference (left). Top row is for RICO and the bottom for PubLayNet.

Unconditional Generation Our conditional CLASS network is also capable of unconditional generations without any reference. We sample a latent embedding from a normal Gaussian distribution and decode the layout using the proposed latent conditional decoder, thanks to the VAE module in the proposed CLASS framework. Fig. 6 shows samples of unconditionally generated visually layouts.

Interpolation CLASS supports interpolation operation between the latent representations and is capable of generating new plausible layouts based on intermediate latent codes. Fig.7 shows layouts generated by a linear walk in latent space between two layouts. Such an application can enables users to blend two layouts. For example, a user may want to combine the top retrieved layouts obtained from a layout search of interest.

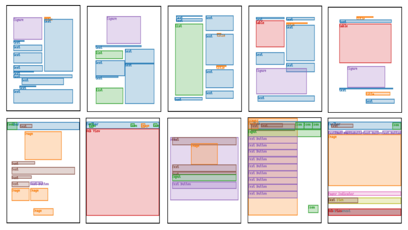


Figure 6. Unconditional layouts generations by the proposed method by sampling the latent vectors from Gaussian distribution.

4.7. Evaluations on Layout Search

We use CLASS embedding for the layout retrieval task and compare our method against other baselines AutoEncoder (AE) [7], convolutional autoencoder (CAE) [28], StructureNet [30], (GCN-CNN) [29] as shown in Table 3 . Note that AE, CAE, GCN-CNN are search-tailored methods and do not support generative tasks. Our method

Table 3. Layout search performance comparison on RICO and PubLayNet datasets using MIoU and Mean ED at $k = [1, 5, 10]$. The best values are bold and the runner-ups are underlined. Methods in the upper part of the table are non-generative methods tailored for search, and the lower part shows methods capable of both search and synthesis.

	Method	VAE	Raster	RICO						PubLayNet					
				MIoU (%) ↑			MED ↓			MIoU (%) ↑			MED ↓		
	k			1	5	10	1	5	10	1	5	10	1	5	10
Non-generative	AE [7]			43.0	34.7	28.9	19.44	18.68	19.32	**31.5**	<u>29.5</u>	<u>28.3</u>	2.84	3.08	3.25
	CAE [28]			<u>59.5</u>	<u>47.1</u>	<u>43.9</u>	8.76	9.56	11.2	<u>31.2</u>	29.2	28.1	3.03	3.2	3.39
	GCN-CNN [29]			**60.0**	**51.6**	**48.3**	7.98	8.91	10.06	31.1	**29.9**	**29.2**	<u>2.10</u>	<u>2.33</u>	<u>2.50</u>
Generative	StructureNet [30] †	✓		51.9	40.3	38.5	4.5	5.8	6.3	-	-		-	-	-
	Trans-Mem [18]	✓		33.3	26.5	24.5	5.96	<u>7.16</u>	**7.47**	20.1	19.4	18.8	**1.53**	**1.65**	**1.77**
	LayoutGMN [35]			44.6	38.4	34.2	-	-	-	-	-	-	-	-	-
	CLASS			54.4	44.3	41.3	**5.54**	**6.91**	7.69	27.3	25.8	25.0	2.44	2.56	2.73
	CLASS	✓		54.1	40.7	37.9	7.68	8.74	9.70	27.0	25.6	24.9	2.43	2.58	2.70
	CLASS		✓	58.5	45.0	42.6	<u>7.52</u>	10.15	10.71	28.8	27.4	26.6	2.41	2.69	2.89
	CLASS	✓	✓	58.2	44.9	42.0	7.54	8.66	<u>9.27</u>	28.4	27.0	26.1	2.31	2.50	2.59

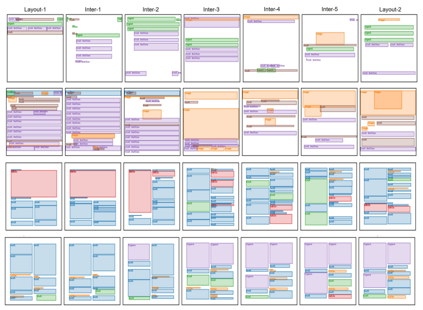


Figure 7. Layout interpolation. Intermediate layouts slowly resemble to Layout2 from left to right and vice-versa.

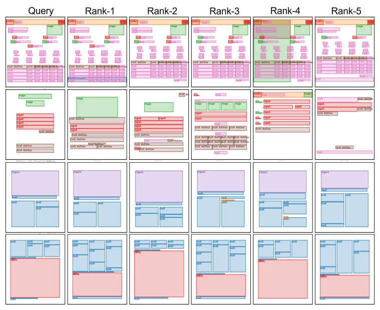


Figure 8. Layout retrievals using the proposed CLASS. Layouts in the first column are query layouts followed by top-5 retrievals. The first two rows show the results for RICO and the bottom two for PubLayNet.

achieves competitive performance on MIoU metrics. For example, CLASS-Raster achieves 58.5% MIoU@1 which is $\sim$ 1.5% lower than the best GCN-CNN method. Our method achieves strong performances on both datasets and even outperforms the state-of-the-art method on RICO dataset obtaining a MED@1 score of 5.54. The AE, CAE, and GCN-CNN are trained based on spatial reconstruction and hence they are better at MIoU metrics but may potentially ignore small components leading to poorer MED values. On PubLayNet, CLASS performs on-par with SOTA with runner-up MED scores. Compared to the transformer-based method [18], our method is better in terms of MIou while achieving on par or slightly lower MED values for Publaynet. Upon visualization, we noted that Trans-Mem retrieves layouts with a similar set of components but often fails to take overall spatial arrangements into accounts. In a nutshell, our method achieves a good balance between MIoU and MED retrieval metrics and performs the best among the generative models. Fig. 8 shows retrieval results obtained using the proposed method. We provide additional results and analysis in the supplementary materials including qualitative retrieval comparisons.

5. Conclusion

We proposed a novel unified framework for search and synthesis of design layouts under an elegant architecture. We introduced a new latent-conditioned decoder that conditions upon the latent vectors during layout synthesis. We learned discriminative latent representation with an encoder trained with a dual pair of a latent conditional decoder and a raster decoder. We demonstrated our model is capable of generating high-quality design layouts with a good balance between perceptual quality and diversity while being capable of various practical sub-tasks such as self-completion, projection and interpolation. We established competitive results for layout retrieval against the state-of-the-art. Overall, we validated the effectiveness of the proposed CLASS framework for both search and synthesis tasks on the benchmark layout datasets. We believe our model can be integrated into AI-assistive design tools for efficient and personalized design workflow.

Acknowledgment

This work was supported in part by Adobe and in part by DECaDE under EPSRC grant EP/T022485/1.

†StructureNet limits the number of children at each hierarchy, making the evaluation possible only on 43 out of 50. This leads to lower MED scores as it omits complex layouts with large number of components.

References

[1] Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13642–13652, 2021. 1, 2, 6, 7

[2] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8:199440–199448, 2020. 6

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 3

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[5] Letizia Bollini. Beautiful interfaces. from user experience to user interface design. *The Design Journal*, 20(sup1):S89–S101, 2017. 1

[6] Niranjan Damera-Venkata, José Bento, and Eamonn O'Brien-Strain. Probabilistic document model for automated document composition. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 3–12, 2011. 2

[7] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 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, 2017. 2, 3, 4, 7, 8

[8] Joe Geigel and Alexander CP Loui. Automatic page layout using genetic algorithms for electronic albuming. In *Internet Imaging II*, volume 4311, pages 79–90. SPIE, 2000. 2

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[10] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 3

[11] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021. 1, 2, 3, 4, 6, 7

[12] Steven J Harrington, J Fernando Naveda, Rhys Price Jones, Paul Roetling, and Nishant Thakkar. Aesthetic measures for automated document layout. In *Proceedings of the 2004 ACM symposium on Document engineering*, pages 109–111, 2004. 2

[13] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[14] Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1942–1951, 2023. 1, 3, 6, 7

[15] Nathan Hurst, Wilmot Li, and Kim Marriott. Review of automatic document formatting. In *Proceedings of the 9th ACM symposium on Document engineering*, pages 99–108, 2009. 2

[16] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023. 1, 3, 5, 6, 7

[17] Zhaoyun Jiang, Jiaqi Guo, Shizhao Sun, Huayu Deng, Zhongkai Wu, Vuksan Mijovic, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. Layoutformer++: Conditional graphic layout generation via constraint serialization and decoding space restriction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18403–18412, 2023. 3, 6, 7

[18] Zhaoyun Jiang, Shizhao Sun, Jihua Zhu, Jian-Guang Lou, and Dongmei Zhang. Coarse-to-fine generative modeling for graphic layouts. 2022. 1, 2, 3, 5, 6, 7, 8

[19] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9895–9904, 2019. 1, 2

[20] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 88–96, 2021. 1, 2, 4, 5, 6, 7

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 4

[22] Xiang Kong, Lu Jiang, Huiwen Chang, Han Zhang, Yuan Hao, Haifeng Gong, and Irfan Essa. Blt: Bidirectional layout transformer for controllable layout generation. *arXiv preprint arXiv:2112.05112*, 2021. 6, 7

[23] Ranjitha Kumar, Jerry O Talton, Salman Ahmad, and Scott R Klemmer. Bricolage: example-based retargeting for web design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2197–2206, 2011. 2

[24] Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. Neural design network: Graphic layout generation with constraints. In *European Conference on Computer Vision*, pages 491–506. Springer, 2020. 2, 7

[25] Hsin-Ying Lee, Lu Jiang, Irfan Essa, Phuong B Le, Haifeng Gong, Ming-Hsuan Yang, and Weilong Yang. Neural design network: Graphic layout generation with constraints. In *European Conference on Computer Vision*, pages 491–506. Springer, 2020. 6

[26] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767*, 2019. 1, 2

[27] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves control-

lable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022. 7

[28] Thomas F. Liu, Mark Craft, Jason Situ, Ersin Yumer, Radomir Mech, and Ranjitha Kumar. Learning design semantics for mobile apps. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pages 569–579, New York, NY, USA, 2018. ACM. 1, 2, 3, 7, 8

[29] Dipu Manandhar, Dan Ruta, and John Collomosse. Learning structural similarity of user interface layouts using graph networks. In *European Conference on Computer Vision*, pages 730–746. Springer, 2020. 1, 2, 3, 4, 5, 7, 8

[30] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 3, 5, 6, 7, 8

[31] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Designscape: Design with interactive layout suggestions. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 1221–1224, 2015. 2

[32] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Learning layouts for single-page graphic designs. *IEEE transactions on visualization and computer graphics*, 20(8):1200–1213, 2014. 2

[33] Xufang Pang, Ying Cao, Rynson WH Lau, and Antoni B Chan. Directing user attention via visual flow on web designs. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016. 2

[34] Akshay Gadi Patil, Omri Ben-Eliezer, Or Perel, and Hadar Averbuch-Elor. Read: Recursive autoencoders for document layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 544–545, 2020. 2

[35] Akshay Gadi Patil, Manyi Li, Matthew Fisher, Manolis Savva, and Hao Zhang. Layoutgmn: Neural graph matching for structural layout similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11048–11057, 2021. 1, 2, 3, 8

[36] Lumpapun Punchoojit and Nuttanont Hongwarittorrn. Usability studies on mobile user interface design patterns: a systematic literature review. *Advances in Human-Computer Interaction*, 2017, 2017. 1

[37] Chunyao Qian, Shizhao Sun, Weiwei Cui, Jian-Guang Lou, Haidong Zhang, and Dongmei Zhang. Retrieve-then-adapt: Example-based automatic generation for proportion-related infographics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):443–452, 2020. 1, 3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 3, 4, 5

[39] Junyi Zhang, Jiaqi Guo, Shizhao Sun, Jian-Guang Lou, and Dongmei Zhang. LayoutDiffusion: Improving Graphic Layout Generation by Discrete Diffusion Probabilistic Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 3, 6, 7

[40] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019. 2, 4