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

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

1. Details on Evaluation Metrics

Here we present details on evaluation metrics used for generation quality: Alignment, Overlap, FID, and MaxIoU. We follow [3] for definitions of all metrics with modifications when needed.

Alignment Alignment is crucial for the perceptual quality of layouts. Alignment metrics show how well the components are aligned with one another. Components adjacencies are defined with 6 types of alignments: Left, X-Center, Right, Top, Y-Center, and Bottom. The alignment loss is then defined using

$$L_{\text{ALIGN}} = \frac{1}{N'} \sum_{i=1}^{N'} \min \begin{pmatrix} g(\Delta x_i^L), & g(\Delta x_i^C), & g(\Delta x_i^R) \\ g(\Delta y_i^T), & g(\Delta y_i^C), & g(\Delta y_i^B) \end{pmatrix} \tag{1}$$

where $g(x) = \log(1 - x)$ and $\Delta x_i^*(* = L, C, R)$ is given by

$$\Delta x_i^* = \min_{\forall j \neq i} |x_i^* - x_j^*| \tag{2}$$

Similarly, $\Delta y_i^*(* = T, C, B)$ are computed.

Overlap The overlap value gives indicate a degree of overlap between component pairs given by

$$L_{\text{OVERLAP}} = \frac{1}{N'} \sum_{i=1}^{N'} \sum_{\forall j \neq i} \frac{a_i \cap a_j}{a_i} \tag{3}$$

where a_i indicates the area of the component and $a_i \cap a_j$ gives the overlapping area.

FID. We follow [3] and train a network to extract layout features in order to compute the FID score. In particular, a transformer-based encoder-decoder network is trained similarly to [3] to differentiate between real and fake layouts. A reconstruction loss is also used along with the GAN loss for training. The fake layouts are generated by adding Gaussian noise on bounding boxes. Thus the trained network is used to extract features for real and generated layouts for different methods to obtain the FID score for the respective methods.

MaxIoU. We used maxIoU implementation by [3] but modify it such that IoU between two layouts with different sets of component classes are be computed.

2. Aggregation Techniques

Our CLASS framework aggregates the tokens from the encoder to form a single latent representation for a layout. We represented Average Pooling in the main paper as our aggregator. Besides this, we explored various other techniques of aggregations to obtain latent representations as described in the following. In this section, we study these aggregators and their impact on conditional generations and retrieval performance. In the following, we explain these methods in detail.

1. **AveragePool**. This method aggregates all the N token representation from a sequence $\mathbf{l} = \{\langle\text{bos}\rangle\ \mathrm{l}_1, l_2, \cdot, l_{N-1},\ l_N, \langle\text{eos}\rangle\ \}$ where N represents the maximum length of the sequence in the dataset.
$\mathbf{z} = \frac{1}{N} \sum_i \mathbf{h}_i$

2. **AveragePool-Mask**: We use a special padding token $\langle\text{pad}\rangle$ to match the maximum sequence length in the dataset and form batched sequence. In this method, we average pool all the tokens but mask out the representations for $\langle\text{pad}\rangle$.

3. **MLP-Aggregation**: We concatenate all the hidden vectors and then use a stack of linear layers to obtain a representation of model dimension $\mathbf{z} = \text{MLP}([\mathbf{h}_i])$

4. **FirstOut**: In the transformer network, all the tokens attend one another in the encoder, we use the first token as the latent representation of the layout. $\mathbf{z} = \mathbf{h}_0$

5. **Attention**: We use a $\tanh$ attention to compute the attention values att_i for each token and perform weighted averaging. $\mathbf{z} = \frac{1}{N} \sum_i att_i \mathbf{h}_i$.

6. **Latent-Token**: We add an additional token in the input as a latent-token and use the output corresponding to this token as latent representation. This is similar to class token used in vision transformers [1].

Table 1 compares various aggregators in terms of latent-conditioning capability and retrieval. We observe that AveragePool method performs the best with a Class-IoU of 0.455 and ED distance of 4.2. AveragePool-Mask closely follows this with slightly lower performance. This potentially means that all the tokens in the sequence including $\langle$pad$\rangle$ have information about the layouts as they attend to one another in the attention module.

In terms of retrieval, AveragePool achieves the best performance with MIoU@1 of 54.5. We note that AveragePool-Mask obtains the lowest values for ED@k as this method only encodes the representation for existing components in the layouts. Similarly, the attention method achieves the second highest in terms of ED@k as it sets the attention values for $\langle$pad$\rangle$ tokens to zeros only encoding the components in the layouts. However, this method achieves lower performance in terms of MIoU. Overall, the AveragePool method achieves a good balance of mIoU@k and ED@k retrieval performances while providing the best conditional generation results.

Table 1. Ablation study on the choice of aggregation techniques

	Generation		Retrieval					
Method	IoU ↑	ED↓	MIoU (%) ↑			MED ↓		
k			1	5	10	1	5	10
Attention	0.255	18.8	45.6	36.0	34.8	4.2	5.8	6.3
MLP	0.207	51.7	45.5	32.3	29.2	7.8	9.3	9.8
FirstOut	0.205	53.6	39.6	32.4	31.4	11.2	13.5	13.6
Latent-Token	0.167	16.1	42.4	33.0	31.7	9.8	11.9	12.5
AveragePool-Mask	0.366	10.9	49.3	41.4	39.4	**3.5**	**4.6**	**5.2**
AveragePool	**0.455**	**4.2**	**54.5**	**44.3**	41.3	5.5	6.91	7.6

3. Evaluation on Magazine dataset [5]

We attempt to evaluate transferability of the proposed method by deploying Publaynet-trained CLASS model on Magazine layouts without any training or finetuning on Magazine dataset. This is a challenging setting which previous works often do not explore. To this end, we map magazine component categories to the most relevant publaynet's categories , for example, figure → image. Fig 1 shows sample conditional generation results. The generated layouts are well-aligned, visually plausible and have styled inspired from PublayNet, expected as it is a zero-shot transfer. We obtained an class-IoU of 0.2225 and ED of 0.6454 on conditional generation. Table 3 presents generation quality metrics and shows that generated layouts are more aligned and less overlapped compared to the real ones indicating that PublayNet are better aligned with lesser overlap than Magazine. Overall, we achieve decent generation results on Magazine dataset. This could be further improved with finetuning or a new research direction on cross-domain/domain adaptation techniques for layouts.

Table 2. **Retrieval performance using different representations, aggregated vector, learned mean vector of the VAE module, and sampled vector from the VAE on RICO dataset.**

Method	Embedding	MIoU (%)			MED		
k		1	5	10	1	5	10
CLASS-VAE	Agg	54.1	40.7	37.9	7.6	8.7	9.7
CLASS-VAE	Mean	54.4	41.3	38.8	8.8	9.2	9.8
CLASS-VAE	Sampled	41.7	30.1	26.5	11.1	11.2	13.0
CLASS-VAE-Raster	Agg	58.2	44.9	42.0	7.5	8.6	9.2
CLASS-VAE-Raster	Mean	54.6	45.0	41.6	8.4	9.0	9.8
CLASS-VAE-Raster	Sampled	51.6	39.2	36.5	8.0	10.1	11.1

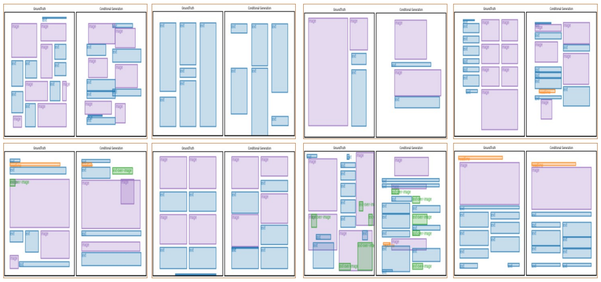

Figure 1. Conditional generation on Magazine dataset.

Table 3. **Magazine Dataset**. Absolute values reported for alignment and overlap for meaningful comparison against real layouts.

	Alignment↓	**Overlap↓**	**maxIoU↑**	**FID↓**
Real	0.0063	0.2752	1	-
CLASS	0.0041	0.0914	0.2900	40.87

4. More Qualitative Results

4.1. Visualization of Conditional Generation.

We provide easy visualization and compare different methods on conditional generations. Please unzip the folder and open 'Conditional_Visualizations/index.html'.

The visualization further justifies the quantitative scores obtained by different methods presented in the main paper. Trans-Mem often generates unrelated layouts of limited variety. StructureNet seems to handle simple layouts but struggles with complex layouts with a large number of components. LayoutGAN++ performs well but has unwanted overlapping and misalignments. In general, we observe that CLASS generates well aligned, diverse and plausible layouts compared to other methods.

4.2. Multiple Generations Using the Same Condition

We demonstrated in the main paper that our CLASS generates high-quality layout based on reference layout. The proposed CLASS is capable of multiple generations based on the same reference/latent-condition. Figure 2 and Figure 3 show examples of 5 versions of conditional generation on RICO and PubLayNet datasets respectively. CLASS generates a corpus of conditioned layouts allowing the users to choose one/multiple based on their necessity.

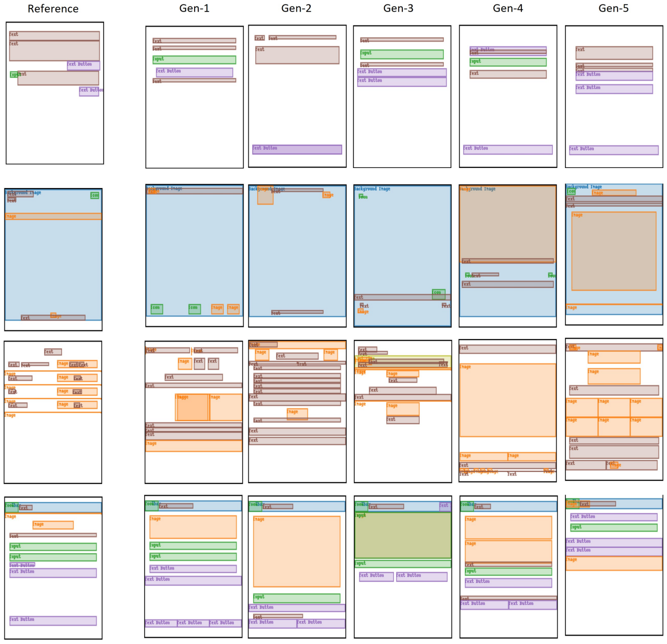

Figure 2. CLASS generates multiple variants of plausible layouts conditioned upon a reference layout. Samples from RICO dataset.

4.3. Interpolation Results

We provide more interpolation results and compare with StructureNet [4] in Fig. 4

4.4. More Qualitative Retrieval Result

In Figure 5 and Figure 6, we present layout retrieval results for RICO and PubLayNet dataset and compare with the state-of-the-art GCN-CNN method. Figure 7 compares the retrievals obtained by Trans-Mem [2] and the proposed CLASS method. Trans-Mem method tends to retrieve layouts with a similar set of components but often lacks visual similarity which indicates that the method does not take the spatial arrangements of components into account. In contrast, our method achieves better retrieval results with similar components and their spatial arrangements. We attribute this to our dual-decoder architecture of CLASS that enables us to obtain discriminative search embeddings.

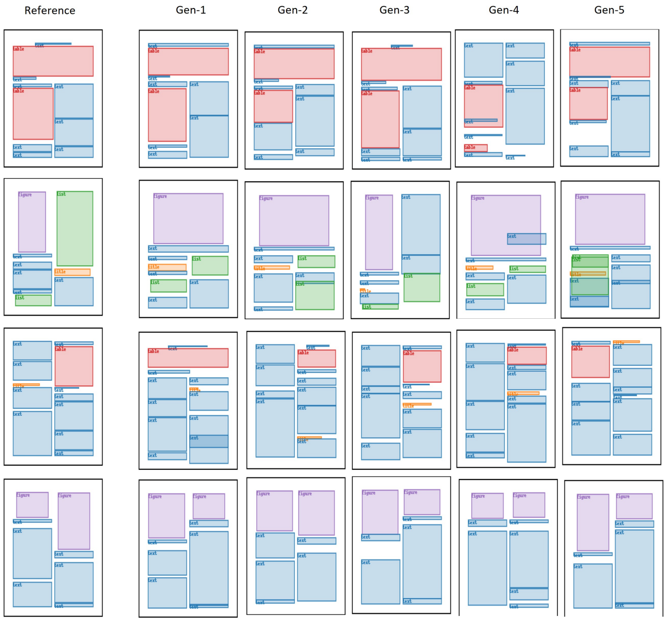

Figure 3. CLASS generates multiple variants of plausible layouts conditioned upon a reference layout. Samples from PubLayNet dataset.

5. Additional Experiments

5.1. Training CLASS in Mixed Mode

We trained a CLASS architecture in a mixed mode which enables the model to operate in both conditional mode and unconditional mode during inference. In order to train the network under this setting, we randomly set roughly half of the latent vectors in a training batch to zeros and let them train only with auto-regression. The remaining half having latent information, we train them with additional KLD/raster losses. Table 4 shows generation quality metrics obtained for the mixed-mode training and compare against separately trained conditional and unconditional networks.

For the RICO dataset, the mixed-trained model achieves similar Alignment scores compared to the separately trained model under unconditional evaluation, but performs inferior under conditional settings. For Overlap, separately trained models perform superior to the mixed model under both evaluation modes. We obtain no significant difference for the maxIoU metric. Finally, we observe that the mixed model under-performs the separately trained conditional model. Overall, separately trained networks performs better than network trained in the mixed mode for the RICO dataset.

For PubLayNet dataset, we observe that the mixed trained model achieves better performance than separately trained model for Alignment and Overlap scores under most

Table 4. Comparison various training modes for CLASS: unconditional, conditional, and mixed using Alignment, Overlap, maxIoU, FID.

Train Mode	Eval Mode	VAE	Raster	RICO				PubLayNet			
				Align ment↓	**Over lap↓**	**max IoU↑**	**FID ↓**	**Align ment↓**	**Over lap↓**	**max IoU↑**	**FID ↓**
Unconditional	Unconditional			0.085	0.010	0.5688	26.19	0.230	0.166	0.4932	18.53
Conditional	Conditional			0.037	0.047	0.5714	3.8	0.948	8.234	0.6049	3.89
Conditional	Conditional	✓		0.026	0.004	0.5667	2.57	1.455	5.541	0.6406	3.65
Conditional	Conditional		✓	0.014	0.018	0.5739	2.52	1.751	5.548	0.6910	3.87
Conditional	Conditional	✓	✓	0.035	0.016	0.5655	2.28	1.441	4.435	0.6547	3.56
Mixed	Conditional			0.079	0.021	0.5773	11.23	0.174	0.013	0.594	10.28
Mixed	Unconditional			0.080	0.034	0.5521	27.07	0.310	0.025	0.4965	17.25
Mixed	Conditional	✓		0.080	0.091	0.6079	12.24	1.793	1.136	0.5712	4.76
Mixed	Unconditional	✓		0.075	0.117	0.5517	24.74	0.169	0.006	0.4834	19.17
Mixed	Conditional		✓	0.064	0.047	0.5589	8.36	1.379	9.839	0.587	4.11
Mixed	Unconditional		✓	0.078	0.112	0.4924	30.24	0.458	0.229	0.5334	14.97
Mixed	Conditional	✓	✓	0.071	0.069	0.5953	9.16	0.646	3.222	0.5801	4.27
Mixed	Unconditional	✓	✓	0.080	0.091	0.5657	19.02	0.479	0.169	0.4990	23.99

of the settings. This is a different observation compared to the results obtained for the RICO dataset. The introduction of random dropping of latent vectors in mixed training aids in the regularisation of the network during training. The benefit of mixed mode training is observed on PubLayNet which may indicate it indeed regularises the network potentially avoiding overfitting issues due to much simpler layouts in PubLayNet than RICO layouts. We observe a slight compromise on FID with mixed mode training. Overall, mixed mode training offers the benefit of having a single network capable of inference under both unconditional and conditional modes. For RICO, mixed training compromises this capability with slightly inferior quality metrics whereas for the PubLayNet dataset, the mixed training seems to benefit Alignment and Overlap properties with a minimal compromise on FID scores.

5.2. Retrieval using representation from different layers of CLASS

In the proposed framework, the layout embeddings can be obtained from various layers: 1. Aggregated outputs of the token (Agg), the mean vector output from the VAE module *i.e.* $\mathbf{z} = \mu$, and 3. sampled vector *i.e.* $\mathbf{z} = \mu + \sigma \cdot \epsilon$ as described in the Section Encoder in the main paper. The retrieval performances of the CLASS-VAE and CLASS-VAE-Raster on the RICO dataset are presented in Table 2. The aggregated representations perform better in most cases, *e.g.* 58.2% MIoU@1 and 7.5 ED@1 on CLASS-VAE-Raster. The mean vector representation also achieves competitive or slightly lower performance. The sampled representations achieve the lowest among the three embeddings as it adds Gaussian noise into the representation. Overall, the use of aggregated representation as a search embedding performs the best.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

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

[3] 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

[4] 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] Ying Cao Xinru Zheng, Xiaotian Qiao and Rynson W.H. Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2019)*, 38, 2019. 2

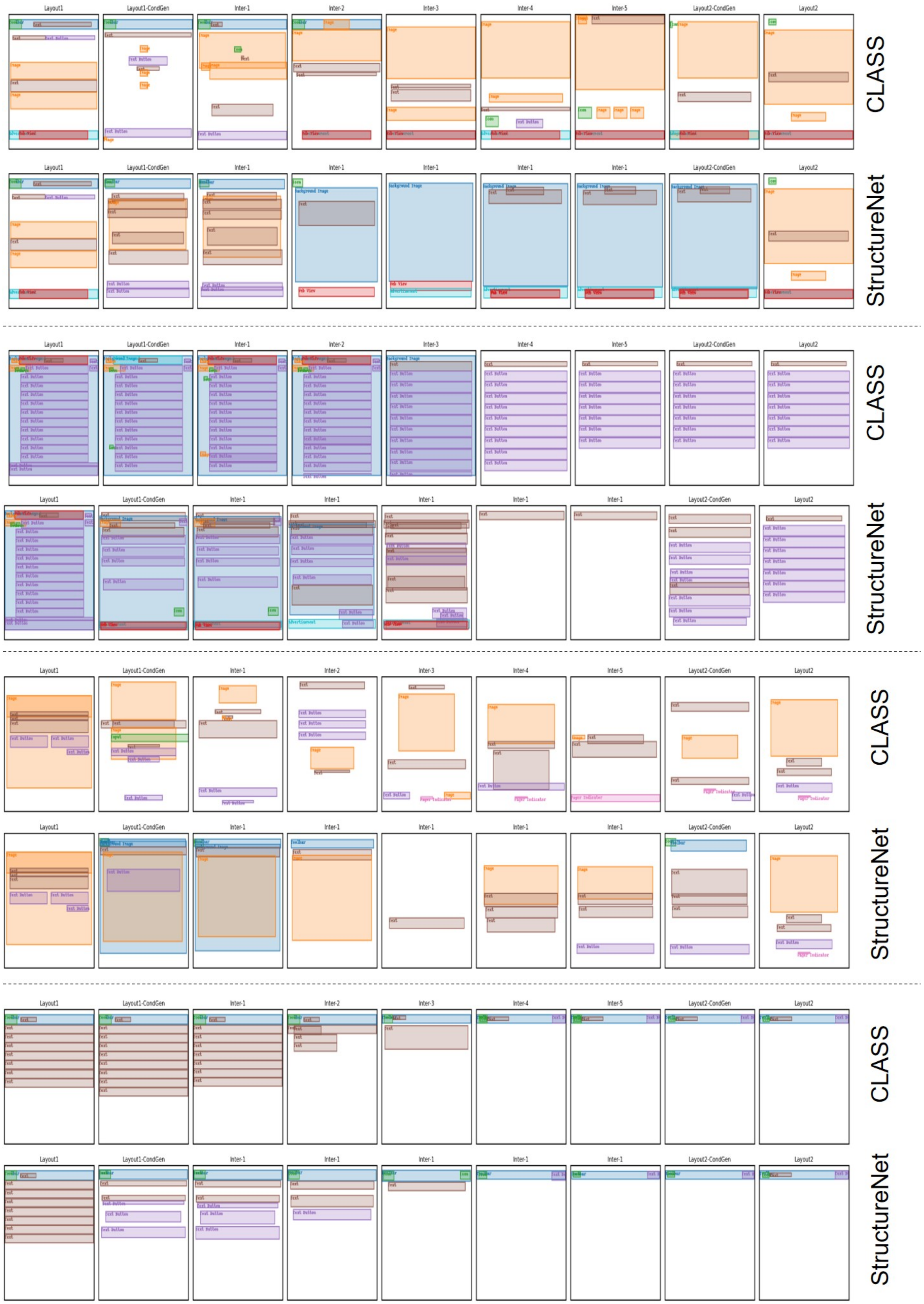

Figure 4. Interpolation between two layouts. The extreme layouts represent two validation samples followed by their conditional generation from the latent space and then intermediate interpolated layouts.

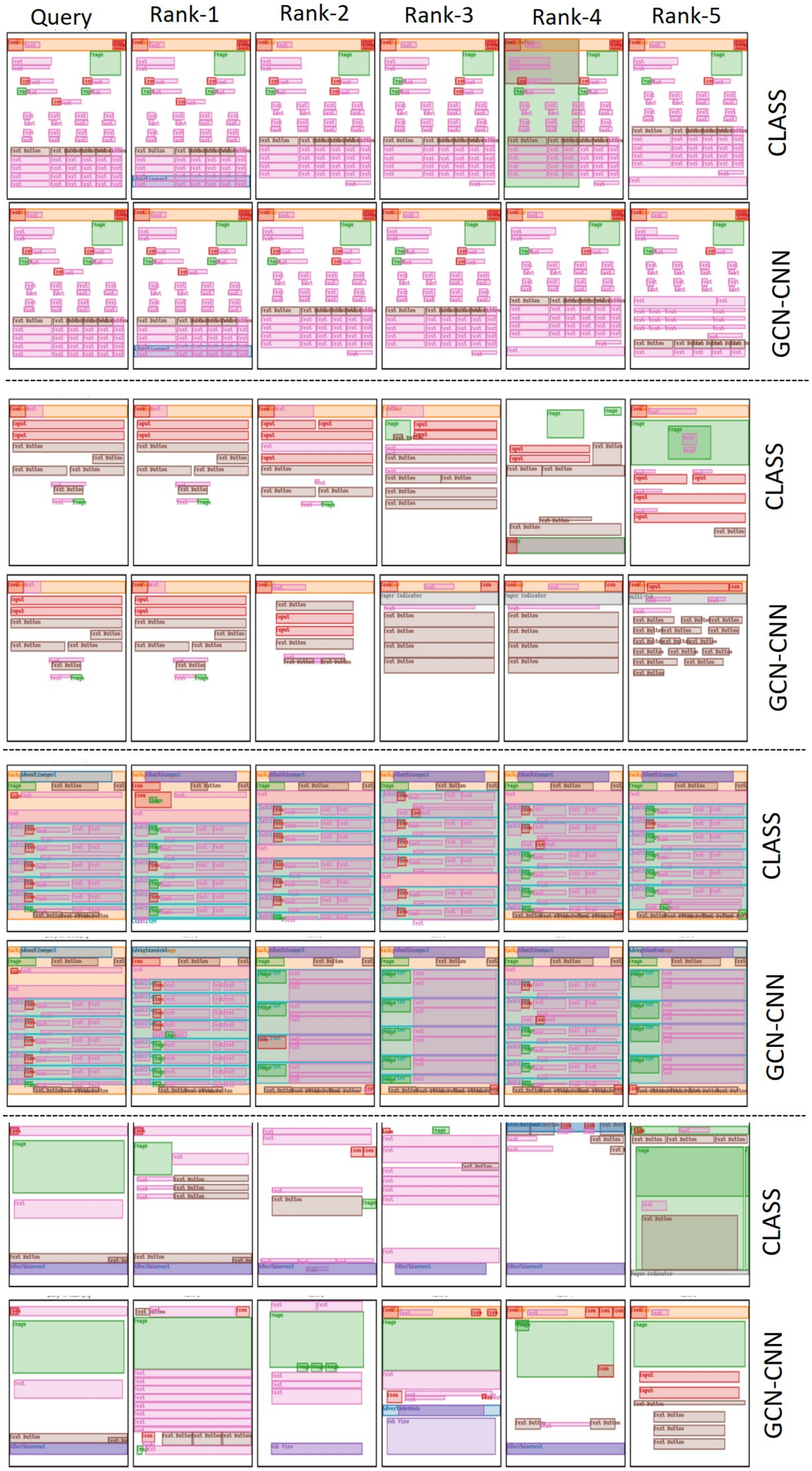

Figure 5. Qualitative results on layout retrieval on RICO dataset.

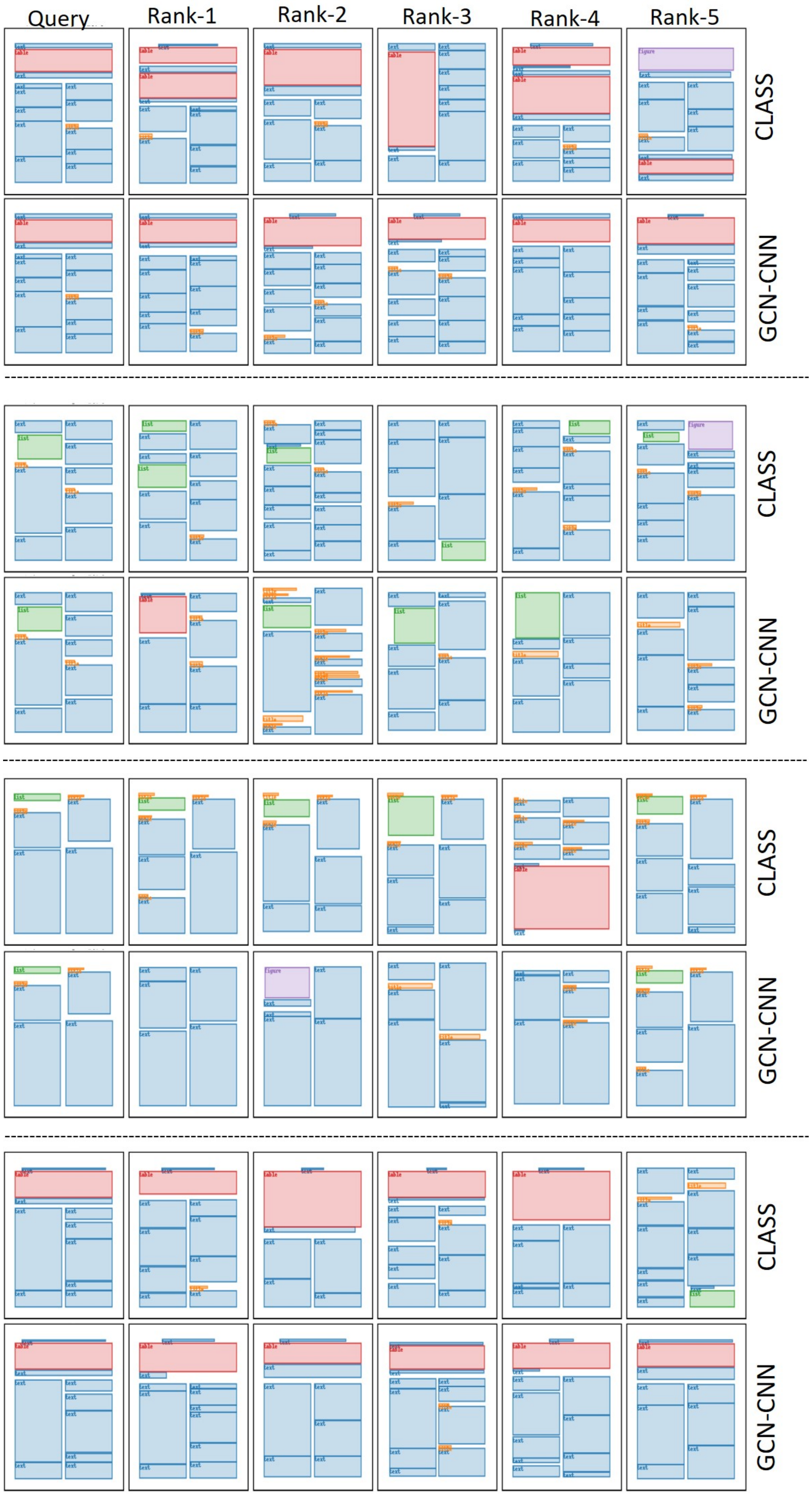

Figure 6. Qualitative results on layout retrieval on PubLayNet dataset.

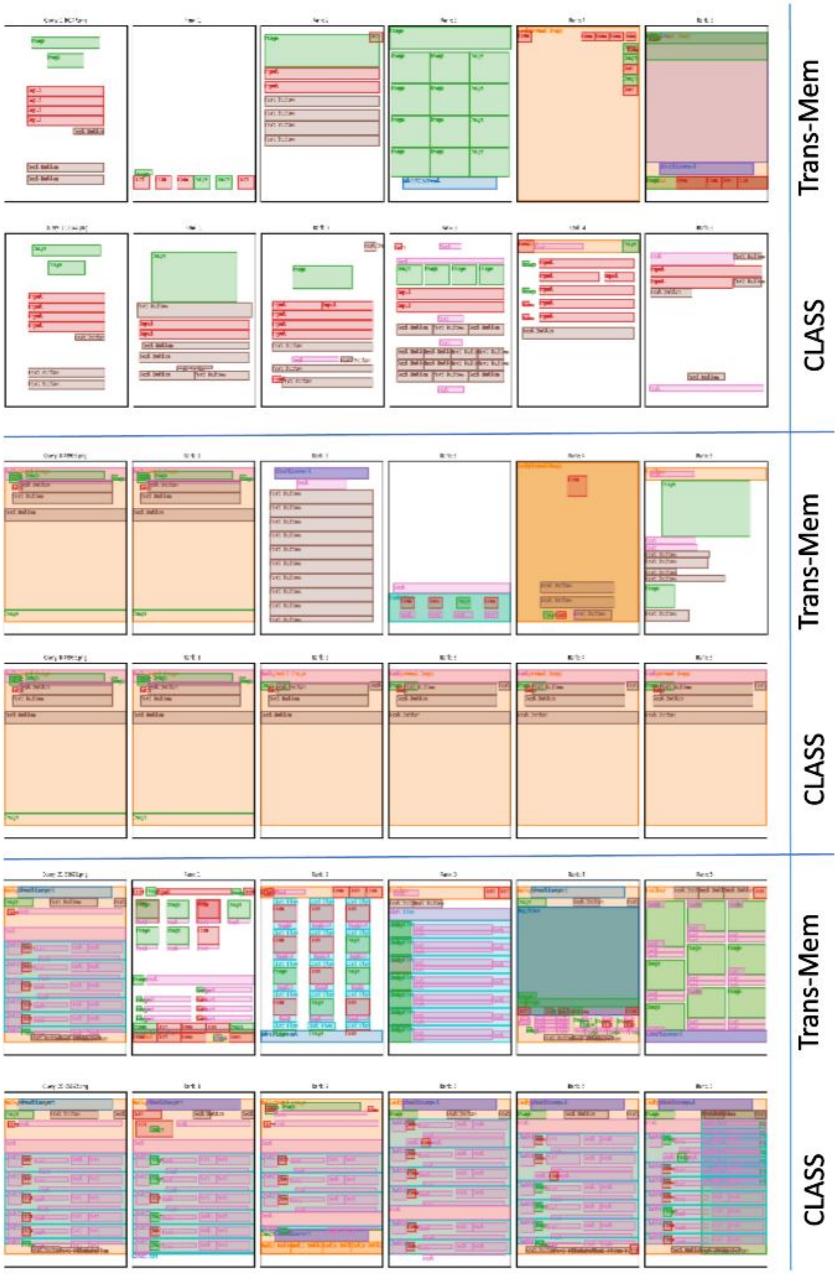

Figure 7. Qualitative retrieval result - Transformer-based methods: CLASS vs TransMem