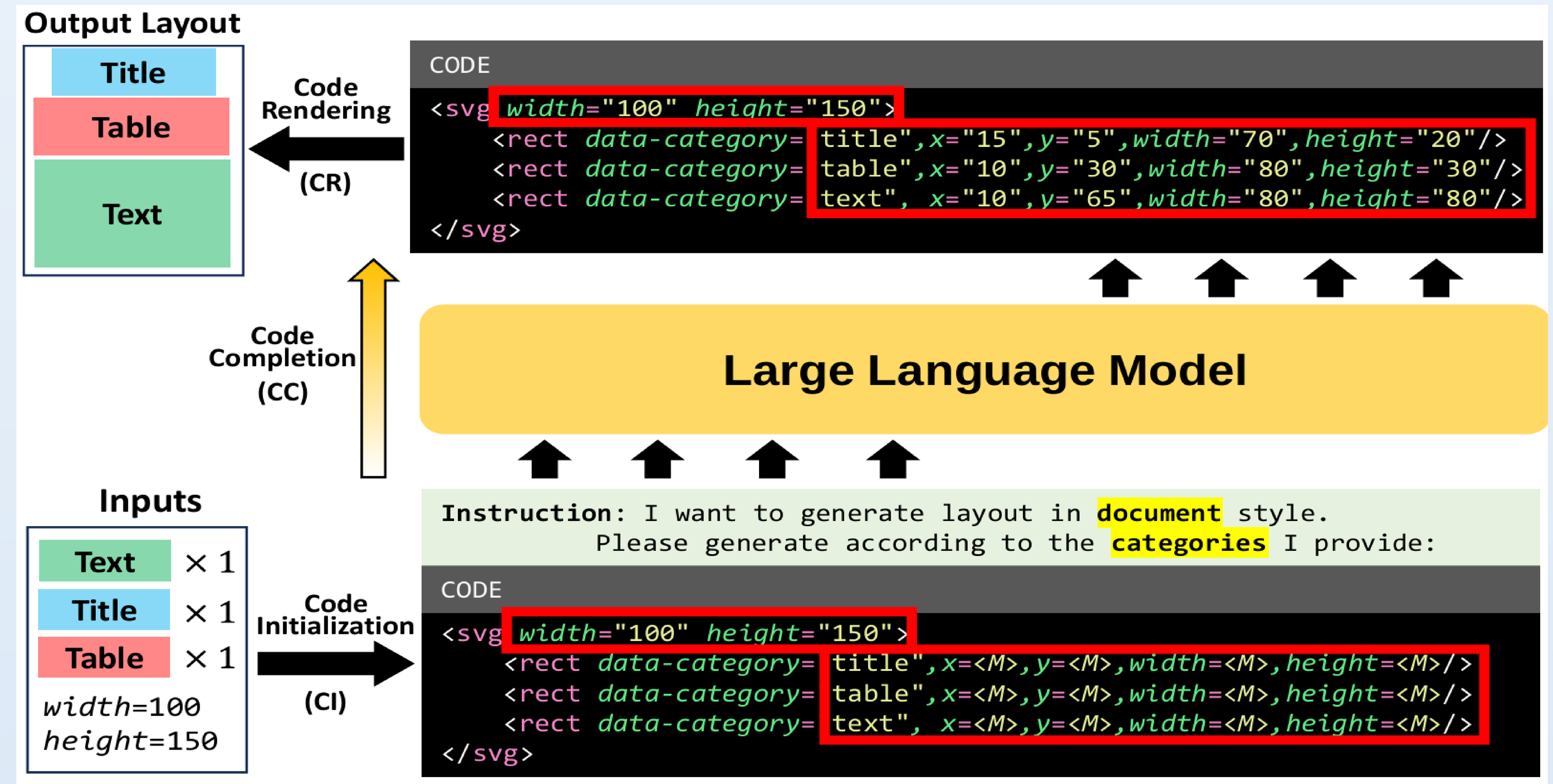
LayoutNUWA: Revealing the Hidden Layout Expertise of Large Language Models

Zecheng Tang [1], Chenfei Wu [2], Juntao Li [1], Nan Duan [2]* [1] *Soochow University, China* [2] *Microsoft Research Asia*

Code/Project Homepage: https://github.com/ProjectNUWA/LayoutNUWA

Overview

- Overview of LayoutNUWA. We propose a Code Instruct Tuning~(CIT) approach that consists of three modules:
- 1) the Code Initialization~(CI) module quantifies the numerical conditions and initializes them as an HTML code with masks;
- 2) the Code Completion~(CC) module utilizes the knowledge of large language models to complete the masked portions within the HTML code;
- 3) the Code Rendering~(CR) module directly renders the completed code into the final graphic layout.



Methodology

Code Initialization

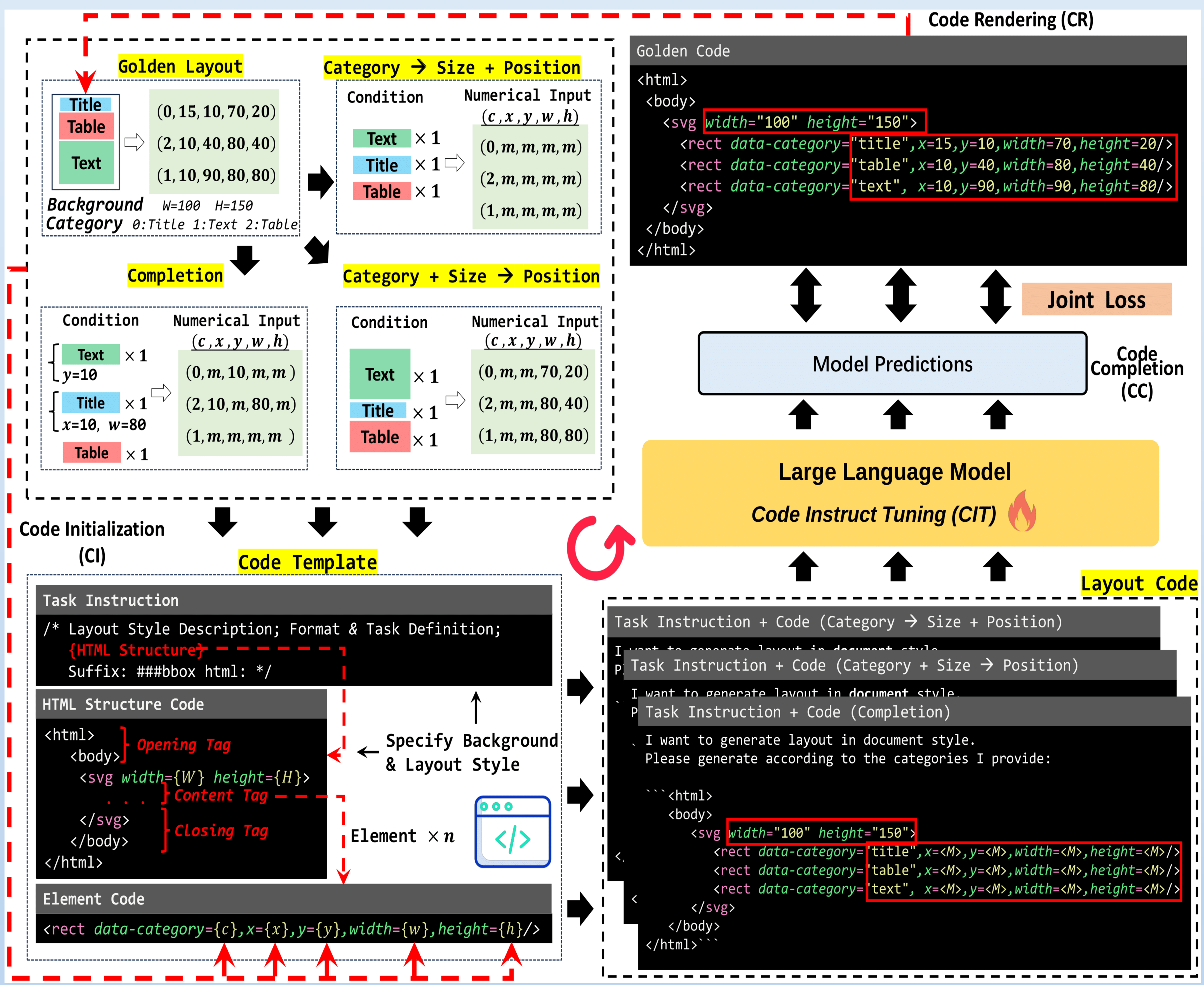
- **Adaptive Quantization**: Clusters element positions and sizes using k-Means algorithm.
- **Precision**: One decimal place.
- **Representation**: Uses absolute positions for direct rendering and precision.

Template Construction

- **HTML-based**: Constructs templates from common web layout code.
- **Tags**: Describes elements with quantified positions and sizes.
- **Layout Structure**: Defines layout boundaries with opening and closing tags.

Code Completion

- **Mask Tokens**: Represents masked values for LLM prediction.
- **Direct Token Vocabulary**: Utilizes LLM's numerical token knowledge.



Streamline Layout Generation Process

The layout generation process begins with adaptive quantization to cluster and precisely define the positions and sizes of the layout elements. This is followed by the creation of HTML templates that use `<rect>` tags to represent each element's layout. To enable diverse layout predictions, language models are utilized to fill in masked values within the HTML code, with a self-consistency strategy that randomizes element order. The output is directly rendered using absolute positions, avoiding conversion losses, and is refined with regular expressions and clipping to ensure accurate webpage rendering. This method streamlines the creation of detailed and accurate layouts for various design requirements.

Experiments

Model	Layout Format	LLM	Domain	C → S + P mIOU (↑)	C → S + P FID (↓)	C + S → P mIOU (↑)	C + S → P FID (↓)	Completion mIOU (↑)	Completion FID (↓)
LayoutTrans	Numerical	-	Specific	0.116	36.207	0.153	33.931	0.228	25.804
BLT	Numerical	-	Specific	0.087	65.372	0.126	41.089	0.103	97.142
LayoutGAN++	Numerical	-	Specific	0.259	16.952	0.293	11.569	-	-
MaskGIT	Numerical	-	Specific	0.059	140.94	0.100	78.226	0.024	152.591
DiffusionLM	Numerical	-	Specific	0.151	32.114	0.144	24.370	0.138	33.172
LayoutDM	Numerical	-	Specific	0.234	19.206	0.308	14.265	0.328	15.804
LayoutNUWA-L2-DS (ours)	Code	LLaMA2	Specific	0.260	9.741	0.358	6.682	0.418	8.257
LayoutNUWA-L2-DA (ours)	Code	LLaMA2	Agnostic	0.293	9.632	0.394	7.238	0.413	8.734
LayoutNUWA-CL-DS (ours)	Code	CodeLLaMA	Specific	0.293	8.985	0.348	**5.355**	0.410	**7.341**
LayoutNUWA (ours)	Code	CodeLLaMA	Agnostic	**0.312**	**8.791**	**0.418**	6.755	**0.495**	7.572
Real Data	-	-	-	0.348	6.695	0.348	6.695	0.348	6.695

Partial experimental results, please refer to the paper for more experimental results. LayoutNUWA achieved SoTA results in all settings.

Generated Cases

Conclusion and Future Outlook

We propose LayoutNUWA, a groundbreaking approach that treats layout generation as a code generation task, effectively enriching the semantic information of layouts and leveraging the hidden expertise of LLMs. Extensive experiments on multiple datasets have demonstrated the superiority of our method. This research has the potential to revolutionize the field of layout generation and pave the way for further exploration and development of semantic-aware layout generation approaches in various applications.