Pacific Graphics 2021
E. Eisemann, K. Singh, and F.-L Zhang
(Guest Editors)

Volume 40 (2021), Number 7

Modeling Visual Containment for Web Page Layout Optimization — Supplemental Material —

1. Overview

In this supplemental material we include implementation details and additional results that were omitted from the main document for brevity. This supplemental document is organized in roughly the same order as the main paper.

2. Dataset

2.1. Collection Issues

Fig. 1 shows some example of websites that were discarded from the dataset due to PHP errors, 404 errors, or domain sales. Despite focusing on the most popular websites, a surprisingly number of domains have to be discarded during collection.

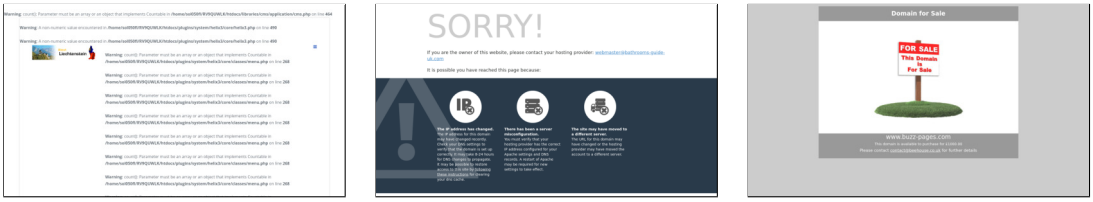


Figure 1: Downside of automatic collection of web pages.

2.2. Labels and Statistics

We choose six labels and design rules to determine the value for each element in each web page. An overview of those rules can be seen in Table 1. We also summarize the key statistics of our dataset in Fig. 2.

Table 1: Labeling rule.

Label name	Description
Text	An element having inner texts and whose line height is positive.
Button	An element whose class names contain "btn" or "button".
Input	An element whose HTML tag is "input".
Image	An element whose HTML tag is "img". / An element whose rendered image is filled with non-negative alpha values and has higher variance in RGB values.
Container	An element whose rendered image is filled with non-negative alpha values and has lower variance in RGB values.
Graphic	None of the above.

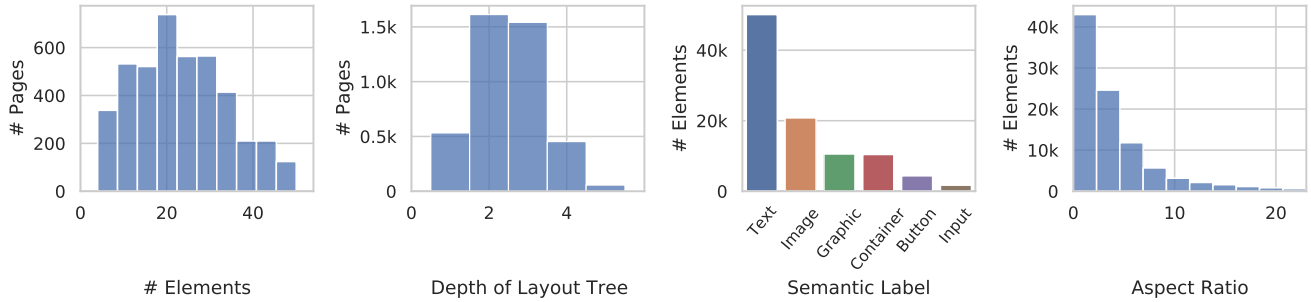


Figure 2: Statistics of our dataset. From left to right, these show page-level frequencies of the number of elements and depth of layout trees, and element-level frequencies of the semantic labels and aspect ratios.

3. Implementation Details

3.1. Layout Parameterization

The layout parameters are a vector of concatenated parameters for each element, including l_x for left-coordinate, l_y for top-coordinate, and l_H for height.

$$\mathbf{X} = \Big\|_{i=1}^{N} [l_x^{(i)}, l_y^{(i)}, l_H^{(i)}]^\top \in [0,1]^{3N} \tag{1}$$

where $\|$ is the concatenation operator.

We restrict the parameter space to preserve the visual containment defined by a layout tree T. The height H and width W of an element are computed as follows:

$$\begin{aligned} H &= l_H(H_{\text{ub}} - H_{\text{lb}}) + H_{\text{lb}} \,, && (2)\\ H_{\text{ub}} &= \min(\hat{H}, \hat{W}/r, H_{\max}) \,, && (3)\\ H_{\text{lb}} &= \max_{\check{H}_{\text{lb}}, \check{W}_{\text{lb}}} \max(\check{H}_{\text{lb}}, \check{W}_{\text{lb}}/r, H_{\min}) \,, && (4)\\ W &= rH \,, && (5) \end{aligned}$$

where r is the aspect ratio, $\hat{H}$ and $\hat{W}$ are the height and width of the parent element, and $\check{H}_{\text{lb}}$ and $\check{W}_{\text{lb}}$ are the lower bound of height and lower bound of width of the descendant element. $H_{\max}$ and $H_{\min}$ are hyperparameters for the maximum and minimum height, which we set to 2 and 0.5 times the height of the ground-truth, respectively, in the experiment.

The left-coordinate x of an element are computed as follows:

$$\begin{aligned} x &= l_x(x_{\text{ub}} - x_{\text{lb}}) + x_{\text{lb}} \,, && (6)\\ x_{\text{lb}} &= \hat{x} \,, && (7)\\ x_{\text{ub}} &= x_{\text{lb}} + \hat{W} - W \,, && (8) \end{aligned}$$

where $\hat{x}$ is the left coordinate of the parent element. The top-coordinate y is defined similarly.

All the parameters have a value from 0 to 1 and have the upper bound and the lower bound. We use un-normalized values for computing energy function values, while we use normalized values for the optimization.

3.2. Layout Tree

A complete list of element features that we used in the tree property estimators is shown in Table 2.

To evaluate the estimated layout tree against the ground-truth, we used three metrics: the F1 score for ancestors F_{anc}, siblings F_{sib}, and

Table 2: Element features used to predict tree partial properties. In the case of predicting properties defined on two elements, the features of both elements are concatenated together.

Name (Dim.)	Description
isText (1)	1 if text element; 0 otherwise.
aspectRatio (1)	element width divided by its height.
meanTrans (1)	mean of transparency values.
meanRGB (3)	mean of RGB values.
varRGB (3)	variance of RGB values.
importance (5)	one-hot vector of importance metadata.

leaves F_{leaf}. Let A^* be the ancestor matrix for ground-truth tree, the F1 score for ancestors is defined as:

$$P_{\text{anc}}(\mathsf{A},\mathsf{A}^*) = \left(1+\frac{\sum_i^N\sum_j^N \mathsf{A}_{i,j}(1-\mathsf{A}^*_{i,j})}{\sum_i^N\sum_j^N \mathsf{A}_{i,j}\mathsf{A}^*_{i,j}}\right)^{-1} \tag{9}$$

$$R_{\text{anc}}(\mathsf{A},\mathsf{A}^*) = \left(1+\frac{\sum_i^N\sum_j^N (1-\mathsf{A}_{i,j})\mathsf{A}^*_{i,j}}{\sum_i^N\sum_j^N \mathsf{A}_{i,j}\mathsf{A}^*_{i,j}}\right)^{-1} \tag{10}$$

$$F_{\text{anc}}(\mathsf{A},\mathsf{A}^*) = \frac{2P_{\text{anc}}(\mathsf{A},\mathsf{A}^*)R_{\text{anc}}(\mathsf{A},\mathsf{A}^*)}{P_{\text{anc}}(\mathsf{A},\mathsf{A}^*)+R_{\text{anc}}(\mathsf{A},\mathsf{A}^*)} \tag{11}$$

Note that the F1 score cannot be defined for layouts where A^* is a zero matrix. We exclude such layouts when calculating the average over the test set. We defined F_{sib} and F_{leaf} similarly.

3.3. Layout Energy Model

3.3.1. Alignment

We consider six possible alignment types for spatially adjacent sibling elements: Left (L), X-center (XC), Right (R), Top (T), Y-center (YC), and Bottom (B). In the energy terms we will use the abbreviated forms of the alignment types to refer to them. We define energy terms that encourage coarse alignment; using Left alignment as an example, which is calculated as follows:

$$E_{\text{AlignL}} = 1-\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} I[|c_i^{\text{L}}-c_j^{\text{L}}|<\theta] \tag{12}$$

where c_i^{L} is the left coordinate of i-th element, θ is the threshold parameter, and $I[\text{condition}]$ is 1 when the condition satisfies, and 0 otherwise. We also define E_{AlignXC}, E_{AlignR}, E_{AlignT}, E_{AlignYC}, E_{AlignB}, similarly. We set $\theta = 0.02$ in our experiment.

We penalize misalignment as follows:

$$E_{\text{FineAlignX}} = 1-\frac{1}{\theta_{\text{X}}}\sum_{\text{align}\in\{\text{L,XC,R}\}}\sum_{i=1}^{N}\sum_{j=1}^{N} I[|c_i^{\text{align}}-c_j^{\text{align}}|<\theta](c_i^{\text{align}}-c_j^{\text{align}})^2 \tag{13}$$

We also define $E_{\text{FineAlignY}}$ similarly. We set $\theta_{\text{X}} = 0.03$ and $\theta_{\text{Y}} = 0.01$ in our experiment.

We define an alignment group as a set of consecutive aligned elements of the same alignment type. We encourage a larger alignment group, *i.e.*, a smaller number of alignment groups in a layout as follows:

$$E_{\text{AlignGroup}} = \min_{\text{align}\in\{\text{L,XC,R,T,YC,B}\}}\frac{|\mathcal{A}^{\text{align}}|}{N} \tag{14}$$

where $\mathcal{A}$ is a set of alignment groups of a particular alignment type. The number of alignment groups can be efficiently calculated as the number of components in a graph constructed with elements as nodes and the presence of alignment as edges.

We also consider the alignment between the parent element and its child elements. Using Left alignment as an example, the energy is calculated as follows:

$$E_{\text{ParAlignL}} = 1-\frac{1}{N}\sum_{i=1}^{N} I[|c_i^{\text{L}}-\hat{c}_i^{\text{L}}|<\theta] \tag{15}$$

where $\hat{c}_i^{\text{L}}$ is the left coordinate of the parent element of i-th element. We also define $E_{\text{ParAlignXC}}$, $E_{\text{ParAlignR}}$, $E_{\text{ParAlignT}}$, $E_{\text{ParAlignYC}}$, $E_{\text{ParAlignB}}$, similarly.

3.3.2. Symmetry

We evaluate the global trend for symmetry by flipping the depth mask $\mathbf{M}^{\text{depth}}$ along an axis as follows:

$$E_{\text{SymmX}} = \frac{\sum_{m=1}^{H}\sum_{n=1}^{W}\min(\mathbf{M}^{\text{depth}}_{m,n}, \mathbf{M}^{\text{depth}}_{m,W-n+1})}{\sum_{m=1}^{H}\sum_{n=1}^{W}\mathbf{M}^{\text{depth}}_{m,n}} \tag{16}$$

We also evaluate asymmetry as $E_{\text{AsymmX}} = 1 - E_{\text{SymmX}}$. E_{SymmY} and E_{AsymmY} are defined similarly.

3.3.3. Spacing

Our model evaluates the global white space as follows:

$$E_{\text{Space}} = \frac{1}{HW}\sum_{m=1}^{H}\sum_{n=1}^{W}\max_{i\in\{i\}_{i=1}^{N}}\mathbf{M}^{i}_{m,n} \tag{17}$$

where $\mathbf{M}^{i} \in [0,1]^{H\times W}$ is the i-th element's mask.

Our model also evaluates the white space with respect to visual containment as follows:

$$E_{\text{TreeSpace}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\frac{\sum_{m=1}^{H}\sum_{n=1}^{W}\max_{i\in\text{chi}(p)}\mathbf{M}^{i}_{m,n}}{\sum_{m=1}^{H}\sum_{n=1}^{W}\mathbf{M}^{p}_{m,n}} \tag{18}$$

where $\text{chi}(\cdot)$ is a function that returns a set of child elements and $\mathcal{P} = \{i \mid \text{chi}(i) \neq \varnothing; i \in \{1,2,\ldots,N\}\}$.

Our model evaluates a layout where the main content is spread throughout. Assuming that the leaf elements are the main content, the energy is calculated as:

$$E_{\text{Spread}} = \frac{1}{|\mathcal{G}|}\sum_{(x,y)\in\mathcal{G}}\min_{i\in\mathcal{V}}\min_{\substack{*\in\{\text{L,XC,R}\}\\ \circledast\in\{\text{T,YC,B}\}}}\left\|\begin{bmatrix}x\\y\end{bmatrix} - \begin{bmatrix}c_i^{*}\\c_i^{\circledast}\end{bmatrix}\right\|_2 \tag{19}$$

where $\mathcal{G}$ is a set of coordinates for each crossing point of the $K\times K$ grid, and $\mathcal{V} = \{i \mid \text{chi}(i) = \varnothing; i \in \{1,2,\ldots,N\}\}$. We set $K = 2$ in our experiment.

We encourage the larger spaces between leaf elements as follows:

$$d(i,j) = \min_{\substack{*\in\{\text{L,XC,R}\}\ \bar{*}\in\{\text{L,XC,R}\}\\ \circledast\in\{\text{T,YC,B}\}\ \bar{\circledast}\in\{\text{T,YC,B}\}}}\left\|\begin{bmatrix}c_i^{*}\\c_i^{\circledast}\end{bmatrix} - \begin{bmatrix}c_j^{\bar{*}}\\c_j^{\bar{\circledast}}\end{bmatrix}\right\|_2 \tag{20}$$

$$E_{\text{Dist}} = 1 - \frac{1}{|\text{comb}(\mathcal{V})|}\sum_{(i,j)\in\text{comb}(\mathcal{V})}d(i,j) \tag{21}$$

where $\text{comb}(\cdot)$ is a function that returns a set of combinatorial pairs of elements in the given set.

The above energy terms facilitate a layout in which elements are spread throughout. Our model also evaluates the larger global margin between the outermost elements and the canvas boundaries as follows:

$$f(\mathbf{b},\mathcal{B}) = \sum_{l=1}^{4}\min_{\mathbf{b}'\in\mathcal{B}}|\mathbf{b}_l - \mathbf{b}'_l| \tag{22}$$

$$E_{\text{Margin}} = 1 - \frac{1}{H+W}f([0,0,W,H]^{\top}, \{\mathbf{b}_i\}_{i=1}^{N}) \tag{23}$$

Our model also evaluates the outermost margin with respect to visual containment as follows:

$$E_{\text{TreeMargin}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\left(1 - \frac{1}{H_p+W_p}f(\mathbf{b}_p, \{\mathbf{b}_i \mid i\in\text{chi}(p)\})\right) \tag{24}$$

Our model encourages the vertical spacing of adjacent text elements to be uniform.

$$E_{\text{UniSpace}} = \text{var}(\{v_{i,j} \mid (i,j)\in\mathcal{U}\}) \tag{25}$$

where $v_{i,j}$ is a vertical space between i-th element and j-th element, $\mathcal{U}$ is a set of pairs of adjacent text elements, and $\text{var}(\cdot)$ is a function that returns the variance of a given set.

3.3.4. Scale

In general, the size of the element should be large enough to be seen, but not too large to be aesthetically unpleasant. Our model has per-label energy terms that encourages the larger sizes of content elements. Using button label as an example, the energy is calculated as follows:

$$E_{\text{EnlargeButton}} = 1 - \frac{1}{|\mathcal{E}_{\text{button}}|} \sum_{i \in \mathcal{E}_{\text{button}}} s_i \tag{26}$$

where s_i is the size of i-th element and is $H_i/[\#\text{lines}]$ when the label is text, and normalized area $H_iW_i/(HW)$ otherwise. $\mathcal{E}_{\text{button}}$ is a set of leaf elements labeled as button. $E_{\text{EnlargeText}}$, $E_{\text{EnlargeInput}}$, $E_{\text{EnlargeGraphic}}$, $E_{\text{EnlargeImage}}$, and $E_{\text{EnlargeContainer}}$ are defined similarly.

Our model also evaluate the variance of element sizes.

$$E_{\text{VarButton}} = \text{var}(\{s_i \mid i \in \mathcal{E}_{\text{button}}\}) \tag{27}$$

E_{VarText}, E_{VarInput}, $E_{\text{VarGraphic}}$, E_{VarImage}, and $E_{\text{VarContainer}}$ are defined similarly.

To manage the size ordering, our model encourages the element sizes to be correlated with the given importance metadata. This is equivalent to a term called Emphasis in other research:

$$E_{\text{CorrText}} = \frac{1 - \text{corr}(\{(s_i, \acute{s}_i) \mid i \in \mathcal{E}_{\text{text}}\})}{2} \tag{28}$$

We also defined $E_{\text{CorrNonText}}$ for non-text elements.

3.3.5. Position

Since web pages have complex layouts, it is difficult to reflect the tendency of the reference design with simple positional statistics for each label. We represent the position of an element as a mask and evaluate its consistency with the mask in the reference design. The mask should cover the reference mask without over or under coverage, so borrowing the concept of the F1 score, we designed the energy terms as follows:

$$P(\mathbf{M}, \mathbf{M}^{\text{ref}}) = \frac{\sum_m^H \sum_n^H \min(\mathbf{M}_{m,n}, \mathbf{M}^{\text{ref}}_{m,n})}{\sum_m^H \sum_n^H \mathbf{M}_{m,n}} \tag{29}$$

$$R(\mathbf{M}, \mathbf{M}^{\text{ref}}) = \frac{\sum_m^H \sum_n^H \min(\mathbf{M}_{m,n}, \mathbf{M}^{\text{ref}}_{m,n})}{\sum_m^H \sum_n^H \mathbf{M}^{\text{ref}}_{m,n}} \tag{30}$$

$$F(\mathbf{M}, \mathbf{M}^{\text{ref}}) = \frac{2P(\mathbf{M}, \mathbf{M}^{\text{ref}})R(\mathbf{M}, \mathbf{M}^{\text{ref}})}{P(\mathbf{M}, \mathbf{M}^{\text{ref}}) + R(\mathbf{M}, \mathbf{M}^{\text{ref}})} \tag{31}$$

$$E_{\text{MatchLabel}} = 1 - F(\mathbf{M}_{\text{label}}, \mathbf{M}^{\text{ref}}_{\text{label}}) \tag{32}$$

$$E_{\text{MatchDepth}} = 1 - F(\mathbf{M}_{\text{depth}}, \mathbf{M}^{\text{ref}}_{\text{depth}}) \tag{33}$$

3.3.6. Overlap and Ordering

We assume that sibling elements do not overlap each other, and penalize overlap as follows:

$$\mathcal{S} = \big\{(i,j) \mid (i,j) \in \text{comb}\big(\text{chi}(p)\big); p \in \mathcal{P}\big\} \tag{34}$$

$$E_{\text{Overlap}} = \frac{1}{|\mathcal{S}|} \sum_{(i,j) \in \mathcal{S}} \frac{\ddot{a}_{ij}}{\min(a_i, a_j)} \tag{35}$$

where a_i is the area of i-th element, and $\ddot{a}_{ij}$ is the area of the intersection of i-th and j-th element.

To preserve the read-order of the elements, we use the following energy term, with the given read-order metadata.

$$\mathcal{O} = \big\{(i,j) \mid \acute{o}_i < \acute{o}_j; (i,j) \in \mathcal{S}\big\} \tag{36}$$

$$o(i,j) = \begin{cases} I[c_j^{\text{XC}} < c_i^{\text{XC}}] & \text{if } e_i \text{ and } e_j \text{ are overlapped along the x-axis} \\ I[c_j^{\text{YC}} < c_i^{\text{YC}}] & \text{if } e_i \text{ and } e_j \text{ are overlapped along the y-axis} \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

$$E_{\text{Order}} = \frac{1}{|\mathcal{O}|} \sum_{(i,j) \in \mathcal{O}} o(i,j) \tag{38}$$

3.4. Optimization

We use *pycma* for CMA-ES implementation. We set the initial standard deviation to 0.99, and the population size to five times the default value.

4. Automatic Evaluation Details

4.1. Reference Search

For searching similar reference designs from ground-truth designs, we use an autoencoder trained with images of size $(192, 342)$ px. The autoencoder we use has 6 convolutional layers followed by one fully connected layer in the encoder, and one fully connected layer followed by 6 convolutional layers in the decoder. All layers use batch normalization and ReLU non-linear activation function, except for the last layer of the decoder that does not use ReLU.

The output of the first convolutional layer in the encoder has 8 channels and uses a stride of 2. Every layer afterwards uses a kernel of size $(3,3)$ px, a stride of 2 px, and doubles the number of output channels. The fully connected layer outputs a 512 dimensional vector.

The decoder is a mirror image of the encoder. It starts with a fully connected layer that converts the 512 dimensional vector into a 4608 dimensional vector, that can be reshaped into a $(3,6)$ px image with 256 channels. Afterwards, each convolutional layer uses a kernel of size $(3,3)$ px, a stride of 2 px, and halves the number of output channels. The final layer outputs an image of the same size as the input.

Training is done with the AdaDelta algorithm [Zei12] and a batch size of 256 layouts for 2000 epochs, and the 512 dimensional vector output by the encoder is used for searching for similar layouts.

4.2. Evaluation Metrics

We evaluate generated layouts with the reconstructive correctness metrics: IoU (d_{IoU}), position error (d_{pos}), and scale error (d_{scale}). The metrics are defined as follows.

$$d_{\mathrm{IoU}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\cap(\mathbf{b}_i,\mathbf{b}_i^{\mathrm{t}})}{\cup(\mathbf{b}_i,\mathbf{b}_i^{\mathrm{t}})} \tag{39}$$

$$d_{\mathrm{pos}} = \frac{1}{N}\sum_{i=1}^{N}\left\| \begin{bmatrix} c_i^{\mathrm{XC}} \\ c_i^{\mathrm{YC}} \end{bmatrix} - \begin{bmatrix} c_i^{\mathrm{XC,t}} \\ c_i^{\mathrm{YC,t}} \end{bmatrix} \right\|_1 \tag{40}$$

$$d_{\mathrm{scale}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\max(a_i,a_i^{\mathrm{t}})}{\min(a_i,a_i^{\mathrm{t}})} \tag{41}$$

where $\mathbf{b}_i$, $\mathbf{b}_i^t$ are the i-th element bounding box for the output layout and the target layout, respectively.

4.3. Ablation study

We investigate how our key components, the hierarchical parameterization via layout tree and the improved energy model, contribute to the performance. The experimental results in various settings are summarized in Table 3, where the same reference designs are used for training unless mentioned. We can see that d_{scale} and d_{IoU} are improved by using the estimated layout tree and the improved energy model, respectively. These can be explained by the fact that the search space for the height parameters is greatly reduced by the layout tree, and by introducing a new energy term that measures the matching with the reference layouts. We can also see that using the improved energy model with the estimated tree instead of the flattened tree improves d_{pos} significantly, which may be comes from the tree-aware energy terms, especially the matching term of the depth mask.

The better results using oracle trees suggest that further improvements in layout estimation can be expected by improving our tree estimation method. The significant improvements by the self-reference setting show that references play an important role in the performance of our layout estimation. We believe that increasing the size of the dataset to pool more diverse references and an efficient interactive search are important.

Table 3: An ablation study of automatic layout optimization.

	Layout tree			Energy model		Self-reference	Metrics		
Method	Flattened	Estimated	Oracle	Base	Improved		$d_{\mathrm{IoU}}\uparrow$	$d_{\mathrm{pos}}\downarrow$	$d_{\mathrm{scale}}\downarrow$
LLSPGD	✓			✓			0.080	0.472	2.384
Ablation-1	✓				✓		**0.098**	0.484	2.250
Ablation-2		✓		✓			0.076	0.476	**2.136**
Ours		✓			✓		0.091	**0.448**	2.152
Ours (oracle w/o self-ref.)			✓		✓		0.117	0.346	2.098
Ours (oracle)			✓		✓	✓	0.330	0.235	1.622

4.4. Additional Results

We show some additional results comparing our method with LLSPGD in Figures 3 and 4.

5. Interactive Evaluation Details

5.1. Additional Energy Term

The additional local exploration term we used is defined as:

$$E_{\text{Local}} = \frac{1}{3N}\sum_{i=1}^{N}(c_i^{\text{XC}} - \bar{c}_i^{\text{XC}})^2 + (c_i^{\text{YC}} - \bar{c}_i^{\text{YC}})^2 + (H_i - \bar{H}_i)^2 \tag{42}$$

where $\bar{c}$ and $\bar{H}$ represent the coordinate and the height of the current layout, respectively.

5.2. Questionnaires

The questionnaires are the five-point Likert scale (1: "strongly agree", 5: "strongly disagree"), and include about the usability of the interface - *"The design interface is easy to use."*, and the suggestions - *"The AI-generated suggestions are helpful."*. The workers who are assigned to the proposed model were asked additional questionnaire about the usability of the treeview - *"The tree view is useful."*.

The results of the questionnaire are summarized in Fig. 5. Our interface received favorable scores overall. The mean values were 2.00 for interface (ours), 2.06 for interface (baseline), 2.54 for suggestion (ours), 2.48 for suggestion (baseline), and 2.36 for suggestion (ours), respectively. Ours got less positive answers than baseline about suggestions. We assume that this is due to LLSPGD having a poorer initialization and thus the suggestions seem to be more useful than those provided by our approach. We note, however, that more users found the suggestions by LLSPGD to be very unhelpful with respect to our approach.

We also asked 58 workers about their experience in professional user interface design: 27 workers had no experience, 7 workers had less than 1 year, 12 workers had 1-3 years, 10 workers had 3-5 years, and 2 workers had more than 5 years. Additionally, we added two qualification requirements when issuing tasks to weed out bad workers: "HIT Approval Rate (%) for all Requesters' HITs greater than or equal to 95" and "Number of HITs Approved greater than or equal to 50".

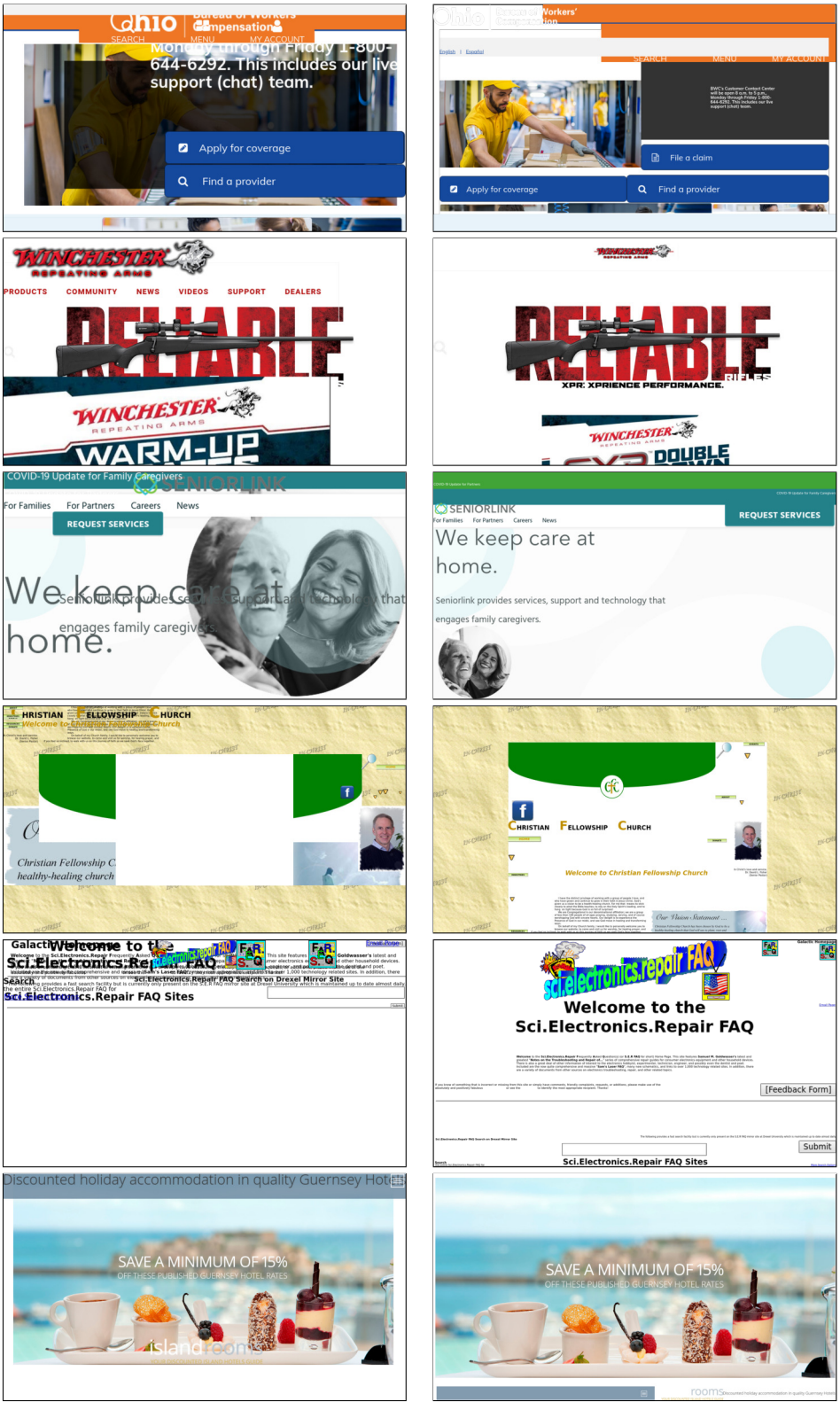
5.3. Feedback

We allowed users to freely input feedback comments about the task. Feedback on the interface was positive, highlighting that it was straightforward and easy to use. The main negative points about the interface were that more functionality would be useful, with customization like commercial web design tools. The feedback regarding the suggestions was also positive, especially helpful during brainstorming. For the approach of LLSPGD people complained about the inconsistency of the results and that it was necessary to lock many elements to obtain good results. Finally, many users found the task to be enjoyable.

Some of the actual positive feedback is listed below.

- Interface
  - (LLSPGD) *Overall, the interface was very easy to use, though, and I enjoyed it very much.*
  - (Ours) *The interface itself is easy to use, but I would have liked more choices. I think that's the creative person in me.*
  - (Ours) *I didn't face any problem. It was easy to use.*
- Suggestion
  - (Ours) *I thought some of the AI suggestions were actually really helpful as I moved through my own ideas.*
  - (Ours) *Most part it was not useful but sometimes it was great giving good ideas about the size of the icons. The AI should automatically resize the text which are beside it or near to it or in the same bar.*
- Enjoyment
  - (LLSPGD) *Thanks for the opportunity, I found this to be rather fun and engaging.*
  - (LLSPGD) *Interesting task.*
  - (LLSPGD) *Enjoy while I doing this designing task.*
  - (LLSPGD) *Happy to participate on this AI survey.*
  - (Ours) *That was fun, thanks.*
  - (LLSPGD) *Nice task and It is very easy.*

Some of the actual negative feedback is listed below.



(a) LLSPGD (b) Ours

Figure 3: Additional results comparing LLSPGD (a) with our proposed approach (b).

(a) LLSPGD (b) Ours

Figure 4: Additional results comparing LLSPGD (a) with our proposed approach (b).

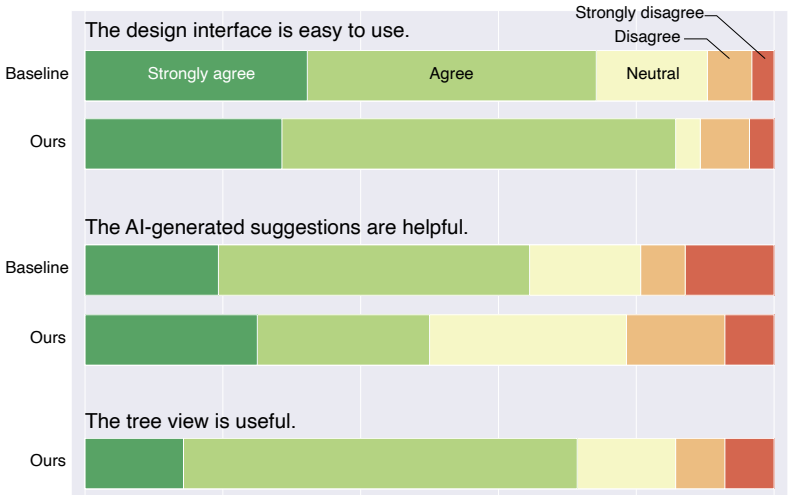


Figure 5: Summary of questionnaire answers. The users responded favorably to all questions.

- Interface
  - (LLSPGD) *I found some of the controls difficult to use, but of course, I have no experience.*
  - (LLSPGD) *The AI and interface were a little difficult to use. If they ran a little smoother, it would be more helpful.*
  - (Ours) *I wish it was more customizable, something like wix.com*
  - (LLSPGD) *More option required to customize the website design.*
- Suggestion
  - (LLSPGD) *I felt I had to lock too many things for the AI to make any marginally helpful decisions – in other words, I had to make nearly all of the decisions to get useful suggestions, which isn't very helpful. I also felt it didn't order the text well, and it seemed to just throw it anywhere. I used it for vague ideas, but it wasn't very helpful for that, either.*
  - (LLSPGD) *I think the tool sometimes helped me find a suitable design quickly but other times seemed not to be effective.*

5.4. User Behavior Analysis

We investigate how the participants used our design tools. The transition of user actions is shown in Fig. 6, in which the edges represent the probability of taking a head action after a tail action. Action transitions with small probabilities are removed for simplicity. Most of the actions were to move and scale elements. We can see that all actions tend to be taken consecutively. We also see a tendency to accept suggestions right after the session starts. Since the initial layout is already the optimal solution for the energy model, this may indicate a discrepancy between the user's preferences and the energy model.

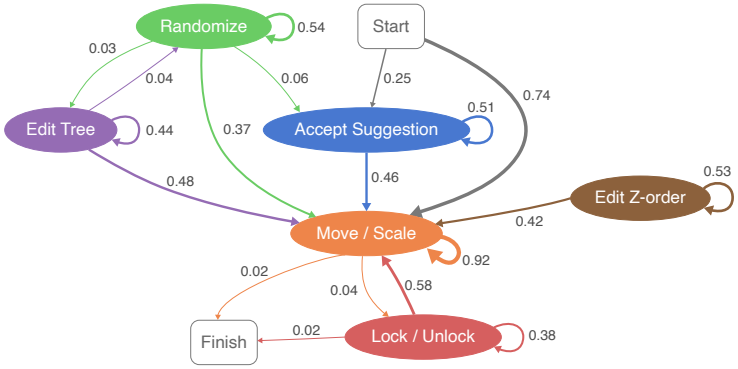


Figure 6: Graph of users' action transition. The probably of each transition is shown on the edges.

5.5. Evaluation of User-Generated Designs

We performed pairwise comparisons of user-generated designs with the methods in the same way as in the prior experiment for automatic evaluation. We selected the first 100 designs for both models and collected five votes for each comparison. 65 workers participated in the experiment. Using the Pearson's chi-square test, we do not found a significant difference in the number of votes for both questions about quality ($p = 0.53$) and similarity ($p = 0.33$), which is to be expected as the users are allowed to edit the web page until they are satisfied with the results. We also observed some cases where the user found a good solution that was completely different from the original design.

Table 4: User voting result for user-generated designs with both our approach and LLSPGD.

	# Votes	
Method	Quality	Similarity
LLSPGD	**257**	**261**
Ours	243	239

References

[Zei12] ZEILER M. D.: ADADELTA: an adaptive learning rate method. *CoRR abs/1212.5701* (2012). 6