LayoutEnhancer: Generating Good Indoor Layouts from Imperfect Data

KURT LEIMER, NJIT/TU Wien, United States/Austria
PAUL GUERRERO, Adobe Research, United Kingdom
TOMER WEISS, NJIT, United States
PRZEMYSLAW MUSIALSKI, NJIT, United States

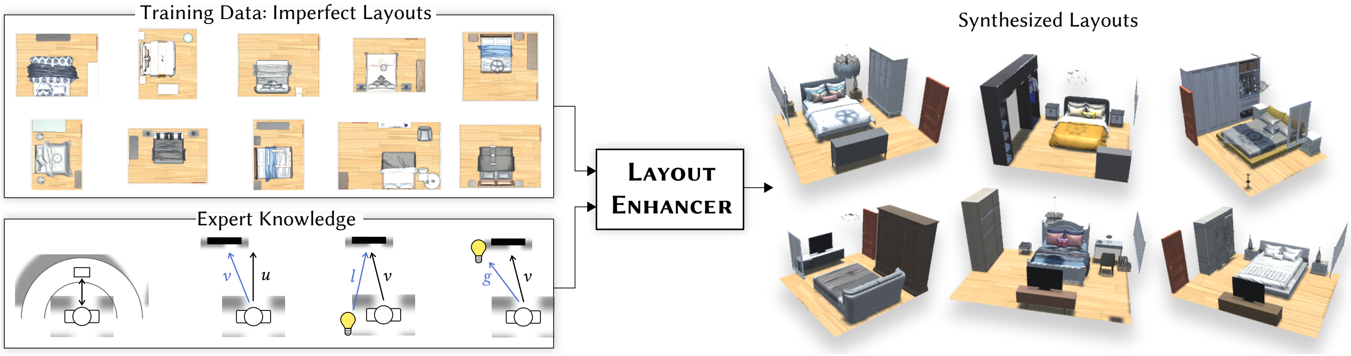


Fig. 1. Our proposed LayoutEnhancer combines data-driven learning from potentially imperfect data with expert knowledge. Generated layouts are biased to follow rules laid out in the expert knowledge, effectively reducing the impact of data imperfections. See Figure 2 for examples of imperfections that are avoided due to the inclusion of expert knowledge.

We address the problem of indoor layout synthesis, which is a topic of continuing research interest in computer graphics. The newest works made significant progress using data-driven generative methods; however, these approaches rely on suitable datasets. In practice, desirable layout properties may not exist in a dataset, for instance, specific expert knowledge can be missing in the data. We propose a method that combines expert knowledge, for example, knowledge about ergonomics, with a data-driven generator based on the popular Transformer architecture. The knowledge is given as differentiable scalar functions, which can be used both as weights or as additional terms in the loss function. Using this knowledge, the synthesized layouts can be biased to exhibit desirable properties, even if these properties are not present in the dataset. Our approach can also alleviate problems of lack of data and imperfections in the data. Our work aims to improve generative machine learning for modeling and provide novel tools for designers and amateurs for the problem of interior layout creation.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; **Neural networks**; *Computer vision.*

Additional Key Words and Phrases: neural networks, layout synthesis, interior design, interior layout generation

ACM Reference Format:
Kurt Leimer, Paul Guerrero, Tomer Weiss, and Przemyslaw Musialski. 2022. LayoutEnhancer: Generating Good Indoor Layouts from Imperfect Data. In *SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers), December 6–9, 2022, Daegu, Republic of Korea.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3550469.3555425

SIGGRAPH Asia '22 Conference Papers, December 6–9, 2022, Daegu, Republic of Korea

1 INTRODUCTION

Indoor spaces play a central role in our everyday lives. The synthesis and design of indoor layouts (apartment layout, workplace layout) is a long-standing problem in several disciplines, including graphics [Fisher et al. 2012; Merrell et al. 2011].

In this paper, we address the problem of data-driven layout synthesis that has recently gained renewed interest in computer graphics due to the advent of a novel neural methods in generative machine learning [Para et al. 2021; Paschalidou et al. 2021]. However, despite recent progress, interior layout synthesis is still challenging for machine learning algorithms. The problem is twofold:

First, reliable training data is difficult to obtain. Designs need to be crafted manually by professionals, making the process labor- and time-intensive and hence expensive.

Second, readily available datasets may have been created by non-experts and may contain several issues like incorrect intersections, unrealistic placement, misplaced objects, etc. (cf. Figure 2). At the same time, high-quality indoor design requires expert knowledge because good furniture arrangements are connected to several considerations like functionality, usability, aesthetics, cost-effectiveness, and ergonomics. These may not all be reflected in a dataset, which contains layouts that were most likely not created by interior design experts.

We address these problems by using a Transformer-based generative model with additional expert knowledge "injected" into the data-driven training process. Transformers are generative models originally proposed for natural language processing that have proven very successful in a wide range of domains [Vaswani et al.

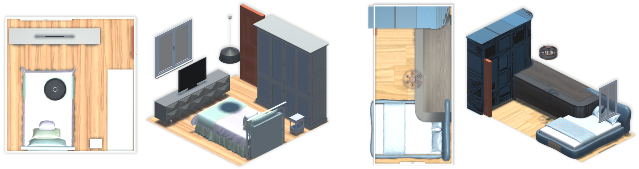


Fig. 2. LayoutEnhancer can learn to improve issues found in imperfect data like *ergonomic issues* (left room): (i) a window directly behind the TV causes glare on sunny days, making it difficult to watch due to a big contrast in brightness. (ii) Insufficient illumination for reading a book without a light source behind or beside the bed; and *geometric issues* (right room): (i) desk is intersecting with the bed and the closet; (ii) closet is covering the door.

2017]. Recently, several methods have successfully used transformers for layout generation [Para et al. 2021; Paschalidou et al. 2021; Wang et al. 2020].

In our approach, a layout S is defined as a sequence of discrete elements $S := \{F_0, \dots, F_N\}$, each represented with a fixed-length parameter vector. A traditional generative model learns to generate new layouts according to a probability distribution $p(S)$ that approximates the probability distribution of the dataset $p(S) \approx p_{\text{data}}(S)$.

We propose to inject additional information based on expert knowledge into the learning process to obtain a learned distribution $p'(S)$ that reflects both the dataset distribution and the additional information. The expert knowledge biases the learned probability distributions to emphasize or de-emphasize specific properties of the layouts. In Section 3 we derive a set ergonomic rules from expert literature [Kroemer 2017].

We integrate this information into the loss function of our transformer-based generative model in two ways: (i) as weights of training samples and (ii) as additional loss that assesses the quality of samples proposed during the training process. In the second case, expert knowledge needs to be differentiable w.r.t. the predicted probabilities. We discuss the details in Section 4.

In Section 5, we evaluate the proposed method and compare it to a recent data-driven method that does not utilize expert knowledge [Paschalidou et al. 2021]. We demonstrate that with our approach we can improve the ergonomic quality of generated layouts, effectively increasing the perceived realism compared to others.

In summary, the contributions of this paper are three-fold:

- We introduce a differentiable ergonomic loss that can be used to assess the ergonomic quality of interior layouts. We derive this loss from the expert knowledge in ergonomics (Section 3).
- We integrate this differentiable loss into the training of a Transformer network (Section 4).
- We empirically show that we can train a generative model with this loss that creates samples with increased ergonomic quality and realism compared to the state of the art (Section 5).

2 RELATED WORK

Interior spaces and their layouts are part of everyday life. For example, organizations such as Ikea and Wayfair are actively working toward understanding their customers needs [Ataer-Cansizoglu et al. 2019]. Typically, each domain has different requirements and needs, which require manual design [Wayfair 2022].

In practice, designing layouts is a laborious task due to high dimensional design space, ranging from selecting relevant furniture pieces, to arranging the target space to fit the design goals. To alleviate such manual workflow, researchers have proposed multiple computational methods to assist in layout design. Below we classify previous work based on their approach.

Deep Learning Methods. Such methods employ neural networks, in which the network learns layout patterns from images, graphs, or other data. Such 3d scene data and the data modality is an important factor in deep learning [Fu et al. 2021a]. Early deep learning work utilizes top-down images of layouts to understand object-object layout relationships [Wang et al. 2018]. However, images do not naturally contain sufficient detail for the network to synthesize complex human-centered layouts. Graphs have also been proposed as a means to encode spatial layout information [Luo et al. 2020; Wang et al. 2019; Zhou et al. 2019].

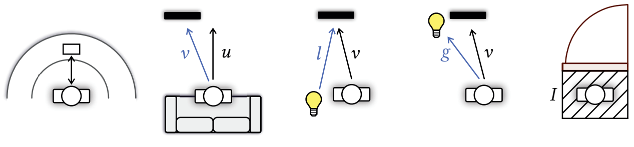
In addition to images and graphs, researchers explored how to use other 3d scene data representations for synthesis. Zhang et al. [2019] synthesize scenes by sampling from a vector that represents the spatial structure of a scene. Such structure encodes a hierarchy of geometrical and co-occurrence relations of layout objects. [Zhang et al. 2020] proposed a hybrid approach that combines such vector representation with an image-based approach. Also other utilize graph structures to describe scene layouts [Di et al. 2020]. Yang et al. [2021] combine such vector representation with Bayesian optimization to improve furniture placement predictions of the generative network. Recently, variational autoencoders have been proposed for indoor layout synthesis [Chattopadhyay et al. 2022].

Most recently, researchers have proposed to use neural networks based on transformers [Paschalidou et al. 2021; Wang et al. 2020]. However, in contrast to our method, their work does not account for ergonomic qualities which results in misplaced furniture items.

Other Approaches. Before the era of deep learning, early work considered layout synthesis as a mathematical optimization problem, where a set of constraints describe the layout quality in terms of energy an energy functional [Merrell et al. 2011; Weiss et al. 2018; Yu et al. 2011]. The layout is then optimized via stochastic or deterministic optimization process.

Other researchers proposed data-driven methods. Qi et al. [2018] use interaction affordance maps for each layout object for stochastic layout synthesis. Similarly, Fisher et al. [2015] used annotated 3d scans of rooms to identify which activities does an environment support. Other researchers also learn layout structure from 3d scans for scene synthesis [Kermani et al. 2016]. They extract manually defined geometric relationships between objects from such scans, which are then placed using a stochastic optimization.

Other research has made progress towards incorporating human-centered considerations for 3d scene synthesis. Fu et al. [2017] use a graph of objects to guide a layout synthesis process. However, they only consider static human poses in relation to activities. Zhang et al. [2021] and Liang et al. [2019] focus on optimal work-space design. While the authors demonstrate novel use of simulation and dynamic capture of agent in action metrics, they only focus on



a) reach b) visibility c) lighting d) glare e) access

Fig. 3. Ergonomic rules implemented in our system. We chose these guidelines as they are essential in most indoor scenarios, like reading a book, watching TV, or working at the desk or the computer. We convert the rules to scalar cost functions and evaluate them using activities (cf. Section 3).

mobility and accessibility based factors. In [Puig et al. 2018], the authors demonstrate how to evaluate the functionality of layouts. However, this work does not include 3d scene synthesis.

Early work [Merrell et al. 2011; Yu et al. 2011] has also included ergonomic and interior design knowledge into the layout design process. Our approach differs from these existing methods in two major aspects. First of all, their methods require the manual definition of a number of additional layout design rules. Second, their methods are designed to optimize the arrangement of an existing furniture layout, while our approach can synthesize entirely new layouts with desired characteristics.

3 ERGONOMIC COSTS

To derive a set of rules used to quantify an ergonomic quality of a design, we studied the literature of ergonomic guidelines [Kroemer 2017]. As a result, we order the information in a hierarchical manner, using the building blocks of activities, actions and ergonomic costs.

An activity is a set or sequence of actions that need to be performed to accomplish a specific goal [Puig et al. 2018]. An activity could be, for instance, reading a book or watching TV. A single action puts specific elements of a layout into a common context, for example looking at the TV while sitting on the sofa. Ergonomic costs are evaluated for each action to quantify how suitable the arrangement of the layout elements is in an ergonomic sense.

The ergonomic losss obtained for each evaluated ergonomic rule are then aggregated up the hierarchy to obtain the losss for each action, activity and finally for the whole layout. This formulation makes it easy to define new evaluation functions for different activities by combining the various building blocks. In our approach, we consider the following ergonomic costs (cf. Figure 3):

- Reach measures how easy it is to interact with a target object from a given position.
- Visibility measures how visible a target object is for a given position and viewing direction.
- Lighting measures how well an object is illuminated by light sources in the room.
- Glare measures the decrease in visual performance from strong brightness contrast caused by having bright light sources in the field of view.
- Accessibility measures how much free space is in front of a target object to allow easy interaction and walking by.

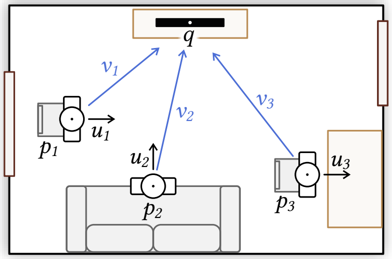


Fig. 4. Human activity in the room based on the example of *Watch TV*. For all possible sitting locations p_j an avatar is sampled and the ergonomic rules for visibility and glare are evaluated. The final contribution is the weighted sum of costs over every combination of a sitting possibility p_j and all TVs q_k. Please refer to Section 3 for more details.

We choose the above five rules as examples for two reasons. First, they are all relevant for the kinds of activities that are often performed in the prevalent room types that are included in publicly available indoor layout datasets . The second reason is a practical one, since these rules can be defined as (piecewise) differentiable scalar functions in a range of $[0, 1]$, which perfectly suits our needs.

For instance, given a target object at position q_k viewed from position p_j and viewing direction u_j, we define the *visibility cost* as smooth scalar function E_V of two vectors u_j and $v = \frac{q_k - p_j}{\|q_k - p_j\|}$ which can be minimized:

$$E_V = 1 - \left(\frac{1 + \langle u_j, v \rangle}{2} \right) .$$

Together with the *glare cost* function $E_G \left(p_j, B, q_k \right)$ with light sources B, we can compute the loss for the activity *Watch TV* (cf. Figure 4):

$$e^{tv}_{j,k} = \frac{E_V \left(p_j, u_j, q_k \right) + E_G \left(p_j, B, q_k \right)}{2} .$$

Since there can be multiple TVs in a room in addition to multiple pieces of seating furniture, we need to compute the weighted sum of costs over every combination of p_j and q_k, using $e^{tv} = [e^{tv}_{j,k}]_{j \in P, k \in Q}$:

$$E_{tv} = \langle e^{tv}, \mathrm{softmin}(\beta \cdot e^{tv}) \rangle .$$

The costs of every possible activity are then aggregated to obtain the total **ergonomic loss** E of the layout. Figure 3 depicts all five ergonomic cost functions implemented in our framework in a similar differentiable fashion. We refer the reader to supplemental material for details on the implementation of the other ergonomic cost functions and activities.

4 LAYOUT GENERATION WITH EXPERT KNOWLEDGE

We build on top of Transformers [Vaswani et al. 2017] as generative model for layouts [Para et al. 2021; Paschalidou et al. 2021; Wang et al. 2020]. In this section, we first present our model and then describe how we integrate our ergonomic loss into the training.

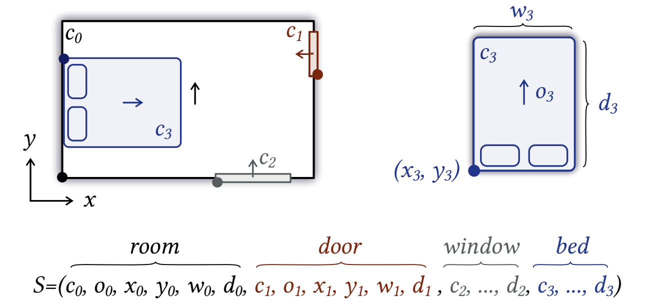

c_0
c_1
c_3
c_2
y
x
w_3
c_3
o_3
d_3
(x_3, y_3)
room
door
window
bed
$S=(c_0, o_0, x_0, y_0, w_0, d_0, c_1, o_1, x_1, y_1, w_1, d_1, c_2, ..., d_2, c_3, ..., d_3)$

Fig. 5. A layout is represented as a sequence $S = (s_1, \ldots, s_n)$. Each individual token s_i in the sequence represents an attribute of a furniture object, such as its category, orientation, position or dimensions.

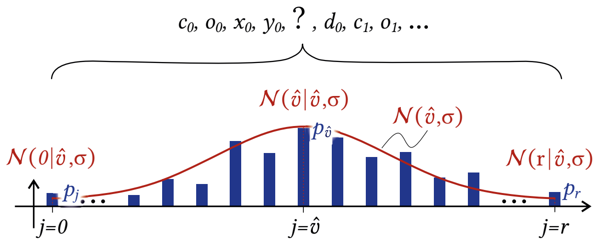

$c_0, o_0, x_0, y_0, ?, d_0, c_1, o_1, \ldots$
$\mathcal{N}(\hat{v}|\hat{v},\sigma)$
$\mathcal{N}(\hat{v},\sigma)$
$\mathcal{N}(0|\hat{v},\sigma)$
$\mathcal{N}(r|\hat{v},\sigma)$
$p_{\hat{v}}$
p_j
p_r
$j=0$
$j=\hat{v}$
$j=r$

Fig. 6. To propagate the ergonomic loss back to the token probabilities, we choose the maximum of the discrete values of the predicted token and convolve the neighborhood with a Gaussian kernel, centered at the discrete maximum. The resulting token value is a weighted sum of the discrete values in this neighborhood, weighted by the probability and distance to the kernel center of each discrete value. Please refer to Section 4.2 for more details.

4.1 Generative Model

Transformers are sequence generators that originate from natural language processing. A layout is generated step-wise as a sequence of discrete tokens $S = (s_1, \ldots, s_n)$, one token s_i at a time. Thus, we first need to define a sequence representation of our layouts.

Sequence representation. Each furniture object is represented as a 6-tuple $F_i = (c_i, o_i, x_i, y_i, w_i, d_i)$, with c_i indicating the object category, such as *chair* or *table*, o_i the orientation, x_i and y_i being the x- and y-coordinates of the bottom left corner of the furniture object, w_i being the width, and d_i the depth of the furniture object (cf. Figure 5) . Since previous work [Paschalidou et al. 2021] has shown that randomizing the order of objects that do not admit a consistent ordering can be beneficial, we follow a similar approach. The bounding box of the room itself is represented as the furniture object F_0 and is thus always the first of the ordered furniture objects, followed by the doors and windows of the layout. The order of all other furniture objects is not consistent and instead randomized during training. We concatenate the 6-tuples of the ordered furniture objects and add a special stop token to the end of the sequence to obtain the sequence S. An example can be seen in Figure 5.

Similar to previous work [Wang et al. 2020], we use two additional parallel sequences to provide context for each token in S: a position sequence $S^P = (1, 2, \ldots, n)$ that provides the global position in the sequence, and an index sequence $S^I = (1, 2, \ldots, 6, 1, 2 \ldots, 6)$ that describes the index of a token inside the 6-tuple of a furniture object.

Our approach also supports an alternate method of providing the room shape as a binary map of the floor plan, similar to ATISS [2021]. While specifying the room as part of the sequence allows the network to learn how to synthesize arbitrary rectangular rooms, using a binary map instead lets the network learn how to generate furniture layouts for more complex non-rectangular room shapes.

Quantization. Transformers typically operate with discrete token values. By learning to predict a probability for each possible value of a token, a transformer can model arbitrary distributions over token values. To obtain discrete values, we quantize all object parameters except orientations o_i and categories c_i uniformly between the minimum and maximum values that occur in the dataset. Orientations o_i are uniformly quantized in $[0, 2\pi)$, adjusting the resolution to preserve axis-aligned orientations as integer values. We use a resolution of $r = 256$. Categories c_i do not require quantization as they are already integers. We use categorical distributions for all tokens.

Sequence generation. Our Transformer-based sequence generator f_θ factors the probability distribution over sequences S into a product of conditional probabilities over individual tokens:

$$p(S|\theta) = \prod_i p(s_i|s_{<i}, \theta),$$

where $s_{<i} \coloneqq s_1, \ldots, s_{i-1}$ is the partial sequence up to (excluding) i. Given a partial sequence $s_{<i}$, our model predicts the probability distribution over all possible discrete values for the next token: $p(s_i|s_{<i}, \theta) = f_\theta(s_{<i}, s^P_{<i}, s^I_{<i})$ that can be sampled to obtain the next token s_i. Here $s^P_{<i}$ and $s^I_{<i}$ are the corresponding partial position and index sequences that are fully defined by the index i. We implement f_θ as a GPT-2 model [Radford et al. 2019] using the implementation included in the Huggingface library [Wolf et al. 2020].

4.2 Ergonomic Loss

A loss designed by an expert, such as an ergonomic rule, defines desirable properties of layouts that may not be fully realized in a dataset. However, while minimizing the expert loss may be *necessary* to obtain a desirable layout, it is usually not *sufficient*, since a manually defined loss can usually not describe *all* desirable properties of a layout exhaustively. Our goal is thus to combine the expert loss with a data-driven generative model for layouts. However, integrating the ergonomic loss in a transformer-based generative model poses two main challenges:

C1: Transformers generate layouts in multiple steps, each step generating a small part of the layout such as a single object or a single object attribute. Each step, where only a partial layout has been generated, requires supervision, but the ergonomic loss cannot reliably be computed on a partial layout.

C2: The ergonomic loss is defined over continuous parameters, such as object positions or orientations. However, transformers typically output a probability distribution over a discrete set of values in each step, such as quantized object positions or orientations. This makes gradient propagation from the ergonomic loss to the transformer difficult.

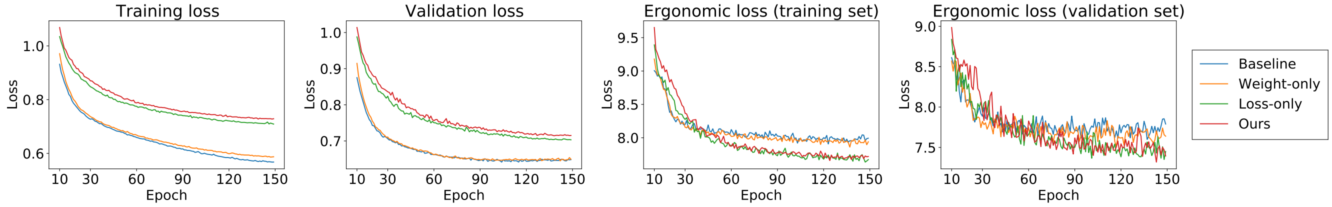


Fig. 7. Cross-entropy loss and ergonomic loss for our model and its ablations, evaluated on the Bedrooms dataset. The training loss and validation loss refer to the cross-entropy loss on the training and validation sets, respectively. By including our proposed ergonomic loss term during training we can significantly decrease the ergonomic loss of synthesized layouts.

To tackle the first challenge (**C1**), we observe that transformers are typically trained with a strategy called *teacher forcing*, where the partial sequence $s_{<i}$ preceding the current token s_i is taken from a ground truth layout. Thus, when generating a token s_i, we can evaluate the ergonomic loss on the layout defined by $s_{<i}, s_i, s_{>i}$, where only s_i is generated and both the preceding tokens $s_{<i}$ and the following tokens $s_{>i}$ are taken from the ground truth, effectively evaluating s_i in the context of the ground truth layout.

To solve the second challenge (**C2**) we need an ergonomic loss that is differentiable w.r.t. the probabilities $p(s_i|s_{<i}, \theta)$ predicted by our generative model. A straight-forward solution computes the expected value of the ergonomic loss E over all possible values v_j of a token $\sum_j E(s_{<i},\ v_j,\ s_{>i}) P(s_i = v_j|s_{<i}, \theta)$. This solution is differentiable w.r.t. the probabilities, but requires an evaluation of the ergonomic loss for each possible value of a token, which is prohibitively expensive. Instead, we opt for a less exact but much more efficient approach, where only a single evaluation of the ergonomic loss per token is needed. We compute the ergonomic loss $\mathcal{L}_E$ as the ergonomic loss for the expected value of a token in a small window around the most likely value of the token:

$$\mathcal{L}_E = E(s_{<i},\ \bar{v},\ s_{>i}),\ \text{with} \tag{1}$$

$$\bar{v} = \frac{\sum_j \left(\mathcal{N}(v_j|\hat{v}, \sigma)\ P(s_i = v_j|s_{<i}, \theta)\ v_j\right)}{\sum_j \left(\mathcal{N}(v_j|\hat{v}, \sigma)\ P(s_i = v_j|s_{<i}, \theta)\right)},$$

where $\mathcal{N}(x|\hat{v}, \sigma)$ is the normal distribution centered at $\hat{v}$ with standard deviation σ. $\hat{v}$ is the token value with highest probability, and σ is set to $1/r$ in our experiments. Figure 6 illustrates the approach. This loss provides gradients to all values in smooth window. Note that increasing the size of the window by increasing σ would propagate the gradient to a larger range of token values, but could also result in expected token values $\bar{v}$ that are in low-probability regions of the distribution $p(s_i|s_{<i}, \theta)$, since the distribution may be multi-modal. The total loss function $\mathcal{L}$ is then given by

$$\mathcal{L}\left(S^k\right) = \beta_T \mathcal{L}_T \left(S^k\right) + \beta_E \mathcal{L}_E \left(S^k\right), \tag{2}$$

with $\mathcal{L}_T$ being the cross-entropy loss, $\mathcal{L}_E$ being our proposed ergonomic loss and β_T, β_E being weights that determine the influence of the two loss terms to the overall loss. We use $\beta_T = 1 - E\left(S^k\right)$ and $\beta_E = E\left(S^k\right)$, such that the cross-entropy loss has higher influence for training samples with better ergonomic loss while the ergonomic loss is more important for samples with lower ergonomic loss. Essentially, we want the network to learn about the general target distribution from examples that are already considered good, while learning how to improve the ergonomic loss from bad examples. In Section 5.1, we discuss the influence of the weights β_T and β_E in more detail.

4.3 Training and Inference

We train our models using the 3DFRONT dataset [Fu et al. 2021a,b] as training data. During training, we randomly augment each training sample by horizontal mirroring and/or rotation in 90° steps, in addition to applying a random permutation on the order of furniture objects other than the room, windows and doors. For inference, we follow a similar approach to the strategy proposed by Sceneformer [Wang et al. 2020], using top-p nucleus sampling with $p = 0.9$ for the object categories, as well as the attributes of the room, doors and windows. For the attributes of other object categories, we always pick the token with the highest probability. We also check for intersections after sampling each furniture object and re-sample the current object if it cannot be inserted into the layout without intersecting other objects.

5 RESULTS AND EVALUATION

5.1 Ablation

To evaluate the influence of our proposed ergonomic loss, we define 3 ablations of our network that are trained with different loss functions. Recall that the total loss function of our approach given in Eq. 2 is defined as the weighted sum of the cross-entropy loss $\mathcal{L}_T$ and the ergonomic loss $\mathcal{L}_E$ with weights β_T, β_E. Using these weight parameters, we define the following 3 ablations of our network:

- Baseline, with $\beta_T = 1$ and $\beta_E = 0$,
- Weight-only, with $\beta_T = 1 - E\left(S^k\right)$ and $\beta_E = 0$,
- Loss-only, with $\beta_T = 1$ and $\beta_E = 1$.

In other words, the baseline model only uses the cross-entropy loss with each input sample having equal weight and is thus without any of our enhancements. The weight-only model uses the cross-entropy loss with each sample being weighted by its ergonomic loss, while the loss-only model uses the sum of cross-entropy loss and ergonomic loss with each input sample having equal weight.

Figure 7 depicts the cross-entropy loss and ergonomic loss evaluated on both the training and validation sets for each version, using the Bedroom dataset for training. The results show a decrease in ergonomic loss for both the loss-only model and our full model which make use of our ergonomic loss term during training. While

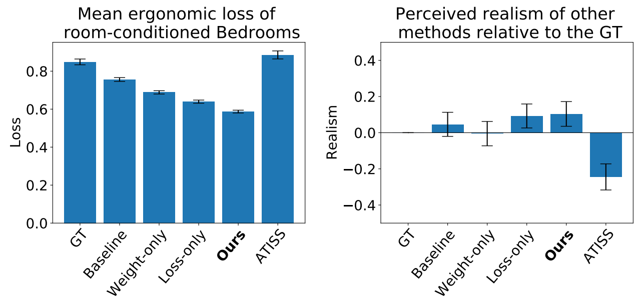


Fig. 8. Room-conditioned layout synthesis. We synthesize 20 layout variations for each floor plan in the Bedrooms validation set and evaluate the ergonomic loss. The left chart shows the mean ergonomic loss of the synthesized layouts, with the 80% confidence interval of the mean shown in black. The realism of the synthesized layouts is evaluated in a user study. The right chart shows how the layouts synthesized using each method are perceived compared to the ground truth, with a negative value meaning that the ground truth is seen as more realistic. Our proposed approach improves the ergonomic loss of the scenes, while also being perceived as more realistic than the ground truth.

the decrease may seem small relative to the overall loss, please keep in mind that the loss is computed for the entire scene with only one token predicted by the network. The weight-only model only yields a small decrease of ergonomic loss during training, since weighting the training samples by their ergonomic loss only reduces the influence of bad training samples without teaching the network how to improve the sample. However, this still has a noticeable effect on the synthesized scenes as we will discuss in Section 5.2. Please note that our loss-only model and our full model exhibit a higher cross-entropy loss for both training and validation set. This result is expected, since we aim to improve the ergonomic qualities of the synthesized layouts instead of perfectly recreating the distribution of the dataset.

5.2 Room-conditioned Layout Synthesis

We use our proposed model and its ablations introduced in the previous section for layout synthesis and evaluate the results in terms of both realism and ergonomic loss. In order to evaluate the realism of our generated results, we perform a perceptual study using Amazon Mechanical Turk in which we ask participants to compare pairs of Bedroom layouts with the question of which layout is more realistic on a 7-point scale. We compare layouts from 6 sources in this study: the ground truth layouts from the 3DFRONT dataset [Fu et al. 2021a,b], layouts generated with our proposed model and its ablations, and another state-of-the-art method ATISS [Paschalidou et al. 2021], which we train using the code provided on their website, modified to include windows and doors in the same manner as our model. In each layout pair, a synthesized layout is compared to a ground truth layout. A total of 330 users participated in the study. Each pair of layouts was shown 3 times to 10 different users each for a total of 30 comparisons per layout pair.

The left side of the Figure 8 shows the mean ergonomic loss of all layouts created for the user study. As can be seen, our approach performs the best at generating layouts with lower ergonomic loss, reducing the mean ergonomic loss by 30.8% compared to the ground truth data. The ablations of our model also improve the ergonomic loss to a lesser extend, including the baseline model which we attribute to our sampling strategy making it less likely to generate arrangements that are learned from outliers in the training data. On the other hand, layouts created with ATISS show the highest ergonomic loss because the layouts are perceived as less realistic than even our baseline model.

This can be seen on the right side of Figure 8 which shows how the users perceive the realism of synthesized layouts compared to those of the ground truth in a range of $[-1, 1]$, with a negative value meaning that the ground truth is seen as more realistic. The responses show that ATISS is considered significantly less realistic than the ground truth. On the other hand, the layouts generated by all our models are seen as at least equally realistic as the ground truth layouts, with users even preferring layouts created with our full model over the ground truth. This shows that our approach can not only improve the ergonomic quality in a purely quantitative sense, but also improve the perceived realism of the layouts.

A qualitative comparison is shown in Figure B.16. While all of the methods produce plausible layouts, our approach generates, on average, layouts with fewer ergonomic issues like missing light sources or poor accessibility. Layouts sampled unconditionally for multiple room categories are shown in Figure 10. In these examples, all layout elements including the rooms, doors and windows are generated by the network.

6 LIMITATIONS AND CONCLUSIONS

6.1 Limitations

Our proposed approach has a number of limitations. Designing layouts is a complex high dimensional problem that includes modalities including selecting 3D furniture model that fit well together stylistically [Lun et al. 2015; Weiss et al. 2020]; architectural elements such as room shapes walls and floor plans [Wu et al. 2019]; and various other aspects of lighting and illumination conditions [Vitsas et al. 2020]. While important, such methods are orthogonal to our scope layout synthesis focused scope.

Furthermore, while our ergonomic loss functions are derived from ergonomics literature, they are only theoretical models and and have not been evaluated in a real-life setting. We think that the problem of translating the vast number of ergonomic rules and interior design guidelines into differentiable functions can be a promising topic of further research [Schwartz 2021].

While we have demonstrated that our approach of incorporating expert knowledge into the Transformer training process produces promising results, we think that this is only the first step in combining data-driven and rule-based learning using state-of-the-art deep-learning models such as Transformers. We believe that future research in this direction can assist with making data-driven learning approaches more applicable to domains where large amounts of high-quality data with desired properties are not readily available.

6.2 Conclusions

We presented a novel method for the synthesis of indoor layouts, which combines data-driven learning and manually designed expert

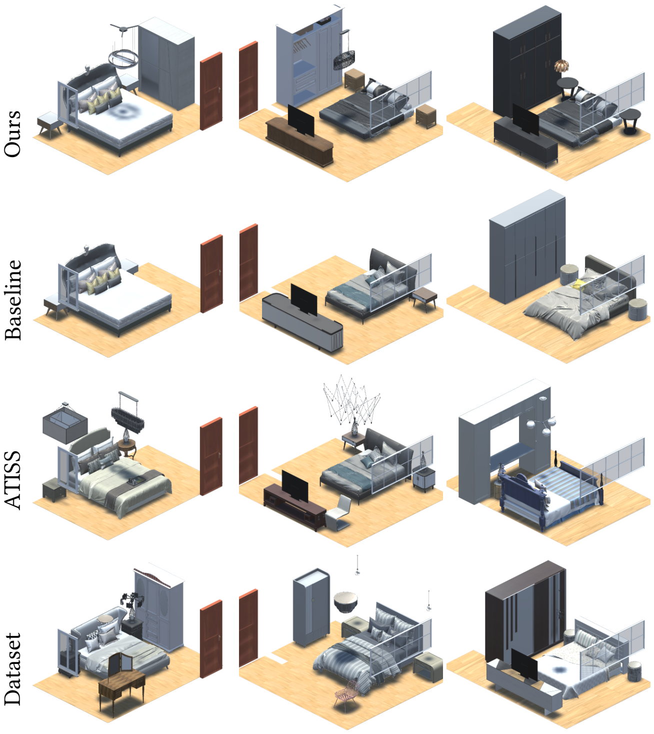


Fig. 9. Conditional synthesis results as described in Section 5. Methods in a column receive the same room boundary, windows, and doors as input condition. Our approach produces on average layouts with less ergonomic issues like missing light sources (e.g. Baseline columns 1, 2, 3) and poor accessibility (e.g. blocked path in ATISS column 3).

knowledge. To our knowledge, we are the first to propose such a solution to the problem. The main benefit of our approach is that it allows emphasizing features that might be underrepresented or not contained at all in the data. Simultaneously, we maintain the benefits of a data-driven approach which is important for layout generation which is high-dimensional and ill-defined. Manually crafting all design rules needed to synthesize comparable results would be very difficult and time consuming. Combining both expert knowledge and a distribution learned from data gives us the benefits from both worlds.

As a technical contribution, we proposed a modern Transformer network that can be trained using a loss function composed of cross-entropy and additional knowledge. We have shown that weighting the two loss terms on a per-sample basis leads to results that fulfill the additional objective well and still maintain a high degree of realism. Further, we introduced expert knowledge in the form of cost functions derived from ergonomics, whose goal is to improve layouts to be more usable and comfortable for humans.

We described the details of our implementation (we will release our code on GitHub), and we evaluated the method thoroughly. We showed numerical quantitative results and performed a perceptual study where our model out-performs recent related work. We also used our system to synthesize a large set of realistically looking results. Our method is meant to help professionals and amateurs in the future to address the problem of interior layout design.

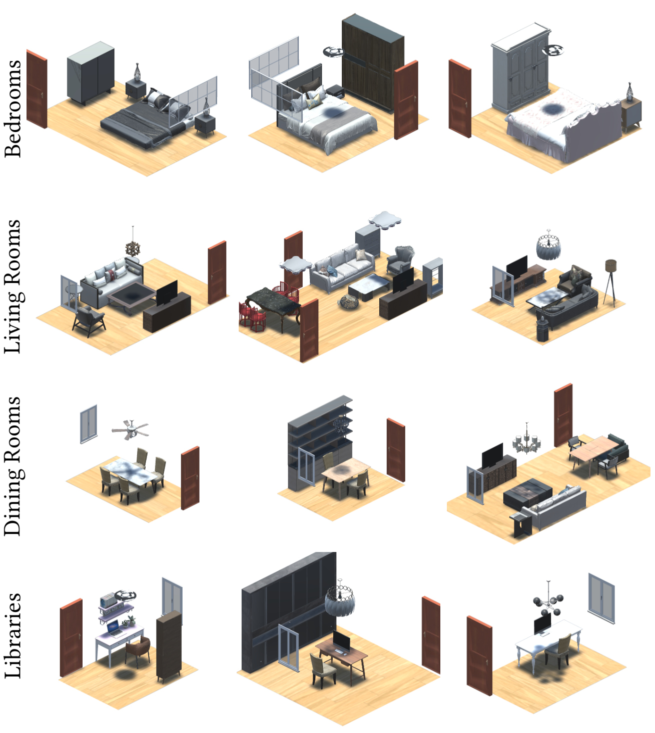


Fig. 10. Generated layouts for different room types. Since the attributes of the rooms were represented as part of the input sequences during training, all layout elements including rooms, doors, and windows can be generated by the network. Our method can generate furniture arrangements typical for each room type even with small training sets.

ACKNOWLEDGMENTS

This research was funded by NJIT's faculty startup funds, gifts from Adobe, and by Austrian Science Fund FWF P 29981 (https://www.fwf.at). In addition, the authors would like to thank Stefan Pillwein for his help with paper production.

REFERENCES

Esra Ataer-Cansizoglu, Hantian Liu, Tomer Weiss, Archi Mitra, Dhaval Dholakia, Jae-Woo Choi, and Dan Wulin. 2019. Room style estimation for style-aware recommendation. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 267–2673.

Aditya Chattopadhyay, Xi Zhang, David Paul Wipf, Himanshu Arora, and René Vidal. 2022. Structured Graph Variational Autoencoders for Indoor Furniture layout Generation. *ArXiv* abs/2204.04867 (2022).

Xinhan Di, Pengqian Yu, Hong Zhu, Lei Cai, Qiuyan Sheng, Changyu Sun, and Lingqiang Ran. 2020. Structural Plan of Indoor Scenes with Personalized Preferences. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12538 LNCS. https://doi.org/10.1007/978-3-030-66823-5_27

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. 2012. Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–11.

Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric scene synthesis for functional 3D scene modeling. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–13.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 2021a. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10933–10942.

Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 2021b. 3d-future: 3d furniture shape with texture. *International*

Journal of Computer Vision (2021), 1–25.

Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. 2017. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. https://doi.org/10.48550/ARXIV.1512.03385

Z Sadeghipour Kermani, Zicheng Liao, Ping Tan, and H Zhang. 2016. Learning 3D Scene Synthesis from Annotated RGB-D Images. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 197–206.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe, Nevada) *(NIPS'12)*. Curran Associates Inc., Red Hook, NY, USA, 1097–1105.

Karl H.E. Kroemer. 2017. *Fitting the Human: Introduction to Ergonomics / Human Factors Engineering, Seventh Edition.* CRC Press.

Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. 2019. Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)* 38, 2 (2019), 1–16.

Wei Liang, Jingjing Liu, Yining Lang, Bing Ning, and Lap-Fai Yu. 2019. Functional Workspace Optimization via Learning Personal Preferences from Virtual Experiences. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 1836–1845.

Zhaoliang Lun, Evangelos Kalogerakis, and Alla Sheffer. 2015. Elements of style: learning perceptual shape style similarity. *ACM Transactions on graphics (TOG)* 34, 4 (2015), 1–14.

Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B. Tenenbaum. 2020. End-to-end optimization of scene layout. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* https://doi.org/10.1109/CVPR42600.2020.00381

Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive furniture layout using interior design guidelines. *ACM transactions on graphics (TOG)* 30, 4 (2011), 1–10.

Wamiq Para, Paul Guerrero, Tom Kelly, Leonidas J Guibas, and Peter Wonka. 2021. Generative layout modeling using constraint graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 6690–6700.

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. 2021. ATISS: Autoregressive Transformers for Indoor Scene Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS).*

X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. 2018. VirtualHome: Simulating Household Activities Via Programs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society, Los Alamitos, CA, USA, 8494–8502. https://doi.org/10.1109/CVPR.2018.00886

Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. 2018. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 5899–5908.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

Mathew Schwartz. 2021. Human centric accessibility graph for environment analysis. *Automation in Construction* 127 (2021), 103557.

Turbosquid. 2022. 3D Model Collection. https://www.turbosquid.com/. [Online; accessed Jan-2022].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems.* 5998–6008.

Nick Vitsas, Georgios Papaioannou, Anastasios Gkaravelis, and Andreas-Alexandros Vasilakis. 2020. Illumination-Guided Furniture Layout Optimization. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 291–301.

Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2019. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2018. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. 2020. SceneFormer: Indoor Scene Generation with Transformers. *arXiv preprint arXiv:2012.09793* (2020).

Wayfair. 2022. Room Planner. https://www.wayfair.com/RoomPlanner3D. [Online; accessed Jan-2022].

Tomer Weiss, Alan Litteneker, Noah Duncan, Masaki Nakada, Chenfanfu Jiang, Lap-Fai Yu, and Demetri Terzopoulos. 2018. Fast and scalable position-based layout synthesis. *IEEE Transactions on Visualization and Computer Graphics* 25, 12 (2018), 3231–3243.

Tomer Weiss, Ilkay Yildiz, Nitin Agarwal, Esra Ataer-Cansizoglu, and Jae-Woo Choi. 2020. Image-Driven Furniture Style for Interactive 3D Scene Modeling. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 57–68.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. 2019. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–12.

Haitao Yang, Zaiwei Zhang, Siming Yan, Haibin Huang, Chongyang Ma, Yi Zheng, Chandrajit Bajaj, and Qixing Huang. 2021. Scene Synthesis via Uncertainty-Driven Attribute Synchronization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 5630–5640.

Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86* 30, 4 (2011).

Yongqi Zhang, Haikun Huang, Erion Plaku, and Lap-Fai Yu. 2021. Joint computational design of workspaces and workplans. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–16.

Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. 2020. Deep generative modeling for scene synthesis via hybrid representations. *ACM Transactions on Graphics (TOG)* 39, 2 (2020), 1–21.

Yang Zhou, Zachary While, and Evangelos Kalogerakis. 2019. SceneGraphNet: Neural Message Passing for 3D Indoor Scene Augmentation. In *Proceedings of the IEEE International Conference on Computer Vision.* 7384–7392.

A IMPLEMENTATION DETAILS

A.1 Ergonomic rules

We consider the following ergonomic rules which are expressed as scalar cost functions in the range of $[0, 1]$, where a lower value indicates a better score: (1) Reach, (2) Visibility, (3) Lighting, (4) Glare and (5) accessibility. In this section, we first describe the individual ergonomic cost functions for each rule, followed by the activities we use to evaluate the layouts.

A.1.1 Reach. While being seated, a person only has limited mobility and thus objects that need to be interacted with should be within a distance that is easy to reach without the need to stand up. We can broadly categorize the area around a seated person into 3 zones. In the inner zone, objects can be reached without much effort, while objects in the outer zone are beyond reach. Objects in the middle zone can still be reached, but require more effort the further away they are. We model this reach cost E_R as a sigmoid function that measures how difficult it is to reach an object at position q from position p:

$$E_R = \frac{1.0}{1.0 + \exp\left(-\beta_R\left(\|q - p\| - d_R\right)\right)}\,. \tag{3}$$

The function is centered at d_R with scaling parameter β_R. We use $d_R = 0.8$ and $\beta_R = 15$ to model the zones of easy and extended reach. These parameters roughly correspond to an easy reach up to $0.5m$ up to which the cost is close to 0 and an extended reach up to $1.0m$, towards which the cost increases to 1.0.

A.1.2 Visibility. Visibility cost measures how visible an target object is from the viewpoint of the avatar given by position p and viewing direction u. This measure is important for activities like watching TV or using the computer (cf. Table 1), since seating furniture with sub-optimal positions or orientations may require the user to take on unhealthy postures. To introduce this cost as smooth scalar function E_v which can be minimized, we define the cost to increase with the angle between the two vectors u and $v = \frac{q-p}{\|q-p\|}$:

$$E_V = 1 - \left(\frac{1 + \langle u, v\rangle}{2}\right)\,. \tag{4}$$

A.1.3 Lighting. Lighting cost measures how well an object is illuminated by light sources in the room. Ideally, when looking at an object, the viewer and the light source should be positioned in the same half-space of the viewed object, as otherwise the object itself would partially obstruct the direct illumination and cause self-shadowing. A light source b_i is thus well suited for illuminating the object at position q when viewed from position p as long as the position-to-object vector $v = \frac{q-p}{\|q-p\|}$ and the vector $l_i = \frac{q-b_i}{\|q-b_i\|}$ pointing from a light source at position b_i to q do not point in opposite directions:

$$e_i^L = \left(1 - \frac{1 + \langle v, l_i\rangle}{2}\right)\,.$$

Since multiple light sources can contribute to this cost, we compute their contribution by applying the softmin function to the vector $e^l = [e_i^l]_{i \in B}$ and using them as weights for computing the weighted sum:

$$E_L = \langle e^l, \mathrm{softmin}(\beta \cdot e^l)\rangle, \tag{5}$$

with β being a temperature parameter that determines the hardness of the softmin function. We use $\beta = 10$. Since the computation of indirect illumination is prohibitively expensive, we only consider direct lighting.

A.1.4 Glare. Glare cost E_g measures the decrease in visual performance from strong brightness contrast caused by having bright light sources in the field of view. Given position-to-object vector $v = \frac{q-p}{\|q-p\|}$ and glare vector $g_i = \frac{b_i - p}{\|b_i - p\|}$ pointing from p to the light source at b_i, the cost increases as the angle between the vectors decreases:

$$e_i^G = \left(\frac{1 + \langle v, g_i\rangle}{2}\right)\,.$$

Similar to the lighting cost we compute the weighted sum of multiple light sources using the softmax function for computing the weights:

$$E_G = \langle e^g, \mathrm{softmax}(\beta \cdot e^g)\rangle\,. \tag{6}$$

For simplicity, we do not consider indirect glare, such as light sources that are reflected by a computer screen. Ceiling lights such as chandeliers are also excluded from this rule since light sources positioned above the field of view have a smaller impact on visual performance [Kroemer 2017].

A.1.5 Accessibility. The accessibility cost E_A measures how much space is available in front of a target object to allow easy interaction and walking through the room. For example, it is necessary to provide sufficient space between a bed and a wardrobe so that the wardrobe can be easily opened. We quantify this cost by defining an interaction region I_j for each object F_j that should not intersect with the bounding box A_k of any another object F_k in the layout. For most object categories, this region is located in front of the object itself, with a width equal to that of the object and an empirically chosen depth of $0.5m$. An exception is made for beds, since they are usually interacted with from the sides, so we define 2 such regions on either side with a width equal to half the depth of the bed and a depth of $0.5m$. Given a furniture object F_j with interaction region I_j, we define the accessibility cost E_a as

$$E_A = \sum_{k=0}^{N} \frac{|I_j \cap A_k|}{|I_j|}\,. \tag{7}$$

A.2 Activity Evaluation

We evaluate the ergonomic loss of a layout in the context of activities that are typically performed in rooms of a given category. Based on research on this topic [Puig et al. 2018], we select 4 such activities which we label as *Read book*, *Watch TV*, *Use computer* and *Work at desk*. To evaluate an activity, it is necessary to compute the ergonomic costs relevant to that activity (cf. Table 1). We furthermore use a logarithmic function to re-scale the ergonomic cost functions to more strongly punish scenes with high costs, for example

$$\bar{E}_R = -\ln(1.0 + \epsilon - E_R), \tag{8}$$

with the scaling functions for the other rules defined analogously. We use $\epsilon = \exp(5)$, so that when $E_R = 1$, then $\bar{E}_R = 5$. We found this scaling function to be beneficial for minimizing the ergonomic loss during network training.

Since the accessibility cost E_A is relevant for every activity, we decide to compute this term once for the entire layout instead of computing it separately for every activity for performance reasons. We thus define the accessibility cost for the entire layout as

$$E_{access} = \langle e^{access}, \mathrm{softmax}(\beta \cdot e^{access}) \rangle, \tag{9}$$

with $e^{access} = [\bar{E}_A(I_j)]_{j=1,\dots,N}$ being the vector containing the accessibility cost of every object and using $\beta = 10$. The softmax function is used to normalize the total cost of the layout such that a single badly-placed object increases the cost by roughly the same amount regardless of the number of objects in the layout.

For the activity *Read book*, proper illumination conditions are the most important factor, so we need to apply the rules for lighting and glare. Given the position p_j of seating furniture (like beds, chairs, or sofas), an associated object position q_j (a book close to p_j) and light sources B we define

$$e_j^{book} = \frac{\bar{E}_L(p_j, B, q_j) + \bar{E}_G(p_j, B, q_j)}{2}.$$

Since we do not require all possible positions to have a good score for every activity, we once again use the softmin function to compute a weighted sum of costs for the layout. That way, if there is only one position that is suitable for an activity, it will be the only one with a large contribution to the layout cost, while having multiple suitable positions will have them contribute equally. For a set of positions $p_j \in P$ we therefore have

$$E_{book} = \langle e^{book}, \mathrm{softmin}(\beta \cdot e^{book}) \rangle, \tag{10}$$

with $e^{book} = [e_j^{book}]_{j \in P}$ and using $\beta = 10$.

The other activities are defined similarly. For *Watch TV*, we require the TV to be visible from a piece of seating furniture and there should not be a light source in the field of view. We therefore compute the visibility and glare costs for positions p_j with orientation u_j (for chairs, beds, sofas) and TVs with position q_k:

$$e^{tv}_{j,k} = \frac{\bar{E}_V(p_j, u_j, q_k) + \bar{E}_G(p_j, B, q_k)}{2}.$$

Since there can be multiple TVs in a room in addition to multiple pieces of seating furniture, we need to compute the weighted sum of costs over every combination of p_j and q_k, using $e^{tv} = [e^{tv}_{j,k}]_{j \in P, k \in Q}$:

$$E_{tv} = \langle e^{tv}, \mathrm{softmin}(\beta \cdot e^{tv}) \rangle. \tag{11}$$

The same rules are required for the activity *Use computer*, in addition to the reach rule since the seating furniture and computer should be in close proximity. We do not evaluate the lighting rule because the direction from which the light illuminates the computer is not as important, since the computer screen is already illuminated. Using q_k to denote the positions of computers we define

$$e^{comp}_{j,k} = \frac{\bar{E}_V(p_j, u_j, q_k) + \bar{E}_G(p_j, B, q_k) + \bar{E}_R(p_j, q_k)}{3}.$$

Table 1. Associations of rules to activities that can be performed in an environment. Not all activities require all rules to be fulfilled.

	Reach	Visibility	Lighting	Glare	Accessibility
Read book			yes	yes	yes
Watch TV		yes		yes	yes
Use computer	yes	yes		yes	yes
Work at desk	yes	yes	yes		yes

Finally, for the activity *Work at desk* we apply the rules visibility, lighting and reach. Since the viewing angle is mostly directed downward toward the desk during this activity, it is not necessary to consider direct glare caused by light sources in the room. Given chair positions p_j, table positions q_k and light sources B we compute

$$e^{work}_{j,k} = \frac{\bar{E}_V(p_j, u_j, q_k) + \bar{E}_L(p_j, B, q_k) + \bar{E}_R(p_j, q_k)}{3}.$$

To obtain the overall **ergonomic loss** E for a layout we take the average of all activity costs that are possible in the layout (e.g. if there is no computer in the scene, we do not evaluate the cost for *Use computer*):

$$E = \frac{\sum_a \delta_a E_a}{\sum_a \delta_a},$$

with $a \in \{access, book, tv, comp, work\}$ and $\delta_a = 1$ if the corresponding activity can be performed in the layout and $\delta_a = 0$ otherwise.

A.3 Dataset and Training Details

A.3.1 Dataset. We train our models using the 3DFRONT dataset [Fu et al. 2021a,b] as training data. In a pre-processing step, we parse the data to extract rooms belonging to the categories Bedroom, Dining Room, Living Room and Library. For this purpose we use the filter criteria provided by ATISS [Paschalidou et al. 2021], consisting of a list of rooms for each category, as well as a split into training, testing and validation data. We use the rooms marked as *train* for our training sets and combine those marked as *test* and *val* for our validation sets. Depending on the use case, we apply also some additional filtering. If we want to provide the attributes of the room shape as part of the input sequence, we can only use rectangular rooms and thus filter out rooms with more complex shapes. For the Bedrooms dataset, this results in 4041 rooms for the training set and 324 rooms for the validation set. For the direct comparison with ATISS, we provide the room shape using a binary map of the floor plan, allowing us to also use non-rectangular rooms, but we need to exclude rooms that have also been filtered by the pre-processing algorithm of ATISS. This leaves in 3526 rooms for the training set and 289 rooms for the validation set.

For most furniture objects, their attributes such as the category and the transformation of the corresponding 3d model data can be directly extracted from the room data. Since separate 3d models for doors and windows are not provided with the dataset, we extract their positions and bounding box dimensions from the mesh data with corresponding labels. Since doors are only provided with each house and not attached to individual rooms, we include a door with the furniture objects of a room if its distance to the closest wall of the room is lower than a chosen threshold and its orientation is aligned with that of the wall.

Additionally, we group some of the object categories in the dataset that are very similar to each other, while filtering out some others that occur only in very few rooms, for a total of 31 categories that we use across all room types.

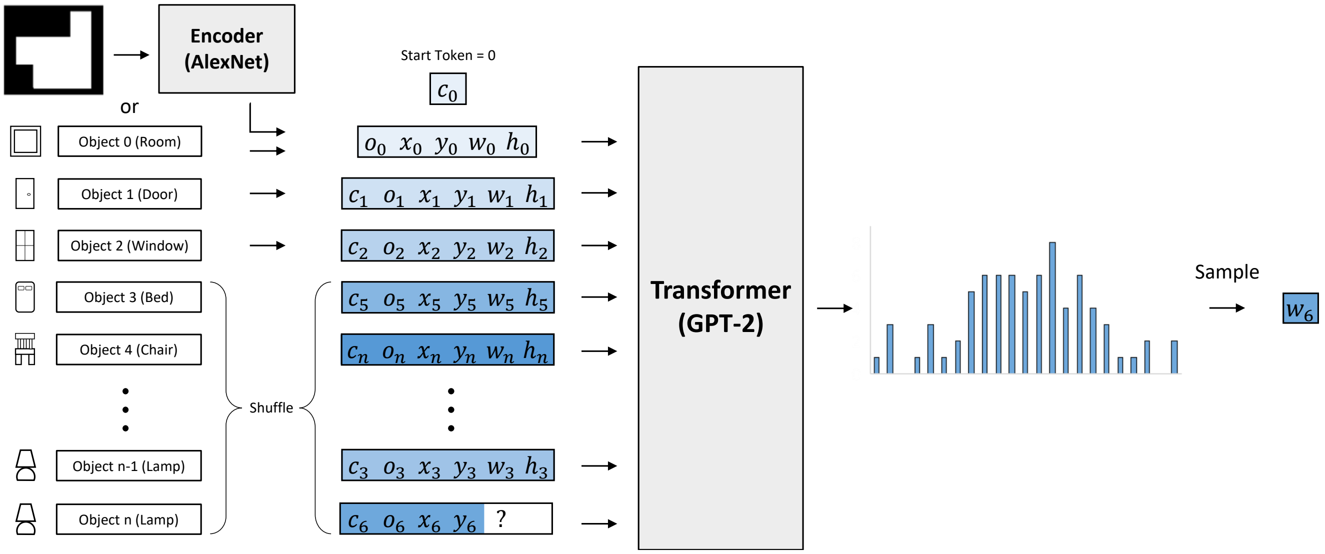


Fig. A.11. Overview of our model. A room layout consisting of individual furniture objects is mapped to a sequence of tokens which serves as the input to the transformer model. Given this sequence, the network predicts a categorical distribution for the next token from which we randomly sample the actual token value. During training, the order of objects other than the room, doors and windows is shuffled in the sequence. Furthermore, the attributes of the room can be either mapped to tokens directly (for rectangular rooms only), or by using an additional encoder network given a binary image of the floor plan as input.

Since the dataset is typically lacking object categories that are necessary to properly evaluate the ergonomic loss of a layout, we augment the dataset with additional objects in the following way. For each layout, there is a 50% chance to place a furniture object of the indoor lamp category in the center of every stand and side-table object. In the same manner, a computer object is placed at the center of each desk object in a layout with a probability of 50%. Finally, every TV stand object is augmented with a TV object.

A.3.2 Training. As hyperparameters for our networks we use 12 hidden layers, 8 attention heads, embedding dimensionality of 256, dropout probability of 0.1 and a batch size of 64. Each network is trained for 150 epochs, with the number of steps per epoch being equal to the number of training samples, such that each sample is used once per epoch. We use a learning rate of 0.0001 with a linear rate of decay and a warm-up period of 10 epochs. These parameters were determined empirically in preliminary experiments. For layout synthesis, we always choose the learned network parameters of the epoch with the smallest validation loss during training.

When the shape of the room is provided as a binary map of the floor plan, we use an additional convolutional neural network to convert the binary map into a feature embedding. For this network we directly use the implementation of the AlexNet architecture [Krizhevsky et al. 2012] provided in the code framework of ATISS [Paschalidou et al. 2021]. We also experimented with a ResNet-18 architecture [He et al. 2015], but found that AlexNet is more successful at discriminating between mirrored floor plans. The computed feature embedding then replaces the embedding that is otherwise computed from the orientation, position and dimension tokens of the room furniture object.

Our networks are trained on Google Colab, using a machine with a NVIDIA Tesla P100 GPU. When only using the cross-entropy loss, training for one epoch takes 13 seconds on average. Adding our ergonomic loss increases training times to 123 seconds per epoch on average, since we cannot make use of parallelization for layout evaluation as easily. There is room for further optimizations in this aspect.

A.3.3 Inference. During inference, we follow a similar approach to the strategy proposed by Sceneformer [Wang et al. 2020], using top-p nucleus sampling with $p = 0.9$ for the object categories, as well as the attributes of the room, doors and windows. For the attributes of other object categories, we always pick the token with the highest probability.

The layouts synthesized by the transformer network sometimes include intersecting objects which greatly disturb the perceived realism of a layout. We therefore follow the approach of similar methods like Sceneformer and check for object intersections during inference. After the attributes of a furniture object have been generated, we check if the object can be inserted into the scene without causing large intersections. If this is not the case, we re-sample the category and other attributes of the current object. If this re-sampling approach fails too often (we choose a limit of 20 attempts experimentally), we discard the entire layout and start anew. Certain pairs of object categories are excluded from this check, e.g. chairs can be put underneath a table and thus do not cause collisions.

In terms of computation time, the intersection-detection process is the bottleneck of the inference process. If we do check for intersection during inference, it takes 1653 seconds for our models to synthesize 1000 layoutsequences, for 1.653 seconds per layout on average. If we do not perform intersection-checks between objects,

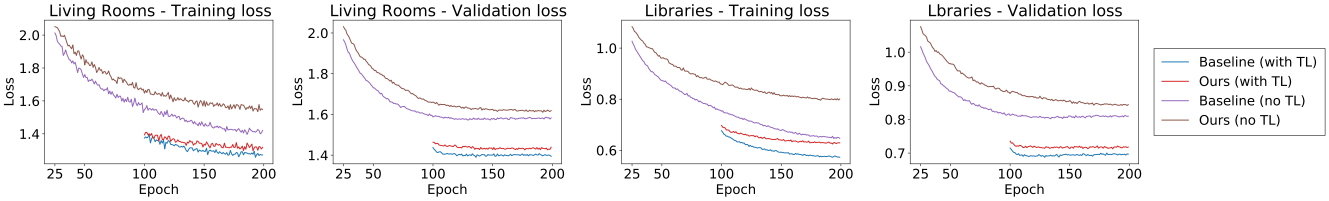


Fig. A.12. By pre-training the network on a general dataset containing samples from all room types and then fine-tuning the network for a specific room type, the validation loss can be decreased significantly, especially for small datasets.

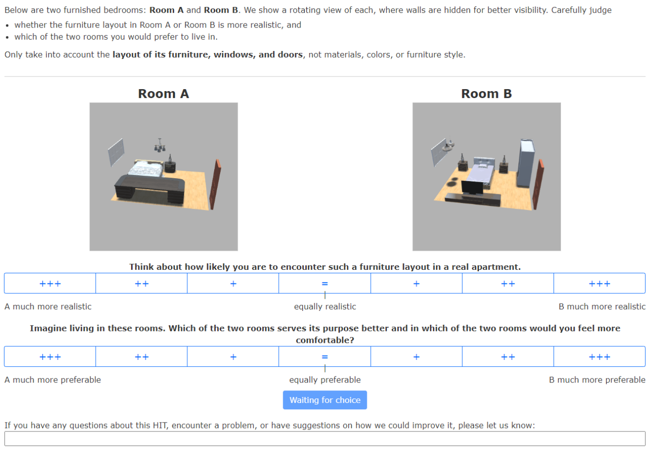
we can make use of parallelization to greatly reduce inference time. In such a setup, our networks can synthesize 1000 layout sequences in 27 seconds for 0.027 seconds per scene on average.

A.3.4 Scene reconstruction. Since our networks only generate the 2d bounding boxes of furniture objects, we use an additional post-processing step to reconstruct a 3d scene from the generated layout. For each furniture object, we select the 3d model of the same category with the smallest difference in bounding box dimensions from the models in the 3DFRONT dataset [Fu et al. 2021a,b]. For categories not included in the dataset, such as doors and windows, we handpick a few suitable models from online sources [Turbosquid 2022].

As a final step, the vertical position of each object is adjusted based on its category. The position of some categories like windows and chandeliers are set to a fixed height. We label some categories as supporting objects (like tables and stands) and others as supported objects (like indoor lamps and TVs). If there is an intersection between a supporting and supported object, the vertical position of the supported object is adjusted to be placed on top of the supporting object.

A.4 Perceptual Study Setup

In order to evaluate the realism of our generated results, we perform a perceptual study using Amazon Mechanical Turk in which we ask participants to compare pairs of Bedroom layouts with the question of which layout is more realistic.


Below are two furnished bedrooms: **Room A** and **Room B**. We show a rotating view of each, where walls are hidden for better visibility. Carefully judge

- whether the furniture layout in Room A or Room B is more realistic, and
- which of the two rooms you would prefer to live in.

Only take into account the **layout of its furniture, windows, and doors**, not materials, colors, or furniture style.

Think about how likely you are to encounter such a furniture layout in a real apartment.

Imagine living in these rooms. Which of the two rooms serves its purpose better and in which of the two rooms would you feel more comfortable?

If you have any questions about this HIT, encounter a problem, or have suggestions on how we could improve it, please let us know:

Fig. A.13. The interface of the user study. Participants were asked which of the 2 displayed scenes is more realistic.

To allow for a direct comparison, we use the binary map of the floor plan and the attributes of the doors and windows from the ground truth data for each layout and only generate the rest of the furniture objects using the selected methods. For each of the 289 partial layouts in the validation set consisting of binary map, doors, and windows, we generate 20 furniture layout variations using both ATISS and our trained networks, resulting in a total of 5780 sets of 6 layouts each, one for each method/variant. Since ATISS does not handle any intersections between furniture objects and even some of the ground truth layouts may contain such intersections, we discard the entire set if one of its layouts contains an intersection between furniture objects larger than a threshold, which we set as 20% of the smaller bounding box area. For our networks, we perform intersection-checks during inference, only discarding a set if an intersection-free layout cannot be generated after 20 attempts. For comparison, ∼ 75% of ATISS layouts contain bounding box intersections, compared to ∼ 48% of layouts generated by our model with intersection check turned off. Additionally, ∼ 56% of the ground truth layouts also contain bounding box intersections. These numbers likely include some false positives where the bounding boxes intersect but the 3d meshes do not. We decided to be more conservative since intersections can significantly influence the perceived realism. Furthermore, since both ATISS and our model may try to generate additional windows or doors, we simple resample the category in such a case.

For the user study, we randomly select 50 sets from all sets of synthesized layouts and ask users to compare the layouts in terms of realism. In each comparison, the user is shown a pair of layouts from the same set, each represented by an animated 3d rendering with the camera rotating around the scene. Users are asked which layout is more realistic on a 7-point scale (ranging from *Scene A much more realistic* to *Scene B much more realistic*, including a neutral *Equally realistic* option). Figure A.13 shows the screenshot of the UI. Each user sees the same scenes three times, and we use this redundancy to keep track of each user's consistency. We discard users that chose options more than two points apart for the same scene in more than 10% of their comparisons. Additionally, we discard users that spent less than 10 second on average per comparison. To evaluate the results, we compute a realism score for each method, that we obtain by assigning scores from −1 to 1 to the 7 possible user choices and averaging over all comparisons.

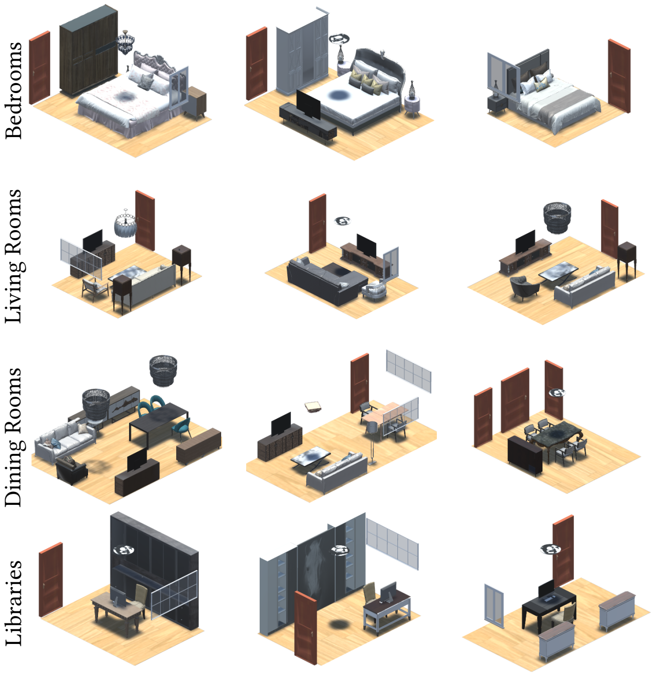


Fig. B.14. Additional generated layouts for different room types. Since the attributes of the rooms were represented as part of the input sequences during training, all layout elements including rooms, doors, and windows can be generated by the network. Our method can generate furniture arrangements typical for each room type even with small training sets.

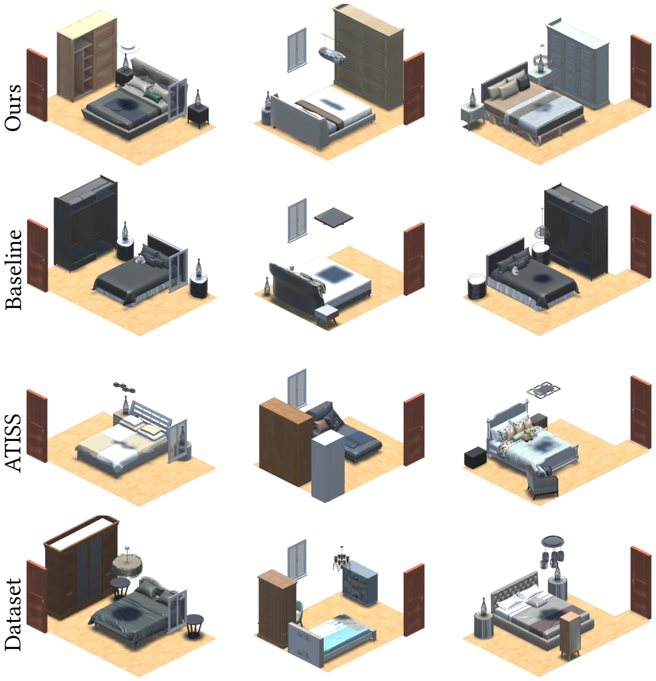


Fig. B.15. Additional qualitative comparisons of conditional synthesis results. Methods in a column receive the same room boundary, windows, and doors as input condition. Our approach produces on average layouts with less ergonomic issues like missing light sources and poor accessibility.

A.5 Additional Room Types

Since some room types in the 3DFRONT dataset only contain few samples (588 living rooms, 554 dining rooms and 424 libraries for training set after our pre-processing), we make use of a transfer learning strategy. We first train a base model containing training samples of all room types for 100 epochs using a learning rate of 1×10^{-4}. This base model is then fine-tuned for each room type using a learning rate of 2×10^{-5} to prevent overfitting to the smaller datasets.

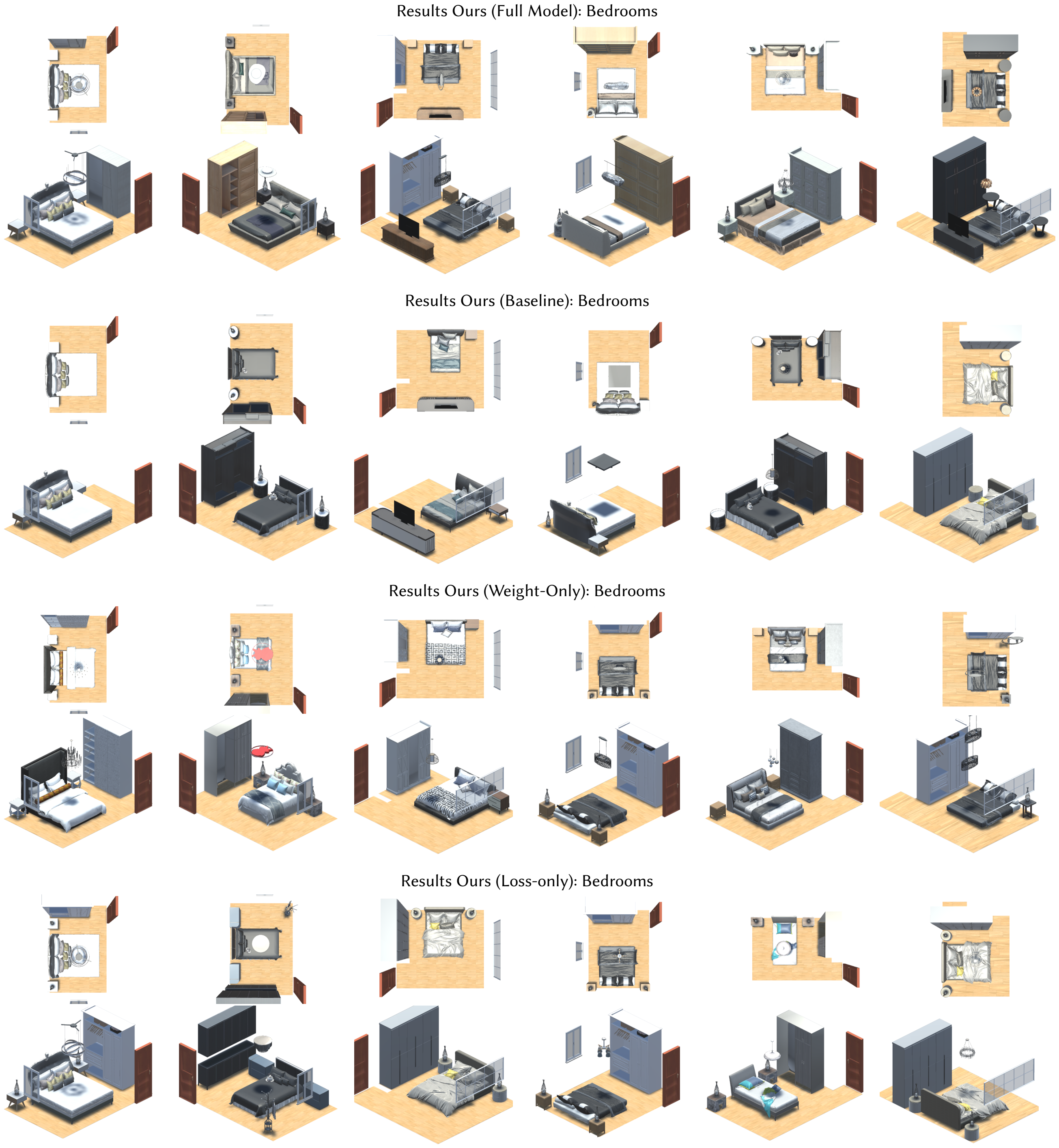
To evaluate the effectiveness of this approach, we train networks from scratch using only the training data from each individual room category and compare the cross-entropy loss to that of our networks which are first trained on a general set of training data before being fine-tuned for a room category. Figure A.12 shows that the transfer learning strategy already yields a lower training and validation loss after the first epoch of fine-tuning. While the training loss for networks that are trained from scratch eventually approaches that of the pre-trained network, the validation loss remains higher throughout. As can be seen, for small training datasets, transfer learning proves to be a good strategy for improving the training process.

B ADDITIONAL QUALITATIVE RESULTS

Please see Figures B.14 and B.15 for additional qualitative results supplementing those shown in the paper. Additionally, Figure B.16 shows additional qualitative results for our model and its ablations.

C SOURCE CODE

The author's source code is available at https://github.com/kleimerTU/HumanCentricLayouts



Results Ours (Full Model): Bedrooms

Results Ours (Baseline): Bedrooms

Results Ours (Weight-Only): Bedrooms

Results Ours (Loss-only): Bedrooms

Fig. B.16. Room-conditioned synthesis results for our model and its ablations.